Using the visibility complex for radiosity computation

Rachel Orti Frédo Durand Stéphane Rivière Claude Puech

iMAGIS [*]/ GRAVIR - IMAG, BP 53, F-38041 Grenoble Cedex 09, France

(email: Rachel.Orti@imag.fr)

Abstract

The radiosity method is particularly suitable for global illumination calculations in static environments. Nonetheless, for applications of image synthesis such as lighting design or architectural simulation, we have to deal with dynamic environments. To make the method usable in a real case, the illumination has to be updated as fast as possible after an object moves. The efficient way is to find the calculations strictly necessary to be recomputed after a change in the scene. The largest part of the computation time is spent on visibility calculation. In this paper, we investigate the possible speed ups in those calculations.
We propose the use of the visibility complex for radiosity calculations. The presented study is realized for 2D scenes of convex objects in the static case. We show that the visibility complex is very suitable for radiosity calculations in this context, and that it also allows for efficient updates in the dynamic case.

Keywords: radiosity, discontinuity meshing, form factor, visibility complex, dynamic environments

1 Introduction

Visibility is a very important topic in computer graphics, and especially in global illumination calculation. Visibility computation requires an efficient method for determining the objects seen from a point in a scene or from an object in the scene. In the case of the radiosity method (a method for the simulation of light interactions between the objects in a scene), the largest part of the computation time is spent calculating the fraction of light leaving one object that arrives at another one, since it involves many visibility calculations. To deal with the increasing complexity of the scenes considered in computer graphics, some research has been performed in visibility processing in order to accelerate visibility determination during illumination calculations [25, 26]. The idea is to build a special data structure that allows for easy selection of a set of potentially visible objects.

Radiosity was initially applied to static environments, currently it starts to deal with moving objects. The work done so far still involves too many recomputations [1, 3, 7, 23], even if only a single object moves in a scene. No method has yet been proposed which exactly identifies what has to be strictly recomputed. In computational geometry, a data structure called the visibility complex has recently been developed. It deals with visibility between objects in 2D scenes. We think that this structure will allow to better understand the phenomenon of 2D dynamic environments, a step necessary for the development of efficient algorithms for radiosity in dynamic 3D scenes. Some research in lighting simulation has already been carried out in 2D for discontinuity meshing and radiosity calculations [10, 11] or wavelet radiosity approaches [22, 9]. 2D radiosity is a good laboratory for "real" 3D radiosity. The possibility of analytic computations offers a way to validate the models; the scenes are simpler to understand and allow for a better comprehension of the phenomena. Moreover, many of 3D scenes are in fact "$2\frac{1}{2}$D"scenes, like the inside of a building where the occlusions are mainly caused by the walls, which are equivalent to their projections onto a 2D plan.

In this paper, we present the use of the visibility complex for radiosity computation. We show the high suitability of this structure that helps to accurately find the mutually visible parts for all pairs of objects of the scene. Therefore, we can consider the light interactions between only mutually visible parts of objects. We can also directly compute the ratio of the light exchange between two objects, thanks to the information contained in the complex. The study is presently realized in the static case. However, we show how the visibility complex can be used to considerably improve the computation time in the dynamic case. Making the visibility complex dynamic will make possible the strictly necessary recomputations.

[*] iMAGIS is a joint project of CNRS, INRIA, Institut National Polytechnique de Grenoble and Université Joseph Fourier.

2 The radiosity method

2.1 Principles of the method

Radiosity is a global illumination method used to render a scene of objects by computing the lighting for each point in a scene [24]. This approach is based on the physical principle of heat transfer between surfaces and simulates the balance of light energy between radiating surfaces for illumination computations.

The radiosity method was first applied to computer graphics in 1984 by Goral *et al.* [8]. With this method, the light inter-reflections between surfaces in a closed environment are modeled by a system of equations. Surfaces are assumed to be Lambertian (i.e. to reflect incident light in all directions with equal intensity). To set up the system of equations, the environment is divided into small areas (or patches). The radiosity B_i of a patch P_i is the total rate of light energy leaving this patch and is equal to the sum of emitted and reflected energies. This quantity is to be constant over a patch, and can be expressed, for an environment discretized into n patches, as:

$$B_i = E_i + \rho_i \sum_{j=1}^{n} B_j F_{ij} \ , \qquad (1)$$

where E_i is the light emitted from the patch P_i, ρ_i is the reflectivity of this patch, and F_{ij} is the form factor between the patches P_i and P_j, that is the fraction of energy leaving P_i that arrives at P_j.

