

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Емельянов Сергей Геннадьевич  
Должность: ректор  
Дата подписания: 04.11.2024 22:19:45  
Уникальный программный ключ:  
9ba7d3e34c012eba476ffd2d064c65781957ba7304f3374d1667c0ce576f85c6

**МИНОБРНАУКИ РОССИИ**  
Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Юго-Западный государственный университет»  
(ЮЗГУ)

Кафедра биомедицинской инженерии

УТВЕРЖДАЮ  
Проректор по учебной работе  
О.Г. Локтионова  
« 11 09 2023 г.



## ТЕХНОЛОГИИ ОБРАБОТКИ РАСТРОВЫХ ИЗОБРАЖЕНИЙ

Методические указания к самостоятельной работе по дисциплине  
«Интеллектуальные системы классификации и распознавания  
изображений» для студентов направления подготовки 12.04.04  
«Биотехнические системы и технологии»

Курск 2023

УДК 004.93:61

Составители: С.А. Филист

Рецензент

Доктор технических наук, профессор Р.А. Томакова

**Технологии обработки растровых изображений:**  
методические указания к самостоятельной работе / Юго-Зап. гос. ун-  
т; сост.: С.А. Филист. - Курск, 2023. - 66 с.

Предназначено для студентов направления подготовки 12.04.04 «Биотехнические системы и технологии» по дисциплине «Интеллектуальные системы классификации и распознавания изображений». Может быть использована аспирантами, обучающимися по научным специальностям 2.3.1. Системный анализ, управление и обработка информации и 2.2.12. Приборы, системы и изделия медицинского назначения.

Текст печатается в авторской редакции

Подписано в печать . Формат 60×84 1/16. Бумага офсетная.  
Усл. печ. л. 3,8. Уч.-изд. л. 3,5. Тираж 100 экз. Заказ 64 .  
Юго-Западный государственный университет.  
305040, г. Курск, ул. 50 лет Октября, 94.

## САМОСТОЯТЕЛЬНАЯ РАБОТА №1. ИЗУЧЕНИЕ ФОРМАТОВ ФАЙЛОВ ИЗОБРАЖЕНИЙ

**Цель:** познакомиться с различными видами палитры, а также с форматами файлов изображений.

### Задание

1. С помощью команды PrintScreen и графического редактора создать изображения заданным по варианту размером следующих видов:

- цветное полноцветное (фото);
- изображение с несколькими цветами (более 5);
- изображение текста.

2. Сохранить 3 изображения в несжатом bmp-формате и различном цветовом представлении:

- 24 бит/пиксель;
- 256 цветов – полутоновое;
- 256 цветов – цветное;
- 16 цветов;
- бинарное (черно-белое).

3. Для каждого из изображений из п.2 исследовать качество сжатия изображений различными форматами:

- JPEG;
- GIF;
- TIF (CCITT 3 и 4);
- РСХ.

Результаты оформить в виде таблицы и сделать выводы.

Отчет должен содержать титульный лист установленного образца, тему, цель и задание в соответствии с вариантом. В отчете, помимо таблицы, необходимо привести скриншоты изображений п.2, а также, используя информацию из Интернет, - основные характеристики и общее описание применения каждого из перечисленных форматов файлов изображений.

При защите работы знать, что такое растр и пиксель, иметь понятие о дискретизации, выборке и квантовании изображений, иметь представление о различных форматах изображений и используемых в них типах сжатия информации.

### Варианты размеров изображений

1. 320x200; 2. 320x240; 3. 640x200; 4. 640x240; 5. 640x320; 6. 640x480

## Теоретические сведения

### *Растровые графические файлы (bitmaps).*

Растровые изображения напоминают лист клетчатой бумаги, на котором любая клетка закрашена своим цветом, образуя в совокупности рисунок. Каждая такая «клетка» цифрового изображения называется пикселем. Пиксель - основной элемент растровых изображений, наименьший элемент поверхности визуализации (экрана монитора, листа бумаги и т.п), которому независимым образом могут быть присвоены его характеристики: цвет, яркость, прозрачность и т.п.

Цифровое изображение представляет собой совокупность пикселей. Каждый пиксель растрового изображения характеризуется своими координатами  $x$  и  $y$  и яркостью  $Y(x,y)$  (для полутоновых изображений). Каждый пиксель цветного изображения характеризуется координатами  $x$  и  $y$  и тремя яркостями: красного, зеленого синего и цветов  $(R, G, B)(x,y)$ .

Как и любая цифровая информация, цвет любого пикселя растрового изображения хранится в двоичном виде. Для кодирования яркостей принято использовать 1, 2, 4 или 8 бит. Исключение составляют режимы HiColor (см. ниже). Чем больше бит используется для кодирования, тем больше различных цветов можно получить. Для хранения значений яркости в настоящее время отводится не более 1 байт (8 бит, что соответствует 256 возможным градациям). В случае цветного изображения каждый пиксель содержит, максимум, по байту на градации яркостей каждого из трех основных цветов. Принципиально возможно кодирование градаций яркости и большим количеством бит (например, 12 бит), но человеческий глаз в состоянии различить только до 256 градаций на каждый цвет. Более точной передачи цветов может потребовать только специальная аппаратура.

Совокупность пикселей изображения образует растр. Растр – это порядок расположения точек (растровых элементов) на отображающей плоскости. Элементами растра,

как правило, являются прямоугольники, и такой растр называется, соответственно, прямоугольным. Тем не менее, в качестве растровых элементов возможно использование и фигур другой формы; удовлетворяющих следующим требованиям:

- 1) все они должны иметь одинаковые размеры;
- 2) должны полностью покрывать плоскость без перекрытий и пустот.

Помимо прямоугольников, данным условиям удовлетворяют правильные треугольники (рисунок 1.1) или правильные шестиугольники (гексагоны) (рисунок 1.2). Можно строить растры, используя неправильные многоугольники, но практический смысл в подобных растрах отсутствует. Отметим, что прямоугольный растр удобен для хранения в оперативной памяти компьютера, в то время, как треугольный и шестиугольный растры естественны при отображении на экране монитора.

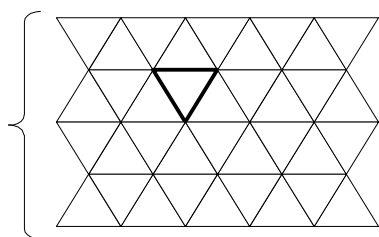


Рисунок 1.1 - Треугольный растр

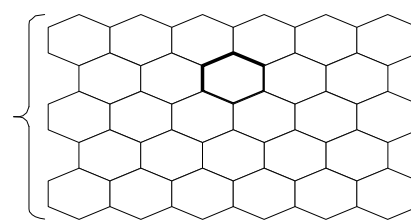


Рисунок 1.2 - Гексагональный растр

### ***Дискретность цвета***

Известно, что цветовое пространство является трехмерным, и произвольный цвет может быть разложен по трем независимым (базисным) компонентам. Эти компоненты могут, в некоторой мере, быть выбраны произвольно. Далее речь будет идти о представлении RGB как наиболее распространенном, поскольку оно непосредственно соответствует формированию цвета на экране монитора. Различные представления могут применяться для взаимодействия компьютерных программ с пользователем и внешними устройствами, но для хранения в памяти компьютера используется именно представление цвета RGB.

Все характеристики, присущие пикселу, дискретны. Дискретными являются и компоненты R, G и B (базисных

цветов: красного, зеленого и синего). Для каждой из них отводится до 1 байта памяти, т.е. до 8 двоичных разрядов. Следовательно, каждая из этих компонент может принимать целые значения в диапазоне от 0 до 255. Поскольку каждая из трех компонент принимает 256 значений, общее количество цветов будет равно 256 в кубе или немногим меньше 17 миллионов (если точно, то  $16\,777\,216 = 2^{24}$ ). При различных манипуляциях в цветовом пространстве удобно применять т.н. приведенные цвета, вещественные значения, принадлежащие отрезку  $[0; 1]$ . В дискретном представлении единице соответствует значение, равное 255.

Заметим, что среди цветов полноцветного представления существуют, в частности, и ненатуральные, т.е. такие, которые нигде не встречаются в природе.

### ***256-цветные изображения***

Изображения требуется не только отображать, но и сохранять в памяти компьютера или на диске. Нередко для хранения одного пикселя используется один байт (8 бит). В этом случае изображение может содержать только 256 цветов. Если их более или менее равномерно распределить по компонентам RGB, то цвета, очевидно, будут передаваться очень грубо. Пусть, например, 8 двоичных разрядов распределены по компонентам так:  $3 + 3 + 2 (=8)$ .

Первая цифра указывает количество разрядов, отводимых для передачи красного цвета, вторая - зеленого, и третья - синего. Синему цвету дается меньше градаций, т.к. человеческий глаз к нему наименее чувствителен. Получаем, что компоненты R и G имеют по 8 градаций яркости, а B и того меньше - всего 4. Это очень мало по сравнению с исходными 256 или даже отображаемыми 64 градациями. Но оказывается, что цвета, доминирующие в изображении, как правило, очень неравномерно распределены по цветовому кубу. Говорят, что изображение выдержано в определенных тонах: голубых, розовых или, скажем, кремовых. Это обстоятельство позволяет применить следующий трюк. Наиболее часто встречающиеся 256 цветов нумеруются, а остальные приводятся к ним. Значения компонент соответствующих RGB хранятся в ячейках специально отведенного массива памяти, называемого **палитрой**. При этом в качестве цвета пикселя записывается номер (индекс

цвета) в палитре, а не соответствующие значения RGB. В связи с этим, любой режим отображения цвета с палитрой часто называют индексированным. А такой 256-цветный режим принято называть MCGA (многоцветный графический режим).

Частным случаем такого 256-цветного MCGA-режима является не цветной Grayscale (полутоновой) режим. В этом случае каждый из 256 цветов в палитре имеет равные значения компонентов ( $R = G = B = Y$ ), которые равны номеру «цвета» (градации яркости  $Y$ ). В этом случае, однако, в отличие от обычных цветов, отображаются все возможные градации яркости.

Заметим, что еще более выгодным оказывается индексированное представление цвета и для режимов, содержащих менее 256 цветов (2, 4 или 16 цветов).

### ***HiColor/RealColor изображения***

Если отводить на каждый пиксель по 2 байта вместо одного, то получится более качественная цветопередача по сравнению с режимом MCGA. В этом случае на три цветовые компоненты приходится 15 или 16 двоичных разрядов. В первом случае для каждой из компонент RGB отводится по 5 бит. Во втором случае, равномерно распределить память для хранения компонент не удастся. Но, зная, что человеческий глаз наиболее чувствителен к зеленому цвету, ему отдается больше разрядов, чем красному или синему. Таким образом, распределение памяти в битах по компонентам RGB: 5-6-5. Компонента G имеет 6 бит (64 возможных значения яркости), а R и B - по 5 бит (по 32 градации). Общее количество цветов в этом режиме составит  $32 * 64 * 32 = 2^{16} = 65536$ . Для цветопередачи используется только 6 бит каждой компоненты, поэтому зеленый цвет передается полностью, а красный и синий - почти полностью. Отсюда и название такого режима – RealColor (режим настоящих цветов).

Заметим, что хранение палитры для этих режимов не оправдано, поэтому здесь пиксель уже непосредственно хранит само значение цвета.

### ***TrueColor изображения***

Режимом с наиболее качественной цветопередачей является тот, в котором на каждый пиксель приходится по 3 байта или 24

бита. Такой режим называется TrueColor (истинно цветной режим). Здесь на хранение каждого из базисных цветов отводится по 8 бит. Выше было показано, что такой режим порождает около 17 миллионов цветов.

Из рассмотренных режимов только два имеют возможность хранить все допустимые цвета. Один из них, очевидно, TrueColor, а второй, как отмечалось выше, - Grayscale (полутоновой). Действительно, оттенков серого может быть всего 256, т.к. они определяются как цвета, имеющие одинаковые значения R, G и B, число таких значений равно 256.

### ***Файлы растровой графики Windows формата BMP***

Рассмотрим формат файлов изображений, которые использует популярная операционная система Microsoft Windows. Эти файлы сохраняются в формате DIB (Device Independent Bitmap), не зависящем от внешних устройств, который позволяет системе отображать картинку на устройствах любого типа. Расширение имен таких файлов - .bmp (от слова «bitmap»).

Изображение в формате .bmp логически состоит из четырех частей:

BITMAPFILEHEADER bmfh; // длина = 14 байт

BITMAPINFOHEADER bmih; // длина = 40 байт

RGBQUAD aColors[]; // переменной длины или отсутствует

BYTE aBitmapBits[]; // переменной длины

Первые две части представляют собой заголовки, соответственно, файла и изображения, третья, необязательная, часть – информация о палитре. Четвертая часть - это собственно изображение, сохраненное слева направо и **снизу вверх**. Хочется еще раз подчеркнуть последнее обстоятельство: порядок следования строк (снизу вверх) **обратный** порядку следования строк в **видеопамяти**.

