NORTHEASTERN UNIVERSITY

Term Project: 3D Structure from Motion

GROUP 11- Members:

Avish Kalpesh Patel, Ashutosh Ravindra Iwale, Jiayuan Huang

Course: EECE 5554 – Robot Sensing and Navigation

Professor: Thomas Consi

4/28/23

Table of Contents

TABLE OF FIGURES

INTRODUCTION

SFM (Structure from Motion) refers to a Computer Vision method that reconstructs a 3D structure of a scene/object using a sequence of 2D images. The technique involves estimating the camera pose and the 3D location of key points or features in an image, that are integrated to create a 3D point cloud (Structure) and camera trajectory (Motion). There are various implementations of SFM depending on the number and type of camera (sensor) and ordering of input images. This project discusses and implements the method of incremental SFM that can work with inputs from different cameras (mobile phones, UAS cameras. etc.), random ordering of images, and images taken during the vivid environmental conditions of the same object (for instance, pictures taken by tourists of the monument, throughout the year and at a different time of the day).

APPLICATIONS

SFM (Structure from Motion) is a versatile technique with extensive applications in different fields like 3D reconstruction of buildings, landscapes, and archaeological sites aiding in architecture, construction, urban planning, and cultural heritage preservation sectors.[5] Moreover, in Robotics for autonomous navigation and localization and mapping using visual sensors (Visual SLAM) SFM can be used to create a 3D map of the environment and estimate the robot's position and orientation within that map. Such applications are useful in industrial automation, service, and mobile robots.[6] Within the space of Augmented Reality for applications to create realistic virtual objects and overlay them onto the real world, SFM allows registering virtual objects within the real world by estimating the camera pose and creating a 3D environment model that can benefit the entertainment, education, and training industries.[7] Furthermore, Medical Imaging applications include creating 3D models of organs or tissues from a sequence of medical images beneficial for surgery planning, medical education, and research.[8] Likewise, in the Agriculture sector, SFM can be used for precision crop monitoring, yield estimation, and mapping. It allows creating 3D models of crops for estimating their health and growth.[9]

OBJECTIVE

The objective of this project is to gain a deeper understanding of computer vision concepts, with a specific focus on the Structure from Motion (SFM) technique. Through this project, we aim to learn the underlying principles, algorithms, and tools involved in SFM and how it can be used to reconstruct 3D structures from 2D images. By working on this project, we implement SFM algorithms using various software libraries and frameworks. Overall, this project will serve as a hands-on learning experience that will enable us to expand our knowledge and proficiency in computer vision, particularly in SFM.

ALGORITHM

The algorithm [1] used for processing the images in this project can be seen as below:

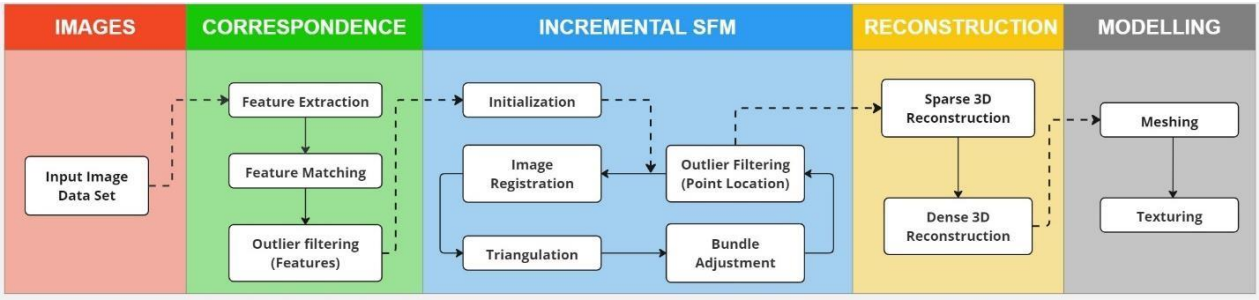

Figure 1: Image Processing Flowchart

Correspondence

Establishing correspondence between images requires features/key points that appear in them. Firstly, all images are processed to identify and characterize local image structures (features). The proposed algorithm extracts SIFT features (Scale Invariant Feature Transforms) as they are invariant to scale, orientation, and illumination changes.[2] Figure 2 shows the efficiency of SIFT features that accurately match between images despite the effect of sunlight.

Following, feature objects are matched in subsequent images if they correspond between them. For matching, a template (section) of the original image, with the feature location (pixel coordinates) as the center, is checked for correlation with the templates of other features in subsequent images.

Feature Matching is error-prone; hence, we need to filter out the false matches (outliers) from the inliers (seen in Figure 3).

