

# Delta Global Illumination for Mixed Reality

Maik Thöner<sup>1</sup> (✉) and Arjan Kuijper<sup>1,2</sup>

<sup>1</sup> Fraunhofer IGD, Darmstadt, Germany

{maik.thoener, arjan.kuijper}@igd.fraunhofer.de

<sup>2</sup> Technische Universität Darmstadt, Darmstadt, Germany

**Abstract.** The focus point in Mixed Reality applications is the merging of objects from different realities into a new, visibly homogeneous scene. To achieve this, next to a spatial registration, a plausible illumination of the objects is required. While shadows and direct illumination can deliver a realistic look to the objects, adding indirect interaction of illumination will result in a seamless integration of objects, to appear as part of the scene instead of glued-on patches. Mixed Reality systems find application in entertainment areas like movies and games, as well as prototype presentations or visualization of planned or damaged constructions. We propose an algorithm based on Voxel Cone Tracing to provide Global Illumination for Mixed Reality that enables diffuse as well as specular lighting and easy to compute soft-shadows.

## 1 Introduction

The aim of Mixed Reality (MR) is to merge two realities into one coherent frame: a virtual object and a real scene. With the rising popularity of smartphones and see-through displays more and more people have access to hardware that can capture and display an image at the same time [1, 2]. One aim of Mixed Reality is to do calculation and presentation in real time, so that these users can interactively see the result, when the digital data or the scene image changes [3, 4]. Applications for this include entertainment, like games or movies, augmentation as work assistance or for prototyping and damaged or unfinished constructions like cultural heritage or future buildings [5, 6]. For most scenarios the presentation of the digital data is required to be as realistic as possible to create the illusion, that the image shows only one scene, while it contains additional content. We present a system for a more realistic presentation of the digital data to allow a more homogeneous result image.

The aim of our algorithm is to render a virtual 3D object in such a way that it is perceptually integrated into a scene image. To achieve this the 3D object is rendered to get a 2D image of the virtual object, which is then blended with the scene image to create an augmented image of the scene. However, to create the illusion that the augmenting virtual object really is part of the scene several conditions for the pose and lighting during the rendering must be met. We therefore present a combination of Differential Rendering [7] and Voxel Cone Tracing [8] to shade a virtual object according to real lighting conditions

and to transfer the indirect illumination computed with the help of a scene reconstruction onto a real or virtual scene background, resulting in the successful relighting of the real scene.

Our algorithm can simulate the exchange of indirect illumination between virtual and real objects in real-time, is temporally stable, can cast soft shadows of virtual objects onto real surfaces and is able to simulate diffuse and glossy reflections with minimal artifacts.

## 2 Related Work

Mixed Reality applications which seek to fuse virtual objects into real environments with proper global illumination need to address several problems. There have been several publications which are of relevance to our work.

### 2.1 Reconstruction

Reconstruction of the real environment and its lighting conditions is crucial for proper computation of global illumination effects. Debevec reconstructs illumination conditions of the real scene from a reflective ball and computes shading on the virtual objects surface with environment mapping [7]. Sorbier and Saito [9] instead progressively reconstruct and environment map from a Kinect camera. Other methods extract single point light sources from such images with importance sampling [10].

Izadi et al. [11] progressively reconstruct real environment geometry with the help of a geometrically registered depth sensing device (the popular Microsoft Kinect). A static scene can be scanned with the depth sensor, systematically updating a global voxel data structure. Because of noise artifacts in the depth image, which can lead to artifacts in the reconstructed geometry, Chatterjee et al. [12] further investigate filters to create smooth, artifact-free surfaces.

Karsch et al. [13] use automatic reasoning from images with associated depth on single, low-dynamic photographs of a real scene. The reconstruction process yields a depth image similar to a Kinect depth sensor, but without the need of a special sensing devices, which is useful for mobile devices that do not have access to such hardware. They furthermore estimate spatially varying parametric materials using a prior of HDR images with known illumination. Reflectance properties are decomposed into two albedo textures and weighting coefficients.

Finally, many MR methods use pre-reconstructed models of the scene. The scene is usually geometrically registered with a marker or any other type of registration. Materials for this model can also be supplied upfront. In case of known, static MR scenes this method is usually the preferred option.

