3D patch-based multi-view stereo for high-resolution imagery

Shizeng Yao, Hadi AliAkbarpour, Guna Seetharaman, Kannappan Palaniappan

Shizeng Yao, Hadi AliAkbarpour, Guna Seetharaman, Kannappan Palaniappan, "3D patch-based multi-view stereo for high-resolution imagery," Proc. SPIE 10645, Geospatial Informatics, Motion Imagery, and Network Analytics VIII, 106450K (27 April 2018); doi: 10.1117/12.2309806

SPIE.

Event: SPIE Defense + Security, 2018, Orlando, Florida, United States

3D Patch-Based Multi-View Stereo for High-Resolution Imagery

Shizeng Yao[a], Hadi Ali Akbarpour[a], Guna Seetharaman[b], and Kannappan Palaniappan[a]

[a]Dept. of EECS, Univ. of Missouri-Columbia, Columbia, MO, USA
[b]U.S. Naval Research Lab. USA

ABSTRACT

This paper proposes an improved solution to image-based three-dimensional (3D) modeling (also known as "multi-view stereo") that outputs surfaces visible in high-resolution wide-area format video also known as wide-area motion imagery (WAMI) consisting of a dense set of small 3D points. The improved approach, named 3D patch-based multi-view stereo, is an expansion of PMVS[1] and is implemented also as a match, expand, and filter procedure. This approach takes a sequence of image frames and corresponding camera parameters together with a sparse set of matched feature points. As an initial step, it formulates a small 3D patch for each of the matched feature points. It then finds the best fitted curved surface inside the 3D patch based on the photometric consistency of each 3D point inside. Expansion and filtering procedures are then recursively applied on those initial surfaces until a certain percentage of image coverage is achieved. The proposed solution is able to precisely preserve small details and automatically detect and discard outliers. Moreover this approach does not require any initialization in the form of a visual hull, a bounding box, or valid depth ranges. We have tested our algorithm on various data sets including single object with fine surface details, and outdoor occluded extremely large WAMI dataset, where moving or static obstacles appear in front of static structures of interest and large areas of repetitive texture are present.

Keywords: 3D Patch-Based MVS, WAMI, 3D Reconstruction, Photometric Consistency

1. INTRODUCTION

Nowadays, Three-dimensional (3D) modeling is an important element to automatically obtain intuitive 3D appearance of geometric object and scene models from multiple photographs or sequential video frames. Many advanced applications require the third dimension to apply better information analysis, including 3D image processing, digital photography, multimedia, 3D visualization, and augmented reality. Based on a survey provided by Georgios et al.,[2] state-of-the-are multi-view stereo reconstruction algorithms achieve relative accuracy better than 1/200 (1mm for a 20CM wide object) from a set of low-resolution (640*480) images. They can be classified into 3 methods: (1) visual hull reconstruction algorithms[3,4,5] can generate full 3D reconstruction of dynamic scenes, but they lack in reconstruction fidelity and are very sensitive to errors in silhouette extraction. (2) Space carving reconstruction methods[6,7] could reconstruct 3D models from multiply images, but they are sensitive to noise and outliers and may yield to noisy reconstructions. (3) The third class of methods[8,9,10] optimize the surface integral of a consistency function over the surface shape, which are simple, effective and robust due to combining both 3D data and 2D image information.

In this paper, we propose an improved method using the third class of algorithms. The improved approach, named 3D patch-based multi-view stereo, is an expansion of PMVS[1] and is implemented also as a match, expand, and filter procedure. This approach takes photographs of small objects or sequential video images of WAMI dataset and corresponding camera parameters together with a sparse set of matched feature points as inputs. As an initial step, it formulates a small 3D patch consisting of a set of 3D points for each of the matched feature points. It then finds the best fitted curved surface inside the 3D patch based on the photometric consistency of each 3D point. Expansion and filtering procedures are then recursively applied on those initial surfaces until a

Further author information: (Send correspondence to Shizeng Yao)
Shizeng Yao: E-mail: syyh4@mail.missouri.edu, Telephone: 1 573 818 5405

