

**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
Панченко С.В.

Перевірив:

Професор кафедри ОТ ФІОТ
Писарчук О.О.

Київ 2024 р.

I. Мета:

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

II. Завдання.

Завдання I рівня складності – максимально 8 балів.

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).

III. Результати виконання лабораторної роботи.

3.1. Синтезована математична модель відповідно до завдання.

Відтворення тривимірної сцени використовуючи пару стереофотографій. Працюючи з двома фотографіями, зробленими стереоапаратурою, можна аналізувати співвідношення між точками на знімках, їх відповідність та проводити необхідні геометричні обчислення для відновлення об'ємної моделі. Кроки процесу:

1. Опрацювання зображень для отримання їх матричних представлень;
2. Створення карт глибини зі знімків;
3. Розробка об'ємного представлення сцени.

3.2. Блок схема алгоритму завдання та її опис.

Застосування синтезованих моделей здійснюється у порядку, що відображає суть алгоритму реалізації завдання лабораторної роботи.

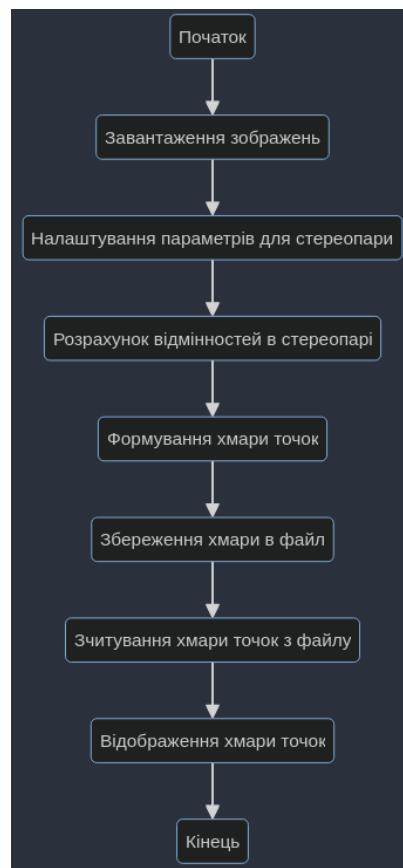


Рисунок 1 — схема алгоритму

Алгоритм запускається з процесу імпорту стереозображень, що виконується на першому етапі. Другий етап полягає у конфігурації налаштувань для пари стереозображень. Третій етап включає в себе обчислення дискрепансії між парою зображень, детальний опис якого представлено раніше. На четвертому етапі проводиться створення тривимірної моделі точок на основі отриманої дискрепансії. П'ятий етап займається архівуванням створеної хмари точок у файл. На шостому етапі система здійснює витяг збереженої інформації з файлу. І, нарешті, сьомий етап демонструє хмару точок в об'ємному форматі.

3.3. Опис структури проекту програми в середовищі PyCharm.

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

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

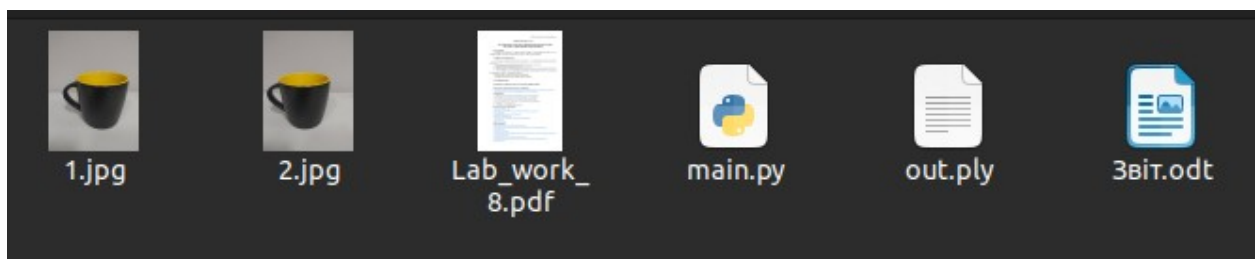


Рисунок 2 — схема проекту

1.jpg, 2.jpg — ракурси чашки
Lab_work_8.pdf — умова завдання
main.py — скрипт виконання
out.ply — хмара точок у файлі
Звіт.odt — звіт

3.4. Результати роботи програми відповідно до завдання.

Результатом роботи скрипта завдання є сукупність послідовності графічних вікон та файл, що містить хмару точок, що реалізують умови завдання лабораторної роботи.

