Enhanced Visual Depth Cues for Collocated Visuo-Haptic Augmented Reality

B. Knörlein
knoerlein@vision.ee.ethz.ch
Computer Vision Laboratory
ETH Zurich

G. Székely
szekely@vision.ee.ethz.ch
Computer Vision Laboratory
ETH Zurich

M. Harders
harders@vision.ee.ethz.ch
Computer Vision Laboratory
ETH Zurich

ABSTRACT

Our current research focuses on the application of visuo-haptic augmented reality in medical training. The setup developed in this context enables collocated haptic interaction with scene objects. In order to allow natural manipulations, provision of appropriate depth cues becomes a crucial factor. Therefore, we have included fast occlusion handling and shadow synthesis in our augmented environment. The occlusion map is initialized using a plane sweep approach, followed by an edge-based optimization via a Mumford-Shah functional. For obtaining the depth map three head mounted cameras are used and a left-right consistency check is performed to provide robustness against half occlusions. Shadowing is implemented via shadow mapping, considering both real and virtual objects. All steps have been implemented on GPU shaders and are performed in real-time.

Keywords: Augmented reality, occlusion handling, shadow casting.

1 INTRODUCTION

Augmented Reality (AR) extends the real environment with virtual components. In a video-see-through system, virtual objects are superimposed onto the image stream of the real world. By integrating a collocated haptic device into the setup, also haptic augmentations can be provided. Our current research focuses on the application of such visuo-haptic AR setups in medical training of open surgical procedures. In this setting, a user actively manipulates the augmented, collocated environment, leading to occlusions between real and virtual objects.

Depth perception has been identified as a major concern in AR systems [28]. This becomes even more critical if interactions with scene objects are performed. Occlusions and shadows provide key cues for visual perception to infer depth, thus also increasing immersion into the simulated environment [4, 29, 22]. Absence or incorrect presentation of these effects can lead to sensory conflicts. Therefore, we developed approaches to incorporate appropriate visual depth cues into our AR rendering pipeline. We focused on accurate occlusion handling and shadow casting. Moreover, since low latency in a visuo-haptic system is crucial for maintaining veridical haptic perception [15], we also targeted providing these visual cues in real-time.

For occlusion handling we first obtain a depth map using a plane sweep algorithm. For this step three cameras integrated into a head mounted display are being used. Thereafter, the depth is used to provide an initial guess of occlusions, which is further optimized through a segmentation process using a Mumford-Shah functional. The occlusions are optimized by incorporating the total variation of the color images to align the occlusions with image edges. The main contribution of our approach is that no a priori information about the environment is required. This is in contrast to most other approaches, which first perform a segmentation in order to extract the possible occluders. Finally, the reconstructed depth is employed to cast shadows in the augmented scene. The details of these steps will be explained below.

2 RELATED WORK

Resolving occlusions is still a major issue in current AR research. Approaches can be divided into four categories: model-, segmentation-, contour-, and depth-based techniques. The former two require a priori information about the real environment. Model based approaches use the shape and pose of the occluder [6], while segmentation-based techniques incorporate knowledge about the appearance of the occluder or the background [7, 25]. The latter two approaches have proven to be more flexible. Nonetheless, silhouette-based approaches often need manual input or are not feasible in real-time [2, 17], and occlusions obtained from depth, in general, lack fidelity [32, 27]. Novel approaches make use of a combination of different information for the detection of occluders. A depth recon-

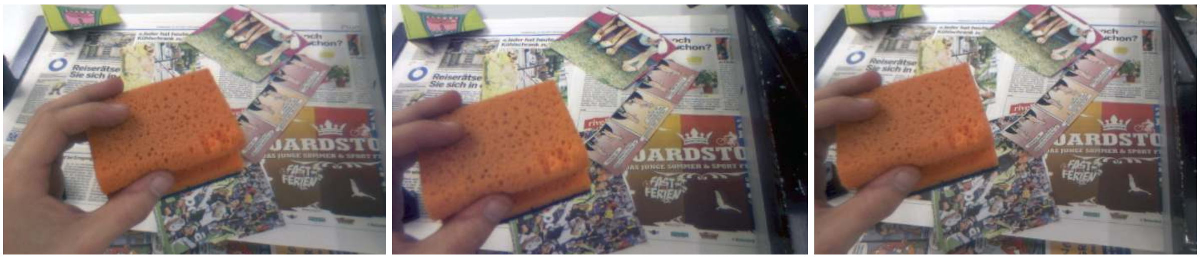

Figure 1: RGB images of left, center, and right camera.

struction step is often combined with a previous segmentation, thus requiring knowledge about appearance [18, 14, 31, 19]. To bypass these problems, a more sophisticated approach, has been presented in [33]. They suggest to employ a probabilistic framework combining color, depth, and neighborhood information to handle occlusions. We also combine depth reconstruction with image segmentation. However, we first reconstruct the depth of the scene and then use this information in a segmentation process in combination with the camera images. No a priori information of the scene is therefore required. Different approaches can be used for the depth reconstruction, e.g. shape from shading [8], but application in AR has to be performed in real time.

