Geometric Intuition

Randy Gaul
Talk Outline

- Dot and cross products
- Matrices
- Geometry
- 3D Animation
- Barycentric Coordinates
Prerequisites

- Matrix multiplication
- How to apply dot/cross product
- Vector normalization
- Basic understanding of sin/cos
- Basic idea of what a plane is
Vectors

• Formal definition
  – Input or output from a function of vector algebra

• Informal definition
  – Scalar components representing a direction and magnitude (length)
  – Vectors have no “location”
Points and Vectors

• A point \( P \) and \( Q \) is related to the vector \( V \) by:
  \[ P - Q = V \]

• This implies that points can be translated by adding vectors
Points and Vectors (2)

• Lets define an implied point $O$
  – $O$ represents the origin
• Any point $P$ can be expressed with a vector $V$:
  – $P = O + V$

• Vectors point to points, when relative to the origin!
Euclidean Basis

- Standard Euclidean Basis (called $E^3$)
  - Geometrically the x, y and z axes

$$E^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
Euclidean Basis (2)

- Given 3 scalars $i$, $j$ and $k$, any point in $E^3$ can be represented as:

$$E^3 = i \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, j \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, k \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$
Linear Combination

• Suppose we have a vector $V$
  – Consists of 3 scalar values
  – Below $V$ written in “shorthand” notation

$$V = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$
Linear Combination (2)

- $V$ can be written as a linear combination of $E^3$
  - This is the “longhand” notation of a vector

$$V = x \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, y \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, z \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$
Basis Matrices

- $E^3$ is a basis matrix
  - Many matrices can represent a basis
- Any vector can be represented in any basis

Note: Different $i$, $j$ and $k$ values are used on left and right

Rotation Matrices

• A rotation matrix can rotate vectors
  – Constructed from 3 orthogonal unit vectors
  – Can be called an “orthonormal basis”

• $E^3$ is a rotation matrix!
Rotation Matrices (2)

• Rotation matrices consist of an x, y and z axis
  – Each axis is a vector

\[
\begin{bmatrix}
a & d & g \\
b & e & h \\
c & f & i
\end{bmatrix}
\]
Rotation Matrices (3)

- Multiplying a rotation and a vector rotates the vector.
- This is a linear combination.

\[
\begin{bmatrix}
a & d & g \\
b & e & h \\
c & f & i \\
\end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \cdot x + \begin{bmatrix} d \\ e \\ f \end{bmatrix} \cdot y + \begin{bmatrix} g \\ h \\ i \end{bmatrix} \cdot z
\]
Dot Product

• Comes from law of cosines
• Full derivation here
• For two vectors $u$ and $v$:

$$u_x v_x + u_y v_y + u_w v_w = |u| |v| \times \cos \gamma$$
Shortest Angle Between 2 Vectors

• Assume \( u \) and \( v \) are of unit length

\[
\text{dot}( \hat{u}, \hat{v} ) = \cos \gamma
\]

• Result in range of 0, 1

• No trig functions required
How Far in a Given Direction?

- Given point P and vector V
  - How far along the V direction is P?
How Far in a Given Direction? (2)

• Normalize $V$
• Dot $V$ and $P$ (treat $P$ as vector)

$$dot(\hat{V}, P) = \cos \gamma |P|$$

• Explanation:
  – Cosine $\gamma$ scaled by the length of $P$ is equivalent to the distance $P$ travels in the direction $V$
Planes

• Here’s the 3D plane equation

\[ ax + by + cz - d = 0 \]
Planes

• Here’s the 3D plane equation

\[ ax + by + cz - d = 0 \]

• WAIT A SECOND
Planes

• Here’s the 3D plane equation

\[ ax + by + cz - d = 0 \]

• WAIT A SECOND

• THAT’S THE DOT PRODUCT
Planes (2)

\[ ax + by + cz - d = 0 \]

- \( a, b \) and \( c \) form a vector, called the normal.
- \( d \) represents distance of plane from origin.
Planes (3)

\[ ax + by + cz - d = 0 \]

