Instant Radiosity

Alexander Keller*

Universität Kaiserslautern

Abstract

We present a fundamental procedure for instant rendering from the radiance equation. Operating directly on the textured scene description, the very efficient and simple algorithm produces photorealistic images without any finite element kernel or solution discretization of the underlying integral equation. Rendering rates of a few seconds are obtained by exploiting graphics hardware, the deterministic technique of the quasi-random walk for the solution of the global illumination problem, and the new method of jittered low discrepancy sampling.

CR Categories: I.3.3 [Computer Graphics]: Picture/ Image Generation—Antialiasing| Bitmap and framebuffer operations| Display algorithms| Viewing algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation| Color, shading, shadowing, and texture| Radiosity

Keywords: Radiance equation, radiosity, shading, Monte Carlo integration, quasi-Monte Carlo integration, quasi-random walk, jittered low discrepancy sampling, hardware, accumulation buffer, realtime rendering algorithms, photorealism.

1 Introduction

Provided a realistic scene description, rendering from the radiance integral equation [Kaj86] yields realistic images. Under the assumption of diffuse reflection, the most popular approaches to approximate the solution of the Fredholm integral equation are radiosity algorithms. In the classical algorithms [CW93], the kernel of the radiance integral equation is projected onto some finite base, yielding the form factor matrix which is of quadratic order in the number of scene elements. For its sparse representation, hierarchical methods with hierarchical base functions have been introduced. Nevertheless, these Galerkin algorithms need to store the kernel and solution discretization of the integral equation. In addition to the high complexity of accurate mesh generation for shadow representation [LTG92], such projections introduce a discretization error.

From the domain of Monte Carlo simulation, algorithms without kernel discretization are available, using the random integration scheme for only projecting the solution onto a finite base. Similar to the random approaches, a deterministic particle simulation scheme based on low discrepancy sampling has been introduced in [Kel96b]. This deterministic scheme converges smoother at a slightly superior rate and exposes no variance as compared to stochastic algorithms. In bidirectional path tracing [LW93, VG94], even the discretization of the solution of the radiance equation has been avoided, but the rendering time is far from realtime.

On the other hand, graphics hardware is capable of illuminating and shadowing textured scenes by extended light sources [HA90, Hei91, SKvW+92] in realtime.

In our new approach we combine the advantages of deterministic particle simulation of light, i.e. the quasi-random walk principle, with the available hardware capabilities to consistently render from the radiance equation, neither using a kernel nor an intermediate solution projection of the integral equation, resulting in a very fast, robust and straightforward to implement procedure.

Following this introduction, in the second section of this paper we briefly resume the mathematical model of the global illumination problem. The third section explains the new rendering procedure and its underlying techniques of quasi-Monte Carlo integration and the quasi-random walk principle. After pointing out some extensions of the basic algorithm for including antialiasing by jittered low discrepancy sampling, specular effects, and modifications for realtime application in section four, the algorithm is discussed in the fifth section. The final section draws the conclusion and points out directions of future research.

2 Global Illumination

Our eyes perceive radiance, which is power per unit area per unit solid angle. In vacuum the radiance L fulfills the *radiance equation* [Kaj86]

$$L(y,\vec{\omega}_r) = L_e(y,\vec{\omega}_r) + \int_\Omega f_r(\vec{\omega}_i,y,\vec{\omega}_r)L(h(y,\vec{\omega}_i),-\vec{\omega}_i)\cos\theta_i d\omega_i,$$

where Ω is the set of all directions $\vec{\omega}=(\theta,\phi)$ of the unit hemisphere aligned normal to the surface in point y. S is the surface of the scene modeled as boundary representation. The function h returns the first point hit when shooting a ray from y into direction $\vec{\omega}_i$. The term $\cos\theta_i$ projects the incoming radiance normal to the surface, where θ_i is the azimuth angle between the surface normal in y and the direction of incidence $\vec{\omega}_i$. The radiance L in a point $y\in S$ into direction $\vec{\omega}_r\in\Omega$ so is the sum of the source radiance L_e and the reflected radiance. Using operator notation we have the shorthand

$$L = L_e + T_{f_r}L.$$

The bidirectional reflectance distribution function f_r accounts for the surface properties like color and gloss. In the general setting this function depends on the incident direction $\vec{\omega}_i$ and reflection direction $\vec{\omega}_r$ of radiance and the location y. In the *radiosity setting* $f_r = f_d(y) := \frac{\rho_d(y)}{\pi}$ is restricted to only diffuse reflection. Then the radiance becomes isotropic, too:

$$L(y) = L_e(y) + \frac{\rho_d(y)}{\pi}\int_\Omega L(h(y,\vec{\omega}_i))\cos\theta_i d\omega_i,$$

*Fachbereich Informatik, AG Numerische Algorithmen, keller@informatik.uni-kl.de, http://www.uni-kl.de/AG-Heinrich/Alex.html

where $\rho_d(y)$ is the reflectivity of the diffuse surface texture.

