MMI 2: Mobile Human-Computer Interaction

Mobile Input and Output

Prof. Dr. Michael Rohs

michael.rohs@ifi.lmu.de

Mobile Interaction Lab, LMU München

Review

- Was ist ein “information appliance”?
- Was sind die technologischen Grundlagen des „mobile computing“?
- Wer hat das Telefon erfunden?

Preview

- Input and output modalities for mobile devices

- Motor system
- Design space of input devices
- Text input for mobile devices
- Touch screen gestures

- (Display technologies)
- (Haptics and audio)

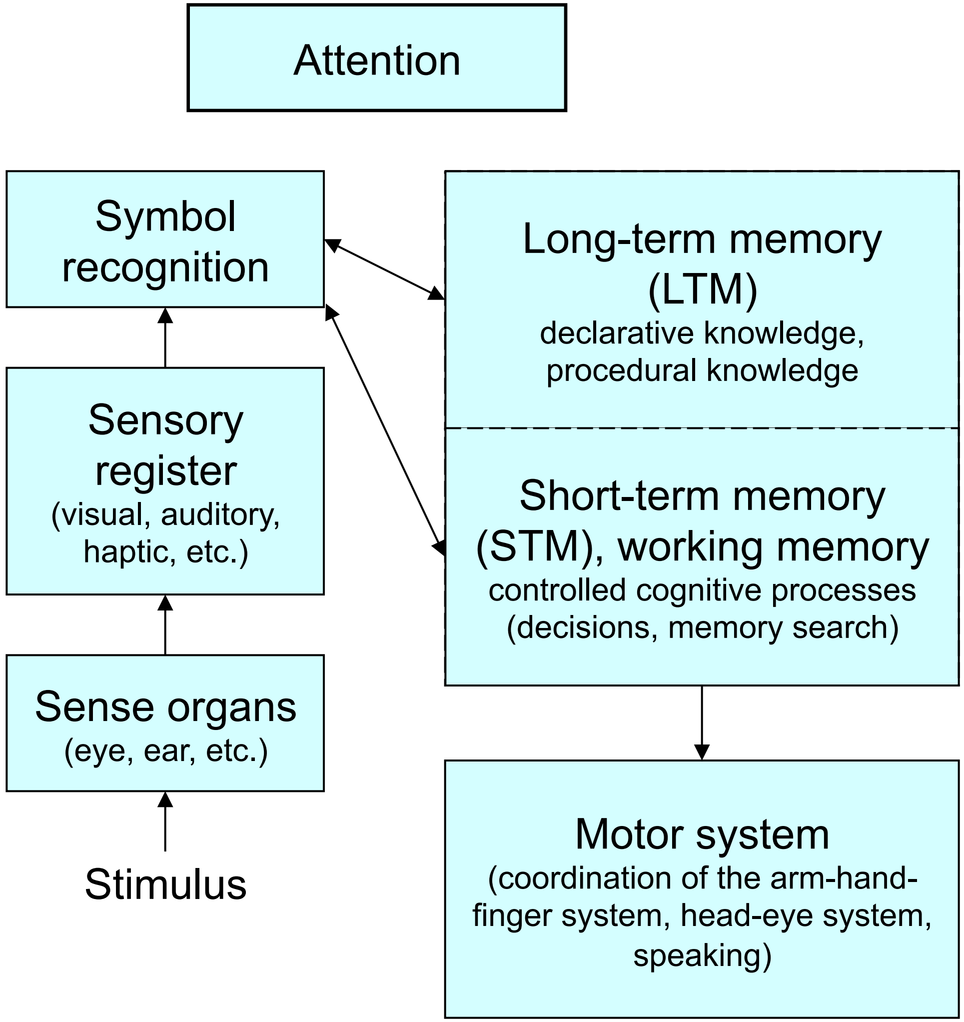
Lectures

#	Date	Topic
1	19.10.2011	Introduction to Mobile Interaction, Mobile Device Platforms
2	26.10.2011	History of Mobile Interaction, Mobile Device Platforms
3	2.11.2011	Mobile Input and Output Technologies, Mobile Device Platforms
4	9.11.2011	Mobile Interaction Design Process
5	16.11.2011	Mobile Communication
6	23.11.2011	Location and Context
7	30.11.2011	Prototyping Mobile Applications
8	7.12.2011	Evaluation of Mobile Applications
9	14.12.2011	Visualization and Interaction Techniques for Small Displays
10	21.12.2011	Mobile Devices and Interactive Surfaces
11	11.1.2012	Camera-Based Mobile Interaction 1
12	18.1.2012	Camera-Based Mobile Interaction 2
13	25.1.2012	Sensor-Based Mobile Interaction 1
14	1.2.2012	Sensor-Based Mobile Interaction 2
15	8.2.2012	Exam

MOTOR SYSTEM

Components of Cognition

- Perception
  - Visual system
  - Auditory system
  - Haptic system
- Action
  - **Motor system**
- Memory
  - Sensory memory
  - Short-term memory / working memory
  - Long-term memory
- Skill acquisition



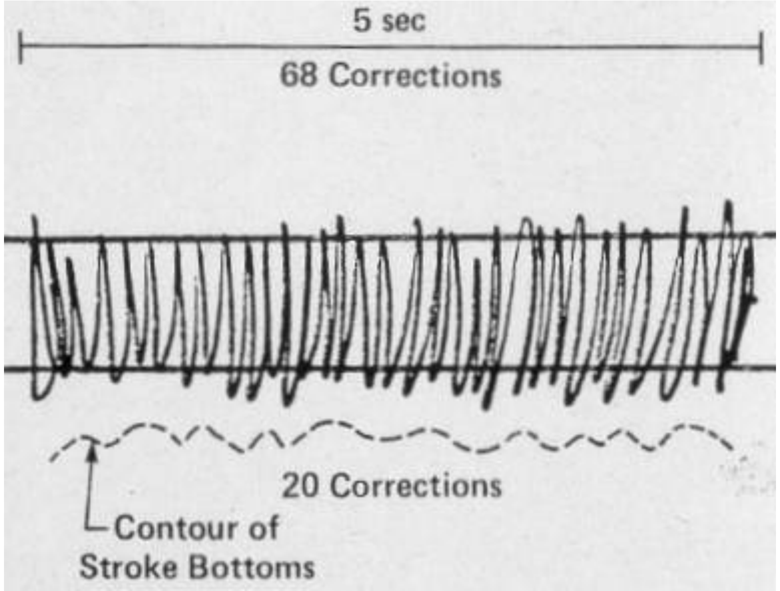
Adapted from: Wandmacher, Software Ergonomie

Motor Control

- Movement affects interaction with computers
  - Example: pressing a button in response to a question
- Movement time depends on age and fitness
- Speed vs. accuracy
  - Higher speed of movement reduces accuracy
  - Depends on skills (e.g. typists with lot of practice are faster and make fewer errors)

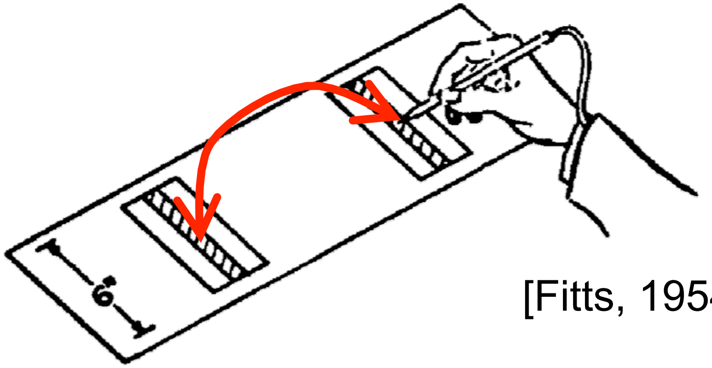
Motor System: Maximum Motor Output Rate

- Movement consists of micromovements of fixed duration
  - τ_M = 70 [30-100] ms
  - Perceptual feedback loop takes longer (240 ms)
- Experiment: Move pen between lines as fast as possible for 5 sec.
- Open loop
  - Without perceptual control
  - 68 pen reversals in 5 sec
  - 74 ms per reversal
- Closed loop
  - Perceptual system controls
  - 20 corrections in 5 sec
  - 250 ms per correction



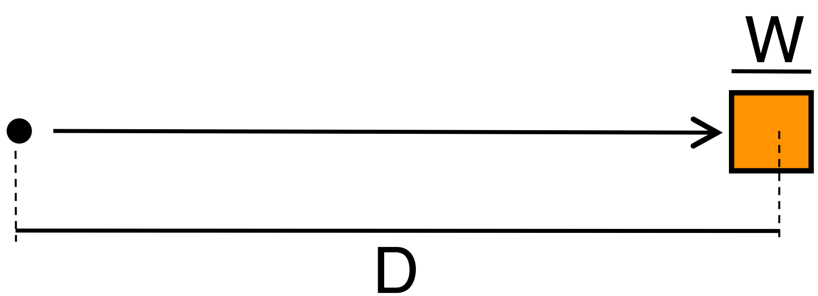
Motor System: Fitts' Law

- Directed movement as an information processing task
  - Not limited by muscles, but by ability to process sensory input
- Index of difficulty (ID)
  - $ID = \log_2(D / W + 1)$
  - $MT = a + b * ID$



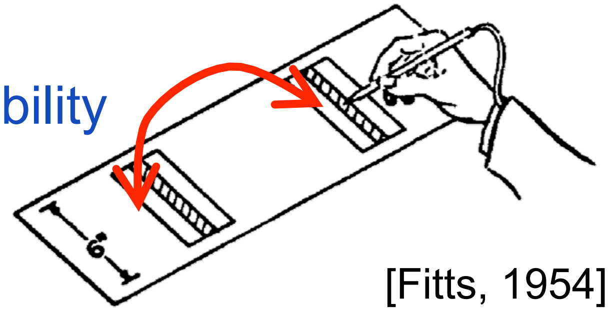
[Fitts, 1954]

- Paul Fitts' original experiments
  - Tapping, disk, and pin transfer
  - Influenced by Shannon's information theory $C = B \log_2((S+N) / N)$
- Robust performance model
  - Originally 1-D movements
  - Applies to 2-D movements



Index of Performance or Throughput

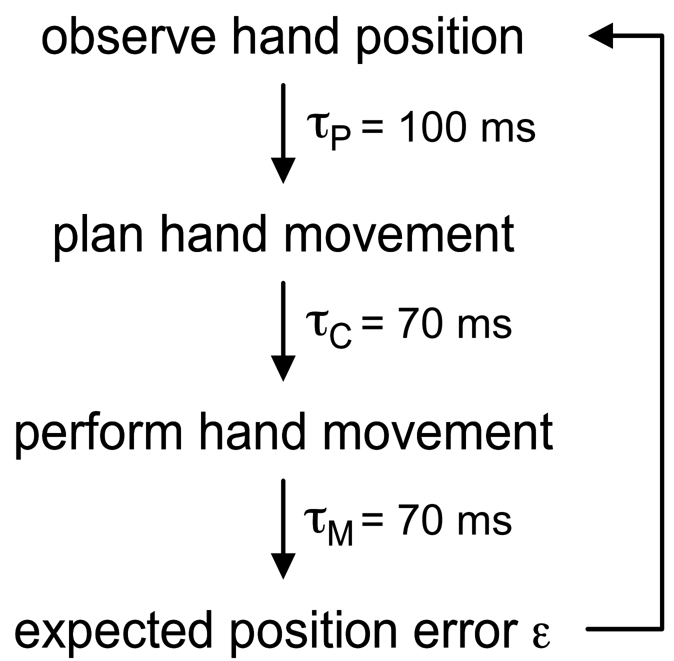
- Fitts' thesis
  - Fixed information-transmission capacity of the motor system
- Tradeoff between speed and accuracy
  - cf. handwriting
  - Relates amplitude, movement speed, variability
- Movement generates information
  - ID = information (number of bits) required to specify movement (amplitude within given tolerance)
- Index of performance
  - IP = ID / MT [bits / sec]

