

# Partial, Multi-scale Precomputed Radiance Transfer

Tamás Umenhoffer  
TU Budapest

László Szirmay-Kalos  
TU Budapest

László Szécsi  
TU Budapest

Balázs Tóth  
TU Budapest

Mateu Sbert  
U of Girona

## Abstract

This paper presents a real-time global illumination method for quasi-static scenes illuminated by arbitrary, dynamic light sources. In order to maintain real-time frame-rates, the light transport problem is decomposed to low-frequency and high-frequency parts. The part that has low-frequency characteristics both in the temporal and in the spatial domains is pre-computed. Due to the low-frequency properties this solution can be compressed by clustered PCA without significant reduction of the accuracy. The spatial high-frequency but temporal low-frequency parts are pre-computed on small-scale using a modified ambient occlusion/obscurances method. The temporally high-frequency components are obtained on the fly. This way we can combine the advantages of pre-computation aided methods and real-time rendering.

## 1 Introduction

The light transport problem can be formalized by equation

$$L(\vec{x}, \vec{\omega}) = \mathcal{T}(L^e(\vec{y}, \vec{\omega}'))$$

where  $L$  is the radiance at point  $\vec{x}$  in outgoing direction  $\vec{\omega}$ ,  $L^e$  is the emission of the light sources at point  $\vec{y}$  in direction  $\vec{\omega}'$ , and  $\mathcal{T}$  is the linear light transport operator, which includes both direct transport  $\mathcal{T}_d$  and indirect transport  $\mathcal{T}_i$ . Since the domain of radiance  $L$  and emission  $L^e$  is an infinite set, the transport operators are infinite dimensional. Pre-computation aided methods pre-compute and store transfer function  $\mathcal{T}$ , which is feasible only for finite dimensional operators. To reduce the dimensionality to finite dimensions, radiance functions are either expressed by *finite element* series or computed only at discrete points and directions and are interpolated for other points and directions. The sample points where the radiance is evaluated are called *exit points*. There are three popular choices for the selection of exit points:

- pixels of the generated image,
- vertices of the mesh (*per-vertex approach*),
- points corresponding to the texel centers of a texture atlas (*per-texel approach*).

Assigning the exit points to pixels is feasible only if the camera is static, which is usually not the case. Per-vertex representations constrain the density of sample points by the tessellation level, which limits the freedom of the modeler. Thus, we prefer the per-texel representation, which is the most flexible. It also has a drawback, namely the requirement of creating an appropriate texture atlas for the scene.

Having the pre-computed transfer operator, during the on-line part of rendering, emission  $L^e$  is set according to the actual lighting, and the finite element coefficients or the radiance of the exit points are obtained by a vector-matrix multiplication. For other visible points, finite element reconstruction or interpolation is used. Bilinear interpolation of the per-vertex or per-texel representation is especially effective since it is directly supported by the graphics hardware.

These approaches have two critical problems, the required storage is usually huge and the assumption on a static scene is invalid. The light transport operator is infinite dimensional, and even its approximate representation requires very many discrete values. On the other hand, pre-computation is feasible if scene parameters do not change in time, which is a serious limitation of these approaches. To reduce the required storage, we may

- reduce the dimensionality of lighting configuration taking several simplifying assumptions, including, for example, that the scene is diffuse or the lighting is only directional,
- reduce the number of exit points,
- compress the transport matrix.

The reduction of the number of exit points as well as the compression of the transport matrix mean that less data and interpolation should provide enough accuracy. This is feasible if these parameters have low-dimensionality or are of low-frequency in the spatial domain.

## 2 Previous work

Several approaches have emerged that limit the freedom of object changes in order to make global illumination computations fast. These approaches can be classified from different points of view. We may distinguish them according to whether they handle large-scale or small-scale light transfers. Large-scale or global approaches transfer light between even the farthest points of the scene, but to reduce the complexity, they also assume that local parts have homogeneous properties. Small-scale or local methods transfer light only in the neighborhood, but cope with the high variance of geometric and optical parameters. Pre-computation aided methods can also be categorized according to the variables of the transfer functions they pre-compute, i.e. whether these transfer functions connect light sources, directions, patches, points, texels, pixels, etc. Finally, we may distinguish methods that aim at the pre-computation of the full transfer, or only the indirect lighting.

### 2.1 Large-scale methods

If the scene is static, then radiance transfer coefficients between surface elements do not change. This has been recognized even in the context of finite-element based radiosity algorithms, where form factors could be computed only once. The price of this is that the required storage is a quadratic function of the number of patches. The idea has been further generalized to include arbitrary number of specular reflections with the introduction of *extended form factors* [Sillion and Puech 1994]. Fixing the light sources, the radiosity solution can be stored in a texture map, called the *light map*. Instead of the transfer functions, here usually per-texel irradiance is stored.

*Pre-computed radiance transfer* (PRT) [Sloan et al. 2002; Sloan et al. 2003] is a method to realistically render rigid bodies assuming that a single object is illuminated by low-frequency hemispherical image based lighting (i.e. it computes *light direction to vertex radiance* or *light direction to texel radiance* transfer functions). Its basic idea is that the incident illumination and reflected lighting func-

tions are approximated by a finite element series, and thus lighting is expressed by vectors of coefficients multiplying the predefined basis functions (usually by spherical harmonics or wavelets). Then, during rendering, whenever the environment lighting changes, the reflected radiance can be quickly updated. The original idea has been improved in many different ways, including arbitrary BRDFs [Kautz et al. 2002], speeding up both preprocessing and rendering phases [Lehtinen and Kautz 2003], the application of principal component analysis to compress data [Sloan et al. 2003], the extension for closer lights [Annen et al. 2004], and the replacement of spherical harmonics by wavelets to allow high frequency environment maps as well [Ng et al. 2003].

