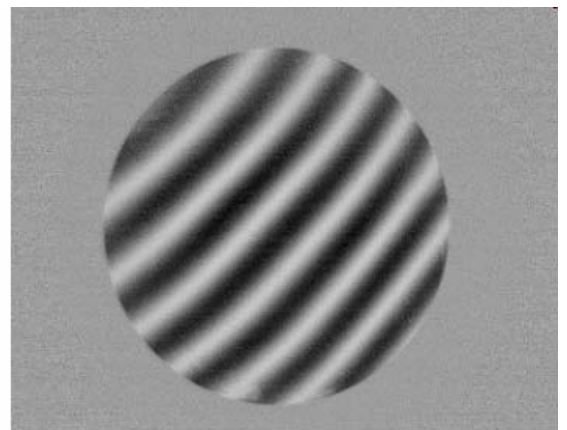
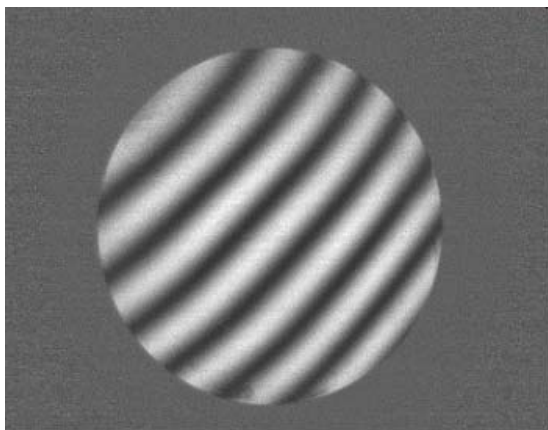


К.В. ЕЖОВА

МОДЕЛИРОВАНИЕ И ОБРАБОТКА ИЗОБРАЖЕНИЙ

УЧЕБНОЕ ПОСОБИЕ



Санкт-Петербург
2011

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

К.В. Ежова

МОДЕЛИРОВАНИЕ И ОБРАБОТКА ИЗОБРАЖЕНИЙ

УЧЕБНОЕ ПОСОБИЕ



**Санкт-Петербург
2011**

Ежова К.В. **Моделирование и обработка изображений. Учебное пособие.** – СПб: НИУ ИТМО, 2011. – 93с.

Учебное пособие подготовлено на кафедре прикладной и компьютерной оптики СПбНИУ ИТМО и предназначено для подготовки магистров по профилю 200400.06 «Компьютерная оптика». В пособии излагаются основные теоретические принципы, необходимые для построения алгоритмов, предназначенных для моделирования и обработки оптических изображений.

Рекомендовано Учебно-методическим объединением ВУЗов Российской Федерации по образованию в области приборостроения и оптоэлектроники для студентов высших учебных заведений, обучающихся по направлению подготовки 200400 «Оптехника», протокол №8 от 25.11.2011г.

Рецензент: проф., д.т.н. Коняхин И.А.



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2011

© К.В. Ежова 2011

ВВЕДЕНИЕ

В настоящее время цифровая обработка и последующее распознавание изображений - одно из интенсивно развивающихся направлений научных исследований. Многие отрасли техники, имеющие отношение к получению, обработке, хранению и передаче информации, в значительной степени ориентируются в настоящее время на развитие систем, в которых информация имеет характер изображений.

В пособии рассматриваются вопросы формирования и кодирования цифрового изображения и дальнейшей его обработки, включающей в себя геометрические, логические и арифметические операции над изображением, действие функций препарирования и фильтрацию изображений. Отдельное внимание уделено моделям линейных искажений изображений, алгебраическим и итерационным методам восстановления изображений.

1. ОСНОВЫ РЕГИСТРАЦИИ, МОДЕЛИРОВАНИЯ И ЦИФРОВОЙ ОБРАБОТКИ ОПТИЧЕСКОГО ИЗОБРАЖЕНИЯ

1.1. Основные понятия компьютерной обработки изображений

При рассмотрении вопросов, связанных с моделированием и обработкой изображений, необходимо сформулировать определение самого понятия «*изображение*». Самое простое и широкое определение этого понятия: изображение – это то, что мы видим. Другое определение: изображение – это информация, пригодная для визуального восприятия. В зависимости от происхождения условно можно выделить следующие *типы изображений*:

1. рисованное или печатное (художник, полиграфия, принтер);
2. оптическое (распределение интенсивности электромагнитного поля создаваемое оптическим прибором в некоторой области пространства (области локализации) например, на сетчатке глаза, на экране при проецировании, в плоскости приемника объектива фотоаппарата);
3. фотографическое (оптическое изображение, зарегистрированное на фотоматериале в результате химического процесса);
4. электронное или цифровое (оптическое изображение, зарегистрированное с помощью электронного приемника, например, ПЗС-матрицы, сканера, микроденситометра). Электронным обычно называют изображение, отображаемое на экране монитора.

Очевидно, что данное деление условное: изображение из одного типа сразу переходит в другой. Необходимо дать изображению формализованное определение, которое позволит описать этот объект математически и манипулировать им для достижения определенных целей. Эти манипуляции принято называть «обработка изображений».

Обработка изображений может производиться в различных целях:

1. Изменение (искажение) изображения с целью достижения каких-либо эффектов (художественное улучшение). Эти преобразования не будут рассматриваться в рамках данного курса;
2. *Image Processing* – визуальное (заметное глазом) улучшение качества изображения (коррекция яркости и контраста, цветокоррекция и т.п.); объективное улучшение качества

изображения (устранение искажений типа дисторсия, смаз, расфокусировка и т.п.);

3. *Image Analysis* – проведение измерений на изображении (анализ интерферограмм, гартманогамм, ФРТ и т.п.);
4. *Image Understanding* – распознавание образов (распознавание символов, отпечатков пальцев, лиц, приборы наведения и т.п.)

Для достижения поставленных целей рассмотрим решение следующих задач:

1. Дискретизация, квантование и кодирование изображений.
2. Геометрические преобразования изображений.
3. Логические и арифметические операции над изображениями.
4. Фильтрация изображений.
5. Препарирование изображений.

1.1.1. Цифровое изображение

Под *изображением* понимается функция двух вещественных переменных $I(x, y)$, где I – это интенсивность (яркость) в точке с координатами (x, y) . Иногда обработке подвергается не все изображение, а некоторая его часть, которую в англоязычной литературе принято называть *region-of-interest*, *ROI* (*область представляющая интерес*, *ОПИ*).

Для обработки на компьютере изображение должно быть дискретизировано и квантовано. Дискретизованное и квантованное изображение называется *цифровым*.

Цифровое изображение $A(m, n)$ представлено в дискретном двумерном пространстве, где m – номер строки, а n – номер столбца. Элемент, расположенный на пересечении m -ой строки и n -го столбца называется *пиксел* (pixel – picture element). Интенсивность пиксела может описываться либо вещественным, либо целым числом. Относительная интенсивность в вещественных числах обычно изменяется от 0 до 1, а в целых числах от 0 до 255.

Обычно физический сигнал, возникающий в точке (x, y) , является функцией зависящей от многих параметров (z – глубина, λ – длина волны, t – время). Однако, в рамках данного курса, будут рассматриваться статические, и чаще монохроматические изображения.

Среди характеристик цифровых изображений следует выделить:

Размер: этот параметр может быть любым, но часто выбирается исходя из особенностей регистрации изображения (например, видеостандарты PAL (625, 4:3), SECAM (625, 4:3), NTSC (525, 4:3)), особенностей последующей обработки (алгоритмы быстрого

преобразования Фурье предъявляют особые требования) и т.п. Хотя в последнее время фреймграбберы интерполируют изображение до любых размеров, библиотеки БПФ справляются с изображением любых размеров.

Количество цветов (глубина цвета): точнее количество бит, отводимое для хранения цвета, определяется упрощением электронных схем и кратно степени 2. Изображение для хранения информации о цветах которого необходим 1 бит называется бинарным. Для хранения полутоновых (gray scale, gray level) изображений используется обычно 8 бит. Цветные изображения хранятся обычно с использованием 24 бит по 8 на каждый из трех цветовых каналов.

Разрешение: измеряется обычно в dpi (dot per inch – количество точек на дюйм). Например, на экране монитора разрешение обычно 72 dpi, при выводе на бумагу – 600 dpi, при регистрации на ПЗС-матрице с размером одного элемента 9 мкм разрешение составит почти 3000 dpi. В процессе обработки разрешение можно изменить: на само изображение это не повлияет, но изменится его отображение устройством визуализации.

1.1.2. Методы обработки изображения

В основе алгоритмов обработки изображений положены в основном интегральные преобразования: свертка, преобразование Фурье и статистические методы.

Методы обработки изображений классифицируют обычно по количеству пикселей участвующих в одном шаге преобразования:

- *поточечные* методы в процессе выполнения преобразуют значение в точке $a(m,n)$ в значение $b(m,n)$ независимо от соседних точек;
- *локальные* (окрестностные) методы для вычисления значения $b(m,n)$ используют значения соседних точек в окрестности $a(m,n)$;

глобальные методы определяют значение $b(m,n)$ на основе всех значений исходного изображения $A(m,n)$.

1.2. Основные понятия теории сигналов

Прежде всего, следует дать определение такому понятию, как сигнал. В зависимости от контекста оно может принимать различные значения. В общем случае *сигнал* – это изменение некоторой физической величины. В зависимости от области определения говорят о временной, частотной или пространственной форме представления сигнала.

Сигналы чаще всего рассматривают как функцию, заданную в некоторых физических координатах. По этому критерию можно выделить одномерные сигналы (зависящие, например, от времени), двумерные сигналы, заданные на плоскости (например, изображение) и трехмерные сигналы (описывающие, например, пространственные объекты). Математической моделью сигнала чаще всего являются скалярные функции. Но в ряде случаев приходится использовать более сложные модели. Например, для описания электромагнитного поля удобно использовать комплексные функции, а для цветных изображений – трехкомпонентные векторные функции.

Если область определения сигнала непрерывна, то он называется *непрерывным* или *аналоговым*. Название «аналоговый» непрерывным сигналам дано потому, что они являются «аналогами» реальных физических процессов, происходящих в действительности. Такой сигнал и его аргументы могут принимать любые значения. Примером аналогового сигнала является изменение напряжения. Сигнал, аргументы которого принимают счетное множество значений, называется *дискретным*. Примером дискретного сигнала может служить совокупность значений напряжения, измеряемых с некоторым интервалом. В этом случае сигнал определен лишь в дискретные, то есть отдельные моменты времени. Если же сам сигнал принимает счетное множество значений, то он называется *квантованным*. *Цифровыми* называются дискретные квантованные сигналы.

1.2.1. Преобразование сигналов оптическими системами

Сигналы в оптических системах претерпевают разнообразные преобразования. Для математического описания этих преобразований в общем случае необходимо задать все возможные пары входных и выходных сигналов. Однако объем такого описания настолько велик, что фактически исключает возможность его практического использования. Поэтому модели оптических систем и описания преобразования сигналов в них строятся по иерархическому принципу. При этом преобразование сигналов представляются как совокупность некоторых элементарных преобразований. Оптический прибор при этом рассматривается как каскад преобразователей информации, а оптическая система является линейным фильтром сигнала.

Каждый изображающий прибор принимает информацию от предыдущего элемента каскада и передает последующему. Входной сигнал называют *предметом*, а выходной – *изображением*. При построении модели абстрагируемся от конкретного физического содержания предмета

и изображения и будем рассматривать их как некоторые *обобщенные сигналы* или функции $I(x)$ и $I'(x')$ обобщенных интенсивностей от векторов обобщенных координат $x = \begin{pmatrix} x \\ y \end{pmatrix}$ $x' = \begin{pmatrix} x' \\ y' \end{pmatrix}$.

Задачей изображающего прибора является преобразование входного сигнала – функции предмета $I(x)$ в выходной сигнал – функцию изображения $I'(x')$. Модель оптического прибора, описывающая общие закономерности формирования изображения в оптических системах, не связанные с физическими принципами их работы (*внешняя функциональная модель*), есть оператор L , осуществляющий преобразование:

$$I'(x') = L[I(x)] \text{ или } I(x) \xrightarrow{L} I'(x') \quad (1.2.1)$$

В теории изображения предполагается, во-первых, что этот оператор должен удовлетворять условию линейности. *Линейные преобразования* – это преобразования, для которых выполняется принцип суперпозиции. Математически это записывается следующим образом: преобразование L является линейным, если для любых сигналов I_k заданных в линейном пространстве, и скаляров a_k справедливо:

$$L\left[\sum_k a_k I_k\right] = \sum_k a_k L[I_k]. \quad (1.2.2)$$

В соответствии с этим выражением, изображение суммы равно сумме изображений.

Система, осуществляющая линейные преобразования, называется линейной. Для таких систем наиболее распространенным является описание с помощью *импульсной реакции*, определяемой как отклик оператора на дельта-функцию:

$$h(x') = L[\delta(x)]. \quad (1.2.3)$$

В теории оптических изображающих систем эта импульсная функция называется *функцией рассеяния точки (ФРТ)*, и представляет собой изображение (пятно рассеяния) светящейся точки единичной энергии.

Во-вторых, для изображающего оператора L должно выполняться условие *изопланатичности* или пространственной инвариантности (инвариантности к сдвигу):

$$I(x - a) = L[I'(x' - a')]. \quad (1.2.4)$$

В соответствии с этим выражением, при смещении предмета на вектор a изображение только смещается на вектор a' , причем a' пропорционален a , а именно:

$$a' = Va, \quad (1.2.5)$$

где V – матрица обобщенных увеличений.

1.2.2. Структурные передаточные характеристики

Изображающие оптические системы могут давать изображения различного качества в плане передачи тонкой структуры предмета. При рассмотрении передачи структуры предмета используют *нормированную ФРТ*, энергия которой равна единице:

$$h_H(x') = \frac{h(x')}{H}; \int_{-\infty}^{+\infty} \int h_H(x') dx' = 1, \quad (1.2.6)$$

где $H = \frac{E'}{E}$ – передняя зональная обобщенная светосила,

$E = \iint_S I(x) dx$ и $E' = \iint_S I'(x') dx'$ – обобщенная энергия участка предмета и изображения соответственно.

Нормированная ФРТ характеризует в чистом виде передачу системой структуры предмета.

В дальнейшем для упрощения будем опускать в обозначении индекс H и под $h(x')$ подразумевать нормированную ФРТ.

Рассмотрим теперь предмет и изображение приведенными на одну поверхность, то есть изображаемыми с единичным увеличением. Благодаря этому исключаются масштабные преобразования, описываемые матрицей обобщенных увеличений. Передача структуры предмета в таком случае будет описываться следующим выражением:

$$I(x') = \int_{-\infty}^{+\infty} \int I(x) h(x'-x) dx, \quad (1.2.7)$$

где координаты x и x' рассматриваются в одном масштабе. В математике подобное выражение называется *сверткой* и записывается следующим образом:

$$I' = I \otimes h. \quad (1.2.8)$$

Таким образом, передача структуры предмета описывается сверткой функции предмета с нормированной функцией рассеяния точки.

1.2.3. Частотные передаточные характеристики

Описание передачи структуры предмета в виде приведенного выше выражения свертки является полным, но не всегда удобно и наглядно. Это связано в первую очередь с тем, что сами функции предмета $I(x)$ и изображения $I'(x')$ не вполне наглядно представляют тонкую структуру предмета и изображения, и должны быть заменены другими.

Наиболее подходящим эталоном структуры является *периодический гармонический объект*. Он характеризуется *пространственной частотой* ν (величиной, обратной периоду T), углом ориентации θ , амплитудой u и начальным сдвигом b (начальной фазой $\varphi = 2\pi\nu b$). Причем u и b могут быть объединены в одну комплексную величину – *комплексную амплитуду* $g = ue^{-i\varphi}$. Чем больше пространственная частота ν , тем тоньше структура предмета. Пространственная частота имеет единицы измерения, обратные обобщенным координатам.

Теперь перейдем к такому важному понятию, как *спектр*. В общем случае спектром называется совокупность значений какой-либо величины, характеризующей систему или процесс.

Структурное содержание сложного объекта $I(x)$ или изображения $I'(x')$ нагляднее описывается *спектрами пространственных частот* $\tilde{I}(\nu)$ и $\tilde{I}'(\nu')$, показывающих распределение комплексных амплитуд по пространственным колебаниям, на которые могут быть разложены $I(x)$ и $I'(x')$. Функции предмета и изображения и их спектры частот связаны между собой преобразованием Фурье в соответствии со следующими соотношениями:

$$I(x) \xrightarrow{F} \tilde{I}(\nu) \text{ или } \tilde{I}(\nu) = \int_{-\infty}^{+\infty} \int I(x) e^{2\pi i(\nu^T x)} dx; \quad (1.2.9)$$

$$I'(x') \xrightarrow{F} \tilde{I}'(\nu') \text{ или } \tilde{I}'(\nu') = \int_{-\infty}^{+\infty} \int I'(x') e^{2\pi i(\nu'^T x')} dx'. \quad (1.2.10)$$

Таким образом, передача изображающей системой структуры предмета нагляднее всего описывается как передача его спектра пространственных частот. Если теперь применить преобразование Фурье к обеим частям выражения свертки, описывающего передачу структуры предмета, учитывая свойства преобразования Фурье, получаем:

$$I'(v) = \tilde{I}(v) \cdot D(v), \quad (1.2.11)$$

где функция $D(v) = F[h(x')]$ представляет собой двумерное преобразование Фурье от ФРТ и называется *оптической передаточной функцией (ОПФ)*. Приведенное выражение называется *соотношением фильтрации* и показывает, что спектр пространственных частот изображения получается как произведение спектра пространственных частот предмета на оптическую передаточную функцию системы. Эта функция служит наиболее наглядной и удобной структурной (частотной) передаточной характеристикой.

ОПФ в общем случае является комплексной функцией:

$$D(v) = F[h(x')], \quad (1.2.12)$$

где ее модуль $T(v)$ – *частотно-контрастная характеристика (ЧКХ)* или *модуляционная передаточная функция (МПФ)*, а ее аргумент $\varphi(v)$ – *частотно-фазовая характеристика (ЧФХ)* или *фазовая передаточная функция (ФПФ)*.

1.3. Дискретизация и квантование непрерывных изображений

Очень редко изображения, получаемые в информационных системах, имеют цифровую форму. Поэтому их преобразование к этому виду является обязательной операцией, если предполагается использовать цифровую обработку, передачу, хранение. Данное преобразование включает в себя две процедуры.

Первая состоит в замене непрерывного кадра дискретным и обычно называется дискретизацией, а вторая выполняет замену непрерывного множества значений яркости множеством квантованных значений и носит название квантования. При цифровом представлении каждому из квантованных значений яркости ставится в соответствие двоичное число, чем и достигается возможность ввода изображения в ЭВМ.

1.3.1. Дискретизация непрерывных изображений

Дискретизация – это преобразование непрерывного сигнала в последовательность чисел (отсчетов), то есть представление этого сигнала по какому-либо конечномерному базису. Это представление состоит в проектировании сигнала на данный базис.

Наиболее удобным с точки зрения организации обработки и естественным способом дискретизации является представление сигналов в виде выборки их значений (отсчетов) в отдельных, регулярно

расположенных точках. Такой способ называют *растрированием*, а последовательность узлов, в которых берутся отсчеты – *растром*. Интервал, через который берутся значения непрерывного сигнала называется *шагом дискретизации*. Обратная шагу величина называется *частотой дискретизации*,

Многообразие базисов дает возможность образования различных дискретных представлений непрерывного изображения. Однако наиболее употребительной является периодическая дискретизация, в частности, дискретизация с прямоугольным растром.

Пусть $x_H(t_1, t_2)$ - непрерывное изображение, а $x(i_1, i_2)$ - соответствующее ему дискретное, полученное из непрерывного путем прямоугольной дискретизации. Рис.1.3.1 иллюстрирует расположение отсчетов на плоскости (t_1, t_2) при прямоугольной дискретизации.

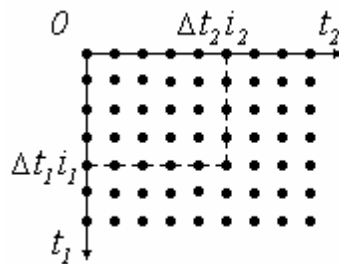


Рис. 1.3.1. Расположение отсчетов при прямоугольной дискретизации

Существенный вопрос, возникающий в ходе дискретизации: с какой частотой брать отсчеты сигнала для того, чтобы была возможность его обратного восстановления по этим отсчетам? Очевидно, что если брать отсчеты слишком редко, то в них не будет содержаться информация о быстро меняющемся сигнале. Скорость изменения сигнала характеризуется верхней частотой его спектра. Таким образом, минимально допустимая ширина интервала дискретизации связана с наибольшей частотой спектра сигнала (обратно пропорциональна ей).

Для случая равномерной дискретизации справедлива *теорема Котельникова*, опубликованная в 1933 году в работе “О пропускной способности эфира и проволоки в электросвязи”. Она гласит: если непрерывный сигнал $x(t)$ имеет спектр, ограниченный частотой $F_{\max}$, то он может быть полностью и однозначно восстановлен по его дискретным

отсчетам, взятым с периодом $T = \frac{1}{2F_{\max}}$, т.е. с частотой $f_d = 2F_{\max}$.

Восстановление сигнала осуществляется при помощи функции $\text{sinc}(x) = \frac{\sin(x)}{x}$. Котельниковым было доказано, что непрерывный сигнал,

удовлетворяющий приведенным выше критериям, может быть представлен в виде ряда:

$$x(t) = \sum_{k \in Z} x(kT) \operatorname{sinc}\left(\frac{t}{T} - k\right). \quad (1.3.1)$$

Эта теорема называется теоремой отсчетов. Функция $\operatorname{sinc}(x)$ называется еще *функцией отсчетов* или *Котельникова*, хотя интерполяционный ряд такого вида изучал еще Уитакер в 1915 году. Функция отсчетов имеет бесконечную протяженность по времени и достигает наибольшего значения, равного единице, в точке $k = \frac{t}{T}$, относительно которой она симметрична.

Каждую из этих функций можно рассматривать как отклик идеального *фильтра низких частот* (ФНЧ) на дельта-импульс, пришедший в момент времени kT . Таким образом, для восстановления непрерывного сигнала из его дискретных отсчетов, их необходимо пропустить через соответствующий ФНЧ. Следует заметить, что такой фильтр является некаузальным и физически нереализуемым.

Приведенное соотношение означает возможность точного восстановления сигналов с ограниченным спектром по последовательности их отсчетов. *Сигналы с ограниченным спектром* – это сигналы, спектр Фурье которых отличен от нуля только в пределах ограниченного участка области определения. Оптические сигналы можно отнести к ним, т.к. спектр Фурье изображений, получаемых в оптических системах, ограничен из-за ограниченности размеров их элементов. Частоту $F_{\max}$ называют частотой Найквиста. Это предельная частота, выше которой во входном сигнале не должно быть спектральных компонентов.

1.3.2. Квантование изображений

При цифровой обработке изображений непрерывный динамический диапазон значений яркости делится на ряд дискретных уровней. Эта процедура называется квантованием. При этом происходит преобразование непрерывной переменной x в дискретную переменную x_{KB} , принимающую конечное множество значений $\{r_1, \dots, r_L\}$. Эти значения называются уровнями квантования. В общем случае преобразование выражается ступенчатой функцией (рис. 1.3.2). Если яркость x отсчета изображения принадлежит интервалу $\left[d_j, d_{j+1}\right]$ (т.е., когда $d_j < x \leq d_{j+1}$), то исходный отсчет заменяется на уровень квантования r_j , где $d_j, j = \overline{1, L+1}$

- пороги квантования. При этом полагается, что динамический диапазон значений яркости ограничен и равен $[d_1, d_{L+1}]$.

