Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Емельянов Сергей Геннадьевич
Должность: ректор
Дата подписания: 16.12.2021 20:54:38
Уникальный программный ключ:
9ba7d3e34c012eba476ffd2d064cf2781953be730df2374d16f3c0ce5338f0f6

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
«___» _____ 2018 г.

МЕТОДЫ АНАЛИЗА И КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ ДЛЯ МЕДИЦИНСКИХ ДИАГНОСТИЧЕСКИХ СИСТЕМ, МЕТОДЫ АНАЛИЗА И КЛАССИФИКАЦИИ СЛОЖНОСТРУКТУРИРУЕМЫХ ИЗОБРАЖЕНИЙ

Методические рекомендации по организации и выполнению практических занятий для аспирантов направлений подготовки 09.06.01 и 12.06.01

Курск 2018

УДК 004.93:61

Составитель: С.А. Филист.

Рецензент
Доктор технических наук, профессор Р.А. Томакова

Методы анализа и классификации изображений для медицинских диагностических систем, методы анализа и классификации сложноструктурируемых изображений: методические рекомендации по организации и выполнению практических занятий / Юго-Зап. гос. ун-т; сост.: С.А. Филист. - Курск, 2018. - 42 с.

Методические указания по структуре, содержанию и стилю изложения материала соответствуют требованиям, предъявляемым к учебным и методическим пособиям.

Предназначены для аспирантов направлений подготовки 09.06.01 «Информатика и вычислительная техника (Системный анализ, управление и обработка информации (технические и медицинские системы)) и 12.06.01 «Фотоника, приборостроение, оптические и биотехнические системы и технологии (Приборы, системы и изделия медицинского назначения)»

Текст печатается в авторской редакции

Подписано в печать 1.03.18. Формат 60x84 1/16.
Усл.печ.л. 2,4. Уч.-изд.л. 2,2 Тираж 100 экз. Заказ 144. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Практическая работа №1 «Локальные методы обработки полутоновых изображений в среде MATLAB»

Цель работы: – изучение типов изображений, функций (средств) системы MatLab, применяемых для работы с файлами изображений и приобретение практических навыков их использования.

Краткие теоретические сведения

Типы изображений

По способу сохранения описание изображения может быть:

Векторным – изображение описывается как набор графических примитивов, из которых и формируется изображение;

Растровым – изображение описывается двумерным массивом, каждый элемент которого представляет собой некоторое описание цвета.

Элемент растрового изображения называют пикселом (от pixel, picture element – элемент изображения), или точкой.

Объем памяти в байтах, необходимый для хранения растрового изображения, можно вычислить по формуле:

$$V = \frac{(c \cdot r \cdot d)}{8};$$

где c – количество столбцов; r – количество строк; d – глубина цвета (бит/ пиксел).

Существуют следующие типы изображений:

Бинарные – (black and white) пикселы могут принимать только два значения: 0 и 1 (черный и белый цвет);

Полутоновые (серые или изображениями в градациях серого intensity, grayscale) – пикселы могут принимать одно из значений интенсивности какого-либо одного цвета в диапазоне от минимальной до максимальной интенсивности;

Палитровые (indexed) – значения пикселов являются ссылками на ячейки карты цветов (colormap), которые и содержат описания цвета пиксела в некоторой цветовой системе (палитре);

Полноцветные или просто цветные (truecolor, rgb) – изображения, пикселы которого непосредственно хранят информацию об интенсивностях цветовых составляющих.

Пикселы изображений, представленных массивами в формате double и uint8, должны удовлетворять требованиям, представленным в таблице 1.

Таблица 1 –Диапазоны представления пикселов изображения

Тип изображения	double	uint8
Бинарное	Значения 0 и 1	Значения 0 и 1
Полутоновое	Диапазон значений [0, 1]	Диапазон значений [0, 255]
Палитровое	Диапазон значений [1, размер палитры], значение 1 указывает на первую строку	Диапазон значений [0, 255], значение 0 указывает на первую строку палитры
Полноцветное	Диапазон значений [0, 1]	Диапазон значений [0, 255]

Полутоновые и бинарные изображения хранятся в виде двумерных массивов. Для доступа к значению (в данном случае яркости) пиксела изображения I надо указать строку r и столбец c: I(r, c).

Полноцветные изображения хранятся в виде трехмерных массивов, где третье измерение – значения интенсивности R, G, B. Для доступа к значениям интенсивности цветовых составляющих пиксела I надо указать строку r, столбец c и номер составляющей: 1 – для R, 2 – для G, 3 – для B, например, I(r, c, 1) позволяет получить значение красной составляющей.

Палитровые изображения хранятся в виде двумерных массивов индексов. Для каждого палитрового изображения существует двумерный массив палитры. Массив палитры всегда имеет тип double, и в трех его столбцах расположены интенсивности R, G, B. Пример палитрового изображения, использующего формат представления данных uint8, приведен на рисунке 1.

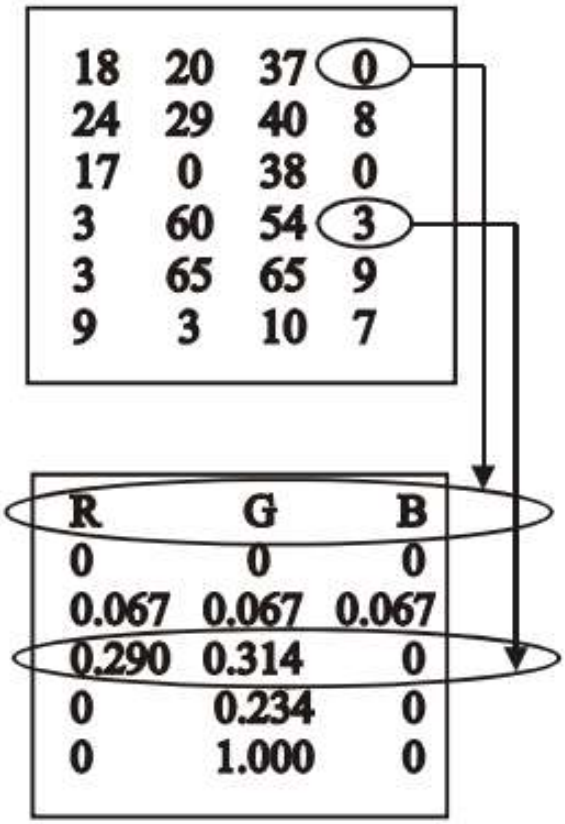

Рисунок 1

Принятые обозначения

В описании функций и в примерах применяются следующие обозначения для различных типов изображений:

I – полутоновые; X – палитровые; RGB – полноцветные; BW – бинарные; S – для исходного изображения любого типа; D – результирующее изображение.

Символ «многоточие» (…) в описании функции означает, что может быть использован любой упомянутый ранее в описании набор входных или выходных параметров.

В системе MatLab функции по обработке изображения находятся в пакете Image Processing Toolboxes (IPT).

Работа с файлами изображений Функция чтения из файла информации об изображении imfinfo

Синтаксис

info = imfinfo(<имя файла>

В структуре info возвращается информация об изображении и способе его хранения из файла с именем <имя файла>; имя

включает в себя путь к файлу, его имя и расширение, например, 'c:\mmm\Earth.bmp'.

Графические форматы файлов, с которыми система MatLab поддерживает работу, приведены в таблице 2.

Таблица 2 - Форматы файлов изображений

Формат файла	Название формата
'bmp'	Windows Bitmap (BMP)
'tif или 'tiff	Tag Image File Format (TIFF)
'jpg' или 'jpeg'	Joint Photographic Experts Group

Информация об изображении и способе его хранения в данном файле возвращается в структуре info. Структуры для разных форматов файлов отличаются друг от друга. Общие для всех форматов первые 9 полей структуры, по которым можно определить формат файла, тип и размеры изображения, приведены в таблице 3.

