A study of incremental update of global illumination algorithms

R. Garcia, C. Ureña, M. Lastra, R. Montes, J. Revelles
Dpto Lenguajes y Sistemas Informáticos
University of Granada
{ruben,curena,mlastral,rosana,jrevelle}@ugr.es

ABSTRACT

Global illumination solutions provide a very accurate representation of illumination. However, they are usually costly to calculate. In the common case of quasi-static scenario, in which most of the scene is static and only a few objects move, most of the illumination can be reused from previous frames, yielding increased performance. This article studies theoretically the performance of global illumination algorithms for the case of interactive recalculation of quasi-static scenes, concentrating in the Density Estimation on the Tangent Plane algorithm, although the study is applicable to other techniques. The results are validated empirically with a test scene. Guidelines are given to choose the best algorithm for each case.

Keywords: Global Illumination, Density Estimation, Range Searching, Interactivity, Cost analysis.

1 INTRODUCTION

Global Illumination algorithms calculate a solution to the rendering equation proposed by [Kaj86]. Montecarlo methods are very often used currently. However comparing the performance of the different methods taking into account the different variance and bias is difficult. BART [LAM00] provides a compendium of scenes with which the algorithms can be compared empirically. The aim of this article is to provide a basis for a theoretical comparison of these algorithms, taking into account both variance and computation time.

In order to do this, the number of rays shoot is used to calculate theoretically the variance of the method and the computation time. Then the computation time versus variance link can be used to compare different algorithms theoretically.

Sometimes different algorithms have the same underlying kernel for the Montecarlo integration, and therefore the same variance. In these cases, a study of computation time suffices.

A theoretical study of incremental recalculation of global illumination is presented in this paper, which indicates the suitability of the different methods for the different parts of the scene.

The following section has a review of Global Illumination algorithms. Section 2 contains an in depth analysis of global illumination using montecarlo methods. Section 3.1 has a review and simplification of formulas from integral geometry which are used to estimate the cost of different algorithms. In section 3.2 a study of incremental recalculation of illumination for quasi-static scenes is presented. Section 4 contains a practical study which validates the theoretical results. Finally, conclusions and some possible future work is presented.

1.1 Global Illumination

Interactive and Real-Time global illumination is becoming increasingly important for many computer graphics applications, such as building design and games.

There have been many important algorithms which attain high frame rates for medium-scale scenes. They are based on highly parallel pathtracers and Photon Maps derivatives. [BWS03], [WGS04] and [GWS04] are examples of parallelism, whereas Density Estimation on the Tangent Plane [LURM02] and Ray Maps [HBHS05] are examples of extensions of Photon Maps to increase performance.

These algorithms recalculate the illumination for every frame, which is the best option for highly dynamic scenes, since most of the illumination changes from frame to frame.

However, in building design and games, the most common scenes are a complex static scene and relatively simple dynamic objects. In building design, the designer will modify one object at a time to choose the best configuration for a given, fixed, lighting setup. In games, the user will control one small character immersed in a large, mostly static, virtual world, while the game controls other small characters.

Under these conditions, most of the illumination can be reused from previous frames, since the small moving objects change the illumination only slightly.

Selective Photon Tracing [DBMS02] recalculates incrementally the illumination by keeping the age of the photons and recalculating as many of the old and no longer valid photons as time allows, every frame.

[GUL+04] provides an algorithm which tests the photons which are not valid and recalculates them. Old valid photons are never discarded. The results are the same as full recalculation of the frame for the Density Estimation on the Tangent Plane algorithm. However, for Photon Maps, while the result has no intrinsic bias, it is different of the full recalculation, since obtaining the same result means that adding the contribution of an impact would require the removal of the contribution of the photon further away from the point, and that requires a photon map query. Recalculation in Ray Maps remains an open topic [HBHS05].

2 DENSITY ESTIMATION METHODS

In order to obtain a radiosity value at a point, an approximation of the integral of the incident radiance for all the directions towards that point must be calculated.

The most basic method is described by Arvo in [Arv86] and Patanaik in [PM92]. It computes the impacts of the photons in the patches and calculates the energy density in those patches. Then vertexes are assigned a radiance which is the average of the patches to which they belong.

One method which has proved useful to obtain an estimate of the integral is the Density Estimation Method, popularized by [WHSG97] and especially [Jen96]. The original method by Shirley consists of three phases. The first phase is based on the particle model of light, and traces a number of photons from the light sources. The second phase (Density Estimation proper) estimates the radiance. The third phase, decimation, simplifies the geometry after illumination has been calculated. This last phase is dropped often because it is considered to be outside the scope of density estimation.