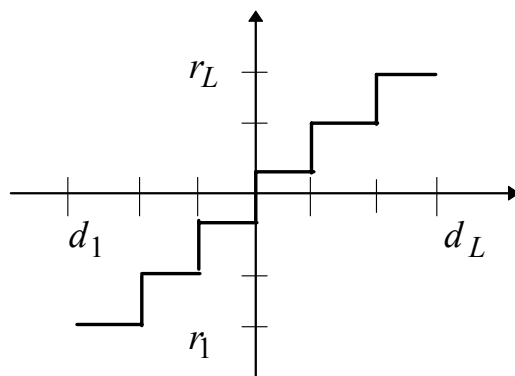


Рис.1.3.2. Функция, описывающая квантование

Основная задача при этом состоит в определении значений порогов d_j и уровней r_j квантования. Простейший способ решения этой задачи состоит в разбиении динамического диапазона на одинаковые интервалы. Однако такое решение не является наилучшим. Если значения интенсивности большинства отсчетов изображения сгруппированы, например, в «темной» области и число уровней L ограничено, то целесообразно квантовать неравномерно. В «темной» области следует квантовать чаще, а в «светлой» реже. Это позволит уменьшить ошибку квантования. $\varepsilon = x - x_{kv}$

В системах цифровой обработки изображений стремятся уменьшить число уровней и порогов квантования, так как от их количества зависит объем информации, необходимый для кодирования изображения. Однако при относительно небольшом числе уровней на квантованном изображении возможно появление ложных контуров. Они возникают вследствие скачкообразного изменения яркости проквантованного изображения и особенно заметны на пологих участках ее изменения. Ложные контуры значительно ухудшают визуальное качество изображения, так как зрение человека особенно чувствительно именно к контурам. При равномерном квантовании типичных изображений требуется не менее 64 уровней.

На рис.1.3.4.(а) и 1.3.4.(б) приведены результаты равномерного квантования изображения соответственно на 256 и 14 уровней квантования.

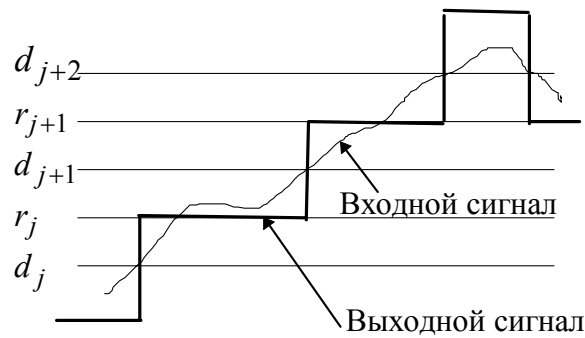


Рис.1.3.3. К механизму возникновения ложных контуров



а)



б)

Рис.1.3.4. Результаты равномерного квантования

В темных частях изображения на рис. 1.3.4.(б) заметны ложные контуры.

Чтобы избежать неравномерного квантования, которое не может быть выполнено с помощью стандартного АЦП, используют нелинейные преобразования (рис.1.3.5). Отсчет x исходного изображения подвергается нелинейному преобразованию, чтобы плотность распределения вероятностей преобразованных отсчетов y была равномерной. Затем отсчеты y квантуются с равномерным шагом и подвергаются обратному нелинейному преобразованию.

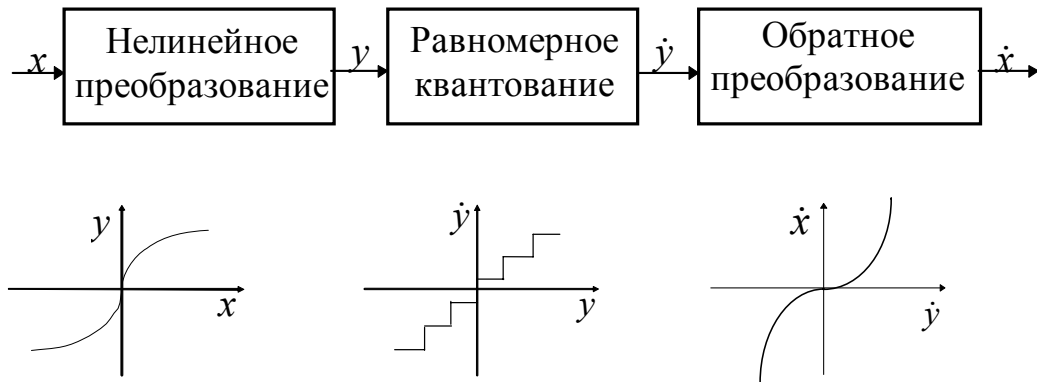


Рис.1.3.5. Квантование с предварительным нелинейным преобразованием

Для разрушения ложных контуров перед равномерным квантованием к отсчетам яркости добавлять шум с равномерной плотностью распределения вероятностей. Такой шум переводит одни отсчеты изображения на уровень выше, а другие на уровень ниже. Тем самым разрушаются ложные контуры. Дисперсия добавляемого шума должна быть небольшой, чтобы не привести к искажениям, воспринимаемым как «снег» на изображении, но достаточной для разрушения ложных контуров. Часто используется равномерно распределенный шум на интервале $[-\Delta/2, \Delta/2]$. Результаты равномерного квантования на 14 и 8 уровней изображения с добавлением шума приведены на рис.1.3.6.(а) и 1.3.6.(б). При 8-ми уровнях квантования добавляемый шум становится слишком заметным, однако ложные контуры разрушены практически полностью.



а)



б)

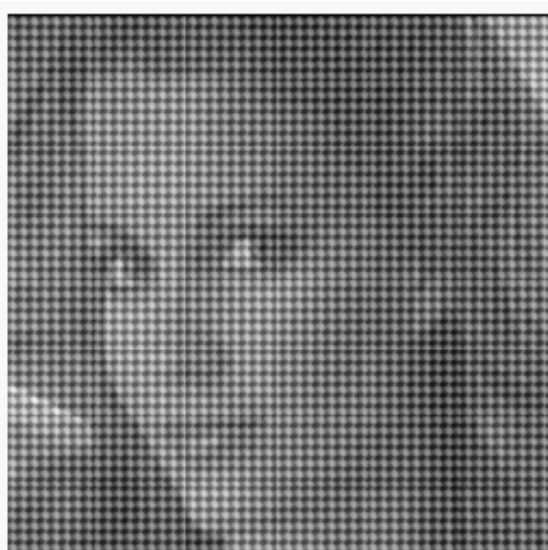
Рис.1.3.6. Результаты равномерного квантования с предварительным добавлением шума

Еще один метод квантования используется в полиграфии. Это метод формирования растровых бинарных (2-х уровневых) изображений из полутоновых. При печати (например, газет или журналов) изображение формируется из белых и черных точек. Для этого все исходное изображение разбивается по пространственным координатам на одинаковые квадратные блоки. Обычно блок содержит 5×5 элементов. К каждому отсчету блока добавляется число с соответствующими координатами из матрицы возмущающего сигнала, размеры которой равны размерам блока. Например, в качестве матрицы возмущающего сигнала используют числа:

$$\begin{bmatrix} 40 & 60 & 150 & 90 & 10 \\ 80 & 170 & 240 & 200 & 110 \\ 140 & 210 & 250 & 220 & 130 \\ 120 & 190 & 230 & 180 & 70 \\ 20 & 100 & 160 & 50 & 30 \end{bmatrix}.$$

Эта операция повторяется для всех блоков. Получаемое при этом изображение квантуется на два уровня. На рис. 1.3.7.(а) приведено полутоновое изображение с добавленным возмущающим сигналом. На рис. 1.3.7.(б), (в) приведены результаты бинарного квантования изображения с добавленным возмущающим сигналом (рис.1.3.7.(б)) и без него (рис.1.3.7.(в)).

Бинарное растровое изображение обеспечивает значительно лучшее зрительное впечатление, чем обычное бинарное изображение. Передача шкалы яркости при растривании достигается благодаря изменению геометрических размеров белого пятна, наблюдаемого на черном фоне. Если в блоке сгруппировались «светлые» отсчеты, то геометрические размеры белого пятна максимальны и равны размеру блока. При уменьшении яркости его геометрические размеры уменьшаются. Глаз человека выполняет локальное усреднение, создавая иллюзию наблюдения полутонового изображения.



а)



б)



в)

Рис.1.3.7. Растривание изображений

Процедура растривания особенно эффективна при печати изображений с высоким разрешением, когда одиночное пятно едва различимо глазом.

1.4. Ввод изображений

Как уже было сказано выше, оптическое изображение с точки зрения теории сигналов является двумерным непрерывным сигналом. В таком виде оно не пригодно для обработки в компьютерных системах, и, следовательно, должно быть преобразовано. Для этого выполняются операции дискретизации (по пространственным координатам) и квантования (по интенсивности). Получившееся цифровое изображение представляет собой массив дискретных отсчетов (последовательность

цифровых значений). Устройство, выполняющее операции дискретизации и квантования, в теории сигналов называют АЦП (аналого-цифровой преобразователь), а выполняющее обратное преобразование – ЦАП (цифро-аналоговый преобразователь).

Все компьютерные изображения, форматы их представления и программы создания и редактирования делят на два больших класса – векторные и растровые.

Векторные изображения представляют собой набор математически описанных объектов. Такой подход применяется при создании и хранении чертежной документации, планов, схем, диаграмм и т.п. Понятно, что векторная графика не применима для представления и обработки фотографических и тому подобных изображений. В рамках данного курса она рассматриваться не будет.

Изображение, представленное в цифровом виде, имеет вид прямоугольной матрицы (*растра*, растровой или битовой карты, *bitmap*), состоящей из набора дискретных элементов. Каждый такой элемент растра носит название *пиксел* (от английского *pixel – picture element*). Графические изображения, представленные в таком виде, получили название *растровых*.

Количество элементов (пикселей) на единицу длины называется – *разрешением*. Оно измеряется в большинстве случаев в *точках на дюйм (dpi, сокращенное от dot per inch)* или *пиксел на дюйм (ppi, сокращенное от pixel per inch)*. По сути оба эти понятия обозначают одно и то же. Разница лишь в том, что в первом случае единичный элемент изображения назван точкой (*dot*), а во втором – пикселем (*pixel*). Разрешение цифровых изображений – понятие неоднозначное, поскольку каждая стадия процесса воспроизведения накладывает свои требования и ограничения.

На этапе выполнений преобразований цифрового изображения в компьютере понятие разрешающей способности в достаточной мере условно. Фактически оно определяет, какого размера будет изображение в случае его вывода (например, на печать или экран). Ни на какие цифровые преобразования разрешение не влияет. Все цифровые преобразования производятся над пикселями, поэтому на этапе компьютерной обработки, разрешения не играет никакой роли.

На этапе вывода может применяться большое количество разнообразных устройств. Здесь под разрешением понимают количество точек, которое может вывести то или иное устройство на единицу длины. Понятно, что при этом один пиксел изображения в цифровом виде может быть не равен одному пикселю устройства вывода.

Для того чтобы определить, какое разрешение устройства ввода должно быть выбрано для сохранения структуры исходного изображения, воспользуемся теоремой Котельникова. Согласно ей, для передачи определенной пространственной частоты (например, 100 лин/мм) потребуется сканирование с вдвое большей частотой (200 лин/мм). И действительно, если при сканировании миры, имеющей шаг 100 лин/мм, используется разрешение в 100 линий на мм, то ее структура будет передана полностью в том случае, если каждая линия миры попадет на каждую линию сканирующего устройства. Если же линии миры окажутся между сканирующими линиями, то получим серое поле. Пространственная частота сканера в 200 лин/мм означает, что имеется 400 переходов черное/белое или 400 считывающих элементов на мм. Таким образом, получаем, что при сканировании миры с частотой 100 лин/мм необходимо иметь разрешение сканирующего устройства как минимум 400 точек на мм (ppm).

1.4.1. Глубина цвета

Цвет каждого пиксела цифрового изображения описывается несколькими числами (в зависимости от используемой цветовой системы). Количество бит, отводимое на представление информации о цвете каждого пиксела, называют *глубиной цвета (color depth)* или битовой глубиной цвета (*bit depth*). Иногда под цветовой глубиной понимают максимальное количество цветов, которые можно представить.

Глубина цвета определяет, как много цветов может быть использовано при отображении одного пиксела. Например, если цветовая глубина равна 1 бит, то пиксел может представлять только один из двух возможных цветов – белый или черный. Если цветовая глубина равна 8 бит, то количество возможных цветов равно $2^8 = 256$. При глубине цвета 24 бит количество цветов превышает 16 миллионов, что фактически превосходит способность глаза человека разрешать цвета. Такой режим называется *True Color (истинный цвет)*. В связи с тем, что 24-разрядное представление неудобно с точки зрения обработки изображения, обычно в режиме True Color используется 32 бита. В случае 32-разрядного представления информации о цвете младшие три байта по-прежнему описывают цвет точки, а старший байт либо управляет дополнительными параметрами (например, альфа-каналом, информацией о взаимном перекрывании объектов или глубине в трехмерном изображении), либо не используется. Понятно, что при таком представлении увеличивается размер изображения, однако существенно возрастает скорость его обработки центральным и графическим процессорами компьютера.

1.4.2. Квантование цвета

Квантование цвета (color quantization) используется для получения малого числа характерных цветов в изображении. Задачу квантования в данном случае можно сформулировать как выбор заданного количества «наилучших» цветов, имеющихся в полноцветном изображении, и замены всех остальных цветов изображения подходящими заместителями из этого списка. Раньше процесс квантования цвета был необходим потому, что видеосистема компьютера могла работать лишь с ограниченной цветовой палитрой (как правило, 256 цветов). Теперь оно используется с целью уменьшения размера графического файла, создания спецэффектов, повышения резкости границ и т.п.

Самым простым подходом здесь является выбор комплекта цветов для палитры с равномерным распределением каждой из цветовых компонент. Он обеспечивает широкий выбор цветов, но при этом не учитывается тот факт, что в большинстве изображений нет равномерного цветового распределения.

На данный момент существует несколько методик квантования цвета. Одним из наиболее эффективных является метод *квантования цветов медианным сечением*. При этом цветовое пространство рассматривается как трехмерный куб. Каждая ось куба соответствует одному из трех основных цветов: красному, зеленому или синему. Каждая из трех сторон разбивается на 255 равных частей, деления на осях нумеруются от 0 до 255, причем большее значение соответствует большей интенсивности цвета. Метод медианного сечения делит куб на 256 параллелепипедов, каждый из которых содержит примерно одинаковое количество пикселей. При таком разбиении куба центральная точка каждого параллелепипеда представляет оптимальный выбор для цветовой палитры. В той области куба, которая густо заполнена точками, будет больше параллелепипедов и, соответственно, в палитру попадет больше цветов. А там, где точек меньше, будет взято меньшее количество цветов. При этом ни один цвет не будет отброшен полностью, а предпочтение будет отдано тем цветам, которые встречаются чаще.

1.5. Сжатие изображения

Хранение и передача изображений является трудоемкой задачей. При хранении изображение занимает не малые объемы памяти, а при передаче по информационным каналам - немало времени. Поэтому задача представления изображений в компактной форме (сжатие данных) является весьма актуальной. При этом должны быть разработаны алгоритмы, как для кодирования, так и для декодирования

(восстановления) изображений. Рассмотрим некоторые из методов сжатия изображений. Особо остановимся на стандартизованных методах: LZW и JPEG.

Методы сжатия можно разделить на две группы: методы без потерь и с потерями. Первая группа методов сжатия обеспечивает восстановление без потерь и искажений. Это не всегда необходимо. Часто исходный сигнал уже содержит искажения и шумы такие, что небольшие искажения при кодировании (в пользу высокой степени сжатия) не испортят качества изображения в целом (методы с потерями).

Методы сжатия часто подразделяют на последовательные и энтропийные. Последовательное сжатие использует в работе при наличии повторяющихся участков. Энтропийное с целью сокращения к минимуму избыточности информации.

1.5.1. Групповое сжатие

Метод группового сжатия основан на том предположении, что в некоторые области сигнала содержат последовательности элементов с повторяющимися значениями. Элементы с одинаковыми значениями заменяются на одно значение и их количество. Существует много схем группового сжатия, одну из которых можно проиллюстрировать следующим образом:

original data stream:

17 8 54 0 0 0 97 5 16 0 45 23 0 0 0 0 3 67 0 0 8

run-length encoded:

17 8 54 0 3 97 5 16 0 1 45 23 0 5 3 67 0 2 8

Чаще всего для кодирования используется схема, которая называется PackBits. По аналогии с хранением отрицательных чисел, каждые 7 бит исходных данных заменяются в результате на 8 бит. Дополнительный девятый бит интерпретируется как флаг сжатия. Например:

input file: 1,2,3,4,2,2,2,2,4

output file: 1,2,3,4,2,&3,4.

Принцип: Последовательности повторяющихся значений цвета заменяются его значением и количеством повторений.

Форматы: bmp, tiff, gif

Коэффициент сжатия: 2

1.5.2.Метод Хаффмана

Этот метод назван в честь его разработчика (1950). Этот метод основан на том предположении, что некоторые значения сигнала встречаются чаще других. Если проанализировать гистограммы изображений, то можно в этом убедиться. Этот факт можно использовать для сжатия изображений. Использовать для хранения значений интенсивности, которые встречаются чаще, меньшее число бит, чем на само значение. Главная проблема в том, чтобы отделять одно значение от другого. Ведь на разные значения отводится разное количество бит. Метод сжатия Хаффмана можно проиллюстрировать следующим образом:

<i>Value</i>	<i>Probability</i>	<i>Huffman code</i>
A	.154	1
B	.110	01
C	.072	0010
D	.063	0011
E	.059	0001
F	.015	000010
G	.011	000011

original data stream: C E G A D F B E A

Huffman encoded: 0010 0001 000011 1 0011 000010 01 0001 1)

grouped into bytes: (0010 0001) (000011 1 0) (011 00001) (0 01 00 1 0 1)

Получившиеся коды уникальны, в то смысле, что могут быть записаны в поток данных без разделителей и маркеров. По количеству нулей до и после 1 программа восстановления может однозначно определить значение элемента.

Этот метод иногда используется в усложненной форме, когда кодируется не одно значение, а последовательности значений. Другая модификация метода - подвергнет коды Хаффмана групповому сжатию.

Форматы: tiff, gif

Коэффициент сжатия: 3

1.5.3.Метод разностного кодирования (дельта-кодирование)

Идея этого метода в том, чтобы сократить количество памяти необходимое для хранения ее значения путем уменьшения его величины. Например,

original data stream: 17 19 24 24 24 21 15 10 89 95 96 96 96 95 94 94 95 93 90 87 86 86

delta encoded: 17 2 5 0 0 -3 -6 -5 79 6 1 0 0 -1 -1 0 1 -2 -3 -3 -1 0

Такой метод лучше всего пригоден для гладких сигналов. Чаще используется для кодирования звуковых сигналов.

1.5.4.Метод LZW

Этот метод назван в честь его разработчиков (Lempel, Ziv, Welch). Это универсальный метод, пригодный для кодирования любых сигналов. Данный метод похож на метод Хаффмана, только для кодирования элементов используются коды равной длины и коды для часто встречающихся последовательностей элементов.

Составляется таблица всех цветов, имеющих в сжимаемом изображении. Таким образом, вместо значения цвета пиксела можно использовать индекс из таблицы. Наиболее часто встречающиеся цвета на изображении имеют меньшие индексы, а редко встречающиеся цвета размещаются в конце таблицы. Таблица цветов (палитра) размещается между заголовком и собственно изображением. Например,

<i>code number</i>	<i>translation</i>
0000	0
0001	1
0254	254
0255	255
0256	145 201 4
0257	243 245
4095	xxx xxx xxx

original data stream: 123 145 201 4 119 89 243 245 59 11 206 145 201 4 243 245

code table encoded: 123 256 119 89 257 59 11 206 256 257

Разработчики предложили не только способ хранения данных, но и хорошо документированные алгоритмы сжатия и восстановления сигнала. Метод был запатентован, стандартизован и теперь используется для сжатия любой информации.

Форматы: tiff, gif

Коэффициент сжатия: 5

1.5.5.Метод JPEG

Среди методов сжатия с потерями следует выделить семейство JPEG, разработанное организацией Joint Photographers Experts Group. Метод основан на частотных представлениях изображения и следующих предположениях. Если к сигналу применить интегральное преобразование (например, Фурье), то в результате получим, что в частотном представлении основную информацию несут низкие частоты. Высокие частоты описывают шум и несущественные детали. Удаление 50% высокочастотной информации повлечет за собой удаление 5% полезной информации содержащейся в изображении.

Допустим у нас черно-белое изображение с целочисленными значениями интенсивности от 0 до 255. JPEG-сжатие начинается с разбиения изображения на квадратные области размером 8 на 8 пикселей (64 пикселя - 64 байта). Эти области обрабатываются независимо. После преобразования в каждой группе остается от 2 до 20 байт. При восстановлении сигнала должна быть выполнена аппроксимация и восстановлена исходная область 8 на 8 пикселей.

Для сжатого представления сигнала могут использоваться различные преобразования. Наиболее адекватный (качественный) результат дает преобразование Karhunen-Loeve, но оно сложно и трудоемко в реализации. Преобразование Фурье просто, но не дает желаемого результата при восстановлении. Наиболее пригодным оказалось дискретное косинусное преобразование (Discrete Cosine Transform, DCT). При использовании DCT не нужно работать с комплексными числами (исходный сигнал и его спектр вещественные).

В результате преобразования выборки 8 на 8 получаем выборку спектра 8 на 8, низкие частоты содержатся в верхнем левом углу спектра,

а высокие в правом нижнем. Высокие частоты можно обнулить и не хранить.

Если представить спектр в виде следующей последовательности (рис. 1.5.1), то можно ее закодировать с использованием группового сжатия.

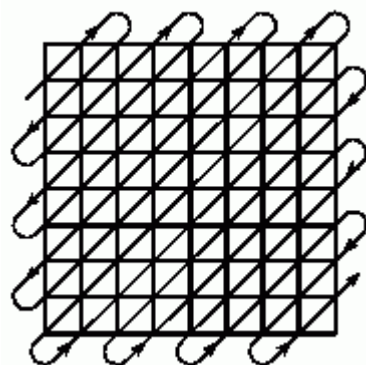


Рис.1.5.1. Представление спектра изображения

Один из способов кодирования - уменьшить количество бит, отводимое на хранение спектра.

Принцип: Используется методика сжатия с потерями. Хранится не информация о цвете пикселей, а коэффициенты разложения по некоторому базису.

Форматы: jpeg

Коэффициент сжатия: в зависимости от качества от 10 до 1000.

1.6. Форматы графических файлов

При организации хранения изображений обычно используется структура данных, состоящая из двух частей:

- **заголовочная часть**, где хранятся признак формата, размер изображения, разрешение, количество бит на пиксел, способ кодирования цвета, параметры кодирования и т.п.;
- **собственно изображение** (массив данных с информацией о цвете пикселей).

1.6.1.BMP

Формат для хранения изображений в операционной системе Windows. Это простейший формат, в котором обычно не используются методы сжатия изображений. Используется в основном для хранения картинок при организации пользовательских интерфейсов. Для хранения изображений используется редко.

1.6.2. TIFF

В этом формате для хранения используются цветовые плоскости (обычно три). Используются методы сжатия без потерь (Хафмана, LZW). В одном файле может храниться несколько изображений. Используется для хранения изображений, когда требуется высокое качество (проведение измерений на изображении и т.п.).

1.6.3. GIF

Сохраняет изображение с использованием палитры, в которой не более 256 цветов. Возможно использование прозрачного цвета. Используются методы сжатия без потерь (Хафмана, LZW). Используется как основной растровый формат для представления графической информации в интернете.

