Dynamic Subdivision in Radiosity

Eric Dubuis Hanspeter Bieri
Institut für Informatik und angewandte Mathematik,
Universität Bern, Länggassstrasse 51, 3012 Bern, Switzerland.
Email: dubuis@iam.unibe.ch, bieri@iam.unibe.ch

Abstract
This paper presents a fast radiosity algorithm for illuminating scenes containing large piecewise polygonal surfaces. Dynamic Subdivision is based on the well known Adaptive Subdivision, introduced by M. Cohen et al. [3]. During the illumination process, patches in areas with high intensity gradients are refined. Contrary to Adaptive Subdivision, the presented algorithm subdivides patches not in a static way. The patch hierarchy is dynamic and adapted to the respective state of the illumination process. To take a decision concerning patch subdivision, more information about the gathered energy of a patch is considered than with Adaptive Subdivision. The results show that this new algorithm can lead to remarkable speedups compared to Adaptive Subdivision.

1 Introduction

The maybe most important part in the generation of photorealistic images by computers is the correct treatment of diffuse interreflections within a virtual scene. The radiosity method was developed in the mid eighties by C.Goral et al. [5] to handle this type of light exchange. It determines the light intensity – more precisely the radiosity – of every surface in the environment.

Radiosity algorithms assume that an environment is discretized into small patches which have constant properties and especially constant brightness. Usually the polygons which form the environment, are too large and must be subdivided. This subdivision is done prior to all illumination calculations. The radiosity value of a patch i can then be determined by means of the equation:

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij}, \tag{1}$$

where B denotes radiosity, E emissivity, ρ diffuse reflectance, F form factor and n the number of patches in the scene. Obtaining this equation for each patch leads to a system of n equations with the n unknown radiosity values B_i.

For an efficient approximation to the solution of this system, the equations may be reformulated:

$$B_j \text{ due to } B_i = \rho_j B_i F_{ij} \frac{A_i}{A_j} \quad \text{for all } j \tag{2}$$

With this formulation it is not necessary to store all n^2 form factors, and another advantage is that the solution process can be stopped when the required accuracy is reached. This technique is well known as *Progressive Refinement* and was introduced for radiosity by M.Cohen et al. in 1988 [2]. It can be physically interpreted as *shooting* energy from one patch to the rest of the environment. In one Progressive Refinement iteration step the brightest patch is chosen. It distributes all its energy to the rest of the environment, i.e. to the other patches. Therefore, the radiosity values of all patches in the environment can be updated in one iteration. As the process continues, the resulting radiosity values get more accurate with each iteration. Usually, the brightest patch in the environment is called the shooting patch or *sender*, all other patches are the receivers. Figure 1 shows the structure of a Progressive Refinement radiosity algorithm.

```
while (not converged) {
  pick brightest patch: i;
  for every patch j
    shoot( i, j);
  update( i);
}
```

Figure 1: Structure of a Progressive Refinement algorithm

After having illuminated the environment, the sender's unshot energy is set to zero in the procedure `update`. This is necessary in order to ensure that the same patch is not repeatedly chosen as the sender.

Adaptive Subdivision

The decision, whether an environment should be subdivided into a few large patches or into many small ones, is important for both, the resulting image and the calculation costs. If patches are too large, details can not be rendered and the resulting image is not accurate. If on the other hand the environment is subdivided into a large number

of small patches, computation time and storage amount will grow fast, since the radiosity method has a time and a space complexity of $O(n^2)$. It is often difficult to make the right choice between accuracy and calculation costs.
To produce high quality images with the radiosity method, it is in general not necessary to refine the whole environment into small patches. Only regions that exhibit high intensity gradients must be subdivided into a finer grid. Unfortunately, these areas are not known a priori and can therefore not be handled correctly within the initial patch subdivision.
In 1986, M.Cohen et.al [3] presented *Adaptive Subdivision*, a substructuring technique which allows to refine areas with high intensity gradients on the fly. With Adaptive Subdivision, all patches with large radiosity gradients are subdivided into smaller patches[1] in order to determine a more accurate radiosity distribution. This concept was one of the major steps in the development of the radiosity method.
After preprocessing and the initial mesh generation, the Adaptive Subdivision algorithm can be applied. A radiosity solution for the initial patches is determined by solving system (1) of the equation or executing the above mentioned Progressive Refinement [2]. Now, all patches with a high intensity gradient are subdivided into smaller subpatches, called *elements*. In a second step, the form factors and with them the radiosity values of these elements are computed. This refinement step can be repeated until a predefined accuracy or a minimum element size is reached.
An algorithm for Adaptive Subdivision with Progressive Refinement may look like the one indicated in Figure 2.

```
level=0;
do {
  while (not converged) {
    pick brightest patch: i;
    for (every element q on level)
      shoot( i, q);
    update( i);
  for (every element q on level)
    if (refineOracle( q, thresh))
      subdivide( q);
  level++;
} while (level < maxLevel)
```

