

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра электронных вычислительных машин

М. М. Лукашевич, Р. Х. Садыхов

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

Лабораторный практикум для студентов специальности I-40 02 01
«Вычислительные машины, системы и сети»
всех форм обучения

Версия 4.0

Минск БГУИР 2013

УДК 621.391(076.5)
ББК 32.811.3я73
Л 84

Рецензент:
ведущий научный сотрудник ОИПИ НАН Беларуси,
канд. техн. наук А.А. Дудкин

Лукашевич, М. М.

Л 84 Цифровая обработка сигналов и изображений: лабораторный
практикум для студ. спец. I-40 02 01 «Вычислительные машины,
системы и сети» всех форм обуч. / М. М. Лукашевич, Р. Х. Садыхов –
Минск : БГУИР, 2010. – 35 с.
ISBN 978-985-488-598-8

В лабораторном практикуме содержатся методические указания к выполнению лабораторных работ, предназначенные для изучения некоторых разделов дисциплины «Цифровая обработка сигналов и изображений». Изложены методы пространственной обработки изображений, алгоритмы фильтрации изображений, основные аспекты кластерного анализа при распознавании образов. Практикум дополняет уже изданные авторами методические указания к лабораторным работам, посвященные анализу сигналов, дискретным преобразованиям и теории нейронных сетей.

УДК 621.391(076.5)
ББК 32.811.3я73

ISBN 978-985-488-598-8

© Лукашевич М. М., Садыхов Р. Х., 2010

© УО «Белорусский государственный
университет информатики
и радиоэлектроники», 2010

Содержание

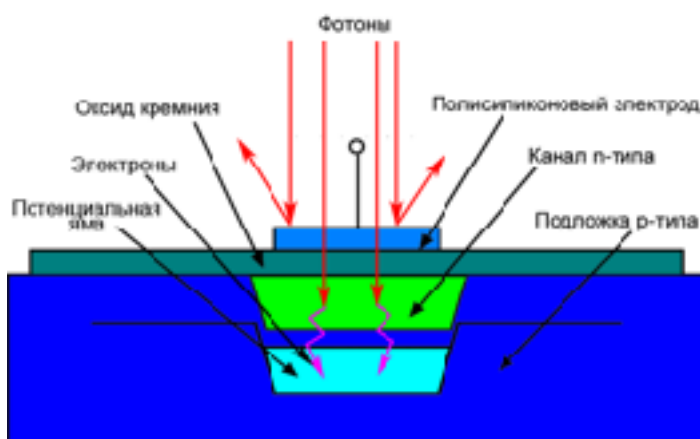
Общие теоретические сведения.....	4
Лабораторная работа №1 Гистограмма и гистограммная обработка. Поэлементная обработка изображений. Фильтрация изображений.....	11
...	11
Теоретические сведения.....	23
Задание для лабораторной работы.....	
Лабораторная работа №2 Кластерный анализ в распознавании образов.....	24
.	32
Теоретические сведения.....	34
Задание для лабораторной работы.....	
Литература.....	

Общие теоретические сведения

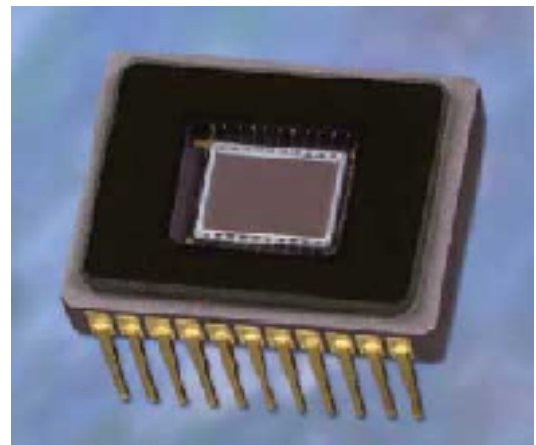
1. Основы цифрового представления изображений

Изображение – это двумерное отображение наблюдаемой сцены, возникающее в результате регистрации лучистой энергии, исходящей из наблюдаемой сцены, с помощью некоторого устройства – сенсора (совокупности сенсоров). Регистрируемый сенсором сигнал возникает в результате взаимодействия источника «освещения» с элементами изображаемой «сцены». Цвет – это психическое свойство человеческого зрения, возникающее при наблюдении объектов и света, а не физические свойства объектов и света; результат взаимодействия света сцены и зрительной системы человека.

Большинство устройств ввода данных в современных системах обработки изображений, а также подавляющее большинство цифровых фото- и видеокамер используют сенсоры в форме двумерного массива (матрицы). Типичным чувствительным элементом является матрица на основе приборов с зарядовой связью (ПЗС) или Charge Coupled Device (CCD). Ответная реакция каждого элемента пропорциональна интегралу световой энергии, попадающей на поверхность этого элемента за время экспозиции (рис. 1).



а



б

Рис. 1: а - элемент CCD-матрицы; б – CCD-матрица

В общем виде конструкция CCD-элемента выглядит следующим образом: кремниевая подложка p-типа оснащается каналами из полупроводника n-типа. Над каналами создаются электроды из поликристаллического кремния с изолирующей прослойкой из оксида кремния. После подачи на такой электрод электрического потенциала в обеднённой зоне под каналом n-типа создаётся *потенциальная яма*, назначение которой – хранить электроны. Фотон, проникающий в кремний, приводит к генерации электрона, который притягивается потенциальной ямой и остаётся в ней. Большое количество фотонов (яркий свет) обеспечивает большой заряд ямы. Затем надо считать

значение этого заряда, именуемого также *фототоком*, и усилить его. Считывание фототоков CCD-элементов осуществляется так называемыми *последовательными регистрами сдвига*, которые преобразовывают строку зарядов на входе в серию импульсов на выходе. Данная серия представляет собой аналоговый сигнал, который в дальнейшем поступает на усилитель.

Существует две основные формы представления изображений: *непрерывная* и *дискретная*. Очень редко изображения, получаемые в информационных системах, имеют цифровую форму. Преобразование изображений к цифровому виду включает в себя два этапа. Первый этап – замена непрерывного кадра дискретным, обычно называется *дискретизацией*, а второй этап выполняет замену непрерывного множества значений яркости множеством квантованных значений (Q дискретных уровней) и носит название *квантования*. При цифровом представлении каждому из квантованных значений яркости ставится в соответствие двоичное число, чем и достигается возможность ввода изображения в ЭВМ.

Компьютеры обрабатывают не изображения, а только массивы дискретных чисел. Таким образом, изображение представляется в виде двумерного массива чисел. Точка в 2D-сетке называется *пикселем* (*pixel*) (рис. 2, а, б). Это название является сокращением понятия *picture element* (элемент изображения). Пиксель представляет энергетическую освещенность в соответствующем месте сетки. Положение пикселя задается с помощью общепринятого обозначения для матриц.

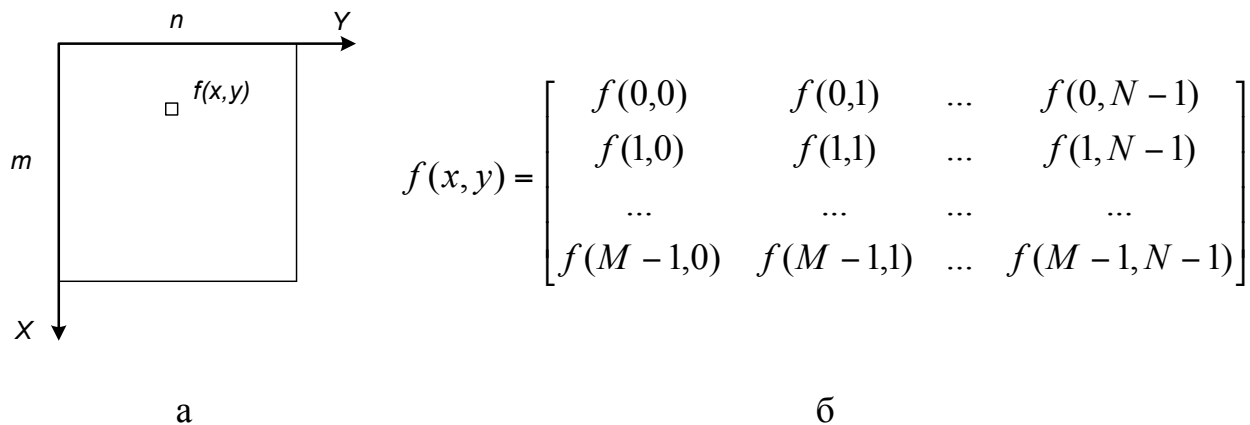


Рис. 2: а - положение пикселя в 2D-сетке; б – матричное представление изображения

Размер пикселя цифрового изображения определяется разрешением, а минимальная разница между яркостями пикселя определяется глубиной цвета. *Разрешение изображения* измеряется количеством пикселей, расположенных на единице длины. Как правило, в качестве единицы длины при оцифровке изображений используется дюйм, в таком случае разрешение измеряется в количестве пикселей на дюйм (dots per inch, *dpi*). *Глубина цвета* определяется количеством различных значений, которое может принимать пиксель.

