Image-Based Lighting on Surfaces with Arbitrary BRDF

An Exercise in BRDF Importance Sampling, Sampling from a 2D Discrete Probability Function, and Multiple Importance Sampling

Jaroslav Křivánek
Jaroslav.Krivanek@mff.cuni.cz

November 13, 2011

1 Introduction

Our task is to compute direct illumination due to an environment map (EM) at a point $\mathbf{x}$ on a surface. The environment map acts as a source of radiance $L(\omega)$ from directions on the unit sphere. We assume that the light is arriving from infinity, so the radiance has no position dependency. Formally, our task is equivalent to evaluating the *illumination integral* in the following form:

$$L_o(\mathbf{x}, \omega_o) = \int_S L(\omega_i)\ V(\mathbf{x}, \omega_i)\ f_r(\mathbf{x}, \omega_i, \omega_o)\ \cos\theta_i\, \mathrm{d}\omega_i, \tag{1}$$

where we integrate over the unit sphere S. Unlike in the more general task of sampling indirect illumination, the incoming radiance $L(\omega_i)$ is known (it is given by the environment map). The visibility function $V(\mathbf{x}, \omega_i)$ equals 1 if an infinite ray from $\mathbf{x}$ in direction ω_i is not blocked by scene geometry, and 0 otherwise.

We are focusing only on one surface point $\mathbf{x}$ observed from one outgoing direction ω_o so let us fix these two variables and drop them along with the subscript i from ω_i to get the illumination integral in a form that reads little better:

$$L_o = \int_S L(\omega)\ V(\omega)\ f_r(\omega)\ \cos\theta\, \mathrm{d}\omega. \tag{2}$$

We will use the usual Monte Carlo quadrature to evaluate this integral:

$$\hat{L}_o = \frac{1}{N} \sum_{j=1}^{N} \frac{f(\Omega_j)}{p(\Omega_j)} \tag{3}$$

Here $f(\omega) = L(\omega)V(\omega)f_r(\omega)\cos\theta$ is the integrand and Ω_j is a random direction drawn from a distribution with probability density (PDF) p. How to choose the PDF and how to sample from it will be discussed in the remainder of this write-up.

Since both the BRDF and the EM can have very sharp peaks, our integrand, given by the product of the two, can vary pretty wildly. When estimating integrals of "wild" functions using MC quadrature, it is essential to apply importance sampling with a carefully chosen probability density function (PDF), in order to keep the variance low. Our goal is to choose a PDF proportional to the integrand $L(\omega)V(\omega)f_r(\omega)\cos\theta$. It is not trivial to take the (à priory unknown) visibility function into account when constructing the PDF, so let us ignore it. There are methods for sampling from the product $L(\omega) f_r(\omega) \cos\theta$, but these are quite involved and go well beyond the scope of this write-up. Our simpler strategy will be to use the following two sampling PDFs:

1. PDF roughly proportional to $f_r(\omega)\cos\theta$ (i.e. the usual BRDF importance sampling)

2. PDF *exactly* proportional to $L(\omega)$ (i.e. sampling from the environment map)

The BRDF importance sampling is useful for very sharp BRDFs (high Phong exponent), because in this case the shape of the integrand is dominated by the BRDE lobe. The EM sampling, on the other hand, is useful when the BRDF is rather diffuse and the environment map spiky. But none of the above strategies works well for all surface roughness values (e.g. exponent in the Phong BRDF model) and all environment maps (uniform to spiky). So we will take advantage of *Multiple Importance Sampling* to combine the two sampling methods in such a way that the advantages of the two strategies are preserved.

