Markerless Augmented Reality with Light Source Estimation for Direct Illumination

Jan-Michael Frahm, Kevin Koeser, Daniel Grest and Reinhard Koch
Institute of Computer Science and Applied Mathematics
Christian-Albrechts-University of Kiel
Hermann-Rodewald-Str. 3, 24098 Kiel
{jmf,koeser,grest,rk}@mip.informatik.uni-kiel.de

Abstract

We present a markerless augmented reality system with intuitive positioning and realistic direct lighting of virtual objects. Our approach exploits a two camera system consisting of a TV-camera and a fish-eye camera. The TV-camera captures the video for the augmentation while the fish-eye camera observes the upper hemisphere to track the light sources. Accordingly, direct lighting can be used to render virtual objects. Additionally, we propose an easy method to place virtual objects in the scene and to compute the shadows of these objects. The proposed tracking methods overcome the limitations of many systems that need markers in the scene for augmentation or need additional equipment in the scene to reconstruct illumination. Our approach will keep the augmented scene unaffected.

1. Introduction

Realistic illumination of virtual objects placed in real scenes is one of the important challenges of an augmented reality system. This is in particular difficult if it is not possible to place additional equipment in the scene to measure the environmental lighting, which is required by most of the existing approaches [5]. The use of such equipment within the scene is usually not possible for augmentations in TV-studios because in TV often the augmentation pops up only for a short time. Before and after that time slot it is unacceptable to have any additional devices like markers or mirror spheres in view.

The proposed approach computes a realistic illumination by observing the scene and its surrounding environment with a fish-eye camera. This fish-eye camera is used to localise the light sources in the surrounding environment.

Current techniques for the tracking of cameras require expensive tracking systems, which make use of artificial markers like the FreeD tracker from BBC [22] or the HiBall system [25]. These systems require the studio to be equipped with calibrated markers, which is a time-consuming and costly operation. Furthermore, those systems cannot be moved easily from one location to another. In contrast, our technique exploits the features already given in the environment for the camera tracking using fast structure-from-motion and pose estimation [15] algorithms. That provides a cheap and easy way to use the system at different locations.

The paper is organised as follows: the next section will give an overview of the proposed augmented reality system. Afterwards we will review the related work in computer vision and computer graphics. The sections 4 to 8 discuss the different parts of the new system in detail. In section 9 experimental results are presented on sequences captured in a TV-studio and section 10 will summarise the approach.

2 System Overview

The paper presents a markerless multi-camera augmented reality system for TV-production with direct illumination for the virtual objects. In the field of augmented reality it is required to have reliable tracks of the camera pose during the recording. This goal is achieved by our system without placing markers in the field of view of the camera as most of the current systems do. Other systems like the FreeD system [22] which avoid placing markers in the field of view of the camera, place them at least at the ceiling of the studio. They are then observed with an upward looking camera, that is coupled rigidly to the main scene camera. Employing the detected markers in the upward looking camera enables computation of the camera position for augmentation. However, a complicated and expensive calibration and setup procedure is needed for these markers.

We use a two camera system as shown in figure 1. It consists of a TV-camera which captures the images used for the augmentation. The second camera is a fish-eye camera with 180 degrees field of view, that captures images of the upper hemisphere of the scene (studio) and the surrounding environment. Applying structure from motion algorithms from computer vision to the images of both cameras allows computation of the camera positions without any markers in the scene. Our system simply uses the scene and the surrounding environment to identify 3D interest points. The 3D interest points are used to estimate the camera position. We run a reconstruction on the image sequences captured by the TV-camera and the fish-eye camera beforehand. It delivers 3D interest scene points for both cameras. These two independent reconstructions are afterwards aligned automatically to each other.

One of the major limitations of the structure from motion algorithms is that the reconstructed 3D point positions and also the camera poses are determined in a local coordinate system, which may be scaled, rotated and translated against the real-world coordinate system, which makes it difficult to place virtual objects within an absolute coordinate system into the scene. Our proposed system overcomes this limitation by capturing the scene beforehand. During the offline phase the 3D interest scene points are computed automatically. This step establishes a coordinate system for the scene which can be recovered by employing the same 3D interest scene points later. It can be seen as a “learning” of the scene structure.