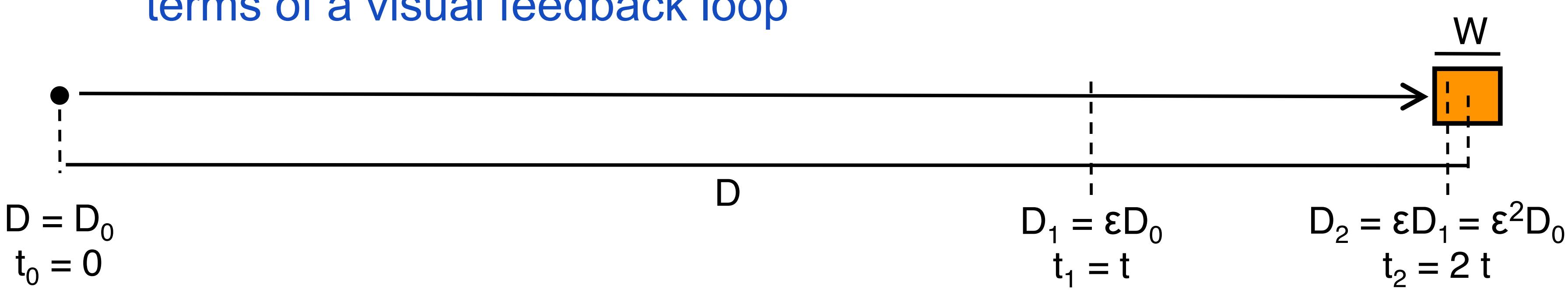


[Fitts, 1954]

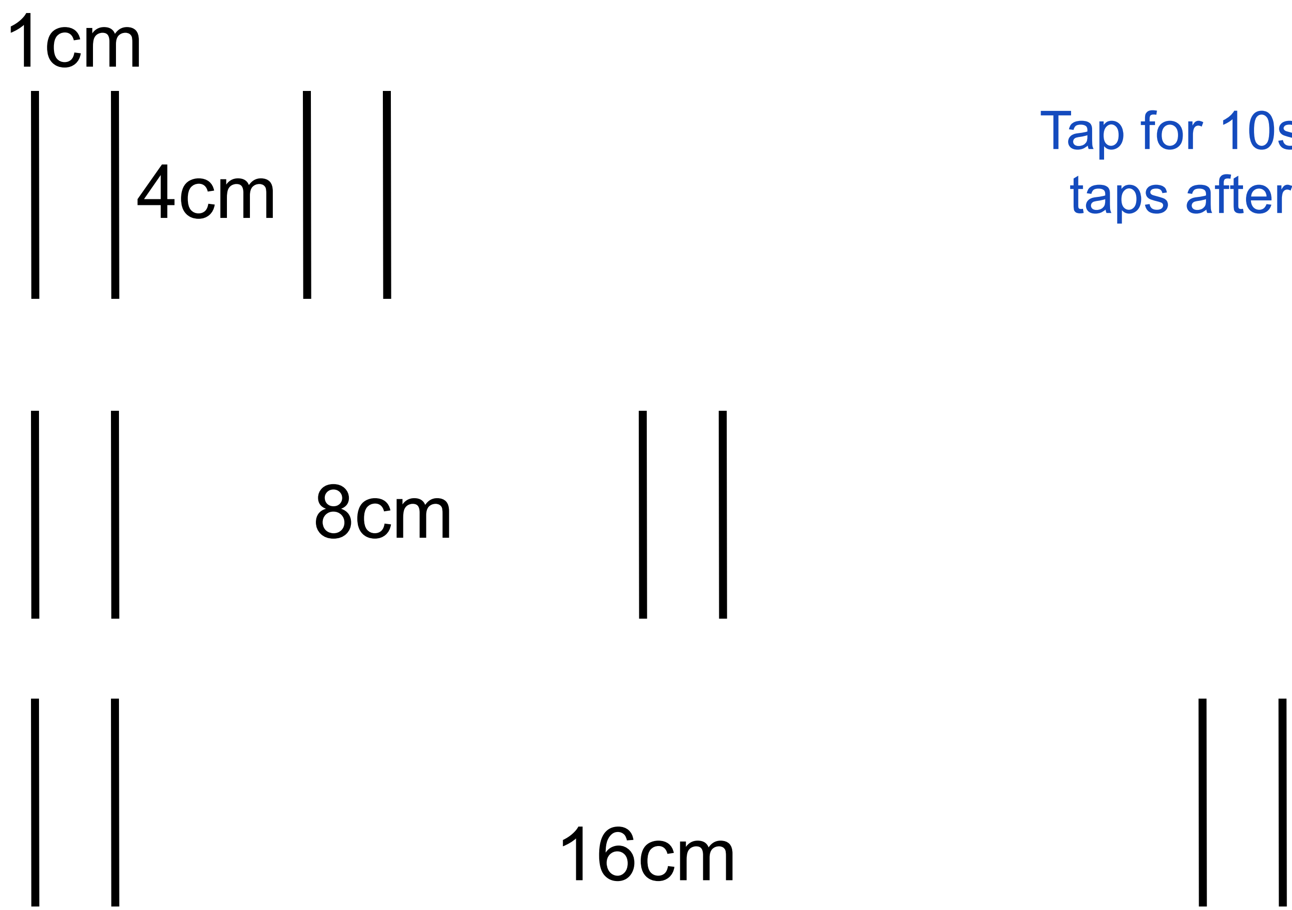
Visual (and Proprioceptive) Feedback Loop

- Assumptions: movement consists of multiple ballistic sub-movements of constant time t and constant error ε
- Deterministic iterative corrections model
  - Movements longer than 200 ms are controlled by visual feedback
  - Interpret constants *a* and *b* in terms of a visual feedback loop





Fitts' Law: Tapping Task



Tap for 10s, count taps afterwards

Determining the Index of Performance

- Draw graph with ID values on the x-axis and average MT values on the y-axis
- Perform a linear regression (e.g., spreadsheet program)

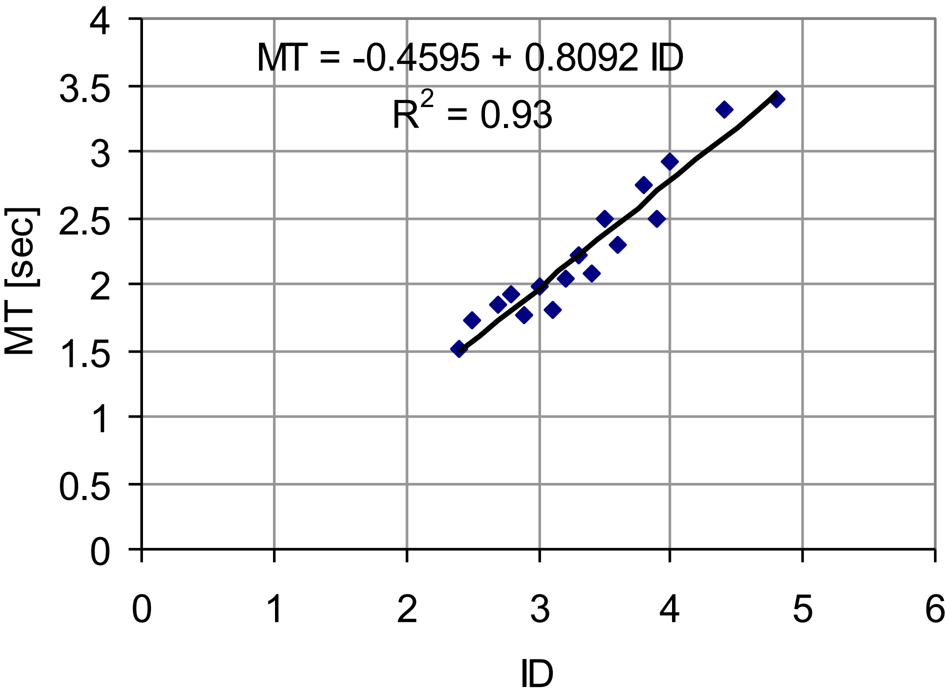
  $MT = a + b\ ID$

  $ID = \log_2(D / W + 1)$

  a = intercept

  b = slope = $1 / IP$

- IP depends on device and limb



THE DESIGN SPACE OF INPUT DEVICES

Input Devices

- “An input device is a transducer from the physical properties of the world into logical parameters of an application” (Card et al.)
- Interaction techniques combine input with feedback
  - Control processes generally need feedback loop
- Input devices enable human-machine dialogues
  - Design of human-machine dialogue = design of artificial languages
  - Communicative intention → movements → application
  - Composition of primitive moves

Properties of Input Devices

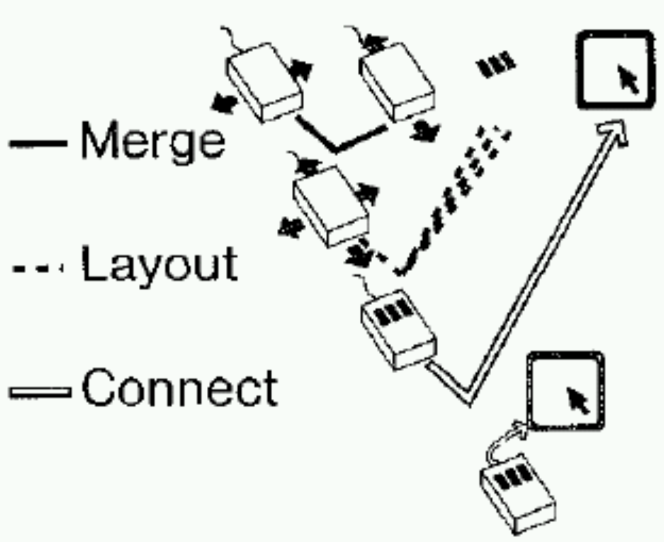
- Property sensed (position, motion, force, etc.)
  - Absolute vs. relative sensing
  - Absolute sensing issue: nulling problem
    (physical position not in agreement with value set in software)
- Number of dimensions
  - 1D, 2D, 3D, 6D
- Indirect vs. direct
  - Indirect: input space and output space are separate
  - Direct: input space = output space
- Device acquisition time
- Control-to-display (C:D) ratio (speed vs. accuracy)
- Issues: clutching, lag, update rate

Generating the Design Space (Card et. al)

- Primitive movement vocabulary

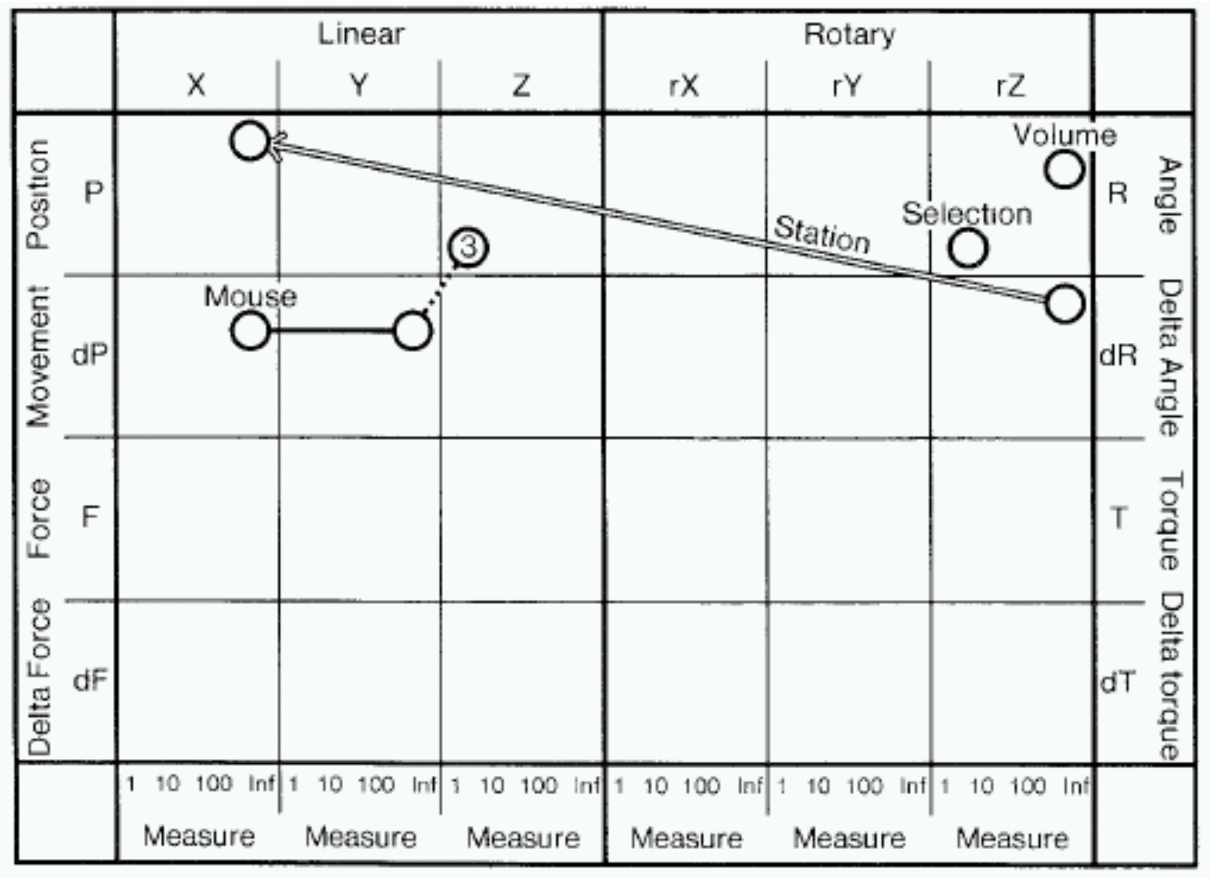
	Linear	Rotary
Position		
Absolute	Position **P**	Rotation **R**
Relative	Movement **dP**	Delta rotation **dR**
Force		
Absolute	Force **F**	Torque **T**
Relative	Delta force **dF**	Delta torque **dT**