1.6.4. JPEG

Формат использует методы кодирования с потерями. Позволяет управлять степенью сжатия изображения, что в данном случае определяет качество восстанавливаемого изображения. Чаще всего используется для хранения изображений. Но если предполагается проводить измерения на изображении, то потеря точности при использовании формата с потерями неизбежно.

1.7. *Кодирование цветных изображений*

1.7.1. Определение цветовой системы

Для измерения и представления информации о цвете в первую очередь необходимо иметь представление о его фундаментальных физических и психологических свойствах. Цвет является результатом взаимодействия света, объекта и наблюдателя (или регистрирующего прибора). При взаимодействии с объектом свет модифицируется таким образом, что регистрирующий прибор (например, как система зрения человека) воспринимает модифицированный свет как определенный цвет. Чтобы цвет как таковой существовал, необходимо присутствие всех трех этих элементов. Фактически цвет – это феномен, вызываемый восприятием аппаратом зрения человека света.

Основой математического описания цвета в колориметрии является экспериментально установленный факт, что любой цвет при соблюдении определенных условий можно представить в виде смеси (суммы) определенных количеств трех линейно независимых цветов, т. е. таких цветов, каждый из которых не может быть представлен в виде суммы каких-либо количеств двух других цветов. Групп (систем) линейно

независимых цветов существует бесконечно много, но в колориметрии используются лишь некоторые из них. Три выбранных линейно независимых цвета называются *первичными* (*primary colors*). Эти цвета определяют *цветовую координатную систему* (ЦКС) или *цветовую схему* (*color scheme*) – набор первичных цветов, используемых для получения всех остальных. Тогда три числа, описывающие данный цвет, являются количествами основных цветов в смеси, цвет которой зрительно неотличим от данного цвета – *цветовая координата* данного цвета.

Будучи отнесены к стандартному наблюдателю в определенных неизменных условиях, стандартные данные смешения цветов и построенные на них колориметрической ЦКС описывают фактически лишь физический аспект цвета, не учитывая изменения цветовосприятия глаза при изменении условий наблюдения и по другим причинам.

Представление цвета с помощью цветовой координатной системы должно отражать свойства цветового зрения человека. Поэтому предполагается, что в основе всех цветовых схем лежит *физиологическая ЦКС*. Эта система определяется тремя функциями спектральной чувствительности трех различных видов приемников света (так называемых колбочек), которые имеются в сетчатке глаза человека и, согласно наиболее употребительной трехцветной теории цветового зрения, ответственны за человеческое цветовосприятие. Реакции этих трех приемников на излучение считаются цветовыми координатами в физиологической ЦКС, но функции спектральной чувствительности глаза не удастся установить прямыми измерениями. Их определяют косвенным путем и не используют непосредственно в качестве основы построения колориметрических систем.

Свойства цветового зрения учитываются в колориметрии по результатам экспериментов со смешением цветов. В таких экспериментах выполняется зрительное уравнивание чистых спектральных цветов (т. е. цветов, соответствующих монохроматическому свету с различными длинами волн) со смесями трех основных цветов. При графическом построении зависимостей количеств основных цветов от длины волны получаются функции длины волны, называемые *кривыми сложения цветов* или просто кривыми сложения.

Цветовые схемы можно разделить на две группы: схемы представления цвета от излучаемого и отраженного света. Мы видим объекты потому, что они либо излучают свет, либо светят отраженным светом. В первом случае предметы приобретают цвет испускаемого ими излучения, а во втором их цвет определяется цветом падающего на них света и цветом, который они отражают. Примером излучающего объекта

является экран монитора, а отражающего – бумага, нанесенная на нее краска.

Дополнительно цветовые модели можно классифицировать по их целевой направленности:

1. XYZ — описание восприятия; Lab — то же пространство в других координатах.
2. Аддитивные модели — рецепты получения цвета на мониторе (например, RGB).
3. Полиграфические модели — получение цвета при использовании разных систем красок и полиграфического оборудования (например, CMYK).
4. Модели, не связанные с физикой оборудования, являющиеся стандартом передачи информации.
5. Математические модели, полезные для каких-либо способов цветокоррекции, но не связанные с оборудованием, например HSV.

1.7.2. Цветовое пространство CIE XYZ

Цветовое пространство XYZ — это эталонная 3-х компонентная цветовая модель, основанная на результатах измерения характеристик человеческого глаза и заданная в строгом математическом смысле организацией CIE (*International Commission on Illumination* — *Международная комиссия по освещению*) в 1931 году. Модель XYZ является мастер-моделью практически всех остальных цветовых моделей, используемых в технических областях. Модель была построена на основе зрительных возможностей «стандартного наблюдателя», то есть гипотетического зрителя, возможности которого были тщательно изучены и зафиксированы в ходе длительных исследований человеческого зрения,

Комитет CIE провел множество экспериментов с огромным количеством людей, предлагая им сравнивать различные цвета, а затем с помощью совокупных данных этих экспериментов построил так называемые функции соответствия цветов (*color-matching functions*) и универсальное цветовое пространство (*universal color space*), в котором был представлен диапазон видимых цветов, характерный для среднестатистического человека.

Функции соответствия цветов — это значения каждой первичной составляющей света — красной, зеленой и синей, которые должны присутствовать, чтобы человек со средним зрением мог воспринимать все цвета видимого спектра. Этим трем первичным составляющим были поставлены в соответствие координаты X , Y и Z .

Необходимо заметить, что в определении кривых модели XYZ заложен фактор неопределенности: форма каждой кривой может быть измерена с достаточной точностью, однако кривая суммарной интенсивности (или сумма всех трех кривых) включает в своем определении субъективный момент, при котором реципиента просят определить, имеют ли два источника света одинаковую яркость, даже если эти источники абсолютно разного цвета. Дополнительно имеется произвольность относительной нормировки кривых X , Y и Z , поскольку можно предложить альтернативную работающую модель, в которой кривая чувствительности X имеет двукратно усиленную амплитуду. При этом цветовое пространство будет иметь иную форму. Кривые X , Y и Z в модели $CIE XYZ$ были выбраны таким образом, чтобы площади поверхности под каждой кривой были равны между собой.

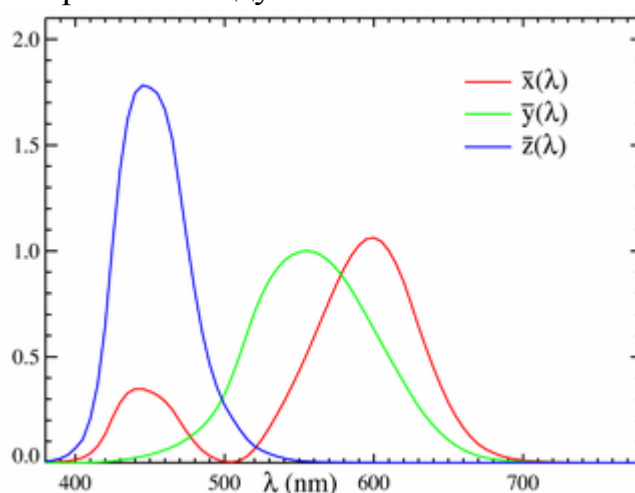


Рис.1.7.1. Функции чувствительности XYZ для стандартного наблюдателя согласно CIE 1931, в диапазоне от 380 - 780 нм

Основное свойство, присущее этой системе — положительная определенность — любой физически ощутимый цвет представляется в системе XYZ только положительными величинами. С другой стороны, не всем точкам в пространстве XYZ соответствуют реальные цвета в силу неортогональности функций соответствия цветов.

Говоря об «эталонных» оттенках, часто говорят только о паре x , y , считая, что $z = 1 - xy$.

1.7.3. Хроматические координаты $Yxу$

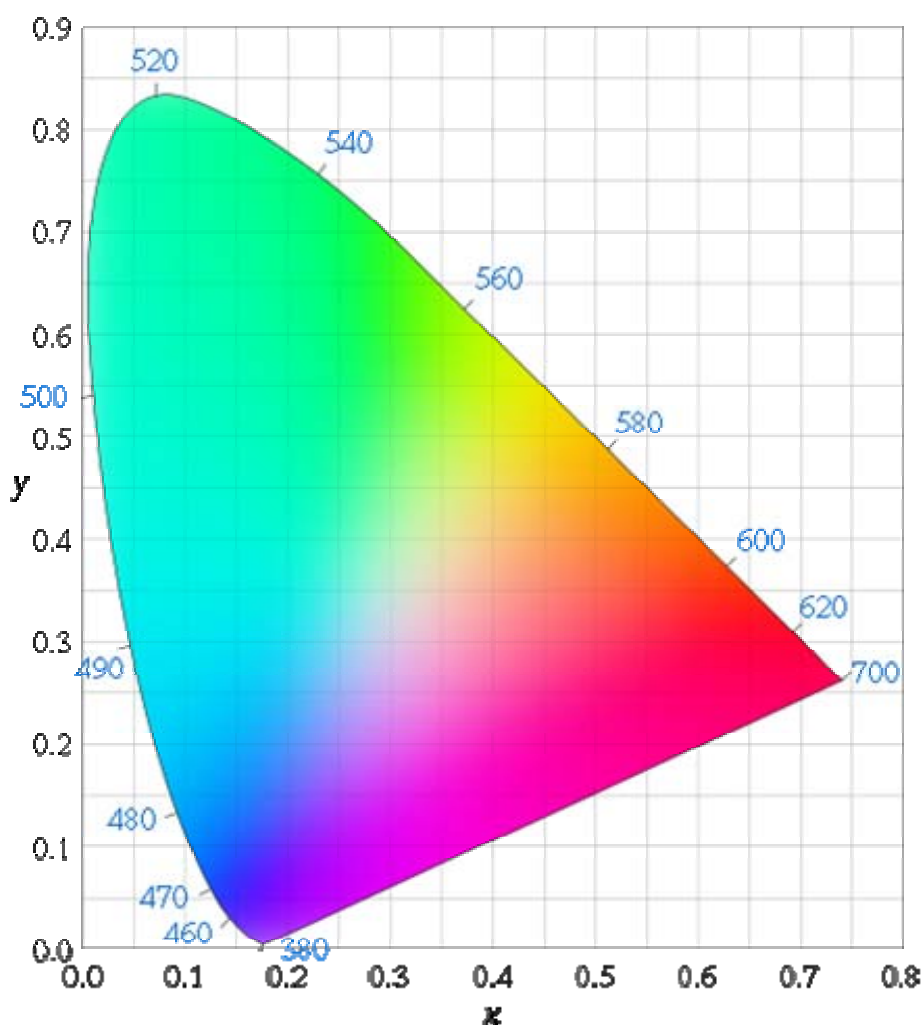


Рис.1.7.2. Хроматическая диаграмма модели XYZ

На рисунке 1.7.2. представлена классическая хроматическая диаграмма модели XYZ с длинами волн цветов. Значения x и y в ней соответствуют X , Y и Z согласно следующим формулам:

$$x = X / (X + Y + Z), \quad (1.7.1)$$

$$y = Y / (X + Y + Z). \quad (1.7.2)$$

В математическом смысле, на данной хроматической диаграмме x и y — это координаты на плоскости проекции. Данное представление позволяет задавать значение цвета через «светлоту» или «яркость» Y (англ. *luminance*) и две координаты x , y .

Обычно диаграмма $Yxу$ используется для иллюстрации характеристик локусов различных устройств воспроизведения цвета — дисплеев и принтеров. Конкретный locus обычно имеет вид треугольника, углы которого образованы точками основных, или первичных, цветов.

Внутренняя область локуса описывает все цвета, которые способно воспроизвести данное устройство.

1.7.4.Lab

Чтобы устранить нелинейность системы XYZ в 1976 году была разработана модель CIE Lab, которая является сейчас международным стандартом. Указанная модель считается аппаратно-независимой и определяет цвета без учета особенностей устройств: монитора, принтера, сканера, печатного станка и т.д.

В цветовом пространстве Lab значение яркости отделено от значения хроматической составляющей цвета (*тон, насыщенность*). Яркость задана координатой L (изменяется от 0 до 100, то есть от самого темного до самого светлого), хроматическая составляющая — двумя декартовыми координатами a и b . Первая обозначает положение цвета в диапазоне от зеленого до пурпурного, вторая — от синего до желтого.

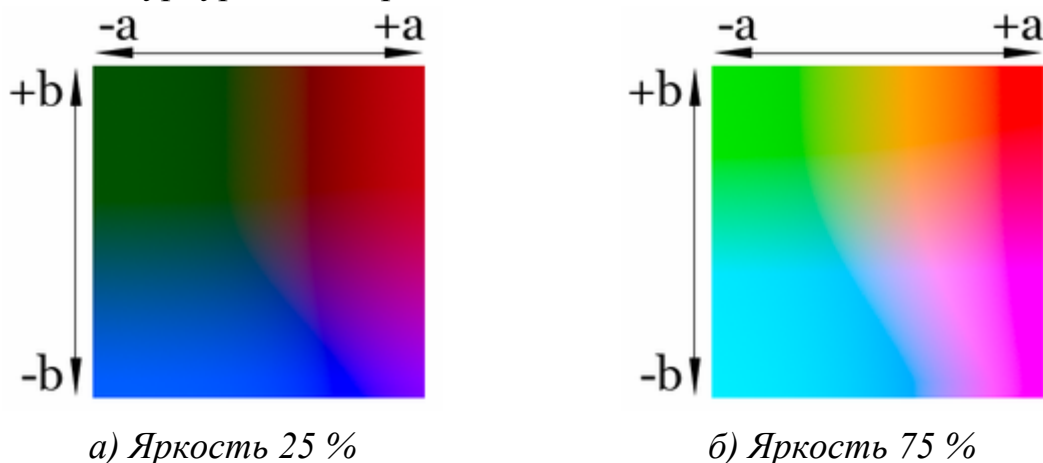


Рис.1.7. 3 Цветовая система Lab

Т.к. система Lab однозначно определяет цвет, она нашла широкое применение в программном обеспечении для обработки изображений в качестве промежуточного цветового пространства, через которое происходит конвертирование данных между другими цветовыми пространствами. При этом особые свойства Lab сделали редактирование в этом пространстве мощным инструментом цветокоррекции.

Благодаря характеру определения цвета в Lab появляется возможность отдельно воздействовать на яркость, контраст изображения и на его цвет. Во многих случаях это позволяет ускорить обработку изображений, например, при допечатной подготовке. Lab предоставляет возможность избирательного воздействия на отдельные цвета в изображении, усиления цветового контраста, незаменимыми являются и возможности, которые это цветовое пространство предоставляет для борьбы с шумом на цифровых фотографиях.

Однако, система Lab при разработке была оптимизирована для применения в условиях известного (как правило студийного) освещения, и из-за этого трудно применима для сцен в которых присутствуют несколько источников света, ни один из которых не может считаться основным.

Цветовое пространство Lab модели может быть условно представлено в виде схемы, все цвета которой расположены внутри и на границе «подковы» и физически реализуемы.

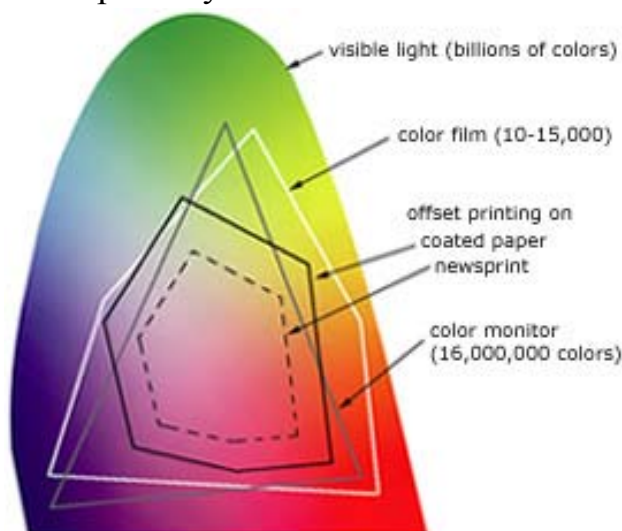


Рис.1.7.4. Представление модели Lab

Схему можно использовать для определения цветового охвата любого устройства, если известны параметры основных цветов, которые применяются для синтеза результирующего цвета. Для этого достаточно на схеме определить точки основных цветов и соединить их прямыми линиями.

$$L \text{ (Hunter Lightness)} = 100 \left(\frac{Y}{Y_0} \right)^{\frac{1}{2}},$$

$$a \text{ (Hunter Red - Green)} = 175 \left(\frac{0.0102X_0}{Y/Y_0} \right)^{\frac{1}{2}} \left(\frac{X}{X_0} - \frac{Y}{Y_0} \right), \quad (1.7.3)$$

$$b \text{ (Hunter Yellow - Blue)} = 70 \left(\frac{0.00847Z_0}{Y/Y_0} \right)^{\frac{1}{2}} \left(\frac{Y}{Y_0} - \frac{Z}{Z_0} \right).$$

1.7.5. RGB.

Фактически основой всех цветовых схем является система, кривые сложения которой были определены экспериментально. Ее основными

цветами являются чистые спектральные цвета, соответствующие монохроматическим излучениям с длинами волн 700,0 нм (красный), 546,1 нм (зеленый) и 435,8 нм (синий). Эта система, принятая Международной комиссией по освещению (МКО) в 1931, получила название международной колориметрической системы *МКО RGB* или просто *RGB* (от англ. red – красный, green – зеленый, blue – синий).



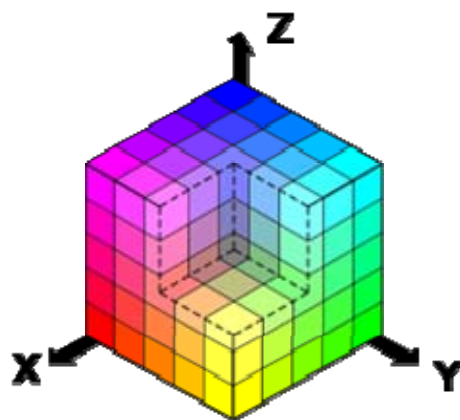
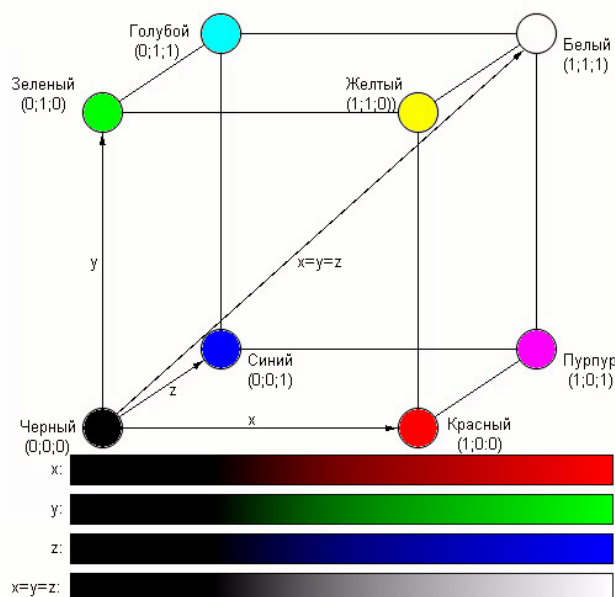
Рис.1.7.5 Аддитивное смешение цветов

Система RGB является *аддитивной* (от англ. *add* – добавлять, складывать). В таких системах цвет получается путем сложения первичных цветов (рис.1.7.5). При этом отсутствие всех цветов представляет собой черный цвет, а присутствие всех цветов – белый. Система аддитивных цветов работает с излучаемым светом, например, от монитора компьютера.

Выбор основных цветов обусловлен особенностями физиологии восприятия цвета сетчаткой человеческого глаза.

Чаще всего данную модель представляют в виде единичного куба (рис.1.7.6) с оортами: (1;0;0)- красный, (0;1;0)- зеленый, (0;0;1)- синий и началом (0;0;0)- черный. На рисунке показан куб и распределение цветов вдоль указанных векторов.

В компьютерах для представления каждой из координат традиционно используется один октет, значения которого обозначаются для удобства целыми числами от 0 до 255 включительно. Следует учитывать, что чаще всего используется гамма-компенсированное цветовое пространство *sRGB*, обычно с показателем 1.8 (Mac) или 2.2 (PC).



а)

б)

Рис.1.7.6. Представление системы RGB

В HTML используется *#RrGgBb*-запись, называемая шестнадцатеричной: каждая координата записывается в виде двух шестнадцатеричных цифр, без пробелов. Например, *#RrGgBb*-запись белого цвета — *#FFFFFF*.

COLORREF — стандартный тип для представления цветов в Win32. Использует для определения цвета в *RGB* виде. Размер — 4 байта. При определении какого-либо *RGB* цвета, значение переменной типа *COLORREF* можно представить в шестнадцатеричном виде:

$$0x00bbggrr, \quad (1.7.4)$$

где *rr, gg, bb* — значение интенсивности соответственно красной, зеленой и синей составляющих цвета. Максимальное их значение — *0xFF*.

Определить переменную типа *COLORREF* можно следующим образом:

$$COLORREF C = (r, g, b);$$

где *b, g* и *r* — интенсивность (в диапазоне от 0 до 255) соответственно синей, зеленой и красной составляющих определяемого цвета *C*. То есть, ярко-синий цвет может быть определен как (0,0,255), красный как (255,0,0), ярко-фиолетовый — (255,0,255), черный — (0,0,0), а белый — (255,255,255).

При всех достоинствах цветовой системы RGB она имеет один существенный недостаток: ограничение по возможности передачи цветов (рис.1.7.7).

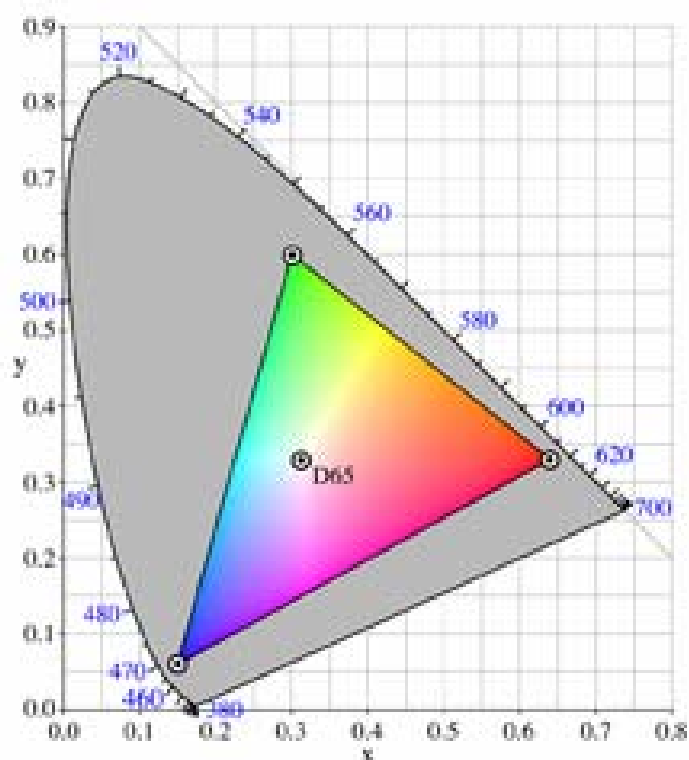


Рис.1.7.7. Ограничение RGB по возможности передачи цветов

1.7.6. Система CMYK

Окрашенные несветящиеся объекты поглощают часть спектра белого света, освещающего их, и отражают оставшееся излучение. В зависимости от того, в какой области спектра происходит поглощение, объекты отражают разные цвета (окрашены в них). Цвета, которые используют белый свет, вычитая из него определенные участки спектра, называются *субтрактивными* («вычитательными»). Для их описания используется субтрактивная модель *CMY* (Cyan, Magenta, Yellow) (рис.1.7.8). В этой модели основные цвета образуются путем вычитания из белого цвета основных аддитивных цветов модели RGB. Понятно, что в таком случае и основных субтрактивных цветов будет три: голубой (белый минус красный), пурпурный (белый минус зеленый), желтый (белый минус синий).