Given the quadruple (S, f_r, L_e, Ψ), the *global illumination problem* consists in calculating functionals of the form

$$\langle L, \Psi \rangle := \int_S \int_\Omega L(y, \vec{\omega}) \Psi(y, \vec{\omega}) \cos\theta d\omega dy$$

either in the radiosity setting or for the full radiance equation. There are various choices for the detector functional Ψ, e.g. the sum of orthonormal base vectors of a finite vector space, as used in classical or hierarchical radiosity approaches [CW93]. Instead of discretizing the solution of the integral equation and then having to render it in a separate pass, we directly select

$$\Psi_{mn}(y, \vec{\omega}) := \frac{\delta(\vec{\omega} - \vec{\omega}_{y_f})}{\cos\theta} \frac{1}{|P_{nm}|} \chi_{P_{mn}}(h(y, \vec{\omega}))$$

detecting the average radiance passing through the pixel P_{mn} of the image matrix as seen by a pinhole camera[1] from the focal point y_f. $\vec{\omega}_{y_f} := P - y_f$ is the direction of a point P in the support of P_{mn} through y_f. $\chi_{P_{mn}}$ is the characteristic function of the pixel's support and δ the Kronecker delta function.

3 The new Algorithm

Our new algorithm generates a particle approximation of the diffuse radiance in the scene using the quasi-random walk [Kel96b] based on the method of quasi-Monte Carlo integration. Then the graphics hardware renders an image with shadows for each particle used as point light source. Global illumination finally is obtained by summing up the single images in an accumulation buffer [HA90] and displaying the result. The algorithm calculates the average radiance

$$\begin{aligned} \overline{L}_{mn} := \langle L, \Psi_{mn} \rangle &= \langle L_e, \Psi_{mn} \rangle + \langle T_{f_r} L, \Psi_{mn} \rangle \\ &= \langle L_e, \Psi_{mn} \rangle + T_{mn} L \end{aligned} \quad (1)$$

passing through a pixel P_{mn}, where the shorthand $T_{mn}L$ defines the rendering operator, which determines the at least once reflected radiance through P_{mn}. If the radiance L in the radiosity setting can be approximated by a discrete density of M point light sources

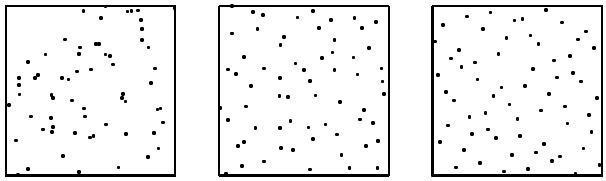
$$L(y) \approx \sum_{i=0}^{M-1} L_i \delta(y - P_i), \quad (2)$$

where L_i is the radiance and P_i is the position of the i-th light source, the application of T_{nm} to the particle approximation yields the very fast rendering algorithm

$$\overline{L}_{mn} \approx \langle L_e, \Psi_{mn} \rangle + \sum_{i=0}^{M-1} T_{mn} L_i \delta(y - P_i).$$

T_{mn} applied to a point light source simultaneously can be evaluated for all pixels of the image matrix by calling a standard graphics hardware illumination routine in the manner of [Hei91, SKvW+92], producing the shaded image of the textured scene including shadows. The directly visible light sources in $\langle L_e, \Psi_{mn} \rangle$ are rendered on the fly by assigning emission to the corresponding surface elements (see the light source in figure 7). The algorithm thus directly operates on the textured scene description in image space and does not apply any kernel or solution discretization to the integral equation. In consequence, no mesh artifacts will occur and also no topological data structure like e.g. a winged-edge representation is required for interpolation or overlapping coefficients evaluation. A small number M (usually $50 \ldots 300$) of point light sources will be sufficient, since from multipass rendering with expensive local pass calculations [CSSD94, Kel95, Kel96a], it is known that a very coarse radiosity solution suffices to produce realistic images. So the speed of the algorithm mainly depends on the frame generation rate of the graphics hardware, promising interactive rates of photorealistic image generation.

[1] For more elaborate camera models we refer to [KMH95].



a b c

Figure 1: Two-dimensional uniform sampling patterns: a) random, b) jittered, and c) Halton for $N = 64$ samples.

