Volume xx (200y), Number z, pp. 1–14

Fullsphere Irradiance Factorization for Real-time All-frequency Illumination for Dynamic Scenes

D. Michael and Y. Chrysanthou

Computer Science Department, University of Cyprus

Abstract

Computation of illumination with soft shadows from all frequency environment maps, is a computational expensive process. Use of precomputations add the limitation that receiver's geometry must be known in advance, since Irradiance computation takes into account the receiver's normal direction. We propose a method that allows the accumulation of the contribution to the Irradiance, on a possible receiver, from all light sources in the scene, without knowing the receiver's geometry. This expensive computation is done at the preprocessing. The precomputed value is used at run time to compute the Irradiance arriving at any receiver with known direction. We show how using this technique we compute soft shadows and self shadows in real-time from all-frequency environments, with only modest memory requirements. A GPU implementation of the method, yields high frame rates even for complex scenes with dozens of dynamic occluders and receivers.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Color, shading shadowing and texture

1. Introduction

A correct account for illumination and shadows produced by complex environmental lighting can greatly improve the visual realism of real-time applications such as training simulators, computer games, CAD programs etc. The process is computation-intensive since it requires a weighted integral over all light sources in the scene, taking into account expensive occlusion calculations. The cost increases with the number of light sources in the scene. Thus, typically the quality is compromised by not considering all-frequency environments [SKS02, KK03, RWS*06].

Real-time illumination methods, in order to reduce the computations needed at run time, make use of precomputations that usually have heavy memory requirements for storing the preprocessing values. In addition, the use of precomputations adds the limitation that static parameters must be used, such as static or rigid receivers [IDYN07].

In this paper we introduce a technique that can be used for real-time illumination with shadows from all frequency envrionment maps, for fully dynamic receivers, and with only moderate needs in memory space, Figure 1.

The method is based on a reformulation of the Irradiance computation that allows us to factor out the direction of the receiver's normal, for diffuse surfaces, [WHG99, Wil00, LLAP05]. We introduce a new notion, the *Fullsphere Irradiance* that is a modification of the Irradiance in that it takes in account energy arriving at a point from all the light sources in the scene, not only those that lie in the positive hemisphere of the receiver. Using the factorization, we are able to encode the Fullsphere Irradiance within a 3D vector, called *Fullsphere Irradiance Vector* (FIV), whose values are valid for any direction of the receiver.

We applied our technique for computing in real-time inter- and intra-object soft shadows by encoding also the occluded irradiance using FIVs. A FIV can be precomputed for the part of the environment hidden by each occluder, placing the occluder each time at a position of a dense set of sample points. Then, at run-time, given a point on a scene surface, we can use the FIV values to quickly compute the unoccluded Irradiance using merely a dot product per occluder, per color channel, of the receiver's normal.

The contribution of our paper is two-fold: (i) it introduces the notion of the *Fullsphere Irradiance Vectors*, which allow

Figure 1: *Real-time illumination with soft shadows for fully dynamic receivers from all frequency environments.*

a pre-integration of light sources contribution, independently to the receiver's normal, (ii) it proposes a simple and scalable method that employs this novel idea and computes both soft and hard shadows in complex scenes from all-frequency lighting conditions (envrionment maps). Our results, show a GPU implementation of the latter yielding high frame rates even for scenes with dozens of dynamic occluders and receivers.

2. Related Work

Illumination algorithms have been at the core of computer graphics from the very early days. Most recent related work on real-time illumination has been focusing on accelerating the illumination calculation by precomputations. We briefly review work relevant to our method.

In Precomputed Radiance Transfer (PRT) the visibility is precomputed and stored, along with the BRDF, as basis functions per vertex of the object. These basis functions are then multiplied with the illumination at run-time. PRT was originally proposed for low-frequency illumination [SKS02, KSS02] and later extended to account for more general incident light and BRDF [LSSS04, NRH04, SM06]. PRT is more general than our method, in terms of the reflectance properties of the objects. However, it is inherently geared towards static scenes since the visibility is precomputed per vertex of the object.

PRT has been used with dynamic scenes. Kautz et al. [KLA04] propose a method for on-the-fly rasterization of the visibility for each vertex, however it is doubtful whether this method can scale to larger scenes. Mei et al. [MSW04] and Zhou et al. [ZHL*05] store the visibility in separate maps while [SM06] extend wavelet product integrals to separate local and global visibility. Despite the progress, the memory requirements typically remain extremely large, and the cost of the on-line step limits the method to scenes with a small number of occluders.

The literature is rich with many different approaches to soft shadow computation [HLHS03]. Some of the more popular ones are extensions of hard shadows computation approaches, taking into account multiple samples on an area light source [GBP06, SS08]. More recent approaches, aim to produce soft shadows from a full environment map, usually making approximations either to the occluders geometry [RWS*06] or to the environment map [ADM*08].

Occlusion computation is typically the most time consuming part of any high quality illumination algorithm. It is thus not surprising that the idea of pre-sampling the occlusion of objects for dynamic scenes, has been around for a while in various applications: radiosity [OCL96], ambient occlusion [KL05, MMAH07], shadows [ZHL*05] and interreflections [MSW04]. Among these, the method of Zhou et al. [ZHL*05] is closer to ours, since it also allows the inclusion of all-frequency illumination. In their method, for each sample view taken around an occluder, a complete occlusion field is stored. In contrast, in our method, the occlusion field is only used at preprocessing, as an intermediate step to compute the FIVs of the occluded part of the environment map. Although their approach can also account for local lights, it requires storing hundreds of Megabytes of memory per object and incurs a computational intensive run-time step. Our factorization reduces the memory requirement by two orders of magnitude and more than an order of magnitude speed up at run-time, at the expense of having a fixed environment map.

The idea of decoupling the receiver's normal form the incident radiance have been employed before. Everitt [Eve00] used this idea to perform per pixel illumination when this was not feasible with graphics hardware, but the method works only for a single light source. The Vector Irradiance, as the part of the Irradiance that is independent of the normal is called, have been used for Radiosity computation [WHG99, Wil00, LLAP05]. However in these cases, it is known in advance which light sources (or emmiting pacthes) lie in the positive hemisphere of the receiving patch. In PRT, Sloan [Slo06] used the idea of decoupling the normal from incident Irradiance, in order to handle normal maps. However since the radiance transfer is precomputed, this technique can not handle fully dynamic receivers and works only for rigid objects.

3. Fullsphere Irradiance Vectors

The reflected radiance L_{out} in direction $\vec{\omega}_o$ at a point p, with surface normal $\vec{N}_p$ and BRDF $f_p(\vec{\omega}_i, \vec{\omega}_o)$ is given by the following equation:

$$L_{out,p}(\vec{\omega}_o) = \int_{\Omega^+_{\vec{N}_p}} L_{in,p}(\vec{\omega}_i) f_p(\vec{\omega}_i, \vec{\omega}_o) \cos <\vec{N}_p, \vec{\omega}_i> d\vec{\omega}_i \quad (1)$$

where $L_{in,p}(\vec{\omega}_i)$ is the radiance coming from direction $\vec{\omega}_i$ and $\Omega^+_{\vec{N}_p}$ is the positive hemisphere of the point p.

