Delaunay Triangulations and the Radiosity Approach

Sven Schuierer*

October 1990

Abstract

The radiosity approach requires the subdivision of complex surfaces into simple components called patches. Since we assume to have constant intensity over a patch, the generation of regular patches is a desirable property of the subdivision algorithm.

We show that constrained Delaunay triangulations produce patches that are as close to equilateral triangles as possible and thus are well suited for the partitioning of surfaces into patches.

Since a number of optimal algorithms to generate constrained Delaunay triangulations have been published, the implementation presented here made use of the earlier work. The implementation consists of a rather simple modeling tool called POLY, a fast triangulation algorithm for arbitrary polygons and the form factor computation combined with a z-buffer output module.

1 Introduction

In recent years the radiosity approach has emerged as the second main technique besides ray tracing for realistic image generation. The correct simulation of light in an environment is the basic problem encountered in the creation of realistic images for a raster display.

Ray tracing in its traditional form tries to solve this problem by using a ray based light model that is well suited to simulate perfect specular reflection and transmission. Since its first publication in 1980 by Whitted [Wh 80] it has become the dominating method for realistic image generation and has produced very impressive results. However, ray tracing has great difficulty in modeling the energy exchange between diffuse surfaces.

It remained to the radiosity approach first published in 1984 by Goral et al. [GTGB 84] to introduce new concepts based on previous work in thermal engineering to simulate the interaction of light between diffusely reflecting surfaces. It thus accurately calculates the global illumination effects. Though the first publications of the radiosity approach were restricted to ideal diffuse environments, many proposals have since been made as how to

*Institut für Informatik, Rheinstr. 10–12, Universität Freiburg, D-7800 Freiburg, West-Germany.

include specular reflections [ICG 86, WCG 87, SPL 88]. Most of these try to combine the advantages of ray tracing and some extension of the "simple" radiosity method which will be considered solely in this paper.

2 The Simple Radiosity Approach

In 1986 Kaijya and Immel et al. [Kai 86, ICG 86] presented independently an equation exactly describing the intensity of light leaving a surface in a given direction at a given point in terms of the total incident light on the surface:

$$i(\Omega_{out}, x) = E(\Omega_{out}, x) + \int_{all\ \Omega_{in}} \varrho(\Omega_{in}, \Omega_{out}, x)\, i(\Omega_{in}, x')\, \cos\theta\, d\omega' \tag{1}$$

where

Ω_{in} resp. Ω_{out} is the direction of incoming resp. outgoing light,

$i(\Omega, x)$ is the intensity of light leaving point x in direction Ω,

ϱ is the reflection factor depending on the location and incoming and outgoing directions,

$E(\Omega_{out}, x)$ is the self-emission of light in direction Ω_{out} at location x, and

x' is the first point visible from x in direction Ω_{in}.

The intensity leaving a surface in a given direction is composed of the self emitted light at that point plus the contributions of light arriving at the point weighed with the surface dependent reflection factor.

If we restrict ourselves to ideal diffuse environments the outgoing and incoming light as well as the reflection factor become independent of the directions and (1) simplifies to:

$$i(x) = E(x) + \int \varrho(x)\, i(x')\, \cos\theta\, d\omega' \tag{2}$$

Although equation (2) is a lot simpler than (1) there is still no obvious way to solve it even for small scenes. The basic idea of the radiosity approach is now to assume a constant intensity over small, finite areas — so called patches — of a surface. This corresponds to a discretization of (2) breaking the integral into a sum. After some simple transformations we get the set of linear equations:

$$i_k = E_k + \varrho_k \sum_{j=1}^{n} F_{kj} i_j \tag{3}$$

where i_k, E_k and ϱ_k are the intensity, self emission and reflection factor of patch k and

$$F_{kj} := \frac{1}{A_k} \int\int \frac{\cos\theta_k \, \cos\theta_j}{r^2} dA_j \, dA_k.$$ [1]

The coefficients F_{kj} of (3) are called form factors, a well known quantity in radiation heat transfer. A form factor F_{kj} represents the fraction of energy that leaves patch k and lands on j. From the definition of the form factors it can be seen that F_{kj} only depends on the geometry of the scene but not on the distribution of light. Further it should be noted that the step from equation (2) to (3) is just a rough approximation which heavily depends on the number, size and location of the patches.