Четвертая часть файла bmp - это массив пикселей изображения, записанных построчно. Пиксели в каждой строке записываются слева направо, а строки изображения сохраняются снизу вверх. Отметим, что данные каждой строки пикселей выравниваются на границу в 4 байта.

Начиная с версии 3.0 Windows допускает сжатие растровых изображений в файлах .bmp, с использованием простого метода группового кодирования (RLE - run-length encoding). Смысл



метода группового кодирования состоит в том, что группы одноцветных пикселей кодируются двумя байтами: количество повторений ( $n$ ) и значение цвета. При отображении упакованного изображения надо просто  $n$  раз повторить пиксель с указанным цветом. Сжатие позволяет сэкономить место на диске, необходимую для хранения изображения. Чем больше в изображении участков одного цвета, и чем больше размеры этих участков, тем эффективнее работает RLE-сжатие изображения. Сжатие в файлах BMP может быть применено для форматов с восемью или четырьмя битами на пиксел, т.е. для 256-ти и 16-цветных изображений. Бинарные и полноцветные (TrueColor) изображения не могут быть сжаты.

### ***Файлы формата PCX***

Формат файлов изображений PCX был разработан в 1982 Корпорацией ZSoft для их программы Paintbrush, но фактически стал промышленным стандартом для сохранения и пересылки растровых изображений на ПК с MS-DOS. Современная версия этого формата может поддерживать дисплеи любой разрешающей способности, использующие палитры с любым количеством цветов, и он очень прост в применении. Сегодня, практически все коммерческие программы поддерживают формат PCX.

В формате PCX также использован метод группового кодирования (RLE - run-length encoding). Групповое кодирование, применяемое в файлах PCX, несколько отличается от кодирования в файлах BMP, хотя сама идея группового кодирования та же: вместо последовательности одинаковых байтов сохранять их количество и значение.

Файл PCX может иметь два или три раздела: 128-байтный заголовок, закодированные данные изображения (этот раздел может иметь произвольную длину) и необязательную палитру. Эта палитра присоединяется к файлу только в том случае, когда изображение содержит 256 цветов. Ее длина равна 769 байт ( $256*3+1$ ).

Групповое кодирование, применяемое в файлах PCX, несколько отличается от рассмотренного для файлов BMP. Изображение просматривается построчно слева направо и сверху вниз. Группы пикселей одного цвета записываются двумя байтами: первый байт содержит количество повторений и второй

- значение цвета. В отличие от формата BMP, сжатию могут быть подвергнуты также и полноцветные (true color) изображения. Они представляются тремя цветовыми плоскостями, и каждая строка развертки сохраняется последовательно по цветовым плоскостям.

Никаких команд, подобных тем, что предусмотрены в стандарте BMP, здесь нет, так что кодирование данных в стандарте РСХ проще.

### ***Файлы формата JPG***

Рассмотрим структуру файлов JPEG. В рассмотренных нами файлах старых форматов заголовки строго фиксированы. Для обеспечения возможности будущих расширений в них вводятся пустые места, которых, тем не менее, не всегда хватает. Посмотрите, например, на заголовок файла формата РСХ. В нем было предусмотрено пустое место для расширений, но, когда понадобилось такое расширение сделать - добавить палитру для 256-цветных изображений, этого места не хватило. Пришлось записывать палитру в конце файла, а пустое место так и осталось "для дальнейших расширений".

Все данные, которые записаны в файл, включая и заголовок представляются в виде отдельных записей или сегментов. Каждый сегмент имеет особый признак - маркер или ТЭГ (от английского tag - ярлык). В сегменте определяется также его длина и содержатся специфические для него данные. Новые сегменты для обеспечения "дальнейших расширений" могут добавляться очень легко. Понадобилось записать в файл что-нибудь новое - определи новый маркер и пиши на здоровье. Программы, которые эту информацию используют, смогут ее найти по введенному вами маркеру, а те, которым она не нужна, просто пропустят этот сегмент, т.к. они не знают этого маркера. Никаких пустых резервных полей при этом не нужно.

Именно так строятся файлы JPEG. Каждый из них содержит несколько сегментов заголовка и один или несколько сегментов данных изображений.

### ***Файлы формата GIF***

Разработан фирмой CompuServe для передачи растровых изображений по сетям. Он использует LZW-компрессию, что

позволяет хорошо сжимать файлы, в которых много однородных заливок (логотипы, надписи, схемы). GIF-формат позволяет записывать изображение "через строчку" (Interlaced), благодаря чему, имея только часть файла, можно увидеть изображение целиком, но с меньшим разрешением. Эта возможность широко применяется в Интернете. Сначала вы видите картинку с грубым разрешением, затем, по мере поступления новых данных, ее качество улучшается. В GIF'e можно назначить один или более цветов прозрачными, они станут невидимыми в web-браузерах и некоторых других программах. Эта прозрачность цветов обеспечивается за счет дополнительного alpha-канала, сохраняемого вместе с файлом.

Кроме того, в файле формата GIF может содержаться не одна, а несколько растровых картинок (кадров), которые web-браузеры могут отображать одну за другой с указанной в файле частотой. Такая разновидность файлов называется анимированными GIF.

Основное ограничение формата GIF состоит в том, что цветное изображение может быть содержать не более 256 цветов.

### ***Файлы формата TIFF***

Изначально данный формат был предназначен для хранения сканированных изображений. Аппаратно независим. На сегодняшний день формат TIFF является одним из самых распространенных и надежных, его поддерживают практически все программы на PC и Macintosh, так или иначе связанные с графикой. TIFF является лучшим выбором при импорте растровой графики в векторную и в форматы издательских систем. Ему доступен весь диапазон цветовых моделей от монохромной до RGB, CMYK и дополнительных цветов Pantone. TIFF может сохранять векторы Photoshop'a, alpha-каналы для создания масок в видеоклипах Adobe Premiere и ряд дополнительных данных. Наибольшие проблемы обычно вызывает LZW-сжатие, иногда применяемое при сохранении в формате TIFF. Некоторые программы (QuarkXPress 3.x, Adobe Streamline и др.) отказываются читать такие файлы. Кроме того, они дольше выводятся на принтеры и фотонаборные автоматы. Выигрыш во времени вывода получается только, если файл сжат в 3-4 раза. Обычно же, при сохранении изображений в данном формате, используется метод сжатия CCITT (International

Telegraph and Telephone Committie), который был разработан для передачи и приема факсов. Он является частным случаем метода сжатия Хаффмана. CCITT Group 3 идентичен формату факсимильных сообщений, а CCITT Group 4 - формату факсов, без специальной управляющей информации.

Более подробную информацию, в т.ч. и техническую, о форматах файлов изображений можно отыскать на сайте <http://devel.archefire.org/mirrors/www.wotsit.org>.

## САМОСТОЯТЕЛЬНАЯ РАБОТА №2. ЦВЕТОВЫЕ ПРЕДСТАВЛЕНИЯ ИЗОБРАЖЕНИЙ. ЦВЕТОВОЙ КУБ

**Цель:** изучить цветовые представления изображений CMY, YIQ, HSL, изучить цветовой куб.

### Задание

1. Написать программу загрузки \*.bmp файлов и вывода изображения на экран.
2. Преобразовать загруженное цветное изображение в:
  - CMY формат;
  - YIQ формат;
  - полутоновое.
3. Решить задачу: осуществить перевод  $RGB_{ВХ} \rightarrow RGB_{ВЫХ}$  по заданному условию. Использовать положение цветов в цветовом кубе.

### *Варианты условий задачи п.3.*

1.  $H=S=const, L \rightarrow max$
2.  $H \rightarrow (H/256) * [H/(H/256)], L=const, S \rightarrow S/4$
3.  $H=L, S=const$
4.  $L \rightarrow max, S=H$
5.  $S \rightarrow S/2, H=const, L \rightarrow max$
6.  $L \rightarrow (L/256) * [L/(L/256)], S \rightarrow S/2, H=const$
7.  $C=Y, M \rightarrow max$
8.  $C=const, Y \rightarrow max, M \rightarrow M/2$
9.  $Y=M, C=const, M=const$
10.  $C=(M+Y)/2, Y=M, M \rightarrow max$
11.  $M \rightarrow |C-M|, C=const, Y \rightarrow Y/2$
12.  $C=M, M=Y, Y=C/2$

### *Оформление отчета*

Перечень обязательных элементов отчета: 1) титульный лист; 2) тема, цель, задание согласно варианту, исходное изображение; 3) краткая теоретическая справка по формулам перевода изображений в заданные форматы, а также по индивидуальному преобразованию изображения.

- 4) экранные формы (4 преобразованных изображения);
- 5) выводы;
- 6) листинг программы с комментариями.

При защите работы необходимо иметь понятие о цветовом кубе (представлениях RGB CMY, CMYK), о цветовом

представлении HSL, цветоразностном YIQ и о преобразовании цветных изображений в полутоновые.

## Теоретические сведения

### *Дискретное представление цветов. Цветовой куб*

Монитор компьютера имеет три независимых электронных луча, каждый из которых высвечивает на экране точки своего цвета - красного, зеленого или синего. Все другие цвета, включая белый, получаются смешением этих *основных* (или *базисных*) цветов в различных пропорциях. Поскольку монитор работает с тремя лучами, задаются интенсивности каждого из них. Яркость выводимой точки определится суммой интенсивностей лучей, соответствующих базисным цветам. Такое представление цвета и яркости пиксела носит название *RGB представление* от английских названий основных цветов: Red - красный, Green - зеленый, Blue - синий. Белый цвет получается сложением всех трех основных цветов с максимальной интенсивностью, черный - полным их выключением. Таким образом, RGB представление является аддитивным:

$$Y = r * R + g * G + b * B.$$

Если отложить компоненты R, G и B на осях декартовой системы координат, то в результате получится так называемый *цветовой куб*:

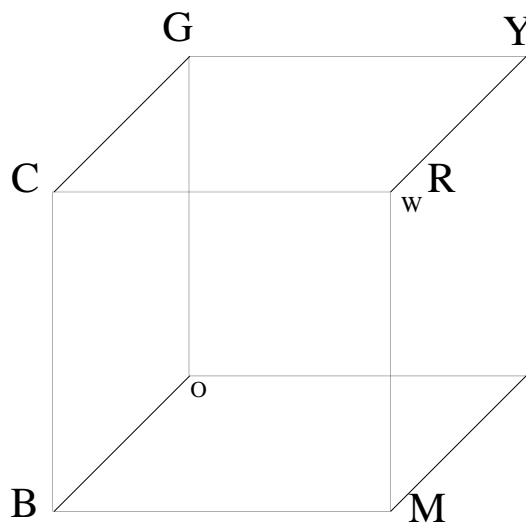


Рисунок 2.1 - Цветовой куб

Начало координат (точка "О") соответствует черному цвету, "W" - белому. Другие вершины куба (кроме уже известных R, G и B) – это "Y" - желтый, "C" - бирюзовый, "M" - сиреневый. Представление цветов RGB естественно для устройств типа монитора, на которых точки светятся. На печатающих устройствах цвет формируется в результате отражения, и смешение красок трех основных цветов даст вовсе не белый цвет. Для таких устройств используются не основные, а *дополнительные* к ним цвета: бирюзовый (Cyan), сиреневый (Magenta) и желтый (Yellow). Соответственно, такое представление цвета и яркости пиксела носит название *СМУ представление*. На цветовом кубе дополнительные цвета представлены вершинами, противоположными основным цветам. Бирюзовый (С) дополнителен к красному (R), сиреневый (M) - к зеленому (G) и желтый (Y) - к синему (B). При сложении любого основного и дополнительного к нему цветов получается белый цвет. Белый цвет в модели СМУ соответствует отсутствию всех красок. Действительно, лист бумаги, на котором ничего не нарисовано, остается белым. По аналогии с RGB представлением, логично было бы предположить, что сложение всех трех дополнительных цветов с максимальной интенсивностью даст черный цвет, но это не совсем так. Можете проверить, смешав бирюзовую, сиреневую и желтую краски на листе бумаги. Поэтому в представление СМУ добавляется еще одна, черная, компонента, и соответствующее цветовое пространство имеет аббревиатуру СМУК (Cyan, Magenta, Yellow, black). Преобразование цветовых компонент из RGB в СМУК и обратно очень просто (формулы приводятся для приведенных значений компонент цветовых пространств):

**RGB ® СМУК:**

$$C = (1-R-K)/(1-K);$$

$$M = (1-G-K)/(1-K);$$

$$Y = (1-B-K)/(1-K);$$

$$K = \min(1-R, 1-G, 1-B).$$

**CMYK ® RGB:**

$$R = 1 - \min(1, C * (1 - K) + K);$$

$$G = 1 - \min(1, M * (1 - K) + K);$$

$$B = 1 - \min(1, Y * (1 - K) + K).$$

***Представления, основанные на цветности***

Типичное представление цвета, основанное на цветности - это *HSL* представление. Аббревиатура HSL образована от английских слов Hue - цвет или оттенок, Saturation - насыщенность и Luminosity - яркость. Если RGB цветовой куб представляет собой декартову систему координат, то HSL и другие ему подобные представления приводят к цилиндрической системе координат. При этом ось симметрии (называемая аппликатой) располагается по пространственной диагонали O-W цветового куба и представляет собой ось яркости (L), насыщенность (S) - это расстояние от центральной оси или радиус, цвет (H) - угол, отсчитываемый от плоскости ORW.

