

Image-Based Trainable Symbol Recognizer

Devin Smith
Christine Alvarado
Sarah Harris

Department of Computer Science
Harvey Mudd College

July 25, 2007

What is classification?

Many real-world problems arise where we would like to place items into groups based on the inherit properties of each item. We can also think of placing an item into a group as simply labeling the item. The process of labeling an item is known as classification.

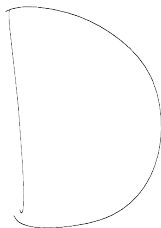
Example

Let us place apples and oranges into their respective buckets!

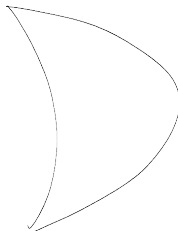


Sketched Digital Logic Symbols

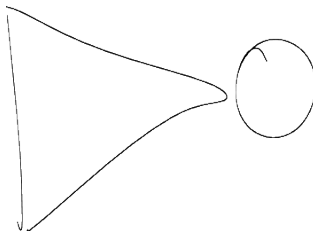
"And"



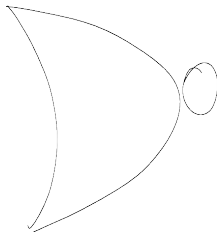
"Or"



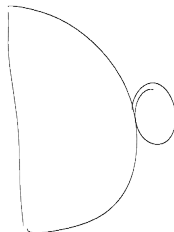
"Not"



"Nor"



"Nand"



Goal

Develop a general sketched-symbol recognizer which returns a table of symbols and their associated probabilities. For example,

Symbol	And	Or	Not	Nor	Nand
Probability	0.20	0.73	0.01	0.05	0.01

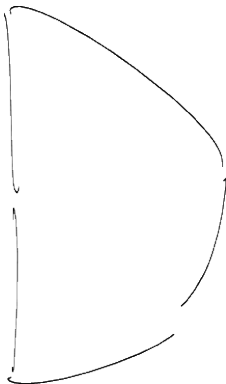
Problems with Raw Sketched-Symbol Data

- Overload of information
- Noise
- Hard to quantify

Potential Problems

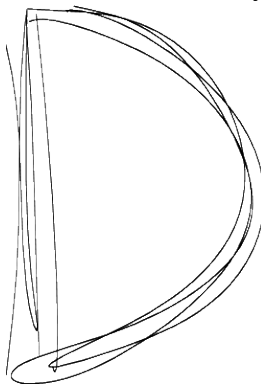
Potential Problems

Understroked Sketched-Symbol



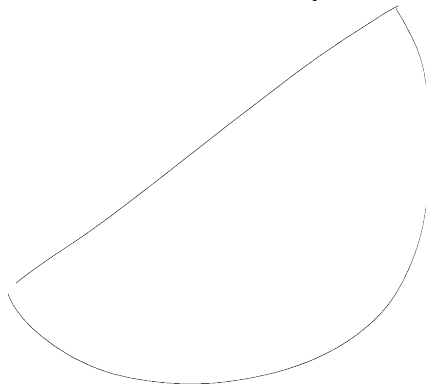
Potential Problems

Overstroked Sketched-Symbol



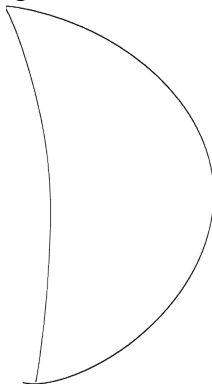
Potential Problems

Rotated Sketched-Symbol



Potential Problems

Ambiguous Sketched-Symbol

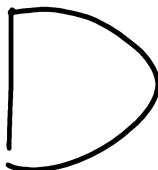


Simplify our Data!

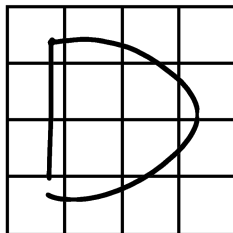
Working with raw sketch-symbol data presents many challenges. One way to approach sketched-symbol recognition is to transform the data into a visual problem.

