Chapter 1

# Introduction

## 1.1. Computer Graphics

Computer graphics has grown at an astounding rate over the last three decades. In the 1970s, frame-buffers capable of displaying digital images were rare and expensive. These days it is rare that we see an image or video sequence that has not been manipulated in some way by a computer, and computer graphics has become ubiquitous; it underpins the most commonly used computer interfaces, drives many of the applications, and is fundamental to almost all computer-based entertainment.

In three-dimensional computer graphics, we are concerned with producing an image or an animation of some 3D world, that may be either purely imaginary, or may be trying to mimic some real-world counterpart. This process features a number of stages:

**Modelling.** Geometrical representations of the scene are produced, consisting of sets of points, faces, and other geometrical entities. These representations can be constructed by hand, modelled mathematically, or even scanned from real-world objects with a laser range scanner.

**Animation.** The various models are composed into a *scene*, and then (usually individually) animated. That is, the position and motion of each object in the scene, including the camera, is specified over time.

**Lighting.** Lighting a scene involves the selection of reflectances, materials and texture maps for the models in the scene, and the positioning of virtual light sources to illuminate the scene. (The former could arguably fall under the “modelling” category, but here I follow the conventions of the animation industry.)

**Rendering.** The rendering stage, the one I am primarily concerned with, is responsible for taking the three dimensional scene specification produced by the previous three stages, and producing the two dimensional representation of it as seen by the camera. This can be separated into two sub-problems; determining which parts of the scene correspond to which parts of the output image (perspective projection, depth sorting), and determining how scene surfaces look after the effects of lighting and reflection have been taken into account (shading).

In *photorealistic* computer graphics, we attempt to model the real world, and to produce a rendering of the result that matches the corresponding photograph of the real world. In *physically-based* rendering (and indeed animation), we attempt to reproduce the physical processes involved in this.

In what follows, we are primarily interested in the photorealistic, physically-based rendering problem, and, specifically, that part of the problem known as *global illumination*.

## 1.2. Global Illumination

We can define global illumination simply as, the propagation of light from light sources throughout the environment defined by our scene. This is in contrast to local illumination, which only considers a light source, and those surfaces it lights directly. Global illumination applies this local operation repeatedly, treating lit surfaces as light sources in turn, allowing multiple bounces of light through the scene and eventually to the camera. Consider **Figure 1**, for instance; the difference between the globally-lit scene with shadows and the direct-lit unshadowed scene more typically seen in interactive graphics is almost as great as the difference between direct lighting and no lighting.

The simulation of global illumination is one of the greatest challenges of realistic image synthesis; it requires modelling the transfer of light through the environment being rendered, including the effects of interreflection between objects. Currently there are two classes of methods for calculating global illumination; Monte Carlo-based and Finite Element-based. The former is usually loosely referred to as *ray tracing* and the latter as *radiosity*. Broadly speaking, Monte-Carlo based methods are better suited to simulating the interreflection between highly

