Features Extraction for Sketch-Based Recognition

Lecture #8: Feature Extraction
Joseph J. LaViola Jr.
Fall 2011

Recall Pen-Based Interface Dataflow

- Raw Stroke Data
- Preprocessing
- Segmentation
- Ink Parsing
- Classification
- Feature Extraction and Analysis
- Sketch Understanding
- Make Inferences
Feature Extraction and Analysis

- What came first, the feature or the machine learning algorithm?
- Want to distinguish sketch components from one another
- Good features are critical
- Extract important information
  - geometrical, statistical, contextual
- Examples include
  - arc length, histograms, cusps, aspect ratio
  - self-intersections, stroke area, etc...

Finding Features

- Challenging problem
  - need fast algorithms for gathering information
  - features must be good discriminators
- Often trial and error
- Can be domain specific
Geometric Features (1)

- Number of strokes
  - if you know how many strokes a symbol has, you can break up your recognizer into pieces (i.e., recognizer for 1 stroke symbols, recognizer for 2 stroke symbols …)

- Cusps
  - smooth vs. jagged strokes
  - distance between cusps
    - useful for when cusps are close together/far apart

Geometric Features (2)

- Aspect ratio (width / height)
  - tall vs. flat

- Self Intersections
  - loops vs. no loops
  - strokes with write over
  - distance between self intersections also useful
  - use line segment intersection algorithm
Geometric Features (3)

- First and last distance
  - Strokes where first and last points are close together vs. far apart
  - Simple computation – $\|p_n - p_1\|

- Arc length
  - Many different symbols have varying arc lengths
  - Simple computation as well –

$$l = \sum_{i=2}^{n} \|p_i - p_{i-1}\|$$

Geometric Features (4)

- Stroke area
  - Area defined by the vectors created with the initial stroke point and consecutive stroke points.
  - Good discriminator for straight vs. curved lines

Given $\mathbf{u}_i = p_{i+1} - p_i$ and $\mathbf{v}_i = p_{i+2} - p_i$

$$s_{area} = \sum_{i=1}^{n-2} \frac{1}{2} (\mathbf{u}_i \times \mathbf{v}_i) \cdot \text{sgn}(\mathbf{u}_i \times \mathbf{v}_i)$$

where $\mathbf{u}_i \times \mathbf{v}_i$ is a scalar
Geometric Features (5)

- Fit line feature
  - sophisticated approach to finding how close a stroke is to a straight line
  - finds a least-squares approximation to a line using principal components and then uses this approximation to find the distance of the projection of the stroke points onto the approximated line
  - outputs a value in [0,1]

- What is another name for this approach?

Fit Line Feature Implementation

Input: A set of stroke points P.
Output: A distance measure

FitLine(P)

1. $x_1 = \sum_{i=1}^{n} X(P_i)$
2. $y_1 = \sum_{i=1}^{n} Y(P_i)$
3. $x_2 = \sum_{i=1}^{n} X(P_i)^2$
4. $y_2 = \sum_{i=1}^{n} Y(P_i)^2$
5. $xy_1 = \sum_{i=1}^{n} X(P_i)Y(P_i)$
6. $x_3 = x_2 - \frac{x_1^2}{n}$
7. $y_3 = y_2 - \frac{y_1}{n}$
8. $x_{y2} = \frac{x_2 - x_1}{y_1}$
9. $rad = \sqrt{(x_3 - y_3)^2 + 4x_{y2}^2}$
10. $error = \frac{(x_3 + y_3 - rad)}{2}$
11. $rms = \sqrt{error/n}$
12. if $x_3 > y_3$
   13. $a = -2x_{y2}$
14. $b = x_3 - y_3 + rad$
15. else if $x_3 < y_3$
   16. $a = y_3 - x_3 + rad$
   17. $b = -2x_{y2}$
18. else
   19. if $xy_2 = 0$
      20. $a = b = c = 0$
      21. $error = +\infty$
   22. else
      23. $a = -1$
      24. $b = -1$
      25. $mag = \sqrt{a^2 + b^2}$
      26. $c = \frac{(a \cdot x_3 - b \cdot y_3)}{n}$
      27. $a = \frac{\text{mag}}{mag}$
      28. $b = \frac{\text{mag}}{mag}$
      29. $min_1 = +\infty$
      30. $max_1 = -\infty$
31. for i=1 to n
32. $err = aX(P_i) + bY(P_i) + c$
33. $pX = X(P_i) - a \cdot err$
34. $pY = Y(P_i) - b \cdot err$
35. $ploc = -b \cdot pX + b \cdot pY$
36. $min_1 = \min(min_1, ploc)$
37. $max_1 = \max(max_1, ploc)$
38. return $\frac{100 \cdot rms}{max - min}$
Statistical Features (1)

- **Side ratios**
  - first and last point of strokes have variable locations with respect to the bounding box
  - **Approach**
    - take the x coordinates of the first and last point of a stroke
    - subtract them from the left side of the symbol’s bounding box (i.e., the bounding box’s leftmost x value)
    - divide by the bounding box width.

