Computer Graphics Lecture 03: Mathematics for Computer Graphics

DR. ELVIS S. LIU

SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

SPRING 2018

Mathematics for Computer Graphics (a revision)

- Mathematics is very important in a traditional Computer Graphics course
  - In the old times, almost half of this course is about math
  - The heaviest math course after Discrete Math and Numerical Analysis for Computer Science students
- Mainly in Linear Algebra
- Calculus (Well may be.....)

Different Objects

- Points
  - Represent locations

- Vectors
  - Represent movement, force, displacement from A to B

- Normals
  - Represent orientation, unit length

- Coordinates
  - Numerical Representation of the above objects in a given coordinate system

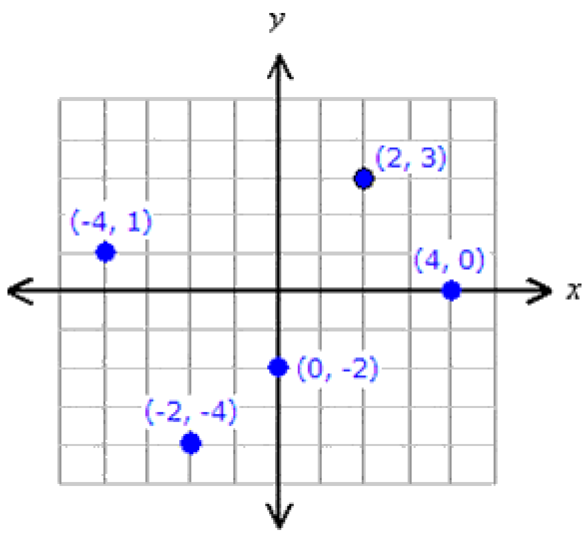
$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Points and Vectors

LECTURE 03: MATHEMATICS FOR COMPUTER GRAPHICS

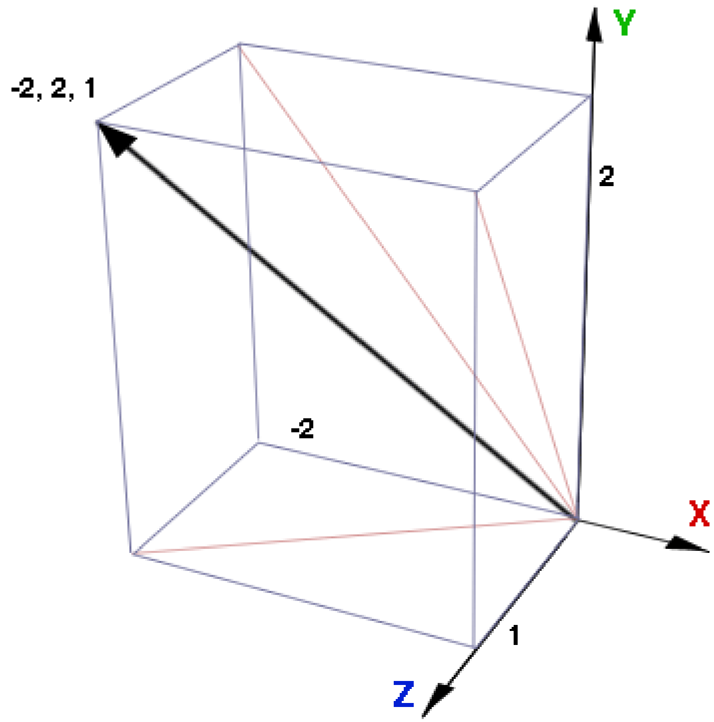
Points

- Much of Computer Graphics involves discussion of points in 2D or 3D
- Usually we write such points as Cartesian Coordinates, e.g.
  - e.g. $\boldsymbol{p} = (\boldsymbol{x},\boldsymbol{y})$ or $\boldsymbol{q} = (\boldsymbol{x},\boldsymbol{y},\boldsymbol{z})$
- Points are scalar quantity



Vectors

- In computer graphics we usually represent coordinates by vector
- Is represented by bold italic serif
  - e.g. ***a AB***
- non-bold italic serif accented by a right arrow
  - e.g. $\vec{a}$ $\overrightarrow{AB}$



