

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:

65ab2aa0d384efe8480e6a4c688eddbc475e4113

## 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.

## ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ 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. Малышев*

**Геометрические преобразования 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. 17с.

Рассмотрена методика изучения функций системы MATLAB, применяемых для выполнения геометрических преобразований различных типов изображений, и приобретения практических навыков их использования.

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                    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.

## ГЕОМЕТРИЧЕСКИЕ ПРЕОБРАЗОВАНИЯ ИЗОБРАЖЕНИЙ

**Цель и задачи лабораторного занятия (лабораторной работы):** изучение функций среды MATLAB, применяемых для геометрических преобразований изображений и их фрагментов; и приобретение практических умений и навыков их использования.

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

*Код и наименование индикатора достижения компетенции, закрепленного за дисциплиной : ПК-4.4*

Выполняет тестирование программного обеспечения

**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 ВЫПОЛНЕНИЯ

1. Получить информацию из файла об изображении (номер изображения).
2. Прочитать изображение из файла (номер изображения).
3. Вырезать квадратный фрагмент D с помощью мыши.
4. Вырезать фрагмент, задавая в команде положение координаты его верхнего левого угла как целую часть от  $1/3$  ширины и высоты изображения соответственно, значения ширины и высоты фрагмента определяются также.
5. Вырезать фрагмент с помощью мыши из текущего окна без задания для него переменной и из полученного окна вырезать квадратный фрагмент с помощью мыши в переменную S.
6. Увеличить фрагмент в (N) раз, используя методы:  
используя методы:  
a) '**nearest**'; b) '**bilinear**'; c) '**bicubic**'.
7. Увеличить изображение в (N) раз, используя методы:  
a) '**nearest**'; b) '**bilinear**'; c) '**bicubic**'.
8. Повернуть изображение на заданный угол по часовой стрелке (угол), используя методы:  
a) '**nearest**'; b) '**bilinear**'; c) '**bicubic**'.
9. Повернуть изображение на заданный угол против часовой стрелки (угол), используя методы:  
a) '**nearest**'; b) '**bilinear**'; c) '**bicubic**'.
10. Уменьшить изображение в (N) раз, используя методы:  
используя методы:  
a) '**nearest**'; b) '**bilinear**'; c) '**bicubic**'.
11. Растянуть изображение по горизонтали и по вертикали в (Sx, Sy) раз.
12. Скомбинировать сдвиг изображения по горизонтали и по вертикали на (a,b) .
13. Перенести изображение по горизонтали и по вертикали на (Sx, Sy)
14. Вывести на экран исходное изображение.
15. Вывести на экран результаты преобразования в разные окна.
16. Вывести на экран результаты преобразования в одно окно.

## 2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

### 2.1 Функции, используемые для преобразования размеров изображения

Геометрические преобразования изменяют пространственное местоположение элементов в изображении.

#### Функция вырезания фрагмента из изображения `imcrop`

Синтаксис

`[D,rect] = imcrop (S)` – для непалитровых изображений, фрагмент задается мышкой.

`[Xd,rect] = imcrop (Xs,map)` – для палитровых изображений, фрагмент задается мышкой.

`D = imcrop (S, rect)` – фрагмент задается в векторе `rect`.

`Xd = imcrop (Xs,map, rect)` – фрагмент задается в векторе `rect`.

`D = imcrop` – функция оперирует с изображением в текущем графическом окне.

`imcrop(S)` – результат вырезания фрагмента из изображения `S` отображается в новом графическом окне.

Для выделения квадратного фрагмента следует при перемещении курсора мыши держать нажатой клавишу `Shift`.

Вектор `rect`, задающий фрагмент, содержит четыре элемента:

`[Xmin Ymin w,h]`, где `Xmin` и `Ymin` – координаты верхнего левого угла прямоугольника; `w` – его ширина; `h` – высота.