Система цветов *CMY* была широко известна задолго до того, как компьютеры стали использоваться для создания графических изображений. Ее основные цвета: голубой, пурпурный и желтый является, по сути, наследниками трех основных цветов живописи (синего, красного и желтого). Изменение оттенка первых двух связано с отличием химического состава художественных красок от печатных.



Рис.1.7.8. Субтрактивное смешение цветов

Как художественные, так и печатные краски не могут дать очень многих оттенков. Черная краска внесена в число основных полиграфических красок и в модель для улучшения качества отпечатка. Именно она добавила последнюю букву в название модели *СМУК* (черный компонент сокращается до буквы *К*, поскольку эта краска является ключевой (Key) в процессе цветной печати). *СМУК* – основная модель полиграфии и используется при выводе графической информации на печать.

В качестве основных причин введения четвертого компонента можно назвать следующие:

1. На практике в силу неидеальности красителей и погрешностей в пропорциях компонентов смешение реальных пурпурного, голубого и желтого цветов дает скорее грязно-коричневый или грязно-серый цвет; триадные краски не дают той глубины и насыщенности, которая достигается использованием настоящего черного. Чистота и насыщенность черного цвета и стабильность оттенка нейтральных (серых) областей чрезвычайно важны в печатном процессе, поэтому и был введен еще один цвет.
2. При выводе мелких черных деталей изображения или текста без использования черного пигмента возрастает риск «неприводки» (недостаточно точное совпадение точек нанесения) пурпурного, голубого и желтого цветов. Увеличение же точности печатающего аппарата требует неадекватных затрат.
3. Смешение 100 % пурпурного, голубого и желтого пигментов в одной точке в случае струйной печати существенно смачивает бумагу, деформирует ее и увеличивает время просушки. Похожие проблемы с «суммой» красок возникают и в офсетной печати. В зависимости от типа материала и технологии печати ограничение по сумме красок может быть ниже 300% (например, в газетной

печати типичное ограничение 260-280%), что делает технически невозможным синтез насыщенного черного из трех стопроцентных компонентов триады.

4. Черный пигмент (в качестве которого, как правило, используется сажа) существенно дешевле остальных трех.

Связь между *RGB* и *CMY* можно выразить через следующую формулу:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \quad (1.7.5)$$

Недостатком системы *CMYK*, как и системы *RGB*, является ограничение по возможности передачи цветов (рис. 1.7.9).

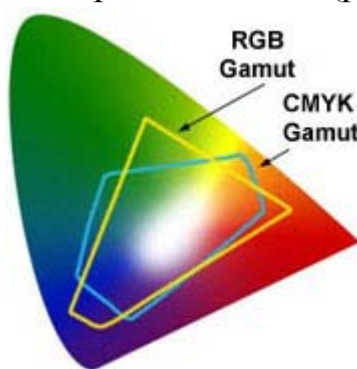


Рис.1.7.9. Цветовой охват *RGB* и *CMYK* цветовых моделей внутри модели *Lab*

1.7.7. Система HSB

Системы цветов *RGB* и *CMYK* базируются на ограничениях, накладываемых аппаратным обеспечением (в случае *RGB* это мониторы, сканеры и т.п., в случае *CMYK* это типографские краски). Более интуитивным способом описания цвета является представление его в виде тона или оттенка (*Hue*), насыщенности (*Saturation*) и яркости (*Brightness* или *Volume*) – система HSB или HSV. Ее вариациями являются система HSL, где используются тон (*Hue*), насыщенность (*Saturation*) и освещенность (*Lightness*) и система HSI – тон (*Hue*), насыщенность (*Saturation*) и интенсивность (*Intensity*).

Тон представляет собой конкретный оттенок цвета, отличный от других: красный, зеленый, голубой и т. п. Варьируется в пределах 0—360°, однако иногда приводится к диапазону 0—100 или 0—1.

Насыщенность цвета характеризует его относительную интенсивность (или чистоту). Варьируется в пределах 0—100 или 0—1. Чем больше этот параметр, тем «чище» цвет, поэтому этот параметр иногда называют

чистотой цвета. А чем ближе этот параметр к нулю, тем ближе цвет к нейтральному серому, т.е. уменьшая насыщенность, например, красного, мы делаем его более пастельным, приближаем к серому.

Яркость (освещенность или интенсивность) цвета показывает величину черного оттенка, добавленного к цвету, что делает его более темным, задается в пределах 0—100 и 0—1.

Система *HSB* имеет перед другими системами важное преимущество: она больше соответствует природе цвета, хорошо согласуется с моделью восприятия цвета человеком. Многие оттенки можно быстро и удобно получить в *HSB*, конвертировав затем в *RGB* или *CMYK*, доработав в последнем случае, если цвет был искажен. Поэтому система *HSB* часто используется при выборе пользователем цвета.

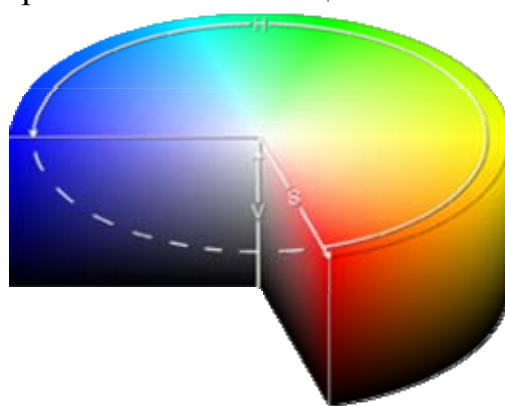
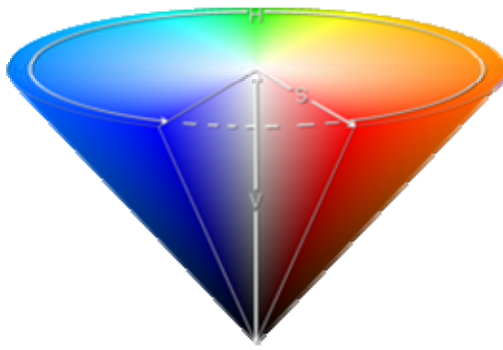
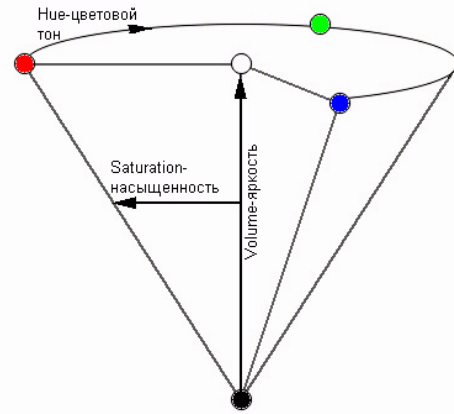


Рис.1.7.10. Цилиндрическое представление системы *HSV*.

Простейший способ отобразить *HSV* в трехмерное пространство (рис.1.7.10) — воспользоваться цилиндрической системой координат. Здесь координата *H* определяется полярным углом, *S* — радиус-вектором, а *V* — *Z*-координатой. То есть, оттенок изменяется при движении вдоль окружности цилиндра, насыщенность — вдоль радиуса, а яркость — вдоль высоты. Несмотря на «математическую» точность, у такой модели есть существенный недостаток: на практике количество различимых глазом уровней насыщенности и оттенков уменьшается при приближении яркости (*V*) к нулю (то есть, на оттенках, близких к черному). К тому же на малых *S* и *V* появляются существенные ошибки округления при переводе *RGB* в *HSV* и наоборот. Поэтому чаще применяется коническая модель (рис.1.7.11).



а)



б)

Рис.1.7.11. Конусное представление системы HSV

Как и в цилиндре, оттенок изменяется по окружности конуса. Насыщенность цвета возрастает с отдалением от оси конуса, а яркость — с приближением к его основанию. Иногда вместо конуса используется шестиугольная правильная пирамида.

Для перехода между цветовыми системами существуют следующие выражения:

HSV в RGB:

$$H = \frac{3H}{\pi}; F = H - \text{int}(H)$$

$$M = V(1 - S); I = \text{int}(H)$$

$$N = V(1 - SF)$$

$$K = V(1 - s(1 - F))$$

$$(R, G, B) = \begin{cases} (V, K, M) & \text{при } I = 1 \\ (N, V, M) & \text{при } I = 2 \\ (M, V, K) & \text{при } I = 3 \\ (M, N, V) & \text{при } I = 4 \\ (K, M, V) & \text{при } I = 5 \\ (V, M, N) & \text{при } I = 6 \end{cases} \quad (1.7.6)$$

RGB в HSV:

$$V = \max(R, G, B); v = \min(R, G, B)$$

$$S = \begin{cases} 0; \text{если } V = 0 \\ (V - v) / V; \end{cases}$$

$$Cr = (V - R) / (V - v); Cg = (V - G) / (V - v); Cb = (V - B) / (V - v)$$

$$H = \frac{\pi}{3} \cdot \begin{cases} Cb - Cg; \text{если } R = V \\ 2 + Cr - Cb; \text{если } G = V \\ 4 + Cg - Cr; \text{если } B = V \end{cases} \quad (1.7.8)$$

1.7.8. Особенности использования цветовых моделей

При цветовых преобразованиях необходимо помнить, что между цветовыми моделями *CIE*, *CMY*, *RGB*, *YIQ* существуют аффинные преобразования, тогда, как между *HLS* и *HSV*- нет. Данное обстоятельство будет заметно, если изображение, содержащее непрерывные цветовые переходы, переводить, например, из *HLS* в *RGB* (на изображениях может появиться разрыв непрерывности).

1.8. Форматы хранения изображений

1.8.1. Черно-белый

Каждый элемент матрицы представлен одним битом. Если он равен единице, то он отождествляется с черным цветом, если равен нулю - с белым. Это самый простой формат, он применяется при печати газет, распознавании текстов и подписей.

1.8.2. Grayscale(градации серого)

Отличие данного формата от предыдущего в том, что для каждого элемента матрицы отводится 8 битов (байт). Это позволит нам использовать $2^8 = 256$ уровней серого цвета. Если $a(i, j) = 0$, то имеем белый цвет, с возрастанием до 255 мы будем терять яркость и при $a(i, j) = 255$ получим черный цвет. В промежутке от 0 до 255 будут располагаться серые цвета по правилу: чем ближе значение к 255, тем чернее будет серый. Данный формат позволяет получать довольно качественные черно-белые изображения. Значения $a(i, j)$ содержат обратную яркость, т.е. значение $(1-L) \cdot 255$, где L - яркость, которая может быть получена, например, из *RGB* цветовых изображений по формуле:

$$L = aR + bG + cB, \quad (1.8.1)$$

где R, G, B лежат в интервале $[0;1]$, а веса a, b, c в сумме дают единицу.

Иногда, для хранения grayscale изображений используют на точку 4-7 и 16 битов. В таком случае мы имеем 16–128 или 65536 оттенков серого цвета.

1.8.3. Многоканальные

В данном случае $a(i, j)$ представлен в виде вектора с координатами используемой цветовой модели. Обычно вектор трехмерный, так как природа глаза реагирует на три различных цветовых составляющих. Каждый компонент вектора чаще всего занимает байт. Рассмотрим наиболее распространенные многоканальные форматы:

Название	Соотношение бит	1-ый компонент	2-ой компонент	3-ий компонент
RGB - Truecolor	8:8:8	Красный 0..255	Зеленый 0..255	Синий 0..255
RGB - Highcolor	5:6:5/5:5:5	Красный 0..31	Зеленый 0.63..31	Синий 0..31
RGB - Extended	12:12:12/ 16:16:16	Красный 0..4095 или 0..65535	Зеленый 0..4095 или 0..65535	Синий 0..4095 или 0..65535
CMY	8:8:8	Голубой 0..255	Пурпур 0..255	Желтый 0..255
LAB	8:8:8	Яркость 0..255	Канал А 0..100%	Канал В 0..100%
YIQ	8:8:8	Яркость 0..255	Синфазный 0..255	Интегрированный 0..255
HLS	8:8:8	Тон 0..360°	Яркость 0..100%	Насыщенность 0..100%
HSB	8:8:8	Тон 0..360°	Насыщенность 0..100%	Яркость 0..100%

Встречаются четырех и более мерные вектора, например, модель *СМУК*, она применяется, когда имеются четыре основных цветовых красителя. Двумерные модели называют дуплексами. Их применяют в полиграфии, например, при печати стандартного *grayscale* изображения, реально в промышленности оно будет выполнено лишь в ~ 50 градациях серого, и для повышения числа градаций вводят вторую краску.

1.8.4. Индексированный

Для уменьшения объемов изображения или для использования определенных цветов используют данный формат. Элемент матрицы $a(i, j)$ является указателем на таблицу цветов. Число используемых цветов равно $2K$, где K - количество бит, используемый для хранения элемента матрицы. Цвета в указываемой таблице могут кодироваться другим числом бит. Например, в 256 цветовых режимах видеоадаптеров выбирается 256 цветов из 262144 возможных, так как выбираемые цвета представляются в *RGB* формате и для каждой цветовой компоненты кодируется 256-ю битами. Существует много методов преобразования многоканальных изображений в индексированные (Error diffusion, ближайшего цвета и др.).

1.9. Улучшение визуального качества изображений путем поэлементного преобразования

В большом числе информационных систем применяется представление результатов обработки данных в виде изображения, выводимого на экран для использования наблюдателем. Процедуру, обеспечивающую такое представление, называют *визуализацией*. Желательно при помощи обработки придать выводимому изображению такие качества, благодаря которым его восприятие человеком было бы по возможности комфортным. Часто бывает полезным подчеркнуть, усилить какие-то черты, особенности, нюансы наблюдаемой картины с целью улучшения ее субъективного восприятия.

Последнее - субъективность восприятия - сильно усложняет применение формализованного подхода в достижении данных целей. Поэтому при обработке изображений для визуализации получили распространение методы, в которых часто отсутствуют строгие математические критерии оптимальности. Их заменяют качественные представления о целесообразности той или иной обработки, опирающиеся на субъективные оценки результатов.

Подавляющее большинство процедур обработки для получения результата в каждой точке кадра привлекает входные данные из некоторого множества точек исходного изображения, окружающих обрабатываемую точку. Однако имеется группа процедур, где

осуществляется так называемая *поэлементная* обработка. Здесь результат обработки в любой точке кадра зависит только от значения входного изображения в этой же точке. Очевидным достоинством таких процедур является их предельная простота. Вместе с тем, многие из них приводят к очевидному субъективному улучшению визуального качества. Этим определяется внимание, которое уделяется поэлементным процедурам. Не преувеличивая их роли, отметим, что очень часто поэлементная обработка применяется как заключительный этап при решении более сложной задачи обработки изображения.

Сущность поэлементной обработки изображений сводится к следующему. Пусть $x(i, j) = x_{i,j}$, $y(i, j) = y_{i,j}$ - значения яркости исходного и получаемого после обработки изображений соответственно в точке кадра, имеющей декартовы координаты i (номер строки) и j (номер столбца). Поэлементная обработка означает, что существует функциональная однозначная зависимость между этими яркостями:

$$y_{i,j} = f_{i,j}(x_{i,j}), \quad (1.9.1)$$

позволяющая по значению исходного сигнала определить значение выходного продукта. В общем случае, как это учтено в данном выражении, вид или параметры функции $f_{i,j}(x_{i,j})$, описывающей обработку, зависят от текущих координат. При этом обработка является неоднородной. Однако в большинстве практически применяемых процедур используется однородная поэлементная обработка. В этом случае индексы i и j в выражении (1.9.1) могут отсутствовать. При этом зависимость между яркостями исходного и обработанного изображений описывается функцией:

$$y = f(x), \quad (1.9.2)$$

одинаковой для всех точек кадра.

1.9.1. Линейное контрастирование изображения

Задача *контрастирования* связана с улучшением согласования динамического диапазона изображения и экрана, на котором выполняется визуализация. Если для цифрового представления каждого отсчета

изображения отводится 1 байт (8 бит) запоминающего устройства, то входной или выходной сигналы могут принимать одно из 256 значений. Обычно в качестве рабочего используется диапазон 0...255; при этом значение 0 соответствует при визуализации уровню черного, а значение 255 - уровню белого. Предположим, что минимальная и максимальная яркости исходного изображения равны $x_{\min}$ и $x_{\max}$ соответственно. Если эти параметры или один из них существенно отличаются от граничных значений яркостного диапазона, то визуализированная картина выглядит как ненасыщенная, неудобная, утомляющая при наблюдении. Пример такого неудачного представления приведен на рис. 1.9.1.(а), где диапазон яркостей имеет границы $x_{\min} = 180$, $x_{\max} = 240$.

При линейном контрастировании используется линейное поэлементное преобразование вида:

$$y = a \cdot x + b, \quad (1.9.3)$$

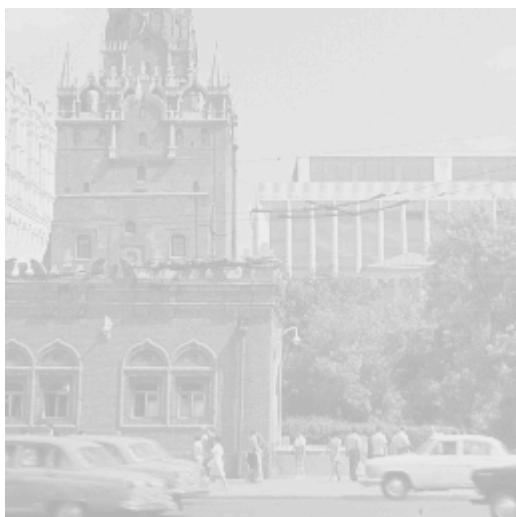
параметры которого a и b определяются желаемыми значениями минимальной $y_{\min}$ и максимальной $y_{\max}$ выходной яркости. Решив систему уравнений:

$$\begin{cases} y_{\min} = a \cdot x_{\min} + b \\ y_{\max} = a \cdot x_{\max} + b \end{cases} \quad (1.9.4)$$

относительно параметров преобразования a и b , получаем:

$$y = \frac{x - x_{\min}}{x_{\max} - x_{\min}} (y_{\max} - y_{\min}) + y_{\min}. \quad (1.9.5)$$

Результат линейного контрастирования исходного изображения, представленного на рис. 1.9.1.(а), приведен на рис.1.9.1.(б) при $y_{\min} = 0$ и $y_{\max} = 255$.



а)



б)

Рис 1.9.1 Пример линейного контрастирования

Сравнение двух изображений свидетельствует о значительно лучшем визуальном качестве обработанного изображения. Улучшение связано с использованием после контрастирования полного динамического диапазона экрана, что отсутствует у исходного изображения.

1.9.2. Соляризация изображения

При данном виде обработки преобразование (1.9.2) имеет вид:

$$y = k \cdot x \cdot (x_{\max} - x), \quad (1.9.6)$$

где $x_{\max}$ - максимальное значение исходного сигнала, а k - константа, позволяющая управлять динамическим диапазоном преобразованного изображения.

Функция, описывающая данное преобразование, является квадратичной параболой, ее график при $k = 1$ приведен на рис.1.9.2. При $y_{\max} = x_{\max}$ динамические диапазоны изображений совпадают, что может быть достигнуто при $k = 4/x_{\max}$.

Другой путь нормализации динамического диапазона может состоять в применении одного из методов контрастирования, например, описанного выше линейного контрастирования.

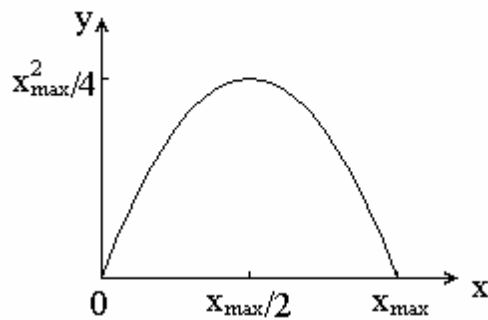


Рис. 1.9.2. Функция, описывающая соляризацию

Как следует из рис.1.9.2, смысл соляризации заключается в том, что участки исходного изображения, имеющие уровень белого или близкий к нему уровень яркости, после обработки имеют уровень черного. При этом сохраняют уровень черного и участки, имеющие его на исходном изображении. Уровень же белого на выходе приобретают участки, имеющие на входе средний уровень яркости (уровень серого). Пример применения соляризации приведен на рис.1.9.3.



а)

б)

Рис. 1.9.3. Пример соляризации

На рис.1.9.3(а) показано исходное изображение, а на рис.1.9.3(б) - результат его соляризации. На втором этапе обработки здесь применено линейное контрастирование при $y_{\min} = 0$ и $y_{\max} = 255$. Обработка приводит к повышению четкости деталей изображения: улучшены изображения глаз, повышен контраст на переходе «лицо – волосы» и т. д..

1.9.3.Препарирование изображения

Препарирование представляет собой целый класс поэлементных преобразований изображений. Характеристики применяемых на практике процедур препарирования приведены на рис.1.9.4. Остановимся на описании некоторых из них.

Преобразование с пороговой характеристикой (рис.1.9.4.(а)) превращает полутоновое изображение, содержащее все уровни яркости, в бинарное, точки которого имеют яркости $y = 0$ или $y = y_{\max}$. Такая операция, называемая иногда *бинаризацией* или *бинарным квантованием*, может быть полезной, когда для наблюдателя важны очертания объектов, присутствующих на изображении, а детали, содержащиеся внутри объектов или внутри фона, не представляют интереса.

Основной проблемой при проведении такой обработки является определение порога x_0 , сравнение с которым яркости исходного изображения позволяет определить значение выходного изображения в каждой его точке.

Замена исходного полутонового изображения бинарным препаратом решает две основные задачи: достигается большая наглядность при визуальном восприятии, чем у исходного изображения, и ощутимо сокращается объем запоминающего устройства для хранения изображения, поскольку бинарный препарат для записи каждой точки бинарного изображения требует лишь 1 бит памяти, в то время как полутоновое изображение для решения той же задачи при наиболее часто применяемом формате представления - 8 бит. Пример бинаризации изображения отпечатка пальца приведен на рис.1.9.5(б).

Смысл других преобразований, представленных на рис. 1.9.4, нетрудно понять, рассматривая их характеристики. Например, преобразование рис. 1.9.4(б) выполняет яркостный срез изображения, выделяя те его участки, где яркость соответствует выделенному интервалу. При этом остальные участки оказываются полностью «погашенными» (имеют яркость, соответствующую уровню черного). Перемещая выделенный интервал по яркостной шкале и изменяя его ширину, можно детально исследовать содержание картины.

