Representation

- Ed Angel
- Professor of Computer Science, Electrical and Computer Engineering, and Media Arts
- University of New Mexico

Objectives

- Introduce concepts such as dimension and basis
- Introduce coordinate systems for representing vectors spaces and frames for representing affine spaces
- Discuss change of frames and bases
- Introduce homogeneous coordinates

Linear Independence

- A set of vectors $v_1, v_2, \ldots, v_n$ is *linearly independent* if

  $$a_1 v_1 + a_2 v_2 + \ldots + a_n v_n = 0 \text{ iff } a_1 = a_2 = \ldots = 0$$

- If a set of vectors is linearly independent, we cannot represent one in terms of the others
- If a set of vectors is linearly dependent, as least one can be written in terms of the others

Dimension

- 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 $v1, v2, \ldots, vn$, any vector v can be written as

$$v = a1 v1 + a2 v2 + \ldots + an vn$$

where the {ai} are unique

Representation

- Until now we have been able to work with geometric entities without using any frame of reference, such as a coordinate system
- Need a frame of reference to relate points and objects to our physical world.
  - For example, where is a point? Can't answer without a reference system
  - World coordinates
  - Camera coordinates

Coordinate Systems

- Consider a basis $v1, v2,\ldots., vn$
- A vector is written v=a1$v1$+ a2$v2$ +….+anvn
- The list of scalars {a1, a2, …. an}is the *representation* of v with respect to the given basis
- We can write the representation as a row or column array of scalars

a=[a1 a2 ….
an]T=

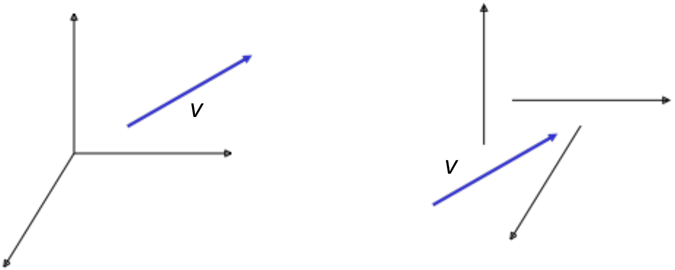
$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ \alpha_n \end{bmatrix}$$

Example

- v=2v1+3v2-4v3
- **a**=[2 3 –4]T
- Note that this representation is with respect to a particular basis
- For example, in OpenGL we start by representing vectors using the object basis but later the system needs a representation in terms of the camera or eye basis

Coordinate Systems

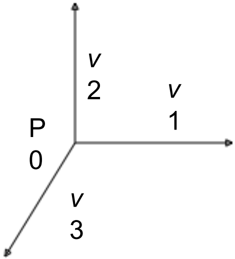
- Which is correct?



- Both are because vectors have no fixed location

Frames

- A 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 (P_0, v_1, v_2, v_3)
- Within this frame, every vector can be written as

  $v = a_1 v_1 + a_2 v_2 + \ldots + a_n v_n$
- Every point can be written as

  $P = P_0 + b_1 v_1 + b_2 v_2 + \ldots + b_n v_n$

Confusing Points and Vectors

Consider the point and the vector

$$P = P_0 + b_1 v_1 + b_2 v_2 + \ldots + b_n v_n$$

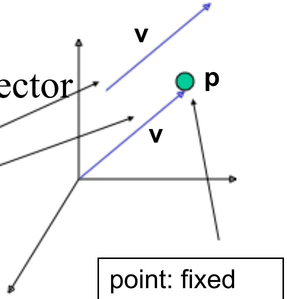
$$v = a_1 v_1 + a_2 v_2 + \ldots + a_n v_n$$

They appear to have the similar representations

p=[b1 b2 b3] **v**=[a1 a2 a3]

which confuses the point with the vector

A vector has no position



Vector can be placed anywhere

point: fixed

The University of New Mexico

A Single Representation

If we define $0 \cdot P = \mathbf{0}$ and $1 \cdot P = P$ then we can write

$v = a_1 v_1 + a_2 v_2 + a_3 v_3 = [a_1\ a_2\ a_3\ 0]\ [v_1\ v_2\ v_3\ P_0]^T$

$P = P_0 + b_1 v_1 + b_2 v_2 + b_3 v_3 = [b_1\ b_2\ b_3\ 1]\ [v_1\ v_2\ v_3\ P_0]^T$