The learned scene structure will be used during the recording of the images for the subsequent augmentation. It overcomes the problem of inconsistent coordinate frames because now the pose estimation during the augmentation uses the same coordinate system as the reconstruction performed beforehand. Accordingly, the learned 3D scene structure can be employed to place virtual objects.

The fish-eye camera of our multi-camera system is exploited to detect the light sources of the scene. The light sources are usually visible in the fish-eye camera only, while the TV-camera avoids to look into the light sources. Our system will exploit the light sources for direct illumination of the virtual objects placed in the scene.

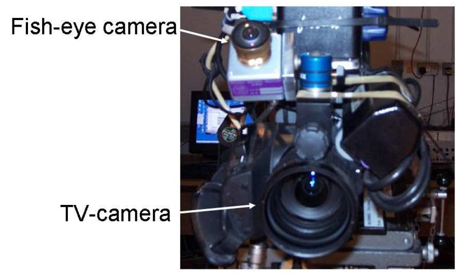

Figure 1. Multi-Camera system of a TV-camera (front camera) and a fish-eye camera.

The estimated light sources are transferred from the fish-eye coordinate system into the coordinate frame of the TV-camera by applying the transformation between both reconstructions. This enables a direct illumination for the virtual object placed in the coordinate system of the TV-camera.

Outline of the approach

1. Take an image sequence before the show using the multi-camera system such that the front camera sees the scene and the fish-eye camera sees the lights
2. Make reconstructions and pose estimations of the videos from the two cameras (see section 4)
3. Temporal Alignment (if the cameras were not aligned and synchronised, section 5.1)
4. Spatial Alignment: Fuse coordinate systems of both reconstructions to allow for position estimation in the scene reconstruction (see section 5.2)
5. Position the virtual object (see section 6)
6. Estimate the light sources using the fish-eye camera (see section 7)
7. Online Tracking and Augmentation (see section 8)

3 Related Work

During the last years augmented reality became more popular due to achievements in computer graphics and computer vision. We will describe the related work in both areas in this section. At first we review the work in the area of computer vision followed by a discussion of the work in the area of computer graphics.

3D scene reconstruction and pose estimation is still a current subject of research in the field of computer vision. In the early nineties the first approaches proved that scene reconstruction was theoretically and practically feasible for a camera moving through an unknown scene with constant but unknown intrinsics [7]. Various methods have been developed that weakened the constraints on the camera [18, 23]. The proposed method for marker-less augmented reality will exploit these approaches to estimate the camera poses. Furthermore there were a few approaches for pose estimation from multi-camera systems proposed during the last year [21, 11]. These approaches exploit the rigid coupling of the cameras to improve the camera pose estimation. An open point of these approaches is the estimation of the rigid transformation between the different cameras. We present an approach to align multiple reconstruction in terms of rotation and scale.

There were several approaches for illumination reconstruction proposed in the last years. In Fournier et al. [10] a radiosity algorithm for diffuse lighting was used considering direct and indirect illumination of the virtual object placed in the real scene. This approach was later modified by Drettakis et al in [6]. Other approaches of considering environmental lighting for rendering of virtual objects were introduced by Debevec et al. in [5, 24]. These approaches use High-Dynamic Range images as environmental illumination. All these approaches place measuring devices like mirror spheres in the scene to capture the environmental illumination. In contrast to these approaches the presented method only uses direct illumination of the virtual object from the light sources of the scene. These light sources are captured by the fish-eye camera. Accordingly, the light poses can be estimated without constraining the camera used to record the images for augmentations. Furthermore "scanning" the lights enables the system to compute direct illumination without measuring with any device in the visible parts of the scene.

4 Markerless camera pose estimation

For an accurate augmentation of the scene the camera pose has to be estimated precisely. Our approach uses a 2-pass-approach: First the 3D features are computed in an offline phase. Afterwards they are tracked in the online augmentation phase. The next section will describe the computation of the offline model from the TV-camera and from the fish-eye camera. Section 4.2 shows the online tracking algorithm for the TV-camera for augmentation.

Figure 2. Images of the original scene used for the offline modelling.