Figure 2: Structure of an Adaptive Subdivision algorithm

[1]There are several ways to subdivide a patch. Usually, a quadrilateral patch is subdivided into 4 subpatches with equal area while a triangular patch is subdivided into 2 or 4 subpatches [1].

```
boolean refineOracle( Patch *p, float thresh)
{
  maxGrad = maximum intensity gradient over p

  if ( maxGrad > thresh )
    return true;
  else
    return false;
}
```

Figure 3: Structure of a refinement oracle

The function `refineOracle`[2] which is shown in Figure 3 plays a key role in this algorithm. It is responsible for all refinement decisions and is based on the intensity gradient of the concerned patch. If this gradient is higher than a predefined threshold `thresh`, the patch has to be subdivided. Information about the intensity gradient can be retrieved from neighboring patches.

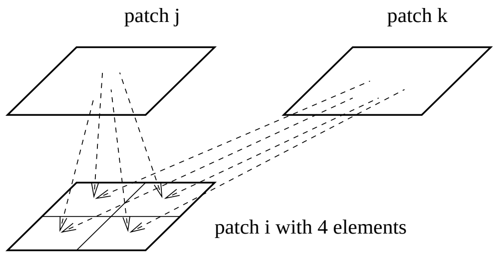

Figure 4: Energy transfer in case of Adaptive Subdivision

Figure 4 shows a very simple environment with 3 patches. It is assumed that patches j and k are light sources illuminating patch i. That is, the Progressive Refinement algorithm chooses patches j and k as senders. If patch k is to send before patch j, patch i gets an unhomogeneous illumination, and therefore has to be refined i. e. subdivided into smaller elements. The radiosity values of these elements are determined in a second step, again using (1). The resulting energy transfer in Figure 4 is shown using dashed arrows.
In Adaptive Subdivision (Figure 2) the radiosity equation system is solved level by level. The radiosity value of each patch on a certain level are determined before any subdivision is performed. Depending on the intensity gradient, patches are refined in a second step. Only then the radiosity values of all previously created elements are computed. All these elements lie in the following level in the hierarchy and their illumination is again obtained by gathering energy from all sending patches.
In this subdivision process, no information from previous energy transfers will be used. These val-

[2]The name *oracle* is quite common for functions that are responsible to take subdivision decisions [6], [4].

ues are ignored and thus must be recalculated for each element.

2 Dynamic Subdivision

Adaptive Subdivision was a very important step in the development of the radiosity method, and the main idea, i. e. to refine patch meshes only where necessary, is still one of the most important approaches among the radiosity acceleration techniques. But nevertheless, this algorithm can be optimized by taking more information into account before subdividing a patch: With **Dynamic Subdivision**, radiosity values of patches that are going to be refined, are reused to determine the radiosity values of the corresponding elements in a more efficient way.

Let us consider again the situation in Figure 4. Patches j and k are emitters and therefore transfer a certain amount of energy to patch i. Patch i is refined because its intensity gradient is higher than the threshold. The energy sent from patch j has an influence on the radiosity of patch i, but does not change the intensity gradient of patch i. Despite of this, with Adaptive Subdivision the energy transfer from patch j to all elements of the subdivision of i has to be computed. No additional accuracy is obtained by doing this computation.

With Dynamic Subdivision, it is possible to avoid these unnecessary energy transport computations. Radiosity values of subpatches are not always computed by using only the energy transfer from the shooting patches. Whenever possible, these radiosity values are derived from the radiosity values of the corresponding superpatches. It follows that Dynamic Subdivision reduces the number of energy transports to be performed. Since the computation of these energy transports from one patch to another is the most expensive part in radiosity calculations, Dynamic Subdivision leads to an significant decrease of computation time without deteriorating the resulting image.

