Efficient Radiosity in Dynamic Environments

David Forsyth, Chien Yang, Kim Teo

August 20, 1995

Abstract

A method of determining radiosity in an environment containing moving objects, is described. This method uses the hierarchical techniques of Hanrahan *et al.* to obtain a static solution. Hanrahan's techniques efficiently create a hierarchical meshing of the environments geometry, and create links from element to element based on the magnitude of the form-factor between the elements. These ideas extend naturally to a dynamic environment, as only three atomic editing operations are required to update a hierarchy when an object moves: a link can be moved up or down the hierarchy, or a link can be occluded. Our algorithm exploits these simple editing processes to maintain the hierarchy, and then uses an iterative technique to solve the resulting linear system. The approach is extremely efficient, requiring little work between frames. **Keywords:** Radiosity, Computer Graphics.

0.1 Introduction

Diffuse interreflections are a source of various substantial effects in the radiance of scenes. Reproducing these effects appears to be essential to creating realistic renderings of scenes [14]. Diffuse interreflections are accurately modelled by a Fredholm equation of the second kind, which gives the radiance of a patch as that on the patch due to the source alone plus that on the patch arising from coupling to other patches.

Consider a set of surfaces in space, parametrised as $\mathbf{x}(\mathbf{u})$, where u is a two-dimensional vector of parameters. Every surface reflects light onto every other surface within a line of sight; the resulting radiance of the collection is:

$$N(\mathbf{u}) = N_0(\mathbf{u}) + \rho(\mathbf{u}) \int_S K(\mathbf{u}, \mathbf{v}) N(\mathbf{v}) dA$$

where $N(\mathbf{u})$ is the radiance at $\mathbf{u}$, $N_0(\mathbf{u})$ is the "initial radiance" at $\mathbf{u}$ (that is, the radiance at $\mathbf{u}$ if all other surface patches are absent), $\rho(\mathbf{u})$ is the reflectance at $\mathbf{u}$ and $K(\mathbf{u}, \mathbf{v})$ is the *form factor* from patch $\mathbf{v}$ to patch $\mathbf{u}$. K expresses the gain in transferring radiance from patch $\mathbf{v}$ to patch $\mathbf{u}$, and has the form:

$$K(\mathbf{u}, \mathbf{v}) = \frac{1}{\pi} \frac{(\mathbf{n}(\mathbf{u}).d_{uv})(\mathbf{n}(\mathbf{v}).d_{vu}) View(\mathbf{u}, \mathbf{v})}{(d_{uv}.d_{uv})^2}$$

where $d_{uv} = \mathbf{x}(\mathbf{v}) - \mathbf{x}(\mathbf{u})$ and $View(\mathbf{u}, \mathbf{v})$ is one if there is a line of sight from $\mathbf{u}$ to $\mathbf{v}$, and zero if there is not. Analytical results are hard to establish, though some exist [5]. The $View$ term is discontinuous and bears a complicated relationship to shape, and the relationship between an object's shape and its radiance - the main matter to study - is exceptionally complex due to the form of the kernel.

0.1.1 Numerical Solutions

Usually, finite elements are used in constructing radiosity solutions, and the main thrust of work has been on point collocation methods using constant elements. The result is an $O(n^2)$ algorithm, where the dominant problem is computing the approximation to the kernel. Galerkin methods, though analytically elegant, tend to be inappropriate as a result of the extra integrations required, but have recently become more popular (e.g. [18]). Typical recent approaches to obtaining efficient numerical solutions include: **progressive radiosity**, where an incremental approximate solution follows from allowing elements to "shoot" light onto every other patch [3, 13]; and **importance-driven radiosity** where for a particular view of a radiosity solution, an adjoint equation is used to determine the patches that make the most crucial contribution to that view [17]. When constant elements are used, the solution is smoothed using an interpolatory scheme to avoid a harlequin appearance. Achieving aesthetically pleasing solutions this way is often rather difficult, as human observers tend to be insensitive to the accuracy (in, say, an L^2 norm) of a solution, but highly sensitive to certain kinds of error, typical of this form of method. These effects are well illustrated in a recent text-book on the subject, which also develops the theory in reasonable detail [4].