The accuracy of the radiosity solution depends on the discretization (or mesh) of the environment. To have good results the mesh should follow the distribution of light in the environment, with a higher density of elements in areas where the illumination changes rapidly [24]. The best method is to use a *discontinuity meshing* that places mesh boundaries on the radiosity discontinuities caused by occlusions [10] (i.e. on shadow limits). This meshing strategy traditionally requires many geometric calculations which make it very expensive to use.

Most of the computation time in the radiosity method is spent on the calculation of the form factors (still more than 50% of the time in the case of efficient algorithms, e.g. [12]) because of the visibility computations. Thus, it is important to have an efficient method to compute form factors, especially in dynamic environments where the geometry of the scene can change and causing the need for some form factors being recomputed. Below we show that the visibility complex offers a way to easily compute both the discontinuity mesh and the form factors between objects for a given scene.

2.2 The form factor

The form factor is the fraction of energy leaving one surface that arrives at another one [24]. The form factor F_{ij} between two surface elements A_i and A_j is defined as:

$$F_{ij} = \frac{\text{Radiative energy reaching } A_j \text{ from } A_i}{\text{Total radiative energy leaving } A_i} \ . \qquad (2)$$

The form factor is expressed in 3D as a double integral over areas which takes into account the visibility between surfaces. In the 2D case, it becomes a double integral over lengths. The form factor between two length elements L_i and L_j corresponds to:

$$F_{ij} = \frac{1}{L_i} \int_{L_i} \int_{L_j} \frac{\cos\phi_i \cos\phi_j}{2r} H_{ij} dL_j dL_i \ , \qquad (3)$$

where H_{ij} is the visibility between the elements dL_i and dL_j (1 if dL_i and dL_j are mutually visible, 0 otherwise) and r the distance between these elements (see Figure 1).

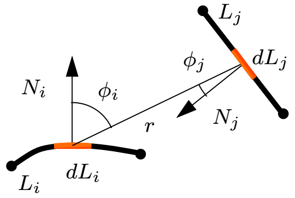


Figure 1: Notation for the form factor.

The form factor is, in fact, a strictly geometric quantity: it depends only on the shape and relative location of surfaces in the scene.

We will consider two other equivalent expressions for the form factor based on this property: the second being valid only in 2D.

• A **ratio of measures of lines**: as the energy is transferred through light rays, the fraction of energy corresponding to the form factor can be formulated as a ratio of measures of sets of lines. Sbert [21] has used this principle to compute an approximation of the form factors between two surfaces in 3D. We consider here the *measure of lines* as defined within the context of Integral Geometry in [20]. Using this measure, the expression for the 2D form factor between two portions of curves C_i and C_j becomes:

$$F_{ij} = \frac{\text{Measure of lines intersecting } C_i \text{ and } C_j}{\text{Measure of lines intersecting } C_i} \ . \qquad (4)$$

• A **weighted sum of curves lengths**: a formula called the "string rule," established in thermal engineering [13], allows for a simple computation of the 2D form factor between two portions of curves C_i and C_j by computing the length of "strings" drawn between the endpoints of C_i and C_j (see Figure 2). The strings stretched from the endpoints of C_i to the corresponding endpoints of C_j (i.e., a to c and b to d) are called *un-crossed strings*, and those drawn to the opposite endpoints on C_j (i.e., a to d and b to c) are called *crossed strings*. The form factor between two portions of curves C_i and C_j (with lengths L_i and L_j respectively) becomes:

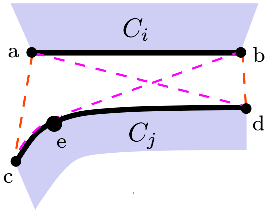


Figure 2: Strings for two portions of curves C_i and C_j.

$$F_{ij} = \frac{\sum \mathcal{L}(crossed\ strings) - \sum \mathcal{L}(un\text{-}crossed\ strings)}{2L_i}, \quad (5)$$

where $\mathcal{L}(c)$ represents the length of the curve c.
This formulation is valid even if some parts of C_i and C_j are not mutually visible. In the case of an obstructed view, the strings are simply imagined to stretch around the obstruction (see the portion of curve between e and c).
For the example given in Figure 2, the form factor F_{ij} between C_i and C_j corresponds to:

$$F_{ij} = \frac{d_e(a,d) + d_e(b,e) + d_c(e,c) - d_e(a,c) - d_e(b,d)}{2L_i},$$

where $d_e(m,n)$ represents the Euclidean distance between the points m and n, and $d_c(m,n)$ the Curvilinear distance between m and n.

