Touch Interfaces

Multi-touch displays

Input & interaction

Mobile design

In this course, we have mostly discussed the development of computer interfaces that rely on standard input devices (e.g., mouse, keyboards).

Mobile devices often rely on direct input using touch interaction or a stylus. Touch has become the dominant form of interaction for specific categories of devices (smartphones, tablets, table-tops)

Why touch?

Space optimization!

- Touch screens combine input and output, which optimizes the display/output area
- Allow interfaces to be customized

Multitouch Technology

- Who invented multi-touch?

- Steve Jobs iPhone launch:
  - https://www.youtube.com/v/9hUIxyE2Ns8?start=390 (6:30)
- Jeff Han
  - https://www.youtube.com/watch?v=89sz8ExZndc
- Bill Buxton
  - https://www.youtube.com/watch?v=Arrus9CxUiA

- What is actually new/novel in an iPhone? iPad?
  - Every single UI technology you leverage when using a computer is over 30 years old.

Sources

- “Input Technologies and Techniques” (Ken Hinckley and Daniel Wigdor, 2002)
- “Imprecision, Inaccuracy and Frustration: The Tale of Touch Input” (Benko and Wigdor, 2010)
- “Mobile UI Design Patterns” (Bank and Zuberi, 2014)
- User-Defined Gestures for Surface Computing (Wobbrock, Morris, Wilson, CHI 2009)
- Informing the Design of Direct Touch Tabletops (Chen et al., 2006)

Display

Input

Interaction

Design

Direct Touch Technology

Resistive

- –comprises of two transparent conductive layers separated by a gap
- –when pressure is applied, the two layers are pressed together, registering the exact location of the touch

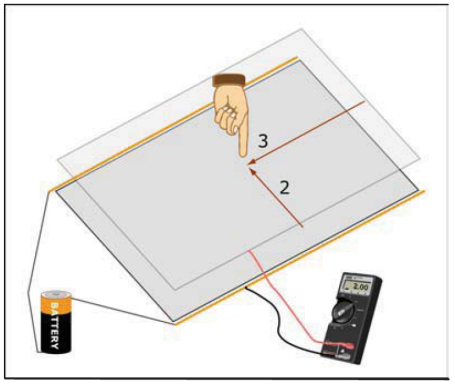


Image by MIT OpenCourseWare.

Capacitive

- –emitters at 4 corners of the screen
- –senses the conductive properties of an object (e.g., finger)
- –the location of the touch is determined indirectly from the changes in capacitance measured from four corners of the panel.

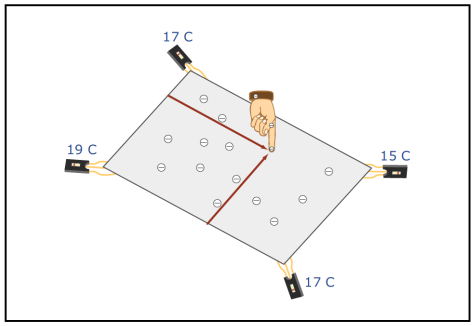
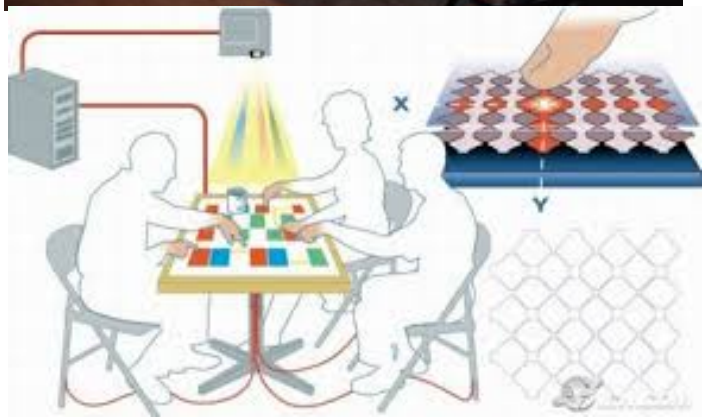


Image by MIT OpenCourseWare.

Mutual Capacitance

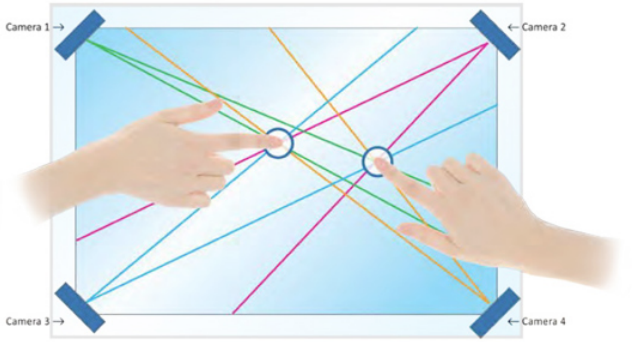
- Capacitors are arranged in a grid coordinate system
- Touch location is determined by measuring capacitance change at every point on the grid.
- Allows detection of simultaneous touches in multiple location, and tracking of multiple fingers
- Two distinct layers of material:
  - –driving lines carry current,
  - –sensing lines detect the current at nodes.



http://electronics.howstuffworks.com/iphone.htm

Inductive

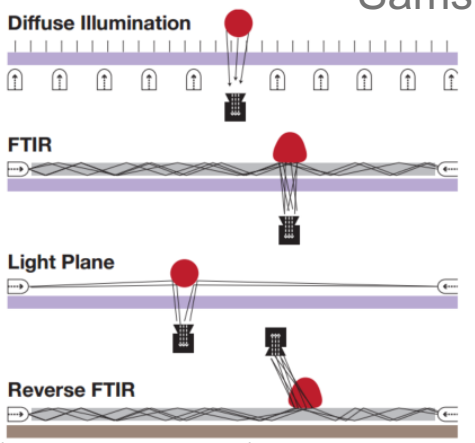
- –uses a magnetized stylus to return a magnetic signal to a sensing layer at the back of the display
- –e.g. Wacom in Samsung Galaxy Note
- –expensive!