Расположим цветовой куб так, чтобы пространственная диагональ OW (идущая от черного угла к белому) проходила снизу вверх, и посмотрим на него сверху, со стороны точки W (см. рис. 2.1, на котором представлена проекция цветового куба на плоскость, перпендикулярную оси OW и проходящую через вершины R, G и B). Мы увидим правильный шестиугольник (рисунок 2.2).

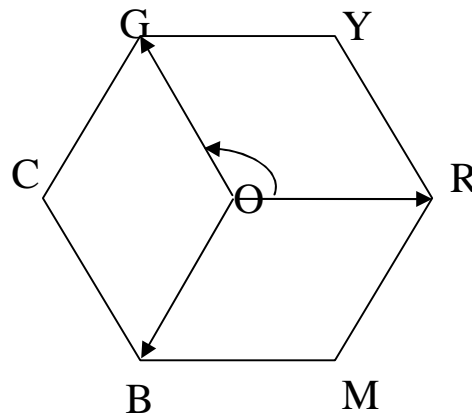


Рисунок 2.2 - Проекция цветового куба на плоскость, перпендикулярную оси OW



Точка O соответствует оси OW; R, G и B - вершины куба, соответствующие основным цветам. Проекция на эту плоскость ребра OR соответствует углу H (цвету) равному 0 градусов, проекция OG – 120 градусов, проекция OB - 240 градусов.

Предполагая, что ребра цветового куба имеют единичную длину, т.е. значения R, G и B лежат в диапазоне от 0 до 1, получим следующие формулы преобразования:

|   |
|---|
| <p>Цвет (Hue):</p> $H = \frac{1}{2\pi} \left( \text{Alpha} - \arctg \frac{2R - G - B}{\sqrt{3}(G - B)} \right);$ <p>Насыщенность (Saturation):</p> $S = \sqrt{R^2 + G^2 + B^2 - RG - RB - BG};$ <p>Яркость (Luminosity)</p> $L = \frac{R + G + B}{3};$ <p>где</p> $\text{Alpha} = \begin{cases} \frac{\pi}{2}, G \geq B; \\ \frac{3\pi}{2}, G < B. \end{cases}$ |
|---|

Значения H, S и L также лежат в диапазоне от 0 до 1.

В телевидении часто используется цветоразностная модель YIQ. Перевод из системы RGB в YIQ производится по формулам:

|   |
|---|
| $Y = 0.30R + 0.59G + 0.11B$ $I = 0.60R - 0.28G - 0.32B$ $Q = 0.21R - 0.52G + 0.31B$ |
|---|

### ***Преобразование цветных изображений в полутоновые***

В ряде задач цвет объектов на изображении не имеет значения, а все определяется их яркостью. Из этого следует

необходимость преобразования цветных изображений в полутоновые, в которых уровень серого соответствовал бы яркости.

Некоторые программы для преобразования цветного изображения в серое использовали формулу:

$$Y = 0.299 * R + 0.587 * G + 0.114 * B,$$

или близкое к ней целочисленное представление:

$$Y = (3 * R + 5 * G + 1 * B) / 9.$$

Сейчас принято считать, что более точное выражение имеет вид:

$$Y = 0.222 * R + 0.707 * G + 0.071 * B.$$

Для того, чтобы получить полутоновое изображение, необходимо каждой из цветовых компонент присвоить найденное значение яркости:

$$R = G = B = Y.$$

Значения весов в приведенных формулах преобразования выбираются так, чтобы максимально соответствовать кривой спектральной чувствительности человеческого глаза.

## САМОСТОЯТЕЛЬНАЯ РАБОТА №3. ИНТЕГРАЛЬНЫЕ СТАТИСТИЧЕСКИЕ ХАРАКТЕРИСТИКИ ИЗОБРАЖЕНИЙ

**Цель:** изучить статистические характеристики распределения яркости и научиться строить гистограмму распределения яркостей на изображении.

### Задание

**Написать программу, выполняющую следующие действия:**

1. Загрузка изображения из файла и вывод его на экран.
2. Преобразование загруженного цветного изображения в полутоновое.
3. Расчет и отображение статистических характеристик распределения яркости: а) среднего значения;  
б) СКО;  
в) коэффициента асимметрии;  
г) коэффициента эксцесса;  
д) моды.
4. Построение гистограммы распределения яркости и отображение ее на экране.

*Примечание. На осях гистограммы отметить значения интервалов.*

### ***Оформление отчета***

- Перечень обязательных элементов отчета: 1) титульный лист;
- 2) тема, цель, задание, исходное изображение;
  - 3) расчетные формулы статистических характеристик;
  - 4) экранные формы: полутоновое изображение, рассчитанные статистические характеристики, гистограмма распределения яркости.
  - 5) выводы;
  - 6) приложение: листинг программы с комментариями.

При защите работы знать следующие теоретические положения:

1. Статистические характеристики изображений, формулы их расчета.

2. Каков смысл каждой из статистических характеристик.
3. Для чего нужна гистограмма изображения и каким образом она строится.

## Теоретические сведения

### *Статистические характеристики*

Яркость полутонового изображения, вследствие случайного характера отображаемых элементов сцены, наличия шумов и т.п., может рассматриваться как случайная величина. Случайные величины могут быть дискретными или непрерывными. Дискретная случайная величина может принимать значения из некоторого счетного множества. Эти значения отделены друг от друга, и их можно перенумеровать. Это, например, целые числа. Возможные значения непрерывной случайной величины заполняет некоторый заданный промежуток. При определении статистических характеристик дискретной случайной величины используется суммирование, которое в случае непрерывной случайной величины заменяется интегрированием. Яркость пикселей изображения принимает дискретные значения, поэтому в дальнейшем будем использовать суммы.

### *Среднее значение*

Среднее значение или *математическое ожидание* яркости в некоторой области определяется суммой значений яркости во всех точках этой области, деленной на количество точек:

$$M(I) = \frac{1}{N} \sum_{(x,y)} I(x, y) . \quad (3.1)$$

### *Дисперсия*

*Дисперсия* яркости определяется как математическое ожидание квадрата отклонения яркости от его среднего значения:

$$D(I) = \frac{1}{N} \sum_{(x,y)} (I(x, y) - M(I))^2 . \quad (3.2)$$

Размерность дисперсии - квадрат яркости. Это не всегда удобно. Поэтому, часто используют квадратный корень из дисперсии, называемый *среднеквадратичным отклонением (СКО)*:

$$\hat{\sigma}(I) = \sqrt{D(I)}, \quad (3.3)$$

которое имеет размерность яркости.

Дисперсия и среднеквадратичное отклонение являются мерой рассеяния яркости вокруг среднего значения.

### ***Моменты яркости***

Центральный статистический момент яркости  $r$ -го порядка определяется, как математическое ожидание возведенного в степень  $r$  отклонения яркости от его среднего значения:

$$M^r(I) = \frac{1}{N} \sum_{(x,y)} (I(x, y) - M(I))^r. \quad (3.4)$$

Заметим, что дисперсия - это второй центральный момент. Чаще других моментов высших порядков используются третий и четвертый, но, как правило, не сами по себе, а в виде коэффициентов *асимметрии*:

$$As(I) = \frac{M_3(I)}{\hat{\sigma}^3(I)}, \quad (3.5)$$

и *эксцесса*:

$$Ex(I) = \frac{M_4(I)}{\hat{\sigma}^4(I)} - 3. \quad (3.6)$$

Коэффициент асимметрии, как это понятно из названия, характеризует асимметрию распределения случайной величины (в нашем случае - яркости). Если он положителен, распределение яркости более пологое в области больших значений, если отрицателен - в области меньших.

Коэффициент эксцесса характеризует отличие формы распределения от нормального закона, о котором речь пойдет ниже. Чем он больше, тем более острую вершину имеет распределение, и наоборот. Нулевое значение коэффициента эксцесса соответствует нормальному распределению. Чтобы обеспечить такое его свойство, в формуле (3.6) и вычитается тройка.

### ***Мода***

Еще один простой, но важный параметр случайной величины носит название *мода*. Математически, мода - это значение случайной величины, которое она принимает с максимальной вероятностью. Однако, здесь под модой практически понимается локальный максимум функции распределения яркости. Как известно, количество локальных максимумов в общем случае не ограничено. Соответственно, распределения, имеющие один, два или более максимумов, называются *унимодальными, бимодальными или полимодальными*.

### ***Гистограмма***

Для того чтобы построить график функции распределения, нужно знать аналитическое выражение закона распределения случайной величины и его параметры. Если мы имеем дело с некоторым полутоновым изображением, мы не знаем точно ни закона распределения яркости, ни его параметров. Тем не менее, в этом случае удастся получить статистическую оценку плотности распределения, не зависящую ни от вида закона, ни от параметров. Такой оценкой плотности распределения является так называемая *гистограмма*, представляющая собой графическое изображение рассчитанной по реальным данным плотности распределения случайной величины.

Для того, чтобы построить гистограмму полутонового изображения, вся область допустимых значений яркости разделяется на интервалы, и рассчитываются относительные частоты попадания яркости пикселей изображения в каждый интервал. Затем строится график, на котором по горизонтальной оси откладываются интервалы, и на каждом из них, как на основании, строится прямоугольник, площадь которого равна вероятности попадания яркости в данный интервал.

Чем меньшей выбирается ширина интервала, тем более подробной получается гистограмма. С другой стороны, если на интервал приходится мало точек, достоверность оценки плотности распределения для этого интервала будет мала. Кроме того, изображения часто содержат не все допустимые уровни яркости, а с пропусками. Например, у изображения есть пиксели с яркостями, равными 0, 4, 8, 16, 32 и т.д., а с промежуточными (1,2, 3, 5, 6, 7, ...) нет. В этом случае гистограмма, построенная с интервалом шириной 1 или 2 уровня яркости не будет показательной.

## САМОСТОЯТЕЛЬНАЯ РАБОТА № 4. ПОЭЛЕМЕНТНЫЕ ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЙ. ПРЕОБРАЗОВАНИЕ ГИСТОГРАММ

**Цель:** изучить наиболее распространенные процедуры поэлементной обработки изображения; научиться определять итоговую плотность распределения яркости.

### Задание

#### 1. На компьютере.

1. Реализовать алгоритм, осуществляющий линейное контрастирование с параметрически задаваемым выходным динамическим диапазоном.

2. Выполнить соляризацию изображения, расширив динамический диапазон до 0-255 по индивидуальному варианту из таблицы 1.1, где выбор вариантов осуществляется по формуле:

$(N \bmod 3)+1$ , где N – номер по журналу.

3. Реализовать процедуру препарирования по индивидуальному варианту из таблицы 1.2, где вариант вычисляется по формуле:  $(N \bmod 25)+1$ , где N – номер по журналу.

4. Реализовать алгоритм эквализации и гиперболизации гистограммы изображения. Использовать табличный метод поэлементных преобразований.

*Примечание:* преобразовывать полутоновое изображение.

#### 2. Аналитически.

Определить итоговую плотность распределения яркости, если известна начальная и функция поэлементного преобразования.

Начальная плотность – равномерное распределение на интервале 0-255. Функция поэлементного преобразования задается формулой:

$$y = \frac{1}{255} x^2.$$



## **Оформление отчета**

Перечень обязательных элементов отчета:

- 1) титульный лист;
- 2) тема, цель, задание согласно варианту, исходное изображение;
- 3) расчетные формулы заданных поэлементных преобразований;
- 4) аналитические вычисления плотности распределения яркости;
- 5) экранные формы всех преобразований (5 преобразованных изображений);
- 6) выводы (сравнение исходного полутонового изображения с каждым изображением после поэлементных преобразований);
- 7) листинг программы с комментариями.

