**Лабораторная работа № 1**

**«Основы работы с цифровыми изображениями в Matlab»**

**Цель работы.** Изучение стандартных команд в Matlab, предназначенных для считывания изображений из файла, выполнения простейших преобразований изображений, формирования новых изображений, записи изображений в файл.

Простейшими преобразованиями изображений являются: вставка простых геометрических фигур (квадрат, круг, треугольник и др.), изменение цвета или назначение цвета, выделение фрагментов изображения и их перестановка и т.д.

К простейшим операциям не относятся методы попиксельной обработки изображений (изменение яркости, контраста и др.), фильтрации и восстановления изображений.

1. **Основные сведения о работе с изображениями в Matlab**

Основные сведения о работе с изображениями в Matlab изложены в лекциях 1, 2. Для детального изучения любой функции необходимо научиться пользоваться встроенной документацией Matlab: кнопка на панели управления **Help  Documentation**.

Информацию о функции с именем **xxx** можно получить непосредственно в командном окне, набрав команду:

**help xxx;**

Для выполнения задания лабораторной работы № 1 необходимо ознакомиться с работой следующих функций:

**imread; imshow; figure; imwrite;**

**size; round; floor; fix; ceil;**

**rand; randn; zeros; ones; max; min;**

**rgb2gray; im2double;**

* 1. **Типы растровых изображений в Matlab**

Растровые изображения, применяемые в системе Matlab и в пакете расширения Image Processing Toolbox (IPT), могут быть следующего типа:

* Бинарные
* Полутоновые
* Палитровые
* Полноцветные

***Бинарное изображение (Black and White – BW)***. В нем каждый пиксель представлен двумя цветами – белый (1) и черный (0). Таким образом, каждый элемент матрицы бинарного изображения .

***Полутоновые изображения (Grayscale)*** могут иметь пиксели с множеством оттенков серого цвета. Обычно яркость оттенка серого цвета задается действительным числом из отрезка [0, 1]***.*** Эти изображения хранятся в виде двумерных массивов.

В полутоновых изображениях яркость пикселей может быть представлена также действительными числами из любого диапазона [*a*, *b*], где параметры  могут принимать как положительные, так и отрицательные значения.

В полутоновых изображениях яркость пикселей может быть представлена также целыми положительными числами из диапазона [0, 255].

***Палитровые изображения***. В них используются две матрицы: одна, имеющая такой же размер, как само изображение, хранит для каждого пикселя значения индексов палитры, а вторая, называемая ***матрицей палитры***, – цвета палитры в виде трех элементов цветов R,G,B, соответствующие индексам первой матрицы. Поскольку основная матрица изображения хранит не сами цвета палитры, а лишь их индексные значения, то такие изображения называют также ***индексированными изображениями***.

Матрица палитры, называемая также ***цветовой картой***, содержит три группы столбцов R, G, B со значениями элементов из отрезка [0, 1]. Они и задают цвета пикселя с соответствующим значением индекса из первой матрицы.

***Полноцветные изображения.*** В системе Matlab полноцветные изображения строятся в формате RGB. В этом формате изображения хранятся в трехмерном массиве размером . Элементы этого массива , ,  содержат соответственно информацию о яркости красного, зеленого и синего цветов для пикселя .

Каждый элемент массива (пиксель)  хранится в виде целого числа, занимающего 8, 16 или 64 бит, или в виде действительного числа в диапазоне [0, 1].

Для полутоновых и палитровых цифровых изображений для описания яркостей используются значения действительных чисел (формат **double**).

Для полноцветных изображений используются целые числа диапазона [0, 255] – формат **uint8** или диапазона [0, 65535] – формат **uint16**.

В Matlab используется пиксельная координатная система с прямоугольными координатами и начальным пикселом (1,1) в левом верхнем углу. При этом нумерация пикселов идет по строкам – слева направо, по столбцам – сверху вниз.

* 1. **Типы данных в системе Matlab**

В системе Matlab для описания данных в изображениях используются следующие типы:

* **logical** – логический тип
* **double** – тип действительных чисел двойной точности
* **uint8** – целые неотрицательные числа диапазона [0, 255]

Для полноцветных изображений могут использоваться также целые числа **uint16** и **uint64** (соответствующие диапазоны значений  и ), однако в Matlab такое описание используется редко.

Тип **logical** определяет логические (бинарные) переменные, принимающие одно из двух значений: 0 или 1. Этот тип данных используется для описания бинарных изображений (BWI – Black and White Image – черно-белое изображение), для которых значение 0 соответствуют пикселю черного цвета, а 1 – белого.

