**Міністерство освіти і науки України**

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

**Кафедра обчислювальної техніки ФІОТ**

**ЗВІТ**

**з лабораторної роботи №8**

**з навчальної дисципліни «Computer Vision»**

**Тема:**

**ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ ТРИВИМІРНОЇ РЕКОНСТРУКЦІЇ**

**ОБ’ЄКТІВ ЗА ЦИФРОВИМИ ЗОБРАЖЕННЯМИ**

**Виконав:**

Студент 3 курсу кафедри ІПІ ФІОТ,

Навчальної групи ІП-11

Трикош І. В.

**Перевірив:**

Професор кафедри ОТ ФІОТ

Писарчук О.О.

**Київ 2024**

## **І. Мета:**

Дослідити методологію і технології реконструкції 3D просторових об’єктів за їх 2D зображеннями методами багатовидової (стерео / сигнатурна) обробки.

## **ІІ. Завдання:**

Розробити програмний скрипт мовою Python що реалізує тривимірну реконструкцію об’єктів за цифровими зображеннями:

Організувати та реалізувати роботу стереопари та отримати цифрове статичне зображення самостійно обраного об’єкту із двох каналів з різними значеннями кутового ракурсу. Або обрати із відкритих джерел результати роботи стереопари. Здійснити 3D реконструкцію обраного об’єкту та дослідити якість результату від параметрів стереопари: база, ракурс на об’єкт (за умов наявності стереопари, або відомих параметрів, що супроводжують відкриті джерела даних від стереопари).

Об’єкт, що підлягає реконструкції – бюст.

## **ІІІ. Результати виконання лабораторної роботи.**

* 1. **Синтезована математична модель**

Відповідно до умов задачі синтезовано математичну модель, що реалізує тривимірну реконструкцію об’єктів за стереопарою.

Для знаходження невідповідностей на зображення використаємо StereoSGMB (Semi-Global Block Matching) – алгоритм для знаходження відповідностей між зображеннями. Його основні параметри – мінімальна невідповідність, їх кількість, розмір блоку. З допомогою них можна покращити точність знаходження невідповідностей на зображення. Невідповідності знаходитимуться між лівим та правим зображеннями.

Перетворення невідповідностей у 3D координати відбувається з допомогою матриці Q. Вона враховує геометрію камер та їх фокусну відстань.

Перетворені з допомогою матриці Q формують хмару точок, яка відображається з допомогою засобів Open3D.

Таким чином, синтезована математична модель знаходить невідповідності на стереопарі, з допомогою матриці перетворення формує хмару точок та відображає їх як реконструйований 3D об’єкт.

* 1. **Результати архітектурного проектування та їх опис**

На рис. 1 зображена блок-схема до моделі до програмного скрипту:

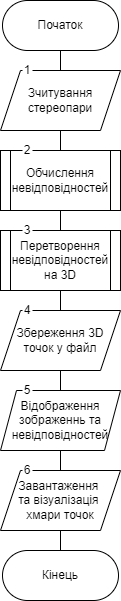


Рисунок 1 – Блок схема алгоритму

Робота алгоритму розпочинається із зчитування стереопари. Сюди входить зчитування лівого та правого зображень. Ці дії виконуються у блоці 1 блок-схеми алгоритму рис.1.

У блоці 2 виконується обчислення невідповідностей стереопари. Спочатку обираються параметри алгоритму StereoSGBM, а потім з допомогою цього алгоритму знаходяться невідповідності.

У блоці 3 виконується перетворення (з допомогою матриці Q) невідповідностей на точки у 3D просторі.

У блоці 4 виконується збереження 3D точок та їх кольорів у файл PLY.

У блоці 5 відбувається відображення зображень та їх невідповідностей.

У блоці 6 виконується завантаження хмари точок з файлу PLY та їх відображення з допомогою Open3D. На цьому виконання алгоритму завершене.

* 1. **Опис структури проекту програми**

Для реалізації розробленого алгоритму мовою програмування Python з використанням можливостей інтегрованого середовища PyCharm сформовано проєкт.

Проєкт базується на лінійній бізнес-логіці функціонального програмування та має таку структуру.