Samsung S-Pen

Optical

- –cameras watch the surface
- –responds to everything
- –DSI or FTIR or Overlays (e.g. light plane)



https://www.touchsystems.com/opticaltouch

Display

Input

Interaction

Design

Indirect input device
- device is moved on one surface to indicate a point on another
- e.g., mouse, touchpads, multi-touch pads, trackballs, joysticks

Direct input device
- has a unified input and display surface
- e.g., touch screens, tablets with stylus

Stylus versus Finger

Stylus

by Cindy Packard

Finger

Stylus versus Touch

PROPERTY	PEN	TOUCH
Contacts	**1 point** A single well-defined point.	**1-10+ contact regions** Often with shape information (Cao et al. 2008).
Occlusion	**Small** (pen tip) But hand still occludes screen.	**Moderate** ("fat finger") to **Large** (pinch, palm, whole hand gestures)
Precision	**High** Tripod grip / lever arm affords precision, writing & sketching tasks.	**Moderate** Nominal target size for rapid acquisition via touch is about 10-18 mm^2 (Vogel and Baudisch 2007) (Sears 1993) (Lewis, Potosnak, and Magyar 1997)
Hand	**Preferred** hand	**Either** hand / **Both** hands
Elementary Inputs	Tap, Drag, Draw Path	Tap, Hold, Drag Finger, Pinch
Intermediary	**Mechanical Intermediary** Takes time to unsheathe the pen. Pen can be forgotten.	**None**: Bare-Handed Input Nothing to unsheathe, nothing to lose. No lever arm.
Acquisition Time	**High** (first use: unsheathe the pen) **Moderate** on subsequent uses: pen tucked between fingers.	**Low** No mechanical intermediary to acquire.
Buttons	**Barrel Button, Eraser** *(some pens)*	**None**
Activation Force	**Non-Zero** Tip switch or minimum pressure.	**Zero** (capacitive touch). Note that resistive touch requires some force.
False Inputs	**Palm Rejection:** Palm triggers accidental inputs, fingers drag on screen while writing, etc. This is a difficult problem. Designs must accommodate incidental palm contact when it inevitably occurs.	**"Midas Touch Problem"** Fingers brush screen, finger accidentally rests on screen while holding device, etc. **"Chess Player's Syndrome"** Device senses touch when none occurred. Common problem on optical touch-screens.

"Input Technologies and Techniques" by Hinckely and Wigdor.

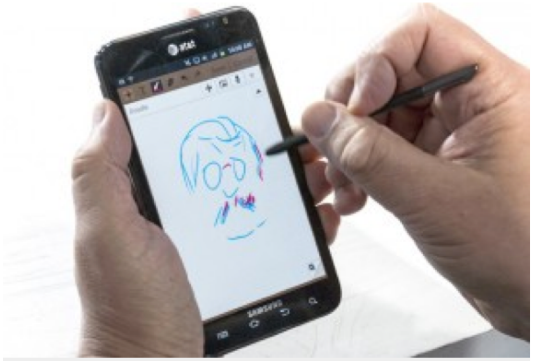
A common headline …

Is the Pen Mightier Than the Finger? Drawing Apps Boost Sales of Stylus

BY MARK MILIAN | MARCH 29, 2012 3:19 PM EDT | POSTED IN ANDROID, APPLE, APPS, GOOGLE, MOBILE, POSTS, STEVE JOBS | 0 COMMENTS

Even Steve Jobs didn't see this one coming: The stylus is back.

The late Apple co-founder derided the idea of using plastic pens with smartphones and tablet computers. "Who wants a stylus?" Jobs said in 2007. "You have to get 'em and put 'em away, you lose 'em. Yuck! Nobody wants a stylus." Then in 2010 after the release of the iPad, he said, "If you see a stylus, they blew it."



Photographer: David Paul Morris/Bloomberg

Is the stylus now in style? Makers of the pen have seen an increase in sales.

http://go.bloomberg.com/tech-blog/2012-03-29-is-the-pen-mightier-than-the-finger-drawing-apps-boost-sales-of-stylus/

I would rather draw with my fingers on the dusted windshield of my car than draw with [on] the phone.

