Real-Time 3D Reconstruction for Occlusion-aware Interactions in Mixed Reality

Alexander Ladikos and Nassir Navab

Chair for Computer Aided Medical Procedures (CAMP)
Technische Universität München
Boltzmannstr. 3, 85748 Garching, Germany

Abstract. In this paper, we present a system for performing real-time occlusion-aware interactions in a mixed reality environment. Our system consists of 16 ceiling-mounted cameras observing an interaction space of size 3.70 m x 3.20 m x 2.20 m. We reconstruct the shape of all objects inside the interaction space using a visual hull method at a frame rate of 30 Hz. Due to the interactive speed of the system, the users can act naturally in the interaction space. In addition, since we reconstruct the shape of every object, the users can use their entire body to interact with the virtual objects. This is a significant advantage over marker-based tracking systems, which require a prior setup and tedious calibration steps for every user who wants to use the system. With our system anybody can just enter the interaction space and start interacting naturally. We illustrate the usefulness of our system through two sample applications. The first application is a real-life version of the well known game Pong. With our system, the player can use his whole body as the pad. The second application is concerned with video compositing. It allows a user to integrate himself as well as virtual objects into a prerecorded sequence while correctly handling occlusions.

1 Introduction

The integration of virtual objects into a real scene and their interaction with real objects is one of the key aspects in mixed reality. However, this is a non-trivial problem. To create the illusion of actually belonging to the scene, a virtual object has to behave properly in the face of occlusion. Many existing systems are not capable of handling this case, leading to unconvincing augmentations, where the virtual object appears in front of the occluder. Another important aspect for a convincing presentation is the ability to interact with virtual objects. For instance, this would allow the user to pick up a virtual object and place it at another position. In this paper we present a system which is capable of addressing both the occlusion and the interaction issue to create a convincing mixed reality environment. The user can interact with virtual objects without requiring any additional tools or external tracking while at the same time occlusions are seamlessly handled. Our system is based on the reconstruction of the 3D shape of objects inside an interaction space. To this end we equipped our laboratory

with 16 cameras which are mounted on the ceiling. Each camera creates a foreground/background segmentation which is used to construct the visual hull [1] of all the objects in the scene. This gives us a 3D representation for every object, which in turn allows us to convincingly add virtual objects into the scene and to handle occlusions automatically. One of the key advantages of such a system over more traditional tracking-based systems is that we do not require any a priori information about the objects in the scene. We also do not need any prior setup or calibration for someone to use our system. There can even be multiple people in the interaction space at the same time. This makes our system a good candidate for use in real environments where people can just enter the interaction space, start to interact naturally with the virtual scene and then leave, without having to put on any special equipment. This significantly lowers the barrier to try the system and makes it attractive for presenting it to a wider audience, for instance in museums. We implemented two exemplary applications which highlight the aspects of interaction and occlusion handling respectively. The first application is loosely based on the game Pong. The goal of the game is to prevent a virtual ball which is bouncing between the user and a wall from leaving the interaction space, by placing oneself in its path (see figure 1). This application shows the interaction between real and virtual objects. The second application is more focused on providing correct occlusion handling. This is done in the context of video compositing. We record several sequences in the interaction space at different points in time and use the depth map computed from the 3D reconstruction to join the sequences while correctly handling occlusions.

2 Related Work

In recent years, several real-time 3D reconstruction systems which explicitly recover the visual hull have been proposed [2–7]. However, only [6, 5, 7] actually run at frame-rates which allow interactivity. Other researchers have focused on implicitly computing the visual hull [8–12]. The main difference to the explicit systems is that they only generate an image of the visual hull from a novel viewpoint without recovering an explicit 3D reconstruction. This is acceptable in some cases, but does not allow any form of interaction which requires the full 3D shape (e.g. taking the volume of the object into account). However, it is still possible to use it for collision detection [12, 11]. As is to be expected these systems run faster than comparable systems performing an explicit 3D reconstruction. However, today the explicit reconstruction systems reach real-time performance, so that there is no drawback to making use of the additional information.

