

**Министерство науки и высшего образования  
Российской Федерации**

**Федеральное государственное автономное  
образовательное учреждение высшего образования**

**«Национальный исследовательский университет  
ИТМО»**

**Факультет информационных технологий и  
программирования**

Лабораторная работа № 2

Геометрические преобразования изображения

**Выполнил студент группы № М3302**

Суворин Ярослав Владимирович

Санкт-Петербург  
2025

## 1. Цель работы

Получение основных навыков работы с геометрическими методами обработки цифровых изображений.

## 2. Ход выполнения работы

### 2.а Исходные изображения



### 2.б Листинги программных реализаций

Листинг программ: отражение по горизонтали/вертикали, сдвиг и скос

```
cat=imread('kot1.jpg');
cat_gray=rgb2gray(cat);
cat_flip_gorizontal=flip(cat, 1);
cat_gray_flip_gorizontal=flip(cat_gray, 1);
cat_flip_vertical=flip(cat, 2);
cat_gray_flip_vertical=flip(cat_gray, 2);

T=[1 0 0;
   0 1 0;
   100 100 1];
tform = affine2d(T);
cat_shift = imwarp(cat, tform, 'Interp', 'nearest', 'OutputView', imref2d(size(cat), [1 size(cat,2)], [1 size(cat,1)]));
cat_gray_shift = imwarp(cat_gray, tform, 'Interp', 'nearest', 'OutputView', imref2d(size(cat_gray), [1 size(cat_gray,2)], [1 size(cat_gray,1)]));

T = [1 0 0;
     0.3 1 0;
     0 0 1];
tform = affine2d(T);
cat_bevel = imwarp(cat, tform);
cat_gray_bevel = imwarp(cat_gray, tform);
```

Листинг программ: поворот вокруг начала координат и вокруг центра

```
phi = 30*pi/180;
T = [cos(phi) sin(phi) 0;
     -sin(phi) cos(phi) 0;
     0 0 1];
tform = affine2d(T);
cat_rotate = imwarp(cat, tform, 'OutputView', imref2d(size(cat)));
cat_gray_rotate = imwarp(cat_gray, tform, 'OutputView', imref2d(size(cat_gray)));

center = [size(cat, 2) / 2, size(cat, 1) / 2];
T_center_to_coordinates = [1 0 0;
                           0 1 0;
                           -center(1) -center(2) 1];
T_rotate = [cos(phi) sin(phi) 0;
            -sin(phi) cos(phi) 0;
            0 0 1];
T_coordinates_to_center = [1 0 0;
                           0 1 0;
                           center(1) center(2) 1];
T_center = T_center_to_coordinates * T_rotate * T_coordinates_to_center;
tform_center = affine2d(T_center);
cat_rotate_center = imwarp(cat, tform_center, 'OutputView', imref2d(size(cat)));
cat_gray_rotate_center = imwarp(cat_gray, tform_center, 'OutputView', imref2d(size(cat_gray)));
```

## Листинг программ: масштабирование

```
a=2;
b=2;

T = [a 0 0;
      0 b 0;
      0 0 1];
tform = affine2d(T);
cat_scale_1= imwarp(cat, tform);
cat_gray_scale_1= imwarp(cat_gray, tform);

a=0.1;
b=0.1;

T = [a 0 0;
      0 b 0;
      0 0 1];
tform = affine2d(T);
cat_scale_2= imwarp(cat, tform);
cat_gray_scale_2= imwarp(cat_gray, tform);

a=5;
b=2;

T = [a 0 0;
      0 b 0;
      0 0 1];
tform = affine2d(T);
cat_scale_3= imwarp(cat, tform);
cat_gray_scale_3= imwarp(cat_gray, tform);

a=0.1;
b=0.5;

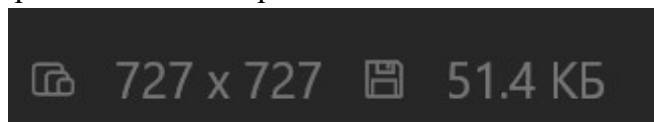
T = [a 0 0;
      0 b 0;
      0 0 1];
tform = affine2d(T);
cat_scale_4= imwarp(cat, tform);
cat_gray_scale_4= imwarp(cat_gray, tform);
```

### 2.с Комментарии

Зеркальное отображение сделал с помощью функции `flip`, для остальных же использовал различных матриц преобразования.

