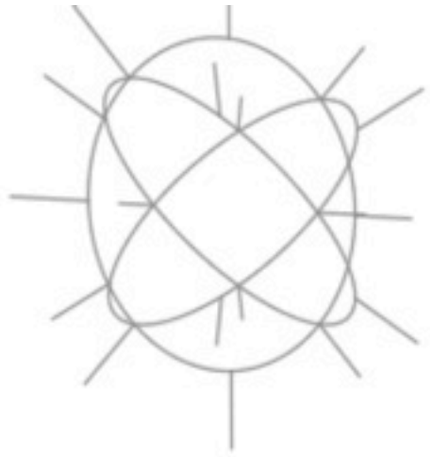
Computer Graphics Introduction to 3D

EDA 221

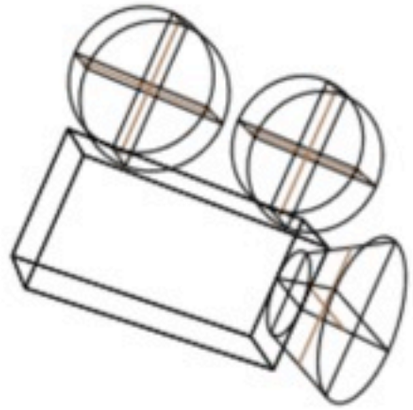
Jacob Munkberg

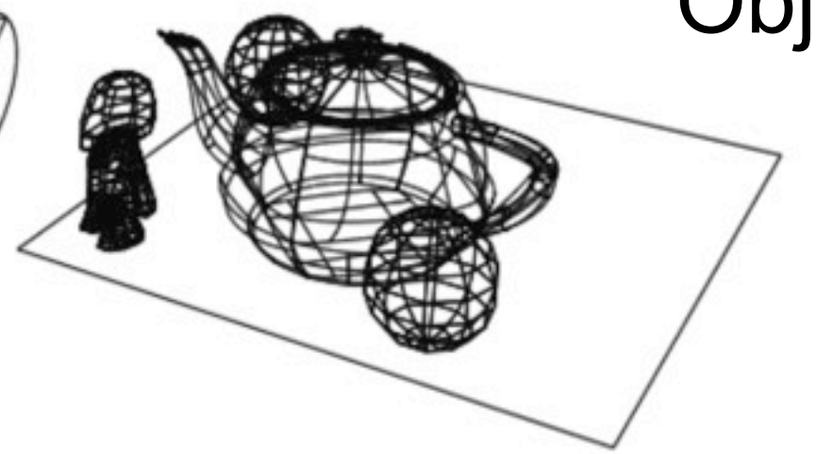
Create Virtual Scenes



Light source

Camera





Objects

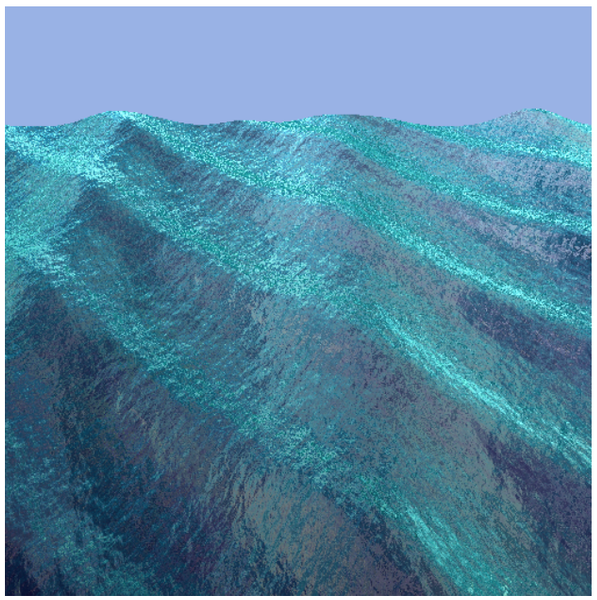
Rendered Image

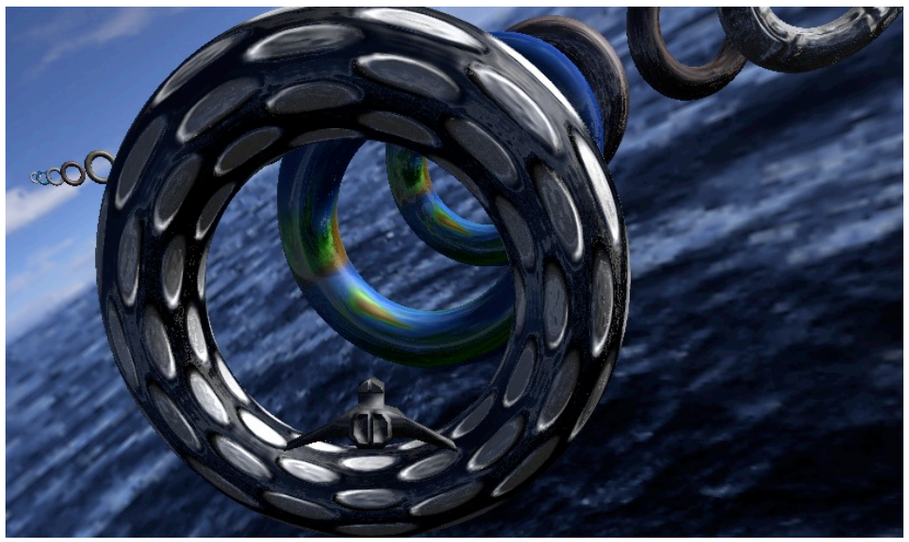
Course Goals

- Introduction to Computer Graphics
- Create and **position** objects in 3D
- Create materials - write **shaders**
- Visualize 3D scenes - **render** images
- Introduction to **OpenGL**
  - Create interactive graphics

Assignments

- Five mandatory programming assignments
- Done in pairs
- Seminars Thursdays 13-15 in E:C





Organization

- Lectures : Theory and concepts
- Seminars
  - Applied theory & hands-on examples
- Assignments
  - Computer Graphics in practice, C++ & GLSL
- Examination
  - All five assignments approved
  - Written exam: Oct 22, 14-19 in Eden 022/026

Website

- http://cs.lth.se/eda221/
- Lectures will be posted online
- Online discussion forum
- Booking system for assignment approval sessions
- Code & assignments
- Links

Course Material

- Literature
  - Edward Angel, Dave Shreiner, **Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL**, Pearson Education, 6th edition
  - Lectures & Seminars
  - Handouts for Lab 4
- Prerequisites
  - EDAA01 & Linear Algebra

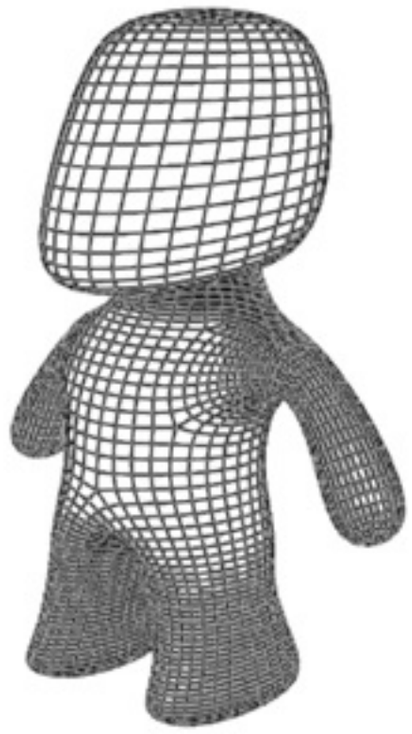
Staff

- Jacob Munkberg - Lectures, Exam
  - Graphics researcher at Intel Corporation
  - PhD in Computer Graphics from LTH
- Carl-Johan Gribel - Seminars, Labs
  - PhD student in Computer Graphics
- Rasmus Barringer - Labs
  - PhD student in Computer Graphics

Definitions

Geometry

Mesh



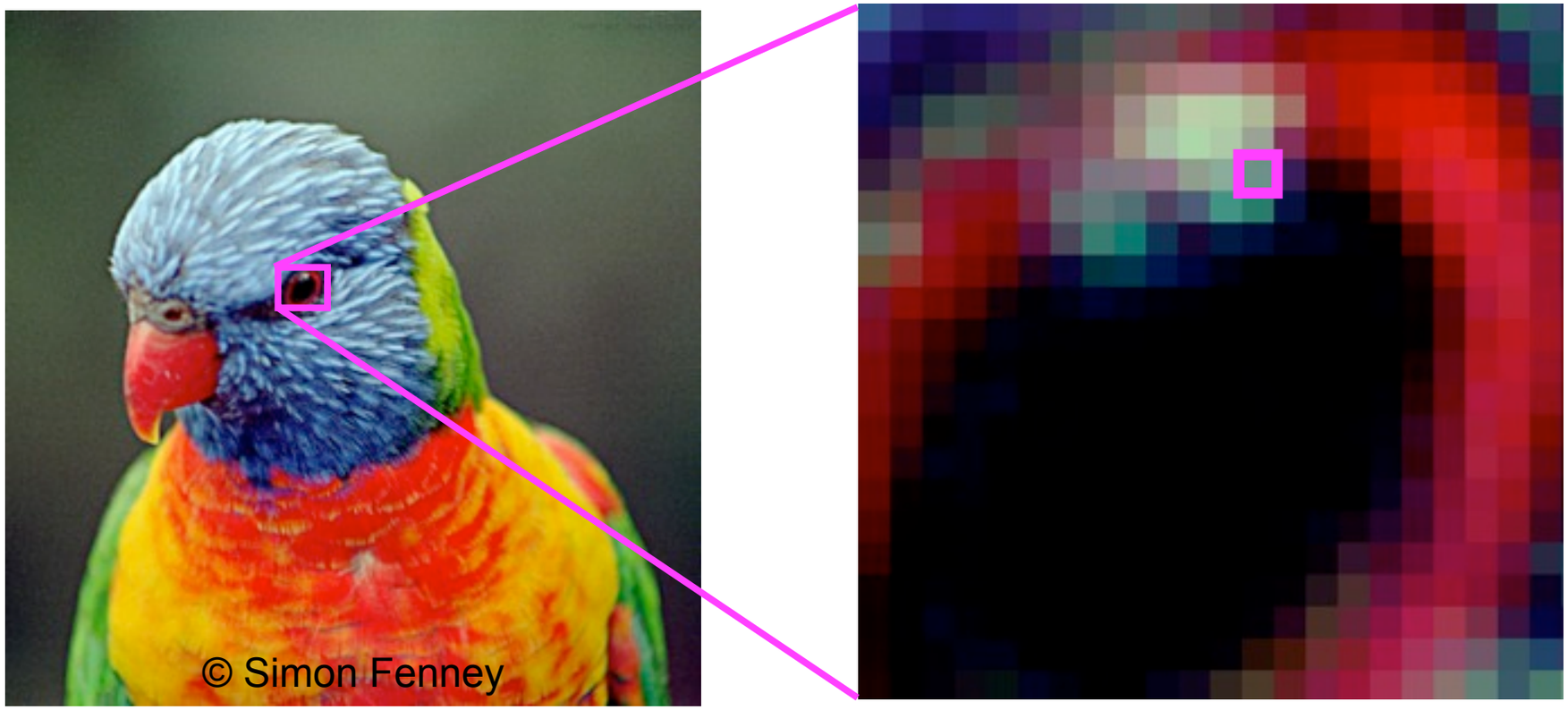
Tessellated Mesh

Lit & Shaded Mesh

We need ways to represent geometry and move objects in three-dimensional coordinate systems

