Vision, Modeling, and Visualization (2011)
Peter Eisert, Konrad Polthier, and Joachim Hornegger (Eds.)

Meshless Hierarchical Radiosity on the GPU

M. Zollhöfer[†1] and M. Stamminger[‡1]

[1] Computer Graphics Group, University Erlangen-Nuremberg

Abstract

Meshless radiosity is a radiosity method that is based on a point-based hierarchical discretization of the scene. This better decouples the runtime complexity from the geometric complexity of the scene and allows for an adaptive high-quality simulation of the diffuse global light transport. In this paper, we analyze the bottlenecks of this approach and examine the possibilities for an efficient and parallel implementation of this paradigm on the GPU. We show how by modifying the hierarchical data structures and the computation of the transport operator, a highly efficient GPU-based solution can be achieved which is by orders of magnitude faster and allows to compute high-quality global illumination solutions within seconds.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Radiosity

1. Introduction

Simulating the global light transport in complex environments is a computationally expensive task. Especially the indirect diffuse illumination, which has a huge impact on the realism of the generated images, is hard to simulate. The classical solution approach is to use radiosity-based methods. These are instances of the finite element method and are based on a mesh-based discretization of the scene. Meshless radiosity [LKSA, LZT*08] decouples the light transport from the geometric complexity of the scene by using a hierarchical and point-based basis. The computed solutions are inherently smooth and do not require post-processing. In this paper, we show how by modifying the underlying data structures and the computation process, we can shift the computations to a GPU and well exploit the GPU's computational horsepower. The core idea is to use an approximate hierarchy evaluation scheme based on a hierarchy which stores absolute values on all levels, in order to get rid of the expensive basis function evaluations in the computation of the transport operator. As a result, we can compute high-quality global illumination solutions within a few seconds.

In the remainder, we discuss the following: Section 2 gives a short history and introduction to radiosity-based methods. In Section 3, we reiterate the basic concepts of the meshless radiosity method. We discuss the existing bottlenecks and analyse the parallelism in the light transport step. Our GPU implementation is discussed in Section 4. The approximate evaluation of the incident illumination and how to efficently propagate illumination down the modified hierarchy is discussed in Section 5. Timings and results are presented in Section 6. A summary and an outlook is given in Section 7.

2. Related Work

The classical radiosity method [GTGB84] was presented in 1984. The authors construct a discretized representation of the scene by subdividing it into a set of patches. By using a constant basis function per patch, the global illumination solution is restricted to a finite dimensional subspace. To describe the discrete energy transport, form factors are introduced to measure the fraction of energy which is transported between patches. This allows to state the energy distribution equilibrium as the solution of a system of linear equations. The main bottleneck of this approach is the computation of the form factor matrix, because of the quadratic number of form factors.

In 1991, the hierarchical radiosity method [HS91] has been introduced. By using an adaptive refinement approach, the authors can solve the light transport up to a user-specified

† michael.zollhoefer@informatik.uni-erlangen.de
‡ marc.stamminger@informatik.uni-erlangen.de

precision. The light transport is based on a hierarchical patch-based representation of the scene. The coarsest hierarchy level is given by the primitives of the scene and finer levels are obtained by recursive splitting. This makes the number of required form factor computations linear in the number of finest level basis functions. Because the number of form factors still quadratically depends on the coarsest level, the scene complexity is the limiting factor.

To eliminate this dependence, clustering based algorithms have been introduced. These approaches can be classified in two categories. Volume clustering methods [SAG94, Sil95, GH96] build a hierarchy of volume clusters on the primitives of the scene. Such a cluster abstracts from the contained primitives. The second class are face clustering methods [WHG99]. Face clusters are groups of primitives which partition an object. By using an automatically generated multi-resolution hierarchy of such clusters, the energy transport between two sets of primitives can be approximated using clusters if the introduced error is small. Clusters are constructed on top of the geometry which allows to decouple the runtime complexity from this parameter. This allows to process scenes with higher complexity.

Besides these methods, there are numerous other approaches. The integration of glossy transport into the radiosity framework has been proposed and discussed in various papers [DS95, RT90]. In [TM93, Zat93], the authors investigate the use of higher order basis functions. By construction, those generate smoother solutions. The authors of [GSCH93] propose to use a wavelet basis, this directly leads to a hierarchical formulation of the radiosity problem.

