

# Chapter 1

## Introduction

Computer graphics is not only fun and exciting, it is empowering. It shows you how to use the computer to make a picture or a movie of anything you want, with whatever characters you like, in any scene you can imagine, doing whatever you want. The only obstacle is learning how to do it.

One thing strive to learn in computer graphics is realism: how to make fake scenes that seem indistinguishable from real world scenes. We are all familiar with blockbuster movies with thrilling computer graphic special effects, with dinosaurs and exploding worlds that are obvious special effects. You might be surprised that many ordinary films include subtle computer graphics that go undetected and unnoticed.

The photorealistic graphics used for feature films can take from minutes to hours to produce each image, of which 25 or so are needed for each second of film. Each frame of a film are rendered once, from a single point of view, assembled into a film, and reproduced for playback to tens of thousands of filmgoers. While these frames are photorealistic and lush, they are not interactive.

So a second thing we strive to learn in computer graphics is interactivity: how to generate graphics fast enough to respond to user input. Videogames are interactive. Each frame of a videogame must be rendered individually on the fly for each of tens of thousands of players, from a unique point of view, and faster than a 30th of a second for the game to appear “responsive.” The fact that videogames can still generate near-film-quality photorealistic and lush imagery is due to a lot of hard work, fast hardware, precomputation and sneaky tricks. We will learn many, many sneaky tricks.

We begin the study of computer graphics with interactive-quality computer graphics, using simpler and more direct methods for generating reasonably realistic shapes, pictures and video that can run at interactive rates to respond to user input on ordinary personal computers and devices. The techniques in this book can be used to create custom graphics for applications, virtual environments, visualizations, user interfaces, simulations,

computer aided design and videogames<sup>1</sup>.

In addition to real objects, we can use interactive graphics to display shapes that do not exist in reality, or give us novel views of real-world shapes that we could not ordinarily see. Graphics gives us the power to *visualize*. We can use geometry to plot otherwise obscure tables of numbers to gain insight into what they represent. For example, we visualize temperature, pressure and airflow data to predict our local weather. Other visualizations can help us decide what movie to watch, what stocks to buy and even help us live longer.

The interactive computer graphics you will learn in this book can be used to simulate a scene for real-time display and interaction, which supports the construction of entire virtual environments. These virtual environments can be displayed on an ordinary computer screen controlled with a mouse, on a head-mounted goggle display, on the walls of a virtual reality room, or even on a cellphone or tablet display using it as a window into your own secret world. We can also overlay our graphics on top of a cellphone/tablet camera image, producing an augmented reality that combines our virtual world with the physical world.

## 1.1 Image Formation

We perceive the world around us largely through imagery. Light emanates from a source, flying through 3-D space and bouncing off materials until it hits us in the eye. There a lens projects it onto the back of the eye where nerves convert it into the electrochemical impulses processed by the human perceptual system. Hence we perceive our 3-D world by processing 2-D projections of it, and much of computer graphics is based on understanding this process so that it can be quickly simulated on the computer.

Hence computer graphics teaches us how to lie, visually.

The goal of photorealistic rendering is to make it impossible to distinguish a simulated image of a geometric model from a photograph of a real scene. Photographs have become less convincing in court cases. It has become difficult if not impossible to determine where special effects transitions into reality in feature films. I am counting on each of you to use your graphics superpowers for good.

First we need an object to display. The shapes of these objects can be represented in the computer in a variety of ways, but most commonly by a mesh of triangles. We'll use vectors to describe the mesh vertices and linear algebra to manipulate the shapes. The appendices review the necessary

---

<sup>1</sup> Modern videogames employ some advanced techniques beyond the scope of this introductory to computer graphics. Further study in computer graphics, beyond this book, reveals more fundamental concepts in graphics needed for production applications such as feature film special effects and commercial videogames.

