OPTIMIZING CAMERA POSITIONS FOR MULTI-VIEW 3D RECONSTRUCTION

Ningqing Qian, Chao-Yang Lo

Institute of Communications Engineering
RWTH-Aachen University
52056 Aachen, Germany
qian@ient.rwth-aachen.de

ABSTRACT

In the last decades, a number of high-quality multi-view 3D reconstruction algorithms have been developed to reconstruct a 3D object model with a collection of images captured from different points of view. Intuitively, with more images from different viewpoints, more information can be utilized for the reconstruction algorithms. However, the acquisition effort, storage and computing cost will grow correspondingly with the number of the images. Furthermore, the possible noise and unrelated information introduced by additional images make the reconstruction more challenging. In this work, the relation between the reconstruction quality and the distribution of camera positions is analyzed and a new camera positioning strategy based on the properties of the synthetic object surface is proposed to achieve the same or even better reconstruction quality with equal or fewer viewpoints.

Index Terms— Multi-view, 3D reconstruction, camera positions

1. INTRODUCTION

Multi-view 3D reconstruction is one of the most important topics in the field of computer vision. The technique has been widely used in various applications and areas including 3D printing and virtual navigation system. It can also be used to restore and archive historical buildings and landmarks and to be used in driver assistance system to model the road and traffic environment.

The objective of image-based 3D reconstruction is to retrieve the most likely 3D shapes of an object or a scene with a set of given images. Multi-view 3D reconstruction is a general term of this technique which restores the geometric primitives in 3D space from more than one image. The proposed algorithms until 2006 are summarized and categorized into four classes in [4]. The first simple class is based on the cost function upon a 3D volume and the surface is extracted when the cost function of the 3D voxels is below a certain threshold. The second class is to iteratively adjust voxels or surface meshes to minimize the cost function. The most recent appealing approach based on total variational [3] is grouped into this class, where the cost function is constructed as a convex form and avoids trapping in the local minimum. The third class is to calculate the depth maps from image pairs and further to merge a set of depth map into a consistent scene [2, 5]. The fourth class is based on the feature growing. Furukawa et al. [1] proposed a method to generate quasi-dense accurate oriented patches. The approach extracts a set of sparse feature points in images and iteratively expands into the neighborhood. The redundant and noisy patches are filtered by enforcing geometric, photometric and visibility constraints.

Different multi-view algorithms utilize different cues from the images in variant ways. However, none of them investigates the reconstruction quality and the positioning of the cameras. They use as many images as available which may introduce false reconstruction or redundant information. The rest of this paper is organized as follows: Section 2 introduces the virtual image acquisition system. A set of experimental results with various positioning distribution of the cameras is analyzed in Section 3. In Section 4, a novel positioning strategy based on the properties of the object surface is proposed. Section 5 concludes the paper and gives the outlook for the future research.

2. VIRTUAL IMAGE ACQUISITION SYSTEM

In order to concentrate on the relation of camera positions and the reconstruction quality, we synthesize the camera as pinhole camera with graphics software OpenGL. The function `gluPerspective(fovy, aspect, zNear, zFar)` sets up the perspective transformation with the camera parameters shown in Fig. 1.

Here focal length $f = \mathbf{cot}\frac{fovy}{2}$ is the normalized focal length and is dependent of the field of the view $fovy$. When the perspective transformation is decomposed into intrinsic and extrinsic parts, we can write it as

$$\mathbf{P} = \mathbf{K}\left[\ \mathbf{R}\ \middle|\ \mathbf{t}\ \right]. \tag{1}$$

where $\mathbf{t} = -\mathbf{R}\widetilde{\mathbf{C}}$. The intrinsic matrix $\mathbf{K}$ is directly derived

