

# Disaster Site 3D Reconstruction using Stereo Cameras

Rushikesh Jalvi

Department of Information Technology  
St. Francis Institute of Technology  
Mumbai, India

Harrshada Bhagath

Department of Information Technology  
St. Francis Institute of Technology  
Mumbai, India

Sanket Dalvi

Department of Information Technology  
St. Francis Institute of Technology  
Mumbai, India

Purnima Kubde

Department of Information Technology  
St. Francis Institute of Technology  
Mumbai, India

**Abstract**—Building of 3D models often requires specialized hardware, such as laser range finders, RGBD cameras and 3D rigs, which are expensive and complex to manipulate. In this paper we propose a system using a mobile phone camera to capture images of the disaster site. These recorded images are then processed to generate a 3D mesh file which is then visualized using an open source software for path planning and extensive analysis of the site.

## I. INTRODUCTION

3D Reconstruction of a scene, an evolving technique in the field of computer vision, finds applications in various fields. One such application is in the field of disaster management. Natural disasters typically damage infrastructure, cause injury and massive loss of life. An immediate life-saving response is essential to rescue those who are trapped and stabilize or evacuate survivors. In these critical situations, a 3D model of the disaster site will help in analyzing and planning efficient rescue operations. The 3D model helps in providing in depth analysis of the damage and can be studied to improve the structural soundness of the site. It helps protect the lives of rescuers from any unwanted dangers which they might head into because of no prior knowledge of the scene. Trapped victims can be located and a direct path can be mapped reducing time to save the life of the victim. We propose a model which captures 2D images of a disaster site and renders a 3D model for the same. The images will be captured using the MI A1 phone camera. camera matrix will be obtained by training a model using sample images of a 6x9 chessboard. Camera calibration is done by detecting the edge intersection of each box using OpenCV libraries in python. The camera matrix along with the two input images are used to generate a depth map and a mesh file with .ply extension using OpenCV. A depth map contains information relating to the distance of the surfaces of scene objects from a viewpoint and the mesh file stores information about the 3D model. This point cloud is visualized using MESH LAB.

## II. RELATED WORK

In [1], a method to construct a three-dimensional (3D) model of an outdoor scene containing a large amount of information by combining local stereo images which are acquired from multiway is proposed. 3D model reconstructed by PMVS is

very accurate, some holes remain that can not be spread because 3D model of PMVS originates from a sparse set of feature points. In this study, visibility constraints also reduce the elements of the 3D model because the images captured in the sky have fewer inter- sections. Individual stereo images can reconstruct denser but less accurate 3D models than PMVS. The algorithm needs less intersections of images to reconstruct the accurate 3D model using these images than PMVS needs. The increase of intersections is able to improve the accuracy of the 3D model. This paper uses a constructive solid geometry (GCS) model to express the visual hull. The Proposed method was compared with the existing PMVS method and the results obtained were denser than the PMVS algorithm. In addition, this model shows that the side surface of the model is filled compared to PMVS. This method is expensive as it contains many mechanical parts to capture the data.

In [2], a method that achieves 3D reconstruction just using one consumer grade family camera is proposed. Traditional 3D models are often built around specialized hardware, such as laser range finders and stereo rigs, which are expensive and complex to manipulate. Only one consumer grade digital camera is used which is cheaper than active range sensors and stereo vision system so that it can be afforded by any amateur. We apply calibration procedure only once before hand in order to get the intrinsic parameters of the camera. SIFT is a successful feature matching method which is robust to image scale and illumination changes. It can also handle affine transformations between images to some extent. An efficient NCC calculation algorithm based on integration image and greatly accelerates the image matching process. The author took two pictures of rock from different angles and successfully constructed a 3D model. Moreover, a hill at a distant which cannot be constructed using TOF sensors was constructed using this method. The methodology is efficient in scanning single objects but not compatible with reconstructing the entire scene.