To extend PRT to local lights, Kristensen et al. [Kristensen et al. 2005] discretized the incident lighting into a set of localized point lights (i.e. they used *3D light position to texel* transfer functions). The main problem of this approach is that it further increases the dimensionality of the PRT solution, which is already quite high.

Mei et al [Mei et al. 2004] have proposed a solution to render both shadows and self illumination. Their method is based on *Spherical Radiance Transfer Maps* that have to be pre-computed for every vertex and hundreds of sampled directions. It is also assumed that lights are distant. The main caveat of the Spherical Shadow Map method is that shadow maps are rendered for the vertices. Then, when the illumination of a vertex is computed, sampled directions are tested against this map to see whether the environment map is occluded in that direction. This setup requires a large number of shadow maps, and could result in inaccurate per-pixel results, depending on the tessellation.

Radiance transfer precomputation has also been applied to translucent objects [Lensch et al. 2003], where the *vertex to vertex radiosity* transfer factors are stored and the response to an illumination is obtained by a finite element series using these factors as weights. Connecting the transfer factors to the vertices poses similar problems as in spherical transfer maps. The storage requirement is a quadratic function of the number of vertices, which prohibits complex or highly tessellated objects and requires well-shaped patches.

*Light path reuse* [Sbert et al. 2004] is a Monte Carlo technique to speed up animated light sequences assuming that the objects and the camera are still. Under these circumstances, *light hit to pixel* transfer function obtained in a frame may also be used in other frames if the ray between the light point and the first hit point is not invalidated by new occlusions. The combination of all paths in a single frame is governed by multiple importance sampling guaranteeing that the variance of the solution in a frame is close to what could be obtained if we dedicated all paths solely to this particular frame.

The *light path maps* method obtained the transfer function between uniformly sampled random points on the surface and points corresponding to texel centers [Szécsi et al. 2006], which is stored in a texture atlas. During real-time rendering, the direct illumination is obtained separately, and this texture atlas is taken advantage of to speed up the indirect illumination calculation. Note that this step requires visibility information between the light sources and entry points, but this is already available in the shadow maps obtained during direct illumination computation [Dachsbacher and Stamminger 2005].

## 2.2 Small-scale approaches

Large-scale analysis does not allow very detailed surfaces since they would increase the complexity that should be considered at once beyond the possibilities of real-time rendering. Large-scale analysis is thus executed for the *mean surface* that is a simplified

mesh ignoring mesostructure details like bumps and displacements. On the other hand, when detailed geometry is inspected, we may ignore distant parts of the scene and concentrate just on the local neighborhood since the illumination influence of a unit area surface diminishes with the square of the distance, thus distant points might be replaced by some “average”. These methods usually examine the openness of points of the detailed surface. A point is said to be “open” in directions where the local neighborhood is not visible from it. In open directions an *ambient* illumination is assumed, which represents the “rest”. In not open directions the ambient light is assumed to be blocked.

According to these considerations, the reflected indirect illumination is approximated as

$$L^{ind}(\vec{x}) = a(\vec{x})W(\vec{x})L^a(\vec{x}), \quad (1)$$

where  $a(\vec{x})$  is the *albedo* of the point,  $L^a(\vec{x})$  is the direction independent ambient light intensity around  $\vec{x}$ , and  $W(\vec{x})$  is a scalar factor in  $[0, 1]$  that expresses how open point  $\vec{x}$  is. Scalar factor  $W(\vec{x})$  is called the *obscurance* [Zhukov et al. 1998; Iones et al. 2003] of point  $\vec{x}$ , and is defined as:

$$W(\vec{x}) = \frac{1}{\pi} \int_{\Omega'} \rho(d(\vec{x}, \vec{\omega}')) \cos^+ \theta_x' d\omega' \quad (2)$$

where  $\Omega'$  is the set of all directions,  $\rho(d)$  is in  $[0, 1]$  and is a fuzzy measure of openness depending on distance  $d$  of the visible surface from  $\vec{x}$  at direction  $\vec{\omega}'$ , and  $\theta_x'$  is the angle between this direction and the surface normal at  $\vec{x}$ . Superscript  $+$  indicates that the negative cosine values should be replaced by zero. Function  $\rho(d)$  increases with distance  $d$  allowing the ambient light to take effect if no occluder can be found nearby. If we use a clear distinction between open and closed directions, i.e. if  $\rho$  is zero when distance  $d$  is smaller than a threshold and 1 otherwise, the obscurance value is called the *ambient occlusion* [Pharr and Green 2004]:

$$W(\vec{x}) = \frac{1}{\pi} \int_{\mathcal{O}} \cos^+ \theta_x' d\omega' \quad (3)$$

where  $\mathcal{O}$  is the set of open directions.

To account for color bleeding, the obscurances formula has been modified to include the diffuse BRDF of points visible from  $\vec{x}$  [Mendez et al. 2005]:

$$W(\vec{x}) = \int_{\Omega'} \frac{f_r(\vec{y}(\vec{x}, \vec{\omega}'))}{\bar{a}} \rho(d(\vec{x}, \vec{\omega}')) \cos^+ \theta' d\omega' \quad (4)$$

where  $\vec{y}(\vec{x}, \vec{\omega}')$  is the point visible from  $\vec{x}$  at direction  $\vec{\omega}'$  and  $\bar{a}$  is the average albedo in the scene. This modification introduces nice color bleeding effects, but loses the advantage of obscurances that only the local neighborhood needs to be processed.

