Efficient Self-Shadowing Using Image-Based Lighting on Glossy Surfaces

M. Knuth[1], C. Altenhofen[1], A. Kuijper[2] and J. Bender[2]

[1]Fraunhofer IGD, Germany
[2]TU Darmstadt, Germany

Figure 1: *Several results created with our new method.*

Abstract

In this paper we present a novel natural illumination approach for real-time rasterization-based rendering with environment map-based high dynamic range lighting. Our approach allows to use all kinds of glossiness values for surfaces, ranging continuously from completely diffuse up to mirror-like glossiness. This is achieved by combining cosine-based diffuse, glossy and mirror reflection models in one single lighting model. We approximate this model by filter functions, which are applied to the environment map. This results in a fast, image-based lookup for the different glossiness values which gives our technique the high performance that is necessary for real-time rendering. In contrast to existing real-time rasterization-based natural illumination techniques, our method has the capability of handling high gloss surfaces with directional self-occlusion. While previous works exchange the environment map by virtual point light sources in the whole lighting and shadow computation, we keep the full image information of the environment map in the lighting process and only use virtual point light sources for the shadow computation. Our technique was developed for the usage in real-time virtual prototyping systems for garments since here typically a small scene is lit by a large environment which fulfills the requirements for image-based lighting. In this application area high performance rendering techniques for dynamic scenes are essential since a physical simulation is usually running in parallel on the same machine. However, also other applications can benefit from our approach.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Colour, shading, shadowing, and texture/Radiosity/Raytracing

1. Introduction

In garment design interactive virtual prototyping systems require both a fast simulation system and a fast visualization in order to allow a real-time coupling of both. The scenes processed by such systems typically show a dressed character, which is illuminated by an environment. This is an ideal scenario for image-based lighting techniques. However, since self-occlusion and the material properties are important for the perception of the garment, a suitable visualization method must be able to handle self-shadowing of the scene as well as various materials used for the fabrics. As an additional constraint, the visualization must not allocate too

much of the system's computational power to avoid a slow-down of the simulation. When industrial prototype development comes into play, this gets apparent since the quality and accuracy of a virtual prototyping system depends on the ability to simulate high-resolution models.

Many-light approaches [DKH*12], which replace the environment map by a set of light sources, create nice self-shadowing results for diffuse reflection models. However, increasing the reflectivity reveals the light sources since they cannot capture the high frequency information stored inside the environment map. Increasing the number of light sources can solve the problem. However, a large number of light sources, which are required for realistic results, will slow down the rendering process far below an interactive level.

We solve this problem by introducing a novel tunable lighting model combined with an environment map and a directional occlusion technique which directly exploits the information stored in the environment map to prevent an information loss. Our lighting model has several advantages. It can handle both low frequency shadows and high frequency environments at the same time without introducing additional artifacts. Furthermore, it provides a continuous transition between diffuse and reflecting surfaces and supports directional occlusion. Our approach requires only little computational effort and does not rely on mesh-based pre-computations. In summary, it can be stated that our novel method allows to compute rasterization-based natural illumination with real-time directional occlusion while considering the surfaces' glossiness.

2. Related Work

The problem we target consists of three parts. First, a model based on a bidirectional reflectance distribution function (BRDF) is needed which is capable to handle various material effects. Second, natural illumination and directional occlusion of the scene are required. Third, in order to be used in combination with a real-time simulation system, everything which is related to a simulated geometry needs to be computed from scratch in each frame. The technique we present belongs to the category of natural illumination techniques, where a scene is illuminated by a distant high dynamic range light source. A good overview of the different methods in this area is given by Ritschel et al. [RDGK12]. Their state-of-the-art report provides a good survey over current global illumination techniques. Using images of environments in order to illuminate objects has a long tradition in computer graphics. Environment mapping techniques can be used to simulate a wide set of phenomena in real-time computer graphics [HS99]. An overview is given by Akenine-Möller et al. [AMHH08]. One application of environment maps is to use them as a lookup for the convolution between a BRDF and the environment map. Since a given reflection model can cover a large portion of the environment map, the computation of the convolution is a slow process. To achieve a higher performance Kautz et al. [KVHS00] present a multi-resolution approach to speed up the convolution process, which leads to one convolution for a given reflection model. Kautz and McCool [KM00] approximate a BRDF with an individual set of pre-filtered environment maps convoluted by exponents of cosine-based lobes. Our approach is an extension to this work with the novelty that a single environment map can be used to approximate all exponents. This allows us to approximate several BRDFs with the same environment map.

