**Міністерство освіти і науки України**

**Національний технічний університет України «КПІ» імені Ігоря Сікорського**

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи №2**

**з навчальної дисципліни «Технології Computer Vision»**

**Тема:**

**ДОСЛІДЖЕННЯ АЛГОРИТМІВ ФОРМУВАННЯ ТА ОБРОБКИ РАСТРОВИХ**

**ЦИФРОВИХ ЗОБРАЖЕНЬ**

**Виконав:**

Студент 3 курсу кафедри ІПІ ФІОТ,

Навчальної групи ІП-11

Лошак В.І.

**Перевірив:**

Професор кафедри ОТ ФІОТ

Писарчук О.О.

**Київ 2024**

**І. Мета:**

Виявити дослідити та узагальнити особливості реалізації алгоритмів растрової

цифрових зображень на прикладі застосування алгоритмів растеризації, побудови складних 3D растрових об’єктів та застосування технологій корекції характеристик кольору окремих растрів цифрових зображень. формування та перетворення координат площинних (2d) та просторових (3d) об’єктів.

**ІІ. Завдання:**

***Завдання ІІІ рівня – максимально 9 балів.***

Реалізувати розробку програмного скрипта, що реалізує корекцію кольору цифрового растрового зображення з переліку: зміна яскравості, відтінки сірого, негатив, серпія – в градієнтах: діагональ (будь-який напрям); від центру; до центра. Обробку еалізувати на рівні матриці растра. Зображення обрати самостійно.

**ІІІ. Результати виконання лабораторної роботи.**

Для реалізації перетворення цифрового зображення відповідно до вказаних інструкцій було створено два переліки, кожен з яких відповідно містить перетворення що є взаємовиключними. Два переліки описують тип градієнту та тип ефекту що застосовується до зображення.

A screenshot of a computer program

Description automatically generated

Рис. 1— Визначення типів перетворення.

Операція перетворення зображення вказаним чином реалізується за допомогою функції що модифікує матрицю растра зображення. Спочатку функція визначає вигляд матриці градієнту що буде застосовуватися до зображення а потім створює зображення що відповідає запитаній користувачем модифікації зображення. На фінальному етапі матриця растру, та градієнту об’єднуються в одне зображення.

*Код функції:*

def apply\_gradient\_correction(img, effect\_type: EffectType, gradient\_type: GradientType):

    rows, cols, \_ = img.shape

    gradient = np.zeros((rows, cols))

    if gradient\_type == GradientType.DIAGONAL:

        for i in range(rows):

            for j in range(cols):

                gradient[i, j] = (i + j) / (rows + cols)

    elif gradient\_type == GradientType.FROM\_CENTER:

        center\_i, center\_j = rows // 2, cols // 2

        for i in range(rows):

            for j in range(cols):

                distance = np.sqrt((center\_i - i) \*\* 2 + (center\_j - j) \*\* 2)

                max\_dist = np.sqrt(center\_i \*\* 2 + center\_j \*\* 2)

                gradient[i, j] = 1 - (distance / max\_dist)

    elif gradient\_type == GradientType.TO\_CENTER:

        center\_i, center\_j = rows // 2, cols // 2

        for i in range(rows):

            for j in range(cols):

                distance = np.sqrt((center\_i - i) \*\* 2 + (center\_j - j) \*\* 2)

                max\_dist = np.sqrt(center\_i \*\* 2 + center\_j \*\* 2)

                gradient[i, j] = (distance / max\_dist)

    gradient = np.repeat(gradient[:, :, np.newaxis], 3, axis=2)

    corrected\_img = None

    if effect\_type == EffectType.BRIGHTNESS:

        corrected\_img = np.clip(img \* (1 + gradient), 0, 255)

    elif effect\_type == EffectType.GRAYSCALE:

        gray\_img = img.mean(axis=2)

        corrected\_img = np.repeat(gray\_img[:, :, np.newaxis], 3, axis=2)

    elif effect\_type == EffectType.NEGATIVE:

        corrected\_img = 255 - img

    elif effect\_type == EffectType.SEPIA:

        sepia\_filter = np.array([[0.393, 0.769, 0.189],

                                 [0.349, 0.686, 0.168],

                                 [0.272, 0.534, 0.131]])

        sepia\_img = img.dot(sepia\_filter.T)

        corrected\_img = np.clip(sepia\_img, 0, 255)

    if corrected\_img is not None:

        corrected\_img = img \* (1 - gradient) + corrected\_img \* gradient

        return np.clip(corrected\_img, 0, 255).astype(np.uint8)

    else:

        raise ValueError("Effect type not recognized or not implemented.")

*Результат виконання функції в залежності від параметрів:*A collage of a person

Description automatically generated

**IV. Висновки.**

В ході цієї лабораторної роботи я дослідив особливості реалізації алгоритмів растрової обробки цифрових зображень на прикладі застосування алгоритмів растеризації. Я вивчав методи застосування різних ефектів та градієнтів до зображення за допомогою бібліотеки Python, Pillow, та NumPy. Я визначив чотири типи ефектів (яскравість, чорно-білий, негатив, та сепія) та три типи градієнтів (діагональний, від центру, до центру), реалізувавши їх через визначення класів та функції для застосування вказаних ефектів. Результати показали, як різні комбінації ефектів та градієнтів впливають на кінцеве зображення, надаючи інструменти для подальших досліджень в області обробки зображень.

Виконав: студент ФІОТ Лошак В.І. ІП-11