Raw Sketched-Symbol to Image

Raw Sketched-Symbol to Image



Raw Sketched-Symbol to Image



Raw Sketched-Symbol to Image

1.0	1.0	1.0	0.0
1.0	0.0	1.0	1.0
1.0	0.0	1.0	1.0
1.0	1.0	1.0	0.0

More Information from the Image

As we will see later, it is important that we can access a “minimum-distance-to-pixel” metric for every point (x, y) in an image. Thus, we will also need to create what is known as a distance transform.

Image to Distance Transform

Image

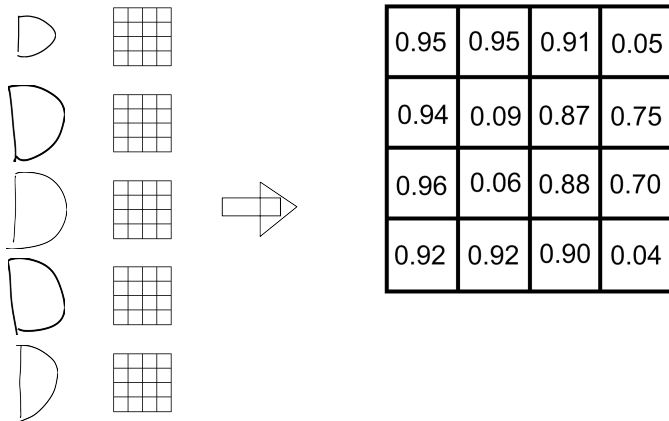
1.0			
			1.0

Distance Transform

0.0	1.0	2.0	3.0
1.0	1.4	2.2	2.0
2.0	2.2	1.4	1.0
3.0	2.0	1.0	0.0

Sketched-Symbols to an Average Image

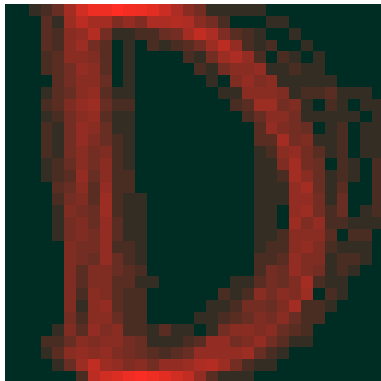
Sketched-Symbols to an Average Image



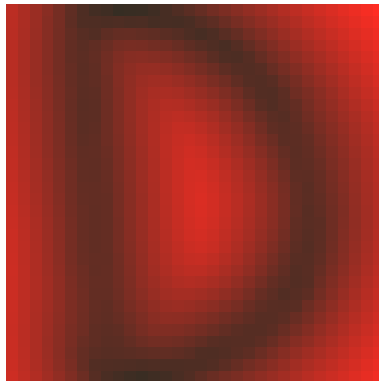
Real Average Images

And

Average Image



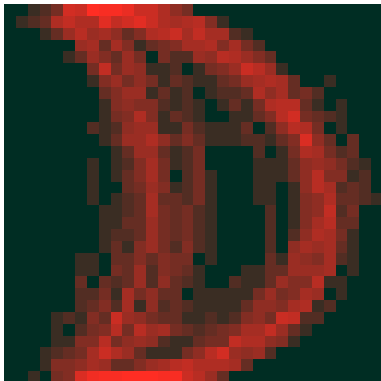
