



Marker Based Augmented Reality with 3D Model Generation using Photogrammetry

Aishwarya M¹, Apurva², Irene Getzi S³

¹Student, Department of MCA, Jyoti Nivas College, Bangalore, India

²Student, Department of MCA, Jyoti Nivas College, Bangalore, India

³Faculty, Department of MCA, Jyoti Nivas College, Bangalore, India

ABSTRACT

Augmented reality (AR) is an increasingly recognized paradigm of information visualization that merge a computer-generated virtual object onto a real scene to visually augment reality. To achieve these requirements, numerous technologies on computer vision and image processing have made immense contributions. As of now, augmented reality mostly makes use of plane detection and marker detection to find and locate objects. While instant Augmented Reality (AR) applications can be created using some of the out-of-the-box libraries or software available on the market, the use of Python for AR, while still a bit nascent, is a gateway to a multitude of opportunities. At its heart, AR can be said to be simply an issue with computer vision with some twists, and that's something that Python has already taken good care of with numerous libraries such as OpenCV. This work focuses on the development of AR Markers to position 3D objects on the plane using photogrammetry techniques. The markers are identified by using keypoint descriptors such as SURF and SIFT.

Keywords: Augmented Reality, Augmented Reality, Photogrammetry, Python, OpenCV, marker-based AR and 3D model Creation.

1. INTRODUCTION

Technology has had a huge impact in modern age on the quality of life we lead. Many technologies have been used in the various sectors that have been tremendously benefited as a result of technological advancement. Augmented Reality is one such emerging technology that has tremendous promises in various aspects in the different fields. The idea is to give the user a view of the virtual object in the real world using an Object based AR system. Augmented Reality (AR) uses technologies for computer vision, image processing, and computer graphics to merge digital content into the real world. It allows user, physical objects, and virtual objects to communicate in real-time. For instance, AR can be used to insert 3D graphics into a video, as if the virtual elements were part of the actual environment.

A basic computer of augmented reality is made up of a camera, a computing unit and a monitor. The camera captures an image, then the machine increases the virtual objects on top of the image and shows the result. Enhanced reality technology is useful for several fields of use. It is well suited for both indoor and outdoor on-site visualization, for visual guidance in assembly, maintenance, and training. Increased reality allows for interactive games and new forms of publicity. Many location-based services make use of browsers with augmented reality. In printed media, augmented reality connects 3D graphics and videos with printed publications. This work provides an experience that will help users through the use of AR Markers to position artificial

2D and 3D objects in the real world. Using OpenCV and Python, 3D object positioning is performed on the marker. The marker will be identified using Feature Identification, Feature Definition, and matching feature. It includes feature matching method and key detection system to use RANSAC algorithm to locate the homography. The method of detection only works for detecting two similar objects. Various tools, techniques, libraries, technologies and languages such as Unity3D, Augmented Reality, and so forth were used to develop this work.

2. BACKGROUND STUDY

This study offers a comprehensive overview of Marker Based Augmented Reality with 3D Model Generation using Photogrammetry. The learning about the emerging technology Augmented Reality that has tremendous promises in various aspects in the different fields and its types [1]. This technology allows the user to decide, where to place the object in real world. Once the object has been placed in the scene, it will be displayed accurate according to the perspective in the original scene and also involves different phases like Image Capturing, Marker Tracking and Rendering [2]. An intuitive approach for AR content creation that comprises two steps: creation of 3D models from real objects using an image-based modelling tool (Autodesk Memento, in their case), and conversion of the 3D model into an interactive AR element using an AR authoring tool (Aumentaty Author, in their study) was presented in [3].

A fast marker detection front end for Augmented Reality (AR) applications using an edge-based approach was found in [4]. Styliani Verykokou et al. in their work [5], provides an overview of image matching feature tracking and photogrammetry. The paper used a template matching method combined with a homography-based technique for transfer of the GCPs in another image, along with an incremental photogrammetry-based Structure from Motion (SfM) workflow, coupled with robust iterative bundle adjustment methods that reject any remaining outliers, which have passed through the checks and geometric constraints imposed during the image matching procedure.