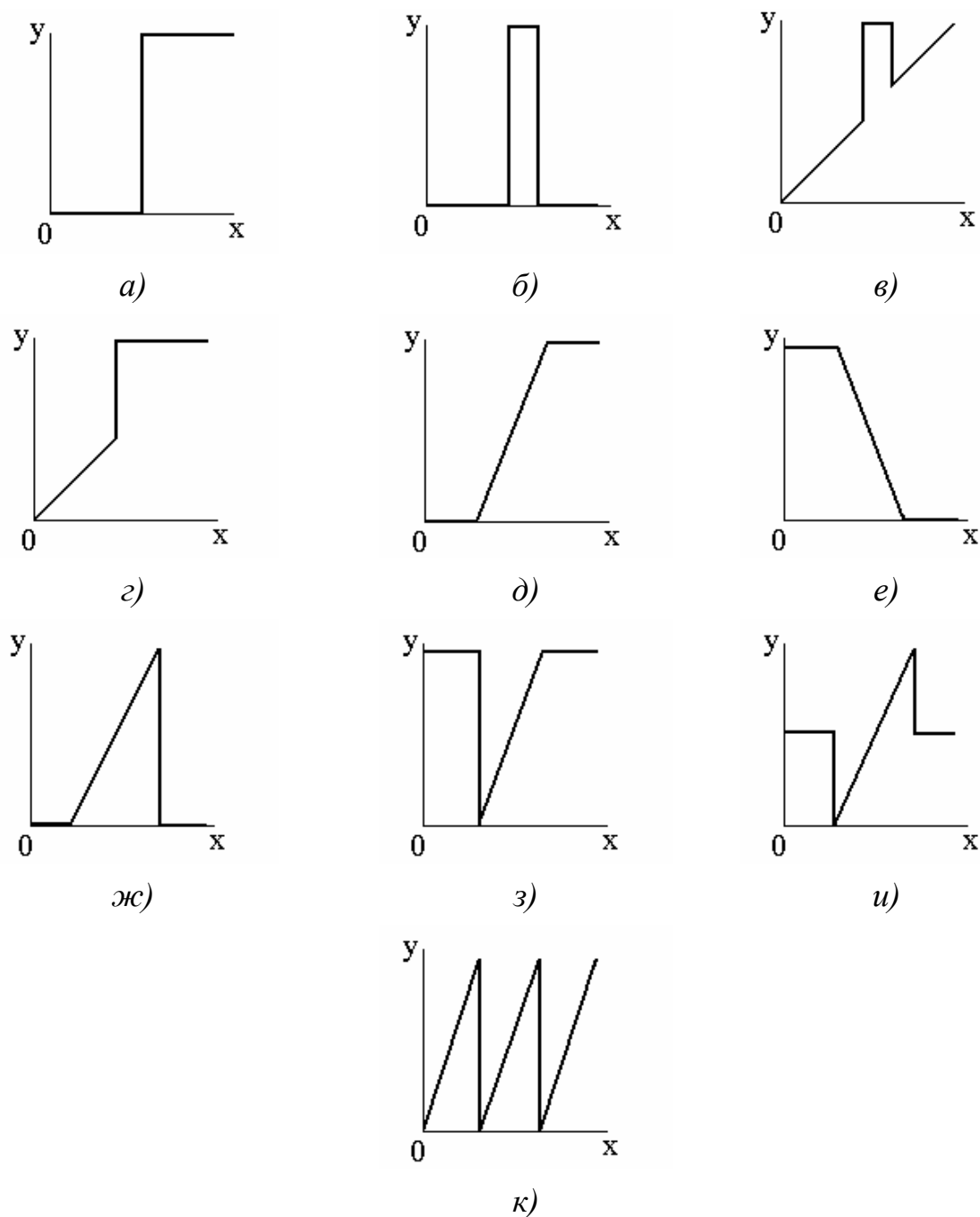


Рис.1.9.4.Примеры преобразований, используемых при препарировании

Преобразование, приведенное на рис. 1.9.4(ж), позволяет повысить детальность наблюдаемой картины в выбранном диапазоне яркостей, однако в отличие от предыдущего здесь выходное изображение использует полный динамический диапазон. По существу, это преобразование представляет собой линейное контрастирование, применяемое к избранному диапазону входного изображения. Как и в предыдущем варианте, участки, не попавшие в этот диапазон, образуют после препарирования черный фон.

Иногда наглядность изображения повышается применением преобразования типа пилообразного контрастирования (рис.1.9.4(к)). При этом различные яркостные диапазоны одновременно подвергаются локальному яркостному контрастированию. Однако необходимо иметь в виду, что данное преобразование, как и некоторые другие, может сопровождаться появлением ложных контуров на получаемом препарате.

Аналогично можно качественно рассмотреть и остальные процедуры препарирования, представленные на рис. 1.9.4.

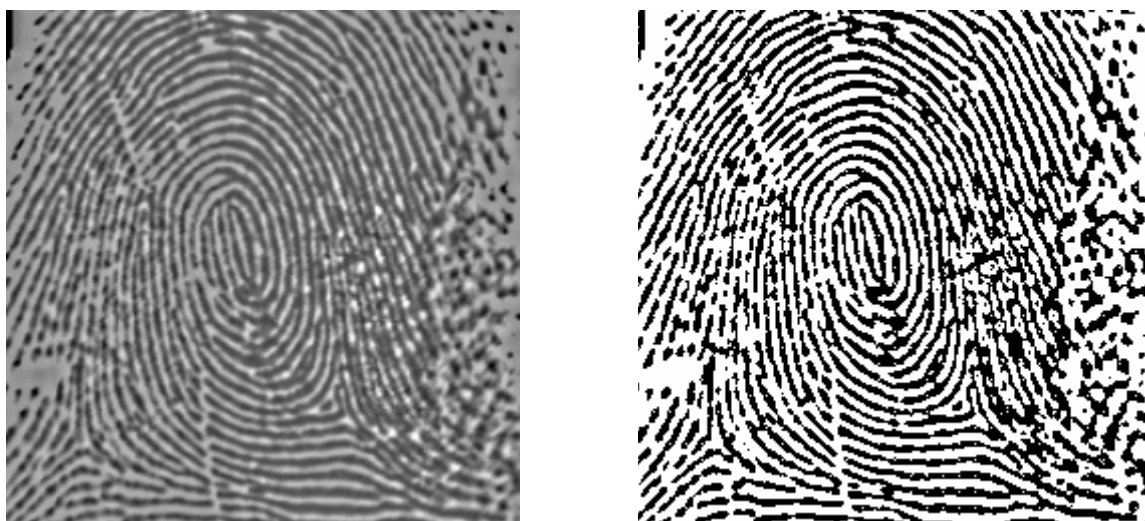


Рис. 1.9.5. Пример бинаризации изображения

На рис.1.9.6 приведены результаты эксперимента, в котором к аэроснимку участка земли (рис. 1.9.6(а)) применялись преобразования типа пороговая обработка (рис. 1.9.6(б)) и пилообразное контрастирование (рис. 1.9.6(в)). Первое приводит к выявлению границ отдельных участков, создавая общее интегральное представление о наблюдаемой сцене. Второе, наоборот, дает возможность наблюдения мелких деталей на всех участках изображения. Сочетание двух таких возможностей может оказаться полезным наблюдателю.

Препарирование часто используется и в автоматических системах обработки визуальной информации, поскольку подготавливаемый при этом препарат может содержать всю информацию, необходимую для последующей (вторичной) обработки. Например, если при наблюдении из космоса требуется автоматически обнаружить на изображении некоторый

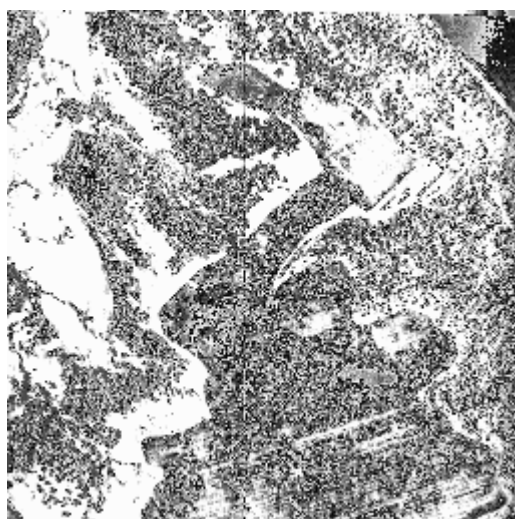
объект, имеющий известную конфигурацию, то для этого может быть достаточно бинарного препарата, передающего эту конфигурацию.



а)



б)



в)

Рис. 1.9.6. Примеры препарирования изображения

1.9.4. Преобразование гистограмм, эквализация

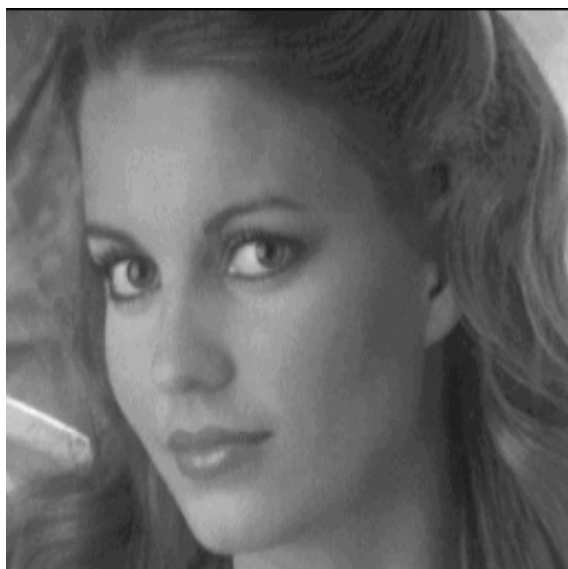
В практике обработки изображений преобразование распределений выполняют в два этапа.

На первом этапе измеряется *гистограмма* исходного изображения. Для цифрового изображения, шкала яркостей которого, например, принадлежит целочисленному диапазону $0 \dots 255$, гистограмма представляет собой таблицу из 256 чисел. Каждое из них показывает количество точек в кадре, имеющих данную яркость. Разделив все числа

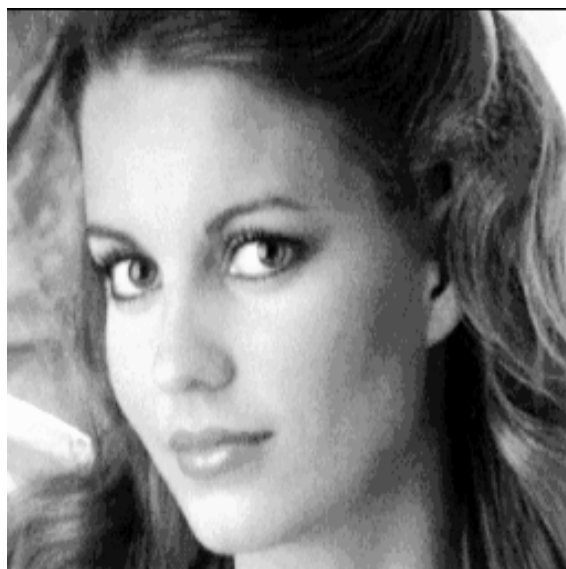
этой таблицы на общий размер выборки, равный числу используемых точек изображения, получают оценку распределения вероятностей яркости изображения.

На втором этапе выполняется нелинейное преобразование, обеспечивающее необходимые свойства выходного изображения. При этом вместо неизвестного истинного интегрального распределения используется его оценка, основанная на гистограмме. С учетом этого все методы поэлементного преобразования изображений, целью которых является видоизменение законов распределения, получили название *гистограммных методов*. В частности, преобразование, при котором выходное изображение имеет равномерное распределение, называется *эквализацией* (выравниванием) гистограмм.

Процедуры преобразования гистограмм могут применяться как к изображению в целом, так и к отдельным его фрагментам, что может быть полезным при обработке нестационарных изображений, содержание которых существенно различается по своим характеристикам на различных участках. В этом случае лучшего эффекта можно добиться, применяя гистограммную обработку к отдельным участкам.



а) исходное изображение



б) результат обработки

Рис. 1.9.7. Пример эквализации изображения

Необходимо иметь в виду, что в результате обработки не удастся получить идеальное распределение вероятностей выходного изображения, поэтому полезно проводить контроль его гистограммы.

На рис. 1.9.7. приведен пример эквализации, выполненной в соответствии с изложенной методикой. Характерной чертой многих изображений, получаемых в реальных изображающих системах, является значительный удельный вес темных участков и сравнительно малое число участков с высокой яркостью.

Эквализация призвана откорректировать картину, выровняв интегральные площади участков с различными яркостями.

Сравнение исходного (рис. 1.9.7(а)) и обработанного (рис. 1.9.7.(б)) изображений показывает, что происходящее при обработке перераспределение яркостей приводит к улучшению визуального восприятия.

2. МЕТОДЫ И АЛГОРИТМЫ ОБРАБОТКИ ОПТИЧЕСКОГО ИЗОБРАЖЕНИЯ

2.1. Геометрические операции над изображением

При рассмотрении геометрических преобразований плоских изображений (то есть относящихся к двумерному случаю - 2D), предполагается, что изображение обрабатывается в *евклидовом* пространстве, где имеется *ортонормированная декартова* система координат, в которой координатные оси взаимно ортогональны, а соответствующие им единичные отрезки имеют одинаковую длину. Тогда каждой точке изображения ставится в соответствие упорядоченная пара чисел (x, y) декартовых координат: их можно интерпретировать как двумерный вектор x , геометрически представляемый отрезком прямой линии из точки $(0,0)$ в точку (x, y) .

Двумерные преобразования на плоскости интерпретируются как движения точек по отношению к фиксированному базису.

В частности, линейные преобразования, представляемые матрицами, т.е. преобразования, при которых новые координаты точки линейно зависят от старых координат этой точки можно записать следующим образом:

$$x' = Tx. \quad (2.1.1)$$

Линейные преобразования могут быть различного типа, начиная от общего случая произвольных элементов матрицы T до специальных случаев, когда на элементы матрицы накладываются те или иные ограничения.

Любое *аффинное* преобразование имеет аффинное обратное преобразование. Произведение прямого и обратного преобразований дает единичное преобразование, оставляющее все на месте. Аффинное преобразование является самым общим взаимно однозначным отображением плоскости на плоскость, при котором сохраняются прямые линии. Сохраняются отношения длин отрезков, лежащих на одной или на параллельных прямых, и отношения площадей фигур. Параллельные прямые переходят в параллельные.

В аффинных преобразованиях плоскости особую роль играют несколько важных частных случаев, имеющих простой и наглядный геометрический смысл, и хорошо прослеживаемые геометрические характеристики.

К геометрическим аффинным операциям над изображением относятся (рис.2.1.1):

1. сдвиг (смещение) изображения;
2. растяжение (масштабирование изображения);
3. скос изображения;
4. поворот изображения.

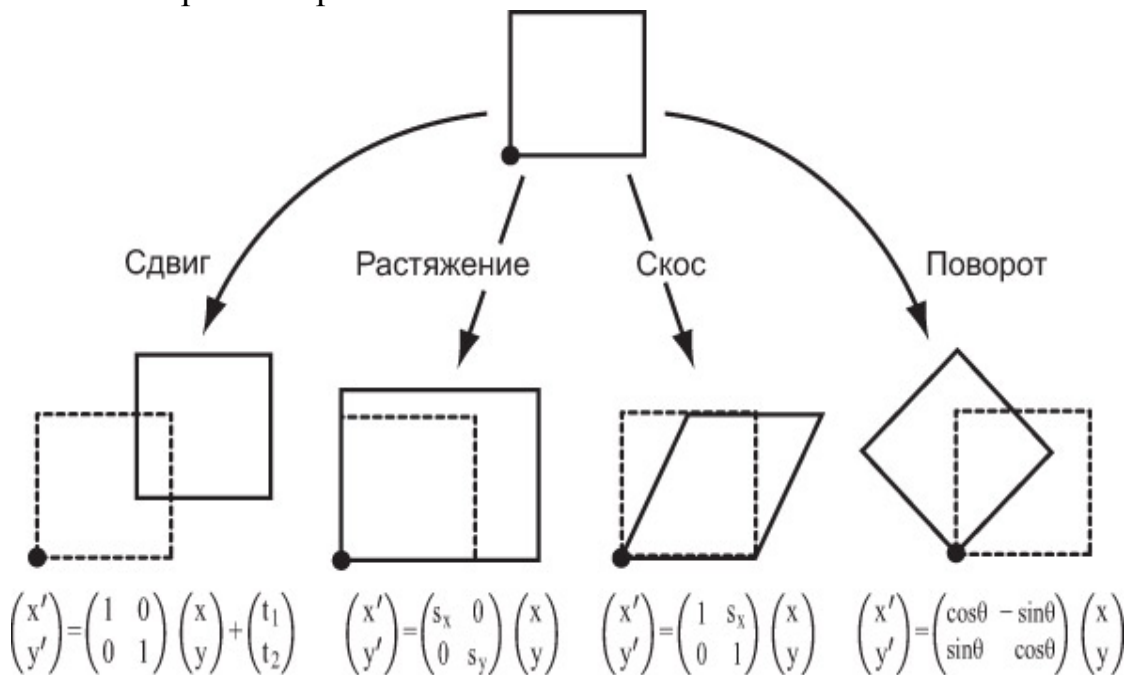


Рис.2.1.1. Геометрические операции над изображением

1. Скос изображения

$$\begin{aligned} x' &= ax + by + c, \\ y' &= dx + ey + f, \end{aligned} \tag{2.1.2}$$

или, в матричном виде

$$\mathbf{x}' = \mathbf{Ax} + \mathbf{c}, \quad \mathbf{x}' = \begin{pmatrix} x' \\ y' \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} a & b \\ d & e \end{pmatrix}, \quad \mathbf{c} = \begin{pmatrix} c \\ f \end{pmatrix}. \tag{2.1.3}$$

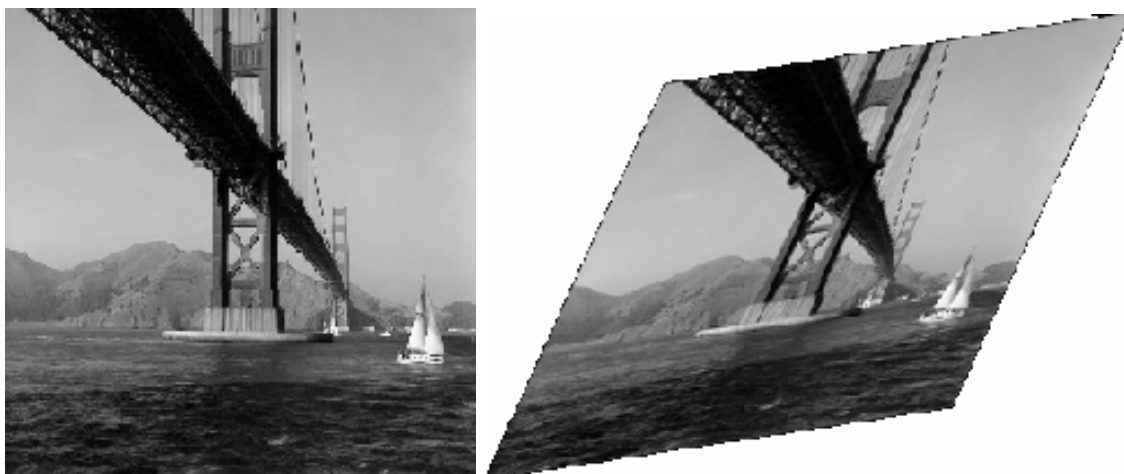
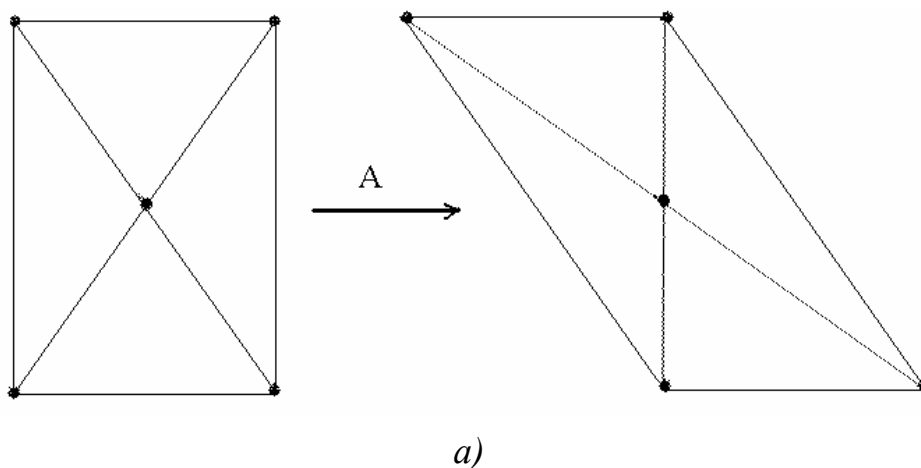


Рис.2.1.2. а) действие аффинного преобразования на пять точек (сдвиг, поворот, изменение масштабов вдоль осей, косоугольность с сохранением параллельных линий); б) исходное изображение (слева) и его аффинно-преобразованная копия (справа); параметры аффинного преобразования:

$$x' = 1.3x + 0.5y; y' = 0.2x + 1.1y$$

2. Растяжение (сжатие) вдоль координатных осей, задаваемое в виде:

$$\begin{aligned} x' &= \alpha x, \alpha > 0, \\ y' &= \beta y, \beta > 0. \end{aligned} \tag{2.1.4}$$

Растяжению вдоль соответствующей оси соответствует значение масштабного множителя большего единицы. В однородных координатах матрица растяжения (сжатия) имеет вид:

$$T_{af} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2.1.5}$$



Рис.2.1.3. Растяжение изображения

3. **Поворот** вокруг начальной точки на угол φ , описываемый формулой:

$$\begin{aligned} x' &= x \cos \varphi - y \sin \varphi, \\ y' &= x \sin \varphi + y \cos \varphi. \end{aligned} \quad (2.1.6)$$

Матрица вращения (для однородных координат):

$$\mathbf{T}_{af} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.1.7)$$

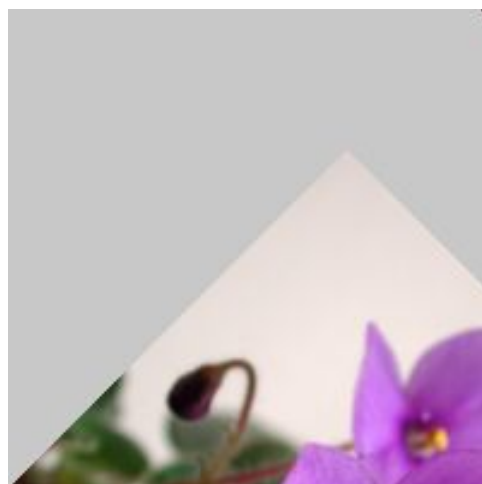


Рис.2.1.4. Поворот изображения

4. **Сдвиг**, задаваемый простейшими соотношениями:

$$\begin{aligned} x' &= x + c, \\ y' &= y + f. \end{aligned} \quad (2.1.8)$$

Матрица переноса имеет вид:

$$\mathbf{T}_{af} = \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & f \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.1.9)$$

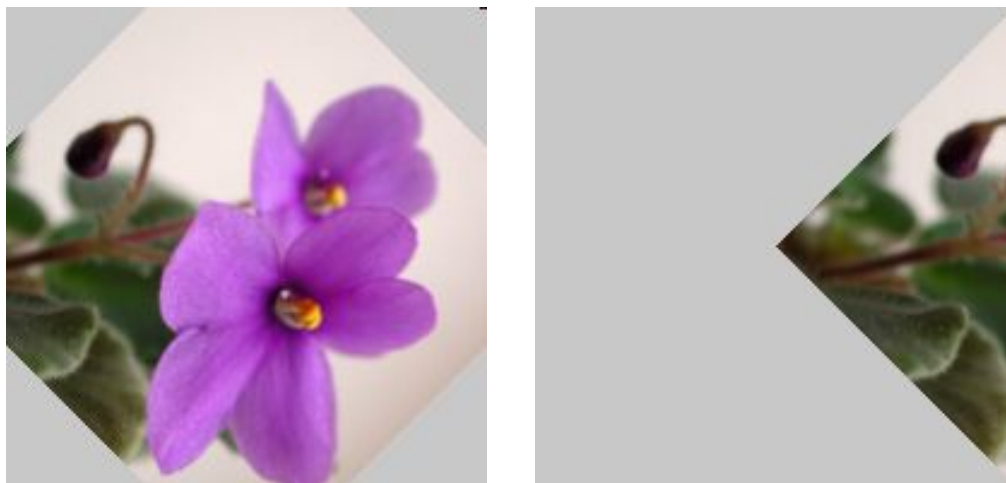


Рис.2.1.5. Сдвиг изображения

5. **Отражение** (относительно какой либо из осей, например оси абсцисс) задается при помощи формулы:

$$\begin{aligned} x' &= x, \\ y' &= -y. \end{aligned} \quad (2.1.10)$$

Матрица отражения, соответственно

$$\mathbf{T}_{af} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2.1.10)$$



Рис.2.1.6. Отражения изображения

2.2. Выполнение логических и арифметических операций над изображениями

Простейшими средствами, которые позволяют решать задачи обнаружения объектов на изображении и распознавания образов, являются логические и арифметические операции. Для выполнения над изображениями арифметических и логических операций функции можно разделить на две группы:

- **одноместные** (monadic) функции, которые в операции участвуют изображения и константа;
- **двуместные** (dyadic) функции, когда в операции участвуют два изображения.

