Large Scale Scenes Reconstruction from Multiple Views

Shi Limin, Zhang Feng, Xu Zhenhui, Hu Zhanyi
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of sciences
Beijing, china
{Lmshi, fzhang, zhxu, huzy}@nlpr.ia.ac.cn

***Abstract*—In this paper, we present a novel method to reconstruct the large scale scenes from multiple calibrated images. It first generates a quasi-dense 3D point cloud of the scene by matching key points across images. Then it builds a tetrahedral decomposition of space by computing the 3D Delaunay triangulation of the 3D point set. Finally, a triangular mesh of the scene is extracted by labeling Delaunay tetrahedra as inside or outside. A globally optimal label assignment is efficiently found as a minimum cut solution in a graph. Experimental results demonstrate the effectiveness of the proposed algorithm.**

Keywords-reconstructiont; Delaunay triangulation; Voronoi diagram; graph cut

1. INTRODUCTION

A fundamental problem in computer vision is to reconstruct the shape of a 3D scene from multiple photographs. And most existing multi-view stereovision approaches require some knowledge of the scene extent and often even of its approximate geometry. In these methods, using visual hull or silhouettes is a major line of research in computer vision. The visual hull is defined as the intersection of cones generated by the silhouettes of the objects in the input views [3]. In the last few years, a number of multi-view stereovision algorithms exploiting visual hull have been proposed. They rely on visual hull either as an initial guess for further optimization [4, 5, 6], as a soft constraint [10] or even as a hard constraint [4] to be fulfilled by the reconstructed shape. This technique requires an accurate segmentation of input images. It makes these approaches mainly suited to compact objects admitting a tight enclosing box, imaged on a simple or a known background. But for most of the real-life examples, however, such segmentation is not available or even feasible, specially for the large scale scene.

Volumetric multi-view stereovision method is another kind of major methods, including space carving [7, 9, 11], level sets [8, 12], and volumetric graph cuts [13, 5, 15]. Such methods need a regular decomposition of the domain into voxels. Obviously, this kind of methods is mainly suited to compact objects admitting a tight enclosing box, as its computational and memory cost quickly becomes prohibitive when the size of the domain increases.

Large-scale scenes for which no reliable initial guess of geometry is available also disqualify the deformable model framework [10, 12, 15]. As a result, it is highly sensitive to initial conditions.

The multi-view stereovision methods which have proven more adapted to reconstruct large-scale scenes (e.g. outdoor architectural scenes) are those representing geometry by several depth maps [16, 17, 18]. However, their performance for complete reconstruction seems to be lower than previously discussed approaches, either in terms of accuracy or the completeness of the obtained model. Moreover, in the complete reconstruction case, the several partial models of the scene have to be fused at post-processing step using a volumetric technique [14].

Patrick Labatut et al. [19] use Delaunay triangulation and graph cuts to reconstruct the large scale scene, and good results are obtained in their experiments. Their method does not require any knowledge of the extent of the scene, and can deal with large-scale scenes at a reasonable computational cost. It can also handle either closed or open scenes. But when the many noisy points are present, it is hard to choose proper parameters of the energy function to get satisfactory results.

In this paper, we present a new approach that also uses the Delaunay triangulation and graph cuts to label the tetrahedra. So our method also has the virtue of the method in paper [19]. In addition, compared to the method in [19], our approach is more efficient and robust.

1. More efficient.

We decompose the tetrahedra into two sets, and label them respectively. Then the min cut computing is on a smaller graph than that of paper [19]. So the total efficiency is higher.

2. More robust.

We use the pole which is more reasonable for guiding the position of surface rather than the visibility information coming from keypoints as in [19] which is not robust especially for noisy point set, see figure 1. Moreover, to deal with the noisy points, we add the restriction (2) or (4). All these disposals make our method more robust and get more satisfactory results in practical applications.