2 Evaluating the illumination integral using BRDF-proportional importance sampling

The first ingredient we need is a procedure for sampling from a given BRDF. As an example, I describe how to sample from the modified Phong BRDF, given by the following equation:

$$f_r(\omega_i, \omega_o) = \begin{cases} \dfrac{\rho_d}{\pi} + \rho_s \dfrac{n+2}{2\pi} \cos^n \theta_s & \text{if } \omega_i.z > 0 \text{ and } \omega_o.z > 0 \text{ (i.e. above tanglent plane)} \\ 0 & \text{otherwise (on and below the tangent plane)} \end{cases} \tag{4}$$

where ρ_d is the diffuse reflectivity (a.k.a. albedo) between 0 and 1, ρ_s is the specular reflectivity between 0 and 1, n is the specular exponent (a.k.a. shininess) and θ_s is the angle between ω_o and the perfect mirror reflection of ω_i. In a RGB renderer, the values of ρ_d and ρ_s will be RGB triples (in a spectral renderer, they will be longer vectors). $\frac{n+2}{2\pi}$ is a normalizing factor introduced so that the total energy reflected by the specular component is independent of the exponent n (which is not the case in the classical Phong shading model, which does not have any such normalization).

The sampling of the BRDF proceeds in two stages: first, we choose the BRDF component to sample from (i.e. are we sampling from the diffuse or the specular lobe?) and then then we generate the direction from the chosen component.

2.1 Choosing the BRDF component to sample from

We will use the values of ρ_d and ρ_s (converted to scalars) to choose the component. This makes sense: we want to sample most of the rays from the component that reflects most of the energy. Here's how we do it:

1. Generate a random number ξ_0 from R(0, $\rho_d + \rho_s$) (i.e. uniformly distributed between 0 and $\rho_d + \rho_s$).

2. If $\xi_0 < \rho_d$ then sample from the diffuse component. Otherwise, sample form the specular component.

2.2 Sampling from the diffuse component

Our goal is to sample from $f_r(\omega_i, \omega_o)\cos\theta_i$ (i.e. from the cosine-weighted BRDF). The diffuse component of the BRDF is constant so we want our PDF to be proportional to $\cos\theta_i$. To make the PDF integrate to unity over the hemisphere, we introduce a normalizing factor $1/\pi$, so the sampling PDF is:

$$p_d(\omega) = \frac{\cos\theta}{\pi}. \tag{5}$$

To sample from this PDF, we generate two random numbers ξ_1, ξ_2 from $R(0,1)$ and transform them into a direction in spherical coordinates using the following formulas:

$$\phi = 2\pi\xi_1 \tag{6}$$

$$\cos\theta = \sqrt{\xi_2} \tag{7}$$

For practical calculations we need the Cartesian coordinates:

$$x = \cos\phi\sin\theta \tag{8}$$

$$y = \sin\phi\sin\theta \tag{9}$$

$$z = \cos\theta. \tag{10}$$

Note that the generated direction is defined with respect to the local coordinate frame, the z-axis of which is aligned with the surface normal at $\mathbf{x}$. The direction has to be transformed into the world (global) coordinate frame before being used in a ray tracer.

2.3 Sampling from the specular component

In the specular case, there is no easy way to sample from $f_r(\omega_i, \omega_o)\cos\theta_i$ so we will only sample from $f_r(\omega_i, \omega_o)$. The specular term is given by a cosine lobe centered around ω_s (the mirrored ω_i direction), but unlike in the diffuse case, the lobe is raised to n. The corresponding PDF is:

$$p_s(\omega) = \frac{n+1}{2\pi}\cos^n\theta. \tag{11}$$

Similar to the diffuse case, we sample from this PDF by generating two random numbers ξ_1, ξ_2 from $R(0,1)$ and transforming them into the direction. This time the transformation formulas read:

$$\phi = 2\pi\xi_1 \tag{12}$$

$$\cos\theta = \xi_2^{\frac{1}{n+1}} \tag{13}$$

Note that this direction is defined with respect to a coordinate frame where the z-axis is aligned with the mirror reflection direction ω_s and has to be transformed to an appropriate frame for further calculations. The direction generated from the specular component may end up below the tangent plane of the surface. Such a sample is valid - only the BRDF value for it is zero.

