Chapter 4

Vector Graphics

2017.03

Prof. Park Kyusik

Contents

- Vector Graphic Characteristics
- Anti-aliasing
- Open GL
- 3D Vector Graphic – Modeling and Rendering

Vector Graphic

- Main Characteristics
  - **Compact** (less memory than image for storage)
  - **Scalable** (No quality change with scaling, zoom in/out) → resolution-independent

- Small file size and memory < bitmap image
  - Attractive for networked multimedia, but not widely used on WWW
  - SVG (W3C standard) < Shockwave Flash (SWF)
    (de facto standard)
  - Common formats for PDF, SVF, SWF

Modeling - Rendering

- Modeling
  - Represent lines, curves, shapes,... by parameters of their defining equations
  - Ex) line $y = mx + c$, store m and c (or endpoints)

- Rendering
  - Compute pixels (location, color) which must be set in order to draw the line, curves,…
  - Generally, rendering result will exhibit 'Jaggies'
    - Pixels' coordinates are integers, equations in vector graphic uses real numbers – no exact matching

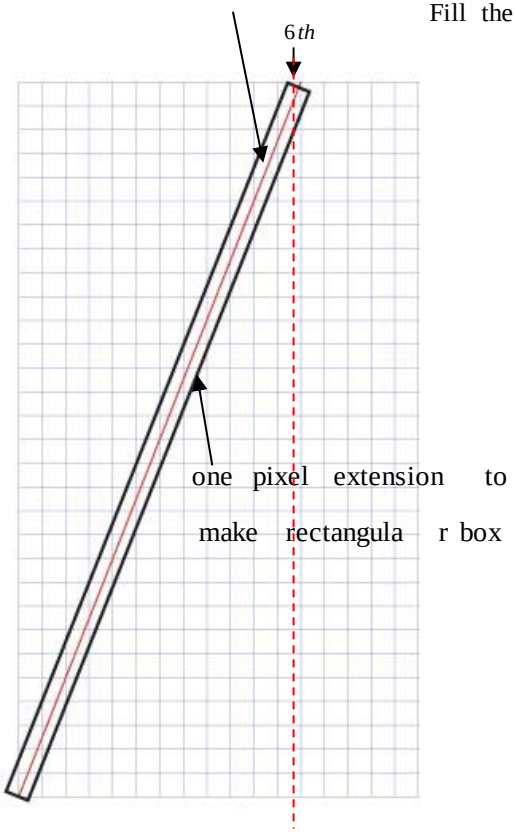
Anti-aliasing

- Rendering a vector object to produce an image made up of pixels
  - Jaggies - called aliasing
  - Ex) Line - continuous line must be approximated by discrete pixels of finite size

- Anti-aliasing
  - Mitigate aliasing by coloring pixels in shades of grey (for a black line on figure in next page)
  - **Shading grey (gray) algorithm**

Anti-Aliasing Example

Original line draw

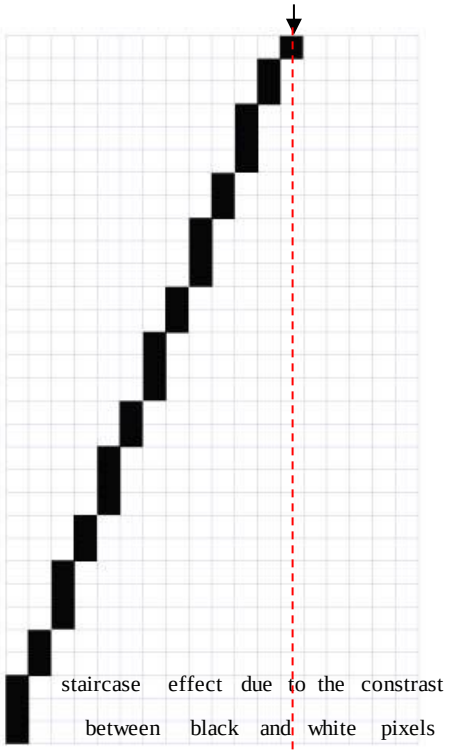


Original line draw

Single pixel approximation to a line

Line Drawing Algorithm :

Fill the pixels intersect with original line as black

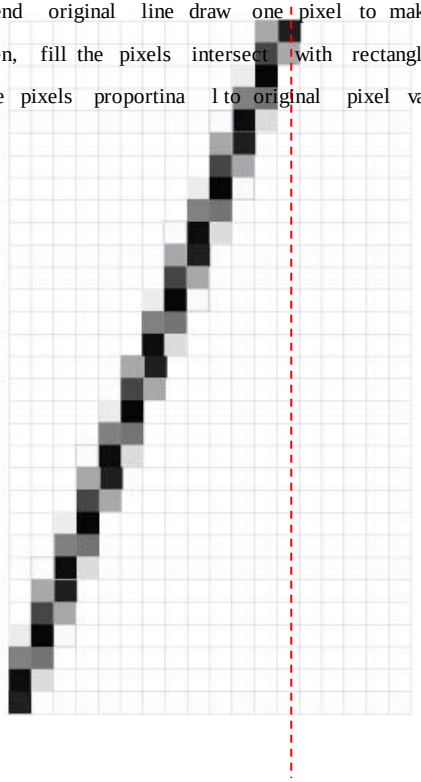


Approximation to original line draw

(Aliasing)

Shading Gray Algorithm

- Soften effect by using intermediate gray values for some pixels
- Extend original line draw one pixel to make rectangular box
- Then, fill the pixels intersect with rectangle box with gray (fill the pixels proportional to original pixel values by multipling)



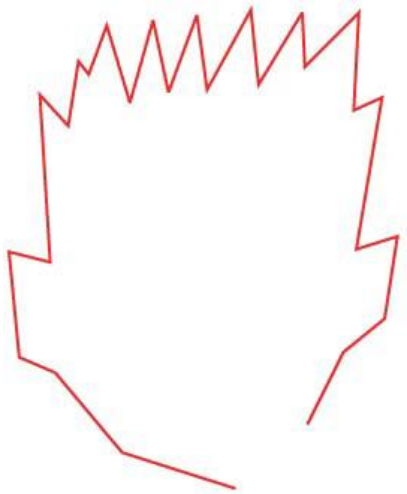
Anti - aliasing

Primitive Shapes for Graphic

- Generally, drawing programs such as Visio, Adobe illustrator, provide primitive shapes that can be easily represented mathematically

