

Федеральное агентство по образованию
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Факультет математики и информатики

Н.П. Семичевская, Л.А. Соловцова

ОБРАБОТКА ИЗОБРАЖЕНИЙ В СРЕДЕ MATLAB
С ИСПОЛЬЗОВАНИЕМ IMAGE PROCESSING TOOLBOX

ЛАБОРАТОРНЫЙ ПРАКТИКУМ

Благовещенск 2005

СОДЕРЖАНИЕ

Введение	3
Лабораторная работа №1 Представление изображений в среде MATLAB. Типы изображений.	5
Лабораторная работа №2 Битовая карта изображения. Функции PIXVAL и MEAN2.	9
Лабораторная работа №3 Работа с контрастностью изображения (IMADJUST, HISTEQ)	13
Лабораторная работа №4 Построение гистограммы распределения яркостей элементов изображения.	17
Лабораторная работа №5 Дискретная линейная двумерная обработка. Двумерное дискретное преобразование Фурье.	21
Лабораторная работа №6 Двумерные унитарные преобразования. Преобразование Адамара (Уолша). Преобразование Хаара.	26
ПРИЛОЖЕНИЕ 1 Исходные образцы изображений для обработки	30
ПРИЛОЖЕНИЕ 2 Титульный лист отчета по лабораторной работе	31
СПИСОК ЛИТЕРАТУРЫ	32

Методическое пособие предназначено для студентов специальности 230201 по дисциплине «Цифровая обработка изображений» для выполнения лабораторного практикума. Пособие содержит 6 лабораторных работ с кратким изложением теоретического материала, примерами программ, пояснениями и иллюстрациями, в конце каждой лабораторной работы приведены задания, которые необходимо выполнить.

Требования к отчету по оформлению лабораторных работ

1. Для выполнения лабораторных работ студенту выдается тестовое изображение из 12-ти предложенных (смотреть приложение 1).
2. Титульный лист к отчету по лабораторной работе оформляется в соответствии со стандартом (смотреть приложение 2)
3. В отчете следует отразить этапы выполнения лабораторной работы, указать основные методы, алгоритмы или функции, которые использовались при выполнении работы. Листинг программы сохранять в **.m** файле.

Результаты полученные в процессе работы представлять как исходное изображение и результативное изображение (изображение обработанное).

ВВЕДЕНИЕ

Интегрированные среды для моделирования и исполнения программ цифровой обработки изображений и сигналов содержат мощные средства для инженерно–научных расчетов и визуализации данных. Большинство современных пакетов поддерживает визуальное программирование на основе блок–схем. Это позволяет создавать программы специалистам, не владеющим техникой программирования. К таким пакетам относится Image Processing Toolbox системы MATLAB, разработанный фирмой MathWorks. Этот пакет владеет мощными средствами для обработки изображений. Они имеют открытую архитектуру и позволяют организовывать взаимодействие с аппаратурой цифровой обработки сигналов, а также подключать стандартные драйвера.

Система MATLAB и пакет прикладных программ Image Processing Toolbox (IPT) является хорошим инструментом разработки, исследования и моделирования методов и алгоритмов обработки изображений. При решении задач обработки изображений пакет IPT

позволяет идти двумя путями. Первый из них состоит в самостоятельной программной реализации методов и алгоритмов. Другой путь позволяет моделировать решение задачи с помощью готовых функций, которые реализуют наиболее известные методы и алгоритмы обработки изображений. И тот, и другой способ оправдан. Но все же для исследователей и разработчиков методов и алгоритмов обработки изображений предпочтительным является второй путь.

Это объясняется гибкостью таких программ, возможностью изменения всех параметров, что очень актуально при исследовании, разработке, определении параметром регуляризации и т.д. Прежде чем использовать для решения каких-либо задач обработки изображений стандартные функции пакета IPT, разработчик должен в совершенстве их исследовать. Для этого он должен точно знать, какой метод и с какими параметрами реализует та или иная функция.

В том и другом подходе к решению задачи обработки видеоданных объектом исследования является изображение. Для этого рассмотрим коротко особенности представления изображений в IPT.

Лабораторная работа №1

(2 часа)

Тема. Представление изображений в среде MATLAB. Типы изображений

Цель. Научиться считывать изображения из файлов, просматривать их в среде MATLAB преобразовывать их типы, менять размеры смотрового окна.

Краткие теоретические сведения

Типы изображений

Изображения бывают *векторными* и *растровыми*. Векторным называется изображение, описанное в виде набора графических примитивов. Растровые же изображения представляют собой двумерный массив, элементы которого (пикселы) содержат информацию о цвете. В цифровой обработке используются растровые изображения. Они в свою очередь делятся на типы - *бинарные*, *полутоновые*, *палитровые*, *полноцветные*.

Элементы *бинарного* изображения могут принимать только два значения - 0 или 1. Природа происхождения таких изображений может быть самой разнообразной. Но в большинстве случаев, они получаются в результате обработки полутоновых, палитровых или полноцветных изображений методами бинаризации с фиксированным или адаптивным порогом. Бинарные изображения имеют то преимущество, что они очень удобны при передаче данных.

Полутоновое изображение состоит из элементов, которые могут принимать одно из значений интенсивности какого-либо одного цвета. Это один из наиболее распространенных типов изображений, который применяется при различного рода исследованиях. В большинстве случаев используется глубина цвета 8 бит на элемент изображения.

В *палитровых* изображениях значение пикселов является ссылкой на ячейку карты цветов (палитру). Палитра представляет собой двумерный массив, в столбцах которого расположены интенсивности цветовых составляющих одного цвета.

В отличие от палитровых, элементы *полноцветных* изображений непосредственно хранят информацию о яркостях цветовых составляющих.

Выбор типа изображения зависит от решаемой задачи, от того, насколько полно и без потерь нужная информация может быть представлена с заданной глубиной цвета. Также следует учесть, что использование полноцветных изображений требует больших вычислительных затрат.

