State-Based Visibility for 3D Reconstruction from Multiple Views

Liuxin Zhang, Yumo Yang, Yunde Jia

State-Based Visibility for 3D Reconstruction from Multiple Views

Liuxin Zhang, Yumo Yang and Yunde Jia

Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081 PRC
{zhangliuxin,yangyumo,jiayunde}@bit.edu.cn

Abstract. Estimating visibility is one of the most important steps for multi-view reconstruction using volumetric scene representation. In this paper, we propose a simple approach to estimating visibility based on the current state of a scene which is implicitly represented as the zero level set of a function. Our method can determine the regions of a 3D space visible to several given viewpoints efficiently. Using this visibility method, we also present a new variational formulation for multi-view reconstruction. We cast the multi-view reconstruction problem as an optimization of a novel energy functional amenable for minimization with an Euler-Lagrange driven evolution. The proposed algorithm has been applied to both synthetic and real datasets with promising results.

1 Introduction

Volumetric multi-view stereo reconstruction, originally introduced by Seitz group [1, 2], has recently been shown to produce 3D models from photographs or video sequences with fairly high quality [3, 4]. The basic principle in volumetric reconstruction is to find a classification for all elements (*voxels or grid points*) within a discrete volume space whether they belong to the surface of a 3D object or not.

Basically, the surface of a 3D object is nearly Lambertian, i.e., only elements belonging to this surface should have a consistent appearance in the input images (*be photo-consistent*). Therefore, the most central aspect of all the existing volumetric multi-view stereo reconstruction techniques is the estimation of the so called *photo-consistency* of a given element.

We assume that all the views (*cameras*) are well calibrated within the global world coordinate system, i.e., for each point in the world space, it is possible to determine the coordinates of its projection onto each view. In this case, before estimating the photo-consistency of a given point, we need to compute the visibility of this point with respect to all the views. This visibility problem can be described as follows: given a state of a scene, determine whether a point is occluded by other points from the view. This problem is one of the most important steps for multi-view reconstruction based on a volumetric scene representation.

In this paper, we propose a simple approach to visibility estimation based on the current state of a scene which is implicitly represented as the zero level

set of a function. Our method can obtain the visibility information of each point in two or three dimensional space to all the views through an efficient recursion strategy. Using this visibility estimation, we also present a novel model for 3D reconstruction from multiple views in a variational framework. In contrast with some other multi-view reconstruction methods, the main benefit of our method is its independence from initialization.

In the following, we first review related work in Section 2 and discuss our visibility algorithm in Section 3. In Section 4, we describe the reconstruction method and its conversion to an optimization problem with efficient implementation by an Euler-Lagrange driven evolution. The experimental results are shown in Section 5 to validate our proposed approach and conclusions are made in Section 6.

2 Related Work

The overview of state-of-art algorithms for multi-view stereo reconstruction reported by Seitz [5] shows a massive domination of volumetric method. Nearly all of them use a photo-consistency measure such as normalized cross correlation (NCC) or sum of squared differences (SSD) to evaluate how consistent a reconstruction is with a set of images. Visibility plays a very important role in volumetric reconstruction because the only requirement to compute this photo-consistency is that camera visibility is available.

Lempitsky et al. [6] state that visibility estimation is arguably the most difficult problem in 3D reconstruction from multiple views. To estimate the true visibility of some surface element, one needs to know the true scene geometry and vice versa. Currently there are numerous algorithms for approximately solving this chicken-and-egg problem, which can be mainly divided into two categories: *state-based visibility* and *oriented visibility*. State-based visibility can be considered as a type of global visibility algorithm which determines the visibility information by the state of the whole scene. It is often used in the approaches which reconstruct the scene geometry based on iterative process [2, 7], and at each moment of time, a point is considered visible to a viewpoint if it is not occluded with current scene configuration (Fig. 1 left). By contrast, oriented visibility is a type of local visibility algorithm. The visibility of a patch to a viewpoint is estimated by using the patch position and normal direction. In oriented visibility, a local patch is considered visible to a viewpoint if and only if it locates in front of this viewpoint according to a predefined angle from the normal direction (Fig. 1 right). Although this local method is simple, sometimes when the global scene geometry is more complex, such as a surface with high curvature (Fig. 2 left) or a surface with several unattached parts (Fig. 2 right), it is not reliable to compute the visibility of a local patch only from its orientation because of the self-occlusion.

