Scene Graphs and Barycentric Coordinates

Jason Lawrence

CS445: Graphics

Acknowledgment: slides by Misha Kazhdan, Allison Klein, Tom Funkhouser, Adam Finkelstein and David Dobkin
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 \( (x/w, y/w, z/w) \)

• Represent transformations by 4x4 matrices
  The top-left 3x3 block represents the linear part of the transformation
  The last column represents the translation

\[
\begin{bmatrix}
    x' \\
    y' \\
    z' \\
    w
\end{bmatrix} = \begin{bmatrix}
    a & b & c & t_x \\
    d & e & d & t_y \\
    g & h & e & t_z \\
    0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    z \\
    w
\end{bmatrix}
\]

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

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

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

• 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

- Floor Furniture
  - Office 1
  - Office N

- Office Furniture
  - Bookshelf 1
  - Desk 1
  - Desk 2
  - Chair 1
  - Chair K

Definitions
Instances
Transformation Example 2

- Well-suited for humanoid characters

```
Root
  /     \
|      |
|      |
Chest  LHip  RHip
/     /     /
|     |     |
|     |     |
Neck  LCollar  RCollar  LKnee  RKnee
/     /     /     /     /
|     |     |     |     |
|     |     |     |     |
Head  LShld  RShld  LAnkle  RAnkle
/     /     /     /     /
|     |     |     |     |
|     |     |     |     |
LElbow  RElbow
/     /     /
|     |     |
|     |     |
LWrist  RWrist
```

Rose et al. '96
Scene Graphs

• Allow us to have multiple instances of a single model – providing a reduction in model storage size

• 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 – providing a reduction in model storage size
- 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 hierarchical structure that can be used for bounding volume testing
Ray Casting with Hierarchies

\[ M = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \]
Ray Casting with Hierarchies

- Transform the shape ($M$)
- Compute the intersection

\[ M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \frac{1}{2} \end{bmatrix} \]
Ray Casting with Hierarchies

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

$M = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$
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
Applying a Transformation

- Position
- Direction
- Normal

\[
\begin{bmatrix}
a & b & c & tx \\
d & e & f & ty \\
g & h & i & tz \\
0 & 0 & 0 & 1
\end{bmatrix} = \begin{bmatrix}
1 & 0 & 0 & tx \\
0 & 1 & 0 & ty \\
0 & 0 & 1 & tz \\
0 & 0 & 0 & 1
\end{bmatrix} \times \begin{bmatrix}
a & b & c & 0 \\
d & e & f & 0 \\
g & h & i & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
Applying a Transformation

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

• Direction

• Normal

\[
\begin{bmatrix}
  a & b & c & tx \\
  d & e & f & ty \\
  g & h & i & tz \\
  0 & 0 & 0 & 1
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & 0 & tx \\
  0 & 1 & 0 & ty \\
  0 & 0 & 1 & tz \\
  0 & 0 & 0 & 1
\end{bmatrix}
\times
\begin{bmatrix}
  a & b & c & 0 \\
  d & e & f & 0 \\
  g & h & i & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]
Applying a Transformation

- Position
- Direction
  - Apply the linear component of the transformation:
    \[ p' = M_L(p) \]
- Normal

<table>
<thead>
<tr>
<th>Affine</th>
<th>Translate</th>
<th>Linear</th>
</tr>
</thead>
</table>
| \[
\begin{bmatrix}
  a & b & c & tx \\
  d & e & f & ty \\
  g & h & i & tz \\
  0 & 0 & 0 & 1 \\
\end{bmatrix}
\] | \[
\begin{bmatrix}
  1 & 0 & 0 & tx \\
  0 & 1 & 0 & ty \\
  0 & 0 & 1 & tz \\
  0 & 0 & 0 & 1 \\
\end{bmatrix}
\] | \[
\begin{bmatrix}
  a & b & c & 0 \\
  d & e & f & 0 \\
  g & h & i & 0 \\
  0 & 0 & 0 & 1 \\
\end{bmatrix}
\] |

\[ M \times M_T \times M_L \]
Applying a Transformation

• Position

• Direction
  ▪ Apply the linear component of the transformation:
    \[ p' = M_L(p) \]

A direction vector \( v \) is defined as the difference between two positional vectors \( p \) and \( q \): \( v = p - q \).
Applying a Transformation

- Position
- Direction
  - Apply the linear component of the transformation:
    \[ p' = M_L(p) \]

A direction vector \( v \) is defined as the difference between two positional vectors \( p \) and \( q \): \( v = p - q \).

Applying the transformation \( M \), we compute the transformed direction as the distance between the transformed positions: \( v = M(p) - M(q) \).
Applying a Transformation

- Position
- Direction
  - Apply the linear component of the transformation:
    \[ p' = M_L(p) \]

A direction vector \( \mathbf{v} \) is defined as the difference between two positional vectors \( \mathbf{p} \) and \( \mathbf{q} \):
\[ \mathbf{v} = \mathbf{p} - \mathbf{q} \].

Applying the transformation \( M \), we compute the transformed direction as the distance between the transformed positions:
\[ \mathbf{v} = M(p) - M(q) \].

The translation terms cancel out!
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 transformation mapping a scene-graph node into the global coordinate system, then we transform a ray $r$ by:
  - $r\.start = M^{-1}(r\.start)$
  - $r\.direction = M_L^{-1}(r\.direction)$

<table>
<thead>
<tr>
<th>Affine</th>
<th>Translate</th>
<th>Linear</th>
</tr>
</thead>
</table>
| \[
\begin{bmatrix}
a & b & c & tx \\
d & e & f & ty \\
g & h & i & tz \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\] | \[
\begin{bmatrix}
1 & 0 & 0 & tx \\
0 & 1 & 0 & ty \\
0 & 0 & 1 & tz \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\] | \[
\begin{bmatrix}
a & b & c & 0 \\
d & e & f & 0 \\
g & h & i & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\] |
Applying a Transformation

- Position
- Direction
- Normal

\[ p' = ? \]

Affine \[ M \]

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

Translate \[ M_T \]

\[
\begin{bmatrix}
  1 & 0 & 0 & t_x \\
  0 & 1 & 0 & t_y \\
  0 & 0 & 1 & t_z \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]

Linear \[ M_L \]

\[
\begin{bmatrix}
  a & b & c & 0 \\
  d & e & f & 0 \\
  g & h & i & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\]
Normal Transformation

2D Example:

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

\( M \quad M_T \quad M_L \)
Normal Transformation

2D Example:

If $v$ is a direction in 2D, and $n$ is a vector perpendicular to $v$, we want the transformed $n$ to be perpendicular to the transformed $v$:

$$\langle v, n \rangle = 0 \quad \Rightarrow \quad \langle M_L(v), n' \rangle = 0$$

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

$M$, $M_T$, $M_L$
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 \( \mathbf{v} = (2,2) \)...
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 \( v = (2,2) \)… then \( n = (-\sqrt{5}, \sqrt{5}) \)
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 \( \mathbf{v} = (2,2) \)… then \( \mathbf{n} = \left(-\sqrt{5}, \sqrt{5}\right) \)

Transforming, \( M_L(\mathbf{v}) = (2,4) \)…

\[\langle \mathbf{v}, \mathbf{n} \rangle = 0\]
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 \( v = (2,2) \ldots \) then \( n = (-\sqrt{5}, \sqrt{5}) \).
Transforming, \( M_L(v) = (2,4) \ldots \) and \( M_L(n) = (-\sqrt{5}, \sqrt{2}) \).

\[ \langle v, n \rangle = 0 \]
\[ \langle M_L(v), M_L(n) \rangle \neq 0 \]
Normal Transformation

2D Example:

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

\[M = M_T \times M_S\]

Simply applying the directional part of the transformation to \(n\) does not result in a vector that is perpendicular to the transformed \(v\).