This new technique is well suited for all scenes containing large polygonal patches. By subdividing these patches whenever necessary, an accurate solution for the energy transfer among all elements is found. A more precise discussion of the efficiency of Dynamic Subdivision follows in Section 4.

The Dynamic Subdivision Algorithm

After a patch has been subdivided, no information about the gathered energy is available and it is not known, which energy transfer caused an unhomogeneous illumination and therefore a high intensity gradient. It is possible to store the required information, but this would neutralize one of the benefits of the underlying Progressive Radiosity: the economy of memory.

A better way to make this supplementary information available and to benefit from it is to reformulate the algorithm of Figure 2. Figure 5 shows the new algorithm.

```
while (not converged) {
  pick brightest patch: i;
  for (every patch j)
    level=0;
    do {
      for (every element q (of j) on level)
        shoot( i, q);
        if (refineOracle( q, i, thresh))
          subdivide( q);
      level++;
    } while (level < maxLevel)
  update( i);
}
```

Figure 5: Structure of the Dynamic Subdivision algorithm

With this radiosity algorithm, energy is shot from a selected sender to one patch and to all of its subpatches in one iteration. When taking the decision whether to refine a patch or not, information about the energy transfer from the current sender is still available.

Refinement Oracle

As is seen in Figure 5, a refinement decision is taken with respect to one particular energy transport. It is not necessary to rely only on one radiosity value, determined from the energy transport from all sending patches. Using the Dynamic Subdivision algorithm, it is possible to use an improved oracle for patch refinement. In addition the cumulated radiosity, i. e. the radiosity gathered in all previous iterations, the radiosity gathered from the current energy transport is accessible for a refinement decision.

With the new oracle, this additional information is used to improve refinement decisions. To justify a patch subdivision, three conditions must simultaneously be met:

- The cumulated intensity gradient is higher than a predefined threshold `thresh`.
- The intensity gradient caused by the current energy transport is higher than `thresh`.
- The quotient of the current and the cumulated radiosity is higher than `thresh`.

The main structure of a possible oracle function is shown in Figure 6

Justification: If the cumulated – but not the current – gradient requests a refinement, it is useless to refine the patch with respect to the current

```
boolean refineOracle( Patch *p, float thresh)
{
  maxGrad = maximum cumulated gradient;
  maxNewGrad = maximum current gradient;
  intensityProportion = currentRadiosity /
                        cumulated Radiosity;

  if ( maxGrad > thresh    and
       maxNewGrad > thresh and
       intensityProportion > thresh )
    return true;
  else
    return false;
}
```

Figure 6: Refinement oracle for Dynamic Subdivision

sender. This would not lead to any improvement. If on the other hand only the current intensity gradient is higher than the threshold, this energy transport has no influence on the overall intensity gradient, and again a subdivision would not improve the result. The third criterion, too, is quite important. A refinement only makes sense if the current energy transport actually has an influence on the patch's intensity.

Element Hierarchy

The initial patch meshing is performed as a preprocessing step. This mesh will not change during the illumination process. It is important to choose the patch size depending on the actual environment. If patches are too large, fine details can not be handled; if they are too small, too many computations are necessary for large areas with low intensity gradients. All patches created in this preprocessing are potential senders. The Progressive Refinement algorithm will pick the brightest patch in every iteration step out of these initial patches, not out of the elements. This means, that a patch which has been subdivided can not be deleted. The elements created during the radiosity process are therefore stored in a hierarchy, for instance a quadtree.

The patch hierarchy established during the energy exchange using Adaptive Subdivision is static. If a patch has been subdivided in one iteration, the resulting elements have to be treated as receivers in every following iteration, whether or not the energy transport requires a subdivision of the superpatch. It is therefore not possible to change this hierarchy or to suppress any energy transport. Dynamic Subdivision avoids this by redefining the hierarchy in every iteration.

