Radiosity for Highly Tessellated Models

Leonardo Spanò and Enrico Gobbetti
CRS4
Visualization and Virtual Reality Group
VI Strada Ovest, Z.I. Macchiareddu, C.P. 94, I-09010 Uta (CA), Italy
WWW: `http://www.crs4.it/vvr` – *E-mail:* `{leonardo,gobbetti}@crs4.it`

Abstract

The radiosity method is one of the methods of choice used in global illumination simulation. It is a finite element technique that is particularly well suited for computing the radiance distribution in an environment exhibiting only diffuse reflection and emission. We discuss a multiresolution implementation of the technique, that has been developed to rapidly compute radiosity solutions for scenes composed of highly tessellated models. The application context is an interactive lighting design tool being developed in the framework of the DIVERCITY project (EU–IST–13365), funded under the European IST programme (Information Society Technologies).

1 Background and motivation

The radiosity method is one of the methods of choice used in global illumination simulation. It is a finite element technique that is particularly well suited for computing the radiance distribution in an environment exhibiting only diffuse reflection and emission. As for all finite element techniques, its performance heavily depends on the complexity of the input mesh.

The most successful radiosity technique for dealing with complex scenes is currently hierarchical radiosity [11]. The algorithm constructs a hierarchical representation of the form factor matrix by adaptively subdividing planar patches into sub-patches according to a user-supplied error bound. By treating interactions between distant patches at a coarser level than those between nearby patches, the algorithm reduces the cost from quadratic to linear in the number of sub-patches used. However, since an initial transport link has to be computed from each of the original patches to all others, the cost is also quadratic in the number of input polygons, which is the major bottleneck for highly tessellated scenes, in which the geometric complexity is much larger than the illumination complexity. Volume clustering methods [13, 11, 6] combat this problem by grouping input patches into volume clusters. While volume clustering avoids the initial quadratic transport link step, handling the light incident on a cluster is a difficult problem and all presented solutions are more suitable to handling unorganized sets of polygons rather than highly tessellated models [15, 8, 9]. In particular, it is typically extremely difficult to obtain continuously shaded surfaces, since interpolating scalar irradiances across volumes does not lead to good results because of the varying orientations of surfaces within the cluster [8]. At the same time, pushing irradiances to leaves on-the-fly [12, 13, 3], makes it difficult to construct higher order representations of polygon irradiances, makes the method complexity dependent on input model size, and drastically reduces the memory locality of the solution phase.

A number of authors have recognized the mesh simplification techniques for handling large tessellated surfaces in radiosity [10, 7, 4]. The most advanced solution is possibly the face cluster radiosity approach introduced by Willmott and Heckbert [15]. It is a hierarchical radiosity algorithm that operates on face clusters and focuses on vector irradiance rather than radiosity. Since vector irradiance conserves directional information, the push-to-leaves phase is avoided, and the method memory and time complexity are made independent from the input mesh complexity. The method is limited to handling a single irradiance vector per cluster, which leads to "blocky" solutions or fine subdivisions. As for volume clusters, the classic smoothing post-pass is difficult to apply, and re-evaluating visibility at the input polygon level is prohibitively expensive for highly tessellated scenes. For this reason, Willmott [14] proposes a final post-processing stage in which irradiance vectors are recomputed at the corners of each node throughout the hierarchy and interpolated at each input model vertex for computing radiosity. Our work improves over this method by using higher order bases during the solution, leading to better error control and reduced refinement.

2 Methods and tools

In this paper, we briefly illustrate a higher order extension of the face cluster radiosity technique. It combines face clustering, multiresolution visibility, vector radiosity, and higher order bases with a modified progressive shooting iteration to rapidly produce visually continuous solutions with limited memory requirements. In particular, since the method focuses on smoothly representing vector irradiance rather than radiosity, its memory and time complexity are practically independent from the input model size. The output of the method is a vector irradiance map that partitions input models into areas where global illumination is well approximated using the selected basis.

