7

Refinement of Visual Hulls for Human Performance Capture

Toshihiko Yamasaki and Kiyoharu Aizawa
The University of Tokyo
Japan

1. Introduction

Generation of dynamic three-dimensional (3D) mesh sequences of human performance using multiple cameras has been actively investigated in recent years (de Aguiar et al., 2008; Hisatomi et al., 2008; Kanade et al., 1997; Kim et al., 2007; Matsuyama et al., 2004; Nobuhara & Matsuyama, 2003; Snow et al., 2000; Starck & Hilton, 2007; Tomiyama et al., 2004; Toyoura et al., 2007; Tung et al., 2008; Vlasic et al., 2008). The topic is drawing a lot of attention because conventional 3D shape measurement tools, such as laser scanners, shape (structure)-from-motion (Huang & Netravali, 1994; Poelman & Kanade, 1997), shape-from-shading (Zhang et al., 1999), etc., are difficult to apply to dynamic scenes. On the other hand, depth cameras, such as time-of-flight (Foix et al., 2011) and structured light (Fofi et al., 2004) cameras can measure depth only from the viewpoint, and they do not measure the entire 3D shape of objects. There are many attractive applications of 3D human performance capture such as movies, education, computer aided design (CAD), heritage documentation, broadcasting, surveillance, gaming, etc.

Shape-from-silhouette (or volume intersection) (Laurentini, 1994) is a fundamental process in generating the convex hulls of the 3D objects. Because the shape-from-silhouette algorithm is directly affected by the foreground/background segmentation, a well-controlled monotone background is often employed (de Aguiar et al., 2008; Kim et al., 2007; Starck & Hilton, 2007; Tomiyama et al., 2004; Toyoura et al., 2007; Vlasic et al., 2008). However, proper segmentation has been a serious problem even in such studios. Therefore, a number of approaches have been proposed for refining the geometrical data of the objects in both the spatial and temporal domains.

This chapter reviews recent works on the refinement of visual hulls and describes our contribution featuring iterative refinement of foreground/background segmentation and visual hull generation.

The rest of this chapter is organized as follows. Section 2 reviews related works for the robust 3D model reconstruction. Section 3 describes our 3D studio and our proposed algorithm. Experimental results are presented in Section 4. Finally, concluding remarks are given in Section 5.

2. Related works

This section summarizes related works on 3D model refinement. The spatial and the temporal domain approaches are orthogonal and are independent of each other. They can be combined to generate more accurate 3D models, although this is out of the scope of this chapter.

2.1 Spatial domain approaches

Spatial domain approaches can be categorized into those that refine the foreground/background segmentation and those that refine the generated visual hulls by additional algorithms.

2.1.1 Refinement of silhouette extraction

One of the straightforward approaches is improving the foreground/background segmentation (Benezeth et al., 2008; McIvor, 2000; Piccardi, 2004) regardless of a 3D modeling context. However, none of the previous algorithms is perfect.

Toyoura (Toyoura et al., 2007) proposed a silhouette extraction algorithm using a random pattern background. By using small patches of a random color pattern, the probability of the foreground color coinciding with that of the background in all viewpoints becomes very small. Even when the color of the background is close to that of the foreground object in a certain view, the background color from a different view is far from that of the foreground object. Therefore, misclassification of the foreground as the background can be suppressed. This approach can reduce the loss of voxels, but on the other hand tends to yield a voxel surplus. In addition, a proper design of a random pattern background depending on the size of the studio is required.

Kim (Kim et al., 2007) introduced a reliability map of foreground/background segmentation. Foreground and background regions were modeled by the stochastic approach, which was named generalized Gaussian family (GGF), and confidence scores were assigned to each pixel. In the modeling process, rule-based error correction was employed to reduce voxel loss because of segmentation errors and occlusions. However, this approach also tends to yield superfluous voxels. In addition, the GGF model needs to be trained in each environment.

An object silhouette extraction method with error detection and correction using multiviewpoint images was proposed by Nobuhara (Nobuhara et al., 2007). In this approach, two constraints were introduced: "intersection," which assumes that the projection of the visual hull on every viewpoint was equal to the silhouette on each viewpoint; and "projection," which implies that projection of the visual hull should have an outline that matches with the apparent edges of the captured image on each viewpoint. This algorithm required several hundreds of iterations and took 0.5–3 days to process only a single frame. Therefore, it was not feasible for our purpose.

2.1.2 Refinement of generated visual hulls

Shape-from-silhouette (or volume intersection) generates a convex hull model and concave parts cannot be modeled properly. The key information to eliminate unnecessary voxels in concave parts is photo consistency.

In (Kutulakos & Seitz, 2000; Seitz & Dyer, 1999; Slabaugh et al., 2001), voxel colorization and photo consistency evaluation was done voxel-by-voxel. When the differences between the voxel color and the corresponding pixel values were above the threshold, the voxel was eliminated. In this approach, the problem was setting the proper threshold value.

Tomiyama (Tomiyama et al., 2004) and Starck (Starck & Hilton, 2007) employed stereo matching to calculate a more detailed shape of the object. The depth search range was restricted by the visual hull model with the assumption that the actual surface point should exist on or inside the visual hull according to the theory of space carving (Kutulakos & Seitz, 2000). By this constraint, the computational cost was reduced and at the same time, the depth estimation error due to mismatching was reduced. A similar idea can also be found in (Fua & Leclerc, 1995), but this work was meant for 2.5D (multiview + depth) model reconstruction.

