Combining Stereo and Visual Hull Information for On-line Reconstruction and Rendering of Dynamic Scenes

Ming Li, Hartmut Schirmacher, Marcus Magnor and Hans-Peter Seidel
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, D-66123, Saarbrücken, Germany
{ming,htschirm,magnor,hpseidel}@mpi-sb.mpg.de

***Abstract*— In this paper, we present a novel system which combines depth-from-stereo and visual hull reconstruction for acquiring dynamic real-world scenes at interactive rates. First, we use the silhouettes from multiple views to construct a polyhedral visual hull as an initial estimate of the object in the scene. The visual hull is then used to limit the disparity range during depth-from-stereo computation. The restricted search range improves both speed and quality of the stereo reconstruction. In return, stereo information can compensate for some of the inherent drawbacks of the visual hull method, such as inability to reconstruct surface details and concave regions. Our system achieves a reconstruction frame rate of 4 fps.**

I. Introduction

In recent years, the acquisition of dynamic scenes using multiple cameras has found exciting applications in the field of telecommunication, human-computer interaction and entertainment. However, a number of technical problems still need to be overcome, such as geometry reconstruction, photometry and lighting estimation, motion recovery, etc. Among these problems, the reconstruction of 3D geometry information is one key issue. Once the 3D scene structure is recovered, we can examine the scene from arbitrary viewpoints, and the recovery of photometric properties of the scene becomes possible. Also, scene geometry allows editing and mixing real and virtual environments.

There are many ways to reconstruct 3D information from images. Correlation-based "depth from stereo" [17] is one classical method. Fuchs et al. [8] and Kanade et al. [10] apply this method to reconstruct the real-world scene from a large amount of fixed cameras in off-line systems. Today, thanks to higher CPU speed and great progress of digital video camera technology, some commercial products [15] as well as academic research [5] prove that stereo reconstruction can be carried out by general-purpose computers in real time. An alternative approach to 3D reconstruction is shape-from-silhouette, which recover the scene as a visual hull [11]. Roughly speaking, the visual hull is a conservative shell that envelopes the true geometry of the object. Moezzi et al. [14] construct the visual hull using voxels in an off-line processing system. Cheung et al. [4] show that the voxel method can achieve interactive reconstruction results. The polyhedral visual hull system developed by Matusik et al. [12] also runs at interactive rate.

Yet both methods have their inherent drawbacks. The correlation-based stereo method is unstable when handling textureless surfaces and occluded regions. The visual hull cannot recover concave regions no matter how many images are used. Also it needs a large number of different views for recovering subtle details. However, the two methods are quite complementary in nature. As Simon Baker et al. point out [1], the Visual Hull method makes use of rays that are tangent to the surfaces of the object, while the stereo method primarily matches rays that are radiating from the object surface area. Therefore, these two methods can be combined to overcome their drawbacks and improve the reconstruction quality.

In this paper, we propose a combined method which follows the direction of Vedula's work [18]. A polyhedral visual hull is first constructed using silhouette information. This visual hull serves as an initial geometry estimate to limit the search range of the following stereo algorithm. While the visual hull improves both the quality and the speed of the stereo reconstruction, depth-from-stereo can recover more geometry details as well as concave regions of the object. Our method differs from Vedula's work in that we extract silhouette information from original images in the earlier processing stage and use a polyhedral visual hull representation for a direct and faster reconstruction of an initial geometric estimate. As a result, our system performs at interactive reconstruction and rendering rates. As far as we know, this is the first effort to combine the two classic 3D reconstruction methods in real-time.

The remainder of this paper is organized as follows. Sect. 2 gives an overview of our system. We explain our reconstruction algorithm in Sect. 3. The visualization of the reconstructed result is described in Sect. 4. After implementation details are described in Sect. 5, we conclude by presenting some ideas for future research.

II. System overview

Our system consists of several cameras observing the scene from different viewing directions. These cameras are grouped pairwise and arranged along an arc. Each camera pair is connected to one client computer. These computers communicate with the server via the standard TCP/IP network. All cameras are calibrated in advance and image acquisition is synchronized at run time. Fig 1 shows the acquisition setup.

