Computer Graphics

Coordinate Systems and Change of Frames

Based on slides by Dianna Xu, Bryn Mawr College

Linear Independence

- A set of vectors $\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_{n-1}$ is ***linearly independent*** if

$$\alpha_0 \vec{u}_0 + \alpha_1 \vec{u}_1 + \ldots + \alpha_{n-1} \vec{u}_{n-1} = 0$$
$$\Leftrightarrow \alpha_0 = \alpha_1 = \ldots = \alpha_{n-1} = 0$$

- **If a set of vectors is linearly independent, we cannot represent any one in terms of the others**

Dimension and Basis

- In a vector space, the maximum number of linearly independent vectors is fixed and is called the *dimension* of the space

- In an *n*-dimensional space, any set of n linearly independent vectors form a *basis* for the space
- Given a basis $\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_{n-1}$, any vector v can be written as (a linear combination of the basis)

$$v = \alpha_0 \vec{u}_0 + \alpha_1 \vec{u}_1 + \ldots + \alpha_{n-1} \vec{u}_{n-1}$$

Basis

- In 2-space, any vector can be represented by $\vec{v} = \alpha_0 \vec{u}_0 + \alpha_1 \vec{u}_1$ where $(\vec{u}_0, \vec{u}_1)$ is a basis
- How does this basis look like?
  - The two vectors are orthogonal
  - The standard basis

- What about 3-space?

- n-space?

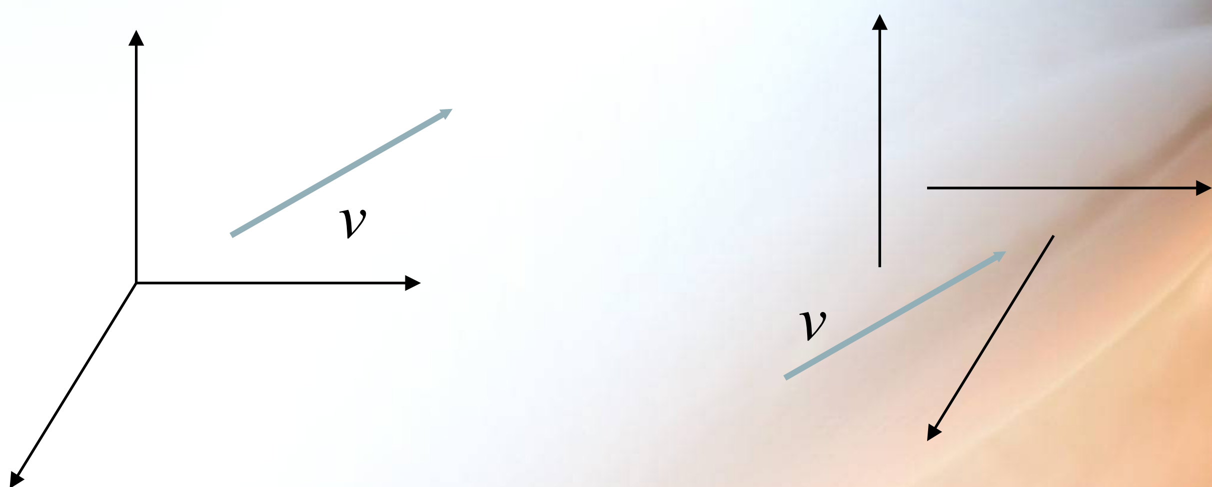
Coordinate Systems

- **Consider a basis** $\vec{u}_0, \vec{u}_1, \ldots, \vec{u}_{n-1}$
- **A vector is written** $v = \alpha_0\vec{u}_0 + \alpha_1\vec{u}_1 + \ldots + \alpha_{n-1}\vec{u}_{n-1}$
- **The list of scalars** $\{\alpha_0, \alpha_1, \ldots. \alpha_{n-1}\}$**is the** ***representation of*** v ***with respect to the given basis***
- **We can write the representation as a row or column array of scalars**

$$\left[\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\right]^T = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \ldots \\ \alpha_{n-1} \end{bmatrix}$$

Coordinate Systems

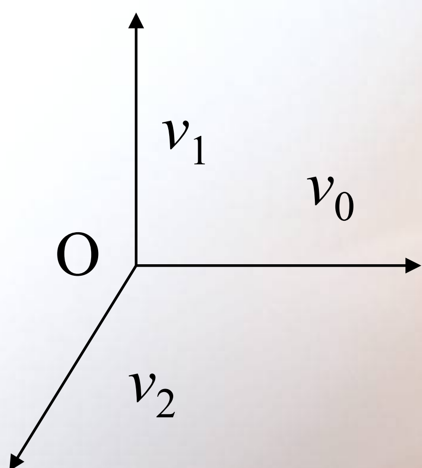
- Which is correct?