Таблица 3 - Описание первых девяти полей структуры info

Имя поля	Тип	Описание
Filename	Строка	Имя файла, если файл находится в текущей директории, или полный путь к
FileModDat	Строка	Дата и время последней модификации
FileSize	Число	Размер файла в байтах
Format	Строка	Формат файла
FormatVersi	Строка или	Версия формата
Width	Число	Ширина изображения в пикселах
Height	Число	Высота изображения в пикселах
BitDepth	Число	Глубина цвета изображения в битах на
ColorType	Строка	Тип изображения: 'truecolor' или 'RGB' – для полноцветных изображений; 'grayscale' – для полутоновых изображений; 'indexed' – для палитровых изображений

В файлах форматов TIFF и HDF может храниться несколько изображений. В этом случае info является массивом структур.

Порядок выполнения работы

1. Получить информацию об изображении, хранящемся в файле на диске c:

>> f = imfinfo(‘c:\Image\Athena.bmp’).

Функция чтения изображения из файла imread Синтаксис

D = imread(<имя файла>) – читает из файла <имя файла> непалитровое изображение и помещает его в массив D, (<имя файла>) включает в себя путь к файлу, его имя и расширение, например, 'c:\mmm\Athena.bmp'.

[X, map] = imread(<имя файла>) – читает из файла с именем <имя файла> палитровое изображение X с палитрой map.

Прочитанное из файла изображение имеет формат представления данных uint8.

2. Прочитать палитровое изображение из файла Handshak.bmp.

>> [X,map] = imread('c:\Image\Handshak.bmp');

Функция записи изображения в файл imwrite Синтаксис

imwrite(S, <имя файла>) – записывает в файл с именем <имя файла> бинарное, полутоновое или полноцветное изображение S.

imwrite(X, map, <имя файла>) – записывает в файл с именем <имя файла> палитровое изображение X с палитрой map.

Вывод изображения на экран

Функция вывода изображения на экран imshow

Синтаксис

imshow(S) – вывод непалитрового изображения.

imshow(I,n) – вывод полутонового изображения I, по умолчанию n = 256. **imshow(I,[low high])** –вывод полутонового изображения с контрастированием.

imshow(X,map) – вывод палитрового изображения X, map – палитра. **imshow (<имя файла>)** – вывод изображения непосредственно с диска. Функция imshow(I,[low high]) выводит на экран полутоновое изображение I, дополнительно контрастируя выводимое изображение. Пикселы, яркость которых

меньше/равна low, отображаются черным цветом, больше/равна high – белым. Все уровни серого равномерно распределены от low до high. Если вторым параметром указать пустой массив [], то low = = min(I(:)), high = max(I(:)).

3. Вывести на экран прочитанное палитровое изображение из предыдущего примера, используя команду:

>> imshow(X,map).

Вывод нескольких изображений в одном окне

В сочетании с функцией subplot для вывода нескольких изображений в одном окне используется функция subimage:

Синтаксис

subimage(S) – вывод в графическое окно бинарного, полутонового или полноцветного изображения.

subimage(X,map) – вывод в графическое окно палитрового изображения X с палитрой map.

Преобразования классов данных и типов изображений

Классы данных представления изображений

Двумя основными классами данных (форматами) представления элементов массива изображений являются:

double – в виде действительных чисел двойной точности; каждый элемент формата double занимает 8 байт памяти;

uint8 – значение пиксела есть беззнаковое целое однобайтовое число, диапазон возможных значений которого [0, 255].

Для представления изображения в формате double используется функция im2double.

Синтаксис

D = im2double(S) – преобразует бинарное, полутоновое или полноцветное изображение S в формат double и осуществляет приведение значений пикселов к диапазону [0,1].

X_D = im2double(X_S,'indexed') – преобразует палитровое изображение X_S в формат double.

Функция mat2gray позволяет преобразовывать произвольный массив S формата double в перенормированный D.

Изображение D имеет значения пикселов в интервале от 0 (белый) до 1 (черный).

Синтаксис

D = mat2gray(S)

Для представления изображения в формате uint8 используется функция im2uint8.

Синтаксис

D = im2uint8 (S) – преобразует бинарное, полутоновое или полноцветное изображение S в формат uint8; представляет все пикселы в виде целых неотрицательных чисел в диапазоне [0, 255].

X_D = im2uint8 (X_S, 'indexed') – преобразует палитровое изображение X_S в формат uint8.

Преобразования типов изображений

Существуют ряд функций для преобразований изображений из одного типа в другой:

I = rgb2gray(RGB) – преобразование полноцветного изображения в полутоновое;

I = ind2gray(X,map) – преобразование палитрового изображения в полутоновое;

[X map] = gray2ind(I,n) – преобразование полутонового изображения в палитровое, n по умолчанию равно 64;

RGB = ind2rgb(X,map) – преобразование палитрового изображения в полноцветное;

[X map] = rgb2ind(RGB) – преобразование полноцветного изображения в палитровое.

Содержание 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 (компьютерная распечатка работ);
5. Выводы;

Контрольные вопросы

1. Какие классы данных (форматы) представления пикселов изображения существуют?
2. Какие типы растровых изображений используются в пакете IPT?
3. С помощью какой функции можно получить информацию о размере, типе изображения?
4. С какими форматами графических файлов можно работать в системе MatLab?
5. С помощью каких функций можно прочитать изображение из файла на диске и записать изображение на диск?
6. Какие аргументы функции imshow изменяют контраст полутонового изображения при его выводе на экран?
7. Какие вы знаете функции преобразования типов изображений.

Практическая работа №2 «Спектральные методы обработки полутоновых изображений в среде MATLAB»

Цель работы – изучение методов спектрального анализа изображений и приобретение практических навыков их использования.

Краткие теоретические сведения

Функции преобразования Фурье 1. Прямое преобразование Фурье Синтаксис

Y = fft2(X)

Y = fft2(X,m.n)

Функция Y = fft2(X) – двумерное быстрое преобразование Фурье (БПФ) – возвращает результат в матрице комплексных чисел Y, имеющей тот же размер, что и матрица X.

Функция Y = fft2(X,m.n) вычисляет БПФ матрицы m на n, при необходимости дополняя нулями или усекая исходную матрицу X. Возвращается матрица Y размером m на n.

Массив Y имеет формат представления данных double.

Алгоритм: двумерное преобразование Фурье вычисляется выполнением двух одномерных преобразований fft(fft(x).')'. Сначала вычисляется ДПФ каждого столбца, а затем каждой строки результата.

2. Обратное двумерное БПФ Синтаксис

Y = ifft2(X)

Y = ifft2(X,m,n)

Функция Y = ifft2(X) вычисляет обратное двумерное ДПФ, возвращая результат в матрице Y, имеющий тот же размер, что и матрица X.

Функция Y = ifft2(X,m,n) вычисляет обратное двумерное преобразование Фурье матрицы m □□n, при необходимости дополняя нулями или усекая исходную матрицу X. Возвращается матрица Y размером m на n.