Как правило, изображения квантуются в 256 уровней яркости. Тогда каждый пиксель занимает 8 бит или 1 байт. Этот битовый размер хорошо подходит к архитектуре стандартных компьютеров, которые могут обращаться к памяти побайтово. Кроме того, разрешающая способность является приемлемой для того, чтобы создать иллюзию непрерывного изменения в уровнях яркости, поскольку относительное разрешение зрительной системы человека по интенсивности не лучше, чем приблизительно 2%. В общем виде значение Q (число уровней (градаций) яркости) обычно выбирают равным целочисленной степени двойки:

$$Q = 2^k \quad (1)$$

Дискретные уровни яркости расположены с постоянным шагом и принимают целые значения в интервале $[0, Q-1]$. Общее количество битов b , необходимое для хранения цифрового изображения, определяется по формуле

$$b = M \times N \times k \quad (2)$$

Наилучшая глубина цвета зависит от специфики задачи и типа исходных изображений. Наиболее часто встречаются следующие случаи:

- 2 уровня яркости (черный и белый) – бинарное изображение;
- 256 уровней яркости (1 байт на пиксель) – полутоновое изображение;
- 65 536 уровней яркости (2 байта на пиксель) – полутоновое изображение;
- 16,7 млн цветов (3 байта – красный, синий, зеленый – на пиксель) – цветное изображение;
- 4,3 млрд цветов (4 байта – голубой, сиреневый, желтый, черный – на пиксель) – цветное изображение.

Последнее представление используется, как правило, лишь в издательских системах, тогда как первые четыре способа широко распространены в обработке изображений.

2. Основные взаимосвязи между пикселями изображения

Важным свойством дискретных изображений являются их окрестные соотношения, поскольку они определяют связную область. *Связная область* – множество пикселей, у каждого из которых есть хотя бы один сосед, принадлежащий данному множеству.

Пиксель p с координатами (x, y) имеет четыре горизонтальных и вертикальных соседних пикселя с координатами: $(x+1, y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$. Эта группа пикселей, называемая четыре соседа p , обозначается через $N_4(p)$ (рис. 3, а).

Четыре диагональных соседних пикселя по отношению к пикселю p имеют координаты $(x+1, y+1)$, $(x+1, y-1)$, $(x-1, y+1)$, $(x-1, y-1)$ и обозначаются через $N_d(p)$. Эти четыре точки вместе с четырьмя указанными выше называются

восемь соседей p и обозначаются через $N_8(p)$ (рис. 3, б). Некоторые из точек $N_4(p)$, $N_d(p)$, $N_8(p)$ могут выходить за пределы изображения, если (x,y) находится на границе изображения.

	1	
2	*	3
	4	

а

1	2	3
4	*	5
6	7	8

б

Рис. 3: а – 4-связность; б – 8-связность

В общем случае можно выделить три типа связей пикселей:

1) *четырёхсвязный*. Два пикселя p и q с определенными значениями интенсивности являются четырёхсвязными, если q относится к группе $N_4(p)$.

2) *восьмисвязный*. Два пикселя p и q со значениями интенсивности из Q , где Q -ряд значений интенсивности, являются восьмисвязными, если q относится к группе $N_8(p)$.

3) *m-связный (смешанная связь)*. Два пикселя p и q со значениями интенсивностей из Q являются m -связными, если:

а) q относится к группе $N_4(p)$;

б) q относится к группе $N_d(p)$ и множество $N_4(p) \cap N_4(q)$ – пустое.

Иными словами, это множество пикселей, являющихся четырьмя соседними как по отношению к p , так и по отношению к q со значениями интенсивности из Q .

3. Метрические свойства для изображений

Введем понятие *расстояние между объектами*, отражающее меру сходства, близости объектов между собой по всей совокупности используемых признаков. Расстояния между двумя векторами X и Y размерностью $n \times 1$ удовлетворяют трем аксиомам:

- 1) $d(X,Y)=0$, если $X=Y$;
- 2) $d(X,Y)=d(Y,X)$;
- 3) $d(X,Y) \leq d(X,Z)+d(Z,Y)$ для всех Z .

Наиболее важной метрикой является l^p -метрика:

$$l^p(x,y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p} \quad \text{для } p \geq 1. \quad (3)$$

Если $p=2$, получим классическую Евклидову метрику.

Для пикселей p и q с координатами (x,y) и (s,t) соответственно определим функцию расстояния или метрику D следующим образом:

1) евклидово расстояние:

$$D_E(p,q)=[(x-s)^2+(y-t)^2]^{1/2}; \quad (4)$$

2) модульное расстояние (метрика городских кварталов):

$$D_4(p,q)= |x-s| + |y-t| ; \quad (5)$$

3) шахматное расстояние:

$$D_8(p,q)= \max \{ |x-s| , |y-t| \}. \quad (6)$$

Геометрические места точек, удаленных на единичное расстояние от начала координат, построенные для различных метрик, будут иметь следующий вид (рис. 4):

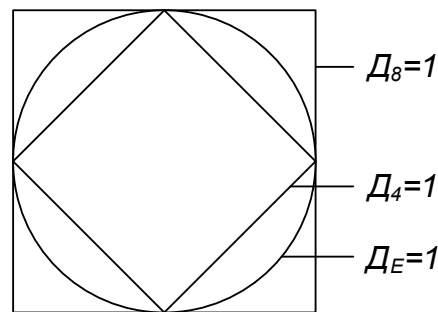


Рис. 4. Геометрическое место точек

4) Основные преобразования изображений

Основными преобразованиями при обработке изображений являются вращение, изменение масштаба и смещение (сдвиг) изображения. Все преобразования записываются в трехмерной декартовой системе координат, в которой координаты точки записываются как (X, Y, Z) .

Смещение (сдвиг) изображения. Предположим, что требуется сместить точку с координатами (X, Y, Z) в новое место, используя перемещения (X_0, Y_0, Z_0) . Смещение выполняется в соответствии с соотношениями: $X^*=X+X_0$, $Y^*=Y+Y_0$, $Z^*=Z+Z_0$, где X^*, Y^*, Z^* - координаты новой точки. В матричном виде это можно записать как $V^*=TV$, то есть:

$$\begin{bmatrix} X^* \\ Y^* \\ Z^* \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & X_0 \\ 0 & 1 & 0 & Y_0 \\ 0 & 0 & 1 & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (7)$$

Изменение масштаба изображения. Масштабирование на коэффициенты Sx, Sy, Sz по осям X, Y, Z производится с помощью матрицы преобразования:

$$S = \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

Вращение изображения. Вращение точки относительно координатной оси Z на угол θ реализуется с помощью преобразования:

$$R_\theta = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

Вращение точки вокруг оси X на угол α выполняется с помощью преобразования:

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

Вращение точки вокруг оси Y на угол β реализуется с помощью преобразования:

$$R_\beta = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

Контрольные вопросы

1. Основы цифрового представления изображений.
2. Основные взаимосвязи между пикселями изображения.
3. Метрические свойства для изображения.
4. Основные преобразования изображений.

Лабораторная работа № 1

ГИСТОГРАММА И ГИСТОГРАММНАЯ ОБРАБОТКА. ПОЭЛЕМЕНТНАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ. ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ

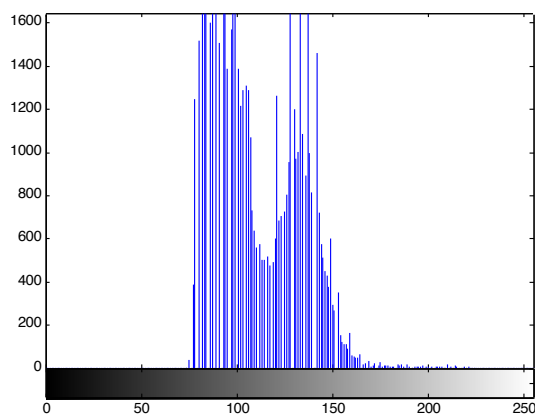
Цель работы: исследование методов «низкоуровневой» обработки изображений. Улучшение качества изображения и преобразование его в форму, наиболее удобную для визуального или машинного анализа.

Теоретические сведения

1. Гистограмма и гистограммная обработка

Каждый пиксель цифрового изображения характеризуется тремя независимыми величинами – $\{x, y, Q\}$, где пара целочисленных значений (x, y) описывает *геометрическое положение* данного пикселя в плоскости изображения, а величина Q характеризует его *яркость*, или *интенсивность*, в данной точке изображения (геометрическая и яркостная составляющие). Эти составляющие могут изменяться независимо друг от друга. Отсюда вытекает идея возможного *раздельного анализа яркости и геометрии изображения*.

