Reconstructing Illumination Functions with Selected Discontinuities

David Salesin Dani Lischinski Tony DeRose*

Program of Computer Graphics
Cornell University
Ithaca, New York 14853

**Department of Computer Science and Engineering*
University of Washington
Seattle, Washington 98195

Abstract

Typical illumination functions contain boundaries that are discontinuous in intensity or derivative. These discontinuities arise from contact between surfaces, and from the penumbra and umbra boundaries of shadows cast by area light sources. In this paper, we present an algorithm that allows for smooth (C^1) reconstruction of intensity everywhere across a surface except along selected edges of intensity or derivative discontinuity. The reconstruction algorithm is based on a piecewise-cubic scattered data interpolation method originally proposed by Clough and Tocher. Our results show marked improvement over piecewise linear or C^1 quadratic reconstructions of some simple illumination functions.

1 Introduction

One way to render a shaded surface is to evaluate an illumination function at every pixel and display the result. However, for illumination functions that vary smoothly across a surface, such a method is unnecessarily expensive. For smoothly-varying functions, it is much more efficient to *sample* the illumination function at various points across the surface, and then *reconstruct* a smooth function to approximate the true function everywhere.

This reconstruction problem arises in several different contexts. For example, in rendering a smooth curved surface, the surface is often approximated by a polygonal mesh. The illumination function is then evaluated at the vertices of the mesh and the resulting intensities interpolated across each polygon. As another example, radiosity algorithms typically discretize each surface (planar or curved) into a mesh of polygonal elements, and radiosity values are computed for each element. These radiosity values are then interpolated over the original surface.

The simplest reconstruction technique is constant shading—each polygon is rendered with a single color. However, this approach introduces positional discontinuities into the reconstructed function. Unless the polygonal elements are extremely small (e.g., less than a pixel in size), these discontinuities will result in a faceted appearance.

For this reason, a linear interpolation scheme, as introduced by Wylie *et al.* [20] and Gouraud [8], is often used instead. However, even the linear reconstruction is prone to

many objectionable artifacts [7, 9]. In particular, although a linear reconstruction can guarantee continuity of intensities across element boundaries, it nevertheless introduces derivative discontinuities where the original function is smooth. These discontinuities may appear as noticeable Mach bands in the reconstructed image.

Higher-order interpolation schemes have also been suggested; for example, Kirk and Voorhies [12] describe a hardware implementation of quadratic interpolation. However, even such higher-order schemes can give rise to Mach bands, particularly for large surface elements, as they do nothing to ensure slope continuity across element boundaries.

In order to ensure slope continuity between elements, Reichert [18] uses a reconstruction technique that is C^1 (continuously differentiable) everywhere across a surface. This method is based on an approach originally described by Powell and Sabin [16] and further developed by Cendes and Wong [2]. To approximate a smooth function, the scheme fits six quadratic Bézier subtriangles over each triangular element. (Max [15] also uses Powell-Sabin interpolation for normals and other values used in the illumination function, although in his work the function itself is re-evaluated at every pixel.)

A C^1 reconstruction works well if the illumination function is continuous and continuously differentiable everywhere. However, in many situations this is not actually the case. For example, typical illumination functions contain boundaries that are actually C^{-1} (discontinuous in intensity) or C^0 (continuous in intensity, but discontinuous in derivative) [1, 10, 13, 14]. Discontinuities in intensity arise from shadows cast by point light sources, as well as from contact between surfaces. Discontinuities in derivative occur at penumbra and umbra boundaries of shadows cast by area light sources. Several algorithms have recently been proposed for constructing meshes that contain these discontinuities as boundaries between elements [11, 14]. Ideally, these C^{-1} and C^0 boundaries should be preserved in the reconstructed illumination function—rather than artificially smoothed over as part of a C^1 function.