Assuming diffuse reflectance only, the BRDF is constant and the term $f(\vec{\omega}_i, \vec{\omega}_o)$ can be substituted by a reflectance factor ρ_{dif} which can be factored out from the integral, thus the radiance equation for diffuse surfaces becomes:

$$L_{out,p}(\vec{\omega}_o) = \rho_{dif} \int_{\Omega^+_{\vec{N}_p}} L_{in,p}(\vec{\omega}_i) \cos <\vec{N}_p, \vec{\omega}_i> d\vec{\omega}_i \quad (2)$$

$$L_{out,p}(\vec{\omega}_o) = \rho_{dif} I_p(\vec{N}_p)$$

where $I_p(\vec{N}_p)$ is the Irradiance arrived at the point p from all directions in the positive hemisphere of p, i.e.

$$I_p(\vec{N}_p) = \int_{\Omega^+_{\vec{N}_p}} L_{in,p}(\vec{\omega}_i) \cos <\vec{N}_p, \vec{\omega}_i> d\vec{\omega}_i \quad (3)$$

We introduce here a new term, *Fullsphere Irradiance*, FI, that differs from the Irradiance equation in that the integral is over **all** directions of the sphere $\Omega_{\vec{N}_p}$.

$$FI_p(\vec{N}_p) = \int_{\Omega_{\vec{N}_p}} L_{in,p}(\vec{\omega}_i) \cos <\vec{N}_p, \vec{\omega}_i> d\vec{\omega}_i \quad (4)$$

where $\Omega_{\vec{N}_p}$ denotes all directions over the whole sphere.

Conceptually, the *Fullsphere Irradiance* differs from the *Irradiance*, in that "we let" lighting behind the surface of the point p to affect the value of $FI_p(\vec{N}_p)$. Note that $\cos < \vec{N}_p, \vec{\omega}_i >$ may have negative values.

Assuming distant lighting, light sources become directional thus the computation of the Irradiance becomes independent of the position's coordinates of the receiving point p. This is the case when the illumination is computed from an environment map [Deb98]. We will refer to this environment map, as the *lighting environment*, *Env*. For M discrete directional light sources in the scene, from which the M^+ lie in the positive hemisphere of the receiver the equations of the *Irradiance* and *Fullsphere Irradiance* become as below:

$$I_p(\vec{N}_p) = \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i) \cos < \vec{N}_p, \vec{\omega}_i >) \quad (5)$$

$$FI_p(\vec{N}_p) = \sum_{\vec{\omega}_i \in M} (L_{in}(\vec{\omega}_i) \cos < \vec{N}_p, \vec{\omega}_i >) \quad (6)$$

Note that in cases when all light sources in the scene, are in the positive hemisphere of the receiver, then the values of the *Irradiance* and the *Fullsphere Irradiance* are equal:

$$I_p(\vec{N}_p) = FI_p(\vec{N}_p) \; when \; M = M^+ \quad (7)$$

In the Irradiance equation 5, the normal $\vec{N}_p$ of the receiving point p is within the summation. This means the the Irradiance can be accumulated only once the normal $\vec{N}_p$ is known. For dynamic receivers this implies that this costly computation will necessarily have to be computed online. The *Irradiance* equation can be rearranged to factor out the normal from the summation. By replacing the cosine of the unit vectors $\vec{N}_p$ and $\vec{\omega}_i$, with the dot product of the two vectors, the summation can be broken into three components.

$$\begin{aligned} I_p(\vec{N}_p) &= \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i) \cos < \vec{N}_p, \vec{\omega}_i >) \\ &= \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i)(\omega_{i_x} N_x + \omega_{i_y} N_y + \omega_{i_z} N_z)) \\ &= N_x \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i)\omega_{i_x}) + N_y \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i)\omega_{i_y}) + N_z \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i)\omega_{i_z}) \\ &= (N_x, N_y, N_z)(\sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i)\omega_{i_x}), \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i)\omega_{i_y}), \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i)\omega_{i_z})) \end{aligned}$$

$$I_p(\vec{N}_p) = \vec{N}_p * IV(Env_{M^+}) \quad (8)$$

where $IV(Env_{M^+})$is the Irradiance Vector of an environment map or a scene *Env* with M^+ light sources lying in the positive hemisphere of p, similar to the Vector Irradiance used for radiosity computation in [WHG99, Wil00, LLAP05]. The Irradiance Vector is a 3D vector per color channel. Having computed the $IV(Env_{M^+})$ the Irradiance can be calculated using only a dot product between the normal of the receiver $\vec{N}_p$ and the $IV(Env_{M^+})$, see Equation 8. However we still have the issue of determining the M^+ light sources which still depend on the orientation of the receiving surface.

To eliminate this constrain of the dependency on the normal of the receiver, we introduce here the term of *Fullsphere Irradiance Vector*. The *Fullsphere Irradiance Vector* is similar to the *Irradiance Vector* in the same way that the *Fullsphere Irradiance* is similar to the *Irradiance*. That is, in the *Fullsphere Irradiance Vector* we consider to the calculation all the light sources, even those that are in the negative hemisphere of the receiver.

$$FIV(ENV_M) = (\sum_{\vec{\omega}_i \in M} (L_{in}(\vec{\omega}_i)\omega_{i_x}), \sum_{\vec{\omega}_i \in M} (L_{in}(\vec{\omega}_i)\omega_{i_y}), \sum_{\vec{\omega}_i \in M} (L_{in}(\vec{\omega}_i)\omega_{i_z})) \quad (9)$$

Note that in the same way we prove the Equation 8, we can prove that:

$$FI_p(\vec{N}_p) = \vec{N}_p * FIV(Env_M) \quad (10)$$

Since FIV does not have any dependency on the normal of the receiver, FIV for any *Env*, can be precomputed once and be valid for any dynamic receiver.

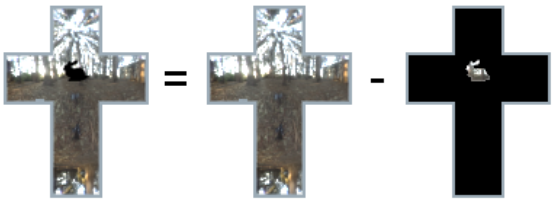

Figure 2: *The Irradiance I_p arriving at a point p that lies under the bunny (left), is equal to the total Irradiance $I_{tot,p}$ (middle) minus the Irradiance intercepted by the bunny $I_{occ,p}$ (right).*

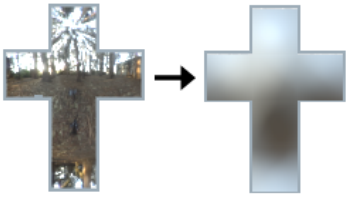

Figure 3: *The total diffuse Irradiance I_{tot} from the whole environment map, for each possible normal direction $\vec{N}_i$ of the receiver (left) is precomputed and stored in a cube texture (right).*

4. Illumination Using FIVs

4.1. Overview

Irradiance $I_p(\vec{N}_p)$ at any given point p in the scene can be computed as the total Irradiance from the whole environment assuming no occluders in the scene $I_{tot}(\vec{N}_p)$ minus the Irradiance intercepted by the occluders $I_{occ,p}(\vec{N}_p)$ [DSDD07], see Figure 2.

$$I_p(\vec{N}_p) = I_{tot,p}(\vec{N}_p) - I_{occ,p}(\vec{N}_p) \tag{11}$$

This can be proved as it is shown in the Appendix section. In a similar way we can prove that:

$$FI_p(\vec{N}_p) = FI_{tot,p}(\vec{N}_p) - FI_{occ,p}(\vec{N}_p) \tag{12}$$

Note that $FIV(Env_M)$ that can be used to calculate $FI_{tot,p}(\vec{N}_p)$ is independent of the normal of the receiver since for any N_p we sum up all M light sources of the environment map. This is not the case for $I_{tot,p}(\vec{N}_p)$ since in $IV(Env_{M^+})$, the determination of the M^+ light sources require prior knowledge of $\vec{N}_p$.

