On the Computational Complexity of Hierarchical Radiosity

Robert Garmann[2] and Heinrich Müller[1]

[1] Informatik VII (Computer Graphics)
University of Dortmund
D-44221 Dortmund, Germany
mueller@cs.uni-dortmund.de
http://ls7-www.cs.uni-dortmund.de
[2] robert.garmann@gmx.de
http://www.weller-garmann.de

Abstract. The hierarchical radiosity algorithm is an efficient approach to simulation of light with the goal of photo-realistic image rendering. Hanrahan et. al. describe the initialization and the refinement of links between the scene's patches based upon a user-specified error parameter ε. They state that the number of links is linear in the number of elements if ε is assumed to be a constant. We present a result based upon the assumption that the geometry is constant and ε approaches 0 in a multigridding-procedure. Then the algorithm generates $L = \Theta(N^2)$ links where N denotes the number of elements generated by the algorithm.

1 Introduction

Radiosity is used in calculations of computer graphics which determine the diffuse interreflection of light in a three-dimensional environment for photo-realistic image synthesis. The task is to determine the radiosity function B by solving the radiosity equation

$$B(\mathbf{y}) = B_e(\mathbf{y}) + \rho(\mathbf{y}) \int_S G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y}) B(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{1}$$

$\mathbf{x}, \mathbf{y}$ denote surface points of the environment S. The radiosity B located at a point $\mathbf{y}$ is determined by integrating over the radiosity incident from all points $\mathbf{x}$ of the environment S, and weighting the result by a reflection coefficient ρ. $B_e(\mathbf{y})$ denotes the radiosity emitted by $\mathbf{y}$ if the point belongs to a light source. This work concerns radiosity in *flatland*, that is $S \subset I\!R^2$.

The transfer of energy from point $\mathbf{x}$ to point $\mathbf{y}$ is weighted by the terms G and V. $G(\mathbf{x}, \mathbf{y}) = \cos\theta_x \cos\theta_y / (2r_{xy})$ defines the purely geometric relationships in flatland. θ_x and θ_y are the angles between the edge $\overline{\mathbf{xy}}$ of length r_{xy} and the surface normals at $\mathbf{x}$ and $\mathbf{y}$. V is a visibility function which equals one if $\mathbf{x}$ and $\mathbf{y}$ are mutually visible, otherwise zero. See [1, 6] for a derivation of the radiosity equation from physics.

Sometimes the functions ρ, G and V are combined into one function

$$k(\mathbf{x}, \mathbf{y}) = \rho(\mathbf{y}) G(\mathbf{x}, \mathbf{y}) V(\mathbf{x}, \mathbf{y})$$

which is called the *kernel* of the radiosity equation.

Since the radiosity equation cannot be solved analytically in general, approximating algorithms have been developed. One of these algorithms is the hierarchical radiosity algorithm. The hierarchical radiosity algorithm, together with several modifications proposed since its appearance, is widely used because of its practical computational efficiency. In the original paper by Hanrahan et al. [3], a theoretical argument for the good computational behavior was given, too. In this contribution we refine and generalize the complexity analysis, with the surprising result that in a multigridding setting which is relevant in practice, the asymptotic worst-case behavior is an order worse than that given by Hanrahan et al. under a more restrictive assumption.

Section 2 is devoted to a summary of the hierarchical radiosity algorithm. In section 3 we give a proof of the computational complexity of the algorithm for a special simple two-dimensional scene. One basic aid for the proof is the decision on a tractable oracle function. In section 4 we complement the theoretical result by experimental investigations of several simple scenes, using an exact oracle function.

2 Hierarchical Radiosity

The hierarchical radiosity algorithm of Hanrahan et al. [3] is a procedure which solves the radiosity equation using a finite element approach. The radiosity equation is transformed into the linear system of equations

$$b_i = e_i + \sum_{j=1}^{n} k_{ij} b_j, \quad i = 1, \ldots, n, \tag{2}$$

where n denotes the number of surface elements into which the surface of the given scene is decomposed. The coefficients b_i, e_i and k_{ij} represent discrete approximations of B, B_e and k over constant basis functions corresponding to the elements.

All radiosity algorithms have roughly two components. These can be described by *setting up* the equations, i. e. computing the entries of the linear system, and *solving* the linear system. The latter typically invokes some iterative solution scheme.

