Computer Graphics

Bing-Yu Chen
National Taiwan University

Introduction to OpenGL

- General OpenGL Introduction
- An Example OpenGL Program
- Drawing with OpenGL
- Transformations
- Animation and Depth Buffering
- Lighting
- Evaluation and NURBS
- Texture Mapping
- Advanced OpenGL Topics
- Imaging

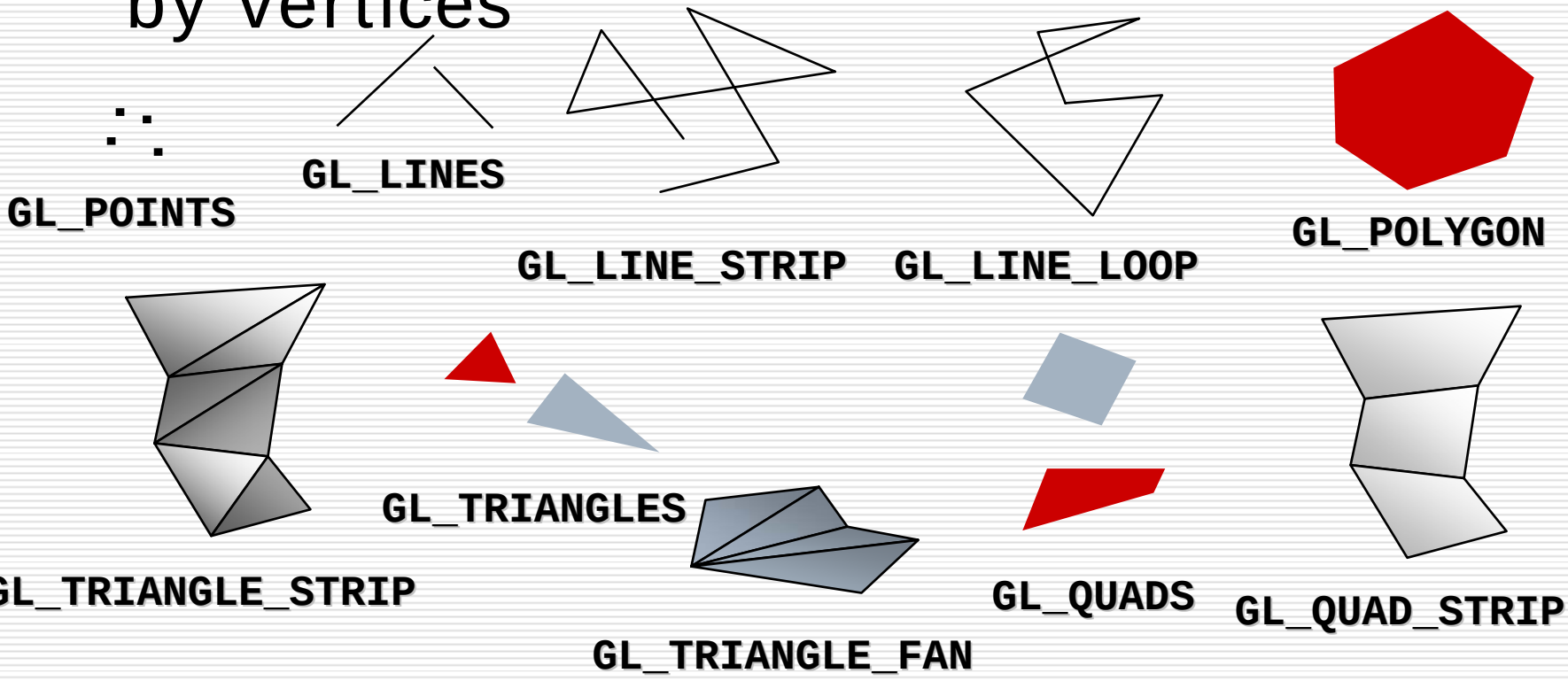
modified from
Dave Shreiner, Ed Angel, and Vicki Shreiner.
An Interactive Introduction to OpenGL Programming.
ACM SIGGRAPH 2001 Conference Course Notes #54.
& ACM SIGGRAPH 2004 Conference Course Notes #29.

What can OpenGL Draw?

