Feature-Based Warping

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

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
- Featured-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 $p'$

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

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

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

Vector Representation (Cont’d)
- Scalar-vector multiplication:
  \[ k \cdot A = (k \cdot A_x, k \cdot A_y) \]
  - Scale the length of vector \( A \)
- Dot product:
  \[ A \cdot B = A_xB_x + A_yB_y \]
  - Result is a scalar
  - Equal to the projection length if \( A \) is a unit vector

Global to Local
- Calculate \( u \):
  \[ |AQ| = \frac{(P-A) \cdot (B-A)}{|AB|} \]
  - \( u = \frac{|AQ|}{|AB|} \)
- Find direction \( D \) that is perpendicular to \( AB \):
  \[ D_x = (B_y - A_y) \]
  \[ D_y = -(B_x - A_x) \]
  \[ D = \frac{D}{|D|} \]
- 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 $P'$:
  - $P' = A' + u * (B' - A') + v * D'$

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: $w_i = (l_i^b / (a + d_i)^c)$
  - $l_i$: 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:
  - $P' = \sum (P_i' * w_i) / \sum w_i$

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
  - $d = |v|$

Pseudocode

```c
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) = transfer P to the local coordinate of line i;
        w = weight calculated based control line i;
        P_src = global coordinate of (u,v) in source image;
        sum_P += P_src * w; sum_w += w;
    }
    P_src = sum_P / sum_w;
    destination(P) = sampleSource(P_src);
}
```