CSE167: Introduction to Computer Graphics

Lecture #2

Jürgen P. Schulze
University of California, San Diego
Fall 2009
Course web site:

• http://graphics.ucsd.edu/twiki/bin/view.pl/Classes/CSE167Fall2009
Miscellaneous

- Project description on course web site at “Projects and Homework” link
- Use base code from web site
- Lab sessions with TAs this week in lab 250:
  - Jason: Wednesday 2-5pm; homework introduction 3-4pm
  - Daniel: Thursday 11am-1:45pm
- Use the discussion board on WebCT: http://webctweb.ucsd.edu
- Project 1 due Friday October 9, presentation in lab 250
Today

Linear Algebra

- Vectors
- Matrices
Linear algebra review

Why linear algebra?

• Need to describe 3D scenes
  - Position, orientation, motion of objects
  - Relation of objects to virtual camera
  - Projection of scene onto image plane

• Linear algebra provides mathematical tools
Vectors

- Direction and length in 3D
- Vectors can describe
  - Difference between two 3D points
  - Speed of an object
  - Surface normals (directions perpendicular to surfaces)
Vectors

Multiplication by scalar

\[ a \quad 0.5a \quad -1a = -a \]
Vectors

Addition
Vectors

Addition
Vectors

Linear combination

\[ sa + tb, \quad s, t \in \mathbb{R} \]

\[ \sum_{i=1}^{n} s_i a_i, \quad s_i \in \mathbb{R} \]
Vectors

Linear combination

$$sa + tb, \quad s, t \in \mathbb{R}$$

$$\sum_{i=1}^{n} s_i a_i, \quad s_i \in \mathbb{R}$$

Linearly dependent vectors

• A set of vectors $a_i, i = 1 \ldots n$ is linearly dependent if there exist scalars $s_i$ such that

$$a_j = \sum_{i=1, i\neq j}^{n} s_i a_i$$

• Otherwise, they are linearly independent
Coordinate systems

- Describe any vector with respect to basis vectors $x, y, z$

\[ \mathbf{a} = a_x \mathbf{x} + a_y \mathbf{y} + a_z \mathbf{z} \]

- The basis vectors form a coordinate system
- $a_x, a_y, a_z$ are coordinates of vector $\mathbf{a}$
- Same vector can be represented by different coordinate systems
Coordinate systems

• Any three vectors that are linearly independent could be used as a basis

• Vectors do not need to
  - have the same lengths
  - be perpendicular to each other
Coordinate systems

- Any three vectors that are linearly independent could be used as a basis
  - Can have different lengths
  - Do not have to be perpendicular to each other
- Why linearly independent?
- Why exactly three vectors?
- Other coordinate systems?
Coordinate systems

Euclidean coordinate systems

• Basis vectors
  - Have unit length
  - Are perpendicular to each other

• Also called *orthonormal basis*
Coordinate Systems

Handedness

Right handed

Left handed
Vector arithmetic using coordinates

\[ \mathbf{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \]

\[ \mathbf{a} + \mathbf{b} = \begin{bmatrix} a_x + b_x \\ a_y + b_y \\ a_z + b_z \end{bmatrix} \quad \mathbf{a} - \mathbf{b} = \begin{bmatrix} a_x - b_x \\ a_y - b_y \\ a_z - b_z \end{bmatrix} \]

\[ -\mathbf{a} = \begin{bmatrix} -a_x \\ -a_y \\ -a_z \end{bmatrix} \quad s\mathbf{a} = \begin{bmatrix} sa_x \\ sa_y \\ sa_z \end{bmatrix} \]

where \( s \) is a scalar
Vector Magnitude

- The magnitude (length) of a vector is:

\[ |\mathbf{v}|^2 = v_x^2 + v_y^2 + v_z^2 \]

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

- A vector with length of 1.0 is called unit vector

- We can also normalize a vector to make it a unit vector

\[ \frac{\mathbf{v}}{|\mathbf{v}|} \]

- Unit vectors are often used as surface normals
Questions?
Dot product

- The dot product is a scalar value that tells us something about the relationship between two vectors
  - Angles between vectors
  - Lengths of vectors
- Independent of coordinate system
Dot Product