978-1-5090-1265-7/15/$31.00

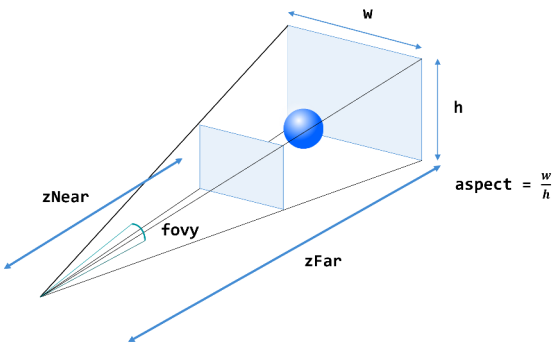

Fig. 1. Pin-hole Camera implemented with OpenGL, where w and h are the width and height of the image frame in pixel.

from the camera property.

$$\mathbf{K} = \begin{bmatrix} \frac{-w \times \cot\left(\frac{fovy}{2}\right)}{2 \times aspect} & 0 & \frac{w}{2} \\ 0 & \frac{-h \times \cot\left(\frac{fovy}{2}\right)}{2} & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

We take the plane object for example to clarify the coordinate used in our experiment setup. As depicted in Fig. 2, the whole scene is viewed towards $-Y$ axis and the principal camera is located at a point on $+Z$ axis and looks at $-Z$ direction. The plane object in color expands along X and Y. The distance between camera and object center is denoted by d and the angle of viewpoint is denoted by φ.

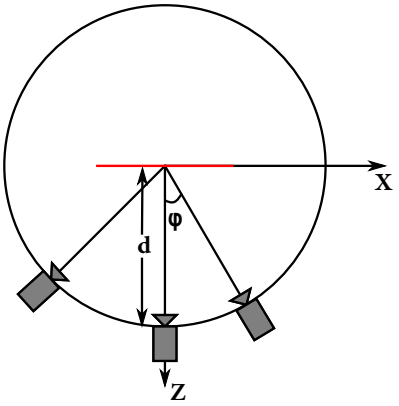

Fig. 2. Coordinate system of camera position

Accordingly, we can write the rotation matrix of rotating $-\varphi$ around Y axis as

$$\mathbf{R}(\varphi) = \begin{bmatrix} cos(-\varphi) & 0 & sin(-\varphi) \\ 0 & 1 & 0 \\ -sin(-\varphi) & 0 & cos(-\varphi) \end{bmatrix} \tag{3}$$

and the camera position can be represented by

$$\widetilde{\mathbf{C}} = \begin{bmatrix} d \times \sin\varphi \\ 0 \\ d \times \cos\varphi \end{bmatrix} \tag{4}$$

3. EXPERIMENTS

We start from the most simple shape, a plane, to find the reconstruction quality against different parameters and the analysis can be easily extended to other shapes. The reconstruction algorithm [5] that we choose to test is based on the depth-maps fusion. Besides the two metrics mentioned in [4], *the accuracy* and *the completeness*, we also take *the number of reconstructed vertices* and *the subjective visual observation* into account. For convenience, we first define the interested parameters. The *viewing direction*, denoted by φ, is defined to be the orientation of the camera lens deviated from the Z axis as shown in Fig. 2. The *distance* between the camera center to the object is represented as d. The physical size of object is represented as the form $[width \times height]$.

3.1. Viewing direction

Since the reconstruction algorithm investigated [5] requires at least three supporting views to reconstruct a point , we use three images with the viewing direction $[-\varphi, 0, \varphi]$ for each interested viewing direction φ. The camera position parameters are shown in Table. 1. Different colors in Fig. 4 indicates the portion of reconstructed points located within the tolerated distance. The colors in Fig. 5 show presummed error values used to compute the portion of reconstructed ground truth. The accuracy is better if its value is smaller since it is presented by tolerated error distance, whereas the completeness is better with larger value.

