Interactive Computer Graphics

--Shader-Based OpenGL (Sixth Edition)

Edward Angel & Dave Shreiner

University of New Mexico

主讲教师：陈泽琳

Introduction to the Course

- ***Prerequisites*** **:** sound programming skills C, C++, data structure, discrete mathematics, linear algebra, solid 3D analytic geometry and spatial thinking.
- Semester total class hours : 48

  Teaching hours: 32

  Experiment hours: 16

Conception & Programming

- Learn the **conception** of CG, **Programming** with OpenGL and Shader language (GPU), some applications

- **Conceptions:** geometry, transformations, viewing and projections, lighting, shading, rendering, texture mapping, and scene graphs modeling, and interaction such as events, callbacks functions

Textbook

Chapter 1: Graphics Systems and Models	(3 class hours)
Chapter 2: Graphics Programming	(3 class hours)
Chapter 3: Interaction and Animation	(2 class hours)
Chapter 4: Geometric Objects and Transformations	(5 class hours)
Chapter 5: Viewing	(4 class hours)
Chapter 6: Lighting and Shading	(4 class hours)
Chapter 7: Discrete Techniques	(4 class hours)
Chapter 8: From Vertices to Fragments	(5 class hours)
Chapter 9: Modeling and Hierarchy	(2 class hours)

课件、作业、实验要求等在教学在线上

References

- www.opengl.org
  - Standards documents
  - Sample code
- The OpenGL Programmer's Guide (the Redbook) 7th Edition
  - The definitive reference
  - Mixes 3.0 and 3.1
- OpenGL Shading Language, 3rd Edition
- OpenGL ES 2.0 Programming Guide

Grading and exams

- ***Four experiments for each student:*** individual fulfills the first three; group fulfills the last
- ***Grading and exams:***

Home work + four experiments	40%
Final Examination	60%

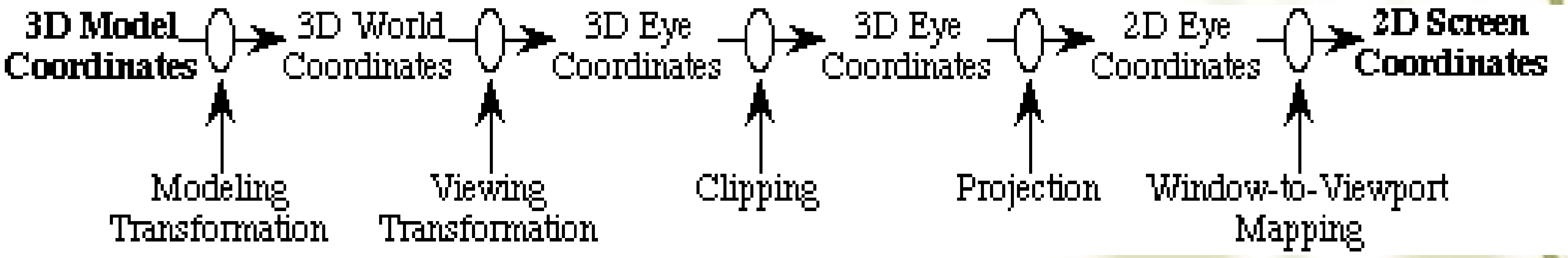
4. The Graphics Pipeline

- Computer graphics has two phases:
  - *Modeling* geometry from model to screen space
  - Creating the desired *image* in screen space

- For the graphics pipeline, they have two parts:
  - The *Modeling* pipeline
  - The *rendering* pipeline

Modeling Pipeline

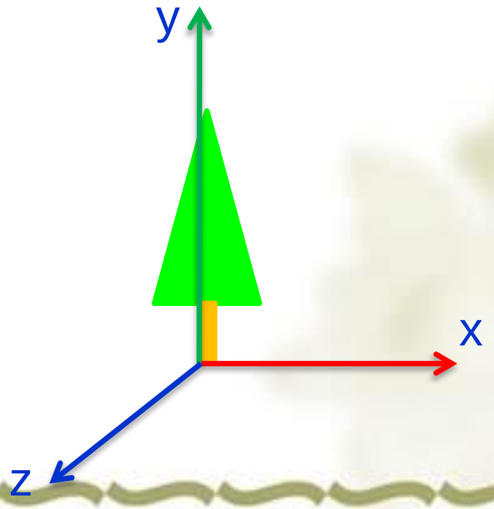
- We begin with the modeling(geometry) pipeline, because you have to start with geometry



- Notice that this pipeline involves several **spaces** and **transformations** between them

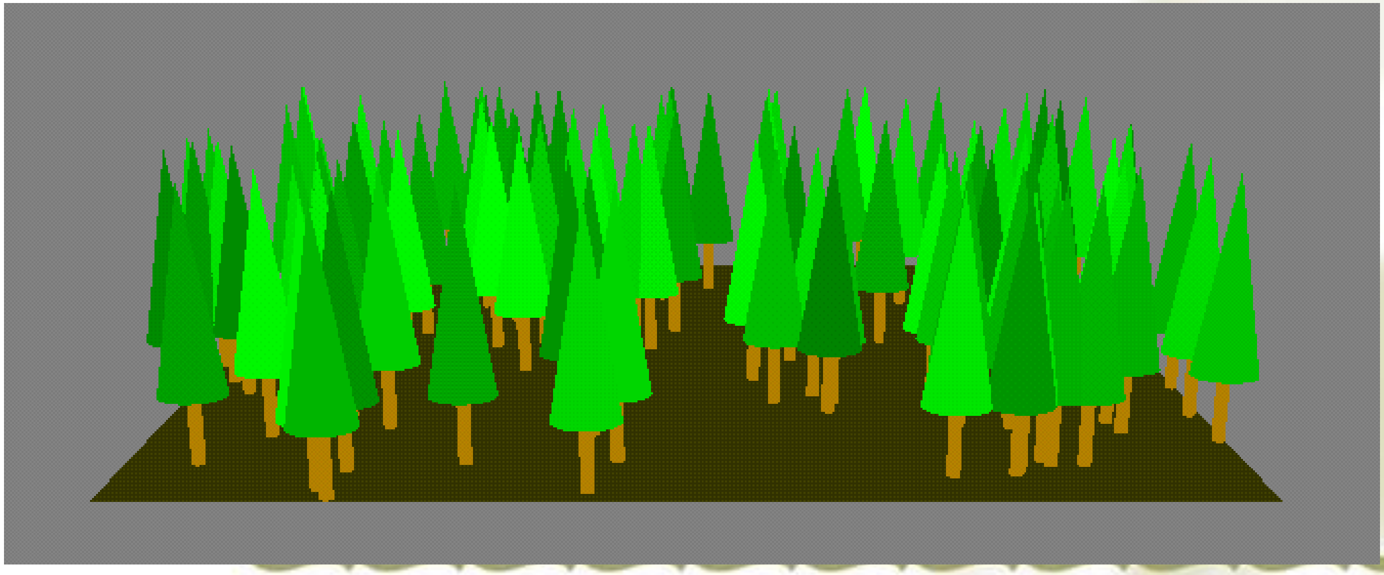
4.1. 3D Model Coordinates

- You define the parts of your model in whatever coordinates are natural for them
- You use modeling transformations to put the parts of your model together and then to place your model in a world space



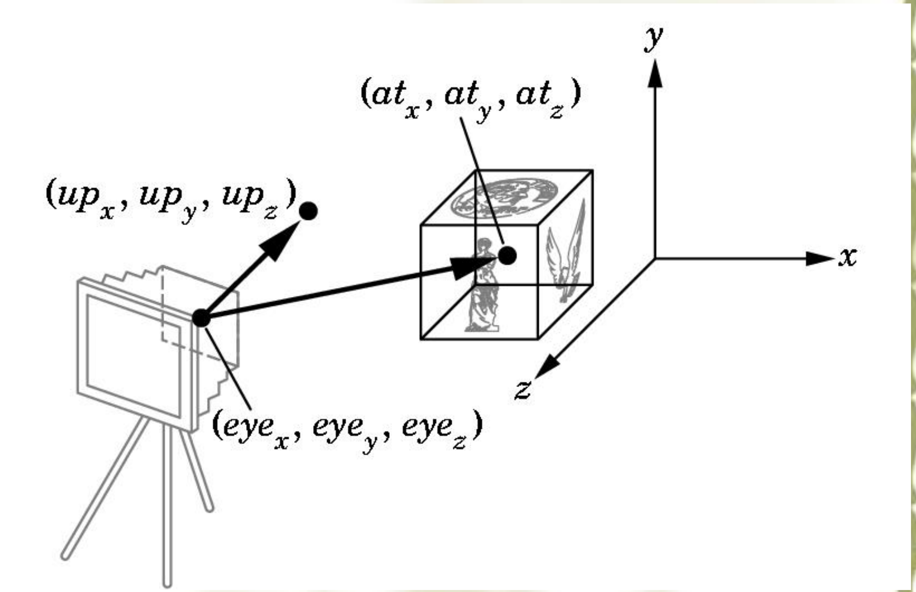
4.2. 3D World Coordinates

- This is a single 3D coordinate system in which all the parts of a scene are placed
- The scene is independent of the viewer