Average Distance Transform



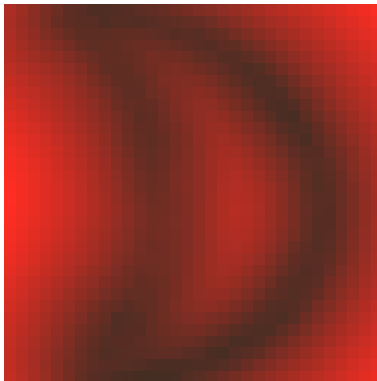
Real Average Images

Or

Average Image

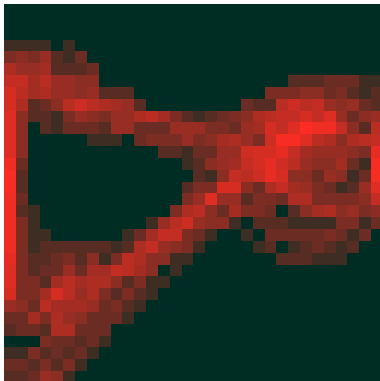


Average Distance Transform



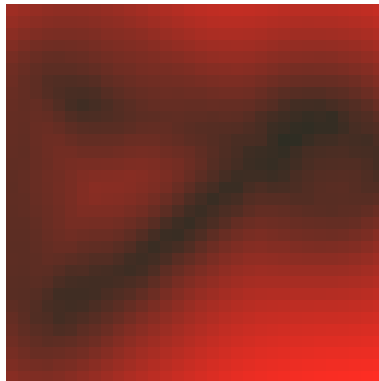
Real Average Images

Average Image



Not

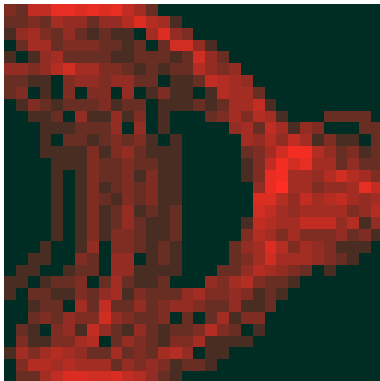
Average Distance Transform



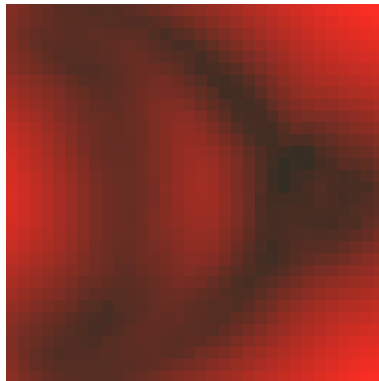
Real Average Images

Nor

Average Image



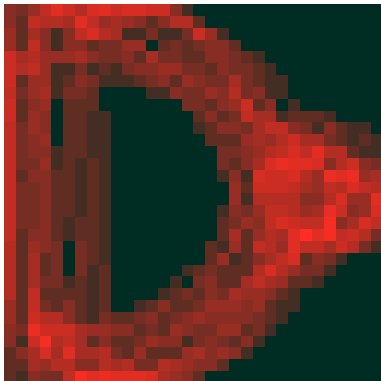
Average Distance Transform



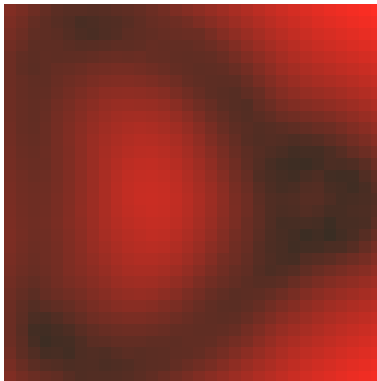
Real Average Images

Nand

Average Image



Average Distance Transform

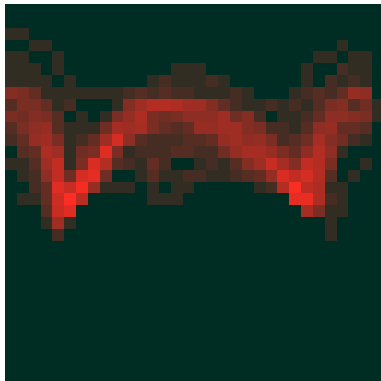


Raw Sketched-Symbol to Polar Image

We also create average polar images, and average polar distance transforms. This helps our sketched-symbol recognizer deal with rotated sketched-symbols, as we will see later.

