REAL-TIME MIXED REALITY WITH GPU TECHNIQUES

Tobias Franke and Yvonne Jung
GRIS TU-Darmstadt/ Fraunhofer IGD, Fraunhoferstr. 5, Darmstadt, Germany
tobias.franke@gris.informatik.tu-darmstadt.de, yvonne.jung@igd.fraunhofer.de

Keywords: Mixed Reality, Image Based Lighting, Shader, Real-time, Differential Rendering.

Abstract: In this paper, we propose a combination of modern GPU-based methods that are able to generate high-quality, interactive real-time rendering for augmented and mixed reality applications. We also present a new approach to estimate surface reflection functions and materials from images using genetic algorithms.

1 INTRODUCTION

The fusion of real and virtual worlds is the foundation for a range of computer graphics applications: complex augmented and mixed reality, movie effects and the ability to advertise products not yet completed, such as houses still being built or prototypes of cars in real environments. To achieve a mixed reality feeling, complex lighting interaction between real and virtual objects has to be simulated in real time. Only if the user is not able to clearly distinguish between real and virtual objects, this aim is reached.

This paper presents a "piped" combination of current methods to generate high quality images in real time with real world lighting. New GPU-based techniques allow us to limit the needed processing power, thus making the system mobile and ready for further enhancements. We also describe a new approach to derive complex materials from the original scene radiance given by HDR images.

2 PREVIOUS WORK

In (Debevec, 1998) "differential rendering" was presented in order to merge virtual objects with real scenes. By calculating the difference of two rendered images of the reconstructed virtual scene via a radiosity simulation, once with and once without the objects, the change that is introduced by placing the virtual objects into a scene is recovered. Combined with the rendered objects the difference can be added to a real image. In (Grosch, 2005) this method was extended to also handle reflective and refractive objects correctly. Illumination of the real environment is passed to the virtual object via irradiance maps from a light probe, captured as HDR images. In (Kautz et al., 2004), the authors discuss ways of filtering environment maps to create different types of irradiance maps. Spherical harmonics (SH) are used to extract the diffuse frequencies of the environment map, while hardware generated mipmaps are used to create glossy maps. A combination of the original and both filtered images yields the final reflection that is mapped onto the virtual object. In (King, 2005), the author proposes to use irradiance maps in conjunction with ambient occlusion (AO), a statistical method that determines shadows under complete diffuse lighting conditions, without considering any light sources. These values can be used to attenuate colors from the irradiance maps to simulate self-shadowing under the assumption of distant global illumination.

3 LIGHTING RECONSTRUCTION

3.1 Irradiance Mapping

The first important step to enable high quality rendering of real and virtual lights is the lighting reconstruction phase, where real world lighting is captured to transfer effects from the real scene onto a virtual model. For instance, a light-switch could be toggled in the real environment, which should have an effect on the virtual objects, otherwise it will be clear rather soon that they are just an augmentation. For static configurations, the incident radiance is captured with a light probe. Dynamic scene lighting can be captured
with a 180° fish eye lens. The acquisition should be
done in HDR, otherwise lighting mapped for different
materials will appear to have no contrast. Our simu-
lation relies on image based lighting, a method that
derives all information about the environment light-
ning from images. Usually, these images are cube- or
sphere maps and can be used to simulate highly re-
fective or mirror-like surfaces. Unlike “simple” en-
vIRONMENT MAPPING, the idea of irradiance mapping is
to use environment maps for a range of basic trans-
fer functions. By filtering this map, incident light for
glossy or diffuse surfaces can be simulated. Currently
we use the spherical harmonic basis to simulate dif-
fuse and low order glossy irradiance, because high
frequencies are captured inadequately.