Для типа **double** можно использовать действительные числа из любого диапазона. При описании яркостей пикселей в полутоновых, палитровых (для матриц палитры) и полноцветных изображений используют, как правило, действительные числа в диапазоне [0, 1]: 0 – минимальная яркость или черный цвет, 1 – максимальная яркость или белый цвет.

Для полноцветных изображений, которые в системе Matlab могут быть представлены только в формате RGB (т.е. в виде трехмерной матрицы), яркость пикселя каждого из трех цветов R, G, B может быть представлена значениями двух типов:

1. **uint8**, реже **uint16** и **uint64**;
2. **double** в диапазоне [0, 1].

В Matlab имеются функции, которые позволяют переводить элементы массивов из одного типа в другой. Эти функции имеют такие же названия, что и тип данных, в который переводится массив: **double, logical, uint8.**

* 1. **Вывод изображения на экран**

В Matlab для вывода изображения на экран используется функция **imshow**, которая для разных форматов изображений может включать различные параметры.

1. **imshow(BM)**

– выводит на экран бинарное изображение, соответствующее массиву **BM**, в котором нулевые элементы массива отображаются черным цветом, а единичные – белым.

1. **imshow(G)**

– выводит на экран полутоновое изображение, соответствующее массиву **G**.

Если элементы массива являются целыми числами, то по умолчанию используется 256 градаций серого цвета.

Если элементы массива являются действительными числами, то необходимо указать диапазон минимального и максимального значения яркостей – параметры **low** и **high** соответственно. Это выполняется путем использования дополнительного векторного параметра в виде

**imshow(G,[low high])**;

Все элементы массива **G**, имеющие значения **low** или меньше, отображаются черным цветом, все элементы со значениями больше или равными **high** отображаются белым цветом. Значения между **low** и **high** отображаются промежуточными оттенками серого цвета.

Если в качестве входного параметра задана пустая матрица **[]** вместо **[low high]**, то функция **imshow** в качестве значений **low** и **high** использует по умолчанию минимальное и максимальное значения яркостей массива **G**, т.е. **low = min(G(:))**, **high = max(G(:))**.

Если значения яркостей заданы действительными числами в диапазоне [0, 1], то параметр **[low high]** можно не указывать, так как диапазон яркостей автоматически устанавливается в интервале [0, 1].

1. **imshow(P,Map)**

– выводит на экран палитровое изображение, соответствующее массиву **P**, с цветовой картой (матрицей палитры) **Map**.

1. **imshow(PRGB)**

– выводит на экран полноцветное RGB изображение, соответствующее матрице **PRGB**, в которой яркости пикселей могут быть представлены значениями двух типов:

**double** на интервале [0, 1];

**uint8**

При использовании функции **imshow** для полноцветного RGB изображения распознавание двух указанных случаев типов данных происходит автоматически. Однако во втором случае, как и для полутоновых изображений, необходимо указать диапазон целочисленных значений яркостей (параметр **[low high]**), который для типов данных **uint8** равен [0, 255].

При использовании функции **imshow** применительно ко всем видам изображений важным параметром является **'InitialMagnification',XXX** , в котором числовое значение **XXX** определяет коэффициент увеличения изображения на экране. Для малоразмерных изображений порядка 10х10 пикселей значение указанного параметра целесообразно выбирать от 500 и более.

Описанные операции по созданию и выводу на экран изображений в Matlab представлены в программе **Image\_Build**, текст которой приведен ниже. В этой программе продемонстрированы основные принципы создания изображений следующих видов:

– бинарные (Black and White – BW);

– полутоновые (Grayscale или Intensity);

– палитровые или индексированные (Indexed);

– полноцветные (RGB).