Several techniques have been used for shadow casting in AR, e.g. shadow volumes using stencil buffer [10] or shadow textures [13]. More sophisticated approaches use image based information [30] or target the detection, fusion and adjustments of virtual and real shadows [24, 12, 23]. While knowledge about scene depth has been used in other contexts in AR [9, 16], the application to shadow casting for dynamic objects has not been considered before. We use a soft shadow algorithm combining shadow maps with the scene depth reconstruction. A similar approach for casting static shadows has recently been reported in [20].

3 VISUO-HAPTIC ENVIRONMENT

We use a stereoscopic video see-through setup with three dragonfly2 firewire cameras, weighing in total less than 300g. Two of the cameras are placed in front of the head mounted display (HMD) to record the live streams shown to the user. They are aligned with the HMD displays with a baseline of 65mm. A third, centered camera is mounted on the HMD to support the depth reconstruction process. The HMD provides a 40 degree field of view (FOV). To increase camera overlap for the depth reconstruction, the image size of the cameras is increased towards the opposite side by 60 pixels (i.e. 5 degrees). However, only a 400x300 pixel region corresponding to 40 degrees FOV is presented to the user. The center camera acquires 500x300 pixel images at a FOV of about 50 degrees. The colorspaces of the three cameras have been precalibrated to each other by using a color checker. An example view of the environment captured by the three cameras is shown in Figure 1. The head pose of the user is provided by an external infrared tracking system. Haptic feedback is rendered on a PHANToM 1.5 interface, simulating the kinesthetic force response from virtual scene objects. The overall setup and the HMD used are depicted in Figure 2. More details on the system and the calibration procedures can be found in [11]. Known real objects in the scene, e.g. the haptic device stylus or the table, are registered in the environment and represented by virtual counterparts. The latter can for instance be used for shadow casting as well as occlusion handling.

4 DEPTH RECONSTRUCTION

In order to recover the scene depth in real-time we extend the approach suggested in [3]. Depth is reconstructed via a plane sweep technique. The basic idea of this class of algorithm is to sweep the space between a near and far limit with parallel planes at an arbitrary number of discrete depths. Pixels in a reference image are projected onto corresponding pixels in additional test images at each plane depth. Thereafter, the errors between the RGB values of these pixels are determined. The depth of a pixel in the reference image is then set to that of the plane resulting in a minimal error measure. Cornelis et al. further increase the robustness of this technique by performing a hierarchical plane sweep with connectivity constraints.

When reconstructing depth in a video-see through setup, camera baseline and FOV are given by the hardware specifications. Moreover, the working distance is in general below 1m. This leads to strong

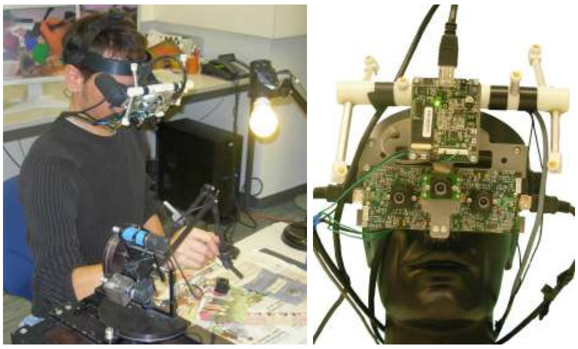

Figure 2: Collocated visuo-haptic AR environment (left) and three camera HMD setup (right).

half occlusions and differences in object appearance in the images (see Figure 1). The original approach by Cornelis et al. is not sufficiently robust in this case. Smooth transitions or wrong depth values for the half occluded regions result. Therefore, we have applied several modifications to their approach.

Coarse level depth initialization: Our depth reconstruction process is performed on a four-level hierarchy. The coarsest level is a 8x8-downsampling of the original reference image. In the first step, the depth for the center view at the coarsest level is determined. The space from 1000 cm to 10 cm is swept by 64 planes. These are equally distributed in reciprocal depth space, thus providing a more equally distributed sampling in image space. For a plane located at depth d, the pixel at $\mathbf{p}_c = (x_c, y_c)$ with color value $\mathcal{C}_c$ in the reference image is projected onto the left and right view, resulting in image coordinates $\mathbf{p}_l^d$ and $\mathbf{p}_r^d$. Using homogeneous coordinates the projection from view a to b is given by

$$\begin{pmatrix} x_b^d \\ y_b^d \\ w_b^d \end{pmatrix} = [\mathbf{Pr}_b^a|\mathbf{0}] \begin{pmatrix} x_a \\ y_a \\ 1 \\ 1 \end{pmatrix} + \frac{1}{d}\mathbf{Pt}_b^a \qquad (1)$$

$\mathbf{Pr}_b^a$ and $\mathbf{Pt}_b^a$ are the 3×3 and 3×1 submatrices of the 3×4 projection matrix given by

$$[\mathbf{Pr}_b^a|\mathbf{Pt}_b^a] = [\mathbf{K}_b\mathbf{R}_b^a\mathbf{K}_a^{-1}|\mathbf{K}_b\mathbf{t}_b^a] \qquad (2)$$

with the internal camera parameters $\mathbf{K}$ and the transformation $[\mathbf{R}_b^a|\mathbf{t}_b^a]$ from view a to b. The projected pixels are then normalized resulting in the image coordinates $\mathbf{p}_{l,r}^d = (x_{l,r}^d/w_{l,r}^d, y_{l,r}^d/w_{l,r}^d)$ and the color values $\mathcal{C}_l(\mathbf{p}_l^d)$ and $\mathcal{C}_r(\mathbf{p}_r^d)$ are interpolated in the full resolution images. Based on these, the matching score for the depth d is determined through sums of squared differences (SSD).