2.2.1. Выполнение логических операций

Выполнение логических операций происходит попиксельно в соответствии с таблицами действия логических функций. Действие операций распространяется на яркость (интенсивность) каждого пиксела изображения.

1. Одноместная функция для выполнения логической операции И

Результат выполненного преобразования представлен на рисунке 2.2.1. С помощью этой операции нам удалось выделить область изображения, интенсивность которой больше либо равна 128.

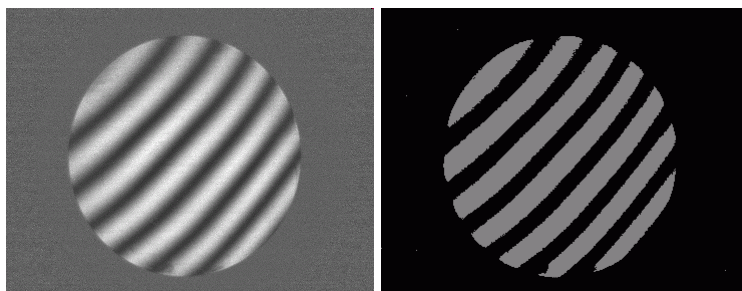


Рис. 2.2.1. Действие функции И с аргументом value=128

2. Двуместная функция для выполнения логической операции И

Результат выполнения этой функции представлен на рисунке 2.2.2. С помощью этой операции удастся выделить на изображении те области, которые имеют одинаковый цвет.

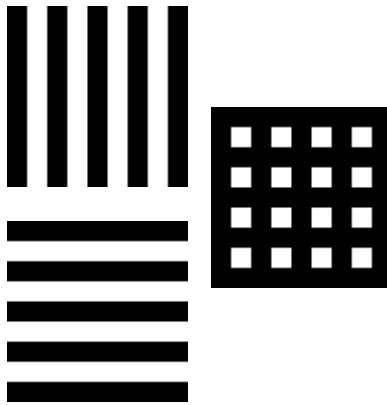


Рис. 2.2.2. Действие функции И

3. Одноместная функция для выполнения логической операции **ИЛИ**

Результат выполнения побитового ИЛИ с аргументом $value = 63$ представлен на рисунке 2.2.3. В данном примере операция ИЛИ дает четырехцветную топографическую карту исходного изображения.

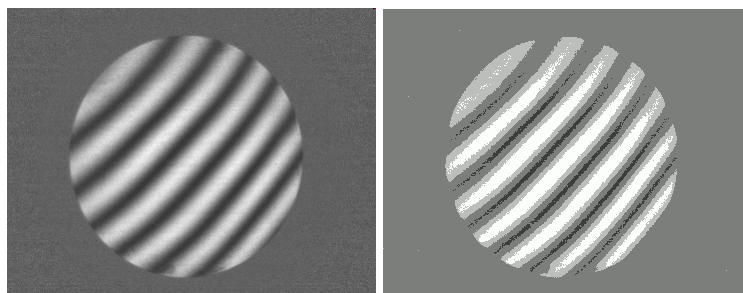


Рис. 2.2.3. Действие функции ИЛИ с аргументом $value=63$

4. Двуместная функция для выполнения логической операции **ИЛИ**

Результат выполнения этой функции представлен на рисунке 2.2.4. С помощью этой операции на изображении выделены пиксеты, которые хотя бы на одном из исходных изображениях имеют белый цвет.

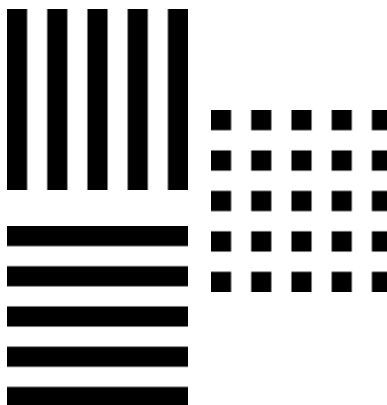


Рис. 2.2.4 Действие функции ИЛИ

5. Одноместная функция для выполнения логической операции ИСКЛЮЧАЮЩЕЕ ИЛИ

Результат выполнения побитового ИСКЛЮЧАЮЩЕГО ИЛИ с аргументом $value = 127$ представлен на рисунке 2.2.5.

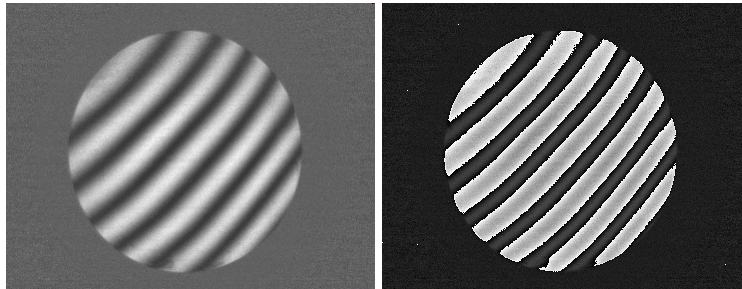


Рис. 2.2.5. Действие функции ИСКЛЮЧАЮЩЕЕ ИЛИ с аргументом $value=127$

6. Двуместная функция для выполнения логической операции ИСКЛЮЧАЮЩЕЙ ИЛИ

Результат выполнения этой функции представлен на рисунке 2.2.6. С помощью этой операции на изображении выделены пикселы, цвет которых на исходных изображениях не совпадает.

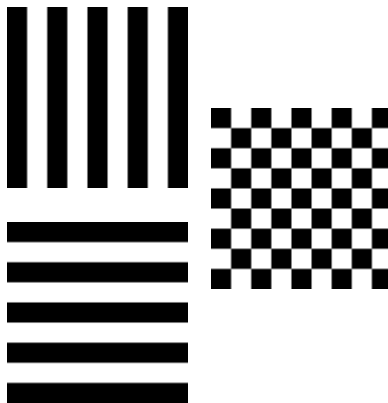


Рис. 2.2.6. Действие функции ИСКЛЮЧАЮЩЕЙ ИЛИ

7. Функция для выполнения логической операции НЕ

Операция НЕ позволяет инвертировать изображение (рис. 2.2.7).

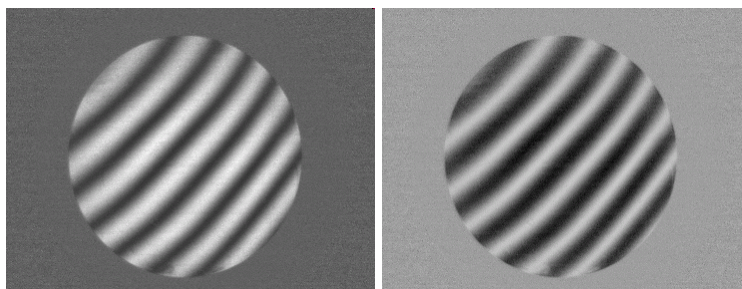


Рис. 2.2.7. Действие функции НЕ на полутоновое изображение

Все функции для выполнения арифметических и логических операций могут применяться не только к полутоновым, но и к цветным изображениям. Тогда побитовые операции выполняются с пикселем каждого цветового канала отдельно. На рисунке 2.2.8 представлен результат выполнения логической операции НЕ с цветным изображением.



Рис. 2.2.8. Действие функции НЕ на цветное изображение

2.2.2.Выполнение арифметических операций

Арифметические изображения выполняются попиксельно над интенсивностью (яркостью) каждого пикселя.

1. Одноместная функция для выполнения арифметической операции сложения

На рисунке 2.2.9 представлен результат выполнения функции сложения, с помощью которой в данном случае была повышена яркость исходного изображения.

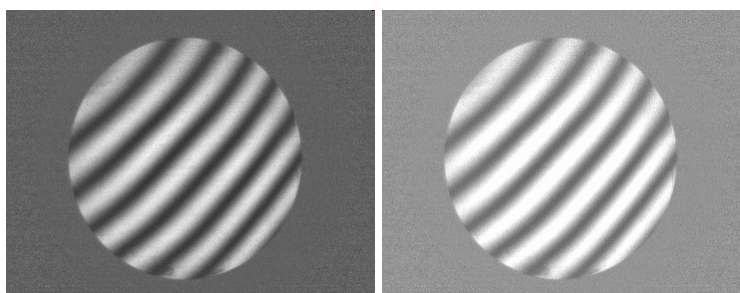


Рис. 2.2.9. Действие функции СЛОЖЕНИЕ с аргументом value=50

2. Двуместная функция для выполнения арифметического сложения двух изображений

Результат выполнения этой функции, приведенный на рисунке 2.2.10, показывает, что для данных исходных изображений операция алгебраического сложения эквивалентно операции логического ИЛИ.

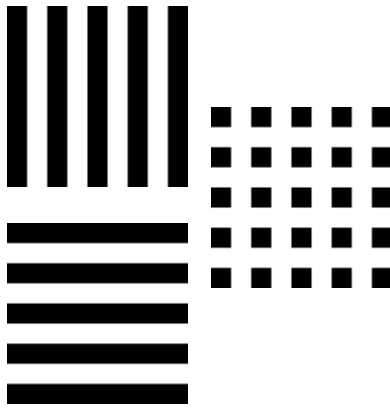


Рис. 2.2.10. Действие функции СЛОЖЕНИЕ

3. Одноместная функция для выполнения арифметической операции вычитания

Результат, представленный на рисунке 2.2.11 и 2.2.12 наглядно показывает зависимость результирующего изображения от порядка вычитания.

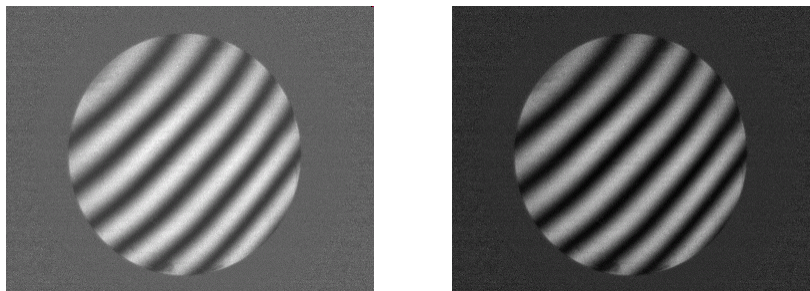


Рис. 2.2.11. Действие функции ВЫЧИТАНИЕ с аргументом value=50

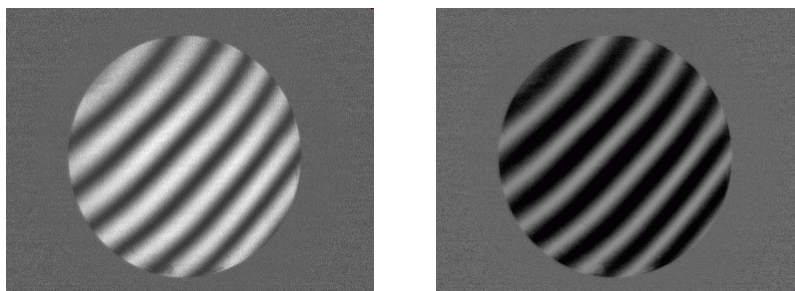


Рис. 2.2.12. Действие функции ВЫЧИТАНИЕ с аргументом value=200

4. Двуместная функция для выполнения арифметического сложения двух изображений

На рисунке 2.2.13 представлен результат вычитания изображения с горизонтальными линиями из изображения с вертикальными, а на рисунке 2.2.14. наоборот – результат вычитания изображения с вертикальными линиями из изображения с горизонтальными.

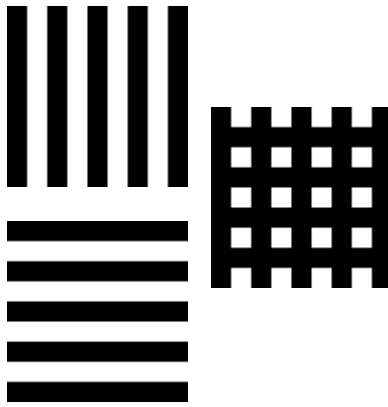


Рис. 2.2.13. Действие функции ВЫЧИТАНИЕ

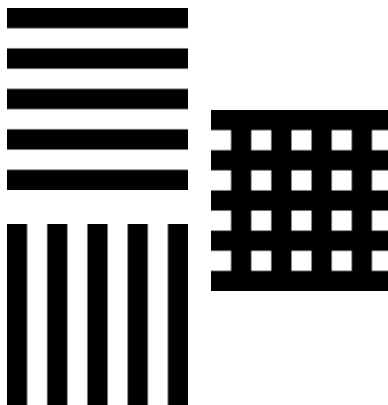


Рис. 2.2.14. Действие функции ВЫЧИТАНИЕ

5. Одноместная функция для выполнения арифметической операции умножения

Применение этой функции позволяет добиться существенного повышения яркости изображения (Рис. 2.2.15).

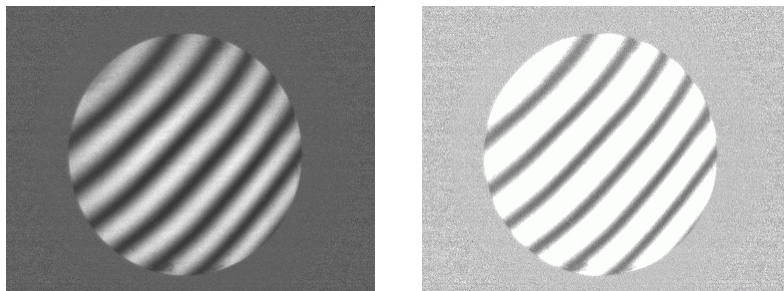


Рис. 2.2.15. Действие функции УМНОЖЕНИЕ с аргументом $value=2$

6. Двуместная функция для выполнения арифметического умножения двух изображений

Результат выполнения этой функции, приведенный на рисунке 2.2.16, показывает, что для данных исходных изображений операция алгебраического сложения эквивалентно операции логического И.

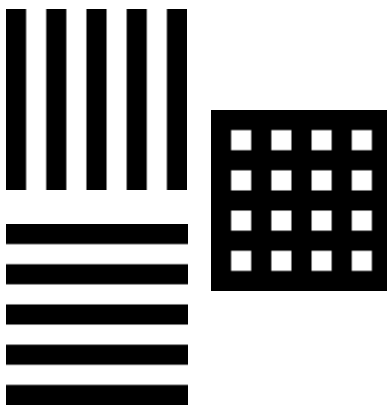


Рис. 2.2.16. Действие функции УМНОЖЕНИЕ

2.3. Фильтрация изображений

2.3.1. Общие понятия фильтрации изображений

Изображения, сформированные различными оптико-электронными системами и зарегистрированные с помощью разнообразных приемников искажаются действием помех различного характера. Искажения изображения вносятся всеми компонентами изображающего прибора, начиная с осветительной системой (например, неравномерность освещенности предмета). Искажения, которые вносит оптическая система, известны еще на этапе ее проектирования и называются аберрации. Искажения, которые вносят электронные приемники излучения, например ПЗС-матрицы, называются электронный шум. Помехи затрудняет визуальный анализ изображения и его автоматическую обработку.

Ослабление действия помех достигается фильтрацией. При фильтрации яркость (сигнал) каждой точки исходного изображения, искаженного помехой, заменяется некоторым другим значением яркости, которое признается в наименьшей степени искаженным помехой. Для выполнения фильтрации необходимо выработать принципы таких преобразований, которые основываются на том, что интенсивность изображения изменяется по пространственным координатам медленнее, чем функция помех. В других случаях, наоборот, признаком полезного сигнала являются резкие перепады яркости.

В методах фильтрации при оценке реального сигнала в некоторой точке кадра принимают во внимание некоторое множество (окрестность) соседних точек, воспользовавшись определенной похожестью сигнала в

этих точках. Понятие окрестности является достаточно условным. Окрестность может быть образована лишь ближайшими по кадру соседями, но могут быть окрестности, содержащие достаточно много и достаточно сильно удаленных точек кадра. В этом случае, степень влияния (вес) далеких и близких точек на решения, принимаемые фильтром в данной точке кадра, будет различной. Таким образом, идеология фильтрации основывается на рациональном использовании данных из рабочей точки и из ее окрестности.

При решении задач фильтрации используют вероятностные модели изображения и помехи, и применяют статистические критерии оптимальности. Это связано со случайным характером помехи и стремлением получить минимальное в среднем отличие результата обработки от идеального сигнала. Многообразие методов и алгоритмов фильтрации связано с большим разнообразием математических моделей сигналов, помех и различными критериями оптимальности.

Пусть $x_{i,j}$ - значение яркости изображения - полезного сигнала на пересечении i -ой строки и j -го столбца, а наблюдаемое на входе фильтра изображение описывается моделью:

$$y_{i,j} = f(x_{i,j}, n_{i,j}), \quad i = \overline{0, I-1}, j = \overline{0, J-1}. \quad (2.3.1)$$

Здесь $n_{i,j}$ - значение помехи в точке с координатами (i, j) , $f(\cdot)$ - функция, описывающая взаимодействие сигнала и помехи, а I и J - соответственно число строк и столбцов в кадре.

На рис. 2.3.1 показаны примеры окрестностей различных типов, изображенные в виде совокупностей точек. Центром окрестностей, рабочей точкой, в которой осуществляется обработка, является точка с координатами i, j (на рис. 2.3.1 не зачернена). В зависимости от типа окрестности различают каузальную (а), некаузальную (б) и полукаузальную (в) фильтрацию изображений. Понятие каузальности (причинно-следственной зависимости) связывают с соотношением координат текущей точки и точек, входящих в окрестность. Если обе координаты (номер строки и номер столбца) всех точек окрестности не превышают соответствующих координат текущей точки, то окрестность и

использующая ее обработка называется *каузальными*. Если точки окрестности удовлетворяют принципу каузальности, фильтрация, опирающаяся на использование такой окрестности, называется *некаузальной*. Если среди точек окрестности есть точки, координаты которых не превышают рабочую точку в одном направлении, но превышают ее в другом, то фильтрация является *полукаузальной* фильтрацией.

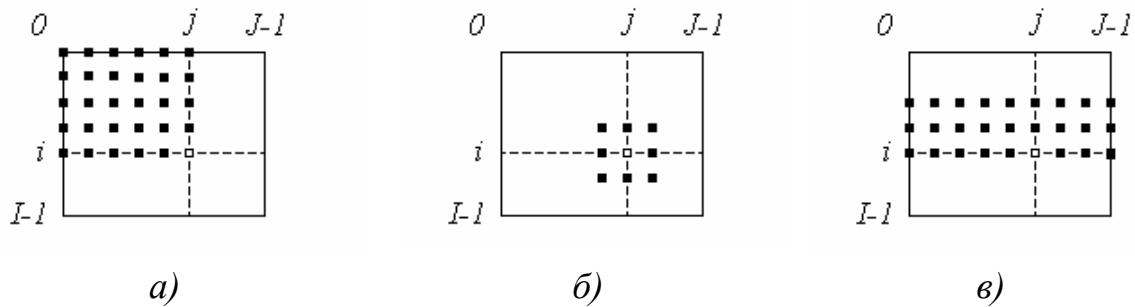


Рис. 2.3.1 Примеры окрестностей различных видов

При линейной фильтрации выходной эффект определяется линейной комбинацией входных данных:

$$x^*(i, j) = \sum_{(i_1, j_1) \in S} a(i_1, j_1) \cdot y(i - i_1, j - j_1). \quad (2.3.2)$$

В этом выражении $x^*(i, j) = x_{i, j}^*$ - результат фильтрации полезного сигнала $x(i, j) = x_{i, j}$ в точке кадра с координатами (i, j) ; S - множество точек (точнее - множество их координат), образующих окрестность; $a(i_1, j_1)$ - весовые коэффициенты, совокупность которых представляет собой *двумерную импульсную характеристику (ИХ)*. Если область S конечна, то импульсная характеристика имеет конечную длину и фильтр называется КИХ-фильтром. В противном случае импульсная характеристика имеет бесконечную длину, а фильтр название БИХ-фильтра. В выражении (2.3.2) принято, что ИХ не зависит от координат точки (i, j) , в которой определяется выходной эффект. Процедуры обработки изображений, обладающие свойством независимости от координат, называются *однородными*.

Наиболее распространенным критерием оптимальности, применяемым для оценки качества обработки, является критерий

минимума среднего квадрата ошибок. Применительно к фильтрации запишем его выражение в виде:

$$E \left\{ \left[x(i, j) - \sum_{(i_1, j_1) \in S} a(i_1, j_1) \cdot y(i - i_1, j - j_1) \right]^2 \right\} = \min_{a(\cdot)}, \quad (2.3.3)$$

где $E\{\circ\}$ - символ математического ожидания. Согласно (3.3) отыскание оптимального фильтра заключается в определении его ИХ таким образом, чтобы средний квадрат ошибки $\varepsilon(i, j) = x(i, j) - x^*(i, j)$, выражающей различие между сигналом $x(i, j)$ и оценкой $x^*(i, j)$, формируемой фильтром, был минимальным.

Очень часто при обработке стремятся сохранить среднюю яркость изображения. Математическое ожидание от обеих частей уравнения фильтрации можно записать следующим образом:.

$$m_{x^*} = m_y \cdot \sum_{(i_1, j_1) \in S} a(i_1, j_1), \quad (2.3.4)$$

где принято, что средняя яркость m_y входного изображения $y(\circ)$ не зависит от координат и, как результат, получено, что и средняя яркость m_{x^*} выходного изображения $x^*(\circ)$ постоянна во всех точках кадра. Очень часто при обработке стремятся сохранить среднюю яркость изображения. Как следует из полученного выражения, достичь этого удастся при выполнении равенства

$$\sum_{(i_1, j_1) \in S} a(i_1, j_1) = 1, \quad (2.3.5)$$

которое является дополнительным требованием к импульсной характеристике фильтра.

Потенциально наилучшие результаты фильтрации, достигаются при использовании некаузального принципа, поскольку этот принцип основан на применении абсолютно всех исходных данных при обработке каждой точки кадра.

2.3.2. Масочная фильтрация изображений

В практике цифровой обработки изображений широко используется *масочная фильтрация*. Ее линейная разновидность является одним из вариантов двумерной КИХ-фильтрации. В качестве маски используется множество весовых коэффициентов, заданных во всех точках окрестности S , обычно симметрично окружающих рабочую точку кадра. Распространенным видом окрестности, часто применяемым на практике, является квадрат 3×3 с рабочим элементом в центре, изображенный на рис. 2.3.1(б). Применяют различные разновидности масок, одним из эвристических вариантов является равномерная маска, все девять весовых коэффициентов которой равны $1/9$. Такой выбор коэффициентов отвечает условию сохранения средней яркости, и поэтому в процессе обработки центрировать изображение не требуется.

Визуально эффективность фильтрации можно оценить с помощью рис. 2.3.2. На рис. 2.3.2(а) показан зашумленный портрет при отношении сигнал/шум равном -5 дБ. Результат масочной фильтрации при оптимальном виде ИХ приведен на рис. 2.3.2(б).