2.4 Completing the BRDF sampling procedure

In practical implementations, the BRDF sampling procedure usually returns the sampled direction, the PDF for that direction, and the BRDF value for that direction. One possibility would be to evaluate and return the total BRDF value no matter which BRDF component we sampled from. This is a very,

very bad idea, though. Here is why: Suppose that we sample the diffuse component (pretty uniform over the hemisphere) but the direction happenes to "hit" the narrow specular lobe. So the total BRDF value for this sample is very high, but it is not compensated in the Monte Carlo estimator by dividing by a large PDF value (remember the sample came from the rather uniform diffuse PDF). As a result, we have generated a sample with disproportionately large contribution to the estimator (3), which directly translates into high variance of the estimator (and a lot of noise in our image). Clearly, that's not what we want. But we can be smarter:

1. If we used the diffuse lobe sampling to sample the BRDF, return $\frac{\rho_d}{\pi}$ as the BRDF value and $\frac{\rho_d}{\rho_d+\rho_s} p_d(\omega)$ as the PDF.
2. If we used the specular lobe sampling to sample the BRDF, return $\rho_s \frac{n+2}{2\pi} \cos^n \theta_s$ as the BRDF value and $\frac{\rho_s}{\rho_d+\rho_s} p_s(\omega)$ as the PDF.

This means that the discrete decision we made at the beginning wasn't only about which BRDF component to sample from, but we actually randomly chose the BRDF component which will be evaluated. Since the diffuse component has only $\frac{\rho_d}{\rho_d+\rho_s}$ chance of being selected, the corresponding PDF is "boosted" by this value. Similar argument holds for the specular component. The discussion in this subsection generalizes to other BRDF models than Phong.

2.5 Implementation of the estimator based on BRDF sampling

```
Lo := 0;
for ( j = 1; j <= N; j++ )
{
  [wj, p(wj), fr(wj)] := sampleBRDF();
  if ( fr(wj) == 0 ) continue; // sample has zero contribution
  L(wj) := lookUpEM(wj);
  V(wj) := castRay(wj);
  if ( V(wj) == 0 ) continue; // direction is blocked => reject
  Lo += L(wj) * fr(wj) * cos(thetaj) / p(wj);
}
Lo := Lo / N;
```

3 Evaluating the illumination integral using EM-proportional importance sampling

This time the sampling PDF is given by the luminance of the environment map. The question is how to sample from a PDF given by an image (that actually represents a function defined on the unit sphere).

3.1 A small digression: Sampling from a 1D discrete probability

One of the subroutines that we need for sampling from the environment map is drawing a random variable I from a univariate (1D) discrete probability distribution described by the probability mass function $p(i) = \Pr(I = i)$, where i are integers between 0 and $n - 1$. As a matter of fact, we don't even require that the individual probabilities sum to one, so we can call $p(i)$ a "discrete importance" function. In the first step, done in preprocess, we construct the corresponding cumulative distribution function (CDF):

$$F(i) = \sum_{k=0}^{i} p(k) \tag{14}$$

To generate a random variable I distributed according to $p(i)$, we generate a random number ξ from $R(0, F(n-1))$. The generated random variable equals to the first index i for which $\xi < F(i)$. (In practice, this search is implemented by bisection.)

3.2 Loading the EM

The EM usually comes in the form of a high dynamic range (HDR) image, where each pixel of the image corresponds to a direction on the unit sphere. The mapping used for the light probes from Paul Debevec's web site (http://ict.debevec.org/ debevec/Probes/) is described on the same page. For a given direction dir (with Cartesian coordinates $(dir.x, dir.y, dir.z)$), the following code snippet gives the corresponding normalized (u, v) coordinate in the image. This conversion is the basis of the implementation of the `lookUpEM()` function.

```
float d = sqrt(dir.x*dir.x + dir.y*dir.y);
float r = d>0 ? 0.159154943*acos(dir.z)/d : 0.0;
u = 0.5 + dir.x * r;
v = 0.5 + dir.y * r;
```

3.3 Converting EM into a PDF

For the sake of comfortable sampling, we will use the latitude-longitude mapping for the PDF instead of original Debevec's mapping. In this mapping, the horizontal index j (running from 0 to $w-1$, where w is the image width) corresponds to ϕ in spherical coordinates (running from 0 to 2π over the width of the image). Similarly, the vertical index i (running from 0 on the top of the image to $w-1$ at the bottom) corresponds to θ in spherical coordinates (running from 0 to π).