Although the problem of constructing surfaces with prescribed discontinuities has been addressed in fields such as computer vision using global, physically-based methods (cf. Terzopoulos [19]), to our knowledge no efficient local method has yet been developed. Moreover, there appears to be no simple way of extending the quadratic Powell-Sabin scheme to handle derivative discontinuities: the scheme imposes a tight coupling between adjacent elements that is difficult to break up in a selective fashion.

We propose a cubic reconstruction scheme, based on a scattered data interpolation method originally proposed by Clough and Tocher [3] and later described by Farin [4]. The scheme breaks up each triangular element into three cubic Bézier subtriangles. Selected discontinuities are easily introduced by relaxing the constraints on the control points of adjacent triangles in a simple and natural way.

In the rest of this paper, we give a formal description of the reconstruction problem (Section 2), describe the algorithm in detail (Section 3), and present results of some simple tests that demonstrate the benefits of this method (Section 4).

2 Problem formulation

By assigning intensities to the vertices of a planar triangulation, we can formulate the reconstruction problem as one of finding an interpolating surface lying in the three-dimensional space defined by the two-dimensional vertex positions and one-dimensional intensities. This surface can be thought of as a "height field" of intensity lying above the planar triangulation. In order to find a good interpolant, we also require a normal vector to this surface at each vertex as part of the input.

Actually, since illumination functions are generally expressed as three-dimensional quantities themselves (e.g., with red, green, and blue components), we typically solve three instances of the problem—one for each color component. However, for clarity, in the rest of the presentation we will treat the illumination function as a unidimensional quantity.

Here, then, is a formal statement of the problem we wish to solve:

Given: A planar triangulation with vertex set V, edge set E, and triangular face set T, along with:

- an intensity z_v and normal vector N_v for every vertex v of every triangle of T; together, (v, z_v, N_v) define a tangent plane t_v;
- a continuity flag $C_{uv} \in [C^{-1}, C^0, C^1]$ for every edge uv of E, such that if (u, v, w) and (u', v', w') are two triangles of T that share edge uv (with $u'v'$ incident to uv) then
  - $C_{uv} \geq C^0$ implies $z_u = z_{u'}$ and $z_v = z_{v'}$;
  - $C_{uv} = C^1$ implies $N_u = N_{u'}$ and $N_v = N_{v'}$.

Find: A function $S : \Re^2 \rightarrow \Re$ such that:

- for every vertex v of every triangle of T,
  - $S(v) = z_v$;
- for every vertex v of V,
  - S is C^{-1} at v if any edge in E with endpoint v has continuity flag C^{-1};
  - otherwise, S is C^0 at v if three or more edges of E with endpoint v have continuity flag C^0, or if there are exactly two such edges and they are linearly dependent;
  - otherwise, S is C^1 at v;
- for every open edge uv of E, the function S is C_{uv} across uv;
- S is C^1 everywhere else.

The careful reader may note what appears at first to be a surprising condition in the output: the function S is C^1 at any vertex with two incident linearly independent C^0 edges. This condition in the output is in fact a very general one, as the following theorem suggests:

Theorem: *Given three points in the plane u, v, w and any piecewise C^1 function $S : \Re^2 \rightarrow \Re$ that is C^0 everywhere and C^1 in the closed region on either side of the broken line u, v, w, then S has a unique tangent plane at v whenever u, v, and w are not collinear.*

Figure 1 A C^0 function S that is C^1 away from u, v, w

Proof: Let S_1 denote the restriction of S to one side of u, v, w, and let S_2 denote the restriction of S to the other side (Figure 1). By hypothesis, S_1 and S_2 are C^1 functions that meet with C^0 continuity on the broken line u, v, w.

Let $\overrightarrow{uv} = v - u$ be the vector from u to v, and let $\overrightarrow{vw} = w - v$ be the vector from v to w. Let $dG(x, \vec{y})$ be the directional derivative of function G at point x in the direction $\vec{y}$.