Thus we obtain the four-dimensional *homogeneous coordinate* representation

$\mathbf{v} = [a_1\ a_2\ a_3\ 0]^T$

$\mathbf{p} = [b_1\ b_2\ b_3\ 1]^T$

Homogeneous Coordinates

The homogeneous coordinates form for a three dimensional point [x y z] is given as

$$\mathbf{p} = [x'\ y'\ z'\ w]^T = [wx\ wy\ wz\ w]^T$$

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

Note that homogeneous coordinates replaces points in three dimensions by lines through the origin in four dimensions

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

Homogeneous Coordinates and Computer Graphics

- Homogeneous coordinates are key to all computer graphics systems
  - All standard transformations (rotation, translation, scaling) can be implemented with matrix multiplications using 4 x 4 matrices
  - Hardware pipeline works with 4 dimensional representations
  - 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. The representations are

$\mathbf{a}$=[a1 a2 a3

]$\mathbf{b}$=[b1 b2

b3]

where

v=a1v1+ a2v2 +a3v3 = [a1 a2 a3] [v1

v2 v3] T

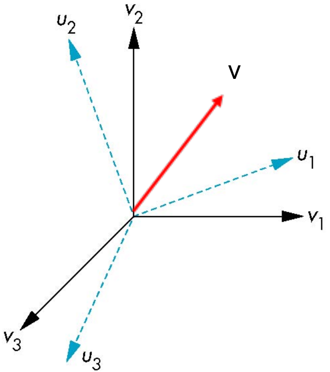
=b1u1+ b2u2 +b3u3 = [b1 b2 b3] [u1 u2

u3] T

Representing second basis in terms of first

Each of the basis vectors, u1,u2, u3, are vectors that can be represented in terms of the first basis

u1 = g11v1+g12v2+g13v3
u2 = g21v1+g22v2+g23v3
u3 = g31v1+g32v2+g33v3



Matrix Form

The coefficients define a 3 x 3 matrix

$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix}$$

and the bases can be related by

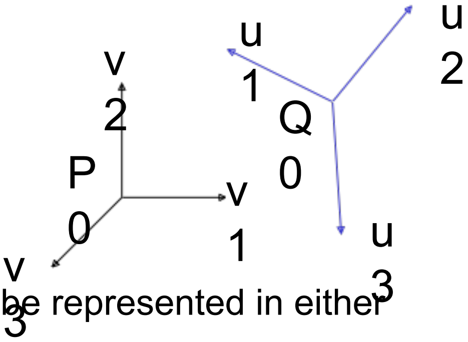
$$\mathbf{a}=\mathbf{M}^T\mathbf{b}$$

see text for numerical examples

Change of Frames

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

Consider two frames:
(P0, v1, v2, v3)
(Q0, u1, u2, u3)



- Any point or vector can be represented in either frame
- We can represent Q0, u1, u2, u3 in terms of P0, v1, v2, v3

Representing One Frame in Terms of the Other

Extending what we did with change of bases

u1 = g11v1+g12v2+g13v3
u2 = g21v1+g22v2+g23v3
u3 = g31v1+g32v2+g33v3
Q0 = g41v1+g42v2+g43v3 +g44P0

defining a 4 x 4 matrix

$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & 0 \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & 0 \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & 0 \\ \gamma_{41} & \gamma_{42} & \gamma_{43} & 1 \end{bmatrix}$$

Working with Representations

Within the two frames any point or vector has a representation of the same form

a=[a1 a2 a3 a4] in the first frame

b=[b1 b2 b3 b4] in the second frame

where a4 = b4 = 1 for points and a4 = b4 = 0 for vectors and

$$\mathbf{a}=\mathbf{M}^T\mathbf{b}$$

The matrix **M** is 4 x 4 and specifies an affine transformation in homogeneous coordinates

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 ($\mathbf{M}=\mathbf{I}$)

Moving the Camera

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

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