Parameter	Value
Viewing angle φ	$[10, 20, 30, 40, 50, 60]$
Distance d	5
Object shape	plane
Object size	$[6 \times 3]$

Table 1. Experiment parameters: Viewing direction

From both Fig. 4 and Fig. 5 we can observe that the accuracy and the completeness perform well when $\varphi = 20°$ and $\varphi = 30°$. However, once φ exceeds $50°$, the completeness decreases rapidly. When $\varphi \geq 60°$, only few points can be reconstructed. Therefore for a plane area under the same conditions, we should not put cameras with the viewing direction

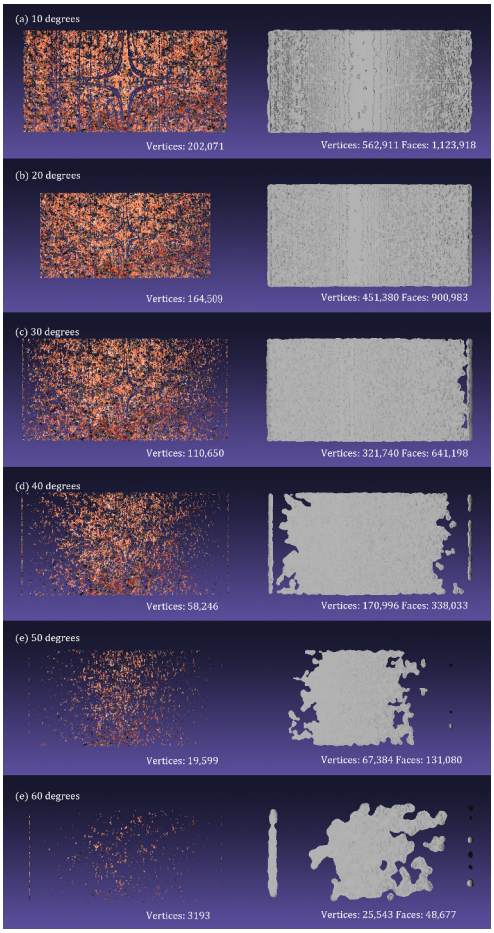

Fig. 3. Reconstruction versus viewing direction

which is too large. The threshold can be adjusted according to the quality requirement.

3.2. Distance

As in the last experiment, we use three views for simulation The viewing directions are fixed to $[-20°, 0, 20°]$. The parameters under investigation are shown in Table. 2 and the experiment results are shown in Fig. 7 and Fig. 8. Comparing to the viewing direction, the distance affects the reconstruction quality less. Both the accuracy and the completeness decrease when the distance is increased to 10, where the distance is twice of the width of the object. Under the projective transformation, the projected plane occupies only a very small area in the center of the image. Therefore the limitation of visibility in terms of distance can be more relaxed.

Parameter	Value
Viewing angle φ	20
Distance d	$[4, 6, 8, 10, 12, 14]$
Object shape	plane
Object size	$[6 \times 3]$

