Mobile Input & Output Technologies

Mensch-Maschine-Interaktion 2, WS 2010/2011

Michael Rohs
michael.rohs@ifi.lmu.de
MHCI Lab, LMU München

Lectures & Exercises

Lecture	Date	Topic
1	12.1.	Mobile Device Platforms
2	19.1.	Introduction to Mobile Interaction
3	26.1.	Prototyping and Evaluation of Mobile Systems
4	2.2.	Mobile Input & Output Technologies
5	9.2.	Location & Context, UI Design for Small Displays

Exercise	Date	Topic
0		Developing countries + Android-Eclipse
1	10.1.	Recipe input
2	17.1.	Touch input, gestures
3	24.1.	Evaluation of mobile LMU Web portal
4	31.1.	Location-based audio

Mobile Text Entry

Partly based on slides by Scott MacKenzie:
Text input for mobile devices by Scott MacKenzie. Tutorial at Mobile HCI 2008.

Text Entry on Mobile Devices

- Mobile text entry is huge
  - SMS (>2.5 billion users; 4.1 billion SMSs each day, US, 2009)
  - 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

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

Key-based

Finger-based

Stylus-based

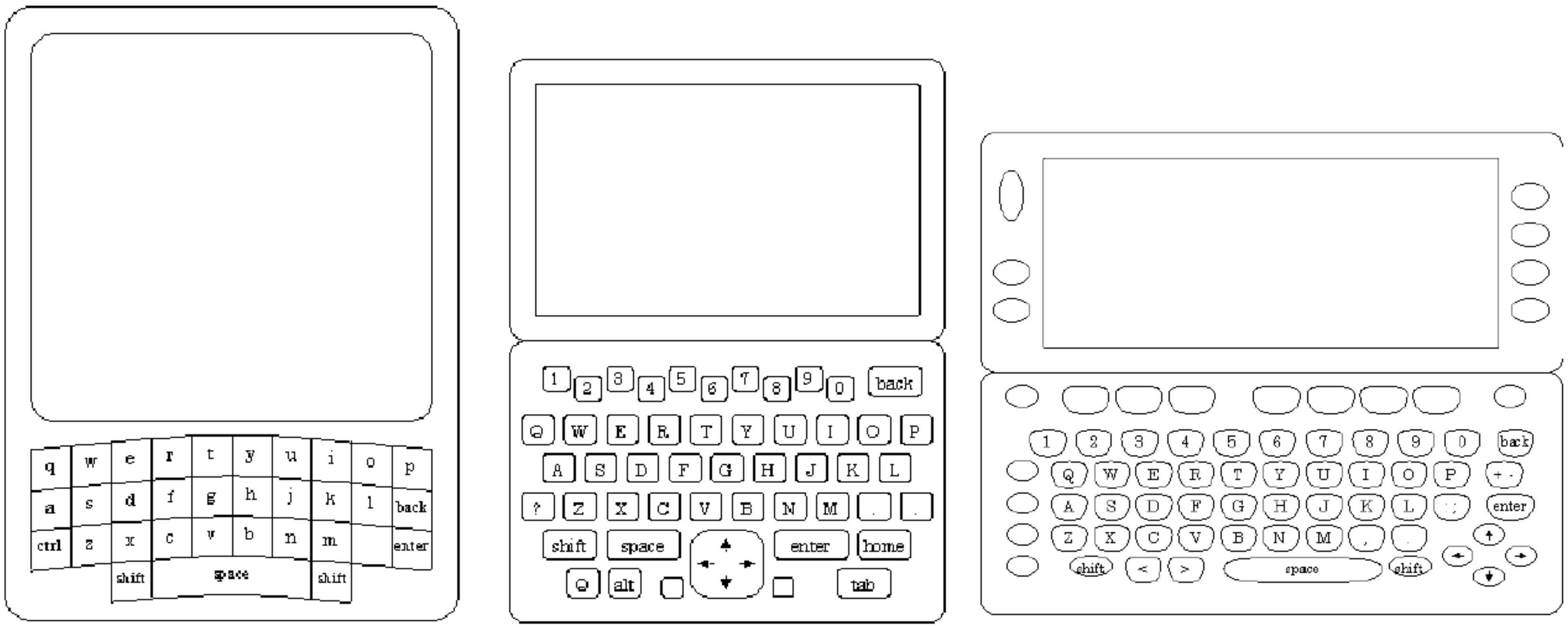
Tilt-based

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

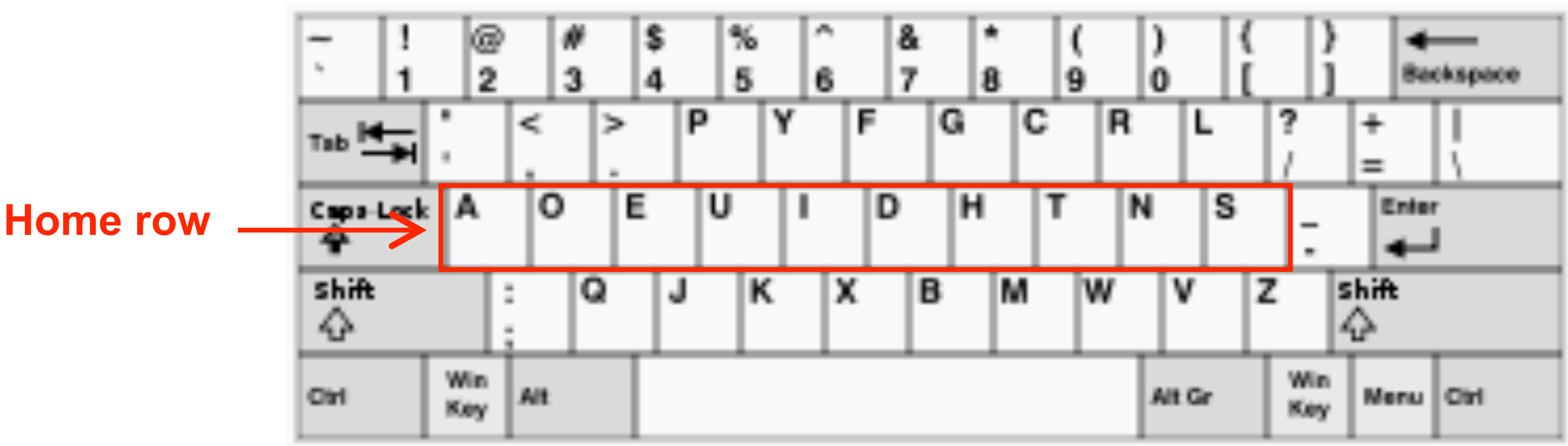
- Querty variations
  - Querty designed to be slow
  - Prevented typing machines from jamming
    - alternate between sides of the keyboard



Source: Text input for mobile devices by Scott MacKenzie. Tutorial Mobile HCI 2008.

Dvorak Keyboard

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

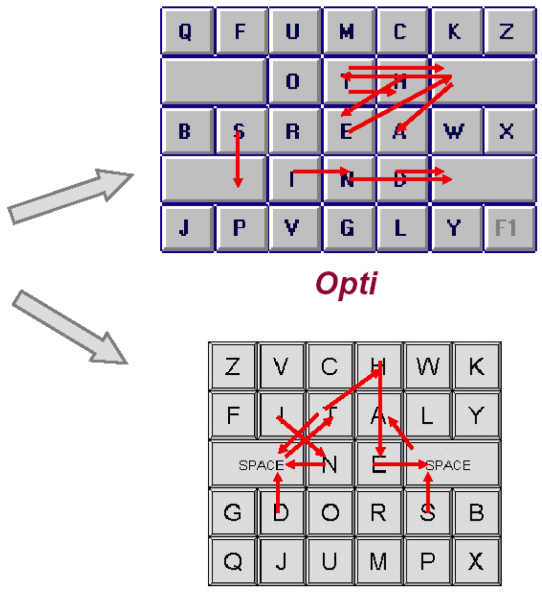


Fitaly and Opti Keyboards

- Designed for stylus input on soft (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

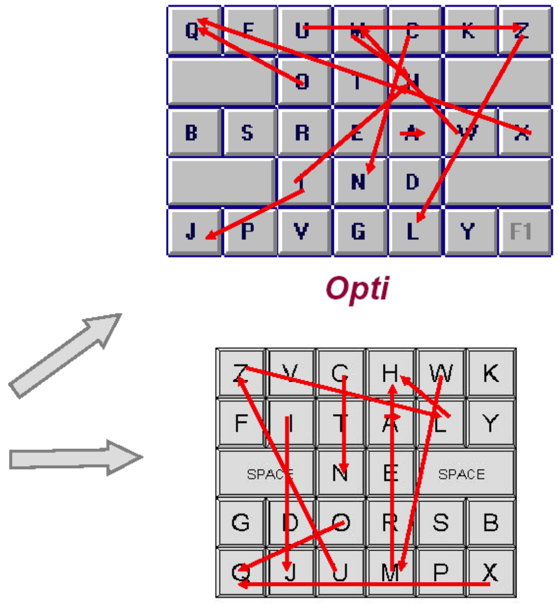


Opti

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



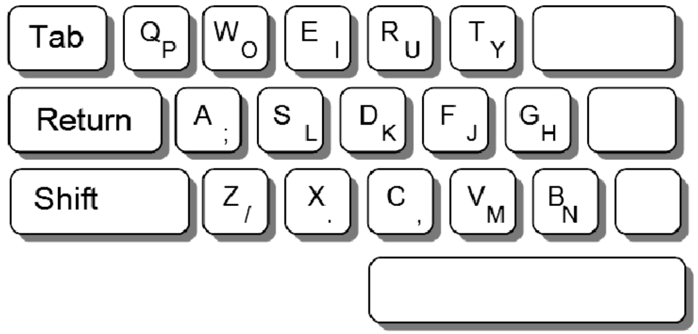
Opti

Fitaly

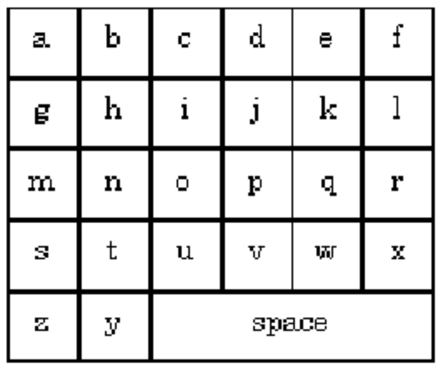
Source: Text input for mobile devices by Scott MacKenzie. Tutorial Mobile HCI 2008.