Pixel

- Pixel - Picture element
- Our task - compute color of each pixel

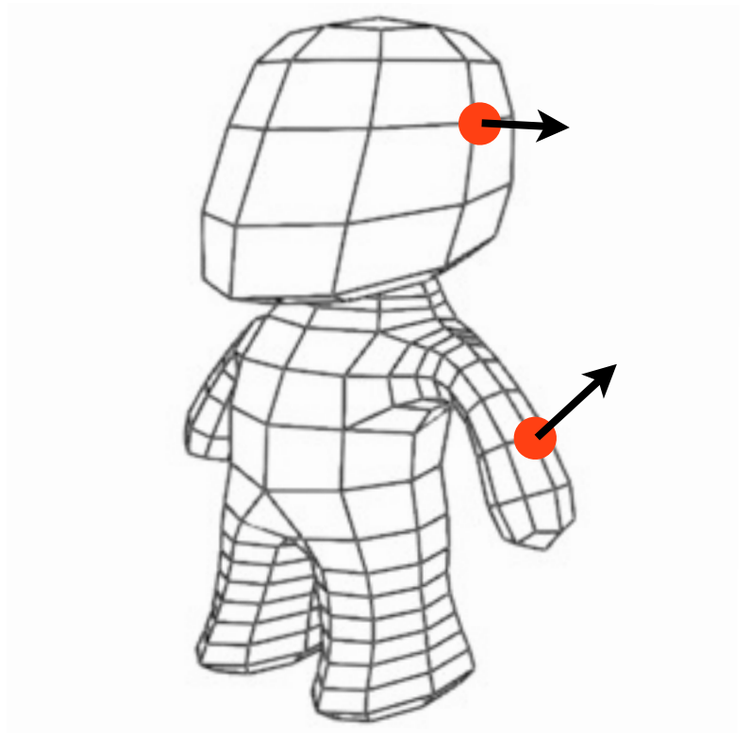


512x512 pixels

22x22 pixels

Vertex

- A set of attributes describing a point in space



```
struct Vertex
{
    float x,y,z;        // pos
    float nx, ny, nz; // normal
    float r,g,b;        // color
};
```

Materials

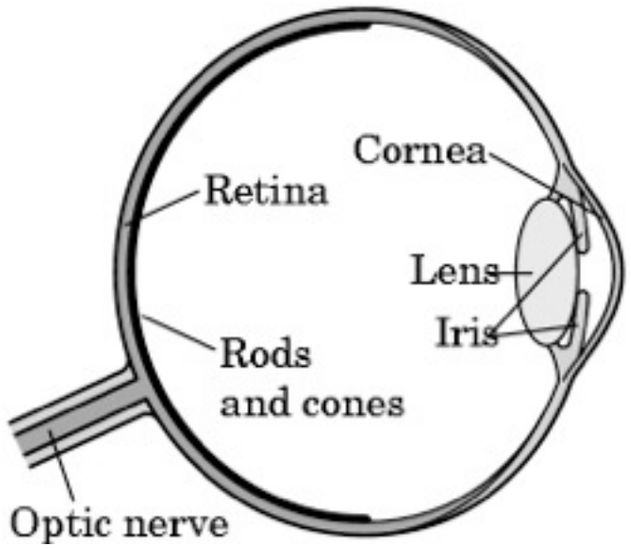
- Determine the appearance of objects
  - How objects interact with light
- Specified as small programs called **shaders**

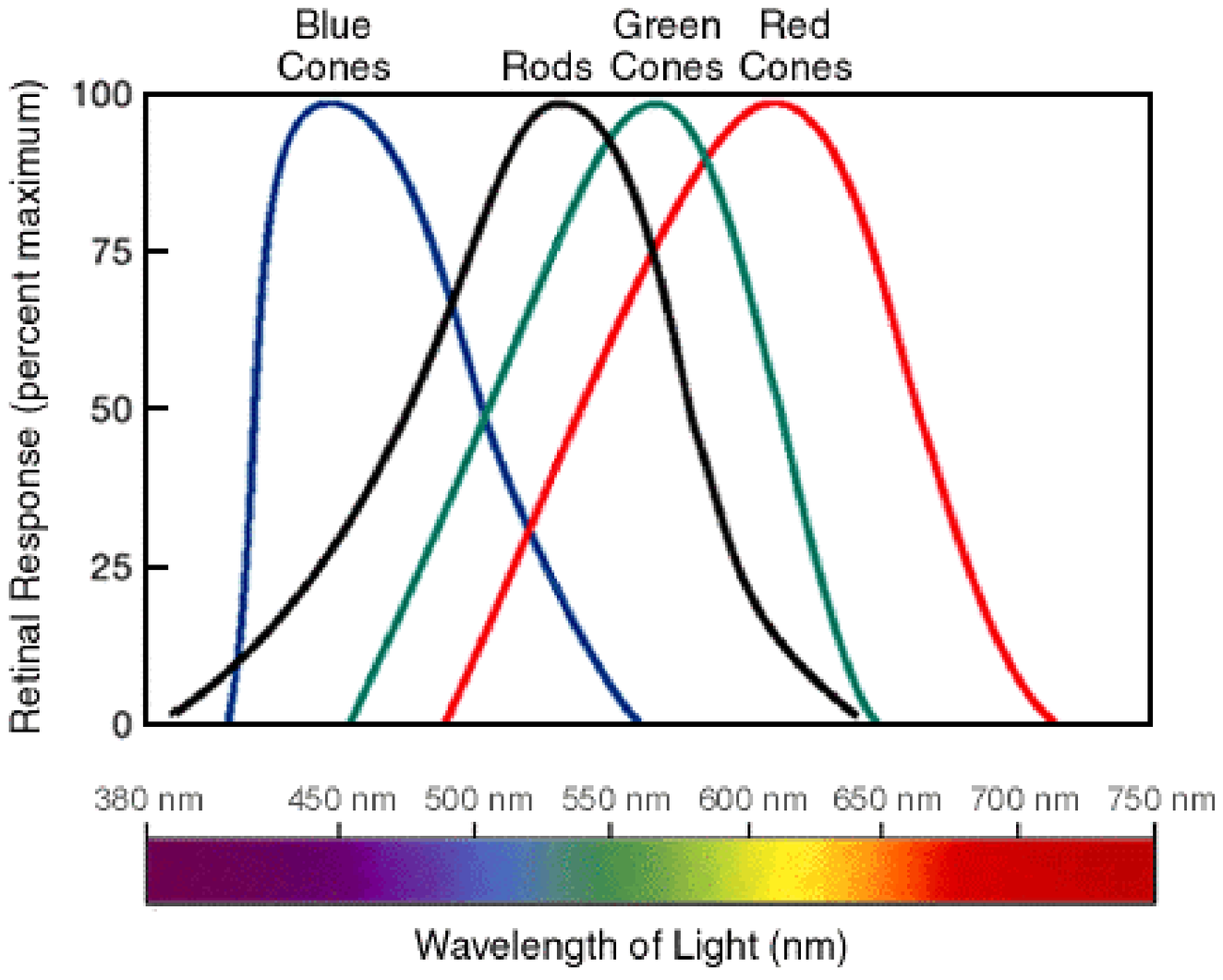
Light

- Light is the part of the electromagnetic spectrum that causes a reaction in our visual systems
- Wavelengths in 380-750 nanometers
- Long wavelengths appear as reds and short wavelengths as blues

Color Perception

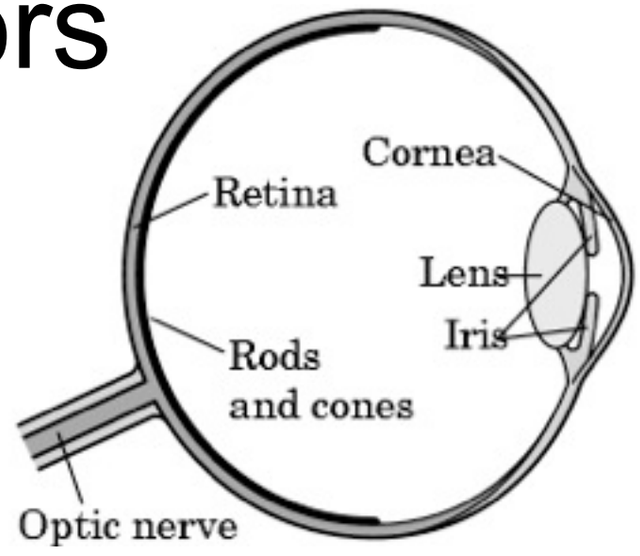
- Color stimulates cones in the retina
- Three different kinds of cones





Human Visual System (HVS)

- HVS has two types of sensors
  - **Rods**: monochromatic, night vision
  - **Cones**: Color sensitive, three types of cones

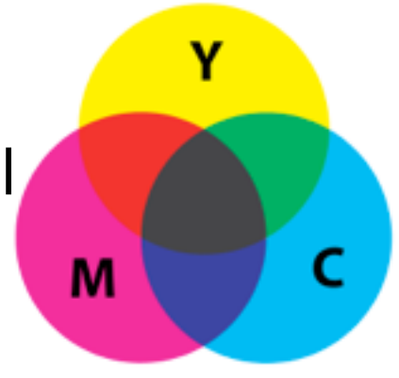


  - Only three values (the *tristimulus* values) are sent to the brain
- Need only match these three values
- Need only three primary colors: RGB