Hierarchical radiosity considers the possible set of interactions between elements in a recursive enumeration scheme. An interaction between two elements is called a *link*. The algorithm has to insure that every transport, i. e. every surface interacting with other surfaces, is accounted for exactly once. This goal is achieved by applying the following procedure to every input surface with every other input surface as a second argument [5]:

```
ProjectKernel(Element i, Element j)
  error= Oracle(i,j);
  if (Acceptable(error) || RecursionLimit(i,j))
    link (i,j);
  else
    if (PreferredSubdivision(i,j) == i)
      ProjectKernel(LeftChild(i), j);
      ProjectKernel(RightChild(i), j);
    else
      ProjectKernel(i, LeftChild(j));
      ProjectKernel(i, RightChild(j)).
```

First a function `Oracle` is called which estimates the error across a proposed interaction between elements `i` and `j`. If this estimated error satisfies the predicate `Acceptable`, a link is defined between the elements `i` and `j`, and the related coefficient of the kernel is calculated. Resource limitations may require to terminate the recursion even if the error is not acceptable yet. This predicate is evaluated by `RecursionLimit`. If the error is too high then the algorithm recurs by subdividing elements into two child elements. Typically it will find that the benefit in terms of error reduction is not equal for the two elements in question. For example, one element might be larger than the other one, and it will be more helpful to subdivide the larger one in order to reduce the overall error. This decision is taken by `PreferredSubdivision`, and a recursive call is initiated on the child interactions which arise from splitting one of the parent elements.

There are different ways to estimate the error involved with a link between elements `i` and `j`. Hanrahan et al. [3] calculate an approximate form-factor corresponding to the link, and use it to obtain an approximate upper bound on the transferred energy. Several other estimation techniques are possible [4].

The predicate `Acceptable` may itself become more and more stringent during the calculation, creating a fast but inaccurate solution first and using it as a starting point for successive solutions with less error. This technique is called *multigridding*.

3 Analysis for a Simple Scene

3.1 The Result

We are going to formulate a complexity result for the number of elements, N, and the number of links, L, which are generated by the hierarchical radiosity algorithm. In order to do so, we have to specify the behavior of the functions and predicates of the algorithm.

Let $\varepsilon > 0$. We accept a link if the error predicted by the oracle function is less than or equal to ε. The oracle function itself used in the analysis is the approximation

$$\frac{\max(A_x, A_y)}{r_{xy}}$$

of the form-factors between two elements, where A_x, A_y denote the sizes of the two elements involved, and where r_{xy} is the distance between the centers of the elements. The form-factors express the geometric relationship relevant for radiosity exchange between two elements [1, 6]. They are defined as an integral of G($\mathbf{x}$,$\mathbf{y}$)V($\mathbf{x}$,$\mathbf{y}$) over the two elements, divided by the size of the radiosity-sending surface. We use this simple oracle because of clarity of the proof[3].

For the analysis we modify the algorithm slightly. Instead of splitting just one of the elements `i` and `j`, we now split both elements and then recur four times:

```
ProjectKernel(Element i, Element j)
  error= Oracle(i,j);
  if (Acceptable(error) || RecursionLimit(i,j))
    link (i,j);
  else
    ProjectKernel(LeftChild(i), LeftChild(j));
    ProjectKernel(RightChild(i), LeftChild(j));
    ProjectKernel(LeftChild(i), RightChild(j));
    ProjectKernel(RightChild(i), RightChild(j));
```

This modification does not affect the relationship between the number of patches and links seriously. The modified algorithm does not generate more than twice the number of links established by the original algorithm. The number of elements differs by not more than a constant factor less than two.

Finally, in the analysis the predicate `RecursionLimit` is assumed to return `false` for arbitrary pairs of elements.

Using these modifications and specifications we will prove that

$$\begin{aligned} N &= \Theta\left(g\varepsilon^{-1}\right), \\ L &= \Theta\left(N\varepsilon^{-1}\right), \end{aligned} \tag{3}$$

where g is a parameter determined by the initial geometry of the scene.

We can interpret the terms for L and N of this result in different ways. Let us first assume that ε is fixed and g variable. Then

$$L = \Theta(N\varepsilon^{-1}) = \Theta(N),$$

that is, the number of links scales linearly with the number of elements. Hanrahan et al. [3] arrived at this result, because they assumed a constant error threshold.