Hierarchical Radiosity

Using an analogy with the recent fast numerical techniques for the n-body problem (e.g. [6], and in a different form, [8]). Hanrahan *et al.* [10] recently constructed a *hierarchical radiosity algorithm* which is extremely efficient. Hanrahan's approach takes a system of polygons, and constructs a hierarchy of elements using a quadtree, with a polygon at the root of each tree. Links are constructed between elements of the hierarchy which have a different root by testing the form-factor between the elements (which also falls off as the reciprocal square of distance); if it is smaller than some threshold, a link is established (figure 0.1). Elements are then given an initial radiance, and radiosity is propagated along these links using a simple "up-and-down" algorithm. This approach is efficient, because it allows multiple representations of the same radiator, and because it constructs links which have the same radiometric significance (rather than having elements at either end be the same size).

0.2 Radiosity in Dynamic Environments

The obvious approach to obtain a radiosity solution for a sequence of pictures of an environment containing moving objects is computing a new radiosity solution for each frame is recognised to be hopelessly inefficient, because it does not use the information contained in the previous solution. Baum *et al.* attributed the temporal coherence of the kernel and of solutions to object coherence, and

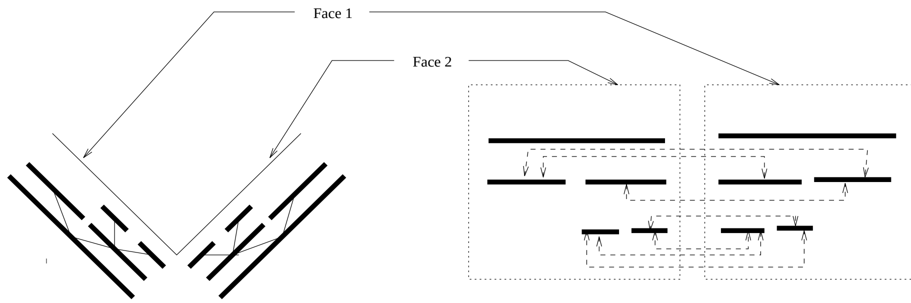

Figure 0.1: The left half of the figure shows an element hierarchy superimposed on a simple translationally symmetric geometry - a gutter formed by two plane walls at 45^o. The symmetry has been integrated out. Elements are given as thick lines, and the geometry is drawn using thin lines; the relationship in the hierarchy is given by the position of the elements. The right half shows the element-element links for a particular threshold in this geometry.

exploited it to obtain a factor of 25 speedup for image sequences where all moving objects and their paths were known in advance, by computing static relationships in a preprocessing pass [1]. George *et al.* use the concept of *negative energy*; when an object moves, some patches distribute negative energy to account for the new shadows now cast by the patch [7]. Incremental information about form-factors and solutions is not accumulated; however, Müller and Schöffel gained efficiency by using sophisticated data structures to keep track of occlusion and repropagation information [15]. Chen [2] reformulated progressive radiosity to address dynamic environments; in his approach, when an object moves, affected patches compute incremental form-factor terms, and propagate incremental radiosity. Each patch that has contributed to the existing solution must then re-shoot its incremental radiosity. The approach is particularly appropriate for interactive modelling systems, as it is not necessary to wait for the algorithm to converge before an object is moved again.

Hanrahan's method does not address a dynamic environment, but its efficiency is extremely attractive. There is a natural way to extend the method to a dynamic environment, which appears to be quite efficient. This approach assumes that objects are neither created nor destroyed, but is otherwise quite general. The key to the approach is a study of the behaviour of links as objects move.

