RADIOSITY ENGINE COMMENTS

A further discussion of features and techniques
for the Electric Image Universe Radiosity Engine.

CONTENTS

GENERAL DISCUSSION

Basically, a radiosity engine calculates how much energy reaches each surface in a scene—both from light sources and from all other surfaces in the scene. *Radiosity* is the rate that energy leaves a light source by emittance, or other surfaces by reflectance. An application that calculates radiosity values throughout a scene is called a *radiosity engine*, the process is called a *simulation*, and the result is referred to as a *solution*. The radiosity can be represented in the visual spectrum—a color, and as such, is useful for computer graphics.

The engine determines the *visibility* between two surfaces, or how much of a *source* surface is visible to a *receiver* surface. The visibility is multiplied by the emittance or reflectance of the source surface to determine the *form-factor*, or the amount of energy transferred between the two surfaces. The engine repeats this process many times for every possible pairing of surfaces in the scene.

Most rendering methods, including Phong and ray-tracing, treat light sources as distinctly different entities from the surfaces they illuminate. Radiosity methods, however, treat all surfaces equally—a light source is simply a surface emitting energy at the beginning of the simulation. All surfaces in the scene are inherently one-sided, have surface area, and have an orientation, or normal.

Clearly, a great many calculations are required to run a simulation on a scene of more than a few surfaces. Over the years, researchers have made significant optimizations to the process—the quality of the solution is increased and the time required to reach a solution is decreased dramatically.

FORM-FACTORS

One significant optimization is to the method of calculating form-factors. Instead of finding the form-factor for each source-receiver combination, the source surfaces may be formed into groups and a form-factor for the entire group may be used instead. Furthermore, these form-factors can be approximated with fewer calculations. These approximations are not perfect, however, but approximated form-factors are held within optimal levels.

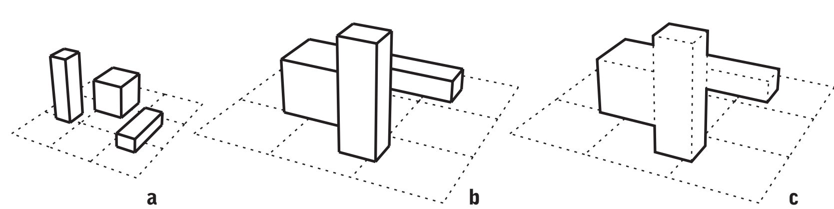


FIGURE 1:
a) a number of simple objects comprising a scene;

b) those same objects when viewed from a given point;

c) an abstraction of those objects.

For example, consider three objects resting on a plane (**Figure 1a**). Instead of using eight individual form-factors, one for each visible surface (**Figure 1b**), a single form-factor approximation may be substituted (**Figure 1c**) for the entire group of visible surfaces.

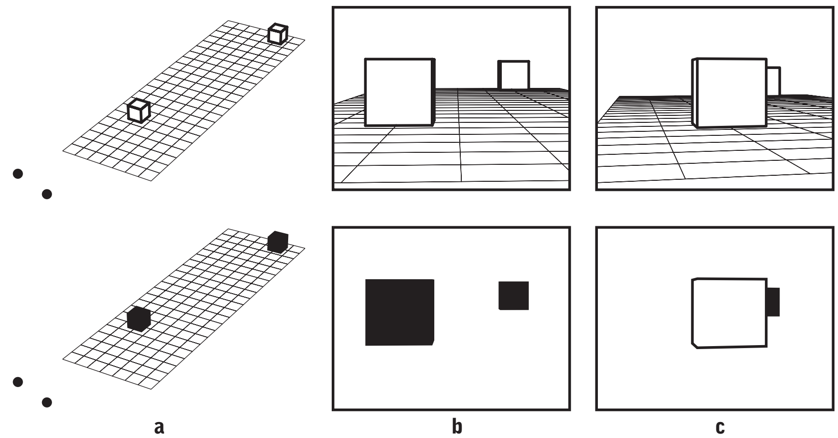


FIGURE 2:
a) the scene showing the two viewpoints and the visibility of each cube from an aerial viewpoint;

b) the view and visibility of each cube from the right-most viewpoint in (a);

c) the view and visibility of each cube from the left-most viewpoint in (a).

The form-factor is always relative to the receiver. The greater the distance between the source and receiver, the less of a form-factor. Take two identical cubes on a plane, one further from a viewpoint than the other. From an aerial viewpoint (**Figure 2a**), the form-factor between the viewpoint and each of the cubes is similar. From a lower viewpoint (**Figure 2b**) however, the form-factor between the viewpoint and the closer cube is much greater.

Both visibility and form-factor are dependent on how much of one surface is visible from the receiver surface. When less of a surface is visible, the form-factor is likewise decreased. If the viewpoint of the two cubes is changed slightly such that the closer cube somewhat obscures the further cube (**Figure 2c**), the form-factor of the further curve is reduced accordingly.

The smaller the form-factor between two surfaces or groups of surfaces, the less energy is transmitted between them.

HIERARCHY

In order to efficiently estimate form-factors for groups of surfaces, another optimization organizes the surfaces into a *hierarchy*—each level of the hierarchy more abstract than the levels below.

Imagine a table in the center of a room. Atop the table are a vase and place setting. The top hierarchy level represents the table, vase, and place setting as a monolithic object, while lower levels distinguish between the different objects and ultimately the surfaces that comprise each object. The Radiosity Engine chooses a hierarchy level appropriate to the current calculation, and tracks the difference, or error, between the hierarchical approximation and the original model. When this error is greater than a threshold amount, based on distance and orientation, a lower level is chosen instead.