3.1 Quasi-Monte Carlo Integration

Similar to [Kel96b], we use the method of quasi-Monte Carlo integration for the evaluation of the integrals. For the approximation

$$\int_{[0,1)^s} f(x) dx \approx \frac{1}{N} \sum_{i=0}^{N-1} f(x_i)$$

we generate the N sample points $x_0, \ldots, x_{N-1} \in [0,1)^s$ using the Halton sequence. Based on the radical inverse function

$$\Phi_b(i) := \sum_{j=0}^{\infty} a_j(i)\, b^{-j-1} \in [0,1) \Leftrightarrow i = \sum_{j=0}^{\infty} a_j(i)\, b^j,$$

the s-dimensional Halton low discrepancy sequence (see comparison in figure 1) is

$$x_i = (\Phi_{b_1}(i), \ldots, \Phi_{b_s}(i)), i \in \mathbb{N}_0,$$

where b_j is the j-th prime number. Note that each segment $P_{N'}$ of a larger segment P_N, $N' < N$, of successive points of the Halton sequence is a quadrature rule, too, which is not the case for variance reduced sampling methods like jittered or N-rooks sampling. In addition, the Halton points are available for any choice of N independent of dimension s. Compared to N-rooks sampling, where for each dimension a random permutation of size $\mathcal{O}(N)$ and unknown quality has to be stored, the Halton points can be generated for arbitrary i (see figure 2) or successively (see figure 3) by the algorithm of [HW64] at a speed comparable to usual pseudo-random generators without additional storage.

The integrands in computer graphics are discontinuous, allowing only very pessimistic upper error bounds (for details see [Nie92]) for the integral approximation. Nevertheless, the numerical evidence in [Kel95, Kel96a, Kel96b] shows that the calculation of functionals of the solution of the radiance equation by means of low discrepancy sequences results in a much smoother convergence at a slightly superior rate as compared to random sampling. In [PTVF92] a plausibility argument gives the rate of $\mathcal{O}(N^{-\frac{s+1}{2s}})$ as upper bound for the quasi-Monte Carlo method applied to discontinuous functions. For high dimension, this rate converges to the random rate of $\mathcal{O}(N^{-\frac{1}{2}})$, but since $s \ll \infty$, the rate of sampling with Halton points is superior to random sampling. The above arguments also apply to jittered sampling [Mit96], but the low discrepancy pattern is deterministic and therefore works without variance!

```
void Φ(int b, int i)
{
    double x = 0.0, f = 1/b;

    while(i)
    {
        x += f * (double) (i % b);
        i /= b;
        f *= 1/b;
    }

    return x;
}
```

Figure 2: Direct calculation of the radical inverse function.

```
void Φ(int b, double x)
{
    double h, hh, r = 1.0 - x - 1e-10;

    if(1/b < r)
        x += 1/b;
    else
    {
        h = 1/b;

        do
        {
            hh = h;
            h *= 1/b;
        }
        while(h >= r);

        x += hh + h - 1.0;
    }

    return x;
}
```

Figure 3: Incremental calculation of the radical inverse function.

3.2 The Quasi-Random Walk

In realistic applications the transport operator norm $||T_{f_r}|| < 1$, meaning that less than 100% of the incident radiance is reflected. So the Neumann series

$$L = (I - T_{f_r})^{-1} L_e = \sum_{j=0}^{\infty} T_{f_r}^j L_e$$

converges and can be used to solve the integral equation. Inserted in (1), after some transformations, for the radiosity setting we get

$$T_{mn}L = \frac{1}{|P_{mn}|} \sum_{j=0}^{\infty} \int_{P_{mn}} \int_{\Omega^j} \int_{S_e} p_j(y_0, \vec{\omega}_0, \ldots, \vec{\omega}_j) V(y_j, y') f_d(y') \frac{\cos\theta_j \cos\theta'}{|y_j - y'|^2} dy_0 d\omega_0 \cdots d\omega_j dP. \quad (3)$$

Here $S_e := \text{ supp } L_e \subseteq S$ is the support of the light sources, and $y' = h(y_f, P - y_f) \in S$ is the first point hit when shooting a ray from the eye at y_f through the point $P \in P_{mn}$ into the scene. $V(y_j, y')$ checks the mutual visibility of the points y_j and y', yielding 1 in case of visibility and 0 else. The radiance density

$$p_j(y_0, \vec{\omega}_0, \ldots, \vec{\omega}_j) := L_e(y_0) \prod_{l=1}^{j} (\cos\theta_{l-1} f_d(y_l))$$

represents the source radiance after j reflections. Here $y_0 \in S_e$ is a point on a light source, and the subsequent points $y_{l+1} := h(y_l, \vec{\omega}_l) \in S, 0 \le l < j$, of a path are determined by ray shooting. Taking the diffuse part of the scene, the operator norm can be estimated by the mean reflectivity

$$\overline{\rho} := \frac{\sum_{k=1}^{K} \rho_{d,k} |A_k|}{\sum_{k=1}^{K} |A_k|} \approx ||T_{f_d}||,$$

where the scene $S := \cup_{k=1}^{K} A_k$ is composed of K surface elements A_k with average diffuse reflectivity of $\rho_{d,k}$.

