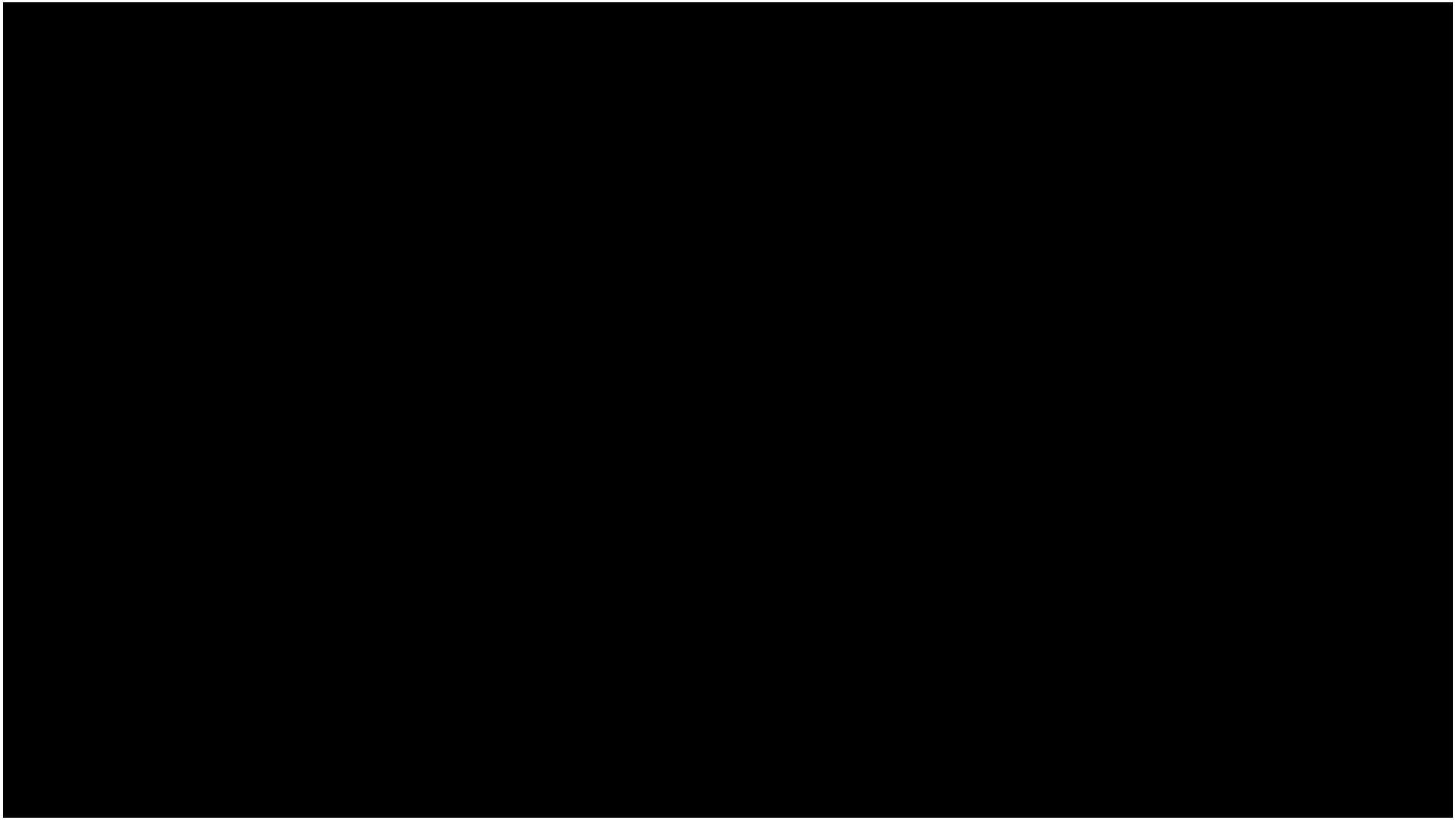
INPUT DEVICES

31 SEPT 2015

ERIC PAULOS
www.paulos.net

UNIVERSITY OF CALIFORNIA Berkeley

PYRO WATCH



Suggested by Casey Rogers

Suggested by Richard Jiang

3D TOUCH

Suggested by Eric Pai

Glimpse: a Novel Input Model for Multi-level Devices

CHI 2005

Clifton Forlines, Chia Shen
Mitsubishi Electric Research Labs (MERL)
201 Broadway, Cambridge MA 02139 USA
{forlines, shen}@merl.com

Bill Buxton
Buxton Design
888 Queen St. East, Toronto
Ontario Canada, M4M 1J3
bill@billbuxton.com

ABSTRACT
We describe a technique that supports the previewing of navigation, exploration, and editing operations by providing convenient *Undo* for unsuccessful and/or undesirable actions on multi-level input devices such as touch screens and pen-based computers. By adding a *Glimpse* state to traditional three-state pressure sensitive input devices, users are able to preview the effects of their editing without committing to them. From this *Glimpse* state, users can undo their action as easily as they can commit to it, making *Glimpse* most appropriate for systems in which the user is likely to try out many variations of an edit before finding the right one. Exploration is encouraged as the cumbersome returning to a menu or keyboard to issue an *Undo* command is eliminated. *Glimpse* has the added benefits that the negative effects of inconsistencies in the *Undo* feature within an application are reduced.

Author Keywords
Pressure Sensitive Input, Undo, Direct Manipulation, Three-State Input, Touch Screens, Stylus, Navigation

ACM Classification Keywords
H.5.2.h Information interfaces and presentation (e.g., HCI): User Interfaces - Input devices and strategies

H.5.2.i Information interfaces and presentation (e.g., HCI): User Interfaces – Interaction Styles

INTRODUCTION
Undo is a critical feature in many computer applications as it frees the user from the fear of experimenting with changes to the application's state. Creativity is enhanced when users are able to easily retract their changes if the result is unsatisfactory. People in creative industries will often state that the quality of their end product is a direct function of how many variations they *tried out and threw away* during the development of their product. Because of the frequency with which *Undo* is used, even small improvements to this feature can have a large positive effect. Additionally, many editing operations that take place

We propose a system-wide method of providing a *Glimpse* of the results of any operation completed with multi-level input devices. By multi-level, we mean that the input device is capable of sensing at least two levels of input (e.g. a stylus that senses light and heavy pressure or a mouse with a two-state button [5]) in addition to providing positional feedback. This positional feedback can be on-screen, in the case of an on-screen mouse pointer that tracks the movement of the mouse, or implicit, in the case of a finger or stylus with which the user operates directly on the display surface with graphical elements that are positioned directly underneath the input device. Our method has the added benefit that opting out of an action is as easy to perform as committing to it, making *Glimpse* most appropriate for systems in which the user is likely to try out many variations of an edit before deciding on any particular one.

BACKGROUND PART 1: PRESSURE SENSITIVE INPUT
Figure 1 shows Buxton's three-state model for stylus input [2]. In this model, light pressure input results in the

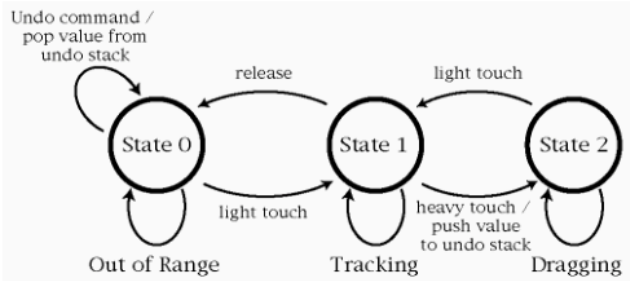


Figure 1. Buxton's three-state model for pressure sensitive input. Dragging an object results in its value being pushed onto the system undo stack. Restoring this saved value occurs after a separate command.

"tracking" of the input device (similar to moving the mouse while its button is up). While "tracking", a graphical pointer follows the movement of the input device on screen. Heavy pressure input results in "dragging" operations (similar to

CHI 2004

Pressure Marks
Gonzalo Ramos
Ravin Balakrishnan
Dynamic Graphics Project
University of Toronto

Pop Through Mouse Button Interactions

UIST 2001

Robert Zeleznik, Timothy Miller and Andrew Forsberg
Brown University
Department of Computer Science
Providence, RI 02912
(401) 863-7653; {bcz,tsm,asf}@cs.brown.edu

ABSTRACT
We present a range of novel interactions enabled by a simple modification in the design of a computer mouse. By converting each mouse button to *pop through* tactile pushbuttons, similar to the focus/shutter-release buttons used in many cameras, users can feel, and the computer can sense, two distinct "clicks" corresponding to pressing lightly and pressing firmly to pop through. Despite the prototypical status of our hardware and software implementations, our current pop through mouse interactions are compelling and warrant further investigation. In particular, we demonstrate that pop through buttons not only yield an additional button activation state that is composable with, or even preferable to, techniques such as double-clicking, but also can endow a qualitatively novel user experience when meaningfully and consistently applied. We propose a number of software guidelines that may provide a consistent, systemic benefit; for example, light pressure may invoke default interaction (short menu), and firm pressure may supply more detail (long menu).

KEYWORDS: mouse, double-action, gesture, interaction, click through, buttons, pop through, input devices, haptics.

INTRODUCTION
Inspired both by the fluid interactivity of the camera button mechanism and the observation that people often press elevator buttons harder as they grow impatient, we considered a number of approaches for integrating pressure controls into computer interfaces. The choice we present, replacing mouse buttons with pop through pushbuttons (see Figure 1), increases the information bandwidth of a mouse, but we hypothesize may well be easier for users to learn, remember, and control than other choices, such as adding spatially distinct or continuously pressure-sensitive buttons. Moreover, our device is easy to deploy because it does not require visible changes to a commercial mouse, and legacy applications can be compatible simply by ignoring the added button state.

After developing a collection of novel interaction techniques to demonstrate the potential of pop through mouse interaction, we performed a patent seach that uncovered a 1994 patent on a substantially similar device by Apple Computer [1]. Interestingly, the patent does not consider any of the interaction techniques that we present.

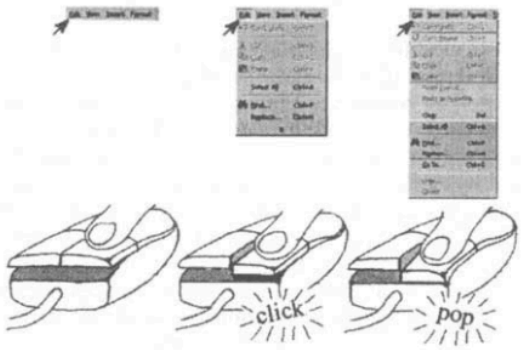


Figure 1: Pressing the button lightly clicks and in this example invokes a short menu; pressing harder pops through with a second click, in this example invoking a longer menu. (Button displacement exaggerated.)

PROTOTYPE HARDWARE AND SOFTWARE
Hardware Design For the first of two different pop through mouse prototypes, we glued a small pushbutton on top of a mouse button and connected it to an RS232 port for software polling.[1] Because of the differing activation forces of the buttons, pressing down on the pushbutton first activates the button native to the mouse and then, with more pressure, pops-through to activate the additional pushbutton. In informal evaluation, users had no difficulty distinguishing or controlling the activations of the native or additional button.

For our second protoype, we replaced the internal button mechanism with an integrated double-action surface-mount pushbutton[2], similar to a camera's focus/shutter-release button. The activation forces for this pushbutton are quite subtle, with no perceptible click for the first contact and a very soft feel, not really a click, for the second, pop through, contact

