Isotropic Clustering for Hierarchical Radiosity — Implementation and Experiences

Marc Stamminger, Wolfram Nitsch, Philipp Slusallek,
Hans-Peter Seidel
Computer Graphics Group, University of Erlangen
{stamminger,wmnitsch,slusallek,seidel}@informatik.uni-erlangen.de

Abstract

Although Hierarchical Radiosity was a big step forward for finite element computations in the context of global illumination, the algorithm can hardly cope with scenes of more than medium complexity. The reason is that Hierarchical Radiosity requires an initial linking step, comparing all pairs of initial objects in the scene. These initial objects are then hierarchically subdivided in order to accurately represent the light transport between them. Isotropic Clustering, as introduced by Sillion, in addition creates a hierarchy above the input objects. Thus, it allows for the interaction of complete clusters of objects and avoids the costly initial linking step.

In this paper, we describe our implementation of the isotropic clustering algorithm and discuss some of the problems that we encountered. The complexity of the algorithm is examined and clustering strategies are compared.

1 Introduction

Photo-realistic rendering requires a "realistic" illumination computation in a virtual world. Most professional rendering systems still use ray-tracing to compute the illumination in a scene. As many indirect illumination paths, especially via diffuse surfaces, are not considered by ray tracing, "tricks" such as ambient illumination or additional light sources simulating the indirect light are necessary. Of course, these tricks have to be designed by the user and achieving good results requires a lot of experience.

Physically-based rendering circumvents these problems using a physically-based model of light spreading in a scene. The indirect illumination is thus computed automatically, which requires more computation time, but less modeling time. Unfortunately, physically-based rendering still is only possible for scenes of moderate complexity and is normally restricted to diffuse surfaces (radiosity), but hybrid methods allow for adding some specular effects leading to very realistic pictures with reasonable effort in modeling.

Although there has been significant progress in global illumination research, the Isotropic Clustering Method of Sillion [6] is the first algorithm that can deliver first coarse results even for rather complex scenes in short time and also computes more accurate solutions in reasonable times. In the remainder of this section, we describe this algorithm, beginning with a description of the underlying Hierarchical Radiosity algorithm. In the following sections, we present our implementation and show results obtained by the algorithm. We then compare the solutions with pure Hierarchical Radiosity solutions and examine the impact of different clustering strategies on the computation time. Finally, we conclude.

1.1 Hierarchical Radiosity

As Hierarchical Radiosity is now well established, we only give a brief description here, mainly to introduce terminology.

In Hierarchical Radiosity a multi-resolution representation of the radiosity is used. This allows for the adaptive use of an appropriate subdivision level to compute the light transport between two patches. Distant patches can exchange light on the root level, whereas patches that are closer to each other transport light on a finer subdivision level. In the following, we distinguish between *initial patches*, which are the patches defined in the scene description and are subdivided hierarchically during the run of the algorithm, and *patches*, which are nodes in the subdivision tree of an initial patch.

In an iteration step the algorithm considers all pairs of initial patches. The light transport between the pair of patches is approximated and tested against a threshold value. If the transported energy is small, the subdivision level is considered sufficient and a link is created between the patches. Light is then transported via this link. Otherwise, the algorithm decides to subdivide one of the patches (e.g. the larger one) and computes the light exchange between the new pairs recursively.

This algorithm performs well if the scene consists of a few large patches, which are finely subdivided. In that case, the complexity of the algorithm is $\mathcal{O}(n)$, where n is the number of resulting patches. Complex scenes, however, often contain many small patches to describe complex geometry. In that case, the outer loop of the algorithm, considering all pairs of initial patches, dominates the computation time. If the number of initial patches is k, the complexity of the algorithm is $\mathcal{O}(k^2 + n)$. k^2 describes the costly "initial linking" phase; if n is not much larger than k computation time grows mainly with k^2. This problem is handled by the Isotropic Clustering method described in the following section.

1.2 Isotropic Clustering

The aim of Isotropic Clustering [7, 6] is to get rid of the complexity of initial linking. This is achieved by continuing the hierarchy within the initial patches of Hierarchical Radiosity upwards, i.e. by clustering groups of initial patches, and grouping these clusters to other clusters until a complete hierarchy with a single root cluster is obtained. In this hierarchy, every cluster can contain other clusters or initial patches. Every patch, however, can only contain subpatches (see Figure 1).

The goal is to model the light transport not only between patches, but also between a patch and a cluster or directly between clusters. In contrast to convex patches, clusters can exchange light with themselves, so a link from a cluster to itself (self-link) is necessary. The initial linking now consists of building this single link from the root cluster to itself. This link is then refined as in Hierarchical Radiosity. The time for the initial link is constant, so there is hope to achieve an $\mathcal{O}(n)$ algorithm due to the hierarchical nature of the algorithm.

