Volume xx (200y), Number z, pp. 1–10

Fast Multipath Radiosity using Hierarchical Subscenes

F. Castro,[1] M. Sbert[1] and L. Neumann[2]

[1] Institut d'Informàtica i Aplicacions, Universitat de Girona, Spain
[2] Researcher of the ICREA at the University of Girona, Spain

Abstract

This paper presents an efficient acceleration technique for the global line radiosity Multipath method. In this approach, the scene is subdivided in a hierarchy of box bounded subscenes, the boxes subdivided in a grid of virtual patches which store angular information. A new recursive (according to the hierarchy of subscenes) function allows to execute the Multipath algorithm at different levels of the hierarchy. A speed up factor up to $3-4$ *has been obtained on some of the tested scenes.*

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

1. Introduction

Radiosity techniques [2,3] estimate the illumination in a diffuse scene discretized in patches by solving a linear system of equations (one equation and unknown per patch). The Multipath algorithm [8] is a Monte Carlo technique that solves the radiosity problem. This technique uses global random lines to simulate the exchange of radiant power among the patches, and it can be seen as a random walk where the transition probabilities are the form factors. (Note: In the paper we use the terms "radiant power" and "power" with identical meaning.)

In the present paper we introduce an acceleration technique for the Multipath algorithm. This acceleration technique is based on the use of a hierarchy of bounded-by-boxes subscenes. The hierarchy allows to execute the Multipath algorithm at different levels, involving a better exploitation of the global lines.

The different levels in which the Multipath algorithm is executed are communicated by the surfaces of the bounding boxes, considered as *virtual walls* [1], which accumulate radiant power entering or leaving the subscenes. These virtual walls are subdivided in a grid of virtual patches and angular regions, allowing to consider angular information at a given accuracy level. Moreover, we obtain, for each of these subdivisions, some additional information that permits to accelerate the process of casting the lines.

The better exploitation of the random lines, plus the acceleration in casting these lines, have produced a speed up factor between 2 and 3.7 on our tests.

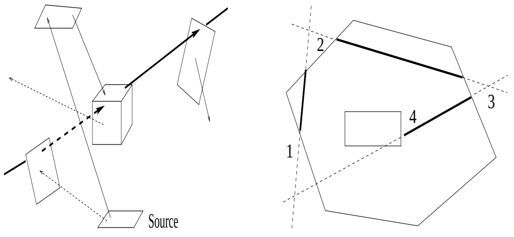
The paper is organized as follows. In section 2 we review the Multipath algorithm and review several approaches to the use of hierarchies of subscenes and virtual walls in the context of radiosity and global illumination. Section 3 presents the new algorithm. Results are introduced in section 4. Finally, section 5 summarizes the conclusions of the paper and some future lines of research.

2. Previous work

2.1. The Multipath algorithm

The radiosity Multipath algorithm was first described in [8]. It is a member of a family of methods called by different authors global Monte Carlo, global Radiosity or transillumination methods [7,9,10]. They use random global lines (or directions) to transport energy. Global lines are independent of the surfaces or patches in the scene, contrary to local lines, used in the classic methods, which are dependent on the patches they are cast from. Global lines can take advantage of all the intersections with the scene.

The Multipath method shows that it is possible to simulate a random walk from a global density of lines. Each global line will simulate the exchange of energy between several pairs of patches. In this way, each global line contributes to several geometric paths (Fig. 1a). We will briefly review the algorithm.



(a) (b)

Figure 1: *Multipath algorithm. (a) A global line (the thick one) simulates two paths, indicated with the continuous stroke and the dashed stroke (b) A path can contribute to the emission of radiant power from several patches. In the figure, path 1-2-3-4 simulates logical paths 1-2-3-4, 2-3-4 and 3-4.*

Global lines are intersected with the scene. For each line intersections are sorted by distance, resulting in an intersection list. Each patch keeps two quantities. One records the outgoing power accumulated, the other one is the unshot outgoing power. For every pair of patches along the intersection list, the first patch in the pair accumulates to its outgoing power its reflectance times the unshot power of the second patch (and vice-versa). This received unshot power times reflectance constitutes also the new unshot outgoing power. If a patch is a source, we keep also a third quantity, the emitted power per line exiting the source. Thus, if one of the patches of the pair is an emitter patch, we must add this emitted power per line to its unshot power. This "emitted per line" power is previously computed dividing the total emitted power of the patch by the forecast number of lines crossing the patch. This number of lines can be easily estimated since it is proportional to the area of the patch [11].

The advantages of the Multipath method are the following: First, all intersections of a line are used. Second, the radiant power transfer is bidirectional. And third, each path is used to transport different logical paths (see Fig. 1b), as with the covering paths [12]. But this method has also an important drawback. In first stages the distribution of radiant power is only possible from light sources, and so most of the lines cast in these first stages are wasted (lines that do not cross any light source). To avoid this behavior a process, called *first shot*, is done before running the Multipath algorithm. In this first shot, primary power is cast from the source patches by generating local lines that exit from the surface of each emitter patch. After that, the patches that have received some power will be the new sources instead of the original ones. Note that after this previous process the radiant power to be emitted is more widely distributed, decreasing the initial waste in global lines. A detailed comparison of the Multipath method against the classic radiosity methods can be found in [6].