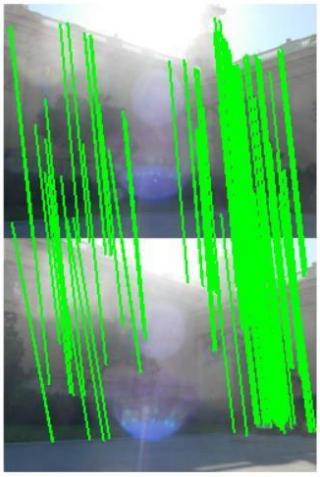

Figure 2:SIFT feature efficiency

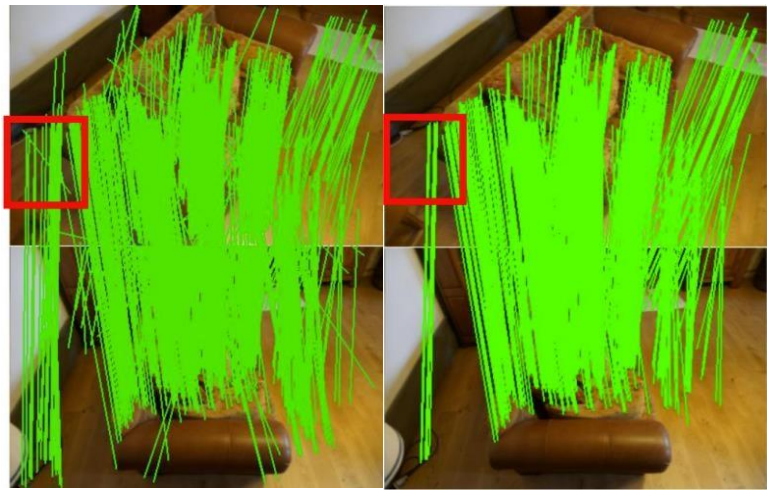

Figure 3: Before (left) and after (right) outlier filtering of feature matching

Incremental SFM

Initialization

A single image can provide planar information of the point (pinhole camera model). However, for a depth-estimation minimum of two views are needed to implement stereo vision principles. Thus, initialization requires two views (seed images) with the largest feature matching (previous step). Following the 3D location of the point can be estimated in

the camera frame using triangulation and the camera matrices (intrinsic and extrinsic parameters) for the seed images. Finally, the point location is transformed into the world frame using the camera's extrinsic parameters.[3]

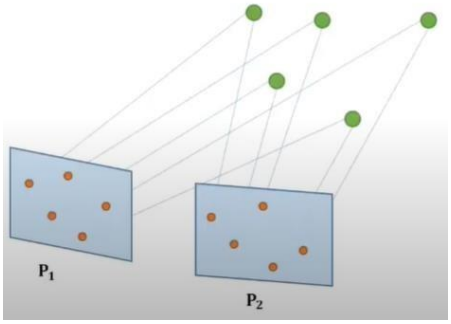

Figure 4: Two seed images to find the position of 3D points w.r.t to world frame.

Subsequent views

Following, subsequent image data is incorporated incrementally as follows:[3]

- 2D-3D correspondence: 3D locations of the points appearing in seed frames are established using triangulation.
- 2D-2D correspondence: Add P_3 frame with largest feature matching for the P_2 frame.
- PnP (Perspective-n-Point): with the use of this algorithm estimate the pose of the 3rd camera using the 3D location of the point and their 2D perspective projection seen in the P_3 image.