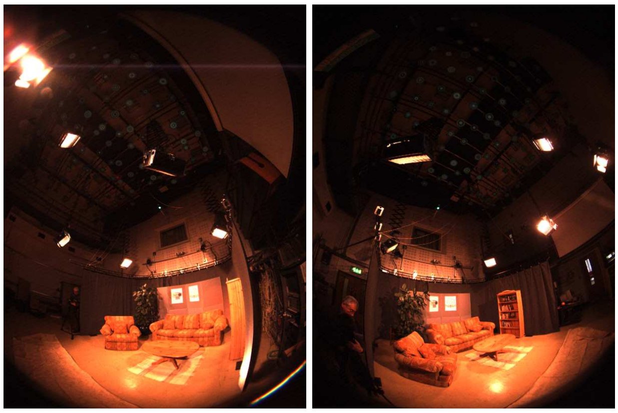

Figure 3. Images of the original scene from the fish-eye camera used for the offline modelling and light source tracking.

4.1 Offline model

For estimating the camera poses a structure-from-motion (SfM) approach is exploited based on [18]. It obtains a metric scene model for each camera as well as it delivers the metric camera projection matrices P_i for each camera for every frame. The scene model is scaled, translated and rotated with respect to the real world coordinate system. The metric camera projection matrices are defined in the scaled metric scene coordinate system. They can be written as $P = K[R^T| - R^T C]$, where K is a 3×3 matrix holding the internal camera calibration, R is a 3×3 matrix for the orientation of the camera and C is the position of the camera center in the scaled metric scene coordinate system.

The exploited structure-from-motion algorithm [18] computes relative poses and can also estimate the internal camera calibration for each camera i as long as it is a perspective camera. In augmented reality applications in TV production the cameras are often equipped with lens sensors which measure the internal calibration of the camera. If this data is available, the structure from motion approach can exploit e.g. the given focal length, which improves accuracy. Therefore the calibration matrix K is assumed to be known here.

The fish-eye camera has a different camera model [9] that cannot be used in the standard structure from motion algorithms. We calibrated the intrinsics of the fish-eye camera beforehand with the techniques similar to those in [3, 13]. Afterwards we applied a mapping of fisheye image coordinates into coordinates of a perspective camera with a given field of view and resolution. The resulting virtual camera can have a very wide field of view, because no physical rectification of the image is performed. We use a virtual field of view of 160 degrees. Accordingly, for both cameras the same structure from motion approach can be used.

The SfM approach [18] exploits a 2D feature detector to find good points to track. To find correspondences between different views of the scene, we track the features throughout the 2D sequence using the KLT-tracker [17, 20] and additional feature-based matching. The correspondence matching is guided by the epipolar constraint and robust matching statistics using random sampling consensus (RANSAC) [8, 12] to handle tracking outliers.

The image correspondences are tracked through many images viewing the same scene. All the points referring to a 2D correspondence chain result from projection of the same 3D feature into the images, hence we compute the 3D intersection of all viewing rays by simultaneous estimation of the camera poses and 3D feature points. Afterwards we use every recurrence of a feature in a new image to reduce the uncertainty of the triangulation by an Extended Kalman Filter, which exploits the fact that the scene is assumed to be static. Accordingly, the projection equation

$$x = K[R^T| - R^T\bar{C}]X, \tag{1}$$

of a 3D X point into the camera image point x can be used as measurement equation.

We cannot compute absolute camera pose as the overall scale of the scene is not known, but a scaled metric pose estimate is determined [12]. Figure 2 shows some images of the scene employed for tracking and reconstruction from the TV camera, in figure 2 corresponding frames of the fish-eye camera are shown. An overview of the sparse scene (3D interest scene points) and the resulting camera poses (shown as pyramids) is given in figure 4. Both reconstructions are in their own local coordinate systems. Note that the fish-eye reconstruction contains a much larger volume of 3D points, because points on the ceiling, on the floor and in the environment of the scene are also seen by the camera .

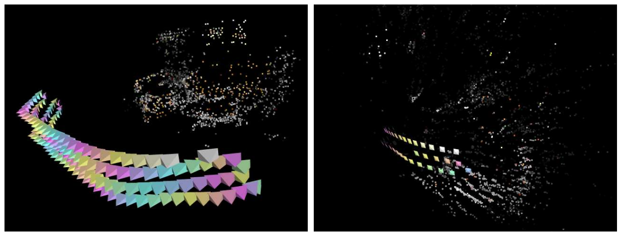

Figure 4. Camera poses and 3D interest scene points of the front camera on the left and of the fish-eye camera on the right.