- Geometric Primitives
  - points, lines and polygons
- Image Primitives
  - images and bitmaps
- Separate pipeline for images and geometry
  - linked through texture mapping
- Rendering depends on state
  - colors, materials, light sources, etc.

OpenGL Geometric Primitives

- All geometric primitives are specified by vertices



Simple Example

```
void drawRhombus( GLfloat color[] )
{
 glBegin( GL_QUADS );
 glColor3fv( color );
 glVertex2f( 0.0, 0.0 );
 glVertex2f( 1.0, 0.0 );
 glVertex2f( 1.5, 1.118 );
 glVertex2f( 0.5, 1.118 );
 glEnd();
}
```

Specifying Geometric Primitives

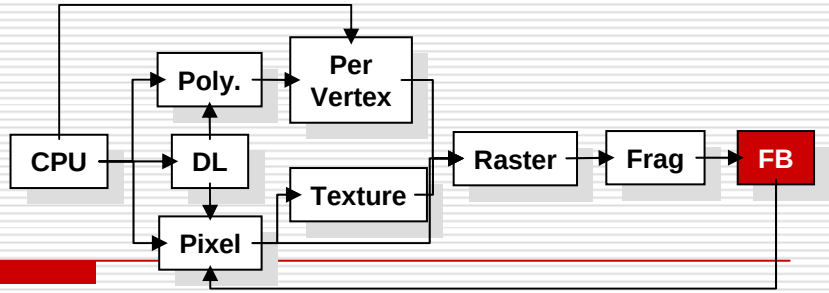
- Primitives are specified using

```
glBegin( primType );
glEnd();
```

  - *primType* determines how vertices are combined

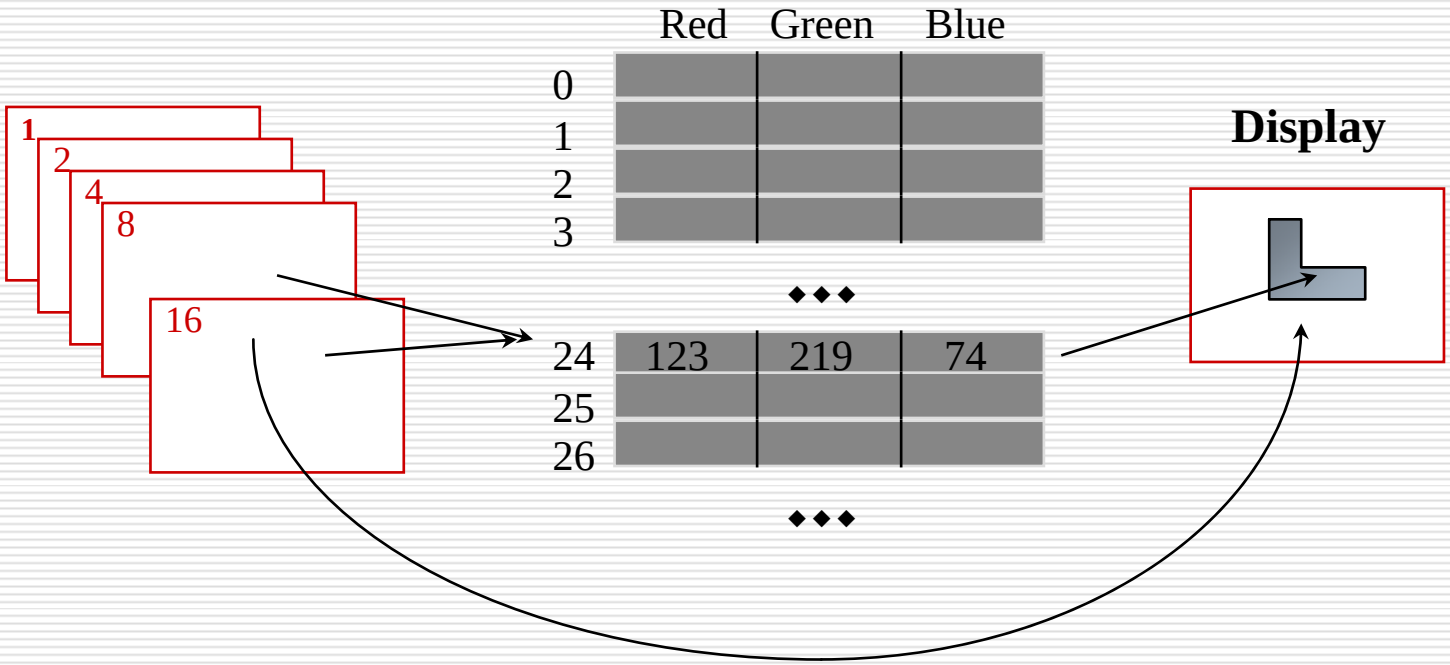
```
GLfloat red, green, blue;
Glfloat coords[3];
glBegin( primType );
for ( i = 0; i < nVerts; ++i ) {
  glColor3f( red, green, blue );
  glVertex3fv( coords );
}
glEnd();
```