Geospatial Informatics, Motion Imagery, and Network Analytics VIII, edited by Kannappan Palaniappan, Peter J. Doucette, Gunasekaran Seetharaman, Proc. of SPIE Vol. 10645, 106450K · © 2018 SPIE
CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2309806

certain percentage of image coverage is achieved. Although our 3D patch-based algorithm is similar to method proposed by Yasutaka and Jean,[1] it replaces their 2D patch by 3D curved plane, which allows us to reconstruct fine details more effectively. Like PMVS,[1] our algorithm does not require any type of initialization such as visual hull model, a bounding box, or depth ranges. As shown by our experiments, the proposed algorithm could effectively handle small details as well as large city-scale reconstructions.

The rest of this article is organized as follows: Section 2 presents the 3D patch-based curved surface generating algorithm. Section 3 presents our 3D patch-based multi-view stereo reconstruction pipeline. Experimental results and discussions are given in Section 4, and Section 5 concludes the paper with some future work.

2. 3D PATCH-BASED CURVED SURFACE GENERATING

This section describes our novel 3D patch-based curved surface generating algorithm.

In PMVS,[1] a patch, which is a local tangent plane approximation of a surface, was used to represent the surface. This algorithm is simple and effective but in certain situations it may not be able to reconstruct small fine details based on our experiments. In our algorithm, we replace the local tangent plane by a curved surface consisting of 3D points such that it could represent the real surface in a better quality.

At the first step, for each matched feature point, we create a small 3D patch for it in 3D space, which consists of $\alpha * \alpha * \alpha$ of 3D points (α is 5 in our experiments). The center of this 3D patch is the triangulation point from all the matched feature points (SIFT) from reference image (R) and all the visible images (V) in which the feature point is visible. The 3D patch is then oriented around its center such that one of its faces is parallel to the current reference image and one of its edges is parallel to the x-axis of the reference image (see Fig. 1(a) for an example). In our experiments we treat an image as a visible image as long as the angle between the normal of reference image and the normal of that image is below $\pi/6$.

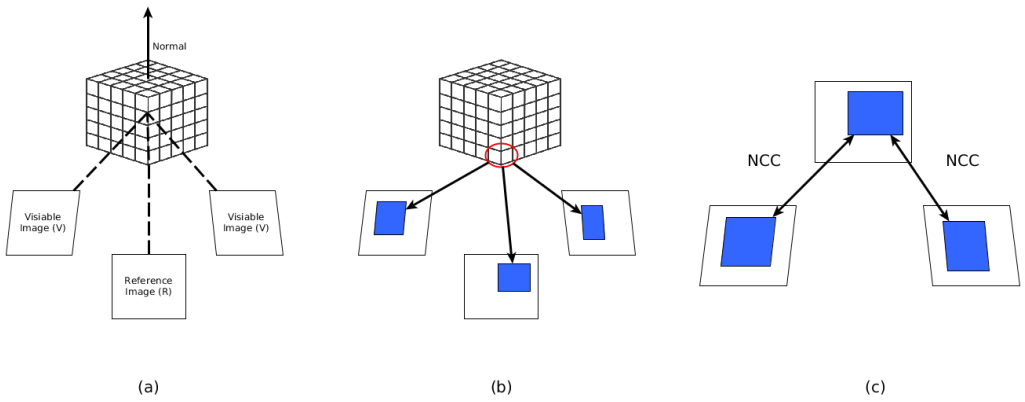

Figure 1. (a) A 3D patch consists of $\alpha * \alpha * \alpha$ of 3D points (α is 5 in our experiments). The center of the 3D patch is the triangulation point from all the matched SFT feature points from reference image (R) and all visible images (V). (b) Each 3D point is then back projected to the reference image (R) and all the visible images (V), and a small 2D patch is obtained from each of the images. (c) Normalized cross correlation is then applied between the 2D patch from reference image and each of the 2D patches from the visible images.

