**Минобрнауки РФ**

**ЮЗГУ**

**ФФиПИ**

Кафедра программной инженерии

**ЛАБОРАТОРНАЯ РАБОТА №10**

**ИНТЕРПОЛЯЦИЯ ИЗОБРАЖЕНИЙ**

Выполнил: ст. гр. ПО-22б

Алешин К.Р.

Проверил: Ефремов В. В.

Курск, 2024 г

**Задание:**  
Задание выполнить аффинные преобразования масштабирования, поворота и скоса с разными параметрами, применяя разные методы интерполяции.

**Листинг программы:**import cv2

import numpy as np

from matplotlib import pyplot as plt

def apply\_affine\_transformation(image, transformation\_matrix, interpolation):

"""

Применяет аффинное преобразование к изображению с заданной матрицей и методом интерполяции.

"""

rows, cols = image.shape[:2]

transformed = cv2.warpAffine(image, transformation\_matrix, (cols, rows), flags=interpolation)

return transformed

def main():

# Загружаем изображение

image\_path = "input.jpg"

image = cv2.imread(image\_path)

if image is None:

print("Ошибка: изображение не найдено.")

return

rows, cols = image.shape[:2]

# Аффинные преобразования

# 1. Масштабирование

scale\_x = float(input("Введите коэффициент масштабирования по оси X: "))

scale\_y = float(input("Введите коэффициент масштабирования по оси Y: "))

scaling\_matrix = np.array([[scale\_x, 0, 0],

[0, scale\_y, 0]], dtype=np.float32)

# 2. Поворот

angle = float(input("Введите угол поворота (в градусах): "))

rotation\_matrix = cv2.getRotationMatrix2D((cols // 2, rows // 2), angle, 1)

# 3. Скос

skew\_x = float(input("Введите коэффициент скоса по оси X: "))

skew\_y = float(input("Введите коэффициент скоса по оси Y: "))

skew\_matrix = np.array([[1, skew\_x, 0],

[skew\_y, 1, 0]], dtype=np.float32)

# Применение преобразований с различными методами интерполяции

interpolation\_methods = {

"Ближайший сосед": cv2.INTER\_NEAREST,

"Линейная интерполяция": cv2.INTER\_LINEAR,

"Кубическая интерполяция": cv2.INTER\_CUBIC

}

# Отображение результатов

plt.figure(figsize=(12, 12))

plt.subplot(3, 4, 1)

plt.imshow(cv2.cvtColor(image, cv2.COLOR\_BGR2RGB))

plt.title("Исходное изображение")

plt.axis('off')

idx = 2

for name, interpolation in interpolation\_methods.items():

# Масштабирование

scaled\_image = apply\_affine\_transformation(image, scaling\_matrix, interpolation)

plt.subplot(3, 4, idx)

plt.imshow(cv2.cvtColor(scaled\_image, cv2.COLOR\_BGR2RGB))

plt.title(f"Масштабирование\n({name})")

plt.axis('off')

idx += 1

# Поворот

rotated\_image = apply\_affine\_transformation(image, rotation\_matrix, interpolation)

plt.subplot(3, 4, idx)

plt.imshow(cv2.cvtColor(rotated\_image, cv2.COLOR\_BGR2RGB))

plt.title(f"Поворот\n({name})")

plt.axis('off')

idx += 1

# Скос

skewed\_image = apply\_affine\_transformation(image, skew\_matrix, interpolation)

plt.subplot(3, 4, idx)

plt.imshow(cv2.cvtColor(skewed\_image, cv2.COLOR\_BGR2RGB))

plt.title(f"Скос\n({name})")

plt.axis('off')

idx += 1

plt.tight\_layout()

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Пример работы программы:  
Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание**

**Изображение выглядит как снимок экрана, текст, фиолетовый, Красочность

Автоматически созданное описание**