Difference between Points and Vectors

- Vector represents direction
- The 0 vector has a fundamental meaning
  - No movement
  - No force
- It is meaningful to add vectors, not points
  - e.g. London location + New York location = ?
- Moving car
  - Points describe location of car elements
  - Vector describe velocity, distance between pairs of points
  - If a moving car is translated to a different road
    - The points (location) change
    - The Vectors (speed, distance between points) do not

Vector Notations

- In the Computer Vision community, column vectors are often used

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

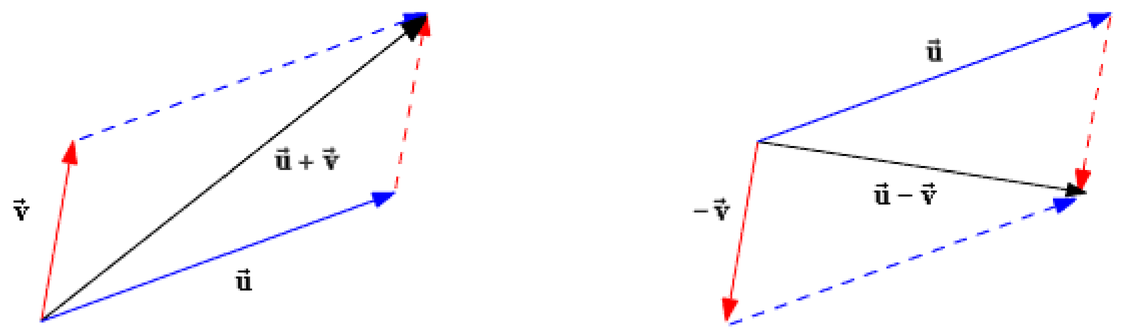
- On the other hand, in the Computer Graphics community, row vectors are often used

$$\begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix}^{\mathrm{T}}$$

- The transpose (indicated by T) of a row vector is a column vector, and vice versa

Basic Vector Algebra

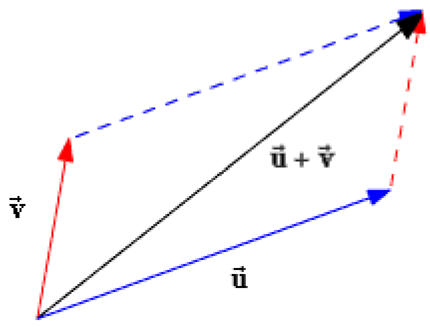
- Just as we can perform basic operation such as addition, multiplication etc. on scalar vales, so we can generalise such operations to vectors



Vector Addition

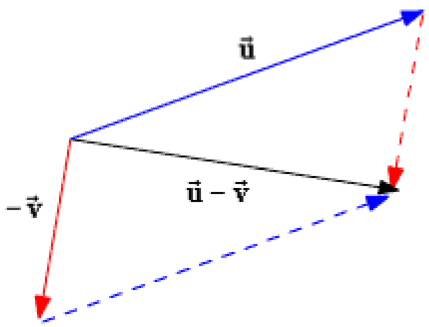
- When we add two vectors, we simply sum their elements at corresponding positions

- For a pair of 2D vectors $\vec{u} = \begin{bmatrix} a \\ b \end{bmatrix}$ and $\vec{v} = \begin{bmatrix} c \\ d \end{bmatrix}$, we have
  - $\vec{u} + \vec{v} = \begin{bmatrix} a + c \\ b + d \end{bmatrix}$



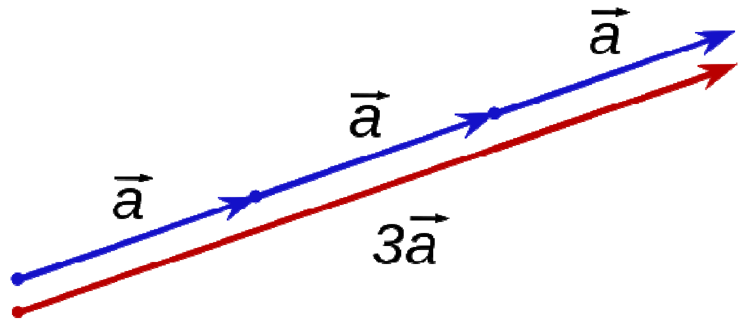
Vector Subtraction

- Vector subtraction is identical to the addition operation with a sign change, since when we negate a vector we simply flip the sign on its elements
- $-\vec{v} = \begin{bmatrix} -c \\ -d \end{bmatrix}$
- $\vec{u} - \vec{v} = \vec{u} + (-\vec{v}) = \begin{bmatrix} a - c \\ b - d \end{bmatrix}$



