MixIn3D: 3D Mixed Reality with ToF-Camera*

Reinhard Koch, Ingo Schiller, Bogumil Bartczak,
Falko Kellner and Kevin Köser

Institute of Computer Science
Christian-Albrechts-University (CAU)
24098 Kiel, Germany
{rk,ischiller,bartczak,fkellner,koeser}@mip.informatik.uni-kiel.de

Abstract. This work discusses an approach to seamlessly integrate real and virtual scene content by on-the-fly 3D scene modeling and dynamic scene interaction. The key element is a ToF-depth camera, accompanied by color cameras, mounted on a pan-tilt head. The system allows to scan the environment for easy 3D reconstruction, and will track and model dynamically moving objects like human actors in 3D. This allows to compute mutual occlusions between real and virtual objects and correct light and shadow generation with mutual light interaction. No dedicated studio is required, as virtually any room can be turned into a virtual studio with this approach. Since the complete process operates in 3D and produces consistent color and depth sequences, this system can be used for full 3D TV production.

1 Introduction

In movie and television productions, there is a great demand to augment a captured scene by including virtual objects and computer generated elements. In movie production, the effects of computer generated augmentation are usually inserted in post production with very high quality. For TV applications, high-end post production is often too expensive, or not feasible at all if the effects are needed on-the-fly during a live broadcast.

For video augmentation, three components are of importance. First, each frame of a video has to be separated into regions showing virtual content and into regions which should retain the real scene. This separation-process is called keying. Furthermore, the tracking of camera motion has to be performed for a proper alignment of virtual and real content. Finally, the interaction of virtual and real content through mutual occlusions, correct shadow casting and reflections is needed for a convincing augmentation. The typical approach to simultaneously solve all of these challenges, is to build a studio environment equipped

* This work was partially supported by the German Research Foundation (DFG), KO-2044/3-2 and the Project 3D4YOU, Grant 215075 of the ICT (Information and Communication Technologies) Work Programme of the EU's 7^{th} Framework program.

with controlled lighting conditions, chroma keying installations, multiple cameras and sophisticated camera tracking systems using markers and sensors. The construction, maintenance and operation of such studios requires a lot of experience, is expensive and, in certain circumstances, impractical. Even if such a studio is available, the issue of mutual interactions between real and virtual content is unsolved unless a complete 3D scene representation can be computed on-the-fly.

In this work we therefore propose an approach exploiting the capabilities of Time-of-Flight depth cameras (ToF-Cameras) [1, 2][1]. These devices are capable of providing instantaneous depth maps over a limited field of view (app. $40-50°$) at high frame rates (up to 25 fps). Using such a depth camera in combination with a standard or high-definition video camera, our approach is able to provide all the necessary information to obtain fully automatic 3D object keying and camera tracking, which allows for real-virtual interaction without the need of chroma-keying installation, expensive tracking systems or multiple camera sets. An additional benefit is that each video frame is supplied with full depth information, giving the system the potential to be applied for 3D television production.

MixIn3D addresses all of these challenges. In the next section we will present the system's architecture and the building blocks. Section 3 details how interaction between real and virtual objects can be performed. The presented results are discussed in the concluding section 4.

2 System Architecture

The key components of an augmented reality system are keying, camera tracking and interaction between real and virtual content [3]. Keying is the process in which the foreground object regions are separated from the background in an image. One popular way to achieve this is the chroma keying technique. Here the foreground object is captured in front of a screen of constant color, typically green or blue. Under the assumption that the background color is known, deviation from this assumption can be exploited to detect the image regions occupied by foreground objects and to extract an alpha matte [4]. In cases where the screen's color itself does not fulfill the constant color assumptions, due to lighting conditions, this extraction process can deliver faulty results. To deal with this problem, well lit studio setups (virtual studios) can be used. However these lighting conditions, also captured in the image of the keyed foreground, are difficult to match with arbitrary virtual surroundings. The BBC therefore developed True-matting[2], a chroma keying approach where the constantly colored screen is replaced by retro-reflective cloth, which efficiently reflects light only into the direction, where it was coming from. This way a camera equipped with a ring of LEDs, emitting colored light of low intensity, can be used to generate

[1] Companies producing these devices are represented at: www.3dvsystems.com / www.canesta.com / www.mesa-imaging.ch / www.pmdtec.com