In the preprocess (after loading the image) we convert the luminance from the input EM into this representation and normalize it so that it integrates to unity over the full sphere. The normalization factor is given by the inverse of the integral of the luminance over the sphere, which can be computed while creating the PDF map. In this way, we have defined a piece-wise constant PDF over the directions on unit sphere, which is constant over the area of each pixel in our PDF map. Now we need a procedure to sample from such a PDF.

3.4 Sampling from the PDF map in latitude-longitude mapping

The sampling procedure will proceed in two stages: First, we discretely pick the pixel of the PDF map to sample from, and then uniformly select a direction from within this pixel. The discrete probability for choosing one pixel is given by the integral of the PDF over the area of the pixel. This is simply given by the area of that pixel on the sphere (integral over the pixel boundaries in spherical coordinates) multiplied by the (constant) PDF value for that pixel. The joint discrete probability for choosing pixel (i, j) is given by the probability mass function denoted $p(i, j)$.

Choosing the pixel. To sample form the bivariate (2D) probability mass function $p(i, j)$ one option is to put all tuples (i, j) in a long 1D vector, and sample from the vector as if the probability distribution was univariate (1D). However, this is usually not optimal with respect to the distribution of the resulting samples (e.g. when applying stratified sampling or low-discrepancy random sequences). It is better to sample $p(i, j)$ as a true 2D probability. The procedure is as follows:

1. Pick the row index I from the marginal probability mass function $p_I(i) = \sum_{j=0}^{w-1} p(i,j)$.

2. Given the row I, pick the column J from the conditional probability $p_J(j|I=i) = \frac{p(i,j)}{p_I(i)}$

In other words, sampling from the 2D probability is reduced to sampling from two 1D probabilities: one marginal $p_I(i)$ and the other conditional $p_J(j|I=i)$. The division by $p_I(i)$ in the definition of the conditional probability is actually never needed since our 1D sampling procedure tolerates "probabilities" that do not sum to one.

Uniform sampling inside the pixel area. For the practical purposes, it is usually sufficient to return the direction corresponding the center of the pixel we have selected using the procedure described above. However, that's not a clean solution in the mathematical sense, since our PDF is piecewise constant, and every direction inside the pixel should have an equal chance of being sampled. One way to approximate this behavior would be to reformulate the procedure for sampling from a 1D discrete distribution to accept a piecewise constant PDF (instead of a probability mass function). The corresponding CDF would be piece-wise linear and could be inverted using a simple modification of the discrete sampling procedure described earlier.

3.5 Implementation of the estimator based on EM sampling

```
Lo := 0;
for ( j = 1; j <= N; j++ )
{
  [wj, p(wj)] := sampleEM();
  fr(wj) := evaluateBRDF(wj, wo);
  if( fr(wj) == 0 ) continue; // zero contribution (wrong side of the surface etc.)
  L(wj) := lookUpEM(wj);
  V(wj) := castRay(wj);
  if ( V(wj) == 0 ) continue; // direction blocked => reject
  Lo += L(wj) * fr(wj) * cos(thetaj) / p(wj);
}
Lo := Lo / N;
```

4 Combined estimator using Multiple Importance Sampling

As mentioned at the beginning of the write-up, none of the sampling methods described so far (BRDF and EM importance sampling) works well over the entire range of BRDFs and environment maps. Usually, one has low variance when the other has high variance. We want to combine the two sampling techniques in a way that preserves their strengths. We use Veach and Guilbas' Multiple Importance Sampling for that purpose.

