# Custom Dataset-Based 3D Reconstruction: Utilizing Structure from Motion Algorithm for Transforming 2D Images into 3D Models

1st Muhammad Osama Nusrat
*Dept of Computing*
*Fast Nuces*
Islamabad, Pakistan
i212169@nu.edu.pk

2nd Hasan Mujtaba Kiyani
*Dept of Computing*
*Fast Nuces*
Islamabad, Pakistan
hasan.mujtaba@nu.edu.pk



***Abstract*—This paper aims to develop a 3D model of an object based on the methods described in the paper titled "Building Rome in a Day". The paper was reproduced using a custom dataset. The primary objective was to perform a 3D reconstruction using 2D camera images. The study considered motion from the perspective of the observer and the observed objects, with the camera having the freedom to move while the scene remained static. A lamp was chosen as the object of interest, and the goal was to accurately perceive its three dimensions within a specific scale. The process involved capturing images using a simple mobile phone camera, calibrating the cameras, and applying the incremental structure from motion computer vision technique to process the images.We successfully visualized the lamp in 3D using the implemented methods. The custom dataset allowed for a replication of the techniques described in the original paper, enabling the generation of a 3D model. The replication of the "Building Rome in a Day" approach on a custom dataset demonstrated the feasibility of 3D reconstruction from 2D camera images. The study validated the use of the incremental structure from motion technique and highlighted the potential of mobile phone cameras for achieving such reconstructions. This work opens avenues for further research in the field of 3D modelling and computer vision.**

## I. Introduction

Structure from motion is a technique used in computer vision that uses a collection of overlapping 2D images to develop a 3D model of the scene captured by the images. It is used in many disciplines where 3D information is vital for further processing like autonomous navigation, trajectory tracking, and surface conditioning. The best thing about this algorithm is that it is applicable on low cost cameras, hence it is an efficient technique for the 3D problems. For our case, we used a single mobile camera since it seemed the most suitable option. Recently, SFM has been integrated with technologies such as stereo and lasers to produce an absolute reconstruction of the scene, correct to a margin of few millimetres. There are some preliminary steps required to develop an SFM pipeline. The first of them is to determine the intrinsic and extrinsic matrix (optional) of the camera. This can be done by using the MATLAB camera calibration app. Intrinsic matrix contains different parameters like focal length (in pixels), resolution of image captured by camera etc.collectively used with camera extrinsic matrix which contains distance information with respect to the real world coordinates. The second step in the pipeline is to capture the images and sequentially process them to detect matching keypoints in the images. With the help of the matching keypoints and assuming the first camera pose to be a world coordinate (0,0,0) we can easily predict the new camera poses which represents their orientation and location with respect to the first camera position. The algorithms used for matching keypoints in images were SIFT/SURF. To run a reconstruction, we used non linear optimization techniques that involved concepts such as fundamental matrix, essential matrix, projection matrix, and bundle adjustment. These are discussed later in detail in the methodology section. Thus, the end goal of mapping the 2D image pixels to their respective 3D world position was obtained. Another objective was to compute a dense reconstruction if the key points in images were sparse(dispersed or scattered).

## II. Literature Review

### A. Structure From Motion

Structure from motion is basically used to predict the geometry of a material. Multiple images from different camera angles are taken of the same object and then we match the features of 2 images and reconstruct the 3d model of the object. First of all, take multiple images of your desired object from different camera positions. Then we take one image at a time and undistort that image.

In figure below we have a checkers board original image is distorted so we have to undistort it.

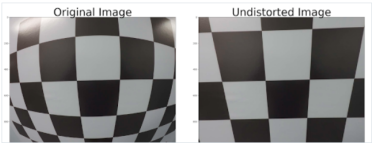


Fig. 1. Original image and Undistorted image

The image taken by first camera position is taken as a reference and we set its coordinate as (0,0,0).After that the image background is changed from RGB to gray because SFM works at gray scale images. In the next step SFM detect key points in our image. Keypoints are also called features of an image. Region of interest is found in our image and we crop it. From keypoint we can make a feature vector Feature vector are used to find the matching points in the two images. Same step is repeated from second image which is taken from different camera position. SFM detects key points from that image also. Note both images are of same object. The difference is that they are taken at different camera positions. Now these key features of both images will help to generate projection matrix Camera has 2 parameters intrinsic and extrinsic parameters. In intrinsic parameter include focal length and image size and extrinsic parameters include rotation and translation. when intrinsic and extrinsic matrix are combined we get projection matrix. With the help of projection matrix we can find a 3d points of an image. At this stage we have found matching points in two images and we know position of camera 1 which is origin.

Now we need to find position of camera 2. After finding the position of camera 2 we can find 3d points. Bundle adjustment is used to reduce error in case our 3d points are not accurate.

### B. What is SIFT

SIFT detects features in an image which are also called key points. We can make a 3d model from a set of 2d images using SIFT, a scale-invariant feature transform. What it does is that it takes two images of the same object at different camera orientations and then finds the key points in an image and matches the key point. It is called scale-invariant because we take images at different camera orientations but the point is that by changing orientation, the features of the image don't change.