An other assumption may be that the geometry term g is fixed and ε is varied, e. g. in a multigridding procedure. Then the number of elements becomes $N = \Theta(\varepsilon^{-1})$, or, vice versa, $\varepsilon^{-1} = \Theta(N)$, and we get

$$L = \Theta(N\varepsilon^{-1}) = \Theta(N^2),$$

i. e. the number of links scales quadratically with the number of elements.

[3] The analysis may also be applied to the oracle $\left(\frac{\max(A_x,A_y)}{r_{xy}}\right)^2$, which is a stronger bound of the real error [5]. It can be easily shown that this does not affect the relationship between L and N.

Theorem 1. *Under the assumption that the initial geometry is fixed and the error parameter is variable, the hierarchical radiosity algorithm generates a quantity of links which grows quadratically with the number of elements.* □

The remainder of this section is devoted to the proof of equations (3).

3.2 The Scene

Our analysis concerns the simple scene depicted in Fig. 1. The scene consists of two directly facing parallel two-dimensional patches, each having a length of l units. The perpendicular distance of the two patches is d. An analysis of a general two-dimensional two-patch scene is described in [2].

Let A_x and A_y denote the lengths of two elements generated by the algorithm. Since both initial patches have an equal length l, the modified algorithm assures that $A_x = A_y$ if A_x and A_y are connected by a link. In the following, $A := A_x$ denotes the size of the elements and r the distance between their centers.

3.3 Number of Links

Figure 2 shows the subdivision hierarchies generated by the algorithm for each of the two initial patches. Double arrows are drawn between elements that are linked. Links occur at different levels of the hierarchies. All plotted links together are responsible for connecting a single point of one patch, indicated by a small dot, to all points of the other patch. Nearby points are accounted for by links on deep levels; distant points are represented by links on coarser levels. Every single element is connected to a small subset of elements of the other hierarchy. Figure 3 shows the region of neighbors of an element. Hanrahan et al. [3] stated that the length D of the region of neighbors is constant. We will derive a precise formula which describes how D depends on ε.

Consider any link $A_x \leftrightarrow A_y$ established by the algorithm. The link was introduced because the predicate `Acceptable` was satisfied for the link, thus

$$\frac{A}{r} \leq \varepsilon.$$

¿From this we get the following lower bound for the distance of an established link:

$$r \geq r_{min} := A\varepsilon^{-1}. \tag{4}$$

The link $A_x \leftrightarrow A_y$ may have been established by the algorithm only if its parent link has been refined. The parent link was refined, because it did not satisfy the predicate `Acceptable`:

$$\frac{2A}{r_{parent}} > \varepsilon.$$

¿From this we get

$$r_{parent} < 2A\varepsilon^{-1}.$$

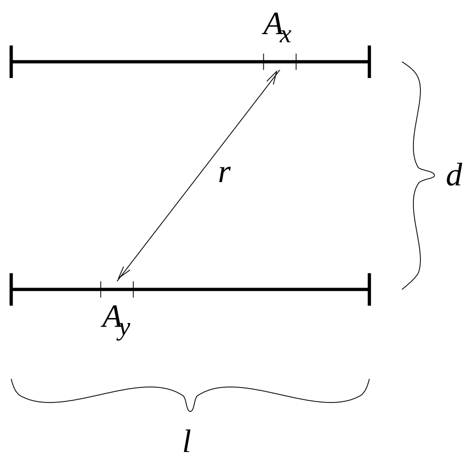

Fig. 1. A simple scene.

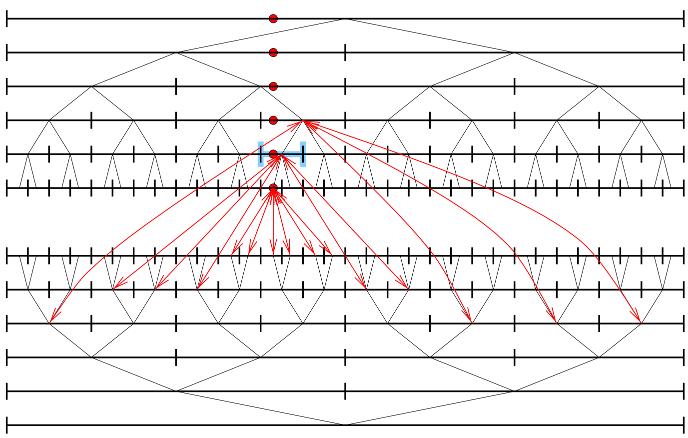

Fig. 2. Links connect elements at different levels.