PROGRESSIVE REFINEMENT

The surface with the greatest amount of energy to impart to a scene on a per-surface-area-unit basis, will be the first surface chosen at any given time as the primary source. Once the energy from this surface is distributed throughout the scene, a new primary source is chosen using the same criteria. This process continues until the energy to be distributed is below the **Minimum Energy Level** for the scene.

THE DIFFERENT METHODS

Both methods, **Progressive** and **Hierarchical**, apply many of the same optimizations, including surface hierarchy and progressive refinement. Both methods alternate between energy-distribution calculations and receiver subdivision when the radiosity at different parts of the same surface vary by a certain amount. However, each method takes a very different approach towards reaching a solution, and settings that work well for one method are not likely to work well for the other.

As the name implies, the **Progressive** method performs one energy-distribution and subdivision cycle, or *iteration*, for each hierarchical group of objects. This cycle continues until either the **Time Limit** or the **Minimum Energy Level** is reached. The number of iterations necessarily varies, often widely, with the character of the geometry submitted.

The **Hierarchical** method distributes all the energy on a single hierarchical level until the **Minimum Energy Level** is reached during a single iteration. Each iteration uses successively lower hierarchy levels until the number of **Iterations** is reached. This is generalized, somewhat, as an accurate description of clustered hierarchical radiosity is beyond the scope of this discussion.

SOLUTION SETTINGS

In most cases, only a few of the fourteen possible settings in the **Solution** tab of the **Radiosity Settings** dialog (**Figure 3**) may need adjustment for each scene. The remaining are, typically, special-case settings. Some of the settings affect both methods, while others affect one method more than the other.

The ***Light Size Experiment*** *and* ***Light Shape Experiment*** *at the end of this document discusses how changes to some of these settings affect the radiosity solution of a simple scene.*

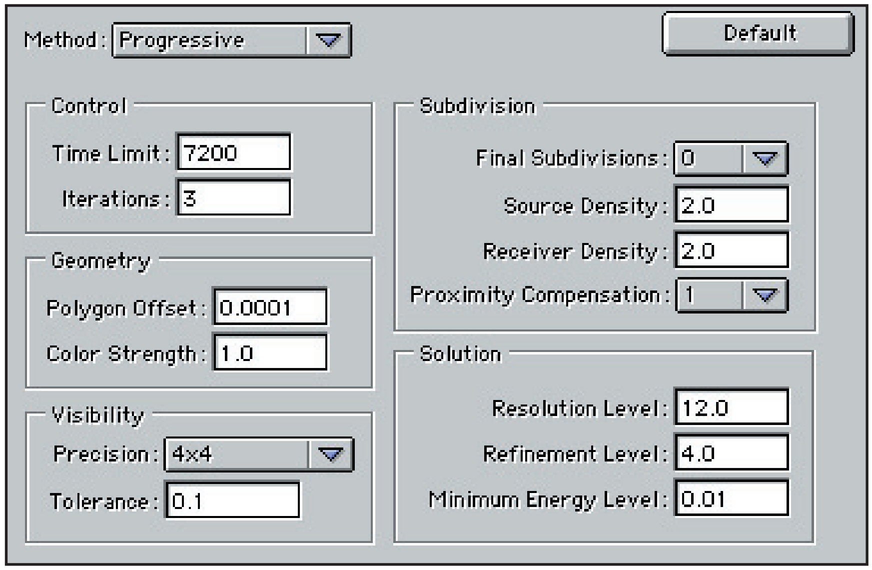


FIGURE 3.
The Radiosity Settings dialog box.

MINIMUM ENERGY LEVEL

The Engine solves the radiosity solution until a percentage of the total energy radiated into the scene remains unaccounted for; i.e. the **Minimum Energy Level**. This is the most critical setting for achieving an acceptable distribution of energy throughout the scene with either method. As more and more energy is accounted for, the engine solves for more and more subtle energy bounces.

SOURCE AND RECEIVER DENSITY

Subdivision in the Progressive method, is driven by two settings—**Source Density** and **Receiver Density**. The first controls the subdivision of source surfaces, either emitters or reflectors, resulting in increasingly accurate form-factor calculations. The second controls the subdivision of receiver surfaces, increasing the number of polygons and apparent resolution of the solution mesh. Both of these values vary with the scale of the scene, but not the size of the scene.

In general, where the energy comes from is less important than where it arrives. A good rule of thumb is to set the **Receiver Density** to at least twice that of the **Source Density**.

RESOLUTION AND REFINEMENT LEVELS

The quality of a Hierarchical method solution is primarily controlled by two settings—**Resolution Level** and **Refinement Level**. Both control error levels and thresholds that trigger changes in hierarchy level and subdivision density.

The **Resolution Level** controls the conditions that trigger subdivision, effectively controlling the resolution of the solution mesh. The **Refinement Level**, on the other hand, controls the transfer of energy—the radiosity solution itself, or the refinement of the solution. Adjustments to the **Refinement Level** have some effect when using the Progressive method.

VISIBILITY TOLERANCE

In some situations, increasing the **Visibility Tolerance** will result in a more accurate solution. A number of assumptions and approximations are made when determining the visibility component of the form-factor. The **Visibility Tolerance** is used to determine whether a more precise method of visibility calculation should be used.