The situation in Figure 7 is the same as the one in Figure 4. We assume that the energy transfer from patch k to patch i causes a refinement of i and therefore an energy transfer from k to the 4

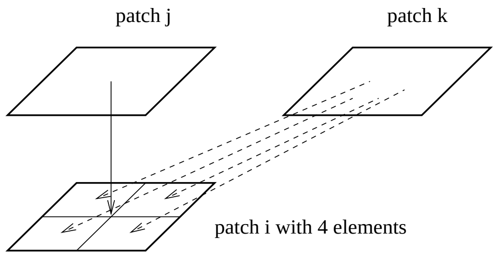

Figure 7: Energy transfer in case of Dynamic Subdivision

subpatches of i. This transfer is indicated by the dashed arrows. The energy transfered from patch j to Patch i has no influence on the intensity gradient of i. i needs not be refined with respect to the sending patch j. This energy transfer (indicated by the solid arrow) leads to an accurate illumination of patch i by sender j.

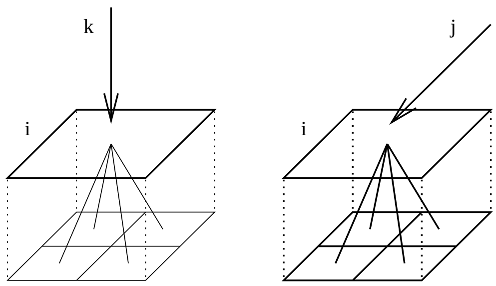

Figure 8: The resulting hierarchy of patch i with Dynamic Subdivision. Patch i receives energy from patch j (left) and from patch k (right)

This example and Figure 8 show how the hierarchy can change during the illumination process. Although patch i has been subdivided due to the energy from patch k, the resulting elements need not be considered when calculating the energy transfer from patch j to patch i. The subdivision hierarchy of patch i depends on whether j or k is the sender. Three different cases are possible:

- The hierarchy caused by the current energy transfer is the same as the one in previous iterations. In this case, radiosity calculations are the same as those with Adaptive Subdivision. Exactly the same energy transfers are computed.
- The current energy transfer causes a finer subdivision than those in all previous steps. The elements created in this iteration have to be initialized with an appropriate radiosity value, reflecting previous energy transfers. This can be achieved by simply initializing them with the cumulated radiosity values of their superpatch.
- If the current energy transfer leads to a coarser subdivision, or no subdivision at all, the en-

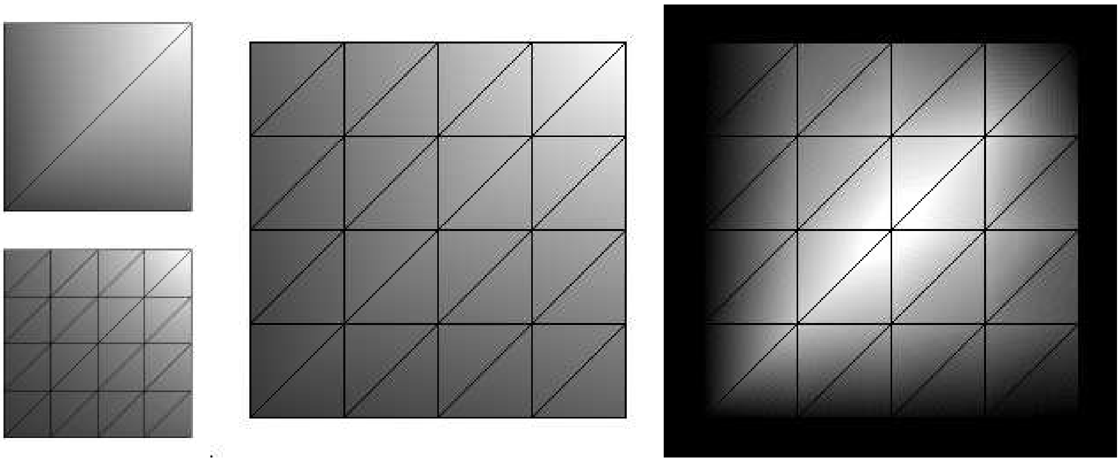

Figure 9: Comparison between Adaptive and Dynamic Subdivision.

ergy transfer from the current sender to the elements lower in the hierarchy has to be derived.