2.2. Use of hierarchies of subscenes and discretized directions

Some early radiosity approaches estimate the form factors from a patch by constructing an hemicube around the center of the patch. This hemicube is uniformly subdivided in a grid, and the rest of patches in the scene are projected onto the hemicube. This involves a discretization of the directions over the patches, used in this case to compute the form factors.

A hierarchy of subscenes is used in [15], applied to the form factor computation. Subscenes are bounded by spheres, that are used to generate locally global densities of lines, improving the performance of the global Monte Carlo algorithm presented in [16]. This paper is a first approach to the use of *locally global lines*, that is, densities of lines adapted to different regions of the scene. The present paper exploits this idea in the context of the Multipath algorithm.

In [18] a parallel solution of the hierarchical radiosity method is introduced that allows to deal with very large environments. The algorithm consists of dividing the scene in groups of patches. During a single iteration power is bounced around between the patches within a group until convergence. No interaction with other groups occurs. After this internal balance, power is exchanged with other groups. This process can be repeated several times. This is somewhat similar to our algorithm in the sense that it alternates internal with external exchanges of radiant power.

In [19] a cluster hierarchy is also used. Transmittance has been defined here as the ratio of power that passes through a cluster in a particular direction. That is, if the cluster is totally opaque in this direction, transmittance is 0, whereas if some light can travel through the cluster in this direction transmittance is a positive value less than or equal to 1. This is used to estimate the form factors in very complex environments consisting of a great number of small objects (vegetation environments). This approach considers the visibility function between two surfaces (patches) i and j to be constant. If the only occlusions between i and j are included in a cluster C, the form factor can be estimated as a product of the unoccluded form factor and the directional transmittance through C in the direction d_{ij} (d_{ij} being the mean direction between patches i and j). The same kind of complex scenes are treated in [20], where similar objects are replaced by instances of the same element. Thus this algorithm substitutes a very large hierarchical radiosity problem by a collection of smaller hierarchical radiosity problems.

Virtual walls are used in [1], where they subdivide the original environment into local environments with the purpose of reducing the complexity of big environments. The local environments are treated separately and local results are transferred to the neighboring local environments using the virtual walls.

Adaptive representation of radiance is better suited for sharp-varying illumination, as noted in [26]. We will only represent secondary radiosity illumination, not sharp-varying direct illumination, thus a constant representation is well suited (and cheaper) in our case.

Use of virtual walls and boxes recalls the so called irradiance volumes [25], that present significant differences with our approach. Irradiance volumes store in each direction hemisphere integrals: the values of directional irradiance distribution. Irradiance and radiosity differ only by a constant factor, namely the reflectivity. We will use, for a given direction (or, in a discrete approach, for an angular region), radiances on the walls of the virtual boxes. Finally, we will gather from different directions the incoming radiance values to compute final radiosities on the real surfaces, but never on the walls of the virtual boxes.

Finally, in [22] they get rid of the patch-to-patch form factor computation by grouping patches in clusters and computing form factors between clusters. This results in encapsulating necessary information for form factor calculation and visibility estimation for each object, involving the use of angular information. This strategy allows to insert, move or remove objects in the scenes in a efficient way.

3. The new algorithm

This paper presents a variant of the Multipath algorithm based on the subdivision of the scene in a hierarchy of subscenes that allows running the Multipath algorithm not only for the whole scene but also for the subscenes, submitting each subscene to its own density of lines (referred to as *locally global lines*). This arises from the idea of using more lines where they are more necessary, allowing a better exploitation of the lines.

Each subscene is bounded by a box subdivided in a structure of virtual patches (VP) and angular regions (AR), that act as accumulators of incoming and outgoing power for the subscene, connecting the different levels of the hierarchy.

From the *locally global lines* generated in a pre-process for each subscene, we obtain some geometrical information that will be useful to reduce the cost of the algorithm. On one hand, we estimate the *transmittances*, which give information about the opaqueness of a subscene in a given direction, permitting to ignore its interior when executing Multipath at a higher level. On the other hand, we know the most-probably-intersected polygon for a line entering a subscene in a given direction. This allows to reduce the cost of the local lines.

We also save the intersection lists corresponding to every line, permitting to reuse the lines once and again in an iterative process without the costly computation of the intersections (only computed the first time).

Next we summarize the main issues of the new algorithm:

- Densities of lines adapted to each subscene.
- Use of virtual patches and angular regions.
- Directional information about the interior of the subscenes.
- Reuse of lines.

3.1. Hierarchy of subscenes

Several heuristics of subdivision can be applied to the scene. We have chosen an algorithm based on a bottom-up strategy [23], although other algorithms could be used. We next describe this strategy.