### 2.2 Shading

When rendering a virtual object into a real context, shading on its surface has to match real lighting conditions. The reconstructed conditions can be fed to an appropriate real-time rendering algorithm to do so.

Many modern real-time global illumination methods build on Instant Radiosity [14]. To approximate indirect illumination, small Virtual Point Lights (VPL) are introduced where a light path intersects geometry. For instance, a bounce off a blue surface is replaced with a blue VPL. A method for rasterizers to create first bounce VPLs has been presented by Dachsbacher and Stamminger [15] by extending a regular shadow map into a full Geometry Buffer (GBuffer) called Reflective Shadow Map (RSM). Assuming purely diffuse reflection, a query into an RSM can be used to turn a pixel with albedo, normal and depth into a VPL with position, direction and flux. A state-of-the-art overview of current real-time global illumination methods can be found in [16].

Viriyothai and Debevec [10] extract multiple point light sources from a spherical image and use regular shading algorithms. However, in dynamic environments many samples are needed to produce a temporally coherent image. Knecht et al. [17] and Lensing and Broll [18] use RSMs for reconstructed real point light sources to introduce first bounce indirect reflections from real reconstructed geometry. Because the number of VPLs need to be low to ensure real-time rendering behavior, both methods use either temporal filtering or importance sampling techniques to create them.

Franke and Jung [19] use irradiance mapping on spherical images in combination with dynamic ambient occlusion. Temporally stable diffuse reflection can be simulated under varying illumination conditions, but virtual objects do not support indirect reflections or more sophisticated reflections off specular materials. Our method aims to support these effects.

### 2.3 Relighting

Introducing an object into a real environment changes its lighting conditions. New objects cast shadows on existing geometry or reflect light back onto real surfaces. Extending global illumination methods to simulate this type of behavior is important to maintain the illusion of a proper augmentation.

Knecht et al. [17] and Lensing and Broll [18] employ Differential Rendering to extract indirect bounces caused by introducing virtual objects. Because this is the same method use to shade the object, the same restrictions apply: only diffuse bounces are supported. Our method can simulate glossy and specular reflections also.

Franke [20] uses a volumetric approach, enhancing Light Propagation Volumes [21]. A small, low-resolution volume around the object is injected with the difference of two RSMs (one containing the real-reconstructed scene and the virtual object, one without the object). The differential inside the volume is then propagated until no more energy can be distributed. Pixels of real-reconstructed geometry covered by this volume can query indirect diffuse bounces off the virtual object. The method solves temporal stability issues and is very efficient, but tends to cut off illumination once it reaches the volume boundaries. The low volume resolution also causes heavy bleeding artifacts through thin geometry. Franke [22] subsequently uses Voxel Cone Tracing [8] to combat bleeding artifacts. The volume resolution is increased, but instead of propagating light

inside the volume, a pre-filtering process creates a chain of smaller volumes. These are used to efficiently cone-trace for indirect reflections, which allows to produce glossy and specular indirect reflections. Our algorithm closely relates to this method.

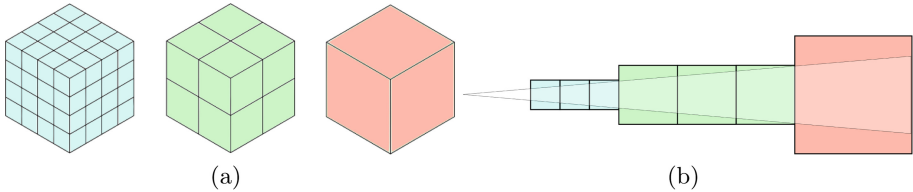
Finally, many MR methods employ regular shadow mapping to introduce shadows of virtual objects on real surfaces. Shadow maps are the subject of heavy investigation and come in a wide variety.

### 3 Delta Global Illumination

In our approach we use Voxel Cone Tracing [8] for the shading and relighting of the 3D object and the scene, as it is the most promising real-time Global Illumination techniques, which produces glossy and specular indirect reflections for a reasonable computational cost.