Then for each point inside this 3D patch, we project it back to the reference image and all the visible images. A small 2D patch is then obtained from each of the images (see Fig. 1(b) for an example). We then apply normalized cross correlation between the 2D patch from reference image with each of the 2D patches from visible images such that each visible image has a photometric consistency score for current 3D point (see Fig. 1(c)). A overall photometric consistency score P for that 3D point is then computed using formula 1.

$$P = \frac{1}{|V|} \sum_{V} ncc(R, V) \tag{1}$$

After computing photometric consistency scores for all the points inside the 3D patch, we rank all the 3D points based on their overall photometric consistency scores and only keep the top 20% of those 3D points. At last, a best fitted plane is generated to contain the remaining 20% 3D points.

3. 3D PATCH-BASED MULTI-VIEW STEREO RECONSTRUCTION PIPELINE

This section presents our 3D patch-based multi-view stereo reconstruction pipeline. Our 3D patch-based MVS algorithm attempts to reconstruct a 3D curved plane for each feature point, then expansion and filtering procedures are recursively applied on those initial planes until a certain percentage of image coverage is achieved. The whole pipeline contains four key elements: (1) feature detection and feature matching, (2) initial curved planes generating, (3) plane expansion, and (4) plane filtering. Information flow from input images to the reconstructed point cloud is illustrated in Fig. 2.

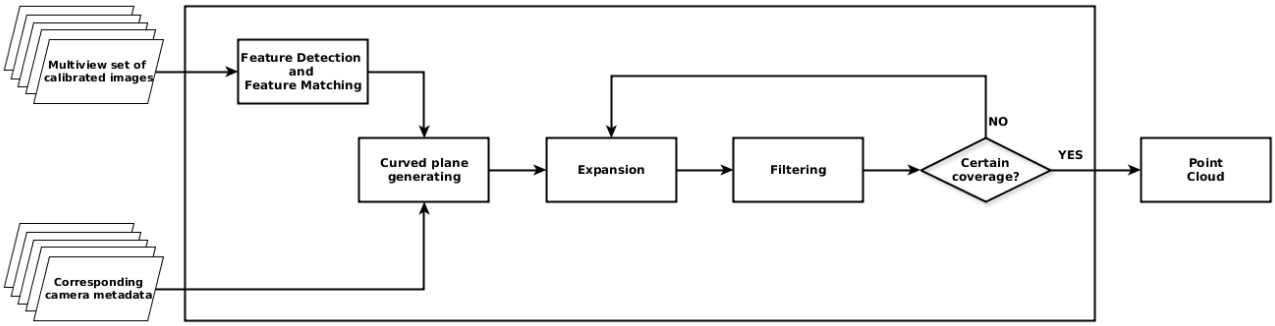

Figure 2. Flow diagram for the proposed multi-view stereo algorithm (3D patch-based MVS reconstruction)

3.1 Feature Detection and Feature Matching

We firstly detect feature points on each image frame using scale-invariant feature transform (SIFT) detector. Secondly, we apply SIFT descriptor to match feature points between image frames to form tracks of matched feature points. All the tracks are stored in a feature list.

Figure 3. WAMI dataset and one feature point from SIFT detector

For each track of feature points, we treat the feature point in the middle of this track as the current feature point, and treat the image on which the current feature point is locating as the reference image (R). Then we

treat an image as a visible image (V) as long as on which a matched feature point is present, and the angle between the normal of reference image and the normal of that image is below $\pi/6$.

3.2 Curved surface Generating

For each track of feature points, we use all matched feature points from reference image and all visible images to perform triangulation and obtain a triangulated 3D point, which is the center of our later 3D patch. After obtaining the center of the 3D patch, We expand this 3D point to a $\alpha * \alpha * \alpha$ 3D patch (α is 5 in our experiments). After having the 3D patch, we apply our curved surface generating algorithm to it and create a small curved surface for current feature point (see section 2). Once a curved surface is generated from a track of feature points, all the feature points on that track are removed from feature list.