Порядок выполнения работы

1. Необходимо выполнить следующие действия:

Сформировать двумерный сигнал, являющийся суммой двух пространственных волн, I = sin(2*pi*x/8)+sin(2*pi*y/16); вывести его на экран:

```
>> [x,y] = meshgrid(1:32);
>> I = sin(2*pi*x/8)+sin(2*pi*y/16);
>> figure,imshow(mat2gray(I);
>> figure,surf(x,y,I);
>> colormap(gray)
>> shading interp
```

Вычислить спектр сигнала и вывести его на экран (так как спектр симметричен, четыре отчетливых пика соответствуют двум частотам волн, из которых сформировано исходное изображение).

```
>> h = fft2(I
>> figure,surf(x,y,abs(h));
>> colormap(gray)
>> shading interp
```

Удалить из спектра одну из частот и восстановить сигнал с помощью обратного преобразования Фурье:

```
>> h(1:32,1:2) = 0;
>> figure,surf(x,y,abs(h));
>> colormap(gray)
>> shading interp
>> I = ifft2(abs(h));
>> figure,imshow(mat2gray(I));
>> figure,surf(x,y,abs(I));
>> colormap(gray)
>> shading interp
```

Сокращение избыточности изображения

Избыточность информации может быть описана функцией корреляции между отсчетами изображения. Она проявляется во взаимной статистической прогнозируемости близлежащих отсчетов, взятых с изображения. Конечной целью операций

сжатия является устранение этой статистической прогнозируемости, то есть необходимо уменьшить коррелируемость отсчетов. Фурье-преобразование изображений переводит изображение в новую систему координат, что приводит к уменьшению коррелируемости отсчетов, после чего основные принципы сжатия пространственных преобразований заключаются в избирательной передаче коэффициентов разложения, что приводит к уменьшению полосы частот.

Многие методы сокращения полосы частот могут быть проанализированы с точки зрения двумерной дискретизации. На рисунке 1 приведена схема двумерной дискретизации.

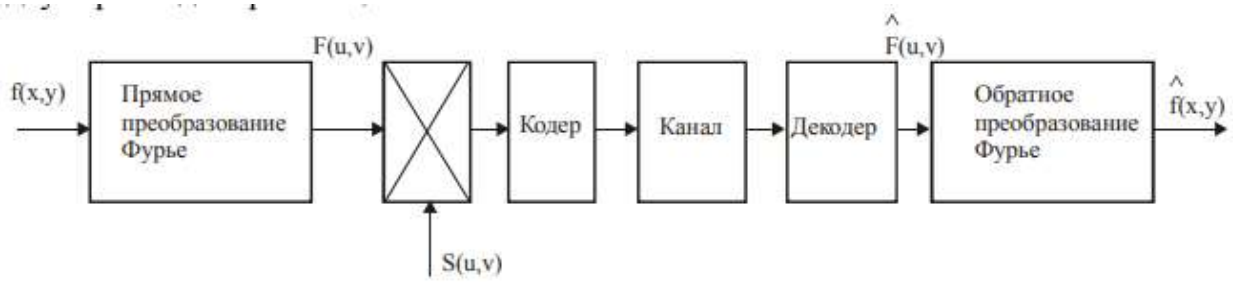

Рисунок 1 – Схема двумерной дискретизации

Для функции дискретизации типа шахматного поля, которая вычисляется по формуле

$$S(u,v)=\frac{1+(-1)^{v+u}}{2}$$

Восстановленное изображение (после обратного преобразования Фурье) состоит из исходного изображения и наложенного на него такого же изображения, сдвинутого по вертикали и горизонтали:

$$\hat{f}(x,y)=\frac{1}{2}\left[f(x,y)+f(x+\frac{N}{2},y+\frac{N}{2})\right]$$

2. Выполнить восстановление изображения после двумерной дискретизации с использованием функции дискретизации типа шахматного поля.

Пояснение. Изображение берется из файла 'Technlgy.bmp', преобразовывается в полутоновое. Для создания изображения, сдвинутого по вертикали и горизонтали, вырезается соответствующий фрагмент из исходного изображения (этого надо предварительно определить его размеры) и дополняется нулями до размера исходного изображения. Затем восстановленное изображения получается из сложения исходного и сдвинутого изображения.

```
>>[X,map] = imread('C:\Image\Technlgy.bmp');
>> I = ind2gray(X,map);
>> figure,imshow(I)
>> [M N] = size(I);
>> rect = [floor(N/2), floor(M/2), floor(N/2)–1, floor(M/2)–1];
>> I1 = imcrop(I,rect);
>> IM = zeros(floor(M/2),floor(N/2));
 >> IP = [I1,IM];
>> IN = zeros(floor(M/2),N);
>> IR = [IP;IN];
>> figure,imshow(IR)
>> D = I+IR;
>> figure,imshow(D)
```

Содержание 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 (компьютерная распечатка работ);
5. Выводы;

Контрольные вопросы

1. Какие функции используются для выполнения двумерного прямого и обратного преобразования Фурье в системе MatLab?

2. Зачем используется двумерная дискретизация? Приведите примеры функции дискретизации.

Практическая работа №3 «Сегментация полихроматических изображений в среде MATLAB»

Цель работы – изучение операций по сегментации изображений, функций, реализующих операции по сегментации изображения, и приобретение практических навыков использования этих функций.

Краткие теоретические сведения

Сегментация изображения представляет собой разделение изображения на области по сходству свойств (признаков) в их точках. Признаки подразделяются на естественные и искусственные. Естественные признаки устанавливаются простым (визуальным) анализом изображения, а искусственные – в результате специальной обработки различных измерений. Примерами естественных признаков являются структура, текстура, яркость объекта. Примеры искусственных признаков: гистограммы распределения яркости, спектр и др.

К основными видами сегментации изображений относится сегментация по яркости, цветовым координатам, контурам, форме.

Методы сегментации

Сегментация методом выращивания областей

Для сегментации изображения можно использовать метод выращивания областей – группирование пикселов или подобластей в более крупные области по заранее заданным критериям роста. Берутся «центры кристаллизации», а затем на них наращиваются области путем добавления к каждому центру тех соседних пикселов, которые по своим свойствамблизки к центру кристаллизации (например, имеют яркость или цвет в определенном диапазоне). Ниже приведена функция regiongrow, которая выполняет выращивание областей.

Синтаксис

[g, NR, SI, TI] = regiongrow(f, S, T),

где f – это сегментируемое изображение, а параметр S – массив (с размерами как у f) или скаляр. Если S – массив, то он содержит 1 в тех позициях, где расположены центры кристаллизации и 0 во всех остальных местах. Если S является скаляром, то он задает

значение яркости пикселов, которые становятся центрами кристаллизации. Аналогично, T может быть массивом (с размерами, как у f) или скаляром. Если T – массив, то его элементы являются локальными пороговыми значениями для f. Скаляр T определяет глобальный порог.

Порядок выполнения работы

1. Создать функцию regiongrow. function [g, NR, SI, TI] = regiongrow(f, S, T) if numel(S) == 1
SI = f == S; S1 = S
else
SI = bwmorph(S, 'shrink', Inf); J = find(SI);
S1 = f(J);
end
TI = false(size(f)); for K = 1:length(S1) seedvalue = S1(K);
S = abs(f - seedvalue) <= T; TI = TI | S;
end
[g, NR] = bwlabel(imreconstruct(SI, TI));

2. Выполнить сегментацию наращиванием областей для изображения, хранящегося в файле Finance.bmp, используя функцию regiongrow.

[x,map]= imread('c:\image\Finance.bmp'); I = im2double(ind2gray(x,map)); figure,imshow(I)

S = 0.9783; T = 0.0651; % эти значения находятся экспериментально (взяты из изображения)

[g, NR, SI, TI] = regiongrow(I, S, T); figure,imshow(TI)

Также для сегментации используется метод разделения. Функция MatLab, реализующая подобный алгоритм, приведена ниже.

Сегментация методом разделения

Изображение разбивается на непересекающиеся блоки, которые с помощью некоторого критерия проверяются на однородность.

