Pen-Based Gestural User Interfaces

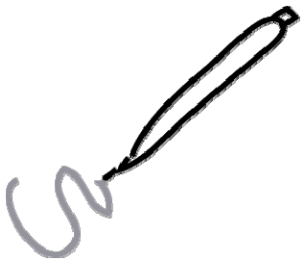
Lecture #6: Gestures

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

Single stroke gesture

Multi-stroke gesture

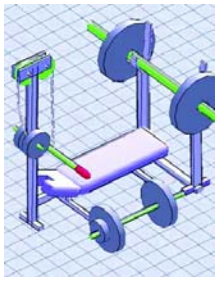
Gestures in Modeling

- 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

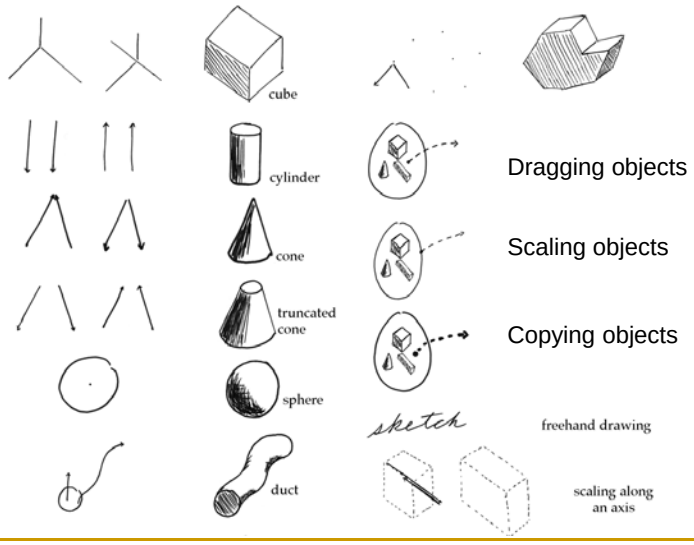
www-ui.is.s.u-tokyo.ac.jp/~takeo/research/teddy/teddy.htm

SKETCH

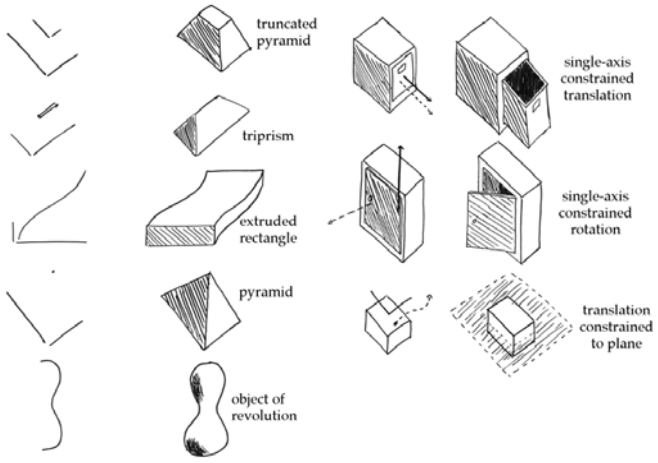
- Seminal work by Zeleznik et al. (1996)
- Conceptual modeling
- Uses simple lines and curves to build geometric primitives
  - cubes, cylinders. pyramids, etc...
- No machine learning-based recognition used
  - simple FSA
- Does make use of modifier keys



SKETCH Gesture Set (1)



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

Smooth Teddy

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

Nealen et al. (2005)

Gestures as Command Languages

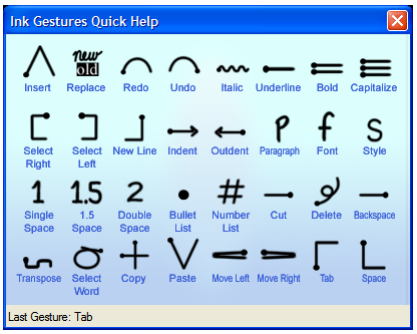
- Gestural commands
  - replace traditional WIMP user interfaces
  - also used to invoke interface widgets
- Notion of in-band gestures
  - invoking commands and operations at the location of interaction
  - contrasts with having to move to top/side of the screen to press a button or find a menu item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

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

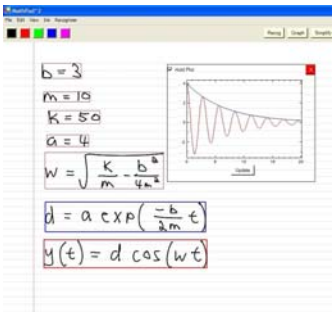
- Example of a gesture set taken from real world and developed for pen computers
- Natural connection between pencil and paper and computer



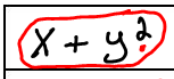
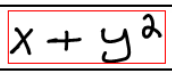
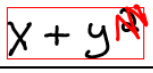
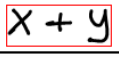
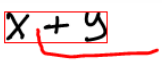
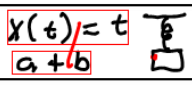
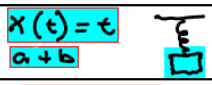
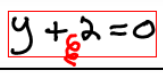
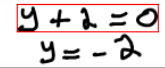
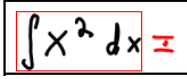
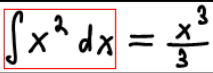
www.jumpingminds.com