4.2 Online tracking

After the computation of the 3D interest point set as described above the set of interest points will be employed for online tracking. This section explains this camera pose estimation during the augmentation. It is assumed that the camera moves in the scene which was reconstructed beforehand. Estimating the camera's position in the coordinate system of the offline scene reconstruction requires to solve two tasks:

1. Initial localization of the first camera.
2. Tracking of the camera pose using the 3D interest scene points of the offline reconstruction.

The initial localization of the camera in the previously known scene can be done by feature matching. That means, robust and descriptive signatures (e.g. [16]) are attached to the 3D interest points during the offline phase. When starting the online phase a robust matching algorithm [8, 12] is used to register the camera pose against the scene model.

The initialization provides a camera pose for the first image during the augmentation. Accordingly, it is easy to decide which 3D interest scene points are visible. These 3D interest scene points can be tracked to the following frames with KLT-tracking [17, 20]. It follows that the same structure from motion algorithm can be used for camera pose estimation as for the offline reconstruction. However, in the offline phase we emphasize on exact model building and feature extraction, while in the online phase we go for speed to achieve real-time performance. This extracts the poses of the cameras in exactly the same coordinate system as the previous reconstruction. The online estimated camera poses will be used for the augmentation.

5 Alignment of cameras

The previous section described the tracking of the perspective front camera and the fish-eye camera. If a highly integrated system is used the temporal correspondence of a frame of the fisheye camera and one of the perspective camera is known. However, we provide a solution for the problem when both camera streams are recorded separately and the image sequences have to be aligned in time. That means we have to match the corresponding frames in the two sequences. The correspondence is needed to compute two reconstructions, one from each camera with common scale. Furthermore the reconstructions are rotated and shifted against each other as a consequence of the physical mount and the reconstruction technique used. It is important to understand that not only the physical cameras are rotated against each other in the world coordinate system, but also the reconstructions have their own coordinate systems. Therefore it is not sufficient to simply compare a front and a fisheye camera after the time alignment to get the transformation between the two coordinate systems. We call the procedure to solve that problem the "spatial alignment" of the cameras. The next section will introduce the technique used for alignment in time and afterwards section 5.2 will discuss the spatial alignment of the sequences.

5.1 Alignment in time

The alignment of the image streams in time to get the corresponding frames in both sequences will exploit the relative orientation of consecutive cameras. From the pose estimation we got camera matrices P_i^{front} for the front camera and camera projection matrices for the fish-eye camera $P_m^{\text{fish-eye}}$. Accordingly we are able to compute the relative rotation $R_{i,j}$ between two consecutive frames P_i^{front} and P_j^{front} for the front camera as well as the relative rotation $R_{m,n}$ of $P_m^{\text{fish-eye}}$ and $P_n^{\text{fish-eye}}$ for the fish-eye camera.

A general rotation can be represented by an axis and an angle for the rotation about the axis [2]. Accordingly from the relative rotations $R_{i,j}$ between the frames of the front camera and the relative rotations $R_{m,n}$ of the fish-eye camera we get the corresponding angles $\phi_{i,j}$ and $\phi_{m,n}$. As known from hand-eye calibration techniques in robotics these angles are equal for corresponding frame pairs [19, 4].

To compute the alignment in time we measure the similarity of the relative rotation angles $\phi_{i,j}$ and $\phi_{m,n}$ between consecutive frames. It is done by minimising the sum of absolute differences between the relative rotation angles

$$Sim(i_{\text{first}}, m_{\text{first}}, \Delta i, \Delta m) = \sum_{i,m} |\phi_{i,i+\Delta i} - \phi_{m,m+\Delta m}| \,, \tag{2}$$

where Δi is the offset for the next frame in the front camera and Δm is the offset for the next frame in the fish-eye camera. The first frame of each sequence is determined by i_{first} for the front camera and by m_{first} for the fish-eye camera.