При защите работы знать следующие теоретические положения:

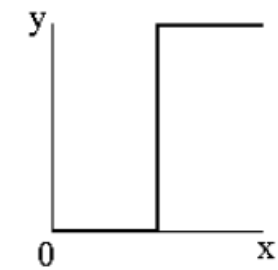
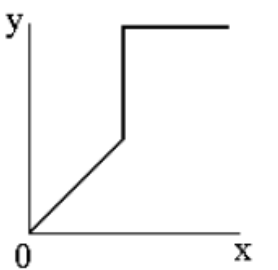
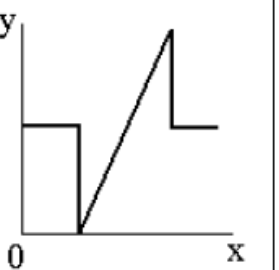
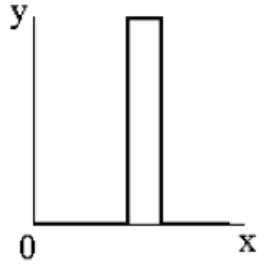
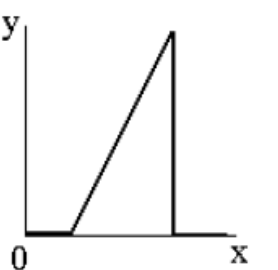
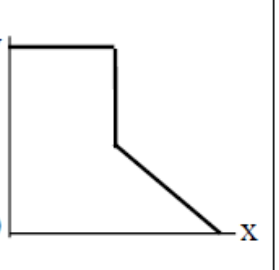
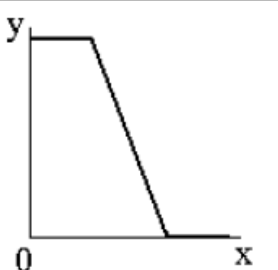
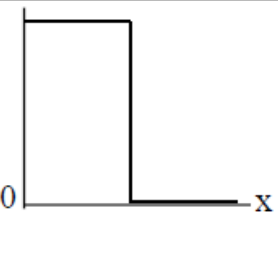
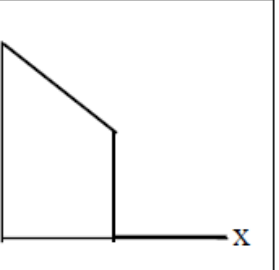
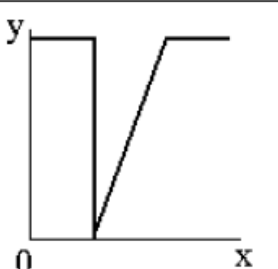
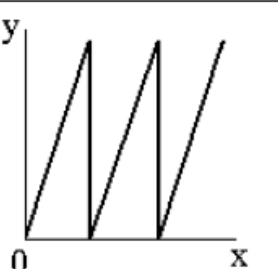
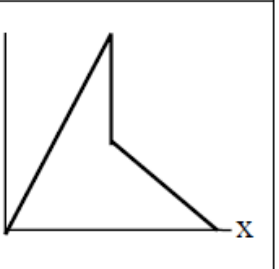
1. Распространенные процедуры поэлементной обработки изображения.
2. Линейное контрастирование.
3. Соляризация изображения.
4. Препарирование изображения.
5. Преобразование гистограмм. Эквиализация и гиперболизация.

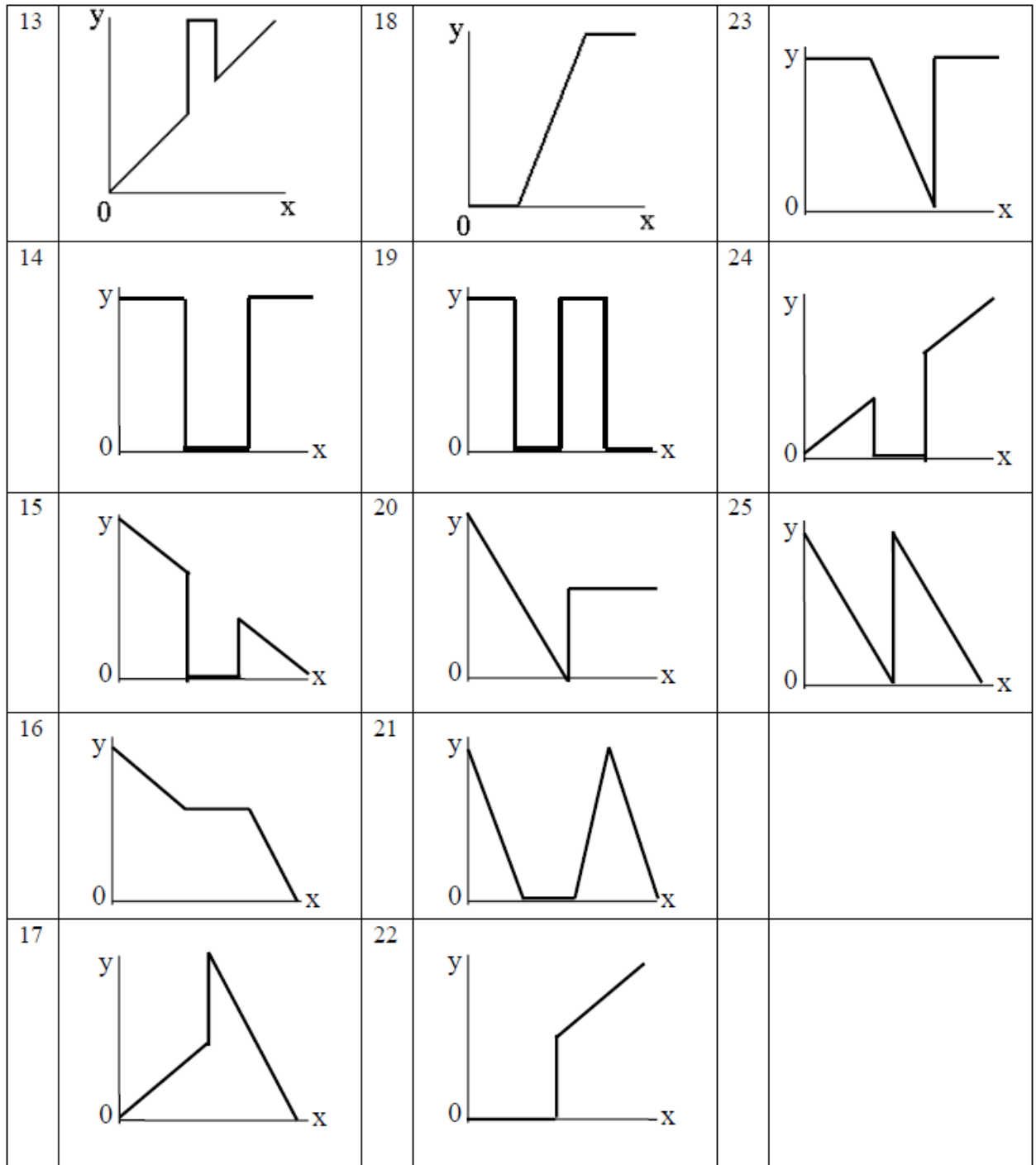
Таблица 4.1 – Варианты для соляризации

| <b>№ варианта</b> | <b>Задание</b>   |
|-------------------|--|
| 1                 | Преобразование вида $y = k \oplus x \oplus (x_{\max} - x)$ |
| 2                 | Преобразование с использованием линейного контрастирования |
| 3                 | Преобразование с использованием гамма коррекции            |

Варианты заданий

Таблица 4.2 – Варианты для препарирования

| № | Процедура препарирования  | № | Процедура препарирования   | №  | Процедура препарирования  |
|---|---|---|--|----|---|
| 1 |    | 5 |    | 9  |    |
| 2 |    | 6 |    | 10 |    |
| 3 |   | 7 |   | 11 |   |
| 4 |  | 8 |  | 12 |  |



### Теоретические сведения

#### *Улучшение визуального качества изображений путем поэлементных преобразований*

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

$$y_{i,j} = f_{i,j}(x_{i,j}), \quad (4.1)$$

позволяющая по значению исходного сигнала определить значение выходного продукта. В общем случае, как это учтено в данном выражении, вид или параметры функции  $f_{i,j}(\oplus)$ , 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 (4.1) могут отсутствовать. При этом зависимость между яркостями исходного и обработанного изображений описывается функцией:

$$y = f(x), \quad (4.2)$$

одинаковой для всех точек кадра.

### ***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 рисунке 4.1 а, где диапазон яркостей имеет границы  $x_{\min} = 180$ ,  $x_{\max} = 240$ .

При линейном контрастировании используется линейное поэлементное преобразование вида:

$$y = a \cdot x + b, \quad (4.3)$$

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

$$\begin{cases} y_{\min} = a \cdot x_{\min} + b \\ y_{\max} = a \cdot x_{\max} + b \end{cases},$$

относительно параметров преобразования  $a$  и  $b$ , нетрудно привести (4.3) к виду:

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

где  $y$  – выходное значение яркости пиксела,  $x$  – входное (текущее) значение яркости пиксела.

Результат линейного контрастирования исходного изображения, представленного на рисунке 4.1,а, приведен на рис.4.1,б при  $y_{\min} = 0$  и  $y_{\max} = 255$ .



а)



б)

Рисунок 4.1 - 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.

### *Соляризация изображения*

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

Преобразование имеет вид:

$$y = -(x - x_{\min}) \cdot (x - x_{\max}) \cdot k - y_{\min},$$

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

$$k = 4 \frac{y_{\max} - y_{\min}}{(x_{\max} - x_{\min})^2}$$

Функция, описывающая данное преобразование, является квадратичной параболой. 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 рисунке 4.2.



Рисунок 4.2 - Пример соляризации

На рисунке 4.2,а показано исходное изображение, а на рисунке 4.2,б - результат его соляризации. На втором этапе обработки здесь применено линейное контрастирование при  $u_{\min} = 0$  и  $u_{\max} = 255$ . Обработка приводит к повышению четкости деталей изображения: улучшены изображения глаз, повышен контраст на переходе “лицо - волосы” и т. д.

### ***Препарирование изображения***

Препарирование представляет собой целый класс поэлементных преобразований изображений. Например, преобразование с пороговой характеристикой (рис.4.3) превращает полутоновое изображение, содержащее все уровни яркости, в бинарное, точки которого имеют яркости  $y = 0$  или  $y = u_{\max}$ .

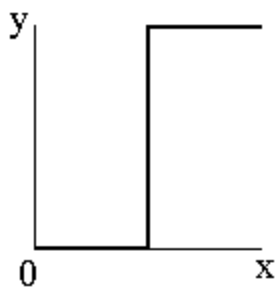


Рисунок 4.3 - Пример преобразования, используемого при препарировании

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.

### *Преобразование гистограмм, эквализация*

При всех поэлементных преобразованиях происходит изменение закона распределения вероятностей, описывающего изображение. Рассмотрим механизм этого изменения на примере произвольного преобразования с монотонной характеристикой, описываемой функцией  $y = f(x)$  (рис. 4.4), имеющей однозначную обратную функцию  $x = \varphi(y)$ .

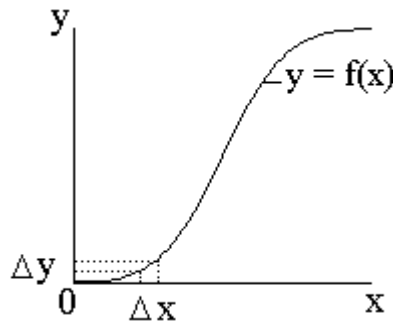


Рисунок 4.4 - Нелинейное преобразование случайной величины

Предположим, что случайная величина  $x$  подчиняется плотности вероятности  $w_x(x)$ . Пусть  $\Delta x$  - произвольный малый интервал значений случайной величины  $x$ , а  $\Delta y$  - соответствующий ему интервал преобразованной случайной величины  $y$ . Попадание величины  $x$  в интервал  $\Delta x$  влечет за собой попадание величины  $y$  в интервал  $\Delta y$ , что означает вероятностную эквивалентность этих двух событий. Поэтому, учитывая малость обоих интервалов, можно записать приближенное равенство:

$$w_x(x)|\Delta x| \approx w_y(y)|\Delta y|,$$



где модули учитывают зависимость вероятностей от абсолютных длин интервалов (и независимость от знаков приращений  $\Delta x$  и  $\Delta y$ ). Вычисляя отсюда плотность вероятности преобразованной величины, подставляя вместо  $x$  его выражение через обратную функцию и выполняя предельный переход при  $\Delta x \rightarrow 0$  (и, следовательно,  $\Delta y \rightarrow 0$ ), получаем:

$$w_y(y) = w_x(\varphi(y)) \cdot \left| \frac{d\varphi(y)}{dy} \right|. \quad (4.4)$$

Это выражение позволяет вычислить плотность вероятности продукта преобразования, которая, как видно из него, не совпадает с плотностью распределения исходной случайной величины. Ясно, что существенное влияние на плотность  $w_y(y)$  оказывает выполняемое преобразование, поскольку в (4.4) входит его обратная функция и ее производная.

Определение вероятностных характеристик изображений, прошедших нелинейную обработку, является прямой задачей анализа. При решении практических задач обработки изображений может быть поставлена обратная задача: по известному виду плотности вероятности  $w_x(x)$  и желаемому виду  $w_y(y)$  определить требуемое преобразование  $y = f(x)$ , которому следует подвергнуть исходное изображение.