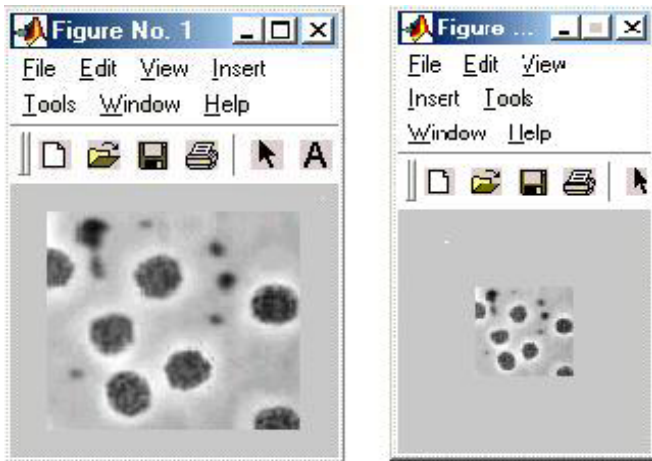
Пример

```
I=imread('micro9.jpg'); %чтение рисунка в матрицу I
imshow(I); %вывод рисунка в графическом окне, описанного матрицей
I
pause; %пауза
W=gcf; %запоминание графического окна в качестве объекта W
trueSize(W,[100,50]); %изменение размера окна до размера 100x50
pause;
trueSize; %возвращение размера окна к реальному размеру W
h=gcf; %запоминание графического окна в качестве объекта h
pause; %пауза
close(h); %закрытие окна
A=imread('isberg.jpg'); %чтение цветного рисунка в матрицу A
fprintf('\n вывод A\n'); %вывод текста в командную строку
imshow(A); %вывод рисунка, описанного матрицей A
pause; %пауза
[m,n] = size(A);
figure('Units','pixels','Position',[100 100 n m])%вывод рисунка,
описанного матрицей A
image(A);
set(gca,'Position',[0 0 1 1])
fprintf('\n вывод B\n');
B=im2double(A); %преобразует входной рисунок в рисунок со знач-ми
%класса double
imshow(B); %вывод рисунка
pause; %пауза
fprintf('\n вывод C\n');
figure('Units','pixels','Position',[100 100 n m])
C=im2uint8(A); %преобразует входной рисунок в рисунок со знач-ми
класса uint8
imshow(C);
I=imread('isberg.jpg'); %чтение рисунка в матрицу I
imshow(I);
```

```

pause;
[X,map]=rgb2ind(I, 8); %преобразует цветной рисунок в
индексированное изобра-ие
Y=ind2gray(X,map); %преобразует входной рисунок в рисунок в 8-ми
градациях серого
Imshow(Y);
pause;
Z = im2bw(I, 0.4); %преобразует входной рисунок в двоичное
изображение
figure('Name','im2bw(I,0.4)');
imshow(Z);
pause;

```



а)

б)

Рис. 1 Вид изображений в окне MATLAB: а) вывод считанного изображения из файла micro9.jpg; б) изменение размера окна до размера 100x50 (TRUE_SIZE).

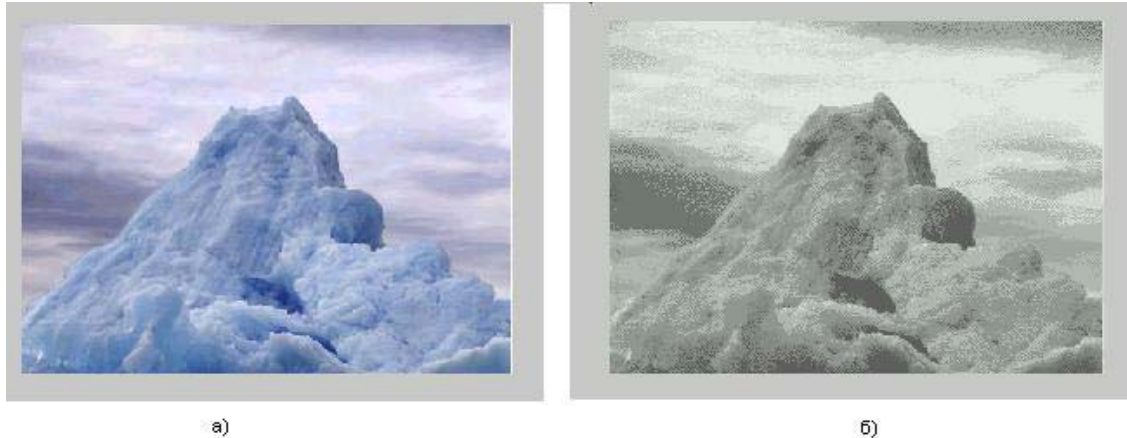


Рис.2 Преобразование исходного цветного изображения в рисунок с 8 уровнями серого.



Рис.3 Двоичное изображение.

Задание к лабораторной работе №1

1. Используя функции `IMAGE`, `IMAGESC`, `IMREAD`, `IMWRITE`, `IMSHOW`, `TRUESIZE` считать и воспроизвести изображение в рабочем окне `MATLAB`.
2. Используя функции перевести исходное изображение в классы `double`, `uint8`.
3. Преобразовать исходное изображение в двоичное.
4. Используя функцию `IMFINFO` получить информацию о файле с исходным изображением.

Лабораторная работа №2

(2 часа)

Тема. Битовая карта изображения. Функции PIXVAL и MEAN2.

Цель. По исходному изображению получить его битовую карту. Построить окно 50*50 пикселей на изображении и получить информацию о распределении яркости в построенном окне. Сделать окно скользящим.

Краткие теоретические сведения

Цифровые изображения (поля) принято представлять в виде матриц (двумерных прямоугольных массивов чисел) с неотрицательными элементами (для полутоновых, палитровых и полноцветных изображений значения элементов заключены в диапазоне [0 255]). Каждый элемент матрицы отвечает одному элементу изображения - пикселю. Матрицу значений яркостей изображения называют битовой картой изображения.

В зависимости от типа изображения они по-разному представляются в разных форматах. Этот момент будет очень важным при создании программ в среде IPT. Наиболее удобно зависимость способов представления элементов изображения (диапазон их значений) от типа и формата представить в виде таблицы.

Тип изображения	double	uint8
Бинарное	0 и 1	0 и 1
Полутоновое	[0, 1]	[0, 255]
Палитровое	[1, размер палитры], где 1 - первая строка палитры	[0, 255], где 0 - первая строка палитры.*
Полноцветное	[0, 1]	[0, 255]

* **Примечание.** При программной реализации лучше избегать использования такой индексации строк. MATLAB корректно воспринимает индексацию с первой, а не нулевой строки.

