3D Reconstruction from Multi-View Stereo: From Implementation to Oculus Virtual Reality

Andrew Moran
MIT, *Class of 2014*
andrewmo@mit.edu

Ben Eysenbach
MIT, *Class of 2017*
bce@mit.edu

Abstract

Multi-View Stereo reconstructs a 3D model from a set of images. Each image is a projection of a 3D model onto the camera plane ($\mathbb{R}^3 \rightarrow \mathbb{R}^2$), which inherently results in a loss of information. With enough images taken from a variety of perspectives, a reasonable model of the original scene can be reconstructed. These reconstructions usually begin by determining where each image was taken from. Once the cameras are calibrated, a dense point cloud can be generated. A mesh can be fit over the point cloud to represent structures in the original scene. This entire process can be visualized through an Oculus Rift.

1. Introduction

Most current approaches to Multi-View Stereo can be broken down into three steps: feature matching, camera calibration, and dense reconstruction. In pipelines which result in a point cloud, surface reconstruction is applied as a final step.

1.1. Feature Matching

The first step in most Multi-View Stereo pipelines is finding correspondence points between images. Once we know these points, we can determine the relative position of the cameras which took the images.

First, we select features of interest in each image. There exist numerous algorithms for selecting these points (SIFT, SURF). The general idea behind most of them is to find features which are unique enough such that finding a similar feature in another image indicates with high probability that the two features corresponds to the same object in the scene.[2]

Once features have been extracted from each image, pairwise matches must be found. Matches will not exist for all features, so some criteria must be specified for when to accept a match. One such criteria is to accept a match if the two features are each the best match for the other. Another approach is to match one feature to another feature if the best alternative match is a much worse match than the best match.[2]

1.2. Camera Calibration

Now that we have correspondence points, we want to compute a homography relating one image to another. If we only wanted to find the orientation of one camera relative to another, we could use RANSAC to fit a homography (with the Discrete Linear Transform).[3] However, if we only found the optimal pairwise relative positions, they probably would not be globally consistent.

Instead, we want to find the *global* optimal camera positions. One approach, Bundle Adjustment, incrementally alters the positions of the cameras to as to minimize the *reprojective error* of the found correspondence points with respect to the images in which they appear. This process can be seen as minimizing a series of nonlinear equations; the Levenberg Marquardt Algorithm for nonlinear least-squares is commonly used as a subroutine. At the end of this process, we have a calibration matrix for each camera, relating the pose of each camera to a global coordinate system.[13, 11]

1.3. Dense Reconstruction

We can now reconstruct the scene from calibrated cameras. We want to find eventually output a scene which is *photo consistent*. Common approaches to this problem include: (1) building up a scene from points whose locations are found by triangulating between images; (2) starting with a volume which encloses the region of interest, and removing *voxels* which are not photoconsistent; and (3) generating stereo depth maps for pairs of images, and then fusing them together.[5]

1.3.1 Point Based Approaches

Once we have calibrated cameras with a sparse reconstruction, we can search along equipolar lines to find more correspondence points. The real-world locations of these points

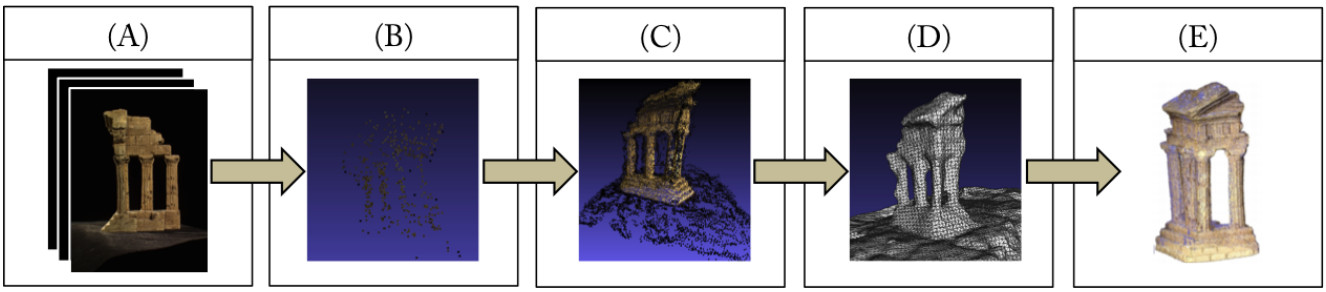

Figure 1. Proposed Multi-View pipeline. (A) Source Images. (B) Bundler Adjustment utilizes feature matching to create a sparse point cloud. (C) Dense Reconstruction develops a more compact point cloud. (D) Surface Reconstruction fits a mesh to the points. (E) Applying colored textures on the mesh results in a realistic simulated model.