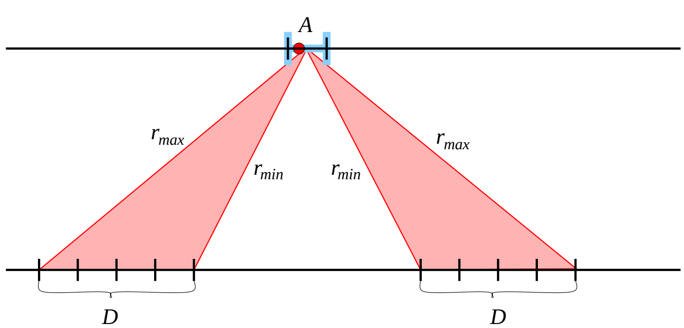

Fig. 3. Region of neighbors of a single element.

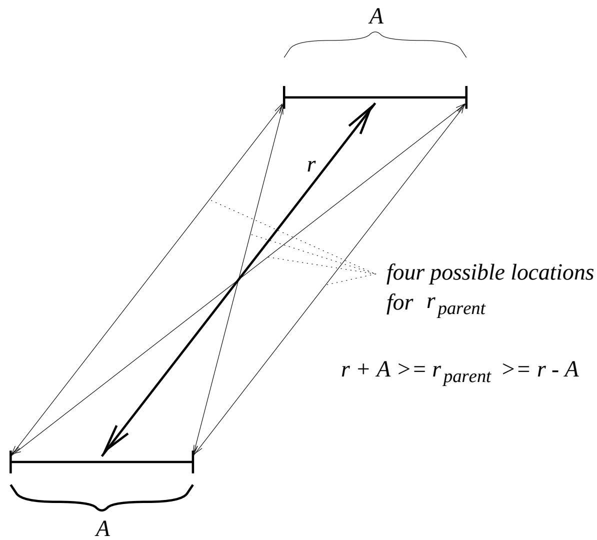

Fig. 4. Relationship between the distance r of a link connecting two elements of size A and the distance r_{parent} of the parent link.

Consider the relationship between r and r_{parent}. Figure 4 shows the link with distance r and the four possibilities where the parent link could be located. By the triangle inequality we know that

$$r - A \leq r_{parent} \leq r + A,$$

or equivalently

$$\exists \delta \in [-1, 1] : \quad r = r_{parent} + \delta \cdot A.$$

¿From this we get the following upper bound on the distance of an established link:

$$r < r_{max} := 2A\varepsilon^{-1} + \delta \cdot A. \tag{5}$$

The length of the region of neighbors, D, can be bounded by the triangle inequality as follows (see Fig. 3):

$$r_{max} - r_{min} \leq D \leq r_{max} + r_{min}$$

where

$$r_{max} - r_{min} = A \cdot (\varepsilon^{-1} + \delta)$$
$$r_{max} + r_{min} = A \cdot (3\varepsilon^{-1} + \delta).$$

Since both bounds are of order $A \cdot \Theta(\varepsilon^{-1})$, we have $D = A \cdot \Theta(\varepsilon^{-1})$ as well.

So far we have derived a formula that describes how the length of the region of neighbors of a single element, D, depends on ε and the size of the element, A.

The number of neighbors of the element is simply $\Theta\left(\frac{D}{A}\right)$, because the neighbors of the element have got the size A. The total number of links in the scene is the number of neighbors times the number of elements, which is

$$L = \Theta\left(N\frac{D}{A}\right) = \Theta\left(N\varepsilon^{-1}\right).$$

This proves the second equation of (3). □

3.4 Number of Elements

The fact that the initial patches we are dealing with are in parallel assures that both initial patches are subdivided uniformly into leaves of equal size. Every leaf is linked to at least its directly facing counterpart, otherwise it would have been subdivided. Let A_{leaf} denote the size of the elements at the leaves. We show that

$$\frac{1}{2}d \cdot \varepsilon < A_{leaf} \leq d \cdot \varepsilon.$$

The bounds can be derived from the value of the predicate `Acceptable` if it is applied to links that have got the minimum distance $r = d$. The upper bound arises from the fact that the predicate `Acceptable` is satisfied for those links which connect pairs of leaves with a minimum distance d between. The lower bound holds because the predicate `Acceptable` was not satisfied for links between two patches of size $2 \cdot A_{leaf}$.