Environment maps can be used in conjunction with high dynamic range (HDR) images in order to capture natural illumination. Debevec [Deb98] presents an acquisition method for such images and how they can be used to render scenes under very realistic lighting conditions. This process is called image-based lighting. The tricky part for image-based lighting is the integration of self-shadowing of the scene. Contrary to ambient occlusion [SZ98] the environment is not a homogeneous light source but colored, which makes the incoming light dependent of its direction. However, often image-based lighting and ambient occlusion are combined, effectively splitting up the lighting and the shadow computations. For real-time applications this can be done by a set of lights generated by an importance sampling of the environment map. Examples for this are given in [Deb08, ADM*08]. Lighting and shadow computations can be split her, too like it is presented by DeCoro and Rusinkiewicz [DR08].

However, all current approaches in this area do not perform well with highly glossy surfaces [RDGK12]. The reason for this is that the all-frequency information of the environment map is substituted with a compressed one. Using highly glossy surfaces will reveal the compressed information, which is given by single light sources. Annen et al. [ADM*08] reduce this problem by using area light sources. However, these light sources still represent one constant value over an area, not revealing the full environment map information in case of highly glossy surfaces.

Our approach circumvents this problem by using the original environment map in the illumination computation. This allows us to use the original all-frequency information of the light probe. To support the usage of shadowing techniques, e.g. [DL06, AMB*07], we substitute the environment map by a set of light sources. Annen et al. [ADM*08] show that a shadowing technique requires soft shadowing and a fitting penumbra computation in order to obtain a smooth transition between the shadows of the single light sources. In the work of Dong et al. [DGR*09] this technique is extended to virtual point lights in conjunction with clustering in order to achieve interactive indirect illumination of diffuse scenes. There exist various soft shadowing techniques for real-time rendering, which have all their advantages and disadvantages. A survey of suitable methods is given in [KS14].

Recently, screen space methods have been developed

which address ambient and directional occlusion computations. These methods have the big advantage to be independent of the scene complexity. Screen space ambient occlusion (SSAO) was first discussed by Crytek [Mit07] and used in their video game Crysis. Since its introduction numerous extensions and contributions in order to improve it were developed [RGS09,KRES11,BKB13]. An extension of SSAO towards directional occlusion is presented by Ritschel et al. [RGS09]. Klehm et al. [KRES11] present an alternative approach, which samples an environment map within a cone approximating the area the surface sees from the environment map. This is similar to our used scheme. However, the sampling process still needs to be performed per pixel with multiple lookups into the environment map to get the area information for the visible portion of the map. Such techniques mainly rely on depth information of the current view for their computations. This is problematic since cavities are ignored, which are hidden in the current view. Thus, the whole scene does not contribute to the occlusion and the deviations from a ground truth differ more and more for the low frequency shadowing effects.

3. Concept

Our main goal is to achieve natural lighting for real-time rendering with directional occlusion, not only for diffuse and low gloss surfaces but also for high glossiness up to mirror like surfaces.

We identified two main problems of existing methods that we must address at the same time to reach this goal. First, the necessity to use all information of the environment map for the lighting process. Existing methods substitute the environment map with a set of light sources which is problematic when dealing with high gloss surfaces and limits the algorithms to diffuse and low gloss surfaces [RDGK12]. In contrast to previous methods we use the full image information of the environment map in the lighting process and compute only shadows with a set of virtual point light sources. This allows us to handle high gloss surfaces without problems. Second, a lighting model is required which supports a smooth transition between diffuse and mirror like glossiness. To meet this requirement we propose a lighting model based on a filtering of the environment maps.

The basic outline of our new algorithm is:

1. Subdivide the original environment map into discrete regions and create a placeholder light source for each region (see Section 3.1). In this way we will do the occlusion computations for the self-occlusion of the scene.
2. Filter the generated discrete regions with our proposed method to get a lookup for the lighting model (see Section 3.2). We call the resulting filtered maps partial environment maps (PEM). This filtering allows us to use the PEMs as a direct lookup table for the lighting already including the glossiness of the materials. The novelty of

this approach is that the lighting information is not only stored for a direction but also for a given glossiness value.
3. Use shadow mapping to get the visibility information for every light source providing information about the scene's self-occlusion (see Section 3.3).
4. Determine the color for each pixel by combining the visibility information of the shadow maps with the color information of the (filtered) environment maps (see Section 3.4). This concludes the algorithm by combining occlusion values with their corresponding light information. Since this light information already contains the glossiness term of the material system, the result is a value containing material properties, occlusion and the lighting information of the whole environment map.

