Abstract

The project aims to use a camera to build a 3D model of an object so that a robotics arm can manipulate it. To create the 3D model, the structure from motion technique will be utilized. The robotics task simulation will be conducted using GAZEBO environment using ROS to control the motion of the robot. The result of SfM was good in the context of dimension recovery and camera position estimation but wasn’t good enough to build the faces and structure of the object.

Methodology

This chapter will discuss building a 3D model using the Structure from Motion (SfM) methodology. In this project, we take multiple photos of an object from different positions and angles, then track and transform the SIFT features between the first image and all other images to create a 3D model. This process includes several steps, such as capturing pictures and calibrating the camera, detecting matched features, estimating camera positions, and triangulation.

1.1 Photo Capturing and Camera Calibration (Intrinsic Matrix).

An initial step in all computer vision tasks is to find the camera matrix to estimate the relation between the dimensions in the photos and in the real world. In this project, the Samsung s23 ultra 200 MP camera was used to take the photos; it has a focal length of 23 mm and a sensor size of 1/3.52” [1]. The methodology for estimating the camera matrix in this project utilizes the OpenCV library, which detects chessboard corners to establish correspondences between 3D world coordinates and 2D image coordinates over seven images of (9 x 6) chessboard taken by the phone's camera from different angles. Initially, the algorithm processes chessboard images from a directory, converting each to grayscale and employing the “cv.findChessboardCorners” function for corner detection. Subsequently, corner accuracy is refined via “cv.cornerSubPix”, adhering to specified termination criteria(30 iterations or error less than 0.01). A predefined 3D object grid representing the chessboard's geometry facilitates mapping these 3D points to 2D image space. Collecting 3D-2D correspondences across images enables the “cv.calibrateCamera” function to compute the camera intrinsic matrix [2].

1.2 Detecting Matched Features.

For feature extraction and description, we use the Scale-Invariant Feature Transform (SIFT) "cv.sift.detectAndCompute", followed by the Brute Force (BF) matcher for feature matching between two images "bf.knnMatch(descriptors1, descriptors2, k=2)". SIFT identifies key points and descriptors, ensuring invariance to scale, translation and rotation and facilitating robust matching. The BF matcher employs a k-nearest neighbours’ approach with a ratio test, retaining matches with a distance less than 75% of the second nearest match to filter out less reliable correspondences. This process results in high-quality feature matches, which are required for Structure from Motion (SfM), where feature matching across images must be accurate to enable 3D reconstruction from 2D views [2].

1.3 Camera Position Estimation

To determine the positions of the camera, we first need to calculate the transformation matrix between the camera positions of first and target photos. This process consists of 3 steps: estimating the Fundamental matrix, the Essential matrix, and the use of them to estimate the transformation matrix.

To estimate the Fundamental matrix, for each pair of points that match in the first and second images, a row is added to a matrix (A) based on their coordinates (x, y) and (x', y') respectively. The structure of each row in matrix A is designed to encapsulate the relationship between the point pairs and the elements of the Fundamental Matrix, adhering to the epipolar constraint $x'^T F\, x = 0$ [3].

Next, a Singular Value Decomposition (SVD) is performed on matrix A to decompose it into three matrices: U, S, and V. The singular vectors that correspond to the columns of matrix A are contained in matrix V. The last row of matrix V is then taken as the estimated flat vector of the Fundamental Matrix (F), which is then reshaped into a 3x3 matrix [3].

Since the Fundamental Matrix should have a rank of 2 due to its inherent properties, another SVD is performed on F to enforce this property. The smallest singular value in the diagonal matrix S is set to zero, reducing the rank of the Fundamental Matrix to 2, ensuring it adheres to its theoretical constraints. Finally, the modified singular values are used along with the original U and V matrices to reconstruct the Fundamental Matrix (F) that best fits the epipolar constraint for the given set of corresponding points [3].