3 The visibility complex

3.1 Introduction

We consider a set of n disjoint objects in the plane that represent obstacles to the propagation of either the light or a robot. The visibility graph is an important data structure in this context. For a scene of convex objects in the plane, it is made of the bitangents which do not intersect any object. But this structure is too poor to allow for global visibility queries such as maintaining a view around a point or other global visibility computation.

That is why Pocchiola and Vegter [15] introduced the *visibility complex*, a data structure for visibility of 2D scenes.

3.2 The visibility complex

We limit ourselves to convex objects and add an infinite “blue sky” object to the scene for the sake of coherence. We consider the set of maximal free segments of the scene, that is, segments in free space (interior of the objects removed) of maximal length. The visibility complex is a 2-dimensional cell complex which is a partition of the set of maximal free segments according to their visibility. It is composed of three types of elements (see Figure 3):

- *Vertices* are 0-d components: a vertex corresponds to a segment tangent to two objects and touching two other objects at its endpoints.
- *Edges* are 1-d components: an edge corresponds to the segments tangent to one object and touching two other objects.
- *Faces* are 2-d components: a face is the set of segments that touch the same pair of objects, i.e., that “see” the same objects.

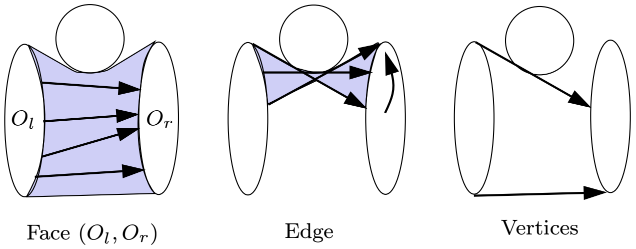


Figure 3: Elements of the visibility complex.

In order to visualize these elements we consider a duality relation which associates a point with a line and vice versa. If we represent the lines in a dual space, for example $-x\sin\theta + y\cos\theta = u \mapsto (\theta, u)$, then a given object has for each θ two tangents, $(\theta, \lambda(\theta))$ and $(\theta, \mu(\theta))$. $\lambda(\theta)$ and $\mu(\theta)$ describe two curves in the dual space. Each line (θ, u) such that $\lambda(\theta) < u < \mu(\theta)$ intersects the object. For a scene of objects, these curves partition the dual space into connected components corresponding to lines intersecting the same objects. This partition is called the *dual arrangement* (see Figure 4).

The maximal free segments can be canonically projected onto the lines, and the edges of the complex are projected onto the curves of tangency, which can help to visualize them (see Figure 5). In the dual space, we notice that:

- An *edge* is delimited by two vertices and incident to three faces.
- A *vertex* is incident to four edges and six faces.
- A *face* is delimited by two chains of vertices and edges.

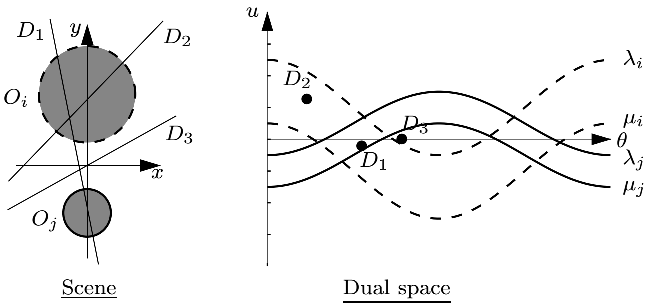


Figure 4: Dual arrangement of a scene and some lines.

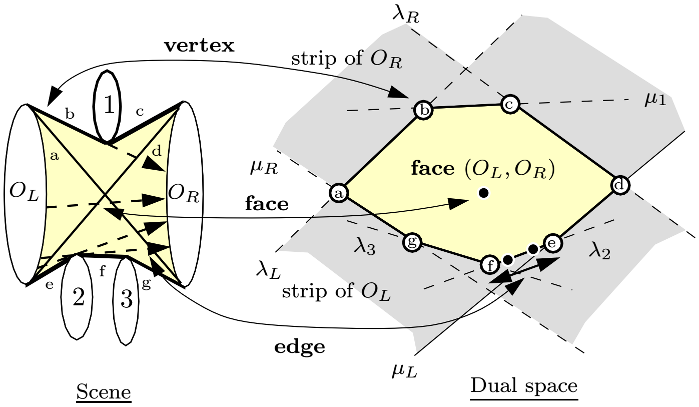


Figure 5: Structure of a face in the scene and its projection onto the dual plane.

