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

Anonymous CVPR submission

Paper ID ****

Abstract

Multi-View Stereo reconstructs a 3D model from 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, an 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, colored 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 a feature which is unique enough such that finding a similar feature in another images indicates with high probability that the two features corresponds to the same object in the scene.[1]

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 match two features if the first is the best match for the second, and the second is the best match for the first. Another approach is to match one feature to another feature if the second best match is a much worse match than the best match.[1]

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 to make it linear).[2] However, if we only found the optimal pairwise relative positions, we would not be guaranteed that they would be consistent.

Instead, we want to find the *global* optimal camera positions. This process is known as Sparse Bundle Adjustment, and can been seen as minimizing a series of nonlinear equations. The LevenbergMarquardt for nonlinear least-squares is commonly used as a subroutine. Sparse Bundler 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. 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.[9, 7]

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.[4]

1.3.1 Point Based Approaches

Once we have calibrated cameras with a sparse reconstruction, we can search along equipolar lines to find more corre-

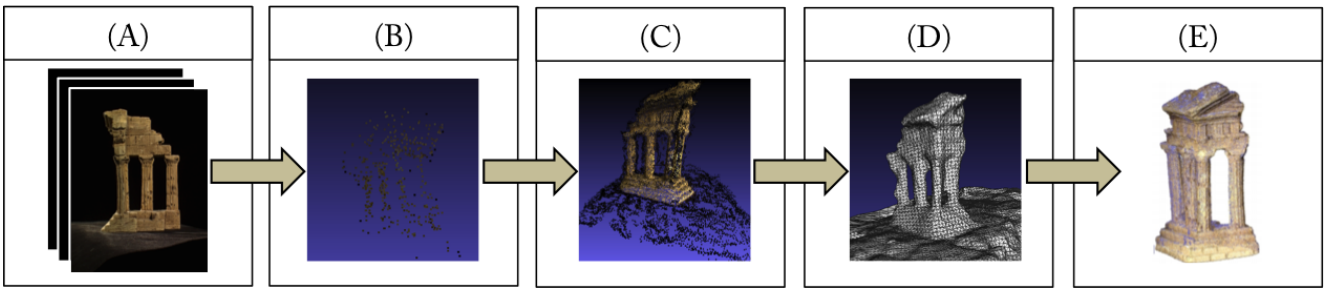

Figure 1. Proposed Multi-View pipeline. (A) Source Images with known camera parameters. (B) Bundler Adjustment that utilizes feature matching to create a sparse point cloud. (C) Dense Recontruction that performs clustering to develop a more compact point cloud. (D) Surface Reconstruction adds a mesh to the points. (E) Applying colored textures on the mesh results in a realistic simulated model.

spondence points. The real-world locations of these points 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.[4]

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.[6]

1.3.3 Stereo Depth Approaches

A final approach is to build 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.[5, 8] These depth maps can be fused to extract the structure of the scene.[3]

2. Surface Reconstruction

After producing a dense reconstruction composed of multiple vertices, the next step is to apply a surface or mesh. Common approaches to undergo surface reconstruction include (1) iteratively using combinatorial structures that interpolate the points; (2) define implicit functions that fit the all points (globally) or a subset of the points (locally); and (3) provide additional methods that augment global/local fitting or utilize segmentation.

2.1. Combinatorial Structures

Several approaches are based on schemes that typically create a triangle mesh that interpolates all or most of the points. Methods include Delaunay Triangulation, α-shapes, or Voronoi diagrams [0]. Extentions on these algorithms have been made to include theoretical gaurantee, such as PowerCrust [4]. If the data is noisy, the resulting surface can be jagged and usaully requires post-processing to refit or smooth the points.

2.2. Implicit Functions

Global fitting techniques commonly define the implicit function as the sum of radial basis functions (RBFs) centered at the points. Ideal RBFs, known as *polyharmonics*, often result in non-sparse solutions. The Multipole Method and The Marching Cube Algorithm are examples of extracting surfaces from the RBF function [0].

Local fitting methods consider subsets of nearby points at a time. The main idea is to estimate tangent planes and define the implicit function to be the distance from this plane to the closest point. Moving Least Square (MLS) is a common approach to blend nearby points together [0]. Another approach includes subdividing the space with an adaptive octree [0]. This often involves multi-level paritioning and heursitics.

2.3. Other Methods

Addtional methods for surface reconstruction usually combine the benefits of both the global and local fitting schemes. *Poisson Surface Reconstruction* converts a set of oriented points into a triangulated mesh model [0]. No heuristics are needed to form local neighborhoods, patch types, etc. Also, the basis functions are assoicated with ambient space rather than data points.

Iterative Snapping utilizes segmentation information to initailize a mesh [0]. A visual hull model is initially generated and then iteratively deformed towards reconstructed patches. Additional algorithms for surface recontructions usually extend the above methods.

3. Our Approach

There are various algorithms and tools in the computer vision community designed for Multi-View Stereo. Our approach consists of utilizing point-based methods to develop a densely reconstructed point cloud. From there, we will

compare various mesh-fitting surface techniques and potentially implement our own, such as Poisson Surface Reconstruction. We plan to visualize our results from each step.

3.1. Proposed Pipeline

Figure 1 summarizes our pipelined approach of displaying a 3D model from 2D images. After extracting features and camera calibration parameters from imported source images, point cloud reconstructions can be created from Surface-from-Motion (SfM) tools. We will utilize robust computer vision packages such as Bundler and CMVS/PMVS [0]. As a result, we will attempt to compare surface reconstruction techniques that will be used to fit a detailed mesh to the point cloud. 3D editing software tools such as MeshLab and Blender[0] can assist in further configuration and production of desired meshes/textures. These can be visualized in the Unity3D game engine, which supports Oculus Rift OVR integration.

3.2. Visualization

With a complete 3D reconstruction, the objective is to then use innovative technology to view and interact with the final scene/model. The Unity3D Game Engine is the utlimate tool for video game development, architectual visualizations, and interactive media installations [0]. By simulating a virtualized environment, the user can continue to explore/examine the results. As an extension, Unity3D has a fully integrated plugin for the Oculus Rift. This developemnt tool is an augmented reality head-mounted display that creates a stereoscopic 3D view. The rift in conjunction with the game engine can produce an immersive setting that enhances the user experience.

3.3. Conclusion/Motivation

There are many applications to Multi-View Stereo. They include (but are not limited to) reverse engineering, industrial design, performance analysis and simulations, realistic virtual environments, and medical imaging. Converting a series of 2D images to a 3D model can be a challenging process. However, feature matching and dense/surface reconstruction are techniques that produce appealing results. Visualizing our results using enhanced 3D tools such as Unity3D and Oculus Rift will assist in understanding and recognizing the 3-Dimensional product.

References

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

[2] 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

[3] 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

[4] 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

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

[6] 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

[7] 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

[8] 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

[9] 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