[2] www.bbc.co.uk/rd/projects/virtual/truematte/

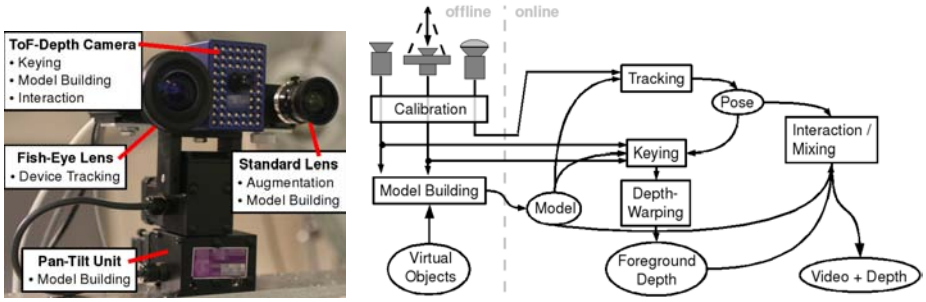

Fig. 1. Major hardware components (left) and their tasks in the system (right).

the required equi-colored background. This technology is expensive and requires to tightly control the environment for a proper segmentation. In order to grant more flexibility, different propositions for the extraction of alpha mattes from images with arbitrary background can be found in the literature [5]. These approaches typically need an initial segmentation of foreground and background and are less reliable. In our approach, we replace the color keying with depth keying, which is discussed in section 2.4.

The 2D keying methods are restricted, since the camera is not allowed to move. To convey a convincing impression of real and virtual content existing together, the camera must be allowed to move, while perspectively correct images of the virtual objects are generated and combined with the real image. This requires the virtual content to be modeled in three dimensions. Furthermore, the real camera pose and projection parameters need to be determined online with the camera motion, so that the virtual object can be rendered perspectively correct. For this purpose, virtual studios are equipped with installations for camera parameter tracking. This tracking equipment ranges from cameras moved by a robot, or using expensive and bulky sensors, to more flexible marker based systems[3] [6]. The installation and proper calibration of such systems is tedious and expensive. Moreover it is required to have a 3D model of the real environment, which contains the relation between the real (physical) scene and the tracking coordinate frame in order to properly align the real and virtual content during the augmentation. In our approach, we automatically model the real environment of the studio by 3D depth scanning. This allows to seamlessly integrate and fuse virtual 3D objects in the real environment for augmentation. Furthermore, the environment model allows for full camera tracking without the need of dedicated markers or robot cameras. Sections 2.3 and 2.5 will address these issues.

The key component of our system is a camera head with a ToF depth camera and a color camera with a field of view of 50°, rigidly coupled together on a computer-controlled pan-tilt unit (PTU). The PTU allows for scanning of the environment in a full sphere of up to $360 \times 180°$, to overcome the limited field of

[3] www.orad.co.il

view of the camera images. In addition, a second camera equipped with a fish-eye lens delivers images with very wide circular field of view of 190°. The use of a fish-eye camera facilitates camera head tracking with high reliability, as noted in [7, 8]. Figure 1 on the left shows the camera head with the cameras rigidly mounted to the PTU. On the left and the right side, two HD color CCD cameras, one equipped with a fish-eye lense, the other viewing through a standard lens, are framing the ToF depth camera in the center. The system operates in two modes, as shown in figure 1 on the right. In an offline phase before the actual shooting, the 3D environment model is scanned by combining the depth and color images in a 3D panoramic model. In the online phase, the model is used for depth keying, camera tracking, content mixing and interaction.

2.1 System Calibration

In order to reliably calibrate the intrinsic and relative extrinsic parameters of all cameras of the rig, a procedure based on a known planar checkerboard calibration pattern is applied. We follow the approach discussed in [9] and [10], which was extended to include the calibration of fish-eye lenses. The procedure starts with capturing a sequence of calibration images of the planar checkerboard pattern in different poses and distances. The checkerboard corners are detected and used to find the extrinsic and intrinsic parameters for each camera individually. At this stage the parameter of the ToF-camera and the color camera with the standard lens are computed as in [11, 12], while the parameters for the camera with the fish-eye lens are estimated using the results from [13].

These initial parameter estimates are used as the starting-point for a non linear optimization over all parameters, integrating the constraints of fixed relative orientations between all cameras. Furthermore, a final bundle adjustment step optimizes the parameters in an iterative analysis-by-synthesis approach. The initial parameters estimated in one iteration are used to synthesize a color image and a depth image of the known checkerboard pattern. The deviation between the real images and the synthetic data is used to compute an optimized set of parameters for the next iteration. Since the synthesized data is free of noise and every point lying on the checkerboard pattern is contributing to the optimization, the reliability of the calibration is significantly improved.