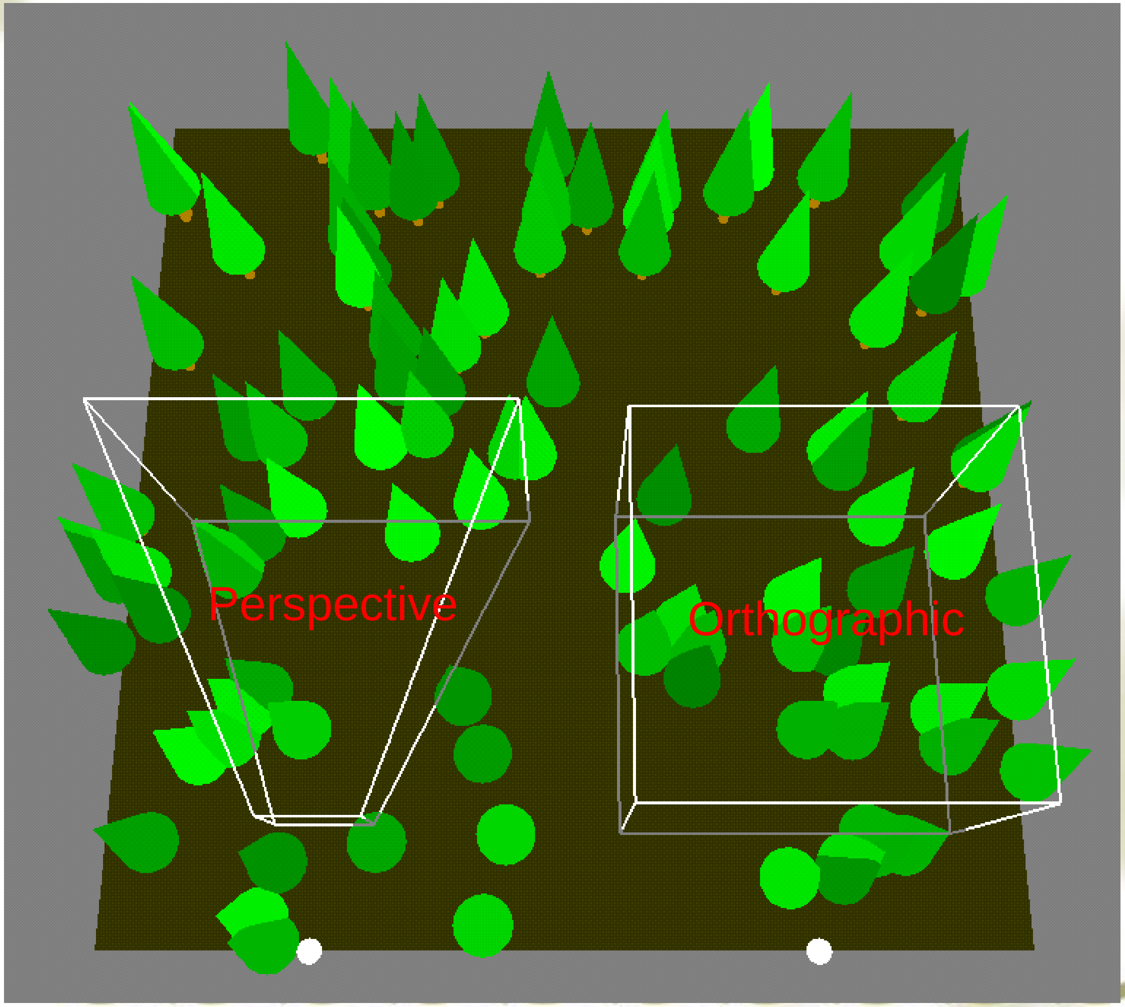


4.3. 3D Eye Coordinate System

- A scene becomes an image when there is a viewer and a viewing context
- A viewer (or camera) is placed in the world space with a position and orientation