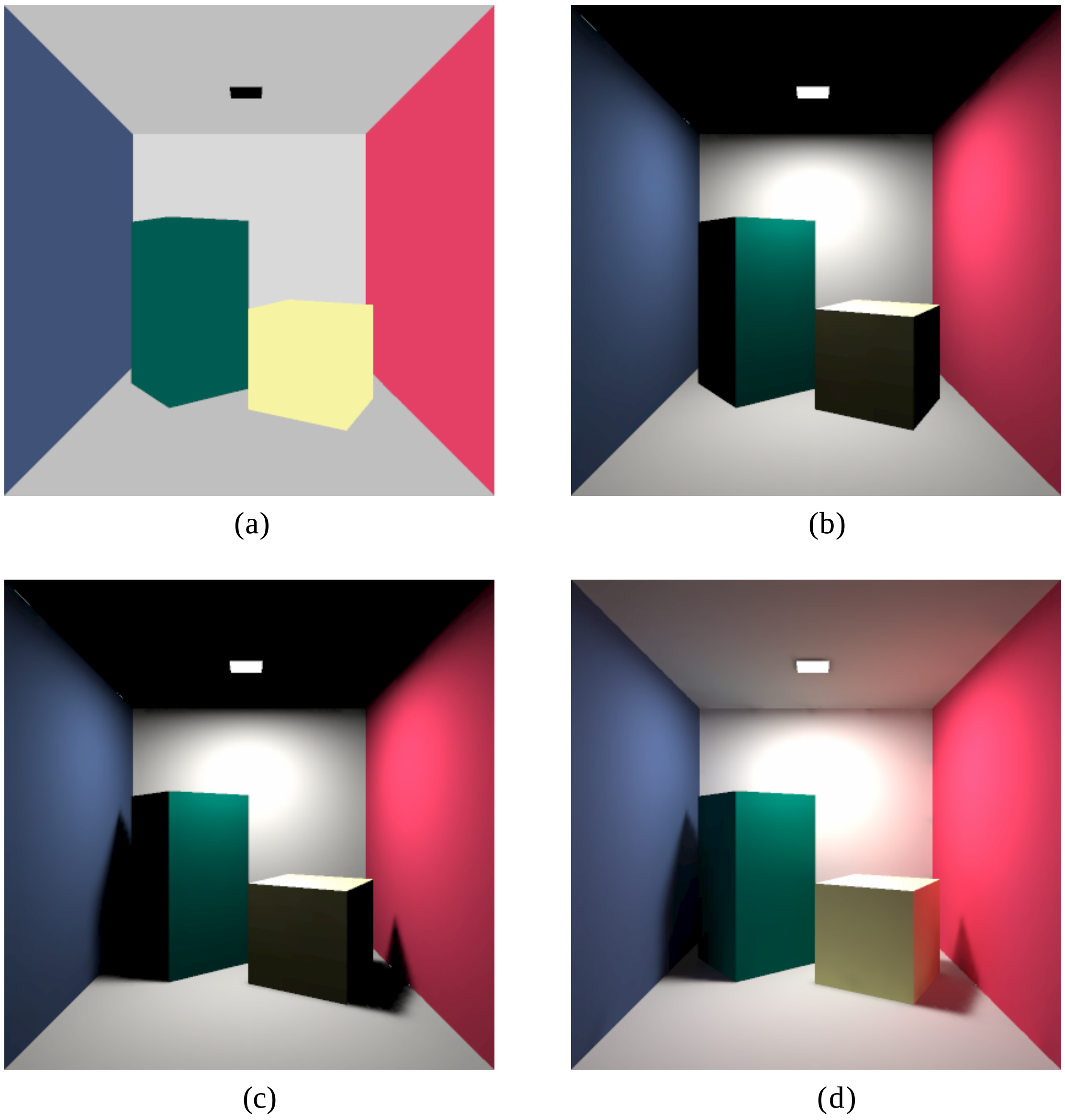

**Figure 1:** The need for global illumination. A variant of the famous "Cornell Box" scene is shown under several different lighting conditions. (a) No lighting; flat shading only. (b) Direct lighting only; notice the completely black ceiling and cube sides. (c) Direct lighting with shadows. (d) A complete solution for propagated light in the box. Notice the illuminated ceiling, and the cube's sides are now lit by the red side wall.

specular (mirror-like) surfaces, and finite element methods to diffuse (matte-like) surfaces (see **Figure 2**).