В дальнейшем, при рассмотрении методов обработки изображений, будем считать, что изображение представляется матрицей чисел $M \times N$ (размер матрицы), где значение каждого элемента отвечает определенному уровню квантования его энергетической

характеристики (яркости) – это и есть битовая карта изображения. Это так называемая *пиксельная система координат*. Она применяется в большинстве функций пакета IPT.

Существует также *пространственная система координат*, где изображение представляется непрерывным числовым полем квадратов с единичной величиной. Количество квадратов совпадает с числом пикселей. Значение интенсивности элемента в центре квадрата совпадает со значением соответствующего пиксела в пиксельной системе координат. При решении практических задач, связанных с измерениями реальных геометрических размеров объектов на изображении, удобно использовать пространственную систему координат, так как она позволяет учитывать разрешение (количество пикселей на метр) системы.

Пример: Неподвижное окно на исходном изображении

```
imfinfo('micro9.jpg')
[I,x]=imread('micro9.jpg'); %чтение рисунка в матрицу I
for j=1:50,
    for k=1:50,
        for l=1:3
            I2(j,k,l)=I(j,k,l);
        end
    end
end
imshow(I2);%вывод окна 50*50 в графическом окне
pause;
[m,n,t] = size(I);
figure('Units','pixels','Position',[100 100 n m]);
%регулирование значений интенсивности изображения в окне
I3=imadjust(I2,[0.4 0.6],[,]);
imshow(I3);%вывод окна 50*50 с измененной интенсивностью в
графическом окне
pause;
[m,n,t] = size(I);
figure('Units','pixels','Position',[100 100 n m]);
for j=1:50,
    for k=1:50,
        for l=1:3
            I(j,k,l)=I3(j,k,l);
```

```

    end
  end
end
imshow(I);
pause;
%Вывод битовой карты окна t+10*t+10
t=15;
for j=t:t+10,
  for k=t:t+10,
    for l=1:3
      I4(j,k,l)=I(j,k,l);
    end
  end
end
end
I4(:,:,1)
imshow(I4);

```

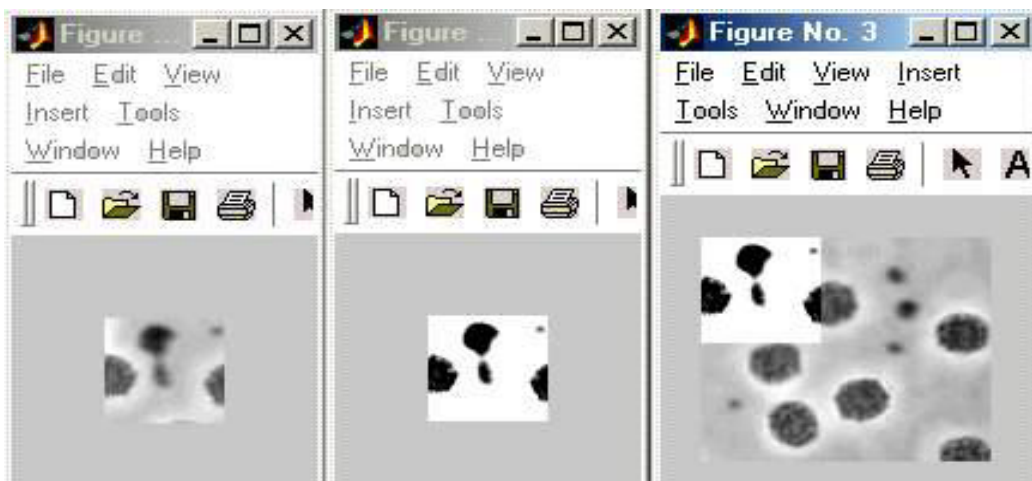


Рис.1 а) окно 50*50; б) окно 50*50 с измененной интенсивностью; в) исходное изображение с наложением окна 50*50 с измененной интенсивностью.

Задание к лабораторной работе №2

5. Используя функцию IMFINFO получить информацию о файле с исходным изображением.
6. Получить битовую карту исходного изображения

7. На исходном изображении получить окно 50*50 пикселей. Для того чтобы окно было видимым на изображении изменить его интенсивность с помощью функции IMADJUST.
8. Получить битовую карту окна 50*50.
9. Используя функцию PIXVAL получить информацию о значении яркости в конкретном пикселе исходного изображения.
10. Используя функцию MEAN2 вычислить среднее значение элементов матрицы исходного изображения и окна 50*50.
11. Написать программу передвижения окна 50*50 по исходному изображению.

Лабораторная работа №3

(2 часа)

Тема. Работа с контрастностью изображения (IMADJUST, HISTEQ)

Цель. Изменение контраста (увеличить/уменьшить) всего изображения и в окне заданного размера.

Краткие теоретические сведения

Технология повышения контрастности изображений

Пакет Image Processing Toolbox содержит несколько классических функций улучшения изображений. Эти функции являются очень эффективными при повышении контраста: IMADJUST, HISTEQ. Приведем их сравнительный анализ при использовании для улучшения полутоновых изображений.

Считывание изображений.

Считаем полутоновое изображение: micro9.jpg . Также считаем индексное RGB-изображение: isberg.jpg.

```
M=imread('micro9.jpg');
figure('Name','Исходник');
imshow(M);
I = imread('isberg.jpg');
figure('Name','Исходник');
imshow(I);
[X map]=rgb2ind(I,16);
J = ind2rgb(X,map);
figure('Name','Индексированное RGB');
imshow(J);
```

Улучшение полутоновых изображений.

Проведем сравнительный анализ эффективности использования следующих функций к улучшению изображений:

IMADJUST - увеличение контраста изображений путем изменения диапазона интенсивностей исходного изображения.

HISTEQ - выполнение операции эквализации (выравнивания) гистограммы. В этом подходе увеличение контрастности изображения происходит путем преобразования гистограммы распределения значений интенсивностей элементов исходного изображения. Существуют также другие подходы к видоизменению гистограмм.

ADAPTHISTEQ - выполнение контрастно-ограниченного адаптивного выравнивания гистограммы. Здесь методика повышения контрастности изображений базируется на анализе и эквализации гистограмм локальных окрестностей изображения.