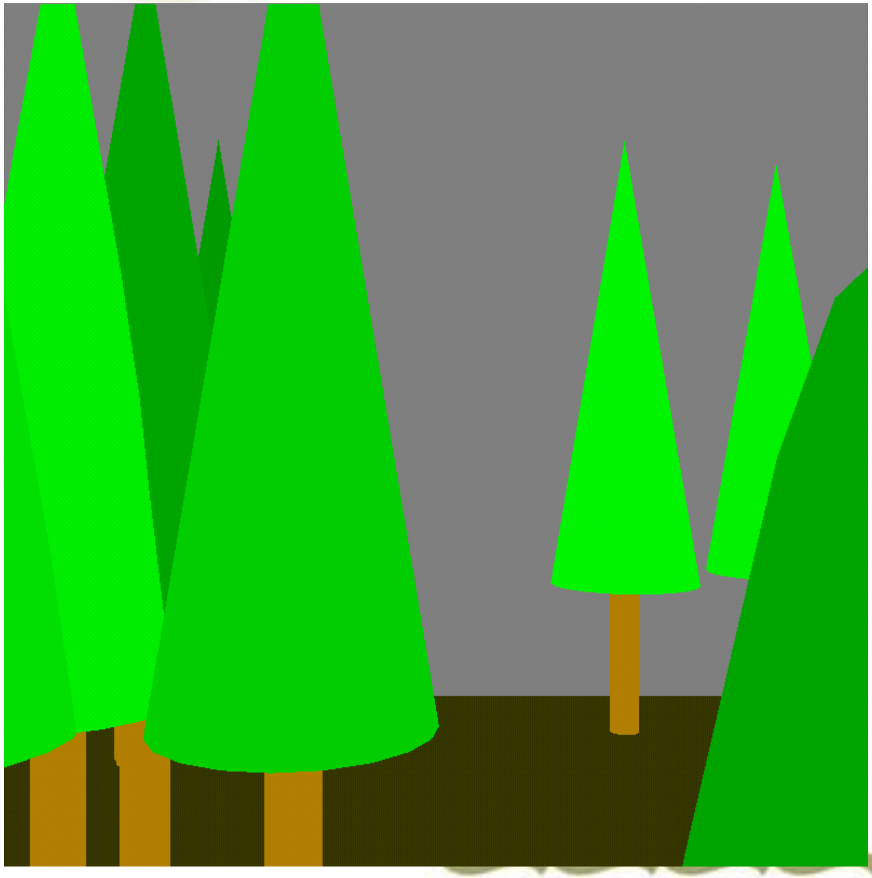


4.4. Perspective and Orthographic Viewing

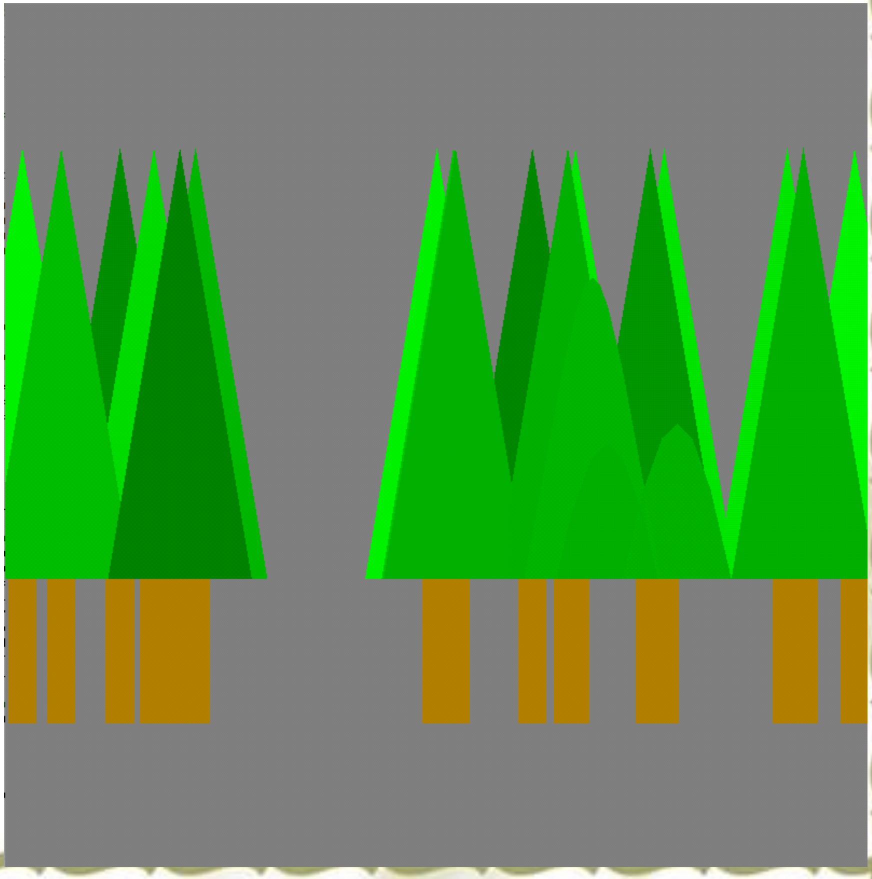


Perspective and Orthographics Views

- Perspective view

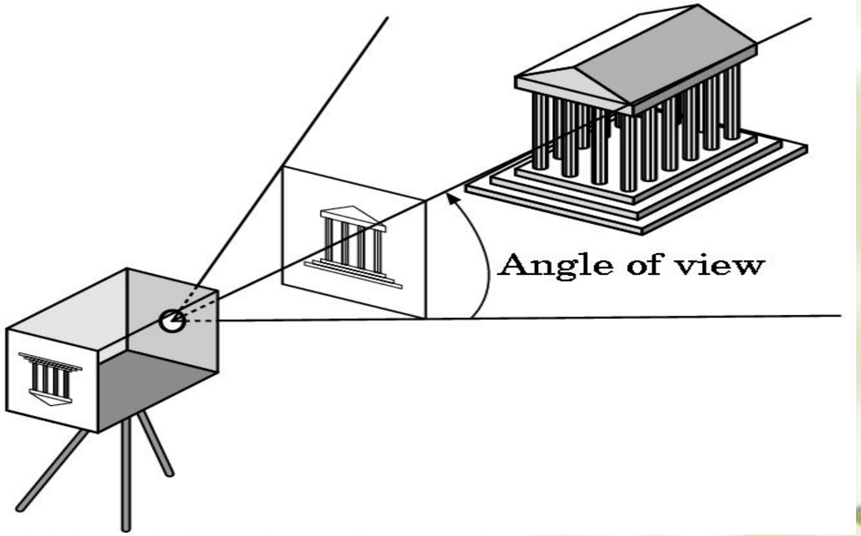


- Orthographic view



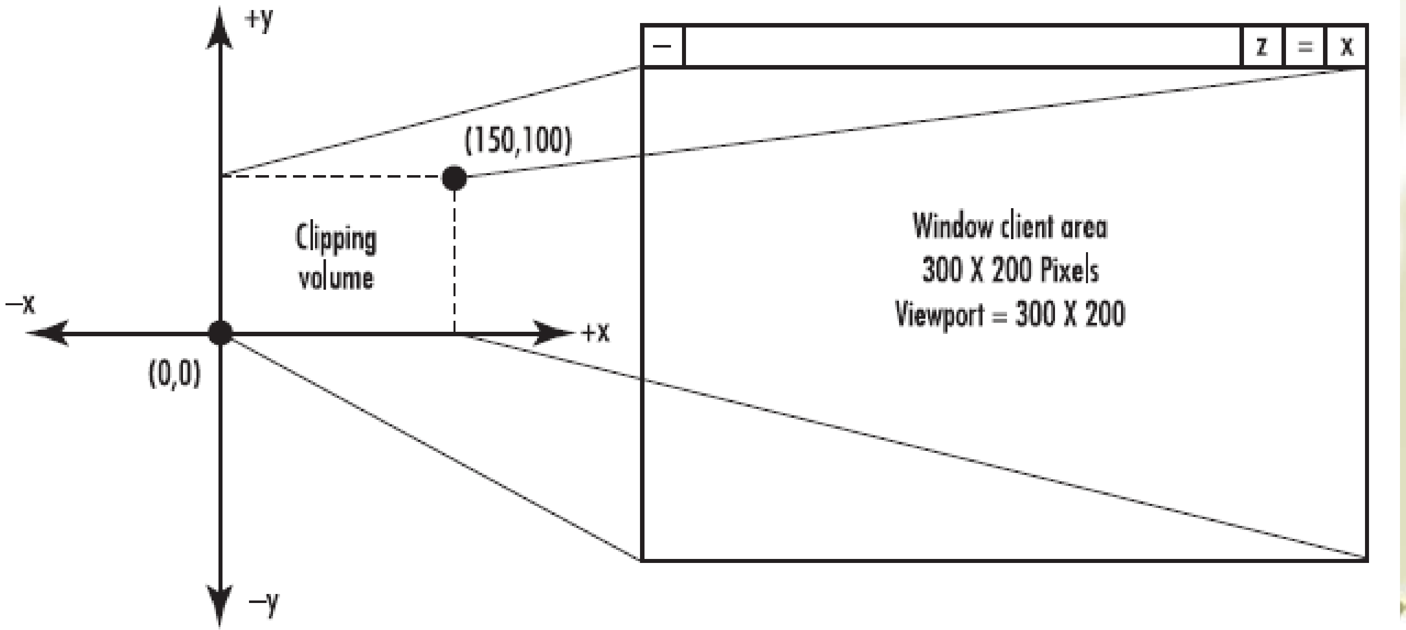
4.5. 2D Eye Coordinates

- The scene is transformed into this coordinate system by **projecting** each vertex in the scene to its corresponding point in the plane
- Depth information is lost in the view



4.6. 2D Screen Coordinates

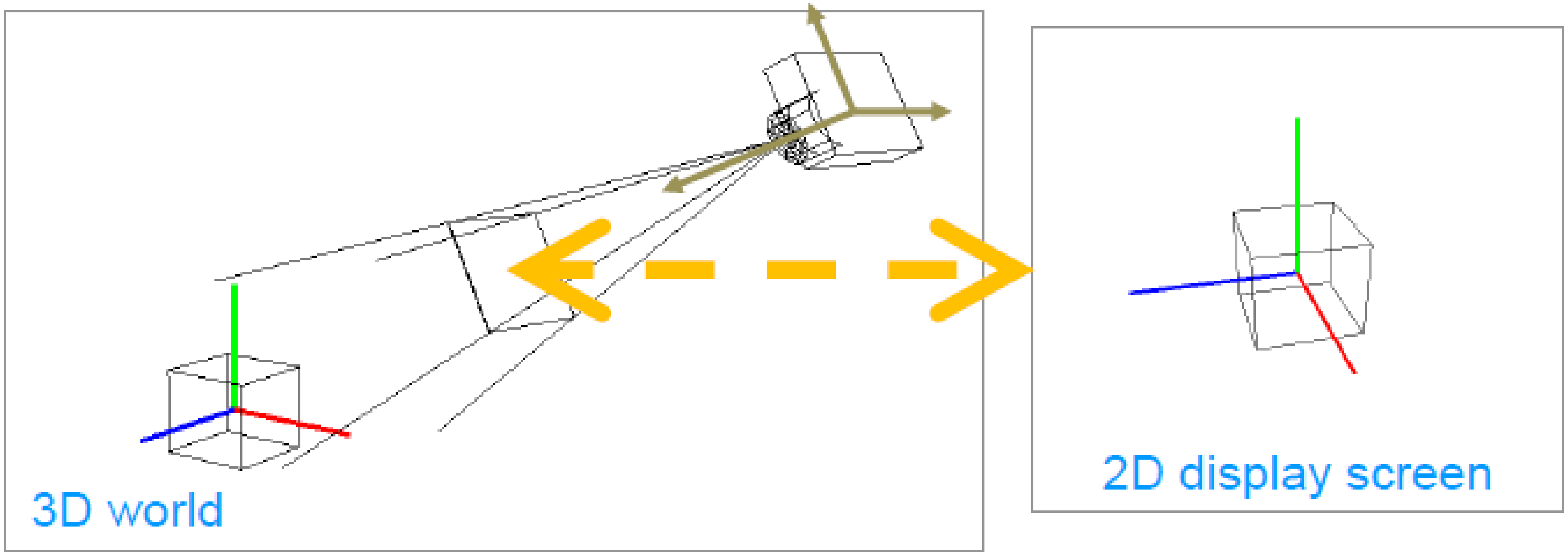
- The 2D eye coordinates are scaled to fit the screen dimensions
- The resulting **real** coordinate values are truncated to the **integer** coordinates that match the pixel addresses on the screen



