# Lecture Notes in Computer Science 4073

*Commenced Publication in 1973*
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Andreas Butz Brian Fisher
Antonio Krüger Patrick Olivier (Eds.)

# Smart Graphics

6th International Symposium, SG 2006
Vancouver, Canada, July 23-25, 2006
Proceedings

Springer

Volume Editors

Andreas Butz
Ludwig-Maximilians-Universität München
Institut für Informatik, LFE Medieninformatik
Amalienstrasse 17, 80333 München, Germany
E-mail: butz@ifi.lmu.de

Brian Fisher
Simon Fraser University at Surrey
13450 102 Ave, Surrey BC V3T 5X3, Canada
E-mail: fisher@cs.ubc.ca

Antonio Krüger
Westfälische Wilhelms-Universität
Robert-Koch-Str. 26-28, 48149 Münster, Germany
E-mail:antonio.krueger@uni.muenster.de

Patrick Olivier
University of Newcastle Upon Tyne
Information Research Institute
Newcastle upon Tyne NE1 7Ru, UK
E-mail: p.l.olivier@ncl.ac.uk

Library of Congress Control Number: 2006928506

CR Subject Classification (1998): I.3, I.2.10, I.2, I.4

LNCS Sublibrary: SL 6 – Image Processing, Computer Vision, Pattern Recognition, and Graphics

ISSN 0302-9743
ISBN-10 3-540-36293-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-36293-7 Springer Berlin Heidelberg New York



Springer is a part of Springer Science+Business Media

springer.com


Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11795018 06/3142 5 4 3 2 1 0

# Preface

The International Symposium on Smart Graphics 2006 was held during July 23–25, 2006, at the University of British Columbia in Vancouver, Canada. It was the seventh event in a series which originally started in 2000 as an AAAI Spring Symposium.

In response to the overwhelming success of the 2000 symposium, its organizers decided to turn it into a self-contained event. With the support of IBM, the first two International Symposia on Smart Graphics were held at the T.J. Watson Research Center in Hawthorne, New York, in 2001 and 2002. The 2003 symposium moved to the European Media Lab in Heidelberg. Since then the conference has alternated between North America and Europe. It was held at Banff Alberta Canada in 2004 and at the cloister Frauenwörth on the island of Frauenchiemsee in Germany in 2005.

The core idea behind these symposia is to bring together researchers and practitioners from the field of computer graphics, artificial intelligence, cognitive science, graphic design and the fine arts. Each of these disciplines contributes to what we mean by the term "Smart Graphics": the intelligent process of creating effective, expressive and esthetic graphical presentation. While artists and designers have been creating communicative graphics for centuries, artificial intelligence focuses on automating this process by means of the computer. While computer graphics provides the tools for creating graphical presentations in the first place, the cognitive sciences contribute the rules and models of perception necessary for the design of effective graphics. The exchange of ideas between these four disciplines has led to many exciting and fruitful discussions and the Smart Graphics symposia draw their liveliness from a spirit of open minds and the willingness to learn from and share with other disciplines

Many Smart Graphics symposia emphasize a particular aspect of the field in the call for papers. In a wrap-up session in 2005, workshop participants identified three key challenges for Smart Graphics that formed the basis for the 2006 workshop: (a) to understand human reasoning with visual representations, (b) in human decision support, to reconcile the complexity of problems that must be solved with the simplicity of representation and interaction that is desired by users, and (c) to build systems that can reason about and change their own graphical representations to meet the needs and abilities of their users and the nature of the information they present.

Accordingly this year's SG emphasized the "smart" in Smart Graphics. This includes human individual, group, and distributed cognition as well as artificial intelligence applied to the design and testing of graphically rich systems: smart design, smart systems, and systems for smart users. In order to facilitate interaction with the AI and Cogsci communities, we co-located SG with the

28th Annual Meeting of the Cognitive Science Society and the IEEE World Congress on Computational Intelligence.

We would like to thank all authors for the effort that went into their submissions, the Program Committee for their work in selecting and ordering contributions for the final program, and of course the participants who made Smart Graphics 2006 such a success.

Juli 2006

Andreas Butz
Brian Fisher
Antonio Krüger
Patrick Olivier

# Organization

## Organization Committee

Andreas Butz (University of Munich, Germany)
Brian Fisher (University of British Columbia, Canada)
Antonio Krüger (University of Münster, Germany)
Patrick Olivier (University of Newcastle Upon Tyne, UK)

## Program Committee

Elisabeth André (University of Augsburg)
William Bares (Millsaps College)
Marc Cavazza (Teesside University)
Marc Christie (University of Nantes)
Sarah Diamond (Banff Centre)
Steven Feiner (Columbia University)
Sidney Fels (University of British Columbia)
Knut Hartmann (University of Magdeburg)
Rainer Malaka (University of Bremen)
Shigeru Owada (Sony Computer Science Laboratory, Tokyo)
W. Bradford Paley (Digital Image Design)
Bernhard Preim (University of Magdeburg)
Thomas Rist (University of Applied Sciences, Augsburg)
Stefan Schlechtweg (University of Magdeburg)
Lucia Terrenghi (University of Munich)
Sha Xin Wei (Concordia University)
Kay C. Wiese (Simon Fraser University)
Massimo Zancanaro (ITC-irst Trento)
Michelle Zhou (IBM T.J. Watson Research Center)

## Supporting Institutions

The Smart Graphics Symposium 2005 was held in cooperation with Eurographics, AAAI and ACM Siggraph and the University of British Columbia.

# Table of Contents

## Intelligent Text Processing



## Perceptive Systems



## Smart Visualization

## Visual Features, Sketching and Graphical Abstraction



## Intelligent Image and Film Composing



## Smart Interaction

## Short Papers

# Efficient View Management for Dynamic Annotation Placement in Virtual Landscapes

Stefan Maass and Jürgen Döllner

Department of Computer Graphics Systems
University of Potsdam, Hasso-Plattner-Institute
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
{maass, doellner}@hpi.uni.potsdam.de

**Abstract.** We present a dynamic placement technique for annotations of virtual landscapes that is based on efficient view management. Annotations represent textual or symbolic descriptions and provide explanatory or thematic information associated with spatial positions. The technique handles external annotations as 2.5 dimensional objects and adjusts their positions with respect to available space in the view-plane. The approach intends to place labels without occlusions and, if this cannot be achieved, favors those annotations that are close to the observer. This technique solves the visibility problem of annotations in an approximate but user-centric way. It operates in real-time and therefore can be applied to interactive virtual landscapes. Additionally, the approach can be configured to fine tune the trade off between placement quality and processing time with a single parameter.

## 1 Introduction

Annotations are essential elements to enhance depictions with meta information such as explanations and thematic information. While annotation techniques are well studied and developed for traditional static two-dimensional media, e.g., in geographic maps or medical illustrations, annotation techniques for dynamic three-dimensional virtual environments still represent an important challenge for computer graphics and visualization.

The depiction of annotations in interactive virtual 3D environments shows fundamental difficulties because annotations by their nature are not inherently three-dimensional geometric objects and, therefore, cannot directly be represented as regular scene elements. Partly, these difficulties arise from the general problem of integrating text and image representations in a perceptive and cognitive efficient as well as aesthetically convincing way.

In this paper, we present a new technique for the management of annotations in virtual landscapes such as 3D maps, 3D landscape models, or 3D city models. By a virtual landscape we refer to a virtual 3D environment that contains as predominant element a terrain surface. The annotations refer to point features of these landscapes. The technique handles annotations as 2.5 dimensional objects of the scene and adjusts their positions with respect to available space in the view-plane (cp. Fig. 1). Our technique intends to place annotations without occlusions. If this cannot be achieved, it favors those annotations that are close to the observer. The view management

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 1 – 12, 2006.


**Fig. 1.** Sample virtual landscape with dynamically placed annotations

operates in real-time and, therefore, can be applied to interactive virtual landscapes We used scene integrated annotations for our approach. They provide the important property of a depth cue because an annotation's size depends on its 3D position in the virtual landscape.

Applications of our approach include the annotation of spatial objects and spatially referenced data. In particular, we can label features of high interest or mark inspected positions or regions. Additionally, annotations play an import role in interactive collaborative geovirtual environments, and are required to include spatial comments and explanations.

## 2 Related Work

In general, we can differentiate between internal and external annotation techniques. An internal annotation is drawn inside the visual representation of the referenced object and partially obscures that object. Internal annotations directly establish a mental link between annotation and annotated object. They are preferably used if the depiction of their referenced object offers sufficient space to enclose the annotation and does not distort the depiction. To share the space between an image and text explanations Chigona et al. [4] extend this to the principle of "dual use of image space", where a depiction can transform between the representation as an image and the representation as text. A technique for dynamic placement of object-integrated internal annotations of typical objects of geovirtual environments (e.g., buildings) is presented in [12].

An external annotation is drawn outside the visual representation of the referenced object and uses a connecting element such as a line or arc to associate the annotation with the referenced object. To avoid ambiguities, crossings or long distances between the object and the annotation, therefore, should be avoided. External annotations are preferably used to annotate small objects as well as large numbers of objects or to group objects spatially by a specific topic.

General criteria for the quality of an annotation technique include non-overlapping placement, the support of annotations with different priorities, interactive placement, and aesthetic label layout [1], [6], [10].

### 2.1 Label Placement Techniques

In cartography, the static label placement for point, line, and area features represents a classical problem and has been investigated yet for a long time, where typically text is integrated into 2D maps; for a survey of algorithms see [5]. For some detailed labeling problems it was shown that finding an optimal solution is NP-hard [13].

To achieve a high quality annotation placement, criteria such as disambiguation, selectivity, and expressivity of annotations are approximated [6], [7], [9]. Some approaches optimize these criteria with force-based algorithms [6], [9]. The annotations are placed at initial positions on the view plane. Attracting and repulsive forces are defined among between them and the border of the view. A relaxation process then minimizes the overall forces over multiple iterations, so that the annotations obtain improved positions. The computational costs do not allow for real-time label placement and, hence it needs to be performed in a post-processing step.

Visual depictions in 3D demand dynamic and different types of annotation techniques which are both conceptually and algorithmically more complex. Preim et al. [15] present a first approach for 3D label placement, where fixed containers are reserved on the view plane to hold textual descriptions linked by lines to the referenced objects. Ritter et al. [16] introduce the concept of illustrative shadows: Annotations are linked to reference points in the shadows of objects projected onto an additional shadow plane and, thereby, support an intuitive visual association. Sonnet et al. [17] investigate annotations in the context of interactive explosion diagrams intended to explore complex 3D objects. Kolbe investigates the annotation of buildings in pre-recorded path videos [11], which augment the geo-referenced frame sequence of the video. To calculate the placement of annotations an additional 3D city model is required.

### 2.2 View Management Techniques

The term *view management* denotes techniques that handle the available space in the 2D view plane. In the context of labeling, view management is used to avoid the overlapping of labels after their projection. All listed approaches assume that the view planes as well as all free or occupied regions on it are represented with axis parallel rectangles. Typical operations are the mark and release of regions, as well as query operations, e.g., if a given rectangle intersects some occupied region.

Bell et al. develop a view management technique for the placement of widgets on the screen and use this later on to annotate buildings with text and symbols in an interactive 3D environment [2], [3]. In this approach the unoccupied regions are efficiently represented by a set of maximum extended rectangles. Thereby all non-occupied regions are easily accessible. In contrast to our approach, it is not possible to calculate directly an alternative position in case a label intersects an occupied region; all free regions must be iterated.

The work of Müller and Schödel [14] describes a technique to label the components of column charts. Their approach uses up to four borders to manage the remaining space, each one growing from a different edge of the view plane. However, they focus on a different, two dimensional scenario with other constraints.

## 3 Annotation Placement Strategy

Our dynamic placement technique for annotations in virtual landscapes takes advantage of two general characteristics of perspective views of terrain-based scenes:

- Typically, users look from a position at the upper hemisphere down to a point on the terrain surface. Therefore, the anchor points of annotations near to the observer are rather located in the lower area while reference points far away tend to be located in the upper area of the view plane.
- Annotations near the observer tend to be of higher interest than annotations far away.

Taking these characteristics into account we can define an adequate placement by arranging annotations close to the observer in the lower part of the view plane and raising their height with increasing distance to the viewer. Close annotations are connected with short pole lines to their anchor points. Because we can assume that these annotations are in the current focus of the user, the placement strategy fulfills one important criterion for a good placement [6]. Placing the annotations the other way around would generate close annotations in the upper screen area whose pole lines would overlay annotations that are far away and located in the lower area.

The annotation placement strategy avoids overlapping annotations by reserving disjunctive screen space in the view-plane for each annotation. Since overlaps cannot be avoided if the number of annotations and the available screen space do not correspond, a non-overlapping annotation near the observer should be preferred to a non-overlapping annotation far away. In our implementation, we sort the annotation anchor points on the terrain ground according to their depth value, and process them in a front-to-back order. Processing the annotations in this order supports our objective because annotations close to the viewer can reserve their space by the view management at an early stage in the placement process.

Fig. 2 illustrates the placement strategy for a sample virtual landscape. The annotations in front are already processed and placed within the view-plane. The projection of the currently processed annotation can collide with already processed annotations, so that the view management determines alternative placements. To avoid mixing of orthogonal and straight-line label layouts ([1], [10]) annotations are elevated until a free view-plane position is found. To find an alternative position the strategy does not look into directions other than the look-up direction.

To achieve high readability, the annotation placement strategy orients the annotations parallel to the view-plane. Nevertheless, they are treated as 3D objects in the virtual landscape having a depth value and being projected accordingly. The view management uses the view-plane up-right extends of the oriented label to detect overlaps with others. The height over the anchor point is the only degree of freedom for the label, which allows the algorithm to find a new and probably collision-free position.

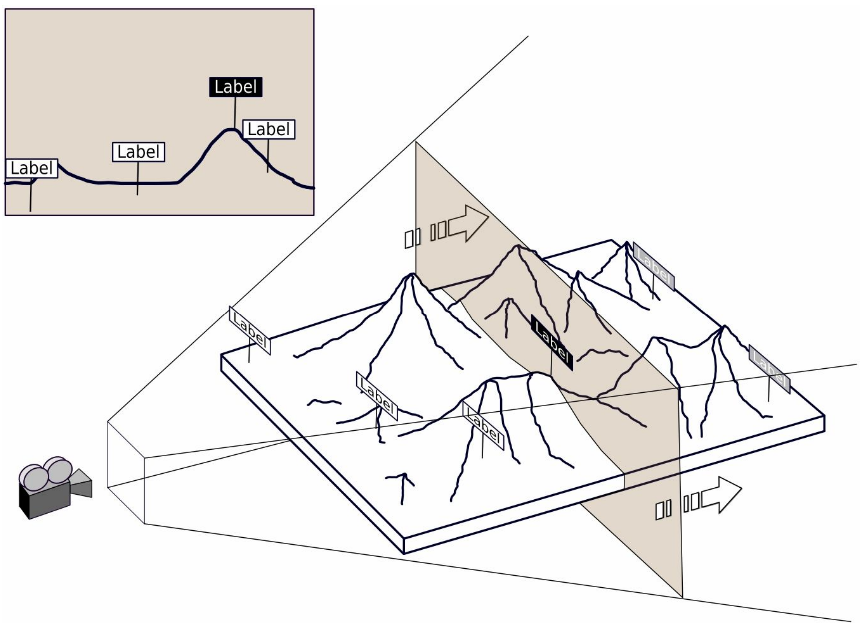



**Fig. 2.** Illustration of the placement strategy for virtual landscapes: Labels are processed in front-to-back order. Possible collisions are than resolved within their two-dimensional plane.

For each frame, our annotation technique performs three main steps.

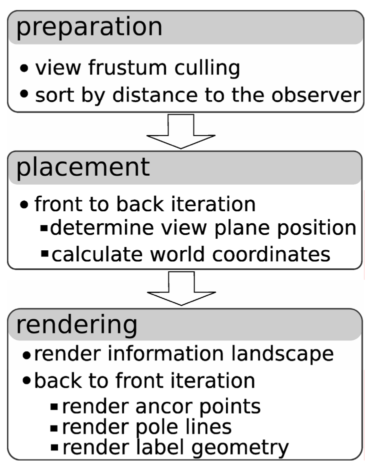



**Fig. 3.** Overview of the rendering process

1. The annotation anchor points are culled against the view frustum to skip all labels that are not visible or too close to the view-plane. The technique uses a conservative culling because anchor points may be outside the view frustum while the annotation is partially visible. Then, the remaining anchor points are sorted by their distance to the observer. This ordering is used in the next two steps.
2. For the actual placement, the algorithm processes the anchor points for the visible annotations in front-to-back order and determines the positions of each annotation on the view plane. For the final view-plane position, the corresponding world coordinates are calculated and stored. Note that the probability to find free positions on the view plane is higher for the labels close to the observer. In addition, close labels are placed below distant labels.
3. The virtual landscape, anchor points, pole lines, and annotations are rendered. The labels are drawn in back-to-front order to guaranty that transparent parts are shown without artifacts.

# 4 View Management

The view management divides the screen space horizontally into discrete slots due to the vertical search direction in our technique. As a result, query and update operations can easily be restricted to the horizontal band that an annotation occupies.

Dividing the vertical dimension in discrete steps as well would lead to a raster structure. To achieve high placement precision these cells would have to be small involving high costs for query and mark operations. Since the spatial extents of annotations on the screen are larger than the targeted precision, we use a linked list for each slot, where each element stores a non-occupied interval, starting from the lower edge of the screen.

For the view management a normalized view-plane coordinate system is used, with coordinates $x_{min}, y_{min} := (0,0)$ at lower left and $x_{max}, y_{max} := (1,1)$ at the upper right corner. Starting with a completely empty view-plane, rectangular regions $(x_0, y_0, x_1, y_1)$, $x_i, y_i \in [0,1]$ can be added to mark occupied areas.

To detect collisions efficiently and to perform the label placement in real-time, the view management needs efficient operations for the following tasks:

- Mark new regions as occupied
- Test if a given rectangle overlaps occupied regions
- Find the closest empty region for a given rectangle in the case of overlaps

Classical space partitioning techniques, e.g., regular grids or quad-tree structures, can be easily adapted to fulfill the first and second task. But they are less suitable for searching empty regions or to directly determine closest unoccupied region. For this reason we develop two variants of view management that consider the special properties of our setting; the *growing-border* and *interval-slot view* management.

## 4.1 Growing-Border View Management

For the growing-border view management each slot contains only one *y*-interval during the whole placement process. It is initialized with $(y_{min} = 0, y_{max} = 1)$ to mark the whole slot as empty. If an area $(x_0, y_0, x_1, y_1)$ should be marked as occupied, the algorithm iterates over all slots that are overlaid by the interval $[x_0, x_1]$ and replaces the value $y_{min}$ in every slot with $\max(y_1, y_{min})$.

To query for a collision, all these slots are iterated in the same way. If one of them contains a $y_{min}$-value greater than the query rectangle's $y_0$, the iteration can stop because a collision is found.

Furthermore, the query algorithm can directly determine at which height the rectangle must be placed to avoid the collision. For this, the iteration continues until all slots are taken into account. The maximum $y_{min}$ value found in the slots during this iteration is the new height for the placement. To avoid placement outside of the view-plane a horizon defines a maximum *y* value. With this, the maximum *y* value that a label can reach is $y - labelheight$. Fig. 4 illustrates the placement of four labels A-D with the growing-border technique.

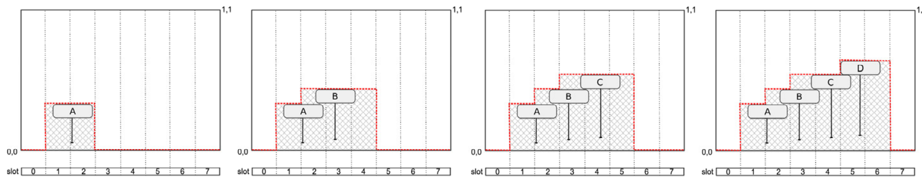



**Fig. 4.** Placing the labels A-D by the growing-border view management. The occupied view-plane region is hatched.

### 4.2 Interval-Slots View Management

With the growing-border technique annotations are simply stacked, so that typically large parts of screen space are unnecessarily marked as occupied. For this reason we extend the view management in such a way that we store for every slot an ascending sorted list of empty intervals $(y_{min}, y_{max}), y_{min} < y_{max}$ . Initially, each slot contains one interval $(y_{min} = 0, y_{max} = 1)$ , to mark the whole slot as empty.

To test a new rectangle for a collision with, the algorithm needs to iterate over the involved slots as before. Additionally, for each slot list we search for empty intervals within the region $[y_0, y_1]$ . If there is one slot that shows a collision with the rectangle, the query can stop with a negative result.

With a slight modification the algorithm can be extended to return the next empty position for a query rectangle. As long as there is no collision during the iteration, we keep track of the empty region above the current label position. If one slot raises a collision, we check if the label fits into the current empty region after we have lifted it above the obstacle. In this case the iteration can continue. Otherwise, it is restarted after the rectangle has been lifted above the whole empty region.

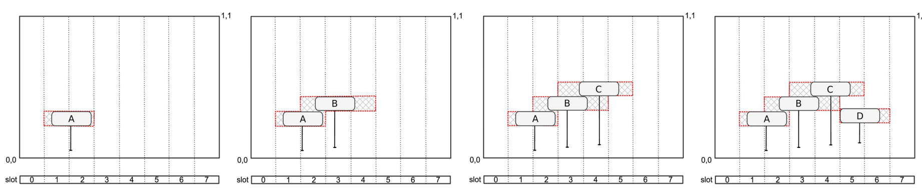



**Fig. 5.** Placing the labels A-D by the interval-slot view management

## 5 From Planar Maps to 3D Landscapes

The algorithm as described so far works perfectly for (near) planar maps shown in perspective views. For 3D landscapes, which show significant non-planar morphology, the 3D geometry has to be considered to avoid the intersection of labels. To accomplish this, we extend the test of our placement strategy.

For the calculated view-plane position of an annotation we additionally verify for the screen area the label overlays that no part of the 3D landscape is closer to the observer than the label. If this test fails the label is raised until this conflict is resolved as well.

For the depth value test we render the 3D landscape in a separate depth buffer with a low resolution before the placement starts. Because of the planar shape of the label and its orientation to the observer, the depth value of the label is constant and needs to be calculated only for one point. To speed-up this test a hierarchical z-pyramid can be used [8]. For 3D landscapes with predominant convex features the evaluation of the depth values can be restricted to sample positions. We found that a sampling at the label corners and at the midpoints of the horizontal label borders suffice in the most cases and introduce artifacts only in rare situations.

As an alternative, the depth criteria can be validated with explicit ray tests as well. This method is useful if the scene has a low depth complexity, contains only a few visible labels, or the precision of the discrete depth buffer does not suffice.

## 6 Extensions

**Visibility of Scene Elements.** The view management can be used to manage the visibility of other scene elements as well. For example, in some scenarios users report the need of a clear indication where anchor points for the label are exactly located. Here, additional spheres or boxes are used to mark these points. To guaranty the visibility of the nearest, their bounding boxes are projected onto the view plane and inserted into the view management before the label placement starts. Another example is the initial masking of a screen space application logo to avoid label placements in this region.

**Priorities.** In some applications, specialized labels exist that must be preferred in terms of their visibility. For this the placement process can be extended to support labels with different priorities. It iterates first over all priorities and in an inner loop over the labels with the current priority from a front-to-back order. For the rendering step this attribute can be ignored so the labels are drawn back-to-front without the consideration of priorities.

**Dynamic Aspects.** During navigation through the 3D landscape, the label positions can change at a high frequency and possibly spread over the whole view-plane space. This causes unsteady jumps in the visualization, irritating the user and makes it nearly impossible to trace labels that where currently in focus. To ease this situation we add a hysteresis characteristic to the placement. This way the labels keep their current positions during the 3D interaction and perform a continuous movement to their final destination once the user pauses or finishes the interaction.

## 7 Results

We have developed a prototypic implementation of our dynamic annotation placement approach. Fig. 6 shows the different placement strategies on a flat map. In the example, 100 labels with the same text are placed randomly with a constant height

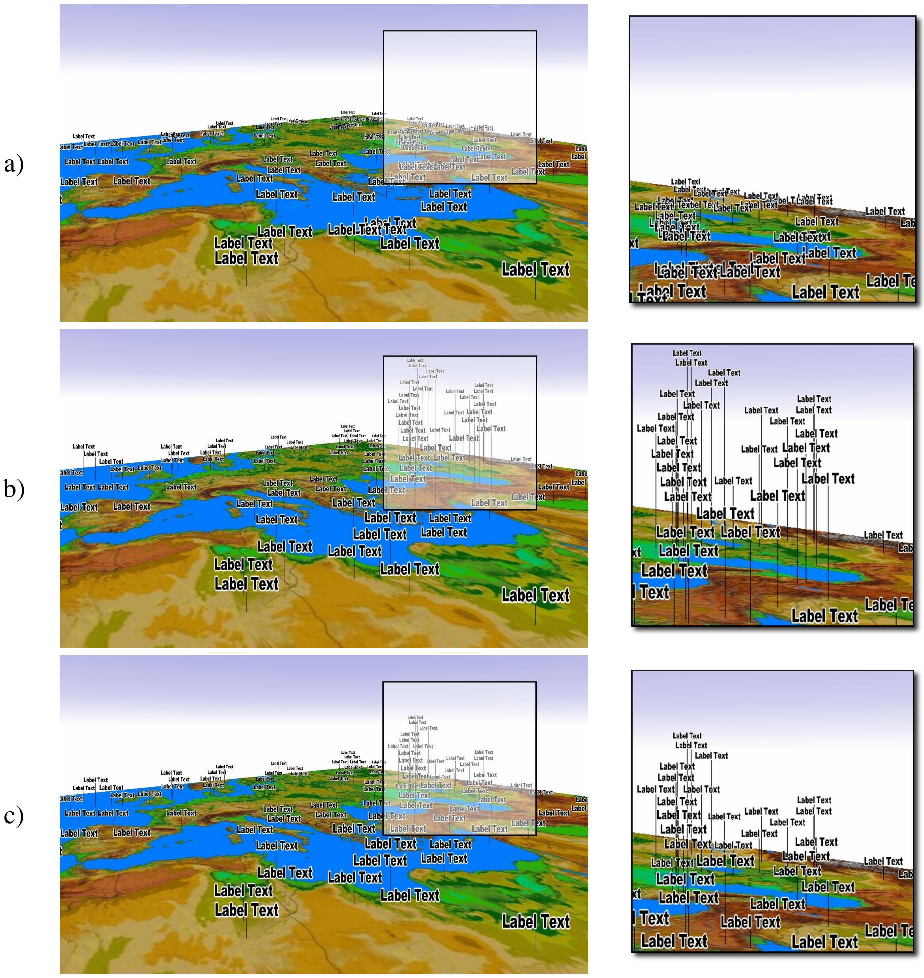



**Fig. 6.** Label placement with a) constant height, b) growing border, and c) interval slot view management

(Fig. 6a), with the growing border (Fig. 6b), and with the interval-slots view management (Fig. 6c). In contrast to the constant height placement, the view management strategies try to avoid overlapping labels and the occlusion of labels from nearer scene objects (Fig. 1, Fig. 7). These visibility criteria, most likely achieved for labels near the observer, result in an improvement of legibility.

We expected that the interval-slot view management results in a too compact label arrangement but due to the perspective scaling closely placed labels can be distinguished from each other. It seems that the greater pole lengths generated by the growing-border approach are more visual disturbing than the compact label layout of the interval-slot approach. This needs to be evaluated in a user study.

Currently all labels are placed centered over their pole line. This provides the user with a clear mapping between the text, the line and the anchor point. Introducing an additional degree of freedom at this point could be used to avoid a clustering of labels in a too compact layout and to support explicit aesthetic parameters in the placement process.

The number of interval slots controls the precision of the placement. If the view plane is divided in only a few slots, the region marked as occupied for each label expand along the horizontal axis of the screen, so that the whole labeling extents into the height. An upper bound for the number of interval slots is the number of pixels used in this dimension. Then neighboring labels can be placed with maximum accuracy using all available screen space with the downside of higher costs for update and query operations. Adjusting the number of interval slots allows the user to balance between placement precision and processing time.

Fig. 7 shows label placements with the interval-slot view management at a 3D landscape.

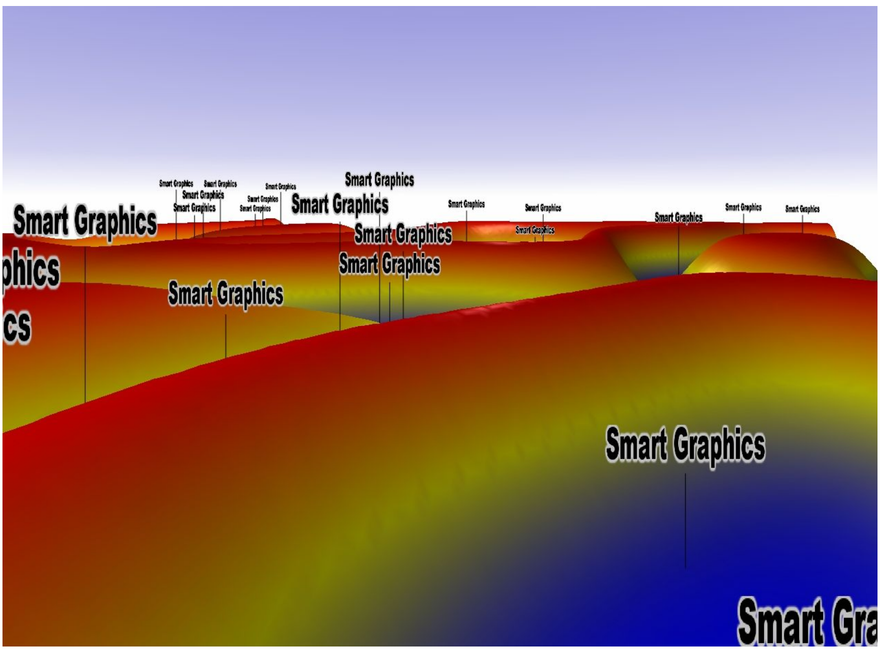



**Fig. 7.** Label placement on a 3D landscape represented by a height field

## 8 Conclusions and Future Work

This paper presents a point-feature dynamic annotation placement strategy for virtual landscapes. Our concept uses the properties of virtual landscape views in combination with an efficient view management. Annotations are not only two-dimensional screen objects but represent three-dimensional scene objects so that they appear as integrated elements of the 3D scene and provide inherent depth cues.

The implementation operates in real-time and, therefore, can be applied in interactive applications. It can also be easily extended to force the visibility of other scene elements, supports labels with different priority, and provides smooth transitions between changing annotation positions.

Future work will focus on interactive exploration techniques. We also plan to investigate how to add interactive behavior to annotations. For a top view on the 3D landscape established cartographic label placement techniques could be used to find placements that are more common for traditional map views.

## Acknowledgements

We would like to thank the Neanderthal Museum in Mettmann, Germany, especially Jan Kegler, for providing us with the cartographic material.

## References

1. Ali, K., Hartmann, K., and Strothotte, T. Label Layout for Interactive 3D Illustrations. Journal of the WSCG, 13 (1):(2005), 1–8.
2. Bell, B. and Feiner, S. Dynamic Space Management for User Interfaces. In Proceedings of the 13th ACM Symposium on User Interface Software and Technology (UIST), volume 2 of CHI Letters. San Diego, CA, 2000. pp. 239–248.
3. Bell, B., Feiner, S., and Höllerer, T. View Management for Virtual and Augmented Reality. In Proceedings of the 14th ACM Symposium on User Interface Software and Technology (UIST). ACM Press, 2001. pp. 101–110.
4. Chigona, W., Schlechtweg, S., Thomas, S. Dual Use of Image Space: The Challenges of Explaining Visualizations from Within. In Schulze, T., Schlechtweg, S., Hinz, V. (eds). Simulation und Visualisierung 2001, SCS-Society for Computer Simulation Int., Delft, Belgium, 175-185, 2001.
5. Christensen, J., Marks, J., and Shieber, S. An Empirical Study of Algorithms for Point-Feature Label Placement. ACM Transactions on Graphics, 14 (3):(1995), 203–232.
6. Ebner, D., Klau, W.K., Weiskircher, R. Force-Based Label Number Maximization, Technical Report TR-186-1-03-02, Technical University Vienna, June 2003, available at http://www.apm.tuwien.ac.at/publications/bib/ pdf/ebner-03.pdf.
7. Edmondson, S., Christensen, J., Marks, J., Shieber, S. M. A general cartographic labeling algorithm. TR1996-004, Cartographica 33, 4, 13-23, 1996, available at http://www. merl. com/pub lications/TR1996-004/.
8. Greene, N., Kass, M., and Miller, G. Hierarchical Z-Buffer Visibility. Computer Graphics, 27 (Annual Conference Series):(1993), 231–238.
9. Hartmann, K., Ali, K., and Strothotte, T. Floating Labels: Applying Dynamic Potential Fields for Label Layout. In Smart Graphics: 4th International Symposium (SG 2004), volume 3031. Springer-Verlag, 2004. pp. 101–113.
10. Hartmann, K., Götzelman, T., Ali, K., and Strothotte, T. Metrics for Functional and Aesthetic Label Layouts. In A. Butz, B. Fisher, A. Krüger, and P. Olivier, editors, Smart Graphics: 5th International Symposium, SG, volume 3638. Springer, Frauenwoerth Cloister, Germany, 2005. pp. 115–126.

11. Kolbe, T. H. Augmented Videos and Panoramas for Pedestrian Navigation. In G. Gartner, editor, Proceedings of the 2nd Symposium on Location Based Services and TeleCartography, 2004.
12. Maass, S. and Döllner, J. Dynamic Annotation of Interactive Environments using Object-Integrated Billboards. In J. Jorge and V. Skala, editors, 14-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG'2006. Plzen, Czech Republic, 2006. pp. 327–334.
13. Marks, J. and Shieber, S. The Computational Complexity of Cartographic Label Placement. Technical Report TR-05-91, Harvard University, 1991.
14. Müller, S. and Schödl, A. A Smart Algorithm for Column Chart Labeling. In Smart Graphics: 5th International Symposium (SG 2005), 2005. pp. 127–137.
15. Preim, B., Raab, A., Strothotte, T. Coherent Zooming of Illustrations with 3D-Graphics and Textual Labels. Proceedings. of Graphics Interface, 105-113, 1997.
16. Ritter, F., Sonnet, H., Hartmann, K., and Strothotte, T. Illustrative Shadows: Integrating 3D and 2D Information Displays. In IUI'03: Proceedings of the 8th International Conference on Intelligent User Interfaces. ACM Press, 2003. pp. 166–173.
17. Sonnet, H., Carpendale, S., and Strothotte, T. Integrating Expanding Annotations with a 3D Explosion Probe. In AVI '04: Proceedings of the Working Conference on Advanced Visual Interfaces. ACM Press, 2004. pp. 63–70.

# Predictive Text Fitting

Xiaofan Lin

Hewlett-Packard Laboratories
1501 Page Mill Road MS 1203
Palo Alto, CA 94304, USA
xiaofan.lin@hp.com

**Abstract.** A new predictive text fitting algorithm is introduced in this paper. Through this algorithm, the ideal font size of given text content can be efficiently determined to fit the text into frames of arbitrary shapes. After trying out the initial font size, the algorithm measures a fit factor based on ratio of the area used and the area needed. Then it predicts the next font size according to the fit factor, and it also adjusts the key model parameter based on previous iterations. Compared with methods that change the font size at fixed amount each time or use predetermined models, the advantages of this algorithm include fast convergence, as well as insensitivity to text placement engine and initial values of parameters. This algorithm has a number of potential applications, such as variable data printing, multimodal publishing, and adaptive graphic presentation.

## 1 Introduction

Text fitting refers to the process of placing given text content into a frame of the specified shape and dimension while satisfying two conditions: 1) All text content falls inside the frame without overflow; and 2) Most area of the frame is covered by the text without significant underflow.

There are different methods of text fitting, either manually performed or automated by computer algorithms. First, if the text initially already roughly fits into the frame with only minor overflow or underflow, fine-tuning the word and letter spacing can achieve the objective. Professional graphic artists are routinely practicing this technique in their designs. Knuth [1] also designed an algorithm in the TeX system to automate such process. However, this method will not work if initial rough fit is not available. Second, the text content can be expanded or shortened to fit into the frame. This is another common technique employed by graphic artists, who may ask the text writer to modify the text to help with the overall graphic design. Although not exactly for text fitting purpose, Jacobs et al. [2] introduced an adaptive document layout system that can automatically pick from different versions of text contents to construct a good layout. Considering the limitations of state-of-the-art natural language processing technologies, the different versions usually have to be manually prepared, if possible at all. Third, typesetting parameters such as font size can be systematically

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 13 – 23, 2006.

optimized to accomplish good text fit. Compared with the first two methods, this method has the widest applicability because it does not require a good initial state or manual modification of the content. Such parameter optimization is also one area where computers can excel humans because of the computation or iterations involved. Thus, this paper focuses on the third method. This problem does not have one-step close-form solution because many complex nonlinear factors will affect the text fitting process: line breaking, variable font widths (the width of individual characters can be different), non-rectangular frames, paragraph breaking, and so on. A naïve algorithm may try out a number of font sizes at fixed interval. If the interval is too large, we may not find a good solution. If the interval is very small, too much iteration is needed and speed will be too slow to be accepted. Cedar and Rhoten [3] proposed a more efficient algorithm, which uses a "fullness ratio" of the current iteration to guide the font size selection for the next iteration. However, it only works for rectangular frames and assumes a fixed quadratic model in modifying the font size.

This paper introduces a new predictive text fitting algorithm. Through this algorithm, the ideal font size of given text content can be efficiently determined to fit the text into frames of arbitrary shapes. After trying out the initial font size, the algorithm measures a fit factor based on ratio of the area used and the area needed. Then it predicts the next font size according to the fit factor. Instead of the fixed quadratic model, a key parameter in text area estimation model is also predicted after each time. Compared with methods that adjust the font size at fixed amount each time or use predetermined models, the advantages of this algorithm include fast convergence, as well as insensitivity to text placement engine and initial values of parameters. Section 2 explains the algorithm in detail. Section 3 presents the different applications. Conclusions are drawn in Section 4.

## 2 Predictive Text Fitting Algorithm

In this section, we first formally define the targeted text fitting problem (Section 2.1). Then we introduce the key steps of the algorithm (Section 2.2). After that we analyze the algorithm performance (Section 2.3).

### 2.1 Problem Definition

On the functionality level, text fitting can be regarded as black box shown in Fig. 1. It takes a number of inputs: shape description, text content, fixed text formatting parameters, and text placement engine. The output is the remaining text formatting parameters so that we can uniquely render the text. For completeness, we are going to describe the details of each input. However, it is worth mentioning that the core text fitting algorithm is independent of the exact format or implementation of the input. For example, if the format of the shape description changes or another text placement engine is adopted, the same fitting algorithm still applies.

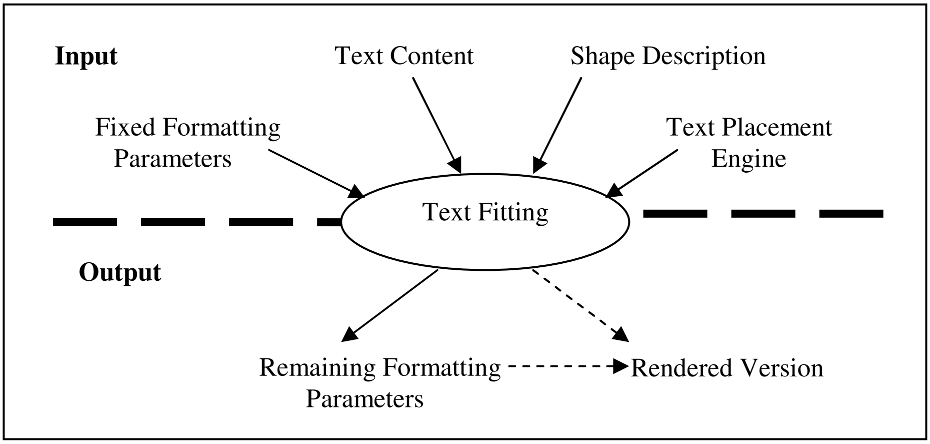
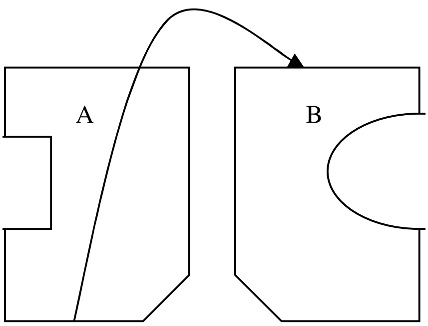



**Fig. 1.** Black box view of text fitting

**Input:**

1). <u>Shape description **S**</u>**.** Our shape description schema allows polygons, eclipses, and concatenation of multiple blocks (sometimes the text will be placed into several disconnected blocks). Fig. 2 illustrates the features and schema of the shape description.




```
<shape>
<!- - Describe a polygon - ->
<point px="748" py="138"/>
<point px="900" py="138"/>
…
<!- - Switch to the next block - ->
<next/>
<point px="1106" py="32"/>
<!- - Describe a eclipse segment - ->
<eclipse a="127" b="127" stx="0"
sty="0" bitewidth="254" direction="-
1"/>
…
</shape>
```

(a) (b)

**Fig. 2.** Illustration of the shape description: (a). Text is first placed into Block A and then Block B. (b). Example shape description XML files.

2). <u>Text content **T**</u>**. T** can change in a wide range, from a single word to a number of paragraphs.

3). <u>Fixed text formatting parameters **F1**</u>**. F1** consists of the parameters that keep constant values throughout the text fitting process. The following is an example:

**Table 1.** Example of fixed text formatting parameters

| Font family | Font weight | Font style | Alignment |
|---|---|---|---|
| Times-Roman | Bold | Italic | Justified |

4). Text placement engine **A.** Given all of the formatting parameters and the text content, the text placement engine distributes the text into a text frame by breaking the text into lines and deciding the exact position of each character. Apache's XSL-FO [4] Processor (FOP) [5] serves as a good baseline text placement engine that can place text into a rectangular frame using a greedy line-by-line line breaking algorithm. As described in our earlier paper [6], we have extended XSL-FO and FOP to build an advanced text handling adaptor in order to support placing text into an arbitrary shape as well as Knuth's paragraph-based line breaking algorithm [1].

**Output:**
The output of the text fitting algorithm is the remaining formatting parameters **F2** of **T** (for example, font size) so that **T** can fit into **S** and satisfy two requirements:

- **T** must fall into **S** in its entirety;
- **T** should occupy as much as possible space of **S**.

### 2.2 Specifics of the Algorithm

Fig. 3 summarizes the algorithm's major steps, which are described as follows:

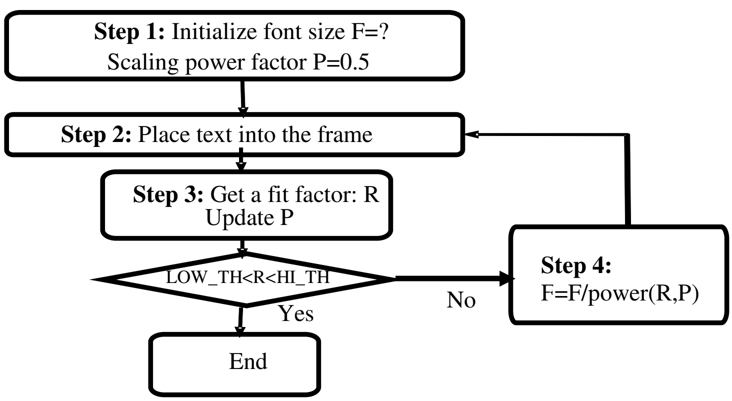
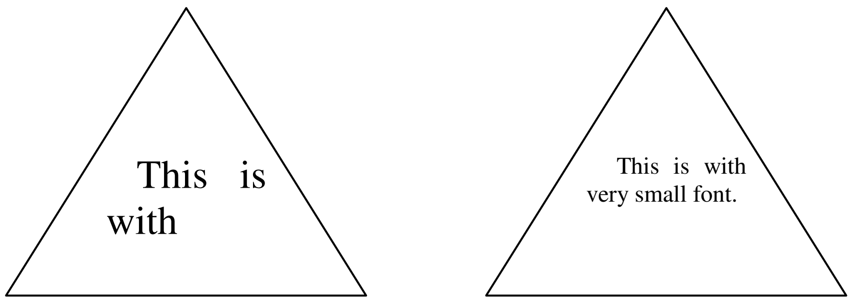



**Fig. 3.** Workflow of the predictive text fitting algorithm

**Step 1:** Assign the initial formatting parameters **F2**, and adaptation factor **P.**
The initial formatting parameters are specified. For example, we can set the font size **F** to the one defined in the template document. The adaptation factor **P** will be used and explained in later steps. Here it can be set to a reasonable value, such as 0.5.

**Step 2:** Place text **T** into **S** according to **F1** plus current **F2.**
A text placement engine, such as that introduced in Section 2.1, is used to place the text into the frame. The first trial usually will not be satisfactory. The text may either overflow (see Fig. 4 (a)) or underflow (see Fig. 4 (b)).




(a) Text overflow (b) Text underflow

**Fig. 4.** Mismatch between the text and the text frame

**Step 3:** Measure the fit factor by a filled area ratio **R**.
This step will measure how much space is currently covered by the text. One measurement used in existing methods [3] is the height of covered text. However, height is not an accurate metric if the text frame is not rectangular. For example, in the case of Fig., equal vertical space on the top and bottom will reflect substantially different usable areas. In order to overcome the drawbacks associated with height, we use area to measure the fit factor **R**:

$$R = \frac{\text{Occupied Text Area}}{\text{Total Text Frame Area}}. \quad (1)$$

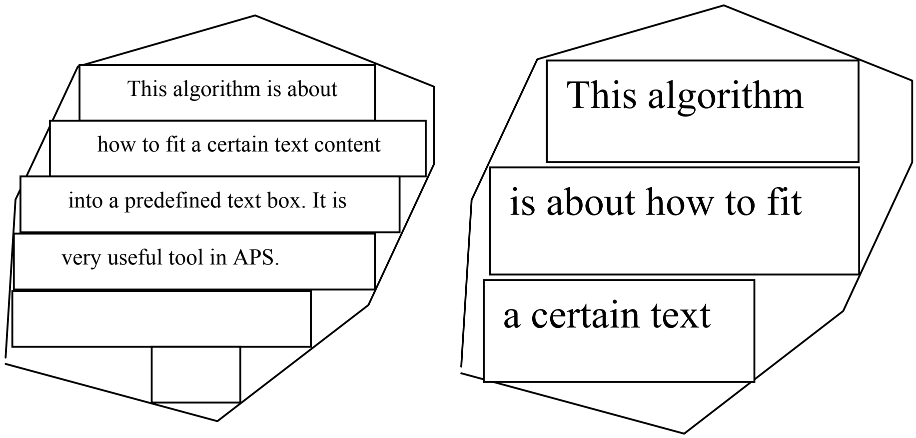
R will be close to 1 when the text perfectly fits into the text box. A value less than 1 signals underflow, and a value larger than 1 means overflow. In order to support non-rectangular text frames, the following techniques are used to calculate **R**:

- When there is underflow (**R**<1):

The text placement engine will signal that there is more than enough space to place the text. The text placement engine will also give us the text line boxes. Some lines are occupied by the text and some are not. Then **R** is calculated as:

$$R = \frac{\sum_{i=1}^{NT} \text{LineArea[i]}}{\sum_{i=1}^{N} \text{LineArea[i]}}. \quad (2)$$

where NT is the number of lines occupied by text and N is the total number of line boxes. LineArea[i] is the area of the *i*th line box, as decided by the product of the line height and the line width.




(a) When R<1 (b) When R>1

**Fig. 5.** How to calculate **R**?

In the example shown in Fig. 5(a), if the six line boxes have areas of 8, 10, 10, 9, 7, and 2 respectively, then **R** is (8+10+10+9)/(8+10+10+9+7+2)=0.805.

- When there is overflow (R>1):

The text placement engine will signal that there is not enough apace to put the text. It will also give us the area needed for each character. Some characters fall into the text frame and some are not. Then R is calculated as:

$$R = \frac{\sum_{i=1}^{C} \text{CharArea[i]}}{\sum_{i=1}^{CI} \text{CharArea[i]}}. \tag{3}$$

where CI is the number of characters falling into the text frame and C is the total number of characters in the text. CharArea[i] is the area of the *i*th character box, as decided by the product of the character height and the character width (such information is available as part of font metrics from the text placement engine). If the width of every character is the same or exact character widths are not available, we can also use C/CI to approximate **R**. For example, the **R** in Fig. 5 (b) is calculated to be 2.25.

Obviously, when the fit is almost perfect, **R** will be close to 1 using either method. Otherwise the iterations may oscillate around the optimal values. After R is calculated, it is compared with predefined range [LOW_TH, HI_TH]. For example, LOW_TH=0.9, HI_TH=1.0. If **R** falls into that range, the process will be terminated.

**Step 4:** Scale parameters in **F2** based on **R.**

If **R** is still not good enough, parameters in **F2** need to be adjusted. Although it is easy to know the direction of the adjustment, it is challenging to decide how much to adjust. We choose the following prediction scheme:

Calculate scale factor $S(n) = 1/R(n)^{P(n)}$ (4)

When n=1, P(n)=0.5

Otherwise, P(n)= ln(S(n-1))/ ln(R(n)/R(n-1)) (5)

Calculate next F2 as F2(n+1)=F2(n)*S(n) (6)

(n is the number of iterations and ln() is the natural log function)

The heuristics behind adopting this function are: If there is overflow (R>1), the font size should be reduced; if there is underflow (R<1), the font size should be increased. The positive number **P** is the model parameter controlling the strength of adjustment. In theory, if every parameter (the gap between the lines, the gap between the characters, etc.) is proportional to the font size, the total occupied area is approximately a quadratic function of the font size. So a good estimate of **P** is 0.5 under this assumption. However, if the text placement engine does not change everything in proportion to font size (for instance, keep the same line height when changing font size), this model will not be valid at all. Besides, other factors such as line breaking and paragraph breaking will also make the model less accurate. Thus, an adaptive selection strategy for **P** has been proposed instead of fixing the value of **P**. In the initial iteration, **P** is assigned to some value, such as 0.5. After running the actual text placement, we update P based on the previous two R values. The rationale of Equation 5 is as follows:

Let us assume:

$R(n+1)=R(n)* S(n)^x$ (x>0)

We can estimate x from preceding iterations:

$R(n)=R(n-1)*S(n-1)^x$

So we have:

x=ln(R(n)/R(n-1))/ln(S(n-1))

Ideally, R(n+1) should be 1. So we have:

$1 = R(n)*S(n)^x$

Thus,

$$S(n) = 1/R(n)^{1/x} = 1/R(n)^{\ln(S(n-1))/\ln(R(n)/R(n-1))}$$

And thus:

P(n)= ln(S(n-1))/ ln(R(n)/R(n-1))

After Step 4, we move back to Step 2. The process will iterate until a good fit is achieved. Table 2 gives a concrete example of the process that includes four iterations.

**Table 2.** Example of iterations

| Iteration | F2(n) (font size) | R(n) | P(n) | S(n) |
|---|---|---|---|---|
| 1 | 12 | 4 | 0.5 | 0.5 |
| 2 | 6 | 0.64 | 0.378 | 1.18 |
| 3 | 7.1 | 0.8 | 0.74 | 1.18 |
| 4 | 8.37 | 0.95 (done) | | |

### 2.3 Performance Analysis

First, we want to understand how sensitive this algorithm is to the initial parameters (font size). So for the same text shape and content (displayed in Fig. 7), we initialize the font size to different values and measure the number of iterations needed to reach good fit. Fig. 6 shows the result. When the initial font size changes in a large range (from 3 to 30 points), it only takes maximal three iterations to converge to the final font size, which is around 15 points.

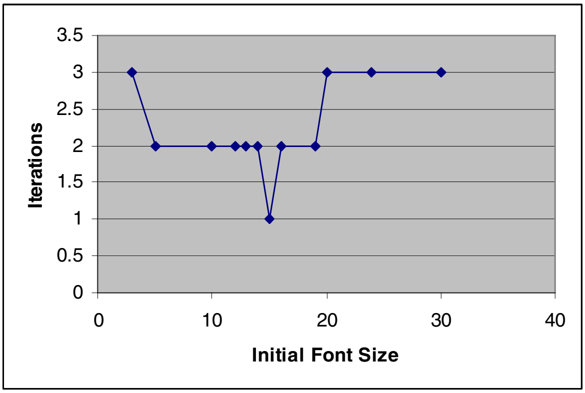


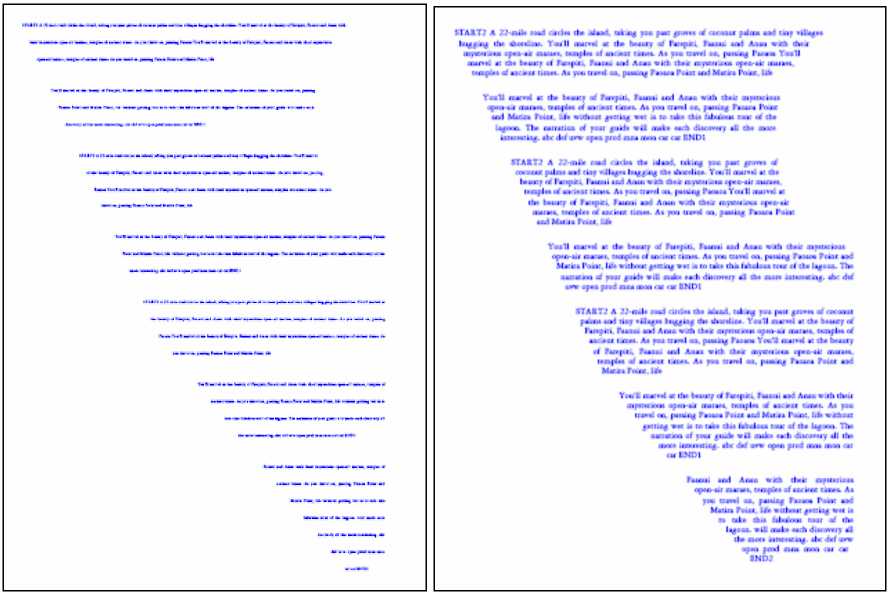

**Fig. 6.** Number of iterations as a function of the initial font size

Next, we would like to know how this algorithm will perform with different text placement engines. One text placement engine keeps fixed line height, and Fig. 7 (a) is the result. The other text placement engine scales proportionally every parameter (the gap between the lines, the gap between the characters, etc.) with the font size, and Fig. 7 (b) is the result. The first row in Table 3 corresponds to the numbers of iterations needed with the predictive text fitting algorithm. Even if a bad initial P (0.5) value is selected for the first text placement engine, only one extra cycle is executed. In contract, if we do not adjust P [3], six iterations will be needed for the first text placement engine (the second row).

The above two experiments demonstrate that the proposed algorithm is not sensitive to the initial values of parameters and the text placement engines. This unique and advantageous characteristic is the result of adaptively predicting the parameters and the key model parameters.

**Table 3.** Comparison of two scaling strategies

| No | Strategy of changing P | Iteration needed (Line height constant. Fitting result is in Fig. 7 (a)) | Iteration needed (Everything proportional to font size. Fitting result is in Fig. 7 (b)) |
|---|---|---|---|
| 1 | Initial is 0.5 and then adaptive | 3 | 2 |
| 2 | 0.5 | 6 | 2 |



(a) Line height is constant

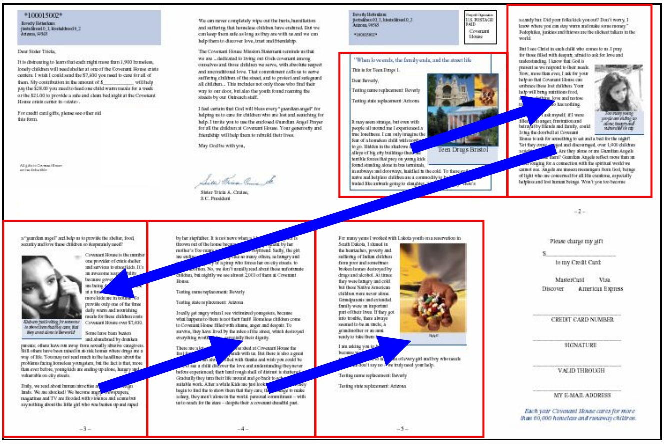
(b) Everything proportional to font size

**Fig. 7.** Text fitting results

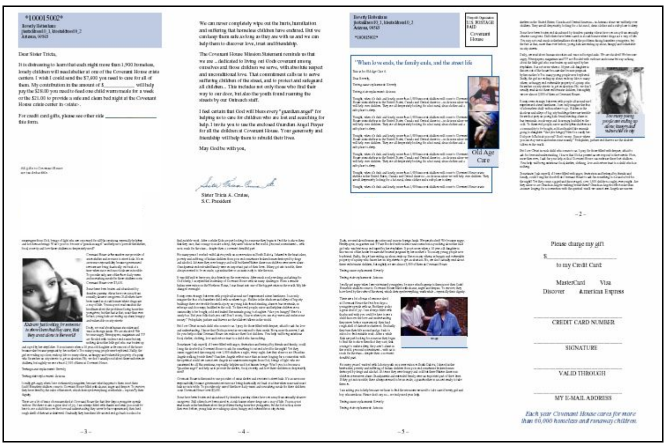
## 3 Applications

Text fitting has a number of applications. In Variable Data Printing (VDP), customized contents are dynamically composed into professional quality documents to be printed and distributed. In HP Labs, we have built an automated marketing campaign prototype system [7], which selects the targeted customers based on the campaign objective, personalize the contents for each selected customer, and then composes the campaign brochure using the proposed predictive text fitting algorithm. In addition, we are using Active Layout Template [8], which extends XSL-FO by introducing variables to describe adjustable template. As shown in Fig. 8, the central piece of this brochure is a long pledge letter spanning five pages. The copy holes are non-rectangular on four of the five pages. With the predictive text fitting algorithm, we can fit a personalize letter into the five pages by automatically adjusting the font size.

As part of our future research plan, we are considering other applications. For example, text fitting is also a powerful tool is multimodal publishing, where the same content is to be displayed on devices of different dimensions and aspect ratios, such as cell phones, notebook computers, and large screen projectors. Even on the same device, text fitting can enhance the interactive graphic presentation by always fitting the text content nicely into the view window set by the user. In addtion, the speed advantage of the proposed algorithm can be fully exploited in this application where realtime interaction is essential.



(a) Brochure for one customer (font size=10.3 points)



(b) Brochure for another customer with much more text content (font size=6.7 points)

**Fig. 8.** Personalized marketing campaign brochures generated by predictive text fitting

# 4 Conclusions

In summary, this paper makes the following contributions:

- It formally defines the problem of text fitting and systematically reviews various existing techniques and practices.
- It introduces a new predictive text fitting algorithm that works for text frames of arbitrary shapes. The font size is modified based on the fit factor of the recent iteration. To accelerate the process, the key model parameter is also adjusted based the fit factors of two previous iterations.

- It also studies the convergence speed under different initial states and text placement engines. The experimental results confirm that this algorithm is not sensitive to initial parameters and text placement engines.
- It describes a range of applications of the algorithm. In particular, the VDP usage in an automated campaign system is demonstrated with real-world results.

## Acknowledgments

The author is grateful to Menaka Indrani for integrating the predictive text fitting algorithm into the VDP system, and to Hui Chao for providing the testing layout template files. Anna Durante, Gary Vondran, and Henry Sang have all supported this research from the start.

## References

1. Knuth, D. E.: Breaking Paragraphs into Lines. Software Practice and Experience, Vol. 11 (1981) 1119-1184
2. Jacobs, C., Li, W., Schrier, E., Bargeron, D., Salesin, D.: Adaptive Grid-based Document Layout. ACM Transaction on Graphics, Vol. 22 No. 3 (2003) 838-847
3. Cedar, C. A., Rhoten, M. P.: Method and System for Automatically Causing Editable Text to Substantially Occupy a Text Frame. US Patent 6,256,650 (2001)
4. World Wide Web Consortium: XSL Specification. http://www.w3.org/TR/xsl
5. Apache Software Foundation: FOP Project. http://xml.apache.org/fop/index.html
6. Lin, X.: Intelligent Content Fitting for Digital Publishing. Proceedings of SPIE Digital Publishing Conference, San Jose (2006) CID: 60760J
7. Vondran, G. L., Chao, H., Lin, X., Joshi, P., Beyer, D., Atkins, C. B., Obrador, P.: Automated Campaign System, Proceedings of SPIE Digital Publishing Conference, San Jose (2006) CID: 607605
8. Lin, X.: Active Layout Engine: Algorithms and Applications in Variable Data Printing. Computer-Aided Design (2006)

# Agent-Based Annotation of Interactive 3D Visualizations

Timo Götzelmann, Knut Hartmann, and Thomas Strothotte

Department of Simulation and Graphics
Otto-von-Guericke University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
{timo, knut, tstr}@isg.cs.uni-magdeburg.de

**Abstract.** This paper presents a novel real-time algorithm to integrate internal and external labels of arbitrary size into 3D visualizations. Moreover, comprehensive dynamic content can be displayed in annotation boxes. Our system employs multiple metrics in order to achieve an effective and aesthetic label layout with adjustable weights. The layout algorithm employs several heuristics to reduce the search space of a complex layout task. Initial layouts are refined by label agents, i.e., local strategies to optimize the layout and to minimize the flow of layout elements in subsequent frames.

## 1 Introduction

The efficiency of learning or instructive material relies on a smooth integration and coordination of visual and textual elements. Thus, visualizations in scientific textbooks and technical documentations are usually enhanced with textual annotations, legends, and figure captions.

Interactive 3D visualizations can complement or even substitute their conventional (printed) counterparts as they ease the understanding of complex spatial configurations. Here, users can adjust the viewing direction while those visual objects which are relevant for the current interaction context are emphasized automatically. Moreover, the integration of textual annotations presented in *labels* can support a wide range of communicative functions (e.g., to establish the denotation of visual elements).

Our automatic label layout approach for interactive 3D visualizations incorporates 3 label classes (see Fig. 1): *External* labels require additional meta-graphical objects like *connecting lines* and *anchor points*, while *internal* labels overlay their reference objects. Finally, *annotation boxes* may contain more comprehensive explanations, descriptions, or additional visualizations. In contrast to external labels, the relation to their associated visual objects is not made explicit through connecting lines but through proximity. The integration of several label classes into a 3D browser both supports the interactive exploration of complex information spaces and eases the layout task.

**Interactive Exploration:** We aim to support Shneiderman's visual information seeking mantra: *overview first, zoom and filter, then add details on demand* [21]. Internal and external labels provide an overview of relevant visual objects, whereas annotation boxes convey comprehensive descriptions for selected objects. As these descriptions

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 24–35, 2006.

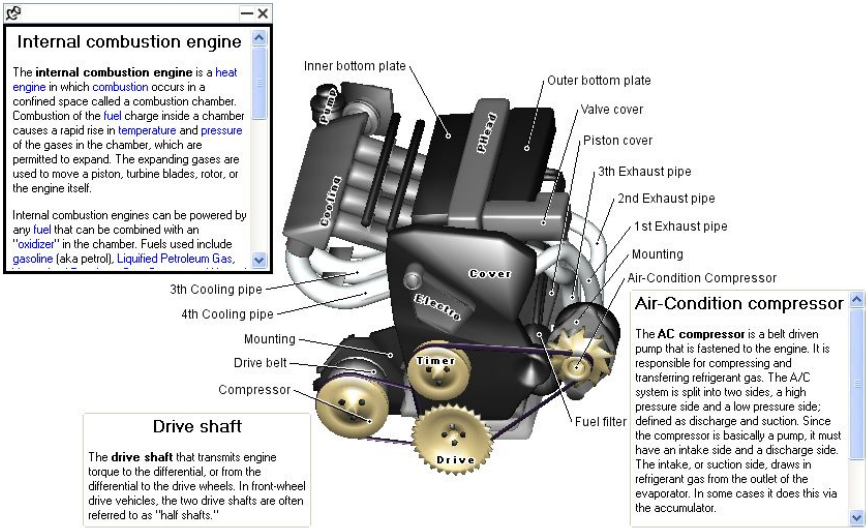



**Fig. 1.** Label classes: annotation boxes, external and internal labels

mark crucial aspects of the learning task, the layout system aims at displaying annotation boxes as large as possible while preventing overlaps with scene objects. Moreover, the distance to their associated visual elements should be minimized according to Mayer's spatial contiguity principle [15]. Users can pin annotation boxes on their current position, enlarge or minimize them. If an annotation box which is associated to a visual object is minimized, it again becomes a label — if it is associated to the over all illustration, it becomes a figure caption.

**Layout Task:** Internal and external labels have to obey different functional and aesthetic requirements (see [10]). Their almost complementary properties are exploited by our layout algorithm: Internal labels directly convey the co-referential relation between annotations and their visual reference objects. Furthermore, no additional meta-graphical objects overlay the visualization. Therefore, our layout system prefers an internal placement whenever possible. However, (i) visual objects may not offer enough space to accommodate internal labels or (ii) an internal placement would hide important features of the visual object. In these situations an external placement is a good alternative. Therefore, the layout system evaluates the score of an external and internal placement and classifies them according to these results. Users can adjust the weights of all evaluation criteria.

**Frame-Coherency:** In interactive environments the visual discontinuities of layout elements during user interactions should be minimized. We assign the topmost priority to stabilize annotation boxes so that external labels have to be placed at unused backgrounds.

We introduce a novel approach, which combines 3 types of labels within a uniform framework based on distance fields and which employs agents to guarantee a smooth and frame-coherent transition of layout elements during an interactive exploration of complex 3D models. Moreover, a new layout algorithm for internal labels improves their quality while they can still be computed in real-time. Finally, our layout algorithm only relies on analyzing ID-buffers while it overlays scene renditions with its results. Hence, they can be integrated into arbitrary applications as long as they provide synchronization means.

This paper is organized as follows: Sec. 2 reviews the related work. The interface of our layout system with external applications is presented in Sec. 3. The main steps of the layout algorithm: (i) the determination of candidates (Sec. 4), (ii) their evaluation (Sec. 5), and (iii) the incremental improvement of an initial layout (Sec. 6) are discussed separately. We present some examples and discuss directions of future research (Sec. 7). Finally, we summarize the contributions of this paper (Sec. 8).

## 2 Related Work

There are few guidelines for illustrators how to place labels in scientific illustrations (but see [11, 24]). Frequently, label layout systems transform Imhof's [12] informal guidelines to place labels on maps into optimization problems (see [6]). However, the majority of layout algorithms are restricted on placing labels for point features in static maps. Recently, also line and area features were integrated into real-time layout algorithms for dynamic maps [17]. The computational complexity of these approaches [14] and similarities with graph drawing problems attracted researchers from computational geometry to develop efficient algorithms [13]. Here, the layout of external labels was first analyzed [2]. Finally, these algorithms are also applied to automatically label complex charts [16] or to enhance information visualization with dynamic labels [8].

In Augmented and Virtual Reality the term *view management* was introduced for a more general, but related problem: The smooth integration of additional 2D information (images or texts) into the view plane [4]. While the underlying layout algorithm [3] manages free spaces in an efficient manner, it suffers from the approximation of layout elements via bounding boxes. Consequently, it does not recognize all visible objects and can place anchor points and internal labels on invalid positions.

Other researchers integrate textual annotations into interactive 3D visualizations [18, 20, 1] or renditions of volumetric data [5]. But these systems either rely on a manual label placement [20], implement a single layout style with fixed weights for the evaluation criteria [18], or employ shape approximations by bounding boxes [4] or convex hulls [1, 5].

Few systems integrate internal labels, as their layout algorithm has to consider the shape and extent of visual objects on the projection. A horizontal placement of internal labels within the center of the bounding box [4] often results in an incorrect placement, i.e., label texts do not overlay their associated visual object. Skeleton-based layout approaches suffer from high curvatures and their sensitivity to changing silhouettes [9].


(a) Rendering queue (b) Frame buffer (c) Distance field

**Fig. 2.** Process flow and important data structures

## 3 Framework

In order to ease the integration of our label layout algorithm into arbitrary applications, the scene rendition and the layout are decoupled (see Fig. 2). The layout system uses several G-buffers. The application has to provide a color-coded rendition (ID-buffer) as well as the scene rendition (frame-buffer). Moreover, the color-coding schemata for the individual scene elements has to be conveyed during the initialization phase. The contents of textual annotations can be accessed and altered during an interactive exploration of the scene. This information is represented within a XML file, which can be loaded and stored dynamically. It contains the object identifiers of the 3D model, and their associated internal, external, and extensive descriptions. Finally, the layout and the application system have to be synchronized, as the layout system purely overlays the scene renditions.

**Distance Transformations:** Hartmann's discussion of evaluation criteria for an effective and aesthetic label layout [10] heavily refers to the distance between textual annotations and visual elements. In a common label layout style, external labels are arranged within a constant distance to the object's silhouette (see [1]). Analogously, the distance between annotations boxes and their associated visual elements should be minimized in order to ease the recognition of their co-referential relation. Finally, internal text strokes should be placed over the most salient regions of visual objects (i.e., the skeleton). Thus, the efficient computation of distances to the nearest point on the silhouette is crucial for all components of our layout system.

Using the ID-buffer we first perform a fast *distance transformation* to propagate the distance to the objects' silhouettes for all pixels in linear time. Our distance transformation incorporates a Manhattan distance metric and results in a *distance field* (see Fig. 2-right) As a trade-off between computational speed and quality we use a four pass algorithm [7], while reducing the resolution of the resulting distance field to one fourth of the original ID-buffer image. All subsequent steps for the determination of the candidates are based on the distance field and the ID-buffer.

**Label Classification:** Our approach assigns priorities to different layout elements. Both the labels and the annotation boxes can be considered as different views onto underlying knowledge: labels present the denotation for visual elements while annotation boxes convey comprehensive descriptions or explanations. As annotation boxes are only presented as a result of user interactions, the topmost priority is assigned to their optimal placement.

Short descriptions can be displayed within internal or external labels. Our layout system prefers an internal placement as (i) internal labels directly convey the co-referential relation between textual and visual elements and (ii) external labels compete with annotation boxes for this common resource — the unused background. However, visual objects might not offer enough space to accommodate an internal label as well es the quality of the resulting text stroke might spoil the readability in some cases.[1] In these situations an external placement is necessary. Therefore, we first evaluate the quality of internal placements and then determine the locations and extensions of annotation boxes . Finally, the remaining space is shared among the external labels.

**Architecture:** The classification of labels (internal labels, external labels, or annotation boxes) as well as their positions, extend or alignment defines the *search space* for the label layout problem. We apply several heuristics to prune the search space of this complex optimization problem. First, the number of label placement candidates is reduced (Sec. 4). Then, these candidates are evaluated according to class-specific metrics (see Sec. 5). Finally, local strategies are applied to enhance an initial layout and to minimize the flow of layout elements during user interactions (Sec. 6).

## 4 Determination of Candidates

In our approach, some functional or aesthetic requirements are used to filter out unfavorable candidates in order to solve the layout optimization in real-time.

**Internal Labels and Annotation Boxes:** Internal labels should be placed within salient regions of visual objects, while annotation boxes are placed on the unused background. The ability of an internal label or an annotation box to present comprehensive descriptions is estimated by the maximal size of a square inside a visual object or on the background. We exploit distance transformations, i.e., minimal distances to object silhouettes, to determine the salience of candidates for both label types: For any location with a distance value $d$, it is possible to allocate a square with an edge length $d$ centered on this position. The square may contain arbitrary shaped objects. Note, that only the area required to accommodate the object is allocated.

In other words, these layout algorithms are based on tilings of the area of visual objects or the unused background with squares, which host text strokes of internal labels or the content of annotation boxes (see Fig. 3-left). We experimented with tilings with maximal squares and tilings with equal sized squares and have chosen the latter one in our current prototype. In order to obtain candidates for an internal placement or for annotation boxes, we determine local maxima within the distance field masked with the color-coded region of their respective objects or the background code in the ID-buffer. To reduce the number of convenient label placement candidates, we also consider a minimal labels extent $d_{min}$. For internal labels we use the size of a letter and for descriptions the size of the smallest reasonable annotation box.

[1] These conditions are evaluated by several metrics with user-defined weights.

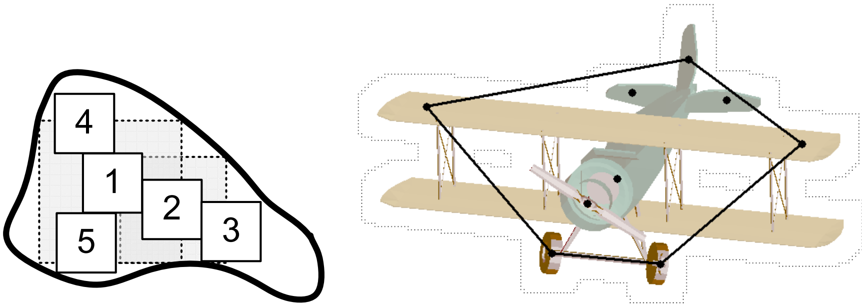



**Fig. 3. Left:** Label candidates on complex shaped visual objects with minimal (numbered boxes) and maximal (filled dashed boxes) extent. **Right:** Plane with orbit (dotted line) and convex hull (lines) of anchor points (dots).

Initially, all the space is unallocated. Then we determine the global maximum of the unallocated space. If its value $d$ is greater than $d_{min}$, we store its position and the distance value in a result list. Subsequently, we allocate an area of the size $d_{min}$ around that maximum. This process is continued until no more maxima greater than $d_{min}$ can be found in the remaining unallocated space. As the result we get disjoint label areas.

**External Labels:** For objects which are neither annotated internally nor expanded to annotation boxes, external labels are placed on the remaining background space. Our approach implements a silhouette-aligned label layout style with a constant distance between the object's silhouette and external labels (see Fig. 3-right). Violating this *label orbit* may disturb the viewer and means an additional cognitive load to interpret this differentiation.

The label orbit can be computed very efficiently on the distance field masked with the background code in the ID-buffer.[2] The candidates to place external labels are extracted in a single pass by detecting positions with the correct distance.[3] Finally, a label alignment has to be determined, which does not result in overlaps with visual objects, annotation boxes, or with already placed labels.

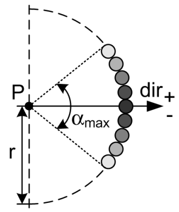
*Alignment:* Our system implements a *circular* label layout. The current implementation purely tests a left and a right aligned candidate. If such an external label is successfully placed, it allocates a rectangular area corresponding to its bounding box.

*Relevance:* All external labels share the same resource, the unused background. The label layout is done in a greedy manner: whenever annotation boxes or external labels are placed, they reserve their display area. Hence, the most difficult candidates should be placed first. Our new layout algorithm adapts the strategy of experienced human

[2] Our previous approaches relied on shape approximations with convex hulls. However, this is not appropriate for objects with an irregular shape. Therefore, additional repair strategies such as to shift external labels towards their associated visual objects had to be applied.

[3] Note, that this results in candidates with a constant distance to the objects silhouette as well as to the image boundaries. However, unwanted boundary candidates can be filtered out during the determination pass.

| Internal Labels | External Labels | Annotation Boxes |
|---|---|---|
| ↑Salience | ↓Distance | ↓Distance |
| ↓Steepness | ↓Angle | ↑Area |
| ↓Curvature | ↓Repelling | |
| ↓Delta | ↓Delta | ↓Delta |
| ↓Info.-Hiding | ↓Info.-Hiding | |




| Label Path | $\alpha_{max_0}$ | $\alpha_{max}$ | $dir$ |
|---|---|---|---|
| Horizontal | 0° | 0° | |
| Straight | >0° | 0° | |
| Convex | >0° | >0° | + |
| Concave | >0° | >0° | - |
| Sidled | >0° | >0° | |

**Fig. 4.** Labeling metrics (left). ↑ high or ↓ are low values are preferred. Text stroke parameters (center) and features (right).

illustrators: label complicated areas from inside out. Therefore, we determine the convex hull of all anchor points. Objects which anchor points are located with in a big distance to the nearest point on the convex hull are labeled first (see Fig. 3-right).

## 5 Initial Layout

This section introduces the metrics which aim at guaranteeing a functional and aesthetic layout of labels with the help of distance transformations and ID-buffers. Subsequently, we employ greedy placement algorithms to determine an initial layout.

**Metrics:** The majority of the metrics in Table 4-left to evaluate different label types were introduced by Hartmann et.al. [10]. Internal labels should be placed in *salient* regions of visual objects and *steep* and highly *curved* text strokes should be avoided. For external labels the length of the connecting lines or the *distance* to their anchor points should be minimized. Furthermore, some label styles aim at minimizing the *angle* between the connecting lines and the main axis. Finally, illustrators might target at a uniform spacing between labels. Therefore, *repelling* forces between labels are defined. Annotation boxes should be placed near to their associated objects (*distance*). It is more likely, that large background regions could host an annotation box during user interactions for a longer time. Therefore, we prefer maximal region within the available background space (*area*). All layout elements aim at minimizing their flow during user interactions (*delta*).

Internal labels and meta-graphical objects (connecting lines and anchor points) directly overlay parts of the visual object. Therefore, regions with few visual features should be preferred for their placement. Within a new metric we examine the amount of *information hiding*. We determine the amount of information hiding by analyzing the standard deviation of the pixel's color within the occluded areas in the frame buffer.

**Internal Labels:** Since mono-spaced internal text stokes do not look appealing, we determine individual spacings $r$ between its letters. The text stroke layout algorithm is based on two assumptions: (i) subsequent letters are placed on an orbit with the radius $r$ and (ii) for smoothly curved text strokes the curvature must not exceed a constant angle $\alpha_{max}$ at any letter.

Our layout algorithm can be controlled via two parameters, the maximal steepness angle $\alpha_{max_0}$ and the maximal angle between two subsequent letters $\alpha_{max}$ (see

Fig. 4-center). Additionally, the text stroke should be placed over salient regions of the visual object. Therefore, the layout algorithm considers the distance field values to determine the best position $P$ for any subsequent letters.

The text strokes of internal labels are placed within squared tiles, where with a local maxima of the distance field in the tile's center. Figure 5 illustrates the algorithm to layout the text stroke *'Labelpath'* within a candidate tile. We first place the midst letter (*'l'*) on the center point of the square, i.e., the position with the highest distance value, indicated by intensity values. Then the direction of the path extension (on the right hand side of Fig. 5) is determined according to the peak of the distance field in the radius $r$ and according to the maximal steepness angle $\alpha_{max_0}$ (see Fig. 4-center). The next letter (*'p'*) is placed on the new determined position $P'$. Then the direction *dir* of the vector between both letters is determined. For the subsequent letters the algorithm processes in the same fashion. However, we now have to consider the distance values with the maximal angle $\alpha_{max}$ relatively to the last direction *dir*.

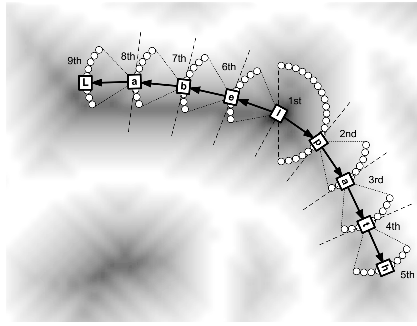



**Fig. 5.** Chain-like layout of the labeling path on an underlying distance field

When the algorithm reaches the end of the second segment of the text stroke, the positions of the first segment are determined in reverse order (on the left hand side of Fig. 5), starting from the mid position. For performance purposes, a set of rasterized concentric circles is precomputed. Therefore, the determination of maxima of the distance field within a segment of a circle is very efficient.

There are even more possibilities to influence the paths' appearance. By setting a specific value of the angle $\alpha_{max_0}$ (while $\alpha_{max} = 0$), any direction of the text stroke can be enforced (see Fig. 4-right). Furthermore, by adding weights (intensities in Fig. 4-center) to the elements of the circles' sector, low angles can be privileged.

The *readability* of internal labels depends on an optimal background contrast. The approach presented in this paper first considers the information hiding, based on the standard deviation of the pixel's color in the hidden areas. First of all, important visual features should be preserved. Moreover, more uniform regions provide a better contrast to overlaying objects. So, both aims are well aligned.

In order to guarantee a minimal contrast, we add a white halo to black letters. This technique is used by many illustrators. Some invert the color of internal text strokes or even single letters in dark regions. Our implementation shows, however, that the visual clutter outweighs the benefit of the readability. Currently, readability problems can still occur due to aliasing effects. We are examining how to use adaptive sampled distance fields to get a grip on that problem.

For each candidate, this analysis is done with several variations of the starting angle and the maximal angle, in order to get horizontal, straight and curved samples of labelpaths. Next, these path candidates are evaluated and scored using the metrics described above. The winning candidate is chosen while the others are discarded.

**Annotation Boxes:** The candidates are evaluated and the best combination of them is used while the others are discarded. Initially, annotation boxes of minimal size are evaluated. After the selection procedure, all the annotation boxes are expanded until they hit other annotation boxes or they reach their maximal size (according to the distance field value).

