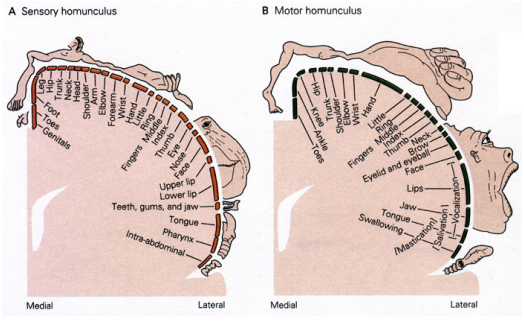
COMS W4170
Interaction Devices 2

Steven Feiner
Department of Computer Science
Columbia University
New York, NY 10027

October 30, 2018

Interaction Device Categorization

- Body part used
  - hand, head, eyes,…
- Type of motion
  - linear, rotary,…
- Property sensed
  - position, angle, force, torque,…



W. Penfield's homunculus

Interaction Device Categorization

- Physical resistance
  - *Isotonic* devices
    - Zero (or constant) resistance: Device moves with limb
      - Measures position, angle
      - E.g., mouse, isotonic joystick

  - *Isometric* devices
    - Infinite resistance: Device does not move
      - Measures force, torque
      - E.g., isometric joystick

3

Interaction Device Categorization

- Physical resistance (cont.)
  - Points on the isotonic ←→ isometric continuum
    - Many devices fall on a continuum (e.g., TrackPoint moves slightly)
    - *Elastic* devices
      - Finite resistance: Increases with distance, so device movement is opposed by restoring force
      - E.g., spring-loaded joystick

  - Also, *viscous* (resistance increases with velocity), *inertial* (resistance increases with acceleration)
  - Isometric and elastic devices are self-centering
    - Can return automatically to "zero" position

4

Interaction Device Categorization

- Coordinate system
  - *Absolute*
    - E.g., data tablet
  - *Relative*
    - E.g., mouse
    - "Clutching" [as in an automotive clutch]
      - Relative devices rely on clutching (engaging) & declutching (disengaging) the device to extend their range
        - Picking up (declutching) a mouse to move it to the side of the table opposite to the direction of motion before putting it down (clutching) again
        - Releasing your grip (declutching) on a rotary control before twisting the hand and gripping (clutching) it again
      - Need less space
      - Can use more comfortable space/pose

Note: Some folks confusingly use "clutching" in the opposite sense to refer to "declutching" (and do not use the term "declutching" at all), thinking of "clutching" as "grabbing to pick up" (disengaging the device)

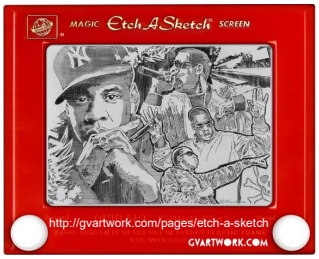
Others use "reclutching" to refer to the process of disengaging and engaging

Interaction Device Categorization

- Granularity
  - *discrete* vs. *continuous*
- Resolution
  - Smallest change that can be measured
- Accuracy
  - Closeness to actual value
- Precision
  - Reproducibility
- Type of control
  - *direct* vs. *indirect*
    - Whether device is [= *direct*] or is not [= *indirect*] at the same location as what it controls in the user interface
- Number of DOF (Degrees of Freedom) [aka Dimensions]
  - 1/2/3 linear, 1/2/3 angular,…

Interaction Device Categorization Coordination of DOF

- *Separable* vs. *integral*
- Example (after W. Buxton)
  - Etch A Sketch
    - 2 *separable* positional DOF
      - separate control knobs
    - Good for vertical/horizontal lines
  - Skedoodle
    - 2 *integral* positional DOF
      - integrated control joystick
    - Good for free-form doodles
    - Templates provide constrained control



7

Stimulus–Response (S–R) Compatibility

- Whether mapping of user stimulus (e.g., limb motion) to system response (e.g., cursor motion) is compatible with regard to
  - Direction
  - Orientation
  - …

Compare orienting your mouse

like *this* vs. like *this*

8

Control-to-Display (C/D) Ratio

[more generally, Control-to-Response (C/R) Ratio]

- Ratio of movement of input device (user stimulus) to movement of controlled display object (system response) for multiplicative *transfer function*

  Alternatively, C–D gain = $(\text{C/D ratio})^{-1}$

  - High C/D ratio → accuracy
  - Low C/D ratio → speed, space savings

Notation varies! You will see {CD, C–D, C:D, C/D} {ratio, gain}

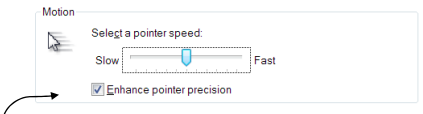
Control-to-Display (C/D) Ratio

[more generally, Control-to-Response (C/R) Ratio]