Table 2. Experiment parameters: Distance

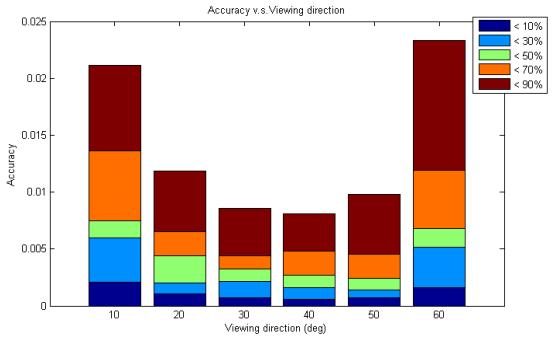

Fig. 4. Accuracy versus viewing direction

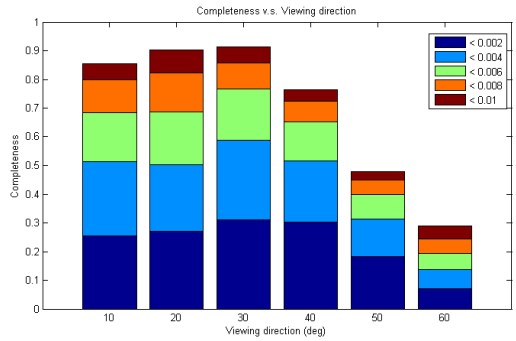

Fig. 5. Completeness versus viewing direction

3.3. Texture

Since the reconstruction method we used is based on the search of point correspondences, which are described by the feature descriptors, the complexity of the texture determines how well the feature points can be retrieved. It is hard to define a universal metric for the texture properties. Here, the standard deviation of the texture luminance is used to give a sense of how "rich" the texture is. In this experiment, the accuracy and the completeness are not calculated. Instead, only the visual representation with the standard deviation of texture luminance and the numbers of reconstructed points is provided in Fig. 9.

Parameter	Value
Viewing angle φ	20
Distance d	5
Object shape	plane
Object size	$[6 \times 3]$

Table 3. Experiment parameters: Texture

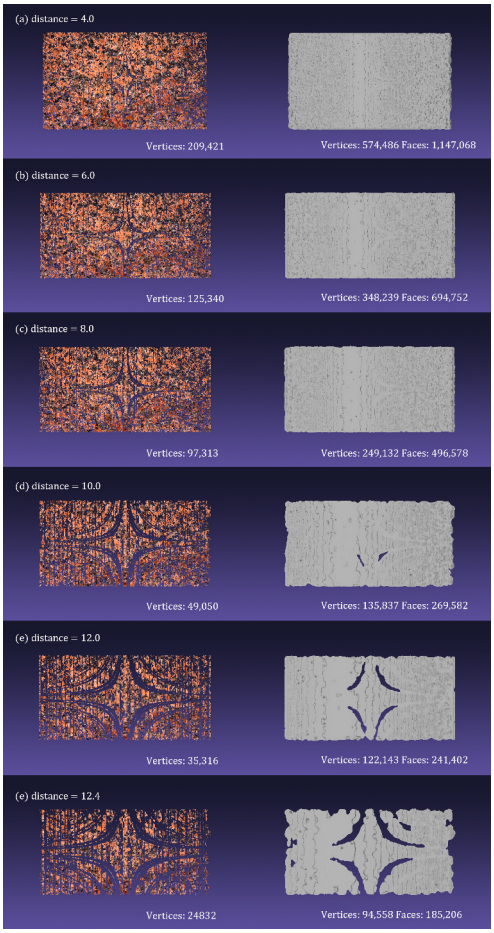

Fig. 6. Reconstruction versus distance

4. OPTIMIZED CAMERA POSITIONING STRATEGY

4.1. Geometric Analysis

The property of shape on the object surface constraints the camera positioning. As in the signal theory, where an arbitrary signal in one dimension can be represented by the linear combination of the sinusoidal waves, we attempt to simulate an arbitrary form with the linear combination of the sinusoidal wave planes. Consider an object with sinusoidal property along the plane XOZ shown in Fig. 10 , the camera is assumed to be oriented toward the center of the red concave part. The sinusoidal function can be therefore described as $y = \alpha \sin(\beta x)$.

As shown in Fig. 10, only cameras in the blue area can theoretically see the whole concave part. Our first target is the blue tangent line and the corresponding θ, which determines the angle of view in this area. To find the tangent line equation, we first determine the derivative of y with respect