We deal with a set of non-grouped items (single objects or subscenes). Initially, this set is constituted just by the single objects. Subscenes are created by grouping the non-grouped items in successive iterations. Each new subscene is immediately added to the set of non-grouped items. The grouping criterion is based on the quotient of the areas of the boxes bounding the subscenes (or single objects) we want to group. We do group if the sum of these areas divided by the area of the resultant box is large enough (over a threshold). Incidentally, this quotient is half the average number of intersections for a random global line crossing the box [11]. Thus, this criterion establishes a minimum for the average number of intersections per global line.

Fig. 2 shows a simple example of a 2-level hierarchy, although the algorithm allows any number of levels. Note in the example that a subscene can contain other subscenes and/or single objects.

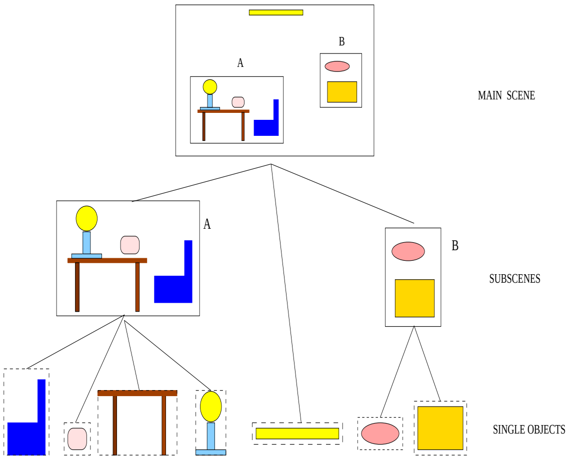


Figure 2: *A* 2*-level hierarchy of subscenes.*

3.2. Subdivision of the bounding boxes

We are using a hierarchy of bounding boxes. Each face of a bounding box is subdivided in a grid of *virtual patches* (VP), as seen in Fig. 3 (left). Moreover, each hemisphere over a virtual patch is subdivided in *angular regions* (AR), so that the directions over the virtual patch are discretized (see Fig. 3 (right)). Each angular region will act as an accumulator of undistributed incoming and outgoing power. So, the angular regions participate in the energy balance between the inside and the outside of the box, connecting in this way the different levels of the hierarchy.

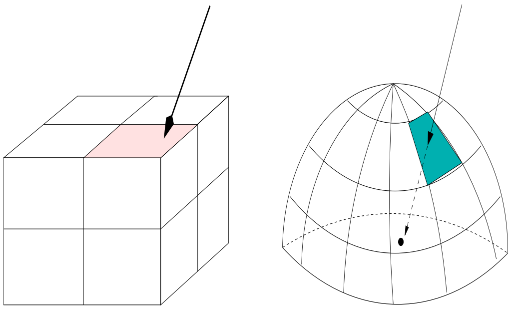

(a) (b)

Figure 3: *(a) Subdivision of the bounding box faces in a grid of virtual patches. (b) Subdivision of each hemisphere over the center of a virtual patch in angular regions.*

We have to note that, for a ray intersecting a bounding box, we can easily determine which virtual patch is intersected and, moreover, the corresponding angular region. Note that the angular region is determined just from the direction of the ray, and this does not depend on the ray being internal or external to the box.

3.2.1. Appropriate level of subdivision

The heuristic we have taken consists of establishing an upper threshold for the area of the virtual patches. Then, if a virtual patch is over the threshold, we divide it in four virtual patches. Moreover, this threshold is established in function of the threshold for the subdivision of the polygons (that is, the finer is the subdivision of the polygons in the subscene, the finer is the subdivision of the box faces). We have used, in our tests, a ratio factor of 16, i.e., the maximum area allowed for a virtual patch is 16 times bigger than the one for the patches in which polygons are subdivided. This is an *a priori* strategy of subdivision.

On the other hand, the subdivision of the directional hemisphere above the virtual patches has been experimentally fixed, after numerous test, to 24 angular regions: the longitude has been uniformly divided in 8 sectors (45 degrees each), and the latitude has been divided in 3 sectors according to cosine distribution (note that this cosine distribution guarantees the expected value of lines intersecting each angular region to be the same).

3.3. The pre-process

After building the hierarchy of subscenes and subdividing the boxes a pre-process is done in which global lines are cast for each subscene. For each line cast in subscene S, we have to compute its sorted-by-distance intersection list. We consider the intersections of the line against:

- Single objects (non-grouped in further subscenes) inside S.
- Inner subscenes. In this case we ignore their interior, considering only their virtual walls and angular regions. This strategy supposes an important reduction of cost, as described in section 4.
- Virtual walls of S, considering the corresponding angular regions.