The depth measurements of the ToF-camera suffers from systematic errors [14], which is not only a constant offset but a higher order function [15]. Therefore the iterative optimization is also estimating the parameters of a spline for depth correction as described in [10]. After calibration, residual reprojection errors of 3D scene objects are well below a pixel, yielding sufficient accuracy for our application.

2.2 Time-of-Flight-Camera (ToF) Principle and Image Fusion

ToF is a sensor principle which delivers dense depth images with up to 25 frames per second and currently a resolution of 176x144 pixel. The camera actively

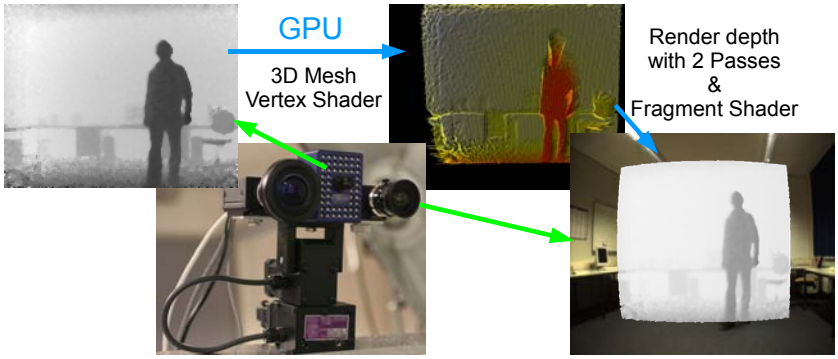

Fig. 2. Depth warping of ToF-image (top left) into the color image (bottom right) with a 3D wire-frame on the GPU (top right).

illuminates the scene by sending out incoherently modulated light from an LED-array with a typical modulation frequency of 20 MHz. The light is reflected at the objects and received by the image sensor of the ToF-camera (cf. [2]). Depending on the object-camera distance, a phase shift in the reflected signal is observable. The ToF-camera is able to extract this phase shift in every pixel and to compute depth from it. For measuring the reflected light, the ToF-camera uses a dedicated semiconductor structure [1].

The phase difference is measured by cross correlation between the sent and received modulated signal of the camera's image sensor. Due to the used modulation frequency, the non-ambiguous range of the ToF-camera is 7.5 meters. From calibration we know that the depth accuracy of the ToF-camera is 20mm or better. Since the resolution of the phase difference measurement is independent from distance, the achievable depth resolution is in a first approximation independent from scene depth. However, since the light intensity fall-off is squared with depth, the signal-noise ratio deteriorates with larger distances and fast sampling rates. Currently we operate the camera with 12.5 fps (80 ms integration time) to keep the depth noise low.

The depth image must be fused with the color images to obtain a combined color-depth video stream. Since the color cameras are displaced from the projection center of the depth camera, a depth-dependent forward-warping is performed that maps the depth pixel into the color camera. We perform the depth warping on the GPU by generating a 3D mesh from the depth camera image, whose z-buffer values are then rendered into the view of the color cameras and scaled for correct depth. This is possible since we obtained all calibration parameters, like projection matrices, radial distortion effects and depth correction from the calibration process with high fidelity. Figure 2 shows an example for depth warping. A challenge is the low resolution of the depth image in comparison to our color camera (1024×768 pixel), where each depth pixel covers an area of about 5×5 color pixels. Thus, the depth image must be upscaled. Depth upscaling to the proper camera viewport is performed automatically during GPU warping, as the wire-frame depth is interpolated during rendering. The low im-

Fig. 3. 3D Panorama of the environment (top row: color and depth) with rendered views of the resulting textured surface (bottom left). Bottom right is showing the position of the camera rig during the model generation.

age resolution of depth is not really a problem, as the spatial depth variation is usually of low frequency w.r.t. color spatial frequencies. The only region where this warping fails are the occlusion regions due to parallax effects of a foreground object occluding parts of the background, as seen in the color camera. However, since in our approach we are only interested in the foreground object area and since we can use the precomputed environment as 3D background model to segment the foreground object, we can eliminate the parallax error. In section 2.4 we will discuss the foreground-background segmentation in more detail.