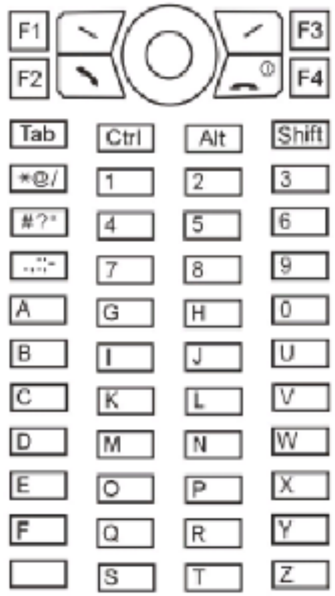
Half-Qwerty and ABC Keyboards

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



- ABC keyboards
  - Familiar arrangement
  - Non-qwerty shape

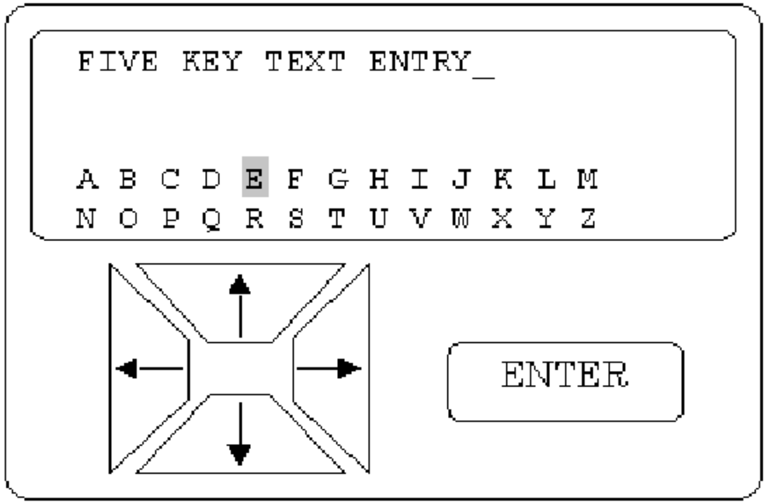




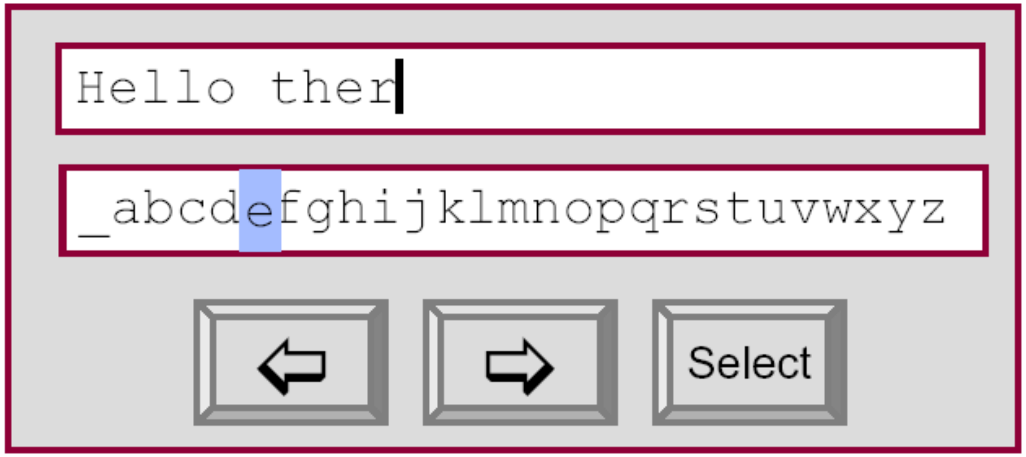
Source: Text input for mobile devices by Scott MacKenzie. Tutorial Mobile HCI 2008.

Very Small Devices

- 5 keys (e.g., pager)



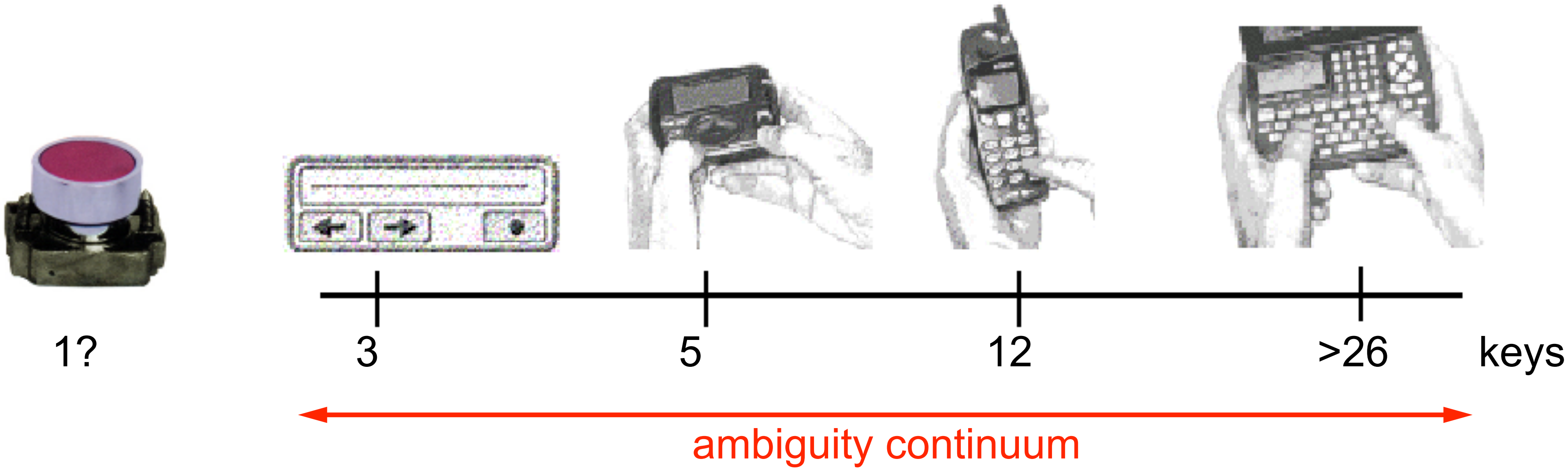
- 3 keys (e.g., watch)



Source: Text input for mobile devices by Scott MacKenzie. Tutorial Mobile HCI 2008.

Keyboards and Ambiguity

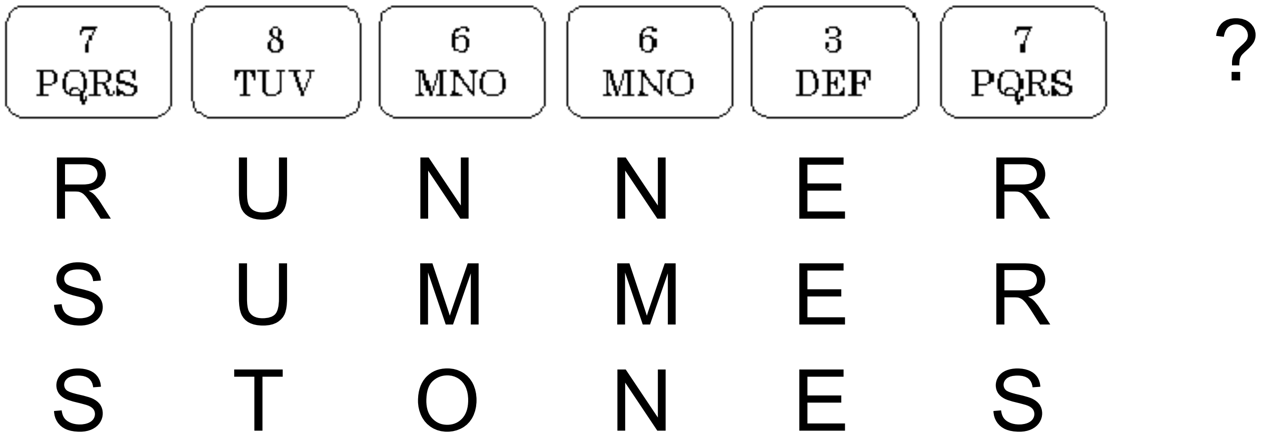
- Keyboard miniaturization: Smaller keys, Less keys
- Unambiguous keyboards
  - One key, one character
- Ambiguous keyboards
  - One key, many characters
  - Disambiguation methods (manually driven, semiautomatic)



Source: Text input for mobile devices by Scott MacKenzie. Tutorial Mobile HCI 2008.

Ambiguity

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



Source: Text input for mobile devices by Scott MacKenzie. Tutorial Mobile HCI 2008.

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

7	8	6	6	3	7
PQRS	TUV	MNO	MNO	DEF	PQRS

```
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

Source: Text input for mobile devices by Scott MacKenzie. Tutorial Mobile HCI 2008.

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)

7	8	6	6	3	7
PQRS	TUV	MNO	MNO	DEF	PQRS

- 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
```

Source: Text input for mobile devices by Scott MacKenzie. Tutorial Mobile HCI 2008.

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

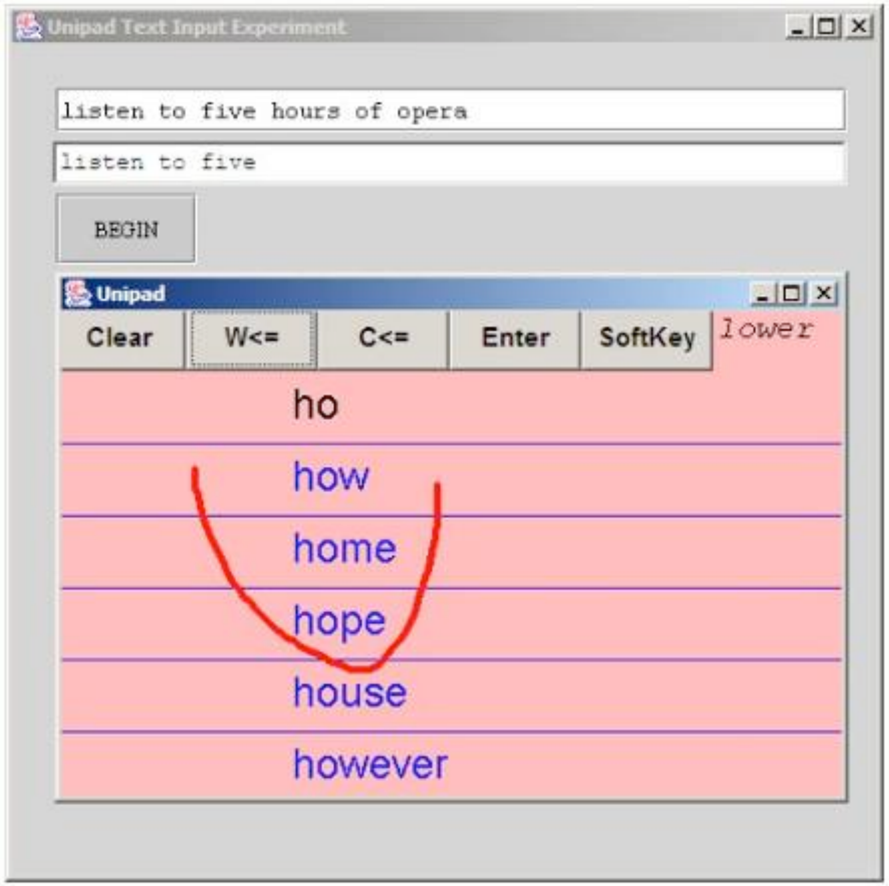
- Speeding up stylus-based text entry
  - Eyes-free entry possible for unistroke
  - Look at suggestions during eyes-free unistrokes

- 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

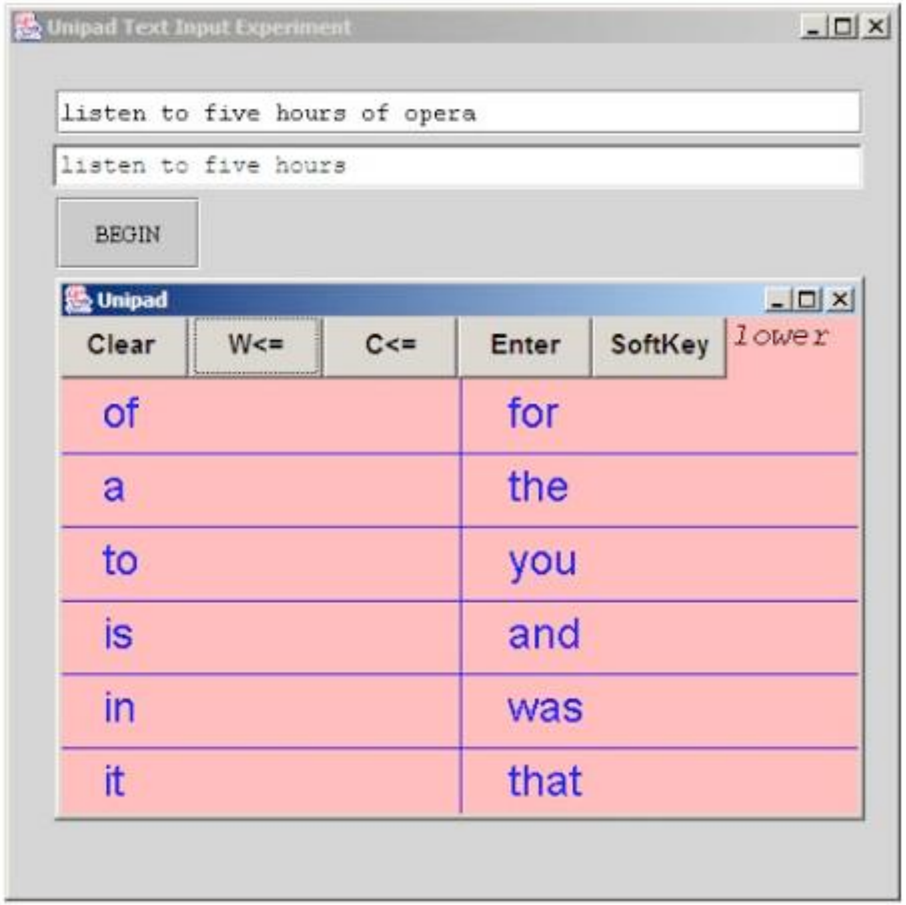


MacKenzie, Chen, Oniszczak: Unipad: Single-stroke text entry with language-based acceleration. NordiCHI 2006.