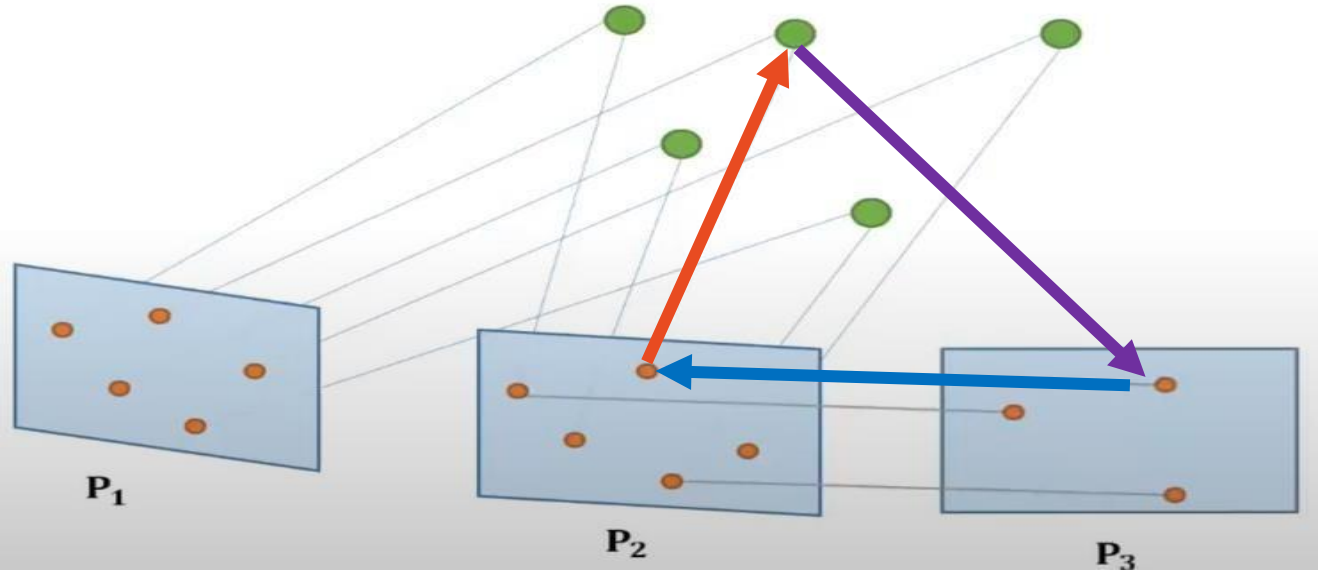

Figure 5: Incorporating subsequent data for Incremental SFM

Bundle Adjustment

Bundle Adjustment (BA) is a critical step in the SFM pipeline that refines camera poses and 3D structure by minimizing reprojection error. It is typically used after earlier SFM stages have obtained initial estimates of camera poses and 3D structure. The Bundle Adjustment reprojects the 3D points back to each camera poses' 2D frame and calculates the

reprojection error, it then uses non-linear least squares to adjust the camera poses and the 3D points simultaneously to create the least reprojection error as seen in Figure 6 below.

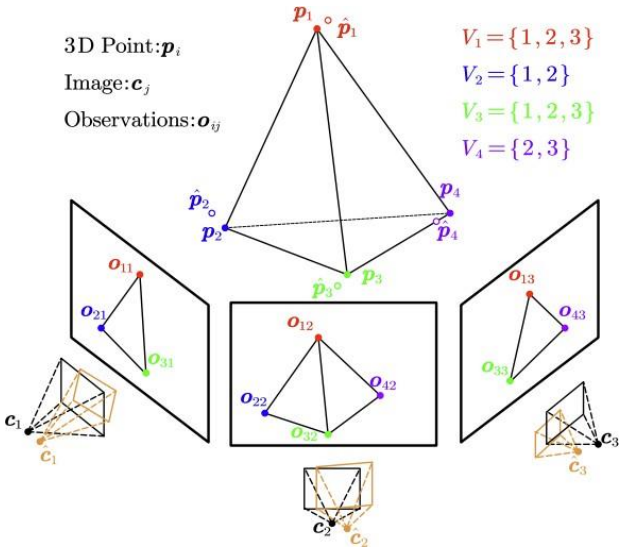

Figure 6:Representation of Bundle Adjustment with reprojection error

There are two types of Bundle Adjustment: Partial and Full. Partial Bundle Adjustment (PBA) optimizes only a subset of camera poses and 3D structure, while Full Bundle Adjustment (FBA) optimizes all parameters simultaneously, providing better accuracy but being computationally more expensive.[4]

The proposed algorithm incorporates Bundle Adjustment as a fundamental step for minimizing reprojection error. Specifically, we use PBA and FBA to comprehensively adjust the resulting point clouds and minimize reprojection errors. The use of PBA and FBA ensures that our algorithm produces accurate and consistent 3D reconstructions. The success of the BA step depends on several factors, including the quality of camera calibration, feature matching, and image measurements, as well as the complexity of the scene.

Reconstruction

The algorithm so far generates a sparse dense point cloud using VisualSFM toolbox.[15]

Sparse Reconstruction

Having a large set of features that correspond to many images can increase the computation cost while estimating subsequent camera poses. Hence, only the strongest features are matched over the image sequence. However, such a reconstruction struggles to portrait visible 3D structure (seen in Figure 7 for Sofa data set.)

Dense Reconstruction

To compensate for the lack of visibility of sparse reconstruction, a dense reconstruction of point cloud is generated. Once the significant features are reconstructed depicting a sparse structure, Multi View Stereo (MVS) is used for creating denser point clouds using previously computed camera poses and the provided image data (seen in Figure 7 for Sofa data set.). We can get a dense reconstruction using toolbox CMVS. [10]

Figure 7: Sparse Reconstruction (left) and Dense Reconstruction (right) of Sofa image data.