3.3 Expansion

Lacking of neighbor information in 3D space is always a challenge for MVS algorithms. In our 3D patch-based MVS algorithm, we use the slope of current curved surface to predict the location of neighbor surfaces. More concretely, given a curved 3D surface, we firstly compute the slopes of the surface in all possible directions. Then we create a 3D point along each possible direction at a distance $\alpha*\beta$ to current 3D patch center (β is 1.5 in our experiments to avoid overlapping). All the new points are then treated as new 3D patch centers. The reference image and visible images of current 3D patch are treated as reference image and visible images for all the newly created 3D patches (see Fig. 4). All the newly created 3D patches are then oriented and applied curved surface generating algorithm to form more surfaces.

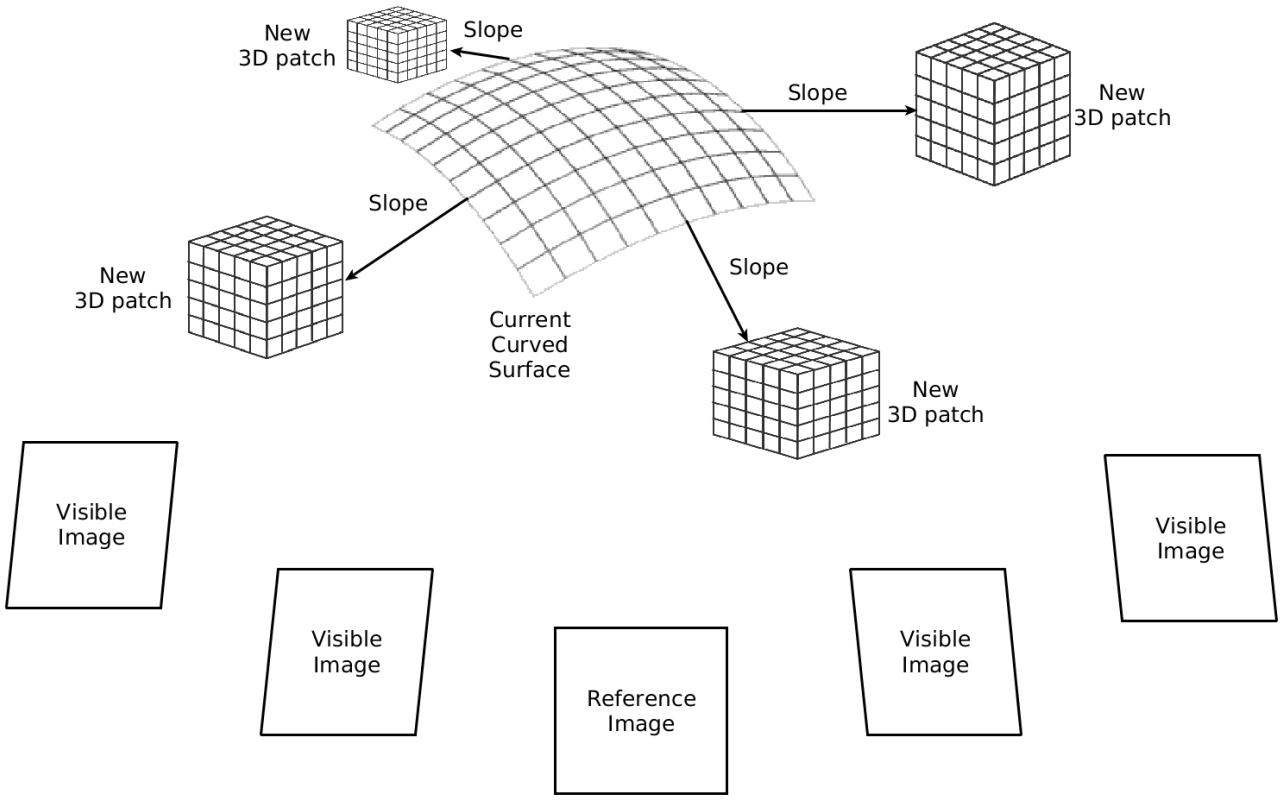

Figure 4. Given an existing curved surface, an expansion procedure is performed to generate new 3D patches and then further new surfaces

3.4 Filtering

During the expansion procedure, a large amount of false positive surfaces will be generated, and to remove those surfaces, we apply three filters in our experiments, which are *normal filter*, *neighbor knowledge filter*, and *occlusion filter*.

