A New Visibility Model for Surface Reconstruction

Yang Zhou[1,2], Shuhan Shen[1,2(✉)], and Zhanyi Hu[1,2]

[1] NLPR, Institute of Automation, Chinese Academy of Sciences,
Beijing 100190, People's Republic of China
{yang.zhou,shshen,huzy}@nlpr.ia.ac.cn
[2] University of Chinese Academy of Sciences,
Beijing 100049, People's Republic of China

Abstract. In this paper, we propose a new visibility model for scene reconstruction. To yield out surface meshes with enough scene details, we introduce a new visibility model, which keeps the relaxed visibility constraints and takes the distribution of points into consideration, thus it is efficient for preserving details. To strengthen the robustness to noise, a new likelihood energy term is introduced to the binary labeling problem of Delaunay tetrahedra, and its implementation is presented. The experimental results show that our method performs well in terms of detail preservation.

Keywords: Surface reconstruction · Visibility · Scene detail

1 Introduction

Scene reconstruction from images is a fundamental problem in Computer Vision. It has many practical applications in entertainment industry, robotics, cultural heritage digitalization and geographic systems. For decades, it has been studied due to its low cost of data acquisition and various usages. In recent years, researchers have made tremendous progress in this field. For small objects under controlled conditions, current scene reconstruction methods could achieve great results [7,16]. However, when it comes to large scale scenes with multi-scale objects, they would have some problems with the completeness and accuracy, especially when dealing with scene fine details [18].

Scene details like small scale objects and object edges are essential part of scene surfaces. However, preserving scene details in multi-scale scenes has been a difficult problem. The existing surface reconstruction methods either ignore the scene details or rely on further refinement to restore them. This is because firstly, compared with noise, the supportive points in such areas are sparse, making it difficult to distinguish true surface points from false ones; and secondly, the models and associated parameters employed in existing methods are not particularly adept for large scale ranges, where scene details are usually compromised for overall accuracy and completeness. While the above first case seems unsolvable

due to the lack of sufficient information, we only focus on the second case in this work. In particular, we propose a new visibility model along with a new energy formulation for surface reconstruction. Our method follows the idea in [6,11], with the visibility model and energy formulation replaced with ours. To make our method adept for large scale ranges, we propose a new visibility model with adaptive ends of lines of sight; in the meantime, a new likelihood energy term is introduced to keep the proposed method robust for noise. Experimental result shows that our visibility model is efficient for preserving scene details.

2   Related Work

Recently, a variety of works have been done to advance 3D scene reconstruction. To deal with small objects, silhouette based methods [5,13] are proposed. The silhouettes provide proper bounds for the objects, which helps to reduce the computing cost and yield a good model for the scene. However, good silhouettes rely on effective image segmentation, which remains a difficult task so far. Volumetric methods like space carving [1,10,24], level sets [8,15] and volumetric graph cut [14,20,22] often yield good results for small objects. But the computational and memory costs increase rapidly as scene scale grows, which makes them unsuitable for large scale scene reconstruction.

For outdoor scenes, uncontrollable imaging conditions and multiple scale structure make it hard to reconstruct scene surfaces. A common process of large scale scene reconstruction is to generate the dense point cloud from calibrated images first, and then extract the scene surface. The dense point cloud can be generated by depth fusion method [2,17,19] or by feature expansion methods [3]. Once the dense point cloud is generated, scene surface can be reconstructed by Possion surface construction [9], by level-set like method [21] or by graph cut based methods [6,11,12]. Graph cut based methods are exploited most for its efficiency and ability of filtering noise. In these methods, Delaunay tetrahedra are generated from dense point cloud first, then a visibility model is applied to weight the facets in Delaunay tetrahedra, finally the binary labeling problem of tetrahedra is solved and the surface is extracted. The typical visibility model is introduced in [12]. The basic assumption of the visibility model in [12] is that the space between camera center and 3D point is free-space, while the space behind 3D point is full-space. Then this typical visibility model is promoted by a refined version in [11], namely soft-visibility, to cope with noisy point clouds.