Modelling

Meshing is applied to create a surface comprised of interconnected polygons. Following the color/texture information from images is mapped onto meshed surface to finally create a 3D model. For modelling we use MeshLab to process the dense point cloud output from our algorithm.[11]

Figure 8:Meshed Surfaces (left) and Textured model (right)

RESULTS AND ANALYSIS

The Sofa Data Set consists of images captured by us using a mobile phone camera. The buddha and tree data set images are obtained from [14] and [13]

For Analysis purposes, we compared the distance between 2 dense point clouds for each Image data set, one generated using our algorithm discussed above and the other created using a photogrammetry software (MESHROOM) serving as the ground truth.[12] The Appendix section includes the results of the dense point clouds, meshes, and textured models using both methods.

Moreover, it is difficult to analyze the distance error between 2-point clouds as both methods provide different number of points for dense reconstruction. For instance, for Buddha image data set, our algorithm generated 787,045 points while the MESHROOM generated a Point Cloud consisting of 655,861 points. Hence, for calculating the error we make use of MESHLAB application and do the analysis as follows:

1. Refine point clouds (i.e., deleted unwanted points or surfaces)
2. Normalize point clouds with respect to coordinate system (i.e., the bounding box of each point cloud has a unit length in the largest dimension (height, width, or length))

3. Shift the point clouds centers to the origin of coordinate system.
4. Align the points clouds with respect to each other.

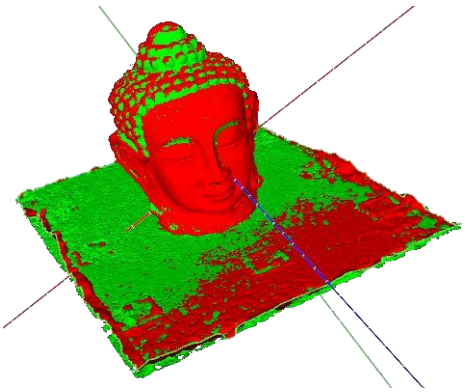

Figure 9: Scaling and Aligning the point clouds. Output of MESHROOM serving as Ground Truth (Green) and Our Output (Red)

After alignment, there is still a difference in the number of points in both clouds. Hence, we find the error by comparing the distance of a point (our output) that is closest to a point (ground truth).

The Error is given by:

$$Euclidean(P_i, \hat{P}_i) = \sqrt{(X_i - \hat{X}_i)^2 + (Y_i - \hat{Y}_i)^2 + (Z_i - \hat{Z}_i)^2}$$

$$Mean\ Error = \frac{1}{N}\sum_{i=1}^{N} Euclidean(P_i, \hat{P}_i)$$

$$Root\ Mean\ Squared\ Error = \sqrt{\frac{1}{N}\sum_{i=1}^{N}[Euclidean(P_i, \hat{P}_i)]^2}$$

$$\hat{P}_i = True\ Location\ of\ Point\ (Ground\ Truth)$$

$$P_i = Location\ of\ Point\ (Our\ Result)$$

$$N = Number\ of\ points\ (Our\ Result)$$

Table 1: Error analysis of the point clouds generated using our algorithm in comparison to point clouds using MESHROOM.

Data Set	Mean Error (units)	Root Mean Squared Error (units)	Maximum (units)
Buddha	0.000315	0.003127	0.051469
Sofa	0.001320	0.015927	0.181690
Tree	0.001166	0.006589	0.126989

NOTE: Units

As discussed earlier the point clouds are scaled such that its largest dimension (height, width, or length) is of unit (equal to 1) measure in the MESHLAB's coordinate frame. Hence, the errors are normalized with respect to the largest dimension.

However, it is important to note that the point cloud we assume to be ground truth is not the reality but an output of different SFM algorithm. From the table above we can see that for all the data set, the largest error (Sofa Data set) has mean error < 0.132 % and RMSE < 1.6% of its largest dimension. Hence, we can say the performance of our algorithm is quite accurate in reference to conventional photogrammetry tools (such as MESHROOM).

CONCLUSION

The implementation of Structure from Motion (SFM) that utilizes feature extraction and matching, PnP algorithm, and Bundle Adjustment has demonstrated to be a potent technique for generating three-dimensional (3D) scenes from two-dimensional (2D) images. By extracting and matching features among images, we initialized the camera poses, which resulted in a sparse point cloud. The use of Bundle Adjustment allowed us to refine the camera poses and 3D points, producing a more precise reconstruction. Furthermore, the incorporation of dense reconstruction techniques allowed us to interpolate the missing information and obtain a detailed and accurate 3D model. Overall, the SFM pipeline employed in this study has yielded a valuable instrument for 3D scene reconstruction that has the potential to be applied in diverse fields, including robotics, augmented reality, computer vision, and medicine, among others.

