Differential Progressive Path Tracing for High-Quality Previsualization and Relighting in Augmented Reality

Peter Kán* and Hannes Kaufmann**

Institute of Software Technology and Interactive Systems,
Vienna University of Technology, Vienna, Austria

Abstract. In this paper we present a novel method for real-time high quality previsualization and cinematic relighting. The physically based Path Tracing algorithm is used within an Augmented Reality setup to preview high-quality light transport. A novel differential version of progressive path tracing is proposed, which calculates two global light transport solutions that are required for differential rendering. A real-time previsualization framework is presented, which renders the solution with a low number of samples during interaction and allows for progressive quality improvement. If a user requests the high-quality solution of a certain view, the tracking is stopped and the algorithm progressively converges to an accurate solution. The problem of rendering complex light paths is solved by using photon mapping. Specular global illumination effects like caustics can easily be rendered. Our framework utilizes the massive parallel power of modern GPUs to achieve fast rendering with complex global illumination, a depth of field effect, and antialiasing.

1 Introduction

Inserting synthetic objects into real videos is required by many real-time applications in computer graphics. Accurate offline algorithms which are capable of producing images indistinguishable from reality were developed. These methods are based on mathematical models describing light transport. Computationally expensive calculations are required to produce a full solution of global light transport involving multidimensional integration which is described by the rendering equation [1]. Therefore physically based algorithms have not been suitable for real-time applications. Applications including cinematic relighting, movie previsualization, and others can benefit from real-time light transport computation.

Direct illumination is traditionally used for the real-time preview of mixed virtual and real scenes. However, the reflected indirect light component is missing in the rendering. We propose a rendering framework using a physically based algorithm to render the composited video in preview quality during interaction. In addition it supports progressive refinement to converge to the full solution.

* e-mail:peterkan@peterkan.com
** e-mail:kaufmann@ims.tuwien.ac.at

Modern GPUs are employed to increase the speed of the path tracing algorithm. We use an Augmented Reality (AR) scenario to allow users to interact with virtual objects inserted into the real world. This scenario can be especially useful during movie production where virtual and real content is mixed. A novel one-pass differential progressive path tracing algorithm is introduced which quickly calculates two illumination solutions needed for compositing. Our framework operates in two main modes allowing interaction and high-quality convergence: An interactive preview mode and a progressive refinement mode. The problem of noise in the interactive preview mode is solved by allowing users to increase the quality of the result by increasing the sampling rate. Users can switch to the progressive refinement mode any time to see the full quality solution (Figure 1).

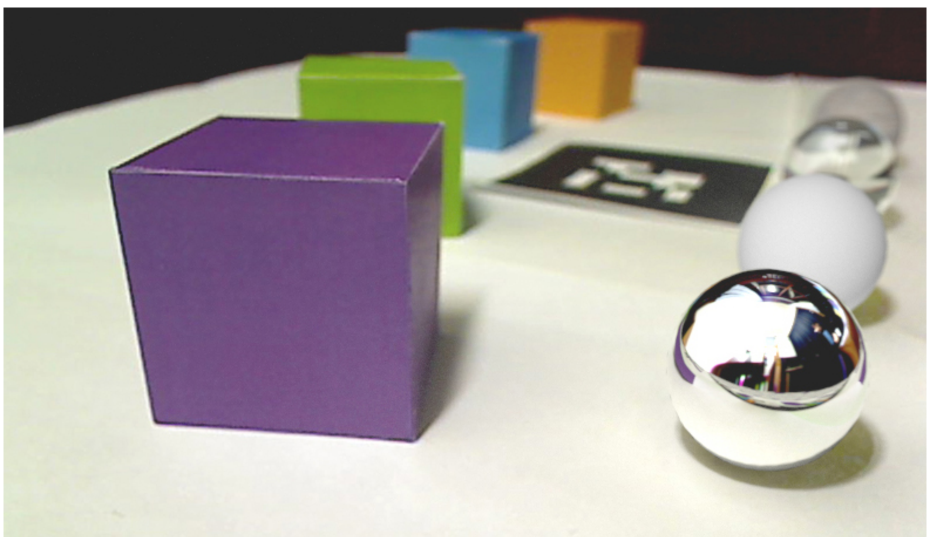

Fig. 1. Converged AR image in progressive refinement mode. Cubes on the left side of the table are real and spheres on right are virtual. The converged image was progressively rendered within 1 minute. Note the similar depth of field effect on real cubes and virtual spheres. The real environment is correctly reflected in the virtual metal ball.