The remainder of this paper is organized as follows. Section 2 gives some important definitions in this paper. In section 3, we describe our reconstruction method in details, including interest point detectors, Delaunay triangulation and graph cuts. Section 4 presents some numerical experiments to demonstrate the potentials of our approach for reconstructing large scale scenes from real-world images.

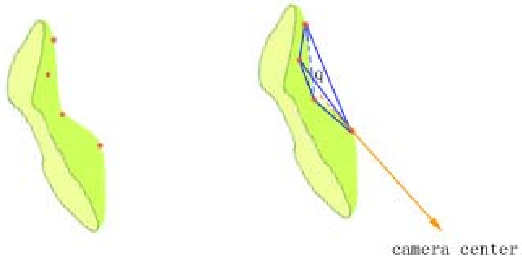


Figure 1. The left picture is a local surface patch. The Four red points are located on the patch. Using the method in [19], the tetrahedron q is very probably labeled inside, see right picture. But it is outside actually.

2. DIFINITIONS

2.1 Voronoi disgram

Let P be a set of points in R^d. The Voronoi cell associated to a point p□P, denoted by *V(p)*, is the region of space that is closer from p than from all other points in P :

$$V(p) =\{q\in R^d | \forall\ p,p'\in P,\text{and } p \neq p', ||q-p'|| \geq ||q-p||\}$$

The Voronoi diagram of P, denoted by *Vor(P)*, is the partition of space induced by the voronoi cell *V(p)*.

2.2 Delaunay triangulation

Delaunay triangulation *D(P)* of P is defined as the geometric dual of the voronoi diagram: there is a edge between two points p and p' in the Delaunay triangulation if and only if their Voronoi cell *V(p)* and *V(p')* have a non-empty intersection. It yields a triangulation of P, that is to say a partition of the convex hull of P into d-dimensional simplices. See figre 2 (left).

2.3 Pole

Pole is defined by Ammenta and Bern in [1]. Each point p in P has two poles. The Voronoi cell *V(p)* is a convex polyhedron whose vertices are Voronoi vertices. It is easy to compute the vertices of *V(p)*, because they are the centers of the circumscribing spheres of the tetrahedra in *D(P)* that have p for a vertex. Let u be the vertex of *V(p)* furthest from p; u is called a pole of p. Let v be the vertex of *V(p)* furthest from p for which the angle $\angle upv$ exceeds 90°; v is also called a pole of p. Figure 2(middle) illustrates the two poles of a typical sample point.

2.4 Graph cut

Let $G = \{V, E\}$ be a weighted digraph with two distinguished terminal vertices s, t called the source and the sink. The incoming degree of s is zero, and the out going degree of t is also zero. The edges in E linking with s or t are named t-link. The other edges are named n-link. An s-t cut $C = (V_s, V_t)$ is a partition of the vertices into two sets such that s in V_s and t in V_t. (Note that an s-t cut can also be equivalently defined as the set of edges from V_s to V_t.) The cost of the s-t cut, denoted |C|, equals the sum of the weights of the edges from V_s to V_t. The minimum s-t cut problem is to find the s-t cut with the smallest cost. This problem can be solved very efficiently by computing the maximum flow between the terminals, according to a theorem due to Ford and Fulkerson [22].See figure2(right).

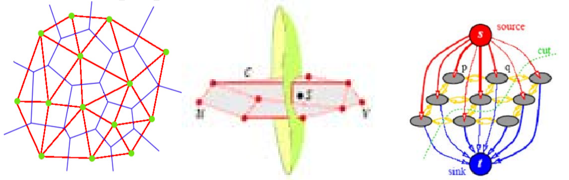


Figure 2. The left picture: Voronoi diagram(blue edge) of a set of 2D points(green dots) and its associated Delaunay triangulation(red edges.) The middle picture: The poles of s—the Voronoi vertices u and v—typically lie on opposite sides of the surface being recovered. The right picture: a graph and its s-t cut.