**External Labels:** Like annotation boxes, external labels are also evaluated and placed sequentially. Thus, it is important to place the most relevant labels first. In order to determine the alignment on the objects' orbit, the current implementation tests both the left- and the right-aligned label candidates. First, it tests whether these candidates are inside the view-port. Overlaps with visual objects, annotation boxes, and already placed labels are prohibited as well. All validity checks are based on point-area tests on the ID-buffer and on rectangle-rectangle tests. Then, all remaining candidates are evaluated according to our weighted metrics. The labels are placed at the positions with the highest score and allocate rectangular areas corresponding to their bounding boxes.

## 6 Label Agents

Our main goal is to support the interactive exploration of complex spatial configurations. Internal and external labels are used as links to more comprehensive explanations and descriptions displayed in annotation boxes. Hence, the main focus of the user is on inspecting and navigating through 3D scenes and to study information displayed in annotation boxes. As any recognizable change attracts the attention of the user, it is crucial to reduce the flow of supporting layout elements. However, coherence preserving techniques and the requirements of a pleasing static label layout might contradict. Therefore, we have to find a trade-off between an optimal (static) layout and (dynamic) coherency.

The layout algorithm of the previous sections employs several heuristics and relevance measures to reduce the search space of a complex layout task and thus might result in non-optimal layouts. The algorithms presented in this section aim at refining an initial layout by *label agents*: local strategies to optimize the layout and to minimize the flow of layout elements. We define two classes of agents:

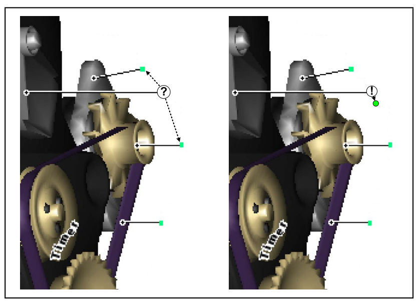


**Fig. 6.** Label-neighborhood agent

*Pixel-neighborhood agents:* These agents evaluate potential improvements if the associated label would slightly change its position. Hence, they consider all alternative placements in the 8-neighborhood in the viewing space and evaluate them according to the metrics presented in the previous section. If they detect a better alternative, they proceed to this new position (*hill climbing*). This class of agents is used for all 3 types of labels.

*Label-neighborhood agents:* Reynolds [19] proposed a behavior model of agents which purely relies on the position and target vector of agents within a restricted sensing area. Simple behavior patterns determine new target vectors; their combination results in astonishing complex behavior patterns of agent groups. In this paper we present a new application of Reynolds' flocking behaviors to improve complex label layouts. User interactions or animations introduce a position and target vector for all types of labels. We will refer with the *agent's position* to the position of the associated label in the previous frame and the *agent's target vector* refers to a vector defined by the suggested position for the current frame.

These agents are employed to redirect external labels. They consider the distance to their immediate label neighbors and implement the *separation* behavior with repulsive forces to the neighbors (see Fig. 6) to distribute labels uniformly or the *cohesion* with attracting forces and *alignment* behavior to form clusters. To ensure the ideal distance of labels to the silhouette, the agents are only allowed to move along the previously computed orbit.

**Internal Labels:** These label agents aim at preserving the position of all letters on the internal text stroke. Internal text strokes are lazy, i.e., they try to remain on their previous position in screen space. However, we also have to ensure unambiguity, i.e., we have to test whether the text stroke is placed over its associated visual object. Therefore, we test the ID-buffer on the area of the letters. If one of the letters is outside its object, the label path is a re-layouted. Otherwise a pixel neighborhood agent determines, if there is a better alternative in the neighborhood and shifts the entire text stroke.

We also applied *flocking behaviors* to layout internal labels. This concept seemed to be intuitive to achieve aesthetic text strokes. Individual letter agents aim at being on an ideal distance (*separation* and *cohesion*) while being oriented homogeneously (*alignment*). However, our chain-like text stroke layout outperforms this approach both in terms of efficiency and quality.

**Annotation Boxes:** The agents which are associated to annotation boxes evaluate at each frame whether their current position and extent is still valid. They redirect the box if they observe a position in the direct neighborhood with higher distance value. If the distance field reveals that an annotation box is slightly too big, they employ the shrinking behavior. The annotation box can be extended when there is more available space. Finally, it collapses and fades in on a new position if a minimal size has been reached.

**External Labels:** External labels employ both types of agents. Pixel-neighborhood agents aim at smoothing transitions of anchor points and label-neighborhood agents aim at achieving label layout styles.

Anchor points should be placed over salient regions of a visual object. Therefore, they are placed on the global maximum of the distance field masked with the objects ID-buffer. This might imply drastic jumps between subsequent frames during user interactions. Anchor point agents consider the neighboring positions of the distance field and change their position frame per frame toward the maximum distance value.

External labels have to reside on a point of the objects orbit. To ensure a clustered or evenly distributed layout, label-neighborhood agents are used for a smooth transition to a new valid position on the objects orbit.

**Agents Communication:** Internal and external label agents communicate to reduce the visual flow of layout elements. Labels might jump between internal and external placements. These label re-classifications, however, imply a disrupt jump of layout elements and should be minimized. Therefore, the *layout gain*, i.e., the quality difference between an internal and an external candidate, has to exceed a cutoff value before the layout system allows the reclassification from an external to an internal label. In these situations, the area of the associated visual object is likely be able to host an internal label for some time with high quality internal text strokes.

## 7 Results and Future Work

The layout algorithms presented in this paper are designed to be used in a wide range of applications. Our current prototype employs Coin3D to render 3D scenes and a graphical user interface based on Qt. Since our layout algorithms work in image space, the computational effort is mostly independent of the complexity of the 3D models. Currently, we conduct a user to evaluate our new approach.

The performance of our application has been tested on several computers with different models of the Viewpoint 3D library with a resolution of 1024*768 pixels. Independently from the 3D model we used, for the layout time we got almost stable timings. On our test system (Intel P4 3.3 GHz, GeForce 6600) the computation of one frame took about 20 milliseconds. Without user interactions, the label agents reached local maxima of the evaluation function in less than 1 millisecond. For supplemental material see our video at: http://wwwisg.cs.uni-magdeburg.de/∼timo/AgentBasedLabeling/.

Since the computation of the distance transformation uses the majority of the time, we plan to speed it up by employing GPU-based distance transformations (e.g., footprint splatting [22] or pre-computation of distance fields in 3D [23]).

## 8 Conclusion

This paper offers complex layouts which (i) integrates internal and external labels of arbitrary size and shape, (ii) provides parametrized functions for the layout styles, and (iii) maintains the frame-coherency through label agents. We present new real-time layout algorithms for all label classes which employs the same data structure — distance fields. Even though qualitative evaluations are still missing, we believe that their quality outperforms all previous approaches.

## References

1. K. Ali, K. Hartmann, and T. Strothotte. Label Layout for Interactive 3D Illustrations. *Journal of the WSCG*, 13:1–8, 2005.
2. M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary Labeling: Models and Efficient Algorithms for Rectangular Maps. In *12th Int. Symp. on Graph Drawing*, pages 49–59, 2004.
3. B. Bell and S. Feiner. Dynamic Space Management for User Interfaces. In *Symp. on User Interface Software and Technology*, pages 238–248, 2000.
4. B. Bell, S. Feiner, and T. Höllerer. View Management for Virtual and Augmented Reality. In *Symp. on User Interface Software and Technology*, pages 101–110, 2001.

5. S. Bruckner and E. Gröller. VolumeShop: An Interactive System for Direct Volume Illustrations. In *IEEE Visualization*, pages 671–678, 2005.
6. J. Christensen, J. Marks, and S. Shieber. An Empirical Study of Algorithms for Point-Feature Label Placement. *ACM Transactions on Graphics*, 14(3):203–232, 1995.
7. O. Cuisenaire and B. Macq. Fast Euclidean Distance Transformations by Propagation Using Multiple Neighbourhoods. *Computer Vision and Image Understanding*, 76(2):163–172, 1999.
8. J.-D. Fekete and C. Plaisant. Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. In *SIGCHI Conf. on Human Factors in Computing Systems*, pages 512–519, 1999.
9. T. Götzelmann, K. Ali, K. Hartmann, and T. Strothotte. Form Follows Function: Aesthetic Interactive Labels. In *Computational Aesthetics 2005. EG WS on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 193–200, 2005.
10. K. Hartmann, T. Götzelmann, K. Ali, and T. Strothotte. Metrics for Functional and Aesthetic Label Layouts. In *5th Int. Symp. on Smart Graphics*, pages 115–126, 2005.
11. E. R. S. Hodges, editor. *The Guild Handbook of Scientific Illustration*. John Wiley & Sons, New York, 2nd edition, 2003.
12. E. Imhof. Positioning Names on Maps. *The American Cartographer*, 2(2):128–144, 1975.
13. K. G. Kakoulis and I. G. Tollis. A Unified Approach to Labeling Graphical Features. In *14th Ann. Symp. on Computational Geometry*, pages 347–356, 1998.
14. J. Marks and S. Shieber. The Computational Complexity of Cartographic Label Placement. Technical Report TR-05-91, Center for Research in Computing Technology, Harvard University, 1991.
15. R. E. Mayer. *Multimedia Learning*. Cambridge University Press, Cambridge, UK, 2001.
16. S. Müller and A. Schödl. A Smart Algorithm for Column Chart Labeling. In *5th Int. Symp. on Smart Graphics*, pages 127–137, 2005.
17. I. Petzold, G. Gröger, and L. Plümer. Fast Screen Map Labeling — Data Structures and Algorithms. In *21st Int. Cartographic Conf.*, 2003.
18. B. Preim, A. Raab, and T. Strothotte. Coherent Zooming of Illustrations with 3D-Graphics and Text. In *Graphics Interface*, pages 105–113, 1997.
19. C. Reynolds. Steering Behaviors for Autonomous Characters. In *Game Developers Conf.*, 1999.
20. F. Ritter, H. Sonnet, K. Hartmann, and T. Strothotte. Illustrative Shadows: Integrating 3D and 2D Information Display. In *Int. Conf. on Intelligent User Interfaces*, pages 166–173, 2003.
21. B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualization. In *IEEE Symp. on Visual Languages*, pages 336–343, 1996.
22. R. Strzodka and A. Telea. Generalized Distance Transforms and Skeletons in Graphics Hardware. In *EG/IEEE TCVG Symp. on Visualization*, pages 221–230, 2004.
23. A. Sud, M. A. Otaduy, and D. Manocha. DiFi: Fast 3D Distance Field Computation Using Graphics Hardware. *Computer Graphics Forum*, 23(2), 2004.
24. E. R. Tufte et al. Ask E.T.: Mapped Pictures: Image Annotation, 2005. http://www.edwardtufte.com/bboard/.

# Experiments in the Perception of Causality

Eric Neufeld, Jeff Solheim, and Sonje Kristtorn

Department of Computer Science, University of Saskatchewan, 176 Thorvaldson Building, 110 Science Crescent, Saskatoon, SK, Canada, S7N 5C9
eric@cs.usask.ca

**Abstract.** Michotte argued that humans almost immediately perceive causality in simple animations as a consequence of spatial and temporal localities. To explore the usefulness of these ideas in animation, Ware *et al.* introduced a semiotic method called a visual causal vector. This work studies their wave metaphor, shows a method for calibrating timings to produce a causal impression, and explores the effect of animations timed in this way.

**Keywords:** Perception, Causality, Data Visualization.

## 1 Introduction

In the 1930s, Michotte [8] claimed that perception of causality was preattentive[1] or something like that. In his first chapter, Michotte cites Hume and (briefly) Kant, who, in broad strokes, inquired how inferences of causality from data could be justified. In the end, Hume saw only conjunction, but not connections, between cause and effect. Hume "expressly asserted" that causality is not perceived directly. In contrast, Michotte saw a different phenomenological view of instantaneously perceived causality:

> … certain physical events give an immediate causal impression, and that one can 'see' an object act on another object, produce in it certain changes, and modify it in one way or another … e.g., a hammer driving a nail into a plank, and that of a knife cutting a slice of bread. The question that arises is this: when we observe these operations, is our perception limited to the impression of two movements spatially and temporally co-ordinated … Or rather do we directly perceive the action as such – do we see the knife actually cut the bread?

Michotte carried out studies of perception, and concluded that the perception of causality can be as direct and immediate as the perception of simple form. Using a mechanical apparatus consisting of little mirrors and beams of light (to avoid the expense of celluloid), one rectangular patch of light would be moved from left to right until it just touched a second patch of light then stopped (illustrated in Figure 1). At about this point in time the second patch of light would be made to move. Depending on the temporal relationships between the moving light events and their relative

[1] Strictly speaking, *preattentive* seems to have entered the lexicon possibly with Triesman[10], long after Michotte did most of his work, and after his major work was translated,). Healy [1] defines preattentive processing as processing that takes place in less than 200 ms. Healy argues preattentive figures may be useful for visualizing multivariate data.

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 36 – 49, 2006.

velocities, observers reported perceiving different kinds of causal relationships that Michotte variously categorized as *launching*, where one object causes another to move by striking or expelling it, *entraining*, where one object causes another to move by carrying it along with it, and *triggering,* where the action of an object depends upon a preceding action. These causal relationships were not reported when the second rectangle moved at right angles to the first.

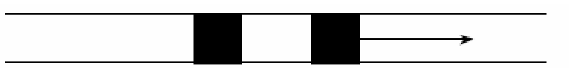


**Fig. 1.** Michotte studied the perception of causal relationships between two patches of light that moved always along the same line but with a variety of velocity patterns

Although developmental work by Leslie and Keeble[6] has shown that infants at only 27 weeks of age can perceive causal relations such as launching, Michotte's work does not seem to be well known. According to Lucio and Milloni [7], recent renewal of interest in his work is due mainly to researchers interested in human-machine interfaces, such as Ware *et al.* [15], and Kerzl and Hecht [4]. If Michotte's claims are correct, his results may be of value in the production of graphical animations depicting causal events and causal relationships [9].

To address these questions, we performed a series of experiments to determine the usefulness of what Ware *et al.* [15] introduced as VCVs (Visual Causality Vectors). In particular, we explored the wave metaphor of Ware *et al* as a device for implementing node-link diagrams ultimately intended to abstract cause-effect relationships. In the first experiments, the wave animation was calibrated to ensure that its timing (e.g., spatiotemporal locality) was both consistent with Michotte's reported numbers and produced a causal impression.

As some writers dispute the value of animation in data visualization [12], the second set of experiments, measured the value of such causal animations. Following Irani and Ware [2], we designed experiments to measure the memorability of causal animations, as compared to both non-causal animations and static diagrams. The results suggest that causal animation can significantly improve performance, but may not be the only factor.

## 2 Visual Causality Vectors

Ware et al. [15] define a visual causality vector as "a graphical device that conveys the sense that a change in one entity has caused a change in another entity".

Previous research into the perception of causality uses simple, animations of objects such as squares and circles, which are presumed to have no expected behaviour associated with them. In most studies that examine the launching effect, one of the two objects is always stationary. Two of the visual causality vectors (VCVs) defined by Ware *et al.* follow this paradigm.

In the pin-ball metaphor, a ball representing the VCV moves up to, and strikes another ball representing a node in a graph, which then oscillates. The prod metaphor (not shown) is similar to the pin-ball metaphor in that the target node oscillates when it is struck by the VCV. In this metaphor the VCV is represented by a rod. Although the movement of the target is more complex, both of these metaphors closely resemble Michotte's launching effect. Only one object is moving at any given time, and the structure of both the VCV and the node is quite simple in both cases.

The wave metaphor is based on a wave-like (spline) function animated along an arc. The impact of a wave is shown by a node with a circular component that "floats" as the wave arrives.

The wave metaphor has more representational flexibility. Causal power or force can be shown by varying the size or speed of the wave or the node, or negative causal force could be represented by an upside-down wave [15]. A dying wave can represent weak causal links, and so on. This flexibility was the main factor in the choice of the wave metaphor for these experiments.

The node animation in the wave metaphor was modified here. Rather than floating when the wave arrived, the node grew in size when the wave arrived. This was intended to give the node the appearance of filling up as the wave arrived. Then the node contracted to its original size to expel the next wave in the graph. This representation created a more complex timing situation however. Therefore one aspect of Experiment 1 was to refine the timing of the animation.

# 3 Experiments

## 3.1 Experiment 1: Calibrating Causal Parameters for the Wave VCV

Due to space constraints, and the results of Experiment 2, the description of this experiment has been shortened.

The first experiment tried to establish timings for our implementation the wave VCV. Subjects were presented with a software wave animation together with a set of software controls that they could use to modify the timings of the different events in the animation. This observer control of the animation is a unique feature of the experiment. Our hypothesis was that the expected value of the inter-event timings would be consistent with those reported by Michotte.

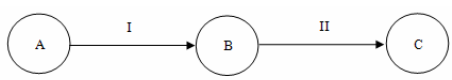



**Fig. 2.** Experiment 1 Graph

As a training exercise and to control for learning effects, subjects, 50 first year undergraduate engineering computer science students, were first provided with the software and given (intentionally vague) instructions to "correct" the animation. Then they were given a graph similar to Figure 2 and asked to adjust control sliders so that the arrival of a wave appeared to cause a node to expand, and so the contraction of a node caused a wave to be expelled. According to Michotte, for perception of causality (launching) to occur, the second (struck) object should start moving within 160 ms of contact. He breaks this interval into direction launching (0-70 ms) and delayed launching (70-160). Because the nodes take 200 milliseconds to fill and the wave takes 1000 ms to travel the length of the arc, our expected value for $t_1$, the launch time of the wave was 200 ms, and our expected value for $t_2$, the relative start time for expansion of the second node, was 1000 ms. Experimental results agreed with expectations for $t_1$, but results for $t_2$ were much larger than expected.

In Michotte's work, boundaries between causes and effects are sharp. One square strikes another, and the second one moves. The boundaries of our waves are not like this. Our expectation for $t_2$ was based on the wave traveling the entire length of the arc. In the end, we noticed another potential triggering event, namely, the point at which the end of the wave first touches the second node. The second experiment was designed to confirm this, but also changed the animation make the boundaries between events easier to perceive.

### 3.2 Experiment 2: Finer Calibration

To improve the precision of the measurements, we doubled the animation time for each component, as Michotte's work suggests that causal perception is mostly dependent on the spatiotemporal locality of the associated events, but less dependent of the speed of the animation. Thus, in the new animations, a node's cycle is 1400 milliseconds long – 400 to expand to full size and 1000 to contract. The wave animation takes 2000 milliseconds to complete. The animations were simplified (see below) and enlarged. Given these parameters and the results for Experiment 1, we hypothesized the following ideal causal timings: a wave will start 400 milliseconds after a node ($t_1$ = 400 ms) and a node will start 1600 milliseconds after a wave ($t_2$ = 1600 ms).

**Method.** The animations were changed as described above, and user-controls were more fine-grained. The experiment was conducted one subject at a time, subjects were informed of the purpose of the experiment, provided both verbal and written instructions, and interviewed at the conclusion of the experiment.

The subjects viewed three graphs. Graph 1 consisted of a single node and a single wave that appeared to move off the screen. A slider control controlled the start time of the wave. In Graph 2, a single wave appeared to move onto the screen and towards a node on the right, and the node's start time could be set with a slider. Graph 3 consisted of a wave that appeared from the left side of the screen and moved towards a node and a second wave that moved from the node towards the right side of the screen. Two sliders controlled the start time of the node, and the start time of the second wave. The component not subject to user control was fixed to the beginning of the animation.

**Procedure.** The 25 subjects were university volunteers, mostly computer science graduate students. Each subject performed three tests, one for each graph.

Test 1 measured only $t_1$. Subjects were asked to set the timing so that the node appeared to cause the movement of the wave. Test 2 tested $t_2$, and subjects were told to set the timing so that the wave appeared to cause the node to expand. At the conclusion of each test, subjects were asked, "Was there a particular point in the animation of the node (wave) where you were trying to get the wave (node) to start?" Test 3 tested both variables. The instructions were to set the timings so that the first wave appeared to cause the node to expand and the node to expel the second wave. The purpose of this third test was to determine whether the results of the first two tests would hold for a sequence of events. At the conclusion of this test, the question was, "were you trying to achieve a different impression in this test than in the previous tests?"

**Results.** Tests 1 and 2 yielded, respectively, average times of 472 milliseconds for $t_1$ ( s = 395) and 1522 milliseconds for $t_2$ (s = 235). Both results pass the *t* test at the 0.05 significance level. The results of Test 1 contained two potential outliers, deviating from the sample mean by than three standard deviations, and the results are much stronger if these are excluded. Test 3 produced an average time of 758 milliseconds for $t_1$ and 1635 milliseconds for $t_2$. In the Test 1 interviews, 17 of the 25 (68%) subjects indicated that they had tried to match the start of the wave to the point of maximum expansion of the node from which it is expelled. For Test 2, 18 subjects (72%) stated that their intention had been that the node should begin to expand when the front edge of the wave reaches it. These results, and the Test 3 result for $t_2$ support our hypothesis.

Test 3 didn't confirm the hypothesized value for $t_1$. Perhaps the discrepancy in $t_1$ values can be explained by the increased complexity of Test 3. This is supported by the difficulty several subjects reported in achieving the desired effect and dissatisfaction with their results.

Together, Experiments 1 and 2 gave several insights. In the real world, individuals may not universally agree on the spatiotemporal location of causal events. Nonetheless, the spatiotemporal location can be collectively closely approximated by nonexperts, provided interactions are minimized. While the measurement variance is high, the exit interviews suggested a high degree of qualitative agreement about the location of the causal event among. Interestingly, the variance did not change as a consequence of slowing down the animation.

The timings derived here were applied to the animations in Experiments 3 and 4.

### 3.3 Experiment 3: Memorability of Causal Animations

This experiment was modeled on Irani and Ware [2,3], which tested the memorability of diagrams incorporating geons as compared to Unified Modeling Language (UML) diagrams. (UML diagrams use generic 2d geometries, while geons have additional, but superfluous, 3d geometry.) To see whether the additional information made the graphs easier to remember, Ware and Irani showed seven UML diagrams to subjects. One hour later, they asked the subjects to pick these seven UML diagrams from a larger sequence of fourteen UML diagrams. The experiment was repeated with different subjects, using geons. Irani et al found subjects made half as many errors recalling node-link diagrams drawn with geons. Analogously, we hypothesized that subjects would find graph structures with an associated causal animation more memorable than static graph structures with just arrows showing the information flow.

Thus, an arrow representation was used for the static base case and the causal wave animation as defined in Experiments 1 and 2 was used for the second case. This animation contained no other information to indicate direction. Following Lee et al [5], our experiment included a third representation combining standard static arrows and the causal wave animation. The results from Lee et al suggested that this would be the most effective representation. Figure 3 shows screen captures of two representations.

**Hypotheses.** We hypothesized that the added information of causal timing and wave structure, would improve performance on the memory task. We also hypothesized the third animation would result in better performance than the first two.

**Method.** Subjects were Computer Science undergraduates in classes with non-overlapping enrolments. Experiment 3 consisted of three parts, corresponding to the three types of representation.

In each part, subjects were shown fourteen moderately complex graphs. Each graph had between six and nine nodes. Four pairs of graphs had identical node structure and only the arcs differed. The other six graphs differed both in node structure and arc layout. A subset of seven graphs was selected from the full set of fourteen to be shown to students at the beginning of the experiment.

The experiment was conducted during class time. At the beginning of the class, subjects were shown the seven-graph subset. Each graph was displayed for 15 seconds on a large screen at the front of the classroom, enough time to let the animations run three times. At the end of the class (approximately 60 minutes later), the subjects were presented with the full set of fourteen graphs and asked to indicate on a sheet of paper whether or not each graph had appeared at the beginning of the class.

Subjects in each part of the experiment saw the same seven-graph subset, and the graphs were presented in the same order. Subjects were told at the start of the experiment that all seven of the graphs in the subset would appear in the full set.

**Results.** The experimental results supported the first hypothesis, but not the second. Causal animation provide a substantial benefit over a static representation with respect to memorability, but the combination of static information and causal animation was not significantly better than the animation alone.

Subjects were scored by number of correct responses out of a possible 14. The subject pool consisted of 48 subjects in Group 1, 49 subjects in Group 2, and 39 subjects in Group 3. Results were tested for significance using the z-test using $p = 0.05$.

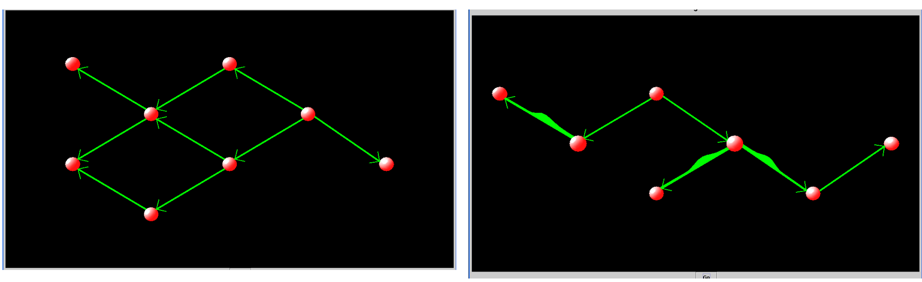


**Fig. 3.** Screen Captures from Experiment 3

Subjects had an average error rate of 28% when presented with static graphs (Group 1), as compared to 23% for animated graphs (Group 2), and 22% for graphs with both animation and static information (Group 3). A z-test ($p < 0.05$) shows this to be a significant improvement from Group 1 to Group 2 and from Group 1 to Group 3 ($p < 0.01$), but the improvement from Group 2 to Group 3 ($p < 0.35$) is not significant. (Eliminating outliers, and applying a binomial test to thresholded scores did not alter the character of the results.)

**Conclusions.** Experiment 3 suggests that the causal wave animation improves performance on a memory task. However, an animated wave is a semiotically rich metaphor. Waves convey certain different kinds of information though shape and motion. Experiment 4 separates out this interaction.

### 3.4 Experiment 4: Causal and Non-causal Animations

Experiment 4 compared both causal and "non-causal" animations to the standard static representation. (A non-causal animation is one that deviates from the established timings.) To separate effects from causal timing from other effects, we introduced a second minimalist visualization consisting only of lighted circles (Figure 4) similar to "comets" [14] moving along the arcs and lighting up the nodes on arrival. Altogether we used five different representations: static, causal waves, causal lights, non-causal waves, and non-causal lights.

**Method.** The subjects, fifty volunteers from first year undergraduate computer science classes, were presumed to have relatively uniform experience with computers in general and graph diagrams in particular. Subjects received a questionnaire to determine their ability level in relevant areas, and an Eals and Silverman spatial ability test.

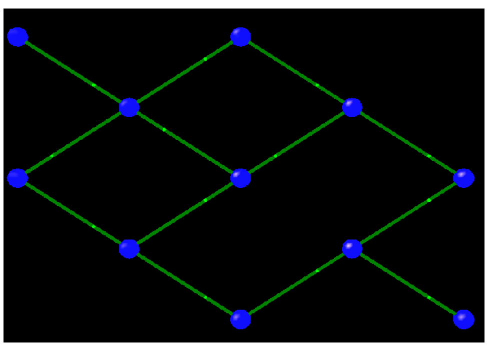


**Fig. 4.** The light metaphor

The subjects were divided into five groups. Each group was tested three times on one of the five representations. Test 1 examined the effectiveness of representation on short-term memory. Subjects were allowed to view a training graph with the given representation for twenty seconds. After this they performed six trials, each consisting of an introduction page and a trial graph with a yes or no question. Figure 5 shows screen captures of some of the trial graphs. The trial graphs in Test 1 had no arcs between the nodes. Trial graphs had one green node and one red node, and subjects were instructed to indicate whether or not there was a directed path (i.e. directed motion or arcs) from the green node to the red node in the training graph "as quickly and accurately as possible”. Response times were tracked and compared against the response times of the other training scenarios. Following the sixth trial, the subjects were asked to recreate the location and direction of the arcs on paper.

Test 2 examined the effectiveness of the various representations on task performance, but using graphs connected with static arrows as trial graphs. Otherwise, Test 2 followed the same procedure as Test 1.

Test 3 examined the effectiveness of each representation on task performance. It followed the same procedure as Tests 1 and 2 except that the trial graphs used the same representation as the training graphs.

Prior to testing, subjects received a short tutorial that provided detailed instructions and showed examples of a training graph and a trial graph. The example graphs prepared the subjects for what they would see, and helped eliminate training effects. Detailed instructions were also summarized on paper to provide the subjects with the tutorial information later in the experiment if they needed it. As well, the tester gave a brief presentation outlining the layout and purpose of the experiment.

A different graph was prepared for each of the three tests. In each test, all subjects saw a graph with the same topology, though the representation varied. All graphs had similar complexity (approximately twelve nodes and thirteen arcs).

After the third test, subjects were given an exit questionnaire to evaluate the subject’s perception of their own success in this task, as well as to elicit responses about how they arrived at the responses they did.

**Hypotheses.** We made the following hypotheses:

1. Subjects trained on a causal animation will make fewer mistakes and have shorter response times than subjects trained on a non-causal animation.
2. Timing fixed, the light animations will outperform the wave animations.
3. Timing of the animation is more important than its structure. Put another way, Hypothesis 1 takes precedence over Hypothesis 2, so causal waves will outperform non-causal lights.
4. The static training will be less effective than causal animations, but better than non-causal animations.

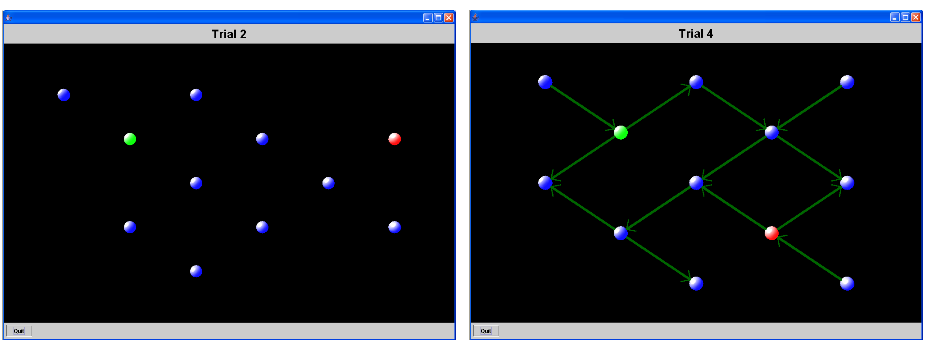



**Fig. 5.** Some trial graphs for Experiment 4

These hypotheses predict the following effectiveness ordering for the training scenarios (from best to worst): causal lights (CL), causal waves (CW), static (STAT), non-causal lights (NCL), non-causal waves (NCW).

For Test 2, we hypothesized that the static graph presentation in the trials would overpower the training. We expected similar performance across all representations in Test 2.

**Results (Summary).** Experiment 4 measured response time, or speed, and accuracy. We say that a representation is better if it shows improvement in both dimensions. As it is rare for this improvement to be significant to $p = 0.05$ for both variables, a representation with a significant improvement in one dimension and no significant improvement in the other is also referred to as a better representation.

Two methods were used to eliminate outliers based on response times. The first method simply removed the slowest response time for each person on each test. The only exception to this rule was one subject, who seemed to have simply clicked through a trial, recording a response time of 0.01 seconds. This happened once for each test, so for this subject the lowest time was removed. The second method involved calculating an upper and lower bound for each subject group on each test. The subject groups were compared as a whole so it didn't seem necessary to remove data points for each person. The upper bound was set at two standard deviations from the mean. The lower bound was, somewhat arbitrarily, set at 1 second. A response time of less than 1 second was presumed to indicate that no attempt was made to actually identify a path in the trial. In addition to these two methods, data was also analyzed with no outliers removed.

Results are reported here using the second method (upper and lower bounds on the test groups as a whole) to remove outliers. All significance tests used the Tukey HSD test with significance level of $p = 0.05$.

**Table 1.** Test 1: Average accuracy and response times in seconds

| | STAT | CW | CL | NCW | NCL |
|---|---|---|---|---|---|
| Speed | 7.94 | 5.99 | 4.53 | 3.99 | 6.97 |
| Accuracy | 62% | 73% | 58% | 67% | 58% |

**Test 1 Results.** Based on initial subjective experience with the animations, we expected the less intrusive light animation to outperform the wave animation. We also expected subjects trained with causal animations would always outperform subjects trained on non-causal animations, and that subjects trained on the static representation would fall in the middle. Table 1 summarizes average accuracy and response times.

All animation techniques resulted in speedups, compared to the static base case. A Tukey test showed two were significant: causal lights ($p < 0.0001$), and non-causal waves ($p < 0.0001$). The increase in speed comes with a trade-off in accuracy of 4% for causal and non-causal lights. Though not all speed increases were significant, the best overall increase over the static representation comes from the non-causal and causal wave animations. The non-causal waves show an increase in speed of almost 4 seconds ($p < 0.0001$) and an increase in accuracy of 5%. Causal waves showed a speed increase of about 2 seconds ($p = 0.06$) and an increase in accuracy of 11%.

The other comparison of note in Test 1 is between non-causal lights and other animations. Causal lights and non-causal waves showed an improvement in response time: 2.4 seconds ($p < 0.01$) and 3 seconds ($p < 0.0005$) respectively. Causal waves showed a (non significant) improvement in speed (0.98 seconds) and accuracy (15%).

While causal timing was better than non-causal timing for each animation structure, structure proved to be more important overall. Though not significantly so, non-causal waves were both faster and more accurate than causal lights. This, along with the good overall performance of the causal wave animation, contradicts the hypothesis that the waves would outperform lights. Overall, causal lights as well as causal and non-causal waves seem to be better than the static representation, and the non-causal lights appear to be no better than the static representation.

**Test 2 Results.** In Test 2, all trials used static arrows. Because it seemed obvious that the representation of direction during trials would overpower any benefits from the training representations, we expected both accuracy and speed to be uniform across all representations. The results are summarized in Table 2. As expected, accuracy is uniformly high. Speed however, varies significantly. As in Test 1, the wave structure proves better overall than the light structure. The increase in speed over the static case was significant for both causal ($p < 0.05$) and non-causal ($p < 0.0001$) waves. As well, these animations showed a small improvement in accuracy. Also as in Test 1, the causal lights perform about as well as non-causal and causal waves, and the other three animation techniques outperform non-causal lights in response time, though this is significant only for non-causal waves ($p < 0.0001$).

**Table 2.** Test 2: Average accuracy and response times in seconds

| | STAT | CW | CL | NCW | NCL |
|---|---|---|---|---|---|
| Speed | 7.01 | 5.47 | 6.00 | 4.37 | 6.88 |
| Accuracy | 93% | 98% | 95% | 96% | 97% |

These results closely resemble results from Test 1. The causal lights and both wave animations seem better than the static representation and the non-causal lights animation performed at approximately the same level as the static representation. Contrary to the hypotheses, there appears to be a benefit from training with animation.

**Test 3 Results.** Test 3 tested the representational abilities of the various animation techniques. As in test 2, we expected that accuracy would be uniformly high across all representational techniques, since all the information needed to find the path was present in each trial. We also hypothesized that the ordering of the effectiveness of the animations would be the same as stated for Test 1. Table 3 shows results for Test 3.

**Table 3.** Test 3: Average accuracy and response times (seconds)

| | STAT | CW | CL | NCW | NCL |
|---|---|---|---|---|---|
| Speed | 5.30 | 4.69 | 4.31 | 4.05 | 6.10 |
| Acc | 79% | 95% | 95% | 88% | 77% |

Table 3 shows that causal animations are significantly more accurate than static graphs: causal waves ($p < 0.05$), causal lights ($p < 0.05$). Both causal animations showed a 16% improvement in accuracy ($p = 0.10$). Except for non-causal lights, all animation techniques significantly outperform static graphs for response times: causal waves ($p < 0.005$), causal lights ($p < 0.0001$), and non-causal waves ($p < 0.0001$).

Although the ordering of the representations does not exactly match the hypothesized ordering, the ordering is similar to those from the other two tests. The three effective animations showed an average increase of about 1 second or 19% in response time over the static representation, and an average increase of 14% in accuracy. The primary difference is that the static representation clearly outperformed the non-causal lights.

**Tradeoff Between Structure and Timing.** We expected causal timing to always outperform non-causal timing, but this was not the case. Table 4 compares timings taken as a whole and structures taken as a whole.

A two-factor ANOVA test showed no significant difference between timing and structure for either response speed or accuracy in Test 1, though the interaction is significant for response speed ($p < 0.0005$). In Test 2 the wave structure proved to result in significantly faster response speeds than the light structure ($p < 0.0001$), while the causal and non-causal timings showed no difference. The interaction was also significant in this test ($p < 0.05$). None of the accuracy differences in Test 2 were significant. Test 3 response speeds showed a significant difference both between structures ($p < 0.005$) and between timings ($p < 0.05$), as well as a significant interaction ($p < 0.0001$). The results of Test 3 also showed a significant improvement in accuracy for casual timing compared to non-causal timing ($p < 0.01$).

**Table 4.** Average accuracy and response times (in seconds)

| | Causal | NC | Waves | Lights |
|---|---|---|---|---|
| Test1 speed | 5.24 | 5.50 | 4.98 | 5.75 |
| Test 1 accuracy | 65% | 62% | 70% | 58% |
| Test 2 speed | 5.73 | 5.64 | 4.93 | 6.00 |
| Test 2 accuracy | 97% | 97% | 97% | 96% |
| Test 3 speed | 4.50 | 5.06 | 4.36 | 5.21 |
| Test 3 accuracy | 95% | 82% | 91% | 86% |

Our hypothesis that timing is more important than structure appears to be wrong for these tasks. The wave structure shows an improvement over the light structure in both speed and accuracy in all three tests, though not all are statistically significant. The only exception is Test 3, where causal timing is both significantly faster and significantly more accurate than the non-causal timing.

**Conclusions.** Experiment 4 generally confirmed expectations. The timing of an animation is important for short term memory, but not necessarily more important than the structure of the animation. It was surprising to learn that an animated training session improved response speed in Test 2 where the static representation was present in all trials.

Some of our hypotheses turned out to be incorrect. We expected both well-timed animations to outperform the static base case, and the two poorly timed animations to be worse. We also expected the light structure to outperform the wave structure. The results indicate that three of our animations are better than the static representation. The causal waves performed well, as expected, and the non-causal lights performed poorly as expected. The surprises came from the causal lights and the non-causal waves. Although still better than the static representation, the causal lights exhibited much lower than expected performance throughout the experiment, and the non-causal waves performed much better than expected, sometimes even outperforming the causal lights.

Some of the anomalous results might be explained by the make-up of the subject groups. The five groups were fairly homogeneous with a few exceptions. The first was video game experience. We asked subjects to rank their video game experience on a scale of one to five. The group trained with non-causal waves ranked their video game experience about 20% higher than did the other groups. This is difficult to interpret, but the difference is quite large.

The Eals and Silverman spatial ability test also shows some variation across groups. This test matches up well with the accuracy factor in this experiment. The subjects in the causal waves group were about 7% higher than average, and the subjects in the causal lights group about 7% lower. This may explain the uniformly high accuracy scores across the three tests for the causal waves, as well as the uniformly low accuracy scores for the causal lights. However, neither of these concerns changes the main character of our results regarding the value of causal timings.

## 4 Summary and Discussion

The training exercise for Experiment 1 – where subjects were given a badly-timed animation and were asked to correct it – suggested that humans might impose causal relationships on a scenario to interpret it, even if they may also impose other relationships such as symmetries.

Experiments 1 and 2 asked subjects to calibrate animations so as to produce a causal impression. Our technique for calibrating causal timings appears to be novel in itself. Although humans can be prone to error, given sufficiently sensitive controls, sufficiently simple animations, the observation of Ware *et al.* [15] that since humans perceive "some relation" in the intervals just before and after the perceived causal event, it seems reasonable to expect the subjects' errors to have a two-tailed normal distribution, plus the concordance of the results with Michotte together suggest that it is reasonable to use the quantitative outcomes of Experiments 1 and 2 in the animations of the subsequent experiments.

Experiment 3 was designed to show the value of causal animation to memorability. Networks with static causal arrows *and* wave animations outperformed static graphs, but did not significantly outperform the animated graphs. Experiment 3 suggested that animation was helpful, but did not clearly show the benefit of *causal* animation.

Experiment 4 was designed to control for causal timing. The four animations produced controlled for both structure and timing by introducing the more neutral "lights" structure. *Non-causal* animations used "random" timings, i.e., timings not expected to generate a causal response. That the wave metaphor (perceived by the designers as intrusive) generally outperformed the light metaphor initially came as a surprise. However, working constantly with a visualization is very different from performing a task (e.g., the Paper Clip phenomenon), and the preference of users might switch with exposure.

More surprising were the strong speed results of the non-causal waves. In all three tests, the subjects performed the task more quickly with non-causal waves, although with some loss of accuracy. However, an interesting explanation emerges if we reconsider these results in light of the aggregate data, which shows that in Test 3, causal timings were significantly better than non-causal timings. Looking at the subcategories, however, non-causal waves outperform causal waves with respect to speed by a small amount, whereas the performance improvement for causal lights over noncausal lights is huge – almost 50%. Thus, we must consider the value of causal timings in the context of the structure.

During design, our subjective impression was that adding causal timings to the non-causal lights were made the animations 'snap' into semantic focus. Meanwhile, it seems waves, with or without causal timings have strong causal content. Part of this may derive from the fuzzy boundaries in the wave animation, which ameliorated the randomness introduced to the timing. This suggests that the value of causal timing is especially valuable if the visualization is not inherently causal.

## Acknowledgements

Thanks to Pourang Irani for discussions and Carl Gutwin for help with the design of Experiment 4. This work was supported by a Discovery Grant from the Natural Science and Engineering Research Council of Canada, a Graduate Teaching Fellowship and a University Graduate Scholarship from the University of Saskatchewan.

## References

1. Healy, C., Booth, K.S., and Enns, J.T. Harnessing Preattentive Processes for Multivariate Data Visualization. In Proceedings of Graphics Interface '93, (1993) 107-117
2. Irani, P. and Ware, C. Diagramming Information Structures using 3D Perceptual Primitives. ACM Transactions on Computer-Human Interaction 10:1 (2003) 1-19
3. Irani, P., Ware, C., & Tingley, M. Using Perceptual Syntax to Enhance Semantic Content in Diagrams. IEEE Computer Graphics and Applications 21:5 (September 2001)76-85
4. Kerzel, D. and Hecht, H. Visual causality. In *Interdisciplinary Perspectives on Causation* Bern:Bern Studies in the History and Philosophy of Science (2000) 119-130
5. Lee, P., Kippel, A., & Tappe, H.. The Effect of Motion in Graphical User Interfaces. In *Proceedings of Smart Graphics 2003*, LNCS 2733 (2003) 12-21
6. Leslie, A.M., and Keeble, S. Do six-month-old infants perceive causality? *Cognition* **25** (1987) 265-288

7. Luccio, R. and Milloni. D. Perception of causality: A dynamical analysis. In *Mind and Causality (Advances in Consciousness Research, V. 55)* , Alberto Peruzzi (Editor) (2004) John Benjamins 19-35
8. Michotte, A. *The Perception of Causality*. London: Methuen & Co. Ltd. (1963)
9. Neufeld, E, Kristtorn, S., Guan, Q., Sanscartier, M. and Ware, C. Exploring Causal Influences. In *Proceedings of Visualization and Data Analysis 2005*, San Jose, 52-62
10. Triesman, A. Preattentive Processing in Vision. *Computer Vision, Graphics and Image Processing* **31** (1985) 156-177
11. Tversky, B. Zacks, J., Lee, P. and Heiser, J. Lines, Blobs, Crosses and Arrows: Diagrammatic Communication with Schematic Figures. In *Proceedings of the First International Conference on Theory and Application of Diagrams 2000:* 221-230
12. Tversky, B., Morrison, J., and Betrancourt, M. (2002). Animation: can it facilitate? International Journal of Human-Computer Studies **57 :4** (2002) 247-262
13. Ware, C. Information Visualization: Perception for Design, Morgan Kaufman, San Francisco (2000)
14. Ware, C., and Bobrow, R.. Motion to support rapid interactive queries on Node-Link diagrams. ACM Transactions on Applied Perception **1** (2000) 1-15
15. Ware, C., Neufeld, E., and Bartram, L.. Visualizing Causal Relations. IEEE Information Visualization: Late Breaking Hot Topics (1999) 39-4

# Causal Perception in Virtual Environments

Jean-luc Lugrin[1], Marc Cavazza[1], and Marc Buehner[2]

[1] School of Computing, University of Teesside, TS1 3BA Middlesbrough, United Kingdom
{j-l.Lugrin, m.o.cavazza}@tees.ac.uk
[2] School of Psychology, Cardiff University,CF10 3 AT, Cardiff, UK
BuehnerM@Cardiff.ac.uk

**Abstract.** Causal perception is an important cognitive phenomenon, which plays a central role in our understanding of the world. As such, its study is also relevant to interactive graphics systems. In this paper, we introduce a virtual reality system which can elicit causal perception. The system operates by intercepting the consequences of subjects' actions and modifying them to generate new event co-occurrence. This process is based on an explicit representation of action structure, which supports the generation of event co-occurrences on a principled basis. We present results from a user experiment, which confirm a high level of causal perception in subjects having used the system.

## 1 Introduction

An essential part of interactivity in 3D graphics is about users perceiving consequences of their actions, as well as making sense of object behaviors in the environment. To a large extent, interactivity is deeply rooted in the recognition of causal action. There is ample illustration of this at theoretical level, for instance in the literature on Presence in Virtual Environments (see e.g. [19] ). One simple illustration of this is the extent to which items of Presence questionnaires (such as the Witmer and Singer [16] questionnaire) explicitly refer to action consequences.

Recently, Causal perception has become a research topic for a variety of graphic interface systems, as an understanding of causal perception has implications to develop better visualisation systems [14], animation systems [10], and virtual reality systems [4].

In the field of computer graphics, there has been little specific work on users' causal perception, with the notable exception of research by O'Sullivan and Dingliana [10], who have worked on the relation between causal perception and collision rendering. In the field of cognitive psychology, most research on causal perception has been carried out, as far as experiments are concerned, using simplified environments similar to those used by Michotte [9]. Only Wolff and Zettegren [18] have made extensive use of 3D animations of realistic scenes (however, these were non-interactive).

Our interest is in the study of causal perception in realistic 3D settings, i.e. involving real-world objects rather than symbolic ones. We posit that causal perception plays a major role in the user's physical interpretation of virtual world as it plays in the real world. A better understanding of the determinants of causal perception could

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 50 – 61, 2006.

thus open the way to new VR technology. Our long-term objective is to develop new kind of virtual reality technologies making causality one programmable parameter of the virtual environment. Early experiments in that direction using the technologies described here have been carried out in the field of VR Art [4]. This paper describes the technical approach behind the creation of causality-inducing co-occurrences.

## 2 System Overview

Human subjects show a strong propension to perceive causality between co-occurring events. For instance, the perception of causality from collision events has been qualified by Scholl & Nakayama [12] as "phenomenologically instantaneous, automatic and largely irresistible".

Our system's main objective is to be able to elicit such causal perception within a realistic 3D interactive environment. It should achieve this by creating artificial co-occurrences between user-initiated events and events affecting objects in the environment.

The system presents itself as a desktop virtual environment supporting real-time interaction with a whole range of objects. It is developed on top of the Unreal Tournament 2003™ game engine, which supports visualisation and basic interaction mechanisms [8]. One key feature of our approach has been to take advantage of the discretisation of events in game engines to develop a system in which all events can be intercepted. However, an important feature of our system is that it supports the definition of high-level events, in which action and effect can be properly formalised, hence manipulated by dissociating causes and default effects. These real-time transformations constitute the basis for the generation of co-occurring events that in turn should induce causal perception (Fig. 1).

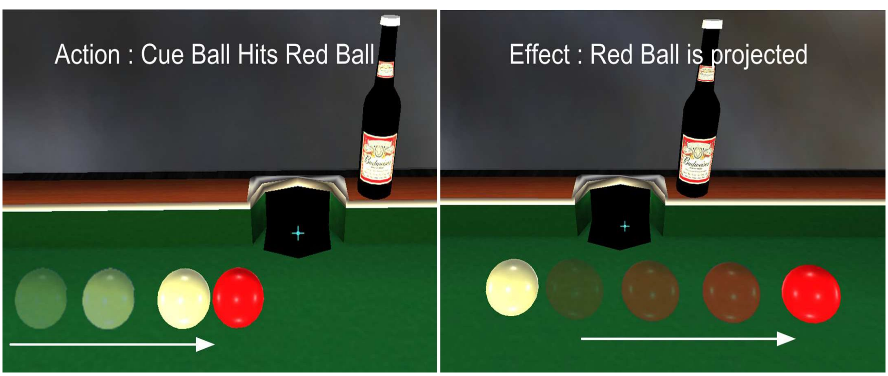



**Fig. 1.** Causal Perception from Co-occurring Events

The main technical innovation of this approach with respect to previous research in causal perception is to make action representation the formalism that supports system behaviour; this enables to experiment with the generation of event co-occurrences on a principled basis, addressing temporal and spatial contiguity as well as semantic properties of objects involved, etc. This description should be supported by an appropriate formalism for change-inducing events, which should clearly identify actions and their consequences. Our representation was inspired from planning formalisms, most specifically the operator representation in the SIPE system [15]. Our formalism is a Cause-Effect representation (CE), where the "cause" part is a formula containing elementary physical events (such as collisions) plus semantic properties of the objects involved, and the "effect" part the transformation undergone by these objects. Figure 2 shows the CE representation for a bouncing action `Bounce-from(?obj, ?surface)`. Its "trigger" part corresponds to the event initiating the action (`Hit(?obj, ?surface)`) and its "effect" part to its effect (that fact that the object bounces back). The "condition" field corresponds to physical properties that have to be met by its objects for the CE to be instantiated.

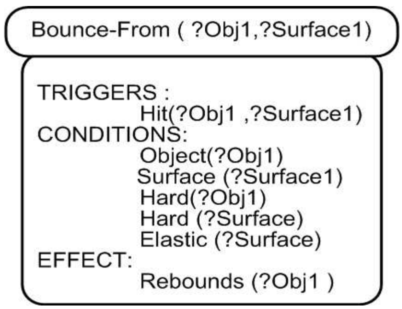



**Fig. 2.** The Cause-and-Effect (CE) Formalism

## 3 Implementation: From System Events to Event Co-occurrences

Many recent graphic engines rely on some kind of event-based system to interpret interactions between world objects or user intervention in the environment [6]. These event systems tend to be derived from low-level graphic primitives which detect collisions between objects, or between objects and volumes, for the purpose of visualisation or elementary behaviour. Game engines, such as Unreal Tournament 2003™ rely on a sophisticated event system to define interactive behaviour. To a large extent, such an event system discretises the Physics of a virtual environment. While motion and trajectories remain under the control of physical (or kinematic) simulations, the event systems permits to interpret certain interactions without having to resort to complex physical calculations. One traditional example consists in determining the consequences of an impact (e.g. a glass panel shattering) by associating consequences to the detection of an impact possessing certain parameters (e.g. missile momentum). The same mechanisms supports interaction with devices for which detailed physical calculations would be irrelevant.

This discretisation of interactions through an event system provides an excellent basis to manipulate the consequences of actions, hence generating artificial co-occurrences that could elicit causal perception. However, as explained previously, we want such generation to take place on a principled basis, rather than by directly scripting pre-defined associations between actions and effects. We have thus defined a software layer dedicated to the parsing of low-level events into the CE formalism we have introduced above. We have termed such a layer an Event Interception System (EIS).

The Unreal Tournament™ (UT) engine constantly produces low-level system events each time an interaction between objects takes place in the graphic world. The EIS thus constantly receives a stream of such events which it "parses" into candidate CE representations corresponding to an ontology of the environment actions. This parsing process involves the recognition of actions applied to objects with specific physical properties. For instance, a "breaking" type event will be recognised from such elements as a high-velocity impact of a solid missile on a fragile, breakable object. These elements are acquired through the interception of low-level events.

More specifically, this event parsing is a bottom-up process involving the following steps (Fig. 3):

- The system constantly receives low-level UT system events, such as `Bump(actor Other)`, `ActorEnteredVolume(actor Other)`, etc. The event also returns an object ID, which can be used to access objects' physical properties. For instance the cue ball hitting the front cushion would generate the following system event: `Bump(#CueBall, #F-cushion )`[1].
- The "trigger" fields of the CE (that determine its activation) have been defined using an ontology of motion events, in order to be generic and not depend on a particular system implementation. The next step thus consists in recognising proper motion primitives (hitting, pushing, touching ) from the system events. This is done through specific filters that define each of the motion events in terms of UT system events and kinetic parameters, such as collision momentum. In the above example, considering the cue ball momentum, this will generate a `Hit(#CueBall, #F-cushion)` event.
- The final step instantiates candidate CE from a list of prototype CEs. For each CE, its "trigger" field is matched to the motion events produced by the previous step. In case of a successful match, the object properties defined in the CE condition field are tested on the triggers' objects. For instance, from a `Hit(#CueBall, #F-cushion)`, a `Bounce-from(?obj, ?surface)` CE can be activated. Its object conditions include `Hard(?obj)` and `elastic(?surface)`, which are matched with `(?obj = #CueBall)` and `(#F-cushion = ?surface)`. The final result of the event parsing is the instantiation of a `Bounce-from (#CueBall, #F-Cushion)` CE, whose effects are suspended as the CE is generated under a "frozen" form to withstand modification.

Upon instantiation of the CE, its "effects" field is also instantiated, from the same objects taking part in the triggers. The result is thus an integrated semantic representation for an action, together with its instantiated effects, ready to be re-activated at the end of the sampling cycle, which would result in the effects being enacted in the 3D world (see Fig 4)

---

[1] We have adopted the following notation: instances are prefixed by "#" and free variables by "?".

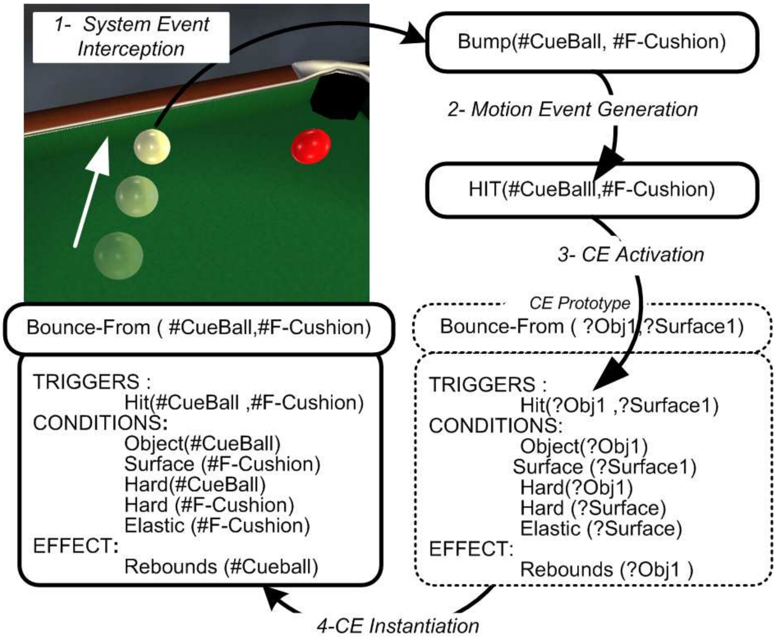



**Fig. 3.** Instantiation of a "Bounce-From" CE

This is where the causal engine comes into action as a mechanism for creating artificial co-occurrences, which should induce causal perception. The principle underlying the causal engine is to modify the "effects" part of a CE representation while it is frozen. Upon reactivation of the CE, this will result in its new (modified) effects being triggered rather than those corresponding to the normal/expected consequence for the initiating action. Hence this creates a co-occurrence between the original event triggering the CE (e.g., the cue ball hitting the cushion) and new events resulting from CE modification (e.g., a bottle on the border falling down onto the pool table). The two main principles for the creation of co-occurrences are the modification of default consequences and the addition of new effects (Fig. 4).

The rules for creating these "alternative" co-occurrences are embedded in a set of so-called macro operators, or MOp. These macro-operators are knowledge structures specifying the type of transformation to be applied to the effect part of a CE. Overall, an effect can be subject to two main types of transformations. The physical event constituting the effect can be modified, e.g. the `rebounds(?ball)` can be transformed into a `shatters(?ball)`. The object itself can be modified, as in `rebounds(?ball)` being replaced by `rebounds(?bottle)`. Sequences of MOp can be applied to a CE (usually up to three only) to obtain a variety of effects. However, co-occurrences can also be created by generating additional effects, as illustrated in our experiments below.

MOp can thus be seen as the elements of a high-level language for specifying the creation of co-occurrences. It is possible to design experiments exploring the role of spatial contiguity for instance, by dynamically selecting MOp affecting objects as a function of the objects' spatial distribution, etc.

The behaviour of the causal engine, in terms of creation of event co-occurrences, can now be illustrated on an example corresponding to the system's behaviour represented in Figure 5. The user strikes the cue ball, which then hits the front cushion of the pool table. Upon hitting the cushion, the bottle that was standing on the pool border falls over the table. The system starts with instantiating a `Bounce-from (#CueBall, #F-Cushion)` CE, from the recognition of the event `Hit(#CueBall, #F-cushion)`. While the `bounce-from` CE was "frozen", a MOp has been selected randomly, which propagates the effects of that CE to neighbouring objects, in that case the bottle `#Bottle`. The default effect is for the ball to move back from the cushion `(rebound(#CueBall)`.The selected MOp will propagate that effect to the nearby bottle. In the case of the bottle, the momentum transmitted will just project it towards the pool table, in a motion which somehow replicates that of the ball bouncing back, obviously creating a different pattern of motion considering the physical properties of the bottle[2].

On average, the whole process of MOp selection and application (CE transformation) is consistently completed in less than 90 ms, giving a system response time of less than 100ms. In the original experiments from Michotte [9], events delayed by more than 150 ms progressively ceased to be perceived as causally linked. Similar values have been provided by Kruschke and Fragassi [7], and in the field of computer graphics, O'Sullivan and Dingliana [10] These data suggest that the system's response time is compatible with results from the psychological literature: as a consequence, the co-occurrences generated should be perceived by the vast majority of subjects as sufficiently temporally close to induce causal perception.

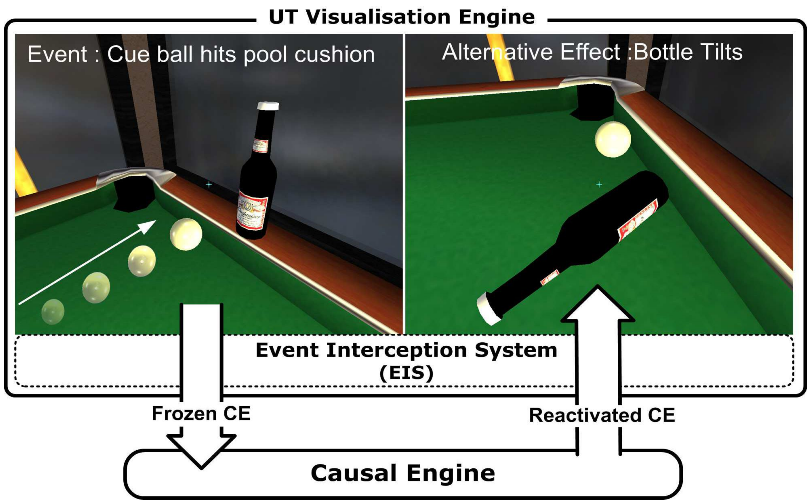



**Fig. 4.** System Architecture and Event Interception

[2] The physical effects of a CE are actually passed back to the native Physics engine of UT 2003 that computes the motion trajectories. This is why the "rebound" of the bottle actually makes it tilt and fall on the pool table.

## 4 User Experiments: Causal Perception from Co-occurrences

In order to evaluate the system capacity to elicit causal perception, we carried an experiment with 24 subjects using the system in a desktop configuration. Subjects were facing an 18-inch screen from a distance of 30-45 cm. The corresponding field of vision in the virtual environment was approximately 80 degrees. The virtual environment consisted in a set of pool tables (an implicit tribute to Michotte, but also a realistic environment in which to study causal perception) each supporting various objects on the front edge of the pool table. These were a lit candle, an empty glass and a bottle. A pool stick was also positioned vertically against one of the sides of the pool table.

The subjects were presented with the following task, which consisted in trying to strike the red ball with the cue ball, taking one rebound on the front cushion. The respective positions of the two balls was modified in the four pool tables so as to force the rebound to take place next to different objects on the cushion, assuming the right aiming angle was taken. Aiming was taking place, like in most computer pool games, by pointing at the cue ball with the mouse pointer. Subjects were allowed to practice this skill on a dedicated table prior to the experiment so as to familiarise themselves with the interface. The force with which the cue ball could be struck was left constant and not controllable by the user, as distances from the balls to the front edge did not vary from table to table.

For each pool table, the impact of the cue ball on the cushion creates a bouncing event (interpreted as a `bounce-from` CE) that is the main target for our causal simulation. This event has an action part, which is the impact and an effect part, which is the new motion of the ball. In this experiment, new effects were propagated to the nearby objects, creating co-occurrences. Each subject has to repeat this task four times, on four different pool table each with its own disposition of the red ball and cue ball, but an identical line-up of objects standing on the border. The position from which they could shoot was also constrained, so that their viewpoint on events would be largely identical across experiments.

The alternative effects that have been generated for the various objects are as follows: the candle can either fall or be blown off, and the glass (and bottle) can either fall or shatter (see Fig 5 ). In addition, the pool stick leaning against the border can fall to the ground. In this experiment, these effects are chosen randomly for a given object, while the object affected is determined as the closest to where the cue ball hit the cushion. Figure 6 shows another example of co-occurrence creation taking place in this experiment: it consists for a glass standing on the pool border to shatter upon impact of the cue ball on the cushion below its position.

For each experiment, the subjects where instructed to give a free text explanation of the events immediately after each try. The instructions given to them did not contain any mention of cause, causality, etc. or what the experiment was about.

The method we decided to use consists in analysing each individual explanation for causal expressions corresponding to linguistic descriptions identified by Wolff [18]. We considered each explanation that included such expressions as a causal explanation, regardless of the number of occurrences of causal expressions in the explanation.

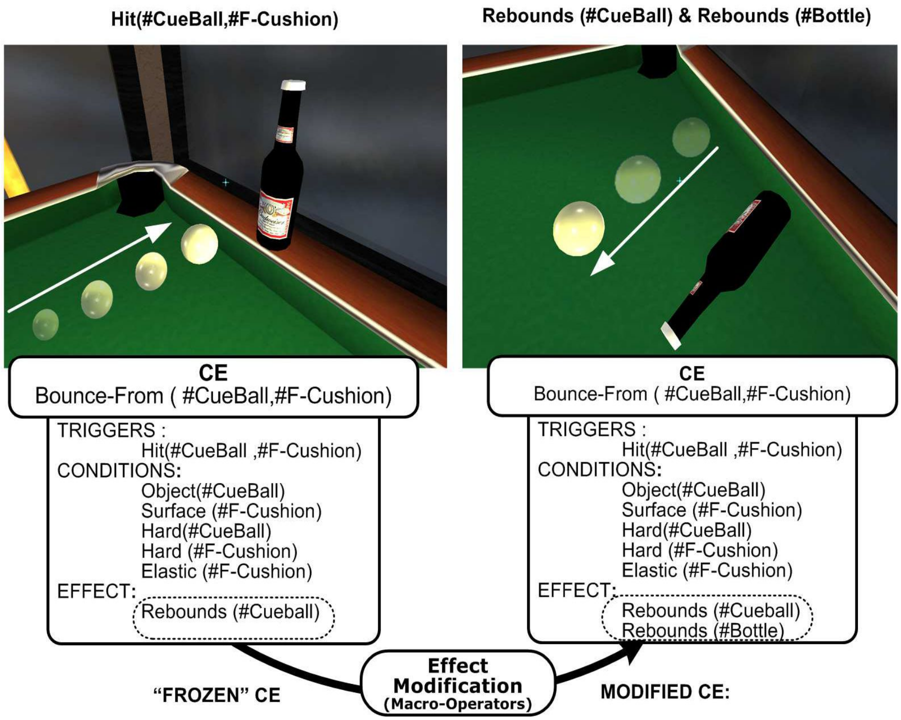



**Fig. 5.** Creation of an artificial co-occurrence through CE modification (Example 1: the ball rebound is propagated to the bottle)

We have applied this method in a rather conservative way, preferring to underestimate causal perception rather than over-estimate it. For instance, we discarded statements such as "the ball hit the glass, the glass broke", which is actually a description of the co-occurrence itself that may or may not intend to convey a causal content.

We have retained the following expressions as causal:

- Use of explicit causal vocabulary ("*causes*", "*causing*", "*caused by*"), as in "*this caused a pool cue to fall over*", "[…] *causing the glass to fall onto the pool*"
- Expressions introduced by "*because*" provided they included an action description (such as "*hit*", "*stroke*", "*bounced*"). In that sense "the bottle fell off because the cue ball is hitting the cushion hard" will be considered as causal, but not "*the glass broke because I aimed at it*"
- Any expression such as "make N V", where N refers to an experiment's object and V stands for a verb indicating motion transfer or change of state (the effect), such as "*fall*", "*move*", "*tilt*" or "*break*", "*shatter*", etc. For instance, "*the force brought by the white ball made the empty glass fall down*"
- Action verbs relating an agent object to a patient one, as in "*the movement created extinguished the flame*", "*the force of the cue ball knocked an object off the side of the table*", etc.

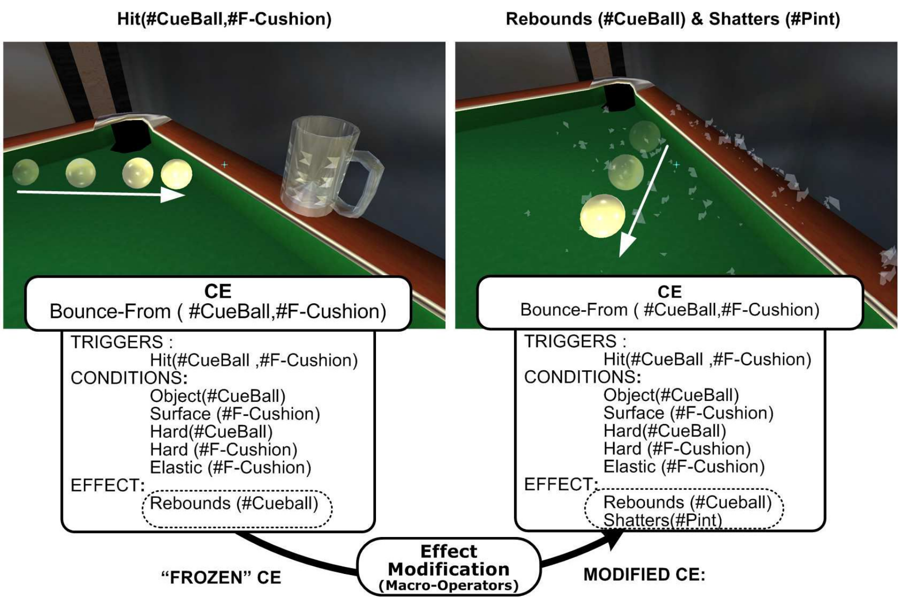



**Fig. 6.** Creation of an artificial co-occurrence through CE modification (Example 2: The glass shatters as the cue ball bounces)

In addition, each subject was asked, after completion of the four trials, to identify the subject of the experiment among: i) Causality ("causes and effects") ii) Physics and iii) Interaction with objects.

Results are presented on Figure 7 and 8. Figure 7 plots the distribution of subjects as a function of the number of causal explanations produced in the course of the experiment. Overall, 71% of subjects have produced two or more causal explanations for the four trials (Fig. 8). This has to be interpreted considering that we have taken a

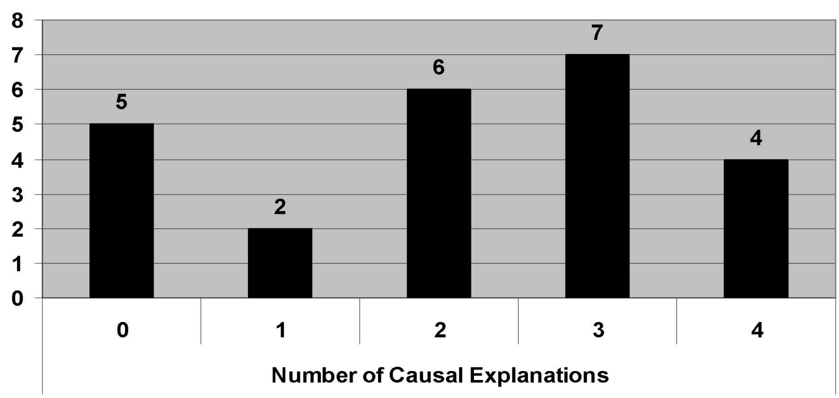



**Fig. 7.** Frequency of causal explanation by subject

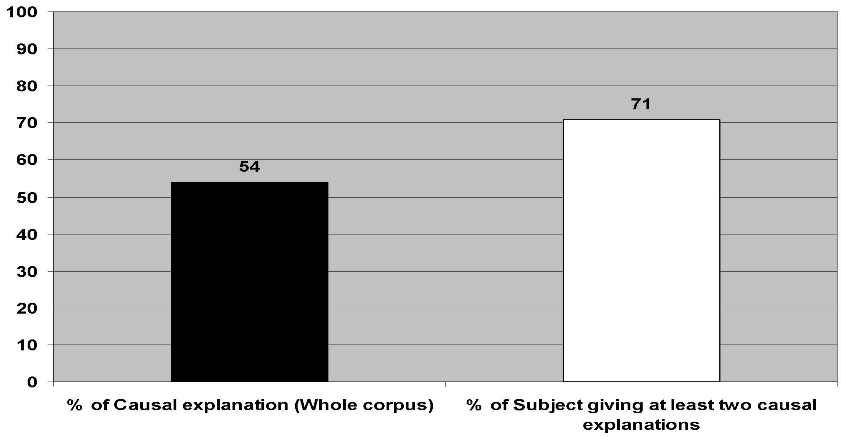



**Fig. 8.** Subject identification of experiment topic

conservative approach to textual interpretation, preferring to under-estimate the number of causal explanations. In terms of identification of the experiment subject, 85% of subjects recognise “causes and effects” as the main subject of this experiment (Fig. 9). These results suggest that the generation of co-occurrences by the causal engine actually induces a high level of causal perception in the test subjects.

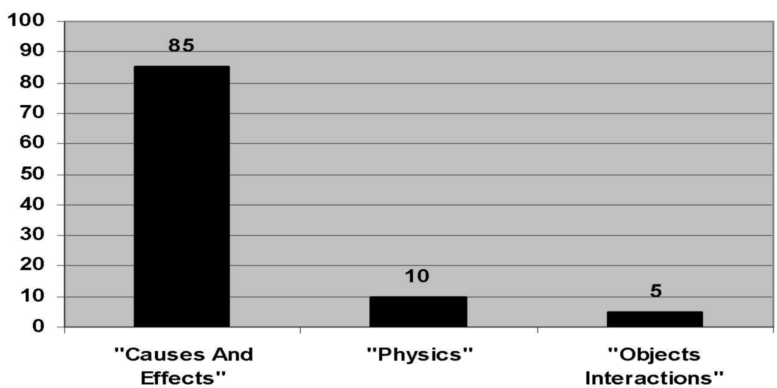



**Fig. 9.** Occurrences of causal descriptions in textual explanations

# 5 Conclusions

Previous research in the psychology of causal perception has shown that temporal contiguity plays a privileged role in human causal induction. While earlier research [13] argued that contiguity is necessary for people to identify causal relations, more recent investigations [1],[2],[3] have shown that this is not the case: people are quite able to appraise non-contiguous causal relations, if they have reason to expect the relation to involve a delay. In contrast to the knowledge-mediation hypothesis [5] cf. [1], however, this research also showed that contiguous relations are often deemed causal, even when participants know that the causal mechanism under investigation involves a delay (see also Schlottmann [11]).

These latter findings suggest that a contiguity-bias may over-ride high-level considerations of causal mechanism. The causality engine presented in this paper facilitates the creation of regularities in an artificial world, and allows the experimenter to control each of the Humean cues (contingency, contiguity) individually. Effect substitution presumably removes the canonical mechanism linking the cause with its effect. The results presented here corroborate proposals for a bottom-up contiguity basis. When two events co-occurred, the sequence had strong causal appeal, even in those cases where there was no plausible causal mechanism linking the events.

The causal engine offers complete control over the evidence perceived by the participant and yet retains a naturalistic and realistic appearance. Effect propagation and substitution are of particular interest, because they allow researchers to present participants with completely novel causal relations. More specifically, events that – according to prior knowledge and preconceived notions of mechanism – are not causally linked can be connected within the engine.

## Acknowledgments

This work was funded in part by the European Commission through the ALTERNE (IST-38575) Project.

## References

1. Buehner, M. J., And May, J.: Knowledge mediates the timeframe of covariation assessment in human causal induction. Thinking and Reasoning, 8(4), (2002) 269-295
2. Buehner, M. J., And May, J.: Abolishing the effect of reinforcement delay on human causal learning. Quarterly Journal of Experimental Psychology, 57B(2), (2004) 179-191
3. Buehner, M.J. And May, J.: Rethinking Temporal Contiguity and the Judgment of Causality: Effects of Prior Knowledge, Experience, and Reinforcement Procedure. Quarterly Journal of Experimental Psychology, 56A(5), (2003 )865-890
4. Cavazza, M. Lugrin, J.L. Crooks, S. Nandi, A. Palmer, M. And Le Renard, M.: Causality and Virtual Reality Art. Fifth International Conference on Creativity and Cognition, Goldsmiths College, London, ACM Press (2005)
5. Einhorn, H. J., And Hogarth, R. M.: Judging probable cause. Psychological Bulletin, 99(1),( 1986) 3-19
6. Jiang, H., Kessler, G.D And Nonnemaker, J.: Demis: a Dynamic Event Model for Interactive Systems. ACM Virtual Reality Software Technology, (2002) ACM Press
7. Kruschke, J. K. And Fragassi, M.. The perception of causality: Feature binding in interacting objects. Proc. of the Eighteenth Annual Conference of the Cognitive Science Society, (1996) 441-446
8. Lewis, M. And Jacobson, J.: Games Engines in Scientific Research. Communications of ACM, Vol. 45, n. 1, (2002) 27-31

9. Michotte, A.: The perception of causality. New York: Basic Books. Translated from the French by T. R. and E.Miles (1963)
10. O'sullivan, C. Dingliana, J.: Collisions and Perception, ACM Transactions on Graphics, Vol. 20, No. 3. (July 2001).
11. Schlottmann, A.: Seeing it happen and knowing how it works: How children understand the relation between perceptual causality and underlying mechanism. Developmental Psychology, 35(5), (1999) 303-317
12. Scholl, B.J., Nakayama, K..: Causal Capture: Contextual Effects on the Perception of Collision Events, Psychological Science, vol. 13, issue. 6, (2002) 493-498(6).
13. Shanks, D. R., Pearson, S. M., & Dickinson, A.: Temporal contiguity and the judgment of causality by human subjects. Quarterly Journal of Experimental Psychology, 41B(2), (1989) 139-159
14. Ware, C., Neufeld, E And Bratram, L.: Visualizing Causal Relations, IEEE Visualization (1999)
15. Wilkins, D. E. : Causal reasoning in planning. Computational Intelligence, vol. 4, no. 4, (1988) 373-380
16. Witmer, B.G. And Singer, M.J.: Measuring presence in virtual environments: A presence questionnaire. Presence: Teleoperators and Virtual Environments, 7, (1998) 225-240.
17. Wolff, P., And Zettergren, M.: A vector model of causal meaning. In Proceedings of the 23rd Annual Conference of the Cognitive Science Society, Hillsdale, NJ: Erlbaum (2002)
18. WOLFF. P.: Direct causation in the linguistic coding and individuation of causal events. Cognition, 88, ( 2003) 1-48.
19. ZAHORIK, P. AND JENISON, R. L. : Presence as Being-in-the-World, Presence: Teleoperators and Virtual Environments, 7, (1998) 78-89.

# Deep Surrender: Musically Controlled Responsive Video

Robyn Taylor[1] and Pierre Boulanger[2]

[1] Advanced Man-Machine Interface Laboratory,
Department of Computing Science, University of Alberta
T6G 2E8 Edmonton, Alberta. Canada
robyn@cs.ualberta.ca
[2] pierreb@cs.ualberta.ca

**Abstract.** In this paper we describe our responsive video performance, *Deep Surrender*, created using Cycling '74's Max/MSP and Jitter packages. Video parameters are manipulated in real-time, using chroma-keying and colour balance modification techniques to visualize the keyboard playing and vocal timbre of a live performer. We present the musical feature extraction process used to create a control system for the production, describe the mapping between audio and visual parameters, and discuss the artistic motivations behind the piece.

## 1 Introduction

We have used Cycling '74's Max/MSP and Jitter packages [1] to create a visualization environment responsive to vocal timbre and piano chord manipulation. This visualization system was used to create an interactive performance called *Deep Surrender*, a multimedia piece written for soprano, synthesizer, and responsive video (see Figure 1.)

Data from live musical performance is extracted and used to control parameters in video processing routines. The video manipulation is, therefore, responsive to the musician's performance, allowing the media piece to be dynamic and expressive of the nuances of live performance. This project creates an interaction platform that a musical performer can use to manipulate visual imagery in a fluid and natural way.

Responsive artistic spaces have been created by Ox [4] who visualized harmonic relationships in musical pieces through the use of colour, and Oliver *et al.*[3] who interpreted sung vocalizations through responsive audio-visual feedback. Our system uses similar strategies to map chord data and vocal information to responsive imagery in order to create a visualization environment that enhances our specially composed audio-visual performance piece.

This document outlines the process used to extract musical feature data from a live musical performance, the mapping between musical features and visual elements in the performance environment, and the artistic concept behind the *Deep Surrender* production.

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 62–69, 2006.

**Fig. 1.** A performance of *Deep Surrender*

## 2 Musical Feature Extraction

Our musical feature extraction routines are encapsulated into a Max/MSP patch called the Musical Perception Filter Layer. The patch analyzes a live musical performance in real-time and communicates vocal pitch, amplitude and timbral information as well as information about chords played on a digital piano. We developed this patch as part of our previous work, where extracted musical data was used to control the behaviour of responsive avatars [7] [8].

Inside the Musical Perception Filter Layer, MIDI events are monitored in order to identify the chords that are played on the keyboard. Vocal pitch and amplitude information is interpreted from sung input using the functionality provided by the `fiddle~` object created by Puckette *et al.*[5]. The `fiddle~` object also outputs the raw frequency and amplitude data describing each of the partials forming the harmonic spectrum of the user's singing voice. Examining the amplitude of the energy found at each of these partial frequencies allows us to view a measure of a singer's timbre.

In our production, we use the extracted piano chord data and information about the peformer's vocal timbre to modify the colours in the *Deep Surrender* video stream.

## 3 Mapping Vocal Timbre to Colour

A sung sound consists of a *fundamental frequency* and a series of *harmonics* or *partials* that exist at multiples of the fundamental frequency. In the *Deep Surrender* visualization, vocal timbre is illustrated by visually representing the distribution of energy amongst partials in the sound, thereby creating a parameter that a singer controls by modulating her vocal timbre.

We have chosen to map the first three amplitudes output by the `fiddle~` object (the amplitudes of the fundamental and the first two partial frequencies) to the inputs of a Max colour-chooser object. This object takes as its input parameters red, green, and blue colour component values. The amplitude of the fundamental frequency is mapped to the red colour component, while the second and third partials are mapped to the green and blue components respectively. This mapping results in the generation of an RGB colour value whose hue is dependent upon the weighting of tone amplitude amongst the partial frequencies found in the singer's vocal output.

Assigning colour in this way yields predictable and repeatable results in response to vocalization. Different vowel sounds produce different colours, as the differences in the formant structures of each vowel sound produce different amplitude weightings at the partial frequencies.

If a soprano sings the closed vowels /i:/ as in 'free' or /u:/ as in 'fool' on a mid-range note (such as A4 which has a frequency of 440 Hz), the vowel's formant structure is such that it closely resembles with the fundamental frequency of the singer's pitch. In these cases, tone amplitude is high at the fundamental and the resulting colour output is in the red-to-yellow range. If she sings the open vowel /a:/ as in 'car', the vowel's formant structure sits higher than the phonated pitch, resulting in increased amplitude at the second and third partials in the harmonic spectrum. Our system's colour output is then in the green-blue range.

Vocalization of any vowel at a high pitch (soprano high C and above) allows the singer to produce intense red/orange colours. This occurs because extremely high pitches are phonated at frequencies higher than the vowel characterizing formant frequencies. Research in vocal production conducted by Sundberg [6] indicates that at high pitches, soprano singers modify the vowel's formant frequency so that it rises to approximately equal the value of the fundamental frequency. This results in a high weighting of tone amplitude at the fundamental, causing our system to display a red/orange colour.

Using this mapping strategy, extreme sounds (the high pitched notes or closed vowels) can be characterized with the vibrant reds and oranges, and less dramatic sounds (the open /a:/ vowel sung at moderate pitches) with the more restful blues and greens. This colour dynamic is used as a way to visualize the intensity of a singer's vocalization.