Rendering synthetic objects into a real scene requires the estimation of real lighting. For this purpose we use a camera with a fish eye lens to capture the environment illumination. Two lighting algorithms are available in our framework: (1) Light source estimation by processing an environment map or (2) image based lighting, where the whole captured environment map is used to light the scene. In our rendering framework a physically based camera model with finite sized aperture is used, which enables the simulation of a high-quality depth of field effect (Figure 1). Difficult light paths needed to generate caustics are hard or in some cases impossible to simulate by path tracing. We overcome this problem by using Photon Mapping to handle these light paths separately (Figure 2). Thus our framework is capable of simulating complex global lighting between real and virtual scenes. Our framework naturally supports reflection and refraction on specular surfaces. Moreover antialiasing can be enabled by supersampling an area of pixels and it is inherently supported in progressive refinement mode. Artificial lights can be added to the rendering to support artistic lighting situations in the process of movie production. If a high quality result is needed, the captured video, lighting and camera position can be recorded during interaction. Afterwards the scene can be rendered in high-quality in post-processing to create the full movie quality.

Our main contributions of this paper are the novel differential progressive path tracing algorithm and the overall framework for previsualization and cinematic relighting. Differential progressive path tracing utilizes the massive parallel power of GPUs and produces a real-time preview of global light transport. It progressively renders a fully converged image within a short time. Our interactive framework uses this rendering algorithm together with photon mapping to simulate the full range of Global Illumination (GI) effects.

2 Related Work

The simulation of global illumination in AR is required for proper lighting of virtual objects and for achieving visual coherence between virtual and real scenes. Several methods were presented in previous research to solve this problem and to calculate the interreflections between real and virtual worlds. In this section we refer to recent research about global illumination in AR and we mention the approaches for previsualization and relighting.

2.1 Global Illumination in Augmented Reality

A pioneering work in light transport for AR was presented by Fournier [2]. It was later extended by Debevec [3] using an image based lighting approach to simulate the natural appearance of virtual objects. Caustics and accurate refractions for AR were first presented in differential photon mapping [4]. Recently high-quality augmentations of photographs with virtual objects were presented [5]. This work is similar to ours in the sense that it calculates the physically based lighting solution and it uses a user driven approach. All mentioned approaches require offline calculation. The advantage of our work is that we present a real-time approach allowing interaction of a user with the scene and we include the automatic light source estimation.

Recently rasterization based solutions for light transport in AR were presented. These solutions can achieve real-time performance with the cost of an approximation error. The work of Knecht et al. demonstrates diffuse global illumination in AR by using Differential Instant Radiosity [6]. This approach achieves real-time performance, but is limited to diffuse global illumination. Recently Lensing and Broll [7] proposed a method for calculating fast indirect illumination in mixed scenes with automatic 3D scene reconstruction. The disadvantages of both approaches are that they introduce approximation errors and are limited to diffuse global illumination. Our approach can calculate the full range of global illumination effects. In our solution the error is introduced in the form of noise, which can be reduced by increasing the sampling rate. After request the solution quickly converges to the noise-free image. A real environment map for image-based lighting of virtual objects was used in [8, 9]. The disadvantage of these techniques is that they ignore visibility in the irradiance calculation and therefore introduce an approximation error. In our solution we calculate interreflections between virtual and real worlds in both ways in a physically based fashion. We solve the visibility problem accurately by ray-tracing. Near-field illumination in AR was correctly simulated by Grosch et al. [10]. The disadvantage of their method is that a vast precomputation of irradiance is required and big objects have to be static to not change the irradiance. The advantage of our method is that no precomputation is required and we support fully dynamic geometry, materials and lighting.