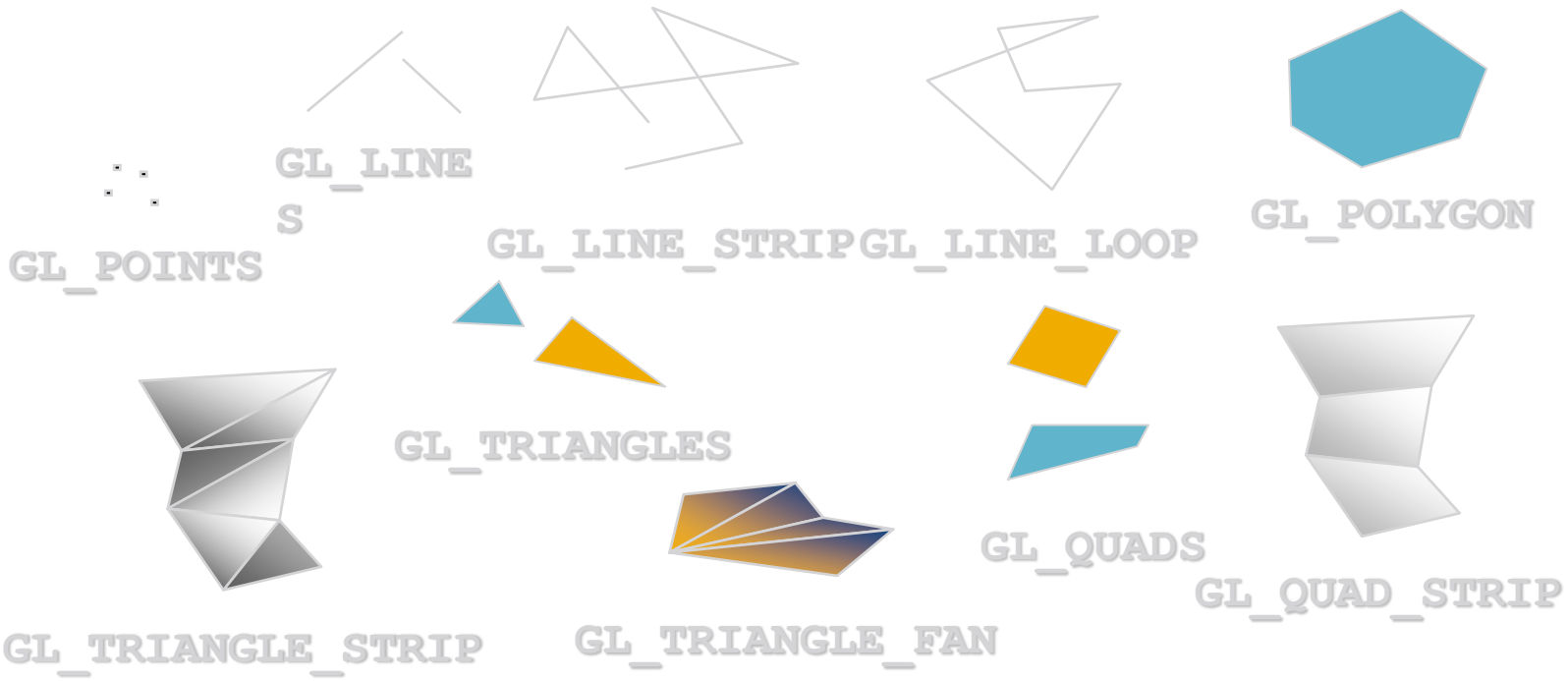
  - Rectangles and squares (may rounded corners)
  - Ellipses and circles
  - Straight lines, polylines and polygons
  - Smooth (Bézier) curves

polyline = shape made with a lot of line connection



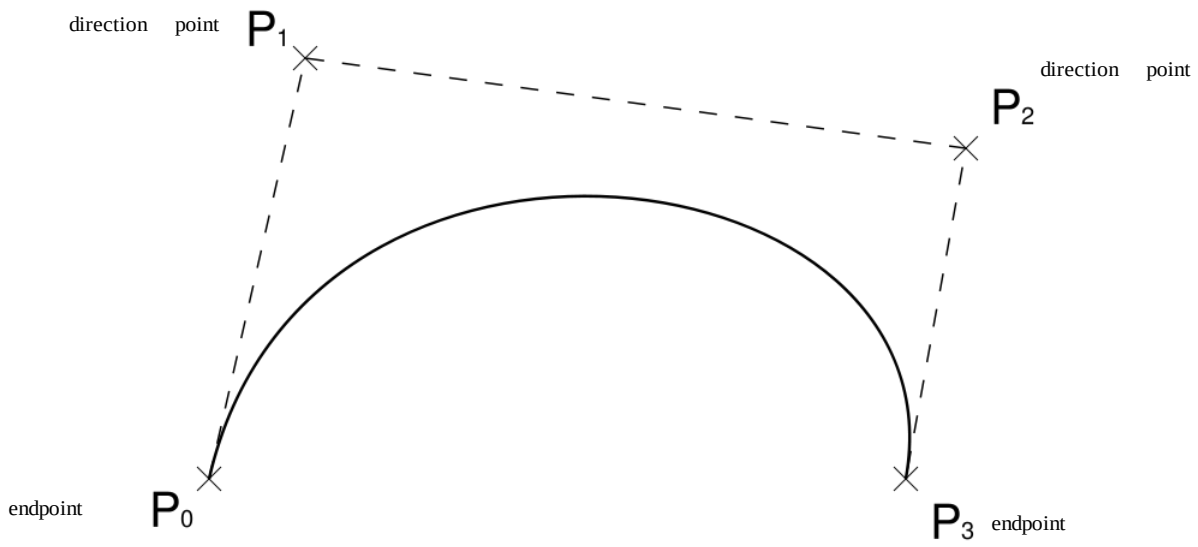
OpenGL Geometric Primitives

- Open Graphic Library by Silicon Graphic in 1992
  - 2D and 3D graphic standard API
  - All geometric primitives are specified by vertices