- If $\mathbf{a} \cdot \mathbf{b} > 0$ then $\theta < 90^\circ$
  - Vectors point in the same general direction
- If $\mathbf{a} \cdot \mathbf{b} < 0$ then $\theta > 90^\circ$
  - Vectors point in opposite direction
- If $\mathbf{a} \cdot \mathbf{b} = 0$ then $\theta = 90^\circ$
  - Vectors are perpendicular
  - (or one or both vectors are degenerate (0,0,0))
Dot Product using coordinates

- Result is independent of coordinate system!

\[ \mathbf{a} \cdot \mathbf{b} = \sum a_i b_i \]

\[ \mathbf{a} \cdot \mathbf{b} = a_x b_x + a_y b_y + a_z b_z \]

\[ \mathbf{a} \cdot \mathbf{b} = |a||b| \cos \theta \]
Angle between vectors

- How do you find the angle $\theta$ between vectors $\mathbf{a}$ and $\mathbf{b}$?
\[ a \cdot b = |a||b| \cos \theta \]

\[ \cos \theta = \left( \frac{a \cdot b}{|a||b|} \right) \]

\[ \theta = \cos^{-1} \left( \frac{a \cdot b}{|a||b|} \right) \]
Dot products with unit vector

- If $|\mathbf{u}|=1.0$ then $\mathbf{a} \cdot \mathbf{u}$ is the length of the orthogonal projection of $\mathbf{a}$ onto $\mathbf{u}$

\[ \mathbf{a} \cdot \mathbf{u} = |\mathbf{a}| \cos \theta \]
Dot products with unit vectors

\[ \mathbf{a} \cdot \mathbf{b} = 1 \]

|a| = |b| = 1.0

\[ \mathbf{a} \cdot \mathbf{b} = \cos(\theta) \]
Dot products with unit vectors

\[ 0 < a \cdot b < 1 \]

\[ a \cdot b = 1 \]

\[ |a| = |b| = 1.0 \]

\[ a \cdot b = \cos(\theta) \]
Dot products with unit vectors

- $0 < a \cdot b < 1$
- $a \cdot b = 0$
- $a \cdot b = 1$

$|a| = |b| = 1.0$

$a \cdot b = \cos(\theta)$
Dot products with unit vectors

- $0 < a \cdot b < 1$
- $a \cdot b = 0$
- $-1 < a \cdot b < 0$
- $a \cdot b = 1$

$a \cdot b = \cos(\theta)$

$|a| = |b| = 1.0$
Dot products with unit vectors

- $0 < a \cdot b < 1$
- $a \cdot b = 0$
- $-1 < a \cdot b < 0$
- $a \cdot b = -1$
- $a \cdot b = 1$

$|a| = |b| = 1.0$

$a \cdot b = \cos(\theta)$
Questions?
Cross product

\( \mathbf{a} \times \mathbf{b} \) is a vector perpendicular to both \( \mathbf{a} \) and \( \mathbf{b} \), in the direction defined by the right hand rule.

If vectors \( \mathbf{a}, \mathbf{b} \) are unit length and perpendicular, then \( \mathbf{a}, \mathbf{b}, \mathbf{a} \times \mathbf{b} \) is a right handed coordinate system.
Cross product

\[ \mathbf{a} \times \mathbf{b} \] is a vector *perpendicular* to both \( \mathbf{a} \) and \( \mathbf{b} \), in the direction defined by the right hand rule.

Vectors \( \mathbf{a}, \mathbf{b} \) lie in the plane of the projection screen. Does \( \mathbf{a} \times \mathbf{b} \) point towards you or away from you? What about \( \mathbf{b} \times \mathbf{a} \)?
Cross product

\( \mathbf{a} \times \mathbf{b} \) is a vector \textit{perpendicular} to both \( \mathbf{a} \) and \( \mathbf{b} \), in the direction defined by the right hand rule

\[
|\mathbf{a} \times \mathbf{b}| = |\mathbf{a}| |\mathbf{b}| \sin \theta
\]

\[
|\mathbf{a} \times \mathbf{b}| = \text{area of parallelogram } \mathbf{a} \mathbf{b}
\]

\[
|\mathbf{a} \times \mathbf{b}| = 0 \text{ if } \mathbf{a} \text{ and } \mathbf{b} \text{ are parallel}
\]

