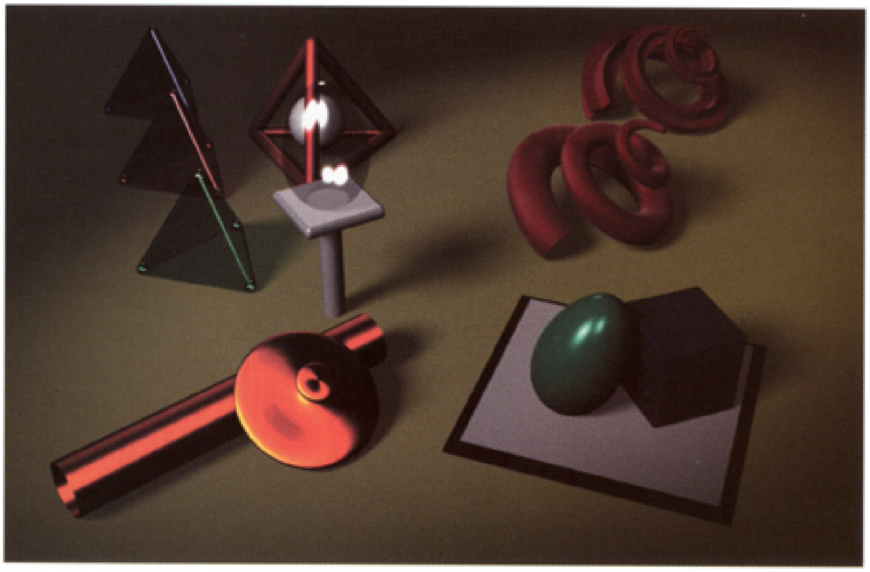


PENUMBRAE In the real world, shadows often have soft edges because light sources radiate from more than a single point. In this image, such an area light source has been simulated. Shadows are now softer. The shadow from the table, for example, now varies continuously from light to dark, a transition known as a penumbra.

sources in a scene, only those that reach the viewpoint, either directly or through reflection, need be considered. The vast majority of light rays are inconsequential because they never reach the image plane. Ray tracing and ray casting, therefore, make an intractable problem more manageable by following rays from the viewpoint through each pixel into the scene and ultimately to the light sources.

Starting from the viewpoint and passing through a single pixel in the image plane, a ray will eventually encounter either a surface or a light source. If it is a surface, the question arises as to whether that surface point is illuminated. This question can be answered by attempting to draw a straight line between the surface point and all of the points in the scene that are defined as light sources. These lines are called *shadow rays* because they determine whether or not the surface point is in shadow. If an uninterrupted line can be drawn from the surface point to a light source, then the point is not in shadow. If, on the other hand, the shadow ray intersects another surface on its way to the light source, the surface point will be in shadow with respect to that light source.

For each light source that is not shadowed, the computer utilizes a reflection model, as noted earlier, to determine the light from that light source that is scattered to the viewpoint. This color contribution is, along with all other light source contributions, added to the pixel color.

GLOBAL ILLUMINATION The human visual system is extremely sensitive to subtle qualities of illumination that can be simulated by lighting models that include interreflections. Such lighting models are called *global* and are distinguished from lighting models that simulate only local interactions of light and surface. In this image, highly specular surfaces such as the glossy tile and the egg (lower right) can be seen to reflect their environment, and light refracts through transparent objects such as the egg on the table and the tetrahedra.

Computer graphics would be a far simpler discipline if we could stop right here, and, in fact, that is what we do with ray casting and other techniques that use local illumination models. Modeling interreflections, although it requires the additional hard work of ray tracing or other global illumination models, is worthwhile, however, because we can satisfy our perceptual hunger for verisimilitude.

Computer scientists have developed various ray-tracing algorithms to model light from interreflections. In its original form, ray tracing assumes that the initial ray from the viewpoint into the scene may be a mirrorlike reflection. Euclid, in fact, established the principle that when light reflects from a perfectly smooth surface, "the angle of incidence equals the angle of reflection." When a light ray encounters a surface, the angle between the surface and the direction of approach of the ray will be equal to the angle between the surface and the reflected ray. Ray tracing, in this case, follows the presumed incident ray to determine whether it encounters any surfaces. If it does encounter a surface, that surface, if illuminated, will function as a light source and will illuminate the original surface point that defines the color of the pixel.

Of course the contribution of the second surface point will depend upon the interaction of all of the illuminants and its own surface reflectance. The second surface point therefore must be treated exactly as the first surface point in the sense that shadow rays and incident rays will define the illumination affecting its apparent color. The incident ray from the second surface point may encounter a third surface that will need to be treated as a third surface point that is influencing the second and the first. In this sense, ray tracing is recursive: each successive surface point becomes a viewpoint in relation to the subsequent surface point.

