Bottom and Concave Surface Rendering in Image-based Visual Hull

Jie Feng, Bencong Song and Bingfeng Zhou

Institute of Computer Science and Technology, Peking University, China

Abstract—**Bottom and concave shapes on object surface are difficult to reconstruct in image-based visual hull method. In this paper, we propose a simple but efficient method to solve these problems in regular image-based visual hull framework. With the help of a simple image acquiring platform which involves a glass and a mirror, we can capture images of the object from both upper and lower side at the same time. Using these images, silhouette cones necessary for reconstructing the bottom and the concave surfaces could be generated. Therefore the final rendering result of the object can be significantly improved in accuracy and reality, especially in the parts of bottom and concaves.**

Index Terms—**Bottom, Concave, Image-based Rendering, Shape-from-Silhouette.**

I. INTRODUCTION

Visual hull is an efficient technique for image-based rendering and modeling of real objects. Its main idea comes of "Shape-from-Silhouette", which takes a group of images of a target object as input, and produces an approximate convex hull that contains the object. Compared with traditional geometry-based methods, visual hull method has lower cost in data acquiring, and could produce more realistic result.

However, there are two important problems in visual hull rendering: the bottom of the object and concave shapes on the object surface are difficult to reconstruct, because of the fundamental character of this kind of methods.

Ideally, input reference images of the visual hull should be taken from different viewpoints around the object to reconstruct a more accurate visual hull model. However, the images of the bottom of the object usually cannot be acquired at the same time as other surrounding reference images. Lacking of them, models cannot be correctly reconstructed. Fig. 1 shows a typical example: the flat bottom of the teapot becomes a cone, and its texture is totally absent. If turning over the object and take another bottom image, the new image and the old ones do not lie in the same reference frame. Thus complex alignment calculations must be introduced to make them work together, which will obviously increase computing cost.

As for a concave object, such as a cup, the concave shape is impossible to be distinguished from the object silhouettes. Therefore, it can not be accurately rebuilt by regular visual hull methods. However, given some prior knowledge, we can

reconstruct an approximation of such concave surface, and the rendering result can be significantly improved.
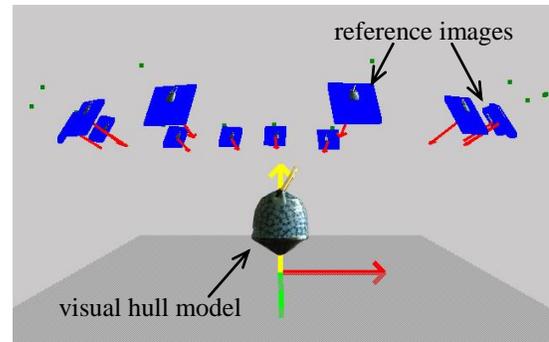

Fig. 1. The bottom problem in visual hull modeling: Lacking of images of the object bottom, reconstructed model is not correct both in shape and texture.

Hence, solving these two problems will be of great benefit to some application, such as archaeology or artwork exhibition, where the shape and texture of the bottom or the concave surface may be very important.

There have been some researches that enrolled planar mirrors in image-based modeling. For example, [1] use a single camera and two mirrors to reflect the scene to produce stereo images, and the geometry and calibration using point correspondences are discussed in this paper.

[2]-[3] set up a two mirror system to capture five views of an object in one single image and use them to construct the visual hull model. A remarkable advantage of this method is that the camera is not required to be calibrated. However, it cannot model either the bottom or the concave surface. Though multiple input images from different viewpoints could be merged into a large silhouette set, the relative position between the object and the mirrors must be fixed.

In this paper, we propose an improved visual hull rendering method to solve the bottom and concave problem. The method takes advantage of a new image acquiring platform, which involves a planar glass and a planar mirror. Utilizing this simple and efficient platform, we can easily obtain images of the target object and its bottom at the same time; images of the concave surface are also available. These images can be used as reference images of the Image-Based Visual Hull method [4] after simple transformations. With our improved algorithm, photo-realistic rendering result of the object from arbitrary viewpoint can be produced. The bottom and concave surface could be well approximated.