• d is interesting

– d is scaled by length of normal \( \begin{pmatrix} a \\ b \\ c \end{pmatrix} \)
Planes and the Dot Product

• Dot $P$ with the plane normal:

$$P : \{x \ y \ z\} \star \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \delta$$

• $\delta$ is distance along normal that $P$ travels
• $\delta$ is scaled by length of normal
Planes and the Dot Product (2)

• 2D visualization, assume normal ($\hat{n}$) is unit

\[
dot(P, \hat{n}) = \delta = | / |\]
Signed Distance $P$ to Plane

- Take $\delta$ subtract $d$

- Explanation:
  - $\delta$ is distance $P$ travels along normal
  - Subtract $d$ (distance of plane from origin)
  - This gives distance of $P$ to the plane
Project P onto Plane

• 2D visualization, assume normal ($\hat{n}$) is unit

$$P - (\hat{n} \ast (\text{dot}(P, \hat{n}) - d))$$
Rotation Matrices and Dot Product

• Given matrices A and B
• $A \times B$ is to dot the rows of A with columns of B
• Lets assume A and B are rotation matrices

\[
AB = \begin{pmatrix}
    a & b & c \\
p & q & r \\
u & v & w \\
\end{pmatrix}
\begin{pmatrix}
    \alpha & \beta & \gamma \\
\lambda & \mu & \nu \\
\rho & \sigma & \tau \\
\end{pmatrix}
= \begin{pmatrix}
    a\alpha + b\lambda + c\rho \\
p\alpha + q\lambda + r\rho \\
u\alpha + v\lambda + w\rho \\
\end{pmatrix}
\begin{pmatrix}
    a\beta + b\mu + c\sigma \\
p\beta + q\mu + r\sigma \\
u\beta + v\mu + w\sigma \\
\end{pmatrix}
\begin{pmatrix}
    a\gamma + b\nu + c\tau \\
p\gamma + q\nu + r\tau \\
u\gamma + v\nu + w\tau \\
\end{pmatrix}
\]

http://en.wikipedia.org/wiki/Matrix_multiplication
Rotation Matrices and Dot Product (2)

• That’s a lot of dot products!

• Try: $A \times B \times \text{vector } V$
  – $V$ is rotated by $A$, and then rotated by $B$

• What does $A \times B$ mean?
  – Each element is $\cos \gamma$ between a column of $A$ and row of $B$
Rotation Matrices and Dot Product (3)

• $A = \begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$, $B = \begin{bmatrix} j & m & p \\ k & n & q \\ l & o & r \end{bmatrix}$

• Let's start multiplying these guys
• We start by performing: 
\[
\begin{bmatrix}
a \\
b \\
c
\end{bmatrix}
\begin{bmatrix}
j & m & p
\end{bmatrix}
\]

• \[
\begin{bmatrix}
a \\
b \\
c
\end{bmatrix}
\] is the x axis of A

• \[
\begin{bmatrix}
j & m & p
\end{bmatrix}
\] is x components of the x, y and z axis of B
Rotation Matrices and Dot Product (5)

• What does do? \[
\begin{bmatrix}
    a \\
    b \\
    c
\end{bmatrix}
\begin{bmatrix}
    j & m & p
\end{bmatrix}
\]

  – Computes the contribution of B’s x components along the x axis of A!

• Every single element of A * B can be thought of in this way
Convex Hulls
Point in Convex Hull

• Test if point is inside of a hull
  – Compute plane equation of hull faces
  – Compute distance from plane with plane equation
  – If all distances are negative, point in hull
  – If any distance is positive, point outside hull

• Works in any dimension

• Can by used for basic frustum culling
Point in OBB

• An OBB is a convex hull!
  – Hold your horses here...

• Lets rotate point P into the frame of the OBB
  – OBB is defined with a rotation matrix, so invert it and multiply P by it
  – P is now in the basis of the OBB

• The problem is now point in AABB
Point in Cylinder

- Rotate cylinder axis to the z axis
- Ignoring the z axis, cylinder is a circle on the xy plane
- Test point in circle in 2D
  - If miss, exit no intersection
- Get points A and B. A is at top of cylinder, B at bottom
- See if point’s z component is less than A’s and greater than B’s
  - Return intersection
- No intersection
Bumper Car Damage?