The graph cuts algorithm was also employed after the shape-from-silhouette for refining the concave part of objects (Hisatomi et al., 2008; Liu et al., 2006; Tung et al., 2008). In (Hisatomi et al., 2008), the constraint term imposed by silhouette edges was introduced to preserve thin parts. Tung (Tung et al., 2008) combined both superresolution and dynamic 3D shape reconstruction problems into a unique Markov random field (MRF) energy formulation and optimized the cost function by graph cuts.

These approaches are used only for removing unnecessary voxels; the loss of voxels deriving from erroneous silhouette extraction cannot be recovered. Therefore, these algorithms should be applied after the shape-from-silhouette processing with perfect foreground/background segmentation to remove only surplus voxels.

2.1.3 Other approaches

An alternative approach to using shape-from-silhouette is using graph cuts in the 3D space (Snow et al., 2000). The difference from (Hisatomi et al., 2008; Liu et al., 2006; Tung et al., 2008) is that this approach does not use the volume intersection. In (Snow et al., 2000), the data term was the sum of the values attached to the voxels, where the value was based on the observed intensities of the pixels that intersect it, and the smoothness term was defined as the number of empty voxels adjacent to filled ones. However, the accuracy of the modeling was not discussed in (Snow et al., 2000). As pointed out in (Hisatomi et al., 2008), combining the shape-from-silhouette and the graph cuts algorithm yields better results for flat color and repetitive color pattern regions.

The probabilistic model (Broadhurst et al., 2001) calculates the photo-consistency energy of the two cases; i.e., whether the voxel exists or not. The probability of the existence of each voxel was calculated by Bayse's rule to choose which case is more likely. Similar stochastic approaches can also be found in (Bonet & Viola, 1999; Isidoro & Sclaroff, 2002).

2.2 Temporal domain approaches

In temporal domain approaches, 3D models are generated by deforming and refining the reference 3D models in different frames. Therefore, the manner of taking the correspondence between the feature points in neighboring frames (models) is important for extracting deformation and refinement parameters. The temporal domain refinement not only generates more accurate shapes of the 3D objects, but also keeps the geometry and the topology of the generated 3D models coherent throughout the frames (i.e., the number of vertices and their connectivity are consistent). This would also facilitate better quality texture mapping, compression, and motion tracking and analysis of the generated 3D model sequences.

Nobuhara (Nobuhara & Matsuyama, 2003) proposed a deformable mesh model taking into account five constraints, such as photo consistency, silhouette, smoothness, 3D motion flow, and inertia. First, intraframe deformation was conducted considering the first three

constraints (this part constitutes spatial refinement) and the 3D model in the previous frame was deformed to match the model in the present frame considering the last two constraints.
In (Vlasic et al., 2008), a skeleton model was used to track the motion of the object and the template model was deformed using linear blend skinning to meet the silhouette fitting constraint. The algorithm depended only on the silhouette and no color information was utilized.
A feature-based tracking in captured 2D images using scale-invariant feature transform (SIFT) features (Lowe, 2004) was proposed by de Aguiar (de Aguiar et al., 2008). The model was then deformed based on the extracted motion. Details were recovered by adjusting the vertices to the silhouette contours and by estimating the depth using multiview stereo. In this work, the initial model was generated using a laser scanner.
In (Luo et al., 2010), a modified annealed particle filtering was proposed to track the motion, and deformation and shape refinement were performed considering the silhouette of the human body.

2.3 Proposed work in this chapter

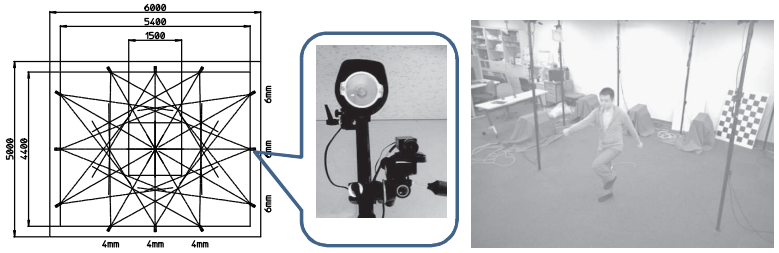
Most of the algorithms, for spatial refinement in particular, are designed only to eliminate unnecessary voxels, not to recover erroneously removed voxels (exceptions can be found, for example, in (Kim et al., 2007)). Therefore, the misclassification of the foreground object region as background in segmentation is a critical problem, not to mention that the excess number of voxels in the dilation process utilized for solving such a problem is difficult to remove even with the fancy algorithms listed above.
Therefore, we have developed a 3D model generation algorithm with smaller numbers of lost and surplus voxels (Yamasaki et al., 2009), which can be categorized as the spatial domain approach. This algorithm works well even without a monotone background. Our algorithm is based on the iterative feedback between the silhouette extraction and the 3D modeling; namely, the generated 3D models are rendered and used as a seed for the graph cuts algorithm (Boykov & Jolly, 2001; Rother et al., 2008) for better silhouette extraction. The improved silhouette images are used to reconstruct the 3D models. This iterative process is repeated until the geometrical shape of the 3D models converges. As a result, both the voxel loss and surplus can be suppressed drastically compared with conventional algorithms. The difference from (Kim et al., 2007; Toyoura et al., 2007) is that the generated 3D models are improved iteratively, not by a single-shot correction. In addition, the computational cost is not very large because the number of required iterations is quite small, as discussed in 4. Whereas (Nobuhara et al., 2007) updates the silhouette image one by one sequentially, which is therefore time consuming, the proposed method updated all the silhouette images in each iteration.