Touch

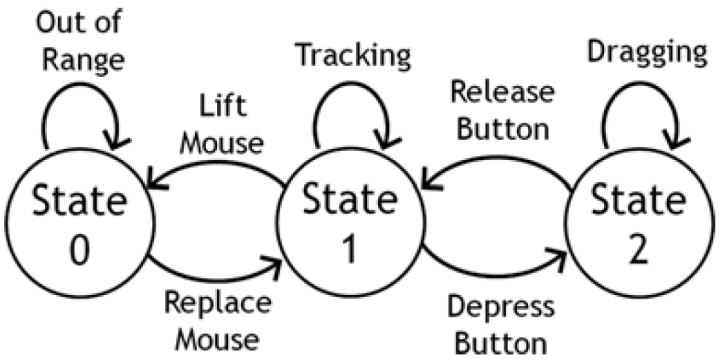
- How many points of contact are supported?
- Is the touch reported as an (x,y) position, or is the contact area or pressure sensed and reported?

Stylus

- Is a specialized stylus required, or can the surface detect any hard objects?
- Can pen contacts be distinguished from touch contacts?
- Can pen contact and touch be sensed simultaneously?

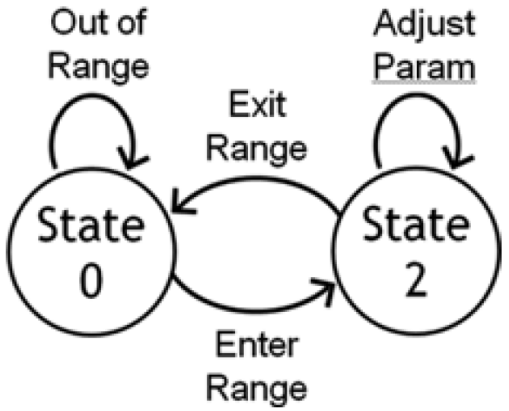
Touch Input: Input States

Mouse input typically supports 3-states (not touching, dragging, mouse-down)



Touch-input only supports 2-states (i.e. touching or not-touching the screen).

Pen can support 2 states (passive stylus) or 3 states (active stylus)



Pressure is the measure of force that users exerts on an input device.

- available on many pen-operated devices and some laptop touchpads
- also on new iPhone screens with 3D touch.

Contact area sensing

- Proxy for pressure sensing on most touch-screen devices
- Focus on changes in contact area as a controllable parameter

Touch Input: Challenges

Allowing input with the human finger is non-trivial

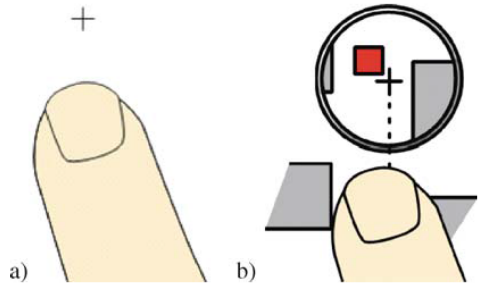
5 major challenges:

1. Finger occlusion
2. Reduced precision
3. Touch feedback ambiguity problem
4. Lack of hover state
5. Physical constraints

#1: The Fat Finger Problem

Occlusion:

- the user's finger occludes the target before touching the display
- a common technique to to display cursor at a fixed offset, but this breaks direct manipulation paradigm.



Imprecision:

- the touch area of the finger is many times larger than a pixel of a display

http://www.youtube.com/watch?v=qbMQ7urAvuc

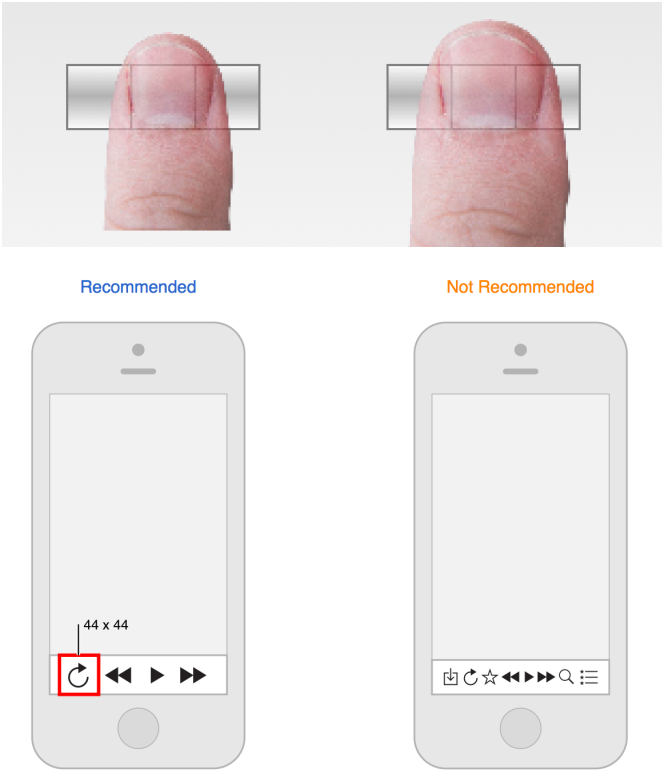
"Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input" by Hrvoje Benko and Daniel Wigdor

#1: The Fat Finger Problem

Apple:
Originally recommended 44x44 pixels – but that's dependent on display density.
44pixels on iPHone 3 was 10.5mm
Later updated to points; doesn't currently (appear to) include recommendations

Microsoft:
recommended 9mm; minimum 7mm; minimum spacing 2mm

Nokia:
recommend 10mm, minimum – 7mm, minimum spacing 1mm



source: iOS Human Interface Guidelines

#2: Ambiguous Feedback

When interacting with a traditional system, users feel a physical "click" when they depress the mouse button.