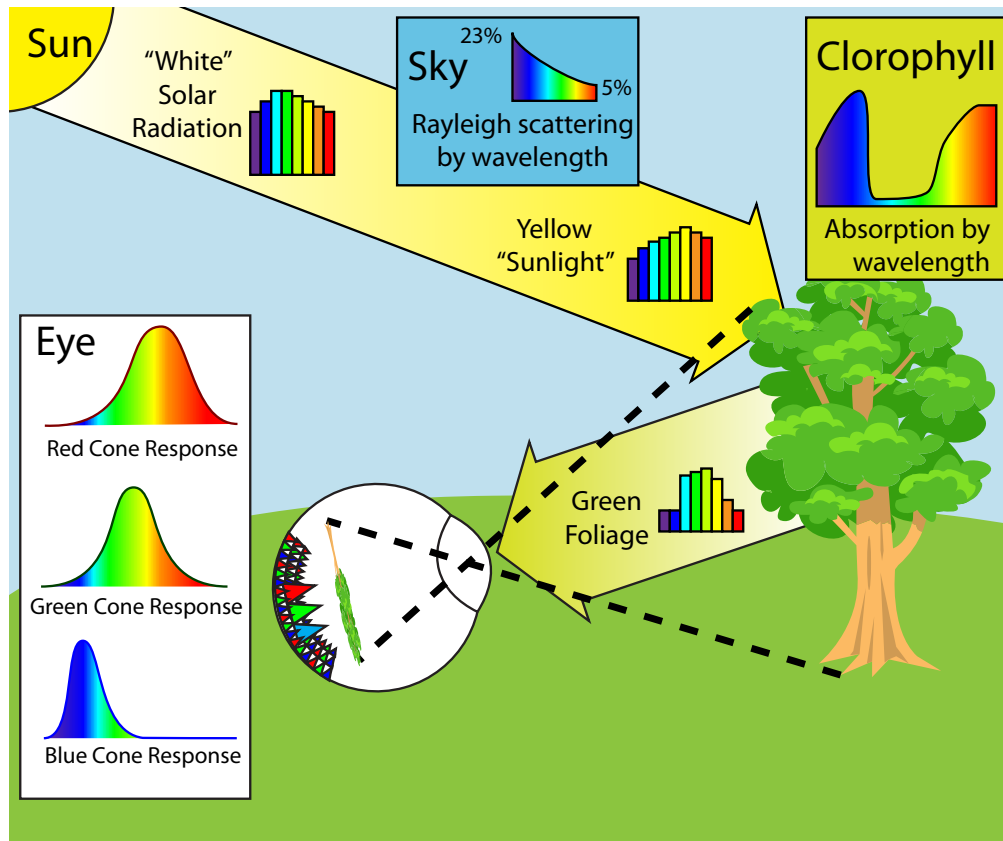


Figure 1.1: Light reflects off of an object, which projects it onto our retina, where it is converted to a perceptual signal for our brain to process.

elements of these subjects. We will also use a little calculus (derivatives) when we discuss interpolation and smooth surface representations.

All of this is built on the image, which is a simple representation of what gets projected onto the retina. An image in front of the eye is a back-projection of the image on the retina, and the key idea is that one can trace light from the environment through this image on its way to the eye.

In computer graphics, we replace the complex interplay of photons and materials with sequences of straight lines. We approximate intricate shapes with meshes of triangles, among other approximations. And we store the results of straight-line light bouncing off triangle mesh objects in a 2-D array of "light records" we call pixels, short for picture elements. The light records stored in each pixel are often represented as an RGB triple of values, utilizing the red-green-blue primaries of the additive color system, which closely matches the response curves of the color-perceiving cone cells in the human retina.

## 1.2 Physics

To understand how light illuminates an object so we can see it, we study (geometric) optics, an area of physics. Optics simplifies light into a stream of photons where each photon has an associated wavelength, ranging from the violet (380nm) to the red (760nm) ends of the visible spectrum. A light source emits a stream of photons, often at different wavelengths, creating a graph of the power of the light at each wavelength, which is roughly proportional to the number of photons per second at each wavelength. (The actual power of each photon is inversely proportional to its wavelength. A “violet” photon is almost twice as powerful as a “red” one.)

When light reaches a surface, it interacts with the surface’s material. This material absorbs some of the photons and reflects others, based in part on the wavelengths of the photons. Hence grass is green because chlorophyll reflects more photons in the “green” region of the spectrum and absorbs more photons at other wavelengths. The sky is blue because the air scatters photons with blue wavelengths more broadly than it does photons with other wavelengths, which is why the sun, which generates photons over the entire spectrum of white light, appears yellow or orange.

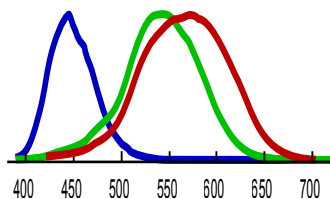


Figure 1.2: Cone spectral response functions (normalized).

## 1.3 Perception

The human retina consists of two kinds of photoreceptors: rods, which measure photon power across all wavelengths, and cones which measure power over a narrow range of wavelengths. There are three kinds of cones: red, green and blue, organized by the (overlapping) regions of the spectrum over which they are most sensitive.

And we study physiology and psychology to understand how the brain reconstructs a 3-D environment from 2-D projections, and how we can generate images that at least partially fool people into thinking they are looking at a 3-D environment.

## 1.4 Overview

Graphics can be simply broken down into four main areas: image manipulation, rendering, (shape) modeling and animation, though there are sub-areas that cross these categories, like visualization, and others that don’t fit into any of them, like interaction. But we have to start somewhere and this breakdown has been useful in the past, if only to sort through research papers.

Much of what we do in graphics focuses on images of some sort, ranging from arrays of colored pixels in a raster image to the lists of strokes and fills that represent an image as vector art. These images can represent the result of graphics, as a still display image, a 3-D movie or even a sculpture, but