$$\epsilon^d(\mathbf{p}_c) = \frac{SSD(\mathbf{p}_c, \mathbf{p}_l^d) + SSD(\mathbf{p}_c, \mathbf{p}_r^d) + SSD(\mathbf{p}_l^d, \mathbf{p}_r^d)}{3} \qquad (3)$$

determined for coordinates $\mathbf{p}_a$ and $\mathbf{p}_b$ and color values $\mathcal{C}_a$ and $\mathcal{C}_b$ via

$$SSD(\mathbf{p}_a, \mathbf{p}_b) = \frac{1}{|\mathcal{N}_W|} \sum_{\substack{\tilde{\mathbf{p}}_a \in \mathcal{N}_W(\mathbf{p}_a) \\ \tilde{\mathbf{p}}_b \in \mathcal{N}_W(\mathbf{p}_b)}} \|\mathcal{C}_a(\tilde{\mathbf{p}}_a) - \mathcal{C}_b(\tilde{\mathbf{p}}_b)\|_2^2 \qquad (4)$$

in the 3x3 neighborhood

$$\mathcal{N}_W = \{(x, y) : |x - x_0| \leq 1, |y - y_0| \leq 1\} \qquad (5)$$

The initial depth estimate is then obtained through

$$d_{rec}(\mathbf{p}_c) = \begin{cases} d & : \quad \min_d(\epsilon^d(\mathbf{p^c})) \leq \boldsymbol{\epsilon_{max}} \\ \infty & : \quad \epsilon^d(\mathbf{p}^c) > \epsilon_{max} \, \forall d \end{cases} \qquad (6)$$

Thus, if for a pixel no matching score below a threshold ϵ_{max} is found, we assume that a half occlusion occurs. However, it is still possible to determine depths for these points in the subsequent steps. Parameter ϵ_{max} is set to 0.01 for the normalized pixel color range $[0, 1]$.

Hierarchical refinement: In the next steps, the lower-level depth estimates are repeatedly upsampled to higher levels until the original resolution is reached. After upsampling one level, two iterations of a median filter are performed first to reduce the influence of outliers. Thereafter, the depth map is refined using a connectivity constraint. In this step, the depth values of neighboring pixel are evaluated for the current pixel and the best match is set. The search neighborhood is alternated in each iteration between an orthogonally-adjacent

$$\mathcal{N}_{ort} = \{(x, y) : |x - x_0| + |y - y_0| = 1\} \qquad (7)$$

and a diagonal one

$$\mathcal{N}_{diag} = \{(x, y) : |x - x_0| + |y - y_0| = 2\} \qquad (8)$$

Only valid, non-occluded depth values are considered in this step, thus resulting in a first set of neighboring depth values $\mathcal{D}_\mathcal{N}$. Note that these have either been determined during the initial sweep or during one of the following refinements. In order to obtain a piecewise smooth depth approximation, this set is then extended with the mean value of the neighboring depth values $\bar{d}_\mathcal{N}$. This set of possible depth values is then evaluated for the current pixel. To determine the best match, a modification of Equation 6 is used. For computing the SSD this time no neighborhood window is applied.

$$SSD_{fine}(\mathbf{p}_a, \mathbf{p}_b) = \|\mathcal{C}_a(\mathbf{p}_a) - \mathcal{C}_b(\mathbf{p}_b)\|_2^2 \qquad (9)$$

Note that to further prioritize a smooth reconstruction, the matching score of the mean depth is multiplied by 0.75 when finding the depth according to Equation 6. The best matching depth is then assigned to the currently examined pixel.

The described refinement process with connectivity constraints is carried out five times on each level. Additional iterations were not useful, since only negligible changes occurred. The resulting fine-level depth reconstruction for the central view is shown in Figure 3 (right).

Consistency check: The reconstructed depth of the center camera is finally projected onto the left and right view. In order to enhance the result, the previously described refinement step is applied two times, using the left and right view as starting reference images, respectively. However, especially in half occluded regions still mismatches are present. These can for instance be seen in the example image in Figure 3 (middle) in the occluded region right to the sponge.

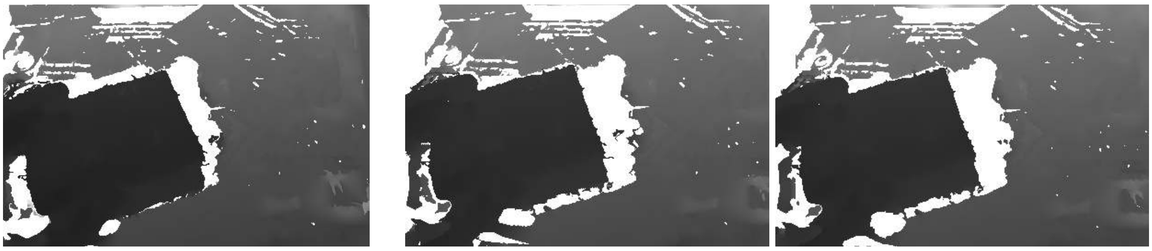

Figure 3: Depth reconstructed for center camera in Figure 1 (left). Fully refined depth map for right camera before and after outliers removal (right).