As in [Kel96b], the radiance density is simulated by its particle nature using the technique of the quasi-random walk. Since in realistic scene models the actual diffuse reflectivity has only small deviation from $\overline{\rho}$, we can use fractional absorption and avoid Russian Roulette absorption [AK90]. So from N particles started at the light sources, $\overline{\rho}N$ particles are supposed not to be absorbed by the first reflection, $\overline{\rho}^2 N$ survive the second reflection and so on. Then the number M of radiance points generated is bounded by

$$M < \sum_{j=0}^{\infty} \overline{\rho}^j N = \frac{1}{1 - \overline{\rho}} N =: \overline{l} N$$

and thus is linear in N depending on the average scene reflectivity $\overline{\rho}$, where $\overline{l}$ is the mean path length. The quasi-random walk scheme now evaluates $T_{mn}L_e$ using N point lights, $T_{mn}T_{f_d}L_e$ by using $\lfloor \overline{\rho}N \rfloor$ point lights, and so on, where the particles are generated using the Halton sequence. As a consequence, the particles are concentrated in the lower powers of the reflection operator, which due to the operator norm contribute the most important parts of the image, thus fully exploiting the advantages of low discrepancy sampling.

3.3 Implementation

The pseudocode of the instant radiosity algorithm is given in figure 4. To generate the discrete density approximation of p_j by low discrepancy points, we first fix the number N of particles to start off the light source. By an isometry y_0 (e.g. see the collection in [Shi92]) the first two components of the Halton sequence are mapped from the unitsquare onto the surface of the light source, yielding the starting point $y = y_0(\Phi_2, \Phi_3)$ with power $L = L_e(y)$ supp L_e. In the case of multiple light sources, first a light source is selected by the composition method identical to [Kel96b], then the isometry is applied. Exploiting the property of the Halton sequence that segments of the sequence have small discrepancy, too, the $\lfloor \overline{\rho}N \rfloor$ first points are used to shoot a ray into direction $\vec{\omega}$ using

$$\vec{\omega} = \vec{\omega}_d(\Phi_{b_{2j+2}}, \Phi_{b_{2j+3}}) = (\arcsin\sqrt{\Phi_{b_{2j+2}}}, 2\pi\Phi_{b_{2j+3}}),$$

where the direction already is distributed with respect to the cosine-term in the density p_j. In the next hitpoint $y = h(y, \vec{\omega})$ the particle's radiance is attenuated by $f_d(y)$. From these particles the first $\lfloor \overline{\rho}^2 N \rfloor$ continue their paths, repeating the diffuse scattering procedure until no particles remain. The starting points and the subsequent hitpoints of the above quasi-random walk then form the discrete density approximation (2) which is used for the hardware lighting call `glRenderShadowedScene`, i.e. the scene is rendered with shadows and the point light source located in y with the power $\frac{N}{\lfloor w \rfloor} L$. The term $w = \overline{\rho}^j N$ is used to compensate the attenuation by $\prod_{l=1}^{j} f_d(y_l)$, making the contribution of each image equally important. Finally the quasi-Monte Carlo integration is performed by accumulating all images with the weight $\frac{1}{N}$.

```
void InstantRadiosity(int N, double ρ̄)
{
    double w, Start; int End, Reflections = 0;
    Color L; Point y; Vector ω⃗;

    Start = End = N;

    while(End > 0)
    {
        Start *= ρ̄;

        for(int i = (int) Start; i < End; i++)
        {
            // Select starting point on light source
            y = y0(Φ2(i), Φ3(i));
            L = Le(y) * supp Le;
            w = N;

            // trace reflections
            for(int j = 0; j <= Reflections; j++)
            {
                glRenderShadowedScene(N/⌊w⌋ L, y);
                glAccum(GL_ACCUM, 1/N);
                // diffuse scattering
                ω⃗ = ω⃗d(Φb2j+2(i), Φb2j+3(i));
                //trace ray from y into direction ω⃗
                y = h(y, ω⃗);
                // Attenuate and compensate
                L *= fd(y);
                w *= ρ̄;
            }
        }

        Reflections++;
        End = (int) Start;
    }

    glAccum(GL_RETURN, 1.0);
}
```

Figure 4: Instant radiosity pseudocode (see section 3.3).

4 Extensions

The fast, deterministic radiosity algorithm introduced in the previous section consistently renders diffuse global illumination for still images. By the new concept of jittered low discrepancy sampling, we treat issues of antialiasing in order to improve image quality at low sampling rates by random elements. Then specular effects are added to the algorithm. Finally modifications for realtime walkthroughs are indicated.