Our algorithm uses the Equations 11 and 12 proved above, to compute at run time the Irradiance arrived at a point p of a dynamic receiver, for shadows and self-shadows respectively. The proposed algorithm has two stages; the preprocessing and the runtime stage. At the preprocessing we precompute values that are independent of the dynamic parameters of the scene, such as occluders positions and receivers' geometry.

Precomputed data include: Irradiance arriving for each possible normal direction assuming no occlusion, $I_{tot,p}(\vec{N}_p)$, Fullsphere Irradiance Vector for the whole environment, $FIV(Env_M)$ and $FIVs$ for occluded parts of the environment placing each occluder in a number of sample positions.

The precomputed data are used at run time to compute Irradiance $I_p(\vec{N}_p)$, arriving at each point p of the geometry seen from each pixel. To evaluate Irradiance $I_p(\vec{N}_p)$, taking into account occlusions by other objects, precomputed values of $I_{tot,p}(\vec{N}_p)$ and $FIVs$ are used. In case of self-shadows the $FIVs$ are used among the $FIV(Env_M)$ of the whole environment. In both cases, computation of shadows and self-shadows, the equality between *Fullsphere Irradiance* and *Irradiance*, is used whenever is valid, see Equation 7, to transform from the notion of *Fullsphere Irradiance* for which we have precomputed information, to the notion of *Irradiance* that is what we want to evaluate.

In the next sections we describe in detail the preprocess and run time stages.

4.2. Preprocessing

At the preprocessing step we have to compute two categories of data that will be needed at run time:

- Total diffuse Irradiance for each sample normal direction, assuming no occlusion.
- A *Fullsphere Irradiance Vector*, FIV for a number of environment maps (the *Lighting* and *Occluded Environment Maps*).

In both cases we treat the environment map as a distant scene [Deb98]. The distant scene is considered far enough from the objects so its incoming light is seen as directional and the Irradiance arriving at any point of the local scene is independent of its absolute position. The Irradiance at a point is affected though, by the relative position of the other objects in the scene (local scene), due to shadows. In other words, each point is illuminated as the whole scene is translated to place that point, in the origin of the coordinate axes.

For the precomputation of the total diffuse Irradiance we create the *Diffuse Environment Irradiance Map*, ($DifMap$) [MH84]. It is stored in an HDR Cube Texture, Figure 3. Each texel of the $DifMap$, stores the total Irradiance $I_{tot,p}(\vec{N}_p)$ arriving at the surface with normal $\vec{N}_p$ from the *lighting environment Env*, assuming no occluders in the scene (computed using Equation 5). The $DifMap$ is indexed by the direction of the surface normal $\vec{N}_p$.

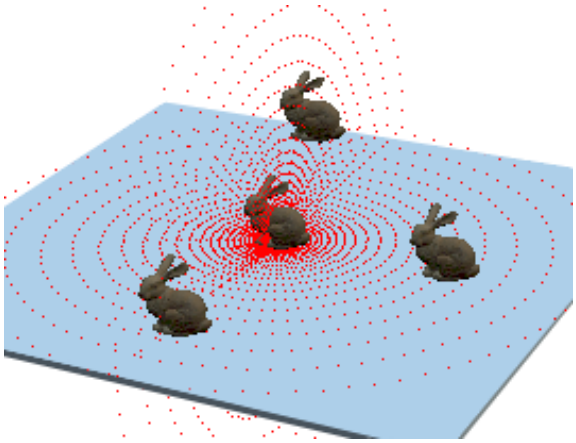

Figure 4: *Sample positions of the occluders.*

Figure 5: *The binary mask denoting the part of the environment map covered by the occluder (left) is ANDed with the environment map (middle) to get the occluded part of the environment map (right).*

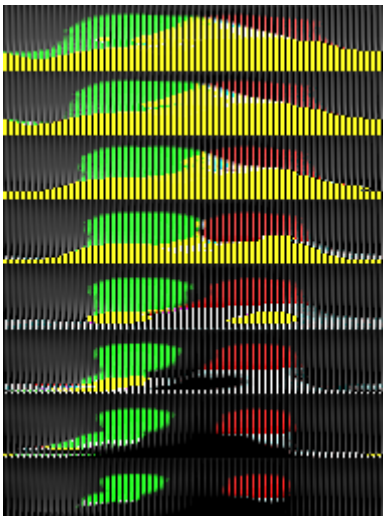

Figure 6: *The part of the texture that stores the precomputed FIVs for 8 different distance samples for the bunny in an environment map with one green and one red area light sources.*

The second category of the data precomputed, *Fullsphere Irradiance Vectors*, are computed using Equation 9, for a number of environment maps. The $FIV(Env)$ of the *lighting environment map Env* is computed and stored in a structure of a 3D color vector. This will be used to evaluate $FI_{tot,p}$, that is the *Fullsphere Irradiance* arriving at a point p from all over the environment *Env*.

Moreover we compute the *FIV* of each *Occluded Environment Map*, *EnvOcc*. $EnvOcc_{(\vartheta_i,\varphi_j,\rho_k)}$ is the environment map representing the part of the *Lighting Environment Map*, *Env*, occluded by an occluder placed at the position $(\vartheta_i,\varphi_j,\rho_k)$ in spherical coordinates, (see Figure5, right). An *EnvOcc* is computed at a number of sample positions for each occluder. The sample positions form a dense set (ϑ,φ) around the center of the scene and they are taken in concentric spheres of increasing distance ρ, Figure 4, similar to [ZHL*05].

To create an *Occluded Environment Map*, $EnvOcc_{(\vartheta_i,\varphi_j,\rho_k)}$, we place the occluder at the sample position $(\vartheta_i,\varphi_j,\rho_k)$ and compute the binary cube mask that defines the occluded part of the environment (Figure 5, left), similar to object occlusion field in [ZHL*05]. Using an "AND" boolean operation, pixel to pixel between environment map and the binary mask we get the part of the environment map that is occluded by the occluder in the specific sample position (Figure 5, right).

The FIV is computed for each one of the *Occluded Environment Maps* created. All $FIVs(EnvOcc)$ precomputed, are encoded within a floating point 2D texture, that we call *FIVs texture*, Figure 6. Since we may have negative values for the $FIVs(EnvOcc)$ components, and within a texture negative values are clipped to zero, we add an offset in order to get only positive values and be able to store the values within the texture. This offset will be deducted when the $FIV(EnvOcc)$ values are used at run time.