To remove these outliers, a left-right consistency check is carried out. For both views, pixels are projected from one into the other. The pixel is then backprojected into the first one by using the depth of the second view. If the distance in image space between the backprojected and the original pixel is below a threshold $dist_{max}$, the depth is assumed to be correct. Otherwise the left and right images do not describe the same world point and we set $d_{rec} = \infty$ for that pixel. To allow for small outliers and perspective distortions, $dist_{max}$ was set to 3 pixels.

After this outliers removal the neighbor refinement is again performed two times to fill gaps in the depth maps. The final result after consistency check is also depicted in Figure 3 (right).

In order to determine occlusions, the reconstructed depth d_{rec} could be compared to that of the virtual scene. However, the obtained depth map is still of insufficient quality, exhibiting several inaccuracies as can be seen in Figure 4 (left). Therefore, the determined depths are used to provide an initial guess in a further optimization step.

5 OCCLUSION OPTIMIZATION

The central idea of the occlusion optimization is to align the initial occlusion with image edges. This assumes that the occluding objects exhibit visible contours. To this end, we apply the Mumford-Shah segmentation model [21], resulting in a piecewise smooth approximation of image intensity by minimizing the energy functional

$$E_{MS} = \int_{\Omega} (u-g)^2 \, d\Omega + \alpha \int_{\Omega \setminus \Gamma} |\bigtriangledown u|^2 d\Omega + \beta \, len(\Gamma) \tag{10}$$

where g is the image, u the smooth approximation, and $len(\Gamma)$ the length of the set of evolved edges Γ in g. The positive tuning parameters α and β are related to scale and contrast in the image. Pock et al. [26] showed that the model can be applied, as well, to evolve a piecewise smooth approximation of a color image and a smooth depth map exhibiting the same set of edges Γ by using an iterative update process.

However, this scheme does not allow to account for missing depth values due to half occlusions. Therefore, we first determine a dense occlusion map and assume that half occlusions only effect image background. This is done by comparing the reconstructed depth map of the real environment d_{rec} with that of the virtual scene d_{vr}. In addition, the depth of any known precalibrated objects d_{pre}, e.g. the haptic device stylus, is also integrated. The initial dense occlusion map o is thus determined according to

$$o(x,y) = \begin{cases} 1 & : \ d_{rec}(x,y) < d_{vr}(x,y) \\ 0 & : \ d_{vr}(x,y) \leq d_{rec}(x,y) \\ 0.25 & : \ d_{vr}(x,y) \leq d_{pre}(x,y) \wedge \\ & \quad d_{vr}(x,y) = \infty \\ 0.75 & : \ d_{pre}(x,y) < d_{vr}(x,y) \wedge \\ & \quad d_{vr}(x,y) = \infty \end{cases} \tag{11}$$

An example of an initial occlusion map and the corresponding augmentation is illustrated in Figure 4 (left).

Starting from the initial RGB image $\mathbf{u}_0$ and occlusion map o_0, an iterative update process is then carried out, evolving both towards the edges Γ through linear diffusion. The RGB images are updated by

$$\mathbf{u}^{k+1}(x,y) = \frac{\mathbf{u}^0(x,y) + \sum_{(\tilde{x},\tilde{y}) \in \mathcal{N}(x,y)} \mu^k(x,y,\tilde{x},\tilde{y}) \mathbf{u}^k(\tilde{x},\tilde{y})}{1 + \sum_{(\tilde{x},\tilde{y}) \in \mathcal{N}(x,y)} \mu^k(x,y,\tilde{x},\tilde{y})} \tag{12}$$

where the μ^k are diffusion weights defined below. The neighborhood $\mathcal{N}$ is alternated in each iteration between the previously used $\mathcal{N}_{diag}$ and $\mathcal{N}_{ort}$ to remove sampling artifacts. Next, we update the occlusion map according to

$$o^{k+1}(x,y) = \frac{\delta o^0(x,y) + \sum_{(\tilde{x},\tilde{y}) \in \mathcal{N}(x,y)} \mu^k(x,y,\tilde{x},\tilde{y}) o^k(\tilde{x},\tilde{y})}{\delta + \sum_{(\tilde{x},\tilde{y}) \in \mathcal{N}(x,y,\tilde{x},\tilde{y})} \mu^k(x,y)} \tag{13}$$

where δ is a smoothing weight. However, the latter hinders evolution and does not allow for strong outliers. Pock et al. [26] apply a term including multiple depth hypotheses to encounter the problem. In our case, we use the matching cost ϵ of the assigned depth value $d_{rec}(x,y)$ to regulate the smoothness of the occlusion map with a maximum weight δ_{max}.

$$\delta = \begin{cases} 0 : & d_{vr}(x,y) = \infty \\ \delta_{max} \frac{\epsilon_{max} - \epsilon}{\epsilon_{max}} : & else \end{cases} \tag{14}$$

Thus, the error norm is removed if no correspondence can be detected, while it is preserved if an accurate