(or one or both degenerate)
Cross product

• Using coordinates

\[ \mathbf{a} \times \mathbf{b} = \begin{bmatrix}
    a_y b_z - a_z b_y \\
    a_z b_x - a_x b_z \\
    a_x b_y - a_y b_x
\end{bmatrix} \]
Example: Align two vectors

- We are heading in direction $\mathbf{h}$. We want to rotate so that we will align with a different direction $\mathbf{d}$. Find a unit axis $\mathbf{a}$ and an angle $\theta$ to rotate around.
Example: Align two vectors

\[ a = \frac{h \times d}{|h \times d|} \]
Example: Align two vectors

\[ a = \frac{h \times d}{|h \times d|} \]

\[ \theta = \sin^{-1}\left( \frac{|h \times d|}{|h||d|} \right) \]

\[ \theta = \cos^{-1}\left( \frac{h \cdot d}{|h||d|} \right) \]

\[ \theta = \tan^{-1}\left( \frac{|h \times d|}{h \cdot d} \right) \]

\[ \text{theta} = \text{atan2}\left(|h \times d|, h \cdot d\right) \]
Questions?
class Vector3 {
    public:
        float x, y, z;

        Vector3() { x=0.0; y=0.0; z=0.0; }
        Vector3(float x0, float y0, float z0) { x=x0; y=y0; z=z0; }
        void Set(float x0, float y0, float z0) { x=x0; y=y0; z=z0; }
        void Add(Vector3 &a) { x+=a.x; y+=a.y; z+=a.z; }
        void Add(Vector3 &a, Vector3 &b) { x=a.x+b.x; y=a.y+b.y; z=a.z+b.z; }
        void Subtract(Vector3 &a) { x-=a.x; y-=a.y; z-=a.z; }
        void Subtract(Vector3 &a, Vector3 &b) { x=a.x-b.x; y=a.y-b.y; z=a.z-b.z; }
        void Negate() { x=-x; y=-y; z=-z; }
        void Negate(Vector3 &a) { x=-a.x; y=-a.y; z=-a.z; }
        void Scale(float s) { x*=s; y*=s; z*=s; }
        void Scale(float s, Vector3 &a) { x=s*a.x; y=s*a.y; z=s*a.z; }
        float Dot(Vector3 &a) { return x*a.x+y*a.y+z*a.z; }
        void Cross(Vector3 &a, Vector3 &b) { x=a.y*b.z-a.z*b.y; y=a.z*b.x-a.x*b.z; z=a.x*b.y-a.y*b.x; }
        float Magnitude() { return sqrtf(x*x+y*y+z*z); }
    void Normalize() { Scale(1.0f/Magnitude()); }
};
class Vector3 {
public:
    float x, y, z;
    Vector3() {x=0.0; y=0.0; z=0.0;}
    Vector3(float x0, float y0, float z0) {x=x0; y=y0; z=z0;}
    void Set(float x0, float y0, float z0) {x=x0; y=y0; z=z0;}
    void Add(Vector3 &a) {x+=a.x; y+=a.y; z+=a.z;}
    void Add(Vector3 &a, Vector3 &b) {x=a.x+b.x; y=a.y+b.y; z=a.z+b.z;}
    void Subtract(Vector3 &a) {x=a.x-b.x; y=a.y-b.y; z=a.z-b.z;}
    void Negate() {x=-x; y=-y; z=-z;}
    void Negate(Vector3 &a) {x=-a.x; y=-a.y; z=-a.z;}
    void Scale(float s) {x*=s; y*=s; z*=s;}
    void Scale(float s, Vector3 &a) {x=s*a.x; y=s*a.y; z=s*a.z;}
    float Dot(Vector3 &a) {return x*a.x+y*a.y+z*a.z;}
    void Cross(Vector3 &a, Vector3 &b) {
        x=a.y*b.z-a.z*b.y; y=a.z*b.x-a.x*b.z; z=a.x*b.y-a.y*b.x;
    }
    float Magnitude() {return sqrtf(x*x+y*y+z*z);}
    void Normalize() {Scale(1.0f/Magnitude());}
};

Optional:
Use C++ operator overloading where appropriate
Today

- Vectors
- Matrices
Matrices

Motivation

• Need matrices to implement geometric transformations on vectors, later points, in 3D

  - Rotation, translation, scaling, shearing, etc.
Matrices

Abstract point of view

• Mathematical objects with set of operations
  - Addition, subtraction, multiplication, multiplicative inverse, etc.