3. Meshless Radiosity

The meshless radiosity method [LKSA, LZT*08] uses the so-called meshless hierarchy to discretize the radiosity equation. Because of its point-based and hierarchical nature, the resolution of the light transport computations can be locally adapted to eliminate costly computations. By decoupling the runtime complexity from the geometric complexity of the scene, the meshless radiosity method can much better handle detailed scenes with small triangles – a setup that typically generates problems with the mesh-based radiosity solvers described in Section 2. Typically, these approaches require a final gathering step to remove artifacts, because the computed solutions inherently contain the structure of the discretization scheme. In contrast, the meshless radiosity method generates smooth solutions which allow for a direct visualization.

A meshless hierarchy consisting of m levels represents the incident illumination at a point $\mathbf{p}$ in the following way:

$$F_m(\mathbf{p}) = \sum_{l=0}^{m-1} \sum_{j=0}^{N_l} \alpha_j^l \cdot B_j^l(\mathbf{p}).$$

Thereby, the l-th level is given by N_l basis functions B_j^l. The basis functions have compact support and each hierarchy level has to cover the entire scene. Normally, a few hundred basis functions are used on the coarsest level. On finer discretization levels a steadily increasing number of basis functions is used. The coefficients α_j^l of the basis function expansion encode the illumination. The evaluation point $\mathbf{p} = (\mathbf{x}, \mathbf{n})$ is specified by its position $\mathbf{x}$ and the associated surface normal $\mathbf{n}$.

The basis functions are normalized versions of weight functions w_j^l which ramp the influence of the associated coefficient smoothly from one to zero, depending on the distance to the evaluation point:

$$B_j^l(\mathbf{p}) = \frac{w_j^l(\mathbf{p})}{\sum_{i=0}^{N_l} w_i^l(\mathbf{p})}.$$

Distance to the evaluation point is measured considering both the Euclidean distance as well as the difference in the surface normal orientation. To allow a better approximation on finer hierarchy levels, the support of the basis functions is successively reduced. A detailed discussion of the used weight functions can be found in the original papers [LKSA, LZT*08].

One important design decision in the original work is to store absolute values only at the coarsest hierarchy level and to store differences on finer levels. Because each hierarchy level uses Shepard Approximation [She68], this leads to a Multi-level Shepard Approximation scheme. The coefficients representing the direct illumination in the scene can be computed using the directly incident illumination f_j^l at the basis functions:

$$\alpha_j^l = \begin{cases} f_j^l & l = 0 \\ f_j^l - F_{l-1}(\mathbf{p}_j^l) & l > 0 \end{cases}.$$

To compute the hierarchy, we use the approach described in the original papers [LKSA, LZT*08], which involves sampling the scene using ray tracing and generating multiple Poisson Disk distributions of increasing density by a Dart Throwing [Coo86] algorithm. Adjacent hierarchy level are connected by a father-child relation, where a node c of level $l+1$ is a child of a node p at level l, if the basis function associated with p at the center of c is non-zero. Figure 1 shows how the distribution of the basis functions looks like in the Crytek Sponza scene.

The radiosity equation is discretized using a reformulation in terms of irradiance, because irradiance is the smoother function [GSH94]. This allows to state the global illumination $\mathbf{e}_{in}$ as the solution of the following equation:

$$\mathbf{e}_{in} = \mathbf{e}_d + \mathbf{T}\mathbf{e}_{in}.$$

Thereby, $\mathbf{e}_d$ represents the directly incoming irradiance. By recursively expanding the given equation, one obtains the following form:

$$\mathbf{e}_{in} = \mathbf{e}_d + \mathbf{T}\mathbf{e}_d + \mathbf{T}^2\mathbf{e}_d + \dots$$

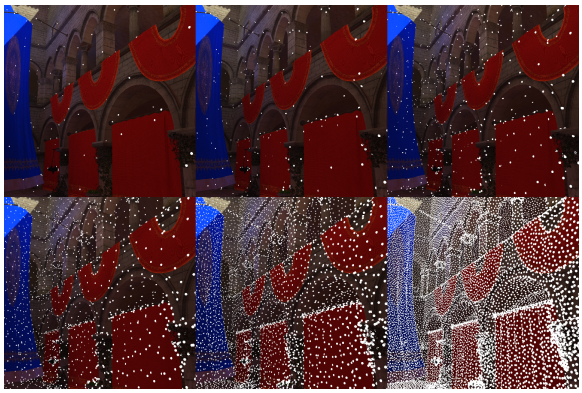

Figure 1: *Distribution of the basis functions on the six computed hierarchy levels in the Crytek Sponza scene.*