The aim of this *pre-process* is to obtain information about the geometry of the subscenes. This information is used with two different purposes:

- **Transmittances**.
  For each angular region R in which virtual patches in the box of a subscene S are subdivided, its transmittance T_R (value between 0 and 1) represents the fraction of radiant power entering S in the direction of R that goes through the subscene finding no obstacle in its way. Transmittances give an idea about the opaqueness of a subscene in each direction. The closer is T_R to 0, the more opaque is S in the directions associated to angular region R, and *vice versa*. The transmittance T_R is estimated as

  $$T_R \approx \frac{n_R^d}{n_R} \qquad (1)$$

  where n_R^d is the number of lines crossing R and hitting no object, and n_R is the total number of lines crossing R. If no line crosses region R, we interpolate the value of the transmittances of the neighboring regions. This allows us to obtain a good approximation to the transmittances using only a moderate number of lines.
  Transmittances allow us to ignore the interior of the subscenes when executing Multipath at a higher level, having only a moderate loss of accuracy.
  Note that the idea of transmittance is somehow similar to the one introduced in [19]. However, in [19] they consider the transmittance of a cluster for a given direction, that is, they do not subdivide the walls of the cluster in virtual patches but just the directions above the cluster.

- **Polygon associated to an angular region**.
  The second benefit obtained from the pre-process allows to notably accelerate the first shot, that is, the expansion of primary power using local lines. It consists in computing, for each angular region in a bounding box, the polygon associated to this region. That is, if all lines cast in the pre-process (and hitting some polygon in the subscene) have the same polygon p as the nearest intersection when crossing region R, we associate polygon p to region R. This means that a line entering the box by this region R is likely to have polygon p as the nearest intersected polygon, and then it makes unnecessary to test the line against

the rest of the subscene. A variant of this strategy consists of considering not only a single associated polygon for region but several ones. This variant has been adopted in our implementation.

One of the main features of our algorithm is that we use a density of lines specifically generated for each subscene, so that this density is uniform in the context of the subscene but not in the context of the whole scene.

On the other hand, the intersection information is stored for its later use. We store, for each line and after sorting by distance, the identifiers of the intersected patches and angular regions. This information will be repeatedly used later in the iterative process, when running different levels of Multipath, avoiding to compute again the intersections.

3.3.1. Number of lines to cast in each subscene

The question about how many lines to cast in each subscene has an heuristic nature. After several tests using different percentages, we have adopted the criterion of destinating two thirds of the lines to the more external level (main scene) and one third to the subscenes. In the case of the subscenes, the percentage of lines to cast in has been established proportional to the quotient of areas used to group the boxes in 3.1. This means that boxes where the average number of intersections per global line is higher are favored.

Other heuristics have been experimented to establish the number of lines per box. It has been set proportional to the number of objects inside, taking the number of objects as a naive measure for the complexity of the subscenes. The results using this second heuristic have been similar to the ones with the first one (slightly better or worse depending on the tested scene).

3.4. The first shot

The distribution of primary power or *first shot* is done in the Multipath algorithm by using random local lines exiting from light sources. Only nearest intersected polygon has to be computed. This computation can be accelerated using the information obtained in the pre-process about polygons associated to angular regions. This means that if a local line gets into a subscene by angular region *R*, we only have to test the intersection of the line against the polygon (or polygons) associated to region *R* (if any). This allows to clearly reduce the cost of casting these local lines.

On the other hand, note that primary power is not accumulated in the angular regions but in the patches. Angular regions will act as accumulators in the following stages, but not in the first shot, where is preferred to exactly determine, for the sake of accuracy, the nearest intersected patch.

3.5. The iterative process

Once primary radiant power has been expanded, it is time to estimate the indirect illumination by executing the Multipath algorithm at different levels in successive iterations (usually 4 or 5 are enough). We accumulate incoming and outgoing power to/from the subscenes in the angular regions in which the bounding boxes are discretized. This allows the communication between the different levels. Note that no new intersections have to be computed, since we read and reuse the intersection lists obtained in the pre-process.

3.6. The algorithm

The new algorithm is presented in Fig. 4. Note that the recursive function MP deals with the exchange of power inside and between each subscene. In Fig. 5 we present the algorithm corresponding to this recursive function.

```
Generate hierarchy of subscenes
Subdivide each box in VP and AR
for each subscene S (including the whole scene)
  Cast global lines and store intersection lists
  if S is not the whole scene
    for each AR in S
      Compute transmittances and associated polygons
    endfor
  endif
endfor
First shot (accel. by polygons associated to each AR)
for each iteration (4 or 5 are usually enough)
    MP(whole scene) // recursive function
endfor
```

Figure 4: *The new algorithm.*

We have to remark the following points:

- Since we have cast lines in a pre-process, we know in advance the number of lines that cross each patch and angular region. This means that it is not necessary to forecast this number of lines to compute the power corresponding to each line (note that this forecast has to be done in the classic Multipath algorithm).
- Transmittances are used to establish the fraction of power crossing the subscenes in each direction, allowing to ignore the interior of the subscenes without losing much accuracy.