а)



б)

Рис. 2.3.2. Пример масочной фильтрации при $q^2 = -5$ дБ

Необходимо отметить, что существует различие в оценках качества, даваемых человеческим глазом и применяемыми количественными показателями. Поэтому некоторые результаты, рассматриваемые с точки зрения математических показателей как катастрофические, визуально могут быть вполне удовлетворительными. Цифровая обработка

изображений находит применение в различных информационных системах с автоматическим принятием решений, основанным на этой обработке.

Функционирование таких систем, где отсутствует человеческий глаз, полностью подчинено математическим критериям, и качество их работы оценивается только математическими показателями. Аналогично, и качество изображений, используемых в этих системах, должно оцениваться только математическими критериями.

Существует несколько разновидностей масочной фильтрации. В таблице представлены наиболее распространенные фильтры и значения весовых коэффициентов для их масок.

Градиентный фильтр Превитта (вертикальный)	-1 0 1 -1 0 1 -1 0 1
Градиентный фильтр Превитта (горизонтальный)	1 1 1 0 0 0 -1 -1 -1
Градиентный фильтр Собеля (вертикальный)	-1 0 1 -2 0 2 -1 0 1
Градиентный фильтр Собеля (горизонтальный)	1 2 1 0 0 0 -1 -2 -1
Фильтр повышения резкости	-1 -1 -1 -1 16 -1 -1 -1 -1
Фильтр высоких частот Лапласа (3X3)	-1 -1 -1 -1 8 -1 -1 -1 -1
Фильтр высоких частот Лапласа (5X5)	-1 -3 -4 -3 -1 -3 0 6 0 -3 -4 6 20 6 -4 -3 0 6 0 -3 -1 -3 -4 -3 -1

Фильтр низких частот Гаусса (3X3)	1 2 1 2 4 2 1 2 1
Фильтр низких частот Гаусса (5X5)	2 7 12 7 2 7 31 52 31 7 12 52 127 52 12 7 31 52 31 7 2 7 12 7 2
Фильтр высоких частот (3X3)	-1 -1 -1 -1 8 -1 -1 -1 -1
Фильтр высоких частот (5X5)	-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 24 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1

2.3.3. Нелинейная фильтрация

Все линейные алгоритмы фильтрации приводят к сглаживанию резких перепадов яркости изображений, прошедших обработку. Линейные процедуры являются оптимальными при гауссовском распределении сигналов, помех и наблюдаемых данных. Обычно этому условию отвечают шумовые помехи на изображениях, поэтому при их подавлении линейные алгоритмы имеют высокие показатели. А если, например, задача обработки изображения состоит в выявлении границ объекта, то линейная фильтрация не подходит для ее решения.

Реальные изображения не подчиняются данному распределению вероятностей (разнообразные перепады яркости на границах, переходы от одной текстуры к другой и т. п.). Приходится иметь дело с изображениями, искаженными помехами других типов. Одной из них является импульсная помеха. При ее воздействии на изображении наблюдаются белые или (и) черные точки, хаотически разбросанные по кадру. Применение линейной фильтрации в этом случае неэффективно - каждый из входных импульсов (по сути - дельта-функция) дает отклик в виде импульсной характеристики

фильтра, а их совокупность способствует распространению помехи на всю площадь кадра.

Удачным решением перечисленных проблем является применение *медианной фильтрации*. Отметим, что медианная фильтрация представляет собой эвристический метод обработки, ее алгоритм не является математическим решением строго сформулированной задачи. Как и в методе масочной фильтрации, при применении медианного фильтра происходит последовательная обработка каждой точки кадра, а для вычисления оценки используется некоторая окрестность (окно). Наиболее часто применяются варианты окон в виде креста и в виде квадрата. Размеры окна изменяются в зависимости от задачи и характера изображения.

Отсчеты изображения, оказавшиеся в пределах окна, образуют *рабочую выборку* текущего шага.

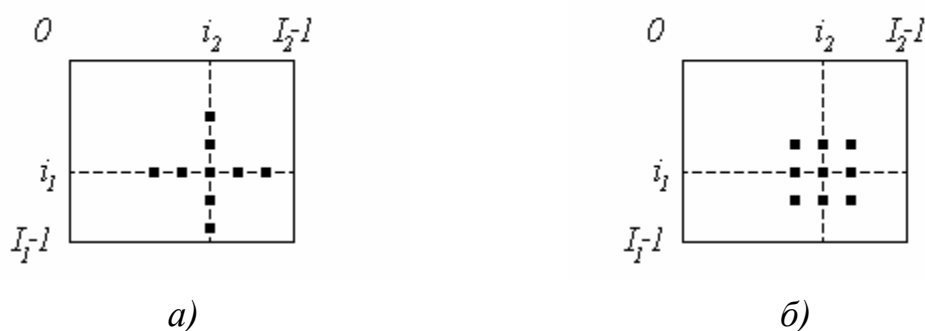


Рис. 2.3.3. Примеры окон при медианной фильтрации

Двумерный характер окна позволяет выполнять, по существу, двумерную фильтрацию, поскольку для образования оценки привлекаются данные как из текущих строки и столбца, так и из соседних. Обозначим рабочую выборку в виде одномерного массива $Y = \{y_1, y_2, \dots, y_n\}$; число его элементов равняется размеру окна, а их расположение произвольно. Обычно применяют окна с нечетным числом точек n (это автоматически обеспечивается при центральной симметрии апертуры и при вхождении самой центральной точки в ее состав).

Если упорядочить последовательность $\{y_i, i = \overline{1, n}\}$ по возрастанию, то ее медианой будет тот элемент выборки, который занимает центральное положение в этой упорядоченной последовательности. Полученное таким образом число и является продуктом фильтрации для текущей точки кадра.

Понятно, что результат такой обработки не зависит от того, в какой последовательности представлены элементы изображения в рабочей выборке Y . Введем формальное обозначение описанной процедуры в виде:

$$x^* = med(y_1, y_2, \dots, y_n). \quad (2.3.5)$$

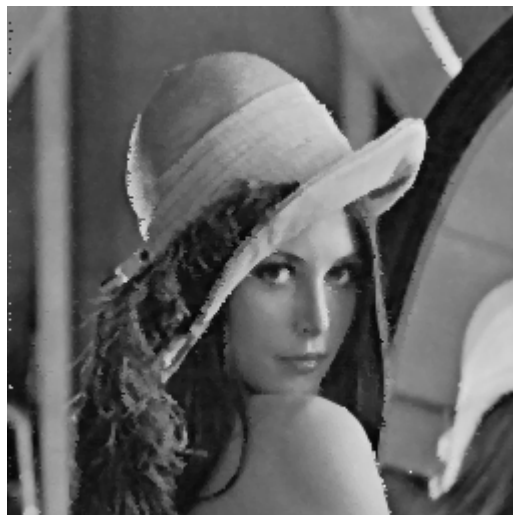
Предположим, что апертура фильтра находится вблизи границы, разделяющей светлый и темный участки изображения, при этом ее центр располагается в области темного участка. Тогда, вероятнее всего, рабочая выборка будет содержать большее количество элементов с малыми значениями яркости, и, следовательно, медиана будет находиться среди тех элементов рабочей выборки, которые соответствуют этой области изображения. Ситуация меняется на противоположную, если центр апертуры смещен в область более высокой яркости.

Рассмотрим пример. Предположим, что выборка имеет вид: $Y = \{136, 110, 99, 45, 250, 55, 158, 104, 75\}$, а элемент 250, расположенный в ее центре, соответствует текущей точке фильтрации (i_1, i_2) (рис. 2.3.3). Большое значение яркости в этой точке кадра может быть результатом воздействия импульсной (точечной) помехи. Упорядоченная по возрастанию выборка имеет при этом вид $\{45, 55, 75, 99, 104, 110, 136, 158, 250\}$, следовательно, в соответствии с процедурой (2.3.5), получаем $x^* = med(y_1, y_2, \dots, y_9) = 104$. Видим, что влияние «соседей» на результат фильтрации в текущей точке привело к «игнорированию» импульсного выброса яркости, что следует рассматривать как эффект фильтрации. Если импульсная помеха не является точечной, а покрывает некоторую локальную область, то она может быть подавлена. Это произойдет, если размер этой локальной области будет меньше, чем половина размера апертуры МФ. Поэтому для подавления импульсных помех, поражающих локальные участки изображения, следует увеличивать размеры апертуры МФ.

Действие МФ состоит в «игнорировании» экстремальных значений входной выборки - как положительных, так и отрицательных выбросов. Такой принцип подавления помехи может быть применен и для ослабления шума на изображении.



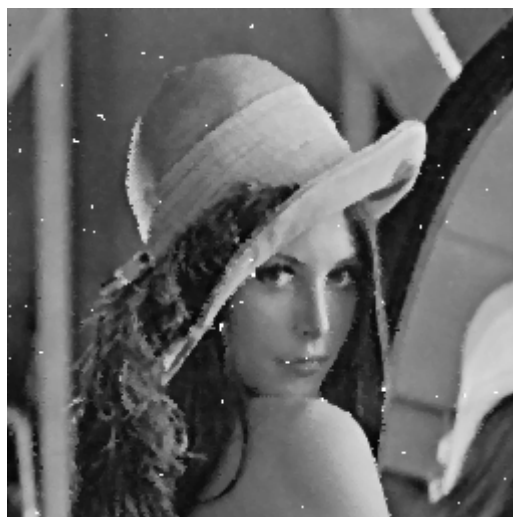
а)



б)



в)



г)



д)



е)

Рис. 2.3.4. Примеры медианной фильтрации

Однако исследование подавления шума при помощи медианной фильтрации показывает, что ее эффективность при решении этой задачи ниже, чем у линейной фильтрации. Результаты работы медианного фильтра представлены на рисунке 2.3.4.

Аналогично, строятся фильтры максимумов и минимумов.

2.4. Восстановление изображений на основе обратной фильтрации

Математическую модель формирования некогерентного изображения можно записать в виде выражения с четырьмя преобразованиями Фурье:

$$I'(\eta'_x, \eta'_y) = F^{-1} \left[F \left[\left| T(\rho_x, \rho_y) \right|^2 \right] \cdot F \left[\left| F^{-1} \left[f(\rho_x, \rho_y) \right] \right|^2 \right] \right]. \quad (2.4.1)$$

В частотных координатах это выражение можно переписать следующим образом:

$$\tilde{I}'(v'_x, v'_y) = \tilde{I}(v_x, v_y) \cdot D(v_x, v_y), \quad (2.4.2)$$

где $\tilde{I}(v_x, v_y)$ – спектр предмета, $\tilde{I}'(v'_x, v'_y)$ – спектр изображения, $D(v_x, v_y)$ – оптическая передаточная функция.

Оптическую передаточную функцию можно представить, как произведение всех ОПФ каскада преобразователей, составляющих моделируемый прибор:

$$D = D_1 \cdot D_2 \cdots D_n = \prod_{k=1}^n D_k. \quad (2.4.3)$$

Каскад составляют такие преобразователи, как смаз $D(v_x, v_y) = \text{sinc}(\pi v_x a)$, вибрации $D(v_x, v_y) = J_0(\pi v_x a)$, двоение $D = \cos(\pi v_x a)$ и другие. Все они ухудшают качество формируемого изображения. Если $a = 1$, то МПФ этих преобразователей выглядит следующим образом:

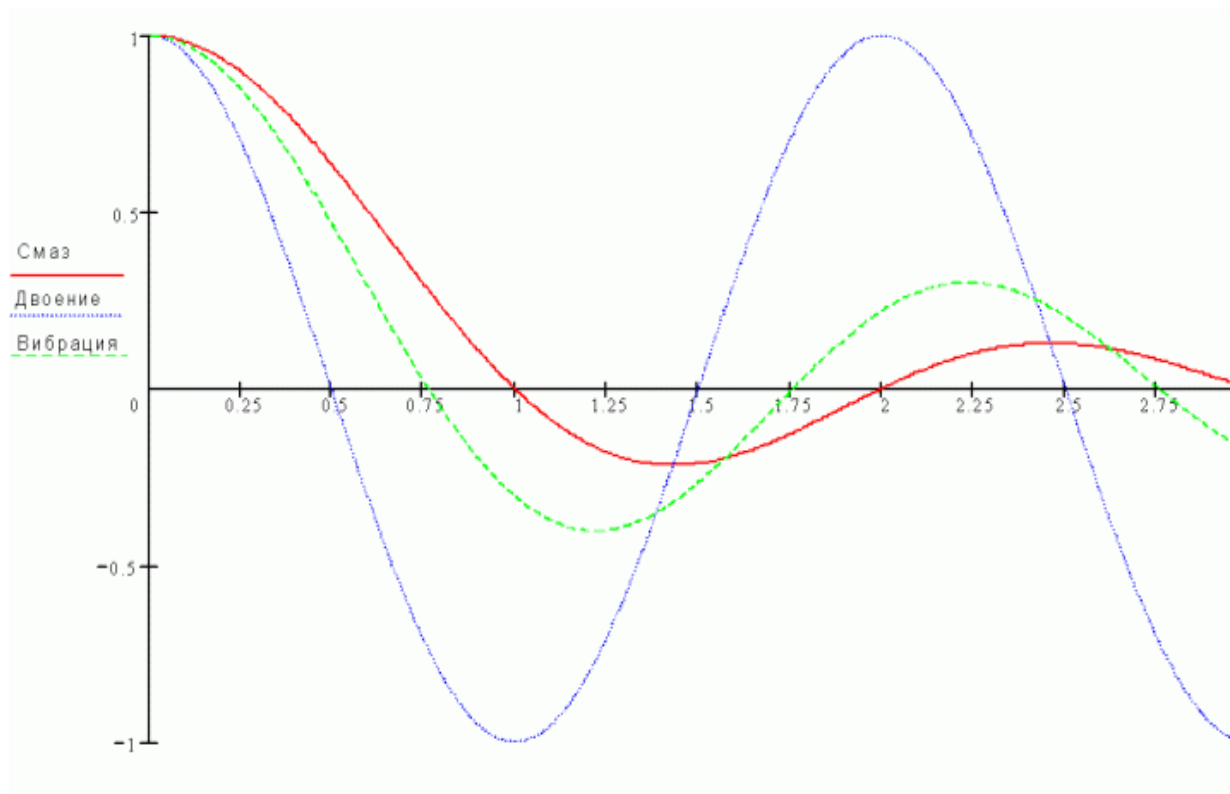


Рис. 2.4.1. График МПФ смаза, двоения и вибрации

Возникает вопрос: можно ли, зная характеристики искажающих звеньев, улучшить качество изображения? Ведь достаточно добавить в каскад звено, выполняющее обратное преобразование с передаточной функцией:

$$R(v_x, v_y) = \frac{1}{D(v_x, v_y)}. \quad (2.4.4)$$

Тогда, домножение ОПФ на обратную функцию приведет к преобразованию приведенного выше выражения:

$$\tilde{I}(v'_x, v'_y) = \frac{\tilde{I}(v_x, v_y) \cdot D(v_x, v_y)}{D(v_x, v_y)} = \tilde{I}(v_x, v_y). \quad (2.4.5)$$

То есть спектр итогового изображения будет подобен спектру исходного предмета. Возможно ли это? В общем случае нет по следующим причинам:

1. Если ОПФ имеет точки, в которых $D = 0$, то восстанавливающий преобразователь должен иметь ОПФ, в которой в этих точках $R = \infty$. Все известные нам ОПФ имеют такие точки. Информация

о предмете на частотах, в которых значение ОПФ близко к 0, безвозвратно утеряна. Хотя эти области небольшие, но все-таки построить ОПФ, полностью восстанавливающую исходный предмет, невозможно.

2. Регистрируется интенсивность изображения, а не его спектр. Информация о фазе спектра изображения оказывается утеряна. Таким образом, восстановление амплитудно-фазовых предметов методом обратной фильтрации не представляется возможным.

Хотя в некоторых случаях можно добиться улучшения качества изображения путем обработки изображений, построенной на этих принципах. Такие методы называются *методы обратной фильтрации*. Обратная фильтрация чаще всего выполняется на компьютере численными методами, но может быть выполнена и оптическими методами (методами Фурье-оптики).

Рассмотрим обратную фильтрацию на примере изображения, искаженного смазом. Оптическая передаточная функция смаза определяется выражением:

$$D(v_x, v_y) = \text{sinc}(\pi v_x a) = \frac{\sin(\pi v_x a)}{\pi v_x a}. \quad (2.4.6)$$

Попытаемся построить ОПФ восстанавливающего преобразователя:

$$R(v_x, v_y) = \frac{\pi v_x a}{\sin(\pi v_x a)}. \quad (2.4.7)$$

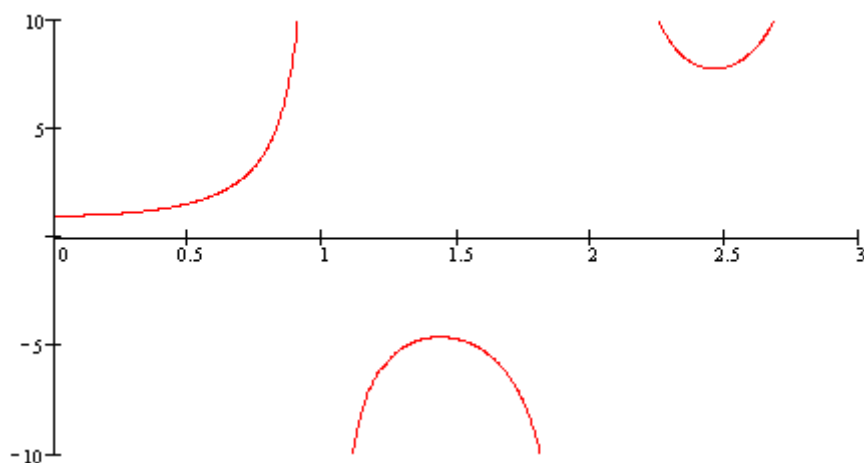


Рис. 2.4.2. ОПФ восстанавливающего преобразователя для смаза

В области целых значений v_x эта функция уходит в бесконечность. Удобнее представить ее в виде произведения двух функций:

$$R(v_x, v_y) = \pi v_x a \cdot \frac{1}{\sin(\pi v_x a)} = R_1(v_x, v_y) \cdot R_2(v_x, v_y). \quad (2.4.8)$$

Тогда, первая часть восстанавливающей функции $R_1(v_x, v_y) = \pi v_x a$ представляет собой прямую:

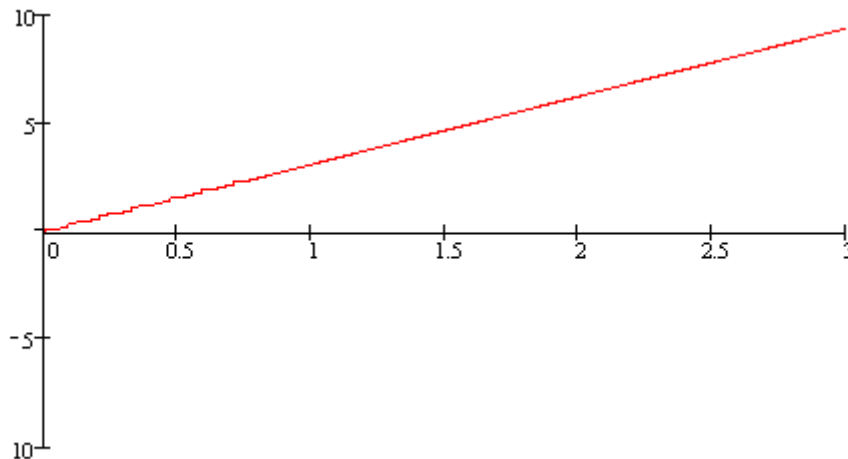


Рис. 2.4.3. Первая часть восстанавливающей функции

После умножения функция sinc, которая имеется в выражении передаточной характеристики смаза, превратится в синус, и ОПФ всего прибора будет иметь вид синусоиды, а это ОПФ искажения подобного двоению. Таким образом, путем обратной фильтрации вместо искажения типа смаз получили искажение типа двоения. Если объект мал, а смаз велик, то для оценки формы и размеров объекта и в некоторых других задачах такой фильтрации достаточно.

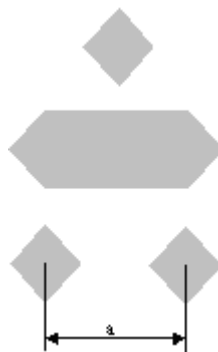


Рис. 2.4.4. Смаз и двоение изображения

Если смаз мал по сравнению с объектом, то такое воздействие на спектр не приведет к разделению изображения двоящегося объекта. Дальнейшая модификация ОПФ восстанавливающего преобразователя

заключается в следующем. Для того чтобы убрать двоение изображения, можно применить преобразователь со следующей *ОПФ*:

$$R(v_x, v_y) = \text{sign}\left(\frac{1}{\sin(\pi v_x a)}\right). \quad (2.4.9)$$

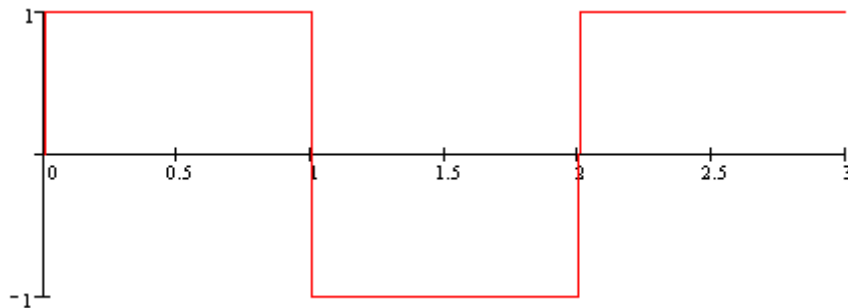


Рис. 2.4.5. Преобразователь для устранения двоения

Тогда синус меняет отрицательные части на положительные, и мы избавимся от двоения. Итоговая *ОПФ* каскада будет выглядеть следующим образом:

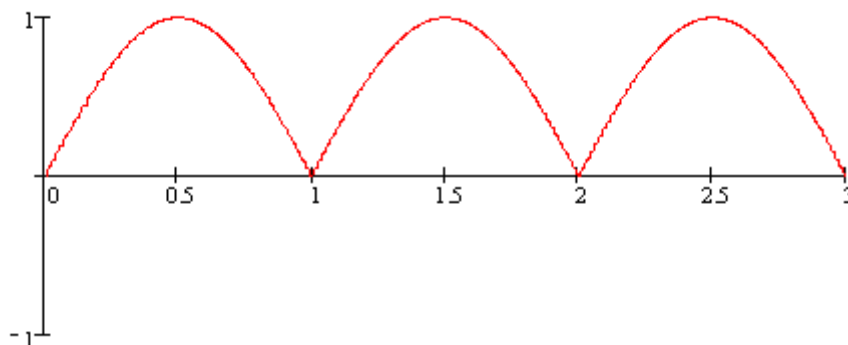


Рис. 2.4.6. Итоговая ОПФ каскада преобразователей

Эта *ОПФ* отличается от *ОПФ* идеального прибора $D=1$, тем, что есть участки с отсутствующими частотами, но во многих случаях это не важно. На практике гораздо важнее восстановить форму и размеры предмета.