## III. Methodology

### A. Camera Model

A camera is represented by a matrix PP which transforms a point X in the real world to a pixel x in the image. This relationship is given by the following equation

$$x = PX$$

To be able to find X, we need to do the following

$$X = P^{-1}x$$

In order to do the 3D reconstruction, we need to find the camera pose represented by the above camera matrix P for each of the images (the 2D snapshots or views).

### B. Incremental Structure From Motion

Finding the camera matrix or the camera pose for each of the images (the 2D snapshots or views) is a part of the process of reconstructing 3D coordinates. SFM, thus, involves estimating the 3D points along with the camera pose from a sequence of images. The 3D points are recovered by a procedure known as triangulation that uses camera poses to accurately locate the points in space. Incremental SFM chooses two images as the baseline views, obtains an initial reconstruction, and incrementally adds new images. The first camera position is assigned the location of the world coordinate (0,0,0) and the rest is computed as the images come along.

### C. Point Correspondences

Once we have a set of corresponding points using techniques like SIFT/SURF, we refine these matches and filter the wrong matches using RANSAC. The 8-point algorithm is used to compute the fundamental matrix. It requires a minimum of 8 points, however, the more good matching points the better.

### D. View

From the fundamental matrix, we calculate the new view which represents the position of the incoming camera. Thus, now we have the camera locations of the two cameras and the matching points in images taken by those cameras We can triangulate them for the corresponding 3D world points. The 3D points can be kept track of using a point tracker. Thus, with every incoming camera, we can update the point cloud with the new points and thus our reconstruction gets better Another algorithm to refine the reconstruction is bundle adjustment which is merely a non-linear least square optimization algorithm that can help refine camera poses by minimising reprojection errors. The refined point cloud can be used to extract valuable depth information about the object, or the shape of the object, or even create a map of the surroundings.

## IV. Evaluation and Experiments

The evaluation and experiments are discussed below.

### A. Camera Calibration

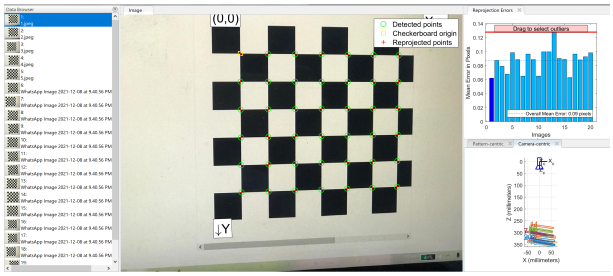


Fig. 2. Camera Calibration

In order to transform from 2D world coordinates (Camera Coordinates) to 3D World Coordinates, there are some transformations which need to be done using certain camera characteristics (Intrinsic + Extrinsic). Intrinsic parameters deal with the camera's internal characteristics, such as its focal length, skew, distortion, and image centre. Extrinsic parameters describe its position and orientation in the world Knowing intrinsic parameters is an essential first step for 3D computer vision, as it allows you to estimate the scene's

structure in Euclidean space and removes lens distortion, which degrades accuracy. Camera calibration contains both Intrinsic and extrinsic camera calibration.

### *B. Camera Intrinsic*

The intrinsic matrix of the camera can be given as:

$$M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Fig. 3. Intrinsic Matrix

Where (fx, fy) are the focal point and (cx, cy) are the principal point.

Generally, the camera calibration process uses images of a 3D object with a geometrical pattern (e.g. checkerboard).

The pattern is called the calibration grid. The 3D coordinates of the pattern are matched to 2D image points.

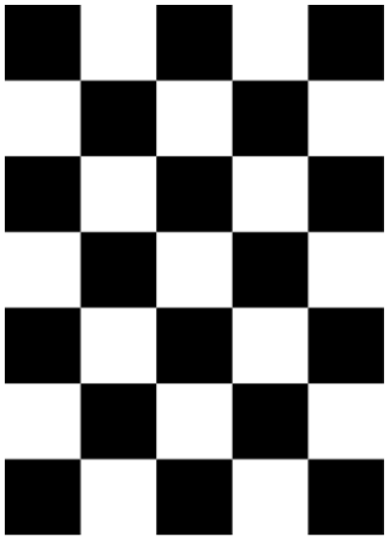

Fig. 4. Calibration Grid

In order to measure the intrinsic parameters of camera, we have followed the following steps:

- Take 15-20 pictures of checkerboard patterns from different angles.
- Open the Matlab camera calibrator app from Apps –¿ Image processing and Computer Vision –¿ Camera Calibrator.
- Load all the images into the calibrator app

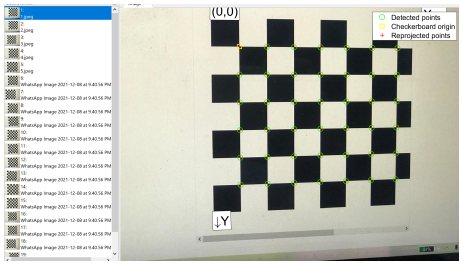


Fig. 5. Images Loaded for Calibration

- App will automatically detect the checkerboard patterns from all the images
- Click on calibrate button
- After processing, Camera intrinsic parameters will be calibrated. One can import the parameters to the workspace and can also generate the file for auto-calibration for future needs.