The related research works on visual hulls are briefly

reviewed in section II. In section III, we describe our new image acquiring platform in detail. The bottom rendering method in Section IV shows how virtual cameras can be used to reconstruct the object bottom; and the method of concave surface approximation is followed in section V. Finally, some experimental results are given in section VI, followed by the summary in the section VII.

II.   VISUAL HULL RECONSTRUCTION METHODS

The main idea of visual hull reconstruction comes from *Shape-from-Silhouette*. Each reference image is separated into foreground and background. The foreground mask, i.e. the silhouette, along with the calibration information of the camera, defines a back-projected cone in 3D space that contains the target object. Thus, the intersection of all silhouette cones forms a convex hull, i.e. the *visual hull* of the object.

There are mainly two sorts of visual hull construction methods: voxel-based and boundary-based methods [5].

Voxel-based methods [6]-[7] usually start from a working volume that contains the target object and quantized into voxels. The voxels are one by one put through tests: those lie inside all silhouette cones are preserved; the others are cleared. Therefore, the remaining voxels form the visual hull. These methods are able to reconstruct very complex objects, such as trees. However, they usually suffer from quantization artifacts and cannot get smooth modeling results.

In boundary-based methods [4, 8], silhouette cones are represented as boundary elements, such as surfaces or lines. The visual hull is constructed by computing the intersection of these elements, and the result could be composed by a group of surfaces patches, line segments, or points. Such methods consume less memory and run faster than voxel-based methods, and quantization artifacts could be avoided. If necessary, boundary-represented visual hull can also be triangulated into explicit 3D meshes.

2.1 Image-Based Visual Hull

In this paper, we follow the main framework of a method called *Image-based Visual Hull (IBVH)* [4]. It is a boundary-based method. In this method, the input is a group of images taken around the object with calibrated cameras. The silhouette cones of these reference images are represented as a bench of 3D rays that is emitted from the camera centers, each corresponds to a pixel in the silhouette. Then the intersection of the cones, i.e. the visual hull, is composed of a group of line segments. It is remarkable that in this method the computing is limited to the image space of the reference images, and the result is view-dependent, which makes the method quite efficient.

As illustrated in Fig. 2, the kernel algorithm of IBVH method includes three main steps:

1. For each pixel of the desired view, a ray emitted from the camera center and passing through current pixel is calculated. Using the calibration information and epipolar geometry theory, the projection of this ray on a reference image, i.e. the epipolar line could be computed.
2. The epipolar line intersects the 2D silhouette on the reference image, and results in a group of 2D intervals.

3. The 2D intervals are projected back to 3D space and get corresponding 3D segments on the viewing ray. Then the intersection of all the segments from all the reference images indicates the visual hull boundary at current pixel. Projecting the nearest endpoint of final 3D segments onto the reference images, the color of current pixel could be synthesized.
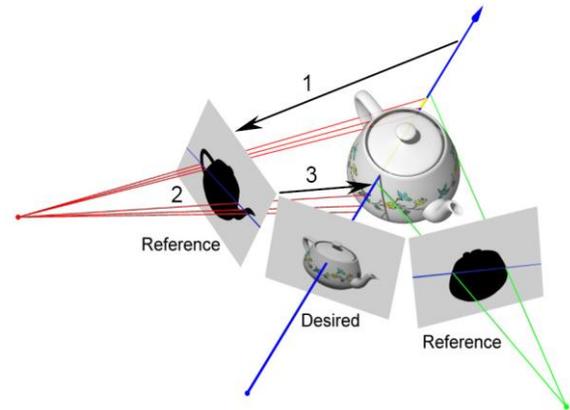

Fig. 2. The three steps of IBVH construction [4].

To reduce the computation cost and increase speed, the original algorithm sorts silhouette edges in so-called *bin* structures by their slopes during step 2. Here in our implementation, we make an improvement in the strategy of building *bins*. Instead of slopes, the silhouette edges are sorted by their direction angles, so that the intersection could be properly performed even when the epipole (the intersection point of all epipolar lines) falls in the object area.

As a boundary-based method, the resulting model of IBVH is implicitly represented. When necessary, it can be triangulated into meshes by methods such as Marching Cubes [9].