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

Unipad: Acceleration by Frequent Word

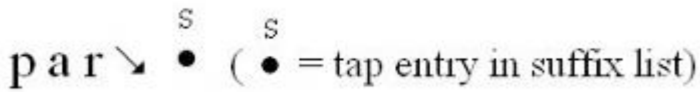
- Frequent word example
  - User is about to enter “of”
- Pad shows frequent word list
  - User taps “of”



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

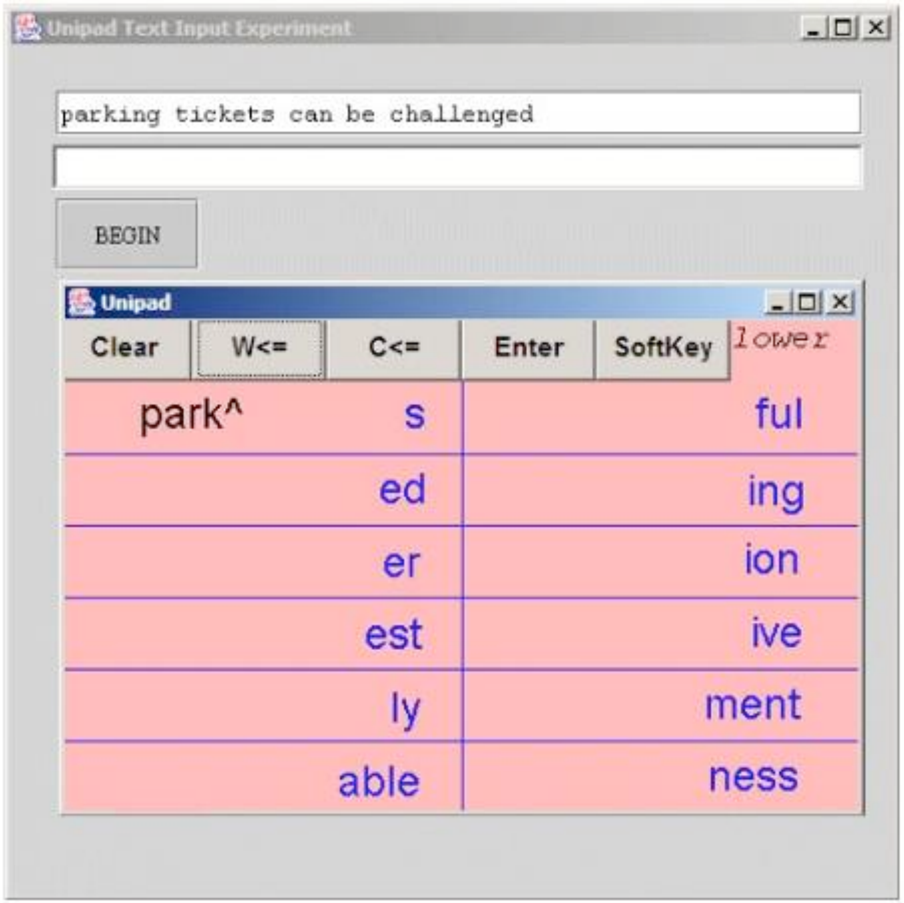
http://www.yorku.ca/mack/nordichi2006.html

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”

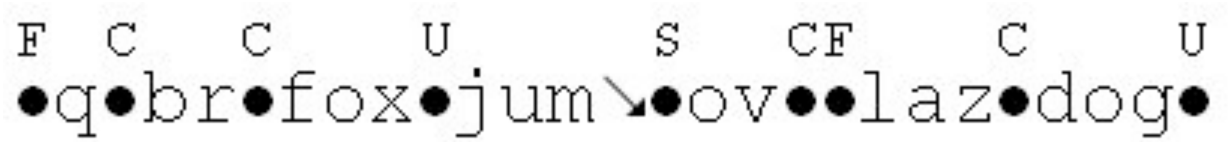


MacKenzie, Chen, Oniszczak: Unipad: Single-stroke text entry with language-based acceleration. NordiCHI 2006.

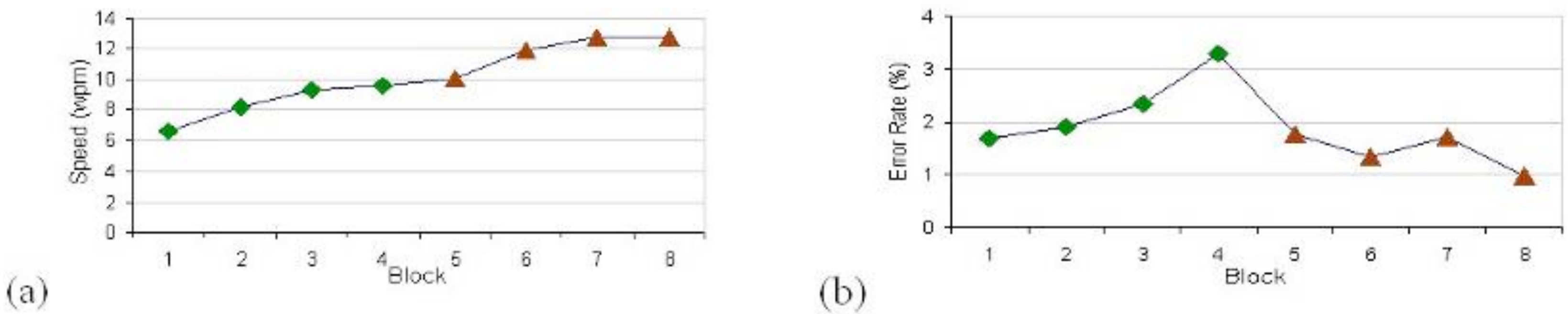
http://www.yorku.ca/mack/nordichi2006.html

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)

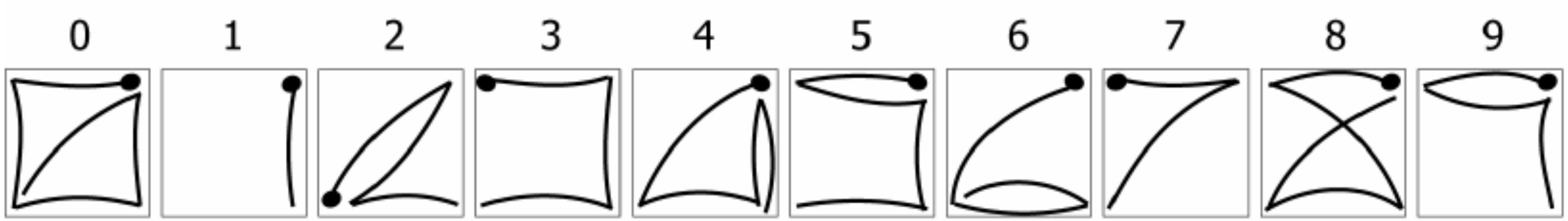


MacKenzie, Chen, Oniszczak: Unipad: Single-stroke text entry with language-based acceleration. NordiCHI 2006.

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

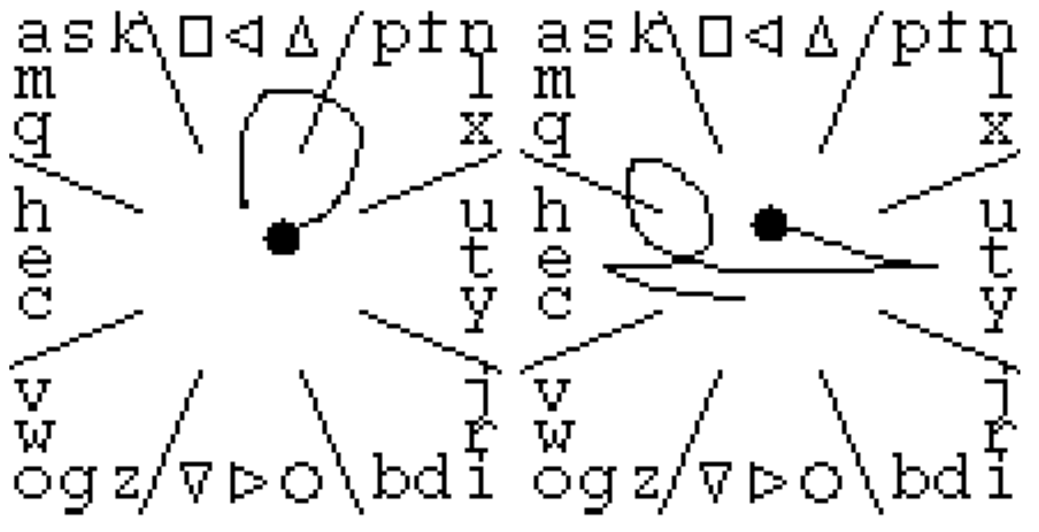


Wobbrock, Myers, Kembel: EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. UIST'03. http://depts.washington.edu/ewrite/

QuickWriting: Gesture-Based Input

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

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

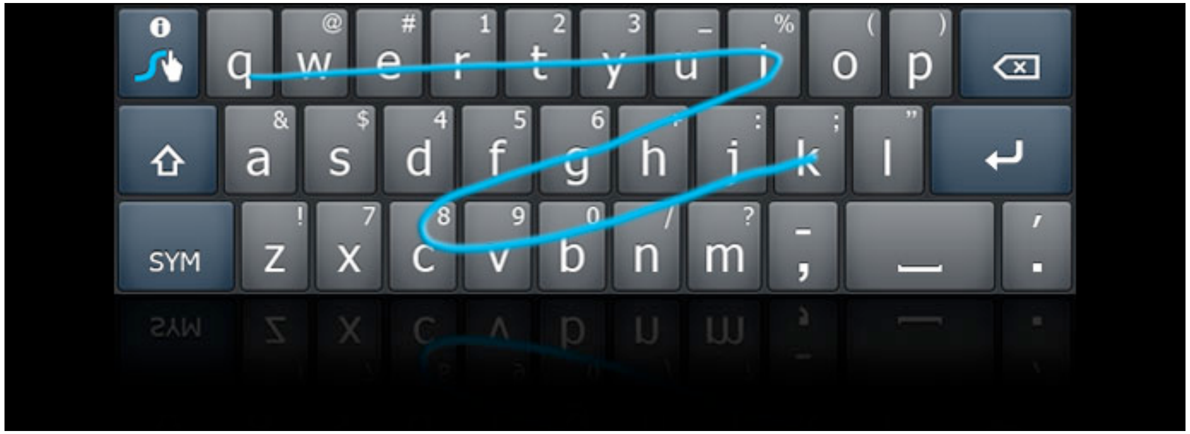


Writing the letter 'f', and the word "the"