ENERGY REFLECTION

White velvet reflects more light than black velvet, but black lacquer reflects more than either, color aside. The **Energy Reflection** field in the **Diffuse** tab of the **Group Info** dialog sets the amount of energy, with 1.0 as a baseline, to be reflected back into the scene off a surface in that group.

Taken as an average, most materials reflect approximately sixty-five percent of the light that strikes them, or 0.65. A mirror, however, typically reflects over ninety-five percent, or 0.95, while a woven rug might reflect only about thirty-percent, or 0.30. Super-reflective surfaces, or those whose Energy Reflection is greater than 1.0, while possible to specify, do not exist in nature.

LIGHTS

The lights built into Universe are idealized light sources that, while generally based upon the laws of nature, are customizable in ways that real-life light sources never could be. They have neither size nor surface area (except when rendering a glow layer), and allow adjustment to both falloff curve and falloff distance as well as other characteristics.

Radiosity light sources, however, explicitly follow the laws of physics—emitting energy uniformly with a falloff curve and distance consistent with the behavior of visible energy, or light, in the gaseous medium of the atmosphere.

A reasonable understanding of the differences between the two is essential when lighting a radiosity scene. Furthermore, rather than using the methodology and tricks of CG lighting, emulating reality is generally most effective—radiosity light sources have size, surface-area, and shape in addition to intensity and color.

LIGHT SIZE

A household light bulb is simply a vacuum chamber fitted with a socket, designed to support a small wire, or filament. The filament converts electrical energy into light energy and heat energy through resistance. The glass vacuum chamber itself does not emit light. A frosted bulb is different. Light from the filament is diffused as it passes through the frosted coating, effectively emitting from the glass itself.

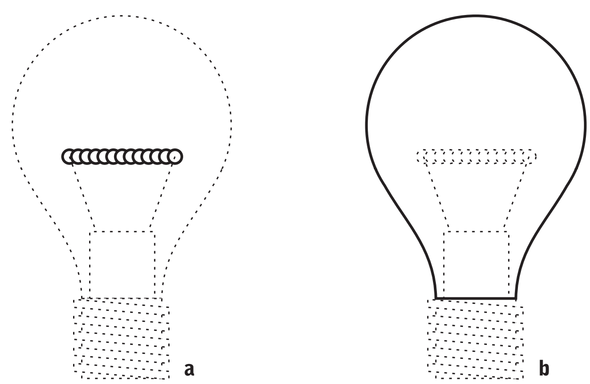


FIGURE 4:
a) a clear bulb (filament emitter) 0.000000032 square meters;

b) a frosted bulb (glass emitter) 0.18 square meters.

The emitting surface of a common household 60-watt clear bulb (**Figure 4a**), the filament, has a surface area of approximately 0.000000032 square meters. The emitting surface of a 60-watt frosted bulb (**Figure 4b**), effectively the glass, is approximately 0.018 square meters. Both emit roughly the same amount of energy, but the frosted bulb does so over a much larger area.

Thousands of times more energy will be added to a scene from the frosted bulb than from the filament, even if each are assigned the same intensity. The filament would emit 0.0000019 watts-per-square-meter, while the frosted bulb would emit slightly more than one watt-per-square-meter.

Bulbs with diffusers such as frosted bulbs, are used when the very bright filament inside would be uncomfortable or undesirable to look at. The character of the light itself is generally identical to an clear bulb unless the diffusing surface imparts a color or otherwise filters the light passing through.

Although a direct correspondence exists between the energy a bulb uses and the amount of light it emits, the "wattage" of a bulb generally refers to the quantity of electrical energy converted to light energy (and heat) by the filament.

Understanding how real-world light bulbs and fixtures operate is essential for reproducing their effects in a radiosity scene. Fluorescent lights, for example, energize a gas that in turn causes a coating on the inside of the glass tube to fluoresce, emitting a cool white light. To emulate a fluorescent tube, use an appropriately sized cylinder, and adjust the intensity to account for an emitting are comprised of the entire tube. Another example, halogen fixtures, use a very small tube as the emitter and a reflector to direct the light and alter the way that light is distributed—modifying, in effect, the falloff curve and falloff distance for the light source.

REFLECTORS

Surfaces constructed very near the emitter with the purpose of redirecting light from the tube or filament emitter in specific and calculated directions are called reflectors. Spot lights, flood lights, and halogen fixtures all rely on reflectors. Changing the geometry of a reflector will change the character of the light coming from the fixture (**Figure 5**). The same holds true for light sources in a radiosity scene.

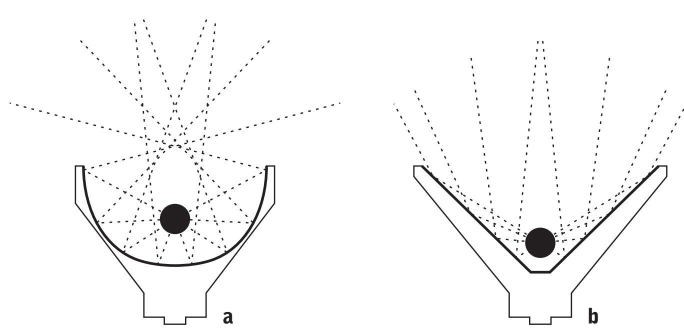


FIGURE 5:
Reflector bulb cross-sections with emitter in center with reflector below

a) with even light dispersion;

b) with edge-weighted light dispersion.

Although the energy distribution from light sources with reflectors can be very different from those without reflectors, the additional effort to construct reflector geometry is wasted without an understanding of how the reflector will actually affect energy distribution. Often a light source with a naive reflector produces a remarkably similar radiosity solution to a simple light source.