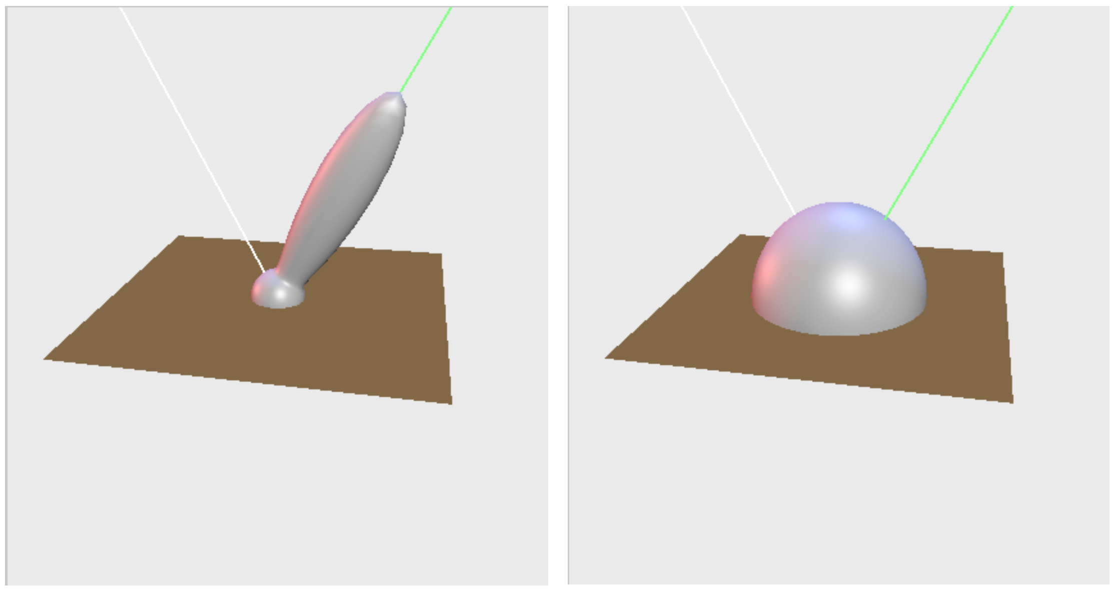

**Figure 2:** Different reflectance characteristics. Shown are polar plots of outgoing radiance, for light incident along the white ray. Left, a specular surface reflects light directionally, usually close to the angle of reflection. Right, a diffuse surface reradiates light equally in all directions.

### 1.2.1. Radiosity Methods

The basic radiosity algorithm is limited to simulating scenes with diffuse surfaces; this is both a strength and a weakness. It is a strength because any diffuse-only global illumination solution is *view independent*. This means that the output of most radiosity methods, instead of being a single image, is illuminated geometry. This geometry can then be quickly rendered from any viewpoint, without repeating any global illumination calculations; indeed, without doing any lighting or shading calculations at all. On the other hand, the weakness of radiosity as a global illumination method is precisely that it ignores any specular reflection in the scene. Such effects must be added afterwards, with a ray-tracing pass. Even then, this does not give us a full global illumination solution, as there is no chance for mixing specular and diffuse light bounces.

Still, radiosity methods that produce illuminated geometry are worth pursuing for a number of reasons. The interreflection of diffuse light is an important phenomenon in realistic image synthesis, particularly for indoor scenes, where a significant fraction of the light illuminating a surface comes not directly from a light source, but indirectly, by bouncing off one or more other surfaces. Although some ray-tracing methods that cache diffuse samples [Ward88] are currently faster than those radiosity methods used in industry, the hierarchical nature of state-of-the-art radiosity systems promises better performance, if only the dependence of those systems on a scene's geometrical complexity, as opposed to its inherent illumination complexity, can be removed. Finally, as well as illuminated geometry, many radiosity methods are capable of producing a global representation for the light flow in a scene. This is useful both as a guide to a ray-tracing postprocess, and as an aid to initialising more complicated Monte Carlo-based global illumination simulations.

## 1.2.2. Applications

Radiosity's areas of application include the following.

### Architectural and Lighting Design

One of the major applications of radiosity methods is the prototyping of various architectural and lighting designs. A big advantage of a general radiosity solution is that its output is illuminated geometry, suitable for interactive walkthroughs. Such walkthroughs can be used to gain a more realistic sense of space and light in an architectural design before that design is actually implemented. Lighting fixtures can be rearranged, interior walls relocated, materials changed, and the results evaluated.

Radiosity methods have been used extensively in these areas, from building and luminaire design to theatre lighting design; an example is shown in **Figure 3**.

### Virtual Sets

It is becoming commonplace to build "virtual sets" for television shows, such as sports shows, game shows, or news broadcasts. These sets must be combined with live footage in real-time, but a high level of visual realism can still be