Jensen [Jen96], devised the well known Photon Maps method. It consists in finding the nearest n ray impacts (n is predefined) to the point where radiance is being estimated, adding their energy, and dividing it by the area of the greatest circle of the sphere which contains the n impacts.

The most known limitation of Photon Maps, is that when radiance on a point is calculated, the surface in the neighborhood should be relatively planar and large. [HP02] presents an algorithm which solves this limitation by using geometry information near the point.

Another less known limitation of Photon Maps and [HP02], mentioned in [LURM02] and [HBHS05], is that if relatively very small surfaces exist in the scene, these zones have a comparatively high variance, and they tend to appear either too bright (in a few cases) or too dark (which is more frequent) if the number of photons is not large enough.

Ray Maps [HBHS05] is a new method which uses a lazily constructed kd-tree to store the ray trajectories. An efficient method to calculate rays which intersect a given subset of the space, or nearest rays to a point according to different metrics is given.

2.1 Density Estimation on the Tangent Plane

A method to avoid the high variance of Photon Maps mentioned in the previous section consists in storing the rays in the scene and using a fixed size disc centered in the point where radiance is being calculated, and contained in the plane tangent to the surface. Rays intersecting a given disc are used to calculate the radiance at the point on which the disc is centered [LURM02]. The algorithm is called Density Estimation on the Tangent Plane (DETP). Note that this algorithm keeps track of the trajectory of the photons (origin, direction and impact point) unlike the original Photon Maps. See Figure 1. To avoid self-shadowing in concave surfaces, the second intersection of the ray and the scene is used instead of the first. This method uses discs of fixed radius [LURM02].

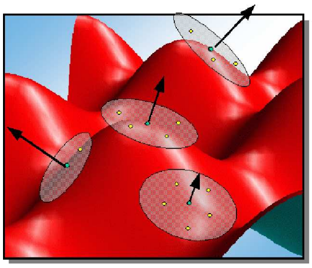

Figure 1: Density Estimation on the Tangent Plane. The yellow dots on the disc represent the intersection of the rays

The algorithm has optimum trade-off between accuracy and variance when the disc radius (which is a user defined constant) is in the order of magnitude of half the distance between irradiance calculations. If disc radius were smaller, rays intersecting the tangent plane near the middle point of two irradiance calculations would be ignored. If it were larger, intersections would be used for various calculations, hiding small illumination features which otherwise could be reconstructed.

2.2 Averaging over photons versus averaging over surface

Most of Density Estimation algorithms for global illumination consider a fixed number of samples near the point where radiance is calculated, and project the energy into a disc of minimum radius containing all the samples. This approach averages the samples over constant energy.

Density Estimation on the Tangent Plane, however, takes a disc of fixed radius and uses all the samples contained there. This approach averages the samples over constant surface area.

Both approaches are statistically equivalent with respect to the bias and noise involved, trading spatial accuracy for energy accuracy.

 ISBN 978-80-86943-02-2

For fast incremental recalculation of illumination, however, averaging over surface means photons can be added and removed very efficiently, since the distance between a photon and a point is the only data needed to know whether the energy of the photon should be added to the irradiance of the point. On the contrary, averaging over energy means that adding the energy of one photon requires finding the photon further away and removing its energy.

This article concentrates therefore on DETP and provides a theoretical study of complexity which complements [GUR$^+$06] when these algorithms are used for recalculation.

2.3 Sphere cache

The limitation of DETP is that the number of disc-ray intersections is high, therefore increasing the computation time. To address this, the sphere cache [LURM02] was developed.

The sphere cache consists in creating a hierarchy of spheres of decreasing radius and storing the rays which intersect each sphere in order to decrease the number of ray-disc intersection tests.

Firstly, a sphere tangent (i.e. circumscribed) to the bounding box of the scene is built. This sphere intersects all the rays.

Then, as Figure 2 shows, spheres of decreasing radius are built one inside the other (the ratio between two consecutive spheres is a parameter called Q), until the radius is just above the disc radius mentioned in the previous section. However, spheres with less than a given number of rays (usually 100) are not subdivided, for efficiency reasons.

Each sphere has an associated data structure which contains the rays which the aforementioned sphere intersects. These rays are calculated by the intersection of the sphere with the rays in the immediately enclosing sphere.