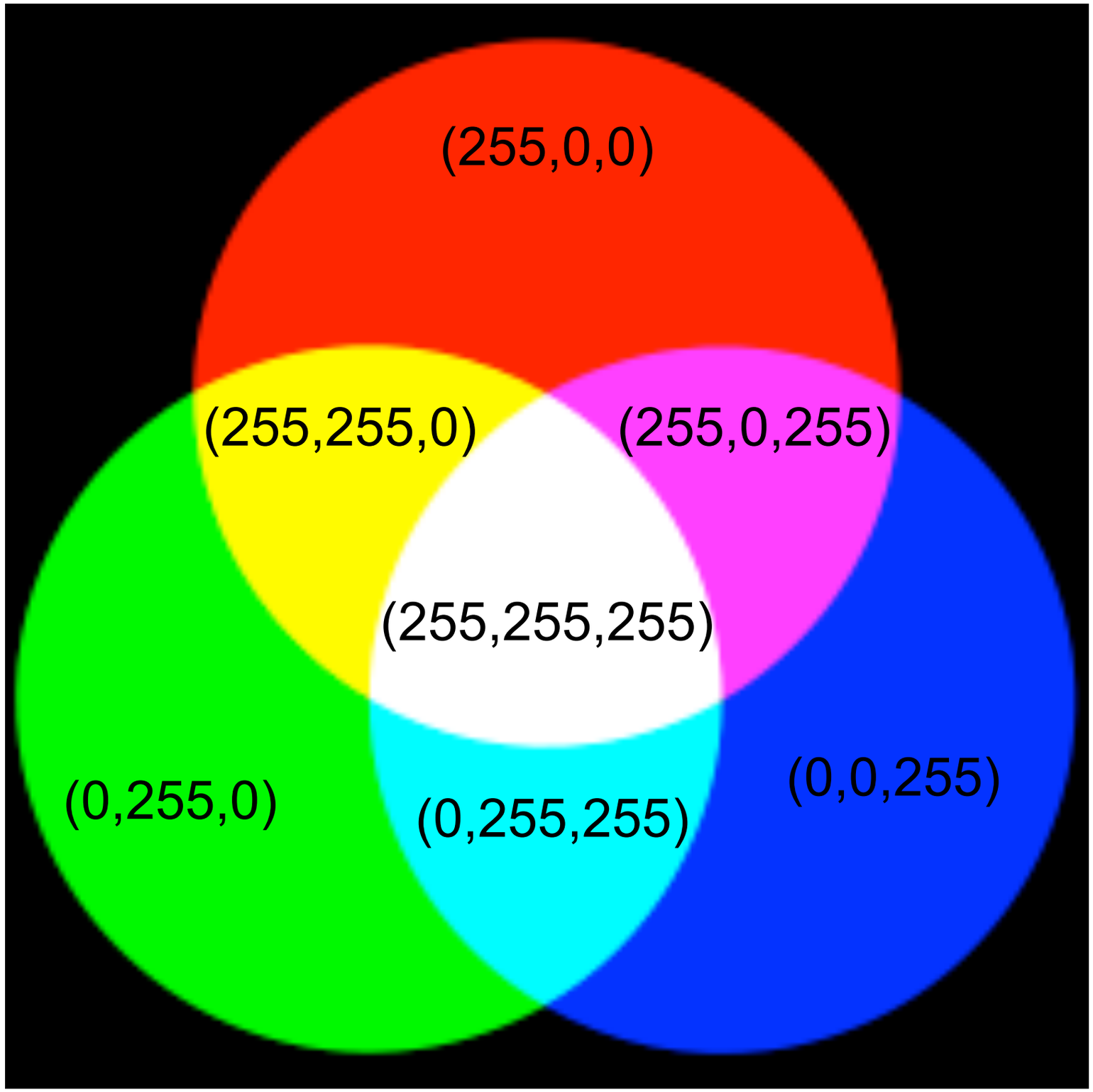
Color

- Additive color
  - Form a color by adding amounts of three primaries: Red (R), Green (G), Blue (B)
  - Often stored using 8 bits per primary which gives $(2^8)^3$ = 16.8M unique colors

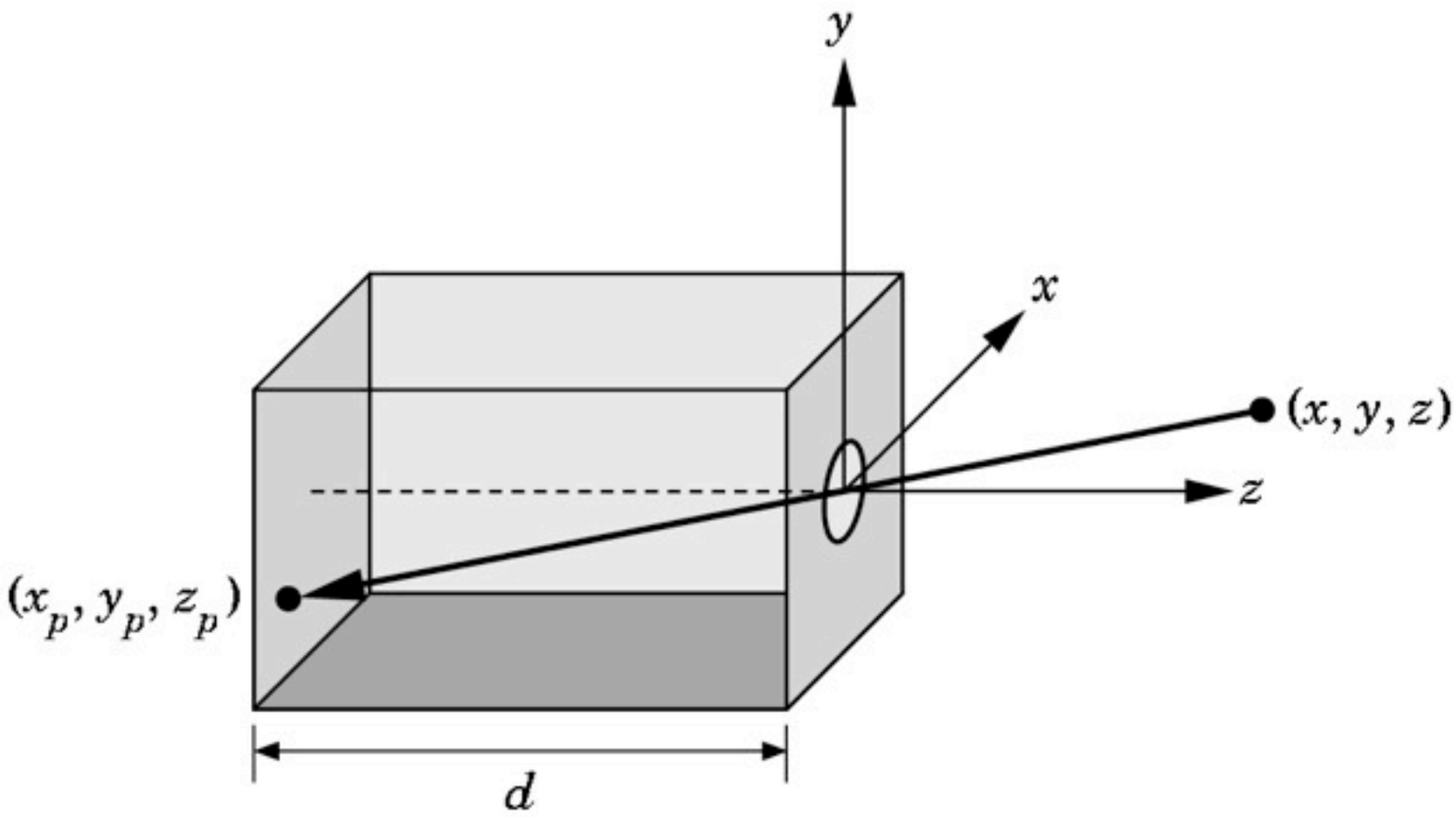
- Subtractive color
  - Form a color by filtering white light with Cyan (C), Magenta (M), and Yellow (Y) filte
  - Light-material interactions, printing