The hypothesis that S_1 and S_2 meet C^0 on u, v, w implies that for all t in $[0, 1]$,

$$S_1(u + t \cdot \overrightarrow{uv}) = S_2(u + t \cdot \overrightarrow{uv}), \quad (1)$$

$$S_1(v + t \cdot \overrightarrow{vw}) = S_2(v + t \cdot \overrightarrow{vw}). \quad (2)$$

Differentiating (1) with respect to $\overrightarrow{uv}$ and evaluating at $t = 1$ gives

$$dS_1(v, \overrightarrow{uv}) = dS_2(v, \overrightarrow{uv}), \quad (3)$$

and differentiating (2) with respect to $\overrightarrow{vw}$ and evaluating at $t = 0$ gives

$$dS_1(v, \overrightarrow{vw}) = dS_2(v, \overrightarrow{vw}). \quad (4)$$

If u, v, and w are not collinear, then $\overrightarrow{uv}$ and $\overrightarrow{vw}$ are linearly independent, implying that the tangent plane of S_1 at v is the plane through $S_1(v)$ spanned by the vectors $dS_1(v, \overrightarrow{uv})$ and $dS_1(v, \overrightarrow{vw})$. Similarly, equations (1), (3), and (4) imply that the same plane is also tangent to S_2 at v. Thus, the functions S_1 and S_2 share a common tangent plane at v, so S is C^1 at v. □

Thus, any C^1 piecewise polynomial interpolant will yield a surface that is C^1 at a vertex with two incident linearly independent C^0 edges. Note that if the actual illumination function is also piecewise C^1, then it too must exhibit this same behavior. This is the case, at least, for certain simple radiosity functions in the presence of occlusion, as we note in Section 4.

3 Algorithm

Our description of the algorithm is given in three parts:

Figure 2 The Bézier ordinates for a cubic polynomial

1) a summary of the basic Clough-Tocher interpolant, which ensures a C^1 surface everywhere;

2) a description of the modifications necessary in order to handle selected edges with positional and derivative discontinuities;

3) an elaboration of the resulting algorithm.

3.1 The Clough-Tocher interpolant

The following description is rather terse. For a good overview of Bézier surfaces in geometric design, see the excellent book by Farin [6].

A cubic Bernstein-Bézier polynomial P defined over a triangle (u, v, w) is given by the equation

$$P(\beta_u, \beta_v, \beta_w) \;=\; \sum_{\substack{0 \le i,j,k \le 3 \\ i+j+k=3}} \frac{3!}{i!j!k!}\, \beta_u^i \beta_v^j \beta_w^k\, b_{ijk}\ ,$$

where $\beta_u, \beta_v, \beta_w$ are the barycentric coordinates with respect to the domain triangle, and the scalar values b_{ijk} are called the *Bézier ordinates* of P.

For convenience, we will use a notation inspired by Ramshaw's work on blossoming [17] and denote the 10 Bézier ordinates as uuu, vvv, www, uuv, uvv, uuw, uww, vvw, vww, and uvw; the correspondence between these labels and the more conventional b_{ijk} is illustrated in Figure 2.

Note that the ordering in the labels is unimportant; for example, $uvw = uwv$. Note also that each ordinate corresponds to a particular point in the domain triangle; for example, the ordinate uuu corresponds to the point u, the ordinate uuv corresponds to the point $\frac{1}{3}(2u + v)$, and the ordinate uvw corresponds to the point $\frac{1}{3}(u + v + w)$.

Figure 3 The Clough-Tocher construction

For the Clough-Tocher construction, each triangle is split at the centroid into three subtriangles, and a cubic Bernstein-Bézier polynomial is defined for each subtriangle. In order to ensure C^1 continuity everywhere on the interpolating surface, the construction imposes the following constraints:

1) The ordinate vvv for each vertex v is set to z_v.

2) The ordinate uvv for each ordered edge uv must lie in t_v.

3) The points $(uuv, uvv, cuv, c'uv)$ must lie in the same plane, for each edge uv whose adjoining triangles have centroids c and c'.

4) The quadruples $(cuu, uuu, uuv, uuw), (ccu, cuu, cuv, cuw)$, and (ccc, ccu, ccv, ccw) must each lie in the same plane, for each ordered triangle (u, v, w) with centroid c.

