Radiance Space, as Represented by the Visibility 2-Skeleton

Jared Hoberock, Samuel Hornus, John C. Hart

Radiance Space, as Represented by the Visibility 2-Skeleton

Jared Hoberock[1], Samuel Hornus[2], and John C. Hart[3]

[1] NVIDIA Corp.
[2] INRIA Sophia Antipolis
[3] University of Illinois, Urbana-Champaign

Abstract. Since radiance in a vacuum remains constant along rays, the radiance in a scene's free space can be represented as a real function over the space of maximal segments. This space is four dimensional and organized by the visibility complex. To date only the 0-D and 1-D cells of the visibility complex have been constructed and implemented. In order to use this topological space to support radiance samples for image-based rendering, form-factor computations, global illumination and shadowing, we also need to construct its 2-D cells. This paper expands the frontier of our understanding of the visibility complex by constructing and implementing its 2-D cells. We demonstrate the capabilities of these 2-cells by using them to organize sparse illumination samples in a scene. These samples interpolate only within the boundaries of the 2-cells, thus avoiding aliasing while respecting sharp changes in illumination.

1 Introduction

In a closed 3-D scene, both sightlines and light transport occur along extremal line segments that extend between one surface point to a second surface point (including those on lights and sensors), so the space of these extremal suface-to-surface segments is four-dimensional. This extremal segment space is organized by the *visibility complex,* which partitions it into subspaces whose dimensionality corresponds to the degrees of freedom of the maximal segments they represent. The visibility complex organizes these subspaces as a cell complex, though unlike ordinary cells, the subspaces are not necessarily simply connected.

The goal of this paper is to consider the visibility complex as a support for measuring radiance in a scene. The radiance is constant along a maximal line segment in a vacuum, but unlike visibility it is not symmetric and each maximal line segment supports two radiances, one in each direction. Thus there is a 2:1 map from the space of radiance to the space of visibility, and the "radiance" complex is a double cover of the visibility complex.

Image based modeling and rendering methods use a variety of spaces to organize spaces of radiance samples. The two-plane parameterization stores all of the external radiance passing through a double-paned window in $I^2 \times I^2$ [1,2]. Surface light fields store a hemisphere of internal radiance samples over the surfaces in the scene S in the space $I^2 \times \partial S$ but requires a parameterization of

every surface in the scene. The photon map records the position and incident direction of photons as they hit surfaces in a scene. Extending a segment from each photon in its incident direction yields a directed version of the maximal segment space organized by the visibility complex. Thus the directed segment space organized by radiance complex is equivalent to the completion of a photon map. We thus investigate the visibility complex as a natural topological space for organizing radiance samples.

Though the visibility complex of a 3-D scene is well understood [3], it has never been constructed. Its size is $O(n^4)$ which would be quite large for modern scenes of millions of polygons. The 1-skeleton (nodes, arcs) of the visibility complex has been constructed procedurally for on-demand evaluation of shadow boundary information that can be used for discontinuity meshing [4]. This paper constructs the visibility 2-skeleton which expands the utility of the visibility complex beyond discontinuity meshing to more substantial applications.

Contributions. We show how the 2-cells of the visibility complex can be interrogated by walking the 1-cells around their boundary. To this end, we extend the half-edge representation of the scene mesh into a half-cell representation of the visibility 2-skeleton. This analysis leads to the following contributions.

- **Construction.** A top-down construction of the visibility 2-skeleton that can be further extended to also support higher (3- and 4-) dimensional cells in a more natural setting than previous work [3, 4].
- **Implementation.** An implementation, complete with results and discussion, of the visibility 2-skeleton. Previous implementations have been limited to the full visibility complex of a 2-D scene or the 1-skeleton of a 3-D scene [4]. Durand *et al.* [3] described a construction algorithm for the 3-D complex but did not implement it.
- **Sampling.** As alluded to previously [3], the visibility complex can be used to store radiance, making it a natural data structure for image-based rendering. Our implementation allows us to not only store but organize radiance samples.
- **Interpolation.** The visibility 2-skeleton allows us to interpolate these samples relative to visibility events, which extends Bala *et al.* [5], which was limited to a subset of the visibility 1-skeleton.

2 Previous Work

Data Structures. There have been a variety of data structures devised to organize information about visibility in a scene. A simple quadratic visibility graph [6] connects mutually visible vertices of a scene with edges, but ignores visual events involving faces and edges. The aspect graph partitions space into viewpoints from which hidden-line renderings are isomorphic [7, 8], but an n-polygon scene leads to worst-case $O(n^6)$ orthographic and $O(n^9)$ perspective aspect graphs. The visibility complex [9, 10] organizes visual events into a more tractable data structure which is worst-case $O(n^4)$ but probabilistically $O(n^2.67)$ [3] which makes it attractive for exact visibility computations.