4.3. Runtime

In order to illuminate our scene, at run time we need to compute the radiance exiting each pixel, $L_{out,p}$. Since we assume only diffuse objects in the scene, the radiance at any pixel p is equal to the Irradiance $I_p(\vec{N_p})$ multiplied by the diffuse coefficient of the material of the receiver, Equation 2. The run time computation of $I_p(\vec{N_p})$ is done using the precomputed values of *FIVs*, taking into account shadows and self-shadows as described in the next subsection.

All runtime computations have been implemented on GPU to speed up the computation time. Moreover, the implementation in a fragment shader, allow us to have per pixel illumination.

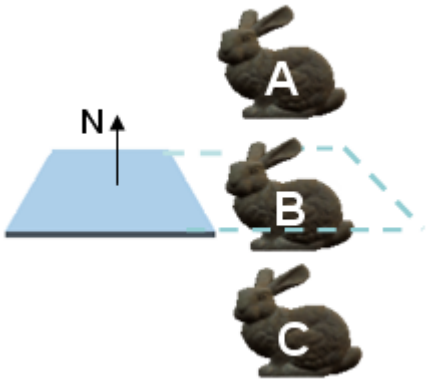

Figure 7: *Depending on the relative position of the occluder with regard to the receiver, there are 3 different cases for computing the occluded Irradiance: case A is computed using the FIV values, case C is totally ignored and case B is handled as special case.*

4.3.1. Shadows

To compute the Irradiance arriving at each pixel $I_p(\vec{N_p})$ we sum up the Irradiance occluded $I_{occ,p}$ by all occluders and deduct it from the total Irradiance, $I_{tot}(\vec{N_p})$, Equation 11.

Total Irradiance

To get the $I_{tot}(\vec{N_p})$, within the pixel shader, we use the normal (in world coordinates) of the geometry visible through the pixel, to look up in the cube texture of the *Diffuse Environment Irradiance Map*.

Occluded Irradiance

To get the total occluded Irradiance $I_{occ,p}$ we sum up the occluded Irradiance from each object in the scene. Since we assume a distant lighting environment, we only need to consider the relative position of each occluder to the receiver. There are three cases; the occluder may lie fully in the positive hemisphere of the receiver (Figure 7 case A), partially in the positive hemisphere (Figure 7 case B) or fully in the negative hemisphere of the receiver (Figure 7 case C). Cases C, do not occlude any Irradiance so are completely ignored in the computations. Cases B are special cases; in the subsection *Partial Occluders* we describe how we handle these cases.

For the occluders that lie fully in the positive hemisphere of the receiver (case A), we can compute the occluded Irradiance using $FIVs(EnvOcc)$. Based on the relative position (in terms of distance and direction) of occluder from the receiver, we find the closest sample point and look up the corresponding $FIV(EnvOcc)$ in the $FIVs$ *texture*. Knowing the normal of the receiver, the occluded Fullsphere Irradiance $FI_{occ,p}(\vec{N_p})$ is calculated as the dot-product of the normal $\vec{(N_p)}$, with the $FIV(EnvOcc)$, as defined in Equation 10. To further enhance the results, the 8 nearest samples of the $FIVs(EnvOcc)$ are trilinearly interpolated.

In case A, the occluder and all the light sources it occludes, are entirely in the positive hemisphere of the receiver. As a consequence, based on Equation 7, Occluded Fullsphere Irradiance is equal to Occluded Irradiance, $FI_{occ,p}(\vec{N_p}) = I_{occ,p}(\vec{N_p})$. The computed occluded Irradiance $I_{occ,p}(\vec{N_p})$ is deducted from the $I_{tot}(\vec{N_p})$ to get the Irradiance arriving at the pixel $I_p(\vec{N_p})$. $I_p(\vec{N_p})$ is used to shade the pixel.

Partial Occluders

In the preceding discussion we estimated the occluded Irradiance using the precomputed $FIV(EnvOcc)$, under the assumption that all light sources occluded are in the positive hemisphere of the receiver. However, this is not the case when an occluder is not fully in the positive halfspace of the receiver, case B in Figure 7. In such cases the use of $FIV(EnvOcc)$, result in an underestimation of the occluded Irradiance since the result would correspond to the Irradiance due to the light sources in the positive hemisphere (marked with green color in Figure 8) minus the Irradiance due to the light sources in the negative hemisphere (marked with orange color in Figure 8), $I_{FIV} = FI_{FIV_+} - FI_{FIV_-}$. The deduction of FI_{FIV} for the light sources in the negative hemisphere is because in these cases the dot product between receiver's normal and light direction gives negative value.

The correct value of the occluded Irradiance would be equal to the occluded Fullsphere Irradiance computed using only FIV_+, Figure 8. However, the FIV_+ can not be precomputed since it would rely on a prior knowledge of the receiver's orientation. The FIV_+ can be approximated by scaling the $FIV(EnvOcc)$ based on the percentage of the occluder that lies in the positive hemisphere over p. The percentage of the occluder in the positive hemisphere, is approximated with $(d+R)/2R$, at run time in a similar way that Malmer et al [MMAH07] do at their preprocessing step; d is the distance of the centre of the occluder form the plane of the receiver and R is the radius of the bounding sphere of the occluder. We count as occluded Irradiance $I_{occ,p}(\vec{N_p})$ the approximation of FI_{FIV_+}. The fact that the Irradiance is cosine weighted minimizes any error, that may occur by the approximation, Figure 17.

Multiple Occluders

In scenes with multiple occluders, pixels are shadowed by more than one occluder. In cases where two occluders hide a common part of the environment, the Irradiance occluded by the overlapping part, should be deducted only once from the total Irradiance. For clarity in the description, we explain in this section how we handle cases only of two overlapped

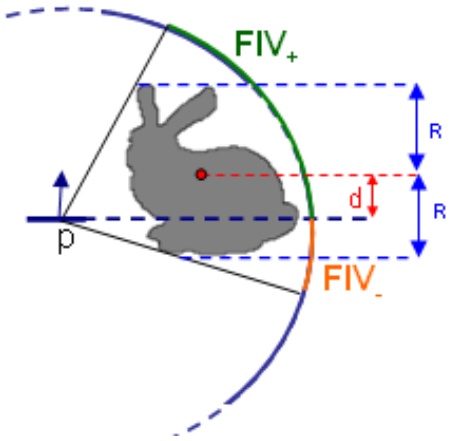

Figure 8: *At border cases only a part of the occluder intercepts Irradiance from the receiver.*

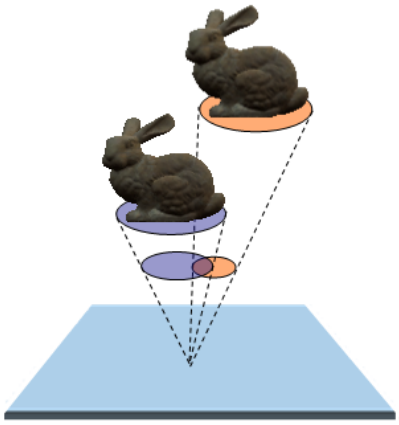

Figure 9: *The occlusion part of overlapping occluders is taken into account only once.*

occluders. The same concept can be used for an arbitrary number of overlapped occluders.