Research efforts aiming at including meso-structure properties into material data are also related to this paper since these methods also store self-illumination capabilities compactly and combine them with the actual lighting. These methods are different extensions of texture maps, such as *bi-directional texture functions* [Dana et al. 1997; Suykens et al. 2003], or *polynomial texture maps* [Malzbender et al. 2001], that are usually obtained by a measuring process. The self-shadowing of meso-structure features can also be pre-computed and used in interactive rendering [Heidrich et al. 2000].

Combining small-scale and large-scale pre-computed approaches has been proposed for light maps. The basic problem of light maps that they store the irradiance of the mean surface, which cannot take

into account the high-frequency variations of the normal vector and the detailed surface. The solution of this problem is to store the incoming radiance field, usually at a few sampled directions, and to obtain the reflected radiance on the fly [McTaggart 2004; Sloan 2006]. This approach can also be combined with ambient occlusion [Green 2007].

### 3 The new method

The method proposed by this paper builds upon previous work and combines their results in a novel way. According to frequency space behavior [Durand et al. 2005] the light transfer is decomposed to high-frequency direct lighting and indirect lighting that consists of high-frequency light source to surface transfer, low-frequency mean surface to mean surface transfer, high-frequency mean surface to detailed surface transfer, and finally high-frequency detailed surface to eye transfer.

Let us first examine the possibility of this decomposition problem formally. The global illumination rendering is equivalent to the evaluation of high dimensional integrals. Thus, from a mathematical point of view, the decomposition corresponds to factorization of integrals. Let us consider a simple one-variate integral of the product of two functions:

$$I = \frac{1}{T} \int_0^T f(x)g(x,y) dx \quad (5)$$

In order to consider the frequency space behavior of functions  $f$  and  $g$ , let us mirror them onto the  $x = 0$  line, and repeat it with periodicity  $2T$ . These periodic functions are even and thus can be expressed by Fourier series:

$$f(x) = \sum_{i=0}^{\infty} a_i \cos \frac{i\pi x}{T}$$

where

$$a_0 = \frac{1}{T} \int_0^T f(x) dx, \quad a_i = \frac{2}{T} \int_0^T f(x) \cos \frac{i\pi x}{T} dx.$$

Similarly

$$g(x,y) = \sum_{i=0}^{\infty} b_i(y) \cos \frac{i\pi x}{T}$$

where

$$b_0(\vec{y}) = \frac{1}{T} \int_0^T g(x,y) dx, \quad b_i(\vec{y}) = \frac{2}{T} \int_0^T g(x,y) \cos \frac{i\pi x}{T} dx.$$

Exploiting the orthogonality of the cosine series, the integral of equation 5 can also be written as:

$$I = \frac{1}{T} \int_0^T f(x)g(x,y) dx = \sum_{i=0}^{\infty} a_i b_i(y).$$

If apart from the zero frequency term,  $f$  and  $g$  are not dominant for the same frequencies, i.e. where  $f$  is significant,  $g$  is small, and vice

versa, then  $a_i b_i(y)$  ( $i > 0$ ) can be neglected. Thus we can use the following approximation

$$I = \sum_{i=0}^{\infty} a_i b_i(y) \approx a_0 b_0(y) = \frac{1}{T} \int_0^T f(x) dx \cdot \frac{1}{T} \int_0^T g(x,y) dx.$$

Thus, if we can identify factors in an integrand that correspond to different frequency ranges, then the integral of their product can be approximated by the product of their integrals.

#### 3.1 Decomposition of the lighting problem

A typical game or virtual reality scene consists of a static environment (e.g. city, terrain, building, etc.) and of dynamic objects (e.g. enemy character, vehicle, weapon, etc.), where the static environment is usually much larger. The direct illumination of the static environment and of dynamic objects is strongly affected by the movement of dynamic objects, thus this is a temporally high-frequency component. The direct illumination is computed on the fly using shadow mapping. Indirect illumination of the static environment is built from light paths starting at the light source and including more than one scattering point either on dynamic or on static objects. If the light sources may move and may get close to the surfaces, then the intensity of the first hit points has high-frequency characteristics both temporally and spatially. However, due to the size differences of dynamic and static objects, the irradiance of paths arriving at a static surface area is roughly independent of the dynamic objects and is smooth. However, the reflected radiance caused by the average irradiance has also spatially high-frequency behavior since the variation of the normal vector and displacement mapped surfaces significantly modify the smooth irradiance. Finally the indirect illumination of the smaller dynamic objects is also of high-frequency in the temporal domain, so it should be approximated on the fly.

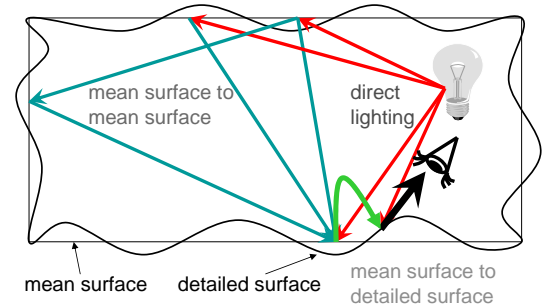


Figure 1: Decomposition of the lighting problem to direct illumination and indirect illumination, which is further decomposed to direct lighting, mean surface to mean surface transfer, and mean surface to detailed surface transfer.