In this paper, we follow the idea of [6,11] and exploit the visibility information in the binary labeling problem of Delaunay tetrahedra. The main contributions of our proposed method are the introductions of (a) a new visibility model and (b) a new energy term for efficient noise filtering. Experimental comparison results show that our method rivals the state-of-the-art methods [2,3,6,19] in terms of accuracy and completeness of scene reconstruction, but performs better in terms of detail preservation.
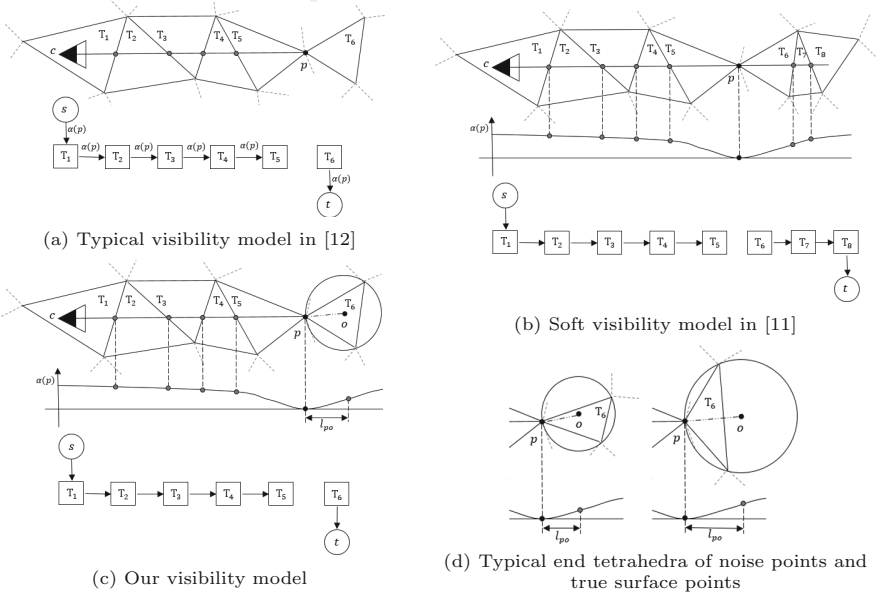
(a) Typical visibility model in [12]

(b) Soft visibility model in [11]

(c) Our visibility model

(d) Typical end tetrahedra of noise points and true surface points

Fig. 1. (a) Typical visibility model, (b) soft visibility model and (c) our visibility model in 2D. (d) From left to right: typical end tetrahedron (in 2D) of noise points and that of true surface points on densely sampled surfaces.

3   Visibility Models and Energies

The input of our method is a dense point cloud generated by MVS methods. Each point is attached with the visibility information recording that from which views the point is seen. Then, Delaunay tetrahedra are constructed, and the scene reconstruction problem is formulated as a binary labeling problem with a given visibility model. The total energy of the binary labeling problem is

$$E_{total} = E_{vis} + \lambda_{like}E_{like} + \lambda_{qual}E_{qual} \tag{1}$$

where λ_{like} and λ_{qual} are two constant balancing factors; E_{vis} and E_{like} are defined in Sects. 3.2 and 4.1; E_{qual} is the surface quality energy in [11]. By minimizing E_{total} with Maxflow/Mincut algorithm, the label assignment of Delaunay tetrahedra is yielded and a triangle mesh is extracted, which consists of triangles lying between tetrahedra with different labels. Further optional refinement can be applied as described in [23]. In the following subsections, we introduce the existing visibility models and our new one.

3.1   Existing Visibility Models

The typical visibility model [12] assumes that, for each line of sight v, the space between camera center c and point p is free-space, and that behind point p

(a) Point cloud with heavy noise (b) reconstructed meshes with- (b) reconstructed meshes with
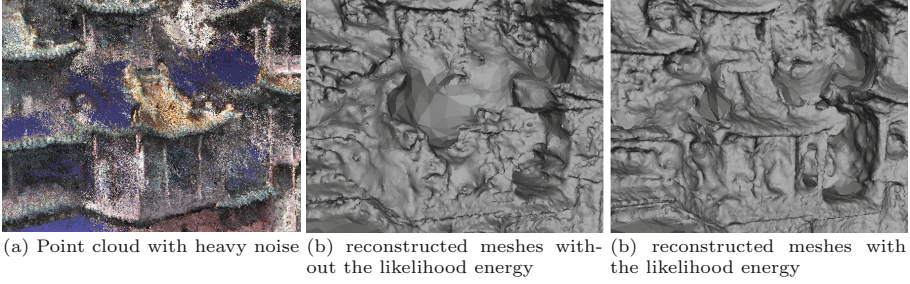   out the likelihood energy   the likelihood energy

Fig. 2. Surface reconstruction without and with the likelihood energy.