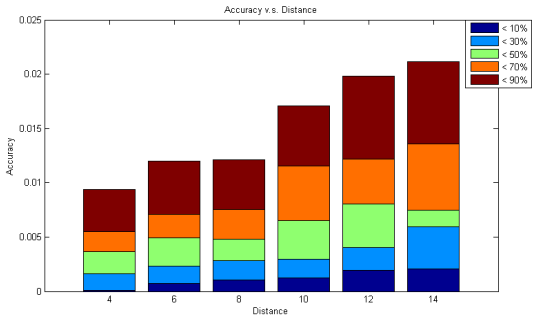

Fig. 7. Accuracy versus distance

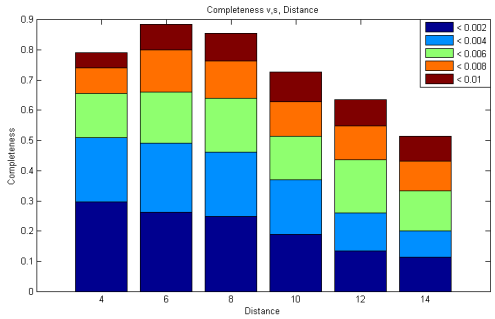

Fig. 8. Completeness versus distance

to x as

$$y\prime(x) = \alpha\beta\cos(\beta x) \tag{5}$$

and the line that is interested in is on the blue point, $(\frac{\beta}{\pi}, 0)$, and the slope of blue tangent line at this point is

$$y'\left(\frac{\beta}{\pi}\right) = -\alpha\beta \tag{6}$$

then the angle θ can be derived as,

$$\theta = \tan^{-1}(-\alpha\beta) \tag{7}$$

Next to be considered is the yellow region, where the view is partially blocked. The critical angle of view is φ. Any view of point with a smaller angle with respect to x axis than φ cannot detect the red concave part. To determine the angle φ, it is necessary to find the green point (x_φ, y_φ). It can be determined as follows,

$$\begin{cases} y = y'(x)x = (\alpha\beta\cos(\beta x))x \\ y = \alpha\sin(\beta x) \end{cases} \tag{8}$$

Let $t = \beta x$, we have

$$\begin{cases} y = \alpha t\cos t \\ y = \alpha\sin t \end{cases} \tag{9}$$

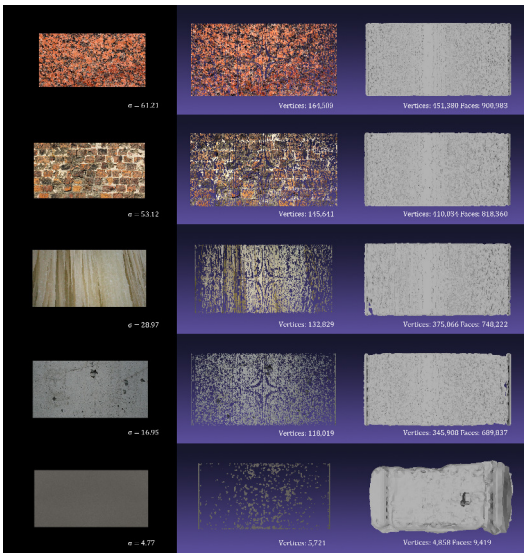

Fig. 9. Reconstruction versus texture

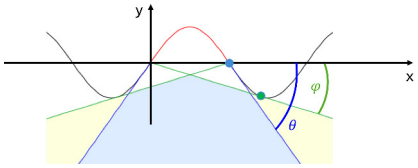

Fig. 10. Concave part of sinusoidal wave

$$t_\varphi = \tan t_\varphi \tag{10}$$

The analysis of convex part is similar to the concave one as shown in Fig. 11. Blue area indicates the camera positions with the visibility of the whole convex part and yellow area indicates partial visibility. Although the areas are determined slightly different from the concave part. The definition of the slopes and the corresponding viewing directions, θ and φ , are the exactly same and the derivation is identical to the concave part. With these results, we can obtain the visibility information if a region of shape can be modeled as a concave or convex part of a sinusoidal function.

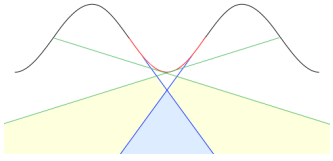

Fig. 11. Convex part of sinusoidal wave

4.2. Positioning Strategy

Take an object shown in Fig. 12 as an example. The form of the object can be described with the fucntion

$$y = 0.7\cos\left(\frac{\pi}{6}x\right) + 0.4\cos\left(5x - 0.3\right) \tag{11}$$

We segment the object into different parts and model them as simple shape and consider their visibilities separately. In Fig. 12, the regions around the blue points, whose curvatures are locally maximal, can be modeled as sinusoidal functions locally and they are called *peak points*. On the other hand, the regions near to the red points can be modeled as plane locally. The red points with locally minimal curvatures are called *plane points*. All points in union of peak points and plane points are called *key points*.