Our ray-traced world is almost complete. If we were to calculate images at this point, however, windows would be opaque, water would appear as mercury, and a cup of tea might resemble wood. The procedures we have described so far do not account for the fact that some materials are transparent or translucent. To model these properties of objects, ray tracing generally assumes that the initial ray from the viewpoint may also be *refracted*. The index of refraction is used to compute the deflection of a light ray both as it enters the transparent material and as it exits the other side. This refracted ray then is recursively traced into the environment in the same manner as reflected rays.

Ray tracing simulates reflection and refraction with great accuracy and, as a result, produces highly realistic images. For example, if a scene contains a magnifying glass, the index of refraction and curvature of the lens can be specified to the computer. The ray tracing calculations will then render objects seen through the glass as magnified. Even rainbows and mirages can be reproduced by modeling the refraction of the water droplets or heated air that give rise to these phenomena in nature.

Ray tracing, in its original form, assumes the specular reflection, the mirrorlike property of surfaces, to be extremely important and, in fact, follows only the mirrorlike reflections from surface to surface. Ray tracing thus treats diffuse reflection locally; it does not attempt to model diffuse interreflection. The computational demands of modeling all of the interreflecting diffuse rays can be excessive; however computer scientists have extended ray tracing to model these diffuse rays. Instead of recursively tracing only one or two light rays from a surface point into its environment, some models follow hundreds or thousands of rays of light. Sophisticated algorithms have been developed to track only those light rays that contribute significantly to pixel color. These techniques result in images with an even more faithful reproduction of the environment of light.

Simulating Light Direct comparison of the final six images in the series illustrates the high degree of control that the algorithm designer has over properties of light in the computer. Of course, there is no actual separable light and surface in these rendered images; they are simply patterns of colors on the printed page. Nonetheless, each rendering algorithm, each image, will trigger your visual system to synthesize complementary illumination and surface worlds.