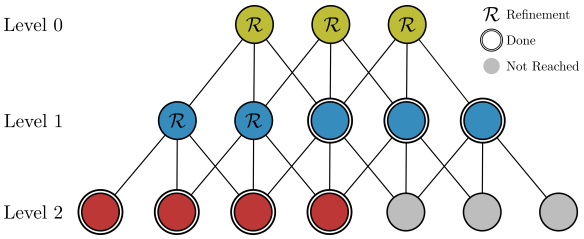

Figure 2: *Parallel computation of the transport operator: All equally colored gathering integrals can be evaluated in parallel. Gray integrals are not processed because of the adaptive refinement.*

To obtain an approximate global illumination problem, this equation is evaluated up to a finite number of bounces. The transport operator **T** can be evaluated in two different ways: By computing a set of transport links which can be reused in each bounce [LZT*08], or by directly transporting the "un-shot"-energy in each bounce [LKSA]. The second approach has the lower memory footprint, but the higher runtime. By precomputing transport links, the global illumination solution can be updated at interactive rates for static scenes if the direct illumination is changing [LZT*08]. Despite this fact, we focus on the second approach [LKSA], because it will allows us to apply rigid transformations to objects, modifiy the direct illumination and recompute a new global illumination solution within seconds.

In both approaches, gathering integrals are evaluated on an adaptive basis using an oracle which decides when to refine the transport. Each gathering integral is computed using Monte Carlo integration.

An interactive visualization of the computed global illumination solution can be obtained using basis function splatting. Care has to be taken that splats are not clipped away as long as the corresponding basis functions have influence on visible surface locations [LKSA]. The different hierarchy levels have to be splatted separately to allow for a correct normalization of the weight functions. Because this approach introduces a high amount of overdraw, Lehtinen et al. advice to compute a flat representation of the hierarchy before rendering.

4. Meshless Radiosity on the GPU

The most compute intensive part of the meshless radiosity method is the computation of the light transport operator **T**. Despite its hierarchical nature, the computation of all required gathering integrals can take several minutes. In this section, we describe a GPU-based implementation of the variant which directly transports the "un-shot"-energy. This implementation reduces the computation time from minutes to seconds. At the end of this section, we will analyze the bottlenecks of this implementation. Based on this analysis, we present a GPU-optimized version in Section 5, that makes even faster simulation possible.

All following implementations use the NVIDIA® OptiX™ ray tracing engine [NVI10b, PBD*10] and the NVIDIA® CUDA™ architecture [NVI10a, NVI10c] to leverage the GPU's computational power.

4.1. Parallelization

Each gathering integral in the computation of the transport operator is solved using the Monte Carlo Integration method. To obtain accurate results a high number of gathering rays is required (we decided to use 512 rays per integral). At each hit-point, the incoming "un-shot"-energy has to be reconstructed using the hierarchy. This requires to find and evaluate all influencing basis functions on all hierarchy levels.

The adaptive refinement starts by computing gathering integrals at each of the coarsest level basis functions. Next, an oracle is used to decide if these results have to be further refined. All these computations are inherently parallel allowing for a parallel implementation, because the computations of gathering integrals on the same hierarchy level do not depend on each other. We use a breadth-first traversal of the hierarchy and compute all gathering integrals on common levels in parallel. A parallelization across multiple levels is not possible, because the refinement oracle introduces a dependency to the previous level. Figure 2 illustrates the potential parallelism using a small artificial example.

Besides this per-level parallelism, all gathering rays in the Monte Carlo integration are independent and can be traced in parallel. Additionally, all further computations for the hit-point are independent and can be processed in parallel.