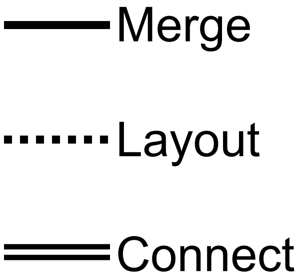
- Composition operators
  - Merge composition: cross product
  - Layout composition: collocation
  - Connect composition: output → input
- Design space of input devices
  - Possible combinations of composition operators with the primitive vocabulary



The Design Space of Input Devices (Card et. al)

- Set of possible combinations of composition operators with the primitive vocabulary



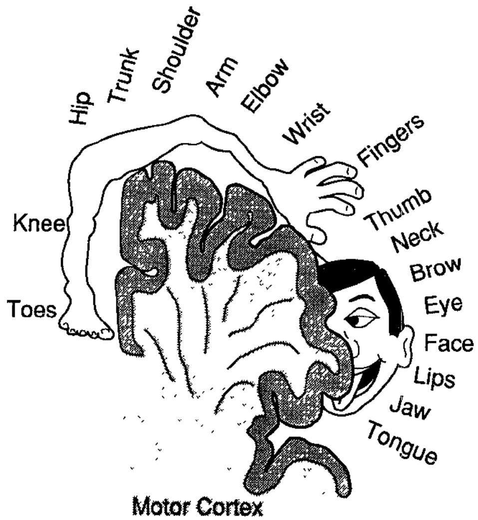


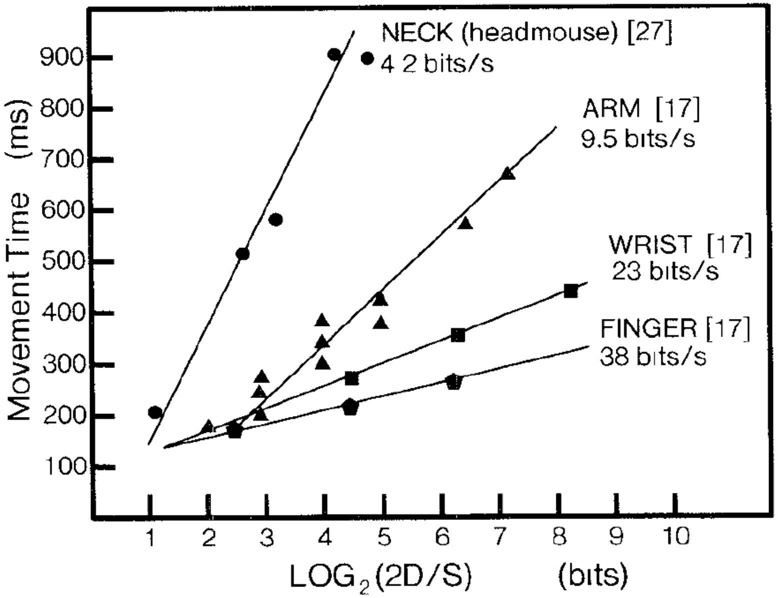
- Touch screen?
- Keyboard?
- Trackball?

Match Input Device to Task

- Use the space to evaluate devices
- Expressiveness
  - "The input conveys exactly and only the intended meaning"
  - Problematic if Out → In do not match
    - Out $\supset$ In: can input illegal values
    - Out $\subset$ In: cannot input all legal values
  - Example: 3D position with touch screen
- Effectiveness
  - "The input conveys the intended meaning with felicity"
  - Pointing speed: device might be slower than unaided hand
  - Pointing precision: convenient selection of small target
  - Example: Augmented reality pointing

Bandwidth





- Speed of use depends on
  - Human: bandwidth of muscle group to which input device attaches
  - Application: precision requirements of the task
  - Device: effective bandwidth of input device

Target	Size S[a] (cm)	I_D (bits)	Mouse (ms)	Headmouse (ms)	Fingers (ms)
Paragraph[b]	5.5	1.18	113	280	30
Word[c]	2.3	2.24	220	540	56
Character[d]	0.41	4.59	440	1100	115
Period[e]	0.069	7.14	690	1710	179

MOBILE TEXT ENTRY

Text Entry on Mobile Devices

Source: http://digitaldaily.allthingsd.com/20091008/omfg-4-1-billion-text-messages-sent-every-day-in-us/

- Mobile text entry is huge
  - SMS (117 million SMS/day in Germany, 2011; 2.5 bln. USA?)
  - Twitter (80 million mobile users)
  - Email, calendars, notes, passwords, etc.
- Small devices require alternative input methods
  - Smaller keyboards, stylus input, finger input, gestures
- Many text entry methods exist
  - Companies are ambitiously searching for improvements

Key-based

Finger-based

Stylus-based

Tilt-based

SMS and Twitter on Mobile Devices

- SMS
  - Average US teenager sends 3339 text messages a month (in 2010, Source: Mobile Future)
  - Texts per day: adults: 10, boys 14-17: 30, girls 14-17: 100 (Source: mashable.com/2010/08/17/text-messaging-infographic)
- Twitter
  - 80 million Twitter mobile users (2011, Source: realtimemarketer.com)
  - Mobile Twitter usage increases by 347% from 2009 to 2010 (Source: Mobile Future)
  - Twitter has 165 million users, 50% use Twitter mobile (April 2011, Source: www.digitalbuzzblog.com/2011-mobile-statistics-stats-facts-marketing-infographic/)

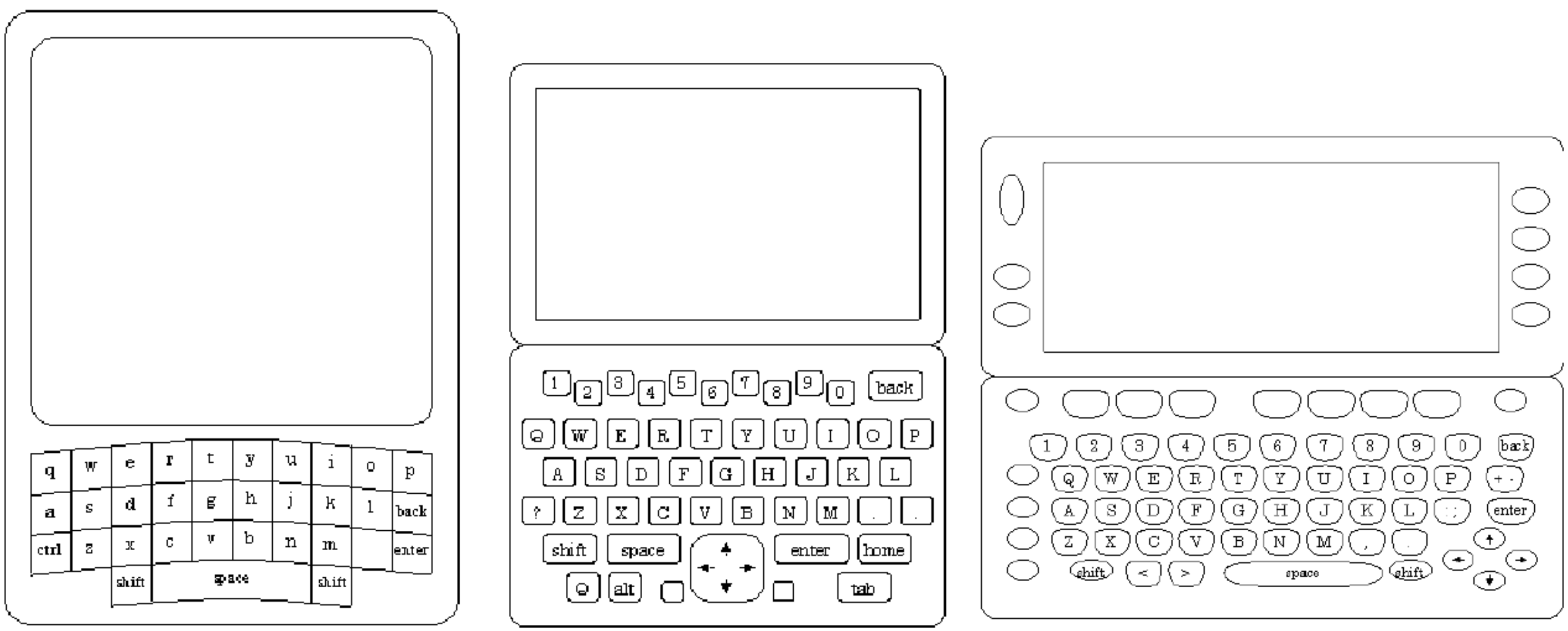
http://www.mobilefuture.org

Text Entry Speed on Mobile Devices

- Goal: High-speed entry at low error rates
  - Movement minimization
  - Low attention demand
  - Low cognitive demand
- Entry speeds depend on task type and practice
- Typical text entry speeds
  - Handwriting speeds: 13-22 words per minute (wpm)
  - Desktop touch typing: 60+ wpm
  - Soft (on-screen) keyboards:
    40+ wpm after lots of practice,
    typically 18-28 wpm for qwerty,
    5-7 wpm for unfamiliar layout

Keyboard Layouts for Mobile Devices

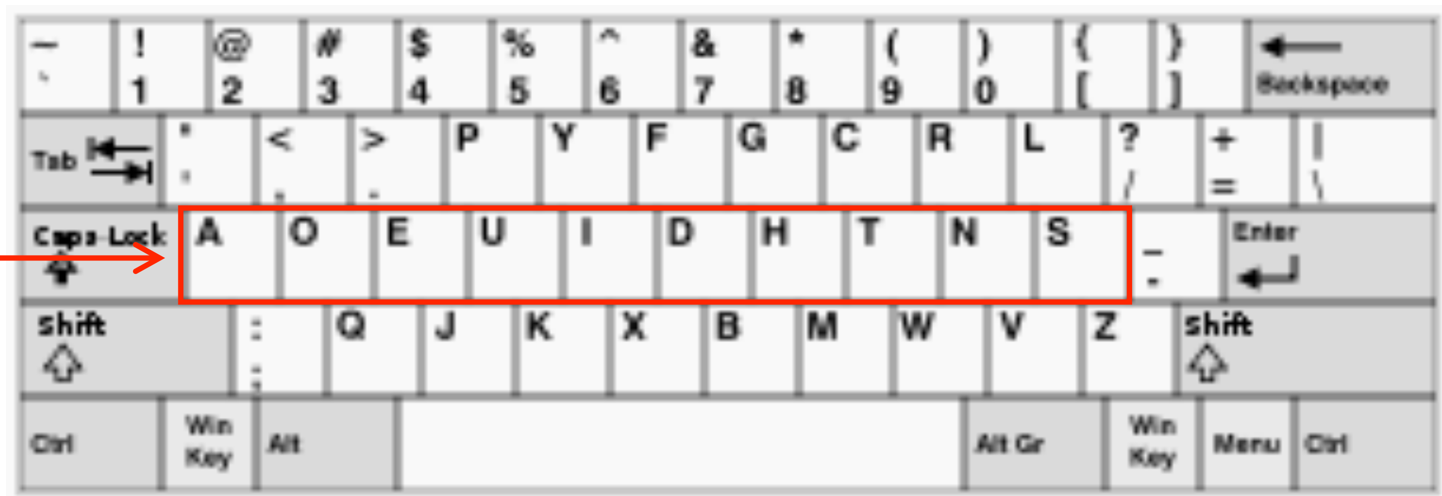
- Querty variations
  - Querty designed to prevent typing machines from jamming
    - alternate between sides of the keyboard



Dvorak Keyboard

- Speed typing by
  - Maximizing home row (where fingers rest)
  - Alternate hand typing
- Most frequent letters and digraphs easiest to type