Our visibility algorithm falls into the first category, and we take the global scene geometry to obtain more reliable visibility information. Our work is similar to that of Tsai et al. [8], who describe an implicit ray tracing technique to

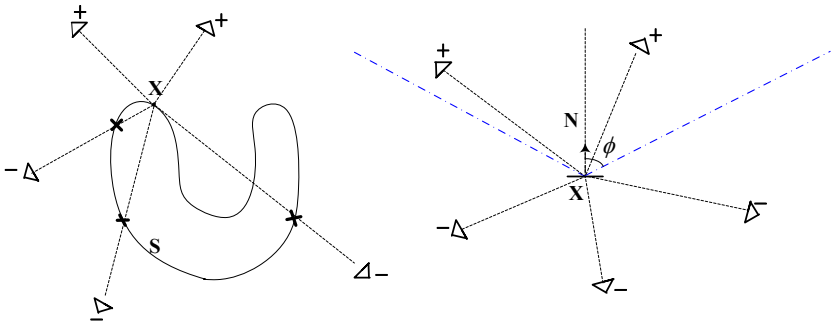

Fig. 1. Two approaches to visibility approximation (Lempitsky et al. [6]). On the left is a *state-based visibility*, the visibility information of a point **X** with respect to all the views is determined by the current global scene geometry **S**. On the right is an *oriented visibility*, a local patch (**X**, **N**) is considered visible from the viewpoints within a predefined angle ϕ from the normal direction **N**.

determine the visibility of a single viewpoint in two or three dimensional space. In their algorithm, a Cartesian grid is used to discretize the space and the viewpoint is supposed to lie on a grid node. They organize the entire grid in a special sweeping sequence so that each grid node can obtain its own visibility information to the viewpoint without violation. Although their method can solve the visibility problem of a single viewpoint quickly and efficiently, it can not be used directly in the field of multiple views. It is impossible for us to establish a uniform Cartesian grid to ensure all the views being on the grid nodes at the same time. In our work, we organize all the grid nodes in a more reasonable way that does not need the viewpoint to lie on the grid node. So, our algorithm is able to be used in any case including multiple views.

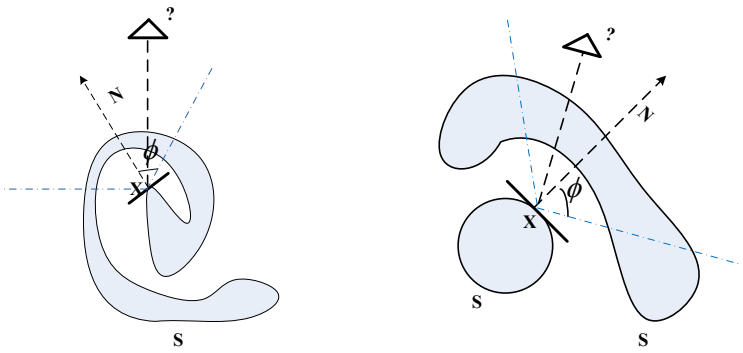

Fig. 2. The fallibility of oriented visibility in a scene with complex geometry.

3 State-Based Visibility in an Implicit Framework

3.1 The Level Set Method and Implicit Representations

The level set method proposed by Osher and Sethian [9] has found many applications in recent years. The basic idea behind this method can be thought of using a higher-dimensional representation for the problem of *front tracking*. This method provides us with a robust representation of even a more complex surface and its evolution.

We assume that the whole scene (or its part we are interested in) is located within some bounding volume $\Omega \subset \Re^3$. Each allowed scene configuration is characterized by some occupied subvolume $M \subset \Omega$ with the piecewise-smooth boundary $S = \partial M$ (the scene surface). We define a Lipschitz smooth function $\phi : \Re^3 \to \Re$ in Ω, which represents the scene surface as the zeros of ϕ, with the following properties:

$$\begin{cases} \phi(\mathbf{X}) < 0 & \text{for} \quad \mathbf{X} \in M, \\ \phi(\mathbf{X}) > 0 & \text{for} \quad \mathbf{X} \notin \overline{M}, \\ \phi(\mathbf{X}) = 0 & \text{for} \quad \mathbf{X} \in \partial M = S, \end{cases} \tag{1}$$

where $\overline{M}$ denotes the closure of the set M. ϕ is commonly referred to as the *level set function*. With such a choice of ϕ, we no longer have to worry about the scene surface. We should pay more attention to the domain of ϕ, and the related calculations are independent of the location and topology of S.

