Chapter 10:
Computer Graphics

Computer Science: An Overview
Tenth Edition

by
J. Glenn Brookshear
Chapter 10: Computer Graphics

- 10.1 The Scope of Computer Graphics
- 10.2 Overview of 3D Graphics
- 10.3 Modeling
- 10.4 Rendering
- 10.5 Dealing with Global Lighting
- 10.6 Animation
Overview

• Branch of Computer Science that applies computer technology to produce/manipulate visual representations.

• Term is increasingly being used specifically in reference to 3-D Graphics
Scope of Computer Graphics

- Emergence of digital cameras has made image manipulation software popular
  - Software allows users to "touch up" photographs
  - Special effects in motion pictures & television often use similar techniques
  - Field of research is *image processing*
    - Focuses on analyzing pixels in an image to identify patterns used to enhance or "understand" the image
2D Versus 3D Graphics

• **2D Graphics:** Deals with manipulating two-dimensional images
• **3D Graphics:** Deals with producing and displaying images of three-dimensional virtual scenes.
2D Versus 3D Graphics

• Wide variety of utility/application software for producing 2-D images
  – Field of research is 2-D graphics
  • Focuses on converting 2-D shapes into pixel patterns to produce an image
2D Versus 3D Graphics

- Field of 3-D graphics deals with converting three-dimensional shapes into images
- Analogous to traditional photography
  - Real world vs. virtual world
Figure 10.1  A “photograph” of a virtual world produced using 3D graphics (from Toy Story by Walt Disney Pictures/Pixar Animation Studios) © Corbis/Sygma
3-D Graphics

• Creation of 3-D image has 2 steps:
  – Creation, encoding, storage and manipulation of scene
  – Process of producing the image

• 3-D graphics is ideal for use in interactive video games and animated motion pictures
3-D Graphics

• Modeling
  – Analogous to designing/constructing a set in traditional motion picture industry
  – 3D graphics scene is “constructed” from digitally encoded data and algorithms
  – Scene produced by computer graphics may never exist in reality
3-D Graphics

• Rendering
  – Applying analytic geometry to compute projection of objects in the scene onto a flat surface (the *projection plane*)
    • Analogous to a camera projecting a scene onto film
  – The image window is the restricted portion of the projection plane that defines the boundaries of the final image
Figure 10.2 The 3D graphics paradigm
3-D Graphics

• Rendering (cont'd)
  • Appearance of each pixel in the final image is computed once the portion of the scene that projects into the image window is identified
  • Results are stored as a bit map representation of the image in a storage area called the *frame buffer*
3-D Graphics

• Displaying
  • Image in frame buffer is either displayed for viewing, or transferred to more permanent storage
Modeling Objects

• **Shape**: Represented by a polygonal mesh obtained from
  – Traditional mathematical equations
  – Berzier curves and surfaces
  – Procedural models
  – Other methods being researched

• **Surface**: Can be represented by a texture map
Modeling Objects

• A set must be designed and the required props collected or constructed
• In 3D graphics, the set is called a scene, and props are called objects
Modeling Objects

• Accuracy of software model of object depends on requirements of situation
  • More detail is needed for foreground objects
  • More detail can be produced for cases not under stringent, real-time constraints

• More precise models:
  • Higher-quality images
  • Longer rendering times
Modeling Objects

- Shape of 3D graphics object is usually described as a collection of **planar patches**
  - Each planar patch is the shape of a polygon
  - These polygons form a **polygonal mesh** that approximates the shape of the object
  - Planar patches in a polygonal mesh are often chosen to be triangles (minimum flat surface in 3-D space)
Figure 10.3 A polygonal mesh for a sphere
Modeling Objects

• More general shapes can be described by more sophisticated analytical means
  – e.g., Bezier curves: a curved line segment in 3-D space defined by only a few (control) points
  – Bezier technique can be extended to describe 3-D surfaces (Bezier surfaces)
Figure 10.4 A Bezier curve