2.3 Environment Model Building

The 3D environment model is a key feature of our system, which allows for depth keying, camera tracking and 3D interaction. During an offline phase before the actual shooting, a 3D model of the surrounding environment (the studio or any other room) is built by systematically scanning the room with the camera head mounted on the PTU. Each view has only a limited field of view, but stitching together all views will give a complete 3D-color representation of the scene. The PTU yields a very precise orientation stepping of the camera head, and the exact camera pose is obtained by the hand-eye calibration of the camera head. We currently cover an extended field of view of $120 \times 100^\circ$ and allow for image overlap of 50%, which is used for blending of color and depth images. The resulting 3D model is stored as cylindrical panorama for color and depth, since the camera head is not moved during scanning. Note that also a spherical

panorama could be used which allows better modeling of floor and ceiling. The panoramic representation limits the operating range to some extent, since some scene parts are occluded, but this does not pose a fundamental limit, and multiple 3D panoramas could be integrated as well. During the online phase, the 3D panorama is unrolled to a full 3D surface representation.

We have chosen this method since it delivers dense and reliable depth very fast, basically at the speed of the image acquisition. The complete scanning takes about 1 minute for the environment. There are, of course, other possibilities to acquire the environment model. Huhle [16] propose to use a rotation sensor and 3D registration for this. Although very flexible, the danger of errors is high, as the camera motion needs to be computed from the sequence. Stereo estimation, on the other hand (see [17, 18]), might fail in untextured regions. A laser scanner, or structured light approaches as presented in [19], might also solve the problem, but will fail in the online phase for dynamic object tracking.

Figure 3 shows an environment panorama and the corresponding surface model. The number of pixels in this panorama can be very high [4], which consequently leads to very large triangle meshes. Therefore a reduction of the redundant triangles is applied [20].

2.4 Actor Segmentation by Depth Keying

During the online phase, a scene is shot with actors moving through the studio. Real-virtual interaction is possible only if we know the 3D position and geometry of the actors in the room, and if we know the surrounding 3D environment. Both is now possible with the proposed system, without any special keying modification like a blue screen. Since the 3D environment was captured before, and the actor is currently observed by the color-depth camera head, depth based segmentation is easily obtained in any environment. The 3D model is rendered into the current camera view and compared with the acquired depth image. Since the fore- and background are separated in depth and since ToF-cameras capture the distance of a surface to the camera regardless of its shape and color and regardless of the lighting conditions, these devices present a good alternative for color keying, even with cluttered backgrounds and changing illumination.

If D_M is a depth map from the background model and D is a corresponding depth map observed by the ToF-Camera, the resulting mask D_{key} is defined as follows:

$$\mathrm{D}_{key}(u,v) = \begin{cases} \mathrm{D}(u,v) \,, if & \mathrm{D}_M(u,v) - \mathrm{D}(u,v) > \sigma \\ 0 \quad \,, otherwise & \end{cases} . \tag{1}$$

D_{key} is hereby containing the depth value observed by the ToF-Camera. If the depth difference exceeds some threshold $\sigma > 0$, then the pixel is on the foreground object. This pixel-wise decision is filtered to remove spurious measurement noise.

[4] in this work we use 4096×3072 pixel panoramas

Fig. 4. Depth keying: Original color and depth image. Bottom: depth-based segmentation.

As discussed in section 2.2, the measurement noise, the low resolution of the depth camera and the parallax error due to warping attribute to keying errors at the object boundary. Thus, the object boundary might contain segmentation errors that will be visible in the final rendering. Due to the low resolution, a 5 pixel boundary error might occur in the segmentation which is not acceptable for keying. However, we know that we observe an unknown foreground object superimposed on the known background. Thus, we can compare the current color image with the background image and obtain an improved object boundary. This is however sensitive to illumination changes and shadows eventually caused by the person. As can be seen in figure 4 at the bottom left image the segmentation is not perfect, mainly due to the signal noise. Situations in which foreground and background are connected (e.g. the feet of the person) are difficult to handle and further work is needed. A possible enhancement would be to perform the refinement in a different colorspace (e.g. HSL) which is more invariant towards illumination changes. Alternatively a bilateral filter as described in [21] is used to segment the boundary more precisely. This is very time consuming and not possible in real-time. Additionally depth super-resolution upsampling like [22] can help for better segmentation.

Once the object is segmented, a 3D object surface mesh can be computed since both color and depth is available. This object mesh is later used for in-

teraction with the computer generated elements. By merging the object with the background scene model, we have a full 3D reconstruction of the real scene geometry at hand, even the occluded background behind the object.

2.5 Moving the Camera Head

Free movement of the camera is a prerequisite for a versatile virtual studio. Virtual studios with provision to move the camera exist but mostly rely on complex and expensive camera tracking devices [6]. In our approach, we can relax these requirements by exploiting the known environment model. Köser et.al. [23] propose to use visual tracking of a previously constructed environment model with the help of a fish-eye camera. Hereby the tracking does not rely on any artificial markers, since the model itself is used as 3D reference system. The wide field of view of the fish-eye lens will show large parts of the tracked model, even if the camera moves quickly or a dynamic object is occluding parts of the scene. We follow this approach and apply fish-eye camera pose tracking within the environment. The camera tracking can be sustained over long sequences without drift, because the 3D environment model does not change over time.