Now we use the term $A_{leaf} = \Theta(d \cdot \varepsilon)$ for the size of the leaves in order to count the number of leaves, N_{leaf}, of the element hierarchy of a single initial patch. The number simply is

$$N_{leaf} = \frac{l}{A_{leaf}} = \Theta\left(\frac{l}{d}\varepsilon^{-1}\right).$$

Since we have only two initial patches, and since a complete binary tree has $2 \cdot N_{leaf} - 1$ nodes, we conclude

$$N = 2 \cdot (2 \cdot N_{leaf} - 1) = \Theta\left(\frac{l}{d}\varepsilon^{-1}\right).$$

By setting $g := \frac{l}{d}$ we have proven the first equation of (3). □

4 Experiments with an Exact Oracle

A potential objection to the given proof is that the oracle function used yields a bad estimation of the error. For that reason, we have numerically analyzed the four simple scenes depicted in Fig. 5 and 6 with an *exact* oracle for the geometry error. Reducing the geometry error is a justified way to gain a small error in the

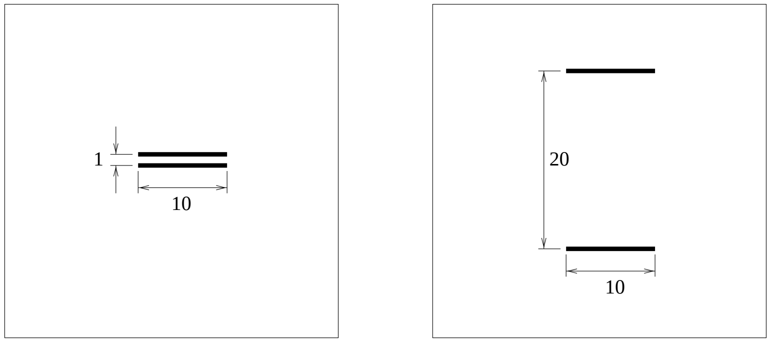

Fig. 5. Two simple scenes. The left scene consists of two parallel patches with a small distance. In the right scene the distance is large.

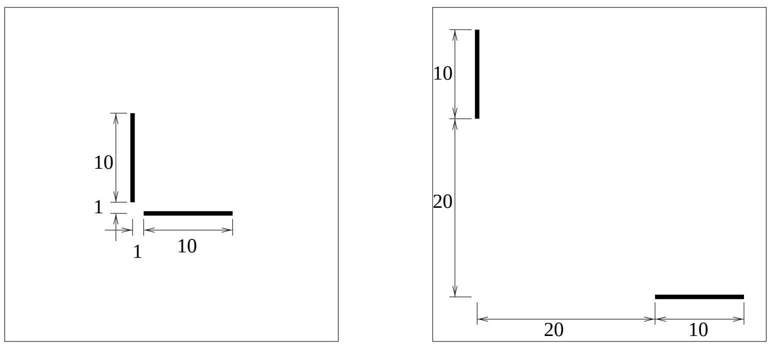

Fig. 6. Two simple scenes with orthogonally located patches. In the left scene the distance is small, in the right scene the distance is large.

resulting radiosity function [5]. We have calculated the L^1-error of the geometry term G evaluated at the centers $\mathbf{x}_c$ and $\mathbf{y}_c$ of two elements:

$$\int_{A_x} \int_{A_y} \mid G(\mathbf{x}_c, \mathbf{y}_c) - G(\mathbf{x}, \mathbf{y}) \mid \mathrm{d}\mathbf{y}\mathrm{d}\mathbf{x}.$$

Actually the L^1-error was approximated using a 1024×1024 discretization of the geometry term G between the two initial patches. The `RecursionLimit` predicate was forced to always return `false`. Furthermore we used the original version of the algorithm where only one of two elements is subdivided if the error is not accepted. In the test the function `PreferredSubdivision` selected the larger element out of its parameters.