More surfaces mean higher solution times. Emitters should never contain more polygons than necessary to minimally describe the shape of the emitting surface. The same holds true for reflectors. Unnecessary tessellation will not yield higher-quality solutions.

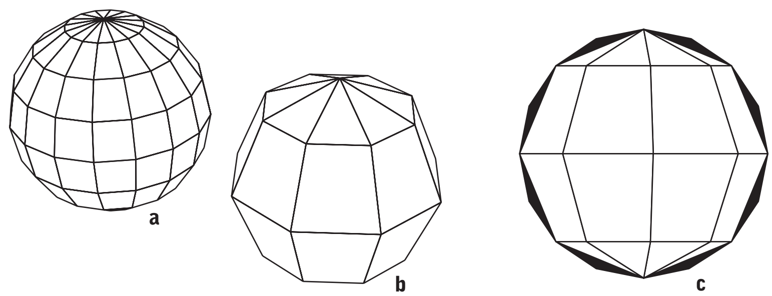


FIGURE 6:
a) 128 polygon sphere;
b) 32 polygon sphere;
c) difference between the two spheres.

Consider two spherical light sources. One (**Figure 6a**) has many more polygons than the other (**Figure 6b**), but from any viewpoint, the difference between the two is negligible (**Figure 6c**). The form-factor of both will be practically identical, but the solution time will be longer with the higher-polygon-count sphere.

SIZE AND DISTANCE

The amount of energy reaching a surface from a light source is a factor not only of intensity and color, but also size and distance. Less energy will reach a surface three meters away than an identical surface only one meter away. In order for the same amount of energy to reach the further surface, the size or intensity of the light source must be increased.

By the same token, less energy will reach a surface from a one-meter-square light source than from a half-meter-square light source of equal intensity and equal distance from the surface. The intensity of the smaller light source must be increased if the energy reaching the surface is to be the same.

DISTANT LIGHT SOURCES

Many real-world criteria are used to determine the energy striking the surface of the Earth from the Sun at any given moment—atmospheric conditions, latitude, and time of day to name a few. Simulating distant light sources is very difficult without locating the light source an unreasonable distance from the object to be illuminated.

Unless the scene is at a universal scale, it makes little sense to use a spherical light source to emulate the sun. Certainly from any point on any single planet, such as Earth, a single quadrangle will suffice. The intensity of the "Sun" surface depends on it's size and it's distance from the rest of the scene. Placing it twenty meters from the model of a house is not likely to result in acceptable results, but there isn't any reason to place it a realistic 150 million kilometers either. The compromise depends on the scale of the scene it is intended to illuminate.

GEOMETRY

Metaphysics aside, everything exists in three-dimensions, with thickness, a front, and a back (**Figure 10a**). In the realm of CG however, surfaces are inherently one-sided, and without thickness—simply 2d planes rotated in space, oriented in a single direction (**Figure 7b**). The "front" of a surface is determined by the order of the vertices, clockwise or counter-clockwise. The orientation is perpendicular to the plane of the surface, or it's normal.

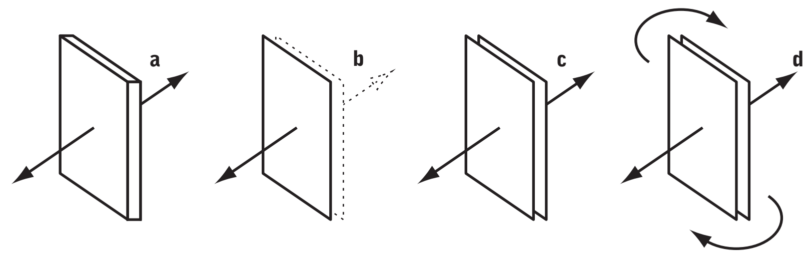


FIGURE 7:
a) a real world surface with thickness;

b) a typical 3d single-sided surface;

c) a 3d representation of a real-world surface with thickness;

d) a 3d representation using a duplicate surface facing the opposite direction.

In Animator, enabling **Cull Backfaces** causes any surface whose normal points opposite the direction of the camera, not to be displayed or rendered. Normals may be displayed by selecting **Show Normals** -> **Polygon Only** from the **Shade** ball above the display in the **Link Editor** for an object. Polygon orientation may be reversed by enabling **Reverse Normals** in the **Geometry** tab of the **Group Info** window for each object.

DOUBLE-SIDED GEOMETRY

Double-sided geometry is commonly simulated by creating a duplicate surface in the identical position but oriented in the opposite direction (**Figure 7c**). The Radiosity Engine does just that. When **Cull Backfaces** for a group is not enabled, a back side is created and offset from the front side by **Polygon Offset** in order to avoid potential visibility issues (**Figure 7d**). The effect is that of double-sided geometry.

Orientation is of particular importance with light source surfaces. If a light source faces away from the rest of the scene, none of the energy will reach the scene and the radiosity solution will appear black.

The more geometry the Radiosity Engine must work with, the longer it will take to calculate a reasonable solution. Whenever possible, use objects intentionally modelled as single-sided surfaces, and enable **Cull Backfaces** *for each group and light source included in the radiosity scene.*

SUBDIVISION

The Radiosity Engine is very fast at subdivision. Subdividing geometry by hand, before submitting it to the Engine rarely result in higher quality solutions, and often increases solution times. There are exceptions, however.