2.1 Hierarchical data structure

As in the original face cluster radiosity algorithm[15], highly tessellated geometric models are represented with a face cluster hierarchy that has the original model polygons as leaves. Each cluster in the hierarchy groups a set of connected faces and behaves like a geometric object on its own, answering queries regarding its geometry (e.g. bounding volume, normal, total area, projected area) and attributes (e.g. reflectance, emission). Currently, each face cluster is represented by an oriented bounding box with the local z axis aligned with the area averaged normal of the contained surface and the x and y axis assigned by a rotating caliper algorithm that minimizes the box volume. Hierarchy construction is done in a preprocessing step on an object by object basis using a greedy algorithm based on the method of Garland et al. [5] that we have extended to handle vertex attributes as in our earlier simplification tool [2].

2.2 Higher-order Vector Radiosity Approximation

The radiosity distribution $b(\mathbf{x})$ in an environment composed only of Lambertian diffuse reflectors and emitters is described by the following integral equation:

$$b(\mathbf{x}) = e(\mathbf{x}) + \rho(\mathbf{x}) \int_A E(\mathbf{x}, \mathbf{y}) dA_y \tag{1}$$

where $e(\mathbf{x})$is the diffuse emittance at point $\mathbf{x}$, $\rho(\mathbf{x})$is the diffuse reflectance at point $\mathbf{x}$, $E(\mathbf{x}, \mathbf{y})$ is the irradiance at point $\mathbf{x}$ due to the light emitted at point $\mathbf{y}$, and the integral is over the surface A of all objects of the environment.

The face cluster radiosity method approximates equation 1 by discretizing the environment into face clusters A_j and by assuming, when computing energy transfer, that all points j within an emitting cluster are close together and far from the receiver [15].

The irradiance vector at a point $\mathbf{x}$ can thus be approximated by

$$\mathbf{E}_{\mathbf{x}} = \sum_j \int_{Aj} \mathbf{m}(\mathbf{x}, \mathbf{y}) b(\mathbf{y}) dA_{\mathbf{y}} \tag{2}$$

and equation 1 thus becomes:

$$b(\mathbf{x}) = e(\mathbf{x}) + \rho(\mathbf{x}) \mathbf{n}_{\mathbf{x}} \cdot \mathbf{E}_{\mathbf{x}} \tag{3}$$

The derivation of a higher-order finite element method for solving this equation follows closely that of the standard scalar radiosity[16, 1]. This equation can be solved approximately by assuming that the radiosity $b(\mathbf{x})$ on patch i can be well approximated by a linear combination $b(\mathbf{x}) = \sum_{i,\alpha} b_{i,\alpha} \Phi_{i,\alpha}(\mathbf{x})$ of a set of non-overlapping orthogonal basis functions $\Phi_{i,\alpha}$ defined on patch i. With this approximation, equation 3 becomes:

$$\begin{aligned} \mathbf{E}_{\mathbf{x}} &\approx \sum_{j,\beta} b_{j,\beta} \left(\int_{A_j} \mathbf{m}(\mathbf{x}, \mathbf{y}) \Phi_{j,\beta}(\mathbf{y}) dA_{\mathbf{y}} \right) && (4) \\ \sum_{i,\alpha} b_{i,\alpha} \Phi_{i,\alpha}(\mathbf{x}) &\approx \sum_{i,\alpha} e_{i,\alpha} \Phi_{i,\alpha}(\mathbf{x}) + \rho(\mathbf{x}) \mathbf{n}_{\mathbf{x}} \cdot \mathbf{E}_{\mathbf{x}} && (5) \end{aligned}$$

