3D-Structure-Reconstruction

3DRCSFM

3D Reconstruction using Structure from Motion(SFM)

Bhagesh Gaur

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY **DELHI**

Motivation

- 3D scene representation is a challenging and ongoing research area in computer vision, with numerous techniques proposed, from traditional image processing to advanced methods like deep neural networks and transformer models.

- This work employs traditional image processing techniques and Incremental Structure from Motion (SfM) algorithm for 3D reconstruction to create 3D point clouds of objects.

- 3D reconstruction from multiple photographs is crucial for applications like autonomous driving and augmented reality; however, specialized sensors like LIDAR can be expensive and complicated to use.

Motivation

- Advances in digital cameras, improved resolution, and clarity have enabled new, more affordable 3D reconstruction techniques using only RGB cameras, without the need for expensive sensors.

- Structure from Motion (SfM) reconstructs 3D structures from multiple photographs taken from different viewpoints, with the main challenges being resilience, precision, completeness, and scalability; an incremental approach is used to address these issues.

Problem Statement

- Structure from Motion (SfM) reconstructs 3D structures from multiple photographs taken from different viewpoints.
- This work employs traditional image processing techniques and Incremental Structure from Motion (SfM) algorithm for 3D reconstruction to create 3D point clouds of objects.
- The proposed method is evaluated using specific 3D reconstruction datasets to assess its performance.

Dataset Description

- The dataset comprises of images of different 3D structures along with intrinsic camera matrix of the camera used to capture these images.
- All images for the given structure are captured using the same camera such that the order of images showcase a 360 degree motion of the camera around the structure.
- For each structure the number of images and their resolution is variable.

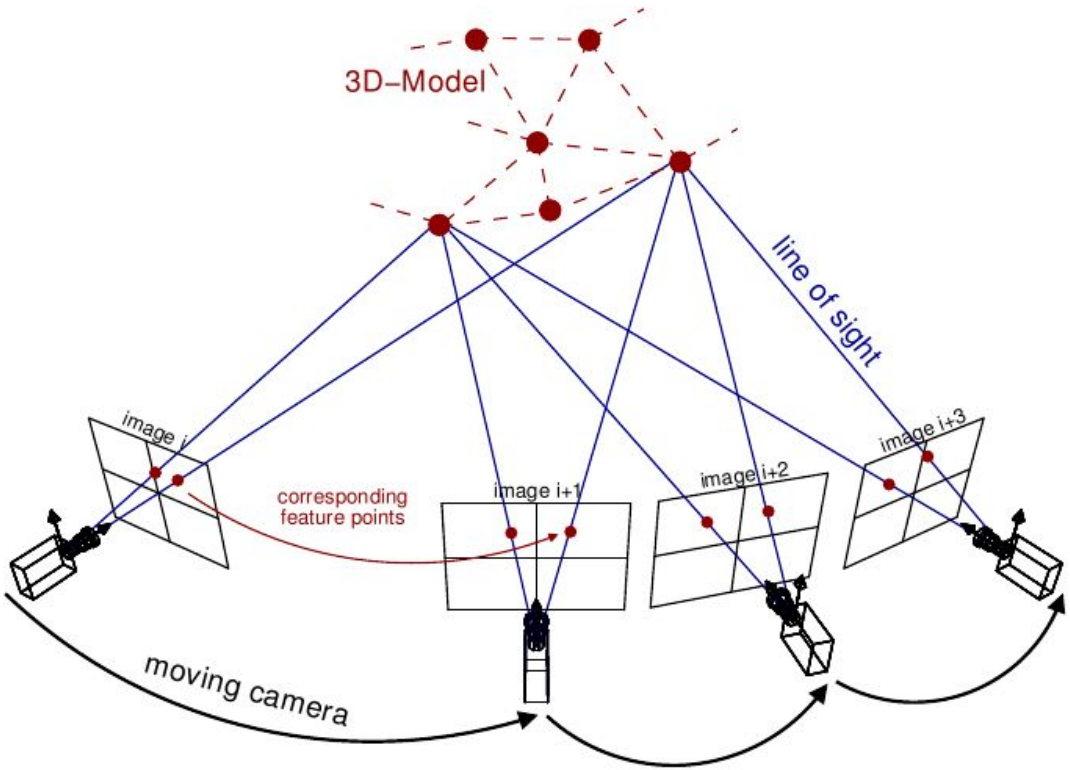
Dataset Description

- We searched the web to obtain some of the 3D structure images available on the internet provided their intrinsic camera matrix was also available.