Проведем краткий анализ эффективности обработки изображений с помощью функций **IMADJUST**, **HISTEQ**. В основе этих функций лежат разные методы: в **IMADJUST** - преобразование диапазона яркостей элементов изображения, в **HISTEQ** - эквализация гистограммы. Каждый из этих методов (функций) нацелен на устранение некоторого недостатка, поэтому может быть применен для эффективной обработки такого класса изображений, на которых есть такой тип искажений.

Пример.

```
M=imread('micro9.jpg');
figure('Name','Исходник');
imshow(M);
% HISTEQ
J = histeq(M(:,:,1));
figure('Name','255');
imshow(J);
J = histeq(M(:,:,1),10);
figure('Name','10');
imshow(J);
J = histeq(M(:,:,1),4);
figure('Name','4');
imshow(J);
%IMADJUST
J = imadjust(M,[0. 1],[0. 1]);
figure('Name','imadjust(I,[0. 1])');
imshow(J);
J = imadjust(M,[0.1 0.5],[0. 1]);
figure('Name','imadjust(I,[0.1 0.5])');
imshow(J);
J = imadjust(M,[0.5 0.7],[0. 1]);
figure('Name','imadjust(I,[0.5 0.7])');
imshow(J);
%Работа с индексированным изображением
I = imread('isberg.jpg');
figure('Name','Исходник');
```

```

imshow(I);
[X map]=rgb2ind(I,16);
J = ind2rgb(X,map);
figure('Name','ind2rgb(X,map)');
imshow(J);
% HISTEQ
JJ = histeq(J(:, :, 1), 28);
%figure('Name','28');
%imshow(JJ);
% IMADJUST
JI = imadjust(J,[0.3 0.7],[0.3 0.7]);
%figure('Name','imadjust(I,[0.3 0.7])');
%imshow(JI);

```

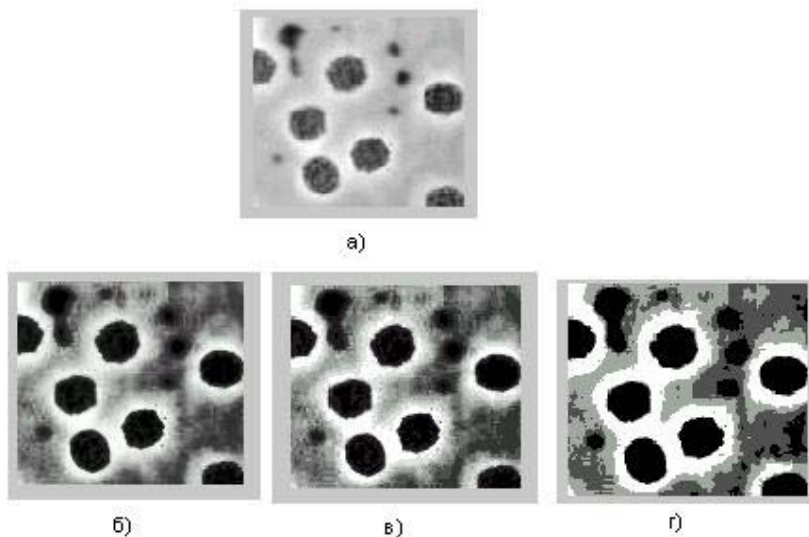


Рис.1 Регулирование контрастности исходного изображения а) с использованием гистограммных преобразований HISTEQ: б) 256 уровней яркости; в) 10 уровней яркости; г) 4 уровня яркости.

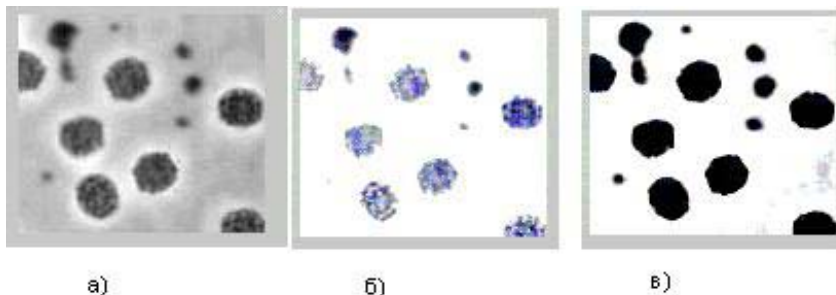


Рис.2 Регулирование контрастности исходного изображения а) путем изменения диапазона интенсивностей исходного изображения: б) в диапазоне [0.1 0.5]; в) в диапазоне [0.5 0.7].

Задание к лабораторной работе №3

12. Используя функции `IMADJUST`, `HISTEQ` изменить контрастность тестового изображения при различных параметрах функций. Исходное и полученные изображения выводить в рабочих окнах `MATLAB`.
13. Используя функции `IMADJUST`, `HISTEQ` увеличить контрастность в движущемся окне 50×50 (наложить окно на исходное изображение).
14. Получить двухградационное изображение, используя функцию `HISTEQ`.

Лабораторная работа №4

(2 часа)

Тема. Построение гистограммы распределения яркостей элементов изображения.

Цель. Построение общей гистограммы исходного изображения и локальной гистограммы для движущегося по изображению окна. Определение границ диапазона яркостей.

Краткие теоретические сведения

Гистограмма этот термин впервые был использован Пирсоном в 1895 г. Гистограмма является графическим представлением распределения частот выбранных переменных, на котором для каждого интервала (класса) рисуется столбец, высота которого пропорциональна частоте класса.

Для построения общей гистограммы полутонового изображения используется, как правило, 256 уровней яркости (интенсивности) изображения (ширина интервалов гистограммы равна 1), а высота каждого столбца – это количество пикселей изображения соответствующей яркости (интенсивности).

Можно рассчитывать максимально допустимый уровень яркости и минимальный уровень яркости для заданного изображения, и тогда диапазон уровней яркости может быть взят меньшим, чем 256 уровней.