Яркостная составляющая изображения характеризуется одномерным массивом гистограммы. *Гистограмма* характеризует частоту встречаемости на изображении пикселей одинаковой яркости. Так для 8-битного полутонового изображения гистограмма яркости представляет собой одномерный целочисленный массив из 256 элементов. Элементы гистограммы i содержат целые значения, равные количеству пикселей изображения, яркость которых равна значению i . Визуально гистограмма обычно отображается как обычный график одномерной функции (рис. 1.1).



а

б

Рис. 1.1: а – изображение; б – гистограмма изображения

2.

Поэлементная обработка изображений

Одной из основных задач цифровой обработки изображений является придание изображению таких качеств, благодаря которым его восприятие человеком будет по возможности комфортным. Часто бывает полезным подчеркнуть, усилить какие-то черты, особенности, нюансы наблюдаемой картины с целью улучшения ее субъективного восприятия. Подавляющее большинство процедур обработки для получения результата в каждой точке кадра привлекает входные данные из некоторого множества точек исходного изображения, окружающих обрабатываемую точку. Однако имеется группа процедур, где осуществляется так называемая *поэлементная обработка*. Здесь результат обработки в любой точке кадра зависит только от значения входного изображения в этой же точке.

Сущность поэлементной обработки изображений сводится к следующему. Пусть $f(x, y) = f_{x,y}$ и $g(x, y) = g_{x,y}$ – значения яркости исходного и полученного после обработки изображения в точке кадра, имеющей декартовы координаты x (номер строки) и y (номер столбца) соответственно. Поэлементная обработка означает, что существует функциональная, однозначная зависимость между этими яркостями:

$$g_{x,y} = h_{x,y}(f_{x,y}), \quad (1.1)$$

позволяющая по значению исходного сигнала определить значение выходного сигнала.

Линейное контрастирование изображения (линейная коррекция). 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 при визуализации соответствует уровню черного, а значение 255 – уровню белого. Предположим, что минимальная и максимальная яркости исходного изображения равны f_{\min} и f_{\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. При линейном контрастировании используется линейное поэлементное преобразование вида

$$g = a \cdot f + b, \quad (1.2)$$

параметры которого (a и b) определяются желаемыми значениями минимальной g_{\min} и максимальной g_{\max} выходной яркости. Решив систему уравнений

$$\begin{cases} g_{\min} = a \cdot f_{\min} + b \\ g_{\max} = a \cdot f_{\max} + b \end{cases} \quad (1.3)$$

относительно параметров преобразования a и b , нетрудно привести (1.2) к виду

$$g = \frac{f - f_{\min}}{f_{\max} - f_{\min}} (g_{\max} - g_{\min}) + g_{\min} \quad (1.4)$$

Нелинейное контрастирование изображения (нелинейная коррекция). Линейная коррекция изображения не всегда дает необходимый результат (например там, где наблюдается неравномерная освещенность изображения), поэтому в ряде случаев применяется нелинейная коррекция изображения. Часто применяются следующие функции.

Гамма-коррекция, изначальная цель которой – коррекция для правильного отображения на мониторе:

$$g = c \cdot f^{\gamma}, \quad (1.5)$$

где c и γ – константы.

Некоторые компьютерные системы имеют встроенную частичную гамма-коррекцию. Кроме этого, гамма-коррекция используется для универсального управления контрастом, в частности при обработке медицинских изображений. В зависимости от значения γ возможно целое семейство преобразований.

Логарифмическая коррекция, цель которой – сжатие динамического диапазона при визуализации данных:

$$g = c \cdot \log(1 + f), \quad (1.6)$$

где c – константа.

Использование логарифма позволяет узкий диапазон малых значений яркости преобразовать в более широкий диапазон выходных значений. Для больших значений входного сигнала верно противоположное утверждение. Такой тип преобразования используется для растяжения диапазона значений темных пикселей на изображении с одновременным сжатием диапазона значений ярких пикселей. При использовании обратного логарифмического преобразования происходит растяжение диапазона значений ярких пикселей на изображении с одновременным сжатием диапазона значений темных пикселей. Логарифмиче-

ская функция имеет важную особенность – позволяет сжимать динамический диапазон изображений, имеющих большие вариации в значениях пикселей.

Зависимость яркости исходного и результирующего изображений при линейном и нелинейном контрастировании представлена на рис. 1.2.

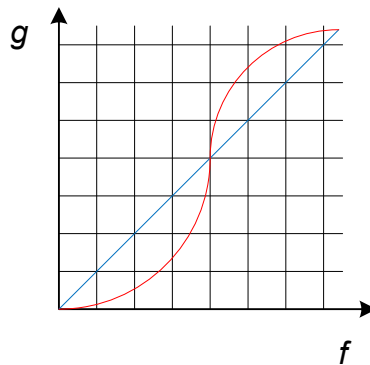
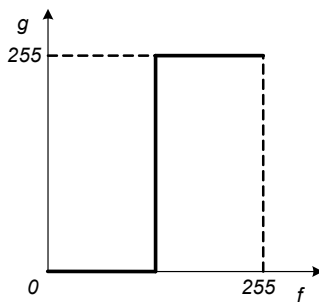
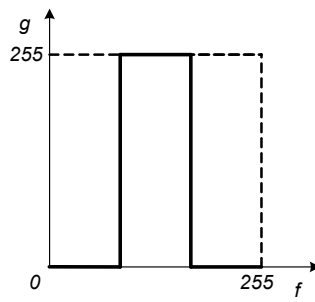


Рис. 1.2. Зависимость яркости исходного и результирующего изображений при линейном и нелинейном контрастировании

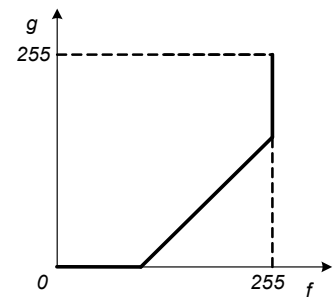
Препарирование изображения. Препарирование представляет собой целый класс поэлементных преобразований изображений (рис. 1.3, а-е).



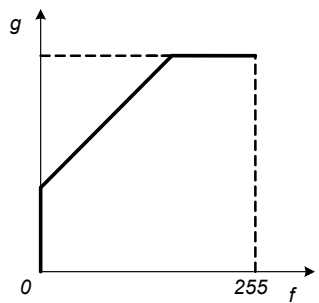
а



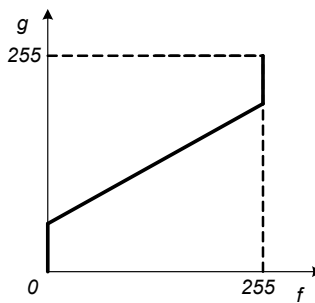
б



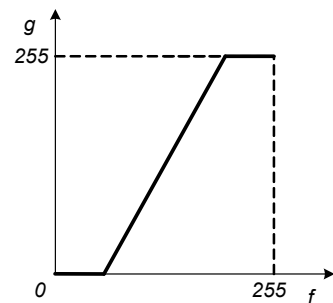
в



г



д



е

Рис. 1.3. Примеры преобразований, используемых при препарировании

Преобразование с пороговой характеристикой (см. рис. 1.3, а) превращает полутонное изображение, содержащее все уровни яркости, в бинарное, точки

которого имеют яркости $g = 0$ или $g = g_{\max}$. Такая операция, называемая бинаризацией, может быть полезной, когда для наблюдателя важны очертания объектов, присутствующих на изображении, а детали, содержащиеся внутри объектов или внутри фона, не представляют интереса.

На рис. 1.3 б преобразование выполняет яркостный срез изображения, выделяя те его участки, где яркость соответствует выделенному интервалу. При этом остальные участки оказываются полностью «погашенными» (имеют яркость, соответствующую уровню черного). Перемещая выделенный интервал по яркостной шкале и изменяя его ширину, можно детально исследовать содержание картины.

На рис. 1.3 в-е преобразования выполняют процедуры изменения яркости и контраста изображения. Преобразования, показанные на рис. 1.3 в, г, позволяют, соответственно, понизить или повысить яркость редактируемого изображения, используя следующие соотношения входного и выходного диапазона яркости изображения:

$$\begin{aligned} f_{\min} = 255 - g_{\max}, \quad g_{\min} = 0, \quad f_{\max} = 255 & \text{ (понижение яркости);} \\ g_{\min} = 255 - f_{\max}, \quad f_{\min} = 0, \quad g_{\max} = 255 & \text{ (повышение яркости).} \end{aligned} \quad (1.7)$$

Преобразования, показанные на рис. 1.3 в-е, позволяют повысить контраст наблюдаемого изображения в выбранном диапазоне яркости, когда выходное изображение использует полный динамический диапазон 0-255. По существу, это преобразование представляет собой линейное контрастирование, применяемое к избранному входному диапазону яркости исходного изображения. Точки, попавшие в интервал $[0, f_{\min}]$, после препарирования образуют черный фон, а точки, попавшие в интервал $[f_{\max}, 255]$, – белый фон.

Существует еще ряд характеристик, в соответствии с которыми также можно провести препарирование изображения.