3 Real-time 3D Interaction

The previous section described the components of the system. In this section we will discuss the interaction capabilities.

The ability to combine virtual content with real image footage using keying and camera parameter tracking already extends the possibilities of virtual studios. However, without a 3D reconstruction of the real part, an interaction between the real and virtual content is difficult to establish. Regarding [24], the most important optical interactions, which significantly improve the augmentation, are occlusions, shadow casting and reflections. It is possible to use a chroma keyer to segment a real object's shadows or to key the reflection of the real object from a shiny surface, but this requires to physically model the real surface. This might be feasible for floors and walls, however for more complicated objects it is normally too expensive to manually construct a virtual model and a physical counterpart for capturing the shadows and reflections, even more so if dynamic virtual content shall be used.

Furthermore, it is not possible to automatically determine mutual occlusions between the real and virtual content without extracting their relative depth distribution w.r.t. the augmented view. Since the content is dynamic and should be reconstructed in real-time, approaches using passive stereoscopy from images and laser scanners are at their limits. In virtual studios, shape-from-silhouette algorithms are used due to their stability and speed, with multiple cameras capturing different views of the dynamic real content. Chroma keying is used to retrieve view-dependent silhouettes, which are combined to a 3D visual hull. Although this hull is not as detailed as laser scans, it often suffices to handle

occlusion and to integrate shadow casting. [24] gives an overview on the limitations and variants of this approach. For good results, multiple wide baseline viewpoints[5] have to be calibrated, captured and evaluated simultaneously. An alternative is provided by the use of ToF-Cameras. As discussed in [25, 26] the depth can be used to handle mutual occlusions without complex algorithmic effort, even without the use of multiple viewpoints. Although simple in its construction and handling, a system using a single viewpoint is only able to deliver a 2.5D model of the reconstructed object. This restricts the scenarios in which correct shadow and reflection calculation can be performed. An approach for visual hull calculation using multiple ToF and color cameras is presented in [27].

We will exploit our proposed system to handle occlusions, shadows, and reflections. Most of these tasks can be performed in real-time or near real-time, allowing instantaneous feedback between director and actor during shooting. More advanced interactions could be added in a post production phase as the full 3D geometry is at hand.

3.1 Mutual Occlusions between Background, Objects and CG Elements

The computation of mutual occlusions between computer generated elements and background, and between the dynamically moving person and computer generated elements is straight forward with our approach. The depth keying delivers all necessary information to compute pixel-accurate depth for the moving object and the background, and a virtual object will be either occluded by the person or will occlude the person and the background, depending on the relative depth w.r.t. the viewing camera. Since the 3D environment was scanned with true 3D metrics, one can locate the computer generated objects at the correct position and with correct metric size without any problem. The objects are placed onto the floor by simply dropping them, and gravity and collision detection with the modeled floor will automatically put the object in place. The camera view is then augmented by color mixing of the computer generated elements into the image at positions where the computer generated objects are not occluded. The rest of the image is taken from the current color image, so consistency is guaranteed, like the correct shadows of the real person on the real walls. Figure 5 shows an example of such depth-based color mixing. Note the mutual occlusions: the statue is occluding the background while being occluded by the real person, which again is occluded by the artificial plants. In the occluding case, a very precise depth segmentation is crucial, and measurement noise might degrade the segmentation. This is also due to the fact that currently our depth camera delivers 12.5 fps while the color camera is running with 30 fps. This is a technical problem which will be solved in the near future, since better and synchronized depth cameras are already announced[6]. The calibration/registration error is well

[5] Following [24] 6 up to 12 cameras are required to deliver reasonable results.

[6] New ToF-cameras are announced which allow full synchronized depth images at 25 fps.

Fig. 5. Color and depth mixing for a frame of an input sequence. Left: Original image of person walking, augmented by virtual objects with mutual occlusion. Right: corresponding depth map used for mixing.

below one pixel so a precise pixel mapping between depth and color camera ist achieved using the mentioned approaches.

Occlusions may appear, but as the baseline of the cameras is small compared to the scene distance occlusions are small. Alternatively 2D/3D-cameras are already in development which use a single lens and a beam splitter for high resolution intensity and depth information. Using this new camera occlusions will no longer be an issue.

