Pen-Based Gestural User Interfaces

Lecture #6: Gestures
Joseph J. LaViola Jr.
Fall 2016

What is a Pen Gesture?

- Simple ink stroke or strokes to convey an idea
  - fast to perform
  - easy to remember
- Typically disappear after they are recognized
- Supports in-band interaction
**Gesture Types**

- Single stroke
- Multi-stroke
  - compound gestures
  - punctuated gestures
- Trade-off in recognition between single and multiple stroke gestures
- Used in
  - modeling
  - command languages
  - invoking interface widgets

**Gestures in Modeling**

- Used in 2D/3D object modeling
- Distinction between sketch-based modeling and gestures in modeling
- Used to
  - create geometry
  - manipulate geometry
  - guidance for computational algorithms

[www-ui.is.u-tokyo.ac.jp/~takeo/research/teddy/teddy.htm](www-ui.is.u-tokyo.ac.jp/~takeo/research/teddy/teddy.htm)
SKETCH

- Seminal work by Zeleznik et al. (1996)
- Conceptual modeling
- Uses simple lines and curves to build geometric primitives
  - cubes, cylinders, pyramids, etc...
- No machine learning-based recognition used
  - simple FSA
- Does make use of modifier keys

SKETCH Gesture Set (1)

- Dragging objects
- Scaling objects
- Copying objects
- Frehand drawing
- Scaling along an axis
SKETCH Gesture Set (2)

Teddy

- Seminal work by Igarashi et al. (1999)
  - did for organic modeling what Zeleznik et al. did for primitive-based modeling
- Supports
  - Object creation
  - Cutting
  - Extrusion
  - Smoothing
- No machine learning used
  - Simple FSA and geometric construction techniques
Surface/Mesh Editing

- Fine line between sketching and gestures
- Uses simple gesture as input to a surface editing algorithm
- This type of approach has been used for image processing as well
  - see work of Salesin

Gestures as Command Languages

- Gestural commands
  - replace traditional WIMP user interfaces
  - also used to invoke interface widgets
- Notion of in-band gestures
  - invoking commands and operations at the location of interaction
  - contrasts with having to move to top/side of the screen to press a button or find a menu item
- Used in
  - entering text
  - text editing
  - note taking
  - mathematical apps
  - etc…

Nealen et al. (2005)
Graffiti

- Language for entering text
- Maps to keyboard
- Used with Palm Pilot
- Single stroke language
  - Has prefix for some symbols
- Takes a while to learn

Text Editing

- Example of a gesture set taken from real world and developed for pen computers
- Natural connection between pencil and paper and computer

www.jumpingminds.com
MathPad²

- Simple gesture set for
  - invoking operations
  - manipulating ink
- Uses notion of punctuated gestures
  - multi-stroke (gesture + punctuation)
  - makes use of context
- Why?
  - reduce number of gestures
  - overload appropriate gestures
  - reduce conflicts

MathPad² Gesture Set (1)

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x + y^2$</td>
<td>$x + y^2$</td>
<td>Lasso and tap to recognize an expression</td>
</tr>
<tr>
<td>$x + y$</td>
<td>$x + y$</td>
<td>Scribble and tap to delete ink</td>
</tr>
<tr>
<td>$\frac{x}{a + b}$</td>
<td></td>
<td>Creates a graph, line starts in recognized math, no cusps or intersections</td>
</tr>
<tr>
<td>$\mathcal{L}$</td>
<td></td>
<td>Line through math and click on drawing makes association, Release makes rotation point</td>
</tr>
<tr>
<td>$y + 2 = 0$</td>
<td>$y = -2$</td>
<td>Solves equation, includes simultaneous and ordinary differential equations</td>
</tr>
<tr>
<td>$\int x^2 , dx$</td>
<td>$\frac{x^3}{3}$</td>
<td>Evaluate an expression, includes integrals, derivatives, summations, etc.</td>
</tr>
</tbody>
</table>
MathPad\textsuperscript{2} Gesture Set (2)

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_x = 3$</td>
<td>$P_x = 3$</td>
<td>Makes implicit association using label family $P$</td>
</tr>
<tr>
<td>$P_x = 3$</td>
<td>$P_x = 3$</td>
<td>Makes implicit association with explicit tap on object</td>
</tr>
<tr>
<td>$\alpha = 1.57$</td>
<td>$\alpha = 1.57$</td>
<td>Implicit angle association and rectification</td>
</tr>
<tr>
<td>$\text{ee} \rightarrow \text{ee}$</td>
<td>$\text{ee} \rightarrow \text{ee}$</td>
<td>Nail two drawing elements by small circle and tap</td>
</tr>
<tr>
<td>$y = x^4$</td>
<td>$y = x^4$</td>
<td>Lasso and drag symbol to change position</td>
</tr>
</tbody>
</table>