4.2. Implementation Details

Although the computation of the transport operator consists of many independent parallel tasks, the interplay has to

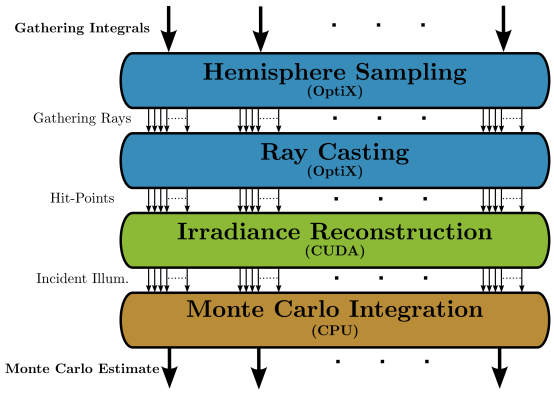

Figure 3: *GPU-based evaluation of the gathering integrals*

be managed well to achieve good performance. Before the transport operator can be computed in parallel, all required data structures have to be transferred to the GPU's device memory. This includes the basis functions and data structures for locating them. A GPU-based implementation of a median split kd-tree which stores each node in a compressed format requiring only 8 Bytes [Wal04] per node is used for range searching. We recursively split until each leaf contains exactly one basis function.

Thereafter, we compute the gathering integrals using the pipeline in Figure 3. In the *Hemisphere Sampling* stage, we compute all gathering rays in parallel using Halton sequences [WLH97]. As in the original work, we also apply a random rotation around the normal for each gathering integral to eliminate artifacts. The resulting rays are then cast in the *Ray Casting* stage to compute the hit-points with the scene and the associated surface normals. Both of these steps are implemented using OptiX. The *Irradiance Reconstruction* stage, reconstructs the "un-shot"-energy at the hit-points by evaluating the basis function expansion using the CUDA architecture. We store the per hit-point traversal stack of the used kd-tree in local device memory, as suggested by [ZHWG08]. Influencing basis functions are directly evaluated and used to compute the incident irradiance in the current bounce. Thereafter, we transfer the computed data to the CPU and compute the Monte Carlo estimates in the *Monte Carlo Integration* stage. At the moment we use a single-threaded implementation for this processing step, but it could also be mapped to the GPU.

On the coarsest hierarchy level, the Monte Carlo estimates can be directly used to update the coefficients of the basis function expansion. Because finer levels only store deltas, the incident irradiance already gathered at the coarser levels has to be subtracted. We compute this data using the newly computed coefficients and the *Irradiance Reconstruction* stage of the presented pipeline. As input the positions and surface normals of the basis functions have to be used. This means that the coefficients representing the currently gathered illumination have to be kept on the GPU and have to be updated each time a hierarchy level has been completely processed. Ideally, we would like to trace all gathering rays and perform all subsequent range searches in parallel. In practice, this is not possible because of hardware and memory constraints. Therefore, we handle 1000 gathering integrals in parallel which leads to $512k$ traced rays and an equal number of evaluations of the basis function expansion.

4.3. Problems

The meshless hierarchy represents the illumination in a delta encoded way. This allows for an adaptive computation of the light transport. But this type of storage format also has a negative effect on the performance:

- The computation of the light transport operator has to consider multiple hierarchy levels. This leads to many range searches and basis function evaluations. In addition, in a GPU-based implementation all basis functions and associated data structures have to be kept in device memory.
- To obtain the new delta coefficients, the incident illumination which is already represented by the coarser levels has to be subtracted. This step is also required when constructing the hierarchy.
- The hierarchy has to be flattened to allow for an efficient visualization.

5. Modified Hierarchy

In the following section, we redefine the meshless hierarchy allowing us to eliminate the described drawbacks and to simplify the light transport. We definitely want to maintain the adaptive nature of the original algorithm, because it significantly reduces the number of required gathering integrals. Based on the above observation, we decided to store absolute values on all hierarchy levels. This means, that our modified version of the hierarchy consists of multiple absolute Shepard Approximations of the illumination in the scene.

Therefore, the hierarchy represents not only one basis function expansion but multiple ones with varying level of detail. Each of the hierarchy levels can be interpreted as a flattened version of the original hierarchy up to that level. Now, only the selected sender level has to be kept in the GPU's device memory to compute the incident irradiance at the basis functions. In the original approach the whole hierarchy up to a selected sender level had to be used. Because we store absolute values on each level, we do no longer have to subtract the energy transported to the parents. This holds for the construction of the hierarchy as well as for the computation of the transport operator, therefore speeding up the computation time of both subtasks. A nice side-effect is that each hierarchy level is already a flat representation of the computed global illumination solution. Therefore, an efficient visualization of a selected level is directly possible.