2.2 Previsualization and Cinematic Relighting

Real world videos mixed with computer generated content are often used in movie production. Previsualization solutions are used on set while shooting the film to see a real-time preview of the final compositing of virtual and real worlds. A previsualization system for filmmaking using MR was proposed by Ikeda et al. [11] and Northam et al. [12]. The authors used rasterization-based rendering and did not calculate any light transport between real and virtual worlds. A good overview of previsualization techniques can be found in [13].

Systems for cinematic relighting handle a huge amount of geometry and produce an image with global light transport. A ray-traced occlusion caching of massive scenes for cinematic relighting was used by Pantaleoni et al. [14]. Sparse directional occlusion caches were precomputed in their system to accelerate a lighting pipeline working in the spherical harmonics domain. A radiosity caching algorithm to preview the final quality in movies was proposed by Christensen et al. [15]. Authors used three resolutions of radiosity and the appropriate resolution was chosen depending on the ray differentials during rendering. The mentioned approaches for cinematic relighting focus on rendering virtual content. None of them addresses the combination of real and virtual objects. Our solution can be used both for previsualization and relighting in movie production.

3 Differential Progressive Path Tracing

The core of our framework is the differential progressive path tracing algorithm running on the GPU. This algorithm uses Monte Carlo integration in ray-tracing to evaluate the global light transport in a mixed reality scene. Two solutions of light transport are needed for differential rendering to composite the final image. The first solution is the light transport in a real scene only and the second global illumination solution is within a mixed scene taking into account the geometry of both real and virtual objects. These GI results are composited with the image taken from the camera using the following equation [3, 5]:

$$L_{of} = M \odot L_{om} + (1 - M) \odot (L_{oc} + L_{om} - L_{or}) \tag{1}$$

L_{of} denotes the radiance of the final composite image for one pixel. This radiance is tonemapped to fit the low dynamic range (LDR) displaying capabilities of current displays. L_{om} is the mixed radiance result where both real and virtual objects are used. L_{or} is the lighting solution that only uses real objects. The term M denotes the mask, which defines the amount of blending between virtual objects and the real world. The differential radiance $L_{om} - L_{or}$ is added to the radiance obtained from the camera image L_{oc}. This difference presents the changes in lighting caused by adding virtual objects. Inverse tonemapping is applied to the captured LDR camera image to obtain the high dynamic range (HDR) value L_{oc}.

We use a one pass algorithm to calculate both mixed and real radiance together. Four ray types are used to enable the calculation of two rendering solutions: **a mixed radiance ray, a real radiance ray, a mixed shadow ray, and a real shadow ray**. The mixed radiance ray returns both mixed and real radiances while the real radiance ray calculates only real radiance. The mixed radiance ray type is always used in primary rays shot from the camera, because both mixed and real radiances have to be evaluated. The ray type can change if the ray intersects geometry. Our algorithm is described in detail in [16].

To calculate the light exiting from a scene point x in a direction ω_o towards the camera, the integration of incoming light at this point is necessary. The integral equation can be written as [1]:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_{\Omega} f_r(x, \omega_o, \omega_i) L_i(x, \omega_i) \mid n \cdot \omega_i \mid \, d\omega_i \tag{2}$$

where L_o is the outgoing radiance coming from point x to direction ω_o, L_e is the radiance emitted from point x towards direction ω_o and L_i is radiance of incident light incoming from direction $-\omega_i$ to point x, while the incoming light radiance is integrated on the hemisphere Ω. f_r denotes the BRDF function of the surface at position x, with surface normal n. Monte Carlo integration can be used to solve the recurrent rendering equation. The recursive path tracing algorithm [1] is used to perform this numerical integration. In original differential rendering the integral in Equation 2 has to be evaluated two times to be used in the compositing equation (Equation 1). Our one pass differential rendering algorithm can produce both solutions together. The algorithm starts by shooting the mixed radiance rays from the camera towards the scene. The numerical integration is performed on the hitpoints of primary camera rays with geometry to evaluate the contribution of light reflected from a surface point. The random light paths are sampled from this point and the visibility of a light source is tested in every hitpoint to calculate the light contribution. Mixed radiance rays will evaluate both mixed and real radiances which are needed for the differential rendering equation. The user can set up the number of samples taken during interaction to control the performance and quality of the live preview. If a user requests the full quality image, the progressive algorithm starts to run and all samples are averaged to obtain the final value.