In [3], Face as a biometric identification in computer vision is an important medium, in areas such as video surveillance, animation games, security anti-terrorist has a very wide range of applications, creating vivid, strong visibility of 3d face model, now has become a challenging in the field of computer vision is one of the important

topics. At first, the Zhongxing micro ZC301P cameras to build a binocular stereo vision system for recording images. After the camera calibration and binocular calibration, the three-dimensional data of facial images were extracted using the functions of OpenCV computer vision library, and then 3d face model were reconstructed preliminarily by DirectX. According to the reconstruction process, the human face three-dimensional reconstruction software was designed and developed. This system does not capture facial information and construct 3D facial models on a real-time basis. The accuracy of the system seems to be very low. The methodology is efficient in scanning single objects but not compatible with reconstruction of the entire scene.

In [4], A novel volumetric multi-resolution mapping system for RGB-D images that runs on a standard CPU in real-time. This approach generates a textured triangle mesh from a signed distance function that it continuously updates as new RGB-D images arrive. Reconstructing the geometry and texture of the world in real-time for a sequence of images is a key challenge. Reconstructing a regular grid requires an enormous amount of memory and computation time. Architects would greatly benefit from a wearable 3D scanning device that generates and visualizes a 3D model in real-time. A robot navigating through an unknown environment benefits from an up-to-date 3D map to support obstacle avoidance, path planning, and autonomous exploration. As most space is either free or unknown, only those voxels are allocated and updated that are located in a narrow band around the observed surface. This approach generates a textured triangle mesh from a signed distance function that it continuously updates as new RGB-D images arrive. An octree as the primary data structure is generated which allows us to represent the scene at multiple scales. It allows the growth of the reconstruction volume dynamically. As most space is either free or unknown, it allocates and updates only those voxels that are located in a narrow band around the observed surface. In contrast to a regular grid, this approach saves enormous amounts of memory and computation time. An octree data structure that supports volumetric multiresolution 3D mapping and meshing, A speeded-up data fusion algorithm that runs at a resolution of up to 5 mm in real-time on a single CPU core, A multi-resolution, incremental meshing algorithm that runs asynchronously on a second CPU core and outputs an up-to-date mesh at approximately 1 Hz. The geometry was represented using a signed distance function (SDF), that provides any point the signed distance to the closest surface. The SDF is represented in an octree, where only cells close to the surface are actually allocated. As different parts of the scene will be observed at different distances, the geometry information will be saved at different levels in the tree. A single voxel stores the truncated signed distance, the weight, and the color. The time of the first thread to traverse the tree to allocate and queue branches, bricks, and mesh cells in the tree was measured and the results were displayed. The sequential behavior of the algorithm was studied by evaluating the computational load over time. A total time needed per RGB-D frame for the traversal and the SDF update was plotted. The processing time of the meshing

queue, and thus the latency at which the updated mesh becomes available was studied and observed. The voxel neighborhood information provided by the mesh cells could be used to perform efficient regularization of the SDF. Camera tracking based on the computed map, with the goal to assist or replace an external SLAM system serves as an avenue for further research.

In [5], the author aims at creating a new approach by combining two techniques: 1. Generating a detailed volumetric model of the environment and 2. Combining with highly simplified representation of the environment, containing only geometric aspects. This allows segment volumetric representations of interior objects in the environment. The input scans come from an ambulatory indoor scanning system by walking through the indoor environment at normal speeds. Since the system is mobile, the resulting point cloud has higher noise than traditional static scanners. The first mesh only represents the building geometry, including floors, walls. The second mesh represents only the objects in the environment. Both the meshes are processed differently: The building geometry is split into planar surfaces and triangulated efficiently, preserving sharp corners. The object geometry is refined and meshed uniformly to preserve its fine structure. Then using probabilistic model and carving an effective 3D model is generated of both meshes, which are then combined re-carving and obtaining a final 3D model. The scanning took 2 minutes and processing took 6 hours. The output relies only on assumptions made by the scanned environment. The input procedure produces a lot of noise which can be eliminated by using multiple images as input instead of scanning the system.