3. 3D model generation based on iterative feedback between silhouette extraction and geometry modeling

3.1 Studio setup

Our 3D modeling studio is illustrated in Fig. 1. The studio consisted of 12 sets of capturing units: a camera with 1360 $\times$ 1024 resolution and camera-link interface, light, and personal computer (Intel Core2 Duo 2.4 GHz, 4 GB memory, RAID 0 HDD operating at 3 GB/s) attached to a pole. All the cameras were synchronized by an external signal generator. The frame rate was up to 34 fps. The system was built in our laboratory room (Fig. 1(b)). No special background such as a blue sheet was utilized. Only the computers were covered with

cloths because they are shiny and affect the silhouette extraction. Camera calibration was done using Tsai's method (Tsai, 1987).
The system was easy to set up and portable. Disassembling and setting up the studio again can be achieved in a few hours. The size of the studio was about 6 m $\times$ 5 m but these dimensions are flexible, depending on the size of the object and the area required for the object to move around.



(a) Floor plan (b) A view from a certain camera.

Fig. 1. Studio setup.

3.2 Flow of the algorithm

The flowchart of our 3D modeling algorithm is shown in Fig. 2. In the initial step, we conducted conventional silhouette extraction and 3D modeling. Then, we proceeded to the iterative processing between silhouette refinement using the rendered images and the 3D model reconstruction with error compensation. When the generated 3D model converged and was not very different from that of the previous step, the iteration was terminated and the final 3D mesh was obtained.
For higher-quality modeling, especially for reconstructing concave parts, sophisticated model refinement algorithms are required after the shape-from-silhouette, such as deformable mesh (Matsuyama et al., 2004), stereo matching, (Starck & Hilton, 2007; Tomiyama et al., 2004) and graph cuts in the 3D space (Hisatomi et al., 2008; Tung et al., 2008). However, such a model refinement process is out of the scope of this chapter. Our target was to generate shape-from-silhouette-based 3D mesh models with loss of fewer voxels while suppressing surplus of voxels for such refinement algorithms to work better.

3.3 Shape-from-silhouette with error compensation

The shape-from-silhouette is a 3D modeling algorithm that works by taking the intersections of visual cones of all the cameras surrounding the object, as shown in Fig. 3. In other words, if a voxel is seen from all the cameras, the voxel remains. Otherwise, the voxel is removed. In this manner, the visual hull of the 3D object is estimated. Then, various refinement algorithms are applied for modeling convex parts or smoothing the model. One of the most significant disadvantages of this approach is that when a voxel is invisible from even a single camera due to erroneous silhouette extraction, it is eliminated. On the other hand, the probability of a nonobject voxel to be visible to all the cameras is quite low because the voxel can be labeled as a nonobject by other cameras. Such loss of voxels degrades the visual quality of the model. An example is shown in Fig. 4. In this case, the left arm in camera #10 was missing because of erroneous silhouette extraction and the error significantly affected the generated 3D model. Note that the error in Fig. 4 was an actual result, not a simulation. The refinement algorithms

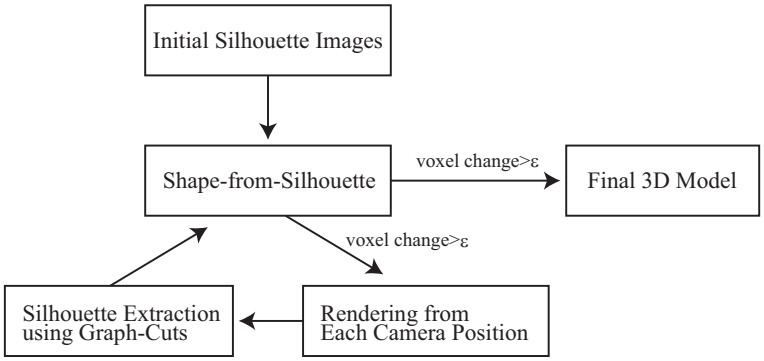


Fig. 2. Flowchart of the proposed algorithm.

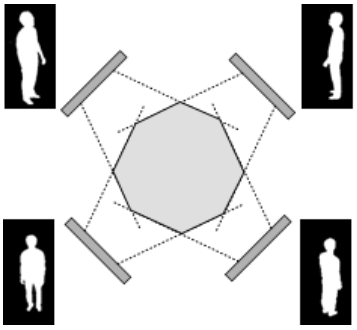
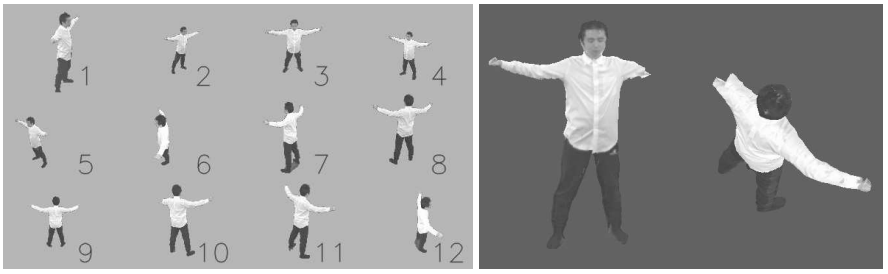


Fig. 3. The shape-from-silhouette algorithm.



(a) Error in silhouette extraction only in camera #10.