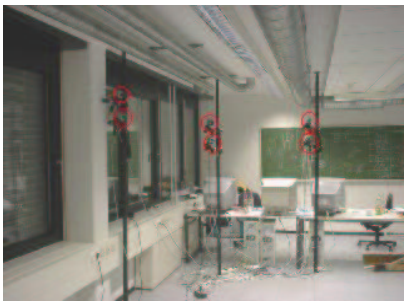


Fig. 1. Acquisition setup: 3 camera pairs are arranged along an arc.

The system initialization includes recording a background image for each camera and sending calibration information from each client to the server. After the initialization, the system enters the *processing cycle*, which is defined as the time of processing one synchronized image set collected by all cameras. According to the direction of the network transfer, we divide one cycle into three stages, illustrated in Fig. 2.

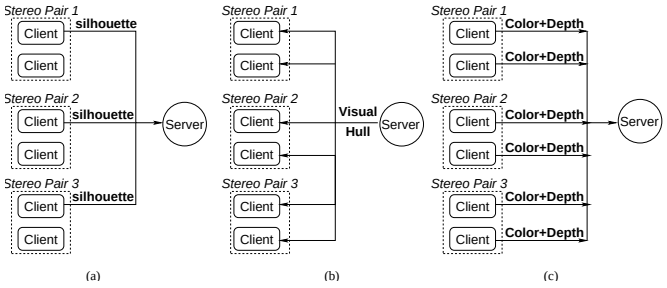


Fig. 2. Processing stages of the combined visual hull and depth-from-stereo algorithm. (a) Silhouette extraction (b) Polyhedral visual hull computation (c) stereo computation.

In stage 1, the object's silhouette is estimated. First, the stereo image pairs are individually rectified to align along scanlines [7]. For each image, the moving foreground object is segmented out from the previously acquired background. Since stereo cameras are very close to each other, using both silhouettes does not improve visual hull reconstruction significantly. Therefore, as seen in Fig. 2(a), we only extract the silhouette for one camera of the stereo pair and transfer it to the server.

In stage 2, when all silhouette information is available, the server computes a polyhedral visual hull using general 3D intersection and then broadcasts the polyhedral model back to all clients.

In the last stage, all clients use the visual hull to guide the depth computation. Note that since we already have the silhouette information, the stereo computation only needs to be performed on the foreground object mask instead of the whole image. The depth maps, together with the color images, are then sent back to the server for rendering.

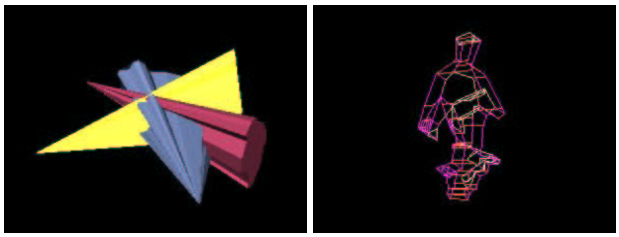
The system architecture has been designed to distribute the computational load between the server and the clients. The time needed for image acquisition, rectification, silhouette extraction and stereo reconstruction is independent of the number of stereo pairs. This provides good scalability and allows us to achieve interactivity.

III. Reconstruction of dynamic events

A. Visual hull reconstruction

There are two different methods for the visual hull reconstruction: volumetric [4], [14] and polyhedral [12]. Since volumetric reconstruction requires intensive memory, provides limited recovery precision and is time-consuming during rendering, we choose the polyhedral method which uses a more elegant polyhedral representation, requires less memory, and is suitable for fast direct rendering.

In our system *image differencing* [3] is used for segmenting the moving foreground object. Then we eliminate the moving shadow region by using color information as in [4] and apply morphological operators to fill small holes in the foreground object mask. Finally, the contour can be retrieved from the silhouette information as a 2D polygon. This polygon is sent back to the server and extruded to form a cone-like volume using the camera calibration information. We use the BREP library [2] to carry out the 3D intersection of several cone-like volumes. Fig. 3 illustrates the visual hull obtained from three cone-like volumes.