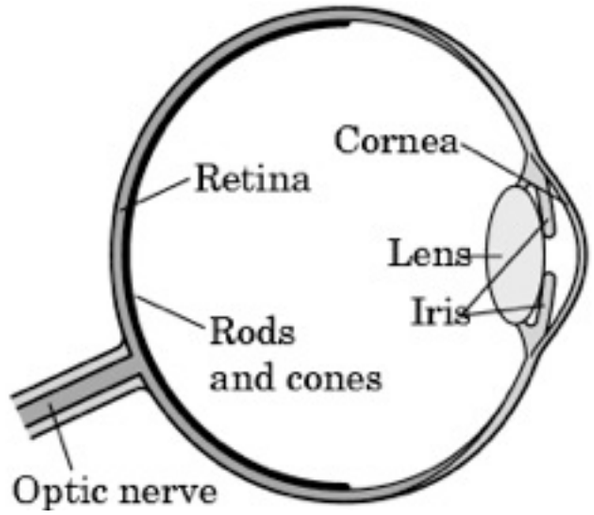


RGB Color



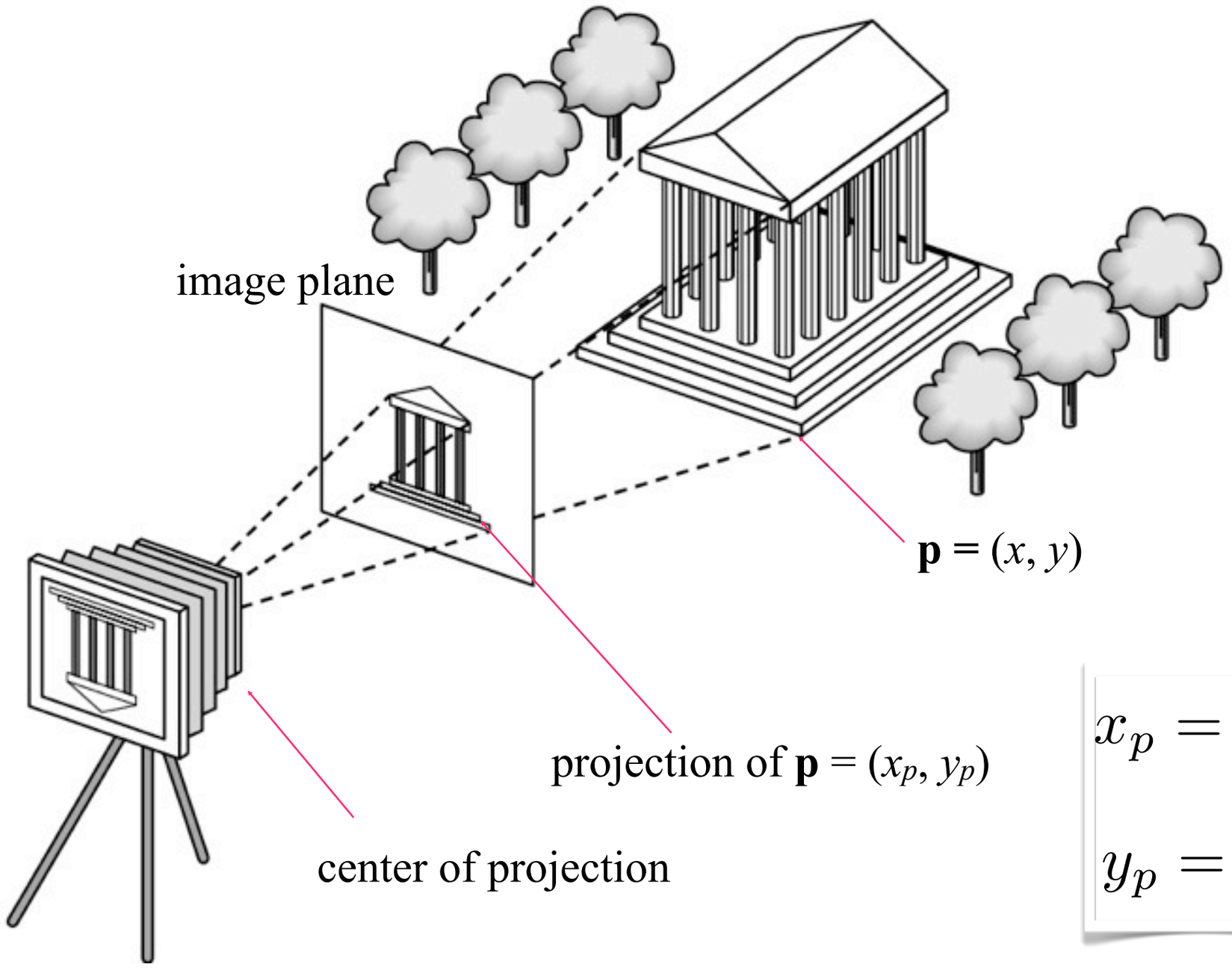
Pinhole Camera





- Projection of a 3D point (x,y,z) on image plane: $x_p = -d\frac{x}{z}, \quad y_p = -d\frac{y}{z}$
  - Equal triangles: $\frac{x}{z} = \frac{x_p}{z_p} \Leftrightarrow x_p = z_p\frac{x}{z} = -d\frac{x}{z}$

Synthetic Camera Model



$$x_p = d\frac{x}{z}$$

$$y_p = d\frac{y}{z}$$

Image Formation

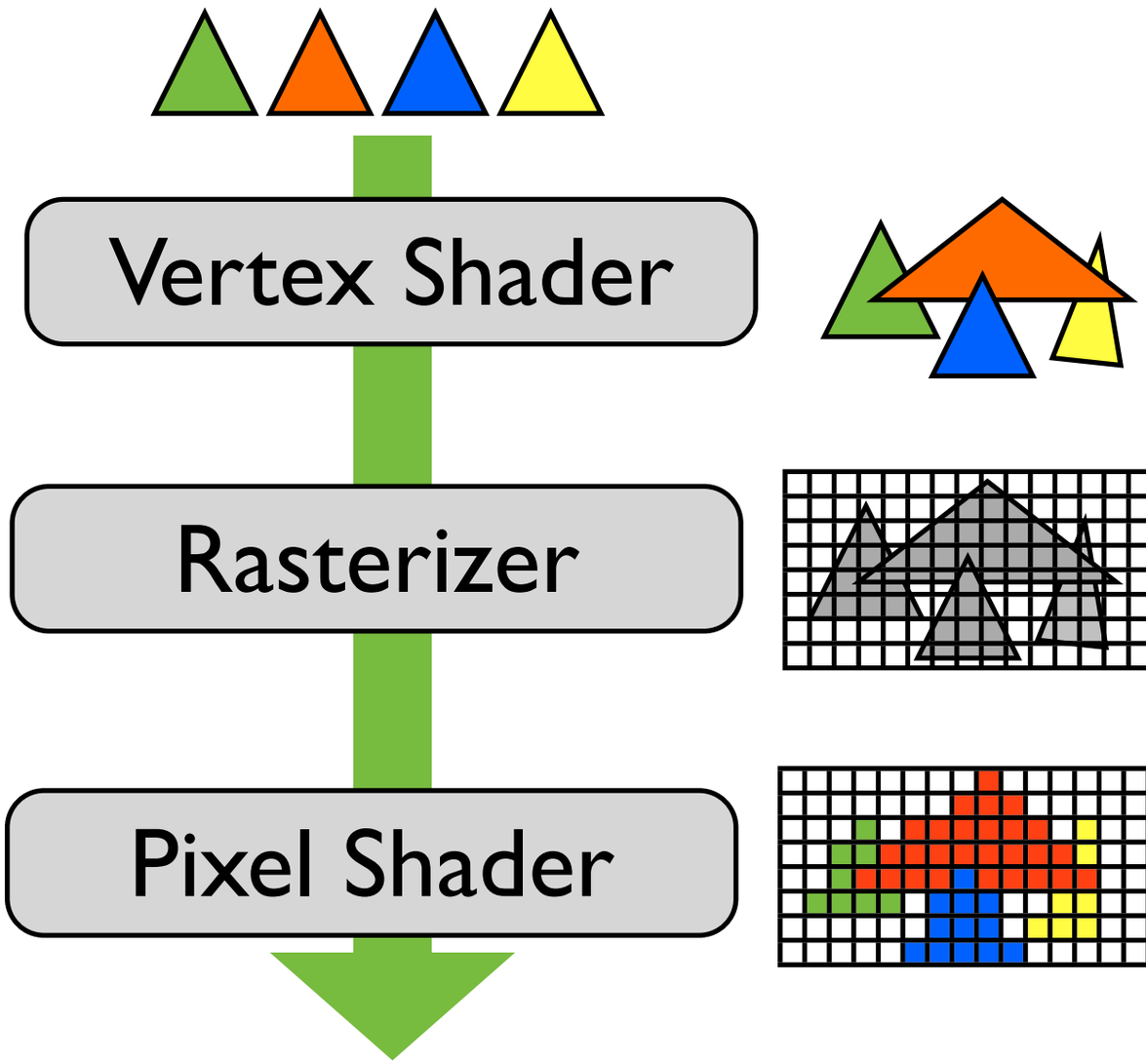
- Create geometric models
  - Position the models in a 3D scene
  - Assign materials to each model
  - Add lights & position a virtual camera
- For each pixel - find visible object
- Compute color of pixel based of the visible object's material and light

Challenges

- Create geometric models 1M triangles
  - Assign materials to each model
  - Position the models in a 3D scene (Move 1M tris)
  - Add lights & position a virtual camera
- For each pixel (5M), compute which (of the 1M) objects that is visible
- Compute color of pixel (5M) based from the object's material

Graphics Hardware

- **Pipeline** that accelerates the costly tasks of rendering



Move geometry

Compute visibility per pixel

Compute color per pixel

Real-time (Battlefield 3)

Example images at:

http://dice.se/games/battlefield-3-close-quarters/

Offline (Pixar - Ratatouille)

Example images at:

http://renderman.pixar.com/view/pixar-ratatouille-shading-food

Offline Rendering

Example images at:

http://forums.cgsociety.org/forumdisplay.php?f=121&daysprune=-1&order=desc&sort=voteavg

Real-time vs Offline

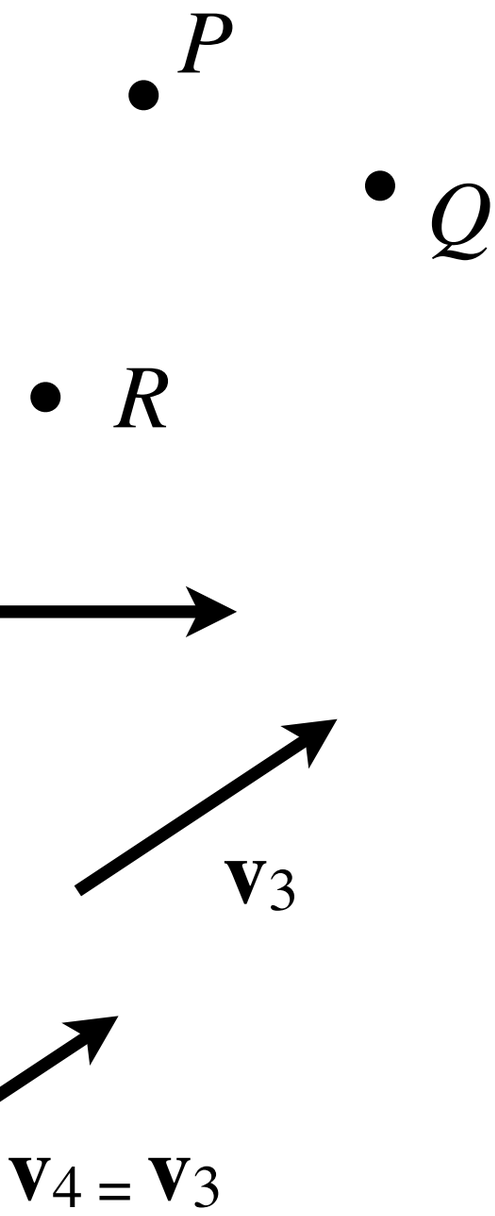
- Real-time
  - Render image in ~20 ms
  - Instant feedback
  - User interactions
- Offline (feature films)
  - Each image may take hours or days
  - Photorealism
  - No user interaction

Linear Algebra

- Points vs Vectors
- Angles between vectors
- Dot (scalar) product
- Cross product
- Coordinate system

Points & Vectors

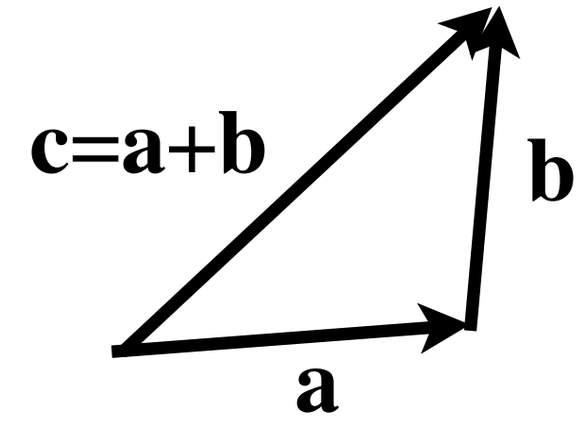
- A **point** is a location in space



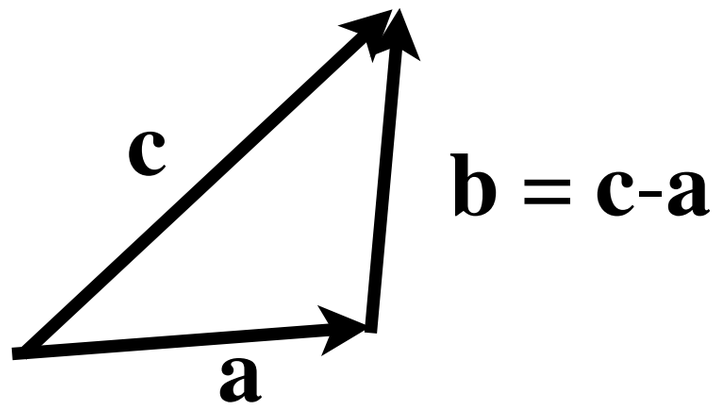
- A **vector** represents a direction and magnitude