4. Results

4.1. Tested scenes

In our experiments we have employed several scenes. The ones whose results are included in this paper are described next. Results of other tests can be found in *ima.udg.es/∼castro/cgf.*

- **Scene *ROOM*.** This scene represents a room with a table, some chairs and a desk, and several small objects on the

```
MP(scene S)
Compute power per line for each patch in S
not included in any subscene of S
for each AR in the box of S
    Compute incoming power per line
    Set to 0 accumulated outgoing power
endfor
for each first-level subscene B inside S
    for each AR in the box of B
        Compute outgoing power per line
        Set to 0 accumulated incoming power per line
    endfor
endfor
for each line cast in S
    Read intersection list
    The line carries the power (from/to patches/angular regions):
        * Patches send unshot power + power per line
        * Angular regions of S send incoming power per line
        * Angular regions of the subscenes of S send
          outgoing power per line
        * Power leaving S is accumulated as outgoing
           power in the corresponding AR
        * Power crossing a subscene of S is accumulated as incoming
           power in the corresponding AR and attenuated by the
           corresponding transmittance
endfor
for each first-level subscene B inside S
    MP(B)            // recursive call
endfor
```

Figure 5: *The recursive function MP.*

table and the desk. The light source is a lamp with a lampshade stuck on the center of the ceiling.

- **Scene *BIG ROOM***. This scene involves several groups of pieces of furniture (chairs, tables, desks and shelves) and some small objects on the top of tables and desks. The light source is a lamp with a lampshade stuck on the center of the ceiling.
- **Scene *OFFICE***. This scene represents a room with some pieces of furniture (tables, chairs and one desk), lighted by a lamp with a lampshade stuck on the central part of the ceiling, plus two table-lamps. Several small objects are placed on the furniture.

The first scene (*ROOM*) involves a two-level hierarchy, whereas the scenes *BIG ROOM* and *OFFICE* result in a three-level hierarchy. According to the heuristic introduced in 3.2.1, we have subdivided (for the 3 scenes) the walls of the boxes in a grid of $8 \times 8 = 64$ virtual patches, and the hemisphere over each virtual patch in $8 \times 3 = 24$ angular regions.

Table 1: *Scene ROOM. Execution times and number of lines used to generate both images in Figure 8.*

n. of polygons	524	
n. of patches	10956	
	CLASSICAL	NEW
hierarchy levels		2
First shot	8 M lines	8 M lines
	195 sec.	117 sec.
Global lines	8 M lines	7.3 M lines
	294 sec.	123 sec.
Reusing lines		25 sec.
TOTAL LINES	16 M lines	15.3 M lines
TOTAL TIME	489 sec.	265 sec.
MSE	0.0095	0.0095
SPEED UP FACTOR		1.9

4.2. Comparing classic Multipath with the new algorithm

We have tested the new algorithm versus the classic version of Multipath for the previously mentioned scenes. In Fig. 6 we have plotted time vs. Mean Square Error (MSE) comparing the performance of the new algorithm with the classic Multipath for the 3 scenes. Some pictures obtained using both algorithms are also shown in Figs. 8, 10, and 9. Tables 1, 2, and 3 show details about the executions corresponding to these pictures. Note that in the first case (scene *ROOM*) we present two images with a similar quality, highlighting the lower execution time when using the new algorithm. Conversely, for both scenes *BIG ROOM* and *OFFICE* two images are presented obtained with similar cost, remarking the higher quality when using the new algorithm.

Summarizing, the new algorithm has produced a speed-up factor between 1.9 and 3.7 in our tests: 1.9 in scene *ROOM*, 3.1 in scene *BIG ROOM* and 3.7 in scene *OFFICE*. The gain increases if the complexity of the scene grows. This gain basically obeys to three different factors:

- **Reduction of the local lines cost**
  The information obtained in the pre-process about polygons associated to the angular regions allows to reduce the cost of the local lines involved in the first shot. Note that this information is obtained nearly for free, since lines cast in the pre-process are stored and reused later.
- **Reduction of the global lines cost**
  The use of the angular regions as accumulators of incom-

Table 2: *Scene BIG ROOM. Execution times and number of lines used to generate both images in Figure 10.*

n. of polygons	4712	
n. of patches	49124	
	CLASSICAL	NEW
hierarchy levels		3
First shot	5 M lines	9 M lines
	197 sec.	208 sec.
Global lines	5 M lines	8.6 M lines
	256 sec.	234 sec.
Reusing lines		29 sec.
TOTAL LINES	10 M lines	17.6 M lines
TOTAL TIME	453 sec.	471 sec.
MSE	0.373	0.115
SPEED UP FACTOR		3.1

Table 3: *Scene OFFICE. Execution times and number of lines used to generate both images in Figure 9.*

n. of polygons	30954	
n. of patches	58256	
	CLASSICAL	NEW
hierarchy levels		3
First shot	15.5 M lines	22.7 M lines
	587 sec.	606 sec.
Global lines	15.5 M lines	21.7 M lines
	719 sec.	643 sec.
Reusing lines		61 sec.
TOTAL LINES	31 M lines	44.1 M lines
TOTAL TIME	1306 sec.	1310 sec.
MSE	0.467	0.125
SPEED UP FACTOR		3.7