В практике цифровой обработки изображений часто к полезному результату приводит преобразование изображения к равновероятному распределению. В этом случае

$$w_y(y) = \begin{cases} 1/(y_{\max} - y_{\min}), & \text{при } y_{\min} \leq y \leq y_{\max}, \\ 0, & \text{при других } y, \end{cases} \quad (4.5)$$

где  $y_{\min}$  и  $y_{\max}$  - минимальное и максимальное значения яркости преобразованного изображения.

Определим характеристику преобразователя, решающего данную задачу. Пусть  $x$  и  $y$  связаны функцией (4.5), а  $F_x(x)$  и  $F_y(y)$  - интегральные законы распределения входной и выходной величин. Учитывая (4.5), находим:

$$F_y(y) = \int_{y_{\min}}^y w_y(y) dy = \frac{y - y_{\min}}{y_{\max} - y_{\min}}.$$

Подставляя это выражение в условие вероятностной эквивалентности  $F(x) = F_y(y)$ , после простых преобразований получаем соотношение

$$y = (y_{\max} - y_{\min}) \cdot F_x(x) + y_{\min}, \quad (4.6)$$

представляющее собой характеристику (4.2) в решаемой задаче. В частности, данное преобразование, при котором выходное изображение имеет равномерное распределение, называется *эквализацией* (выравниванием) гистограмм.

Аналогичным образом могут быть получены решения других подобных задач, в которых требуется привести законы распределения изображения к заданному виду. Преобразование, называемое *гиперболизацией* распределения, предполагает приведение плотности вероятности преобразованного изображения к гиперболическому виду:

$$w_y(y) = \begin{cases} \frac{1}{y(\ln y_{\max} - \ln y_{\min})}, & \text{при } y_{\max} \leq y \leq y_{\min}, \\ 0, & \text{при других } y \end{cases} \quad (4.7),$$

Можно показать, что изображение с плотностью вероятности (4.7) получается на выходе нелинейного элемента с характеристикой:

$$y = y_{\min} \cdot \left( \frac{y_{\max}}{y_{\min}} \right)^{F_x(x)}, \quad (4.8)$$

также определяемой интегральным законом распределения исходного изображения.

Таким образом, преобразование плотности вероятности предполагает знание интегрального распределения для

исходного изображения. Как правило, достоверные сведения о нем отсутствуют. Использование для рассматриваемых целей аналитических аппроксимаций также малопригодно, т.к. их небольшие отклонения от истинных распределений могут приводить к существенному отличию результатов от требуемых. Поэтому 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...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. Обозначим эту оценку  $w_x^*(j), 0 \leq j \leq 255$ . Тогда оценка интегрального распределения получается по формуле:

$$F_x^*(j) = \sum_{i=0}^j w_x^*(i).$$

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.

На рисунке 4.5 приведен пример эквализации, выполненной в соответствии с изложенной методикой. Эквализация призвана откорректировать картину, выровняв интегральные площади участков с различными яркостями.

Сравнение исходного (рис. 4.5,а) и обработанного (рис. 4.5,б) изображений показывает, что происходящее при обработке перераспределение яркостей приводит к улучшению визуального восприятия.



а) исходное изображение



б) результат обработки

Рисунок 4.5 - Пример эквализации изображения

## САМОСТОЯТЕЛЬНАЯ РАБОТА № 5. ШУМОПОДАВЛЯЮЩАЯ ФИЛЬТРИЦА ИЗОБРАЖЕНИЙ

**Цель работы:** Изучить виды шума и фильтраций изображения; научиться рассчитывать коэффициенты фильтров.

### Задание

#### 1. На компьютере.

1. Сгенерировать и применить к полутоновому изображению последовательно два вида шума: 1) аддитивный с равномерным распределением, характеризующийся средним 0, СКО 20; 2) импульсный – добавить 100 точек яркостью 255 со случайными равномерно распределенными координатами.

2. Вывести зашумленные изображения в левой части экрана.

3. Обработать изображения заданными фильтрами (см. список в табл.10.1). 4. Вывести результат обработки на правой половине экрана.

#### 2. Аналитически

Рассчитать коэффициенты фильтров  $3 \times 3$ , которые выполняют линейную и квадратическую аппроксимацию области изображения из условия минимума СКО от значений яркости изображения.

### ***Оформление отчета***

Отчет должен содержать:

- 1) титульный лист;
- 2) тема, цель, задание в соответствии с вариантом, исходное изображение;
- 3) теоретические сведения о шумах и заданных фильтрах;
- 4) аналитические вычисления коэффициентов фильтров  $3 \times 3$ ;
- 5) экранные формы зашумленных изображений и отфильтрованных;
- 6) выводы;
- 7) приложение: листинг программы с комментариями.

При защите работы знать следующие теоретические положения:

1. Виды шумов: аддитивный, мультипликативный и импульсный.

2. Линейные алгоритмы (линейный с маской, гауссовский).
3. Ранговые алгоритмы: (медианный, оценка середины диапазона, K ближайших соседей, сигма-фильтр, МТМ фильтр).
4. Локально-адаптивные алгоритмы: (медианный stick фильтр, направленный фильтр, взвешивание их значений обратным градиентом).
5. Алгоритмы, основанные на группировке пикселей (фильтр SUSAN, Leu).

### *Варианты заданий*

Варианты заданий приведены в таблице 5.1.

Таблица 5.1 – Список фильтров

| №   | Фильтры  |
|-----|--|
| 1.  | Линейный с маской:<br>$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$   |
| 2.  | Гауссовский (с t задаваемым с клавиатуры)  |
| 3.  | Медианный фильтр   |
| 4.  | Оценки середины диапазона  |
| 5.  | K ближайших соседей (с K задаваемым с клавиатуры)  |
| 6.  | Сигма-фильтр (с С задаваемым с клавиатуры)   |
| 7.  | МТМ фильтр (с С задаваемым с клавиатуры)   |
| 8.  | Медианный stick фильтр с 4 направлениями:<br>а) выбирать ближайшее к центральному пикселу значение;<br>б) выбирать направление с наименьшим разбросом. |
| 9.  | Направленный фильтр  |
| 10. | Взвешивания обратным градиентом  |
| 11. | Фильтр Лео (Leu)   |
| 12. | SUSAN  |
| 13. | $\begin{matrix} -13 & 2 & 7 & 2 & -13 \\ 2 & 17 & 22 & 17 & 2 \\ 7 & 22 & 27 & 22 & 7 \\ 2 & 17 & 22 & 17 & 2 \\ -13 & 2 & 7 & 2 & -13 \end{matrix}$   |

## Теоретические сведения

### ***Шумы на изображении***

Изображения, введенные в компьютер, редко бывают идеального качества. Чаще всего, кроме собственно изображения они содержат так называемый "шум". Это может быть искажение цвета или яркости некоторых групп пикселей, или же совершенно неправильные значения отдельных пикселей никак не связанные с их истинным цветом.

По способу искажения изображения шум можно разделить на несколько разновидностей:

1) *аддитивный шум* - к значению цвета или яркости каждого пикселя добавляется некоторая случайная величина, значения которой в каждой точке различны;

2) *мультипликативный шум* - значение цвета или яркости каждого пикселя умножается на некоторую случайную величину, значения которой близки к единице и в каждой точке различны. Частным случаем мультипликативного шума является *спекл-шум*, присутствующих на изображениях, полученных в результате отражения когерентного излучения. Спекл-шум присутствует на радиолокационных изображениях и изображениях ультразвуковой диагностики;

3) *импульсный шум* - значения цвета или яркости некоторых отдельных пикселей искажаются настолько, что теряют всякую информацию о их неискаженных значениях.

### ***Линейные алгоритмы***

Наиболее простые алгоритмы подавления шума на изображении - *линейные*. Это значит, что каждый пиксел обработанного изображения получается линейной комбинацией нескольких пикселей исходного изображения. Напомню, что линейная комбинация любых величин представляет собой сумму этих величин, каждая из которых умножена на свой постоянный коэффициент. Таким образом, новое значение пиксела  $J$  определится через старые значения нескольких соседних пикселей  $I$ :

$$J_{ij} = \frac{\sum_{kl} c_{kl} I_{i+k, j+l}}{\sum_{kl} c_{kl}}.$$

Коэффициенты  $c_{kl}$  называются *весами* и не зависят от положения на изображении точки  $(i, j)$  и значений пикселей  $I_{ij}$ . Деление на сумму всех весов необходимо для получения единичного коэффициента передачи, чтобы процедура подавления шума не вызывала смещения средней яркости обработанного изображения.

В *гауссовском фильтре* веса соседних пикселей соответствуют пространственному гауссовскому распределению:

$$c(r, r_0) = e^{-\frac{|r-r_0|^2}{t^2}}, \quad (5.1)$$

где  $r_0$  - центральная точка (та, для которой рассчитывается новое значение);  $r$  - текущая точка;  $c(r, r_0)$  - вес, придаваемый точке  $r$  при расчете значения точки  $r_0$ ;  $|r-r_0|$  - расстояние от точки  $r$  до точки  $r_0$ ;  $t$  - ширина гауссовского распределения.

### ***Ранговые алгоритмы***

Очень много нелинейных фильтров относятся к категории операторов ранговой или порядковой статистики соседей. Это означает, что ближайшие соседи сортируются в порядке возрастания их значений, и яркость пиксела отфильтрованного изображения получается на основе анализа этого упорядоченного списка.

Самый простой оператор порядковой статистики - *медиана*, где в качестве нового значения яркости пиксела используется центральное значение в упорядоченном списке его соседей. *Медианный фильтр* значительно лучше сохраняет правильную структуру границ, чем гауссовское сглаживание, но если граница изогнута, то происходит потеря данных.



Существуют также различные модификации медианного фильтра. Например, *взвешенный медианный фильтр*, который выборочно включает некоторые пиксели (обычно центральные) в упорядоченный список несколько раз, или придает пикселу дополнительный вес, величина которого зависит от расстояния до центральной точки. Фильтр, который дает центральному пикселу более высокий вес, лучше сохраняет углы, но имеет меньший эффект сглаживания.

Еще один ранговый метод – *оценка середины диапазона*. Новое значение яркости пиксела устанавливается равным середине между максимальными и минимальными значениями яркости его соседей.

*Оператор  $K$  ближайших соседей* проводит осреднение только по  $K$  ближайшим по величине соседям к центральному значению из упорядоченного списка. Значение  $K$  обычно устанавливается таким (5 или 6, для маски  $3 \times 3$ ), что углы и тонкие линии сильно разрушаются. Конечно, это делается не специально, но меньшие значения будут плохо бороться с шумом.

*Сигма-фильтр* усредняет значения только тех соседних пикселей, чьи значения находятся в пределах  $C \cdot \sigma$  от значения центрального пиксела в окне, где  $C$  - некоторая константа, дисперсия значений яркости  $\sigma^2$  определяется один раз для всего изображения. Этот метод стремится усреднять пиксел только с теми соседями, которые имеют значения "близкие" к нему, по сравнению со среднеквадратичным отклонением шума изображения. Заметьте, в отличие от двух предыдущих фильтров опорным служит центральный пиксел в окне, а не в упорядоченной выборке. Такой фильтр не сможет устранить импульсный шум.

*Модифицированный обрезающий* (MTM = Modified Trimmed Mean) *фильтр* использует для осреднения только те из соседних пикселей, чьи значения находятся в пределах  $C \cdot \sigma$  от значения центрального в упорядоченном списке пиксела. Этот метод представляет собой нечто среднее между фильтром  $\alpha$ -усеченных средних и сигма-фильтром. С одной стороны, как и в сигма-фильтре, осредняются пиксели, имеющие значения, которые отличаются не более, чем на

$C*\sigma$ , но не от центрального пиксела в окне, а от медианного значения в выборке, как и в фильтре  $\alpha$ -усеченных средних.

Существуют также ранговые фильтры, основанные на так называемой *STICK-технике* (stick - палка, прут). Через каждую точку изображения проводится несколько отрезков прямой под разными углами и вычисляется набор порядковых статистик яркости (либо медиана, либо середина диапазона) для всех отрезков. В качестве нового значения яркости центрального пиксела может выбираться либо ближайшее к центральному пикселу значение, либо то, в направлении которого разброс минимальный. Такие методы способны сохранять линейные детали на изображении.

### ***Локально-адаптивные алгоритмы***

Простейшим локально-адаптивным фильтром является *направленный фильтр*. Маска фильтра строится таким образом, чтобы не допустить усреднения точек в направлении границы. Для четырех различных значений угла  $\varphi$ , образуемого границей с горизонталью, маски имеют вид:

| $\varphi=0^\circ$   | $\varphi=45^\circ$  | $\varphi=90^\circ$  | $\varphi=135^\circ$   |
|---|---|---|---|
| $\begin{bmatrix} 0 & 0 & 0 \\ 1 & 2 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ |

(5.2)

Нормировочный множитель здесь равен  $1/4$ . Полученный фильтр носит название *направленного*, поскольку он сглаживает изображение в направлении параллельном предполагаемой границе.