• Two bumper cars hit each other
• One car takes damage
• How much damage is dealt, and to whom?
Bumper Car Damage Answer

• Take vector from one car to another, $T$ (normalized)
• Damage dealt:
  – $1.0 - \text{Abs}(\text{Dot}(\text{velocity}, T) \times \text{collisionDamage})$
Distance Point P to Line $\overline{AB}$

http://www.randygaul.net/2014/07/23/distance-point-to-line-segment/
Closest Point to Segment $\overrightarrow{AB}$ from P

- Take $\overrightarrow{AB}$ and compute a plane C
- C’s normal is $\overrightarrow{AB} / |\overrightarrow{AB}|$, and offset d is:
  - $d = \text{dot}(\overrightarrow{AB} / |\overrightarrow{AB}|, A)$
- Compute distance P from C
  - If C is negative, closest point is A
- Repeat process for B
Visualization in 2D
Cross Product

• Operation between vectors

\[ |v \times w| = |v||w|\sin(\gamma) \]

• Produces a vector orthogonal to both input vectors
Cross Product Handedness

Cross Product Details

http://upload.wikimedia.org/wikipedia/commons/thumb/6/6e/Cross_product.gif/220px-Cross_product.gif
Plane Equation from 3 Points

• Given three points A, B and C
• Calculate normal with: Cross( C – A, B – A )
• Normalize normal
• Compute offset d: dot( normal, A (or B or C) )
Intersection Segment $\overrightarrow{AB}$ and Plane

- Compute distance A to Plane: $d_a$
- Compute distance B to Plane: $d_b$
- If $d_a \times d_b < 0$
  - Intersection = $A + \left(\frac{d_a}{d_a - d_b}\right) \times (B - A)$
- Else
  - No intersection
Segment $\overrightarrow{AB}$ Intersects Triangle UVW

- If $\overrightarrow{AB}$ intersects triangle’s plane
  - Compute intersection point $P$ with plane
  - For each edge $V_aV_b$ along UVW, cross $V_a - P$
  - Dot result with plane normal
  - If dot result < 0
    - No intersection
  - Hit back side of UVW if dot($\overrightarrow{AB}$, normal) > 0
- Else
  - No intersection
Segment Intersects $\overrightarrow{AB}$ Triangle UVW (2)
Volume of Polygon

- Given reference point \( p \) (use origin to simplify computation)
- For each edge \( \{ a, b \} \) form triangle \( \{ p, a, b \} \) with normal \( \hat{n} \)
- Sum area of each triangle:

\[
A = \frac{1}{2} \sum (a - p) \times (b - a) \cdot \hat{n}
\]
Volume of Triangle Mesh

• Exact same idea except:
  – Form tetrahedrons from triangles and sum tetrahedron volumes

\[ V = \frac{1}{6} \sum (u \times v \cdot w) \]
Center of Mass of Triangle Mesh

- Sum all tetrahedron centroids, weight them by volume
- Center of mass is called centroid $C$

\[ C_i = \frac{1}{4} (a + b + c + p) \]

\[ C = \frac{1}{V} \sum_{i=1}^{n} V_i C_i \]
Affine Transformations

- Given matrix $A$, point $x$ and vector $b$
- An affine transformation is of the form:
  - $Ax + b$
Affine Transformations

• Given 3x3 matrix A, point x and vector b
• An affine transformation is of the form:
  \[ Ax + b \]
• With a 4x4 matrix we can represent Ax + b in block formation:

\[
\begin{bmatrix}
  A & b \\
  0^T & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  1
\end{bmatrix}
= 
\begin{bmatrix}
  A_{00} & A_{10} & A_{20} & b_0 \\
  A_{01} & A_{11} & A_{21} & b_1 \\
  A_{02} & A_{12} & A_{22} & b_2 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_0 \\
  x_1 \\
  x_2 \\
  1
\end{bmatrix}
\]
Translation