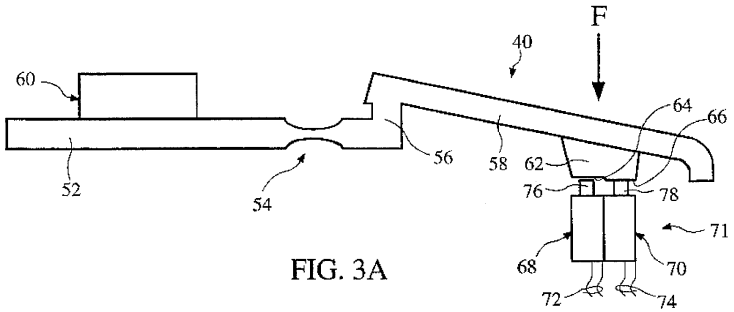


FIG. 3A

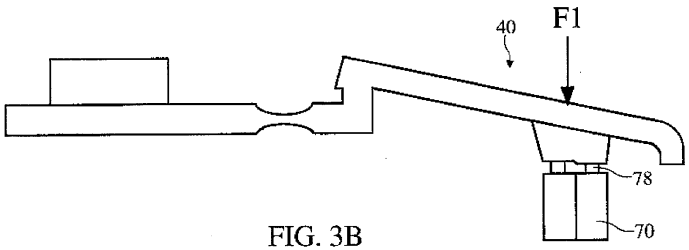


FIG. 3B

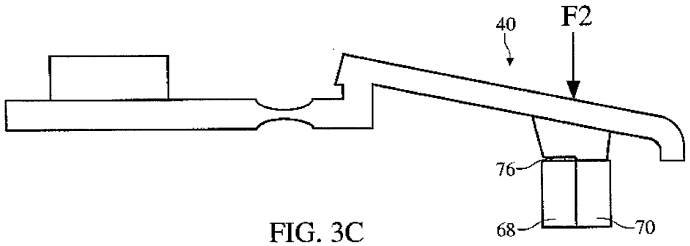


FIG. 3C

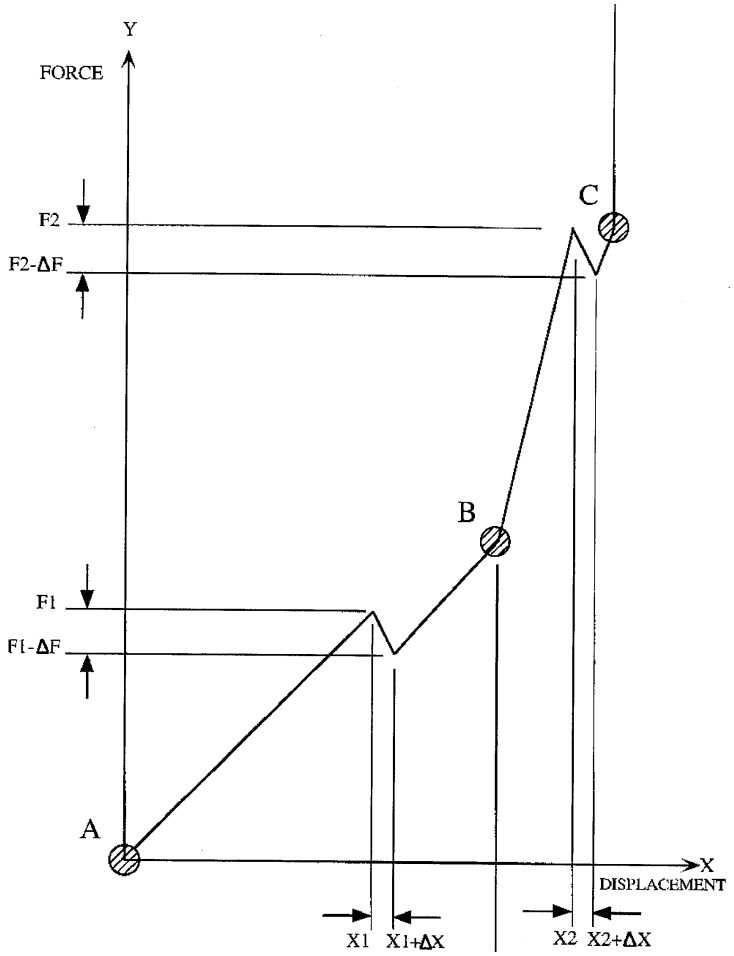


FIG. 4

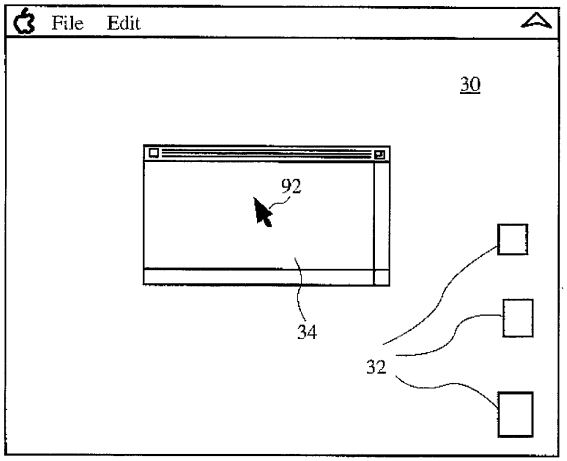


FIG. 6A

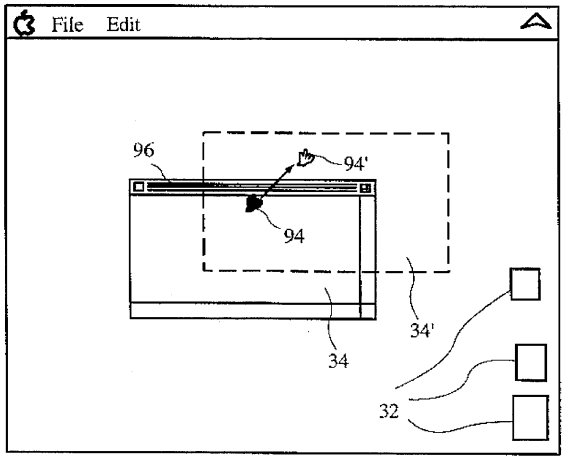


FIG. 6B

QUESTIONS:

What (low-level) tasks are the users trying to accomplish with an input device?

How can we think about the space of possible input devices?

What interaction techniques are encouraged/ discouraged by a particular device?

IMPORTANT TASKS

Text Entry

Pointing/Marking

- Target acquisition
- Steering / positioning
- Freehand drawing
- Drawing lines
- Tracing and digitizing
- ...

TEXT ENTRY: KEYSTROKE DEVICES

Array of Discrete Inputs

Many variants of form and key layout

Can be one-handed or two

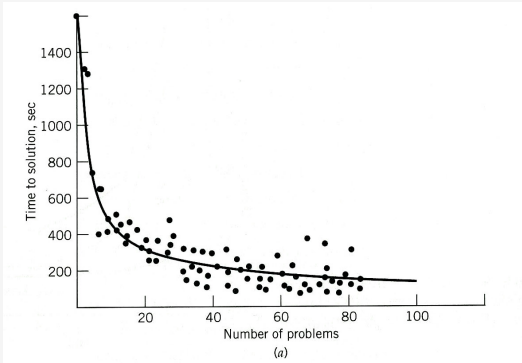
Wide range of sizes

Two-hand full keyboard is relatively standardized, Less standardization on others: Command keys, generic function keys, cursor movement, numeric keypad,...

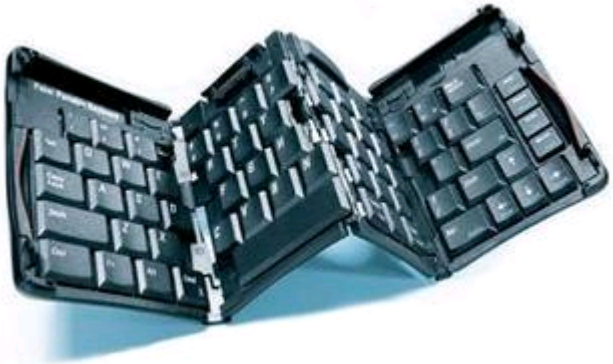
Take advantage of procedural memory

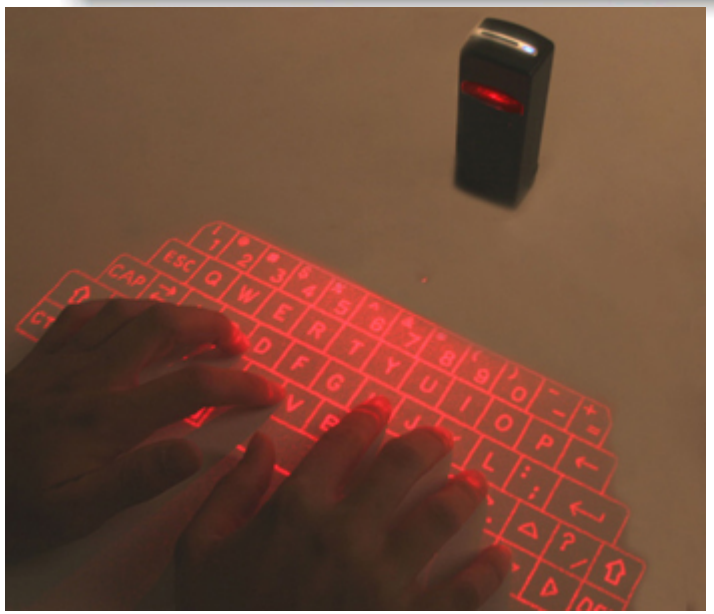
Power law of practice

$$T_n = T_1 n^{-a} + c$$



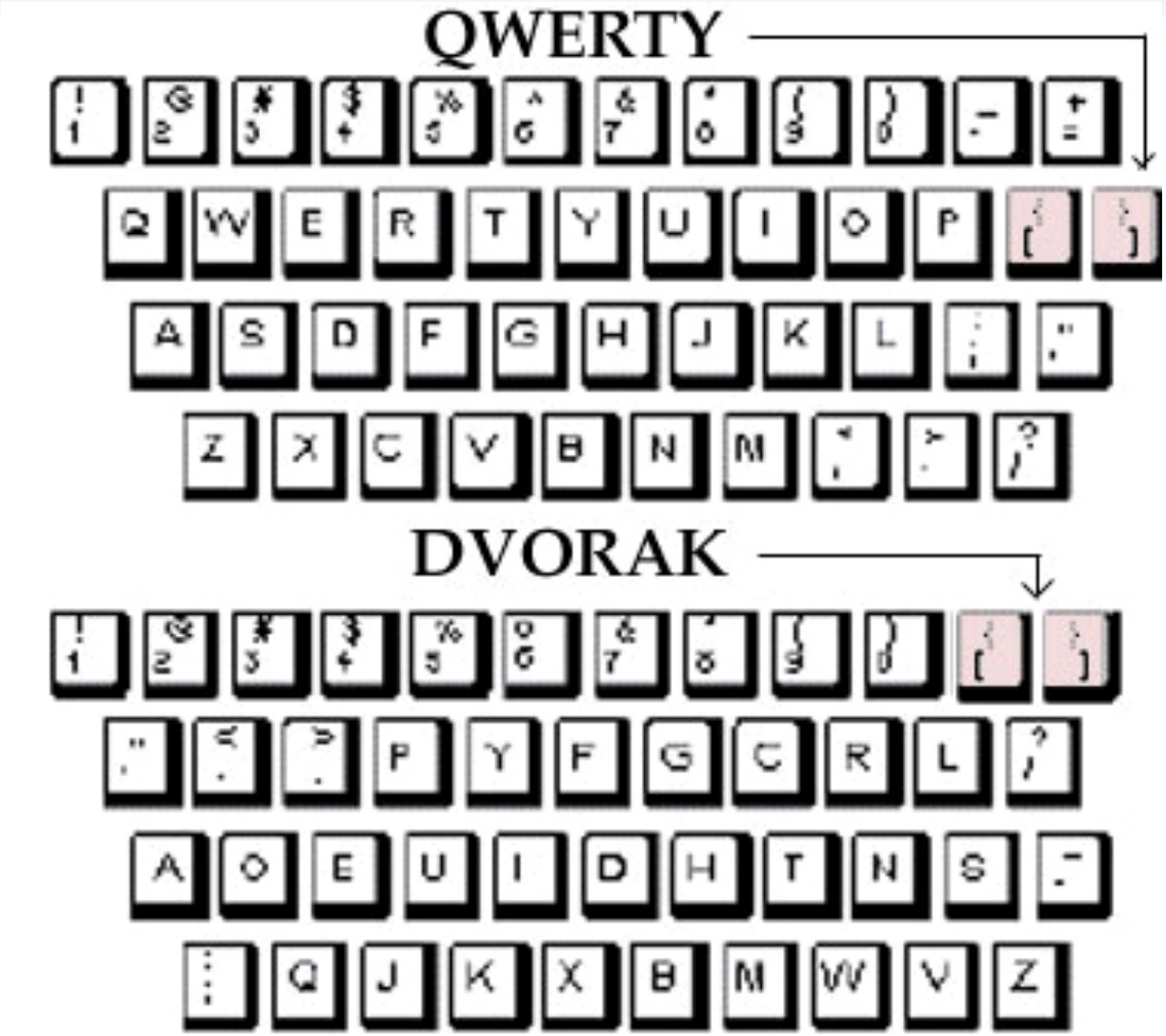
KEYBOARDS





Maxim

KEY LAYOUTS



DIFFICULTY: TEXT ENTRY

Still very hard on mobile devices

Keyboards (on-screen and thumb)

Full hand-writing recognition

Graffiti

EdgeWrite

ShapeWriter

MOBILE TEXT ENTRY: KEYPADS

Multi-tap mappings

Multiple presses per letter

Ambiguity resolution

One press per letter, dictionary lookup

MOBILE TEXT ENTRY: KEYPADS

Chording

Multiple keys pressed simultaneously

2^n combinations for n keys

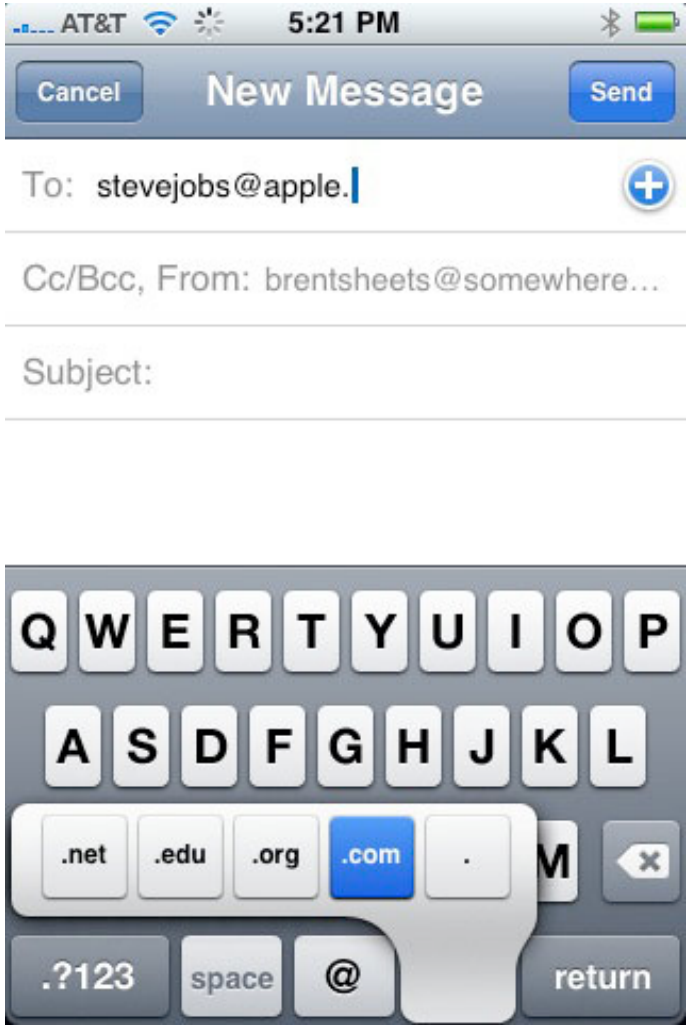
Twiddler2, HandyKey

Doug Engelbart

MOBILE TEXT ENTRY: SOFT KEYS

Soft Keyboards

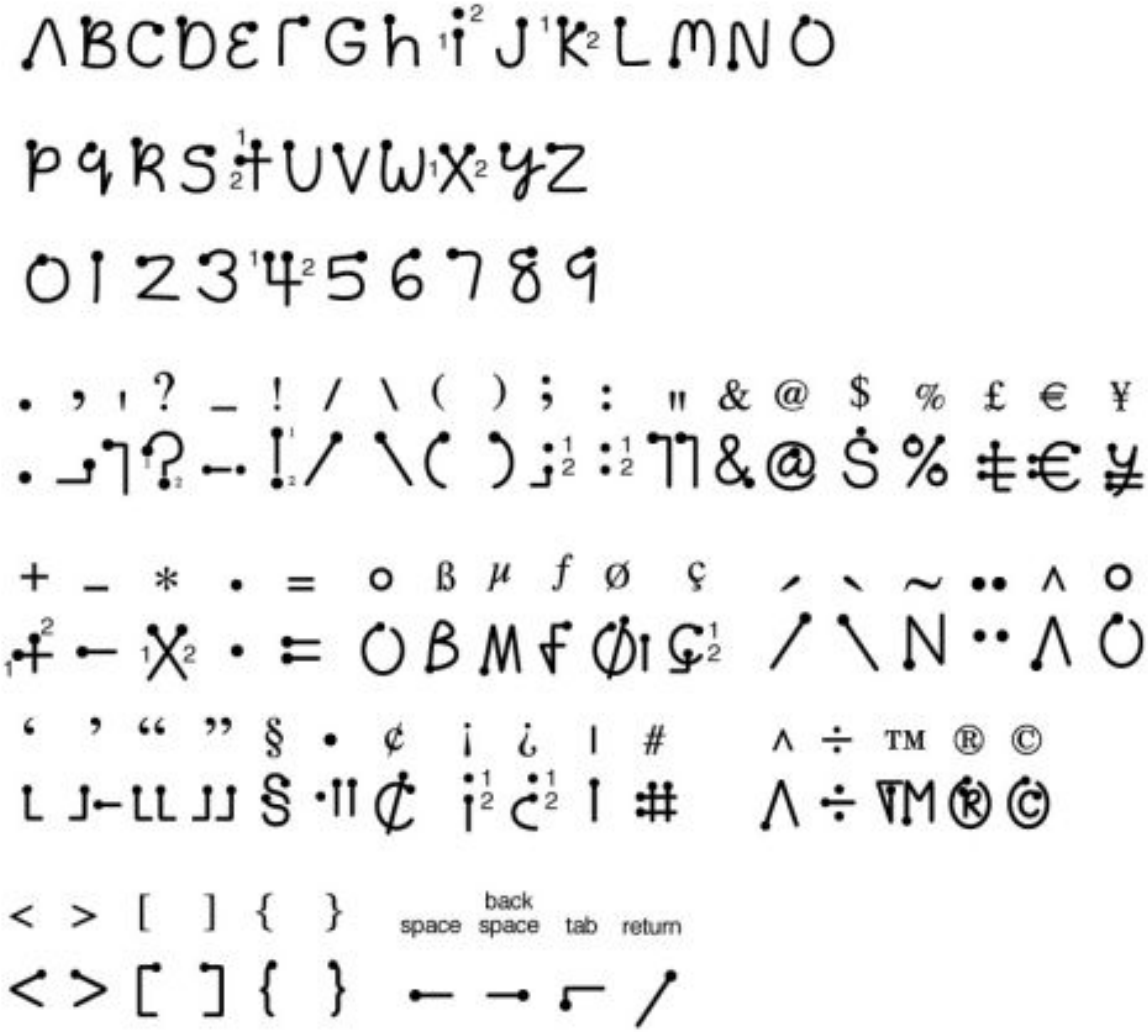
Benefits? Drawbacks?