3.1. Environment Map Subdivision

In order to compute shadows with image-based lighting, the environment map is discretized into a set of light sources, where each light represents a certain area of the map. Similar to the work of Annen et al. [ADM*08] we perform an importance sampling over the environment map to place the light sources depending on the brightness distribution of the environment map. The resulting partitioning should fulfill the following requirements for a good approximation:

1. Each point in the image space of the environment map belongs to exactly one partition:

$$\bigcup_{i=1}^{N} p_i = P, \qquad \bigcap_{i=1}^{N} p_i = \emptyset, \tag{1}$$

where p_i represents the i-th partition of the original environment map P and N is the total number of partitions.
2. Each partition is equally important for the final lighting.

In order to work directly with a set of PEMs as a lookup table for the illumination for a given glossiness value, each PEM must embed the lighting model. The lighting of the scene is the sum of the single light sources. This is also the case for summing up the single regions of the environment map. Our choice to represent the material system as a set of energy-conserving filter functions exploits this similarity: It is equivalent whether we filter the sum of the single regions or whether we sum up the filtered regions. The latter allows us to use the single PEMs directly for lighting purposes including the lighting model: Instead of using the light sources directly to perform the lighting computations, we use the lights only for shadow computation while using their corresponding PEMs for the lighting itself.

3.2. Convolution-Based Model

Our motivation to use a simple, cosine-based lighting model is based on the results of Lafortune et al. [LFTG97] and Filip et al. [FH04] showing that these functions can be used to approximate complex BRDFs and Bidirectional Texturing

Functions (BTF). The work of Kautz and McCool [KM00] discusses the process of pre-convolution of given environment maps with cosine-based symmetrical lobes for real-time rendering. The lobes can be used to approximate diffuse, Phong and mirror reflection depending on an exponentiation factor s_{Phong} chosen for the cosine function. However, in order to simulate a given BRDF it needs to be broken up into single lobes/direction pairs like it is done in [KM00]. The convolution for these pairs can then be directly looked up in the environment map generated by our approach.

The goal of the process we describe in the following is to get a single lookup table which allows us to get a continuous range from diffuse ($s_{\text{Phong}} = 1$) up to mirror like reflectivity ($s_{\text{Phong}} = \infty$).

Cosine-Based Lighting Model For completeness we will shortly describe the convolution of cosine based lobes and their convolution with an environment map. An environment map is described by a function $f : \mathbb{R}^3 \mapsto \mathbb{R}^3$ providing a color **c** for a given direction **v**:

$$\mathbf{c} = f(\mathbf{v}). \tag{2}$$

Additionally we define a glossiness factor $g \in [0,1]$ which maps $s_{\text{Phong}} \in [1,\infty]$ into a range which is simpler to address:

$$s_{\text{Phong}} = \frac{1}{1-g}. \tag{3}$$

Let f_{light} be such a function for the light probe which tells us the amount of incoming light for a given direction **v**. We use environment maps to store the result of the spherical convolution of the light probe f_{light} with our reflection model for a given glossiness value g and the direction $\mathbf{r}'$:

$$f_{\text{embedded}}(\mathbf{r}', g) = \int_{\Omega^+} f_{\text{light}}(\omega) \cdot (\mathbf{r}' \cdot \omega)^{s_{\text{Phong}}} d\omega. \tag{4}$$

Here, Ω^+ is the positive hemisphere for the direction **n** and ω describes all vectors belonging to this hemisphere. This gives us a set of functions which allows to directly look up the result of the convolution of f_{light} for a given glossiness g. These functions allow us to embed our lighting model in the environment map. The function f_{embedded} has the form $f : \mathbb{R}^4 \mapsto \mathbb{R}^3$. Hence, mapping it to environment maps requires one environment map per value of g. Kautz and McCool use a similar technique in [KM00]. They compute a set of environment maps for a given set of g values in order to approximate single BRDFs. When using several BRDFs, their technique requires storing a lot of environment maps which is unpractical since a lot of memory is consumed. Thus, our goal is to get one lookup table which can approximate Equation (4) directly.