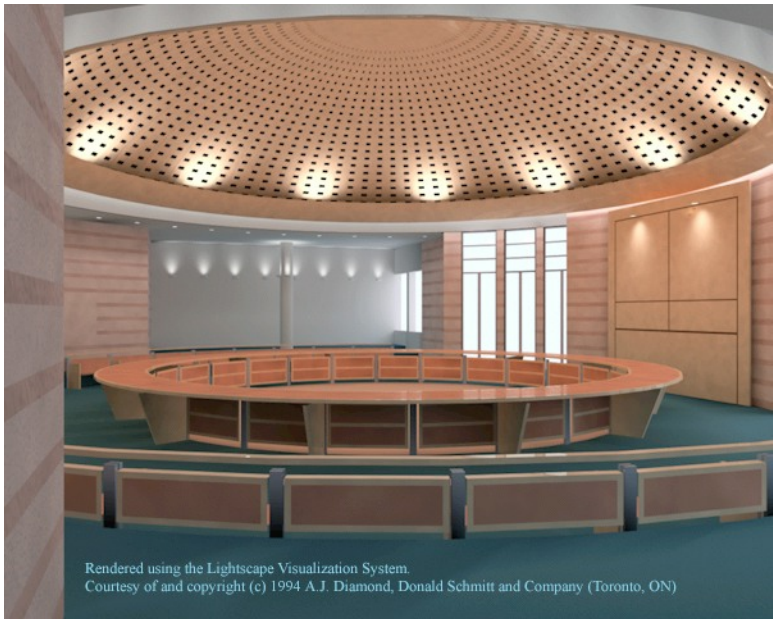
Rendered using the Lightscape Visualization System.
Courtesy of and copyright (c) 1994 A.J. Diamond, Donald Schmitt and Company (Toronto, ON)

**Figure 3:** Radiosity used for architectural and lighting design. Note the soft lighting and the large number of light sources. (Picture from Lightscape.)

attained by mixing a precalculated diffuse-light solution with hardware lighting for (view-dependent) specular effects.

### 3D Games

Today, convincing realism is an important factor for the success of a computer game title. More and more games use techniques from realistic image synthesis such as precomputed global illumination, shadow maps, as well as advanced mirror and refraction effects to increase the illusion of reality. An example is shown in **Figure 4**. The game level shown is reminiscent of some office scenes that have appeared in global illumination papers, although the low polygon count and obvious discretisation artifacts make clear its interactive nature.

Arguably, this area is a more promising one for global illumination algorithms than computer animation for movies or television. The level-of-detail

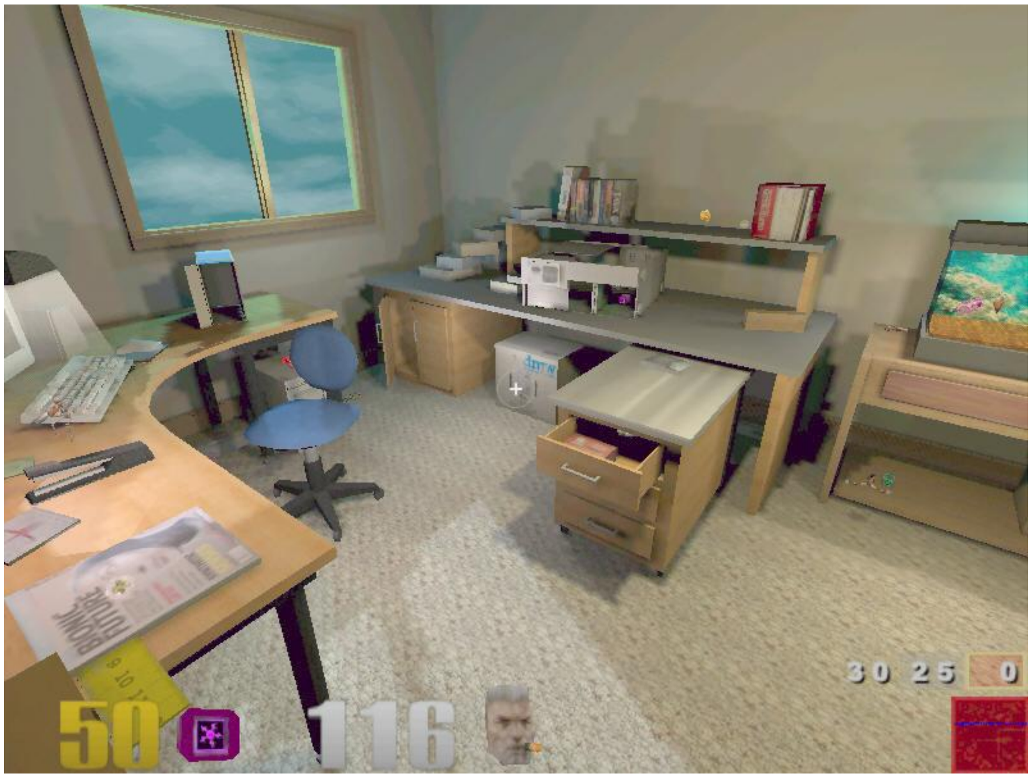

**Figure 4:** A screen-shot from the interactive 3D game, Quake. The level shown in this picture features precomputed global illumination effects, in spite of needing to be rendered at interactive rates in graphics hardware. (Picture from ID Software.)

trade-off in a typical piece of such computer-generated animation is tilted more towards detail and less towards interactive rendering times, making the barrier to entry for such algorithms much higher. This may not make sense at first, as surely, without the need to be interactive, there is more time to amortize the cost of any illumination algorithm. Unfortunately, it is a rule in Computer Graphics that scene complexity expands to fill the rendering time available, and, as illumination algorithms scale at best linearly (and usually worse than linearly) with scene complexity, there is little time left over for such effects, except in small, limited situations.

