

Lecture Notes: Hierarchical Radiosity

Dani Lischinski

March 31, 2004

1 Problem with Plain Radiosity

In order to get nice pictures, we need to discretize the scene surfaces into fine meshes. Large elements (with low order) cannot provide a good approximation to the radiosity functions in interesting scenes.

Fine mesh means more elements, but the cost of the solution grows superlinearly with the number of elements: n elements implies n equations and n^2 form factors.

2 Substructuring

Considering two large surfaces far away from each other, it is clear that if each surface is subdivided into k elements, then computing k^2 form factors to represent the exchange of energy between the two surfaces is an overkill. While the energy arriving at a surface must be represented accurately using many elements, when the surface is functioning as a light source, the variations in energy between the element can often be replaced by their average value.

This observation has led to the idea of *substructuring*, a 2-level hierarchy of elements. First subdivide the scene surfaces coarsely into “patches”, then subdivide each patch further into smaller elements. Patches play the role of sources, while elements receive energy from the patches. Thus, in the previous example, instead of computing k^2 form factors, only $2k$ form factors are used to approximate the light transfer between the two patches.

Let N be the number of patches, n the number of elements ($n \gg N$). Using the substructuring idea above reduces the number of form-factors from n^2 to nN . The radiosity of each patch is computed by area-averaging its elements.

The problem with the substructuring is that its 2-level hierarchy is not flexible enough. For example consider two large surfaces meeting along an edge. Using large patches might lead to inaccuracies in areas where the distance is small.

3 N-Body simulation

In 1991 Hanrahan, Salzman, and Aupperle came up with a more general idea. They were inspired by efficient N-body simulation algorithms [Appel 85, Barnes and Hut 86, Greengard 88].

Suppose you want to simulate the evolution of a galaxy of stars. You have some initial configuration, with each star having a mass, an initial position and an initial velocity. To simulate the evolution of the system over time, we should take small time steps. At each timestep, we compute the gravitational forces on each star, derive the resulting acceleration, and modify the position and the velocity of the star accordingly. How much work must we do in each time step? The naive answer is $O(n^2)$, since for each star we have to compute and sum the gravitational forces from all $n - 1$ remaining stars. Thus, the simulation time grows quadratically in the number of stars.

However, note that gravitational forces drop off with the square of the distance. This means that the farther away a star is, the less important the force it creates. The idea to speed up the simulation is then as follows: an entire cluster of stars can be represented as a single mass for the purpose of computing the gravitational forces at a distant point. So, let's create a hierarchical subdivision of space (an octree). Given an error threshold, for each point we can traverse the tree and create a link (representing a force computation) to a cluster of all points inside an octree node once the node is far enough. It can be shown that if we do it in this way, each point will have $O(cn)$ interactions, where c is constant for a fixed desired accuracy ϵ . As ϵ approaches 0, we'll get the same n^2 behavior that the naive approach gives, but things will be much better as ϵ increases (even slightly).

4 Hierarchical radiosity

The analogies to radiosity are obvious: in radiosity we also compute terms between elements (the form factors), whose magnitude drops off as the square of the distance. This means that for a fixed accuracy, the farther the surfaces are from each other, the more elements can be clustered together into a single interaction (form-factor).

The hierarchical radiosity algorithm starts with a set of coarse patches (typically the input surfaces, if they are polygons). Given an error tolerance and minimal element size as inputs, the algorithm refines the interactions until all interactions are within the desired tolerance. Each interaction is then stored as a link. The set of links is really a hierarchical representation of the form-factor matrix. The system of equations is solved by iteratively performing the following two steps:

1. Gather: Each patch gathers energy through all of its links;
2. Push-pull: On each surface, collected energy is propagated down and up the hierarchy.

Each such iteration is equivalent to a Jacobi iteration of classical radiosity, and iterations continue until convergence has been achieved (the change in the patch radiosities between successive iterations falls below some small threshold).