The size of the complex is characterized by the number m of vertices which is $\Omega(n)$ and $\mathcal{O}(n^2)$ (with n the number of convex objects).

To have a more complete introduction to the visibility complex, refer to [15], [5] or [4].

3.3 Construction

The construction of the complex can be achieved by a sweep of its vertices. When a vertex is swept, the relations between the edges and the faces have to be updated.

The straight sweep proposed by Pocchiola and Vegter [14], running in $\mathcal{O}(m \log n)$ time is the solution currently implemented (see [19] for implementation details).

Some care has to be taken to handle problems due to degeneracies and numerical errors (see [18]).

An implementable optimal algorithm has been proposed in [16], using a topological sweep. It runs in optimal $\mathcal{O}(m + n \log n)$ time and optimal $\mathcal{O}(n)$ space.

In these algorithms objects were assumed to have $\mathcal{O}(1)$ complexity which can be restrictive. Our current implementation computes the bitangents with a dichotomy method, which takes non negligible time. Rivière [17] has proposed and implemented an algorithm for the construction of the visibility complex of scenes of simple polygons. It runs in optimal $\mathcal{O}(m + n \log n)$ time and $\mathcal{O}(n)$ space, and handle the problems of degeneracies.

4 Using the visibility complex

4.1 Computing the form factors

The form factor between two elements is null if these elements are not mutually visible. In practice, such situations are frequent and should be taken into account in order to avoid unnecessary computations. Thanks to the visibility complex only pairs of mutually visible objects are considered, and only the mutually visible parts of these objects are accurately examined.

When considering the complex in the dual space (θ, u), the computation of the 2D form factor can be re-expressed by:

- A **ratio of areas of regions**:

Proposition 1 *The measure of the set of lines that intersect the object O_i is equal, in the dual space (θ, u), to the area between the two curves of tangency λ_i and μ_i for θ going from 0 to π.*

Proposition 2 *The measure of the set of lines that intersect the objects O_i and O_j is equal, in the dual space (θ, u), to the area of the corresponding face of the visibility complex.*

These propositions can be proved by using the Integral Geometry which allows to study and measure sets of lines in the plane [20]. This interpretation of measures of lines depends on the duality relation used. It is valid in duality (θ, u) because the lines are equidistributed. It is not true in duality (a, b) in which the line $l : y = ax + b$ is associated with the dual point $l^* : (a, b)$.

Using the equation 4, the form factor between two convex objects O_i and O_j can be expressed by:

$$F_{ij} = \frac{\text{Area of the face associated with } O_i \text{ and } O_j}{\text{Area between } \lambda_i \text{ and } \mu_i \text{ for } \theta \text{ going from 0 to } \pi} .$$

- A **weighted sum depending on vertices and edges**:

We have seen that the form factor can be expressed by a weighted sum of curves. This expression can also be interpreted in the visibility complex. Consider the example in Figure 6. In order to determine the form factor between the objects O_i and O_j, we have to compute the lengths of curves (or "strings") stretched between the points (P_{u_l}, P_{d_r}), (P_{d_l}, P_{u_r}), (P_{u_l}, P_{u_r}), and (P_{d_l}, P_{d_r}). The straight portions of "strings" correspond to the bitangents associated with the vertices incident to the corresponding face. The curved portions of "strings" correspond to the edges bounding the face. Actually, going