Преобразование изображения в негатив. Такой переворот уровней яркостей цифровых изображений создает эффект фотографического негатива. Преобразование используется для усиления белых или серых деталей на фоне темных областей изображения, особенно когда темные области имеют преобладающие размеры, и широко применяется при обработке цифровых медицинских снимков. Цифровой негатив изображения получается путем вычитания значения пикселя из 255 (при условии, что 256 градаций цвета). Для цветного изображения

$$R' = 255 - R; \quad G' = 255 - G; \quad B' = 255 - B. \quad (1.8)$$

Преобразование к оттенкам серого, яркость точки изображения.

Данное преобразование заключается в получении яркости каждой точки и последующем копировании полученного значения во все три канала ($R = G = B = Y$):

$$Y = 0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B. \quad (1.9)$$

Соляризация изображения. Смысл соляризации заключается в том, что участки исходного изображения, имеющие уровень белого или близкий к нему уровень яркости, после обработки приобретают уровень черного. При этом уровень черного сохраняют и те участки, которые имели его на исходном изображении. А уровень белого на выходе приобретают участки, имеющие на входе средний уровень яркости (уровень серого). При данном виде обработки преобразование имеет вид

$$g = k \cdot f \cdot (f_{\max} - f), \quad (1.10)$$

где k – константа, позволяющая управлять динамическим диапазоном преобразованного изображения, а f_{\max} – максимальное значение исходного сигнала. Функция, описывающая данное преобразование, является квадратичной параболой, ее график при $k = 1/f_{\max}$ приведен на рис. 1.4. При $g_{\max} = f_{\max}$ динамические диапазоны изображений совпадают, что может быть достигнуто при $k = 4/f_{\max}$.

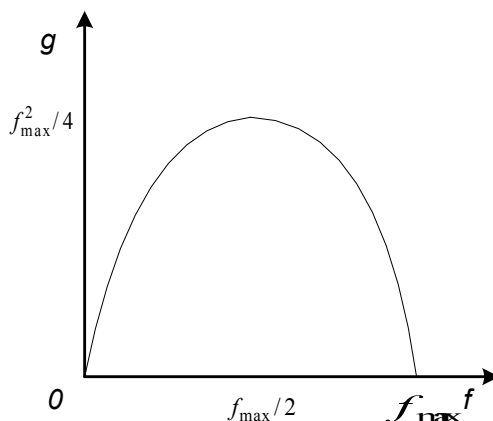


Рис. 1.4. Функция, описывающая соляризацию

3. Фильтрация изображений

Множество подходов к улучшению изображений распадается на две категории: методы обработки в пространственной области (пространственные ме-

тоды) и методы обработки в частотной области (частотные методы). К пространственной области относится совокупность пикселей, составляющих изображение. Функция предварительной обработки в пространственной области записывается в виде

$$g(x, y) = h[f(x, y)], \quad (1.11)$$

где $f(x, y)$ – входное изображение,

$g(x, y)$ – выходное (обработанное) изображение,

h – оператор функции f , определенный в некоторой области (x, y) .

Операции такого вида относятся к общему классу *операций над соседними элементами*. Эти операции являются основным инструментарием при *низкоуровневой обработке изображений* или *обработке изображений в пространственной области*.

Основным подходом при определении окрестности точки (x, y) является использование квадратной или прямоугольной области части изображения с центром в точке (x, y) . Центр этой части изображения перемещается от пикселя к пикселю начиная, например, с левого верхнего угла. При этом для получения $g(x, y)$ оператор применяется для каждого положения (x, y) . Хотя используются иногда и другие формы окрестности (например, круг), квадратные формы более предпочтительны из-за простоты их реализации.

Один из наиболее применяемых методов пространственной области основан на использовании фильтров (масок свертки, шаблонов, окон). Обычно маска фильтра представляет собой небольшую (например, размерность 3×3) двумерную систему, коэффициенты которой выбираются таким образом, чтобы обнаружить заданное свойство изображения (рис. 1.5, а).

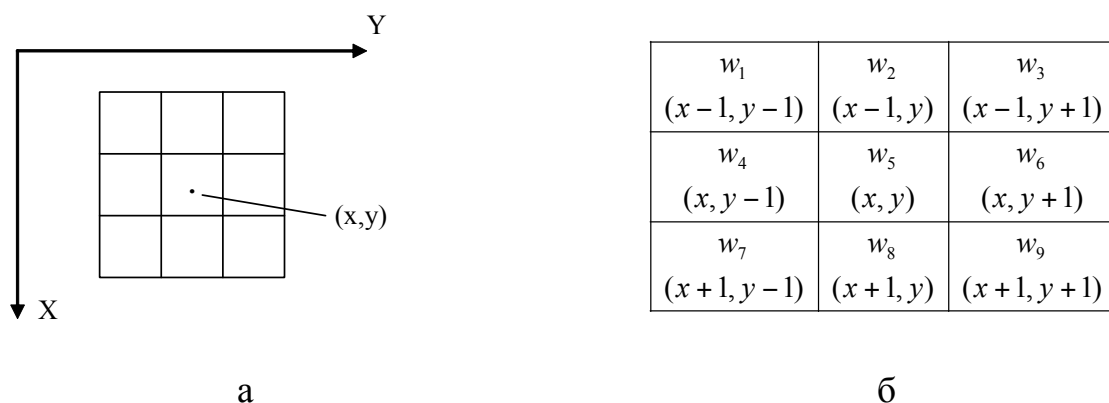


Рис. 1.5: а – маска фильтра; б – коэффициенты маски фильтра

Если величины w_1, w_2, \dots, w_9 представляют собой коэффициенты, маски пикселя (x, y) и его восьми соседей (рис. 1.5, б), то алгоритм можно представить как выполнение следующей операции на окрестности 3×3 точки (x, y) :

$$h[f(x, y)] = w_1 f(x-1, y-1) + w_2 f(x-1, y) + w_3 f(x-1, y+1) + w_4 f(x, y-1) + w_5 f(x, y) + w_6 f(x, y+1) + w_7 f(x+1, y-1) + w_8 f(x+1, y) + w_9 f(x+1, y+1). \quad 1.12$$

Под задачей фильтрации изображений в широком смысле понимают любые процедуры обработки изображений, при которых на вход процедуры подается растровое изображение и на выходе формируется растровое изображение. Однако чаще под «фильтрацией» понимают так называемую *помеховую фильтрацию*. Главная цель помеховой фильтрации заключается в такой обработке изображений, при которой результат оказывается более подходящим с точки зрения конкретного применения. В общем случае можно выделить *линейные фильтры* (сглаживающие фильтры, контрастоповышающие фильтры, разностные фильтры) и *нелинейные фильтры* (медианный фильтр).

Приведем краткое описание наиболее распространенных методов фильтрации.

Низкочастотный фильтр – ослабляет высокочастотные компоненты и усиливает роль низкочастотных. Частота в применении к изображениям отражает количество имеющихся в изображении деталей. Резкие перепады яркости, помехи и шумы являются примером высокочастотных элементов в изображении. Сглаживание изображения реализуется с помощью следующих ядер:

$$H_1 = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad H_2 = \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad H_3 = \frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \quad (1.13)$$

Высокочастотный фильтр – ослабляет низкочастотные компоненты в изображении и усиливает роль высокочастотных. Фильтры высокой частоты применяются для выделения таких деталей, как контуры, границы или для повышения резкости изображения. Каждый скачок яркости и каждый контур представляют собой интенсивные детали, связанные с повышенными частотами. Выделение высокочастотных компонент осуществляется с помощью следующих ядер:

$$H_1 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad H_3 = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix} \quad (1.14)$$

Оператор Робертса. Оператор Робертса является примером нелинейного фильтра. Преобразование каждого пикселя перекрёстным оператором Робертса может показать производную изображения вдоль ненулевой диагонали, и комбинация этих преобразованных изображений может также рассматриваться как градиент от двух верхних пикселей к двум нижним. Оператор Робертса используется ради быстроты вычислений, но проигрывает в сравнении с альтернативами из-за значительной проблемы чувствительности к шуму. Он даёт линии тоньше, чем другие методы выделения границ.

В обработке участвуют четыре пикселя, расположенные следующим образом (рис. 1.6).

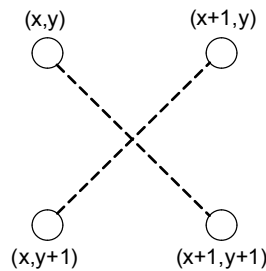


Рис. 1.6. Пиксели, участвующие в обработке оператором Робертса

Отклик оператора Робертса:

$$S(x, y) = \sqrt{[f(x+1, y) - f(x, y+1)]^2 + [f(x, y) - f(x+1, y+1)]^2} \quad (1.15)$$

Ядра свертки в данном случае будут выглядеть таким образом:

$$H_1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (1.16)$$

Свертка для каждого ядра вычисляется отдельно. В качестве отклика данного фильтра выступает величина

$$S = \sqrt{P^2 + Q^2}, \quad (1.17)$$

где P и Q – отклик ядер H_1 и H_2 .