Discontinuity Sensitive Reconstruction. To accelerate global illumination algorithms, much research has been spent on sparse sampling and reconstruction. Early methods reconstructed images by interpolating samples without regard to discontinuities such as the edges of objects and shadows [11]. More recent approaches account for discontinuities explicitly in the reconstruction process [5]. The color at a pixel is approximated by interpolating only the nearby samples that can be reached without crossing a discontinuity, though until now these boundaries were limited to the visibility 1-skeleton.

Robust Epsilon Visibility. Duguet and Drettakis [12] construct a more robust implementation of the visibility 1-skeleton by merging small visibility features. They thicken vertices into spheres and edges into cylinders by an ϵ factor to avoid small features that create numerical problems and increase the size of the visibility complex. Our work on the 2-skeleton is based on their 1-skeleton approach, and we have used their ϵ-thickening to avoid small features in our construction of the 2-skeleton.

3 Background

Half-Edge Mesh Datastructure. Let $S \subset R^3$ be a closed polygonal manifold scene represented by a half-edge data structure [13]. The directed edge $\mathsf{e} \in S$ emanates from the vertex e.start and terminates at the vertex e.end, whereas its opposite e.opp extends from e.end to e.start. The next edge counterclockwise when viewed from outside the manifold is e.next and these two edges share the face e.left.

Segment Space. The collection of all oriented segments is R^6, the space consisting of all line segments from $x \in R^3$ to $y \in R^3$. The collection of undirected segments is *segment* space, the quotient of R^6 with the equivalence $(x, y) = (y, x)$ which still describes a 6-D space. Maximal free segment space is a quotient space of segment space by identifying a segment with its subset elements in segment space. In a closed scene of 2-D surfaces, elements of maximal free segment space start and end on a surface and so maximal free segment space is four dimensional.

Visibility Complex. The visibility complex $\mathcal{V}(S)$ is a paritioning of the maximal free segment space of a scene S into cells corresponding to the dimensionality of the family of segments they represent. Segments between two faces form 4-cells. Those tangent to a single edge form a 3-cell, since one dimension is reduced by the loss of a degree of freedom due to the edge tangency. Fig. 1 illustrates the four cell and six surrounding 3-cells for the maximal segment space between two triangles. A family of segments tangent to two edges (or a vertex if the two edges meet) form a 2-cell. Those tangent to three edges (or a vertex and an edge) form a 1-cell, and those tangent to four edges, two vertices or a vertex and a pair of edges form a 0-cell. Some of these "cells" can be a manifold-with-boundary, not necessarily homeomorphic to a disk, e.g. some 2-cells have holes. Segment families eventually reach an additional edge, so higher dimensional cells are bounded by lower dimensional cells, which forms the visibility complex $\mathcal{V}(S)$.