0.2.1 Link Activities

An element hierarchy can be updated to reflect a movement by relatively simple editing. At any timestep, the hierarchy of elements for polygons that have not moved since the last frame, and the links between those elements, do not change over time; the only possibility is that elements can become temporarily occluded. Links to or from dynamic objects are more interesting, but display relatively simple behaviour. The possibilities for both dynamic and static links are:

- **Occlusion initiated or terminated:** in this case, where a link is broken by occlusion, the form-factor across the link may be modified or zeroed (if the link is completely occluded), but the link is retained as the occlusion may disappear later; if the occlusion disappears, the form-factor can be restored. These events can occur for either static or dynamic links (figure 0.2).
- **Promoting a link:** as polygons move away from one another, the form factor between elements falls off; for sufficiently distant polygons, the form factor may become so low that a link can move up the hierarchy to larger elements. If a link moves up the hierarchy, then it may be possible to reap some elements from the base of the hierarchy (for example, figure 0.3). This is easily accomplished, as the condition can be detected when the link is moved. This event can occur only for a dynamic link (figure 0.3).
- **Demoting a link:** as polygons move closer, the form-factor increases, and elements may need to be split and their children linked to maintain numerical accuracy. If a link moves down the hierarchy, elements may need to be created; again, this is easily accomplished, as the condition can be detected when the link is moved. This event can occur only for a dynamic link (figure 0.3).

Once it is known that only a limited range of events can occur at a link, it is important to be able to predict which links are affected in the next frame; with an effective prediction component, the algorithm takes the following form:

- Establish a hierarchy of elements and links for the geometry in the first frame. For each time-step:
  1. Predict which links will be promoted, which will be demoted, and which experience an occlusion event.
  2. Update only those links, and edit the hierarchy to reflect promotions and demotions.
  3. Use a prediction of the solution (usually an extrapolate based on previous solutions) to start an iterative solver.

Note that the solution process and the process of editing the hierarchy do not depend on the classification of links as static or as dynamic. The classification is extremely important for predicting link events, which can have a significant effect on efficiency. In our experience with implementation, the dominant cost is in occlusion testing, and it is important to be able to predict effectively which

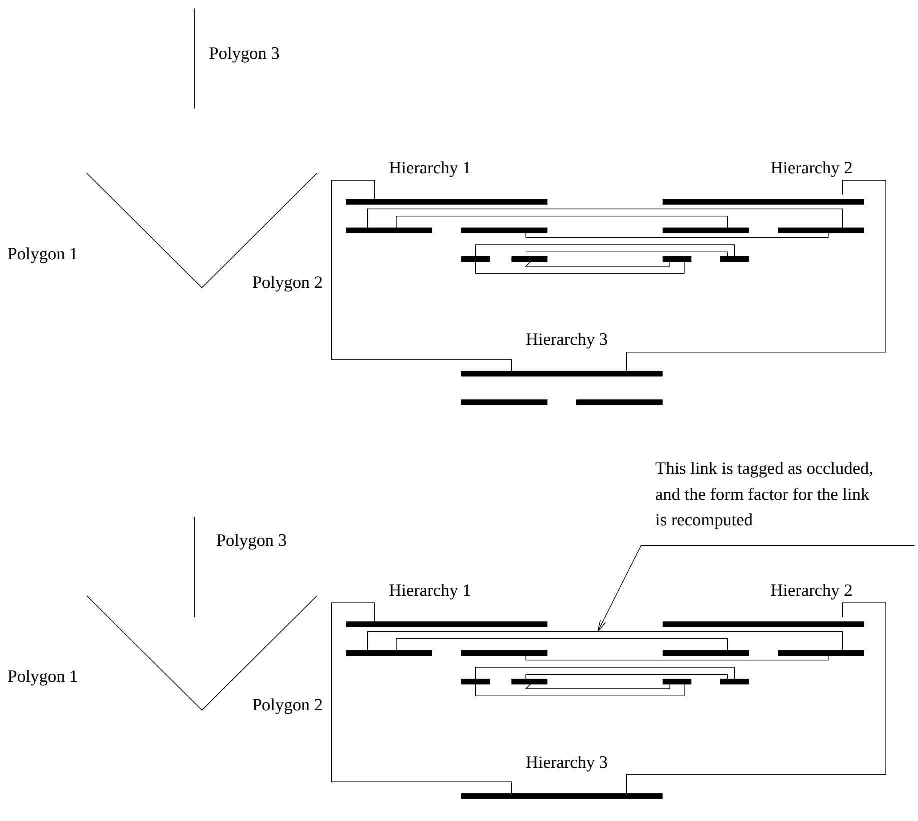

Figure 0.2: As polygon 3 approaches 1 and 2, it occludes links across the hierarchies of elements; these links are marked as being occluded, and the form factor is recomputed. Even if the link has form-factor zero, it is retained for future reference as polygon 3 might move away.