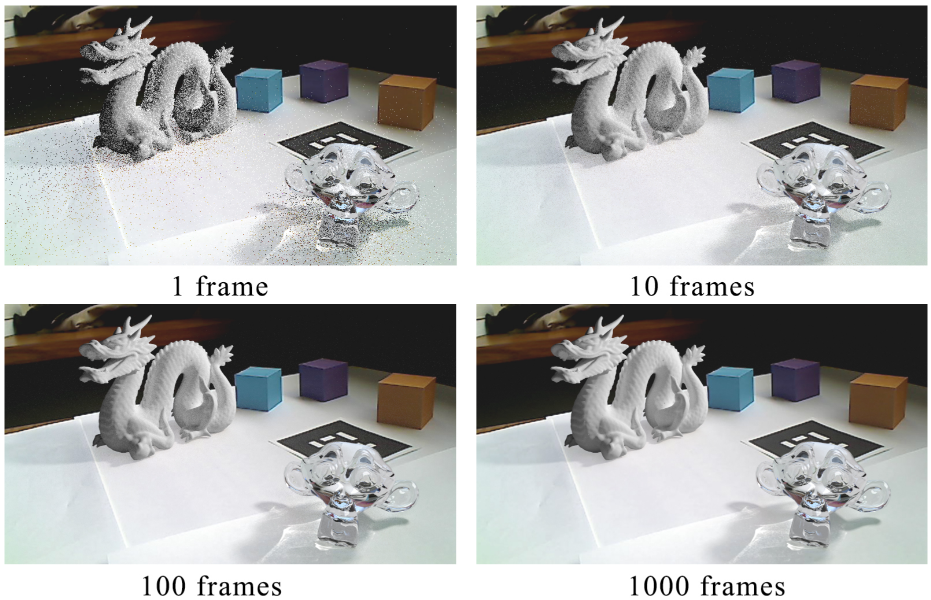

Fig. 2. Rendering by differential progressive path tracing. The scene contains a virtual monkey and a virtual dragon (27K triangles). The solution after 1, 10, 100 and 1000 frames of refinement is shown. Caustics are simulated by photon mapping.

When the user requests to see the converged AR image, the system stops tracking and camera capturing and progressively converges to a high-quality image without noise. The last camera image, pose and lighting is used for progressive rendering. Newly calculated frames are accumulated to the accumulation buffer and the average solution of all samples is progressively calculated. The new calculated radiances are blended with the accumulated values:

$$L_{acc} = \frac{(n-1)L_{acc} + L_{of}}{n} \tag{3}$$

where L_{acc} stands for the accumulated value. This value is calculated as an average of all samples of previous frames. L_{of} is the result of the compositing equation of the differential rendering algorithm (Equation 1) and n is the number of frames in progressive rendering mode.

The light transport calculation handles all radiance values as float numbers in HDR. The tonemapping operator is applied to the accumulated value to convert HDR to LDR values for displaying the composited frame on the screen. We use an inverse tonemapping operator to convert LDR camera image into HDR radiance [17]. Differential progressive path tracing is implemented using the Optix [18] ray-tracing engine. The results of progressive rendering in AR can be seen in figure 2.

4 Previsualization Framework

We created a novel previsualization and relighting framework capable of delivering a high-quality rendering result by using the differential progressive path tracing algorithm. The presented framework can operate in two different modes:

– **Interactive preview-mode:** The 3D pose of the camera is tracked and the user can interact with the scene in real-time. The quality of output can be controlled by the number of samples per pixel. The preview mode suffers from noise caused by the high variance in Monte Carlo integration.
– **Progressive refinement mode:** This mode is used when a high-quality converged image is required. The interaction is paused and the last captured camera image is used for compositing. The synthetic image is progressively refined as more samples are used. In our experiments a fully converged image could be obtained within 2 minutes in average.

