



**HAL**  
open science

# Feature-based Control of Visibility Error : A Multi-resolution Clustering Algorithm for Global Illumination

François X. Sillion, George Drettakis

► **To cite this version:**

François X. Sillion, George Drettakis. Feature-based Control of Visibility Error : A Multi-resolution Clustering Algorithm for Global Illumination. [Research Report] RR-2503, INRIA. 1995. inria-00074174

**HAL Id: inria-00074174**

**<https://inria.hal.science/inria-00074174>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

***Feature-based Control of Visibility Error:  
A Multi-resolution Clustering Algorithm  
for Global Illumination***

François Sillion , George Drettakis

CNRS

ERCIM

iMAGIS

(Projet commun CNRS/INRIA/INPG/UJF)  
BP 53, F-38041 Grenoble Cedex 9

**N 2503**

Mars 1995

PROGRAMME 4



***rapport  
de recherche***



## Feature-based Control of Visibility Error: A Multi-resolution Clustering Algorithm for Global Illumination

François Sillion\* , George Drettakis\*\*

Programme 4 — Robotique, image et vision  
Projet iMAGIS

Rapport de recherche n° 2503 — Mars 1995 — 18 pages

**Abstract:** In this paper we introduce a new approach to controlling error in hierarchical clustering algorithms for radiosity. The new method provides control mechanisms which attempt to ensure that just enough work is done to meet the user's quality criteria. To this end the importance of traditionally ignored visibility error is identified, and the concept of *features* is introduced as a way to evaluate the quality of an image. A methodology to evaluate error based on features is presented, which leads to the development of a *multi-resolution visibility* algorithm. An algorithm to construct a suitable hierarchy for clustering and multi-resolution visibility is also proposed. Results of the implementation show that the multi-resolution approach has the potential of providing significant computational savings depending on the choice of feature size the user is interested in. They also illustrate the relevance of the feature-based error analysis. The proposed algorithms are well suited to the development of interactive lighting simulation systems since they allow more user control. Two additional mechanisms to control the quality of a simulation are presented: The evaluation of internal visibility in a cluster produces more accurate solutions for a given error bound; A progressive multi-gridding approach is introduced for hierarchical radiosity, allowing continuous refinement of a solution in an interactive session.

**Key-words:** Visibility error, Clustering, Feature-based error metric, Multi-resolution visibility, Hierarchical radiosity, Progressive multi-gridding, Global Illumination.

(Résumé : *tsvp*)

\*CNRS. iMAGIS est un projet commun CNRS/INRIA/INPG/UJF. BP 53, F-38041 Grenoble cedex 9.

\*\*Travail effectué au sein du projet iMAGIS lors d'un séjour post-doctoral ERCIM. Adresse actuelle: Department of Software, Technical University of Catalunya, Diagonal 647, 08028 Barcelona, Spain.

# Contrôle de l'Erreur de Visibilité à partir des Caractéristiques de l'Éclairage : A Multi-resolution Clustering Algorithm for Global Illumination

**Résumé :** Cet article propose une nouvelle approche pour le contrôle de l'erreur dans les algorithmes de "clustering" hiérarchiques pour la radiosité. La nouvelle méthode comprend des mécanismes de contrôle permettant de limiter l'effort à ce qui est nécessaire pour vérifier les critères de qualité de l'utilisateur. Pour cela on relève l'importance, habituellement ignorée, de l'erreur sur la visibilité, et la notion de propriété visuelle est proposée pour évaluer la qualité d'une image. Une méthode pour évaluer l'erreur à partir des propriétés est proposée, qui débouche sur un algorithme de *visibilité multirésolution*. On décrit aussi un algorithme pour une construction de hiérarchie adaptée. Les résultats de l'implantation montrent que l'approche multi-résolution permet de grandes économies de calcul en fonction du choix de la taille des propriétés qui intéressent l'utilisateur. Ils démontrent également la pertinence de l'analyse d'erreur fondée sur ces propriétés. Les algorithmes proposés sont bien adaptés pour le développement de systèmes de simulation interactive de l'éclairage puisqu'ils permettent un meilleur contrôle de l'utilisateur. Deux mécanismes de contrôle supplémentaires sont présentés : l'évaluation de la visibilité à l'intérieur d'un cluster produit des solutions plus précises pour une borne d'erreur donnée ; une approche de *multi-gridding* progressif est proposée pour la radiosité hiérarchique, permettant un raffinement continu de la solution dans une session interactive.

**Mots-clé :** Erreur de visibilité, Normes d'erreur, Visibilité multi-résolution

# Feature-based Control of Visibility Error: A Multi-resolution Clustering Algorithm for Global Illumination

## 1 Introduction

Modern global illumination algorithms allow the precise simulation of interreflection effects, penumbræ caused by extended light sources, and subtle shading variations caused by complex reflectance properties [2, 13]. Lighting simulation systems operate under very tight and often contradictory constraints: users typically require guaranteed and easily controllable precision levels, with maximum speed for interactive design. An important goal of rendering research is thus to enable the user to reduce the solution error where such reduction is deemed desirable, while at the same time limiting the time spent to achieve this reduction.

Unfortunately, the algorithmic complexity of radiosity methods (quadratic in the number of objects) in effect impairs their use for scenes containing more than a few thousands objects, while Monte-Carlo methods are unable to provide low and medium-quality solutions without too much noise. Therefore means must be found to focus the effort on the most important parts of the calculation.

This paper presents new algorithms and criteria that together allow very fine and efficient user control of the perceived quality of a solution. This is accomplished by first acknowledging the importance of the traditionally ignored *visibility error*, and using the concept of *features* to evaluate the quality of an image. This leads to the introduction of *multi-resolution visibility*, which allows precise control of the quality vs. time tradeoff. Additional mechanisms are then discussed to control the quality of a simulation in a working system.

### Previous work: error-driven computation and visibility

The introduction of the hierarchical radiosity algorithm [5] was a major step towards the design of practical lighting simulation systems. First, it effectively reduces the overall resource requirements for a given solution. Second, it uses a surface subdivision criterion as an explicit control mechanism. The subdivision criterion embodies the priorities used to guide the simulation, as it directs the computational effort to “areas of interest”, introducing a natural tool for error estimation.

Hierarchical radiosity (HR) remains quadratic in the number of input objects (since each pair of objects must be linked before hierarchical subdivision begins), and therefore is not suited to large collections of small objects. *Clustering*, the operation of grouping objects together into composite objects that can interact, provides a means to eliminate the quadratic complexity term. Such clustering can be performed manually [10, 6] or automatically [14, 12].

Historically, subdivision criteria for HR first consisted of simple bounds on either the form factor or the exchange of radiosity between two surface patches[5], under the assumption that the error incurred is proportional to the magnitude of the transfer. Using the concept of importance these bounds can be made dependent on the user’s interest for each region [15]. However such bounds tend to be quite conservative and thus produce unnecessary subdivision.

Recent work has attempted to characterize possible sources of error in global illumination [1], and establish error bounds on radiosity solutions [7]. These error bounds can then be used in the subdivision criterion of a hierarchical algorithm. Since the estimation of the error is decoupled from that of the actual transfer, subdivision can be avoided in regions where significant transfers take place without much error, resulting in better focus of the computational expense.

Existing error controls however typically ignore visibility as a possible source of error, or simply increase the error estimate by a constant factor in situations of partial visibility. In practice trivial bounds of 0 (total occlusion) and 1 (total visibility) are often used. While these bounds are always valid, their use results in unnecessary work being done to narrow down other error bounds by increasing the subdivision. Global visibility algorithms can be used to exploit the structure of architectural scenes and produce guaranteed visibility information [16], but they are not suited to large collections of independent objects. For exchanges between surfaces, *discontinuity meshing* also provides explicit visibility information, and indeed considerably improves the efficiency of HR [9]. However for Monte-Carlo or clustering approaches it is either impossible or impractical to calculate analytic visibility and error bounds must be used. For exchanges between clusters, an approximate visibility estimate

can be derived using equivalent volume extinction properties [12], but the error introduced in the process has not yet been analyzed.

Visibility error is admittedly difficult to evaluate, particularly since the computation of visibility itself is a costly process. Still, controlling this source of error is imperative since the quality of shadows plays a significant role in determining the user’s perception of image quality. In complex environments where clustering is most useful, a dominating part of computation time is spent in visibility calculations involving small, geometrically complex objects. Resulting visibility variations produce fine detail shadows, which may be of little interest to the user, or may be lost in the implicit averaging over a surface patch.

## Paper overview

The preceding discussion has shown that a key issue in designing efficient lighting simulation systems is to provide adequate control mechanisms to ensure that just enough work is done to meet the user’s quality criteria. It appears that control of visibility error has not yet been attempted, despite its great potential for tightening global bounds and reducing computation costs.