Based on the recognition of the possibility of factoring integrals and on the observations made on the typical game scenes, we propose the application of pre-computation only for low-frequency parts of the light transfer problem. The pre-computation is executed on two scales. The rest is computed on the fly.

The lighting problem is decomposed to several components (Figure 1). We identify the *direct illumination*, which includes the lighting of the detailed surface points visible from the camera, and also the lighting of the mean surface points that will be the first hits of longer light paths responsible for indirect lighting. As a part of the indirect illumination, we distinguish the *mean surface to mean surface transfer*, which connects hits of the rays emitted by the light

sources and other points assuming only the mean surface, i.e. only simplified geometry. These connections may involve arbitrary number of multiple reflections. Finally, we also identify a *mean surface to detailed surface transfer* that describes the modulation of the average incoming radiance arriving at the hypothetical mean surface by the local properties of the detailed surface. Finally the direct illumination is added and the radiance reflected toward the eye is obtained.

The low-frequency mean surface to mean surface transfer is pre-computed between sampled entry points and points corresponding to the texel centers of a texture atlas, and the results are compressed by clustered PCA. We propose a novel sampling scheme for entry points that select them independently of the mesh complexity and also takes into account the a-priori knowledge of possible lighting configurations.

The small scale transfer is simulated by a modified obscurances method, which — unlike ambient occlusion and scalar obscurances — also takes into account the spectral properties of materials in the local neighborhood, but — unlike the spectral obscurances — still preserves the advantage of the local processing.

The decomposition of the light transfer from light source to mean surface, mean surface to mean surface, and mean surface to detailed surface factors is possible because of the different frequency characteristics of these phenomena.

Concerning the temporal behavior of these transfers, those factors that are based on single rays are significantly affected by the movement of dynamic objects. Thus, these factors, including the light source to mean surface and the detailed surface to eye transfers, must be refreshed in each frame. However, those factors that are composed by connections of many light paths of arbitrary number of steps are rather invariant to the movement of dynamic objects. Even if some rays are invalidated by a moving objects, the other connection paths keep the transfer function from changing significantly. These factors, including the mean surface to mean surface transfer, and the mean surface to detailed surface transfer, can be assumed to be static and pre-computed.

### 3.2 Large-scale low-frequency transfer

To handle large-scale low-frequency transfer we use a modified light path maps method where the small-scale transfers are eliminated in the last step, the entry points are generated in a special way, and the transfer matrices are compressed by clustered PCA. The preprocessing step determines the indirect illumination capabilities of the static scene. This information is computed for finite number of *exit points* on the surface, and we use interpolation for other points. Exit points can be defined as points corresponding to the texel centers of the texture atlas of the surface, and the interpolation is provided by the bi-linear texture filtering feature of the graphics hardware.

The pre-processing algorithm consists of the following steps:

- Generation of entry points on the surface.
- Computation of entry point to exit point transfer factors using the virtual light sources algorithm, which results in a texture of transfer factors for each entry point.
- Clustering entry points that are close and are on similarly oriented surface, and PCA compression for each cluster.

#### 3.2.1 Entry point generation

The first step of preprocessing is the generation of certain number of *entry points* on the surface with some density  $d(\vec{e})$ . These entry points will be samples of first hits of the light emitted by moving light sources. Entry points are assumed to have unit irradiance and are used as the start of a given number of random light paths.

The simplest way to place entry points on surfaces is to sample them from uniform distribution. On the other hand, if we have some a-priori knowledge of the possible future lighting, then we can build this information into this probability. In games, we usually do have such information, for example, we know that the light source is flying close to the ceiling or is at the hand of a moving avatar.

Initially we place  $M$  test points  $\vec{Y}_1, \dots, \vec{Y}_M$  in the 3D space to sample the domain of possible light positions. These sample points can be placed manually, or obtained randomly from a distribution in 3D. From each test point  $\vec{Y}_m$  we sample  $N_t$  number of random or quasi-random directions from a uniform density  $p_m(\omega)$  which mimics the light source radiance. The test point with the sampled direction defines a ray, which is traced to look for a surface intersection. The found surface point will be entry point  $\vec{e}$ . The probability that we hit the differential area  $de$  by a ray originating at test point  $\vec{Y}_m$  is

$$p_m(\omega_{\vec{Y}_m \rightarrow \vec{e}}) \cdot \frac{de \cdot \cos \theta_e^{in}}{|\vec{Y}_m - \vec{e}|^2} \cdot v(\vec{Y}_m, \vec{e}),$$

where  $\theta_e^{in}$  is the angle between the surface normal at  $\vec{e}$  and the direction of  $\vec{Y}_m$  from  $\vec{e}$ , and  $v$  is the binary visibility function.

If we shoot  $N_t$  rays from each test point  $\vec{Y}_m$ , then the density of entry points  $\vec{e}$  is

$$d(\vec{e}) = N_t \cdot \sum_{m=1}^M p_m(\omega_{\vec{Y}_m \rightarrow \vec{e}}) \cdot \frac{\cos \theta_e^{in}}{|\vec{Y}_m - \vec{e}|^2} \cdot v(\vec{Y}_m, \vec{e}).$$

#### 3.2.2 Transfer function computation

The transfer function between entry points and *exit points* corresponding to texel centers is computed by random walk. Random light paths are initiated from the entry point according to BRDF sampling and Russian roulette, and the visited points of the generated paths are considered as *virtual light sources* [Keller 1997; Wald et al. 2002] and are connected to all those exit points that are visible from them. In this way we obtain a lot of paths originating at an entry point and arriving at one of the exit points.