The following stages are performed per frame if the interactive preview mode is enabled: The environment lighting is captured by a camera with a fish eye lens and the positions of light sources are estimated if light source estimation is enabled. The observer camera image is taken and a 3D pose is calculated by the tracking system. Data from previous stages are sent to the ray-tracing engine running on the GPU. Mixed radiance rays are sent from the camera position towards the scene to evaluate both mixed and real radiances and to composite virtual objects with the real camera image. Direct illumination is evaluated at hit points of rays with the geometry. Indirect reflected light is estimated by shooting rays in random directions within the hemisphere above the hit point and using Monte Carlo numerical integration. A low number of ray samples (1-50) is used during interaction which leads to a noisy preview image. Temporal coherence between successive frames is achieved by using the same random parameters across each frame. A video can be recorded during the interactive session and the full quality movie can be generated in postprocessing.

When the user switches to the progressive refinement mode, the tracking of the camera, light source estimation and camera capturing is paused. The ray tracing engine uses the data from the last interaction frame and starts to accumulate the calculated light values. The progressive refinement is displayed to the user. The user can either wait until the image is fully converged to see the final result or can interrupt the progressive refinement to continue in interactive mode.

4.1 Estimation of Illumination

The estimation of real light is an important step to achieve visual coherence between lighting of virtual and real objects. Two different lighting methods are available in our framework. The first is Image Based Lighting [3]. There the whole environment map that is captured by a camera with a fish eye lens is used as a source of light. Inverse tonemapping is applied to the environment image in every frame to obtain HDR radiance values. This radiance is later accessed in the ray miss program when a ray misses any geometry. This method can simulate natural lighting but requires more samples to calculate the converged solution.

The second option is light source reconstruction using image processing. Our framework utilizes the method proposed in [19] which uses connected component analysis to extract the positions of light sources. We run environment image capturing and light source estimation asynchronously in a separate thread.

4.2 Interaction

During the interactive preview mode a user is able to interact with virtual objects to see the rendering result in preview quality. The user can switch between interactive preview mode and progressive refinement mode by pressing a button. We use marker-based visual tracking to estimate the 3D pose of the camera although any other tracking system could be used. Our rendering framework supports dynamic geometry, materials, lighting and camera. Therefore interaction between virtual and real worlds can be achieved in real-time. Our current implementation does not use the movement of real objects, because predefined phantom objects are used. The system can be extended by automatic scene reconstruction.

4.3 Caustics

It is difficult to correctly simulate some light paths by path tracing. Caustics are especially problematic because the probability that the specular reflection will hit the light source is low in case of area light sources and it is zero in case of point light sources. Therefore we employ Photon Mapping [20] to simulate caustics. We use a GPU implementation of photon mapping using the OptiX ray-tracing engine [18] in order to achieve interactive frame rates while keeping quality of the created caustics high. We extended the interactive photon mapping implementation from [19] to work in path tracing. Caustic light is integrated in path tracing by using kernel density estimation at each ray hit point.

5 Results

Rendering results with our framework can be seen in figure 3. Each row shows a different scene, frame rates are depicted below the pictures. The first column shows interactive rendering with 1 ray sample shot per pixel. The second column contains interactive results with 9 rays per pixel and the third column shows the