is full-space. Thus, the tetrahedra (T_1–T_5 in Fig. 1(a)) intersected by segment (c, p) should be labeled as outside, and that (T_6 in Fig. 1(a)) right behind point p as inside. For a single line of sight, the above label assignment is desirable. While taking all lines of sight into consideration, some surface part might not be handled properly. This violates the label assignment principle described above. To quantize the conflicts, the facets intersected by lines of sight are given weights of $W(l_{T_i}, l_{T_j})$, and those for punishing bad label assignments of the first and the last tetrahedron are $D(l_{T_1})$ and $D(l_{T_{N_v+1}})$, respectively. Therefore, the visibility energy is the sum of the penalties of all the bad label assignments in all the lines of sight, as

$$E_{vis}^{typical} = \sum_{v \in \mathcal{V}} [D(l_{T_1}) + \sum_{i=1}^{N_v-1} W(l_{T_i}, l_{T_{i+1}}) + D(l_{T_{N_v+1}})] \qquad (2)$$

where if $l_{T_1} = 1$, $D(l_{T_1}) = \alpha(p)$; otherwise, $D(l_{T_1}) = 0$; if $l_{T_{N_v+1}} = 0$, $D(l_{T_{N_v+1}}) = \alpha(p)$; otherwise, $D(l_{T_{N_v+1}}) = 0$; if $l_{T_i} = 0 \wedge l_{T_j} = 1$, $W(l_{T_i}, l_{T_j}) = \alpha(p)$; otherwise, $W(l_{T_i}, l_{T_j}) = 0$; $\alpha(p)$ is the weight of point p, which is its photo-consistency score; $\mathcal{V}$ is the visibility information set; N_v is the number of tetrahedra intersected by the line of sight v, indexed from the camera center c to the point p; $N_v + 1$ denotes the tetrahedron behind the point p; l_T is the label of tetrahedron T, with 1 stands for inside and 0 for outside.

The typical visibility model described above is effective, but it has several flaws in dealing with dense and noisy point clouds. In such cases, the surfaces yielded by [12] tend to be overly complex, with bumps on and handles inside the model [11]. One possible solution is the soft visibility in [11], shown in Fig. 1(b). The basic assumptions of the two visibility models are similar. The differences are that in the soft visibility model, the edge weights are given a factor $(1-e^{-d^2/2\sigma^2})$ according to the distance d between the point p and the intersecting point, and the end of the line of sight is shifted to a distance of $k\sigma$ along it.

3.2   Our Proposed Visibility Model

Though the soft visibility model is effective to filter noise points, it sometimes performs badly in preserving details, especially in a large scene containing small
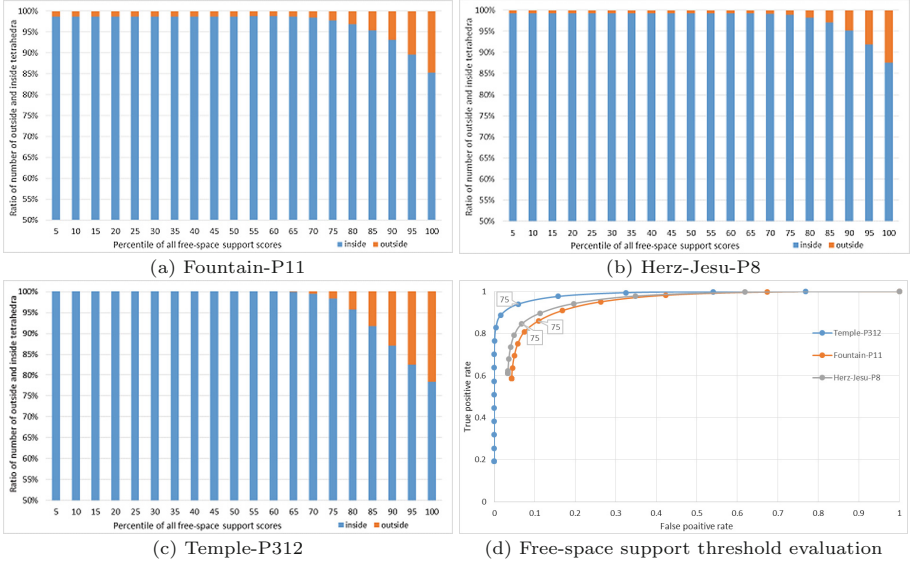
(a) Fountain-P11

(b) Herz-Jesu-P8

(c) Temple-P312

(d) Free-space support threshold evaluation

Fig. 3. This figure shows the ratios of number of outside and inside tetrahedra with different percentiles of all free-space support scores on (a) Fountain-P11 [18], (b) Herz-Jesu-P8 [18] and (c) Temple-P312 [16]. (d) The true positive rates and false positive rates of different free-space support threshold.

scale objects (see experimental results). According to our observations, this happens mainly because of the strong constraint imposed on the end of line of sight in the tetrahedron $k\sigma$ from the point p along the line of sight. It could be free-space even the point p is a true surface point in some cases.