The last two constraints are depicted in Figure 3: To ensure C^1 continuity, the four vertices of each of the shaded quadrilaterals must be coplanar.

3.2 Introducing discontinuities

The construction above guarantees C^1 continuity everywhere. Selected discontinuities can be introduced as follows.

Suppose that vertices u and v should be C^1, but that edge uv should be C^0. In this case, the coupling between cuv and $c'uv$ is removed by eliminating constraint (3).

Suppose that vertex v should also be C^0. In this case, there is no longer any unique tangent plane t_v, so constraint (2) is removed. Each ordinate vvx for any vertex x in the original triangulation can be set to any arbitrary value. Note, however, that in order to maintain positional continuity, triangles sharing the same edge in the triangulation must still supply the same four ordinates along that edge.

If an edge uv is to be discontinuous in position (C^{-1}), then the ordinates uuv and uvv split into two independent values each—one for each triangle sharing the edge.

Finally, if a vertex v is also to be C^{-1}, then the ordinate vvv splits into a different value for each triangle incident at v.

Note that in every case, constraint (4), which ensures C^1 continuity across the subtriangle boundaries, remains unchanged.

3.3 A discontinuity reconstruction algorithm

We present here an algorithm that considers triangles one at a time, producing a C^1 piecewise Bézier surface over each triangle having the requisite continuity with respect to neighboring surfaces.

For each triangle (u, v, w) with centroid c, the algorithm outputs a set of 19 values uuu, vvv, www, uuv, uuw, uvv, uww, vvw, vww, cuu, cvv, cww, ccu, ccv, ccw, cuv, cuw, cvw, and ccc, which are the ordinates of the three cubic Bézier subtriangles.

Pseudocode for finding the 19 Bézier ordinates for a triangle $\triangle$ is given in Figure 5. In this code, the notation $\langle r, s \rangle$ is used to denote a three-dimensional vector whose first two components are given by the vector r in the plane, and whose third component is given by the scalar s.

The algorithm described here is intended mostly as an existence proof of a simple discontinuity reconstruction algorithm that satisfies the constraints of Sections 3.1 and 3.2. Many other formulations are possible. For example, a more sophisticated algorithm might attempt to use the extra degrees of freedom in choosing the Bézier ordinates to produce the "smoothest possible" interpolant across triangle edges. Since the choices made by our algorithm are not intended to be optimal or definitive, we describe them here only briefly.

Our algorithm works by partitioning the faces belonging to each vertex v into a set of C^1 *wedges*, with each wedge delimited by a pair of edges that are either C^{-1} or C^0 (Figure 4). For each wedge, the algorithm determines a single normal $\widehat{N}_v$, which is used in the computation of constraint (3) in order to ensure C^1 continuity across all faces in the wedge.

Most of the subroutines called are very simple: The routine $InPlane(v, z, N, u)$ returns a scalar giving the height of plane p above point u, where p is the (unique) plane with normal N whose height is z above v. The routine $Avg(N, N')$ returns the (normalized) average of the two normals N and N'. The routine $Project(N, N')$ returns the (normalized) projection of vector N in the plane normal to N'.

Only one of the subroutines is nontrivial, the routine to compute a single normal for a wedge, which is given in Figure 6. The routine begins by calling a subroutine *AverageNormalForWedge*, which computes an average normal for all the faces within a certain wedge. This normal is returned if the vertex v is C^1 (i.e., if there is only a single wedge). Note that a vertex with two linearly independent C^0 edges is actually C^1, as proved in Section 2.

Figure 4 Partitioning the faces about a vertex into a set of wedges

On the other hand, if other wedges about the vertex exist, then it is important to adjust the computed normal $\widehat{N}_v$ so that the partial derivatives along the C^0 edges of the wedge will agree with those of adjacent wedges; otherwise, positional continuity is not guaranteed along these edges. This adjustment is made in step 4 of the pseudocode. The code is complicated somewhat by the need to treat 180° wedges specially in steps 3 and 4. The routine *Collinear*(W) returns true if and only if the two edges bounding wedge W form a 180° angle.