The first filter we apply in our experiments is the *normal filter*. In this filter, for each existing 3D point, we firstly find its 10 nearest neighbors in 3D space to form a tangent surface and compute a normal for that surface, which will be treated as the normal of this 3D point. After obtaining the normals for all existing 3D points, we compare the normal of each point with its 20 neighbors again, and if fewer than 5 neighbors have similar normals (angle between the current point's normal and a neighbor point's normal is less than $\pi/12$), then the current point is treated as a false positive and will be removed at the end of this procedure (see Fig. 5 (a)).

The second filter we apply in our experiments is the *neighbor knowledge filter*. For each 3D point, we project the whole point cloud back to its reference image. Then based on the back projection, we determine the 2D neighbor information based on the 2D pixel distance. More concretely, if the distance between the 2D projection location of a certain 3D point and the projection location of current 3D point is less than σ pixels (σ is 2.5 in our experiments), that certain 3D point is treated as a "2D neighbor" of current 3D point. Among all the 3D points who are "2D neighbors" of current 3D point, we then compute the proportion of 3D points that are both "2D neighbors" and "3D neighbors" to current 3D point (if the distance between a certain 3D point and current 3D point in 3D space is less than $\sigma * 0.2$, then it is considered as a "3D neighbor"). If the proportion is less than 25%, then the current 3D point is treated as a false positive and will be removed at the end of this procedure (see Fig. 5 (b)).

The last filter we apply in our experiments is the *occlusion filter*. In this filter, we check the occlusion information for each point and if a certain 3D point is not occluded in an image frame, we mark it as a "really visible" point. After checking all of the image frames, we remove the points which are never marked as "really visible".

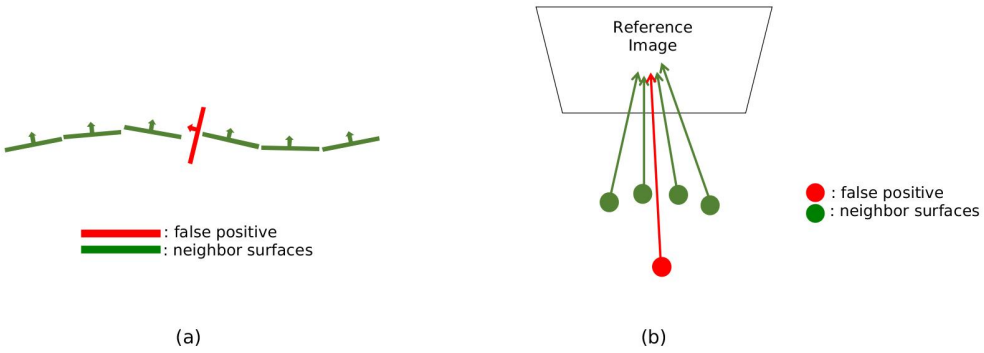

Figure 5. (a) normal filter: an arrow represents the normal of a 3D point and a bar represents a tangent surface of a 3D point. If a 3D point has different normal compared with its neighbors, then it is considered as false positive. (b) neighbor knowledge filter: an arrow represents the back projection of a 3D point and a dot represents a 3D point. If a 3D point has too different 2D-3D neighbor information, then it is considered as false positive. In both (a) and (b), red color represents a false positive 3D point and green color represents neighbor 3D points.

3.5 Image Coverage Testing

After applying the three filtering procedures, we apply our image coverage test to check whether the current point cloud is covering certain proportion area of all the image frames or not. If the point cloud has already covered certain proportion of all the image frames, then it will be treated as our final point cloud. Otherwise, we will feed it into our expansion and filtering procedures again to achieve more image coverage. More specifically, we back project the current point cloud into all the image frames, and for each image frame we compute the

proportion of pixels that are occupied. In our experiments, if more than 60% of all pixels are occupied in certain image frame, then that image frame is considered as "covered". After obtaining the information from all the image frames, if more than 60% of all image frames are "covered", then we consider the point cloud as our final output. If less than 60% of all image frames are "covered", then we will feed the current point cloud into our expansion and filtering procedures again.