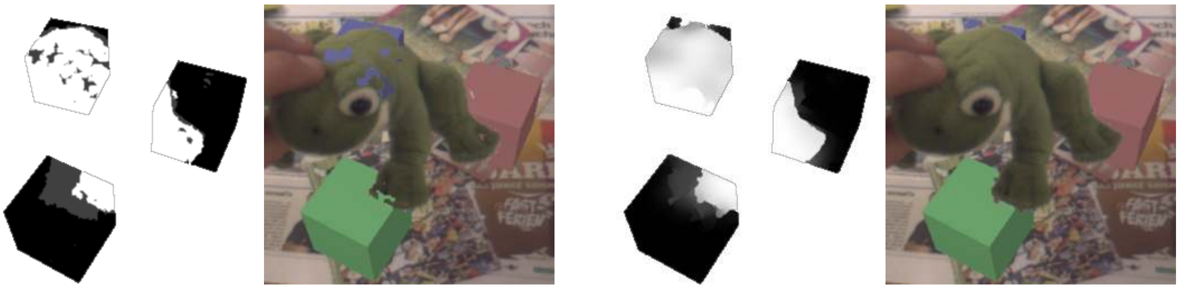

Figure 4: Occlusion map and augmentation before (left) and after refinement (right).

match is found. This results in stronger diffusion in regions where no or less accurate depth values could be found and weaker in those with accurate correspondences. The parameter δ_{max} was set to 100. The diffusion weights μ^k are given by

$$\mu^k(x,y,\tilde{x},\tilde{y}) = \frac{\mathcal{A}_{\mathcal{N}} \cdot \mathcal{B}_{\mathcal{N}}}{1 + \mathcal{B}_{\mathcal{N}} \cdot G(x,y,\tilde{x},\tilde{y})} \tag{15}$$

with $\mathcal{A}_{\mathcal{N}}$ and $\mathcal{B}_{\mathcal{N}}$ being positive constants and G a joint color-occlusion gradient. The former parameters depend on the constants α and β as well as on the specific neighborhood. In our case $\mathcal{A}_{\mathcal{N}}$ and $\mathcal{B}_{\mathcal{N}}$ were given as 15.54 and 705.09 for $\mathcal{N}_{ort}$, and 11.64 and 470.4 for $\mathcal{N}_{diag}$, respectively. Finally, the joint gradient term is specified by

$$\begin{aligned} G(x,y,\tilde{x},\tilde{y}) = \; & \gamma \|\mathbf{u}(x,y) - \mathbf{u}(\tilde{x},\tilde{y})\|_2^2 \\ & + (1-\gamma)\,|o(x,y) - o(\tilde{x},\tilde{y})|^2 \end{aligned} \tag{16}$$

where the γ weights the influence of color and occlusion on the evolved edges. Since strong discontinuities are present in the initial occlusion map, we determine the gradients at the beginning of the iterations only from the color images by setting $\gamma = 1.0$. After 50 iterations the value is changed to 0.9. The overall update process is only carried out inside regions where virtual objects are rendered. If none is present, no information about occlusions is required.

Sufficient results using this optimization were achieved after 100 iterations. Figure 5 illustrates the evolution outcome for a subregion in the example scene of Figure 4. The left image shows the original RGB data located under the rightmost virtual cube, while the other two images depict the evolved edges and the diffused RGB data. The final evolved occlusion map as well as the augmentation is shown on the right of Figure 4. The final occlusion is determined by a binarization of the evolved result with a threshold of 0.5.

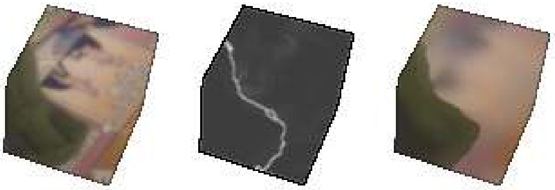

Figure 5: Original RGB image before, and evolved edges and diffused RGB image after optimization.

6 SHADOW CASTING

In AR shadow casting should occur between real and virtual objects. In general, a light emitter casts upon an occluding object, while light occluder and shadow receiver can both be real or virtual. By using shadow maps discussed in [20], convincing soft shadowing can be provided in real-time [1]. If a depth map of the scene is acquired, the approach shows another advantage. A key step for the algorithm is the determination of the depths of the first light-blocking objects. This allows to adjust the shadow map using the previously reconstructed depth map and thereby to include dynamic shadow casting of real objects.

When rendering virtual objects, shadows are cast from virtual and real entities. Therefore, the first occluder in either of the two environments has to be determined. To this end, shadow maps are determined for the virtual S_{virt} and the real S_{real} scene. The former is acquired by rendering the scene from the viewpoint of the tracked real light source. For the latter, the reconstructed depths of the camera views are rendered as point clouds from the view of the light source. Outliers are removed by median filtering. Moreover, the geometries of calibrated objects, e.g. the haptic device stylus or the table, are also integrated. For these, a shadow map S_{pre} is determined by rendering their virtual representation from the direction of the light source. A modified map is obtained by combining this shadow map with the real by determining

$$\tilde{S}_{real} = \begin{cases} S_{real} & : \; S_{real} < (S_{pre} - t_{s1}) \\ S_{pre} & : \; else \end{cases} \tag{17}$$

The threshold t_{s1} is introduced to increase robustness against outliers. In our examples it is set to 10mm. The final shadow map for rendering the augmented scene is then obtained as $S_{fin} = \min(\tilde{S}_{real}, S_{virt})$.