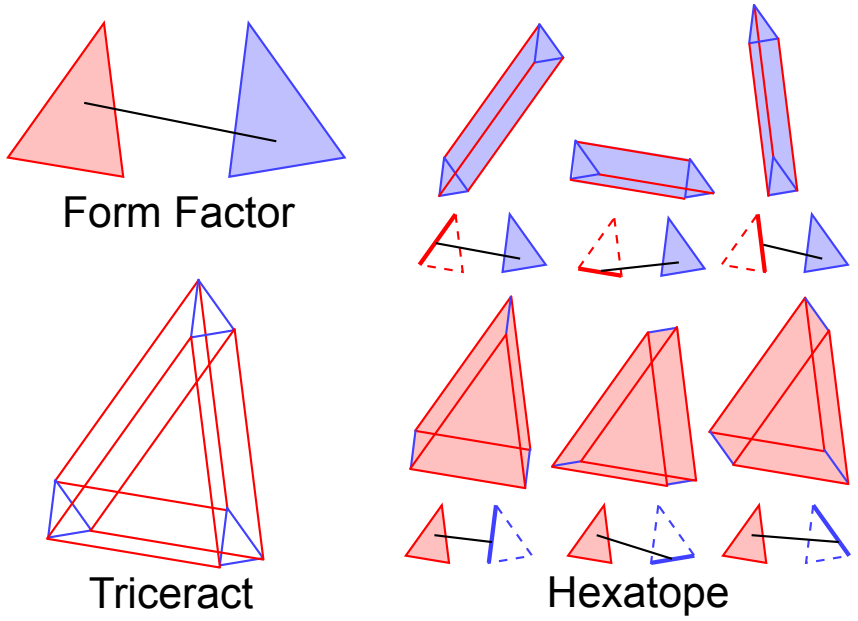

Fig. 1. Form Factor: The form factor between two triangles measures the ratio of foreshortened radiance transported along segments with endpoints on each face. Triceract: The space of maximal segments between two triangles is the 4-D space $\Delta \times \Delta$ consisting of the red triangle whose "points" are in this case smaller copies of the blue triangle. Hexatope: The triceract is also a hexatope, a six-sided 4-D polytope, whose boundary consists of six 3-cells $\Delta \times |$ each containing the family of segments from one triangle's edge to the other's face.

Figure 2 shows a subset of the 2-cells between two triangles. These 2-cells collectively cover the 3-D space occupied by the 4-cell of line segments between the two triangles. Thus any triangle intersecting this 4-cell will intersect the union of its 2-cells. Thus the visibility complex can be constructed and maintained from its 2-skeleton.

4 The Visibility 1-Skeleton

We denote by $\mathcal{V}_k(S)$ the k-skeleton of the visibility complex of scene S containing the subset of cells of dimension k or less. We first construct $\mathcal{V}_0(S)$ with an $O(n^4)$ iteration over all sets of four edges. Each set of four edges can yield zero, one or two E4 cells [14]. These sets of four edges may contain one or two pairs that share endpoints, yielding VEE- and VV-cells respectively.

We construct the 1-skeleton $\mathcal{V}_1(S)$ of the visibility complex following the methods of Durand *et al.* [4], and Duguet *et al.* [12], but with the addition of oriented VE-cells computed on a half-edge mesh data structure that eases the

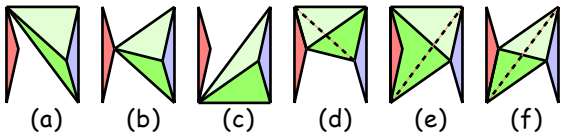

Fig. 2. A subset of the eighteen 2-cells between two triangles (red, blue), including three (a-c) of the nine V-cells and three (d-f) of the nine EE-cells.

implementation of the topological reasoning needed to connect the 1-cells to each other through the existing 0-cells.

4.1 VE Half-Cells

Each VE-cell corresponds to the set of maximal free segments passing through a vertex v and a maximal subset of an edge e. We first define a canonical VE half-cell as

$$\mathsf{halfve} = (\mathsf{upface}, \mathsf{v}, \mathsf{e}, \mathsf{downface}, \mathsf{start}, \mathsf{end}). \tag{1}$$

The upface and downface indicate the pair of faces the family of segments extend between, and the half-cell is oriented such that the downstream direction is *from* the vertex *to* the edge.

Using the values indicated in Fig. 3 we represent a VE cell by the two half-VE-cells

$$\mathsf{ve} = (\mathsf{up.left}, \mathsf{v}, \mathsf{e}, \mathsf{e.left}, \mathsf{start}, \mathsf{end}), \tag{2}$$

$$\mathsf{ve.opp} = (\mathsf{up.left}, \mathsf{v}, \mathsf{e.opp}, \mathsf{down.left}, \mathsf{end}, \mathsf{start}), \tag{3}$$

where edges up and down are projected from e in the ve case, but projected from e.opp in the ve.opp case.

Each VE half-cell has a blocker pair corresponding to the two faces that terminate the family of segments. We represent the blocker with a projected edge. For example the edge up is a projection of e through the vertex v onto the "up" blocker face. (This projection reverses its direction). In general position, this projection is embedded in the middle of a face, and up.left = up.right return a pointer to this face. In degenerate cases, such as highly aligned architectural models, the edge e can project onto the boundary of (the closure of) one of the blocker faces, in which case up.left $\neq$ up.right, which occurs often in highly aligned (e.g. architectural) scenes.

We also need to handle self-occlusion when v and e share the same face. We indicate this special case by setting the down member of the ve half-cell to the vertex v. The mesh supporting vertex v might contain the entire "up" side of the VE cell, in which case the vertex v serves as the up member of both ve and ve.opp.