3. RECONSTRUCTION METHOD

Our method consists of the following three main steps: the first step is to extract features from the input views, then match pair-wise between different views by taking epipolar geometry into account. And a 3D point cloud is generated by computing the 3D position associated to each match. In the second step, an adaptive tetrahedral decomposition of space is built by computing the 3D Delaunay triangulation of the 3D point set. Finally a graph cut optimization is used to extract the surface of the scene from this triangulation.

3.1 Quasi-dense 3D point cloud generatios

The first step in our method is the generation of a quasi-dense 3D point cloud. To this end, pairs of keypoints are matched across different views and a 3D point cloud is generated by computing the 3D position associated to each match. This part is rather straightforward, here is only a sketch. In our work, SIFT in [21] is used for feature extraction and matching. In order to get quasi-dense 3D point cloud , all the matches whose matching scores are not significantly lower than the best match are used. Thus many false matches are susceptible to be contained. However, the global optimization in the final step is able to remove false matches.

3.2 Delaunay triangulatios

We assume the reader is familiar with the algorithm of computing Delaunay triangulations and Voronoi diagrams in three dimensions. So here we don not explain them in detail. In our experiment, the Delaunay triangulation and Voronoi diagram are computed using the Computational Geometry Algorithms Library (CGAL) [23]. Here is a sketch: Let P be the quasi-dense 3D point cloud obtained above. We augment P to $P+$ which is a union of P and eight bounding box vertices, the corners of a large cube that encloses P. (The width of the cube should be much greater than the diameter of P, so that no point lies near any side of the cube). Let D be the Delaunay tetrahedralization of $P+$, and V be the Voronoi diagram of $P+$ (and the geometric dual of D). For each tetrahedron t in the tetrahedralization D, there is a dual vertex v of the Voronoi diagram V, and v is the center of the sphere that circumscribes t. From the Voronoi diagram, we can also easily get the poles of every point in P(The eight bounding box vertices in $P+$ are not considered to have poles. And through augmenting P to $P+$, all the poles of P are finite values.).

The goal of next step is to label each tetrahedron—or equivalently, each Voronoi vertex—inside or outside.

3.3 Surface extraction

The final step in our method consists in labeling each tetrahedron of the Delaunay triangulation 1 (inside) or 0 (outside). The output triangular mesh is then obtained by taking the triangular facets between adjacent tetrahedra having different labels. This constrains the reconstructed surface to be included in the Delaunay triangulation. We decompose this labeling process into two steps -labeling the poles, followed by labeling the remaining tetrahedra.

3.3.1 Labeling poles

Amenta and Bern show that in the absence of noise, the tetrahedra that are the duals of the poles are likely to extend well into the interior or exterior of the object whose surface is being recovered, as Figure 1(middle) shows. So the poles are very useful for guiding the position of surface. The next we use the information of visibility to determine which poles are inside and which poles are outside the object.

Let I_i be the i-th image, and C_i the position of the i-th camera corresponding to I_i. Suppose the point $p \in P$ is visible in I_{i1}, I_{i2}... I_{im}, and invisible in the other images. v_1, v_2 are the two poles of p, if

$$\overrightarrow{pv_j} \bullet \frac{1}{m} \sum_{k=1}^{m} \overrightarrow{pC_{ik}} > 0 \qquad (1)$$

And

$$\| \overrightarrow{pv_j} \| > L \qquad (2)$$

Then we label v_j (or equivalently, the tetrahedron corresponding to v_j) as 0.

If

$$\overrightarrow{pv_j} \bullet \frac{1}{m} \sum_{k=1}^{m} \overrightarrow{pC_{ik}} < 0 \qquad (3)$$

And

$$\| \overrightarrow{pv_j} \| > L \qquad (4)$$

Then we label v_j (or equivalently, the tetrahedron corresponding to v_j) as 1. j=1, 2, L is a proper positive values.

If a tetrahedron corresponding to a pole is both labeled 0 and 1, we remove its labels.