- We utilised our **own phone's(OnePlus 7) camera** to obtain photographs of structures and obtained the camera's intrinsic matrix using camera calibration methods discussed in the course utilising a 12x9 checkerboard.

- In the Dataset folder, there is a separate folder for each structure. In a given structure folder there are images for that structure and a K.txt file containing the intrinsic camera matrix.

Dataset Description



Method used to capture structure images

Dataset Description

buddha1 (20 images)

Other examples:
buddha2 (41 images)
multi_building1 (19 images)
multi_building2 (30 images)
building_front (8 images)
fountain (11 images)
entrance (10 images)

statue (57 images)

speaker (20 images)

Methodology

- **Inputs**
  - Set of images { I_1, I_2 ….. I_N } from different view points
  - Camera calibration matrix, K
- **Output**
  - 3D Geometrical Scene, (x,y,z) corresponding projected points.

We use the incremental SFM (structure of motion) to find the motions of the cameras with respect to a world coordinate frame F_W, which can be also denoted as camera projection matrices { P_1, P_2 ….. P_N }.

1. Keypoint Extraction and Matching using SIFT Algorithm

- Given a set of unstructured images, we first find the connected components in these images. This helps us to find overlapping views in images.

a) **Key points** and **Descriptors** are extracted using SIFT algorithm on the 2 images.
b) These descriptors are then used to perform keypoint matching using a nearest neighbour search.

We extract the 2 Nearest neighbour for each key point based on euclidean distance between the descriptors and then check if

```
Dist of closest neighbours < 0.6*Distance of 2nd closest neighbour
```

We consider this **good keypoint matching** and will consider it for further computations.

2. Compute Essential matrix and Transformation between the 2 Image views

- Given matched keypoints in the 2 images, we compute the essential matrix using **8-point algorithm** which need at least 8 corresponding point pairs.

- To overcome the outliers or incorrect keypoint match, we use the RANSAC (Random Sample Consensus) algorithm to filter out outliers and estimate a more robust Essential Matrix.

```
ess_mat, em_mask = cv2.findEssentialMat(f0, f1, K, cv2.RANSAC, 0.999, 0.4)
```

- Threshold (0.4) : maximum distance between a point and its epipolar line for the point to be considered an inlier.
- Probability (0.999) : specifies a desirable level of confidence (probability) that the estimated matrix is correct.

```
_, rotn, trans, em_mask = cv2.recoverPose(ess_mat, f0, f1, K, em_mask)
```

- Uses the SVD method to decompose the Essential Matrix into a rotation matrix and a translation vector corresponding to two views of a scene. Assumption : Intrinsic parameters of both cameras are identical, K.

$$E = T_x R$$

3. Use Tx, R to find 2D-3D correspondence

- Given the Rotation, Translation matrix between the 2 views we calculate the transformation matrix and then using the Camera matrix K, we find the projection matrix for both cameras.

- Linear triangulation algorithm to compute the correspondences and obtain the 2D-3D correspondence (DLT Algorithm).