To balance noise filtering and detail preserving, we propose a new visibility model shown in Fig. 1(c). In our visibility model, we keep the relaxed visibility constraints in the space between the camera center c and the point p, and set the end of line of sight in the tetrahedron right behind the point p. We observe that the end tetrahedra of noise points and true surface points have different shapes, as shown in Fig. 1(d). In addition, to determine the weight of the t-edge of the end tetrahedron, we compare the end tetrahedra of noisy points and true surface points on public datasets. Figure 1(d) shows a typical end tetrahedron of a noisy point and that of a true surface point on densely sampled surfaces in 2D space. Noise points tend to appear a bit away from true surface, which makes the end tetrahedra thin and long, and true surface points are often surrounded by other true surface points, which makes their end tetrahedra flat and wide. Based on these observations, we set a weight of $\alpha(p)(1-e^{-kr^2/2\sigma^2})$ to the t-edge of the end tetrahedron, where r is the radius of the circumsphere of the end tetrahedron. Thus the corresponding visibility energy is formulated as
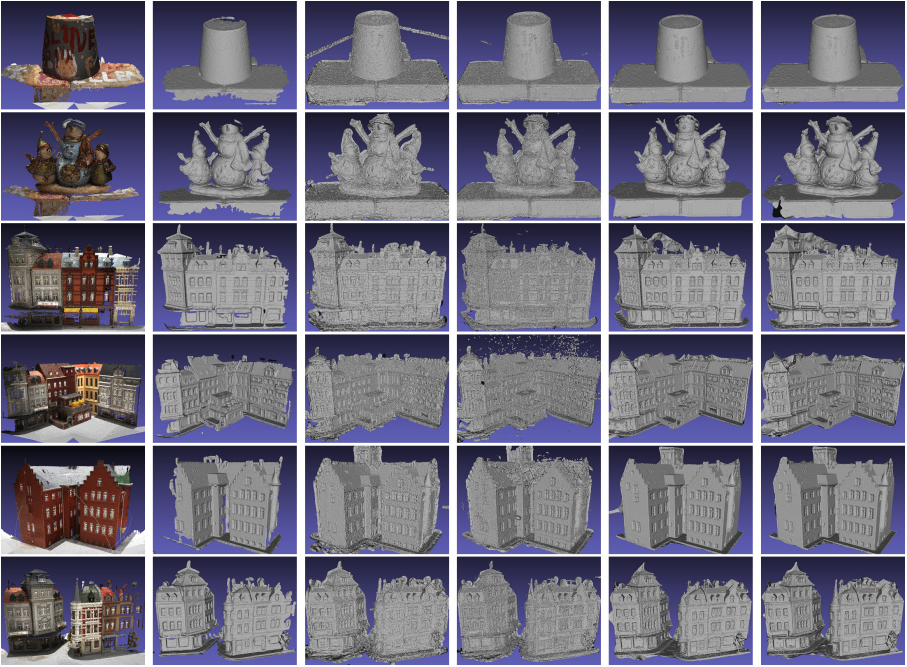
Fig. 4. Results of the five methods on MVS Data Set [7]. From left to right: the reference model, Tola et al. [19], Furukawa and Ponce [3], Campbell et al. [2], Jancosek and Pajdla [6] and ours.

$$E_{vis} = \sum_{v \in \mathcal{V}} [D(l_{T_1}) + \sum_{i=1}^{N_v-1} W(l_{T_i}, l_{T_{i+1}}) + D(l_{T_{N_v+1}})] \tag{3}$$

where if $l_{T_1} = 1$, $D(l_{T_1}) = \alpha(p)$; otherwise, $D(l_{T_1}) = 0$; if $l_{T_{N_v+1}} = 0$, $D(l_{T_{N_v+1}}) = \alpha(p)(1 - e^{-kr^2/2\sigma^2})$; otherwise, $D(l_{T_{N_v+1}}) = 0$; if $l_{T_i} = 0 \wedge l_{T_j} = 1$, $W(l_{T_i}, l_{T_j}) = \alpha(p)(1 - e^{-d^2/2\sigma^2})$; otherwise, $W(l_{T_i}, l_{T_j}) = 0$. Although Eqs. 2 and 3 look identical, they differ from each other in term $W(l_{T_i}, l_{T_{i+1}})$ and $D(l_{T_{N_v+1}})$. One may refer to the text for details.