Функция сегментации полутоновых изображений методом разделения qtdecomp

Синтаксис

A = qtdecomp(I,threshold,mindim)

Функция qtdecomp осуществляет сегментацию полутоновых изображений методом разделения. В функции qtdecomp каждый блок разбивается на 4 неперекрывающихся блока одинакового размера. На первом шаге алгоритма блоком считается все изображение. Мельчайшим по размерам является блок, который нельзя разделить на 4 блока одинакового размера, т. е. такой блок, у которого число строк или число столбцов нечетное. Таким образом, в функции qtdecomp рекомендуется использовать изображения с размерами, равными степеням двух. Функция A = qtdecomp(I,threshold,mindim) осуществляет сегментацию полутонового изображения I методом разделения и помещает результат в разреженный массив A (тип данных sparse MatLab). Элементам матрицы A(r,c), соответствующим координатам левых верхних углов блоков на исходном изображении I, присваиваются значения, определяющие размеры каждого блока. Блок считается однородным, если разница между максимальным и минимальным значением пикселов блока меньше параметра threshold. Параметр mindim определяет минимальный размер блока.

Функция получения блоков из квадро-дерева результатов сегментации qtgetblk

Синтаксис

[vals, idx] = qtgetblk(I, A, dim)

Функция возвращает в массив vals все блоки размером dim, получившиеся в результате сегментации полутонового изображения I с помощью функции qtdecomp. В параметре A передается разреженный массив, описывающий квадро-дерево с результатами сегментации. Координаты левых верхних углов блоков, помещенных в массив vals, находятся в векторе idx. Если нет ни одного блока размером dim, то всем возвращаемым параметрам присваиваются значения пустых vals матриц.

Функция замены блоков – результатов сегментирования qtsetblk

Синтаксис

I_D = qtsetblk(I_S, A, dim, vals)

Функция создает новое полутоновое изображение I_D, заменяя в исходном полутоновом изображении все блоки размера dim, полученные в результате сегментации с помощью функции qtdecomp, на блоки из массива vals. В параметре A передается разреженный массив, описывающий квадро-дерево с результатами сегментации. Данная функция используется для преобразования изображения в соответствии с результатами сегментации методом разделения.

Рассмотрим работу функции qtdecomp совместно с функциями qtgetblk и qtsetblk для полутонового изображения размера 8×8 пикселов. Формат представления данных – uint8. Будем считать, что блок изображения является однородным, если величина разброса яркостей пикселов в блоке не превышает 10 градаций яркости. Установим минимально возможный размер блока. В нашем случае он равен двум.

Будем считать, что к объекту относятся блоки, средняя яркость которых не превышает 50. Требуется изменить исходное изображение так, чтобы пикселам блоков, относящихся к объекту, было присвоено значение 1, а пикселам блоков, не относящихся к объекту, 0.

3. Выполнить сегментацию небольшого текстового изображения методом разделения.

Исходное изображение:

```
>> I = [ 10 11 10 15 20 25 47 51 11 14
17 13 27 29 52 55
12 13 11 10 24 47 56 60 13 14 11 13 49 54 74 77 15 16 43 48 79 82
87 86 17 18 45 50 85 80 80 84 29 51 50 59 80 83 83 85 59 61 58 61
81 85 86 88 ];
```

Сегментация методом разделения: размер минимального блока 2 на 2; блок считается однородным, если в его пределах яркость изменяется меньше, чем на 10 градаций.

```
>> A = qtdecomp(I,10,2);
```

Для удобства визуального анализа предварительно преобразуем разреженную матрицу A в обычную матрицу M с помощью функции full.

```
>> M = full(A) M =
```

4 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 2 0 2 0
4 0 0 0 0 0 0 0 0 0 0 0 2 0 2 0 0 0 0 0 0 0 0 0 0 0 0

В результате сегментации получили 2 блока размером 4 на 4 (левая верхняя и правая нижняя части изображения и 8 блоков размером 2 на 2).

Переберем в цикле все возможные размеры блоков: 8, 4, 2.

```
>> dim = 8;
>> while dim> = 2
% получить в переменной blocks все блоки размера dim. [blocks,idx] = qtgetblk(I,A,dim);
[x y n] = size(blocks);
% если блоки такого размера есть в квадро-дереве, if n>0
% то перебираем все блоки размера dim for j = 1:n
% если среднее значение яркости пикселов в пределах блока меньше 50 if (mean2(blocks(:,:,j))<50)
% то заменяем значения всех пикселов блока на 1, blocks(:,:,j) = ones(dim,dim);
else
% иначе заменяем значения всех пикселов на 0. blocks(:,:,j) = zeros(dim,dim);
end;
end
%
end
for
% устанавливаем новые значения всех пикселов размера dim I = qtsetblk(I,A,dim,blocks);
end;
%
end
if dim = dim/2;
end
%
end
while
% ссылка на 1 из списка литературы
% получившееся изображение I>>II=
1 1 1 11
```

```
1 1 0 0 1
1 1 1 1 1
0 0 1 1 1
1 1 1 0 0
1 1 1 1 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
```

4. Выполнить сегментацию реального изображения из файла cotton3.bmp

```
rgb = imread('c:\Image\cotton3.bmp');
I = im2double(rgb2gray(rgb));
figure,imshow(I)
T=graythresh(I);
A = qtdecomp(I,0.1,2);
dim = 8;
while dim >= 2
[blocks,idx] = qtgetblk(I,A,dim);
[x y n] = size(blocks);
if n>0
for j = 1:n
if (mean2(blocks(:,:,j))<T) blocks(:,:,j) = ones(dim,dim);
else
blocks(:,:,j) = zeros(dim,dim);
end
end
I1 = qtsetblk(I,A,dim,blocks);
end
dim = dim/2;
end
figure,imshow(I1)
```

Функция выбора интересующей области по цвету roicolor

Синтаксис

BW = roicolor(S,low,hig) BW = roicolor(S,v)

Для любого варианта вызова функции roicolor бинарное изображение формируется по следующему алгоритму: пикселу бинарного изображения BW(r, c) присваивается значение 1, если яркость пиксела S(r, c) исходного полутонового изображения или индекс S(r, c) палитрового изображения

принадлежит диапазону [low, high] или любому из значений вектора v. В противном случае BW(r, c) присваивается значение 0.

Функция выбора интересующей области по цвету roicolor

Синтаксис

BW = roicolor(S,low,high) BW = roicolor(S,v)

Для любого варианта вызова функции roicolor бинарное изображение формируется по следующему алгоритму: пикселу бинарного изображения BW(r, c) присваивается значение 1, если яркость пиксела S(r, c) исходного полутонового изображения или индекс S(r, c) палитрового изображения принадлежит диапазону [low, high] или любому из значений вектора v. В противном случае BW(r, c) присваивается значение 0.

5. Выбрать цветовые области из изображения файла chip.bmp, задавая индексы с помощью гистограммы.

```
>> [x,map]=imread('C:\Image\chip.bmp')
>> figure,imhist(x,map),title('histogramma')
>> figure,imshow(x,map),
>> bw=roicolor(x,9,12);
>> figure,imshow(bw),title('9 - 12')
 >> x1=immultiply(bw,x);
>> figure,imshow(x1,map),title('9 - 12')
>> bw=roicolor(x,3,8);
>> figure,imshow(bw),title('3 - 8')
>> x1=immultiply(bw,x);
>> figure,imshow(x1,map),title('3 - 8')
```

Яркостный срез

Этот метод помогает выделить определенный диапазон яркости

$$\forall A_{i,j}\ B_{i,j} = \begin{cases} 0 & A_B < A_{i,j} < A_H \\ A_{i,j} & A_H \le A_{i,j} \le A_B \end{cases} \text{ или}$$

$$\forall A_{i,j}\ B_{i,j} = \begin{cases} 0 & A_B < A_{i,j} < A_H \\ K & A_H \le A_{i,j} \le A_B \end{cases}$$

где АН – нижнее граничное значение определяемого диапазона яркости; Ав – верхнее граничное значение определяемого диапазона яркости.