ОПФ восстанавливающего преобразователя в данном случае определяется выражением:

$$R(v_x, v_y) = \pi v_x a \cdot \text{sign}\left(\frac{1}{\sin(\pi v_x a)}\right). \quad (2.4.10)$$

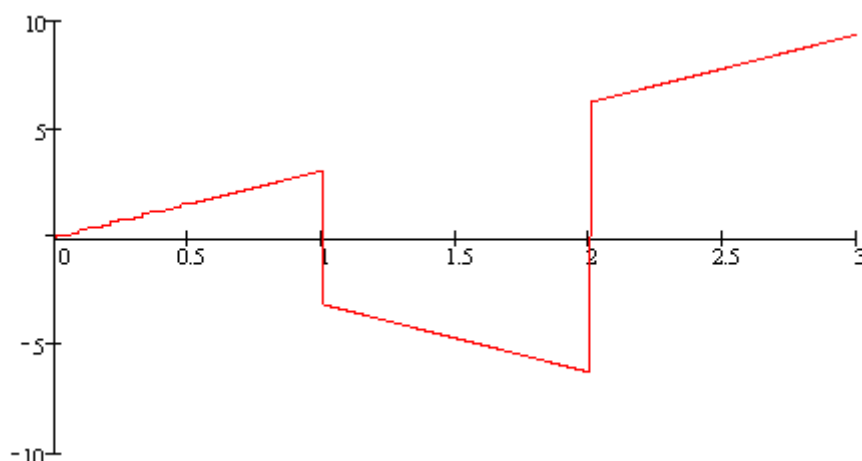


Рис. 2.4.7. ОПФ восстанавливающего преобразователя

Методы обратной фильтрации имеют ряд недостатков:

1. При синтезе восстанавливающего фильтра не учитывается факт влияния объектов, расположенных вне поля зрения объектива, на значение яркости на краях искаженного изображения. При восстановлении реально искаженных изображений таким фильтром возникают краевые эффекты, компенсация которых является отдельной задачей при реализации алгоритмов восстановления изображения на основе пространственной фильтрации.
2. На практике изображение обычно бывает зашумлено. Это связано с особенностями регистрации. Полное выражение, определяющее спектр изображения, имеет следующий вид:

$$\tilde{I}(v'_x, v'_y) = \tilde{I}(v_x, v_y) \cdot D(v_x, v_y) + N(v_x, v_y), \quad (2.4.11)$$

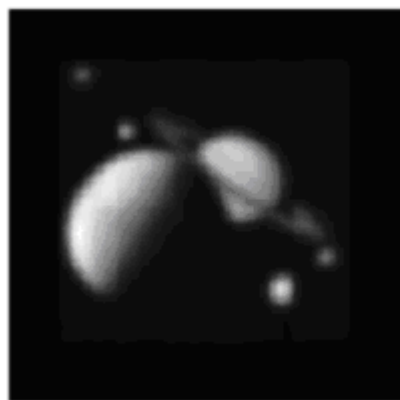
где $N(v_x, v_y)$ – спектр функции шума $n(x, y)$. Обратная фильтрация приведет к тому, что спектр шума будет отмасштабирован:

$$\tilde{I}(v'_x, v'_y) = \tilde{I}(v_x, v_y) + \frac{N(v_x, v_y)}{D(v_x, v_y)}. \quad (2.4.12)$$

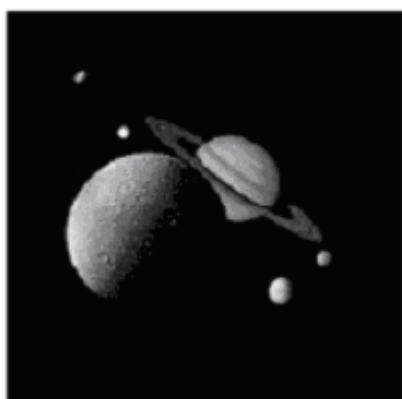
Это означает, что на частотах, в которых значение $D(v_x, v_y)$ невелико (а это и есть высокие частоты, характерные для шума), значение спектра существенно возрастет. Изображение, которое должно быть восстановлено, окажется существенно зашумлено (рис.2.4.8).



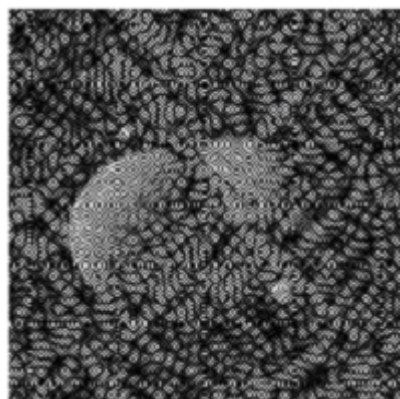
Исходное изображение



*Расфокусировка и обрезание
краев*



Восстановленное изображение



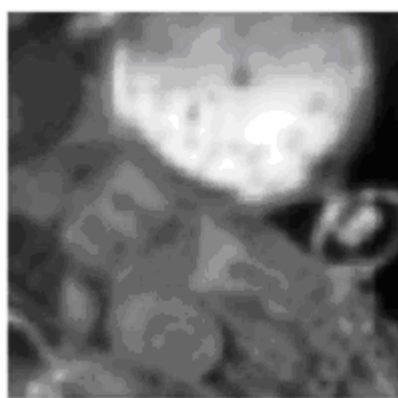
*Восстановленное зашумленное
изображения*

Рис. 2.4.8. Пример восстановления изображений

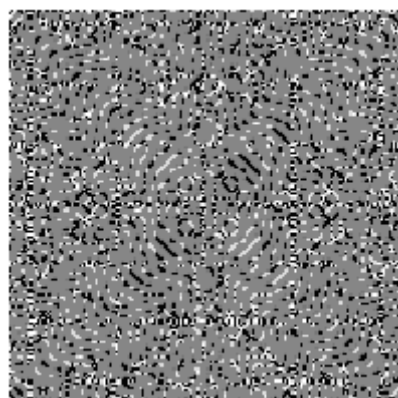
На рис. 2.4.9 приведены результаты восстановления изображений инверсным фильтром. Исходные изображения размером 200×200 элементов были подвергнуты дефокусировке с последующим «обрезанием» краев до размеров 170×170 элементов. Практически идеальное восстановление изображения объясняется тем, что объекты наблюдаются на фоне постоянной яркости и расположены в центре кадра. Наличие даже относительно слабого шума приводит к появлению интенсивных шумовых составляющих на выходе инверсного фильтра, полностью разрушающих изображение.



а) Исходное изображение



*б) Расфокусировка и обрезание
краев*



*в) Восстановленное
изображение*

Рис. 2.4.9. Пример восстановления изображений

К дефокусированному изображению был добавлен аддитивный шум. Из восстановленного изображения видно, что даже при малом уровне шума метод инверсной фильтрации дает очень плохой результат.

Изображение может иметь такие свойства, что восстановить его инверсным фильтром просто не удастся из-за краевых эффектов. Таким образом, несмотря на очевидную простоту метода инверсной фильтрации, он может успешно использоваться для восстановления ограниченного класса изображений, у которых уровень фона на краях постоянен. Кроме того, метод инверсной фильтрации обладает чрезвычайно низкой помехоустойчивостью.

2.5. Восстановление изображения на основе фильтрации Винера

Инверсная фильтрация обладает низкой помехоустойчивостью, потому что этот метод не учитывает зашумленность наблюдаемого изображения. Значительно менее подвержен влиянию помех и сингулярностей, обусловленных нулями передаточной функции искажающей системы, *фильтр Винера*, т.к. при его синтезе наряду с видом *ФРТ* используется информация о спектральных плотностях мощности изображения и шума.

Спектральная плотность сигнала определяется соотношением:

$$S_{II}(v) = F[R(\omega)], \quad (2.5.1)$$

где $R(\omega) = \int I(x)I(x - \omega)dx$ – автокорреляционная функция.

Взаимная спектральная плотность сигнала определяется соотношением:

$$S_{II'}(v) = F[R(\omega)], \quad (2.5.2)$$

где $R(\omega) = \int I(x)I'(x - \omega)dx$ – функция взаимной корреляции.

При построении фильтра Винера ставится задача минимизации среднеквадратического отклонения обработанного изображения от предмета:

$$E\{[I(x, y) - I'(x, y)]^2\} = \min, \quad (2.5.3)$$

где $E\{\}$ – математическое ожидание. Преобразуя эти выражения можно показать, что минимум достигается, когда передаточная функция определяется следующим выражением:

$$D(v_x, v_y) = \frac{S_{II'}(v_x, v_y)}{S_{II}(v_x, v_y)}. \quad (2.5.4)$$

Дальнейший анализ показывает, что восстановление изображения, формирование которого описывается выражением должно осуществляться с использованием следующего *ОПФ* восстанавливающего преобразователя:

$$R(v_x, v_y) = \frac{1}{D(v_x, v_y)} \cdot \frac{|D(v_x, v_y)|^2}{|D(v_x, v_y)|^2 + [S_{NN}(v_x, v_y)/S_{II}(v_x, v_y)]}. \quad (2.5.5)$$

Если шума на изображении нет, то спектральная плотность функции шума равна 0 и фильтр Винера превращается в обычный обратный фильтр.

При уменьшении спектральной плотности мощности исходного изображения передаточная функция фильтра Винера стремится к нулю. Для изображений это характерно на верхних частотах.



ОПФ

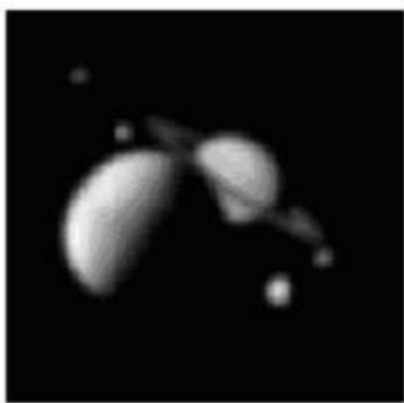


Инверсные фильтры



Фильтры Винера

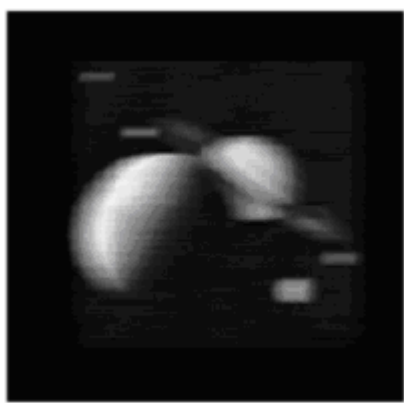
Рис. 2.5.1. Примеры фильтров



*а) Расфокусировка, шум и
обрезание краев*



*б) Восстановленное
изображение*



в) Смаз, шум и обрезание краев г) Восстановленное изображение

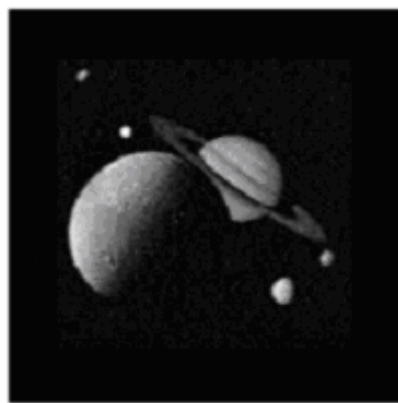
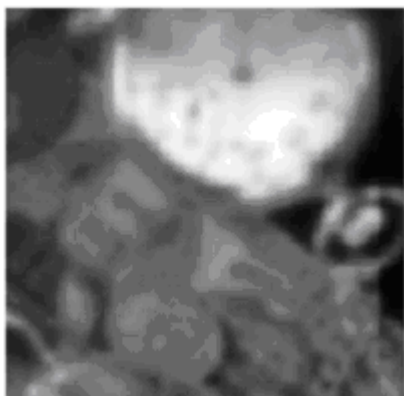


Рис. 2.5.2. Примеры восстановления изображений

Примеры восстановления (рис.2.5.2 и рис. 2.5.3) показывают, что фильтр Винера значительно лучше подавляет шумы. Осциллирующая помеха на результатах восстановления изображения вызвана краевыми эффектами. Очевидно, что ее уровень существенно меньше, чем при инверсной фильтрации, однако винеровский фильтр лишь частично компенсирует краевые эффекты, которые делают качество восстановления неудовлетворительным. Компенсацией краевых эффектов занимаются специально. Однако эти методы не являются оптимальными и не всегда обеспечивают эффективную компенсацию искажений и избавление от краевых эффектов одновременно.

На частотах, соответствующих нулям передаточной функции формирующей системы, передаточная функция фильтра Винера равна нулю. Таким образом, решается проблема сингулярности восстанавливающего фильтра.



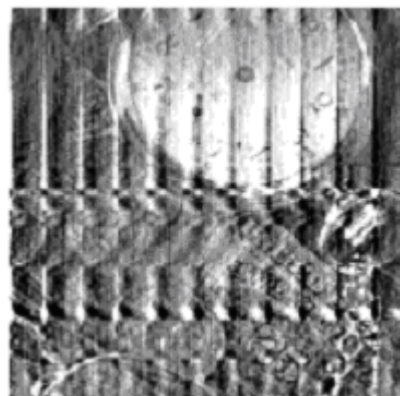
д) Расфокусировка, шум и обрезание краев



е) Восстановленное изображение



ж) Смаз, шум и обрезание краев



з) Восстановленное изображение

Рис. 2.5.3. Примеры восстановления изображений

2.6. Итерационные методы восстановления изображений

Итерационными методами называют способы решения задач, в которых, выбирая некоторое начальное приближенное решение, вычисляют следующие, более точные приближения, используя предыдущие. Рассмотрим один из способов построения итерационных процедур, основанный на разложении в ряд частотной характеристики инверсного фильтра. Представим передаточную функцию инверсного фильтра в виде геометрической прогрессии:

$$R(v_x, v_y) = \frac{1}{D(v_x, v_y)} = \sum_{k=0}^{\infty} \left(1 - D(v_x, v_y)\right)^k. \quad (2.6.1)$$

Искомое изображение находится пошаговым приближением:

$$\tilde{I}_0(v_x, v_y) = \tilde{I}'(v_x, v_y)$$

$$\tilde{I}_1(v_x, v_y) = \tilde{I}'(v_x, v_y) + (1 - D(v_x, v_y)) \tilde{I}'(v_x, v_y) =$$

$$= \tilde{I}'(v_x, v_y) + (1 - D(v_x, v_y)) \tilde{I}_0(v_x, v_y) \quad \dots$$

$$\tilde{I}_n(v_x, v_y) = \tilde{I}'(v_x, v_y) + (1 - D(v_x, v_y)) \tilde{I}_{n-1}(v_x, v_y) \quad (2.6.2)$$

Таким образом, каждое последующее приближение вычисляется по предыдущему. Если в результате последовательных приближений на некотором шаге будет найдено решение, то на последующих шагах, как нетрудно убедиться, изображение изменяться не будет.

При использовании итерационных алгоритмов необходимо знать, сходится ли итерационный процесс. Сходимость алгоритма к решению определяется сходимостью ряда бесконечной геометрической прогрессии.

Этот ряд сходится при $|1 - D(v_x, v_y)| < 1$, т.е. когда передаточная функция искажающей системы удовлетворяет условию $0 < D(v_x, v_y) < 1$.

Это условие часто не выполняется (например, ОПФ смаза может быть меньше 1). Тогда приведенное выше выражение заменяют на эквивалентное соотношение:

$$R(v_x, v_y) = \frac{D^*(v_x, v_y)}{|D(v_x, v_y)|^2} = \frac{D^*(v_x, v_y)}{M(v_x, v_y)} = D^*(v_x, v_y) \sum_{k=0}^{\infty} (1 - M(v_x, v_y))^k \quad (2.6.3)$$

Этот метод позволяет эффективно бороться с краевыми эффектами и чрезмерным усилением шумов при восстановлении изображений.

Итеративный процесс всегда можно остановить, если шум и осциллирующая помеха на изображении резко усиливаются. Остановка итеративного процесса означает усечение ряда, что приводит к ограничению коэффициента усиления за пределами некоторой граничной частоты. С увеличением длины ряда возрастают граничная частота и коэффициент усиления фильтра.



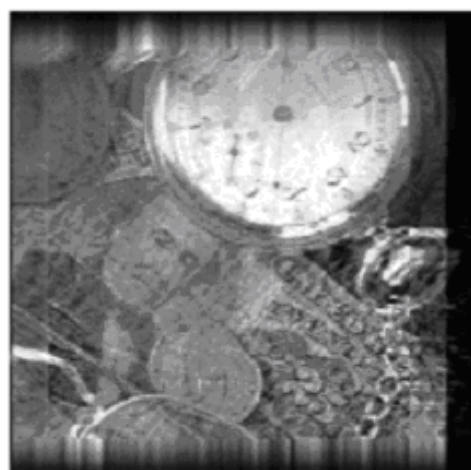
Исходное изображение



Смаз



*Восстановленное изображение
(количество итераций – 105)*



*Восстановленное изображение
(количество итераций – 135)*

Рис. 2.6.1. Примеры восстановления изображений

2.7. Алгебраические методы восстановления изображений

Другой подход к восстановлению изображений основывается на использовании алгебраических методов, в которых делается попытка представить процесс формирования изображений в виде матричной операции:

$$I' = I \cdot D. \quad (2.7.1)$$

Зная изображение I' и передаточную характеристику D , делаются попытки восстановить предмет I . Решение этой задачи может быть сформулировано следующим образом:

$$I = I' \cdot D^{-1}. \quad (2.7.2)$$

Недостатком алгебраических алгоритмов, лишаящим их практической ценности, является необходимость выполнения трудоемких операций обращения, умножения и транспонирования матриц огромных размеров. Подобные методы если и используются, то в очень редких случаях, когда размеры изображений невелики.

Таким образом, можно выделить три основных фактора, которые существенно усложняют решение проблемы восстановления изображений.

1. Искажения типа расфокусировка или смаз проявляются в ослаблении верхних пространственных частот изображения, т.к. формирующие системы представляют собой фильтры нижних частот. При этом отношение сигнал/шум на верхних частотах, определяющих четкость изображения, будет значительно хуже, чем для изображения в целом. Если система, формирующая изображение, ослабляет сигнал на каких-то пространственных частотах, то при восстановлении он должен быть усилен в той мере, в какой был ослаблен. Вместе с сигналом будут усиливаться и шумы. Поэтому улучшение качества изображения по резкости может привести к ухудшению его качества по зашумленности.
2. Яркость на краях кадра искаженного изображения зависит от яркости объектов, расположенных вне кадра, за счет свертки исходного изображения с ΦPT . При восстановлении изображений из-за неполной информации о сигнале вне кадра возникают краевые эффекты. Влияние краевых эффектов на качество восстановления в ряде случаев оказывается даже более существенным, чем зашумленность изображения.
3. При искажениях, вызванных движением или расфокусировкой камеры, передаточные функции имеют нули, наличие которых обусловлено

осциллирующим характером передаточных функций. Наличие нулей приводит к полной утрате данных об исходном изображении на соответствующих частотах. По этой причине не удастся абсолютно точно восстановить исходное изображение по наблюдаемому изображению, даже если отсутствуют шумы наблюдения и размеры кадров неограниченны.

При решении задач восстановления изображений используются различные алгоритмы, как имеющие строгое математическое обоснование, так и эмпирические. Для искажений, описываемых уравнением свертки, эти алгоритмы условно можно разделить на три основные группы: алгоритмы решения системы алгебраических уравнений, алгоритмы фильтрации изображений в частотной области и итерационные алгоритмы.

ЛИТЕРАТУРА:

1. Абламейко С.В., Лагуновский Д. М. Обработка изображений: технология, методы, применение.— Минск: Амалфея, 2000 .— 303с.
2. М. Борн, Э. Вольф Основы оптики. М.: Наука, 1970.- 856 с.
3. Гонсалес Р., Вудс Р. Цифровая обработка изображений.; пер с англ. под ред. П. А. Чочиа .— М.: Техносфера, 2006 .— 1070 с.
4. Даджион Д., Мерсеро Р. Цифровая обработка многомерных сигналов. -М. Мир, 1988. с. 15-141.
5. Климов А. С. Форматы графических файлов.— [Киев: ДиаСофт, 1995] .— 479 с.
6. Ландсберг Г.С. Оптика М., Физматлит, 2003
7. Методы компьютерной оптики. Под. ред. В.А.Сойфера М., 2000
8. Нейман В. И. Цифровая обработка изображений в современном приборостроении / / Приборы. Ежемесячный научно-технический, производственный и справочный журнал.— М. .— 2007 .— № 2 .— С. 57-63.
9. Рабинер Л. Р., Гоулд Б. Теория и применение цифровой обработки сигналов. -М. Мир, 1978. с. 62-72, 394-421.
10. Родионов С. А. Автоматизация проектирования оптических систем. -Л. Машиностроение, Ленингр. отд-ние, 1982. -270 с.
11. Ярославский Л.П. Введение в цифровую обработку изображений. – М.: Сов. радио, 1979.
12. Ярославский Л.П. Цифровая обработка сигналов в оптике и голографии: Введение в цифровую оптику. – М.: Радио и связь, 1987.

Оглавление

ВВЕДЕНИЕ	3
1. Основы регистрации, моделирования и цифровой обработки оптического изображения	4
1.1. <i>Основные понятия компьютерной обработки изображений</i>	4
1.1.1. Цифровое изображение	5
1.1.2. Методы обработки изображения	6
1.2. <i>Основные понятия теории сигналов</i>	6
1.2.1. Преобразование сигналов оптическими системами	7
1.2.2. Структурные передаточные характеристики	9
1.2.3. Частотные передаточные характеристики	10
1.3. <i>Дискретизация и квантование непрерывных изображений</i>	11
1.3.1. Дискретизация непрерывных изображений	11
1.3.2. Квантование изображений	13
1.4. <i>Ввод изображений</i>	18
1.4.1. Глубина цвета	20
1.4.2. Квантование цвета	21
1.5. <i>Сжатие изображения</i>	21
1.5.1. Групповое сжатие	22
1.5.2. Метод Хаффмана	23
1.5.3. Метод разностного кодирования (дельта-кодирование)	24
1.5.4. Метод LZW	24
1.5.5. Метод JPEG	25
1.6. <i>Форматы графических файлов</i>	26
1.6.1. BMP	26
1.6.2. TIFF	27
1.6.3. GIF	27
1.6.4. JPEG	27
1.7. <i>Кодирование цветных изображений</i>	27
1.7.1. Определение цветовой системы	27
1.7.2. Цветовое пространство CIE XYZ	29
1.7.3. Хроматические координаты Y_{xy}	31
1.7.4. Lab	32

1.7.5.	RGB.....	33
1.7.6.	Система CMYK	36
1.7.7.	Система HSB.....	38
1.7.8.	Особенности использования цветовых моделей.....	41
1.8.	<i>Форматы хранения изображений.....</i>	41
1.8.1.	Черно-белый	41
1.8.2.	Grayscale(градации серого)	41
1.8.3.	Многоканальные	42
1.8.4.	Индексированный	43
1.9.	<i>Улучшение визуального качества изображений путем поэлементного преобразования.....</i>	43
1.9.1.	Линейное контрастирование изображения.....	44
1.9.2.	Соляризация изображения	46
1.9.3.	Препарирование изображения	48
1.9.4.	Преобразование гистограмм, эквализация	51
2.	Методы и алгоритмы обработки оптического изображения.....	54
2.1.	<i>Геометрические операции над изображением</i>	54
2.2.	<i>Выполнение логических и арифметических операций над изображениями</i>	59
2.2.1.	Выполнение логических операций.....	59
2.2.2.	Выполнение арифметических операций.....	62
2.3.	<i>Фильтрация изображений</i>	65
2.3.1.	Общие понятия фильтрации изображений.....	65
2.3.2.	Масочная фильтрация изображений	69
2.3.3.	Нелинейная фильтрация	71
2.4.	<i>Восстановление изображений на основе обратной фильтрации.....</i>	75
2.5.	<i>Восстановление изображения на основе фильтрации Винера... </i>	83
2.6.	<i>Итерационные методы восстановления изображений</i>	86
2.7.	<i>Алгебраические методы восстановления изображений</i>	88
	Литература:	91
	Оглавление	92

Ежова Ксения Викторовна
Моделирование и обработка изображений
Учебное пособие

В авторской редакции

Редакционно-издательский отдел НИУ ИТМО

Зав. РИО

Н.Ф. Гусарова

Лицензия ИД № 00408 от 05.11.99

Подписано к печати _____

Заказ № _____

Тираж 50 экз.

Отпечатано на ризографе