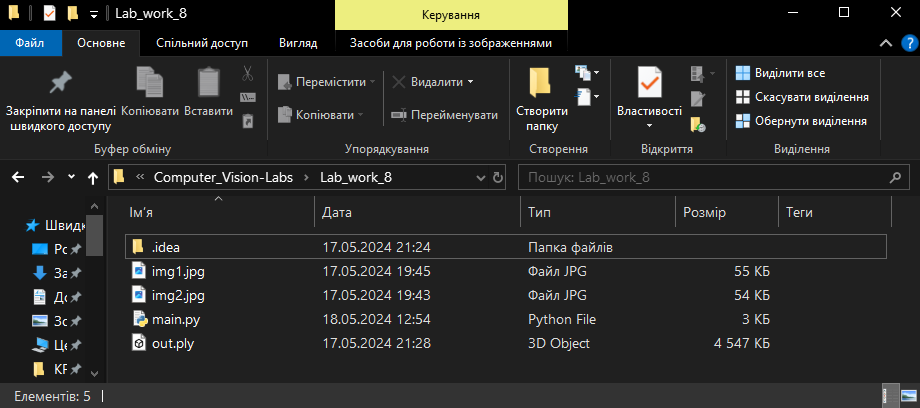


Рисунок 2 – Структура проєкту

Lab\_work\_8 – головний каталог проєкту

main.py – файл програмного коду

img1.jpg – ліве зображення стереопари

img2.jpg – праве зображення стереопари

out.ply – файл зі збереженими 3D точками

* 1. **Результати роботи програми відповідно до завдання (допускається у формі скриншотів)**

Результатом роботи програми є сукупність послідовності графічних вікон, що реалізують умови завдання лабораторної роботи.

1. Ліве зображення:



Рисунок 3 – Ліве зображення

1. Праве зображення:



Рисунок 4 – Праве зображення

1. Невідповідності стереопари:



Рисунок 5 – Невідповідності лівого та правого зображень

Бачимо, що алгоритм StereoSGBM знайшов немалу кількість невідповідностей, проте, нажаль, не всі. Тепер визначимо хмару 3D точок невідповідностей.

1. Хмара точок невідповідностей з різних ракурсів (рис. 6-8):

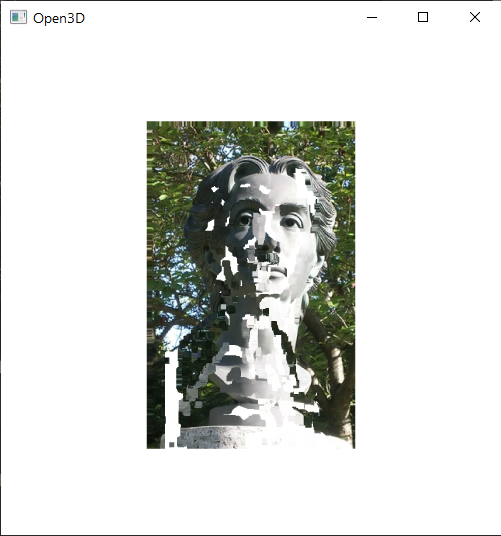


Рисунок 6 – Хмара точок невідповідностей

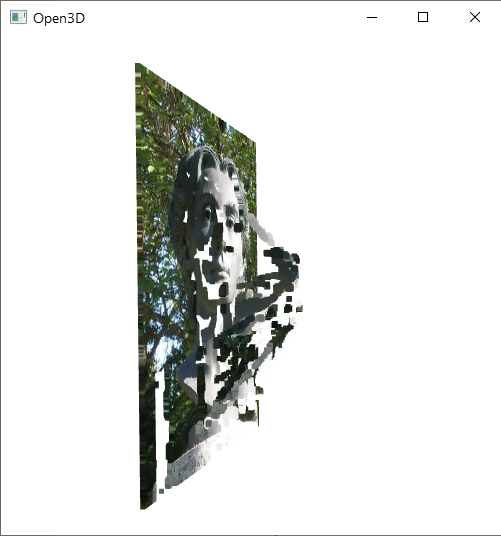


Рисунок 7 – Хмара точок невідповідностей

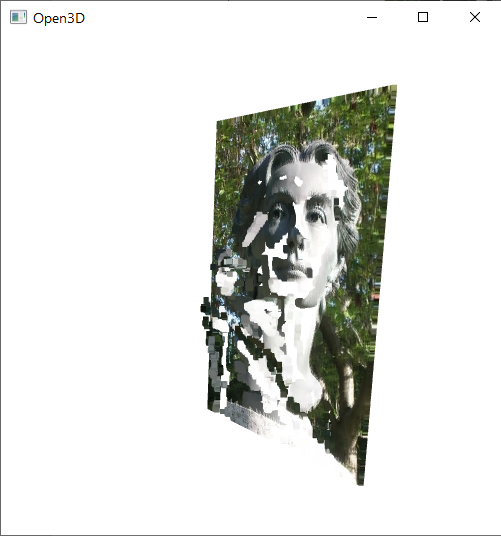


Рисунок 8 – Хмара точок невідповідностей

Бачимо, що реконструкція 3D об’єкта відбулася не зовсім вдало. Для покращення результату можна обрати інші параметри зйомки, наприклад кут зйомки. Також, можливо підібрати кращі параметри для алгоритму виявлення невідповідностей, або обрати інший алгоритм.