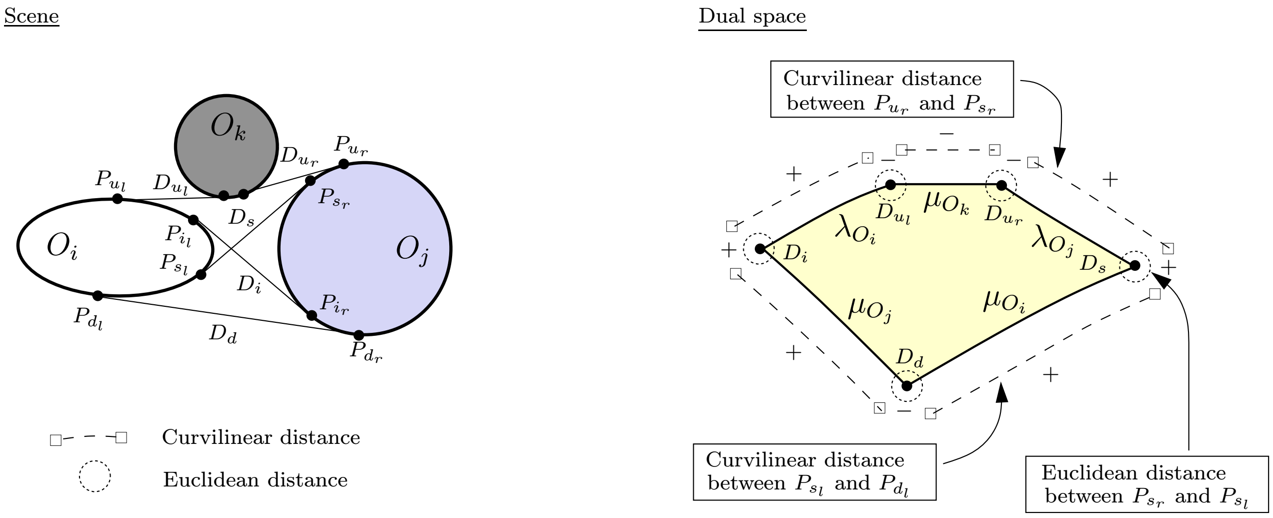


Figure 6: Two objects O_i and O_j and their associated face in the complex.

from P_{u_r} to P_{s_r} along the object O_j can be considered as going from D_{u_r} to D_s along the curve of tangency λ_{O_j} in the dual space.
With each vertex v of the face we associate a value $d_e(v)$ corresponding to the Euclidean length of the associated bitangent in the scene, and with each edge e of the face a value $d_c(e)$ corresponding to the curvilinear length of the corresponding portion of the associated object.

The form factor between two convex objects O_i and O_j can be expressed as:

$$F_{ij} = \frac{\sum_{v\ bounding\ face} d'_e(v) + \sum_{e\ bounding\ face} d'_c(e)}{2L_i} \ ,$$

$$\text{where } \ d'_e(v) = \begin{cases} d_e(v) & \text{if } v \text{ is an extremal vertex} \\ & \text{(first or last vertex)} \\ -d_e(v) & \text{otherwise} \end{cases}$$

$$\text{and } \ d'_c(e) = \begin{cases} d_c(e) & \text{if the object associated with } e \\ & \text{is } O_i \text{ or } O_j \\ -d_c(e) & \text{otherwise} \end{cases}$$

Figure 6 shows the different signs that modify the different values obtained for each vertex and each edge of the associated face for a given pair of objects (O_i, O_j).

Note that this method is simpler than the previous one since it does not require a numerical integration.

4.2 Computing the discontinuity mesh

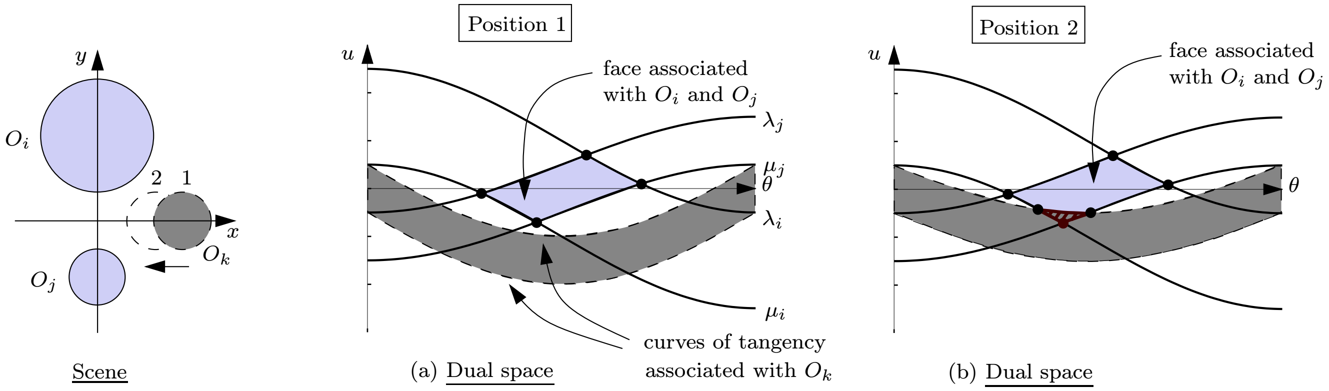
The visibility complex allows for easy computation of the discontinuity mesh. The limits of visibility of the objects, defined by the bitangents of the scene, define discontinuities in the radiosity function.
Figure 6 shows a simple example with an obstacle that limits the visibility between the objects O_i and O_j. The bitangent D_{u_l} in the scene corresponds to the limit of the shadow caused by O_k on O_j. The intersection of this bitangent with O_j represents a point of discontinuity. A bitangent in the scene corresponds to a vertex in the complex. The set of vertices of the complex is the set of discontinuities of a scene.