#### 3.1 Voxel Cone Tracing

Voxel Cone Tracing uses cones to query direct or indirect light that comes from the cone direction to aggregate the illumination. The light data itself is stored in a voxel grid which is inserted by voxelizing VPL candidates. The cone casting itself is simulated by casting a normal ray, but reading from an increasing size of voxels in the volume. Bigger voxels can be achieved by mipmapping the 3D texture which contains the voxel structure. Each mipmap level then contains voxels of  $2^3$ . This is visualized in Fig. 1.



**Fig. 1.** (a) visualizes the filtering of the voxel volumes, (b) the mipmap levels that are read from the cone depending on the distance.

In each tracing step the light information is aggregated depending on the occlusion in and before that step. For that we use two accumulator, one for the light information and one for the occlusion value, which are updated with the following formulas:

$$light_{acc} = light_{acc} \cdot occlusion_{acc} + light_{local} \cdot occlusion_{local} \cdot (1 - occlusion_{acc})$$

$$occlusion_{acc} = occlusion_{acc} + occlusion_{local} \cdot (1 - occlusion_{acc})$$

where  $light_{acc}$  and  $occlusion_{acc}$  are the accumulators and  $light_{local}$  and  $occlusion_{local}$  the values read from the voxel volumes. The occlusion value is thereby adapted with the normal that is stored in the volume. At the end of the tracing the  $light_{acc}$  value is used as the light value of that cone.

The step size during the tracing of the cone is adjusted to the voxel size at that and the next step to avoid reading from overlapping voxels. Because of this the tracing cost of a cone can be controlled by the aperture size. Narrow cones require a smaller step size, since they operate mostly on low mipmap levels, whereas bigger cones will get to higher mipmap levels and thus fasten up the tracing. To calculate the color for a fragment, we cast a series of different cones into the scene. For the diffuse light we use a higher number of cones with a bigger aperture angle which results in some kind of blurriness. These diffuse cones are arranged in a way, that they cover as much of the hemisphere as possible. For sharp highlights like specular reflections we use one narrow cone, whose direction is the view ray reflected with the surface normal. The aperture size defines the sharpness of the features and relates to the surfaces glossiness.

The light information of all cones is then aggregated with a factor based on the viewing and cone angles and the surface hardness.

### 3.2 Algorithm Overview

For the shading of the virtual object it must receive indirect illumination from the surrounding scene, as well as from itself. For the relighting the scene must only receive indirect light from the virtual object. To achieve this with Voxel Cone Tracing, two separate light volumes are needed, one for shading and one for relighting. The one for shading contains the light information of the whole scene including the virtual object. For that we render a reflective shadow map of the virtual object and the scene reconstruction and insert all fragments into the voxel volume via a simple transformation.

Creating the light volume for the scene relighting is more complex. Just rendering a reflective shadow map for the virtual object is not correct, since it would result in indirect light bounces on the virtual object that are in reality occluded by scene geometry. Furthermore the object will occlude scene geometry and to account for that we want to insert negative light so that the light reflected from those surfaces will be subtracted during the scene's relighting step. To create this volume we calculate a delta volume by rendering a reflective shadow map of only the scene reconstruction and subtract the light information from the other volume. This way all surface geometry of the scene reconstruction which is visible in the scene-only reflective shadow map will result in a negative light value inside the voxel grid.

In addition to the light volumes for shading and relighting we use two additional volumes to store the visibility information. These volumes store the normal of the surface and an occlusion factor in each voxel. The normal is the average normal of all geometry that falls into one voxel. The occlusion factor is set to 1 for every voxel that has geometry in it and 0 otherwise. In the higher mipmap levels this factor then denotes the fraction of subvoxels which contained geometry and is used to calculate the occlusion in each tracing step.

Overall we use four volumes in contrast to [20], where only one volume is used for the visibility information. Using an additional volume comes of course with an increased memory cost, but it also solves a problem with the combined volumes. When the visibility information of the scene and the virtual object are mixed the relighting of the scene will introduce additional shadows for occluders in the scene that already cast this shadow, which results in a doubled shadow effect on scene objects. For the cones that query the indirect light information still both visibility volumes have to be used or indirect light from the virtual object will travel through real scene geometry.