The contribution of a path assuming unit irradiance at the entry point, divided by the probability of the path generation and of the entry point sampling, is a Monte-Carlo estimate of the indirect illumination. The average of the Monte-Carlo estimates of paths associated with the same entry point and exit point pair is stored.

A fundamental problem of the virtual light source algorithm (also called as *instant radiosity* or *indirect photon mapping*) is that the length of connection rays might be small, which leads to spiky artifacts at corners. In our multi-scale approach the small-scale transfer is handled by another method, thus when implementing the virtual light sources algorithm short connection rays are ignored. This eliminates the double consideration of light transfer paths and also significantly improves importance sampling.

For a given entry point, the collection of exit points is a texture map, thus this data structure, called the *light path map*, or *LPM* for short, can be defined by as many textures as entry points we have (Figure 2).



Figure 2: A few tiles of an LPM texture used to render the image above. Illumination corresponding to clusters of entry points is stored in tiled atlases.

### 3.2.3 LPM compression

The fundamental problem of pre-computation aided global illumination algorithms is that they require considerable memory space to store the transfer function coefficients. To cope with the memory requirements, data compression methods should be applied. A popular approach is the *principal component analysis (PCA)* [Sloan et al. 2003]. The transfer function can be imagined as separate  $N \times M$  resolution textures for the  $E$  entry points:  $\mathbf{T}_1[u, v], \dots, \mathbf{T}_E[u, v]$ . These textures can be imagined as  $E$  number of points in an  $N \times M$ -dimensional space. To compress this data according to clustered PCA, we first cluster those entry points that are close to each other and are on similarly oriented surface, then apply PCA compression separately for each cluster. PCA finds a subspace (a “hyperplane”) in the high-dimensional space and projects the  $e$ th point  $\mathbf{T}_e$  into the basis of its cluster  $c(e)$ . Since the subspace has lower dimensionality, the projected points can be expressed by fewer coordinates, which results in data compression.

Let us denote the means and the basis vectors of cluster  $c$  by  $\mathbf{M}^c$ , and  $\mathbf{B}_1^c, \dots, \mathbf{B}_D^c$ , respectively. Each entry point  $e$  belongs to exactly one cluster, which is denoted by  $c(e)$ . Projecting into this subspace means the following approximation:

$$\mathbf{T}_e \approx \tilde{\mathbf{T}}_e = \mathbf{M}^{c(e)} + w_1^e \mathbf{B}_1^{c(e)} + \dots + w_D^e \mathbf{B}_D^{c(e)},$$

where  $w_1^e, w_2^e, \dots, w_D^e$  are the coordinates of the projected point in the subspace coordinate system. Orthogonal projection results in the following approximating coordinates:

$$w_j^e = (\mathbf{T}_e - \mathbf{M}^{c(e)}) \cdot \mathbf{B}_j^{c(e)}.$$

Origin  $\mathbf{M}^c[u, v]$  is the average texture of the cluster, and basis vectors  $\mathbf{B}_1^c[u, v], \dots, \mathbf{B}_D^c[u, v]$  are the *eigen-textures*. They must be selected to minimize the total approximation error, i.e. the sum of the square distances between the original and projected points. As can be shown, the error is minimal if the origin is the mean of the original data points and the basis vectors are the eigenvectors cor-

responding to the largest  $D$  eigenvalues of the *covariance matrix*

$$\sum_{e \text{ in cluster } c} (\mathbf{T}_e - \mathbf{M}^c)^T \cdot (\mathbf{T}_e - \mathbf{M}^c),$$

where superscript  $T$  denotes transpose operation (a row vector is turned to be a column vector). Intuitively, this error minimization process corresponds to finding a minimal ellipsoid that encloses the original points, and obtaining the center of this ellipsoid as  $\mathbf{M}^c$  and the longest axes of the ellipsoid to define the required hyperplane. Indeed, if the remaining axes of the ellipsoid are small, then the ellipsoid is flat and can be well approximated by the plane of other axes.

When processing the Escher staircase scene (Figure 2) we used just  $128 \times 128$  resolution textures on each object, and the 4096 entry points were organized in 32 clusters. Surprisingly small, i.e. at most 2 basis vectors were enough to accurately represent these clusters, which shows that the self-illumination capabilities of the scene belong to low-frequency phenomena.

### 3.3 Small-scale pre-computation

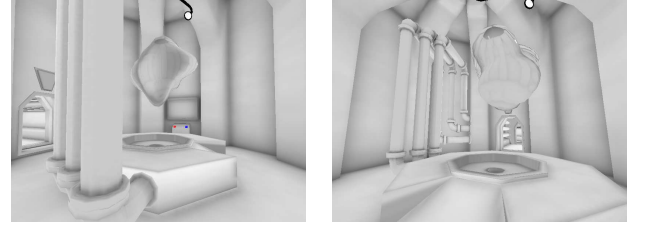


Figure 3: The Space Station rendered with scalar obscurances only.