3.2 Implicit Ray Tracing

We now set up the foundation of our algorithm and derive properties of ray tracing of a single viewpoint in an implicit framework.

Let $\mathbf{V}_0$ denote the position of the viewpoint in Ω, and any other point in Ω is represented by $\mathbf{X}_i (i = 1, 2, 3, \ldots)$. Then, a point $\mathbf{X}_1$ is considered invisible to the viewpoint $\mathbf{V}_0$ if it lies inside the scene surface or if it is occluded by another point $\mathbf{X}_2$ in the radial direction $\overrightarrow{\mathbf{X}_1\mathbf{V}_0}$ (Fig. 3).

We observe further that the visibility status of points, which share the same radial direction, satisfy a causality condition: if a point is occluded, then all other points behind it in the same radial direction are also occluded. This means that a point $\mathbf{X}_1$ is considered invisible to the viewpoint $\mathbf{V}_0$ if and only if there is at least one point in the line segment $\overrightarrow{\mathbf{X}_1\mathbf{V}_0}$ that lies inside the scene surface. Under this condition, the idea of out algorithm is to start from each point in Ω, shoot a ray towards the viewpoint, and check whether there is a point being inside the scene surface in this ray.

Sometimes, with an explicit representation of the surface it could be difficult to determine whether a point is inside or outside the surface. In contrast, implicit surface representations include some very powerful geometric tools. We can determine which side of the surface a point is on simply by looking at the local sign of the level set function ϕ. That is, $\mathbf{X}$ is inside the surface when $\phi(\mathbf{X}) < 0$,

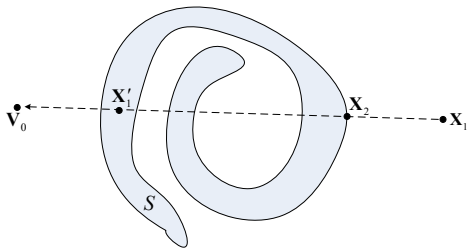

Fig. 3. Ray Tracing. A point is considered invisible to the viewpoint only in two cases. The first case is that it lies inside the scene surface and therefore it is occluded by the scene surface. The second case is that although it lies outside the scene surface, it is still occluded by another point before it in the same radial direction. $\mathbf{X}'_1$ is invisible to $\mathbf{V}_0$ in the first case and $\mathbf{X}_1$ is invisible to $\mathbf{V}_0$ in the second case.

outside the surface when $\phi(\mathbf{X}) > 0$, and on the surface when $\phi(\mathbf{X}) = 0$. Thus, if we use another function $\psi(\mathbf{X})$ to implicitly represent the visibility status of point $\mathbf{X}$, the problem becomes to compute

$$\psi(\mathbf{X}) = \min_{\xi \in \overrightarrow{\mathbf{XV}_0}} (\phi(\xi)), \tag{2}$$

where ξ is a point on $\overrightarrow{\mathbf{XV}_0}$. If $\psi(\mathbf{X})$ is negative, then $\mathbf{X}$ is occluded. As a matter of fact, our visibility algorithm is an approximation of this formula.

3.3 Numerical Implementation

In order to implement Eq. (2) numerically, a uniform Cartesian grid defined as $\{(x_i, y_j, z_k)|1 \leq i \leq m, 1 \leq j \leq n, 1 \leq k \leq l\}$ in 3D space is used to discretize the closed volume domain Ω, and all of the functions and quantities are defined on this grid. We will use the conventional indexing scheme to enumerate the grid nodes in Ω; i.e., $\mathbf{X}_{i,j,k} = \mathbf{X}_{ll} + \Delta x(i, j, k)$, where $\mathbf{X}_{ll}$ is the lower left corner of Ω, and Δx is the mesh size. With this notation, we define a voxel to be the cube with vertices $\{\mathbf{X}_{i+p,j+q,k+r} : p, q, r \in \{0, 1\}\} \subset \Omega$ for some (i, j, k). We need to update Eq. (2) in order to take advantage of this Cartesian grid.