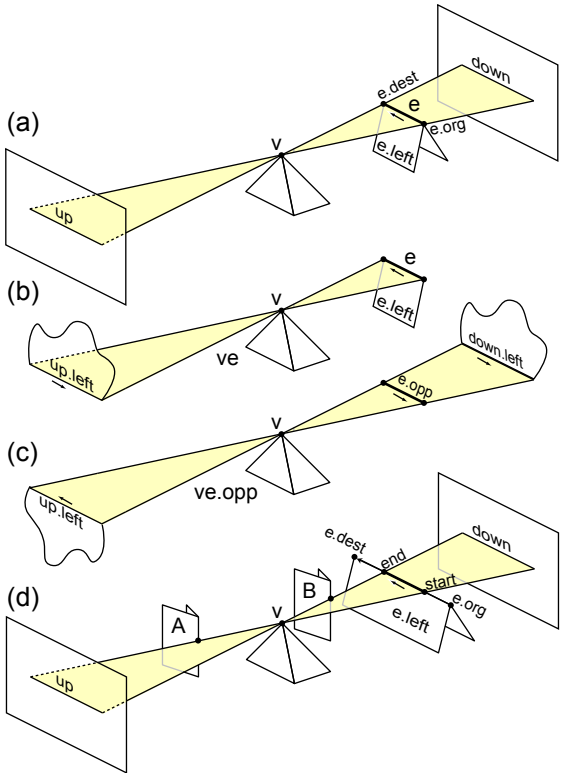

Fig. 3. A simple VE cell (a) representing the one-cell of maximal segments passing through vertex v and oriented edge e between blockers containing e's projections up and down. This VE cell is oriented and represented by the half-VE cell (b) and its opposite (c). A more complicated version (d) trimmed by occluders A and B demonstrates the need for also recording start and end edge endpoints.

For efficiency, we assume edge e is a silhouette edge, such that e.left faces toward v but e.right faces away. We can also handle non-silhouette edges, where v can see both e.left and e.right, by setting down = e in the ve.opp half-cell. In general including non-silhouette edges leads to a larger visibility complex than necessary.

4.2 Orienting EEE-Cells

One-cells describe a 1-parameter family of lines and the direction of parameterization is useful when determining their topological organization in the visibility complex (e.g. to which 0-cells they attach). Whereas the VE cell is parameterized by its single edge, the E^3 cell, given as $\{e_1, e_2, e_3\}$, can be parameterized by any of its three edges.

We use the expression $\mathrm{sgn}(d\mathsf{e}_i/d\mathsf{e}_j|_{\{e_i,e_j,e_k\}})$ to determine the parametric orientation of an EEE cell, where the derivative $d\mathsf{e}_i/d\mathsf{e}_j$ represents the change in the position of the stabbing line along edge i that occurs in response to a change in the position of the stabbing line along edge j. These derivatives are constant for a given EEE cell and can be computed exactly using forward differences.

5 The Visibility 2-Skeleton

The visibility complex contains two kinds of 2-D elements. A V cell represents the space of all maximal free segments stabbing a vertex between the same two faces, and resembles a double prism. An EE cell represents the space of all maximal free segments stabbing two edges, which forms a tetrahedron between the two edges, along with the extensions of the segments beyond the edges.

5.1 V Cells

We represent a V cell by the 4-tuple

$$\mathsf{vcell} = (\mathsf{upface}, \mathsf{v}, \mathsf{downface}, \{\partial_i\}) \tag{4}$$

where upface and downface are the two face blockers (in either order) and $\{\partial_i\}$ are a list of the outer ∂_0 and possibly inner $\partial_{i\geq 1}$ boundaries of the V cell, each represented as closed loops of VE-cells. Since there is only a vertex and two blockers, there is no need to orient, though if v is one of the blockers, then it is always the "up" blocker. Some sample V cells appear in Figure 4.

Figure 4(a) demonstrates that a V cell may not be simply connected. It consists of the yellow tri-cone less the middle blue tri-cone. It may contain one or more holes each created by an occluder between v and one of its blockers that projects to the interior of the blocker. There does not exist a path of visual events which connect the inner boundary to the outer boundary of the VE cell, which is one reason the interrogation of 2-cells has been elusive.

V-Cell Boundaries The VE cells can be linked into a chain based on the connectivity of the oriented half-edge segments. In other words, the end of one VE cell will correspond to the start of the next VE cell in a chain. These chains eventually form closed loops, such that the end of the tail VE cell corresponds to the start of the head VE cell. In general position, these VE-cell chains are simple, but in degenerate cases these chains can merge and divide at 0-cells.

