

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Таныгин Максим Олегович

Должность: и.о. декана факультета фундаментальной и прикладной информатики

Дата подписания: 21.09.2023 13:13:44

Уникальный программный ключ:

65ab2aa0d384efe8480e6a4c688eddbc475e411y

МИНОБРНАУКИ РОССИИ

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

Кафедра программной инженерии



УТВЕРЖДАЮ:

проректор по учебной работе

Локтионова О.Г.

2022 г.

ОБРАБОТКА РАЗЛИЧНЫХ ТИПОВ ИЗОБРАЖЕНИЙ В СРЕДЕ MATLAB

Методические указания для выполнения лабораторных работ по
дисциплине «Методы и алгоритмы обработки изображений» для
студентов направления подготовки 09.03.04 «Программная
инженерия»

Курск 2022

УДК 004.932

Составитель: Р.А. Томакова

Рецензент

Кандидат технических наук, доцент *А.В. Мальшев*

Обработка различных типов изображений в среде MATLAB: методические указания для проведения лабораторных работ и выполнения самостоятельной внеаудиторной работы по дисциплине «Методы и алгоритмы обработки изображений» для студентов направления подготовки бакалавров 09.03.04 «Программная инженерия»/ Юго-Зап. гос. ун-т; сост. Р.А. Томакова. Курск, 2022. –21с.

Рассмотрена методика изучения различных типов изображений, функций (средств) системы MATLAB, применяемых для работы с файлами изображений.

Методические указания составлены в соответствии с федеральным государственным образовательным стандартом высшего образования направления подготовки 09.03.04 «Программная инженерия» и на основании учебного плана направления подготовки 09.03.04 «Программная инженерия» (профиль «Разработка программно-информационных систем»).

Предназначены для студентов, обучающихся по направлению подготовки 09.03.04 «Программная инженерия» (профиль «Разработка программно-информационных систем») всех форм обучения.

Текст печатается в авторской редакции

Подписано в печать 2022г. Формат 60×84 1/16.
Усл. печ. л. . Уч.- изд. л. . Тираж 25 экз. Заказ. Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

ТИПЫ ИЗОБРАЖЕНИЙ И РАБОТА С ФАЙЛАМИ ИЗОБРАЖЕНИЙ

Цель и задачи лабораторного занятия (лабораторной работы): изучение возможностей обработки различных типов изображений, функций (средств), используя систему MatLab, приобретение практических навыков их приложения.

Планируемые результаты обучения (формируемые знания, умения, навыки и компетенции):

Код и наименование индикатора достижения компетенции, закрепленного за дисциплиной : ПК-2.1 -Готовит отчёты, публикации, презентации по результатам выполненной работы

Необходимые материально-техническое оборудование и материалы:

1. Класс ПЭВМ - Athlon 64 X2-2.4; Cel 2.4, Cel 2.6, Cel 800.
2. Мультимедиа центр: ноутбук ASUS X50VL PMD T2330/14"/1024Mb/ 160Gb/ сум-ка/проектор inFocus IN24+ .
3. Экран мобильный Draper Diplomat 60x60
4. Доступ в сеть Интернет.

Шкала оценивания и критерии оценивания выполненной лабораторной работы:

Форма контроля	Минимальный балл		Максимальный балл	
	балл	примечание	балл	примечание
Лабораторная работа № 1 (Работа с файлами различных типов изображений в среде MATLAB)	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»

План проведения лабораторного занятия (лабораторной работы)

1. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

1. Получить информацию об изображении из файла 'butterfly.bmp'.
2. Прочитать изображение из файла 'butterfly.bmp', преобразовать его в изображение: а) полутоновое б) палитровое. Вывести на экран полутоновое изображение, дополнительно осуществив контрастирование таким образом, чтобы пиксели, яркость которых меньше или равна 0.2, отображалась черным цветом, а пиксели, яркость которых больше или равна 0.8 – белым.
3. Преобразовать пиксели полноцветного и полутонового изображений в формат double.
4. Вывести полноцветное и полутоновое изображения на экран в одно окно, разместив их по вертикали.
5. Вырежьте произвольный фрагмент одного из загруженных ранее изображений. Выведите на экран исходное изображение и вырезанный фрагмент в среде Matlab.
6. Сохраните вырезанный фрагмент в файл, в формате JPEG и посмотрите его в программе промотора изображений.
7. Склейте несколько заранее выделенных фрагментов и посмотрите полученный результат в окне figure.
8. Произвести сравнительный анализ результатов.
9. Сделать выводы по работе.
10. Представить отчет.

