Chapter 1

Introduction

Darkness is the absence of light. Shadow is the diminution of light. Primitive shadow is that which is attached to shaded bodies. Derived shadow is that which separates itself from shaded bodies and travels through the air. Repercussed shadow is that which is surrounded by an illuminated surface. The simple shadow is that which does not see any part of the light which causes it. The simple shadow commences in the line which parts it from the boundaries of the luminous bodies.

Leonardo da Vinci (1452 to 1519)

The study of the behavior of light and its effect in the surrounding environment has been, and is still nowadays, a challenging field of research. The above citation is an excerpt from Leonardo da Vinci's notebooks [Leonardo da Vinci45] where Leonardo has sketched his observations on the interaction of light and objects. He was one of the leading minds during the Italian Renaissance (1420-1600), in which artists started to observe the real world and came up with rules for applying light and shadow in a very natural way. The main stylistic element that was characteristic for that period was *chiaroscuro*[1]. The arrangement of light and dark regions in an image was extensively used to produce the illusion of depth in paintings. The rules of properly shading an object were defined by identifying different regions on the object and the surrounding environment. The impression of a glossy object was achieved by drawing highlights at that part of the object where the light source is most dominant, whereas the overall lit parts where drawn as the combination of object and light color, with decreasing intensity as the light source influence becomes less dominant. To emphasize the object's position in the scene, cast shadows are applied for the surrounding environment. Self-shadowing of an object was defined as a continuous color fade from a dark color (shadow) to black (core shadow). With more than one light source in the scene, an artist could further improve the three dimensional effect by adding reflected light to shadow and core shadow regions.

[1] from the italian works "chiaro" (clear or light) and "oscuro" (obscure or dark)

History in computer graphics made a quite similar way. In the early days, graphical output was mostly in wire frame mode and used for engineering applications, e.g., visualizing machine parts. During the 1960s, realism went a step further by having objects displayed as shaded solids, done with half-toning, pattern-based approaches similar to artistic pencil drawings techniques. This way depth and spatial relationship in computer generated images was enhanced. At the end of the sixties Arthur Appel presented his work on the shaded rendering of solids [Appel68]. This was probably one of the first publications on shadow techniques in the history of computer graphics. In this paper, different ways of shading solid objects, including shadows, were presented: A brute force, point by point shading approach but also optimizations, e.g., detecting contour edges relevant for shadow boundaries. With Appel's work the importance of shadows in computer generated images was emphasized: Shadows are important visual cues that make the spatial relationship of objects easier to understand.

In these early days of computer graphics, rendering an image took from several minutes up to hours or days, even for very trivial scenes. Algorithms like ray tracing or radiosity pushed realism to photo-realistic quality, but computation time was so enormous that no interaction was possible.

This changed immediately by the introduction of hardware-accelerated rendering. Companies like SGI developed powerful systems that had dedicated hardware support for lighting, texturing, and hidden surface removal. Although the amount of realism produced by hardware-accelerated rendering could not compete with offline methods of that time, the systems were able to produce images in a fraction of a second. This opened a new branch for computer graphics, called real-time rendering.

Using rasterization based graphics hardware involves a trade-off between quality and speed. One key concept that allows extreme parallel processing and fast rendering is the restriction to local illumination and simple reflection models. In contrast to global illumination, the appearance of an object depends only on a small number of parameters, e.g. the light source's position and direction, viewer, and surface material. Adding shadows to this type of architecture is difficult. Polygons are processed independently of each other, but shadow computation is based on the global arrangements of objects and light sources in a scene.

Two kind of algorithms are suitable to solve the global shadow task. One are the so called off-line methods which classify the shadow regions in a preprocessing step and only use the graphics hardware to visualize the final result. A popular method in this category are light maps generated from a global illumination system which are then applied as surface texture maps. In terms of quality this approach is one of the most accurate shadow techniques, if texture resolution is sufficiently high enough so that rasterization artifacts can not be seen. However, computing an accurate global illumination solution is an expensive, time-consuming task, so in terms of speed this approach is only suitable for static environments, e.g. architectural visualization where interaction is completely restricted to the change of viewing parameters.

For dynamic environments with changing objects and illumination, global shadows need to be updated at ideally the same rate as the display device. Interactive and real-time applications require updates of about 10 up to more then 60 frames per second, which puts a heavy load on the shadow computation. Furthermore, there are also special applications, such as virtual studios or simulation systems, which require a fixed output frame rate, regardless of the scene's complexity or current viewing parameters.

Shadow algorithms therefore have to trade-off between quality and speed: Shadows need to be visually pleasing but also computed at a fraction of a second, even for complex environments.

In addition to the requirement of handling fully dynamic environments efficiently we further specify the characteristic of the shadow algorithms discussed in this thesis by three more terms:

Realistic

As already mentioned before, the quality of the resulting shadows is among the most important properties of any shadow technique. The generated shadows should always correspond to the spatial relationship of objects in the scene and the type of light source being used. Especially in the context of real-time rendering, photorealistic, physically correct shadows are often not possible due to computation time restrictions. For most applications it is sufficient to have a shadow approximation that looks visually pleasing.