On a touch screen devices, users are missing this haptic feedback.

In case of unsuccessful actions, users is usually left to deduce the cause of error from little or no application feedback

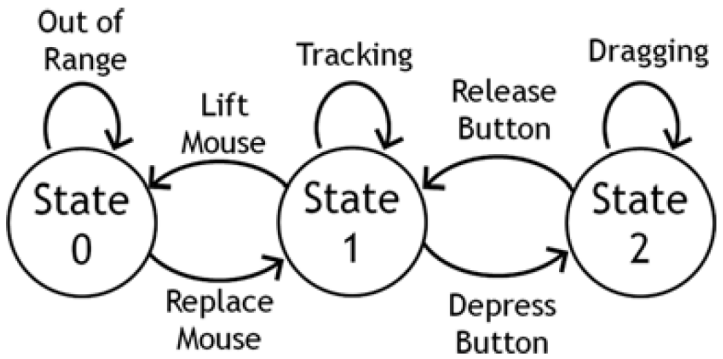
- system is non responsive?
- hardware failed to detect input?
- input delivered to wrong location?
- input does not map to expected function?
- e.g., tapping on an object using a mouse versus a finger

"Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input" by Hrvoje Benko and Daniel Wigdor

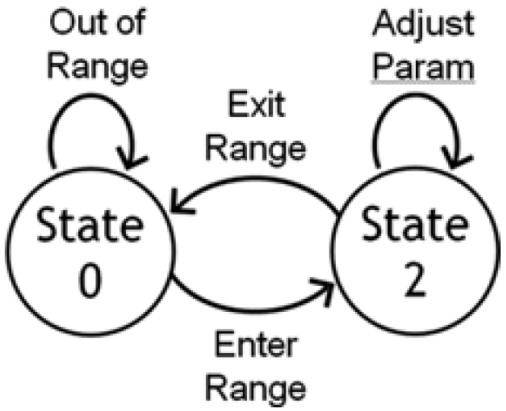
#3: Lack of Hover State

Having a third-state allows for hover.

This is useful, in that it allows users to preview one's action before committing to that action.



On touch screen devices, the hover state is missing.



"Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input" by Hrvoje Benko and Daniel Wigdor

#4: Multi-touch Capture

- In WIMP system, controls have “captured” and “un-captured” states.
- In multi-touch, multiple fingers may capture a control simultaneously, leading to ambiguity.
- when is click event generated?
  - e.g., Microsoft Surface: “tap” (~click) events are generated for buttons only when the last capture contact is lifted from the control.
  - e.g., DiamondSpin: “click” events are generated every time a user taps a button, even if another finger is holding it down”.
- over-capture: multi-touch controls captured by more than 1 contact simultaneously (e.g., selecting the thumb of a slider with two fingers can mean that it will not track directly under a single finger when moved.)

“Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input” by Hrvoje Benko and Daniel Wigdor

#5: Physical Constraints

Touch input relies on the principle of direct manipulation, i.e., user places their fingers onto an object, moves their fingers, and the object changes its position, orientation and size to maintain the contact points.

Direct touch breaks when movement constraints are reached (e.g., moving beyond bounds, or resizing past size limits).

Solution:

- elastic effects (e.g., apple iPhone scrolling past a list)
- snapping
- “catch-up zones”
- limits reaching (hybrid pointing)

“Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input” by Hrvoje Benko and Daniel Wigdor

Display

Input

Interaction

Design

Mobile devices support multiple forms of interaction:

1. Keyboard
2. Direct Manipulation w. touch
3. Surface gestures
4. Voice

Vendors have also experimented with in-air gestures, facial recognition.

- These haven't been widely adopted yet.

Tasks often utilize one or more of these together.

Keyboards

Devices are optimized towards touch

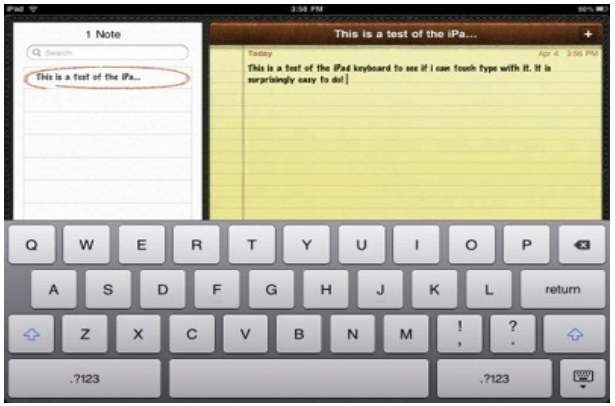
- Virtual keyboard for text
- Touch for pointing

Virtual keyboards

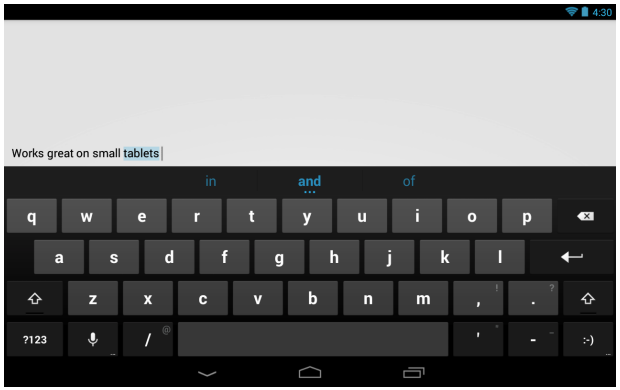
- improves the aesthetics of device,
- reduce thickness, size, and weight,
- Increase usable screen space.