The piece is composed with this mapping between timbre and colour in mind. The early sections are sung on open vowels in the singer's mid-range, producing blue-green visual imagery, while the more dramatic later passages require the soprano to exercise the extremes of her vocal range (from high C to a high F) in order to illustrate intensity through vibrant colour.

## 4 Visualizing Chord Relationships Through Colour

The Circle of Fifths (see Figure 2) is a music-theoretical device that geometrically represents the way the twelve major and twelve minor key signatures relate to one another [2]. Each key signature is rooted by one of the twelve notes of the chromatic scale. If the Circle is traversed in a clockwise manner, each subsequent step to the right on the Circle represents an interval of a Perfect Fifth.

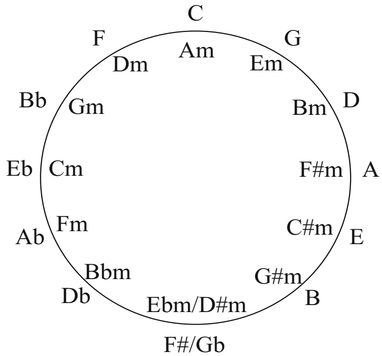



**Fig. 2.** The Circle of Fifths

The placement of chords on the Circle of Fifths indicates the similarity between the chords. C major and A minor appear in the same position on the Circle since A minor is the relative minor of C major. Both C major and A minor are key signatures that contain no sharps or flats. E minor is the relative minor of G major (both E minor and G major contain one sharp) and so on.

In this application the key signature relationships contained in the Circle of Fifths are visualized by mapping the Circle to the colour wheel. Chords that are adjacent on the Circle will have similar colour values.

The parallel between the circular structure of the Circle of Fifths and the colour wheel has been exploited in previous music visualization works. Jack Ox used a similar mapping in her *Color Organ* visualization [4]. Ox's work used the colour-associated Circle of Fifths to visualize the existing chord relationships in musical pieces. To represent harmonic structure in music, Ox mapped a visual colour wheel to the Circle of Fifths. Each chord present in the musical input triggered a colour representing the chord's position mapped on the colour wheel.

The mapping between the colour wheel and the Circle of Fifths was used for a subtly different purpose during the compositional process of creating *Deep Surrender*. Since the mapping between chords and colours was fixed, the colours and chords could be treated synonymously. The music was composed visually by selecting colours on the wheel that the composer felt would help illustrate the mood of each section of the piece. The sound of the song was determined by choosing between the relative major and minor chords found at the selected colour locations.

## 5 The Video Processing Effects

The unprocessed video footage used in this production contains images of white jellyfish on plain blue backgrounds. We use chroma-keying and colour balance modification to adjust the appearance of these video clips in response to the musician's performance.

Chroma-keying (commonly known as blue-screening or green-screening) refers to the process of isolating objects from one image source and inserting them into a different image source, making them appear to be part of a new scene. Jitter provides chroma-keying functionality through the `jit.chromakey` object.

Colour balance modification causes the entire hue of a video stream to be changed. Jitter allows the colour balance of video streams to be modified through the `jit.scalebias` object.

In the *Deep Surrender* project, we use the colours produced by the chords played on the keyboard to affect the overall colour balance of the video playback. The singer's vocalizations are used to affect the colour balance of the imagery that is chroma-keyed into the video.

Chroma-keying and colour balance modification are used in tandem to create the visual effect that certain jellyfish within the environment are changing colour while the environment remains unchanged. This is done by compositing colour-adjusted video streams in a multi-step process:

– The video stream containing the original jellyfish image is duplicated into a second Jitter matrix.
– The colour balance of the duplicate stream is altered to the colour determined by the singer's vocalization.
– The duplicated (colour-altered) jellyfish image is chroma-keyed into the original stream, overlaying the original jellyfish.

The resulting video stream then contains a colour-manipulated jellyfish while the rest of the image remains unchanged.

## 6 The *Deep Surrender* Performance

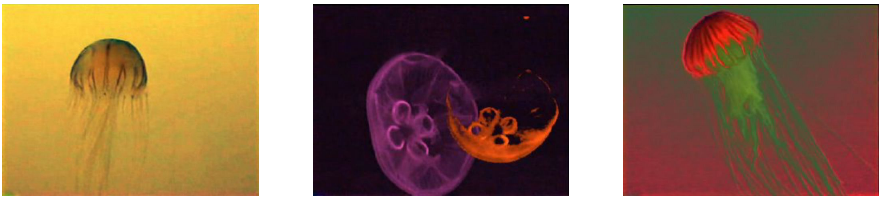
The intention of the piece is to illustrate how an artist can harness anxiety and adrenalin to produce a beautiful performance. This is achieved by utilising the visual metaphor of a jellyfish – a creature both beautiful and terrifying. The artist's musical performance manipulates the jellyfish representation in order to convey how the artist interacts with and overcomes her anxiety.

The *Deep Surrender* performance uses the video effects described in the previous section of this paper to respond to a live performer's vocalization and keyboard playing by manipulating aspects of the video footage.

In each section of the performance, different videos are played and different video processing strategies are controlled by the performer's musical input. Her interaction with the visualization is intended to illustrate her emotional state.

### 6.1 Part One

In Part One, the performer is merely an observer of the fearful environment. The visualization is simple, with the artist controlling only the colour of the image by playing simple chord progressions.



(a) Part 1 (b) Part 2 (c) Part 3

**Fig. 3.** Images from the Deep Surrender Performance

- The bold colours in this section are controlled through the `jit.scalebias` object which is triggered by keyboard input.

In Figure 3a, we see the results of the performer playing a G major chord on the keyboard. The G major chord corresponds to the orange-yellow colour tone according to our mapping between the Circle of Fifths and the colour wheel.

### 6.2 Part Two

In Part Two, the performer begins to experiment. She uses her voice to introduce entities to the visual environment and realizes that her actions elicit a colourful response from the ominous creatures.

- The composition modulates to an eerie set of chords which correspond to icy colour tones in the colour spectrum (purples, pinks, blues.)
- A dark and sinister video sequence begins.
- The colour balance is still modified by keyboard input.
- The performer's voice brings new and vibrantly coloured entities into the environment, using the `jit.chromakey` functionality.
- The performer's voice is used to offset the composed colour choices, making this section visually dynamic.

In Figure 3b, we see the orange jellyfish introduced to the scene as a result of the performer's vocalization. The jellyfish is orange because she is using a focused tone on a closed vowel (/i:/) which corresponds to an orange colour.

### 6.3 Part Three

In Part Three, the performer realizes that the most beautiful results are produced when she combines her voice with the fearful imagery. The visual effects flourish through her willingness to overcome her hesitation and thrive upon the adrenalin rush of fear.

- The song returns to the original theme, and the first video is replayed.
- Additional sound layers join the synthesized accompaniment, helping to convey new energy in this section.

– Using her voice as input, the performer can modify the video playback (through chroma-keying applied to duplicate image streams) to produce the piece's most vivid colour effects.

In Figure 3c, the performer sings extended high pitches (soprano high C and above) in order to superimpose the vivid red tones upon the scene. The underlying colour balance is controlled by the keyboard (the green colour is produced by the A chord) and the extreme vocalizations affect the highlighted portions of the jellyfish, yielding an intense visual result.

## 7 Discussion

A traditional musical performance conveys emotive cues to the audience through the musical structure of the composition and through the performer's nuances and body language. In addition to these traditional methods of communication, the multimedia performance environment we have created allows visual information to be presented to the audience. The content of the video streams and the visual responsivity, controlled by the performer, augment the performance and assist in communicating the concept of the musical piece.

The interaction techniques used in this application allow the performer to guide the visualization in an intuitive fashion. She need only play the piano and sing in order to manipulate the visualization, rather than function as a computer operator during the performance.

During the performance, the singer observes the visualization and adjusts her vocalizations (by modifing her vowel formation or pitch) in order to achieve the most pleasing visual results. Since the visualization of vocal timbre responds to vocal subtlety, the slight variations in vocal production that occur each time the singer performs the piece make each repetition of the visualization unique.

The *Deep Surrender* production has been performed several times in concert settings. Additionally, it is often performed during tour sessions of the Advanced Man-Machine Interface Laboratory in order to show visitors how visualization technologies can be used for artistic purposes.

## Acknowledgments

The jellyfish video footage used in the *Deep Surrender* production was filmed by Melanie Gall.

## References

1. Cycling '74. Max/MSP and Jitter.
2. Krumhansl, C.L. The Geometry of Musical Structure: A Brief Introduction and History. Computers in Entertainment, Volume 3, Issue 4, pages 1-14. ACM Press, 2005.

3. Oliver, W., Yu, J. and Metois, E. The Singing Tree: Design of an Interactive Musical Interface. In DIS'97: Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods and Techniques, pages 261-264. ACM Press, 1997.
4. Ox, J. Two Performances in the 21st Century Virtual Color Organ. In Proceedings of the Fourth Conference on Creativity and Cognition, pages 20-24. ACM Press, 2002.
5. Puckette, M., Apel, T. and Zicarelli, D. Real-Time Audio Analysis Tools for Pd and MSP. In Proceedings of the International Computer Music Conference, pages 109-112. International Computer Music Association, 1998.
6. Sundberg, J. The Science of the Singing Voice. Northern Illinois University Press, 1987.
7. Taylor, R., Boulanger, P. and Torres, D. Visualizing Emotion in Musical Performance Using a Virtual Character. In Proceedings of the Fifth International Symposium on Smart Graphics, pages 13-24. Springer LNCS, 2005.
8. Taylor, R., Torres, D. and Boulanger, P. Using Music to Interact with a Virtual Character. In Proceedings of New Interfaces for Musical Expression, pages 220-223, 2005.

# Hierarchical-Temporal Data Visualization Using a Tree-Ring Metaphor*

Roberto Therón

Departamento de Informática y Automática,
Universidad de Salamanca, Salamanca, 37008, Spain
theron@usal.es

**Abstract.** This paper describes a novel and efficient visualization technique intended for hierarchical-temporal data using a tree-ring like layout. Temporal hierarchies appear in numerous fields such as genealogy, evolution taxonomies or time lines. In many cases, state-of-the-art static diagrams are produced in these fields. By using several information visualization strategies, such as focus + context, the tree-ring approach has the ability to visualize and navigate these, potentially complex, hierarchies trough time. Thus, a deeper insight into the problem at hand can be gained.

**Keywords:** tree-ring layout, information visualization, focus + context.

## 1 Introduction

In the last few years the visualization of hierarchical data has been a subject of great interest in information visualization research. Hierarchies are often represented as trees, i.e., a special kind of graph. Graphs are the fundamental structural representation of structured data [1], so the hierarchy visualization problem becomes a part of the graph visualization problem.

Hierarchies are used in many areas of application, including computer file systems, taxonomies, genealogy, phylogenesis, evolutionary trees, etc. The information visualization community has explored many different approaches, focusing on navigation and interaction, that help to overcome some of the limitations present in graph drawing. For the latter, the book of Battista *et al.*[2] covers all the key aspects, while Herman *et al.* provide a thorough survey of information visualization applied to graph visualization and interaction[1].

Although the visualization of hierarchies has been extensively studied in the literature, very few works related to the visualization and exploration of hierarchical data sets that represent a temporal evolution have been done. Furthermore, in many situations visualizations should emphasize temporal relationships and patterns. This paper presents Treevolution, a novel visualization tool that

* This work was supported by the MCyT of Spain under Integrated Action (Spain-France) HF2004-0277 and by the Junta de Castilla y León under project SA042/02. The author would like to acknowledge Antonio Hernández Serrano for his assistance in the implementation of this work.

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 70–81, 2006.

uses a tree-ring metaphor in combination with focus + context techniques, that can be useful for hierarchical-temporal data visualization.

The rest of this paper is organized as follows: next, a review of works related to hierarchical data and time visualization is presented. In Section 2 the tree-ring metaphor is explained. The third section is devoted to a case study: the visualization of the computer language history using Treevolution. To finalize, the main conclusions and future work are described.

### 1.1 Related Work

Many display layout techniques have been developed for the visualization of hierarchical data sets like Cone tree, Treemap[3], Hyperbolic tree[4], etc. Several works have dealt with focus + context techniques for visualizing and manipulating large hierarchies. Yee *et al.* [5] use the well known radial tree layout method[6][2], in which the focused node is placed in center of display, and all other nodes are rendered on appropriate circular level around that selected focused node. Also this approach makes use of different animation techniques for supporting interactive exploration of the graph. In [7] a software framework for creating dynamic visualizations of both structured and unstructured data was developed.

An important aspect is how the process of determining the position and size of each visual object that is displayed in a presentation can automated. Effective layout is one of the most important aspects of creating an information presentation, because that presentation it is intended to be viewed and manipulated by people. A survey of automated layout techniques for information presentations can be found in [8].

In [9] focus + context techniques were used to compare the structure of large phylogenetic trees. Munzner *et al.* proposed a new rectilinear focus + context technique for navigation that is well suited to the dynamic linking of side-by-side views while guaranteeing landmark visibility and constant frame rates.

Recently, Morris *et al.*[10] have dealt with the visualization of temporal hierarchies. In their work, documents from a research front are plotted by time along a horizontal track in the time line, with related research fronts being plotted in nearby tracks according to the hierarchical structure produced during clustering.

In [11] tree-ring metaphors are introduced to enhance information content of focus areas in large time-line visualizations of objects that split or merge. The tree-ring representation shows the relative timing of splits and merges of each object. In the tree-ring metaphor for splits and merges, radial distance indicates time with the birth at the center. At a given radial distance, pieces of arc represent descendant nodes in existence at that time. A node that neither splits nor merges is represented by a circle whose radius indicates the lifetime of the node.

Finally, we would like to point out that Treevolution was developed taking into account the Computational Information Design (CID) process proposed by Benjamin Fry [12]. CID brings together design, information, and computation with a focus on how they support one another as parts of a combined methodology for the exploration, analysis, and representation of complex data, and processing, his Java-based software development environment aimed to simplify the construction of graphically-oriented software.

## 2 The Tree-Ring Metaphor

As stated above, Treevolution uses a tree-ring metaphor for the temporal hierarchy layout. The following definitions will be useful for the rest of the paper[1]:

**Fig. 1.** Cross section of a *Pseudotsuga menziesii* showing almost perfect tree-rings

- ***Dendrochronology*** can be defined as *The science that uses tree rings dated to their exact year of formation to analyze temporal and spatial patterns of processes in the physical and cultural sciences.*
- ***tree ring:*** A layer of wood cells produced by a tree or shrub in one year, usually consisting of thin-walled cells formed early in the growing season (called *earlywood*) and thicker-walled cells produced later in the growing season (called *latewood*). The beginning of earlywood formation and the

[1] The source of these elementary definitions, as well as figure 1, is the comprehensive collection of information related to tree-rings available in Henri D. Grissino-Mayer's Ultimate Tree-Ring Web Pages, http://web.utk.edu/ grissino/.

end of the latewood formation form one annual ring, which usually extends around the entire circumference of the tree.

– ***tree-ring chronology:*** A series of measured tree-ring properties, such as tree-ring width or maximum latewood density, that has been converted to dimensionless indices through the process of standardization. A tree-ring chronology therefore represents departures of growth for any one year compared to average growth. For example, an index of 0.75 (or 75) for a given year indicates growth below normal (indicated by 1.00, or 100).

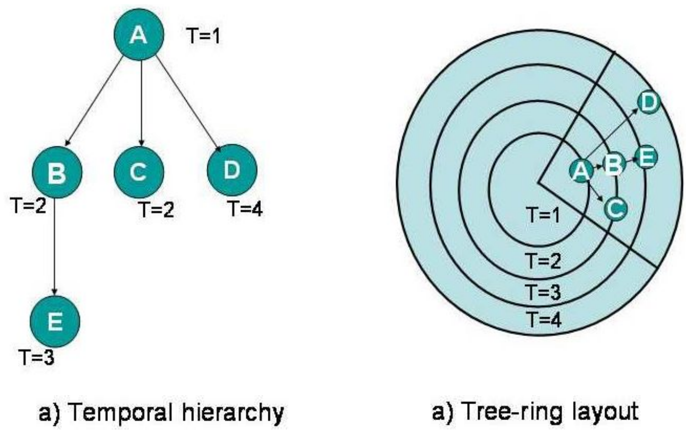



**Fig. 2.** The tree-ring metaphor

Hence, the idea behind the tree-ring metaphor is to provide a way to visualize both timing and structure in a single diagram. In Figure 1, a cross section of a tree can be seen. In it, each tree-ring shows the annual growth; the with of the ring depends on how productive this year has been for the tree. In order to convey hierarchical information, while keeping the time information, a modification of the radial layout [6] can be used: instead of placing nodes on concentric circles according to their depth in the tree, they are placed according to its temporal information.

Figure 2 shows a simple hierarchy with temporal information. In the tree-ring layout, node E is placed in the fourth circle because its time is $t = 4$; in this case the fact that it is a direct descendant of the root node (A) is not important for node placement. The hierarchy information is kept in the directed arc (from A to E), though.

Also note that the hierarchy has been drawn occupying a circular sector. The free space could be used, in the case of having a more complex hierarchy, by placing each subtree or family in an individual sector. This concept can be seen in Figure 3. The picture shows how Treevolution, by using the tree-ring layout, visualizes several subtrees of a complex temporal hierarchy.

**Fig. 3.** Default layout of a temporal hierarchy in Treevolution

In the following sections we will see how this simple idea can be further exploited by using different information visualization techniques.

## 3 Case Study: Browsing the History of Computer Languages with Treevolution

In order to describe the benefits of the Treevolution technique, a computer languages history diagram[2] will be studied.

Figure 4 shows the evolution of computer languages from 1957, with the advent of FORTRAN, to the current releases of popular languages such as Java or C#. Actually, the picture only shows a small part of the history, from 1957 to 1989, because the diagram is intended to be either horizontally scrolled with

[2] Diagram produced by Eric Levenez, http://www.levenez.com.

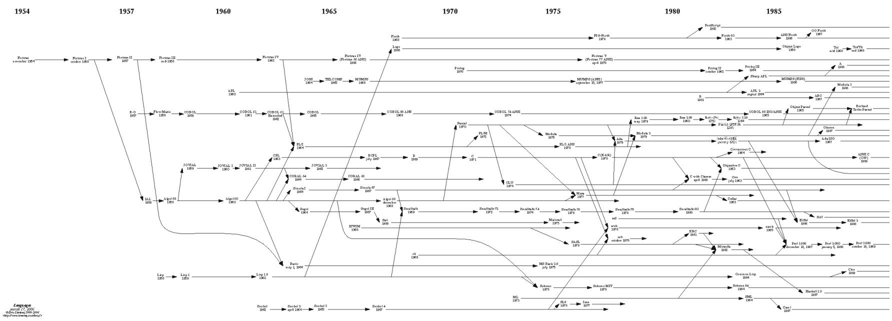



**Fig. 4.** Partial diagram (1957-1989) of computer languages history

an Internet browser or to be exhibited on a wall. For our purposes, this piece of history is enough to understand that each family/tree runs from left to right, as times does. With this diagram it is very difficult to form a mental map of the evolution, since all connections between fathers and sons are mixed.

At this point, it is important to understand that the presence of a temporal component changes the types of analysis that may be done [13]. Traditional approaches for the representation of hierarchical data are not valid for faithfully preserve the information and knowledge content of the data, and prevent the viewer from understanding complex hierarchical-temporal relationships.

A user analyzing the complexity of the history of computer languages would need to answer questions like the following: How many languages have been developed? How many different families are there? Which one is the most fruitful? Is there a particular time where the evolution was faster? How many ancestors does one particular language have? How many descendants? Which is the language that had the biggest number of descendants in the smallest period of time?

Some of these questions can be answered after a tedious and meticulous analysis of the Figure 4 diagram. However, it is impossible to have an overview of the full history. The analyst would need to start from left to right, take some notes, and go on scrolling toward the present time. This means that important questions cannot be visually answered.

With Treevoltion the whole history (1957-2006) can be seen in a single and compact diagram. Actually, Figure 3 showed the history of computer languages in Treevolution. The same hierarchy, but this time all nodes are labeled, is shown in figure 5. Note that light red edges depict the ancestors of nodes pertaining to other families.

Furthermore, the Levenez diagram suffers from the well-known problem of focus: as we scroll when we are following a particular time line, all contextual information is lost. Some tasks, common in the analysis of any hierarchical-temporal data set, where the context is needed are:

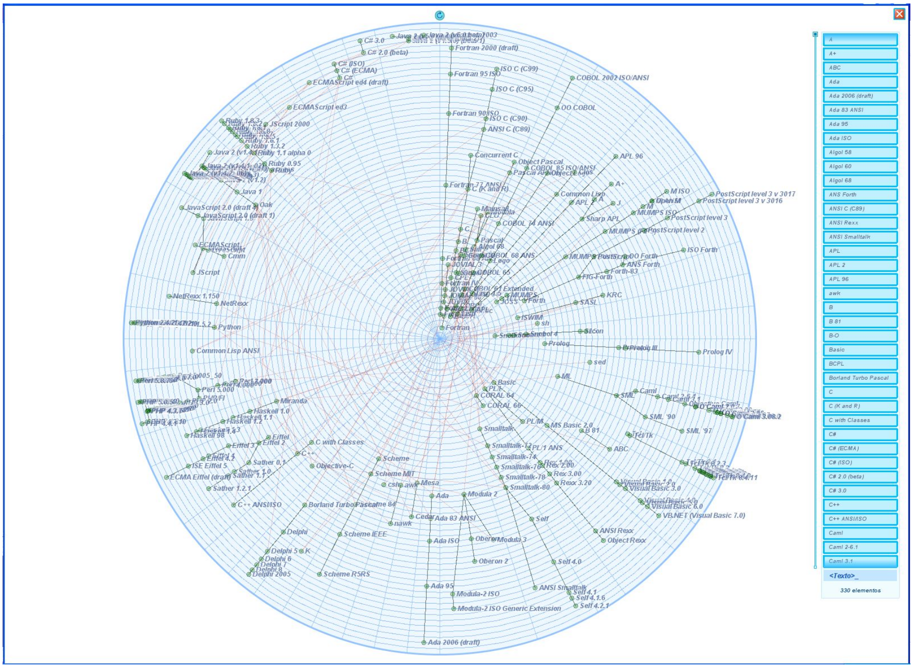


**Fig. 5.** Computer languages history, from 1957 to 2006, in Treevolution

- Have a clear overview of the whole evolution.
- Focus on a family or a subfamily while keeping its relationships with other families.
- Compare the evolution of two or more families.
- Identify a subfamily within a family.
- Focus on a particular period of time across families.

The trade-off between global views and visibility of details is a long-standing problem in visualizations of large data sets [1]. Traditional distortion techniques can be used to focus on detail in particular areas while keeping the context. In addition to the visualization technique, for an effective data exploration, it is necessary to use some interaction and distortion techniques. Interaction techniques allow the data analyst to directly interact with the visualizations and dynamically change the visualizations according to the exploration objectives[14]. This approach has been followed in two ways in Treevolution:

- **Sector distortion.** The angle of any sector can be interactively opened in order to have more space for the layout of a particular subtree (or computer language family in this example). As one sector is opened, the remaining sectors are proportionally closed. This can be seen in Figure 6, where the PostScript sector (upper left quadrant) has been opened by the user in order to clearly see its hierarchy.

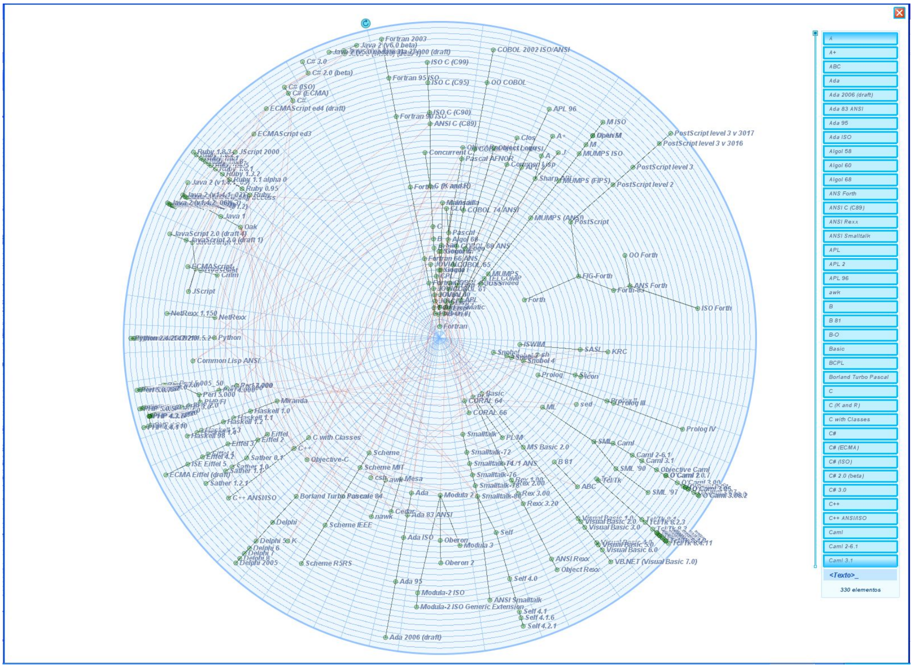


**Fig. 6.** The user is focused on the Postcript family (upper left quadrant)

– **Ring distortion.** The width of any ring can be interactively increased in order to view more clearly that period of time. At this point, the tree-ring metaphor offers its best, since all rings cover the same amount of time, but the number of nodes within that period is variable (different growth index, in tree-ring chronology terms). This situation can be seen in Figure 7, where years 2002 and 2003 were very productive, with several versions. This way, although all rings represent a year time, the width of the rings has been increased in order to clearly view the rapid evolution of Java during that years. Note that inner years have their width decreased in order to make room for the productive years.

Furthermore, Treevolution can automatically weight the width of each family and the productivity of each year, so each sector and each ring is automatically distorted according to those weights. Thus, the resulting tree-ring layout can easily convey this complex information, that more traditional approaches cannot deal with (this is the case of the diagram in Figure 4, which is manually drawn by the author without using any automatic tool).

Besides, some more interactions [14] have been added in the Treevolution implementation. On the right hand size, an alphabetical browser of node labels permits to find a particular node within a complex hierarchy. Also, it is possible to filter the label browser with a word provided from the keyboard. This is what it was done in Figure 7: Oak was typed so the node was selected. As a result, the

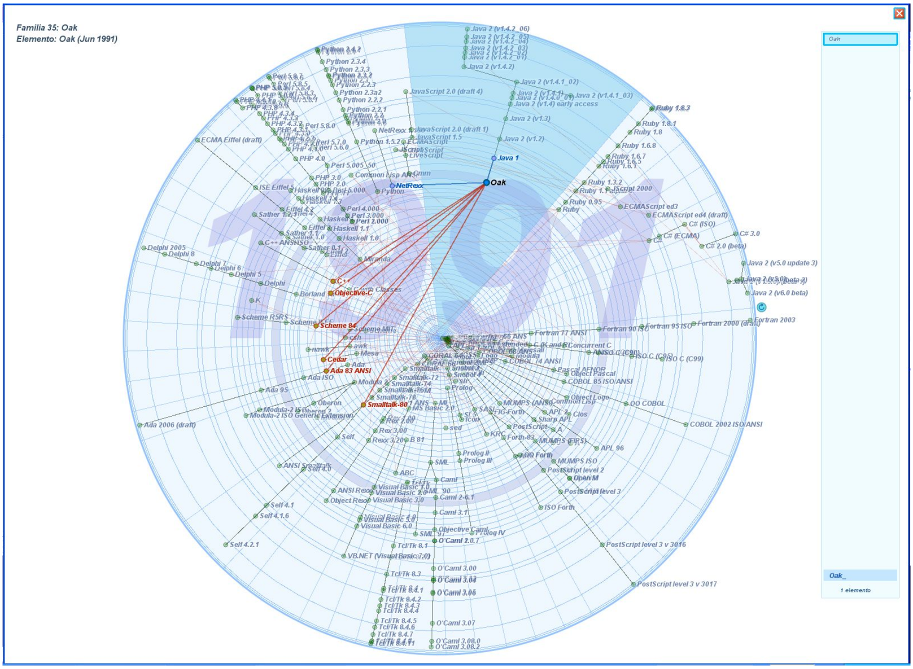



**Fig. 7.** Full details of the evolution of the Java hierarchy

node is highlighted, its direct descendant is highlighted in blue, and its ancestors are highlighted in red. The oak sector (family) and the ring (year) are shadowed and the particular time is shown in the background (1991).

Figure 8 shows a similar situation but for the C language (Kernighan and Ritchie) in 1977. Note that a subfamily, C (K and R) is highlighted with a darker shadow than the whole family (Fortran). Also note that the Smalltalk family (in the bottom left quadrant) has a wider sector. This is the way Treevolution can maintain several foci while maintaining the context.

Another important feature of Treevolution is that the whole diagram can be rotated, in order to provide a way of avoiding the label cluttering that the automatic layout can produce. This a very important issue, since the design of an aesthetic label layout is an efficient way to increase the functionality (speeding up search tasks) of technical and scientific illustration [15]. The user only needs to drag the small round button that is located outside the circle. Note the different locations of this button in figures 5 (default location), 6,7 and 8.

Finally, it should be noted that the visual engagement evoked by the natural tree is also due to its irregularities. These irregularities have not been taken into account for the current implementation of Treevolution. However, the variations in the ring width for different radios (differential growing) might be useful to display subtree features, and the variations in the colors (from heartwood to outer wood) might be useful to bring more clearly to mind the tree-ring outward-growing metaphor). This will be subject of further research.

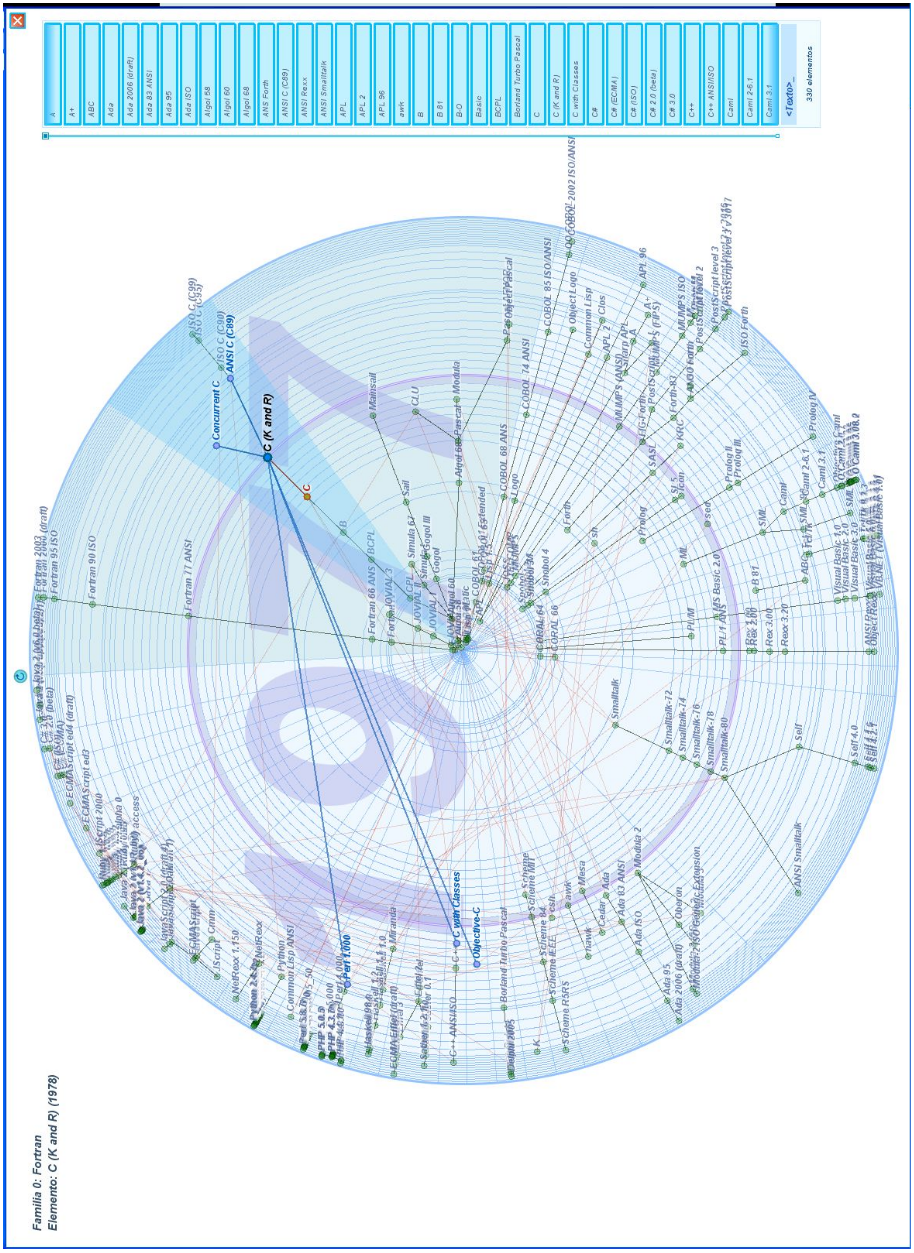



**Fig. 8.** Treevolution focusing on 1977 (Kernighan and Ritchie C family)

## 4 Conclusions

A novel method for the visualization and navigation of hierarchical-temporal data was presented. The tree-ring metaphor provides an elegant solution to this particular problem, while the focus + context and interaction techniques permit the expert to see and understand a big and complex dataset at once. The rotation of the tree-ring is a simple but very efficient way of label uncluttering. As case study, Treevolution was successfully applied to the popular history of computer languages diagram, showing its potential for exploring hierarchies and helping the user to gain a deeper understanding of the evolution and relationships among the different elements in the hierarchy.

Future extensions will be providing multiple linked views of selected subtrees, fisheye view for rings, node filtering depending on a period of time, and test the technique with larger datasets. Also, the optimization of the layout algorithm, so the number of crossing edges can be minimized, will be studied.

Another important aspect that will be tested is the user performance (for instance, how the user judgments are affected by the orientation and order of the particular sequence around the circle).

Finally, a challenging issue is how to deal with trees whose structure changes over time, rather than just adding more nodes.

## References

1. Herman, Melançon, G., Marshall, M.S.: Graph visualization and navigation in information visualization: A survey. IEEE Transactions on Visualization and Computer Graphics **6** (2000) 24–43
2. Battista, G.D., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Prentince Hall, Upper Saddle River, N. J. (1999)
3. Shneiderman, B.: The eyes have it: A task by data type taxonomy for information visualizations. In: IEEE Visual Languages. Number UMCP-CSD CS-TR-3665, College Park, Maryland 20742, U.S.A. (1996) 336–343
4. Munzner, T., Burchard, P.: Visualizing the structure of the world wide web in 3d hyperbolic space. In: VRML '95: Proceedings of the first symposium on Virtual reality modeling language, New York, NY, USA, ACM Press (1995) 33–38
5. Yee, K.P., Fisher, D., Dhamija, R., Hearst, M.A.: Animated exploration of dynamic graphs with radial layout. In: INFOVIS. (2001) 43–50
6. Eades, P.: Drawing free trees. Bulletin of the Institute for Combinatorics and its Applications **5** (1992) 10–36
7. Heer, J., Card, S.K., Landay, J.A.: prefuse: a toolkit for interactive information visualization. In: Proceedings of SIGCHI Human Factors in Computing Systems, New York, NY, USA, ACM Press (2005) 421–430
8. Lok, S., Feiner, S.: A survey of automated layout techniques for information presentations. In: Proceedings of SmartGraphics 2001. (2001)
9. Munzner, T., Guimbretière, F., Tasiran, S., Zhang, L., Zhou, Y.: Treejuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. ACM Transactions on Graphics **22** (2003) 453–462

10. Morris, S.A., Yen, G., Wu, Z., Asnake, B.: Time line visualization of research fronts. Journal of the American Society for Information Science and Technology **54** (2003) 413–422
11. Robbins, K.A., Jeffery, C.L., Robbins, S.: Visualization of splitting and merging processes. Journal of Visual Languages and Computing **11** (2000) 593–614
12. Fry, B.: Computational Information Design. PhD thesis, MIT (2004)
13. National Visualization and Analytics Center: Illuminating the Path: The Research and Development Agenda for Visual Analytics. IEEE Press (2005)
14. Keim, D.A.: Information visualization and visual data mining. IEEE Transactions on Visualization and Computer Graphics **8** (2002) 1–8
15. Hartmann, K., Götzelmann, T., Ali, K., Strothotte, T.: Metrics for functional and aesthetic label layouts. Lecture Notes in Computer Sceinces, Smart Graphics: 5th International Symposium **3638** (2005) 115–127

# AudioRadar: A Metaphorical Visualization for the Navigation of Large Music Collections

Otmar Hilliges[1], Phillipp Holzer, Rene Klüber, and Andreas Butz

University of Munich,
Media Informatics,
Amalienstrasse 17,
80333 Munich, Germany
otmar.hilliges@ifi.lmu.de, philipp.holzer@t-online.de,
klueber@cip.ifi.lmu.de, butz@ifi.lmu.de

**Abstract.** Collections of electronic music are mostly organized according to playlists based on artist names and song titles. Music genres are inherently ambiguous and, to make matters worse, assigned manually by a diverse user community. People tend to organize music based on similarity to other music and based on the music's emotional qualities. Taking this into account, we have designed a music player which derives a set of criteria from the actual music data and then provides a coherent visual metaphor for a similarity-based navigation of the music collection.

## 1 About Songs, Playlists and Genres

In the January 27, 2006, edition of People magazine, reviewer Chuck Arnold likens new Australian duo The Veronicas to 'such pop-rock princesses as Avril Lavigne, Hilary Duff and Ashlee Simpson.' He goes on to state that '*Everything I'm Not*, a cut off their album *The Secret Life Of...* was produced by frequent Britney Spears collaborator Max Martin and has a chorus that echoes Kelly Clarkson's *Behind These Hazel Eyes*.'

When we talk about music and try to explain its properties to others, we frequently use constructs describing similarity. One reason for this is that it is much easier to imagine how a piece of music might sound if we can relate it to a song we already know. This also makes it easier to decide whether or not we might like a song or record that is being discussed.

Describing music by similarity to other music seems to work quite well and is widely used in the music press. However, state of the art digital music players like iTunes [2], Winamp [18] or XMMS [28] do not take this into account. All of these players organize digital music libraries using meta information about the songs/albums (e.g. artist, title) and/or a limited set of predefined genres.

This works quite well as long as we know most of the songs that occur in a library and we know into what genre a song or artist fits. But this approach has several implicit problems:

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 82–92, 2006.

1. Genres aren't expressive enough to cover the breadth of an artist's repertoire. Almost no artist would agree that his entire work can be classified into one single category.
2. Genres are too imprecise to guide users through the vast amount of available music (e.g. the iTunes music store categorizes such diverse artists as punk-rockers Anti-Flag and singer/songwriter James Blunt into the genre "Rock").
3. Genres are very little help to users who want to explore and discover new and unknown music libraries, especially if the artist name is unknown or hard to classify into one of the existing categories.
4. Genres, in general, don't match very well with our moods, e.g. a song from the category rock could be a slow and calm ballad or a fast, rough and loud song.

A major reason for these problems is the imprecise nature of the whole genre concept. With this concept, attempts to classify music often fail because of reasons like ambiguities, subjective judgment and marketing interests. In general, there is a conflict between the broad variety of music (and music properties) and the relatively rigid and error-prone classification system. The fact that meta information is stored in ID3 tags [17], which are created and applied by humans, adds to this problem. In real life most ID3 tags are obtained via online databases like Gracenote CDDB or FreeDB, which are created and maintained by a large community of volunteers. This information is very useful in many scenarios (e.g. displaying song title, album and duration), but there is no quality assurance and, in fact, genre information is often incorrect. For music classification it is a problem that the information is *assigned to* the music and not *derived from* the music.

In response to the problems above, we propose a radically different approach for organizing, browsing and listening to digital music, which is based on two main steps:

1. Instead of relying on meta information, we analyze the music itself, derive a number of meaningful descriptive features from it, and organize the music library by the similarity between songs according to these features.
2. Using this analysis we create a graphical representation for all songs in the library based on similarity. Our visualization employs a radar metaphor as a coherent conceptual model, where similar songs are grouped close together, and the user navigates a musical seascape.

This allows users to surf through their music library (or a music store) guided by similarity instead of scrolling through endless lists.

Our prototype is a new digital music player called AudioRadar. Currently the player has two main functionalities; library browsing and a playlist editor. Both parts of the application are centered around the properties of the actual music and their similarity.

The browser resembles a ship's radar, and the current song is the centroid and similar songs are grouped around it. So a user can immediately understand that nearby songs are similar to the active song but a bit faster/slower or rougher/calmer and so on. The distance from the centroid (along the according dimensions axis) shows how different the songs are.

In the playlist editor users can choose from several dimensions (e.g. speed, rhythm, tone) and specify a range of values she wants to have in her playlist. Thus users can effectively create playlists that suit their mood. This allows the user to create, for example, a playlist containing songs that are relatively slow and calm.

## 2 Related Work and Contribution

Two different aspects need to be addressed in our discussion of related work to the AudioRadar system; the extraction of musical features and the visualization of the music collection.

Our claim, that the automatic extraction of features from musical data can improve music browsing, is backed up by a number of projects in the music information retrieval community, and an overview of MIR systems is given in Typke et al. [22]. Classification mechanisms range from Metadata-based via collaborative filtering approaches to purely feature-based approaches. McEnnis et al. [15] present a library for feature extraction from musical data and discuss other similar work. Liu et al. [13] propose a method for mood detection from low level features, and Li and Sleep [12] as well as Brecheisen et al. [7] even derive genres from low level features. The Music Genome Project [27] relies on features entered by human listeners to classify music, but uses a collaborative filtering approach to create coherent playlists. Uitdenbogerd and van Schyndel [23] discuss collaborative filtering approaches for music information retrieval and how they are influenced by different factors. Schedl et al. [20] propose to use the co-occurrence of artists on Web pages as a measure of similarity and derive a degree of prototypicality from the number of occurrences. Berenzweig et al. [5] give an overview of similarity measures and discuss how subjective they are, and Ellis et al. [9] question whether there even is a ground truth with respect to musical similarity, but try to provide a number of viable approximations. We do not claim to make a technical contribution in the actual analysis of music, but rather use known methods for extracting the features used in our visualization.

The second aspect of our work is the actual visualization of the music collection. The Information visualization community has come up with a number of ways to present big data sets interactively. Classical examples are Starfield displays and scatter plots. The Film Finder[1] applies the Starfield concept to a movie database with several thousand entries. The motivation behind this work is exactly the same as ours, namely to browse and navigate a complex and high-dimensional space according to some meaningful criteria. The MusicVis system[6] uses a scatterplot-like display. It arranges songs as grey, green or blue blobs in a plane and determines proximity between them by their co-occurrence in playlists. MusicVis can also create playlists from its database, which represent coherent subsets of the music collection with familiar song sequences.

The Liveplasma Web site [25] presents a graphical interface to a musician and to a movie database. Starting from a search term it presents the closest match in

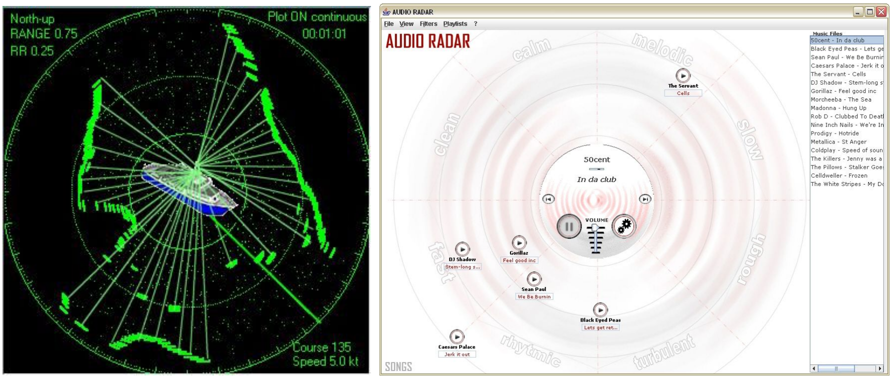



**Fig. 1.** Left: a real radar screen showing own ship as centroid, green dots symbolize other vessels. Right: the AudioRadar application in browsing mode. The active song is the center point, similar songs are grouped around it.

the center of the zoomable display and groups similar artists or movies around it using a spring model based layout mechanism. By clicking on another artist, this one becomes the new centroid and the similar neighbors are dynamically rearranged. Torrens et al. [21] describe visualizations of a personal music collection in the shape of a disc, a rectangle or using a tree map. Vignoli et al. [24,26] present the artist map, a space-conserving visualization for music collections for use on a PDA screen. Our work goes beyond these existing approaches in providing a coherent mental model, the radar metaphor, for the actual visualization as well as for the navigation of the music collection. Pampalk et al. [19] propose a visualization of feature-based clusters of music as "island of music"; but do not provide a means for navigating this seascape.

Our main contribution over these existing visualizations of music collections is the provision of a metaphor from the physical world. Most of us have an intuitive understanding of what a radar is and how it works. We understand the spatial mapping, which tells us where different objects around us are, particularly in which direction and how far away. This spatial mapping is perfectly applied in our interface, since the more similar songs are displayed closer, and the direction tells us, in which aspect they differ. While this is also the case in the film finder, Liveplasma or with the islands of music, the radar metaphor conveys the feeling of literally navigating the musical seascape and supports casual meandering and browsing.

## 3 Navigating the Sea of Sounds

The name AudioRadar obviously refers to the metaphor of a ship's radar, a system that is used to detect, range and map objects such as aircrafts and other ships. In our application we calculate the distance of songs between each other by analyzing the audio stream. We use this information to position songs on a radar-like map where the current song is the centroid (Figure 1).

The center area of the AudioRadar player shows the active song and some controls known from standard music players (play, pause, loudness, progress). Radiating out from that centroid are similar songs positioned along four axes. The direction of their offset is determined by the dominant difference from the active song. As shown in Figure 1 this means that "Gorillaz - Feel Good Inc." is faster than "50 Cent - In Da Club". The distance from the center symbolizes the difference in similarity. A song on the outer rim of the radar could be, say, 100% faster then the centroid. The concentric circles in the background function as visual aides to help users judge the distance of two songs.

By double clicking one of the songs that appear on the radar (or one song from the list on the right) the user can assign the respective song to become the new centroid. The other songs are relocated according to their similarity toward the new centroid. Each of the similar songs further offers a quick-play option that enables the user to just listen to that song without changing the current setup of the songs.

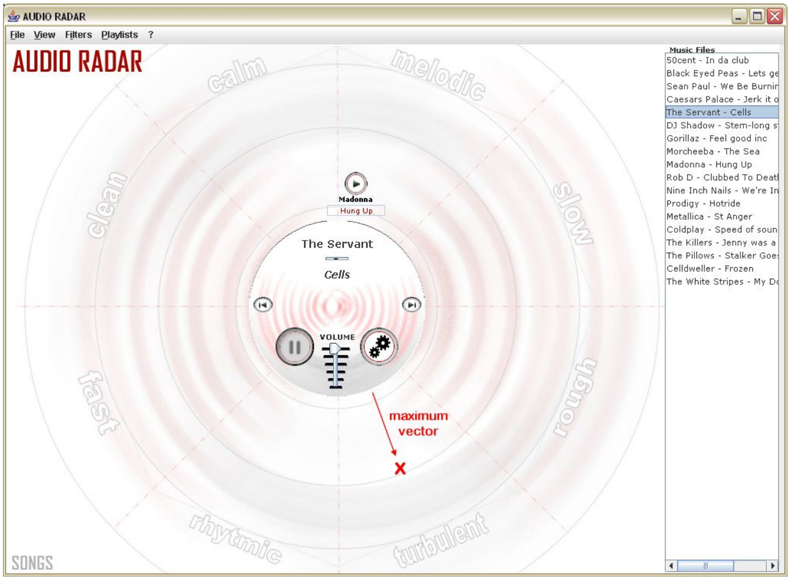
We experimented with different strategies to position the secondary songs. First we calculated the mean value of all extracted attributes and placed the songs accordingly. In some cases this leads to misleading and even wrong placements (see Figure 2 (a)). For example a song that is more turbulent than the centroid could end up in the melodic sector of the radar because the slow and melodic attributes had high values as well. But it was our intention to create a design that contains all attribute dimensions at once and still allows the user to comprehend the most significant type of similarity at fist glance.

One solution for this problem is to dispose all values but the maximum (see Figure 2 (b)). Thus the placement becomes more coherent with the idea that a song is similar to the current one but only faster, for example. This can lead to visual clutter because songs are only placed on the axes of the radar screen. To avoid this problem we use the second highest value to compute an offset from the axes so that the songs get distributed within the maximum sector (see Figure 2 (c)). Utilizing the second highest value as offset in addition makes the offset meaningful for the user.

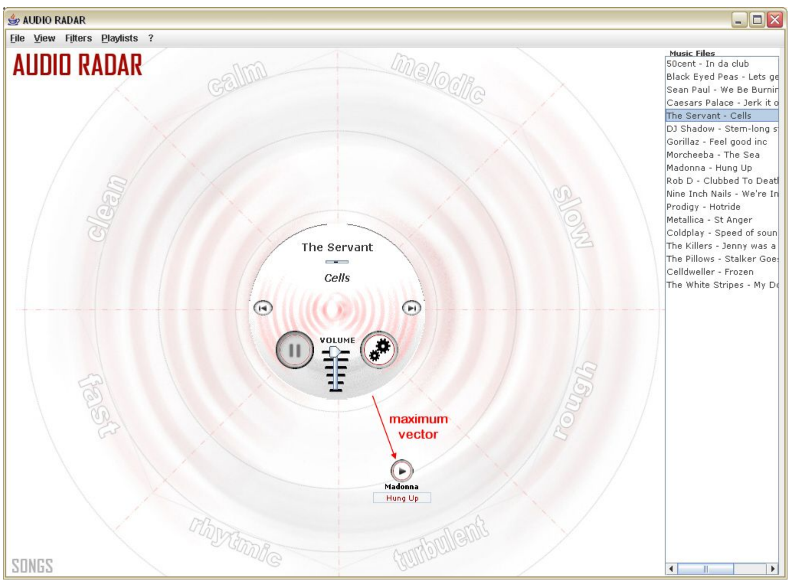
### 3.1 Automatic Audio Analysis

To obtain the data for the placement of each song we analyze the actual audio stream. The four extracted attributes describe each song's position in a four-dimensional feature space. The dimensions are slow vs. fast, clean vs. rough, calm vs. turbulent and melodic vs. rhythmic (see Figure 3). This four-dimensional space is projected onto the two-dimensional display by selecting two of the four dimensions and ignoring the other two (see figure 4). Since the main focus of our work is on the visualization, we used a given analysis library [16] to derive these features. The current results are mostly plausible, but as better algorithms for analysis become available, these can be exchanged in a modular way.

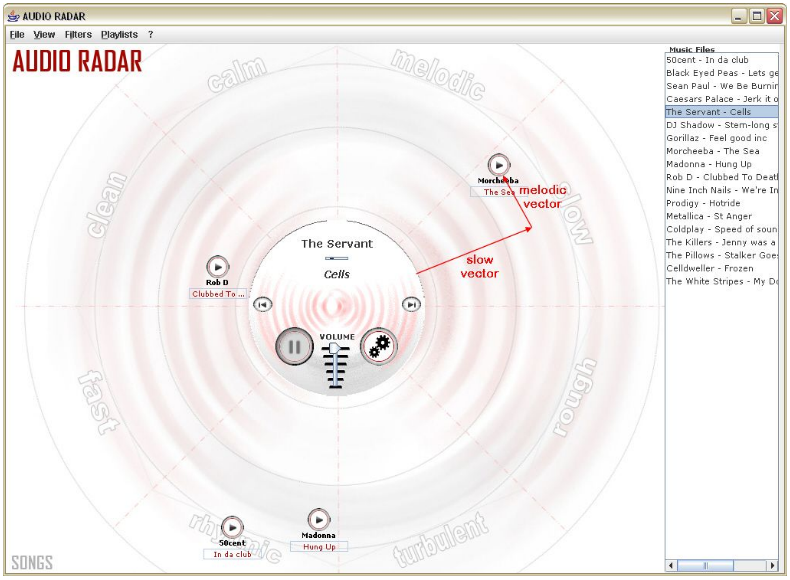
The first attribute we extract is the speed of a song. Basically our algorithm counts the beats per minute. However with some songs, especially non electronic ones, we encountered some difficulties with this approach. Finally we modified


AUDIO RADAR
File View Filters Playlists ?
AUDIO RADAR
calm
melodic
clean
slow
fast
rough
rhytmic
turbulent
SONGS
Madonna
Hung Up
The Servant
Cells
VOLUME
maximum vector
Music Files
50cent - In da club
Black Eyed Peas - Lets ge
Sean Paul - We Be Burnin
Caesars Palace - Jerk it o
The Servant - Cells
DJ Shadow - Stem-long s
Gorillaz - Feel good inc
Morcheeba - The Sea
Madonna - Hung Up
Rob D - Clubbed To Deatl
Nine Inch Nails - We're In
Prodigy - Hotride
Metallica - St Anger
Coldplay - Speed of soun
The Killers - Jenny was a
The Pillows - Stalker Goe
Celldweller - Frozen
The White Stripes - My D

(a)


AUDIO RADAR
File View Filters Playlists ?
AUDIO RADAR
calm
melodic
clean
slow
fast
rough
rhytmic
turbulent
SONGS
The Servant
Cells
VOLUME
maximum vector
Madonna
Hung Up
Music Files
50cent - In da club
Black Eyed Peas - Lets ge
Sean Paul - We Be Burnin
Caesars Palace - Jerk it o
The Servant - Cells
DJ Shadow - Stem-long s
Gorillaz - Feel good inc
Morcheeba - The Sea
Madonna - Hung Up
Rob D - Clubbed To Deatl
Nine Inch Nails - We're In
Prodigy - Hotride
Metallica - St Anger
Coldplay - Speed of soun
The Killers - Jenny was a
The Pillows - Stalker Goe
Celldweller - Frozen
The White Stripes - My D

(b)




(c)

**Fig. 2.** Song placement strategies in AudioRadar. a) A misplaced song positioned with mean value placement. b) The same song positioned utilizing only the maximum attribute. c) Preventing visual clutter along axes using second highest value to calculate offset.

the algorithm so that it is capable of identifying major repetitive elements and count there occurrence over time.

To determine a song's level of noise we simply consider several intervals of the song and measure the difference between the single intervals. This approach can certainly be improved since we don't take the peculiarities of each song into account. We could, for example, achieve much better results by adjusting the intervals length according to the song's structure (verse, refrain etc.). Another improvement would be to extract continuous elements (baseline, chorus) and specifically consider disharmonies, offbeats and related noise.

The dimension calm vs. turbulent is closely related to the previous one but considers the changes in a song over a greater period of time or, in other words, the amount of differing intervals. Again the same limitations as above apply here.

The last dimension we consider is melodic vs. rhythmic and this dimension is the most problematic one. First of all, our very simple algorithm only extracts very basic information about occurring harmonics and rhythm in the song.

Second, this dimension is perceived and judged very subjectively even though it is a very important factor in our liking or disliking off a song. Recent research in the field of music information retrieval has shown that this analysis can be done with quite satisfying results [10,11]. However, applying state of the art technologies would have gone beyond the scope of this project and remains future work.

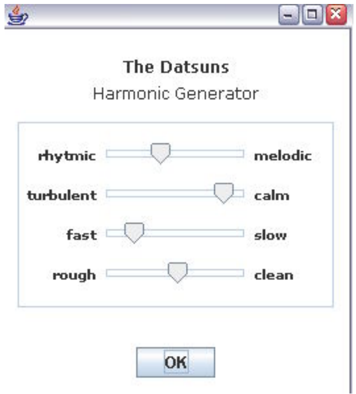



**Fig. 3.** Results of a song analysis showing the values computed for each of the four dimensions

### 3.2 Mood-Based Playlist Generation

Most digital music player software is based on the concept of playlists, which provides users with a way to organize playing orders for their songs. Many digital music libraries easily exceed thousands of songs. Because playlists of this size are no longer easy to manage, text-based search functions are provided to retrieve a specific song or artist. Some programs incorporate specific tools to assemble playlists based on meta information [1]. Or we can give up control altogether and use a randomized playing function.

None of those techniques allows users to create playlists based on their current mood. AudioRadar offers such a functionality by letting users define the preferred range of attributes they would like to listen to. With this approach it is not necessary to know all the songs in the library by name (or even at all), and it is not necessary for the user to know what sort of music lays behind a songs name.

The AudioRadar playlist generator gives the user an overview of the complete library (see Figure 4), with the songs represented as small dots. The user now utilizes sliders to specify a range of values for every dimension that she wants to have in her playlist, thus defining a four-dimensional volume. Songs inside this volume are included in the playlist, and songs outside are not. Since it is impossible to render a four dimensional volume into a 2D view we decided to adopt the well know mechanism of color choosers from common painting programs where multi dimensional color spaces are mapped onto two dimensions by enabling the user to specify the two displayed dimensions.

In our case this means that each of the four attribute pairs can be mapped to one of the two axes of the radar screen. According to this setting all songs are positioned with their computed values. This view denotes a cut through the 4D volume along the two axes. Any combination of two attribute-pairs can be chosen to determine the allocation of all songs within the view.

[1] e.g. iTunes smart playlists.

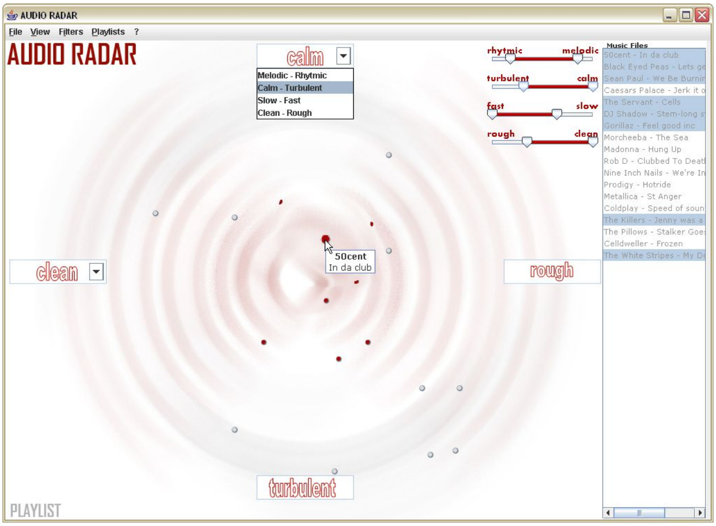



**Fig. 4.** The AudioRadar playlist creator. A red dot indicates that a song is within the currently selected attribute range while a grey dot signifies that it is excluded from the playlist. Four sliders with two thumbs on the upper right corner of the view control the ranges of the attributes. The list view on the right is linked to the radar view and selected songs are highlighted.

Due to the size of digital music libraries, we soon encountered visual clutter to the extent of complete loss of usability. To solve this problem we chose to use very small symbols for the songs and additionally we implemented a fish-eye-lens to magnify the area around the cursor. Hence information about songs can still be retrieved and even songs that are very close to others can be singled out.

## 4 Conclusion and Future Work

In this paper we have presented a digital music player that supports a new way to 1) browse a digital music library and 2) to generate playlists based on the properties of the music itself and the users mood, not based on names or genres applied to music.

The current status of our work is a fully working prototype that implements all the described functionalities. However it has several limitations that need to be addressed in the future. The most urgent is the quality of the audio analysis which, right now, is still very basic and not accurate enough. Early work in that field has been conducted by Logan et al. [14] and Aucouturier et al. [3] give a good summary of state of the art techniques and their application for music information retrieval. Baumann et al. [4] have shown that the retrieved information can be used very effectivly to increase users music browsing experience.

A more problematic issue is that some aspects important to our liking or disliking of music are very subjective and can't be retrieved from the music itself.

We encounter some factors that just can't be measured, such as inspiration, taste or originality. Hence it is very hard to tell whether two songs are perceived as being similar in quality just because they have similar measurable attributes. We simply can't distinguish an uninspired rip-off of a great song by just considering their technical qualities. A solution for this problem might be to consider social and collaborative filtering techniques [8] to incorporate reviews and opinions of music journalists and fans into the rating of songs.

We have not formally evaluated the AudioRadar player, but informal user tests with different target groups, including tech-savvy colleagues, music fans and even musicians, uniformly resulted in very encouraging feedback. In the near future, we plan to improve the audio analysis functionality and conduct a formal evaluation. An especially interesting aspect would be to assess how closely subjective measurements of similarity in songs and the audio analysis of similarity in songs are to one another. Also, we plan to explore in further detail the feature for discovering little- or unknown songs, which a lot of test users especially liked.

## Acknowledgments

This work has been funded by Deutsche Forschungsgemeinschaft (DFG) and the Bavarian state. We thank Amy Ko for valuable feedback on our manuscript.

## References

1. Christopher Ahlberg and Ben Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Human Factors in Computing Systems. Conference Proceedings CHI'94*, pages 313–317, 1994.
2. Apple Computer Inc. iTunes Music player. `http://www.apple.com/itunes`.
3. JJ. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *ISMIR The International Conferences on Music Information Retrieval and Related Activities. Conference Proceedings*, 2002.
4. S. Baumann. Music similarity analysis in a p2p environment. In *WIAMIS '03: Proceedings of the 4th European Workshop on Image Analysis for Multimedia Interactive Services*, 2003.
5. Adam Berenzweig, Beth Logan, Daniel P.W. Ellis, and Brian Whitman. A large-scale evalutation of acoustic and subjective music similarity measures. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2003*, 2003.
6. Blaine Boman, James Pak, and Sajid Sadi. MusicVis. `http://www1.cs.columbia.edu/ paley/spring03/assignments/HWFINAL/bgb10/`.
7. Stefan Brecheisen, Hans-Peter Kriegel, Peter Kunath, and Alexey Pryakhin. Hierarchical genre classification for large music collections. In *IEEE International Conference on Multimedia & Expo (ICME)*, 2006.
8. William W. Cohen and Wei Fan. Web-collaborative filtering: recommending music by crawling the Web. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):685–698, 2000.

9. Daniel P.W. Ellis, Brian Whitman, Adam Berenzweig, and Steve Lawrence. The quest for ground truth in musical artist similarity. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2002*, 2002.
10. J. Foote, M. Cooper, and U. Nam. Audio Retrieval by Rhythmic Similarity. In *ISMIR The International Conferences on Music Information Retrieval and Related Activities. Conference Proceedings*, 2002.
11. Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating Rhythmic Descriptors for Musical Genre Classification. 29, 2005.
12. Ming Li and Ronan Sleep. Genre classification via an lz78-based string kernel. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.
13. Dan Liu, Lie Lu, and Hong-Jiang Zhang. Automatic mood detection from acoustic music data. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2003*, 2003.
14. B. Logan and A. Salomon. A music similarity function based on signal analysis. 2001.
15. Daniel McEnnis, Cory McKay, Ichiro Fujinaga, and Philippe Depalle. jaudio: An feature extraction library. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.
16. M. McGowan, M. Pfisterer, M. Scheerer, D. Szabo, M. Spears, and P. Santon. JLayer - MP3 Library for the Java Platform. `http://www.javazoom.net/javalayer/about.html`.
17. Nilsson M. Informal standard for ID3 tags version 2.3.0. `http://www.id3.org/id3v2.3.0.html`.
18. Nullsoft. Winamp homepage. `http://www.winamp.com`.
19. Elias Pampalk, Simon Dixon, and Gerhard Widmer. Exploring music collections by browsing different views. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2003*, 2003.
20. Markus Schedl, Peter Knees, and Gerhard Widmer. Discovering and visualizing prototypical artists by web-based co-occurrence analysis. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.
21. Marc Torrens, Patrick Hertzog, and Josep Lluis Arcos. Visualizing and exploring personal music libraries. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2004*, 2004.
22. Rainer Typke, Frans Wiering, and Remco C. Veltkamp. A survey of music information retrieval systems. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.
23. Alexandra Uitdenbogerd and Ron van Schyndel. A review of factors affecting music recommender success. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2002*, 2002.
24. Rob van Gulik and Fabio Vignoli. Visual playlist generation on the artist map. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2005*, 2005.
25. Frederic Vavrille. liveplasma. `http://www.liveplasma.com/`.
26. Fabio Vignoli, Rob van Gulik, and Huub van de Wetering. Mapping music in the palm of your hand, explore and discover your collection. In *Proceedings of the International Conference on Music Information Retrieval ISMIR 2004*, 2004.
27. Tim Westergren. The music genome project. `http://www.pandora.com/`.
28. X Multi Media System. XMMS homepage. `http://www.xmms.org`.

# Visually Supporting Depth Perception in Angiography Imaging

Timo Ropinski, Frank Steinicke, and Klaus Hinrichs

Department of Computer Science, University of Münster, Germany
{ropinski, fsteini, khh}@math.uni-muenster.de
http://viscg.uni-muenster.de/

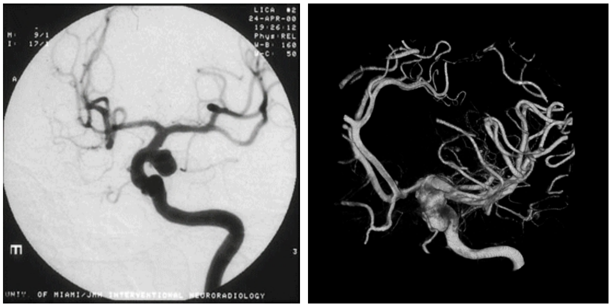
**Abstract.** In this paper we propose interactive visualization techniques which support the spatial comprehension of angiogram images by emphasizing depth information and introducing combined depth cues. In particular, we propose a depth based color encoding, two variations of edge enhancement and the application of a modified depth of field effect in order to enhance depth perception of complex blood vessel systems. All proposed techniques have been developed to improve the human depth perception and have been adapted with special consideration of the spatial comprehension of blood vessel structures. To evaluate the presented techniques, we have conducted a user study, in which users had to accomplish certain depth perception tasks.

## 1 Introduction

In the past years medical imaging techniques have advanced and their application has become essential in medical diagnosis. Nowadays medical scanners allow to acquire high-resolution datasets at relatively low costs. Thus physicians can conduct examinations non-invasively and examine parts of the human body that otherwise would be not accessible. In many areas of medical imaging, 2D visualizations of the - in most cases inherently 3D - datasets are sufficient to communicate the desired information. Therefore medical images are traditionally viewed in 2D, as done for when instance examining x-ray images. 3D is exploited only in application areas where the depth information is essential to provide physicians insights. One of these application areas is angiography imaging used to examine abnormalities of blood vessel structures that usually have a complex 3D structure not easy to comprehend. An example of a cerebral vessel structure with an aneurism acquired through angiography is shown in Figure 1. Although the visualization techniques presented in this paper are applicable to any blood vessel dataset, we focus on cerebral vessel structures acquired through angiography.

Cerebral angiography is commonly used to detect significant stenosis as well as aneurisms. While stenosis are constrictions, aneurisms are expansions of a vessel arising from a too thin vessel membrane. Both stenosis as well as aneurisms cause an increased risk of stroke and must therefore be identified and treated early. There are several ways to treat detected abnormalities. The most common

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 93–104, 2006.



(a) 2D angiogram (b) 3D angiogram

**Fig. 1.** Two angiogram showing cerebral blood vessels. (a) 2D view (source: Neurosurg Focus, American Association of Neurological Surgeons) and (b) a 3D visualization.

cerebral interventional procedure to treat aneurisms is the endovascular embolisation, where the aneurism needs to be packed with coils. This process as well as other treatments require a good spatial comprehension of the vessel structures. As the images in Figure 1 show, the vessel structures segmented from angiography datasets are of a complex nature in terms of furcations and the number of vessels overlapping in depth. The spatial comprehension of these complex structure is impaired by mainly three reasons. First, the structure of the vessels and the accompanying abnormalities may vary. Thus the viewer can not resort to experiences and expectations in the perception process, which is done when perceiving things in everyday life where we have an *unconscious* understanding of the depth structure. Therefore it is especially hard for novices and medical students to spatially comprehend angiograms. The second reason making depth perception of angiography visualizations difficult is the loss of perspective distortion. Since physicians need to conduct measurements within an angiogram, e.g., measuring the length and thickness of a vessel, an orthographic projection has to be used in the visualization process to ensure that angles and lengths do not get distorted. However, orthographic projection does not allow to exploit the depth cue of perspective distortion, which we experience in everyday life when we perceive distant objects smaller. Another difficulty in depth perception of angiography images results in the loss of binocular disparity. In nature we perceive the depth information of close objects by exploiting the viewing parallax given by the distance between our eyes. Although in computer graphics this can be simulated by using stereoscopic techniques, either instrumenting the user with special glasses or using expensive autostereoscopic displays is necessary. Both techniques are not widely available and also not sufficient for everyday use as well as multiple observers. Thus binocular disparity is in most cases not present when viewing angiography images.

To improve depth perception, physicians often view angiograms under motion by rotating them interactively. Because of the rigidity of the structures, the rotation of a vessel complex may give clues about its structure. However, since an orthographic projection is used, no motion parallax is present. Thus the dynamic change of occlusion is the only depth cue arising when performing a rotation. It is obvious that motion cannot be used to enhance spatial comprehension of static images as they may appear in reports or print publications. Furthermore when a group of people watches a dataset it is not easy to talk about certain medical aspects while the object of interest is in motion, since it is not easy to find common reference points.

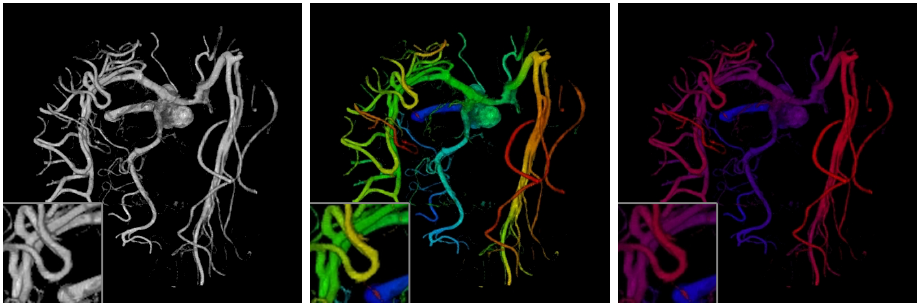
For a more effective as well as efficient analysis it is therefore necessary to develop visualization techniques which support the depth perception of potentially complex vessel structures without requiring stereoscopic display hardware. In this paper we propose and evaluate 3D visualization techniques developed with this goal in mind. Although the techniques have been developed for static images, they can be applied in real-time and are therefore also usable in interactive visualization applications. Furthermore they are of monoscopic nature, which allows to apply them in almost every standard desktop setting without instrumenting the user with special glasses. However, they may be combined with stereoscopic display technologies.

In the next section we are going to describe related work. We will focus on vessel visualization techniques with special consideration of angiography and some proposed models describing the process of depth perception. The visualization techniques we propose to enhance spatial comprehension of angiography images are described in Section 3, where we explain in detail how to color-code depth information, enhance occlusion, simulate perspective distortion and apply an effect similar to depth of field. All these presented techniques are evaluated and discussed in Section 4. Finally, the paper concludes in Section 5.

## 2 Related Work

The most common technique used for visualizing vessel structures is the maximum intensity projection (MIP). In contrast to this direct rendering, model-based approaches generate and visualize a model of the vessel system to support generation of high quality images. The initial work on this topic has been done by Gerig et al. in 1993 [1]. Hahn et al. describe an image processing pipeline to extract models of vascular structures in order to generate high quality visualizations [2]. In 2002 Kanitsar et al. have proposed a model-based visualization technique based on curved planar reformation [3]. Oeltze and Preim have introduced in 2005 the usage of convolution surfaces to further enhance visualization quality especially at vessel furcations [4]. While all other techniques focus on visualizing the vessel structures without contextual information, the *VesselGlyph* [5] provides also context information given by surrounding tissue.

Since we introduce monoscopic visualization techniques to enhance depth perception, a few references regarding this topic are given in this section. We do not discuss monoscopic depth cues in detail, instead we refer to [6,7] which give a good overview. A detailed comparison of the influence of depth cues on depth perception can be found in [8]. For estimating the interplay of depth cues,



(a) regular coloring (b) chromadepth (c) pseudo chromadepth

**Fig. 2.** Depth cues through color coding. The depth is not encoded in the color (a), chromadepth (b) pseudo chromadepth reduced to red and blue (c).

different models have been proposed. They all take into account the combination of different depth cues and postulate a way in which these depth cues contribute to the overall depth perception. The models incorporate for instance the weighted linear sum [9], a geometric sum [10], or the reliability of depth cues in the context of other cues and additional information [11].

## 3 Enhancing Depth Perception in Angiography

We have developed 3D visualization techniques to support depth perception in angiography datasets in order to make the comprehension of these datasets more effective and more efficient. All the presented visualization techniques are of monoscopic nature and can be applied to both still as well as dynamic images. Thus angiography datasets can be explored interactively.

Although shadows are commonly known to support depth perception of computer generated images [8], we have decided not to exploit shadow casting. With shadow casting a new *level of contrast* is introduced, given by the shadow borders. In order to remain simple images in terms of displayed borders and color variations, we are focussing on visualization techniques, which emphasize the present structure instead of introducing new structures, as for instance shadow borders.

The results of the evaluation we have conducted for the presented techniques are discussed in Subsection 4.2.

### 3.1 Pseudo Chromadepth

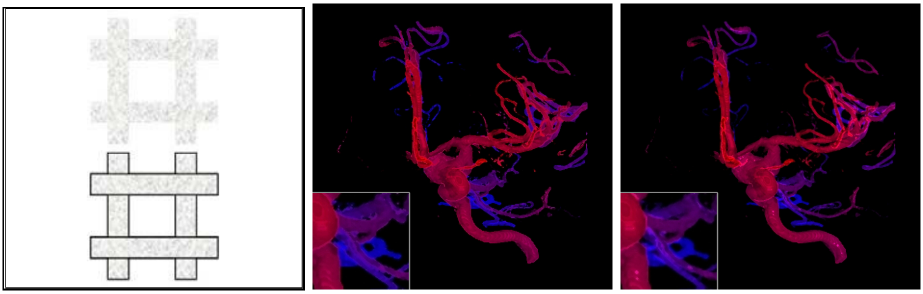
The chromadepth technique [12,13] supports depth perception based on the fact that the lens of the eye refracts colored light with different wavelengths at different angles. Although this effect can be supported by diffraction grating glasses watching appropriate images without instrumentation can also provide the depth effect. Thus a color-coding can be applied to an image, whereas the depth value at each pixel position is mapped to a corresponding color (see Figure 2 (b)). The advantage of this technology is that one can perceive depth in chromadepth pictures also without wearing eyeglasses. However, the selection of colors is limited, since the colors code the depth information of the picture. If the color of an object is changed, then its observed distance will also be changed. When visualizing vessel systems, except for the shading information the color channel usually contains no relevant information. Therefore we were able to apply a depth based color-coding to angiography.

An application of chromadepth color coding to an angiogram is shown in Figure 2 (b). It can be seen that there is a variety of hues present in the image. Since this variety distracts from the shading information, which is necessary to perceive the 3D structure, our goal was to reduce the number of hues used but still allowing a good depth perception. A reduction to a gradient of only two hues already allows a good visual separation of foreground and background elements (see Figure 2 (c)). As it can be seen in the image, we decided to use a gradient running from red to blue for increasing depth values due to two facts. First, the high wave length difference of red light (780nm) and blue light (450nm) results in a different focus point within the human eye. The shorter wavelength of the blue light results in a higher refraction and therefore the point of focus lies closer to the lens, therefore blue objects are perceived to be farther away. This is also the effect exploited in chromadepth images, whereas we concentrated on only two colors, namely those with the highest *contrast* in terms of depth perception. Another aspect important for choosing blue and red for color-coding depth information is the fact that the human eye has a higher color resolution for the colors red and green than for blue. Furthermore the time to respond to a signal varies according to the color used; dark colors lead to a relative high response time whereas light colors ensure quick response times.

When color coding the depth information we had to decide whether a color represents an absolute depth value given by the distance to the viewer or a relative depth value, measured by the depth expansion of the viewed dataset. Since usually only one angiogram is viewed at a time and the resulting colors in the image serve only the spatial comprehension within this particular view, the colors do not have to be comparable across different views or datasets. Therefore we have decided to normalize the depth values $frag_z$ to lie in the interval $[0.0, 1.0]$ and assign the color values by using $frag_{rgb} = (1.0 - frag_z, 0.0, frag_z)$, with the $rgb$ components lying in $[0.0, 1.0]$. Since the normalization does not allow to draw conclusions about the quantitative depth expansion, we have added an appropriately labeled color gradient legend widget showing the color mapping for the occurring depth values.

### 3.2 Occlusion

As denoted by Ware in 2004 [14] occlusion is probably the strongest depth cue. This is due to the binary nature of occlusions which leaves not much space for misinterpretation. For instance, consider the WIMP paradigm, where multiple windows are displayed. Although the windows do not expand in depth, have



(a) occlusion illustration (b) overdrawn edges (c) blended edges

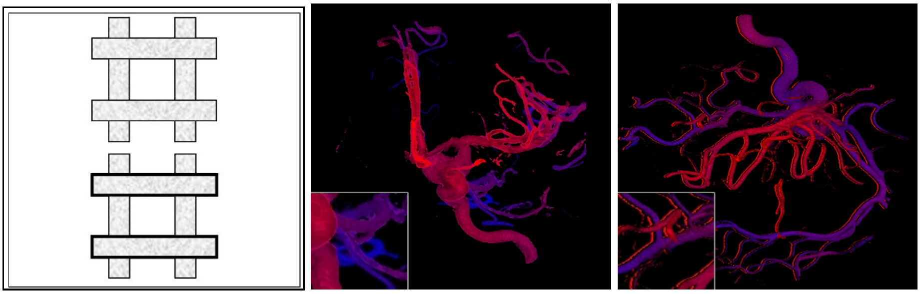
**Fig. 3.** Emphasizing the binary occlusion relation. Occlusions can be emphasized by enhancing the edges of similar colored objects (a), edges overlaid on the vessel structures (b) and edges blended with an opacity of 50% on the vessel structures (c).

the same distance to the viewer and no depth information is given except the occlusion, we get a sense of depth. This is also reflected in the vocabulary used when talking of the *topmost* window. Thus as Ware has stated, "occlusion is the strongest [depth cue], since when an object overlaps another, we receive it as *in front of* the other object, despite there are other depth cues present, stating the opposite".

Thus we have to ensure that occlusion supports the user when viewing angiograms. As it can be seen in Figure 1 (b) occlusion is already present when using regular volume rendering techniques. However, an occlusion is harder to perceive when the overlapping objects have a similar surface shading resulting in a low contrast. This effect, which can also be seen in Figure 1 (b), is illustrated in Figure 3 (a). Two overlapping polygons are shown, which have a similar surface texture, similar to vessels in an angiogram. In the lower picture the edges of the polygons are enhanced and thus the binary occlusion relation is more easy to perceive. To achieve this effect in vessel visualization, we apply an image-based edge enhancement technique [15] with the goal to emphasize the occlusions given by a vessel arrangement more clearly.

A drawback of the introduced edges is the fact, that the edges occlude parts of the shading. This is especially visible when looking at thin vessels. Since shading itself also serves as an important depth cue this effect has to be avoided. We do this by blending semi-transparent edges instead of rendering them with full opacity. The difference can be seen in Figure 3 (b) and (c), where in (b) an opaque edge is overlaid, while in (c) a semi-transparent edge with an opacity of 50% is blended. Notice that in both images the pseudo-chromadepth coloring introduced in the preceding subsection is applied to the edges.

When applying this technique it is important that the thickness of a vessel as displayed on the screen is not affected by the thickness of its edges. Otherwise it would not be possible to measure and compare the dimensions of vessels having a different distance to the viewer. Therefore we ensure that the edge is only visualized on top of those pixels belonging to the vessel anyway.



(a) perspective illustration (b) inverse linear mapping (c) edge shading

**Fig. 4.** Simulation of perspective distortion. Perspective distortion is depicted by varying edge thickness (a), the edge thickness varies inverse linearly to the distance (b) the edge is determined by considering the gradient with the south-west neighbor of each pixel (c).

### 3.3 Simulation of Perspective Projection

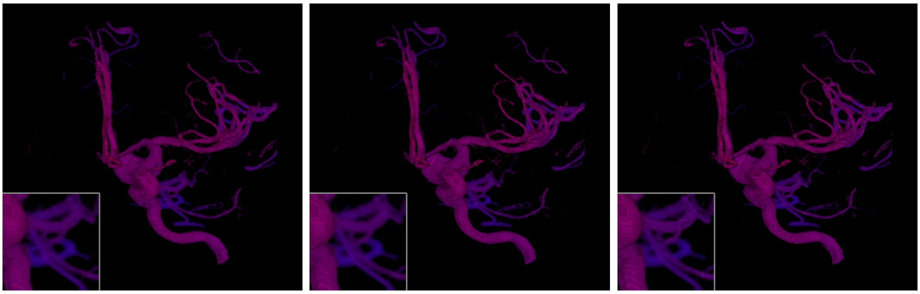
Since an orthographic projection is applied in angiography visualization vessel length and thickness is maintained regardless of the distance to the viewer. Thus the perspective distortion, which plays an important role in depth perception, is not present. However, in everyday live this distortion plays an important role, and we judge the distance of objects depending on their size.