In [6], an active 3D mapping method for depth sensors, allows individual control of depth-measuring rays is proposed. Task-driven reactive control steering hundreds of thousands of rays per second using only an onboard computer is a challenging problem, which calls for highly efficient parallelizable algorithms. As a first step towards this goal, the author proposes an active mapping method for SSL-like sensors, which simultaneously (i) learns to reconstruct a dense 3D occupancy map from sparse depth measurements (ii) optimizes the reactive control of depth-measuring rays. The main contribution of this paper lies in proposing a computationally tractable approach for very high dimensional active perception task, which couples learning of the 3D reconstruction with the optimization of depth measuring rays. An assumption was made that the vehicle follows a known path consisting of  $L$  discrete positions and a depth measuring device (SSL) which can capture at most  $K$  rays at each position. Based on this, algorithms for Active Mapping and Learning of Active Mapping were written. A greedy planning algorithm was employed for the same. The greedy algorithm successively selects rays that reduce the cost function the most. To show how cost function differs from OPT, an upper bound on the cost function was derived. A significant speedup of the greedy planning algorithm was observed by imposing prioritized search for a specific argument. All experiments were conducted on selected sequences from categories City and Residential from the KITTI dataset. The 3D reconstruction CNN outperforms a

state-of-the-art approach by 20 percent in recall, and it is shown that when learning is coupled with planning, recall increases by additional 8 percent on the same false positive rate. The experimental setup focuses majorly on mapping the data of object through sensors and then evaluated using KITTI datasets. The output produced is a very sparse reconstruction of the scene, rendering a model of very low accuracy.

In [7], The paper discussed an approach to reconstruct a 3D surface of small objects using low cost options such as a stereo camera. 3D objects traditionally created with standard design method takes a long time and need skill from the 3D artist, while the need for 3D objects keeps increasing. 3D data also can be made into a real object with 3D printer technology. The paper focuses on how to get good quality object reconstruction using low cost. It was about this research that is done, performance analysis that affects the quality of the 3D object reconstruction so that the results are more detailed and more accurate. Lasers are used to sweep the object completely to register minute details which are then merged with the image data. The images provided by the stereo camera will provide the depth data. The geometry is calculated by use of triangulation. The object to be reconstructed is setup on a rotating assembly for laser to sweep the entire surface. The image data and laser obtained data then is used to generate a 3D point cloud which on processing gives complete reconstruction of the object. The surface from 3D objects can be obtained by combining point clouds of an object from multiple angles. Texture reconstruction can be done by attaching a picture of the object into a reconstructed 3D shape. Precise calibration and accurate camera positioning are necessary for successful reconstruction results objects. Surface reconstruction with this system still has the surrounding point cloud that should be removed manually through a 3D program to obtain surface objects only. The author kept a stuffed toy on the rotating assembly to obtain the aforementioned results. Light conditions strongly affect the reconstruction process. Also the noise present will affect the reconstruction. The methodology is efficient in scanning single objects but not compatible with reconstruction of the entire scene.

In [8], the current 3D reconstruction techniques and proposing a new technique based on inertial sensors and uncertainty modelling is discussed. One of the main problems in human body 3D reconstruction is the data registration from different planes which is different from multisensory image fusion methods. The paper provides an explorative review of data registration frameworks in terms of 3D reconstruction, that involves geometric relations, and handling underlying uncertainties. The reconstruction of the internal human body would be of much significance as it will develop medical science to a new level. A reconstruction model using inertial sensors data and images. The model makes use of homographs-based techniques. The data from inertial sensors consists of orientation, distance, etc. the model makes use of homograph-based techniques. The inertial sensor data is fused with image matrix giving another 3D matrix having orientation, distance information. An

important assumption is done that there should be a planar ground to mark some points for reference. Thus, giving a complete 3D reconstruction of the object. The experiment is conducted by placing a model or an object in a room full of other materials. The reconstructed mesh of only the object is obtained. This technique makes no extraction of feature instead uses an outline of objects. An important assumption is done that there should be a planar ground for reference.

In [9], This paper discusses the various 3D reconstruction techniques used in civil engineering and their applications in various fields. 3D reconstruction techniques in civil engineering and their applications have been summarized in a number of review papers. This paper systematically summarizes 3D reconstruction techniques and the up-to-date achievements and challenges for the applications of the techniques in civil engineering and proposes key future research directions in the field. The author reviewed the existing 3d reconstruction techniques in civil engineering. Also, a brief summary about the techniques used on input obtained from monocular images, stereo images and video frames is done. The two most important steps of 3d reconstruction are generation of point clouds and processing of point clouds. Generation of point clouds is subdivided into 7 steps and processing is divided further into 4 steps. The linear execution of these steps results in 3D reconstruction of the entire scene. The comparison of various algorithms used in these steps and difference in operation time, efficiency is noted. There is no such tangible output mentioned in the paper as it contains only reviewing of various techniques. There was no experiment performed to test the techniques by the author. Automation of 3d reconstruction can be achieved by automating the sub steps which will reduce the operation time and thus giving an efficient model.