- Both are because vectors have no fixed location

Frames

- **Coordinate system is insufficient to represent points**
- **If we work in an affine space we can add a single point, the *origin*, to the basis vectors to form a *frame***



Representation in a Frame

- Frame determined by $(O, \vec{u}_0, \vec{u}_1, \ldots, \vec{u}_{n-1})$
- Within this frame, every vector can be written as

$$v = \alpha_0 \vec{u}_0 + \alpha_1 \vec{u}_1 + \ldots + \alpha_{n-1} \vec{u}_{n-1}$$

- Every point can be written as

$$P = O + \alpha_0 \vec{u}_0 + \alpha_1 \vec{u}_1 + \ldots + \alpha_{n-1} \vec{u}_{n-1}$$

Confusing Points and Vectors

Consider the point and the vector

$$P = O + \alpha_0 \vec{u}_0 + \alpha_1 \vec{u}_1 + \alpha_2 \vec{u}_2$$

$$v = \beta_0 \vec{u}_0 + \beta_1 \vec{u}_1 + \beta_2 \vec{u}_2$$

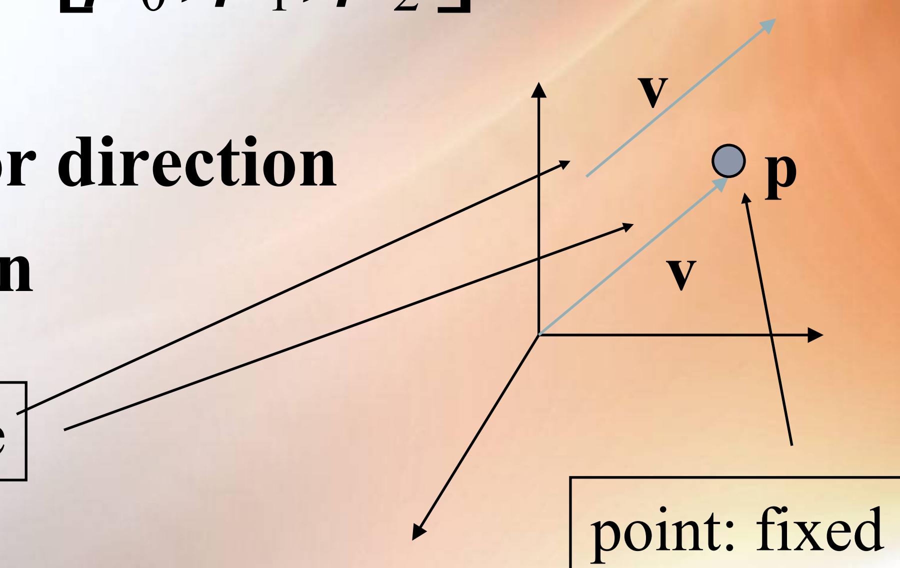
They appear to have the similar representations

$$P = \left[\alpha_0, \alpha_1, \alpha_2\right]^T \quad v = \left[\beta_0, \beta_1, \beta_2\right]^T$$

A point has no length or direction

A vector has no position

vector: can place anywhere



A Single Representation

Define $0 \bullet P = \vec{0}$ **and** $1 \bullet P = P$ **then we have**

$$P = O + \alpha_0 \vec{u}_0 + \alpha_1 \vec{u}_1 + \alpha_2 \vec{u}_2 = \left[\alpha_0, \alpha_1, \alpha_2, 1\right]^T$$

$$v = \beta_0 \vec{u}_0 + \beta_1 \vec{u}_1 + \beta_2 \vec{u}_2 = \left[\beta_0, \beta_1, \beta_2, 0\right]^T$$