The goal of this paper is twofold. First, a new approach to visibility error estimation is proposed, based on the central concept of *features* to evaluate the quality of an image. This leads to the introduction of *multi-resolution visibility*, a simple methodology that can be used whenever a hierarchical spatial structure is used. In this paper its application to clustering algorithms is discussed, but it is equally promising for Monte Carlo techniques. Second, quality control mechanisms are discussed in view of the development of interactive simulation systems. In particular the use of a high-quality and expensive local pass is dismissed in favor of a direct computation driven by user’s requests.

We begin with the introduction of a feature-based approach to error analysis. In Section 2 we show that existing error metrics are not capable of determining when a given level of detail is satisfactorily represented. A simple metric is then proposed to illustrate how to take into account the user’s interest in a minimal *feature size*.

In Section 3 we explain how to compute multi-resolution visibility information using a spatial hierarchy augmented with equivalent *extinction* properties. Selection of a hierarchical level for visibility computation can then be based on the resulting feature size on the receiver. Multi-resolution visibility can be used in clustering or Monte-Carlo algorithms.

Construction of a hierarchy well suited to clustering and multi-resolution visibility is discussed next in Section 4. Results presented in Section 5 show that the multi-resolution visibility algorithm successfully generates images in which selected features sizes are accurately represented. This calculation results in significant computational savings when the detailed representation of small features is not desired.

Section 6 presents more quality controls for clustering algorithms, including intra-cluster visibility determination in linear time and progressive multi-gridding. These controls are especially important for the envisioned use in interactive systems.

## 2 Feature-Based Error Analysis

To a large extent the quality of an image is judged based on how well features of different sizes are represented. In this section we show how traditional error metrics are incapable of performing such a characterization. We then investigate means to identify illumination features based on their object-space size, in order to develop a formulation of the solution error adapted to visual constraints.

It is by no means obvious to characterize what constitutes an illumination feature. For the purposes of this paper, we will consider image features to be the connected regions of varying illumination corresponding to shadows (regions in umbra or penumbra).

### 2.1 Usual error metrics are inadequate for “feature detection”

A major difficulty present in all lighting simulation algorithms is that in general the exact solution is not known at the time of computation. Thus the estimation of the error in a proposed approximation is particularly difficult, and must rely on the computation of bounds for all algorithmic operations.

Even in the case where an exact solution is available, it is not necessarily a simple task to define the quality of a given approximation. This is done by choosing a particular error metric to quantify the distance between an approximate solution and the true solution. A “good” metric should therefore convey a sense of the user’s

requirements. A central observation in this paper is that when simulating a complex scene, the user is typically interested in capturing illumination variations down to a certain scale. Very small details are not as important, or at least not in all areas of the scene. We strive to define a control mechanism that will avoid any work that would only generate such small details.

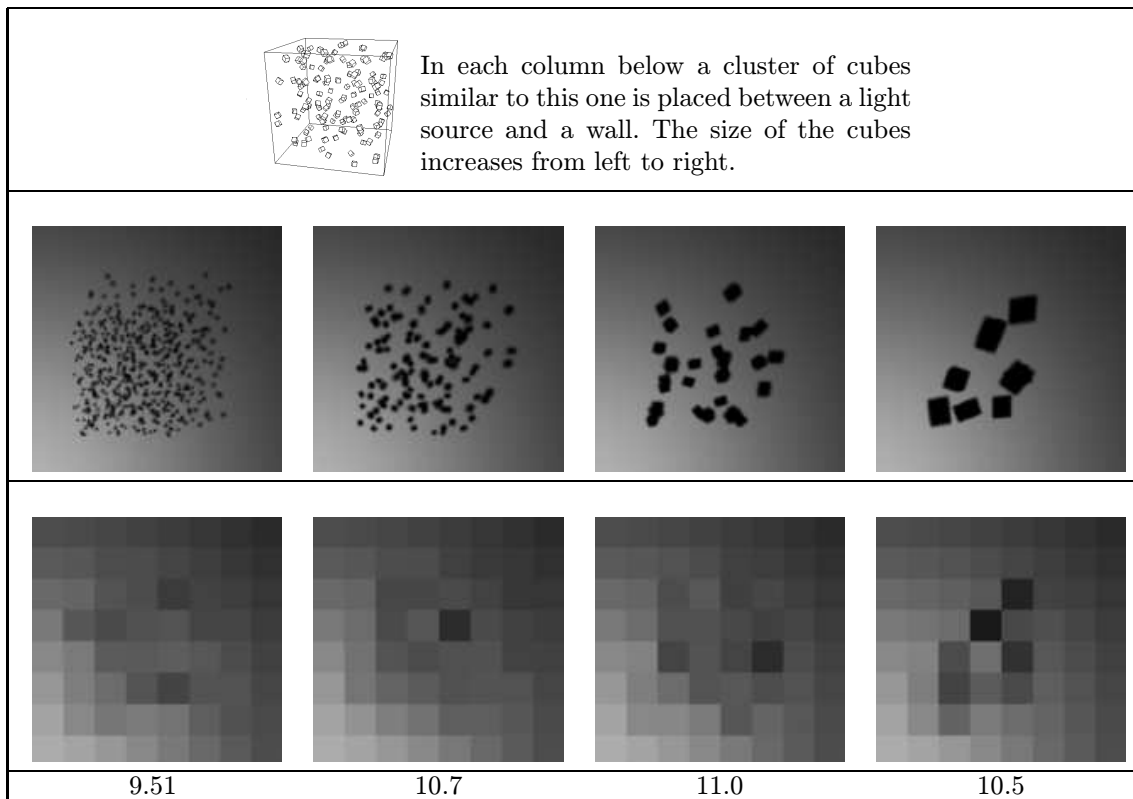


Figure 1: Comparison of approximate illumination solutions using four different clusters. Top row: reference images (illumination of the wall). Middle row: approximate images using a coarse mesh. Bottom row:  $L^2$  error norms. Note that the four approximate images have similar  $L^2$  error values, and all hide some illumination information. However the varying size of the missing illumination features cannot be discovered.

Figure 1 illustrates the issue by showing shadows cast on a wall by four different groups of objects, each group being composed of several cubes of a different uniform size. Four approximate images, all computed using the same mesh size, are shown below the “exact” images. Consider a user who is interested in shadows of a specific size, e.g. those of the image on the extreme right, but is satisfied by the averaging of the smaller, detailed shadows on the left<sup>1</sup>. The user thus does not wish more work to be done for the detail shadows, but wishes to have a more accurate representation at the larger scale. The subdivision criterion used in a HR algorithm for instance should be capable of halting the subdivision for the left-hand group, while ordering further computation for the group on the right. Thus it would be desirable for an error measure to distinguish between the four cases.

Traditional error metrics are incapable of making such a distinction. As an example consider the commonly used family of error metrics expressing the distance between a reference function  $f$  and an approximate function  $\hat{f}$  as the  $L^p$  norm

$$\|\hat{f} - f\|_p = \left( \int |\hat{f}(x) - f(x)|^p dx \right)^{\frac{1}{p}}$$

$L^p$  norms simply add error contributions from all points on a surface (or in an image), and do not take into account higher-level properties of the radiance distributions, such as the size and shape of illumination features. This is illustrated by the similar values obtained for the four groups in Figure 1. Appendix A shows that in fact

<sup>1</sup>Perhaps a more realistic example would be a situation where a user is viewing an office scene from the doorway, and in which accurate shadows for chairs and desks are important, but averaged, low quality shadows from details such as pens on a desk are satisfactory.



for a point light source the  $L^1$  or  $L^2$  error introduced by averaging all visibility variations depends only on the average visibility, and not on the size or shape of the shadows.

Note that the problem here is not the lack of a reference solution. Such a solution is used but does not help to differentiate the four cases: the difficulty is to recognize that the hidden features have different sizes and that the user may not be equally interested in all these feature sizes.

## 2.2 A first proposal for an error metric based on feature size

Our hypothesis is that illumination features (shadows or bright areas) are important only as far as they have a significant visual impact. Therefore it is possible to define a *feature size* on a receiving surface, and decide that features smaller than that size are “unimportant”: their absence should not contribute to the error.

We begin by recognizing that there is an infinite choice of error metrics for images (for simplicity we consider the illumination on a surface as a two-dimensional image). Therefore the particular metric suggested below is merely one possibility, and not necessarily the best choice. Our aim for now is simply to show the relevance of the feature-based approach. To this end, we also assume for now that we have access to all the information in a reference solution. The multi-resolution visibility technique of Section 3 will show how the ideas developed here can still be used in the absence of the reference solution.

A simple way to define an error metric based on features is to segment the image  $f$  into two components by means of a *feature mask*  $\mathcal{F}^s(f, x)$ : a binary function that equals one at points  $x$  that belong to a “feature” (of size greater than  $s$ ) of function  $f$ . Computation of feature masks from the reference solution will be described in the next section. Simply put, since we want an accurate representation of our features, we compute an  $L^p$  norm of the difference between the approximate function and the reference function over the mask region. For points outside the feature mask, we are content with an average value (remember that illumination features present at those points are smaller than the minimum feature size  $s$ ). Thus in our current implementation we compute average values at each point, for both the approximate and reference functions, using a box filter of size  $s$  around the point of interest, and compute an  $L^p$  norm of the difference between the averages.