However

- No tactile feedback
- Resting of hands compromised
- Bad option if device requires frequent text input



iOS



Android

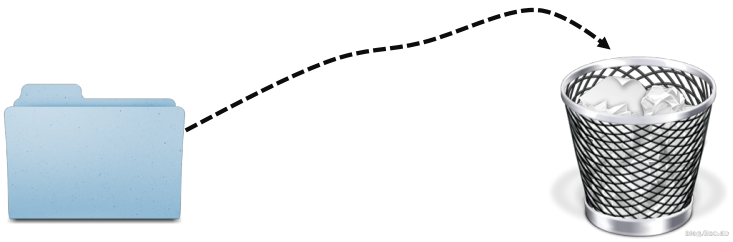
Direct Manipulation w. Touch

Touch interfaces utilize **direct manipulation (DM)**.

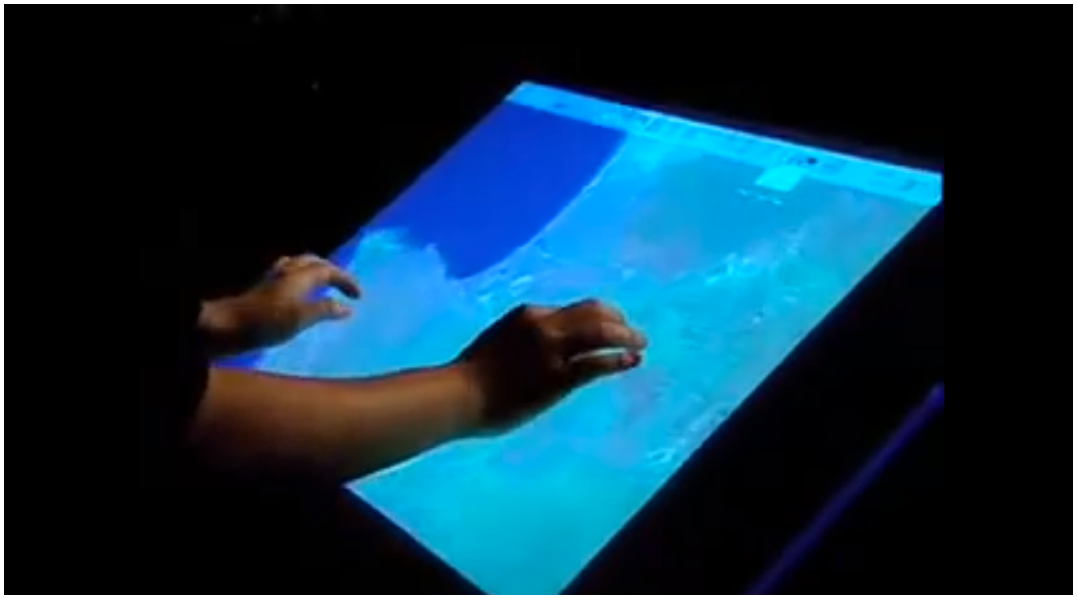
- A direct manipulation interface allows a user to **directly** act on a set of objects in the interface, similar to how we naturally use tools to manipulate objects in the physical world.

- Touch interfaces utilize direct input
  - low indirection (no temporal or spatial offsets)
  - high compatibility (similarity of action and effect)

"A user places their fingers onto an object, moves their fingers, and the object changes its position, orientation and size to maintain the contact points."
— Benko and Wigdor