**Задание 1.** Вырезать фрагмент с помощью мышки.

```
>> [S,map] = imread('c:\Image\Athena.bmp');
>> figure,imshow(S,map);
>> [A,rect] = imcrop(S,map); % кадрирование мышкой
>> figure,imshow(A,map);
>> rect
```

**Задание 2.** Вырезать фрагмент, заданный с помощью вектора `rect`

```
>> rect = [0,0,112.5,68.5]; % задание информации о фрагменте
>> P = imcrop(S,map,rect);
>> figure,imshow(P,map)
```

#### Функция изменения размеров изображения `imresize`

## Синтаксис

**D = imresize (S, m, method)**

Функция создает изображение **D** меньше **S**, если **m** принадлежит диапазону от 0 до 1. Если **m** больше 1, то **D** больше **S**. Для изменения размеров применяется один из методов интерполяции, который задается во входном параметре **method** в виде одной из следующих строк:

„**nearest**” – использует значение ближайшего пикселя (установлено по умолчанию);

„**bilinear**” – употребляет интерполяцию по билинейной поверхности;

„**bicubic**” – применяет интерполяцию по бикубической поверхности.

**Задание 3.** Увеличить полутоновое изображение

```
>> [S,map] = imread('c:\Image\Athena.bmp');
>> I = ind2gray(S,map); %перевод в полутоновое изображение
>> imshow(I)
>> figure,imshow(imresize(I,2)), title(„nearist“)
>> figure,imshow(imresize(I,2,'bilinear')), title('bilinear')
>> figure,imshow(imresize(I,2,'bicubic')), title('bicubic')
```

**Задание 4.** Уменьшить полутоновое изображение

```
>> X1 = imresize(I,0.5);
>> figure,imshow(X1,[]); title(„nearest“)
>> X2 = imresize(I,0.5,'bilinear');
>> figure,imshow(X2,[]); title('bilinear')
>> X3 = imresize(I,0.5,'bicubic');
>> figure,imshow(X3,[]); title('bicubic')
```

## Функция поворота изображения **imrotate**

### Синтаксис

**D = imrotate(S, angle, method)**

Функция создает изображение **D**, соответствующее повернутому *исходному изображению* **S**, используя один из предопределенных методов интерполяции (см. функцию **imresize**). Угол поворота **angle** задается в градусах. Положительные значения

данного параметра соответствуют повороту против часовой стрелки, а отрицательные – по часовой.

**Задание 5.** Повернуть палитровое изображение на 45 градусов по часовой стрелке.

```
>> [D,map] = imread('c:\image\Chip.bmp');
>> figure,subplot(1,3,1),subimage(imrotate(D,45),map)
>> subplot(1,3,2),subimage(imrotate(D,45,'bilinear'),map);
>> subplot(1,3,3),subimage(imrotate(D,45,'bicubic'),map);
```

## 2.2 Аффинные преобразования

Поворот, изменение размеров изображения относятся к геометрическим преобразованиям, представители которого называются аффинными преобразованиями. Аффинное преобразование можно записать в матричной форме:

$$[x \ y \ 1] = [w \ z \ 1] T \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 0 \end{bmatrix}$$

Такой формулой можно задать *сжатие*, *поворот*, *перенос* или *сдвиг*, соответствующим образом определяя элементы матрицы T.

В таблице 1 показано, как выбирать эти величины для совершения различных преобразований.

## Типы аффинных преобразований

Тип	Аффинная матрица T	Координатное уравнения
Растяжение	$\begin{vmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{vmatrix}$	$X=S_x W$ $Y=S_y Z$
Поворот	$\begin{vmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{vmatrix}$	$x = w \cos \alpha - z \sin \alpha$ $y = w \sin \alpha + z \cos \alpha$
Сдвиг (горизонтальный)	$\begin{vmatrix} 1 & 0 & 0 \\ a & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$	$x=w+az$ $y=z$
Сдвиг (вертикальный)	$\begin{vmatrix} 1 & b & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$	$x=w$ $y=bw+z$
Перенос	$\begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ S_x & S_y & 1 \end{vmatrix}$	$X=W+S_x$ $Y=Z+S_y$

В пакете IPT пространственное преобразование задается в виде так называемой **tform**-структуры.