The feature-based error metric (FBEM) is thus summarized by the following formula, where  $\overline{f^s}$  represents the filtered version of  $f$ :

$$\|\widehat{f} - f\|_p^s = \left( \int |\widehat{f}(x) - f(x)|^p \mathcal{F}^s(f, x) dx + \int |\overline{\widehat{f}^s}(x) - \overline{f^s}(x)|^p [1 - \mathcal{F}^s(f, x)] dx \right)^{\frac{1}{p}}$$

## 2.3 Examples





Feature size:				
5	14.76	16.34	17.25	17.31
16	9.37	12.24	15.76	15.80
24	4.78	6.50	9.06	14.74
30	4.23	3.16	6.90	13.37
40	3.65	2.33	3.35	6.94

Table 1: Feature-based error metric (FBEM) for the four approximate images of Figure 1 and five different feature sizes. The four measures are equivalent for small feature sizes, and decrease at different rates as a function of  $s$ . The corresponding reference images are shown again for clarity.

Table 1 shows the FBEM values computed for the four groups of Figure 1 and different values of the minimum feature size  $s$ . For small  $s$  values, all FBEM values are high since the metric is equivalent to an  $L^2$  metric in the limit of  $s = 0$ . As  $s$  increases, FBEM values decrease more rapidly for the groups containing smaller objects, as expected. There appears to be a residual error of about 3 due to the mesh size used for the approximate solutions.

Assume the user is interested in clearly seeing features of size 30 or greater (which is approximately the size of the shadows on the extreme right), while being content with an average for all features smaller than this size. The extreme right-hand image of Figure 1 requires more work since the FBEM value for  $s = 30$  is high. The approximation for the other three images is deemed satisfactory since the error is low.

Thus, using the FBEM presented above, it is possible to reveal the presence of features greater than a given threshold in the approximate images, opening the way for selective subdivision based on the user's minimum feature size of interest. Of course this could not be used *as is* in a subdivision criterion for HR, since it uses a reference solution, but it is useful for *a posteriori* validation of control mechanisms. The intuitions obtained by this approach also lead to the development of quality control presented later.

### 2.4 Computation of feature masks

According to the definition of features given above, computing a feature mask amounts to identifying connected regions of "significant" size. Mathematical morphology (the study of shapes) provides tools to isolate features based on their size [11]. Consider a binary image, representing for example the characteristic function of an object. We define the action of an *Erosion* operator as follows: all points outside the object (white) are untouched. All points inside the object that have a neighbor outside become white. All other points remain black. An *Expansion* operator is defined similarly by including in the objects all outside points that have a neighbor in the object. Figure 2 shows a reference image and images obtained after a number of erosions (top) or expansions (bottom).

Clearly an object of diameter  $2d$  will disappear after  $d$  erosions are applied in sequence. Thus applying a sequence of  $n$  erosions followed by  $n$  expansions will successfully eliminate all small regions, but keep larger regions (slightly modifying their shape in the process). This process is illustrated in Figure 3.

The erosion and expansion operators thus provide a means to isolate illumination features of significant size: starting with a binary image representing all illumination features, all features smaller than  $p$  pixels in diameter are eliminated by applying  $p/2$  successive erosions, followed by  $p/2$  successive expansions. Computing the effect of the Erosion operator on a binary image is straightforward using bitwise operations: the result is the logical

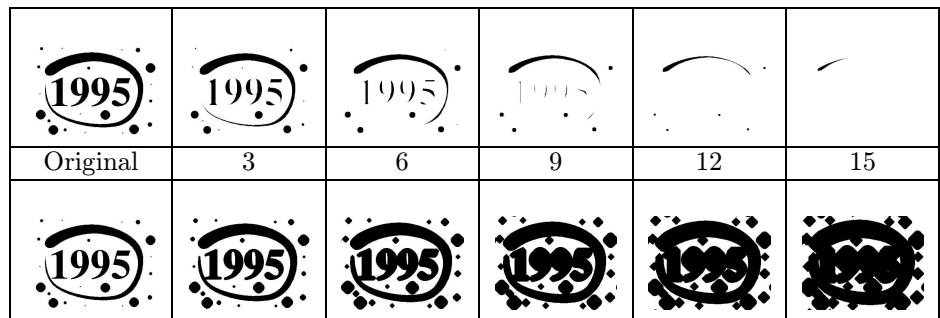


Figure 2: Effect of Erosion (top row) and Expansion (bottom row) operations on a binary image. The reference image appears in the left column, and the number of applications of the operators increases from left to right.

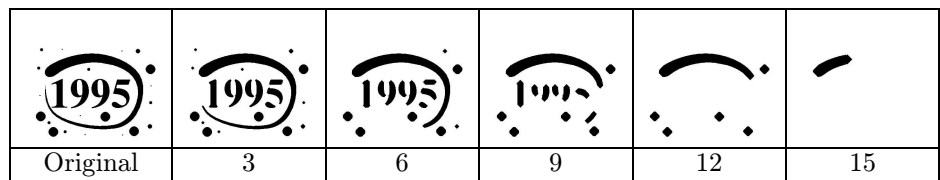


Figure 3: Effect of successive Erosion and Expansion sequences.  $n$  erosions are applied to the reference image (shown at left), followed by  $n$  expansions, for increasing values of  $n$ . Only regions of diameter larger than  $2n$  survive the operation.

OR of the image and the four translated copies of itself (by one pixel) in the  $+x$ ,  $-x$ ,  $+y$  and  $-y$  directions<sup>2</sup>. For the Expansion operator the logical operator AND is used.

In our implementation, the original binary image is computed by recording all areas of the receiver that have a partial or occluded view of the light source. Feature masks are computed by applying the proper number of successive erosions and expansions to eliminate unwanted features. Figure 4 shows some feature masks for the four groups used above.










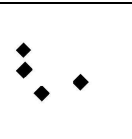
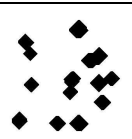
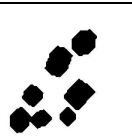



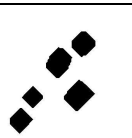
Original mask				
Feature size 12				
Feature size 18				
Feature size 24				

Figure 4: Some feature masks computed from the reference images in Figure 1.

The analysis presented in this section has shown how to evaluate image quality automatically by taking into account the feature sizes which are well represented by a simulation. In the following section we show how this fundamental concept can be used to develop a *multi-resolution* (MR) visibility algorithm. With the multi-resolution algorithm, expensive high quality visibility calculations are only performed when they result in accurate representation of the feature size chosen by the user.

### 3 A Multi-resolution Visibility Algorithm

In the previous section we presented the concept of a *feature size* and introduced an error metric which permits the evaluation of image quality determined by how well illumination features are represented. We now proceed to demonstrate how these concepts can be applied to construct an efficient multi-resolution visibility algorithm.

Hierarchical spatial subdivision structures are often used in the calculation of global illumination algorithms, in particular when form-factor estimation is performed with ray-tracing [17, 4, 5, ...]. In radiosity clustering algorithms the hierarchy of clusters is also used for radiometric calculations, by letting clusters represent their contents for some energy transfers [12, 14]. The following *multi-resolution visibility* algorithm naturally extends previous clustering approaches by allowing clusters to also represent their contents in some visibility calculations. If a specific feature size  $s$  has been chosen, it is unnecessary to consider the contents of a cluster for visibility if these contents will produce features which are smaller than  $s$ .

<sup>2</sup>This definition implicitly uses 4-connectivity to define neighbors. In our implementation we perform in-place modifications of the image which amounts to using 8-connectivity.

### 3.1 Approximate visibility computation between clusters using an extinction model

Let us assume that we have grouped all objects in the scene into a hierarchy of clusters. It has been shown that approximate visibility calculations can be performed using an analogy between clusters and absorbing volumes [12]. The approximation (asymptotically exact for homogeneous isotropic clusters when the size of the objects goes to zero) consists of associating an *extinction coefficient*  $\kappa$  with each cluster. The transmittance function between two points  $P$  and  $Q$  in the scene is then given by

$$\begin{aligned} T(P, Q) &= e^{-\int_{PQ} \kappa(u) du} \\ &= e^{-\sum_{i \in \mathcal{C}(PQ)} \kappa_i l_i} \end{aligned}$$

where  $\mathcal{C}(PQ)$  is the set of clusters traversed by the ray joining  $P$  and  $Q$ ,  $\kappa_i$  is the extinction coefficient of cluster  $i$ , and  $l_i$  is the length traveled inside cluster  $i$  by the ray.

Extinction coefficients are given by the following formula

$$\kappa_i = \frac{\sum_j A_j}{4V_i}$$

where the area of all surface patches contained in cluster  $i$  is summed and divided by the cluster's volume.