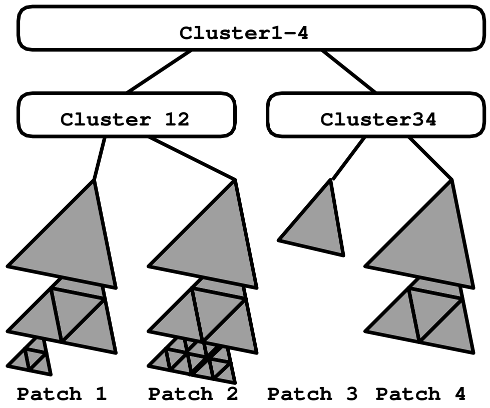

Figure 1: Hierarchy for Isotropic Clustering

In the case of Isotropic Clustering, a cluster is represented as an isotropically scattering volume. Methods to compute global illumination solutions in the presence of such isotropic volumes are described in [4]. As in surface radiosity, the outgoing radiance of such a volume can be described by its radiosity value and it is sufficient to compute irradiance, neglecting the directional distribution of incoming radiance. Using this volume radiosity, form factors between volumes and patches and between volumes and volumes can be computed. A detailed derivation of these form factors can be found in [7].

Using this approach, the complexity of the algorithm is $\mathcal{O}(n)$. However, due to the assumption of isotropic volumes, artifacts appear, when a cluster is illuminated from a preferred direction. As only the irradiance of a cluster is computed, and no information about the incoming direction is stored, light is pushed to the children independently of their orientation. The resulting artifacts can be avoided by pushing the incoming light to the children as soon as the cluster receives it. At that moment the direction of illumination is still known and can be used to distribute the light to the child patches, depending on the cosine between the patch's normal and the direction of illumination. Unfortunately, this increases the complexity of the algorithm to $\mathcal{O}(n \log n)$ [12], but in practice the additional effort can be neglected (see Section 3 below).

2 Implementation

We implemented the Isotropic Clustering algorithm within the Vision rendering system [3]. Vision is an object-oriented global illumination rendering system, developed at the Computer Graphics Group of the University of Erlangen [9, 11]. In this architecture, global illumination algorithms can be implemented as C++-classes, derived from the abstract Vision-class `Lighting`, using well-defined interfaces to access objects such as geometry, shaders, light sources etc. Implementations of several finite element and Monte-Carlo algorithms (Hierarchical Radiosity, Wavelet Radiosity, Wavelet Radiance, Path Tracing, Bidirectional Estimators...) are part of Vision and can be plugged into the system to compute a global illumination solution of a scene.

2.1 Patches and Cluster

As proposed by Sillion, Patches and Clusters are both derived from the same base class `Object`. Every object contains a number of sub-objects. For a patch the sub-objects are the patches produced by subdivision, whereas the sub-objects of a cluster can be other clusters or patches. By providing the "equivalent area" for clusters [8], the major part of the implementation does not have to differentiate between patches and clusters.

2.2 Clustering

We implemented two methods to find a clustering hierarchy. One is to build clusters using a uniform octree subdivision of the bounding box of the scene. Every patch is then put into the smallest cluster completely containing it [7]. To restrict the depth of the clustering, a maximum octree subdivision level is used. The impact of this maximum level on the computation time is examined in the following section.

Another possibility is to exploit the hierarchical structure of the input model. In Vision, the scene is described as a file in RenderMan RIB-format. This file is converted to a directed acyclic graph, which reflects the hierarchical structure of the scene description. We used a special RenderMan container object to allow the user to construct the cluster hierarchy by hand. We show in the following section that computation can be significantly accelerated by using these "hand made" clusters. As many rendering systems have more information about the scene than just a list of patches and can use this information to generate such a hierarchy, this clustering approach surely is an interesting alternative.

2.3 Refinement

In our implementation an energy-based oracle is used to decide about the refinement of links [2]. A link is refined if the product BFA is greater than a user-selected error threshold, where B is the radiosity of the sender, F the form factor and A the area of the receiver. Thus, the error threshold ϵ of the algorithm is in units of W (power) not W/m^2 (radiosity). Unfortunately, this threshold is not scale invariant, i.e. if the geometry of the scene is scaled uniformly, the threshold value must be scaled accordingly to obtain the same solution. On the other hand, this oracle avoids potentially infinite subdivision near singularities [10]. For that reason a minimal area threshold is usually not necessary anymore.

To handle partial visibility between objects, we use the same oracle as for links between totally visible objects but with a stricter error threshold $\epsilon_{\text{partial}} < \epsilon$. In practice, a value of $\epsilon_{\text{partial}} = \epsilon/10$ produces good results.