Some early work on real-time 3D content capture for mixed reality, was presented in [10]. In this paper a novel view generation system was used to insert 3D avatars of real objects into a virtual environment. The system runs at approximately 25 fps using 15 cameras. However, the aspect of interaction between real and virtual objects was not considered. In [11] the authors present a collision detection scheme which extends the work in [12] allowing the interaction

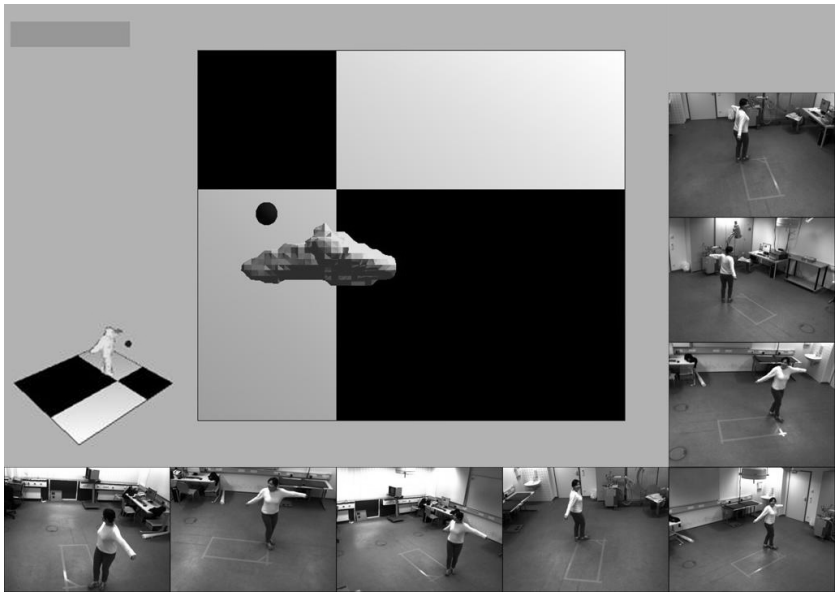

Fig. 1. Our system allows the user to interact with virtual objects using natural movements due to a real-time 3D reconstruction. The images placed around the center show some of the input views while the center and the left side show a orthographic and a perspective view of the scene respectively.

between real and virtual objects. However, they are also not using an explicit 3D reconstruction. In addition their system is running at only 10 fps using 7 cameras which is rather low for real interactivity.

Our system recovers the explicit 3D reconstruction of all objects in the scene at a real-time frame rate of 30 Hz using 16 cameras. This allows us to also perform interactions with objects which are occluded by other objects and would therefore not be visible in a system based on an implicit reconstruction.

3 Real-time 3D Reconstruction System

3.1 System Architecture

Hardware Our system consists of 4 PCs used for the reconstruction, 1 PC used for visualization and 16 cameras mounted on movable aluminum profiles on the ceiling (see figure 2). The cameras have an IEEE 1394b interface and provide color images at a resolution of 1024x768 and a frame rate of 30 Hz. To cover a big working volume we use wide angle lenses with a focal length of 5 mm. The cameras are externally triggered to achieve synchronous image acquisition. Groups of four cameras are connected to one PC using two IEEE 1394b adapter cards. There are four PCs (slaves) dedicated to capturing the images and computing

Fig. 2. Lab setup for our real-time 3D reconstruction system. The cameras are mounted on movable profiles to allow an easy reconfiguration of the camera setup.

the visual hull and one PC (master) dedicated to visualizing the result and controlling the acquisition parameters. The four PCs used for image acquisition and reconstruction are equipped with an Intel 2.6 GHz Quad-Core CPU (Q6700), 2 GB of main memory and a NVIDIA 8800 GTX graphics board with 768 MB of memory. The master PC uses an Intel 3.0 GHz Dual-Core CPU (E6850), 2 GB of main memory and a NVIDIA 8800 GTS graphics board with 640 MB of memory. The PCs are connected through a Gigabit Ethernet network.

Software To achieve real-time performance the reconstruction process (running on the slave PCs) is implemented as a four stage pipeline consisting of image acquisition, silhouette extraction, visual hull computation and volume encoding and transmission. Each pipeline step is realized as a thread and will be described in detail in the following sections. On the master PC the processing is also distributed into several steps. There is a separate thread for handling network communication, compositing the partial reconstructions, visualizing the result and performing the application-specific logic, such as the interactions. Figure 3 gives an overview of the processing steps in the system.