Иногда в качестве оператора Робертса берется величина $S = |P| + |Q|$.

Оператор Собеля. Оператор Собеля применяют в алгоритмах выделения границ. Это дискретный дифференциальный оператор, вычисляющий приближенное значение градиента яркости изображения. Результатом применения оператора Собеля в каждой точке изображения является либо вектор градиента яркости в этой точке, либо его норма. Метод усиления края с помощью оператора Собеля рассматривает два различных ядра свертки:

$$H_1 = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}, \quad H_2 = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}. \quad (1.18)$$

Исходя из этих свертки вычисляется величина и направление краев. Свертка для каждого ядра вычисляется отдельно. В качестве отклика данного фильтра выступает величина

$$S = \sqrt{P^2 + Q^2}, \quad (1.19)$$

где P и Q – отклик ядер H_1 и H_2 .

Иногда в качестве оператора Собеля берется величина $S = |P| + |Q|$.

Оператор Превитта. Аналогично оператору Собеля действует оператор Превитта. Детектор границ Превитта является подходящим способом для оценки величины и ориентации границы. В то время как детектор с дифференциальным градиентом нуждается в трудоёмком вычислении оценки ориентации по величинам в вертикальном и горизонтальном направлениях, детектор границ Превитта даёт направление прямо из ядра с максимальным результатом. Метод усиления края с помощью оператора Превитта рассматривает два различных ядра свертки:

$$H_1 = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}, \quad H_2 = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}. \quad (1.20)$$

Результат работы оператора Превитта есть

$$\max\{P, Q\}, \quad (1.21)$$

где P и Q – отклик ядер H_1 и H_2 .

Оператор Лапласа. Дискретный оператор Лапласа часто используется в обработке изображений, например в задаче выделения границ или в приложениях оценки движения. Дискретный лапласиан определяется как сумма вторых производных и вычисляется как сумма перепадов на соседях центрального пикселя. Метод усиления края по Лапласу рассматривает целый ряд различных ядер свертки. Приведем некоторые из них:

$$H_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad H_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad H_3 = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}. \quad (1.22)$$

Как видно, сумма элементов матриц равна нулю, поэтому отклик фильтра может быть отрицательным. В этом случае значение отклика берется по модулю. В результате обработки области с постоянной или линейно возрастающей интенсивностью становятся черными, а области быстро изменяющихся значений интенсивности ярко высвечиваются.

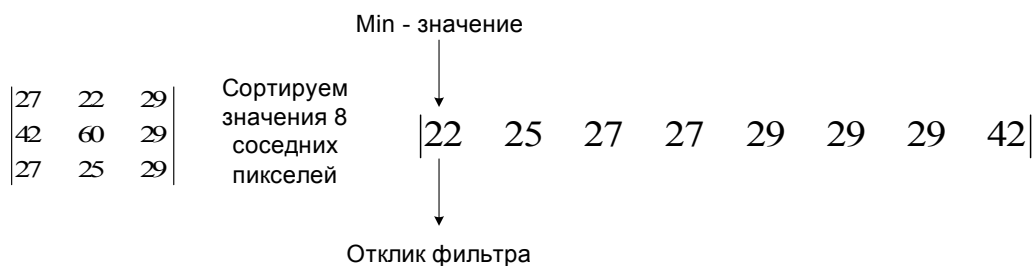
Ниже приведем некоторые пространственные процессы, которые не подпадают под категорию свертки и могут применяться для устранения различного вида шума.

Фильтр «гармоническое среднее». Гармоническое среднее ряда $f(n)$ вычисляется по формуле

$$G(f) = \frac{n}{\sum_{i=1}^n \frac{1}{f_i}}. \quad (1.23)$$

В процессе фильтрации значение текущего пикселя изображения заменяется на $f(n)$ множества значений девяти пикселей, включая текущий и соседние.

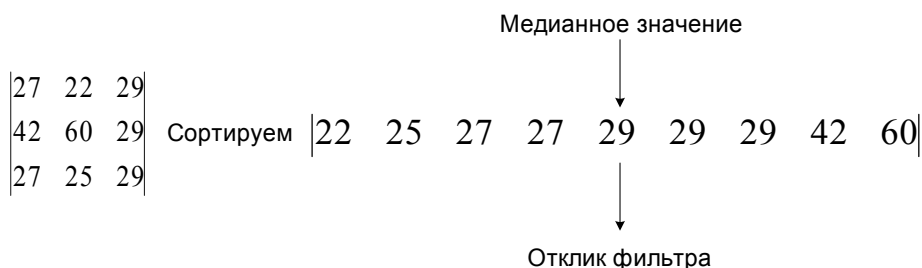
Min – фильтр. В процессе фильтрации значение текущего пикселя заменяется на минимальное значение соседних пикселей. Так, например, для ядра размерности 3 будем иметь:



Max – фильтр. В процессе фильтрации значение текущего пикселя заменяется на максимальное значение соседних пикселей (по аналогии с предыдущим фильтром).

Min-Max-фильтр. В процессе фильтрации значение текущего пикселя изображения сначала заменяется на минимальное значение соседних пикселей, а при повторном проходе на максимальное.

Медианный фильтр. Усредненное фильтрование использует значения элементов, содержащихся в области примыкания, для определения нового значения. Фильтр располагает элементы области примыкания в отсортированном порядке и отбирает среднее значение. Так, например, для ядра размерности 3 медианное значение будет пятым:



С помощью методов пространственной обработки изображений можно получить ряд интересных эффектов. Приведем некоторые из них.

Эффект тиснения. С помощью операции свертки можно реализовать преобразование, дающее эффект тиснения на изображении.

$$H = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}. \quad (1.24)$$

Бинарное «псевдополутонное» изображение. Исходное изображение обрабатывается при помощи маски D2 или D4: если значение пикселя меньше пропорционального значения соответствующего ему элемента маски, то он

обнуляется, иначе ему присваивается 255. Маска накладывается на изображение без перекрытия. Маски D2 и D4:

$$D_2 = \begin{pmatrix} 0 & 2 \\ 3 & 1 \end{pmatrix}, \quad D_4 = \begin{pmatrix} 0 & 4 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{pmatrix}. \quad (1.25)$$

При использовании пространственных процессов могут возникнуть следующие вопросы, связанные с особенностями обработки пикселей:

1. Устранение краевых эффектов;
2. Значение отклика выходит за пределы $[0, \dots, 255]$.

Для первого вопроса возможны следующие пути решения:

- Исключить из преобразования граничные пиксели изображения в этом случае выходное изображение будет иметь меньшие размеры, либо закрасить граничные пиксели, например черным цветом;
- Не включать соответствующий пиксель в суммирование, равномерно распределив его вес среди других пикселей окрестности;
- Дополнить (достроить) исходное изображение, добавив необходимое количество пикселей по границе. Количество достраиваемых строки столбцов, как правило, зависит от размера ядра. Здесь возможны два варианта:
 - Доопределить значения пикселей за границами изображения при помощи экстраполяции. Например, считать постоянным значение интенсивности вблизи границы или считать постоянным градиент интенсивности вблизи границы;
 - Доопределить значения пикселей за границами изображения при помощи зеркального отражения.

Для решения проблем, связанных с выходом значения за пределы $[0, \dots, 255]$, возможны следующие действия:

- Масштабировать полученные значения при положительных откликах фильтра;
- При отрицательном отклике фильтра брать либо абсолютное значение (по модулю), либо приводить к нулю.

Также в данном разделе стоит привести возможную «классификацию» шума на изображении:

1. Шум «соль и перец» – случайные белые и черные пиксели;
2. Импульсный шум – случайные белые пиксели;
3. Гауссов шум – колебания интенсивности, распределенные по нормальному закону.

Задание для лабораторной работы

ЧАСТЬ 1

Написать программу, выполняющую следующие операции над изображениями (обязательна обработка как цветных, так и полутоновых изображений):

1. Загрузить и отобразить изображение на экране. Реализовать возможность сохранения и вывода на экран обработанного изображения.
2. Вывести гистограмму яркости изображения.
3. Выполнить поэлементную обработку изображения в соответствии с вариантом, выданным преподавателем. Вывести гистограмму яркости обработанного изображения.
4. Выполнить фильтрацию изображения (либо иной пространственный процесс) в соответствии с вариантом, выданным преподавателем. Вывести гистограмму яркости обработанного изображения.

ЧАСТЬ 2

Выбрать одну из своих фотографий (изображение студента на данной фотографии приветствуется 😊), которая требует определенной обработки (например, изменения контраста, яркости и т.д.), выполнить данную обработку. Показать преподавателю исходную фотографию, результат, обосновать необходимость определенной обработки, пояснить алгоритм. Можно использовать один из алгоритмов, приведенный в методичке, либо иной, но в рамках тематики лабораторной работы.

Примечание: Варианты для выполнения лабораторной работы выдается преподавателем. Варианты поэлементной обработки и фильтрации (либо иного пространственного процесса) могут не совпадать.