Animator and Camera typically handle quadrangles and triangles, although they both accept points, lines, and complex polygons in certain situations. Upon import, complex polygons with more than four edges are tessellated into triangles. The triangulation algorithm favors a low-polygon-count and speed over other considerations such as polygon aspect-ratio.

One strength of Universe is the ability to handle radically varied and otherwise “unfriendly” polygons. The Radiosity Engine, however, relies on surfaces with surface-area, discarding points, lines, and non-tessellated complex polygons, usually with little impact on the appearance of the radiosity solution.

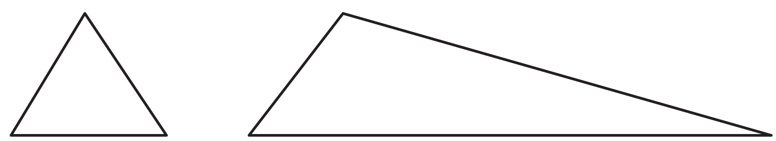


FIGURE 8:
Low-aspect-ratio (left) and high-aspect-ratio (right) triangles.

A polygon wider than it is taller, or vice-versa, is regarded as a high-aspect-ratio polygon (**Figure** 8). These polygons are usually subdivided by the Radiosity Engine into even higher-aspect-ratio polygons (**Figure 9**).

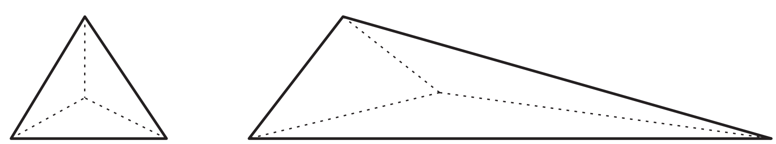


FIGURE 9:
The same triangles when subdivided.

High-aspect-ratio triangles, in particular, can result in inaccurate radiosity solutions and visual artifacts in the solution mesh. Not every high-aspect-ratio polygon will cause problems, but when the aspect-ratio is significant, issues may develop.

The radiosity of a polygon is determined by taking the average of the radiosity values at each vertex. Take the extreme example of a single quadrangle several times wider than it is tall, with a very different radiosity value at each vertex (**Figure 10a**). The entire quadrangle must share the average radiosity value (**Figure 10b**). Dividing the quadrangle in half results in average radiosity values for each, that better approximate the original quadrangle (**Figure 10c**). The closer to a one-to-one ratio, the better the approximation, however, only extreme cases require manual intervention

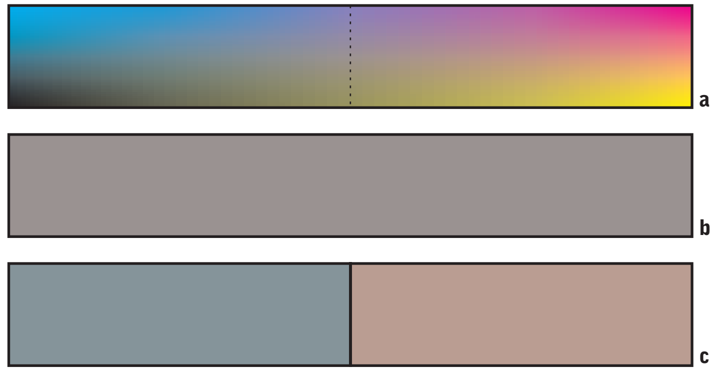


FIGURE 10:
a) a quadrangle with a different radiosity value at each vertex;

b) the same quadrangle assigned the average of the vertex radiosity values;

c) subdivisions of the quadrangle with corresponding average vertex radiosity values.

(average colors are simulated)

High-aspect-ratio polygons typically occur when Animator or another application tessellates a complex polygon with more than four vertices. Carefully subdividing these polygons before passing them to the tessellating application often limits the number of high-aspect-ratio polygons created.

SPECIAL CASES

Polygons with multiple contours pose a common problem. Take a single polygon with a hole cut into the center (**Figure 11**). The polygon has two contours—one inside and one outside—for a total of eight points and eight edges. The polygon will be tessellated as it is brought into Animator.

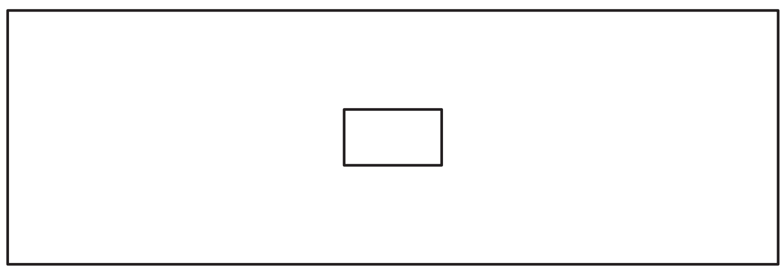


FIGURE 11:
The original, multi-contour polygon.