Для выполнения яркостного среза можно использовать функцию impixel. *Синтаксис*

P = impixel(S,c,r)

Функция impixel возвращает значения красной, зеленой и синей составляющих цвета для заданных координат – c и r – векторов значений столбцов и строк.

6**.** Выполнить яркостный срез полноцветного изображения файла cotton3.bmp.

```
[img] = imread('c:\Image\cotton3.bmp');
[m,n,k] = size(img)
img = im2double(img);
R = zeros(m,n,3);
z = [0.1,0.8;0.1,0.8; 0.1,0.9];
for y = 1:m
for x = 1:n
b = impixel(img,x,y);
if ((b(1)>= z(1,1))&(b(1)<= z(1,2)))& ((b(2)>= z(2,1))&(b(2)<= z(2,2))) … & ((b(3)>= z(3,1))&(b(3)<= z(3,2)))
R(x,y,1) = b(1);
R(x,y,2) = b(2);
R(x,y,3) = b(3);
else
R(x,y,1) = 0;
R(x,y,2) = 0;
R(x,y,3) = 0;
end
end
end
figure, imshow(img)
figure, imshow(R
```

Содержание 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 (компьютерная распечатка работ);

5. Выводы;

Контрольные вопросы

1. В чем заключается сегментация изображения?
2. Какие признаки используются для сегментации?
3. В чем заключается метод выращивания областей, использующийся для сегментации изображения?
4. В чем заключается метод разделения, использующийся для сегментации изображения?
5. Что является входными параметрами функции сегментации методом разделения?
6. В чем заключается преобразование яркостного среза?
7. Какие параметры возвращает функция impixel?

Практическая работа №4 «Морфологические методы обработки полутоновых изображений в среде MATLAB»

Цель работы – изучение морфологических методов, функций по реализации морфологических операций, приобретение практических навыков использования этих функций

Краткие теоретические сведения

Рассмотрим несколько простых, но важных взаимосвязей между пикселами в цифровом изображении. Рассматриваемые понятия широко применяются в обработке бинарных изображений и морфологических операциях, при выделении объектов и вычислении их признаков. Для определенности в системе MatLab полагают, что пикселы со значениями, равными 1, относятся к объектам, а со значениями, равными 0, – к фону.

Соседние пикселы

Пиксел p с координатами (x, y) имеет четыре горизонтальных и вертикальных соседних пиксела с координатами: (x + 1, y), (x – 1, y), (x, y + 1), (x, y – 1). Эта группа пикселов x_1, x_3, x_5, x_7, называемая «четыре соседа p» или «квартетом соседей», обозначается через $N_4(p)$:

x4	x3	x2
x5	p	x1
x6	x7	x8

Данные четыре пиксела находятся на одном расстоянии от (x, y), а также некоторые из соседних пикселов p могут быть за пределами цифрового изображения, если (x, y) находится на границе изображения. Четыре диагональных соседних пиксела 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 точки вместе с четырьмя указанными выше называются восьми-соседями, или октетом соседей пиксела p и обозначаются через $N_8(p)$. Некоторые из точек $N_D(p)$ и $N_8(p)$ также могут выходить за пределы изображения, если (x, y) находится на границе изображения.

Связи

Объект называется четырехсвязный, если для каждого пиксела объекта среди квартета соседних пикселов существует хотя бы один, равный 1 и принадлежащий этому объекту.

Объект называется восьмисвязный, если для каждого пиксела объекта среди октета соседних пикселов существует хотя бы один, равный 1 и принадлежащий этому объекту.

Аналогичные определения связности можно ввести для фона. Четырехсвязность фона автоматически означает восьмисвязность объектов и наоборот. Среди связных областей объектов могут встречаться связные области из пикселов фона. Они называются дырами.

Морфологические операции

Теория морфологии (морфологии – наука о форме) рассматривает бинарное изображение в виде множеств его пикселов переднего плана (со значениями 1), которое лежит в пространстве Z^2 (двухмерное пространство). Над бинарными изображениями можно совершать операции как над множествами с помощью следующих логических операций MatLab:

OR (|) – логическое сложение

AND (&) – логическое умножение –

NOT (~) – отрицание

DIFFERENCE (X & ~Y) – разностью множеств X и Y

Логические операции MatLab, применяемые к бинарным изображениям, приведены в таблице 1.

Таблица 1 - Логические операции MatLab для бинарных изображений

Операции	Выражения MatLab для бинарных изображений	Имя
$X \cap Y$	X & Y	AND
$X \cup Y$	X | Y	OR
X^c	~X	NOT
X\Y	X & ~Y	DIFFERENCE

Одним из основных понятий математической морфологии является понятие структурообразующего, или структурного элемента. Структурный элемент B – это множество, состоящее из двух непересекающихся подмножеств B1 и B2, для которых определено общее начало.

Функция создания структурообразующего элемента strel

Функция strel строит структурообразующие элементы различных форм и размеров.

Синтаксис

se = strel(shape, parameters),

где se – структурообразующий элемент; shape – строка, задающая форму структурообразующего элемента; parameters – дополнительные параметры, которые уточняют информацию о его форме.

В таблице 2 приведены некоторые варианты задания формы структурообразующего элемента.

Таблица 2 - Способы определения структурообразующего элемента

Вызов функции strel	Форма структурообразующего элемента	Дополнительные параметры
se = strel('diamond', R)	ромб	R – расстояние от центра структурообразующего элемента до крайней точки ромба

se = strel('disk', R)	круг	R – радиус структурообразующего элемента
se = strel('line', LEN, DEG)	линейный элемент	LEN – длина, DEG – угол наклона линии против часовой стрелки от горизонтальной оси (в градусах)
se = strel('pair', OFFSET)	две точки: первая точка находится в центре, а вторая смещена от первой на вектор OFFSET	OFFSET – двумерный вектор с неотрицательными целыми компонентами
se = strel('rectangle', MN)	прямоугольник	MN – двумерный вектор с неотрицательными целыми компонентами. MN(1) – число строк, MN(2) – число столбцов
se = strel(NHOOD)	элемент произвольной формы	NH00D – матрица из нулей и единиц, задающая его форму

Результатом выполнения функции strel является так называемый strel-объект. Для его разложения используется функция getsequence.

Синтаксис

decomp = getsequence(se)

Структурообразующие элементы разложения можно получить, индексируя переменную decomp.

Морфологические операции дилатации и эрозии имеют основополагающее значение при морфологической обработке изображений. Операция эрозии «ужимает» или «утончает» объекты двоичных изображений. Операция дилатации «наращивает» или «утолщает» на объекты двоичных изображениях. Способ и степень этих преобразований контролируется формой структурного элемента.

Эрозия

Эрозия X по B состоит из пикселов с такими координатами, для которых сдвиг множества в эту точку не пересекается с

фоном изображения X. Эрозия выполняется функцией imerode пакета IPT.

Синтаксис

D = imerode(S, se),

где S – это двоичное или полутоновое изображение; se – матрица из нулей и единиц, которая определяет структурообразующий элемент.