Direct Manipulation w. Touch



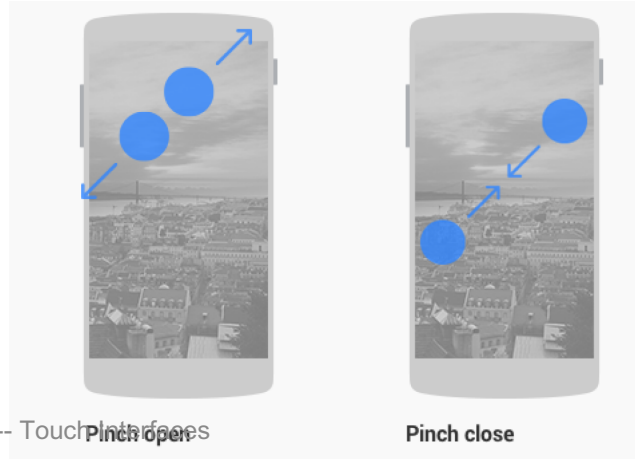
Jeff Han's "Lava Lamp" demonstration

https://www.youtube.com/watch?v=QKh1Rv0PlOQ&noredirect=1

Surface Gestures

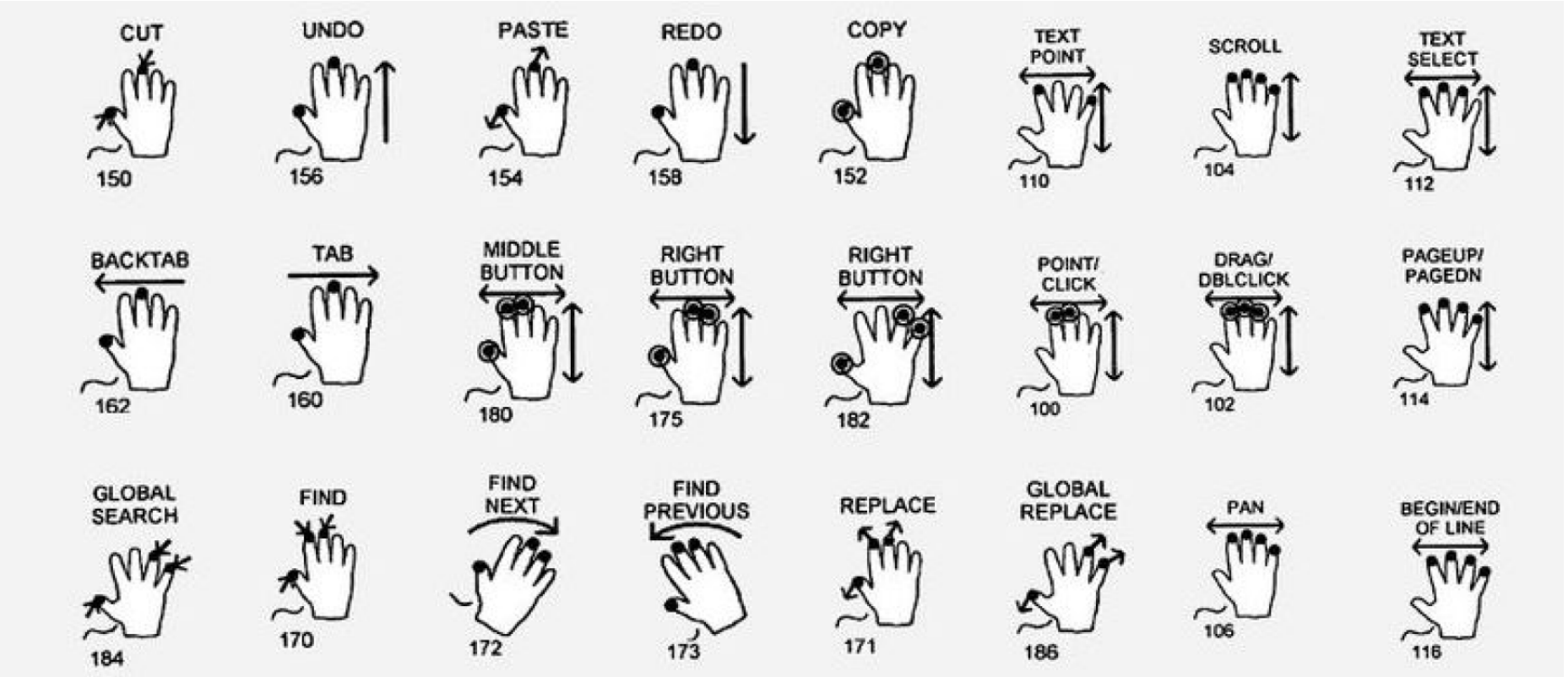
Interaction on a mobile device includes a full fledged set of **surface gestures** (“surface”, to distinguish from “in-air” gestures).

But what do these gestures mean?