В качестве примера рассмотрим гистограммы изображений `micro9.jpg` и `micro2.jpg`. Гистограмма изображения `micro2.jpg` занимает только центральную часть возможного диапазона яркостей Рис.2.

```
I=imread('micro9.jpg');
figure('Name','micro9.jpg');
    imshow(I);
    figure;
    imhist(I(:, :, 1));
J=imread('micro2.jpg');
figure('Name','micro2.jpg');
    imshow(J);
    figure;
imhist(J(:, :, 1));
```

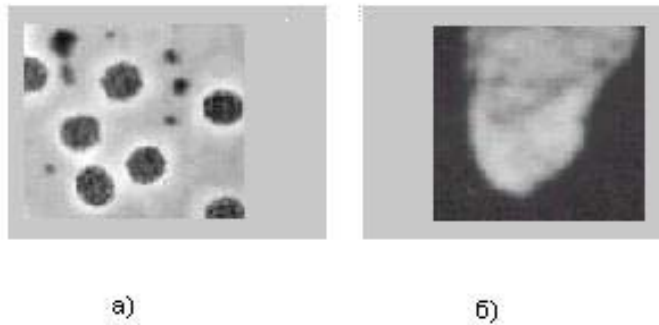


Рис.1 Исходные изображения а) micro9.jpg; б) micro2.jpg.

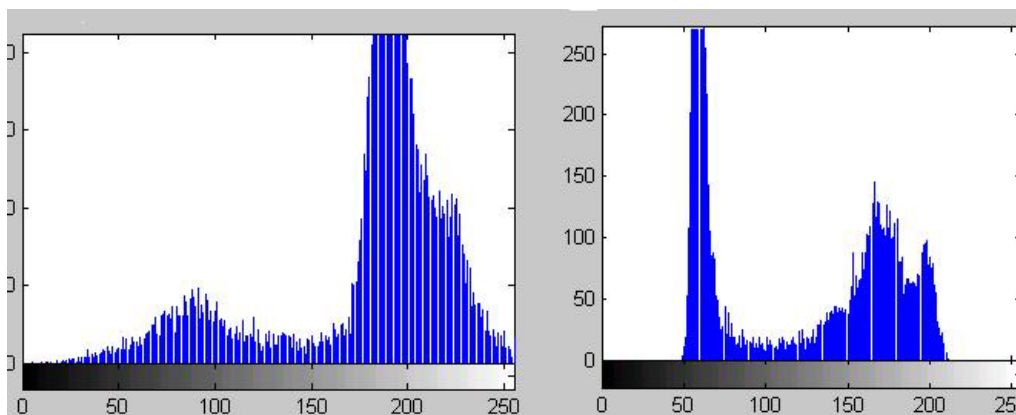


Рис.2 Гистограммы изображений micro9.jpg и micro2.jpg.

Один из критериев выбора того или иного метода обработки может базироваться на анализе гистограммы распределения значений яркостей элементов изображения.

С целью повышения контраста изображений, на которых мелкие детали на темных участках видимы плохо, и сами изображения характеризуются низким контрастом, используют методы видоизменения гистограммы. Суть этих методов состоит в преобразовании яркостей исходного изображения таким образом, чтобы гистограмма распределения яркостей приобрела желательную форму. Оптимальным с точки зрения зрительного восприятия человеком является изображение, элементы которого имеют равномерное распределение яркостей. Получают улучшенные изображения путем выравнивания гистограммы, то есть стремятся достичь равномерности распределения яркостей обработанного изображения. У. Фрэй исследовал метод видоизменения гистограмм, который обеспечивал экспоненциальную или гиперболическую форму распределения яркостей улучшенного изображения. Д. Кетчам

усовершенствовал этот метод, применив скользящую "локальную" гистограмму, полученную для некоторого участка изображения.

Это преобразование эффективно для улучшения визуального качества низко контрастных деталей. Существует также ряд известных методов видоизменения гистограммы, которые приводят к получению изображений с заранее заданным распределением.

Описанные методы преобразования гистограммы могут быть глобальными, то есть использовать информацию обо всем изображении, и скользящими, когда для преобразования используются локальные области изображения. Рассмотренные выше подходы служат основой широкого класса гистограммных методов преобразования изображений.

Основным этапом при создании методов гистограммных преобразований является корректное построение гистограммы распределения яркостей элементов изображения. Для этого в большинстве случаев используется встроенная функция IMHIST.

Пример

```
I=imread('micro9.jpg');
figure('Name','Исходник');
imshow(I);
[m n t]=size(I);
figure;
imhist(I(:, :, 1));
J=imread('micro2.jpg');
figure('Name','Исходник');
imshow(J);
[k l]=size(J);
figure;
imhist(J(:, :, 1));
%Построение общей гистограммы исходного изображения micro9.jpg
for s =1:256
    H(s)= 0;
end
for i=1:m
    for j=1:n
        H(I(i,j,1)) = H(I(i,j,1)) + 1;
    end
end
%grid on
```

```
figure;  
for s =1:256  
    line([s s], [0 H(s)])  
end  
axis([0 255 0 350])  
pause;
```

Для растяжения динамического диапазона можно использовать функцию IMADJUST. Если гистограмма изображения характеризуется неравномерностью распределения, то для выравнивания гистограммы можно применить функцию HISTEQ.

Задание к лабораторной работе №4

1. Построить общую гистограмму исходного изображения, используя функцию IMHIST.
2. Для окна 50*50 построить гистограмму программным методом (написать программный код) и с помощью стандартной функции IMHIST, сравнить полученные гистограммы.
3. Определить диапазон яркостей исходного изображения.

Лабораторная работа №5

(2 часа)

Тема. Дискретная линейная двумерная обработка. Двумерное дискретное преобразование Фурье.

Цель. Используя стандартную функцию FFT2 получить преобразованное изображение. Написать программу для двумерного преобразования Фурье, сравнить действия стандартного фильтра и своего.

Краткие теоретические сведения

Основой большинства методов линейной фильтрации в пространственной области являются ортогональные преобразования. Существует три основных области применения двумерных ортогональных преобразований для обработки изображений. Во-первых, преобразования используют для выделения характеристик признаков изображения. Второй областью применения является кодирование изображений, когда ширина спектра уменьшается за счет отбрасывания или грубого квантования малых по величине коэффициентов преобразования. Третья область применения - это сокращения размерности при выполнении вычислений. К таким преобразованиям принадлежат преобразования Фурье, синусные, косинусные, волновые преобразования, а также преобразования Карунена-Лоева, Уолша, Хаара и Адамара. Однако и эти преобразования не обеспечивают обработки изображений в масштабе реального времени из-за своей вычислительной сложности.