Control points marking the ends of the curve

Control points used to distort the curve

Curve
Modeling Objects

• Can also construct a polygonal mesh using brute force method
  • Build physical model of the object
  • Record points on surface by touching surface with pen device that records position in 3-D space
  • Process is known as *digitizing*
  • Collection of points then used as vertices to obtain polygonal mesh
Modeling Objects

• Some shapes are too complex
  – e.g., trees, mountain ranges, clouds, smoke, flames
  – Polygonal meshes generated automatically by programs called *procedural models*
  – Next example shows generation of mountain ranges
Figure 10.5
Growing a polygonal mesh for a mountain range

a. Identify the midpoints

b. Connect the midpoints

c. Move the midpoints

d. Repeat the process on the smaller triangles
Modeling Objects

- Can simulate underlying structure of object as large collection of particles
  - Such procedural models are called *particle systems*
- Examples: pouring water, flickering fire flames, clouds, crowd scenes
Modeling Objects

- Polygonal mesh only captures shape of object
- Various techniques encode additional information about object; e.g., color
- *Texture mapping*: applying wallpaper to surface of object
Modeling Objects

• Building realistic models is ongoing research topic
  – e.g., living characters: hair, skin, fur, etc.
• Use techniques in both modeling and rendering processes
Figure 10.6  A scene from Shrek 2 by Dreamworks SKG (© Dreamworks/The Kobal Collection)
Modeling Scenes

- **Scene graph**: data structure collecting objects in a scene, and assigning them location, size, orientation
- Also contains links to light sources and the camera
- Analogous to studio set-up
Reflection Versus Refraction

• **Reflection**: Light rays bounce off surface.
  – Specular light
  – Diffuse light
  – Ambient light

• **Refraction**: Light rays penetrate surface.
Reflection

• **Incident angle**: angle at which ray of light strikes a surface
• **Specular light**: light reflected off a smooth surface, traveling away as parallel rays
• **Diffuse light**: light reflected off unsmooth surface, traveling away in different directions
Figure 10.7  Reflected light

- Light source
- Normal
- Incoming ray
- Incidence angle
- Reflected ray
- Angle of reflection
Reflection

• **Ambient light**: “stray” (scattered) light not associated with any particular source or direction

• **Anisotropic surface**: appearance shifts from shiny to dull as it rotates

• **Insotropic surface**: reflection patterns are symmetric
Figure 10.8 Specular versus diffuse light
Figure 10.9  Refracted light
Rendering

- **Clipping**: Restricts attention to objects within view volume
- **Scan Conversion**: Associates pixel positions with points in scene
- **Shading**: Determines appearance of points associated with pixels
Figure 10.10 Identifying the region of the scene that lies inside the view volume

View volume

Image window

Center of projection

Only the portion of an object inside the view volume will appear inside the image window
Figure 10.11
The scan conversion of a triangular patch
Shading Techniques

- **Flat Shading**: Creates faceted appearance
- **Gouraud and Phong Shading**: Creates smooth, rounded appearance
- **Bump Mapping**: Creates bumpy, rounded appearance
Figure 10.12  A sphere as it might appear when rendered by flat shading
Figure 10.13  A conceptual view of a polygonal mesh with normal vectors at its vertices

Vectors indicate the orientation of the original surface.
Figure 10.14  A sphere as it might appear when rendered using bump mapping
Rendering Pipeline

• Consists of traditional algorithms for clipping, scan conversion, and shading
• Often implemented in firmware
• Used as an abstract tool in graphics applications
Local Versus Global Lighting

• Local Lighting Model: Does not account for light interactions among objects
• Global Lighting Model: Accounts for light interactions among objects
  – Ray Tracing
  – Radiosity
Figure 10.15 Ray tracing

Light source

Ball

Mirror

This point on the mirror should be rendered as the back of the ball illuminated by light reflected from the mirror

Light ray

Pixel location

Image window

Center of projection