Two methods were used to estimate the Essential matrix. The first method involved using the OpenCV library function "cv.findEssentialMat" with the RANSAC algorithm. This method takes the normalized matched points as input and produces the Essential matrix as output. The second method involved using the fundamental matrix in equation (1).

$$E = camera\_matrix^T \times F \times camera\_matrix \tag{1}$$

Two methods were employed to determine the change in camera position and orientation between consecutive shots. The first method involved using the OpenCV function "cv.recoverPose." This function yields the rotation matrix and translation vector for a given two matched point sets, the essential and camera intrinsic matrices. The second technique involved decomposing the essential matrix into three matrices: U, S, and V, using singular value decomposition (SVD). The rotation matrix and translation vector were then calculated using equations (2) and (3) [4].

$$T_{Vector} = -\mathrm{U}(:,3) \tag{2}$$

$$R_{Matrix} = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} V^T \tag{3}$$

1.4 Triangulation

In this project, we use a linear triangulation method to rebuild 3D points from their corresponding 2D projections across 2 images. The triangulation methodology begins by constructing camera projection matrices $P1$ and $P2$ for the first and second camera views using the intrinsic camera matrix, transformation matrix (R, T) as shown in equation (3).

$$P = IntrinsicMatrix \times [R \quad T] \tag{3}$$

Next, we create homogeneous coordinates for the corresponding points in both images. These coordinates help to create a constraints matrix A for each point pair. The matrix is based on the epipolar geometry constraint. SVD is then used to solve for the homogeneous 3D coordinates, which are normalized. This process is repeated for all point pairs, resulting in the accumulation of reconstructed 3D points. However, nonlinear triangulation is needed to transform all 3D points to the same world frame [4].

Main Results

This chapter presents the key findings of the project, which include feature detection and matching, camera positioning, and generation of a 3D model.

2.1 Feature Detection and Matching

Despite having many objects in the images' background and human movements in the environment, the SIFT detector and BFMatcher managed to detect a good feature and match them efficiently. As shown in Figure 1, most matched features are on the target object, which significantly improves the 3D structure.

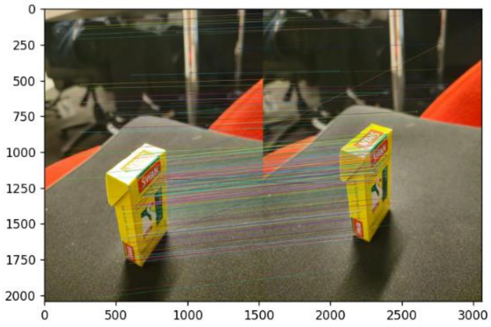

Figure 1 Matched Features

2.2 Camera Positioning

We experimented to test the estimated camera position, fundamental matrix, and essential matrix. We took two photos in the experiment and measured the distance between the first and second camera positions. We then estimated the camera position using three different methods. Figure 2 displays the output of the first two approaches: the first approach used equations (2) and (3) with the essential matrix estimated using the OpenCV function. In contrast, the second approach used the "cv.recoverPose" function. The third method is the same as the first, but it uses the essential matrix driven from the fundamental matrix using equation (1). Although the first two methods provided perfect position estimation, the third method provided poor estimation. Therefore, we will use the first method to find the 3D construction of the model, which provides the camera path shown in Figure 4 for the 13 object photos used in SfM.

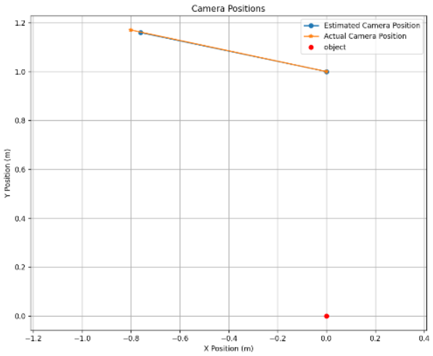

Figure 2 Calculated position using estimated Essential matrix using OpenCV.

