Feature-Based Warping

Outline
- Feature-based warping
- Mapping relations
  - Under single pair of control line segments
  - Under multiple pairs of control line segments

Feature-Based Warping

- Proposed by Beier & Neeley
- Part of their feature-based morphing algorithm
- Basic ideas:
  - Use line segments to define features
  - Backward warping is used

Advantages of Feature-Based Approach
- Function-based:
  - The mapping relationship is specified using a global function
  - Any change to the function affects the whole image
  - Hard to design a function that can achieve a given effect
- Feature-based:
  - The mapping relationship is specified using control line segments
  - User can control where each feature in the image warps to
  - Easy to specify a warping effect through human interaction

Mapping Relation
- A control line segment in the destination image defines a local coordinate
  - Pixel p is converted to local coordinate (u, v)
  - Corresponding control line in the source image also defines a local coordinate
    - The same (u, v) is used to find pixel P'

Local Coordinate
- Defined by the control line and the direction perpendicular to the line
  - v is a distance:
    - Signed distance between point and the control line
  - u is a ratio:
    - Relative position of the point along the line
**Vector Representation**

- Coordinates of a point can be represented as a 2D vector:
  - \( A = (A_x, A_y) \)
- Length (\( l^2\)-norm):
  - \( |A| = \sqrt{A_x^2 + A_y^2} \)
- Vector arithmetic:
  - \( A + AB = B \)
  - \( B - A = AB \)
- Dot product:
  - \( A \cdot B = A_xB_x + A_yB_y \)

**Global to Local**

- Calculate \( u \):
  - \( |AQ| = (P-A) \cdot (B-A) / |AB| \)
  - \( u = |AQ| / |AB| \)
  - \( u = (P-A) \cdot (B-A)|AB|^2 \)
- Find direction \( D \) that is perpendicular to \( AB \):
  - \( D_x = (B_y - A_y) \)
  - \( D_y = -(B_x - A_x) \)
  - \( D = D / |D| \)
- Calculate \( v \):
  - \( v = (P-A) \cdot D \)

**Local to Global**

- Find direction \( D' \) that is perpendicular to \( A'B' \):
  - \( D'_x = (B'_y - A'_y) \)
  - \( D'_y = -(B'_x - A'_x) \)
  - \( D' = D' / |D'| \)
- Calculate \( P' \):
  - \( P' = A' + u \cdot (B' - A') + v \cdot D' \)

**Warping Effects (Translation)**

- The corresponding control lines:
  - Same lengths
  - Same directions
  - Different positions

**Warping Effects (Scale)**

- The corresponding control lines:
  - Same directions
  - Different lengths

**Warping Effects (Rotation)**

- The corresponding control lines:
  - Same lengths
  - Different directions
Multiple Control Lines

- When more than one control lines are defined:
  - Each control line is used to compute a candidate point
  - Different candidate points are weighted-averaged to get the corresponding point

![Diagram of Multiple Control Lines]

Weighted Average

- Motivations:
  - Longer control lines have stronger control
  - Closer control lines have stronger control
- Weight function: \( w_i = \left( \frac{l_i}{a + d_i} \right)^b \)
  - \( l_i \): length of control line \( i \);
  - \( d_i \): distance between point and control line \( i \);
  - \( a, b, c \): adjustable parameters
  - Suggested setting: \( a=0.0001; b=1; c=2; \)
- Final corresponding point:
  \[ P' = \frac{\sum (P_{i'} \cdot w_i)}{\sum w_i} \]

Distance Calculation

- The closest distance between a point & the line segment
  - NOT the line
- Algorithm:
  - if \( u < 0 \)
    \[ d = |P-A| \]
  - else if \( u > 1 \)
    \[ d = |P-B| \]
  - else
    \[ d = |v| \]

Warping Effects (Two Control Line Segments)

- Under two control line segments:
  - Both line segments try to control the warping effects
  - The final result is a compromise between the two control features

![Diagram of Warping Effects]

Pseudocode

- for ( each pixel \( P \) in the destination image ) {
  \[ \sum_P = (0,0); \sum_w = 0; \]
  - for ( each control line \( i \) in destination image ) {
    - \((u,v) = \text{transfer } P \text{ to the local coordinate of line } i;\)
    - \( w = \text{weight calculated based control line } i; \)
    - \( P_{\text{src}} = \text{global coordinate of } (u,v) \text{ in source image}; \)
    - \( \sum_P += P_{\text{src}} \cdot w; \sum_w += w; \)
  - }
  \[ P_{\text{src}} = \frac{\sum_P}{\sum_w}; \]
  \[ \text{destination}(P) = \text{sampleSource}(P_{\text{src}}); \]
}

![Diagram of Pseudocode]