Given a VE half-cell and its corresponding blocker pair (upBlocker, downBlocker) we form a boundary loop that wraps around this blocker pair. For example, the hexagonal V cell in Fig. 4(c) consists of six ve half-cells of type (2), though their "downstream" direction from vertex to edge alternates. The V cell corresponding to the top of the rear triangle in Fig. 4(b) consists of one ve.opp and two ve half-cells. Thus we interrogate V-cells in $\mathcal{V}_2$ by finding chain boundary loops

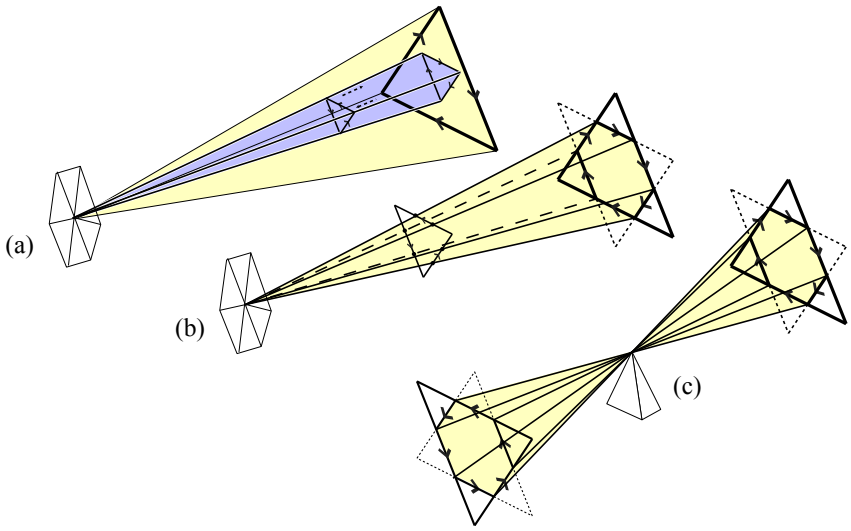

Fig. 4. Examples of V cells. A V cell is formed from the vertex to the small triangle (a), which creates a hole in the second V cell on the larger triangle. The V cell for the smaller triangle in (b) creates three V cells on the larger triangle. The V cell between two triangles (c) is a projected hexagon.

of VE half-cells which is very similar to finding faces in a manifold mesh by traversing half-edges.

Since we know a VE cell's opposite exists, we store only a single half-VE-cell and discard its opposite. If we find a half-VE-cell with the wrong orientation, we reconstruct its opposite on the fly.

As shown in Fig. 4(a), a V-cell can contain a hole, which means the family of lines from a vertex through a blocker pair can completely surround another V-cell. In general, the ve half-cells (2) wrap counterclockwise, constructively surrounding the V-cell, whereas the ve.opp half-cells (3) wrap clockwise, destructively trimming the V-cell.

V Cell Enumeration The V cells of the visibility complex are interrogated by the algorithm described in Fig. 5. This algorithm begins by creating a set VE containing all VE half-cells generated by vertex v. We then pick each blocker pair in turn and seek to establish a V cell for it. The variable $VEb1b2$ contains all of the VE half-cells that contain $b1$ and $b2$ as blockers, in either order.

The procedure FormLoops($VEb1b2$) picks a random VE half-cell from $VEb1b2$ to initiate traversal. This traversal forms a chain by finding a VE half-cell in $VEb1b2$ whose start point corresponds to the previous cell's end point, until a loop is formed. The process repeats until no more VE half cells remain in $VEb1b2$, and a list of loops is returned.

VE = FindVECellsWithVertex(v)
Foreach $\{b1, b2\} \in$ AllBlockerPairs(VE)
    $VEb1b2$ = FindVECellsWithBlockers($VE, \{b1, b2\}$)
    L = FormLoops($VEb1b2$)
    $\{\partial_i\}$ = OuterInnerBoundaries(L)
    $V = V + (b1, v, b2, \{\partial_i\})$
Endfor

Fig. 5. V Cell enumeration algorithm.

If more than one boundary loop is returned, they must be identified as inner or outer. We assume a closed scene such that one of the blockers is a polygon (e.g. no $\{\infty, v, \infty\}$ V cells allowed) and project the boundary loop onto the plane P of one of the blockers. Outer boundary loops wind counterclockwise whereas inner boundary loops wind clockwise about the stabbing direction, where winding is computed as the sum of the turning angles.

Inner-Outer Chain Matching Once chains have been created and classified, we must decide which inner-chain holes belong in which outer boundaries. We cannot infer this from cell topology since there is no link from a hole to its outer boundary. We must use geometry: it is a non-convex polygon inside non-convex polygon problem.

The procedure *Find-Holes* takes a single outer boundary ∂ and a list of possible holes (inner chains) H, and returns the subset list of inner chains $\overline{\Delta} \subset H$ corresponding to the interior of the region surrounded by the outer boundary ∂. We project the chains onto **P,** a plane through one blocker of the V cell. Using the Jordan curve theorem, we cast a plane-ray from an inner-chain vertex and count its intersections with the outer boundary. An odd number indicates that the inner chain is inside the outer chain.