For noisy point cloud, the restriction (3) or (4) is strongly recommended. As pointed out by Amenta et al. if there is no noise in point cloud, the poles should lie near the medial axis of the original object, and dualize to tetrahedra that extend deeply into the object's interior. However, random measurement errors in the point coordinates can create spurious poles that are closer to the surface than the medial axis. Fortunately, a spurious pole is usually easy to recognize: it dualizes to a small tetrahedron that is entirely near the object surface. Kolluri et al. [20] used this restriction to deal with surface reconstruction from noisy point cloud, and got good results. In our experiments, this restriction also substantially improves the results

3.3.2 Labeling remaining tetrahedra

In this step, for the remaining tetrahedra, a globally optimal label assignment is efficiently found using graph cuts. We consider the dual graph to the Delaunay triangulation, in other words, the graph whose vertices correspond to Delaunay tetrahedra, and whose edges correspond to the triangular facets between adjacent tetrahedra(every edge has two directions). In addition, in this graph, we take all the tetrahedra having been labeled 1 as a vertex- the sink t, and all the tetrahedra having been labeled 0 as a vertex-the source s. Then we get a directed graph with two terminals(s and t).

In the sequel, we note S the surface to be reconstructed. As discussed above, S is a union of Delaunay triangles. We wish to minimize an energy functional composed of two terms, one dealing with photoconsistency and one dealing with surface smoothness:

$$\mathrm{E(S)} = \mathrm{E}_{\mathrm{photo}}(\mathrm{S}) + \lambda_{\mathrm{smooth}}\ \mathrm{E}_{\mathrm{smooth}}(\mathrm{S}) \text{--}(5)$$

where λ smooth is positive weight. Compared to the energy function in [19], the visibility energy is absent here. That is because in our experiments we find this term has little influence on the results. Next we give the exact definition of each of the term and describe how it can be implemented in the graph cuts framework.

Surface photo-consistency

The photo-consistency term Ephoto(S) of our energy measures how well the given surface S matches the different input images in which it is seen. It is defined as the sum over the whole surface of some photo-consistency measure $\rho \geqslant 0$ (in our case, every triangle of the surface has a uniform photo-consistency):

$$\mathrm{E}_{\mathrm{photo}}(\mathrm{S}) = \sum_{\mathrm{T} \in \mathrm{S}} \rho(\mathrm{T})\mathrm{A}(\mathrm{T}) \qquad (6)$$

A(T) is the area of triangular patch T. The photo-consistency of each triangle is computed only in the views from which its three vertices were reconstructed. Furthermore, as a triangle of the surface S lies by definition on the interface between the inside and the outside of the reconstructed object(s), its orientation needs to be taken into account, which means that the two possible orientations of a given triangle get different photo-consistencies, each computed only in the subset of the considered views compatible with the given orientation of the triangle.

This maps pretty easily onto the graph cuts framework:

for each directed pair of tetrahedra (represented by nodes p and q in the graph) which shares a triangle T with normal $\vec{n}$ (pointing from tetrahedron p to tetrahedron q), an edge $\mathrm{p} \rightarrow \mathrm{q}$ is added with a weight $\mathrm{w}_{\mathrm{pq}} = \rho\{\mathrm{C}_{\mathrm{i}} | \overrightarrow{\mathrm{d}_{\mathrm{i}}} \bullet \vec{\mathrm{n}} > 0\}(\mathrm{T})$,where $\overrightarrow{d_i}$ is the direction from the center of the triangle to the center of the i-th camera C_i.

The photo-consistency is evaluated with a software rasterizer with sweeps the projection of each triangle of the Delaunay triangulation in the chosen views and computes the mean of the color variance of the pixels in this triangle.

Surface smoothness

In [19], surface smoothness is enhenced by minimizing the area of the surface. But it most probably retain thin and long triangular patches in the final mesh. To avoid this, here, we use radius of circumcircle of the triangular patches. Then the smoothness energy term is:

$$E_{\mathrm{smooth}}(S) = R(S) = \sum_{T\in S} R(T) \qquad (7)$$

This can also trivially be minimized in the graph cuts framework: for each pair of tetrahedra (sharing a triangle T) represented by nodes p and q in our graph, an edge $p \rightarrow q$ is added with a weight w_{pq}=R(T) . An opposite edge $q \rightarrow p$ with the same weight $w_{pq}=w_{qp}$ is also added.