Regarding the different frame rates of the cameras similar to the technique from [4] we use different Δi and Δm for the sequences. To be synchronous, both cameras were triggered by dividing the same master signal of 100Hz. A front camera with a frame rate of 25Hz and a fish-eye camera with frame rate of 10Hz, have a ratio of $25 : 10 = 5 : 2$. Accordingly Δi and Δm are chosen as

$$\Delta i = 5 \text{ and } \Delta m = 2.$$

In order to find the alignment we compute the minimal $Sim(i_{\text{first}}, m_{\text{first}}, \Delta i, \Delta m)$ from (2) for a given range of i_{first} and m_{first} by

$$\min_{i_{\text{first}}, m_{\text{first}}} \{Sim(i_{\text{first}}, m_{\text{first}}, \Delta i, \Delta m)\}. \tag{3}$$

It delivers the alignment in time between the image sequences of both cameras. For the resulting minimum, we found a mean absolute rotation angle error between the fish-eye and the TV-camera of $e_m = 0.025°$, measured over 500 reconstructed camera pairs. Once this alignment in time is known the rotation between both cameras can be computed.

5.2 Spatial alignment

We assume knowledge of temporal correspondence between the two image sequences to compute the spatial alignment. For this task it would be beneficial to use point correspondences between both cameras.

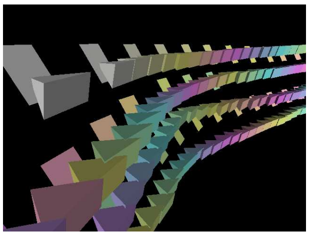

Figure 5. Camera paths aligned in space and time: The pyramids show the TV-camera poses, the flat rectangles show the fish-eye cameras. Due to the different sampling rates not every front camera corresponds to a fish-eye camera.

However, in general an overlapping view of the cameras cannot be assumed. Accordingly, we have to employ some kind of correspondences that can be estimated even for cameras which look into different directions.

The epipolar geometry provides such correspondence [4]. The (calibration-normalised) epipoles e_i and e_m define points on the line at infinity which are only dependent on the rotation between the cameras. The rotation R_{HE} between the cameras maps the epipole of the fish-eye camera into the epipole of the front camera by

$$e_i = R_{\text{HE}} e_m, \tag{4}$$

where R_{HE} is constant for all corresponding frames. It follows that the epipoles of all corresponding image pairs of the two cameras can be used to constrain the rotation R_{HE} between the two cameras. The rotation R_{HE} can be computed by (4) from more than three corresponding epipoles. These epipoles are easily computed using the projection matrices of the reconstructed cameras. Since the TV camera and the fish-eye camera are coupled very closely to each other compared to the distance to the scene, we assume a common projection center for the two cameras. The different scale of the two coordinate systems can then be computed using the temporal correspondence.

The resulting alignment of both cameras using the projection matrices of the initial reconstruction is shown in figure 5. Exploiting the alignment in space and time of the coordinate systems of the two reconstructions from the image sequences of the front camera and the fish-eye camera, objects can be transformed from the fish-eye camera system into the coordinate system of the front camera, which will be exploited for the light sources later on.

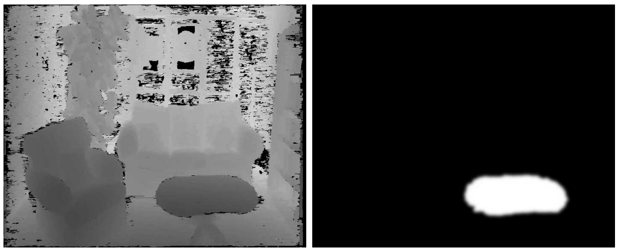

Figure 6. Left: Depth map for calibrated reference image. Right: Automatic plane segmentation after selecting table region.

6 Plane segmentation and coordinate system transformation

Having computed the camera pose we still need to place the virtual object into the 3D scene. Using commercial tools to position and rotate objects in 3D space is still the work of an expert, because tuning the 6 degrees of freedom of an object in space is not very intuitive. Since we usually have 2D screens, the problem to exactly position an object on some arbitrary surface gets even harder. We propose a method to automatically extract a 3D region from the image data with mouse-clicking into the image and fine-tuning the object's pose in a very intuitive way.