Часто в локально-адаптивных фильтрах при суммировании пикселов маски используется взвешивание их значений обратным градиентом – *фильтр взвешивания обратным градиентом*. Такое взвешивание препятствует сглаживанию границ, поскольку обратный градиент минимален в направлении поперек границы. В качестве градиента можно использовать разность между значениями центрального и

соседнего с ним пикселей. Тогда сглаженное значение центрального пикселя маски будет равно:

$$J(x, y) = \frac{\sum_w \frac{I(x+i, y+j)}{\max\left(\frac{1}{2}, |I(x+i, y+j) - I(x, y)|\right)}}{\sum_w \frac{1}{\max\left(\frac{1}{2}, |I(x+i, y+j) - I(x, y)|\right)}} \quad (5.3)$$

где  $I(x, y)$  - первоначальное изображение,  $J(x, y)$  – сглаженное изображение, и суммирование производится в квадратной окрестности  $3 \times 3$ . Этот метод хорошо уменьшает гауссовский шум, сохраняя при этом структуру изображения.

#### ***Алгоритмы, основанные на группировке пикселей***

В последнее время активно развиваются методы сглаживания изображения, основанные на группировке пикселей в зависимости от их близости по яркости к центральному.

В методе *LEU* пиксели в окне  $5 \times 5$  разделяются на 3 группы: яркие, темные и примерно равные по яркости центральному. Близкими по яркости к центральному пикселу считаются те точки, чья яркость отличается не более, чем на определенный порог. Величина порога выбирается равной  $1/3$  от разности средних значений 20% самых ярких и 20% самых темных пикселей в окне. Затем подсчитывается количество точек в группах. Если количество пикселей в группе с приблизительно равными центральному яркостями превышает  $P\%$  от общего числа точек в окне, новым значением принимается среднее по этой группе. В противном случае, ищется группа, для которой это условие выполняется, и принимается среднее по этой группе. Если же такой группы не оказывается, то новое значение яркости получается взвешенным суммированием по всему окну. Величина  $P$  определяется по уровню шума на обрабатываемом изображении: чем больше  $P$ , тем более интенсивный шум может быть удален. Кроме того, значение  $P$  определяет, какого рода детали изображения не будут разрушаться при сглаживании. Для окна

5x5  $P \leq 20\%$  обеспечивает сохранение линий толщиной в 1 пиксел,  $P \leq 36\%$  - прямых углов,  $P \leq 50\%$  - границ типа скачка яркости. Как отмечают авторы, применение нескольких (в частности, двух) итераций не увеличивает существенно разрушения деталей изображения, но оказывается более эффективным для удаления гауссовского и импульсного шумов.

Рассмотрим фильтр *SUSAN* подробнее. Вес каждого пиксела в окне состоит из двух частей: пространственной, подобной гауссовскому фильтру, и яркостной, которая определяется близостью яркости данного пиксела к яркости центрального:

$$cI(r, r_0) = \exp\left(-\left(\frac{I(r) - I(r_0)}{t}\right)^2\right) \quad (5.4)$$

где  $r$  и  $r_0$  - текущая и центральная точки;  $I(r)$  - значение яркости в точке  $r$ ;  $t$  - порог;  $cI(r, r_0)$  - яркостная составляющая веса в точке  $r$  для области с центром  $r_0$ .

Обычно используется круглая маска радиусом 3.4, содержащая 37 пикселов. Причем сам центральный пиксел в расчетах не участвует, что позволяет устранять импульсный шум.

Окончательное уравнение, используемое для фильтра *SUSAN*, имеет вид:

$$J(x, y) = \frac{\sum_{i,j \neq 0} I(x+i, y+j) \exp\left[-\frac{r^2}{2\sigma} - \left(\frac{I(x+i, y+j) - I(x, y)}{t}\right)^2\right]}{\sum_{i,j \neq 0} \exp\left[-\frac{r^2}{2\sigma} - \left(\frac{I(x+i, y+j) - I(x, y)}{t}\right)^2\right]} \quad (5.5)$$

где,  $r = \sqrt{i^2 + j^2}$ ,  $\sigma$  - масштаб пространственного сглаживания (не связанный с дисперсией яркости изображения),  $t$  - порог яркости (масштаб "яркостного сглаживания") и, когда знаменатель равен нулю, применяется вышеупомянутое правило медианы восьми ближайших соседей.

## САМОСТОЯТЕЛЬНАЯ РАБОТА №6. ОБНАРУЖЕНИЕ ГРАНИЦ ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ

**Цель:** изучить основные методы выделения перепадов яркости (контуров).

### Задание

1. Преобразовать изображение к полутоновому виду.
2. Реализовать следующие алгоритмы обнаружения границ:
  - модуль градиента;
  - градиент Робертса;

$$- \quad |\nabla I| = |I(i, j) - I(i + 1, j)| + |I(i, j) - I(i, j + 1)|;$$

$$- \quad |\nabla I| = \max(|I(i, j) - I(i + 1, j)|, |I(i, j) - I(i, j + 1)|);$$

$$- \quad |\nabla I| = |I(i, j) - I(i + 1, j + 1)| + |I(i + 1, j) - I(i, j + 1)|.$$

$$- \quad |\nabla I| = \sum_{k,l \in W} |I(i, j) - I(i + k, j + l)|, \quad |\nabla I| = \max_{k,l \in W} |I(i, j) - I(i + k, j + l)|;$$

здесь  $W$  – скользящее окно  $3 \times 3$ , исследуемая точка находится в центре окна;

- аппроксимацию в виде:

$$\begin{array}{cc} 1 & 0 \\ -1 & \\ c & 0 - \\ c & \\ 1 & 0 \\ -1 & \end{array}$$

и

$$\begin{array}{cc} -1 & -c \\ -1 & 0 \\ 0 & 0 \\ 1 & c \\ 1, & \end{array}$$

где  $c$  – задается параметрически от 1 до 10;

- с масками вычисления лапласиана:

$$1) \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix},$$

$$2) \begin{pmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{pmatrix},$$

$$3) \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix},$$

$$4) \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}.$$

Реализовать процедуру повышения резкости изображения с одной из масок по варианту (см. выше). Т.е. предварительно нужно сложить заданную матрицу  $c$ :

$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 \end{pmatrix}$$

- согласование с плоскостью в окне  $2 \times 2$ ;
- оператор Преувеличения;
- маска согласования с поверхностью второго порядка для оператора Лапласа;
- лапласиан гауссиана в окне  $5 \times 5$ .

3. Построить гистограмму градиента. Оценить параметр распределения ( $a$ ) исходя из предположения распределения Релея

$$(P(x) = \frac{x}{a^2} e^{-\frac{x^2}{2a^2}}, x \geq 0).$$

Сравнить гистограмму градиента с распределением Релея.

*Примечание:* Гистограмму градиента и распределение Релея отобразить на одном графике.

4. Реализовать процедуру проверки точки, является ли она локальным максимумом вдоль одной из осей, если является, то присвоить значение 255, иначе 0.

### Оформление отчета

Перечень обязательных элементов отчета:

- 1) титульный лист;
- 2) лист задания: тема, цель, задание, исходное изображение;
- 3) формулы и маски свертки для всех алгоритмов обнаружения границ;
- 4) теория по распределению Релея;
- 5) экранные формы результатов работы алгоритмов обнаружения границ, гистограмма градиента и распределения Релея, процедуры проверки точки на локальный максимум;
- 6) выводы;
- 7) листинг программы с комментариями.

При защите работы знать следующие теоретические положения:

1. Что такое градиент, модуль градиента.
2. Градиент Робертса.
3. Аппроксимация градиента, чувствительность градиентных методов к выбору направления. Аппроксимация первой производной.
4. Оператор Лапласа (вид, маски для аппроксимации) – 2-е производные.
5. Принцип согласования с плоскостью в окне 2x2, 5x5.
6. Оператор Превитта.
7. Лапласиан гауссиана.

## Теоретические сведения

### ***Обнаружение границ на изображениях***

Исследованиями психологов установлено, что с точки зрения распознавания и анализа объектов на изображении наиболее информативными являются не значения яркостей объектов, а характеристики их границ – контуров.

Будем называть контуром изображения совокупность его пикселей, в окрестности которых наблюдается скачкообразное изменение функции яркости. Так как при цифровой обработке изображение представлено как функция целочисленных аргументов, то контуры представляются линиями шириной, как минимум, в один пиксел. При этом может возникнуть неоднозначность в определении линии контура.

При разработке алгоритмов выделения контуров нужно учитывать указанные особенности поведения контурных линий. Специальная дополнительная обработка выделенных контуров позволяет устранять разрывы и подавлять ложные контурные линии.

### ***Градиентные методы***

Различным объектам на изображениях соответствуют области с более или менее одинаковыми значениями яркости. На границах же яркость существенно меняется. Как вы знаете, мерой изменения некоторой величины является ее производная. На изображении величина яркости изменяется в пространстве. Пространственная производная - это градиент, который кроме величины имеет еще и направление, т.е. представляет собой вектор.

Величина или модуль вектора градиента определяет "силу" границы, т.е. насколько в данной точке границы отличаются яркости объекта и его окружения. Направление вектора градиента показывает направление наибольшего изменения яркости, т.е. этот вектор направлен перпендикулярно границе. Зачастую направление границы не имеет значения, и в таких случаях достаточно определить только величину *модуля градиента*:

$$|\nabla I| = \sqrt{(I(i, j) - I(i + 1, j))^2 + (I(i, j) - I(i, j + 1))^2} .$$



Максимальные значения градиента соответствуют участкам наибольшего изменения яркости. На изображениях, содержащих строго однородные области, ненулевые значения градиента будут иметь место только на границах областей.

Для вычисления градиента вместо производных по  $x$  и  $y$  можно использовать производные по любой паре перпендикулярных направлений. Если в качестве этих направлений взять диагональные и применить соответствующие дискретные разности, то получим *градиент Робертса*:

$$|\nabla I| = \sqrt{(I(i, j) - I(i + 1, j + 1))^2 + (I(i + 1, j) - I(i, j + 1))^2}.$$

При реализации процедуры подчеркивания контуров стремятся избежать трудоемких операций (умножения и извлечения корня), поскольку их приходится выполнять для каждой точки изображения. Поэтому часто используют более простые выражения, которые называют *аппроксимациями градиента*. Обычно модуль градиента приближают следующими выражениями:

$$\begin{aligned} |\nabla I| &= |I(i, j) - I(i + 1, j)| + |I(i, j) - I(i, j + 1)|, \\ |\nabla I| &= \max(|I(i, j) - I(i + 1, j)|, |I(i, j) - I(i, j + 1)|), \\ |\nabla I| &= |I(i, j) - I(i + 1, j + 1)| + |I(i + 1, j) - I(i, j + 1)|. \end{aligned}$$

Однако эти выражения уже не являются одинаково чувствительными к различным направлениям границ: 1 и 2 более чувствительны к вертикальным и горизонтальным границам, 3 – к диагонально ориентированным.

Иногда используют другие приближения градиента:

$$\begin{aligned} |\nabla I| &= \sum_{k, l \in W} |I(i, j) - I(i + k, j + l)|, \\ |\nabla I| &= \max_{k, l \in W} |I(i, j) - I(i + k, j + l)|, \end{aligned}$$

где  $W$  – область окна.

Для аппроксимации первой производной функции яркости изображения в окне  $3 \times 3$  обычно применяются маски свертки вида:

$$\begin{array}{ccc} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{array} \quad (6.1a)$$

и

$$\begin{array}{ccc} -1 & -c & -1 \\ 0 & 0 & 0 \\ 1 & c & 1, \end{array} \quad (6.1b)$$

первая из которых учитывает изменения яркости в горизонтальном направлении, а вторая в вертикальном. Значение  $c$  обычно выбирается равным 1 или 2.

Более сложные алгоритмы такого рода учитывают изменения яркости не только в двух перпендикулярных направлениях, но и в двух диагональных, порождая четыре маски.

### ***Использование вторых производных***

Для подчеркивания перепадов яркости изображения можно использовать вторые производные. Двумерный дифференциальный оператор носит название *оператора Лапласа* или *лапласиана* и имеет следующий вид:

$$\nabla^2 I = \frac{d^2 I}{dx^2} + \frac{d^2 I}{dy^2} \quad (6.2)$$

Применение этого оператора к изображению  $I(i,j)$  сводится к свертке изображения с маской вида:

$$\begin{array}{c} 0 & -1 \\ 0 & -1 \\ 4 & -1 \\ 0 & -1 \\ 0. \end{array}$$

Так как лапласиан может принимать как положительные, так и отрицательные значения, в операторе подчеркивания

контура необходимо взять его абсолютное значение. Если рассматривать диагональные направления, то аналогичным образом можно получить следующую маску расчета лапласиана:

$$\begin{array}{c} -1 \ 0 \ - \\ 1 \ 0 \\ 4 \ 0 \\ -1 \ 0 \ - \\ 1. \end{array}$$

Множество различных дискретных лапласианов можно получить суммированием этих аппроксимаций с различными весами. Например, взяв весовые коэффициенты 1 и 1, получаем еще один вариант оператора подчеркивания контуров:

$$\begin{array}{c} -1 \ -1 \\ -1 \\ -1 \ 8 \ - \\ 1 \\ -1 \ -1 \\ -1. \end{array}$$

Пример другой маски, используемой для аппроксимации оператора Лапласа:

$$\begin{array}{c} 1 \ -2 \\ 1 \\ -2 \ 4 \\ -2 \ 1 \\ -2 \ 1. \end{array}$$

### ***Объединение дифференциальных и сглаживающих операторов***

Для повышения помехоустойчивости рассмотренных методов, перед выполнением операции дифференцирования производят сглаживание функции яркости в пределах окна.