In [10], A simple method for calibrating a set of cameras not having overlapping field of views was presented. The problem of calibrating the non-overlapping cameras was reduced to the problem of localizing the cameras with respect to a global 3D model reconstructed with a simultaneous localization and mapping (SLAM) system. Specifically, a global 3D model was reconstructed using a SLAM system which used an RGB-D sensor. Localization and intrinsic parameter estimation were performed for each camera using 2D-3D correspondences between the camera and the 3D model. This method locates the cameras within the 3D model, which is useful for visually inspecting camera poses and provides a model-guided browsing interface of the images. The advantages of this method were demonstrated using several indoor scenes. The accuracy of the calibration method is bounded by the accuracy of the external RGB-D SLAM system. However, the accuracy of recent SLAM systems has reached a sufficient level to be used for the calibration purpose as demonstrated. If the descriptor matching fails to identify the closest images to the query image, this method fails to estimate the correct pose. Placing some discriminative reference object in the FOV of the camera would resolve such cases.

In [11], planes are dominant in most indoor and outdoor scenes and hence it focuses on the development of a hybrid algorithm that incorporates both point and plane features. It presents a tracking algorithm for RGB-D cameras using both points and planes as primitives and shows how to extend the standard prediction and correction framework to include planes in addition to points. By fitting planes, the noise in the depth data that is typical in many commercially available 3D sensors was implicitly taken care of. In comparison with the techniques that use only points, this tracking algorithm has fewer failure modes, and the reconstructed model is compact and more accurate. The tracking algorithm is supported by re-localization and bundle adjustment processes to demonstrate a real-time simultaneous localization and mapping (SLAM) system using

a hand-held or robot-mounted RGB-D camera. The experiments show large-scale indoor reconstruction results as point-based and plane-based 3D models, and demonstrate an improvement over the point-based tracking algorithms using a benchmark for RGB-D cameras. The tracking framework accelerated the feature detection and correspondence search, and enabled avoidance of incorrect correspondences in areas with repetitive texture. There are some drifts along the directions which are not supported by planes

III. SYSTEM DESIGN

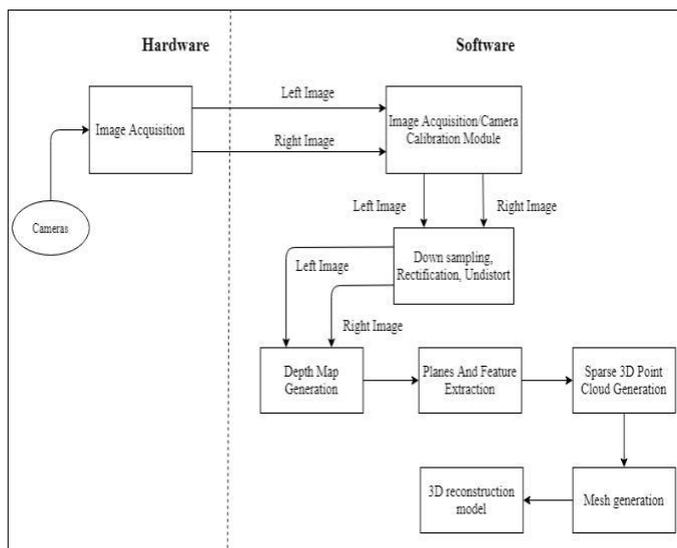


Figure 3. Architecture Design

**Image Acquisition/ Camera Calibration Module:** Images of object are captured using MI A1 phone camera having a resolution of 12 MP. To obtain the camera matrix, we train a model using sample images of a 6x9 chessboard. For, camera calibration, the edge intersection of each box was detected using OpenCV libraries in python.

**Depth Map Generation:** The camera matrix is passed here along with the two input images for calculating the

disparity map. The disparity map is calculated using OpenCV.

**Mesh Generation:** A mesh file is generated with a .ply extension which is rendered in MESHLAB.

**Point Cloud Reconstruction:** In MESHLAB, using Poisson surface reconstruction, 3D model of the object in the image is generated.