Для задания структуры можно использовать функцию **maketform**.

**Синтаксис**

**tform = maketform(transftype, T),**

где **transftype** – тип преобразования (строковая константа); **T** – матрица задания аффинного преобразования.

В таблице 2 приведены значения типов преобразования **transftype**.



Типы преобразований функции `maketform`

Тип преобразования	Описание
Affine	Комбинация растяжения/сжатия, поворота, сдвига и переноса. Прямые линии остаются прямыми, параллельные линии остаются параллельными
Box	Независимое растяжение/сжатие и перенос по любой размерности; подмножество аффинных преобразований
Composite	Семейство пространственных преобразований, которые применяются последовательно
Custom	Пространственное преобразование, заданное пользователем, который определяет функции для вычисления $T$ и $T^{-1}$
Projective	Как и при аффинных преобразованиях, прямые линии остаются прямыми, однако параллельные переходят в непараллельные с удаленной точкой пересечения

Для выполнения аффинных преобразований над изображением используется функция **`imtransform`**.

### Синтаксис

**`D = imtransform(I, tform, type, P)`**,

где **`I`** – исходное изображение; **`tform`** – `tform`-структура пространственного преобразования; **`type`** – строковая константа, определяющая метод интерполяции ближайших пикселей для вычисления значения выходного пикселя. Может принимать одно из следующих значений: **`'nearest'`**, **`'bilinear'`** и **`'bicubic'`**. По умолчанию используется **`'bilinear'`**. **`P`** – дополнительные параметры, например, параметр **`P = 'FillValue'`** контролирует цвет, который функция использует для пикселей, которые находятся вне исходного изображения:

```
>> D = imtransform(I, tform, 'FillValue', 0.5);
```

Для демонстрации преобразований часто применяется изображение шахматной доски, которое создается функцией **checkerboard**.

### Синтаксис

**I=checkerboard(N,P,Q),**

где параметр **N** – число пикселей, определяющее размер клетки доски; параметр **P** определяет количество клеток по вертикали (**2P**); параметр **Q** определяет количество клеток по горизонтали (**2Q**). Если параметры **P** и **Q** не указаны, то создается квадратная доска размерностью  $8 \times 8$ .

**Задание 6.** Выполнить аффинные преобразования, в качестве тестового изображения выбрав шахматную доску. Для этого необходимо выполнить следующее:

1. Создать M-функцию `affintr`:

```
function affintr(I,T,type)
tform=makeform(„affine”,T);
I1 = imtransform(I, tform) ;
figure, imshow(I1)
title(type)
```

2. Создать тестовое изображение:

```
>> I=checkerboard(40);
>> figure, imshow(I)
```

3. Выполнить преобразование „Растяжение“:

```
>> T=[3 0 0;0 2 0;0 0 1]; type= „resize”;
>> affintr(I,T,type);
```

4. Выполнить преобразование „Сдвиг“:

```
>> T=[1 0 0;0 .2 0;0 0 1]; type=„Sdvig”;
>> affintr(I,T,type);
```

5. Выполнить преобразование „Поворот“:

```
>> T = [cos(pi/4) sin(pi/4) 0;-sin(pi/4) cos(pi/4) 0; 0 0 1];
type=„Rotate”;
>> affintr(I,T,type);
```

6. Выполнить преобразование комбинацией растяжения, поворота и сдвига:

```
>> Tscale = [1.5 0 0; 0 2 0; 0 0 1]; % растяжение
>> Trot = [cos(pi) sin(pi) 0;-sin(pi) cos(pi) 0; 0 0 1]; % поворот
>> Tshear = [1 0 0; .2 1 0; 0 0 1] ; % сдвиг
```

```

>> T1 = Tscale*Trot*Tshear;
>> tform=maketform('affine',T1); type=„All”;
>> affintr(I,T1,type);
7. Выполнить преобразование „Перенос”:
>> T = [1 0 0; 0 1 0; 50 50 1];
>> tform=maketform(„affine”,T);
>> I1 = imtransform(I, tform, 'XData', [1 320], 'YData', [1 320],
'FillValue', 0.5);
>> figure, imshow(I1)