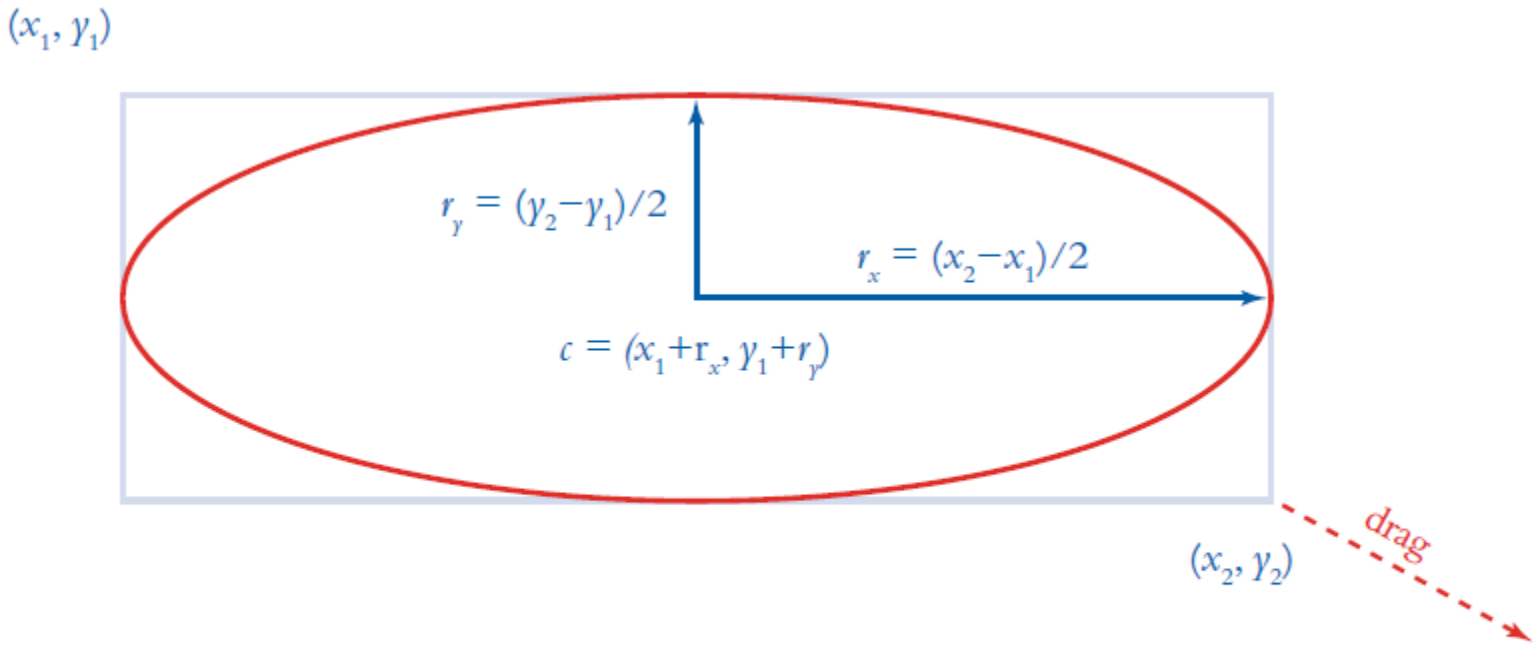


Smooth (Bézier) Curves

- Smooth curves completely specified by four control points
  - 2 endpoints (P_0, P_3)
  - 2 direction points (P_1, P_2)
- Making a curve
  - Define a curve by length and direction of lines from endpoints to direction points



$$\mathbf{B}(t) = \mathbf{P}_0(1-t)^3 + 3\mathbf{P}_1 t(1-t)^2 + 3\mathbf{P}_2 t^2(1-t) + \mathbf{P}_3 t^3 \ , \ t \in [0,1].$$



Drawing an ellipse

Definition of Path

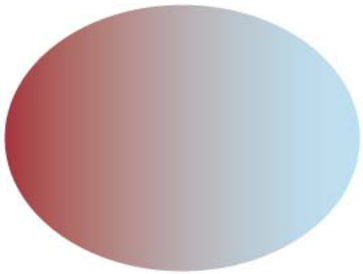
- Define Bézier curves in terms of path
  - Closed path (curves connected end-to-end) and open path
  - Use Pen tool in drawing programs to build up path

Stroke and Fill

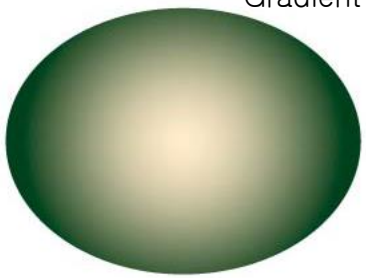
- Apply Stroke to make path visible
  - Like tracing with ink, specify width and color

- Apply Fill to closed path or shape
  - Like coloring it in, specify color or a gradient fill or pattern fill
  - Gradients may be linear or radial (방사형태)

