

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет информационных технологий и управления
Кафедра интеллектуальных информационных технологий
Дисциплина «Обработка изображений в интеллектуальных системах»

ОТЧЁТ
к лабораторной работе №5
на тему
«СТЕРЕОСКОПИЧЕСКИЕ НАБЛЮДЕНИЯ И ИЗМЕРЕНИЯ»

БГУИР 6-05-0611-03 130

Выполнил студент группы 321701
СЕМЕНЯКО Владимир Дмитриевич

(дата, подпись студента)

Проверил
САЛЬНИКОВ Даниил Андреевич

(дата, подпись преподавателя)

Минск 2025

1 ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

Реализовать приложение, принимающее на вход два изображения (стереопару) и формирующее на их основе стереоскопическое изображение.

2 ВЫПОЛНЕНИЕ РАБОТЫ

Программа реализована на языке Python с использованием библиотек OpenCV, NumPy и Pillow. Были реализованы два метода отображения стереоизображений: анагlyph и wiggle-анимация

Листинг 1 – Код программы (анаглиф)

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
import sys
import os

def create_anaglyph(left_img, right_img):

    if len(left_img.shape) == 3:
        left_gray = cv2.cvtColor(left_img, cv2.COLOR_BGR2GRAY)
    else:
        left_gray = left_img

    if len(right_img.shape) == 3:
        right_gray = cv2.cvtColor(right_img, cv2.COLOR_BGR2GRAY)
    else:
        right_gray = right_img

    h = min(left_gray.shape[0], right_gray.shape[0])
    w = min(left_gray.shape[1], right_gray.shape[1])
    left_gray = cv2.resize(left_gray, (w, h))
    right_gray = cv2.resize(right_gray, (w, h))

    anaglyph = np.zeros((h, w, 3), dtype=np.uint8)
    anaglyph[:, :, 0] = right_gray
    anaglyph[:, :, 1] = right_gray
    anaglyph[:, :, 2] = left_gray

    return anaglyph

def show_images(left, right, anaglyph):
    plt.figure(figsize=(15, 5))

    plt.subplot(1, 3, 1)
    plt.imshow(cv2.cvtColor(left, cv2.COLOR_BGR2RGB) if len(left.shape) == 3
               else left, cmap='gray')
```

```

plt.titleЛевое(' изображение')
plt.axis('off')

plt.subplot(1, 3, 2)
plt.imshow(cv2.cvtColor(right, cv2.COLOR_BGR2RGB) if len(right.shape) == 3
else right, cmap='gray')
plt.titleПравое(' изображение')
plt.axis('off')

plt.subplot(1, 3, 3)
plt.imshow(cv2.cvtColor(anaglyph, cv2.COLOR_BGR2RGB))
plt.titleАнаглиф(' стерео()')
plt.axis('off')

plt.tight_layout()
plt.show()

def main():
    if len(sys.argv) == 3:
        left_path = sys.argv[1]
        right_path = sys.argv[2]
    else:
        left_path = 'left.jpg'
        right_path = 'right.jpg'

    if not os.path.exists(left_path):
        print(f'Ошибка': файл '{left_path}' не найден.')
        return
    if not os.path.exists(right_path):
        print(f'Ошибка': файл '{right_path}' не найден.')
        return

    left_img = cv2.imread(left_path)
    right_img = cv2.imread(right_path)

    if left_img is None:
        printНе(" удалось загрузить левое изображение.")
        return
    if right_img is None:
        printНе(" удалось загрузить правое изображение.")
        return

    anaglyph = create_anaglyph(left_img, right_img)

    show_images(left_img, right_img, anaglyph)

    cv2.imwrite('stereo_anaglyph.jpg', anaglyph)
    printСтереоскопическое(" изображение сохранено как 'stereo_anaglyph.jpg'
'.')
}

if __name__ == '__main__':

```

```
main()
```

Листинг 2 – Код программы (wiggle-эффект)

```
import cv2
import numpy as np
from PIL import Image
import os
import sys

def create_wiggle_gif(left_path, right_path, output_path='stereo_wiggle.gif',
                      duration=300, n_cycles=5):
    """
    Создаёт wiggleанимацию- (GIF) из стереопары.

Параметры:
    left_path      : путь к левому изображению
    right_path     : путь к правому изображению
    output_path    : имя выходного GIFфайла-
    duration       : задержка между кадрами в миллисекундах рекомендуется( –
                     200500 мс)
    n_cycles       : сколько раз повторить цикл лево[ □ право □ лево]
    """
    # Загрузка изображений
    left_img = cv2.imread(left_path)
    right_img = cv2.imread(right_path)

    if left_img is None:
        raise FileNotFoundError(f'Не найдено левое изображение: {left_path}')
    if right_img is None:
        raise FileNotFoundError(f'Не найдено правое изображение: {right_path}')

    # Преобразование в RGB и приведение к одинаковому размеру
    h = min(left_img.shape[0], right_img.shape[0])
    w = min(left_img.shape[1], right_img.shape[1])

    left_resized = cv2.resize(left_img, (w, h))
    right_resized = cv2.resize(right_img, (w, h))