Home row

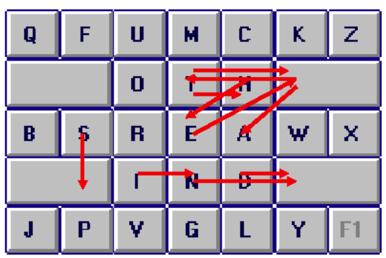


Fitaly and Opti Keyboards

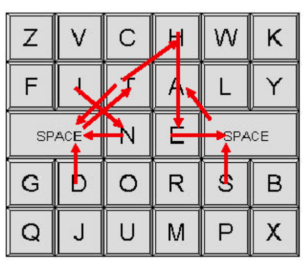
- Designed for stylus input on on-screen keyboards
- Minimizing stylus movement during text entry
- Stylus movement for entering the ten most and least frequent digrams:

Top 10

Digram	Frequency
e_	14788053
_t	11565380
th	10301807
he	9436372
_a	8454634
s_	8359914
n_	6419069
t_	6336756
d_	6235838
in	5534329
...	...
uz	626
zl	626
mh	613
lh	584
cn	567
oq	547
aa	546
xq	543
wm	540
ij	536



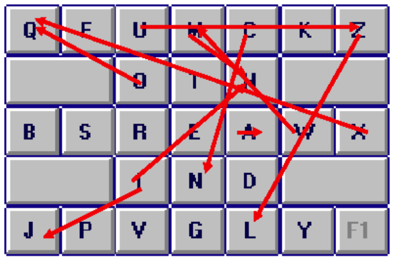
Opti



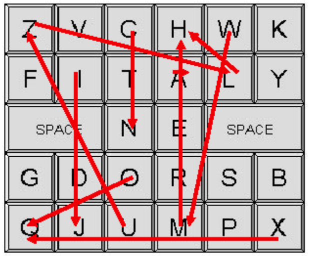
Fitaly

Bottom 10

Digram	Frequency
e_	14788053
_t	11565380
th	10301807
he	9436372
_a	8454634
s_	8359914
n_	6419069
t_	6336756
d_	6235838
in	5534329
...	...
uz	626
zl	626
mh	613
lh	584
cn	567
oq	547
aa	546
xq	543
wm	540
ij	536



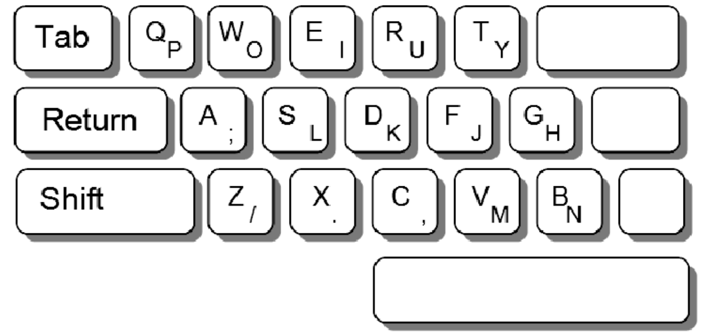
Opti



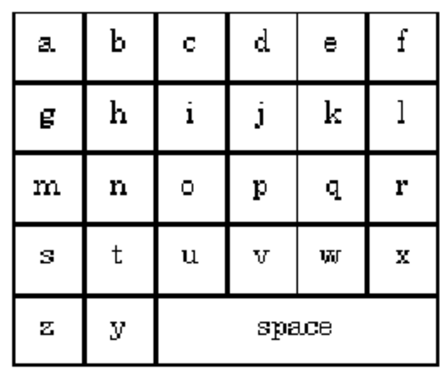
Fitaly

Half-Qwerty and ABC Keyboards

- Half-qwerty
  - One-handed operation
  - 30 wpm

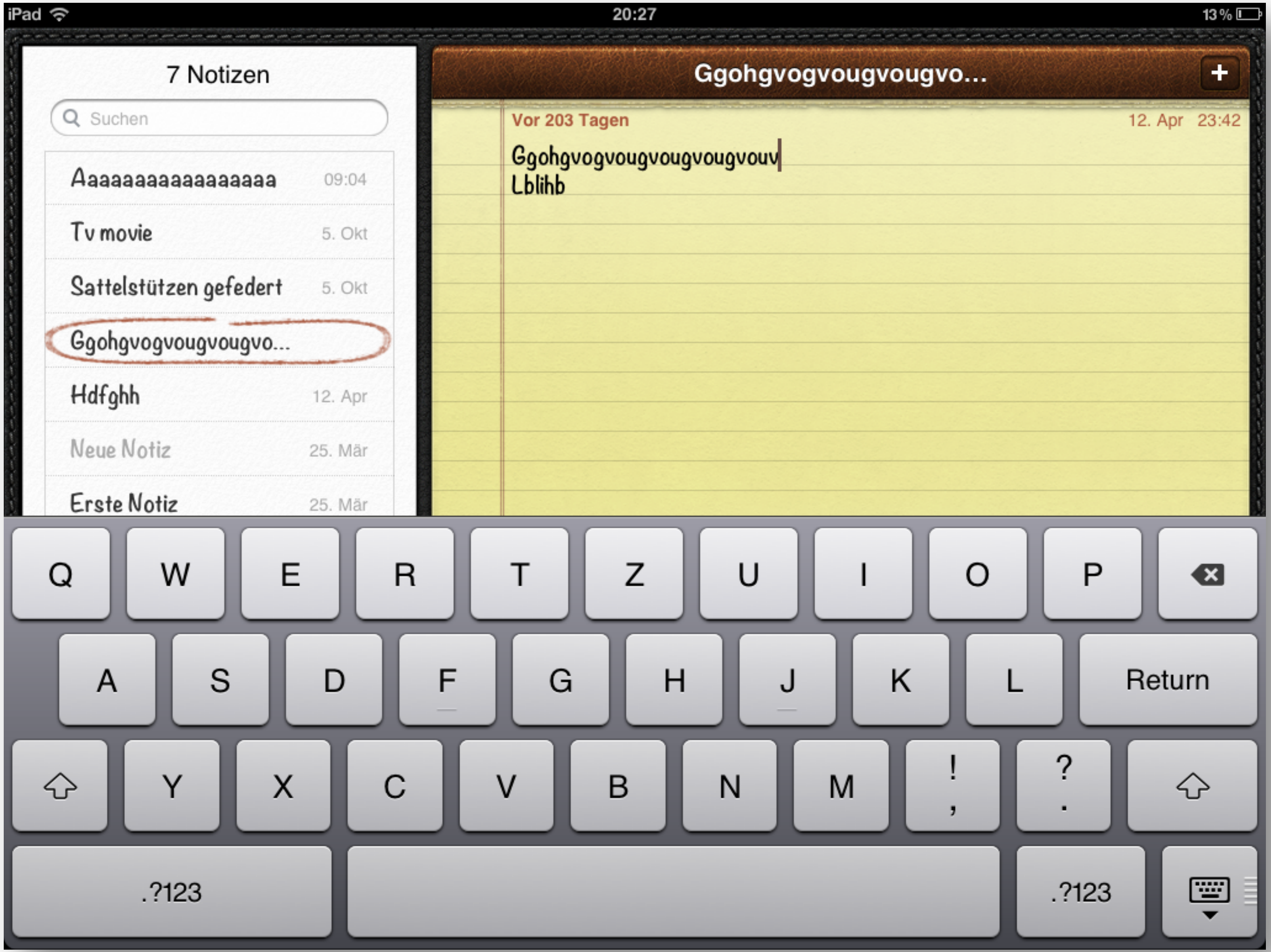


- ABC keyboards
  - Familiar arrangement
  - Non-qwerty shape



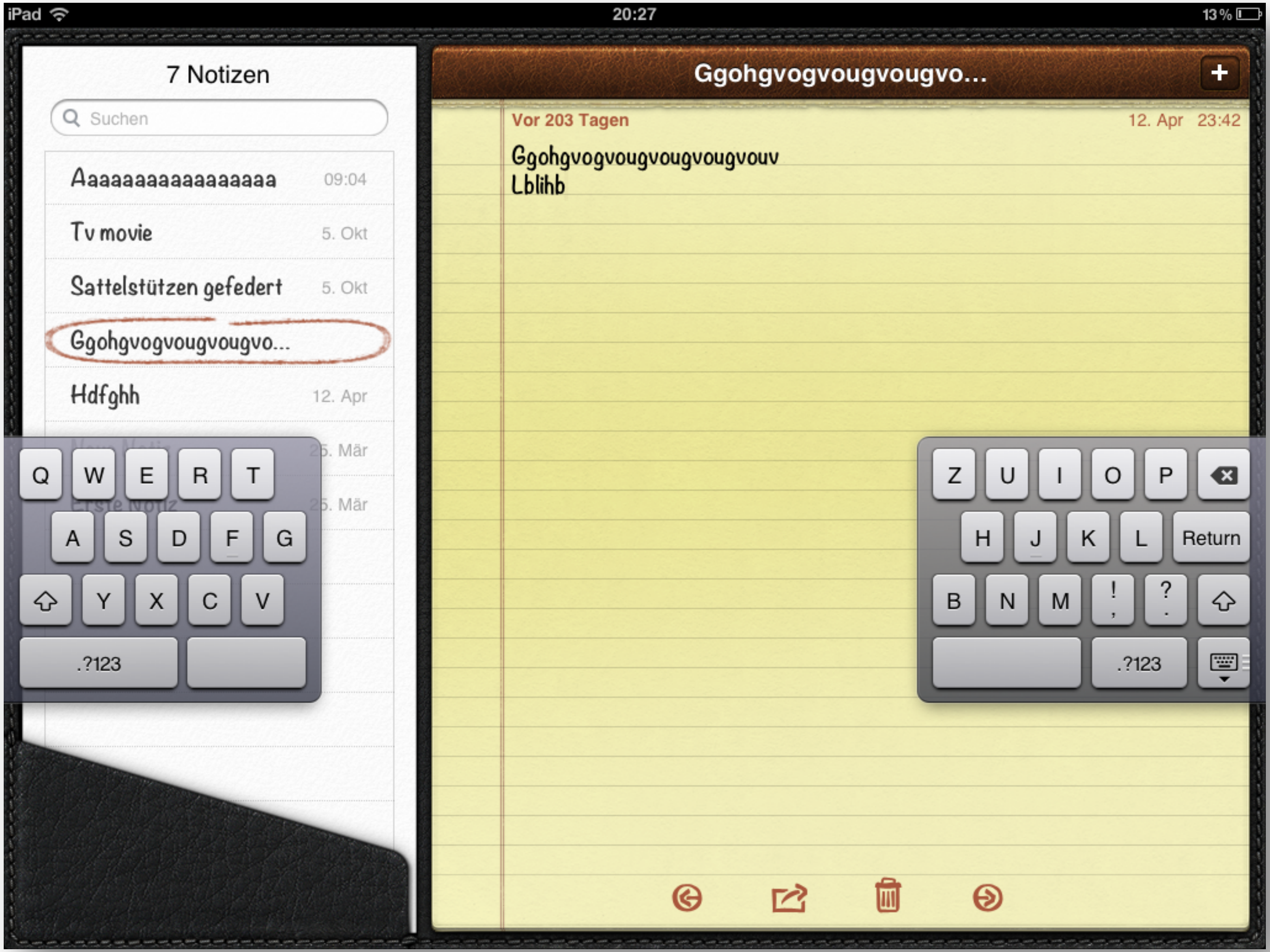
Keyboard Layouts for Tablets

- Problem?



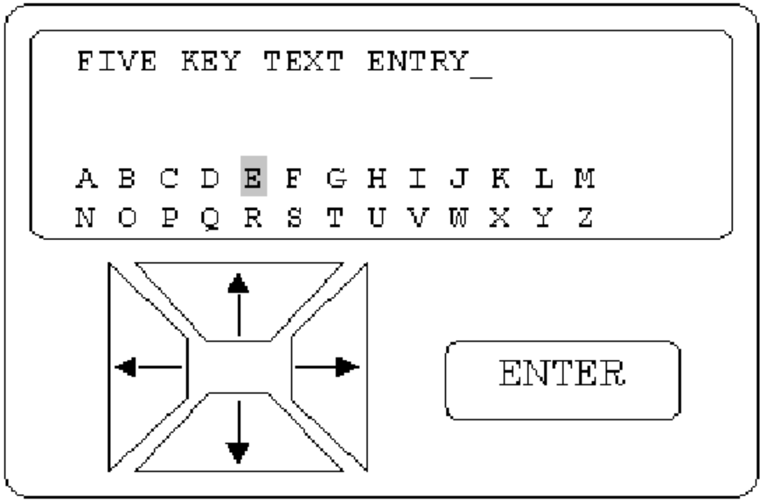
Keyboard Layouts for Tablets

- Vorteile?
- Nachteile?