Синтаксис команд для выполнения двумерного преобразования Фурье:

$Y = \text{fft2}(X)$

$Y = \text{fft2}(X, m, n)$

Описание:

Функция $Y = \text{fft2}(X)$ вычисляет двумерное ДПФ, возвращая результат в матрице комплексных чисел Y , имеющей тот же размер, что и матрица X ; X может быть вектором, в этом случае возвращается вектор Y , имеющий такую же ориентацию, что и вектор X .

Функция FFT2(X, m, n) вычисляет двумерное преобразование Фурье матрицы $m \times n$, при необходимости дополняя нулями или усекая исходную матрицу X. Возвращается матрица Y размера $m \times n$. Массив Y имеет формат представления данных double.

Двумерное преобразование Фурье вычисляется выполнением двух одномерных преобразований FFT(FFT(x).!); сначала вычисляется ДПФ каждого столбца, а затем каждой строки результата [6].

Пример1

```
I1=imread('micro9.jpg');
[M,N,t]=size(I1);
I2=double(I1)+1;
for j=1:M
    for k=1:N
        I3(j,k)=I2(j,k,1);
    end
end
J=fft2(I3);
J1=real(J);
J2=imag(J);
J3=abs(J);
a=-2*pi*i/sqrt(M*N);
for j=1:N
    for k=1:N
        Ac(j,k)=exp(a*j*k);
    end
end
for j=1:M
    for k=1:M
        Ar(j,k)=exp(a*j*k);
    end
end
P=Ar*I3*Ac/sqrt(M*N);
P1=real(P);
P2=imag(P);
P3=abs(P);
figure('Name','Исходное изображение');
imshow(I1);
```

```

figure('Name','Преобразованное (1) - вещ часть');
imshow(J1);
figure('Name','Преобразованное (1) - мним часть');
imshow(J2);
% figure('Name','Преобразованное (1) - модуль');
% imshow(J3);
figure('Name','Преобразованное (2) - вещ часть');
imshow(P1);
% figure('Name','Преобразованное (2) - мним часть');
% imshow(P2);
% figure('Name','Преобразованное (2) - модуль');
% imshow(P3);

```

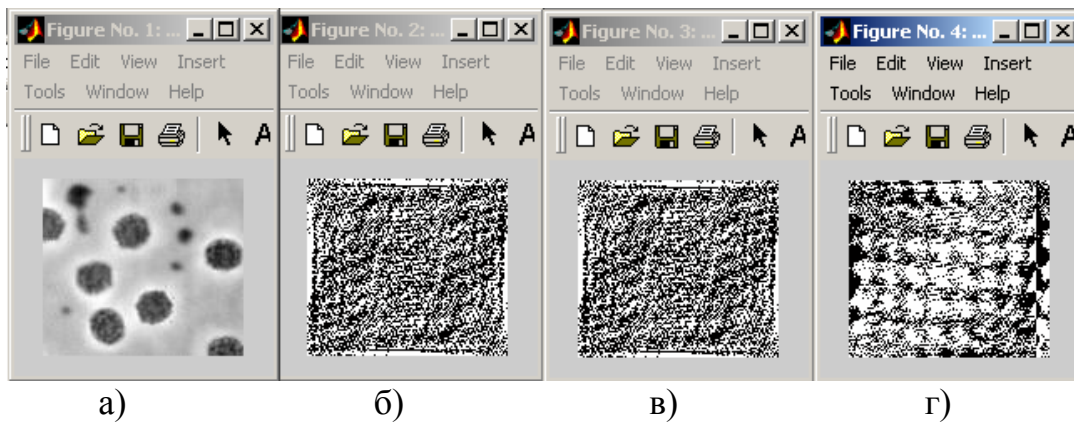


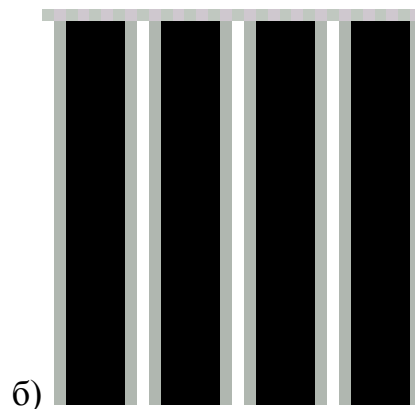
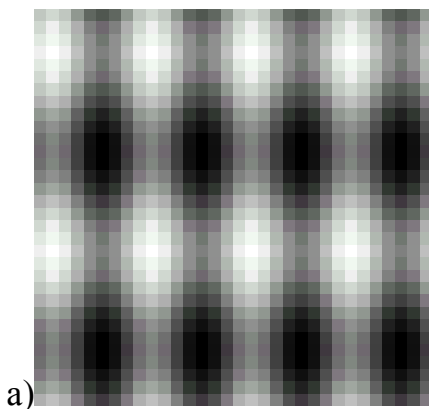
Рис. 1. Преобразование Фурье исходного изображения а): б) преобразование Фурье функцией FFT2; в) действительная часть преобразования Фурье; г) мнимая часть преобразования Фурье.