3.2 Ambient Occlusion

Using precomputed radiance transfer, more complex
transfer functions can be simulated. Additionally to
the irradiance map \( L \), the surface’s transfer function
\( T \) is moved to frequency space, and by exploiting
\( \int_{\Omega} L \cdot T \approx \sum L_i \cdot T_i \) (which is easily evaluable inside
a shader), the integral of the rendering equation is
approximated while retaining high rendering speed.
The perhaps most obvious difference between local
and global illumination are shadows, especially self-
shadowing. Without, objects seem to have no detail
in structure, but with self-shadowing, tiny structures
become emphasized and add to the overall realism.
Many of those details can be recovered with ambient
occlusion. We use AO as a substitute for the trans-
fers function in PRT, because it can be calculated in
real time also for non-rigid objects. Before the irra-
diance maps are applied to the objects surface, AO is
determined through a modified version of the method
described in (Sattler et al., 2004). Real scenes con-
tain much indirect lighting, so only using direct light
sources as sampling positions is unrealistic. Hence,
random sample positions are used to determine the
ambient occlusion on the model. In table 1, we have
measured the difference between the generated AO
values of a 1000 sample reference model and the same
model with a lower sample count. One can see that
even for complex geometry, the error drops below
10% with 25 samples. The relative difference with 5-
sample-steps drops below 1% at 50 samples for most
models. For high-polygon or rigid objects, we pre-
compute and store the same information with its col-
ors. In combination, the colors from the irradiance
maps are attenuated with the ambient occlusion val-
ues on the surface. A result is shown in figure 1.

3.3 Shadows

One problem is that irradiance mapping does not ad-
dress any positional information about light sources,
whereas AO is not considering light sources at all.
Without this information, casting shadows into the
right direction will become difficult. In our simu-
lation, we first extract possible direct light sources
from images in a pre-process. While the direction can
be taken directly from the irradiance map, the posi-
tion is determined by intersecting its boundary points
with the surrounding reconstructed scene model. The
extracted positions are then used to project shad-
ows onto geometry. Visually pleasing results were
achieved with PCF and PCSS shadows, although
the latter caused some performance hits (Jung et al.,
2007). To determine the shadow’s intensity during
runtime, we used the first coefficient of the SH analy-
sis of the surrounding lighting configuration. Because
the first SH function is constant, the first coefficient
will statistically provide information about the ambi-
tent brightness in the scene. Thus, the inverse value
given that \( c_0 \) is normalized to [0, 1] can be used as
shadow intensity. The brighter the ambient lighting
is, the less intense the shadow will be and vice versa.
The same value can be used to adjust the AO values.

4 DIFFERENTIAL RENDERING

Differential rendering is a multi pass compositing
technique that is feasible for augmenting images or
videos with consistent illumination. It requires two lighting simulations, one with the real scene only and a second one with the additional virtual objects inserted. For real-time appliances the rendering should be hardware accelerated, therefore both before mentioned scenes are rendered into different textures using standard rasterization methods. Let \( L_{\text{orig}} \) be the original scene radiance given by the background image, \( L_{\text{with}} \) the rendered scene with virtual objects and \( L_{\text{without}} \) the rendering without them. Then the error in the rendered scene is \( \Delta L_{\text{err}} = L_{\text{without}} - L_{\text{orig}} \). As can be seen, the better geometry and material reconstruction are, the smaller the resulting error is. By subtracting the error from \( L_{\text{with}} \), the changes in illumination caused by inserting the virtual object then can be represented as \( L_{\text{final}} = L_{\text{orig}} + (L_{\text{with}} - L_{\text{without}}) \). Finally a window-sized, view-aligned quad is rendered with a special shader program, which combines all images according to this formula.

5 MATERIAL RECONSTRUCTION

To accurately map virtual light or shadows onto real surfaces, their properties have to be known upfront. For instance, to simulate interaction between a virtual light and a real surface it has to be clear whether or not that surface is diffuse or mirror-like. If these properties are unknown, the differential rendering will produce wrong colors for shadows and lights (or other artifacts). In our simulation, an off-line process tries to analytically estimate material properties from camera images. This process is a modified implementation of (Gibson et al., 2001). Combined with real lights in the image, placeholders for unknown lights, so called virtual lights, are adapted to match the irradiance of the surface. Diffuse materials can then be estimated iteratively with a linear equation system. As soon as non-linear components are added to the surface BRDF, other solutions have to be found. The authors proposed minimizing a cost function with non-linear optimization for all unknown variables. Instead, we used genetic algorithms as a consistent substitute for all material functions.