Following the Galerkin approach, we take the inner product of the left and right side of this equation with each basis function $\Phi_{i,\alpha'}$, obtaining a set of linear equations from which to compute the unknown irradiances and radiosities:

$$\begin{aligned} \mathbf{K}_{i,\alpha;j,\beta} &= \frac{\int_{A_i} \Phi_{i,\alpha}(\mathbf{x}) \int_{A_j} \mathbf{m}(\mathbf{x}, \mathbf{y}) \Phi_{j,\beta}(\mathbf{y}) dA_{\mathbf{y}} dA_{\mathbf{x}}}{\int_{A_i} \Phi_{i,\alpha}(\mathbf{x})^2 dA_{\mathbf{x}}} && (6) \\ \mathbf{E}_{i,\alpha} &= \sum_{j,\beta} \mathbf{K}_{i,\alpha;j,\beta} b_{j,\beta} && (7) \\ b_{i,\alpha} &= e_{i,\alpha} + \rho_i \mathbf{n}_i \cdot \mathbf{E}_{i,\alpha} && (8) \end{aligned}$$

where ρ_i is the average reflectance of patch i; $\mathbf{n}_i$ is the average normal of patch i and

$$\mathbf{m}(\mathbf{x}, \mathbf{y}) = vis(\mathbf{x}, \mathbf{y}) \frac{(-\mathbf{r}_{\mathbf{xy}} \cdot \mathbf{n}_{\mathbf{y}})_+}{\pi \|\mathbf{r}_{\mathbf{xy}}\|^4} \mathbf{r}_{\mathbf{xy}}$$

These equations revert to the scalar Galerkin radiosity equations in case of perfectly planar elements, and revert to the Willmott's face cluster radiosity equations when using constant bases for both irradiance and radiosity. A hierarchical method for solving these equations is presented in the following section.

2.3 A practical solution method

The following design decisions led to our algorithm:

- avoiding to push irradiances to the leafs is possible using a different hierarchy decoupled from the geometric detail;

- higher order elements make subdivision depths shorter during the solution stage;

- memory needs are further reduced by using different basis for radiosities and irradiance vectors. We employ lower order basis for radiosity (used for emitted light only) and higher order basis for irradiances (used for local illumination detail);

- more storage is saved solving the problem with a shooting approach, thus avoiding to keep the memory intensive coupling coefficients (i.e. a whole transport vector) between two patches. The shooting algorithm avoids the storage of links. By carefully ordering energy exchanges, irradiance is accumulated into a temporary vector and stored only at the leafs of the solution hierarchy.

In our method, each solution element i stores the current unshot radiosity $\Delta B_{i,\alpha}$, the next iteration's unshot radiosity $\Delta B'_{i,\alpha}$, and a list of potential shooters, i.e., the elements that are candidates for transfering light to the element during the current iteration. The algorithm is structured in a way that the vector irradiance $\mathbf{E}_{i,\alpha}$ needs only be stored at the leafs of the solution hierarchy.

At the beginning of the algorithm, a top level solution element is created for each of the top-level face clusters of the scene, with unshot radiosity initialized to the emittance, next iteration unshot radiosity initialized to zero, and an empty list of potential shooters. Multiple instances of the same model are possible. In that case, multiple top-level solution elements would reference the same face-cluster.

At each iteration step, the algorithm starts by initializing each of the top level elements' list of potential shooters with the other top-level elements that have a positive unshot radiosity and are facing towards the potential receiver. The list of potential shooters is then used in the multiresolution light transport phase. In this phase, the hierarchy of each of the top-level solution elements is traversed top-down to transport light from the potential shooters to the receivers. At each element i in the hierarchy, the unshot vector irradiance $\Delta\mathbf{E}_i$ is computed by summing the unshot vector irradiance of the parent with the unshot vector irradiance coming from the potential shooters list. The algorithm cyclically extracts a potential shooter j from the list until the list becomes empty. The coupling coefficients $\mathbf{K}_{i,\alpha\,j,\beta}$ and the error $\delta_{i,,j}$ are computed. If the accuracy of the light transport is considered acceptable, the unshot vector irradiance $\Delta\mathbf{E}_i$ is incremented by $\sum_{j,\beta}\mathbf{K}_{i,\alpha;j,\beta}\Delta B_{j,\beta}$. Otherwise, the algorithm decides to compute the transport at a finer resolution. If the emitter is selected for refinement, the sub-elements of the emitter that are facing towards the receiver are inserted into the receiver's potential shooter list and will be treated later during the same iteration. Otherwise, the emitter is inserted into the list of potential shooters of the receiver's sub-elements that are facing towards it and will be treated later during the top-down element traversal. Self-link refinement is handled similarly by updating the potential shooters lists of the sub-elements in case of subdivision. When the potential shooters list is exhausted, $\Delta\mathbf{E}_i$ contains the unshot vector irradiance of the enviroment that is transfered directly to element i or at coarser level in the solution hierarchy. If element i is a leaf, the vector irradiance $\mathbf{E}_{i,\alpha}$ is incremented by $\Delta\mathbf{E}_i$ and the next iteration's unshot radiosity $\Delta B'_{i,\alpha}$ is set to $(1-F_{i,i})\rho_i\mathbf{n}_i\cdot\Delta\mathbf{E}_{i,\alpha}$. Otherwise, light transport is recursively applied to the sub-elements, and the next iteration's unshot radiosity $\Delta B'_{i,\alpha}$ is computed by pulling the unshot radiosity of the sub-elements.