General

A shadow algorithm should make only few to no assumptions about the scene description itself or other parameters (camera, animation paths, etc.). Shadows should be of the same quality, putting no restrictions to the processed scene.

Hardware-accelerated but flexible

The shadow technique should make use of as much hardware-assistance as possible. This is not only a requirement in order to be able to handle very complex scenes at reasonable frame rates, but also in terms of resource management. Since the CPU is more and more dedicated to non-graphic work, e.g. sound processing or numerical simulation, CPU-based computation must to be minimized. In addition to this, the algorithm should be easy to integrate into an existing interactive rendering system. The cost of implementation and required changes to the core functionality of the system should therefore be minimized.

Designing suitable algorithms is a process of finding a reasonable trade-off between all these criteria. A fully hardware-accelerated method for example may only work for a specific class of scene objects, while a more hybrid approach, that involves much more CPU based computation, would support all types of objects.

An application developer should therefore carefully specify which requirements are most important and which can be restricted to special cases. Such a

special case could be the type of light sources that are supported, e.g. an algorithm that computes shadows for spot lights with a limited cut-off angle can be implemented quite efficiently in contrast to shadow algorithms that support more general light source types.

The algorithms proposed in this thesis are all examples of emphasizing one or more criteria, as in the light source type example. Given the requirements the resulting shadow quality is then often dominated by the clever use of available resources, like CPU time or hardware-capabilities.

1.1 Contribution and Overview

The reminder of this thesis is organized as follows. The next chapter serve as an introduction to digital image synthesis and hardware-accelerated rendering techniques. This includes basic methods of computing visible surfaces as well as an introduction to reflection models, light sources, and shadows. And the end of the chapter we will also show how these concepts are realized on modern graphics hardware architectures.

The third chapter focuses on related work in the context of shadow algorithms. Here we will describe the major shadow algorithms that are suitable for efficient shadow computation in a real-time environment. Since there exists a huge number of publications in this area, we will concentrate on those being widely used in applications like games and virtual reality and those relevant for our work. This chapter also contains a detailed description of shadow maps [Williams78] and shadow volumes [Crow77] since many of our proposed algorithms are extended variants or special implementations of these techniques.

In the first part of this thesis we present several enhancements to Williams' shadow map technique for point light sources that can greatly improve shadow quality and rendering speed:

- A method that adjusts the light source's viewing frustum in order to reduce sampling artifacts [Brabec02a].
- An hardware-accelerated method for shadow map filtering [Brabec01] that can be implemented on standard OpenGL hardware.
- A specialized shadow map parameterization for hemispherical or omnidirectional light sources [Brabec02b].
- A combined light map / shadow map approach [Brabec00] that is capable of saving valuable hardware-resources.

The second part describes two approaches that are based on the shadow volume algorithm:

- A shadow volume implementation for complex environments that can be used for many light sources and special light source characteristics [Dmitriev02].

- A full hardware-accelerated implementation of the shadow volume approach, including silhouette detection and extraction [Brabec03].

In the third part, two novel algorithms are presented that can be used to create realistic shadows caused by area or linear light sources:

- A hardware-accelerated soft shadow technique for linear light sources [Heidrich00].
- A hybrid-method that approximates soft shadows using only a single shadow map [Brabec02c].

We will conclude this thesis and discuss future work in Chapter 12.

Chapter 2

Background

In this chapter we will describe some of the concepts needed for understanding the algorithms and methods presented in the following chapters of this thesis. A more detailed and complete overview of computer graphics techniques can be found in [Foley96]. A very extensive overview of the techniques used in digital image synthesis is presented in [Glassner95].

We start with a short introduction to hidden-surface removal techniques. The determination of visible (or hidden) surfaces is one of the major tasks when generating a digital image. As we will see in the next chapters, there is a duality between the computation of visible surfaces seen from the camera and the determination of lit and shadowed surfaces as seen from the light source.

Next, we introduce some of the lighting and shading models used in computer graphics today. Since there exists a vast number of different techniques, we will only focus on those relevant for real-time rendering. A major part of this section is the discussion of shadows caused by various types of light sources.

The last section deals with the architecture of graphics hardware and the principles of hardware-accelerated rendering. We will review the main parts of the general graphics pipeline but also discuss recent trends and features of graphics hardware.

2.1 Hidden-Surface Removal

One of the fundamental tasks in the process of digital image synthesis is the determination of visible surfaces for a given view and scene. Given a set of 3D opaque, solid objects only parts of the scene can be seen from a given point of observation. Objects far away may be completely hidden from the viewer by objects in between, whereas cases exists were only a fraction of an object may be visible.

Although the basic task sounds quite simple, one can imagine that as the number of objects in the scene or the final image resolution increases, the exact and efficient visible-surface determination gets more complicated.