Ряд методов основан на подгонке или согласовании с функцией яркости выбранной поверхности первого или второго

порядка. Такой подход приводит к методу согласования. Рассмотрим принцип согласования на примере окна  $2 \times 2$ .

По наблюдаемым значениям функции яркости в пределах этого окна

$$\{I(m, n), I(m + 1, n), I(m, n + 1), I(m + 1, n + 1)\}$$

построим *аппроксимирующую плоскость*:

$$f(x, y) = ax + by + c.$$

Определив коэффициенты  $a$ ,  $b$ ,  $c$ , найдем частные производные и модуль градиента:

$$\frac{\partial F}{\partial x} = a, \quad \frac{\partial F}{\partial y} = b, \quad |\nabla f(x, y)| = \sqrt{a^2 + b^2}.$$

Коэффициенты  $a$  и  $b$  формируются при линейной обработке изображения в окне  $2 \times 2$  соответственно масками:

$$\frac{1}{2} \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \text{ и } \frac{1}{2} \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}.$$

При аппроксимации плоскостью с использованием окна  $3 \times 3$  можно получить по аналогии с окном  $2 \times 2$  следующие маски для формирования коэффициентов  $a$  и  $b$

$$\frac{1}{6} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad \frac{1}{6} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

Вычисление модуля градиента  $\sqrt{a^2 + b^2}$  с использованием этих масок определяет так называемый *оператор Превитта*, который часто используется на практике. Здесь также можно применить аппроксимации градиента  $|a| + |b|$  или  $\max\{|a|, |b|\}$ .

На окне  $3 \times 3$  можно построить аппроксимирующую поверхность второго порядка:

$$f(x, y) = ax^2 + by^2 + cxy + px + qy + z.$$

Тогда величина лапласиана по методу согласования будет равна:

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 2a + 2b,$$

где  $a$  и  $b$  формируются масками:

$$\frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ 1 & -2 & 1 \\ 1 & -2 & 1 \end{bmatrix}, \frac{1}{6} \begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix}.$$

Очевидно, дискретный лапласиан может быть вычислен с помощью единственной маски, являющейся удвоенной суммой указанных масок:

$$\frac{1}{3} \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix}.$$

Другой метод основан на объединении гауссовского сглаживания и оператора Лапласа. Операторы Гаусса и Лапласа линейны, а это означает, что они коммутативны, и их можно объединить в один, изменив порядок выполнения операций:

$$L * (G * I) = (L * G) * I = L \circ G * I.$$

Здесь  $*$  - означает свертку. Вместо того, чтобы сначала сгладить изображение, а потом взять лапласиан от результирующего изображения, сначала выполняется свертка обоих операторов, а затем полученный оператор применяется к

изображению. Свертка операторов выполняется один раз в самом начале, и изображение обрабатывается один раз. При последовательном сглаживании и дифференцировании изображения, оно обрабатывается дважды. Полученный таким образом оператор получил название *лапласиан гауссиана* (LoG - Laplacian of a Gaussian). Иногда, по форме соответствующей одномерной кривой, этот оператор еще называют «сомбреро» («*Mexican hat*»)

Двумерный оператор Лапласа определен в (6.2), а двумерный оператор Гаусса, напомним, имеет вид (5.1):

$$G = \exp(-(x^2+y^2)/(2t^2)).$$

Их свертка, в таком случае, запишется следующим образом:

$$\text{LoG} = L * G = \exp(-(x^2+y^2)/(2t^2)) * (x^2+y^2-2t^2)/t^4.$$

Эта формула порождает маску, с которой необходимо свертывать изображение. Например, такую (при  $2t^2=1.44$ ):

$$\begin{array}{cccc} 1 & 4 & 6 & 4 \\ 1 & 4 & 5 & -8 \\ 5 & 4 & 6 & -8 \\ 52 & -8 & 6 & 4 \\ 5 & -8 & 5 & 4 \\ 1 & 4 & 6 & 4 \\ 1 & & & \end{array}$$

## САМОСТОЯТЕЛЬНАЯ РАБОТА № 7. СПЕКТР ИЗОБРАЖЕНИЯ. ПРЕОБРАЗОВАНИЯ ФУРЬЕ И АДАМАРА

**Цель работы:** Изучить преобразования Фурье и Адамара, научиться получать спектр изображения, аналитически рассчитывать дискретное преобразование Фурье (ДПФ) от двумерной функции Уолша (прямоугольник с единичной яркостью).

### Задание

#### *1. На компьютере*

1. Реализовать алгоритм быстрого преобразования Фурье (БПФ).
2. С помощью БПФ получить спектр изображения размером 256\*256.
3. Вывести на экран изображение, соответствующее модулю спектра.
4. Выбрать круг с задаваемым радиусом и центром, в центре изображения п.1.3 обнулить точки, входящие в круг или вне его, оставив остальные точки без изменения. Применить к результату обратное БПФ. Сравнить полученное изображение с исходным. Сделать выводы.
5. Выполнить аналогичные действия, используя вместо круга ромб с задаваемыми диагоналями.
6. Умножить спектр изображения на  $e^{i\pi k}$ . Выполнить обратное преобразование результата.
7. Реализовать преобразование Адамара. Выполнить аналогичные п. 1.3-1.5 исследования. Сделать сравнительные выводы.

#### *2. Аналитически*

Рассчитать ДПФ от двумерной функции Уолша, с произвольными координатами на изображении размером  $N \times N$ . Яркость остальных точек изображения равна 0.

### **Оформление отчета**

Отчет должен содержать:

- 1) титульный лист;

- 2) тему, цель, задание, исходное изображение;
- 3) теорию по алгоритмам БПФ и преобразованию Адамара;
- 4) аналитические расчеты ДПФ от двумерной функции Уолша;
- 5) листинг программы с комментариями;
- 6) экранные формы;
- 7) выводы.

При защите работы знать ответы на следующие вопросы:

1. Что такое спектр изображения?
2. Какие свойства спектра изображения?
3. Как представить спектр изображения через одномерное преобразование Фурье?
4. В чем заключается быстрое преобразование Фурье?
5. Что такое циклическая свертка?
6. Как представляется операция свертки через преобразование Фурье?
7. Как располагаются высокие и низкие частоты в спектре изображения?

### Теоретические сведения

Много полезной информации об изображении можно узнать, анализируя его спектр (или частотную область). Кроме того, многие методы обработки изображений основаны на анализе и трансформации спектральных характеристик изображения.

Спектром изображения будем считать результат применения к нему двумерного преобразования Фурье. В непрерывном случае преобразование Фурье имеет вид:

$$F(\omega_x, \omega_y) = \left( \frac{1}{2\pi} \right)^2 \int_{-\infty-\infty}^{+\infty+\infty} \int f(x, y) e^{i(\omega_x x + \omega_y y)} dx dy .$$

Обратное преобразование определяется формулой:



$$f(x, y) = \left( \frac{1}{2\pi} \right)^2 \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(\omega_x, \omega_y) e^{-i(\omega_x x + \omega_y y)} dx dy.$$

Здесь  $i = \sqrt{-1}$ . Коэффициент перед интегралом определяется условием:

$$F^{-1}(F(I)) = I.$$

Здесь  $F$  и  $F^{-1}$  - операторы прямого и обратного преобразования Фурье.

Преобразование Фурье или получение спектра оптического сигнала наглядно можно продемонстрировать при помощи обычной оптической линзы. Всем с детства известен простой физический опыт, заключающийся в фокусировке линзой солнечных лучей, в результате чего в фокальной плоскости формируется небольшое пятно, энергия которого является суммарной энергией лучей попадающих на линзу. Из оптики известно, что плоская выпуклая линза осуществляет преобразование Фурье. Рассмотрим, что при этом происходит. Солнечные лучи, падающие на линзу, в хорошем приближении представляют собой плоскую волну:

$$I = A e^{ikx}.$$

Здесь  $k$  - вектор нормали к волновому фронту волны,  $x$  - вектор пространственных координат. Уравнение  $(k, x) = \text{const}$  определяет плоскость одинаковой фазы волны. Предположим, что вектор нормали перпендикулярен плоскости линзы, которая совпадает с плоскостью  $x$ . Тогда  $(k, x) = 0$ , а  $I = A$ . Учитывая это, получим результат в фокусной плоскости линзы:

$$F(\omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} A e^{i\omega x} dx = A \delta(\omega).$$

Здесь  $\delta(\omega)$  - дельта функция Дирака. На самом деле в фокусе линзы получается точка конечных размеров, что связано с конечным размером линзы. В этом случае получим:

$$F(\omega) = \frac{1}{2\pi} \int_{-r}^{+r} A e^{i\omega x} dx = \frac{A}{2\pi} \frac{1}{i\omega} [e^{i\omega r} - e^{-i\omega r}] = \frac{A}{\pi} \frac{\sin \omega r}{\omega}.$$

Эта функция называется sinc. В дискретном случае преобразование Фурье имеет вид:

$$F_{kl} = F(f) = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f_{nm} e^{i\frac{2\pi}{N}kn + i\frac{2\pi}{M}lm}.$$

Часто используется аббревиатура ДПФ.

$$\frac{1}{N} \sum_{n=0}^{N-1} e^{i\frac{2\pi}{N}kn} = \frac{1}{N} \frac{e^{i2\pi k} - 1}{e^{i\frac{2\pi}{N}k} - 1} = \begin{cases} 0, k = 1..N-1, \\ 1, k = 0 \end{cases} = \delta(k).$$

Таким образом, выполняется условие  $F^{-1}(F(I)) = I$ .

Ясно, что число степеней свободы в спектре не может отличаться от числа степеней свободы в самом сигнале. Из формулы же следует получение из изображения двух частотных плоскостей, соответствующих действительной и мнимой составляющим.

Остановимся на наиболее существенных свойствах двумерного дискретного спектра Фурье. Вычислим спектральные коэффициенты в частотных точках  $k + jN, l + j_2M$ :

$$F_{k+j_1N, l+j_2M} = F(f) = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f_{nm} e^{i\frac{2\pi}{N}kn + i\frac{2\pi}{M}lm} e^{i2\pi j_1n + i2\pi j_2m}.$$

Поскольку при любых целых значениях  $j_1$  и  $j_2$  последний множитель в полученном выражении равен единице, то отсюда имеем равенство:

$$F_{k+j_1N, l+j_2M} = F_{k,l},$$

означающее *прямоугольную периодичность двумерного ДПФ*. Однако необходимо иметь в виду, что спектральные коэффициенты  $F_{k,l}$  являются *комплексными числами*, в том числе и при вещественном сигнале. Но тогда возникает вопрос. Общее количество спектральных компонент, как установлено, равно  $NM$ . Комплексное число эквивалентно паре вещественных чисел - действительной и мнимой частям при алгебраическом или модулю и фазе при экспоненциальном представлении. Следовательно, полный спектр описывается  $2NM$  вещественными числами, что вдвое превышает размерность самого сигнала. В этом, на первый взгляд, содержится противоречие. Оно находит свое разъяснение при дальнейшем изучении свойств двумерного ДПФ.

Преобразуем соотношение для ДПФ следующим образом. Во-первых, вместо частот  $k,l$  подставим частоты  $N-k, M-l$ . Во-вторых, выполним комплексное сопряжение. В результате нетрудно получить выражение:

$$F_{N-k, M-l} = F_{k,l}^*,$$

которым устанавливается однозначная связь между спектральными коэффициентами в двух различных точках спектрального прямоугольника. Полученным соотношением и снимается противоречие, поскольку количество независимых спектральных коэффициентов уменьшается благодаря данной спектральной симметрии в два раза. Согласно установленному свойству, спектрально-сопряженной зависимостью связаны между собой спектральные коэффициенты, принадлежащие левому верхнему и правому нижнему углам прямоугольника (центрально симметричные точки). Аналогично также связаны между собой коэффициенты Фурье из правого верхнего и левого нижнего участков спектрального прямоугольника.

