Physically-Based Depth of Field in Augmented Reality

P. Kán[1] and H. Kaufmann[1]

[1]Institute of Software Technology and Interactive Systems, Vienna University of Technology, Vienna, Austria

Abstract

We present a novel method for rendering and compositing video in augmented reality. We focus on calculating the physically correct result of the depth of field caused by a lens with finite sized aperture. In order to correctly simulate light transport, ray-tracing is used and in a single pass combined with differential rendering to compose the final augmented video. The image is fully rendered on GPUs, therefore an augmented video can be produced at interactive frame rates in high quality. Our method runs on the fly, no video postprocessing is needed. In addition we evaluated the user experiences with our rendering system with the hypothesis that a depth of field effect in augmented reality increases the realistic look of composited video. Results with 30 users show that 90% perceive videos with a depth of field considerably more realistic.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing I.3.8 [Computer Graphics]: Applications— H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1. Introduction

According to Azuma's [Azu97] definition augmented reality (AR) is a combination of reality and virtuality. Real and virtual objects must be registered in 3D to accurately position objects in AR environments. In addition AR has to be interactive in real-time. Real-time performance has hindered the application of high qualitative rendering in the early years. Today different rendering approaches exist to display AR scenes in high quality. Our work focuses on the Depth of Field (DoF) effect to increase the visual realism of augmented video. If the DoF is ignored, very close or distant virtual objects which are out of the camera's focus still appear sharp and focused in the image. They look unnatural and can degrade the visual coherence of the video. Applications can be found in all areas where high-quality AR videos are required, as in medicine, marketing, education and in the entertainment industry. During movie production the immediate view of the composited video is of high interest.

DoF is a natural feature of many optics systems such as the human eye or camera lenses. We can define the depth of field as a range of distances near the point of focus where the user perceives the image acceptably sharp [HLCC08]. To produce this effect in computer graphics, rendered objects have to be blurred according to their distance from the plane of focus. The blurring can be simulated in different ways. Usually this effect is calculated in image space by blurring the rendered image according to the depth values of virtual objects. However this kind of DoF blur calculation is not physically correct and can introduce various visual artefacts. To overcome this problem we use ray-tracing with a physically-based lens model. We solve the ray-scene intersection for rays sampled from many lens points for each pixel to obtain a correct result.

Compositing in augmented reality is often solved by Differential Rendering [Deb98]. It is a two-pass method, where real and mixed objects have to be rendered separately. We introduce a more effective one pass compositing method based on differential rendering which calculates the result directly within the ray-tracing pipeline. We improved ray-tracing to calculate both the radiances from real and mixed objects together. Our method composites the final result directly in the ray-generation program. We use the massive power of modern parallel GPUs to run our algorithm fully on GPUs with the OptiX [PBD*10] ray-tracing engine.

The advantage of our method in comparison to previous techniques is that it produces a highly qualitative, physically correct result. It can also simulate DoF blur everywhere in the scene, which is an advantage compared to [OKY06]. Moreover we can naturally produce specular reflection (Figure 1) or refraction and shadows by ray-tracing.

DOI: 10.2312/conf/EG2012/short/089-092

The main contributions of our work are:

- It is the first method for physically correct DoF in interactive AR videos.
- A novel one-pass compositing method based on differential rendering is performed directly in ray-tracing.
- User studies confirm the importance of DoF in AR.

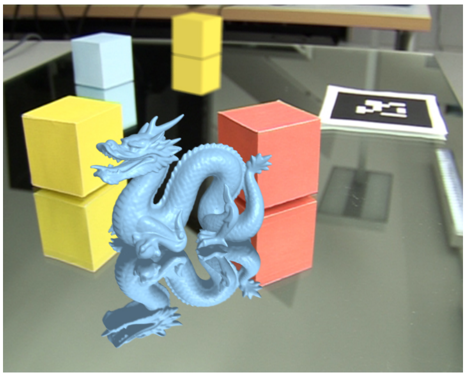

Figure 1: *Using ray-tracing in AR can naturally provide features like specular reflection on mirroring surfaces. Moreover the physically-based DoF technique can improve visual realism of the produced image. Note that there are two virtual objects in the image: The dragon and the out of focus yellow cube in background. The dragon model is courtesy of Stanford University.*

The paper is organized as follows. In section 2 we discuss related research and compare it to our method. Section 3 describes the model of optics we use. The DoF rendering and AR compositing algorithms are described. System design and implementation are discussed in section 4. Our evaluation and results are shown in section 5.