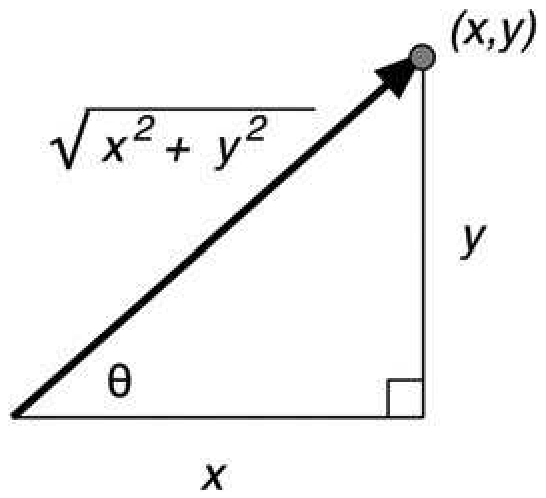
Vector Scaling

- If we wish to increase or reduce a vector quantity by a scale factor λ then we multiply each element in the vector by λ
- $\lambda\vec{u} = \begin{bmatrix} \lambda a \\ \lambda b \end{bmatrix}$



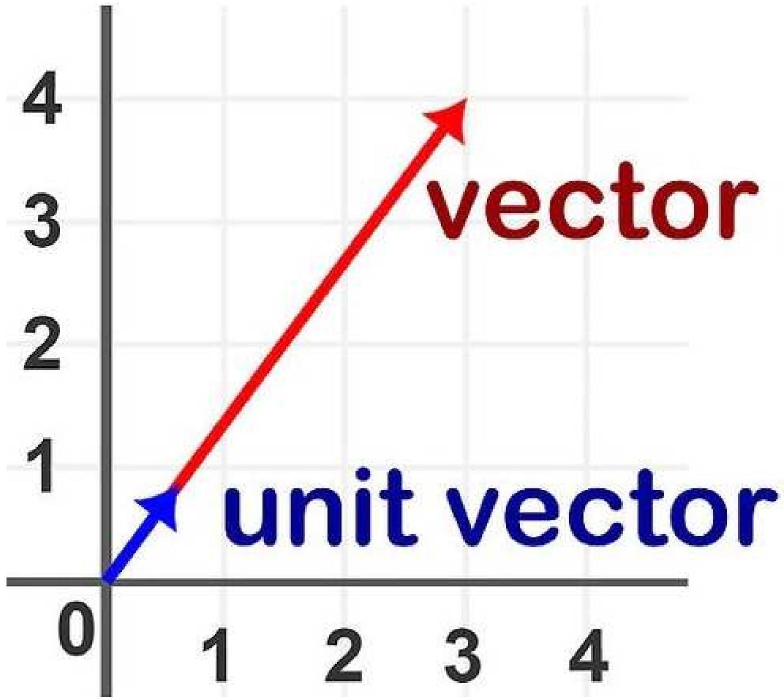
Vector Magnitude

- We write the length of magnitude of a vector $\vec{v}$ as $|v|$
- We use Pythagoras' theorem to compute the magnitude
  - $|v| = \sqrt{x^2 + y^2}$



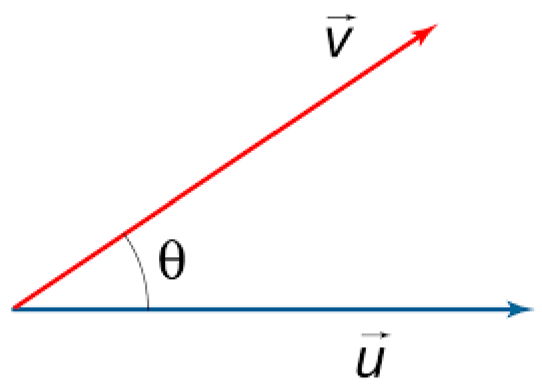
Vector Normalisation

- We can normalise a vector a by scaling it by the reciprocal of its magnitude
  - $\hat{v} = \frac{\vec{v}}{|v|}$
- This produces a normalised vector pointing in the same direction as the original (un-normalised) vector, but with unit length (i.e. length of 1)
- We use the superscript 'hat' notation to indicate that a vector is normalized
  - $\hat{v}$