* 1. **Програмний код, що забезпечує отримання результату (допускається у формі скриншотів)**

Програмний код послідовно реалізує алгоритм на рис.1 та спрямований на отримання результатів, поданих на рис.3-8.

Для спрощення програмного коду і раціоналізації обчислень застосовано функціональні механізми створення підпрограм.

При цьому використано можливості Python бібліотек: numpy, OpenCV та Open3D.

Контекстні коментарі пояснюють сутність окремих скриптів наведеного коду програми.

main.py

*"""  
Реконструкція 3D просторових об'єктів за їх 2D зображеннями  
"""  
  
import* numpy *as* np  
*import* cv2  
*import* open3d *as* o3d  
  
*# Формат заголовка PLY файлу*ply\_header = '''ply  
format ascii 1.0  
element vertex %(vert\_num)d  
property float x  
property float y  
property float z  
property uchar red  
property uchar green  
property uchar blue  
end\_header  
'''  
  
  
*def* write\_ply(filename, vertices, vertices\_colors):  
 *""" Записування точок та їх кольорів у файл """* vertices = vertices.reshape(-1, 3)  
 vertices\_colors = vertices\_colors.reshape(-1, 3)  
 vertices = np.hstack([vertices, vertices\_colors])  
 *with* open(filename, 'wb') *as* file:  
 file.write((ply\_header % dict(vert\_num=len(vertices))).encode('utf-8'))  
 np.savetxt(file, vertices, fmt='%f %f %f %d %d %d ')  
  
  
*# Зчитування зображень*imgL = cv2.imread('img1.jpg')  
imgR = cv2.imread('img2.jpg')  
  
*# Параметри для алгоритму зіставлення стереоблоків*window\_size = 4  
min\_disp = 4  
num\_disp = 18 - min\_disp  
stereo = cv2.StereoSGBM\_create(minDisparity=min\_disp,  
 numDisparities=num\_disp,  
 blockSize=16,  
 P1=8 \* 3 \* window\_size\*\*2,  
 P2=32 \* 3 \* window\_size\*\*2,  
 disp12MaxDiff=0,  
 uniquenessRatio=10,  
 speckleWindowSize=100,  
 speckleRange=32)  
  
*# Обчислення невідповідностей*disparities = stereo.compute(imgL, imgR).astype(np.float32) / 16.0  
  
*# Розміри зображення*h, w = imgL.shape[:2]  
*# Фокусна відстань*f = 0.5 \* w  
*# Обчислення матриці для проєктування невідповідностей у 3D просторі*Q = np.float32([[1, 0, 0, -0.5\*w],  
 [0, -1, 0, 0.5\*h],  
 [0, 0, 0, -f],  
 [0, 0, 1, 0]])  
*# Проєктування невідповідностей на 3D точки*points = cv2.reprojectImageTo3D(disparities, Q)  
*# Отримання кольорів із зображення*colors = cv2.cvtColor(imgL, cv2.COLOR\_BGR2RGB)  
*# Записування точок у файл*write\_ply('out.ply', points, colors)  
  
*# Вивід зображень та невідповідностей*cv2.imshow('left', imgL)  
cv2.imshow('right', imgR)  
cv2.imshow('disparity', (disparities - min\_disp) / num\_disp)  
  
*# Завантаження хмари точок з файлу*pcd = o3d.io.read\_point\_cloud('out.ply')  
*# Візуалізація хмари точок*o3d.visualization.draw\_geometries([pcd], width=500, height=500, left=20, top=20)  
  
*# Закриття вікон*cv2.waitKey()  
cv2.destroyAllWindows()

## **IV. Висновки.**

У ході виконання лабораторної роботи було проведено дослідження методологію і технології реконструкції 3D просторових об’єктів за їх 2D зображеннями методами багатовидової (стерео / сигнатурна) обробки. Для цього було обрано стереопару, з допомогою алгоритму StereoSGBM визначено невідповідності на зображеннях та з допомогою матриці перетворення Q сформовано хмару точок, яка формує 3D об’єкт. В результаті виконання цього алгоритму вдалося частково відновити 3D об’єкт.

Виконав: студент Трикош І. В.