4.1 Jittered Low Discrepancy Sampling

Taking a look at the two-dimensional Hammersley sequence $(\frac{i}{N}, \Phi_2(i))_{i=0}^{N-1}$ in figure 5, it becomes obvious that the low discrepancy points based on radical inversion are aligned to a grid. This grid structure guarantees a minimum distance property, and thus an implicit stratification, but is prone to aliasing. The new concept of jittered low discrepancy sampling joins the two worlds of Monte Carlo and quasi-Monte Carlo integration by using low discrepancy point sets as stratification. Approximating the grid resolution by $\frac{1}{N}$, this is done by randomizing each low discrepancy point in its raster cell, i.e. replacing the radical inverse Φ_b by $\Phi_b + \frac{\xi}{N}$, where ξ is a random variable, assuring that the sample remains in the unit interval. From the images in figure 5 it also becomes obvious that the jittered Hammersley sequence in two dimensions is a special case of N-rooks sampling. The Hammersley points, however, can

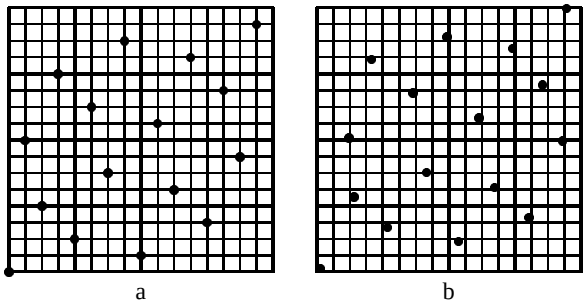


Figure 5: Grid structure of the a) Hammersley and b) jittered Hammersley sampling patterns for $N = 16$.

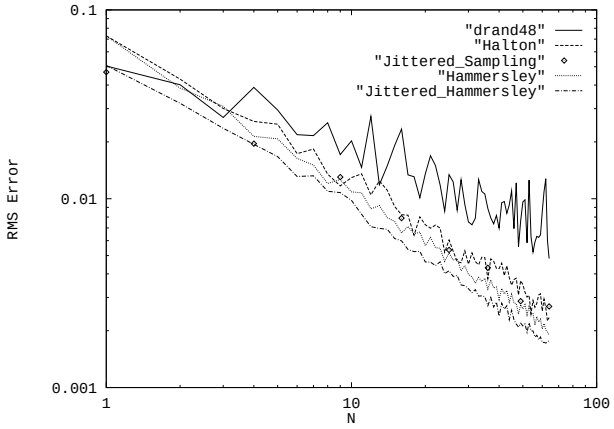


Figure 6: Convergence of different sampling patterns.

be generated without storing a random permutation; they implicitly are a permutation with low discrepancy.

Applied to pixel supersampling for antialiasing as in [HA90], the two dimensional jittered Hammersley sequence exposes an even faster convergence than standard variance reduced sampling. This can be seen in figure 6, where the RMS-error is plotted versus the sampling rate N for an experiment where images of the textured scene in figure 8 at a sampling rate of $N = 1 \ldots 64$ were compared to a master calculation at 640 samples per pixel.

In our new algorithm then, according to the path number i, the corresponding x_i of the Hammersley sequence is jittered for each image produced by the particles of the path. Using this sampling pattern in the manner of [HA90], hardware antialiasing, as available on some graphics accelerators, becomes redundant. Concerning the quasi-random walk, the components of the Halton vector have to be jittered by different ranges. Starting on the light sources we have N particles, where the coordinates (Φ_2, Φ_3) will be jittered by $\frac{\xi}{N}$, in the next step only $\lfloor \overline{\rho} N \rfloor$ particles are traced, so the jitter range is $\frac{1}{\lfloor \overline{\rho} N \rfloor}$, and so on. By this procedure aliasing is reduced, even improving the convergence rate.

4.2 Specular Effects

To add specular effects, we first let T_{mn} use the full BRDF f_r in the hardware lighting pass, enabling specular highlights as can be seen in figure 8. In the particle generation phase, by a random decision each surface is tested to be specular or diffuse according to its BRDF [CRMT91, War92]. In case of specular reflection a virtual light source is generated, i.e. the origin of the ray is mirrored by the specular surface under consideration. The virtual light source now illuminates the part of the scene inside the pyramid spanned

by itself and the contour of the reflecting element using techniques of [DB94, Die96]. Note that virtual light sources can only be applied at planar polygon level. Afterwards the incoming particle is randomly scattered according to the specular part of the BRDF. Particles hitting specular surfaces so produce a virtual light source raising M and cause a lengthening of the low discrepancy path by a random piece. If the graphics hardware supports spot lights and the particles $(L_i, P_i, \vec{\omega}_i)$ are equipped with their direction of incidence, even more general light source emission, i.e. with a $\cos^d$-distribution, and caustics can be simulated. Finally the visible specular objects have to be treated separately by ray tracing or advanced hardware techniques as illustrated in [DB94, Die96].

4.3 Realtime Walkthroughs

The algorithm designed so far produces still images. In an animated environment, the quasi-random walk is substituted by tracing fixed length paths generated by the Halton sequence as introduced in [Kel95, Kel96a]. All images produced by one path are accumulated and the resulting image is stored with its time of generation. Keeping the last N images of the last N paths, each time a new path is completed, the oldest image is replaced by the new one. The current N images then are accumulated and displayed, thus implicitly performing temporal antialiasing. As path length we choose the maximal length of the paths obtained in the quasi-random walk procedure

$$l_{max} := \left\lfloor -\frac{\log N}{\log \overline{\rho}} \right\rfloor .$$

Then the maximal frame rate of the graphics accelerator is reduced by the factor $\frac{1}{l_{max}}$ for one time step, allowing for realtime rendering rates (e.g. $\overline{\rho} = 0.5774$ of the scene in figure 7 with $N = 64$ yields $l_{max} = 7$), and since the algorithm directly operates on the scene graph, dynamic environments can be treated without further effort!