FUTURE WORKS

There are several avenues that can be pursued to enhance the accuracy and efficiency of the 3D reconstruction software as delineated above. Specifically, the precision and robustness of the reconstruction may be advanced through improvements in feature extraction, matching, and Bundle Adjustment optimization. These enhancements could involve the incorporation of more refined feature extraction algorithms or refinement of the Bundle Adjustment algorithm parameters. Additionally, the computational efficiency of the software may be improved by implementing algorithmic optimizations or leveraging parallel computing. By pursuing ongoing learning and experimentation with novel techniques and approaches, the software may be continually updated and customized to address the diverse needs of distinct users and applications.

REFERANCES

[1] C. Wu, “Towards Linear-time Incremental Structure from Motion.” Available: http://ccwu.me/vsfm/vsfm.pdf

[2] “OpenCV: Introduction to SIFT (Scale-Invariant Feature Transform),” *Opencv.org*, 2020. https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html

[3] G. H. Lee, “3D Computer Vision | Lecture 10 (Part 2): Structure-from-Motion (SfM) and bundle adjustment,” *www.youtube.com*, Mar. 31, 2021. https://www.youtube.com/watch?v=j3VSmUCKZDM (accessed Apr. 28, 2023).

[4] C. Wu, S. Agarwal, B. Curless, and S. Seitz, “Multicore Bundle Adjustment.” Available: http://grail.cs.washington.edu/projects/mcba/pba.pdf

[5] M. J. León-Bonillo, J. C. Mejías-García, R. Martínez-Álvarez, A. M. Pérez-Romero, C. León-Ortíz, and C. Marín-Buzón, “SfM Photogrammetric Techniques Applied in the Building Archaeology Works of the Old Cloister of the Monastery of San Francisco from the 16th Century (Cazalla de la Sierra, Seville),” *Heritage*, vol. 5, no. 4, pp. 3901–3922, Dec. 2022, doi: https://doi.org/10.3390/heritage5040201.

[6] D. S. Smith and H. E. Sevil, “Design of a Rapid Structure from Motion (SfM) Based 3D Reconstruction Framework Using a Team of Autonomous Small Unmanned Aerial Systems (sUAS),” *Robotics*, vol. 11, no. 5, p. 89, Sep. 2022, doi: https://doi.org/10.3390/robotics11050089.

[7] M.-D. Yang, C.-F. Chao, K.-S. Huang, L.-Y. Lu, and Y.-P. Chen, “Image-based 3D scene reconstruction and exploration in augmented reality,” *Automation in Construction*, vol. 33, pp. 48–60, Aug. 2013, doi: https://doi.org/10.1016/j.autcon.2012.09.017.

[8] D. Um and S. Lee, “Microscopic Structure from Motion (SfM) for Microscale 3D Surface Reconstruction,” *Sensors*, vol. 20, no. 19, p. 5599, Sep. 2020, doi: https://doi.org/10.3390/s20195599.

[9] A. Ehrhardt, D. Deumlich, and H. H. Gerke, “Soil Surface Micro-Topography by Structure-from-Motion Photogrammetry for Monitoring Density and Erosion Dynamics,” *Frontiers in Environmental Science*, vol. 9, Jan. 2022, doi: https://doi.org/10.3389/fenvs.2021.737702.

[10] P. Moulon and A. Leroy, “|-- Open Source contribution --|,” *francemapping.free.fr*. http://francemapping.free.fr/Portfolio/Prog3D/CMVS.html (accessed Apr. 29, 2023).

[11] “MeshLab,” *www.meshlab.net*. https://www.meshlab.net/

[12]“AliceVision | Photogrammetric Computer Vision Framework,” *alicevision.org*. https://alicevision.org/

[13] “dataset_monstree,” *GitHub*, Apr. 23, 2023. https://github.com/alicevision/dataset_monstree (accessed Apr. 29, 2023).

[14] “Buddha dataset,” *GitHub*, Apr. 11, 2023. https://github.com/alicevision/dataset_buddha (accessed Apr. 29, 2023).
[15] C. C. Wu, “VisualSFM : A Visual Structure from Motion System,” *ccwu.me*. http://ccwu.me/vsfm/index.html

APPENDIX

The appendix section displaying the results is created as a separate file due to image compatibility that leads to corruption of the word file containing the report.