• Similar to integers, real numbers, etc.

• But properties of operations are different
  - E.g., multiplication is not commutative
Matrices

Practical point of view

- Rectangular array of numbers

\[ M = \begin{bmatrix}
  m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\
  m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\
  \vdots & \vdots & \ddots & \vdots \\
  m_{m,1} & m_{2,2} & \cdots & m_{m,n}
\end{bmatrix} \in \mathbb{R}^{m \times n} \]
Matrices

Practical point of view

• Rectangular array of numbers

\[
M = \begin{bmatrix}
m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\
m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
m_{m,1} & m_{2,2} & \cdots & m_{m,n}
\end{bmatrix} \in \mathbb{R}^{m \times n}
\]

• Square matrix if \( m = n \)

• In graphics often \( m = n = 3, m = n = 4 \)
Matrix addition

\[ A + B = \begin{bmatrix}
  a_{1,1} + b_{1,1} & a_{1,2} + b_{1,2} & \ldots & a_{1,n} + b_{1,n} \\
  a_{2,1} + b_{2,1} & a_{2,2} + b_{2,2} & \ldots & a_{2,n} + b_{2,n} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{m,1} + b_{m,1} & a_{2,2} + b_{2,2} & \ldots & a_{m,n} + b_{m,n}
\end{bmatrix} \]

\[ A, B \in \mathbb{R}^{m \times n} \]
Multiplication with scalar

\[ sM = Ms = \begin{bmatrix}
sm_{1,1} & sm_{1,2} & \ldots & sm_{1,n} \\
sm_{2,1} & sm_{2,2} & \ldots & sm_{2,n} \\
\vdots & \vdots & \ddots & \vdots \\
sm_{m,1} & sm_{2,2} & \ldots & sm_{m,n}
\end{bmatrix} \]
Matrix multiplication

\[ AB = C, \quad A \in \mathbb{R}^{p,q}, B \in \mathbb{R}^{q,r}, C \in \mathbb{R}^{p,r} \]
Matrix multiplication

\[ AB = C, \quad A \in \mathbb{R}^{p,q}, \ B \in \mathbb{R}^{q,r}, \ C \in \mathbb{R}^{p,r} \]

\[(AB)_{i,j} = C_{i,j} = \sum_{k=1}^{q} a_{i,k} b_{k,j}, \quad i \in 1..p, \ j \in 1..r\]
Matrix multiplication

Special case: matrix-vector multiplication

$$Ax = y, \quad A \in \mathbb{R}^{p,q}, \ x \in \mathbb{R}^q, \ y \in \mathbb{R}^p$$

$$(Ax)_i = y_i = \sum_{k=1}^{q} a_{i,k}x_k$$
Linearity

• Distributive law holds

\[ A(sB + tC) = sAB + tAC \]

• But multiplication is not commutative,

\[ AB \neq BA \]

in general
Identity matrix

\[ I = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix} \in \mathbb{R}^{n \times n} \]

\[ MI = IM = M, \quad \text{for any } M \in \mathbb{R}^{n \times n} \]
Matrix inverse

Definition

If a square matrix $M$ is non-singular, there exists a unique inverse $M^{-1}$ such that

$$MM^{-1} = M^{-1}M = I$$

• Note

$$(MPQ)^{-1} = Q^{-1}P^{-1}M^{-1}$$

• Computation

- Gaussian elimination, Cramer’s rule
- Review in your linear algebra book, or quick summary
  
  [Link to source](http://www.maths.surrey.ac.uk/explore/emmaspages/option1.html)
OpenGL matrices

• Vectors are column vectors

• “Column major” ordering

• Matrix elements stored in array of floats

  float M[16];

• Corresponding matrix elements

  \[
  \begin{bmatrix}
  \end{bmatrix}
  \]
Next class

• More on homogeneous coordinates and transformations

• Lab hours, lab 250:
  - Wednesday 2-5pm with Jason; homework introduction at 3pm
  - Thursday 11am-1:45pm with Daniel