### *C. Camera Extrinsic*

Camera Extrinsic describes the camera position and orientation with respect to certain world coordinates. They are calibrated with respect to certain reference world coordinates. Matlab perceive top left corner of image as (x, y) = (0,0). x increases horizontally, from left to right, while y increases vertically, from top to bottom.

### *D. Feature Matcing*

In the following figure, we can see the feature matching of images.

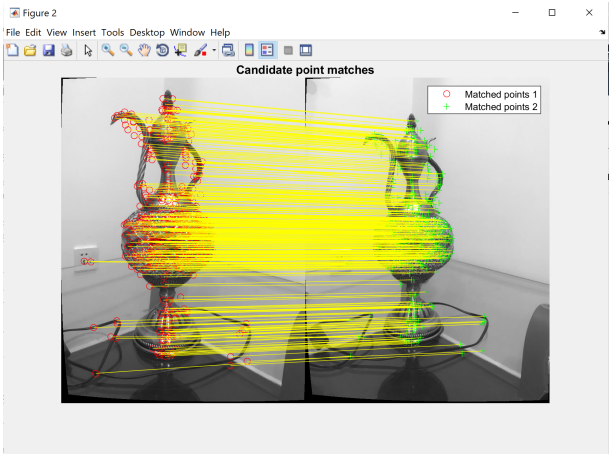


Fig. 6. Feature matching of images

Feature matching is a fundamental process in computer vision, particularly in the context of the Scale-Invariant Feature Transform (SIFT) algorithm. SIFT is a widely used method for extracting and matching distinctive features in images.

The feature matching step in SIFT involves finding correspondences between the keypoints detected in different images. Keypoints are robust and distinctive image features

that can be identified in different scales and orientations. To perform feature matching, SIFT computes a descriptor vector for each keypoint, which represents its local appearance.

In the matching process, the descriptor vectors of keypoints in one image are compared to the descriptor vectors of keypoints in another image. Various distance metrics, such as Euclidean distance or cosine similarity, can be employed to measure the similarity between descriptors. The keypoints with the most similar descriptors are considered matches.

### *E. Final Output*

The reconstructed image has been shown in the figure below.

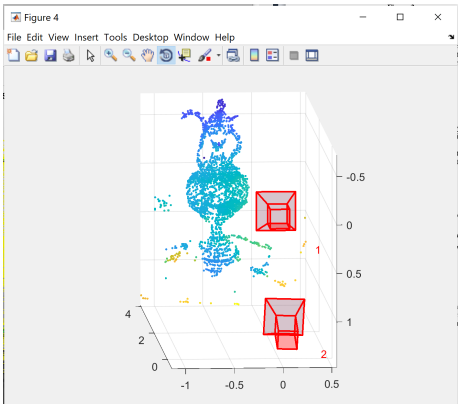


Fig. 7. Reconstructed image

## V. CONCLUSION

We successfully replicated the techniques presented in the paper Building Rome in a Day on a custom dataset. The aim was to develop a 3D model of an object using 2D camera images. By applying the incremental structure from motion technique and utilizing a mobile phone camera, the study achieved the visualization of a lamp in 3D, accurately perceiving its three dimensions within a defined scale. The findings validate the feasibility of 3D reconstruction from 2D images and highlight the potential of mobile phone cameras in this context. This research opens avenues for further exploration and advancements in the fields of 3D modelling and computer vision.

## VI. DATASET AND CODE

We used Matlab for coding purposes.The dataset and the developed code are made available on Github: https://github.com/mnusrat786/3D-Reconstruction-from-2D-images-Using-Structure-from-Motion-Algorithm

## REFERENCES

[1] Agarwal, S., Furukawa, Y., Snavely, N., Simon, I., Curless, B., Seitz, S. M., & Szeliski, R. (2011). Building rome in a day. Communications of the ACM, 54(10), 105-112.

[2] Agarwal, S., Snavely, N., Seitz, S. M., & Szeliski, R. (2010). Bundle adjustment in the large. In Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part II 11 (pp. 29-42). Springer Berlin Heidelberg.

[3] Haner, S., & Heyden, A. (2012). Covariance propagation and next best view planning for 3d reconstruction. In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12 (pp. 545-556). Springer Berlin Heidelberg.

[4] Frahm, J. M., Fite-Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., ... & Pollefeys, M. (2010). Building rome on a cloudless day. In Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11 (pp. 368-381). Springer Berlin Heidelberg.

[5] Agarwal, S., Furukawa, Y., Snavely, N., Curless, B., Seitz, S. M., & Szeliski, R. (2010). Reconstructing rome. Computer, 43(6), 40-47.

[6] Ham, H., Wesley, J., & Hendra, H. (2019). Computer vision based 3D reconstruction: A review. International Journal of Electrical and Computer Engineering, 9(4), 2394.