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

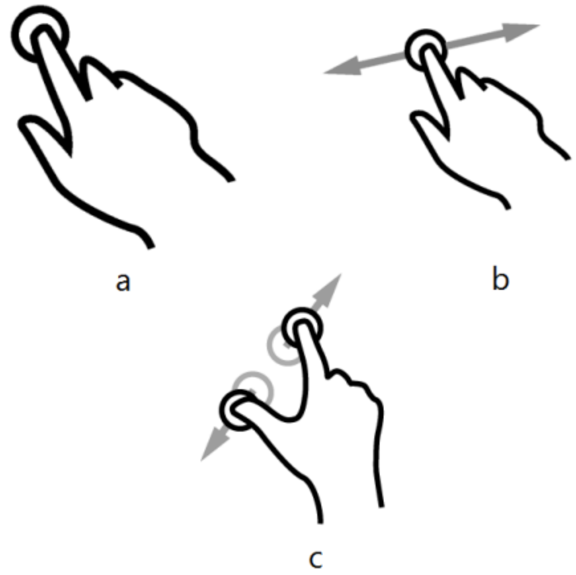
Swype

- 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

Touch Screen Gestures

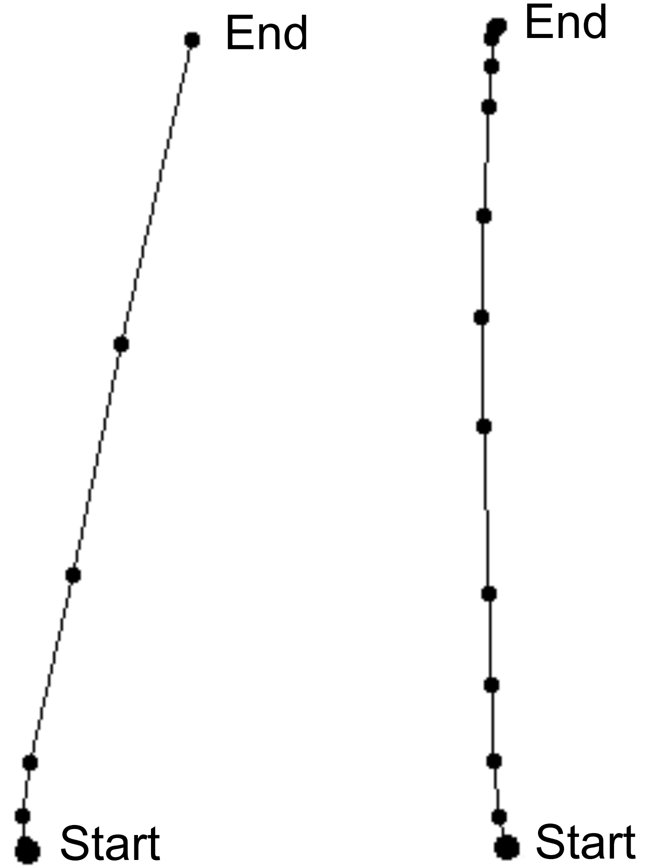


Source: GestureWorks.com

Which Gestures are These?

- Hint: one is "flick" and one is "drag"

- Relevant gesture parameters
  - Velocity profile
  - Shape
  - Direction



And this one?

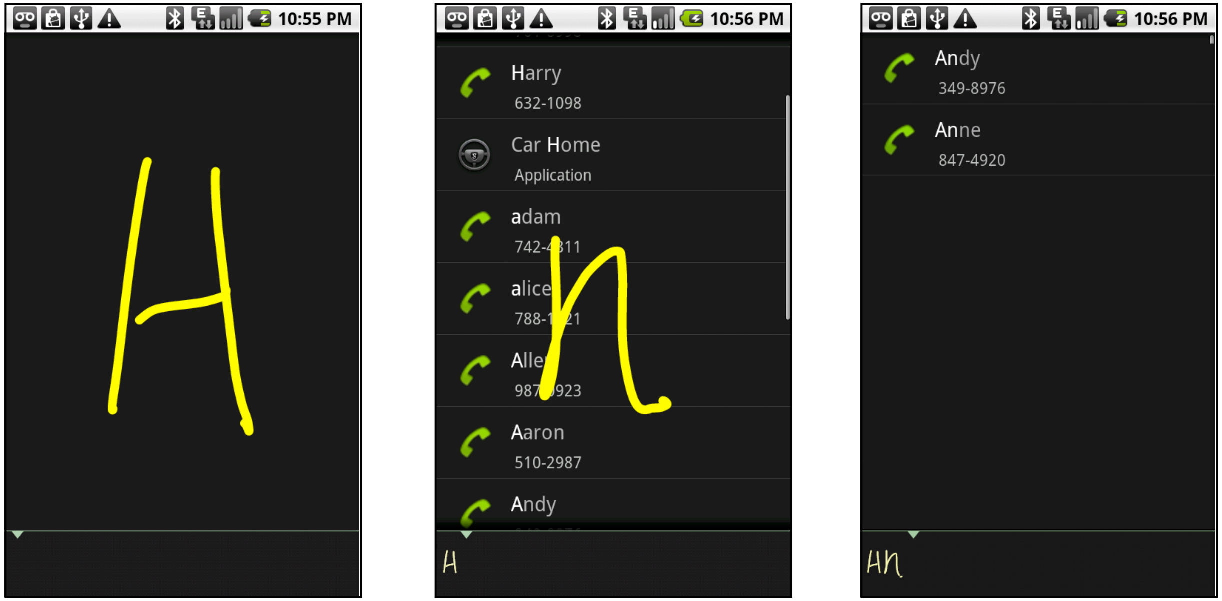
- Multi-touch pinch inwards
  - Typically mapped to “zoom out”
- Relevant gesture parameters
  - Number of touch points
  - Shape
  - Direction
- Challenge: finding intuitive mappings
  - Who should do this?
  - Developers? Designers? Users? Ergonomists?

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

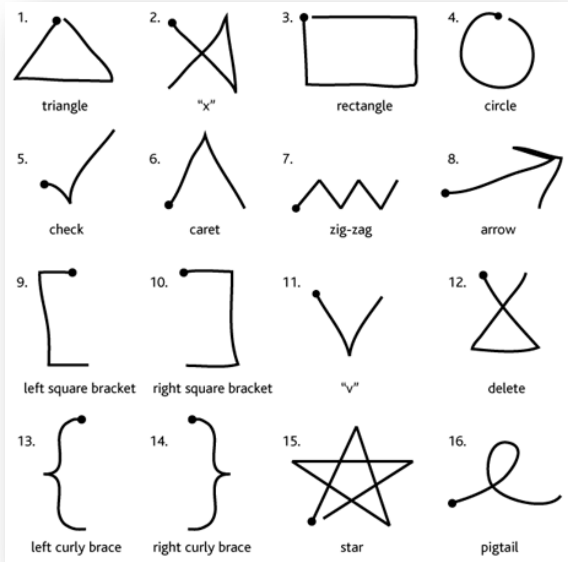
Example Application: Gesture Search

- 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



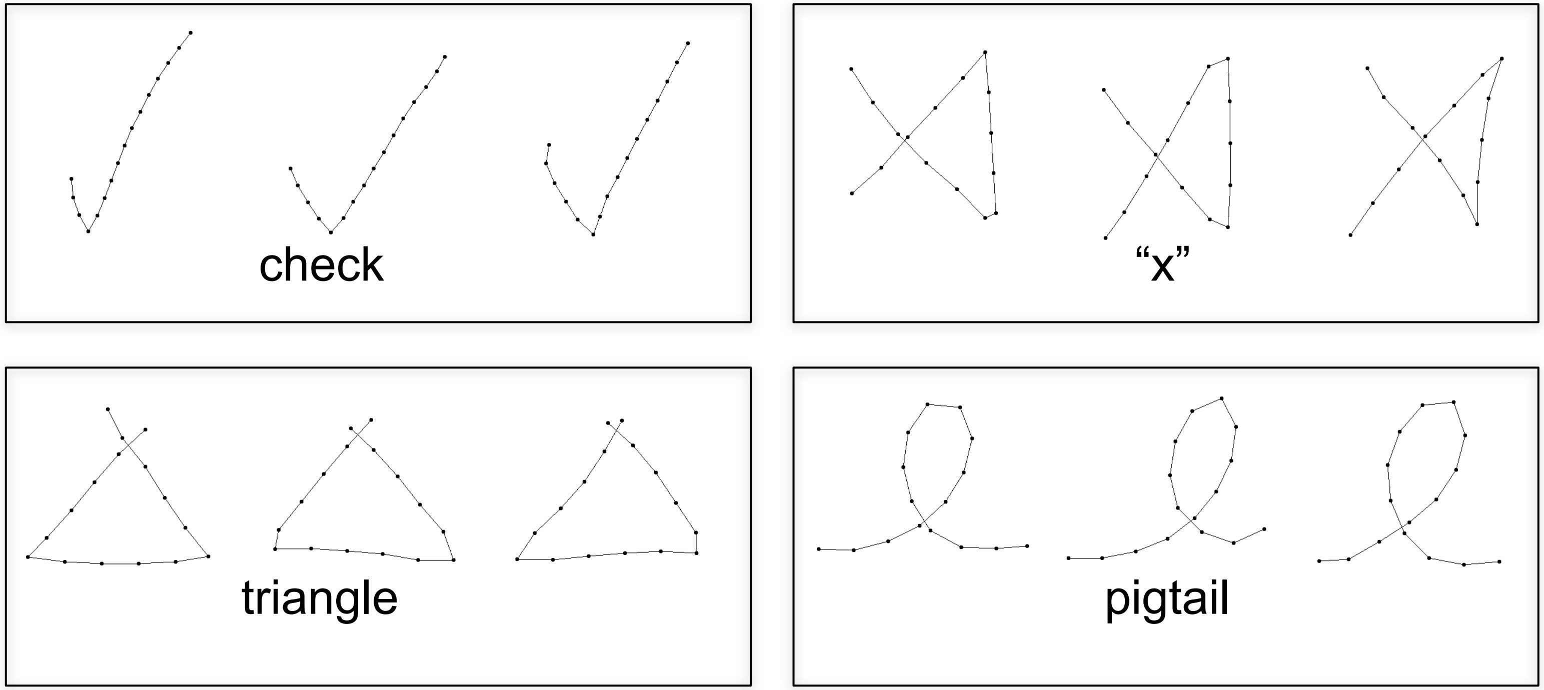
- Touch screens on many mobile devices
  - Mostly used for tapping (pointing tasks)
  - Suitable for swiping (crossing tasks)
  - Suitable for entering complex gestures
- Gesture recognition challenging
  - Pattern matching, machine learning
- Approaches for simple UI prototyping
  - $1 Recognizer
    - 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

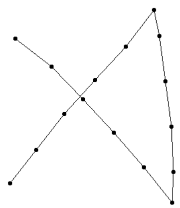
- Template-based recognizers
  - Template preserves shape and sequence of training gesture
  - Nearest neighbor approach
- Process
  - Store training samples as templates (multiple templates per gesture)
  - Compare unknown gesture against templates
  - Choose class of most similar template
- Advantages
  - 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)