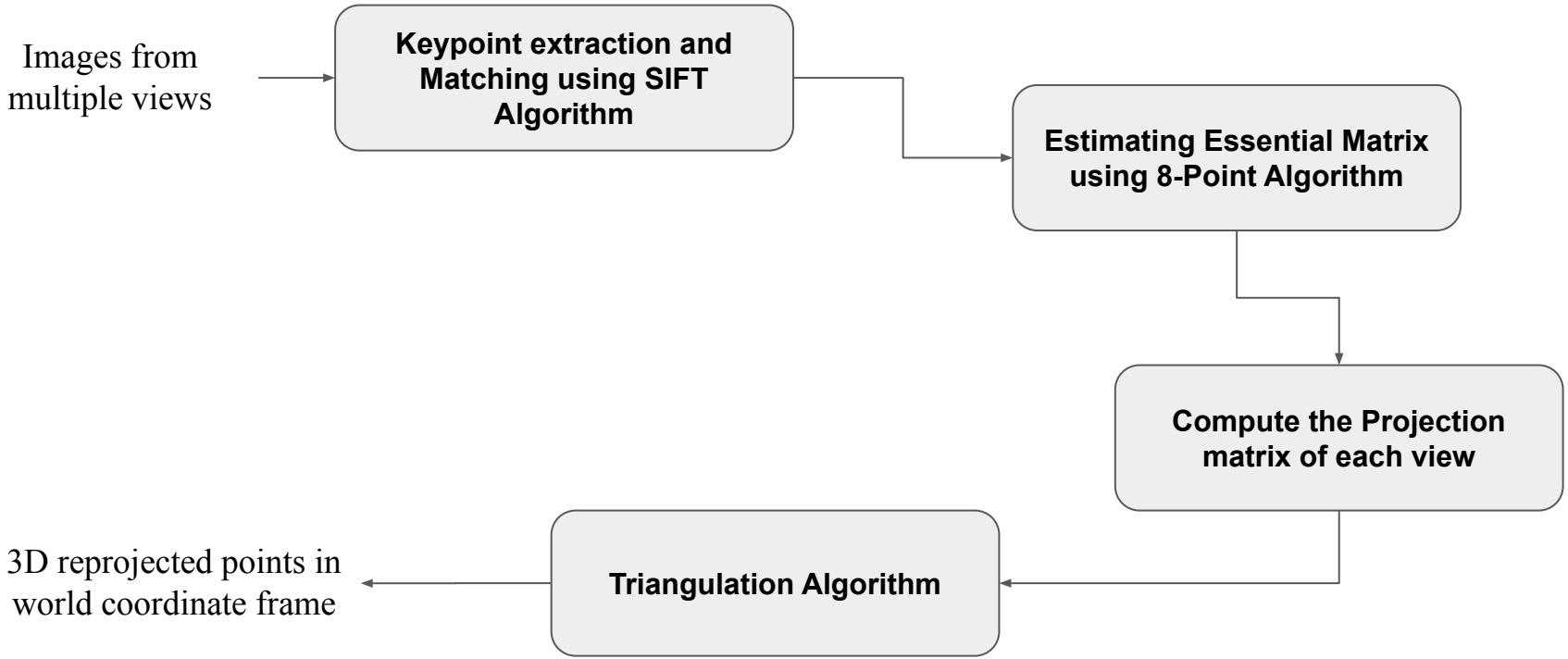
4. Add more views

- To add more views into the system, we first establish 2D - 2D correspondence between the newly added image and the previous image and 2D - 3D correspondence is then established to get the 3D points in similar fashion as done earlier. It helps us in getting more dense point clouds.

- Hence images should be passed in a order such that overlap of two images must be present to some extent. Hence, video frames is a good source to get such ordering.

- After getting the 2D - 3D correspondences, we use the Perspective-n-Point (PnP) algorithm to compute the pose of the images with respect to the world frame coordinates.