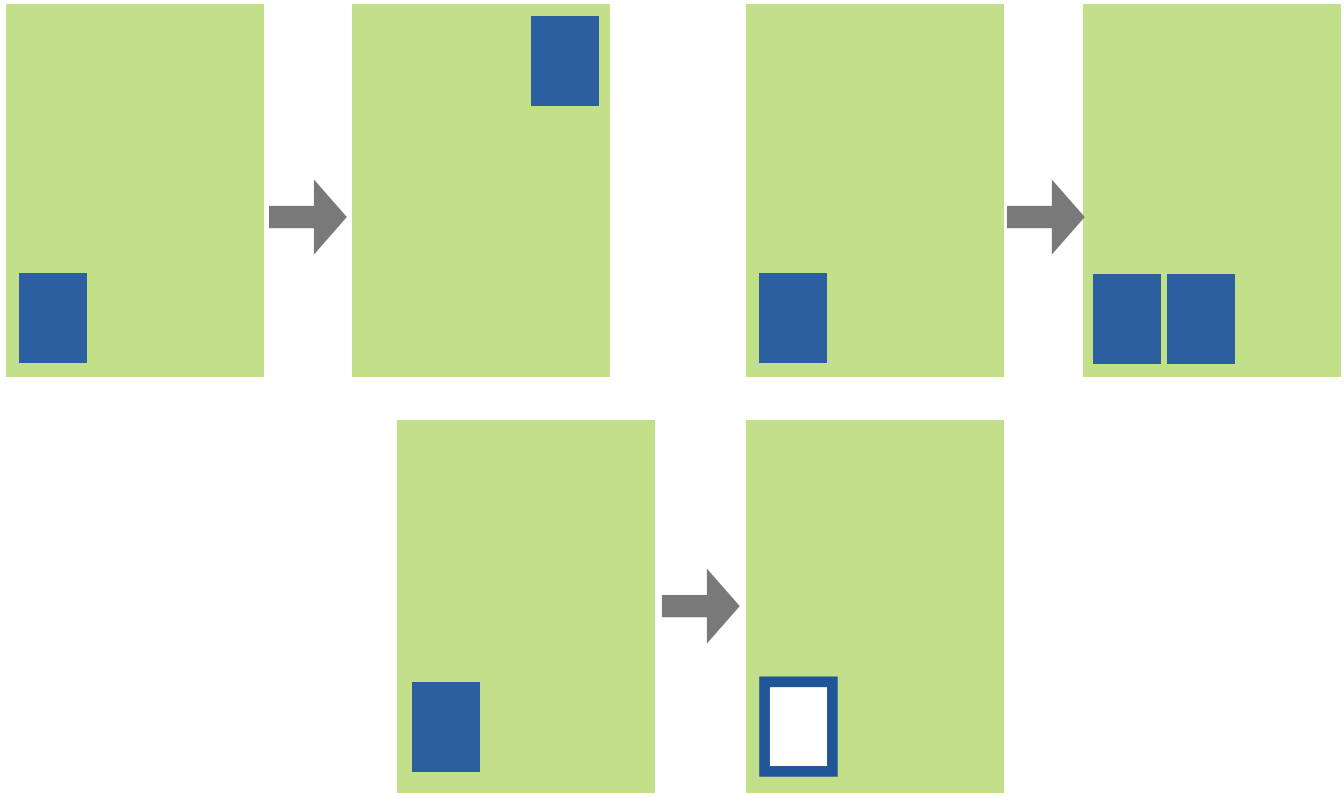
developer.android.com

Direct Manipulation via Gestures



patentlyapple.com

Gestures as "Natural" Input



What gesture would you use?

"Input Technologies and Techniques" by Hinckley and Wigdor.

Surface gestures are high varied — almost anything one can do with one's hand is accepted.

- Gestures have often been defined based on what was easy to implement, without much thought given to what makes sense for a user.

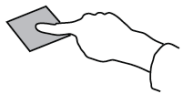
Wobbrock et al., asked questions

- What kinds gestures do non-technical users make?
- What are the important characteristics in gestures?
- How consistent are these gestures?
  - –Guess-ability study; think-aloud protocol and video analysis

"User-Defined Gestures for Surface Computing" by Wobbrock, Morris and Wilson, CHI 2009

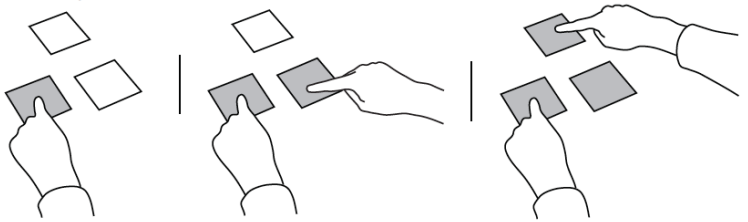
User-Defined Gestures

Select Single$_1$: tap



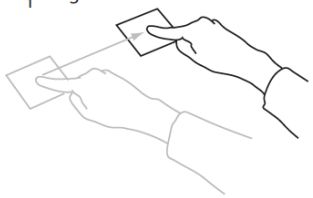
Select Single$_2$: lasso

Select Group$_1$: hold and tap

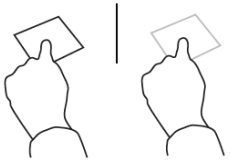


Select Group$_2$ and *Select Group*$_3$: Use *Select Single*$_1$ or *Select Single*$_2$ on all items in the group.

Move$_1$: drag

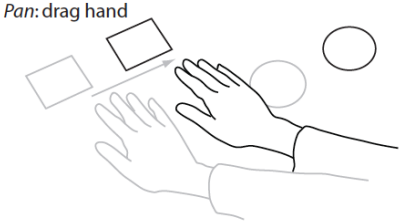


Move$_2$: jump

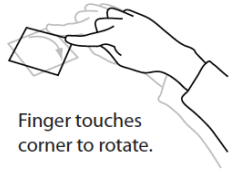


Object jumps to index finger location.

Pan: drag hand

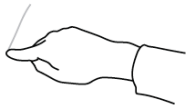


Rotate: drag corner



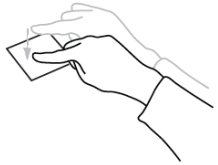
Finger touches corner to rotate.

Cut: slash

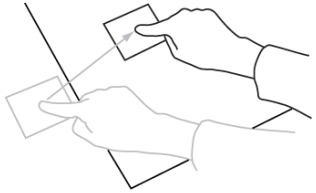


Cuts current selection (made via *Select Single* or *Select Group*).

Paste$_1$: tap



Paste$_2$: drag from offscreen



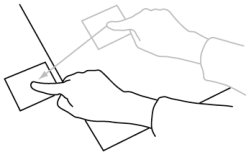
Paste$_3$: Use *Move*$_2$, with off-screen source and on-screen destination.

Duplicate: tap source and destination

After duplicating, source object is no longer selected.

User-Defined Gestures (2)

Delete$_1$: drag offscreen



Delete$_2$: Use *Move*$_2$ with on-screen source and off-screen destination.

Accept: draw check

Help: draw '?'