The first point at which radiance is to be calculated is the center of the spheres of decreasing radius. Therefore, the first disc is contained in the inner sphere. Irradiance can be calculated by checking which rays in the inner sphere intersect the disc as well, and adding their energy. The number of ray-disc intersection tests is clearly reduced.

For the rest of the points, if the disc centered in the point is contained in the inner sphere, the disc is intersected against the rays in this sphere. Otherwise, the sphere is discarded, and the rest of the spheres are tested in order, until one is found to enclose the disc. Then the hierarchy of spheres is recalculated, using this point as center. See Figure 2 right.

Finally, the disc is intersected against the rays in the innermost sphere, in the same way as when no recalculation of spheres is needed.

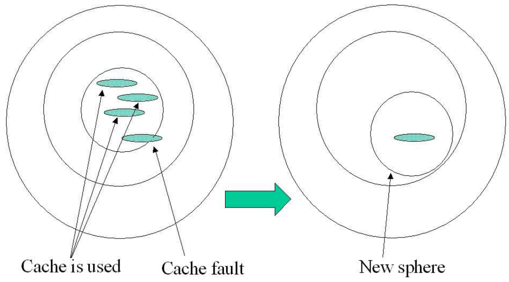

Figure 2: Sphere cache. If a disc lays partially out of a sphere, the sphere is discarded and a new one is created.

Lastra et al. [LURM02] demonstrated that the use of space filling curves to reorder the points increments spatial coherence, and therefore reduces computation time. This approach is called point sorting.

2.4 Disc indexing

The disc indexing technique creates a spatial indexing of the discs in the scene. This is accomplished by considering the discs as real geometry, and applying a space partitioning method to them. The discs are initialized with a radiance value of zero. Then the rays traverse the spatial index adding their contribution to the discs they intersect. See Figure 3. The ray need

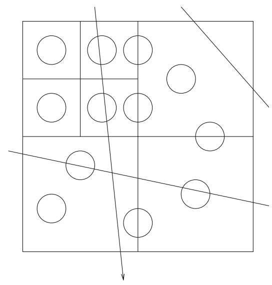

Figure 3: Disc indexing. The rays traverse the indexing, adding their contribution to the discs they intersect

only be followed until the first intersection with the real scene (or the second if concave surfaces exist). The spatial indexing should be able to store discs and to calculate efficiently all the intersections with a segment (the endpoints of this segment are the origin of the ray and the intersection with the real scene). All the published algorithms meet this criterion.

After radiance has been calculated, each disc contains an estimate of the radiance according to the DETP scheme. The data structure can be considered a sort of irradiance cache [WRC88]; therefore new irradiance values can be estimated using the same interpolation which that paper proposes, or by using Irradiance Gradients from [WH92].

Some work [HP03] has been done on studying characteristics of the scene which make some indexing techniques more efficient than others. Other studies [HPP00][RLMC03] use a fast simulation with few rays

 ISBN 978-80-86943-02-2

to choose the most appropriate indexing method. Since the disc position follows the surface of the objects, this research is applicable for this technique as well.

This method has higher performance than the original sphere cache intersection method when the discs have a radius which is in the order of magnitude of the mean distance among the points in which radiance is being calculated. In other situations, the performance of the sphere cache is higher. Details are provided in [GUR+06].

3 TIME COMPLEXITY OF INCREMENTAL CALCULATION OF ILLUMINATION

[GUL+04] described an incremental method for the recalculation of global illumination in a quasi-static scene. In this type of scene, there is a static scenario, which contains most of the geometry complexity of the scene, and a dynamic object or objects, which are relatively simple. When the objects move, most of the radiance information can be reused from previous frames.

In the static scene, only rays which intersect the dynamic object in the previous frame or in this frame can change their contribution to the static scene. Dynamic objects, on the other hand, must have their whole radiance estimate recalculated taking into account all the rays in the scene.

The static scene can have the spatial indexing of disc indexing reused for the whole simulation. The dynamic scene, on the contrary, needs to have it rebuilt on each frame, since it moves and rotates.

With respect to sphere cache, the dynamic scene is very localized in space; therefore the sphere around the bounding box of the moving objects will reject most of the rays. Since objects move smoothly, sphere cache can be very efficient for these objects.

These reasons suggest using sphere cache for the dynamic scene, and choosing the most appropriate method, according to the characteristics of the scene, to the static scene.