One might expect the polygon to neatly tessellate into four quadrangles, each roughly a square—one on each side, above and below the hole, but that is rarely the case. Most often, the polygon would tessellate into eight triangles—some with high-aspect-ratios (**Figure 12**). Triangles in the upper-right and lower-left are highly suspect. If the radiosity solution contains artifacts in those

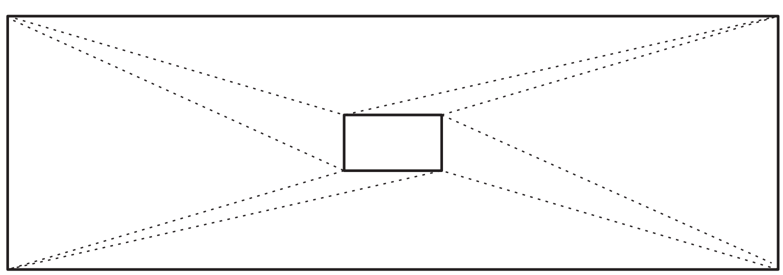


FIGURE 12:
The same polygon when tessellated automatically by Animator on import.

regions, it would be worthwhile to manually subdivide the polygon into the ideal—four quadrangles (**Figure 13**). Not only is the resulting polygon count cut in half, the radiosity solution is likely to improve as well.

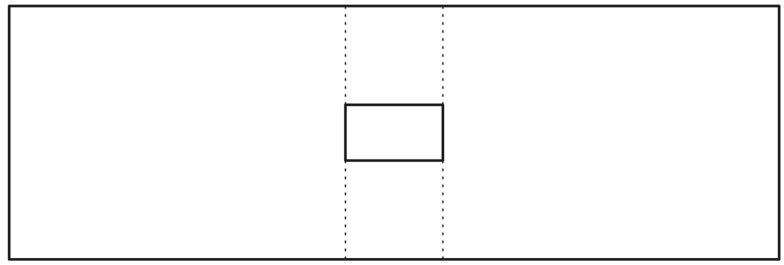


FIGURE 13:
The same polygon after manual subdivision. Note that Animator performs no further tessellation.

Any polygon with more than four vertices is subject to tessellation and may very well result in high-aspect-ratio polygons. Polygons with holes, archways, and curves are particularly suspect. Simple, careful remediations can make a big difference in the resulting radiosity solution.

LIGHT SIZE EXPERIMENTS

We create a simple experiment to gain a better understanding how the size and intensity of a light source affects the radiosity solution. We pay particular attention to the scale of the models so the knowledge we gain will be applicable or easily scaled to future projects.

In our modelling application, we create a five-meter-square cube and remove the front face, allowing us to see the inside of the cube. We create a one-meter square near the upper-left-front corner of the cube, and offset it slightly outward from the cube (**Figure 14**). We color the left-wall—red, the right-wall—blue, and all other surfaces of the cube—light gray. The square, acting as the light source for the scene, is colored white.

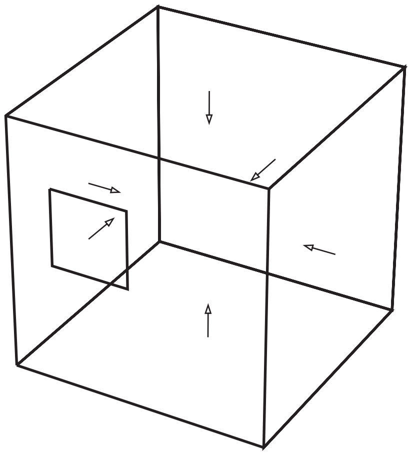


FIGURE 14.

We want to submit as few polygons as possible to the Radiosity Engine, so we ensure the normals of the box and square point inward, and export those normals with the model. When we import the model into Animator, we are careful not to check the **Recalculate Normals** option.

We enable **Cull Backfaces** in each object's **Group Info** dialog, set the **Radiosity Intensity** of the square, or light plane, to 50.0 and add it to the **Lights** list; then add the cube surfaces to the Objects list. In the **Radiosity Settings** window, we press the **Default** button, change the **Method** to Progressive, and hit **Go**.

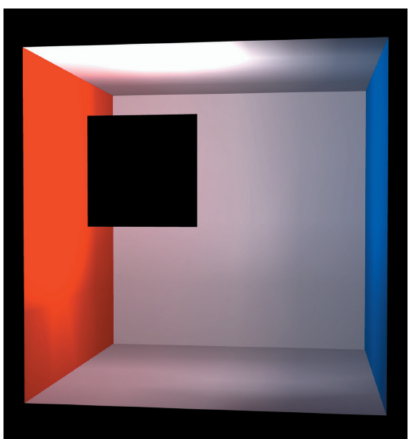


FIGURE 15.

When the Radiosity Engine finishes, we press the **Update** button to load the solution back into Animator (**Figure 15**). Notice the apparent brightness of the ceiling and red wall, as well as the subtle color bleed on the floor, back wall, and ceiling.

Our first experiment is to scale the light plane to 0.5 (XYZ), and recalculate the radiosity solution. As expected, the result is quite dark (**Figure 16**)—the reason is simple. The original one-meter-square light plane emitted 50 watts per meter, or 50 watts of energy into the scene, while the quarter-meter-square light plane, still emitting 50 watts per meter, contributes only 12.5 watts to the scene.