Occluded Irradiance by two overlapping occluders $I_{occ,p}$ at the receiving point p, is equal to the summation of the occluded Irradiance by each individual occluder $I_{occluder_i,p}$ minus the occluded Irradiance by the overlapped part of the two occluders, $I_{overlapped,p}$.

$$I_{occ,p} = I_{occluder_1,p} + I_{occluder_2,p} - I_{overlapped,p} \quad (13)$$

The occluded Irradiance by each occluder, $I_{occluder_i,p}$, is computed as described at the Section 4.3.1. The occluded Irradiance of the overlapping part is approximated as the percentage $Overlapped_{\%}$ of the area of the overlapping part $Area_{overlapped}$, over the summation of the area of the parts covered by the two occluders $Area_{occluder_i}$, of the total Irradiance occluded by the two.

$$Overlapped_{\%} = \frac{Area_{overlapped}}{Area_{occluder_1} + Area_{occluder_2}}$$

$$I_{overlapped,p} = Overlapped_{\%} * (I_{occluder_1,p} + I_{occluder_2,p}) \quad (14)$$

The area covered by each occluder, $Area_{occluder_i}$, is approximated as the area of the *Unit Bounding Disc* of the occluder. We define the *Unit Bounding Disc* as the disc centered along the line formed by the centre of the bounding sphere of the occluder and point p, and translated in the space (perpendicular to the surface of p), such as to be away from the point p a unit distance. The overlapping area of the two occluders $Area_{overlapped}$, is approximated by the crescent formed by the overlapping of the two *Unit Bounding Discs*, Figure 9.

4.3.2. Self shadowing

Self shadows are also computed using $FIVs$. Note that in case of self shadows, all **unoccluded** light sources, definitely lie in the positive hemisphere of the receiving pixel p. This is the main difference from the case of the shadows, where all **occluded** light sources lie in the positive hemisphere, and not the unoccluded. This is because in case of self-shadows all light sources in the negative half space of the receiving point p, are occluded at least from the surface that p lies on. Figure 10 shows with dotted red line, the part of the enviornment map the object does not self-occlude, $EnvUnocc$.

Exploiting the fact that all unoccluded light sources $EnvUnocc$ which contribute to the illumination, are in the positive hemisphere of the receiving point p, based on Equation 7, the Irradiance at a point p is equal to the Fullsphere Irradiance at that point, $I_p(\vec{N}_p) = FI_p(\vec{N}_p)$.

$FI_p(\vec{N}_p)$ is computed by deducting occluded Fullsphere Irradiance $FI_{occ,p}(\vec{N}_p)$ from the total Fullsphere Irradiance, Equation 12. $FI_{tot,p}(\vec{N}_p)$ is the same for all $\vec{N}_p$ and has been computed once and stored at the preprocessing. Thus, we don't have to do anything else than jusr use the precomputed value.

To compute $FI_{occ,p}(\vec{N}_p)$ we use the nearest sample of FIV precomputed with a dot product with the normal of the receiver $\vec{N}_p$, Equation 10. In the same way as in case of shadows, we trilinearly interpolate up to 8 nearest samples of $FIVs$. Samples that fall within the geometry of the object are not taken into account.

5. Results

The results were taken on an Intel Core 2 Duo 2.4 GHz machine, with 2GB Ram and an NVIDIA GeForce 8800 GTS graphics card. A CPU implementation was used for the pre-

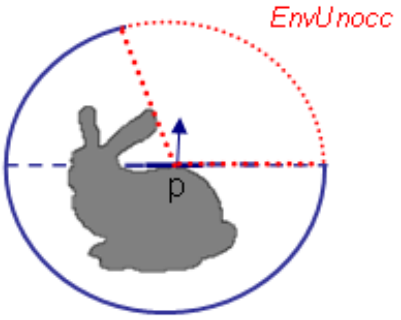

Figure 10: *In case of self shadows, all the unoccluded part of the environment EnvUnocc, always lies in the positive hemisphere of the receiving point.*

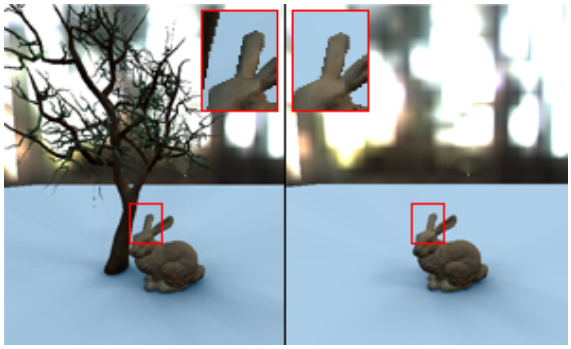

Figure 11: *An object (bunny) may act as both occluder and receiver. The bunny as a receiver (left) has shadows cast on it, in contrast to the bunny (right) which act only as an occluder.*

computations and a GPU implementation for run time calculations. All images are taken with a window resolution of 512x512, with trilinear interpolation of the FIVs values and per pixel illumination. The environment maps used for illumination are 6x32x32 pixels. The FIVs are computed considering the whole environment map and not only on samples on it. Using a higher resolution environment map would have caused longer precomputation times but no difference at the run-time frames per second.

In Table 1, we show the precomputation statistics for the individual objects used in our experiments. We demonstrate the results of objects with different complexity with regards to the number of vertices, (column *Vertices*), and their type of geometry. The *Precmp* column shows the time, in minutes, taken to precompute the FIVs. Computation of the *Diffuse Environment Irradiance Map* takes approximately about 1 minute. Here it is worth noting that our precomputation was computed entirely on CPU. A GPU implementation would run considerably faster. As expected, the table indicates that the precomputation time is linear to the number

Object	Vertices	Samples	Precmp (mins)	Memory (MB)
Bunny	35947	256x129x30	397	34.0
		128x65x30	99	8.56
		64x33x30	24	2.17
		64x33x15	12	1.08
		32x17x15	3	0.28
Tree	20614	256x129x30	222	34.0

Table 1: *Pre-computation times and memory requirements for different objects at various sampling rates.*

Scene	Objects	Vertices	Memory (MB)	FPS
1 bunny	1	35 947	8.56	112
1 tree	1	20 614	8.56	112
10 bunnies	10	359 470	8.56	68
10 trees	10	206 140	8.56	68
5 x (bunny+tree)	10	282 805	17.12	68
10 x (bunny+tree)	20	565 610	17.12	46
15 x (bunny+tree)	30	848 415	17.12	35
20 x (bunny+tree)	40	1 131 220	17.12	29
30 x (bunny+tree)	60	1 696 830	17.12	22

Table 2: *Statistics for combinations of objects for constant samples* $(\vartheta, \varphi, \rho)$ *= (128,65,30) .*

of sample points. Under the *Memory* column we show the memory usage in Megabytes. Memory needed to store the precomputations is also linear to the sample points that were used and independent of the complexity of the object. This is illustrated by the fact that any object needs the same memory capacity for constant number of samples.

Table 2 shows the statistics for combinations of different objects. The *Objects* and *Vertices* columns show the total number of objects and number of vertices in the scene. Memory needed, is increased linearly with the number of different objects in the scene but having multiple times the same object does not increase the memory requirements. Frames per second (illumination only) depend only on the number of objects but not on their type, as it is illustrated by the different scenes with 10 objects. FPS demonstrate that the proposed method, scales well and achieves real-time frame rates even for complex scenes with millions of vertices and tenths of occluders.