At the end of each iteration, the current ΔB values are set to those collected into $\Delta B'$, and $\Delta B'$ is cleared. The algorithm terminates when the (infinite) norm of ΔB falls below a user-defined threshold.

3 Implementation and Results

An experimental software library and a radiosity renderer application supporting the hierarchical higher order face cluster radiosity algorithm described in this paper has been implemented and tested on Linux, Silicon Graphics IRIX and Windows NT machines. The software supports combinations of constant, linear, bilinear, quadratic, and cubic bases for representing radiosity and vector irradiance functions. We have implemented both a gathering solver based on the Jacobi iteration and the linkless shooting solver discussed in this paper.

The preliminary results presented here were obtained on a Dell Inspiron 8100 laptop with a Pentium III 1.13GHz and 512 MB RAM running Linux (kernel 2.4). We plan to expand this section in the updated version of this report.

As in Willmott [14], a multiresolution model is stored using a face cluster table, a triangle table (with three vertex indices per triangle), and a vertex table with three coordinates per entry. Materials are stored at the level of clusters in the form of minimum, maximum, and area averaged emittance and reflectance. Face clusters and triangles are sorted to permit direct sequential access.

Using our current implementation, that does not employ particular compression schemes, the memory required for a face cluster node is 110 bytes, while a triangle and a vertex require 12 bytes each using 32 bits integer and floating point values. The memory required for a clustered geometric model of N faces is thus, assuming $2N$ clusters and $N/2$ vertices, of about $238N$ bytes. Only the parts of the model that participate to the solution will need to be swapped into core memory.

Using our shooting algorithm, a solution element has to store a push-pull matrix, two unshot radiosities and the references to the two subelements and to the associated face cluster. Vector irradiances are stored only at the leaf elements. The size of a solution element is thus $12 + 24N_b + 4N_e^2$ bytes for an internal element and $12 + 24N_b + 4N_e^2 + 36N_e$ for a leaf element, where N_b is the number of radiosity coefficients per element and N_e is the number of irradiance coefficients per element. The typical combinations we select are:

- a constant basis for radiosity and a linear basis for vector irradiance (3 coefficients); this combination requires 60 bytes for an internal node and 168 bytes for a leaf node;
- a linear basis for radiosity and a quadratic basis for for vector irradiance (6 coefficients); this combination requires 168 bytes for an internal node and 384 bytes for a leaf node.

In the example presented here, global illumination is computed for a scene containing a highly tessellated object (the Cyberware Venus head, 100K triangles), positioned near three flat colored walls and illuminated by an area light source. Preprocessing time takes 23 s, and the memory required for the geometric model is about 24MB. The preprocessing time can be amortized over multiple renderings. Moreover, since the solution hierarchy is separate from the model hierarchy, multiple instances of the same model may be referenced in the same scene.