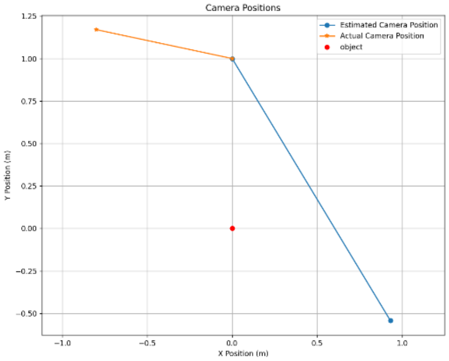

Figure 3 Calculate position using calculated Essential matrix from Fundamental matrix.

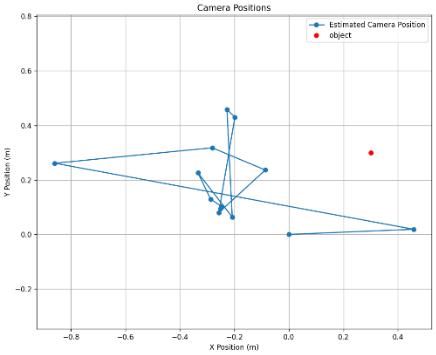

Figure 4 Camera position for taking the SfM photos.

2.3 Generation of the 3D Model

Our project aimed to reconstruct the 3D model of an object with a rectangular shape, with dimensions of 0.08m x 0.035m x 0.015m, from its 2D images. Isometric and the three standard orthogonal views of the reconstructed 3D points are shown in Figures 5 to 8. In these figures, each color corresponds to a specific pair of images used in the reconstruction. A noticeable separation between points of different colors can be observed, indicating that the points do not merge into a unified structure. This is back to relying only on linear triangulation methods and ignoring nonlinear triangulation techniques, which are critical for achieving more integrated 3D structures. To evaluate the reconstructed model, we used MeshLab as shown in Figures 9 and 10. The visualization revealed that the dimensions of the model closely approximate those of the actual object, measuring around 0.087m x 0.045m.

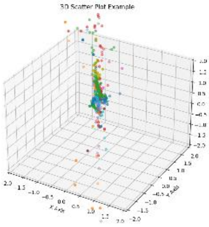

Figure 5

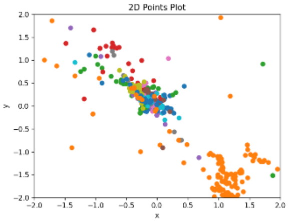

Figure 6

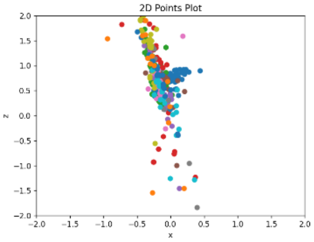

Figure 7

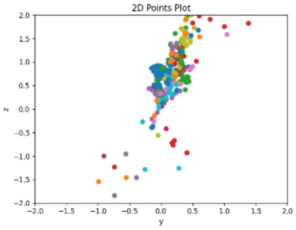

Figure 8

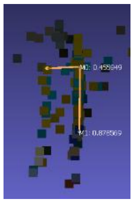

Figure 9

Figure 10

Although the reconstructed model is similar in size to the actual object, it struggles to create precise walls and edges. This difficulty arises because SIFT features are proficient at recognizing and pairing unique points across images, but they may not grasp the complete structural intricacies of objects.

Conclusion

For this project, we used a camera to create a 3D model of an object that a robotic arm could manipulate. We utilized the structure from the motion technique to build the 3D model, resulting in satisfactory dimension recovery and camera position estimation.

References

[1] P. Arena, "Samsung Galaxy S23 Ultra Specs," 17 2 2023. [Online]. Available: https://www.phonearena.com/phones/Samsung-Galaxy-S23-Ultra_id12002. [Accessed 24 2 2024].

[2] "Library., OpenCV. (2015). Open Source Computer Vision," [Online].

[3] U. o. Oxford, "Epipolar Geometry and the Fundamental Matrix," [Online]. Available: https://www.robots.ox.ac.uk/~vgg/hzbook/hzbook2/HZepipolar.pdf. [Accessed 25 2 2024].

[4] n.d, "Structure from Motion," CMSC426 Computer Vision, [Online]. Available: https://cmsc426.github.io/sfm/. [Accessed 17 2 2024].