**clear all; close all;**

**%% Image\_Build**

**%% 1.Изображения бинарные: Black and White (BW)**

**nb = 17; % размер бинарного изображения**

**nb2 = floor(nb/2); % номер центрального пикселя**

**B = ones(nb,nb);**

**B(:,nb2:nb2+2) = zeros(nb,3);**

**B(1:2,nb2-2:nb2-1) = zeros(2,2);**

**B(nb-1:nb,nb2-2:nb2+4) = zeros(2,7);**

**% A - бинарное изображение с действит. числами двух видов {0,1}**

**figure(10); imshow(B,'InitialMagnification',2000); pause;**

**B = logical(B); % преобразование в битовый формат**

**figure(11); imshow(B,'InitialMagnification',2000); pause;**

**%% 2.Изображение полутоновое: Grayscale или Intensity**

**ng = 17; % размер изображения**

**G0 = rand(ng,ng); % случайное распределение яркости**

**G0u = uint8(255\*G0); % преобразование в целочисленный формат**

**G0d = 10\*G0 - 100; % изменение диапазона действительных чисел**

**figure(20); imshow(G0, 'InitialMagnification',2000); pause;**

**figure(25); imshow(G0u,'InitialMagnification',2000); pause;**

**figure(26); imshow(G0d,[],'InitialMagnification',2000); pause;**

**G1 = G0.^4; % ослабление яркости**

**figure(21); imshow(G1, 'InitialMagnification',2000); pause;**

**G2 = G0.^0.25; % усиление яркости**

**figure(22); imshow(G2, 'InitialMagnification',2000); pause;**

**%% 3.Изображения Палитровые: Indexed**

**np = 100; % размер изображения**

**P = round(np\*rand(np,np)); % случайное распределение цветов**

**P\_palet\_w = winter(np); % матрица палитры winter (ColorMap)**

**P\_palet\_h = hot(np); % матрица палитры hot**

**figure(30); imshow(P,P\_palet\_w,'InitialMagnification',2000); pause;**

**P1 = sort(P); % сортировка элементов в столбце**

**figure(31); imshow(P1,P\_palet\_w,'InitialMagnification',2000); pause;**

**P2 = zeros(np,np);**

**for in = 1:np**

**P2(in,1:np) = 1:np; % распределен. яркости (по возр.)**

**end**

**figure(32);imshow(P2,P\_palet\_w,'InitialMagnification',2000);**

**pause;**

**figure(33); imshow(P2,P\_palet\_h,'InitialMagnification',2000); pause;**

**%% 4.Изображения полноцветные: True Color или RGB-изображения**

**ntr = 100; % размер изображения**

**% задание цвета с помощью действительных чисел [0,1]**

**Black\_img = zeros(ntr,ntr,3); % черный квадрат**

**White\_img = ones (ntr,ntr,3); % белый квадрат**

**Redqw\_img = Black\_img; Redqw\_img(:,:,1) = 1; % красный квадрат**

**Green\_img = Black\_img; Green\_img(:,:,2) = 1; % зеленый квадрат**

**Blueq\_img = Black\_img; Blueq\_img(:,:,3) = 1; % синий квадрат**

**Grayq\_img = 0.5\*White\_img; % серый квадрат**

**figure(41); imshow(Black\_img,'InitialMagnification',2000);pause;**

**figure(42); imshow(White\_img,'InitialMagnification',2000);pause;**

**figure(43); imshow(Grayq\_img,'InitialMagnification',2000);pause;**

**figure(44); imshow(Redqw\_img,'InitialMagnification',2000);pause;**

**figure(45); imshow(Green\_img,'InitialMagnification',2000);pause;**

**figure(46); imshow(Blueq\_img,'InitialMagnification',2000);pause;**

**figure(50);**

**imshow(Redqw\_img,[0 255],'InitialMagnification',2000);**

**%% 5. Запись изображения в файл**

**imwrite(B,'Img\_BW.bmp','bmp');**

**imwrite(G0,'Img\_Gray.bmp','bmp');**

**imwrite(P2,P\_palet\_h,'Img\_Pal','bmp');**

**imwrite(Redqw\_img,'Img\_RGB.bmp','bmp');**

**imwrite(B,'Img\_BW.jpg','jpg');**

**imwrite(G0,'Img\_Gray.jpg','jpg');**

**imwrite(P2,P\_palet\_h,'Img\_Pal','jpg');**

**imwrite(Redqw\_img,'Img\_RGB.jpg','jpg');**

**imwrite(B,'Img\_BW.tif','tif');**

**imwrite(G0,'Img\_Gray.tif','tif');**

**imwrite(P2,P\_palet\_h,'Img\_Pal','tif');**

**imwrite(Redqw\_img,'Img\_RGB.tif','tif');**

* 1. **Запись изображения в файл**

После создания и обработки изображения его можно записать в файл с помощью функции **imwrite**. Параметрами этой функции являются: имя массива данных изображения; имя файла, в который записывается изображение (указывается в апострофах); формат файла изображения **bmp**, **jpg**, **tif**, **pcx** и др. (указывается в апострофах). Для палитровых изображений после имени массива данных указывается также имя матрицы палитры.

Примеры записи различных изображений в файлы различных форматов приведены в программе **Image\_Build**.

1. **Порядок выполнения работы и содержание отчета**
   1. Получить у преподавателя задание на обработку изображения или создание нового изображения (задание состоит из двух частей).
   2. Составить описание алгоритма и код Matlab-программы.
   3. Выполнить задание, построив преобразованные изображения.
   4. Составить отчет, содержащий:

– титульный лист, оформленный по стандартной форме;

– задание на лабораторную работу;

– описание алгоритма и текст Matlab-программы;

– изображения, соответствующие выполненному заданию на работу.