3.2 Light and Shadow Casting

While correct depth keying is the key to proper occlusion handling, proper lighting is important for realistic appearance of the computer generated objects. Therefore, also the lights must be modeled accordingly so that the computer generated elements are lit similar to the real scene objects. Furthermore, the computer generated elements must cast realistic shadows onto the elements of the real scene (background and person) and vice versa. Correct shadows are of particular importance for virtual objects placed inside a real scene, since without correct shadow on the floor, the object seems to float in space.

Light interaction is possible in our system once we add a model of the real light sources and light source position. Currently, the light source model is de-

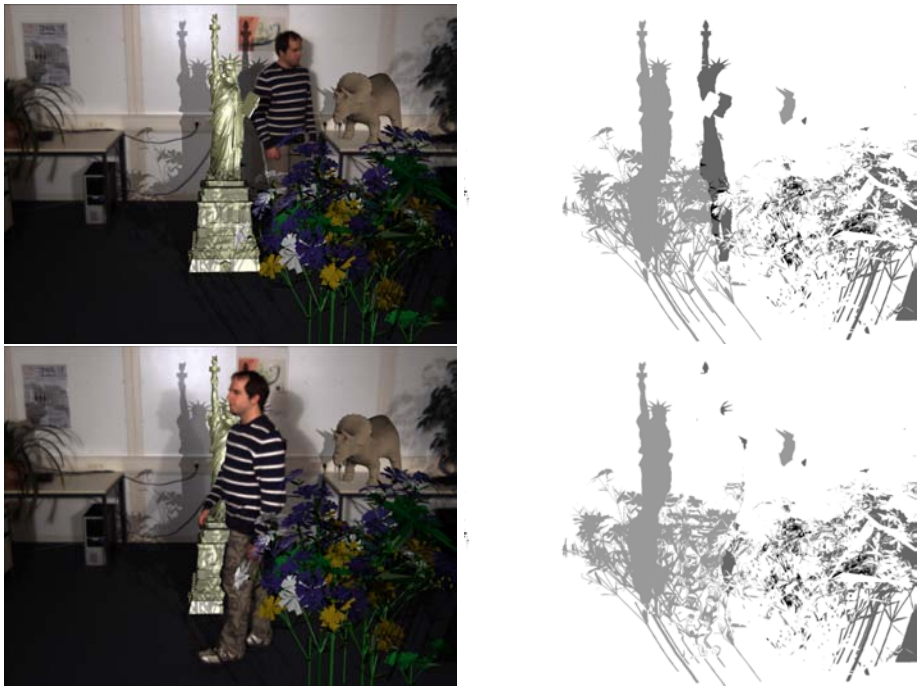

Fig. 6. Shadow casting. Left: mixed image with added shadows, right: shadow map for two light sources.

fined manually and the positions of the lights are selected in the background model by hand, but automated detection of the light source's position is not difficult. For example, the linear light arrays in the ceiling are already part of the model geometry and can be detected easily. Once the light geometry, fraction and temperature is defined, the geometric scene model allows to cast mutual shadows. Note that the light sources are only approximated by point lights for simplicity and real-time capability.

To add the shadows, which are cast by virtual objects onto the real images, light maps are calculated for each video frame. These maps basically encode how much light is reaching a particular pixel of the image when virtual content is present. Each pixel in the light map contains a factor $0 \leq s \leq 1$, which is used to scale the RGB color values in the respective augmented image. A scale factor of 1 corresponds to no shadowing, 0 renders a pixel absolutely black and values in between model partial soft shadowing. The light map operates on the 2D image and reduces only those image parts, which are visible to the user but shaded by an object.

The light maps are generated using the shadow mapping technique [28]. For each light source, a depth map is rendered for all objects that cast shadows. These are the computer generated objects as well as the dynamically moving foreground person. Next, the background model and all (real and virtual) ob-

jects are rendered from the camera's point of view, shading the scene with the calculated lights' depth maps using projective texturing.