Basic Operations

- Vector-vector addition

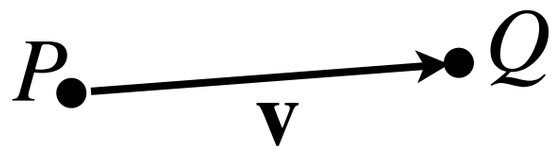


- Vector-vector subtraction



- Point-point subtraction

$$\mathbf{v}=Q-P \Leftrightarrow Q = P + \boldsymbol{v}$$



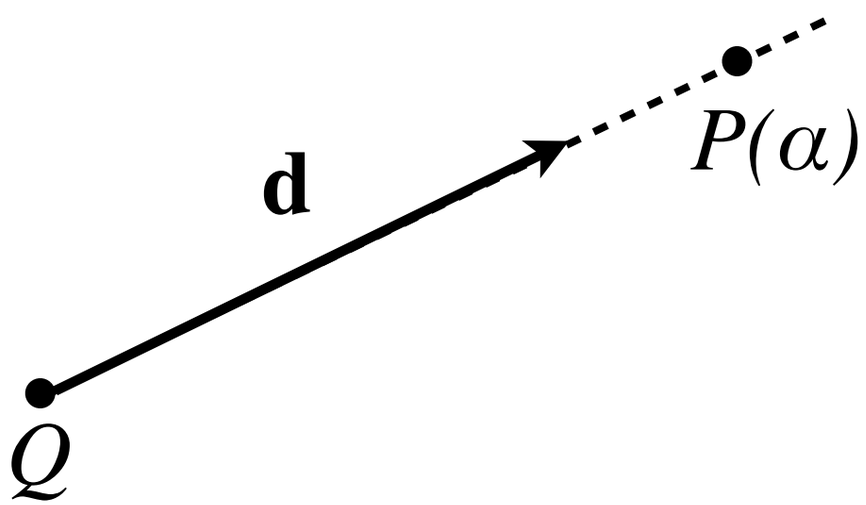
- Point addition

$$P+Q = ?$$

Lines

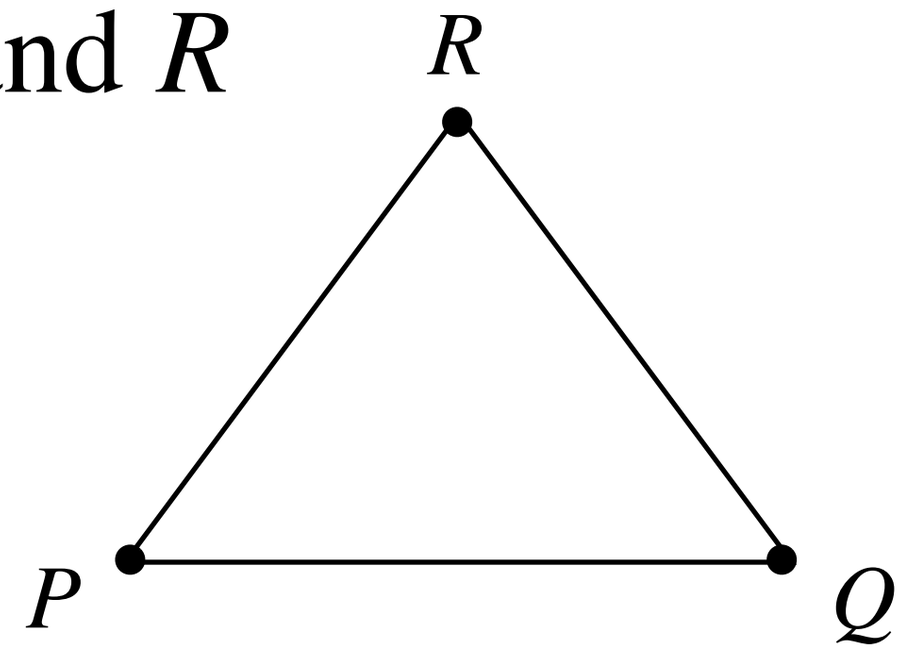
- Parametric form
  - Start with a point Q and a vector $\mathbf{d}$
  - If $\alpha>0$, the line is called the **ray** from Q in the direction $\mathbf{d}$

$$P(\alpha) = Q + \alpha\mathbf{d}$$



Triangle

- Defined by three points P, Q and R



- Points inside triangle:

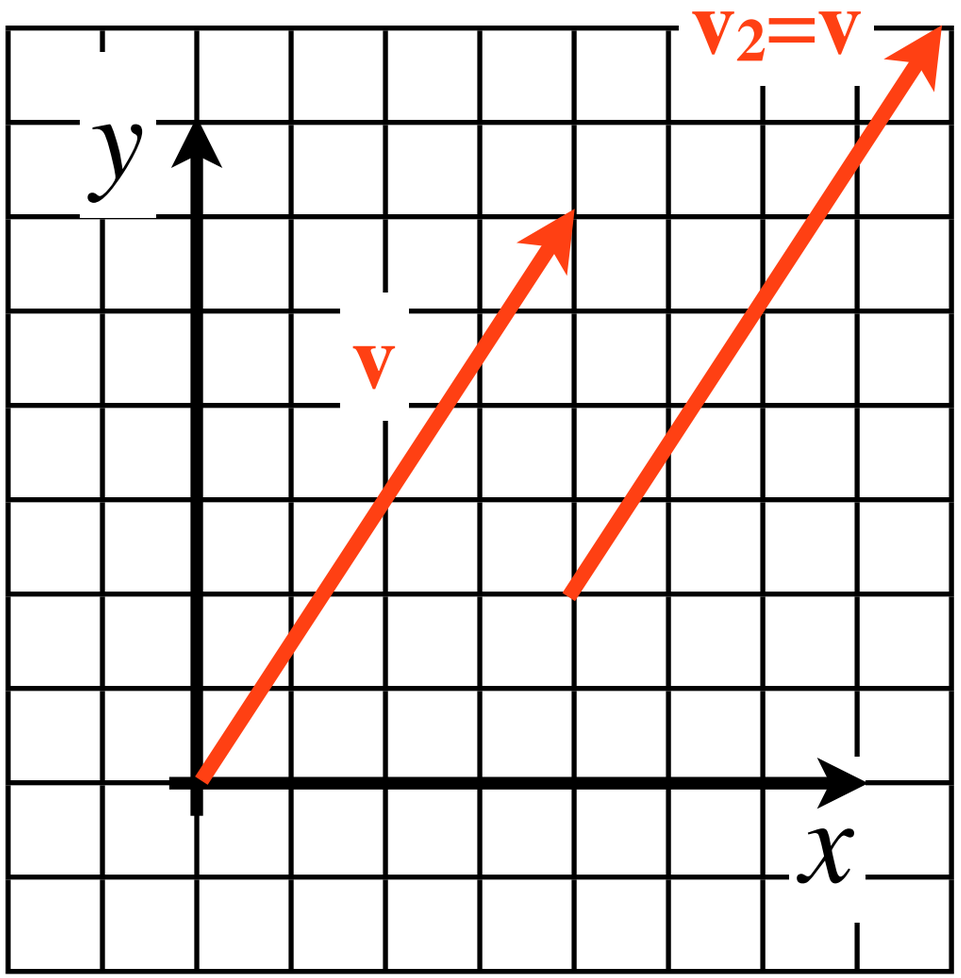
$$wP + uQ + vR$$

$$u + v + w = 1$$

$$u, v, w \geq 0$$

- u,v,w : barycentric coordinates

Cartesian Coordinates



$$\mathbf{v} = 4\mathbf{x} + 6\mathbf{y}$$

$$= 4\begin{bmatrix} 1 \\ 0 \end{bmatrix} + 6\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

$$|\mathbf{v}| = \sqrt{(4^2 + 6^2)}$$

Cartesian Coordinates in 3D

- Express vector in terms of three orthonormal basis vectors

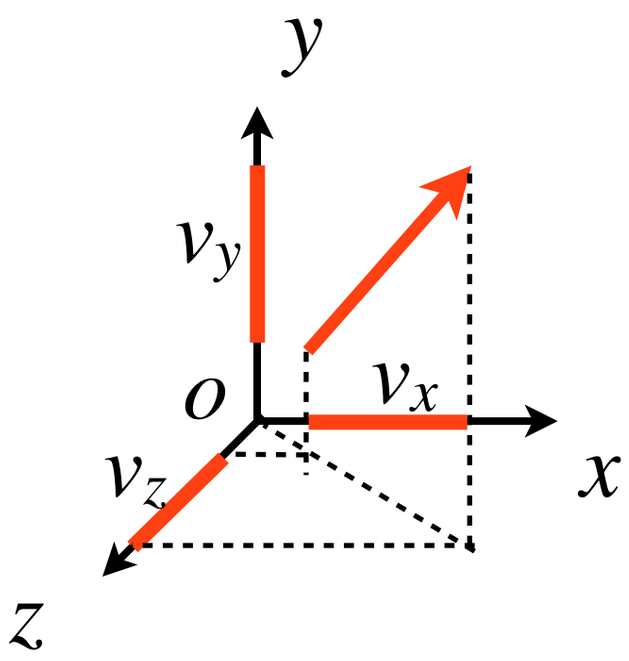
$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = v_x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + v_y \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + v_z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- Notation

$$\mathbf{v} = (v_x, v_y, v_z)^T = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