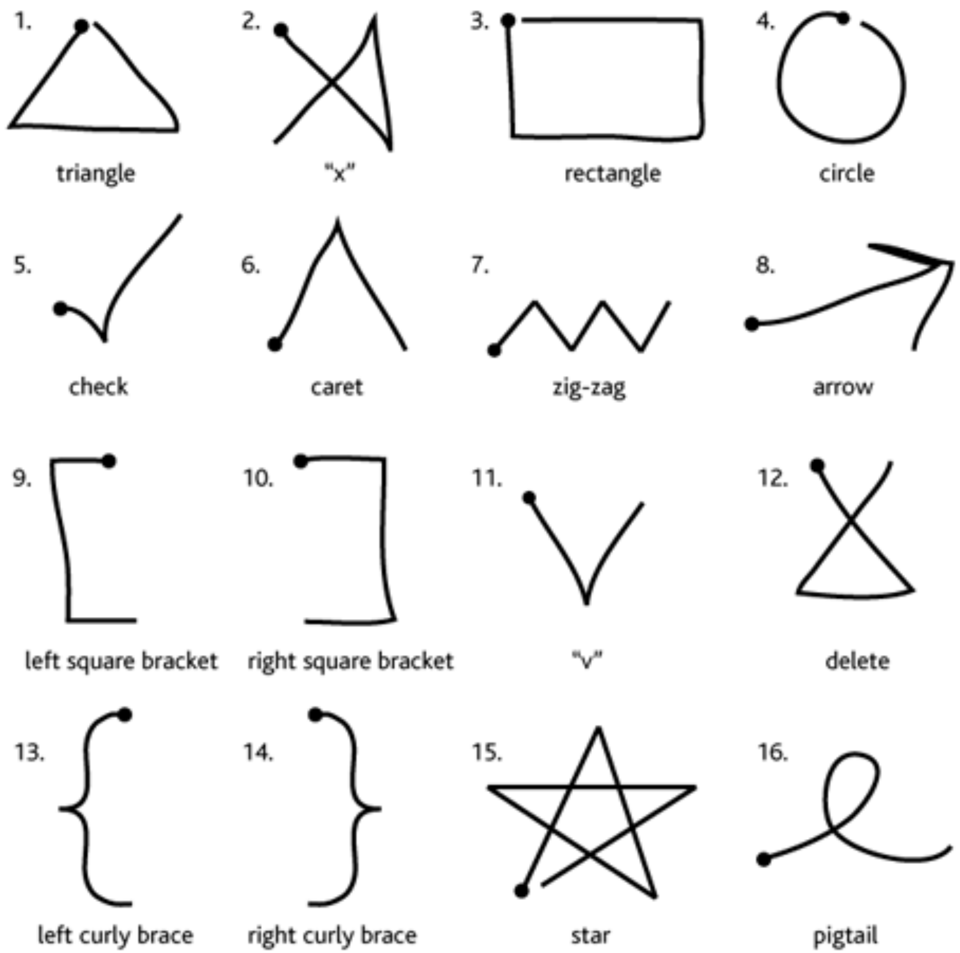


- Query gesture



Gesture Set of “$1 Recognizer”

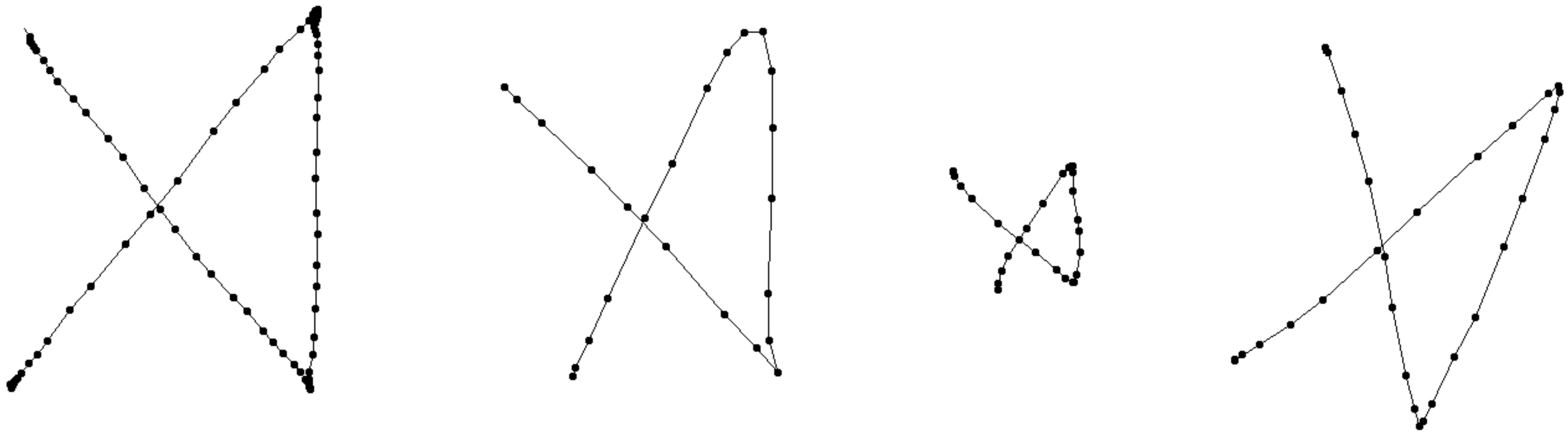
- Unistroke gestures (touch – move – release)
- Dot indicates start point



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

Variability in Raw Input

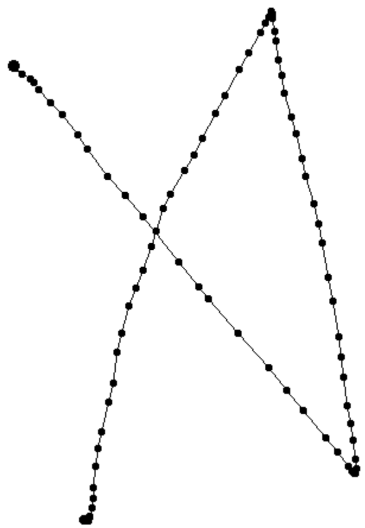
- Number and distribution of sample points depends on
  - Sampling rate
  - Movement speed and variability
  - Movement amplitude (scale)
  - Initial position and orientation



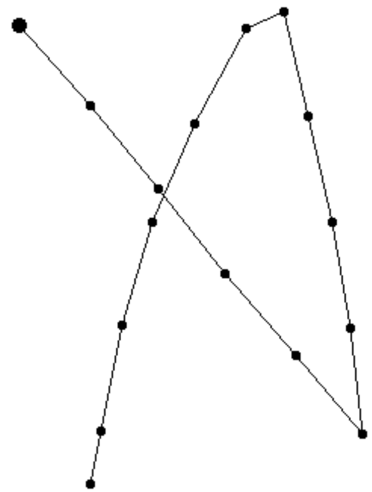
Slow Fast Small Rotated

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}: $v = x_1, y_1, x_2, y_2, \ldots, x_N, y_N$



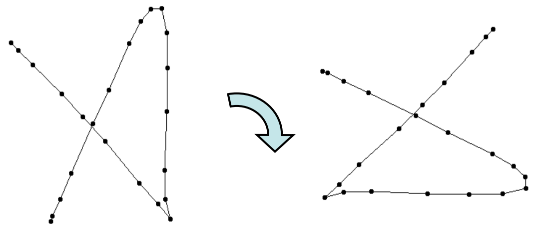
Original trace



Resampled (N = 16)

Gesture Recognition

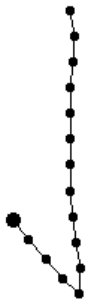
- Gesture recognition = search for most similar template
- Preprocessed query gesture g and templates t_j
  - Resampled (N=16), centroid translated to origin, normalized
- "Most similar" metric?
  - Sum of squared differences between points
    $\min_{j = 1..M} \{ \text{sum}_{i = 1..2N} \{ (g_i - t_{ji})^2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



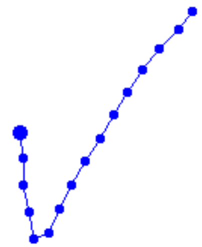
Optimal Angular Distance

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

check



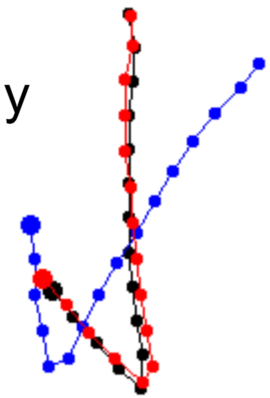
(resampled) query gesture



best-matching template

best-matching template optimally rotated to match query

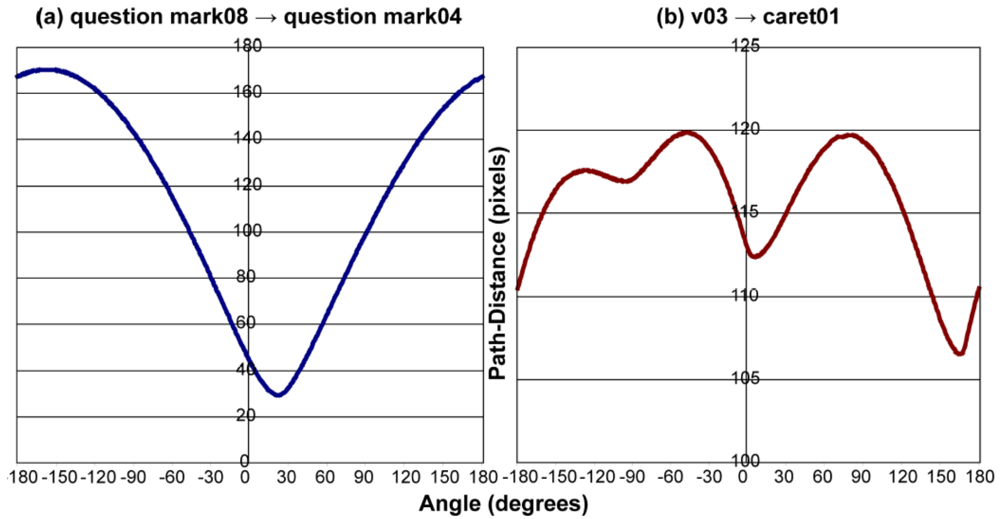
Overlaying query gesture (black) and optimally rotated best-matching template (red):



- How to find the optimal angle?

Finding the Optimal Angular Distance

- Wobbrock et al., UIST'07
  - "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!



Wobbrock et al., UIST'07

- 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

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

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

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

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

  Find extremum of f by deriving f w.r.t. θ and setting $f'(\theta) = 0$:

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

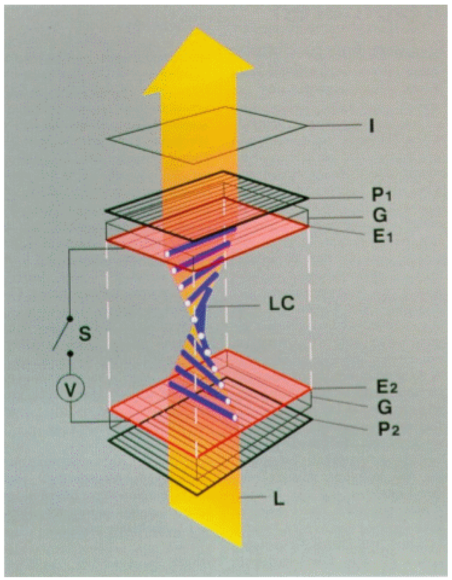
  $\Leftrightarrow \quad \sin\theta / \cos\theta = b / a = \tan\theta$

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

Display and Touch Screen Technologies

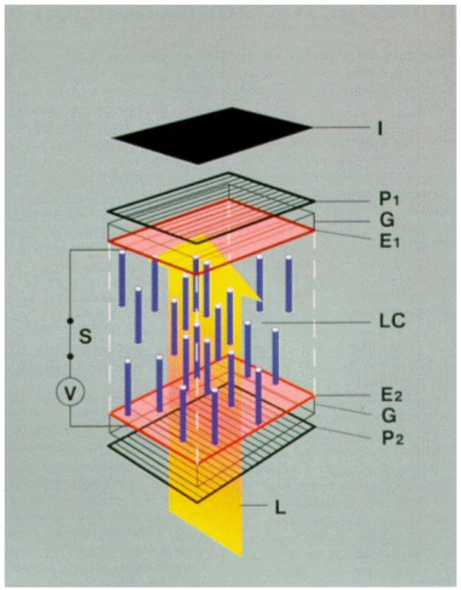
Liquid Crystal Display (LCD)

- An LCD cell is a voltage-controlled “light valve”
- Twisted nematic effect
  - Orientation of molecules controls orientation of polarized light
  - Off: Liquid crystal molecules form helix structure, 90° rotation
  - On: Electric field aligns molecules, second polarizer blocks light