4.1 Multiple Importance Sampling

Assume our task is to compute the integral

$$I = \int_D f(\mathbf{x}) \mathrm{d}\mathbf{x} \tag{15}$$

over some domain D and we have n different sampling techniques (i.e. PDFs) $p_1(\mathbf{x}), p_2(\mathbf{x}), \ldots, p_n(\mathbf{x})$ for importance sampling. We define a *combined estimator*:

$$F = \sum_{i=1}^{n} \frac{1}{n_i} \sum_{j=1}^{n_i} w_i(X_{i,j}) \frac{f(X_{i,j})}{p_i(X_{i,j})}, \tag{16}$$

where n_i is the number of samples drawn from the i-th technique (PDF), and $X_{i,j}$ is the j-th sample drawn from the i-th technique and $w_i(\mathbf{x})$ are the weighting functions (one per sampling technique) that we use to combine the samples from individual techniques. Note that the weight is a function of $\mathbf{x}$, i.e. different strategies can get different weight in different parts of the domain. This estimator is unbiased if $\sum_{i=1}^{n} w_i(\mathbf{x}) = 1$ whenever $f(\mathbf{x}) \neq 0$, which is a hard requirement on the weighting functions. Apart from this requirement, we are free to choose any weighting function we like. Of course, we want to make a good choice, and a good choice means low variance of F. Veach and Guibas prove that the following choice, called *balance heuristic*, produces a low-variance estimator:

$$\hat{w}_i(\mathbf{x}) = \frac{n_i p_i(\mathbf{x})}{\sum_k n_k p_k(\mathbf{x})} \tag{17}$$

This is how the combined estimator looks like with the balance heuristic:

$$F = \sum_{i=1}^{n} \sum_{j=1}^{n_i} \frac{f(X_{i,j})}{\sum_k n_k p_k(X_{i,j})}, \tag{18}$$

The name "balance heuristic" stems from the fact that the contribution of each sample to the sum in the above estimator is independent of the sampling technique it came from.

4.2 Using MIS to combine BRDF and EM importance sampling

We apply MIS to combine the BRDF and EM sampling as follows:

- the integration domain is the unit sphere
- the integration variable $\mathbf{x}$ are directions over the sphere
- the integrand is $L(\omega)V(\omega)f_r(\omega)\cos\theta$
- BRDF importance sampling is the first sampling technique (the corresponding PDF is denoted p_1)
- EM importance sampling is the second sampling technique (the corresponding PDF is denoted p_2)
- the number of samples from both strategies is the same, and is given by a user-defined parameter N, i.e. $n_1 = n_2 = N$

The implementation follows the double sum in the combined estimator (16). In the actual implementation, the outer sum over the sampling techniques is "unrolled".

```
Lo := 0;

// technique 1: BRDF importance sampling
for ( j = 1; j <= N; j++ )
```

```
{
  [wj, p1(wj), fr(wj)] := sampleBrdf();
  if ( fr(wj) <= 0 ) continue;
  L(wj) := lookUpEM(wj);
  V(wj) := castRay(wj);
  if ( V(wj) == 0 ) continue;
  w = misWeight( p1(wj), p2(wj) ); // evaluate the balance (or any other) heuristic
  Lo += L(wj) * fr(wj) * cos(thetaj) * w / p1(wj);
}

// technique 2: EM importance sampling
for ( j = 1; j <= N; j++ )
{
  [wj, p2(wj)] := sampleEM();
  fr(wj) := evaluateBRDF(wj, wo);
  p1(wj) := evaluateTotalPDF(wj, wo);
  if( fr(wj) == 0 ) continue;
  L(wj) := lookUpEM(wj);
  V(wj) := castRay(wj);
  if ( V(wj) == 0 ) continue;
  w = misWeight( p2(wj), p1(wj) ); // evaluate the balance (or any other) heuristic
  Lo += L(wj) * fr(wj) * cos(thetaj) * w / p2(wj);
}
```

The total PDF for BRDF sampling is given by the "blend" of the PDFs for the individual components of the BRDF, because we used a probabilistic selection of the BRDF component to sample from:

$$p_1(\omega) = \frac{\rho_d}{\rho_d + \rho_s} p_d(\omega) + \frac{\rho_s}{\rho_d + \rho_s} p_s(\omega). \tag{19}$$

(See Section 2 on BRDF importance sampling for the explanation of the individual terms.)