Modeling Pipeline

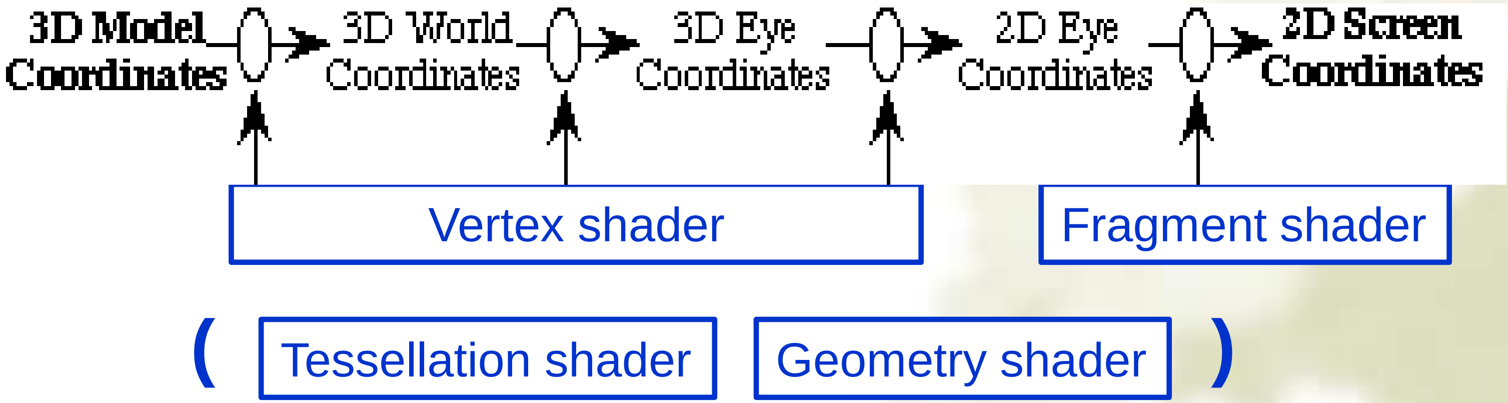
- By **Transformations** to implement the modeling pipeline

$$\boxed{p' = Mp}$$



★ Rendering Pipeline

- The modeling pipeline only maps vertices between the various spaces
- All vertices are processed to form an image in frame buffer

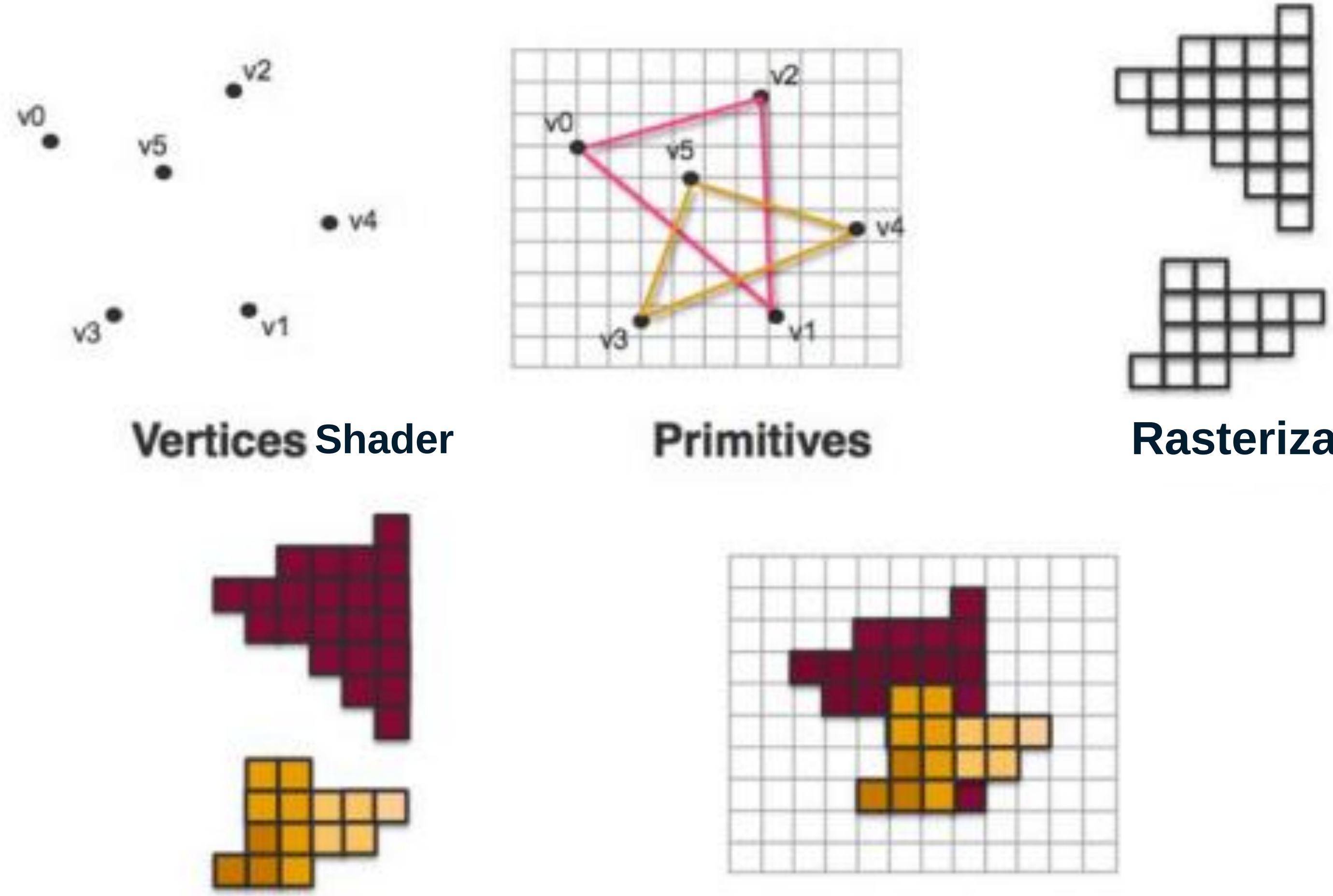


Vertex shader

Fragment shader

(Tessellation shader Geometry shader)