Few work has been done in introducing new techniques for object detection and pose tracking with a monocular camera for augmented reality applications. ARtoolkit and ARtag for AR application and provides algorithm for marker detection and pose tracking [6]. The authors has also mentioned different types of image Thresholding and Line detection algorithms which are used in marker detection procedure.

The projection models and pose estimation depending on the shape of objects, few evaluation datasets and evaluation procedures was introduced in [7], The paper also provides a classified and summarized report on the recent progress of detection and tracking techniques. G M Moura et al. in their work [8], provides a comparison and analysis of combinations of the algorithms presented in OpenCV library that allow the detection of pre-established patterns in images and videos and determine the best combinations for Augmented Reality systems.

One of the papers introduced us to Registered Localization Method through OpenCV. It studies the implementation of the camera calibration, the extraction of characteristic target and its recognition. The development of OpenCV applied to Augmented Reality has greatly improved the efficiency of the development and lessened the developmental difficulty, which lays a good foundation for the application of Augmented Reality [9]. An overview of Unity3D tool, the difference between Augmented Reality, Mixed Reality and Virtual Reality, how this is different from a real scene was presented in [10]. It also shows detailed steps to develop the idea using Unity3D and Vuforia.

In their work [11], Hayet Belghit at el. presented the technical classification of most of approaches for 3D pose estimation. They also focused on key-points-based techniques and presented a reach comparison of both detectors and descriptors of the state of the art [11]. A study on different 3D models generated by using four specialized software packages for photogrammetric 3D scanning (Agisoft Photoscan (now available as Agisoft Metashape), CapturingReality RealityCapture, 3DFLOW 3DF Zephyr, AliceVision

Meshroom) and a comparative analyses and visual inspections of the resulting 3D models were provided in [12].

3. PROPOSED METHODOLOGY

In this work, we developed a system which can detect a marker and can place a 3D object on that marker. This task can be divided into three steps:

- **Developing a 3D model:** Developing the 3D model involves Photogrammetry technique which uses photographs as the fundamental medium for metrology (or measurement). By taking photographs from at least two different locations, so-called “lines of sight” can be developed from each camera to points on the object.
- **Detecting the marker:** This step involves feature matching method and key detection method to find the homography with the help of RANSAC algorithm. The detection method works only to track two identical objects.
- **Pose Estimation and placing the 3D object:** This step involves methods such as Render and 3D model loader which is used to place the object on the marker detected. The above can also be implemented with the help of tools like Unity and Vuforia Engine as Unity has built in libraries of OpenCV.

The figure, Fig.1. depicts the process flow of proposed methodology of how a real scene is converted into a 3D model and different AR content creation methodologies to produce an AR scene. The 3D content creation involves capturing the scene as a sequence of photographs, preprocess the images, and model clean up. The 3D model generated is rendered to a AR scene by applying techniques such as target surface recognition, homography estimation, and projection.