Since equation (3) is linear, it can be easily solved using standard methods like the Gauss-Seidel-iteration once the form factors F_{kj} are known.

The actual computation of the form factors represents the core of the radiosity method. Since the integral defining a form factor can not be solved analytically but for very simple geometries, a numerical integration method for the calculation of F_{kj} has to be developed. It is in no way obvious how to do this efficiently.

Cohen et al. overcame these difficulties with two now famous algorithms. In 1985 the hemi-cube method was published which computes a close approximation of the form factors simply by projecting patches onto a hemi-cube and weighing the projections according to their contribution to the value of the form factor [CG 85]. Thus it is possible to compute the F_{kj} for arbitrary environments.

The second algorithm introduced the technique of substructuring patches into elements thus greatly reducing the number of form factors to be computed as well as the size of the set of linear equations to be solved [CGI 86]. Still the computational requirements for the form factor calculations remain the bottleneck of the radiosity method.

3 Patch Generation

As it was pointed out before, the geometry of patches plays an important role in the radiosity approximation of global illumination effects. In our implementation we restricted ourselves to planar surfaces due to the modeling tool used.

In previous publications little was said about how to achieve an appropriate subdivision of complex surfaces.

Nishita and Nakamae are the only ones to describe in detail the generation of rectangular patches [NN 85]. Their method basically involves partitioning the enclosing rectangle of a given surface. As a result patches can lie partially or fully outside the actual surface and have to be weighed accordingly.

Cohen et al. [CGI 86] allowed surfaces to be arbitrary polygons with holes but required the user to divide complex faces interactively into simple components such as rectangles, parallelograms etc.. For large scenes this procedure leads to rather awkward results.

[1] If the radiosity B is used in (3) instead of i, then F_{kj} has to be divided by π.

A user-independent partitioning algorithm is required that can handle arbitrarily complex polygons. The patches generated by this algorithm should, of course, ly completely inside the considered face.

It is a well known fact that every polygon can be partitioned into triangles. So a triangulation algorithm seems to be the natural choice to subdivide polygons into patches.

Since a triangulation is usually not unique, the question arises what further properties of triangular patches should be regarded as desirable and how they can be achieved.

3.1 A desirable property of patches

In order to find a useful property that we can require the patches to comply with we have to recall the basic assumption of the radiosity method. It regards the intensity distribution over a patch P as constant. Another way to view this assumption is to consider the intensity at the centre of P as an approximation for the rest of P. So, in fact, we are dealing with a sample of surface points and the subdivision into patches determines the scope of one sample point, i.e. the points that are regarded to have the same intensity.

From the above observation it is clear that a patch of a given size should ideally be circle-shaped in order to minimize the distance from the centre to the outermost points of P.

If we restrict ourselves to triangles the equilateral triangle is obviously the ideal shape. So we require a triangulation that produces patches that are as "regular" or as "equilateral" as possible. Fortunately, there exists a well studied triangulation that satisfies these conditions: the Delaunay triangulation.

The Delaunay triangulation is the straight line dual of the Voronoi diagram. See [PS 85] for definitions and some applications of the Delaunay triangulation and the Voronoi diagram.

Instead of being defined as the dual of the Voronoi diagram the Delaunay triangulation can be characterized independently [Ed 87]:

Definition 1 *Let S be a set of points in the plane. A triangulation T is a Delaunay triangulation of S if for each triangle $t \subset T$ the circle through the vertices of t contains no other vertex $v \in S$.*

The property which makes this triangulation so attractive for the partitioning of polygons into patches is that it maximizes the minimum angle over all triangulations for a given point set. Moreover, the Delaunay triangulation (DT) produces the lexicographically biggest sequence of angles possible, i.e. if the angles are sorted in non-decreasing order:

$$\alpha_1^{DT} \leq \alpha_2^{DT} \leq \cdots \leq \alpha_i^{DT} \leq \cdots \leq \alpha_{3m}^{DT} \qquad \text{(for } m \text{ triangles)}$$

then we have for any triangulation T that differs for the first time in position i (i.e. $\alpha_1^T = \alpha_1^{DT}$, $\alpha_2^T = \alpha_2^{DT}$, $\ldots$, $\alpha_i^T \neq \alpha_i^{DT}$) that $\alpha_i^T < \alpha_i^{DT}$.