3.3.3 *Computing minimum s-t cuts*

Form what discussed above, we can extract the triangular mesh of the scene by computing the minimum s-t cut of the graph that we have constructed above. There are a large number of fast algorithms for computing a minimum s-t cut. The worst case complexity is low-order polynomial; however, in practice the running time is nearly linear for graphs with many short paths between the source and the sink, such as the one we constructed. In our experiments, we use the software described in [24] which is better suited for regular grid-based graphs more commonly found in computer vision.

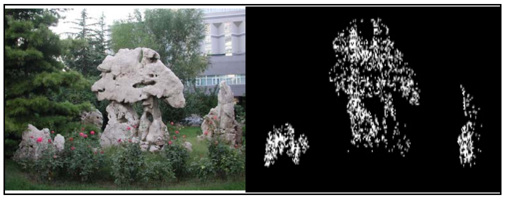
3.4 Smoothing

Triangulated surfaces extracted from noisy point cloud are bumpy. The final optional step is to use standard Laplacian smoothing [25] to make the mesh surface smoother. Laplacian smoothing visits each vertex in the triangulation in turn, and moves it to the centroid of its neighboring vertices. We performed three iterations of smoothing in each of our following experiments.

4. Experimental results

In this section we present our experimental results from a variety of image sequence of outdoor scenes.

Figure 3 illustrates the performance of our method on an image sequence of 9 images. The left column is three of the 9 images. The right column is the 3D point cloud we have generated and the rockery surface reconstructed by our method.



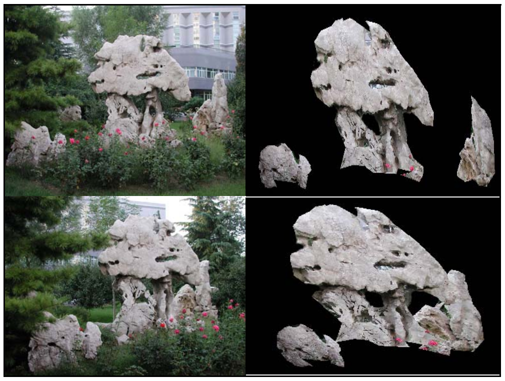


Figure 3. The performance of our method on an image sequence of 9 images. The left column is three of the 9 images. The right column is the 3D point cloud we have generated and the rockery surface reconstructed by our method.

The second experiment is the result of our method on an image sequence of 26 images, see figure 4. The left column is three of the 26 images. The right column is the 3D point cloud we have generated and the scene surface reconstructed by our method.

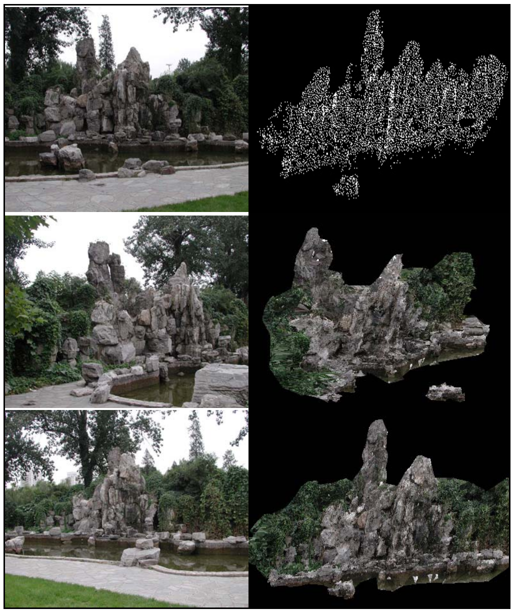


Figure 4. The result of our method on an image sequence of 26 images. The left column is three of the 26 images. The right column is the 3D point cloud we have generated and the scene surface reconstructed by our method.