3.2 Calibration

In order to perform the reconstruction the cameras have to be calibrated. The calibration is performed using the multi-camera calibration method proposed in [13]. The method relies on point correspondences between the cameras created

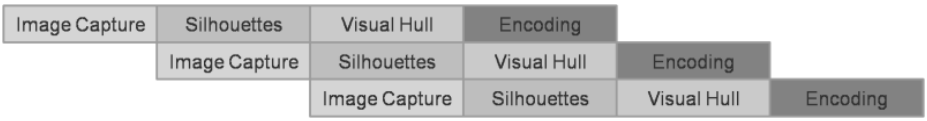

Fig. 3. Reconstruction pipeline on the slave PCs. Each row represents the processing steps taken for each group of simultaneously acquired input images. The processing times are 8 ms, 15 ms, 15 ms and 10 ms respectively. This leads to a very low latency on the slave PCs, which is pushed to about 100 ms when considering image exposure time and computations on the master PC. Due to the pipelining the whole system can run at a frame rate of 30 Hz or higher.

by means of a point light source such as an LED. First, the lighting in the room is dimmed, so that it becomes easier to extract the point created by the LED in the camera images. We run the cameras with a short exposure time and a low frame rate (1 fps) to obtain well-defined light points. By moving the light source through the reconstruction volume a large number of correspondences is created which is then used in a factorization-based algorithm to determine the camera intrinsic and extrinsic parameters. This requires synchronized cameras to make certain that the point seen in each image is created by the same physical point. The method is robust to occlusions of the points in some cameras. The computed camera coordinate system is registered to the room coordinate system by using a calibration target at a known position in the room.

3.3 Reconstruction

Silhouette Extraction The silhouettes are computed using a robust background subtraction algorithm [14] working on color images. Before the system is used background images are acquired. During runtime the images are first corrected for illumination changes using a color mapping table which is built using the color distributions of corresponding non-foreground regions in the current image and the background image. After applying this mapping to the background image, a thresholding is applied to extract the foreground pixels in the current image. Small holes in the segmentation are filled using morphological operations.

Handling Static Occluders While our system seamlessly handles multiple persons in the scene which can occlude each other, one problem which has to be addressed during silhouette extraction is the presence of static occluders. Static occluders are objects inside the working volume which cannot be removed, such as tables mounted to the floor. Hence static occluders are also present in the background images. The assumption during background subtraction, however, is that all foreground objects are located in front of the background. This is not the case in the presence of an occluder because a foreground object could move behind the occluder and effectively disappear from the silhouette image. This will

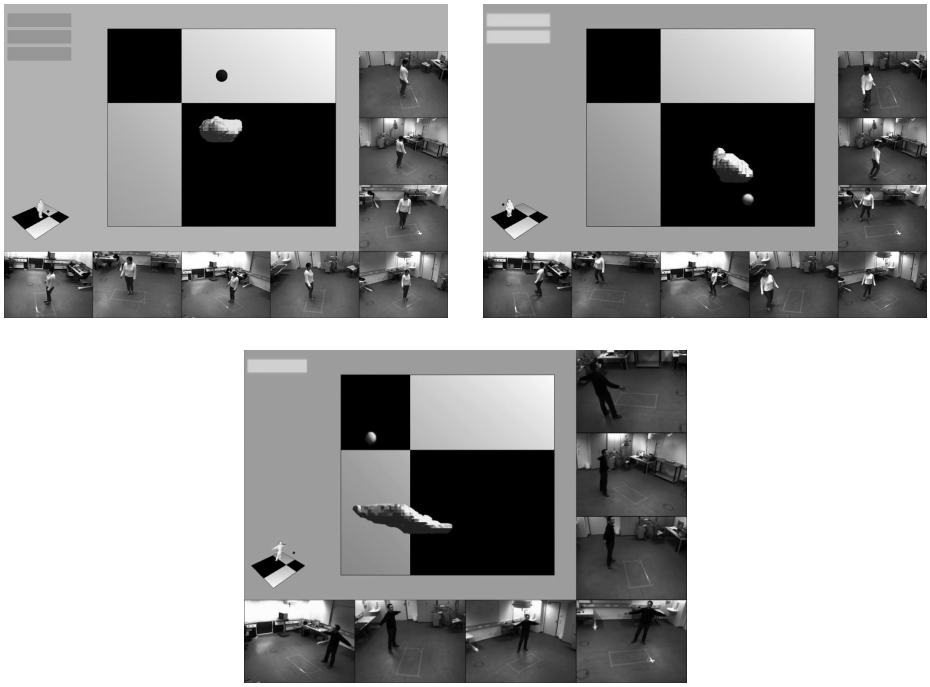

Fig. 4. Our system allows the user to play a game by interacting with a virtual ball. The images placed around the center show some of the input views while the center and the left side show a orthographic and a perspective view of the scene respectively. In the upper left corner the player's remaining life points are shown.

result in the partial or complete removal of the object from the reconstruction. To overcome this problem we use a method which is similar to the one proposed in [15]. The areas in the silhouette images corresponding to the static occluder have to be disregarded during the visual hull computation. We achieve this goal by building a 3D representation of the object and projecting it into the cameras or by manually segmenting the object in the reference images. This gives us a mask for every camera in which the static occluder is marked as foreground. This mask is then added (logical OR) to the silhouette images computed during runtime.