(a) (b)

Fig. 3. Visual hull reconstruction. (a) Three cone-like volumes extruded from different views. (b) The intersection result — a polyhedral visual hull.

B. Basic stereo algorithm

The stereo algorithm can be decomposed into three steps. (1) Rectification, (2) matching, and (3) depth recovery. Rectification is applied to the stereo images to align the epipolar lines to make pixel traversal and matching faster. Then we try to match the correspondences by comparing small neighborhoods around

each pixel. There exist several criteria to evaluate the difference between pixel neighborhoods, such as *normalized cross-correlation* (NCC), *sum of squared difference* (SSD), *sum of absolute difference* (SAD), etc. Among them, SAD is the fastest method and proves to be able to yield satisfactory results. We have further optimized the SAD computation and made it independent of the window size by exploiting the coherence of neighboring pixels [6].

The disparity is defined as the coordinate difference between corresponding pixels along image scanlines. Once the correspondence between pixels is established, we can generate a disparity map, from which the depth map can be obtained in a straightforward way by using camera information. To improve the quality of the depth map, we interpolate neighboring SAD scores to obtain sub-pixel accuracy [9] for the disparity. We also compute the disparity map for both stereo images and then check the left-right consistency to remove false matches.

C. Combination of visual hull and stereo

One of the key factors influencing the performance of stereo computation is the disparity range. For stereo matching, typically, one fixed depth range is specified for all video frames. This range must be large enough to accommodate all depths in the dynamic scene. A large depth range corresponds to a large disparity search range for the stereo algorithm, which means more correspondences must be examined. As a result, the large disparity range will not only slow down overall performance but also increase the possibility to find false matches, which lead to false depth values. Therefore, our approach is based on restricting the disparity search range for the stereo matching by using the visual hull information.

1) global range constraint: We can get global a disparity range constraint for all pixels of one video frame in the following way. First, a bounding box for the visual hull is computed at the server and transferred to each client. We transform the vertices of the box from the world coordinate system to the camera coordinate system. Then the minimum and maximum depth values of these transformed vertices can be converted to a disparity range. Recalling that the visual hull is a conservative estimate of the actual object, therefore the disparity range computed from the bounding box can be applied to restrict the correspondence search. This computation is carried out for every frame. The disparity range varies from frame to frame and is tighter than one fixed range.

The bounding box computation can be done very quickly. It also removes the need of transferring the whole visual hull information to the client. Despite its speed, the dynamic range constraint turns out to be only a rough estimate since it is global with respect to all the pixels of the foreground object.

2) Per-pixel range constraint: So far, we haven't taken full advantage of the visual hull information. Actually, if we send back the complete polyhedral visual hull to the client, the disparity range can be refined to a per-pixel level. Thus, the stereo algorithm can further benefit from the more precise constraint.

Given the polyhedral geometry and the camera pose information, we are able to generate a depth map using hardware accelerated off-screen rendering. Once we read the depth map from the Z-buffer, it can be easily converted to a disparity map. This conversion can be decomposed into two simple arithmetic image operations which can be performed using fast image processing library routines.

The Z-Buffer contains the depth value for the front surface, so the disparity map is the upper bound of the disparity for every pixel. In order to generate the lower bound to form a range, we can simply subtract the upper bound by a common threshold. The selection of the threshold depends on the maximum concavity of the actual object. Since the concavity of the object is not known in advance for a dynamic scene, this approach does not guarantee the correct disparity range coverage for the object. However, for capturing the action of a moving person who does not exhibit obvious concavities, this approximate range constraint works very well. Fig. 4 shows the result.