Figure 1 presents two solutions computed using constant bases for both irradiance vectors and radiosity, which corresponds to the original face cluster radiosity algorithm. Both images where produced with four shooting iterations. The left image has a link error theshold of 0.001 times the power of the emitter, while the left image has a link error threshold ten times smaller. The rendering time for the left image was 8 s, and the number of leaf elements in the solution is 2403. The higher quality rendering took 120 s and

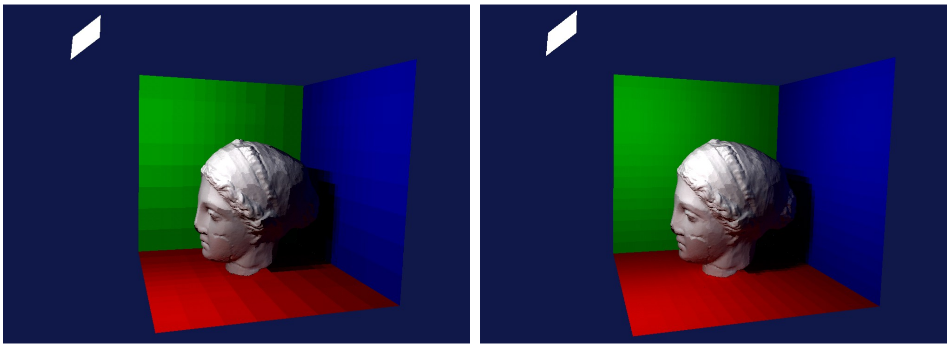

Figure 1: Renderings of the venus with constant radiosity basis and constant irradiance basis, using two different link errors.

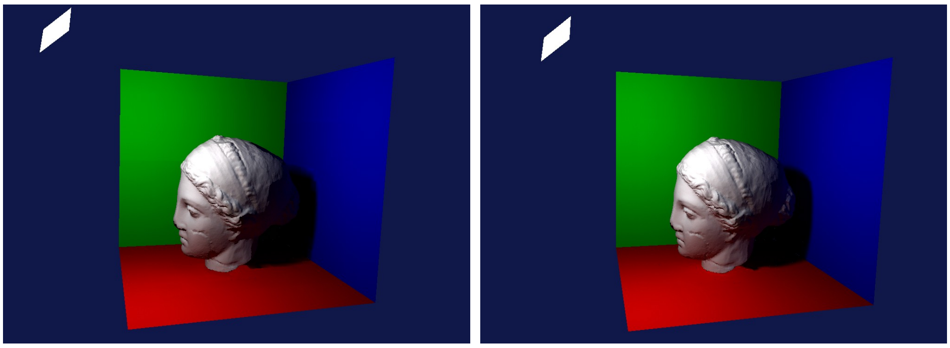

Figure 2: Rendering of the venus with constant radiosity basis and linear irradiance basis, using two different link errors.

produced 16676 leaf elements. Storage costs for the solution hierarchy range from 165KB to 1.54Mb. While the number of elements is sensibly smaller than the the number of input polygons, fine illumination effects are clearly visible. Blocking effects are however clearly visible even in the higher quality image.

Figure 2 presents a solution computed using constant bases for the radiosity, but linear bases for irradiance vectors, using similar renderer settings. The left rendering took 8 s and produced 2370 leaf elements, while the right rendering took 122 s and produced 16654 leaf elements. Storage costs for the solution hierarchy range from 545KB to 3.8Mb. Rendering times are similar to the previous ones, since they are dominated by visibility computations, that use the same cubature rules. However, both solutions are clearly smoother than the higher quality solution using constant bases.

4 Conclusions and Future Work

We have briefly discussed an algorithm for simulating diffuse interreflection in scenes composed of highly tessellated objects. The method is a higher order extension of the face cluster radiosity technique. It combines face clustering, multiresolution visibility, vector radiosity, and higher order bases with a modified progressive shooting iteration to rapidly produce visually continuous solutions with limited memory requirements. The output of the method is a vector irradiance map that partitions input models in areas where global illumination has a good approximation using the selected irradiance basis.

The application context is an interactive lighting design tool being developed in the framework of the DIVERCITY project (EU–IST–13365), funded under the European IST programme (Information Society Technologies).