Gradient = gradual color change

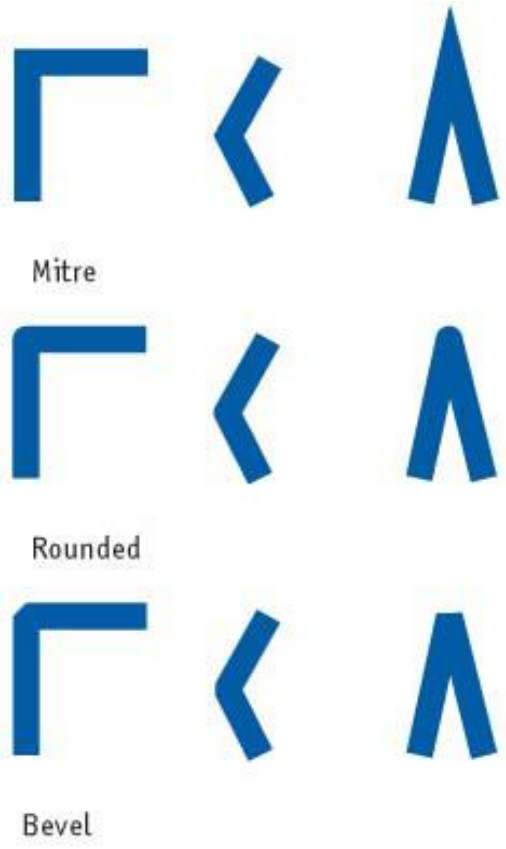


Linear Fill

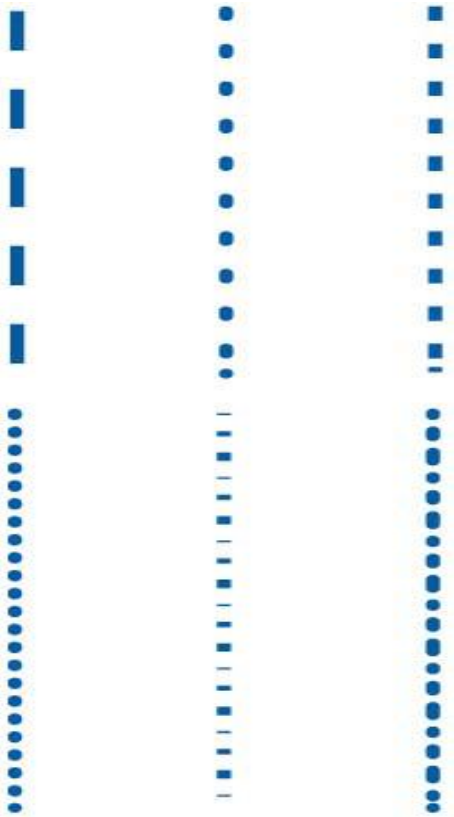


Radial Fill

Stroke Styles

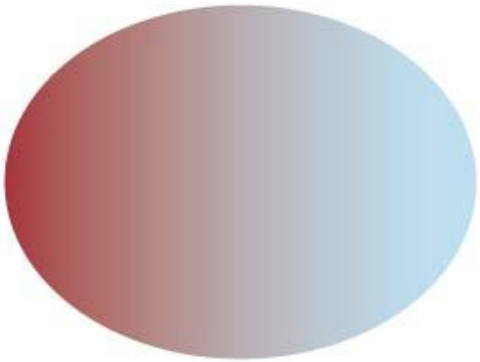


Joining Styles

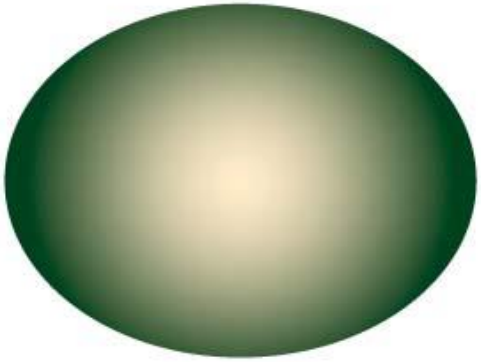


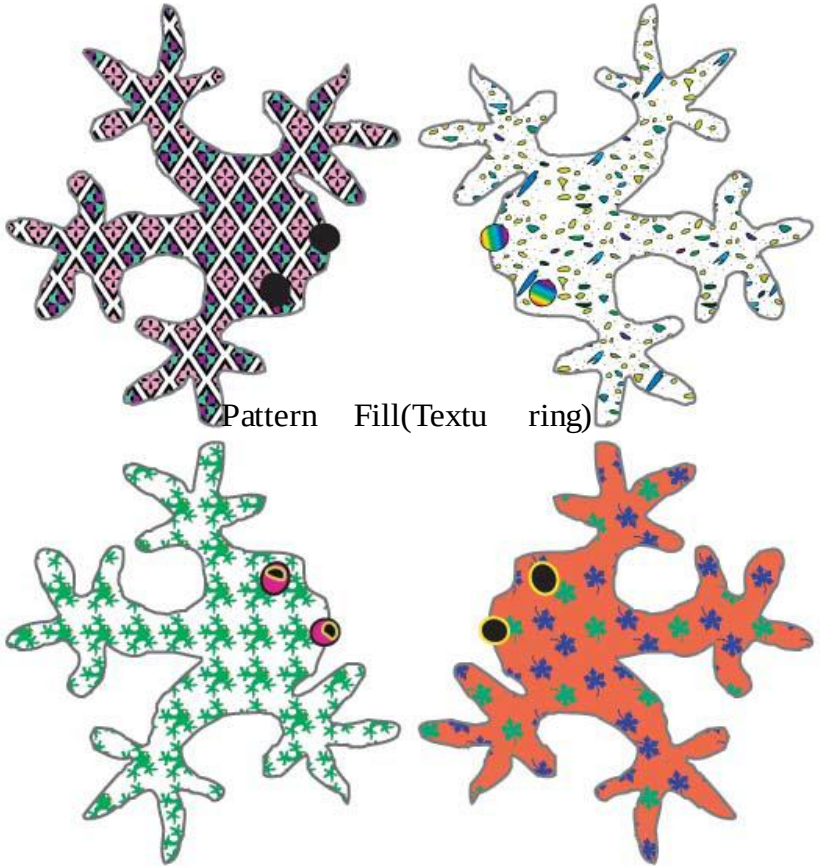
Dashed Effects

Gradient Fill, Pattern Fill



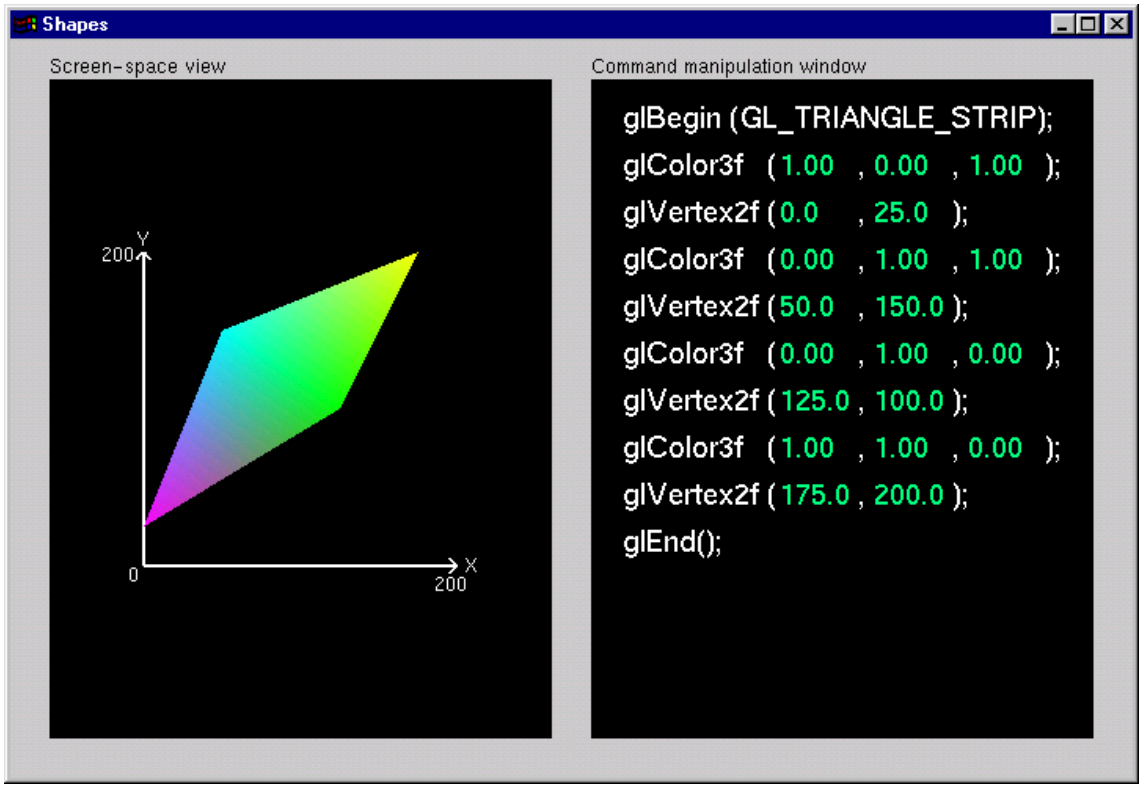
Gradient Fill





Pattern Fill(Textu ring)

OpenGL Example - Gradient Fill



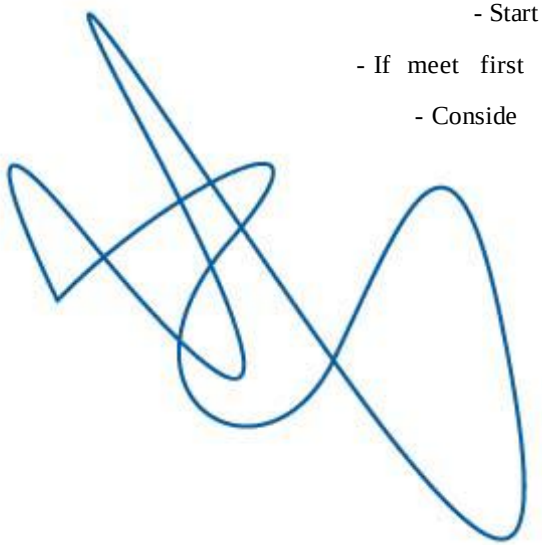
glColor3f(R, G, B)

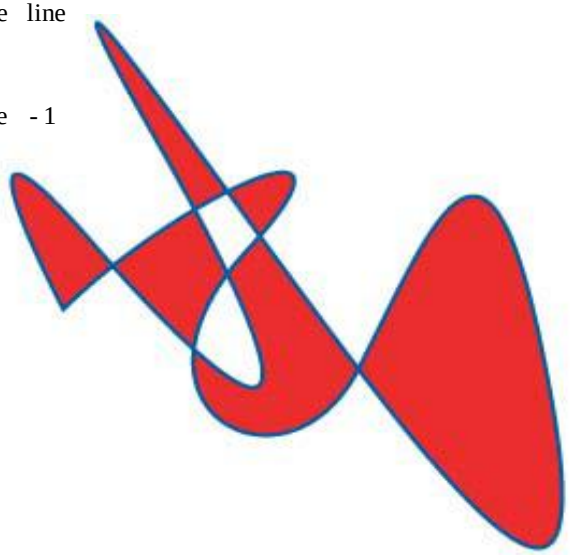
Fill : Coloring inside path Which is Inside and Outside ?

- Non-zero winding number rule

Draw some arbitray line and trace the line

- Start with counter 0
- If meet first edge, + 1, second edge - 1
- Conside area with 1 as inside





0 (+1) 1 (-1) 0

0: outside, 1: inside

Vector Transformations

- Typical transformation of vector object can be done by changing stored vector values

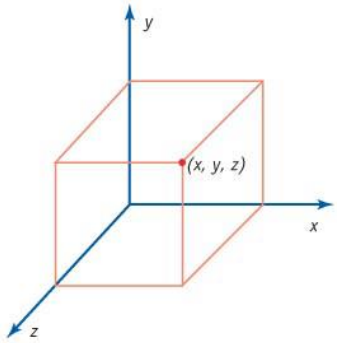
  - Translation (linear movement)
  - Scaling (Zoom in and out)
  - Reflection
  - Rotation
  - Shearing (skewing)

Transformations

An object being scaled, rotated, reflected, sheared and translated

3D Vector Graphics

- Conceptually simple extension of 2-D
  - Add z-axis at right angles to x- and y-axes
  - Point is defined by (x, y, z) coordinates
  - Generalize coordinate geometry from 2D to 3D



Ex) square → cube, circle → sphere

- Practically complicated and difficult
  - Requires 3D visualization skills
  - Complex tools, need high computational power
  - 3D modeling - rendering
    - Model - geometrical features of object
    - Rendering - surface characteristics, colors and textures

to give 3D realism

3D Modeling

- Three general approach to model 3D objects
  - Constructive solid geometry
    - Use of few geometric basis model such as cube, cylinder, sphere, pyramid → Set operations of union, intersection, difference
    - Good for modeling man-made objects, architecture (CAD)
  - Free-form modeling
    - Use representation of an object's boundary surface as the basis of model – ex) polygon mesh
  - Procedural modeling
    - Use of some algorithm or procedure
    - Ex) **Fractal**, Metaballs, Particle systems