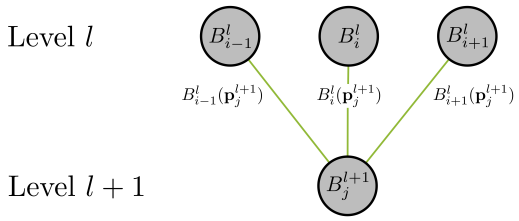

Figure 4: *Pulling energy down the hierarchy to keep it consistent*

This modification of the hierarchy is not for free, because it has an impact on the adaptive light transport algorithm and on the quality of the approximation. Both approaches lead to different basis function expansions and therefore have different approximation properties. We will discuss this in Section 6. To keep the absolute coefficients on all hierarchy levels consistent, we have to propagate the gathered illumination through the hierarchy, which means, that we have to pull the illumination gathered on the coarser levels down the hierarchy. To allow for an efficient implementation of this operation, we store the influence of the parent basis functions on their children in the parent relation. For a parent basis function B^l_i and a corresponding child B^{l+1}_j, the influence of the parent on the child is given by $B^l_i(p^{l+1}_j)$. Because the basis functions on a common hierarchy level are a partition of unity, the illumination at a child can be approximated as a linear combination of the incident illumination at its parents. This approach is illustrated in Figure 4. The used weights are a byproduct of the hierarchy construction, therefore precomputing them requires no additional computation time. Now, the refinement oracle can be based on the difference between the pulled down energy and the gathered illumination at the corresponding basis function.

5.1. Approximate Reconstruction

To further improve performance, we apply an approximate reconstruction scheme which speeds up the computation of the light transport. Our approach is similar to the one used by Christensen [Chr99] to speed up photon mapping. The basic idea is to reconstruct the illumination only at a fixed number of positions in the scene and use the illumination at the nearest photon as an approximation. Here, we use the basis function centers of a sender hierarchy level as evaluation points. Because we have eliminated the delta coefficients, the coefficients can be directly used as an approximation of the "un-shot"-energy. Since many of the hit-points computed during the light transport are spatially close, similar computations in the evaluations of the basis function expansion can be saved.

Instead of using the illumination at the nearest center, we can make a further approximation by using the basis function in the first traversed leaf of the used median split kd-tree. We reject basis function centers which have a strongly differing normal compared to the actual hit-point. Because the functions on the used sender hierarchy level are equally distributed and densely cover the surface of the scene, the introduced error is evenly spread and cancels out as in the photon mapping context. Such an optimization is not possible using the original hierarchy, because the basis functions on the first hierarchy levels are too far apart.

6. Results

We have tested the GPU-based implementation and the proposed acceleration techniques, using different test scenes and illumination conditions. The properties of the scenes and the used parameters can be found in Table 1. All renderings have been computed using the modified hierarchy and the approximate reconstruction based on kd-tree leafs. The renderings are gamma corrected using a gamma of 2.2, which even emphasizes artifacts. We show our results for the computed direct, indirect, and total illumination in Figure 6. For the Cornell Box, we used a spot light source. As stress test, we placed the Happy Buddha mesh in the Cornell Box, this scene consists of many small triangles and is illuminated using a spot light. To test our approach in a complex and textured environment, we have used the Crytek Sponza scene illuminated by a directional light source.

To visualize the computed global illumination solution, we opted to use the hybrid rendering approach proposed by [LKSA, LZT*08]. This means that the direct illumination is rendered using rasterization and the indirect illumination is splatted. We do this, because our modified hierarchy can not handle direct illumination as well as the original hierarchical approach. The illumination looks smoother, similar to the effect introduced by flattening the hierarchy for efficient visualization. As starting point for the computation of the indirect diffuse illumination our representation seems to be sufficient. Because the indirect diffuse illumination is an inherently smooth function, the modified hierarchy is a suitable representation for this kind of illumination. For an increasing number of basis functions, our as well as the original approach should converge to the same solution.

Because we compute a meshless hierarchy per scene object, we can rigidly transform them without having to recompute the point hierarchy. We transform the hit-points to the objects local coordinate frames to evaluate the corresponding basis function expansion. After a modification of the scene, the light transport has to be computed again. This includes sampling the direct illumination and computing the subsequent light bounces. The GPU-based implementation allows to compute a global illumination solution in a reasonable time frame, this allows for a modify-review editing cycle. Figure 5 shows the influence of different object positions on the computed global illumination solution.