To correctly place an object on the table, we select a calibrated image from the offline sequence showing the table. Using standard stereo algorithms [14], we can compute a depth map for the image (see figure 6). Afterwards we simply have to select a rectangle on the table. Using the information from the depth map, a local plane π is fitted to the region in 3D and the standard deviation σ_π of 3D distances to the plane within the region is computed. Afterwards the depth map is segmented into plane inliers and plane outliers using σ_π and the boundaries of the 3D plane are computed using morphological operations and connected component analysis (see figure 6).

The center of gravity of this region together with the plane equation is used to position the 3D object onto the table. The object is automatically transformed in such a way that its normal coincides with the plane's normal and that it is placed exactly on the surface of the plane. Afterwards one may rotate it around the normal and move it within the plane, which is much more intuitive than moving in the complete 6 degrees of freedom. A placed virtual object is shown in figure 7.

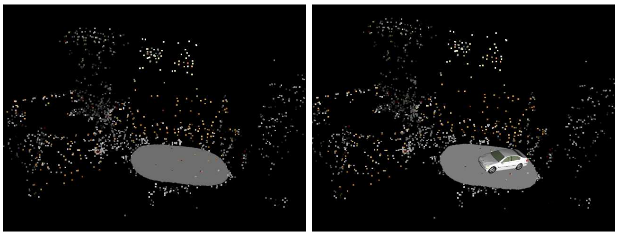

Figure 7. Left: Scene model augmented with segmented plane on the table. Right: Virtual object placed onto the segmented plane.

7 Light source position estimation

The light sources are only visible in the images of the fish-eye camera because the front camera usually avoids to capture the light sources. Two images of the fish-eye camera are shown in figure 3. In these images the light sources can be seen as saturated regions. Hence the image sequence recorded by the fish-eye camera will be exploited to estimate the (point) light source positions. Intensity or colour of the light sources are not recovered so far.

First of all the light sources are selected as image regions with saturation in all channels. For normal cameras this is a natural choice because the light sources are saturated due to the choice of shutter and gain. They are usually chosen to capture the scene information which has a strongly lower intensity than the light sources.

After the segmentation of the light sources the image region covered by the light source is segmented automatically. The automatic segmentation computes in the neighbourhood of the selected point of the light source the mean value of the pixels belonging to the light source and the variance of these pixels. Then a region growing segments the pixel that belong to the light source by exploiting a 99% confidence intensity interval for the light source pixels. Two images with the segmented light sources are shown in figure 8. The center of gravity of the segmented region is assumed to be the light sources image position. The 3D position of the light source in the coordinate system of the fish-eye reconstruction is afterwards triangulated from the different positions of the light source in the images. Exploiting the rotation between the two cameras and the scale between the reconstructions the estimated light source positions are transformed into the coordinate system of the front-camera. The positions of the transformed light sources are shown in figure 9. The

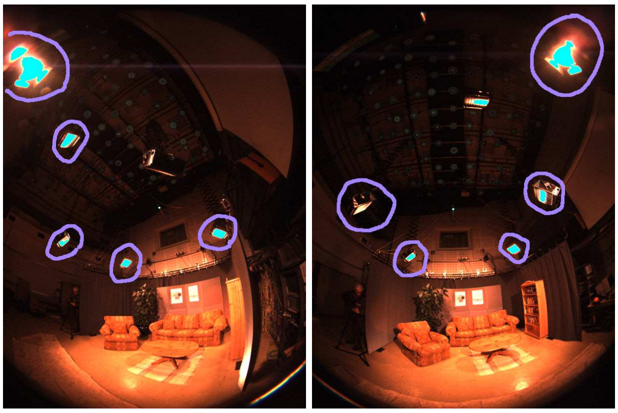

Figure 8. Segmented saturated regions of two images used for triangulation of the light source positions.

Figure 9. Transformed point light source positions in the reconstruction of the front camera, together with 3D scene plane of table and augmented object.

known camera poses of the TV-camera together with the estimated light source positions makes it possible to render augmentations with correct direct illumination. Since the plane geometry and the plane boundaries are known also, this information can be used to compute the shadows of the virtual object for more realistic augmentation.

8 Augmentation

As the lights' positions within the scene are known, it is possible to render objects with realistic shading and shadows. We use OpenSG [1] for rendering, which is based on OpenGL. The image from the original scene is used as a background and the virtual camera is positioned according to the estimated pose of the real camera.