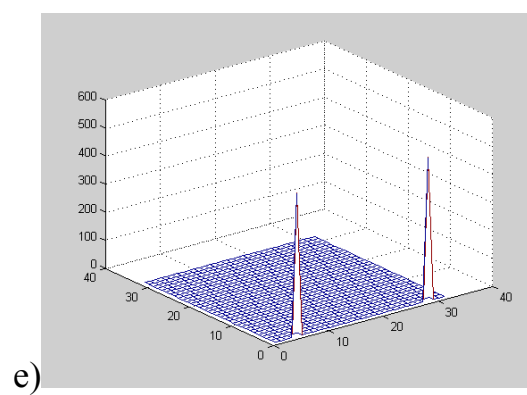
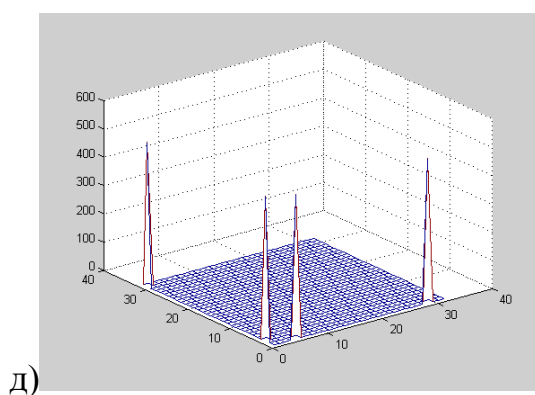
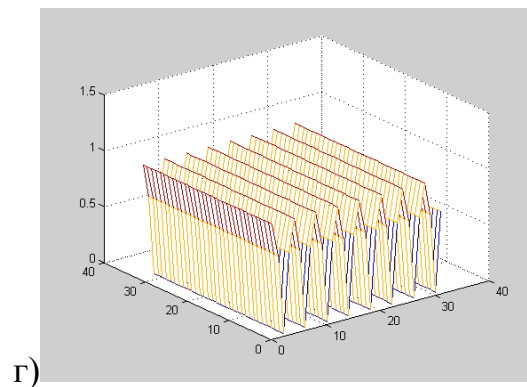
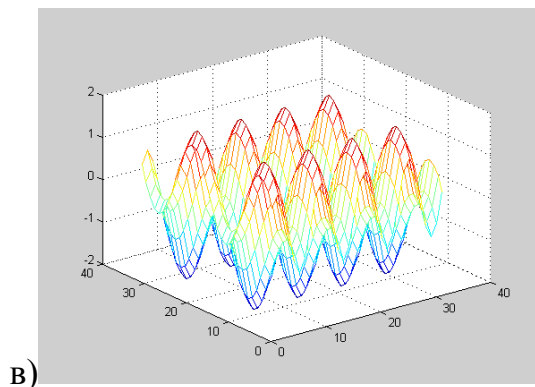
Сформируем двумерный сигнал, являющийся суммой двух пространственных волн. Выведем его на экран в виде полутонового изображения (рис. 2, а) и трехмерной поверхности (рис.2, в). Затем вычислим спектр сигнала и выведем его на экран (рис. 2, д). Так как спектр симметричен, 4 отчетливых пика на рис. 2, д соответствуют двум частотам волн, из которых сформировано исходное изображение.

Удалим из спектра пики, соответствующие одной из частот (рис. 2, е). Восстановим с помощью обратного преобразования Фурье из отредактированного спектра двумерный сигнал. Результат восстановления выводится на экран в виде полутонового изображения (рис. 2, б) и трехмерной поверхности (рис. 2, г).

Пример 2

```
% Пример демонстрирует двумерное
% прямое и обратное преобразования Фурье
% Формирование исходного изображения,
[x, y]=meshgrid(1:32);
I=sin(2*pi*x/8)+sin(2*pi*y/16);
% Вывод исходного изображения на экран.
imshow(mat2gray(I));
% Вывод исходного изображения в виде 3D-поверхности.
figure, mesh(x,y, I);
% Прямое преобразование Фурье.
h=fft2(I);
% Вывод на экран спектра исходного изображения.
figure, mesh(x,y, abs(h));
% Обнуление части спектра.
h(1:32,1:2)=0;
% Вывод на экран получившегося спектра.
figure, mesh(x,y, abs(h));
% Обратное преобразование Фурье.
I=ifft2(abs(h));
% Вывод на экран результирующего изображения.
figure, imshow(mat2gray(I));
% Вывод на экран результирующего изображения в виде 3D-
поверхности.
figure, mesh(x, y, abs(I));
```





Задание к лабораторной работе №5

1. Для исходного изображения получить преобразованное изображение, используя функцию FFT2.
2. Написать программный код для преобразования Фурье и преобразовать исходное изображение, сравнить полученные преобразованные изображения.
3. Получить смещение спектра в центр картинки.
4. Вывести исходное изображение в виде 3D поверхности, удалить пики из спектра. Вывести исходное и полученное изображения.

Лабораторная работа №6

(2 часа)

Тема. Двумерные унитарные преобразования. Преобразование Адамара (Уолша). Преобразование Хаара.

Цель. Получить преобразования Адамара и Хаара для исходного изображения.

Краткие теоретические сведения

Унитарные преобразования являются частным случаем линейных преобразований, когда линейный оператор точно обратим, а его ядро удовлетворяет условиям ортогональности. В результате прямого унитарного преобразования матрицы изображения $F(n_1, n_2)$ размера $N_1 N_2$ образуется матрица преобразованного изображения того же размера, элементы которой равны

$$G(m_1, m_2) = \sum \sum F(n_1, n_2) A(n_1, n_2; m_1, m_2), \quad (1)$$

где $A(n_1, n_2; m_1, m_2)$ – ядро прямого преобразования.

1) Преобразование Адамара основано на квадратной матрице Адамара, элементы которой равны $+1$ или -1 , а строки или столбцы образуют ортогональные векторы. Нормированная матрица Адамара N -го порядка удовлетворяет соотношению

$$H_N H_N^T = I_N \quad (2)$$

Среди ортонормальных матриц Адамара наименьшей является матрица второго порядка

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Пока не удалось определить, существуют ли матрицы Адамара для произвольных N , но почти для всех допустимых N , доходящих до 200, найдены правила построения соответствующих матриц. Наиболее просто удается построить такие матрицы при $N=2^n$, где n -целое. Если H_N матрица Адамара N -го порядка, то матрица

$$H_{2N} = \frac{1}{\sqrt{2}} \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}, \quad (3)$$

Также является матрицей Адамара порядка $2N$.

Матрицы Адамара четвертого и восьмого порядка, полученные по формуле (3) имеют вид

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix},$$

$$H_8 = \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}. \quad (4)$$

Строки матрицы Адамара можно рассматривать как последовательность отсчетов прямоугольных периодических колебаний (сигналов), период которых кратен $1/N$. Подобные непрерывные функции, называемые функциями Уолша, связаны с импульсными функциями Радемахера. Следовательно, матрица Адамара описывает преобразование, связанное с разложением функций по семейству прямоугольных базисных функций, а не по синусам и косинусам, характерным для преобразования Фурье.

Преобразование Адамара, определяемое с помощью матриц вида (3) известно также под названием преобразования Уолша [1].

2) Преобразование Хаара основывается на ортогональной матрице Хаара. Рассмотрим матрицы четвертого и восьмого порядка

$$H_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix},$$

$$H_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}. \quad (5)$$

Матрицы Хаара более высокого порядка строятся по тем же правилам, что и матрицы H_4 H_8 .

В задачах обработки изображений хааровский спектр описывает распределение энергии компонент, соответствующих разностям средних значений яркостей соседних элементов, разностям средних значений яркостей соседних пар элементов и вообще разностям средних значений яркостей соседних групп из 2^m элементов [1].