- Length of vector

$$|\mathbf{v}| = \sqrt{v_x^2 + v_y^2 + v_z^2}$$

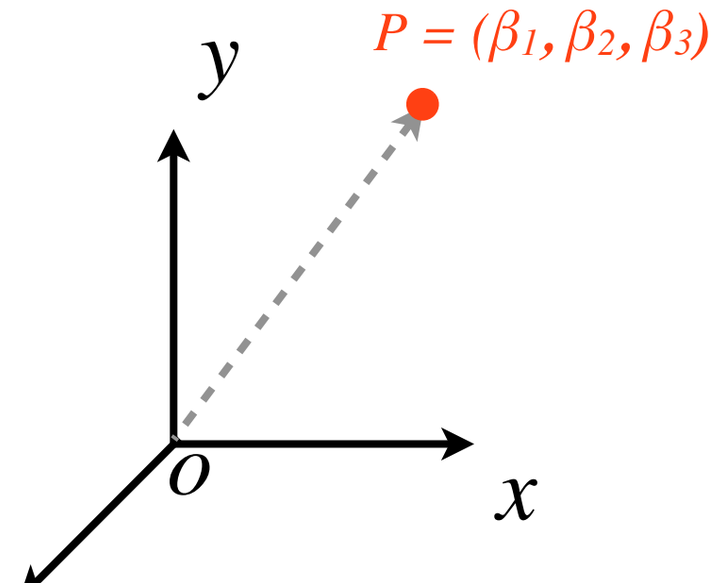


Points in 3D

- Coordinate **frame**
  - Basis vectors + **origin**

$$\begin{aligned} \mathbf{v}_1 &= (1,0,0)^T \\ \mathbf{v}_2 &= (0,1,0)^T \\ \mathbf{v}_3 &= (0,0,1)^T \end{aligned}$$

$$\boxed{P_0 = (0,0,0)^T}$$



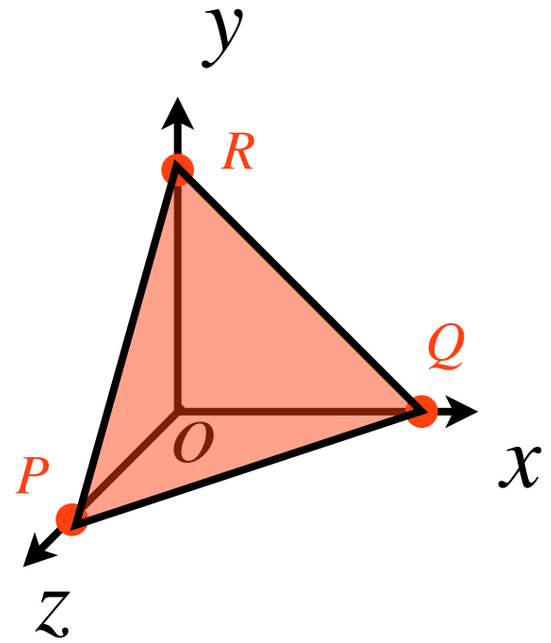
- In this space, a **point** is given by:

$$\begin{aligned} P &= P_0 + \beta_1\mathbf{v}_1 + \beta_2\mathbf{v}_2 + \beta_3\mathbf{v}_3 \\ &= \beta_1(1,0,0)^T + \beta_2(0,1,0)^T + \beta_3(0,0,1)^T \\ &= (\beta_1,\beta_2,\beta_3)^T \end{aligned}$$

Example: Triangle in 3D

- Defined by three points, specified in the Cartesian system

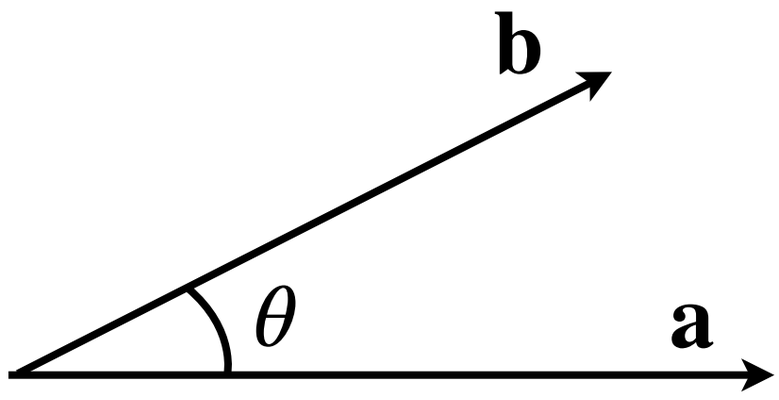
$$P = (0,0,1)^{\mathrm{T}}$$
$$Q = (1,0,0)^{\mathrm{T}}$$
$$R = (0,1,0)^{\mathrm{T}}$$



Dot Product (scalar product)

- Given two 3D vectors $\mathbf{a} = (a_x, a_y, a_z)^{\mathrm{T}}$ and $\mathbf{b} = (b_x, b_y, b_z)^{\mathrm{T}}$, the dot product is given by

$$\mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y + a_z b_z = |\mathbf{a}||\mathbf{b}| \cos\theta$$



- Angle between $\mathbf{a}$ and $\mathbf{b}$: $\theta = \arccos\left(\frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}||\mathbf{b}|}\right)$

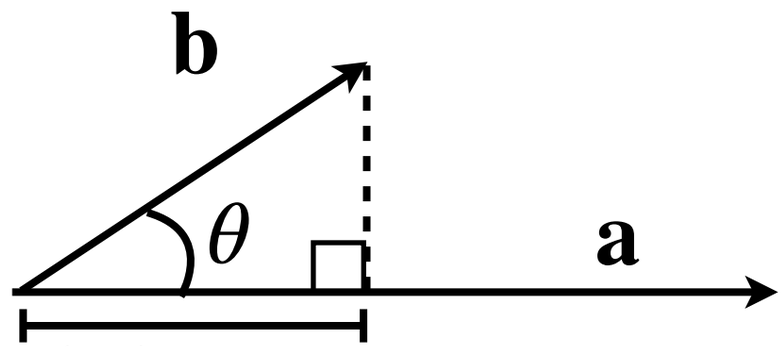
Dot Product (scalar product)

- Use cases

$$\mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y + a_z b_z = |\mathbf{a}||\mathbf{b}| \cos\theta$$

  - The projection of $\mathbf{b}$ on $\mathbf{a}$

$$|\mathbf{b}| \cos\theta = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}|}$$



  - Square magnitude of vector

$$\mathbf{a} \cdot \mathbf{a} = |\mathbf{a}|^2 \Leftrightarrow |\mathbf{a}| = \sqrt{\mathbf{a} \cdot \mathbf{a}}$$

  - $\mathbf{a}$ and $\mathbf{b}$ orthogonal if

$$\mathbf{a} \cdot \mathbf{b} = 0$$

Cross Product

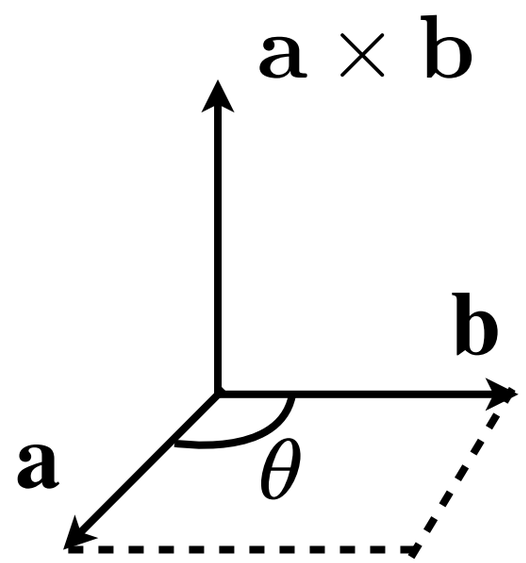
- Given two 3D vectors $\mathbf{a} = (a_\mathrm{x}, a_\mathrm{y}, a_\mathrm{z})^\mathrm{T}$ and $\mathbf{b} = (b_\mathrm{x}, b_\mathrm{y}, b_\mathrm{z})^\mathrm{T}$, the cross product is given by

$$\mathbf{a} \times \mathbf{b} = (a_y b_z - a_z b_y, a_z b_x - a_x b_z, a_x b_y - a_y b_x)$$

- Signed area of the parallelepiped spanned by vectors $\mathbf{a}$ and $\mathbf{b}$:

$$|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}||\mathbf{b}| \sin\theta$$

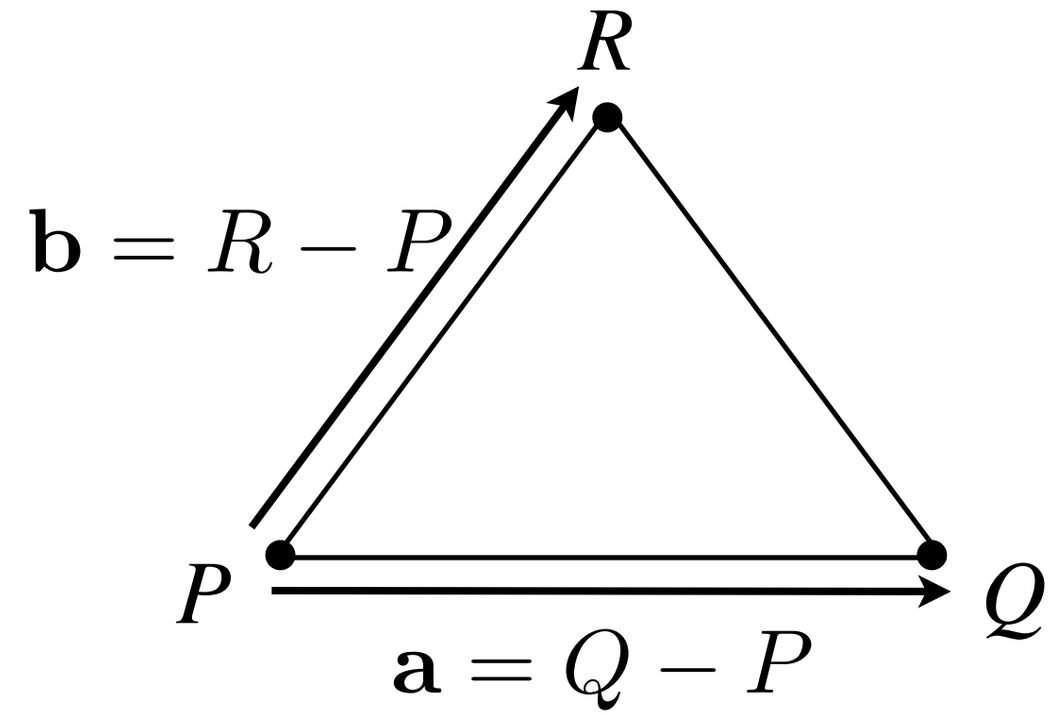
- $\mathbf{a} \times \mathbf{b}$ is orthogonal to both $\mathbf{a}$ and $\mathbf{b}$
- $\mathbf{a} \times \mathbf{a} = \mathbf{0}$
- Note: $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$