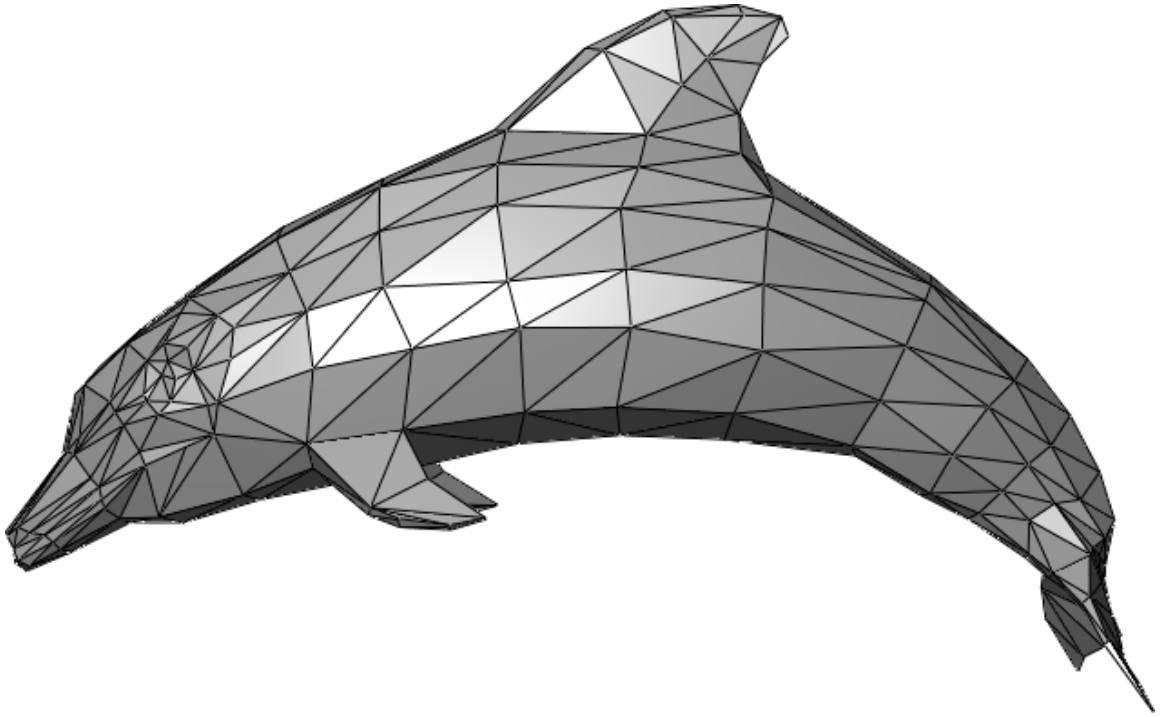
Constructive solid geometry model

Set operation (union, difference , intersecti on)

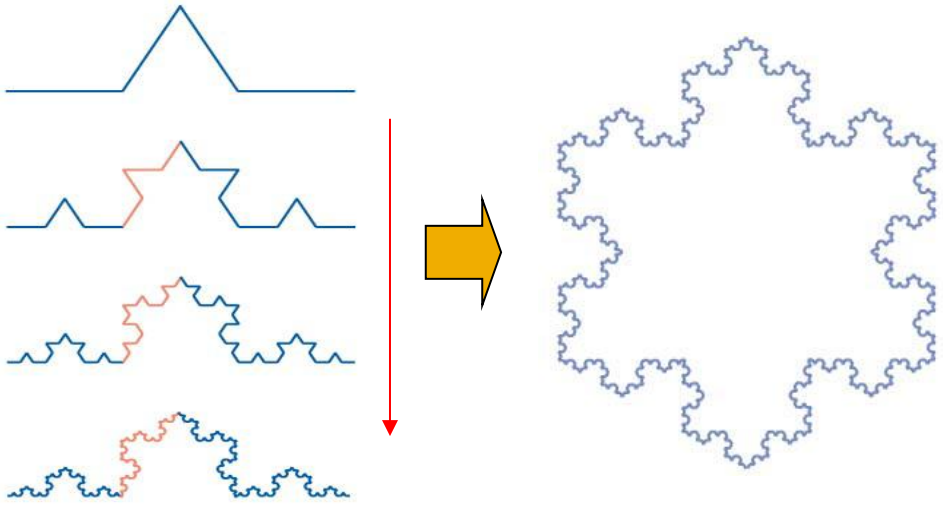
The union, intersection and difference of two solid objects

Free-form modeling (Polygon Mesh)

represent dolphin' s boundary surface as polygon mesh

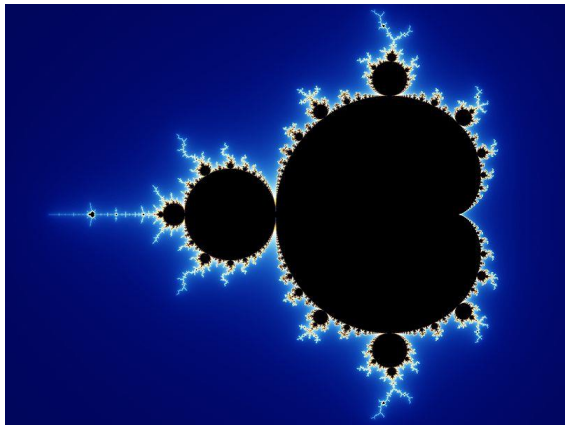


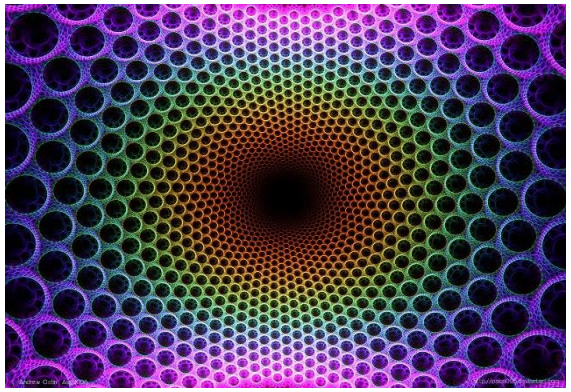
- Fractals (Procedural modeling)
  - In 1975, Benoit Mandelbrot
  - Self-similarity
    - Exhibit same structure at all levels of detail similar to whole structure (each part is reduced size copy of whole)
    - Recursively generated

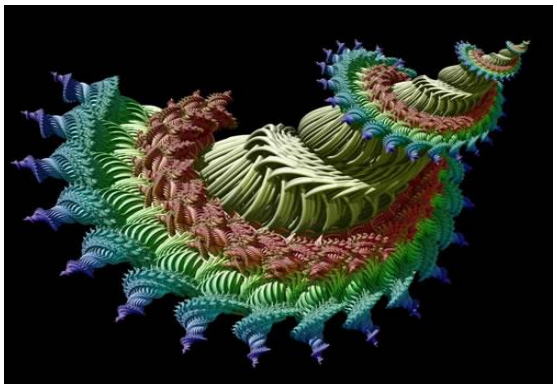


Mandelbrot set

Look up
http://en.wikipedia.org/wiki/Mandelbrot_set







3D Rendering

- A procedure giving a 3D realism to the model
  - Surface characteristics (shading), colors, textures, etc.