Energy Transport Derivation
The energy transport to an element can be derived in a very simple way. If the illumination from the currently sending patch causes no refinement of the receiver, it has no influence on the patch's intensity gradient. The energy can therefore be distributed in proportion to the areas to all of its elements. Since radiosity is independent of the area, the radiosity caused by the current sender can directly be passed to the elements.
With this method, the radiosity values of every element in a hierarchy can be determined without computing the exact amount of transfered energy for each element-patch pair.
Figure 9 shows a meshed surface, with its radiosity computed once with Adaptive and once with Dynamic Subdivision. The environment was chosen similarly to the one showed in Figures 4 and 7. A surface is initially subdivided into two triangular patches during preprocessing and refined into 32 elements during the radiosity process. The top-left picure shows the radiosity distribution without any refinement, the bottom-left picture shows it after applying Adaptive Subdivision, and the picture in the middle shows the result from Dynamic Subdivision. The rightmost picture is a visualization of the radiosity difference between Adaptive and Dynamic Subdivision. Regions with larger differences are visualized brighter.

3 Determining the Radiosity Value of Subpatches

After a Progressive Refinement iteration, i. e. after a selected patch has shot its energy to the rest of the environment, the radiosity values of all leaf elements in the hiearchy are determined. In order to determine the next sending patch, these radiosity values are used to compute the respective radiosity value of all root elements in the hierarchy.
Assuming that patch i has been subdivided into r elements, this updating is done by summing up the area-weighted radiosity values of these r elements:

$$B_i = E_i + \rho_i \sum_{j=1}^{n} \left[\frac{1}{A_i} \sum_{q=1}^{r} B_j F_{qj} A_q \right] \qquad (3)$$

Applying this updating procedure recursively to the elements at every level of the hierarchy leads to the desired result. Now the next sending patch will be chosen out of the initial patches.

4 A Concrete Example

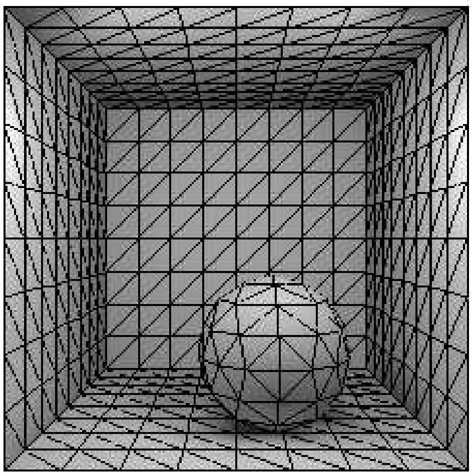

Figure 10: The initial mesh

The effects of applying Dynamic Subdivision to a scene will be illustrated using a simple model of a square room containing a sphere illuminated by two light sources placed on the left and the right side of the front. Figure 10 shows the mesh

resulting from initial subdivision, rendered with the Progressive Radiosity algorithm, as shown in Figure 1.
The acceleration we can achieve with Dynamic Subdivision compared to Adaptive Subdivision depends on two parameters: the number of Progressive Refinement iterations and the depth of the subdivision hierarchy.

Iterations	Energy Transfers Adaptive	Energy Transfers Dynamic	Speedup Factor
1	4604	4604	1.00
10	63440	19980	3.18
100	699600	93108	7.51
1000	7112000	792584	8.97

Table 1: Comparison between the numbers of energy transfers performed by Adaptive and Dynamic Subdivision, depending on the number of iterations

Level	Elem.	Energy Transfers Adapt.	Energy Transfers Dynamic	Speedup Factor
1	776	77600	77600	1.00
2	2836	283600	83176	3.41
3	6996	699600	93108	7.51
4	14936	1493600	112624	13.26

Table 2: Effects of the depth of the subdivision hierarchy

- The number of Progressive Refinement iterations determines the accuracy of the energy distribution within the environment. The efficiency of Dynamic Subdivision increases with the number of iterations. In Table 1 the numbers of energy transfers performed by Adaptive and Dynamic Subdivision are compared. The test environment was initially subdivided into 776 patches and then refined into 4604 – 7112 elements. The maximum subdivision level was limited to 3. Figure 11 illustrates the results. For every row in Table 1 a picture is shown.

- The depth of the subdivision hierarchy is responsible for the accuracy of the resulting image. The deeper the hierarchy, the more elements are generated. Table 2 shows for both algorithms the number of energy transfers for different refinement levels. Again, initial subdivision leads to 776 patches. The results are shown in Figure 12.

