Gestures as Point Clouds: A $P$ Recognizer for User Interface Prototypes

Radu-Daniel Vatavu (University Stefan cel Mare of Suceava, vatavu@eed.usv.ro)
Lisa Anthony (UMBC Information Systems, lanthony@umbc.edu)
Jacob O. Wobbrock (Information School | DUB Group, University of Washington, wobbrock@uw.edu)

ICMI 2012
Pen and Finger Gesture Input

- My Word Coach (DS)
- Obenkyo (Android)
- Mr. Spiff’s Revenge (PC)
$\text{-Family of Gesture Recognizers}$

$\text{-family of recognizers (}$N$, $P$) are simple, accurate multistroke recognizers (built on $1$)

- $1$ is a simple, accurate unistroke recognizer
- “one dollar” = cheap, fast, easy
- $N =$ multistrokes (e.g., $n$ strokes) *
- $P =$ point clouds (e.g., $p$ points)

$-Family are Template Matchers

Template matcher: point per point correspondence
$N$ Multistroke Recognizer

Multi-stroke representation of an “x”
$\textit{N} \text{ Multistroke Recognizer}

Multi-stroke permutations of an “x”

the 8 ways to write an ‘x’ gesture  $\textit{N}$’s internal representation of an ‘x’
**Gesture Corpus Tested**

Mixed Multistroke Corpus (MMG) ▲
- 20 able-bodied adults
- Gesture and UI oriented symbols (1-3 strokes)

$N 98\%$ accurate with 8 training examples per symbol

Costly to Represent Permutations

$N$ experiences exponential growth in number of permutations to represent.

Number of Strokes in the Gesture

Number of Permutations (logarithmic scale)
$P$ Point Cloud Recognizer

$P$ abstracts execution details: number of strokes, stroke order, and stroke direction.
Hungarian algorithm * solves assignment problem

Hungarian outperforms template-matching competitors on multistroke recognition for both user-dependent and user-independent cases.
Evaluating the Hungarian Algorithm

Feasible labelings

Equality graphs

Alternating paths

Alternating trees

But, Hungarian is too costly in terms of execution time, and complexity.
Approximate ideal min-cost matching with greedy techniques
Selecting the Best Approximation

Greedy5 with $\varepsilon$ of 0.50 is the best performer, even outperforms Hungarian.
$P$ Matching Demo: Unistroke
$P$ Matching Demo: Multistroke
Performance of $P$

- $P$ is **99% accurate** with **5 training examples** per symbol (user-dependent)
- $N$ was **98% accurate** with **8 training examples** per symbol
- $P$ is also **99% accurate** with **10 participants' data** (user-independent)
Performance of $P$

- $P$ is **99% accurate** with **5 training examples** per symbol (user-dependent)
- $N$ was **98% accurate** with **8 training examples** per symbol
- $P$ is also **99% accurate** with **10 participants' data** (user-independent)

$P$ now state of the art $-$ family recognizer for multistrokes.
Performance of $P$

- $P$ is **99% accurate** with **5 training examples** per symbol (user-dependent)
- $N$ was **98% accurate** with **8 training examples** per symbol
- $P$ is also **99% accurate** with **10 participants' data** (user-independent)

$P$ now state of the art $-$ family recognizer for multistrokes.
Performance of $P$

- $P$ is 99% accurate with 5 training examples per symbol (user-dependent)
- $N$ was 98% accurate with 8 training examples per symbol
- $P$ is also 99% accurate with 10 participants' data (user-independent)

$P$ now state of the art $-$ family recognizer for multistrokes.
Limitations of $P$

$P$ is not rotation-invariant

$P$ is not direction-invariant
$-Family is Open Source

Reference implementations available for $1, $N, $P (pseudocode, JavaScript, C#)

Supports independent app development
- Rapid pick-up by community ($1, $N)
- AlphaCount iPhone app ($N)
- iOS port of the recognizer ($1, $N)

$P$ has ~70 lines of pseudo-code
- 50% reuse $1$ pseudo-code
$\$-Family Cheat Sheet

<table>
<thead>
<tr>
<th>Criteria</th>
<th>$$1</th>
<th>Protractor</th>
<th>$$N</th>
<th>$$P</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gesture types</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>recognizes single strokes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>recognizes multistrokes</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>is scale-invariant</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>is rotation-invariant</td>
<td>✓</td>
<td>can be</td>
<td>can be</td>
<td>×</td>
</tr>
<tr>
<td>is direction-invariant</td>
<td>×</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Performance</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>user-dependent accuracy</td>
<td>99.5% / ×</td>
<td>99.4% / ×</td>
<td>98.0% / 97.7%</td>
<td>99.3% / 98.4%</td>
</tr>
<tr>
<td>(single/multi-strokes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>user-independent accuracy</td>
<td>97.1% / ×</td>
<td>95.9% / ×</td>
<td>95.2% / 96.4%</td>
<td>96.6% / 98.0%</td>
</tr>
<tr>
<td>(single/multi-strokes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>algorithmic complexity</td>
<td>$O(n \cdot T \cdot R)$</td>
<td>$O(n \cdot T)$</td>
<td>$O(n \cdot S! \cdot 2^S \cdot T)$</td>
<td>$O(n^{2.5} \cdot T)$</td>
</tr>
<tr>
<td>memory to store the training</td>
<td>$O(n \cdot T)$</td>
<td>$O(n \cdot T)$</td>
<td>$O(n \cdot S! \cdot 2^S \cdot T)$</td>
<td>$O(n \cdot T)$</td>
</tr>
<tr>
<td>set</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Code writing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>needs rotation search (GSS)</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>needs writing filters to</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>speed-up matching</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>approx. # of lines of code</td>
<td>100</td>
<td>50</td>
<td>200</td>
<td>70</td>
</tr>
</tbody>
</table>

---

1. $n$ is the number of sampled points; $T$ = the number of training samples per gesture type; $R$ = the number of iterations required by the Golden Section Search (GSS) algorithm used by $\$1$ (experimentally set to 10 [23]); $S$ = the number of strokes in a multistroke; $S! \cdot 2^S$ = the number of different permutations of stroke ordering and direction needed by $\$N$ [2].

2. A $\mathcal{G}$ symbol means more coding is required (e.g. GSS, filters, or simply more lines of code).

Cheat Sheet for $\$-family helps app developers choose best approach for app context.
Summary

✓ $\text{-family of recognizers has a new member: } P.

✓ $P$ is highly accurate.

✓ $P$ uses simple geometric principles to match gestures.

✓ $P$ is invariant to (1) number of strokes, (2) stroke type, (3) stroke direction, and (4) stroke ordering.

✓ $P$ saves both storage space and execution time for multistroke recognition, compared to $N$.

✓ $\text{-family of recognizers continues tradition of easy-to-use, open-source, accessible gesture recognition.}$
Questions?

Contact:
- Radu-Daniel Vatavu, PhD
  • vatavu@eed.usv.ro
- Lisa Anthony, PhD
  • lanthony@umbc.edu
- Jacob O. Wobbrock, PhD
  • wobbrock@uw.edu

Online demo:
References