- Shading
  - Means surface characteristics of object under different lighting conditions or light sources such as – spots, diffuse, ...
  - Need a computing model based on physics of light

Shading example

Shading = different coloring for each face of box or giving a different amont of light for each face of box

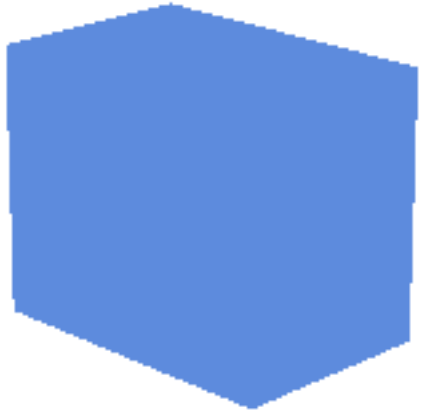


Image with the edge lines removed.

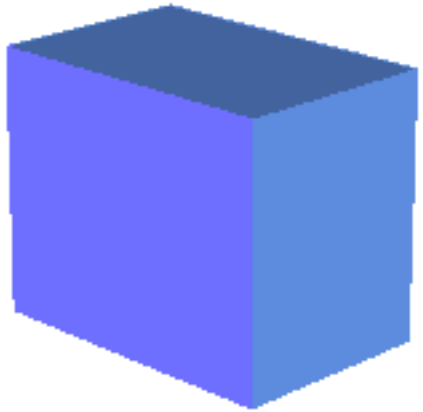
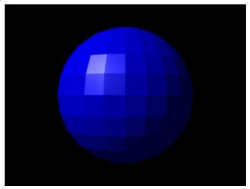


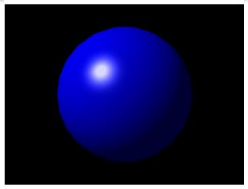
Image rendered with shading surface to make 3D object

Shading Algorithms

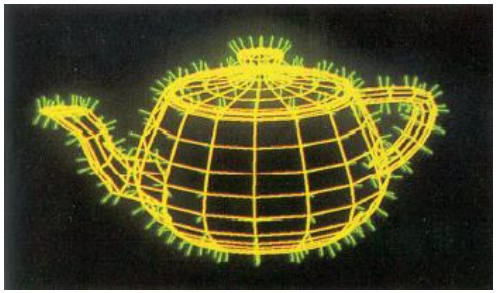
- Ray casting
  - Different color across each polygon mesh surface to simulate 3D object
  - Not considering light interaction between object
  - Flat, Gouraud, Phong shading
- Ray tracing
  - Take account of interaction between objects
- Radiosity
  - Simulate light diffusion and reflections between the objects and shadows

Shading-Ray Casting

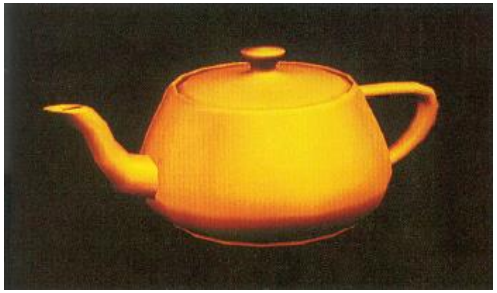

FLAT SHADING


PHONG SHADING

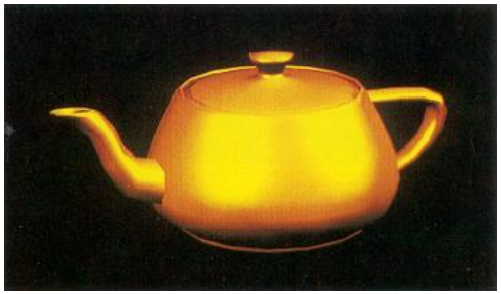
Wireframe (3D model)