### 3.2 Multi-Resolution Visibility

In the rest of this section we consider the emitter-blocker-receiver configuration shown in Figure 5, which consists of two surfaces, the emitter  $E$  and the receiver  $R$ , in two-dimensions. This restriction is for presentation purposes only and is removed in the following section.

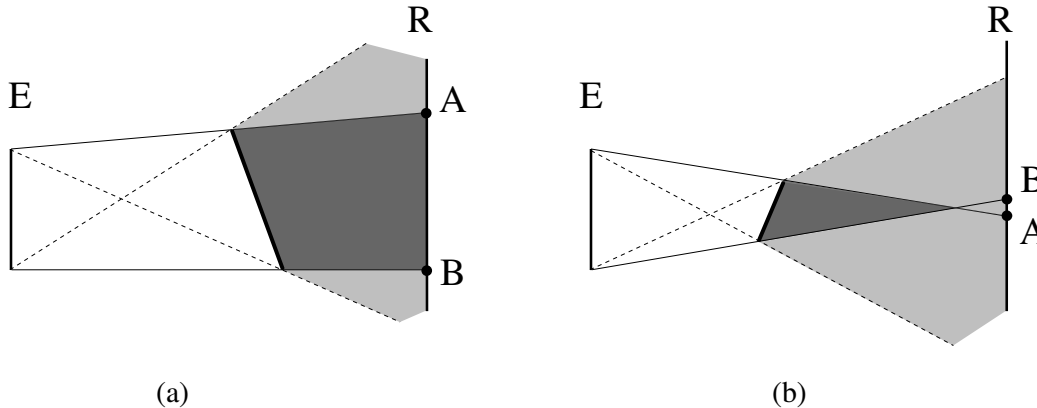


Figure 5: Definition of shadow features created by a blocker. (a) There is always an umbra region since the blocker is larger than the emitter. (b) The umbra region may disappear for some positions of the blocker.

If a blocker (which for now we also consider to be a surface) is placed between the emitter and the receiver, *umbra* and *penumbra* regions are created on the receiver. Depending on the position of the blocker, the umbral region may be inexistent (Figure 5.b). Given the definition discussed above the size of the umbral zone ( $AB$  in Figure 5), if it exists, is the *feature size*.

The blocker may actually be a hierarchical representation of a collection of objects (a *cluster*) as pictured in Figure 6.a. In this case, at each level of the hierarchy an extinction coefficient is stored allowing the approximate calculation of the attenuation of a ray if it passes through the cluster, as described previously.

*Multi-resolution visibility* can be performed by avoiding the descent into the hierarchy after a certain level. When the required conditions are met the extinction coefficient is used instead, thus avoiding the intersection

of the ray with all the objects in the cluster hierarchy. Evidently, the effect is that visibility is no longer exact, but an average estimation of transmittance. It is here that a large potential gain in computation time can be achieved. In scenes where the small detail objects (e.g., models of phones, keyboards, small objects on a desk etc.), comprise the largest part of the geometric complexity, the intersection with these objects can quickly become the overwhelming expense of visibility (and overall) computation. By considering the higher level clusters for visibility computation instead of the numerous contents, when such a choice is dictated by the chosen feature size, this expense can be completely avoided.

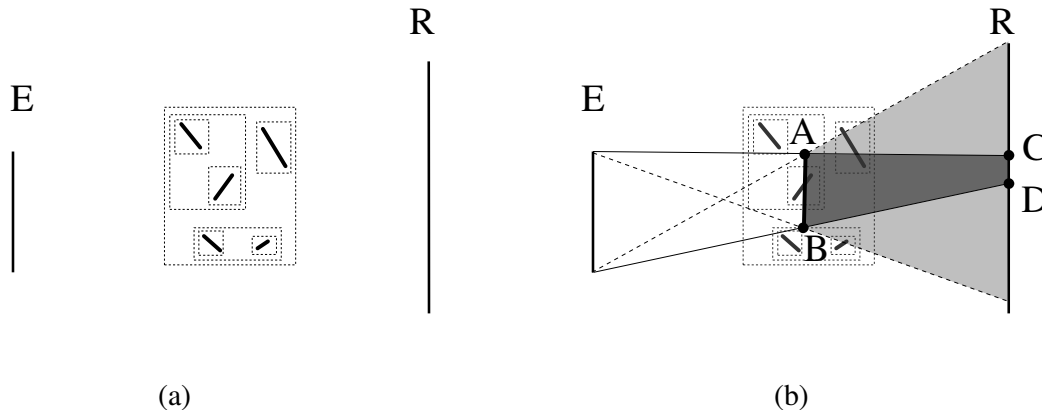


Figure 6: Visibility estimation through a cluster. (a) the blocker is a hierarchy of clusters. (b) an “equivalent blocker” is used to estimate the maximum feature size on the receiver.

Recall the discussion in Section 2 in which the user wishes to accurately represent all features of size greater than  $s$  on the receiver. To achieve this, all that is required is to descend sufficiently far into the hierarchy so that the large shadows are accurately calculated, while performing the approximate calculation for small, detail shadows.

To facilitate such a choice each cluster is augmented with a description of the maximum blocker size  $BSIZE$  of its contents (we give a precise definition of this in the following section). It then suffices to place a fictitious blocker of size  $BSIZE$ , at the center of the actual blocker ( $AB$  in Figure 6.b). The descent in the blocker hierarchy can be terminated if the projected umbral region of the fictitious blocker ( $CD$  in Figure 6) is smaller than the chosen feature size  $s$ .

In the same manner as for blockers, contiguous regions which let light traverse must also be considered as feature creators. This is due to the nature of features which can be considered “negative” in the sense of umbra in a bright region, or “positive” in the sense of lit areas inside a dark region. Thus we now extend our definition of features from Section 2 to include connected regions of illumination in a shadowed region. This is consistent with the symmetric expression of the visibility error (incurred when averaging illumination of a point light source) with respect to umbra and light, as presented in Appendix A. As a consequence we define  $BSIZE$  to be the maximum of the connected regions of light or shadow.

### 3.3 Characterization of a Cluster for MR Visibility

All that is required in order to apply the preceding algorithm is the determination of  $BSIZE$  for each cluster. The restriction to two-dimensions is now lifted, and the treatment for three-dimensional clusters is described. For now clusters are assumed to contain objects placed so that the cluster density can be considered isotropic, and thus does not depend on the direction of incidence of a ray.

The goal is to determine a representative size for a blocking cluster, which will allow the calculation of the maximum feature size given a specific emitter-receiver configuration. At first glance it may seem natural to take  $BSIZE$  to be the size of the largest object contained in the cluster. However there is one important consideration: it is the *connected region* of shadow on the receiver which we wish to consider. Furthermore, as discussed above, the regions of light potentially blocked by the contents of the cluster *and* the regions of light which pass through must be considered separately.

A preprocessing step is performed to calculate *B*SIZE for all clusters in the hierarchy. For each cluster its contents (i.e., all the contained objects) are orthographically projected into a binary image. This is where the assumption of isotropic clusters produces a simplification, since a single orthographic projection can be used. For non-isotropic clusters the *B*SIZE parameter is a function of the direction of interest<sup>3</sup>.

The Erosion and Expansion operators from Section 2.4 are then used to compute the maximum sizes for blockers and free regions inside a cluster. Erosions (respectively Expansions) are computed until all objects have disappeared (respectively until all free space has disappeared). The *number* of erosion or expansion operations defines the value of *B*SIZE for the blocked and free regions respectively. In our implementation we do the projections and erosions using Graphics hardware<sup>4</sup>.

## 4 A Hierarchical Structure for Multi-Resolution Visibility and Clustering

Previous automatic clustering approaches have used spatial data structures developed for ray-tracing. In this section we show that given the calculation of average visibility based on extinction coefficients in the manner of [12], it is beneficial to develop a special-purpose hierarchical data structure, such that the resulting clusters have certain required properties.

In [14], the hierarchical bounding boxes approach presented by Goldsmith and Salmon [3] was used, while in [12] a K-D tree was used. In this section we introduce an efficient algorithm to construct a hierarchy suitable for cluster-based hierarchical radiosity, which is much better adapted to the calculation of visibility using extinction coefficients and is computationally efficient.

### 4.1 Construction of the hierarchy

By definition, clusters are constructed to *represent* as accurately as possible the collection of objects they contain. By introducing computation of visibility using extinction coefficients and also multi-resolution visibility, apart from the representation of energy transfer of the contained objects as a whole, the clusters also need to correctly represent the transmission properties of the collection of contained objects.

These two modes of representation place different constraints on the cluster hierarchy. From the point of view of energy exchanges, good clusters allow tight bracketing of radiance or visibility functions (thus surfaces with similar orientation that do not shadow each other are preferred). From the point of view of visibility approximation, good clusters are ones for which the extinction property is plausible (thus homogeneous isotropic clusters are preferred). Given these constraints, we have identified two key properties for clusters. (a) *proximity* and (b) close *homogeneity* of the contained objects. Maintaining proximity seems to be a natural way to group objects, in particular when the cluster is used to represent radiative transfers. In addition, for the purposes of multi-resolution computation it is important that objects contained in a cluster are close so that the averaging performed does not introduce unacceptable artifacts. Homogeneity here means that we want a cluster to group objects of similar size, and plays a crucial role in the quality of the average visibility computation.