Very Small Devices

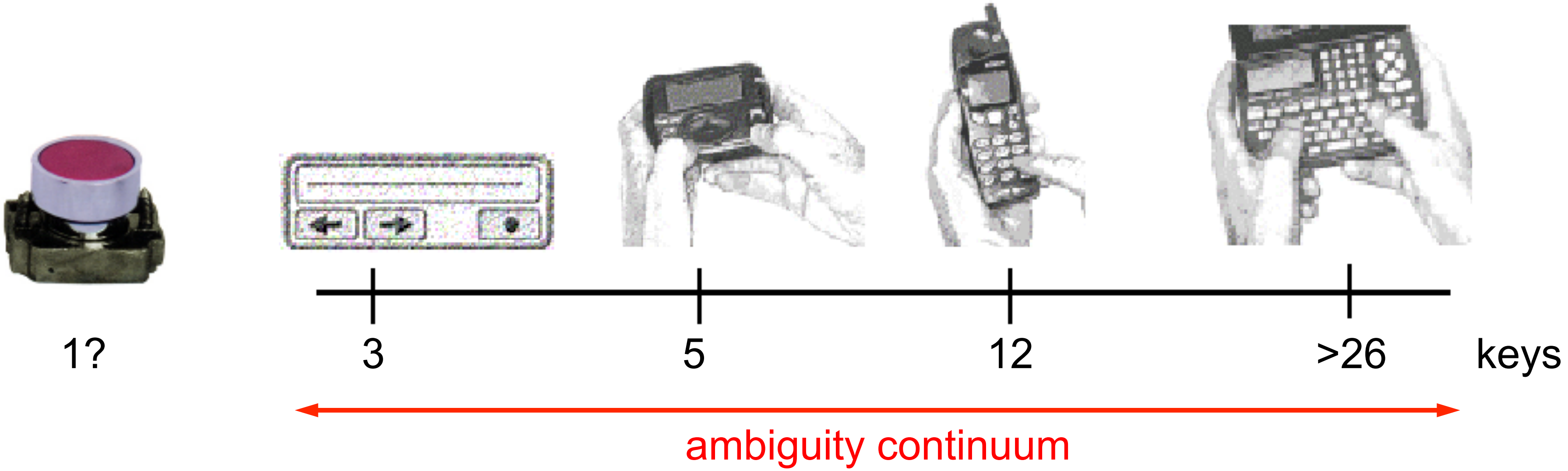
- 5 keys (e.g., pager)



- 3 keys (e.g., watch)

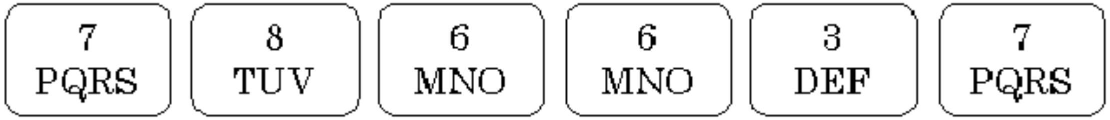
Keyboards and Ambiguity

- Keyboard miniaturization: smaller keys, fewer keys
- Unambiguous keyboards
  - One key, one character
- Ambiguous keyboards
  - One key, many characters
  - Disambiguation methods (manually driven, semiautomatic)



Ambiguity

- Ambiguity occurs if fewer keys than symbols in the language
- Disambiguation needed to select intended letter from possibilities
- Typical example: Phone keypad



?

R	U	N	N	E	R
S	U	M	M	E	R
S	T	O	N	E	S

Unambiguous Keyboards

- One key, one character
- FasTap keyboard
  - Keys in space between keys
  - 9.3 wpm

FastTap keyboard

Ambiguous Keyboards

- One key, many characters
- Standard 12-button phone keyboard, larger variants

Nokia N73

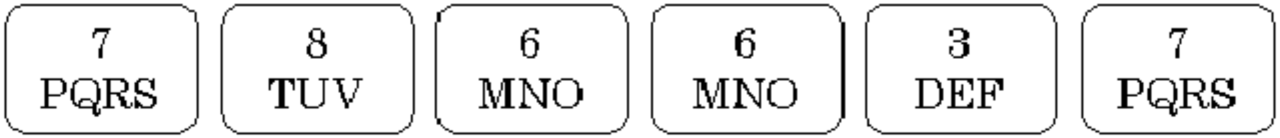
Twiddler, chord keyboard

Blackberry 7100

Manual Disambiguation

- Consecutive disambiguation
  - Press key, then disambiguate
  - Example: Multitap
    - Disambiguating presses on same key (timeout or timeout kill)
- Concurrent disambiguation
  - Disambiguate while pressing key (via tilting or chord)
  - Example: Tilting
    - Tilt in a certain direction while pressing
  - Example: Chord-keyboard on rear of device
    - Not widely used

Disambiguation by Multitap

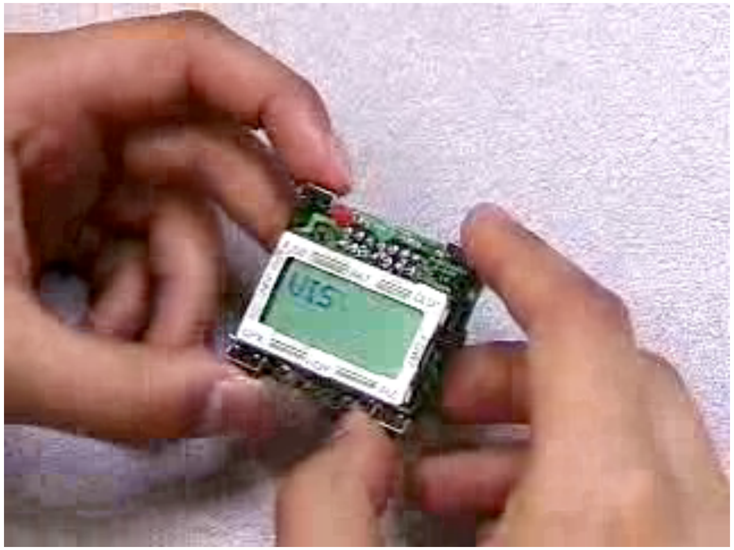



```
RUNNER = 7778866n6633777
           R U N  N E  R

SUMMER = 7777886n633777
            S UM M E  R

STONES = 77778666N66337777
            ST  O  N E   S
```

“n” = next character on key

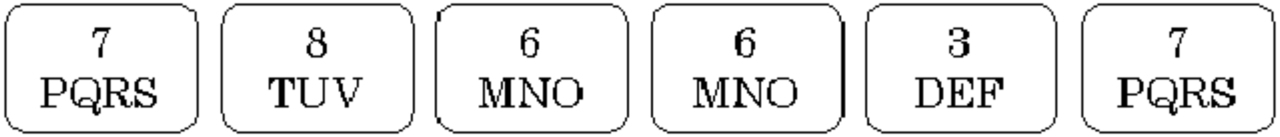
TiltType, Univ. Washington

- Text input method for watches or pagers
- Press and hold button while tilting device
- 9 tilting directions (corners + edges)
- Buttons select to character set



Kurt Partridge et al.: TiltType: Accelerometer-Supported Text Entry for Very Small Devices. UIST 2002 technote
portolano.cs.washington.edu/projects/tilttype

Dictionary-Based Disambiguation (T9)



- Term frequency stored in dictionary
- Most frequent possi-bility presented first
- "n" = key for next frequent possibility

```
RUNNER = 786637nn
         RUNNE  R

SUMMER = 786637
         SUMMER

STONES = 786637n
         STONE S
```

Simplified Handwriting: Unistroke

- Single-stroke handwriting recognition
  - Each letter is a single stroke, simple recognition
  - Users have to learn the strokes
  - “Graffiti” intuitive unistroke alphabet (5 min practice: 97% accuracy)

- Slow (15 wpm)
- Users have to attend to and respond to recognition process
- Recognition constrains variability of writing styles

Unipad: Language-Based Acceleration for Unistroke

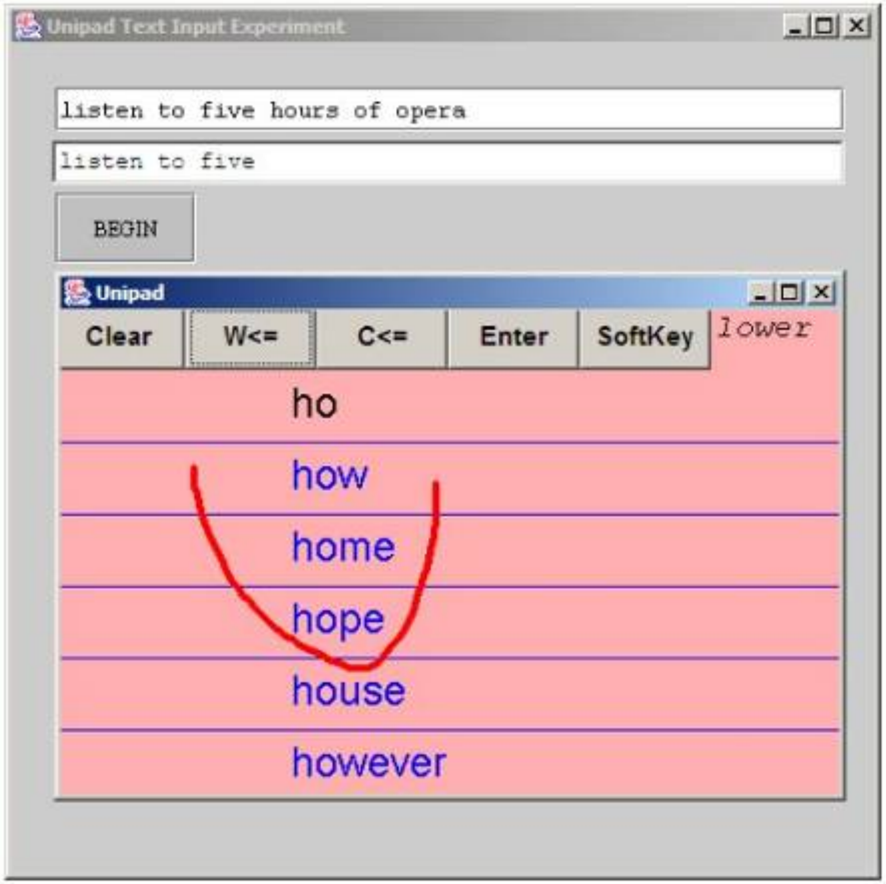
- Speeding up stylus-based text entry
  - Eyes-free entry possible for unistroke
  - Look at suggestions during eyes-free unistrokes
- Language-based acceleration techniques
  - Word completion list based on corpus (word, frequency)
    - Tap candidate
  - Frequent word prompting (“for”, “the”, “you”, “and”, etc.)
    - Tap frequent word
  - Suffix completion based on suffix list (“ing”, “ness”, “ly”, etc.)
    - Top-left to bottom-right stroke, tap suffix

MacKenzie, Chen, Oniszczak: Unipad: Single-stroke text entry with language-based acceleration. NordiCHI 2006.