Flat



Gouraud

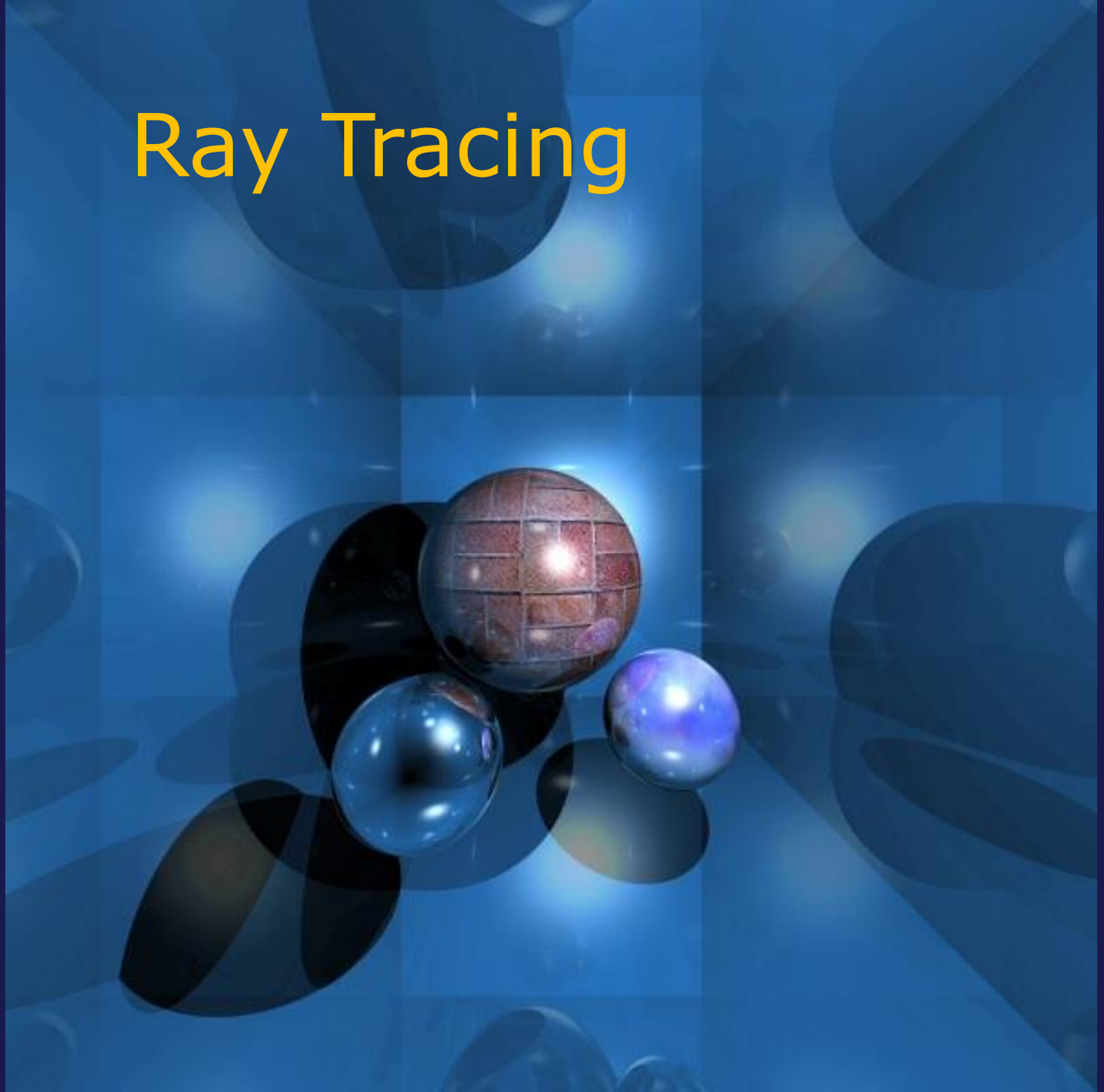


Phong

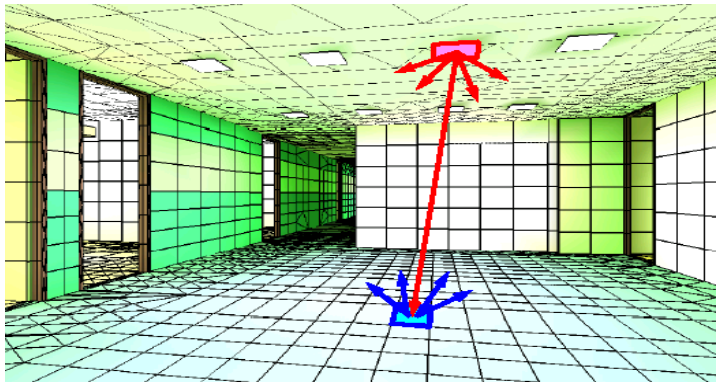
Wire frame, flat, Gouraud and Phong shading


Ray Tracing

Ray Tracing



❖ Radiosity



Simulate light diffusion and reflection s between th e objects, and shadows

Treat every polygon as light source and

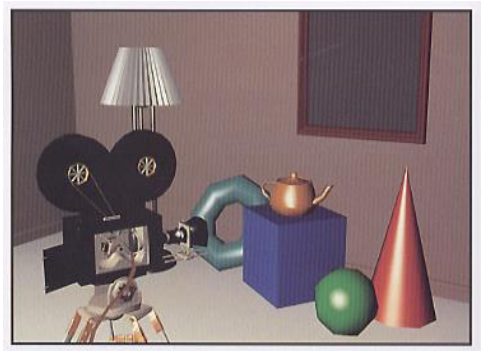
compute the interactio ns between polygons repeatedly

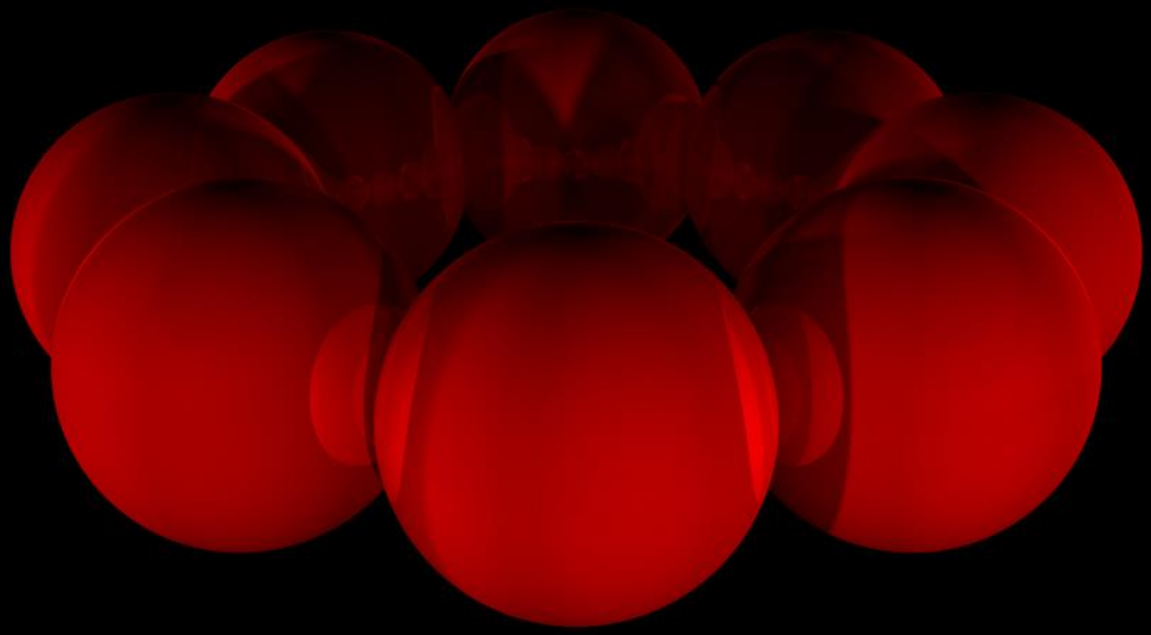
Summary

- Ray Casting
  - Flat
  - Gouraud
  - Phong
- Ray Tracing
- Radiosity

Less Expensive

More Accurate





Homework #4

- Read Chap 3

- Fractal
  - Summary more about fractal theory
  - Find out free fractal SW on WWW
    - Generate your favorite pattern (**ultra fractal**)

- Book
  - Add more contents about vector graphics, open GL…