We call a triangulation with this property according to [Ed 87] globally equilateral.

Unfortunately, the above definition of a Delaunay triangulation applies only to point sets and does not mention prespecified edges which are needed to triangulate polygons. To this end we have to introduce the concept of a constrained Delaunay Triangulation [LL 86]:

Definition 2 *Let G be a straight line planar graph. A triangulation T is called a constrained Delaunay Triangulation (CDT) of G if each edge of G is an edge of T and for each triangle $t \subset T$ the circle through the vertices v_1, v_2, v_3 of t contains no vertex v such that the line segment $\overline{vv_i}$ is not intersected by an edge of G for at least one $i \in \{1, 2, 3\}$.*

Figure 1 illustrates definition 2. Intuitively the edges of G cut off a part of the Delaunay

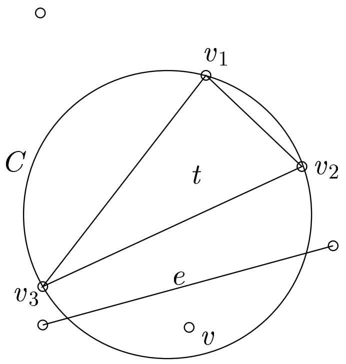

Figure 1: Definition of a Constrained Delaunay Triangulation

circle of t. In order to see that definition 2 is a proper generalization of definition 1 we first note that if G has no edges then both definitions are equivalent, i.e. the CDT of a graph G with $E(G) = \emptyset$ is a Delaunay triangulation.

The reason we introduced Delaunay triangulations was that they are globally equilateral in the unconstrained case. This property carries over to CDTs as the following theorem shows [LL 86]:

Theorem 1 *Let $G = < V, E >$ be a straight line planar graph and T a triangulation of $V(G)$ such that each edge of G is in T. Then we have for the ordered sequences of angles:*

$$\alpha_1^T \le \alpha_2^T \le \cdots \le \alpha_{3m}^T \qquad \text{and} \qquad \alpha_1^{CDT} \le \alpha_2^{CDT} \le \cdots \le \alpha_{3m}^{CDT}$$

where m is the number of triangles in T resp. $CDT(G)$ that

$$\text{if } \alpha_k^T = \alpha_k^{CDT} \text{ for } k = 1, \ldots, i-1 \text{ and } \alpha_i^T \ne \alpha_i^{CDT} \text{ then } \alpha_i^T < \alpha_i^{CDT}.$$

Proof: The claim can be proven in exactly the same manner as described in [Ed 87, p. 300–303] for ordinary Delaunay triangulations. The basic idea is to show that for any triangulation T the following property holds: Consider any two adjacent triangles t_1 and t_2. If one is completely contained in the Delaunay circle of the other then we can get

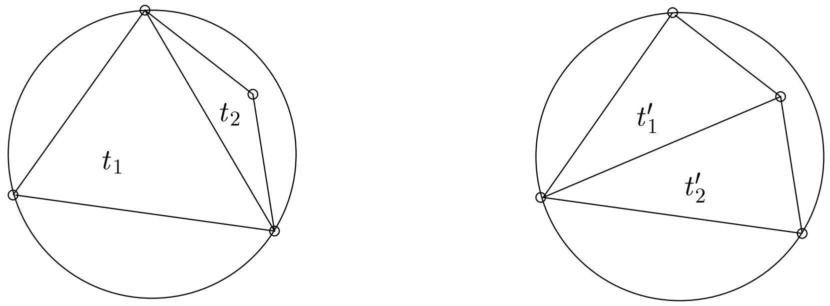

Figure 2: A pair of triangles in T and their counterpart in T'

a lexicographically bigger triangulation T' by switching the diagonals as shown in figure 2. Straightforward elementary geometry can be used to prove this fact. Hence a globally equilateral triangulation can not contain such a pair of triangles.