To simulate perspective distortion in angiography imaging we use a varying thickness of the enhanced edges we have proposed in Subsection 3.2. Our technique ensures that edges become thinner with increasing distance to the viewer. The concept is illustrated in Figure 4 (a), where the edges of closer objects are thicker than those of objects being farther away. The technique has been applied to the angiograms shown in Figure 4 (b) and (c). While in (b) the thickness of the edge is inverse linearly proportional to the distance, in (c) a different edge detection algorithm. Therefore we take into account only depth gradients with the south-west neighboring fragment and achieve an appearance, which is similar to shading. The idea is to make the round structure of the vessel more clear, by applying these *fake highlights*.

Again, we ensure that the edges are only rendered on top of the vessels to preserve their thickness.

### 3.4 Depth of Field

When we look around in a real world environment we focus on certain objects located at a particular distance. All objects having the same distance and projection region on the fovea as the objects in focus are perceived sharply, while all other objects being closer or farther away from the viewer are perceived blurry. This rather gradual effect increases with distance and is denoted as *depth of field* or *depth of focus*. It can also be simulated in computer graphics. However, a



(a) distance threshold 10% (b) distance threshold 30% (c) distance threshold 60%

**Fig. 5.** Application of a modified depth of field effect with varying distance thresholds: 10% (a), 30% (b) and 60% (c)

problem occurs since as long as no head tracking device is used, it is not possible to measure which object is in focus of the viewer.

As shown in Figure 5 we have integrated a modified depth of field effect (further referred to as depth of field) into angiogram images in order to support depth perception. As expected and as it can be seen in the images, some parts of the displayed vessel structures become more blurry (for consequences of this effect refer to the discussion in Subsection 4.2). Since in our setup we aim to avoid the use of additional hardware, we had to apply a heuristic to find out on which objects the user is focussing on. When viewing angiogram images the view is usually changed by rotating the object. Through this rotation some of the vessel structures lying in the back get occluded by the parts lying in front. Thus more distant objects are harder to perceive, and therefore we assume that the viewer is focussing on the vessels being closer to the camera. For this reason, we have applied the depth of field effect only for those parts of a vessels whose distance from the viewer exceeds a certain threshold distance. The three images shown in Figure 5 show the depth of field effect applied with different threshold distances. This threshold distance is given as percentage of the depth interval of the vessel dataset. For instance, 40% means that when all vessels lie in the depth interval $[0.0, 1.0]$, depth of field is applied to all structures having a distance greater or equal to 0.4. In the images shown in Figure 5 we have chosen a threshold distance of 10% (a), 20% (b) and 30% (c).

Similar to the application of the depth based color encoding described above, we normalize all depth values of the vessel structure to lie in the interval of $[0.0, 1.0]$. Thus it is not possible to estimate the absolute distance from the viewer, but instead the relative distance of the vessels.

## 4 Evaluating the Proposed Visualizations

The 14 users participating in our user study had to perform simple tasks allowing us to estimate how effectively and efficiently depth can be perceived when viewing angiography images.

### 4.1 User Study

To perform the user study, we have implemented a simple evaluation application. The goal of the study was to measure the effectivity and efficiency of the depth perception when viewing 3D angiogram images. Since the influence of depth cues is task dependent [14,16], we had to choose a task similar to the diagnosis performed by physicians. However, there is no standardized viewing procedure, so we have decided to keep the task simple and hence minimize the influence on the perception of depth cues. Therefore the 14 participating users had to compare the depth of two vessels contained in an angiogram. Since we wanted to eliminate the influence of structure from motion, we made the task static, i.e., the users should estimate the depth based on a single image.

We compared the visualization techniques proposed in Section 3 to a standard 3D visualization as shown in Figure 1 (b) as well as a stereoscopic visualization generated using an autostereoscopic display. During the tests we have shown each user 10 series, each consisting of 5 images rendered using the same visualization technique. Since the perceptual interpretation of a single depth cue shown in isolation is affected by a prior presentation of a different depth cue in the same setting [17], we had to ensure that all the images a user was confronted with either show a different dataset or show the same dataset from a different perspective. The series were: standard rendering (see Figure 1 (b)), stereoscopic rendering, chromadepth (see Figure 2 (b)), pseudo chromadepth (see Figure 2 (c)), overlaid edges (see Figure 3 (b)), blended edges (see Figure 3 (c)), perspective edges (see Figure 4 (b)), edge shading (see Figure 4 (c)), depth of field (10%) and depth of field (10%) combined with pseudo chromadepth (see Figure 5 (a)). Because of their diverse functionality we decided not to evaluate a combination of depth of field with any of the edge based techniques.

In each image two random vessels have been highlighted by flashing square-shaped outlines of equal size. To not distract the user these outlines are displayed only the first 800 ms each image came up. In case the user wants to see these regions later on, she could make them appear as long as the spacebar is pressed. Before each image we have presented a message asking the user to select either the front most or the back most of the two highlighted vessels. This selection could be simply performed by clicking with the mouse inside or near the area of the corresponding square. We have measured the time needed for the selection for each image. Although each user was aware of the fact that we measure the time, we asked them to primarily focus on performing a correct selection than on being fast.

### 4.2 Results and Discussion

After each series the user had to answer a short questionnaire consisting of six questions displayed on the screen. The questions asked were regarding the depth impression, confidence to have performed a correct selection, approximation of selection time, reliability of the images, whether the images are considered usable for daily work and how appealing the images appear. Each question had to be answered on a six point Likert scale, were 1 is considered as a positive answer

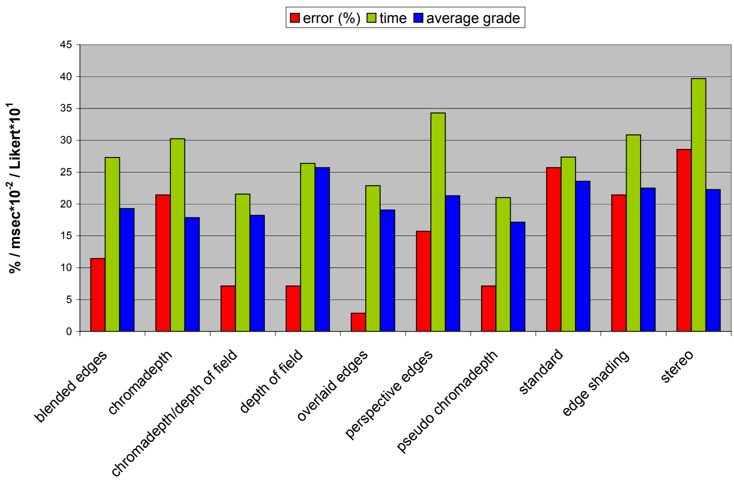



**Fig. 6.** Results of the user study: percentage of erroneous selections, used time and average rating based on questionnaire

and 6 is considered as a negative one. The average percentage of non correct selections, the mean time used to perform a selection as well as the average results of the questionnaire are shown in the diagram shown in Figure 6.

In an informal interview conducted immediately after the user study, the users were asked to denote their preferred visualization technique. Only 2 users said that they prefer to work with the autostereoscopic display, 4 preferred the chromadepth, 4 the pseudo chromadepth, 1 user the depth of field, 2 user the edge shading and 1 user the standard rendering technique. As shown in Figure 6, the standard rendering technique as well as the stereoscopic one did not lead to good results. While with the standard technique this is obvious since the depth information is not available, we expected better results for stereoscopically rendered images. We believe, the reason for the bad results is, that only 4 of the participating users had experience using autostereoscopic displays. Although we asked each participant to test the display before the study and we ensured that there is time available for finding the sweet-spot before the stereoscopic series, the users seemed to have problems perceiving the stereoscopic images.

Chromadepth uses more hues than pseudo chromadepth and should therefore allow to better measure differences in depth. However, it can be seen in Figure 6, that with pseudo chromadepth the users have achieved in average more correct and faster selections. It reduces the error rates about 18% in comparison to the average error rate given by the standard technique. A statistical analysis has revealed, that the hypothesis, that pseudo chromadepth reduces the error rate can be assumed as true with a level of significance of 5%. Furthermore pseudo chromadepth received a slightly better average grade in the questionnaire. This may be due to the fact which came up in the following interview, that 6 users were overwhelmed by the amount of hues present in a chromadepth image. However, as mentioned above we did not explain the color coding in advance;

maybe chromadepth would be more effective and efficient in case we would have explained the techniques before performing the study.

Depth of field also gave quite good results in terms of correctness. However, the response time has been increased, which we ascribe to the fact, that some users may had the impression that the picture is out of focus and hence did need some time to try to get it in focus. This is also reflected in the bad given average grade shown in Figure 6. Again this could be different when we would have informed the users about the technique in advance. Although the overlaid edge technique did lead to very few wrong selections, the edge techniques in general seem not to be very effective nor efficient. Especially the perspective edge technique might be distracting and lead to over 15% of selection errors and quite long selection times.

## 5 Conclusions

We have presented a set of monoscopic 3D visualization techniques serving as depth cues and thus supporting the depth perception of vessel structures shown in angiography images. After we have given a brief overview of depth perception related to angiography, we have proposed the application of interactive 3D visualization techniques to support exploring 3D angiography images. Namely, we applied color coding depth information, depicting occlusions by means of edge detection, simulating perspective by varying edge thickness as well as style and the application of depth of field.

Our evaluation has indicated, that angiography seems to benefit from color coding depth information. Especially the pseudo chromadepth technique combined with a depth of field effect is promising to improve spatial comprehension. In the future it would be necessary to validate our results by performing a more broad user study. In this study a closer look to the usage of the colors is necessary, e.g., to distinguish clearly between the physical effect caused by the different refraction of different wavelengths and the cognitive effect of color coding as well as how a semantic color coding used in certain domains influences the effect. To achieve a better image quality it would be interesting to combine the proposed techniques with model-based visualization of vessel trees.

## Acknowledgements

For a color version of the figures go to viscg.uni-muenster.de/sg06. The authors thank Hagen Schiffbauer for providing valuable insights into how to perform a medical diagnosis based on angiogram datasets and Lucia Terrenghi for discussing ideas about the evaluation. This work was partly supported by grants from the Deutsche Forschungsgemeinschaft (DFG), SFB 656 MoBil Münster, Germany (project Z1).

## References


1. Gerig, G., Koller, T., Szekely, G., Brechbühler, C., Kübler, O.: Symbolic description of 3-d structures applied to cerebral vessel tree obtained from mr angiography volume data. In: IPMI '93: Proceedings of the 13th International Conference on Information Processing in Medical Imaging, Springer-Verlag (1993) 94–111
2. Hahn, H.K., Preim, B., Selle, D., Peitgen, H.O.: Visualization and interaction techniques for the exploration of vascular structures. In: VIS '01: Proceedings of the conference on Visualization '01, IEEE Computer Society (2001) 395–402
3. Kanitsar, A., Fleischmann, D., Wegenkittl, R., Felkel, P., Gröller, M.E.: Cpr: Curved planar reformation. In: VIS '02: Proceedings of the conference on Visualization '02, IEEE Computer Society (2002) 37–44
4. Oeltze, S., Preim, B.: Visualization of Vascular Structures: Method, Validation and Evaluation. IEEE Transactions on Medical Imaging **24** (2005) 540–548
5. Straka, M., Cervenansky, M., Cruz, A.L., Kochl, A., Sramek, M., Gröller, E., Fleischmann, D.: The vesselglyph: Focus & context visualization in ct-angiography. In: VIS '04: Proceedings of the conference on Visualization '04, IEEE Computer Society (2004) 385–392
6. Lipton, L.: Stereographics developers handbook. Technical report, StereoGraphics Corporation (1997)
7. Pfautz, J.: Depth Perception in Computer Graphics. PhD thesis, University of Cambridge (2000)
8. Wanger, L.C., Ferwerda, J.A., Greenberg, D.P.: Perceiving spatial relationships in computer-generated images. IEEE Computer Graphics and Applications **12** (1992) 44–51, 54–58
9. Bruno, N., Cutting, J.: Minimodality and the perception of layout. In: Journal of Experimental Psychology. Volume 117. (1988) 161–170
10. Dosher, B.A., Sperling, G., Wurst, S.A.: Tradeoffs between stereopsis and proximity luminance covariance as determinants of perceived 3d structure. Journal of Vision Research **26** (1986) 973–990
11. Young, M.J., Landy, M.S., Maloney, L.T.: A perturbation analysis of depth perception from combinations of texture and motion cues. Journal of Vision Research **33** (1993) 2685–2696
12. Steenblik, R.: The chromostereoscopic process: A novel single image stereoscopic process. In: Proceedings of SPIE - True 3D Imaging Techniques and Display Technologies. (1987)
13. Bailey, M., Clark, D.: Using chromadepth to obtain inexpensive single-image stereovision for scientific visualization. Journal of Graphics Tools **3** (1998) 1–9
14. Ware, C. In: Information Visualization. Morgan Kaufmann (2004)
15. Saito, T., Takahashi, T.: Comprehensible rendering of 3-d shapes. In: SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques, ACM Press (1990) 197–206
16. Knill, D.C.: Reaching for visual cues to depth: The brain combines depth cues differently for motor control and perception. Journal of Vision **5** (2005) 103–115
17. Domini, F., Caudek, C., Skirko, P.: Temporal integration of motion and stereo cues to depth. Perception & Psychophysics (2003) 48–57

# A Modified Laplacian Smoothing Approach with Mesh Saliency

Mao Zhihong, Ma Lizhuang, Zhao Mingxi, and Li Zhong

Dept. of Computer Science and Engineering, Shanghai Jiao Tong University
Shanghai 200030, P.R. China
{mzh_yu, ma-lz, zhaomx, lizhong}@cs.sjtu.edu.cn

**Abstract.** A good saliency map captures the locally sharp features effectively. So a number of tasks in graphics can benefit from a computational model of mesh saliency. Motivated by the conception of Lee's mesh saliency [12] and its successful application to mesh simplification and viewpoint selection, we modified Laplacian smoothing operator with mesh saliency. Unlike the classical Laplacian smoothing, where every new vertex of the mesh is moved to the barycenter of its neighbors, we set every new vertex position to be the linear interpolation between its primary position and the barycenter of its neighbors. We have shown how incorporating mesh saliency with Laplacian operator can effectively preserve most sharp features while denoising the noisy model. Details of our modified Laplacian smoothing algorithm are discussed along with the test results in this paper.

**Keywords:** Mesh Fairing; Lapalacian Smoothing Operator; Perceptually Salient; Shape Features; Mesh Saliency.

## 1 Introduction

Since Taubin [1] presented a signal processing approach to fair surface design, there has been a substantial amount of work for surface faring of very large meshes, resulting in a variety of algorithms such as the Laplacian operator [1], anisotropic diffusion [2, 3, 4, 5], diffusion of the normal field [6, 7] and bilateral filtering [8, 9, 10]. Much of this work has focused on geometric properties of the mesh, for example, normal vector and curvature. There has been less attention paid to the use of perception-inspired metrics for processing of mesh. In human vision, visual input is first decomposed into a set of topographic feature maps. Different spatial locations compete for saliency within each map, such that only locations that locally stand out from their surround can persist [11]. In the image fields, a saliency map that assigns a saliency value to each image pixel has been successfully applied to object recognition and image registration. In 2005, Lee [12] introduces the concept of mesh saliency for 3D meshes and explores the application to mesh simplification and view selection. Inspired by these successes, we present a novel approach based on mesh saliency, the perception-inspired metrics, for surface fairing in this paper. Our proposed method modifies the Laplacian operator based on mesh saliency to denoise 3D meshes while preserving features.

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 105 – 113, 2006.

The paper is organized as follows. In section 2 we review the related works on the topic of mesh fairing and saliency map. Section 3 discusses the mesh saliency. In section 4 we propose a modified Laplacian smoothing operator based on mesh saliency. Section 5 gives some results of our method and compares our method with other denoising methods. We summarize our work and sketch directions for future research in section 6.

## 2 Related Works

In computer graphics field, robust processing of noisy data is a subject of intensive research over the last decade. But mesh saliency is still a new research domain in computer graphics field. People pay less attention to the use of saliency features for the processing of meshes. In this section, we will give a simple review on mesh fairing and saliency map respectively. In the next section we will discuss the mesh saliency in more details.

**Saliency Map.** Saliency map relies on the fact that human eye movements give strong evidence about the location of meaningful content in an object. People can examine only a small visual area at one time and find the salient location that locally stands out from their surround. In [11], Itti proposed a center-surround operation and has been the most effective techniques for computing saliency. Ruth [13] introduces a simple saliency model for motion search. Based on the detection of salient portions of general images, Bongwon [14] presented thumbnail images technique to help users scan large numbers of images quickly. Saliency algorithms also have been successfully applied to object recognition and image registration [15, 16]. Recently, Saliency maps begin to draw people's attentions in 3D graphics field. Masayuki [17] presents a skeleton-based approach for robust detection of perceptually salient shape features. Simone [18] gives a recognition system for the fast detection and classification of objects in spatial 3D data. Similar to Itti's center-surround operation, Lee [12] introduces the idea of mesh saliency as a measure of regional importance for graphics meshes and discusses how to incorporated it in graphics applications such as mesh simplification and viewpoint selection. He anticipates mesh saliency will be a rich area for further research.

**Mesh Fairing.** The classical Fourier transform of a signal can be seen as the decomposition of the signal into a linear combination of the eigenvectors of the Laplacian operator. Taubin [1] defines the discrete Laplacian operator of a discrete surface signal by weighted averages over the neighborhoods

$$\Delta u_i = u_i^{t+1} - u_i^t = \sum_{j \in neighbor(i)} \varpi_{ij} (u_j^t - u_i^t) \qquad (1)$$

Where the weights $\varpi_{ij}$ are positive numbers that add up to one for each $i$. So Laplacian smoothing is the process, where in each step every vertex of the mesh is moved to the barycenter of its neighbors. Laplacian operator treats features (large variations) and noises (small variations) identically, so it is not feature preserving. Anisotropic diffuse, $\partial_t u - div(g(\nabla u)\nabla u) = 0$, concentrates on the preservation of important surface features like sharp edges and corners by applying direction dependent smoothing. Discretizing the anisotropic diffusion equation, we can get [19]:

$$\Delta u_i = u_i^{t+1} - u_i^t = \frac{\lambda}{|neighbor(i)|} \sum_{j \in neighbor(i)} g(u_j^t - u_i^t)(u_j^t - u_i^t) \tag{2}$$

Carefully choosing $g(\bullet)$, we can set $\omega_{ij} = \frac{\lambda}{|neighbor(i)|} g(u_j^t - u_i^t)$, so in natural, anisotropic diffusion is a modified Laplacian operator. Recently, some people extended bilateral filtering algorithm of image to denoise 3D mesh while preserving sharp features [8, 9, 10]:

$$\Delta u = u' - u = \frac{1}{k(u)} \sum_{q \in neighbor(u)} (\Pi_q(u) - u) \alpha_q f(\| c_q - u \|) g(\| \Pi_q(u) - u \|) \tag{3}$$

where $k(u) = \sum_{q \in neighbor(u)} \alpha_q f(\| c_q - u \|) g(\| \Pi_q(u) - u \|)$ . From eq. 3, we can get $\omega_{ij} = \alpha_q f(\| c_q - u \|) g(\| \Pi_q(u) - u \|) / k(u)$, so bilateral filtering is also a modified Laplacian operator. Many of fairing methods seem very different at the first glance. But in fact they are only variations of Laplacian operator by choosing different $\omega_{ij}$ carefully.

## 3 Mesh Saliency

In this section we only give a simple review for Lee's mesh saliency, please refer to [12] for details. Lee points out the mean curvature in 3D mesh is synonymous with the intensity of an image for saliency computation, so he formulates mesh saliency in terms of the mean curvature used with the center-surround mechanism.

The first step is to compute surface curvature. There are many excellent approaches that generalize differential-geometry-based definition of curvatures to discrete meshes [20, 21, 22]. Let $\ell(u)$ denote the absolute value of mean curvature of vertex $u$ .

The second step is to define the neighborhood of a vertex. In order to make the computation more efficient, unlike Lee's method, we set $N(u,k)$ be the k-neighborhood of vertex $u$ instead of setting $N(u,\sigma)$ be the set of points within a distant $\sigma$ .

The third step is to compute the Gaussian-weighted average of the mean curvature in k-neighborhood:

$$G(u,k) = \frac{\sum_{x \in N(u,k)} \ell(x) \exp[-|x-u|^2 / (2\sigma^2)]}{\sum_{x \in N(u,k)} \exp[-|x-u|^2 / (2\sigma^2)]} \tag{4}$$

Assume that the average length of edges of the mesh is $\bar{e}$ , we set $\sigma = \lambda \bar{e}$ , $\lambda = 1$ is a good selection in our fairing algorithm.

The final step is to compute the saliency $\varphi(u)$ of a vertex $u$ as the absolute difference between the Gaussian-weighted averages computed at small and big neighborhoods:

$$\varphi(u) = | G(u,k1) - G(u,k2) | \quad (5)$$

Where $k1 > k2$ . Shown as Fig.1, high saliency value usually corresponds to the saliently sharp feature.

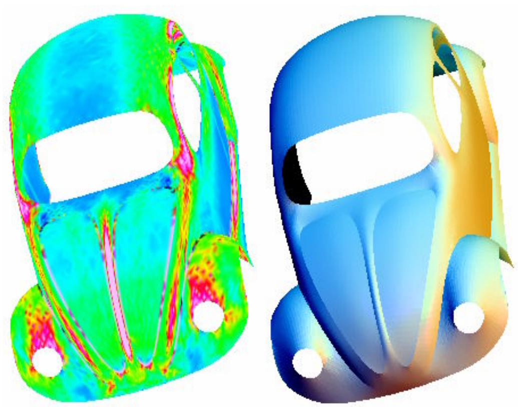


**Fig. 1.** Right: the car body model; Left: Visualize the saliency value of each mesh vertex with color, high saliency value usually corresponds to the sharp feature

## 4 Modified Laplacian Smoothing with Mesh Saliency

Mesh saliency method merges perceptual criteria inspired by low-level human visual system cues with geometric properties based on discrete differential geometry for 3D mesh, so it can capture locally salient regions in meshes successfully, From section 2 we know that many feature-preserving fairing algorithm are originated from the classical Laplacian operator. In this section we discuss how to modify Laplacian smoothing operator to preserve the sharp features of surfaces. Unlike the Laplacian smoothing, where every new vertex of the mesh is moved to the barycenter of its neighbors, we set every new vertex position is the linear interpolation between its primary position and the barycenter of its neighbors.

$$u_i^{t+1} = \alpha u_i^t + (1-\alpha) \sum_{j \in neighbor(i)} \varpi_{ij} \, u_j^t \quad (6)$$

Where $0 \le \alpha \le 1$, $\omega_{ij} = 1/K$ , $K$ is the number of the neighbors (Laplacian operator is the case of $\alpha = 0$ ). In order to preserve the sharp feature points in the fairing process, we need to set $\alpha$ be high where the saliency value is high and vice versa. Denote *maxSaliency* is the maximal saliency value and *minSaliency* is the minimal saliency value in the mesh. We compute $\alpha$ as

$$\alpha = {(\varphi(u) - \min Saliency)} \Big/ {(\text{maxSaliency - minSaliency})} \tag{7}$$

In order to enhance the $\alpha$ value, we introduce a nonlinear *sine* transformation [23].

$$\alpha = \sin(\tfrac{\pi}{2}\alpha), \quad 0 \le \alpha \le 1 \tag{8}$$

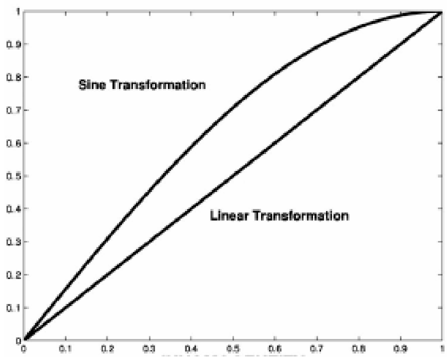



**Fig. 2.** Enhance the $\alpha$ value by nonlinear *sine* transformation

Usually high saliency values correspond to the sharp features. The higher its $\alpha$ value is, the better sharp features preserves. So we amplify the saliency values using a threshold $\beta$ and an amplification factor $\lambda$.

$$A(\varphi(u), \beta, \lambda) = \begin{cases} \lambda\varphi(u), & \text{if } \varphi(u) \ge \beta \\ \varphi(u), & \text{if } \varphi(\mathrm{u}) < \beta \end{cases} \tag{9}$$

## 5 Result

Inspired by Lee's mesh saliency and its successful application to mesh simplification and viewpoint selection, we modified Laplacian smoothing operator with mesh saliency to denoise noisy model while preserving most sharp features. We have implemented our modified Laplacian operator algorithm as described in the previous sections and compared our results to Lapalacian smoothing and bilateral filtering algorithm. The comparison has demonstrated that our method can preserve most detailed features when smoothing a noisy mesh. Fig. 3 gives a comparison to classical Laplacian smoothing algorithm. After 5 iterations, the middle image is blurred and our algorithm (5 iterations, $\lambda = 3.0$ , $\beta$ = $80^{\text{th}}$ percentile saliency value.) still preserves most sharp features. It is obvious in the bottom of the body. Fig. 4 gives a comparison to bilateral filtering algorithm. After 5 iterations, both methods give almost the same smoothing quality. In our modified Laplacian algorithm, we set $\lambda = 2.0$, $\beta = 60^{th}$ percentile saliency value. Usually high saliency values correspond to the sharp features. In order to preserve the sharp features effectively, we amplify the saliency values using a threshold $\beta$ and an amplification factor $\lambda$. Fig. 5 gives some results for different $\lambda$. The higher its $\lambda$ value is, the better the sharp features preserve. Notice the details such as the hair, the eyes and the mouth. Fig. 6 gives some results for different $\beta$. The higher its $\beta$ value is, the better the sharp features preserve. Notice the details such as the eyes and eyebrows. Comparing Fig. 5 to Fig. 6, we can find $\lambda$ is more important than $\beta$. Usually we set $\beta = 60^{th}$ percentile saliency value and $1 \le \lambda \le 3$, the appropriate $\lambda$ value is 2 for most models. The big $\lambda$ can preserve the sharp features while denoising 3D noisy model, but if we arbitrarily amplify $\lambda$, it will make the result worse.

**Fig. 3.** From left to right: noisy model; Laplacian algorithm with 5 iterations; our algorithm with 5 iterations, $\lambda = 3.0$, $\beta = 80^{th}$ percentile saliency value

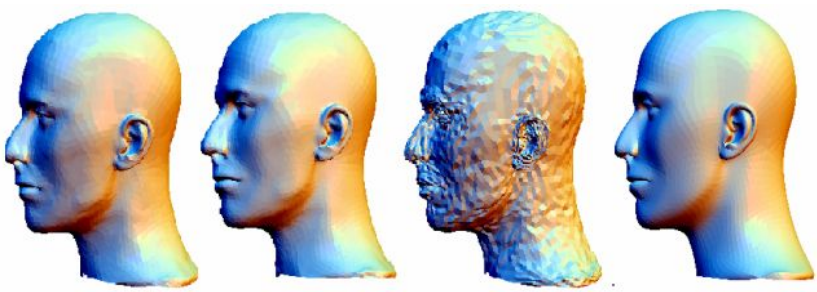


**Fig. 4.** From left to right: our algorithm with 5 iterations, $\lambda = 2.0$, $\beta = 60^{th}$ percentile saliency value; bilateral filtering algorithm with 5 iterations; noisy man head model; origin model

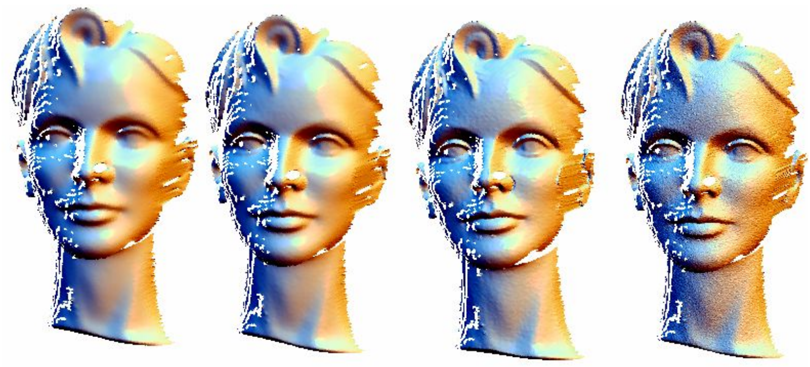


**Fig. 5.** Noisy model smoothed by our modified Laplacian operator with different $\lambda$, from left to right: $\lambda = 1.0$, $\lambda = 2.0$, $\lambda = 3.0$. The last is noisy model. (All with 10 iterations and the same $\beta = 80^{th}$ percentile saliency value).

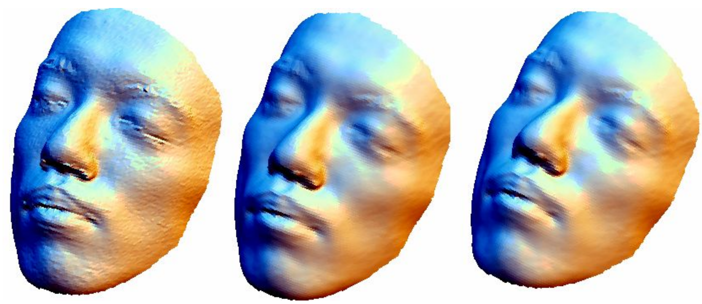


**Fig. 6.** Noisy model smoothed by our modified Laplacian operator with different $\beta$, the first is noisy model. The middle is smoothed model ($\lambda = 2.0$, $\beta = 80^{th}$ percentile saliency value, 5 iterations). The right is smoothed model ($\lambda = 2.0$, $\beta = 0^{th}$ percentile saliency value, 5 iterations).

## 6 Conclusion and Future Work

We compare classical Laplacian smoothing algorithm to anisotropic diffusion and bilateral filtering algorithm. These methods seem very different at the first glance. But in fact they are only variations of Laplacian operator by choosing different $\omega_{ij}$ carefully. In this paper we give a novel approach based on mesh saliency. A good saliency map can capture the interesting local sharp features effectively. So a number of tasks in graphics can benefit from a computational model of mesh saliency. We have shown how incorporating mesh saliency with Laplacian operator can effectively preserve the mot sharp feature while denoising the noisy model. We have demonstrated our algorithm on several models for both noise removal and feature preserving. In Fig. 4, the result of our algorithm is almost similar to the result of bilateral filter algorithm. Currently in graphics field, there has been less attention paid to the use of perception-inspired metrics for processing of meshes. We believe, as Lee says in [12], mesh saliency will promise to be a rich area for further research. For example, most algorithms for detection of surface creases are not immune from the noisy model. How to integrate mesh saliency with robust detection of surface creases is an interesting work in the future.

**Acknowledgement.** We wish to thank Yutaka Ohtake and Prasun Choudhury for providing mesh data. We also gratefully thank Yutaka Ohatake for his meshviewer. The work is supported by national natural science foundation of PR China (Grand No. 60373070) and postdoctoral research foundation of Shanghai, China (Grant No. 05R214129).

## References

1. G. Taubin: A Signal Processing Approach to Fair Surface Design. Proceedings Siggraph'95, (1995) 351–358.
2. C. Bajaj and G. Xu: Anisotropic Diffusion of Noisy Surfaces and Noisy Functions on Surfaces. Texas Institute for Computational and Applied Mathematics, The University of Texas at Austin (2001) TICAM Report 01-07.
3. Klaus Hildebrandt, Konrad polthier: Anisotropic Filtering of Non-Linear Surface Features. EUROGRAPHICS 04, Vol. 23, No. 3. Zuse Institute Berlin (2004).
4. Desbrun, M., Meyer, M., Schr oder, P., and Barr, A. H.: Anisotropic Feature-Preserving Denoising of Height Fields and Bivariate Data. In Graphics Interface, (2000) 145-152.
5. M. Desbrun, M. Meyer, P. Schr¨oder, and A. H. Barr: Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow. Proceedings of SIGGRAPH 99, (1999) 317–324.
6. Tasdizen, T., R. Whitaker, P. Burchard, and S. Asher: Geometric Surface Smoothing via Anisotropic Diffusion of The Normals. Proc. IEEE Visualization 02, (2002) 125–132.
7. Ohtake, Y., Belyaev, A., and Seidel, H.-P.: Mesh Smoothing by Adaptive and Anisotropic Gaussian Filter Applied to Mesh Normal. In Vision, modeling and visualization, (2002) 203-210.
8. Jones, T. R., Durand, F., and Desbrun, M.: Non-Iterative, Feature-Preserving Mesh Smoothing. ACM Transactions on Graphics, Vol.21, No.3 (2003) 943-949.
9. Fleishman, S., Drori, I., and Cohen-Or, D.: Bilateral Mesh Denoising. ACM Transactions on Graphics, Vol.21, No.3 (2003) 950-953.
10. Choudhury, P., and Tumblin, J.: The Trilateral Filter for High Contrast Images and Meshes. Proc. of the Eurographics Symposium on Rendering, (2004) 186-196.
11. Itti L., Koch C. and Niebur E.: A Model of Saliency-based Visual Attention for Rapid Scene Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20, No.11 (1998) 1254-1259.
12. Chang Ha Lee, Amitabh Varshney, David W. Jacobs: Mesh Saliency. Proceedings of ACM Siggraph'05, (2005).
13. Ruth Rosenholtz: A Simple Saliency Model Predicts A Number of Motion Popout Phenomena. Vision Research. Vol.39 (1999), 3163-3175.
14. Bongwon Suh, Haibin Ling, Benjamin B. Bederson and David W. Jacobs: Automatic Thumbnail Corpping and its Effectiveness. In. ACM Proceedings of UIST 03, (2003) 11–19.
15. Vidhya Navalpakkam and Laurent Itti.: Modeling The Influence of Task on Attention. Vision Research . Vol.45 (2005) 205-231.

16. Simone Frintrop, Andreas Nüchter, Hartmut Surmann, and Joachim Hertzberg: Saliency-based Object Recognition in 3D Data, In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'04), (2004).
17. Masayuki Hisada, Alexander G. Belyaev, and Tosiyasu L. Kunii: A Skeleton-based Approach for Detection of Perceptually Salient Features on Polygonal Surfaces. Computer Graphics Forum, Vol. 21, No. 4, (2002) 689-700.
18. Simone Frintrop, Andreas N¨uchter, Hartmut Surmann: Visual Attention for Object Recognition in Spatial 3D Data. In: Proceedings of the 2nd International Workshop on Attention and Performance in Computational Vision, (2004) 168-182.
19. Black, M., Sapiro, G., Marimont, D., and Heeger, D.: Robust Anisotropic Diffusion. IEEE Trans. Image Processing, Vol. 7,No.3, (1998) 421-432.
20. TAUBIN, G.:Estimating the tensor of curvature of a surface from a polyhedral approximation. In Proceedings of IEEE International Conference on Computer Vision, (1995) 902-907.
21. Gady Agam and Xiaojing Tang: A Sampling Framework for Accurate Curvature Estimation in Discrete Surfaces. IEEE Transactions on Visualization and Computer Graphics, Vol. 11, No.5, (2005).
22. Jack Goldfeather and Victoria Inaterrante: A Novel Cubic-Order Algorithm for Appximating Principal Direction Vectors. ACM Transactions on Graphics, Vol. 23, No. 1, (2004) 45-63.
23. Ke Chen: Adaptive Smoothing via Contextual and Local Discontinuities. IEEE Traction On Pattern Analysis and Machine Intellignece, Vol. 27, No.10, (2005).

# 3D Sketching with Profile Curves

Florian Levet, Xavier Granier, and Christophe Schlick

IPARLA project (INRIA futurs - LaBRI),
UMR 5800, Université Bordeaux 1; 351, cours de la Libération
33405 Talence, France
{levet, granier, schlick}@labri.fr

**Abstract.** In recent years, 3D sketching has gained popularity as an efficient alternative to conventional 3D geometric modeling for rapid prototyping, as it allows the user to intuitively generate a large range of different shapes. In this paper, we present some sketching interactions for 3D modeling, based on a set of two different bidimensional sketches (profile curve and silhouette curve). By using these two sketches and combining them with a gesture grammar, a very large variety of shapes (including shapes with topological holes) can be easily produced by our interactive modeling environment.

**Keywords:** Sketch-based 3D Modeling.

## 1 Introduction

Most existing geometric modeling softwares require a lot of skill from the user, to really get the shape he wants to obtain. The complexity of conventional geometric modeling has recently led to the development of some alternative modeling techniques that can be categorized as *3D sketching*. With a sketching metaphor, users can rapidly generate a 3D prototype to illustrate the 3D object they have in mind.

3D sketching is also well suited for being used on mobile devices, such as smartphones, PDAs or tablet PCs. Even if the storage and computation resources of such devices have greatly increased in recent years, the limitation to one-handed (as the other one is usually carrying the device) stylus-based interaction, makes it difficult to adapt existing modeling software that take full advantage of the mouse+keyboard interface.

With a stylus, a drawing metaphor appears quite natural for 3D modeling tasks. Different sketching approaches have already been presented as efficient and intuitive interfaces for creating and editing 3D models. They have shown that a shape can be reconstructed either from strokes and lines [26], or curves and gestures [10,21,11,4,19]. Building upon the latter curves and gestures systems, our goal is to extend the ability of these sketching approaches, while keeping efficient and intuitive interaction.

There are four main contributions in this paper. (i) The first one is a technique to account for a profile curve to infer a 3D model – Section 4. With this approach, the user is not limited to "blobby" shapes but can easily include sharp edges.

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 114–125, 2006.

Moreover, shapes that include topological holes can also be easily generated (see Figure 1). (ii) The second contribution is a set of improvements over the original Teddy system [10] – Section 3. (iii) The third one is a method for local and global edition of the shape by a local or global change of the profile – Section 5. (iv) Finally, we present several gesture-based interactions to create a complex shape from the profile and silhouette sketches – Section 5.

## 2 Previous Work

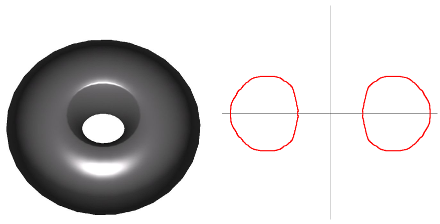
Many techniques are used by artists in order to suggest the object's shape such as characteristic lines (or contour lines) or shaded areas. Since drawing is a familiar task for a lot of people, "sketching" has been introduced as a natural alternative for 3D modeling tasks. Existing sketching approaches can be divided into three categories: the line-and-stroke approach, the painting-or-shading approach, and the curve-and-gesture approach.

The principle of the line-and-stroke approach is to infer a 3D volume from characteristic lines drawn by a user [5,23]. Such a solution can even be interactive [15,18]. When ambiguities occur, they can be removed by a user selection of the correct model into a list of possible reconstructions [14,6]. Most of these approaches are limited to polyhedral models, pre-defined shapes or parametrized objects [25]. With recent methods [3,21], more complex 3D lines can be sketched, but the final model is still limited to a wireframe one. Moreover, stroke-based approaches reduce the limitations on possible 3D lines and curves [20,2], but they are mostly limited to illustration since they cannot really reconstruct a full 3D object.

The painting-or-shading approach allows the generation of highly detailed models. Extending the work of Williams [24], Overveld introduced a geometric modeling technique based on the painting of a surface gradient [22]. Shape editing techniques based on shading information was introduced by Rushmeier et al. [16]. More recently, 3D height-field generation by 2D shading has been presented by Kerautret et al. [12].

Currently, the most powerful techniques have been based on the curve-and-gesture approach. These techniques allow the user to create a large variety of free-form shapes [26,10,13] by using a gesture grammar which converts some drawn curves into a corresponding modeling operation: extrusion, deformation, cutting, etc.

Either variational surfaces [4,11,27], implicit surfaces [17] or convolution surfaces [1] have been used by curve-and-gesture approaches. This has the advantage to generate smooth surfaces, but also emphasizes a "blobby" aspect for the resulting shapes. To reduce this blobby aspect, Tai et al. [19] have proposed to use a profile curve, defined in polar coordinates. Profile curves have been used previously in a sketching environment limited to generalized cylinders [7]. One nice property of using implicit surfaces for the inferred geometry is that the resulting shapes can be easily merged by using classical CSG operators. On the other hand, the main drawback is that some expensive tessellation step has to



(a) (b)

**Fig. 1.** A tore created with two sketches (silhouette curve and profile curve)

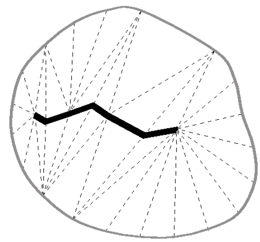


**Fig. 2.** Silhouette curve (grey), reconstructed skeleton (bold black) and corresponding triangulation (dot black)

be employed to convert the surface into a set of triangles that can be sent to the graphics hardware, and complex shapes can require a lot of CSG operations.

Based on these observations, we have designed our sketching environment on the following criteria. First, our system will directly infer a mesh from a set of curve-and-gesture elements. In addition to speed, using a mesh offers the possibility of adapting the sampling according to the local curvature of the surface (i.e., improve the precision where needed). Second, our system will extend the grammar defined in Teddy [10] by creating a profile curve for the geometric model that can be edited either globally or locally.

## 3 General Approach

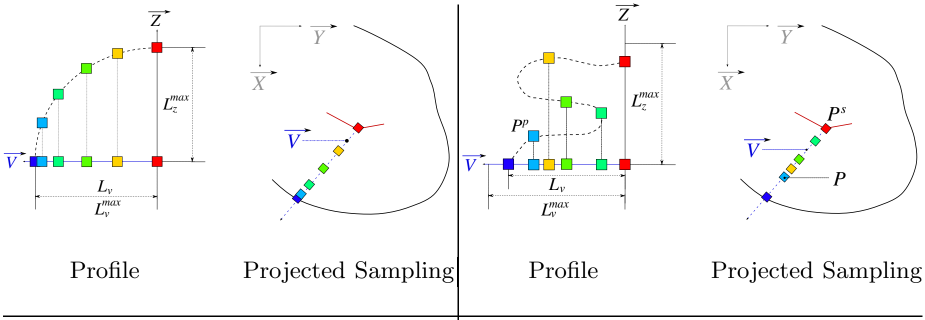
As said above, our approach is basically an extension of the Teddy system, and thus uses a similar four-step process as the one proposed by Igarashi et al [10]:

1. A silhouette curve is sketched by the user and then sampled.
2. A constrained Delaunay triangulation (CDT) is computed from the silhouette samples, and used to extract a skeleton (see Figure 2).
3. For each skeleton point, an elevation distance is computed, as the average distance to connected silhouette samples.
4. Each internal edge of the triangulation is sampled, and the corresponding points are translated according to the elevation distance, to create the final mesh vertices.

Compared to the original Teddy implementation, our system proposes two improvements of this four-step process. First, the elevation step is slightly modified, in order to account for non-convex profile curves. Second, we propose a better estimation of the normal vector at the resulting vertices, in order to obtain a smoother appearance of the final 3D object.

### 3.1 Elevation of Internal Points

With Teddy, mesh vertices are created by sampling the internal edges of the triangulation on the silhouette plane $(\overrightarrow{X}, \overrightarrow{Y})$ and elevating them according to




(a) Convex profile curve (Teddy) | (b) Non-convex profile curve

**Fig. 3.** Difference between a convex and a no-convex profile curve

a circular profile curve (see Figure 3(a)). Unfortunately, this can only work for convex profile curves (i.e., profile curves that can be expressed as a height-field).

When using non-convex profile curves, we propose to compute a set of sample points of the profile curve $P^p = (P^p_v, P^p_z)$, and use these samples along each internal edge to obtain the position of the corresponding mesh vertices $P = (P_x, P_y, P_z)$. The new elevation of a final 3D vertex is given by:

$$P_z = (P^s / L^{max}_z)\, P^p_z,$$

where $L^{max}_z$ is the maximum height of the profile curve (see Figure 3(b)). This approach guarantees that the value of the highest possible elevation is still equal to the original elevation $P^s$ of the corresponding skeleton point. Note that, depending on the profile curve, the elevation distance may be null for some skeleton points, which mean that topological holes may be created along the skeleton, on the resulting 3D model.

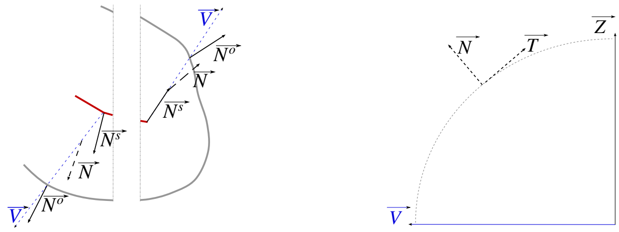
We also need to ensure that the first sample for the profile curve does correspond to a silhouette point. This can be obtained by computing the two other coordinates $P_x$ and $P_y$ as:

$$P_x = P^s_x + V_x \times (P^p_v / L_v) \quad \text{and} \quad P_y = P^s_y + V_y \times (P^p_v / L_v)\,.$$

Note that with convex profile curves as in Teddy, we always have the following condition: $L_v = L^{max}_v$ (see Figure 3(a)).

### 3.2 Improving the Appearance of the Model

One limitation of the original Teddy approach is the lack of smoothness of the resulting shape. To get a pleasant appearance, a smoothing step is generally required [9], which slows down the sketching process. In our approach, we propose to directly enhance the appearance during the creation of the vertices, by using an ad-hock process to estimate normal vectors. This process is based on the following observations: (i) a visual continuity is required on the silhouette points




(a) On silhouette plane (b) On profile curve

**Fig. 4.** Geometric configuration for normal generation

(i.e., the projection of the normal on the silhouette plane has to be collinear with the normal vector to the silhouette); (ii) at non-extremal skeleton points, the projection of the normal on the silhouette plane has to be collinear with the normal vector to the skeleton. (iii) the normal to the surface has to be orthogonal to the profile curve.

Our normal vector computation is totally local, which means that it can be done simultaneously to the vertex generation. In order to guarantee that the two first conditions are fulfilled, a linear interpolation is done between the normal to the silhouette $\overrightarrow{N^o}$ and the normal to the skeleton $\overrightarrow{N^s}$, on the silhouette plane and along each internal edge (see Figure 4 for the notations):

$$N_x = (1-\rho)N_x^s + \rho N_x^o \quad \text{and} \quad N_y = (1-\rho)N_y^s + \rho N_y^o.$$

Note that, for the extremal points of the skeleton, we set the normal as the direction of the internal edge: $\overrightarrow{N^s} = \overrightarrow{V}$. As the normal has to be orthogonal to the profile, we have the condition $\overrightarrow{N} \cdot \overrightarrow{T} = 0$, where $\overrightarrow{T}$ is the tangent to the profile (see Figure 4). This leads to

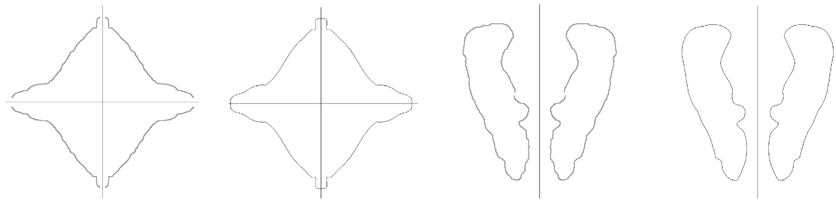
$$N_z = (-T_x/T_z)\,(N_x V_x + N_y V_y).$$

The last step is simply a unit-length normalization of the resulting vector $(N_x, N_y, N_z)$.

# 4 Profile Generation

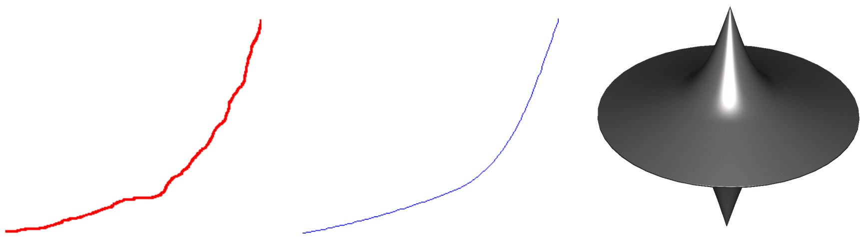
## 4.1 Designing a Profile Curve

As said above, in order to create a geometric model with our sketching environment, the user has to design a second sketch, the profile curve, in addition to the usual silhouette curve. As many shapes include some symmetries, the user may wish to draw only a part of it. The system permits to draw either a quarter of the curve, completed with a double symmetry (see Figure 5(a)) or a half of the curve, completed with a single symmetry (see Figure 5(c)).



(a) (b) (c) (d)

**Fig. 5.** (a)-(c) show two profile curves sketched by users that have been completed by symmetry. (b)-(d) show the corresponding automatically closed and smoothed curves.



(a) Input sketch of the user (b) Smoothed profile curve (c) Resulting model

**Fig. 6.** From the input profile curve to the resulting 3D model

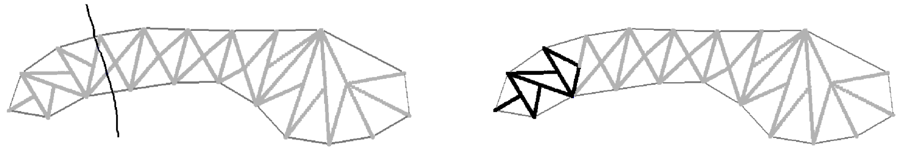
As can be seen in Figure 5(a) and Figure 5(c), the curves sketched by users may be unclosed. In order to generated solid (i.e., closed) objects, the profile curve is automatically closed after sketching, either by projecting its starting and ending points on the axes (see Figure 5(b)), or by linking its first and last points (see Figure 5(d)), according to some distance criterion.

### 4.2 Smoothing the Profile Curve

As a hand-drawn sketch can be very noisy, it is usually more pleasant to smooth out high frequency features before inferring the 3D shape. One easy solution is to use some approximation spline (in our implementation, we use the NURBS++ library[1]) but any low-pass filtering technique may be employed.

Figure 6 shows the relevance of our idea. The profile curve sketched by the user is given in Figure 6(a). It is very noisy with a lot of small discontinuities. The reconstructed curve is shown in Figure 6(b). We can see that the discontinuities have disappeared and that the curve is now very smooth. The resulting 3D model on Figure 6(c) does not present the rough-and-dirty appearance often associated with sketch-based tools.

[1] http://libnurbs.sourceforge.net



(a) User has drawn a sketch of selection (b) Selected points are in black

**Fig. 7.** Local selection of a model

Moreover, since we are using parametric curves, we can easily control the density of their sampling, according to the local curvature (refer to Section 3 for the transformation from the sampling to the mesh vertices).

## 5 Interaction

The design of the profile curve adds a new step in the interaction loop between users and the sketching application. Now they have to draw two sketches in order to create a model contrary to most of the sketching applications that reconstruct a model only with the silhouette sketch. So users have one more gesture to do in order to generate a model. But this gesture allows the creation of more details on the surface, and in only two gestures, it is now possible to create more complex shapes.

Since users have designed the silhouette and profile curves and the model shape was inferred, users can globally edit this shape by modifying the profile curve. When a user sketches a new profile curve, the model shape is deformed on-the-fly in order to match this new profile (see Figure 8). For a more intuitive interaction, a shape can be directly infered from the silhouette by using a default profile, and then, this profile can be globally edited.

Moreover, we have developed a local edition. This increases the range of possible shapes by enabling the combination of different profile curves along the resulting surface. In order to apply a change only locally, users have to first select a region of the surface. The system determines which internal edges have been selected (see Figure 7(b)) from a user-drawn sketch on the screen (see Figure 7(a)). Then, users sketch a new profile curve and the 3D positions of the points on the selected internal edges (i.e., a part of the object shape) are changed in order to match the new profile.

## 6 Results and Discussion

The prototype system was written in the C++ language using the OpenGL library to render models and the Qt library to design the interface. All example models shown in this paper were built with this prototype system running on a standard desktop PC.

Our framework allows users to create a wide range of different models by only drawing two sketches. By using a default circular profile curve, any "Teddy-like"

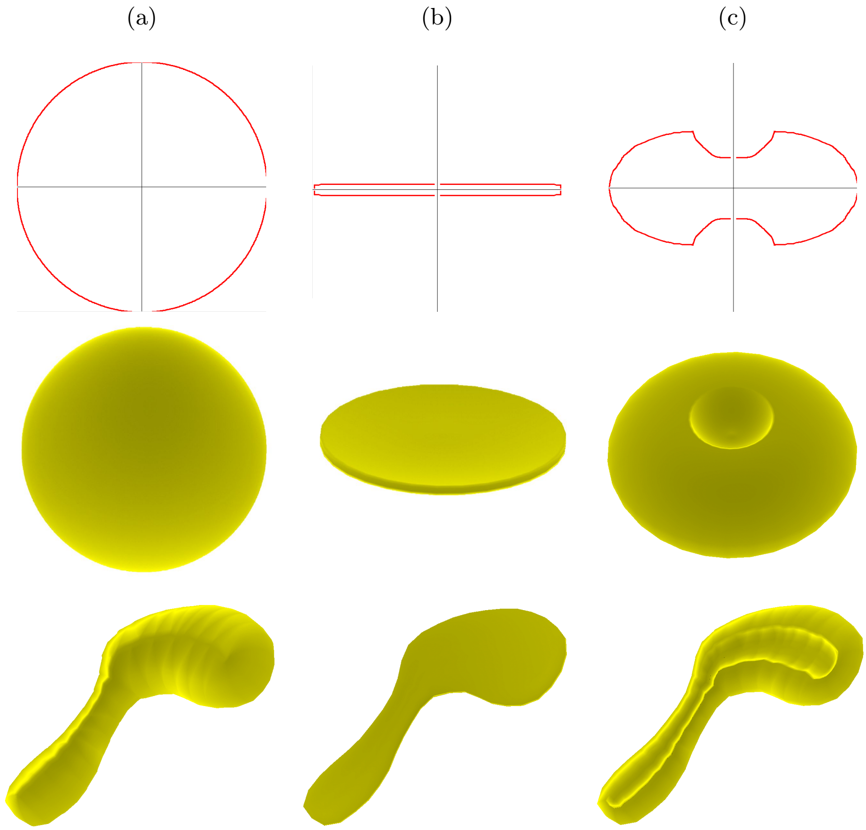



**Fig. 8.** Global edition of two models. (a) shows the creation of two models (one with a circular silhouette curve and the other with a more complex one) with the default circular profile curve. (b)-(c) show the effects of the global edition of the profile curve.

shape can be obtained (see Figure 8(a) column). By offering the possibility to change the profile curve, we greatly increase the variety of shapes that can be generated (see, for instance, the flowerpot on Figure 9).

Figure 8 illustrates the principle of global edition, as discussed in Section 5. First we start from a "blobby" object by drawing a silhouette curve combined with a circular profile curve (Figure 8(a)-up). The profile curve can then be interactively modified (Figure 8(b)-up or 8(c)-up) until the user is satisfied with the resulting shape.

The modification of the profile curve may also be applied only on a part of the shape, as presented in Figure 14. The initial model was a sphere. After selecting a part of the model (in our case, the right half of the sphere), a new profile curve is designed (see Figure 14(a)) and applied on the selected part. The resulting shape is presented in Figure 14(b) and Figure 14(c).

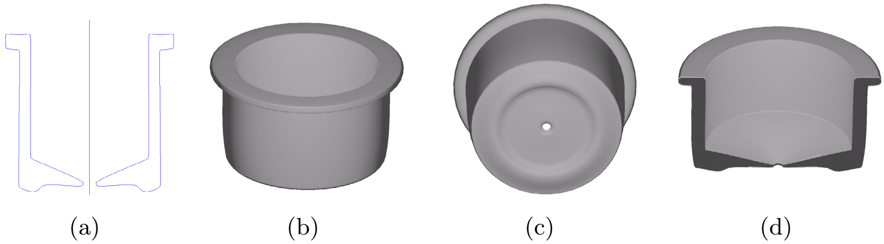



**Fig. 9.** Creation of a flowerpot. This model was designed with the profile curve shown in (a) combined with a circular silhouette curve. (b) and (c) show the pot from different points of view while (d) presents a cut-view. Note that the pot is actually a genus 1 surface.

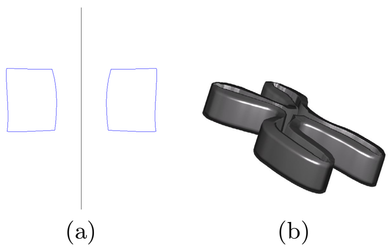



**Fig. 10.** A model with a more complex silhouette curve

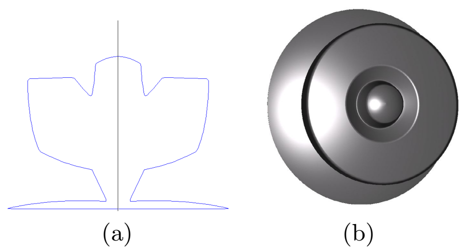


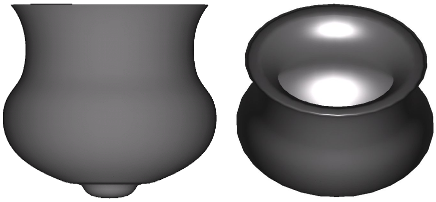

**Fig. 11.** Creation of a door handle with only two curves

More complex shapes can be obtained by using more complex profile curves as in Figure 13. The hammer of Figure 13(a) would require a lot of surface modifications in state-of-the-art sketching systems. But, with our system, we created it with only five sketches. We first sketched the hammer silhouette curve. We then did a first local selection of its head and sketched a square profile curve. Finally we selected the points of its pommel and sketched a profile curve with a crease (quite similar to the profile curve of Figure 8-upright).

## Limitations

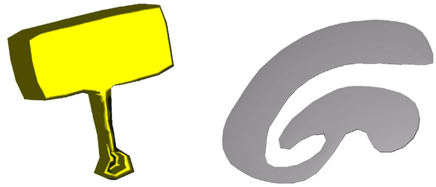
We still have some problems in the sketched shape, mostly due to the skeleton and internal points creations (these points follow the internal edges). As can be seen in Figure 15, even with simple silhouette curves, the skeletons created have a lot of small discontinuities, leading to annoying height differences for their points (as can be seen in Figure 10 and Figure 8-bottom).

Since the creation of the mesh vertices follows the internal edges, the sampling of the model and the resulting triangulation can be under-sampled in certain regions of the model. Because more than one internal edge can end on one outline point, the resulting triangulation is far from being equilateral (with lots of thin triangles) and thus consistent evaluation of normal vectors may be difficult.



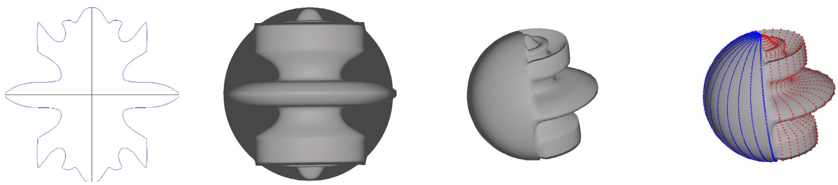
(a) (b)

**Fig. 12.** A vase created with two sketches



(a) (b)

**Fig. 13.** More complex models created with local (a) and global (b) edition. The resulting shape in (a) is a hammer and in (b) is a letter G in 3D.



(a) (b) (c) (d)

**Fig. 14.** Illustration of the local edition. The profile curve applied on the selected point is given in (a). (b-c) shows two views of the resulting model. In (d), mesh vertices are shown.

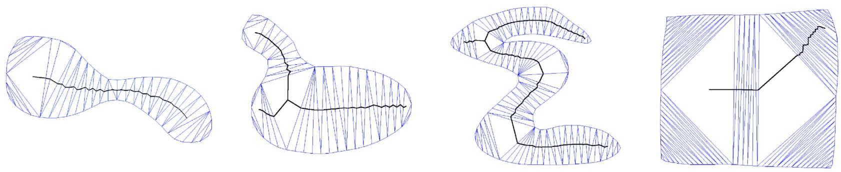


**Fig. 15.** CDT of different silhouette curves. We can see the skeletons discontinuities.

Using the approach described in [8] may be a solution to generate better skeletons without the need to use a CDT. It would require developing a new algorithm to compute internal points.

## 7 Conclusion and Future Work

In this paper, we have presented new sketching interactions for the creation of complex 3D shapes starting from a reduced number of sketched curves. A general non-convex profile curve can be combined with an arbitrary silhouette curve to generate a large variety of resulting 3D shapes.

We have also presented some ideas to improve the mesh generation from the silhouette and profile curves, by locally estimating the normal vector for each mesh vertex. Finally, after the initial mesh generation, some interactive editing, either local or global, may be used to finely adjust the final shape of the object.

Since our approach is based on the initial Teddy system, all the other gesture and curve interactions would be still usable. We haven't currently implemented all of them in our prototype software. Furthermore, a local interpolation of profile curves between selected zones would provide a smoother transition. This can be integrated by using curve-morphing.

The main remaining problem, as described in the Section 6, is the generation of a correct skeleton and the accurate sampling of the silhouette. These steps may lead to visual artifacts, even with our improved normal computation. We believe that a more robust algorithm for the skeleton [8], with an adaptive sampling of the silhouette may provide a solution to the problem.

## References

1. A. Alexe, L. Barthe, M.P. Cani, and V. Gaildrat. Shape modelling by sketching using convolution surfaces. In *Pacific Graphics (Short Papers)*, October 2005.
2. D. Bourguignon, M.-P. Cani, and G. Drettakis. Drawing for Illustration and Annotation in 3D. *Computer Graphics Forum (Proc. Eurographics 2001)*, 20(3):114–122, September 2001.
3. Jonathan M. Cohen, Lee Markosian, Robert C. Zeleznik, John F. Hughes, and Ronen Barzel. An interface for sketching 3d curves. In *SI3D '99: Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 17–21, New York, NY, USA, 1999. ACM Press.
4. A. Cuno, C. Esperança, P. Roma Cavalcanti, and R. Farias. 3D Free-form Modeling with Variational Surfaces. In *WSCG (Journal Papers) 2005*, pages 9–16, January 2005.
5. L. Eggli, C. y. Hsu, B. D. Brüderlin, and G. Elber. Inferring 3D models from freehand sketches and constraints. *Computer-Aided Design (JCAD)*, 29(2):101–112, February 1997.
6. M. J. Fonseca, A. Ferreira, and J. A. Jorge. Towards 3D Modeling using Sketches and Retrieval. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, August 2004.
7. Cindy Grimm and John Hughes. Implicit generalized cylinders using profile curves. In *Implicit Surfaces*, pages 33–41, June 1998. Creating sweep surfaces using sketching.
8. L. He. *A Comparison of Deformable Contour Methods and Model Based Approach Using Skeleton for Shape Recovery From Images*. PhD thesis, University of Cincinnati, 2003.
9. T. Igarashi and J. F. Hughes. Smooth meshes for sketch-based freeform modeling. In *SI3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 139–142, New York, NY, USA, 2003. ACM Press.
10. T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

11. O. Karpenko, J. Hughes, and R. Raskar. Free-form sketching with variational implicit surfaces. In *EUROGRAPHICS '02*, 2002.
12. Bertrand Keraütret, Xavier Granier, and Achille Braquelaire. Intuitive shape modeling by shading design. In *International Symposium on Smart Graphics*, volume 3638 of *Lecture Notes in Computer Science*, pages 163–174. Springer-Verlag GmbH, aug 2005.
13. S. Ohwada, F. Nielsen, K. Nakazawa, and T. Igarashi. A Sketching Interface for Modeling the Internal Structures of 3D Shapes. In *Proc. Smart Graphics*, pages 49–57. Springer-Verlag, July 2003.
14. J. P. Pereira, V. A. Branco, J. A. Jorge, N. F. Silva, T. D. Cardoso, and F. Nunes Ferreira. Cascading Recognizers for Ambiguous Calligraphic Interaction. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, August 2004.
15. D. Pugh. Designing solid objects using interactive sketch interpretation. In *Proc. SI3D '92*, pages 117–126. ACM Press, 1992.
16. H. Rushmeir, J. Gomes, L. Balmelli, F. Bernardi, and G. Taubin. Image-Based Object Editing. In *Proc. 3DIM '03*, pages 20–28, October 2003.
17. R. Schmidt, B. Wyvill, M.C. Sousa, and J.A. Jorge. Shapeshop: Sketch-based solid modeling with blobtrees. In *2nd Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 53–62, 2005.
18. A. Shesh and B. Chen. SMARTPAPER–An Interactive and User-friendly Sketching System. *Computer Graphics Forum (Proc. Eurographics 2004)*, 24(3), September 2004.
19. C.-L. Tai, H. Zhang, and C.-K. Fong. Prototype modeling from sketched silhouettes based on convolution surfaces. *Computer Graphics Forum (Eurographics '04)*, 2004.
20. O. Tolba, J. Dorsey, and L. McMillan. Sketching with projective 2D strokes. In *Proc. UIST '99*, pages 149–157. ACM Press, 1999.
21. S. Tsang, R. Balakrishnan, K. Singh, and A. Ranjan. A suggestive interface for image guided 3d sketching. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 591–598, New York, NY, USA, 2004. ACM Press.
22. C. W. A. M. van Overveld. Painting gradients: free-form surface design using shading patterns. In *Proc. Graphics interface '96*, pages 151–158. Canadian Information Processing Society, 1996.
23. P. A. C. Varley, Y. Takahashi, J.Mitani, and H.Suzuki. A Two-Stage Approach for Interpreting Line Drawings of Curved Objects. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, August 2004.
24. L. Williams. 3D paint. In *Proc. SI3D '90*, pages 225–233. ACM Press, 1990.
25. Chen Yang, Dana Sharon, and Michiel van de Panne. Sketch-based modeling of parameterized objects. In *Eurographics Workshop on Sketch-Based Interfaces and Modeling*, pages 63–72, 2005.
26. R. C. Zeleznik, K. P. Herndon, and J. F. Hughes. SKETCH: an interface for sketching 3D scenes. In *Proc. SIGGRAPH '96*, pages 163–170. ACM Press, July 1996.
27. R. Zenka and P. Slavik. New dimension for sketches. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, pages 157–163, New York, NY, USA, 2003. ACM Press.

# Feature-Preserving, Accuracy-Controllable Freeform Surfaces for Web-Based Surgical Simulations

Akira Wakita[1], Masahiro Kobayashi[2], and Hiroaki Chiyokura[1]

[1] Faculty of Environmental Information, Keio University,
5322 Endo Fujisawa Kanagawa 252-8520, Japan
wakita@sfc.keio.ac.jp, chiyo@sfc.keio.ac.jp
[2] Faculty of Nursing and Medical Care, Keio University,
35 Shinanomachi, Shinjuku, Tokyo, 160-8582, Japan
kkbb@sfc.keio.ac.jp

**Abstract.** In this paper we present a method to generate compact and accuracy-controllable free-form surfaces for web-based surgical simulations. Users can input an important area where high accuracy is required, such as an affected area. The input area can be reflected as meshes with high level of details. By utilizing display tablet interfaces, surgeons are able to draw incision lines or boundaries between organs on an on-screen 3D model, just like they draw lines on paper in surgical planning procedures. Input lines can be reflected as boundaries between patches on free-form surfaces. Practical surgical simulators are also presented to evaluate the efficiency of our framework.

## 1 Introduction

Medical information system is one of the most important applications in virtual environment. Recent years web-based technologies have created completely new possibilities for various medical information systems. The Visible Human Project[17] distributes CT and MRI images on the Internet. Consequently this project offers opportunities to use those medical images through the Internet. Using those data, many research projects have developed medical information systems[12,15,18]. In spite of those developments, each application is not opened to information sharing on the Internet because of lack of data compression abilities.

To provide an efficient way for sharing 3D data on the Internet, many projects have developed 3D medical systems using web3D environments. Volume rendering on the Internet[1] is the most efficient way to visualize 3D medical information. John et al.[9] have developed a suite of VRML based simulators as surgical training tools. In these projects, however, 3D medical models are generated manually by using 3D modeling software packages or they are generated automatically by using 3D reconstruction techniques. 3D models generated by those techniques are represented with triangle meshes. When practical 3D medical models are represented with triangle meshes, the data size will be quite

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 126–137, 2006.

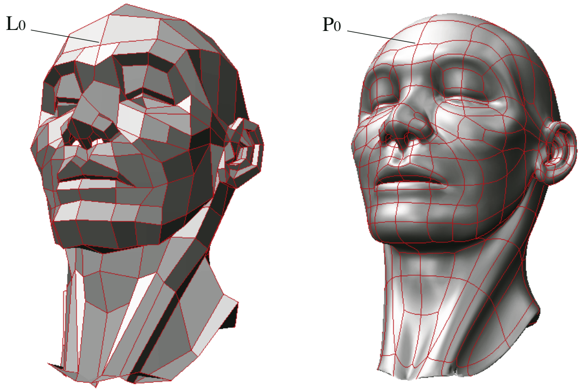



**Fig. 1.** Basic model of lattice surface

huge. In addition, transferring 3D patient models is time consuming, so that interactive deformation or incision simulation is not straightforward. Therefore, a method of representing 3D medical models with small data size is strongly required to realize interactive and practical medical systems.

To address this problem, we have proposed a new data representation called lattice surface[19], and we have applied it for medical information systems[20]. lattice surface is a new free-form surface representation for web3D, which has been proposed as an extension of X3D (Extensible 3D) specifications[13]. Free-form surface techniques enable to transfer high-quality 3D shape rapidly. Taking advantage of this feature, we have developed some web-based medical information systems. These systems worked well to share 3D medical models on the Internet, although, for interactive surgical simulations, e.g., incisions, we still need a method to reflect incision lines and an affected area explicitly to a model. We also need a method to interact with 3D patient models intuitively for practical surgical simulations in virtual environment. To address these problems, we have developed a framework to generate feature preserving and accuracy controllable lattice surfaces. Feature lines are automatically detected or manually input by using tablet interfaces[8,10,14].

We use display tablets to realize intuitive and iterative drawing on 3D models. Surgeons are able to draw incision lines or boundaries between two organs on an on-screen 3D model just like they draw lines on paper in surgical planning procedures. Input lines are reflected as boundaries between patches on lattice surfaces. Users can also input an important area where high accuracy is required, such as an affected area. In this case, patches are kept small and initial triangle meshes are used in order to keep tolerance small.

We have developed a simulator for osteotomy of mandible[21] and a 3D textbook for craniosynostosis[5]. These applications are based on our framework and work on the Internet. We have estimated the efficiency of our framework by using these applications in clinical conferences. Applications are proven to be efficient for surgeons for the following reasons: First, users can transfer high-quality 3D models within a few seconds. Second, since incision lines or organ boundaries are explicitly visualized as boundaries between patches, interactive deformations or cutting operations can be realized. Third, users can easily and intuitively draw lines on 3D models by using display tablets.

In section 2, we review basic models of lattice surface. Section 3 presents a method to convert triangle meshes to lattice surface by using QEM-based surface fitting techniques[7,16]. A method to draw feature lines and an important region on a 3D model is also presented. Section 4 introduces some practical applications based on our framework. In section 5, we evaluate our framework and we present some feedbacks from surgeons. This paper concludes with section 6 with some tasks and future works.

## 2 Basic Models

Lattice surface[19] is a compact and high-quality web3D data representation. Since lattice surfaces are based on free-form surface techniques, the amount of data is extremely small compared with triangle meshes. As shown in figure 1, a basic model of lattice surfaces consists of base mesh (left) and rounded surface (right). A rounded surface has a free-form surface represented by Gregory patches [2]. A base mesh is a simple polygonal mesh consisting of connectivities (vertices/edges/faces), 3D coordinates and rounding weights. These two kinds of representation, base mesh and rounded surface, have one-to-one correspondence about connectivities. In Figure 1, a vertex $\mathbf{L}_0$ of a base mesh correspond to a vertex $\mathbf{P}_0$ of a rounded surface. Based on an invertible rounding algorithm[19], rounded surfaces can be quickly transformed into base meshes, and vice versa. Free-form surface techniques transfer high-quality surfaces rapidly. Users can control the level of rendering in detail by changing the tessellation number according to the machine power or the use of models.

Rounded surfaces represented with Gregory patches are generated by rounding operations[3]. The rounding operation is a popular modeling method to generate a smooth surface from a simple polygonal mesh. We make use of the Doo-Sabin subdivision surface scheme[6] to generate a rounded surface from a base mesh. In our rounding operation, we approximate the limit surface of Doo-Sabin subdivision surface with Gregory patches. The limit surface of Doo-Sabin subdivision surfaces is a set of $C^1$ continuous quadratic B-spline surfaces. On the limit surface, however, it is difficult to represent the control points explicitly. On the other hand, our rounding method outputs control points of free-form surfaces. This is the clear difference between Doo-Sabin scheme and our scheme. Control points of a Gregory patch can be calculated with simple linear transformations. This advantage makes it possible to reconstruct a base mesh from a rounded surface with inverse transformation. This operation is called the inverse rounding operation. A wide range of surface shapes, including piecewise smooth surface, can also be represented by specifying weighting parameters to each edge and vertex. Detailed information of the rounding operation is referred to from our previous paper [19,20].

## 3 Surface Fitting

### 3.1 QEM-Based Surface Fitting

Most of 3D models generated from CT/MRI images are represented with triangle meshes. To use such 3D models in our framework, we introduce a QEM-based surface fitting technique. QEM (Quadric Error Metrics)[7] is an error function that is used in triangle mesh simplifications. Takeuchi et al.[16] have proposed a method to fit Doo-Sabin subdivision surface to triangle meshes by using the QEM function. In their method, mesh simplification based on QEM is applied to construct a base mesh of Doo-Sabin subdivision surface. The limit surface generated from the base mesh will approximate the original triangle meshes with high accuracy.

We extend their method as follows: At the mesh simplification scheme, we use the points on a rounded surface as evaluators of QEM and construct a base mesh for lattice surfaces.

Since a vertex $\mathbf{L}_0$ of a base mesh is mapped uniquely to a vertex $\mathbf{P}_0$ of a rounded surface, $\mathbf{P}_0$ is represented by $\mathbf{L}_0$ and its neighbors as follows:

$$\mathbf{P}_0 = (1 - \omega_0)\mathbf{L}_0 + \omega_0 \sum_{j \in star(\mathbf{L}_0)} k_j \mathbf{l}_j \tag{1}$$

where $\omega_0$ denotes a weighting parameter necessary to generate a piecewise smooth surface.

As shown in [7], in each edge collapse operation, the evaluation function $Q^v$ to calculate an optimized position of $\mathbf{v}$ is:

$$Q^v(\mathbf{v}) = \sum area(f) Q^f(\mathbf{v}) \tag{2}$$

$$Q^f(\mathbf{v}) = (\mathbf{n}^T\mathbf{v} + d)^2 = \mathbf{v}^T\mathbf{A}\mathbf{v} + \mathbf{b}^T\mathbf{v} + c \tag{3}$$

$\mathbf{n}$ denotes the normal vector of a face $f$ and $d$ is a scalar. $\mathbf{A}$ is a symmetric $3 \times 3$ matrix, $\mathbf{b}$ is a vector and $c$ is a scalar.

As $Q^v(\mathbf{v})$ is a quadric function, a minimum value is found by simply solving a linear equation $\nabla Q = \mathbf{0}$. We apply $\mathbf{P}_0$ to Equation (2), which is to solve $\nabla Q^v(\mathbf{P}_0) = 0$. This extended simplification constructs a base mesh instead of a control mesh that approximates a Doo-Sabin subdivision surface.

We assume this simplified mesh as a base mesh to apply to our rounding operation. The rounded shape contains the information of free-form surfaces and it approximates original triangle meshes.

Figure 2 shows an example of surface fitting. The model at the top represents original triangle meshes. The model at middle left is a base mesh (1000 vertices), and at middle right is the rounded surfaces generated from the base mesh. The

**Fig. 2.** Models of jaw

model at bottom left is also a base mesh (500 vertices) and at bottom right is the rounded surfaces generated from the base mesh. Tolerance for approximation is going to be described in section 5.

### 3.2 Reflecting Feature Lines and Accurate Regions

**QEM Control.** For interactive and practical surgical simulations, we still need a method to reflect feature lines explicitly to 3D models. Feature lines are used as incision lines or boundaries between organs. The following our method will reflect feature lines as boundaries between patches on lattice surfaces.

In the QEM-based mesh simplification technique, edges, to which low QEM values are assigned, will be removed first. In other words, when high QEM values are assigned to edges (or vertices), they represent characteristic regions of a model. In that event, edges will be removed at a later time, and in some cases, edges will not be removed. Paying attention to this property, we take the

**Fig. 3.** Drawing on an on-screen model with a display tablet

following approach: High QEM values are set to edges to construct feature lines, so that the edges will not be removed during edge collapse procedure. Consequently, feature lines are reflected to lattice surfaces. By setting high QEM values to edges to construct a specific region, accuracy of the specific region is also maintained.

When a vertex $\mathbf{v}_j$ is included in a highly accurate region $\mathcal{X}$, the evaluation function $Q^v(\mathbf{v}_j)$ will be represented as

$$Q^v(\mathbf{v}) = Q^v(\mathbf{v}^{min}) \cdot U \tag{4}$$

where $U$ is a constrain value. The higher the value of $U$ is, the higher the accuracy of region $\mathcal{X}$ will be maintained.

**Drawing Lines with Display Tablets.** We need a method to intuitively input feature lines and accurate regions in virtual environment[14]. To realize intuitive and interactive drawing on a 3D model, we have developed the environment to use display tablet interface. As shown in figure 3, a surgeon is able to draw incision lines or boundaries between two organs on an on-screen 3D model just like he draws lines on paper in surgical planning procedure.

We take the following steps: Draw feature lines or paint feature regions on an on-screen 3D model with a display tablet[8,10]. Lines drawn on the display coordinate are transformed to those on the model coordinates, and then the texture of lines is mapped onto the surfaces of the 3D model. At this moment, users can browse the 3D model and the feature lines from a variety of viewpoints. Users may continue to draw feature lines from a different viewpoint. After that, the lines are divided and sampled to a set of points, which will be connected by using the shortest path approximation technique proposed by Kanai and Suzuki[11].

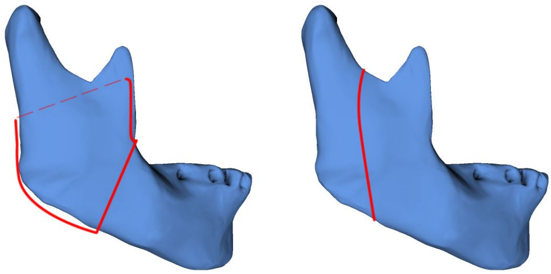


**Fig. 4.** Sagittal Osteotomy (left) and vertical osteotomy (right)

### 3.3 Pairing

The steps described in the previous section realize high accuracy kept in affected area or incision lines. Compared with those important regions, other regions are not required to be highly accurate. To visualize the difference between these two kinds of regions and to reduce data size of a 3D model, we set a pair of adjacent triangles and remove the diagonal edge. In that event, we try to find a pair to form a quadrilateral that approximates a rectangle.

## 4 Application

### 4.1 Osteotomy of Mandible

Osteotomy of mandible is a surgery for congenital asymmetric distortion of a mandible. This surgery consists of sagittal osteotomy and vertical osteotomy (figure 4). Vertical osteotomy cuts a mandible with almost vertical cutting planes. On the other hand, the cutting planes for sagittal osteotomy are quite complicated. To support easy input and iterative trial-and-error steps for the sagittal osteotomy, we have developed a surgical simulator. As shown in figure 3, a surgeon can draw cutting lines directly on the display by using a display tablet. Since cutting planes are recognized as complicated surfaces, it is efficient to draw the cutting lines from multiple viewpoints. As shown in figure 5, incision lines are reflected through the cutting procedure. Figure 6 shows a result of sagittal osteotomy simulations. The complicated cutting planes are interpolated automatically since Gregory patches can interpolate n-sided regions.