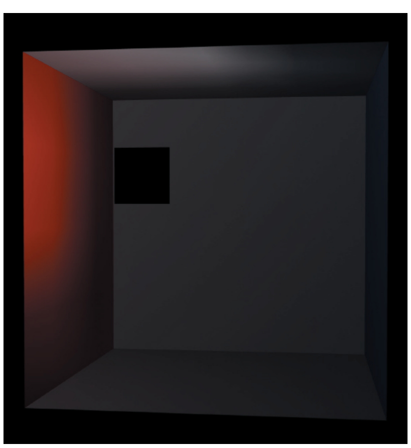
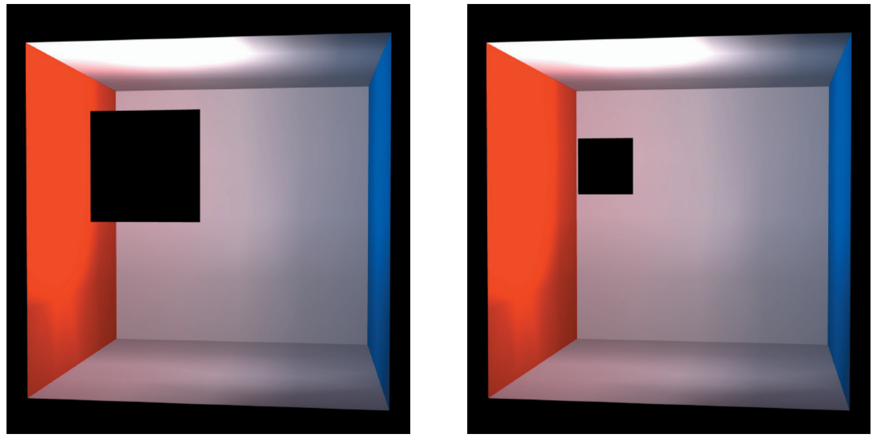


FIGURE 16.

In order for the smaller light plane to contribute the same amount of energy to the scene, we must increase the **Radiosity Intensity** by the inverse of the scale of the object or, a factor of 4.0, to 200.0. When we recalculate the solution, we get an image nearly identical to our original image (**Figure 17**).

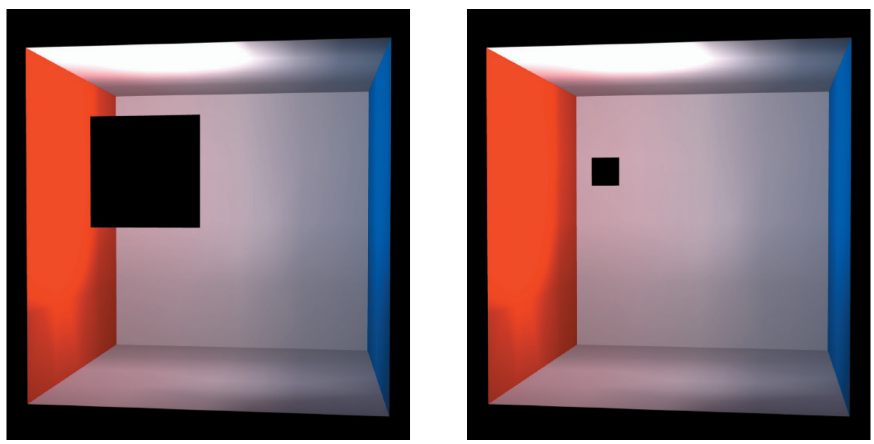


a b

FIGURE 17:
a) the original scene with a 1 meter-square light plane;

b) the same scene with a 0.5 meter-square light plane.

Our second experiment is to reduce the size of the light plane and increase it's intensity, to determine how the resulting solution compares with the original solution. We reduce the scale of the light plane to 0.25 (XYZ) and increase the **Radiosity Intensity** by the inverse of the change in scale, to 800.0, and recalculate the solution. The result is once again, remarkably similar to the original rendering (**Figure 18**).



a b

FIGURE 18:
a) the original scene with a 1 meter-square light plane;

b) the same scene with a 0.25 meter-square light plane.

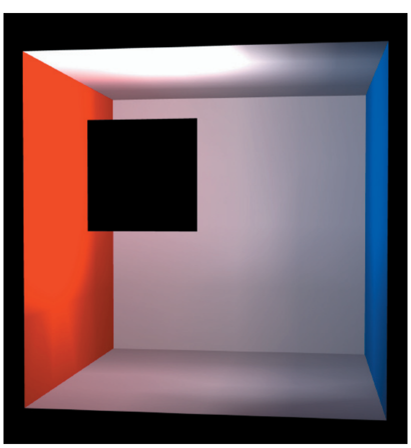


FIGURE 19.

As a final experiment, we scale the entire scene, including the light plane, without modifying any other settings. The results are nearly identical to the original (**Figure 19**).

The amount of energy reaching a surface is determined by calculating the form-factor between the source and receiver. Although the scale of the surfaces has been reduced, the distance between them has also been reduced, so the amount of energy transmitted per-square-meter is the same.

LIGHT SHAPE EXPERIMENTS

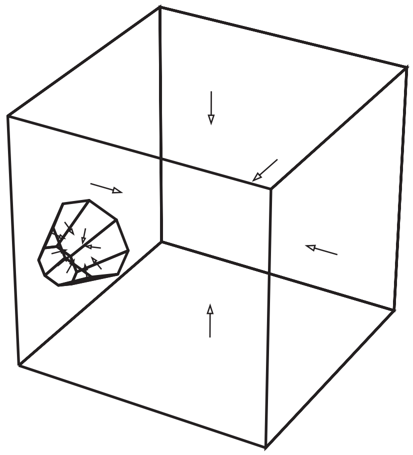


FIGURE 20.

Just as the size and intensity of a light source affects the radiosity solution, the shape of the light source and whether it has an reflector also affects the solution. We create a similar experiment, and expect the character of the light to be different not only because the emitting surface is smaller, but also because most energy in the scene will be bounced off the reflector. We also expect to lower the **Minimum Energy Level** to account for more reflected energy.