IV. METHODOLOGY

A. Camera calibration

Images of an object were captured using an MI A1 phone camera having a resolution of 12MP. The goal of camera calibration is to learn about any radial and tangential distortion in the image, and to obtain intrinsic and extrinsic parameters of camera [13]. These parameters are used to undistort the image. To solve these distortions, we need to find the distortion coefficients:

$$Distortion\ coefficients = (m_1\ m_2\ \dots\ n_1\ n_2\ m_3)$$

The radial distortion is solved as follows:

$$x_{corrected} = x (1 + m_1r^2 + m_2r^4 + m_3r^6)$$

$$y_{corrected} = y (1 + m_1r^2 + m_2r^4 + m_3r^6)$$

The tangential distortion is solved as follows:

$$x_{corrected} = x + [ 2n_1xy + n_2(r^2 + 2x^2) ]$$

$$y_{corrected} = y + [ 2n_2xy + n_1(r^2 + 2y^2) ]$$

In addition to these we need to find the focal length and optical centers and express them as a 3x3 camera matrix.

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

Setup:

For camera calibration, 90 test images of a 6x9 chessboard pattern taken from different angles were used. The OpenCV function cv2.findChessboardCorners() returns the corner points and retval if the chessboard pattern is obtained.

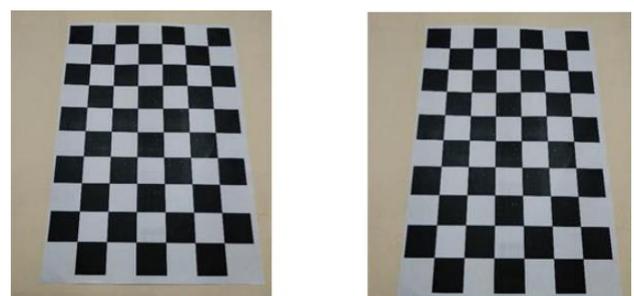


Figure 1. (a) original image of (b)

6 x 9 chessboard with distortion (b) image without distortion after applying calibration technique

**B. 3D Reconstruction**

First, we calculate the optimal camera matrix in case the image size has changed. The images are then undistorted and down sampled to reduce the frame size to reduce the computational time. OpenCV's StereoSGBM (Stereo semi global block matching) technique is used and the input parameters of the technique are tuned appropriately to obtain the best depth map. We need to specify to what point a disparity (offset) is acceptable. For this, specify the minimum and maximum disparities has to be specified. Subtracting minimum and maximum disparities we calculate the number of disparities, which is a way to specify the acceptable range for which pixels can move in the picture.

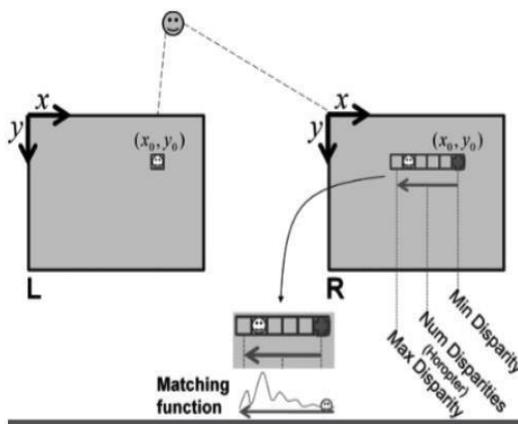


Figure 2. Underlying algorithm for calculating disparity and generating point cloud [12].

The algorithm to generate a point cloud is present in the OpenCV documentation. It reshapes the colors and vertices and then stacks them one over the other. The output array is written into file with a header that is saved as a .ply file. This file can be visualized using MeshLab.