• Construct an affine transformation such that when multiplied with a point, translates the point by the vector $b$

$$\begin{bmatrix} I & b \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}$$

• $I$ is the identity matrix, and means no rotation (or scaling) occurs
Translation (2)

\[
\begin{bmatrix}
I & b \\
0^T & 1
\end{bmatrix}
\begin{bmatrix}
x \\
1
\end{bmatrix}
\]

- The 1 is important, it means it is a point
- If we had a zero here \( b \) wouldn’t affect \( x \)
Translation (3)

• Proof that translation doesn’t affect vectors:
  – Translate from P to Q by T
    • \( P - Q = T \)
    • \( = (P + T) - (Q + T) \)
    • \( = (P - Q) + (T - T) \)
    • \( = P - Q \)
Rotation

• Orthonormal basis into the top left of an affine transformation, without any translation vector:

\[
\begin{bmatrix}
R & 0 \\
0^T & 1
\end{bmatrix}
\begin{bmatrix}
x \\
1
\end{bmatrix}
\]
Scaling

• Given scaling vector $S$
• Take the matrix $A$
• Scale $A$’s $x$ axis’ $x$ component by $S_0$
• Scale $A$’s $y$ axis’ $y$ component by $S_1$
• Scale $A$’s $z$ axis’ $z$ component by $S_2$
Scaling (2)

\[
\begin{bmatrix}
A_{00} \ast S_0 & A_{10} & A_{20} & b_0 \\
A_{01} & A_{11} \ast S_1 & A_{21} & b_1 \\
A_{02} & A_{12} & A_{22} \ast S_2 & b_2 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
x_0 \\
x_1 \\
x_2 \\
1
\end{bmatrix}
\]
Exercises

• Extract scale, rotation and translation matrices from a single transformation matrix
• Efficiently invert (no determinant or Gaussian elimination) a transformation matrix
Camera - LookAt

• Given Point P
• Calculate vector V from Camera to LookAt( P )

\[
\begin{bmatrix}
0 \\
1 \\
0
\end{bmatrix}
\]

• Cross V and up vector \[\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\], this is Right vector

• Construct rotation matrix with V, Up and Right
  – These vectors (once normalized) form an orthonormal basis
  – This is the A matrix
• Negated camera position is the b vector:

\[
\begin{bmatrix}
A & b \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
1
\end{bmatrix}
\]
Camera – LookAt (2)

• Won’t work when player tries to look straight up or down
  – Parallel vectors crossed result in the zero vector

• Possible solutions:
  – Snap player’s view away from up/down
  – Use an if statement and cross with a different up vector
  – More solutions exist!
3D Animation

bone
3D Animation
3D Animation
3D Animation
3D Animation

bone = transformation
3D Animation

Transformation = Ax + b

\[
\begin{bmatrix}
A \\
0^T \\
1
\end{bmatrix}
\]
3D Animation

• 3D model is a tree of basis transformations
• Animation defined as:
  – Scaling, rotating and translating bones over time
• Given two keyframes, interpolate between them
  – Interpolation of angles
Angle Interpolation

interpolation

t₀  interpolation  t₁
Angle Interpolation (2)

• LERP, easy solution, can work
• Produces vectors that are “too short”
Angle Interpolation (3)

- LERP a lot (smaller timesteps)
- Better approximation
Angle Interpolation (4)

- Quaternion SLERP
- The correct angular interpolation
Angle Interpolation (5)

• But which space to interpolate in?
  – World?
  – Model?