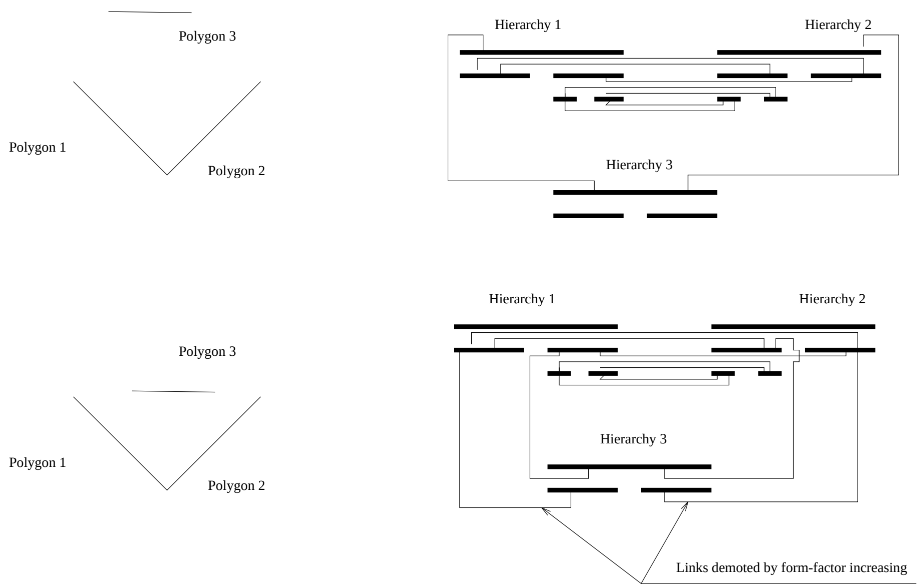

Figure 0.3: As the geometry changes from the top to the bottom, the links between elements of hierarchies 1 and 2 and elements of hierarchy 3 are demoted because polygon 3 approaches the other two polygons and the form-factors increase. Promotion is the reverse of this effect; as the geometry changes from the bottom to the top, the links between elements of hierarchies 1 and 2 and elements of hierarchy 3 are promoted, because polygon 3 moves away and the form-factors decrease.

links will have their form-factors changed by occlusions, to avoid examining each link at each step.

Figure 0.4 (see color plate 12) shows dynamic link events for an orthogonal gutter being approached by a patch (the translational symmetry has been integrated out). Figures are read top to bottom, left to right. The links, shown in pink, are to the centre of the relevant element in the hierarchy, and are drawn as lines joining element centres. Links to the patch are dynamic, others are static. In the first frame, many links go the centre of the patch (hence, to the top element in the hierarchy); by the second frame, these links have been demoted, and so are represented as going to the centre of each half of the patch. As the patch gets closer to the gutter, more links are demoted, and links move to lower levels of the hierarchy, and so move to points given by finer subdivisions of the patch.

0.2.2 Predicting Link Events

Our present implementation predicts link events using an inefficient but conservative strategy. It is known in advance which objects are dynamic, and dynamic and static links are treated differently. To date, we have worked only with geometrical layouts that guarantee that static links are not occluded (for example, a static room with a moving light in the ceiling), thereby bypassing the problem of predicting occlusion events with static links.