### 3.3 Shading vs. Relighting

For the shading and the Relighting we use the same Voxel Cone Tracing technique. The difference lies in the volumes that we query during the cone tracing. Whether the current fragment is part of the virtual object or the scene geometry can be easily determined with help of the depth buffers. For the shading of the virtual object we use the light volume which contains the injected light for all objects, the scene reconstruction and the virtual object itself. That way the virtual object receives indirect light from the scene but also from itself. For the relighting of the scene on the other side we use the delta volume which only contains the radiance field changes introduced by the virtual object.

### 3.4 Volume Extent

The voxel density of the volumes and thus the highlight accuracy is directly dependent on the size of the volumes in the scene. Under the assumption that the significant influences of the introduced virtual object to the scene is limited to a local region around it, the volume for relighting only must cover that region without introducing too much errors. The same is true for the shading volume which defines the region of the scene reconstruction that can influence the virtual object. So there is always a trade-off between the impact range and the voxel density in the volumes.

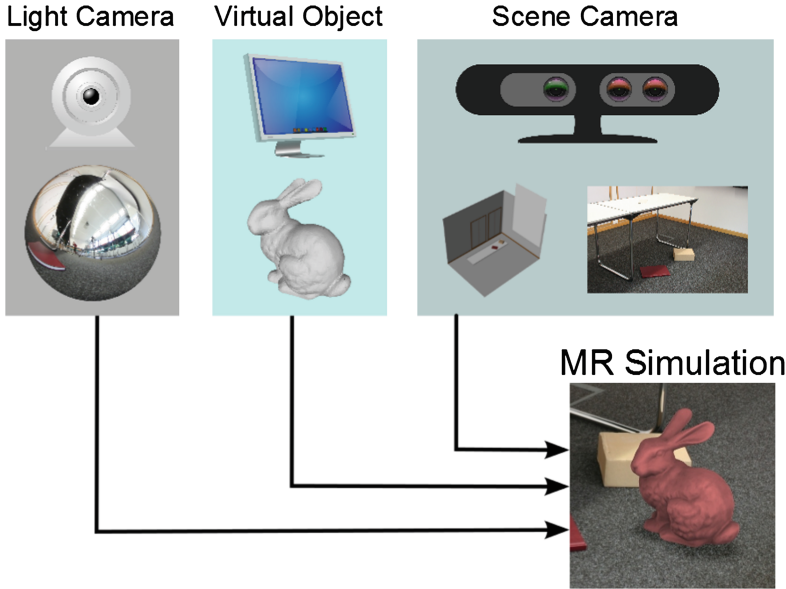
Because the delta volume is calculated by subtracting two volumes, we chose the same region for both volumes. As the region we used the double sized bounding box of the virtual object.

### 3.5 Shadows

The Voxel Cone Tracing is also used to create soft shadows on both, the virtual object and the scene. For the shadow we cast an additional cone from the fragment position to the light source to test for the occlusion in that cone. The nearer the occluding object is to the light source the smoother the shadow. This is a natural effect of the Voxel Cone Tracing as the cone will be wider and the traced occlusion value blurred.

## 4 Results and Discussion

We have implemented the system described in Sect. 3 using Direct3D 11 on an Intel i7-3770K and a NVIDIA GTX 660. A Microsoft Kinect camera was used to capture the background image, and an UEye UI-2230-C camera with a fish-eye lens to capture surrounding real light. We used ARToolKit [23] for geometric registration through marker based tracking. A manually pre-reconstructed scene with known materials has been used.

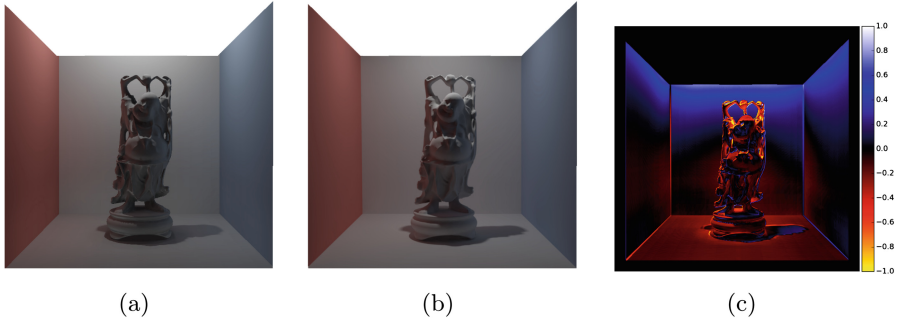


**Fig. 2.** Architecture of our prototype. The system is fed by three inputs: A virtual object, a *Light Camera* which records real illumination from the environment and a *Kinect camera* which captures the real scene from a viewpoint.