2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Система MATLAB (MATrix LABoratory – матричная лаборатория) была разработана фирмой MathWorks в 1984 году, в США. MATLAB представляет собой программный комплекс, дающий пользователю большие возможности для осуществления инженерных и научных расчетов в интерактивном режиме [1-5].

Пакет прикладных программ MATLAB предоставляет широкие возможности для работы с изображениями. Раздел Image Processing Toolbox (IPT) содержит исчерпывающий набор стандартных алгоритмов и графических инструментов, применяемых для обработки изображений. Большинство функций

этого раздела выполнены на языке высокого уровня MATLAB и могут быть реализованы и адаптированы под конкретные задачи [7, 8].

В университетской среде MATLAB является стандартом вычислительных инструментов, используемых для изучения технических дисциплин.

Хранение информации об изображениях выполнено в форме матриц, к которым применимы практически все матричные операции среды Matlab. В зависимости от особенностей обрабатываемых изображений, Matlab использует 4 основных формата их представления.

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 **пикселем** (от *pixel, picture element* – элемент изображения), или точкой.

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

$$V = (c \cdot r \cdot d) / 8,$$

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

Существуют следующие типы изображений:

бинарные – (*black and white*) пиксели могут принимать только два значения: 0 и 1 (черный и белый цвет);

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

палитровые (*indexed*) – значения пикселей являются ссылками на ячейки карты цветов (*colormap*), которые и содержат описания цвета пикселя в некоторой цветовой системе (палитре);

полноцветные или просто **цветные** (*truecolor, rgb*) – изображения, пиксели которого непосредственно хранят информацию об интенсивностях цветовых составляющих.

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

Таблица 1

Диапазоны представления пикселей изображения

Тип изображения	<code>double</code>	<code>uint8</code>
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 палитры	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) пикселя 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 пикселя **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**.

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
<code>'bmp'</code>	Windows Bitmap (BMP)
<code>'tif'</code> или <code>'tiff'</code>	Tag Image File Format (TIFF)
<code>'jpg'</code> или <code>'jpeg'</code>	Joint Photographic Experts Group (JPEG)

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 девяти полей структуры `info`

Имя поля	Тип данных	Описание
<code>Filename</code>	Строка	Имя файла, если файл находится

		в текущей директории, или полный путь к файлу
FileModDate	Строка	Дата и время последней модификации файла
FileSize	Число	Размер файла в байтах
Format	Строка	Формат файла
FormatVersion	Строка или число	Версия формата
Width	Число	Ширина изображения в пикселях
Height	Число	Высота изображения в пикселях
BitDepth	Число	Глубина цвета изображения в битах на пиксель
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.

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

```
>> f = imfinfo(,с:\Image\Athena.bmp').
```

Загрузка изображений в среду MATLAB.

Загрузка изображения из файла осуществляется с помощью команды

```
>> A = imread(filename, fmt);
```

где **A** – имя переменной, в которую будет загружаться изображение, оно может быть выбрано любым.

Если имя переменной не указывать, и ввести команду

```
>> imread(filename, fmt);
```

то имя переменной, в которую будет загружено изображение, автоматически определяется, как **ans**.

В параметре **filename** в одинарных кавычках указывается имя загружаемого файла и путь к нему, например

```
>>A = imread('E:\image' , 'jpg').
```

В параметре **fmt** в одинарных кавычках указывается расширение загружаемого файла. Matlab поддерживает следующие расширения:

BMP – Windows® Bitmap;

CUR – Cursor File;

GIF – Graphics Interchange Format;

HDF4 – Hierarchical Data Format;

ICO – Icon File;

JPEG – Joint Photographic Experts Group;

PBM – Portable Bitmap;

PCX – Windows® Paintbrush;

PGM – Portable Graymap;

PNG – Portable Network Graphics;
PPM – Portable Pixmap;
RAS – Sun™ Raster;
TIFF – Tagged Image File Format;
XWD – X Window Dump.