5.2 Refinement of Sparse Samples

Robust Epsilon V Cells General position ensures that the *next* operation at a 0-cell is unambiguous, but degenerate, aligned scenes can cause problems. The extremal stabbing line in Fig. 6 generated by $(v, v2)$, also stabs edge $e5$. This yeilds two VE cells originating at $vv2e5$ winding counter-clockwise to any V cell of the form $\{upBlocker, v, f2\}$.

Even for a degenerate 0-cell, its number of terminating VE cells must equal its number of originating VE cells. We resolve the 0-cell degeneracy with an arbitrary but robust choice that maps a unique previous VE cell terminating at a 0-cell to a unique next VE cell emanating from the 0-cell. Our *next* operation collects all the next VE cells and all the previous VE cells into two lists, N and P, respectively. We sort the lists by a unique ID per VE cell (e.g. pointer value).

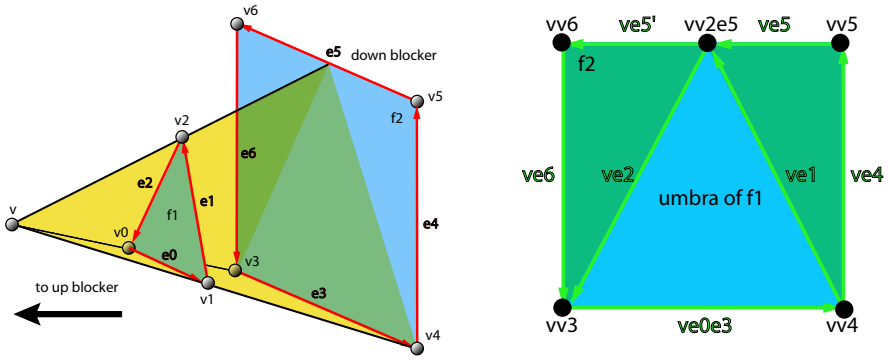

Fig. 6. A degenerate scene (left) where the 0-cell vv2e5 stabs two edges. The view through v (right). Note that vv2e5 has two "next" cells: ve5' and ve2.

We use this arbitrary ordering to create a bijection from previous VE cells to next VE cells. This mapping requires that the next operation at a 0-cell can access the previous VE cell visited.

5.3 EE-cells

We enumerate EE-cells similarly, once we clarify the the notion of being locally to the right or to the left of the EE-cell, and the notion of winding around an EE-cell. We first parameterize the EE-cell, consisting of segments tangent to two edges e_1 and e_2 ordered arbitrarily but consistently. We parameterize each edge e_i with $x_i \in [0, 1]$ and thus the EE-cell with $(x_1, x_2) \in [0, 1]^2$. The image of the boundary of this parametric domain consists of hyperbola arcs [15]. We can thus define left, right and winding with respect to this parametric domain, as shown in Fig. 7.

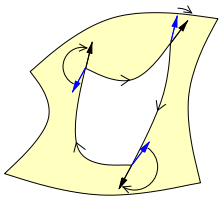

Fig. 7. An EE cell is bounded by hyperbolic arcs. Its winding is computed by the difference between the arc's tangent vector (black) and the tangent vector of the previous arc connecting to it (blue).

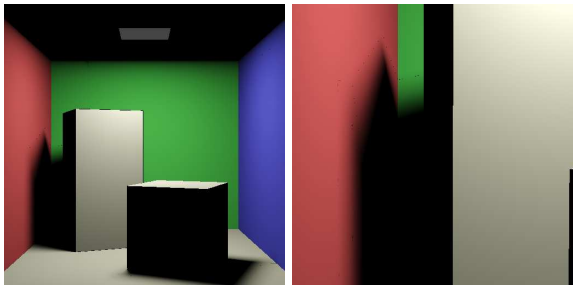

Fig. 8. Ray-traced Cornell box with exact direct illumination from an area light (left) and zoom (right). Each rendering took 27 minutes for a 512x512 image, unoptimized.

6 Results

We demonstrate the visibility 2-skeleton with a pair of rendering applications, as a support for the exact evaluation of area lighting and to manage discontinuities in the reconstruction of illumination from sparse samples.

6.1 Exact Penumbrae

The visibility 1-skeleton can be used for discontunuity meshing of shadows, to refine a mesh such that its edges are places along penumbra boundaries. The regions between these 1-cell boundaries are spanned by 2-cells that can be used as support for the exact measurement of the penumbrae.