MathPad²

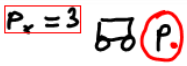
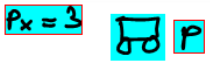
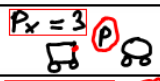
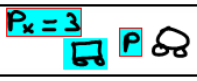
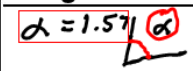
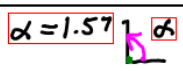
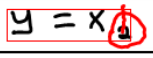
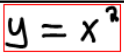
- Simple gesture set for
  - invoking operations
  - manipulating ink
- Uses notion of punctuated gestures
  - multi-stroke (gesture + punctuation)
  - makes use of context
- Why?
  - reduce number of gestures
  - overload appropriate gestures
  - reduce conflicts



MathPad² Gesture Set (1)

Gesture	Result	Description
		Lasso and tap to recognize an expression
		Scribble and tap to delete ink
		Creates a graph, line starts in recognized math, no cusps or intersections
		Line through math and click on drawing makes association, Release makes rotation point
		Solves equation, includes simultaneous and ordinary differential equations
		Evaluate an expression, includes intergrals,derivatives, summations, etc.

MathPad² Gesture Set (2)

Gesture	Result	Description
		Makes implicit association using label family 'P'
		Makes implicit association with explcit tap on object
		Implicit angle association and rectification
		Nail two drawing elements by small circle and tap
		Group strokes
		Lasso and drag symbol to change position

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

Gestures in top row, regular ink in bottom row.

Flick, mnemonic flick, ink

Gesture Class	Context	Gesture	Terminal	Example
Mnemonic flick	flick (/)	letter		⚷ saves the file
Punctuated:				
self-contained		lasso (⌓), scribble (≋), or crop (⌊)	tap or pause	≋ deletes ink under it
mnemonic	lasso (⌓)	letter or scribble (≋)	tap or pause	⎘ copies ink contained in the lasso
mimetic	lasso (⌓)	stroke hook (⁄)	tap or pause	⌓ applies NE menu option to lasso contents

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.

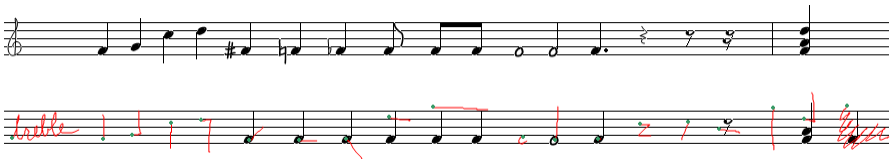
Input: Strokes s_{i-1} and s_{i-2}, a bounding box threshold ϵ_{box}, and a line difference threshold ϵ_{diff}.
Output: True or false.

DetectEqualSign(s_{i-1},s_{i-2},ϵ_{box},ϵ_{diff})

(1) $P \leftarrow Points(s_{i-1})$
(2) $Q \leftarrow Points(s_{i-2})$
(3) $b_1 \leftarrow BoundingBox(s_{i-1})$
(4) $b_2 \leftarrow BoundingBox(s_{i-2})$
(5) $slen_1 \leftarrow \sum_{i=2}^{n} \|P_i - P_{i-1}\|$
(6) $slen_2 \leftarrow \sum_{i=2}^{n} \|Q_i - Q_{i-1}\|$
(7) **if** $slen_1 > \epsilon_{box}\sqrt{Width(b_1)^2 + Height(b_1)^2}$ **or** $slen_2 > \epsilon_{box}\sqrt{Width(b_2)^2 + Height(b_2)^2}$
(8) **return** false
(9) **if** $Width(b_1) < Height(b_1)$ **or** $Width(b_2) < Height(b_2)$
(10) **return** false
(11) $diff_1 = |X(P_1) - X(Q_1)|$
(12) $diff_2 = |X(P_n) - X(Q_n)|$
(13) **if** $LineOverlap(P_1, P_n, Q_1, Q_n)$ **and** $diff_1 < \epsilon_{diff}$ **and** $diff_2 < \epsilon_{diff}$
(14) **return** true
(15) **else**
(16) **return** false

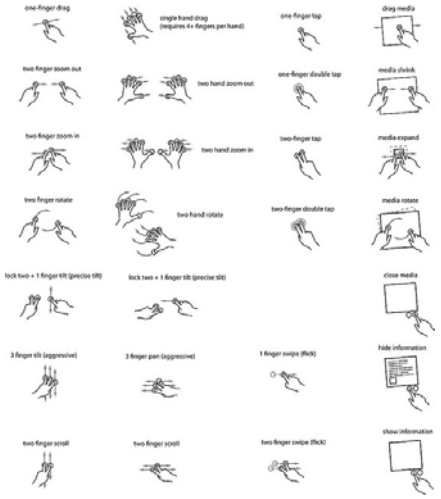
Learning Gestures

- 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

- Zeleznik, R., K. Herndon, and J. Hughes. SKETCH: An Interface for Sketching 3D Scenes. Proceedings of SIGGRAPH'96, ACM Press, 163-170, 1996.
- Igarashi, T., S. Matsuoka, and H. Tanaka. Teddy: A Sketching Interface for 3D Freeform Design. Proceedings of SIGGRAPH'99, ACM Press, 409-416, 1999.
- Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H., and Buxton, B. Pen + Touch = New Tools. In *Proc. UIST 2010 Symposium on User interface Software and Technology*, 27-36, October 2010.
- Zeleznik, R., Bragdon, A., Adeputra, F., and Ko. H. Hands-On Math: A Page-based Multi-touch and Pen Desktop for Technical Work and Problem Solving. In *Proceedings of the 23rd Annual Symposium on User Interface Software and Technology* (UIST 2010), 17-26, October 2010.