Homogenous representation of a point

$$P = \left[\alpha_0, \alpha_1, \alpha_2, 1\right]^T$$

Homogenous representation of a vector

$$v = \left[\beta_0, \beta_1, \beta_2, 0\right]^T$$

Homogeneous Coordinates

In general, the homogeneous coordinates form for a three dimensional point [x y z] is given as

$$P = [wx, wy, wz, w]^T = [x', y', z', w]^T$$

We return to a three dimensional point (for $w \neq 0$) by

$$x \leftarrow x' / w$$
$$y \leftarrow y' / w$$
$$z \leftarrow z' / w$$

If $w=0$, the representation is that of a vector

If $w=1$, the representation of a point is [x y z 1]

Homogeneous Coordinates and Computer Graphics

- All standard viewing transformations (rotation, translation, scaling) can be implemented by matrix multiplications with 4 x 4 matrices
- Hardware pipeline works with 4 dimensional representations
  - Change of coordinate systems
  - Projection
    - For orthographic viewing, we can maintain $w=0$ for vectors and $w=1$ for points
    - For perspective we need a *perspective division*

Change of Coordinate Systems

- Consider two representations of a the same vector with respect to two different bases:

$$\alpha = [\alpha_0, \alpha_1, \alpha_2]$$

$$\beta = [\beta_0, \beta_1, \beta_2]$$

- Where

$$w = \alpha_0 v_0 + \alpha_1 v_1 + \alpha_2 v_2 = [\alpha_0, \alpha_1, \alpha_2][v_0, v_1, v_2]^T$$

$$w = \beta_0 u_0 + \beta_1 u_1 + \beta_2 u_2 = [\beta_0, \beta_1, \beta_2][u_0, u_1, u_2]^T$$

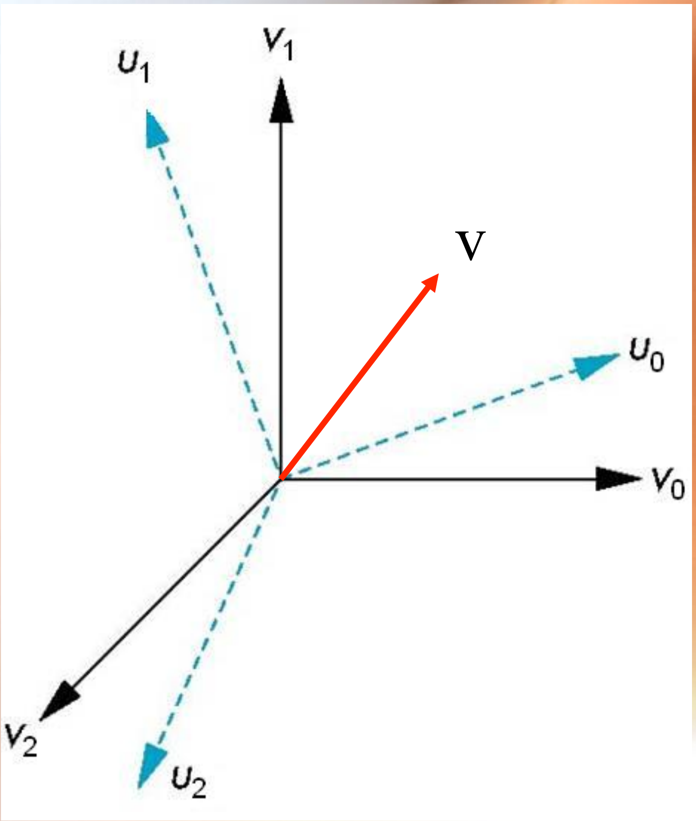
Representing one basis in terms of the other

Each of the basis vectors, u0,u1, u2, are vectors that can be represented in terms of the other basis

$$u_0 = \gamma_{00}v_0 + \gamma_{01}v_1 + \gamma_{02}v_2$$

$$u_1 = \gamma_{10}v_0 + \gamma_{11}v_1 + \gamma_{12}v_2$$

$$u_2 = \gamma_{20}v_0 + \gamma_{21}v_1 + \gamma_{22}v_2$$



Matrix Form

The coefficients define a 3 x 3 matrix

$$M = \begin{bmatrix} \gamma_{00} & \gamma_{01} & \gamma_{02} \\ \gamma_{10} & \gamma_{11} & \gamma_{12} \\ \gamma_{20} & \gamma_{21} & \gamma_{22} \end{bmatrix}$$