A Monte Carlo ray-tracer approximates the direct illumination at a point P by stochastically sampling the area light with shadow rays. If we compute V_P, not only can we use an analytical solution to compute this direct illumination, but one can also achieve better sampling of the directions around P for integrating indirect illumination in a Monte-Carlo algorithm. The exact direct illumination of P from an area light can be computed since we know the exact parts of the area light that are visible from P [16], as demonstrated in Figure 8.

One applications of V-cell construction is computing the view from a query point. Let P be a point in space to serve either a scene vertex or a view point. All V-cells adjacent to P consitutes the visibility polyhderon V_P of P that encode all scene points visible from P. From V_P, one can extract the set of visible polygons (the set of blockers of V-cells inside V_P), which yields the hidden surface rendering from viewpoint P. Placing P on a light source generates shadow volumes from the visibility 2-cells V_P.

If C is the camera location, then the shape of the V-cells in V_C are the visible parts of the scene. When projected on the image plane, their boundaries indicate radiance discontinuities, and use these boundaries to limit the interpolation of sparse radiance samples. Figure 9 shows the results of repeated refinement of these image regions, where the illumination has been computed only at the vertices the image-space mesh.

7 Conclusion

We have described a half-edge organization for VE cells, and used it to link the 1-cells of the visibility complex to define its 2-cells. We used these 2-cells for the direct evaluation of penumbrae and to organize and interpolate sparse samples of scene illumination. These examples demonstrate the use of the visibility complex as a representation of radiance space, to support the direct integration or the uniform sampling and interpolation of homogeneous regions of scene radiances.

An additional application we did not investigate is the application of the visibility complex 2-cells to photon mapping [17]. The caustic photon map is a photon map localized to specular surfaces to more efficiently detect and sample caustic effects. It is usually set up manually based on scene composition. The set of V-cells V_P around a sample point P on an area light can aid in choosing a direction for each caustic photon and in evenly distributing photons in directions where specular objects lie. This process can be further refined by restricting V_P to only the V-cells where on blocker is specular.

The shadow at the bottom left of the larger box In Figure 9 could be further improved by a more advanced eye-light refinement heuristic that computes the sets V_P of each vertex P of a polyhedral area light, together with the sets E_e of EE-cells adjacent to each edge e of that area light. The boundaries of the 2-cells in $\bigcup_{P,e} V_P \cup E_e$, when projected on their blockers, draw the loci of discontinuities in illumination. This resulting disconinuity mesh would then need to be incorporated into the existing visibility complex refinement.

Furthermore, sampling an EE-cell in the $[0,1]^2$ parameterization would indicate important places to sample the scene polygons for reconstruction from sparse samples [5]. This is indeed where penumbra variation is most subtle. Our implementation focused on the more difficult construction of V-cells, and we did not implement the EE-cells needed to detect the loci of penumbrae in this test scene.

Acknowledgments. This work was supported in part by the NSF under the ITR CCR-0219594 and a CNRS-UIUC partnership.

References

1. Levoy, M., Hanrahan, P.M.: Light field rendering. In: Proc. SIGGRAPH. (1996) 31–42
2. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proc. SIGGRAPH. (1996) 43–54
3. Durand, F., Drettakis, G., Puech, C.: The 3d visibility complex. ACM TOG **21**(2) (April 2002) 176–206
4. Durand, F., Drettakis, G., Puech, C.: The visibility skeleton: A powerful and efficient multi-purpose global visibility tool. Proc. SIGGRAPH 97 (Aug. 1997) 89–100
5. Bala, K., Walter, B., Greenberg, D.P.: Combining edges and points for interactive high-quality rendering. In: Proc. SIGGRAPH. (2003) 631–640