We now assume that a globally equilateral triangulation T contains a triangle t and a point p in the Delaunay circle of t that is not cut off by a G-edge. Thus a pair (t, p) can be chosen such that the distance $d(t, p)$ is minimal. Let e be the edge of t closest to p. Then the triangle adjacent to e can not contain p because of the above argument and its distance to p is smaller than $d(t, p)$ (see figure 3). This contradicts our choice of t and p.

Since all the triangles of T must satisfy definition 2, T is a constrained Delaunay triangulation.

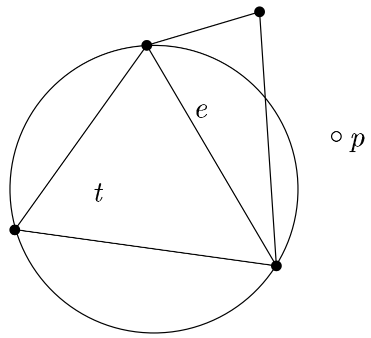

Figure 3: e is not closest to p

3.2 The Triangulation Algorithm

With Theorem 1 we have shown that we can achieve globally equilateral triangular patches by partitioning a polygonal face into a CDT. In recent years several worst-case optimal algorithms to construct a CDT of a graph have been published [Chew 87, WS 87, Sei

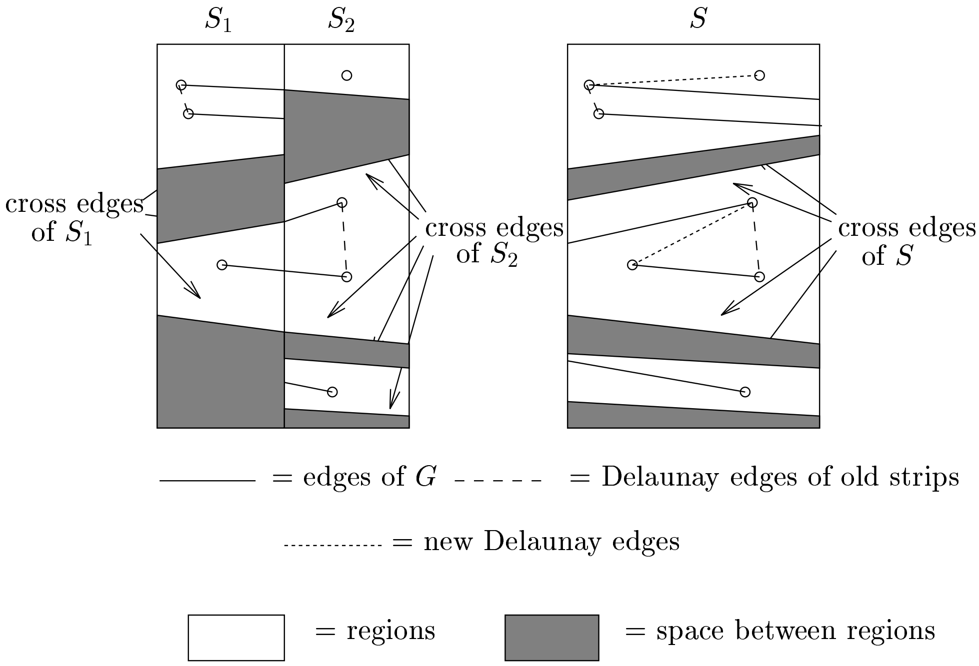

Figure 4: Region of two adjacent strips.

88]. In our implementation we decided to use the algorithm proposed by Chew in 1987. His algorithm is the only one that constructs the CDT directly without computing the Voronoi diagram analog for the constrained case. This makes the algorithm numerically more reliable since newly generated edges only depend on test operations but not on the actual coordinates of additionally computed vertices.

The algorithm is based on a divide and conquer approach. We will only give an outline of the algorithm. For full details please refer to [Chew 87].

We start by dividing the enclosing rectangle of G into vertical strips each containing one vertex. In the general merge step we then assume that we have computed a CDT in two adjacent strips that have then to be pasted together to form a new strip containing twice as many vertices.