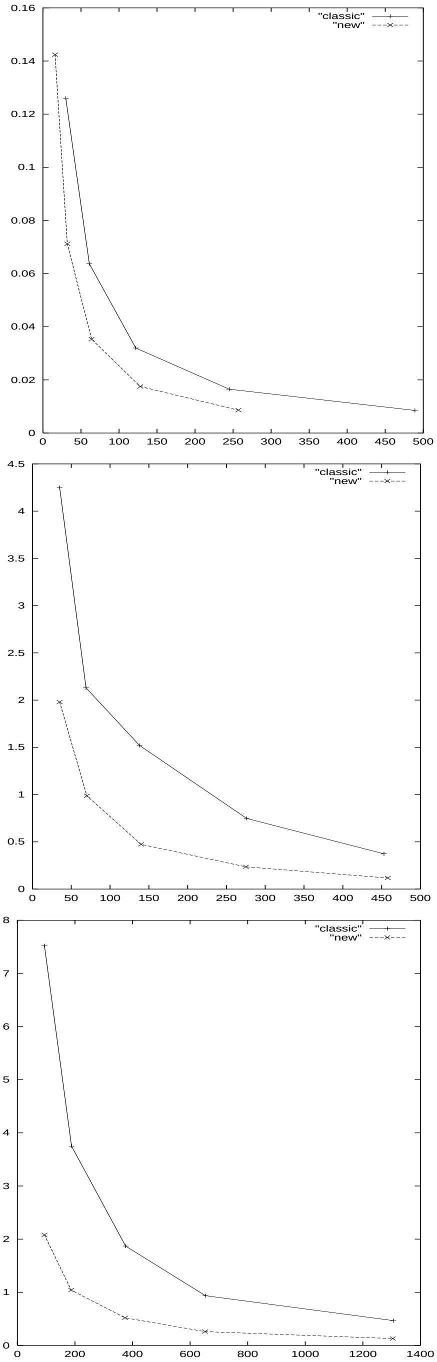


Figure 6: *Graphs of total running time (horizontal axis) vs. MSE (vertical axis) for (top) Scene ROOM. (mid) Scene BIG ROOM. (bottom) Scene OFFICE.*

Table 4: *Additional memory requirements for the experiments corresponding to Fig. 8, 9, and 10 using the new algorithm.*

	angular regions (internal memory)	inters. lists (external memory)
room	432 Kb	122.3 Mb
office	1080 Kb	148.2 Mb
bigroom	1944 Kb	155.0 Mb

ing and outgoing power and also the use of the transmittances allows to skip the interior of the subscenes when dealing with global lines. This produces a noticeable decrease in the cost of global lines, since it eliminates part of the intersection computations.

- **Reuse of the global lines**
  The iterative process in which the Multipath method is successively executed at different levels has a low cost, since we reuse the intersection lists corresponding to the global lines saved in the pre-process.

4.2.1. Memory storage

The use of the new algorithm involves an increase of the memory consumption with regard to the classic approach. On one hand we have to store, for each angular region, the incoming and the outgoing power. On the other hand we have to store the intersections lists (for each global line). This second storage has been done, in our experiments, in an external file instead of the internal memory for the sake of capacity (this fact involves a small and acceptable increase of cost, that has been already included in Tables 1, 3, and 2).

Table 4 shows the additional requirements of memory for the executions corresponding to the pictures presented in this article. We have distinguished in the table between memory required by angular regions and memory required by intersections lists. In case of angular regions, the subdivision level used in our experiments involves a total of $216Kb$ per virtual box. With reference to intersection lists, memory requirements depend on the number of lines cast in each box and also on the average number of intersections per list.

4.3. Analysis of the error

The previous results show the superiority of the new algorithm in front of classical Multipath. However, we have to remark that there exists a bias between the results obtained with classic Multipath and the ones obtained with the new approach. This bias, produced by the discretization in virtual patches and angular regions, does not bear any visual difference, so it is considered acceptable (to avoid the bias effects in the comparisons, we have computed the error with respect to the reference values generated using each algorithm).

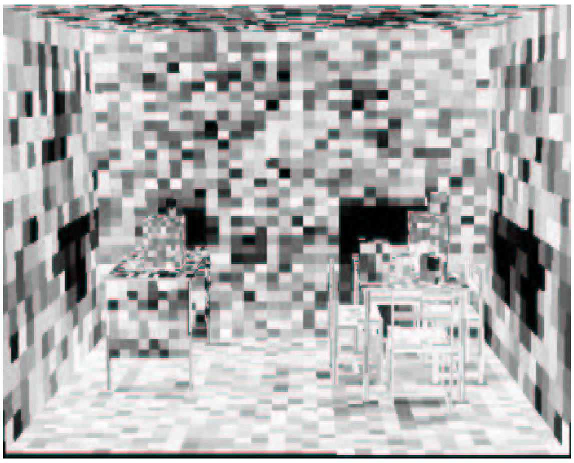


Figure 7: *Distribution of the relative error for scene ROOM. Darker zones present a higher relative error respect to the converged image.*

We have studied and represented in Fig. 7 the relative error distribution for scene *ROOM*. The darker the patches, the higher the error. Note that the maximum error occurs near the boxes (there are two boxes, one bounding the desk and cubes on the left and the other bounding the table, chairs and cubes on the right).