At each grid point $\mathbf{X}$, we first find the voxel containing $\mathbf{X}$. From this voxel, we find the face F that intersects the ray segment $\overrightarrow{\mathbf{XV}_0}$. If $\overrightarrow{\mathbf{XV}_0}$ intersects F at a point, we denote this intersection by $\mathbf{X}'$ (Fig. 4 left). If $\overrightarrow{\mathbf{XV}_0}$ lies in F, this problem is reduced to a 2D problem, and $\mathbf{X}'$ is the intersection of $\overrightarrow{\mathbf{XV}_0}$ and an edge of F (Fig. 4 right). In fact, $\mathbf{X}'$ can be regarded as the closest point to $\mathbf{X}$ in $\overrightarrow{\mathbf{XV}_0}$ within the accuracy of the Cartesian grid. Hence, we use a recursion to rewrite Eq. (2) as

$$\psi(\mathbf{X}) = \min(\psi(\mathbf{X}'), \phi(\mathbf{X})). \tag{3}$$

Note that if $\psi(\mathbf{X}') < 0$ or $\phi(\mathbf{X}) < 0$, then $\psi(\mathbf{X})$ will be less than zero in the end. It means that $\mathbf{X}$ is invisible if $\mathbf{X}'$ is invisible or if $\mathbf{X}$ lies inside the scene

surface. Therefore, Eq. (3) is essentially the causality condition of visibility we mentioned above.

In most cases, $\mathbf{X}'$ does not lie on the grid node, and we need to interpolate the value of ψ from the grid points closest to $\mathbf{X}'$. For example, we use $\psi(A_i), i = 1, 2$, for linear interpolation of $\psi(\mathbf{X}')$ in 2D case (Fig. 4 right) and use $\psi(A_i), i = 1, 2, 3, 4$, for bilinear interpolation of $\psi(\mathbf{X}')$ in 3D case (Fig. 4 left). So, the values of ψ on all the neighboring grid points around $\mathbf{X}'$ must have been computed before estimating the value of $\psi(\mathbf{X})$. A key part of this visibility algorithm is to organize all the grid points in Ω in a special sequence to make this interpolation procedure feasible. We describe such a sequence in detail below, and use a queue data structure denoted by Q to store this sequence for convenience.

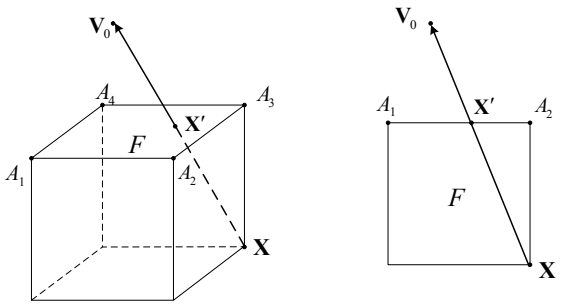

Fig. 4. An interpolation in 2D and 3D space.

Suppose the grid in Ω is bounded by 0 and nx in the X direction, 0 and ny in the Y direction and 0 and nz in the Z direction. We first find the cube voxel C_0 in Ω containing the viewpoint $\mathbf{V}_0$, i.e., $\mathbf{V}_0 \in C_0 := [x_{i_0}, x_{i_0+1}) \times [y_{j_0}, y_{j_0+1}) \times [z_{k_0}, z_{k_0+1})$. We can therefor start from this voxel C_0 and store the grid points into Q following the ray directions outwards. In order to represent our idea simply, we use three parameters denoted by $s1$, $s2$ and $s3$ to describe multiple directions from C_0 for i, j and k index respectively. The relationship between s1 and i is as follows:

$$\begin{cases} i = i_0 - 1 \text{ to } 0 & \text{when } s1 = -1, \\ i = i_0 \text{ to } i_0 + 1 & \text{when } s1 = 0, \\ i = i_0 + 2 \text{ to } nx & \text{when } s1 = 1, \end{cases}$$

and the relationship between $s2$ and j, $s3$ and k is similar to this. In the first step, we put the grid points $\mathbf{X}_{i,j,k}$ where (i, j, k) satisfy $(s1, s2, s3)=(0, 0, 0)$ into Q. Actually, $\mathbf{X}_{i,j,k}$ are the vertices of C_0 (Fig. 5a red points) in this case. In the second step, put the grid points $\mathbf{X}_{i,j,k}$ where (i, j, k) satisfy $(s1, s2, s3)=(0, 0, -1)$, $(0, 0, 1)$, $(0, -1, 0)$, $(0, 1, 0)$, $(-1, 0, 0)$ or $(1, 0, 0)$ into Q (Fig. 5b blue points). In the third step, put the grid points $\mathbf{X}_{i,j,k}$ where (i, j, k) satisfy $(s1, s2, s3)$ $=(0, -1, -1)$, $(0, -1, 1)$, $(0, 1, -1)$, $(0, 1, 1)$, $(-1, 0, -1)$, $(-1, 0, 1)$, $(1, 0, -1)$, $(1, 0, 1)$, $(-1, -1, 0)$, $(-1, 1, 0)$, $(1, -1, 0)$ or $(1, 1, 0)$ into Q (Fig. 5c green points). Finally,