In order to derive a normal for each triangular face, we take an extra radiosity sample at the midpoint of each edge of the triangulation. Together, the three samples along each edge are used to define an interpolating parabola. The tangents to the two parabolas incident at each vertex of a triangular face determine a unique normal for the face.

4 Results

We have implemented the reconstruction algorithm on an HP 720 in C^{++}, as a back-end to the discontinuity-meshing radiosity algorithm described by Lischinski *et al.* [14].

We compared a piecewise linear interpolation (a), a C^1 quadratic (Powell-Sabin) reconstruction (b), and our C^1 cubic reconstruction (c) against a reference solution (d), in which the illumination function was evaluated at every pixel. For each comparison, the input to the different reconstruction methods was a single triangular mesh, with intensities sampled at every vertex and at the midpoint of every edge. The quadratic and cubic reconstructions interpolate the vertex samples and use the midpoint samples only for estimating normals. To make the comparison fair, the linear reconstruction splits each original triangle into four subtriangles and interpolates all the samples.

The first example, shown in Figure 7, shows a simple illumination function of unoccluded light reflecting off a plane. The triangular mesh for this example contained just

FindOrdinates($\triangle$): *Given a triangle* $\triangle$, *compute the 19 cubic Bézier ordinates of the three subtriangles of* $\triangle$.

1. let c be the centroid of $\triangle$

2. for each vertex v of $\triangle$ do
   $vvv \leftarrow z_v$
   $\widehat{N}_v \leftarrow \mathit{WedgeNormal}(\triangle, v)$
   end for

3. for each (unordered) edge uv of $\triangle$ do
   let w be the other vertex of $\triangle$ besides u and v
   $uuv \leftarrow \mathit{InPlane}(u, uuu, \widehat{N}_u, \frac{1}{3}(2u+v))$
   $uvv \leftarrow \mathit{InPlane}(v, vvv, \widehat{N}_v, \frac{1}{3}(2v+u))$
   if $C_{uv} = C^1$ then
      $N \leftarrow \mathit{Avg}(\widehat{N}_u, \widehat{N}_v)$
   else
      $N \leftarrow \mathit{Avg}(N_u, N_v)$
   end if
   $N \leftarrow \mathit{Project}(N, \langle v-u, 3(uvv-uuv)\rangle)$
   $cuv \leftarrow \mathit{InPlane}(\frac{1}{2}(u+v), \frac{1}{2}(uuv+uvv), N, \frac{1}{9}(4u+4v+w))$
   end for

4. for each vertex v of $\triangle$ do
   let u, w be the other two vertices of $\triangle$ besides v
   $cvv \leftarrow \frac{1}{3}(uvv+vvv+vvw)$
   $ccv \leftarrow \frac{1}{3}(cuv+cvv+cvw)$
   end for

5. $ccc \leftarrow \frac{1}{3}(ccu+ccv+ccw)$, where u, v, w are the vertices of $\triangle$

Figure 5 Finding the Bézier ordinates

WedgeNormal$(\triangle, v)$: *Given a triangle* $\triangle$ *and distinguished vertex* v, *return a single normal for the* C^1 *wedge about* v *containing* $\triangle$.

1. let $W = w_{\text{prev}} v w_{\text{next}}$ be the wedge about v containing $\triangle$
   $\widehat{N}_v \leftarrow \mathit{AverageNormalForWedge}(W)$

2. if v is a C^1 vertex then
   return $\widehat{N}_v$
   end if

3. for each adjacent wedge $W' \in \{$ prev, next $\}$ do
   let vw be the edge separating W' from W
   $N \leftarrow \widehat{N}_v$
   if $C_{vw} = C^0$ then
   $N' \leftarrow \mathit{AverageNormalForWedge}(W')$
   switch (*Collinear*(W), *Collinear*(W'))
   case (FALSE, FALSE) : $N \leftarrow \mathit{Avg}(N, N')$
   case (TRUE, TRUE) : $N \leftarrow \mathit{Avg}(N, N')$
   case (FALSE, TRUE) : $N \leftarrow N'$
   case (TRUE, FALSE) : $N \leftarrow N$
   end switch
   end if
   $s_{W'} \leftarrow \mathit{InPlane}(v, z_v, N, \frac{1}{3}(2v + w))$
   end for