One major difficulty in this process is presented by cross edges, i.e. edges of G that cross a strip without having an endpoint in it. It is easily seen that as many as $O(N)$ (N = number of edges) edges can cross one strip. This leads in the worst case to a total of $O(N^2)$ intersection points for all the strips. So instead of keeping track of *all* the cross edges in one strip we only store those *visible* form at least one vertex in the strip (at most 2) which restricts the number of stored cross edges to $2N$. Thus each strip is divided by cross edges into vertex containing regions (see figure 4).

So combining strips falls into two steps:

1. Computing the regions of the new strip and

2. Merging those whose boundaries overlap.

If we assume that we have a partial CDT in two adjacent regions, these CDTs must then be stitched together yielding the CDT of the newly formed region.

It can be shown that only edges that cross the common boundary of the regions have to be added. The core of Chew's algorithm consists of a method to generate the next Delaunay edge above or below a given edge of the CDT crossing the border. Thus once we have found a starting edge e the CDTs can be merged by successively creating all the Delaunay edges above and below e.

The way we can find a starting edge is simply by adding four "infinite" vertices, i.e. vertices that are placed at the locations $(\pm\infty, \pm\infty)$, to each region. If these are treated as if they were ordinary vertices then there always exists an edge between the left and right bottommost infinite vertices that can be used as a start edge.

The whole algorithm runs in time $O(N \log N)$ which is worst case optimal.

4 Implementation

The radiosity approach was implemented on a colour MacIntosh II. As a modeling tool for the creation of scenes we used a simple CAD-system called POLY [ML 87]. It is based on constructive solid geometry (CSG). In this approach objects are generated either as primitives such as cubes, spheres, cylinders etc. or as set unions, intersections or differences of previously generated objects. Thus each object can be described as a tree (CSG-tree) over set operations at the internal nodes and primitives at the leaves.

In order to be able to display the actual geometric form of the object the CSG-tree has to be converted into a second more explicit description of the object. This so-called boundary representation describes the object as a collection of vertices, edges and faces. Since POLY can handle only planar faced objects, all the faces can be regarded as polygons. It is, however, easily possible to generate any multiply-connected polygon as one face of an object.

The triangulation algorithm takes as input one face at a time and transforms it into a sequence of triangular patches. The size of the patches can be determined to be smaller than threshold values for the area and the length of the longest edge of the triangles. If a patch exceeds one value, a refinement procedure is called that either divides the patch into two or four pieces and is then applied recursively to the new pieces. Figure 5 illustrates how the partition is performed. The main aim is to decrease both the area and the length of the longest side by a significant factor without destroying too much of the near equiangularity of the Delaunay triangulation. Often it is desirable to have as large patches as possible since the number of form factors to be computed increases with the square number of patches. In order to get an accurate approximation of the light distribution we then use the same subdivision algorithm to obtain the element substructuring of Cohen et al. mentioned earlier. Now the generated subtriangles represent the elements of a patch. Their size can again be determined by independent threshold values for length and area.

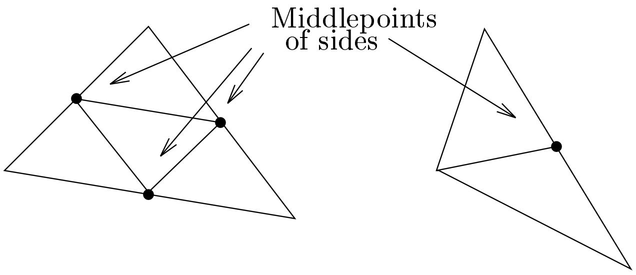

Figure 5: Subdivision scheme for patches and elements

We thus have a parameter driven user independent patch generation procedure that can be used for any object created by POLY.

The triangulation algorithm produces a list of triangles which is then used for the form factor computation. The form factor calculations are performed using the standard hemi cube algorithm with a z-buffer. Each patch is projected onto the hemi-cube of the current element thus yielding the element to patch form factors.

