



# Handwriting Recognition: The Case for Unistrokes (and how to roll your own recognizer)

I. Scott MacKenzie  
York University

1



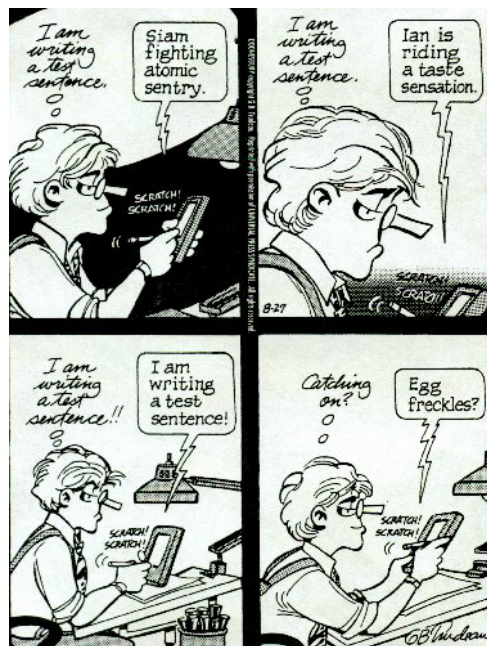
## Plan

- Pen-based computing
- Unistrokes
- Recognition
- Demo

2

## A Pen as an Input Device

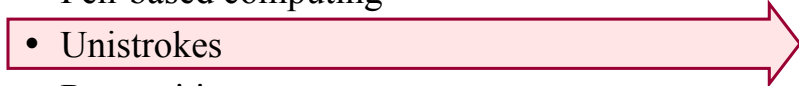
- Advantages
  - considerable existing skill
  - good for annotating, sketching, etc.
  - handwriting is natural
- Disadvantages
  - handwriting recognition is hard!
  - in early products, goods delivered did not live up to promises





## Plan

- Pen-based computing
- Unistrokes
- Recognition
- Demo



5



## Unistrokes

- What is a Unistroke?
  - A single stylus or finger gesture
  - Begins when stylus touches tablet surface
  - Ends when stylus is raised from tablet surface
  - Each “symbol” is one stroke

6

## Unistrokes vs. Graffiti

- Unistrokes
  - Invented by David Goldberg, Xerox PARC <sup>a</sup>
  - Patent held by Xerox (issued Jan. 1997)
  - First presented at *INTERCHI '93*
- Graffiti
  - An ‘example’ of unistrokes
  - Commercial product
  - 3COM *PalmPilot*, HP *Omnigo*, etc.

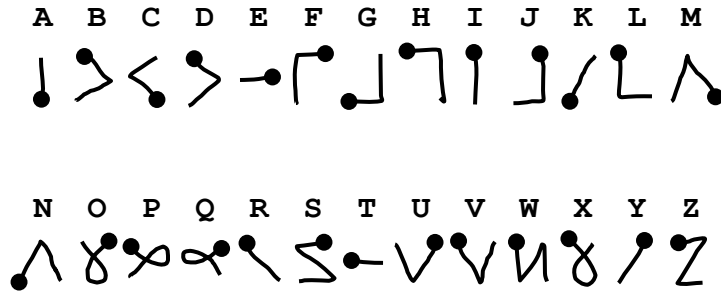
<sup>a</sup> Goldberg, D., & Richardson, C. (1993). Touch-typing with a stylus. *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, 80-87. New York: ACM.

## Palm Pilot



## Goldberg's Unistrokes

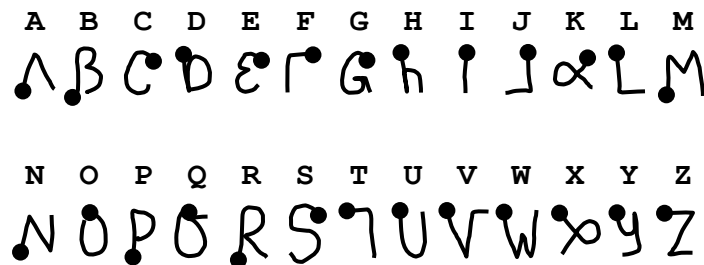
Goal: Fast for experts



9

## Graffiti

Goal: Easy for novices



See...