Note that run time frames per second (FPS), are independent of the number of sample points used, and independent of the complexity of objects. This allows us to increase the realism in the scene by using high quality models and a dense FIV sample set without additional cost at run time for the illumination process.

Figures 12 and 13 demonstrate the quality of the results of the proposed technique. The three small images at the bottom show: the shadow on the ground plane using the pro-

Figure 12: *The bunny at different sampling rates (ϑ, φ, ρ): Samples = (32,17,15) (NRMSE = 0.023) (left), Samples = (64,33,30) (NRMSE = 0.008) (middle), Samples = (256,129,30) (NRMSE = 0.007) (right). The small images show the shadow only of the result of the proposed technique (left), the ground truth shadow (middle) and the difference of the two (right).*

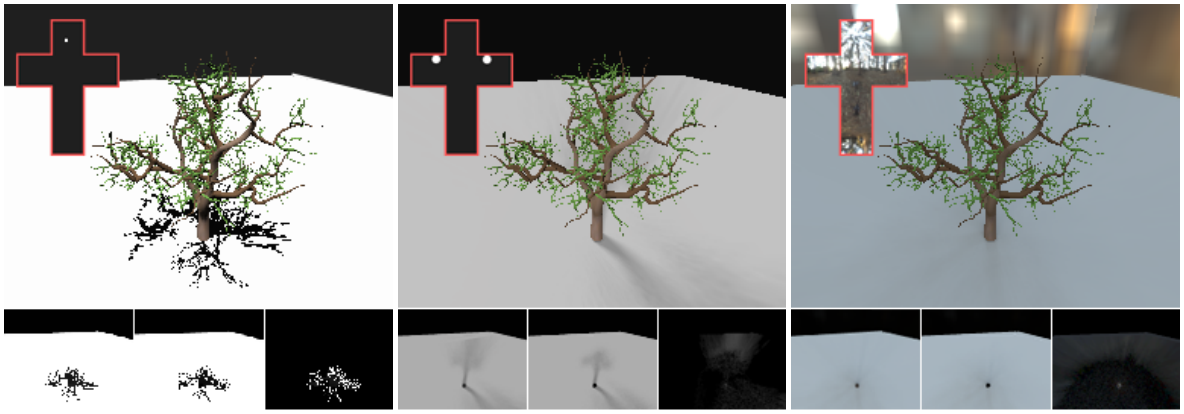

Figure 13: *All frequency environment maps can be used with our technique: one directional light (NRMSE = 0.1) (left), two area light sources (NRMSE = 0.035) (middle), Eucalyptus Grove Environment Map(NRMSE = 0.022) (right).*

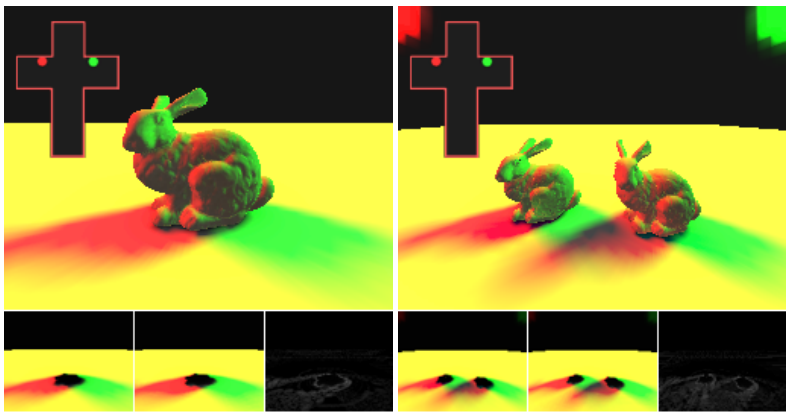

Figure 14: *Illumination from area lights of different color: one occluder (NRMSE = 0.019) (left), two occluders with overlapping shadows (NRMSE = 0.019) (right).*

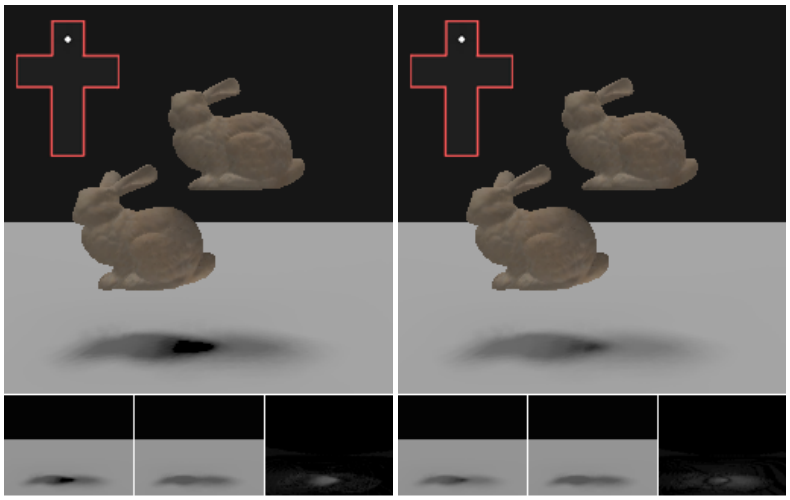

Figure 15: *Illumination of multiple occluders: Correction for multiple occluders is disabled (NRMSE = 0.027) (left), enabled (NRMSE = 0.019) (right).*

Figure 16: *Illumination of multiple occluders in different parameters: one area light source (NRMSE = 0.019) (left), Eugalyptus Grove environment map (NRMSE = 0.015) (middle), occluders in shorter distance (NRMSE = 0.013) (right).*

posed method (left), the shadow using the reference (brute force) solution (middle) and the difference of the two images (right). The difference image is displayed with high exposure in order even pixels with small error to be visible. The reference solution has been evaluated using the standard rendering formula, Equation 2, with an overall integration of the unoccluded part of the environment map as it is seen from each pixel. A numerical evaluation of the difference of the two shadow images (proposed technique and reference solution), is given using the Normalized Root Mean Square Error (NRMSE).

Figure 12, shows how the number of sample points used to precompute FIVs affect the quality of the images. As it was expected, increasing number of the sample points reduces the error. However, even with few samples, the NRMSE is only 0.023 as it is shown in Figure 12 left. Note that for this image, 3 minutes of precomputations and 0.28 MBytes of memory for the FIVs are enough for a good quality result, see Table 1.