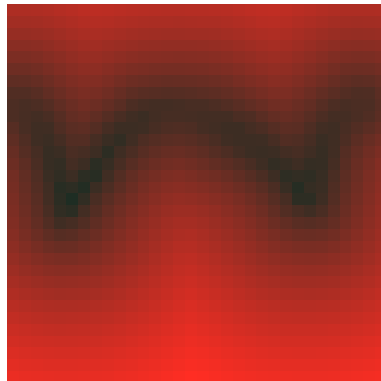
Real Average Polar Images

Average Polar Image



And

Average Polar Distance Transform



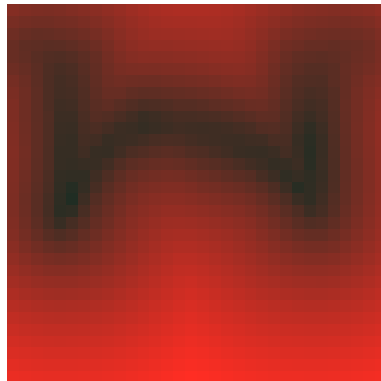
Real Average Polar Images

Average Polar Image



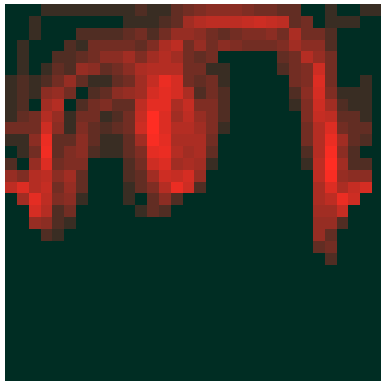
Or

Average Polar Distance Transform



Real Average Polar Images

Average Polar Image



Not

Average Polar Distance Transform



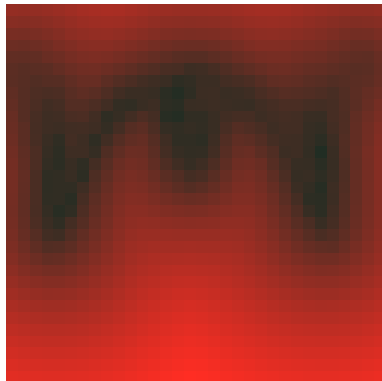
Real Average Polar Images

Average Polar Image



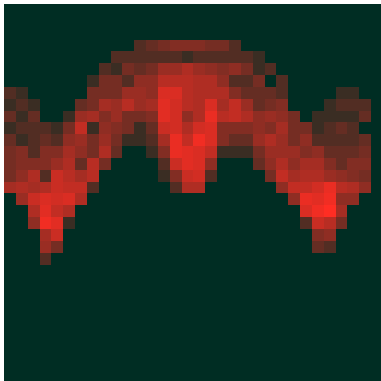
Nor

Average Polar Distance Transform



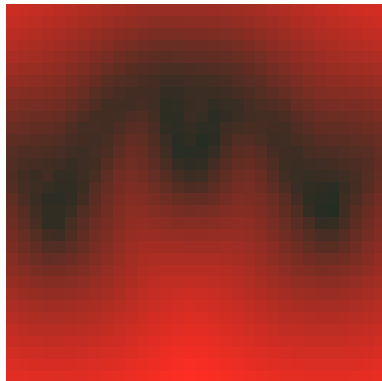
Real Average Polar Images

Average Polar Image



Nand

Average Polar Distance Transform



Rotating a Sketched-Symbol to Match a Definition

- Compute sketched-symbol's polar image around the weighted center.

Rotating a Sketched-Symbol to Match a Definition

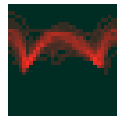
- Compute sketched-symbol's polar image around the weighted center.
- Translate the polar image along the theta axis to match each the definition.

Rotating a Sketched-Symbol to Match a Definition

- Compute sketched-symbol's polar image around the weighted center.
- Translate the polar image along the theta axis to match each the definition.
- Rotate the initial sketched-symbol so that the orientation best matches that of the definiton.

Illustrated Rotation

Sketched-Symbol Image Polar Image And Definition



Illustrated Rotation

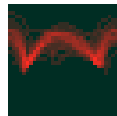
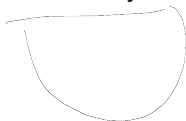
Sketched-Symbol