**IV. RESULTS**



(a)

(b)

Figure 4. (a) input left image (b) input right image. input stereo images provided to be reconstructed into a 3D mode

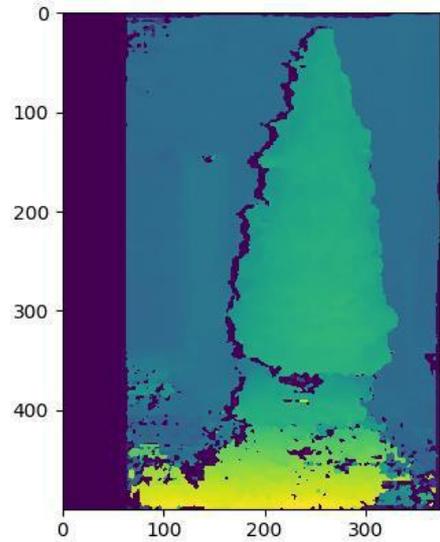


Figure 4. generated depth map of the input stereo images. Brighter the color on the depth map, closer is the voxel in 3D space. It can be observed from the depth map how the color gradients from bright yellow to blue as the elements in the image get farther away. Depth Map generation depends upon the extent to which the images are down sampled. Our program took 10 seconds for depth map creation.



(a)



## VI. REFERENCES

- [1] T. Yoshida and T. Fukao, "Dense 3D reconstruction using a rotational stereo camera," *2011 IEEE/SICE International Symposium on System Integration (SII)*, Kyoto, 2011, pp. 985-990.
- [2] Y. Shen, P. Peng and W. Gao, "3D reconstruction from a single family camera," *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, Nanjing, 2012, pp. 108-112.
- [3] J. Yin and X. Yang, "3D facial reconstruction of based on OpenCV and DirectX," *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, Shanghai, 2016, pp. 341-344.
- [4] F. Steinbrücker, J. Sturm and D. Cremers, "Volumetric 3D mapping in real-time on a CPU," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 2021-2028.
- [5] E. Turner and A. Zakhor, "Automatic Indoor 3D Surface Reconstruction with Segmented Building and Object Elements," *2015 International Conference on 3D Vision*, Lyon, 2015, pp. 362-370.
- [6] K. Zimmermann, T. Petricek, V. Šalanský and T. Svoboda, "Learning for Active 3D Mapping," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, pp. 1548-1556.
- [7] Ma, Zhiliang & Liu, "Shilong. (2018). A review of 3D reconstruction techniques in civil engineering and their applications," *Advanced Engineering Informatics*. 37. 163-174.
- [8] H. Aliakbarpour, V. B. S. Prasath, K. Palaniappan, G. Seetharaman and J. Dias, "Heterogeneous Multi-View Information Fusion: Review of 3-D Reconstruction Methods and a New Registration with Uncertainty Modeling," in *IEEE Access*, vol. 4, pp. 8264-8285, 2016.
- [9] A. Harjoko, R. M. Hujja and L. Awaludin, "Low-cost 3D surface reconstruction using Stereo camera for small object," *2017 International Conference on Signals and Systems (ICSigSys)*, Sanur, 2017, pp. 285-289.
- [10] E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam and Y. Miki, "Calibration of Non-overlapping Cameras Using an External SLAM System," *2014 2nd International Conference on 3D Vision*, Tokyo, 2014, pp. 509-516.
- [11] E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam and T. Garaas, "Tracking an RGB-D Camera Using Points and Planes," *2013 IEEE International Conference on Computer Vision Workshops*, Sydney, NSW, 2013, pp. 51-58.
- [12] A. Bradski, *Learning OpenCV, Computer Vision with OpenCV Library; software that sees*, 1. ed. O'Reilly Media, 2008.
- [13] "Camera Calibration," *OpenCV*. [Online]. Available: [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_calib3d/py\\_calibration/py\\_calibration.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html).

Figure 5. (a) front view of 3D model (b) view of the model at angle (c) side view of the 3D model. 3 different angles of the reconstructed 3D model. Execution time here again depends on the down-sampled images. It took again 10 seconds to create a ply file.

## V. CONCLUSION

In this paper we have implemented algorithms and techniques to match features, calibrate the camera, obtain a depth map and generate a point cloud. Images of an object were captured using MI A1 phone camera having a resolution of 12MP. To obtain the camera matrix, a model was trained using sample images of a 6x9 chessboard. For camera calibration, the edge intersection of each box was detected using OpenCV libraries in python. For training 50-60 images of chessboard from different angles were used and the camera matrix was generated. After generating the camera matrix, two images of the object to be reconstructed were taken and processed on the calibrated camera module. OpenCV libraries in python were used to calculate the depth map using the two images as input. After generation of depth map, a mesh file is generated with a .ply extension which is rendered in MESH LAB. The python code for mesh file uses OpenCV commands.