III.   NEW REFERENCE IMAGE ACQUIRING PLATFORM

Image-based visual hull is an efficient method to reconstruct 3D models of an object. However, it has difficulty in rendering the bottom of the object and concave surfaces. For the bottom problem, it is because all of the reference images must be registered to the same reference frame in the calculation, therefore the relative position of the object and the reference system must be fixed while the camera moving around the object. Thus, turning over the object and taking another bottom image will require further complex alignment calculation to make the images work together.

The concave problem comes from the fundamental idea of visual hull (the concave shapes are impossible to be distinguished in silhouettes), therefore is more difficult to solve. However, given some prior knowledge, we can still reconstruct an approximation of the concave. For this purpose, an image from the top of the concave region is also necessary.

In order to solve these problems, we design a simple but practical image acquiring platform. By the aid of this new platform, we can easily obtain the images of the object from both upper and lower side without alignment calculation.
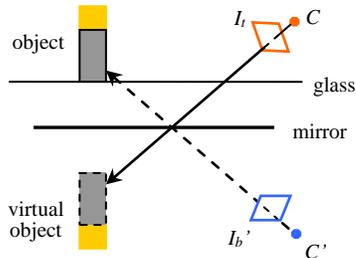
Fig. 3. New image acquiring platform enrolls a planar
Glass and a planar mirror.

As shown in Fig. 3, the platform enrolls a piece of planar glass board and a planar mirror. The object is placed on the glass, which lies right above the mirror. Therefore, the reflection of the object bottom in the mirror could easily be caught. Through this way, we can get two views of the object (from top and bottom side) at the same time in one single image. That means we need not acquire more images than regular visual hull method for bottom modeling. As for the concave object, such as a cup, the silhouette of the concave region would be useful in reconstruction. Hence, a special image should be acquired from the top of each concave region at the same time.

This platform greatly extended the range of possible camera viewpoints, and all of the acquired images are in the same reference frame. Therefore, no further alignment is needed.

We utilize a simple camera calibration system here. The colored concentric circles on the mirror are used as correspondence targets, and the parameters of the cameras could be calculated by common P4P calibration methods [10]. Note that the X-Z plane of the coordinate system is identical with the mirror plane; hence, the distance between the mirror and the glass becomes inessential. This will give more freedom to place the object and take images.

In some cases, the reflection and refraction of the glass will slightly affect bottom images. To minimize this effect, we choose as thin glass as possible. In this example, it is 5mm thick, and 800*800mm in size. The mirror is 500*500*5mm. A circular polarize lens will also help to alleviate the reflection on the glass.

IV.　BOTTOM RENDERING IN IBVH

When the reference images including both top and bottom of the object are acquired by our new platform, we propose an improved algorithm that could conveniently rebuild correct 3D model of the object with these images.

4.1 Virtual Camera and Virtual Image

When all of the reference images are acquired, the object in each image should be segmented from the background to get its silhouette. At this step, each reference image I will be separated into two: I_t contains the top image of the object, and I_b contains the bottom, as shown in Fig. 4.
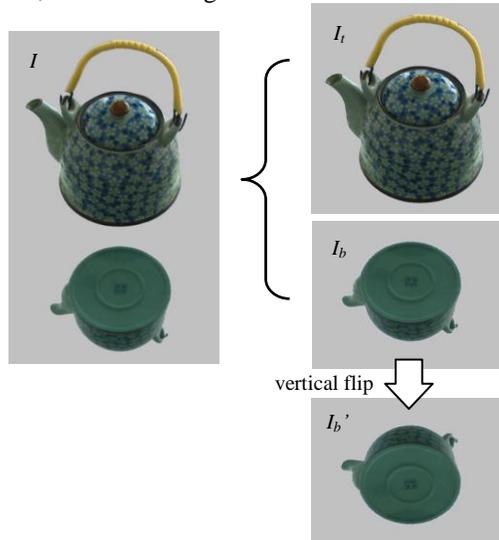

Fig. 4. An illustration of virtual images: Virtual image I_b' is identical
with vertically flipped bottom image I_b.