Два зображення стереопари:



Рисунок 3 — ракурси чашки

Розглянемо мапу невідповідності:



Рисунок 4 — мапа невідповідності

Розглянемо хмару точок:

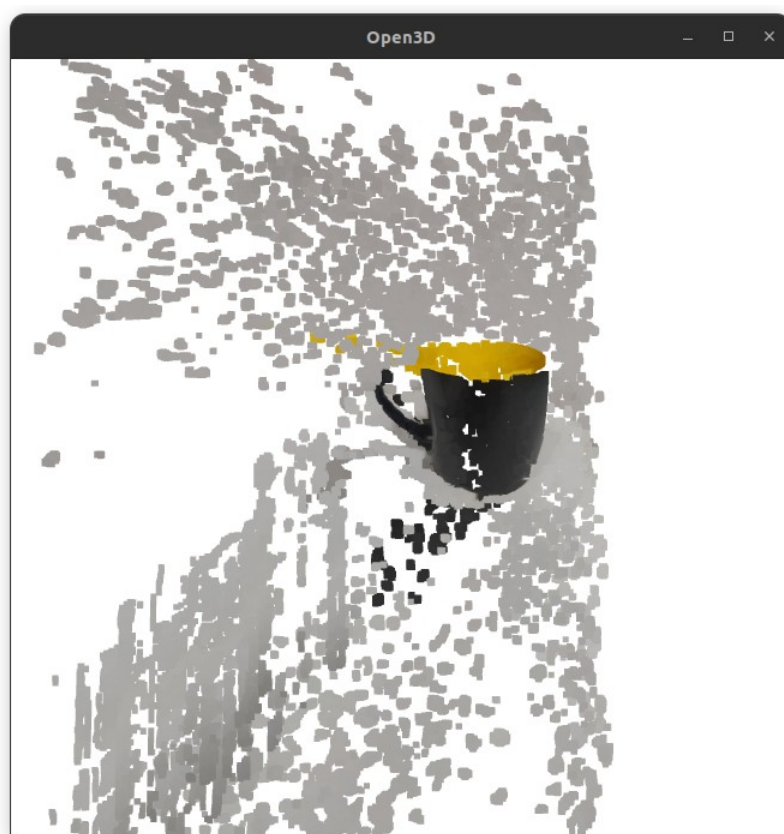


Рисунок 5 — Хмара точок

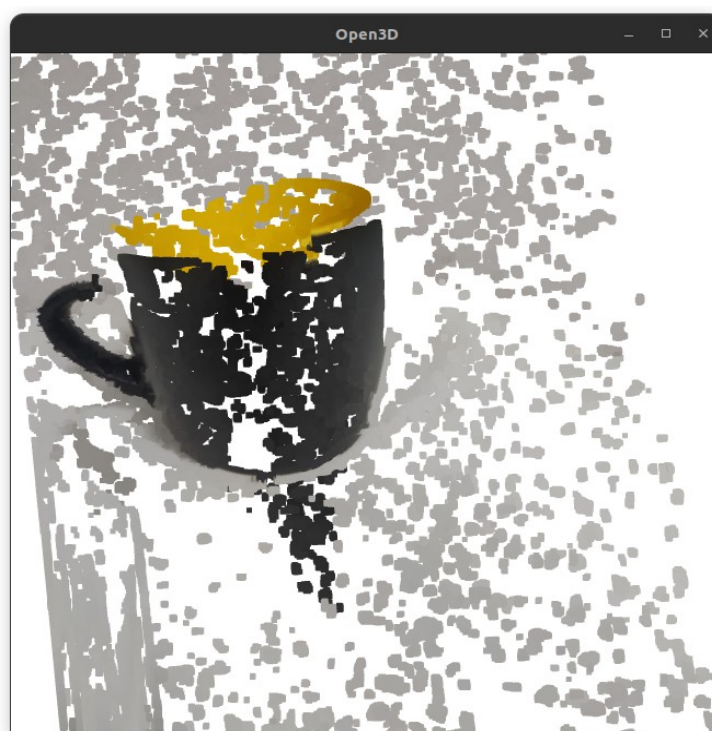


Рисунок 6 — Хмара точок

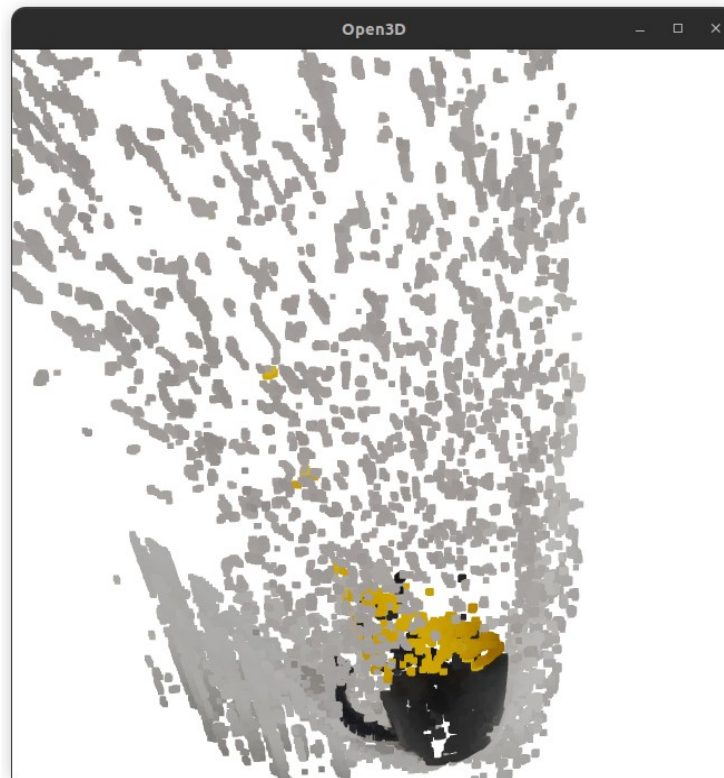


Рисунок 7 — Хмара точок

Бачимо, що чашка проілюстрована у 3D.

3.5. Програмний код.

Програма послідовно втілює процеси, викладені на схемі 1. Вона націлена на вироблення результатів, які були представлені на ілюстраціях 5-7. Для оптимізації програми та ефективності обрахунків було використано техніку розбиття коду на функції. У розробці програми застосовані можливості Python та таких бібліотек, як numpy, opencv та open3d.

main.py

```
import numpy as np
import cv2
import open3d as o3d

def write_ply(filename, vertices, colors):
    vertices = vertices.reshape(-1, 3)
    colors = colors.reshape(-1, 3)
    vertices_with_colors = np.hstack([vertices, colors])
    ply_header = f"ply\nformat ascii 1.0\nelement vertex {len(vertices_with_colors)}\n" \
        "property float x\nproperty float y\nproperty float z\n" \
        "property uchar red\nproperty uchar green\nproperty uchar blue\nend_header\n"
    with open(filename, 'wb') as file:
        file.write(ply_header.encode('utf-8'))
        np.savetxt(file, vertices_with_colors, fmt='%f %f %f %d %d %d')

def compute_3d_reconstruction(image_paths):
    img1, img2 = [cv2.imread(path) for path in image_paths[:2]]

    window_size = 3
    min_disparity = 15
```

```

num_disparities = 120 - min_disparity

stereo = cv2.StereoSGBM_create(
    minDisparity=min_disparity,
    numDisparities=num_disparities,
    blockSize=16,
    P1=8 * 3 * window_size**2,
    P2=32 * 3 * window_size**2,
    disp12MaxDiff=1,
    uniquenessRatio=10,
    speckleWindowSize=100,
