CSE586/EE554 Computer Vision II

Augmented Reality through Multiview Stereo and SfM

Saniya Naphade, Vishakadatta J H

Contents

1 Introduction

Augmented Reality (AR) is an alteration of Virtual Environments (VE), or Augmented Reality as it is more generally called. VE totally submerges a client inside a manufactured domain. While submerged, the client can' t see the genuine world around him. Interestingly, AR permits the client to see this present reality, with virtual objects superimposed upon or composited with this present reality. Consequently, AR supplements reality, instead of totally supplanting it. In a perfect world, it would appear to the client that the virtual and genuine items existed together in a similar space.This is the actual purpose according to us in the project and this has motivate us to try to achieve a small goal by superimposing a polygon on the given image.

1.1 Motivation

The newly released games such as Pokemon Go [2] is the game filled with AR option where in we have to play with our camera' s switched on and the game character appear on the real time mode of the images being shown on the mobile phone.This made us do a small project where in we can project a complete newly designed polygon the the taken images.

1.2 Related Work

The related works in this project includes the start point of our data collection stage which is the COLMAP [4] .In this paper they do 3D reconstruction from unordered image collection has been used for many applications and implementing it with Incremental Structure-from-Motion is a prevalent strategy[1]. The concern with this strategy is that it is not robust ,accurate ,complete and not scalable which deprives it from building a general-purpose pipeline. This paper aims at solving this by providing a new algorithm SfM technique that improves upon the state of the art. The new algorithm discussed here improves the challenges in SfM as it has a new geometric verification strategy, next best view selection, a robust triangulation method ,an iterative BA and finally a more efficient BA parameterization that would augment the scene graph with information which improves the robustness in initialization and triangulation components. The proposed model was evaluated for state-of-art on challenging large-scale datasets were done and the performance of the individual components mentioned earlier, and the overall system were found to be better.

1.3 COLMAP Software

COLMAP [3] is a broadly useful Structure-from-Motion (SfM) and Multi-View Stereo (MVS) pipeline with a graphical and order line interface. It offers a wide scope of highlights for reproduction of requested and unordered picture assortments.With this software we generated various points and also which formed 3D-point cloud which produces three bin files required for the reconstruction to and they are gotten from the software.

2 Data collection and COLMAP

We collected a set of images that show the table area of a 3D scene. We took these 15 images with a single cell phone camera at a single focal length. Figure 1 shows one example of the image we collect. The internal camera parameters are the same. The dominant planar surface is the table cloth, and this surface has a lot of texture as the figure shows.

After downloading the COLMAP and play with the application. We run feature extraction,

Figure 1: Images collected from a single cell phone camera with different angles

matching, incremental reconstruction, and bundle adjustment to get a 3D cloud of points reconstruction of our image set. We use the default parameter settings. But we found that image 7 is ignored by COLMAP because of the camera position is so different from other camera positions.

