Feature-Based Warping

Outline
- Feature-based warping
  - Feature-based vs. function-based
- 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 affect 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 destination image defines a local coordinate
  - Pixel \( p \) is converted to local coordinate \((u, v)\)
  - Corresponding control line in source image also defines a local coordinate
  - The same \((u, v)\) is used to find pixel \( p' \)

Local Coordinate
- Defined by the control line & 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
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

Vector Representation

- Coordinates of a point can be represented as a 2D vector from origin to point location:
  \[ \mathbf{A} = (A_x, A_y) \]
- Vector length (l2-norm):
  \[ |\mathbf{A}| = \sqrt{A_x^2 + A_y^2} \]
  - For unit vector, \(|\mathbf{A}| = 1\)
- Vector addition:
  \[ \mathbf{A} + \mathbf{B} = (A_x + B_x, A_y + B_y) \]

Global to Local

- Calculate \( u \):
  \[ (P - \mathbf{A}) \cdot (\mathbf{B} - \mathbf{A}) = |\mathbf{A}\mathbf{Q}| \times |\mathbf{AB}| \]
  \[ u = \frac{|\mathbf{A}\mathbf{Q}|}{|\mathbf{AB}|} = \frac{(P-A)\cdot(P-\mathbf{A})}{|\mathbf{AB}|^2} \]
- Find direction \( \mathbf{D} \) that is perpendicular to \( \mathbf{AB} \):
  \[ \mathbf{D} = (\mathbf{AB}_y, -\mathbf{AB}_x) = (B_y - A_y, -B_x + A_x) \]
- Calculate \( v \):
  \[ (P - \mathbf{A}) \cdot \mathbf{D} = v \times |\mathbf{D}| \]
  \[ v = \frac{(P - \mathbf{A}) \cdot \mathbf{D}}{|\mathbf{AB}|} \]
Local to Global

- Find direction $\mathbf{D}'$ that is perpendicular to $\mathbf{A}'\mathbf{B}'$:
  \[ \mathbf{D}' = (\mathbf{A}'\mathbf{B}'_y - \mathbf{A}'\mathbf{B}'_x) = (\mathbf{B}'_y - \mathbf{A}'_y, -\mathbf{B}'_x + \mathbf{A}'_x) \]
- The unit vector is $\mathbf{D}'/|\mathbf{D}'|$.
- Calculate $\mathbf{P}'$ through following a set of vectors:
  \[
  \mathbf{P}' = \mathbf{A}' + u(\mathbf{B}' - \mathbf{A}') + \frac{v}{|\mathbf{D}'|},
  \]

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

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

Weighted Average

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

Distance Calculation

- The closest distance between a point $\mathbf{P}$ and the line segment
  - NOT the line
- Algorithm:
  - if $u < 0$
    - $d = |\mathbf{P}-\mathbf{A}|$;
  - else if $u > 1$
    - $d = |\mathbf{P}-\mathbf{B}|$;
  - else
    - $d = |v|$;

Pseudocode

- for each pixel $\mathbf{P}$ in the destination image
  - sum$\_w = 0$; sum$\_w$ = 0;
  - for each control line $i$ in the destination image
    - $(u,v)$ = transfer $\mathbf{P}$ to the local coordinate of line $i$;
    - $w$ = weight calculated for control line $i$;
    - $\mathbf{P}$ = global coordinate of $(u,v)$ in source image;
    - sum$\_\mathbf{P}$ = $\mathbf{P}$; sum$\_w$ = $w$;
  - destination($\mathbf{P}$) = sampleSource($\mathbf{P}$, sum$\_w$);