To simulate small scale transfer, we first tried both the scalar obscurances and the ambient occlusion methods (Figure 3). The problem of scalar obscurances and ambient occlusion is that these methods ignore local color bleeding. The spectral obscurances method does simulate color bleeding but for the price of global scene processing. In order to simulate low scale color bleeding but to keep the advantages of local processing, we propose another obscurances formula that decomposes the directional domain into two parts, an “open” part where the visible points are farther than a threshold, and a “closed” part where the visible point is closer. The sizes of these directional domains are denoted by  $\mathcal{O}$  and  $\mathcal{C}$ , respectively. Similarly to assuming that from far points an average ambient lighting is coming, we assume that the BRDF of far points is replaced by the average BRDF. Using this, the obscurances formula has the following form:

$$W(\vec{x}) = \frac{1}{\pi} \int_{\mathcal{O}} \cos^+ \theta' d\omega' + \int_{\mathcal{C}} \frac{f_r(\vec{y}(\vec{x}, \vec{\omega}'))}{\tilde{a}} \rho(d(\vec{x}, \vec{\omega}')) \cos^+ \theta' d\omega'.$$

The first term is the classical ambient occlusion, and the second is responsible for color bleeding. Both terms can be obtained by processing only the local neighborhood. We define three such equations on the wavelengths of red, green, and blue. Note that the ratio of the BRDF and the average albedo might be 0/0 fractions, which must be replaced by 0 for the correct interpretation.





Figure 4: An image rendered with only direct illumination (left), with only the proposed obscuration formula weighted with direct lighting (middle), and with both direct and approximated indirect illumination (right).

### 3.4 Rendering

Rendering requires the computation of direct lighting and indirect lighting. Indirect lighting is the total contribution of complete light paths of length at least two, including the light source to mean surface transfer, the mean surface to mean surface transfer, the mean surface to detailed surface transfer, and finally the detailed surface to eye transfer. To obtain direct lighting variance shadow mapping is implemented [Donnelly and Lauritzen 2006]. The shadow map is also used to determine the visibility of the entry points.

Indirect lighting is an integral in path space containing paths of lengths at least 2. This integral is factorized into three integrals.

The first integral represents the irradiance  $I_e$  at the entry point caused by a light source of area  $\Delta l$ . This integral represents the light source to mean surface transfer, and can be obtained in the following form:

$$I_e = \int_{\Delta l} L_e \cdot v(\vec{l}, \vec{e}) \cdot \frac{\cos \theta_l \cdot \cos \theta_e}{|\vec{l} - \vec{e}|^2} dl \approx L_e \cdot v(\vec{l}_0, \vec{e}) \cdot \frac{\cos \theta_{l_0} \cdot \cos \theta_e}{|\vec{l}_0 - \vec{e}|^2 + \Delta l / \pi}$$

where  $\vec{l}_0$  is the center of the light source,  $\theta_l$  is the angle between the light's normal and the direction from light source point  $\vec{l}$  and entry point  $\vec{e}$ , and  $\theta_e^{in}$  is the angle between the surface normal at the entry point and the illumination direction.

The second factorized integral is responsible for transferring the irradiance of the entry point to all those points of the scene that correspond to texel centers. Recall that  $\tilde{\mathbf{T}}_e[u, v]$  is the texture of irradiance values caused by setting unit irradiance at entry point  $e$ . These texture values should be weighted by the real irradiance of the respective entry point, and the weighted textures should be added. The total contribution of every entry point to the irradiance at exit point  $[u, v]$  is

$$I[u, v] = \sum_{e=1}^E I_e \tilde{\mathbf{T}}_e[u, v] = \sum_{e=1}^E I_e \mathbf{M}^{(e)}[u, v] + \sum_{e=1}^E \sum_{d=1}^D I_e w_d^e \mathbf{B}_d^{(e)}[u, v].$$

Finally, the irradiance at the exit point is transferred to the eye, taking into account the detailed surface properties. When the scene is rendered from the camera, the direct illumination is evaluated. Simultaneously two textures are fetched, the irradiance texture and the modified obscuration texture. The latter has much higher resolution and might also be a periodic texture. To include the mean surface to local surface transfer, the two texture values are multiplied. This transfer represents the indirect illumination where the

length of the last ray is smaller than the threshold between small-scale and large-scale transfers. To also include indirect illumination of paths of length 2, where the last ray belongs to the small-scale category, the obscuration value are multiplied by the direct illumination. Finally, the direct illumination and indirect illumination terms are added.

Dynamic objects should be rendered by on-the-fly techniques. To take into account the indirect illumination of the static environment on dynamic objects, we used localized cube map based approaches proposed in [Szirmay-Kalos et al. 2005; Szirmay-Kalos and Lazányi 2006].

## 4 Results

The discussed method has been implemented and integrated into the Ogre3D game engine, and used in various demo games. Figure 4 demonstrates the power of low-scale pre-computed transfer. When rendering this scene we set the threshold to be equal to the size of the scene, i.e. all indirect transfers were handled only by the obscuration approach. Figures 5 and 6 show the Space Station game. Here we used the same threshold as when rendering Figure 3. Figure 6 also compares the proposed real-time global illumination method with classical local illumination rendering. Figure 7 shows close ups and an overview of the Moria game. All these games run at 30 FPS on NV7900 GPUs.



Figure 5: The Space Station scene.

