Computer Graphics — Hughes et al.
Chapter 1 — Introduction

1 Reading

1.1. An Introduction to Computer Graphics

- **Computer graphics:** visual communication involving a computer.
- In a simplified sense, CG is *glorified multiplication*: transporting light around a scene by multiplying incoming light by object reflectivities, and repeating the process until the light hits the camera.
- Then CV is *glorified factoring*: determining the descriptions that came together to create a view.
- A few useful principles: (0) *read the works of masters rather than those of their students*, (1) *know what problem you are solving*, and (2) *approximate the solution, not the problem*!
- One of the biggest changes in CG in recent times has been the increase in interactivity. For this reason, user interfaces are becoming increasingly important.
- Nowadays, computer time/resources are cheap while human time/resources are expensive. The UI is where human time is consumed, so it's critical to get this right.
- Graphics is tied to human perception ("how will an application be consumed?") in addition to just getting the physics or algorithms right.

1.2. A Brief History

- **Ambient light:** light that's everywhere in the scene, despite not having any clear origin. Ensures that every visible object has some level of illumination.
- **Diffuse light:** light that's reflected uniformly in all directions.
- **Specular light:** light that's reflected more directionally (i.e. not uniformly in all directions).
- **Raster device:** a display consisting of an array of tiny dots. Has a *resolution* (how small the dots are) and a *dynamic range* (ratio of the brightest to the dimmest possible pixel values).
- Over the previous 50 years, there's been an exponential growth in computational power (see commodity graphics cards) and an instatement of computer graphics to everyday life (powered largely by the adoption of WIMP GUIs; WIMP is *windows, icons, menus, pointers*).
- Also, graphics cards are now programmable. Applications can send programs (**shaders**) to the GPU describing how polygons and images should be processed.

1.3. An Illuminating Example

- Patterns of illumination (e.g. brighter from *here*, dimmer from *here*) are determined by physics. If striving for realism, we should model physics.
- If we have realism, we will also ensure that colors stay the same across different displays.
- Sometimes we don't care about realism, in which case we can talk abstractly about the things we *do* care about (shape, color, form, etc.).

1.4. Goals, Resources, and Appropriate Abstractions

- First, aim to understand the (mathematical?) theory behind the phenomenon you're modeling. Then figure out which approximations will meet your *goals* under the constraints of your *resources*.
- Usually, the end result of computer graphics is to produce something that will be viewed by a human. Thus we can tailor our method to the human visual system (for example, there is typically no need to simulate EMR outside of the visible spectrum because humans won't be able to see it anyway).

1.5. Some Numbers and Orders of Magnitude in Graphics

- At a distance of 1 meter, humans can distinguish things which are about 0.3 mm apart. Apparently, this means that if pixels get much smaller ("twice as small" in 2013 when the book was published), the human eye won't be able to distinguish them.
- Since our resolution in the periphery is terrible, having a high pixel density near the border of the screen is probably unnecessary most of the time.

1.6. The Graphics Pipeline

References

[1] John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, Kurt Akeley. *Computer graphics: principles and practice (3rd ed.)*. Addison-Wesley Professional, 2013.