При загрузке индексного изображения используется команда

```
>> [ A, map] = imread(filename, fmt);
```

где **map** – матрица цветовой палитры.

Загрузка изображения из сетевого ресурса производится аналогично

```
>> A = imread(URL);
```

где **URL** – адрес изображения в сети, в одинарных кавычках, например

```
>>A= imread(192.168.10.200:8080/cam_1.jpg');
```

```
или >>A= imread(10.20100.38:20000/cam_1.jpg').
```

Загрузите изображение из любого графического файла на вашем компьютере с помощью команды **imread** в среду **Matlab**.

После выполнения загрузки в рабочем пространстве системы MATLAB (Workspace) появляется переменная **A**, представляющая массив загруженного изображения, а справа от неё – данные о размере изображения и его формате, которые представлены на рисунке 1.

Задание 2. Прочитать палитровое изображение из файла Handshak.bmp.

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

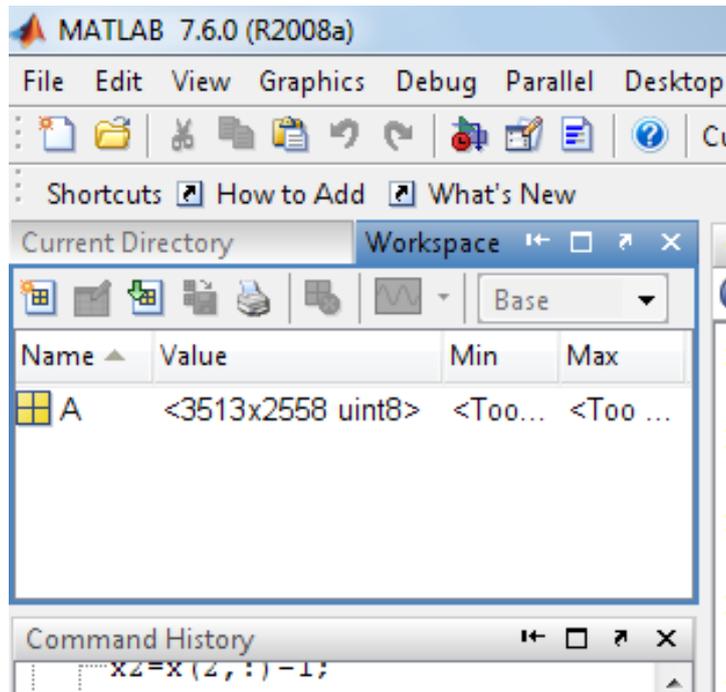


Рисунок 1– Данные об изображении в рабочем окне пространства системы MATLAB

*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.*

Пиксели, 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 $low = \min(I(:))$, $high = \max(I(:))$.

Для того чтобы вывести на экран прочитанное палитровое изображение из предыдущего примера, надо использовать команду:

>> imshow(X,map).

Прочитанное из файла изображение имеет формат представления данных `uint8`.

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 пикселя 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 пикселей к диапазону [0,1].

XD = im2double(XS,'indexed') – преобразует палитровое изображение *XS* в формат *double*.

Функция **mat2gray** позволяет преобразовывать произвольный массив *S*.

Синтаксис

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

XD = im2double(XS,'indexed') – преобразует палитровое изображение *XS* в формат *double*.

Функция **mat2gray** позволяет преобразовывать произвольный массив *S* формата *double* в перенормированный *D*. Изображение *D* имеет значения пикселей 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 пиксели в виде целых неотрицательных чисел в диапазоне [0, 255].

`XD = im2uint8 (XS, 'indexed')` – преобразует палитровое изображение XS в формат uint8.

Преобразования типов изображений

Существуют ряд функций, применяемых 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.

Задание 3. Преобразовать палитровое изображение, прочитанное из файла 'Handshak.bmp' каталога Image, в полутоновое и вывести изображения в одно окно по горизонтали.

```
>> [X,map] = imread('c:\Image\Handshak.bmp');
>> Xd = im2double(X, 'indexed');
>> I = ind2gray(X,map);
>> figure,subplot(1,2,1),subimage(X,map);
>> subplot(1,2,2),subimage(I).
```

Вырезание и склеивание фрагментов изображения.