```

### 2.3 Операции над изображениями, выполняемые на основе индексирования массивов

Изменение размеров изображения можно осуществить, непосредственно задавая необходимую индексацию массива изображения:

*Вырезание фрагмента* –  $I_c = I(Y:Y_m, X:X_n);$

*Зеркальное отражение* изображения

*по вертикали* –  $I_y = I(\text{end}:-1:1, :);$

*Зеркальное отражение* изображения

*по горизонтали* –  $I_y = I(:, \text{end}:-1:1);$

*“прореживание”* изображения –  $I_d = I(1:2:\text{end}, 1:2:\text{end});$

**Задание 7.** Выполнить зеркальное отражение изображения по горизонтали изображения и сделать “прореживание” отраженного изображения по горизонтали.

```

>> [x,map]=imread('c:\image\bigbird.bmp');
>> I=im2double(ind2gray(x,map));
>> figure, imshow(I)
>> Iy = I(:,end:-1:1);
>> figure, imshow(Iy)
>> Id = Iy(:, 1:2:end);
>> figure, imshow(Id)

```

## ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАДАНИЙ

Растянуть изображение по горизонтали и по вертикали в  $(S_x, S_y)$  раз.

```
function affintr(I,T,type)
tform=maketform('affine',T);
I1 = imtransform(I, tform) ;
figure, imshow(I1)
title(type)
```

```
>> T=[4 0 0;0 2 0;0 0 1];
type='resize';
affintr(A,T,type);
```