MacKenzie, I. S., & Zhang, S. (1997). The immediate usability of Graffiti. *Proceedings of Graphics Interface '97*, pp. 129-137. Toronto: Canadian Information Processing Society.

10

## Application



See...

Enns, N., & MacKenzie, I. S. (1998). Touch-based remote control devices. *Extended Abstracts of the CHI '98 Conference on Human Factors in Computing Systems*, 229-230. New York: ACM.

11

## Plan

- Pen-based computing
- Unistrokes
- Recognition
- Demo



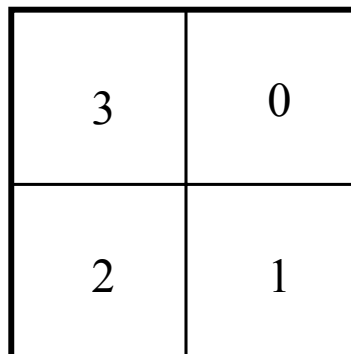
12

## Steps for Recognition

- Input  $x,y$  coordinates for unistroke
- Determine if stroke is a straight line (if so, done!)
- Determine min and max for  $x$  and  $y$
- Normalize points to fit in unit bounding box
- Note: box has four quadrants...

13

## Steps (2)



14

## Steps (3)

- Scan points and gather characteristics of stroke:
  - 1st quadrant
  - 2nd quadrant
  - 2nd last quadrant
  - last quadrant
  - cumulative  $x$  motion (e.g. 'Z' = 3)
  - cumulative  $y$  motion (e.g., 'Z' = 1)

15

## Steps (3)

- starting  $x$  direction (+ve, -ve)
- starting  $y$  direction (+ve, -ve)
- stopping  $x$  direction (+ve, -ve)
- stopping  $y$  direction (+ve, -ve)
- Compare stroke characteristics with dictionary entries
- Find match, done

16

## Dictionary Entries

- Each entry contains
  - symbol (the ‘output’)
  - 1st quadrant
  - 2nd quadrant
  - 2nd last quadrant
  - last quadrant
  - min cumulative  $x$
  - max cumulative  $x$

17

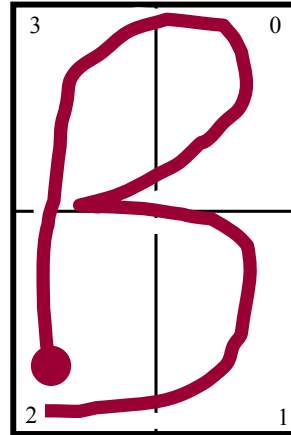
## Dictionary Entries (2)

- min cumulative  $y$
- max cumulative  $y$
- starting  $x$  motion (+ve, -ve, don't care)
- starting  $y$  motion (+ve, -ve, don't care)
- stopping  $x$  motion (+ve,-ve, don't care)
- stopping  $y$  motion (+ve, -ve, don't care)
- Note: can have multiple entries for each symbol (e.g. cw “0”, ccw “0”)

18

## Example

Symbol	B
1st quad	2
2nd quad	3
n-1 quad	1
n quad	2
kx min	2.2
kx max	4
ky min	1.5
ky max	2.5
start x	don't care
start y	+
stop x	-
stop y	don't care



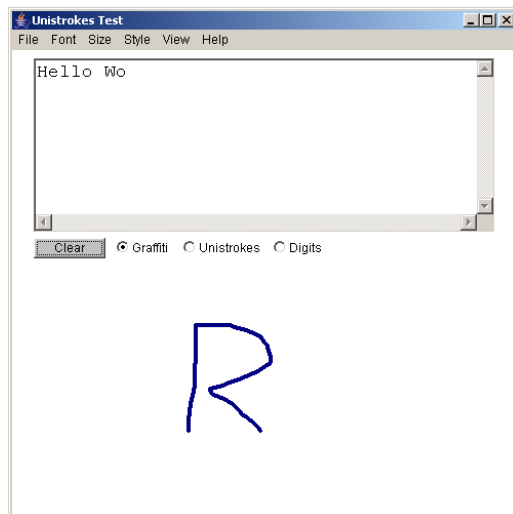
19

## Plan

- Pen-based computing
- Unistrokes
- Recognition
- Demo



20



Thank You