The two columns on the left in figure 0.5 (see color plate 13) show a motion sequence, with the light moving around the ceiling. The mesh elements with links to the light source are shown to the right of the relevant frame; these are the only dynamic links. The figures should be read from top to bottom. Note how dynamic links are demoted as the light approaches a wall, and promoted as it leaves. In the present, early, implementation, all dynamic links have their form-factors recomputed; more sophisticated prediction algorithms will reduce this extra work. The figures were rendered using a Gouraud shading of a Delaunay triangulation of the sample points. Timing information appears in table 0.1. The two right hand columns show a motion sequence, with the light moving around the ceiling. In the central column, the dynamic links are reinspected at each step; in far right-hand column, links are extrapolated up to seven frames ahead, to predict events. Form-factor values are also extrapolated. This leads to a deterioration in rendering quality for some speed-up.

The results in the left-hand columns of figure 0.5 (see color plate 13) are derived from an implementation where each dynamic link is tested at each step; clearly, this is an inefficient approach. Results in the right-hand column of figure 0.5 are obtained from an implementation where dynamic link events are predicted by linear extrapolation. For each dynamic link, a linear extrapolate predicts the frame at which the next link event will occur. This frame is recorded in the link; until that frame is reached, the form-factor and occlusion terms are obtained by a linear extrapolate. When the frame at which an event is predicted, is reached, the form-factor is re-evaluated. The oldest form-factor value involved in the extrapolation is discarded, and the new value is incorporated. The results compare reasonably well in quality, for some speed-up (table 0.1).

0.3 Results

Our present implementation works for an environment where static links will not be occluded, so that predicting static occlusion events is unnecessary. Link promotion and demotion are dealt with by re-examining all dynamic links at each time-step. The geometry used consists of a three-walled room, with a floor, no ceiling, and a box lying on the floor, with a light source moving around in the plane of the ceiling. In the lefthand columns of figure 0.5 (see color plate 13), the sample points of the radiosity solution are Delaunay triangulated and these triangles are Gouraud shaded. This approximation technique causes some annoying perceptual effects, particularly Mach bands and occasional disruptions in the geometry of the isophotes due to a non-unique triangulation, but has the merit of being fast and efficient; furthermore, if the triangulation is done in the frame of the faces, it is affine covariant, meaning that it supports reasonable changes in viewpoint without the radiance appearing to slip across surfaces. As the timings in table 0.1 show, the per-frame timing is extremely fast, with a relatively low initial cost. We are presently exploring the prospects for better approximation techniques that run as fast or faster.

Figure 0.5 and table 0.1 also show results for a slightly different geometry, where polygons are flat-shaded; in the central column of figure 0.5, each dynamic link is re-inspected at every step, whereas in the right-hand column, the link form-factors are extrapolated for up to seven frames in advance. The deterioration in image quality is accompanied by a slight improvement in speed.

0.4 Future Work

Predicting link events is a significant outstanding problem. Visualizing link events as a process where objects enter or leave shafts that run between two elements, or where the shaft geometry changes, appears to be productive. This view can be generalised to a more sophisticated approach to the problem of predicting link events by constructing and maintaining a database, which can answer the query "Which shafts pass through this region of space?". The database is constructed by decomposing free space surrounding objects into cells of various sizes; each cell records the shafts passing through the cells, and hence the links that might be occluded if an object passes through the cell. This means that, by tracking the path of a moving object, the links between static objects that will be occluded by the object can be quickly and efficiently identified and

Sequence	Number of frames	Average time per frame	Standard deviation	Time to set up and solve frame 1
Gouraud shaded corridoor	109	3.81 s	0.760 s	30.95 s
Cubical room	20	4.21 s	1.36 s	22.24 s
Cubical room with extrapolation	20	3.92 s	0.845 s	19.74 s

Table 0.1: Timing information for the three sequences rendered. The time to set up and solve frame 1 is representative of the time required to solve a given frame from scratch; this is substantially larger than the average time taken to solve for radiosity and render a frame, indicating that there are substantial savings to be had in reusing radiosity information. Note that extrapolating form-factors gives about a 7 % saving in the average time taken to render a frame, and substantially reduces the standard deviation; this is probably because extrapolation is poor at predicting events when many elements appear or disappear from view. There are typically 5000-6000 polygons in each frame.

recomputed.