(b) Generated 3D model in which the left arm was not reconstructed properly.

Fig. 4. An example of voxel loss.

(Hisatomi et al., 2008; Matsuyama et al., 2004; Tomiyama et al., 2004; Tung et al., 2008) cannot recover such loss of voxels, because they are designed to eliminate unnecessary voxels, not to add necessary ones. Therefore, two kinds of error (loss) compensation algorithms were introduced in this chapter.

One such algorithm is the voting-based modeling method. Here, we assumed the number of cameras in the studio as n, and m was an integer ranging from 1 to $n-1$. If the voxel is visible from $n-m$ cameras, the voxel survives. Typically, m is set as $1-2$ because the

probability of a voxel that belongs to an object to be invisible from two or more cameras in the view range is quite low. Therefore, voxels that were deleted due to the erroneous segmentation can be recovered. If we increase m, the generated 3D model would expand more than necessary; namely, the voxels that should be deleted remain in the visual hulls. If the error in silhouette extraction occurs in many camera views, we should reconsider the silhouette extraction algorithm itself. In this approach, one 3D model is generated for a single frame, independent of the value m.

The other approach is modeling with the other $(n-1)$ camera views. When generating the foreground/background seeds for the i-th camera view, the $(n-1)$ camera views, excluding the i-th camera view, are used for the modeling, and the generated 3D model is rendered from the i-th camera position only for improving the i-th silhouette. Therefore, we need to conduct the 3D modeling for all the n camera views. This approach implicitly assumes that the segmentation error does not occur in multiple views at the same time, which is reasonable in most cases. It is important to note here that such an error can occur in multiple parts as long as the condition mentioned above holds. The restriction here is that a voxel is misclassified as a nonobject region by not more than a single camera. Modeling with the other $(n-2)$ camera or fewer views is not reasonable because the number of models to generate becomes quite large: $n \times (n-1)$ for the case of $n-2$.

In the iteration process, 3D model reconstruction is conducted multiple times. In particular, the cost for modeling with the $(n-1)$ camera views approach becomes quite expensive as the number of cameras increases. To save computational cost, the 3D modeling in the iteration can be done with rough spatial resolution and only the final modeling should be carried out with finer spatial resolution. Another option is to iterate the refinement process only once because the modeling accuracy by a single iteration becomes sufficiently high, as demonstrated in Section 4.

3.4 Silhouette extraction and updating

In the initial silhouette extraction, conventional background subtraction with the graph cuts was used. The background and foreground regions with high confidence were generated as follows.

$$\begin{aligned} \text{if } |Y(x,y) - Y_{BG}(x,y)| > Th1, \text{ then } (x,y) \text{ is foreground} \\ \text{else if } |Y(x,y) - Y_{BG}(x,y)| < Th2, \text{ then } (x,y) \text{ is background} \\ \text{else unknown} \end{aligned}$$

Here, $Y(x,y)$ is the chroma value of the pixel at (x,y) and $Y_{BG}(x,y)$ is that of the background model. $Th1$ and $Th2$ are predefined threshold values where $Th1 > Th2$ to extract background and foreground regions with high confidence. When $|Y(x,y) - Y_{BG}(x,y)|$ is between $Th1$ and $Th2$, the pixel is left as unknown. Then, the background/foreground maps are fed to the graph cuts algorithms as seeds. The silhouette extraction results are shown in Fig. 4(a).

In the iteration process, we assume that the erroneous loss of voxels is compensated by either of the ways described in 3.2. The silhouette refinement for each camera view was conducted using three images: the original captured image (Fig. 5(a)), the silhouette image in the previous step (Fig. 5(b)), and the rendered 3D image from the camera position (Fig. 5(c)). The background seed was generated by the logical AND operation between the background regions in the previous silhouette image (Fig. 5(b)) and the rendered image (Fig. 5(c)). A similar color region (Fig. 5(d)) between the original captured image (Fig. 5(a)) and the rendered image (Fig. 5(c)) and the eroded silhouette image in the previous step (Fig. 5(e)) were

logically summed to form a foreground seed. As a result, the seeds for the background and the foreground for the graph cuts in the next step were generated, as demonstrated in Fig. 5(f). In the figure, the gray, black, and white regions represent the background, foreground, and unknown regions, respectively. The updated silhouette is shown in Fig. 5(g). This procedure was applied to each camera view independently. The updated silhouette images were again utilized for the 3D modeling. An example of the updated 3D model after a single feedback loop is shown in Fig. 5(h).

(a) Original captured image.

(b) Initial silhouette by background subtraction and graph cuts.

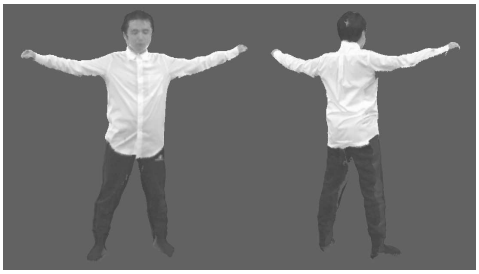
(c) The generated 3D model with the error compensation algorithm described in 3.2.

(d) Close-color map between (a) and (c).

(e) Eroded silhouette using (b).

(f) Updated seeds for graph cuts.

(g) Updated silhouette.



(h) Updated 3D model.

Fig. 5. Silhouette updating using the rendered 3D model.

4. Experiments

4.1 Experimental setup

The experiments were conducted using the 3D studio with 12 cameras, as described in Section 3.1. Consecutive $5-10$ frames of video (12 cameras $\times$ $5-10$ frames = $60-120$ images) were recorded for five people in different clothes and poses. The ground-truth data of the

silhouettes were generated by hand. Then, ground-truth 3D model sequences were generated by the shape-from-silhouette algorithm. Our shape-from-silhouette program was based on (Tomiyama et al., 2004) (courtesy of Tomiyama and colleagues). The stereo matching in (Tomiyama et al., 2004) was disabled in the experiments to investigate the effect of the iterative silhouette updating only. The accuracy of the model was calculated by comparing the voxels. The voxels in the generated model that did not exist in the ground-truth model were regarded as surplus voxels. On the other hand, voxels in the ground truth that were not observed in the generated model were regarded as lost voxels.

4.2 Evaluation of the five different models

Fig. 6 shows 3D models using only the initial silhouettes, those using the voting-based modeling method, and ground-truth models. In model A in Fig. 6(a), for instance, it is observed that the lost voxels at the back of the head and the missing right hand were compensated correctly. On the other hand, there were still some lost voxels at the right leg in model E. In this case, the color of the trousers was very close to that of the carpet and the assumption that "the probability of the voxel that belongs to the object to be invisible from two or more cameras is quite low" made in Section 3.2 did not hold any more. If the cameras were looking down on the objects, the same region of the floor was observed by multiple cameras. Therefore, the color of the floor should be different from that of the trousers of the performer and vice versa. Otherwise a random pattern can be used only on the floor, as in (Toyoura et al., 2007).

The average errors over the frames for the best (model A) and the worst (model B) cases are summarized in Tables 1 and 2, except for model E that does not hold the assumption. The modeling performance by Toyoura et al. (Toyoura et al., 2007) is also shown in Table 3 for comparison. Note that the experimental setup and the target models were very different from (Toyoura et al., 2007). In Toyoura's approach, the loss of voxels is reduced but at the same time the surplus of voxels is increased and the generated models are "fat" compared with the ground-truth model. On the other hand, in our approach, both the loss and surplus of voxels were suppressed effectively.

	Loss	Surplus	Total Error
Modeling using the initial silhouettes	2.1%	9.4%	11%
Modeling with the other $(n-1)$ camera views	0.90%	0.99%	1.9%
Voting-based with iteration ($n-1$ cameras)	0.73%	1.2%	1.9%

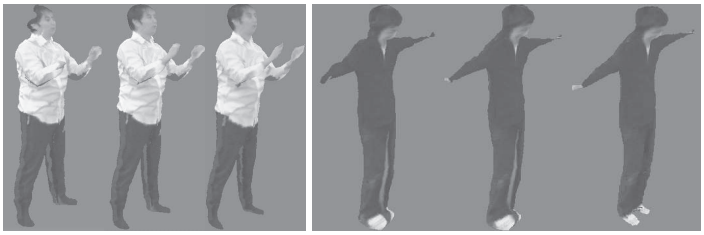
Table 1. Averaged modeling accuracy over the 10 frames of A (the best case among A–D).

	Loss	Surplus	Total Error
Modeling using the initial silhouettes	4.1%	24%	28%
Modeling with the other $(n-1)$ camera views	0.68%	14 %	15%
Voting-based with iteration ($n-1$ cameras)	1.4 %	12 %	14 %

Table 2. Averaged modeling accuracy over the 5 frames of B (the worst case among A–D).

	Loss	Surplus	Total Error
Modeling using the initial silhouettes	58%	3.1 %	60 %
(Toyoura et al., 2007)	2.7%	11%	14%

Table 3. Results in (Toyoura et al., 2007).



(a) Model A. (b) Model B.

(c) Model C. (d) Model D.

(e) Model E.

Fig. 6. Example of the generated models: (left) models using only initial silhouettes, (middle) refined models using the voting-based method, (right) ground-truth models

4.3 Detailed evaluation over the frames

In this section, we further investigate the performance of the proposed algorithm using model A.

The mean errors over the frames are summarized in Table 4. The modeling errors using six different approaches are compared: modeling using the initial silhouettes, modeling with the other $(n-1)$ camera views, voting-based modeling without iteration (using $n-1$ cameras), voting-based modeling without iteration (using $n-2$ cameras), voting-based modeling with iteration (using $n-1$ cameras), and voting-based modeling with iteration (using $n-2$ cameras). The models without iteration were intermediate models used for silhouette refinement and although they were not the final results, they are listed here for comparison. In modeling using the other camera views, n models were generated. Therefore, the modeling errors of the intermediate models (i.e., modeling without iteration) were difficult to analyze and are not shown in the table. The proposed algorithms yielded a good performance, both in terms of loss and surplus of voxels. The total error was less than 2% for both voting-based modeling by $(n-1)$ cameras and for other $(n-1)$ camera views. When the voting-based

modeling method without iteration using $n-1$ camera views was employed, the loss of voxels was quite small. However, generated models contained many surplus voxels, resulting in a larger total error than in the modeling using the initial silhouettes. The region where a major loss of voxels occurred (0.18%) was the region that did not hold the assumption that the probability of a voxel that belongs to the object to be invisible from two or more cameras was quite low. In other words, our assumption was valid for 99.8% of the region.

	Loss	Surplus	Total Error
Modeling using the initial silhouettes	2.1%	9.4%	11%
Modeling with the other ($n-1$) camera views	0.90%	0.99%	1.9%
Voting-based w/o iteration ($n-1$ cameras)	0.18%	26%	26%
Voting-based w/o iteration ($n-2$ cameras)	0.007%	47%	47%
Voting-based with iteration ($n-1$ cameras)	0.73%	1.2%	1.9%
Voting-based with iteration ($n-2$ cameras)	0.64%	2.0%	2.7%

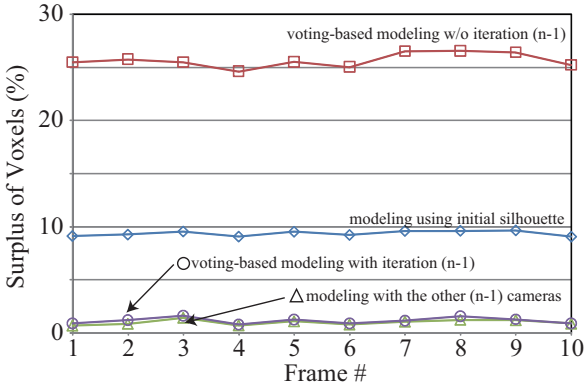
Table 4. Averaged modeling accuracy for model A over the 10 frames.

It can be observed that the proposed approaches can generate better 3D models than a simple volume intersection method in terms of both loss and surplus of voxels. Namely, the iterative processing between silhouette extraction and 3D modeling can reduce voxel loss while suppressing voxel surplus. Among the lost voxels in the initial model (2.1%), 90% of them (1.9% of the whole model) were invisible only from a single camera and the loss was reduced to 0.73% in the voting-based method using $n-1$ cameras and to 0.90% with the other camera views method. In addition, we can see that modeling with the voting-based method was good at reducing voxel loss and modeling with the other camera views method performed well in reducing voxel surplus.

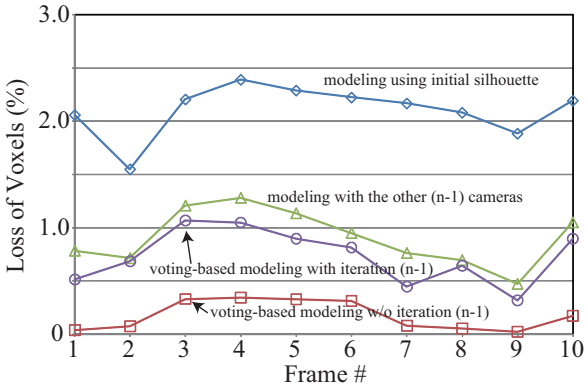
The modeling errors with the looser assumption that the probability of the voxel belonging to the object to be invisible from three (not two) or more cameras is low is also shown in Table 4 (see voting-based methods using $n-2$ cameras with/without iteration). In the voting-based method without iteration, the loss of voxels was as few as 0.007%, almost negligible. On the other hand, the surplus of voxels increased up to 47%. When the voting-based method with iteration using $n-2$ cameras was employed, voxel loss was at the minimum among the proposed methods. However, the surplus of voxels tended to be somewhat more than in the other approaches and was almost the same as the initial model in some frames (not shown). The optimal number of cameras to use in the iteration should be decided considering the number of cameras, the shape refinement process in the following stage, the required error rate, etc.

Fig. 7 shows the modeling accuracy for model A. It is demonstrated that the error was almost constant throughout the frames independent of the poses of the performer.

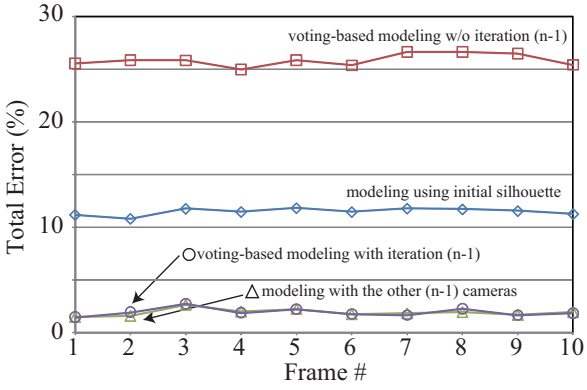
In all the frames, the shape of the model converged at the second iteration (the difference between the models in the first and second iterations was smaller than ϵ). To investigate how the errors change in the iteration process, the errors for model A averaged over the 10 frames as a function of the number of the iteration is shown in Fig. 8. In this experiment, the termination decision was disabled. Iteration zero stands for the initial model. Regardless of whether the algorithm was the voting-based method or modeling with the other cameras, the generated 3D model converged quickly and the errors did not improve very much after the first iteration. Therefore, modeling with only a single feedback is enough in most cases. The mean processing time for the voting-based method using $n-1$ cameras was 35 s and that



(a) Surplus of voxels.

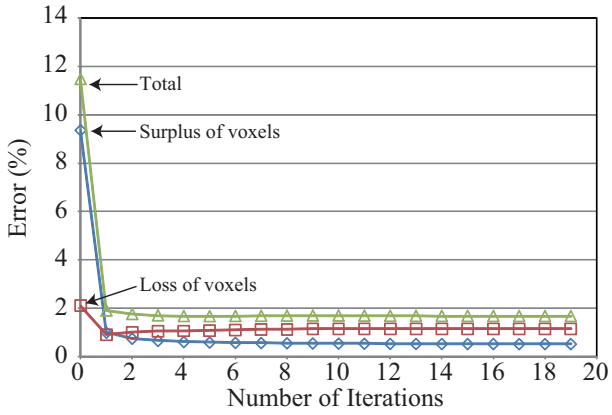


(b) Loss of voxels.

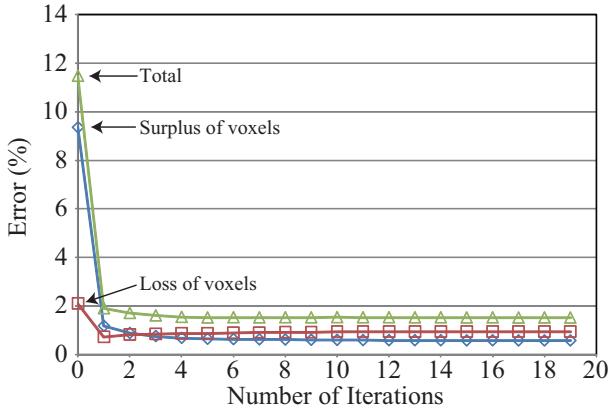


(c) Total error.

Fig. 7. Modeling accuracy for model A.



(a) Modeling with the other $(n-1)$ cameras.



(b) Voting-based modeling by $(n-1)$ cameras.

Fig. 8. Model refinement effects as a function of the number of iterations.

for modeling with the other cameras was 45 s using the Intel Core2 Duo 2.4 GHz and 2.5 GB memory. On the other hand, the simple volume intersection took 2.5 s.

5. Conclusions

In this chapter, we have reviewed visual hull refinement algorithms and presented an iterative refinement algorithm. By the cross-feedback between the 3D model reconstruction with the updated silhouette and the silhouette extraction using the rendered image, the loss and surplus of voxels can be kept very small. We have also proposed two shape-from-silhouette algorithms with error compensation to recover missed segmentation of the background/foreground. Experimental results demonstrated that the loss of voxels was reduced from 2.1% to 0.73–0.90% and the surplus of voxels was reduced from 9.4% to 0.99–1.2%, respectively. Achieving as few a loss of voxels as possible is important because the

surplus of voxels can be eliminated by further postprocessing, whereas it is very difficult to recover the erroneously eliminated voxels.

6. Acknowledgments

We would like to thank Mr. Yamada for his contribution and nac Image Technology, Inc. for the studio design. This work is supported by the Microsoft Institute for Japanese Academic Research Collaboration (IJARC). We would like to thank Dr. Tomiyama and colleagues for providing us with their 3D modeling source code.

7. References

Benezeth, Y., Jodoin, P., Emile, B., Laurent, H. & Rosen-berger, C. (2008). Review and evaluation of commonly-implemented background subtraction algorithms, *Proceedings of IEEE 19th International Conference on Pattern Recognition (ICPR 2008)*, pp. 1–4.

Bonet, J. D. & Viola, P. (1999). Roxels: responsibility weighted 3d volume reconstruction, *Proceedings of Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, Vol. Vol. 1, pp. 418–425.

Boykov, Y. & Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images, *Proceedings of IEEE International Conference on Computer Vision (ICCV 2001)*, Vol. Vol. I, pp. 105–112.

Broadhurst, A., Drummond, T. & Cipolla, R. (2001). A probabilistic framework for space carving, *Proceedings of Eighth IEEE International Conference on Computer Vision (ICCV 2001)*, Vol. Vol. 1, pp. 388–393.

de Aguiar, E., Stoll, C., Theobalt, C., Ahmed, N., Seidel, H. & Thrum, S. (2008). Performance capture from sparse multi-view stereo, *ACM Transactions on Graphics (ACM SIGGRAPH2008)*.

Fofi, D., Sliwa, T. & Voisin, Y. (2004). A comparative survey on invisible structured light, *Proceedings of SPIE 5303*, pp. 90–98.

Foix, S., Alenya, G. & Torras, C. (2011). Lock-in time-of-flight (tof) cameras: A survey, *IEEE Sensors Journal* Vol. 11(No. x): xxx–xxx.

Fua, P. & Leclerc, Y. G. (1995). Object-centered surface reconstruction: Combining multi-image stereo and shading, *International Journal of Computer Vision* Vol. 16(No. 1): 35–56.

Hisatomi, K., Tomiyama, K., Katayama, M. & Iwadate, Y. (2008). 3d reconstruction using graph cut with view-dependent polygon texture blending, *5th European Conference on Visual Media Production (CVMP 2008)*, p. 18.

Huang, T. & Netravali, A. (1994). Motion and structure from feature correspondences: a review, *Proceedings of the IEEE* Vol. 82(No. 2): 252–268.

Isidoro, J. & Sclaroff, S. (2002). Stochastic mesh-based multiview reconstruction, *Proceedings of First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT 2002)*, pp. 568–577.

Kanade, T., Rander, P. & Narayanan, P. (1997). Virtualized reality: constructing virtual worlds from real scenes, *IEEE Multimedia* Vol. 4(No. 1): 34–47.

Kim, H., Sakamoto, R., Kitahara, I., Orman, N., Toriyama, T. & Kogure, K. (2007). Compensated visual hull for defective segmentation and occlusion, *Proceedings of the 17th International Conference on Artificial Reality and Telexistence (ICAT 2007)*, pp. 210–217.

Kutulakos, K. N. & Seitz, S. M. (2000). A theory of shape by space carving, *International Journal of Computer Vision (IJCV)* Vol. 38(No. 3): 199–218.

Laurentini, A. (1994). The visual hull concept for silhouette-based image understanding, *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 16(No. 2): 150–162.

