An Efficient Graph-Based Symbol Recognizer

WeeSan Lee <weesan@cs.ucr.edu>
Levent Burak Kara <lkara@andrew.cmu.edu>
Thomas F. Stahovich <stahov@engr.ucr.edu>

Goals

• Hand-drawn symbol recognizer
• Insensitive to:
  - Uniform / non-uniform scaling
  - Orientation
  - Drawing order
• Efficient
• Easily trainable
Approach

- Representation:
  - Attributed Relational Graph
  - Geometry and topology of a symbol

- Recognition:
  - Approximate graph matching
Recognition

Unknown symbol

Definition symbols
Recognition

Unknown symbol

Definition symbols
Recognition

Unknown symbol

Definition symbols

Arc 0

Line 1

Line 2

Line 3

Arc 0

Line 0

Line 1

Line 2

Line 3
Roadmap

- Representation
- Measuring Similarity
- Graph matching
- User Study
- Conclusions
Representation

Ideal Square

Line 0
R=25%
I=1
A=90°
L=(100%, 0%)

Line 1
R=25%
I=0
A=0°
L=N/A

Line 2
R=25%
I=1
A=90°
L=(100%, 0%)

Line 3
R=25%
I=1
A=90°
L=(100%, 0%)
Representation

Ideal Square

Relative length → scale independence
Large $\sigma$ → insensitive to non-uniform scaling
Representation

Ideal Square

Intersection Angle:
Defined for all pairs of lines
Representation

Ideal Square

I=1
A=90°
L=(100%, 0%)

Intersection Angle:
Defined for all pairs of lines
Intersection Angle: Defined for all pairs of lines

Representation

Ideal Square

Line 0
- I=1
- A=90°
- L=(100%, 0%)
- R=25%

Line 1
- I=0
- A=0°
- L=N/A
- R=25%

Line 2
- I=1
- A=90°
- L=(100%, 0%)
- R=25%

Line 3
- I=1
- A=90°
- L=(100%, 0%)
- R=25%
Representation

• Definition
  - “Average graph”
  - Statistical model $\rightarrow \mu, \sigma$

• Segmentation
  - Pen strokes $\rightarrow$ primitives
    • Lines and arcs
  - Speed-based segmenter
Roadmap

✓ Representation
  • Measuring Similarity
  • Graph matching
  • User Study
  • Conclusions
Measuring Similarity

Unknown symbol

Definition symbol

Arc 0
Line 1
Line 2
Line 3
Line 4

Arc 0
Line 1
Line 2
Line 3
Line 4
Line 5
Measuring Similarity

\[ \text{Similarity Score} = 1 - \sum_{i=1}^{6} w_i E_i \]

<table>
<thead>
<tr>
<th>Error Metrics ((E_i))</th>
<th>Weight ((w_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(E_1): Primitive count error</td>
<td>20%</td>
</tr>
<tr>
<td>(E_2): Primitive type error</td>
<td>20%</td>
</tr>
<tr>
<td>(E_3): Relative length error</td>
<td>20%</td>
</tr>
<tr>
<td>(E_4): Number of intersections error</td>
<td>15%</td>
</tr>
<tr>
<td>(E_5): Intersection angle error</td>
<td>15%</td>
</tr>
<tr>
<td>(E_6): Intersection location error</td>
<td>10%</td>
</tr>
</tbody>
</table>
Primitive Count Error

\[ E_1 = \frac{\text{difference\_in\_primitive\_count}}{\text{MIN\_primitive\_count}} \]

Example:
\[ E_1 = \frac{1}{3} \]
Primitive Type Error

\[ E_2 = \frac{\text{\# of type mismatches}}{\text{MIN primitive count}} \]

Example:

\[ E_2 = \frac{1}{4} \]
Relative Length Error

\[ E_3 = \frac{\sum \text{Relative\_length\_errors}}{\text{MIN\_primitive\_count}} \]

\[ \text{Relative\_length\_error} = 1 - P(R, \mu, \sigma) \]
Relative Length Error

\[ E_3 = \frac{\sum \text{Relative_length_errors}}{\text{MIN_primitive_count}} \]

Relative_length_error = 1 - \( P(R, \mu, \sigma) \)