Figure 13, demonstrates that our technique can handle all frequency environment maps; one directional light source (left), two area light sources (middle), Eucalyptus Grove environment map (right). The error (NRMSE) varies based on the type of environment map, from 0.1 in case of one direc-

Figure 18: *Self shadows are also computed at run time using precomputed FIVs. Only direct illuminatin without considering self-occlusions (left), direct illumination and self-shadows (middle), isolated self-shadow (right).*

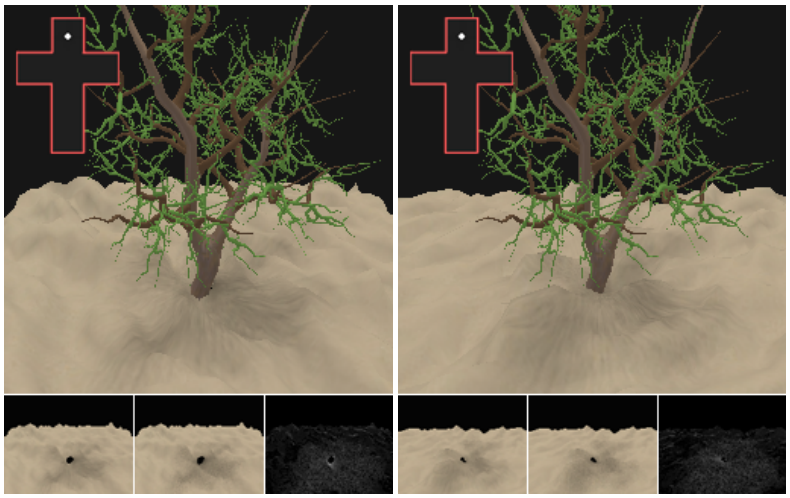

Figure 19: *Illumination of a fully dynamic (deformable) receiver in two different times in left and right images*

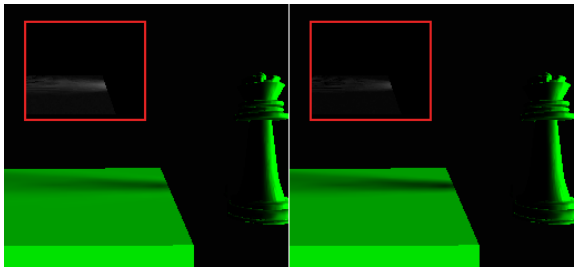

Figure 17: *The queen lies at $\frac{1}{5}$ in the negative half space of the receiver. Illumination of the receiver having the correction for partial occluders disabled (NRMSE = 0.026) (left) and enabled (NRMSE = 0.019) (right). Small images show the difference of the corresponding result with the reference solution.*

tional light to only 0.022 in case of the Eucalyptus Grove environment map. Even for the one directional light source that has the highest NRMSE (mainly because of aliasing), it is hard for a viewer to perceive the difference with the reference solution.

Figure 14, demonstrates that the algorithm works not only for symmetric (or almost symmetric) environment maps. Figure 14, left, shows the shadow of one occluder of an an environment map with one green and one red area light sources, and Figure 14, right, shows overlapping shadows from two occluders using the same environment map.

The same object may act both as an occluder and as a receiver at the same time, see the bunny in Figure 11, left. A receiver may be shadowed by more than one occluders. Note that in Figure 11 left, the plane is shadowed by the tree and the bunny.

Figure 17 demonstrates that our method works also well for the cases that the occluder does not lie fully in the pos-

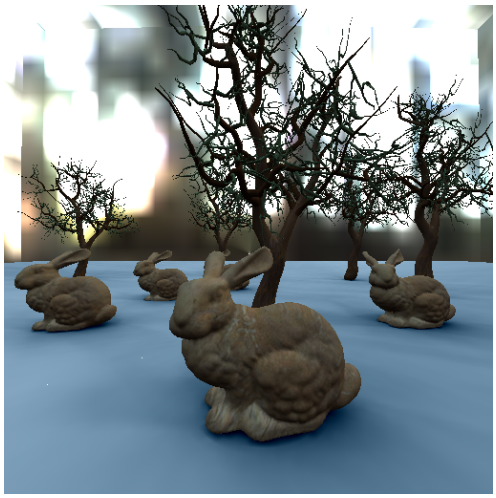

Figure 20: *Complex scenes can be shaded in real-time. This scene with multiple objects and more than 280K thousands vertices runs at 68 fps.*

itive hemisphere of the receiver. Figure 17, left, shows the results using the standard proposed algorithm that uses the values of FIVs as they precomputed. The right image of the Figure 17, shows the results using the modified algorithm for the partial occluder as described in the relevant subsection. In the latter case the error is minimized to only 0.019.

Multiple occluders are also handled by our proposed technique. Figure 15 shows that the error is minimized when the common occluded part by the overlapped occluders, is taken into account only once (correction enabled). Results, for different conditions in the scene with multiple occluders, are demononstrated in the Figure 16.

Self-shadows result are demonstrated in the Figure 18. The image on the left shows illumination without considering any self-occlusions, the image in the middle is the illuminated object with self-shadows, and the image at the right is the difference of the two previous, showing the shadow isolated from the object.

6. Conclusions and Future Work

The proposed method can compute diffuse illumination with soft shadows for fully dynamic receivers, Figure 19, moving occluders and all frequency environment maps at real-time frame rates. Frame rate and memory requirements are independent of the number of light sources and the number of vertices of the occluders. Frame rate depends only on the number of occluders in the scene while the memory requirements depends on the number of sample points used and the different type of objects in the scene. The size and the complexity of the occluders do not affect the computations for illumination at run time. Our technique requires only moderate memory and relative fast precomputations. Our method can run real-time for very complex scenes, Figure 20, with low measured error, Figures 12, 13, 16 & 17.

The main contribution of our technique is the factorization of a new notion that we introduce, the *Fullsphere Irradiance*. The factorization allows us to precompute and store in only a 3D color vector, the contribution to illumination from an arbitrary number of light sources on a reference base system, without requiring the receiver's normal in advance. The precomputed values can be transformed, in a very fast way (using only a dot product), to the Irradiance arriving from the light sources, once the receiver and it's normal is known.

The main limitation of our technique stems from the fact that the precomputed values of FIVs encode the environment map information, rendering our technique applicable only for scenes with static light sources.

Another limitation is that occluders can only be moved in the space, but they can not rotate, since the placement at sample points of the occluder at the preprocessing, define the occluded light sources which their contribution is encoded within the FIV. This limitation can be eliminated by taking samples, not only at different positions of each occluder, but with different rotations as well. However, this solution will make the precomputation more expensive and increase the memory requirements of the algorithm.

The proposed method currently only accounts for diffuse reflectance. Consideration of the specular reflection component would increase the realism of the illumination. However, it does not seem to be a straight forward way to do that using only the factorization, so an extension to the algorithm should be developed to count for specularities.

Another way that the algorithm can be improved is the elimination of precomputations. Online computations of FIVs can be computed only at samples needed (i.e. a receiving point exists), reducing the total time needed to compute the FIVs. Using a GPU implementation we speculate that computations of FIVs at run time, at least at interactive rates, is feasible. For run time computation of the *Diffuse Irradiance Environment Map* the proposed technique of [RH01] can be used. By having no precomputations, we will be able to have fully dynamic conditions such as deformable occluders and dynamic lighting conditions.

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 Transactions on Graphics 27*, 3 (Aug. 2008), 34:1–34:8.

[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 SIGGRAPH 98* (July 1998),

Computer Graphics Proceedings, Annual Conference Series, pp. 189–198.

[DSDD07] DACHSBACHER C., STAMMINGER M., DRETTAKIS G., DURAND F.: Implicit visibility and antiradiance for interactive global illumination. *ACM Transactions on Graphics (SIGGRAPH Conference Proceedings) 26*, 3 (August 2007), 61:1–61:10.