Using texture mapping hardware for displaying illumination maps, generalizing the method of [HH96] yields another approach to realtime walkthroughs for static environments. Instead of using jittered sampling and $\overline{\rho} = 0$, i.e. only direct illumination, we replace the light samples of [HH96] by our discrete density approximation of radiance (see section 3.3). By this simple enhancement the algorithm of [HH96] renders the global diffuse illumination into textures, which then interactively can be displayed. Besides considerable memory consumption, the solution of the radiance equation now is discretized in textures, which may result in visible artifacts if the texture resolution has been chosen too small.

5 Discussion of the Algorithm

The algorithm displayed in figure 4 is illustrated by figure 7. For a path number of $N = 10$ the single images created by the point light sources (red balls) are shown. In addition, the results of accumulating $N \in \{10, 32, 64\}$ paths (i.e. $M = 20$, 42, or 147 images in PAL-resolution 720x576 pixels) are displayed. The images of the scene of 402 quadrangles have been produced on a Silicon Graphics Onyx with Reality Engine^2 graphics and a 75MHz R8000 processor in 24 seconds by the shadow algorithm of [Hei91], i.e. by evaluating the shadow volume of each primitive drawn over the depth buffer image of the scene for stenciling out the shadows. Using the shadow techniques of [SKvW$^+$92] would result in an at least twice as fast algorithm. This emphasizes the fact that the performance of our approach mainly depends on the hardware rendering speed, since the particle approximation can be generated instantly. Note that the smooth shadows and the indirect illumination are obtained without any meshing which would have raised the number of polygons to a multiple.

Two problems of the algorithm become apparent at very low sampling rates N. The first problem is the weak singularity of the operator T_{mn}, when the distance $|y_j - y'|^2$ of point light source y_j and point y' to be lit comes close to zero (see figure 7). Then the value to be entered into the frame buffer is overmodulated and will be clipped to the maximal representable value. The second problem is that each light point colored by a texture has a large influence on the overall color of the scene. Since all images are weighted by $\frac{1}{N}$ in the accumulated image, however, the impact of one of the above cases is at most of order $\frac{1}{N}$, which in most cases is hardly perceivable.

Our access to graphics hardware was restricted to the above example, so we exploited shadow caching and eye ray coherence, to simulate the hardware evaluation of T_{mn} for the point lights by ray tracing. For the conference room consisting of 39584 scene primitives, N was chosen independent of the number of extended light sources, as demonstrated in figure 9, where only $N = 128$ paths are used for a scene with 248 light sources.

Since the algorithm directly operates on the scene graph without additional storage for discretizations animated environments, cyclic graphs as used for plant modeling, or level-of-detail modeling easily can be rendered in a photorealistic way. The only additional data structure required is a space order like e.g. a BSP-tree for accelerating the ray shooting $h(y, \vec{\omega})$. On the one hand it is possible to generate the discrete density approximation $(L_i, P_i)_{i=0}^{M-1}$ of the radiance and to affix these point light sources to the scene description, e.g. MGF or VRML. Then the final rendering process, i.e. loading the scene graph and illuminating it by the point light sources, does not need the space order for ray shooting. On the other hand the BSP can be used for rendering with impostors similar to [SLS$^+$96] and hierarchical clipping, speeding up the frames-per-second rate. These techniques have not yet been included in the implementation used for the time measurements, but reduce the constant of the time complexity $\mathcal{O}(NK)$ of our algorithm, where the number N of paths can be freely chosen with respect to the frame and accumulation buffer accuracy, and K is the number of elements in the scene.

6 Conclusion and Future Work

A new method for rendering from the radiance equation has been introduced. Based on the quasi-random walk, point light sources are generated for fast hardware illumination. The single images are superimposed, yielding one of the fastest physically correct rendering procedures. Working in image space, the algorithm does not need any storage for kernel or solution discretization or topological information. The efficient algorithm itself is very compact and can be easily implemented using a standard graphics API, requiring only a ray intersection routine and an isometry from the unit square onto the surface of each light source. The deterministic algorithm has been extended by the new concept of jittered low discrepancy sampling.

Since our method already includes importance sampling, stratification of low discrepancy, and jittering for antialiasing, future work will be spent on the investigation of the adjoint transport operator for efficiently sampling the light sources with high impact on the final image (especially in large and/or indirectly illuminated environments), reducing the number M of point lights, and decreasing the influence of the weak singularity of the transport operator. Provided that the rendering hardware supplies fog attenuation, even an extension to participating media is possible, since the direct simulation is easily extended for volume scattering.