6. Welzl, E.: Constructing the visibility graph of n line segments in the plane. Information Processing Letters **20** (1985) 167–171
7. Gigus, Z., Malik, J.: Computing the aspect graph for the line drawings of polyhedral objects. IEEE Trans. PAMI **12**(2) (Feb. 1990)
8. Plantinga, H., Dyer, C.R.: Visibility, occlusion, and the aspect graph. International Journal of Computer Vision **5**(2) (1990) 137–160
9. Pocchiola, M., Vegter, G.: The visibility complex. Intl. J. Comp. Geom. & Appl. **6**(3) (1996) 279–308
10. Durand, F., Drettakis, G., Puech, C.: The 3d visibility complex, a new approach to the problems of accurate visibility. Proc. Eurographics Workshop on Rendering (June 1996) 245–256
11. Walter, B., Drettakis, G., Parker, S.: Interactive rendering using render cache. In: Proc. EGRW. (1999) 19–30
12. Duguet, F., Drettakis, G.: Robust epsilon visibility. In: Proc. SIGGRAPH. (2002) 567–575
13. Guibas, L., Stolfi, J.: Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams. ACM Trans. on Graphics **4**(2) (1985) 74–123
14. Teller, S., Hohmeyer, M.: Determining the lines through four lines. J. Graphics Tools **4**(3) (1999) 11–22
15. McKenna, M., O'Rourke, J.: Arrangements of lines in 3-space: A data structure with applications. In: Proc. Symp. Comp. Geom. (1988) 371–380
16. Sillion, F., Puech, C.: Radiosity and Global Illumination. Morgan Kaufmann (1994)
17. Jensen, H.W.: Global illumination using photon maps. In: Rendering Techniques. (1996) 21–30

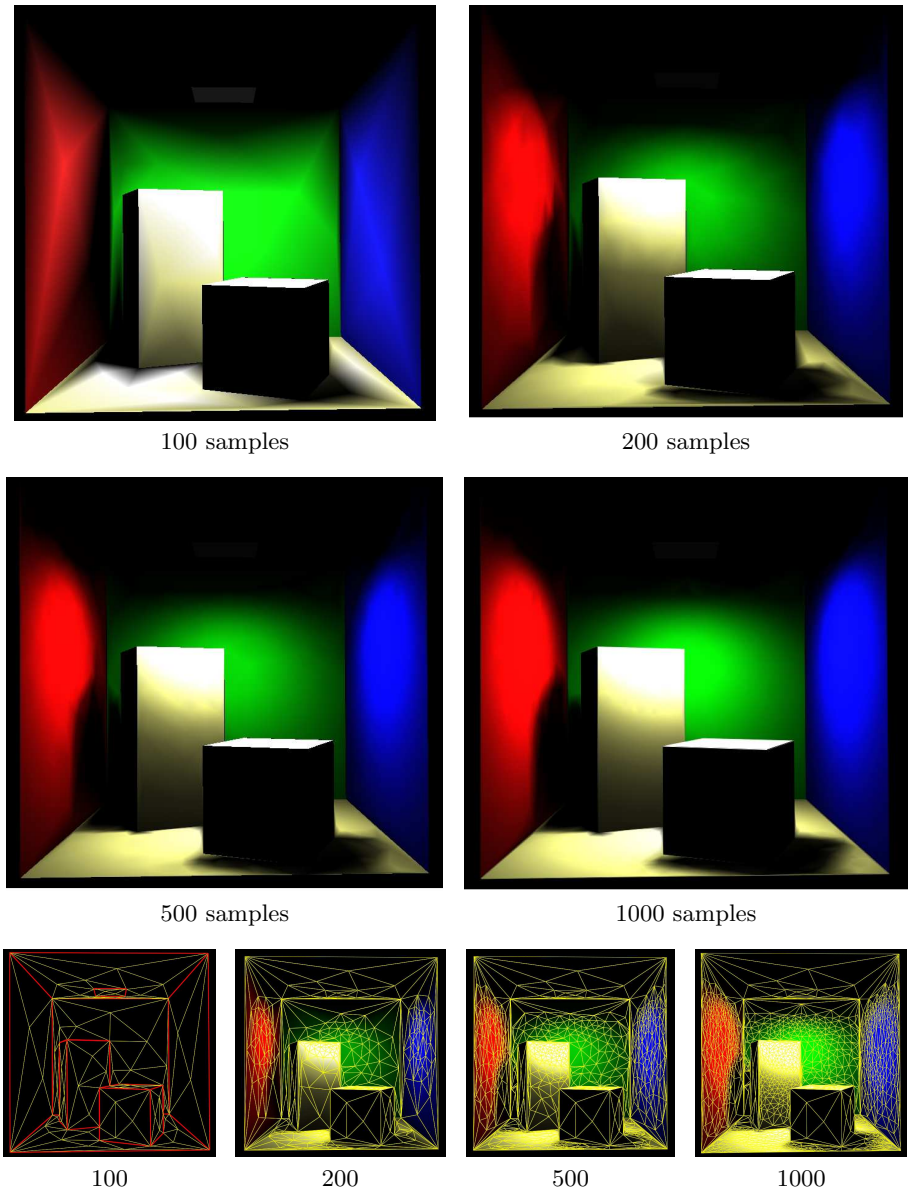

Fig. 9. Illumination reconstructed from a progressively refined image mesh of sparse samples, organized by the 2-cells of the visibility complex.