Dot Product

- we can use the dot product to compute the angle θ between two vectors (if we normalise them first)
- $\vec{u} \cdot \vec{v} = |\vec{u}||\vec{v}| \cos\theta$
- This relationship can be used to define the concept of an angle between vectors in n−dimensional spaces
- It is also fundamental to most lighting calculations in Graphics,
  - e.g. to determine the angle of a surface (normal) to a light source



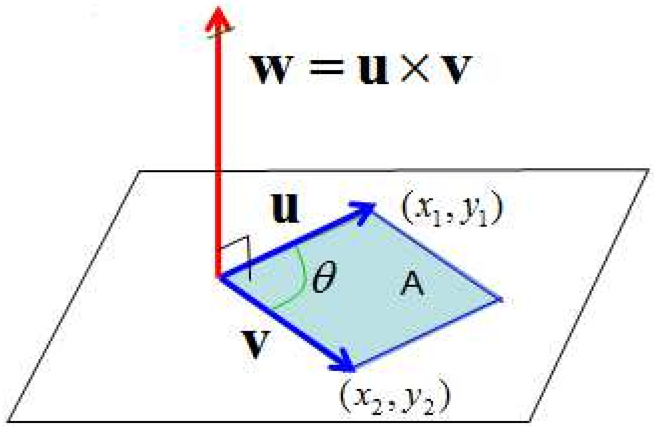
Cross Product

- Taking the cross product (or "vector product") of two vectors returns us a vector orthogonal to those two vectors

- Give two vectors
  - $\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}$ and $\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$

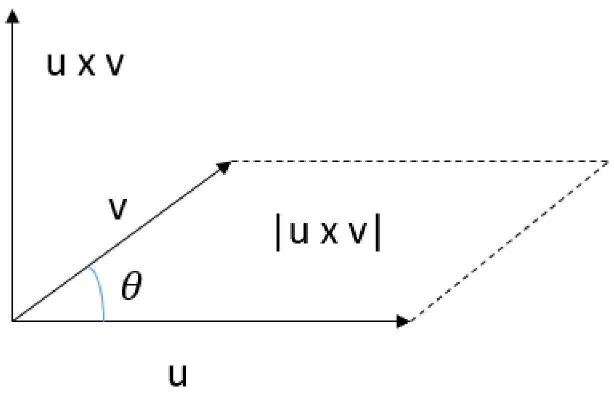
- The cross product of $\vec{u} \times \vec{v}$ is defined as

- $\vec{u} \times \vec{v} = \begin{bmatrix} u_x & - & v_x \\ u_y & - & v_y \\ u_z & - & v_z \end{bmatrix}$



Cross Product (cont.)

- In this course we only consider the definition of the cross product in 3D
- An important Computer Graphics application of the cross product is to determine a vector that is orthogonal to its two inputs
- This vector is said to be normal to those inputs, and is written $\vec{w}$ in the figure
- The cross product of $\vec{u} \times \vec{v}$ can also be defined as
  - $\vec{u} \times \vec{v} = |u||v| \sin\theta \, \vec{w}$



Reference Frames

LECTURE 03: MATHEMATICS FOR COMPUTER GRAPHICS

Cartesian Reference Frame

- When we write down a vector in Cartesian coordinates, for example $\vec{v} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$, we interpret that notation as "the point p is 3 units from the origin travelling in the positive direction of the x axis, and 2 units from the origin travelling in the positive direction of the y axis"

- We can write this more generally and succinctly as:
  - $\vec{v} = x\hat{\imath} + y\hat{\jmath}$
  - where $\hat{\imath} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\hat{\jmath} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

- We call the $\hat{\imath}$ and $\hat{\jmath}$ the basis vectors of the Cartesian space, and together they form the basis set of that space

Cartesian Reference Frames (cont.)

- Sometimes we use the term reference frame to refer to the coordinate space, and we say that the basis set ($\hat{\imath}$,$\hat{\jmath}$) therefore defines the reference frame
- Commonly when working with 2D Cartesian coordinates we work in the reference frame defined by $\hat{\imath} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\hat{\jmath} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- However other choices of basis vector are equally valid, so long as the basis vectors are neither parallel nor anti-parallel (do not point in the same direction)
- We refer our standard reference frame as the root reference frame

Radial-Polar Reference Frame

- We have so far recapped on Cartesian coordinate systems. These describe vectors in terms of distance along each of the principal axes (e.g. x,y) of the space
- This Cartesian form is by far the most common way to represent vector quantities, like the location of points in space
- Sometimes it is preferable to define vectors in terms of length, and their orientation. This is called radial-polar form (often simply abbreviated to 'polar form')

Radial-Polar Reference Frame (cont.)