To create the shadows cast from the virtual object onto the real scene, we use a two pass approach, that involves shadow maps. Shadow mapping was introduced by Lance Williams in 1978 [26] and has been extensively used since that time, both in offline rendering and real-time graphics, e.g. in Pixar's Renderman for films such as "Toy Story". It is just one of many different ways of producing shadows and has the following advantages

- Shadow mapping is an image space technique, working automatically with objects created or altered on the GPU.
- Only a single texture is required to hold shadowing information for each light; the stencil buffer is not used.
- Avoids the high fill requirement of shadow volumes.

and disadvantages

- Aliasing occurs, especially when using small shadow maps.
- The scene geometry must be rendered once per light source in order to generate the shadow map for a spotlight.

Especially the fact, that shadow mapping is an image space technique makes it well suited for our augmentation, because we can use a two-pass approach:

1. Create the shadow map for the virtual object standing on the virtual plane, e.g. the table top
2. Render the scene without the virtual plane and use the shadow map from the previous step with the original camera image as background

Rendering with real-time shadows using shadow maps creation is going to be part of the OpenSG library [1]. We use an unpublished experimental version of it to compute the shadows cast by the virtual object onto a scene surface. For the example scenario of the virtual car on the table plane from figure 7 the shadowing is shown in figure 10.

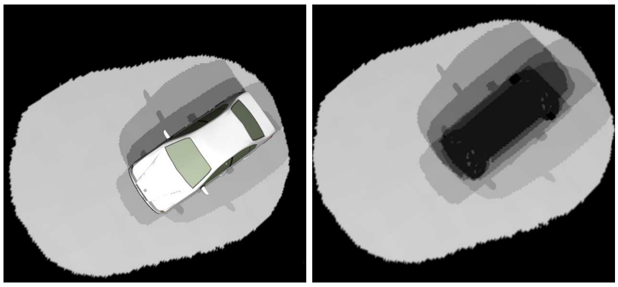

Figure 10. Left: Shadows of the virtual object on the scene plane. Right: Shadow map.

9 Experimental results

To verify the proposed technique we evaluated our system on a sequence made in a real TV-studio. At first in the studio a sequence was taken to compute the 3D interest points of the scene offline. The 3D interest points are shown in figure 4. Afterwards a second image sequence was recorded online in the studio. It contains persons sitting in the studio and talking. For this sequence the offline learned 3D interest point set is used to estimate the camera poses.

The light sources of the scene were located employing the fish-eye images of the offline recording and can be seen in figure 8. The transformation between the front and the fish-eye camera was applied to the triangulated light sources to get their positions in the coordinate system of the front camera, which is used for rendering. The table plane was chosen for augmentation. On this scene plane a virtual object was placed as shown in figure 7. Afterwards the shadows of the 3D object on the plane of the table were computed. These shadows were used as alpha texture for the augmentation of the recorded second sequence.

To evaluate the performance of our approach we tested it on different example sequences consisting of several thousand frames. One interesting sequence was taken by a rapidly moving camera man with a handheld camera. In this sequence the camera motion is fast compared to a camera mounted on a pedestal and movements and rotations occur in all directions. The resulting augmentation is stable and the object remains at the same position in the scene throughout the sequence. Two example images from this sequence are shown in figure 11.

The second shown example sequence is recorded by a camera mounted on a pedestal. In this sequence two talking people are inside the scene. The difficulty for this sequence is that the learned offline model is partially occluded because the talking people hide some parts of the learned scene. Furthermore during talking the people move. Accordingly it is prohibited to learn any features on the

Figure 11. Two views of an augmented handheld camera sequence.

Figure 12. Two views of an augmented sequence with talking people.

people as salient 3D interest points. The augmentation of this scene is as stable as for the handheld sequence and the model remains at the same position the whole time. Two example frames of this sequence are shown in figure 12.

10 Conclusion

We presented a markerless augmented reality system which is able to compute the light source positions and the shadows of the virtual object on a 3D scene plane. The system uses a multi-camera system consisting of a TV-camera and a fish-eye camera. The TV-camera was used to record the images for augmentation and to determine the camera position during the augmentation. The pose estimation of this camera exploits the beforehand learned 3D interest scene points. The fish-eye camera that observed the hemisphere was employed to estimate the light source positions. Furthermore we evaluated the performance of our novel system on several example sequences. The evaluation showed that the novel system can even handle critical sequences with rapid camera motion of a handheld camera and with moving people in the scene. Future work should focus on the online estimation of dynamic lighting as well as colour or intensity of light sources to make the illumination even more realistic.