4   Likelihood Energy for Efficient Noise Filtering

In both the typical visibility model and our proposed one, the end of each line of sight is set in the tetrahedron right behind the point p. This practice could weaken the ability of noise filtering. When the surface is sampled very noisy, part of the surface fails to be reconstructed, as shown in Fig. 2(b). This is mainly due to the unbalanced links of s-edges and t-edges in the s-t graph. The s-edges are heavy-weighted and concentrative, while the t-edges are low-weighted and scattered. The noisier the point cloud is, the greater the gap between s-edges and t-edges will be.

Table 1. Performance of 5 methods on MVS data set [7]

Dataset	scan1				scan24				scan44			
	$\bar{A}$	$A_{0.5}$	$\bar{C}$	$C_{0.5}$	$\bar{A}$	$A_{0.5}$	$\bar{C}$	$C_{0.5}$	$\bar{A}$	$A_{0.5}$	$\bar{C}$	$C_{0.5}$
Tola [19]	**0.2449**	0.1918	0.3994	0.3213	**0.3937**	0.2656	0.8512	0.4340	**0.6410**	**0.3251**	0.7382	0.4160
Furu [3]	0.4533	0.2517	0.4105	0.3533	0.7455	0.3125	0.4828	0.3692	1.0951	0.4906	0.5264	0.3872
Camp [2]	0.4428	0.2550	0.3988	0.3601	3.4601	0.5227	0.4010	0.2954	1.2926	0.5641	**0.3807**	**0.2976**
Jan [6]	0.3595	0.2314	0.4246	0.3444	0.6054	**0.2032**	**0.3488**	**0.2574**	0.8098	0.36739	0.5652	0.3907
ours	0.3053	**0.1760**	**0.3062**	**0.2556**	0.7865	0.2354	0.4230	0.2874	1.2309	0.5410	0.6787	0.4577

$\bar{A}$: mean accuracy; $A_{0.5}$: median accuracy; $\bar{C}$: mean completeness; $C_{0.5}$: median completeness.

4.1 Likelihood Energy

To solve the problem, we introduce a likelihood energy E_{like} to the total energy of the binary labeling problem. E_{like} is defined as

$$E_{like} = \sum_{i=1}^{N} D(l_{T_i}), \text{ where } D(l_{T_i}) = \begin{cases} U_{out}(T_i) & \text{if } l_{T_i} = 1 \\ U_{in}(T_i) & \text{otherwise} \end{cases} \tag{4}$$

where N is the total number of Delaunay tetrahedra. E_{like} measures the penalties of a wrong label assignment. For each tetrahedron, a penalty is given for its wrong label, i.e. $U_{out}(T_i)$ or $U_{in}(T_i)$. To evaluate the likelihood of the label assignment of a tetrahedron, we employ the measure *free-space support* [6], which is used to measure the emptiness of a compact space. For each line of sight (c, p) that intersects tetrahedra T, it contributes to the emptiness of T_i, thus increasing the outside probability of T. The free-space support $f(T)$ of tetrahedron T is defined as

$$f(T) = \sum_{(c,p)\in S_T} \alpha(p), \text{ with } S_T = \{(c,p) \mid (c,p) \cap T \neq \emptyset\} \tag{5}$$

In order for free-space support $f(T)$ of tetrahedron T to adequately describe the likelihood energy, we set $U_{out}(T) = \lambda f(T)$ and $U_{in}(T) = \lambda(\beta - f(T))$, where $f(T)$ is the free-space support of tetrahedron T, λ and β are constants.

4.2 Implementation of the Likelihood Energy

For the likelihood term E_{like}, we link two edges for each vertex i in s-t graph. One is from source s to vertex i, with weight $U_{out}(T_i)$; the other is from vertex i to sink t, with weight $U_{in}(T_i)$. However, we observe that the tetrahedra with lower $f(T)$ are easily to be mislabeled, thus we only link the vertices whose correspondent tetrahedra have lower $f(T)$ than the 75th percentile of all $f(T)$s to sink t, and cross out the s-edges of all vertices.