Visual Hull Computation Using the silhouette images the object shape is reconstructed using the GPU-based visual hull algorithm described in [7]. In order to increase the working volume we also reconstruct regions which are only seen by at least four cameras instead of only using the overlapping region of all 16 cameras. This allows us to avoid the use of extreme wide angle lenses for covering a big area, which also results in a higher spatial resolution of the camera images. To reconstruct the non-overlapping regions, one has to consider

the handling of voxels which project outside of the image in other cameras. The traditional approach is to just mark these voxels as empty. Instead, we do not consider the contribution of the images in which the voxels are not visible, thereby also reconstructing regions only seen by a few cameras. To avoid the introduction of artifacts due to a small number of cameras, we only use regions which are seen by at least four cameras. This is implicitly accomplished in our system by performing an unconstrained reconstruction on the slave PCs which also reconstructs the regions seen by only one camera. On the master PC the local reconstructions are combined using a logical AND operator, which will remove any regions which have not been observed by at least one camera at each of the four slave PCs.

3.4 Visualization

For visualization the voxel representation is converted to a mesh representation using a CUDA-based marching cubes implementation. A CPU-based implementation was not able to generate meshes at the desired frame rate. The meshes are visualized on the master PC on a grid showing the origin and the extent of the reconstruction volume. At this time we do not texture the resulting meshes online. However, it is possible to use an offline texturing step to perform this task when required. This is also not a major concern for our system, since we only need the geometrical and depth information to achieve convincing interaction and occlusion handling results.

4 Applications

4.1 Pong

The idea of using mixed reality to create a game in which users interact with virtual objects has already been introduced with the ARHockey system [16]. The ARHockey system used a lot of tracking devices and HMDs to enable the illusion of having a virtual puck which is controlled by the hands of the users. Picking up on this idea we used our system to implement a game which is loosely based on the game Pong. In the original game two players each control a pad and pass a ball between each other. If a player fails to catch the ball his opponent gains a point. We modified the game so that one player is playing against a wall. The goal is to keep the ball from exiting the scene. The player has a certain amount of life points and has to try to keep the ball in the game for as long as possible.

We use a video projector to display the reconstruction of the interaction space on a wall. The user can see himself moving in 3D and he has to position himself, such that the ball is reflected off of him (see figure 4). The collision test is performed between the virtual object and the visual hull. There are two modes. In the first mode we only use the bounding box of the visual hull to perform the collision test. This has the advantage that it is easier for the user to hit the ball, because there is a bigger interaction area. Using the bounding

Fig. 5. Video compositing. We created a new sequence by composing the same video six times in 1 second intervals. Note the correct occlusion handling with respect to the virtual ball and the different time steps of the original sequence.

box also allows children to easily capture the ball, because they can extend their arms to compensate for their lesser body size. The second mode performs the collision test directly between the visual hull and the virtual object. This leads to a more natural interaction, because it is very intuitive. However, the problem here is that it is hard for the user to estimate the height of the ball, so that it might happen that he extends his arm, but the ball passes below it. This problem can be reduced by showing several views of the reconstruction. Optimally a stereoscopic HMD would allow the most natural interaction.

It is also possible for multiple people to play the game at the same time. Due to the use of our reconstruction system the player can use his whole body to catch the ball. The interaction is very natural and people intuitively know how to move to catch the ball. Their movement is not hindered by any additional equipment as would be the case in a tracking-based solution. Even when using a tracking-based solution it would be quite complex to correctly compute the extents of the body. Due to its easy usability and the fact that no setup or training phase is necessary for the user, our system is well suited for use in a real environment, for instance in a museum.

4.2 Video Compositing

As a second application we implemented a video composition system which properly handles occlusions. This is an important topic in mixed/augmented reality [17, 18]. Using our system we took a sequence of a person walking inside the interaction volume. We subsequently used the reconstruction to compute the depth map for one of the cameras. By using both the information from the depth map and the segmentation we created a sequence which shows the same scene at six time steps with an interval of one second in between at the same time (see figure 5). The effect is that instead of one person you can see a queue of 6 copies of the same person walk inside the room. Due to the use of the depth map we correctly handle the occlusion effects. In addition, we added a virtual bouncing ball to the scene which also correctly obeys the occlusion constraints. For creating the composited scene we currently do not apply any image-based refinement on the silhouette borders, but this could be easily added into the system.