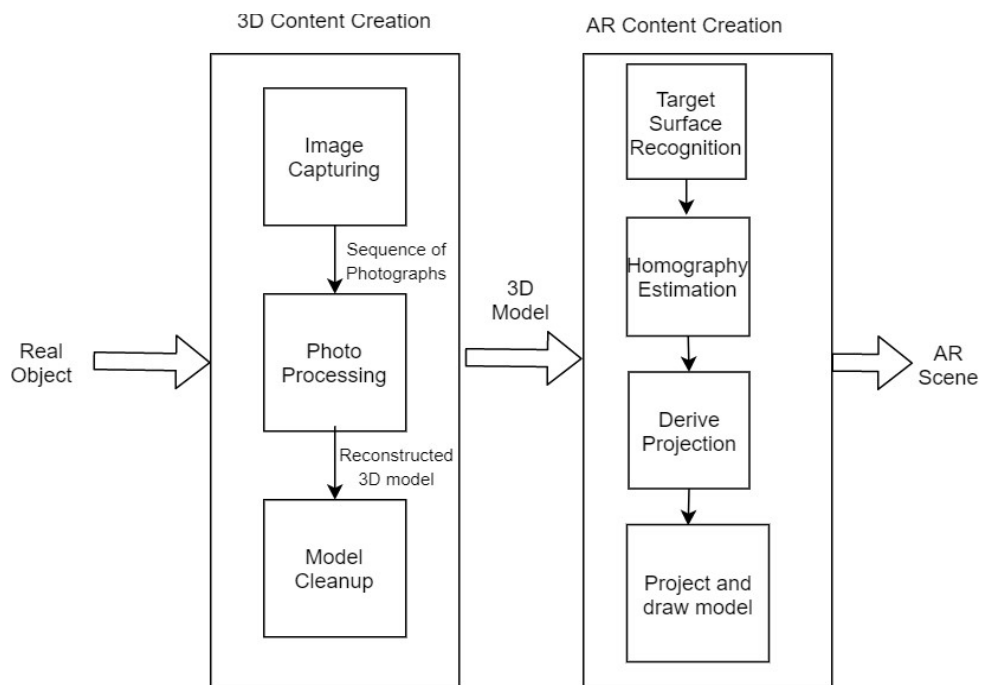


Fig.1. Proposed methodology for AR content creation

4. IMPLEMENTATION STEPS AND RESULTS

The model was developed using two environments – firstly, using OpenCV and Python and secondly, using Unity3D and Vuforia. The following software were used for implementing the work. Meshroom for developing a 3D model, MeshLab for cleaning and reconstructing the model, Python OpenCV library for marker detection and pose estimation. Algorithms such as SIFT, RANSAC were used for feature matching with Homography.

Developing in other environment we installed Unity and Vuforia is installed to implement the Marker Based Augmented Reality. For the development of this model minimum of 8GB RAM and 2GB NVIDIA graphics is required.

Implementation Steps:

To generate a 3D model, an object was selected and around 200 photos were taken from different angles. The dataset thus created was shown in Fig.2.

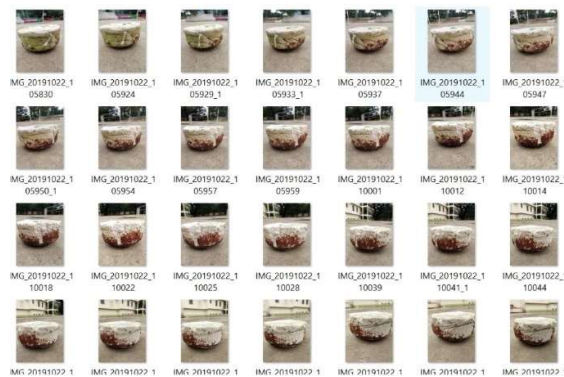


Fig.2. Sequence of images captured for 3D-model generation

4.1.1 Developing a 3D Model:

Meshroom tool was used for 3D model creation. Lot of time is required in DepthMapSection for the formation of point cloud. As the process continues, results can be seen in the 3D viewer. Required 3D model or point cloud or mesh is created. The images are dragged in the left panel in Images section.

The point cloud creation takes lot of time. In our case it took almost 12 hours to build the 3D model. The point cloud or 3D model can be seen in 3D Viewer as depicted in Fig.3.

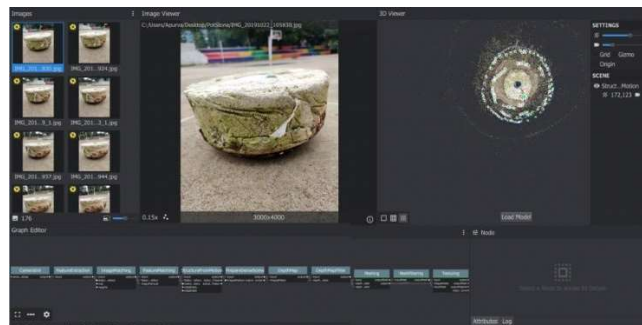


Fig.3. 3D model generated using Meshroom

The following steps were involved to clean and reconstruct the 3d model using MeshLab.