Approximation Our observation is that applying the Phong model or exponents of cosine-based lobes to a light probe results in a behavior similar to a variable low pass filter.
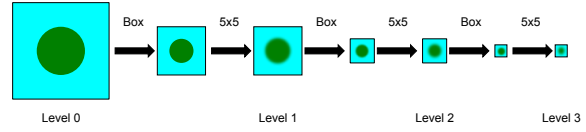

Figure 2: *For computation we perform a shrink and a filter operation in an alternating way. The shrink operation is the classical 2×2 pixel mean operation used for MIP mapping. The additional filter we introduce in between allows us to shape the result. As a result we get a MIP map level pyramid which contains a convolution with a mirror reflective glossiness at the top resolution and diffuse reflection on the lowest resolution level.*

Therefore, environment maps of convolutions of larger lobes will contain less information than environment maps with smaller lobes.

Instead of using a set of full resolution environment maps we use MIP map levels of one environment map to store the low frequency lobe convolutions. A direct lookup of a certain MIP map level is possible for most environment mapping techniques. Using the MIP map lookup as additional degree of freedom allows us to perform a lookup for a given direction and glossiness in one step or texture read. The MIP map levels of an environment map are already low pass filtered versions of the original. Thus, the stack of MIP map environment maps already behaves similar to Equation (4) if g is chosen accordingly (see dotted violet line in Figure 5). In this case a given g corresponds to a given MIP map level, giving us a function $f : \mathbb{R}^4 \mapsto \mathbb{R}^3$ which allows the lookup of direction and glossiness. A continuous range for g is generated by using a hardware trilinear interpolation of the environment map. As shown in Figure 5, MIP maps alone do not perform well and create strong deviations form ground truth lobes. This is especially visible in the lobe cross section plot of the important diffuse reflection case ($s_{\text{Phong}} = 1$) (see "'MIP map only'" lobes in Figure 4). To increase the control of the low pass filtering process we added an additional 5×5 low pass filter for each MIP map generation stage. The whole filtering process is shown in Figure 2. This way the approximation can be improved significantly and we get a simple to use and very fast lookup method for the lighting. As a side effect, the generated lookup table is physically plausible, since no additional energy is introduced as long the 5×5 filters are normalized.

3.3. Occlusion Computations

The introduced MIP map-based approximation allows us to use the generated environment maps as a representation of real area light sources of arbitrary form and coloring while the shadows are computed based on simple directional lights representing these areas. Since the PEMs already contain the convolution of various filter kernel sizes, the material lookup

Figure 3: *Applying different filtering levels to the environment map creates more or less glossy surfaces from mirror-like (back) to completely diffuse (front).*

can be done with just a single lookup in the corresponding PEM.

To compute the visibility for each PEM, we use an approach based on shadow mapping. Since every PEM can be interpreted as an area light source, a soft-shadowing algorithm for area light sources is needed. Annen et al. [ADM*08] introduce a suitable method with high quality shadows for differently large light sources.

3.4. Final Lighting Computation

For each pixel we compute the sample vector for texture lookup just as for traditional environment mapping. We can use the same sample vector for each PEM. Then we compute the final color **c** of a pixel as a weighted sum over the corresponding texels of each PEM:

$$\mathbf{c} = \sum_{i=1}^{N} VIS_i(x,y) \cdot PEM_i(x,y), \tag{5}$$

where N is the number of PEMs and $PEM_i(x,y)$ denotes the color of pixel (x,y) stored in the i-th PEM. The weights are given by the visibility factor VIS_i which is computed using soft shadow mapping.

4. Implementation

We implemented our approach on a Windows-based system using OpenGL in conjunction with GPU shader programs. As an environment map model we used cube maps. They are directly supported by the GPU and allow fast and direct access to their MIP map levels within shaders. To sample the initial environment map we implemented Debevec's median cut algorithm [Deb08]. It is a fast and easy to compute subdivision algorithm which fulfills the requirement that all resulting regions have to be equally important for the global lighting. Bright areas in the map yield multiple small regions

whereas dark areas yield only few large regions. This process gives us a set of light sources and a set of regions. Originally developed for 2D maps, we simply apply the process to the six sides of the cube. Therefore, the number of generated lights is a multitude of six.