Mactoids.com

MOBILE TEXT ENTRY: HANDG RECOG

GRAFFITI – UNISTROKE TEXT ENTRY



EDGEWRITE

Corner-based text input technique

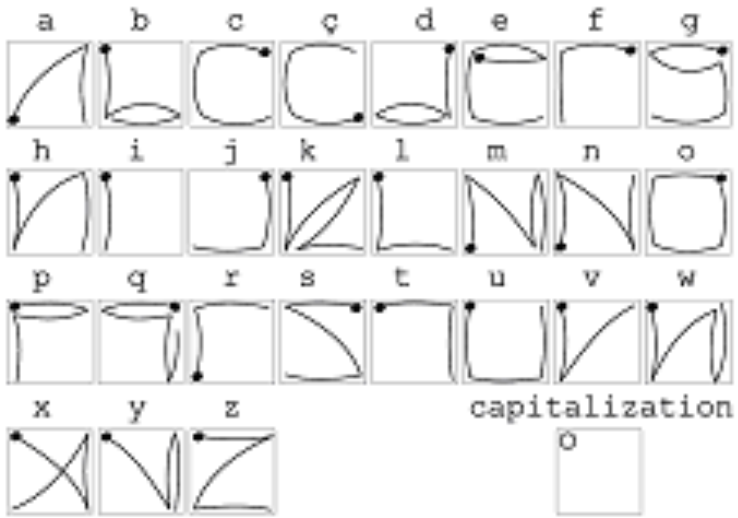
Makes use of physical edges and corners to improve input time

Particularly effective for users with motor impairments

Edges provide stability

Implementable in many different input modalities

stylus, joysticks, trackball

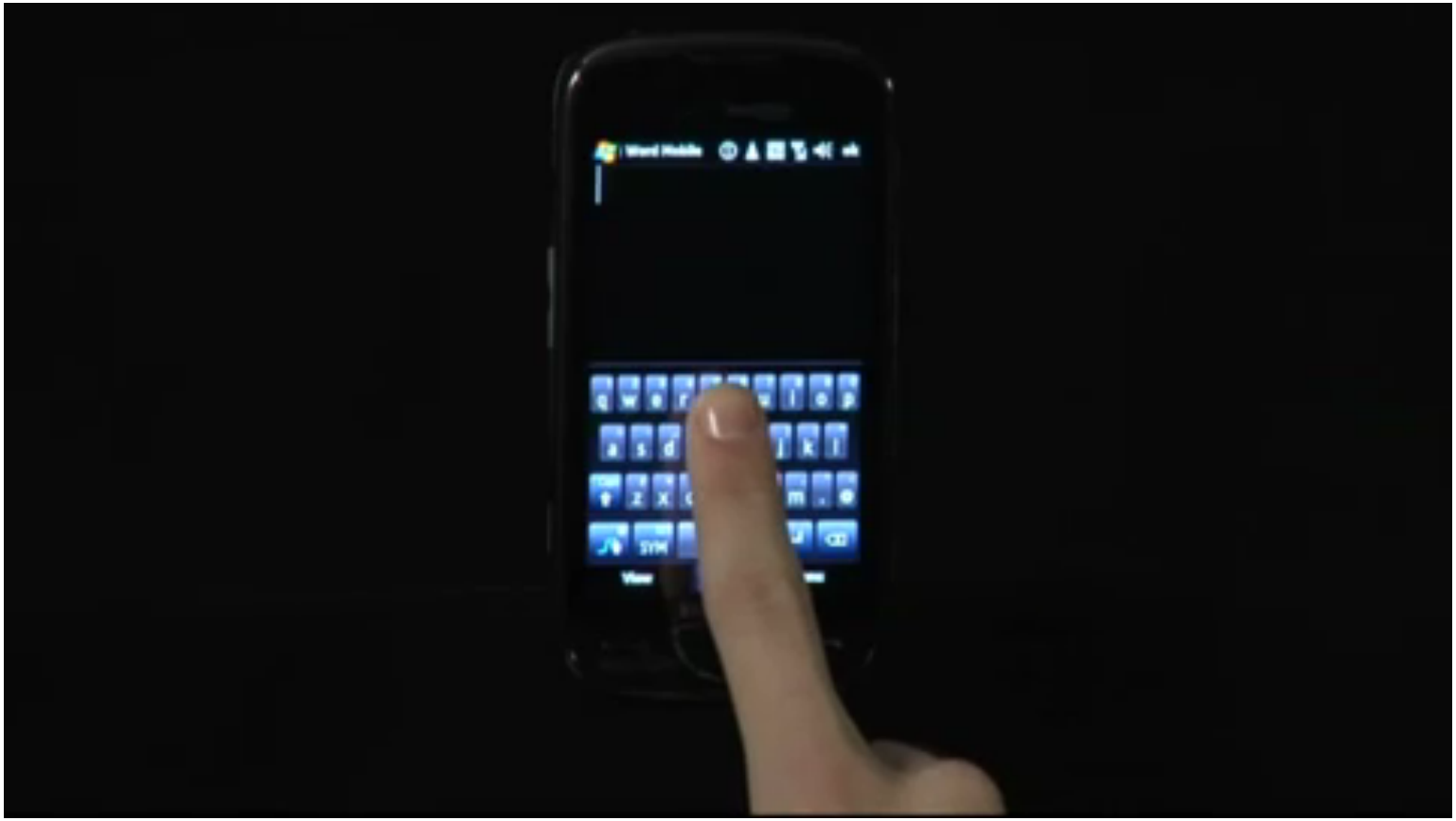


Jacob Wobbrock, UIST 2003

EDGEWRITE VIDEO

MOBILE TEXT ENTRY: TOUCH / STYLUS

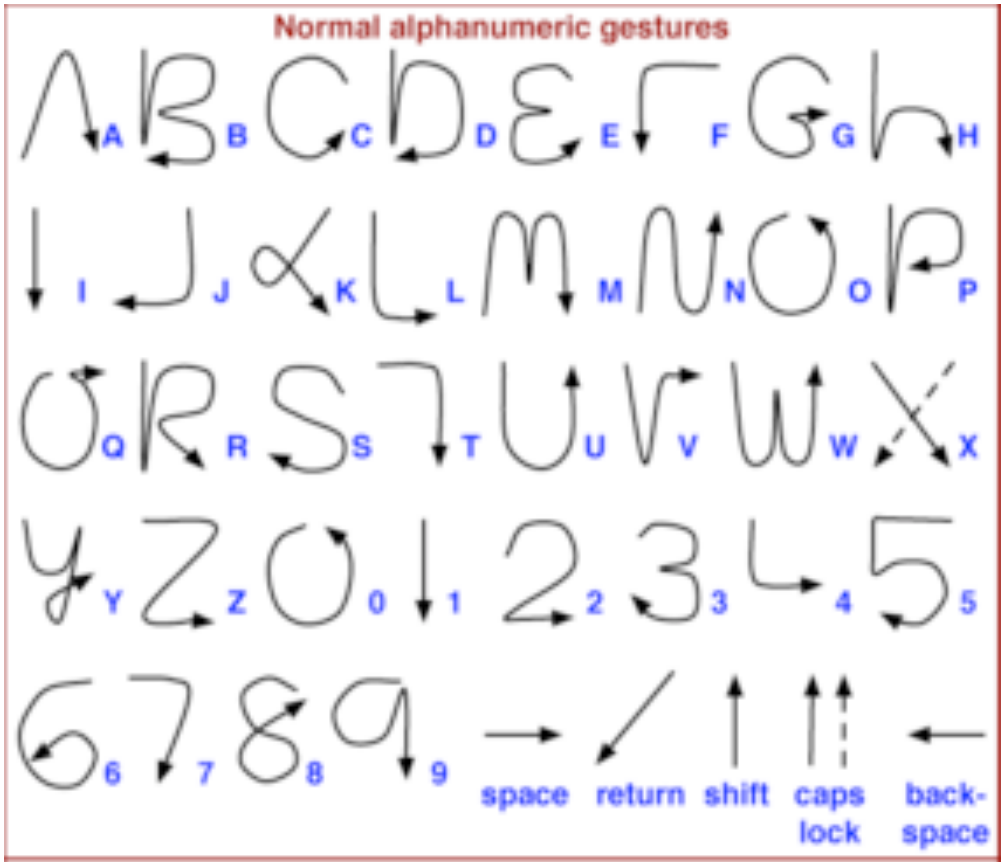
Stroke Entry Methods (e.g., Swype, ShapeWriter)



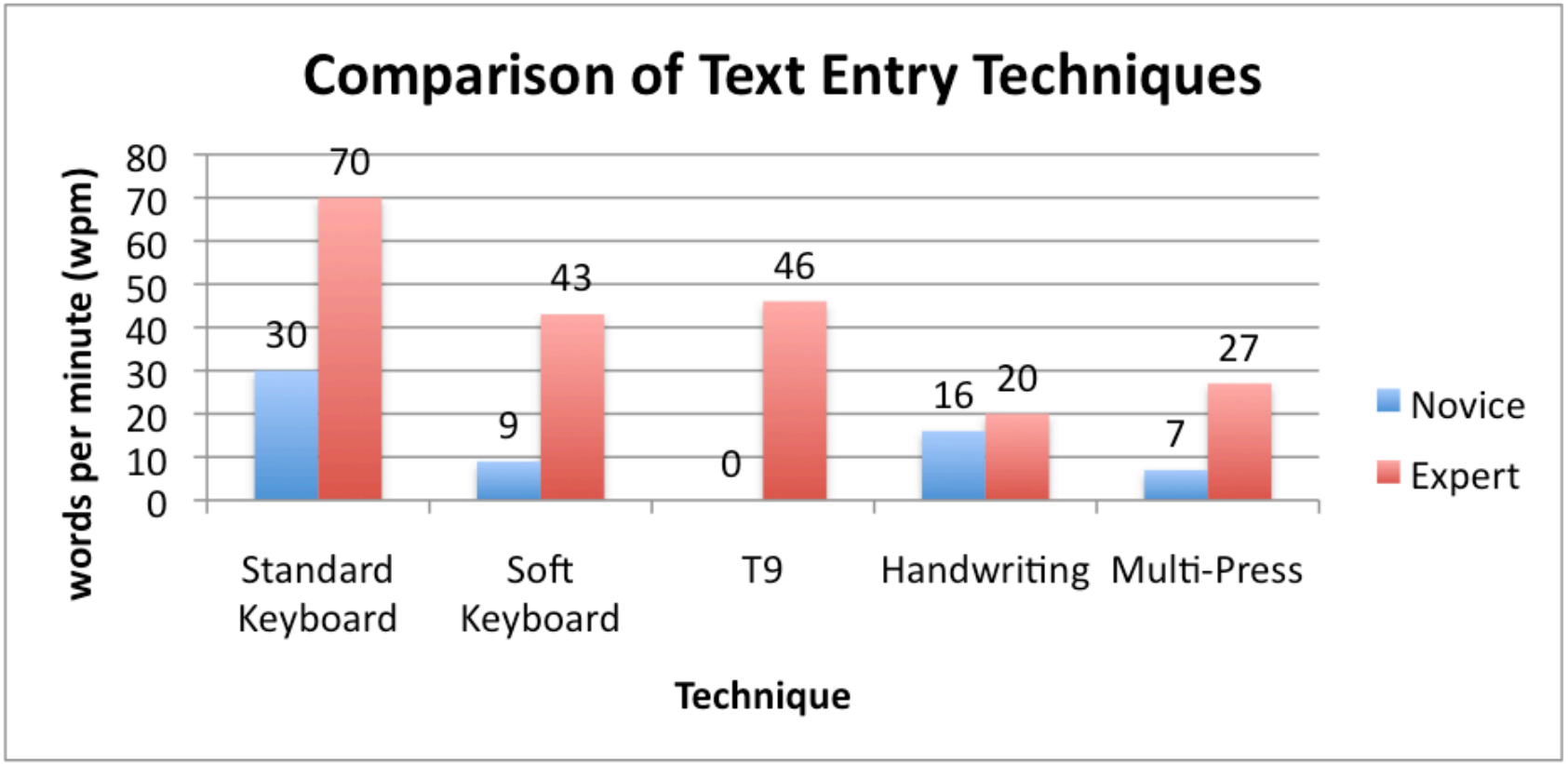
MOBILE TEXT ENTRY: TOUCH / STYLUS

Custom symbol sets improve recognition accuracy;

appropriate for indirect (eyes-free) input



WHICH IS FASTEST?



WHAT ABOUT SPEECH RECOGNITION?

Dictation is faster than typing (~100 wpm)

WHAT ABOUT SPEECH RECOGNITION?