The results presented in [GUR+06] were used to study the performance of such a system. The time needed to construct the spatial index produces higher time in the first recalculations, but after 180 frames (the first seven seconds of the animation), the cost has been amortized for a small radius and number of photons. This time raises to 7000 frames when the radius grows and the number of photons increase. Then Sphere cache becomes more efficient and the indexing cannot be amortized.

3.1 Theoretical study of computing time

Table 1 provides a summary of the symbols used in this section. The formulas presented in [GUR+06] for the complexity of the different algorithms are reviewed, and then they are applied to the case of interactive recalculation of quasi-static scenes.

Sphere related quantities	
$S = \{S_i\}$	Set of Spheres
$r_i = r_{i-1}Q = r_0 Q^i$	Radius of S_i
r_D	Radius of the sphere enclosing the mobile object
m_i	Number of recalculations of sphere S_i with point sorting.
$t_i = t n_R Q^{2i-2}$	Cost of recalculation of sphere S_i with point sorting
k	Number of spheres
Ray related quantities	
R	Set of Rays
$n_R = \#R$	Number of rays
$\tilde{n}_R = n_R \frac{r_D^2}{r_0^2}$	Number of rays which touch the mobile object
Time related quantities	
u	Ray Disc intersection time
t	Ray Sphere intersection time
T_R	Time to recalculate the spheres with point sorting
T_I	Time to intersect the disc against the inner sphere
Other symbols	
$0 < Q < 1$	Ratio of the radii of two spheres
$P = \{P_i\}$	Set of Irradiance samples
n_D	Samples in dynamic objects
n_S	Samples in static objects
$n_P = \#P = n_D + n_S$	Irradiance samples
d	Disc Radius

Table 1: Symbols used in this article

The basis of the study comes from the fact that the probability that a ray (with uniform distribution) which intersects a convex body, intersects a second convex body located inside the first is the ratio of the areas of the bodies. This result can be derived from results of integral geometry from Santalo [San02].

Theorem 1 *The performance estimates of [GUR+06] can be simplified to:*

- *Sphere cache:* $T = \frac{t\, n_R}{Q^2}\left(\frac{\sqrt[3]{n_P}-1}{1-Q}\right) + \frac{4}{3}\, u\, n_R \sqrt[3]{n_P}$
- *Disc Indexing:* $T = u\, n_R \sqrt[3]{n_P}$

Proof

Sphere cache [GUR+06] provides the following formulas for the performance of sphere cache:

$$T = T_I + T_R \quad (1)$$

where T is the total time to calculate sphere cache, formed by T_I, the cost of the inner sphere, and T_R, the cost of the rest of the spheres.

$$T_I = u\, n_P\, n_R \frac{d^2}{r_0^2} \quad (2)$$

$$T_R = \sum_{i=1}^{k} m_i\, t_i = \frac{t\, n_R}{Q^2}\left(\frac{\sqrt[3]{n_P}-1}{1-Q}\right) \quad (3)$$

 ISBN 978-80-86943-02-2

Proposition 1 *The optimal value of Q is 2/3.*

Proof To calculate the optimum value of Q, the derivative of Equation 1 with respect to Q is calculated, and $dT/dQ = 0$ is solved, yielding $Q = 2/3$. $(d^2T/dQ^2)(2/3) > 0$ means this is the minimum of the function.

Since the disc radius d should be the distance between irradiance samples to decrease variance (Section 2.1), Equation 3 can be simplified. Let's suppose the points are distributed in a regular grid. The distance between the points is d and there are $\sqrt[3]{n_P}$ points in a side of the grid. The radius of the sphere circumscribed to the grid is

$$r_0 = \frac{\sqrt{3}d\sqrt[3]{n_P}}{2} \tag{4}$$

Solving in d:

$$d = \frac{2r_0}{\sqrt{3}\sqrt[3]{n_P}} \tag{5}$$

and expanding d in Equation 2 results in

$$T_I = \frac{4}{3}\, u\, n_R \sqrt[3]{n_P} \tag{6}$$

Disc Indexing [GUR+06] gives the following cost for Disc Indexing:

$$T = u\, k\, n_R\, n_P/4^k \tag{7}$$

If the data structure is balanced, and the cost of going to the neighbor node is negligible with respect to the cost of doing the ray-disc intersections, the cost is:

$$T = u\, n_R\, n_P/4^k \tag{8}$$

Again, taking into account that the node should not be subdivided when the size of the side of the voxel reaches the distance between samples:

$$2^k = \sqrt[3]{n_P} \;\rightarrow\; k = \log_2 \sqrt[3]{n_P} \tag{9}$$

Substituting k in Equation 8 yields:

$$T = \frac{u\, n_R\, n_P}{4^{\log_2 \sqrt[3]{n_P}}} = \frac{u\, n_R\, n_P}{n_P^{2/3}} = u\, n_R \sqrt[3]{n_P} \tag{10}$$

These formulas are used in the following section to study quasi-static scenes.