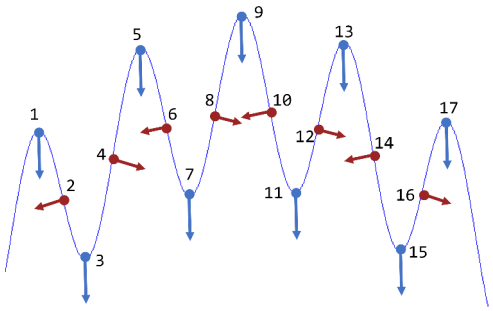

Fig. 12. Key points: blue dots are *peak points*, red dots are *plane points*

We first suppose that the object is a surface with the amplitude variation only along one dimension and all cameras are shooting at the center of object. Due to the relative weakness of DAISY descriptor against different scales that is utilized in the reconstruction algorithm in [5], the same distance d from camera center to the object center for all cameras is preferred. Moreover, the experiment results show that the distance affects the numbers of reconstructed points more significantly rather than the accuracy or the completeness. For the best visibility, we choose the minimal distance d such that the object can be seen in any viewing direction $[-90°, +90°]$. In the next step, we determine the *key points* including plane points and peak points, each with the corresponding normal vector and the range of viewing direction in terms of visibility. All the *plane points* are divided into two groups G_{left} and G_{right} by their normal vector directions with respect to the normal vectors of *peak points*. A binary tree with the candidate viewing direction φ as the value of a node is spanned. The root is the "principal viewing direction", $0°$ in this example. The left and right child of root is the direction of mean of vectors in G_{left} and G_{right}. For every descending level, we divide the step value of view directions by 2 and span it until the viewing directions have spread out a reasonable range. A reasonable range is defined if the tree satisfies one of the 3 conditions:

1. The values of leaf nodes have exceeded the designed limitation range, e.g., $[-90°, +90°]$, in which the viewing directions are considered as "principal".

2. The values of leaf nodes have exceeded the visible range for all key points.

3. The difference of values of leaf nodes from different subtrees is less than the step value of that level

We again take Fig. 12 as an example. Plane points 2, 6, 10, 14 are grouped as G_{left} and points 4, 8, 12, 16 are grouped as G_{right}. The mean value of viewing directions in G_{left} and G_{right} are $-60°$ and $60°$. The step value of the first level is $60°$. In this example, in order to get the simple value for calculation convenience with acceptable errors, we apply the rounding to integer. It spans until the third level since the values of leaves $\pm 105°$ have exceeded designated limitation range $[-90°, +90°]$. The spanning tree is shown in Fig. 13