Fluid Inking (Zeleznik and Miller 2006)

- Approach to augment free-form inking with gestures (collection of techniques)
- Guidelines
  - hardware impartiality (no buttons)
  - performability (minimal targeting)
  - extensibility
  - discoverability
- Uses
  - terminal punctuation
  - flicks

<table>
<thead>
<tr>
<th>Gesture Class</th>
<th>Context</th>
<th>Gesture</th>
<th>Terminal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mnemonic flick</td>
<td>Flick ()</td>
<td>letter</td>
<td>tap or pause</td>
</tr>
<tr>
<td>Self-contained mnemonic mimic</td>
<td>lasso ($\subset$), scalable ($\leftrightarrow$), or crop ($\square$)</td>
<td>letter or scalable ($\leftrightarrow$)</td>
<td>tap or pause</td>
</tr>
<tr>
<td>Lasso ($\subset$)</td>
<td>stroke hook ()</td>
<td>tap or pause</td>
<td></td>
</tr>
</tbody>
</table>

Example:
- $x$ saves the line
- $\text{at}$ deletes ink under it
- $\text{c}$ copies ink contained in the lasso
- $\text{d}$ applies menu option to lasso contents

Gesture in top row, regular ink in bottom row.

Flick, mnemonic flick, ink
Recognizing Gestures

- FSA’s and simple primitive operators
  - conditionals and saving state from one event trigger to another
  - Operators can be features
    - same features used in machine learning!
    - features must be excellent discriminators
- Machine learning techniques
  - SVMs, K-nearest neighbor, AdaBoost
  - more on this soon!

Anatomy of a Gesture

Detecting and equal sign

Note that as the gesture set increases the more tests you typically have to employ to avoid conflicts.

Input: Strokes $s_{-1}$ and $s_{-2}$, a bounding box threshold $\epsilon_{box}$, and a line difference threshold $\epsilon_{dag}$.
Output: True or false.

$\text{DETECT_EQUAL_SIG}(s_{-1}, s_{-2}, \epsilon_{box}, \epsilon_{dag})$

1. $P \leftarrow \text{Points}(s_{-1})$
2. $Q \leftarrow \text{Points}(s_{-2})$
3. $b_1 \leftarrow \text{Bounding Box}(s_{-1})$
4. $b_2 \leftarrow \text{Bounding Box}(s_{-2})$
5. $\text{slen}_1 \leftarrow \sum_{i=2}^{n} ||P_i - P_{i-1}||$
6. $\text{slen}_2 \leftarrow \sum_{i=2}^{n} ||Q_i - Q_{i-1}||$
7. if \[ \text{slen}_1 > \epsilon_{box}(\text{Width}(b_1)^2 + \text{Height}(b_1)^2) \text{ or } \text{slen}_2 > \epsilon_{box}(\text{Width}(b_2)^2 + \text{Height}(b_2)^2) \]
   then \[ \text{return false} \]
8. if Width($b_1$) < Height($b_1$) or Width($b_2$) < Height($b_2$)
   then \[ \text{return false} \]
9. if Width($b_1$) < Height($b_1$) or Width($b_2$) < Height($b_2$)
   then \[ \text{return false} \]
10. else
11. \[ \text{diff}_1 = |X(P_1) - X(Q_1)| \]
12. \[ \text{diff}_2 = |X(P_2) - X(Q_2)| \]
13. if LineOverlap($P_1, P_2, Q_1, Q_2$) and $\text{diff}_1 < \epsilon_{dag} \text{ and } \text{diff}_2 < \epsilon_{dag}$
   then \[ \text{return true} \]
14. else
15. \[ \text{return false} \]
Learning Gestures

- How many gestures is too many?

- Learning strategies
  - Simple tutorials/manuals
  - Gesture practice tools
  - Color coding (useful for multi-stroke gestures)
  - Showing gestures through animations

- Techniques not proven – open research area

Music NotePad gestural tutorial system

Multi-Touch Gestures

http://www.flickr.com/photos/ideum/4380417382/
Readings