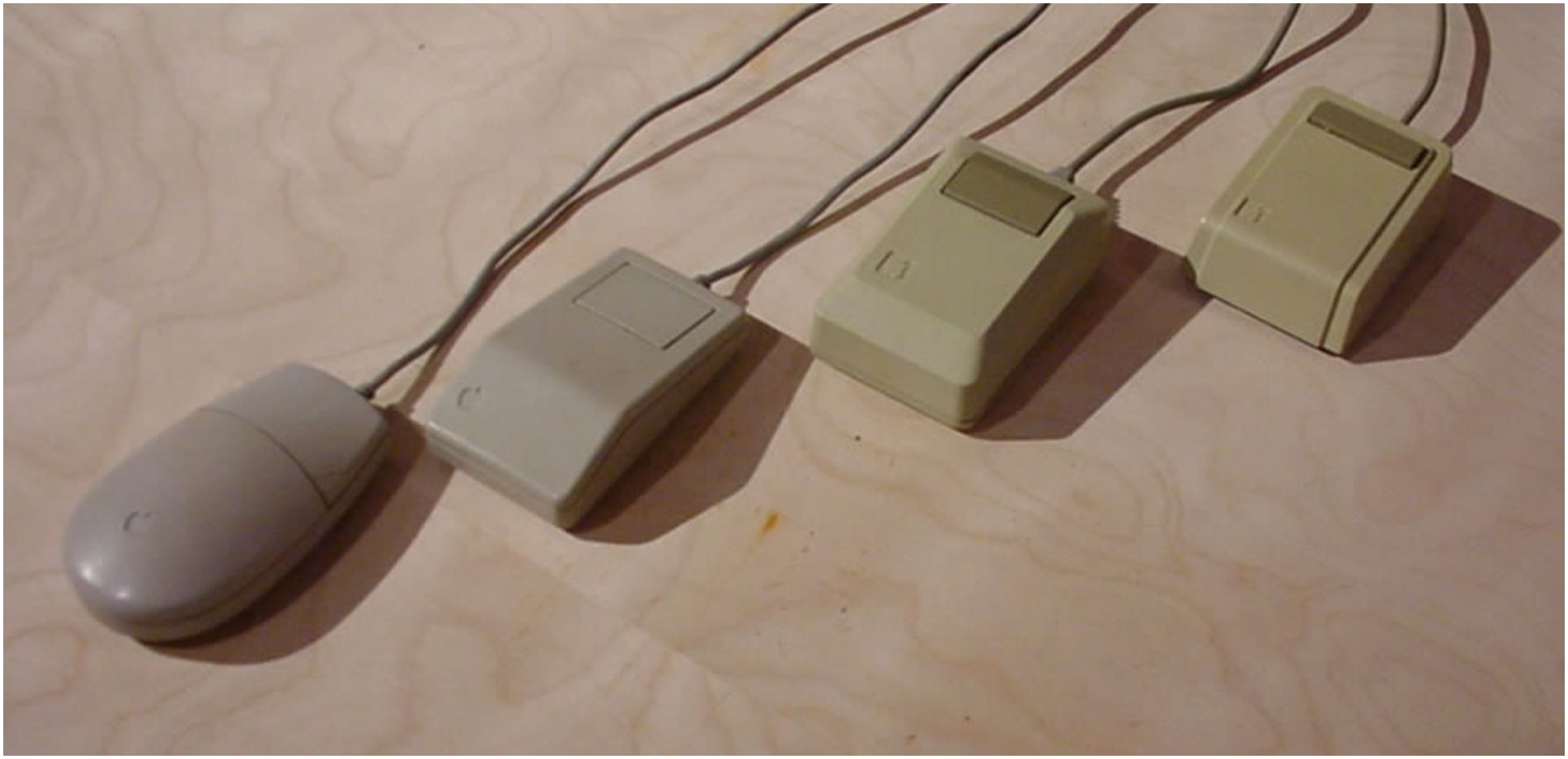
Dictation is faster than typing (~100 wpm), BUT:

Speech is different from written language:
Speaking in well-formed, complete, print-ready sentences is cognitively challenging

High cost of correcting errors through speech channel alone

Social awkwardness?

POINTING DEVICES



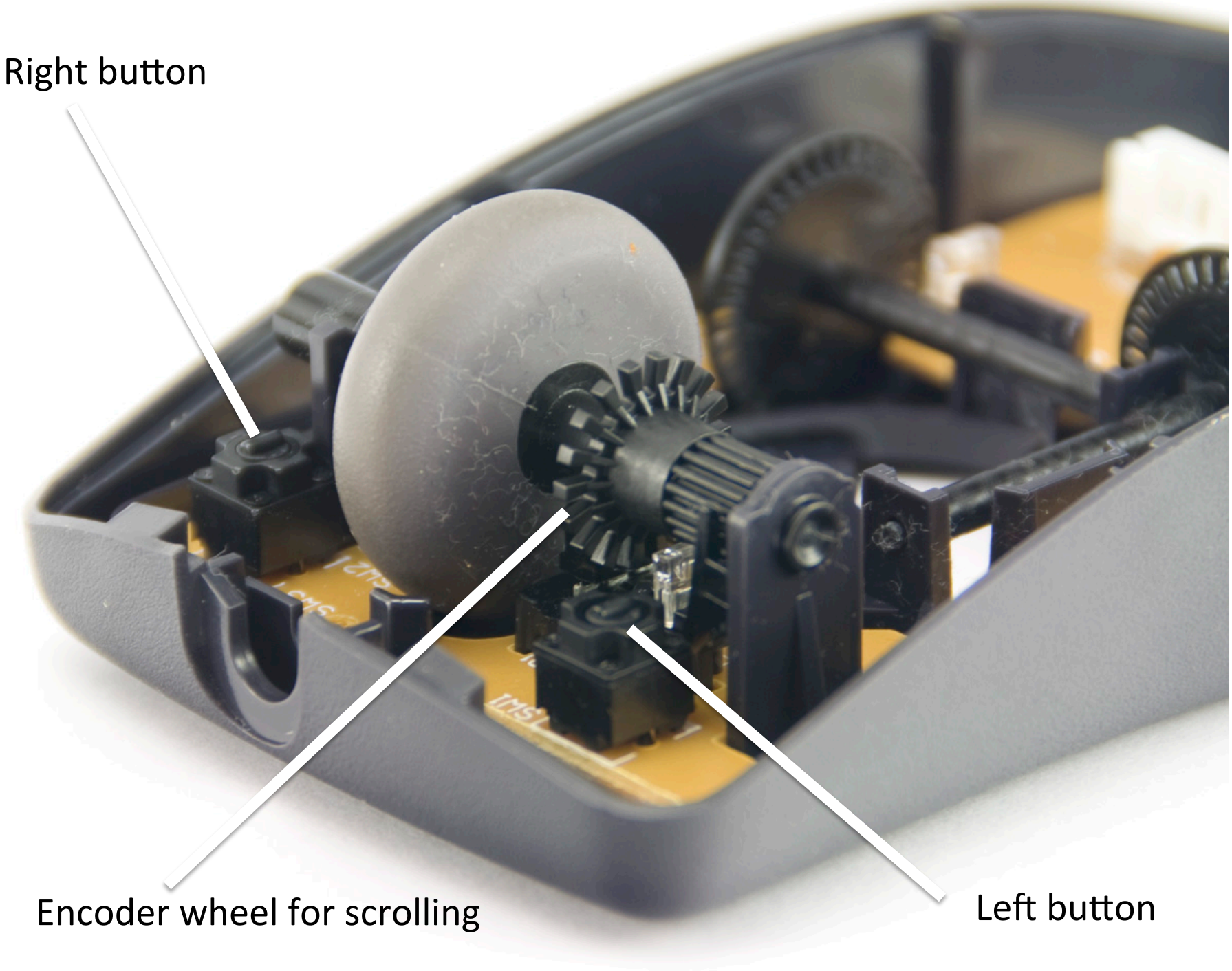
(cc) Flickr photo by Mike fj40

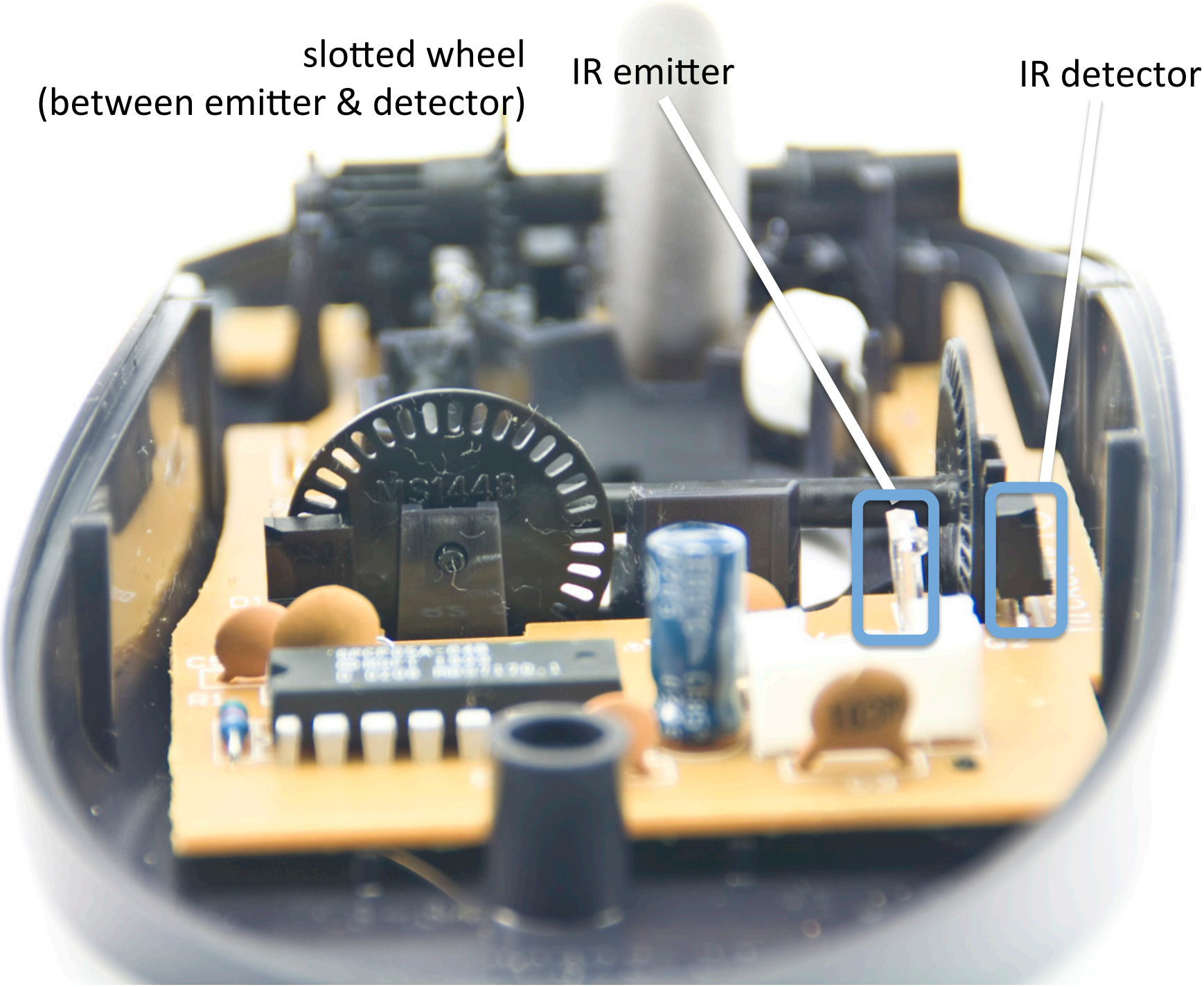
Mouse. Engelbart and English ~1964

Source: Card, Stu. Lecture on Human Information Interaction. Stanford, 2007.