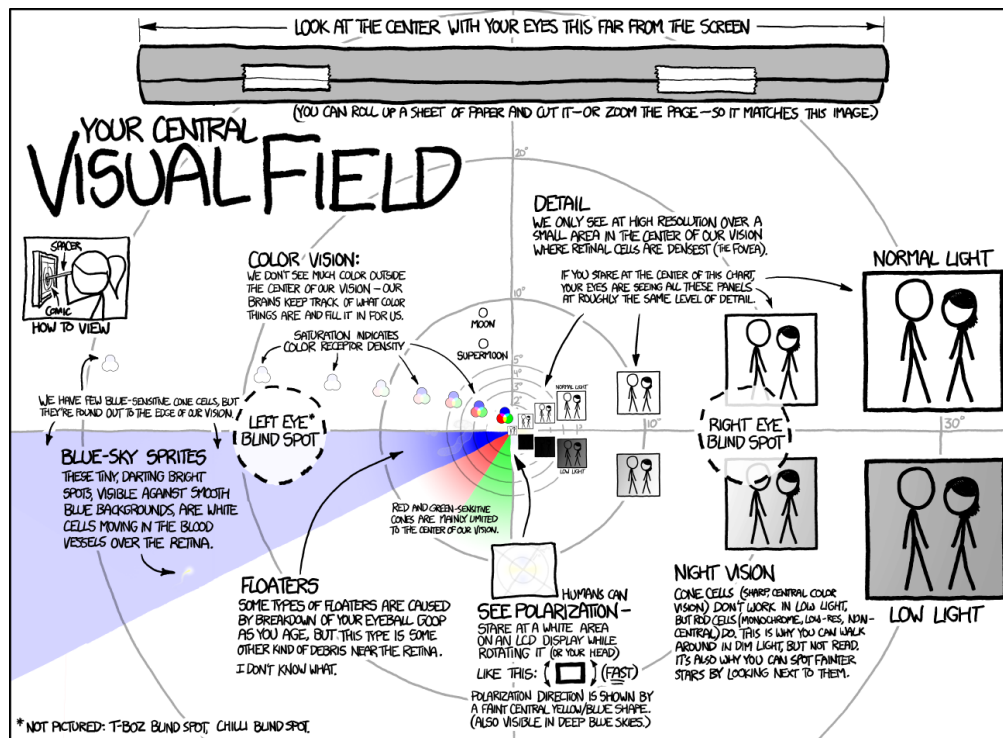


Figure 1.3: Some interesting aspects of the psychophysical perception of the human eye, depicted in the comic xkcd.

they can also represent the input to graphics, used to extract a 3-D scene or to be applied like a decal on a surface, or manipulated into a different but nevertheless realistic image. Graphics shares its concern with images with computer vision and image processing, and different representations of images based on edges, frequencies and other decompositions are useful for its analysis, compression, representation and manipulation.

**Rendering describes the process of making pictures** (though its formal definition describes how to remove fat from a pig carcass). Much of rendering is based on physics, specifically the optics that trace the path light follows from its source eventually to our eye, and the reflectometry of the interaction of this light with various materials. The implementation of this physics often relies on probability, since we try to cleverly sample a small subset of representative paths of light from the overwhelming collection of all the photons bouncing around, and the whole problem becomes one of integration from calculus, since we want the total light reaching the observer summed over all of these different paths. Furthermore, this light forms a **signal and its proper sampling relies on signal processing methods from information theory**. Since the light is intended to reach a human observer, the areas of perception and psychophysics help us understand which paths and other attributes are important to better focus our limited computational resources. Then there's the issue of making this happen sufficiently fast to

generate fifty or so images per second to support videogames and other visual simulations, which relies on the computational notions of parallelism, pipelining and precomputation, and we use the change of coordinates from linear algebra to represent geometry at each stage of the pipeline in its most appropriate coordinate system.

Modeling, specifically shape modeling, is the process of describing the geometry a shape so that it can be processed by a computer. (The related area of computational geometry studies the theoretical computer science of representing and processing shapes.) The most common representation is as a mesh of triangles, which is an irregular array when compared to the ordered layout of a regular array, though graphics has extended the signal and image processing method designed for regular arrays to also work on the irregular arrays of triangle meshes. Hence operations on images developed in image and signal processing such as adjusting their resolution, changing their brightness and contrast, enhancing their edges and other effect have similar analogs for operating on shapes modeled as triangle meshes, and this has grown into an sophisticated extension of differential geometry specifically for meshed surfaces. There are many other shape representations as well, including the smooth curves and surfaces used for computer aided design and manufacture of cars and other physical realizations, based on the mathematics of analysis and approximation. Graphics has also captured shapes through scanning usually as a cloud of points and many methods have been developed to either turn these points into a more sophisticated geometric model, or to just process and render the points directly when they are dense enough.

Animation is the process of changing a stream of images to create the illusion of motion. Some animation results from simulation, ranging from procedures that create in-between images from keyframe storyboard images, to sophisticated mechanical engineering implementations of physical processes ranging from kinematics to fluid dynamics. Alternatively, animation can be acquired and stored in a techniques called motion capture (mocap) that scans the 3-D positions of markers on an moving real-world figure, and this motion can be re-assembled in a different sequence and re-targeted to make a different shape move in a similar manner, using techniques from machine learning and robotics.

Hence computer graphics is a culmination of a bunch of diverse areas from physics to geometry to signal processing to machine learning to psychology, to name a few. The goal of this book is to carefully formulate a fundamental collection of graphics methods along with the background necessary to understand them. The graphics methods are listed as chapters whereas their background is collected as appendices such that they can be referred to repeatedly as needed by the graphics material.