Note that the free space support threshold of the 75th percentile is empirically set. We generate dense point clouds from 3 datasets, Fountain-P11 [18], Herz-Jesu-P8 [18] and Temple-P312 [16]. Then we label the tetrahedra with the method in [6], and take the result as quasi-truth. Finally, we evaluate the ratios
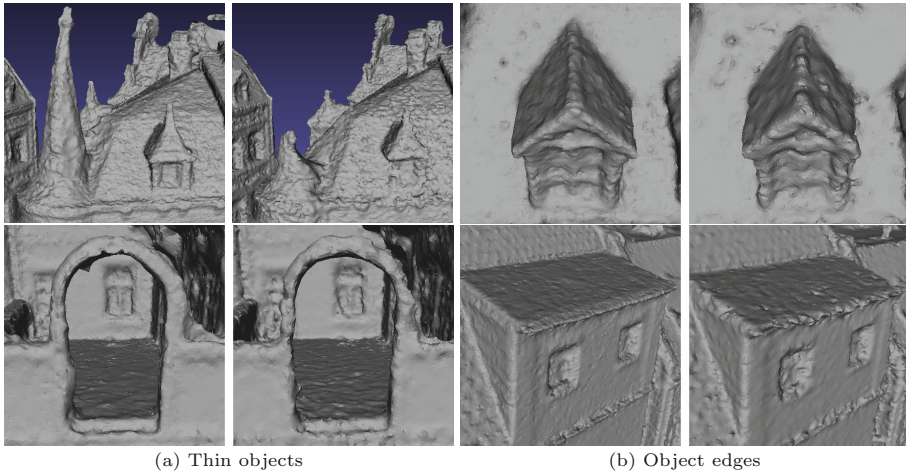
(a) Thin objects   (b) Object edges

Fig. 5. Comparison with the method of Jancosek and Pajdla [6]. The result of our method is on the left columns of (a) and (b) and that of [6] on the right columns of them.

of number of outside and inside tetrahedra in different proportions of all free-space support scores, as shown in Fig. 3. From Fig. 3(a)–(c) we can see that, generally tetrahedron with $f(T)$ lower than the 75th percentile of all $f(T)$s has a high probability to be truly inside. From Fig. 3(d), when the free-space support threshold is set as the 75th percentile of each dataset, both true positive rate and false positive rate are reasonable. Therefore, we set the free space support threshold as the 75th percentile of all free-space support scores. From Fig. 2, we can see that the likelihood energy is helpful for filtering noise.

5   Experimental Results

The input dense point cloud is generated with the open source library Open-MVG[1] and OpenMVS[2]. Sparse point cloud is generated by OpenMVG, then densified with OpenMVS. Delaunay tetrahedralization is computed using CGAL[3] library. Maxflow/Mincut algorithm [4] is used. Our method is tested on public benchmark MVS Data Set [7] and some of our own datasets.

Figure 4 shows the results on the MVS Data Set [7]. From left to right, the reference model, the results of Tola et al. [19], Furukawa and Ponce [3], Campbell et al. [2], Jancosek and Pajdla [6] and ours are shown. The final meshes given by Tola et al. [19], Furukawa and Ponce [3] and Campbell et al. [2] are generated by poisson surface reconstruction method [9] and trimmed, which are provided in