put the remaining grid points $\mathbf{X}_{i,j,k}$ where (i,j,k) satisfy $(s1,s2,s3)$=$(-1,-1,-1)$, $(-1,-1,1)$, $(-1,1,-1)$, $(-1,1,1)$, $(1,-1,-1)$, $(1,-1,1)$, $(1,1,-1)$ or $(1,1,1)$ into Q (Fig. 5c yellow points).

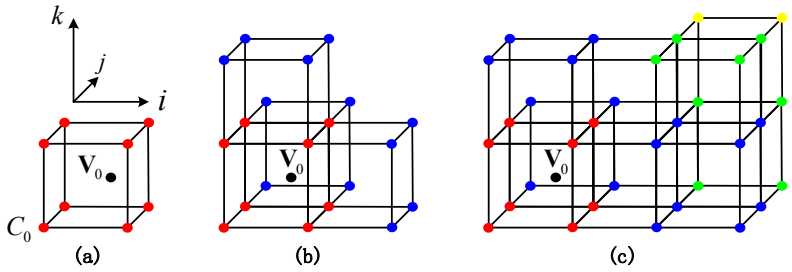

Fig. 5. A special sequence of the grid nodes

It is easy to see that each grid node in such a queue Q can be visited in a specified order that maintains the causality. Thus our algorithm reads:

```
for each X in the queue  Q
    if X is the vertices of C0
        Set ψ(X) = φ(X);
    else
        Find the face F and X' ∈ F;
        Update ψ(X) = min(ψ(X'), φ(X));
    end
end
```

In this algorithm, it is unnecessary for us to restrict the viewpoint $\mathbf{V}_0$ to the grid node of a Cartesian grid. So it can be used to solve the visibility problem easily in multiple views.

4 Variational Stereo with State-Based Visibility

Most of the multi-view reconstruction methods relying on state-based visibility need a good initialization of the scene surface [3, 7, 10]. If the current scene state is far from the true state, state-based visibility will approximate the true visibility with significant errors. Therefore, this type of global visibility algorithm is not thought to be a good choice in the situation when no good initialization is given [6]. In this section, we formulate a variational model for multi-view reconstruction based on our proposed visibility estimate. Compared with other methods, the main benefit of our model is its independence from the initialization. We show that our state-based visibility algorithm is still a good choice when no good initialization is given.

4.1 Energetic Formulation

The goal of our idea is to render a multi-view reconstruction as an energy minimization problem by assigning an energy cost based on visibility estimate. Since the surface of the unknown scene is supposed to be nearly Lambertian, patches belonging to the true surface should have similar colors in the viewpoints observing them. Therefore, the energy cost of a patch can be defined as

$$\mathbf{\Phi}(\mathbf{X}, \mathbf{N}) = \frac{1}{T} \sum_{i,j=1}^{m} \mathbf{\Phi}_{ij}(\mathbf{X}, \mathbf{N}), \tag{4}$$

where T is the number of items in the summation, m is the number of visible views to the current patch and $\mathbf{\Phi}_{ij}$ between two visible views i and j is based on the normalized cross correlation (NCC) and taken as [7]

$$\mathbf{\Phi}_{ij}(\mathbf{X}, \mathbf{N}) = 1 - NCC(\mathbf{I}_i, \mathbf{I}_j). \tag{5}$$

Its value range between 0 (best correlation) and +2 (worst). The term $NCC(\mathbf{I}_i, \mathbf{I}_j)$ is the normalized cross correlation of the projections of the patch in views i and j. We direct readers to [11] for more details on how to compute this term.