Порядок выполнения работы

1. Выполнить эрозию изображения из файла erode.bmp. Удалить тонкие провода с изображения с сохранением всех остальных структур. Необходимо выбрать достаточно малый структурный элемент, чтобы он помещался внутри центрального квадрата, а также внутри толстых полосок у границ, и достаточно большим, чтобы он не помещался внутри удаляемых проводков.

```
>> [x,map] = imread('c:\image\erode.bmp');
>> I=im2double(ind2gray(x,map));
>> figure,imshow(I)
>> T=graythresh(I);
>> BW = I>T;
>> figure,imshow(I)
>> se = ones(15);
>> se(15,15) = 0;
>> se(15,1) = 0;
>> se(1,15) = 0;
>> se(1,1) = 0;
>> BW1 = imerode(BW,se);
>> figure,imshow(BW1)
>> se = ones(18);
>> se(18,18) = 0;
>> se(18,1) = 0;
>> se(1,18) = 0;
>> se(1,1) = 0;
>> BW1 = imerode(BW,se);
>> figure,imshow(BW1)
>> se = ones(60);
```

```
>> se(60,60) = 0;
>> se(60,1) = 0;
>> se(1,60) = 0;
>> se(1,1) = 0;
>> BW1 = imerode(BW,se);
>> figure,imshow(BW1)
```

Дилатация

Операцией, двойственной к эрозии, является дилатация (dilatation).

Дилатация выполняется функцией imdilate пакета IPT.

Синтаксис

D = imdilate (S, se),

где S – это двоичное или полутоновое изображение, se – матрица из нулей и единиц, которая определяет структурообразующий элемент.

2. Выполнить дилатацию изображения из файла TextRoman.bmp.

```
>> I = imread('c:\Image\TextRoman.bmp');
>> bw = I>150;
>> figure,imshow(bw)
>> se = [0 1 0;1 1 1;0 1 0];
>> bw1 = imdilate(bw,se);
>> figure,imshow(bw1)
```

Эрозия и дилатация – операции, предназначенные в первую очередь для выявления различных морфологических особенностей изображения с использованием различных структурных элементов. Так, эрозия посредством круга с радиусом r позволяет найти в изображении объекты, минимальный поперечный размер которых превышает 2r. Если в качестве структурного элемента взять две точки, смещение между которыми определяется вектором h, эрозия позволит выделить объекты, имеющие соседей в направлении и на расстоянии, заданных этим вектором.

3. Найти объекты на изображении файла cgc.bmp, поперечный размер которых превышает 30 пикселов с помощью эрозии посредством круга с радиусом r = 15.

```
>> f1 = imread('c:\Image\cgc.bmp');
>> I=im2double(rgb2gray(f1));
>> T=graythresh(I);
>> bw=I<T;
>> figure,imshow(bw)
>> title('original')
>> r=15;
>> se = strel('disk', r)
>> bw1 = imerode(bw,se);
>> figure,imshow(bw1)
>> title('rezult')
```

Операции замыкание и размыкание являются комбинированием дилатации и эрозии.

Замыкание и размыкание

Морфологическое размыкание X по B обозначается X ◦ B и определяется как эрозия X по B, после которой выполняется дилатация результата по B

Морфологическое замыкание множества X по B обозначается X • B. Эта операция представляет собой эрозию, примененную к результату дилатации

Преобразование размыкания и замыкания реализовано функциями imopen и imclose соответственно.

Синтаксис

D = imopen(S, se) D = imclose(S, se),

где S – это двоичное или полутоновое изображение; se – матрица из нулей и единиц, которая определяет структурообразующий элемент.

4. Выделить объекты на изображении файла Ex4.bmp, используя операции замыкания и размыкания.

```
>> [x,map]= imread('C:\Image\Ex4.bmp');
>> figure,imshow(x,map)
>> I=im2double(ind2gray(x,map));
>> T=graythresh(I);
>> bw=im2bw(I,T);
>> figure,imshow(bw), title('bw')
```

```
>> R=18, se = strel('disk', R);
>> bwo=imopen(bw,se);
>> figure,imshow(bwo)
>> r=int2str(R);
>> s=cat(2,'bwo-',r)
>> title(s)
>> R=5;
>> se = strel('disk', R);
>> bwcl=imclose(bwo,se);
>> figure,imshow(bwcl)
>> r=int2str(R);
>> s=cat(2,'bwcl-',r)
>> title(s)
```

Морфологическая реконструкция

Для поиска нужных объектов изображения можно использовать морфологическую реконструкцию. Одно изображение, которое называется маркером, является исходной точкой преобразования. Другое изображение, маска, накладывает определенные ограничения (связи) на отображение.

Если g – это маска (mask), а f – маркер (marker), то реконструкция g по f определяется следующей итеративной процедурой.

1. Присвоить h_i маркерное изображение f.
2. Построить структурообразующий элемент B = ones(3).
3. Повторять: $h_{i+k} = (h_k \oplus B) \cap g$ до тех пор, пока не станет $h_{i+k} = h_k$.

Маркер f должен быть подмножеством g, т. е. $f \subseteq g$.

Функция imreconstruct из пакета IPT выполняет реконструкцию.

Синтаксис

out = imreconstruct(marker, mask)**,**

5**.** Найти в тексте из файла TextRoman.bmp буквы, у которых имеются длинные вертикальные черточки.

```
>> I = imread('c:\Image\TextRoman.bmp');
>> I=im2double(I);
>> T=graythresh(I);
>> bw=I>T;
>> figure,imshow(bw)
>> title('original')
```

```
>> bwr = imopen(bw, ones(3, 1));
>> figure,imshow(bwr)
>> title('razmikanie')
>> bwR = imreconstruct(bwr, bw);
>> figure,imshow(bwR)
>> title('Rezult')
```

Функция bwmorph

Морфологические операции над бинарным изображением также можно выполнить, используя функцию bwmorph.

Синтаксис

BW_D = bwmorph(BW_S, operation, n)

Функцией bwmorph создается бинарное изображение BW_D при обработке морфологическим фильтром исходного бинарного изображения BW_S n раз (по умолчанию n = 1). Описание некоторых морфологических операций приведено в таблице 3.

Алгоритм.

В структурообразующем элементе, который также называют маской морфологического фильтра, ненулевые значения определяют, какие из соседних пикселов следует учитывать при осуществлении операции. При эрозии бинарного изображения пиксел исходного изображения сбрасываются в ноль, если хотя бы один из пикселов окрестности, соответствующий ненулевому элементу маски, равен 0. При наращивании бинарного изображения пиксел исходного изображения устанавливается в 1, если хотя бы один из пикселов окрестности, соответствующий ненулевому элементу маски, равен 1.

В функции bwmorph в операциях эрозии и наращения используется структурообразующий элемент 3 на 3 вида:

$$\begin{vmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix}$$

Данные правила применяются ко всем пикселам изображения.де marker – это изображения-маркер; mask – это изображения-маска.

Таблица 3 - Морфологические операции функции bwmorph

Тип операци	Описание
‘erode’	Эрозия объекта. Приводит к замене значений граничных пикселов объекта на ноль
‘dilate’	Наращение объекта. Приводит к замене пикселов фона, граничащих с объектом на единицу
‘open’	Открытие. Представляет собой последовательное применение эрозии и наращения. Приводит к соединению областей фона, ранее разъединенных узкими участками пикселов объекта
‘close’	Закрытие. Представляет собой последовательное применение наращения и эрозии. Приводит к удалению
‘tophat’	«Верх шляпы». Соответствует вычитанию из исходного изображения результата его открытия
‘bothat’	«Низ шляпы». Соответствует вычитанию исходного изображения из результатов его закрытия
‘skel’	Построение остова объекта. Результат операции – связная линия толщиной в один пиксел, проходящая посредине
‘thin’	Утончает объекты без дыр до минимальной средней линии. Утончает объекты с дырами до колец
‘shrink’	Сжимает объекты без внутренних дыр в точки. Сжимает объекты с дырами в кольца
‘thicken’	Утолщает объекты без соединения несвязных частей

6. Разделить слишком слипшиеся объекты с помощью морфологических операций. Для этого изображение подвергается эрозии до тех пор, пока не исчезнет слипание объектов. В данном случае необходимо 10 итераций эрозии. Результат эрозии помещается в изображение BW2. Затем для BW2 «утолщаются» объекты (строится остов фона). Результат логического И изображений BW1 & BW2 дает изображение с пикселами фона в местах слипания объектов.

```
>> R = imread('c:\Image\cgc.bmp');
```

```
>> I = im2double(rgb2gray(R));
>> T=graythresh(I); >> BW1 = I<T;
>> figure,imshow(BW1)
>> BW2 = bwmorph(BW1,'erode',10);
>> figure,imshow(BW2)
>> title('erode')
>> BW3 = bwmorph(BW2,'thicken',inf);
>> figure,imshow(BW3)
>> title('thicken')
>> BW4 = BW1&BW3;
>> figure,imshow(BW4)
>> title('Rezult')
```

Содержание 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 (компьютерная распечатка работ);
5. Выводы;

Контрольные вопросы

1. Какие логические операции над бинарными изображениями вы знаете?
2. В чем назначение структурообразующего элемента в морфологических операциях?
3. Для чего используются морфологические операции?
4. Какие морфологические операции обработки изображения относятся к базовым?
5. Какие операции являются комбинированием эрозии и дилатации?

