Lecture 24: Barycentric Coordinates

ECE 417: Multimedia Signal Processing
Mark Hasegawa-Johnson

University of Illinois

11/14/2018
1. Overview of MP4

2. Barycentric Coordinates

3. Conclusion
Goal of MP4: Generate video frames (right) by warping a static image (left).
How it is done (Full walkthrough: Tuesday November 27)

```
lip_height, width = NeuralNet(MFCC)
out_triangles = LinearlyInterpolate(inp_triangles, lip_height, width)
inp_coord = BaryCentric(out_coord, inp_triangles, out_triangles)
out_image = BilinearInterpolate(inp_coord, inp_image)
```
Outline

1. Overview of MP4

2. Barycentric Coordinates

3. Conclusion
Affine Transformations

* Combines linear transformations, and Translations

\[
\begin{bmatrix}
  x' \\
  y' \\
  w'
\end{bmatrix} =
\begin{bmatrix}
  a & b & c \\
  d & e & f \\
  0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  w
\end{bmatrix}
\]

the ones we looked at, that were the you know the rotation scaling and
OK, so somebody's given us a lot of points, arranged like this in little triangles.

We know that we want a DIFFERENT AFFINE TRANSFORM for EACH TRIANGLE. For the $k^{th}$ triangle, we want to have

$$A_k = \begin{bmatrix} a_k & b_k & c_k \\ d_k & e_k & f_k \\ 0 & 0 & 1 \end{bmatrix}$$
Piece-wise affine transform

output point: \( \vec{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \), input point: \( \vec{u} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \)

**Definition:** if \( \vec{x} \) is in the \( k^{th} \) triangle in the output image, then we want to use the \( k^{th} \) affine transform:

\[
\vec{x} = A_k \vec{u}, \quad \vec{u} = A_k^{-1} \vec{x}
\]
If it is known that \( \vec{u} = A_k^{-1} \vec{x} \) for some unknown affine transform matrix \( A_k \),

then

the method of barycentric coordinates finds \( \vec{u} \)

without ever finding \( A_k \).
Barycentric Coordinates

Barycentric coordinates turns the problem on its head. Suppose \( \vec{x} \) is in a triangle with corners at \( \vec{x}_1, \vec{x}_2, \) and \( \vec{x}_3 \). That means that

\[
\vec{x} = \lambda_1 \vec{x}_1 + \lambda_2 \vec{x}_2 + \lambda_3 \vec{x}_3
\]

where

\[
0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1
\]

and

\[
\lambda_1 + \lambda_2 + \lambda_3 = 1
\]
Barycentric Coordinates

Suppose that all three of the corners are transformed by some affine transform $A$, thus

$$
\vec{u}_1 = A\vec{x}_1, \quad \vec{u}_2 = A\vec{x}_2, \quad \vec{u}_3 = A\vec{x}_3
$$

Then if

$$
\vec{x} = \lambda_1 \vec{x}_1 + \lambda_2 \vec{x}_2 + \lambda_3 \vec{x}_3
$$

Then:

$$
\vec{u} = A\vec{x}
= \lambda_1 A\vec{x}_1 + \lambda_2 A\vec{x}_2 + \lambda_3 A\vec{x}_3
= \lambda_1 \vec{u}_1 + \lambda_2 \vec{u}_2 + \lambda_3 \vec{u}_3
$$

In other words, once we know the $\lambda$'s, we no longer need to find $A$. We only need to know where the corners of the triangle have moved.
Barycentric Coordinates

If
\[ \vec{x} = \lambda_1 \vec{x}_1 + \lambda_2 \vec{x}_2 + \lambda_3 \vec{x}_3 \]

Then
\[ \vec{u} = \lambda_1 \vec{u}_1 + \lambda_2 \vec{u}_2 + \lambda_3 \vec{u}_3 \]
How to find Barycentric Coordinates

But how do you find \( \lambda_1 \), \( \lambda_2 \), and \( \lambda_3 \)?

\[
\vec{x} = \lambda_1 \vec{x}_1 + \lambda_2 \vec{x}_2 + \lambda_3 \vec{x}_3 = [\vec{x}_1, \vec{x}_2, \vec{x}_3] \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix}
\]

Write this as:

\[
\vec{x} = X \vec{\lambda}
\]

Therefore

\[
\vec{\lambda} = X^{-1} \vec{x}
\]

This **always works**: the matrix \( X \) is always invertible, unless all three of the points \( \vec{x}_1 \), \( \vec{x}_2 \), and \( \vec{x}_3 \) are on a straight line.
How do you find out which triangle the point is in?

- Suppose we have $K$ different triangles, each of which is characterized by a $3 \times 3$ matrix of its corners

$$X_k = [\vec{x}_{1,k}, \vec{x}_{2,k}, \vec{x}_{3,k}]$$

where $\vec{x}_{m,k}$ is the $m^{th}$ corner of the $k^{th}$ triangle.

- Notice that, for any point $\vec{x}$, for ANY triangle $X_k$, we can find

$$\lambda = X_k^{-1}\vec{x}$$

- However, the coefficients $\lambda_1$, $\lambda_2$, and $\lambda_3$ will all be between 0 and 1 if and only if the point $\vec{x}$ is inside the triangle $X_k$. Otherwise, some of the $\lambda$'s must be negative.
The Method of Barycentric Coordinates

To construct the animated output image frame \( O(x, y) \), we do the following things:

- First, for each of the reference triangles \( U_k \) in the input image \( I(u, v) \), decide where that triangle should move to. Call the new triangle location \( X_k \).
- Second, for each output pixel \((x, y)\):
  - For each of the triangles, find \( \vec{\lambda} = X_k^{-1} \vec{x} \).
  - Choose the triangle for which all of the \( \lambda \) coefficients are \( 0 \leq \lambda \leq 1 \).
  - Find \( \vec{u} = U_k \vec{\lambda} \).
  - Estimate \( I(u, v) \) using bilinear interpolation.
  - Set \( O(x, y) = I(u, v) \).
Outline

1. Overview of MP4
2. Barycentric Coordinates
3. Conclusion
lip_height, width = NeuralNet (MFCC)
out_triangs = LinearlyInterpolate (inp_triangs, lip_height, width)
inp_coord = BaryCentric (out_coord, inp_triangs, out_triangs)
out_image = BilinearInterpolate (inp_coord, inp_image)