Unipad: Acceleration by Word Completion

- Word completion example
  - User is entering word “hours”
  - State after two strokes (“ho”)

- Experimental interface
  - First line shows text to enter
  - Second line shows text already entered
  - Pad below
    - Entering strokes
    - Word completion list

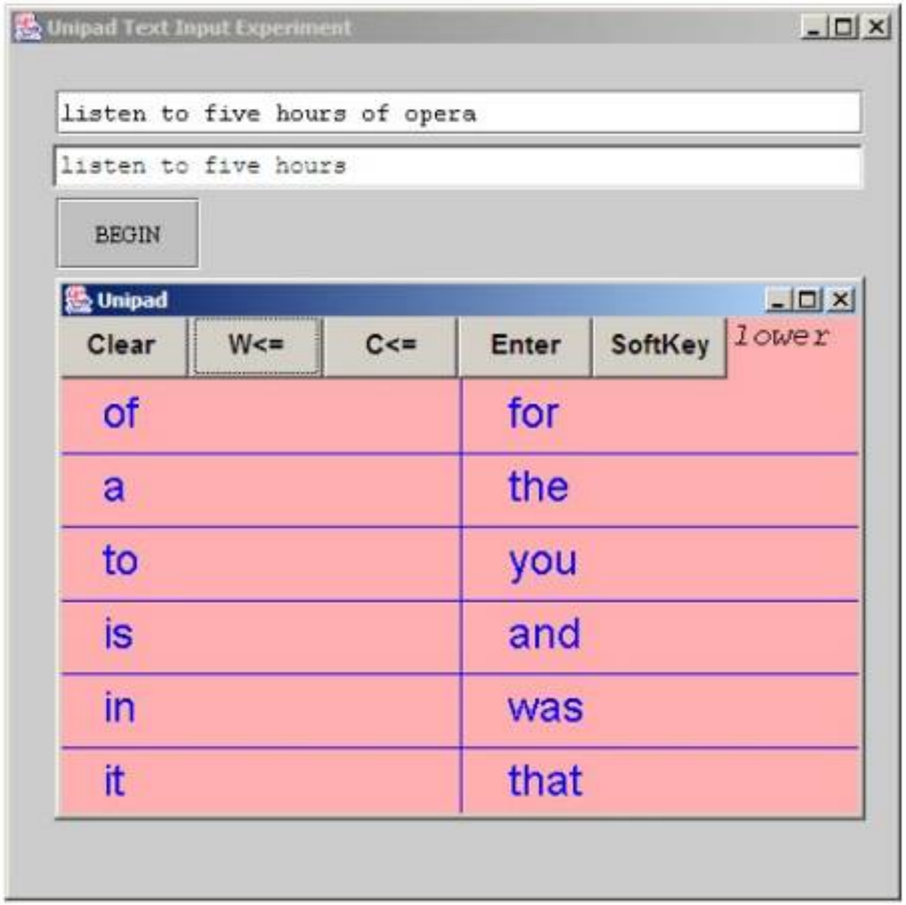


MacKenzie, Chen, Oniszczak: Unipad: Single-stroke text entry with language-based acceleration. NordiCHI 2006.

http://www.yorku.ca/mack/nordichi2006.html

Unipad: Acceleration by Frequent Word

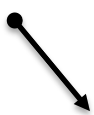
- Frequent word example
  - User is about to enter “of”
- Pad shows frequent word list
  - User taps “of”



MacKenzie, Chen, Oniszczak: Unipad: Single-stroke text entry with language-based acceleration. NordiCHI 2006.

http://www.yorku.ca/mack/nordichi2006.html

Unipad: Acceleration by Suffix Completion

- Suffix completion example
  - User is entering "parking"
  - State after 4 strokes ("park")
- Pad shows word completion list
  - User enters top-left to bottom-right stroke to show suffix list 
- Pad shows suffix list
  - User taps "ing"

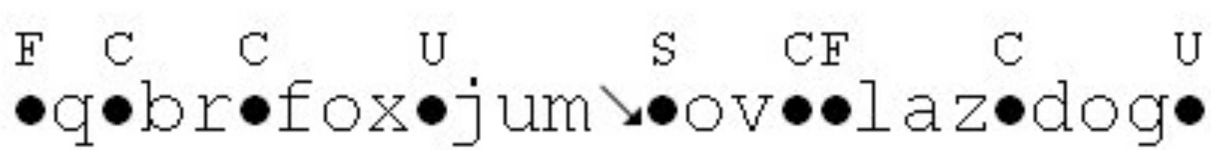
p a r ↘ $\overset{s}{\bullet}$ ($\overset{s}{\bullet}$ = tap entry in suffix list)

MacKenzie, Chen, Oniszczak: Unipad: Single-stroke text entry with language-based acceleration. NordiCHI 2006.

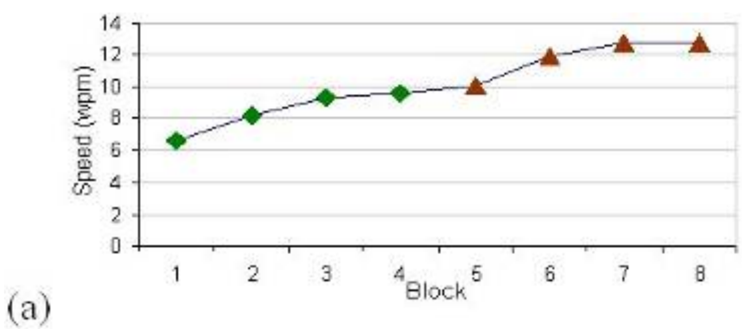
http://www.yorku.ca/mack/nordichi2006.html

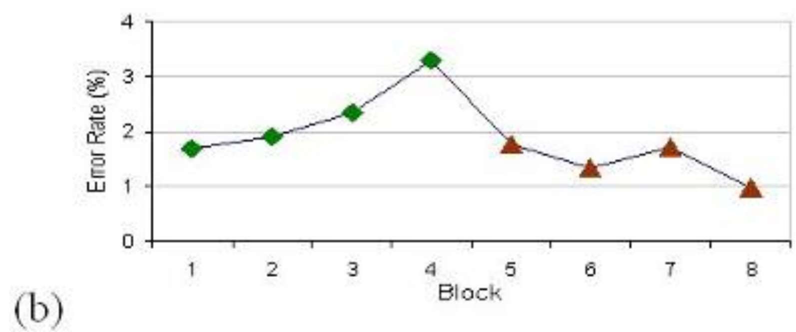
Unipad: Performance

- Entry speed >40 wpm possible
  - KSPC ≈ 0.5 (key strokes per character)
- Expert performance simulated on sentence
  “the quick brown fox jumps over the lazy dog” (43 chars)



(27 strokes)



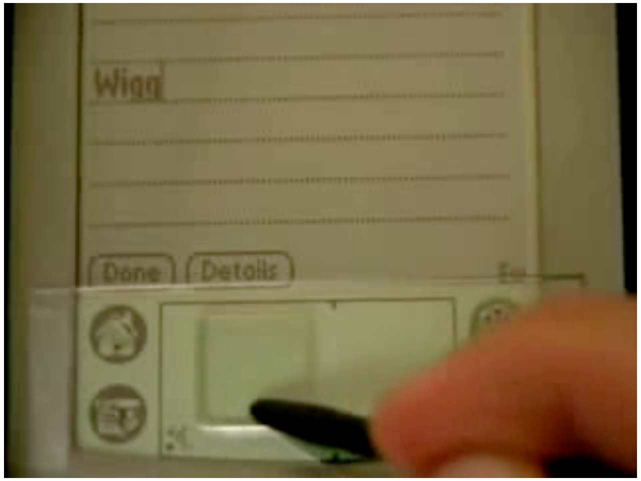


MacKenzie, Chen, Oniszczak: Unipad: Single-stroke text entry with language-based acceleration. NordiCHI 2006.

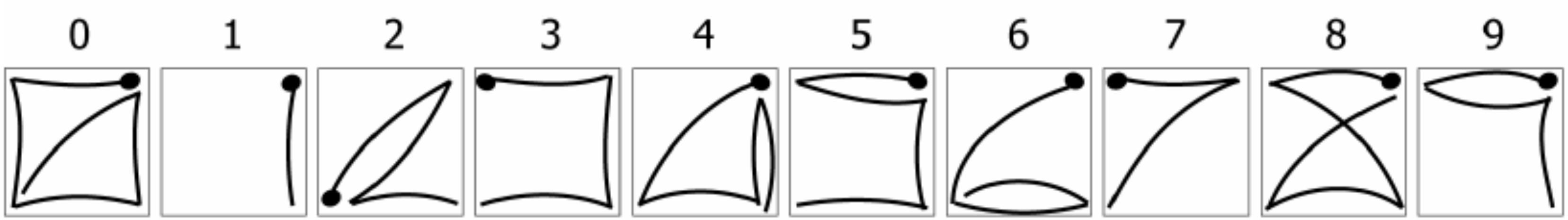
http://www.yorku.ca/mack/nordichi2006.html

EdgeWrite

- Provide physical constraints
- Moving stylus along edges and diagonals of square input area
- People with motor impairments
- Input = Sequence of visited corners



- Example: Digits

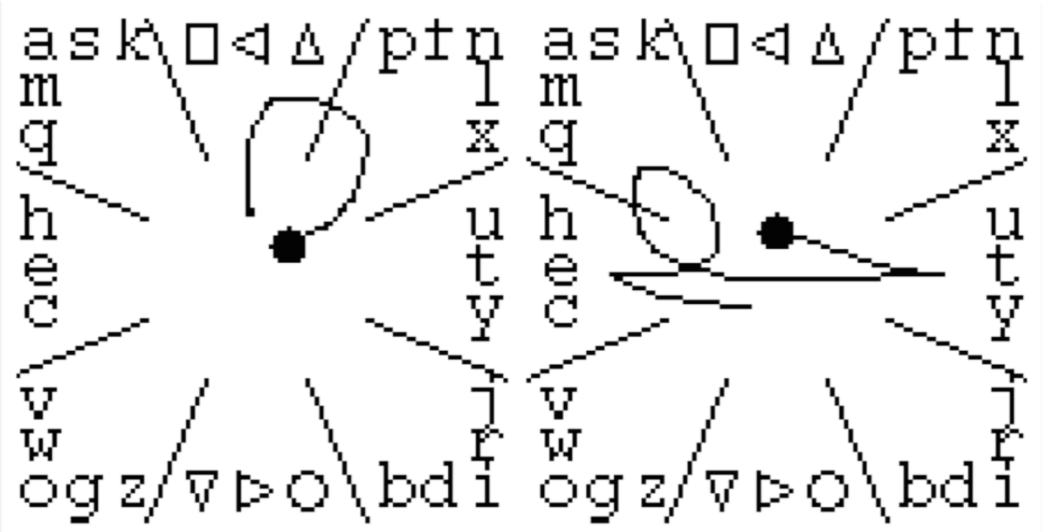


Wobbrock, Myers, Kembel: EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. UIST'03. http://depts.washington.edu/ewrite/

QuickWriting: Gesture-Based Input

- Combine visual keyboards with stylus movements
- Following a path through letters of the word to enter
- Reduced fatigue compared to tapping
- Motor memory for paths

- Ken Perlin: Quikwriting: Continuous Stylus-based Text Entry. UIST'98.



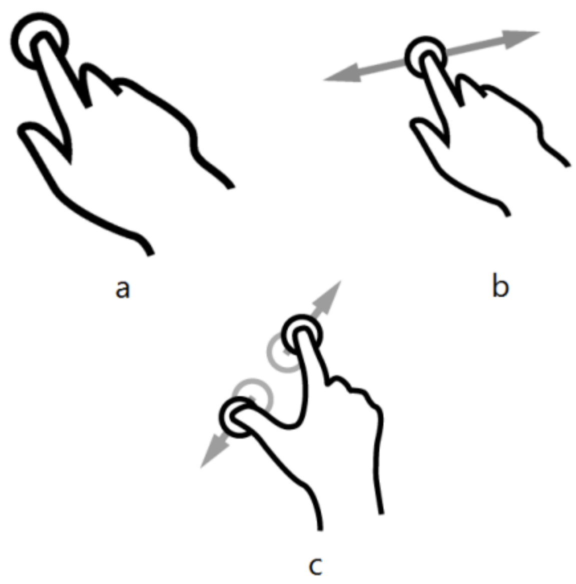
Writing the letter 'f', and the word "the"

Quickwriting, http://mrl.nyu.edu/~perlin/demos/Quikwrite2_0.html

Swype

- Text entry via continuous swipes, lifting between words
- Guesses most likely word from language model
- Manual disambiguation possible
- Example: entering the word “quick”:

- World record text message: 26 words typed in 25.94s
- http://www.swypeinc.com/product.html



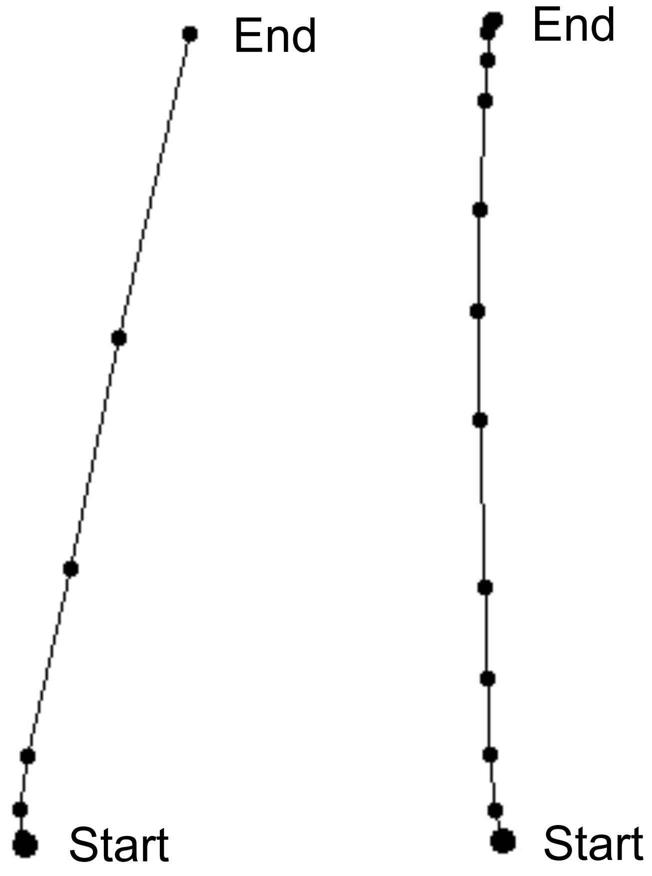
Source: GestureWorks.com

TOUCH SCREEN GESTURES

Difference between these touchscreen gestures?