Statistical Features (2)

- **Top and Bottom ratios**
  - similar to side ratios except we are dealing with y coordinate
  - **approach**
    - take y coordinate of the first and last point of a stroke
    - subtract from the top of the symbol’s bounding box (i.e., the bounding box’s topmost y value)
    - these values are divided by the bounding box height.
Statistical Features (3)

- **Point Histogram**
  - distribution of point locations in stroke bounding box
  - discrimination where point concentrations are high
  - approach
    - break up box into $n \times m$ grid
    - Count number of points in each sub box
    - divide by total number of points

Statistical Features (4)

- **Angle Histogram**
  - similar to point histogram except dealing with angles
  - Approach
    
    \[
    \tilde{v}_j = p_j - p_{j-1} \text{ for } 2 \leq i \leq n \text{ and } \tilde{x} = (1,0)
    \]
    
    \[
    \alpha_j = \arccos \left( \frac{\tilde{x} \cdot \tilde{v}_j}{\|\tilde{v}_j\|} \right)
    \]
    
    - put angles into bins of $n$ degrees
The Rubine Feature Set (Rubine 1991)

- Part of Rubine’s gesture recognition system
  - we will see this next class
- Stroke
  - $P = \text{total number of points}$
  - $p = \text{middle point}$
  - first point $(x_0, y_0, t_0)$
  - last point $(x_{P-1}, y_{P-1}, t_{P-1})$
  - compute $x_{\text{min}}, y_{\text{min}}, x_{\text{max}}, y_{\text{max}}$

Feature $f_1$

- Cosine of starting angle

$$f_1 = \cos(\alpha) = \frac{(x_2 - x_0)}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}}$$
Feature $f_2$

- Sine of starting angle

\[ f_2 = \sin(\alpha) = \frac{(y_2 - y_0)}{\sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}} \]

Feature $f_3$

\[ f_3 = \sqrt{(x_{\text{max}} - x_{\text{min}})^2 + (y_{\text{max}} - y_{\text{min}})^2} \]

- Length of diagonal of bounding box (gives an idea of the size of the bounding box)
Feature $f_4$

- Angle of diagonal
- Gives an idea of the shape of the bounding box (long, tall, square)

$$f_4 = \arctan\left(\frac{y_{\max} - y_{\min}}{x_{\max} - x_{\min}}\right)$$

Feature $f_5$

Distance from start to end of stroke

$$f_5 = \sqrt{(x_{p-1} - x_0)^2 + (y_{p-1} - y_0)^2}$$
**Feature \( f_6 \)**

- Cosine of ending angle

\[
f_6 = \cos(\beta) = \frac{(x_{p-1} - x_0)}{f_5}
\]

**Feature \( f_7 \)**

- Sine of ending angle

\[
f_7 = \sin(\beta) = \frac{(x_{p-1} - x_0)}{f_5}
\]
More Definitions (before we continue)

Let $\Delta x_p = x_{p+1} - x_p$ and $\Delta y_p = y_{p+1} - y_p$

Let $\theta_p = \arctan \frac{\Delta x_p \Delta y_{p-1} - \Delta x_{p-1} \Delta y_p}{\Delta x_p \Delta x_{p-1} + \Delta y_p \Delta y_{p-1}}$  Directional angle

Let $\Delta t_p = t_{p+1} - t_p$  Time delta

Feature $f_8$

- Total stroke length

$$f_8 = \sum_{p=0}^{P-2} \sqrt{\Delta x_p^2 + \Delta y_p^2}$$
Feature $f_9$

- Total rotation (from start to end point)
- (not the same as $\beta-\alpha$ – think of spirals)

$$f_9 = \sum_{p=1}^{p-2} \theta_p$$

Feature $f_{10}$

- Absolute rotation
- How much does it move around

$$f_{10} = \sum_{p=1}^{p-2} |\theta_p|$$
**Feature \( f_{11} \)**

- Rotation squared
- How smooth are the turns?
- Measure of sharpness

\[
f_{11} = \sum_{p=1}^{P-2} \theta_p^2
\]

**Feature \( f_{12} \)**

- The maximum speed reached (squared)

\[
f_{12} = \max_{p=0}^{p-2} \frac{\Delta x_p^2 + \Delta y_p^2}{\Delta t_p^2}
\]
Feature $f_{13}$

- Total time of stroke

$$f_{13} = t_{P-1} - t_0$$

Next Class

- Start discussing machine learning algorithms
  - linear classifiers (e.g., Rubine)
  - template matching
  - SVM
  - AdaBoost
  - etc…