Emitting Patches	Energy Transfers Adaptive	Energy Transfers Dynamic	Speedup Factor
2	9922000	786176	12.62
128	6668000	788868	8.45

Table 3: Comparison of the speedup factors depending on the number of emitting patches.

5 Implementation

The ease of implementation is one of the main advantages of Dynamic Subdivision. In order to extend an Adaptive Subdivision implementation to Dynamic Subdivision only a few changes are necessary: Only the functions handling with

- the Progressive Refinement iterations,
- the subdivision criteria, and
- the subdivision of a patch itself

have to be adapted to the new demands.
Hierarchical Radiosity, introduced by Hanrahan et al. in 1991 [6], has similar advantages as Dynamic Subdivision, but its implementation is more complicated.

6 Conclusions

Compared to the more classical approaches the Dynamic Subdivision algorithm proposed in this report reduces the number of energy transports that need to be considered for generating an accurate energy distribution within an environment.

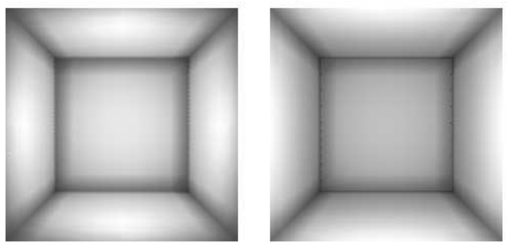

Figure 13: Simple room, illuminated by one small light source (left, 2 emitting patches) and by the whole front wall (right, 128 emitting patches).

Dynamic Subdivision is well suited for environments with a few large patches with high intensity gradients. These large patches need to be subdivided into many finer elements during the radiosity computations. The algorithm works best if an enviroment is only illuminated by a few light sources. In this case, energy transfers from the light sources will cause high intensity gradients and therefore a subdivision of the receiving elements. Energy