As a simple measure of proximity, we use the percentage of empty space resulting from a clustering operation (i.e., the addition of an object or a cluster to another cluster). Thus we prefer clusters in which the empty space is minimized.

The second desired property requires that the clustering algorithm groups objects which are of similar size. Such clustering algorithms can become expensive if performed naively, since before choosing an appropriate cluster for an object it would be necessary to test every cluster created so far.

To address this problem we perform the clustering step by using a hierarchy of  $n$  levels of uniform grids: We start with level 0, which is a single voxel the size of the bounding box of the scene and then at each level  $i$  we create a grid which is subdivided into  $2^i$  voxels along each axis. We then insert each object into the level for which its bounding box fits the voxel size.

Once these grids have been constructed, we start at the lowest level  $M$ , containing the smallest objects. We group the objects entirely contained in each voxel, by attempting to minimize the empty space, in accordance to the proximity criterion described above. In addition, objects which are very small compared to the grid size are grouped into an appropriate cluster, even if the resulting cluster is largely empty. Once all the voxels of a

<sup>3</sup>A simple brute-force approach for general clusters could therefore sample a fixed number of directions and interpolate in-between. We are currently working on more efficient representations of directional information.

<sup>4</sup>This is shown in the accompanying video, where the computation of cluster properties using erosion is shown in slow-motion for the purposes of demonstration.

level have been treated, we attempt to add the objects not entirely contained in a single voxel at this level to the clusters already constructed, again using the same criteria. We then insert the clusters created to the grid of the level immediately above, and iterate.

This construction is somewhat similar in spirit to that of Goldsmith and Salmon with the major difference that the size of objects enters the clustering criterion through the use of grids.

Once the cluster hierarchy has been created, the data structure is augmented with average transmission behavior by propagating the average extinction values up the hierarchical structure as in [12]. When multi-resolution visibility is used, the *B*SIZE estimation is also performed for each cluster in the hierarchy in the manner described in Section 3.

## 4.2 Comparison of Automatic Clustering Algorithms



Figure 7: Timings (in seconds) for surface visibility calculation. The relatively high error tolerance reveals different problems with each method. Recall that exchanges are computed between groups of surfaces at once, sometimes producing odd illumination patterns such as very dark patches. These artifacts disappear with sufficiently low error bounds. Throughout the paper all timing information was obtained on an Indigo R4000 computer.

Figures 7 and 8 present a qualitative comparison of the hierarchy used in [12] and the newly proposed hierarchy construction presented above. In [14] it is mentioned that a *modified* Goldsmith and Salmon algorithm is used. Since the precise modification is not described we do not report here the details of the comparison we have performed with the original Goldsmith-Salmon construction. We first test each data structure using the surface visibility algorithm similar to that of [14], and then the average visibility proposed in [12]. The scene consists of 5380 polygons.

From the timing figures we see that the new hierarchy is faster than the K-D tree hierarchy for surface visibility. Overall the quality of the resulting shadows is comparable for both cases.

However, differences become much more pronounced when we compare the two hierarchies using the average visibility calculation.

It is interesting to observe the significant time gain achievable by the average visibility algorithm, given a suitable hierarchy (for the new method we observe a factor of 4). More importantly, it is clear that the K-D tree is ill-suited to the generation of shadows using extinction coefficients (the shadows are completely missing

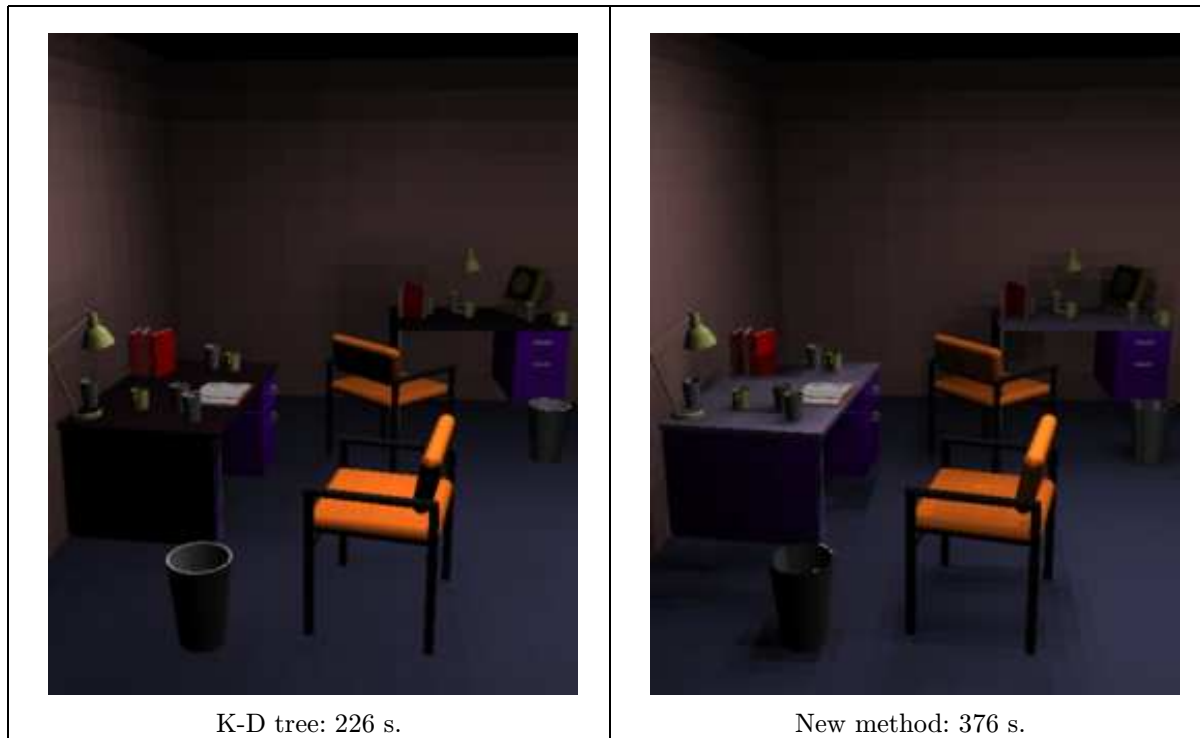


Figure 8: Timings (in seconds) for average visibility calculation using extinction coefficients. Note that the shadows are completely missing from the K-D tree image, showing that the clusters are not suited to the extinction model.

from under the desks). The new algorithm, although slower than the K-D tree, does provide a representation of all shadows (note for instance the shadows on the desk top).

From the above discussion it is clear that for the purposes of the approximate visibility algorithm using extinction, the construction of a suitable hierarchy is imperative. The algorithm presented here is a first approximation based on homogeneity and proximity, but nonetheless presents visible benefits compared to previous approaches. In particular it allows efficient ray casting in the scene, while constructing clusters reasonably well suited to the approximate visibility calculation.

## 5 Results of multi-resolution visibility

We have implemented the hierarchy construction, the calculation of *B*SIZE and the multi-resolution visibility algorithm in a hierarchical radiosity clustering testbed.

To evaluate the results of the multi-resolution visibility approach we have computed images of a test environments using different values for the feature sizes of interest  $s$  on a receiver. In this way, the transitions of visibility levels will become apparent. A low error threshold was used to ensure that the mesh size does not interfere with the effect of multi-resolution visibility<sup>5</sup>

The test scene is shown in Figure 9 on the left. It contains the four clusters used in Section 2 and a small light source (in yellow). The right-hand image is the illumination obtained on the back wall and serves as a reference image. For all these images visibility was always computed solely using extinction properties.

In addition to the reference image (without multi-resolution visibility), we show four other images, where the desired feature size parameter (described in Section 2.2) is set respectively to 2 2.8 3.5 and 5. For each image

<sup>5</sup>Thus the relatively high computation times. The demonstration in the accompanying videotape shows that the same scene can be treated much more quickly: with a coarser mesh images are obtained in a matter of seconds.