5. Conclusion

In this paper, we present a new multi-view reconstruction method adapted to large-scale scenes from multiple images. We also show the performance of our method on a variety of complex scenes. The experimental results demonstrate the effectiveness and efficiency of our method.

ACKNOWLEDGMENT

This work was supported by the national 863 program(2007AA01Z341), the national science foundation of china(60673104) and the national science and technology development plan (2006BAK31B04)..

REFERENCES

[1] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. Discrete and Computational Geometry, 22:481–504, 1999

[2] D. Attali, J.-D. Boissonnat, and A. Lieutier. Complexity of the Delaunay triangulation of points on surfaces: the smooth case. In Annual Symposium on Computational Geometry, pages 201–210, 2003.

[3] A. Laurentini. The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(2):150–162, 1994

[4] Y. Furukawa and J. Ponce. Carved visual hulls for imagebased modeling. In European Conference on Computer Vision, volume 1, pages 564–577, 2006

[5] A. Hornung and L. Kobbelt. Hierarchical volumetric multiview stereo reconstruction of manifold surfaces based on dual graph embedding. In IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages 503–510, 2006.

[6] S. Tran and L. Davis. 3D surface reconstruction using graph cuts with surface constraints. In European Conference on Computer Vision, volume 2, pages 219–231, 2006.

[7] A. Broadhurst, T. W. Drummond, and R. Cipolla. A probabilistic framework for space carving. In IEEE International Conference on Computer Vision, volume 1, pages 388–393, 2001.

[8] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDE's, level set methods and the stereo problem. IEEE Transactions on Image Processing, 7(3):336–344, 1998.

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

[10] C. H. Esteban and F. Schmitt. Silhouette and stereo fusion for 3D object modeling. Computer Vision and Image Understanding, 96(3):367–392, 2004

[11] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. The International Journal of Computer Vision, 35(2):151–173, 1999.

[12] H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. The International Journal of Computer Vision, 63(3):175–189, 2005.

[13] Y. Boykov and V. Lempitsky. From photohulls to photoflux optimization. In British Machine Vision Conference, volume 3, pages 1149–1158, 2006

[14] B. Curless and M. Levoy. A volumetric approach for building complex models from range images. In ACM SIGGRAPH, pages 303–312, 1996.

[15] S. Tran and L. Davis. 3D surface reconstruction using graph cuts with surface constraints. In European Conference on Computer Vision, volume 2, pages 219–231, 2006.

[16] C. Strecha, R. Fransens, and L. V. Gool. Combined depth and outlier estimation in multi-view stereo. In IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 2394–2401, 2006

[17] P. Gargallo and P. Sturm. Bayesian 3D modeling from images using multiple depth maps. In IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 885–891, 2005

[18] M. Goesele, B. Curless, and S. M. Seitz. Multi-view stereo revisited. In IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 2402–2409, 2006

[19] P. Labatut, J. P. Pons, Renaud Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points,delaunay triangulation and graph cuts. In IEEE International Conference on Computer Vision, pp.1-8, 2007

[20] R. Kolluri et al., Spectral Surface Reconstruction from Noisy Point Clouds. Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Vol. 71, ACM Press New York, NY, USA, pp. 11–21.

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. The International Journal of Computer Vision, 60(2):91–110, 2004.

[22] L.Ford, D.Fulkerson. Flows in Networks. Princeton University Press,1962.

[23] J.-D. Boissonnat, O. Devillers, M. Teillaud, and M. Yvinec. Triangulations in CGAL. In Annual Symposium on Computational Geometry, pages 11–18, 2000.

[24] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(9):1124–1137, 2004.

[25] L. R. Hermann, Laplacian-Isoparametric Grid Generation Scheme. Journal of the Engineering Mechanics Division of the American Society of Civil Engineers 102 (Oct. 1976), 749–756.