The Rendering Pipeline



Vertices Shader

Primitives

Rasterization

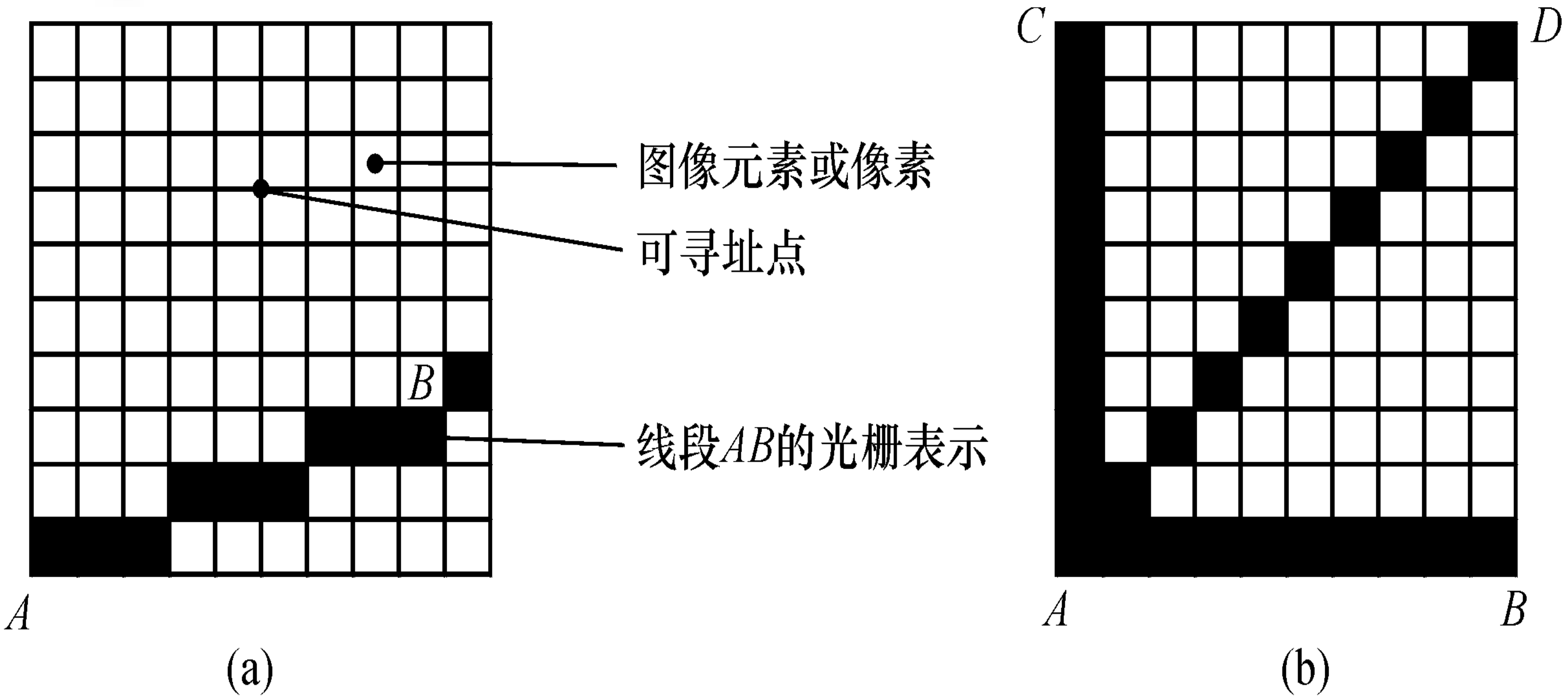
Fragments Shader

Pixels (Frame Buffer)

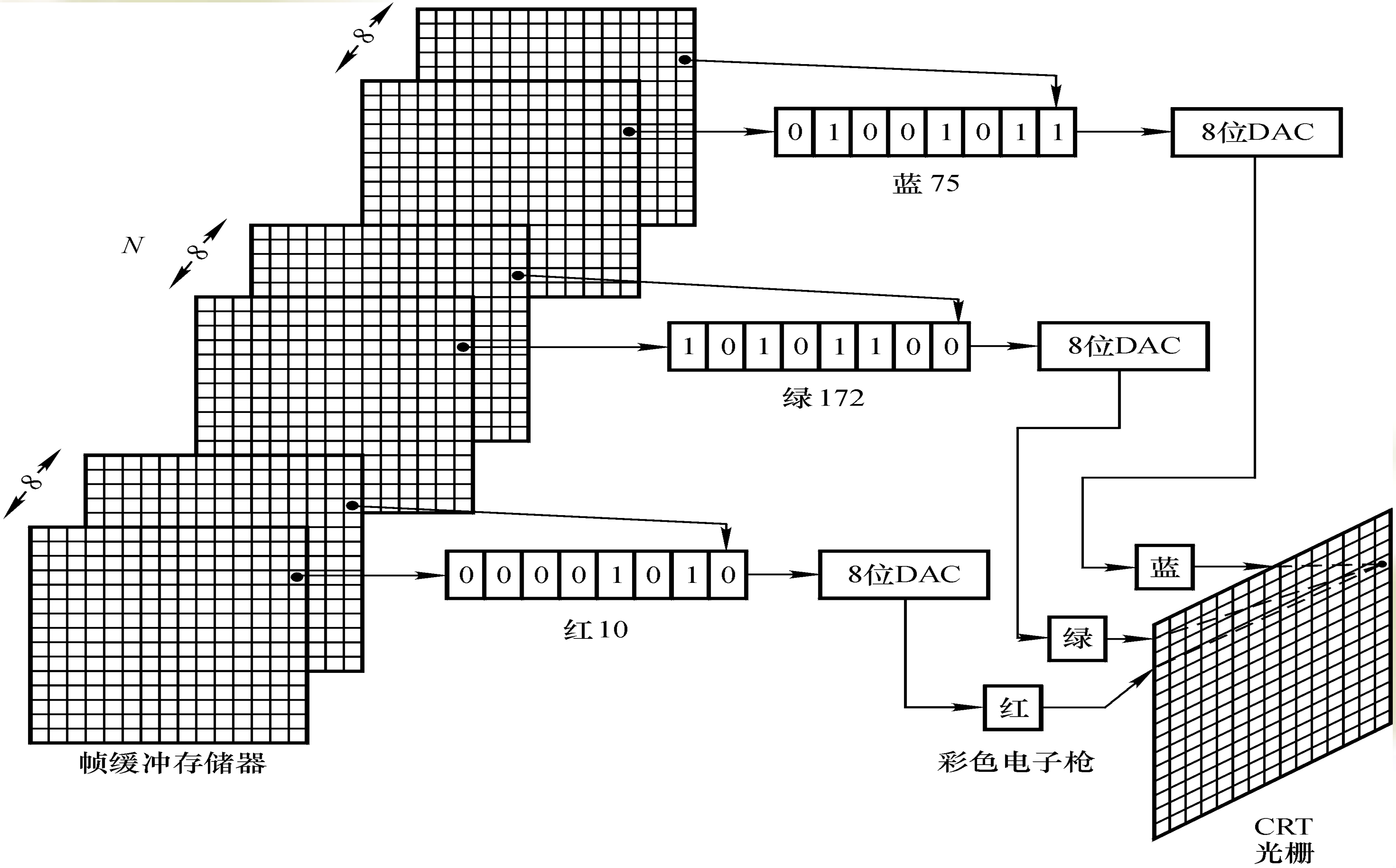
Rasterization - Scan Conversion

Frame Buffer = Screen

Pixel (picture element)

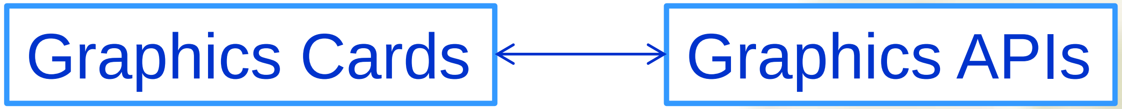


Full-Color Raster真彩色光栅显示器



Graphics Cards -- GPU

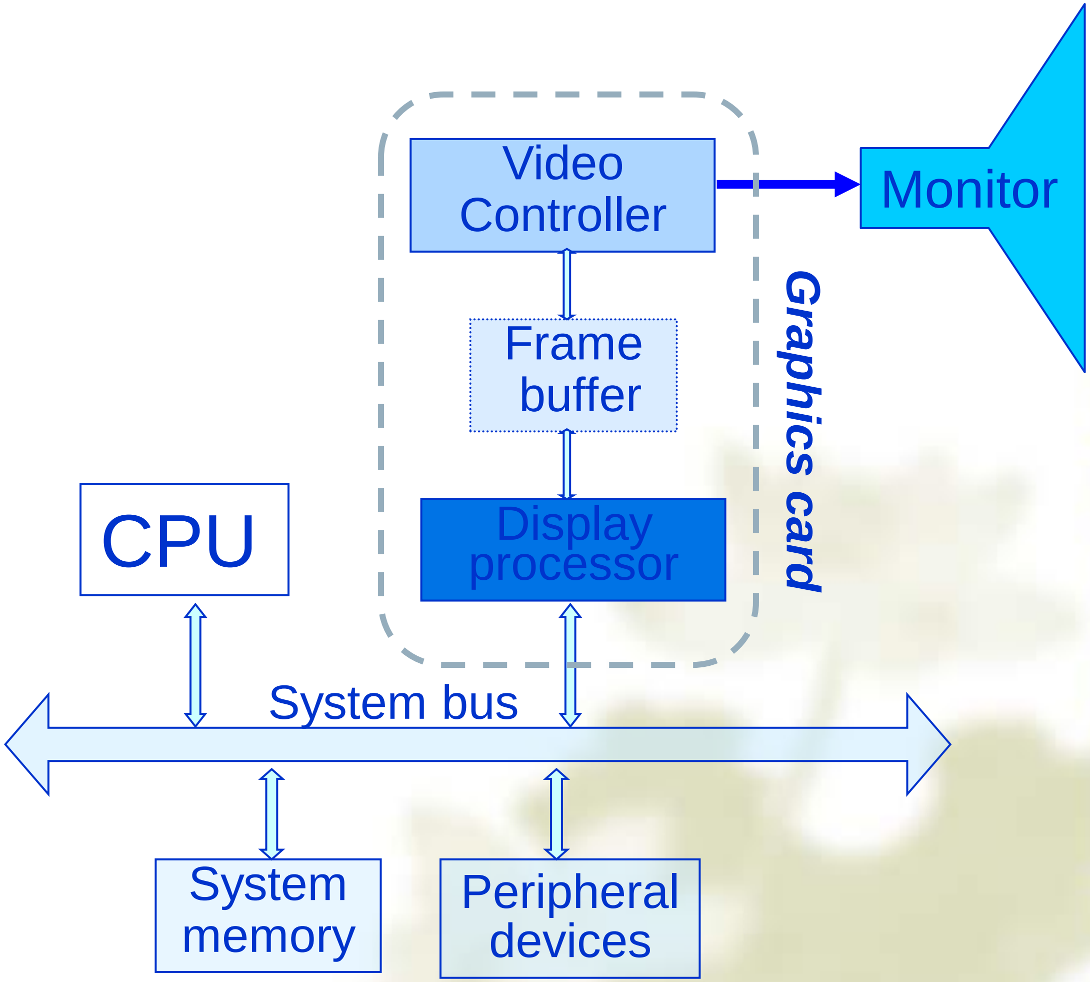
- Operating geometry data: vector, matrix, normal, product
- Rasterization processing
- Shader language for programming
- With graphics memory
  - huge memory & frame buffer