As a filter for the convolution in the MIP map generation process of the PEMs we tested several functions. We investigated filters based on the Gaussian distribution, the cosine function, several exponentiations of the cosine function and the box filter function mapped to the 5×5 filter size. Results of filtering with a square root of the cosine function at different levels are shown in Figure 3. Filtering can improve the approximation significantly. Especially a filter based on the square root of the cosine function shows a nice approximation behavior which is discussed in Section 5. Since the 5×5 filters are symmetric, classical fast filter convolution is used applying a 1D filter separately for rows and subsequently for the columns of a face. At edges, the neighboring face needs to be taken into account. For the cosine-based filters we used exponents of $[\cos(-60), \cos(-30), \cos(0), \cos(30), \cos(60)]$ divided by the sum of the single elements for generating the 1D filters.

The resulting PEMs are stored as individual textures in GPU memory. For an easy and fast implementation we used cube maps. For each PEM we create a plain copy of the original environment map and just blacken all texels that do not belong to the current PEM before we start filtering. This technique increases the memory consumption, but allows some major simplifications while lighting the scene.

We use filtered shadows based on directional light sources to handle occlusions. As a shadowing technique we decided to use Variance Shadow Maps (VSMs) [DL06, AMHH08] since they are simple and fast. The shadows are softened by applying a Gaussian filter over the entire depth texture. The size of the filter kernel determines the how "soft" the shadows will be. The filter kernel size depends on the area of the corresponding PEM to simulate an area light source of this size.

5. Results

Filter Quality Embedding the whole material system into the MIP maps of the environment maps by using low pass filters was not used before in conjunction with natural illumination. Therefore, we first want to investigate the quality and performance of our approximation by filter functions. To compare our model (see Equation (4)) with its approximation differential images could be used. However, in our case it is possible to compare the quality of the basic functions generated by both methods directly which gives us plottable results. To perform this comparison we have fixed the direction vector. For the approximation this means having a black environment map with one white pixel, which
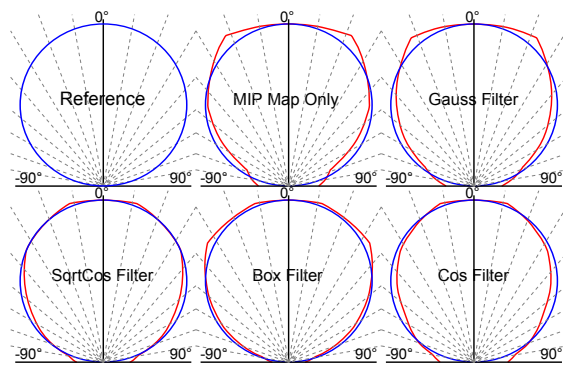
Figure 4: *Cross sections of 3D radial plots of the effects of the filter functions used to approximate lighting lobes. The blue lobe corresponds to a given diffuse reflection while the red lobes are the computed approximations. MIP map only depicts the result without additional filtering. To increase the quality we additionally apply different 5×5 low pass filters (Gaussian, cosine, box and square root cosine filter).*
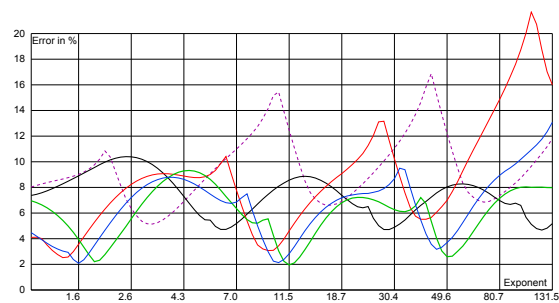

Figure 5: *Results for different filters: no filtering (dotted violet line), Gaussian filter (black), box filter (red), cosine filter (green) and the square root cosine filter (blue). The relative error is computed as the difference between the target lobe and the approximated lobe when both have the same volume. The error is given in percent of the target lobe's volume.*

is then filtered. To compare the generated lobes we fit their volumes by finding the corresponding MIP map level for a given glossiness. The remaining differences in the volumes are then used to compute the relative error. Note that this approach can also be used to calibrate the approximation since the lobe volume is characteristical for a given glossiness. We tested five different filter functions. For $s_{\text{Phong}} = 1$ the cross sections of the radial plots of the function compared to the MIP map-based approximation are shown in Figure 4. The spiky appearance of the curves is caused by the liner interpolation between the single MIP map levels. A slope represents the error introduced by the interpolation, a spike represents the error at an integer MIP map level. In this figure an approximation of $s_{\text{Phong}} = 1$ (blue curve) is performed. The red curves represent the approximations of the different filter functions. Especially for the case $s_{\text{Phong}} = 1$ we aimed to get a good approximation since it corresponds to pure diffuse reflection. As expected, the additional filter functions speed up the decimation of high frequency information in the environment map. The data stored for the diffuse reflection is now located in a higher MIP map level having more sample points to represent it. This improves the quality of the approximation at integer MIP map levels. In our test the box cosine filter performed best with an error of roughly 4%.