Вариант	Поэлементная обработка	Примечание
1	Линейное контрастирование	При реализации интерфейса предусмотреть возможность ввода параметров g_{min} и g_{max}
2	Гамма-коррекция	При реализации интерфейса предусмотреть возможность ввода параметров c и γ
3	Логарифмическая коррекция	При реализации интерфейса предусмотреть возможность ввода параметра c

4	Препарирование изображения (рис. 1.3, а, б)	При реализации интерфейса предусмотреть возможность ввода параметров: рис. 1.3, а – порог бинаризации; рис. 1.3, а – интервал яркостного среза
5	Препарирование изображения (рис. 1.3, в, г)	При реализации интерфейса предусмотреть возможность ввода параметров: рис. 1.3, в, г – пороги для f и g
6	Препарирование изображения (рис. 1.3, д, е)	При реализации интерфейса предусмотреть возможность ввода параметров: рис. 1.3, д, е - g_{min} и g_{max} ; f_{min} и f_{max}
7	Преобразование изображения в негатив	-
8	Соляризация	-

Вариант	Фильтрация, пространственные процессы	Примечание
1	Низкочастотный фильтр	-
2	Высокочастотный фильтр	-
3	Оператор Робертса	-
4	Оператор Собеля	-
5	Оператор Превитта	-
6	Min-фильтр, max-фильтр, min-max-фильтр	-
7	Фильтр «гармоническое среднее», медианный фильтр	При обработке изображения медианным фильтром внести в исходное изображение импульсный шум. При реализации интерфейса предусмотреть возможность ввода размера маски медианного фильтра
8	Эффект тиснения, бинарное «псевдополутонное» изображение	

Рекомендуемая литература

Яне, Б. Цифровая обработка изображений / Б. Янс. – М. : Техносфера, 2007.

Контрольные вопросы

1. Гистограммный анализ. Построение гистограммы яркости изображения.
2. Поэлементная обработка изображения:
 - a. Линейное контрастирование;
 - b. Нелинейное контрастирование;
 - c. Препарирование изображения.
3. Фильтрация изображения: методы пространственной области, основанные на использовании фильтров (масок свертки, шаблонов, окон).
4. Пространственные процессы, не подпадающие под категорию свертки.
5. Пути устранения краевых эффектов в процессе пространственной обработки изображений.
6. Классификация «шума» на изображении.

Лабораторная работа № 2

КЛАСТЕРНЫЙ АНАЛИЗ В РАСПОЗНАВАНИИ ОБРАЗОВ

Цель работы: ознакомиться с интерпретацией многомерных образов в виде кластеров. Освоить статистическое описание образов в многомерном пространстве признаков. Исследовать особенности распознавания образов с использованием различных метрик.

Теоретические сведения

2.1. Выделение связных областей

Есть два метода выделения связных областей:

- Рекурсивный алгоритм
- Итеративный метод (алгоритм последовательного сканирования)

Допустим, мы имеем черно-белое изображение. Таковым будем называть изображение, состоящее из точек только двух цветов: цвета фона и цвета объектов. Чаще всего обозначают 0 – фоновый цвет, 1 – цвет объекта. В таком случае выделенные области всегда будут одинаковыми при неизменном типе связности (4- или 8- связном).

Рекурсивный алгоритм

```
void Labeling(BIT* img[], int* labels[])
{
    // labels должна быть обнулена
    L = 1;
    for(y = 0; y < H; y++)
        for(x = 0; x < W; x++)
        {
            Fill(img, labels, x, y, L++);
        }
}

void Fill(BIT* img[], int* labels[], int x, int y, int L)
{
    if( (labels[x][y] == 0) && (img[x][y] == 1) )
    {
        labels[x][y] = L;
        if( x > 0 )
            Fill(img, labels, x - 1, y, L);

        if( x < W - 1 )
            Fill(img, labels, x + 1, y, L);

        if( y > 0 )
            Fill(img, labels, x, y - 1, L);

        if( y < H - 1 )
            Fill(img, labels, x, y + 1, L);
    }
}
```

Алгоритм последовательного сканирования

1. если пиксель (x, y) помечен единицей на исходном бинарном изображении:
2. если пиксели $(x - 1, y)$, $(x, y - 1)$ не отнесены ни к одной из
3. найденных областей:
4. прибавить счетчик областей $n = n + 1$;
5. приписать пиксель (x, y) к области n ;
6. если только один из пикселей $(x - 1, y)$, $(x, y - 1)$ отнесен к одной из
7. областей i :
8. приписать пиксель (x, y) к области i ;
9. если оба пикселя $(x - 1, y)$, $(x, y - 1)$ отнесены к найденной
10. области i :
11. приписать пиксель (x, y) к области i ;
12. если оба пикселя $(x - 1, y)$, $(x, y - 1)$ отнесены к разным
13. областям i, j : $i \neq j$:
14. приписать пиксель (x, y) к области i ;
15. зафиксировать эквивалентность областей i и j ;
16. перейти к следующему пикселю.

После проведения сканирования производится разрешение эквивалентностей областей и переразметка изображения с учетом этих эквивалентностей. После того, как на изображении обнаружены связные области пикселей, предположительно являющиеся объектами, производится проверка формы этих областей на удовлетворение определенным ограничениям: пропорции, соотношение периметра и площади, эксцентриситет и другие.

Псевдокод:

```
if A = 0
    do nothing
else if (not B labeled) and (not C labeled)
    increment label numbering and label A else if B xor C labeled
    copy label to A
else if B and C labeled
    if B label = C label
        copy label to A
    else
        copy either B label or C label to A
record equivalence of labels
```

0 – фон; 1 – цвет объекта.

2.2.

Вычисление признаков

Геометрические признаки

Каждый физический объект обладает набором некоторых свойств, которые позволяют отличить его от других объектов (рис. 2.1).



Рис. 2.1. Различные объекты

Совокупность свойств, описывающих конкретный объект, называется образом данного объекта. Под классом объектов понимается некоторая совокупность образов, называемых элементами класса, обладающая рядом близких свойств. Измеряемые или вычисляемые свойства объектов, позволяющие отличить классы друг от друга, называются признаками.

Так на изображении объекты (области) можно отличить, используя следующие признаки:

1. Текстурные признаки;
2. Геометрические признаки;
3. Фотометрические признаки и др.

Для каждой области можно подсчитать некий набор простейших числовых характеристик:

- Площадь;
- Центр масс;
- Периметр;
- Компактность;
- Ориентацию главной оси инерции;
- Удлиненность (эксцентриситет).

На основе этих характеристик можно классифицировать получаемые объекты (области).

- **Площадь** – количество пикселей в области (рис. 2.2).

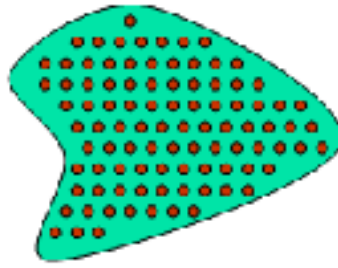


Рис. 2.2. Площадь объекта

$$A = \sum_{x=0}^m \sum_{y=0}^n I(x, y) \quad (2.1)$$

- **Центр масс**

$$\bar{x} = \frac{\sum_{x=0}^m \sum_{y=0}^n xI(x, y)}{A}, \quad \bar{y} = \frac{\sum_{x=0}^m \sum_{y=0}^n yI(x, y)}{A} \quad (2.2)$$

- **Периметр** – количество пикселей, принадлежащих границе области (рис. 2.3).

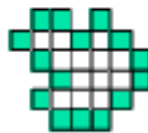


Рис. 2.3. Периметр объекта

Периметр зависит от того, какую связность (4-х или 8-ми) используют при определении соседей. Также отличают внутреннюю и внешнюю границу объекта (рис. 2.4).

1. Пиксель лежит на границе области, если он сам принадлежит области, а хотя бы один из его соседей области не принадлежит (внутренняя граница).

2. Пиксель лежит на границе области, если он сам не принадлежит области, а хотя бы один из его соседей области принадлежит (внешняя граница).

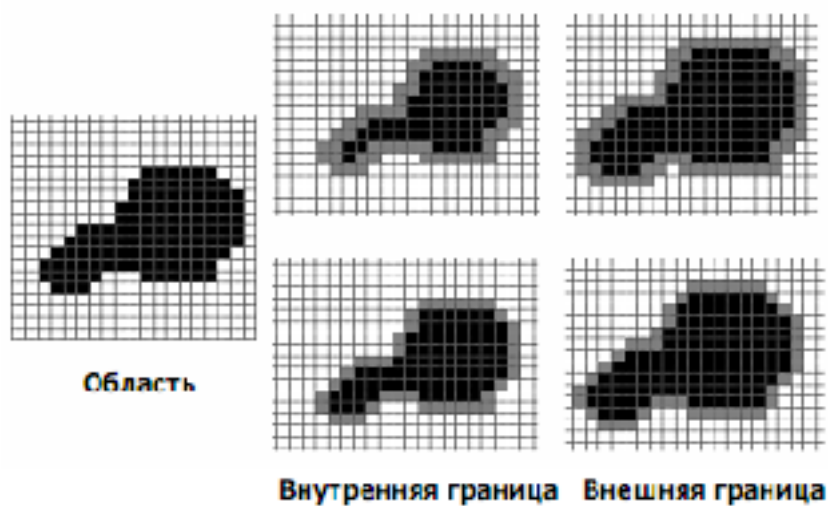


Рис. 2.4. Граница объекта

Для распознавания, как правило, интересуют признаки, инвариантные к изменениям масштаба, переносу и повороту. Такими признаками являются удлинённость и компактность.