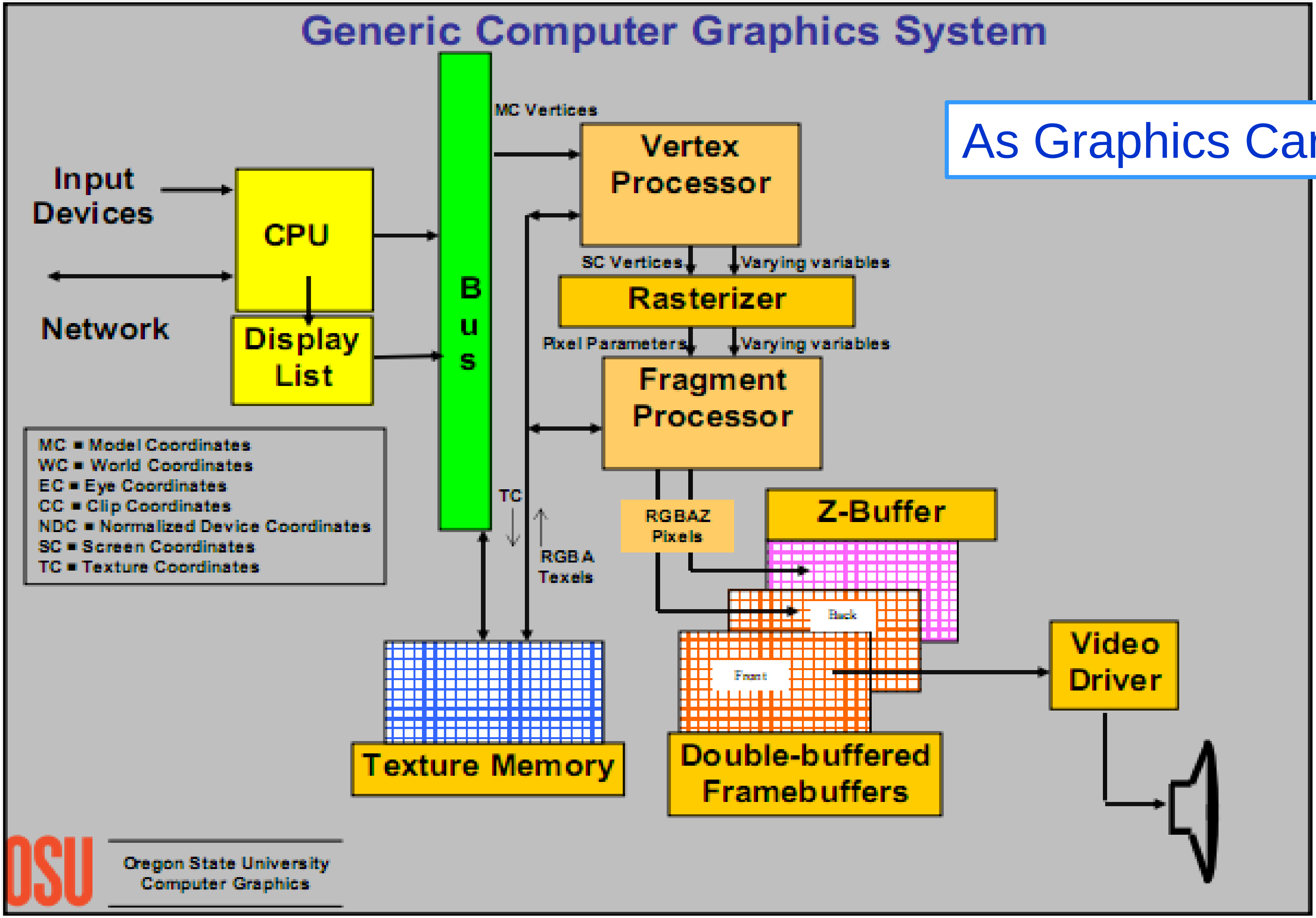
Video Display Devices

Frame buffer size:
For XVGA, altogether 3×1024×768 ≅ 2.4 Mbytes
Since there are other buffers and overhead, a typical memory of display card may contains 2 Gbytes RAM

Video controller:
cycles through the frame buffer **to refresh** the screen 30 - 120 times per seconds