2.4 Visibility

For form factors the average visibility between the two objects must be approximated. This can be done by casting a number of rays between the two objects and approximating the visibility term by the fraction of unoccluded rays. For our oracle it is sensible to adapt the number of test rays to the transported energy. We use a number of rays that is proportional by a factor f_{vis} to the fraction of transported energy (assuming no occlusion) and the energy threshold value ϵ. This means that every ray decides approximately

Algorithm	time	initial Patches	patches produced	links produced
HR	1 350s	2 008	2 940	4 035 032
IC	17s	2 008	2 926	39 593

Table 1: Comparison of computation times for the seminar room scene with Hierarchical Radiosity (HR) and Isotropic Clustering (IC)

about the same amount of energy transport. Furthermore, if the algorithm detects total visibility or occlusion between two patches, the algorithm can "rely" on this result and transform it to small subdivisions of the patches, which is not always true for a fix number of visibility rays.

To avoid artifacts and an exploding number of test rays, we give a lower and an upper bound on the number of visibility rays. In practice, the lower bound turns out to have a significant impact on the overall computation time. This shows that the majority of visibility tests is made for low energy interactions and emphasizes the importance of a proper selection of the number of visibility samples.

3 Results

In this section we describe results obtained by our Isotropic Clustering implementation, compare it to Hierarchical Radiosity and examine different settings for Isotropic Clustering.

3.1 Comparison to Hierarchical Radiosity

In order to compare the Isotropic Clustering (IC) implementation with the standard Hierarchical Radiosity (HR) algorithm, we used a scene of moderate complexity, a *seminar room* with about 2 000 initial patches. We rendered the scene with both algorithms using the same energy threshold and no minimum area bound. Of course, the comparison is a bit unfair, because the resulting error in the Isotropic Clustering solution is probably larger.

Table 1 shows the times for computing a solution with IC and HR, each using the same energy threshold value, and the resulting number of links and patches. The benefit from the clustering strategy is dramatic. An objective measure for the error strongly depends on the reconstruction method and was omitted due to the lack of a good reference solution. However, as can be seen in Figure 2, both solutions are visually very similar.

The reason for the different computation times is obvious: in the initial linking phase in HR more than 4 million initial links have to be considered. Compared to initial linking the number of links produced in the following refinement phase can be neglected. In [1] a method is described to accelerate the initial linking step in HR by dropping initial links with very little energy and gathering the missing energy from the complete environment. This can be interpreted as using one global cluster and replacing low energy links by links to that cluster. However, the efficiency gain cannot compete with the one achieved by Isotropic Clustering.

By avoiding the missing initial linking phase Isotropic Clustering is also able to produce first, coarse result very quickly. Figure 3 shows a picture of an *office scene* with 4 000 initial patches. The coarse result was already obtained after 13.3 seconds. But also for finer resolutions as in Figure 4 the algorithm performs well (388 seconds).

Clustering	time	init. Patches	final Patches	Clusters	Links
OCTREE3	119.2s	4 038	6 858	52	679 400
OCTREE4	90.9s	4 038	6 855	214	374 934
OCTREE5	95.3s	4 038	6 852	582	374 329
OCTREE6	97.7s	4 038	6 879	1 024	374 516
USER	57.2s	4 038	6 828	605	105 087

Table 2: Comparison of computation times for the office scene with different clustering strategies

A comparable result of HR could not be computed due to memory limitations: using 16 bytes per link $4000 \times 4000 \times 16$bytes $= 256$MByte would be required, whereas the Clustering implementation is satisfied with less than one MByte for the links and about the same amount for the subdivision trees.

3.2 Clustering Strategy

As already mentioned, the clustering hierarchy being used has an important impact on both the speed and the quality of the computation. In Table 2 we compare different clustering strategies and the resulting rendering times for the office scene consisting of about 4 000 initial patches. For OCTREE3 - OCTREE6 the uniform octree clustering method was used, where the maximum clustering depth varied from three to six. We also tried to compute a solution for depth two, but we canceled the job when the machine started swapping continuously. For USER the hierarchy, as it was part of the scene description, was used to build the clusters.

The advantages of the user defined clustering over the simple octree clustering technique are significant. Moreover, the octree solutions exhibit typical artifacts that arise from patches belonging to the same object, being distributed over several clustering levels. A comparison of the resulting images for OCTREE5 and USER can be seen in Figures 5 and 6.

The large benefit of using better clusters emphasizes the importance of finding adaptive, but also fast, clustering strategies.

3.3 Pushing Irradiance

As mentioned before, artifacts due to the isotropy assumption of the clusters can appear, if a cluster receives energy from certain preferred directions. To avoid these artifacts, a cluster can "push" irradiance from one link to its children immediately during gathering, when the direction of incidence is still known, instead of pushing the complete irradiance accumulated over all links. In the terminology of Smits et al. [12] this is equivalent to replace the cheaper β-Links, which require constant time, by the more expensive, but also more accurate α-Links, requiring time proportional to the size of the cluster. According to [12] the use of α-links increases the complexity of the algorithm to $\mathcal{O}(n \log n)$.