2. Related Work

Some methods were developed in the past to render a DoF effect in AR. However, techniques for physically-based DoF in interactive AR videos have not been examined before. The majority of the proposed methods blur the rendered image according to blur parameters estimated from the camera. Park et al. [PLW09] extended the ESM algorithm to handle blur in 3D object tracking. They rendered more intermediate images and combined them according to detected blur parameters. They focused mainly on motion blur. Okumura et al. [OKY06] estimated blur parameters according to a point spread function (PSF) detected on fiducial markers. They improved the tracking method to handle blurred markers and rendered the virtual image which was blurred according to the PSF. The disadvantage of their method is that they can blur virtual objects only on the marker position and they used a PSF of only elliptical shape. Our method is capable to render blurred virtual objects at arbitrary scene positions.

Intensive work for visual realism in AR were proposed by Karsch et al. [KHFH11]. However their method produces only still images and works offline. An important part of an AR video system is the compositing method. A widely used method is differential rendering described in [Deb98]. The use of ray-tracing in AR applications was examined by Scheer et al. [SAM07] who used ray-tracing in AR as a way of realistic rendering and Pomi et al. [PS05] who studied the insertion of the real video of characters into a virtual environment in TV Studio applications. A good overview of depth of field algorithms in computer graphics can be found in [BK08].

3. AR System Overview

Our method combines real-time ray-tracing with a differential rendering compositing algorithm. Predefined geometry of the real world is used to enable light interaction between real and virtual objects. The main components of our approach are described in following sections.

3.1. Depth of Field Rendering

The blur caused by a DoF effect can be observed on objects which lie outside the focal plane. A point which lies in the focal plane is projected by the lens onto the image plane as a single point. Therefore it is perfectly sharp. On the other hand a point out of the focal plane is projected onto an area on the image sensor. Therefore it will be blurred in the final image.

A lens with finite sized aperture is the natural type of optics for many lens systems. In order to follow the finite sized aperture model and to reproduce a physically correct DoF effect, the visibility of objects has to be calculated from many points on the aperture. We use an approach similar to the one described in [PH10].

Many rays have to be sampled on different aperture points to obtain the final blurry effect of out of focus objects. We use Stratified jittered sampling [Mit96] to sample both the 2D domain of aperture and the 2D domain of the pixel area. At the beginning we start with the fixed center of projection O. We first sample the pixel area at the image location where we will calculate the final color. Then we calculate the intersection point I of a ray with direction $\vec{d}$ and the focal plane by the following equation.

$$I = O + \vec{d} * d_f \tag{1}$$

Where d_f is the distance from center of projection to the focal plane. Then the new direction $\vec{d}'$ is calculated by subtracting the sampled aperture point O' from the intersection point I. The sampled aperture point is randomly sampled in the planar area around origin O.

$$O' = O + r_x \cdot \vec{X} + r_y \cdot \vec{Y} \quad (2)$$

$$\vec{d}' = I - O' \quad (3)$$

The r_x and r_y in equation 2 are two randomly sampled numbers from domain (-1,1) scaled according to the aperture size. Vectors $\vec{X}$ and $\vec{Y}$ are normalized coordinate axes of the image plane. The sampling point of the image plane remains the same because the image plane has to shift according to the origin offset. If a sufficient number of aperture samples is used we obtain a high quality DoF result. The result of our physically-based DoF technique can be seen in figure 1.

3.2. Compositing

An important part of an AR system is the compositing algorithm which mixes real and virtual images. We use differential rendering, but improve the ray-tracing algorithm to composite the final image directly in one pass. The radiances coming from real objects and mixed objects are calculated together and are stored in a per-ray data structure.