- In the case of 2D point locations, we describe the point in terms of:
  - (a) its distance from the origin (r), and
  - (b) the angle (θ) between a vertical line (pointing in the positive direction of the y axis), and
  - the line subtended from the point to the origin
- To convert from Cartesian form $\begin{bmatrix} x \\ y \end{bmatrix}$ to polar form (r, θ) we consider a right-angled triangle of side x and y
- We can use Pythagoras' theorem to determine the length of hypotenuse r, and some basic trigonometry to reveal that θ = tan(y/x) and so
- $r = \sqrt{x^2 + y^2}$
- $\theta = \tan^{-1} \frac{y}{x}$

Matrix Algebra

LECTURE 03: MATHEMATICS FOR COMPUTER GRAPHICS

Matrix

- A matrix is a rectangular array of numbers
- Both vectors and scalars are degenerate forms of matrices
- By convention we say that an (n × m) matrix has n rows and m columns
- For example a 2 x 2 matrices A and B are
  - $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$
  - $B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$

Matrix Addition

- Matrices can be added, if they are of the same size
- This is achieved by summing the elements in one matrix with corresponding elements in the other matrix
- $A + B = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{bmatrix}$
- This is identical to vector addition

Matrix Scaling

- Matrices can also be scaled by multiplying each element in the matrix by a scale factor
- Again, this is identical to vector scaling
- $sA = \begin{bmatrix} sa_{11} & sa_{12} \\ sa_{21} & sa_{22} \end{bmatrix}$

Matrix Multiplication

- Matrix multiplication is a cornerstone of many useful geometric transformations in Computer Graphics
- $AB = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$
- $= \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{21} + a_{22}b_{22} \end{bmatrix}$
- In general each element c_{ij} of the matrix C = AB, where A is of size (n × P) and B is of size (P × m) has the form
- $c_{ij} = \sum_{k=1}^{P} a_{ik} b_{kj}$

Matrix Multiplication (cont.)

- Not all matrices are compatible for multiplication. In the previous example, A must have as many columns as B has rows
- matrix multiplication is non-commutative, which means that
  - $BA \neq AB$, in general
- Finally, matrix multiplication is associative i.e.:
  - $ABC = (AB)C = A(BC)$
- If the matrices being multiplied are of different (but compatible) sizes, then the complexity of evaluating such an expression varies according to the order of multiplication

Identity Matrix

- The identity matrix *I* is a special matrix that behaves like the number 1 when multiplying scalars
  - i.e. it has no numerical effect
  - $IA = A$
- The identity matrix has zeroes everywhere except the leading diagonal which is set to 1
- For example, the (2 x 2) identity matrix is
  - $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Matrix Inverse

- The identity matrix leads us to a definition of the inverse of a matrix, which we write
  - A^{-1}
- The inverse of a matrix, when pre- or post-multiplied by its original matrix, gives the identity:
  - $AA^{-1} = A^{-1}A = I$

Determinant

- To calculate the matrix inverse, we must learn about how to calculate the determinant
- Determinants are mathematical objects that are very useful in the analysis and solution of systems of linear equations
- Determinant of matrix A is commonly denoted det (A) or |A|
- A 2 x 2 determinant is defined to be
- Det $\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$

Matrix Inverse (cont.)

- The matrix inverse of a 2 x 2 matrix A can be calculated as

$$\mathbf{A}^{-1} = \frac{1}{|\mathbf{A}|}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

- What about a 3 x 3 matrix?
  - It is a bit complicated……

Matrix Transposition

- Matrix transposition, just like vector transposition, is simply a matter of swapping the rows and columns of a matrix
- As such, every matrix has a transpose
- The transpose of A is written A^T
  - $A^T = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}$
- For some matrices (the orthonormal matrices), the transpose actually gives us the inverse of the matrix
- We decide if a matrix is orthonormal by inspecting the vectors that make up the matrix's columns
  - e.g. $\begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}$ and $\begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix}$, these are sometimes called column vectors of the matrix