- Import the 3D model or point cloud developed in Meshroom.
- Remove the outliers using selection tool and then delete it.
- Apply Filters to the object to smoothen the vertices and edges.
- The size of the developed model is big, so to reduce it one can reduce the number of faces by Remeshing, Simplification and Reconstruction process.
- The final 3D model was created by going through different filtering, cleaning and editing process.

The raw model and the resultant model generated from MeshLab tool is as shown in the diagram Fig.4.

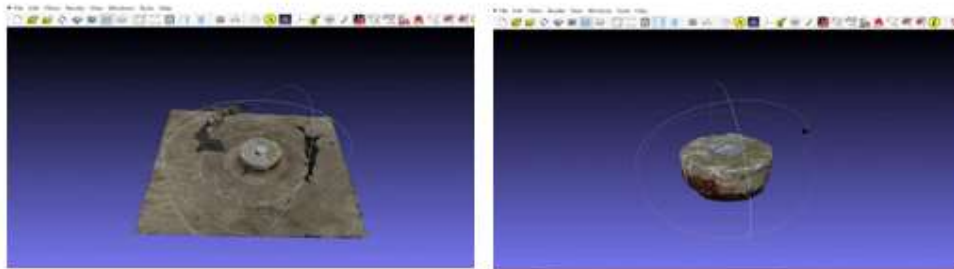


Fig.4. 3D model cleaned and reconstructed using MeshLab

4.1.2 Marker Creation and Detection:

Marker creation involves selecting a target image and a reference image. The following image shown in figure Fig.5. was used as a target or marker image where the 3D model has to be rendered. The reference object is the 3D model developed in step 4.1.1.



Fig.5. Sample image used as image marker/target

The next step involves placing the generated 3D model on the desired position of the target image or image marker. Identifying or recognizing the target surface consists of three steps:

- **Feature Detection/Extraction:** Once the target image where the 3D model will be rendered is identified, the image features unique to the marker has to be extracted from the marker image. The keypoint based image descriptors such as SIFT, SURF or ORB were employed to detect the markers. The following diagram Fig.6. depicts the detected keypoints using SIFT descriptor.
- **Feature Description:** Once features have been found, a suitable representation of the information they provide has to be identified. This will allow us to look for them in other images and also to obtain a measure of how similar two detected features are when being compared. In this work, the following algorithms that extract image features and compute its descriptors such as SIFT, SURF or ORB

were utilized.



Fig.6. Features extracted from sample image

- **Feature Matching:** The extracted keypoint descriptors of target image and reference image were compared to check the valid matches using brute force method. For feature matching, we used descriptors such as SIFT, SURF and ORB. Though there were small difference between SIFT and SURF we obtain better result with SIFT. Expected results were not found with ORB since it considers a smaller number of keypoints. The match between the target image and the reference object is found as shown in Fig.7.

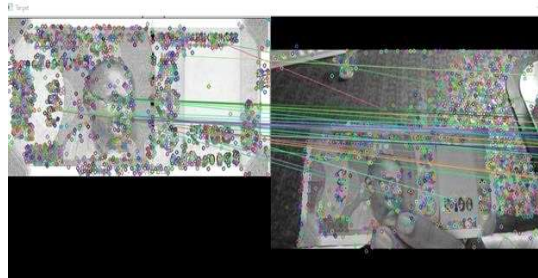


Fig.7. Match between the target and reference images

4.1.3 Pose Estimation and Rendering:

To estimate the pose, the 3D matrix projection, the homography via keypoint matching and RANSAC algorithm were employed. The pose estimation and rendering was implemented using Unity 3D and Vuforia software as well as manually implemented through python OpenCV tool. To estimate the 3D matrix projection following steps are processed: -