These compositing results would be very hard to achieve using purely image-based techniques which do not consider any information about the 3D structure of the scene. It would require a (manual) segmentation of the objects of interest in the entire sequence which is extremely time consuming especially for long sequences. With our solution the segmentation and the depth information is automatically recovered without any additional intervention from the user.

5 Conclusion

We presented a real-time 3D reconstruction system for occlusion-aware interactions in mixed reality environments. Our system consists of 16 cameras and reconstructs the contents of the interaction volume at a frame rate of 30 Hz. The reconstruction is used to allow users to interact naturally with virtual objects inside the scene while correctly handling the problem of occlusions in the augmentation. This is an important aspect in mixed and augmented reality. We demonstrated the results of our system in two application scenarios. The first application is an interactive game which focuses on the interaction aspect, while the second application is a video compositing task which focuses on occlusion handling. In the future we plan to also integrate a tracked HMD into the system, to create an even more immersive experience for the user.

References

1. Laurentini, A.: The visual hull concept for silhouette-based image understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **16** (1994) 150–162
2. Cheung, G., Kanade, T., Bouguet, J.Y., Holler, M.: A real-time system for robust 3d voxel reconstruction of human motions. In: IEEE Conference on Computer Vision and Pattern Recognition. (2000)
3. Borovikov, E., Sussman, A., Davis, L.: A high performance multi-perspective vision studio. In: ACM International Conference on Supercomputing. (2003)

4. Wu, X., Takizawa, O., Matsuyama, T.: Parallel pipeline volume intersection for real-time 3d shape reconstruction on a pc cluster. In: IEEE International Conference on Computer Vision Systems. (2006)
5. Allard, J., Menier, C., Raffin, B., Boyer, E., Faure, F.: Grimage: Markerless 3d interactions. In: SIGGRAPH - Emerging Technologies. (2007)
6. Hasenfratz, J.M., Lapierre, M., Sillion, F.: A real-time system for full body interaction with virtual worlds. Eurographics Symposium on Virtual Environments (2004) 147–156
7. Ladikos, A., Benhimane, S., Navab, N.: Efficient visual hull computation for real-time 3d reconstruction using cuda. In: Proceedings of the 2008 Conference on Computer Vision and Pattern Recognition Workshops. (2008)
8. Matsuik, W., Buehler, C., Raskar, R., Gortler, S., McMillan, L.: Image-based visual hulls. In: SIGGRAPH. (2000)
9. Li, M., Magnor, M., Seidel, H.: Improved hardware-accelerated visual hull rendering. In: Vision, Modeling, and Visualization. (2003)
10. Prince, S., Cheok, A., Farbiz, F., Williamson, T., Johnson, N., Billinghurst, M., Kato, H.: 3D live: Real time captured content for mixed reality. In: ISMAR '02: Proceedings of the 1st IEEE/ACM International Symposium on Mixed and Augmented Reality. (2002)
11. Decker, B.D., Mertens, T., Bekaert, P.: Interactive collision detection for free-viewpoint video. In: GRAPP '07: International Conference on Computer Graphics Theory and Applications. (2007)
12. Lok, B., Naik, S., Whitton, M., Brooks, F.: Incorporating dynamic real objects into immersive virtual environments. In: I3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics. (2003)
13. Svoboda, T., Martinec, D., Pajdla, T.: A convenient multi-camera self-calibration for virtual environments. Presence: Teleoperators and Virtual Environments **14** (2005) 407–422
14. Fukui, S., Iwahori, Y., Itoh, H., Kawanaka, H., Woodham, R.: Robust background subtraction for quick illumination changes. In: PSIVT '06: Pacific-Rim Symposium on Image and Video Technology. (2006)
15. Guan, L., Sinha, S., Franco, J.S., Pollefeys, M.: Visual hull construction in the presence of partial occlusion. In: 3DPVT '06: Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06). (2006)
16. Ohshima, T., Satoh, K., Yamamoto, H., Tamura, H.: Ar2 hockey: A case study of collaborative augmented reality. In: Proceedings of the IEEE VRAIS'98. (1998)
17. Berger, M.: Resolving occlusion in augmented reality : a contour based approach without 3d reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition. (1997)
18. Kim, H., Yang, S., Sohn, K.: 3d reconstruction of stereo images for interaction between real and virtual worlds. In: ISMAR '03: Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality. (2003)