Once the visibility complex is built, the discontinuity mesh can be computed, by considering the vertices of the complex, in $\mathcal{O}(m)$ time (with m the number of vertices of the complex).

4.3 Towards an efficient update of the form factors in dynamic environments

In dynamic environments, two cases may occur : (1) the visibility complex is topologically unchanged: only the form factors concerning elements on the moving object must be recomputed; (2) the visibility between some objects has changed: the complex must be updated, some faces are destroyed (the corresponding form factors become null), others are created (requiring the recomputation of the corresponding form factors).
In practice, finding the form factors which need to be recomputed is not trivial. Previous applications of the radiosity method to dynamic environments, have attempted to reduce the number of unnecessary recomputed form factors ([1, 3, 7] and more recently [23]). But there is still no method exactly identifying the form factors strictly necessary to be recomputed. Making the visibility complex dynamic will make this possible.

Two types of visibility changes occur in dynamic scenes: when inserting or removing an object, and when moving an object. The first situation corresponds to the incremental construction of the visibility complex. The second modification may be handled by updating


Position 1
Position 2
y
O_i
2
1
x
O_k
O_j
u
face associated with O_i and O_j
λ_j
μ_j
θ
λ_i
μ_i
curves of tangency associated with O_k
u
face associated with O_i and O_j
θ

Scene (a) Dual space (b) Dual space

Figure 7: Example of a circle moving from the position 1 to the position 2.

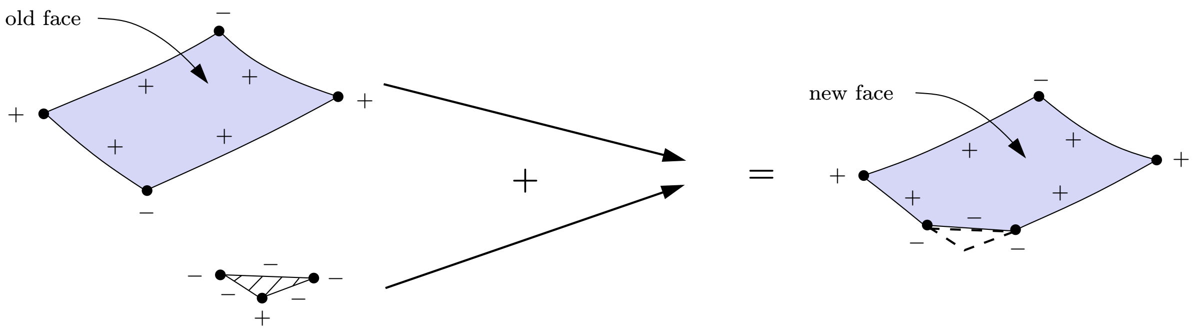


Figure 8: Update of the form factor between two objects.

the visibility complex along the curves of tangency of the moving object. We know that this update can be performed in linear time according to the *zone theorem* (see [6] or [2]). Once we have a dynamic update of the complex, we are able to identify the changes in visibility and to know exactly which form factors have to be recomputed, recomputation done efficiently with the complex. This can be illustrated with the following example.

Consider Figure 7 with an example of a moving object. When the dark gray circle is moved from the position 1 to the position 2, the face associated with the objects O_i and O_j is modified. The circle reduces the visibility between the objects O_i and O_j. A new face associated with these objects with a new position of the dark gray circle corresponds to the old face without the striped part (see Figure 7(b)). To update the form factor between the objects O_i and O_j, we just have to consider the striped part instead of recomputing everything with the new face (see Figure 8).

5 Implementation

We have implemented a system which uses the visibility complex for convex objects. It first uses the vertices of the complex for discretizing the objects of a 2D scene according to the discontinuities of visibility, and uses its faces for computing the form factors between mutually visible elements. Then the program computes the radiosity solution. It provides different visualizations: the scene, the value of radiosity for each element, the matrix of form factors and the visibility complex associated to the scene.

Figure 9 shows a simple example with three circles: the circle in front is a light source which lightens the two others. Figure 9(d) illustrates that the visibility complex is not planar. The dual arrangement of the scene (see Figure 9(b)) corresponds to a projection of the complex onto the dual plane.