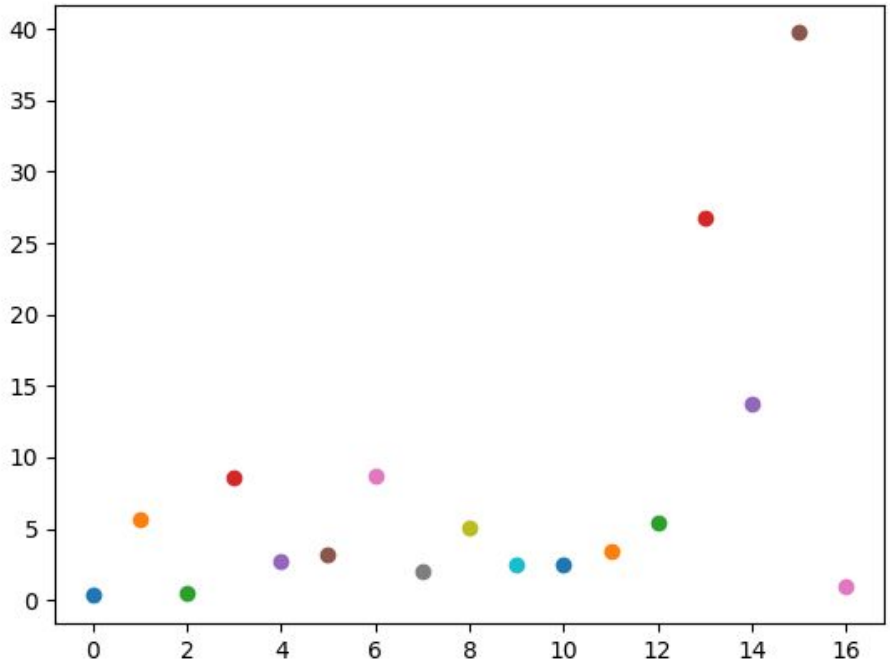
Our Pipeline



Results- buddha1

Results- buddha1

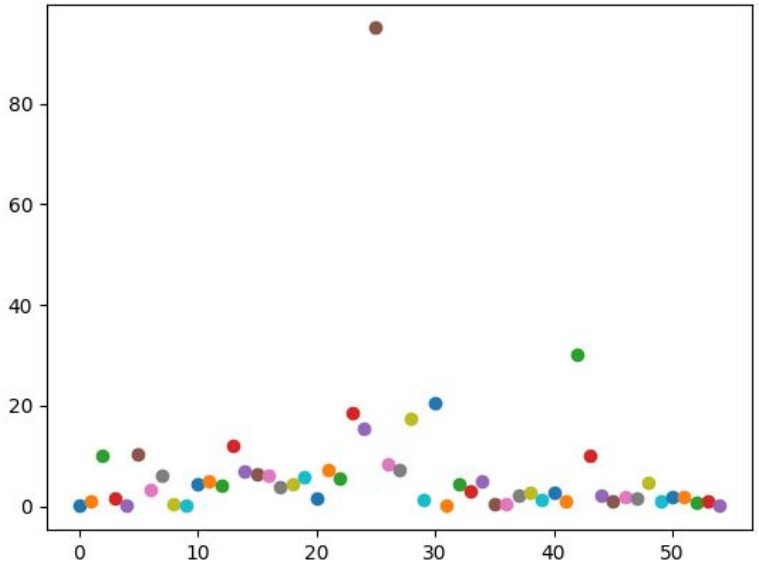
Reprojection Error Vs Image Graph



Results- statue

Results- statue

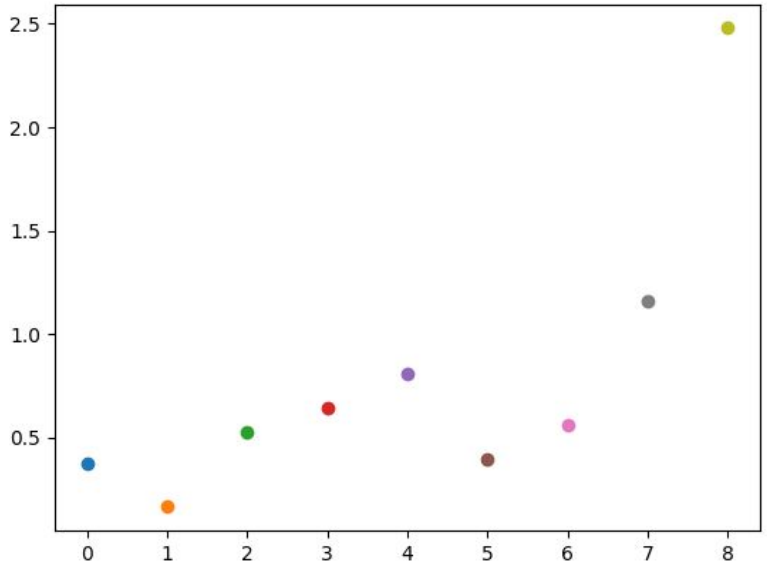
Reprojection Error Vs Image Graph



Results- fountain

Results- fountain

Reprojection Error Vs Image Graph



Results

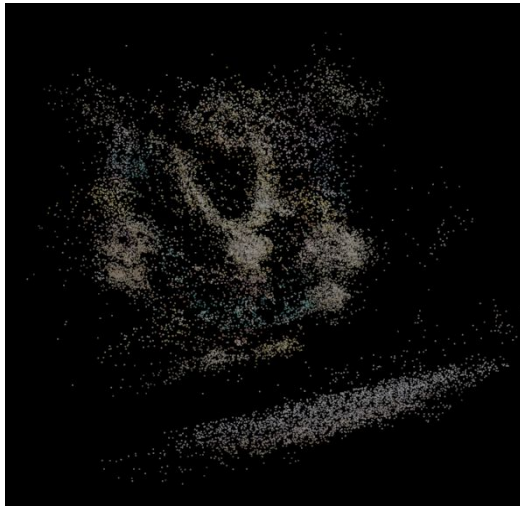
We observe that the quality of reconstructed scenes depends on several factors :

- Resolution and quality of images (Two cameras used)
- Number and distribution of keypoints (Different types of objects used : Speaker, Buddha, Statue)
- Number of image views
- With the number of the images and the quality of the distinct structure features visible in the images, the reprojection error decreases.
- However, we can see certain spikes in the reprojection error graphs, which showcase the importance of images features and positioning for the 3D reconstruction
- More results are available in the folder result folder for reference with different structures.

Results - New Experiments

- Capturing images of the objects with more distinctive angles with larger number of images while emphasizing the key features increases the clarity of the points detected and the points for distinct features increase drastically

Results- buddha2

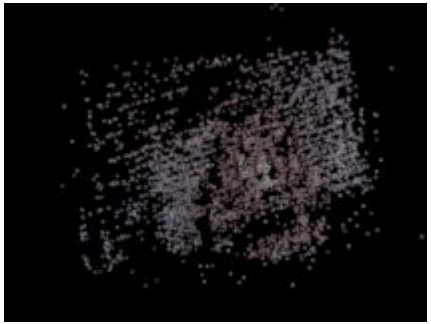


Results - New Experiments

Bundle adjustment algorithm - Refines the 3D points generated from triangulation

- It tries to minimize the difference between the observed feature points in the input images and their corresponding projections onto the reconstructed 3D scene.
- Uses an iterative method of adjusting the parameters of the camera poses, the 3D point positions, and the camera intrinsic parameters until the sum of the reprojection errors (the differences between the observed and projected feature points) is minimized.

Without Bundle



Bundle

Future Works

- Sparse representation of structure from motion lacks definitive quantitative metrics for model comparison and requires images to be passed in a specific order with sufficient overlap.
- Altering the image order may break the model, making it less flexible and adaptable to different situations.
- Despite sparse representations importance, dense image reconstruction is necessary for practical usage, achievable through the Multi-View Stereo (MVS) algorithm.
- We would be working on improving the above shortcomings in the future works.
- Explore the transformer based 2D encoder - 3D decoder kind of approaches to solve this problem.

References

- https://cmsc426.github.io/sfm/
- https://www.cs.cmu.edu/~16385/s17/Slides/11.4_Triangulation.pdf
- CVPR Lab at NUS (Youtube Channel)