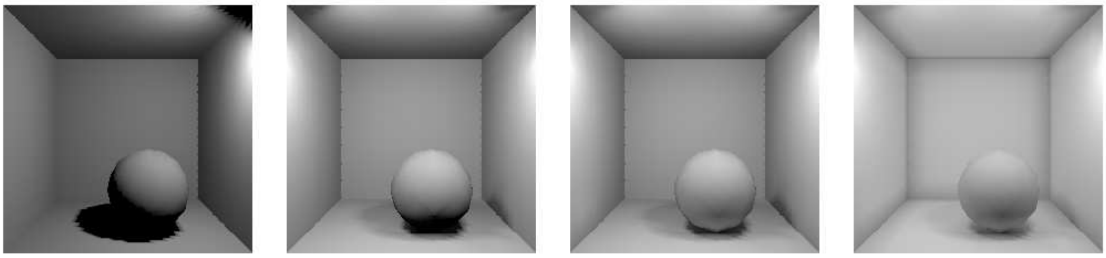

Figure 11: The number of iterations grows from left to right

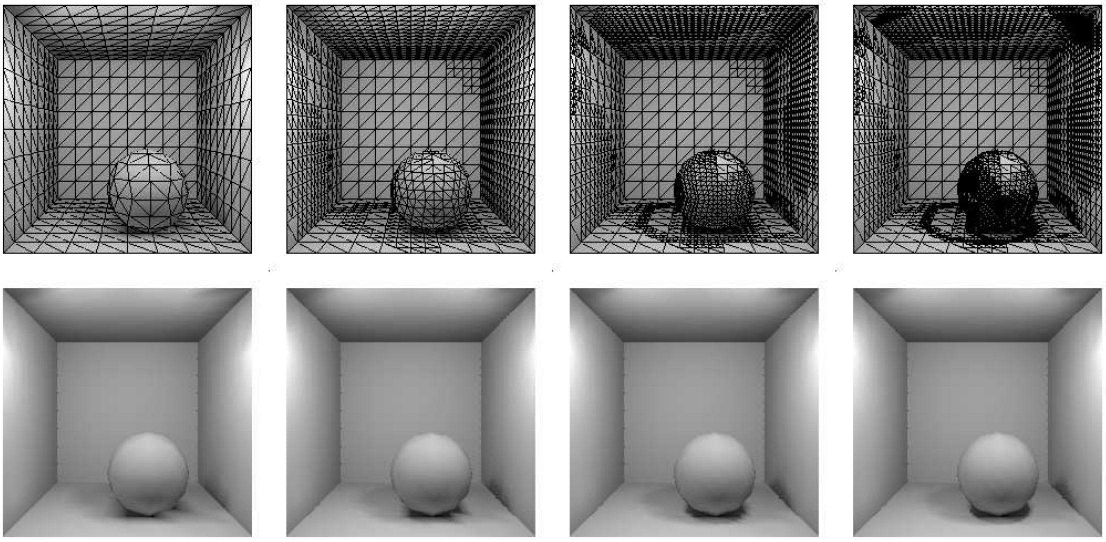

Figure 12: The hierarchy depth grows from left to right

Emitting Patches	Energy Transfers Adaptive	Energy Transfers Dynamic	Speedup Factor
2	9152000	16176	566
128	5900000	20868	283

Table 4: Comparison of the number of patch-element energy transfers

transports from non-emitting patches generally contribute no significant part of the overall illumination of a receiver and, therefore, do not cause any refinement. Figure 13 shows an empty room illuminated one time by only one light source, and one time by the whole front wall. The comparison of the resulting speedup factors with those of Adaptive Subdivision is summarized in Table 3. In both pictures the subdivision level was limited to 4, and 1000 Progressive Refinement iterations have been computed. The room was initially subdivided into 768 patches.

The best way to see the benefits of Dynamic Subdivision is to compare the numbers of patch-element energy transfers, thus ignoring energy transfers to initial patches. In both, Adaptive and Dynamic Subdivision, energy transfers to the initial patches have to be computed anyway. They provide the basis for further patch refinement. In Table 4 the comparison of the number of patch-element energy transports are given for the same environment and conditions as in Figure 13.

When starting with a new radiosity implementation, one should probably turn directly to the Hierarchical Radiosity, as some of the major new developements in Radiosity are based on this technique: for instance Important Driven Radiosity [8] and Clustering [7]. If a radiosity implementation already exists, then including Dynamic Subdivision can make it much more efficient with just a few minor adaptions.

References

[1] Daniel R. Baum, S. Mann, K. P. Smith, and James M. Winget. Making Radiosity Usable: Automatic Preprocessing and Meshing Techniques for the Generation of Accurate Radiosity Solutions. *Computer Graphics (SIGGRAPH Proceedings)*, Volume 25(No. 4):pages 51–60, July 1991.

[2] Michael F. Cohen, S.E. Chen, John R. Wallace, and Donad lP. Greenberg. A Progressive Refinement Approach to Fast Radiosity Image Generation. *Computer Graphics (SIGGRAPH Proceedings)*, Volume 22(No. 4):pages 75–84, August 1988.

[3] Michael F. Cohen, Donald P. Greenberg, David S. Immel, and Philip J. Brock. An Efficient Radiosity Approach for Realistic Image Synthesis. *IEEE Computer Graphics & Applications*, pages 26–35, March 1986.

[4] Michael F. Cohen and John R. Wallace. *Radiosity and Realistic Image Synthesis.* Academic Press, 1993.

[5] Cindy M. Coral, Kenneth E. Torrance, Donald P. Greenberg, and Kennett Battaile. Modelling the Interaction of Light Between Diffuse Surfaces. *Computer Graphics (SIGGRAPH Proceedings)*, Volume 18(No. 3):pp. 213–221, July 1984.

[6] Pat Hanrahan, David Salzman, and Larry Aupperle. A Rapid Hierarchical Radiosity Algorithm. *Computer Graphics (SIGGRAPH Proceedings)*, Volume 25(No. 4):pp. 197–206, July 1991.

[7] Brian E. Smits, James R. Arvo, and Donald P. Greenberg. A Clustering Algorithm for Radiosity in Complex Environments. *Computer Graphics (SIGGRAPH Proceedings)*, Volume 28:pp. 435–442, July 1994.

[8] Brian E. Smits, James R. Arvo, and David H. Salesin. An Importance-Driven Radiosity Algorithm. *Computer Graphics (SIGGRAPH Proceedings)*, Volume 26(No. 2):pp. 273–282, July 1992.