can be found through triangulation. In Patch-based Multi-View Stereo, these points were further expanded to *patches* which included a color and normal vector.[5]

1.3.2 Volumetric Approaches

Another approach to dense reconstruction is to start with a volume which encloses the region of interest, and iteratively remove small sections (*voxels*) which are not photoconsistent. Constructing the initial visual hull requires segmenting the input images into foreground and background.[10]

1.3.3 Stereo Depth Approaches

A final approach, builds off research in stereo matching. Here, we generate depth maps for all pairs of images with overlapping fields of views. Methods include SemiGlobal Matching, Graph Cuts, and Dynamic Programming.[6, 12] These depth maps can be fused to extract the structure of the scene.[4]

1.4. Surface Reconstruction

We next want to reconstruct the surface from a finite set of scattered points. This can be viewed as an optimization problem in which the overall goal is to minimize a global energy function. This function often enforces a smoothness constraint and penalizes the distance between points and the mesh.[8] Two common approaches to solve this problem include (1) interpolating the points with computational geometry and (2) constructing implicit functions to fit the data.

1.4.1 Geometric Estimation

In this approach, the point cloud is partitioned into sections and geometric estimations are calculated at each point. For example, each point can be treated as a tangent plane and neighboring points are then projected onto this plane, giving an estimation of the surface point.[15, 7] This utilizes methods such as Delanunay triangles, α-shapes. [15, 9]

1.4.2 Function Fitting

A more rigorous approach is to fit implicit (inside-outside) functions onto the point cloud. This can be based on signed distance or centered radial basis functions (RBFs).[9] Improvements can be made to handle irregularities and outliers with the use of Moving Least Squares (MLS).[14] Finding a continuous function in which the points approximately lie on the zero set determines the smoothed boundary for the mesh. One way this can be completed is by the Marching Cubes Algorithm.[7]

As an example, *Poisson Surface Reconstruction* utilizes the function fitting approach. An inside-outside function is defined by gradients determined from the point normals.[9] In general, when undergoing surface reconstruction, it is important to ensure photoconsistencey is maintained.

2. Our Approach

We will develop a pipeline which uses point-based methods to construct a densely reconstructed point cloud. From there, we will analyze various mesh-fitting surface techniques. We plan to compare the results quantitatively evaluate them qualitatively, using the Oculus Rift. Figure 1 summarizes the overall reconstruction process.

2.1. Visualization

With a 3D reconstruction, it makes sense to try to view it in three dimensions. We do this using the Oculus Rift, a head-mounted stereoscopic display. We will render the scene in a game engine, which will allow users to move around the scene.[1]

2.2. Conclusion

There are many applications and approaches to Multi-View Stereo. We propose a pipeline which transforms a set of images into a model of the scene. Additionally, we propose to visualize our results using an Oculus Rift, which will shed light on the reconstruction process.

References

[1] Unity3d - game engine. http://www.unity3d.com. 2

[2] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73, 2007. 1

[3] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1

[4] S. Fuhrmann and M. Goesele. Fusion of depth maps with multiple scales. In *ACM Transactions on Graphics (TOG)*, volume 30, page 148. ACM, 2011. 2

[5] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1362–1376, 2010. 1, 2

[6] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, 2008. 2

[7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. *Surface reconstruction from unorganized points*, volume 26. ACM, 1992. 2

[8] H. Hoppe, T. Derose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of the 20th annual conference on Computer Graphics and interactive technologies*. ACM, 1993. 2

[9] M. Kazhdan, B. Matthew, and H. Hugues. Poisson surface reconstruction. In *Geometry Processing, 2006 Eurographics Symposium on*. 2

[10] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. 2

[11] M. I. Lourakis and A. A. Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):2, 2009. 1

[12] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002. 2

[13] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM transactions on graphics (TOG)*, 25(3):835–846, 2006. 1

[14] R. Szeliski. *Computer vision: algorithms and applications*. Springer, 2010. 2

[15] R. Tang, S. Halim, and M. Zulkepli. Surface reconstruction algorithms: Review and comparison. In *Proceedings of the 20th annual conference on Computer Graphics and interactive technologies*. ISDE, 2013. 2