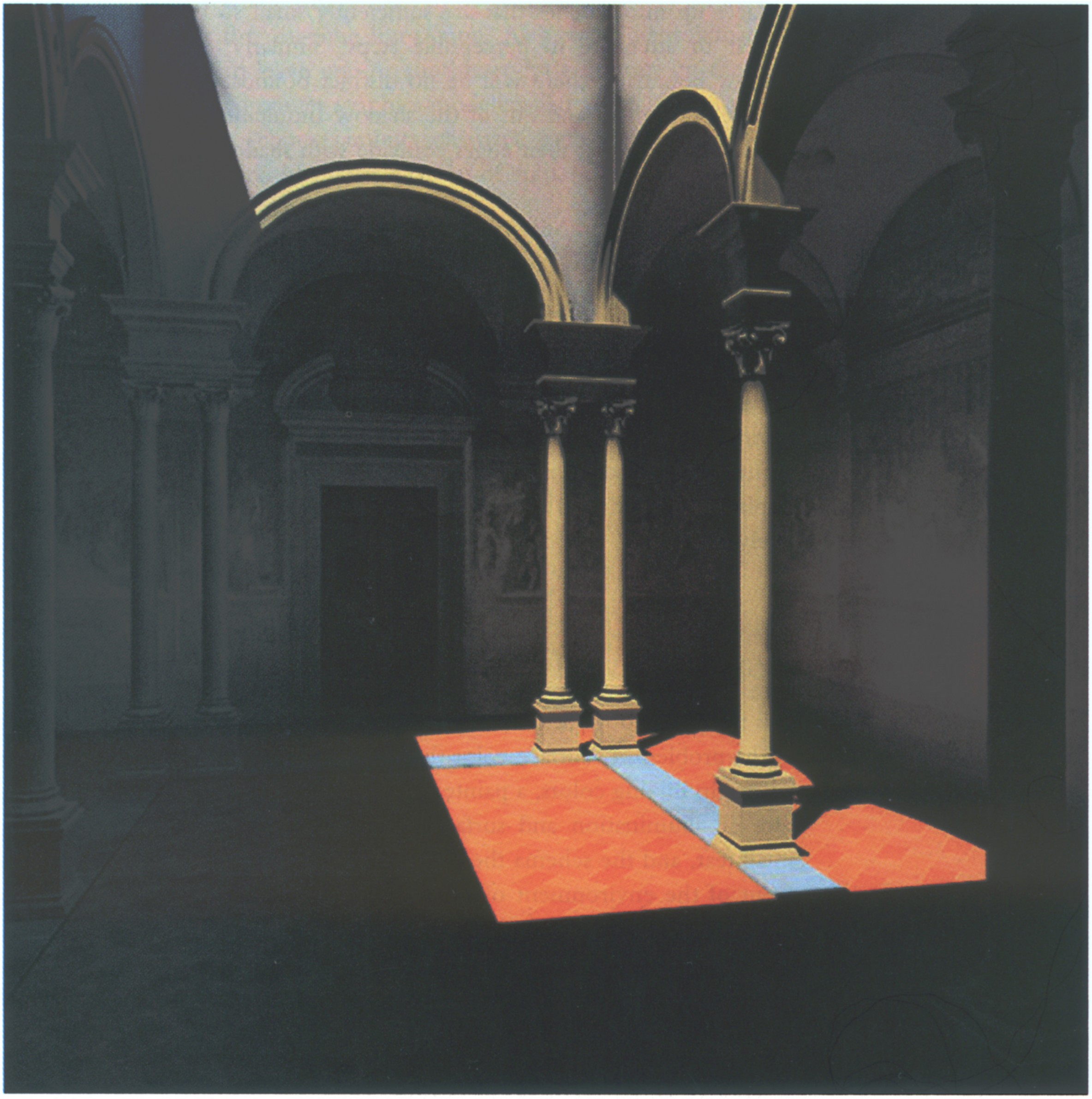
Radiosity

For many years computer scientists struggled to come to terms with the diffuse quality of natural light. As noted earlier, interreflections guarantee there are rarely any completely dark areas in a room even if there is only one light. Radiosity is the computer scientist's preliminary effort to create a warm and diffuse illumination environment.

It is perhaps fitting that the concepts used to "warm up" computer graphics renderings are, in fact, derived from thermal engineering. Radiosity is based upon techniques traditionally used to solve energy balance problems. In furnace design, for example, an engineer typically needs to determine how energy will flow among components. The rate at which heat leaves a surface, the flow of energy per unit time, is equal to the rate at which that surface radiates heat plus the rate at which it reflects heat from other surfaces. In the normal course of events, equilibrium is reached when, for any two surfaces, the amount of energy flowing in one direction is equal to that flowing in the other. Indeed, if they are not equal, then the surface transferring more heat will lose energy to the other surface until the rate of transfer equalizes.

In radiosity, light is modeled in the same manner. All of the surfaces in the scene are divided into regular patches, each of which is considered to be both an emitter and reflector of light energy. The rate at which each patch radiates light is assumed to be equal to the rate of emission plus the rate of reflection from all other patches. Like ray tracing, radiosity is recursive. The model may begin with the light sources and transfer light energy to all of the surfaces in the scene. Each of the surfaces is then treated as a light source that radiates to all of the other surfaces in the scene, and the process continues, eventually reaching an equilibrium, or steady state.

A critical aspect of a radiosity computation is the manner in which the surfaces are broken into discrete patches. An entire wall, a desktop, or a floor could be regarded as a single patch that emits and reflects light. Although the rendering computation would be tremendously simplified by such an assumption, it would not produce a satisfactory visual result. For example, a desk lamp will cast more light on areas near the lamp than on regions of the desktop that are farther away. If the desktop is treated as a single patch, however, it will be rendered with a single value for its light energy. To achieve a more realistic appearance, it is necessary to break the



Radiosity In the natural world, the reflection of light from surface to surface creates a subtle illumination environment that is difficult to model with a computer. One technique for modeling interreflections is radiosity. Radiosity simplifies the computation by assuming that all surfaces are diffuse reflectors. In this synthetic radiosity image, the sunlit patch of bricks illuminates the walls and ceilings of this courtyard with a reddish light. Note that if the computer had omitted interreflections, most of this scene would be black.

surfaces up into patches that are sufficiently small so that light does not appear to attenuate in perceptible steps. Similarly, if a patch spans a shadow boundary, there will be no distinct boundary because the patch cannot be both in and out of the shadow simultaneously. Patches often are created such that their edges coincide with shadow edges to avoid this problem.

The original radiosity methods assume that all the surfaces in a scene are Lambertian diffuse reflectors and, consequently, that the intensity of light reflected from a surface is independent of the viewing direction. This assumption significantly simplifies the mathematical equations. Additionally, the Lambertian assumption eliminates the requirement of knowing the viewpoint when calculating the shading. In essence, a shaded environment is computed only once and can be rendered from any viewpoint. Since shading is not recalculated as the viewpoint changes, the display computation is fast, and radiosity solutions can be viewed interactively.

LIGHT AND VISUAL COMPUTING

It is tempting to think that the computer graphics modeler can bypass all of the problems of surface color that afflict the painter interested in naturalism. In mixing colors, a painter is deciding simultaneously how to capture all illumination and surface reflectance contributions. This is no small feat, and the schema the artist uses will create a characteristic treatment of the environment of light. An inspection of a painting by John Constable or Rembrandt demonstrates that the pigments used to represent objects are hardly the actual surface colors. Constable used dozens of colors to depict, for example, the leaves on a tree, which technically speaking are all the same color before they interact with light in the scene.

Unfortunately the computer scientist is rarely able to "model the physics." The process is just too complicated. Every light source with its particular spectral power distribution and shape, every surface with its particular orientation, roughness, and surface reflectance, and all interreflections would need to be modeled. The computational demands of modeling these properties of light and surface usually preclude this approach, particularly when images need to be generated quickly for an animated sequence or for real-time interaction. As a result, approximations are made that tend to leave one or another stylistic imprint on the rendered image. Thus, computer graphics modelers and artists share a similar dilemma, the need to choose among schemata, each with strengths and limitations for simulating particular qualities of light.