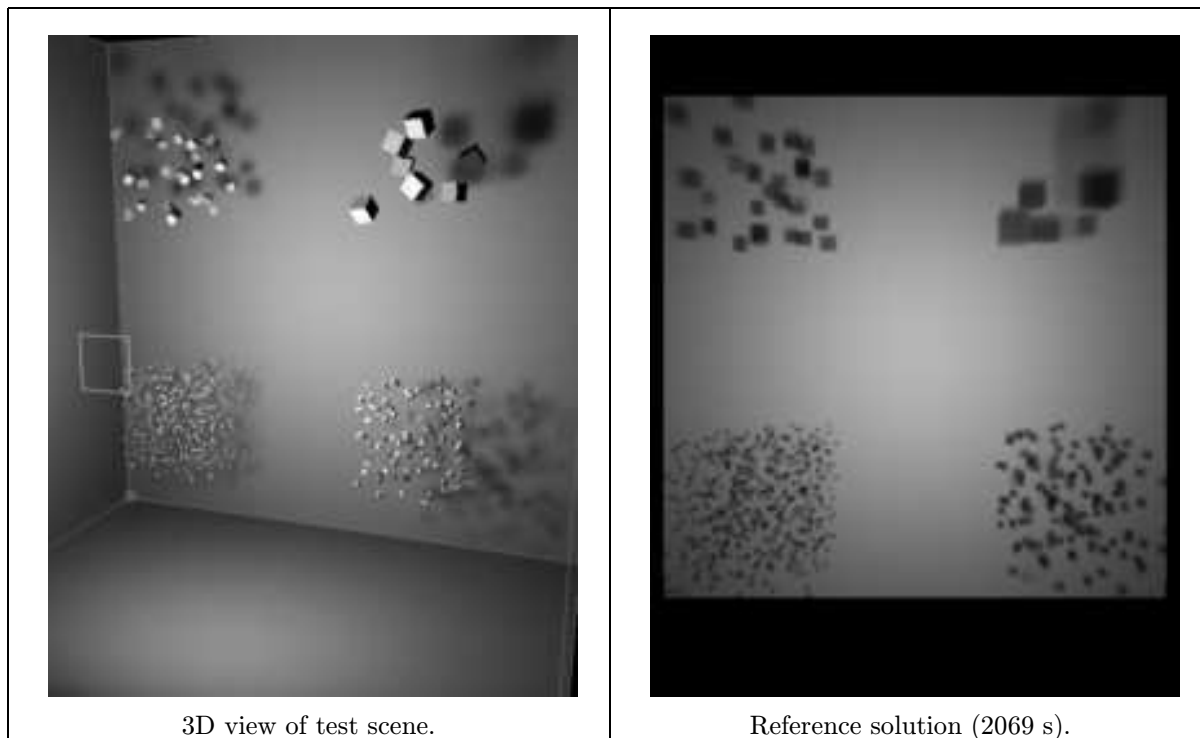


Figure 9: Reference image used in the error comparisons.

the computation time in seconds is given. As can be clearly seen, as the desired feature size becomes larger, the computation time drops significantly<sup>6</sup>.

We next show that multi-resolution visibility is consistent with the feature-based error metric (FBEM) presented in Section 2.2, by computing the FBEM for the images described above. Although the four clusters have been grouped in a single image for simplicity, we apply the error metric only on the region of the image corresponding to each cluster, to obtain an FBEM value for each of the four groups.

For the four images, we show for each cluster the value of  $L^2$  error norm (back row) and the value of the FBEM for a feature size  $s$  equal to that used in the MR Visibility algorithm.

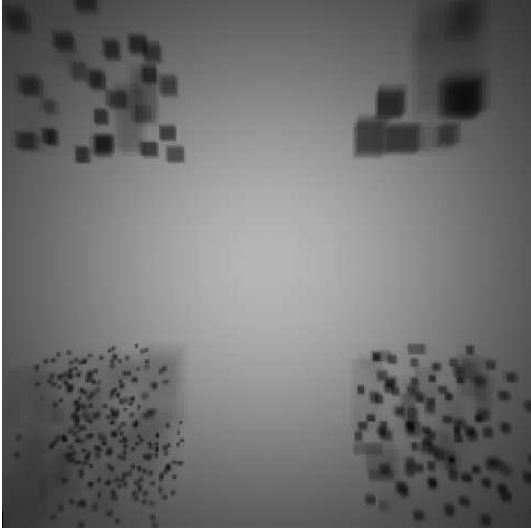
We note that as we increase  $s$  the  $L^2$  norm for all clusters increases, as more and more averaging is being performed. Still the increase appears later for larger objects, as expected. The FBEM values are always of similar magnitude, despite the fact that very different levels of averaging are being used in different clusters in a given image. This shows that the multi-resolution visibility algorithm accomplishes its purpose: given a desired feature size, it ensures that the corresponding FBEM remains low while allowing time gains.

## 6 Control of image quality in clustering algorithms

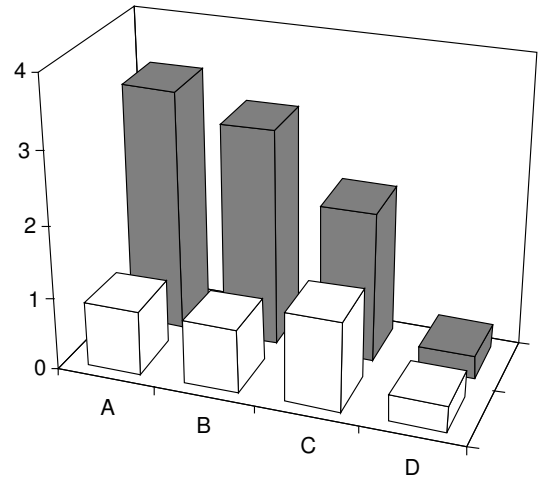
Several authors have recently proposed to separate the computation of high-quality images into two phases: a coarse quality global illumination calculation is first performed using elaborate algorithms such as discontinuity meshing or clustering in a *global pass*. A view-dependent, potentially very expensive *local pass* follows [8, 14]. This local pass is typically a ray-casting operation: at each pixel the energy from all the links is collected. It is thus possible to achieve high quality shadows such as those shown in [14]. The cost of this local pass is however typically many times larger than that of the light-transfer calculation using clusters. In essence this pass eradicates all computation time benefit achieved by using the clusters in the first place, and excludes any possibility for interactivity with quality and error control.

Our approach is different: we attempt to control the amount of computation time spent by using explicit quality controls, most notably in the calculation of visibility. We thus maintain the “progressive refinement”

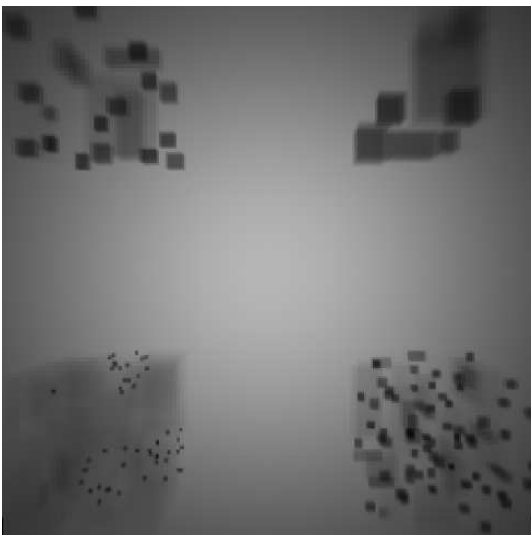
<sup>6</sup>The computation of the multi-resolution visibility criterion is currently not optimized and could probably be made cheaper, resulting in even greater speedups



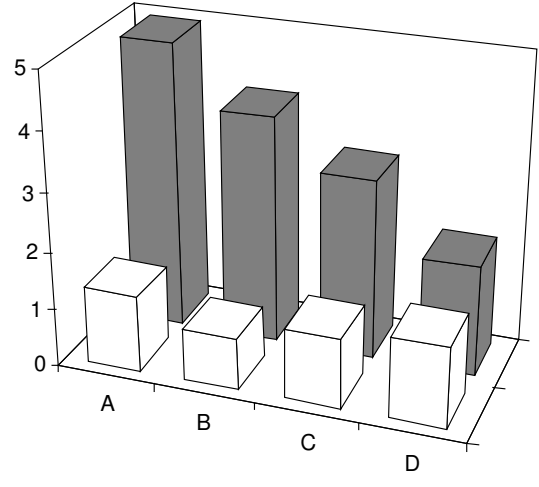
Feature size: 2.0 (1984 s).



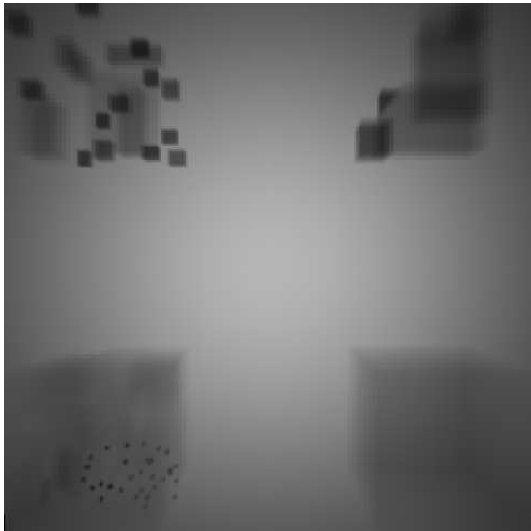
$L^2$  error and FBEM(2.0) for each cluster.