3.2 Application To Quasi-Static Scenes

Previous results can be used to guide the design of hybrid algorithms for global illumination. Here a study of quasi-static scenes is presented.

A quasi-static scene is a scene in which most of the geometry is static; dynamic objects are relatively few and small. [GUL+04] provides an empirical study of these scenes. Illumination in these scenes can be updated efficiently by using the following approach: the first frame is calculated, and the rays are stored. Rays which intersect mobile objects are marked. On subsequent frames, the mobile object is located in a different location. Rays which intersected the mobile object in the previous frame or now do so are recalculated, and stored in two lists of 'old' and 'new' rays, respectively.

- Static points: Radiance is updated by subtracting the contribution of old rays and adding that of new rays.
- Dynamic points: The old radiance value is discarded and all the rays are used to calculate the new value.

A theoretical study of this algorithm follows.

Theorem 2 *The best method for DETP depends on whether the points are static or dynamic:*

- *Static: Disc Indexing is faster than Sphere Cache.*
- *Dynamic: Sphere Cache is faster than Disc Indexing for objects smaller than 35 % of the scene.*

Proof Section 3.1 shows that the probability that a ray which intersects a convex body, intersects a second convex body located inside the first is the ratio of the areas of the bodies.

If we call r_D the radius of the spherical bounding of the dynamic object, it can be seen that the number of recalculated rays is:

$$n_R^{new} \approx n_R^{old} \approx n_R \frac{r_D^2}{r_0^2} =_{def} \tilde{n}_R \tag{11}$$

If we call n_S the number of static points and n_D the number of dynamic points ($n_P = n_S + n_D$), the different possibilities can be studied.

Static points

The cost of sphere cache in this case is (from Equations 1, 3 and 6)

$$T = \frac{t\, \tilde{n}_R}{Q^2}\frac{\sqrt[3]{n_S}-1}{1-Q} + \frac{4}{3}\, u\, \tilde{n}_R \sqrt[3]{n_S} \tag{12}$$

where the first summand corresponds to cache misses and the second to the cost of the inner sphere. Taking into account that the ray-disc and ray-sphere intersection time are similar ($t \approx u$) and the fact that in the limit, for large n_S, $\sqrt[3]{n_S} - 1 \approx \sqrt[3]{n_S}$, and using $Q = 2/3$ as recommended above, the cost is

$$T = 8.08\, t\, \tilde{n}_R \sqrt[3]{n_S} \tag{13}$$

Disc Indexing traverses the index for old rays and new rays, so using Equation 10 for 2 $\tilde{n}_R$ rays gives a cost of

$$T = 2\, t\, \tilde{n}_R \sqrt[3]{n_S} \tag{14}$$

Disc Indexing is clearly the fastest option.

Dynamic points

Taking into account the previous results, the cost of sphere cache is

$$8.08\, t\, \tilde{n}_R \sqrt[3]{n_D} = 8.08\, t\, n_R \frac{r_D^2}{r_0^2} \sqrt[3]{n_D} \tag{15}$$

and that of disc indexing is $t\, n_R \sqrt[3]{n_D}$. If the two costs are used as the sides of an equation, and then solved with respect to n_D, the values of n_D on which each algorithm is optimal can be deduced.

$$8.08\, t\, n_R \frac{r_D^2}{r_0^2} \sqrt[3]{n_D} = t\, n_R \sqrt[3]{n_D} \tag{16}$$

Simplifying, (dividing by $t n_R \sqrt[3]{n_D}$)

$$8.08 \frac{r_D^2}{r_0^2} = 1 \tag{17}$$

Taking a square root and solving for r_D yields

$$r_D = 0.35\, r_0 \tag{18}$$

This means that if the dynamic objects' size is 35% of the scene, both algorithms are equally fast in average. Checking the cost of the methods for objects of size smaller and bigger than 35% of the scene shows that sphere cache is faster for smaller objects; while disc indexing is faster for larger objects.

Therefore, an optimal recalculation algorithm should use disc indexing for static points and ray cache for dynamic points, which by the definition of quasi-static scenes, are smaller than 35% of the scene.