6 Conclusion

The study in the static case has proved that the visibility complex is very useful for radiosity computation in static scenes. The complex directly provides the discontinuity mesh, and prevents unnecessary computation by

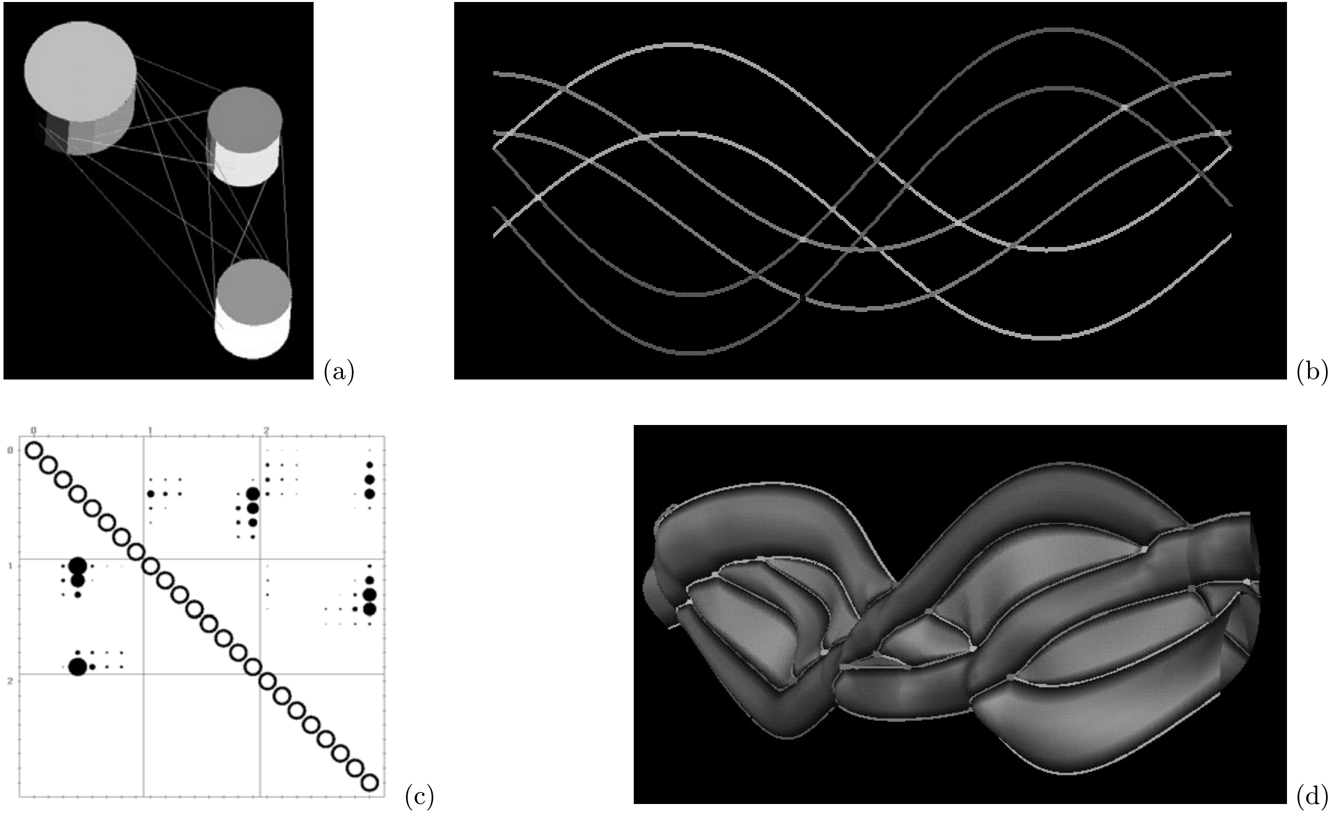


Figure 9: (a) scene and samples with radiosity value; (b) dual arrangement of the scene; (c) matrix of form factors; (d) visibility complex of the scene.

considering only mutually visible parts between objects. It also gives an efficient method for computing the form factors.

The visibility complex should provide important benefits for dynamic environments by offering a way of finding only the necessary re-computations. Work is currently done in this direction. We proved the visibility complex suitable for the realization of a test-bed environment for radiosity in dynamic scenes. A study in 2D makes the phenomenon of radiosity in dynamic environments more comprehensive, thus makes possible the development of efficient algorithms for radiosity in dynamic environments. We have decided to focus on the visibility complex for polygonal scenes since it is easier and more usual to work with polygonal objects in computer graphics. It is also more suitable for the hierarchical adaptations that we are studying.