This approach does not address the problem of objects occluding dynamic links, as it appears to be difficult to update the shaft database quickly when the ends of the shafts move. We are at present experimenting with an approach where dynamic link occlusions are managed by extrapolation, and static link events by a spatial database. In particular, this system will not require dynamic and static objects to be separated in advance. When an object moves, it will be erased from the static database and treated as a dynamic object; if it remains stationary for a sufficient number of frames, it will be reinserted into the static database.

As the results show, hierarchical techniques are not effective in preserving the discontinuities in a radiosity solution. The dynamics of these discontinuities are complicated, and maintaining a discontinuity based mesh in a dynamic environment appears to be extremely difficult; it may be simpler and more effective to predict the discontinuities at each frame, and then impose them on the solution in a "cleaning-up" pass after the solution for a frame has been determined. It appears unlikely that any coherence in the discontinuities can be effectively exploited. Post-processing solution meshes to demote links that transfer too much energy would also improve the solutions. The adaptations made to the mesh in this way could be maintained in the same way that dynamic links are maintained. We expect them to show substantial frame-to-frame coherence. Clustering groups of polygons that interact, as in the work of Sillion [16], and lazy evaluation of links between hierarchy elements, as in the work of Holzschuch *et al.*, both appear to offer substantive savings. A substantial source of these savings appears to be *domain decomposition*, where the geometry is segmented into sets whose radiometric interactions can be approximated easily; we intend to investigate principled techniques for domain decomposition.

Acknowledgements

We are grateful for instructive and helpful comments, including further references, from anonymous referees, and from Andrew Zisserman. This research was carried out at the University of Iowa Department of Computer Science, supported in part by a National Science Foundation Young Investigator Award with matching funds from GE, Eugene Rikel, Rockwell International and Tektronix.

Bibliography

[1] Baum, D.R., Wallace R.W., Cohen M.F., Greenberg D.P. "The Back-Buffer: An Extension of the Radiosity-Method to Dynamic Environments," *The Visual Computer*, **2**,5, 1986, pp. 298-306.

[2] Chen, S.E. "Incremental Radiosity: An Extension of Progressive Radiosity to an Interactive Image Synthesis System," *Computer Graphics (SIGGRPAH '90 proceedings)*, **24**, 4, 1990, pp. 135-144.

[3] Cohen, M., Chen, S.E., Wallace, J.R., and Greenberg, D.P. A progressive refinement approach to fast radiosity image generation. Computer Graphics (SIGGRAPH '88 Proceedings) 22:4 (Aug 1988), pp. 75-84.

[4] Cohen, M.F. and Wallace, J.R., *Radiosity and realistic image synthesis*,Academic, 1993.

[5] Forsyth, D.A. and Zisserman, A.P. "Shape from shading in the light of mutual illumination,", *Image and Vision Computing*, AVC special issue, 42-49,1990.

[6] Greengard, L. (1988) The rapid evaluation of potential fields in particle systems. MIT Press, Cambridget, MA.

[7] George, D.W., Sillion, F.X., and Greenberg, D.P. "Radiosity redistribution for dynamic environments." IEEE Computer Graphics and Applications 10:4 (July 1990), pp. 26-34.

[8] Hackbusch, W. and Nowak, Z.P., "On the fast matrix multiplication in the boundary element method by panel clustering," *Numer. Math.*, **54**, 463-491, 1989

[9] Haines, E. A., and Wallce, J.R. Shaft Culling for efficient ray-traced radiosity. In Second Eurographics Workshop on Rendering (Barcelona, Spain, May 1991).

[10] Hanrahan, P., Salzman, D. and Aupperle, L., "A rapid hierarchical radiosity algorithm," SIGGRAPH-91, 197-206, 1991.

[11] Holzschuch, N., Sillion, F.X. and Drettakis, G., "An efficient progressive refinement strategy for hierarchical radiosity," this volume.

[12] Koenderink, J.J. and Van Doorn, A.J., "Geometrical modes as a general method to treat diffuse interreflections in radiometry," *J. Opt. Soc. Am.*, **6**, 1, 1987, pp15-31.