5. Conclusions and future work

We have presented a new approach to the global line Multipath method. Our new approach is based on the use of a hierarchy of box-bounded subscenes, allowing to execute the Multipath in the context of each subscene (and also of the whole scene). This is the main feature of the new algorithm. Lines used for each subscene are locally global, that is, they are global just in the context of the subscene. The different levels in which Multipath is executed are communicated by the angular regions in which the bounding boxes are subdivided. These angular regions accumulate the radiant power entering or leaving a subscene, allowing its later distribution.

For each angular region, transmittance characterizes the transparency or opaqueness of the region. This directional information is obtained from a pre-process in which global lines are cast for each subscene. We also use the global lines cast in the pre-process to compute the most probably intersected polygon (or polygons) for each angular region. Transmittances permit to ignore the interior of the subscenes when dealing with a higher level scene, resulting in a reduction of cost for the global lines, whereas the use of the associated polygons for each angular region results in a reduction of the cost for the local lines.

Global lines (in fact their intersection lists) are saved during the pre-process, allowing their later reuse (they are reused when running Multipath algorithm in each iteration for each subscene).

These features of the new algorithm produce a speed up factor between 1.9 and 3.7 respect to classic Multipath in the scenes we have presented. According the presented and other tests we see the tendency: the higher the complexity, the higher the speed up factor. On the other hand, using a deeper hierarchy level also seems to increase the speed up factor.

The method presented here is not fully hierarchical in the sense of [5], since it would involve a pre-subdivision for both the bounding boxes and the polygons in the scene. A fully hierarchical approach, allowing to adapt the mesh to the number of lines (instead of adapting the number of lines to the mesh) requires further investigation.

We expect that for fractal-like scenes, like vegetal scenes (trees or plants in a forest, etc.), the speed up factor could grow dramatically using our algorithm, but this promising field was not investigated in this paper.

The parallelization of the algorithm is also one promising line of continuation of this work. Note that the independency of Multipath execution in each subscene allows the distribution of these executions between different processors.

Acknowledgments

This project has been funded in part with a Spanish-Austrian Joint Action n. HU2000-0011, grant number TIC 2001-2416-C03-01 from the Spanish Government and grant number 2001/SGR/00296 from the Catalan Government. Special thanks to ICREA (Catalan Institute of Research and Advanced Studies) from third author.

References

1. B. Arnaldi.,X. Pueyo and J. Vilaplana On the division of environments by virtual walls for radiosity computation. *Photorealistic Rendering in Computer Graphics.*, **1**994.
2. M. Cohen and J. Wallace. Radiosity and Realistic Image Synthesis. *Academic Press Professional, Cambridge.*, **1**993.
3. F.X. Sillion and C. Puech. Radiosity and Global Illumination. *Morgan Kaufmann Publishers, San Francisco.*, **1**994.
4. L. Szirmay-Kalos, B. Csebfalvi, and W. Purgathofer. Importance Driven Quasi-Random Walk Solution to the Rendering Equation. *Proceedings of WSCG'98, 378-385.*, **1**998.
5. P. Bekaert. Hierarchical and Stochastic Algorithms for Radiosity. *Ph.D. thesis, Katholic Univ. of Leuven.*, **1**999.
6. M. Sbert. The Use of Global Random Directions to Compute Radiosity. Global Monte Carlo Methods. *Ph.D. thesis, Universitat Politècnica de Catalunya.*, **1**997.
7. M. Sbert, F. Pérez, and X. Pueyo. Global Monte Carlo. A Progressive Solution. *Rendering Techniques '95. Springer Wien New York, 231-239.*, **1**995.
8. M. Sbert, X. Pueyo, L. Neumann, and W. Purgathofer. Global Multi-Path Monte Carlo Algorithms for Radiosity. *The Visual Computer, 47-61.*, **1**996.
9. L. Neumann. Monte Carlo Radiosity. *Computing, 55, 23-42.*, **1**995.
10. L. Szirmay-Kalos, T. Foris, L. Neumann, and B. Csebfalvi. An Analysis of Quasi-Monte Carlo Integration Applied to the Transillumination Radiosity Method. *Proceedings of Eurographics'97.*, **1**997.
11. L. Santaló. Integral Geometry and Geometric Probability. *Ed. Addison-Wesley, New York.*, **1**976.
12. R.Y. Rubinstein. Simulation and the Monte Carlo Method. *John Wiley and Sons, New York.*, **1**981.
13. F. Castro, R. Martinez, and M. Sbert. Quasi Monte Carlo and Extended First Shot Improvements to the Multipath Method. *Proceedings SCCG'98, (Budmerice, Slovak Republic).*, **1**998.
14. L. Szirmay-Kalos, M. Sbert, R. Martinez, and R. Tobler. Incoming First-shot for Non-diffuse Global Illumination. *Proceedings SCCG'00,(Budmerice, Slovak Republic).*, **2**000.
15. F. Castro, and X. Pueyo. Hierarchical Improvement to the Global Monte Carlo Method for Form-Factors Computation. *Proceedings of Computer Graphics Spanish Conference, 1996.*, **1**996.
16. M. Sbert. An Integral Geometry Based Method for Fast Form-Factor Computation. *Computer Graphics Forum (proc. Eurographics'93).*, **1**993.
17. J. Wallace, K. Elmquist, and E. Haines. A Ray Tracing Algorithm for Progressive Radiosity. *SIGGRAPH 89 Conference Proceedings, Annual Conference Series.*, **1**989.
18. T.A. Funkhouser. Coarse-grained Parallelism for Hierarchical Radiosity using Group Iterative Methods. *SIGGRAPH 96 Conference Proceedings, Annual Conference Series.*, **1**996.
19. C. Soler, and F.X. Sillion. Accurate Error Bounds for Multi-Resolution Visibility. *Proceedings of Workshop on Rendering'96,* **1**996.
20. C. Soler. Hierarchical Instantiation for Radiosity. *Proceedings of Workshop on Rendering'00,* **2**000.