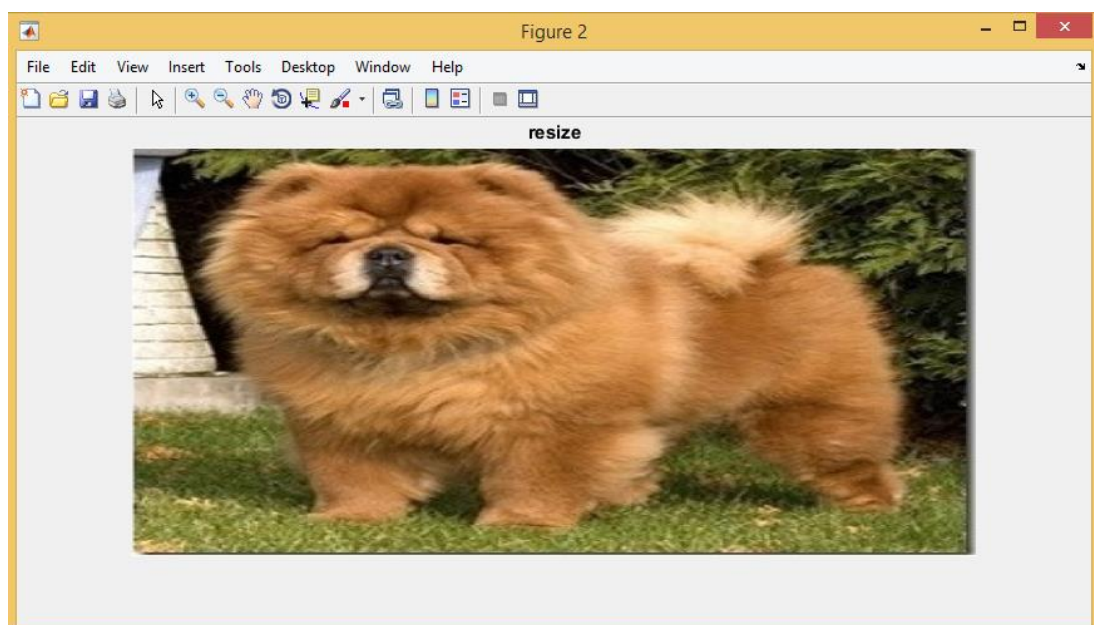


Рисунок 1

Скомбинировать сдвиг изображения по горизонтали и по вертикали на  $(a,b)$ .

```
function DoublSdv(A,T1,T2)
tform=maketform('affine',T1);
A1 = imtransform(A, tform) ;
```

```
tform=maketform('affine',T2);
A2=imtransform(A1, tform) ;
```

```
figure, imshow(A2)
```

```
title('DoubleSdvig');
```

```
>> DoublSdv(A,[1 0 0; 0.3 1 0; 0 0 1],[1 0.5 0; 0 1 0; 0 0 1]);
```

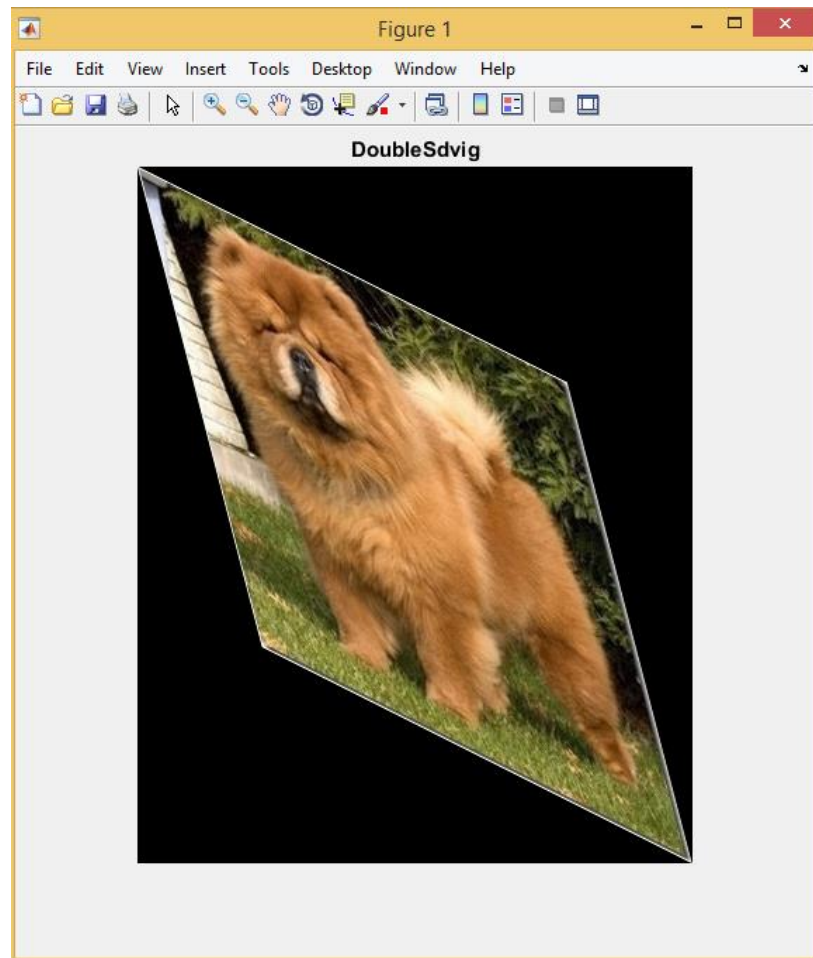


Рисунок 2

Вывести на экран результаты преобразования в одно окно

```
function AllTrans(A)
tform=maketform('affine',[4 0 0;0 2 0;0 0 1]);
A1 = imtransform(A, tform) ;
figure ,subplot (3,1,1), subimage(A1)
title('Stretch');
```

```
tform=maketform('affine',[1 0 0; 0.3 1 0; 0 0 1]);
A11 = imtransform(A1, tform) ;
tform=maketform('affine',[1 0.5 0; 0 1 0; 0 0 1]);
A2=imtransform(A11, tform) ;
subplot (3,1,2), subimage(A2)
title('DoubleSdvig');
```

```

tform=maketform('affine',[1 0 0; 0 1 0; 50 50 1]);
A3 = imtransform(A, tform, 'XData', [1 320], 'YData', [1 320],
'FillValue', 0.5);
subplot (3,1,3), subimage(A3)
title('Replace');

>> AllTrans(A) ;