### 4.2 Craniosynostosis

Craniosynostosis shows early closing of an infant's head structure. It will result in an abnormal shape of head, so as to cause pressures on an infant's brain. To decrease such pressures on a brain and to force the normal growth of a skull, surgeons incise the skull and reshape affected bones. This simulation visualizes a growing process of a skull and compares the shape of the skill between before and after surgery. In figure 7, the models in the upper row show skulls before

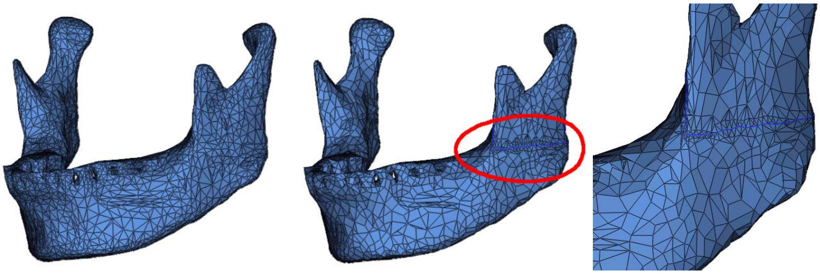


**Fig. 5.** Reflection of incision lines

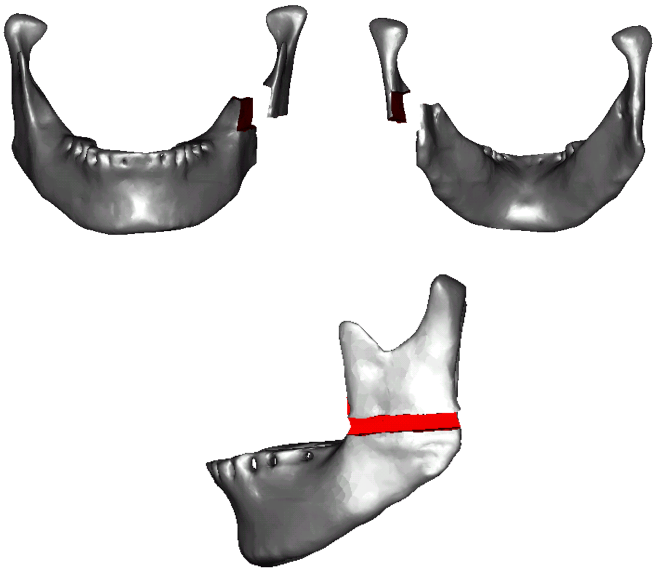


**Fig. 6.** Result of sagital osteotomy

surgery and those in the lower row show the skulls after surgery. The models at the left show original triangle meshes. The models at the center are base meshes of lattice surfaces, and their corresponding rounded surfaces are shown at the right. As shown in figure 8, boundaries of incision lines consist of smaller patches to keep high accuracy. This enables to precisely visualize the growing process of a skull.

## 5 Result

We have estimated the efficiency of our framework through error measurement and clinical conferences. In error measuring estimation, we prepared three types

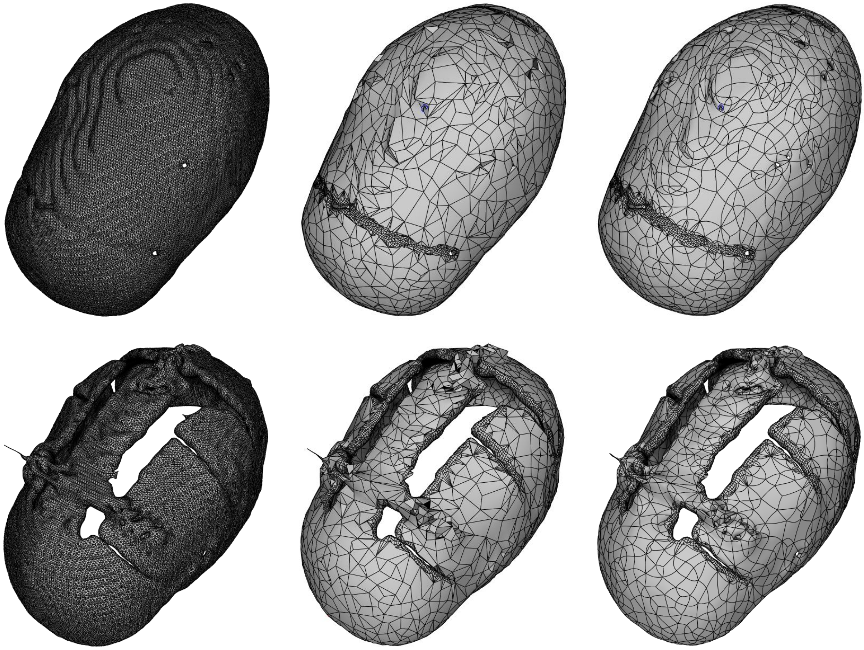


**Fig. 7.** Craniosynostosis

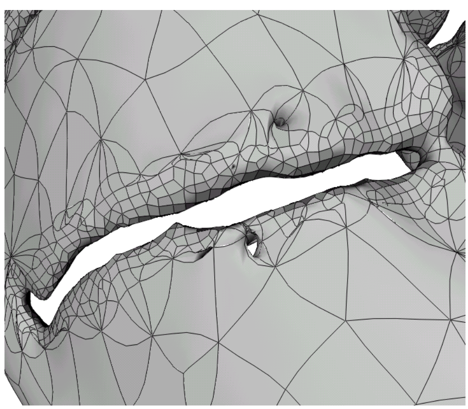


**Fig. 8.** Adaptive control of accuracy

of models: jaw, skull, and skin. For each type of those 3D models, we have measured the execution time of surface fitting and mean square errors between original triangle meshes and resultant rounded surfaces.

In table 1, $V_{org}$ represents the number of vertices in original mesh $\mathcal{M}^0$ and $F_{org}$ represents the number of faces in $\mathcal{M}^0$. Size$_{org}$(KB) shows the data size of

**Table 1.** Result of surface fitting

| Data | $V_{org}$ | $F_{org}$ | Size$_{org}$(KB) | $V_{aft.}$ | $F_{aft.}$ | Size$_{aft.}$(KB) | Time(sec.) | Error(%) |
|---|---|---|---|---|---|---|---|---|
| jaw | 8980 | 17098 | 8590 | 1000 | 2036 | 93.1 | 40 | 0.175 |
| | 8980 | 17098 | 8590 | 500 | 1024 | 47.1 | 45 | 0.264 |
| skull | 20684 | 41399 | 14533 | 2000 | 2031 | 191.0 | 114 | 0.135 |
| | 20684 | 41399 | 14533 | 1000 | 1031 | 96.9 | 128 | 0.234 |
| skin | 13270 | 26547 | 9056 | 1000 | 2007 | 95.6 | 70 | 0.123 |
| | 13270 | 26547 | 9056 | 500 | 1007 | 49.7 | 75 | 0.275 |

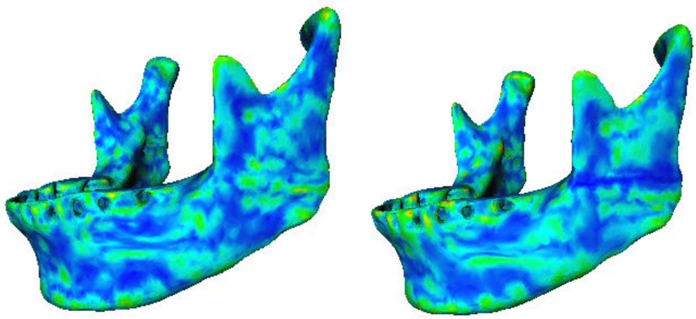


**Fig. 9.** Errors in feature lines and its surrounding region

original meshes $\mathcal{M}^0$ represented with VRML data format. $V_{aft.}$ shows the number of vertices in the simplified mesh (base mesh) $\mathcal{M}^n$ and $F_{aft.}$ shows the number of faces in $\mathcal{M}^n$. Size$_{aft.}$(KB) shows the data size of the base mesh $\mathcal{M}^n$ represented in VRML format. Time (sec.) shows the execution time of mesh simplification and rendering. Error (%) shows mean square errors between the original mesh $\mathcal{M}^0$ and the rounded surface generated from the simplified mesh $\mathcal{M}^n$.

We have used IRI-CNR Metro Tool[4] for mean square error measuring. Monotone increase of mean square errors depending on the number of vertices can be shown. This result shows the properties of QEM correctly and the efficiency of removal from low error values is represented. About the jaw model, the difference of execution time is quite small between the model of 500 vertices and that of 1000, compared with the total execution time. This result shows that the mesh simplification (surface fitting) process itself is quite fast. The data size of base meshes (Size$_{aft.}$(KB)) is about 1% of the data size of original meshes (Size$_{org}$(KB)). It is also shown that the data size depends on the number of vertices of the base mesh of lattice surfaces. This result indicates the efficiency of our framework.

Figure 9 visualizes the difference of mean square errors between a model without feature line input (left) and the one with feature line input (right). Table 2 also shows the results of this visualization with numerals. In figure 9, vertices are colored according to approximation errors. Blue vertices approximate original points with high accuracy and red vertices approximate them with low

**Table 2.** Comparison of errors between models with and without feature lines

| Data | $E_{max}$ | $E_{mean}$ | $E_{ms}$ |
|---|---|---|---|
| with-feature | 0.2652 | 0.05197 | 0.06252 |
| without feature | 0.3209 | 0.06084 | 0.07412 |

accuracy. We can see that feature areas are displayed in blue; that is, high accuracy is maintained in the specified areas.

We have used the applications presented in section 4 at the clinical conference to estimate the efficiency of our framework. It is proved that applications are efficient for surgeons for the following reasons. First, users can transfer high-quality 3D models within a few seconds. Second, incision lines or organ boundaries are explicitly visualized as boundaries between patches, so that interactive deformations or cutting operations are realized. For surgical simulator for the sagittal osteotomy, especially the capability of representing complicated cutting planes is estimated. This depends on the ability of Gregory patches to interpolate n-sided regions. Third, users can easily and intuitively draw lines on on-screen 3D models by using display tablets.

## 6 Summary and Future Work

In this paper, we have presented a new framework to generate feature-preserving and accuracy-controllable free-form surfaces for surgical simulations. Feature lines and regions can be interactively drawn on on-screen 3D models with display tablet interfaces and our software. The data of resultant free-form surface is quite compact, while high accuracy is maintained. Some practical surgical simulators are also presented and the efficiency of our framework is proven through the use of these applications. Our future work is to combine force feedback devices with our framework for more practical and intuitive surgical simulations.

## Acknowledgments

We would like to thank Lattice Technology for the use of their toolkit and models, Daigo Tanaka for initial discussions, and Takashi Kanai for the use of his software and for technical discussions.

## References

1. Johannes Behr and Marc Alexa, Volume Rendering in VRML, ACM Web3D 2001, (2000) 23–27
2. Hiroaki Chiyokura and Fumihiko Kimura, Design of Solids with Free-form Surfaces, Computer Graphics (SIGGRAPH 83 Proceedings) (1983) 289–298
3. Hiroaki Chiyokura, An Extended Rounding Operation for Modeling Solids with Free-Form Surfaces IEEE Computer Graphics and Applications, vol.7 number 12, (1987) 27-36

4. P. Cignoni, C. Rocchini and R. Scopigno, Metro: Measuring Error on Simplified Surfaces, Computer Graphics Forum, vol. 17, number 2, (1998) 167-174
5. M. Michael Cohen and Ruth E. MacLean, Craniosynostosis: Diagnosis, Evaluation, and Management, Oxford University Press, 2000
6. D. Doo and M. Sabin, Behaviour of recursive subdivision surfaces near extraordinary points, Computer Aided Design, vol.10, number 6, (1978) 356-360
7. Michael Garland and Paul S. Heckbart, Surface Simplification Using Quadric Error Metrics, SIGGRAPH 97 Proceedings, (1997) 209–216
8. Takeo Igarashi, Satoshi Matsuoka and Hidehiko Tanaka, Teddy: a sketching interface for 3D freeform design, Computer Graphics (SIGGRAPH 99 Proceedings) (1999) 409–416
9. Nigel W. John and Nicholas Phillips, Surgical Simulators Using the WWW, Medicine Meets Virtual Reality 2000, (2000) 146–152
10. Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes and Adam Finkelstein, WYSIWYG NPR: drawing strokes directly on 3D models, Computer Graphics (SIGGRAPH 02 Proceedings) (2002) 755–762
11. Takashi Kanai and Hiromasa Suzuki, Approximate Shortest Path on a Polyhedral Surface and Its Applications, Computer Aided Design, vol. 33, number 11, (2001) 801-811
12. R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Büren, G. Fankhauser and Y. I. H. Parish, Simulating Facial Surgery Using Finite Element Models, SIGGRAPH 96 Proceedings, (1996) 421–428
13. Lattice Technology, Lattice Extension for X3D, http:// www.web3d.org/x3d/ content/examples/LatticeXvl/docs/LatticeXvl/ LatticeXvlOverview.html 2000
14. Andrew Nealen, Olga Sorkine, Marc Alexa and Daniel Cohen-Or, A sketch-based interface for detail-preserving mesh editing, ACM Trans. Graph., vol. 24, number 3, (2005) 1142–1147
15. Bernhard Pflesser, Andreas Petersik, Andreas Pommert, Martin Riemer, Rainer Schubert Ulf Tiede and Karl Heinz Hohne, Exploring the Visible Human's Inner Organs with the VOXEL-MAN 3D Navigator, MMVR 2001, (2001) 379–385
16. Shingo Takeuchi, Takashi Kanai, Hiromasa Suzuki, Kenji Shimada and Fumihiko Kimura, Subdivision Surface Fitting with QEM-based Mesh Simplification and Reconstruction of Approximated B-spline Surfaces, Pacific Graphics 2000, (2000) 202–212
17. The Visible Human Project, National Library of Medicine, http:// www.nlm.nih.gov/ research/visible/visible_human.html, 1995
18. Virtual Human Project, Oak Ridge National Laboratory, http://www. ornl.gov/ virtualhuman/ 1999
19. Akira Wakita, Makoto Yajima, Tsuyoshi Harada, Hiroshi Toriya and Hiroaki Chiyokura, XVL : A Compact and Qualified 3D Representation based on Lattice Mesh and Surface for the Internet, ACM Web3D-VRML 2000, (2000) 45–51
20. Akira Wakita, Takamichi Hayashi, Takashi Kanai and Hiroaki Chiyokura, Using Lattice for Web-based Medical Applications, ACM Web3D 2001, (2001) 29–34
21. S. Zachow, E. Gladilin, R. Sader and H.-F. Zeilhofer Draw & Cut: Intuitive 3D Osteotomy Planning on Polygonal Bone Models, Computer Assisted Radiology and Surgery, 2003

# The Sketch L-System: Global Control of Tree Modeling Using Free-Form Strokes

Takashi Ijiri[1], Shigeru Owada[2], and Takeo Igarashi[3]

[1] Department of Computer Science, University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 133-0033, Japan
ijiri@ui.is.s.u-tokyo.ac.jp
[2] Sony Computer Science Laboratory, Inc.
3-14-13, Higashigotanda, Shinagawa-ku, Tokyo 133-0033, Japan
sowd@acm.org
[3] Department of Computer Science, University of Tokyo / PRESTO, JST
takeo@acm.org

**Abstract.** L-system is a tool commonly used for modeling and simulating the growth of plants. In this paper, we propose a new tree modeling system based on L-system that allows the user to control the overall appearance and the depth of recursion, which represents the level of growth, easily and directly, by drawing a single stroke. We introduce a new module into L-system whose growth direction is determined by a user-drawn stroke. As the user draws the stroke, the system gradually advances the growth simulation and creates a tree model along the stroke. Our technique is the first attempt to control the growth of a simulation in L-system using stroke input.

## 1 Introduction

Fractals are commonly used in computer graphics, not only for creating visual effects, such as fires, clouds, and lightning, but also for modeling living organisms. Lindenmayer introduced L-system [8], which represents the fractal structure of plants using grammatical expressions. Starting with an initial structure, L-system constructs the fractal structure by applying rewriting rules sequentially. One important aspect of L-system is that the rewriting process closely mimics the growth of real plants, which makes L-system the most popular tool for modeling plants. However, the replacement of parts essentially involves a local description, and local changes strongly affect the global shape. This causes difficulty in creating a desired global shape using L-system.

Our goal is to allow the user to control the global shape of an L-system model easily and directly. In this paper, we propose a system that allows the user to specify the central axis of the plant by drawing a stroke. We also use the stroke to determine the depth of recursion; when the user draws a longer stroke, the system applies generating rule more times. A key idea is introducing a special module whose growth direction is determined by a user-drawn stroke. Figure 1 shows our prototype system. The system provides a predefined generating rule that consists of an internode, lateral apices, and the user-controlled top apex (Figure 1(a)). The user begins modeling by specifying the parameters of the generating rules, and then draws a free-form stroke representing the

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 138 – 146, 2006.

central axis. As the user draws the stroke, the system advances the growth simulation by growing the top apex along the stroke and applying the generating rules to the lateral branches (Figure 1(c)).

Using our system, the user can control the axis and depth of recursion intuitively, by drawing a single stroke. Whereas the generating rule represents local structures, the stroke roughly specifies the global shape. Therefore, the user can easily create a variety of tree models that have the same local structure, but different global shapes, by drawing different strokes with the same generating rules. Figures 1 and 6 show tree models made using our system. The user can create these models easily by manipulating a few control points and drawing a stroke.

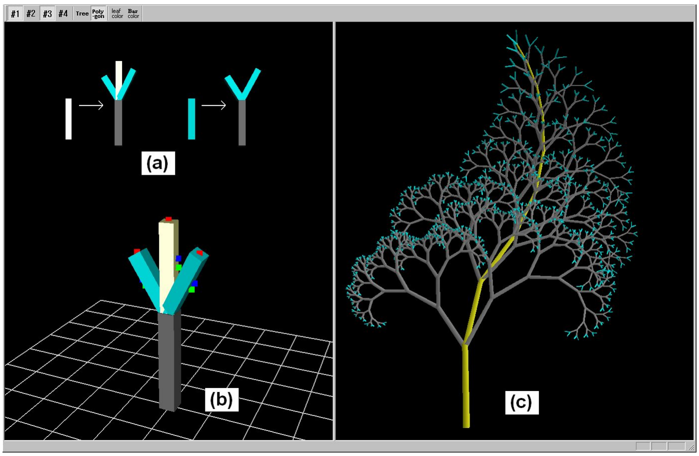



**Fig. 1.** A snapshot of the proposed system. The user first modifies the generating rules (a) by manipulating control points on a handle (b). Next, the user draws a single stroke representing the axis in the right panel (c). Then, the system creates a fractal structure along the stroke. The user-drawn stroke is highlighted in yellow.

## 2 Related Work

Our work is built upon knowledge in several different fields. Here we briefly review representative references in the three most relevant fields: L-System, plant modeling, and sketch-based interface for 3D modeling.

### 2.1 L-System

A. Lindenmayer originally introduced L-system in 1968 to formalize the development of multicellular organisms [8] and subsequently expanded it to represent higher plants and complex branching structures [5]. The framework of L-system consists of an *initial structure* and *rewriting rules* (or *generating rules*). The essence of development is parallel replacement using the rewriting rules. Starting from the initial structure, L-system replaces each part of the current structure by applying the rule sequentially. Figure 2 shows a simple example, the development of a compound leaf [17].

This includes two module types: the *apices* (thin lines) and the *internodes* (thick line). In this example, there are two rewriting rules (Figure 2, top-left): one replaces an apex with an internode, two lateral apices, and a top apex, while the other replaces an internode with a longer one. The initial structure is a single apex. Using these simple rules, the system develops an intricate branching structure over a number of replacing steps. An interesting aspect of the system is that each replacement process corresponds to the growth of part of the plant. Therefore, L-system is not only a heuristic technique that creates fractal-like shapes, but is also a simulation of real-world plant growth.

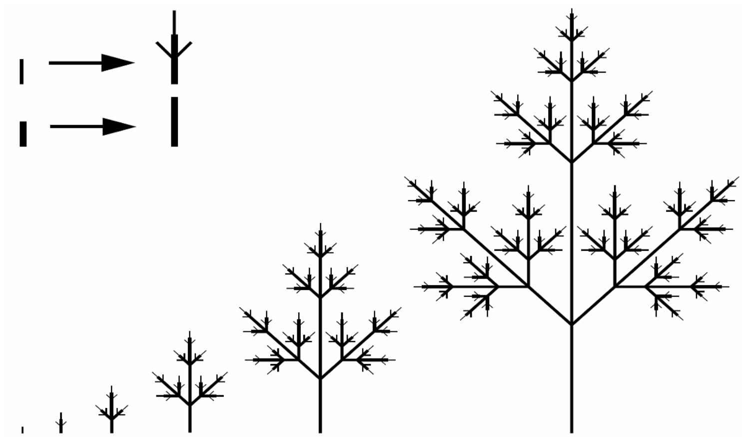


**Fig. 2.** Developmental model of a compound leaf. Beginning with the initial structure (leftmost thin line), the system generates a complicated structure by applying two generating rules (top-left) sequentially. This figure is from [17, page 4].

### 2.2 Plant Modeling

Lindenmayer [8] originally invented L-system, and Prusinkiewicz and Lindenmayer later introduced it to the computer graphics community [16]. L-system has been extended to simulate a wide variety of interactions between plants and their environments [9] [17] [19]. Subsequently, they designed a plant-modeling language called cpfg [18], which supports both context-free and context-sensitive generating rules. They also implemented the plant modeling software L-Studio based on cpfg [20]. Recently, Prusinkiewicz *et al.* [21] also proposed the use of positional information to control parameters along a plant axis. Boudon *et al.* [2] proposed an L-system-based process for designing bonsai tree models, which uses decomposition graphs to facilitate the manipulation of hierarchical parameters. These systems produce intricate, realistic plant models and illustrate the great potential of L-system. However, these systems focus only on manipulating the rules and parameters, and do not support direct manipulation, such as drawing an axis.

Streit *et al.* introduced a biologically based method for modeling plant variation from a basic plant model represented by L-system [22]. Their system mimics the underpinning of variation in real plants by simulating biological growth based on a feedback control system. They allow the user to specify the central axis and use it as an external stimulus that determines the growth direction. Onishi *et al.* proposed a tree-modeling system using 3D gestures [13] [14]. The user specifies the shape of dominant trunks using a 3D tracker, and then the system generates trunk models represented by a so-called *L-String,* which represents the initial structure of the standard L-system. Subsequently, the user attaches branches or leaves to the trunks using L-system. Both systems support the global control of the central axis of plants. However, these systems only modify the shape and do not control the growth of plants. Therefore, it is impossible for these systems to vary the depth of recursion depending on the user-specified axis.

Deussen and Lintermann developed the Xfrog system [4], which combines the power of a rule-based approach and intuitive user interfaces using a graph representation. Users design a graph representing the branching structures of a plant with 11 node types. This system offers an intuitive user interface and the resulting models are very realistic. However, their system does not support to control global shape directly and it is still difficult to design plants that have specific global appearances. Trees-Designer [12] is another commercial system for modeling plants. The user interface is based on supplying proxy geometry, such as boxes or quadrangles, to control the global appearance of trees. However, locating proxy geometry requires a traditional 3D CAD-like user interface that is substantially more difficult than sketching the central axis.

### 2.3 Sketch-Based Interface for 3D Modeling

In recent decades, sketch-based modeling has become popular; instead of creating precise, large-scale objects, a sketching interface provides an easy way to create a rough model that quickly conveys the user's intentions. SKETCH [24] allows users to design 3D scenes consisting of simple primitives, while Teddy allows users to design free-form models [6]. Generating 3D curves through sketching is also a rich research domain; Pentland and Kuo [15] generated a 3D curve from its 2D projection using energy minimization, while Ijiri *et al.* [7] used a constant curvature. Another strategy for defining a 3D curve is to draw strokes twice. For example, combining a screen projection of a curve with its shadow is useful for disambiguating the 3D position [3] [23].

Recently, some systems have applied sketching interfaces to plant modeling. Okabe *et al.* [11] presented a sketch-based tree modeling system. They infer the 3D geometry of a tree using the assumption that a tree tends to maximize the distance between its branches. Ijiri *et al.* [7] introduced a flower-modeling system that separates the modeling process into structure definition and geometry creation. They used floral diagrams and inflorescences for structure definition and sketch interfaces for geometry creation. Neither method used L-system. Although these techniques support the propagation or reuse of existing branches or leaves, the user needs to design a tree in detail manually. Maya Paint Effects [1] allows the user to modify the shape of a model by sketching its skeleton. Although this user interface offers an easy way to control the global shape of plants, it is essentially a deformation technique and does not simulate the growth of plants as our system does.

# 3 User Interface

Figure 1 shows a snapshot of our prototype system. The system consists of two panes; a control pane (left) and a creation pane (right). The user modifies generating rules in the control pane, and model trees in the creation pane. This section describes each of these user interfaces.

## 3.1 Manipulation of the Generating Rule and Its Parameters

The user begins modeling with specifying generating rules of L-system in the control pane. Our system provides predefined generating rules (Figure 3(i)) that contain three module types: top apex (Figure 3a), internode (Figure 3b), and lateral apex (Figure 3c). Currently, our system supports the definition of two rules only. In the future, we will design it to support an arbitrary number of rewriting rules. In this generating rule, the left rule replaces a top apex by another top apex, an internode, and lateral apices, while the right rule replaces a lateral apex with an internode and lateral apices. Although the number of lateral branches is specified in a dialog box, the other parameters are set by directly manipulating control points (Figure 3(iii)). The change in parameters is reflected in the visual representation of the rule immediately. The controllable parameters are the lateral branch length ratio, width ratio, twisting angle, and growth direction, and the top apex length ratio, width ratio, and twisting ratio. Here, "ratio" means the ratio between the current branch parameter and that of the parent branch. The top apex is different from modules in a standard L-system and its growth direction is determined in the subsequent generating process, described in the next section.

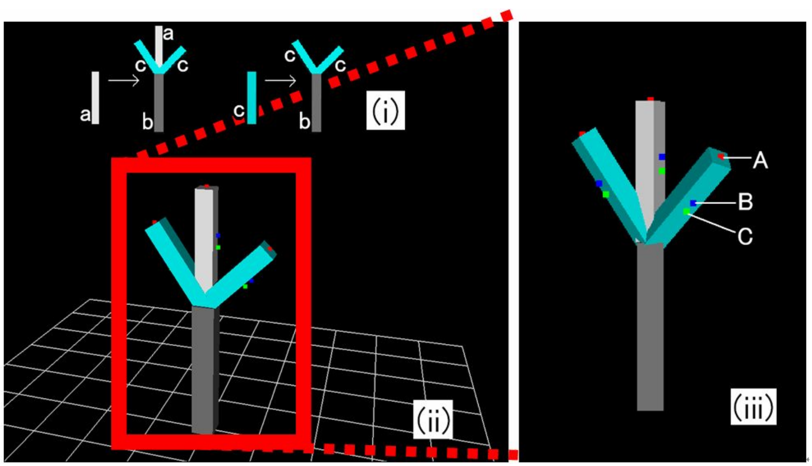



**Fig. 3.** A screenshot of the control pane. The generating rule is displayed above (i). The user modifies the parameters of the rule directly by manipulating control points of the handle (ii, iii). Control point "A" changes the orientation and the ratio of the branch lengths; "B" changes the twisting angle; and "C" changes the ratio of branch widths.

### 3.2 Geometry Creation by Drawing a Stroke

After setting the rules and parameters, the user creates tree geometry in the creation pane. The interface is very simple; the user draws a single stroke representing the central axis (Figure 4). The stroke corresponds to the top apex of the generating rule (Figure 3(i)). Since the top apex has a certain length, the system needs to resample the stroke.

When the user begins drawing a stroke, the system creates a straight line segment connecting the starting point and the current point. If the length of the line reaches a predefined value $l$, the system applies the rule using the line segment as a top apex (Figure 4(a)) and also updates the starting point of the line. Subsequently, when the line length reaches the value $l \times ratio$, the system reapplies the rule (Figure 4(b)). Here, "*ratio*" is the length ratio of the top apex determined in the previous process. As the user draws a stroke, the system repeats this process, growing the tree structure (Figure 4(d)–(h)). The length of the n-th line segment is $l \times ratio^{(n-1)}$. The growth of lateral branches is determined by the parameters specified in the previous process and is independent of the user drawn stroke. The system provides real-time feedback while the user is drawing a stroke.

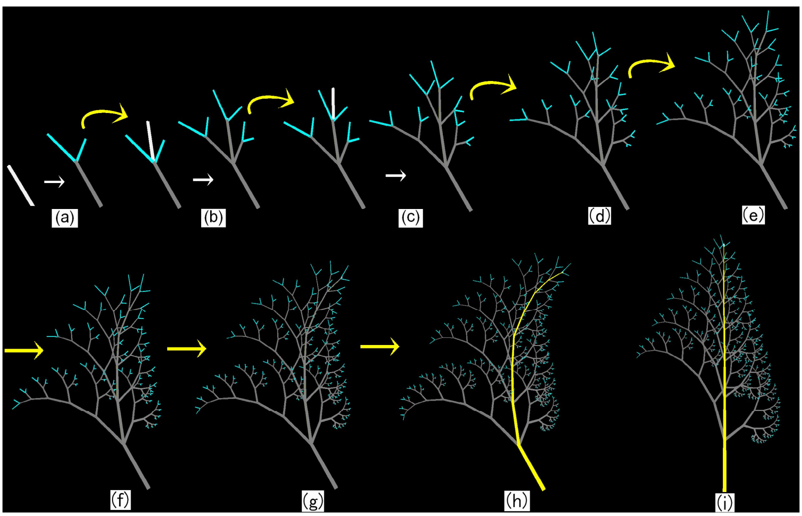



**Fig. 4.** Structure creation by drawing a stroke. The system interprets the user-dawn stroke (white bar) as the top apex, and applies a generating rule (a)–(c). Based on the stroke, the system generates the geometry (a)–(h). (i) The structure created by a straight stroke. In (h) and (i), the axis stroke is highlighted in yellow.

This framework allows the user to specify the depth of recursion, as well as the axis shape with a single stroke. If the user stops drawing when the stroke is short, the number of recursions is small and the resulting structure is thin. On the other hand, the longer the stroke is, the more iterations occur. To improve the visual appearance, the user can add a predefined leaf object on each terminal branch by pressing "tree button" (Figure 5).

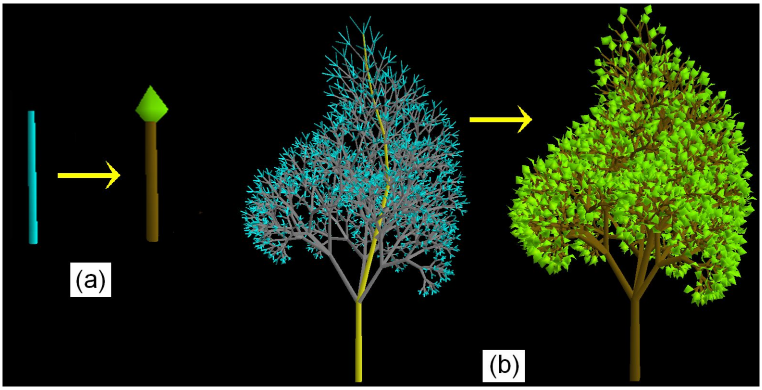



**Fig. 5.** Decorating a generated tree model. The system adds a leaf object to each edge of the terminal branch (a).

Our system also allows modifying the generating rule and parameters *after* constructing the whole geometry. The changes in the rule and parameters are immediately applied to the tree model. This enables the user to determine desirable parameters through intuitive trial-and-error session. This is often difficult in standard L-System framework where the parameter setting and the geometry construction are separated.

## 4 Discussion

In this paper, we have proposed a new tree-modeling system based on L-system. We introduced a special module to the generating rule of L-system whose growth direction is determined by a user-drawn stroke. Our system allows the user to control the global appearance and depth of recursion of the tree model easily and directly by drawing a single stroke.

Figure 1 and 6 show tree models designed using our system. Figures 6(f)–(h) show a variety of models created using a common rule set with different central strokes. Since our system provides a simple and easy-to-use interface, the user can complete each model in a few minutes. Furthermore, the data structure in our system is simple; each model is constructed by only two generating rules and one stroke. Therefore, tree models designed by our system will be easily exported to a standard L-System framework or to a general 3D modeling system. We believe that our system is useful not only for designing tree variations but also as a tool for artistic expression. A media artist, Professor Nagashima, developed a handheld device to interactively control the creation of trees in collaboration with us and created art installation combining our system and music [10].

The most important aspect of our approach is the ability to directly control the global shape by drawing a stroke, while the local structure only emerges as a result of applying local rules in standard L-system. In the future, we would like to combine our approach with other advanced sketch-based plant-modeling systems [7] [11]. Since our system currently supports only a specific set of generating rules, we would also like to extend our system to support a wider range of rules.

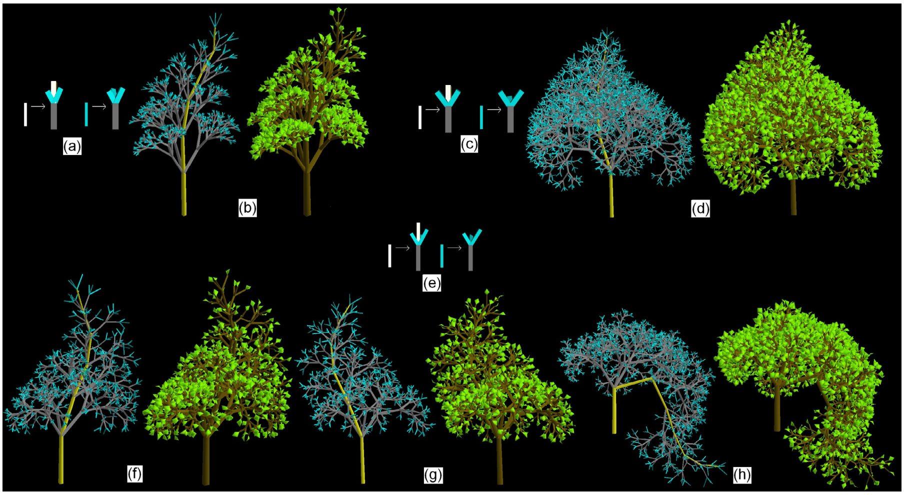



**Fig. 6.** Tree models and the corresponding generating rules. Each tree was designed in a few minutes using our system. Models (f)–(h) were created with the same rule (e).

## Acknowledgements

We thank Professor Yoichi Takebayashi for his comments and sage advice. We also thank Professor Yoichi Nagashima for demonstrating the potential of our system by applying it to his art. This work was funded in part by grants from the Japanese Information–Technology Promotion Agency (IPA) and JSPS Research fellow.

## References

[1] Autodesk and Alias. Maya. www.alias.co.jp
[2] Boudon, F., Prusinkiewicz, P., Federl, P., Godin, C., and Karwowski, R. Interactive Design of bonsai tree models. Computer Graphics Forum, 22, 3, 591–599. (Proc. Eurographics'03)
[3] Cohen, J., Markosian, L., Zeleznik, R., Hughes, J., and Barzel, R. An interface for sketching 3D curves. In Proceedings of ACM I3D 99, 17–21.
[4] Deussen, O., and Lintermann, B. 1999. Interactive modeling of plants. IEEE Computer Graphics and Applications, 19, 1, 56–65.
[5] Frijters, D., and Lindenmayer, A. A model for the growth and flowering of Aster novaeangliae on the basis of table (1,0) L-systems. In G. Rozenberg and A. Salomaa, editors. L Systems, Lecture Notes in Computer Science 15, 24–52. Springer–Verlag, Berlin, 1974.
[6] Igarashi, T., Matsuoka, S., and Tanaka, H. Teddy: a sketching interface for 3D freeform design. In Proceedings of ACM SIGGRAPH 99, ACM, 409–416.
[7] Ijiri, T., Owada, S., Okabe, M., and Igarashi, T. Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints. In Proceedings of ACM SIGGRAPH 2005, ACM, 720–726.
[8] Lindenmayer, A. Mathematical models for cellular interactions in development, I & II. Journal of Theoretical Biology, 280–315.

[9] Měch, R. and Prusinkiewicz, P. Visual models of plants interacting with their environment. In Proceedings of ACM SIGGRAPH 96, ACM, 397–410.

[10] Nagashima, Y. Nature system. IAMAS, inter-college computer music concert. December 2005 in Japan.

[11] Okabe, M., Owada, S., and Igarashi, T. Interactive design of botanical trees using freehand sketches and example-based editing. Computer Graphics Forum, 24, 3, C487–C496. (Proc Eurographics '05)

[12] Olas, P. TreesDesigner. www.polas.net/trees/

[13] Onishi, K., Hasuike, S., Kitamura, Y., and Kishino, F. Interactive modeling of trees by using growth simulation. Proc. of ACM Symposium on Virtual Reality Software and Technology, 66–72, 2003.

[14] Onishi, K., Murakami, N., Kitamura, Y., and Kishino, F. Modeling of trees with interactive L-system and 3D gestures. Lecture Notes in Computer Science (Proc. of BioADIT 2006), 3853, 222–235, Jan. 2006.

[15] Pentland, A., and Kuo, J. The artist at the interface. Vision and Modeling Technical Report 114, MIT Media Lab.

[16] Prusinkiewicz, P., and Lindenmayer, A. The Algorithmic Beauty of Plants. Springer–Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.

[17] Prusinkiewicz, P., Hammel, M., Hanan, J., and Měch, R. L-systems: from the theory to visual models of plants. In Proceedings of the 2nd CSIRO Symposium on Computational Challenges in Life Sciences, 1996.

[18] Prusinkiewicz, P., Hanan, J., and Měch, R. An L-system-based plant modeling language. Lecture Notes in Computer Science 1779, 395–410. Springer–Verlag, Berlin, 2000.

[19] Prusinkiewicz, P., James, M., and Měch, R. Synthetic topiary. In Proceedings of ACM SIGGRAPH 94, ACM, 351–358.

[20] Prusinkiewicz, P., Karwowski, R., Měch, R., and Hanan, J. Lstudio/cpfg: a software system for modeling plants, 2000. Lecture Notes in Computer Science 1779, 457–464. Springer–Verlag, Berlin, 2000.

[21] Prusinkiewicz, P., Mündermann L., Karwowski, R., and Lane, B. The use of positional information in the modeling of plants. In Proceedings of ACM SIGGRAPH 2001, 289–300.

[22] Streit, L., Federl, P., and Sousa M.C. Modelling plant variation through growth. Computer Graphics Forum, 24, 3, C487–C596. (Proc. Eurographics '05)

[23] Tobita, H., and Rekimoto, J. Flat3D: a shared virtual 3D world system for creative activities and communication. IPSJ, 44, 2, 245–255 (in Japanese).

[24] Zeleznik, R.C., Herndon, K.P., and Hughes, J.F. SKETCH: an interface for sketching 3D scenes. In Proceedings of ACM SIGGRAPH 96, ACM, 163–170.

# Through-the-Lens Cinematography

Marc Christie[1] and Hiroshi Hosobe[2]

[1] LINA - Computer Science Laboratory
FRE CNRS 2729 - University of Nantes
2, Rue de la Houssinière, F-44300 Nantes, France
marc.christie@lina.univ-nantes.fr
[2] National Institute for Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
hosobe@nii.ac.jp

**Abstract.** This article presents an extension of the Through-the-Lens Camera Control approach proposed by Gleicher and Witkin. It first provides a higher means of control on the camera by using virtual composition primitives and second offers a means for through-the-lens interaction with both the location of the objects in the scene and the lighting. By setting properties on the composition primitives, users convey constraints to enforce the positioning of the camera, the objects and the lights directly through the lens. The paper presents how to express all three problems of indirect camera, object and light interaction in a consistent way and provides some first results. The solving techniques rely on the expression of the image Jacobian coupled with a constrained optimizer based on Quadratic Programming. The Jacobian expresses the relation between the user input and the possible degrees of freedom on the entity to manipulate; in order to avoid solving failures that are delicate to manage in user interfaces, we propose a mass-spring interaction model. As a result, the user should be able concentrate on higher level properties such as composition, balance and unity through a natural and effective interaction process.

## 1 Introduction

Virtual Cinematography can be defined as a task that encompasses the modelling and the positioning of objects in a 3D scene, as well as the positioning of the lights and the virtual cameras. Classically, we rely on interactive modelling tools that utilize basic and generally direct interaction metaphors provided by the well-known 4-split view. Such interaction metaphors can present some difficulties when manipulating i/ the objects, ii/ the camera and iii/ the lights in the 3D scene.

First, if we consider object manipulation, the input provided by a 2D device (the mouse) must be mapped to the six degrees of freedom (*d.o.f*) of the object (or more if one considers articulated rigid bodies). Classically, such a mapping is expressed as a direct assignment of the object parameters with the mouse inputs. Some interaction metaphors (such as arcballs or virtual axis) do assist the manipulation but remain based on the mathematical representation of the

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 147–159, 2006.

manipulations (rotations and translations). Some tasks such as slightly rotating an object while maintaining one of its corner at a fixed position on the screen can only be performed by a tedious generate and test process. The limitation of the approach actually lays in that the applications try to transform the 2D input into a single 3D primitive manipulation (rotation or translation) and do not consider the properties of the projected image. A possible improvement is to consider setting some constraints on points or properties of the objects in the screen and directly manipulate some other points or properties on the screen.

Second, controlling a camera in a virtual environment presents some difficulties too. A quick glance at the tools proposed by most 3D modellers to build and to edit camera paths in virtual environments shows that all follow a similar approach. A virtual camera is classically set up by providing a 3D point which represents the location of the camera and a vector that represents the *look at* direction of the camera. The animation of the camera relies on classical interpolation methods such as spline curves with keyframes and/or control points. However these tools are not adapted to a fine control over the composition of the screen. A possible improvement is to edit directly the locations of some objects on the screen while maintaining some high-level properties.

Third, the lighting of a scene can be quite a delicate task as the user knows how he wants the scene to be lit and must provide some low-level light coordinates (location/orientation). Consequently the user directly manipulates the light parameters in order to converge towards his mental representation. A possible improvement is to offer indirect manipulation of the lights by considering shadows and specular reflections as interaction metaphors, and manipulating them in order to indirectly update the light parameters.

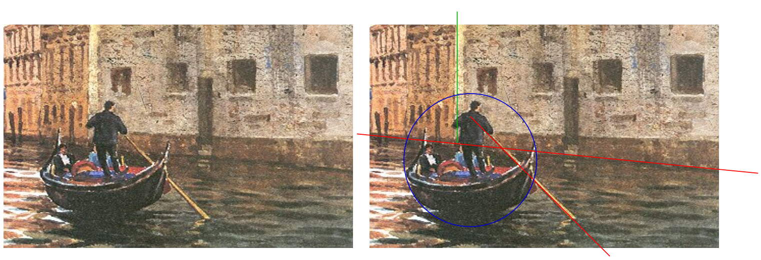


**Fig. 1.** Construction shapes on a painting

In this paper, we propose both an interaction metaphor and a common solving process to directly manipulate the entities relative to virtual cinematography that are objects, cameras and lights. We refer to this work as the Through-The-Lens Cinematography approach after Gleicher and Witkin work [4]. We propose virtual composition shapes such as leading lines, curves and focal points to easily manipulate these entities directly through the screen and rely on the image Jacobian to produce a quadratic constraint satisfaction problem. Figure 1 presents an original painting with some construction shapes. One can for example manipulate the end of the leading horizontal line to change the camera's height and orientation while maintaining the boat projected in the circle, or displace the circle to move the camera correspondingly.

In a first section, we recall the approaches based on the image Jacobian to manipulate a camera. We then identify how such work can be extended to manipulate lights and objects, with high-level interaction paradigms such as virtual construction shapes. The following section presents the linear solving process where the user input is considered as soft constraints and existing relations to be enforced as hard constraints. Finally, we present a first implementation and some results.

## 2 State of the Art

### 2.1 Camera Manipulation

Camera control consists in assisting the computation of the camera parameters considering different degrees of automation, from totally manual where the user's input parameters are directly mapped to the camera degrees of freedom upto completely automated where high-level properties constrain the possible locations and orientations of the camera. A short survey of camera control techniques can be found in [3].

In this work, we concentrate on interactive control systems. Such systems provide the user with a view of the model world and update the camera setups in response to user inputs. This naturally raises the question of how the user input devices will map onto the camera parameters. Ware and Osborne published a review of possible mappings [18] which they referred to as camera control metaphors:

- camera in hand: the camera is directly manipulated as if it were in the user's hand; this comprises rotational and translational movements.
- world in hand: the camera swivels around the world while shooting a fixed point – generally the center of the world. A left movement on the user's device corresponds to a right movement of the scene.
- flying vehicle: the camera is considered as a plane. The metaphor is intuitive and has been extensively used to explore large environments.

A new metaphor can be added to this list which is the *walking* metaphor [5]; the camera moves in the environment while keeping at a constant height from the floor.

The *world in hand* metaphor is restricted to explore details of an object or a group of objects. An interactive control was developed by Phillips *et al.* [14] for the human figure modelling system *Jack*. However intuitive the *Jack* system could not properly manipulate model figures about axes parallel or perpendicular to the view direction. The camera control method prevented this occurring by repositioning the camera.

Shoemake[16] introduces the concept of arcball, a virtual ball that contains an object to manipulate. He proposes to stabilize the computation with the use of quaternions to avoid Euler singularities while rotating around an object (gimbal lock issue). The metaphor is still in use in many modelling tools. Other possible mappings are reported in Chen *et al.* 's work [2] which provides a specific study of 3D rotations from 2D input devices. In [7], Khan *et al.* propose an interaction technique for proximal object inspection that relieves the user from most of the camera control. Basically the approach tries to maintain the camera at a fixed distance around the object and relatively normal to the surface obeying a hovercraft metaphor. A somehow similar approach has been proposed by Burtnyk *et al.* in [1] in which the camera is constrained to a surface defined around the object to explore as in [5]. However the surfaces are restricted to interesting view points of the object that guarantee a certain level of quality in the users exploration experience.

The *flying vehicle* metaphor has received a number of contributions mostly guided by the applications. The major difficulty lays in avoiding the *lost in space* problem encountered when the user has to manage an important number of degrees of freedom. Consequently most of the work concentrates on assisting the control of the camera parameters through techniques such as physically-based models, vector fields or path planning to fix main directions and to avoid obstacles.

Whilst the approaches we mention provide efficient means to move a camera in large environments or to offer proximal exploration of objects, these remain inadequate for tasks related to composition purposes.

The approach that has directly inspired our work, and that can be viewed as an interesting tool for composition is the original *Through-The-Lens Camera Control* approach devised by Gleicher and Witkin [4] that allows the user to control a camera by manipulating the locations of objects directly on the screen. A re-computation of new camera parameters is done to match the user's desired locations. The difference between the actual screen locations and the desired locations input by the user is treated as a velocity. The authors then state the relationship between the velocity ($\dot{\mathbf{h}}$) of $m$ displaced points on the screen and the velocity ($\dot{\mathbf{q}}$) of the camera parameters through the Jacobian matrix that represents the perspective transformation:

$$\dot{\mathbf{h}} = J\dot{\mathbf{q}}$$

The authors propose to solve the non-linear optimization problem which minimizes a quadratic energy function $E = \frac{1}{2}(\dot{\mathbf{q}} - \dot{\mathbf{q}}_0) \cdot (\dot{\mathbf{q}} - \dot{\mathbf{q}}_0)$ that represents a minimal move in the camera parameters ($\dot{\mathbf{q}}_0$ representing the values of the camera's previous velocity). This problem can be converted into a Lagrange equation and solved for the value of $\lambda$:

$$\frac{dE}{d\dot{\mathbf{q}}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_0 = J^T\lambda$$

where $\lambda$ stands for the vector of Lagrange multipliers. The velocity of the camera parameters is thus given by:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_0 + J^T\lambda$$

A simple Euler integration allows to approximate the next location of the camera from the velocity $\dot{\mathbf{q}}$:

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \Delta t \; \dot{\mathbf{q}}(t)$$

The result is that the rate of change of the camera set-up is proportional to the magnitude of the difference between the actual screen properties and the desired properties set by the user.

This work has been improved and extended by Kung, Kim and Hong in [8] with the use of a single Jacobian Matrix. A pseudo inverse of the matrix is computed with the Singular Value Decomposition (SVD) method which complexity is in $O(m)$. Stability and efficiency are both augmented. The SVD method enjoys the following property that the pseudo inverse always produces a solution with the minimal norm on the variation of $\mathbf{q}$.

A similar approach is proposed by [10] in that the techniques rely on the expression of the image Jacobian. The control of the camera is dedicated to the tracking of mobile targets in the environment while satisfying secondary cinematographic tasks and but does not allow user interactions. The principle consists in specifying the target tracking task as the regulation in the image of a set of visual features. Let $\mathbf{P}$ be the set of visual features used in the visual servoing task. To ensure the convergence of $\mathbf{P}$ to its desired value $\mathbf{P_d}$ we need to know the interaction matrix (the image Jacobian) $\mathbf{J}$ that links the motion of the object in the image to the camera motion $\dot{\mathbf{q}}$:

$$\dot{\mathbf{P}} = J\dot{\mathbf{q}} \tag{1}$$

where $\dot{\mathbf{P}}$ is the time variation of $\mathbf{P}$ (the motion of $\mathbf{P}$ in the image) due to the camera motion $\dot{\mathbf{q}}$. If the primary task (following the object) does not instantiate all the camera parameters when solving equation 1, secondary tasks may be added (avoiding obstacles or occlusions, lighting optimization, *etc.* ). Since visual servoing consists in positioning a camera according to the information perceived in the image, the task is specified in a 2D space, while the resulting camera trajectories are in a 3D space. Such approaches are computationally efficient and thus suitable for highly dynamic environments. However those approaches are not free from drawbacks; one cannot determine in advance which degrees of freedom of the camera and how many of them will be instantiated by the primary task. Consequently no guarantee can be given on the satisfaction of secondary tasks.

### 2.2 Object Manipulation

Very few approaches have considered a through-the-lens manipulation of object parameters mainly due the problem of providing a good mapping function. Classical approaches rely on interaction metaphors closely related to the modelling of geometric transformations (local arcballs for rotations, virtual axes for translations). Little to no prior work exists in automated object repositioning which is an interesting direction in the field of automated cinematography. We report He *et al.* work [6] that propose to reposition characters to improve the quality of a shot.

### 2.3 Light Manipulation

A large number of approaches has considered the problem of automated lighting, *i.e.* computing the light parameters from a set of desired properties in the scene (*e.g.* to improve saliency or recognizability). Contrarily, indirect light manipulation encompasses interactive applications that model the light parameters through the interactive manipulation of the lighting effects. Research on indirect light manipulation has received little attention. The problem is however interesting in that it represents a natural way to specify lightings by describing its characteristics and effects on the objects in the scene (which objects are lit, which are shadowed, where are the specular reflections). The first contribution that considers interactive lighting is reported in [15] whereby manipulation of the wireframe-drawn shadows is used to edit the light parameters. An approach proposed by Pellacini *et al.* [12] considers different through-the-lens screen metaphors to design shadows. The approach encompasses interactive moving of the shadows and recomputing either the location of the light or of the occluder, updating the shadow softness, scaling the shadow or moving the light's hotspot. An interesting feature is added in that the user can set constraints which are maintained during the interaction process. Examples of such constraints are encountered when a user wants to maintain an area lit or shadowed. Simple algebraic relations allow to update the camera parameters directly from the mouse locations, depending on the interaction modes. Constraints are enforced by a simple evaluation process that checks if the violation of a relation occurs. A more recent paper by Pellacini *et al.* [13] reports the use of these techniques in a near realtime relighting engine for computer cinematography, but concentrates more on the rendering technique than on the interaction metaphors for lighting. The authors however denote that further innovation in this area is expected.

A contribution by Courty *et al.* [11] is quite noticeable in that it reports a technique to control a virtual camera that can be used to provide light and character editing through-the-lens with a technique based on virtual servoing.

## 3 Through-the-Lens Cinematography

In this paper we propose to encompass the three steps of camera, object and light manipulation in the same process as all are intrinsically linked. We propose to rely on i/ virtual composition primitives, ii/ a mass-spring interaction model and iii/ image Jacobian coupled with a quadratic solving process.

### 3.1 Virtual Composition Primitives

In order to ease the interaction process, we rely on the natural interaction metaphors that are virtual composition primitives used by artists when sketching a piece before painting or that are revealed in photography and cinema. We propose to use these primitives to update the locations of the camera, the objects in the scene, and the lights. The primitives can link multiple points from a single object or from a set of objects. We consider:

- virtual 2D points to control real projections of 3D points,
- virtual 2D lines defined between real projections of two (or more) 3D points; a good example is perspective or leading lines that guide the spectators gaze,
- virtual circles/ellipses on which projection of 3D points must be maintained,
- virtual spline curves to which a set of 3D projected points can be attached.

Examples are presented in Figure 2.

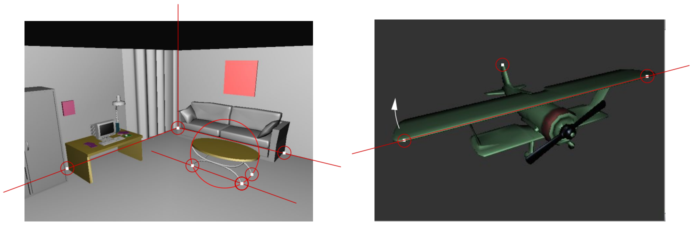


**Fig. 2.** Point and Line, and circle interaction primitives. Each parameter of each primitive can be constrained by the user or set free.

### 3.2 A Mass-Spring Interaction Paradigm

We then propose an interaction paradigm based on a mass-spring system: the user manipulates the primitives which are virtually attached to the projected images of the points in the scene by springs. This allows for example to maximally include a non spherical object inside a circle primitive by linking some of the points to the circle. Playing on the mass of the primitives assists the user in specifying the force of his action on the environment. Such a metaphor helps to overcome simple limitations that occur when the constraints are too strong and happen to prevent any modification. The force applied to each 2D point $p$ is expressed as the sum of all forces applied on it (Hooks law):

$$F_p = -\sum_i K_i(l_i - l_i^0)\frac{l_i}{||l_i||}$$

where $K_i$ is the damping of the $i$-th spring, $l_i$ its current length and $l_i^0$ its length at rest. A simple integration step then computes the velocity $v_p$ of each point $p$ which is the input of the solving process:

$$\begin{cases} a_p(t+\Delta t) = \frac{1}{m}F_p(t) \\ v_p(t+\Delta t) = v_p(t) + \Delta t.a_p(t+\Delta t) \end{cases}$$

### 3.3 Modelling the Camera Interaction

In order to compute the camera parameters that correspond to the user input, we rely on the expression of the image Jacobian. Lets derive the equations related to the projection of a single point on the screen. All other primitives (line, circle) can be extended similarly. The relation is modelled with a quaternion-based camera in order to avoid such singularities as gimbal locks. Let $\mathbf{q}$ denote the camera parameter vector $\left[c_x\ c_y\ c_z\ q_w\ q_x\ q_y\ q_z\ c_\gamma\right]^T \in R^8$ where $(c_x, c_y, c_z)$ is the camera location, $(q_w, q_x, q_y, q_z)$ the unit quaternion for that represents the camera rotation and $c_\gamma$ the focal length. The projection of a 3D point $\mathbf{x}$ on a 2D screen point $\mathbf{x}'$ is expressed by the following relation:

$$\mathbf{x}' = V_\mathbf{q}.\mathbf{x} = P(c_\gamma).R(q_w, q_x, q_y, q_z).T(c_x, c_y, c_z).\mathbf{x}$$

where $P(c_\gamma)$ represents the perspective projection, matrixes $R(q_w, q_x, q_y, q_z)$ and $T(c_x, c_y, c_z)$ represent the change of basis between the world coordinates and the camera coordinates.

In this work, we follow the approach proposed by [4] and model the relation with the image Jacobian: a variation of the camera parameters $\mathbf{q}$ enforces variations on the projected point $\mathbf{x}'$ through the matrix of partial derivates $J$:

$$\begin{aligned} &\dot{\mathbf{x}}' = J\dot{\mathbf{q}} \\ &\text{where } J = \left( \frac{\partial V_\mathbf{q}}{\partial c_x}\ \frac{\partial V_\mathbf{q}}{\partial c_y}\ \frac{\partial V_\mathbf{q}}{\partial c_z}\ \frac{\partial V_\mathbf{q}}{\partial q_w}\ \frac{\partial V_\mathbf{q}}{\partial q_x}\ \frac{\partial V_\mathbf{q}}{\partial q_y}\ \frac{\partial V_\mathbf{q}}{\partial q_z}\ \frac{\partial V_\mathbf{q}}{\partial c_\gamma} \right) \end{aligned} \qquad (2)$$

From there, the user provides the variations of the the projected points $\mathbf{x}'$ by direct $2D$ manipulation on the screen (position variation is considered here as a velocity computed through the Euler integration). Previous approaches propose to compute the pseudo inverse of the Jacobian in order to express the camera parameters *w.r.t.* $\dot{\mathbf{x}}'$. In our approach, we consider solving the simple linearized system formed by equation 2, while minimizing the variation on the camera parameters. A simplified expression of the Jacobian $J$ is provided by the excellent work of Kung *et al.* [8]:

$$J = \begin{pmatrix} \frac{\gamma R_{11}}{\hat{z}} - \frac{\gamma\hat{x}R_{31}}{\hat{z}^2} & \frac{\gamma R_{12}}{\hat{z}} - \frac{\gamma\hat{x}R_{32}}{\hat{z}^2} & \frac{\gamma R_{13}}{\hat{z}} - \frac{\gamma\hat{x}R_{33}}{\hat{z}^2} & -\frac{\gamma\hat{x}\hat{y}}{\hat{z}^2} & \gamma + \frac{\gamma\hat{x}^2}{\hat{z}^2} & -\frac{\gamma\hat{y}}{\hat{z}} & \frac{\hat{x}}{\hat{z}} \\ \frac{\gamma R_{21}}{\hat{z}} - \frac{\gamma\hat{y}R_{31}}{\hat{z}^2} & \frac{\gamma R_{22}}{\hat{z}} - \frac{\gamma\hat{y}R_{32}}{\hat{z}^2} & \frac{\gamma R_{23}}{\hat{z}} - \frac{\gamma\hat{y}R_{33}}{\hat{z}^2} & -\gamma - \frac{\gamma\hat{x}^2}{\hat{z}^2} & \frac{\gamma\hat{y}\hat{x}}{\hat{z}^2} & \frac{\gamma\hat{x}}{\hat{z}} & \frac{\hat{y}}{\hat{z}} \end{pmatrix}$$

where $(\hat{x}, \hat{y}, \hat{z})$ represents the change from global to local basis of $\mathbf{x}$: $(\hat{x}, \hat{y}, \hat{z})^T = R(q_w, q_x, q_y, q_z).T(c_x, c_y, c_z).\mathbf{x}$ and where $R_{ij}$ represents the $i-$th row and $j-$th column of rotation matrix $R(q_w, q_x, q_y, q_z)$. One can refer with profit to [9] for a detailled presentation of the simplification of $J$, and in particular the use of the exponential map to maintain the quaternion on the unit circle (a problem that occurred in [4]).

### 3.4 Modelling the Object Interaction

In order to manipulate the objects directly in the screen, we consider a similar formulation. The unknowns are now the vector $\mathbf{o} = \left[t_x\ t_y\ t_z\ s_w\ s_x\ s_y\ s_z\right]^T$ of parameters of the object ($t$ for translation and quaternion $s$ for rotation) in the 3D world. The relation is expressed as:

$$\mathbf{x}' = F(\mathbf{q}, \mathbf{o}).\mathbf{x} = V_\mathbf{q}.T(t_x, t_y, t_z).R(s_w, s_x, s_y, s_z).\mathbf{x}$$

where $\mathbf{x}$ is the coordinate of a 3D point in the local basis of the object $\mathbf{o}$, translation $T(t_x, t_y, t_z)$ and rotation $R(s_w, s_x, s_y, s_z)$ denote the transformations applied to this object. The expression of the Jacobian is therefore provided by:

$$\dot{\mathbf{x}}' = J_o.\dot{\mathbf{p}}$$
$$\text{where } J_o = \left( \frac{\partial F_{\mathbf{q},\mathbf{o}}}{\partial t_x} \; \frac{\partial F_{\mathbf{q},\mathbf{o}}}{\partial t_y} \; \frac{\partial F_{\mathbf{q},\mathbf{o}}}{\partial t_z} \; \frac{\partial F_{\mathbf{q},\mathbf{o}}}{\partial s_w} \; \frac{\partial F_{\mathbf{q},\mathbf{o}}}{\partial s_x} \; \frac{\partial F_{\mathbf{q},\mathbf{o}}}{\partial s_y} \; \frac{\partial F_{\mathbf{q},\mathbf{o}}}{\partial s_z} \right) \tag{3}$$

### 3.5 Modelling the Light Interaction

The vector of light parameters to consider here is $\mathbf{l} = [l_x \; l_y \; l_z \; s_w \; s_x \; s_y \; s_z]^T$ (location and orientation of a light). In the sequel, we express the user-interaction with the specular reflection of the light source on a surface. A move of the specular reflection on the screen leads to a change in the light parameters. Let $\mathbf{x}'$ be the location on the screen of the specular reflection at position $\mathbf{x}$ on a given surface. The expression of $\mathbf{x}'$ is provided by:

$$\mathbf{x}' = L(x, \mathbf{q}).l$$

In this process, the location of $\mathbf{x}$ must be recomputed for each modification of $\mathbf{x}'$ through a simple ray cast and surface normal evaluation. Phong's model of specular reflection is given by:

$$L_v = k_p(\mathbf{v}.\mathbf{r})^p L_s$$

where $\mathbf{v}$ represents the lookat vector of the camera, $\mathbf{r}$ the direction of the reflected ray (depends on the surface normal $\mathbf{n}$ and the incident ray $\mathbf{s}$ see Fig. 3).

The expression on the velocities is written as:

$$\dot{\mathbf{x}}' = J_l.\dot{\mathbf{l}}$$
$$\text{where } J_l = \left( \frac{\partial L_{\mathbf{q}}}{\partial t_x} \; \frac{\partial L_{\mathbf{q}}}{\partial t_y} \; \frac{\partial L_{\mathbf{q}}}{\partial t_z} \; \frac{\partial L_{\mathbf{q}}}{\partial s_w} \; \frac{\partial L_{\mathbf{q}}}{\partial s_x} \; \frac{\partial L_{\mathbf{q}}}{\partial s_y} \; \frac{\partial L_{\mathbf{q}}}{\partial s_z} \right)$$

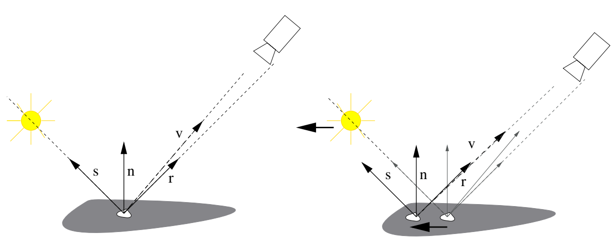



**Fig. 3.** Specular light reflection model

## 4 Constraint Solving

Each interaction process provides a linear system to be solved:

$$A.\mathbf{x} = \mathbf{b}$$

where $\mathbf{b}$ is the user input (the velocity of the manipulated points computed by the mass-spring interaction metaphor), $A$ the Jacobian Matrix computed at the current configuration, and $\mathbf{x}$ the vector of unknowns. In most cases the constraint system is not square. Whenever the system possesses more variables than constraints, an optimisation process minimizes an energy function $E_1$ that is expressed as a least square function on the degrees of freedom thereby enforcing minimal variation on the parameters (as in [4]).

In order to avoid inconsistent systems, we translate the user input as a function to minimize by considering the distance from the current configuration to the desired configuration. The points that are not manipulated by the user can be considered as constraints (that can be activated or not). The system to be solved is therefore expressed as:

$$\begin{array}{ll} \text{minimize} & (1-\alpha)E_1(\mathbf{x}) + \alpha E_2(\mathbf{x}) \\ \text{subject to} & A_2.\mathbf{x} = 0 \end{array}$$

where $E_2(\mathbf{x}) = \frac{1}{2}(A_1.\mathbf{x} - \mathbf{b}).(A_1.\mathbf{x} - \mathbf{b})$ expresses the minimal variation between the user input $\mathbf{b}$ and the projected parameters $A_1.\mathbf{x}$. Matrix $A_1$ represents the Jacobian related to the user manipulation and matrix $A_2$ is related to the user-defined constraints (*e.g.* when pin-pointing a point on the screen).

When the user constrains a point on the screen, its velocity is null which leads to expression $A_2.\mathbf{x} = 0$. The real parameter $\alpha$ allows to balance the influence of each energy function. We rely on a quadratic solving process that manages linear systems for the constraints and quadratic functions for the optimization functions. The solver we use is LOQO [17] which is based on an infeasible interior point method applied to a sequence of quadratic approximations to the problem.

## 5 First Implementation and Tests

At this time, a very first prototype has been devised that features the manipulation of the camera (location, orientation, focal distance) and the objects (location, orientation). The light interaction process has not yet been included. Our prototype implements simple construction primitives such as points, lines and circles. Each primitive can be set free or constrained.

In order to check the feasibility of our approach on manipulating the camera, we propose three possible interactions on a simple 3DS plane model (see Fig. 4).

1. we pin point one end of the wing, and interact with the other end. This problem is globally under-constrained. The minimization process reduces the variations on the degrees of freedom.

2. we align three points of the plane (upper wing, stabilizer, lower wing) and manipulate any other point to move the plane along the axis
3. we make three points of the plane belong to a circle, and manipulate one of the points while maintaining others on the circle.

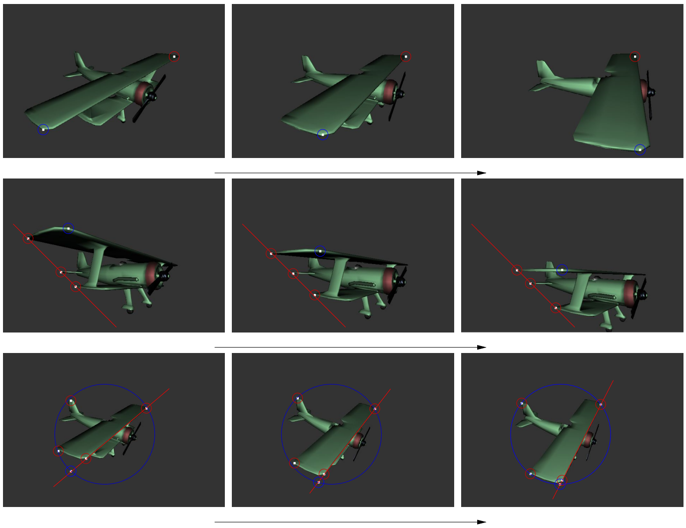


**Fig. 4.** Manipulating the camera by moving the projected points on the screen while maintaining some constraints

The quality of the interaction is highly related on the time spent in the numerical solving process. The process remains stable when slight modifications on the screen are encountered. The greater the norm of the velocity input, the more often the Jacobian must be evaluated (to reduce the linear approximation) and the more expensive the solving process is. On our three simple interaction scenarios, computation times are quite acceptable (more than 10fps on a Linux OS, 1.7GHz, 512 Mo).

## 6 Ongoing and Future Work

The extensions of our current work will follow two axes. First considering the solving process, we do not offer as efficient approaches as those proposed by Gleicher *et al.* [4] due to generality. The linearizing of the problem at the current camera configuration requires the re-computation of the Jacobian for each modification. We intend to include the expression of the Jacobian directly in the set of constraints. The process will be sensibly more expensive but much more precise for higher velocities.

Second, our modelling of the light and object interaction is quite insufficient compared to other specific approaches. More precise interaction models need to be provided and implemented for managing complex shadows and diffuse reflections. Such issues raise essential questions on the interaction metaphors to consider.

Finally, our approach has not been validated by any user study. Once the modelling issues will be improved, we will propose a user study by the computer graphics students of Nantes University.

## 7 Conclusion

In this article, we have proposed a through-the-lens approach to manipulate the three major components of virtual cinematography, namely the object, the camera, and the lights positions. Whilst each component has separately received attention from the computer graphics community, we propose here to integrate all three in the same approach and we propose virtual construction shapes as interaction metaphors coupled with a mass-spring interaction paradigm for managing the user-input. Some first results are presented on camera control. The authors believe that the proposed framework (interaction and solving) is an effective way of achieving fine control over the content of a screen, by assisting the users in enforcing high level properties such as composition, balance and unity.

## Acknowledgments

The authors are grateful to Patrick Olivier for his insightful comments on preliminary discussions that have led to this work.

## References

1. R. Burtnyk, A. Khan, G. Fitzmaurice, R. Balakrishan, and G. Kurtenbach. Stylecam: Interactive stylized 3d navigation using integrated spatial & temporal controls. In *ACM IUST Symposium on User Interface Software & Technology*, 2002.
2. M. Chen, S.J. Mountford, and A. Sellen. A study in interactive 3d rotation using 2d input devices. In Edwin E. Catmull, editor, *Computer Graphics (Proceedings SIGGRAPH '88)*, volume 22-4, pages 121–130, August 1988.
3. Marc Christie, Rumesh Machap, Jean-Marie Normand, Patrick Olivier, and Jonathan Pickering. Virtual camera planning: A survey. In *Smart Graphics*, pages 40–52, 2005.
4. M. Gleicher and A. Witkin. Through-the-lens camera control. In *Proceedings of ACM SIGGRAPH'92*, pages 331–340, 1992.
5. A. Hanson and E. Wernert. Constrained 3d navigation with 2d controllers. In *IEEE Visualization*, pages 175–182, 1997.
6. Li-wei He, M. F. Cohen, and D. H. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 217–224. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

7. Azam Khan, Ben Komalo, Jos Stam, George Fitzmaurice, and Gordon Kurtenbach. Hovercam: interactive 3d navigation for proximal object inspection. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pages 73–80, New York, NY, USA, 2005. ACM Press.
8. M. H. Kyung, M. S. Kim, and S. Hong. Through-the-lens camera control with a simple jacobian matrix. In *Proceedings of Graphics Interface '95*, pages 117–178, 1995.
9. Min-Ho Kyung, Myung-Soo Kim, and Sung Je Hong. A new approach to through-the-lens camera control. *Graphical models and image processing: GMIP*, 58(3):262–285, 1996.
10. E. Marchand and N. Courty. Image-based virtual camera motion strategies. In S. Fels and P. Poulin, editors, *Graphics Interface Conference, GI'00*, pages 69–76, Montreal, Quebec, May 2000. Morgan Kaufmann.
11. E. Marchand and N. Courty. Controlling a camera in a virtual environment. *The Visual Computer Journal*, 18(1):1–19, 2002.
12. Fabio Pellacini, Parag Tole, and Donald P. Greenberg. A user interface for interactive cinematic shadow design. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 563–566, New York, NY, USA, 2002. ACM Press.
13. Fabio Pellacini, Kiril Vidimce, Aaron Lefohn, Alex Mohr, Mark Leone, and John Warren. Lpics: a hybrid hardware-accelerated relighting engine for computer cinematography. *ACM Trans. Graph.*, 24(3):464–470, 2005.
14. C. B. Phillips, N. I. Badler, and J. Granieri. Automatic viewing control for 3d direct manipulation. In *Proceedings of the 1992 symposium on Interactive 3D graphics*, pages 71–74. ACM Press New York, NY, USA, 1992.
15. Pierre Poulin and Alain Fournier. Lights from highlights and shadows. In *Proceedings of the 1992 Western Computer Graphics Symposium*, pages 141–145, Banff, AB, Canada, 6–8 April 1992.
16. K. Shoemake. Arcball: a user interface for specifying three-dimensional orientation using a mouse. In *Proceedings of Graphics Interface '92*, pages 151–156, May 1992.
17. R. J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Methods and Software*, 11:451–484, 1999.
18. C. Ware and S. Osborne. Exploration and virtual camera control in virtual three dimensional environments. In *SI3D '90: Proceedings of the 1990 symposium on Interactive 3D graphics*, pages 175–183, New York, NY, USA, 1990. ACM Press.

# Explorations in Declarative Lighting Design

Hai Nam Ha and Patrick Olivier

Informatics Research Institute
University of Newcastle Upon Tyne
Newcastle Upon Tyne, NE1 7RU, UK
h.n.ha@ncl.ac.uk, p.l.olivier@ncl.ac.uk

**Abstract.** Declarative approaches to lighting design model image quality using an objective function that captures the desired visual properties of an object or scene. The value of the objective function is optimized for a particular camera configuration through the manipulation of the lighting parameters of a scene. We review the notion of declarative lighting design, and introduce LightOp, a tool by which the design of objective functions (the components and settings) and the application of different optimization techniques can be explored. We show how LightOp can be used to explore declarative lighting design through the realization of a number of extensions to existing approaches, including the application and evaluation of stochastic optimization; the use of backlighting to maximize edge enhancement; contrast modeling; and the use of a perceptually uniform color space.

## 1 Introduction

Lighting design is the problem of finding optimal lighting parameters – positions, directions, colors, and intensities of light sources – in order to maximize the perceptual quality of a specific 3D scene model. Effective lighting can convey much information about a scene and the automatic generation of images with highly discernable visual characteristics is a pressing requirement in application domains such as scientific visualization in which the components of a scene and there relative configuration with respect to the viewers cannot be anticipated in advance. Lighting design is a crucial stage in computer-generated image synthesis and movie making. A lighting design tool is expected to support expert and non-expert users in terms of a reduction in design time and maximization of the visual quality of generated images. Insights from studies of visual cognition are increasingly applied to computer graphics. Indeed, much research has been conducted with the aim of narrowing the gap between the perception of real and computer-generated imagery [McN00][FP04]. Realism in a computer-generated image is not only a matter of physical correctness, but also of perceptual equivalence of the image to the corresponding real world scene. Achieving perceptual equivalence between a computer-generated image and the scene is a significant challenge. Since many perception-based problems must be taken in to consideration. In particular, how to quantify the perceptual quality of an image? Much research has been conducted on the development

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 160–171, 2006.

of perception-based image quality metrics[McN00][RP03][FP04][JAP04]. In this paper, we present a prototype declarative lighting design tool, LightOp, and, motivated by psychological research into human perception, the results of the using LightOp to explore a number of extensions to existing perception-based lighting design algorithms. After a review of existing work in lighting design we extend Shacked and Lischinski's [SL01] approach by applying stochastic optimization, and incorporating new quality components. These extensions aim at revealing features of objects in a scene which in particular support the perception of depth.

## 2 Previous Work

The traditional approach to lighting design for image synthesis is based on direct design methods. Users interactively specify values of lighting parameters, and iteratively render the scene and modify the parameters until the desired visual properties of the scene are achieved. Despite the fact that this can be a tedious and time-consuming process there have been relatively few attempts either to automate or assist the process of lighting design. Schoeneman et al [SDS*93] address lighting design as an inverse problem. Here users set up a set of desired properties that are expected to appear in the final image and the system tries to find a solution (a set of light intensities and colors) whose properties are closest the set of desired properties. Kawai et al. [KPC93] optimize light emission, direction and surface reflectances to obtain the desired illumination for an environment rendered using radiosity-based techniques. In this approach, users have to specify the illumination expected in the final image. Poulin and Fournier [PF92][PRJ97] developed an inverse method for designing light positions through the specification of shadows and highlights in a 3D scene. An interactive sketch-based interface enabled users to sketch desired shadow areas with a mouse pointer. An objective function was defined in a way such that the shadow region for a computed point light (and also some extended light geometries) bounds the sketched regions as tightly as possible. Jolivet et al [JPP02] presented an approach to optimizing light positions in direct lighting using Monte-Carlo methods and reported a declarative paradigm aimed at helping users to specify the lighting goal in an intuitive manner.

## 3 Perception and Lighting

There has been significant effort in recent years in the development of approaches to computer graphics based upon explicit models of a viewer's perception of graphical renderings. Perceptually adaptive approaches have ranged across the entire scope of graphics algorithm and interaction development from schemes for polygon simplification and global illumination that take account of limits on visual attention and acuity, to the design of anthropomorphic animations and gaze-contingent displays [RP03][FP04]. Perception-based lighting design has included implicit approaches that aim to maximize illumination entropy for a fixed viewpoint. Gumhold [Gum02] describes a perceptual illumination entropy approach in which he uses limited user studies to model user preferences in relation to brightness and curvature. In [LHV04] a more explicit model of perceptual preferences is used in the Light Collages framework for which lights are optimized such that the diffuse illumination is proportional to the local curvature, and specular highlights are used only for regions of particularly high curvature.

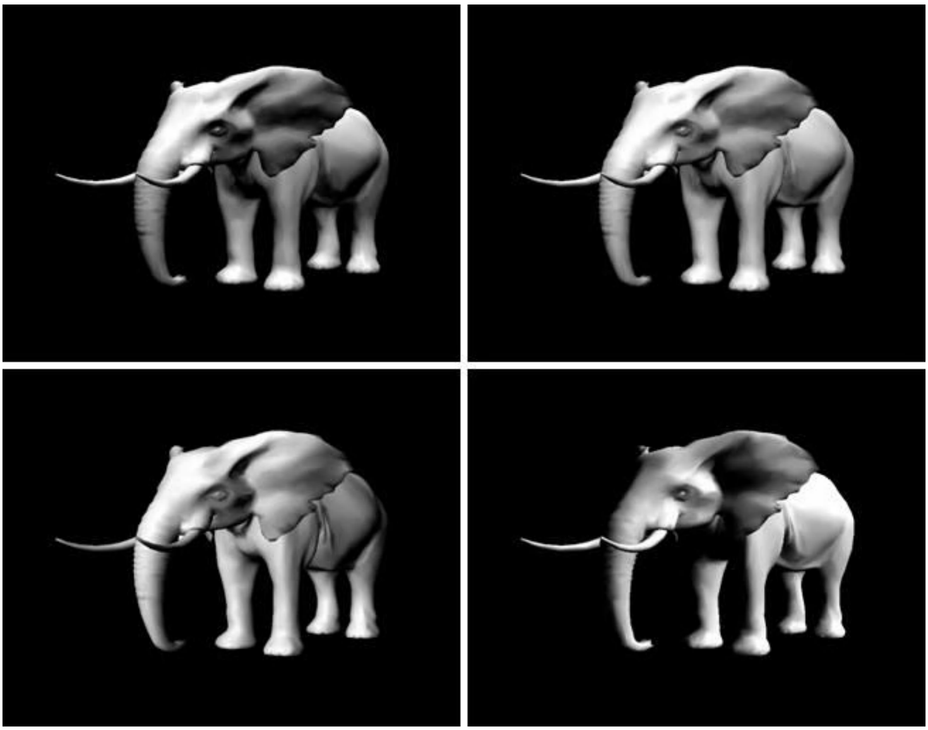


**Fig. 1.** Shacked and Lischinski base algorithm results (greedy search)

We take as a starting point a declarative lighting approach that maintains an explicit model of object perception due to Shacked and Lischinski [SL01]. In their perception-based lighting design scheme the position and intensity (of specular and diffuse components of a local illumination model) of light sources are optimized using an evaluation function that characterizes separate aspects of low-level processing in the segmentation and recognition of objects. At the heart of this approach is an objective function that is the linear combination of five distinct measures of image quality: edge distinctness ($F_{edge}$); mean brightness ($F_{mean}$); mean shading gradient ($F_{grad}$); intensity range ($F_{var}$); and evenness of the distribution of intensity in the image ($F_{hist}$).

$$F(\theta_k, \phi_k, I_{dk}, I_{sk}, R_k) = w_e F_{edge} + w_m F_{mean} + w_g F_{grad} + w_v F_{var} + w_h F_{hist} \quad (1)$$

$\theta_k$: elevation angle of $k^{th}$ light
$\phi_k$: azimuth angle of $k^{th}$ light
$I_{dk}$: diffuse intensity of $k^{th}$ light
$I_{sk}$: specular intensity of $k^{th}$ light
$R_k$: distance $k^{th}$ light (fixed for directional lights)
$w_e$, $w_m$, $w_g$, $w_v$, $w_h$ and $w_c$: weights for different components in the objective function.

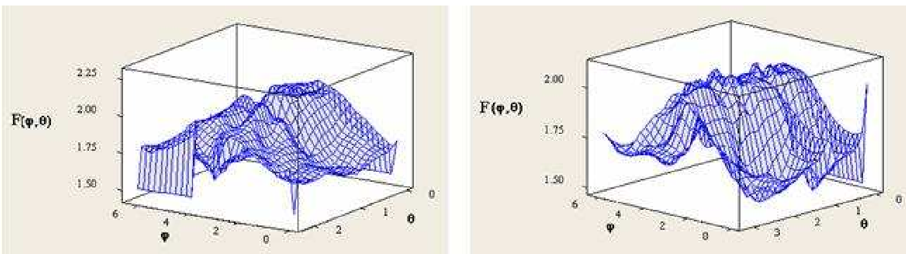



**Fig. 2.** $F(\theta, \phi)$ for one object (elephant) (left) and $F(\phi, \theta)$ for a two object scene

We have omitted the sixth component of the image quality function used by Shacked and Lischinski which biases the optimization of a key light to a particular elevation and orientation above and in front of the object (relative to the viewpoint). Although this is standard practice in photography and might be justified in terms of evolutionary psychology [Mar94][Mil05] – that our perceptual system evolved for scenes lit by the sun or moon – we take the view that the using the direction of the light as a direct component in an objective function is unjustifiably ad hoc (and procedural). Our objective function is formulated such that lower values of $F(\theta_k, \phi_k, I_{dk}, I_{sk}, R_k)$ correspond to configurations with better visual characteristics and a greedy gradient descent minimization algorithm is utilized in the discovery of appropriate lighting configurations.

## 4 LightOp: Exploring Declarative Lighting Design

LightOp is a tool for the interactive configuration of objective functions and optimization schemes, which we have built to explore the problem of declarative lighting design (see figure 4). A range of optimization techniques have been implemented in LightOp including steepest decent, genetic algorithms, and simulated annealing. Lighting parameters for a scene can be optimized with, and without, shadows, and the number of lights used, the components of the objective function used and nature of the color space can all be interactively specified.