Table 2 shows timings for three different version of the

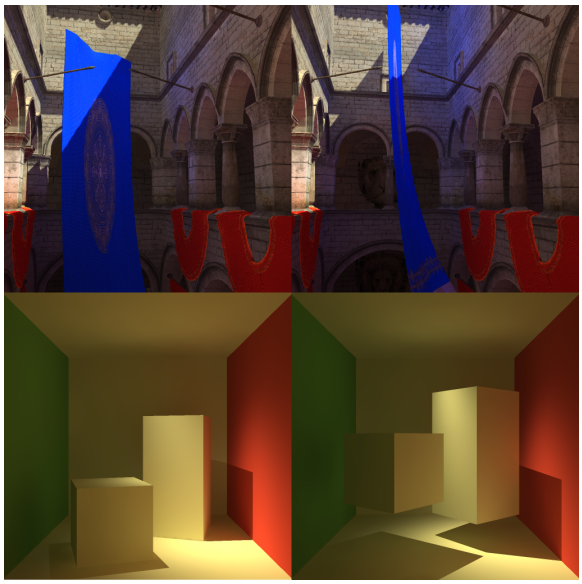

Figure 5: *Influence of object positions on the computed global illumination solution*

meshless radiosity algorithm. We have measured the performance of the original approach, the direct GPU implementation and the new hierarchy in combination with the approximate reconstruction (based on kd-tree leafs) for the three test scenes. The speedup is given relative to our implementation of the original method. All timings have been measured on an Intel® Core™ i7 860 CPU with 6GB RAM equipped with a NVIDIA® GeForce® GTX 480 graphics card with 1.5GB RAM.

For the timings, we performed the ray tracing computations in all approaches using OptiX to ensure comparable results. The light transport has been stopped after the first six bounces. We give the required time for the first three bounces, the number of totally computed gathering integrals and the computation time for the six bounce global illumination solution. Due to the adaptive refinement, the number of computed gathering integrals is much smaller than the total number of possible integrals. In average, a relative speedup of about 26.1x is purely achieved by using the GPU without introducing any simplifications to the algorithm. The additional speedup achieved by using the new hierarchy and the further optimizations leads to even smaller computation times.

7. Conclusion and Future Work

The meshless radiosity method is based on a point-based discretization of the scene. This decouples its runtime complexity from the primitive count. Therefore, this approach is well suited to compute the light transport in scenes consisting of many small primitives.

	Scene		
	Cornell	**Happy**	**Sponza**
Triangles	36	1.06M	279k
Hierarchy Levels	6	6	6
Sender Levels	3	3	3
Sender Level	4	4	4
Func. Coarsest Level	226	582	1438
Func. Finest Level	167k	288k	356k

Table 1: *Scenes properties and used parameters*

We have analyzed the computation of the light transport operator and have found that this task is inherently parallel. This parallelism is exploited using an implementation on the GPU. A modification of the point hierarchy simplifies the transport algorithm even further and allows for an approximate evaluation scheme. In the future, we would like to completely implement the described pipeline on the GPU, this will lead to further runtime improvements.

Acknowledgements

We thank Cornell University (Cornell Box), Stanford University (Happy Buddha) and Crytek (Crytek Sponza) for making their models publicly available. This work was partly funded by the German Research Foundation (DFG) under grant STA-662/3–1.

References

[Chr99] CHRISTENSEN P. H.: Faster photon map global illumination. *Journal of Graphics Tools 4* (1999), 1–10. 5

[Coo86] COOK R. L.: Stochastic sampling in computer graphics. *ACM Trans. Graph. 5*, 1 (1986), 51–72. 2

[DS95] DRETTAKIS F. S. G., SOLER C.: A clustering algorithm for radiance calculation in general environments. In *Eurographics Rendering Workshop* (1995), Springer-Verlag, pp. 196–205. 2

[GH96] GIBSON S., HUBBOLD R.: Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum 15* (1996), 0167–7055. 2

[GSCH93] GORTLER S. J., SCHRÖDER P., COHEN M. F., HANRAHAN P.: Wavelet radiosity, 1993. 2