[Eve00] EVERITT C. W.: *High-Quality, Hardware-Accelerated Per-Pixel Illumination for Consumer Class OpenGL Hardware*. Master's thesis, Mississippi State University, Mississippi State, Mississippi, May 2000.

[GBP06] GUENNEBAUD G., BARTHE L., PAULIN M.: Real-time soft shadow mapping by backprojection. In *Eurographics Symposium on Rendering (EGSR), Nicosia, Cyprus, 26/06/2006-28/06/2006* (http://www.eg.org/, 2006), Eurographics, pp. 227–234.

[HLHS03] HASENFRATZ J., LAPIERRE M., HOLZSCHUCH N., SILLION F.: A survey of real-time soft shadows algorithms. *Computer Graphics Forum 22*, 4 (2003), 753–774.

[IDYN07] IWASAKI K., DOBASHI Y., YOSHIMOTO F., NISHITA T.: Precomputed radiance transfer for dynamic scenes taking into account light interreflection. In *Rendering Techniques 2007: Eurographics Symposium on Rendering* (june 2007), pp. 35–44.

[KK03] KOLLIG T., KELLER A.: Efficient illumination by high dynamic range images. In *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering* (June 2003), pp. 45–51.

[KL05] KONTKANEN J., LAINE S.: Ambient occlusion fields. In *SI3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2005), ACM Press, pp. 41–48.

[KLA04] KAUTZ J., LEHTINEN J., AILA T.: Hemispherical rasterization for self-shadowing of dynamic objects. In *Rendering Techniques 2004: 15th Eurographics Workshop on Rendering* (June 2004), pp. 179–184.

[KSS02] KAUTZ J., SLOAN P.-P., SNYDER J.: Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *Rendering Techniques 2002: 13th Eurographics Workshop on Rendering* (June 2002), pp. 291–296.

[LLAP05] LECOT G., LEVY B., ALONSO L., PAUL J.-C.: Master-element vector irradiance for large tessellated models. In *GRAPHITE '05: Proceedings of the 3rd international conference on Comp uter graphics and interactive techniques in Australasia and South East Asia* (2005).

[LSSS04] LIU X., SLOAN P.-P., SHUM H.-Y., SNYDER J.: All-frequency precomputed radiance transfer for glossy objects. In *Rendering Techniques 2004: 15th Eurographics Workshop on Rendering* (June 2004), pp. 337–344.

[MH84] MILLER G. S., HOFFMAN C. R.: Illumination and reflection maps: Simulated objects in simulated and real environments. *Course Notes for Advances Computer Graphics Animation, Siggraph 1984* (1984).

[MMAH07] MALMER M., MALMER F., ASSARSSON U., HOLZSCHUCH N.: Fast precomputed ambient occlusion for proximity shadows. *Journal of Graphics Tools 12*, 2 (2007), 57–71.

[MSW04] MEI C., SHI J., WU F.: Rendering with Spherical Radiance Transport Maps. *Computer Graphics Forum 23*, 3 (2004), 281–290.

[NRH04] NG R., RAMAMOORTHI R., HANRAHAN P.: Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics (TOG) 23*, 3 (2004), 477–487.

[OCL96] OUHYOUNG M., CHUANG Y., LIANG R.: Reusable radiosity objects. *Computer Graphics Forum 15*, 3 (1996), 347–356.

[RH01] RAMAMOORTHI R., HANRAHAN P.: An efficient representation for irradiance environment maps. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), ACM, pp. 497–500.

[RWS*06] REN Z., WANG R., SNYDER J., ZHOU K., LIU X., SUN B., SLOAN P.-P., BAO H., PENG Q., GUO B.: Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *ACM Transactions on Graphics 25*, 3 (July 2006), 977–986.

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics 21*, 3 (July 2002), 527–536.

[Slo06] SLOAN P.-P.: Normal mapping for precomputed radiance transfer. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2006), ACM Press, pp. 23–26.

[SM06] SUN W., MUKHERJEE A.: Generalized wavelet product integral for rendering dynamic glossy objects. *International Conference on Computer Graphics and Interactive Techniques* (2006), 955–966.

[SS08] SCHWARZ M., STAMMINGER M.: Quality scalability of soft shadow mapping. In *GI '08: Proceedings of graphics interface 2008* (Toronto, Ont., Canada, Canada, 2008), Canadian Information Processing Society, pp. 147–154.

[WHG99] WILLMOTT A. J., HECKBERT P. S., GARLAND M.: Face cluster radiosity. In *IN EUROGRAPHICS WORKSHOP ON RENDERING* (1999), Springer, pp. 293–304.

[Wil00] WILLMOTT A. J.: *Hierarchical Radiosity with Multiresolution Meshes*. PhD thesis, Computer Science, Carnegie Mellon Univerity, 2000.

[ZHL*05] ZHOU K., HU Y., LIN S., GUO B., SHUM H.-Y.: Precomputed shadow fields for dynamic scenes. *ACM Transactions on Graphics 24*, 3 (Aug. 2005), 1196–1201.

Appendix A: Proofs from Section 4.1

Proof of the Equation 11:

M^+ : The set of all light sources in the scene in the positive hemisphere of the receiver

M^+_{occ} : The set of occluded light sources in the positive hemisphere of the receiver

M^+_{unn} : The set of unoccluded light sources in the positive hemisphere of the receiver

$$M^+ = M^+_{occ} \cup M^+_{unn}, \qquad M^+_{occ} \cap M^+_{unn} = \emptyset$$

$$I_{tot,p} = \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i) \cos < \vec{N}_p, \vec{\omega}_i >)$$

$$I_{occ,p} = \sum_{\vec{\omega}_i \in M^+_{occ}} (L_{in}(\vec{\omega}_i) \cos < \vec{N}_p, \vec{\omega}_i >)$$

Proof:

$$\begin{aligned} I_p(\vec{N}_p) &= \sum_{\vec{\omega}_i \in \{M^+ - M^+_{occ}\}} (L_{in}(\vec{\omega}_i) \cos < \vec{N}_p, \vec{\omega}_i >) \\ &= \sum_{\vec{\omega}_i \in \{M^+ - M^+_{occ}\}} (L_{in}(\vec{\omega}_i) \cos < \vec{N}_p, \vec{\omega}_i >) \\ &= \sum_{\vec{\omega}_i \in M^+} (L_{in}(\vec{\omega}_i) \cos < \vec{N}_p, \vec{\omega}_i >) \\ &\quad - \sum_{\vec{\omega}_i \in M^+_{occ}} (L_{in}(\vec{\omega}_i) \cos < \vec{N}_p, \vec{\omega}_i >) \\ I_p(\vec{N}_p) &= I_{tot,p}(\vec{N}_p) - I_{occ,p}(\vec{N}_p) \end{aligned}$$

Proof of the Equation 12:
Note that using the following:

M : The set of all light sources in the scene in both hemispheres of the receiver

M_{occ} : The set of occluded light sources in both hemispheres of the receive

M_{unn} : The set of unoccluded light sources in both hemispheres of the receiver

$$M = M_{occ} \cup M_{unn}, \qquad M_{occ} \cap M_{unn} = \emptyset$$

we can prove in a similar way we used to prove Equation 11, that:

$$FI_p(\vec{N}_p) = FI_{tot,p}(\vec{N}_p) - FI_{occ,p}(\vec{N}_p)$$