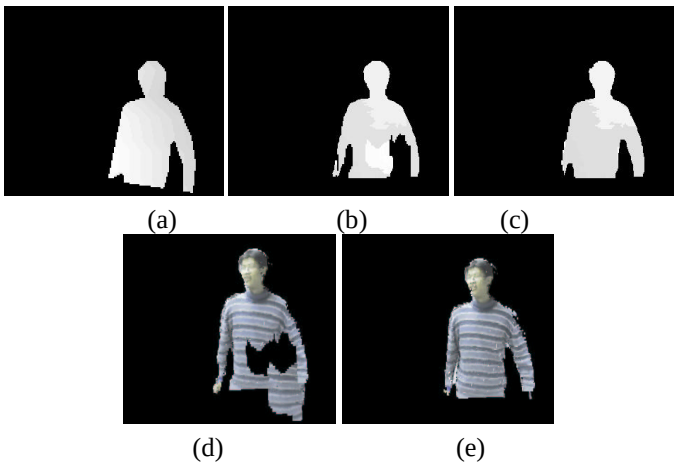


Fig. 4. Improved stereo reconstruction. (a) Disparity map generated from the visual hull. (b) Depth map without using visual hull information (c) Improved depth map using approximate range limits. (d) Rendered view from depth map in *b.* (e) Rendered view from depth map in *c*.

For the previous approach, we only make use of the front surface of the visual hull. If we also know the depth value of the back surfaces, we can get both the upper and lower bound of the disparity. This conservative range is necessary for correct depth recovery. One disadvantage of this approach is that it needs two OpenGL rendering passes to generate the depth map for both front and back surface. But since the visual hull is rendered using graphics hardware, the performance of the whole system is not decreased too much.

IV. Rendering

To synthesize a novel view from multiple fully-calibrated color images with depth information, one can merge the depth maps into a single polyhedral model and render it using texture mapping. Unfortunately, the polyhedral model cannot be generated in real time. Therefore, we use 3D warping [13] for

rendering which directly warps source images to the user view. We warp the stereo pair images that are closest to the user view direction. We have implemented the incremental computation [16] in order to speed up the warping computation. The rendering is decoupled from the 3D reconstruction processing cycle. It can run much faster than the 3D reconstruction. Fig. 5 shows a sequence of dynamic events rendered from a novel viewpoint.

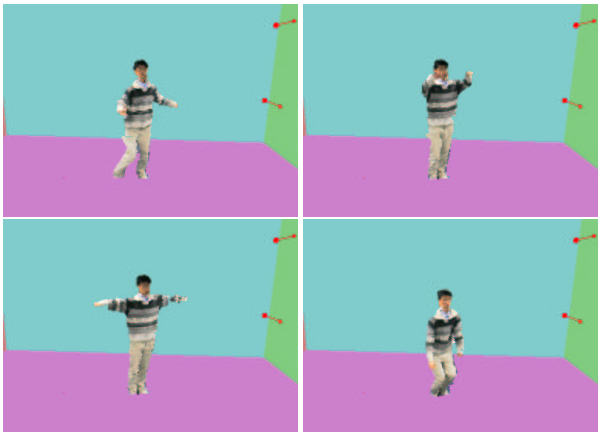


Fig. 5. Rendering result. A moving person rendered from a novel viewpoint at about 10 fps. The two arrows show the positions of the camera pair(about 30° away from the user's viewpoint).

V. Implementation and timings

We use 6 Sony DFW500 FireWire video cameras, which are connected to 3 Linux PCs. All of them have a single Athlon 1.1GHz CPU, nVidia GeForce2 graphics card and 768MB RAM. Another machine with the same configuration is used as the server. The video capturing resolution is set to 320 x 240 pixels, while the off-screen rendering of the depth maps and the stereo computation run at half of that resolution in order to get higher frame rates.

We use a multi-thread implementation on both the client machines and the server to achieve 2.5-3.8 fps for reconstruction. The rendering is running at about 10 fps. Because most of the computation is done in parallel, we can expect a speedup by nearly a factor of 2 when dual-processor machines are used.