[GSH94] GERSHBEIN R., SCHRÖDER P., HANRAHAN P.: Textures and radiosity: Controlling emission and reflection with texture maps. In *In Proceedings of SIGGRAPH 94* (1994), Press, pp. 51–58. 2

[GTGB84] GORAL C. M., TORRANCE K. E., GREENBERG D. P., BATTAILE B.: Modeling the interaction of light between diffuse surfaces. *SIGGRAPH Comput. Graph. 18*, 3 (1984), 213–222. 1

[HS91] HANRAHAN P., SALZMAN D.: A rapid hierarchical radiosity algorithm. In *Computer Graphics* (1991), pp. 197–206. 1

	CPU			GPU			GPU Approx.		
	Cornell	**Happy**	**Sponza**	**Cornell**	**Happy**	**Sponza**	**Cornell**	**Happy**	**Sponza**
1. Bounce	18.8s	8m 19s	11m 11s	1.0s	14.3s	24.4s	0.5s	3.3s	3.5s
2. Bounce	8.7s	2m 33s	37.5s	0.5s	4.0s	1.5s	0.2s	0.9s	0.4s
3. Bounce	2.7s	39.0s	22.6s	0.2s	1.3s	0.9s	0.1s	0.4s	0.3s
Integrals	14k	66k	47k	14k	66k	47k	15k	66k	46k
Total Time	34.6s	12m 1s	13m 19s	2.1s	20.7s	29.6s	1.1s	5.3s	5.1s
Speedup	1x	1x	1x	16.5x	34.8x	27.0x	31.5x	136.0x	157.7x

Table 2: *Timings of the three described versions of the meshless radiosity method*