Our system architecture is depicted in Fig. 2. After capturing a frame of real light with the *Light Camera* and one frame from the viewpoint of the *Scene camera*, a buffer of the illumination, a model of the virtual object and a model of the real scene are fed to a simulation which outputs the final augmented image.

In Fig. 3 we compare our method to a ground truth result rendered with the Mitsuba path tracer [24]. An already lit virtual scene (the Cornell Box), which was rendered with regular Voxel Cone Tracing, is augmented by a Stanford Buddha with a diffuse lime-stone material. DGI captures diffuse bounces from the surrounding scene and properly blocks direct and indirect bounces that have been present in the unaugmented image.

As with most clustering techniques our illumination suffers from minor bleeding artifacts. Under certain circumstances light aliasing effects can be observed,



**Fig. 3.** A groundtruth comparison. (a) An already lit virtual scene is augmented by a Stanford Buddha using Delta Global Illumination. (b) The same scene has been rendered with the open source Mitsuba path tracer. (c) 2X squared difference ((b) - (a)).

depending on the voxel grid resolution. Figure 3(c) suggests that DGI subtracts slightly too much energy in all lit areas, while the reverse is true for shadowed parts of the scene. This error is due to a general overestimation and aliasing present in voxelized structures. In our diffuse case this error however spreads out evenly and is therefore only noticeable in direct comparison with a ground truth result.



**Fig. 4.** Results. (a) Direct shadow vs reflected shadow. (b) The reflection of the virtual dragon is successfully modeled with DGI.

In some scenes additional artifacts can occur with the shadows, especially if shadows are visible in the specular reflections. The shadows created with the Voxel Cone Tracing are very smooth, the shadows seen in the reflection on the other hand are very sharp and blocky. That is because the latter ones are created



by reading from the voxel grid with the very sharp specular cone and thus are not blurred with the mipmapping. An example for that can be seen in Fig. 4(a).

In Fig. 4(b) a result is shown. An XYZRGB dragon is reflected on a mirroring iPad surface while casting a shadow onto it. Glossy reflections are handled well with VCT, however highly specular surfaces easily reveal the voxelized nature of the reflection. This problem can be circumvented with either filtering mechanisms or higher voxel resolution, albeit at much higher memory cost.

The average computational cost for the algorithm are shown in Table 1. Noteworthy is the long time for the specular cone tracing. That time however can be drastically reduced by a wider cone aperture angle and therefore more blurred highlights.

**Table 1.** This table shows the duration of the single pipeline steps of the DGI. Note the high value for the specular cone.

| Task                  | Average duration (in ms) |
|-----------------------|--------------------------|
| Voxelization          | 5                        |
| Reflective Shadow Map | 0.2                      |
| Light Inject          | 3                        |
| Mipmap generation     | 8                        |
| Voxel Cone Tracing    |                          |
| Diffuse               | 6                        |
| Specular              | 64                       |
| Shadow                | 4                        |

## 5 Conclusion

We have presented Delta Global Illumination, a combination of Differential Rendering and Voxel Cone Tracing able to shade and relight Mixed Reality scenes which can simulate diffuse and glossy bounces of real geometry on virtual surfaces and vice versa.

Future work on this topic might include Sparse Voxel Octrees, which contain only the voxels that are actually filled with geometry. The advantage is a possibly much lower memory usage which could result in a higher resolution of the voxel grid. The drawback is that reading from that structure comes with a higher computational cost. Furthermore, the mipmap levels have to be calculated manually, which is presumably slower than the native mipmapping functions.