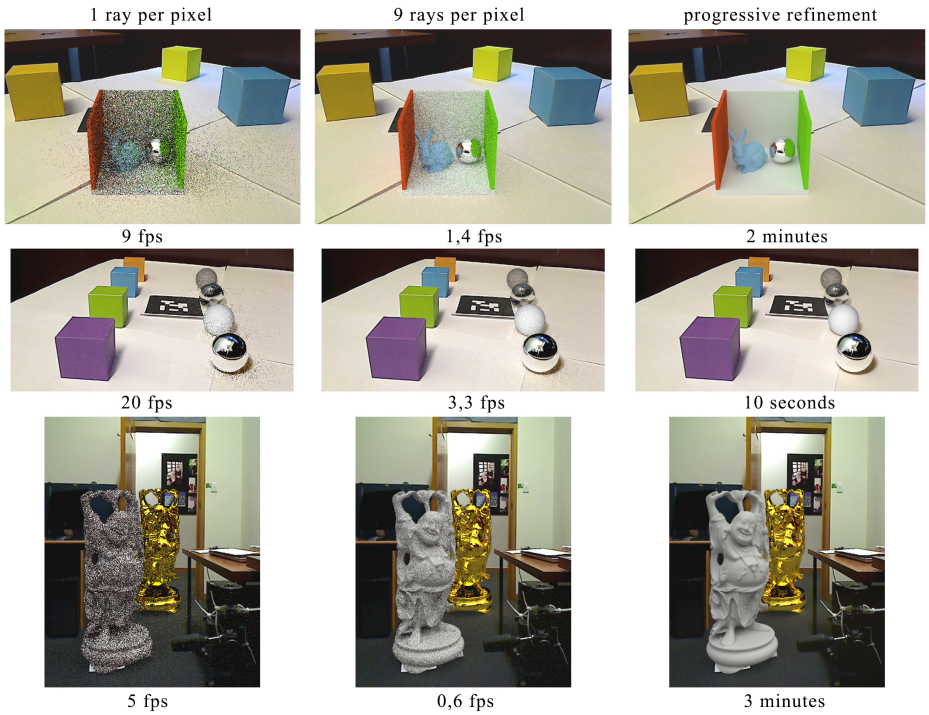

Fig. 3. Rendering results with differential progressive path tracing in our framework.

converged rendering solution in the progressive refinement mode. The scene in the first row contains three real paper cubes on a table and a virtual box with a metallic ball and a bunny (17K triangles). An image based lighting approach was used to lit the scene. The scene in the second row contains real cubes on the left and virtual spheres on the right side (4K triangles). Light source estimation by image processing was used here. Note the very fast convergence when simplified lighting conditions are used (10 s). The third row shows a real scene with inserted Buddha statues (581K triangles). The image based lighting approach was used. Interactivity can be still achieved despite the very high complexity of this scene. The results show the interaction of light between virtual and real objects. We analyzed the dependence of rendering speed on output resolution which is depicted in figure 4. We can see that our path tracing based rendering is interactive even for full HD resolution. It can be seen that our system is well scalable both in terms of triangle count and output resolution. All tests were performed on a laptop with quad-core CPU and a GeForce GTX 680M graphics card. A resolution of 800x600 was used in all evaluations except figure 4.

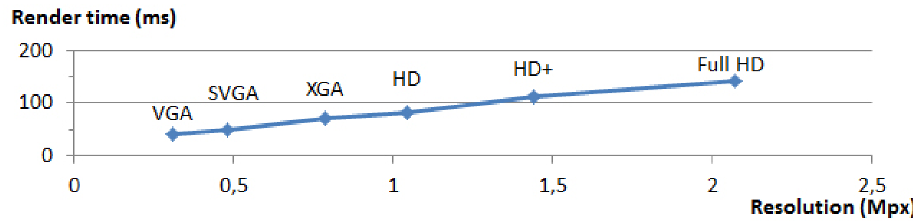

Fig. 4. Dependence of rendering time on resolution. A 3D model consisting of 20K triangles was used. 1 ray per pixel was shot to render the scene.

6 Conclusion and Future Work

In this paper we propose a novel progressive rendering algorithm for augmented reality based on path tracing and differential rendering. A framework capable of simulating complex global light transport between virtual and real worlds is presented. Our framework can possibly be used in movie previsualization, cinematic relighting or other fields.

In the future we plan to extend the framework with automatic real-time 3D scene reconstruction which will allow physical interaction between real and virtual worlds. Moreover we plan to improve our path tracing algorithm with advanced filtering approaches to reduce noise caused by high variance [21]. We believe that due to the improvements of graphics hardware, ray-tracing based rendering will become the standard rendering method for AR applications.

7 Acknowledgements

3D models of Bunny, Dragon and Happy Buddha are courtesy of Stanford Computer Graphics Laboratory. Monkey model is courtesy of Blender. Peter Kán is financially supported by the Vienna PhD School of Informatics.