5.1 Genetic Algorithms

A genetic algorithm is a particular class of evolutionary algorithms that is used for global search and optimization problems. Instead of calculating in a deterministic manner a result is evolved from a population of possible solutions. The main motivation for genetic algorithms as a substitute to approximate surface reflection functions in our implementation is that they require no knowledge of the problem-space. Therefore, one single implementation is sufficient to estimate unknown variables for all kinds of BRDF’s. We encode all variables in a simple vector, which simultaneously serves as a genome. To evaluate the fitness of a possible result, a cost function simply generates values \( i_v \) for all visible pixels of the surface using the evolved genome. All lights, including virtual lights, are taken into the equation. All generated pixel values are then subtracted from the pixel values \( i_r \) of the real surface in the photograph. The fitness of a genome can be calculated with \( q = \sum_{i=0}^{\infty} \frac{i}{100} \). In the unlikely case that the denominator equals zero, a perfect match (i.e. a perfect genome) has been found.

6 RESULTS

An P4/2.4 GHz PC with a NVidia 6600 GT graphics board and 1 GB memory was used to conduct our tests, with the OpenSG (OpenSG, 2007) rendering system and the Avalon (IR, 2007) framework for application description. Test data was acquired with a Canon EOS 350D camera for scene backgrounds and the light probe. HDR photos were generated via Debevec’s HDRShop. The dragon model in figure 2 is rendered in a 1500 × 1000 pixel context with 8 × FSAA, an SH analysis with 9 coefficients, a mixture of 25% diffuse and 75% specular HDR irradiance maps, static AO and PCF shadows. The blending into the real image is performed via differential rendering and the final image is drawn at 9 FPS. Much higher framerates (up to 60) are achieved for low-polygon models such as the Stanford bunny. For less complex models dynamic ambient occlusion can be enabled without major performance hits, though depending heavily on the sampling rate. The differential rendering automatically handles occlusions from real objects to virtual ones or vice versa, as shown in figure 3. To assure that light and shadows are transferred correctly onto real materials, the material reconstruction as described above is used to gather information about the surface the object is placed on.

Diffuse materials were reconstructed through the iterative method described above. We have tested a steady state genetic algorithm on a simulated Phong material to evaluate the quality of a reconstruction. 1000 surface samples were gathered to determine the parameters \( p_d, p_s \) and \( n \), with a population size of 100 genomes. The test results point out that linear parts of the equation \( f(x, \theta_0, \theta_0) = \frac{p_d}{\gamma} + \frac{4p_s}{\gamma} \cos^2 \gamma \) were evaluated with less deviation from the actual pa-
rameters than non-linear parts. While $\rho_d$ was evaluated correctly in most cases, i.e. no mutant or local minima, the deviations in $\rho_n$ and especially in $n$ were generally too high. It is still unclear whether a larger population or higher mutation rates will lead to better results. It should be noted that these test cases exclusively deal with known BRDF’s and do not contain any lighting information from an image whatsoever, neither virtual nor real lights. In the actual implementation, the process iteratively factors out virtual light sources. Ultimately, the calculation of the BRDF parameters that follows this estimation is replaced by the genetic algorithm.

7 FUTURE IMPROVEMENTS

The most urgent matter right now is to have a unified model for creating irradiance maps, because the currently used spherical harmonics for instance are unsuitable for high-frequency functions. Relating to actual reflection model parameters such as those of specular functions will then be much easier. Currently, Haar Wavelets show promising results, because the multi-resolution analysis allows to capture high frequencies with relatively few coefficients. Ambient occlusion as a placeholder for other surface functions is sufficient right now. However, special effects such as interreflections or caustics are currently not handled. A suitable and dynamic method comparable to LDPR (Sloan et al., 2005) has to be included in the near future. Also, the current approach to extract light sources manually from sphere maps does set heavy boundaries to the dynamic usage. A stable real time approach to extract lights from HDR sphere maps such as in (Supan and Stuppacher, 2006) or (Korn et al., 2006) still has to be implemented.

REFERENCES