Acknowledgements

We thank BBC R&D for providing the image sequences and EU project MATRIS IST-002013 (www.ist-matris.org) for supporting our work.

References

[1] OpenSceneGraph library (OpenSG). http://www.opensg.org.

[2] S.M. Alessandrini. A motivational example for the numerical solution of the algebraic eigenvalue problem. *SIAM Review*, 40(4), December 1998.

[3] C. Bräuer-Burchardt and K. Voss. A new algorithm to correct fish-eye- and strong wide-angle-lens-distortion from single images. In *In Proceedings of ICIP 2001*, pages 225–228, 2001.

[4] Yaron Caspi and Michal Irani. Aligning non-overlaping sequences. *International Journal of Computer Vision (IJCV)*, 48(1), 2002.

[5] Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. *Computer Graphics*, 32(Annual Conference Series):189–198, 1998.

[6] G. Drettakis, L. Robert, and S. Bougnoux. Interactive common illumination for computer augmented reality. In *Proceedings of the 8th Eurographics Workshop on Rendering*, pages 45–57, 1997.

[7] O. D. Faugeras and M. Herbert. The representation, recognition and locating of 3-d objects. *Intl. Journal of Robotics Research*, 1992.

[8] M. Fishler and R. Bolles. Random sampling consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM*, 24, June 1981.

[9] Margaret Fleck. Perspective projection: the wrong imaging model. Technical Report TR 95-01, Comp. Sci., U. Iowa, 1995.

[10] A. Fournier, A. Gunawan, and C. Romanzini. Common illumination between real and computer generated scenes. In *Proceedings of Graphics Interface*, pages 254–262, 1993.

[11] Jan-Michael Frahm, Kevin Köser, and Reinhard Koch. Pose estimation for multi-camera systems. In *In Proceedings of DAGM 2004, Tuebingen, Germany*, 2004.

[12] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2000.

[13] Juho Kannala and Sami Brandt. A generic camera calibration method for fish-eye lenses. In *17th International Conference on Pattern Recognition (ICPR 2004)*, pages 10–13.

[14] R. Koch, M. Pollefeys, and L. Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proc. ECCV'98*, volume 1 of *LNCS*, pages 55 – 71, Freiburg, 1998. Springer-Verlag.

[15] Reinhard Koch, Kevin Koeser, Birger Streckel, and Jan-Friso Evers-Senne. Markerless image-based 3D tracking for real-time augmented reality applications. In *WIAMIS 2005, Montreux, Switzerland*, 2005.

[16] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[17] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI81*, pages 674–679, 1981.

[18] Marc Pollefeys, Reinhard Koch, and Luc J. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

[19] R.Y.Tsai and R. K. Lenz. A new technique for full autonomous and efficient 3d hand/eye calibration. *IEEE Journal of Robotics and Automation*, 5(3):345–358, June 1989.

[20] J. Shi and C. Tomasi. Good features to track. In *Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, June 1994.

[21] Sarah Tariq and Frank Dellaert. A multi-camera 6-dof pose tracker. In *IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 04)*, 2004.

[22] G. A. Thomas, J. Jin, T. Niblett, and C. Urquhart. A versatile camera position measurement system for virtual reality tv production. In *In Proceedings of International Broadcasting Convention, Amsterdam*, pages 284–189, 1997.

[23] B. Triggs. Autocalibration and the absolute quadric. In *Proceedings CVPR*, pages 609–614, Puerto Rico, USA, June 1997.

[24] Jonas Unger, Andreas Wenger, Tim Hawkins, Andrew Gardner, and Paul Debevec. Capturing and rendering with incident light fields. In *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering.*, pages 141–149, June 2003.

[25] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, Kurtis Keller, and D'nardo Colucci. The hiball tracker: High-performance wide-area tracking for virtual and augmented environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, 1999.

[26] Lance Williams. Casting curved shadows on curved surfaces. In *SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 270–274, New York, NY, USA, 1978. ACM Press.