4. if *Collinear*(W) then
   $\widehat{N}_v \leftarrow \mathit{Project}(N, \langle w_{\text{next}} - w_{\text{prev}}, 3(s_{\text{next}} - s_{\text{prev}})\rangle)$
   else
   $\widehat{N}_v \leftarrow \langle w_{\text{next}} - v, 3(s_{\text{next}} - z_v)\rangle \times \langle w_{\text{prev}} - v, 3(s_{\text{prev}} - z_v)\rangle$
   end if

5. return $\widehat{N}_v$

Figure 6 Computing a single normal for a wedge

4 triangles. Note the Mach band artifacts in (a), which are eliminated in (b) and (c).

The second example, shown in Figure 8, shows the reconstruction of an illumination function from a partially occluded area light source, reflecting off a plane. In this case, the occluding object is a small square. The triangular mesh for this example contained 58 triangles. All discontinuities on the receiving surface are represented as appropriately-flagged edges of the triangulation. Again, note the incorrect Mach bands created by the reconstructions in (a), which are eliminated in (b) and (c). Also note the *correct* Mach bands, due to the C^0 penumbra edges, that appear in the reference solution (d), and which are preserved in (c). These derivative discontinuities in the actual function cause the C^1 quadratic interpolant (b) to "overshoot" dramatically in order to maintain C^1 continuity across the reconstruction.

Finally, observe how in the reference solution (d), the Mach bands along the penumbra edges begin to disappear in the vicinity of the penumbra corners. These corners are defined by two non-collinear C^0 boundaries of a piecewise C^1 illumination function. Therefore, by the theorem in Section 2, the intensity function at these corners must have a unique tangent plane, which accounts for the smoothing of the discontinuity edge in their vicinity. Note that since the actual illumination function is piecewise C^1, it is amenable to a good reconstruction by a piecewise polynomial interpolant, as demonstrated in (c).

5 Further work

We have presented a C^1 reconstruction algorithm for illumination functions that allows for selected positional and derivative discontinuities. Such discontinuities appear commonly in scenes with abutting objects and occlusions.

There are many aspects of this algorithm that suggest further research:

Better normals. We have described a simple scheme for estimating intensity normals at each vertex. However, it is possible to imagine other schemes that may work equally well or better. For example, it would be interesting to try using two extra samples, instead of just one, at each edge, and fitting a cubic instead of a quadratic curve for estimating normals. Alternatively, it would also be interesting to try using *no* extra samples, and instead estimate the derivatives directly from the nearby interpolated samples. In addition, it may also be worthwhile to look at more sophisticated schemes for estimating wedge normals, for example, using a least-squares approach.

A more sophisticated interpolant. While the Clough-Tocher method imposes a number of constraints in order to ensure C^1 continuity across the interpolant, there is still considerable flexibility in choosing the Bézier ordinates so that the constraints are satisfied. The algorithm presented here resolves these extra degrees of freedom rather naively, using, for example, component-wise averaging and other simple techniques. Better solutions may be possible. For example, Farin [5] describes a modified Clough-Tocher interpolant that comes as close to C^2 continuity as possible across triangle edges.

Handling more general meshes. Our formulation of the input allows for a C^{-1} vertex v only when at least one edge incident on v is also C^{-1}. However, in actual illumination functions, it is possible to have isolated singularities arising at points of contact between

surfaces [14]. In addition, our formulation is restricted to linear discontinuity boundaries, whereas in actual illumination functions these boundaries can be conics [10]. We would like to extend our algorithm to also handle these cases.