Image

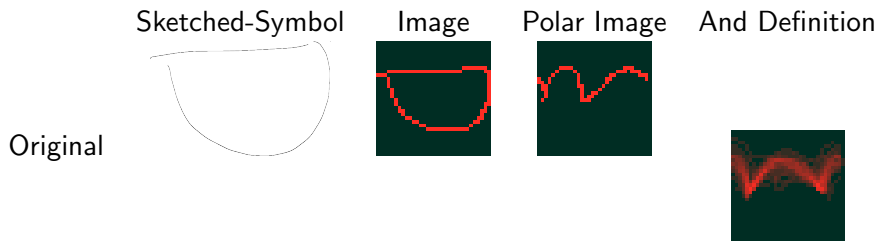
Polar Image

And Definition

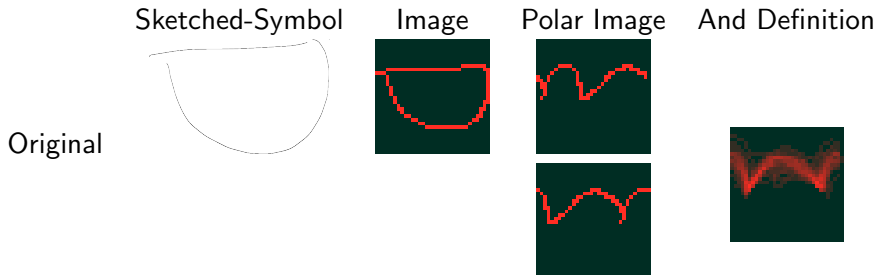
Original



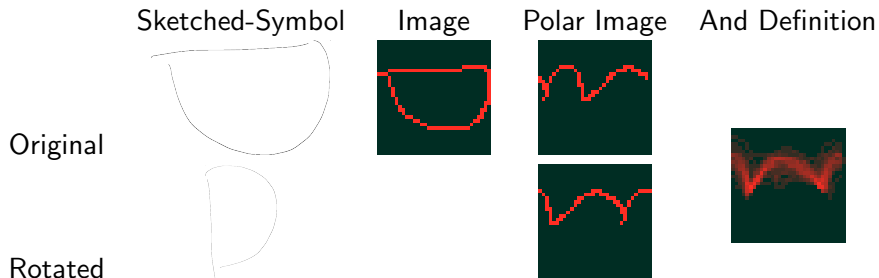
Illustrated Rotation



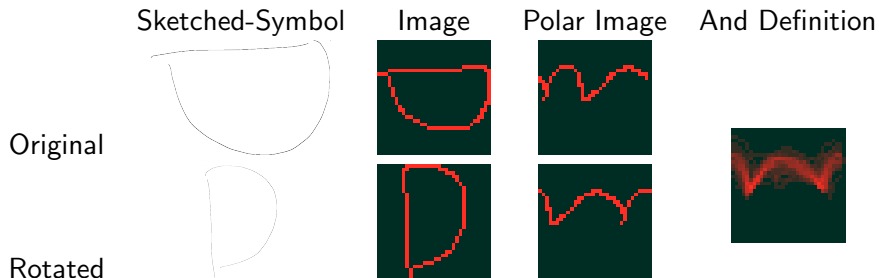
Illustrated Rotation



Illustrated Rotation



Illustrated Rotation



Distance Metrics for Comparing Two Images

- Hausdorff Distance

Distance Metrics for Comparing Two Images

- Hausdorff Distance
- Modified Hausdorff Distance

Distance Metrics for Comparing Two Images

- Hausdorff Distance
- Modified Hausdorff Distance
- Tanimoto Coefficient

Distance Metrics for Comparing Two Images

- Hausdorff Distance
- Modified Hausdorff Distance
- Tanimoto Coefficient
- Yule Coefficient

Hausdorff

The Hausdorff distance between two binary images A and B is $H(A, B) = \max(h(A, B), h(B, A))$, where $h(A, B) = \max_{a \in A}(\min_{b \in B} \|a - b\|)$. $h(A, B)$ is the directed Hausdorff distance. Intuitively, every point in A is at most a distance $h(A, B)$ away from some point in B .

Modified Hausdorff

$$H_{mod}(A, B) = \max(h_{mod}(A, B), h_{mod}(B, A)).$$

$h_{mod}(A, B) = \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \|a - b\|$, where N_a is the number of points in A .

Tanimoto

$T_{sc}(A, B) = \alpha T(A, B) + (1 - \alpha) T^C(A, B)$, where
 $T(A, B) = \frac{n_{ab}}{n_a + n_b - n_{ab}}$, $T^C(A, B) = \frac{n_{00}}{n_a + n_b - 2n_{ab} + n_{00}}$, and
 $\alpha = 0.75 - 0.25(n_a + n_b)/(2N)$, where n_a is the number of points in A , n_b is the number of points in B , n_{ab} is the number of overlapping points in A and B , n_{00} is the number of overlapping white-points in A and B , and N is the number of points in an image.

Yule

$$Y(A, B) = \frac{n_{ab}n_{00} - (n_a - n_{ab})(n_b - n_{ab})}{n_{ab}n_{00} + (n_a - n_{ab})(n_b - n_{ab})}$$