• Interpolate each angle in space local to parent
  – Embarrassingly parallel
3D Mesh Skinning

• Out of scope of this slideshow
• See Engine Architecture by Jason Gregory
• Various Game Programming Gems articles
• Same idea as last few slides for the most part
• Are similar to barycentric coordinates (next slide)
Barycentric Coordinates (2D)
Barycentric Coordinates (2D)

A

P

B

\[ u = 0.5 \]
\[ v = 0.5 \]
Barycentric Coordinates (2D)

$u = 0.25$  \quad  v = 0.75$
Barycentric Coordinates (2D)

\[ u = -0.25 \quad v = 1.25 \]
Barycentric Coordinates (2D)

- $u$ and $v$ express $P$ in terms of $A$ and $B$
- $u$ and $v$ must add to 1
  - Weighted average of $A$ and $B$
- $u$ and $v$ are “fractional lengths”
Barycentric Coordinates (2D)

\( \hat{n} = \frac{B - A}{|B - A|} \)

\( u = \frac{(P - A) \cdot \hat{n}}{|B - A|} \)

\( v = \frac{(B - P) \cdot \hat{n}}{|B - A|} \)
POP QUIZ

• When can normalization be omitted when dealing with barycentric coordinates for a line segment?

• Answer: When only the sign of $u$ and $v$ is needed
ANOTHER POP QUIZ

• Okay not really quiz, just a fact:
  – Line to plane intersection is about computing the parameter \( t \) as \( u/v \)

• Mind = blown
Barycentric Triangle Coordinates

• u, v and w
• Segment u and v were “fractional lengths”
• u, v and w are “fractional areas” on triangle
Fractional Areas
Barycentric Triangle Coordinates

• $u + v + w = 1$

• If either $u$, $v$ or $w$ is negative
  – $P$ is not inside ABC
Fractional Areas

\[ u > 0 \]
\[ w > 0 \]
\[ v > 0 \]

\[ u = v = w = \frac{1}{3} \]
Fractional Areas (2)

C

P

A

B

\[ u > 0 \]

\[ v < 0 \]

\[ w > 0 \]

\[ v < 0 \]
Fractional Areas (3)

\[
\begin{align*}
\text{u} &= \frac{\text{area}(PBC)}{\text{area}(ABC)} \\
\text{v} &= \frac{\text{area}(PCA)}{\text{area}(ABC)} \\
\text{w} &= \frac{\text{area}(PAB)}{\text{area}(ABC)}
\end{align*}
\]
Barycentric Ray (Segment) to Triangle

- Project point to onto triangle plane
- Compute $u$, $v$ and $w$ of triangle to point
- If $u$, $v$ and $w$ are positive
  - Intersection
- Else
  - No intersection
Barycentric Coordinates Summary

• Fractional geometric values
• Useful for:
  – Collision detection
  – Raycasting
  – Nearest feature identification
  – Certain shader effects
  – Software triangle rasterization
  – ...

POP FACT #2

• $u, v$ and $w$ are coefficients in an affine combination of $A, B$ and $C$
POP FACT #2

• $u$, $v$ and $w$ are coefficients in an affine combination of $A$, $B$ and $C$

• What does this mean?
POP FACT #2

• u, v and w are coefficients in an affine combination of A, B and C
• What does this mean?

\[ ax + by + cz - d = 0 \]
• u, v and w are coefficients in an affine combination of A, B and C
• What does this mean?

\[ ax + by + cz - d = 0 \]

\[ ux + vy + wz - d = 0 \]
POP FACT #2

• u, v and w are coefficients in an affine combination of A, B and C

• What does this mean?

\[ ax + by + cz - \alpha = 0 \]
\[ ux + vy + wz - d = 0 \]
Questions?
Further Exercises

- Clip triangle above plane
- Compute intersection volume of two polygons (Sutherland-Hodgman clipping)
- Write software triangle rasterizer (without forward differencing) with barycentric coordinates
- Derive barycentric coordinates for a tetrahedron (should by 4 coordinates)
- Figure out why a non-linear mapping in $d$ dimensions can be linear in $d+1$ dimensions (think about 4D homogenous coordinates)
  - Hint, slides 52 and 53
- Deconstruct a rotation matrix into axis-angle representation
- Derive code to transform a ray from screen to world space
- Invert a 3x3 matrix using the dot and cross product
Further Readings

• Essential Mathematics by Jim Van Verth
• Everything ever published by Stan Melax
• Real-Time Collision Detection, Ericson
• GDC 2010, Erin Catto (Barycentric coordinates)
• 3D Animation and Skinning, Jeff Lander, Game Developer Magazine (2 separate articles)