- Derive the mathematical model of the projection (image formation). Assume that, at this point, everything is an unknown.
- Heuristically estimate the homography via keypoint matching applying RANSAC algorithm. - H is no longer unknown.
- Estimate the camera calibration matrix. -> A is no longer unknown.
- From the estimations of the homography and the camera calibration matrix along with the mathematical model derived in 1, compute the values of $G1$, $G2$ and t .
- Find an orthonormal basis in the plane ($R1'$, $R2'$) that is similar to ($G1$, $G2$), compute $R3$ from it and update the value of t .
- Model Projection: Load the low poly 3D models based on OBJFileLoader Script. A render method is used for rendering the 3D model. This method includes two things: -
 - The scale factor: Since we don't know the actual size of the model with respect to the rest of the frame, we may have to scale it so that it has the

- desired size. The scale matrix allows us to resize the model.
- We would like the model to be rendered on the middle of the reference surface frame. However, the reference frame of the models is located at the center of the model. This means that if we project directly the points of the OBJ model in the video frame, our model will be rendered on one corner of the reference surface.

To locate the model in the middle of the reference surface we have to, before projecting the points on the video frame, displace the x and y coordinates of all the model points by half the width and height of the reference surface.

The results obtained after 3D projection and rendering is as shown in the diagrams Fig.8 and Fig.9.



Fig.8. Result obtained through OpenCV



Fig.9. Rendering of 3D model through Unity & Vuforia

4.2 RESULT ANALYSIS

The proposed work focused on to build a system that can detect a marker and can place a 3D object on that marker. The model was developed using two environments – firstly, using the built-in tools Unity 3D and Vuforia, the working of the Marker Based Augmented Reality concept was studied. The tools required high processing capabilities and the development of this model required a minimum of 8GB RAM and 2GB NVIDIA graphics. The tools namely, MeshLab and Meshroom was used to develop the 3D model and filter the model.

Secondly, the marker creation and placement of 3D model, was implemented using Python Numpy and OpenCV modules. Algorithms such as SIFT, SURF, ORB and RANSAC were employed for feature extraction and matching with Homography.

To develop the 3D model, 176 images of the same object was taken using a good quality camera in different possible angles. The results obtained for building a 3D Model using Meshroom and MeshLab is projected in Table 1. As seen in the table, the model creation of good quality required

a lot of computing capabilities and resources.

When number of photos taken was high, it took lot of time to develop the mesh but the quality of mesh was considerable. In this work, 176 photos were taken and developed into a 3D model with good quality. Due to increase in number of photos the size of the mesh also increases. With less number of photos system could not develop a mesh.

Table 1. Time taken to build 3D Model using Meshroom, MeshLab

No. of photos	Time Taken (Hours)	No. of faces	Quality	Size (Kb)
176	10	32,09,619	Good	2,38,276
70	1	25,000	Not Clear	5028
176	10	4,999	Clear	808

The matching between the target image and reference image is done through the feature descriptors SIFT, ORB and SURF algorithms through keypoints. The number of keypoints detected and mapped using BruteForceMatcher and FlannBasedMatcher are listed in Table 2 and Table 3 respectively.

As seen from these tables, using ORB produces only less number of keypoints and hence was not considered for further computation. Though there were a very small difference between SIFT and SURF, using SIFT slightly better result was obtained.

Table 2. The keypoints detected and mapped using BruteForceMatcher

Methods	Time Taken (Sec)	No. of Keypoints (Reference Image)	No. of Keypoints (Target Image)	Good Points
SIFT	3.84	2070	1476	124
SURF	2.77	1994	1851	102
ORB	4.48	500	500	5

Table 3. The keypoints detected and mapped using FlannBasedMatcher

Methods	Time Taken (Sec)	No. of Keypoints (Reference Image)	No. of Keypoints (Target Image)	Good Points
SIFT	3.84	2070	1548	140
SURF	2.77	1994	1614	83

After the keypoints are detected a perfect marker is marked to place a 3D object.

5. CONCLUSION

Augmented Reality is one such emerging technology that has huge promises in the different fields in different aspects. The idea is to allow the user to view the virtual object in the real world using an Object based AR system. While it is possible to develop instant Augmented Reality (AR) applications using some out-of-the-box libraries or available tools in the market, the use of Python for AR, while still a bit nascent, is a portal to a plethora of opportunities. At its core, AR

can be said to be basically a Computer Vision problem with some twists, and that is something Python has already well taken care of, with numerous libraries such as OpenCV.