For the 3D case, two approaches are being studied. For architectural scenes where the most important occlusions are vertical walls, we plan to use projections of those walls onto a horizontal plane, and to treat the visibility in the resulting 2D case. This will provide a good subset of mutually visible objects. We also envisage to build a 3D equivalent to the complex. This data-structure will have a very significant complexity, but it should have the potential to make of visibility phenomenon in dynamic environments more understandable. A hierarchical approach should decrease the complexity of such a structure.

References

[1] Daniel R. Baum, John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. The back-buffer algorithm: An extension of the radiosity method to dynamic environments. *The Visual Computer*, 2(5):298–306, September 1986.

[2] J.-D. Boissonnat and M. Yvinec. *Géométrie Algorithmique*. Collection Informatique. Ediscience international, 1995.

[3] Shenchang Eric Chen. Incremental radiosity: An extension of progressive radiosity to an interactive image synthesis system. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH'90 Proceedings)*, volume 24, pages 135–144, August 1990.

[4] F. Durand. Etude du complexe de visibilité. Rapport du DEA d'Informatique de Grenoble, France, June 1995.

[5] F. Durand and C. Puech. The visibility complex made visibly simple. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, page V2, 1995.

[6] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*, volume 10 of *EATCS Monographs on Theorical Computer Science*. Springer-Verlag, 1995.

[7] David W. George, Francois X. Sillion, and Donald P. Greenberg. Radiosity redistribution for dynamic environments. *IEEE Computer Graphics and Applications*, 10(4):26–34, July 1990.

[8] C. Goral, K. E. Torrance, and D. P. Greenberg. Modeling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH'84 Proceedings), 18:3*, pages 213–222, July 1984.

[9] S. J. Gortler, P. Schroder, M. F. Cohen, and P. Hanrahan. Wavelet radiosity. In *Computer Graphics (SIGGRAPH'93 Proceedings)*, pages 221–230, August 1993.

[10] P. S. Heckbert. *Simulating Global Illumination Using Adaptive Meshing*. PhD thesis, UC Berkeley, June 1991.

[11] P. S. Heckbert. Radiosity in flatland. In *Computer Graphics forum (EUROGRAPHICS'92 Proceedings), 11:3*, pages 181–192, September 1992.

[12] N. Holzschuch, F. Sillion, and G. Drettakis. An efficient progressive refinement strategy for hierarchical radiosity. In *Fifth Eurographics Workshop on Rendering, Darmstadt, Germany*, pages 343–357, June 1994.

[13] H. C. Hottel. Radiant heat transmission. In W. H. McAdams, editor, *Heat Transmission*, chapter 4. McGraw-Hill, New-York, 3rd edition, 1954.

[14] M. Pocchiola and G. Vegter. Sweep algorithm for visibility graphs of curved obstacles. Manuscrit, Liens, Ecole Norm. Sup., Paris, June 1993.

[15] M. Pocchiola and G. Vegter. The visibility complex. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 328–337, 1993.

[16] M. Pocchiola and G. Vegter. Computing the visibility graph via pseudo-triangulation. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 248–257, 1995.

[17] S. Rivière. Topologically sweeping the visibility complex of polygonal scenes. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C36–C37, 1995.

[18] S. Rivière. Dealing with degeneracies and numerical imprecisions when computing visibility graphs. 12th European Workshop on Computational Geometry CG'96, Muenster, Germany, 1996.

[19] S. Rivière. Experimental comparison of two algorithms for computing visibility graphs. Manuscrit, 1993.

[20] L. A. Santalo. *Integral Geometry and Geometric Probability*, volume 1 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley Publishing Company, 1976.

[21] M. Sbert. An integral geometry based method for fast form-factor computation. In *Computer Graphics forum (EUROGRAPHICS'93 Proceedings), 12:3*, pages 409–420, September 1993.

[22] P. Schroeder, S. Gortler, M. Cohen, and Pat Hanrahan. Wavelet projections for radiosity. In *Proc. 4th Eurographics Workshop on Rendering, Paris, France*, pages 105–114, June 1993.

[23] E. S. Shaw. Hierarchical radiosity for dynamic environments. Master's thesis, Cornell University, August 1994.

[24] F. X. Sillion and C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann Publishers, Inc., 1994.

[25] S. J. Teller. *Visibility Computations in Densely Occluded Polyhedral Environments*. PhD thesis, UC Berkeley, 1992.

[26] S. J. Teller and P. M. Hanrahan. Global visibility algorithms for illumination computations. In *Computer Graphics (SIGGRAPH'93 Proceedings)*, pages 239–246, August 1993.