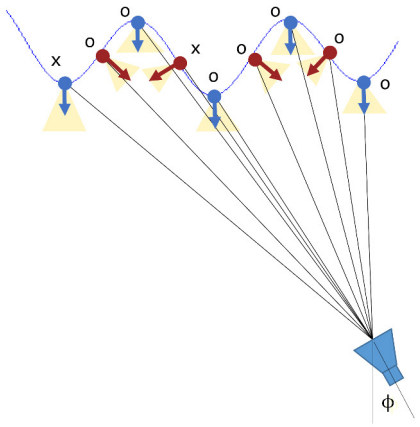

Fig. 14. Test of visibility

After calculating the values of these candidates, we start to decide which candidates to use. Notice that we consider all candidates equivalently instead of level by level. As shown in Fig. 14, for each candidate, we test the visibility of every key points. If the direction from a key point to the camera center falls in the yellow area, which shows the visible viewing direction range, the key point is denoted as *visible* for this camera candidate and marked with an "o". Otherwise the key point is invisible and marked with "x". The visible viewing direction range of a point is either $[-60°, +60°]$ around the normal vector for plane nodes or the visibility range in terms of θ determined by the corresponding amplitude A and curvature κ for peak points. After we complete the visibility test, we have a table of all cameras containing a list indicating the visibility of every key point. The good candidates are determined with a greedy strategy. Since each key point needs 3 supporting camera, we keep finding a candidate that has the most visible key points *and* covers at least one point with less than three supporting cameras before the termination. This algorithm will stop once all key points have at least three supporting cameras. In the example, the nodes with circle in Fig. 13 are the selected candidates.

Finally, we evaluate the reconstruction quality of this strategy. A reconstructed mesh from images taken by equiangularly positioned cameras. The number of cameras in both cases are the same.

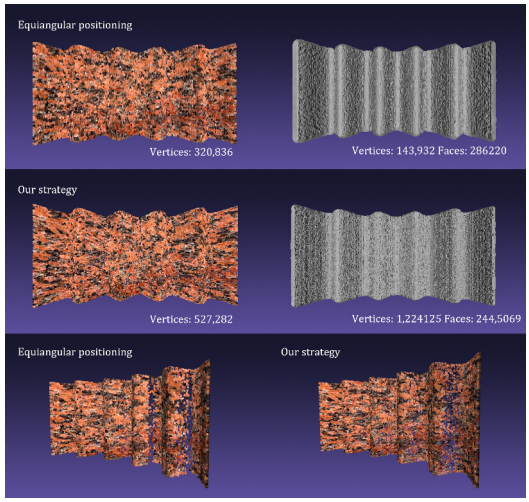

Fig. 15. Evaluation of proposed strategy, combination of sinusoidal waves

As shown in Table. 4 and Fig. 15, although the accuracy and the completeness is a little worse in our proposed strategy, the number of reconstructed points is much larger than those with trivial equiangular positioning. Also some points in the "deep" part of the shape are better reconstructed. We also evaluate this strategy by another object, a triangular wave. Since the definition of curvature can be problematic on an indifferentiable function. We simply set the visibility range of peak points as $[-45°, +45°]$. The parameters are summarized in Table. 5 and the visualization of results is shown in Fig. 16.

Again, the number of reconstructed with our strategy is much larger and the details are covered more plausible although the improvement is not as much as the previous object.

5. CONCLUSION

The reconstruction quality of Multi-view 3D reconstruction algorithms depends on the images and thus the positions of camera are crucial. Our objective is to find an optimized positioning strategy of cameras to get better reconstruction quality with equal or even less images. A series of experiments upon virtual image acquisition system is conducted to find the visibility limitation according to the simple shapes such as planes and sinusoidal waves. These results are used in our proposed strategy which is then demonstrated to improve the overall