- One is "flick" and one is "drag"
  - Which is which?

- Relevant gesture parameters
  - Velocity profile
  - Shape
  - Direction



Do you recognize this gesture?

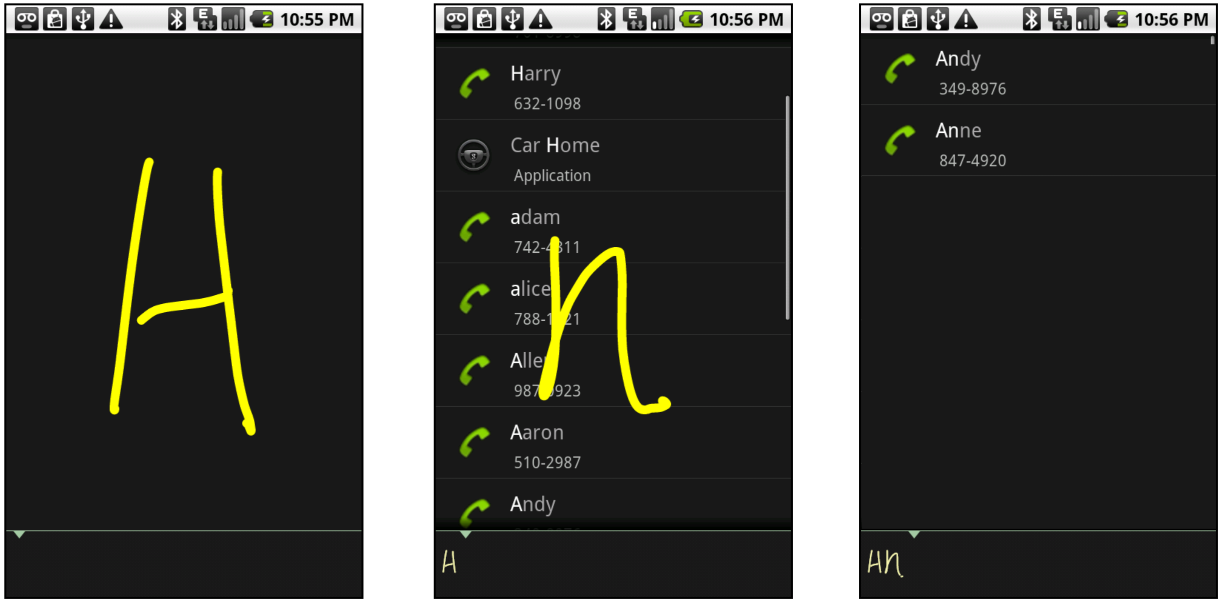
- Multi-touch pinch inwards
  - Typically mapped to “zoom out”
- Relevant gesture parameters
  - Number of touch points
  - Shape
  - Direction
- Challenge: finding intuitive mappings
  - Who should do this?
  - Developers? Designers? Users? Ergonomists?

Gesture Usage

- Letter and digit recognizer
  - Fixed gesture set
  - E.g., based on neural network classifier
  - Trained on large corpus of collected data
- User-customizable recognizer
  - Typically template based
  - Nearest-neighbor matching
- Usage
  - Shortcuts to frequent content
    - Contacts
    - Applications
    - Functionality: “take me home home”
  - Gesture location = operand, gesture shape = operation
    - Annotations, editing marks

Example Application: Gesture Search

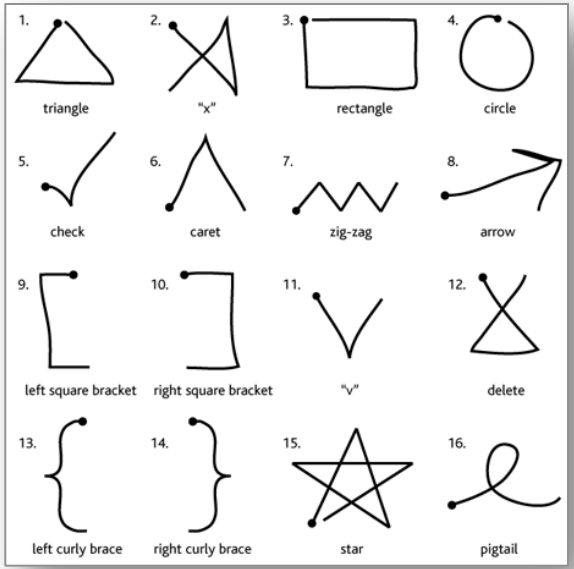
- Find items on Android phones
  - Contacts, applications, songs, bookmarks
  - Drawing alphabet gestures
- http://gesturesearch.googlelabs.com



Yang Li. Beyond Pinch and Flick: Enriching Mobile Gesture Interaction.
IEEE Computer, December 2009. http://yangl.org/pdf/gesturelibrary-ieee2009.pdf

Recognition of Touch Screen Gestures

- Touch screens on many mobile devices
  - Mostly used for tapping (pointing tasks)
  - Suitable for swiping (crossing tasks)
  - Suitable for entering complex gestures
- Gesture recognition challenging
  - Pattern matching, machine learning
- Approaches for simple UI prototyping
  - $1 Recognizer
    - Wobbrock, Wilson, Li. Gestures without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. UIST 2007.
    - http://depts.washington.edu/aimgroup/proj/dollar/
  - Protractor
    - Li. Protractor: A Fast and Accurate Gesture Recognizer. CHI 2010.
    - http://yanglisite.net

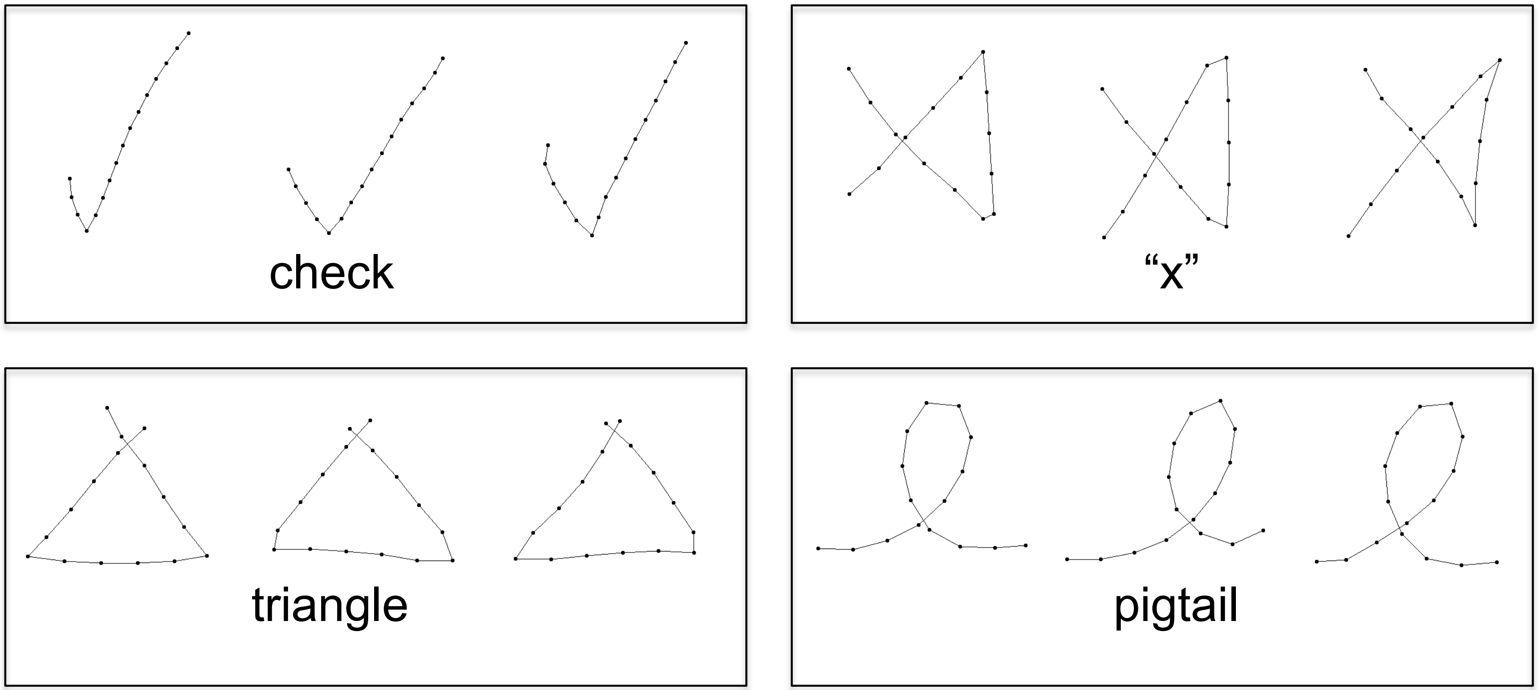


Recognition of User-Defined Touch Screen Gestures

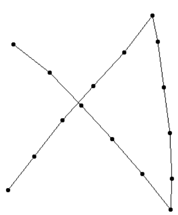
- Template-based recognizers
  - Template preserves shape and sequence of training gesture
  - Nearest neighbor approach
- Process
  - Store training samples as templates (multiple templates per gesture)
  - Compare unknown gesture against templates
  - Choose class of most similar template
- Advantages
  - Purely data-driven, customizable (no assumed underlying model)
  - Small number of examples per class sufficient
- Disadvantages
  - Comparison with all templates can be time and space consuming

Template-Based Recognizers

- Templates (4 classes, 3 examples per class)

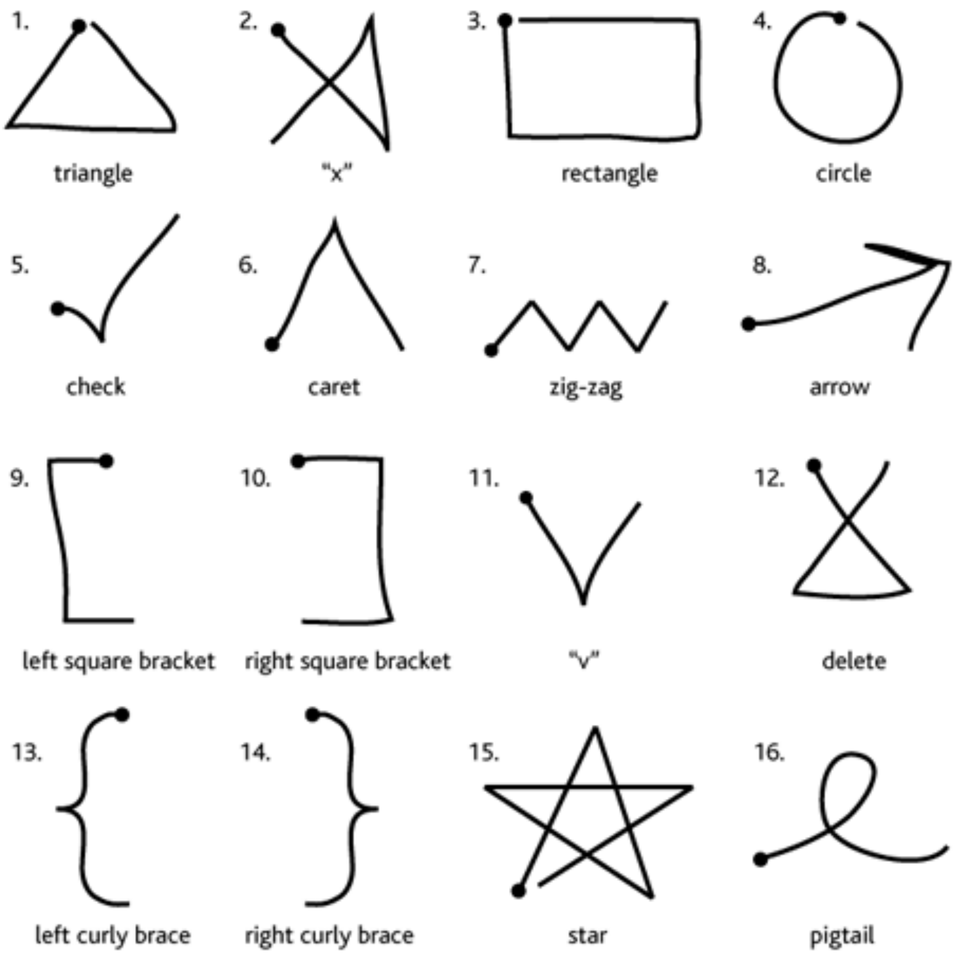


- Query gesture



Gesture Set of “$1 Recognizer”