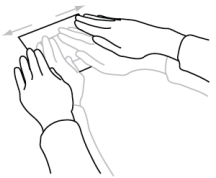
Reject: draw 'X'

Reject$_2$, *Reject*$_3$: If rejecting an object/dialog with an on-screen representation, use *Delete*$_1$ or *Delete*$_2$.

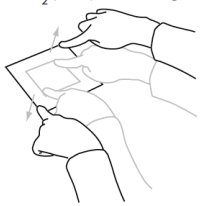
Menu: pull out

Undo: scratch out

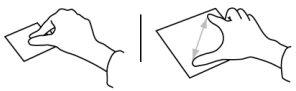
Enlarge (Shrink)$_1$: pull apart with hands



Enlarge (Shrink)$_2$: pull apart with fingers



Enlarge (Shrink)$_3$: pinch

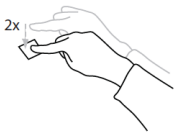


Enlarge (Shrink)$_4$: splay fingers

Zoom in (Zoom out)$_1$: pull apart with hands

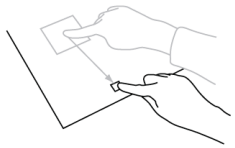
Zoom in (Zoom out)$_{2\text{-}4}$: Use *Enlarge (Shrink)*$_{2\text{-}4}$ performed on background.

Open$_1$: double tap



Open$_{2\text{-}5}$: Use *Enlarge*$_{1\text{-}4}$ atop an "openable" object.

Minimize$_1$: drag to bottom of surface



Minimize$_2$: Use *Move*$_2$ to move object to the bottom of the surface (as defined by user's seating position).

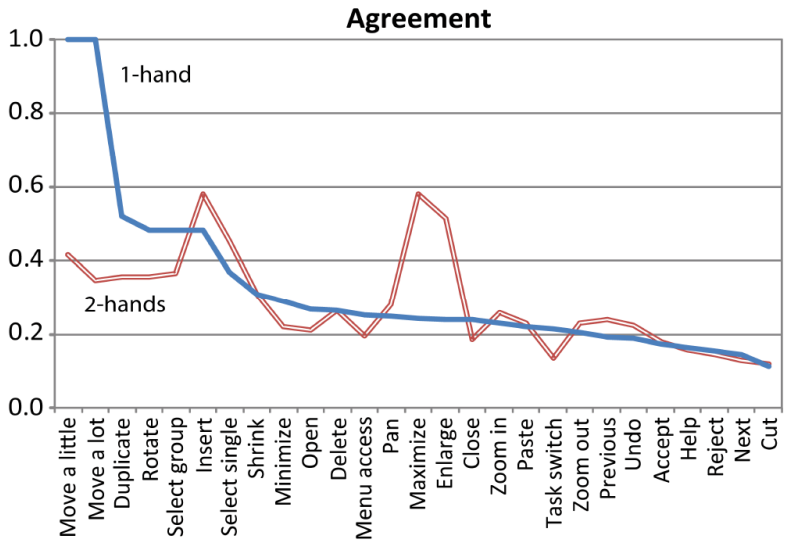
Next (Previous): draw line across object

The more simple the gesture, the more it was correlated/agreed-upon by users.

Old habits stick (*Legacy Bias* – Ruiz and Vogel):
- mouse-like one-point touches or paths.
- select, then gesture
- imaginary widgets (e.g., for the "close" action)

The three authors only came up with ~60% of the users' set. 19% of each author's gestures were never tried by participants.

About 72% of gestures were mouse-like one-point touches or paths



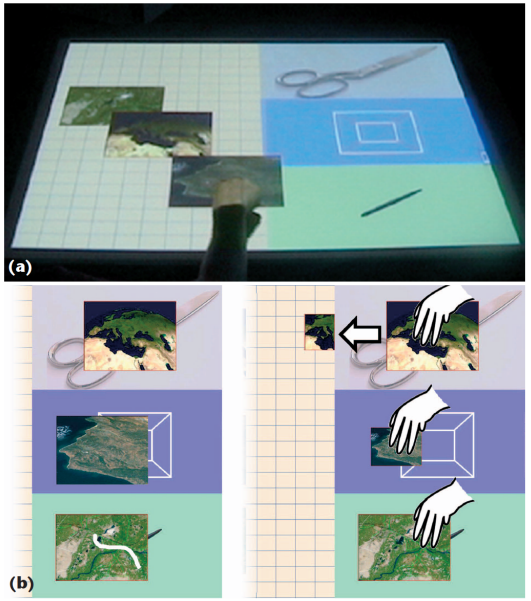
Challenges in Defining Gestures

Fat “body part” problem:
- information obscured under hand, arm, etc

Content Orientation
- people have a tendency to gather around the table for face-to-face interaction
- can affect group social dynamics, readability and performance.

Multiple, multi-touch input

Reach
- too much space
- many areas are unreachable



“Informing the Design of Direct Touch Tabletops” by Chen et al., 2006

Summary

Touch Interfaces introduce new challenges to the design and implementation of user interface.

To build effective user interfaces for mobile devices and tabletop, be aware of the limitations of the sensing display, input methods, then design interfaces and interaction to fit those limitations, e.g.,

- varying screen sizes (too small to too big)
- fat finger problem (occlusion and imprecision)
- high-variable input (i.e., gesture) to output mapping
- ambiguity in input interpretation and feedback