For a sequence of continuously decreasing ε-values the resulting elements and links have been calculated. Figures 7 and 8 show a diagram for each of the four scenes. Every resulting configuration is represented by a dot which has the number of links (vertical axis) and the number of elements (horizontal axis) as coordinates. Clearly, because of the discrete nature of the algorithm, in general

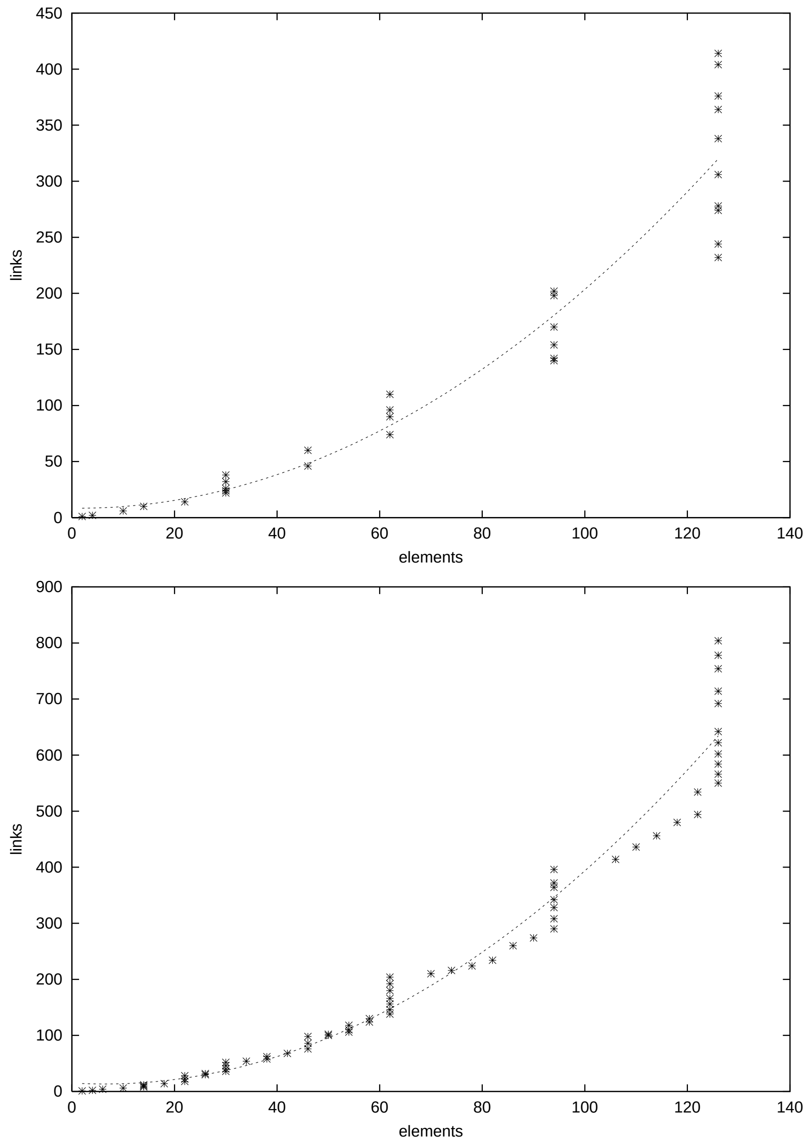

Fig. 7. Empirical analysis for the scenes consisting of parallel patches. The top diagram shows results for the small distance scene, the bottom one for the large distance scene. Measurements have been performed for a sequence of continuously decreasing ε-values. Every dot means one measurement of the number of links (vertical axis) and the number of elements (horizontal axis). The continuous curves have been determined by a least-squares-fit of a quadratic polynomial.