### Sky Illumination and Outdoor Scenes

Radiosity methods can be very useful for outdoor sky illumination; the sky is essentially a large, diffusely-emitting (blue!) light source [Daub97]. While outdoor scenes are generally considered to feature mainly direct lighting from the sun, contributions from the sky are still significant in any scene featuring shadows, as is reflected light from building walls, or scenes from an overcast day. An example is shown in **Figure 5**.

## 1.3. Contents of Dissertation

This thesis is primarily concerned with making the use of finite-element methods for global illumination tractable for scenes containing large models. In the next chapter, I give a brief overview of the radiosity problem, the methods commonly used to solve it, and the hierarchical radiosity method that my work builds on. Chapter 3 then presents the basic face cluster radiosity method. It starts with some motivation, and presents the core of the basic algorithm; the use of face cluster hierarchies, and the use of vector irradiance.

Chapter 4 presents the theoretical justification and error analysis of the radiosity part of the algorithm. Chapter 5 presents some analysis of Garland's original face cluster construction algorithm, along with my modifications to it to improve its speed and memory use, as well as the hierarchies generated, and some examples of its use.

Details of my implementation of face cluster (and hierarchical) radiosity are presented in Chapter 6, and in Chapter 7 I present results for several variants on both a simple test scene, and a more realistic "museum scene".

Finally, we round off with conclusions in Chapter 8.

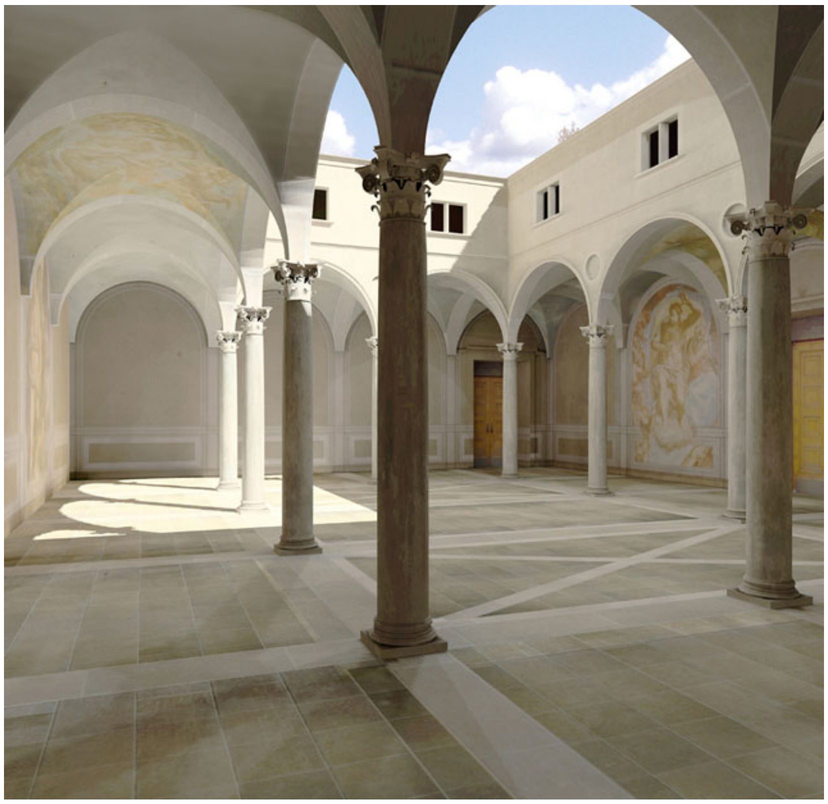

**Figure 5:** Radiosity used for outdoor scenes. Note that, without taking indirect lighting into account, the areas in shadow would be completely black. (Picture from Lightscape.)