7 Acknowledgements

The author would like to thank Stefan Heinrich and Hans-Christian Rodrian for the discussions. Special thanks go to Marc Stamminger for providing access to the Reality Engine[2].

The models of this paper (except for figure 8, which has been modeled by Christian Keller) are taken from the *Material and Geometry Format* (MGF)-package of Greg Ward (available via `http://radsite.lbl.gov/mgf/HOME.html`). The greyscale scene has been provided by Peter Shirley. The conference room has been modeled by Anat Grynberg and Greg Ward. A photograph of the original room can be found in [FvDFH96].

References

[AK90] J. Arvo and D. Kirk. Particle Transport and Image Synthesis. In *Computer Graphics (SIGGRAPH 90 Conference Proceedings)*, pages 63 – 66, 1990.

[CRMT91] S. Chen, H. Rushmeier, G. Miller, and D. Turner. A progressive Multi-Pass Method for Global Illumination. In *Computer Graphics (SIGGRAPH 91 Conference Proceedings)*, pages 165 – 174, 1991.

[CSSD94] P. Christensen, E. Stollnitz, D. Salesin, and T. DeRose. Wavelet Radiance. In *Proc. 5th Eurographics Workshop on Rendering*, pages 287–302, 1994.

[CW93] M. Cohen and J. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press Professional, Cambridge, 1993.

[DB94] P. Diefenbach and N. Badler. Pipeline Rendering: Interactive Refractions, Reflections, and Shadows. *Displays: Special Issue on Interactive Computer Graphics*, 15(3):173–180, 1994.

[Die96] Paul J. Diefenbach. *Pipeline Rendering: Interaction and Realism through Hardware-Based Multi-Pass Rendering.* Ph.D. thesis, University of Pennsylvania, 1996.

[FvDFH96] J. Foley, A. van Dam, S. Feiner, and J. Hughes. *Computer Graphics, Principles and Practice, 2nd Edition in C.* Addison-Wesley, 1996.

[HA90] P. Haeberli and K. Akeley. The Accumulation Buffer: Hardware Support for High-Quality Rendering. In *Computer Graphics (SIGGRAPH 90 Conference Proceedings)*, pages 309–318, 1990.

[Hei91] T. Heidmann. Real Shadows - Real Time. *Iris Universe*, (18):28–31, 1991.

[HH96] M. Herf and P. Heckbert. Fast Soft Shadows. In *Technical Sketches (SIGGRAPH 96 Visual Proceedings)*, page 145, 1996.

[HW64] J. Halton and G. Weller. Algorithm 247: Radical-inverse quasi-random point sequence. *Comm. ACM*, 7(12):701–702, 1964.

[Kaj86] J. Kajiya. The Rendering Equation. In *Computer Graphics (SIGGRAPH 86 Conference Proceedings)*, pages 143–150, 1986.

[Kel95] A. Keller. A Quasi-Monte Carlo Algorithm for the Global Illumination Problem in the Radiosity Setting. In H. Niederreiter and P. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 106, pages 239–251. Springer, 1995.

[Kel96a] A. Keller. Quasi-Monte Carlo Methods in Computer Graphics: The Global Illumination Problem. *Lectures in App. Math.*, 32:455–469, 1996.

[Kel96b] A. Keller. Quasi-Monte Carlo Radiosity. In X. Pueyo and P. Schröder, editors, *Rendering Techniques '96 (Proc. 7th Eurographics Workshop on Rendering)*, pages 101–110. Springer, 1996.

[KMH95] C. Kolb, D. Mitchell, and P. Hanrahan. A Realistic Camera Model for Computer Graphics. In *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 317–324, 1995.

[LTG92] D. Lischinski, F. Tampieri, and D. Greenberg. Discontinuity Meshing for Accurate Radiosity. *IEEE Computer Graphics & Applications*, 12(6):25–39, 1992.

[LW93] E. Lafortune and Y. Willems. Bidirectional Path Tracing. In *Proc. 3rd International Conference on Computational Graphics and Visualization Techniques (Compugraphics)*, pages 145–153, 1993.

[Mit96] D. Mitchell. Consequences of Stratified Sampling in Graphics. In *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 277–280, 1996.

[Nie92] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods.* SIAM, Pennsylvania, 1992.

[PTVF92] H. Press, S. Teukolsky, T. Vetterling, and B. Flannery. *Numerical Recipes in C.* Cambridge University Press, 1992.

[Shi92] P. Shirley. Nonuniform Random Point Sets via Warping. In D. Kirk, editor, *Graphics Gems III*, pages 80–83. Academic Press Professional, 1992.

[SKvW+92] M. Segal, C. Korobkin, R. van Widenfelt, J. Foran, and P. Haeberli. Fast Shadows and Lighting Effects using Texture Mapping. In *Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, pages 249–252, 1992.