- Variable C/D ratio
  - E.g., typical mouse transfer function

    high-speed → low C/D ratio

    low-speed → high C/D ratio

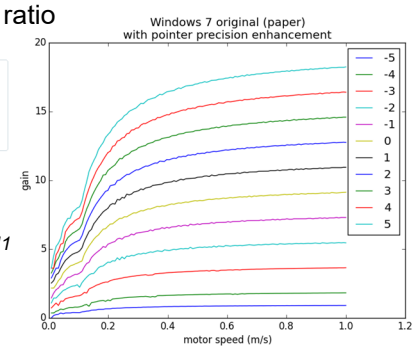


Mouse Properties: Enables variable C/D ratio in Windows (on by default)

G. Casiez and N. Roussel, *UIST 2011*

http://libpointing.org

Note: Graph shows C–D gain, not C/D ratio



An Early GUI Framework: Interaction Tasks *J. Foley et al., 80s*

- *Basic interaction task* is the input by user of atomic "unit" of info
  - Text
  - Select
  - Position
  - Quantify
  - Orient
  - Path

An Early GUI Framework: Interaction Tasks *J. Foley et al., 80s*

- *Basic interaction task* is accomplished by
  - *basic interaction technique*
    - Approach for performing basic interaction task
  - using *logical device*
    - Abstraction of a device based only on its output
      - String, choice, pick, locator, valuator, stroke
  - implemented by *physical device(s)*
    - Actual device with characteristic properties
      - Keyboard, mouse, rotary control, linear slider, joystick, touch screen, touch pad, scroll wheel, buttons,…

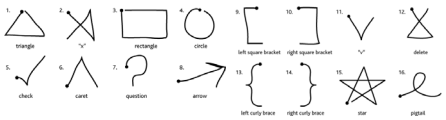
An Early GUI Framework: Interaction Tasks *J. Foley et al., 80s*

- *Composite interaction task* combines basic interaction tasks
  - Dialogue boxes
    - Specify multiple units of information
  - Construction
    - Create objects
  - Manipulation
    - Modify objects (e.g., by scaling)

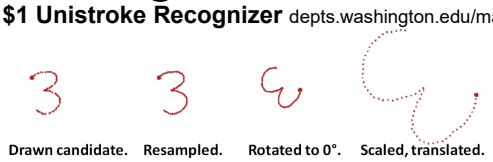
An Early GUI Framework: Interaction Tasks *J. Foley et al., 80s*

- Problem: The notion of a “basic interaction task” accomplished by a single logical device is an oversimplification
- Best suited to “classical” devices processed by simple computation
- What about
  - Multi-touch?
  - Gesture?
  - Full-body?
  - Face?
  - Eyes?
  - Pulse?
  - Fingerprint?
  - Emotion?

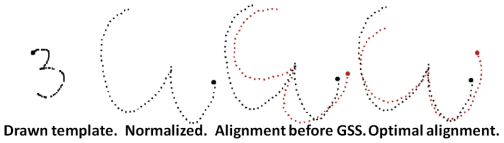
Simple Gesture Recognition



$1 Unistroke Recognizer depts.washington.edu/madlab/proj/dollar/



Drawn candidate. Resampled. Rotated to 0°. Scaled, translated.



Drawn template. Normalized. Alignment before GSS. Optimal alignment.

- J. Wobbrock, A. Wilson, Y. Li, $1 Unistroke Recognizer, *UIST 2007*
  - depts.washington.edu/madlab/proj/dollar/
- L. Anthony & J. Wobbrock, $N Multistroke Recognizer, *GI 2010*
  - depts.washington.edu/madlab/proj/dollar/ndollar.html
- Y. Li, Protractor, *CHI 2010*
  - yangl.org/protractor (More efficient template matching)
- R-D. Vatavu, L. Anthony, J. Wobbrock, $P Point Cloud Recognizer, *ICMI 2012*
  - depts.washington.edu/madlab/proj/dollar/pdollar.html (Independent of # of strokes)

- For each *T* (template) and each *C* (candidate to match)

Normalization:
- Resample point path to *N* equidistant points
- Rotate "indicative angle" (vector from centroid to first point) to 0°
- Scale to reference square and translate centroid to origin

- For each *C* (additional steps)

Matching:
- Refine rotation to determine (heuristically!) minimum distance between corresponding points of *C* and each *T* (GSS = Golden Section Search)
- Select *T* with smallest distance (best match)

15