[13] Lischinski, D., Tampieri, F., and Donald, D.P., "Discontinuity meshing for accurate radiosity". *IEEE Computer Graphics and applications* (November 1992) pp. 25-39.

[14] Meyer, G.W. "Color calculations for and perceptual assessment of computer graphic images," PhD thesis, Program of Computer Graphics, Cornell University, 1986.

[15] Müller, S. and Schöffel, F., "Fast radiosity propagation for interactive virtual environments using a shadow-form-factor-list," this volume.

[16] Sillion, F.X. "Clustering and volume scattering for hierarchical radiosity calculations," this volume.

[17] Smits, B. E., Arvo, J. R., and Salesin, D. H. An importance-driven radiosity algorithm. Computer Graphics (SIGGRAPH '92 Proceedings) 26:4 (July 1992), pp. 273-282.

[18] Zatz, H.R. "Galerkin radiosity: A higher order solution method for global illumination." Master's Thesis, Program of Computer Graphics, Cornell University, Aug. 1992.

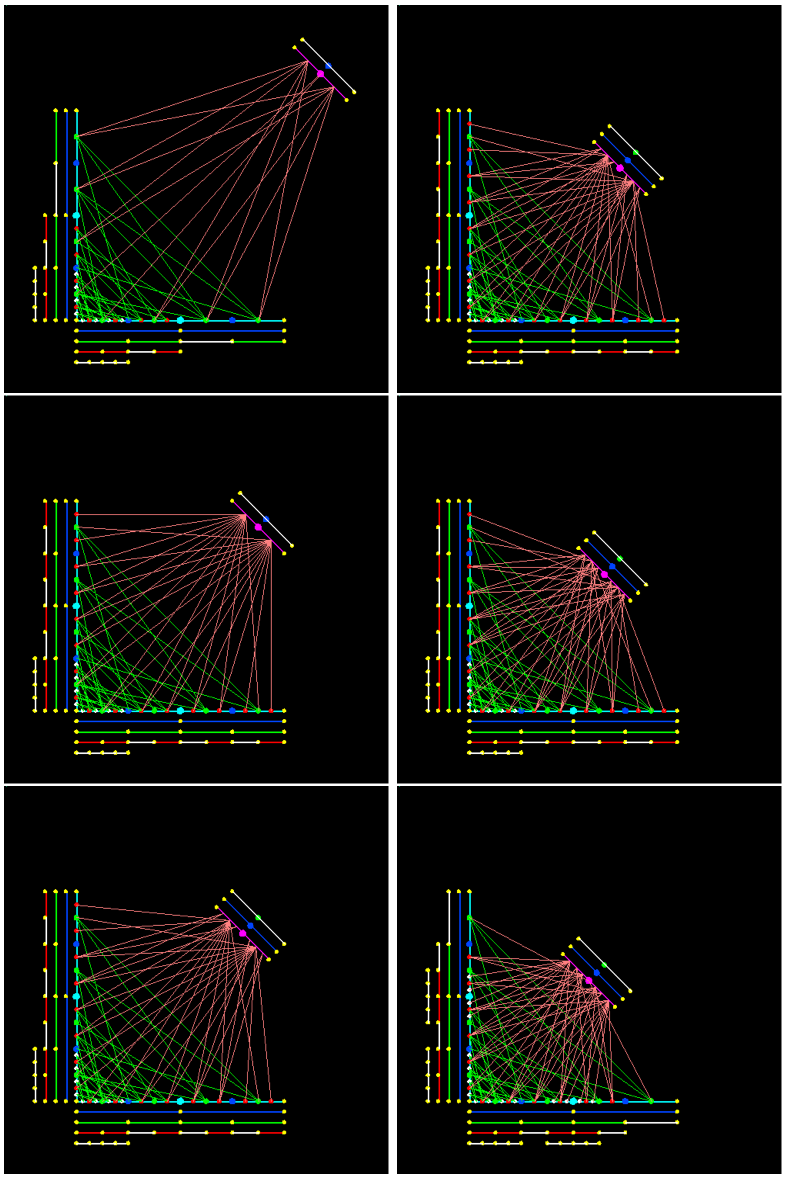

Figure 0.4:

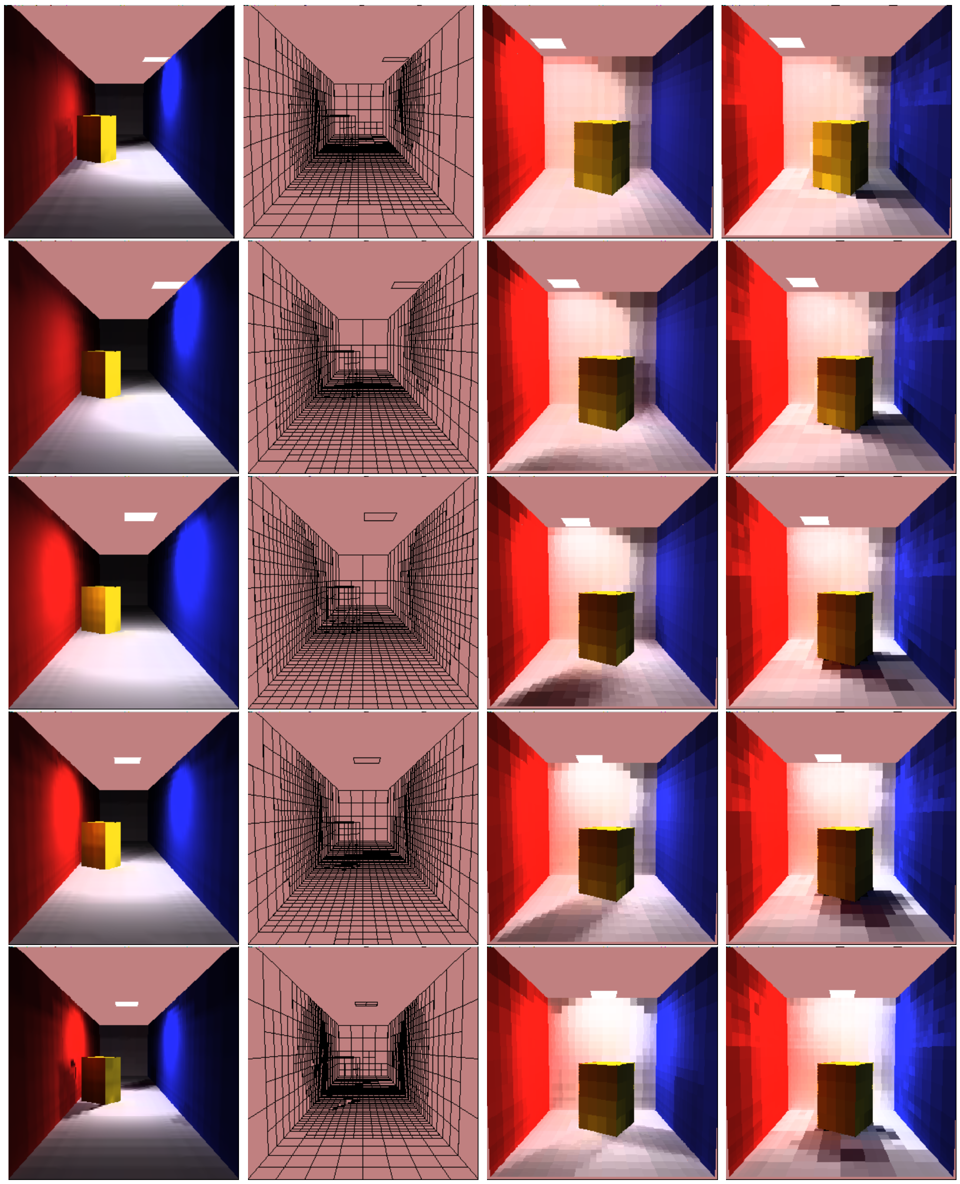

Figure 0.5: