Scene Graphs and Barycentric Coordinates

Michael Kazhdan

(601.457/657)

HB Ch. 5
FvDFH Ch. 5
Announcements

• Midterm
  ○ Wednesday, October 18th
Last Time

• 2D Transformations
  ◦ Basic 2D transformations
  ◦ Matrix representation
  ◦ Matrix composition

• 3D Transformations
  ◦ Basic 3D transformations
  ◦ Same as 2D
Homogeneous Coordinates

• Add a 4\textsuperscript{th} coordinate to every 3D point
  ○ \((x, y, z, w)\) represents a point at location \(\left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right)\)

• Represent transformations by \(4 \times 4\) matrices
  ○ The top-left \(3 \times 3\) block represents the linear part of the transformation
  ○ The last column represents the translation

\[
\begin{bmatrix}
  x' \\
  y' \\
  z' \\
  w'
\end{bmatrix} =
\begin{bmatrix}
  a & b & c & d \\
  e & f & g & h \\
  i & j & k & l \\
  0 & 0 & 0 & m
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  z \\
  w
\end{bmatrix}
\]

○ Transformations (translations/rotations/scales) can be composed using simple matrix multiplication
Overview

• Transformation Hierarchies
  ◦ Scene graphs
  ◦ Ray casting

• Barycentric Coordinates
Transformation Example 1

- An object may appear in a scene multiple times

Draw same 3D data with different transformations
Transformation Example 1

Building

Floor 1  Floor 2  Floor 3  Floor 4  Floor 5

Office 1  Office N

Bookshelf 1  Desk 1  Desk 2  Chair 1  Chair K
Transformation Example 1

Building

Floor 1
Floor 2
Floor 3
Floor 4
Floor 5

Office 1
Office N

Bookshelf 1
Desk 1
Desk 2
Chair 1
Chair K

Instances

Definitions

Bookshelf
Desk
Chair
The transformation applied to a node of the scene graph is the composition of transformations from the root.
Transformation Example 2

- Well-suited for humanoid characters
Scene Graphs

• Allow us to have multiple instances of a single model – reducing model storage size and making it easier to make consistent changes

• Allow us to model objects in local coordinates and then place them into a global frame – particularly important for animation
Scene Graphs

- Allow us to have multiple instances of a single model – reducing model storage size and making it easier to make consistent changes
- Allow us to model objects in local coordinates and then place them into a global frame – particularly important for animation
- Accelerate ray-tracing by providing a hierarchy that can be used for bounding volume testing
Ray Casting with Hierarchies

- Transform the ray ($M^{-1}$)
- Compute the intersection
- Transform the intersection ($M$)

$M = \begin{pmatrix} 1 & 0 \\ 0 & 1/2 \end{pmatrix}$

- Transform the shape ($M$)
- Compute the intersection
- Transform the intersection ($M$)
Ray Casting with Hierarchies

• Transform rays, not primitives
  ○ For each node ...
    » Transform ray by the inverse of matrix
    » Intersect transformed ray with primitives
    » Transform hit information by matrix
Applying a Transformation

- Position
- Direction
- Normal

\[
\begin{align*}
\text{Affine} & : M = \begin{pmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
\text{Translate} & : M_T = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
\text{Linear} & : M_L = \begin{pmatrix} a & b & c & 0 \\ d & e & f & 0 \\ g & h & i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{align*}
\]
Applying a Transformation

• Position
  ◦ Apply the full affine transformation:
    \[ p' = M(p) = (M_T \times M_L)(p) \]

• Direction

• Normal

\[
\begin{pmatrix}
  a & b & c & t_x \\
  d & e & f & t_y \\
  g & h & i & t_z \\
  0 & 0 & 0 & 1
\end{pmatrix}
= \begin{pmatrix}
  1 & 0 & 0 & t_x \\
  0 & 1 & 0 & t_y \\
  0 & 0 & 1 & t_z \\
  0 & 0 & 0 & 1
\end{pmatrix}
\times \begin{pmatrix}
  a & b & c & 0 \\
  d & e & f & 0 \\
  g & h & i & 0 \\
  0 & 0 & 0 & 1
\end{pmatrix}
\]
Applying a Transformation

- Position
- Direction
  - Apply the linear component of the transformation:
    \[ \mathbf{v'} = M_L(\mathbf{v}) \]
- Normal

\[
\begin{pmatrix}
a & b & c & t_x \\
d & e & f & t_y \\
g & h & i & t_z \\
0 & 0 & 0 & 1
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & 0 & t_x \\
0 & 1 & 0 & t_y \\
0 & 0 & 1 & t_z \\
0 & 0 & 0 & 1
\end{pmatrix}
\times \begin{pmatrix}
a & b & c & 0 \\
d & e & f & 0 \\
g & h & i & 0 \\
0 & 0 & 0 & 1
\end{pmatrix}
\]
Applying a Transformation

• Position

• Direction
  - Apply the linear component of the transformation:
    \[ \mathbf{v}' = M_L(\mathbf{v}) \]

A direction \( \mathbf{v} \) represents the difference between two positions: \( \mathbf{v} = p - q \)

The transformed direction is the difference of transformed positions:

\[
\mathbf{v}' = M(p) - M(q) \\
= (M_L(p) + \mathbf{t}) - (M_L(q) + \mathbf{t}) \\
= M_L(p) - M_L(q) \\
= M_L(\mathbf{v})
\]
Ray Casting With Hierarchies

• Transform rays, not primitives
  ○ For each node ...
    » Transform ray by inverse of matrix
    » Intersect transformed ray with primitives
    » Transform hit information by matrix
Transforming a Ray

- If $M$ is the local-to-global map at a scene-graph node, we transform a ray by setting:

$$(p, \vec{v}) \rightarrow \left(M^{-1}(p), M^{-1}_L(\vec{v})\right)$$

**Note:**
Even if $\vec{v}$ is unit-length, $\vec{v}'$ may not be.

<table>
<thead>
<tr>
<th>Affine</th>
<th>Translate</th>
<th>Linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\begin{pmatrix} a &amp; b &amp; c &amp; t_x \ d &amp; e &amp; f &amp; t_y \ g &amp; h &amp; i &amp; t_z \ 0 &amp; 0 &amp; 0 &amp; 1 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 1 &amp; 0 &amp; 0 &amp; t_x \ 0 &amp; 1 &amp; 0 &amp; t_y \ 0 &amp; 0 &amp; 1 &amp; t_z \ 0 &amp; 0 &amp; 0 &amp; 1 \end{pmatrix}$</td>
<td>$\begin{pmatrix} a &amp; b &amp; c &amp; 0 \ d &amp; e &amp; f &amp; 0 \ g &amp; h &amp; i &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 1 \end{pmatrix}$</td>
</tr>
</tbody>
</table>
Applying a Transformation

- Position
- Direction
- Normal

\[ \vec{n}' =? \]

\[
\begin{pmatrix}
    a & b & c & t_x \\
    d & e & f & t_y \\
    g & h & i & t_z \\
    0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    1 & 0 & 0 & t_x \\
    0 & 1 & 0 & t_y \\
    0 & 0 & 1 & t_z \\
    0 & 0 & 0 & 1
\end{pmatrix}
\times
\begin{pmatrix}
    a & b & c & 0 \\
    d & e & f & 0 \\
    g & h & i & 0 \\
    0 & 0 & 0 & 1
\end{pmatrix}
\]
Normal Transformation

2D Example:

\[
\begin{pmatrix}
1 & 0 & 1 \\
0 & 2 & 1 \\
0 & 0 & 1 \\
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & 1 \\
0 & 1 & 1 \\
0 & 0 & 1 \\
\end{pmatrix}
\times \begin{pmatrix}
1 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]

\(M \quad M_T \quad M_L\)

If \(\vec{v}\) is a direction in 2D, and \(\vec{n}\) is perpendicular to \(\vec{v}\), we want the transformed \(\vec{n}\) to be perpendicular to the transformed \(\vec{v}\):

\[
\langle \vec{v}, \vec{n} \rangle = 0 \quad \Rightarrow \quad \langle \vec{v'}, \vec{n'} \rangle = 0
\]

\(\Leftrightarrow\)

\[
\langle \vec{v}, \vec{n} \rangle = 0 \quad \Rightarrow \quad \langle M_L(\vec{v}), \vec{n'} \rangle = 0
\]
Normal Transformation

2D Example:

\[
\begin{bmatrix}
1 & 0 & 1 \\
0 & 2 & 1 \\
0 & 0 & 1 \\
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 1 \\
0 & 1 & 1 \\
0 & 0 & 1 \\
\end{bmatrix}
\times \begin{bmatrix}
1 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
\]

Say \( \vec{v} = (2,2) \)
Normal Transformation

2D Example:

\[
\begin{pmatrix}
1 & 0 & 1 \\
0 & 2 & 1 \\
0 & 0 & 1 \\
\end{pmatrix}
= \begin{pmatrix}
1 & 0 & 1 \\
0 & 1 & 1 \\
0 & 0 & 1 \\
\end{pmatrix}
\times \begin{pmatrix}
1 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]

Translate
Scale

\[M = M_T \times M_L\]

Say \(\vec{v} = (2,2)\)… then \(\vec{n} = (-\sqrt{5}, \sqrt{5})\)
Normal Transformation

2D Example:

\[
\begin{pmatrix}
1 & 0 & 1 \\
0 & 2 & 1 \\
0 & 0 & 1
\end{pmatrix} = \begin{pmatrix}
1 & 0 & 1 \\
0 & 1 & 1 \\
0 & 0 & 1
\end{pmatrix} \times \begin{pmatrix}
1 & 0 & 0 \\
0 & 2 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

Say \( \hat{v} = (2,2) \)... then \( \hat{n} = (-\sqrt{5}, \sqrt{5}) \)

Transforming \( M_L(\hat{v}) = (2,4) \) and \( M_L(\hat{n}) = (-\sqrt{5}, \sqrt{2}) \)

\[ \langle \hat{v}, \hat{n} \rangle = 0 \]

\[ \langle M_L(\hat{v}), M_L(\hat{n}) \rangle \neq 0 \]
Normal Transformation

2D Example:

Say \( \vec{v} = (2, 2) \) then \( \vec{n} = (-2\sqrt{5}, 2\sqrt{5}) \)

Simply applying the directional part of the transformation to \( \vec{n} \) does not give a vector perpendicular to the transformed \( \vec{v} \)!
Recall

Transposes:

• The transpose of a matrix $M$ is the matrix $M^t$ whose $(i, j)$ -th coeff. is the $(j, i)$ -th coeff. of $M$:

$$M = \begin{pmatrix} m_{11} & m_{21} & m_{31} \\ m_{12} & m_{22} & m_{32} \\ m_{13} & m_{23} & m_{33} \end{pmatrix} \quad M^t = \begin{pmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

Recall:

• For matrix $M$, the transpose of the transpose is $M$:

$$(M^t)^t = M$$
Recall

Transposes:

• The transpose of a matrix $M$ is the matrix $M^t$ whose $(i, j)$-th coeff. is the $(j, i)$-th coeff. of $M$:

$$M = \begin{pmatrix} m_{11} & m_{21} & m_{31} \\ m_{12} & m_{22} & m_{32} \\ m_{13} & m_{23} & m_{33} \end{pmatrix} \quad M^t = \begin{pmatrix} m_{11} & m_{21} & m_{31} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{pmatrix}$$

Recall:

• For matrices $M$ and $N$, the transpose of the product is the reversed product of the transposes:

$$(M \cdot N)^t = N^t \cdot M^t$$
Recall

**Dot-Products:**

- The dot product of two vectors \( \mathbf{v} = (v_x, v_y, v_z) \) and \( \mathbf{w} = (w_x, w_y, w_z) \) is obtained by summing the product of the coefficients:
  \[
  \langle \mathbf{v}, \mathbf{w} \rangle = v_x \cdot w_x + v_y \cdot w_y + v_z \cdot w_z
  \]
- We can also express this as a matrix product:
  \[
  \langle \mathbf{v}, \mathbf{w} \rangle = \mathbf{v}^t \cdot \mathbf{w} = (v_x \quad v_y \quad v_z) \cdot \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix}
  \]
Recall

Transposes and Dot-Products:

- If $M$ is a matrix, and $\vec{v}$ and $\vec{w}$ are vectors, then:
  \[
  \langle \vec{v}, M\vec{w} \rangle = \vec{v}^t \cdot (M \cdot \vec{w}) \\
  = (\vec{v}^t \cdot M) \cdot \vec{w} \\
  = (M^t \cdot \vec{v})^t \cdot \vec{w} \\
  = \langle M^t \vec{v}, \vec{w} \rangle
  \]
Applying a Transformation

A normal \( \mathbf{n} \) is defined by being perpendicular to some direction vector(s) \( \mathbf{v} \).

The transformed normal \( \mathbf{n}' \) should be perpendicular to the transformed direction(s):

\[
\langle \mathbf{n}, \mathbf{v} \rangle = \langle \mathbf{n}', M_L \mathbf{v} \rangle = \langle M_L^t \mathbf{n}', \mathbf{v} \rangle \equiv \mathbf{n} = M_L^t \mathbf{n}' \equiv \mathbf{n}' = (M_L^t)^{-1} \mathbf{n}
\]
## Applying a Transformation

- **Position**

\[ p' = M(p) \]

- **Direction**

\[ \vec{v}' = M_L(\vec{v}) \]

- **Normal**

\[ \vec{n}' = (M_L^t)^{-1}(\vec{n}) \]

<table>
<thead>
<tr>
<th>Affine</th>
<th>Translate</th>
<th>Linear</th>
</tr>
</thead>
</table>
| \[
\begin{pmatrix}
  a & b & c & t_x \\
  d & e & f & t_y \\
  g & h & i & t_z \\
  0 & 0 & 0 & 1 \\
\end{pmatrix}
\] | \[
\begin{pmatrix}
  1 & 0 & 0 & t_x \\
  0 & 1 & 0 & t_y \\
  0 & 0 & 1 & t_z \\
  0 & 0 & 0 & 1 \\
\end{pmatrix}
\] | \[
\begin{pmatrix}
  a & b & c & 0 \\
  d & e & f & 0 \\
  g & h & i & 0 \\
  0 & 0 & 0 & 1 \\
\end{pmatrix}
\] |
Ray Casting With Hierarchies

- Transform rays, not primitives
  - For each node...
    - Transform ray by inverse of matrix
    - Intersect transformed ray with primitives
    - Transform hit information by matrix

Robot Arm

Angel Figures 8.8 & 8.9
Transforming a Ray

- If $M$ is the local-to-global map at a scene-graph node, we transform the hit information by setting:

$$ (p, \vec{n}) \rightarrow (M(p), M_L^{-t}(\vec{n})) $$

<table>
<thead>
<tr>
<th>Affine</th>
<th>Translate</th>
<th>Linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\begin{pmatrix} a &amp; b &amp; c &amp; t_x \ d &amp; e &amp; f &amp; t_y \ g &amp; h &amp; i &amp; t_z \ 0 &amp; 0 &amp; 0 &amp; 1 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 1 &amp; 0 &amp; 0 &amp; t_x \ 0 &amp; 1 &amp; 0 &amp; t_y \ 0 &amp; 0 &amp; 1 &amp; t_z \ 0 &amp; 0 &amp; 0 &amp; 1 \end{pmatrix}$</td>
<td>$\begin{pmatrix} a &amp; b &amp; c &amp; 0 \ d &amp; e &amp; f &amp; 0 \ g &amp; h &amp; i &amp; 0 \ 0 &amp; 0 &amp; 0 &amp; 1 \end{pmatrix}$</td>
</tr>
</tbody>
</table>
Overview

• 3D Transformations
  ◦ Basic 3D transformations
  ◦ Same as 2D

• Transformation Hierarchies
  ◦ Scene graphs
  ◦ Ray casting

• Barycentric Coordinates
Triangles

These are the basic building blocks of 3D models.

- Often 3D models are complex, and the surfaces are represented by a triangulated approximation.
Recall:

Given vertices $p_1$ and $p_2$, a point $q$ on the line segment between the vertices is the (non-negatively) weighted average of $p_1$ and $p_2$:

$$q = \{ \alpha_q p_1 + \beta_q p_2 | \alpha_q + \beta_q = 1, \alpha_q, \beta_q \geq 0 \}$$
A triangle is defined by three non-collinear vertices:

- Any point $q$ in the triangle is on the line segment between $p_1$ and a point $q'$ on edge $p_2p_3$. 
Barycentric Coordinates

A triangle is defined by three non-collinear vertices:

- Any point \( q \) in the triangle is on the line segment between \( p_1 \) and a point \( q' \) on edge \( p_2p_3 \).
- Any point on the triangle can be expressed as:
  \[
  q = \{ \alpha_q p_1 + \beta_q p_2 + \gamma_q p_3 \mid \alpha_q + \beta_q + \gamma_q = 1, \alpha_q, \beta_q, \gamma_q \geq 0 \}
  \]
Barycentric Coordinates

A triangle is defined by three non-collinear vertices:

• Any point $q$ in the triangle is on the line segment between $p_1$ and a point $q'$ on edge $p_2p_3$.

• Any point on the triangle can be expressed as:

$$q = \{\alpha_q p_1 + \beta_q p_2 + \gamma_q p_3 \mid \alpha_q + \beta_q + \gamma_q = 1, \alpha_q, \beta_q, \gamma_q \geq 0\}$$

A point $q$ on the segment between $p_1$ and $q'$
Barycentric Coordinates

The barycentric coordinates of a point $q$:

$$q = \alpha_q p_1 + \beta_q p_2 + \gamma_q p_3$$

allow us to express $q$ as the weighted average of the vertices of the triangles.
Barycentric Coordinates

Any point on the triangle can be expressed as:

\[ q = \{\alpha_q p_1 + \beta_q p_2 + \gamma_q p_3 | \alpha_q + \beta_q + \gamma_q = 1, \alpha_q, \beta_q, \gamma_q \geq 0\} \]

Questions:

• What happens if \( \alpha_q, \beta_q, \) or \( \gamma_q < 0 \)?
Barycentric Coordinates

Any point on the triangle can be expressed as:

\[ q = \{\alpha_q p_1 + \beta_q p_2 + \gamma_q p_3 | \alpha_q + \beta_q + \gamma_q = 1, \alpha_q, \beta_q, \gamma_q \geq 0\} \]

Questions:

- What happens if \( \alpha_q, \beta_q, \text{ or } \gamma_q < 0 \)?
  - \( q \) is not inside the triangle but it is in the plane spanned by \( p_1, p_2, \text{ and } p_3 \).
Barycentric Coordinates

Any point on the triangle can be expressed as:

\[ q = \{\alpha_q p_1 + \beta_q p_2 + \gamma_q p_3 | \alpha_q + \beta_q + \gamma_q = 1, \alpha_q, \beta_q, \gamma_q \geq 0\} \]

Questions:

- What happens if \(\alpha_q, \beta_q, \) or \(\gamma_q < 0\)?
- What happens if \(\alpha_q + \beta_q + \gamma_q \neq 1\)?
Barycentric Coordinates

Any point on the triangle can be expressed as:

\[ q = \{\alpha_q p_1 + \beta_q p_2 + \gamma_q p_3 | \alpha_q + \beta_q + \gamma_q = 1, \alpha_q, \beta_q, \gamma_q \geq 0\} \]

Questions:

• What happens if \( \alpha_q, \beta_q, \text{ or } \gamma_q < 0 \)?

• What happens if \( \alpha_q + \beta_q + \gamma_q \neq 1 \)?
  - \( q \) may not be on the plane spanned by \( p_1, p_2, \text{ and } p_3 \).
Barycentric Coordinates

Barycentric coordinates are needed in:

• Ray-tracing, to test for intersection
• Rendering, to interpolate triangle information
Barycentric Coordinates

Barycentric coordinates are needed in:

- Ray-tracing, to test for intersection
- Rendering, to interpolate triangle information

```plaintext
Float TriangleIntersect( Ray r, Triangle tgl )
{
    Plane p = PlaneContaining( tgl );
    Float t = IntersectionDistance( r, p );
    if (t < 0 ) return -1;
    else
    {
        ( α , β , γ ) = Barycentric( r(t) , tgl );
        if( α<0 or β<0 or γ<0 ) return -1;
        else return t;
    }
}
```
Barycentric Coordinates

Barycentric coordinates are needed in:

- Ray-tracing, to test for intersection
- Rendering, to interpolate triangle information
  - In 3D models, information is often associated with vertices rather than triangles (e.g. color, normals, etc.)
Barycentric Coordinates

For example:

- We could associate the same normal/color to every point on the face of a triangle by computing:

\[
\vec{n} = \frac{(p_2 - p_1) \times (p_3 - p_1)}{\| (p_2 - p_1) \times (p_3 - p_1) \|}
\]
Barycentric Coordinates

For example:

- We could associate the same normal/color to every point on the face of a triangle by computing:

\[ \vec{n} = \frac{(p_2 - p_1) \times (p_3 - p_1)}{\| (p_2 - p_1) \times (p_3 - p_1) \|} \]

This gives rise to flat shading/coloring across the faces
Barycentric Coordinates

Instead:

- We associate normals to every vertex:

\[ T = ((p_1, \vec{n}_1), (p_2, \vec{n}_2), (p_3, \vec{n}_3)) \]

⇒ The normal at a point \( q \) in the triangle is the interpolation of the normals at the vertices:

\[
\vec{n}(q) = \frac{(\alpha_q \cdot \vec{n}_1 + \beta_q \cdot \vec{n}_2 + \gamma_q \cdot \vec{n}_3)}{\| (\alpha_q \cdot \vec{n}_1 + \beta_q \cdot \vec{n}_2 + \gamma_q \cdot \vec{n}_3) \|}
\]
Barycentric Coordinates

Instead:

- We associate normals to every vertex:

$$T = ((p_1, \vec{n}_1), (p_2, \vec{n}_2), (p_3, \vec{n}_3))$$

⇒ The normal at a point $q$ in the triangle is the interpolation of the normals at the vertices:

Triangle Normals

Interpolated Point Normals
Barycentric Coordinates

Given the points $p_1, p_2,$ and $p_3$, how do we compute the barycentric coordinates of a point $q$ in the plane spanned by $p_1, p_2,$ and $p_3$?
Barycentric Coordinates

Given the points \( p_1, p_2, \) and \( p_3 \), how do we compute the barycentric coordinates of a point \( q \) in the plane spanned by \( p_1, p_2, \) and \( p_3 \)?

(Signed) Area Ratios:

\[
\alpha_q = \frac{A_1}{A_1 + A_2 + A_3} \\
\beta_q = \frac{A_2}{A_1 + A_2 + A_3} \\
\gamma_q = \frac{A_3}{A_1 + A_2 + A_3}
\]
Barycentric Coordinates

Given the points $p_1$, $p_2$, and $p_3$, how do we compute the barycentric coordinates of a point $q$ in the plane spanned by $p_1$, $p_2$, and $p_3$?

(Signed) Area Ratios:

$$\alpha_q = \frac{A_1}{A_1 + A_2 + A_3}$$

$$\beta_q = \frac{A_2}{A_1 + A_2 + A_3}$$

$$\gamma_q = \frac{A_3}{A_1 + A_2 + A_3}$$

Solving this equation requires computing the areas of three triangles for every point $q$. 
Barycentric Coordinates

Given the points $p_1$, $p_2$, and $p_3$, how do we compute the barycentric coordinates of a point $q$ in the plane spanned by $p_1$, $p_2$, and $p_3$?

Matrix Inversion:

We can approach this is as a linear system with three equations and three unknowns:

\[
\begin{align*}
q_x &= \alpha_q \cdot p_{1x} + \beta_q \cdot p_{2x} + \gamma_q \cdot p_{3x} \\
q_y &= \alpha_q \cdot p_{1y} + \beta_q \cdot p_{2y} + \gamma_q \cdot p_{3y} \\
q_z &= \alpha_q \cdot p_{1z} + \beta_q \cdot p_{2z} + \gamma_q \cdot p_{3z}
\end{align*}
\]
Barycentric Coordinates

Given the points \( p_1, p_2, \) and \( p_3, \) how do we compute the barycentric coordinates of a point \( q \) in the plane spanned by \( p_1, p_2, \) and \( p_3? \)

Matrix Inversion:

We can approach this as a linear system with three equations and three unknowns:

\[
\begin{align*}
q_x &= \alpha_q \cdot p_{1x} + \beta_q \cdot p_{2x} + \gamma_q \cdot p_{3x} \\
q_y &= \alpha_q \cdot p_{1y} + \beta_q \cdot p_{2y} + \gamma_q \cdot p_{3y} \\
q_z &= \alpha_q \cdot p_{1z} + \beta_q \cdot p_{2z} + \gamma_q \cdot p_{3z}
\end{align*}
\]

\[
\begin{pmatrix}
q_x \\
q_y \\
q_z
\end{pmatrix}
= 
\begin{pmatrix}
p_{1x} & p_{2x} & p_{3x} \\
p_{1y} & p_{2y} & p_{3y} \\
p_{1z} & p_{2z} & p_{3z}
\end{pmatrix}
\begin{pmatrix}
\alpha_q \\
\beta_q \\
\gamma_q
\end{pmatrix}
\]
Barycentric Coordinates

Given the points $p_1, p_2, \text{ and } p_3$, how do we compute the barycentric coordinates of a point $q$ in the plane spanned by $p_1, p_2, \text{ and } p_3$?

Solving this equation requires inverting a matrix. However, the matrix is independent of the point $q$ and can be computed once and re-used for all points $q$.

Recall:

The system can be singular even for non-degenerate triangles.

\[
q_x = \alpha_q \cdot p_{1x} + \beta_q \cdot p_{2x} + \gamma_q \cdot p_{3x}
\]
\[
q_y = \alpha_q \cdot p_{1y} + \beta_q \cdot p_{2y} + \gamma_q \cdot p_{3y} \Leftrightarrow \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} (p_{1x} & p_{2x} & p_{3x}) \\ (p_{1y} & p_{2y} & p_{3y}) \\ (p_{1z} & p_{2z} & p_{3z}) \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}
\]
Barycentric Coordinates

Given the points \( p_1, p_2, \) and \( p_3 \), how do we compute the barycentric coordinates of a point \( q \) in the plane spanned by \( p_1, p_2, \) and \( p_3 \)?

Dot Products:

Suppose \( p_1 \) is at the origin.
Barycentric Coordinates

Given the points \( p_1, p_2, \) and \( p_3 \), how do we compute the barycentric coordinates of a point \( q \) in the plane spanned by \( p_1, p_2, \) and \( p_3 \)?

**Dot Products:**

Suppose \( p_1 \) is at the origin

Given \( q \), we want to find \( (\beta_q, \gamma_q) \):
such that \( q = \beta_q p_2 + \gamma_q p_3 \).
Barycentric Coordinates

Given the points $p_1$, $p_2$, and $p_3$, how do we compute the barycentric coordinates of a point $q$ in the plane spanned by $p_1$, $p_2$, and $p_3$?

Dot Products:

Suppose $p_1$ is at the origin

If

$q_1 = \beta_1 p_2 + \gamma_1 p_3$ and
$q_2 = \beta_2 p_2 + \gamma_2 p_3$

Then $q_1 + q_2 = (\beta_1 + \beta_2) p_2 + (\gamma_1 + \gamma_2) p_3$
Barycentric Coordinates

Given the points $p_1, p_2, \text{ and } p_3$, how do we compute the barycentric coordinates of a point $q$ in the plane spanned by $p_1, p_2, \text{ and } p_3$?

Dot Products:

The map taking $q$ to the parameters $(\beta_q, \gamma_q)$ is linear.
Barycentric Coordinates

Given the points $p_1$, $p_2$, and $p_3$, how do we compute the barycentric coordinates of a point $q$ in the plane spanned by $p_1$, $p_2$, and $p_3$?

**Dot Products:**

The map taking $q$ to the parameters $(\beta_q, \gamma_q)$ can be represented by a $3 \times 2$ matrix:

$$
\begin{pmatrix}
\beta_q \\
\gamma_q
\end{pmatrix}
= 
\begin{pmatrix}
a_1 & a_2 & a_3 \\
b_1 & b_2 & b_3
\end{pmatrix}
\begin{pmatrix}
q_x \\
q_y \\
q_z
\end{pmatrix}
$$
Barycentric Coordinates

Linearity:

\[
\begin{pmatrix}
\beta_q \\
\gamma_q
\end{pmatrix} =
\begin{pmatrix}
a_1 & a_2 & a_3 \\
b_1 & b_2 & b_3
\end{pmatrix}
\begin{pmatrix}
q_x \\
q_y \\
q_z
\end{pmatrix}
\]

If we set:

\[
v_\beta = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad \text{and} \quad v_\gamma = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}
\]
Barycentric Coordinates

Linearity:

\[
\begin{pmatrix}
\beta_q \\
\gamma_q
\end{pmatrix} =
\begin{pmatrix}
a_1 & a_2 & a_3 \\
b_1 & b_2 & b_3
\end{pmatrix}
\begin{pmatrix}
q_x \\
q_y \\
q_z
\end{pmatrix}
\]

If we set:

\[
v_\beta = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad \text{and} \quad v_\gamma = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}
\]

the barycentric coordinates can be expressed as dot-products:

\[
\beta_q = \langle q, v_\beta \rangle \\
\gamma_q = \langle q, v_\gamma \rangle
\]
Barycentric Coordinates

Linearity:

\[
\begin{pmatrix}
\beta_q \\
\gamma_q
\end{pmatrix}
= 
\begin{pmatrix}
a_1 & a_2 & a_3 \\
b_1 & b_2 & b_3
\end{pmatrix}
\begin{pmatrix}
q_x \\
q_y \\
q_z
\end{pmatrix}
\]

Goal:

Solve for vectors \( v_\beta \) and \( v_\gamma \) such that:

\[
\beta_q = \langle q, v_\beta \rangle \quad \gamma_q = \langle q, v_\gamma \rangle
\]
Barycentric Coordinates (Dot Products)

Solve for vectors \( v_\beta \) and \( v_\gamma \) such that:

\[
\beta_q = \langle q, v_\beta \rangle \quad \gamma_q = \langle q, v_\gamma \rangle
\]

- If \( q \) is on the line through \( p_1 \) and \( p_2 \) \( \Rightarrow \gamma_q = 0 \)
Barycentric Coordinates (Dot Products)

Solve for vectors $v_\beta$ and $v_\gamma$ such that:

$$\beta_q = \langle q, v_\beta \rangle \quad \gamma_q = \langle q, v_\gamma \rangle$$

- If $q$ is on the line through $p_1$ and $p_2 \Rightarrow \gamma_q = 0$
- If $q$ is on the line through $p_1$ and $p_3 \Rightarrow \beta_q = 0$
Barycentric Coordinates (Dot Products)

Solve for vectors $v_\beta$ and $v_\gamma$ such that:

\[
\beta_q = \langle q, v_\beta \rangle \quad \gamma_q = \langle q, v_\gamma \rangle
\]

- If $q$ is on the line through $p_1$ and $p_2 \Rightarrow \gamma_q = 0$
- If $q$ is on the line through $p_1$ and $p_3 \Rightarrow \beta_q = 0$
- If $q$ is on the line through $p_2$ and $p_3 \Rightarrow \beta_q + \gamma_q = 1$
Barycentric Coordinates (Dot Products)

Solve for vectors $\mathbf{v}_\beta$ and $\mathbf{v}_\gamma$ such that:

$$\beta_q = \langle q, \mathbf{v}_\beta \rangle \quad \gamma_q = \langle q, \mathbf{v}_\gamma \rangle$$

- If $q$ is on the line through $p_1$ and $p_2 \Rightarrow \gamma_q = 0$
- If $q$ is on the line through $p_1$ and $p_3 \Rightarrow \beta_q = 0$
- If $q$ is on the line through $p_2$ and $p_3 \Rightarrow \beta_q + \gamma_q = 1$
- If $q = p_2 \Rightarrow \beta_q = 1$
Barycentric Coordinates (Dot Products)

Solve for vectors $v_{\beta}$ and $v_{\gamma}$ such that:

$\beta_q = \langle q, v_{\beta} \rangle$ \quad $\gamma_q = \langle q, v_{\gamma} \rangle$

• If $q$ is on the line through $p_1$ and $p_2 \implies \gamma_q = 0$
• If $q$ is on the line through $p_1$ and $p_3 \implies \beta_q = 0$
• If $q$ is on the line through $p_2$ and $p_3 \implies \beta_q + \gamma_q = 1$
• If $q = p_2 \implies \beta_q = 1$
• If $q = p_3 \implies \gamma_q = 1$
Barycentric Coordinates (Dot Products)

Solve for vectors $\mathbf{v}_\beta$ and $\mathbf{v}_\gamma$ such that:

$$\beta_q = \langle q, \mathbf{v}_\beta \rangle \quad \gamma_q = \langle q, \mathbf{v}_\gamma \rangle$$

- If $q$ is on the line through $p_1$ and $p_2$ $\Rightarrow \gamma_q = 0$
- If $q$ is on the line through $p_1$ and $p_3$ $\Rightarrow \beta_q = 0$
- If $q$ is on the line through $p_2$ and $p_3$ $\Rightarrow \beta_q + \gamma_q = 1$
- If $q = p_2$ $\Rightarrow \beta_q = 1$
- If $q = p_3$ $\Rightarrow \gamma_q = 1$
- If $q = p_1$ $\Rightarrow \beta_q = \gamma_q = 0$
Barycentric Coordinates (Dot Products)

Solve for vectors \( v_\beta \) and \( v_\gamma \) such that:

\[
\beta_q = \langle q, v_\beta \rangle \quad \gamma_q = \langle q, v_\gamma \rangle
\]

- If \( q \) is on the line between \( p_1 \) and \( p_2 \) \( \Rightarrow \gamma_q = 0 \)
  - Set \( w_\gamma \) to be a vector perpendicular to \( p_2 \).
  - \( \Rightarrow \) If \( q \) is on the line between \( p_1 \) and \( p_2 \), \( \langle q, \overrightarrow{w_\gamma} \rangle = 0 \)
  - \( \Rightarrow \) \( v_\gamma = s_\gamma \cdot w_\gamma \) for some scalar \( s_\gamma \)
Barycentric Coordinates (Dot Products)

Solve for vectors $v_\beta$ and $v_\gamma$ such that:

\[ \beta_q = \langle q, v_\beta \rangle \quad \gamma_q = \langle q, v_\gamma \rangle \]

- If $q$ is on the line between $p_1$ and $p_2 \implies \gamma_q = 0$
- If $q$ is on the line between $p_1$ and $p_3 \implies \beta_q = 0$
 ◦ Set $w_\beta$ to be the vector…
Barycentric Coordinates (Dot Products)

Solve for vectors $v_\beta$ and $v_\gamma$ such that:

\[ \beta_q = \langle q, v_\beta \rangle \quad \gamma_q = \langle q, v_\gamma \rangle \]

- If $q$ is on the line between $p_1$ and $p_2$ $\Rightarrow \gamma_q = 0$
- If $q$ is on the line between $p_1$ and $p_3$ $\Rightarrow \beta_q = 0$
- $v_\beta = s_\beta \cdot w_\beta$ and
  
  $q = p_2$ $\Rightarrow \beta_q = 1$

\[ \Rightarrow v_\beta = \frac{w_\beta}{\langle p_2, w_\beta \rangle} \]
Barycentric Coordinates (Dot Products)

Solve for vectors $v_\beta$ and $v_\gamma$ such that:

$$
\beta_q = \langle q, v_\beta \rangle \quad \text{and} \quad \gamma_q = \langle q, v_\gamma \rangle
$$

- If $q$ is on the line between $p_1$ and $p_2 \Rightarrow \gamma_q = 0$
- If $q$ is on the line between $p_1$ and $p_3 \Rightarrow \beta_q = 0$

- $v_\beta = s_\beta \cdot w_\beta$ and
  
  $q = p_2 \Rightarrow \beta_q = 1$

- $v_\gamma = s_\gamma \cdot w_\gamma$ and
  
  $q = p_3 \Rightarrow \gamma_q = 1$

  ...
Barycentric Coordinates

Given the points $p_1$, $p_2$, and $p_3$, how do we compute the barycentric coordinates of a point $q$ in the plane spanned by $p_1$, $p_2$, and $p_3$?

- When $p_1$ is not at the origin, repeat the process but solving for the coordinates of $q - p_1$ relative to the directions $p_2 - p_1$ and $p_3 - p_1$. 