- **Компактность** – отношение квадрата периметра к площади (рис. 2.5):

$$C = \frac{P^2}{A} \quad (2.3)$$

Наиболее компактная фигура – круг: $C = 4\pi$.



Рис. 2.5. Компактность фигуры

Дискретный центральный момент m_{ij} области определяется следующим образом:

$$m_{ij} = \sum_{x,y \in S}^n (x - \bar{x})^i (y - \bar{y})^j I(x, y) \quad (2.4)$$

где \bar{x}, \bar{y} – центр масс области.

- **Удлиненность, нецентрированность (компактность)** (рис. 2.6):

$$elongation = \frac{m_{20} + m_{02} + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{m_{20} + m_{02} - \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}} \quad (2.5)$$

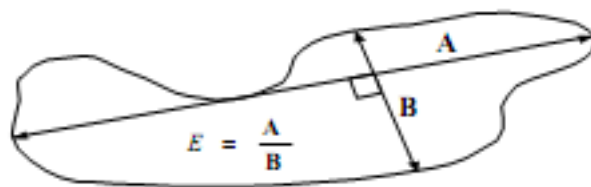


Рис. 2.6. Удлиненность фигуры

- **Ориентация главной оси инерции** не является инвариантной к повороту, но в ряде случаев предоставляет полезную информацию об ориентации объекта (рис. 2.7):

$$\theta = \frac{1}{2} \arctan\left(\frac{2m_{11}}{m_{20} - m_{02}}\right) \quad (2.6)$$

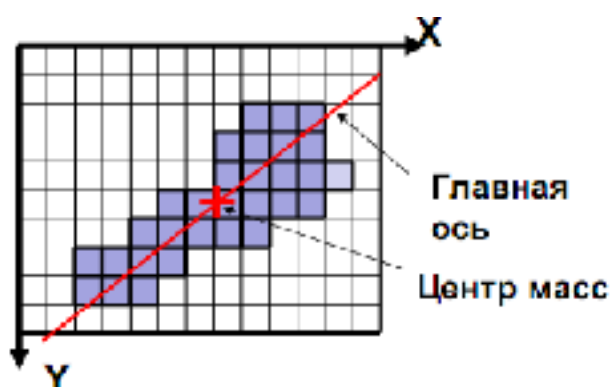


Рис. 2.7. Ориентация главной оси инерции объекта

Фотометрические признаки

Для каждой области можно подсчитать некий набор простейших числовых характеристик:

- Средняя яркость;
- Средний цвет (для цветного изображения);
- Гистограмма распределения яркостей (или для цветного изображения три гистограммы R, G, B);
- Дисперсия (разброс) яркости или цвета.

2.3.

Кластерный анализ

Пусть число признаков равно n , тогда каждый образ можно представить в виде некоторого набора значений признаков или вектора признаков вида:

$$x = (x_1, x_2, \dots, x_n), \quad (2.7)$$

где x_i – значение i -го признака данного образа.

В многомерном пространстве, осями координат которого являются отдельные признаки, каждый образ определяется точкой, причем расстояние от этой точки до начала координат и расстояние между точками определяется выбранной метрикой. Каждый класс может характеризоваться некоторой областью в n -мерном пространстве признаков. Эта область определяется степенью отличия образов, относящихся к данному классу, и может задаваться границами, например плоскостями. Область, в которую с наибольшей вероятностью попадают векторы образов данного класса, обычно называют кластером, а процедуру разбиения пространства признаков на области, соответствующие разным классам, – кластеризацией совокупности образов.

Если образы, относящиеся к одному классу, имеют различные значения признаков, то для описания класса можно использовать статистические характеристики: вектор средних значений или вектор математических ожиданий по всем признакам и вектор среднеквадратичных отклонений или дисперсий признаков. Для многомерных векторов принято использовать матрицу ковариации, на диагонали которой находятся дисперсии соответствующих признаков. Статистические характеристики признаков класса вычисляются следующим образом. Пусть набор n -мерных векторов, полученных в результате предъявления распознающей системе m объектов, относящихся к одному классу, задан в виде таблицы

x_{11}	x_{12}	x_{1m}
...
x_{n1}	x_{n2}	x_{nm}

(2.8)

тогда

$$m = (\mu_1, \mu_2, \dots, \mu_n) : \mu_i = \frac{1}{m} \sum_{k=1}^m x_{ik} \quad (2.9)$$

$$Cov = \begin{pmatrix} D_{11} & D_{12} & \dots & D_{1n} \\ D_{21} & D_{22} & \dots & D_{2n} \\ \dots & \dots & \dots & \dots \\ D_{n1} & D_{n2} & \dots & D_{nn} \end{pmatrix} : \begin{cases} D_{ii} = \frac{1}{m} \sum_{k=1}^m (x_{ik} - \mu_i)^2 \\ D_{ij} = \frac{1}{m} \sum_{k=1}^m (x_{ik} - \mu_i)(x_{jk} - \mu_j), \end{cases} \quad (2.10)$$

где $i, j = 1, \dots, n$ – индексы номеров компонент вектора признаков; m – число образов, составляющих данный класс; x_{ik} – значение i -го признака k -го образа; μ_i – математическое ожидание i -ой компоненты вектора признаков; D_{ii} – дисперсия i -го признака; D_{ij} – коэффициент ковариации i -го и j -го признаков; μ – вектор математических ожиданий; Cov – ковариационная матрица.

Ковариационная матрица симметрична относительно главной диагонали и, следовательно, необходимо вычислять только половину ее элементов. Ковариация характеризует степень линейной зависимости случайных величин. Если ковариация равна нулю, то величины называются некоррелированными.

Таким образом, процесс распознавания включает в себя этап обучения, т.е. определения некоторых характеристик по обучающей репрезентативной выборке образов, отнесенных к известным классам, и собственно распознавание на основе информации, полученной при обучении.

Под распознаванием образа понимается отнесение его к тому классу, расстояние до которого от данного образа в пространстве признаков минимально. Для определения расстояния между точками в метрическом пространстве признаков ранее было введено понятие метрики.

Методы распознавания можно разделить на две группы. К первой группе относятся методы, основанные на последовательном вычислении расстояний между распознаваемым образом и центрами кластеров в пространстве признаков с той или иной метрикой, например Евклидово расстояние. Образ считается принадлежащим классу, расстояние до которого минимально.

Другая группа методов предполагает предварительное разбиение признакового пространства плоскостями или поверхностями более высокого порядка на области, соответствующие известным классам, и определение принадлежности распознаваемого образа одной из областей.

Выбор конкретного метода зависит от решаемой задачи. Существует около 100 разных алгоритмов кластеризации, однако чаще используется иерархический кластерный анализ и кластеризация методом k -средних (k -means).

Алгоритмы k-means и k-medians

Алгоритм k-средних (k-means) – алгоритм кластеризации данных. Целью задачи кластеризации является разбиение множества объектов на кластеры (классы) на основе некоторой меры сходства объектов.

Дано:

Набор векторов $x_i, i = 1, \dots, p$;

k – число кластеров, на которые нужно разбить набор x_i .

Найти:

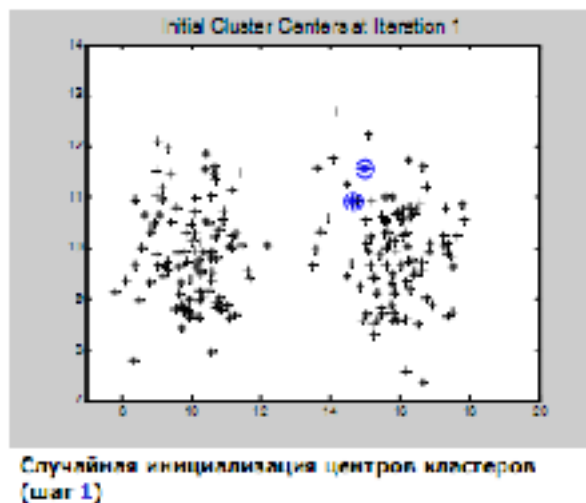
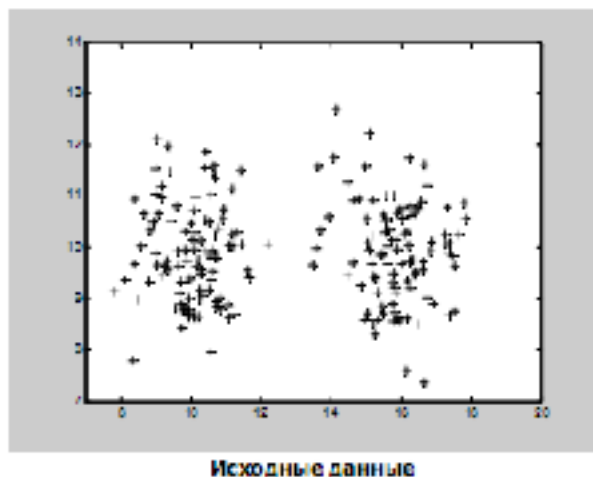
k средних векторов $m_j, j = 1, \dots, k$ (центров кластеров);