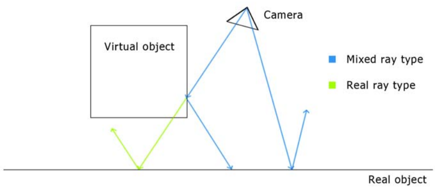

Figure 2: *Ray type can change after hitting virtual objects. Green rays are real radiance type and blue ones are mixed radiance type rays. Shadow rays are not shown here.*

We divide rays in our engine into 4 different types: mixed radiance ray, real radiance ray, mixed shadow ray, real shadow ray. Mixed ray types belong to rays which can intersect with both real and virtual objects. Real ray types take into consideration only real objects when the ray-triangle intersection is calculated. In the ray-generation phase we shoot only mixed rays. Rays can possibly change their type after the object intersection. Our per-ray data (PRD) structure contains four variables: mixed radiance, real radiance, mask, and depth. We use four different rules for ray type changing and shooting (Figure 2):

- Real object hit by a mixed ray - Real and mixed shadow rays are shot. Mixed reflected ray is shot.
- Real object hit by a real ray - Real shadow ray and a real reflected ray is shot.
- Virtual object hit by a mixed ray - Real ray continuing the primary direction, a mixed shadow ray and a mixed reflected ray are shot.
- Virtual object hit by a real ray - This isn't possible because the ray-triangle intersection routine prevents this case.

The corresponding shading results in ray-triangle intersections are stored as appropriate variables in per-ray data. The mask value in PRD is set to 1.0 when a ray hits a virtual object and to 0.0 otherwise. This value is used in the differential rendering compositing equation at the end of the ray-generation program. The compositing equation [KHFH11] is the following.

$$I_f = M \odot I_m + (1 - M) \odot (I_c + I_m - I_r) \quad (4)$$

M stands for the average of mask values of rays shot through pixels. I_m is the rendered result of both real and virtual objects (mixed radiance), while I_r is the rendering result of only real objects (real radiance). The I_c is the source image captured by the camera.

Differential rendering is a two-pass method, however our improvement allows to do it in only one ray-tracing pass. The compositing is performed directly on the GPU in the ray-generation program. We shoot two types of rays (mixed and virtual) instead of rendering two separate results. Our method is more efficient because both types of rays do not have to be shot in all cases.

4. Implementation

Ray-tracing is highly suitable for parallel execution therefore all rendering computation is performed on a modern massive parallel GPU. We use the OptiX [PBD*10] GPU-based ray-tracing engine and we implement the physically-based camera model in a ray generation program. Stratified jittered sampling needs a uniform random number generator. We use a pregenerated array of random numbers in our solution. All evaluations run on a hexa-core machine with a dual-core GPU NVIDIA GeForce GTX 590 graphics card. The Sony HVR-Z1E camcorder is used to capture video; a low aperture value is used to get a good DoF effect. For camera tracking within the scene the marker based ARToolkitPlus [WS07] was applied.

5. Evaluation and Results

We evaluated the users' perception when watching an AR video rendered with DoF in comparison to a video without DoF. Our hypothesis was that depth of field increases the realistic appearance of composited scenes. We showed two recorded videos to users - one video with, the other without a DoF effect. Users were asked to tell which video looked more realistic. We asked 30 people (computer scientists as well as people from other professions), 15 men and 15 women. 90% of them considered the video with DoF more realistic. The results show that DoF is important feature for visual realism in AR.

We measured the performance of our technique on different quality levels. The results can be seen in table 1. Our method is capable of running at interactive frame rates even at high quality settings. Lower quality previews can be rendered in real-time. The results of our method in comparison to rendering without DoF can be seen in figure 3. It is visible that rendering with DoF is more visually consistent than without it. The supplementary high quality videos were rendered with 49 rays per pixel.

rays per pixel	primary rays count	fps
1	0.4 M	59
25	10.4 M	12
49	20.3 M	7

Table 1: *Frame rates when rendering DoF by ray-tracing. The measurements were taken at a resolution of 720x576.*