4 VALIDATION OF THEORETICAL ASSUMPTIONS

This section contains a study of the error of the theoretical study for real scenes, in which the distribution of the rays is not uniform.

Two medium scale scenes were used to test the theoretical results. The first one, Atrium, can be seen in Figure 4. The second one, Expo, can be seen in Figure 5.

Figure 4: Atrium scene

4.1 Mean number of rays in the spheres

In the theoretical study, the assumption that the distribution of the rays was uniform was made. However, in real scenes the distribution depends on the position and intensity of the light sources and the objects. In practice, the light sources are located in order to provide

Figure 5: Expo scene

a sufficient illumination of the interesting part of the scenes. This increases the density of the photons in the zones where we are calculating the radiance. Two tests were performed with the Expo scene. The results can be seen in Figures 6 and 7. Even though the theoretical study underestimates the number of rays, the prediction is quite close to the real value in the middle and low levels of the sphere list, which corresponds to most of the time of the algorithm. Even though the rays are not uniformly distributed, as the spheres become smaller, the density of the rays becomes more uniform, and therefore the error decreases.

In order to increase the accuracy of the estimate for scenes in which the distribution of the rays is not uniform, a hybrid approach can be made. The algorithm can be run with only the upper levels of the sphere list, taking into account only cache misses and no density estimation. Then, the lower levels can be estimated accurately by the theoretical approach, since rays tend to be more uniform over smaller volume. In addition one may divide the number of rays by a large constant, and then multiply the results by this constant to decrease computation time of the estimate, as [RLMC03] suggests.

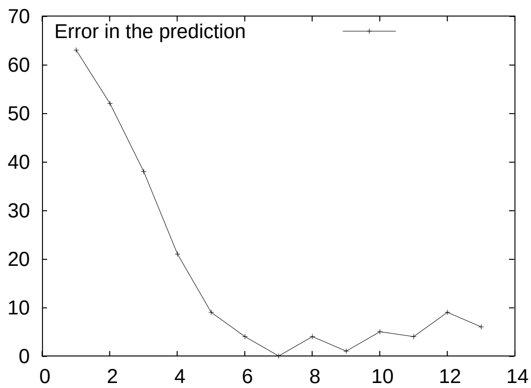

Figure 6: Percental error in the theoretical prediction of the Atrium scene, for each level

4.2 Sphere cache misses for uniform distributions

A program was designed which created a set of rays distributed uniformly in the surface of the unit sphere. Then a set of points distributed uniformly inside the

 ISBN 978-80-86943-02-2

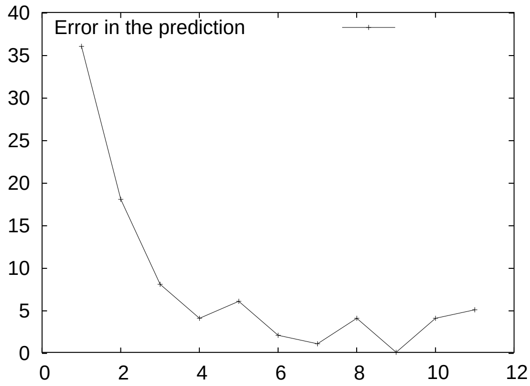

Figure 7: Percental error in the theoretical prediction of the Expo scene, for each level

unit sphere was created, and finally a set of normals distributed uniformly was assigned to each point.

This set of points was tested against the rays using Sphere Cache.

Uniform distribution of lines in a sphere To generate a ray following this distribution, two points distributed uniformly in the surface of the sphere must be generated. We use the method presented in [Sbe] and [TNSP98]. Then, the line which joins the two points follows a uniform distribution in the sphere, as [RWCS05] shows. The algorithm can be seen in Figure 8.

Algorithm 4.1 : POINTONSPHERESURFACE ()

$Z \leftarrow U[-1,1]$
$\phi \leftarrow U[0,2\pi]$
$\theta \leftarrow \arcsin(z)$
$X \leftarrow \cos(\theta)\cos(\phi)$
$Y \leftarrow \cos(\theta)\sin(\phi)$
return $(Point(X,Y,Z))$

Algorithm 4.2 : LINEINSPHERE ()

$a \leftarrow$ POINTONSPHERESURFACE()
$b \leftarrow$ POINTONSPHERESURFACE()
return $(Ray(a, b-a))$

Figure 8: Pseudocode which generates rays uniformly in the unit sphere

Results Figure 9 gives a graph with the number of cache misses at each level. The Lebesgue sorting is used, with a radius factor of 0.6, and the default 100 rays subdividing limit. 1024*1024 rays were used. The graph shows a steady increase in cache misses as the level increases, until level 9 is reached. Then there is a sharp decrease. According to theory, level 9 has 94.6 rays in average, so the probability of subdivision is very small (it is not zero because of the non-zero variance of the distribution of rays). The sharp decrease in cache faults in Figure 9 at level 10 is due to the fact that the mean number of rays is so small that few spheres are being subdivided.