Для вырезания и склеивания фрагментов изображения в среде MATLAB можно использовать стандартные процедуры работы с матрицами.

Для того чтобы вырезать фрагмент из бинарного, полутонового или индексного изображения, следует выполнить команду

```
>>A1 = A(y1 : y2, x1 : x2, :);
```

где y_1 – номер первого пикселя вырезаемого фрагмента, считая сверху вниз;

y_2 – номер последнего пикселя вырезаемого фрагмента, считая сверху вниз;

x_1 – номер первого пикселя вырезаемого фрагмента, считая слева на право;

x_2 – номер последнего пикселя вырезаемого фрагмента, считая слева на право.

Фрагмент полноцветного изображения вырезается командой

$$\gg A1 = A(y_1 : y_2, x_1 : x_2, :);$$

➤ *Вырежьте произвольный фрагмент одного из загруженных ранее изображений. Выведите на экран исходное изображение и вырезанный фрагмент в среде Matlab, как это показано на рисунке 2.*

➤ *Сохраните вырезанный фрагмент в файл, в формате JPEG и посмотрите его в программе промотора изображений (например, в программе paint).*

Горизонтальное склеивание фрагментов выполняется по команде

$$\gg A1 = [A \ B];$$

а **вертикальное склеивание** – по команде

$$\gg A1 = [A; \ B];$$

Необходимым условием успешного склеивания является равенство горизонтальных (в первом случае) или вертикальных (во втором случае) размеров.

➤ *Склейте несколько заранее выделенных фрагментов и посмотрите полученный результат в окне figure.*

3. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАДАНИЙ

Для преобразования полноцветного изображения в палитровое была использована команда:

```
>> [XMap] = rgb2ind(Image, 256);
```

Чтобы вывести полутоновое изображение на экран с контрастированием, была выполнена следующая команда:

```
>>figure, imshow(GrayImage, [255 * 0.2 255 * 0.8]);
```

Сравнение полутонового изображения с контрастированием с изображением без контрастирования представлено на рисунке 2.

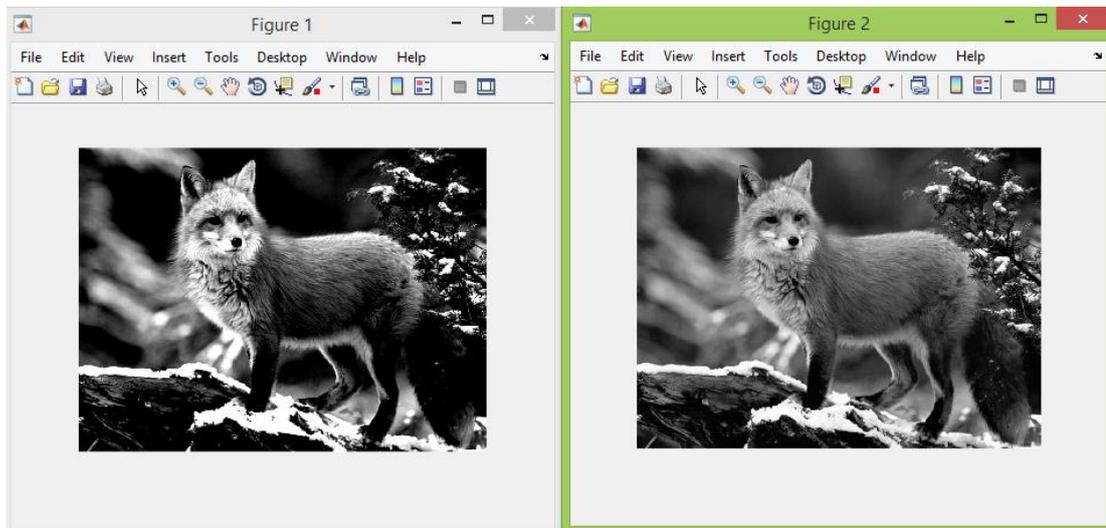


Рисунок 2 – Вывод полутонового изображения с контрастированием (Figure1) с полутоновым изображением без контрастирования (Figure2)

Для размещения двух изображений (полноцветного и полутонового) по вертикали в одном окне была выполнена команда:

```
>>figure,subplot(2,1,1), subimage(D), subplot(2,1,2), subimage(D1);
```

Размещение изображений представлено на рисунке 3.