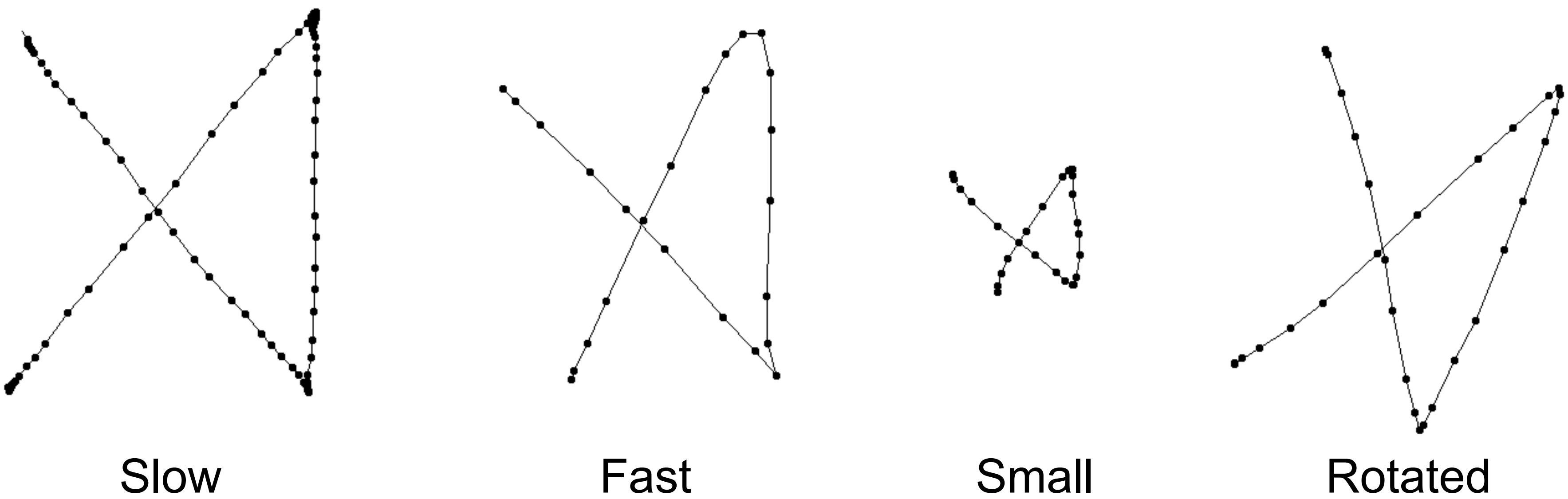
- Unistroke gestures (touch – move – release)
- Dot indicates start point



- http://depts.washington.edu/aimgroup/proj/dollar/

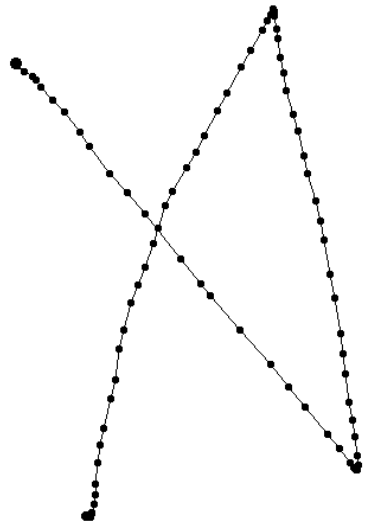
Variability in Raw Input

- Number and distribution of sample points depends on
  - Sampling rate
  - Movement speed and variability
  - Movement amplitude (scale)
  - Initial position and orientation

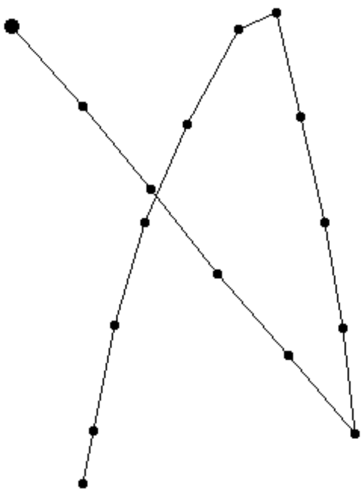


Preprocessing of Gesture Trace

- Resample to fixed number of points
  - E.g., N = 16 points
  - Linear interpolation
  - Length per step = pathLength / (N-1)
- Compute centroid c
- Translate by -c
  - Centered at origin
- Normalize v (to length 1)
  - Treat trace as vector of R^{2N}: $v = x_1, y_1, x_2, y_2, \ldots, x_N, y_N$



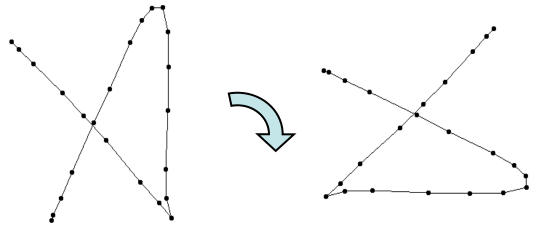
Original trace



Resampled (N = 16)

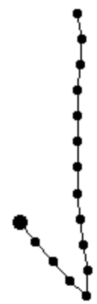
Gesture Recognition

- Gesture recognition = search for most similar template
- Preprocessed query gesture g and templates t_j
  - Resampled (N=16), centroid translated to origin, normalized
- "Most similar" metric?
  - Sum of squared differences between points
    $\min_{j = 1..M} \{ \text{sum}_{i = 1..2N} \{ (g_i - t_{ji})^2 \} \}$
  - Scalar product between query gesture and template
    $\min_{j = 1..M} \{ \text{acos}(\text{sum}_{i = 1..2N} \{ (g_i\, t_{ji})^2 \}) \}$ or
    $\max_{j = 1..M} \{ \text{sum}_{i = 1..2N} \{ (g_i\, t_{ji})^2 \} \}$
- Remaining variability: rotation (and gesture class)

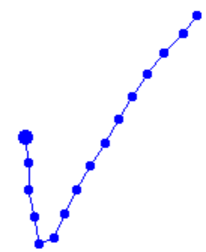


Optimal Angular Distance

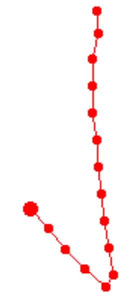
- Orientation of template might be different from query gesture
- Example:



(resampled) query gesture

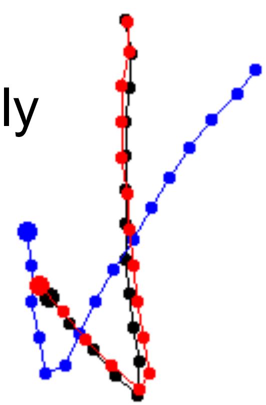


best-matching template



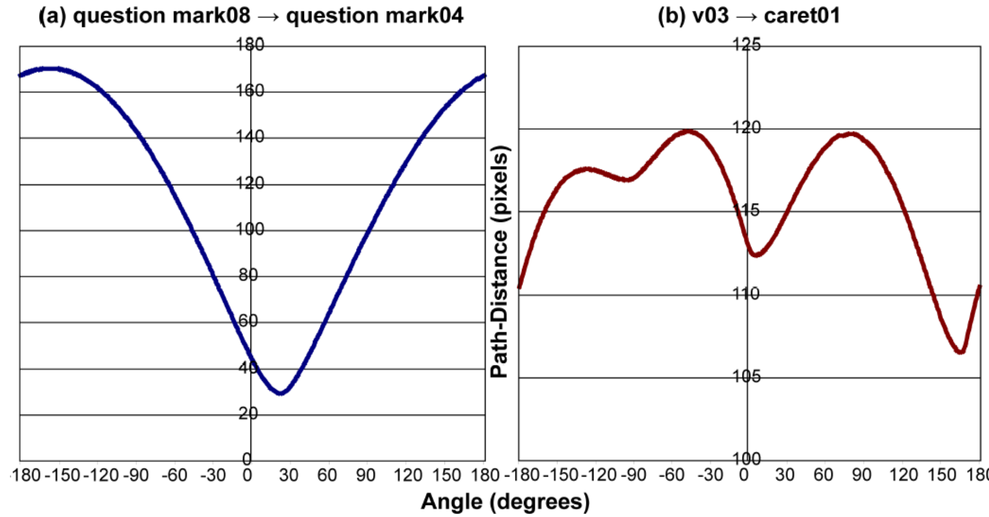
best-matching template optimally rotated to match query

Overlaying query gesture (black) and optimally rotated best-matching template (red):



- How to find the optimal angle?

Finding the Optimal Angular Distance



Wobbrock et al., UIST’07

- Wobbrock et al., UIST’07
  - “Seed and search”: Given query and template, try different orientations and take best one
- Li, “Protractor”, CHI’10
  - Closed form solution!
  - Better speed and performance!
- Closed form solution: Find θ that optimizes metric
  - Metric: Min. angle between query gesture g and template t in R^{2N}
    Optimal angle: $\theta = \text{argmin}_{-\pi \leq \theta \leq \pi} \{ \text{acos}(g \cdot t(\theta)) \}$
  - Equivalent: Max. scalar product between g and t in R^{2N}
    Optimal angle: $\theta = \text{argmax}_{-\pi \leq \theta \leq \pi} \{ g \cdot t(\theta) \}$

Optimal Angular Distance: Closed Form Solution

- Find θ that maximizes scalar product between g and t

  $\theta = \text{argmax}_{-\pi \leq \theta \leq \pi} \{ g \cdot t(\theta) \}$

  $g = x_1, y_1, \ldots, x_N, y_N$

  $t(0) = x^t_1, y^t_1, \ldots, x^t_N, y^t_N$

- Rotate each point in t by θ

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad \begin{aligned} x' &= x\cos\theta - y\sin\theta \\ y' &= x\sin\theta + y\cos\theta \end{aligned}$$

  $t(\theta) = x^t_1 \cos\theta - y^t_1 \sin\theta, \quad x^t_1 \sin\theta + y^t_1 \cos\theta, \ldots$

- Minimize scalar product $g \cdot t(\theta)$

Optimal Angular Distance: Closed Form Solution

- Scalar product $g \cdot t(\theta)$

  $= \text{sum}\{1..N\}(x_i(x^t_i \cos\theta - y^t_i \sin\theta) + y_i (x^t_i \sin\theta + y^t_i \cos\theta))$

  $= \text{sum}\{1..N\}(x_i x^t_i \cos\theta - x_i y^t_i \sin\theta + y_i x^t_i \sin\theta + y_i y^t_i \cos\theta)$

  $= \text{sum}\{1..N\}(\cos\theta\, (x_i x^t_i + y_i y^t_i) + \sin\theta\, (y_i x^t_i - x_i y^t_i))$

  $= \cos\theta\, \text{sum}\{1..N\}(x_i x^t_i + y_i y^t_i) + \sin\theta\, \text{sum}\{1..N\}(y_i x^t_i - x_i y^t_i)$

  $= a \cos\theta + b \sin\theta$

  with $a = \text{sum}\{1..N\}(x_i x^t_i + y_i y^t_i)$

  and $b = \text{sum}\{1..N\}(y_i x^t_i - x_i y^t_i)$

- Remaining task: $\theta = \text{argmin}(a \cos\theta + b \sin\theta)$

  $-a \sin\theta + b \cos\theta = 0 \Leftrightarrow a \sin\theta = b \cos\theta$

  $\Leftrightarrow \sin\theta / \cos\theta = b / a = \tan\theta$

  $\Leftrightarrow \theta = \text{atan}(b / a)$

The End