In this work, we have implemented the background process happening in Unity using Python and Open CV to develop a marker-based pop up 3D object. A 3D model was developed using photogrammetry technique with the help of tools such as MeshLab and Meshroom. At first, an attempt was made to develop a 3D model using 80 photos, but failed to get the desired output as the photos were not taken in clear light and object we took was shining. The object was shining so it reflected light and Meshroom was unable to create the 3D model of good quality. Many pictures which reflected light was discarded automatically by the software. By increasing the number of photos and style of taking pictures the required 3D model was obtained.

The 3D model developed was placed on a 2D image marker to create a projected view. To identify and recognize the target marker, keypoints based image feature descriptors were employed. Once the feature matching happens between the target image and the reference frame, a homography matrix is created upon which the 3D object is placed. For feature matching, algorithms such as SIFT, SURF and ORB and RANSAC were used. The SIFT and SURF algorithms provides better results in detecting the keypoints and thus identifying the marker.

REFERENCES

1. Anuroop Katiyar, Karan Kalra and Chetan Garg, "Marker Based Augmented Reality", *Advances in Computer Science and Information Technology (ACSIT) Volume 2, Number 5*; pp. 441-445 April-June, 2015
2. Komal Gangurde, Bhagyashree pardeshi, Dipika Aher, Trupti Bhabad, Prof. C. B. Patil, "Overlaying Virtual Object Into Real World By Using Augmented Reality ", *International Research Journal of Engineering and Technology Volume: 03 Issue: 03 Mar 2016*
3. Jorge D. Camba and Manuel Contero, "From Reality to Augmented Reality: Rapid Strategies for Developing Marker-Based AR Content Using Image Capturing and Authoring Tools", In the proceedings of IEEE Frontiers in Education Conference (FIE), 21-24 Oct. 2015
4. Martin Hirzer, "Marker Detection for Augmented Reality Applications", Technical Report ICG-TR-08/05 Graz, Inst. for Computer Graphics and Vision Graz University of Technology, Austria October 27, 2008
5. Styliani Verykokou, Charalabos Ioannidis, "A Photogrammetry-Based Structure From Motion Algorithm Using Robust Iterative Bundle Adjustment Techniques", *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV- 4/W6, 1-2 October, 2018.*
6. Vipul P.Chauhan , Dr. Manish M. Kayasth, "Object Detection and Pose Tracking in Augmented Reality", *International Journal of Science and Research (IJSR)(2014), Volume 3 Issue 12, December 2014*
7. Hideaki Uchiyama Eric Marchand , "Object Detection And Pose Tracking For Augmented Reality: Recent Approaches", In the proceedings of IEEE Frontiers in Education Conference (FIE), February 2012
8. Gustavo Magalhães Moura, Rodrigo Luis De Souza Da Silva, "Analysis And Evaluation Of Feature Detection And Tracking Techniques Using Opencv With Focus On Markerless Augmented Reality Applications", *Journal of Mobile Multimedia, Vol. 12,*

No. 3&4 (2017) 291–302

9. Jun-Dai, Li-fen-Zhang, “Registered Localization Method-Based On OpenCV For Augmented Reality” In the proceedings of International Conference of Soft Computing and Pattern Recognition (SoCPaR), 14-16 Oct. 2011
10. Dipti Rajan Dhotre, “Research on Object Based Augmented Reality Using Unity3D In Education System”, International Journal of Advance Research and Innovative Ideas in Education, Vol-2 Issue-3 2016
11. Hayet Belghit , Abdelkader Bellarbi , Nadia Zenati , Samir Otmane , “Vision-Based Pose Estimation For Augmented Reality : A Comparison Study”, IEEE International Conference on Pattern Analysis and Intelligent Systems, 25 Jun 2018
12. Ivan Reljić, Ivan Dunder, Sanja Seljan, “Photogrammetric 3D Scanning Of Physical Objects: Tools And Workflow”, Technology, Education, Management, Informatics TEM Journal Volume 8 / Number 2 / 2019.