This way, for each pixel in the image the distance values R encoded in the light sources' depth maps can be compared to the distances D between a light source and the 3D point corresponding to the pixel. As the light's depth map provides us with the distance between the light source and the first intersection of the light ray with the scene geometry, we can decide whether the pixel is in shadow ($R < D$) or receives light from the light source ($D = R$). Evaluating all light sources and combining them with an ambient light offset yields the view dependent light map used for shadow generation as shown in figure 6. The light maps are additionally filtered with a Gaussian filter to soften the shadows but no real soft shadows are used due to real-time demands.
One exception is made for the interaction between the real foreground and background object. In the target image, the real light source already casts a shadow of the real person onto the background. Hence, the shadow test between real object and background is disabled, while the shadow casting between the real and virtual objects is computed. Figure 6 demonstrates this mutual shadowing. The virtual shadow of the statue of liberty, which is cast onto the back wall, is consistent with the real shadow of the person. Also, the person casts a shadow onto the statue and vice versa. In figure 6 (right), the computed shadow masks can be seen. Only the image areas that are dimmed by the shadows are marked, hence the foreground image region of the statue and the person is left untouched. The problem of double shadows remains, as visible in figure 6 at the bottom left, where the shadows of the Statue of Liberty and of the person are superimposed and doubling each other at the background. This is a known issue and not addressed by our approach. In literature methods are described (e.g. in [29] and [30]) which can be used to solve that issue.

3.3 Surface Reflections

Another cue that is important for correct visual appearance is the reflection of a (real or virtual) object on a shiny surface (see figure 7). A subtle example of this reflection is found on the table surfaces of the background model. A little clay dinosaur is placed onto the table to the right by dropping it there. When looking at the image, something is wrong, since the dinosaur seems to float in space (see figure 7, top right), but there are no visible shadow cues as the light comes from the front. Closer inspection reveals that the table top is slightly reflective, as can be seen by the reflection of the AC current plug and cable on the table. Hence, a reflection of the dinosaur will remedy the problem. In our system, such reflection is easily incorporated, since we know the correct surface normal of the surface, the camera and light viewing direction, and we have a complete 3D environment which is mirrored in the surface. Figure 7 (bottom right) shows the effect of adding the reflection for the dinosaur, where the reflectivity was tuned by hand for this example. Of course, this property can also be exploited to purposely insert mirrored surfaces. In figure 7 (left) we inserted an artificial floor patch with some highly reflective marble surface.

Fig. 7. Object reflection. Right: Dinosaur without (top) and with (bottom) correct surface reflection in the environment model. Left: Reflective marble floor that reflects environment and real person model.

Both, environment model and the dynamic person model, are reflected correctly. This reflection pushes the approach to the limit, since it renders the dynamic object from a very different perspective. Even slight depth errors will produce gross reflection errors, so highest depth-quality is needed.

4 Discussion and Future Directions

Based on a Time-of-Flight depth camera, coupled with color cameras onto a pan-tilt head, we presented a mobile and flexible system for mixed reality applications: MixIn3D. After systematically scanning the environment to set up a background model, the system can be used for keying and occlusion determination between real and virtual objects, for shadowing and reflections.

The major challenge of the current system is to handle correct depth segmentation from the low-resolution depth data of the Time-of-Flight camera. There is much ongoing research and development activity, and improved cameras are already announced. Thus, we are convinced that this problem will be solved in the near future. Also, combining the ToF-data with additional vision-based segmentation algorithms will likely improve the quality further.

The main advantage of MixIn3D, as compared to other mixed reality systems, is that all data is fully available in 3D, so that more complicated interactions are possible - if not in real-time then at least in a post-processing step.

References

1. Xu, Z., Schwarte, R., Heinol, H., Buxbaum, B., Ringbeck., T.: Smart pixel - photonic mixer device (PMD). In: M2VIP 1998 - International Conference on