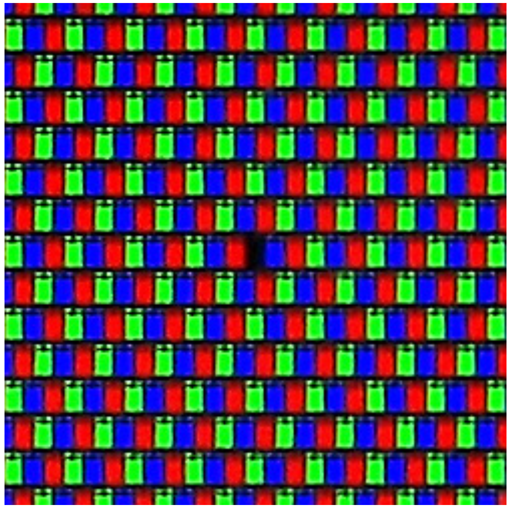


Off state

Source: Wikipedia

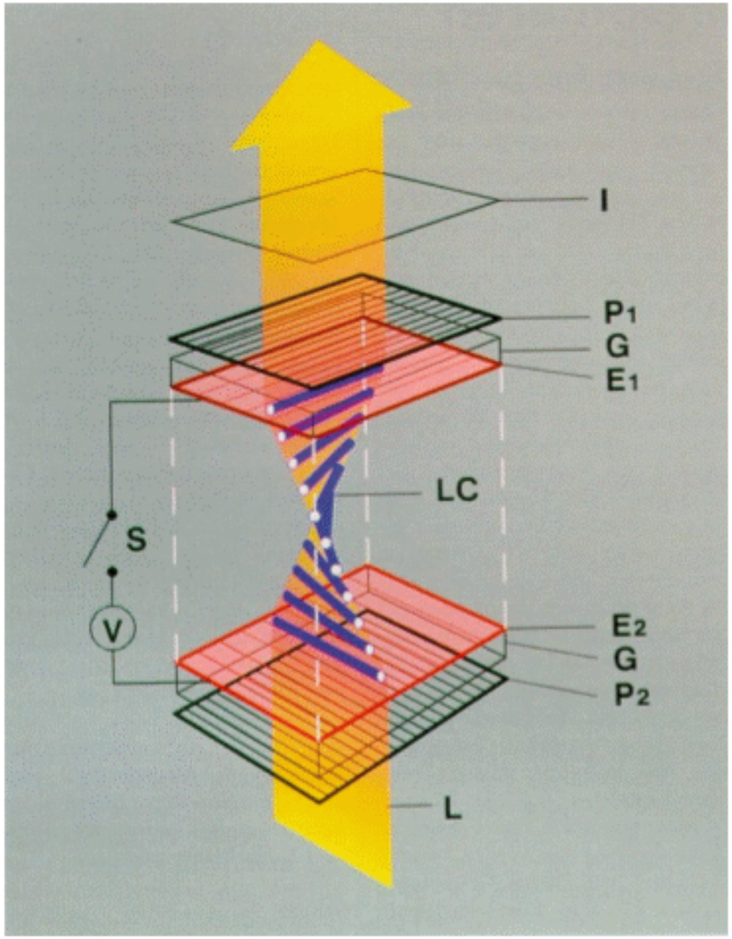


On state

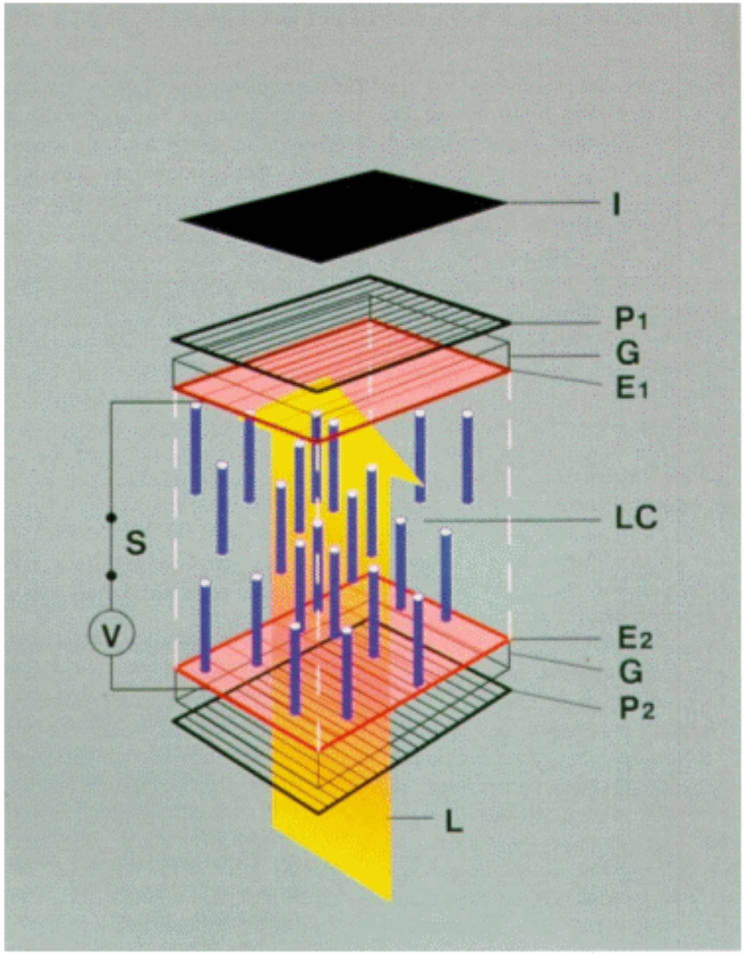


LCD pattern

Liquid Crystal Display (LCD)



Off state



On state

Source: Wikipedia

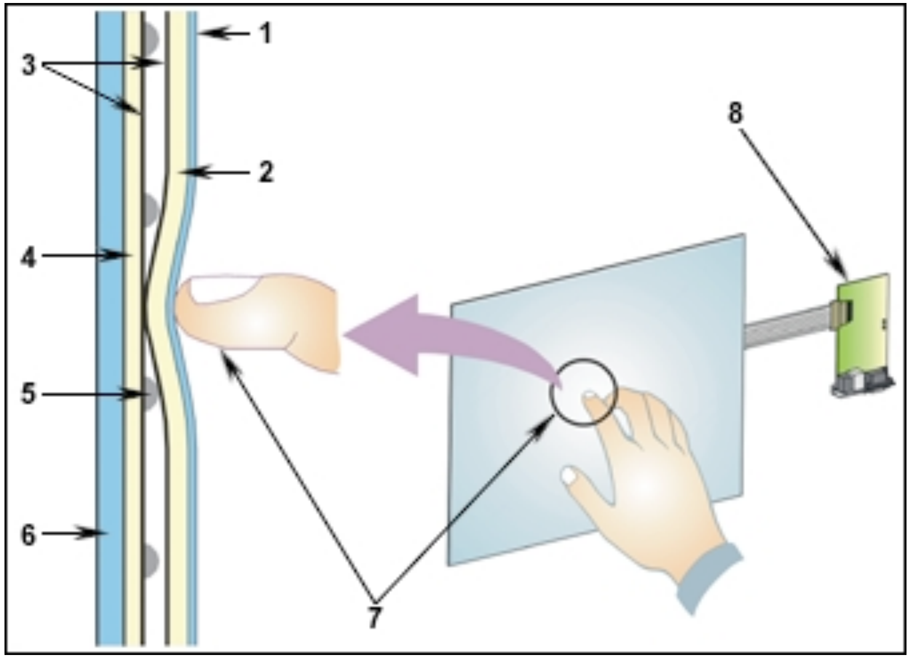
Liquid Crystal Display (LCD)

- Advantages
  - Low power consumption for controlling the twisted nematic effect
    - Low operating voltages (batteries)
    - Now current flow required
  - Cheap
  - Compact: light, small, low depth
  - Flicker free, sharp, undistorted image
- Disadvantages
  - Backlight illumination consumes significant amounts of power
  - Difficult manufacturing process (dead pixels, defective panels)
  - Fixed pixel resolution
  - Limited contrast and viewing angles (early LCDs)

Touch Screens

- Resistive
  - Suitable for stylus input
- Capacitive
  - Direct finger input, e.g., iPhone
- Surface Acoustic Wave (SAW)
  - Senses diffraction of waves on surface
- Frustrated Total Internal Reflection
  - Jeff Han’s multitouch table

Resistive Touch Screens

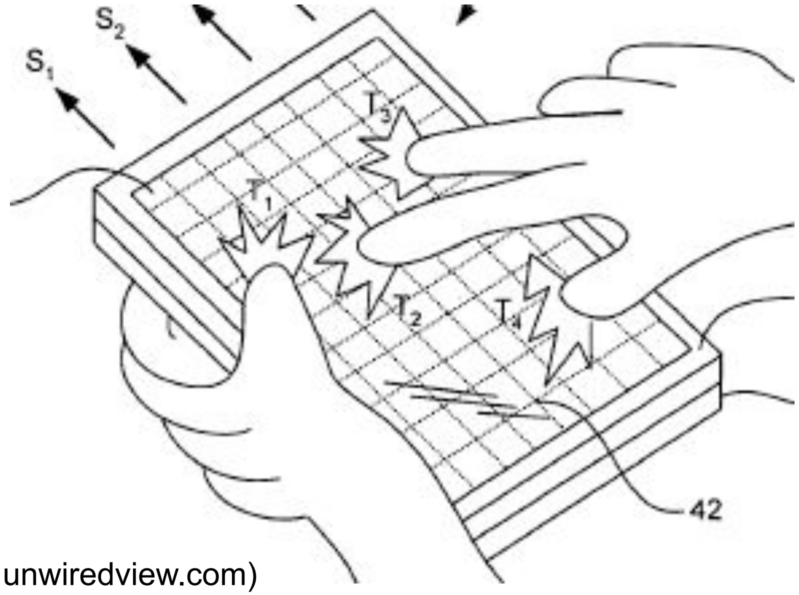


(www.fastpoint.com)

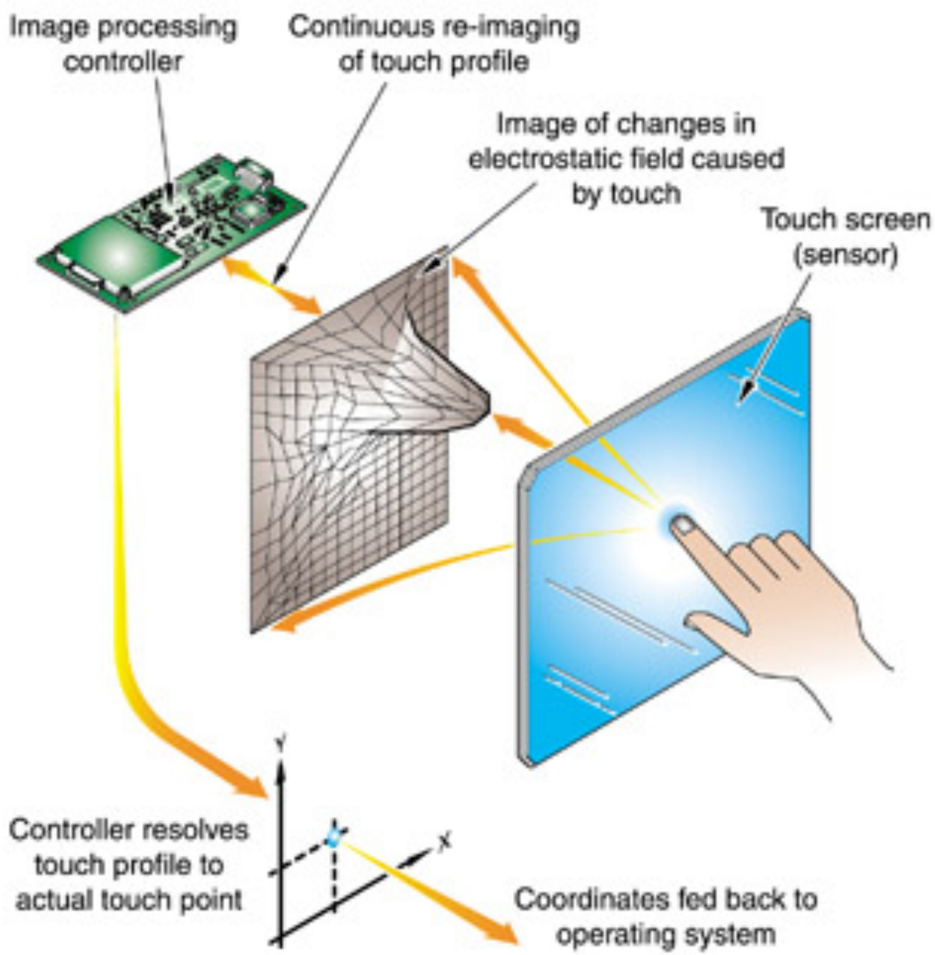
1. Polyester film
2. Upper resistive circuit layer
3. Conductive metal coating
4. Lower resistive circuit layer
5. Insulating dots
6. Glass/acrylic substrate
7. Touching the overlay surface causes (2) to touch (4), producing a circuit switch from the activated area
8. Touchscreen controller measures voltages through resistive layers and converts them into the digital X and Y coordinates of the activated area.