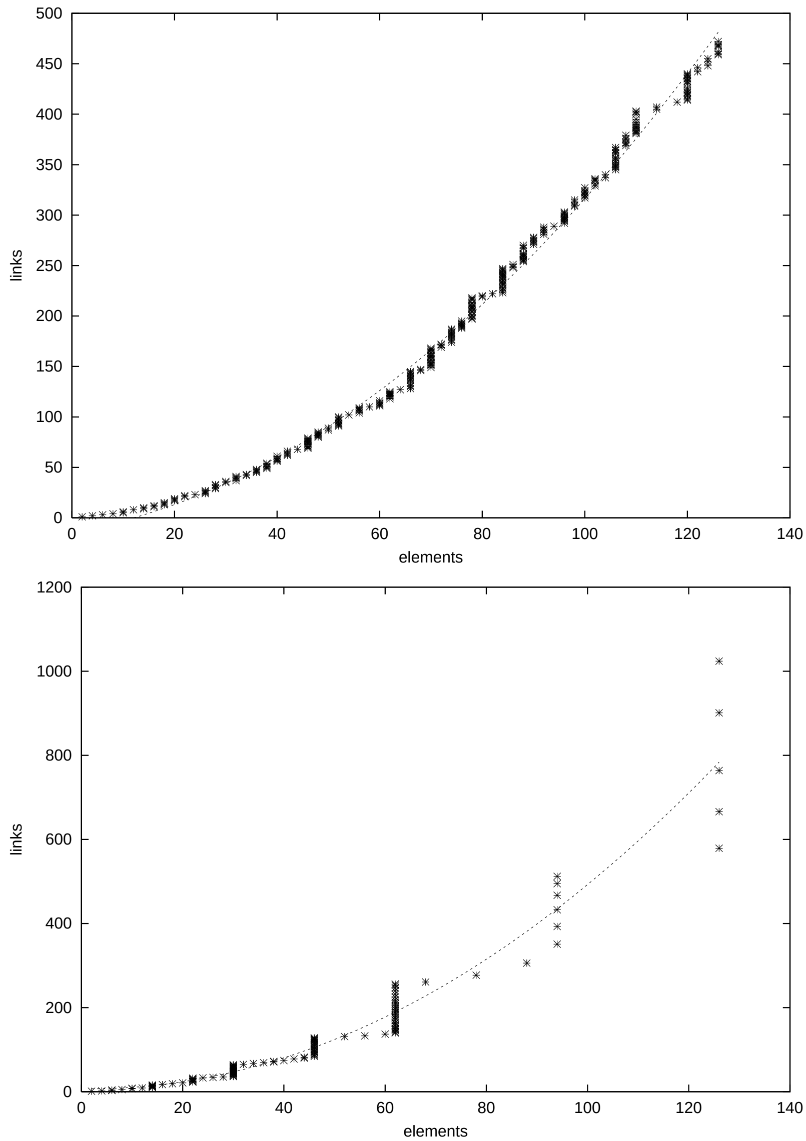

Fig. 8. Empirical analysis for the scenes with two orthogonal patches. The top diagram shows results for the small distance scene, the bottom one for the large distance scene. The continuous curves have been determined by a least-squares-fit of a quadratic polynomial.

there are several different values of L possible for the same N. This arises from the fact that the algorithm may refine links between patches which are already subdivided. Then the number of links increases while the number of patches is not varied. This reveals in "stacked" dots in the diagrams. Nevertheless, the least-squares-fits of quadratic polynomials shown in the diagrams manifest the theoretical result that the number of links depends quadratically on the number of elements.

Acknowledgements

The authors would like to thank Georg Pietrek, Peter Schröder, and Frank Weller for their helpful hints and remarks.

References

1. M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image synthesis.* Academic Press Professional, Cambridge, 1993.
2. R. Garmann. "Hierarchical radiosity - an analysis of computational complexity." Research Report 584/1995, Fachbereich Informatik, Universität Dortmund, Germany, D-44221 Dortmund, August 1995. Also available at `http://ls7-www.informatik.uni-dortmund.de`
3. P. Hanrahan, D. Salzman, and L. Aupperle. "A rapid hierarchical radiosity algorithm." *Computer Graphics*, vol. 25, pp. 197–206, July 1991.
4. D. Lischinski, B. Smits, and D. P. Greenberg. "Bounds and error estimates for radiosity." In *Proceedings of SIGGRAPH 94*, pp. 67–74, 1994.
5. P. Schröder. *Wavelet Algorithms for Illumination Computations.* PhD thesis, Princeton University, November 1994.
6. F. X. Sillion and C. Puech. *Radiosity & Global Illumination.* Morgan Kaufman, San Francisco, 1994.