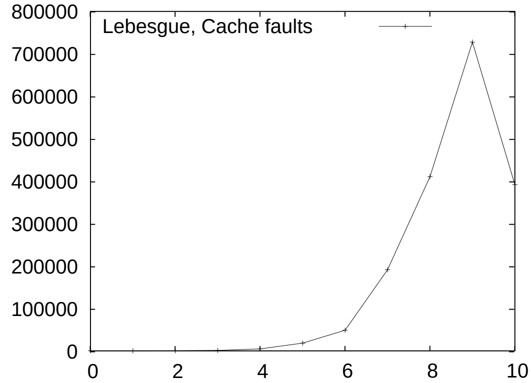

Figure 9: Cache Misses at each level. Lebesgue sorting.

Graph details In order to study in detail the graph, the radius factor was increased to obtain more data points.

Figure 10 shows the cache misses at each level for a radius factor of 0.9, for both the Lebesgue sorting and the Hilbert sorting. It can be seen that Hilbert has much smaller peaks. This is due to Hilbert having more spatial coherence than Lebesgue. In this graph a series of nearby local maxima followed by local minima. These points are located at sphere levels which correspond to a size of 2^{-k} of the scene (k=1,2,...), and are caused by the interaction between the sphere cache and the space filling curve sorting.

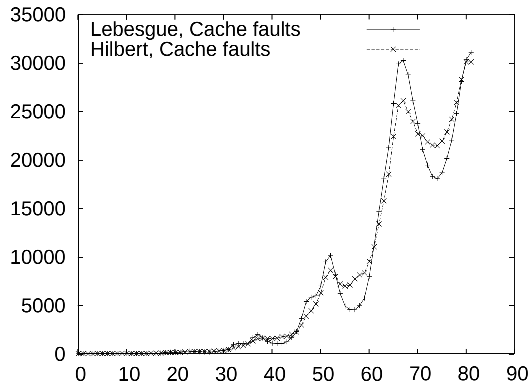

Figure 10: Cache Misses at each level for different sorting methods. $Q = 0.9$

5 CONCLUSIONS AND FUTURE WORK

A framework for the theoretical comparison of different montecarlo-based global illumination algorithms has been presented.

A theoretical study of the complexity of incremental recalculation of illumination has been performed for quasi-static scenes, using Sphere Cache and Disc Indexing. The study suggests a hybrid method which combines the benefits of both, using Disc Indexing for the static scene and Sphere Cache for the dynamic scene.

The ratio of the size of the bounding boxes of the mobile objects and the scene should be used to choose the method. If the dynamic objects have a size of less than 35% of the scene ($r_D/r_0 < 0.35$), Sphere cache should be used. (Theorem 2 in Section 3.2).

After some simplifications, the optimum value of the ratio of spheres in sphere cache has been obtained (Proposition 1 in Section 3.1), yielding $Q = 2/3$.

The error in the theoretical approach has been tested with real scenes, showing less than 5 % in most of the cases, even with non-uniform distribution of rays. Cache faults have also been studied numerically for a uniform distribution of rays, showing the interaction between sphere cache and space filling sortings.

5.1 Future work

Finding distributions of rays which model real scenes better would reduce the differences between the estimated number of rays and the practical results for the first levels of the sphere cache.

The shape of the graphs of the cache misses at each level shows that the interaction between sphere cache and space filling curves has an inner structure. Modeling this interaction is our next step in this research topic.

Other interesting future work is applying this study to Havran's Ray Maps. It is noteworthy that Ray Maps can be used in DETP mode, therefore making the study easier.

6 ACKNOWLEDGEMENTS

This work has been supported by the research project coded TIN2004-07672-C03-02 (Spanish Commission for Science and Technology). We thank all anonymous reviewers for their comments.

REFERENCES

[Arv86] James R. Arvo. Backward Ray Tracing. In *ACM SIGGRAPH '86 Course Notes - Developments in Ray Tracing*, volume 12, pages 259–263, 1986.