For rendering shadows on real surfaces, we currently, only determine virtual shadows on precalibrated objects, e.g. the table. This is done by casting shadows

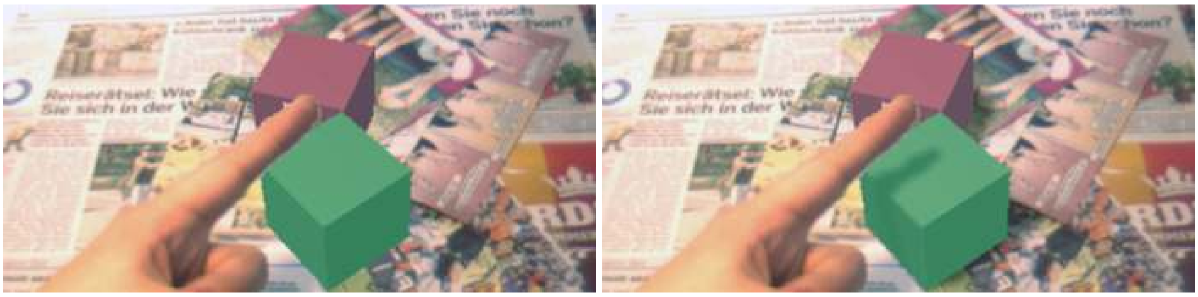

Figure 6: Augmentation without and with shadows cast from a tracked real lightsource.

on the known real world geometry based on the determined virtual shadow map. Here, it has to be noted that occlusion handling has to be performed as well for the newly added shadows. In order to prevent occlusions from the real surface the shadow is cast upon an offset t_{s2} is added to the virtual depth d_{vr} when the occlusion map is initialized through Equation 11. This parameter is also set to 10mm.

Since we do not have information about the diffuse albedo of the real object, the shadowed pixel is modified by multiplying its color value with a shadow coefficient k_s and using an offset off to preserve an ambient light.

$$\mathcal{C}_{out} = \mathcal{C}_{in}\ (off + (1.0 - off)(1.0 - k_s)) \qquad (18)$$

The offset off is set to 0.5. For computing the penumbra of the shadows a percentage-closer filtering is applied [5]. Sixteen texture lookups are used for occluder search and shadow coefficient computation. These are performed based on a randomly rotated Poisson disk distribution. Results of the shadow casting approach are depicted in Figure 6.

7 RESULTS

Visual processing in our AR system is fully implemented on the GPU. Camera images are transferred to GPU texture memory, where all processing is carried out. Unconverted bayer images of all three cameras are uploaded, demosaiced by downsampling to half resolution, color adjusted by precalibrated 3x3 conversion matrices and undistorted. Based on these images the depth reconstruction is carried out. Thereafter,the virtual scene is rendered with shadows and the coarse occlusion map is determined and refined. Finally, the occlusion map is applied and the augmented scene is visualized to the user. Note that rendering the virtual scene and occlusion handing are only performed for the left and right camera only.

The overall process can be carried out at 20 fps on a GTX 285 graphics card. Timings for the different steps of the pipeline are provided in Table 1. The time for the occlusion refinement is given for a masked frame buffer corresponding to the virtual objects. The mask is rendered once to the depth buffer and regions outside the virtual drawing are culled by using an OpenGL depth test. The occlusion handling using the masking requires two rendering passes for either updating the color images or the occlusion map. An alternative implementation without masking can use rendertargets providing updates for both, the color images and the occlusion map, in one render pass. By using this technique occlusion handling can be performed in 38 ms for the full image. Example stereo frames of the full application are shown in Figure 7.

Nevertheless, the depth reconstruction has some limitations. Accurate depth maps can be acquired if the scene is well textured. However, homogeneous areas in the background or repetitive patterns decrease the reliability. In these cases, color keying or background subtraction can be employed to solve for occlusions. Furthermore, specular highlights can lead to additional errors in the depth reconstruction. Still most of these problems can be resolved in the occlusion refinement step.

The choice of the parameter ϵ_{max} directly influences the resulting depth map. A high threshold causes several outliers while lower values result in a sparse depth. The latter, however, is suitable for the occlusion optimization.

The total variation term of the occlusion optimization performs well without incorporation of a priori information. The fidelity of the occlusion is sensitive to the parameters α and β, which can be used to adjust to the scene. High α values result in a kernel (Equation 5) which provides stronger diffusion across image boundaries. In this case, the occlusion in half occluded regions becomes a smooth approximation. In contrast to this, higher values for β increase the influence of

Processing step	Performance in ms
Image upload	1.1
Demosaicing	0.4
Undistortion	0.3
Depth reconstruction	16
Virtual rendering with shadows	2.5
Occlusion refinement	24
Augmentation	0.3
Overall performance	47.6

Table 1: Performance of augmentation pipeline.