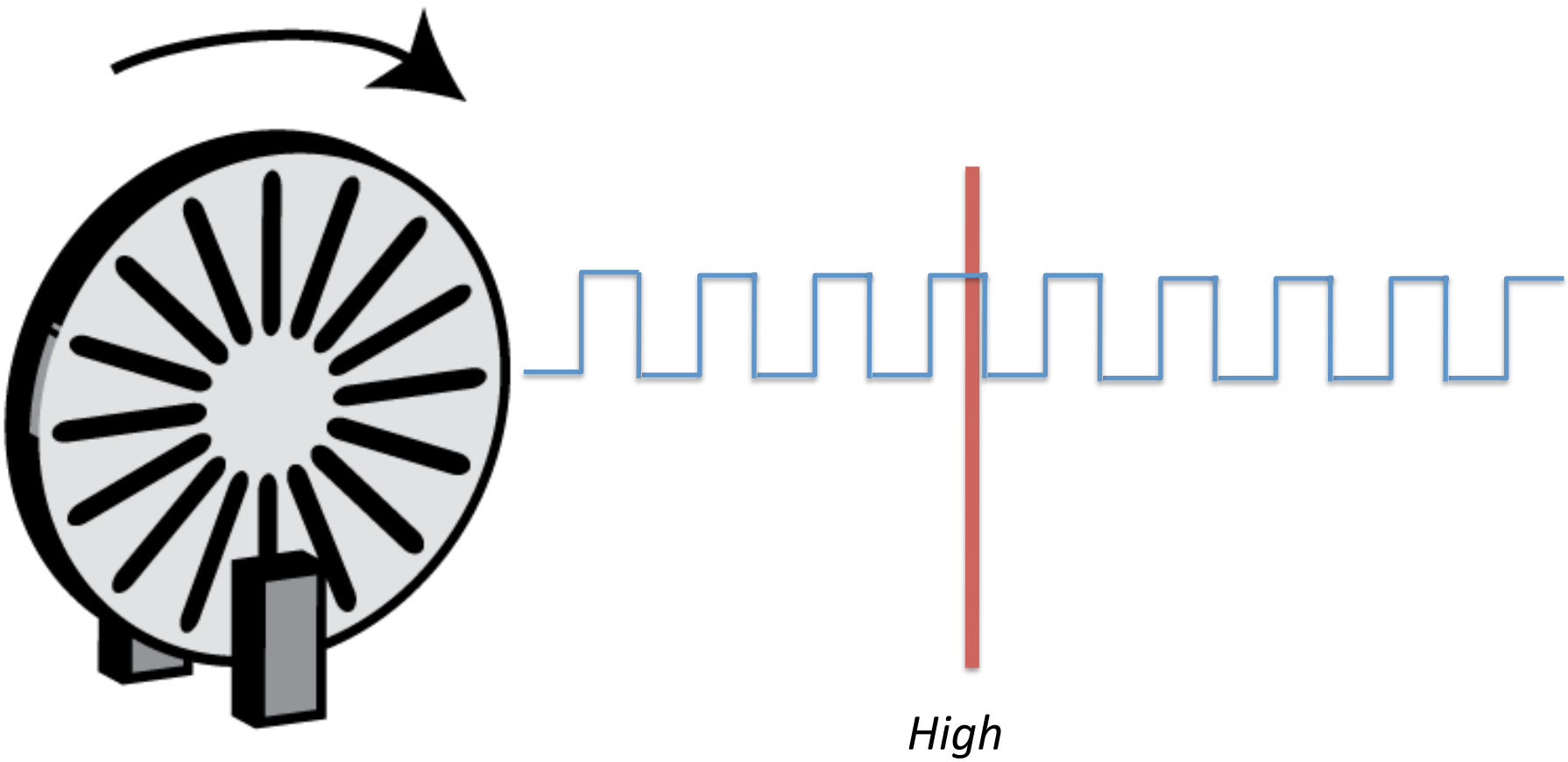
DELL
by Microsoft

D3
D2
Q2
C2
C3
C6
R1
R2
U1
SPCP05A-048
©MSFT 1999
0 0200 MS71170.1

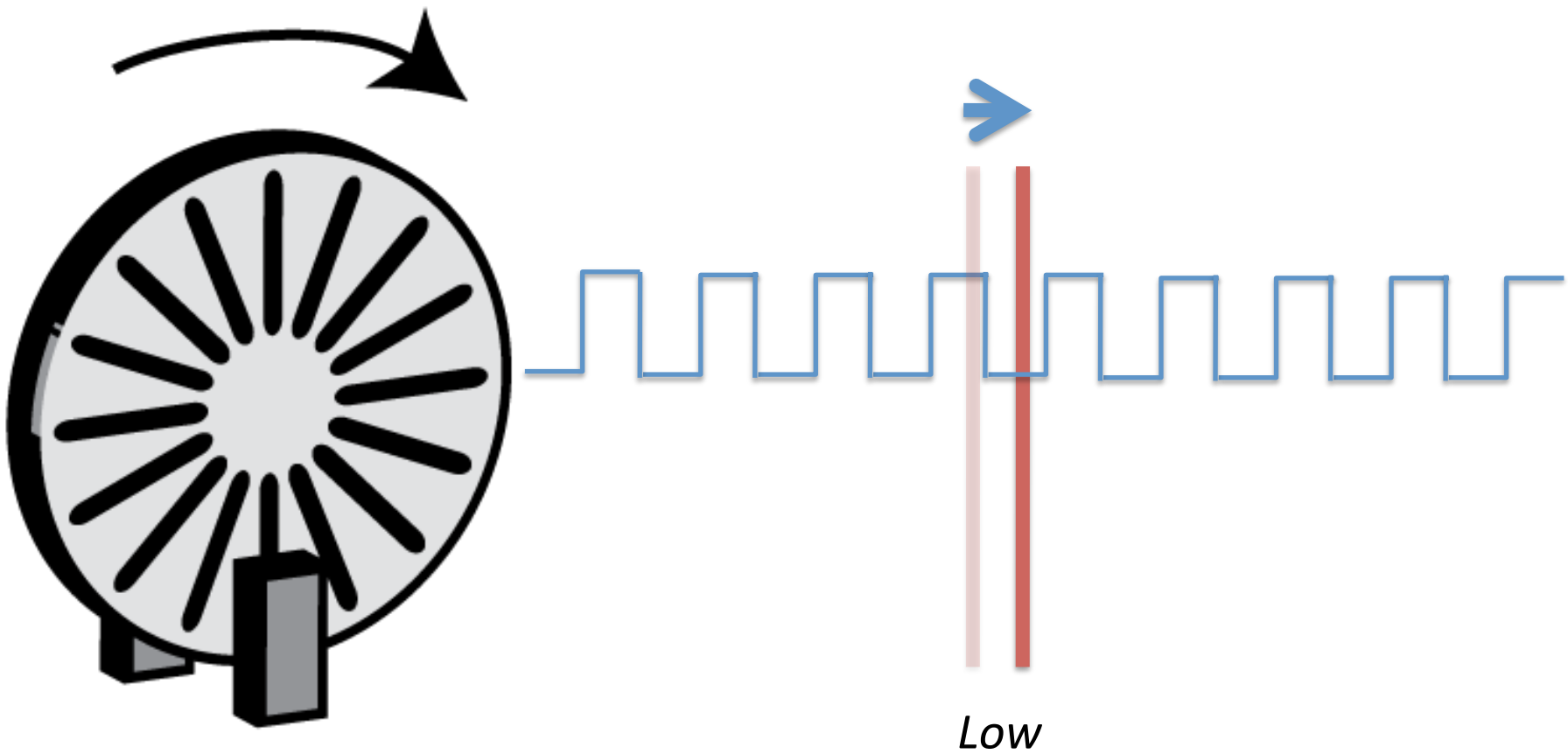

Right button
Encoder wheel for scrolling
Left button


slotted wheel
(between emitter & detector)
IR emitter
IR detector

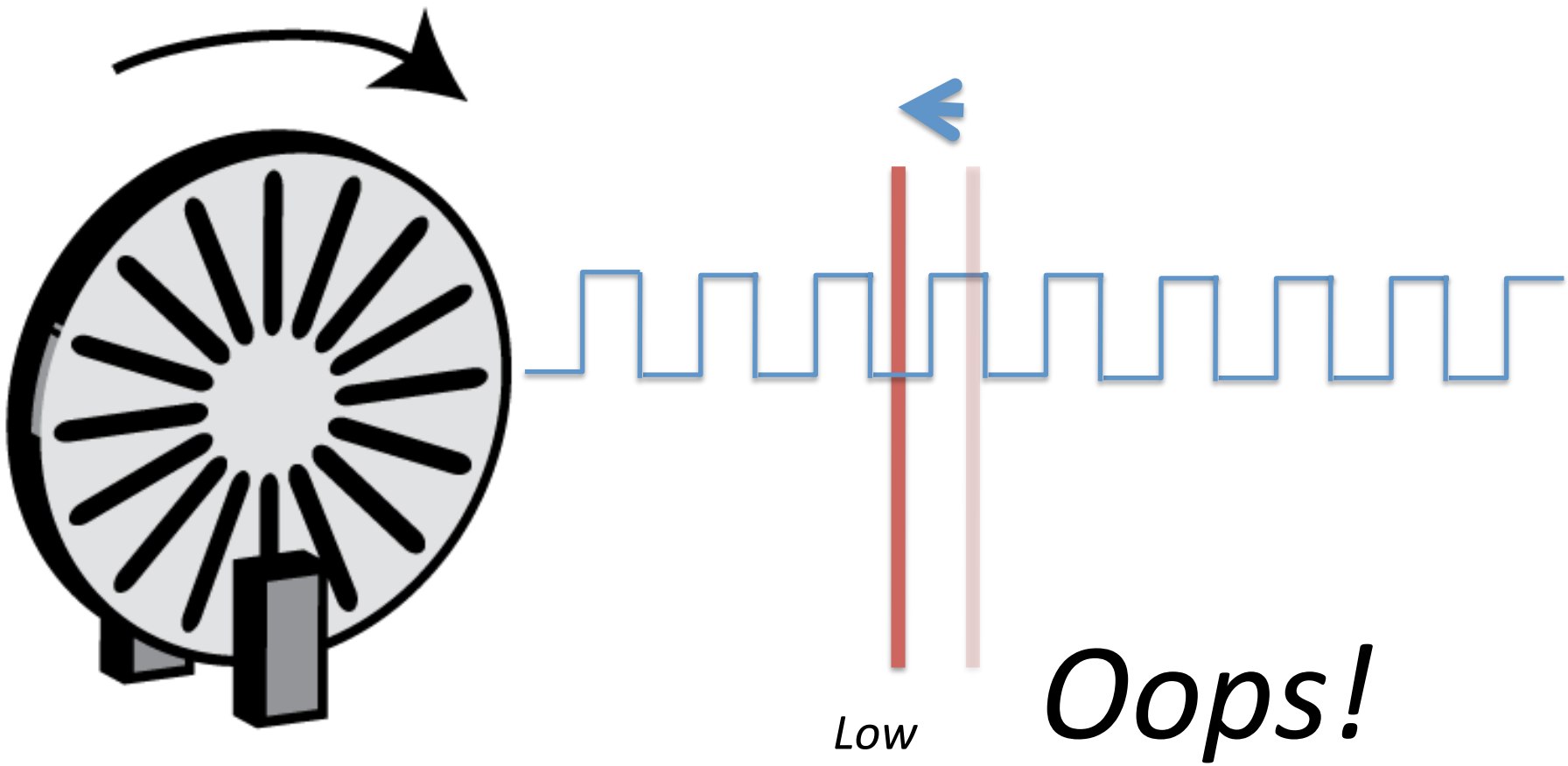
SENSING: ROTARY ENCODER



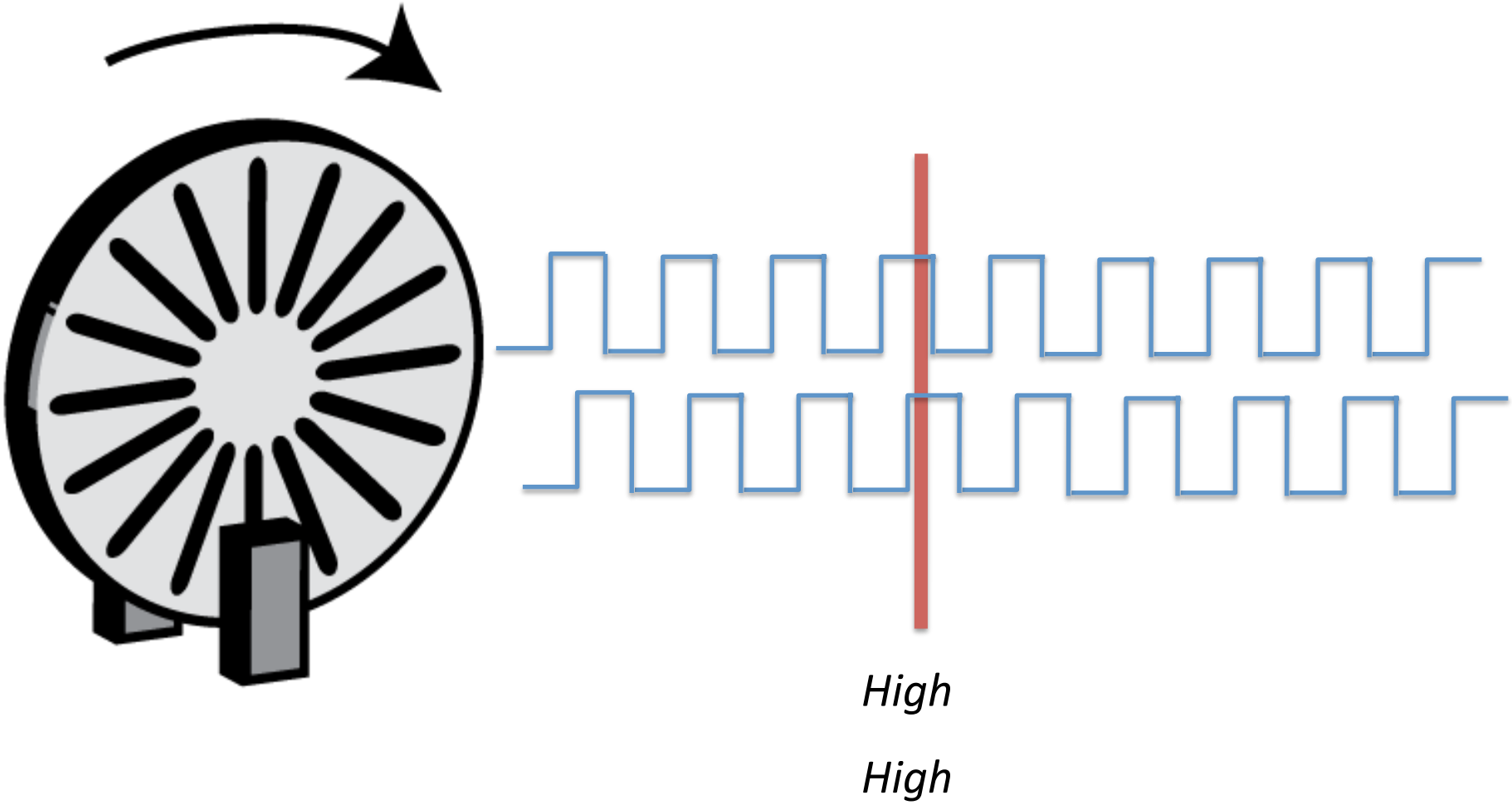
SENSING: FWD ROTATION



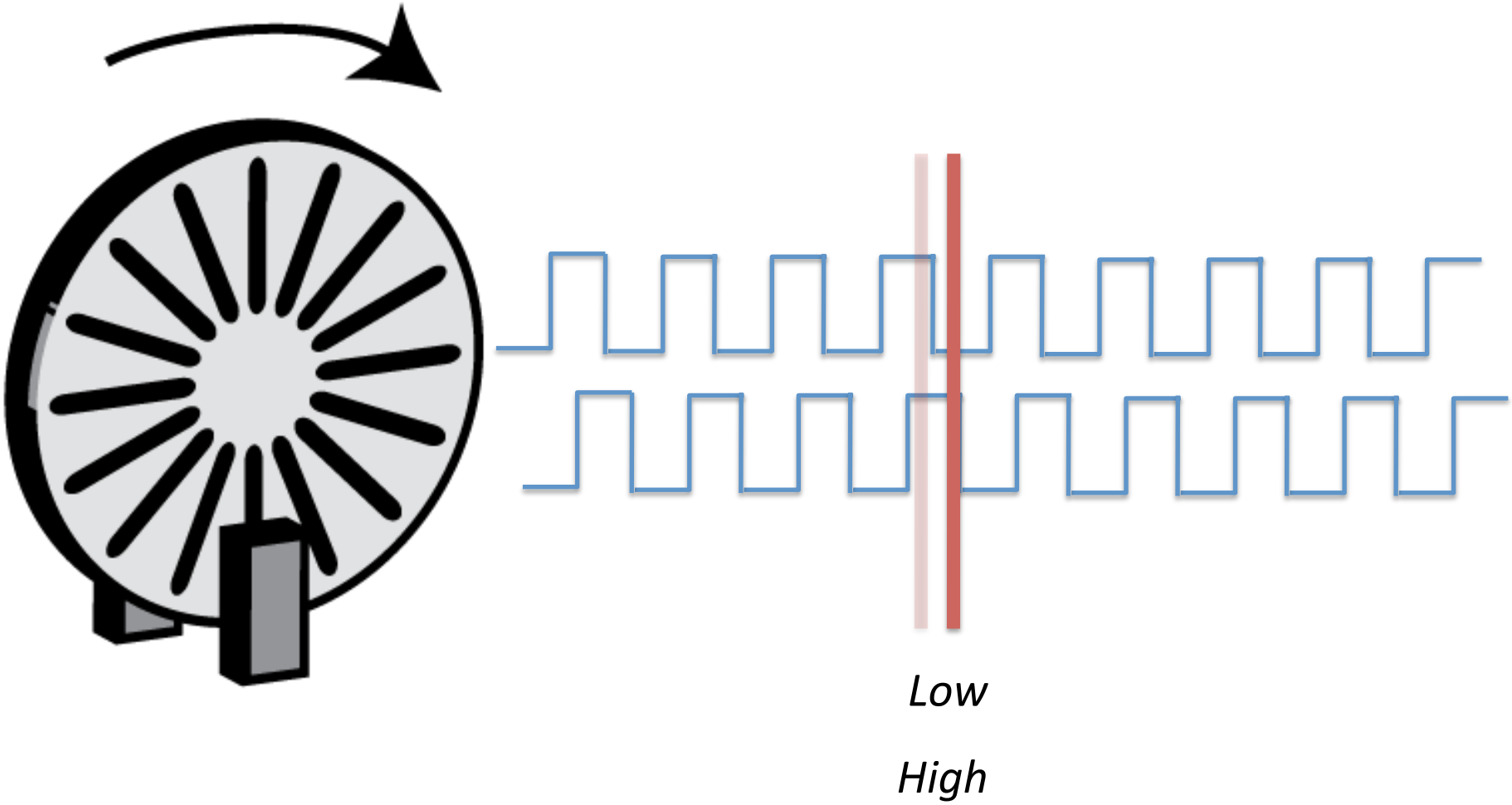
SENSING: BACKWD ROTATION



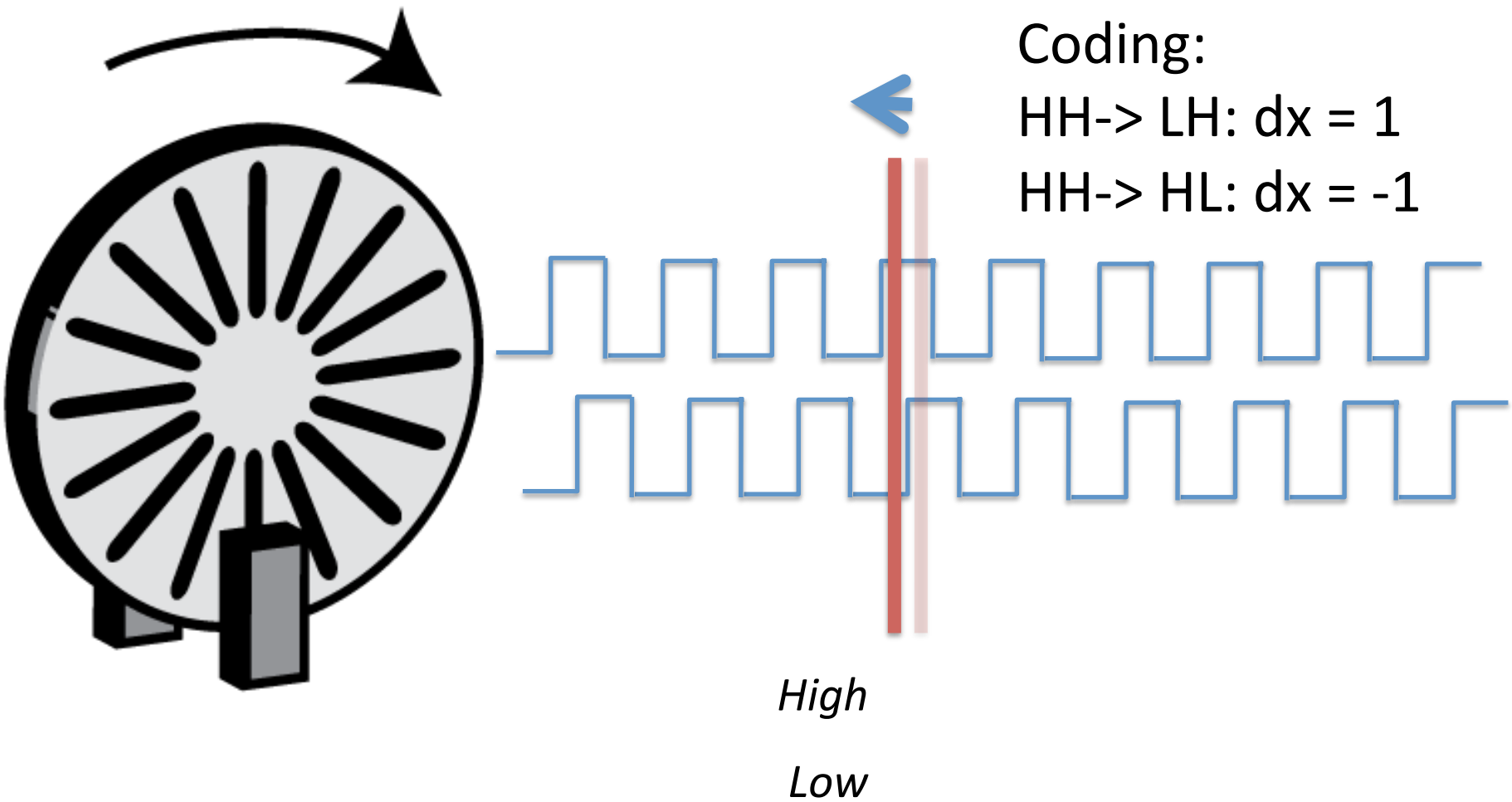
SOLUTION: USE TWO OUT-OF-PHASE DETECTORS



SENSING: ROTARY ENCODER



SENSING: ROTARY ENCODER

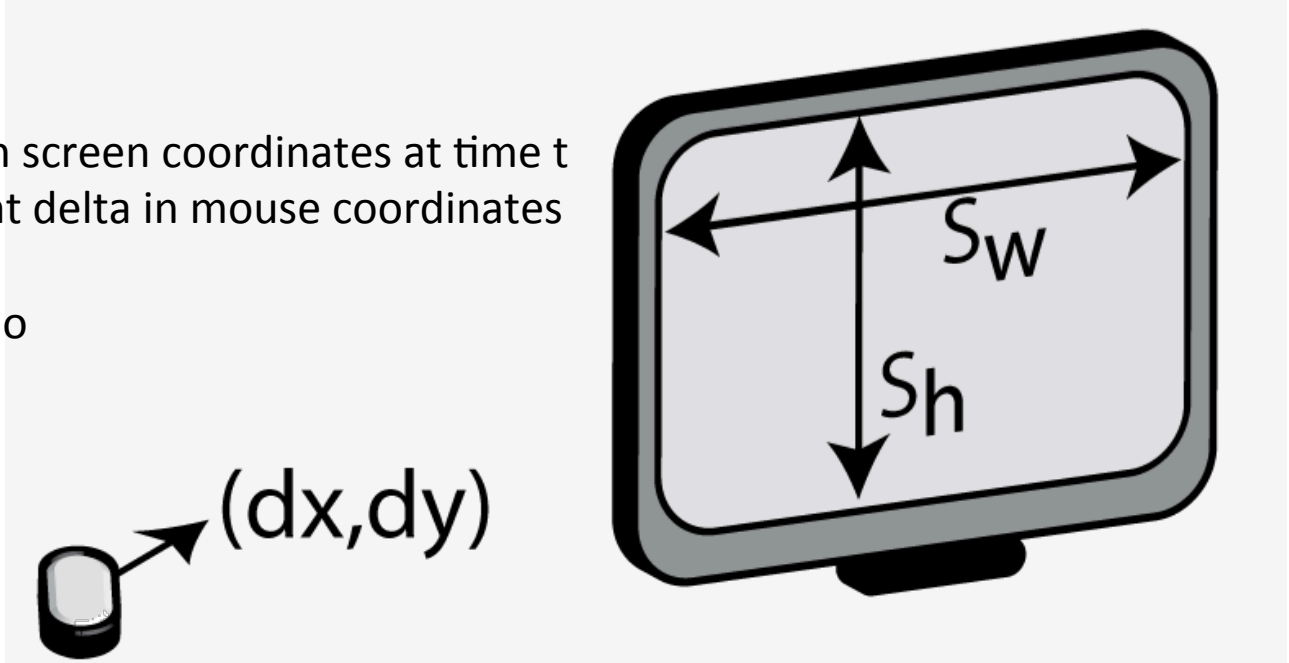


TRANSFORMATION

$cx_t = \max(0, \min(sw, cx_{t-1}+dx*cd))$

$cy_t = \ldots$

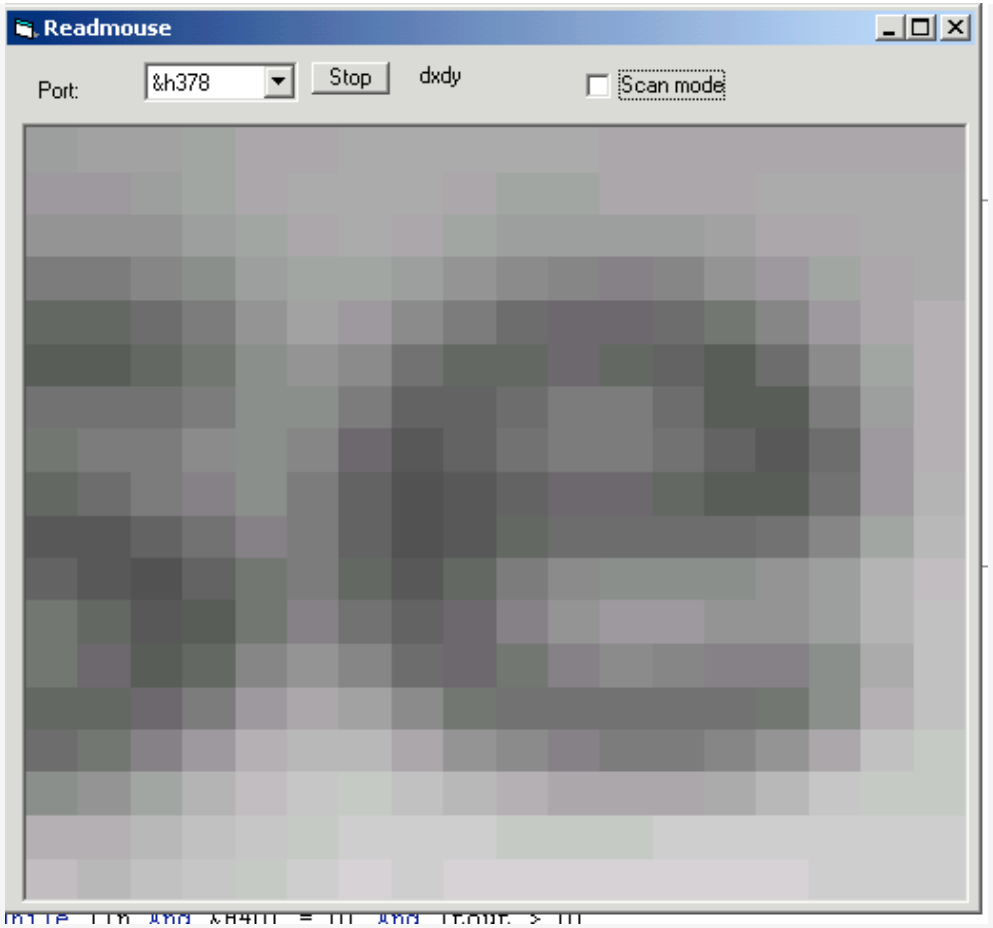
cx_t: cursor x position in screen coordinates at time t
dx: mouse x movement delta in mouse coordinates
sw: screen width
cd: control-display ratio



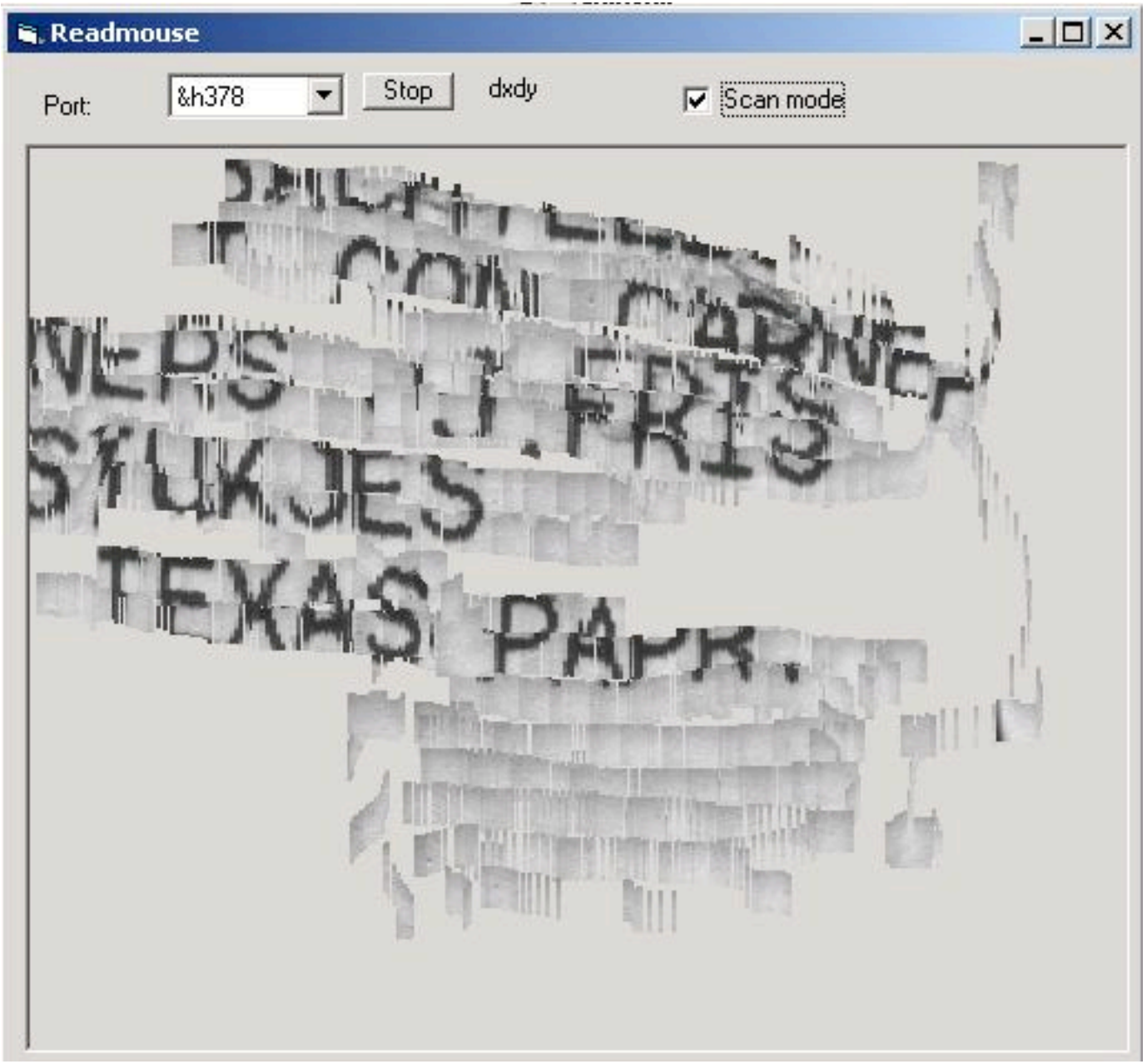
DEVICE ABSTRACTION

Click, DoubleClick, MouseUp, MouseDown, MouseMove ...

WHAT ABOUT OPTICAL MICE?



Source: http://spritesmods.com/?art=mouseeye



Source: http://spritesmods.com/?art=mouseeye

WHAT IS SENSED?

Table I. Physical Properties Used by Input Devices

	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**

Card, S. K., Mackinlay, J. D., and Robertson, G. G. 1991. A morphological analysis of the design space of input devices. *ACM Trans. Inf. Syst.* 9, 2 (Apr. 1991), 99-122.

OTHER DEVICE PROPERTIES:

Indirect vs. Direct

Direct: Input and output space are unified

C:D Ratio

For one unit of movement in physical space, how far does the cursor travel in display space?

Q: What is the C:D ratio for direct touch screen input?