[BWS03] Carsten Benthin, Ingo Wald, and Philipp Slusallek. A scalable approach to interactive global illumination. In *Computer Graphics Forum*, volume 22. Eurographics, Blackwell, September 2003.

[DBMS02] Kirill Dmitriev, Stefan Brabec, Karol Myszkowski, and Hans-Peter Seidel. Interactive global illumination using selective photon tracing. 13^{th} *Eurographics Workshop on Rendering*, 2002.

[GUL+04] R. J. García, C. Ureña, M. Lastra, R. Montes, and J. Revelles. Interactive global illumination for quasi-static scenes. In *Proceedings of the CGI2004*, pages 128–131, June 2004.

[GUR+06] R. J. García, C. Ureña, J. Revelles, M. Lastra, and R. Montes. Density estimation optimizations for global illumination. In *WSCG'2006 Short Communications Proceedings*, 2006.

[GWS04] J. Günther, I. Wald, and P. Slusallek. Realtime Caustics using Distributed Photon Mapping. In *Proceedings of the Eurographics Symposium on Rendering*, 2004.

[HBHS05] V. Havran, J. Bittner, R. Herzog, and H.-P. Seidel. Ray maps for global illumination. *16th Eurographics Symposium on Rendering*, 2005.

[HP02] Heinrich Hey and Werner Purgathofer. Advanced radiance estimation for photon map global illumination. *Computer Graphics Forum*, 21(3):541–546, 2002.

[HP03] V. Havran and W. Purgathofer. On comparing ray shooting algorithms. *Computer and Graphics*, 27, Issue 4:593–604, August 2003.

[HPP00] V. Havran, J. Přikryl, and W. Purgathofer. Statistical comparison of ray-shooting effi ciency schemes. Technical Report TR-186-2-00-14, Institute of Computer Graphics and Algorithms, Vienna University of Technology, may 2000.

[Jen96] H.W. Jensen. Global illumination using photon maps. In *Rendering Techniques'96*, pages 21–30. Springer-Verlag, 1996.

[Kaj86] James T. Kajiya. The rendering equation. In *SIGGRAPH '86 Conference Proceedings*, pages 143–150, 1986.

[LAM00] J. Lext, U. Assarsson, and T. Moeller. Bart: A benchmark for animated ray tracing. Technical report, Dept. of Computer Engineering, Chalmers University of Technology, Goeteborg, 2000.

[LURM02] M. Lastra, C. Ureña, J. Revelles, and R. Montes. A particle-path based method for Monte-Carlo density estimation. *Poster at: 13th EUROGRAPHICS Workshop on Rendering*, 2002.

[PM92] S.N. Pattanaik and S.P. Mudur. Computation of global illumination by Monte Carlo simulation of the particle model of light. *Proceedings of 3rd Eurographics Rendering Workshop, Bristol*, 1992.

[RLMC03] J. Revelles, M. Lastra, R. Montes, and P. Cano. A formal framework approach for ray-scene intersection test improvement. In *WSCG'2003*, 2003.

[RWCS05] Jordi Rovira, Peter Wonka, Francesc Castro, and Mateu Sbert. Point sampling with uniformly distributed lines. In *Point-Based Graphics (PBG'05) Proceedings*, 2005.

[San02] L. Santalo. *Integral Geometry and Geometric Probability*. Cambridge University Press, 2 edition, October 2002.

[Sbe] Mateu Sbert. *The Use of Global Random Directions to compute Radiosity. Global Montecarlo Techniques*. PhD thesis, Universitat Politècnica de Catalunya.

[TNSP98] R. Tobler, L. Neumann, M. Sbert, and W. Purgathofer. A new form factor analogy and its application to stochastic global illumination algorithms. In *Rendering Techniques '98*, 1998.

[WGS04] I. Wald, J. Guenther, and P. Slusallek. Balancing Considered Harmful – Faster Photon Mapping using the Voxel Volume Heuristic. *Computer Graphics Forum*, 22(3), 2004.

[WH92] George J. Ward and Paul S. Heckbert. Irradiance gradients. In *Rendering Techniques '92*, Eurographics, pages 85–98, 1992.

[WHSG97] B. Walter, P. M. Hubbard, P. Shirley, and D. P. Greenberg. Global illumination using local linear density estimation. *ACM Transactions on Graphics*, 16(3):217–259, July 1997.

[WRC88] G. J. Ward, F. M. Rubinstein, and R. D. Clear. A ray tracing solution for diffuse interreflection. In *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 85–92, New York, NY, USA, 1988. ACM Press.