Быстрые алгоритмы вычислений

В общем случае, чтобы выполнить унитарное преобразование матрицы изображения, содержащей $N \times N$ спектральных коэффициентов, необходимо произвести N^4 арифметических операций. Для многих унитарных преобразований существуют эффективные алгоритмы вычислений, позволяющие ускорить выполнение преобразования [1].

Основной идеей этих быстрых вычислительных алгоритмов является разделение всей задачи на ряд этапов, причем результаты, полученные на предыдущих этапах, многократно используются на последующих этапах.

Рассмотрим процесс вычисления коэффициентов преобразования Адамара с неупорядоченной матрицей для последовательности из четырех элементов $f(j)$. При прямом способе вычисления находятся четыре величины по формулам

$$\begin{aligned} f(0) &= f(0)+f(1)+f(2)+f(3), \\ f(1) &= f(0)-f(1)+f(2)-f(3), \\ f(2) &= f(0)+f(1)-f(2)-f(3), \\ f(3) &= f(0)-f(1)-f(2)+f(3). \end{aligned}$$

Для этого необходимо выполнить $N(N-1)=12$ арифметических операций.

Коэффициенты преобразования Адамара можно найти по-другому, разбив процесс вычисления на два этапа:

I-этап

$$a(0) = f(0)+f(2),$$

$$a(1) = f(0)-f(2),$$

$$a(2) = f(1)+f(3),$$

$$a(3) = f(1)-f(3),$$

II-этап

$$f(0) = a(0)+a(2),$$

$$f(1) = a(0)-a(2),$$

$$f(2) = a(1)+a(3),$$

$$f(3) = a(1)-a(3).$$

При этом для определения элементов матрицы A_4 требуется только $N \log_2 N = 8$ операций.

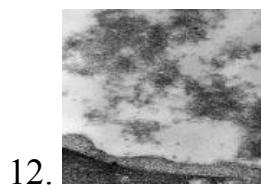
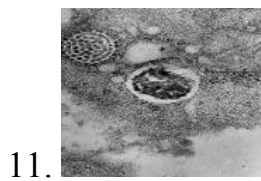
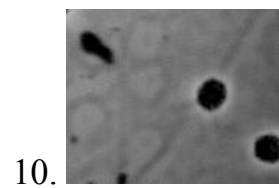
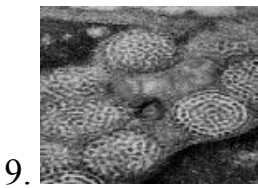
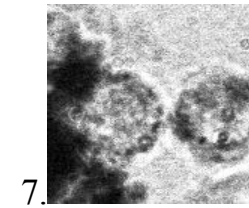
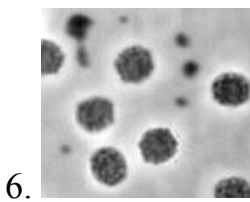
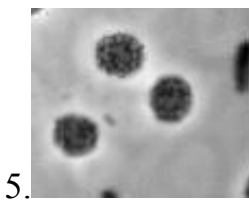
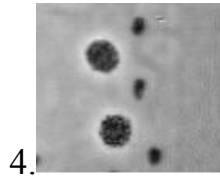
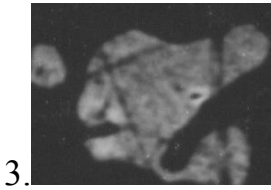
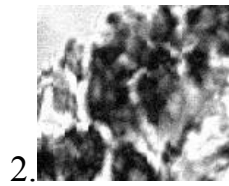
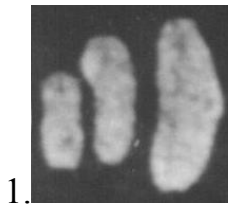
Принципы, описанные выше на примере преобразования Адамара, можно применить для быстрого вычисления многих других преобразований. Разработаны быстрые алгоритмы для преобразований Фурье, четного косинусного, синусного, Адамара, Хаара и наклонного. В общем случае для преобразования Карунена-Лоэва быстрого алгоритма не найдено, однако известны приближенные алгоритмы преобразования Карунена –Лоэва для марковских процессов.

Задание к лабораторной работе №6

1. Для исходного изображения получить преобразованное изображение преобразованием Адамара (быстрым).
2. Для исходного изображения получить преобразованное изображение преобразованием Хаара.

ПРИЛОЖЕНИЕ 1

ОБРАЗЦЫ ИСХОДНЫХ ИЗОБРАЖЕНИЙ ДЛЯ ОБРАБОТКИ



ПРИЛОЖЕНИЕ 2

Министерство образования и науки РФ
Федеральное агентство по образованию
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ГОУВПО «АмГУ»)

Факультет математики и информатики
Специальность 230201 – Информационные системы и технологии
Кафедра информационных и управляющих систем

Отчет
по лабораторной работе № ____

на тему: _____

по дисциплине: «Обработка изображений»

Исполнитель
студент(ка) группы ____

И.О.Ф.

Проверил

Н.П. Семичевская

Благовещенск 2005

СПИСОК ЛИТЕРАТУРЫ

1. **Прэтт У.** Цифровая обработка изображений – М.: Мир, 1982. – 790 с.
2. **Журавлев Ю.И., Гуревич И.Б.** Распознавание образов и распознавание изображений // Распознавание, классификация, прогноз. Математические методы и их применение. – Вып. 2. – М.: Наука. – 1989. – С. 5 – 72.
3. **Павлидис Т.** Алгоритмы машинной графики и обработки изображений. – М.: Радио и связь, 1986. – 400 с.
4. **Ярославский Л.П.** Введение в цифровую обработку изображений. – М.: Сов. радио, 1979. – 312 с.
5. **Ярославский Л.П.** Обработка изображений в медицинской интроскопии / Цифровая оптика в медицинской интроскопии. – М.:ИППИ РАН, 1992. – С. 4–17.
6. **Журавель И.М.** Краткий курс теории обработки изображений/ Консультационный центр MATLAB компании SoftLine, <http://matlab.exponenta.ru/imageprocess/index.php>

Наталья Петровна Семичевская
ст. преподаватель кафедры ИУС

Любовь Александровна Соловцова
ст. преподаватель кафедры ИУС

**Обработка изображений в среде MATLAB с использованием Image
Processing Toolbox.**