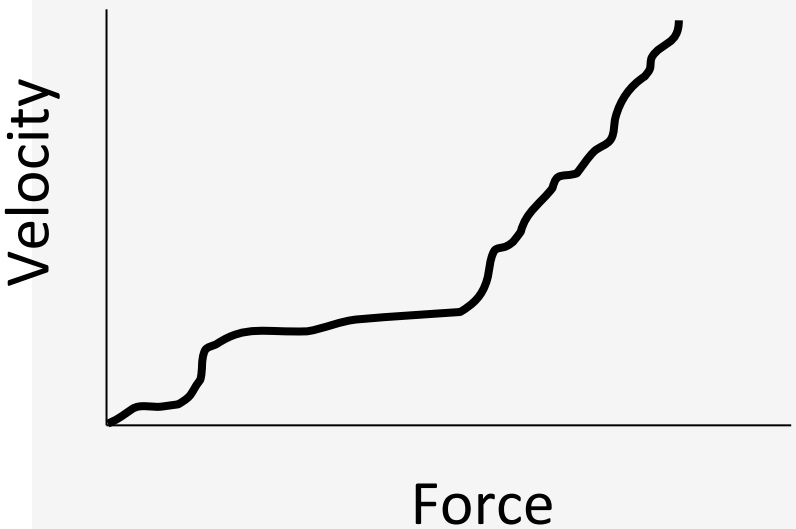
Device Acquisition Time

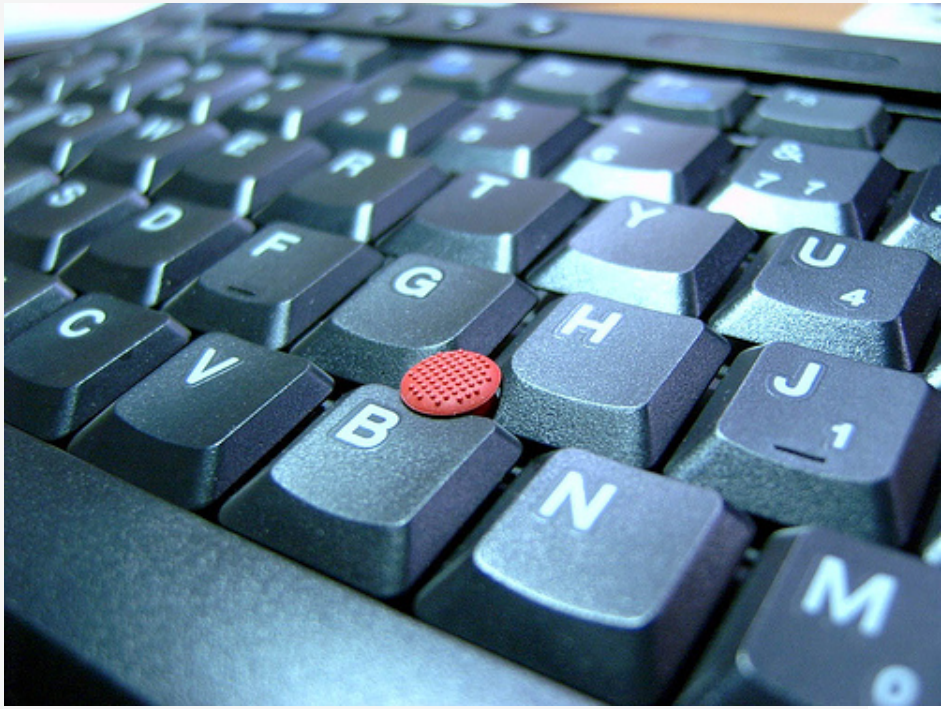
TRACKBALL, TRACKPAD

TRACKPOINT

Indirect, force sensing, velocity control

Nonlinear transfer function





(cc) Image by flickr user tsaiid

MOBILE POINTING

D-Pad
(see: arrow keys)

Trackball

Direct touch
(see: Trackpad)

Stylus

SCRATCH

Chris Harrison, et al

Everything is best for something and worst for something else.

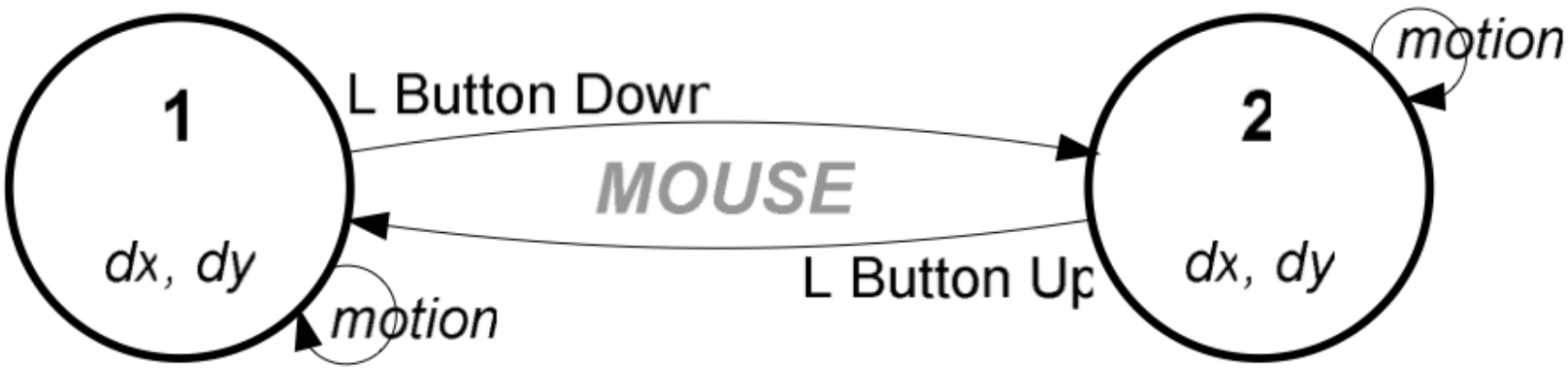
- Bill Buxton

3-STATE MODEL OF INPUT (BUXTON)

State	Description
0	*Out Of Range:* The device is not in its physical tracking range.
1	*Tracking:* Device motion moves only the cursor.
2	*Dragging:* Device motion moves objects on the screen.

(Table from Hinckley Reading)

MOUSE



(Figure from Hinckley Reading)

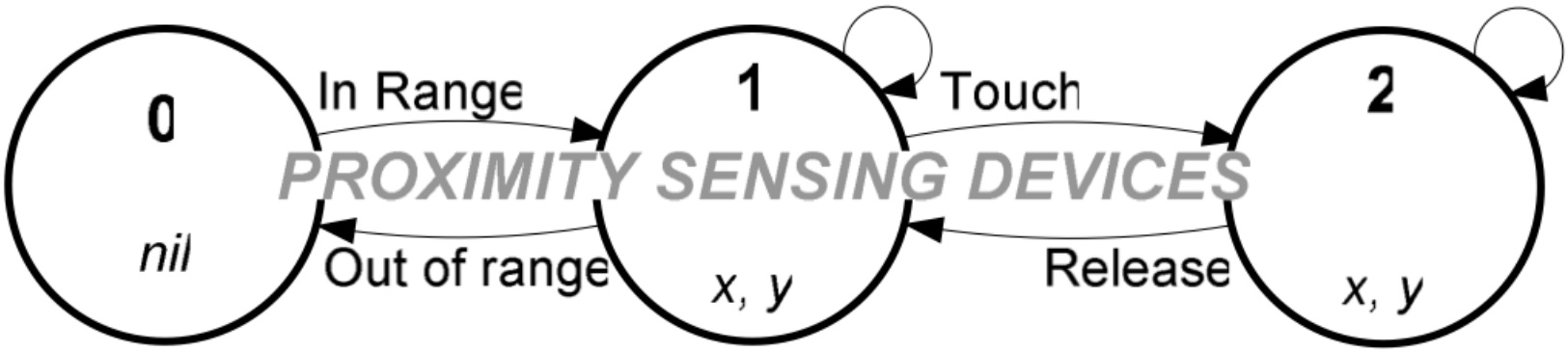
State	Description
0	*Out Of Range:* The device is not in its physical tracking range.
1	*Tracking:* Device motion moves only the cursor.
2	*Dragging:* Device motion moves objects on the screen.

TOUCH SCREEN

(Figure from Hinckley Reading)

State	Description
0	*Out Of Range:* The device is not in its physical tracking range.
1	*Tracking:* Device motion moves only the cursor.
2	*Dragging:* Device motion moves objects on the screen.

STYLUS ON TABLET



PROXIMITY SENSING DEVICES

(Figure from Hinckley Reading)

State	Description
0	*Out Of Range:* The device is not in its physical tracking range.
1	*Tracking:* Device motion moves only the cursor.
2	*Dragging:* Device motion moves objects on the screen.

(MULTI-) TOUCH

STRENGTHS

Direct input allows maximal screen space for mobile devices (ocular centrism).

More degrees of freedom.

“Virtual input devices” are adaptable.

No extra pieces to lose or break (styli!)

CHALLENGES (from Buxton)

No tactile feedback.

Requires free use of (both) hands and eyes.

“Fat Finger” problems – precision & occlusion

TERMINOLOGY (from Buxton)

Touch-tablets *vs* Touch screens

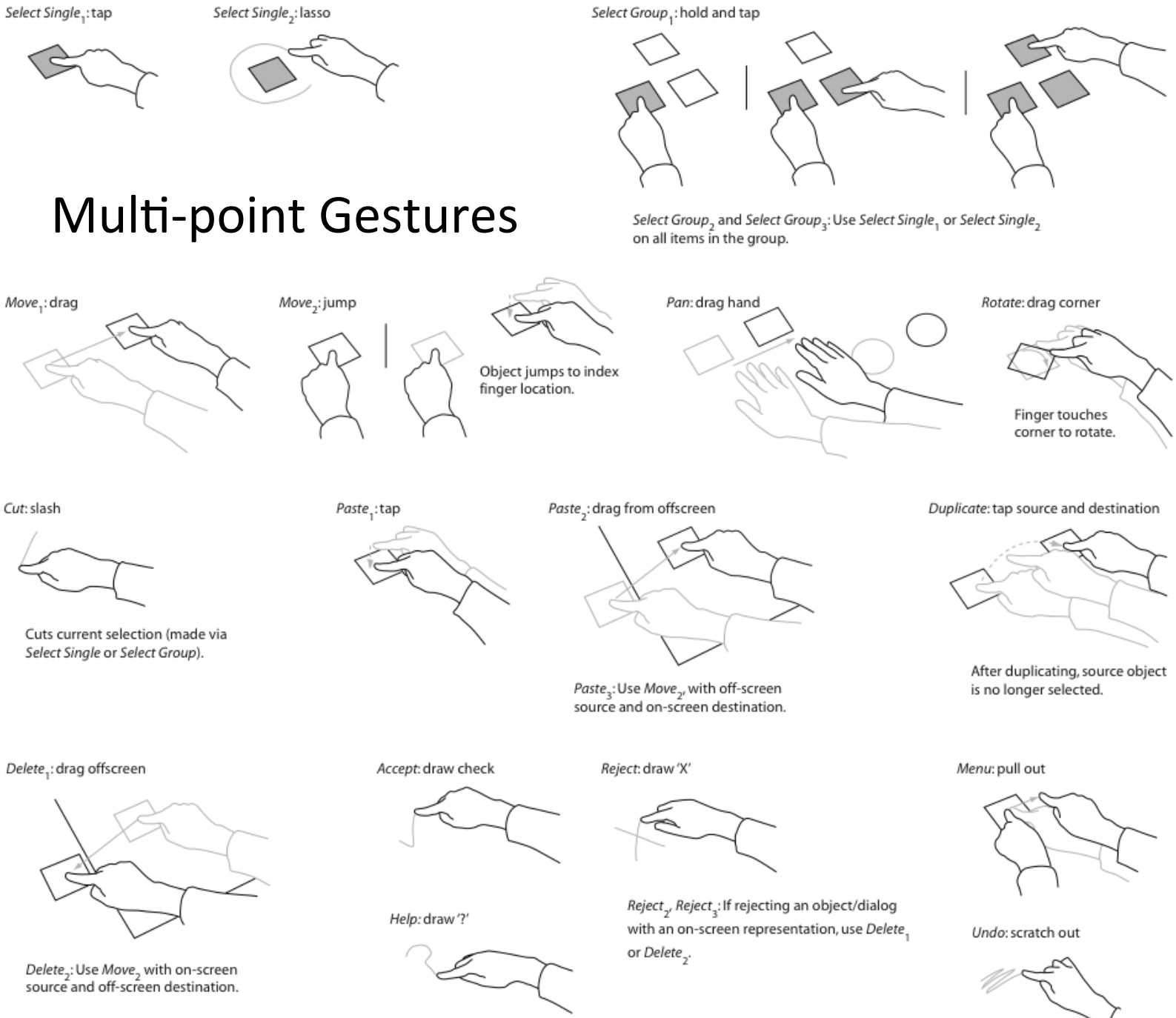
Single-finger *vs* multi-finger

Multi-person *vs* multi-touch

Points *vs* Gesture

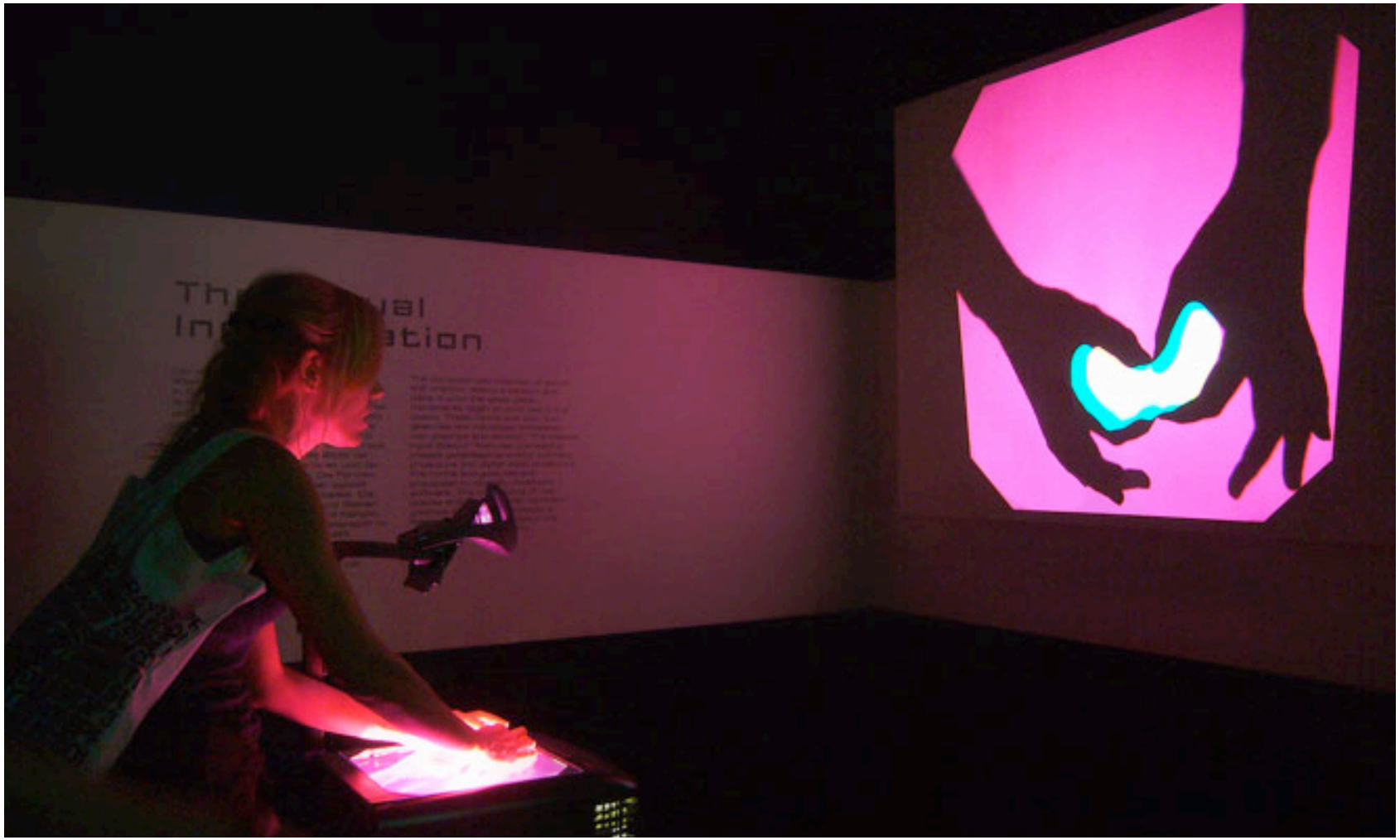
Hands and fingers *vs* Objects

Multi-point Gestures



Wobbrock, J., **Morris, M.R.**, and Wilson, A. User-Defined Gestures for Surface Computing. *Proceedings of CHI* 2009, 1083-1092.