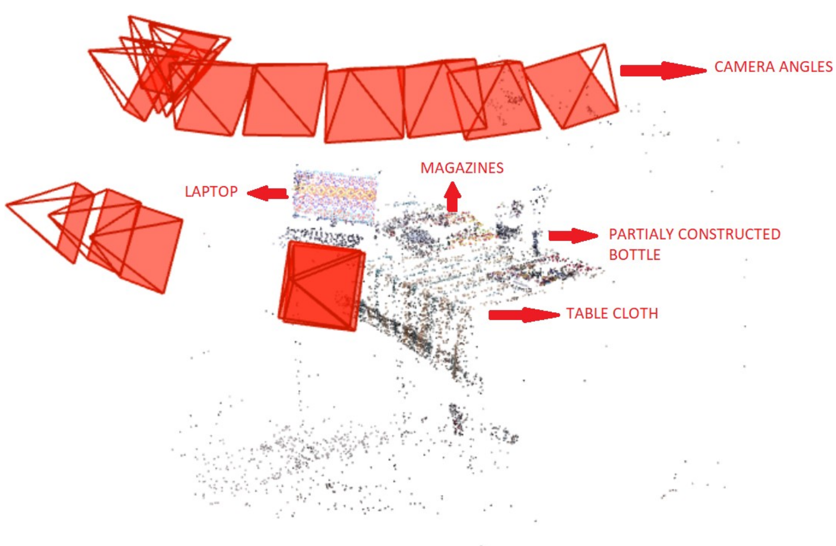

Figure 2: COLMAP reconstruction of the given Input images

3 RANSAC

The RANSAC algorithm is a learning technique to estimate parameters of a model by random sampling of observed data. Given a dataset whose data elements contain both inliers and outliers, RANSAC uses the voting scheme to find the optimal fitting result.
The RANSAC algorithm is essentially composed of two steps that are iteratively repeated:
In the first step, a sample subset containing minimal data items is randomly selected from the input dataset. A fitting model and the corresponding model parameters are computed using only the elements of this sample subset. The cardinality of the sample subset is the smallest sufficient to determine the model parameters.
In the second step, the algorithm checks which elements of the entire dataset are consistent with the model instantiated by the estimated model parameters obtained from the first step. A data element will be considered as an outlier if it does not fit the fitting model instantiated by the set of estimated model parameters within some error threshold that defines the maximum deviation attributable to the effect of noise.
The RANSAC algorithm will iteratively repeat the above two steps until the obtained consensus

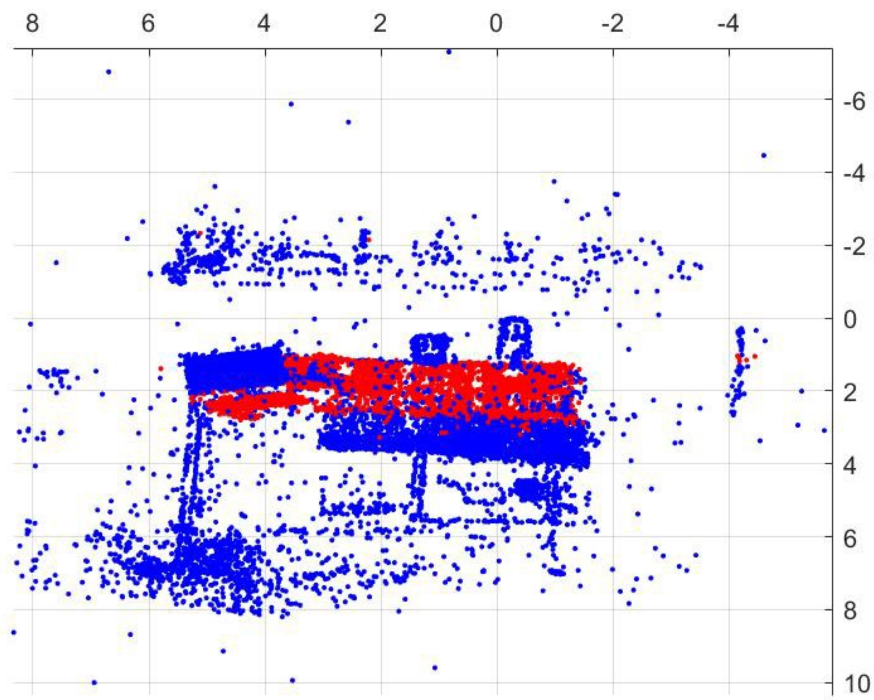

Figure 3: Getting a plane from the RANSAC algorithm and marking the inlier points as red

set in certain iteration has enough inliers. The input to the RANSAC algorithm is a set of observed data values, a way of fitting some kind of model to the observations, and some confidence parameters. RANSAC achieves its goal by repeating the following steps: Select a random subset of the original data. Call this subset the hypothetical inliers. A model is fitted to the set of hypothetical inliers. All other data are then tested against the fitted model. Those points that fit the estimated model well, according to some model-specific loss function, are considered as part of the consensus set. The estimated model is reasonably good if sufficiently many points have been classified as part of the consensus set. Afterwards, the model may be improved by re-estimating it using all members of the consensus set. The threshold of our program is 0.1 and the number of iterations is 10000. We found 2031 inlier points out of all 8794 points. This procedure is repeated a fixed number of times, each time producing either a model which is rejected because too few points are part of the consensus set, or a refined model together with a corresponding consensus set size. In the latter case, we keep the refined model if its consensus set is larger than the previously saved model.

4 Displaying 3D point cloud

By setting the threshold to 0.1,we can divide the points to inlier set and outlier set. Use the scatter3 function we can plot all the data points in the 3D graph, see Figure 2. The blue point is outlier set the while the plane (table plane) consists of red points from inlier set.