Capacitive Touch Screens

- Senses capacitive changes
  - Only works with finger, not with stylus
- iPhone
  - Uses additional grid for better multitouch disambiguation



(www.unwiredview.com)



Self-Capacitance Touch Screen

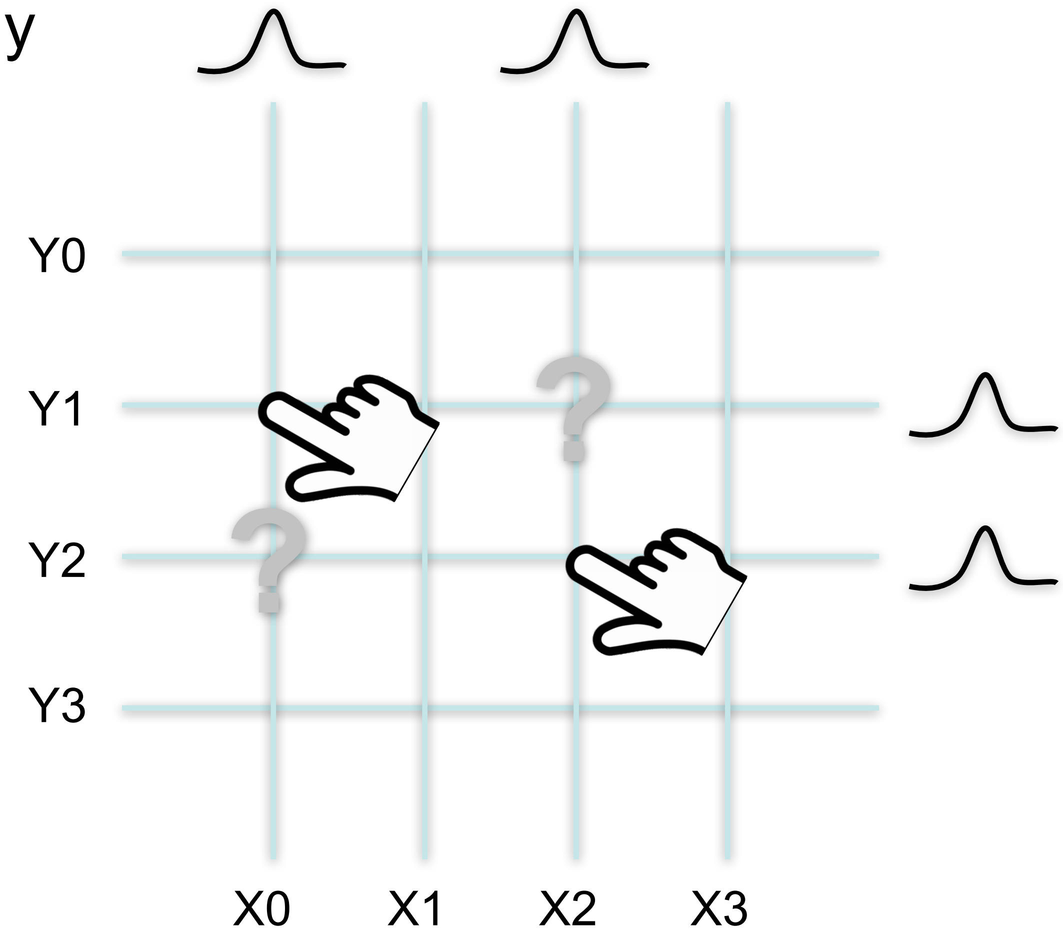
- Detects only a single touch point
- Measures capacitance of electrode to ground
- Finger near electrode: human body capacitance changes self-capacitance of electrode

- Materials: copper, indium tin oxide (ITO), printed ink
  - ITO: (almost) transparent capacitive electrodes
- Rows isolated from columns in grid arrangement
- Size and spacing between electrodes determines precision

Gary Barrett and Ryomei Omote: Projected-Capacitive Touch Technology. Information Display 3/10, pp. 16-21

Self-Capacitance Touch Screen

- Scans each electrode individually
- Sensing only



Mutual Capacitance Screen

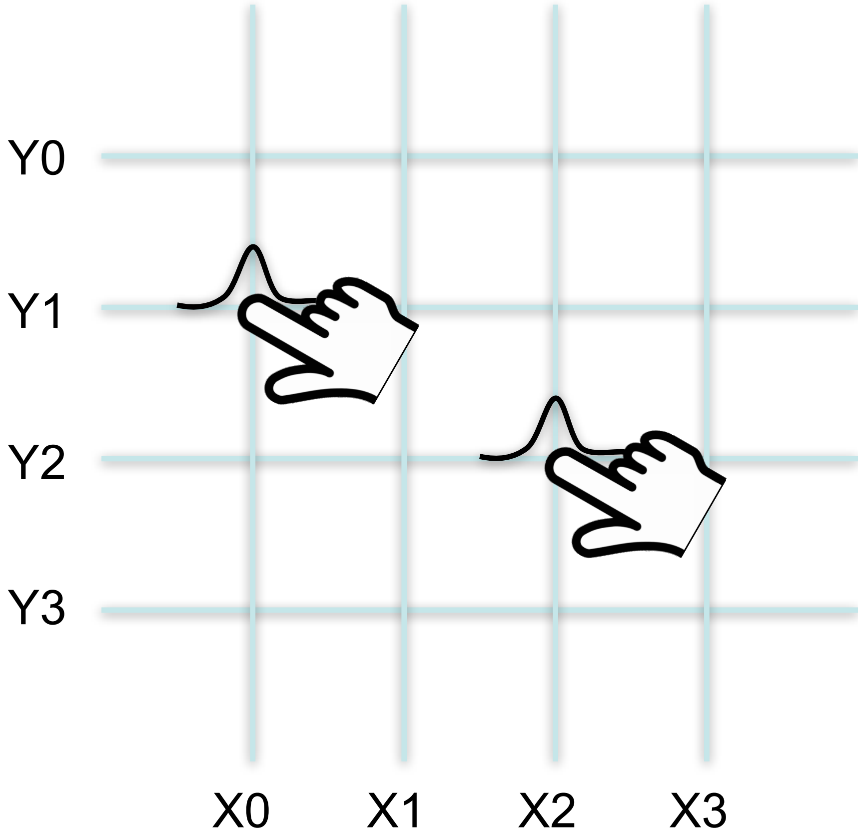
- Unlimited number of touch points
- Measures capacitance of intersections of electrodes
- Human-body capacitance changes capacitance of intersections ("steals" charge)

- High resolution
- Less sensitive to EMI than self-capacitance

- Typically 9 columns, 16 rows = 144 electrode intersections
- Interpolation achieves 1024x1024 (10 bit) resolution

Mutual Capacitance Screen

- Senses each pair (X_i, Y_j) of electrodes individually
- Driving and sensing



It’s Easy:

- http://mediathek.daserste.de/daserste/servlet/content/6099692

Pico Projectors

- Standalone or integrated in mobile phones
- Interesting for collaborative applications
  - Example: sharing media
- Problems (current technology)
  - Availability of projection space
  - Ambient light
  - Power consumption
  - Focusing

Audio and Haptics

Partly based on slides by Stephen Brewster:
Haptics, audio output and sensor input in mobile HCI. Stephen Brewster. Tutorial at Mobile HCI 2008.

Multimodality

- Involve different senses through different modalities
  - Audio, tactile
  - Suit different users, tasks, and contexts
- Problems of visual modality
  - Screen space small
  - Eyes heavily used when mobile
- Reasons for multimodality
  - Sole use of one modality not effective
  - Particular modality may not always be available all of the time
  - User involved in other tasks → Attention may be occupied

Multimodal Interaction

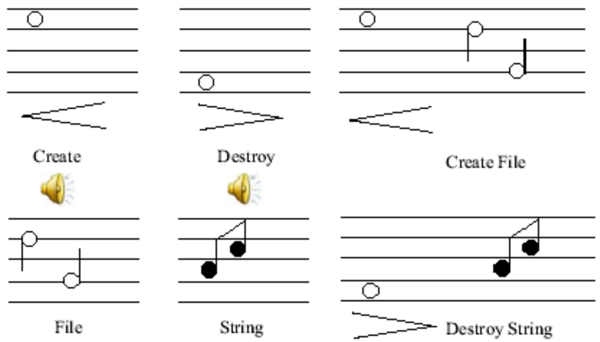
- Allow people to do everyday tasks while using mobile technology
  - “Eyes-free” or “hands-free”
- Interaction techniques that suit real environments
  - Non-speech audio and tactile feedback
  - Sensors for gestural input
  - Speech input

Non-Speech Audio

- Earcons (Blattner)
  – Musically structured sounds (abstract)
- Auditory Icons (Gaver)
  – Natural, everyday sounds (representational)
- Sonification
  – Visualization using sound
  – Mapping data parameters to audio parameters (abstract)

Earcons

- Structured audio messages based on abstract sounds
  – Created by manipulation of sound properties: timbre, rhythm, pitch, tempo, spatial location (stereo, 3D sound), etc.
- Composed of motives
- Can be compound
  – Sub-units combined to make messages
- Or hierarchical
  – Sounds manipulated to make complex structures



Open Close Undo

Examples from: http://www.dcs.gla.ac.uk/~stephen/earconexperiment1/earcon_expts_1.shtml

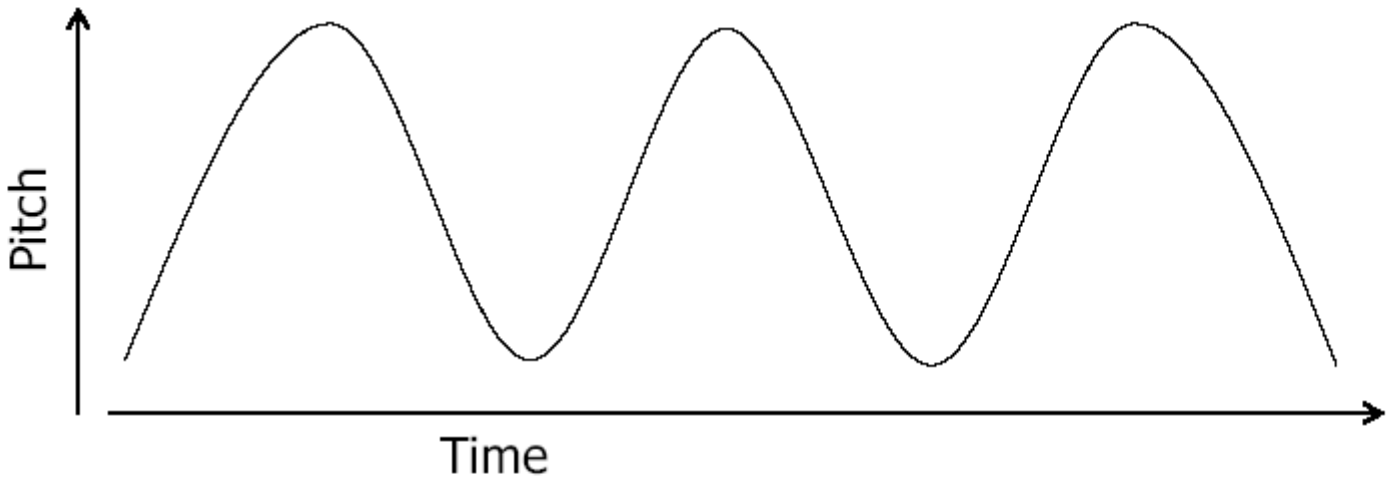
Auditory Icons

- Sounds mapped to interface events by analogy to everyday sound-producing events
  - E.g., selecting — tapping; copying — filling
  - Iconic v. symbolic mapping
- Auditory icons can be parameterized
  - E.g. material for type, loudness for size
  - Multiple layers of information in single sounds
  - Reduces repetition and annoyance
- The SonicFinder
  - Selecting, copying, dragging

Gaver, W. W., (1986). Auditory icons: Using sound in computer interfaces. Human-Computer Interaction, 2, 167-177.