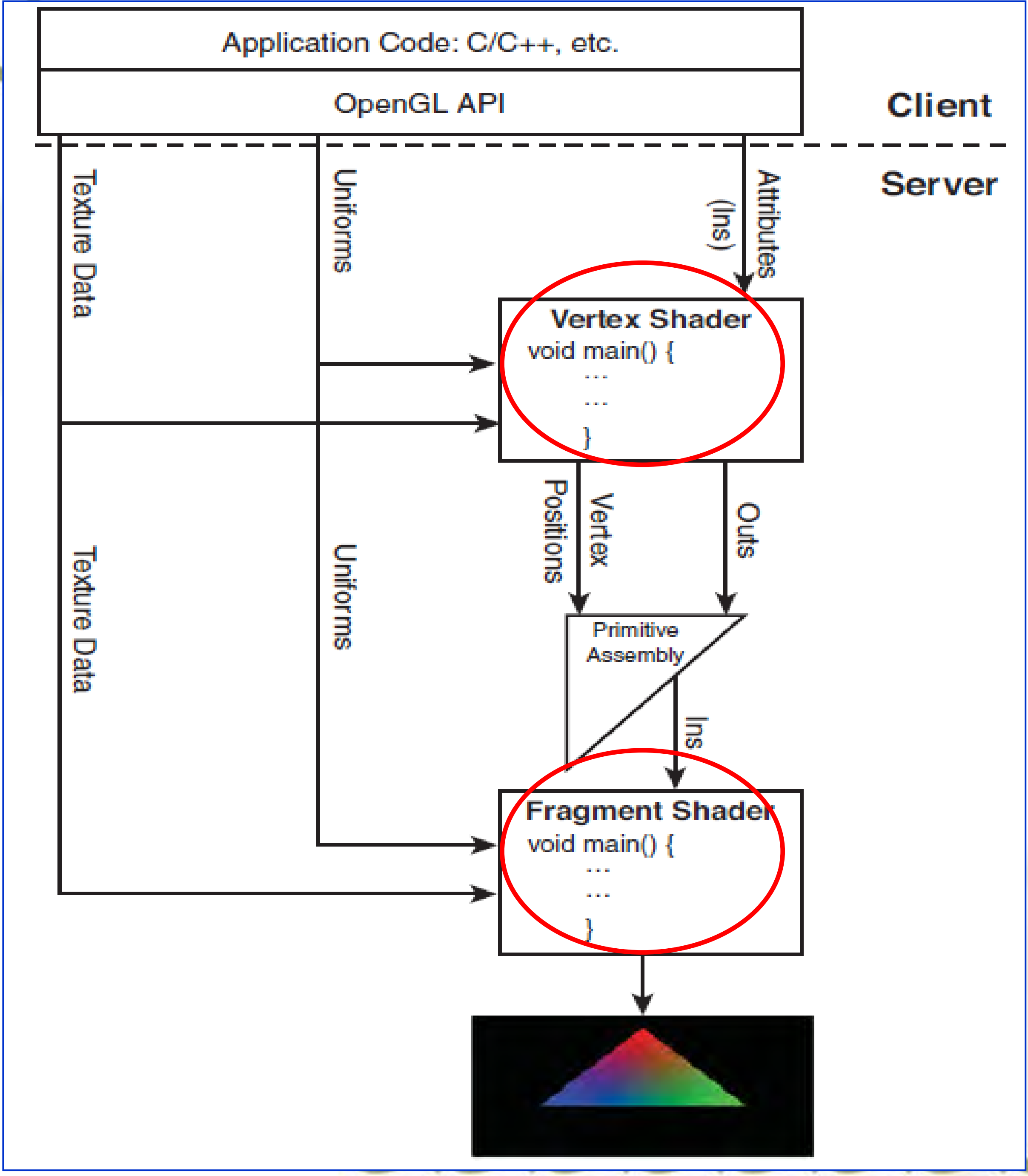
Graphics System



MC = Model Coordinates
WC = World Coordinates
EC = Eye Coordinates
CC = Clip Coordinates
NDC = Normalized Device Coordinates
SC = Screen Coordinates
TC = Texture Coordinates

As Graphics Cards

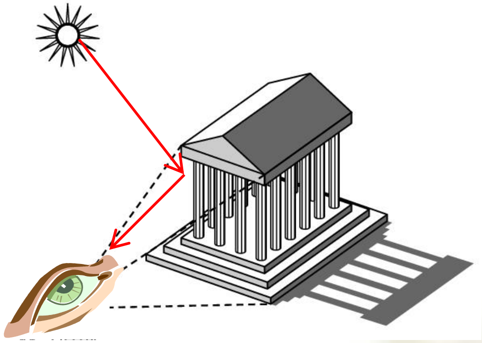
CPU-GPU **Client/Server** architecture



As Graphics APIs

5. Image Formation

- Objects
- Viewer
- Light source(s)



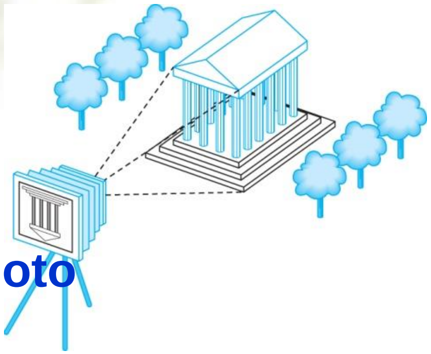
- Attributes that govern how light **reflects off** the materials（objects） to the viewer
- Note the independence of the objects, the viewer, and the light source(s)

Imaging System

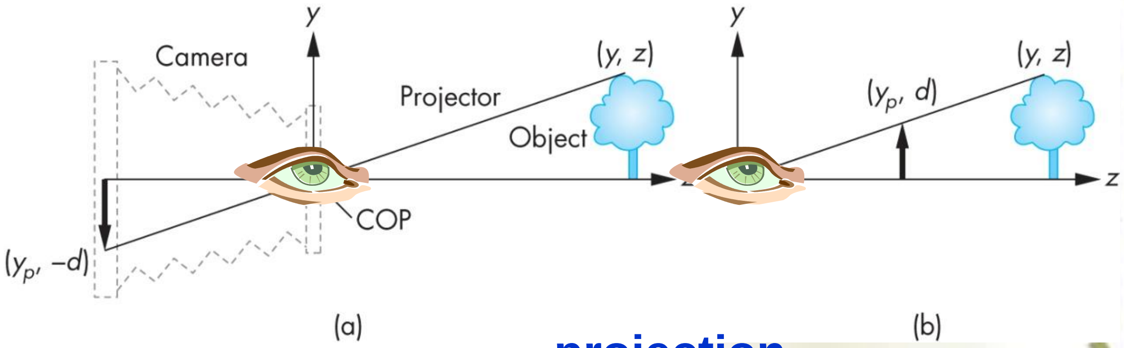
- In computer graphics, we form images which are generally two dimensional using a process analogous to how images are formed by physical imaging systems
  - Cameras
  - Microscopes
  - Telescopes
  - Human visual system
  - 3D film