We use the same box model as in the previous experiment, but for the light source, we use a simple cone with a flattened end on the small side that will serve as the emitter (**Figure 20**). The remainder of the cone will serve as a reflector. Both emitter and reflector surfaces are colored white. We follow the same procedure as the previous experiment to ensure the normals are correct and the engine won't create double-sided surfaces. We set the **Radiosity Intensity** of the emitter to 100.0.

With default settings and the Progressive method selected, we calculate the solution. The result (**Figure 21a**) is very dark. The reason is probably two-fold. For one, we assumed the emitter surface was half-the-size of the previous light plane, when in fact, it is closer to a quarter the size, and the **Radiosity Intensity** for the emitter must be increased accordingly. For another, much of the energy bounces off the reflector on it's way into the box, so the **Minimum Energy Level** should be lowered.

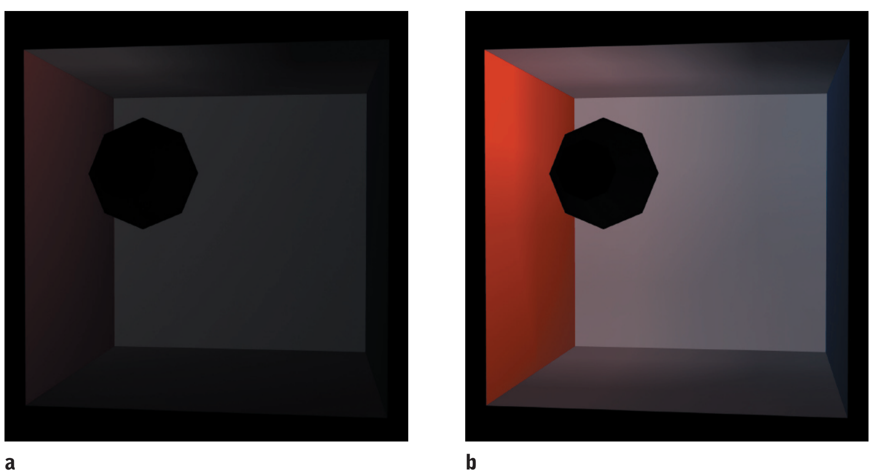
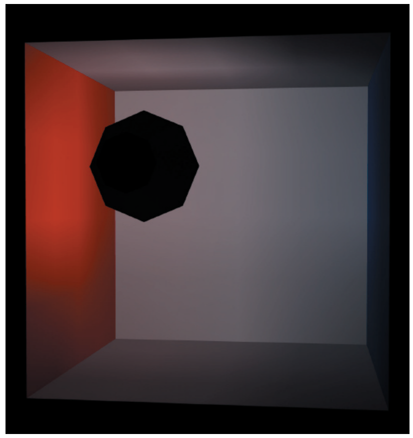


FIGURE 21:
a) the initial solution;

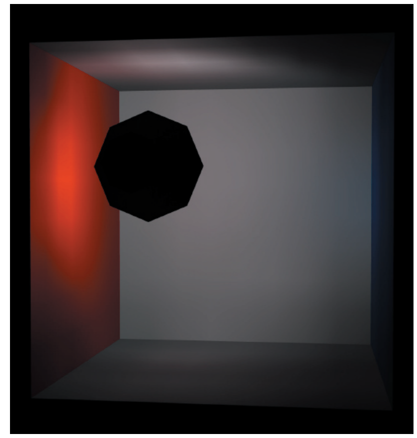
b) Radiosity Intensity increased to 250.0, and Minimum Energy Level decreased to 0.001.

We increase the **Radiosity Intensity** of the emitter to 250.0, reduce the **Minimum Energy Level** to 0.001, and recalculate the solution. The result is closer (**Figure 21b**). The light is diffused by the reflector, but the penumbra is not evident probably because the surfaces are not adequately subdivided.

We increase the **Source Density** to 4.0 and the **Receiver Density** to 8.0 and recalculate the solution. The resulting penumbra is finally evident as we had hoped, but the contrast is not as apparent as we would like (**Figure 22a**).



a



b

FIGURE 22:
a) Source Density set to 4.0 and the Receiver Density set to 8.0;

b) Source Density set to 8.0 and the Receiver Density set to 16.0.

Again, we increase the **Source Density** to 8.0, the **Receiver Density** to 16.0, and recalculate the solution once again. The contrast is apparent, but not bright enough yet (**Figure 22b**).

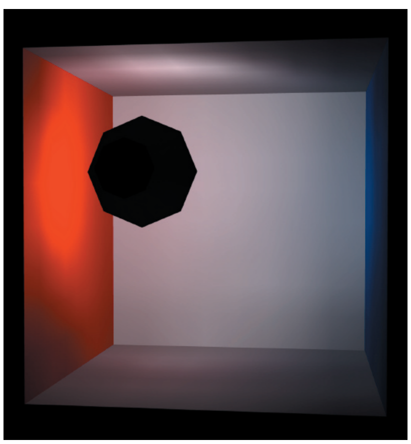


FIGURE 23.

In the previous experiment, we noted that increasing the **Radiosity Intensity** and decreasing the **Minimum Energy Level** both helped in a similar situation, so we do both. The **Radiosity Intensity** is increased to 350.0, and the **Minimum Energy Level** is decreased to 0.0001. When the solution is recalculated, we see the results we were hoping to achieve (**Figure 23**).

In the future, when we use light sources with reflectors, we will be sure to increase the **Radiosity Intensity** *of the light source, decrease the* **Minimum Energy Level** *for the scene, and consider increasing the* **Source Density** *and* **Receiver Density**.