Liu, X., Yao, H., Chen, X. & Gao, W. (2006). Visual hull embossment by graph cuts, *Proceedings of 2006 IEEE International Conference on Image Processing (ICIP 2006)*, pp. 2205–2208.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision (IJCV)* 60(2).

Luo, W., Yamasaki, T. & Aizawa, K. (2010). Articulated human motion capture from segmented visual hulls and surface re-construction, *Proceedings of 2010 APSIPA Annual Summit and Conference (APSIPA ASC 2010)*, pp. 109–116.

Matsuyama, T., Wu, X., Takai, T. & Wada, T. (2004). Real-time dynamic 3-d object shape reconstruction and high-fidelity texture mapping for 3-d video, *IEEE Transactions on Circuit And System For Video Technology* Vol. 14(No. 3): 357–369.

McIvor, A. (2000). Background subtraction techniques, *Proc. Image Video Comput.*, pp. 147–153.

Nobuhara, S. & Matsuyama, T. (2003). Dynamic 3d shape from multi-viewpoint images using deformable mesh model, *Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis (ISPA 2003)*, Vol. Vol. 1, pp. 192–197.

Nobuhara, S., Tsuda, Y., Matsuyama, T. & Ohama, I. (2007). Multi-viewpoint silhouette extraction with 3d context-aware error detection, correction, and shadow suppression, *Proceedings of 4th European Conference on Visual Media Production (CVMP2007)*, pp. 1–9.

Piccardi, M. (2004). Background subtraction techniques: a review, *Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics*, Vol. Vol. 4, pp. 3099–3104.

Poelman, C. & Kanade, T. (1997). A paraperspective factorization method for shape and motion recovery, *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 19(No. 3): 206–218.

Rother, C., Kolmogorv, V. & Blake, A. (2008). "grabcut":interactive foreground extraction using iterated graph cuts, *ACM Trans. Graphics (SIGGRAPH 2004)* Vol. 23(No. 3): 309–314.

Seitz, S. & Dyer, C. (1999). Photorealistic scene reconstruction by voxel coloring, *International Journal of Computer Vision (IJCV)* 25(1).

Slabaugh, G. G., Culbertson, W. B., Malzbender, T. & Schafer, R. W. (2001). A survey of methods for volumetric scene reconstruction from photographs, *Proceedings of International Workshop on Volume Graphics 2001*.

Snow, D., Viola, P. & Zabih, R. (2000). Exact voxel occupancy with graph cuts, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, Vol. 1, pp. 345–352.

Starck, J. & Hilton, A. (2007). Surface capture for performance-based animation, *IEEE Computer Graphics and Applications* Vol. 27(No. 3): 21–31.

Tomiyama, K., Orihara, Y., Katayama, M. & Iwadate, Y. (2004). Algorithm for dynamic 3d object generation from multiviewpoint images, *Proceedings of SPIE*, pp. 153–161.

Toyoura, M., Iiyama, M., Kakusho, K. & Minoh, M. (2007). Silhouette extraction with random pattern backgrounds for the volume intersection method, *Proceedings of the 6th International Conference on 3-D Digital Imaging and Modeling (3DIM 2007)*, pp. 225–232.

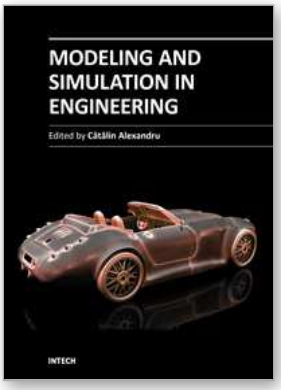
Tsai, R. Y. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses, *IEEE Journal of Robotics and Automation* Vol. 3(No. 4): 323–344.

Tung, T., Nobuhara, S. & Matsuyama, T. (2008). Simultaneous super-resolution and 3d video using graph-cuts, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008)*, pp. 1–8.

Vlasic, D., Baran, I. & Matusik, W. (2008). Articulated mesh animation from multi-view silhouettes, *ACM Transactions on Graphics (ACM SIGGRAPH2008)*.

Yamasaki, T., Yamada, K. & Aizawa, K. (2009). Time-varying mesh generation based on iterative feedback between silhouette extraction and geometry modeling, *Proceedings of 2009 APSIPA Annual Summit and Conference (APSIPA ASC 2009)*, pp. 502–508.

Zhang, R., Tsai, P.-S., Cryer, J. & Shah, M. (1999). Shape-from-shading: a survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 21(No. 8): 690–706.



Modeling and Simulation in Engineering

Edited by Prof. Catalin Alexandru

ISBN 978-953-51-0012-6
Hard cover, 298 pages
Publisher InTech
Published online 07, March, 2012
Published in print edition March, 2012

This book provides an open platform to establish and share knowledge developed by scholars, scientists, and engineers from all over the world, about various applications of the modeling and simulation in the design process of products, in various engineering fields. The book consists of 12 chapters arranged in two sections (3D Modeling and Virtual Prototyping), reflecting the multidimensionality of applications related to modeling and simulation. Some of the most recent modeling and simulation techniques, as well as some of the most accurate and sophisticated software in treating complex systems, are applied. All the original contributions in this book are jointed by the basic principle of a successful modeling and simulation process: as complex as necessary, and as simple as possible. The idea is to manipulate the simplifying assumptions in a way that reduces the complexity of the model (in order to make a real-time simulation), but without altering the precision of the results.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821