---

[1] http://imagine.enpc.fr/~moulonp/openMVG/.
[2] http://cdcseacave.github.io/openMVS/.
[3] http://www.cgal.org/.
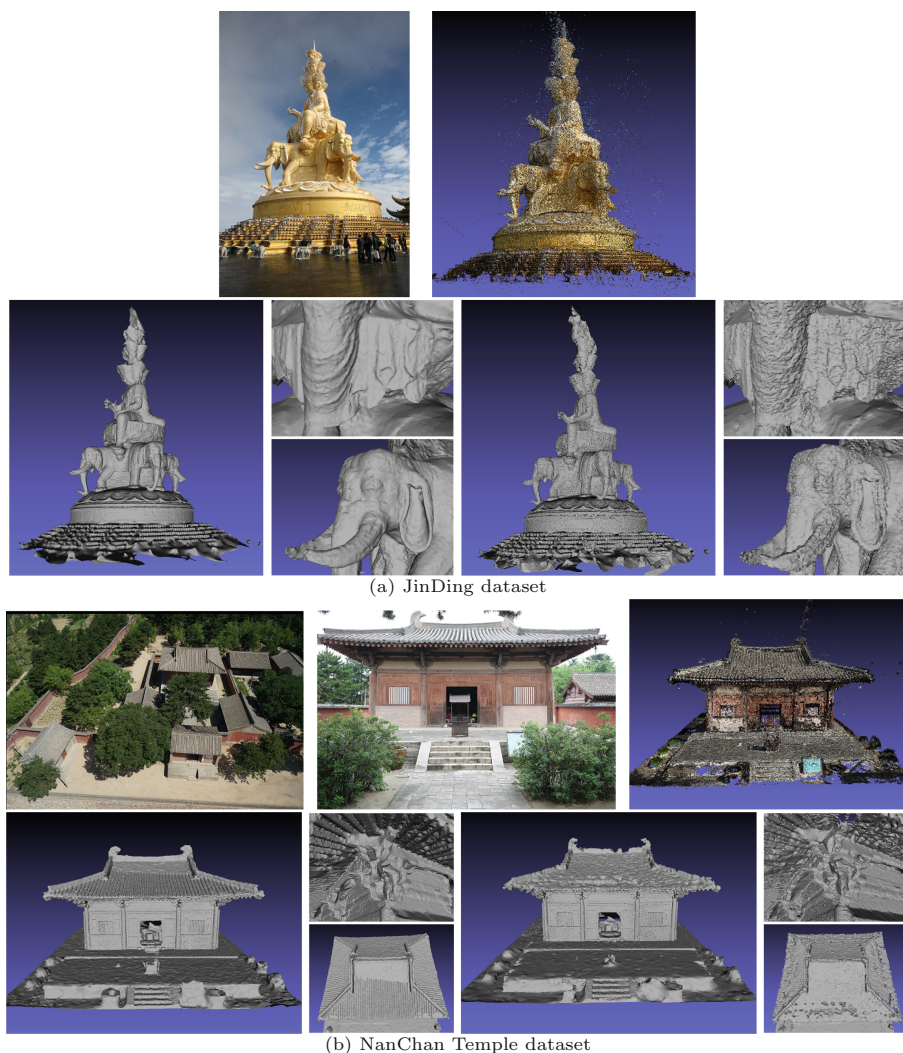
(a) JinDing dataset

(b) NanChan Temple dataset

Fig. 6. Result of (a) JinDing dataset and (b) NanChan Temple dataset. In the first row of (a) and (b), example image(s) and dense point cloud are shown; in the second row of (a) and (b), overview result and details of our method are shown in left 2 columns and those of Jancosek and Pajdla [6] in right 2 columns.

MVS Data Set; the meshes of Jancosek and Pajdla [6] are generated by Open-MVS which contains an reimplementation of [6]. Table 1 shows the quantitative results of the five methods. From Fig. 4 and Table 1, though our method is not quantitatively the best, the meshes generated by our method preserves more details than the others. To better visualize the performances of Jancosek and Pajdla [6] and our method, we enlarge some parts in Fig. 4, and present in Fig. 5,

with the result of our method on the left columns of (a) and (b) and that of [6] on the right columns of them. Figure 5(a) shows the performance of the 2 methods on thin objects. In some cases, [6] fails to reconstruct them completely; even the complete ones are less detailed than those by our method. In Fig. 5(b), our method shows a tremendous capability of preserving sharp object edges, compared with the result of [6]. From Fig. 5, we can notice that our method performs well in dealing with small scale objects and sharp edges, while [6] fails to reconstruct those parts completely and loses some details.

Figure 6 show the results of our method and Jancosek and Pajdla [6] on our own datasets, the JinDing and NanChan Temple datasets. The JinDing dataset consists of 192 images taken by a hand-held camera, with the resolution of 4368×2912. The NanChan Temple dataset contains 2006 images, with 1330 of them taken by a hand-held camera and the rest by an unmanned aerial vehicle (UAV). The ground images have a resolution of 5760×3840 and the aerial ones of 4912×3264. The dense point cloud contains $31M$ and $40M$ points for the two datasets respectively. Note that we downsample the images twice in NanChan Temple dataset so that the point cloud would not be too dense to be processed. In Fig. 6, example image(s), dense point cloud, result overview and details of our method and Jancosek and Pajdla [6] are presented. From the closer view of the result of our method, details like elephant's tusk and ears, cloth folds, beam edges and roof drainages are preserved. In the meantime, the two models of our method avoid over-complexity. The results show that our method performs better than [6] on our datasets.