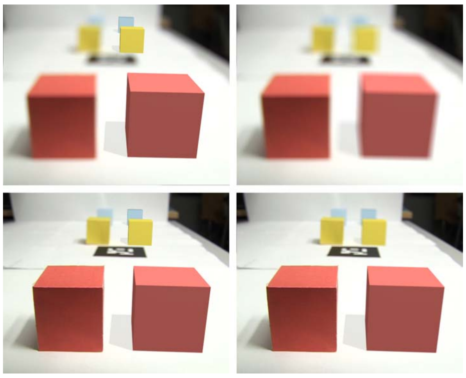

Figure 3: *Results of our technique. Left column: rendering without DoF. Right column: rendering with DoF. First row shows defocused image and second row shows image with camera focused at closest cube. Left cubes in each image are real and right cubes are virtual.*

6. Conclusion and future work

In this paper we propose the first algorithm for physically-based DoF in interactive augmented videos. We combine GPU-based ray-tracing with differential rendering to create a more effective one-pass algorithm. We discuss the capabilities of our method and show that a physically correct DoF implementation can substantially improve the perception of realism in augmented videos. Global illumination (GI) is an important step in increasing visual realism. We plan to add GI calculation to our system in the near future work. A next logical step is also to extend the system with automatic real geometry reconstruction by using a depth camera.

7. Acknowledgments

The research work of Peter Kán is sponsored by the PhD school of informatics at TU Vienna. We would like to thank Christian Schönauer and Thomas Pintaric for their help with tracking equipment. We thank the anonymous reviewers for their valuable comments and suggestions.

References

[Azu97] AZUMA R. T.: A survey of augmented reality. *Presence: Teleoperators and Virtual Environments 6*, 4 (Aug. 1997), 355–385. 1

[BK08] BARSKY B. A., KOSLOFF T. J.: Algorithms for rendering depth of field effects in computer graphics. In *Proceedings of the 12th WSEAS international conference on Computers* (Stevens Point, Wisconsin, USA, 2008), World Scientific and Engineering Academy and Society (WSEAS), pp. 999–1010. 2

[Deb98] 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* (New York, NY, USA, 1998), SIGGRAPH '98, ACM, pp. 189–198. 1, 2

[HLCC08] HILLAIRE S., LÉCUYER A., COZOT R., CASIEZ G.: Depth-of-field blur effects for first-person navigation in virtual environments. *IEEE Comput. Graph. Appl. 28* (November 2008), 47–55. 1

[KHFH11] KARSCH K., HEDAU V., FORSYTH D., HOIEM D.: Rendering synthetic objects into legacy photographs. *ACM Trans. Graph. (Proceedings of ACM SIGGRAPH ASIA) 30*, 6 (2011). 2, 3

[Mit96] MITCHELL D. P.: Consequences of stratified sampling in graphics. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 277–280. 2

[OKY06] OKUMURA B., KANBARA M., YOKOYA N.: Augmented reality based on estimation of defocusing and motion blurring from captured images. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality* (Washington, DC, USA, 2006), ISMAR '06, IEEE Computer Society, pp. 219–225. 1, 2

[PBD*10] 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* (July 2010), 66:1–66:13. 1, 3

[PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann. Elsevier Science, 2010. 2

[PLW09] PARK Y., LEPETIT V., WOO W.: Esm-blur: Handling & rendering blur in 3d tracking and augmentation. In *Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality* (Washington, DC, USA, 2009), ISMAR '09, IEEE Computer Society, pp. 163–166. 2

[PS05] POMI A., SLUSALLEK P.: Interactive Ray Tracing for Virtual TV Studio Applications. *Journal of Virtual Reality and Broadcasting 2*, 1 (Dec. 2005). 2

[SAM07] SCHEER F., ABERT O., MÜLLER S.: Towards using realistic ray tracing in augmented reality applications with natural lighting. *GI Workshop ARVR 07* (2007). 2

[WS07] WAGNER D., SCHMALSTIEG D.: *ARToolKitPlus for Pose Tracking on Mobile Devices*. Tech. rep., Institute for Computer Graphics and Vision, Graz University of Technology, Feb. 2007. 3