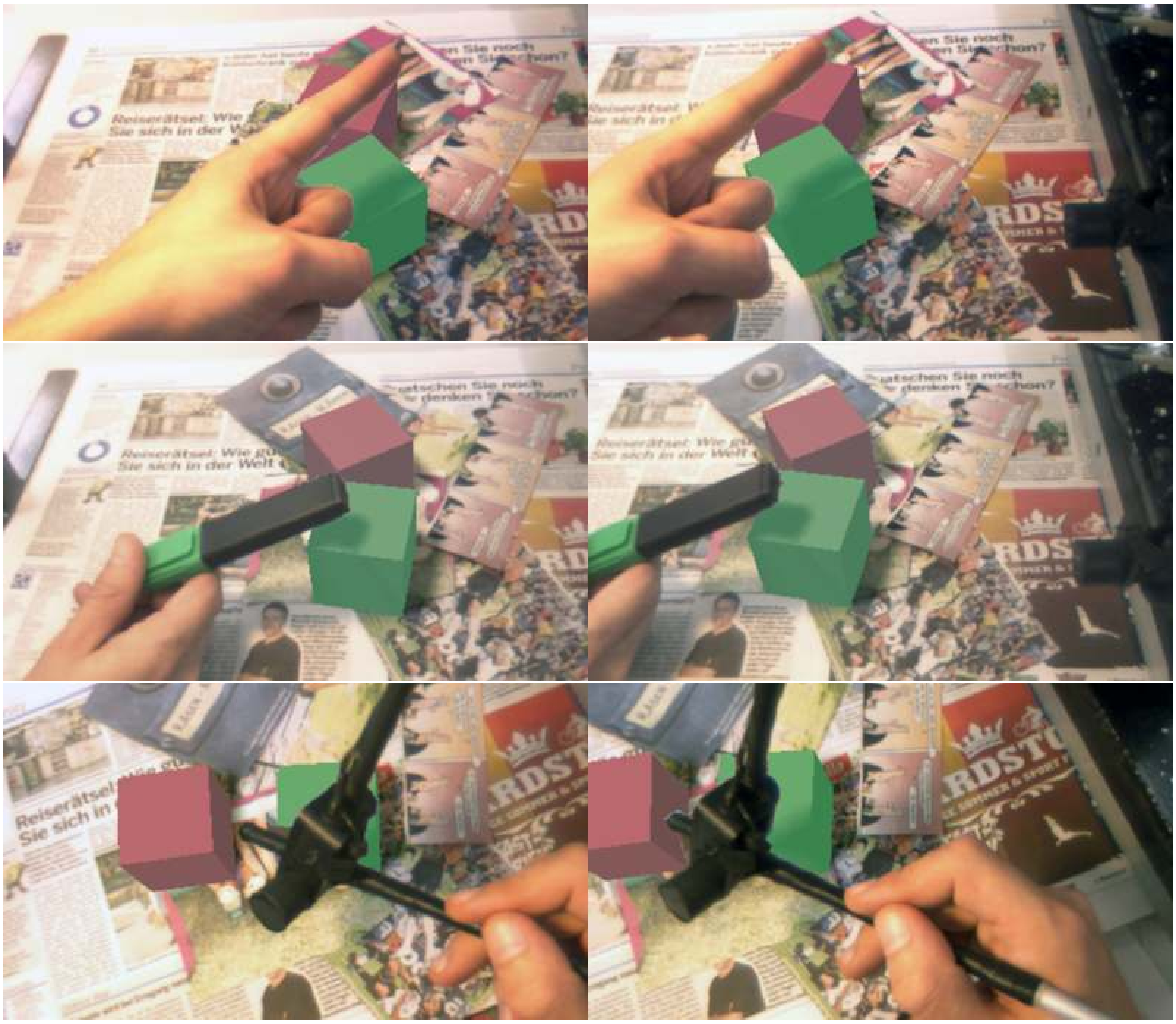

Figure 7: Examples of left and right view of final augmentation during collocated interaction with virtual objects.

edges and reduce the diffusion across image boundaries. However, some problems occur if only small regions are detected for either fore- or background during the depth reconstruction. In this case the refinement tends to prioritize the larger areas. Using an enlarged bounding box for virtual objects could solve this problem. In addition, another problem occurs if no edge between fore- and background objects is evident. This can lead to occlusions leaking from foreground objects into half occluded regions until the diffusion is stopped by a strong edge. Additional stereo or temporal constraints of the occluding contour could be used to encounter this problem. In addition, due to the resemblance of the procedure used to level set implementation, additional energy components, e.g. gradient vector flow or curvature forces could also be integrated. Finally, instead of determining the occluder with a fixed threshold, connectivity of occluding segments and adaptive thresholding could lead to better results.

The depth used for shadow casting was based on the camera view. Projection of the reconstructed depth map only produces correct results, if the perspectives are not too different. It would be better to recover the depth directly from the direction of the light source, for instance by using additional cameras. Noise in the depth maps in general reduces the quality of shadow casting.

8 CONCLUSION

In our current work we are developing a visuo-haptic AR system to be applied in medical training of open surgical procedures. In order to provide enhanced depth cues in the augmented scene, as well as to increase user immersion, we integrated real-time occlusion handling and shadowing in our system. The depth of the scene is recovered using a hierarchical plane sweep process combined with a left-right consistency check. Optimization applying a Mumford-Shah functional is carried out to obtain accurate occlusions. Moreover, the depth map is used for casting dynamic shadows.

Future work will investigate additional constraints for occlusion refinement, e.g temporal or stereo constraints as well as contour curvatures. In addition, a more detailed quantitative evaluation of the approach will be carried out. Furthermore, we will also investigate possibilities for fusing real and virtual shadows in real-time.

Acknowledgments This work has been performed within the frame of the EU project Immersence IST-2006-27141.

REFERENCES

[1] L. Bavoil. Advanced soft shadow mapping techniques. In *GDC slides*, 2008.

[2] M. O. Berger. Resolving occlusion in augmented reality: a contour based approach without 3d reconstruction. In *CVPR*, pages 91–96, 1997.