VI. Conclusion and future work

In this paper, we present a combined method for the reconstruction of dynamic scenes in an on-line processing system. We use the polyhedral visual hull representation as an estimate of the object in the scene. The approximate geometry imposes constraints on the stereo algorithm to improve and speed up the recovery result. The stereo algorithm overcomes the drawbacks of the Visual Hull method and can recover geometry detail and concave regions of the object. Compared to previous systems, better reconstruction results are achieved at interactive frame rates. This is made possible by the fast reconstruction of the visual hull, distributed computation across several machines and processors, and by using optimized image processing libraries.

In the future, we are going to develop advanced reconstruction and rendering algorithms based on the combination of stereo and visual hull, such as using visual hull information to help merge the depth maps into one consistent model and to predict occlusion information for stereo reconstruction. Higher frame rates are always desirable for developing a real time system. Therefore, network transfer, compression, and parallelization also need to be further investigated. Finally, we plan to integrate virtual objects and allow real-time interaction between real and virtual objects. The seamless integration and interaction will provide the user with a great sense of immersion.

References

[1] S. Baker, T. Sim, and T. Kanade. A characterization of inherent stereo ambiguities. In *Proceedings of the 8th International Conference on Computer Vision*, pages 428–435, Vancouver, British Columbia, July 2001.

[2] Phillipe Bekaert. Boundary representation library. http://breplibrary.sourceforge.net/.

[3] M. Bichsel. Segmenting simply connected moving-objects in a static scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(11):1138–1142, November 1994.

[4] Kong Man Cheung, Takeo Kanade, J.-Y. Bouguet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proceedings of the 2000 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '00)*, volume 2, pages 714–720, June 2000.

[5] K. Daniilidis, J. Mulligan, R. McKendall, G. Kamberova, D. Schmid, and R. Bajcsy. Real-time 3D tele-immersion. In A. Leonardis et al., editor, *The Confluence of Vision and Graphics*. Kluwer Academic Publishers, 2000.

[6] O. Faugeras, B. Hotz, H. Mathieu, T. Vieville, Zhang Z., P. Fua, E. Theron, M. Laurent, G. Berry, J. Vuillemin, P. Bertin, and C. Proy. Real time correlation based stereo: algorithm implementations and applications. Technical Report 2013, INRIA, 1993.

[7] Olivier Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[8] H. Fuchs, G. Bishop, K. Arthur, L. McMillan, R. Bajcsy, S. Lee, H. Farid, and T. Kanade. Virtual space teleconferencing using a sea of cameras. In *First International Symposium on Medical Robotics and Computer Assisted Surgery*, pages 161–167, 1994.

[9] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 858–863. IEEE Computer Society Press, June 1997.

[10] T. Kanade, P.J. Narayanan, and P.W.Rander. Virtualized reality: Concept and early results. *IEEE Workshop on the Representation of Visual Scenes*, June 1995.

[11] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(2):150–162, February 1994.

[12] Wojciech Matusik, Chris Bueler, and Leonard McMillan. Polyhedral visual hulls for real-time rendering. In *Proceedings of Twelfth Eurographics Workshop on Rendering*, pages 115–125, June 2001.

[13] Leonard McMillan. *An Image-based Approach to Three-Dimensional Computer Graphics*. PhD thesis, University of North Carolina at Chapel Hill, 1997.

[14] Saied Moezzi, Arun Katkere, Don Y. Kuramura, and Ramesh Jain. Reality modeling and visualization from multiple video sequences. *IEEE Computer Graphics and Applications*, 16(6):58–63, November 1996.

[15] Pt Grey Research. Digiclops stereo vision. http://www.ptgrey.com.

[16] Jonathan W. Shade, Steven J. Gortler, Li-Wei He, and Richard Szeliski. Layered depth images. *Computer Graphics*, 32(Annual Conference Series):231–242, August 1998.

[17] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*, chapter 7. Prentice Hall, 1998.

[18] Sundar Vedula, Peter Rander, Hideo Saito, and Takeo Kanade. Modeling, combining, and rendering dynamic real-world events from image sequences. In *Proc. 4th Conference on Virtual Systems and Multimedia (VSMM98)*, pages 326–332, November 1998.