Mechatronics and Machine Vision in Practice. (1998) 259 – 264
2. Lange, R., Seitz, P., Biber, A., Schwarte, R.: Time-of-Flight range imaging with a custom solid-state imagesensor. In: EOS/SPIE Laser Metrology and Inspection. Volume 3823. (1999)
3. Thomas, G.: Mixed reality techniques for TV and their application for on-set and pre-visualization in film production. In: International Workshop on Mixed Reality Technology for Flimmaking. (2006)
4. Smith, A.R., Blinn, J.F.: Blue screen matting. In: SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM (1996) 259–268
5. Wang, J., Cohen, M.F.: Image and video matting: a survey. Found. Trends. Comput. Graph. Vis. **3**(2) (2007) 97–175
6. Thomas, G.A., Jin, J., Niblett, T., Urquhart, C.: A Versatile Camera Position Measurement System for Virtual Reality. In: Proceedings of International Broadcasting Convention. (1997) 284–289
7. Streckel, B., Koch, R.: Lens model selection for visual tracking. In: Lecture Notes in Computer Science 3663 (DAGM 2005), Vienna, Austria (2005)
8. Chandaria, J., Thomas, G., Bartczak, B., Koeser, K., Koch, R., Becker, M., Bleser, G., Stricker, D., Wohlleber, C., Felsberg, M., Hol, J., Schoen, T., Skoglund, J., Slycke, P., Smeitz, S.: Real-time Camera Tracking in the MATRIS Project. In: Proceedings of International Broadcasting Convention (IBC), Amsterdam,The Netherlands (2006) 321–328
9. Schiller, I., Beder, C., Koch, R.: Calibration of a PMD camera using a planar calibration object together with a multi-camera setup. In: Proceedings of the ISPRS Congress, Bejing, China. (2008)
10. Lindner, M., Schiller, I., Kolb, A., Koch, R.: Time-of-Flight Sensor Calibration for Accurate Range Sensing. Computer Vision and Image Understanding CVIU, Special Issue on Time-of-Flight Camera based Computer Vision. (2009) Accepted for publication.
11. Zhang, Z.: Flexible Camera Calibration by Viewing a Plane from Unknown Orientations. In: Proceedings of the International Conference on Computer Vision, Corfu, Greece (1999) 666–673
12. Bouguet, J.: Visual methods for three-dimensional modelling. PhD thesis, California Institute of Technology (1999)
13. Scaramuzza, D., Martinelli, A., Siegwart, R.: A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion. In: Proceedings of IEEE International Conference of Vision Systems, IEEE (January 2006)
14. Fuchs, S., May, S.: Calibration and Registration for Precise Surface Reconstruction with TOF Cameras. In: Proceedings of the DAGM Dyn3D Workshop, Heidelberg, Germany. (2007)
15. Lindner, M., Kolb, A.: Lateral and Depth Calibration of PMD-Distance Sensors. In: International Symposium on Visual Computing (ISVC06). Volume 2., Springer (2006) 524–533
16. Huhle, B., Jenke, P., Straer, W.: On-the-Fly Scene Acquisition with a Handy Multisensor-System. International Journal of Intelligent Systems Technologies and Applications **5, No.3/4** (2008) 255 – 263
17. Scharstein, D., Szeliski, R., Zabih, R.: A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms. In: Proceedings of IEEE Workshop on Stereo and Multi-BaselineVision, Kauai, HI (December 2001)

18. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In: Proceedings of International Conference Computer Vision and Pattern Recognition (CVPR). (2006)
19. Grundhöfer, A., Bimber, O.: VirtualStudio2Go: digital video composition for real environments. ACM Trans. Graph. **27**(5) (2008) 1–8
20. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: SIGGRAPH 97. (1997) 209–216
21. Anatol, F., Falko, K., Bogumil, B., Reinhard, K.: Generation of 3D-TV LDV-Content with Time-of-Flight Camera. In: 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video, Potsdam, Germany (May 2009) 1–4
22. Yang, Q., Yang, R., Davis, J., Nister, D.: Spatial-Depth Super Resolution for Range Images. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '07. (2007) 1–8
23. Koeser, K., Bartczak, B., Koch, R.: Robust GPU-Assisted Camera Tracking using Free-form Surface Models. Journal of Real Time Image Processing **2** (2007) 133–147
24. Grau, O.: 3D in Content Creation and Post-Production. In Oliver Schreer, Peter Kauff, T.S., ed.: 3D Videocommunication. (2005) 39–53
25. Gvili, R., Kaplan, A., Ofek, E., Yahav, G.: Depth keying. Volume 5006., SPIE (2003) 564–574
26. Bartczak, B., Schiller, I., Beder, C., Koch, R.: Integration of a Time-of-Flight Camera into a Mixed Reality System for Handling Dynamic Scenes, Moving Viewpoints and Occlusions in Real-Time. In: Proceedings of the 3DPVT Workshop, Atlanta, GA, USA (June 2008)
27. Guan, L., Franco, J.S., Pollefeys, M.: 3D Object Reconstruction with Heterogeneous Sensor Data. In: Proceedings of the 3DPVT Workshop, Atlanta, GA, USA (June 2008)
28. Williams, L.: Casting curved shadows on curved surfaces. In: SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques, New York, NY, USA, ACM (1978) 270–274
29. Jacobs, K., Nahmias, J.D., Angus, C., Reche, A., Loscos, C., Steed, A.: Automatic generation of consistent shadows for augmented reality. In: GI '05: Proceedings of Graphics Interface 2005, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, Canadian Human-Computer Communications Society (2005) 113–120
30. Gibson, S., Cook, J., Howard, T., Hubbold, R.: Rapid shadow generation in real-world lighting environments. In: EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2003) 219–229