4. EXPERIMENTS AND DISCUSSION

4.1 Data Sets

Fig. 6 shows sample input images in our experiments. Left image is from a WAMI dataset, which was collected around downtown area in Albuquerque, New Mexico by Transparent Sky.[11] This is a particularly challenging example since it contains large amount of occlusions, motion changes, and light condition changes. Right image is from *dino*, which was published by S. Seitz et al.[12] In our experiments, we only used 24 images in a ring orbit from *dino* dataset.

Figure 6. Sample input images. Left image is downtown area in Albuquerque, New Mexico. Image resolution is 6600*4400. Right image is *dino*. Image resolution is 640*480

4.2 Results

Fig. 7 shows the intermediate results ((a), (b), and (c)) and the final result ((d)) for *dino* dataset. Fig. 7(a) shows that the surfaces generated from the initial feature matching step are noisy and very sparse, and only contain certain feature points. Fig. 7(b) and Fig. 7(c) show that after each expansion and filtering step, new surfaces are created and noisy surfaces are filtered out. Fig 7(d) shows that after three expansion and filtering steps, the certain percentage of image coverage is achieved and the final result is exported.

Fig. 8 shows the initial result after feature matching step (a) and the final result (b) for WAMI dataset. Since this WAMI dataset contains 215 image frames with really high image resolution (6600*4400), the whole pipeline applies expansion and filtering steps 10 times to achieve the required certain percentage of image coverage. Fig. 8(b) shows that the final result is able to reconstruct the whole city including small fine details.

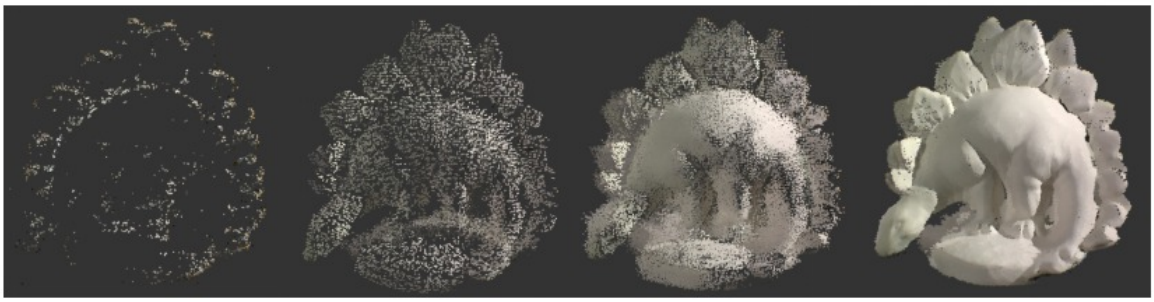

Figure 7. Intermediate and final results from *dino* dataset. (a) Initial feature points. (b) After the first expansion and filtering. (c) After the second expansion and filtering. (d) After the third expansion and filtering, certain percentage of image coverage is achieved, program stops and final result is exported.