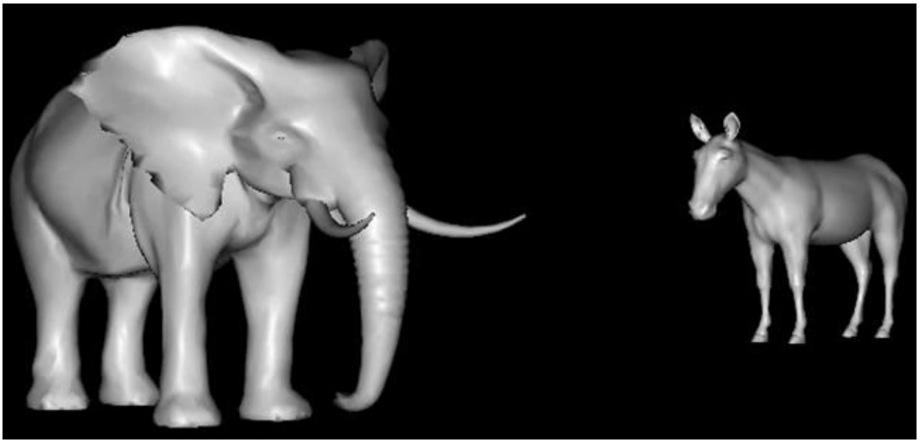


**Fig. 3.** The simple two object scene used in figure 2 (right)

Shacked and Lischinski make no attempt to either characterize the nature of their objective function or assess the suitability of the greedy search employed. From equation (1) it is clear that the multi-objective optimization problem incorporates significant non-linearity, and as the geometric complexity of the scene increases, the number and likelihood of local minima will increase. For a greedy search of a space, such as that illustrated, the solution is highly dependent on the starting condition and figure 1 illustrates the resulting images for four searches conducted from different starting configurations. The nature of the optimization problem suggests a requirement for a more general optimization strategy and we employ a range of stochastic approaches, including a genetic algorithm (GA). The encoding of the lighting problem as a GA is straightforward. The free variables $\theta_k$,$\phi_k$, $I_{dk}$,$I_{sk}$,and $R_k$ are encoded directly in the chromosome as real numbered alleles. We have evaluated the optimization problem with varying population sizes and configurations for the GA. From our experiments we established that a population size of 30, 10% elitism and crossover and mutation rates of 80% and 20% respectively were sufficient for our example scenes (figure 8).

Note that the process of optimizing a single (key) light takes place under the constraint that the light position is limited in the values of elevation and azimuth angles (e.g. the position of a light is limited to a quarter sphere in front and above the center of the scene). This constraint must be respected in all stages of the optimization, but particularly at initialization and during the generation of random mutations (members of the population whose values do not reside in constrained ranges are rejected). Note that as the constrained region is convex, crossover cannot yield offspring that violate the constraint. The GA exhibited consistently better results than the greedy search both in terms of the final value of the objective function and the visual quality of the solution. Figure 6 shows a direct comparison of the greedy and GA optimisation results for the elephant model.

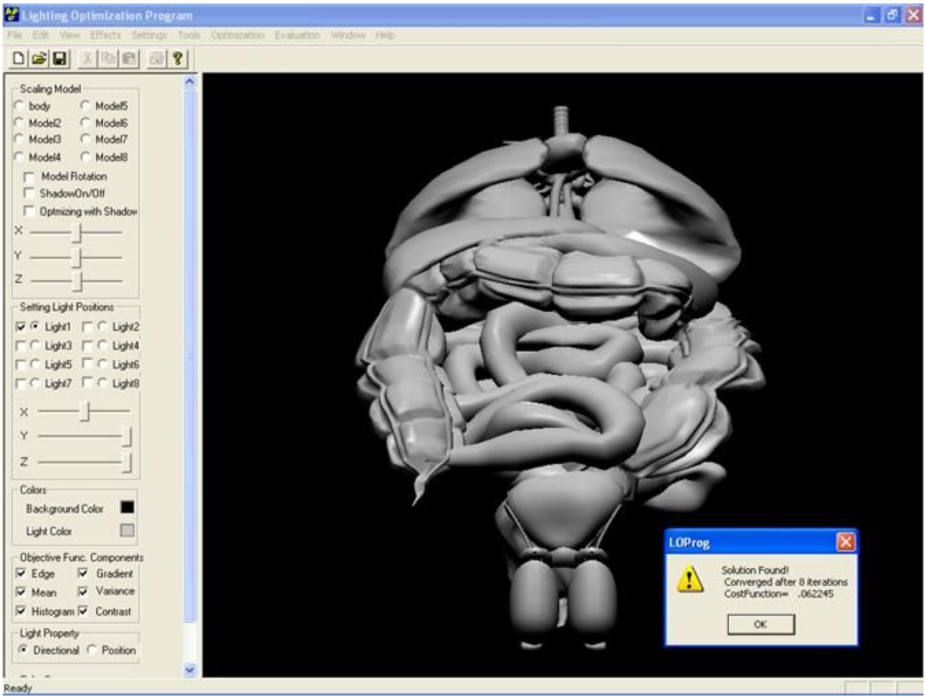



**Fig. 4.** User interface for LightOp showing the objective function & light parameter controls and the result of an optimization

## 5 Enhancing Perception-Based Design

Components of perception-based lighting approaches are motivated directly from the results of studies of human perception, in particular, object recognition [Mar94]. For example, Shacked and Lischinski's edge enhancement criteria relates directly to theories of object segmentation and neuropsychological findings as to the nature of retinal processes [Mar94]. In the course of extending Shacked and Lischinski's approach we identified a number of features currently not addressed in perception-based lighting design: (a) $contrast$ : differences in luminance between different surfaces of an object have been shown to convey significant information about the shape and depth of objects [SJ90]; (b) $back-lighting$ : lighting an object from behind (where the background is dark) is a well established feature of cinematic and photographic practice aimed at maximizing edge enhancement of the silhouette of an object [Mil05]; (c) $perceptuallyuniformcolorspace$ : standard approaches in lighting design implement image quality metrics with respect to RGB (or similar) color spaces, despite the fact that such spaces are highly non-uniform with respect to human judgments of color difference [Mar94].

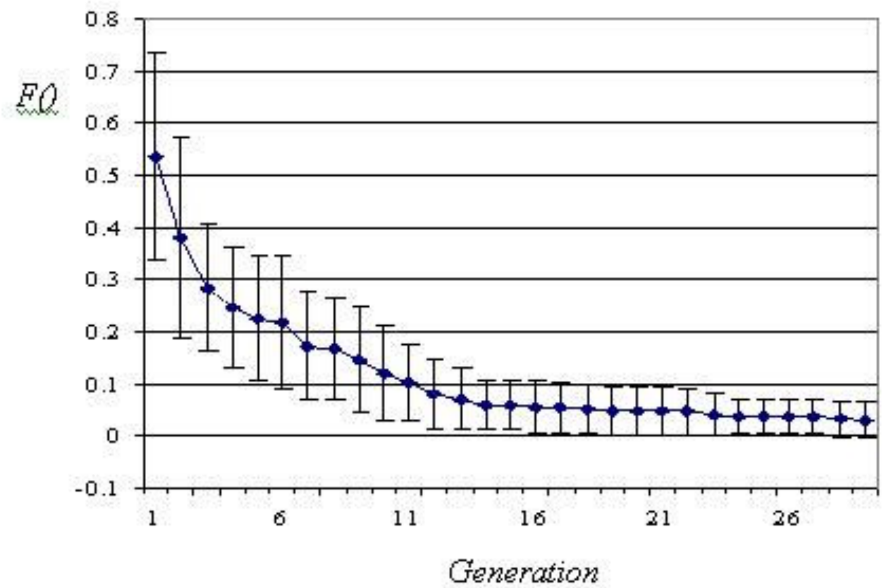



**Fig. 5.** Averaged results for 10 runs of a 30 member population, the value of F() for the best member and standard deviation

### 5.1 Contrast Enhancement

Empirical studies of visual cognition have demonstrated that object perception depends on both absolute luminance levels and differences in an object's luminance from its background [8]. We have included this notion through the provision of a means of evaluating differences in luminance between adjacent parts of an object and incorporating this in our objective function. The contrast between two parts of an object is given by:

$$C_{i,j} = \frac{(Y_i - Y_j)}{Y_i} \tag{2}$$

$C_{i,j}$: contrast between part $i$ and part $j$ $Y_i$: the mean luminance of part $i$ $Y_j$: the mean luminance of part $j$ The mean luminance of a part is calculated as follows:

$$Y_i = \frac{1}{N_i} \sum_{I(x,y) \in P_i} I(x,y) \tag{3}$$

A pixel type map of objects in a 3D scene is extracted by applying an edge detector operator to the depth buffer [ST90]. Edges in the pixel type map correspond to boundaries between parts of an object. With this assumption, we developed an algorithm to calculate the contrast between adjacent parts of an object using the pixel type map. The algorithm, which is supported by definition 1, can be described in pseudo-code as follows:

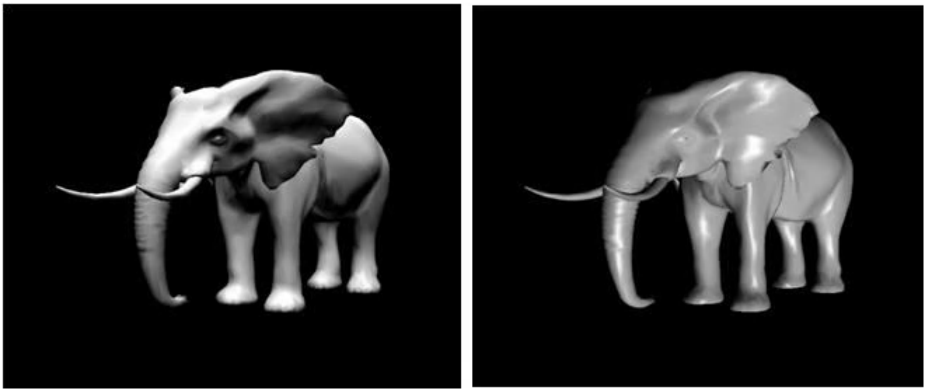


**Fig. 6.** Greedy optimization result for the Shacked & Lischinski [1] objective function (left), and genetic algorithm optimization result for the Shacked & Lischinski [1] objective function (right)

**Definition 1.** *$p(x_i, y_i)$ is a run-length from $p(x_j, y_j)$ if $p(x_i, y_i)$ is the same pixel type as $p(x_j, y_j)$ and all the pixels, which belong to a horizontally continuous connection between $p(x_i, y_i)$ and $p(x_j, y_j)$, are the same pixel type as $p(x_j, y_j)$.*

---

```
For each pixel p(x,y) in the pixel type map
    If (p(x,y) is an EDGE pixel)
        Begin
            x_r = x
            Repeat
                x_r = xr +1
                Calculate mean_right_in
            Until (p(x_r,y)  is BACKGROUND pixel) OR
                     ((p(x_r,y) is EDGE pixel)) AND
                      (p(x_r,y) not RUN-LENGTH from  p(x,y))
            x_l = x
            Repeat
                x_l = x_l -1
                Calculate mean_left_in
          Until (p(x_l,y)  is BACKGROUND pixel) OR
                     ((p(x_l,y) is an EDGE pixel)) AND
                      (p(x_l,y) not RUN-LENGTH from p(x,y))
       Calculate mean_contrast from mean_right_in and mean_left_in
       Add mean_contrast to mean_global_contrast
    End
```

---

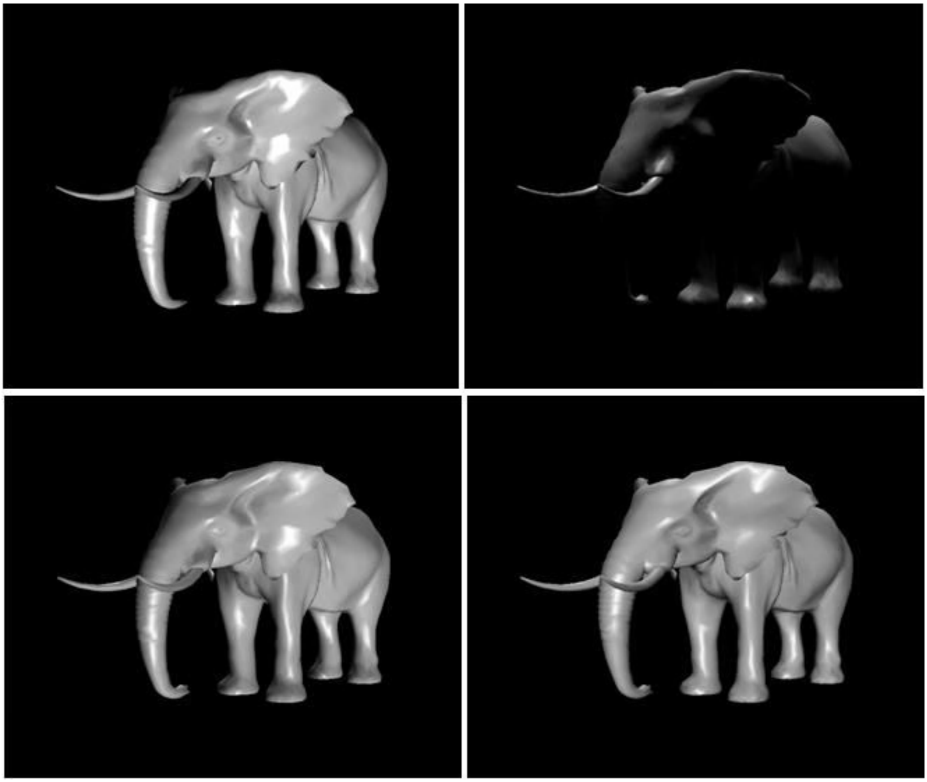


**Fig. 7.** From top-left clockwise: contrast; effect of the back-light; contrast and a back-light; and contrast, back-light, & perceptually uniform color space

Figure 7 (top-left) shows the effect of the contrast component on the example. As can be seen in the other examples shown in figure 8 (top), it is apparent that the contrast component has a tendency to increase the apparent shininess of objects and yields a contrast between adjacent parts of an object that leads to better cues as to depth and the relative positions of parts of an object.

### 5.2 Edge Enhancement and Back-Lighting

In practical contexts, such as a photographer's studio or a film set, backlighting is a useful technique that often makes a valuable contribution to pictorial lighting. Backlighting aids enhancement of external edges and facilitates a viewer in segmenting objects from the background. This is especially true where the surfaces of the object are dark, likewise, backlighting catches the sharply-folded contours and gives them shape and solidity. In our approach, the initial position of the backlight is calculated as a centroid of a set of vertices of objects in a 3D scene. During the optimization process the azimuth angle of the backlight is fixed. Since backlighting primarily effects external edges, we only use edge

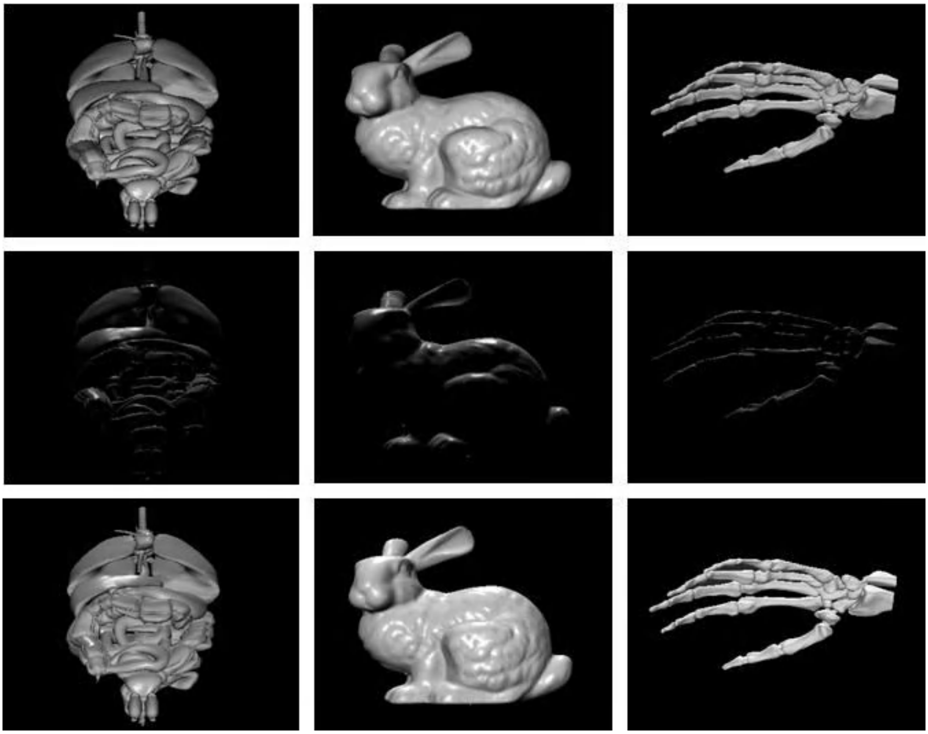


**Fig. 8.** Sample results for three models: contrast (top); effect of back-light (middle); contrast, back-light and perceptually uniform color space (bottom)

component in the objective function when optimizing parameters of the backlight. In reality, a backlight is always positioned above and behind objects, so that the elevation angle of the backlight is confined to a predefined range, and we implement this constraint in our optimization process. Backlight optimization can be viewed as a distinct process from establishing the position and intensity of the key light. We implement it as a second optimization stage using a greedy algorithm. The impact of adding backlighting to the contrast enhanced objective function can be seen in figures 7 (top-right) and 7 (bottom-left). Figure 7 (top-right) shows the result due to the back light alone, and figure 7 (bottom-left) shows the image resulting from the addition of the backlight in which the geometry of the feet is more clearly discernable as are the top of the head and the top of the right ear.

### 5.3 Perceptually Uniform Color Space

Despite the fact that perception-based graphics algorithms attempt to model and quantify human responses to visual stimuli, they routinely use standard RGB (or related) color spaces, and ignore the fact that such color spaces are highly nonlinear with respect to our judgments of color difference. As an alternative, we transform RGB color values to the CIE L*, a*, b* color space – which is approximately uniform [Hal93]. In computing components of the objective function in a perceptually uniform color space the goal is to obtain edge contrasts, gradients and variances that are perceptually distinct. Even for target-based components such as Fmean the error function in the optimization will be enhanced due to a linear relationship between computed (from the color space) and perceived differences. Figure 7 (bottom-right) illustrates a result of incorporating the perceptually uniform color space.

## 6 Discussion

We have used LightOp, our prototype declarative lighting tool, to explore a number of enhancements to the perception-based lighting design approach proposed by Shacked and Lischinski [SL01]. Much work remains to be done and it is our intention that LightOp will serve as the vehicle for a thorough investigation both of the appropriate nature of the objective function, and detailed empirical studies by which users' responses to differently lit scenes are more rigourously investigated. For example, numerous standard tasks in the study of spatial cognition, such as mental rotation and object recognition, rely on the ability of users to identify key aspects of the geometric nature of objects. The performance of users on such tasks is well understood and we intend to leverage such methodologies in charactersing objective functions that are in some way optimal with respect to these tasks.

One should note, however, that this notion of *ideal lighting* is only appropriate with respect to a minority of real-world graphics applications. For example, in some domains, in the automatic lighting of 3D visualizations where the number, position and orientation of 3D glyphs or other objects cannot be predicted in advance (and are typically not textured), the user's ability to recognise and reason about the external representations requires the lighting to be *ideal*. However, in it is more typical for lighting configurations to be used to convey more aesthetic characteristics of a scene such as mood, emotion, and factors other than the purely geometrical. The investigation of aesthetic responses to lighting is a considerable enterprise in itself, though we believe that tools such as LightOp, and a rigorous understanding of perceptually adaptive lighting, are a necessary prerequisite for any such undertaking.

## References

[Bar97] Barzel, R.: Lighting Controls for Computer Cinematography. Journal of Graphics Tool, vol. 2, no. 1, pp 1-20, 1997.

[FP04] Farugia, J. P., Peroche, B.: A progressive rendering algorithm using an adaptive perceptually based image metric. In Eurographics conference proceedings (2004).

[Gum02] Gumhold, S.: Maximum entropy light source placement. IEEE Visualization 2002, pp. 275-282.

[Hal93] Hall, R.: Illumination and Color in Computer Generated Imagery. Monographs in Visual Communication. New York, Springer, 1993.

[JAP04] Jean, P. F., Albin, S., Peroche, B. A perceptual Adaptive Image Metric for Computer Graphics. In Proc. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision WSCG04, 2004.

[JPP02] Jolivet, V., Plemenos, D., Poulingeas, P.: Inverse Direct Lighting with a Monte Carlo Method and Declarative Modelling. CG&GM'02, 2002.

[KPC93] Kawai, J., Painter, J., Cohen, M.: Radioptimization – Goal Based Rendering. Proc. SIGGRAPH '93, pp. 147-154, 1993.

[LHV04] Lee, C. H., Hao, X., Varshney, A., Light Collages: Lighting Design for Effective Visualization. IEEE Visualization 2004, Austin, October 2004.

[MAB*97] Marks, J., Andalman, B., Beardsley, P. A., Freeman, W., Gibson, S., Hodgins, J., Kang, T., Mirtich, B., Pfister, H., Ruml, W., Ryall, K., Seims, J., and Shieber, S.: Design galleries: a general approach to setting parameters for computer graphics and animation. Proc. SIGGRAPH, pp. 389-400, 1997.

[Mar94] Markus G., Visual Computing, Springer-Verlag, 1994.

[Mil05] Millerson, G.: Lighting for Television and Film, 3rd Edition., Focal Press, 2005.

[McN00] McNamara, A.: Comparing real and synthetic scenes using human judgments of lightness. doctoral dissertation , Univ. Bristol , 2000.

[PF92] Poulin, P., Fournier, A.: Lights from highlights and shadows. In Proceedings of the 1992 Symposium on interactive 3D Graphics (Cambridge, Massachusetts, United States). SI3D '92. ACM Press, New York, NY, 31-38, 1992.

[PRJ97] Poulin, P., Ratib, K., Jacques, M.: Sketching Shadows and Highlights to Position Lights. cgi, p. 56, Computer Graphics International 1997 (CGI'97), 1997.

[RP03] Reitsma, P. S. A., Pollard, N.S.: Perceptual metrics for character animation: sensitivity to errors in ballistic motion. ACM Transactions on Graph. 22:3, 537-542, 2003.

[SDS*93] Schoeneman, C., Dorsey, J., Smits, B., Arvo, J., and Greenburg, D.: Painting with light. Proc. SIGGRAPH. 143-146. 1993.

[SJ90] Spillmann, L., John, S. W.: Visual Perception the Neurophisiological Foundations. Academic Press Inc., 1990.

[SL01] Shacked, R.,Lischinski, D.:Automatic Lighting Design using a perceptual quality metric. EuroGraphics, vol. 20, no.3, 2001.

[ST90] Saito, T., Takahashi, T.: Comprehensible rendering of 3-D shapes. Proc. SIGGRAPH, pp. 197-206, 1990.

# A Photographic Composition Assistant for Intelligent Virtual 3D Camera Systems

William Bares

Millsaps College, Department of Computer Science,
Jackson MS 39210, USA
bareswh@millsaps.edu
http://home.millsaps.edu/bareswh/

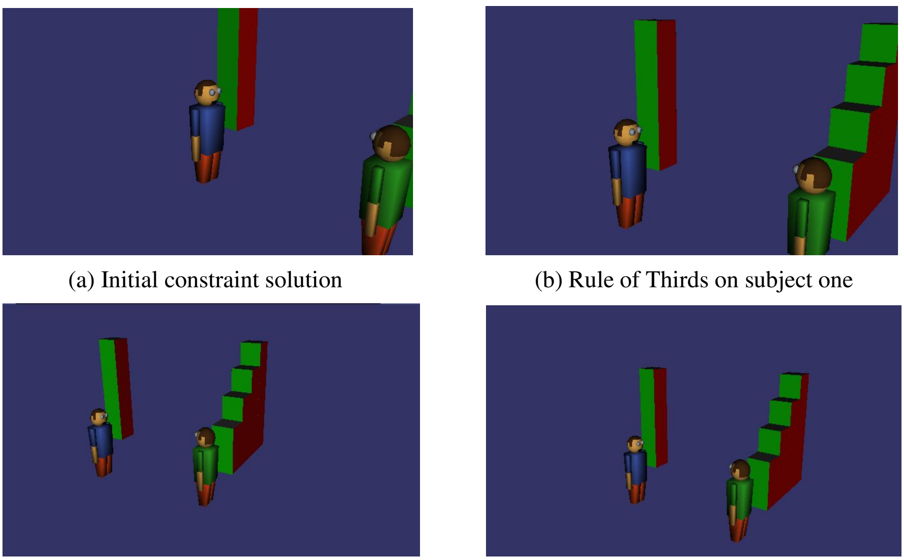
**Abstract.** A human photographer can frame an image and enhance its composition by visualizing how elements in the frame could be better sized or positioned. The photographer resizes elements in the frame by changing the zoom lens or by varying his or her distance to the subject. The photographer moves elements by panning. An intelligent virtual photographer can apply a similar process. Given an initial 3D camera view, a user or application specifies high-level composition goals such as Rule of Thirds or balance. Each objective defines either a One-D interval for image scaling or a Two-D interval for translation. Two-D projections of objects are translated and scaled in the frame according to computed optima. These Two-D scales and translates are mapped to matching changes in the 3D field of view (zoom), dolly-in or out varying subject distance, and rotating the aim direction to improve the composition.

## 1 Introduction

Virtual camera systems automatically compute camera shots using proven cinematography conventions to visualize computer-generated narratives, replay events, or dynamically update camera views so users may interact with a simulation. They analyze scenes to position the camera to view subjects with minimal occlusion from a desirable angle and distance. Consequently, much of the prior work has focused on geometric and temporal properties such as view angle, shot distance, minimizing occlusions, and finding collision- and occlusion-free camera motion paths. Composition is significant since a poorly composed shot detracts from its visual message. For example, a background object may not occlude the subject, but its color, size and location in the frame may carry greater visual weight inadvertently making it the apparent subject. Many automated virtual camera systems are directed only about how a specified set of subjects are to appear in the shot. Other than being instructed to avoid occlusion, non-subject objects are typically ignored. This method augments an existing application's virtual camera system with a composition assistant that analyzes the frame and proposes incremental changes to improve composition properties such as balance or the Rule of Thirds. For example, a typical 3D game camera system positions the camera at a relative view angle and distance, rotates the view angle to avoid occlusion, and aims at the center of the subject(s). This composition

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 172 – 183, 2006.

assistant adds a "post processing" step to improve the composition by adjusting the action game's camera aim direction and distance-to-subject and/or field of view lens angle. For example, our constraint-based implementation of a typical 3D game camera is directed to compute a front-right angle view of a character (at center of frame) whose height should span half that of the frame. Unconstrained non-subjects are visible behind the subject and at the right edge (Figure 1a). Considering only the single constrained subject, use the Rule of Thirds to align it on lines that split the frame into thirds (Figure 1b). Considering all visible elements, the assistant can apply the Rule of Thirds for the whole composition (Figure 1c). The assistant evenly balances the "visual weight" of all elements about the vertical line through the frame center (Figure 1d). Larger elements have greater weight as do elements farther from center [14, 17].



(a) Initial constraint solution

(b) Rule of Thirds on subject one

(c) Rule of Thirds on all elements

(d) Balanced composition

**Fig. 1.** Examples of enhancing the composition of an initial solution

## 2 Related Work

Automatic camera assistants adjust camera position and aim to maintain an occlusion-free view of subjects [16] or project the center of a subject to a desired point on screen [8]. A method similar to this work analyzes a subject's silhouette to optimize aim direction and shot distance to obtain a satisfactory Rule of Thirds layout for a single subject [9]. The Virtual Cinematographer uses Blinn's equations [3] to project an actor to a given point in the frame. It improves a composition by moving virtual actors so they stand on their anticipated staging marks [13].

One method based on visual servos adjusts camera properties to track a moving target [6]. An approach optimized for computer games computes incremental changes to camera properties to maintain subject height, view angle, and visibility [10]. The Virtual Cameraman uses interval calculations to find camera paths whose interpolated positions satisfy desired image properties at specified points in time [12]. A variation on this method uses hypertubes to parameterize camera movements [5]. In another work a digital camera automatically detects the main subject and shifts the image to place it on the Rule of Thirds lines and blurs the background [1].

Constraint-based solvers are told how subjects should appear by view angle, size and location in the frame, and avoidance of occlusion. CAMDROID finds solutions using sequential quadratic programming [7]. CAMPLAN relies on genetic algorithms to satisfy specified visual properties including size, position, relative distance of subjects to the camera, and occlusion [11, 15]. Bounding spheres, wedges, and planes can be used to limit the camera position search space for constraints of shot distance, view angle, and enclosure by the walls of a virtual room [2]. Bounding volumes can also be used to semantically partition space into regions that yield compositionally distinct shots by shot distance, view angle, occlusion, and relative position of subjects [4]. Since constraint specifications are provided prior to computing a shot, it is impractical to specify composition constraints on non-subject objects since one does not know which ones will be visible in the shot.

## 3 Photographic Composition

*Composition* is the arrangement of visual elements in the picture frame. Texts suggest guidelines, but caution that experts sometimes break these "rules." [14, 17].

**Emphasis:** Simplify and reduce distracting clutter around the subject(s).

**Placement or layout:** The Rule of Thirds places subject(s) so they lie along the two horizontal and two vertical lines that divide the frame into nine equal-sized zones. More space should be left in front of a subject in motion.

**Balance:** *Balance* refers to distribution of the visual weight of the elements. Objects that are larger, brighter, closer to the edges, higher, and on the left side have more weight. Several subjects gazing towards one subject enhances its weight.

## 4 Composing in Two-Dimensions

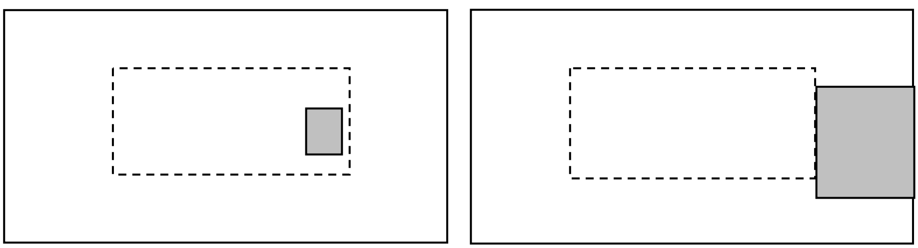
The projections of all visible objects in the frame of an initial camera solution are approximated by a list of bounding rectangles. Rectangle coordinates are normalized with the bottom-left corner of the frame at (-*aspect*, -1) and the top-right corner at (+*aspect*, +1), where *aspect* is the aspect ratio of frame width divided by height. Given one or more user- or application-specified composition goal(s), such as balance or Rule of Thirds, the assistant applies a re-size and/or shift transformation to all elements. A re-size corresponds to a dolly or zoom of the virtual 3D camera and a shift corresponds to a pan. Each composition goal is expressed as one or more re-size and/or shift transforms of a particular element.

### 4.1 Resizing Composition Elements

Resize an element by multiplying its endpoints by a positive constant. This scale transform applies about an origin at the center of the frame. For example, scaling an element that lies to the right of center by a factor of 2 will double its size in the frame and also cause it to shift to the right. For each primitive re-size composition operator, find a `ScalarInterval` of scale factors [minimum, optimum, maximum] that achieves that objective. The following primitive scaling operators are implemented:

**Scale to Fill Rectangle:** Find `ScalarInterval` that scales an element so that it fills as much of the interior of a specified rectangle as possible.

**Scale to Outside of Rectangle:** Find `ScalarInterval` that scales an element so that it lies outside of the specified rectangle. For example, the interval of scale factors [minimum = 1.5, optimum = 1.5, maximum = +infinity] will scale the solid rectangle so that it lies outside of the dashed-line rectangle (Figure 2). This operator can be used to scale up to simulate a dolly-in or zoom-in to crop out a non-subject.



(a) Before scale to outside rectangle (b) After scale to outside rectangle

**Fig. 2.** Scale solid rectangle to lie entirely outside of dashed rectangle

**Scale to Fit Inside Rectangle:** Find `ScalarInterval` that scales an element so that it fits inside the specified rectangle.

**Scale to Line:** Find `ScalarInterval` that scales an element so that it lies on the specified line. The optimal scale places the element's center on the line.

**Scale to Side of Line:** Find `ScalarInterval` that scales an element so that it lies on the designated side of the specified line.

**Scale to Area:** Find `ScalarInterval` that scales an element so that its area lies within the specified [minimum, optimum, maximum] range.

**Scale to Width:** Find `ScalarInterval` that scales an element so that its width lies within the specified [minimum, optimum, maximum] range.

**Scale to Height:** Find `ScalarInterval` that scales an element so that its height lies within the specified [minimum, optimum, maximum] range.

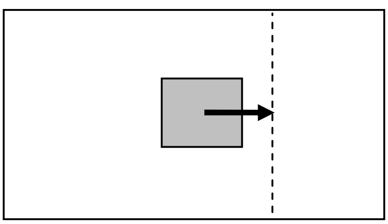
Scale to inside rectangle, scale to outside rectangle, scale to line, and scale to side of line can also be applied to transform points.

### 4.2 Shifting Composition Elements

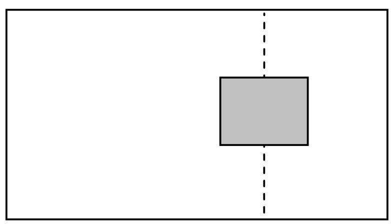
Shift an element by adding a translation vector (dX, dY) to its endpoints. For each primitive shift composition operator, find a `RectangleInterval` [minimumDx, optimumDx, maximumDx] x [minimumDy, optimumDy, maximumDy] that bounds all translation vectors that achieve that objective applied to a rectangle or point element. For transforming a point, a threshold distance is also specified so the interval has sufficient width. The following primitive translation operators are implemented:

**Move to Point:** Find `RectangleInterval` that translates an element so that it lies on a specified point. The optimal (dX, dY) moves the element's center to the point.

**Move to Line:** Find `RectangleInterval` to translate an element so its center optimaly lies on a specified vertical or horizontal line. For example, the arrow is the optimum translation of the solid rectangle (its width is 0.8 units) so that its center lies on the vertical line. The horizontal dimension of the `RectangleInterval` is [0.15, 0.55, 0.95] meaning shift right by no less than 0.15, ideally by 0.55, and no more than 0.95. The vertical dimension of the `RectangleInterval` is [-infinity, 0, +infinity] (Figure 3).



(a) Before move to line



(b) After move to line

**Fig. 3.** Translate rectangle so its center lies on a vertical Rule of Thirds line

**Move to Side of Line:** Find `RectangleInterval` that translates an element so that it lies on the designated side of a specified vertical or horizontal line.

**Move to Inside Rectangle:** Find `RectangleInterval` that translates an element so that it lies inside a specified rectangle. This transform can determine the range of translations that keep an element entirely inside the bounds of the frame.

**Move to Outside Rectangle:** Find `RectangleInterval` that translates an element so that it lies outside a specified rectangle.

## 4.3 Measuring Visual Weight

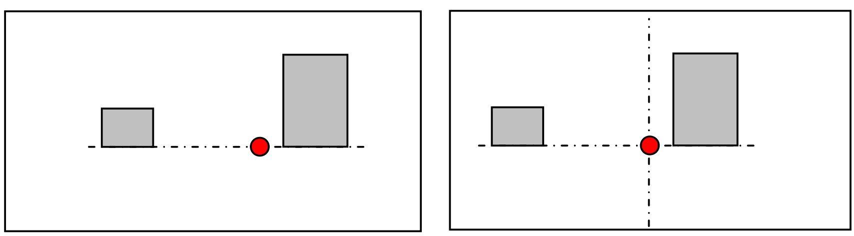
Compute the weight of each visible element in the frame by accumulating the following components, each of which is evaluated as a normalized value between 0 and 1.0.

brightnessWeight = maximum intensity( red, green, blue of element's color )
The application specifies the predominate color of each subject
horizontalWeight = AbsoluteValue( element.CenterX() ) / (0.5 * frame.width())
Increase by 10% if element is left of frame center
verticalWeight = (element.CenterY() - frame.minY()) / frame.height()
sizeWeight = element.diagonalLength() / frame.diagonal()
gazeWeight = number of elements pointing to this element / (numElements-1)

For gaze, compute the angle between element E's projected heading vector and a vector directed from the center of element E to the center of subject being weighed.

## 4.4 Balancing the Composition

Translate elements so the Center of Visual Weight (CoVW) coincides with a specified horizontal U-axis coordinate H. In Figure 4 achieve formal balance by translating Center of Visual Weight to the vertical line through H = 0 at the frame center. The assistant can be instructed to move the Center of Visual Weight to lie on a specific vertical line to place greater weight off center for informal or dynamic balance.



(a) Find Center of Visual Weight (b) Pan to shift CoVW to mid line

**Fig. 4.** Achieve formal balance by shifting elements so CoVW is at horizontal center

Given that the visual weight of each element is $w_i$ and the horizontal u-coordinate of each element is $u_i$. To compute the u-coordinate of the Center of Visual Weight:

$$centerU = \sum_{i=1}^{num\_elements} w_i * u_i$$

The v-coordinate of the Center of Visual Weight is found in a similar fashion.

### 4.5 Compound Composition Goals

For example, suppose we want leading space (60% of frame width) ahead of the subject in the center of Figure 1a and exclusion of non-subjects. To optimize the virtual camera's aim direction, construct an AND/OR Tree for the desired translate transforms. To leave more space to the right of our subject, find the Move to Side of Line `RectangleInterval` that moves the subject's bounding rectangle to lie on the left side of a vertical line left of frame center. Keep the subject entirely visible in the frame by a Move to Inside Rectangle transform. In this example, the user chooses to exclude all non-subjects. Exclude each non-subject (second character and boxes) with a Move to Outside of Rectangle transform which is implemented by a disjunction of four Move to Side of Line transforms, one per edge of the frame. Figure 5 depicts the AND/OR Tree to find a single translation to pan the camera. Internal nodes represent AND or OR operations. An AND node may be "weak" meaning that satisfying a subset of its children is acceptable. Leaf nodes contain `RectangleInterval` transforms. Only one of the five OR nodes to exclude one of the four stacked boxes on the right is depicted. Intervals are marked with benefit scores, which propagate upwards. Sort child nodes by descending score to facilitate finding a partial solution with the greatest benefit when unable to satisfy all objectives.

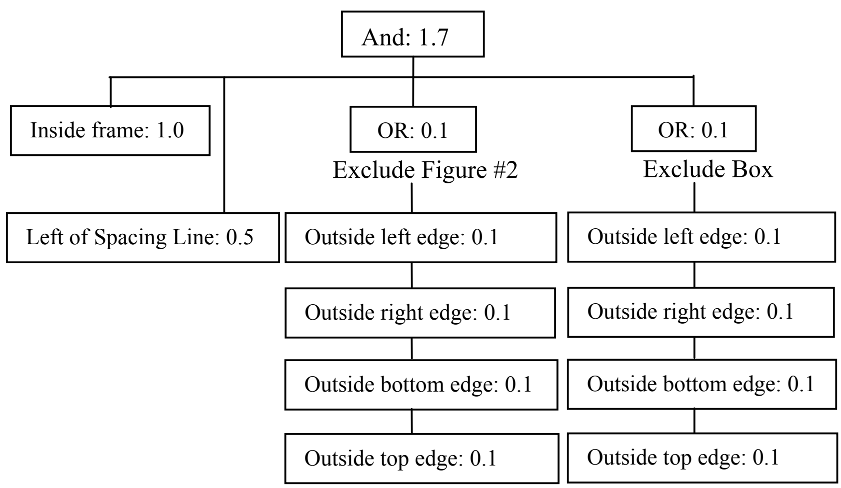



**Fig. 5.** AND/OR Tree to pan to give leading space and exclude non-subjects

In this example, the pan can only satisfy leading space ahead of the subject. Update the aim direction and re-project the bounding rectangles of all elements. Construct a similar separate AND/OR Tree to optimize the dolly distance and field of view/zoom using `ScalarInterval` leaf nodes. This second pass finds a scale transform that successfully zooms-in to exclude all but one of the non-subjects (Figure 6).

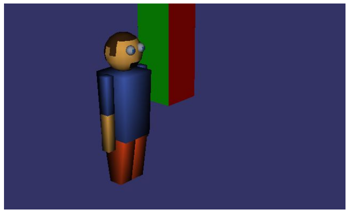


**Fig. 6.** Refined composition for leading space and exclusion of non-subjects

## 5 Optimizing Composition Transforms

The same optimization algorithm is used to compute the optimal interval representing either a range of possible scale factors to resize the composition elements or a range of possible translation factors to pan the composition elements. Each OR-node is treated as a variable whose value represents the selection of one of its child nodes. The backtracking constraint-satisfaction algorithm finds a consistent assignment of values for each OR-node. Figure 5 represents a problem with two variables, one per OR node. Each OR node has four possible values or directions in which to pan the camera to exclude an element. Initially no variables are assigned values.

Let `maxSolution` be the empty interval with a score of 0.
Let the initial `solution` be an empty interval with a score of 0.

```
while ( is unassigned variable ) {
    do {
      For current variable, assign next untested value
      Evaluate AND/OR Tree to find solution interval
    } while(solution is inconsistent and
            untested values remain for current variable )

    if( consistent value is found for current variable )
    {
       current variable = next unassigned variable
       if(solution.score() > maxSolution.score())
         maxSolution = solution
    }
    else backtrack to prior variable with untested value
}
return maxSolution
```

The do-while loop tries OR-node values until it finds a consistent solution. A solution is consistent if the intersection of all AND-node child intervals along with the intervals corresponding to the current assignment of OR-node variables is non-empty. If an AND node is specified as being weak, the algorithm excludes the lowest priority non-intersecting intervals to account for partial solutions in cases of conflicts. If a consistent solution is found, then compute the benefit score for the successfully intersected intervals. The optimum value of the result interval is a weighted average of the optimum values of the two intervals being intersected. If either input interval has an identity optimum, 1.0 for scale or (0, 0) for translation, then the result optimum is taken from the other input interval. The benefit score of the result interval is taken as the sum of the benefits of the two input intervals. An overall benefit is propagated up the tree to rank solutions. Record the non-empty solution interval having the maximum benefit score. The `maxSolution` interval is returned and is used to find an equivalent camera dolly, zoom, or pan to improve the composition.

## 6 Mapping Solution Intervals to Camera Parameters

### 6.1 Camera Field of View Angle for Scale

Given a positive *scale* factor to re-size all elements in the frame, find the vertical field of view angle in radians spanning the full field of view that yields a zoom in or out.

Let `pHeight` be the height of the front face of the perspective frustum. Let `pDistance` be the distance from the camera point to the front face of the frustum.

```
FOV = 2.0 * atan( 0.5 * pHeight / (scale * pDistance) )
```

### 6.2 Camera Dolly Distance for Scale

Given a *scale* factor to re-size all elements in the frame, find the signed dolly distance to translate the camera along its unit-length aim vector. Given reference point `RefXyz` at the center of the 3D subject geometry. Find `RefUvn` by converting into UVN Camera Coordinates with the v-coordinate adjusted to be non-zero if necessary.

Let `pDistance` be the distance of the projection plane to the camera position.

```
v = pDistance * refUvn.v() / refUvn.n()
scaledV = scale * v
```

Using similar triangles, `scaledV / pDistance and refUvn.v() / (refUvn.n() + dolly),` find the dolly distance:

```
dolly = (pDistance*refUvn.v() - v*scale*refUvn.n()) /
        (v * scale)
```

### 6.3 Camera Aim Direction for Translation

Given a translation (`dU, dV`), expressed in normalized frame coordinates, find the new camera aim direction to pan the frame. A reference point `RefXyz` at the center of the subject geometry is given. Compute the new aim direction `N`-axis as follows:

```
v = dV * 0.5 * pHeight
```

Find length `h` of hypotenuse vector `H` of the right triangle formed by camera position, translated point (`dU, dV`) and yet-to-be-determined `N`-axis.

`h = sqrt( v*v + pDistance*pDistance )`
Let `camPosToObj = refXyx – cameraPosition`
`H = h * camPosToObj`
Set local camera system `U`-axis vector ("right hand of virtual camera person").
`U = H x UP,` where `UP` is the world's global "up" vector.

Find angle β in radians between vectors `H` and yet-to-be-determined aim vector `N`.
`β = asin( v / h )`

Let `Q` be the quaternion representing the counter-clockwise rotation about `U` by angle β. Convert the quaternion into an equivalent 4x4 rotation matrix.
`rotMatrix = Q.convertTo4x4Matrix()`
Rotate `H` counterclockwise by angle β about axis `U` to form updated `N` vector.
`N = rotMatrix.transformVector(H)`

Find `u`, the projection of `dU` onto the perspective frustum.
`u = dU*(0.5*pWidth/aspectRatio)`
Find length `h` of the hypotenuse of the right triangle formed by camera position, translated point (`dU, dV`) and to-be-determined `N`-axis.
`h = sqrt( u*u + pDistance*pDistance )`

Angle in radians between vectors `H` and `N`-axis: Θ = `-asin( u / h )`

Let `V = U x N.` `Q` is the quaternion to rotate `N` clockwise by Θ about axis `V`.
`rotMatrix = Q.convertTo4x4Matrix()`
`N = rotMatrix.transformVector(N)`

Normalize the new local camera coordinate system axes vectors `U`, `V`, and `N`. Vector `N` becomes the new camera aim direction vector, `U` its "right hand", and `V` its "hat".

## 7 Implementation

Composition goals are specified via a menu interface. A constraint solver computes an initial solution for projection height, view angle, and avoid occlusion of the subject. In all examples, the preferred view angle is occlusion-free so it only performs ray-to-box queries. If the desired view is occluded the solver would change the view angle. The Open Scene Graph API performs the rendering. Table 1 gives C++

**Table 1.** Average time to compute camera solution for selected figure screen shots

| Figure | 1 box | 5 boxes | 10 boxes |
|---|---|---|---|
| Fig. 1(b) | 2.2 ms | 4.0 ms | 6.6 ms |
| Fig. 1(c) | 3.4 ms | 8.2 ms | 14.8 ms |
| Fig. 1(d) | 3.4 ms | 7.0 ms | 12.4 ms |
| Fig. 6 | 3.4 ms | 6.0 ms | 9.8 ms |

benchmarks on a 2.66 GHz Intel Xeon running LINUX. Columns represent variants of the scene with differing numbers of similarly configured boxes. The column labeled “5 boxes” gives the benchmarks for the images shown in the figures.

## 8 Conclusions and Future Work

This method is fast and easy to install into existing automated camera systems. The method can be used to implement projection size and location constraints for a basic automated camera system. The example implementation does just that. Future work includes fully automating the selection and benefit weighting of composition goals by computing visual attention saliency of visible elements. Unify the separate translate and scale passes by formulating the composition objectives in terms of how to reposition and/or re-size the rectangular bounds of the “picture frame” in one step. Cluster nearby non-subject elements to improve efficiency. User testing can be conducted to assess the aesthetic quality of the compositions.

## Acknowledgements

Thanks to Arnav Jhala, James Lester, Charles McClendon, and the anonymous reviewers for their suggested improvements.

## References

1. Banerjee, S., Evans, B.L.: Unsupervised Automation of Photographic Composition Rules in Digital Still Cameras. In Proc. IS&T/SPIE Conf. on Sensors, Color, Cameras, and Systems for Digital Photography, Jan.18-22 2004, vol. 5301, San Jose, CA (2004) 364-373
2. Bares, W., McDermott, S., Boudreaux, C., Thainimit, S.: Virtual 3D Camera Composition from Frame Constraints. In *ACM Multimedia 2000*, Los Angeles, California, October 30th to November 4$^{th}$ (2000) 177-186
3. Blinn, J.: Where am I? what am I looking at? IEEE Computer Graphics and Applications. (1988) 76-81
4. Christie, M., Normand, J-M.: A Semantic Space Partitioning Approach to Virtual Camera Control in Proceedings of the Annual Eurographics Conference, Computer Graphics Forum, Volume 24-3, (2005) 247-256
5. Christie, M., Languénou, E., Granvilliers, L.: Modelling camera control with constrained hypertubes. In Hentenryck, P. V., editor, Principles and Practice of Constraint Programming CP2002 (Lecture Notes in Computer Science), (2002) 618-632
6. Courty, N., Marchand, E.: Computer animation: A new application for image-based visual servoing. In Proceedings of IEEE Int. Conf. on Robotics and Automation, ICRA'2001, volume 1, (2001) 223-228
7. Drucker, S. M., Zeltzer, D.: Intelligent camera control for virtual environments. In Graphics Interface '94. Morgan Kaufmann Publishers (1994) 190-200
8. Gleicher, M., Witkin, A.: Through-the-lens camera control. In Proceedings of ACM SIGGRAPH'92, (1992) 331-340
9. Gooch, B., Reinhard, E., Moulding, C., Shirley, P. Artistic Composition for Image Creation. In Eurographics Workshop on Rendering, (2001) 83-88

10. Halper, N., Helbing, R., Strothotte, T.: A camera engine for computer games: Managing the trade-off between constraint satisfaction and frame coherence. In Proceedings of the Eurographics'2001 Conference, volume 20, (2001) 174-183
11. Halper, N., Olivier, P.: CAMPLAN: A Camera Planning Agent. In *Smart Graphics. Papers from the 2000 AAAI Spring Symposium (Stanford, March 20-22, 2000)*, Menlo Park, AAAI Press, (2000) 92-100
12. Jardillier, F., Languénou, E.: Screen-space constraints for camera movements: the virtual cameraman. In Eurographics '98, volume 17, Computer Graphics, Special Issue (1998) 174-186
13. Li-wei He, Cohen, M.F., Salesin, D.H.: The virtual cinematographer: A paradigm for automatic realtime camera control and directing. In *Computer Graphics (Proceedings of SIGGRAPH '96)*, (1996) 217-224
14. O'Brien, M., Sibley, N. The Photographic Eye: Learning to See with a Camera. Davis Publications, Inc., Worchester, Massachusetts (1995)
15. Olivier, P., Halper, N., Pickering, J., Luna, P.: Visual composition as optimisation, *AISB Symposium on AI and Creativity in Entertainment and Visual Art*, Edinburgh (1999) 22-30
16. Phillips, C.B., Badler, N., Granieri, J.: Automatic viewing control for 3D direct manipulation. In David Zeltzer, editor, *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, volume 25, March (1992) 71-74
17. Ward, P.: Picture Composition for Film and Television. Focal Press, Woburn, Massachusetts (1996)

# Copy-Paste Synthesis of 3D Geometry with Repetitive Patterns

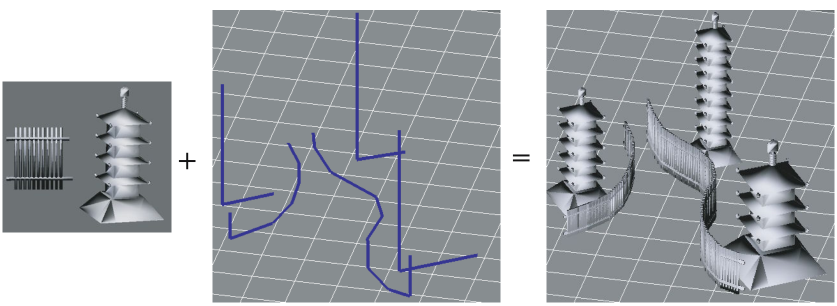
Shigeru Owada[1], Frank Nielsen[1], and Takeo Igarashi[2,1]

[1] Sony Computer Science Laboratories, Inc.
Gotanda, Tokyo, Japan
{sowd, Frank.Nielsen}@acm.org
[2] The University of Tokyo, Bunkyo-ku, Tokyo, Japan
takeo@acm.org

**Abstract.** We propose a new copy-paste user interface for 3D geometry based on repetitive patterns. The system, guided by the user, analyzes patterns of repetition in the source geometry and then pastes the geometry while increasing or decreasing the number of repetitions using scaling and deformation, which is controlled by two freehand strokes called *handles*. The system has two main advantages over existing methods: the entire copy-paste operation is controlled by the user's stroke input and thus can be specified easily without explicitly adjusting the parameters, and splitting the shape information into source geometry and handles can not only significantly reduce the amount of data required but also quickly change a scene's appearance while keeping its structure consistent.

## 1 Introduction

Shape modeling is one of the central challenges in contemporary computer graphics (CG) systems. Because image-generation (rendering) technology already




Source geometry User-defined handles Target geometry

**Fig. 1.** Overview of the proposed copy-paste operation. Using handles, a shape can be dynamically deformed, and the number of times an element is repeated can be modified.

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 184–193, 2006.

achieves excellent results, the main interest in the field is currently how to create shapes more quickly and cost-efficiently without loss of visual quality in the final output. The most time-consuming and costly processes in CG production are shape modeling and animation, because they require human input. Therefore, it is important to improve the user interface to reduce the time and cost required for CG development. The aim of this study was to enhance the process of generating stationary objects, and propose a new, more effective copy-paste tool to improve the user interface.

The copy-paste operation is one of the most important in computer-aided content creation. Its effective use can dramatically accelerate the process of shape creation, so commercial systems usually offer various types of copy-paste operation. However, most existing copy-paste functions require explicit numerical input, which is useful for regular operations but not intuitive for irregular operations such as copying with deformation.

Therefore, we propose the following copy-paste function:

- The source geometry is semi-automatically analyzed to find patterns of repetition, then split into constitutive elements.
- It is not necessary to store the entire geometry. If the original shape contains repeated elements, the system stores only a single copy of the element.
- When pasting the geometry, the user simply draws 2D strokes on the screen. These strokes are then converted to a 3D curve, which we call a *handle*. A handle represents the skeleton of the pasted geometry. The system automatically adjusts the number of times elements are repeated, and deforms the original geometry along the curve.

Some features of the copy-paste operation follow:

- The user can easily specify the information required through stroke inputs.
- The data are compressed because of the lack of repetition of constitutive elements.
- The shape acts like a *skin* on the handles, because a single set of handles can be associated with many different, stylized geometries.

## 2 Related Work

Efficient copy-pasting is one of the key reasons that computers are so useful. Almost all commercial data-editing software supports copy-paste operations, including spreadsheets and text, image, and sound/music editors. In 3D graphics creation, excellent copy-paste skills and functions are critical for 3D CAD designers to create complex scenes quickly. Therefore, commercial 3D modeling software usually supports multiple types of copy-paste operation. For example, Lightwave 3D Ver.8 [1] supports mirror, symmetric, clone, helix, spin-it, and array copy (with the option of a grid-like or circular layout), as well as particle, path, and rail clone, and point-clone plus. However, these functions are batched, low-level operations that require the user to carefully input parameters and select the region to be copied.

A great many articles have proposed methods for efficiently copy-pasting geometry. Biermann et al. proposed a cut-and-paste tool for subdividing surfaces [4]. An implicit surface tool is an extremely useful primitive for copy-pasting that smoothes the boundary in a natural way [18]. Funkhouser et al. extended Intelligent Scissors, originally designed for 2D images, to partition 3D models [19,8], and developed a method to combine models by connecting two open boundaries. They introduced the technique in the context of example-based modeling, which proposes combining models in databases to generate new shapes. There are some other ways to combine models [24]. For certain shapes, repetition is key to performing intelligent copy-paste operations. Procedural modeling systems usually explicitly consider patterns of repeated elements [23,2], and techniques for identifying such repetition in the input have been extensively explored in 2D image processing for the purpose of 3D reconstruction or image segmentation based on texture. [10,13].

Texture synthesis a way to expand textural images without introducing visible seams, while keeping the appearance of the input. A variety of algorithms achieve this goal, including pixel- [7,14], frequency-, [9,6], and patch-based [15,17] synthesis, and non-periodic tiling [21,5]. The system proposed in this paper is very similar to texture synthesis in that copy-pasting a data set is another way of synthesizing textures. There are some existing systems that synthesize 3D geometry [16,3], but these are essentially extensions of pixel-based texture synthesis to 3D volumes and are therefore far slower than the proposed copy-paste operation.

The user interface of the proposed system was inspired by so-called sketch-based modelers. The SKETCH system utilizes a set of 2D gestural operations to input 3D shapes without changing the viewpoint to explicitly input 3D coordinates [25]. The Teddy system extends the idea of using 2D gestural inputs to define 3D geometry [11] by applying a plausible assumption of the target shape to achieve a highly intuitive response to the 2D input. The proposed system basically adopts the user interface of the SKETCH method, because it is suitable for specifying straight lines or planar curves with the help of a ground plane.

## 3 The Proposed System

### 3.1 Shape Representation

The input is a point cloud. If the original data are for a polygonal mesh, they are converted into a point cloud by resampling [20].

### 3.2 User Interface

The user interface is as follows:

**Fig. 2.** Deformation of the loaded model. Left: before. Right: after.

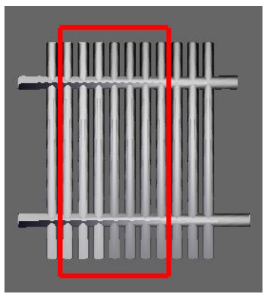


**Fig. 3.** Selection of region of interest

- Loading a model
  A shape model is loaded by dragging-and-dropping the source file into the main window. The system optionally finds the principal axes using principal component analysis (PCA), and rotates the model to align it with the x- or y-axis of the screen.
- Deformation of the model (optional)
  The user has the option to deform the model by drawing a single stroke on the central axis of the model, which is accomplished by pressing the left mouse button and dragging while holding down the Shift button on the keyboard (Fig. 2). The stroke is then straightened to deform the entire geometry.
- Selecting the region of interest
  The user selects the region of interest (ROI) by using the selection tool (Fig. 3) and dragging while pressing the left mouse button. In this operation (and successive operations to split the ROI into elements), all points that are rendered in the 2D region specified on the screen are considered inside the ROI.

  Once the ROI has been selected, a new window pops up that displays the selected region. This window is called the item window (Fig. 4).

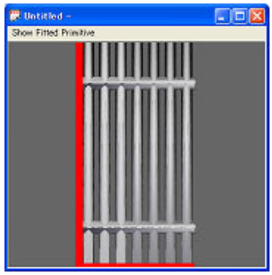
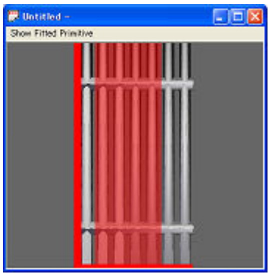



**Fig. 4.** Item window



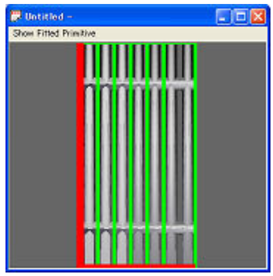




**Fig. 5.** Selection by the user (left) and identification of repeated elements suggested by the system (right)

– Specifying a region that includes repeated elements (optional)
  If the selected ROI contains repeated elements, the user can add a stroke to the item window to explicitly specify that the region contains repetitions (Fig. 5 left). After doing so, the system automatically identifies the repeated elements and splits the model accordingly, separating the repeated elements from their surroundings (Fig. 5 right). Because it is assumed that the number of repetitions is a natural number, the boundary is also improved to avoid overlapping elements.

  Note that this procedure is optional. If no region is designated as containing repetitions, the model is stretched when it is pasted.
– Pasting
  The user draws a handle in the target domain, and the 2D stroke is converted into 3D by projecting it onto the Z-buffer (Fig. 6 left). Then the user draws another stroke touching one of the two ends of the first stroke. This second stroke should be drawn in the vertical direction (Fig. 6 middle). The system again converts the stroke into 3D by projecting it onto the plane that is vertical to the floor and passes through the endpoint of the first stroke.

  The system measures the length of the 3D strokes, and pastes the geometry, the shape of which is deformed by the first stroke. The scale of the pasted geometry is determined by the length of the second stroke, and the number of repetitions is determined by the length of the first stroke (Fig. 6 right).

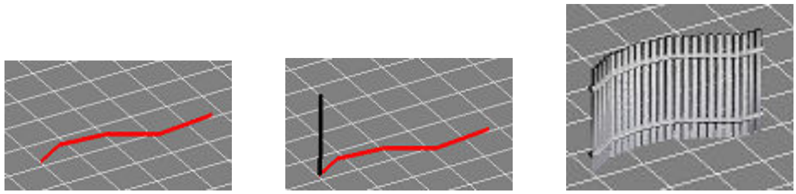


**Fig. 6.** Paste operation

The system stores the original geometry and the handles separately, so it is possible to apply different geometries to the same handles by importing a new geometry set file that includes the same repetitive elements as the original.

### 3.3 Implementation

**Deformation by a single stroke.** Many algorithms are available for deforming geometry using user stroke input. We adopted the 2D 'as rigid as possible' (ARAP) technique [12]. When the user specifies the deformation stroke for the source project (Fig. 7 left), the system first determines whether the shape is aligned along the vertical or horizontal direction, using the difference between the x and y coordinates of the two endpoints of the input stroke. To be concise, the model is assumed to be aligned along the vertical direction (y axis). Then a coarse triangular mesh is constructed that covers the rendered object, using the Z-Buffer information. In our case, the grid size was 40 pixels. Then some vertices in the mesh are snapped to the stroke drawn by the user (See Fig. 7 center). These snapped vertices function as constraints, which are aligned along the vertical axis and retain the same distances between their points. Controlled by these constrained vertices, 2D ARAP deformation is applied to the mesh, and then each point is relocated using the deformation matrix of the triangle that contains the point.

The original 2D ARAP technique [12] consists of two steps: scale-free deformation and scale adjustment. However, applying only the first step returns a more intuitive result than applying both steps.

**Item window and repetition analysis.** The selected ROI is displayed in the item window, and the user selects a region with repeated elements. The direction of the stroke is either vertical or horizontal, assuming that the structure is aligned along one of the axes. Once the region containing repeated elements has been specified, the frequency of the pattern is computed from the current Z-buffer. The algorithm is as follows, assuming that the stroke is drawn along the x axis:

- Each horizontal raster of the current Z-buffer is analyzed by convolving trigonometric functions at various frequencies. This convolution operation is the same as discrete Fourier transform (DFT) but differs in that the wavelength of the convolved trigonometric functions is not the pixel length to

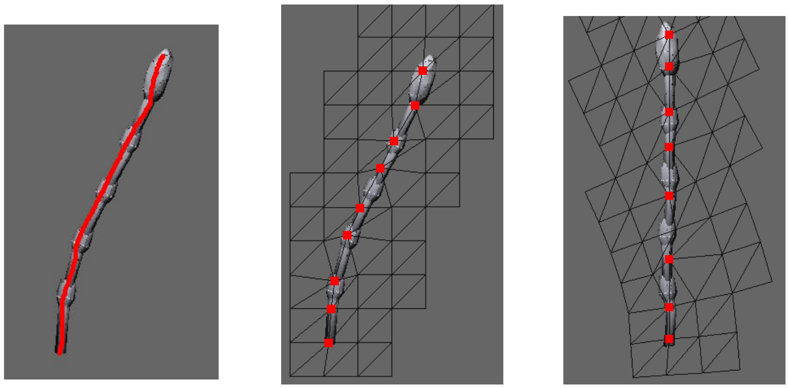


**Fig. 7.** Deformation by 2D ARAP mesh

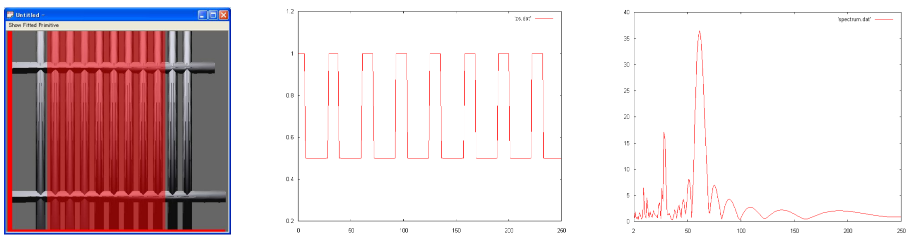


**Fig. 8.** Frequency analysis of the Z-buffer. Left: input. Center: sampled Z-buffer of one raster. Right: the spectrum of the raster.

the power of two. The magnitude of the frequency component is measured for all bands, from the two-pixel wavelength to the half-image-width wavelength, using a step-size wavelength of one pixel (see Fig. 8). For example, the raster with trigonometric functions is convolved with wavelengths of two pixels, three pixels, four pixels,..., w/2 pixels, where w is the width of the ROI.

– Find the peak of the frequency histogram for all rasters.

Once the frequency of elements has been identified, the object is split by its wavelength. If the width of the object does not match the integer multiplication of the wavelength, the surplus points are removed from the region with repeated elements and added to a neighboring non-repetitive region.

**Pasting with repetition and deformation.** The user draws two handles comprising one horizontal 3D curve and one straight, vertical line. This user interface is similar to the one proposed in the SKETCH system [25]. As explained in Section 3.2, the handles consist of two 3D curves. The vertical stroke defines the scale of the pasted model, and the horizontal stroke defines the number of repetitions and deformation.

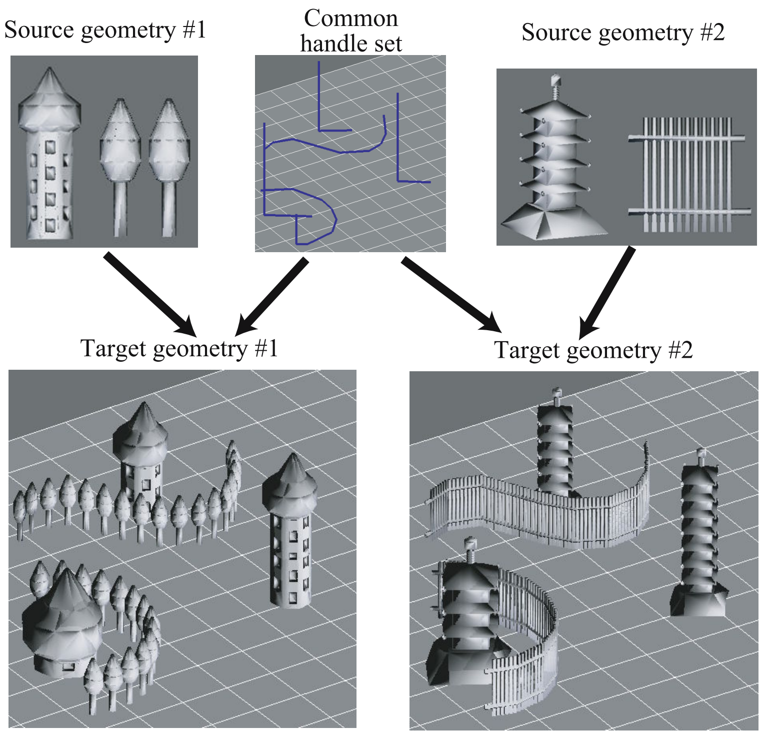



**Fig. 9.** Different source geometries with a common handle set

Again, deformation is performed using the 2D ARAP technique [12]. First, the number of repetitions in the source geometry is fixed and an un-deformed shape is created in the object space. Then, the boundary box of the object is calculated. The bottom face of the boundary box is tessellated by triangles to define the deformation field, and it is assumed that the un-deformed horizontal handle is defined as at the central line of the tessellated field. This line (or more precisely the vertices on this line) is used as the constraint of the 2D ARAP technique and deforms the tessellated bottom face to the target domain. All points in the original shape are then relocated according to the mesh, assuming that the barycentric coordinate of each point with respect to the enclosing triangle is maintained.

## 4 Results

The proposed system can create a model like that shown in Fig. 1 within a few minutes. One interesting feature is that different geometries can be applied to one set of handles (Fig. 9).

This procedure of splitting a source into basic shapes and handles has some interesting potential applications. One is the compression of data. Because handles usually require very little data, a shape can be efficiently represented even if the target geometry contains many repetitions. For example, the handles in Fig. 9 amount to only about 1 kb of data. Another interesting application is skin geometry. Imagine if 3D desktops are developed in the future. We may want to use different desktop styles, each customized to some special purpose, but still need common functional support of the operating system. Splitting styles and skeletons should be strongly advantageous in such a case.

## 5 Conclusions and Future Work

We proposed a new copy-paste technique that is based on pattern analysis. A region containing repeated elements of the object is specified manually, but the frequency is analyzed automatically. Then, based on the analysis, the object is split into elements. We also introduced the notion of handles to ease the pasting operation. Handles specify the scaling factor, number of repetitions, and the deformation path using two 3D paths.

In future work, we hope to improve the model so that it can analyze the structure of repetition (related to the texture-based segmentation of 2D images) automatically, and be applied to scanned data or more generic shapes. This is a more challenging task, because general shapes may contain very complicated structures.

Another possible future direction would be to mix multiple input shapes to create an entirely new model, similar to texture mixture (texture fusion), which combines multiple textures to synthesize a single image [22]. The synthesized texture has an appearance that is intermediate to the input textures. This technique can be directly applied to geometry synthesis, but may require more elaborate processing, such as decomposition of shapes into textures, fine geometry, and base geometry. We believe that a technique that flexibly mixes input models would be a significant contribution to the field of example-based modeling.

## References

1. Lightwave 3D. *NewTek http://www.newtek.com/.*
2. Daniel Bekins and Daniel G. Aliaga. Build-by-number: Rearranging the real world to visualize novel architectural spaces. In *IEEE Visualization*, page 19, 2005.
3. Pravin Bhat, Stephen Ingram, and Greg Turk. Geometric texture synthesis by example. In *Eurographics Symposium on Geometry Processing*, pages 43–46.
4. Henning Biermann, Ioana Martin, Fausto Bernardini, and Denis Zorin. Cut-and-paste editing of multiresolution surfaces. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 312–321, New York, NY, USA, 2002. ACM Press.
5. Michael F. Cohen, Jonathan Shade, Stefan Hiller, and Oliver Deussen. Wang tiles for image and texture generation. *ACM Transtions on Graphics (SIGGRAPH 2003 Proceedings)*, 22(3):287–294, 2003.
6. Jean-Michel Dischler, Djamchid Ghazanfarpour, and R. Freydier. Anisotropic solid texture synthesis using orthogonal 2d views. *Comput. Graph. Forum*, 17(3):87–96, 1998.

7. Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision-Volume 2*, page 1033. IEEE Computer Society, 1999.
8. Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. Modeling by example. *ACM Trans. Graph.*, 23(3):652–663, 2004.
9. David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238. ACM Press, 1995.
10. Ken ichi Kanatani and Tsai-Chia Chou. Shape from texture: general principle. *Artif. Intell.*, 38(1):1–48, 1989.
11. Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 409–416. ACM Press/Addison-Wesley Publishing Co., 1999.
12. Takeo Igarashi, Tomer Moscovich, and John F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141, 2005.
13. John Krumm and Steven Shafer. Texture segmentation and shape in the same image. In *The Fifth International Conference on Computer Vision*, pages 121–127, June 1995.
14. Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Trans. Graph.*, 24(3):795–802, 2005.
15. Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. Graph-cut textures: image and video synthesis using graph cuts. *ACM Transactions on Graphics (Proc. Siggraph 2003)*, 22(3):277–286, 2003.
16. Ares Lagae, Olivier Dumont, and Philip Dutré. Geometry synthesis. In *Siggraph 2004 technical sketch*, 2004.
17. Sylvain Lefebvre and Hugues Hoppe. Parallel controllable texture synthesis. *ACM Trans. Graph.*, 24(3):777–786, 2005.
18. Kevin T. McDonnell, Yu-Sung Chang, and Hong Qin. Digitalsculpture: a subdivision-based approach to interactive implicit surface modeling. *Graph. Models*, 67(4):347–369, 2005.
19. Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198, New York, NY, USA, 1995. ACM Press.
20. Diego Nehab and Philip Shilane. Stratified point sampling of 3d models. In *Eurographics Symposium on Point-Based Graphics*, pages 49–56, June 2004.
21. Jos Stam. Aperiodic texture mapping. In *Technical Report R046, European Research Consortium for Informatics and Mathematics (ERCIM)*, 1997.
22. Li-Yi Wei. *Texture Synthesis by Fixed Neighborhood Searching. Ph.D. Thesis*. Stanford University, 2001.
23. Peter Wonka, Michael Wimmer, François Sillion, and William Ribarsky. Instant architecture. *ACM Trans. Graph.*, 22(3):669–677, 2003.
24. Yizhou Yu, Kun Zhou, Dong Xu, Xiaohan Shi, Hujun Bao, Baining Guo, and Heung-Yeung Shum. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph.*, 23(3):644–651, 2004.
25. Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch: an interface for sketching 3d scenes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 163–170. ACM Press, 1996.

# Smart Sticky Widgets: Pseudo-haptic Enhancements for Multi-Monitor Displays

Malcolm E. Rodgers[1], Regan L. Mandryk[2], and Kori M. Inkpen[1]

[1] Faculty of Computer Science, Dalhousie University,
Halifax, Canada, B3H 1W5
{rodgers, inkpen}@cs.dal.ca
[2] Dept. of Computer Science, University of British Columbia,
Vancouver, BC, V6T 1Z4
{mandryk}@cs.ubc.ca

**Abstract.** Multi-monitor systems are becoming increasingly popular as a way for users to expand their display surface to facilitate multitasking. However, users may have difficulty accessing user interface elements on the borders of the displays, accidentally crossing over to neighbouring displays. We present a smart pseudo-haptic technique to enhance boundary widgets in multi-monitor systems. Our technique was compared to current desktop behaviour as well as to two other pseudo-haptic approaches. All three techniques significantly reduced crossover errors; however, our smart sticky technique also significantly reduced the time to acquire targets on the secondary monitor over the other sticky techniques and was strongly preferred by users.

## 1 Introduction

As the cost and hardware entry barriers to multi-monitor systems decrease, these systems are becoming commonplace. In 2001, Grudin [10] provided a foundational account of how users in multi-monitor environments arrange and organize their information. Interviews conducted with multi-monitor users showed an overwhelming consensus of support. In addition, by displaying processes that support the primary task on a secondary monitor, cognitive load is reduced. Grudin also indicated that software designers tend to design with only the single-monitor user in mind. The end result is that inherent advantages for partitioning information and space are lost.

Researchers have begun exploring appropriate interaction styles for multi-monitor environments [3, 15]. When using multi-monitors, users often maximize application windows to fill one display as opposed to stretching one window across multiple displays [10]. There are difficulties associated with this practice however, when accessing widgets on the borders between the displays, such as scrollbars, window borders, menus, margin markers and the ‘close window’ icon. Users’ cursors may inadvertently cross over to the secondary monitor and disrupt their attention.

Numerous techniques have been proposed to aid target selection in a variety of environments (see [2] for a review). Although these techniques have been shown to be successful for the selection of isolated targets, most have difficulty with multiple,

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 194 – 205, 2006.

closely spaced targets [2]. The goal of this research is to develop a technique that will aid in the selection of boundary widgets while not inhibiting movement to a secondary display or requiring that the size of the widget be increased. This paper presents a pseudo-haptic technique which predicts whether or not a user is attempting to acquire the boundary widget. If we predict that the user is trying to acquire the scrollbar, the control-display gain is modified to simulate stickiness and assist acquisition; otherwise, the user's mouse behaves normally.

In the following section, we review previous work in the area of multi-monitor environments and pseudo-haptics; discuss the design and implementation of our 'smart sticky' technique; detail two user studies conducted to influence and evaluate our implementation; and conclude by discussing implications of our approach for multi-monitor environments and plans for future work.

## 2 Related Work

Past research has investigated how multi-monitor displays (MMDs) impact user performance. Robertson et al. [17] noted that up to 20% of Windows OS users are running MMDs from a PC or laptop. Losing the cursor, distal information access, and bezel problems were noted as the most important categories of usability issues. Czerwinski et al. [7] discussed the productivity benefits of MMDs, including improved complexity management, less time wasted on managing windows, and better performance for cognitively-loaded tasks.

Researchers are introducing innovative interface techniques for MMDs. Baudisch et al. [3] developed Mouse Ether to allow the mouse to exist in gaps between multi-monitors. Mouse Ether improved performance and was ranked well in user satisfaction. Mackinlay and Heer [15] described an approach for creating seam-aware applications for multi-monitor displays. Hutchings et al. examined MMD usage compared to a single monitor [12].

Pseudo-haptics is a software technique that creates the illusion of haptic properties such as stiffness and friction by combining the use of a passive input device with visual feedback [14]. The difference between the visual feedback (i.e. slowing down of the cursor on the screen) and the increasing reaction force applied to the input device to compensate for this disparity provides an illusion of force feedback without expensive technology [13]. Lécuyer et al. [13] illustrated the concept of sticky controls, where altering the control-display gain impacted a user's perception of a manipulated object's mass in a virtual environment.

Lécuyer et al. extended their pseudo-haptic approach to represent textures on-screen by altering the control-display gain of a circular area, creating the sensation of bumps and holes [14]. They concluded that the perception of GUI components (such as edges and buttons) could be one of the several suggested applications of a pseudo-haptic approach. Worden et al. [18] implemented sticky icons and found that when this technique was combined with area-cursors (i.e. cursors with a larger than normal activation area), older adults achieved much better targeting performance.

Other research has also manipulated the visual motion of the pointer for improved targeting [1, 3, 5]. Balakrishnan [2] provides a good overview of artificially enhancing pointer performance including pseudo-haptic techniques. Baudisch et al. [4] used a pseudo-haptic approach to improve user performance for precise positioning of graphical objects. Ahlstrom [1] recently enhanced pull-down menus with 'force fields', created by manipulating control-display gain, resulting in a decrease of 18% in selection time with commonly used input devices.

# 3 User Study 1: Acquiring Boundary Widgets

If we can reliably predict that a user is attempting to access a particular user interface element, then it would be beneficial to assist this action by making the widget 'sticky'. Several approaches have examined ways to manipulate the control-display gain to make the cursor stick to the target [2]. Despite success for isolated target acquisitions, in real usage stickiness can be problematic. If the user is not attempting to select a target, having the cursor 'stick' to it can be frustrating. In particular, near-neighbour targets can be difficult to acquire since users will increase their movement in order to move off the sticky object and overshoot their intended destination.

The initial step of our research was to investigate mouse movements on a typical MMD to inform the design of our pseudo-haptic implementation. This study had three objectives: 1) examine the problem of crossover errors on a MMD; 2) study how velocity profiles differ for aiming on single displays and MMDs; and 3) investigate how users modify their mouse movements over time to adapt for crossover errors.

## 3.1 Experiment Design and Setting

We investigated our three objectives using a target selection task that required scrolling, which was designed to mimic scrolling in a window maximized in the primary monitor. Participants were required to click a start button, then either select a target circle (if visible), or scroll down to select a target circle (see Fig.1). After clicking the start button (in one of three horizontal positions), a grid of 18 coloured circles (3-across x 6-down) appeared. The circles were orange with the exception of a blue target circle and were 40 mm in diameter. Upon selection of the target circle, the grid of circles disappeared and the start button appeared to begin the next trial.

Forty right-handed university students (28 male, 12 female) who passed a colour-blindness test participated in this study. All 40 participants used computers at least a few times per week and 13 had previously used multiple monitors, albeit rarely. A mixed design was used with one within-subjects variable (number of monitors) and two between-subjects variables (identical or different monitors; gap between monitors). In total, five multi-monitor setups were examined: 1) identical monitors, no gap; 2) identical monitors, small gap (12cm); 3) identical monitors, large gap (40cm); different monitors, small gap (12cm); and different monitors, large gap (40cm). All participants completing the tasks using both a single and a multi-monitor setup.

Each participant completed one block of 33 practice trials followed by five blocks of 33 trials (165 trials total) for both the single monitor condition and one of the multi-monitor conditions. Most trials (27/33) required scrolling (nine trials for each of three start positions). The remaining six (non-scrolling) trials were used to minimize the anticipation of scrolling. Only data from the mid-start position was used in the analyses and the practice and non-scrolling trials were not included.

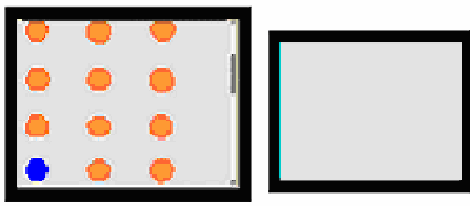


**Fig. 1.** Scrollbar acquired on the primary monitor, user scrolls down to select the target circle

The experiment was conducted using a Windows XP PC with LCD monitors, with a resolution of 1024x768, and 60hz refresh rate. In the single monitor condition, a 20” monitor was used. In the multi-monitor conditions, a 20” monitor was used as the primary display. The secondary display was positioned to the right of the primary monitor and was either an identical 20” monitor or a 17” monitor.

Computer logs were used to determine scrollbar acquisition times and to record the timestamp and X-Y positions for all mouse movements. ANOVAs ($\alpha$=.05) were performed on the data and Huynh-Feldt corrections were used if the sphericity assumption was violated. Velocity profiles were created from the mouse movements, and were used to calculate peak velocity and percentage time after peak velocity. All velocity profiles and timing metrics were calculated for movement from the start button to the scrollbar acquisition. Velocity profiles can provide more information than movement times alone. A slower movement time could be due to either a user moving more slowly (slower peak velocity), or spending more time decelerating towards his target. Precision tasks (e.g. small targets) result in users taking more time to home-in on the target, spending more time in the deceleration phase of the movement, which is reflected in a higher percent time after peak velocity.