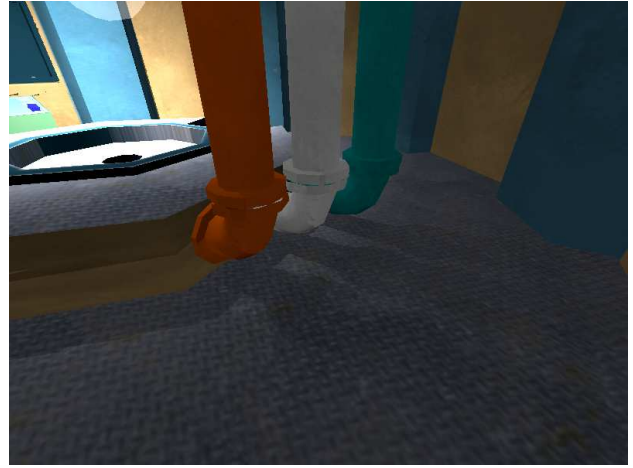
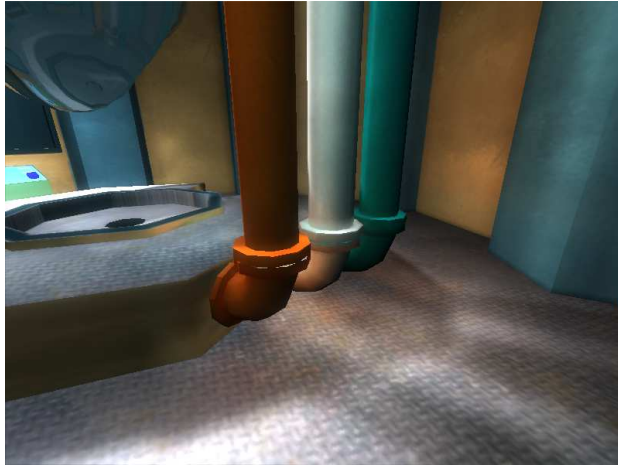


Figure 6: Indirect illumination computed by the light path map method in the Space Station game (left) and the same scene rendered with local illumination (right) for comparison. Note the color bleeding effects.

## 5 Conclusions

This paper proposed a factoring method to present global illumination renderings in games and real-time systems. The basic idea is to decompose lighting phenomena according to their spatial and temporal frequency characteristics, and pre-compute low-frequency components. This way, global illumination effects can be presented in real-time systems, although approximately but with no disturbing visual artifacts.

## Acknowledgement

This work has been supported by the National Office for Research and Technology and by OTKA K-719922 (Hungary), and by the GameTools FP6 (IST-2-004363) project.

## References

- ANNEN, T., KAUTZ, J., DURAND, F., AND SEIDEL, H.-P. 2004. Spherical harmonic gradients for mid-range illumination. In *Eurographics Symposium on Rendering*.
- DACHSBACHER, C., AND STAMMINGER, M. 2005. Reflective shadow maps. In *SI3D '05: Proc. of the 2005 Symp. on Interactive 3D Graphics and Games*, 203–231.
- DANA, K., GINNEKEN, B., NAYAR, S., AND KOENDERINK, J. 1997. Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics* 18, 1, 1–34.
- DONNELLY, W., AND LAURITZEN, A. 2006. Variance shadow maps. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, 161–165.
- DURAND, F., HOLZSCHUCH, N., SOLER, C., CHAN, E., AND SILLION, F. X. 2005. A frequency analysis of light transport. In *SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, 1115–1126.
- GREEN, C. 2007. Efficient self-shadowed radiosity normal mapping. In *ACM SIGGRAPH 2007. Course 28: Advanced real-time rendering in 3D graphics and games*, 1–8.
- HEIDRICH, W., DAUBERT, K., KAUTZ, J., AND SEIDEL, H.-P. 2000. Illuminating micro geometry based on precomputed visibility. In *SIGGRAPH 2000 Proceedings*, 455–464.
- IONES, A., KRUPKIN, A., SBERT, M., AND ZHUKOV, S. 2003. Fast realistic lighting for video games. *IEEE Computer Graphics and Applications* 23, 3, 54–64.
- KAUTZ, J., SLOAN, P., AND SNYDER, J. 2002. Fast, arbitrary BRDF shading for low-frequency lighting using spherical harmonics. In *12th EG Workshop on Rendering*, 301–308.
- KELLER, A. 1997. Instant radiosity. In *SIGGRAPH '97 Proceedings*, 49–55.
- KRISTENSEN, A. W., AKENINE-MOLLER, T., AND JENSEN, H. W. 2005. Precomputed local radiance transfer for real-time lighting design. In *SIGGRAPH 2005*.
- LEHTINEN, J., AND KAUTZ, J. 2003. Matrix radiance transfer. In *SI3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, 59–64.
- LENSCH, H., GOESELE, M., BEKAERT, P., KAUTZ, J., MAGNOR, M., LANG, J., AND SEIDEL, H.-P. 2003. Interactive rendering of translucent objects. *Computer Graphics Forum* 22, 2, 195–195.
- MALZBENDER, T., GELB, D., AND WOLTERS, H. 2001. Polynomial texture maps. In *SIGGRAPH 2001 Proceedings*, 519–528.
- MCTAGGART, G. 2004. Half-life 2 shading. In *GDC Direct3D Tutorial*.
- MEI, C., SHI, J., AND WU, F. 2004. Rendering with spherical radiance transport maps. *Computer Graphics Forum (Eurographics 04)* 23, 3, 281–290.
- MENDEZ, A., SBERT, M., CATA, J., SUNYER, N., AND FUNTANE, S. 2005. Real-time obscurances with color bleeding. In *ShaderX4: Advanced Rendering Techniques*, W. Engel, Ed. Charles River Media.