Modified Probability Density Function:

\[ P(R, \mu, \sigma) = \exp\left[-\frac{1}{50.0} \cdot \frac{(R - \mu)^4}{\sigma^4}\right] \]
Relative Length Error

\[ E_3 = \frac{\sum \text{Relative_length_errors}}{\text{MIN_primitive_count}} \]

Relative_length_error = 1 − \( P(R, \mu, \sigma) \)

\[ E_3 = (1 - 1) + (1 - 0.73) + (1 - 0.73) + (1 - 0.98) / 4 \]

= 0.14

<table>
<thead>
<tr>
<th>Node</th>
<th>( U_R )</th>
<th>( \mu )</th>
<th>( \sigma )</th>
<th>( P(U_R) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.38</td>
<td>0.40</td>
<td>0.04</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>0.25</td>
<td>0.20</td>
<td>0.05</td>
<td>0.98</td>
</tr>
<tr>
<td>3</td>
<td>0.19</td>
<td>0.20</td>
<td>0.005</td>
<td>0.73</td>
</tr>
<tr>
<td>4</td>
<td>0.18</td>
<td>0.20</td>
<td>0.01</td>
<td>0.73</td>
</tr>
</tbody>
</table>
Number of Intersections Error

\[ E'_4 = \sum \frac{\text{Intersection differences}}{\text{# of pairs of primitives}} \]

\[ E_4 = \text{Squash}(E'_4) \]

**Example:**

\[ E'_4 = \frac{1}{6} \]

\[ = 0.17 \]

\[ E_4 = S(0.17) \]

\[ = 0.0068 \]
Intersection Angle Error

\[ E_5 = \frac{\sum \text{Errors \_in\_angles}}{\# \_of \_angles} \]
Intersection Location Error

\[ E_6 = \frac{\sum \text{Errors in the intersection locations}}{2 \times \text{Number of intersections}} \]
Roadmap

✓ Representation
✓ Measuring Similarity
  • Graph matching
  • User Study
  • Conclusions
Graph Matching

Task: Find best node-pair assignment
Graph Matching

Task: Find best node-pair assignment
Graph Matching

• Approach: 4 approximate algorithms
  - Stochastic Matching
  - Error driven Matching
  - Greedy Matching
  - Sort Matching
Stochastic Matching

Definition Symbol

Unknown Symbol

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Stochastic Matching

Definition Symbol

Unknown Symbol

1 2 3 4

a b c d
Error-driven Matching

Definition Symbol

Unknown Symbol

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td>a</td>
</tr>
</tbody>
</table>

1 → 2 → 3 → 4
Error-driven Matching

Definition Symbol

Unknown Symbol

<table>
<thead>
<tr>
<th>Definition Symbol</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

1 \rightarrow 1
2 \rightarrow 4
3 \rightarrow 2
4 \rightarrow a
b \rightarrow b
c \rightarrow c
d \rightarrow d
Greedy Matching

Definition Symbol

Unknown Symbol

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>b</td>
<td>c</td>
<td>a</td>
<td></td>
</tr>
</tbody>
</table>

1 2 3 4

a b c d
# Greedy Matching

<table>
<thead>
<tr>
<th>Definition Symbol</th>
<th>Unknown Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>b</td>
<td>c</td>
<td>a</td>
</tr>
</tbody>
</table>
Greedy Matching

Definition Symbol

Unknown Symbol

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td></td>
<td></td>
<td></td>
<td>a</td>
</tr>
<tr>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Greedy Matching

Definition Symbol

1
2 4
3

Unknown Symbol

a
b
d

Definition Symbol

1 2 3 4

Unknown Symbol

a b c d
Sort Matching

Definition Symbol

1
2 4
3

Unknown Symbol

a
b
d
c

Definition Symbol

1 2 3 4

Unknown Symbol

d b c a
Sort Matching

Definition Symbol

1

2 4

3

Unknown Symbol

a

b
d

c

Unknown Symbol

3 2 1 4

c b a d
Roadmap

✓ Representation
✓ Measuring Similarity
✓ Graph matching
  • User Study
  • Conclusions
User Study

• 23 classes of mechanical symbols
• 9 participants
• 15 examples per symbol
Results

- On a P4 3.2G & 1G RAM machine
Conclusion

• Developed graph-based symbol recognizer
  - Insensitive to scale, orientation

• Stochastic Matching
  - Most accurate, most expensive

• Greedy Matching
  - Good compromise

• Sort Matching
  - Very fast, good for low-power devices
Q&A

Thank You!