5 3D Transformation

The algorithm for construction of the 3D points and the dominant plane was constructed by us.The flowchart of this is shown in the Figure 4. We have taken 3 random points on the plane p1,p2,p3

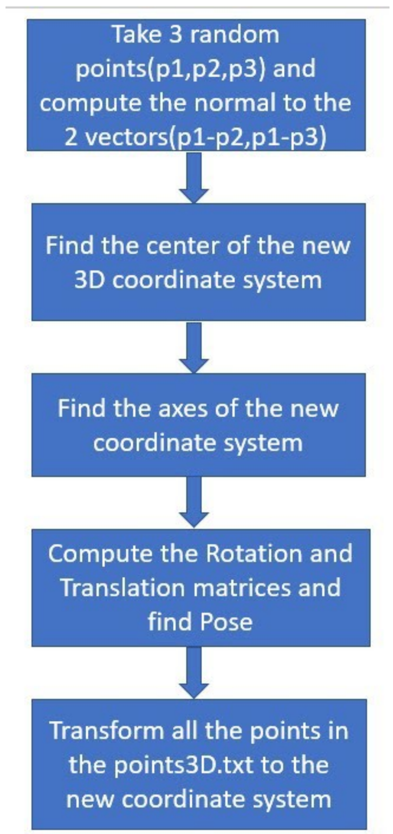

Figure 4: Flowchart for 3D Transformation

and then the computation of the normal was done for them.The computation of the normal is done by the equation as shown in the flowchart.The 2 vectors p1-p2 and p1-p3 was computed and then their normal was taken.
Once we got the normal of these vectors then the center of the entire 3D coordinate system was computed using MATLAB inbuilt functions.

The new axes of the new co-ordinate system was computed as the newly formed coordinate system would want to plot the new points w.r.t to these axes.
The main part of the algorithm is this by finding the rotation matrix .This is done by using the values of the axes of the new coordinate system The dimension of this matrix is 4x3.
Translation matrix is a column matrix formed by using the mean of the in liners the dimension is 4x4

```
X = cross([0 0 1],Z);
X = X/norm(X);

Y = cross(Z,X);
Y = Y/norm(Y);

Z = normal;
Z = Z/norm(Z);

R = [X',Y',Z', [0, 0, 0]'];
R = [R; [0, 0, 0, 1]];
%Dominant plane
pose = R\t;
```

Figure 5: Computation of the Pose matrix

Pose of the matrix is formed using the rotation matrix and the translation matrix.The points in the world coordinate system were transformed into this new coordinate system my multiplying each point by the pose matrix

6 Virtual Object

To create a virtual object, we can first create a virtual object in our coordinates. The 8 points are in the new coordinate. The visualization result for new coordinate is showed in Figure 3. To get the coordinate' s values in the original scene, we apply inverse transformation matrix to the 8 points. The visualization result for old coordinate is showed in Figure 4.

7 Construction of Intrinsic and Extrinsic Parameters Matrices

From the previous section we get the points.txt,images.txt and cameras.txt files for determining parameters such as cameraID,imageID ,height ,width among other parameters.
Using these we will do the 3D-2D conversion of the world points into pixel points. Here we have the u,v,w and also the rotational and translation matrix which the computation of Pose matrix.
We read the external and internal camera parameters from files cameras.txt and images.txt .The output format from the COLMAP website was used to determine the what position of the parameter means what in camera and image dataset.
CAMERA_ID, MODEL, WIDTH, HEIGHT, PARAMS[] are the parameters in camera.txt file and as far as images.txt is concerned we have
IMAGE_ID, QW, QX, QY, QZ, TX, TY, TZ, CAMERA_ID, NAME
POINTS2D[] as (X, Y, POINT3D_ID) by reading all these parameters.We constructed the Rotational matrix and the translational matrix by using the quaternion for the purpose of construction