An additional feature is given by the option to perform an adaptive subdivision, i.e. only those elements are partitioned further whose intensity differs considerably from their neighbors [CGI 86]. This step again is parameter-driven in that the user can define how big the difference between two elements has to be before they are further subdivided. Since the radiosity solution is already known at this stage, we only have to compute the form factors for the newly formed elements. This technique is especially useful to approximate the boundaries between areas of a high intensity difference as, for example, shadows in a brightly lit environment produce.

After the solution of equation (3) with the Gauss-Seidel iteration the scene is displayed by a standard z-buffer output module.

5 Conclusion

We have shown that for planar faces the Delaunay triangulation produces patches that are well suited for the radiosity approach. There exist fast algorithms to compute a Delaunay triangulation for a given set of prespecified edges so that even very complex objects can be handled easily.

However, it remains an open question what properties should be required of patches in non-planar surfaces and how these properties can be achieved.

References

[CG 85] M. F. Cohen, D. P. Greenberg. *A Radiosity solution for Complex Environ-*

ments. Proceedings of SIGGRAPH '85, in Computer Graphics, Vol. 19, No. 3, July 1985, p. 31–40.

[CGI 86] M. F. Cohen, D. P. Greenberg, D. S. Immel. *An Efficient Radiosity Approach for Realistic Image Synthesis.* IEEE Computer Graphics and Applications, Vol. 6, No. 2, March 1986, p. 26–35.

[Chew 87] L. P. Chew. *Constrained Delaunay Triangulations.* Proceedings of the Annual Conference on Computational Geometry, ACM 1987, p. 215–222.

[Ed 87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1987.

[GTGB 84] C. M. Goral, K. E. Torrance, D. P. Greenberg, B. Battaile. *Modeling the Interaction of Light Between Diffuse Surfaces.* Computer Graphics, Vol. 18, No. 3, July 1984, p. 213–222.

[ICG 86] D. S. Immel, M. F. Cohen, D. P. Greenberg. *A Radiosity Method for Non-Diffuse Environments.* Proceedings of SIGGRAPH '86, in Computer Graphics, Vol. 20, No. 4, August 1986, p. 133–142.

[Kai 86] J. T. Kaijya. *The Rendering Equation.* Proceedings of SIGGRAPH '86, in Computer Graphics, Vol. 20, No. 4, August 1986, p. 143–150.

[LL 86] D. T. Lee , A. K. Lin. *Generalized Delaunay triangulation for planar graphs.* Discrete & Computational Geometry, 1 (1986), p. 201–217.

[ML 87] A. Meier, H. Loacker. *POLY Computergeometrie für Informatiker und Ingenieure.* McGrawHill, Hamburg 1987.

[NN 85] T. Nishita, E. Nakamae. *Continuous Tone Representation of Three-Dimensional Objects Taking into Account of Shadows and Interreflection.* Proceedings of SIGGRAPH '85, in Computer Graphics, Vol. 19, No. 3, July 1985, p. 22–30.

[PS 85] F. P. Shamos, M. I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag, New York, 1985.

[Sei 88] R. Seidel. *Constrained Delaunay Triangulations and Voronoi Diagrams with Obstacles.* 1978–1988 Ten Years IIG, Institutes for Information Processing, Graz, Report 260, June 1988.

[SPL 88] Min-Zhi Shao, Qun-Sheng Peng, You-Dong Liang. *A New Radiosity Approach by Procedural Refinements for Realistic Image Synthesis.* Proceedings of SIGGRAPH '88, in Computer Graphics, Vol. 22, No. 4, August 1988, p. 22–30.

[Wh 80] T. Whitted. *An Improved Illumination Model for Shaded Display.* CACM 23, 1980, p. 343–349.

[WCG 87] J. R. Wallace, M. F. Cohen, D. P. Greenberg. *A Two-Pass Solution to the Rendering Equation: A Synthesis of Ray Tracing and Radiosity Methods.* Proceedings of SIGGRAPH '87, in Computer Graphics, Vol. 21, No. 4, July 1987, p. 311–320.

[WS 87] C. A. Wang, L. Schubert. *An Optimal Algorithm for constructing the Delaunay triangulation of a set of line segments.* Proceedings of the Annual Conference on Computational Geometry, ACM 1987, p. 223–232.