Triangle Normal

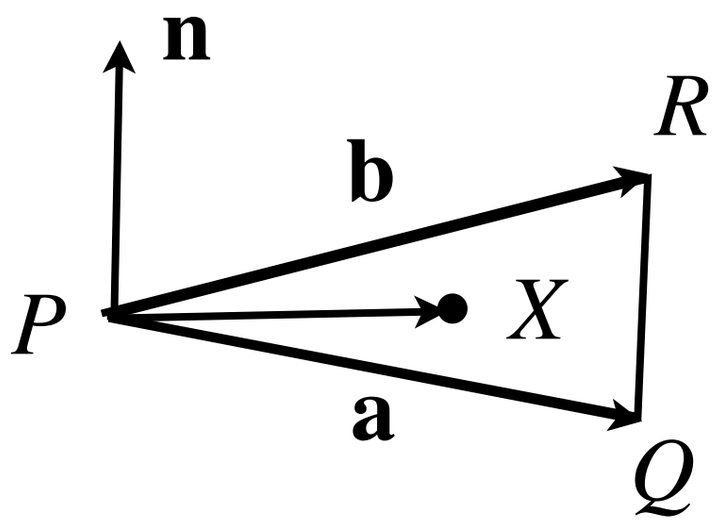
- Introduce two edge vectors $\mathbf{a}, \mathbf{b}$
- Triangle face normal

$$\mathbf{n} = \frac{\mathbf{a} \times \mathbf{b}}{|\mathbf{a} \times \mathbf{b}|}$$



- Plane equation:
  - X belongs to the triangle plane if

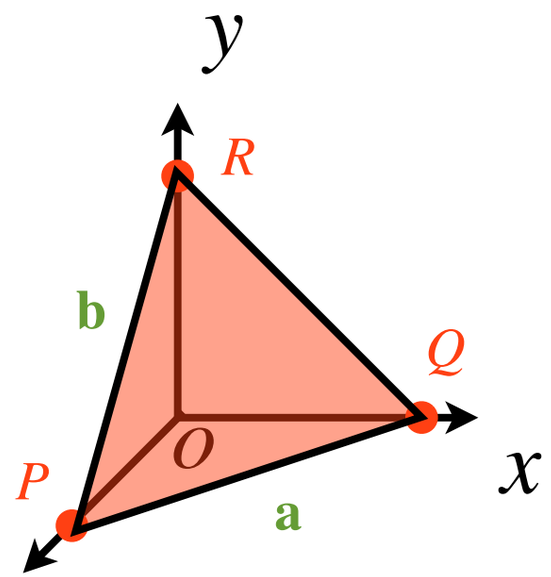
$$\mathbf{n} \cdot (X - P) = 0$$



Example: Triangle in 3D

- Edge vectors

$$\mathbf{a} = Q - P = (1, 0, -1)^T$$
$$\mathbf{b} = R - P = (0, 1, -1)^T$$



$P = (0,0,1)^{\mathrm{T}}$
$Q = (1,0,0)^{\mathrm{T}}$
$R = (0,1,0)^{\mathrm{T}}$

- Face normal

$$\mathbf{n} = \frac{\mathbf{a} \times \mathbf{b}}{|\mathbf{a} \times \mathbf{b}|} = \frac{(1, 1, 1)^T}{|(1, 1, 1)|} = \frac{1}{\sqrt{3}}(1, 1, 1)^T$$

- Plane:

$$\mathbf{n} \cdot (X - P) = 0 \qquad X = (x, y, z)^T$$
$$x + y + z - 1 = 0$$

Matrices

- We will use
  - Matrix-matrix multiplication: $\mathbf{A\,B = C}$
    - Concatenate transforms, change basis
  - Matrix-vector multiplication $\mathbf{A\,x = y}$
    - Transform vectors and points

Matrix-vector multiplication

$$A\mathbf{x} = \mathbf{y}$$

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_{00}x + a_{01}y + a_{02}z \\ a_{10}x + a_{11}y + a_{12}z \\ a_{20}x + a_{21}y + a_{22}z \end{bmatrix}$$

Matrix-matrix multiplication

$$AB = C$$

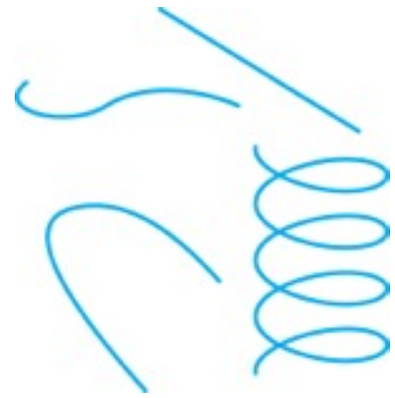
$$\begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{00} & b_{01} & b_{02} \\ b_{10} & b_{11} & b_{12} \\ b_{20} & b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix}$$

$$c_{ij} = \sum_{k=0}^{2} a_{ik} b_{kj}$$

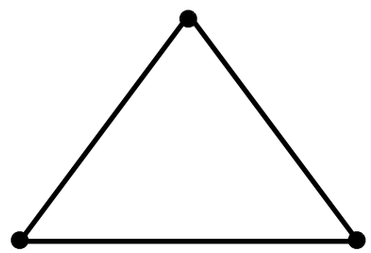
$$\mathbf{AB} \neq \mathbf{BA}$$

3D Primitives

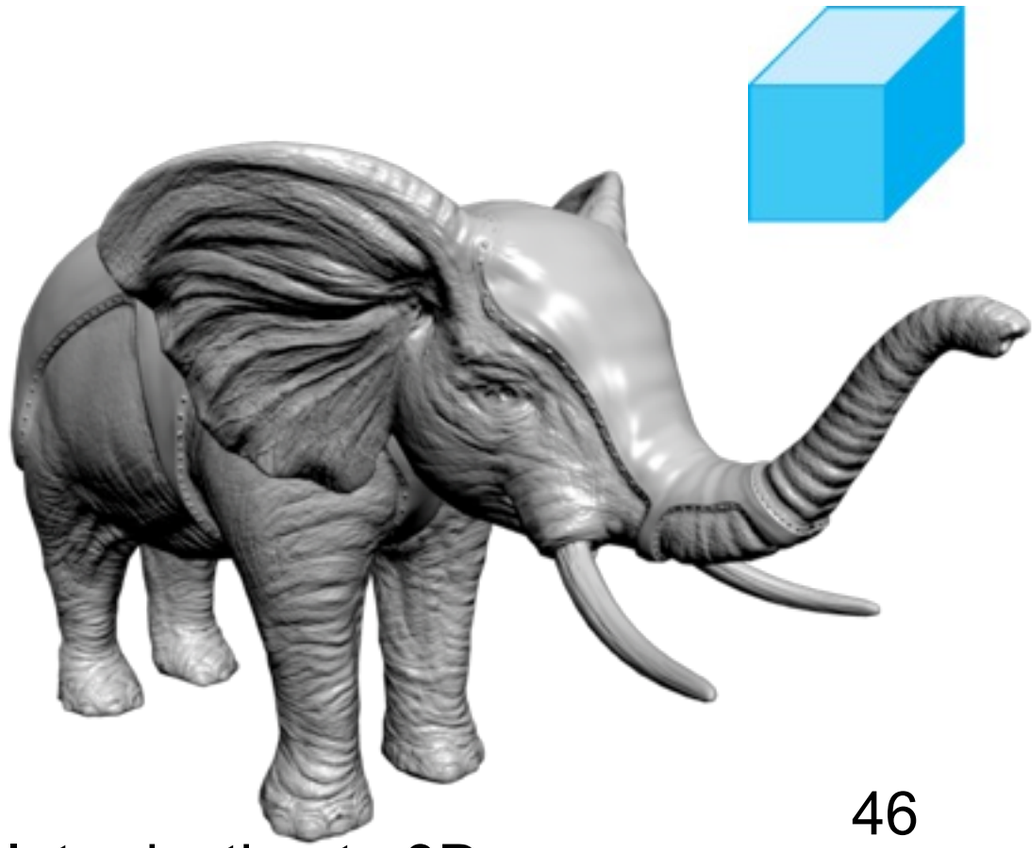
- Curves - paths in space



- Triangle
  - three connected points in 3D

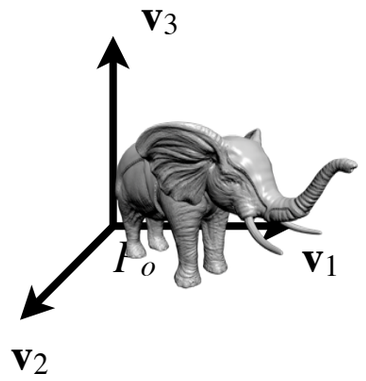


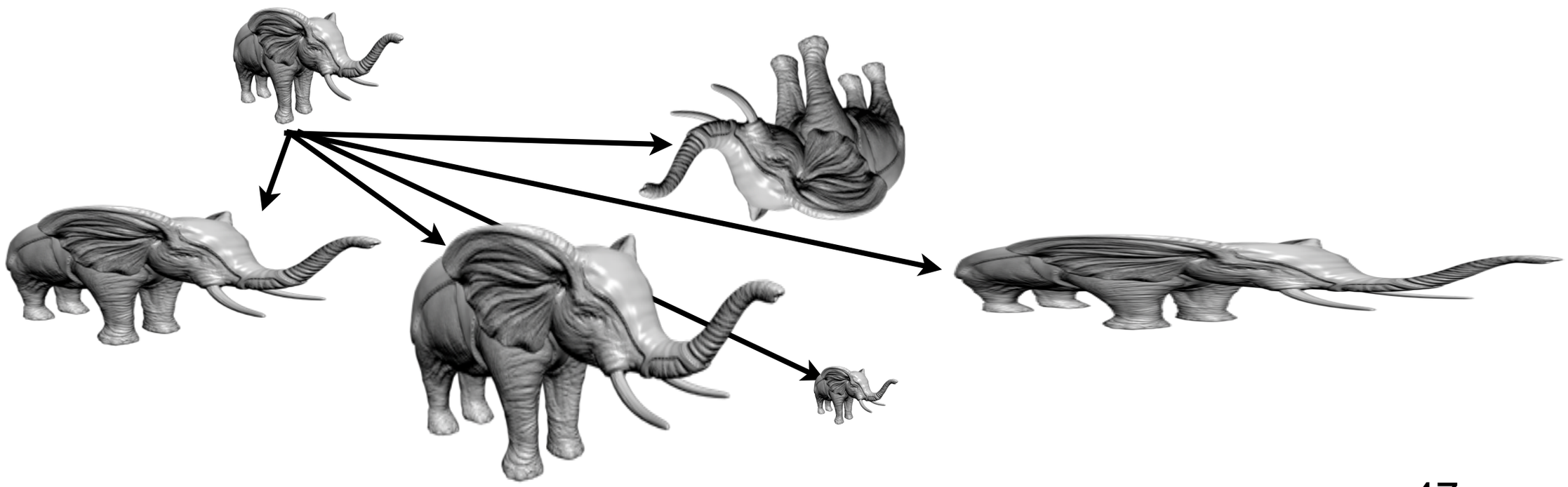
- Cube
  - 8 connected points in 3D (or 12 triangles)
- Elephant
  - 22840 triangles