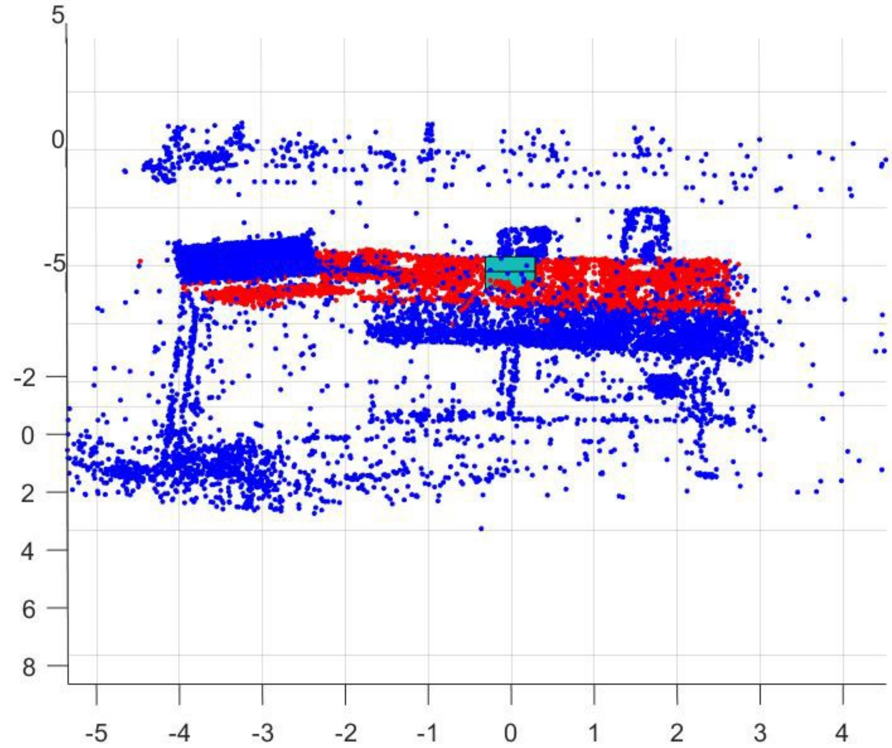

Figure 6: Table plane aligned with the z=0 plane and create a virtual box on the middle of the plane

of the mentioned matrices.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K[R/T] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

These matrices together form the required Pose matrix for the proper reconstruction of the 3D-2D conversion.The matrices as shown earlier have a unique feature and these are used to construct the conversion from 3D-2D.Then the newbox coordinate system was also made by the constructed cube and these will form the new system.The main task after this done by us is that the images are formed and then the projection of the cube was done on the given image and this will form the augmented part of the project then finally the real image would have the same box on the image which would give the effect of the new form of the augmented reality.

8 Object Projection

From section 5 we create a virtual box in new 3D frame and transform it to the old 3D frame. In this section we would like to use the 3D-2D conversion from section 6 to project the virtual box to the 2D pixel points. To do this, we apply equation to the 8 corner points of the box, and get the 2D pixel location of the corners, then we draw the box in 2D frame, making the surface is visible so it would look like a 3D object shown in Figure 5 and 6.

Figure 7: In 2D pixel frame, the virtual box looks on the top of the table.

9 Observations

After developing the project we made few observations that can be improved on.These observations were made in the due course of the project.

- The COLMAP would perform better and produce more defined images if the original images has a lot distinctive characteristics
- Dominant plane will always be the plane having a well defined characteristic
- The construction of Virtual box has to be done in a planar way and not by connecting mere points on the plane
- The mapped virtual box on the real image has to applied with Z-ordering for it have a even more real effect and can blend in the real scene.

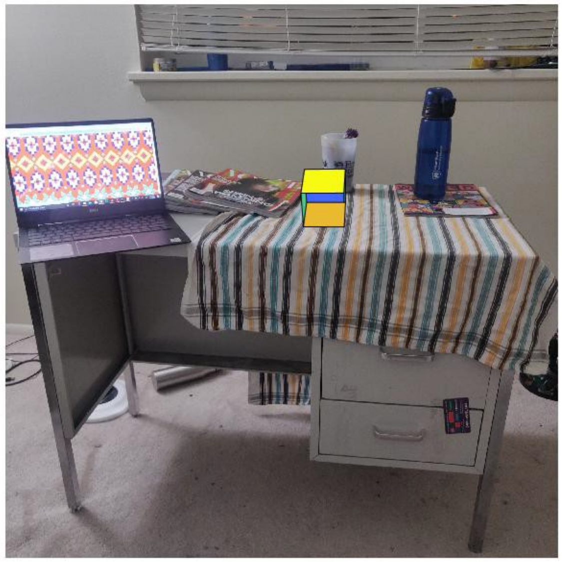

Figure 8: The cube projected on to the image

10 Conclusion

By implementing this project we understood the intricacies involved in the development o f augmented reality.We understood how the intrinsic and the extrinsic impacted on the reconstruction of the image. We successfully placed a virtually created cube on to a real scene using the principles of camera transformation.As mentioned in observation we weren' t able to reproduce the cube completely onto the real scene and this could be done better by using the Z-ordering.This method would make the planes with more points to be plotted from the beginning and then the other planes will be plotted.

11 Contributions

Saniya Naphade	50%
Vishakadatta J H	50%

Table 1: Contributions

References

[1] Inc. Niantic. https://www.pokemongo.com/en-us/. 2020.

[2] Johannes L. Schonberger and Jan-Michael Frahm. https://colmap.github.io/. June 2016.

[3] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.