In the next step we investigated the deviations introduced by the linear interpolation between the MIP map levels. These deviations are shown in Figure 5. The choice of the filter function gradually changes the impact of the deviations. As it can be seen, no filtering (dotted violet) and the box filter (red) both introduce high spikes of relative error when s_{Phong} rises. The Gauss filter (black) is stable, but has a high overall error rate. Using the cosine filter (green) creates results

which are always below 10% of the error but it has a strong deviation for the important diffuse reflection case. The blue curve depicts the behavior for the square root cosine filter. This filter shows promising results but it tends to introduce high deviations when the exponent gets into high regions. In practice the deviation for very high exponents is not that critical since it is near the transition to the reflection case. Note that the resolution of the environment map is the limiting factor of our approximation. For high exponents of the cosine function the resolution limit is reached which leads to stronger deviations. This can be seen on the right side of Figure 5. Due to these results we have chosen to use the square root cosine filter. It it performs nearly as good as the box filter for the diffuse case, but has a much better behavior when increasing s_{Phong}. The results of rendering scenes while using this filter are shown in Figure 1. A comparison to often used Ambient Occlusion techniques demonstrates that considering the reflectivity for the occlusion computation is clearly necessary in case of high glossiness (see Figure 6).

Performance All tests were computed on a desktop PC with an AMD Phenom II X4 955 CPU at 3.2 GHz, 4 GB RAM and an AMD Radeon HD 6850 GPU at a clock rate of 775 MHz with 1 GB memory. Due to the high number of environment maps multi-pass rendering was used applying 12 PEMs and their corresponding shadow maps per pass.

The main performance factor is the number of subdivisions used (see Figure 7). Each region has its own PEM and a shadow map which have to be stored on the GPU. An overview of the necessary passes and the memory consumption is shown in Figure 8

As an alternative to usage of cube maps we have considered parabolic mapping. The parabolic maps can be scaled to fit the used region of the PEM which reduces the amount of required memory significantly. This can be easily explained:
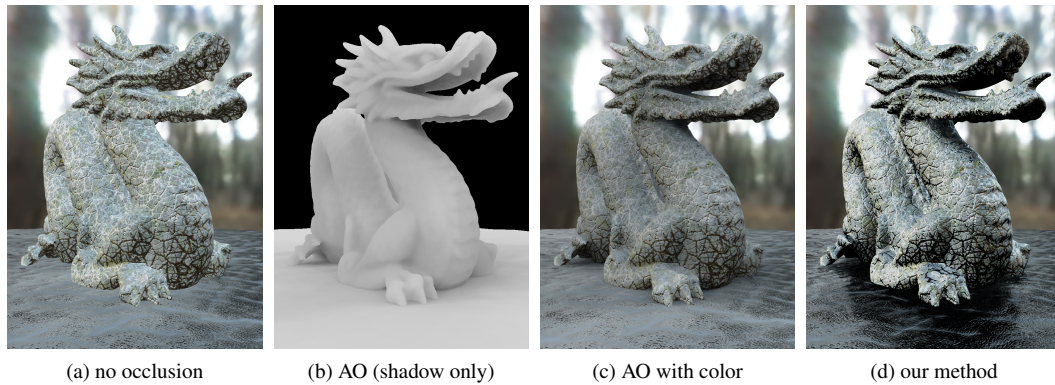
| (a) no occlusion | (b) AO (shadow only) | (c) AO with color | (d) our method |

Figure 6: *Comparison between our method and Ambient Occlusion (AO). Our approach considers the glossiness of the surface in the occlusion computation which leads to a completely different visual appearance of the object. Please note that for this comparison the images use a pure glossy material with no additional diffuse term. A full BRDF based image is shown in Figure 1.*
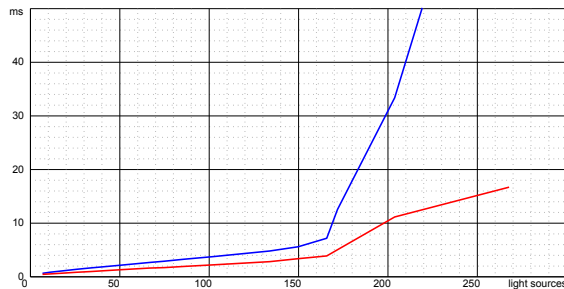