[3] N. Cornelis and L. V. Gool. Real-time connectivity constrained depth map computation using programmable graphics hardware. *CVPR*, 1:1099–1104, 2005.

[4] D. Drascic and P. Milgram. Perceptual issues in augmented reality. In *SPIE Volume 2653: Stereoscopic Displays and Virtual Reality Systems III*, pages 123–134, 1996.

[5] R. Fernando. Percentage-closer soft shadows. In *SIGGRAPH*, page 35, 2005.

[6] M. Fiala. Dark matter method for correct augmented reality occlusion relationships. *HAVE*, pages 90–93, 2006.

[7] J. Fischer, H. Regenbrecht, and G. Baratoff. Detecting dynamic occlusion in front of static backgrounds for ar scenes. In *EGVE*, pages 153–161, 2003.

[8] D. Gelli and D. Vitulano. Surface recovery by self shading projection. *Signal Processing*, 84(3):467–473, 2004.

[9] G. Gordon, M. Billinghurst, M. Bell, J. Woodfill, B. Kowalik, A. Erendi, and J. Tilander. The use of dense stereo range data in augmented reality. In *ISMAR*, pages 14–23, 2002.

[10] M. Haller, S. Drab, and W. Hartmann. A real-time shadow approach for an augmented reality application using shadow volumes. In *VRST*, pages 56–65, 2003.

[11] M. Harders, G. Bianchi, B. Knoerlein, , and G. Székely. Calibration, registration, and synchronization for high precision augmented reality haptics. *TVCG*, 15(1):138–149, 2009.

[12] K. Jacobs, J.-D. Nahmias, C. Angus, A. Reche, C. Loscos, and A. Steed. Automatic generation of consistent shadows for augmented reality. In *GI*, pages 113–120, 2005.

[13] T. Kakuta, T. Oishi, and K. Ikeuchi. Virtual kawaradera: Fast shadow texture for augmented reality. In *CREST*, pages 79–85, 2005.

[14] H. Kim, D. Min, S. Choi, and K. Sohn. Real-time disparity estimation using foreground segmentation for stereo sequences. *Optical Engineering*, 45(3):037402(1–10), 2006.

[15] B. Knörlein, M. di Luca, and M. Harders. Influence of visual and haptic delays on stiffness perception in augmented reality. In *ISMAR*, pages 49 – 52, 2009.

[16] A. Ladikos and N. Navab. Real-time 3d reconstruction for occlusion-aware interactions in mixed reality. In *ISVC*, November 2009.

[17] V. Lepetit and M.-O. Berger. A semi-automatic method for resolving occlusion in augmented reality. *CVPR*, 2:225–230 vol.2, 2000.

[18] L. Li, T. Guan, and B. Ren. Resolving occlusion between virtual and real scenes for augmented reality applications. In *HCI (2)*, pages 634–642, 2007.

[19] Y. Lu and S. Smith. Gpu-based real-time occlusion in an immersive augmented reality environment. *Journal of Computing and Information Science in Engineering*, 9(2):024501(1–4), 2009.

[20] C. B. Madsen and R. E. Laursen. A scalable gpu-based approach to shading and shadowing for photorealistic real-time augmented reality. In *GRAPP (GM/R)*, pages 252–261, 2007.

[21] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

[22] T. Naemura, T. Nitta, A. Mimura, and H. Harashima. Virtual shadows - enhanced interaction in mixed reality environment. In *VR*, pages 293–294, 2002.

[23] G. Nakano, I. Kitahara, and Y. Ohta. Generating perceptually-correct shadows for mixed reality. *ISMAR*, 0:173–174, 2008.

[24] M. Nielsen and C. B. Madsen. Graph cut based segmentation of soft shadows for seamless removal and augmentation. In *SCIA*, pages 918–927, 2007.

[25] J. Pilet, V. Lepetit, and P. Fua. Retexturing in the presence of complex illumination and occlusions. In *Ismar*, pages 249–258, 2007.

[26] T. Pock, C. Zach, and H. Bischof. Mumford-shah meets stereo: Integration of weak depth hypotheses. In *CVPR*, pages 1–8, 2007.

[27] J. Schmidt, H. Niemann, and S. Vogt. Dense disparity maps in real-time with an application to augmented reality. In *WACV*, pages 225–230, 2002.

[28] T. Sielhorst, C. Bichlmeier, S. Heining, and N. Navab. Depth perception a major issue in medical ar: Evaluation study by twenty surgeons. In *MICCAI*, pages 364–372, 2006.

[29] N. Sugano, H. Kato, and K. Tachibana. The effects of shadow representation of virtual objects in augmented reality. In *ISMAR*, pages 76–83, 2003.

[30] P. Supan, I. Stuppacher, and M. Haller. Image based shadowing in real-time augmented reality. *IJVR*, 5(3):1–7, 2006.

[31] J. Ventura and T. Hollerer. Depth compositing for augmented reality. In *SIGGRAPH posters*, page 1, 2008.

[32] M. M. Wloka and B. G. Anderson. Resolving occlusion in augmented reality. In *SI3D*, pages 5–12, 1995.

[33] J. Zhu, Z. Pan, C. Sun, and W. Chen. Handling occlusions in video-based augmented reality using depth information. In *Computer Animation and Virtual Worlds, Special Issue*, page n/a, 2009.