Transforms

- Define object **once** in convenient local coordinate frame
- Apply transformation $\mathbf{M}$ to position each instance of the object

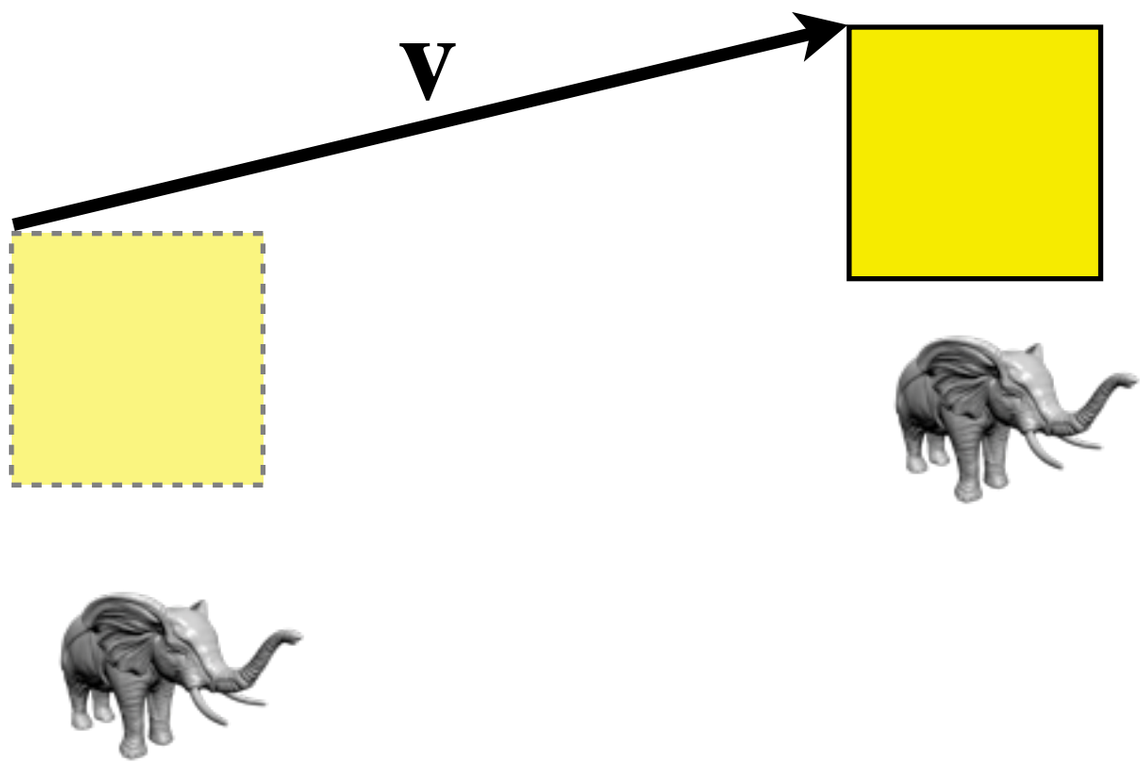




Translation

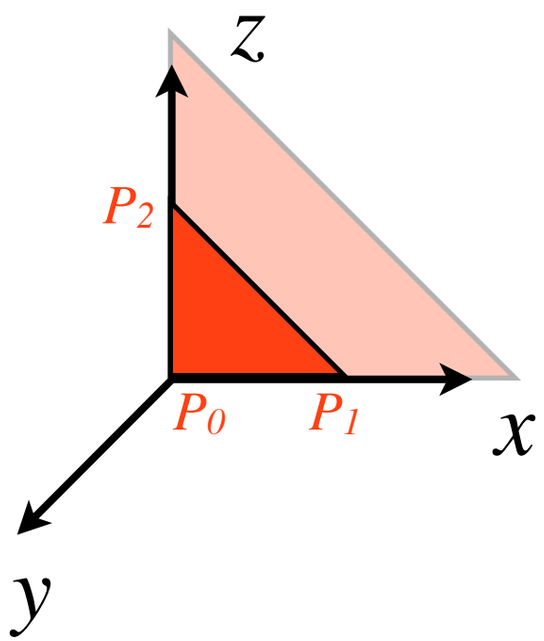
- Move along a vector

$$P' = P + \mathbf{v}$$



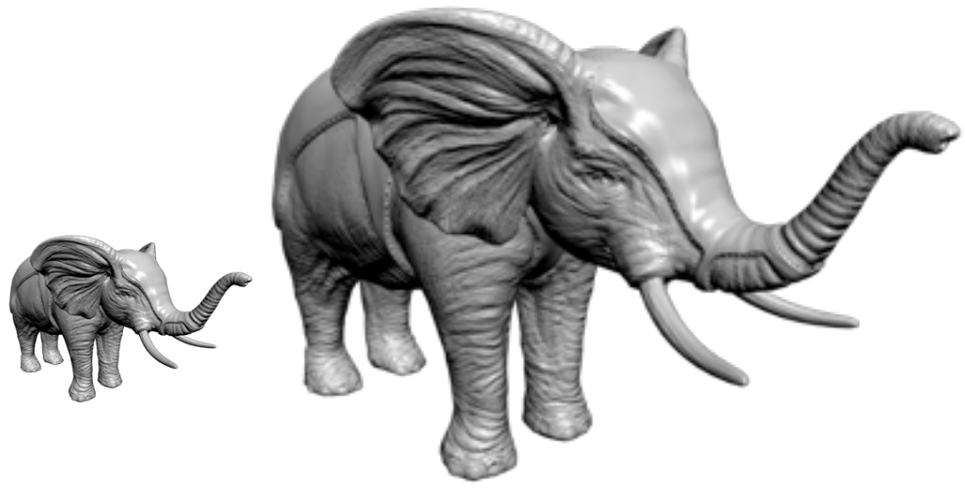
Scaling

- Scale each coordinate with a factor s_x, s_y, s_z
- Uniform scaling
  - $s_x = s_y = s_z$

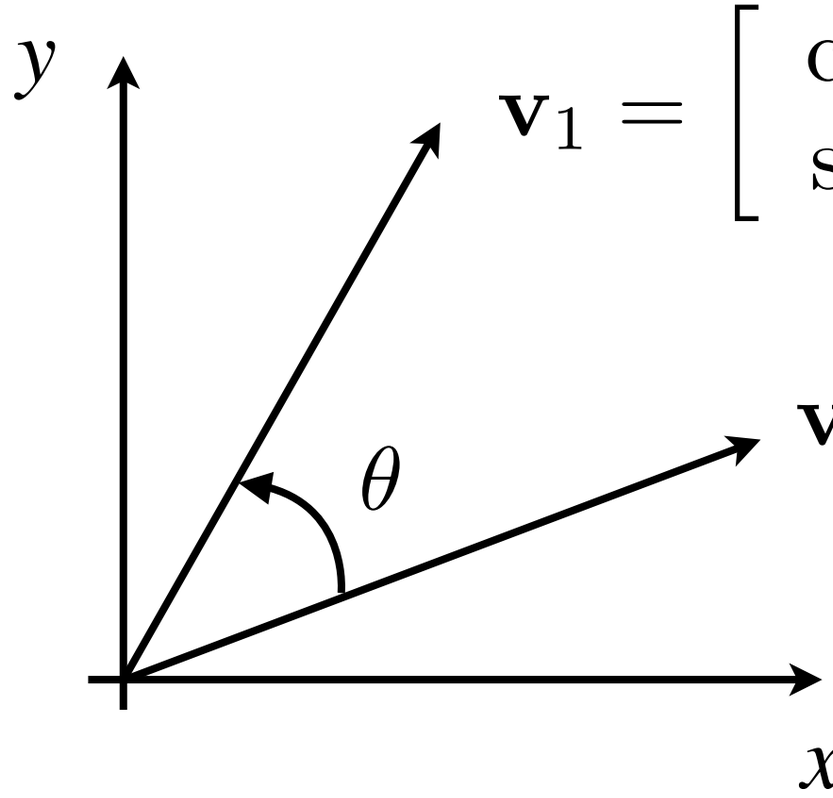


$$P_{scaled} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} P$$

$$\mathbf{S}$$



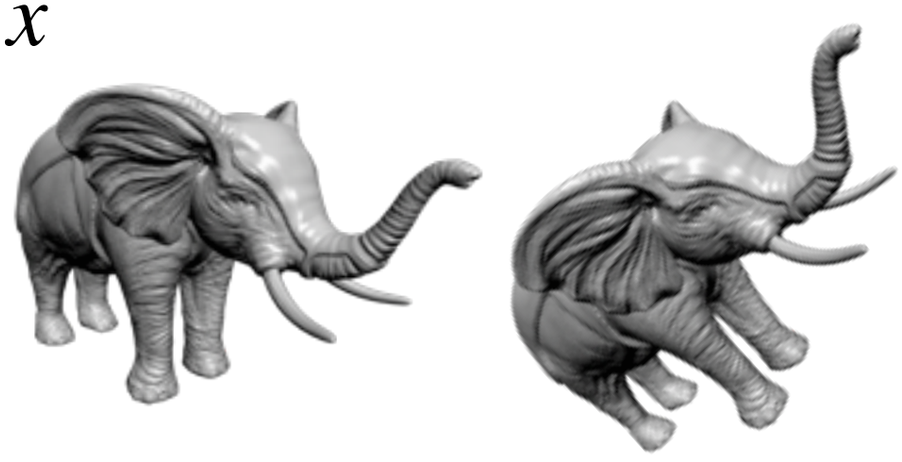
Rotation in the plane



$$\mathbf{v}_1 = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$\mathbf{v}_0 = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$\mathbf{v}_1 = \mathbf{R}\mathbf{v}_0$$



Next lecture

- Transforms
- Coordinate Spaces
- Homogeneous Coordinates

- Come prepared
  - Read chapter 3 in Angel's book