Figure 7: *Frame render-time for the teapot (red) and dragon (blue) scene with increasing number of light sources. Notice the increase for both scenes at about 170 light sources when total memory consumption becomes too large to completely fit into the GPU memory space.*

# Lights	PEMs		VSMs				
	128²	256²	64²	128²	256²	512²	1024²
1	1.5	6	0.015625	0.0625	0.25	1	4
12	18	72	0.1875	0.75	3	12	48
32	48	192	0.5	2	8	32	128
64	96	384	1	4	16	64	256
128	192	768	2	8	32	128	512
256	384	1536	4	16	64	256	1024

Figure 8: *Memory footprint (in MB). The PEMs use 16Bit-floats with 4 color channels (GL_RGBA16F) while the VSMs have only 2 channels (GL_RG16F).*

If we assume that all six sides of a cube map have a resolution of 512×512 pixels and 64 regions and that the corresponding 64 environment maps are stored with MIP maps and floating point color values, this leads to roughly 1.5 GB of memory which is required. Since the parabolic version only needs to cover the parts containing information, we end up using roughly a 256×256 resolution per parabolic map. Using 64 regions about 70 MB of memory is required. However, using the parabolic maps is much more complex than using cube maps. Due to their different sizes they behave differently in the approximation. Thus, for each map different values have to be applied in order to find the corresponding MIP map level for a given g. Therefore, the case $s_{\text{Phong}} = 1$ is not a constant MIP map level anymore which complicates the lookup. Additionally, the MIP maps correspond to different areas of the original environment map. This compli-

cates the lookup even further and is the main reason why we decided to use a cube map-based environment mapping instead.

Discussion In contrast to previous works our natural illumination approach for real-time rendering supports high gloss to mirror-reflective surfaces. Furthermore, our approximation scheme for the material system allows a fast computation of the necessary environment maps. Using our filtering technique speeds up the process to a point where video input can be used as a source for the environment map. Finally, our technique is very fast, easy to implement and creates high quality results.

However, our method has also some limitations. Embedding the material system into PEMs allows a fast evaluation but consumes a lot of memory since maps for several glossiness values have to be stored. In future we plan to investigate the use of texture arrays instead of MIP mapping in combination with a parabolic mapping to reduce the memory consumption significantly. The use of a linear interpolation to get the lookup values for the PEM yields plausible results,

but it is only a first step. In future we plan to use a BRDF approximation instead of the current linear interpolation. This should give us much more accurate results and allow the use of measured BRDFs and BTFs.

6. Conclusion and Future Work

We introduced a novel natural illumination method which allows the real-time rendering of highly glossy surfaces with self-shadowing. Since the evaluation of our tunable lighting model is too slow for real-time rendering, we presented a fast approximation method. This approximation uses information stored in the environment map to compute the illumination under changing glossiness values. Our proposed method of embedding the material system inside the PEMs results in the same run-time characteristics as standard image-based lighting using diffuse materials. The big difference lies within the capability to represent the full range between diffuse and reflective materials. Since we approximate basic exponents of cosine-based lobes, our method can be used to approximate more complex BRDF reflection models by combining several lobes in the style presented by Kautz and McCool [KM00].

As we only use a shadow map-based blocker test, reflections stay black. To solve this problem more complex methods are required like the one presented by Laurijssen et al. [LWLD11]. However, since their method heavily relies on pre-computations, it is limited to static scenes.

An application area for our technique is e.g. virtual prototyping of garments. If fabrics are woven or knitted using different yarns with varying reflectance properties, this will result in surfaces with a complex pattern of changing reflectance factors. With our novel technique, it is for the first time possible to handle these changes in one reflection model. Moreover, several extension of our method are possible. One of the most interesting extensions would be to use the method to approximate measured BRDFs and BTFs. Basically our method allows a fast evaluation of cosine-based radiation lobes on a given environment map. Thus, it can be used to speed up the process of evaluating the single cosine terms of an approximation of measured functions.