[SLS+96] J. Shade, D. Lischinski, D. Salesin, T. DeRose, and J. Snyder. Hierarchical Image Caching for Accelerated Walkthroughs of Complex Environments. In *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 75–82, 1996.

[VG94] E. Veach and L. Guibas. Bidirectional Estimators for Light Transport. In *Proc. 5th Eurographics Worshop on Rendering*, pages 147 – 161, Darmstadt, Germany, June 1994.

[War92] G. Ward. Measuring and Modeling Anisotropic Reflection. In *Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, pages 265 – 272, 1992.

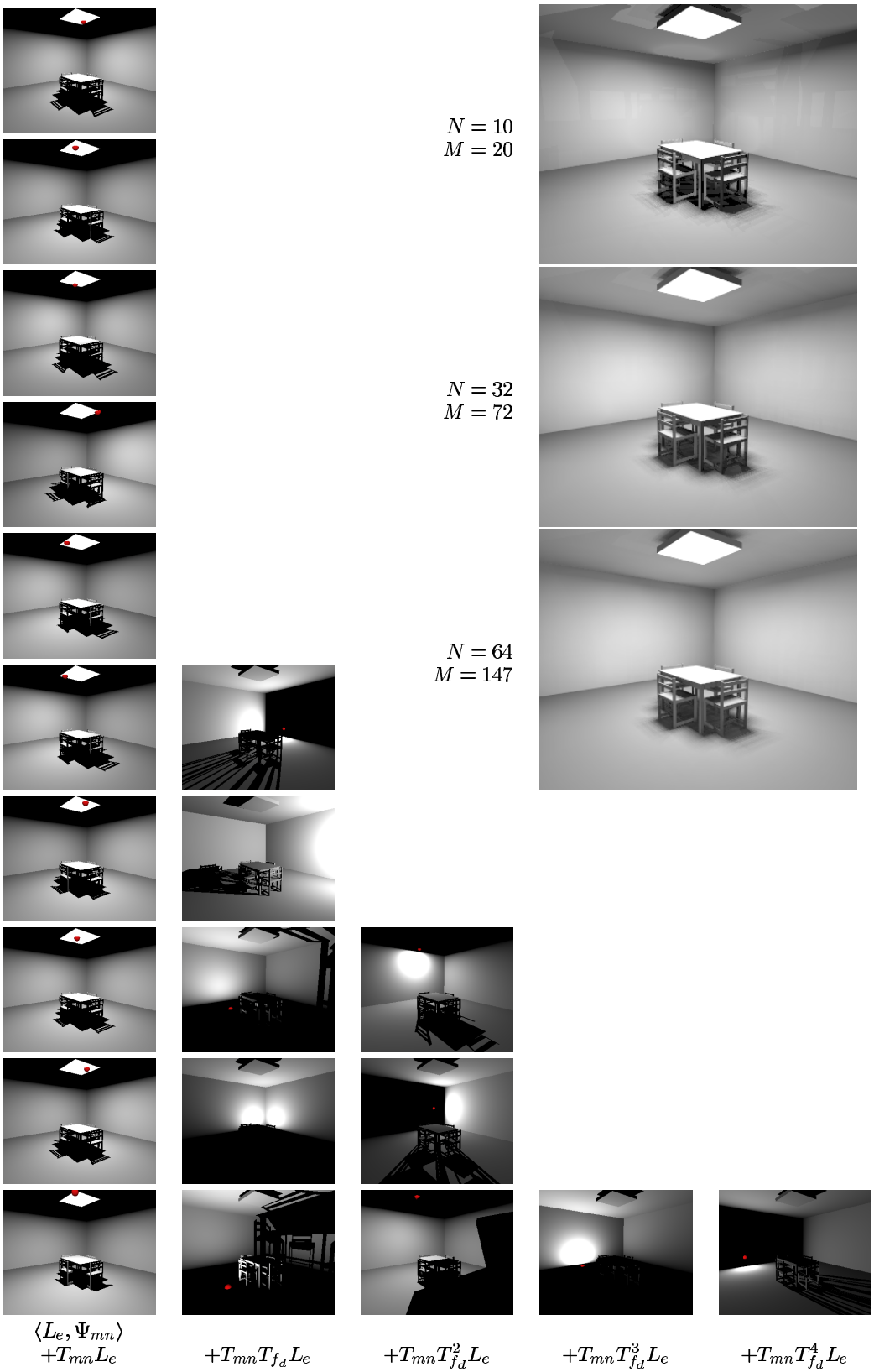


Figure 7: Illustration of the quasi-random walk integration scheme showing the intermediate images accumulated for $N = 10$ paths with $\overline{p} = 0.5774$ and the resulting images for $N \in \{10, 32, 64\}$.

Figure 8: Specular effects of the standlight on the floor by using the full BRDF f_r in T_{mn} for $N = 128$.

Figure 9: Conference room image for $N = 128$.