При масштабировании в результирующих изображениях увеличение не совсем понятно, но при сохранении изображение можно увидеть, что эффект был произведен:

Оригинальное изображение:



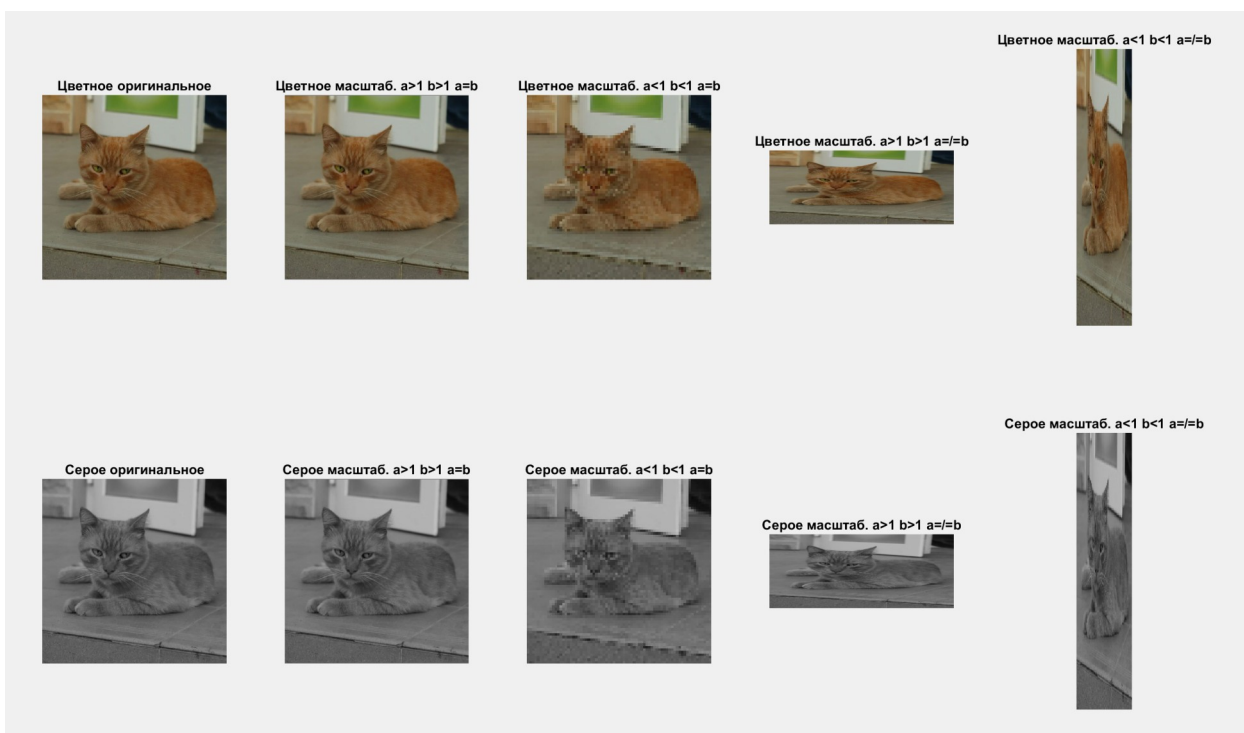
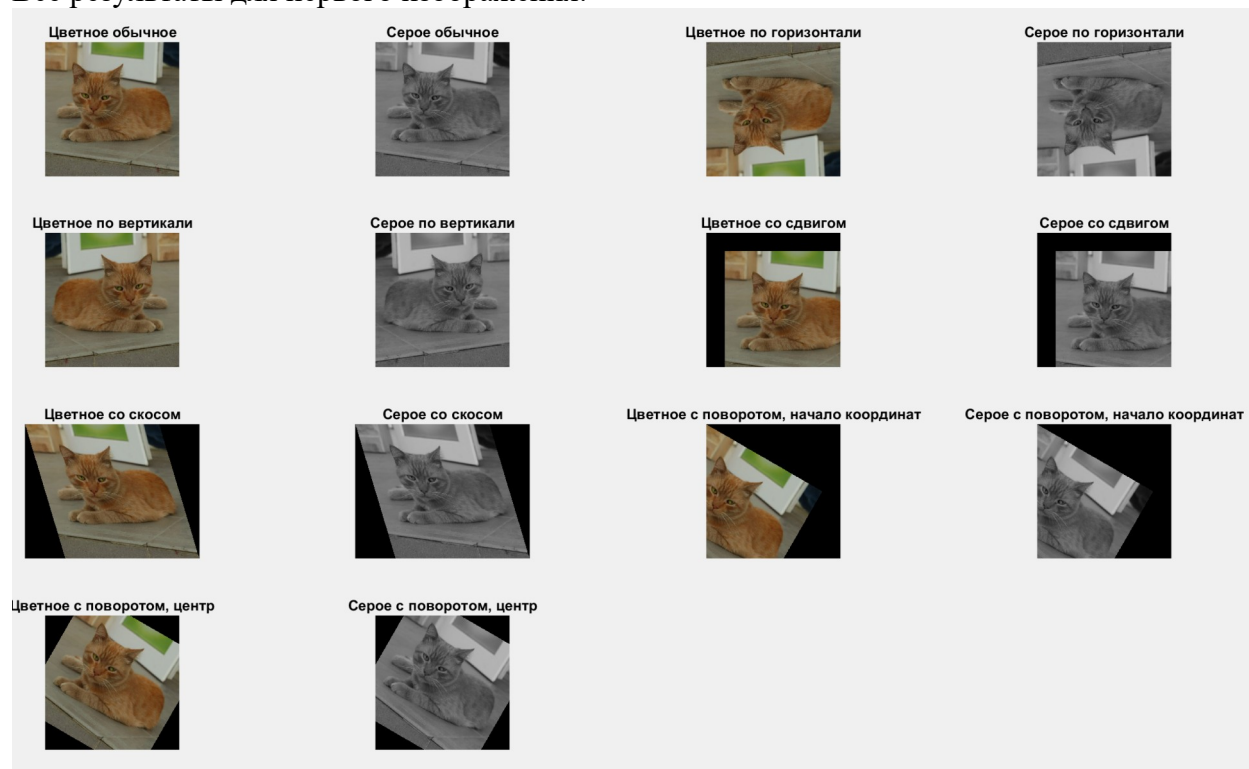
Увеличенное изображение:



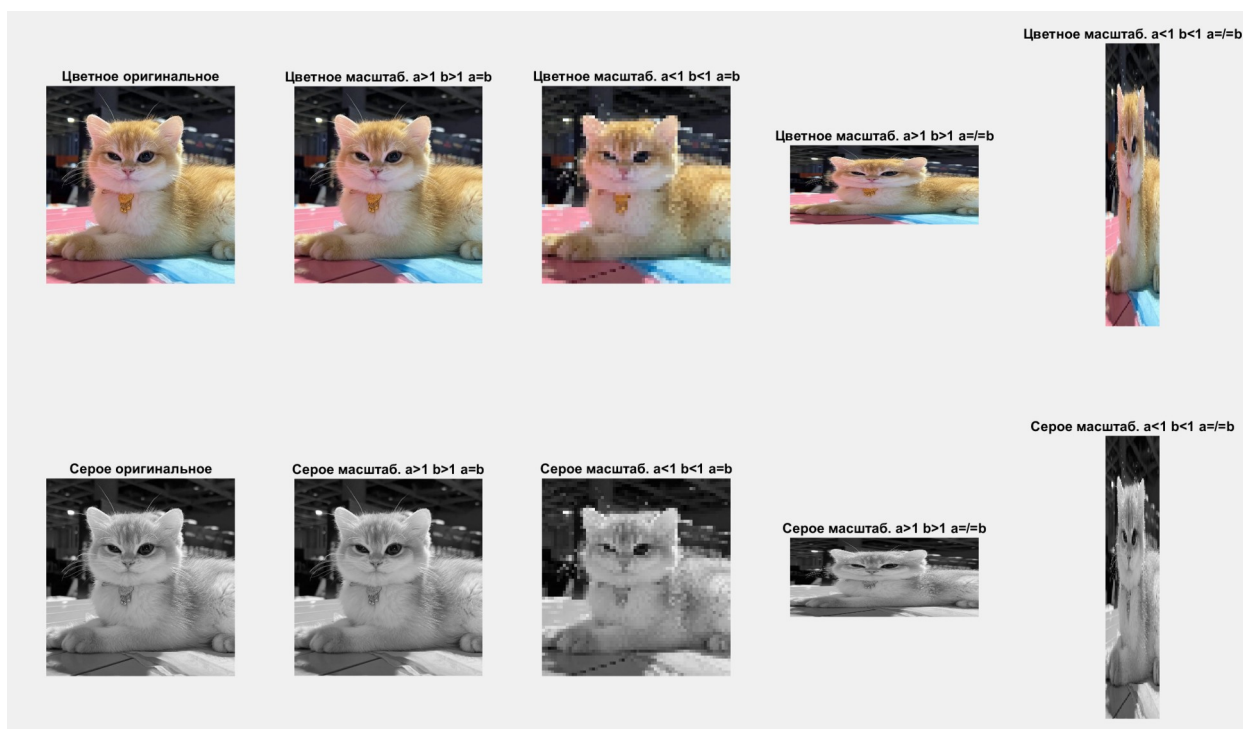
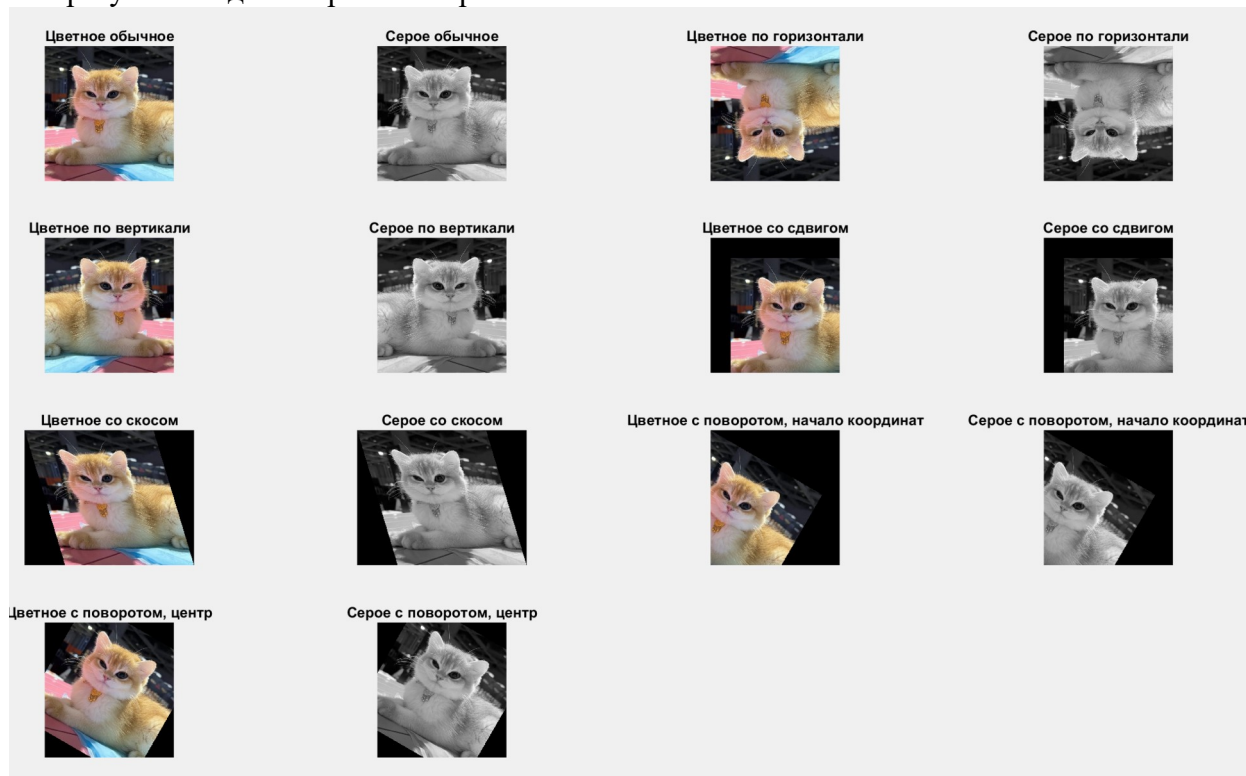
Аналогично как со вторым изображением, так и с grayscale версией.

## 2.d Результирующие изображения

Все результаты для первого изображения:



Все результаты для второго изображения:



### 3. Вывод по работе

В результате проделанной работы получилось многое узнать про геометрические преобразования изображений — на практике использовались преобразования при помощи матриц. Используя перемножение координат точек матрицы изображения и преобразовывающих матриц, точки получали новые координаты. Они могли как просто изменять положение изображения в пространстве (сдвиг, поворот, отражение), так и менять его размер (масштабирование), приближая или отдаляя точки изображения от начала координат, тем самым уменьшая или увеличивая количество пикселей.