    speckleRange=32
)

disparity = stereo.compute(img1, img2).astype(np.float32) / 16.0

height, width = img1.shape[:2]
focal_length = 0.8 * width
Q = np.float32([[1, 0, 0, -0.5 * width],
                [0, -1, 0, 0.5 * height],
                [0, 0, 0, -focal_length],
                [0, 0, 1, 0]])
points = cv2.reprojectImageTo3D(disparity, Q)
colors = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
mask = disparity > disparity.min()
out_points = points[mask]
out_colors = colors[mask]

output_file = 'out.ply'
write_ply(output_file, out_points, out_colors)
print(f"{output_file} saved")

cv2.imshow('Disparity', (disparity - min_disparity) / num_disparities)
cv2.waitKey()
cv2.destroyAllWindows()

print("Load a PLY point cloud, print it, and render it")
pcd = o3d.io.read_point_cloud(output_file)
o3d.visualization.draw_geometries([pcd], width=650, height=650, left=20, top=20)

if __name__ == '__main__':
    image_paths = ['1.jpg', '2.jpg']
    compute_3d_reconstruction(image_paths)

```

3.6. Аналіз результатів відлагодження та верифікації результатів роботи програми.

Результати від лагодження та тестування довели працездатність розробленого коду.

Верифікація функціоналу програмного коду, порівняння отриманих результатів з технічними умовами завдання на лабораторну роботу доводять, що усі завдання виконані у повному обсязі.

IV. Висновки.

Під час виконання лабораторного завдання було здійснено аналіз методів та інструментарію для створення тривимірних моделей об'єктів на основі їх двовимірних знімків, використовуючи багатокутковий (стерео / сигнатурний) підхід обробки даних на мові програмування Python. У рамках дослідження була відновлена тривимірна модель чашки із двовимірних зображень за допомогою стереообробки. Результатом стала точкова хмара, яка відтворює 3D форму об'єкта.

Виконав: студент Панченко С.В.