OpenGL Color Models



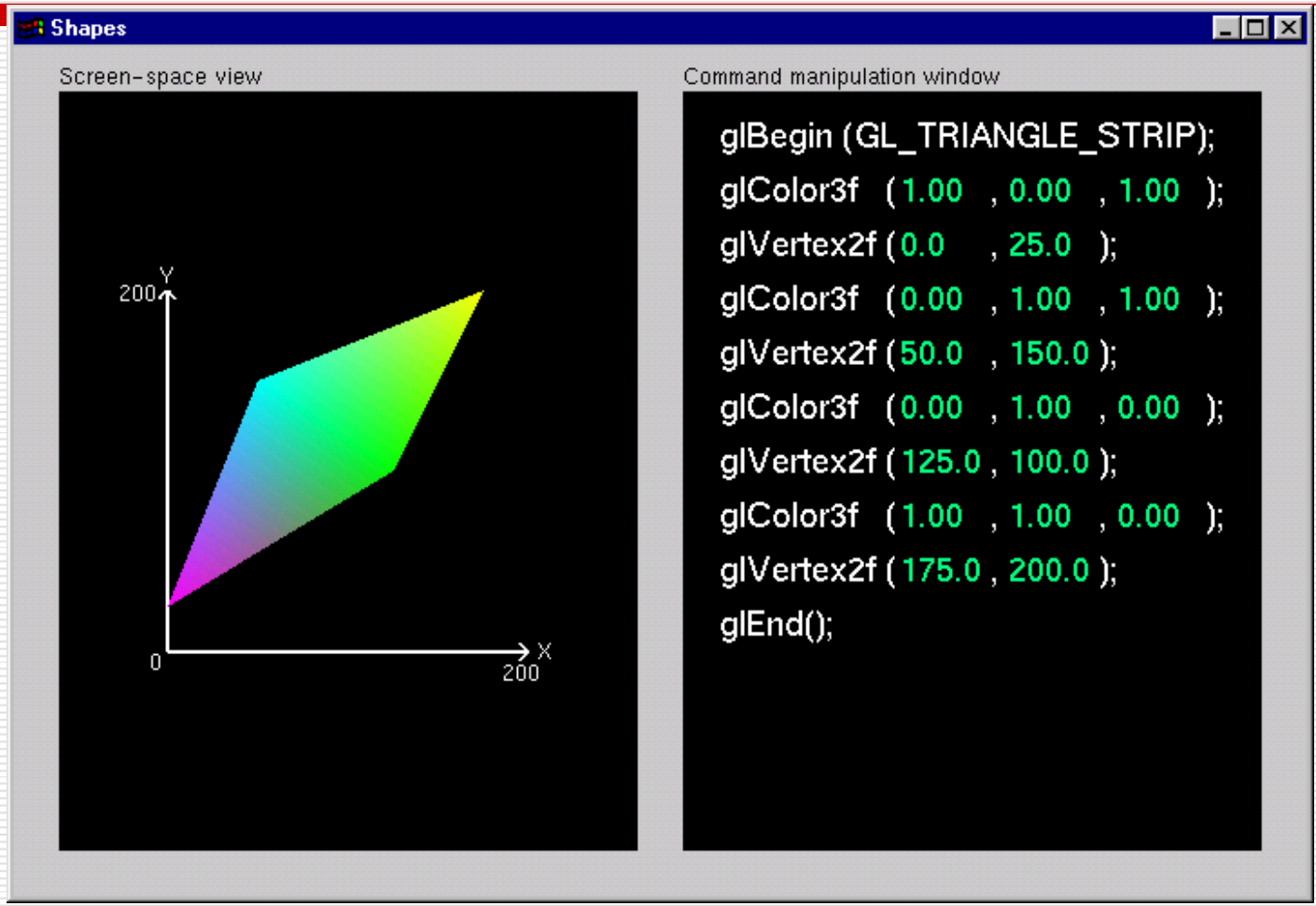
- RGBA or Color Index

color index mode



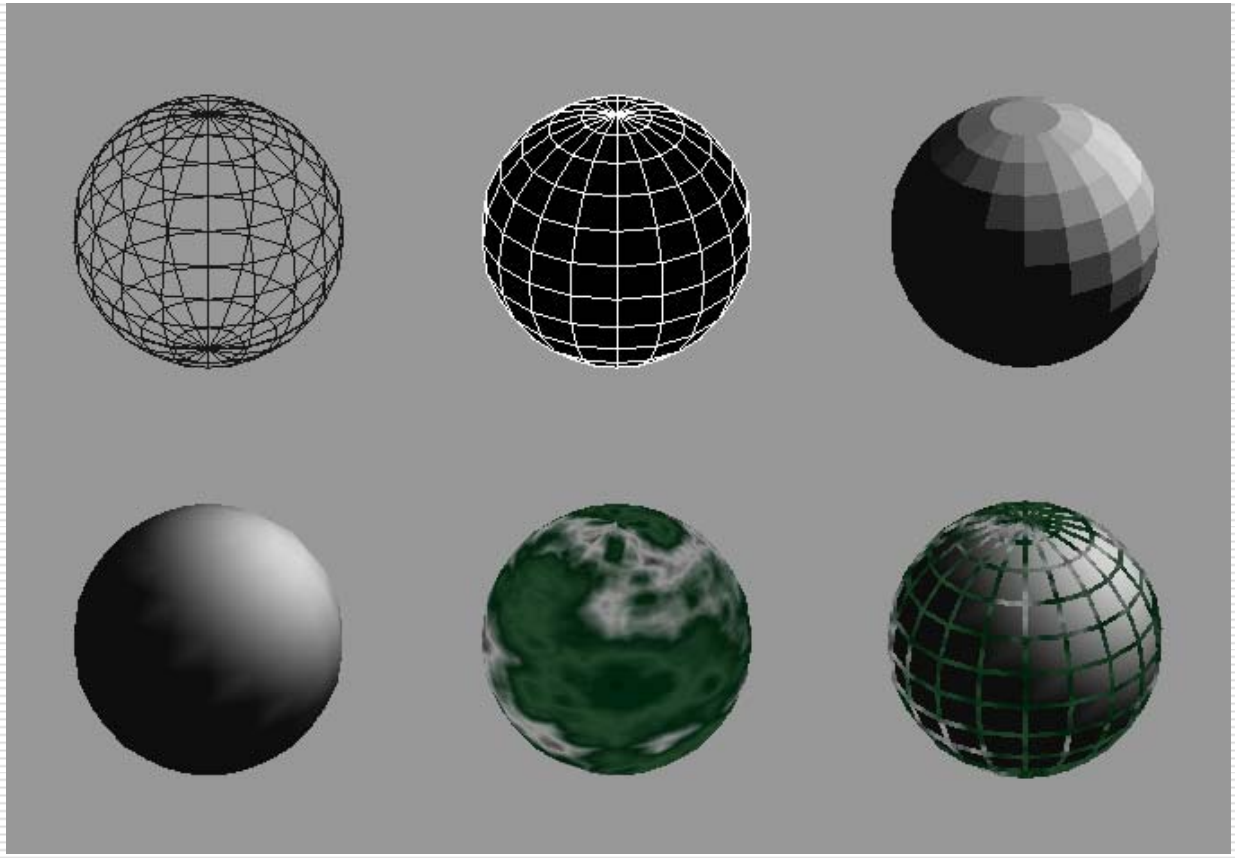
RGBA mode

Shapes Tutorial

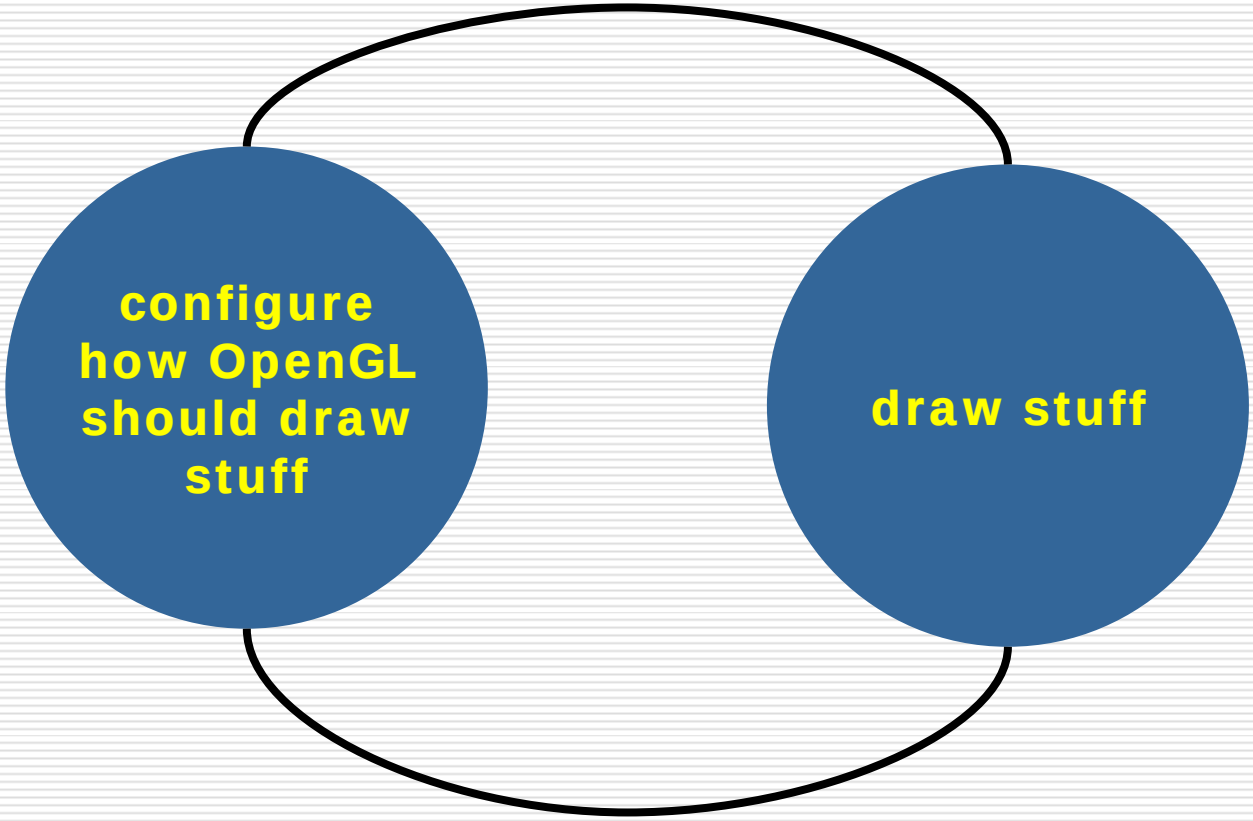


Controlling Rendering Appearance

From Wireframe to Texture Mapped



How OpenGL Works: The Conceptual Model



OpenGL's State Machine

- All rendering attributes are encapsulated in the OpenGL State
  - rendering styles
  - shading
  - lighting
  - texture mapping

Manipulating OpenGL State

- Appearance is controlled by current state

```
for each ( primitive to render ) {
    update OpenGL state
    render primitive
}
```

- Manipulating vertex attributes is most common way to manipulate state

```
glColor*() / glIndex*()
glNormal*()
glTexCoord*()
```

Controlling current state

- Setting State

```
glPointSize( size );
glLineStipple( repeat, pattern );
glShadeModel( GL_SMOOTH );
```

- Enabling Features

```
glEnable( GL_LIGHTING );
glDisable( GL_TEXTURE_2D );
```