### *Циклическая свертка*

Достоинством обычного преобразования Фурье является то, что с его помощью очень просто выполняется стационарная фильтрация сигналов. Как известно, для этого требуется получить частотный спектр сигнала и частотный коэффициент передачи фильтра. Затем, перемножив их, определить частотный спектр выходного сигнала, а выполнив обратное преобразование Фурье - найти сам выходной сигнал.

Двумерный стационарный (инвариантный к сдвигу) фильтр характеризуют двумерной импульсной характеристикой  $a_{i_1, i_2}$ , позволяющей определить выходной сигнал при помощи двумерной свертки:

$$y_{i_1, i_2} = \sum_{r_1=0}^{I_1-1} \sum_{r_2=0}^{I_2-1} x_{r_1, r_2} \cdot a_{i_1-r_1, i_2-r_2}.$$

Это уравнение обычной свертки, в нем фигурируют обычные, непериодические сигналы и непериодическая ИХ фильтра. При цифровой обработке в ЭВМ не существует частотных спектров, соответствующих таким сигналам и ИХ. Для описания сигнала в частотной области привлекается, как установлено выше, периодически продолженный сигнал  $x_{i_1, i_2}$ , которому соответствует дискретный спектр  $X_{k_1, k_2}$ . По аналогии вводится и периодически продолженная ИХ:

$$\tilde{a}_{i_1, i_2} = a_{i_1 - m_1 I_1, i_2 - m_2 I_2},$$

двумерное ДПФ которой  $\tilde{A}_{k_1, k_2}$  имеет смысл частотного коэффициента передачи цифрового фильтра. Покажем, что перемножая  $\tilde{X}_{k_1, k_2}$  и  $\tilde{A}_{k_1, k_2}$ , мы находим спектр сигнала, определяемого циклической сверткой. Циклическая свертка отличается от обычной свертки тем, что вместо функций  $x_{i_1, i_2}$  и  $\tilde{a}_{i_1, i_2}$  в ней представлены периодически продолженные

функции  $x_{i1}, i_2$  и  $a_{i1}, i_2$  в обычном и циклическом вариантах соответственно. Рис. 7.1 поясняет процесс

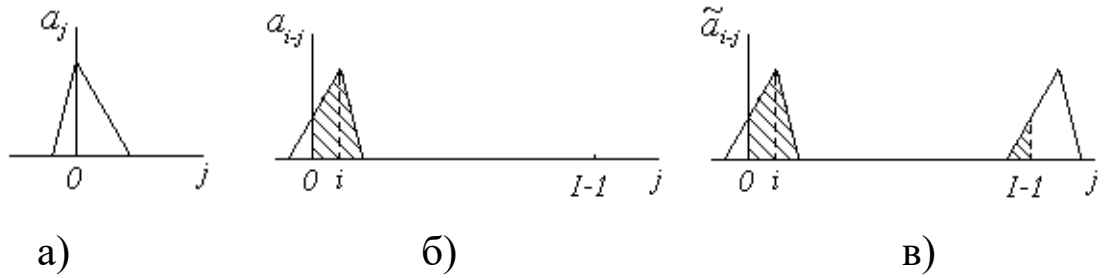


Рис. 7.1. Сравнение обычной и циклической свертки

Пусть заданы дискретные прямое и обратное преобразования Фурье:

$$F_{\omega} = F(f) = \sum_{n=0}^{N-1} f_n e^{i \frac{2\pi}{N} \omega n},$$

$$F_{\omega}^{-1} = F^{-1}(f) = \sum_{n=0}^{N-1} f_n e^{-i \frac{2\pi}{N} \omega n},$$

где  $F$  – оператор прямого преобразования Фурье,  $F^{-1}$  – обратного.

Изменяя порядок суммирования, получим:

$$S_k = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_n g_m \sum_{\omega=0}^{N-1} e^{i \frac{2\pi}{N} \omega n} e^{i \frac{2\pi}{N} \omega m} e^{-i \frac{2\pi}{N} \omega k} = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} f_n g_m \sum_{\omega=0}^{N-1} e^{i \frac{2\pi}{N} \omega (m+n-k)}$$

Используя свойства геометрической прогрессии, имеем:

$$\sum_{\omega=0}^{N-1} e^{i \frac{2\pi}{N} \omega (m+n-k)} = \delta(m+n-k), \text{ где } \delta(m+n-k) = \begin{cases} 1, & m+n-k=0; \\ 0, & m+n-k \neq 0. \end{cases}$$

Подставляя последнее равенство в выражение для  $S_k$ , получим:

$$S_k = \sum_{n=0; N-1} f_n g_{k-n}.$$

Таким образом, для свертки двух функций справедливо выражение:

$$f \otimes g = F^{-1}(F(f)F(g)).$$

Фактически ДПФ оперирует с периодически продолженными сигналами. Это следует из следующего соотношения:

$$f_k = F^{-1}F(f) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} f_n e^{i\frac{2\pi}{N}mn} e^{-i\frac{2\pi}{N}mk} = f_{k+lN}, l \in Z.$$

В одномерном случае это представляет собой повторяющиеся отрезки, в двумерном – прямоугольники изображений.

Как видно из формулы для двумерного ДПФ, оно может быть представлено как последовательные одномерные ДПФ по строкам и столбцам.

$$F_{kl} = F(f) = \frac{1}{NM} \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f_{nm} e^{i\frac{2\pi}{N}kn + i\frac{2\pi}{M}lm} = \frac{1}{NM} \sum_{m=0}^{M-1} e^{i\frac{2\pi}{N}lm} \sum_{n=0}^{N-1} f_{nm} e^{i\frac{2\pi}{N}kn}.$$

Рассмотрим подробнее одномерное ДПФ. Обозначим:

$$W = e^{i\frac{2\pi}{N}}.$$

Тогда одномерное ДПФ будет иметь вид:

$$F_m = F(f) = \sum_{n=0}^{N-1} f_n W^{mn}.$$

Иными словами ДПФ – это произведение исходного сигнала-вектора на матрицу, каждый элемент (n,m) которой, есть константа W в степени nm. Таким образом, процедура вычисления ДПФ, как и произведения матриц требует N\*N комплексных

умножений плюс сравнительно небольшое число операций возведения в степень которое в принципе можно задать таблично. Фактически ДПФ может быть получено с меньшим числом операций  $N \log_2 N$  с помощью алгоритма быстрого преобразования Фурье или БПФ. При  $N = 2^8$  выигрыш составит  $\frac{2^8}{2^3} = 32$  раза.

Аналогичным способом можно задать и преобразования других видов. Матрица  $W$  является унитарной. Т.е. обратная к ней матрица является ее комплексно сопряженной транспонированной матрицей. Это условие фактически означает, что применение к сигналу прямого и затем обратного ДПФ даст нам сам исходный сигнал. Таким образом, преобразования других видов должны сохранять свойство унитарности матрицы преобразования. В частности этому условию удовлетворяет преобразование Адамара. Его матрицу можно рекурсивно задать следующим образом:

$$Z_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, Z_{2N} = \begin{bmatrix} Z_N & Z_N \\ Z_N & -Z_N \end{bmatrix}.$$

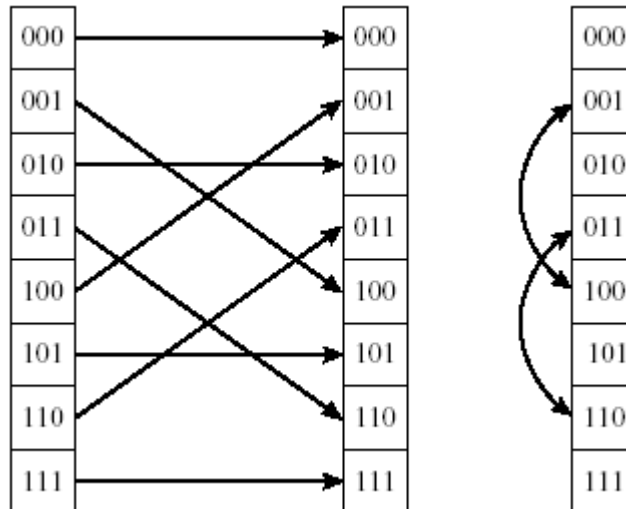
Таким образом, преобразование Адамара соответствует разложению по периодическим функциям прямоугольной формы, в отличие от ПФ, предусматривающего разложение по косинусоидальным и синусоидальным компонентам. Главным преимуществом преобразования Адамара является замена умножений комплексных величин изменениями знаков. Во всех остальных случаях это преобразование позволяет получать точно такие же результаты, что и преобразование Фурье.

Рассмотрим одну из модификаций алгоритма, предложенную Danielson and Lanczos. Суть предложенного алгоритма сводится к тому, что ДПФ размерности  $N$  может быть представлено в виде суммы ДПФ размерности  $N/2$ . Одно из них формируется из четных отсчетов исходного сигнала, второе – нечетных.

$$\begin{aligned}
F_k &= \sum_{j=0}^{N-1} e^{2\pi ijk/N} f_j \\
&= \sum_{j=0}^{N/2-1} e^{2\pi ik(2j)/N} f_{2j} + \sum_{j=0}^{N/2-1} e^{2\pi ik(2j+1)/N} f_{2j+1} \\
&= \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j} + W^k \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j+1} \\
&= F_k^e + W^k F_k^o
\end{aligned}$$

Здесь  $k$  изменяется от 0 до  $N$ ,  $W$  – определена выше, а не до  $N/2$ . Результаты преобразования четных и нечетных отсчетов периодичны по  $k$  с периодом  $N/2$ . Т.о. каждый повторяется два раза, чтобы получить  $k$ . Важный этап предложенного метода заключается в том, что он может быть применен рекурсивно. Таким образом, такой алгоритм можно применить к векторам размером  $N = 2^k$ . Если размер данных не является степенью 2-ки, то выбирается ближайшее большее  $N = 2^k$ , остальные значения заполняются 0. Таким образом, применяя итеративно алгоритм мы можем уменьшать размерность преобразования до 1. Преобразование  $\Phi$  размерности 1 сводится к оператору копирования входного значения в выходное. Чтобы получить правильный порядок коэффициентов Фурье на выходе, не смотря на последовательное разделение на четные и нечетные компоненты на всех этапах, необходимо провести переиндексирование компонент вектора в соответствии с преобразованием двоичной инверсии. Этот процесс можно пояснить следующей схемой:





Таким образом, алгоритм БПФ выполняется в два этапа. Сначала производится двоичная инверсия входных данных. Затем последовательно выполняются преобразования для размеров данных 2, 4, 8, . . . , $N$ . Описанный алгоритм еще называют прореживанием по времени. Или Cooley-Tukey FFT алгоритм. Другой вариант алгоритма начинается сразу со второго этапа. А затем выполняется двоичная инверсия результата. Этот алгоритм называют прореживание по частоте или Sande-Tukey FFT алгоритм. Для ряда приложений использующих БПФ, таких как, например, вычисление свертки, инверсию можно не делать. Так как выполняется прямое и обратное преобразование, а в фурье плоскости поэлементное умножение спектров сигналов.

## Литература

1. Потапов А. А., Пахомов А. А., Никитин С. А., Гуляев Ю. В. Новейшие методы обработки изображений. — М.: Физматлит, 2008. — 496 с.
2. Фисенко В.Т., Фисенко Т.Ю. Компьютерная обработка и распознавание изображений: учеб. пособие.- СПб: СПбГУ ИТМО, 2008.- 192 с.
3. Яне Б. Цифровая обработка изображений. Пер. с англ. - М.: Техносфера, 2007. - 584с.
4. Гонсалес Р., Вудс Р. Цифровая обработка изображений М.:Техносфера, 2006 – 1072 с.
5. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB М.:Техносфера, 2006 – 616 с.
6. Грузман И.С., Киричук В.С. и др. Цифровая обработка изображений в информационных системах: учеб. пособие. - Новосибирск: Изд-во НГТУ, 2002.-352 с.
7. Абламейко С.В., Лагуновский Д.М. Обработка изображений. Технология, методы, применение: учеб. пособие. — Мн.: Амалфея, 2000. — 304 с.
8. Романов В.Ю. Популярные форматы файлов для хранения графических изображений на IBM PC. - М.:Унитех, 1992. -156с.
9. Б.К.П. Хорн, Зрение роботов: перевод с англ. — М.: Мир, 1989.- 487 с.
10. Павлидис Т. Алгоритмы машинной графики и обработки изображений. / М.: Радио и связь, 1986. - 399 с.
11. Прэтт У. Цифровая обработка изображений. - М.:Мир, 1982.- 480 с.
12. Smith S.M., Brady J.M. SUSAN - a New Approach to Low Level Image Processing. // DRA Technical Report TR95SMS1b, 1995. - 57p.
13. Caselles V. Geometric Models for Active Contours. // Proc. of IEEE Int. Conf. on Image Processing. – Washington D.C., 1995, v3. -p.9-12.
14. Canny J.F. A computational approach to edge detection. // IEEE Trans. on Pattern Analysis and Machine Intelligence. - 1986, v.8, N6. -p.679-698.