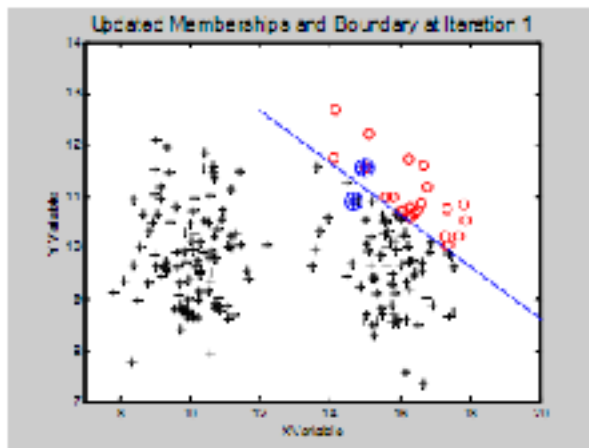
отнести каждый из векторов x_i к одному из k кластеров;

Алгоритм:

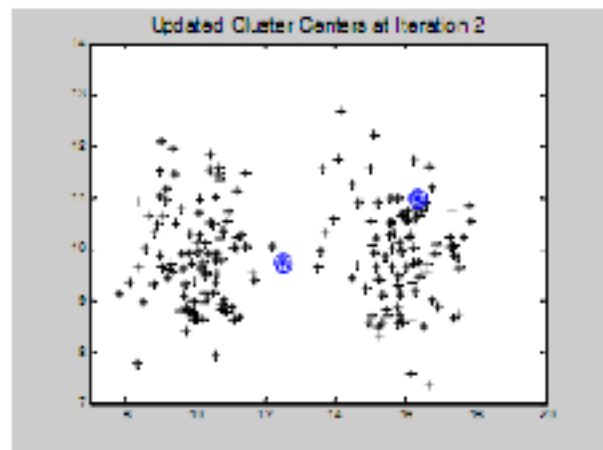
1. Случайным образом выбрать k средних $m_j, j = 1, \dots, k$;
2. Для каждого $x_i, i = 1, \dots, p$ подсчитать расстояние до каждого из $m_j, j = 1, \dots, k$, Отнести (приписать) x_i к кластеру j' , расстояние до центра которого $m_{j'}$ минимально;
3. Пересчитать средние (центр масс) $m_j, j = 1, \dots, k$ по всем кластерам;
4. Повторять шаги 2, 3, пока кластеры не перестанут изменяться.

Пример кластеризации в 2D.

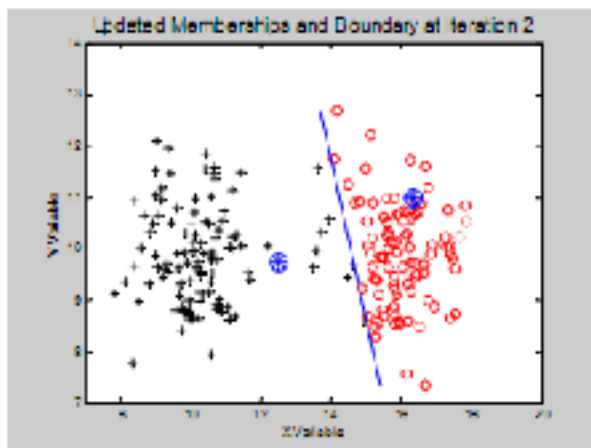




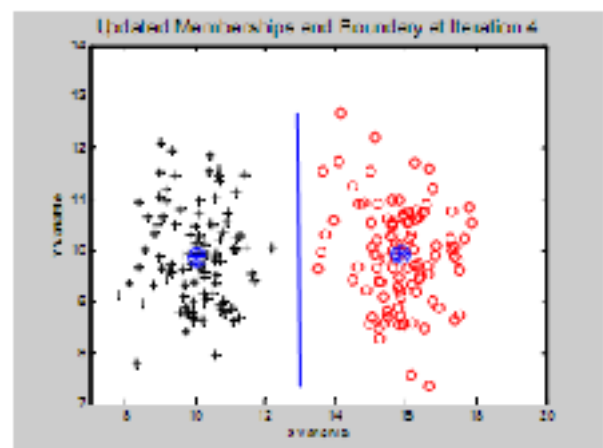
Кластеры после первой итерации (шаг 2)



Пересчет центров кластеров после первой итерации (шаг 3)



Кластеры после второй итерации (шаг 2)



Стабильная конфигурация после четвертой итерации

Алгоритм k-medians - это вариация k-means метода кластеризации, где для определения центра кластера вместо среднего вычисляется медиана (по каждому из измерений).

Алгоритм кластеризации k-medoids похож на алгоритм k-means, но в отличие от него на каждой итерации ищет центры кластеров не как среднее точек, а как медоиды точек. То есть, центр кластера должен обязательно являться одной из его точек. Медоидом для множества точек называется одна из точек множества, сумма расстояний до которой от всех точек множества минимальна.

Алгоритм k-medoids, в отличие от k-means, использует для представления центра кластера не центр масс, а представительный объект – один из объектов кластера. Как и в методе k-means, сначала произвольным образом выбирается k представительных объектов. Каждый из оставшихся объектов объединяется в кластер с ближайшим представительным объектом. Затем итеративно для каждого представительного объекта производится его замена произвольным не-представительным объектом пространства данных. Процесс замены продолжа-

ется до тех пор, пока улучшается качество результирующих кластеров. Качество кластеризации определяется суммой отклонений между каждым объектом и представительным объектом соответствующего кластера, которую метод стремится минимизировать. То есть, итерации продолжаются до тех пор, пока в каждом кластере его представительный объект не станет медоидом – наиболее близким к центру кластера объектом.

Задание для лабораторной работы

1. При необходимости выполнить коррекцию яркости изображения (см. лаб. работу №1).
2. Перевести цветное изображение в бинарное изображение (см. лаб. работу №1).
3. Реализовать пороговую бинаризацию изображения (см. лаб. работу №1).
4. Выделить связные области на изображении (рекурсивный алгоритм или алгоритм последовательного сканирования).
5. Определить свойства объектов, вычислить систему признаков для объектов, представленных на изображении (площадь, периметр, компактность, вытянутость, статистические моменты). Вычисление всех признаков обязательно. Проанализировать, какие из признаков являются наиболее информативными.
6. Используя алгоритм k-means, k-medians либо k-medoids, определить принадлежность объекта к одному из кластеров (классов). Реализация данного алгоритма обязательна.

№ варианта	Алгоритм выделения связных областей	Алгоритм кластеризации
1	Рекурсивный	k-means
2	Последовательного сканирования	k-medians
3	Рекурсивный	k-medoids
4	Последовательного сканирования	k-means
5	Рекурсивный	k-medians
6	Последовательного сканирования	k-medoids

Тестовые базы:

1. Преподавателем будет выдана тестовая база, в которой будут два класса изображений.
2. По желанию студент может реализовать данный алгоритм для своих изображений.



Рис. 2.6. Примеры тестовых изображений

Контрольные вопросы

1. Геометрические признаки при описании объектов.
2. Фотометрические признаки при описании объектов.
3. Кластерный анализ.
4. Алгоритм k-means.
5. Алгоритм k-medians.
6. Алгоритм k-medoids.

Литература

1. Яне, Б. Цифровая обработка изображений / Б. Яне. – М. : Техносфера, 2007.
2. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2006.
3. Дуда, Р. Распознавание образов и анализ сцен / Р. Дуда, П. Харт. – М. : Мир, 1976.
4. Павлидис, Т. Алгоритмы машинной графики и обработки изображений / Т. Павлидис. – М. : Радио и связь, 1986.
5. Прэтт, У. Цифровая обработка изображений. В 2 т. / У. Прэтт. – М. : Мир, 1982.

Учебное издание

Лукашевич Марина Михайловна
Садыхов Рауф Хосровович

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ

Лабораторный практикум для студентов специальностей I-40 02 01
«Вычислительные машины, системы и сети»
всех форм обучения

Редактор М.П. Острикова
Корректор

Подписано в печать	Формат 60×84 1/16.	Бумага офсетная.
Усл. печ. л.	Печать ризографическая.	Гарнитура «Таймс».
Уч.-изд. л. 2,1	Тираж 100 экз.	Заказ.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0494371 от 16.03.2009. ЛП №02330/0494175 от 03.04.2009.
220013, Минск, П. Бровки, 6