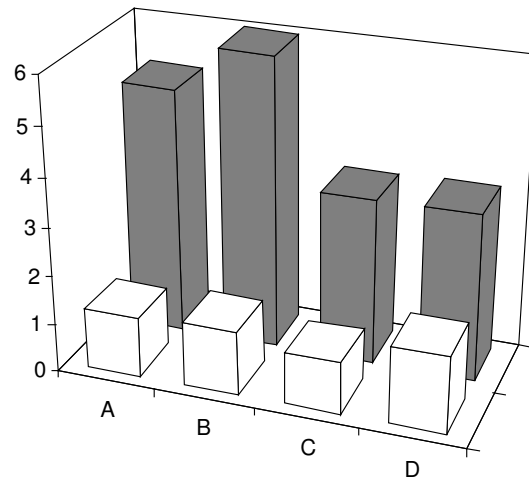
Feature size: 2.8 (1648 s).



$L^2$  error and FBEM(2.8) for each cluster.



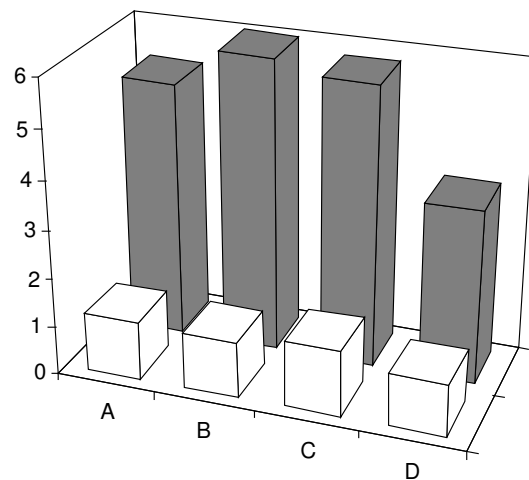
Feature size: 3.5 (1459 s).



$L^2$  error and FBEM(3.5) for each cluster.



Feature size: 5.0 (1356 s).



$L^2$  error and FBEM(5.0) for each cluster.

philosophy to illumination calculation for rendering, which we believe is the path of choice to develop interactive systems for very complex environments. The main thrust of the new approach is based on the feature-based techniques and multi-resolution visibility presented in the previous sections. In particular, a plausible replacement for a global/local pass system consists of using the concept of importance [15] to assign different feature sizes of interest to different surfaces in the scene. There are however two additional issues which further enhance our approach: the first permits the correct treatment of intra-cluster visibility and the second introduces progressive multi-gridding which permits a rapid interactive response for hierarchical radiosity.

## 6.1 Intra-cluster visibility

Previous clustering algorithms ignore visibility issues when distributing energy received by a cluster to its constituents [14, 12]. This potentially results in light leaks at the scale of the cluster itself, or even creation of energy. As described earlier a bound on energy transfer is computed that ignores visibility (bound of 1 on the visibility error). But visibility is also ignored in the distribution of light when establishing a light transfer. This results in light leaking at the scale of the cluster. This behavior is not only visually displeasing but also flawed: since bounds are computed on irradiance values, that irradiance is distributed to many surfaces which should be shadowed. This results in the creation of energy.

If visibility information inside the cluster with respect to a source cluster can be computed (with some approximation) in time linear in the number of contained objects, the overall time and space complexity of  $O(s \log s)$  for clustering is not modified [14].

We propose the use of an item buffer to quickly evaluate this visibility. The cluster's contents are projected in the direction of the light source using a z-buffer to determine visible surfaces from that direction. By counting instances of an item number in the buffer we obtain an estimate of the projected area of each patch visible from the direction of the source. This is used as the projected area in kernel calculations when computing energy bounds. Note that the resolution of the item-buffer can be adapted to the contents of each cluster, provided we know the size of the smallest object in each cluster. Thus the aliasing problems inherent to the item-buffer approach can be reduced. The same technique is also used at the other end of a link, to evaluate the energy leaving a cluster.

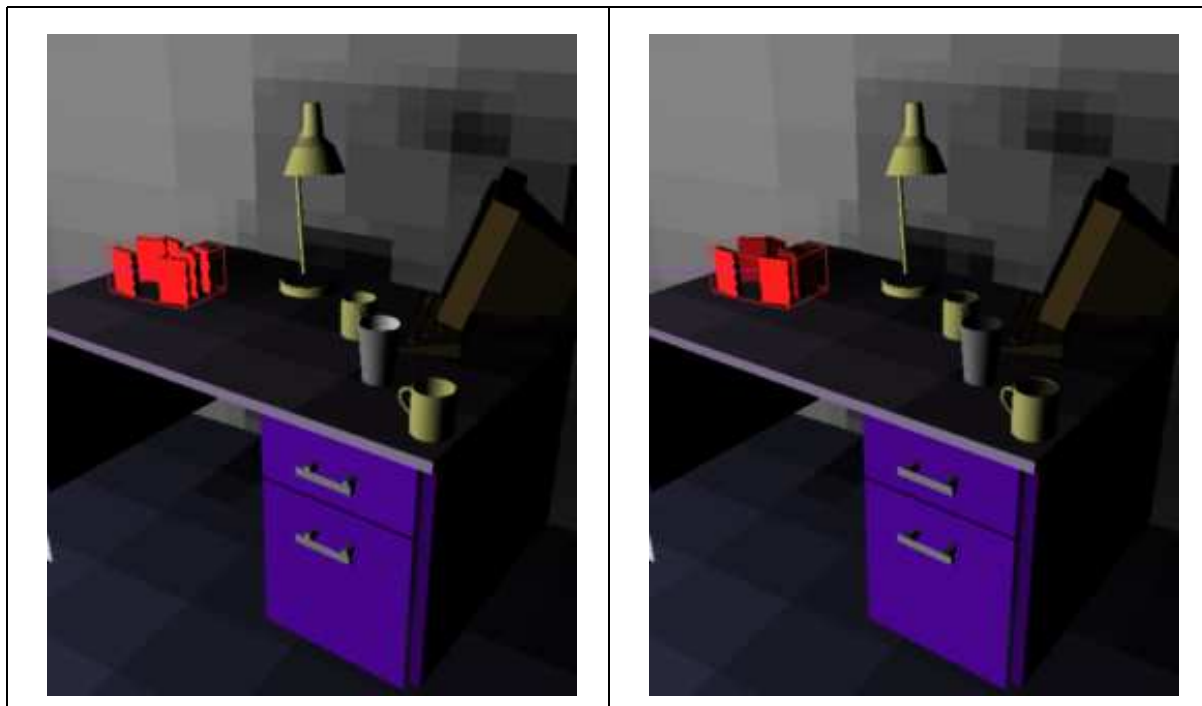


Figure 10: Results of introducing intra-cluster visibility. (left) no intra-cluster visibility: note the relatively uniform illumination of the cluster of books. (right) intra-cluster visibility is used to modulate the energy distribution.

In the images of Figure 10 we present an example where a link (shown in purple) has been created from the light source to a cluster of books. Ignoring intra-cluster visibility results in the creation of energy since all books are fully illuminated. Using the visibility buffer energy is preserved while improving the appearance of the image.

## 6.2 Progressive multi-gridding

In hierarchical radiosity algorithms individual interactions are controlled by the error threshold used to determine subdivision. A global bound on error is difficult to determine and it is consequently difficult for the user to guess what error threshold to use to achieve a certain quality. The problem is exacerbated with clustering, since the subdivision of links is amortized with time, and thus subsequent iterations may become much more expensive than the equivalent operations in simple hierarchical radiosity. Interactivity may then be lost since a rapid first iteration (in the order of a few seconds) may be followed by iterations taking tens of seconds or even minutes.

As a remedy to this we develop a progressive multi-gridding approach. By analyzing the distribution of error bounds on the links created, we can predict how many of these links would survive a given decrease in the error threshold, and thus estimate the time required for a subsequent iteration with the new error threshold. In a manner more intuitive to the user the amount of computation can be specified and the system proceeds to deduce the new threshold to use for hierarchical refinement.

This analysis can be performed at almost no additional cost by recording a histogram of the distribution of error bounds computed. Specifically, the interval between 0 and the current error threshold  $\varepsilon$  is divided into a number of bins, each associated with a counter. Every time a link is created (or left unchanged) during the hierarchical subdivision procedure, we increment the counter for the bin corresponding to the link's error bound. At the start of the next progressive multi-gridding iteration, the new error threshold is chosen such that the sum of all counter for bins with higher error levels is less than a user-specified limit  $k$ . This effectively chooses an error threshold such that at most  $k$  links are refined<sup>7</sup>.

In our implementation of this approach, for a scene with over 6000 polygons we can compute iterations every 2 seconds starting from a coarse solution<sup>8</sup>.

## 7 Conclusions

Important advances towards the goal of providing interactive systems capable of treating very complex environments have been made by hierarchical radiosity [5] and clustering algorithms [14, 12]. Nonetheless several important shortcomings of previous approaches were identified in this paper: (a) visibility error is typically ignored, (b) traditional error metrics do not allow the user to specify a desired level of detail and (c) the introduction of an expensive rendering postprocessing pass is detrimental to the goal of interactivity.