```

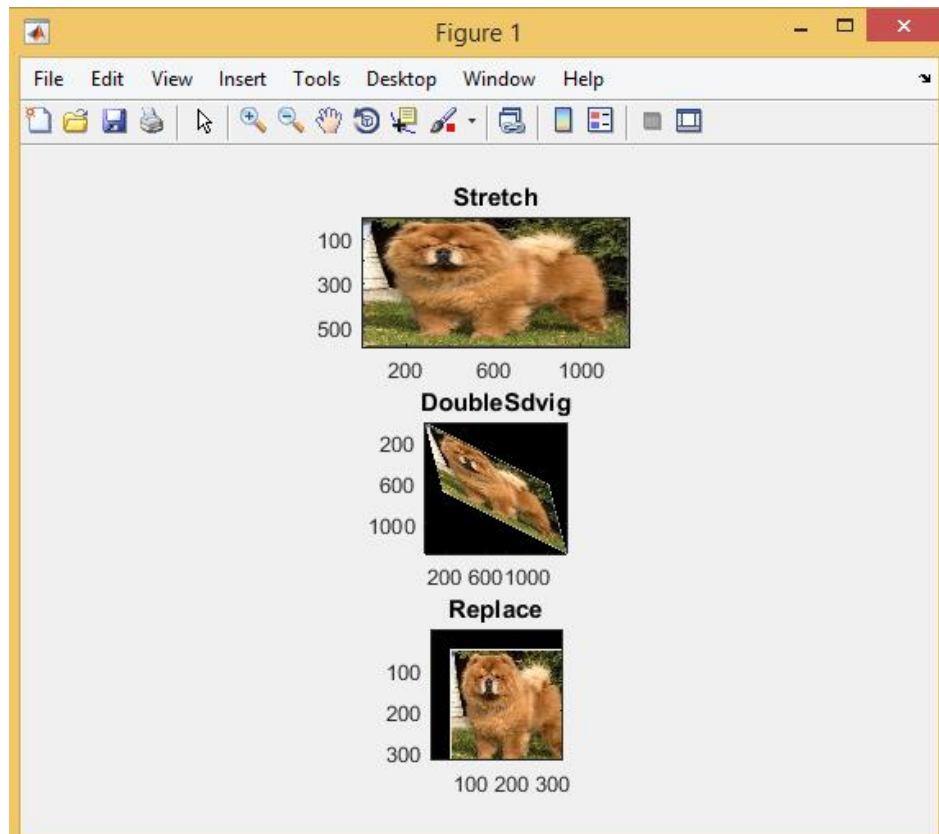


Рисунок 3

### 3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какими способами можно задать вырезание фрагмента для функции `imcrop`?
2. Что является входными аргументами для функции изменения размеров?
3. Что является входными аргументами для функции поворота?
4. Как задать аффинную матрицу?

5. а) для растяжения; б) для сдвига; в) для поворота; г) для переноса?

6. Как можно выполнить зеркальное отображение изображения?

7. Как осуществляется операция «прореживания» изображения?

8. Как можно выполнить комбинированный сдвиг изображения по горизонтали и по вертикали на (a,b)?

9. Как осуществляется операция растяжения изображения по горизонтали и по вертикали в (S<sub>x</sub>, S<sub>y</sub>) раз.

10. Как выполняется изменение размеров изображения, непосредственно задавая необходимую индексацию массива?

## 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. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. - М. :Техносфера, 2006. - 1072 с. - (Мир цифровой обработки). - ISBN 5-94836-028-8 : 394.66 р. - Текст : непосредственный.

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

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