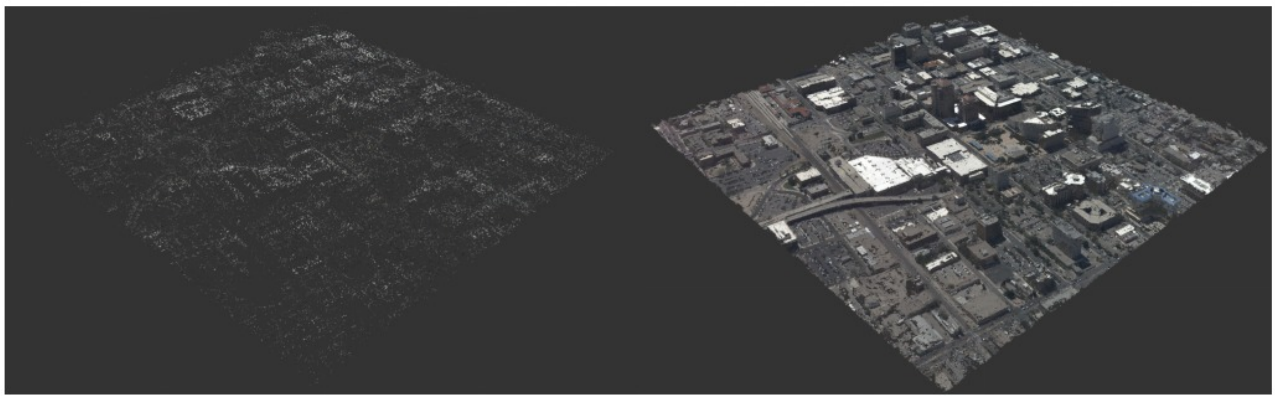

Figure 8. Initial and final results from Albuquerque dataset. (a) Initial result after feature matching step. (b) After the 10th expansion and filtering, certain percentage of image coverage is achieved, program stops and final result is exported.

4.3 Discussion

As mentioned earlier, one difference between the proposed method and PMVS[1] is that our method uses curved planes to represent the surface on a physical model, which helps in reserving fine details. Fig. 7 and Fig. 8 already demonstrate that our 3D patch-based MVS algorithm is capable of effectively reconstructing a physical model from a set of sequential calibrated photographs. However, our experimental results show that reliable texture information is critical for our approach. Large amount of repetitive textures will dramatically affect the quality of our final results. Also our proposed method suffers from execution time. Since for each 3D point, we compute photometric consistency, execution time is longer than PMVS.[1]

5. CONCLUSION

In this paper we have proposed an improved algorithm for calibrated multi-view stereo that outputs a dense set of curved surfaces covering an object or a scene observed by multiple calibrated photographs. Our algorithm starts by detecting feature points in each image frame, then matches feature points across multiple image frames

to form initial 3D patches. Each 3D patch is then verified based on photometric consistency to form initial curved surfaces. Then our algorithm uses expansion procedure to obtain denser point cloud and filtering procedures to eliminate false surfaces recursively until a certain percentage of image coverage is achieved. The proposed algorithm does not require any form of initialization, such as a visual hull model or depth range information. Our results show that the proposed approach is capable of effectively reconstructing a physical object model or a scene. Our future work will be focusing on better performance in terms of execution time and accuracy. Parallel execution and GPU-based computation will dramatically shorten the required execution time, which will bring us higher efficiency.

REFERENCES

[1] Furukawa, Y. and Ponce, J., "Accurate, dense, and robust multiview stereopsis," *Pattern Analysis and Machine Intelligence* **32**, 1362–1376 (2010).

[2] Georgios Kordelas, J.Perez-Moneo Agapito, J. H. and P.Daras, "Stateof-the-art algorithms for complete 3d model reconstruction," *Engage Summer School* (2010).

[3] Tomoya Ishikawa, K. Y. and Yokoya, N., "Real-time generation of novel views of a dynamic scene using morphing and visual hull," *International Conference on Image Processing* (2005).

[4] Grauman, Kristen, G. S. and Darrell, T., "A bayesian approach to imagebased visual hull reconstruction," *Computer Vision and Pattern Recognition* (2003).

[5] PoLun, L. and Yilmaz, A., "Shape recovery using rotated slicing planes," *Image and Signal Processing* (2009).

[6] Kutulakos, K. N. and Seitz, S. M., "A theory of shape by space carving," *International journal of computer vision* **38**, 199–218 (2000).

[7] Rahul Bhotika, D. J. F. and Kutulakos, K. N., "A probabilistic theory of occupancy and emptiness," *European conference on computer vision* (2002).

[8] Jean-Philippe Pons, R. K. and Faugeras, O., "Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score," *International Journal of Computer Vision* (2007).

[9] M, L. and L., Q., "A quasi-dense approach to surface reconstruction from uncalibrated images," *Pattern Analysis and Machine Intelligence* (2005).

[10] Tran, S. and Davis, L., "3d surface reconstruction using graph cuts with surface constraints," *European conference on computer vision* (2006).

[11] "Transparent sky, llc." http://www.transparentsky.net/.

[12] Steven Seitz Brian Curless, James Diebel, D. S. and Szeliski, R., "Multi-view stereo evaluation," (2010).