21. J.Hasenfratz, C.Damez, F.X.Sillion, and G.Drettakis. A Practical Analysis of Clustering Strategies for Hierarchical Radiosity. *Proceedings of Eurographics'99.*, **1**999.

22. M.Ouhyoung, and R.H.Liang. Reusable Radiosity Object. *Computer Graphics Forum, vol.15, n.3 pp. 347-356*, **1**996.

23. J.Goldsmith, and J.Salmon. Automatic Creation of Object Hierarchies for Ray Tracing. *IEEE Computer Graphics and Applications, 7-5, 14-20.*, **1**987.

24. A.C. Telea. Design and Implementation of a Radiosity Renderer. *Ph.D. thesis, University of Eindhoven*, **1**997.

25. G.Greger, P.Shirley, P.M.Hubbard, and D.P.Greenberg. The Irradiance Volume. *IEEE Computer Graphics and Applications, vol. 18-2*, **1**998.

26. M.Stamminger, A.Scheel, X.Granier, F.Perez, G.Drettakis, and F.Sillion. Efficient Glossy Global Illumination with Interactive Viewing. *Computer Graphics Forum, vol. 19-1*, **2**000.

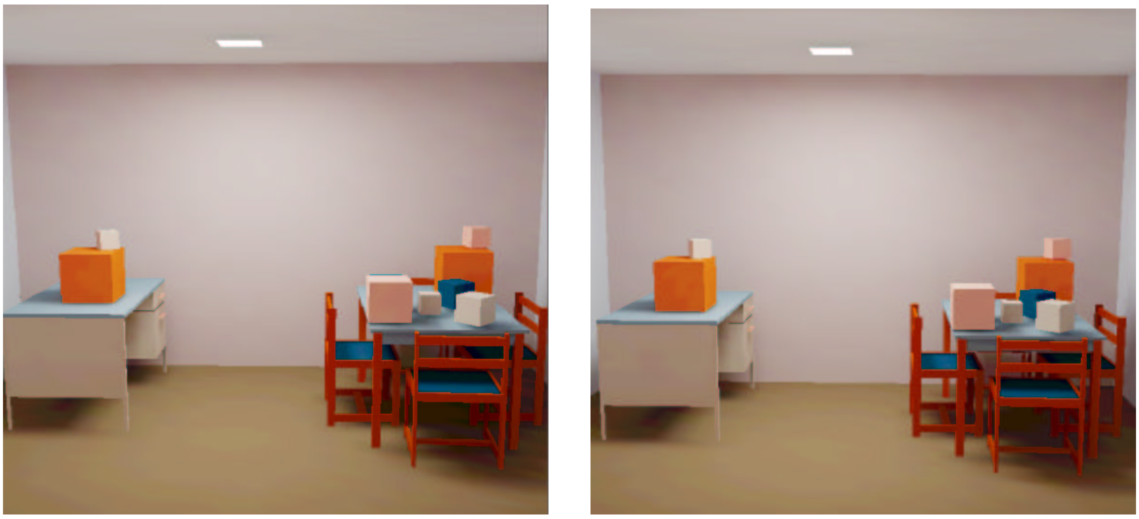


Figure 8: *Scene ROOM (left) Classic Multipath. t=489 sec. MSE =0.0095. (right) New algorithm. t=265 sec. MSE= 0.0095. Speed up factor=* 1.9. *(see more details in Table 1)*

Figure 9: *Scene OFFICE (left) Classic Multipath. t=1306 sec. MSE =0.467. (right) New algorithm. t=1310 sec. MSE= 0.125. Speed up factor=* 3.7. *(see more details in Table 3)*

Figure 10: *Scene BIG ROOM (top) Classic Multipath. t=453 sec. MSE =0.373. (bottom) New algorithm. t=471 sec. MSE= 0.115. Speed up factor=* 3.1. *(see more details in Table 2)*