    # Конвертация из BGR (OpenCV) в RGB (PIL)
    left_rgb = cv2.cvtColor(left_resized, cv2.COLOR_BGR2RGB)
    right_rgb = cv2.cvtColor(right_resized, cv2.COLOR_BGR2RGB)

    # Создание кадров: лево[, право, лево, право, ...]
    frames = []
    for _ in range(n_cycles):
        frames.append(Image.fromarray(left_rgb))
        frames.append(Image.fromarray(right_rgb))

    # Сохранение как GIF
    frames[0].save(
        output_path,
```

```

        save_all=True,
        append_images=frames[1:],
        duration=duration,
        loop=0 # бесконечный цикл
    )
    print(f"Wiggleанимация- сохранена: {os.path.abspath(output_path)}")

def main():
    if len(sys.argv) == 3:
        left_path = sys.argv[1]
        right_path = sys.argv[2]
    else:
        left_path = 'left.jpg'
        right_path = 'right.jpg'

    try:
        create_wiggle_gif(left_path, right_path)
    except Exception as e:
        print(f'Ошибка': {e}')

if __name__ == '__main__':
    main()

```

На вход программы подавалась стереопара изображений с сайта <https://vision.middlebury.edu>
На Рисунках 1-3 показаны исходные изображения и результаты обработки.
Результат wiggle-эффекта в отчете не предоставлен ввиду gif-формата

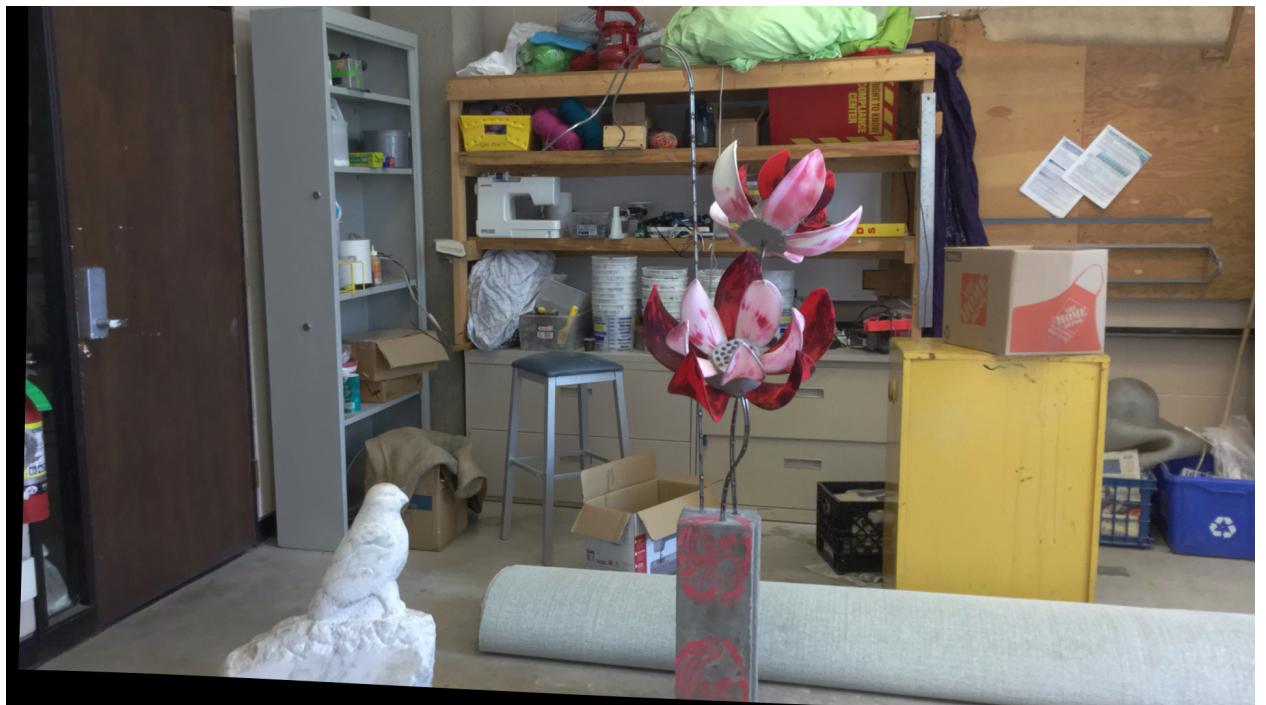


Рисунок 1 – Исходное изображение (левое)



Рисунок 2 – Исходное изображение (правое)



Рисунок 3 – Анаглиф-изображение

ВЫВОД

В ходе выполнения лабораторной работы было разработано приложение для создания стереоскопических изображений двумя методами: анаглифом и wiggle-анимацией. Программа корректно обрабатывает стереопары,

обеспечивая визуальное восприятие глубины. Реализованные подходы демонстрируют основные принципы стереоизрения и могут использоваться в задачах компьютерного зрения, виртуальной реальности и образовательных целях.