## References

1. Ettl, A.S., Kuijper, A.: Text and image area classification in mobile scanned digitized documents. *Int. J. Appl. Pattern Recogn.* **1**(2), 173–198 (2014)

2. Engelke, T., Becker, M., Wuest, H., Keil, J., Kuijper, A.: MobileAR browser - a generic architecture for rapid AR-multi-level development. *Expert Syst. Appl.* **40**(7), 2704–2714 (2013)
3. Wientapper, F., Wuest, H., Kuijper, A.: Reconstruction and accurate alignment of feature maps for augmented reality. In: *3DIMPVT 2011: The First Joint 3DIM/3DPVT Conference* (Hangzhou, China, May 16–19, 2011), pp. 140–147. IEEE (2011)
4. Wientapper, F., Wuest, H., Kuijper, A.: Composing the feature map retrieval process for robust and ready-to-use monocular tracking. *Comput. Graph.* **35**(4), 778–788 (2011)
5. Schwenk, K., Voss, G., Behr, J., Jung, Y., Limper, M., Herzig, P., Kuijper, A.: Extending a distributed virtual reality system with exchangeable rendering back-ends: techniques, applications, experiences. *Vis. Comput.* **29**(10), 1039–1049 (2013)
6. Olbrich, M., Graf, H., Kahn, S., Engelke, T., Keil, J., Riess, P., Webel, S., Bockholt, U., Picinbono, G.: Augmented reality supporting user-centric building information management. *Vis. Comput.* **29**(10), 1093–1105 (2013)
7. Debevec, P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1998*, pp. 189–198 (1998)
8. Crassin, C., Neyret, F., Sainz, M., Green, S., Eisemann, E.: Interactive indirect illumination using voxel cone tracing. *Comput. Graph. Forum (Proceedings of Pacific Graphics 2011)* **30**(7), 1921–1930 (2011)
9. de Sobier, F., Saito, H.: Stereoscopic augmented reality with pseudo-realistic global illumination effects. In: *Proceedings of the SPIE 9011, Stereoscopic Displays and Applications XXV*, pp. 90111W–90111W-14 (2014)
10. Viriyothai, K., Debevec, P.: Variance minimization light probe sampling. In: *SIGGRAPH 2009: Posters. SIGGRAPH 2009*, pp. 92:1–92:1 (2009)
11. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., Fitzgibbon, A.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST 2011*, pp. 559–568 (2011)
12. Chatterjee, A., Jain, S., Govindu, V.M.: A pipeline for building 3d models using depth cameras. In: *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP 2012*, pp. 38:1–38:8 (2012)
13. Karsch, K., Sunkavalli, K., Hadap, S., Carr, N., Jin, H., Fonte, R., Sittig, M., Forsyth, D.: Automatic scene inference for 3d object compositing. *ACM Trans. Graph.* **33**(3), 32:1–32:15 (2014)
14. Keller, A.: Instant radiosity. In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH 1997*, pp. 49–56 (1997)
15. Dachsbacher, C., Stamminger, M.: Reflective shadow maps. In: *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games, I3D 2005*, pp. 203–231(2005)
16. Ritschel, T., Dachsbacher, C., Grosch, T., Kautz, J.: The state of the art in interactive global illumination. *Comput. Graph. Forum* **31**(1), 160–188 (2012)
17. Knecht, M., Traxler, C., Mattausch, O., Purgathofer, W., Wimmer, M.: Differential instant radiosity for mixed reality. In: *Proceedings of the 2010 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2010*, pp. 99–107 (2010)

18. Lensing, P., Broll, W.: Instant indirect illumination for dynamic mixed reality scenes. In: Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2012, pp. 109–118 (2012)
19. Franke, T.A., Jung, Y.: Real-time mixed reality with gpu techniques. In: GRAPP 2008: Proceedings of the Third International Conference on Computer Vision Theory and Applications, pp. 249–252 (2008)
20. Franke, T.A.: Delta voxel cone tracing. In: IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2014, pp. 39–44 (2014)
21. Kaplanyan, A., Dachsbacher, C.: Cascaded light propagation volumes for real-time indirect illumination. In: Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D 2010, pp. 99–107 (2010)
22. Franke, T.A.: Delta light propagation volumes for mixed reality. In: 2013 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2013, pp. 125–132 (2013)
23. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, IWAR 1999, pp. 85–94 (1999)
24. Jakob, W.: Mitsuba renderer (2010). <http://www.mitsuba-renderer.org>