and the bases can be related by $u^T = Mv^T$

$$v^T = M^{-1}u^T$$

and the vectors by

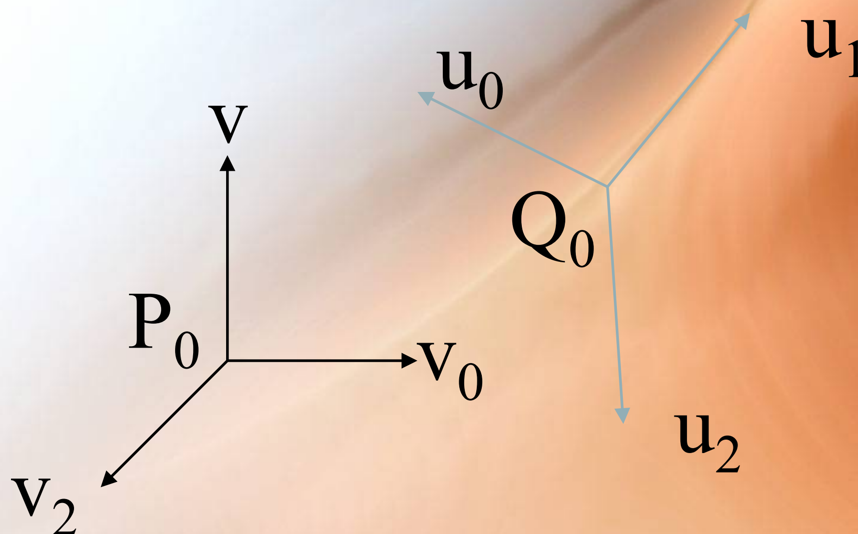
$$w = \beta^T u^T = \beta^T M v^T = \alpha^T v^T \Rightarrow \alpha = M^T \beta$$

$$v^T = M^{-1}u^T \Rightarrow \beta = (M^T)^{-1}\alpha$$

Change of Frames

- We can apply a similar process in homogeneous coordinates to the representations of both points and vectors
- Consider two frames

(P_0, v_0, v_1, v_2)
(Q_0, u_0, u_1, u_2)



- Any point or vector can be represented in each

Change of Homogeneous Coordinates

- **Add point of origin:**

$$u_0 = \gamma_{00}v_0 + \gamma_{01}v_1 + \gamma_{02}v_2$$

$$u_1 = \gamma_{10}v_0 + \gamma_{11}v_1 + \gamma_{12}v_2$$

$$u_2 = \gamma_{20}v_0 + \gamma_{21}v_1 + \gamma_{22}v_2$$

$$Q_0 = \gamma_{30}v_0 + \gamma_{31}v_1 + \gamma_{32}v_2 + P_0$$

$$M = \begin{bmatrix} \gamma_{00} & \gamma_{01} & \gamma_{02} & 0 \\ \gamma_{10} & \gamma_{11} & \gamma_{12} & 0 \\ \gamma_{20} & \gamma_{21} & \gamma_{22} & 0 \\ \gamma_{30} & \gamma_{31} & \gamma_{32} & 1 \end{bmatrix}$$

Working with Representations

- Within the two frames any point or vector has a representation of the same form
  - $\alpha = [\alpha_0, \alpha_1, \alpha_2, \alpha_3]$ in the first frame
  - $\beta = [\beta_0, \beta_1, \beta_2, \beta_3]$ in the second frame
  - where $\alpha_3 = \beta_3 = 1$ for points and $\alpha_3 = \beta_3 = 0$ for vectors
- The matrix M is 4 x 4 and specifies an affine transformation in homogeneous coordinates

$$\alpha = M^T \beta$$

Affine Transformations

- Every linear transformation is equivalent to a change in frames
- Every affine transformation preserves lines
- However, an affine transformation has only 12 *degrees of freedom* because 4 of the elements in the matrix are fixed and are a subset of all possible 4 x 4 linear transformations

The World and Camera Frames

- When we work with representations, we work with n-tuples or arrays of scalars
- Changes in frame are then defined by 4 x 4 matrices
- In OpenGL, the base frame that we start with is the world frame
- Eventually we represent entities in the camera frame by changing the world representation using the model-view matrix
- Initially these frames are the same ($M=I$)
  - Model-view matrix starts out as the identity matrix

Example: Moving the Camera

If objects are on both sides of z=0, we must move camera or world

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$