6. Какие функции пакета IPT выполняют операции эрозии и дилатации, замыкания, размыкани

Практическая работа №5 «Контурный анализ на основе дескрипторов Фурье в среде MATLAB»

Цель работы – проанализировать экспериментальным путем приведенные методы распознавания образов на макете, реализованном в среде Matlab с использованием Image Processing Toolbox.

Порядок выполнения работы

1. Запуск макета осуществляется открытием файла lab.m в среде Matlab с последующим запуском скрипта (рисунок 1)

Рисунок 1 - Запуск макета

2. Макет содержит базу эталонов (рисунок 2) для распознавания (каталог \StandardImages , *.png файлы). Изображения для распознавания находятся в каталоге \BaseImages, в виде *.png файлов.

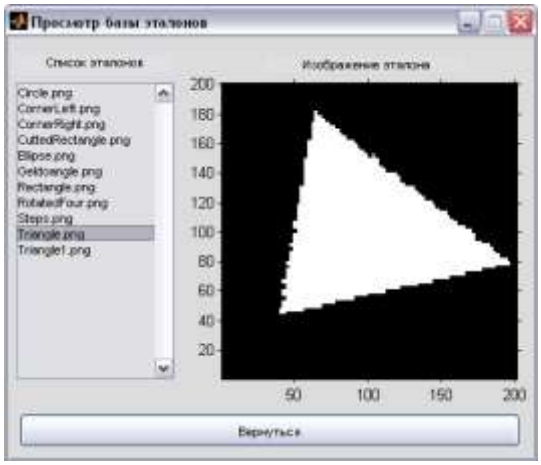

Рисунок 2 - Просмотр эталонов

3. Процедура распознавания состоит из двух шагов:

1. Выбор изображения для распознавания (рисунок 3).

2. Выбор метода распознавания (рисунок 4).

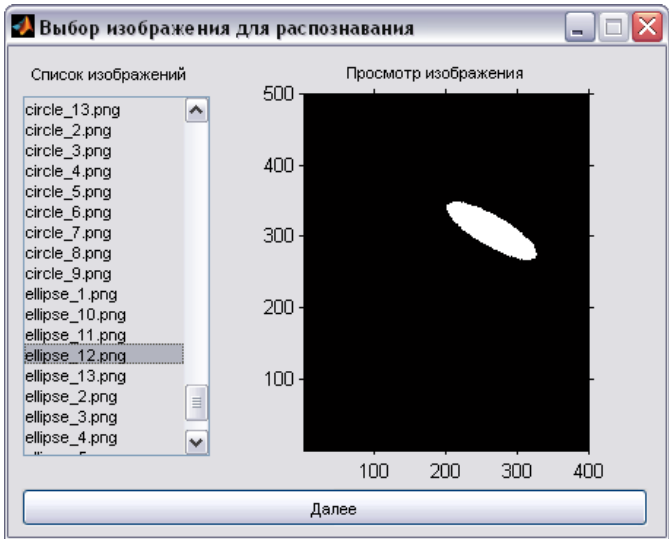

Рисунок 3- Первый шаг – выбор распознаваемого изображения

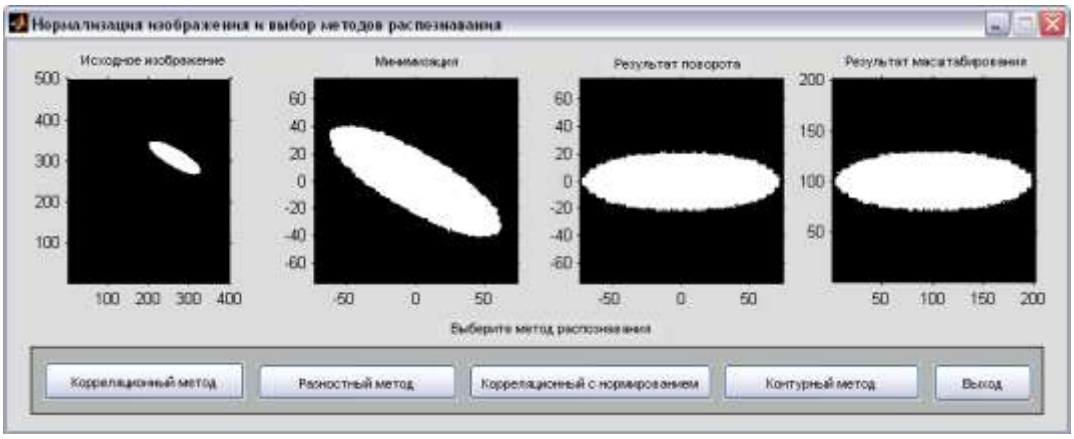

Рисунок 4 - Второй шаг – выбор метода распознавания

4. В каждом конкретном методе распознавания реализован выбор эталона для анализа значения корреляционного признака «вручную» и кнопка запуска выбранного метода распознавания, при нажатии на которую происходит вычисление коэффициента корреляции, его минимизация/максимизация и выбор наиболее подходящего эталона.

5. В корреляциионных методах имеется возможность зашумления распознаваемого изображения. В контурном методе

имеется возможность изменения параметра оператора (Собел, Превитт). Ниже представлены скриншоты работы с макетом:

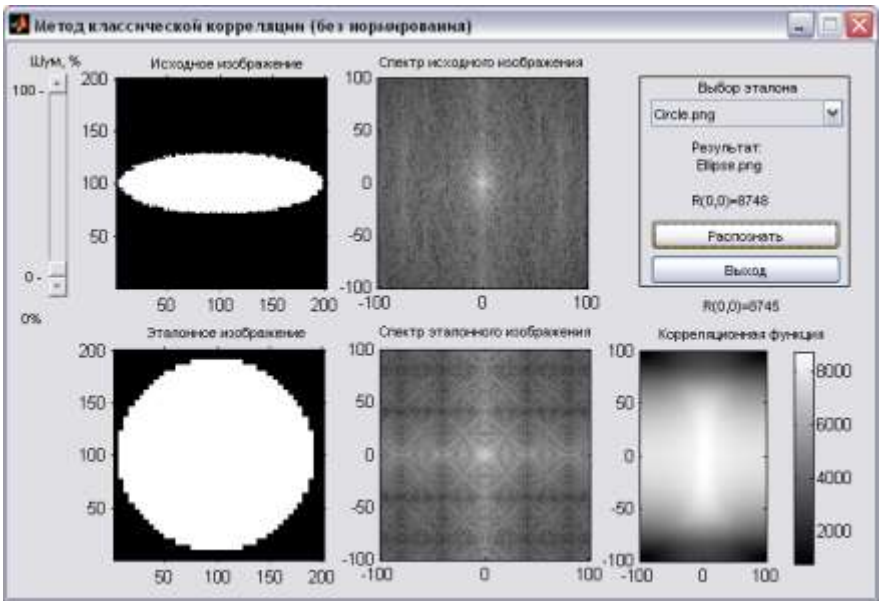

Рисунок 5 - Корреляционный метод(без нормализации)