6   Conclusions and Future Work

In this paper, we present a novel visibility model and a new energy term for surface reconstruction. Our method takes dense point clouds as input. By applying Delaunay tetrahedralization to the dense point cloud, we divide the scene space into cells, and formulate the scene reconstruction as a binary labeling problem to label each tetrahedron inside or outside. To yield out meshes with enough scene details, we introduce a new visibility model to filter out noise and select true surface points. The proposed visibility model keeps the relaxed visibility constraints and takes the distribution of points into consideration, thus it is efficient for preserving details. To strengthen the robustness to noise, a new likelihood energy term is introduced to the binary labeling problem. Experimental result shows that our method performs well in dealing with scene details like thin objects and sharp edges, which will help to attach semantic knowledge to the model. So, for the future work, we would like to investigate to segment the model and add semantic knowledge to each part of the scene.

References

1. Broadhurst, A., Drummond, T.W., Cipolla, R.: A probabilistic framework for space carving. In: Proceedings of 8th IEEE International Conference on Computer Vision, ICCV 2001, vol. 1, pp. 388–393. IEEE (2001)
2. Campbell, N.D.F., Vogiatzis, G., Hernández, C., Cipolla, R.: Using multiple hypotheses to improve depth-maps for multi-view stereo. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 766–779. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88682-2_58
3. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1362–1376 (2010)
4. Goldberg, A.V., Hed, S., Kaplan, H., Tarjan, R.E., Werneck, R.F.: Maximum flows by incremental breadth-first search. In: Demetrescu, C., Halldórsson, M.M. (eds.) ESA 2011. LNCS, vol. 6942, pp. 457–468. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23719-5_39
5. Guan, L., Franco, J.S., Pollefeys, M.: 3D occlusion inference from silhouette cues. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
6. Jancosek, M., Pajdla, T.: Exploiting visibility information in surface reconstruction to preserve weakly supported surfaces. In: International Scholarly Research Notices 2014 (2014)
7. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 406–413. IEEE (2014)
8. Jin, H., Soatto, S., Yezzi, A.J.: Multi-view stereo reconstruction of dense shape and complex appearance. Int. J. Comput. Vis. **63**(3), 175–189 (2005)
9. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of 4th Eurographics Symposium on Geometry Processing, vol. 7 (2006)
10. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. Int. J. Comput. Vis. **38**(3), 199–218 (2000)
11. Labatut, P., Pons, J.P., Keriven, R.: Robust and efficient surface reconstruction from range data. In: Computer Graphics Forum, vol. 28, pp. 2275–2290. Wiley Online Library (2009)
12. Labatut, P., Pons, J.P., Keriven, R.: Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In: 2007 IEEE 11th international conference on computer vision. pp. 1–8. IEEE (2007)
13. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Trans. Pattern Anal. Mach. Intell. **16**(2), 150–162 (1994)
14. Lempitsky, V., Boykov, Y., Ivanov, D.: Oriented visibility for multiview reconstruction. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 226–238. Springer, Heidelberg (2006). https://doi.org/10.1007/11744078_18
15. Pons, J.P., Keriven, R., Faugeras, O.: Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. Int. J. Comput. Vis. **72**(2), 179–193 (2007)
16. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), vol. 1, pp. 519–528. IEEE (2006)
17. Shen, S.: Accurate multiple view 3D reconstruction using patch-based stereo for large-scale scenes. IEEE Trans. Image Process. **22**(5), 1901–1914 (2013)

18. Strecha, C., von Hansen, W., Van Gool, L., Fua, P., Thoennessen, U.: On bench-marking camera calibration and multi-view stereo for high resolution imagery. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, pp. 1–8. IEEE (2008)
19. Tola, E., Strecha, C., Fua, P.: Efficient large-scale multi-view stereo for ultra high-resolution image sets. Mach. Vis. Appl. **23**(5), 903–920 (2012)
20. Tran, S., Davis, L.: 3D surface reconstruction using graph cuts with surface constraints. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 219–231. Springer, Heidelberg (2006). https://doi.org/10.1007/11744047_17
21. Ummenhofer, B., Brox, T.: Global, dense multiscale reconstruction for a billion points. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1341–1349 (2015)
22. Vogiatzis, G., Torr, P.H., Cipolla, R.: Multi-view stereo via volumetric graph-cuts. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), vol. 2, pp. 391–398. IEEE (2005)
23. Vu, H.H., Labatut, P., Pons, J.P., Keriven, R.: High accuracy and visibility-consistent dense multiview stereo. IEEE Trans. Pattern Anal. Mach. Intell. **34**(5), 889–901 (2012)
24. Yang, R., Pollefeys, M., Welch, G.: Dealing with textureless regions and specular highlights-a progressive space carving scheme using a novel photo-consistency measure. In: ICCV, vol. 3, pp. 576–584 (2003)