Sensing and Perception

Dhruv Pandit
K23118144
31 July 2024

I. Introduction

The given assignment contained two tasks - the 3D reconstruction of an object using computer vision and the implementation of a grasping algorithm that can be used to manipulate the reconstructed object.

The following steps are required for the 3D reconstruction of the object -

- Image Capture
- Feature Detection and Matching
- Camera Calibration and Pose Estimation
- Triangulation

The methods used to obtain the reconstruction are elaborated upon in section II

II. Methodology

A. Image Capture

Images were captured using a smartphone camera. First various images of the checkerboard pattern were taken so that they could be used for camera calibration. Images of the object used in the 3d reconstruction were taken using the same camera so that the camera parameters acquired from the camera calibration could be used for reconstruction. One of the original images can be seen in Fig. 1

Fig. 1. Original Image

B. Camera Calibration

Camera calibration is a fundamental step in computer vision and photogrammetry, aimed at determining the intrinsic and extrinsic parameters of a camera such as the focal lengths, principal points and rotation matrix. These parameters are essential for correcting image distortions and understanding the camera's position and orientation in the scene. It was carried out using OpenCV findChessboardCorners() and images of a chessboard pattern. [1]

C. 3D Reconstruction

First, SIFT (Scale-Invariant Feature Transform) was used to detect keypoints. [2].The results of which can be seen in 2 The keypoints were then matched using OpenCVs Brute Force matcher [3] along with K nearest neighbours and RANSAC to remove any outliers as they are simple and efficient while being robust. The results of Bruce Force matching can be seen in Fig.3. Finally the open3d library [4] was used to obtain a pointcloud and the dense reconstruction. It was then used to export the mesh into the STL format that could be imported into Gazebo.

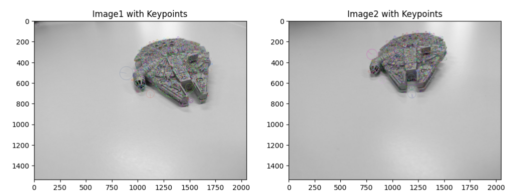

Fig. 2. SIFT Results

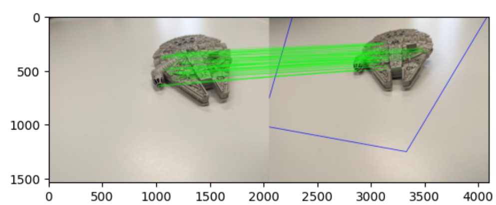

Fig. 3. Matches

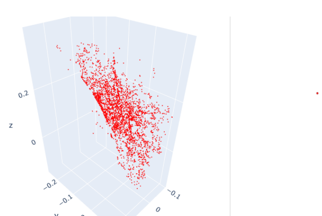

Fig. 4. Sparse Reconstruction

D. Grasping

ROS, MoveIt!, Gazebo and RViz were used for the grasping portion of this assessment witht he Panda Robot being used for execution. MoveIt has an inbuilt pick and place function that was used to carry out the grasping task. [5]

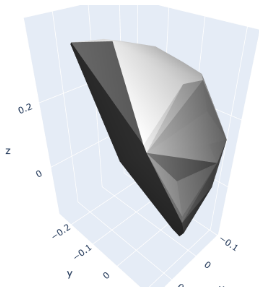

Fig. 5. Dense Reconstruction

III. Challenges Faced

Given that the images were captured using a smartphone camera, the images were distorted by the various filters and processes used by the smartphone's camera app. Results could be improved by capturing images in a different format such as RAW. Original images were captured in a cluttered environment and that were detected by SIFT. To prevent unwanted features from being detected, images were taken against a plain background.

Importing the STL into RVIZ for proper grasping also posed a challenge which is why a random object was used for the simulation.

References

[1] "OpenCV: Camera Calibration — docs.opencv.org." https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html. [Accessed 01-08-2024].

[2] "OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform) — docs.opencv.org." https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html. [Accessed 01-08-2024].

[3] "OpenCV: Feature Matching — docs.opencv.org." https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html. [Accessed 01-08-2024].

[4] "Getting started - Open3D 0.18.0 documentation — open3d.org." https://www.open3d.org/docs/release/getting_started.html. [Accessed 01-08-2024].

[5] "Pick and Place Tutorial &x2014; moveit_tutorials Kinetic documentation — docs.ros.org." http://docs.ros.org/en/kinetic/api/moveit_tutorials/html/doc/pick_place/pick_place_tutorial.html. [Accessed 01-08-2024].