\[
\langle v, n \rangle = 0
\]

\[
\langle M_L(v), M_L(n) \rangle \neq 0
\]
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 $M$:

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

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

- If $M$ and $N$ are two matrices, then the transpose of the product is the inverted product of the transposes:

  $$
  (MN)^t = N^t M^t
  $$
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:

\[
\langle \mathbf{v}, \mathbf{w} \rangle = v_x w_x + v_y w_y + v_z w_z
\]
Recall

Dot-Products:

- The dot product of two vectors $v = (v_x, v_y, v_z)$ and $w = (w_x, w_y, w_z)$ is obtained by summing the product of the coefficients:
  $\langle v, w \rangle = v_x w_x + v_y w_y + v_z w_z$

- Can also express as a matrix product:
  $\langle v, w \rangle = v^t w = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix} \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}$
Recall

Transposes and Dot-Products:

• If $M$ is a matrix, the dot product of $v$ with $M$ applied to $w$ is the dot product of the transpose of $M$ applied to $v$ with $w$: 
Recall

Transposes and Dot-Products:

• If $M$ is a matrix, the dot product of $v$ with $M$ applied to $w$ is the dot product of the transpose of $M$ applied to $v$ with $w$:

$$\langle v, Mw \rangle = v^t (Mw)$$
Recall

Transposes and Dot-Products:

- If $M$ is a matrix, the dot product of $v$ with $M$ applied to $w$ is the dot product of the transpose of $M$ applied to $v$ with $w$:

$$\langle v, Mw \rangle = v^t (Mw)$$

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

Transposes and Dot-Products:

- If $M$ is a matrix, the dot product of $v$ with $M$ applied to $w$ is the dot product of the transpose of $M$ applied to $v$ with $w$:

$$\langle v, Mw \rangle = v^t (Mw)$$

$$= (v^t M)v$$

$$= (M^tv)w$$
Recall

Transposes and Dot-Products:

• If \( M \) is a matrix, the dot product of \( v \) with \( M \) applied to \( w \) is the dot product of the transpose of \( M \) applied to \( v \) with \( w \):

\[
\langle v, Mw \rangle = v^t (Mw) \\
= \langle v^t M \rangle w \\
= \langle M^t v \rangle w \\
\langle v, Mw \rangle = \langle M^t v, w \rangle
\]
Applying a Transformation

• If we apply the transformation $M$ to 3D space, how does it act on normals?
Applying a Transformation

• If we apply the transformation $M$ to 3D space, how does it act on normals?

• A normal $n$ is defined by being perpendicular to some vector(s) $v$. The transformed normal $n'$ should be perpendicular to $M(v)$:

$$\langle n, v \rangle = \langle n', Mv \rangle$$
Applying a Transformation

• If we apply the transformation $M$ to 3D space, how does it act on normals?

• A normal $n$ is defined by being perpendicular to some vector(s) $v$. The transformed normal $n'$ should be perpendicular to $M(v)$:

$$\langle n,v \rangle = \langle n',Mv \rangle$$

$$= \langle M^t n',v \rangle$$
Applying a Transformation

• If we apply the transformation $M$ to 3D space, how does it act on normals?

• A normal $n$ is defined by being perpendicular to some vector(s) $v$. The transformed normal $n'$ should be perpendicular to $M(v)$:

$$
\langle n, v \rangle = \langle n', Mv \rangle \\
= \langle M^t n', v \rangle \\
n = M^t n'
$$
Applying a Transformation

• If we apply the transformation $M$ to 3D space, how does it act on normals?

• A normal $n$ is defined by being perpendicular to some vector(s) $v$. The transformed normal $n'$ should be perpendicular to $M(v)$:

$$\langle n, v \rangle = \langle n', Mv \rangle = \langle M^t n', v \rangle$$

$$n = M^t n'$$

$$n' = (M^t)^{-1} n$$
Applying a Transformation

• Position
  \[ p' = M(p) \]

• Direction
  \[ p' = M_L(p) \]

• Normal
  \[ p' = ((M_L)^t)^{-1}(p) \]

\[
\begin{bmatrix}
  a & b & c & tx \\
  d & e & f & ty \\
  g & h & i & tz \\
  0 & 0 & 0 & 1 \\
\end{bmatrix} =
\begin{bmatrix}
  1 & 0 & 0 & tx \\
  0 & 1 & 0 & ty \\
  0 & 0 & 1 & tz \\
  0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
  a & b & c & 0 \\
  d & e & f & 0 \\
  g & h & i & 0 \\
  0 & 0 & 0 & 1 \\
\end{bmatrix}
\]
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 transformation mapping a scene-graph node into the global coordinate system, then we transform the hit information $hit$ by:
  - $hit \cdot position = M \cdot (hit \cdot position)$
  - $hit \cdot normal = ((M_L)^t)^{-1}(hit \cdot normal)$

Affine
\[
\begin{bmatrix}
    a & b & c & tx \\
    d & e & f & ty \\
    g & h & i & tz \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]

Translate
\[
\begin{bmatrix}
    1 & 0 & 0 & tx \\
    0 & 1 & 0 & ty \\
    0 & 0 & 1 & tz \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]

Linear
\[
\begin{bmatrix}
    a & b & c & 0 \\
    d & e & f & 0 \\
    g & h & i & 0 \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]
Overview

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

• 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.
Triangles

A triangle is defined by three non-collinear vertices:

- Any point $q$ in the triangle is on the line segment between one vertex and some other point $q'$ on the opposite edge.
Barycentric Coordinates

A triangle is defined by three non-collinear vertices:

• Any point \( q \) in the triangle is on the line segment between one vertex and some other point \( q' \) on the opposite edge.

• Any point on the triangle can be expressed as:
  • \( q = \{ \alpha p_1 + \beta p_2 + \gamma p_3 \mid \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \} \)
Barycentric Coordinates

A triangle is defined by three non-collinear vertices:

• Any point $q$ in the triangle is on the line segment between one vertex and some other point $q'$ on the opposite edge.

• Any point on the triangle can be expressed as:
  • $q = \{ \alpha p_1 + \beta p_2 + \gamma p_3 \mid \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \}$

\[
\alpha p_1 + \beta p_2 + \gamma p_3 = \alpha p_1 + (1 - \alpha) \left( \frac{\beta p_2 + \gamma p_3}{1 - \alpha} \right)
\]
Barycentric Coordinates

A triangle is defined by three non-collinear vertices:

• Any point \( q \) in the triangle is on the line segment between one vertex and some other point \( q' \) on the opposite edge.

• Any point on the triangle can be expressed as:
  • \( q=\{ \alpha p_1 + \beta p_2 + \gamma p_3 \mid \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \} \)

\[ \alpha p_1 + \beta p_2 + \gamma p_3 = \alpha p_1 + (1-\alpha) \left( \frac{\beta p_2 + \gamma p_3}{1-\alpha} \right) \]

\[ = \alpha p_1 + (1-\alpha) \left( \frac{\beta p_2 + \gamma p_3}{\beta + \gamma} \right) \]

A point \( q \) on the segment between \( p_2 \) and \( p_3 \)
Barycentric Coordinates

A triangle is defined by three non-collinear vertices:

• Any point \( q \) in the triangle is on the line segment between one vertex and some other point \( q' \) on the opposite edge.

• Any point on the triangle can be expressed as:
  • \( q = \{ \alpha p_1 + \beta p_2 + \gamma p_3 \mid \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \} \)

\[
\alpha p_1 + \beta p_2 + \gamma p_3 = \alpha p_1 + (1-\alpha) \left( \frac{\beta p_2 + \gamma p_3}{\beta + \gamma} \right)
\]

A point \( q \) on the segment between \( p_1 \) and \( q' \)
Barycentric Coordinates

The barycentric coordinates of a point $q$:

$$q = \alpha p_1 + \beta p_2 + \gamma p_3$$

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

Any point on the triangle can be expressed as:

- \( q = \{ \alpha p_1 + \beta p_2 + \gamma p_3 \mid \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \} \)

Questions:

- What happens if \( \alpha, \beta, \) or \( \gamma < 0 \)?
Barycentric Coordinates

Any point on the triangle can be expressed as:

- \( q = \{ \alpha p_1 + \beta p_2 + \gamma p_3 \mid \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \} \)

Questions:

- What happens if \( \alpha, \beta, \) or \( \gamma < 0? \)
  - \( q \) is not inside the triangle but it is in the plane spanned by \( p_1, p_2, \) and \( p_3. \)
Barycentric Coordinates

Any point on the triangle can be expressed as:

- \( q = \{ \alpha p_1 + \beta p_2 + \gamma p_3 \mid \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \} \)

Questions:

- What happens if \( \alpha, \beta, \) or \( \gamma < 0 \)?
- What happens if \( \alpha + \beta + \gamma \neq 1 \)?
Barycentric Coordinates

Any point on the triangle can be expressed as:

\[ q = \{ \alpha p_1 + \beta p_2 + \gamma p_3 \mid \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \} \]

Questions:

• What happens if \( \alpha, \beta, \) or \( \gamma < 0 \)?

• What happens if \( \alpha + \beta + \gamma \neq 1 \)?
  
  \( q \) is not in the plane spanned by \( p_1, p_2, \) and \( p_3 \).
Barycentric Coordinates

Any point on the triangle can be expressed as:

\[ q = \{ \alpha p_1 + \beta p_2 + \gamma p_3 \mid \alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0 \} \]

Questions:

• What happens if \( \alpha, \beta, \) or \( \gamma < 0? \)

• What happens if \( \alpha + \beta + \gamma \neq 1? \)

Note: If we force \( \alpha = 1 - \beta - \gamma, \) we always get \( \alpha + \beta + \gamma = 1 \) so the point \( q \) is always in the plane containing the triangle.
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

```c
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:

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

For example:

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

\[
\mathbf{n} = \frac{\mathbf{p}_2 - \mathbf{p}_1 \times (\mathbf{p}_3 - \mathbf{p}_1)}{\|\mathbf{p}_2 - \mathbf{p}_1 \times (\mathbf{p}_3 - \mathbf{p}_1)\|}
\]

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

Instead:

• We could associate normals to every vertex:

\[ T = ((p_1, n_1), (p_2, n_2), (p_3, n_3)) \]

so that the normal at some point \( q \) in the triangle is the interpolation of the normals at the vertices:

\[
n(q) = \frac{\alpha(q)n_1 + \beta(q)n_2 + \gamma(q)n_3}{\|\alpha(q)n_1 + \beta(q)n_2 + \gamma(q)n_3\|}
\]
Barycentric Coordinates

Instead:

- We could associate normals to every vertex:
  \[ T = ((p_1, n_1), (p_2, n_2), (p_3, n_3)) \]
  so that the normal at some point \( q \) in the triangle
  is the interpolation of the normals at the vertices:

Triangle Normals

Interpolated Point Normals
Barycentric Coordinates

So 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 two unknowns:

$$
q_x = (1 - \beta - \gamma) p_{1x} + \beta p_{2x} + \gamma p_{2x}
$$

$$
q_y = (1 - \beta - \gamma) p_{1y} + \beta p_{2y} + \gamma p_{2y}
$$

$$
q_z = (1 - \beta - \gamma) p_{1z} + \beta p_{2z} + \gamma p_{2z}
$$
Barycentric Coordinates

So 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:

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

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

$$\gamma = \frac{A_3}{A_1 + A_2 + A_3}$$
Barycentric Coordinates

So 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:

$$
\alpha = \frac{A_1}{A_1 + A_2 + A_3} \\
\beta = \frac{A_2}{A_1 + A_2 + A_3} \\
\gamma = \frac{A_3}{A_1 + A_2 + A_3}
$$

Solving this equation requires computing the areas of three triangles for every point $q$. (DERIVATION IN CLASS)