Acknowledgment

References

[ADM*08] ANNEN T., DONG Z., MERTENS T., BEKAERT P., SEIDEL H.-P., KAUTZ J.: Real-time, all-frequency shadows in dynamic scenes. *ACM Trans. Graph. 27*, 3 (2008). 2, 3, 5

[AMB*07] ANNEN T., MERTENS T., BEKAERT P., SEIDEL H.-P., KAUTZ J.: Convolution shadow maps. In *Proc. EGSR* (2007), Eurographics Association, pp. 51–60. 2

[AMHH08] AKENINE-MÖLLER T., HAINES E., HOFFMAN N.: *Real-Time Rendering 3rd Edition*. A. K. Peters, 2008. 2, 5

[BKB13] BAUER F., KNUTH M., BENDER J.: Screen-space ambient occlusion using A-buffer techniques. In *Computer-Aided Design and Computer Graphics* (Nov. 2013), IEEE. 3

[Deb98] DEBEVEC P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proc. SIGGRAPH* (1998), pp. 189–198. 2

[Deb08] DEBEVEC P.: A median cut algorithm for light probe sampling. In *ACM SIGGRAPH Classes* (2008), ACM. 2, 5

[DGR*09] DONG Z., GROSCH T., RITSCHEL T., KAUTZ J., SEIDEL H.-P.: Real-time indirect illumination with clustered visibility. In *VMV* (2009), pp. 187–196. 2

[DKH*12] DACHSBACHER C., KRIVÁNEK J., HASAN M., AR-BREE A., WALTER B., NOVÁK J.: Scalable realistic rendering with many-light methods. In *EG 2013 - STARs* (2012), Eurographics Association, pp. 23–38. 2

[DL06] DONNELLY W., LAURITZEN A.: Variance shadow maps. In *Proc. Interactive 3D graphics and games* (2006). 2, 5

[DR08] DECORO C., RUSINKIEWICZ S.: Subtractive shadows: A flexible framework for shadow level of detail. *J. Graphics Tools 13*, 1 (2008), 45–56. 2

[FH04] FILIP J., HAINDL M.: Non-linear reflectance model for bidirectional texture function synthesis. In *Proc. Pattern Recognition (ICPR)* (2004), vol. 1, pp. 80–83 Vol.1. 3

[HS99] HEIDRICH W., SEIDEL H.-P.: Realistic, hardware-accelerated shading and lighting. In *Proc. SIGGRAPH* (1999), pp. 171–178. 2

[KM00] KAUTZ J., MCCOOL M. D.: Approximation of glossy reflection with prefiltered environment maps. In *Graphics Interface* (2000), pp. 119–126. 2, 4, 8

[KRES11] KLEHM O., RITSCHEL T., EISEMANN E., SEIDEL H.-P.: Bent normals and cones in screen-space. In *VMV* (2011), Eurographics Association, pp. 177–182. 3

[KS14] KOLIVAND H., SUNAR M.: Anti-aliasing in image based shadow generation techniques: a comprehensive survey. *Multimedia Tools and Applications* (2014), 1–27. 2

[KVHS00] KAUTZ J., VÁZQUEZ P.-P., HEIDRICH W., SEIDEL H.-P.: Unified approach to prefiltered environment maps. In *Proc. Eurographics Workshop on Rendering Techniques* (2000), Springer-Verlag, pp. 185–196. 2

[LFTG97] LAFORTUNE E. P. F., FOO S.-C., TORRANCE K. E., GREENBERG D. P.: Non-linear approximation of reflectance functions. In *Proc. SIGGRAPH* (1997), ACM, pp. 117–126. 3

[LWLD11] LAURIJSSEN J., WANG R., LAGAE A., DUTRÉ P.: Pre-computed gathering of multi-bounce glossy reflections. *Comput. Graph. Forum* (2011), 2270–2278. 8

[Mit07] MITTRING M.: Finding next gen: Cryengine 2. In *ACM SIGGRAPH courses* (2007), ACM, pp. 97–121. 3

[RDGK12] RITSCHEL T., DACHSBACHER C., GROSCH T., KAUTZ J.: The state of the art in interactive global illumination. *Comput. Graph. Forum 31*, 1 (Feb. 2012), 160–188. 2, 3

[RGS09] RITSCHEL T., GROSCH T., SEIDEL H.-P.: Approximating Dynamic Global Illumination in Screen Space. In *Proc. Interactive 3D Graphics and Games* (2009). 3

[SZ98] S. ZHUKOV A. IONES G.: An ambient light illumination model. In *Proc. Rendering Techniques* (1998). 2