The top images, whose camera parameters are calculated by calibration methods, will be directly used in visual hull computing, while the bottom images need further transform to get correct parameters. As illustrated in Fig. 3, given one top image I_t, let p be the position of its corresponding camera C, and p' be its symmetrical point about the mirror plane. If we set a camera C' at p', with the symmetrical direction of C, then we will obtain a real bottom image I_b'. It is easy to verify in mathematics that I_b' is identical with the vertically flipped image I_b (Fig. 4). Thus, we name I_b' the *virtual image* at position p', and its C' the *virtual camera*. With the help of the concept of virtual camera and image, we are able to extend regular IBVH algorithm to reconstruct the object bottom correctly.

4.2 The Calibration of Virtual Cameras

As we have mentioned, the parameters of true cameras are computed by common calibration methods. In fact, the parameters of a virtual camera C' could be conveniently deduced from that of its corresponding true camera C.

According to the pin-hole camera model, the mapping from a 3D point (x, y, z) to 2D image pixel (u, v) could be represented as:

$$s \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = A \cdot \left[R \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + T \right]. \tag{1}$$

Here A is the internal parameter matrix of the camera C, while R and T are rotating matrix and translating vector, respectively:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \ T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}. \tag{2}$$

Because *C'* is actually identical with *C* in physics, so the internal parameters of *C'* need not recalculation. When assuming Y-axis of the world coordinate system is vertical to the mirror plane, the translating vector of *C'* could be represented as

$$T' = \begin{bmatrix} t_x \\ -t_y \\ t_z \end{bmatrix}. \tag{3}$$

When a 3D point is transformed to the image plane of *C*, if its rotating angles about X, Y and Z axes are *α*, *β* and *γ*, then when its symmetric point is transformed to *C'*, the rotating angles will be

-*α*, *β* and -*γ*. Therefore, we can deduce the rotating matrix of *C'* as

$$R' = \begin{bmatrix} r_{11} & -r_{12} & r_{13} \\ -r_{21} & r_{22} & -r_{23} \\ r_{31} & -r_{32} & r_{33} \end{bmatrix}. \tag{4}$$

So far, we have found the virtual camera's parameters. This step is quite simple and cost little calculation. With these parameters, a virtual bottom image could be considered as another "true" image, and be used together with other true images in the IBVH framework.

4.3 Trimming & Rendering Object Bottom

Visual hull itself is actually an approximation of the object, formed by the intersection of silhouette cones of the reference images. If all the images are taken at position high above the bottom plane, the bottom of the object cannot be correctly rebuild, and result in a pointing cone (Fig. 1).

One method to solve this problem is to take more images at positions with similar height as the bottom, and use their silhouettes to amend the visual hull. But when shooting at such "horizontal" positions, 2D calibration system would become invalid, and more complex 3D calibration system must be introduced. On another aspect, more reference images will also cost more storage space and computing time.

When such "horizontal" images are absent, we also have a trade off strategy to render approximate object bottom with both efficiency and accuracy. According to the platform setup, we can assume that the bottom plane is parallel with the mirror plane. Then, the height of the bottom plane could be calculated: First, we can mark several (usually 10~20) points at the edge of the object bottom in reference images. Their corresponding 3D points on the visual hull can be easily found. Then, the average Y-coordinate of these points can be considered as an approximation of the bottom height.

When the object is rendered from a new viewpoint, different strategies are used according to the height of the viewpoint: if it is higher than the bottom plane, the rendering is the same as regular IBVH method; if it is lower than the bottom plane, i.e. the camera would see the bottom, we should render each pixel in the new view with the following steps:

1. For current pixel, calculate its corresponding 3D point as in regular IBVH method.
2. When the corresponding point lies higher than the bottom plane, directly calculate its texture for current pixel.

3. When the corresponding point lies lower than the bottom plane, calculate the intersection point of the viewing ray through current pixel and the bottom plane, and consider it as the new correspondence of the pixel. That means we shall "push" the point back on to the bottom plane along the viewing ray (Fig. 5(a)). Then, correct texture of current pixel is calculated according to the new corresponding point.

When getting texture for current pixel, we project its 3D corresponding point onto different reference images. A weighted average of the colors of these projected pixels (where images nearer to the viewpoint will have larger weight) is considered as the final color of current pixel.

Therefore, all of the pixels with correct colors form the desired new view, while the incorrect cone at the bottom can be removed. An example is given in Fig. 5(b) and 5(c).

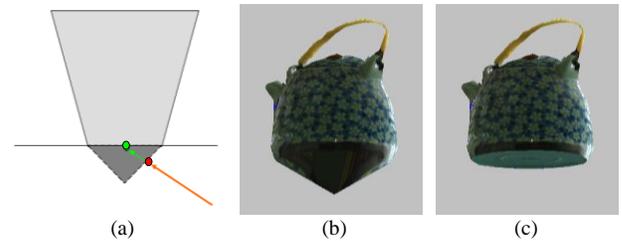(a)   (b)   (c)

Fig. 5. Trimming the bottom of the object with our improved method. (a) "Pushing" the bottom point back on to the bottom plane. (b) Result by regular IBVH. (c) Result by our method.

V.   CONCAVE SURFACE APPROXIMATING

As described above, it is difficult to reconstruct concave shapes in IBVH, because concaves cannot be distinguished only by using silhouettes. Fig. 7(b) shows an example: without concave information, the inner surface of the cup becomes a convex cone at the top; both of its shape and texture are seriously distorted.

5.1 Concave Silhouette Cone

Note that the visual hull is the intersection of all silhouette cones, then if we can place a "*negative*" silhouette cone inside the concave region, the final intersection would became closer to the really object. This is a reasonable approximation, and the rendering result of the concave region would be improved.

For this purpose, we take a special image I_0 from right above the concave region (Fig. 6(a)) and separate its concave region from the background similar as other reference images (Fig. 6(b)). Hence, the silhouette of the concave region is obtained.

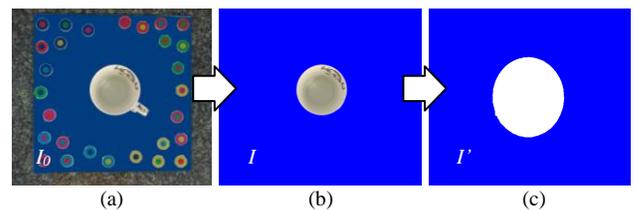(a)   (b)   (c)

Fig. 6. Synthesizing the virtual image of the concave surface. (a) Original reference image. (b) Segmented image. (c) Virtual image of the concave surface, with no texture.
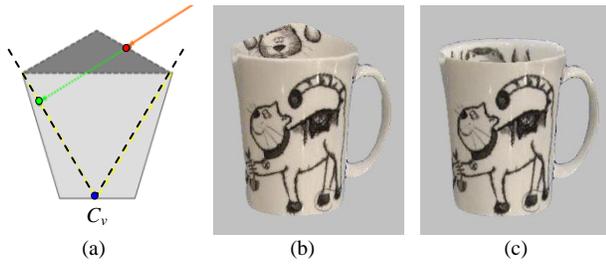
Fig. 7. Concave surface rendering. (a) Correcting the visual hull with a virtual "negative" silhouette cone. (b) Result by regular IBVH. (c) Result by our method.

Then, we set a virtual camera C_v right inside the concave region (Fig. 7(a)). The rotation matrix of C_v can be calculated according to that of the true camera of I_0, in the similar way as in bottom rendering, while its focal length should be properly set so that the concave region can be fully captured. Therefore, projecting the silhouette curves of I onto the image plane of C_v, a new virtual silhouette image I' c be synthesized (Fig.6(c)). It is only used for geometry reconstruction, so texture is unnecessary here.

This I' can be considered as "captured" by C_v. Its silhouette, along with the parameters of C_v, can be used to generate a virtual "negative" silhouette cone, which is an approximation of the concave surface. Subtracting this negative cone from the intersection of other silhouette cones will result in a concave approximation of the object.

5.2 Concave Surface Rendering

When the object is rendered from a new viewpoint, the virtual silhouette image I' provides only geometry information for the concave surface, while its corresponding real image I, and other reference images, provides texture information.

A different strategy is needed to render the concave region. For each pixel p_v in the new view, its 3D corresponding point v can be calculated as in regular IBVH method. Projecting v onto the real image I, let the projected pixel be p_r. If p_r lies in the background, p_v is not a concave pixel and can be rendered by regular method; otherwise, if p_r lies in the foreground, the rendering flow of p_v should be modified as follows:

1. Calculate the viewing ray through pv, and project it onto the virtual image I'.
2. The projected ray intersects the virtual concave silhouette with a group of 2D intervals. Projecting them back into the viewing ray and get a corresponding 3D segment set A.
3. Project the viewing ray onto other reference images as in regular method, and calculate the intersection of all the corresponding 3D segment sets, denoted as B.
4. Subtract A from B, then, the final resulting 3D segment set could be considered as the intersection of the viewing ray and the concave visual hull. The nearer intersection point is the new 3D correspondence of pixel pv and its correct texture can be retrieved in the same way as in section 4.3.

This algorithm helps to relocate those 3D points that should be on the concave surface and find their texture (Fig. 7(a)). It is remarkable that the subtraction calculation of the concave silhouette cone is also restricted to 2D image plane. That makes our new method as efficient as regular IBVH. An example of the improved rendering result is shown in Fig. 7(c), in which the protuberant cone on the top of the cup has been removed, and the texture inside the concave surface is also well retrieved.

VI. EXPERIMENTAL RESULTS

Combining the bottom and concave surface rendering method introduced above, we can obtain a more accurate approximation of the target object. The improved rendering method has been applied to several sets of input images. The information of these images sets is listed in Table 1.

Two experimental examples of them have been shown in Fig. 5 and Fig. 7. More rendering results can be found in Fig. 8. In each data set, the left column is one of the input reference images, while the right column shows several rendered new views. From these results we can find that the new views are highly photo-realistic. The bottom of the object and the concave shape are now correctly rebuilt, not only in geometry, but also in texture. In Set 1-2, the tags on the bottom of the object are

Input images	Rendered new views	Input images	Rendered new views

Figure 8. Experimental results on 4 image sets. Top-left: Set 1, Tea pot; Top-right: Set 2, Blue Cup; Bottom-left: Set 3, White Cup; Bottom-right: Set 4, Jade Cabbage. The rendering results produced by our improved method are highly photo-realistic. Note that the bottoms of the objects and the concave shapes are correctly rendered.

precisely rendered. In Set 3, there's no more distortion on the top of the cup; the pattern on the inner side is also well preserved. In Set 4, the transparent base of the jade cabbage brings more difficulty in reconstruction. However, the result is still satisfying.

More compares between our method and regular IBVH method are also given in Fig. 9. Obviously, the results of our method are significantly improved, and the distortions in both geometry and texture are corrected now.

| Regular IBVH without bottom images | Improved bottom rendering | Improved concave rendering |

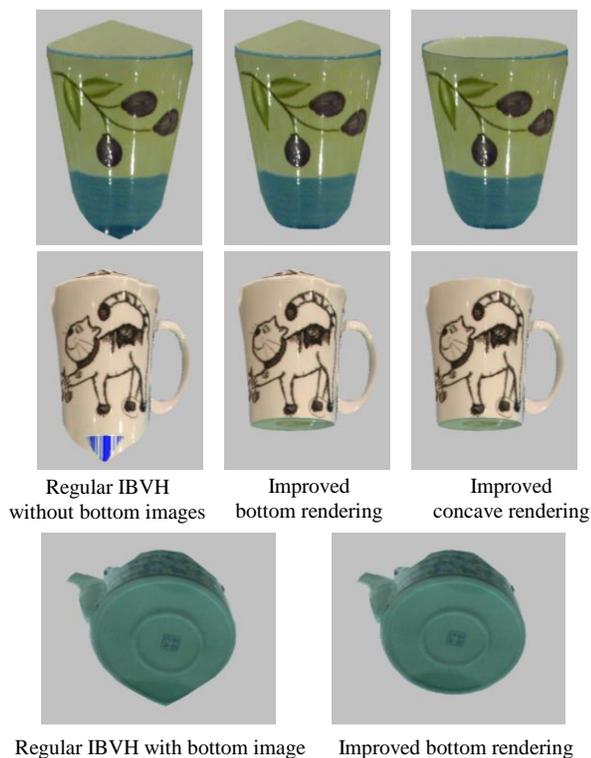| Regular IBVH with bottom image | Improved bottom rendering |

Fig. 9. Compare between regular IBVH and our method.

VII.   CONCLUSION

In this paper, we presented a simple and efficient method to solve the bottom and concave surface rendering problem in image-based visual hull. Our method requires no special equipments but only a planar glass and a planar mirror. With their help, the images of the bottom and concave region on the object could be easily acquired at the same time as other reference images, and can be directly used in the visual hull reconstruction after simple transformation. Therefore, our improved method can produce high-quality photorealistic results without distortion in bottom or concave parts, while the high efficiency of the original method is preserved.

There are still some limitations in our algorithm, and could be further improved in future work:

Currently, the silhouettes in reference images are approximated by line segments. Their precision will affect the final reconstruction result. Increasing the number of segments or using curves would lead to better results.

Our current concave rendering method works well with objects like cups. However, the accuracy must be further improved to render more complex objects. For example, for objects with multiple concaves, more virtual images should be generated for each concave. The calculation of their parameters may be more complex than single-concave cases.

REFERENCES

[1] J.M. Gluckman and S.K. Nayar, Planar Catadioptric Stereo: Geometry and Calibration, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, USA, PP. 1022-1028, 1999.

[2] K. Forbes, A. Voigt, and N. Bodika, Visual Hulls from Single Uncalibrated Snapshots Using Two Planar Mirrors, in *Proc. The 5th Annual Symposium of the Pattern Recognition Association of South Africa (PRASA 2004)*, Grabouw, pp. 29-34, 2004.

[3] K. Forbes, F. Nicolls, G. de Jager, and A. Voigt, Shape-from-Silhouette with Two Mirrors and an Uncalibrated Camera, in *Proc. the 9th European Conference on Computer Vision (ECCV)*, Graz, Austria, pp. 165-178, 2006.

[4] W. Matusik, C. Buehler and R. Raskar, Image-Based Visual Hulls, in *Proc. ACM SIGGRAPH 2000*, New Orleans, USA, pp. 369- 378, 2000.

[5] M. Li, Towards Real-Time Novel View Synthesis Using Visual Hulls, *Ph.D. thesis, Max-Planck-Institut für Informatik*, 2004.

[6] K.N. Kutulakos and S.M. Seitz, A Theory of Shape by Space Carving, *International Journal of Computer Vision, MARR PRIZE SPECIAL ISSUE*, vol. 38, no. 3, pp. 199-218, 2000.

[7] G. Slabaugh, B. Culbertson, T. Malzbender and R. Schafer, A Survey of Methods for Volumetric Scene Reconstruction from Photographs, in *Proc. International Workshop on Volume Graphics 2001*, New York, USA, pp. 81-100, 2001.

[8] G.K.M. Cheung, S. Baker, and T. Kanade. Visual Hull Alignment and Refinement Across Time: A 3D Reconstruction Algorithm Combining Shape-From-Silhouette with Stereo, in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2003*, Madison, USA, pp. II-375-382 , 2003.

[9] W.E. Lorensen and H.E. Cline, Marching Cubes: A High Resolution 3D Surface Construction Algorithm, in *Proc. ACM SIGGRAPH '87*, Anaheim, USA, pp. 163-169, 1987.

[10] Z.Y. Hu, C. Lei, and F.C. Wu, A Note on the Planar P4P Problem, *ACTA Automatica Sinica*, vol.27, no. 6, pp. 770-776, Nov., 2001.

Jie Feng is an assistant researcher in the Institute of Computer Science & Technology at Peking University. She earned her PH.D. from the Center for Information Science at Peking University in 2005.

She has been concentrating on computer vision, digital geometry processing and image-based modeling since her graduation. Her research interests include 3D mesh model compressing, multi-resolution modeling, image-based modeling and relighting.

Bencong Song was a master student in the Institute of Computer Science & Technology at Peking University, where his researches mainly focus on 3D mesh parameterization, image-based modeling and relighting. He had received his master degree in engineering in 2008.

Bingfeng Zhou is a Professor and Ph.D. candidate supervisor in the Institute of Computer Science & Technology at Peking University. He earned his Ph.D. in Peking University in 1993.

He has been working on image processing and computer graphics for many years. His research interests include half-toning technique, digital geometry processing, image-based modeling, and non-photo-realistic rendering etc. He has published several tens of papers on academic conferences and journals, including one in SIGGRAPH 2003.

Prof. Zhou is a member of ACM, China Society of Image and Graphics, and China Computer Federation.