To examine the influence of that theoretical raise in complexity, we computed the office scene with and without immediate pushing of irradiance. The difference in computation time was marginal (about 5%), but the improvement in quality is significant (compare Figures 6 and 7).

4 Conclusion and Future Work

In this paper we have described an implementation of Sillion's Isotropic Clustering algorithm for radiosity computations and examined the algorithm using several test scenes. In practice, the algorithm turned out to be a very stable method that delivers first coarse solutions very quickly and can compute accurate solutions by orders of magnitude faster than standard Hierarchical Radiosity.

Furthermore, we have demonstrated that the performance strongly depends on the strategy used to build the cluster hierarchy. Research on finding good clustering hierarchies is going on, applying the results to Isotropic Clustering surely is a promising field of research.

Another interesting point is visibility computation. In both IC and HR the major computation time is used for visibility tests. Using clusters as "grey boxes" may be a solution to achieve significant performance gains [5].

References

[1] Nicolas Holzschuch, Françoise Sillion, and George Drettakis. An efficient progressive refinement strategie for hierarchical radiosity. In *Fifth Eurographics Workshop on Rendering*, pages 343–357, Darmstadt, Germany, June 1994.

[2] Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and error estimates for radiosity. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94*, pages 75–84, 1994.

[3] W. Nitsch. Clustering für Hierarchische Radiosity. University of Erlangen, 1996. Studienarbeit.

[4] H. E. Rushmeier and K. E. Torrance. The zonal method for calculating light intensities in the presence of a participating medium. In *Proceedings SIGGRAPH '87*, pages 293–302, 1987.

[5] François Sillion and George Drettakis. Feature-based control of visibility error: A multi-resolution clustering algorithm for global illumination. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 145–152, August 1995.

[6] Francois X. Sillion. Clustering and volume scattering for hierarchical radiosity computations. In *Fifth Eurographics Workshop on Rendering*, pages 105–117, Darmstadt, Germany, June 1994.

[7] Francois X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics*, pages 240–254, September 1995.

[8] Francois X. Sillion, George Drettakis, and Cyril Soler. A clustering algorithm for radiance calculation in general environments. In *Sixth Eurographics Workshop on Rendering*, pages 159–171, Dublin, Ireland, June 1995.

[9] Ph. Slusallek. *Vision – an architecture for physically based rendering.* PhD thesis, University of Erlangen, 1996.

[10] Philipp Slusallek, Michael Schröder, Marc Stamminger, and Hans-Peter Seidel. Smart links and efficient reconstruction for wavelet radiosity. In *Sixth Eurographics Workshop on Rendering*, pages 251–263, Dublin, Ireland, June 1995.

[11] Philipp Slusallek and Hans-Peter Seidel. Vision: An architecture for global illumination. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):77–96, March 1995.

[12] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, pages 435–442, 1994.

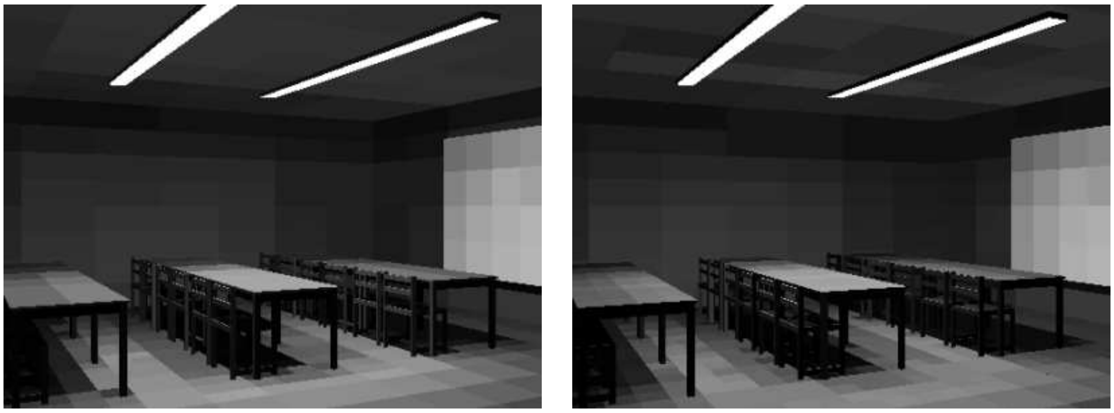

Figure 2: Solutions of seminar room scene obtained by Hierarchical Radiosity (left) and Isotropic Clustering (right)

Figure 3: Fast and coarse solution of the office scene (13.3s)

Figure 4: Fine solution of the office scene (388s)

Figure 5: Solution obtained with octree clustering of depth 5

Figure 6: Solution obtained with user defined clustering

Figure 7: Solution obtained without irradiance pushing