### 3.2 Hypotheses

Our hypotheses were: (H1) in the multi-monitor conditions, participants would frequently cross over to the secondary monitor when acquiring the scrollbar; and (H2) velocity profile data would be different for the single monitor condition compared to the multi-monitor conditions. In particular, that: (H2a) movement times would be shorter in the single-monitor condition; (H2b) peak velocities would be higher in the single-monitor condition; and (H2c) percent time after peak velocity would be lower in the single-monitor condition due to the ‘backing’ that the monitor edge provides. Finally, we felt that (H3) participants in the multi-monitor condition would modify their mouse movements over time because of difficulty in acquiring the scrollbar.

### 3.3 Results

*H1: Crossover Errors:* Overall, participants in the multi-monitor conditions accident-tally crossed over to the secondary monitor in 44% (791 / 1800) of the trials. Fig.2a shows the average number of crossover errors per trial. Participants made signify-cantly more crossover errors per trial when different monitors were used (mean=0.7 errors/trial) than when identical monitors were used (mean=0.4 errors/trial),

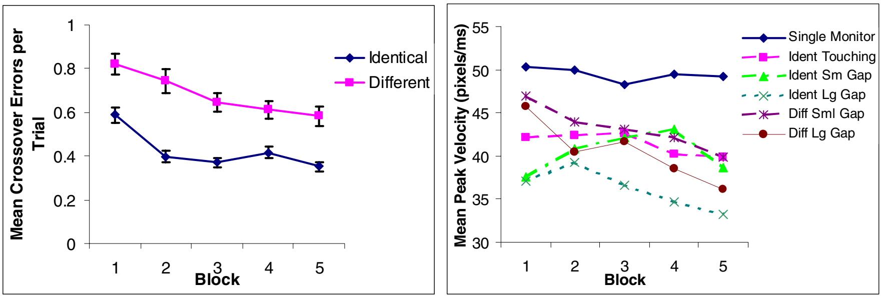



**Fig. 2.** a) Mean crossover errors per trial (±SE) across blocks for identical and different multi-monitor conditions; b) Peak velocity over blocks for single monitor and multi-monitor users

$F_{1,38}$=10.8, p<.002, $\eta_p^2$=.22. No significant differences were found based on the size of the gap between the monitors, $F_{2,37}$=1.4, p<.272, $\eta_p^2$=.068.

*H2: Scrollbar Acquisition Time and Velocity Profiles:* The scrollbar acquisition and velocity profile data is presented in Table 2. As expected, the scrollbar acquisition time (SAT) was significantly longer in the multi-monitor setups than with the single monitor setup. Mouse movement examination showed that peak velocity (PkV) was significantly lower in the multi-monitor conditions, while percent time after peak velocity (PTAPkV) was significantly higher in the multi-monitor conditions.

**Table 1.** Velocity profile data. SM: single monitor; MM: Multi-monitor; PkV: peak velocity; PTAPkV: percentage time after peak velocity; SAT: scrollbar acquisition time.

| | SM Mean(SD) | MM Mean(SD) | F | p | $\eta_p^2$ |
|---|---|---|---|---|---|
| **PkV** | 49.4 (8.3) | 40.4 (7.5) | 83.07 | .000 | .70 |
| **PTAPkV** | 59.6 (3.9) | 68.7 (2.9) | 480.68 | .000 | .93 |
| **SAT** | 1191 (137.7) | 1501 (149.4) | 200.26 | .000 | .85 |

*H3: Adaptation over Time:* In order to examine whether users' standard mouse behaviour changed over time, we examined the velocity profile data across blocks for the single monitor condition. No significant changes across blocks were found for any of the measures (SAT: $F_{4,140}$= 1.42, p<.229, $\eta_p^2$= .04; PkV: $F_{4,140}$= 1.81, p<.131, $\eta_p^2$= .05; PTAPkV: $F_{4,140}$= 0.72, p<.582, $\eta_p^2$= .02). We then examined changes across blocks in the multi-monitor conditions. Subjects significantly reduced the number of crossover errors they were making across blocks ($F_{4,140}$=6.38, p<.001, $\eta_p^2$=.15). Despite the potential time savings inherent with fewer errors, the scrollbar acquisition times did not change significantly, $F_{4,140}$= 2.13, p<.080, $\eta_p^2$= .06. Examination of the velocity profile data revealed that users' peak velocity decreased significantly across blocks, $F_{4,140}$= 10.99, p<.001, $\eta_p^2$= .24 (See Fig.2b) while PTAPkV did not change significantly, $F_{4,140}$= 0.38, p<.821, $\eta_p^2$= .01.

### 3.4 User Study 1: Summary and Discussion

Participants made crossover errors in 44% of the trials, demonstrating a significant problem with MMDs. Earlier research has indicated that users find crossover errors disruptive [16] and that loss of the cursor on the display is a significant problem [17]. We noted that the use of two different monitors resulted in significantly more crossover errors than two identical monitors. The perception of a contiguous workspace created with two identical monitors appears to aid scrollbar acquisition.

It was expected that scrollbar acquisition would be more difficult with MMDs than with a single monitor because in the single-monitor condition the edge of the display acts as a boundary, enabling easier selection. This difficulty with MMDs was reflected in higher movement times (H2a), lower peak velocities (H2b), and higher percentage time after peak velocity (H2c). Because users tend to be bothered by crossover errors [7, 16], we felt that our participants would modify their mouse movements in an attempt to minimize these errors (H3). Over time, participants moved slower (lower PkV), in order to reduce the crossover errors. These lower *movement speeds* were also reflected in a slowing of *movement times* over the course of the experiment. This study showed that users make crossover errors in almost half of their scrollbar acquisitions, and that these errors impacted the kinematics of mouse behaviour. Using this information, we developed a new widget acquisition aid.

## 4 Smart Sticky Widget Implementation

Given the results from our first study, we felt that a pseudo-haptic technique would be a good approach for aiding in the selection of boundary widgets. However, with MMDs, users often cross to the secondary display intentionally, and our technique should not impede these types of movements. Therefore, we chose to implement a 'smart' sticky widget.

Our approach uses mouse velocity to predict user intent for selecting a particular widget. If we predict that the user is trying to acquire the widget, we apply stickiness. If we predict that the user is not trying to acquire the widget, no stickiness is applied. To improve performance for MMDs, we applied our technique to a scrollbar. This technique could easily be applied to any widget, although minor adjustments to the velocity parameters may be needed.

When the scrollbar received a mouse enter event, our system determined whether stickiness should be applied. This determination was based upon whether the aggregate deceleration of the mouse cursor from the time of peak velocity was higher than a certain threshold (0.15 pixels/ms), combined with whether the user was moving below a certain speed threshold (1.5 pixels/ms). The speed threshold requirement was disregarded if the user's aggregate deceleration was high (equal to two times the deceleration threshold requirement). If the predictive model determined that the user was attempting to access the scrollbar, the mouse gain was decreased to 10% of its original value for the duration of mouse movement over the widget.

This implementation was informed by empirical data from Study 1 by examining velocity profile plots and how users' movements differed between successful scrollbar acquisitions and acquisition attempts resulting in crossover errors. We noticed patterns in the peak velocity and deceleration phases when users were targeting the scrollbar. In particular, when users overshot the scrollbar, it was usually due to deceleration occurring too late in the movement, or at too slow of a rate, or because their peak velocity was too high. The implementation and thresholds were further fine-tuned with pilot testing.

# 5 User Study 2: Comparison of Sticky Techniques in MMDs

The goal of the second study was to examine how our sticky widget software implementation performed in comparison to a control condition (no mouse modifications), as well as to two other sticky techniques from the literature. In particular, we wanted to compare performance for both scrollbar acquisition and selection of near-neighbour targets.

We compared our smart sticky technique to the two pseudo-haptic implementations used in Cockburn and Firth's [6] analysis of acquisition of small targets. The first mimicked Worden et al's [18] sticky icon approach where the mouse control-display gain was reduced to 10% of its original value when inside a sticky target (we refer to this technique as sludge sticky). The second approach [6] examined mouse movements within a target. If cursor position for consecutive mouse movement events passed a threshold distance (i.e. moving quickly), the cursor snapped out of the target. If this did not occur, the cursor warped back to the centre of the item (we refer to this technique as snap-out sticky). The main difference between sludge and snap-out sticky is that a slow motion will eventually allow the cursor to leave the target with the sludge technique whereas it will remain stuck with the snap-out technique.

## 5.1 Experiment Design and Setting

Two tasks were used in this study. The first was a scrolling task (shown in Fig.3a) adapted from Hinckley et al's [11] representative scrolling task. Participants first selected a start button that was located in one of three horizontal positions across the top of the screen. They then acquired the scrollbar and scrolled down until a blue target square, 32 pixels high (15mm) was positioned within a target frame (two horizontal lines, 88 pixels apart (35mm), positioned in the centre of the screen). When the participant released the mouse button, a sound indicated success or failure and the trial was complete. Participants were told to only use drag scrolling (i.e. no scrolling by clicking on various positions on the scrollbar or mouse wheel scrolling).

The second task (Fig.3b) was a near-neighbour target selection task with targets appearing on the secondary monitor. Participants selected a start button located in one of three horizontal positions across the top of the screen. Next, three squares appeared on the secondary monitor (two gray and one blue square), placed in the same vertical position as the start squares (46, 246, and 446 pixels away from the left edge of the secondary screen). Participants were required to select the blue target square.

Twelve right-handed university students (11 male, 1 female) who passed a colour-blindness test participated. All 12 participants used computers every day and 8 had previously used multiple monitors. A within subjects design was used with four conditions: no stickiness (control), smart sticky, sludge sticky, and snap-out sticky.

Each participant completed one block of 27 practice trials for both the scrolling and target selection task. For each condition, participants completed 3 blocks of 27 trials for the scrolling task (3 start pos. x 3 scrolling targets x 3 repeated trials) and 3 blocks of 27 trials for the near-neighbour selection task (3 start pos. x 3 near-neighbour target selections x 3 repeated trials). Only data from the mid-start position was used in the analyses and practice trials were not included. Participants ranked their preference for each technique. All trials were completed in a one-hour session.

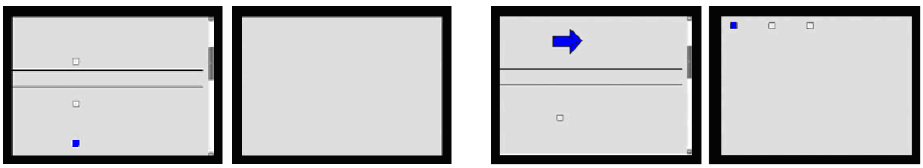


**Fig. 3.** a) Scrolling task, b) Near-neighbour target selection task

Given the results from Study 1, we were concerned that users would adapt their mouse movements to try to reduce the number of crossover errors. Therefore, we had half of the participants perform the control condition (no stickiness) first with the other half performing the control condition last. The order of the sticky conditions was fully counterbalanced. The experiment was run using a Windows XP PC with LCD monitors, at a resolution of 1024x768 pixels per monitor, and 60hz refresh rate. Two identical 20" monitors were used with a slight gap (12cm) between them. Computer logs were used to gather scrollbar acquisition time, crossovers for the scrolling task, and target selection time for the near-neighbour selection task. In addition, logs recorded the number of times stickiness was applied for the smart sticky technique. Scrollbar acquisition times and near-neighbour target selection times were analyzed using ANOVAs ($\alpha$=.05), and Huynh-Feldt corrections were used if the sphericity assumption was violated. Questionnaire data was analyzed using a Friedman two-way ANOVA ($\alpha$=.05). Wilcoxon matched-pairs signed-ranks tests with Bonferroni adjustments were used for all post-hoc pair wise comparisons.

### 5.2 Hypotheses

Our hypotheses were: (H1) scrollbar acquisition time would be higher in the control condition than with any of the stickiness techniques; (H2) Accidental crossover errors would be higher in the control condition than with any of the stickiness techniques; (H3) Our smart sticky technique would outperform the other sticky techniques, in terms of selection time and user preference, for acquiring near-neighbour targets.

### 5.3 Results

Table 3 shows the mean scrollbar acquisition times and the number of crossover errors for Task 1, and the mean selection times for the target selection task for Task 2. The order of conditions had no significant effect.

**Table 2.** Mean times for scrollbar acquisition (SAT) and percentage of crossover error trials in Task 1. Mean movement time to select the near-neighbour target in Task 2.

| Condition | SAT ms (SD) | Crossover Errors % trials (total #) | MT ms (SD) |
|---|---|---|---|
| **No Stickiness** | 1363 (176) | 21% (69) | 1152 (126) |
| **Smart Sticky** | 1214 (170) | 12% (39) | 1189 (178) |
| **Sludge Sticky** | 1128 (198) | 8% (26) | 1382 (145) |
| **Snap-out Sticky** | 1132 (212) | 11% (36) | 1607 (260) |

*H1: Scrollbar Acquisition Time:* For Task 1, an overall main effect of condition was found for scrollbar acquisition time ($F_{3,30}$=24.29, p<.001, $\eta_p^2$=.71) with all three sticky techniques being significantly faster than the control condition (p<.001) (Fig.4a). The pair wise analyses revealed that our smart sticky implementation was significantly slower than the sludge sticky technique (p=.029). Examining the scrollbar acquisition times across blocks revealed a significant main effect of block for the control condition with users significantly reducing their SATs when no stickiness was provided ($F_{2,20}$=12.70, p<.001, $\eta_p^2$=.56). In particular, the first block was significantly slower than the second and third (p<.01). Participants using the snap-out sticky technique also significantly reduced their SATs across blocks, $F_{2,20}$=4.91, p<.018, $\eta_p^2$=.33; however, no differences were found across blocks for the smart and sludge techniques, $F_{2,20}$=0.86, p<.44, $\eta_p^2$=.08, and $F_{2,20}$=0.25,p<.78,$\eta_p^2$=.02, respectively.

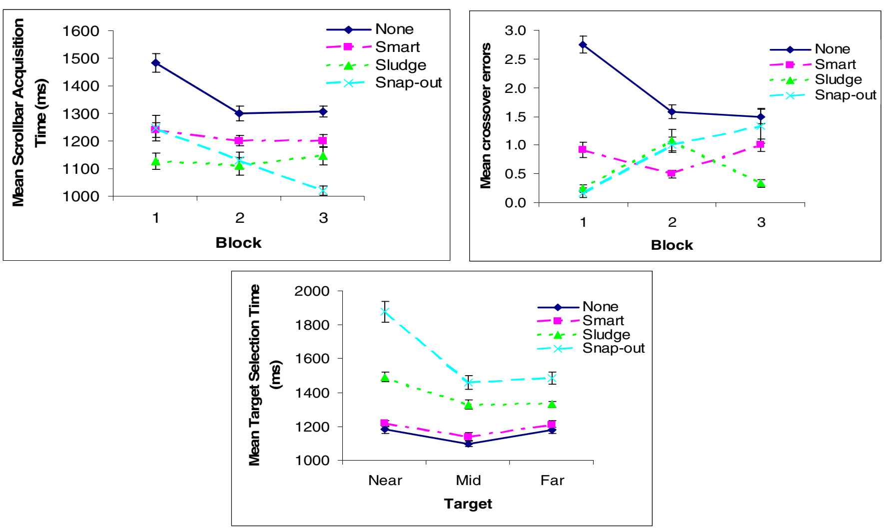



**Fig. 4.** a) Mean scrollbar acquisition time (±SE) for each technique across blocks, b) Average crossover errors (±SE) for each technique, separated by block, c) Average selection times (±SE) for each near-neighbour target on the secondary display, separated by target location

*H2: Accidental Crossover Errors:* A significant main effect of technique was found for the number of crossover errors with all three sticky techniques reducing crossover errors by more than 50% ($F_{3,30}$= 17.43, p<.001, $\eta_p^2$=.64). Post hoc pair wise comparisons revealed that all three techniques significantly reduced crossover errors over the default mouse condition (p<.01), although no significant differences were found between the three sticky techniques (Fig.4b).

Examining the crossover errors across blocks revealed a significant main effect of block for the control condition with users significantly reducing their crossover errors across blocks when no stickiness was provided. In particular, blocks 2 and 3 had significantly fewer crossover errors than block 1. No significant differences across blocks were found for the three stickiness techniques.

*H3: Near Neighbour Task:* A significant main effect of technique was found for target selection time on the secondary monitor, $F_{3,03}$=48.01, p<.001, $\eta_p^2$=.83 (Fig.4c). Post-hoc comparisons revealed no significant differences between the control condition and smart stickiness (p<1.0), while both were significantly faster than sludge and snap-out (p<.001). Sludge was also significantly faster than snap-out (p<.016).

A significant interaction effect was found between technique and block ($F_{3,30}$=48.0, p<.001, $\eta_p^2$=.83). Further analysis revealed that snap-out sticky was the only technique whose target selection time differed significantly across blocks ($F_{2,20}$=6.11, p<.008, $\eta_p^2$=.38), with block 1 being significantly slower than block 3 (p<.05). For near-neighbour selection, a significant interaction was found between target location and sticky technique on selection time ($F_{6,60}$= 5.06, p<.001, $\eta_p^2$=.34), (Fig.4c). Further analysis revealed that the nearest neighbour (closest target) required significantly longer selection times than the other two targets (p<.05) for the sludge and snap-out sticky techniques. This pattern is opposite to what Fitts' Law [9] predicts, and demonstrates unique requirements for aiming in multi-monitor environments.

### 5.3.1 Accuracy of Smart Sticky Prediction

For the scrolling task, smart stickiness was applied 65% of the time, on average. It is unclear how often our users needed stickiness to help them acquire the scrollbar, although in Study 1, participants made crossover errors in approximately 43% of the trials. For the near-neighbour selection task, smart stickiness was applied, on average, in 12% of the trials. Ideally, stickiness would never be applied when users are intentionally crossing between displays; however, the accuracy of our prediction ranged from being applied only 3% of the time for one participant to almost 20% of the time for another participant. Closer examination of this data revealed that stickiness was erroneously applied more often for selection of the nearest-neighbour target (24%), compared to the mid (9%), and far (2%) targets.

### 5.3.2 Post-condition Questionnaire Data

Participants were asked to provide an overall ranking of the techniques. Most participants preferred the smart sticky technique (8/12) while the majority of participants liked the snap-out technique the least (9/12). Participants were also asked to rate each technique on a five-point scale (1=low, 5=high) in terms of speed and accuracy. On average, participants felt that they were able to select the scrollbar most quickly using the sludge sticky technique (4.33), followed by snap-out (4.08), smart

(3.67), and no stickiness (3.33). This overall difference was marginally significant ($\chi^2_{3,12}$=7.81 p<.049). A similar trend was found for how accurately participants felt they could select the scrollbar, although these differences were not significant ($\chi^2_{3,12}$=3.19 p<.363). On average, sludge sticky was rated the most accurate (4.18), followed by smart and snap-out (3.73) and then no stickiness (3.36).

When asked if it was easy to accidentally cross over to the secondary monitor during the scrolling task, participants agreed with this when no stickiness was applied (3.58) followed by smart sticky (3.0), sludge (2.25) and snap-out (1.92) on a scale from 1 (strongly disagree) to 5 (strongly agree). This difference was statistically significant ($\chi^2_{3,12}$=16.98 p<.001) with it being significantly easier to accidentally cross over when no stickiness was applied compared to sludge (p<.007) and snap-out (p<.008). For the near-neighbour selection task, participants indicated that it was easiest to cross over when no stickiness (4.67) was applied, followed by smart stickiness (4.58). Sludge (3.25) and snap-out (1.92) were felt to be impede crossing between displays. This difference was statistically significant ($\chi^2_{3,12}$=25.55 p<.001) with no stickiness and smart sticky being rated as easier than snap-out (p<.003).

### 5.4 User Study 2: Summary and Discussion

Participants felt that the snap-out implementation was very effective for the primary monitor task, but too strong in general. On more than one occasion, the participant would over-shoot the intended target, and while readjusting would get caught by the sticky scrollbar, requiring a second snap-out action. Perhaps this type of 'stickiness' is best suited for configurations with no near-neighbour targets.

The sludge approach performed well for the primary task, however was significantly slower for near-neighbour selection tasks. In our implementation, the scrollbar was the only sticky widget. In an actual application, one could imagine multiple sticky components having a much larger penalty on targeting performance. Our smart sticky technique only applies stickiness to the intended target, thus making it easier for users. Our implementation proved to be a good balance for performing both the scrolling task on the primary monitor as well as the target selection task on the secondary monitor. Crossover errors were reduced by 54% against no modify-cation alone, allowing faster acquisition time of the target scrollbar. When the user attempted to cross between displays, our technique was only applied 12% of the time, resulting in selection times that didn't differ from the control condition, even for the near-neighbour target, which was only 2 cm away.

## 6 Conclusions and Future Work

We presented a smart pseudo-haptic technique that aids users with the acquisition of small targets on the borders of a MMD. Our implementation is grounded in the marked differences in velocity profiles for users in MMDs as compared to single monitor systems. The smart sticky implementation effectively tackles the problem of accidentally crossing over to the secondary display of a MMD, without impeding near-neighbour target acquisition, as in other techniques. It accomplishes this while not requiring any training or increase in displayed visual information. In fact, users would not even need to be aware of the implementation to see performance benefits.

Although our implementation performs well, our prediction algorithm could be improved to further reduce accidental crossovers, thus improving boundary widget selection times. Our technique also needs to be extended to work with a variety of widget sizes, and movement amplitudes. Further work includes combining our approach with other multi-monitor techniques, such as Mouse Ether, and testing in more realistic usage environments for a larger variety and density of software widgets.

## References


1. Ahlstrom, D. (2005). Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields. *CHI 2005*. 61-70.
2. Balakrishnan, R. (2004). "Beating" Fitts' law: Virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies*, 61(6). 857-874.
3. Baudisch, P, Cutrell, E, Hinckley, K, & Gruen, R. (2004). Mouse Ether: Accelerating the Acquisition of Targets Across Multi-Monitor Displays. *CHI 2004*. 1379-1382.
4. Baudisch, P., Cutrell, E., Hinckley, K., & Eversole, A. (2005). Snap-and-go: Helping Users Align Objects Without the Modality of Traditional Snapping. *CHI 2005*. 301-310.
5. Blanch, R., Guiard, Y., & Beaudouin-Lafon, M. (2004). Semantic Pointing: Improving target acquisition with control-display ratio adaptation. *CHI 2004*. 519-526.
6. Cockburn, A. & Firth, A. (2003). Improving the Acquisition of Small Targets *People and Computers XVII*. 181-196.
7. Czerwinski, M, Smith, G, Regan, T, Meyers, B, Robertson, G, & Starkweather, G. (2003). Toward characterizing the productivity benefits of very large displays. *Interact* 2003. 9-16.
8. Douglas, SA, Kirkpatrick, AE, & MacKenzie, IS. (1999). Testing pointing device performance and user assessment with the ISO 9241, Part 9 standard. *CHI 1999*. 215-222.
9. Fitts, PM (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*. 47. 381-391.
10. Grudin, J. (2001). Partitioning digital worlds: focal and peripheral awareness in multiple monitor use. *CHI 2001*. 458-465.
11. Hinckley, K, Cutrell, E, Bathiche, S, & Muss, T. (2002). Quantitative Analysis of Scrolling Techniques. *CHI 2002*. 65-72.
12. Hutchings, D, Meyers, B, Smith, G, Czerwinski, M, & Robertson, G. (2004). Display Space Usage and Window Management Operation Comparisons between Single Monitor and Multiple Monitor Users. *Advanced visual interfaces*. 32-39.
13. Lécuyer, A, Coquillart, S, & Kheddar, A. (2000). Pseudo-Haptic Feedback: Can Isometric Input Devices Simulate Force Feedback? *IEEE VR2000*. 18–22.
14. Lécuyer, A, Burkhardt, JM, & Etienne, L. (2004). Feeling bumps and holes without a haptic interface: the perception of pseudo-haptic textures. *CHI 2004*. 239-246.
15. Mackinlay, J, & Heer, J. (2004). Wideband displays: mitigating multiple monitor seams. *CHI 2004 Ext. Abstracts*. 1521-1524.
16. Mandryk, RL., Rodgers, M., & Inkpen, K. (2004). Sticky Widgets: Pseudo-haptic Widget Enhancements for Multi-Monitor Displays. *CHI 2005 Ext. Abstracts*. 1621-1624.
17. Robertson, G, Czerwinski, M, Baudisch, P, Meyers, B, Robbins, D, Smith, G, & Tan, D. (2005). Large Display User Experience. In *IEEE Computer Graphics & Applications,* Special Issue on Large Displays. July/August 2005. 44-51.
18. Worden, A, Walker, N, Bharat, K, & Hudson, S. (1997). Making Computers Easier for Older Adults to Use: Area Cursors and Sticky Icons. *CHI 1997*. 266-271.

# The EnLighTable: Design of Affordances to Support Collaborative Creativity

Lucia Terrenghi, Torsten Fritsche, and Andreas Butz

University of Munich, Media Informatics
Amalienstrasse 17, 80333 Munich, Germany
{lucia.terrenghi, utz}@ifi.lmu.de,
fritsche@cip.ifi.lmu.de

**Abstract.** In this paper we discuss some interface design concepts for supporting and enhancing collaborative creativity in multi-user interactive environments. We focus on the design of affordances for direct manipulation and collaboration on table-top displays. As an application of our concepts, we introduce the EnLighTable, an appliance for creative teamwork in the selection of pictures and layout design based on a table-top touch-sensitive display. We present our rationale for the design of affordances which metaphorically relate to artifacts – and map to gestures – of the physical world. Finally we discuss the results of our first design iteration and introduce a possible future extension of the table-top appliance to other devices of an instrumented environment.

## 1 Introduction

In the vision of ubiquitous computing, the walls and surfaces of our everyday life environments might have computing, display, and interaction capabilities [26]. This will change our mental model of a computer, which is currently mostly associated with a desktop workstation. Computers in the everyday life won't necessarily be *Personal* Computers: their appearance, on a physical as well as on a digital level, will need to support novel conceptual models concerning what a computer is, and how to work with it, individually and/or cooperatively.

The embodiment [8] of interaction in our physical as well as our social settings will create novel, more natural and casual contexts of interaction, in which people will deal with digital information in a more immediate way. Shared displays can support such a transition from personal to social interaction technically. On a conceptual level, we need visualization and interaction techniques which effectively support users' interaction among each other and with the machine. Novel contexts of co-located collaboration might also alter traditional communication and work-flows, thus affecting collaborative creativity.

In this paper we describe the EnLighTable. It is an appliance based on a table-top touch-sensitive display for creative teamwork in the selection of pictures and layout design, e.g. in advertising agencies. It enables multiple users to simultaneously manipulate digital pictures of a shared collection, and rapidly create and edit simple page layouts. In section 2 we consider the table as an information display and look at the work on interaction techniques that has been done in this field. In section 3 we

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 206 – 217, 2006.

consider the design of affordances for multi-user environments and the main issues to take into account in our scenario. In chapter 4 we introduce the EnLighTable appliance and motivate our design goals. We then present our design solutions as we report on our design of affordances for multi-user direct manipulation of digital information on a table-top display. Finally, we present the results of the first design iteration of the project and discuss future work.

## 2 The Table as an Information Display, Related Work

Tables are used for many daily activities in societies of most cultures. Coffee tables, desktops, meeting tables support different social as well as individual activities, leisure or work related. Tables can easily be shared by several people, who then comfortably face each other with the table between them. This allows easy verbal and gesture-supported face to face communication with the table surface permanently in sight. Tables create a shared space for the collaborative manipulation of objects. Scott et al. found [20] that humans tend to divide this space into different functional areas. In particular, the middle of a table is often used as a shared area for the exchange of objects, while areas closer to a specific user are considered increasingly private by this user. When dealing with rotation-dependent objects, such as text or pictures, users tend to reorient them for their own viewing direction, which can cause conflicts of interest with other users. On the other hand, when a user reorients an object towards another person, this action shows her intention to discuss the object with this person.

Since Wellner's digital desk [27], Krüger's responsive workbench [17] and Ullmer's Metadesk [25], research has investigated various forms of interaction with table-tops for a wide variety of tasks [18], [22], [19]. Closest to our own work are Hinrichs' Interface currents [14] and Apted's SharePic [1] which both deal with the task of collaborative picture selection.

Interface Currents are circular streams in which the images float around the table. As they pass each user, they face her in the right orientation, which nicely solves the orientation problem. Pictures can be fished from the stream and dragged to a private bank between the stream and the table's border. While this metaphor does provide affordances for the manipulation of the images, it is a general purpose metaphor and as such rather unrelated to the physical manipulation of photos. Furthermore, it does not support spatial memory and the possibility to share a consistent visual landscape among team members, because the pictures continuously flow in the currents.

SharePic emphasizes the different functional areas of the table, but leaves the orientation problem to the user. Pictures can be freely rotated, scaled and annotated, but no automatic reorientation is done. While this behavior is closer to the physical world, in which photos have to be rotated by the user as well, it doesn't use the additional capabilities of the new medium in this respect.

## 3 Affordances for Multi-user Interactive Environments

In Terrenghi [24] we presented our work on design of affordances for direct manipulation of digital information. We introduced an interaction paradigm in which surfaces act as interfaces and hands as control devices. Thus, we discussed the design of graphical user interfaces supporting hands-on direct manipulation of digital information in the environment. In order to support users' gesture-based interaction, we suggested the use of affordances. Their design can either rely on the visual appearance of the information, or on the metaphoric link to real world objects and their affordances in the physical world.

The work we describe in this paper builds on those insights and extends the design space to multi-user interaction on a large horizontal display. Specifically, we look at the manipulation of digital picture collections in co-located collaboration. The collaborative setting adds some complexity to the design of affordances, such as:

- Physical arrangement of multiple users around the table;
- Issues of territoriality, reachability, sharability and passing on;
- Orientation of information.

We distinguish between *physical* and *cognitive* affordances, as suggested in [13]. Physical affordances are design features which help users to execute a physical action in the interface. Some physical affordances are provided by the table itself, as an artifact: It allows several users to gather around it; it allows the display of information and at the same time supports physical objects; it naturally fosters two-handed interaction by supporting both forearms. Cognitive affordances, on the contrary, are design features which tell users something, mostly about a physical action they can do with the interface. To give an example from the physical world, a cognitive affordance could be the beer mat we put on the table. It offers a visual indication of where to place the beer glass so as to avoid drops on the table. The visibility of the beer mat and its physical placement on the table tell users what to do with the glass on a cognitive level. Most cognitive affordances are based on conventions, and there is nothing intrinsic in the appearance of the artifact that suggests what it is for.

On top of the distinction between cognitive and physical affordances, social affordances have been defined [16]. These are properties of collaborative environments which serve as facilitators of the social context. Thy enable and provoke social interaction between a member and the group (i.e., perception and action coupling). Perception and action are then the results of the intentions of the group member and of the social affordances of the collaborative environment. As an example in the physical world, a couch provides social affordances for people to sit next to each other. Gaver [10] explores an ecological approach to social interaction using the concept of affordances to describe material properties of the environment that affect how people interact. From his perspective, the design of new technologies can rely on existing affordances provided by the physical environment to elucidate and analyze human behaviors in a new light.

This means that our design needs to consider the following aspects:

- existing physical, as well as social affordances provided by the table as a physical artifact,
- design of cognitive affordances for the manipulation of digital information,
- design of social affordances for collaboration.

This implies an approach to the design and study of hybrid (i.e., physical and digital) technological artifacts that considers their social context of use. In Grint and

Woolgar [11] "Technologies, do not 'by themselves' tell us what they are or what they are capable of. Instead, capabilities – what, for example, a machine will do – are attributed to the machine by humans. Our knowledge of technology is in this sense essentially social".

In the next section we describe the EnLighTable, a multi-user table-top display appliance. In its design, we tried to merge the principles above, in order to create an interactive environment which supports collaborative creativity.

## 4 The EnLighTable Appliance

The EnLighTable is an appliance which aims to support picture selection and layout design in creative teamwork, e.g. in advertising agencies, publishing companies, or catalogue production companies. The cheap production of digital pictures causes a dramatic growth of picture collections. Creative directors, art directors, and graphic designers often need to select the pictures for a defined communication project from a large collection. Selecting the right picture for a flyer, a catalogue or an advertising campaign usually involves several individuals in the discussion, in iterative phases of communication. The patterns of communication may change according to the size of the company and of the project, the hierarchical organization of the company, or the source of the picture collection (cf. Section 6.1). Often, a strategic concept for the communication is initially defined together in a team, and then the different professionals "migrate" to their workstations to work on their individual sub-tasks. Afterwards, they meet again with print-outs of ideas, or move to each other's PCs to visualize and discuss ideas. Most of the creative phase of selecting pictures happens in parallel, in different locations, with limited communication.

Our design aims to explore the potential of digital table-top displays to support collaborative creativity in co-located work. In Fischer [9] Creativity occurs in the relationship between an individual and a society, and between an individual and his or her technical environment. Appropriate socio-technical settings, at the same time, can amplify the outcome of a group of creative people by both augmenting individual creativities and multiplying rather than simply summing up individual creativities. From this perspective the physical, social and interactive contexts acquire a main role in determining cognitive processes. In [23] collaborative creativity is described as a social and communicative transaction between people who in some way share a mutual goal. In this sense an act which is objectively creative is not necessarily "collaboratively creative" if it does not support a shared goal or is not recognized as a contribution by the other team members.

The layout of the environment plays a major role in the perception of ideas [22][23]. Furthermore, visibility of action is a main design principle for embodied interaction [8]. It provides awareness of what other colleagues are doing and how the actions of group members affect the shared artifacts. This relies on existing theories from CSCW [6] [7]. Group awareness (i.e. the condition where members perceive the presence of other group members and the possibility to communicate with them), seems to provide chances for informal communication. Informal communication facilitates the transfer of essential information related to task-specific activities. Most interactions in the work environment take place during chance encounter.

In the EnLighTable appliance we expect the mutual visibility and awareness of others' interactions to provide affordances for discussion and collaborative creativity in the early stage of layout-design. Therefore the design of the appliance on our table-top display focused on the following requirements:

- Up to 3 users can simultaneously see and reach a shared picture collection: each user can see what others select and drag objects out of the collection
- Each user can pick some pictures out of such a collection, without affecting the others' range of selection: thus, each user works on copies of picture items, while the originals are maintained in the shared collection
- Each user can operate basic manipulation tasks on the selected pictures, and save the changes as edited instances
- Each user can draft page layouts on the table and place pictures within the lay-out frames.

In the following section we describe how the design of the graphical user interface tries to meet the requirements above.

## 5 Design of Affordances: Borrowing from the Physical World

The graphical user interface for direct manipulation of information items by multiple users was designed on three different levels:

- design of information arrangement for social interaction
- design of information items for direct manipulation
- design of a virtual tool for additional editing and manipulation of information items.

The three levels are discussed separately in the following.

### 5.1 Information Arrangement for Social Interaction

The arrangement of information items on the table needed to cope with the constraints and affordances of the horizontal surface. When users sit at the sides of a table, personal areas of interaction are implicitly determined by people's area of reach and visual angle, as in the normal physical space. Metaphorically, we adopted an arrangement similar to the one of a set table (see fig. 1, (a)). The perception of a personal area of interaction is suggested by the pre-defined placement of three *Imagetools* (cf. section 5.3) oriented towards the sides of the table. In analogy to plates, this is expected to suggest "guests" where to sit, i.e., their personal area of interaction. In the center, a bigger shared container of information is displayed. It contains the thumbnails of a shared picture collection, e.g. the pictures of a photo shooting or of a shared library. The tray can be dragged with a simple gesture towards the personal area of interaction (see fig. 1, (b)). When such a "shared tray" is dragged in the direction of one of the *Imagetools*, the pictures reorient towards the dragging user. In this way, she can have a better view of the pictures, she can scroll to see more, and drag the preferred ones out of it: i.e., she can "serve" herself with the desired information items. After the user ceases interacting with the shared tray, the latter automatically returns to the original orientation and location. In this way, a consistent spatial arrangement of the pictures is mostly maintained, thus supporting spatial memory for every user. On the other hand, the personal reorientation of the picture collection is allowed, which is important for visualizing and selecting pictures in the design phase. In the same tray, which plays the role of a shared resource, a bin is available. In the same interaction manner, users can move the shared tray towards them and drag the copies which are lying on the table to the bin in order to delete them.

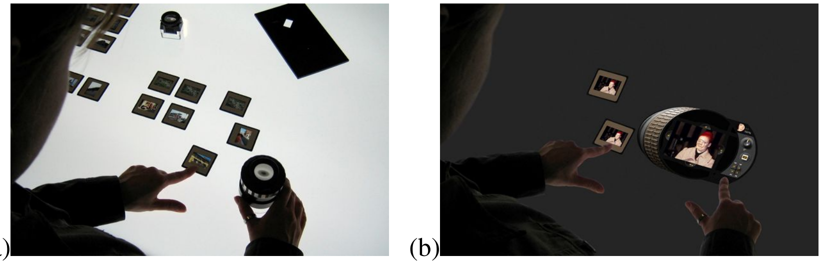
(a)  (b) 

**Fig. 1.** Affordances for social interaction on the table. (a) Arrangement of physical artifacts in the everyday life. (b) Arrangement of digital information items on the EnLighTable appliance.

### 5.2 Information Items for Direct Manipulation

In the real world, negatives or slides are the original sources for analogue photography. In design and photographic working scenarios, slide film is often used and slides are projected, printed, scanned, i.e., edited and processed. Furthermore, they are often arranged on light tables, so as to visualize and compare multiple pictures simultaneously (see fig. 2, (a)). The cognitive relevance of humans' direct manipulation and spatial arrangement of artifacts have been studied by Kirsch [15]. In his study on people's intelligent use of space, Kirsch classifies spatial arrangement that simplify choice, perception and internal computation. In this sense, external representation and spatial arrangement support creativity as they enable the visualization of different possible solutions.

The white plastic frames of physical slides afford their manipulation, so as to handle them without ruining or occluding the negative. In analogy to this conceptual model, our information items visually resemble slides (see fig. 2, (b)). The size of our virtual slides (75x75 pixels) is such that their picture content can easily be recognized at a glance. When displayed in the shared tray, the slide frames are white, as their physical counterparts usually are. In the white area the copyright or source of the picture can be displayed, just as the physical slide frame can be labeled with a pen. Consistently with this metaphor, our digital slides represent the original source of information and provide affordances for manipulation and editing. When users touch the original slide on the shared tray and drag it to empty areas of the table surface, a copy of the slide is created. The copy receives a yellow frame, so as to visually differ from the original source and to remind the user that she is manipulating a copy.



(a) (b)

**Fig. 2.** Affordances for manipulation of pictures. (a) Physical slides on a light table. (b) Digital slides for the representation of image content in the EnLighTable appliance.

### 5.3 Virtual Tools for Direct Manipulation of Digital Information

Graphic designers have accustomed themselves to edit picture files with graphic desktop applications, such as Adobe Photoshop. The graphical user interface of this kind of applications usually provides different icons, representing different tools, e.g. a lasso for drawing a selection, or a bucket to fill color regions. The appearance of these icons suggests their function, but not the type of interaction. When a user, for example, selects the lasso in the toolbar and moves the pointer to the picture area, the cursor changes into a lasso. In order to draw a selection, the user is supposed to click and drag, while keeping the mouse button pressed. In contrast to this, after selecting the bucket from the toolbar and positioning the pointer on the picture, a single click will change the color of the pixels within the same color region. In terms of affordances, this means that no cognitive affordances are provided by the interface to support different mouse-based manipulation techniques.

A tighter mapping between visual representation and functionality is provided by Kai's Power Tools [5]. Based on the metaphor of a toolglass or magic lens [2], these Plug-Ins for Adobe Photoshop and Corel Photopaint enable the direct application and visualization of filters on the image. As a virtual tool, the lens can be dragged over different areas of the picture with the mouse, and different filter parameters can be entered with the keyboard. Moving even more towards a real direct manipulation, and beyond the WIMP paradigm of the desktop, HabilisDraw [21][3] provides a set of virtual tools. These are controlled by fingers on a DiamondTouch table-top display. In this project, tools are just a subcategory of objects: therefore objects and tools share a common manipulation vocabulary (e.g. for moving, rotating, picking up). This means that the actions that can be performed with the different tools are not necessarily coupled to visual cues suggesting how the specific tool needs to be manipulated. Rather, a manipulation vocabulary is predefined, which is the same for every object. To make an example, in HabilisDraw virtual pens are manipulated in the same way as virtual tapes, although they have different functionalities and the effect of their application has different feedbacks.

Our *Imagetool* is a digital virtual tool which supports basic editing of pictures. Similarly to Kai's Power Tools [5], it relies on the conceptual model of a magic lens, which in our case is controlled by two hands directly on the surface of the table. Differently from the concept of HabilisDraw, our virtual tool provides cognitive affordances for direct manipulation relying on the way we manipulate certain physical objects. In figure 3 (a), for example, the zooming gear on the left side of the tool can be “scrolled” with a continuous movement of one hand, as we would do with a physical photographic lens. Discrete interaction, such as tapping, is suggested by the 3D effect of the buttons for mirroring and saving changes, on the right side of the tool (see fig. 3 (b), labels 3 and 4). The image can be cropped by dragging the semi-opaque ledgers within the lens, which resemble the blades of a four bladed easel used in darkrooms (see fig. 3 (a), and (b) label 2). Cognitive affordances are designed to suggest to the user where to position her finger for interaction. The interface supports space multiplexed input control, in the sense that different controls can be mapped to different functions, each independently accessible. We consider the different micro- and macro-metric specialized work of respectively dominant and non-dominant hands [12][4]. The *Imagetool* can be placed with one hand, likely the non-dominant one, in order to identify the frame of reference for the dominant hand. We assume that users, particularly those in our target group of designers, might use a real pen with the dominant hand for more precise interaction. Therefore, on top of the cues for interaction with fingers, we suggest points for the placement of pen-tips by yellow dots. Other basic functionalities are provided, such as rotation (see fig. 3 (b), label 5) and visualization of picture elements in 1:1 scale (see fig. 3 (b), label 6).

The edited pictures can be saved, and remain on the shared surface of the table. Designers can also draft frames for a layout (see fig. 4 (a)). They can place and resize pictures in their layout by simply dragging the pictures inside the frames.

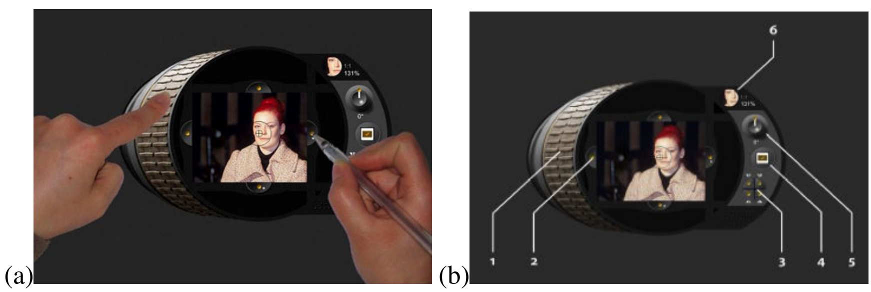



**Fig. 3.** (a) Bi-manual interaction with the *Imagetoool*. (b) Labels 1 to 5 show the interactive areas of the *Imagetool*. Label 6 indicates the area where the picture is displayed in 1:1 scale, and zooming factor is shown.

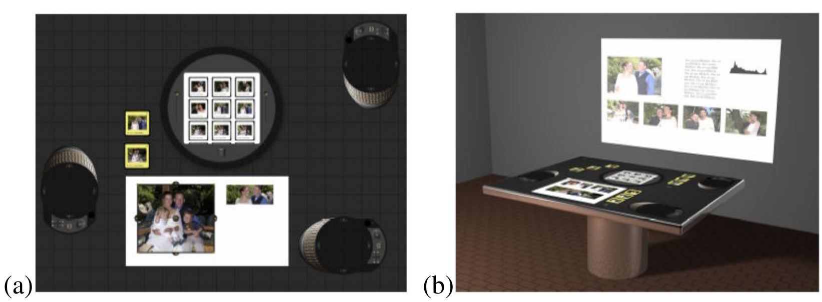



**Fig. 4.** Layout design on the EnLighTable: (a) In the current prototype; (b) In the extended version of the appliance, in which we plan to integrate vertical displays

# 6 Current Prototype and Future Work

Our initial prototype of the EnLighTable was run on the touch-sensitive table of our lab. This table contains a NEC LCD display with a resolution of 1366x768 pixels, over which a SmartTech Actalyst touch panel tracks fingers by 4 infrared cameras placed in the corners. The Prototype was implemented using Flash and the mouse driver provided by the touch panel. While this setup didn't provide full multi-handed input, it allowed us to explore our design ideas and communicate them in order to obtain feedback from potential users. This phase has been important for the specification of the design and the refinement of functional requirements, which will inform the implementation of the appliance in the instrumented room of our lab.

## 6.1 Specifying Requirements for the Next Design Iteration

So far, we adopted a qualitative approach to evaluate the functionality and the user interface of the appliance in informal trials. We conducted 7 in depth interviews with individuals who cover different roles within creative teamwork: two managers of creative agencies, two creative directors, a free lancer working on conceptual design for adverting agencies, a graphic designer, and a professor of art pedagogy. The size of the agencies they worked for was heterogeneous (up to 20 people), as well as the focus of the agencies (print communication as well as web).

Every interview lasted about 1.5 hours and followed a standard format: we first asked the interviewees some questions concerning the specific workflow of their creative teams, e.g., how many people usually work on a project, their roles and professions, how activities of picture selection take place, what activities are worked out in co-location, etc.. We then introduced them to our vision of ubiquitous displays and novel collaborative scenarios: in this context we showed them a video which illustrates how the EnLighTable works. We then lead them to the instrumented room of our lab and showed them the interactive table. In this phase they were first given a short demonstration of how to use the different functionalities and then they were invited to play around with it. We then engaged them in a discussion about the interface, e.g. what they found intuitive or less intuitive in the prototype; and on the potential use and impact of such an appliance in their creative teamwork.

Even though the number of tests we had so far is too limited to make generalizations, the interviews gave us some useful insights for further requirements. Furthermore they provided a better understanding of how the target medium (i.e., paper or digital communication) affects picture selection as well as organizational and collaborative aspects. The possibility to simultaneously view multiple pictures on a table, to easily drag them within the focus of attention and visual angle of team members, and to discuss them together with colleagues was highly appreciated by each interviewee. The interaction technique, which allows natural manipulation without a mouse, was also positively assessed. Three interviewees suggested that such an interaction could make the process of picture selection more accessible to everyone, potentially to customers as well, thus having customers' feedback earlier in the design process. The virtual tool was used correctly by every user and its look and feel was appreciated. Six users suggested that it would be useful to have the possibility to scribble and make annotations on the table. Two users also noted that it would be good to have standard formats (e.g., DIN A4 or letter) visualized on the grid of the digital table.

According to the specialization of the agency, further observations were made. For print agencies, the resolution of the display and the possibility to view pictures in 1:1 scale is very important. Web communication seems to require less team effort in picture selection: in this field ad-hoc shooting is quite rare, and digital libraries are mostly consulted by a single graphic designer. Furthermore, web agencies seem to have a more horizontal structure. In this case the graphic designer is often directly working with the customer, thus having less communication flow within the agency.

### 6.2 The Table in the Environment

In this paper we focused on the design of an interface for multi-user interaction on a table-top display. The peculiarities and the relatively novel features of table-top displays for digital information motivated our specific consideration of the device in isolation from other possible displays and devices in the environment. The progressive cost reduction of projectors and of large, high-resolution displays makes interactive workspaces more and more plausible in real working environments of the near future. Different collaborative tasks will need to be supported by the migration of information and activities across different displays. This will require suitable manipulation techniques and relative affordances. We plan to extend our appliance to a vertical large display in the instrumented environment of our lab (see fig. 4 (b)). Users will be able to display the designed layout on the vertical large display, so as to initiate discussion in the team. The absence of orientation conflicts and the wider area within the users' visual angle make vertical displays more suitable for the presentation activity. In our design we are addressing the ecology of displays in the environment, to meet the different needs of different phases of a collaborative workflow, and to enable a consistent interaction in such a display continuum. In this sense we do not expect that individual work on personal displays, supported by the respective applications for photo editing, will be replaced: rather, it will be integrated into novel collaborative scenarios.

**Acknowledgments.** This research has been funded by Deutsche Forschungsgemeinschaft (DFG) within the FLUIDUM (Flexible User Interfaces for Distributed Ubiquitous Machinery) project. Additional Equipment was provided by the Bavarian State. We also would like to thank the people who kindly participated in our user study.

## References

1. Apted, T, J Kay and A Quigley, Tabletop Sharing of Digital Photographs for the Elderly. In Proceedings of CHI2006, the Conference in Human Factors in Computing Systems, April 24 - 27, 2006, Montréal, Québec, Canada. (to appear).
2. Bier, E., Stone, M., Pier, K., Buxton, W., and DeRose, T.: Toolglasses and Magic Lenses: the Seetrough Interfaces. Proceedings of Computer Graphics, August 1993, pp73-80.
3. Butler, C. G., and St. Amant, R. (2004). HabilisDraw DT: A Bimanual Tool-Based Direct Manipulation Drawing Environment. *ACM Conference on Human Factors in Computing Systems (CHI), short papers*. Pp. 1301-1304.

4. Buxton, W. and Myers, B. A.: A Study in Two-Handed Input, Proc. ACM Conf on Human Factors in Computing Systems (CHI '86), 321-326.
5. Cohen, S.: Kai's Power Tools 3 for Windows Visual Quickstart Guide**.** Peachpit Press, 1997.
6. Dourish, P. and Bellotti, V.: Awareness and Coordination in Shared Workspaces. Proceedings of theConference on Computer Supported Cooperative Work CSCW ´92, Toronto, Canada, ACM Press, New York, pp. 107-114.
7. Dourish, P. and Bly, S.: Portholes: Supporting awareness in a distributed work group. In Proc. of CHI92 : Human Factors in Computing Systems. ACM: New York. p. 541-547.
8. Dourish, P.: Where the Action is. The Foundations of Embodied Interaction. Bradfor Books, (2004).
9. Fischer, G.: Social Creativity, Symmetry of Ignorance and Meta-Design. In L. Candy and E. Edmonds (eds), Proceedings of the Conference "Creativity & Cognition 1999", ACM Press, pp 116-123.
10. Gaver, W.: Affordances for Interaction: the Social is Material for Design. Ecological Psychology 8 (2),
11. Grint, K., and Woolgar, S.: The Machine at Work. Polity, Cambridge, UK, 1997.
12. Guiard, Y.: Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. Journal Motor Behavior, 19 (4), 1987, 486-517.
13. Hartson, R., H.: Cognitive, physical, sensory, and functional affordances in interaction design. In Behavior & Information Technology, September-October 2003, vol. 22, n. 5, 315-338.
14. Hinrichs, U., Carpendale, S., Scott, S. and Pattison E.: Interface Currents: Supporting Fluent Collaboration on Tabletop Displays. In Proceedings of the 5th Symposium on Smart Graphics, August 22-24, 2005. Frauenwörth Cloister, Germany, pp. 185-197.
15. Kirsch , D: The Intelligent Use of Space. Journal of Artificial Intelligence, 73(1-2), 31-68, 1995
16. Kreijns, K. and Kirschner, P. A.: The social affordances of computer-supported collaborative learning environments. In Proceedings of the 31th ASEE/IEEE Frontiers in Education Conference, 12—17, (2001).
17. Krüger, W., Bohn, C.A., Fröhlich, B., Schüth, H., Strauss, W., Wesche, G.: The Responsive Workbench: A Virtual Work Environment. IEEE Computer 28(7): 42-48 (1995).
18. Rekimoto J. and Saitoh, M.: Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments. Proceedings of CHI'99, 1999.
19. Scott, S.D., Carpendale, S., & Habelski, S.: Storage Bins: Mobile Storage for Collaborative Tabletop Displays. IEEE Computer Graphics & Applications: Special Issue on Large Displays, 25(4), July/August 2005, pp. 58-65.
20. Scott, S.D., Carpendale, M.S.T, & Inkpen, K.M.: Territoriality in Collaborative Tabletop Workspaces. In Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW)'04, November 6-10, 2004, Chicago, IL, USA.
21. St. Amant, R., and Horton, T. E. (2002).: Characterizing tool use in an interactive drawing environment. Second International Symposium on Smart Graphics. Pp. 86-93.
22. Streitz, N. A., Geißler, J. Holmer, T., Konomi, S., Müller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P., Steinmetz R.: i-LAND: An interactive Landscape for Creativity and Innovation. In: ACM Conference on Human Factors in Computing Systems (CHI'99), Pittsburgh, Pennsylvania, USA, May 15-20, 1999. pp. 120-127.

23. Sundholm, H., Artman, H., Ramberg, R.: Backdoor Creativity – Technological support and Collaborative Creativity. In F.Darses, Zarate, P., Zackland, M. (Eds.). Proceedings of Conference on the Design of Cooperative Systems (COOP´04). Heyeres, France, May, 2004.
24. Terrenghi, L.: Design of Affordances for Direct Manipulation of Digital Information. In Proc. Smart Graphics 2005, Munich, Germany.
25. Ullmer, B. and Ishii, H., The metaDESK: Models and Prototypes for Tangible User Interfaces, in Proceedings of Symposium on User Interface Software and Technology UIST '97), (Banff, Alberta, Canada, October, 1997), ACM Press, pp. 223-232.
26. Weiser, M.: The computer for the 21st century. Scientific American, Vol. 265, September 1991.
27. Wellner. P.: Interacting with paper on the DigitalDesk. Communications of the ACM, 36(7):86--96, July 1993.

# ArTVox: Evolutionary Composition in Visual and Sound Domains


Artemis Moroni[1], Rafael Maiolla[1], Jonatas Manzolli[2], and Fernando Von Zuben[3]

[1] Renato Archer Research Center , Robotics and Computer Vision Division,
13069-901 Rod. D. Pedro I, km 143,6, Campinas - SP - Brazil
Artemis.Moroni@cenpra.gov.br
http://www.cenpra.gov.br

[2] State University of Campinas, Interdisciplinary Nucleous for Sound Studies,
13091-970 CP 6166, Campinas - SP - Brazil
Jonatas Manzolli@nics.unicamp.br
http://www.nics.unicamp.br

[3] State University of Campinas, School of Electrical and Computer Engineering,
13069-970 CP 6101, 13083-970, Campinas - SP - Brazil
vonzuben@dca.fee.unicamp.br
http://www.dca.fee.unicamp.br/~vonzuben/



**Abstract.** Computational creativity is certainly interesting and potentially important. ArTVox, a Java programmed evolutionary environment, arose from the attempt to emulate computational creativity applied to artistic production, in visual and sound domains, by using interactive genetic algorithms. Objects inspired in Kandinsky's artworks are being programmed in Shape, an auxiliary Java environment, to be inserted in ArTVox. Today, ArTVox creates and evolves visual compositions of geometric primitives that, by their turn, guide the sound production in another evolutionary environment, JaVox, integrated to ArTVox.


## 1 Introduction

Since the advent of computers, somehow it has been attempted to automate human processes such as intelligence and creativity. Artificial intelligence is the field of research concerned with making machines do things that people consider requiring intelligence. According to Minsky [1], there is no clear boundary between psychology and artificial intelligence because the brain itself is a kind of machine, but many critiques have been addressed to the automation of such processes [2]. Despite of this, surprising results were obtained from systems that, if not creative, somehow promote creativity.

But what is creativity? Artists and scientists rarely know how their original ideas came about. They mention intuition, but cannot say how it works. How could science possibly explain fundamental novelties? Sometimes, creativity is explained as the combination of familiar ideas in unfamiliar ways. In other cases, it involves the exploration, and sometimes the transformation, of conceptual spaces in people´s minds [3, 4]. Others authors affirm that an idea or product that deserves the label "creative" arises from the synergy of many sources and not from the mind of a single person.


A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 218 – 223, 2006.

A recent generation of computational creativeness researchers is discovering that by using simulated evolution techniques it is relatively easy to obtain novelty, often complex novelty. The success of evolutionary design systems has resulted in some researchers speculating that they model creativity [5], although most commentators are justly cautious and do not make such claims without some reservations [6].

In what follows, evolutionary computation is introduced as a possible paradigm for emulating some kind of computational creativity. Section 3 presents some of the objects that are being developed from Kandinsky´s works and an interface feature to translate visual into aural characteristics.

## 2 Evolution as a Computational Creativity Paradigm

The genetic algorithm (GA) is perhaps the most well known and popularized of all evolution based search algorithms, although it is fair to say that this is partially a result of the term "genetic algorithm" being often used in the way we use the term "evolutionary algorithm". GAs were developed by John Holland [6, 7] in an attempt to explain the adaptive process of natural systems and to design artificial systems based upon these natural systems. Having become widely used for a broad range of optimization problems, the GA has been described as a "search algorithm with some of the innovative flair of human search".

Genetic algorithms are very forgiving algorithms – even if they are badly implemented, or poorly applied, they will often still produce acceptable results [8]. The simplest form of GA, the canonical or simple GA, is summarized as follows. Step 3 refers to a new cycle. Commonly, while-condition (step 3) occurs after several cycles, or an "era". Steps a. to e. refer to the generation of a new population of individuals, which will be evolved.

1. Initialize population with random "chromosomes"
2. Evaluate all individuals to determine their initial fitnesses
3. While there is not an acceptable solution (use the fitness to verify) do
   /* Reproduction
   a. Randomly take two parents from "mating pool" (higher fitness = more copies of an individual);
   b. Use random crossover to generate two offsprings;
   c. Randomly mutate offsprings;
   d. Place offsprings into population;
   e. Evaluate all individuals to determine their fitnesses.

   Until the population is fulfilled with new individuals.

Genetic algorithms (GA) are potentially relevant to art as well as to science -- especially if the evaluation is done interactively, not automatically. That is, at each generation the selection of items from which to breed the next generation is done by a human being. This methodology is well-suited to art, where the evaluative criteria are not only controversial but also imprecise -- or even unknown. Frequently, this is solved by using human-machine interaction, or interactive genetic algorithms (IGA). The human eye has an active role to play in this approach, it is the selecting agent. It surveys the litter of progeny and chooses one for breeding [9, 10]. The chosen one then becomes the parent of the next generation, and a litter of its mutant children are displayed simultaneously on the screen.

In the computer model, the selection criterion is not survival, but the ability to appeal to human whim. The human being tells the computer which of the current litter of progeny is to breed on. The genes of the chosen one are passed across to *reproduction*, and a new generation begins. This process, like real-life evolution, goes on indefinitely. In this work, the interactive genetic algorithms are being applied for composing in the visual and sound domains, described next.

## 3 Computational Composition in Visual and Sound Domains

Here, an amusing "computational philosophical" question arises: if a machine could create, what would it create? What would be its "problematic artmaking"? Somehow it seems that the basic vocabulary of forms of a computational system would be elementary geometric forms. Certainly, this was a completely *arbitrary* decision, but once taken to exercise the computational creativity in the visual domain, the second decision was to choose Kandinsky-like objects to try. This second decision was not arbitrary, it influenced the fact that, besides being an abstract artist, Kandinsky tried to relate the visual and sound domains.

Kandinsky´s composition VIII 1923 was chosen to start our development [11]. Departing from it, visual objects were created in an auxiliary environment, Shape, in Java. In figure 1, some objects are presented and in figure 2, Kandinsky-like compositions are depicted.

The methods for the objects in Shape environment are being created to be inserted in ArTVox environment. In its turn, ArTVox, or ArTbitrating JaVox, has the facilities for the automatic generation of compositions of geometric curves that can be translated to sound trajectories [12]. In ArTVox, at each iteration, a composition is generated and presented to the user for evaluation. The user can attribute to each one a grade - the *fitness* - from 0 to 10, the default is 0. After the evaluation, the user can evolve *the population of compositions*. Only those compositions with *fitness > 0* are considered for evolution, and eight new compositions are created from the old ones.

ArTVox aims to join two domains: the visual and sound domains. A well succeeded work, VOX POPULI, an evolutionary sound environment, previously developed in Visual Basic, was translated to Java, resulting in JaVox environment. Now, facilities for creation in visual domain are being incorporated in ArTVox, to be transformed in sound attributes.

Since VOX POPULI, in the sound domain, at the ArTVox, interface controls use nonlinear iterative mappings that can give rise to attractors. These attractors are defined as geometric figures that represent a set of stationary states of a dynamic system, or simply, trajectories to which the system is attracted. A piece of music consists of several sets of musical raw material manipulated and exposed to the listener, such as pitches, harmonies, rhythms, timbres, etc.

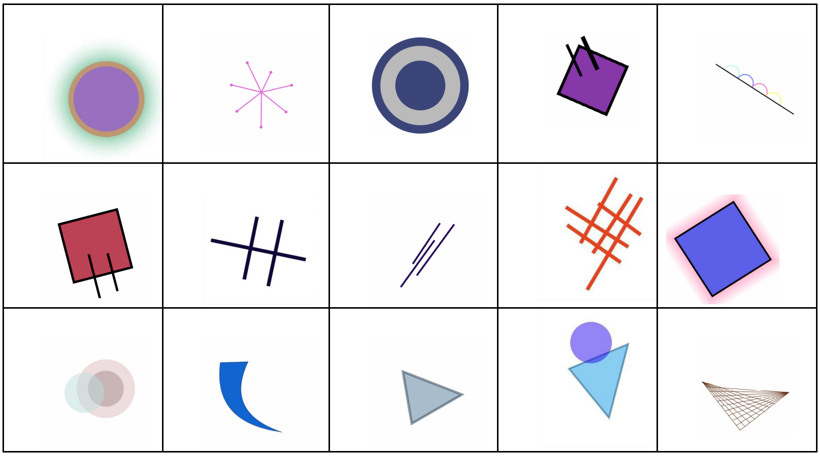


**Fig. 1.** The objects shown above were created from the objects in Kandinsky´s composition VIII 1923

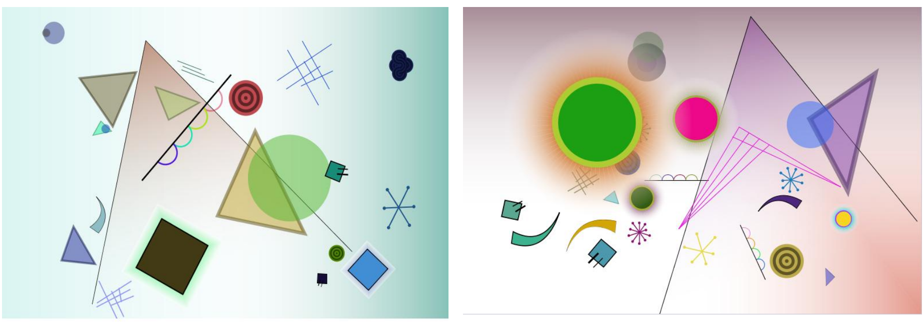


**Fig. 2.** “Kandinsky-like” compositions generated in Shape environment

Modeling a piece of music as a dynamic system implies in a view in which the composer draws trajectories or orbits using the elements of each set. In VOX POPULI, an interactive pad control supplies a graphical area in which two-dimensional (2D) curves can be drawn [13, 14]. These curves may be linked to the controls of the interface.

A difficult problem arises: how to map attributes from visual domain to the sound domain? Here, we recall Kandinsky [15, 16]. In his famous books, he establishes parallels between color, form and music. According to him, “It is evident that many colors are hampered and even nullified in effect by many forms. On the whole, keen colors are well suited by sharp forms (e.g., a yellow triangle), and soft, deep colors by round forms (e. g., a blue circle). But it must be remembered that an unsuitable combination of form and color is not necessary discordant, but may, with manipulation, show the way to fresh possibilities of harmony. … In music a light blue is like a flute, a darker blue a cello; a still darker a thunderous double bass; and the darkest blue of

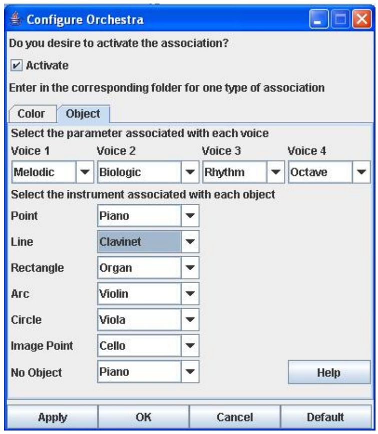



**Fig. 3.** This window interface allows the user to associate visual objects with instruments, according to Kandinsky criterion. Another option allows the association of colors with instruments.

all – an organ." And so on: cool red are the sad, middle tones of a cello; warm red or orange is an old violin; violet an English horn, or the deep notes of wood instrument.

Based on Kandinsky's color and visual proposed relationships, a mapping is being applied from color to instrument in ArTVox, but features of the interface are also opened to the user. After all, in accordance with Kandinsky himself, any parallel between color and music can only be relative. Just as a violin can give various shades of tone, so yellow has shades, which can be expressed by various instruments. Figure 3 depicts one of the ArTVox windows opened to user association.

## 4 Conclusion

Programs using evolutionary algorithms can evolve unexpected structures, if necessary over many thousands of generations, in a form which the human mind could not have produced by itself. Here, we apply evolutionary computation to evolve visual and sound compositions. Kandinsky-like objects are being created in an auxiliary environment to be incorporated to ArTVox. ArTVox, in its turn, evolves visual compositions of geometric primitives that are transformed in sound attributes.

The difficult problem is how to map attributes from visual domain to the sound domains. It seems that, if in the past a lot of technological problems arose for building engines that worked in visual and sound domains, today the problem is more conceptual than technological; a new esthetical era is initiating. Kandinsky suggested mappings, from visual to sound domains, were applied at ArTVox, but features of the interface are also opened to the user/composer who decides the mapping.

ArTVox and Shape environments are available for test at http://www.geocities.com/artbitrating.

## Acknowledgements

We would like to thank Daniel Gurian Domingues and Leonardo Laface de Almeida, who worked in the development of JaVox and ArTVox environments. We would like to thank PIBIC/CNPq program and CenPRA, for making this research possible. This research work is part of the AURAL project, supported by FAPESP process 05/56186-9.

## References

1. Minsky, M. The Society of Mind. Touchstone (1985)
2. Searle, John. R. Minds, brains, and programs. Behavioral and Brain Sciences 3, 3, (1980) 417–457
3. Boden, M. Dimensions of Creativity. New York: The MIT Press (1996)
4. Boden, M. "Creativity and artificial intelligence", Elsevier Science: Artificial Intelligence 103 (1998) 347–356
5. Searle, John. R. Minds, brains, and programs. Behavioral and Brain Sciences 3, 3, (1980) 417–457
6. Goldberg, D. E. "The Race, the Hurdle, and the Sweet Spot: Lessons from Genetic Algorithms for the Automation of Design Innovation and Creativity". In Bentley, P. (ed.), Evolutionary Design by Computers. Morgan Kaufmann, 1999
7. Holland, J. H. Genetic Algorithms and the Optimal Allocations of Trials. Siam Journal of Computing, 2, 2, 88–105. 1973
8. Holland, J. H. Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975
9. Dawkins, R. The Blind Watchmaker. London: Penguin Books. 1991
10. Sims, K. Interactive Evolution of Equations for Procedural Models, The Visual Computer, 9, 1993, 466–476.
11. Kandinsky, W. Composition VIII 1923 (140 Kb): Oil on canvas. Solomon R. Guggenheim Museum, New York http://www.ibiblio.org/wm/paint/auth/kandinsky/
12. Moroni, A., Von Zuben, F.J., Manzolli, J.: ArTbitration: Human-Machine Interaction in Artistic Domains", Leonardo 35, 2, 2002, 185 – 188.
13. Moroni, A., Manzolli, J., Von Zuben, F.J. and Gudwin, R. 2000. Vox Populi: An Interactive Evolutionary System for Algorithmic Music Composition. Leonardo Music Journal, 10, 2000, 49–54
14. Moroni, A., Manzolli, J., Von Zuben, F.J. and Gudwin, R. Vox Populi: Evolutionary Computation for Music Evolution. In Bentley, P. (ed.) Creative Evolutionary Systems, San Francisco, Morgan Kaufmann, 2002, 205–221
15. Kandinsky, W. Concerning the Spiritual in Art. Dover Publications. 1977
16. Kandinsky, W. Point and Line to Plane. Dover Publications. 1977

# An Account of Image Perceptual Understanding Based on Epistemic Attention and Reference

Nicolas J. Bullot

University of British Columbia, Department of philosophy, 1866 Main Mall E370,
Vancouver BC, V6T 1Z1, Canada
nbullot@interchange.ubc.ca

**Abstract.** Technological and scientific images, and other images with epistemic uses, have varied appearances and functions. They seem to be analog or symbolic representations available to researchers for a variety of epistemic purposes such as summarizing data, or presenting, discussing and verifying hypothetic propositions about the world. This article studies the perception and understanding of scientific/epistemic images within a conceptual framework grounded in the notion of reference. It introduces the hypothesis stating that the performance of the perceptual understanding of a particular scientific image depends on the epistemic uses of attention. The hypothesis suggests that understanding a scientific picture requires making an epistemic use of the attentional control of visual routines in order to obtain knowledge on the spatial structure and the referents of a particular image or graphic representation.

Technological and scientific practices rely routinely on the perception of technical tools such as optical systems and measure instruments. By means of these tools, and sometimes without them, theoreticians record data which are usually presented and communicated via a variety of images and graphics such as drawings, photographs, diagrams, mathematical graphs, schemas, IRMf scans, 3D models etc. One can call this variety of non-linguistic but epistemic representations ‘scientific images’ or ‘pictures for epistemic uses.’ Scientific images are analog or symbolic representations available to researchers for a variety of epistemic purposes such as summarizing data, or presenting, discussing and verifying hypothetic propositions about the world. An important question for the epistemology of scientific images is the following:

How are we to describe and explain the semantic and psychological conditions of the epistemic uses of images – as opposed to their socio-political or aesthetic uses?

This problem can lead to different types of investigations. A first set of problems bears on the structure or architecture of the *system* which endow the image with its meaning (e.g., a ‘pictorial system’ in Lopes’ sense), and determine the competence required for its grasping: How does an image obtain its scientific use(s) and its meaning within the scientific community? What are the cues (i.e. the information-bearers features that can be analyzed by visual attention) of the image which are selected by such and such *type* of image? A second set of problems encompasses the questions about *performance* of image producing and understanding. How does an agent understand the meaning of a scientific picture (when he/she is producing it or examining it) in the real time examination or production of the image? How do we use our sensory

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 224 – 229, 2006.

and motor apparatuses for producing or understanding a scientific or a technical picture? Studies about the first group of questions have been introduced namely by Nelson Goodman [1], Dominic Lopes [2] and Keith Stenning [3]. It has lead to the formulation of conceptual debates about the structure of a pictorial system. Although fundamental for the semantic of pictures, these analyses do not answer directly to the second group of questions which bears on the actual cognitive *performance* of image understanding (instead of the semantic system which is required for understanding pictorial systems). The analysis below intends to show that resolving the second kind of question requires a theory of epistemic attention within a referentialist framework for the semantics of image perception. I will first sketch the referentialist framework and then propose a role for epistemic attention within this framework.

This article will suggest a hypothesis which is dependent on a referentialist framework for the cognitive semantics of image uses. Such framework is at least partly consistent with referentialist theories of thought and language [4-6] ; its conceptual foundations have been already considered at least by D. Lopes [2]. When I call this framework *referentialist*, I intend to convey the idea that image understanding is based on the knowledge of the referents that parts of images' surface inform about or denote. In the most straightforward case, a *referent* of a particular image part *e* is any object – or part of an object – that is represented or denoted by *e*. Scientific images refer to (or represent, depict) many kinds of referents such as individual objects, spatial structures, abstract types or variations among magnitudes. The article will not try to expound and classify this extreme diversity. Instead, the argument will hint at the common cognitive ability required to obtain knowledge about image referents and focus on the example of photographic images.

According to my main suggestion, the cognitive foundations of scientific image understanding rest on the thinker's use of visual attention. The basic reason in support of this account is that understanding the referential and epistemic statuses of elements of an image depends on the performance of attentional procedures. Such a hypothesis focuses on the analysis of the selective procedures by which an intentional agent actively obtains knowledge about the element(s) of an image's surface and their referent(s). A general formulation for it is as follows:

> *H, Image Understanding through Epistemic Attention*: Performance of the perceptual understanding of a scientific image depends on the epistemic uses of perceptual attention, conceived of as the system which controls perceptual and motor routines to resolve pragmatic and epistemic queries:
> ($H_1$) in order to evaluate perceptual predicates[1] (or observational propositions) about the *elements* presented by the image's surface and to diagnose the presence of recognizable or analyzable contents; and
> ($H_2$) in order, ultimately, to access information about the *referents* of the image via the singular knowledge of elements displayed by the image's surface.

Hypothesis *H* bears on the mental faculties required to understand the objective and referential characteristics of an image and interpret it as a function of a relevant pictorial or conceptual system.

[1] On the notion of 'perceptual predicate,' cf. e.g., Miller & Johnson-Laird [7], Ullman [8], Pylyshyn [9].

To clarify the meaning of *H* requires specification of the notion of attention which is relevant here. The faculty of attention is traditionally studied in psychology in which it has been frequently conceived as the faculty of selecting information for further processing. For instance, visual attention [10, 11] is basically considered as the faculty which allows selecting and processing the information available within the visual field. There are however arguments to avoid conceiving of the attentional system simply as a spatial or temporal filtering system [12, 13]. Instead of apprehending 'selective attention' as a mere filter, my use of the phrase 'attention' or 'attentional system' refer to a system of control of sensory-motor routines or skills. This analysis belongs to the framework of what one can call a *procedural theory* [2] of attention. The notion of a 'procedural theory' refers here to the accounts that view attentional capacities as being coincident with the exercise of epistemic and pragmatic procedures that are dependent on a context of use and of strategies for reaching particular goals. According to this type of analysis, attention uses strategic and exploratory operations that enable the agent to obtain information (typically) on a particular target element or cues related to one object or spatio-temporal element (and often also to constitute a singular representation of this target). One can give an account of this strategic structure by analyzing selective attention as being dependant on two main components: (*i*) a set of instructions for the control of bodily or mental events, which can be termed either *epistemic* queries or *pragmatic* queries – and (*ii*) a set of elementary operations called *routines* that allow, according to varied and context-dependent combinations, to give an answer to or to satisfy the epistemic and pragmatic queries. The concept of *routine* refers to the perceptual or motor elementary procedures that can be used to satisfy or solve the queries (epistemic or pragmatic) on the basis of the evaluation of perceptual predicates. Attention, as a mediating faculty, seems to be the capacity that organizes the relations between conceptual and non-conceptual routines for demonstrative identification.

A general argument that supports *H* is that attention – as a faculty of controlling perceptual routines – is constitutive of any kind of *epistemic* perception, since the control of perceptual routines is a necessary condition of an epistemic access to the target properties. Although this claim is I think correct, it is not informative about the *specific* procedures required to understand a scientific image. It is possible to formulate more specific arguments – based on the type of cues which are likely to be relevant for understanding the image in its particular context of use. I will sketch two of them: the argument from compositionality and the argument from singular reference.

A first argument is about the 'navigation' in the compositional structure of the image. It states that only the attentional system allows perceivers to retrieve the compositional structure of the image, that is, the spatial relations among the elements displayed by the image's surface. The support for this idea can be derived from the need to appeal to visual routines to explain the visual understanding of basic spatial relations, such as in Ullman's [8] analysis. Let us assume a distinction between the 'automatic' (or 'stimulus-driven') formation of 'early visual representation' and subsequent application of visual routines. The argument is that a number of operations

[2] Among the class of procedural theories, I shall include namely works on perceptual predicates [7], visual routines [8, 14], deictic strategies [15], epistemic uses of eye movements [16, 17], and visual reference [5, 9, 18].

performed during the examination of an image require performing visual operations which, arguably, cannot be accounted for neither by so-called ‘automatic’ or ‘ stimulus-driven’ processing nor by ‘purely conceptual’ abilities based on type/kind identification. One can argue that the control of visual routines is required namely for operations such as *retrieving basic elements or shapes in a display* and *specifying the spatial relations among these basic elements/shapes* (top/bottom, right/left, inside/outside etc). Ullman gives, for instance, the example of determining whether a point lies inside or outside a closed curve, and shows that a base representation of the point and the curve is not sufficient to resolve this query about spatial organization.

Another argument does not refer to combinatorial arrangement of the elements in the image display but bears on singular reference. If the perceiver has some mastery of the compositional characteristics of an image, this means that she has been able to pick up elements and examine their spatial relations. Given that a set of elements has been segmented, a *singular knowledge* of the properties of each element becomes possible, and this singular knowledge can serve as a background for the knowledge referential disposition of the image’s parts. Building incrementally knowledge about the referent of an element *e* of an image (whatever this element might be) is prior to reasoning about the properties of the referent of *e*. In the framework of the procedural theory described above, this building incremental knowledge about *e* is dependent on epistemic visual attention because such a framework assumes that that the singular knowledge of a particular element *e* is acquired incrementally by using sensory-motor routines – or visual object files – to evaluate perceptual predicates.

The hypothesis *H* can be applied to the example of the epistemic uses of photographs. In the case of photographic images, the referentialist account can point to the fact that each photograph carries information about the *photographic referents* (the objects that have been photographed). Such characteristic depends on a mechanical system for image production – cf. the bottom triangle of Figure 1. Consider the case in which a part *e* of a reliable photographic image has the function to *indicate* a referent *r* which has been photographed (*r* can be an individual or a group of individual objects). The referential disposition of the photographic image is determined by a reliable causal process of image production. The camera has recorded, via a system using chemical or digital transduction, an optical projection of the referent. As a result, a number of properties of element *e* within the image (spatial, chromatic, textural properties) are counterfactually dependent on the properties of the referent *r* in a way that can be observed in a final photographic image. But how is this information exploited or extracted by a perceiver in an *epistemic* and objective manner? Here follows a plausible scenario based on hypothesis *H*.

The understanding of an epistemic use of a photograph (for example within a scientific discourse or practice) requires perceptual and conceptual abilities tied by epistemic attention – this is illustrated in the top triangle in Figure 1. Entertaining demonstrative thoughts about elements $e_i$ of the photograph’s surface – such as ‘These dots are *F*’ or ‘This mark is an *F*’ – presupposes the perceptual *segmentation* of these relevant elements. For it is only once a primal segmentation of the elements is secured that the perceiver can formulate demonstrative identifications such as ‘That element $e_1$ is an image of a shadow’ etc. Such demonstrative identifications of the elements allow the perceiver to form beliefs about relevant elements $e_i$ and their relations. For instance, on the basis of the perceptual identification of $e_1$, the perceiver can evaluate perceptual predicates about $e_1$ such as *White*($e_1$) or *Touching*($e_1$, $e_2$) or *Connected*($e_1$, $e_4$). In addition, the perceiver can form singular beliefs about $e_1$ and its relations with other elements such as $e_2$ or $e_3$. It can only be on the basis of the evaluation of perceptual predicates about the identified elements of the image's surface that the perceiver may find a way to conclude that the epistemic status of the element indicate that *its referent* satisfies to the same perceptual predicate.

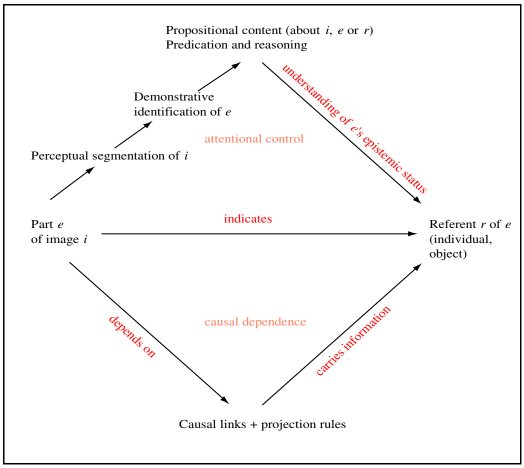



**Fig. 1.** Schema of a reference-based account of the epistemic uses of a photograph

Consider the example given by experimental paradigms in psychology that use video recording and recorded images as data for further analysis. For instance, Land, Mennie, & Rusted [19] recorded simultaneously (i) the activities of subject while performing the task of preparing tea and (ii) their eye movements. The aim of this study was to determine the pattern of fixations during the performance of a well-learned task in a natural setting (making tea), and to classify the types of monitoring action that the eyes perform. The authors used a head-mounted eye-movement video camera, which provided a continuous view of the scene ahead, with a dot indicating foveal direction with an accuracy of about 1 deg. A second video camera recorded the subject's activities from across the room. The videos have been linked and analyzed frame by frame. At least in this task, foveal direction was always close to the object being manipulated, and very few fixations were irrelevant to the task. According to the authors' classification, roughly a third of all fixations on objects could be definitely identified with one of four monitoring functions: (1) locating objects used later in the process, (2) directing the hand or object in the hand to a new location, (3) guiding the approach of one object to another (e.g., kettle and lid), and (4) checking the state of some variable (e.g., water level). If one question how scientists were able to reach such kind of interpretation for the collected data (large number of single video frames), it seems clear that they have had to perform the tasks described by *H*: segmenting, demonstrative identifications of the basic elements, and acquisition of singular knowledge about the referents of these basic element. Ultimately, the classification of the four kind of monitoring functions of eye movements is based on assumptions about the *referents* of the elements of the image and their relations: i.e., the eyes of the agent and the objects which are the targets of her or his actions.