Sonification

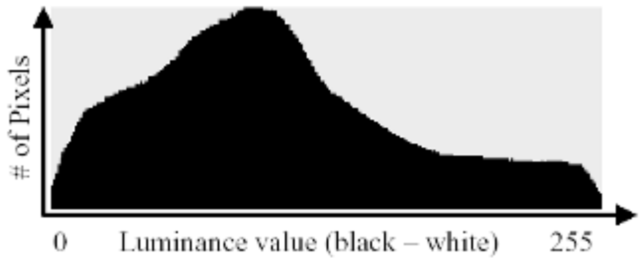
- Mapping of data values to auditory parameters
- Most commonly x-axis to time, y-axis to pitch

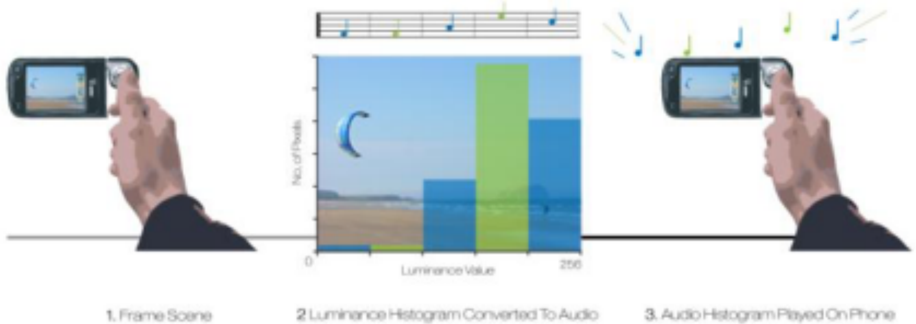


Source: Haptics, audio output and sensor input in mobile HCI by Stephen Brewster. Tutorial Mobile HCI 2008.

Sonification of Luminance Histograms in Digital Cameras

- Difficult to focus visual attention on subject and technical parameters
  – Exposure, aperture, battery life, image mode, etc.
- Idea: Sonified luminance histogram





- Sonification of remaining memory space

Brewster and Johnston: Multimodal Interfaces for Camera Phones. MobileHCI 2008.

3D Audio Interaction

- Increase the audio display space
- 3D audio
  - "Cocktail party effect"
  - Provides larger display area
  - Monitor more sound sources
- "Audio Windows" (Cohen)
  - Each application gets its own part of the audio space
- Pie Menus (Brewster, CHI'03, Marentakis, CHI'06)
  - Audio items placed around the head

Cohen. Integrating graphics and audio windows. Presence: Teleoper. Virtual Environ. 1, 4 (Oct. 1992), 468-481.

Brewster, Lumsden, Bell, Hall, Tasker. Multimodal 'eyes-free' interaction techniques for wearable devices. CHI '03.

Marentakis, Brewster. Effects of feedback, mobility and index of difficulty on deictic spatial audio target acquisition in the horizontal plane. CHI '06.

Source: Haptics, audio output and sensor input in mobile HCI by Stephen Brewster. Tutorial Mobile HCI 2008.

Haptics

- Definition: Sense and/or motor activity based in the skin, muscles, joints, and tendons
- Two parts
  - Kinaesthesis: Sense and motor activity based in the muscles, joints, and tendons
  - Touch: Sense based on receptors in the skin
    - Tactile: mechanical simulation of the skin

Why Haptic Interaction?

- Has benefits over visual display
  - Eyes-free
- Has benefits over audio display
  - Personal not public
  - Only the receiver knows there has been a message

- People have a tactile display with them all the time
  - Mobile phone

Source: Haptics, audio output and sensor input in mobile HCI by Stephen Brewster. Tutorial Mobile HCI 2008.

Tactile Technologies

PMD 310 vibration motor

- Vibration motor with asymmetric weight
  - Eccentricity induces vibrations
  - Speed controls vibration frequency
  - Precision limited (several ms startup time)

Tactaid VBW32 actuator

Phone vibration motor

C2 Tactor actuator

Actuators now in other kinds of devices

3 cell pin array

Source: Haptics, audio output and sensor input in mobile HCI by Stephen BrewsterTutorial Mobile HCI 2008.

Design of Tactons

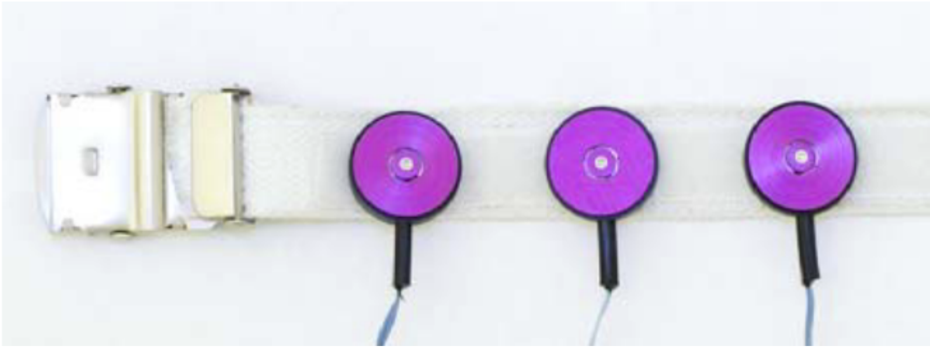
- Tactons = tactile icons
  - Structured, abstract messages that can be used to communicate non-visually (Brown, 2005)
  - Tactile equivalent to Earcons

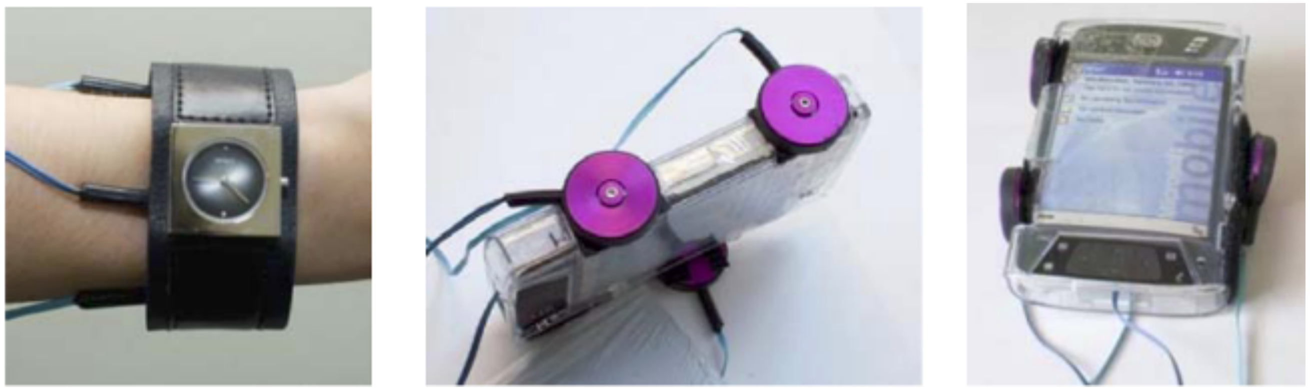
- Encode information using parameters of cutaneous perception
  - Body location
  - Rhythm
  - Duration
  - Waveform
  - Intensity

Source: Haptics, audio output and sensor input in mobile HCI by Stephen Brewster. Tutorial Mobile HCI 2008.

Tacton Parameters

- Spatial location (on forearm, waist, hand) very effective
  - Good performance with up to 4 locations
  - Wrist and ankle less effective, especially mobile





Source: Haptics, audio output and sensor input in mobile HCI by Stephen Brewster. Tutorial Mobile HCI 2008.

Tacton Parameters

- Rhythm very effective
  - Easily identified with three levels
- Waveform
  - Carefully designed sine, square, and sawtooth wave forms very effective (tuned to capabilities of actuator)
- Intensity
  - Two levels
  - Hard to use and may need to be controlled by user

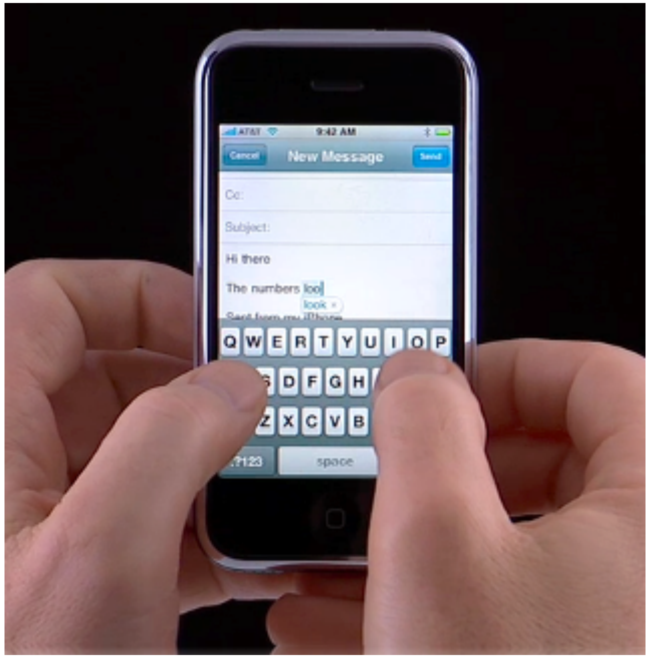
Source: Haptics, audio output and sensor input in mobile HCI by Stephen Brewster. Tutorial Mobile HCI 2008.

Example: Tactile Button Feedback

- Touchscreen phones have no tactile feedback for buttons
  - More errors typing text and numbers

- Performance comparison of physical buttons, touchscreen, and touchscreen+tactile
  - In lab and on subway

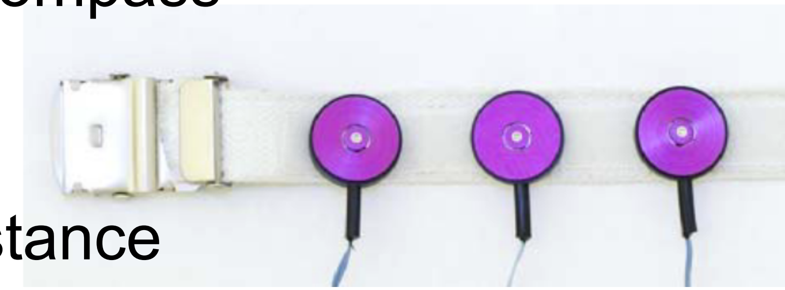
- Touchscreen+tactile as good as physical buttons
  - Touchscreen alone was poor



Brewster, Chohan, Brown: Tactile feedback for mobile interactions. CHI '07.

Example: Tactile Navigation

- Non-visual interface for GPS + compass
- Belt of 4 actuators
  - Placed north, south, east, west
- Vibrations gave direction and distance



- Users could follow paths accurately without a screen

Source: Haptics, audio output and sensor input in mobile HCI by Stephen Brewster. Tutorial Mobile HCI 2008.

The End