In this paper we introduced a new approach to error estimation based on illumination *features*, which allows the user to choose a level of detail relevant to a given simulation. Specifically, a feature-based error methodology was presented which relates the quality of an image to how well features of a user-determined size have been represented. A practical way to compute feature sizes was proposed using Erosion/Expansion operators.

The principles introduced by the feature-based analysis were used to develop a *multi-resolution visibility algorithm*. The hierarchy constructed for clustering contains transmission information as in [12] and is further augmented with an estimate of maximum feature size its contents can produce. This information is then used to effect a multi-resolution visibility computation. An algorithm which efficiently constructs a suitable hierarchy was also presented. The results of the implementation for isotropic environments show significant computational speedup using MR visibility when the user does not require the accurate representation of small features.

Two additional quality control mechanisms were introduced, which enhance the interactivity of a working system: *intra-cluster visibility* which corrects potential light-transfer error suffered by previous clustering algorithms, and *progressive multi-gridding* which is essential for interactive clustering systems.

The algorithms presented here, in particular the feature-based error metric and the hierarchy construction are first attempts at solving a difficult problem. They nonetheless address important shortcomings of previous approaches. We believe that the introduction of feature-based error and quality evaluation is an important step which will lead to significant acceleration of global illumination algorithms (e.g., via multi-resolution visibility).

<sup>7</sup>Note that each link that is refined may produce many links in the course of further subdivision. Also this analysis ignores the increase in radiosity values due to the previous gathering step. However these factors appear to simply result in a linear factor in the number of links refined.

<sup>8</sup>Examples are shown in the accompanying videotape.

In future work the extension of our approach to non-isotropic environments must be completely developed. It will be interesting to observe the results of the application of our approach to Monte Carlo methods. A more in-depth study of feature-based error metrics must be performed and finally better algorithms for hierarchy construction must be investigated.

## 8 Acknowledgements

George Drettakis's ERCIM fellowship is partially financed by the Commission of the European Communities. Many colleagues have greatly helped us by kindly sharing some of their code or databases. Pat Hanrahan provided parts of the hierarchical radiosity code; Kevin Novins, Jim Arvo and David Salesin provided  $k$ - $d$  tree subdivision code. Jean-Daniel Boissonnat suggested to use erosion/expansion operations. The scene in Figure 7 was assembled using pieces of the Berkeley Soda Hall model. Thanks to Seth Teller and all members of the UC Berkeley walkthrough group.

## References

- [1] James Arvo, Kenneth Torrance, and Brian Smits. A framework for the analysis of error in global illumination algorithms. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '94* (Orlando, FL). ACM SIGGRAPH, New York, July 1994.
- [2] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press, Boston, 1993.
- [3] J. Goldsmith and J. Salmon. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics and Applications*, 7(5):14–20, May 1987.
- [4] Eric A. Haines. Shaft culling for efficient ray-traced radiosity. In P. Brunet and F.W. Jansen, editors, *Photorealistic Rendering in Computer Graphics*, pages 122–138. Springer Verlag, 1993. Proceedings of the Second Eurographics Workshop on Rendering (Barcelona, Spain, May 1991).
- [5] Pat Hanrahan, David Saltzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4):197–206, August 1991. Proceedings SIGGRAPH '91 in Las Vegas (USA).
- [6] Arjan J. F. Kok. Grouping of patches in progressive radiosity. In *Proceedings of Fourth Eurographics Workshop on Rendering*, pages 221–231. Eurographics, June 1993. Technical Report EG 93 RW.
- [7] Dani Lischinski, Brian Smits, and Donald P. Greenberg. Bounds and error estimates for radiosity. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '94* (Orlando, FL). ACM SIGGRAPH, New York, July 1994.
- [8] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. Discontinuity meshing for accurate radiosity. *IEEE Computer Graphics and Applications*, 12(6):25–39, November 1992.
- [9] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '93* (Anaheim, CA, USA), pages 199–208. ACM SIGGRAPH, New York, August 1993.
- [10] Holly Rushmeier, Charles Patterson, and Aravindan Veerasamy. Geometric simplification for indirect illumination calculations. In *Proceedings Graphics Interface '93*. Morgan Kaufmann, 1993.
- [11] J. Serra. *Image analysis and mathematical morphology : 1*. Academic Press, London, 1982.
- [12] François Sillion. Clustering and volume scattering for hierarchical radiosity calculations. In *Proceedings of Fifth Eurographics Workshop on Rendering*, Darmstadt, Germany, June 1994. Eurographics.
- [13] François Sillion and Claude Puech. *Radiosity and Global Illumination*. Morgan Kaufmann publishers, San Francisco, 1994.
- [14] Brian Smits, James Arvo, and Donald P. Greenberg. A clustering algorithm for radiosity in complex environments. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '94* (Orlando, FL). ACM SIGGRAPH, New York, July 1994.
- [15] Brian E. Smits, James R. Arvo, and David H. Salesin. An importance-driven radiosity algorithm. *Computer Graphics*, 26(4):273–282, July 1992. Proceedings of SIGGRAPH '92 in Chicago (USA).
- [16] Seth J. Teller and Patrick M. Hanrahan. Global visibility algorithms for illumination computations. In *Computer Graphics Proceedings, Annual Conference Series: SIGGRAPH '93* (Anaheim, CA, USA), pages 239–246. ACM SIGGRAPH, New York, August 1993.
- [17] John R. Wallace, Kells A. Elmquist, and Eric A. Haines. A ray tracing algorithm for progressive radiosity. *Computer Graphics*, 23(3):315–324, July 1989. Proceedings SIGGRAPH '89 in Boston.

## A Notes on visibility error

### A.1 Averaging visibility

The radiosity received from a source at a point  $x$  on a receiver is given by

$$L(x) = \int_y k(x, y)L(y)v(x, y)dy \quad (1)$$

where  $k(x, y)$  is the radiosity kernel,  $v$  is the binary visibility function and  $y$  spans the area of the source. Let us express the error incurred when factoring out the visibility term as suggested in [5]. Denoting by  $V(x)$  the average visibility of the source patch as seen from point  $x$  ( $V(x) = \frac{1}{A_y} \int v(x, y)dy$ ), we have

$$L(x) = V(x) \int_y k(x, y)L(y)dy + \int_y k(x, y)L(y)\Delta v(x, y)dy \quad (2)$$

where  $\Delta v(x, y) = v(x, y) - V(x)$  is the deviation from the average. Clearly the integral of  $\Delta v(x, y)$  over the source patch is zero, therefore the error introduced is linked to the correlation of the intensity variations on the source, the radiosity kernel, and the visibility variations from point  $x$ .

It is very difficult to make assumptions about such correlations, therefore they are usually ignored as a first approximation. This however is a promising direction of research for the development of better error bounds.

### A.2 Estimation of visibility error using $L^1$ and $L^2$ norm

Consider a patch  $P$  of area  $A$  illuminated by a point source at point  $y$ .

To quantify the visibility error on the receiver patch, we compute the  $L^1$  and  $L^2$  norms of the difference between the visibility function  $v(x, y)$  and its average value over patch  $P$ :

$$\bar{v} = \int_{x \in P} v(x, y)dx$$

To this end we define two areas on patch  $P$ , corresponding to the lit and shadowed regions.

$$\|v - \bar{v}\|_1 = \frac{1}{A} \int_{x \in P} |v(x, y) - \bar{v}|dx \quad (3)$$

$$= \frac{1}{A} \int_{x \in P_{\text{vis}}} |1 - \bar{v}|dx + \frac{1}{A} \int_{x \in P_{\text{invis}}} |0 - \bar{v}|dx \quad (4)$$

$$= 2\bar{v}(1 - \bar{v}) \quad (5)$$

$$\|v - \bar{v}\|_2 = \left( \frac{1}{A} \int_{x \in P} |v(x, y) - \bar{v}|^2 dx \right)^{0.5} \quad (6)$$

$$= \left( \frac{1}{A} \int_{x \in P_{\text{vis}}} |1 - \bar{v}|^2 dx + \frac{1}{A} \int_{x \in P_{\text{invis}}} |0 - \bar{v}|^2 dx \right)^{0.5} \quad (7)$$

$$= \sqrt{\bar{v}(1 - \bar{v})^2 + (1 - \bar{v})\bar{v}^2} \quad (8)$$

$$= \sqrt{\bar{v}(1 - 2\bar{v} + \bar{v}^2 + \bar{v} - \bar{v}^2)} \quad (9)$$

$$= \sqrt{\bar{v}(1 - \bar{v})} \quad (10)$$

Note that both estimates only depend on the average visibility across patch  $P$ , and not on the distribution of the visibility function. Also note the dependency in  $\bar{v}(1 - \bar{v})$ , which means that the error is small for either almost complete occlusion or almost complete visibility.



---

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,  
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY  
Unité de recherche INRIA Rennes, Irista, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1  
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

Éditeur  
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)  
ISSN 0249-6399