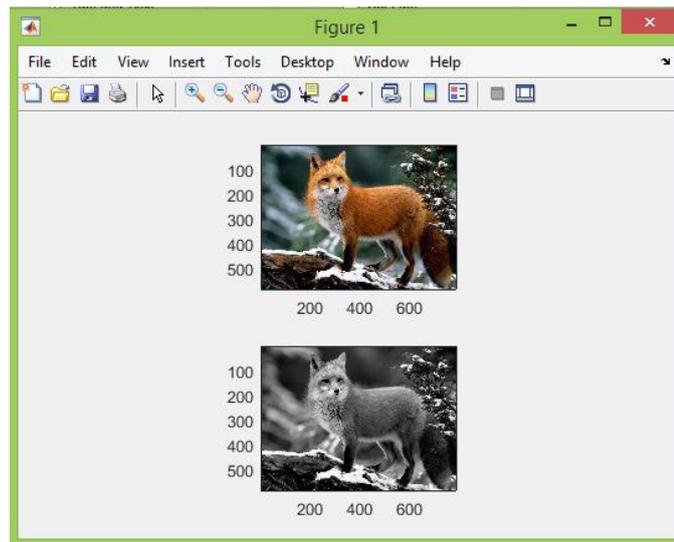


Рисунок 3 – Вывод в одном окне полноцветного и полутонового изображений, размещенных по вертикали

Чтобы вырезать фрагмент из изображения необходимо использовать команду:

```
>>A1 = Image(10 : 100, 200: 300,:);
```

Для отображения вырезанного фрагмента и исходного изображения была вызвана команда:

```
>>figure,subplot(1,2,1), subimage(A1), subplot(1,2,2),  
subimage(Image);
```

Результат отображения представлен на рисунке 4.

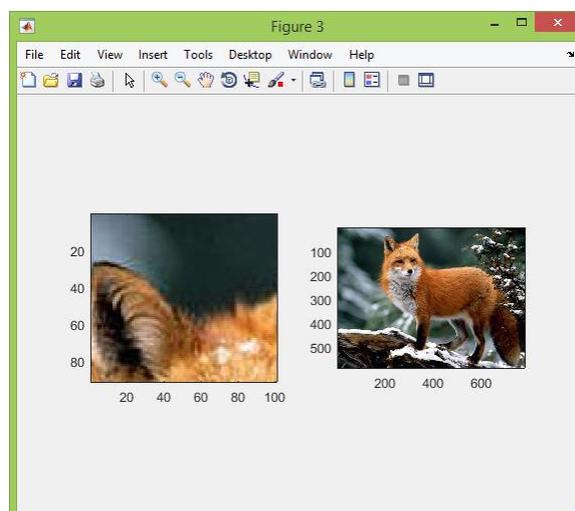


Рисунок 4 – Вывод вырезанного фрагмента и исходного изображения

Выделим еще несколько фрагментов изображения при помощи команд:

```
>> A2= Image (150 : 240, 400 : 500,:);
```

```
>> A3= Image (240 : 310, 400 : 500,:);
```

```
>> A4= Image (240 : 310, 600 : 700,:);
```

Для склеивания фрагментов изображения по горизонтали и по вертикали необходимо использовать команду:

```
V=[A1 A2;A3 A4];
```

Склеенные по горизонтали и по вертикали фрагменты изображения представлены на рисунке 5:

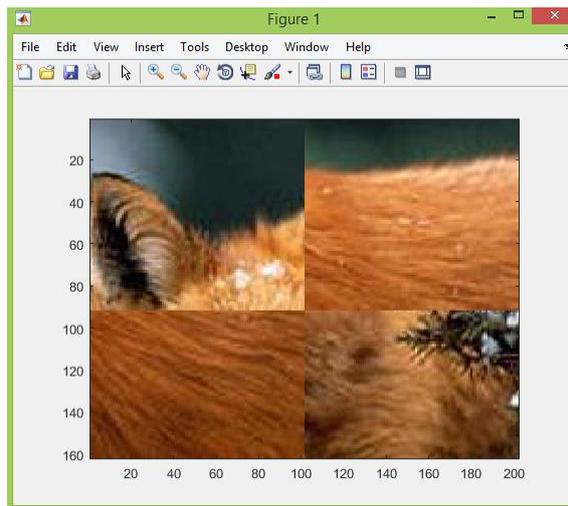


Рисунок 5 – Изображение, полученное путем склейки четырех фрагментов по горизонтали и по вертикали

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие классы данных (форматы) представления пикселей 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?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Томакова, Р. А. Методы и алгоритмы цифровой обработки изображений : учебное пособие для студентов всех форм обучения по направлениям подготовки 09.03.03 Прикладная информатика; 09.03.04 «Программная инженерия», 09.04.04 «Программная инженерия» (профиль «Разработка информационно-вычислительных систем») / Р. А. Томакова, Е. А. Петрик ; Юго-Зап. гос. ун-т. - Курск : Университетская книга, 2020. - 310 с. - Загл. с титул. экрана. – Текст : электронный.

3. Яне, Бернд. Цифровая обработка изображений : [учебное пособие] / пер. с англ. А. М. Измайловой. - М. : Техносфера, 2007. - 584 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Мир цифровой обработки. XI. 06). - Библиогр.: с. 575-583 (221 назв.). - ISBN 978-5-94836-1 22-2 : 285.00 р. - Текст : непосредственный.

4. Красильников, Н. Н. Цифровая обработка 2D- и 3D-изображений : учебное пособие / Н. Н. Красильников. - СПб. : БХВ-Петербург, 2011. - 608 с. : ил. - ISBN 978-5-9775-07 00-4 : 372.50 р. - Текст : непосредственный.

5. Томакова, Римма Александровна. Интеллектуальные технологии сегментации и классификации биомедицинских изображений : монография / Р. А. Томакова, С. Г. Емельянов, С. А. Филист ; Юго-Западный государственный университет. - Курск : ЮЗГУ, 2012. - 222 с. - Текст : электронный.

6. Томакова, Римма Александровна . Теоретические основы и методы обработки и анализа микроскопических изображений биоматериалов : монография / Р. А. Томакова, С. А. Филист, С. Г. Емельянов ; МИНОБРНАУКИ РФ, Юго-Западный государственный университет. - Курск : ЮЗГУ, 2011. - 202 с. - Текст : электронный.

7. Методы цифровой обработки изображений : учебное пособие / А. Е. Архипов, С. В. Дегтярев, С. С. Садыков, С. Н. Середа, В. С. Титов. - Курск : КГТУ, 2002 - Ч. 2. - 115 с. - Текст : непосредственный.

9. Технологии сетей связи. Особенности кодирования цифровых факсимильных сообщений : учебное пособие / А. И. Атакищев [и др.] ; Министерство образования Российской Федерации, Курский государственный технический университет. - Курск : КурскГТУ, 2002. - 159 с. : ил. табл. - Имеется электрон.

аналог. - ISBN 5-7681-0111-X : 62.00 р. - Текст : непосредственный.

10. Гонсалес, Р. Цифровая обработка изображений: практические советы : монография / Р. Гонсалес, Р. Вудс ; пер. П. А. Чочиа, Л. И. Рубанова. – 3-е изд., испр. и доп. – Москва : Техносфера, 2012. – 1104 с. :– (Мир цифровой обработки).– URL: <https://biblioclub.ru/index.php?page=book&id=233465> (дата обращения: 23.03.2022).

11. Фисенко, В.Т. Компьютерная обработка и распознавание изображений / В.Т. Фисенко, Т.Ю. Фисенко. – СПб.: СПбГУ ИТМО, 2008. – 192 с.

12. Гонсалес, Р. Цифровая обработка изображений в среде Matlab / Р. Гонсалес, Р. Вудс, С. Эддинс. – М.: Техносфера, 2006. – 616 с.