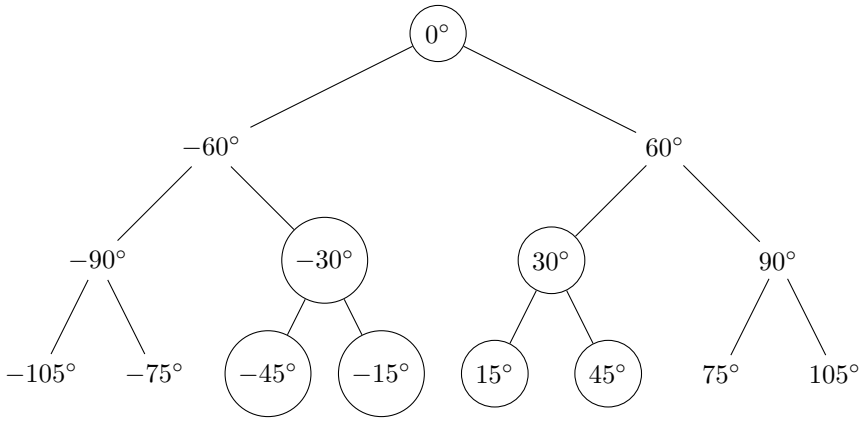

Fig. 13. Spanning tree of viewing direction

Parameter	Equiangular	Our strategy
Viewing angle φ	$[-67, -45, -22, 0, 22, 45, 67]$	$[-45, -30, -15, 0, 15, 30, 45]$
Number of view	7	7
Distance d	5	5
Object shape	combination of sinusoidal waves	combination of sinusoidal waves
Object size	$[6 \times 3]$	$[6 \times 3]$
Accuracy(90%)	0.015605	0.019
Completeness($d = 0.01$)	0.837	0.818

Table 4. Parameters and evaluations of proposed strategy, combination of sinusoidal waves

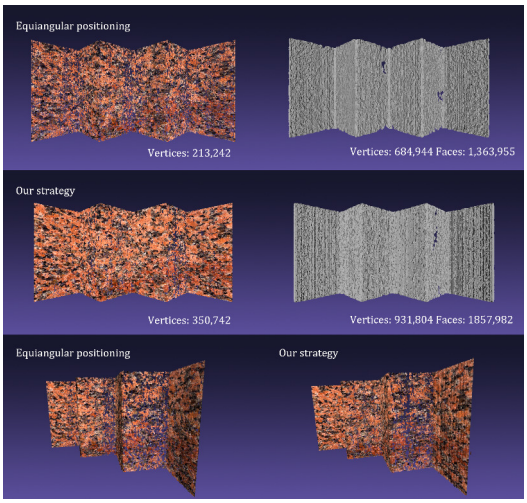

Fig. 16. Evaluation of proposed strategy, triangular wave

number of reconstructed points and perform better in some regions.

The experiments conducted in this work contain some basic parameters. However, there are still some other parameters can be varied for further potential experiment designs. For instance, in this work the cameras are always shooting at the center of objects. Degrees of freedom to shoot at different points may better catch local features and improve the reconstruction quality. Moreover, if we do not know the exact description of the shape, we can only visually estimated it. However, this information is possible to be retrieved by other approaches. The reconstruction from equiangular positioning can be done at a primary stage to estimate the geometric property of the surface and the camera positions can be adjusted iteratively. And more alternative multi-view reconstruction methods besides the depth-map fusion can be under investigation.

6. REFERENCES

[1] Yasutaka Furukawa and Jean Ponce. Accurate, Dense, and Robust Multiview Stereopsis. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 32, pages 1362–1376, 2010.

[2] Pau Gargallo and Peter Sturm. Bayesian 3D Modeling from Images using Multiple Depth Maps. In *IEEE Workshop on Motion and Video Computing*, volume 2, pages 885–891, San Diego, United States, June 2005.

[3] Kalin Kolev. *Convexity in Image-based 3D Surface Reconstruction*. PhD thesis, Computer Vision Group, Department of Computer Science, Technical University Munich, 85748 Garching, Germany, 2011.

Parameter	Equiangular	Our strategy
Viewing angle φ	$[-62, -37, -12, 12, 37, 62]$	$[-33, -22, -11, 11, 22, 33]$
Number of view	6	6
Distance d	5	5
Object shape	triangular wave	triangular wave
Object size	$[6 \times 3]$	$[6 \times 3]$
Accuracy(90%)	0.009623	0.0177
Completeness($d = 0.01$)	0.9024	0.771

Table 5. Parameters of proposed strategy, triangular wave

[4] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A Comparison and Evaluation of Multi-view Stereo Reconstruction Algorithms. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, pages 519–528, 2006. http://vision.middlebury.edu/mview/.

[5] Yu Zhao and Ningqing Qian. Fusion of Depth Maps in Multi-view Reconstruction. In *International Student Conference on Electrical Engineering POSTER*, Prague, Czech Republic, May 2015.