## References


1. Goodman, N., *The Languages of Art*. 1968, Oxford: Oxford University Press.
2. Lopes, D.M.M., *Understanding Pictures*. 1996, Oxford: Oxford University Press.
3. Stenning, K., *Seeing Reason, Image and Language in Learning to Think*. 2002, Oxford: Oxford University Press.
4. Evans, G., *The Varieties of Reference*. 1982, Oxford: Oxford University Press.
5. Campbell, J., *Reference and Consciousness*. Oxford Cognitive Science Series. 2002, Oxford: Clarendon Press.
6. Perry, J., *Reference and Reflexivity*. 2001, Stanford: CSLI Publications.
7. Miller, G.A. and P.N. Johnson-Laird, *Language and Perception*. 1976, Cambridge, MA: Harvard University Press.
8. Ullman, S., *Visual routines.* Cognition, 1984. **18**: p. 97-159.
9. Pylyshyn, Z.W., *Seeing and Visualizing: It's Not What You Think*. 2003, Cambridge, MA: MIT Press.
10. Wright, R.D., ed. *Visual Attention*. 1998, Oxford University Press: New York, Oxford.
11. Pashler, H.E., *The Psychology of Attention*. 1998, Cambridge, MA: MIT Press.
12. Allport, A., *Attention and control: have we been asking the wrong questions. A critical review of the last twenty five years*, in *Attention and Performance XIV*, D.E. Meyer and S. Kornblum, Editors. 1993, MIT Press: Cambridge, MA. p. 183-218.
13. Allport, A., E.A. Styles, and S. Hsieh, *Shifting intentional set: Exploring the dynamic control of tasks*, in *Attention and Performance XV: Conscious and Nonconscious Processing*, C. Ulmità and M. Moscovitch, Editors. 1994, MIT Press: Cambridge, MA. p. 421-452.
14. Ullman, S., *High Level Vision*. 1996, Cambridge, MA: MIT Press.
15. Ballard, D.H., et al., *Deictic codes for the embodiment of cognition.* Behavioral and Brain Sciences, 1997. **20**(4): p. 723-767.
16. Land, M.F. and S. Furneaux, *The knowledge base of the oculomotor system.* Philosophical Transactions: Biological Sciences, 1997. **352**(1358).
17. Land, M.F. and M.M. Hayhoe, *In what ways do eye movements contribute to everyday activities?* Vision Research, 2001. **41**: p. 3559-3565.
18. Kahneman, D., A. Treisman, and B.J. Gibbs, *The reviewing of object files: Object-specific integration of information.* Cognitive Psychology, 1992. **24**(2): p. 175-219.
19. Land, M.F., N. Mennie, and J. Rusted, *The role of vision and eye movements in the control of activities of daily living*. Perception, 1999. **28**: p. 1311-1328.

# Using Rule Based Selection to Support Change in Parametric CAD Models

Davis Marques and Robert Woodbury

Simon Fraser University Surrey,
School of Interactive Arts & Technology,
14th Floor, 13450 102 Avenue,
Surrey, British Columbia, Canada V3T 5X3
{dmarques, rob_woodbury}@sfu.ca
http://www.siat.sfu.ca/

**Abstract.** Parametric CAD applications enable designers to model and maintain both objects and relationships. Selection is the first step in the process of establishing relationships in a CAD model. Because designing is characterized by a process of change and development, establishing and maintaining relationships can contribute significant overhead to a parametric design process. Selection rules can be used to reduce the overhead involved in maintaining relationships and increase the portability of model objects by pushing responsibility for maintaining relations to the computer. We report on two implemented concepts for rule based selection and discuss directions for future research.

## 1 Managing Change in Parametric CAD

Design is a process of making proposals for change and design activity is itself characterized by a process of change and development. Supporting and managing that change is of vital importance in aiding designers to realize better design outcomes. Visual representations support the cognitive processes of designers by enabling them to externalize the design process, see designs in context, explore and map the opportunities available to them. Schmidt and Wagner [1] argue that CAD models are of particular importance in this regard because, “They incorporate, as an ensemble, a project’s trajectory from draft to implementation; they absorb and reflect all decisions taken and changes made, as plans are gradually detailed and modified.”

Current system development is focused on Parametric CAD applications which extend the traditional modeling approach by enabling the designer to parameterize geometric objects and relations, thereby facilitating greater control and expedience in modeling. Aish and Woodbury [2] argue that parameterization is positive in the sense that it can reduce the time and effort needed to make changes, facilitate design exploration, increase the specificity of designs, and aid the designer in clarifying the conceptual structure of what is being designed. Conversely, parameterization can be negative in the sense that it can increase complexity in the modeling task by necessitating a greater level of description and control, compounding the effort needed to simultaneously think through the design.

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 230 – 235, 2006.

Establishing and maintaining relationships between objects is a large part of the parametric modeling and design process. Making it easier for designers to establish and maintain relationships will do much to facilitate the process of designing.

## 2 Explicit Versus Implicit Relational Modeling

Selection is the first step in the process of establishing relationships between objects in a CAD model. Automating selection makes specific relationships implicit in a model.

Typically, a designer will establish a relation by using a mouse or keyboard to select two or more existing objects in the model, then apply a function to generate a relation between them. What results is an *explicit* relationship that binds one particular object or property to another. The system subsequently maintains the relation when the input objects are changed. Explicit relational modeling offers precise control and short execution time. However, changes to relations may entail a significant increase in labour over a traditional modeling approach. To revise explicit relations, the designer must visit each object in question, unbind it from other objects, then modify or establish new relations as required. As the size of the model increases and as the complexity of relations grow, changes to the model require increasing investments of time and effort to execute.

Selections may also be defined by other means such as rules or procedures. In this case, instead of manually selecting the objects to be related, the designer defines the basis by which relations should be made and the computer is then given the responsibility of establishing and maintaining those relationships. When objects in the model satisfy the stated conditions, the computer establishes a relationship. Likewise, when those objects no longer satisfy the relational criteria, the computer removes the relations. We characterize this as an *implicit* relational modeling approach, in the sense that the actually formed relations are an implied consequence of the rules expressed and the current configuration of objects.

While both explicit and implicit approaches result in relationships, important differences occur. In an explicit modeling approach, the selection event occurs only once and therefore the objects in the relation must exist in advance of the selection event. In an implicit modeling approach, objects are not required to exist in advance of the user defined selection rule; selection occurs whenever changes in the model trigger a need to update the selection set, and therefore the relations are always up to date. In this way, an implicit approach might reduce the effort of establishing relationships, particularly in situations where a large number of objects are involved. However, this increased expedience may come at the cost of reduced control and significantly increased execution time.

Existing parametric modeling systems provide limited forms of implicit relationships. For example, a collection of points may be placed on a line with the relationship that the points are to be spaced at some even interval along the line, say, one metre. When the line is extended from, say, four to six metres in length, additional points are instantiated along the line. Note that the number of points in this example is implicit but, the relation between line and point is explicit. Our contention is that the extant facilities for implicit relations are impoverished in comparison to those provided to form explicit relations.

In a scope larger than a simple point on a line, there may also be occasions when combining explicit and implicit selection is desirable. For example, in designing a building we may know that a particular south facing wall will exist but the number of windows on that wall may change through the lifecycle of the design process. If we want to add a sun shade to each window on that wall, we may describe an explicit relation with the wall but an implicit relation to the windows which may be present on that wall.

## 3 Experiments in Implicit Relational Modeling

We report on two implemented prototypes for selection: Range Walker and Rule Based Selection. These prototypes were developed separately in two Generative Components workshops, in November 2004 and January 2006 respectively.

Generative Components is a commercial propagation-based parametric modeling system, produced by Bentley Systems, Inc. [2] It provides several levels of modeling, from specification through the GUI, direct expression of functions in object slots, a scripting language for specifying propagation algorithms, compilation of models into single nodes and C# programming of new object types.

### 3.1 Range Walker

The Range Walker demonstrates a dynamic selection relationship between a point object and a point collection. In Figure 1, `Point01` is a control point; `Point02` is a collection of points; `Range01` is tied to `Point01` and the line carrying `Point02` and delimits a three dimensional space where target points will be selected. As the user drags `Point01` across the screen, `Range01` moves along with it and returns a list of points inside its boundaries. A line is drawn from `Point01` to the points discovered by `Range01`, giving the effect of the control point "walking" along. If the user drags the control point some threshold distance away from `Point02` then the connecting lines cease to be drawn. Likewise, when the control point returns within threshold distance, the connecting lines reappear.

The Walker is implemented using the Generative Components Range Feature object and GScript, GenerativeComponents internal scripting language. The Range object is attached to `Point01` and returns a collection with references to all the objects that are inside the Range volume. The GScript code iterates through the reference collection to find the points in `Point02`, then constructs a line from `Point01` to each of the points. In this demonstration, it is the Range function that provides a dynamic relation between `Point01` and `Point02` by listing only objects that are inside the Range volume.

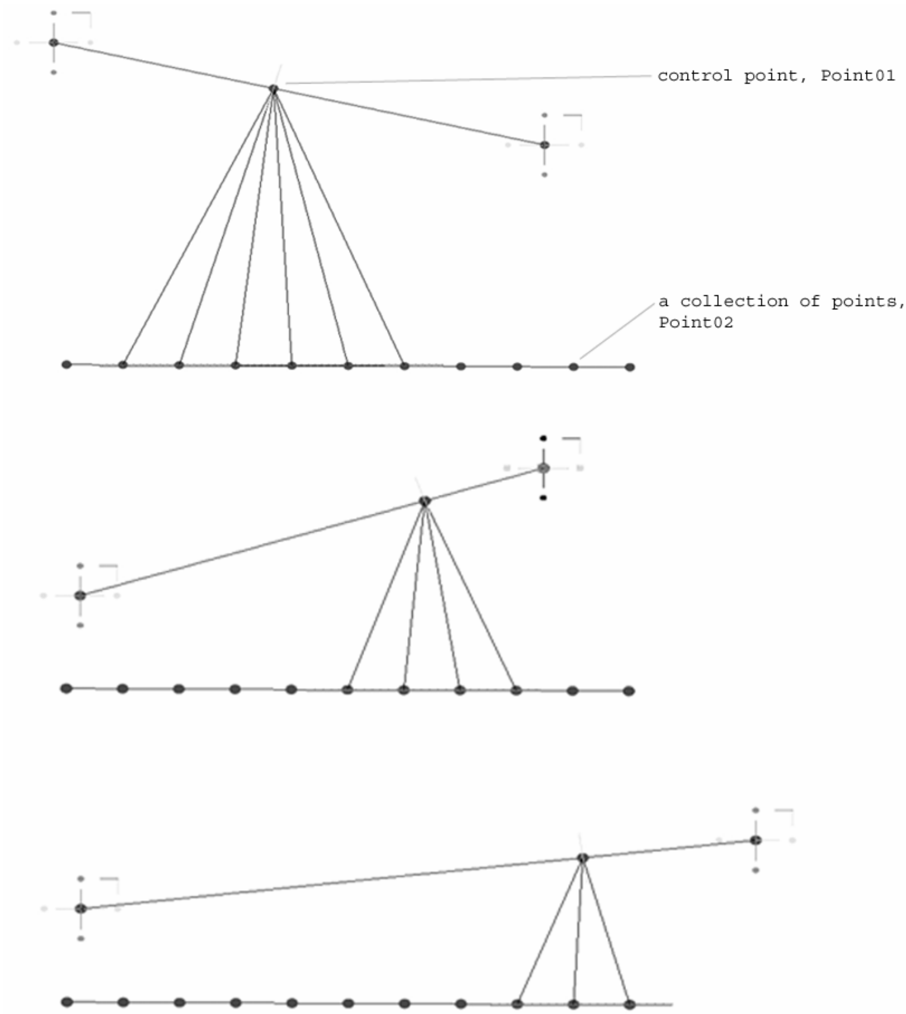



**Fig. 1.** Range Walker. As the user moves Point01 from left to right, the connecting line segments to Point02 are dynamically updated and appear to walk along in unison. Cheryl Qian, Robert Woodbury. (2004)

### 3.2 Rule Base Selection Tool

The Rule Based Selection tool allows designers to make object selections by using rules rather than having to write custom scripts or make selections manually. The benefit of this is that it enables the selections to be updated dynamically, thereby enabling implicit relations between objects and reducing the effort required on the part of the designer to manage those relationships. The Selector is implemented as a Generative Components Feature object using the Generative Components API, and is written in C#.

To employ the Selector, the user creates an instance in the model. The Selector takes one or more selection rule statements and an existing object collection as input, then returns references to a subset of the collection as output. The Selector instance appears in the symbolic graph as a child node of the input object. Five selection rules are available: `Exact Match`, `Range`, `Nearest Neighbor`, `Half Space`, and `Distance`. Rules can match against three basic object properties: spatial coordinates `(x,y,z)`, surface coordinates `(u,v,d)`, and index in a collection `(i,j,k)`. `Exact Match` takes an explicit location or collection index and returns only those objects that exactly match that location or index. `Range Match` takes an explicit location or collection index and returns all objects that fall inside the lower and upper range bounds, inclusively. `Nearest Neighbor` takes an explicit location for a point in space and finds the nearest elements in the input collection to that point. `Half Space` takes a plane, or two vectors that designate a half space, and returns all elements in the space, inclusive of the boundary. Finally, `Distance` takes a point or collection as an input set, and a positive distance value and returns those elements of the input set which are equal to or less than the distance away. The result sets from each rule statement can then be further operated upon using logical operators `AND`, `OR`, and `NOT`. If the designer provides improper selection rules or input objects, or if the result is logically null, the Selector returns an empty set.

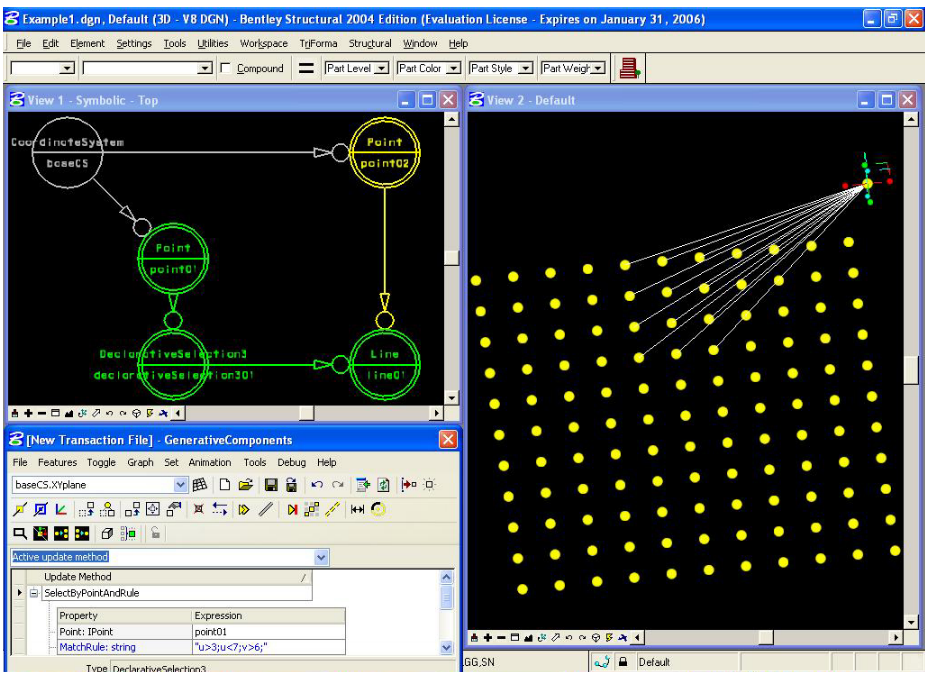



**Fig. 2.** Rule Based Selector tool. The Selector takes a selection rule and a set of objects as input. The resulting selection is here used to generate lines to a control point.

Rules are specified using a functional syntax. For example, `Px(1.0,1.0,1.0)` selects an exact match for an object that is at location `(1.0,1.0,1.0)`. Wildcards may also be entered to create broad matches. For example, `Px(*,1.0,1.0)` selects objects with any `x` coordinate value, and y and z coordinates of `1.0`.

The complexity of the selection operation varies with the selection rule. `Exact Match`, `Range` and `Half Space` have worst case linear time complexity in the size of the model. In the current implementation, `Nearest Neighbor` and `Distance` have worst case time complexity of `O(m*n)` where `m` is the number of elements in the input set, and `n` is the number of elements in the target set.

Future versions of the Selector should order rule execution to reduce the search space in advance. In addition, it is conceivable that numerous instances of the Selector could be in use in a given CAD model at any one time and therefore the search overhead of the tool could become significant. In this case, it may make sense for the CAD environment to support some form of spatial indexing and querying in order to alleviate some of the excessive search overhead of individual Selector instances. Filtering on object types, layers, groupings, other properties commonly found in CAD applications would help to reduce the search space further. The role of the Selector in this circumstance would then be to build properly formed queries, pass them to the environment, then retrieve and cache result sets.

## 4 Conclusions

The two examples of implicit selection reported here are far from a complete set. Just as CAD systems have broad toolkits for such functions as object snaps and object creation, we expect a toolkit for selection to eventually implement a wide range of discrete selectors. We conjecture that good selection operators might be based on inclusion in a range, dynamic selection in composite object properties and on use of generic algorithms such as the convex hull for selection. This is a suggestive, but incomplete catalog. Its extents and details are our current focus.

## References

1. Kjeld Schmidt, Ina Wagner, Ordering Systems: Coordinative Practices and Artifacts in Architectural Design and Planning, Computer Supported Cooperative Work (CSCW), Volume 13, Issue 5 - 6, Dec 2004, Pages 349 – 408.
2. Robert Aish, Robert Woodbury, Multi-level Interaction in Parametric Design, Lecture Notes in Computer Science, Volume 3638, Aug 2005, Pages 151 – 162.

# NEAR: Visualizing Information Relations in a Multimedia Repository

Cheryl Z. Qian, Victor Y. Chen, and Robert F. Woodbury

School of Interactive Arts & Technology, Simon Fraser University,
14th Floor, Central City Tower, 13450 102nd Avenue,
Surrey BC V3T 5X3, Canada
{cherylq, yvchen, rob_woodbury}@sfu.ca

**Abstract.** The NEAR (Navigating Exhibitions, Annotations and Resources) is a compact panel designed to help people navigating, searching and interacting in an information repository by visualizing implicit data relations such as sharing, reference and similarity. It is implemented on A•VI•RE, an online multimedia repository. A•VI•RE supports semi-structured collections (exhibitions) containing various resources and annotations. Its users are encouraged to contribute, share, annotate and interpret resources. Similar to the act of adding items into shopping carts in the e-commence applications, a user's acts of searching and organizing and interpreting data in A•VI•RE are considered as evidence of the user's preferences. The design process of NEAR was guided by several design moves analyzed from literatures. It implements new navigation and communication approaches that support discovery of relations.

## 1 Introduction

Dealing with data relationships and user preferences is becoming a widespread issue in novel data-intensive application domains. Effectively supporting user browsing and search over large repositories entails the problem of properly understanding user needs, filtering out irrelevant items, helping the user to formulate the most appropriate queries, and presenting results ranked according to their presumed relevance. Similar issues arise in multimedia databases and information repositories. A•VI•RE (a Visual Rete, URL: http://www.avire.ca), a generic repository for visual materials, is designed as an interactive online space where users in different roles (such as curators, exhibitors, critics and viewers) work together to create a larger social entity. Users can upload resources, organize exhibitions and annotate resources. The information structure in the system has three primary object types:

- A resource can be an image, a video clip, a file of any type such as PDF or MSWord or even an entire website.
- An exhibition is a collection of resources, annotations and exhibitions.
- An annotation is a mixture of text and reference to exhibitions and resources.
- One resource can be used in multiple exhibitions. Every exhibition or resource may have one or more annotations.
- Metadata exists on all of these object types.

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 236 – 241, 2006.

In the system, one exhibitor may set up an exhibition, add resources and write annotations. Another exhibitor might use the same resources but interpret them from a different perspective. A•VI•RE thus provides a potentially complex information structure among exhibitions, resources and annotations. Users can potentially gain from the collaborative construction of multiple views and interpretations around A•VI•RE's objects. The problem is providing access to the data and relation web that users have made. Our premise is that local links, that is, link paths no more than a few links in length, can provide meaningful access to a community of interpretation.

Similar to many online galleries or information repositories, some navigational problems exist in A•VI•RE system.

- It is hard to provide multiple viewpoints of a resource. A resource can be collected in multiple exhibitions and quoted in multiple annotations, so it naturally bridges different interpretations and connects potentially related ideas. The system should provide a panorama view of a resource from different perspectives.
- It is also hard to see the relations among exhibitions. An exhibition might share resources with other exhibitions or belong to another exhibition. Although reference information among exhibitions is valuable, it could not be explored in the original interface.
- It is easy to find one resource or exhibition through key word searching, but hard to access specific relevant data elements. To search for a specific image among over 3000 images, the user bears a heavy cognitive load and must remember too much information.

## 2 Visualizing and Interacting Local Interconnections

A principal information of A•VI•RE's objects is what users have done with them. Success of collaborative filtering recommender systems such as Amazon is strong evidence that using such information may be effective. In A•VI•RE, users create co-reference and inclusion relations that may be valuable to later users. Co-reference is the phenomenon where two objects (exhibitions or annotations) both refer to or include the same resource. Inclusion refers to the relation between two collections (exhibitions or annotations) that exist when all members (or related objects) of the first are also members (or related objects) of the second. Using these two relations we can construct higher order relations such as similarity and mutual interests in a group. In A•VI•RE, interpretation connections between resources and exhibitions linked by annotations are valuable because they provide later users with access to the carefully constructed thoughts of others. Our design aims to use specific local interconnections as a basis for overview and discovery.

Many applications require a measure of "similarity" between objects. One obvious example is the "find-similar-document" in the World-wide Web [1]. More generally, a similarity measure can be used to cluster objects. Bibliometrics studies the citation patterns of scientific papers. Methods of co-citation [8] and bibliographic coupling [4] have been applied to cluster scientific papers according to topics [6]. SimRank [3] describes an efficient way to measure the structural-context similarity. But the similarity outcomes are demonstrated through many node-link graphs, which are hard to understand at a glance. In A•VI•RE, similarity between exhibitions and annotations can be measured by the Co-reference and inclusion relationships among objects.

NEAR visualizes above A•VI•RE objects relations. Visual depictions of graphs and networks are external representations that aim to exploit human visual processing to reduce the cognitive load of many tasks that require understanding of global or local structure [11]. We outline several important design moves we consider to be essential in the design, along with our current design solution examples.

o **Being compact and demonstrate immediacy.** The physical distance between causally related events should be kept to a minimum for users to recognize. Spatial immediacy, temporal immediacy and semantic immediacy are considered during the design process. Since resolution falls off rapidly from the fovea view of eye (detail is available only at 10 degrees from the fovea) [11], we should avoid spreading the data widely across the screen and should not force users to make large scans to relate data. The visualization should be restricted in a limited space (width at 560 pixels) but easy to be recognized (nodes size at 40x40 pixels) (Figure 1).

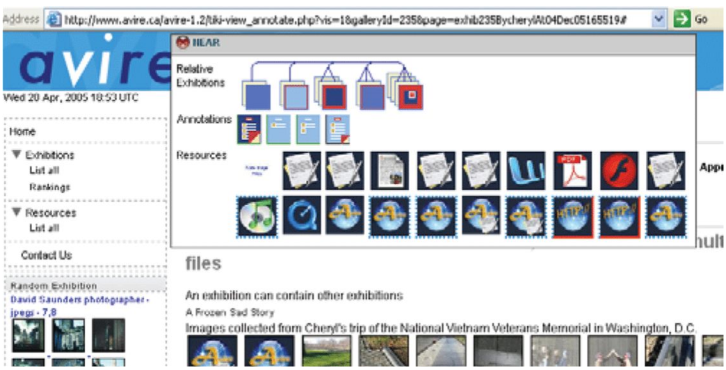



**Fig. 1.** Screenshot of the NEAR panel in A•VI•RE

The panel shows: a list of all resources belong to the current exhibition (bottom two rows), all annotations (the middle row) that have annotated any resources in the list and all related exhibitions (the first row) that share any resources in the list.

o **Show the similarity, co-reference and inclusion.** In NEAR, links are used to show the similarities, references and sharing among exhibitions, annotations and resources. Smooth continuous contours are used to connect nodes and any exhibition sharing the resources are linked. As in Thread Arcs [5], the amount of sharing is represented by the number of branches from one node (Figure 2).

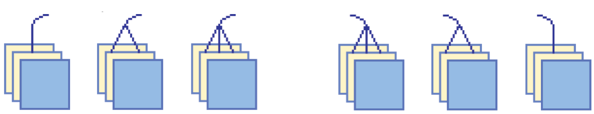


**Fig. 2.** Examples of exhibition nodes using branched-links to share resources

o **Show the attributes in orthogonality.** Since different kind of attribute information is encoded in the graphic nodes, the visual method used for each attribute (size, popularity, etc.) of the data elements should be unique and recognizable. As shown in Figure 3 and 4, for resources other than images, we adopt the design of the file icon from Windows and Mac OS X operation system and create a series of black-background square nodes. Since an exhibition is a collection of exhibitions and resources as organized by users, squares with layers are used as exhibition nodes because this method demonstrates both the different size and hint the organization. Annotations combine text and resources into a linear narrative – in essence, they are papers and their nodes display them as such.

o **Show the popularity.** One should be able to see how frequently a data element has been visited or used by other users. We use blues of different saturations indicate popularity (Figure 3).

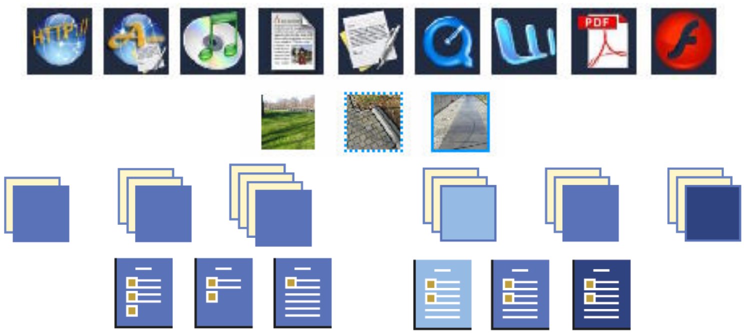


**Fig. 3.** Example nodes of resources, exhibitions and annotations

1st row: Nodes of multimedia resources.
2nd row: Image resource nodes of different popularity (determined by the user visits).
3rd row: Example nodes of exhibitions: Left group: Nodes of different size (determined by number of resources). Right group: Nodes of different popularity (determined by the user visits).
4th row: Example nodes of annotations: Left group: Nodes of different organization (determined by number of resources & amount of text). Right group: Nodes of different popularity (determined by user visits).

o **Show the navigation history.** One should be able to see the hints of his own navigation history such as unvisited and visited items. Context information showing visit status is helpful. We use red borders to hint visiting statues of nodes (Figure. 3).

**Fig. 4.** Examples of nodes with different visit status

Left Group: Exhibition Nodes: unvisited, visited, just visited and current. Middle Group: Annotation Nodes: unvisited, visited and current. Right Group: Resource Nodes: unvisited, visited and current.

o **Show the chronology.** One should be able to see the evolution of an interpret-tation: how understandings and ideas can be built upon or clustered around others' understandings. So nodes of resources, exhibitions and annotations are represented in the generating time sequence (in Figure 1, from left to right: from the newest to the oldest).

o **Being responsive & communicative.** Interaction allows us to drill down and find more data about anything that seems important [11]. Ben Shneiderman also has called a "mantra" to guide visual information-seeking behavior and the interfaces that support it: "Overview first, zoom and filter, then details on demand [9]". "Brushing" [2] is adopted in NEAR interaction to enable visual linking of components of heterogeneous complex objects. When the mouse cursor brushes over a data element, the user can find more relations between data elements and read the description of the element. To avoid flicking, a click on an element freezes the relation view. A click on empty space will re-active the mouse-over effect. Similar to any desktop applications, double clicking will open a separate window to display the full content of the element. Figure 5 shows an instance of how NEAR works.

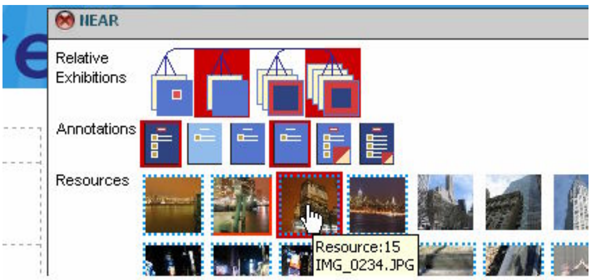



**Fig. 5.** Screenshot of NEAR panel (when the cursor is over a resource)

## 3 Preliminary Evaluation and Problems Identified

We aimed to learn two things from the user study: to optimize the graph visualization design to maximally match user's common sense, and explore the usability of NEAR in providing implicit data relations Five people were invited to test the NEAR panel in different circumstances.

All of the participants gave positive evaluations to the design. Three of them found it helped their understanding of resources and described it as very useful "menu" in term of supporting navigation. However, they also provided critique of the design. To most of the participants, the NEAR panel is more like a tool box instead of a visualization window. They want the panel can be movable, adjustable or even be blended with the background. One participant thinks that the high degree of detail in each icon precludes quick glances across the set to ascertain their meanings. Further, the high degree of detail also requires knowledge on the part of the viewer to ascertain the differences between the icons. Some participants have different opinions on meaning of some nodes. We accept the diversity of interpreting nodes and made some changes on our original designs. There is still much to be improved. The evaluation results show us clearly that the panel is useful and supportive but not self-explanatory.

On the algorithm level, there is a challenge we need to further deal with: the balance between scalability and accuracy [7]: Contemporary systems require searching millions of potential neighbors in real time. A•VI•RE as a web application -- it should have the ability to handle multiple users using the system simultaneously while searching millions of information nodes with high accuracy and consistency. In some ways the two qualities are in conflict: more accurate results normally need more search and calculation that will reduce the speed and scalability. If several exhibitions have hundreds of resources and shared tens of them, how to visualize them with revealing accurate relations in a limited panel, that is a challenge for current design.

## 4 Future Applications

NEAR's design addresses the need for visualizing implied relations such as similarity, co-reference and sharing among different data elements so that a visitor can learn from filtering previous user's interpretation experience. NEAR principles can also be used to improve the design of other academic applications such as the academic citation index. ISI Web of Knowledge [10] provides seamless access to current and retrospective multidisciplinary information from approximately 8,700 of the most prestigious, high impact research journals in the world. Its unique search method: *cited reference searching* helps users to navigate forward, backward, and through the literature, searching from all disciplines. We believe our design of NEAR can help to represent those implied relations among literatures by graph visualization and enhance the navigation efficiency. The idea of NEAR can bring users *near* to implied relations of information and their interpretation *near* to their own preferences.

## References

1. Baeza-Yates, R. Ribeiro-Neto, B.: Modern Information Retrieval, Addison Wesley, Reading, Massachusetts (1999)
2. Becker, R.A., Cleaveland, W.S.: Brushing Scatterplots. In Technometrices 29, 2 (1987). 127-142
3. Jeh, G., Widom. J.: SimRank: A Measure of Structural-context Similarity. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada (2002) 1-11
4. Kessler, M. M.: Bibliographic Coupling between Scientific Papers. In American Documentation, Vol. 12. (1963) 10-25
5. Kerr, B.: Thread Arcs: An Email Thread Visualization. In Proceedings of the 2003 IEEE Symposium on Information Visualization, IEEE Computer Science Press. Seattle, Washington (2003) 211-218
6. Popescul, A., Flake, G., Lawrence, S., Ungar, L.H. Giles, C.L.: Clustering and Identifying Temporal Trends in Document Databases. In Proceedings of the IEEE Advances in Digital Libraries, Washington (2000)
7. Sarwar, B., Karypis, G., Konstan, J. Riedl, J.: Item-Based Collaborative Filtering Recommendation Algorithms. In Proc. of the 10th International World Wide Web Conference (WWW10), Hong Kong (2001)
8. Shneiderman, B.: Designing the User Interface. Addison-Wesley, Reading. MA (1998)
9. Small, H.: Co-citation in the Scientific Literature: A New Measure of the Relationship between Two Documents. In Journal of the American Society of Information Science, Vol. 24 (1973) 265-269
10. Thomson Scientific.: ISI Web of Knowledge Home, [Online], Available: http:// isiwebof-knowledge. com/index.html (Accessed: 2006, Feb.20th).
11. Ware, C.: Information Visualization: Perception for Design (2nd Ed.). Morgan Kaufman, San Francisco, CA (2004)

# A Model for Interactive Web Information Retrieval

Orland Hoeber and Xue Dong Yang

University of Regina, Regina, SK S4S 0A2, Canada
{hoeber, yang}@uregina.ca

**Abstract.** The interaction model supported by web search engines has changed very little since the early days of web search. Users are required to formulate their queries with very little support from the system, and are provided with a list-based representation of the web search results that promotes a sequential evaluation of the document surrogates. The short queries used by web searchers, and the few pages viewed as a result of a web search are indications of the inadequate support provided for the users' information retrieval tasks. We propose a model for web information retrieval that uses visualization and interactive visual manipulation to support the users as they take an active role in satisfying their information needs.

## 1 Introduction

There have been significant advances in recent years in both the sizes of the indexes used by web search engines, and the algorithms used to match users' queries to these indexes [2,3]. Modern web search engines can perform well when users enter very specific queries; many highly relevant documents are often provided in the first few pages of the search results. However, if users are unable to craft adequate queries, or when their information needs are inherently ambiguous, web search engines do not perform well.

Most web search engine interfaces support a model of interaction based on traditional information retrieval, consisting of a cycle of query formulation, examining the search results, and either stopping or reformulating the query [5]. The problem with this model is that there is little support for the users to perform the two fundamental tasks: query formulation and search results evaluation. Evidence of this inadequate support is provided in two independent studies which have shown that web searchers often provide only one or two terms in their queries, and seldom view more than three pages of search results [10,9].

We propose a model for interactive web information retrieval that uses information visualization and interactive visual manipulation to support query formulation and search results exploration. Since the ability to read and assess textual information is a limiting factor [11], visual representations of the users queries as well as the search results can allow the users to more effectively interpret and make sense of the information provided. Interaction techniques allow the users to take a more active role in the web information retrieval process, rather than the passive role supported by the traditional model.

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 242–247, 2006.

In the remainder of this paper, we present a model for interactive web information retrieval, and provide a brief overview of the systems we have implemented based on this model: VisiQ [8] for interactive query refinement; HotMap [6] and Concept Highlighter [7] for interactive search results exploration. The paper concludes with a discussion and future directions.

## 2 Interactive Web Information Retrieval Model

Most web search engines use a traditional model of information retrieval [5]. This model consists of two primary stages: query formulation and search results evaluation. At some point in the search results evaluation stage, the users make a decision as to whether they are satisfied with the search results or not. If they are satisfied, the web search is considered a success. Otherwise, the users must decide whether they wish to reformulate their query. If they choose to not reformulate their query, the web search is considered a failure.

The two primary drawbacks of the traditional model are that there is little support for the users in formulating queries, and little support for users in evaluating the search results. Applying the principles of information visualization and interactive visual manipulation in order to support the users as they perform these complex tasks, we propose a model that allows the users to take an active role in the process. As illustrated in Figure 1, this model adds two new cycles in addition to the traditional query reformulation cycle: interactive query refinement and interactive search results exploration.

**Interactive Query Refinement**

With the traditional model, the query reformulation cycle can not occur until the users have first formulated an initial query and viewed the results of this search. If they make poor choices for their query terms that result in many non-relevant documents, there is little information provided to allow the user improve their query.

The interactive query refinement loop must provide *specific supplemental information* to the users that may be relevant as they craft a query. One of the challenges of query refinement is finding a source for this supplemental information that is relevant to the users' information needs. External, topic-specific knowledge bases may be necessary to resolve this challenge.

Any tool that provides support for interactive query refinement should allow the user to *focus on their query formulation task.* Therefore, any information provided should be within the context of the query. This is in contrast to the traditional model, where the query reformulation cycle flips the user between query formulation and search results evaluation.

Information visualization techniques can promote an easier interpretation and synthesis of the information provided regarding the users' queries. Further, the visual representations of complex data can result in much quicker comprehension than the equivalent textual representations. As such, *visual representations of the query information* for an interactive query refinement tool is highly recommended.

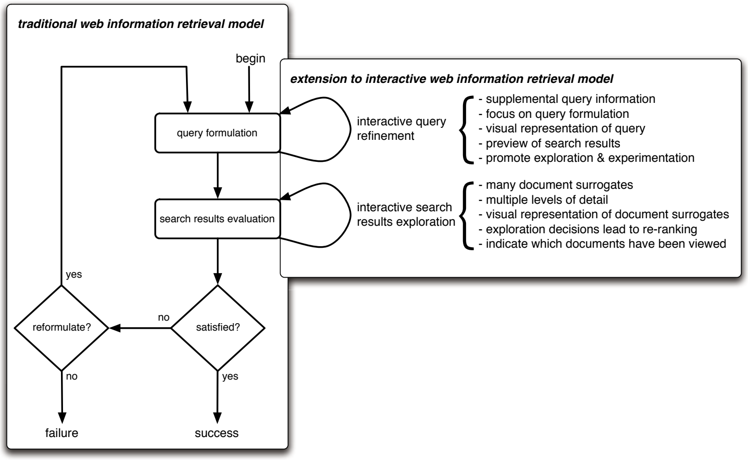



**Fig. 1.** The proposed model for interactive web information retrieval

As the users formulate their queries within this interactive query refinement loop, it is important to allow them to see how they are progressing. As such, *a preview of the search results* should be provided. This will help to promote an understanding of whether the users' queries are an accurate reflection of their information needs.

The interactive query refinement loop should *promote exploration and experimentation* with the query. As the users make refinement decisions, the results of these decisions should instantly be reflected in the visual representation of the query as well as in the preview of the search results. If minimal effort is required to return the users' queries to previous states, they will be able to easily experiment with "what if" scenarios, and explore the query space.

**Interactive Search Results Exploration**

In the traditional information retrieval model, the search results evaluation exists as a static stage in which the users evaluate the document surrogates one-by-one, and in the order provided by the web search engine. While this linear evaluation may be effective when there are a large portion of relevant documents in the first few pages of the search results, it is not very effective when the search results are ambiguous. It is in these cases that an interactive exploration of the search results is beneficial.

In order to promote an effective exploration of the web search results, *many document surrogates* must be retrieved and made available to the users. Providing a large number of document surrogates can allow the user to discover interesting features or patterns in the search results, and can direct the users to sets of documents that are more likely to be relevant.

Since a much larger number of document surrogates will be present in an interactive search results exploration tool, the user may wish to view both the details of specific document surrogates, and the features of the entire search results collection. Providing *coordinated views at different levels of details* can allow the users to take both a micro- and a macro-view of the web search results.

Just as information visualization techniques can allow users to more effectively make sense of their queries, generating *visual representations of the document surrogates* can allow the users to more effectively understand the features of the search results. Although converting the textual features of document surrogates into visual representations can be challenging, the benefits include the ability to provide a compact representation, and allow the users to "see" the information rather than "read" it.

As the users explore the web search results, their decisions should *result in a re-ranking* and subsequent re-sorting of the search results. This has the effect of bringing relevant documents to a more prominent location in the search results based on the features of the search results space. Further, the exploration decisions the users make must be reflected instantly in the visual representation of the search results. This will allow the users to more easily determine the outcome of their choices, and will allow them to decide whether their decisions are bringing them closer to fulfilling their information needs.

In any system that represents web search results, be it a static list or an exploratory system, it should be very easy for the user to access the specific documents. Further, as the users select documents to view, it should be *made clear that these documents have been viewed* when the user returns to the search results. This is especially important for a system that allows the users to re-rank the search results.

## 3 Implementation

**Interactive Query Refinement**
In order to explore the interactive query refinement cycle within this model, we have developed the VisiQ system to allow users to formulate and refine their queries within a visual interface [8]. This system makes use of a concept knowledge base devised for the topics in computer science that are stored in the ACM Computing Classification System [1].

The VisiQ system uses the concept knowledge base to visually represent the concepts related to the users query, and to suggest additional terms that may be relevant to their information need. This information is depicted in a graphical manner, allowing the users to easily identify their query terms, the concepts they are related to, and additional terms that have also been used to describe those concepts. The system allows the users to interactively refine their query by adding or removing terms via simple mouse click actions. A preview of the first five documents in the search results returned by the Google API [4] is provided as the users refine their query, allowing them to judge whether they are improving the accuracy of their query to their information need. More details on this work can be obtained from [8].

**Interactive Search Results Exploration**

We have developed two interfaces to help us understand the complexities of web search results exploration. These systems use a similar search results interface framework that is based on a grid represented at two levels of detail: an overview map and a detailed view. Both systems send the users' queries to the Google API [4], and retrieve the top 100 document surrogates. The overview map provides a compact representation of all 100 document surrogates in a single view. The detailed view shows the specific information about each document surrogate, and provides a link to the document. As the users view the documents, the link colours change from blue to purple (as per the defacto standard for web links), allowing the users to easily see which documents they have already viewed. Both of these systems support techniques for re-ordering the search results; they differ in the supplemental information provided and how this information is used to explore the search results.

HotMap [6] calculates the frequencies of the users' query terms within the document surrogates and visually represents this information via colour coding. Each query term is provided as a column in the grid-based layout; the colour coding allows the users to easily identify "hot" documents which make frequent use of the users' query terms. The search results can be visually explored using the overview map, and a nested sorting feature allows the users to re-rank the search results based on the importance they place on the query terms.

Concept Highlighter [7] matches the users queries to the concept knowledge base in a technique similar to that used by VisiQ. These matched concepts are used as the cluster centroids by a single-pass fuzzy clustering algorithm which classifies each of the document surrogates based on their similarity to the concepts. The system allows the users to select one or more concepts that are related to their information needs. Doing so instantly re-ranks the search results based on the fuzzy membership scores, and visually depicts these scores both in an overview map and a detail window.

These systems were developed to address different situations that may arise when users are evaluating and exploring the search results. HotMap can be very effective when users have a moderate degree of knowledge about their information needs, but use one or more ambiguous terms in their queries. By re-ranking the search results based on the importance they place on the query terms, more relevant documents may be moved to the top of the search results. Concept Highlighter can be very effective when the results of a search are ambiguous. In these cases, documents that are similar to the concepts of interest can be given greater prominence.

## 4 Conclusions and Future Work

In this paper, we have proposed a model for web information retrieval that includes cycles of interactive query refinement and interactive search results exploration. This is an advancement over the traditional information retrieval model, since it promotes the users in taking an active role in formulating and refining their query, and in evaluating and exploring their search results. This model is a step towards the development of a web information retrieval support system [12] that assists users as they seek the answers to their information needs.

Work is currently underway to generate a unified interactive web information retrieval system based on the various prototypes mentioned in this paper. The result will be a complete implementation that will fulfill the features of the model described in this paper. In addition to supporting interactive query refinement and interactive search results exploration, this system will provide a smooth and intuitive transition between the query formulation and the search results evaluation stages. We believe this unified system will result in a vast improvement over the existing web search engines, especially when the users' information needs are ambiguous, or when the users have difficulty in crafting specific queries.

## References

1. ACM. ACM computing classification system. http://www.acm.org/class/.
2. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the Seventh International World Wide Web Conference*, 1998.
3. Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The Google file system. In *Proceedings of the Nineteenth ACM Symposium on Operating System Principles*, 2003.
4. Google. Google web API. www.google.com/apis/, 2005.
5. Marti Hearst. User interfaces and visualization. In Ricardo Baeza-Yates and Berthier Ribeiro-Neto, editors, *Modern Information Retrieval*. Addison-Wesley, 1999.
6. Orland Hoeber and Xue-Dong Yang. The visual exploration of web search results using HotMap. In *Proceedings of the International Conference on Information Visualization*, 2006.
7. Orland Hoeber and Xue-Dong Yang. Visually exploring concept-based fuzzy clusters in web search results. In *Proceedings of the Atlantic Web Intelligence Conference*, 2006.
8. Orland Hoeber, Xue-Dong Yang, and Yiyu Yao. Visualization support for interactive query refinement. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, 2005.
9. Bernard J. Jansen and Udo Pooch. A review of web searching studies and a framework for future research. *Journal of the American Society for Information Science and Technology*, 52(3), 2001.
10. Craig Silverstein, Monika Henzinger, Hannes Marais, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1), 1999.
11. James A. Wise, James J. Thomas, Kelly Pennock, David Lantrip, Marc Pottier, Anne Schur, and Vern Crow. Visualizing the non-visual: Spatial analysis and interaction with information from text documents. In *Proceedings of IEEE Information Visualization*, 1995.
12. Yiyu Yao. Information retrieval support systems. In *Proceedings of the 2002 IEEE World Congress on Computational Intelligence*, 2002.

# Representing and Querying Line Graphs in Natural Language: The *iGraph* System

Leo Ferres[1], Avi Parush[1], Zhihong Li[1,2], Yandu Oppacher[1,2], and Gitte Lindgaard[1]

[1] Human-Oriented Technology Laboratory
Carleton University
Ottawa, Ontario, Canada
[2] School of Computer Science
Carleton University
Ottawa, Ontario, Canada
{leo_ferres, avi_parush, gitte_lindgaard}@carleton.ca
{zhihong, yoppache}@scs.carleton.ca

**Abstract.** Numerical information is often presented in graphs. However, this medium is problematic for certain audiences such as inexperienced graph readers, people with visual impairments and users of mobile technologies featuring small screens. We have developed a system called iGraph which provides short verbal descriptions with the information depicted in graphs and a way of interacting with it by means of dialogue in natural language. In this paper, we present the general architecture of the system and its representational and querying mechanisms, together with a glimpse of the natural language interface.

## 1 Introduction

Presenting numerical data by means of graphs and charts (line graphs, pie charts, etc.) takes advantage of the human ability to recognize certain visual patterns that, in turn, summarize and help interpret otherwise seemingly meaningless collections of numbers. All the cognitive mechanisms used to find these visual patterns (and indeed reason with them in a given domain of knowledge) depend, rather obviously, on processes carried out by the human visual faculty. If this faculty is not available, be it because the person is engaged in other visual tasks (such as driving) or using a small presentation display (e.g. working on small-screen PDAs or cell-phones), or because the faculty is considerably damaged (e.g. in people with visual impairments), then a concrete problem resides in finding alternative ways to model this faculty and make (at least some of) the visual information encoded in graphs available to the audiences described in the above contexts. One possible solution to this problem is to exploit the flexibility and ubiquity of natural language to help gain access to the information locked up in visual representations of data. To this purpose, we have implemented *iGraph*, a system that provides descriptions and a query mechanism to interact with graphs by means of a NLI (Natural Language Interface).

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 248–253, 2006.

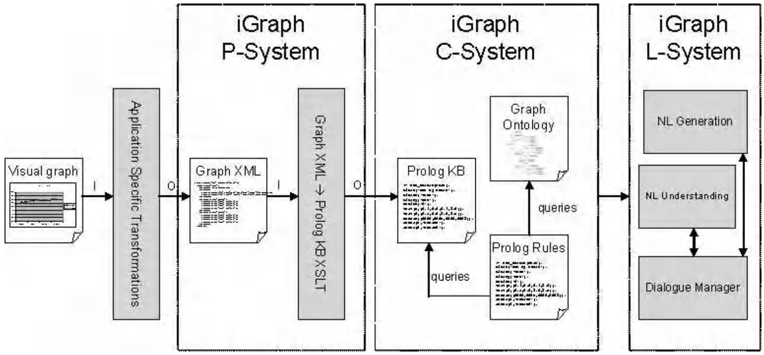



**Fig. 1.** The processes of iGraph

## 2 The *iGraph* System

The *iGraph* (or *inspectGraph*) system is a software application that exploits a reasoning engine and an NLI to provide automatic descriptions of data graphs and to help their interpretation and querying in situations in which visual presentation is not possible or not helpful. It consists of three main sub-systems: The P-, C- and L-systems. First, the P-System is in charge of acquiring information about a graph in some common mark-up language and translating it into a knowledge representation language that will permit inferencing. Second, the C-System is in charge of taking the representation from the P-System and reasoning with it by means of a set of rules. Finally, the L-System provides the human interface to the logical language of the C-System, exploiting a natural language comprehension and generation module and a dialogue manager. Figure 1 shows the basic architecture of the *iGraph* system. The process progresses as follows: once a graph has been chosen (sent by email to a PDA or mobile phone, selected on a spreadsheet, etc.), an application-dependent script translates it into an XML file (for more details see Section 3 below). This XML file is the input to an intermediate function that translates it into a representation used by the reasoning engine (this first step is carried out by *iGraph*'s P-System). A set of rules of inference and domain knowledge about graphs allow its querying (the C-System). Lastly, the L-System (see [1]), allows for natural language querying, either over an Instant Messaging program (MSN Messenger or GoogleTalk) or by some speech recognition engine. In the remaining sections, the P- and C-Systems are explained in more detail. Section 3 below deals with the P-System and an intermediate XML representation of time-series data, which simplifies interoperability with standard applications such as Excel or SPSS.

## 3 The Representation of Graphical Information

The first stage of processing in *iGraph* is the acquisition of an XML file that encodes all the necessary elements and attributes to represent a data graph. To do this, *iGraph*'s input representation of graphs relies on the XML Schema provided by [2]. The second stage of processing in *iGraph* involves making the XML representation of graphs available to a reasoning engine. Given its flexibility, support and wealth of resources and documentation, SWI-Prolog was chosen as the reasoning engine for *iGraph* [3]. Thus, the XML snippet defining a graph has to be translated into the SWI-Prolog format and then provide a set of rules that will query a graph KB. In order to do this, an XSL (EXtensible Stylesheet Language) transformation was developed to translate from the XML tree defining the graph to the SWI-Prolog KB format (the XSLT file, together with the Java classes used to do the transformations can be found in [4]). Once the KB for a given line graph $G$ has been built, the SWI-Prolog engine compiles it with a set of rules designed to query graphs in general and line graphs in particular. These rules are of different kinds and serve different purposes ranging from reasoning about very local phenomena in $G$ (such as the value of a single point) to very global phenomena (such as the general trend of $G$ or what it is about, given the axes). Section 4 below provides more details about these rules and their nature.

## 4 Querying Line Graphs

Given a domain $D$ (line graphs, in this case) and a user $u$ in context $c$, what kinds of queries will $u$ have the need to make? In order to answer this question, User-Needs Analysis (UNA) and Wizard-of-Oz studies were carried out (for a detailed discussion of results, see [5]). The top four requests regarding queries gave a clear idea of the kinds of queries to implement: 1) purpose, type and titles; 2) describe point by point; 3) describe general trends and then details and 4) to "keep it [i.e. the descriptions] simple". For this paper, we will adopt the convention of using a possible natural language translation of the Prolog (logic) queries in order to make he explanation more intuitive. These queries will be written in *italics*. Needless to say, there may be many ways to lexicalize a given semantic representation: the chosen lexicalizations are for exemplification purposes only and do not stand for the only way a user has to pose the query to the system, though, obviously, the semantic representation for all possible lexicalizations will be the same one.

**Purpose, type and axes.** A few rules have been implemented to deal with these requests for information. The three `title` predicates mentioned above, for instance, query the string associated to the ordinate and the abscissa. They correspond to queries such as: *what are the axes (about)?*, *what's axis X/Y/the ordinate/the abscissa (about)?*, *what are the titles (about), what is the title of the X-axis/ordinate (about)?*, *what are the titles/the title of the Y-axis?*, etc. Purpose, in turn, is considerably more complicated and the system runs into a version of the Frame problem trying to find what information may be relevant to reason about the task at hand. A few general domain knowledge propositions have been added to the ontology in order to augment reasoning: for instance, a rule in the general domain ontology states that if the predicate `titleX` is `"year"` or `"yr."` and `xScale` is an integer or `'06`, then every value in the X-Axis will be a year, and related reasoning may be carried out if this information is known.

**Point-by-point queries.** One of the most important rules (if not the most important) of the graph querying subsystem of *iGraph* is the rule that defines the predicate `point`. This predicate allows the querying of every single data point in the graph and several other predicates (most notably `line`) are built on it. This predicate helps answer queries such as the following: *What's the GPA for grade 1 in 2000? When is GPA for grade 2 higher than 2.5? What grade has a GPA of 2.6 and in what year? What are the GPAs for year 2000 for grades 1 and 2?* and so on. There are also several auxiliary predicates that significantly enrich the query system of *iGraph*. Two of these predicates are `greater_than`, `less_than` and `equals`, which help draw comparisons between any two points in a graph, from the same or different ordinate values, series and abscissa values. Finally, two other important predicates are `max` and `min`, which query a given graph for the maximum and minimum values of its $Y$-axis. These two predicates help answer questions such as *what's the maximum? What's the minimum? What's the maximum for grade 1? What's the series of the maximum value? What's the minimum value of 2002 in any series?* and so on and so forth.

**Line queries.** The predicates above help answer queries about given points, including point comparisons. The next set of predicates help answer queries about the different segments that those points define. Given the proposed architecture of the system regarding its scalability and also issues brought about by the natural language processing interface described below, the most important predicate of this section, `line`, has been defined as a rule using the `point` predicate explained in the previous section. The `line` predicate is mostly internal to the workings of *iGraph*, but is central in defining other predicates to successfully query a graph's segments. Three predicates that use the `line` predicate are `increase`, `decrease` and `plateau`, which answer queries such as *Does it increase from 2002 to 2003 in Grade 1? When are there decreases? List all increases in Grade 1. Are there any years for grade 1 with no decreases or increases?*, and so on. Again two more predicates that build on `increase` and `decrease` are `steady_increase` and `steady_decrease`, which help summarize the graph if it contains too many points, for instance. Obviously, these predicates may also be used to query graphs regarding steady increases and decreases between any two points. Potential queries using these predicates include *List all steady increases. What's the initial value of the steady increase starting in 2000 for Grade 1? What year does the steady decrease in Grade 1 end?* and so on.

In order to give other useful information about the segments of line graphs, several predicates that allow for querying more "visual" characteristics of these graphs have also been implemented. Three of these predicates are of particular interest because of their psychological underpinnings: `moderate`, `slight` and `sharp` are predicates that help summarize the mathematical slope (the angle) of the increase or decrease of a particular segment. These higher-order concepts should help deal with slopes more intuitively. These predicates help answer queries such as the following: *List all sharp increases for Grade 1. How many sharp decreases are there for Grade 2?* and so on. These queries are about segments or lists of segments in a graph, but there are also queries that aim at the whole graph: we call these global queries.

**Global graph queries.** Finally, *iGraph* also implements predicates used to query a very general view of the line graph. These predicates are also used for summarization purposes, but differ from other summarizing predicates in that they apply to the whole graph rather than to points or lines. Two of these predicates are `linearInterpolation` and `linearRegression`, which query whether a graph has a general upwards or downwards trend with the two different calculations [6]. These predicates answer queries such as: *does Grade 1 have an upwards or downwards trend?* and so on.

## 5 The Natural Language Interface

This paper is mostly concerned with the architecture and a few research-driven set of predicates of *iGraph*. However, a few words should be said about its NL (Natural Language) interface. The Natural Language Interface (NLI) to the system has two main purposes: i) to generate a short description of the graph and ii) to allow for interaction with the information portrayed in the graph. The first point aims at presenting and summarizing the graph in order to provide the user with the most important information (as uncovered in the User-Needs Analysis reported in [5]) in the graph. This is done by the predicate `describe`, which includes a set of queries in a given syntagmatic order to provide a textual summary of the graph. Below is an example of the kinds of summaries *iGraph* is currently providing:

```
Description: The graph is about paper clip sales. It's a one- variable line graph.
The y axis is millions of dollars and the x axis is month. The maximum value of the
y axis is 93, the minimum is 10. The graph has a general upwards trend. Evolution:
It is quite irregular. It starts at 63. There is a moderate decrease in January to
23. There is a steady increase from 23 to 91 from January to March... Finally, there
is a steady increase from 38 to 69 from September to November.
```

This text is read aloud by the speech-synthesis engine of choice (in our case, this is [7]). For the second objective of *iGraph*, we have implemented a dialogue system that allows for querying the graph in natural language. The main interface is GoogleTalk, an Instant Messaging (IM) program that is connected to *iGraph*'s language understanding and language generation subsystems by means of a dialogue manager. This dialogue manager helps orchestrate user input with the *iGraph*'s language model to answer queries in NL. Below is a simple example of a short interaction between a user and *iGraph*. Presently, however, the grammars used to understand the user's inputs and the dialogue are quite limited. We have prioritized the implementation of the empty data structures needed and, *a posteriori*, fill them out with graph-related content (a richer dialogue, more in-grammar possibilities, etc.). This notwithstanding, as the dialogue above shows, users may query the graphs in a relatively free form.

```
IGRAPH:  This is iGraph, how can I help you?
USER:    Load the graph highlighted in Excel, please.
IGRAPH:  Ok. Done.
USER:    Could you describe it?
IGRAPH:  <Description as in the example above>
USER:    What was the maximum again?
IGRAPH:  The maximum is 93.
USER:    What are the titles?
IGRAPH:  The y-axis is millions of dollars, the x-axis is month.
USER:    What was the y-axis again?
IGRAPH:  The y-axis is millions of dollars.
USER:    Does it increase from January to February?
IGRAPH:  Yes, there is an increase from January to February
```

## Acknowledgements

This paper was written with the generous support of HTX (Health Technology Exchange), Cognos Inc. and the Ontario Research Network for E-Commerce. The authors would also like to thank the members of the HOTLab, especially, Shelley Roberts and Jing Liu for constant support.

## References

1. Ferres, L., Parush, A., Oppacher, Y., Li, Z.: Talkin' to graphs: A natural language interface to working with graphical representations of time-series data. (2006)
2. Wollowski, M.: Search and inference with diagrams. In: Proceedings of the The Ninth IASTED International Conference on Internet & Multimedia Systems & Applications, Honolulu, HI (2005)
3. Wielemaker, J.: SWI-Prolog 5.6 Reference Manual, Updated for version 5.6.4. Human-Computer Studies, University of Amsterdam, Amsterdam, The Netherlands. (2006)
4. Ferres, L., Li, Z., Parush, A., Oppacher, Y.: XSL transformations from Graph XML to SWI-Prolog. http://www.carleton.ca/∼lferres/resources/graph.xsl, Internet Resource. (2006)
5. Ferres, L., Parush, A., Roberts, S., Lindgaard, G.: Helping people with visual impairments gain access to graphical information through natural language: The igraph system. In: Proceedings of the 10th International Conference on Computers Helping People with Special Needs. Lecture Notes in Computer Science, Springer–Verlag (2006)
6. Fasciano, M.: Génération intéegrée de textes et de graphiques statistiques. PhD thesis, Département d'informatique et de recherche opérationnelle, Université de Montréal (1996)
7. Schröder, M., Trouvain, J.: The German text-to-speech synthesis system MARY: A tool for research, development and teaching. International Journal on Speech Technology **6** (to appear)

# MusicSpace: A Multi Perspective Browser for Music Albums

Hans Jörg Müller and Antonio Krüger

University of Münster, Münster, Germany

**Abstract.** Finding music albums in large databases is an inherently difficult problem, especially if users do not know exactly what they are looking for. MusicSpace lays out music albums spatially according to different perspectives and provides a Zoomable User Interface to enable navigation within these perspectives. The choice of perspectives was inspired by the theory of Conceptual Spaces, so similar items regarding each perspective are placed near to each other. We believe that the concept of laying out items according to Conceptual Spaces has the potential to generalize to all kind of real world items like pictures, food, wines or books that can be described by perceptual qualities as time, color, emotion or taste.

## 1 Introduction

Imagine a user entering her favorite music store. She does not know exactly what she is looking for, and the albums are laid out according to some schema. If the store is good, similar albums are placed near to each other, and if she understands how the albums are ordered, she can find albums she is looking for. But there are many dimensions the albums could be ordered by: genre, artist, release date, region, etc. When the collection is digitally available, e.g. as her personal MP3 collection or at an online music store, the spatial arrangement of the items could be adjusted according to her current search criteria. We propose to layout the albums according to Conceptual Spaces and suggest a navigation method within this representation with a Zoomable User Interface. This approach scales both to different display sizes, so the same interface can be used on a mobile MP3 player, a desktop PC or a wall-size display, as to different domains, like picture galleries, restaurants, wine stores, book stores etc.

## 2 MusicSpace

Conceptual Spaces [1] are a framework in the family of geometric models to describe conceptual knowledge using vector spaces. A music album can be identified with a point along multiple dimensions, like time, longitude and latitude of recording place, or the valence and arousal dimensions of associated emotions. The two important aspects of this approach are that 1. in each perspective, items similar regarding this perspective are placed near to each other, and 2. if the user understands the structure of the perspective, she always knows where to look to

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 254–256, 2006.

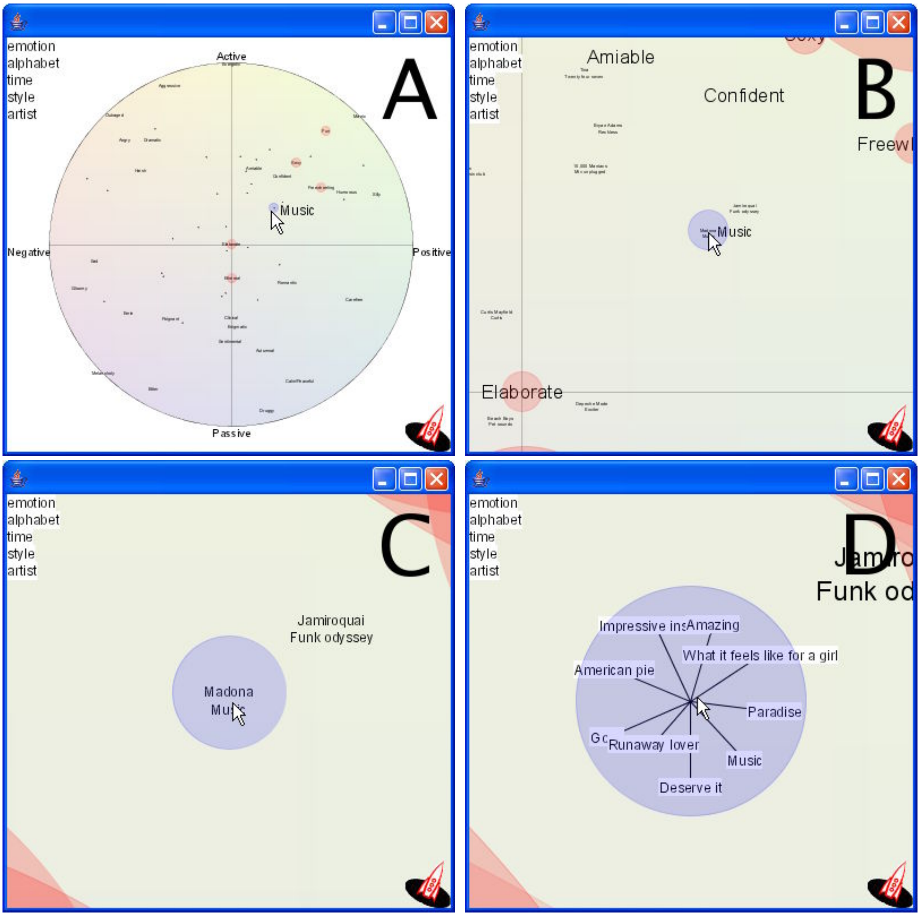



**Fig. 1.** Zooming to an album of Madonna in the emotion perspective (A, B, C), and playing songs from that album (D)

find the items she is looking for. In the time domain, for example, the albums are ordered by their release date. Thus, the user can browse the space to find music albums that fit her search criteria, for example ‘Music from the 80’s’. To deal with the possibly infinite size and resolution of Conceptual Space domains and to solve the focus+context problem, we used a Zoomable User Interface. ZUIs allow for both infinite screen size and infinite resolution by providing continuous zooming in 2D. In the domain of emotion (Fig. 1), for example, an overview can be presented at a low zoom factor (A), and the user can continuously zoom into a certain point to get more details (B, C) and see the individual songs in the album (D). The user can switch between perspectives by clicking on one of the buttons in the upper left corner. Furthermore, it is possible to apply dynamic query filters by selecting regions in multiple perspectives (Fig. 2). We believe

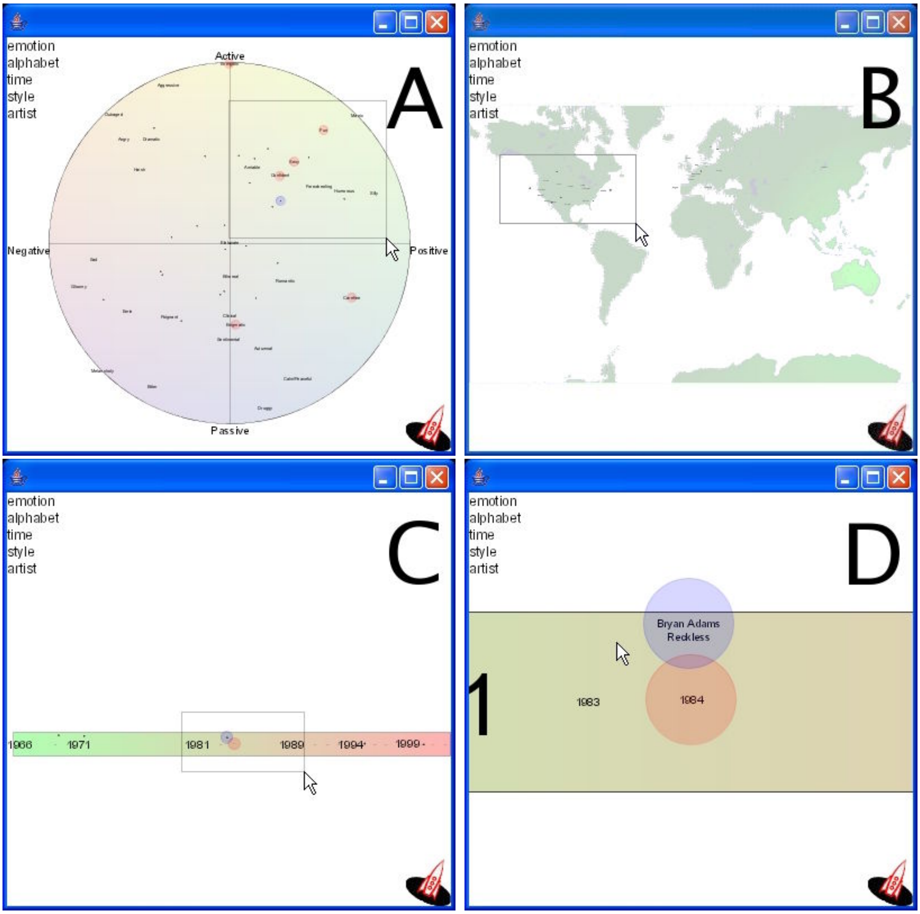



**Fig. 2.** Selecting some happy (A) american (B) music from the 80's (C,D)

that Conceptual Spaces provide a sophisticated conceptual framework for application designers to order items spatially and thus enable visual navigability using multiple perspectives.

*Acknowledgements.* This work was done at the Tangible User Interface group of Sony Advanced Technology Center, Stuttgart.

## References

1. Gärdenfors, P.: Conceptual Spaces: The geometry of thought. MIT Press, Cambridge, MA, 2000.

# Large Display Size Enhances User Experience in 3D Games

Tao Lin, Wanhua Hu, Atsumi Imamiya, and Masaki Omata

Department of Computer Science and Media Engineering, University of Yamanashi,
Takeda 4-3-11, Kofu, Yamanashi Prefecture, 400-8511, Japan
{Lintao, Imamiya, Omata, Huwanhua}@hci.media.yamanashi.ac.jp

**Abstract.** Large displays are becoming commonplace for home and office computers. Although researchers have quantified the benefits of working on large displays, there has been little investigation of how large displays physiologically and emotionally impact the user. Using subjective and physiological measures, we compared the user's experience when game playing on a large display versus a small display. We found that the large display caused greater physiological responses and higher subjective ratings of excitement. These physiological results were mirrored in the participants' subjective reports. The study contributes to understanding of interaction with large displays and refining the requirements for what constitutes effective and desirable human-computer interaction (HCI).

**Keywords:** Large display, User experience, Physiological measure.

## 1 Introduction

Researchers and engineers in the usability field have begun to devote much attention to analyzing user experience during interactions. Their studies mainly investigated the impact on users of interaction tasks ([1], [2]), while little effort has been spent on understanding how physical interfaces influence the user experience. With advances in technologies, large displays are becoming relatively commonplace for both office and home computers. Researchers have suggested that large displays will require new ways of thinking about HCI [3]. For example, several studies have reported that large displays can improve user performance and reduce or eliminate gender bias on certain tasks ([4], [5], [6], [7]). However, there remains a lack of understanding of how physically large displays affect the user experience.

On the other hand, recent developments in physiological sensing technologies have made it feasible to objectively evaluate user experiences using physiological measures. Physiological measures such as galvanic skin response (GSR), heart rate (HR), pupil size, respiration rate, and blood volume pulse reflect autonomic nervous system (ANS) activity in response to a user's emotional state [8]. Furthermore, physiological data are available in high resolution and continuous time series. Physiological measures have been employed as an objective metric in HCI evaluation ([9], [10], [11], [12], [13]).

A. Butz et al. (Eds.): SG 2006, LNCS 4073, pp. 257 – 262, 2006.

This study seeks to answer the following question using subjective and physiological measures: Does display size impact on player experience in a 3D game environment? In order to specify the question, we established the following hypotheses:

**Hypothesis 1:** Participants will report more excitement and stress when playing on large displays than playing on small displays in subjective reports.

**Hypothesis 2:** Participants will experience higher physiological responses when playing on large displays than playing on small displays, due to greater arousal.

**Hypothesis 3:** The difference in participants' HR changes in the two conditions will correlate to the differences in their subjective responses of arousal-related measures (e.g. excitement).

## 2 Experimental Design

Eighteen university students aged 18 to 33 participated in the experiment. We chose a popular 3D video game called *Super Mario 64* as the experimental task, which was manufactured by NINTENDO®. The subjects' task was to run all regulated paths and obtain eight red stars as quickly and accurately as possible. They played the game session in two conditions: on a large display (100 × 75 cm) and on a small display (36 × 28.5 cm). The game was played on a NINTENDO$^{64}$. Players were free to choose their preferred distance from the displays. GSR and electrocardiogram (EKG) were collected using ProComp Infiniti System and Cardipro2.0 Software from *Thought Technologies*™. We placed GSR sensors on the left fingers and pre-gelled surface EKG electrodes in standard configuration of two electrodes on chest and one on the abdomen. Fig. 1 shows the large and small displays, a screen shot of the task and an experimental trial.


**Fig. 1.** Large and small displays, screen shot of the task and an experimental trial

At the outset of the experimental task, a 10-minute baseline (GSR and EKG) was recorded. Participants then played a 10-minute game session on the large display and on the small display, in random order. After each session, participants rated the experience using a 5-point Likert scale, with 1 corresponding to "strongly disagree" and 5 corresponding to "strongly agree." They were asked to rate the statement, "this condition was exciting," as well as to rate how easy, challenging, engaging, and stressful the experience was. The t-test statistic was used to determine whether the subjective ratings were influenced by display size. After the two tasks, participants discussed their impressions of the experiment.

# 3 Results and Discussion

## 3.1 Subjective and Physiological Responses to Display Size

From the subjective reports, there was a significant difference between the large and small displays in exciting (t = 3.951, p = 0.001) and stressful (t = 1.932, p = 0.042), but we did not find significant difference between the screens in the easy, challenging, or engaging ratings. Table 1 summarizes these results.

**Table 1.** Subjective ratings of each experience on a scale from 1 ("strongly disagree") to 5 ("strongly agree")

| Experience | Large Display | | Small Display | | Difference | |
|---|---|---|---|---|---|---|
| | Mean | St. Dev | Mean | St. Dev | t | p |
| Easy | 3.4 | 0.59 | 3.1 | 0.58 | 0.17 | 0.87 |
| Challenging | 2.711 | 0.518 | 2.644 | 0.558 | 0.375 | 0.710 |
| Engaging | 3.381 | 0.832 | 3.441 | 0.780 | -0.23 | 0.820 |
| Exciting | 3.969 | 0.511 | 3.300 | 0.480 | 3.951 | 0.001* |
| Stressful | 3.121 | 0.600 | 2.75 | 0.518 | 1.932 | 0.042* |

* $p < 0.05$

Means for the physiological data were compared using independent samples t-tests. Our second hypothesis was that psychological responses are greater when playing on large displays compared to small displays. As a result, we expected that changes in HR and GSR from baselines (ΔHR and ΔSC) would be greater when playing on the large display. Overall, mean ΔHR was significantly higher when playing on the large display (ΔHR = 5.31 beats per minute) as compared to the small display (ΔHR = 3.52 beats per minute, t = 3.30, $p < 0.05$). This effect was seen in 14 subjects (see Fig. 2).

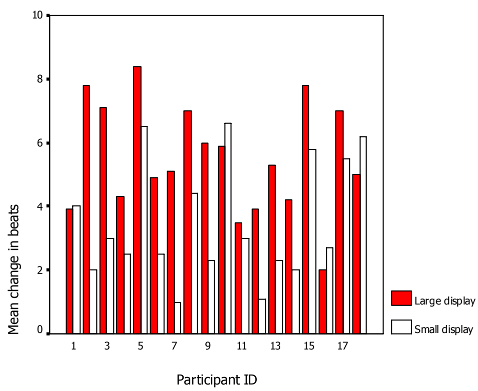



**Fig. 2.** ΔHR was greater in 14 of 18 subjects when playing on the large display than the small display

We also compared GSR changes from the baseline ($\Delta SC$) between the two conditions. Subjects experienced larger GSR when playing on the large display than playing the small display ($\Delta SC_{\text{large display}}$ = 2.54 mSiemens, $\Delta SC_{\text{small display}}$ = 1.65 mSiemens, $t = 2.04$, $p < 0.05$). Fourteen of the 18 subjects are consistent with the pattern. The GSR data also confirmed our second hypothesis.

### 3.2 Correlation Between Subjective Data and Physiological Data

To permit comparison of the time-series physiological data with the one-time subjective data, we normalized the measures, transforming them into dimensionless numbers between negative one and one. For each individual, the difference between the experimental conditions was divided by the maximum range of that individual's response. The time-series physiological data were normalized using the following formula:

$$\text{Physiological}_{\text{normalized}} = \frac{\text{mean L - mean S}}{\max\{\text{peak L - min L, peak S - min S}\}} \tag{1}$$

where L refers to playing on the large display and S refers to playing on the small display. The corresponding normalizing equation for the one-time subjective data was:

$$\text{Subjective}_{\text{normalized}} = \frac{\text{L - S}}{4} \tag{2}$$

These normalized measures were then correlated across all individuals using the Bivariate Person Correlation. We found that normalized HR was correlated with normalized exciting ratings ($R = 0.64$, $p = 0.01$, see Fig. 3).

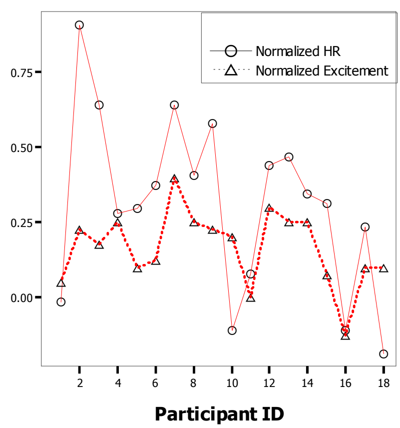



**Fig. 3.** Normalized HR is correlated with normalized excitement ratings ($R = 0.65$, $p = 0.01$)

The positive correlation between them would show that the amount by which subjects' excitement ratings increased when playing on the large display was proportional to the amount that HR increased during that condition.

The results suggested that the large display subjectively and physiologically enhance players' experience when playing the 3D games. This might be attributed to the following reasons. First, the wider visual angle offered by the large display might cause the greater physiological responses. In this study, all participants chose a distance from the large display that was further than for the small display, but less than the proportional increase in size, thus producing a wider FOV. The wider FOV may have contributed to the increased physiological responses. Previous studies have reported that a wider FOV can lead to an increased subjective sense of presence [4], and when the sense of presence increases, physiological reactions could become more intensive. Thus, the results of our study also provided objective evidence for the fact that large display can increase the sense of presence [9]. In addition, researchers in the entertainment industry also reported that large displays filling a wider FOV can increase the level of involvement experienced by users [14].Second, almost all subjects have no or somewhat experience in using large displays, and the novelty to large displays might cause short-term physiological responses at the begin of the task as compared to the small-size screen. Moreover, during the post-gaming interviews, subjects were asked how the game experience was affected by the display size. Sixteen of 18 subjects felt that the 3D game world on a large screen became more realistic and that there was less distraction from the activity. Therefore, they tended to feel more present in the game world. When asked which display they would choose to play on if given a choice, all subjects reported chose the large display. These remarks also supported our hypotheses.

The difference in subjects' responses to large and small displays may also depend on the nature of the task. We also compared large and small displays using a simple 2D game called *Luxor*, in which subjects just need to move their mouse to clear the screen of marbles. For this game, the display size did not have a significant impact on physiological responses or subjective ratings. In our highly interactive 3D game, subjects had to control six degrees of freedom. The weak interactive characteristics of the 2D game may have reduced the effect of large displays. Thus, future work needs to refine the importance of these task characteristics.

## 4 Conclusions

The interaction with large displays and the evaluation of user experience are both areas ripe for advancement. In this experiment, the application of physiological measurement and analysis have showed exciting potential. The experimental results confirm the three hypotheses proposed. The confirmation of the hypotheses suggested that large displays indeed have impact on user experience. Large displays lead to greater physiological responses and enhanced subjective ratings of excitement as compared to small displays. Moreover, the normalized physiological results correlate to subjectively reported experience. However, the relationship between task characteristics and display size should also be further investigated.

The study contributes to understanding of interaction with large displays and refining the requirements for what constitutes effective and desirable human-computer interaction (HCI). Furthermore, the study has practical implications for helping game players and game manufacturers maximize entertainment of games.

## Acknowledgements

We thank the members of the HCI group at the University of Yamanashi for their support of this research. This study was supported in part by the Grants-in-Aid for Scientific Research of the Japan Society for the Promotion of Science and by the RIEC of Tohoku University awarded to A. Imamiya.

## References

1. Ravaja, N., Salminen, M., Holopainen, J., Saari, T., Laarni, J., and Järvinen, A. Emotional Response Patterns and Sense of Presence during Video Games: Potential Criterion Variables for Game Design, in Proceedings of the Third Nordic Conference on Human-Computer Interaction (Tampere, October 2004), ACM Press, (2004) 339–347.
2. Ravaja, N., Saari, T., Laarni, J., Kallinen, K. and Salminen, M.: The psychophysiology of video gaming: Phasic emotional responses to game event, Proceedings Digital Games Research Conference, Vancouver, British Columbia (2005).
3. Swaminathan, N. and Sato, S.: Interaction design for large displays. Interactions, 4(1), (1997) 15–24.
4. Czerwinski, M., Tan, D. S., Robertson, G. G.: Women take a wider view. Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems, (2002) 195–202.
5. Tan, D. S., Stefanucci, J. K., Proffitt, D. R., and Pausch, R.: The infocockpit: Providing location and pace to aid human memory. Workshop on Perceptive User Interfaces (2001).
6. Tan, D., Gergle, D., Scupelli, P. and Pausch, R.: With similar visual angles, larger displays improve spatial performance. CHI 2003, New York: ACM Press (2003).
7. Tan, D. S., Gergle, D., Scupelli, P., Pausch, R.: Physically large displays improve path integration in 3D virtual navigation tasks. Proc. CHI2004, ACM Press (2004).
8. Andreassi, J. L.: Psychophysiology: Human behavior and physiological response (4th Edition). Mahwah, NJ: Lawrence Erlbaum Associates (2000).
9. Meehan, M., Insko, B., Whitton, M. and Brooks, F.: Physiological measures of presence in stressful virtual environments. Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. San Antonio, Texas (2002).
10. Mandryk, R. L. and Inkpen, K.: Physiological indicators for the evaluation of co-located collaborative play. Proceedings of Computer Supported Cooperative Work (CSCW 2004), Chicago, IL, USA (2004).
11. Picard, R. W., Vyzas, E., and Healey, J.: Toward machine emotional intelligence: Analysis of affective physiological state. IEEE PAMI, 23(10), (2001) 1165–1174.
12. Lin, T., Hu, W. H., Omata, M. and Imamiya, A.: Do Physiological data relate to traditional usability indexes. Proceedings of the Australian Conference on Computer-Human Interaction OZCHI'05, Canberra, Australia, November 21–15 (2005).
13. Healey, J. A.: Wearable and automotive system for affect recognition from physiology, doctoral dissertation. Massachusetts Institute of Technology, Cambridge, MA (2000).
14. Childs, I.: HDTV—putting you in the picture. IEE Review, 34(7), (1988) 261–265.

# Author Index