Summation in Eq. (4) is only computed for those patches which are visible to the two concerned views. Thus, estimating $\mathbf{\Phi}$ requires the step of computing the visibility status of the patches for all views. With our visibility algorithm, we rewrite Eq. (4) to involve the visibility status as follows:

$$\mathbf{\Phi}(\mathbf{X}, \mathbf{N}) = \frac{1}{T} \sum_{i,j=1}^{n} \chi(i, \mathbf{X}) \chi(j, \mathbf{X}) \mathbf{\Phi}_{ij}(\mathbf{X}, \mathbf{N}), \tag{6}$$

where n is the number of all views, $\chi(v, \mathbf{X})$ is the Heaviside function of $\psi(\mathbf{X})$ to the view v: $H(\psi(v, \mathbf{X}))$. We have

$$\chi(v, \mathbf{X}) = H(\psi(v, \mathbf{X})) = \begin{cases} 1 & \text{if } \psi(v, \mathbf{X}) \geq 0, \text{ i.e, } \mathbf{X} \text{ is visible to } v, \\ 0 & \text{if } \psi(v, \mathbf{X}) < 0, \text{ i.e, } \mathbf{X} \text{ is invisible to } v. \end{cases} \tag{7}$$

The overall surface energy cost for the scene configuration is then calculated by integrating the patches' costs over the scene surface S:

$$E(S) = \int_S \mathbf{\Phi}(\mathbf{X}, \mathbf{N}) dA. \tag{8}$$

Once $\mathbf{\Phi}$ has been chosen, a surface S which minimizes Eq. (8) can be regarded as the final scene surface (reconstruction result). Faugeras et al. [7] use a gradient descent evolution to solve this problem. However, the integral in Eq. (8) is only calculated on the surface S, and therefore the gradient descent evolution is driven in terms of this surface. For this reason, the final result in [7] tends to be sensitive to the initial position of the surface S. Since the surface to be reconstructed is represented by a level set function ϕ, the energy functional in Eq. (8) can be rewritten as

$$E(\phi) = \int_{\Omega} \mathbf{\Phi}(\mathbf{X}, \mathbf{N}) \delta(\phi) |\nabla \phi| dX dY dZ, \tag{9}$$

where $\delta(\phi)$ is the univariate Dirac function. Compared with Eq. (8), Eq. (9) is directly formulated in the level set domain, and therefore the minimization of it leads naturally to the evolution of ϕ. By introducing the term $\delta(\phi)|\nabla\phi|$ [12] in the integrand, the integral in Eq. (9) can be done on the whole 3D space Ω. So, the final result of ϕ that minimizes this energy functional is not sensitive to where it has started its evolution.

We finally write down the full energy functional guiding the reconstruction:

$$\begin{aligned} E(\phi) =& \alpha \int_{\Omega} \frac{1}{2} (|\nabla \phi| - 1)^2 dX dY dZ + \beta \int_{\Omega} \mathbf{\Phi}(\mathbf{X}, \mathbf{N}) \delta(\phi) |\nabla \phi| dX dY dZ \\ &+ \gamma \int_{\Omega} \delta(\phi) |\nabla \phi| dX dY dZ, \end{aligned} \tag{10}$$

where the first term is a penalizing term [13] which force the level set function to be close to a signed distance function as a numerical remedy for maintaining stable evolution. The third term is a smoothness term that can be used as a shape prior to improve the final reconstruction result. All the terms in the energy functional are weighted by constants α, β, γ.

4.2 Energy Minimization

The Euler-Lagrange equation for ϕ of the functional in Eq. (10) can be given by

$$\frac{\partial E}{\partial \phi} = -\alpha(\Delta\phi - \kappa) - \beta\delta(\phi)(\nabla \mathbf{\Phi} \cdot \mathbf{N} + \mathbf{\Phi} \cdot \kappa) - \gamma\delta(\phi)\kappa. \tag{11}$$

where Δ is the Laplacian operator. κ and $\mathbf{N}$ are the mean curvature and unit normal of the surface respectively, which can be calculated by

$$\kappa = \mathrm{div}(\frac{\nabla \phi}{|\nabla \phi|}), \quad \mathbf{N} = \frac{\nabla \phi}{|\nabla \phi|}. \tag{12}$$

Employing an artificial time variable $t > 0$, the steepest descent process for minimization of the functional $E(\phi)$ is the following gradient flow:

$$\frac{\partial \phi}{\partial t} = \alpha(\Delta\phi - \kappa) + \beta\delta(\phi)(\nabla \mathbf{\Phi} \cdot \mathbf{N} + \mathbf{\Phi} \cdot \kappa) + \gamma\delta(\phi)\kappa. \tag{13}$$