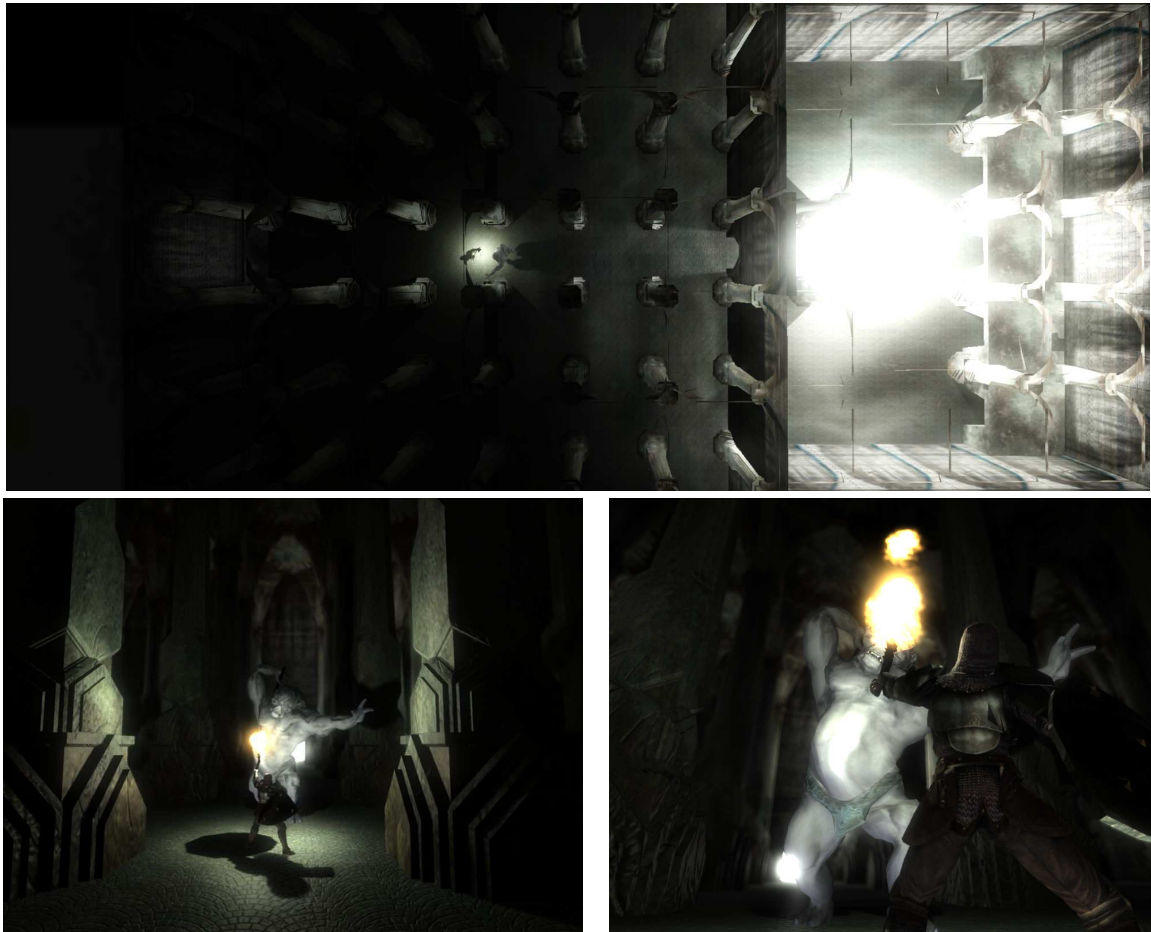


Figure 7: A knight and a troll are fighting in Moria illuminated by the light path map algorithm. The upper image is a bird's eye view of the Moria scene. Note that the hall on the right is illuminated by a very bright light source. The light enters the hall on the left after multiple reflections. The troll and the knight with his burning torch can be observed in the middle of the dark hall.

- NG, R., RAMAMOORTHY, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph.* 22, 3, 376–381.
- PHARR, M., AND GREEN, S. 2004. Ambient occlusion. In *GPU Gems*. Addison-Wesley, 279–292.
- SBERT, M., SZÉCSI, L., AND SZIRMAY-KALOS, L. 2004. Real-time light animation. *Computer Graphics Forum (Eurographics 04)* 23, 3, 291–300.
- SILLION, F., AND PUECH, C. 1994. *Radiosity and Global Illumination*. Morgan Kaufmann Publishers, Inc., San Francisco.
- SLOAN, P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH 2002 Proceedings*, 527–536.
- SLOAN, P., HALL, J., HART, J., AND SNYDER, J. 2003. Clustered principal components for precomputed radiance transfer. In *SIGGRAPH 2003*.
- SLOAN, P.-P. 2006. Normal mapping for precomputed radiance transfer. In *13D 06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, 23–26.
- SUYKENS, F., BERGE, K., LAGAE, A., AND DUTRÉ, P. 2003. Interactive rendering with bidirectional texture functions. *Computer Graphics Forum (Eurographics 03)* 22, 3, 464–472.
- SZÉCSI, L., SZIRMAY-KALOS, L., AND SBERT, M. 2006. Light animation with precomputed light paths on the GPU. In *GI 2006 Proceedings*, 187–194.
- SZIRMAY-KALOS, L., AND LAZÁNYI, I. 2006. Indirect diffuse and glossy illumination on the GPU. In *SCCG 2006*, 29–35.
- SZIRMAY-KALOS, L., ASZÓDI, B., LAZÁNYI, I., AND PREMECZ, M. 2005. Approximate ray-tracing on the GPU with distance impostors. *Computer Graphics Forum (Eurographics '05)* 24, 3, 695–704.
- WALD, I., KOLLIG, T., BENTHIN, C., KELLER, A., AND SLUS-  
SALEK, P. 2002. Interactive global illumination using fast ray tracing. In *13th Eurographics Workshop on Rendering*, 15–24.
- ZHUKOV, S., IONES, A., AND KRONIN, G. 1998. An ambient light illumination model. In *Proceedings of the Eurographics Rendering Workshop*, 45–56.