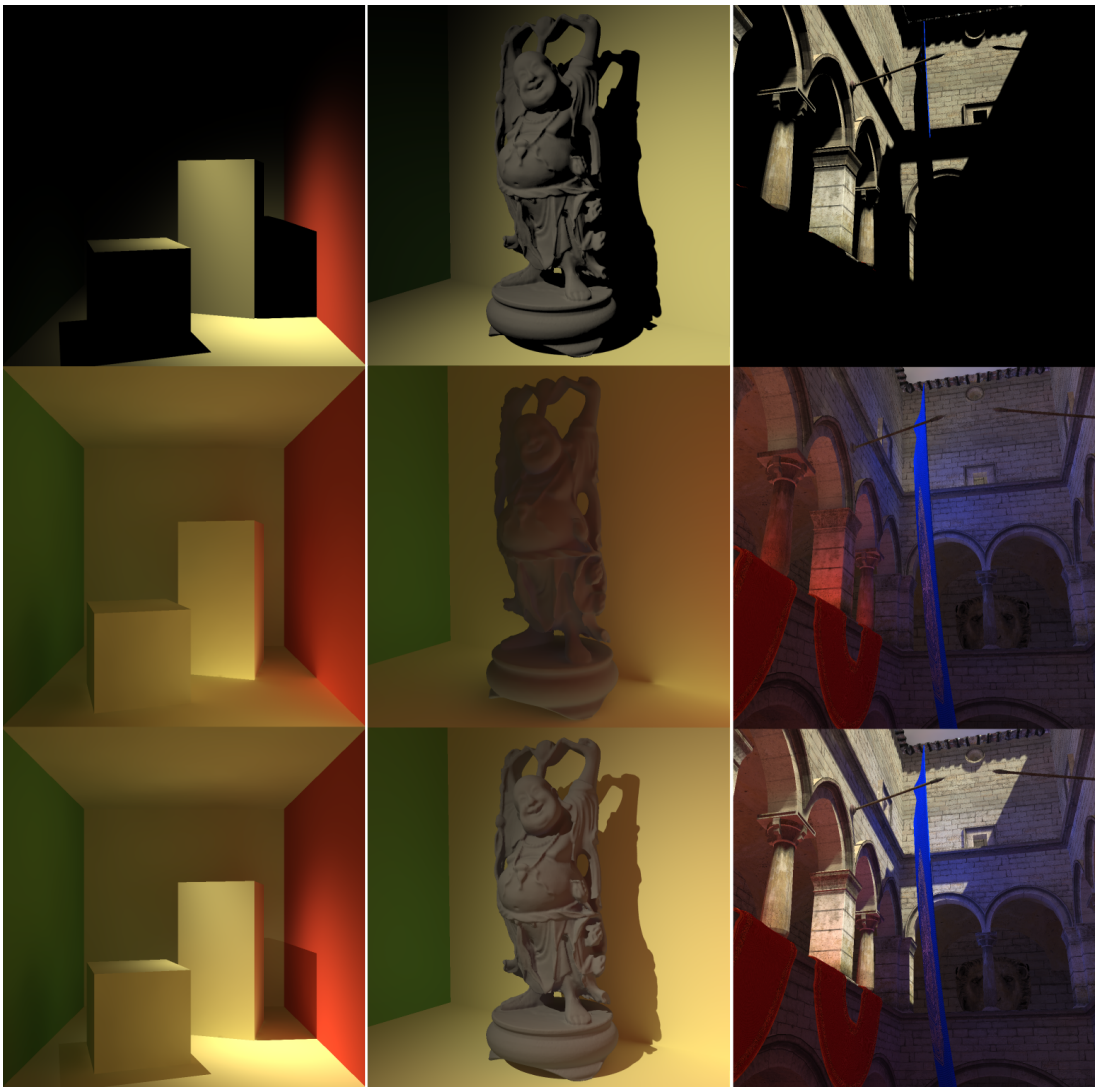

Figure 6: *Computed direct, indirect and total illumination for the Cornell Box, Happy Buddha and Crytek Sponza*

[LKSA] LEHTINEN J., KONTKANEN M. Z. J., SILLION E. T. F. X., AILA T.: Technical report tml-b7, publications in telecommunications software and multimedia, helsinki university of technology meshless finite elements for hierarchical global illumination. 1, 2, 3, 5

[LZT*08] LEHTINEN J., ZWICKER M., TURQUIN E., KONTKANEN J., DURAND F., SILLION F. X., AILA T.: A meshless hierarchical representation for light transport. *ACM Trans. Graph. 27*, 3 (2008), 1–9. 1, 2, 3, 5

[NVI10a] NVIDIA: *NVIDIA CUDA Programming Guide 3.0*. 2/20/2010. 3

[NVI10b] NVIDIA: *NVIDIA OptiX Ray Tracing Engine Programming Guide 2.0*. 3/29/2010. 3

[NVI10c] NVIDIA: *NVIDIA CUDA Reference Manual 3.0*. February 2010. 3

[PBD*10] PARKER S. G., BIGLER J., DIETRICH A., FRIEDRICH H., HOBEROCK J., LUEBKE D., MCALLISTER D., MCGUIRE M., MORLEY K., ROBISON A., STICH M.: Optix: A general purpose ray tracing engine. *ACM Transactions on Graphics* (August 2010). 3

[RT90] RUSHMEIER H. E., TORRANCE K. E.: Extending the radiosity method to include specularly reflecting and translucent materials. *ACM Transactions on Graphics 9* (1990), 1–27. 2

[SAG94] SMITS B., ARVO J., GREENBERG D.: A clustering algorithm for radiosity in complex environments. pp. 435–442. 2

[She68] SHEPARD D.: A two-dimensional interpolation function for irregularly-spaced data. In *ACM '68: Proceedings of the 1968 23rd ACM national conference* (New York, NY, USA, 1968), ACM, pp. 517–524. 2

[Sil95] SILLION F.: A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics 1* (1995), 240–254. 2

[TM93] TROUTMAN R., MAX N. L.: Radiosity algorithms using higher order finite element methods. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1993), ACM, pp. 209–212. 2

[Wal04] WALD I.: *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis, Computer Graphics Group, Saarland University, 2004. 4

[WHG99] WILLMOTT A. J., HECKBERT P. S., GARLAND M.: Face cluster radiosity. In *EUROGRAPHICS WORKSHOP ON RENDERING* (1999), Springer, pp. 293–304. 2

[WLH97] WONG T.-T., LUK W.-S., HENG P.-A.: Sampling with hammersley and halton points. *J. Graph. Tools 2*, 2 (1997), 9–24. 4

[Zat93] ZATZ H. R.: Galerkin radiosity: a higher order solution method for global illumination. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1993), ACM, pp. 213–220. 2

[ZHWG08] ZHOU K., HOU Q., WANG R., GUO B.: Real-time kd-tree construction on graphics hardware. In *SIGGRAPH Asia '08: ACM SIGGRAPH Asia 2008 papers* (New York, NY, USA, 2008), ACM, pp. 1–11. 4