We are currently exploring the usage of OpenGL register combiners extension to render illuminated models directly from the vector irradiance map, exploiting hardware acceleration for computing vertex radiosity on commodity graphics boards.

Acknowledgments

The authors would like to thank Cyberware for making the Venus model available to the research community.

This research is partially supported by the DIVERCITY project (EU-IST-13365), funded under the European IST programme (Information Society Technologies). We also acknowledge the contribution of Sardinian regional authorities.

References

[1] Phillipe Bekaert and Yves D. Willems. HIRAD: A hierarchical higher order radiosity code. In Werner Purgathofer, editor, *12th Spring Conference on Computer Graphics*, pages 213–227. Comenius University, Bratislava, Slovakia, June 1996. ISBN 80-223-1032-8.

[2] E. Bouvier and E. Gobbetti. TOM – Totally Ordered Mesh: a multiresolution structure for time-critical graphics applications. *International Journal of Image and Graphics*, 1(1):115–134, January 2001.

[3] Per H. Christensen, Dani Lischinski, Eric J. Stollnitz, and David H. Salesin. Clustering for glossy global illumination. *ACM Transactions on Graphics*, 16(1):3–33, January 1997. ISSN 0730-0301.

[4] Reynald Dumont and Kadi Bouatouch. Combining hierarchical radiosity with LODs. In *Proc. IASTED International Conference on Graphics and Imaging*, November 2000.

[5] Michael Garland, Andrew Willmott, and Paul S. Heckbert. Hierarchical face clustering on polygonal surfaces. In *ACM Symposium on Interactive 3D Graphics*, pages 49–58, 2001.

[6] S. Gibson and R. J. Hubbold. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Computer Graphics Forum*, 15(5):297–310, 1996. ISSN 0167-7055.

[7] Gene Greger, Peter Shirley, Philip M. Hubbard, and Donald P. Greenberg. The irradiance volume. *IEEE Computer Graphics and Applications*, 18(2):32–43, March/April 1998.

[8] J. M. Hasenfratz, C. Damez, F. Sillion, and G. Drettakis. A practical analysis of clustering strategies for hierarchical radiosity. In *Computer Graphics Forum (Proc. Eurographics '99)*, pages C–221–C–232, September 1999.

[9] Gordon Müller, Stephan Schäfer, and W. Dieter Fellner. Automatic creation of object hierarchies for radiosity clustering. In David Duke, Sabine Coquillart, and Toby Howard, editors, *Computer Graphics Forum*, volume 19(4), pages 213–221. Eurographics Association, 2000.

[10] Holly E. Rushmeier, Charles Patterson, and Aravindan Veerasamy. Geometric simplification for indirect illumination calculations. In *Proc. Graphics Interface '93*, pages 227–236, Toronto, Ontario, May 1993. Canadian Inf. Proc. Soc.

[11] Francois Sillion, George Drettakis, and Cyril Soler. A Clustering Algorithm for Radiance Calculation in General Environments. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 196–205, New York, NY, 1995. Springer-Verlag.

[12] Francois X. Sillion. A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters. *IEEE Transactions on Visualization and Computer Graphics*, 1(3):240–254, September 1995.

[13] Brian Smits, James Arvo, and Donald Greenberg. A clustering algorithm for radiosity in complex environments. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 435–442. ACM Press, 1994.

[14] Andrew J. Willmott. *Hierarchical Radiosity with Multiresolution Meshes*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, November 2000. Available from http://reports-archive.adm.cs.cmu.edu/anon/2000/abstracts/00-166.html.

[15] Andrew J. Willmott, Paul S. Heckbert, and Michael Garland. Face cluster radiosity. In Dani Lischinski and Greg Ward Larson, editors, *Rendering Techniques '99*, Eurographics, pages 293–304. Springer-Verlag Wien New York, 1999.

[16] Harold R. Zatz. Galerkin radiosity: A higher order solution method for global illumination. In James T. Kajiya, editor, *SIGGRAPH 93 Conference Proceedings*, Computer Graphics Proceedings, Annual Conference Series, pages 213–220. ACM SIGGRAPH, ACM Press, August 1993.