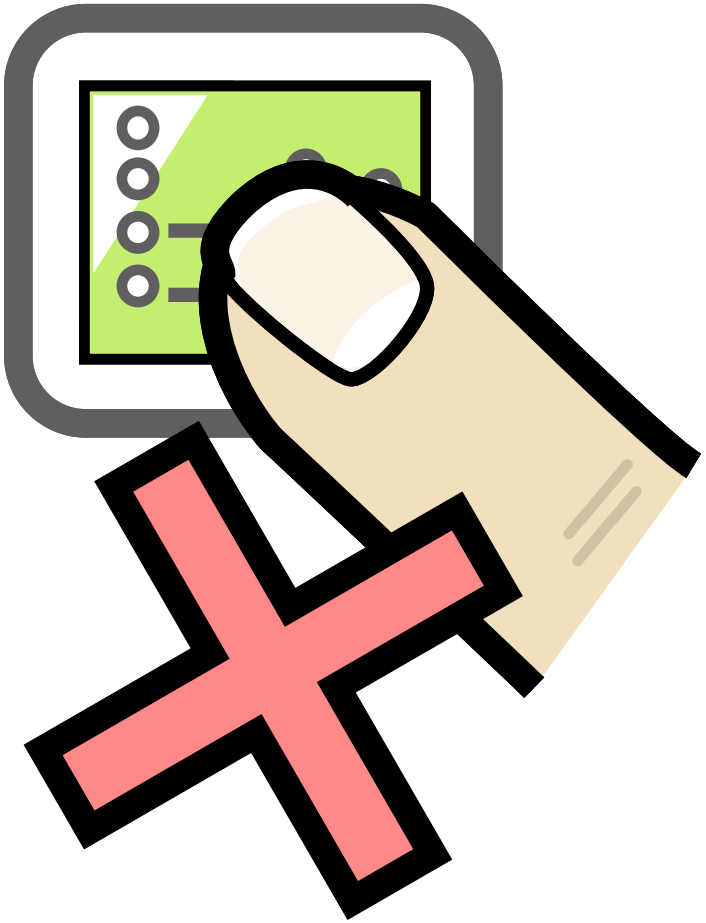
POSTURE-BASED INTERACTION



Golan Levin, Zach Lieberman – The Manual Input Sessions Workstation

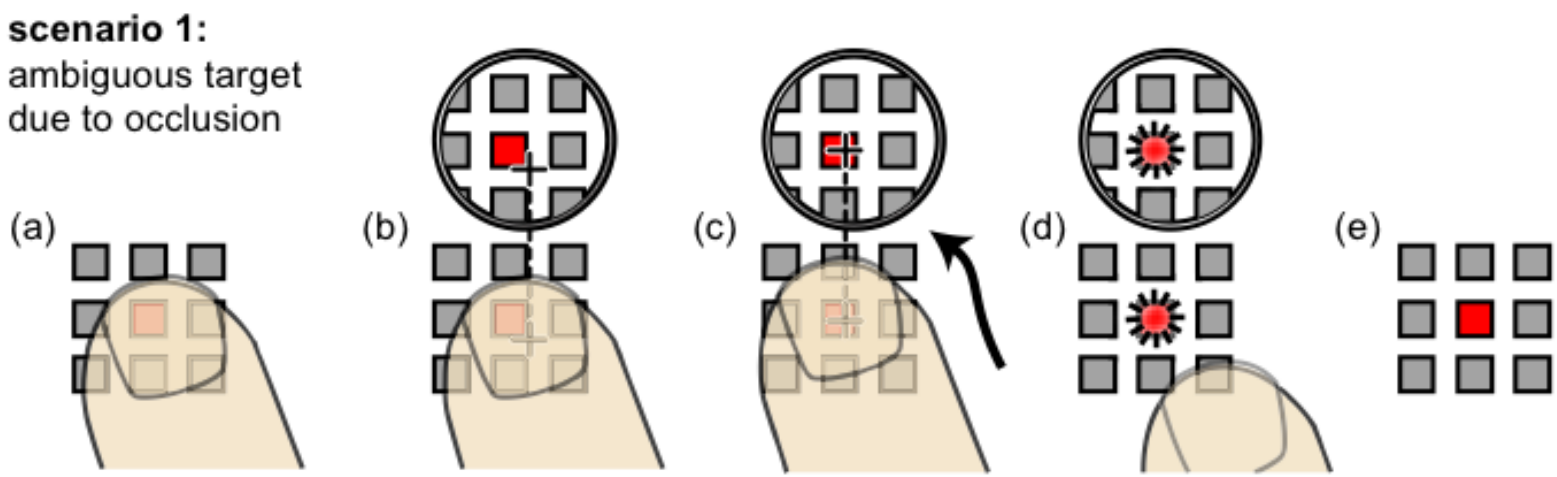
THE MANUAL INPUT SESSIONS: "NEGDROP"
golan levin / zach lieberman . 2004

THE "FAT FINGER" PROBLEM



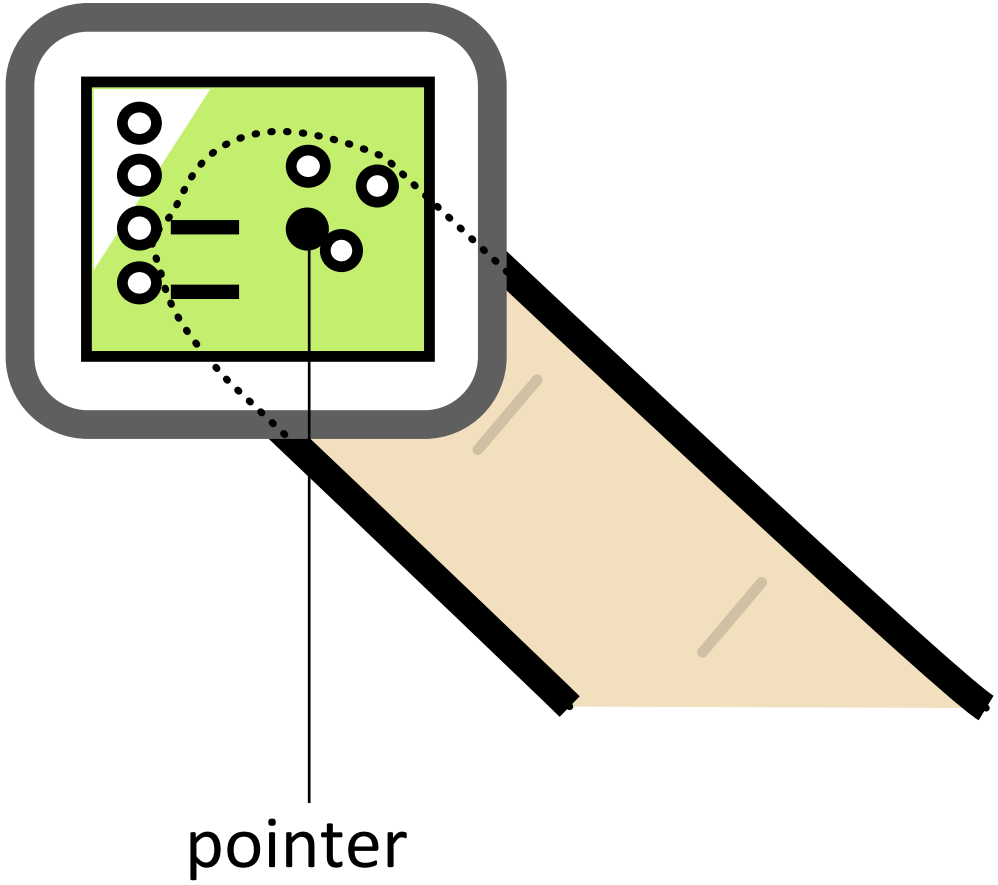
Graphics: Patrick Baudisch, nanoTouch

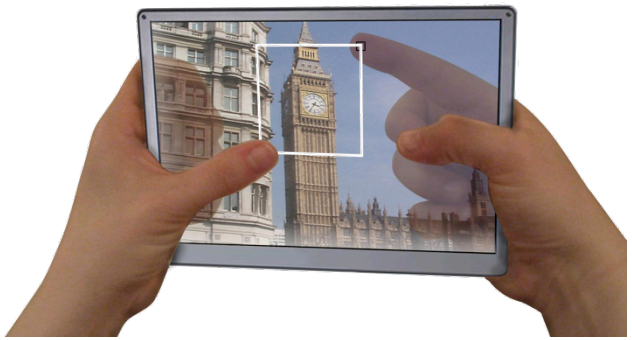
A SOFTWARE SOLUTION



Graphics: D. Vogel, P. Baudisch - Shift

A HARDWARE SOLUTION: USE THE BACKSIDE





Graphics: Patrick Baudisch, nanoTouch

LUCID TOUCH

LUCID TOUCH

HYBRIDS: MULTI-TOUCH ON MICE

Mouse 2.0: Multi-touch Meets the Mouse
Nicolas Villar, Shahram Izadi, Dan Rosenfeld, Hrvoje Benko, John Helmes, Jonathan Westhues, Steve Hodges, Eyal Ofek, Alex Butler, Xiang Cao and Billy Chen.
Proceedings of UIST 2009.

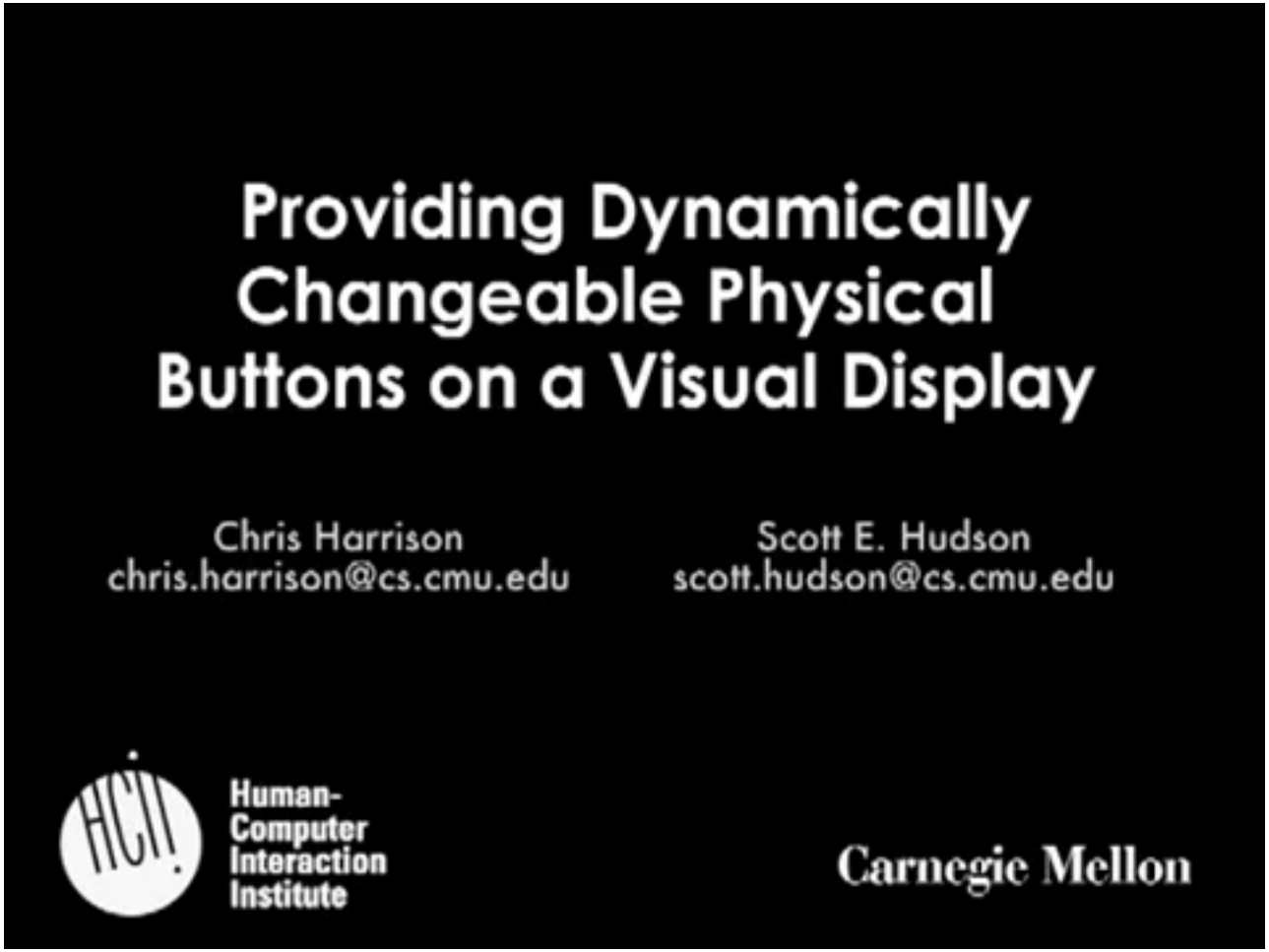
Mouse 2.0: Multi-touch Meets the Mouse
Nicolas Villar, Shahram Izadi, Dan Rosenfeld, Hrvoje Benko, John Helmes, Jonathan Westhues, Steve Hodges, Eyal Ofek, Alex Butler, Xiang Cao and Billy Chen.
Proceedings of UIST 2009.

FTIR Mouse

Applies the principle of Frustrated Total Internal Reflection to illuminate a user's fingers, and uses a camera to track multiple points of touch on its curved translucent surface.

DYNAMICAL BUTTONS

Chris Harrison, et al



ABRACADABRA

Chris Harrison, et al

SKINPUT

Chris Harrison, et al

TOUCHE

Chris Harrison, et al

Touché:

Enhancing Touch Interaction on

Humans, Screens, Liquids, and Everyday Objects

Munehiko Sato, Ivan Poupyrev, Chris Harrison

CHI 2012 Paper Video Figure

Disney Research
Pittsburgh