Acknowledgements

We would like to thank Filippo Tampieri for writing a large share of the discontinuity-meshing radiosity software, which served as a front-end for our algorithm, and for helpful comments on the manuscript.

References

[1] Daniel R. Baum, Stephen Mann, Kevin P. Smith, and James M. Winget. Making radiosity usable: automatic preprocessing and meshing techniques for the generation of accurate radiosity solutions. *Computer Graphics* (SIGGRAPH '91) 25(4): 51–60, 1991.

[2] Zoltan Cendes and Steven H. Wong. C^1 quadratic interpolation over arbitrary point sets. *IEEE Computer Graphics and Applications* 7(11): 8–16, 1987.

[3] R. Clough and J. Tocher. Finite element stiffness matrices for analysis of plate bending. *Matrix Methods in Structural Mechanics* (Proceedings of the conference held at Wright-Patterson Air Force Base, Ohio, 26-28 October 1965), 515–45, 1966.

[4] Gerald Farin. Smooth interpolation to scattered 3D data. In *Surfaces in CAGD*, R. E. Barnhill and W. Boehm, editors, North-Holland Publishing Company, Amsterdam, 43–63, 1983.

[5] Gerald Farin. A modified Clough-Tocher interpolant. *Computer Aided Geometric Design* 2: 19–27, 1985.

[6] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design,* Second Edition. Academic Press, Boston, 1990.

[7] James D. Foley, Andries van Dam, Steven K. Feiner, John F. Hughes. *Computer Graphics: Principles and Practice.* Addison-Wesley, Reading, Massachusetts, 739–41, 1990.

[8] Henri Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers* C-20(6): 623–9, 1971.

[9] Eric Haines. Ronchamp: a case study for radiosity. In the SIGGRAPH '91 "Frontiers in Rendering" course notes, §3: 1–28, 1991.

[10] Paul S. Heckbert. *Simulating Global Illumination Using Adaptive Meshing.* PhD thesis, University of California at Berkeley, 1991.

[11] Paul S. Heckbert. Discontinuity meshing for radiosity. To appear in *Proceedings of the Third Eurographics Workshop on Rendering*, Bristol, England, 1992.

[12] David Kirk and Douglas Voorhies. The rendering architecture of the DN10000VS. *Computer Graphics* (SIGGRAPH '90) 24(4): 299–307, 1990.

[13] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. Improving sampling and reconstruction techniques for radiosity. Technical Report 91-1202, Department of Computer Science, Cornell University, Ithaca, New York, 1991.

[14] Dani Lischinski, Filippo Tampieri, and Donald P. Greenberg. A discontinuity meshing algorithm for accurate radiosity. Submitted for publication.

[15] Nelson Max. Smooth appearance for polygonal surfaces. *The Visual Computer* 5: 160–73, 1989.

[16] M. Powell and M. Sabin. Piecewise quadratic approximation on triangles. *ACM Transactions on Mathematical Software*, 316–25, 1977.

[17] Lyle Ramshaw. Blossoming: a connect-the-dots approach to splines. Technical Report 19, DEC Systems Research Center, Palo Alto, California, 1987.

[18] Mark Reichert. *A Two-Pass Radiosity Method Driven by Lights and Viewer Position.* Masters thesis, Cornell University, 1992.

[19] Demetri Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8: 413–24, 1986.

[20] C. Wylie, G. W. Romney, D. C. Evans, and A. C. Erdahl. Halftone perspective drawings by computer. *Fall Joint Computer Conference '67*, Thompson Books, Washington, DC, 49–58, 1967.

(a) (b)

(c) (d)

Figure 7 Reflection of unoccluded light, reconstructed with (a) piecewise-linear, (b) C^1 quadratic, and (c) the C^1 cubic interpolant described in this paper. A reference solution is shown in (d).

(a) (b)

(c) (d)

Figure 8 Reflection of a partially-occluded light source, reconstructed with (a) piecewise-linear, (b) C^1 quadratic, and (c) the C^1 cubic interpolant described in this paper. A reference solution is shown in (d).