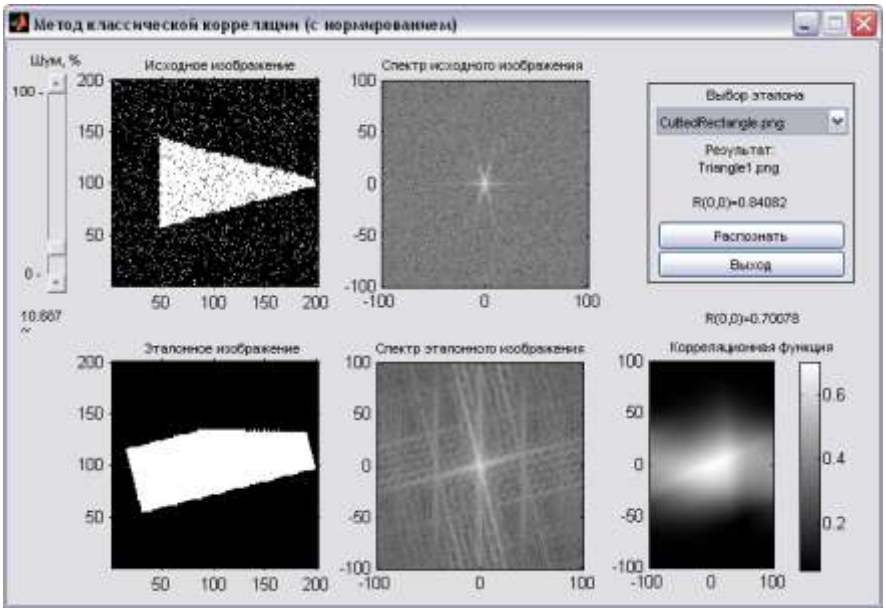

Рисунок 6 - Корреляционный метод (с нормализацией)

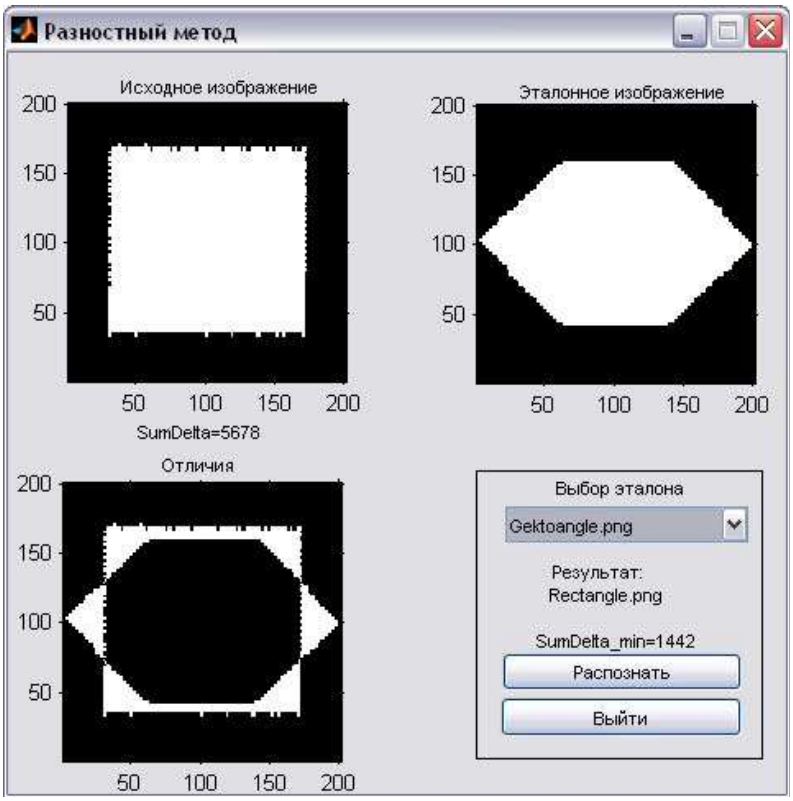

Рисунок 7 - Разностный метод

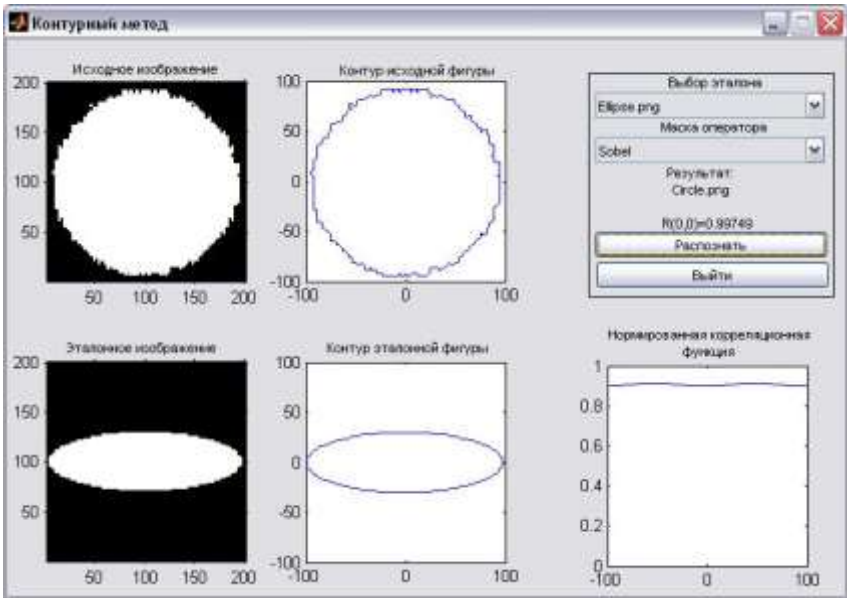

Рисунок 8 - Контурный метод

Содержание отчета

1. Цель работы.
2. Постановка задачи согласно выбранному варианту.
3. Описание самостоятельно проделанной работы (включает анализ поставленной задачи и особенностей применения выбранного метода распознавания, описание проводимых экспериментов и тестов, анализ полученных результатов).
4. Выводы, рекомендации.
5. Приложение: скриншоты.

Контрольные вопросы

1. Правильная последовательность этапов процесса распознавания образов: а) восприятие образа, б) выделение характеристик образа, в) классификация, г) предварительная обработка.
2. Правильная последовательность действий при построении системы распознавания образов: а) выбрать и применить метод обучения распознаванию, б) выбрать значимые признаки, характеризующие образ, в) выбрать модель представления образов, г) оптимизировать (при необходимости) алгоритм распознавания, д) подготовить обучающую выборку, е) проверить качество работы системы распознавания, ж) сформулировать правило классификации.
3. Для успешного достижения цели распознавания необходимо решить следующие проблемы: а) высокая валидность результатов, б) комбинаторный взрыв размерности решаемой задачи, в) корректное снижение размерности пространства признаков без существенной потери значимой информации, г) независимость времени распознавания от размера обучающей выборки, д) отказ от гипотезы компактности.
4. Какие признаки и параметры элементов изображений используются при распознавании: а) дескриптор границы, б) дескриптор яркости, в) интенсивность цвета, г) центр масс, д) эксцентриситет.

5. Вычисление корреляции между признаками возможно осуществить : а) ортоструктурным методом, б) контурным методом, в) разностным методом, г) методом среднеквадратичного отклонения.

6. К операторам выделения границ контурным методом относятся: а) оператор Червоненкинса, б) оператор Собела, в) оператор Робертса, г) оператор Фурье, д) оператор Превитта.