Synthetic-camera Model

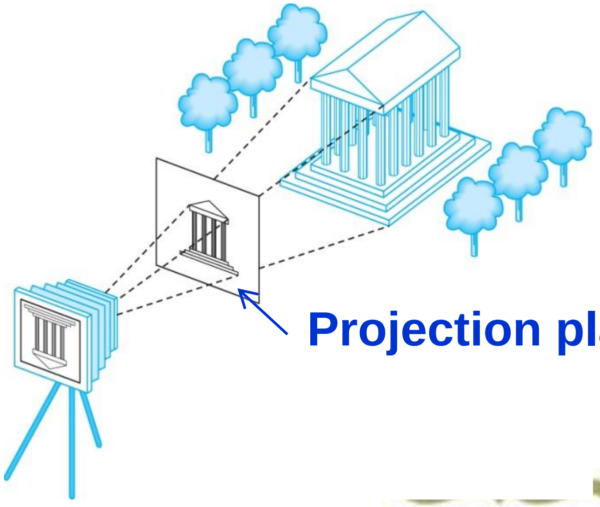
- 3D objects to 2D photo



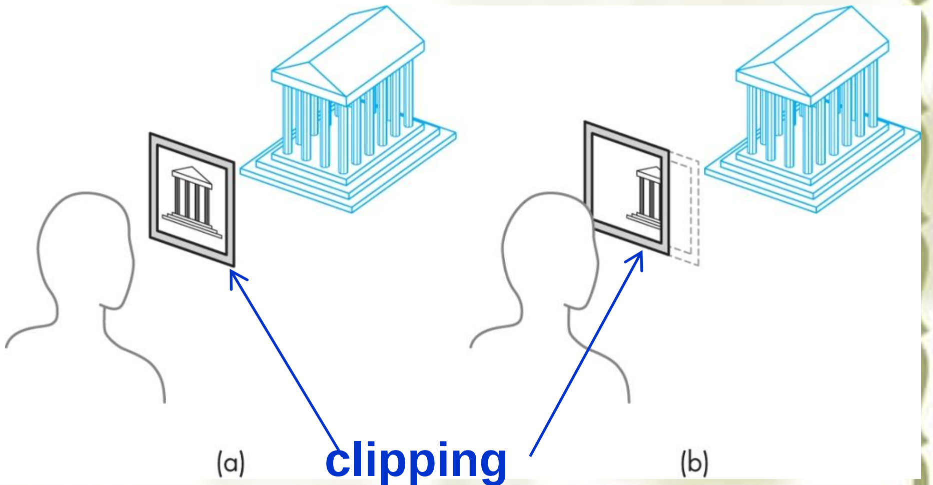
Digital photo



projection



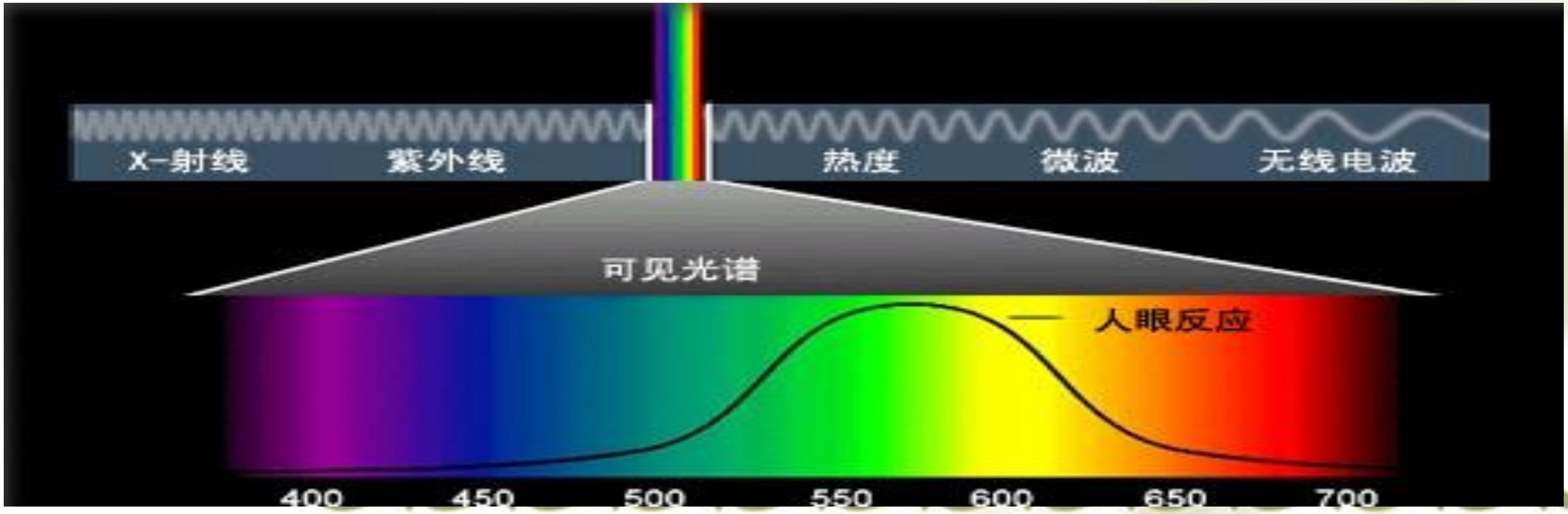
Projection plane



clipping

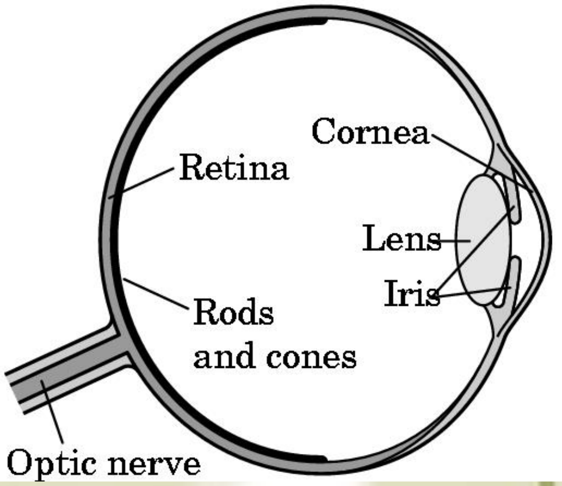
Light

- *Light* is the part of the electromagnetic spectrum that causes a reaction in our visual systems
- Generally these are wavelengths in the range of about 380-760 nm (nanometers)
- Long wavelengths appear as reds and short wavelengths as blues

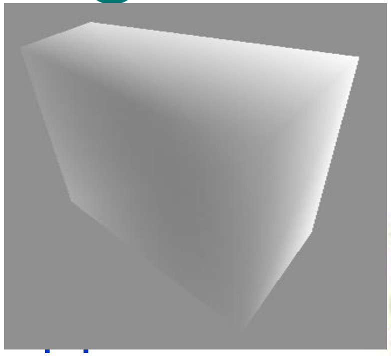


Three-Color Theory

- Human visual system has two types of sensors
  - Rods(杆状细胞): monochromatic, night vision
  - Cones(锥状细胞)
    - Color sensitive
    - Three types of cones
    - Only three values (the *tristimulus* values) are sent to the brain
- Need only match these three values
  - Need only three *primary* colors
- Screen is an emission display, not reflection



Luminance and Color Images

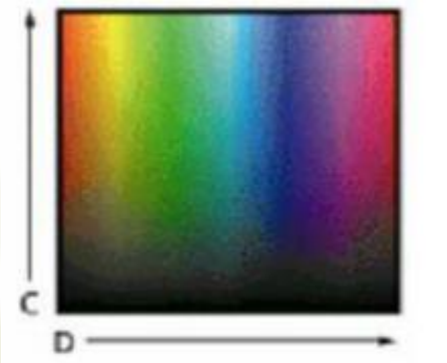


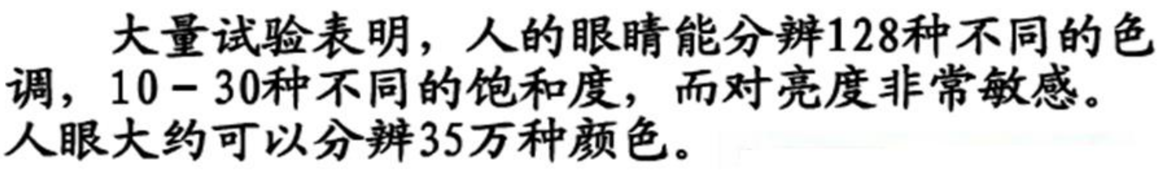
- Luminance Image
  - Monochromatic
  - Values are gray levels
  - Analogous to working with black and white film or television
- Color Image
  - Has perceptional attributes of

    hue(色调)

    saturation(饱和度)

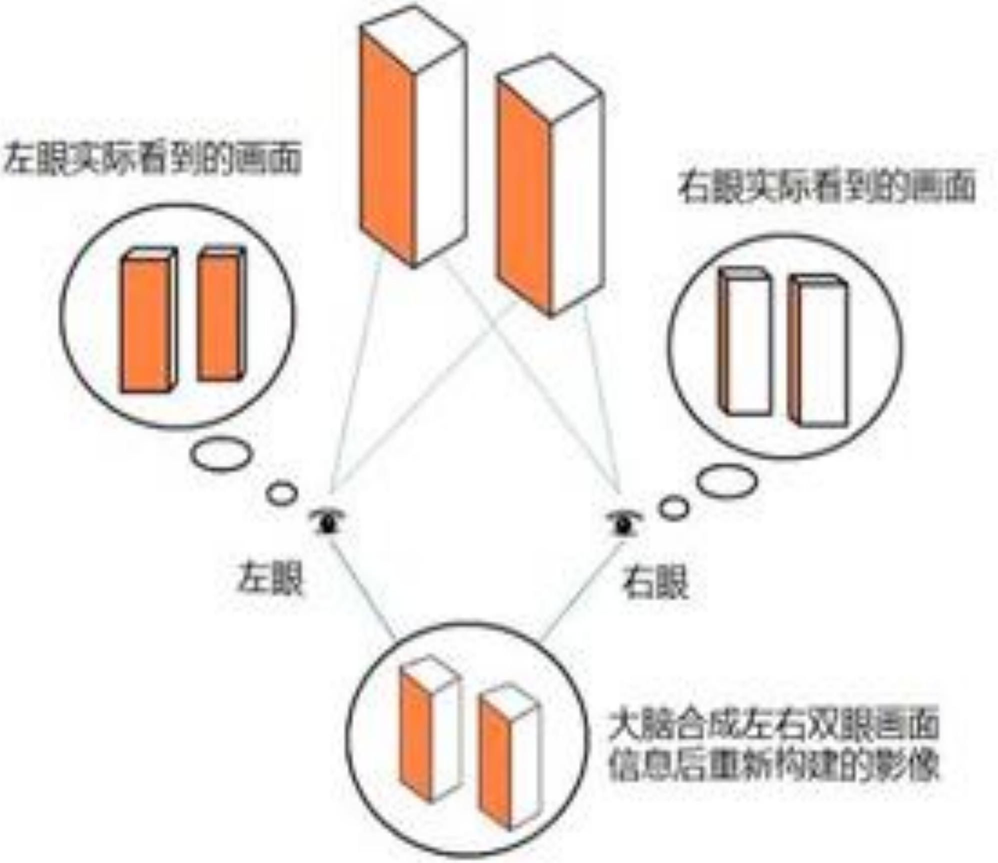
    lightness(亮度)
  - Do we have to match every frequency in visible spectrum? No!





大量试验表明，人的眼睛能分辨128种不同的色调，10－30种不同的饱和度，而对亮度非常敏感。人眼大约可以分辨35万种颜色。

3D Film


左眼实际看到的画面
右眼实际看到的画面
左眼
右眼
大脑合成左右双眼画面信息后重新构建的影像

Panasonic
3D

API Contents

- Functions that specify what we need to form an image
  - Objects(model) and Materials(attributes or texture)
  - Viewer(camera)
  - Light Source(s)
- Other information
  - Input from devices such as mouse and keyboard
  - Capabilities of system

作业1—第一章

1. What is the resolution of the image? What is the aspect ratio of the image?
2. Movies are generally produced on 35mm film that has a resolution of approximately 2000x3000 pixels. What implication does this resolution have for producing animated images for television as compared with film?
3. 查找图形卡有哪些主要品牌？简述其中一种的主要指标有哪些?
4. 简述OpenGL、OpenGL ES和WebGL的相同和不同的特性。

第二周提交，提交方式：教学在线