In practice, the Dirac function $\delta(\phi)$ in Eq. (13) is replaced by the function $\delta_\varepsilon(\phi)$ defined as

$$\delta_\varepsilon(\phi) = \frac{1}{\pi} \frac{\varepsilon}{\varepsilon^2 + \phi^2}, \tag{14}$$

with a small $\varepsilon > 0$. It gives a globally positive approximation to $\delta(\phi)$. We use $\delta_\varepsilon(\phi)$ in Eq. (14) instead of the one in [12, 13] because it can act on all level sets

of ϕ, while the one in [12, 13] can only act on a few level sets around $\phi = 0$. In this way, Eq. (13) has a tendency to obtain a global minimum of $E(\phi)$ with $\delta_\varepsilon(\phi)$ independently to the initial value of ϕ. The final evolution equation of ϕ is hence given by

$$\frac{\partial \phi}{\partial t} = \alpha(\Delta\phi - \kappa) + \beta\delta_\varepsilon(\phi)(\nabla\mathbf{\Phi} \cdot \mathbf{N} + \mathbf{\Phi} \cdot \kappa) + \gamma\delta_\varepsilon(\phi)\kappa. \tag{15}$$

The reconstruction result can be obtained from the zero level set of ϕ when this evolution is over.

5 Experimental Results

In this section, we present the experimental results of our method on both synthetic and real datasets.

In Fig. 6, we validate our visibility algorithm in some synthetic scenes that are composed of several spheres and cuboids (Fig. 6a left and Fig. 6b left). In this situation, since the true scene geometry is known, it can be very easy to determine the real visibility information of the elements with respect to several given viewpoints through our algorithm. In order to show our method usable in any multiple views, we take two cameras with known position as the viewpoints in our experiments. The visibility results (Fig. 6a right and Fig. 6b right) demonstrate the performance of our approach.

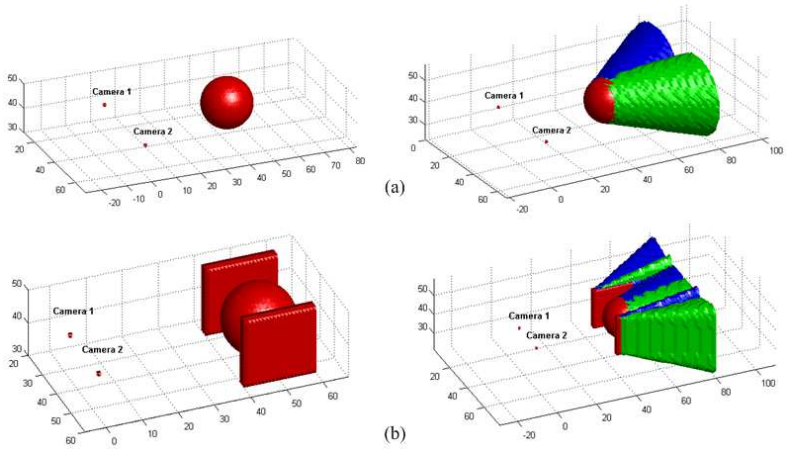

Fig. 6. Visibility results of two cameras applied to the synthetic scene with known geometry. The red surface represents the scene (occluders). The green and blue surface represent the boundary between visible and invisible regions to camera one and camera two respectively. (a) The visibility results of two cameras in a simple sphere scene. (b) The visibility results of two cameras in a scene with more complex geometry.

In Fig. 7, we apply our visibility algorithm to a number of reconstructions with the *Middlebury* datasets [14]. The data consists of two objects, a dinosaur and a temple (Fig. 7a), and three different sets of input images for each one, with viewpoints forming a sparse ring, a full ring, and a full hemisphere around the object. We perform our experiments on the *dinoSparseRing* (16 images) and the *templeSparseRing* (16 images) inputs. Eq. (15) is used to estimate the final 3D models through an evolution process and the initial zero level set is taken as a sphere. Fig. 7b illustrates some of those initial zero level sets (shown in blue) along with the reconstructed 3D models (shown in red). It is clear that our method can successfully reconstruct the surfaces for any initial level set function.