References

1. Kajiya, J.T.: The rendering equation. In: Proceedings of the 13th annual conference on Computer graphics and interactive techniques. SIGGRAPH '86, New York, NY, USA, ACM (1986) 143–150
2. Fournier, A., Gunawan, A.S., Romanzin, C.: Common illumination between real and computer generated scenes. In: Proceedings of Graphics Interface '93, Toronto, ON, Canada (1993) 254–262
3. 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 '98, New York, ACM (1998) 189–198
4. Grosch, T.: Differential photon mapping: Consistent augmentation of photographs with correction of all light paths. In: Eurographics Short Papers, Dublin. (2005)

5. Karsch, K., Hedau, V., Forsyth, D., Hoiem, D.: Rendering synthetic objects into legacy photographs. In: Proceedings of the 2011 SIGGRAPH Asia Conference. SA '11, New York, NY, USA, ACM (2011) 157:1–157:12
6. 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. (2010) 99–107
7. Lensing, P., Broll, W.: Instant indirect illumination for dynamic mixed reality scenes. In: Proceedings of the 11th IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2012). ISMAR2012, IEEE (2012) 109–118
8. Pessoa, S., Moura, G., Lima, J., Teichrieb, V., Kelner, J.: Photorealistic rendering for augmented reality: A global illumination and brdf solution. In: Virtual Reality Conference (VR), 2010 IEEE. (2010) 3 –10
9. Agusanto, K., Li, L., Chuangui, Z., Sing, N.W.: Photorealistic rendering for augmented reality using environment illumination. In: Proceedings of the International Symposium on Mixed and Augmented Reality. ISMAR '03, Washington, DC, USA, IEEE Computer Society (2003) 208–218
10. Grosch, T., Eble, T., Mueller, S.: Consistent interactive augmentation of live camera images with correct near-field illumination. In: Proceedings of the 2007 ACM symposium on Virtual reality software and technology. VRST '07, New York, NY, USA, ACM (2007) 125–132
11. Ikeda, S., Taketomi, T., Okumura, B., Sato, T., Kanbara, M., Yokoya, N., Chihara, K.: Real-time outdoor pre-visualization method for videographers. In: Multimedia and Expo, 2008 IEEE International Conference on. (2008) 949 –952
12. Northam, L., Istead, J., Kaplan, C.S.: A collaborative real time previsualization tool for video games and film. In: ACM SIGGRAPH 2012 Posters. SIGGRAPH '12, New York, NY, USA, ACM (2012) 121:1–121:1
13. Wong, H.H.: Previsualization: assisting filmmakers in realizing their vision. In: SIGGRAPH Asia 2012 Courses, New York, NY, USA, ACM (2012) 9:1–9:20
14. Pantaleoni, J., Fascione, L., Hill, M., Aila, T.: Pantaray: fast ray-traced occlusion caching of massive scenes. In: ACM SIGGRAPH 2010 papers. SIGGRAPH '10, New York, NY, USA, ACM (2010) 37:1–37:10
15. Christensen, P.H., Harker, G., Shade, J., Schubert, B., Batali, D.: Multiresolution radiosity caching for global illumination in movies. In: ACM SIGGRAPH 2012 Talks. SIGGRAPH '12, New York, NY, USA, ACM (2012) 47:1–47:1
16. Kán, P., Kaufmann, H.: Physically-based depth of field in augmented reality. In: EG 2012, Cagliari, Italy, Eurographics Association (2012) 89–92
17. Banterle, F., Ledda, P., Debattista, K., Chalmers, A.: Inverse tone mapping. In: Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia. GRAPHITE '06, New York, NY, USA, ACM (2006) 349–356
18. Parker, S.G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., Stich, M.: Optix: a general purpose ray tracing engine. ACM Trans. Graph. **29** (2010)
19. Kán, P., Kaufmann, H.: High-quality reflections, refractions, and caustics in augmented reality and their contribution to visual coherence. In: Proceedings of International Symposium on Mixed and Augmented Reality, ACM Press (2012) 99–108
20. Jensen, H.: Realistic Image Synthesis Using Photon Mapping. Ak Peters Series. A K Peters, Limited (2009)
21. Gastal, E.S.L., Oliveira, M.M.: Adaptive manifolds for real-time high-dimensional filtering. ACM TOG **31** (2012) 33:1–33:13 Proceedings of SIGGRAPH 2012.