3DCV Project WS 2023/2024

Alejandro Campayo Fernández
alca00005
7058536

Sai Suresh Macharla
sama00014
7047009

Aishwarya Kshirsagar
aiks00001
7056998

Abstract

The goal of this project was to create a 3D object from noisy 2D images of it. It was divided into two parts: First, generating a clear set of correspondences from two images containing the object. Second, using this clear set of correspondences to generate a point cloud.

1 Extracting and Filtering Correspondences

1.1 Image preprocessing

In an attempt to discard points within the desired object, we explored AI image segmentation models. Two promising models, namely SegGPT (Wang et al., 2023) and Dichotomous Image Segmentation (DIS) (Qin et al., 2022), were considered. However, SegGPT was dismissed due to its requirement for GPU usage.

We tested DIS on our images and found promising results. For this reason, we decided to make use of it and adapted its output for our task.

Figure 1: Best results for boot segmentation

However, the results varied significantly depending on the image being processed. We present two images illustrating the poorest outcomes obtained for the boot. One depicts incomplete removal of the background, while the other, representing the worst-case scenario, mistakenly removes parts of the boot from which we aimed to extract features.

Figure 2: Bad background segmentation

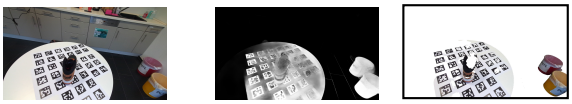

Figure 3: Partially erased object durinng segmentation

While DIS segmentation could enhance our object reconstruction by removing unwanted background noise, we realized that the background contained valuable information for our camera pose estimation. Consequently, correspondences were extracted from both the original and segmented images.

1.2 Keypoint extraction and feature matching

For feature detection, we utilized SIFT and ORB from the OpenCV library. Matching was performed using L2 distance (for SIFT) and Hamming distance (for ORB), along with Lowe's ratio test and ensuring backward and forward consistency. We experimented with various hyperparameters and ultimately selected SIFT with a Lowe ratio of 0.7 and a minimum distance of 150 for raw images. For segmented images, we adjusted the minimum distance to 100. The reasoning behind using different hyperparameters for segmented images is that, due to image processing, we can afford to relax the tests slightly to obtain more keypoints in the object.

Figure 4: 2000 ORB keypoints in raw and masked bot

Figure 5: 2000 SIFT keypoints in raw and masked bot

1.3 Refining and filtering correspondances with Optical Flow asumptions

After matching the keypoints and visualizing them we realised that some matches were obviously incorrect. To understand the matches, we visualized them as an optical flow of features, it was then that we realised that given the nature of the images (images taken around the object with small movements), we could use some of the assumptions used in optical flow problems in order to filter incorrect matches. In optical flow, methods such as Horn-Schunck take some assumptions regarding the smoothness of the optical flow. With the objective of imitating that idea, we chose to compare each match vector to its 3 closest neighbours with cosine similarity and if more than 1 of the match vectors in the neighbourhood had a cosine similarity smaller than a given threshold, we would discard that pair of keypoints. Moreover, we removed the matches whose vector was bigger than a bigger threshold, the idea behind it was that big vectors indicate

2 Structure-from-Motion from Known Correspondences

After obtaining the known correspondences, we estimate the Fundamental matrices of all the correspondences To remove all the outliers, we perform RANSAC algorithm and get all the inliers. With the inlier correspondences, we proceed with estimating the Fundamental matrices of all the correspondences. We estimate using the 8-point algorithm and then SVD decomposition.

To estimate the Essential matrices, we get the intrinsics provided for each object. Both Fundamental and Essential matrices have rank 2 deficiency so we set the last singular values to 0 for both. With the Essential matrix, we can now estimate the camera poses i.e the Rotation and Translation matrices. $C_1 = U(:,3) and \quad R_1 = UWV^T$
$C_2 = -U(:,3) and \quad R_2 = UWV^T$
$C_3 = U(:,3) and \quad R_3 = UWV^TV^T$
$C_4 = -U(:,3) and \quad R_4 = UWV^TV^T$

We obtain 4 camera poses in total. The correct camera pose can be found using Cheirality condition using Triangulation. The linear triangulation minimizes the algebraic error. The geometric error can be found by computing non Linear Triangulation or the reprojection errorusing this formula -

$\min_{\mathbf{x}} \sum_{j=1}^{2} \left(u_j - \mathbf{P}_j^T \mathbf{1} \phi \widetilde{\mathbf{P}}_j^T \mathbf{3x}\right)^2 + \left(v_j - \mathbf{P}_j^T \mathbf{2} \phi \widetilde{\mathbf{P}}_j^T \mathbf{3x}\right)^2$

We used the least squares solver from Scipy to optimize the 3D correspondences. Next, to optimise the camera poses, we solve the Perspective-n-projection problem. The linear PNP problem can be solved with SVD. There can be many outliers after solving the linear PNP as pnp requires only 6 points so we filter out the inliers using RANSAC. To optimise R and C, we minimize the reprojection error. Here, we have used least squares. We proceed to perform bundle adjustment, in order to further optimize all estimated camera pose parameters and the 3D points together to achieve 3D reconstruction until that particular camera under registration. To perform Bundle Adjustment for a given set of poses, 3D points (parameters under optimization) and corresponding 2D points, we need to create a visibility Matrix that records whether a 2D point observation belongs to the particular parameter. For visualization purpose of the point clouds, we have use the Open3d visualization tool.

3 Conclusions

In conclusion, our project aimed to reconstruct a 3D object from noisy 2D images, addressing challenges in correspondence extraction, filtering, and ultimately, structure-from-motion reconstruction. We began by exploring AI image segmentation models such as SegGPT and DIS. Our approach involved linear and non-linear triangulation and pnp techniques, alongside bundle adjustment to optimize camera poses and 3D points jointly.

References

Project 5: Cameras. https://cs.brown.edu/courses/csci1430/2021_Spring/proj5_cameras/. Accessed: January 2024.

Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. 2022. Highly accurate dichotomous image segmentation. In *ECCV*.

University of Maryland. 2021. CMSC 426: Computer vision - structure from motion. https://cmsc426.github.io/sfm/.

Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. 2023. Seggpt: Segmenting everything in context.