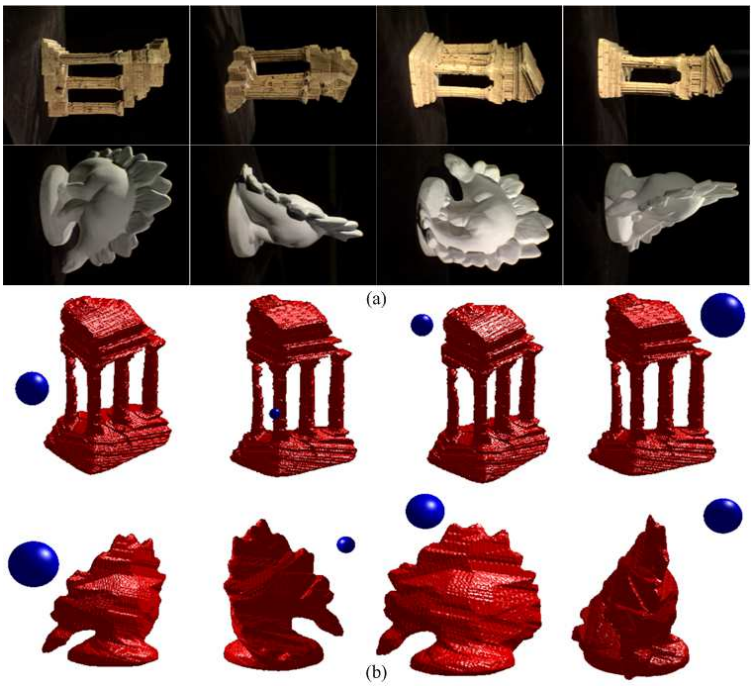

Fig. 7. The reconstruction results. (a) Some views of the dinoSparseRing and templeSparseRing datasets. (b) The reconstruction results of dino and temple (shown in red) with several different initializations (shown in blue).

6 Conclusions and Future Work

In this paper, we have presented a state-based approach to visibility estimation. It is based on an implicit ray tracing and can be used to solve the visibility problem of multiple views efficiently. With such a global visibility estimate, we have

also proposed a variational formulation for multi-view reconstruction. Our main advantage over other reconstruction methods relying on state-based visibility is the independence from the initialization due to the ability of our optimization to yield a global minimum of the energy functional. Future work will consist of improving the efficiency of the photo-consistency measure as well as developing better shape priors. Quantitative experiments are also needed to evaluate the accuracy and completeness of the reconstruction results. We also plan to incorporate strategies to apply our method to non-closed surfaces.

Acknowledgements

This work is partially supported by the Natural Science Foundation of China (60675021) and the Chinese High-Tech Program (2006AA01Z120).

References

1. Seitz, S.M., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. In: CVPR. (1997) 1067-1073
2. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. International Journal of Computer Vision **38** (2000) 199-218
3. Vogiatzis, G., Torr, P., Cipolla, R.: Mult-view stereo via volumetric graph-cuts. In: CVPR. (2005) 391-398
4. Esteban, C.H.: Stereo and silhouette fusion for 3D object modeling from uncalibrated images under circular motion. PhD thesis, Ecole Nationale Supérieure des Télécommunications (2004)
5. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR. (2006) 519-528
6. Lempitsky, V., Boykov, Y., Ivanov, D.: Oriented visibility for multiview reconstruction. In: ECCV. (2006) 226-238
7. Faugeras, O., Keriven, R.: Variational principles, surface evolution, PDE's, level set methods, and the stereo problem. IEEE Transactions on Image Processing **7** (1998) 336-344
8. Tsai, Y.H., Cheng, L.T., Burchard, P., Osher, S., Sapiro, G.: Dynamic visibility in an implicit framework. UCLA CAM Report. 02-06
9. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics **79** (1988) 12-49
10. Sinha, S.N., Pollefeys, M.: Multi-view reconstruction using photo-consistency and exact silhouette constraints: a maximum-flow formulation. In: ICCV. (2005) 349-356
11. Alejandro, T., Kang, S.B., Seitz, S.: Multi-view multi-exposure stereo. In: 3DPTV (2006) 861-868
12. Osher, S., Fedkiw, R.: Level set methods and dynamic implicit surfaces. Springer Verlag (2002), Chapter 1, 13-16
13. Li, C.M., Xu, C.Y., Gui, C.F., Fox, M.D.: Level set evolution without re-initialization: a new variational formulation. In: CVPR. (2005) 430-436
14. mview. http://vision.middlebury.edu/mview/.