Touch Interfaces

Jeff Avery
Touch Interfaces

In this course, we have mostly discussed the development of web interfaces, with the assumption that the standard input devices (e.g., mouse, keyboards) are used.

Many applications, including web applications, also need to work on mobile devices, which operate on direct input using touch sensing or a stylus.
Touch Interfaces

Display

Input

Interaction

Design

Implementation
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

Capacitive
• 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.
Mutual Capacitance

- Capacitors are arranged in a grid coordinate system.
- Touch location is determined by measuring capacitance change at every individual point on the grid.
- Allows detection of simultaneous touches in multiple locations, 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
Direct Touch Technology

Inductive
- uses a magnetized stylus to induce an electro-magnetic field in a sensing layer at the back of the display
- expensive!

Optical
- cameras watch the surface
- responds to everything

Samsung S-Pen

https://www.touchsystems.com/opticaltouch
PixelSense

- shines infrared back light onto contact (finger, object)
- sensors detect light reflected back from the contact, and convert light signals to electrical values
- values reported from all the sensors are used to create a picture of what’s on the display
- picture is analyzed using image processing techniques
- output is sent to the PC, including the corrected sensor image and various contact types

Touch Interfaces

Display

Input

Interaction

Design

Implementation
Input Devices

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

Finger

by Cindy Packard
Stylus versus Touch

<table>
<thead>
<tr>
<th>PROPERTY</th>
<th>PEN</th>
<th>TOUCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contacts</td>
<td><strong>1 point</strong></td>
<td><strong>1-10+ contact regions</strong></td>
</tr>
<tr>
<td></td>
<td>A single well-defined point.</td>
<td>Often with shape information (Cao et al. 2008).</td>
</tr>
<tr>
<td>Occlusion</td>
<td><strong>Small</strong> (pen tip)</td>
<td><strong>Moderate</strong> (“fat finger”) to <strong>Large</strong> (pinch, palm, whole hand gestures)</td>
</tr>
<tr>
<td></td>
<td>But hand still occludes screen.</td>
<td></td>
</tr>
<tr>
<td>Precision</td>
<td><strong>High</strong></td>
<td><strong>Moderate</strong></td>
</tr>
<tr>
<td></td>
<td>Tripod grip / lever arm affords precision, writing &amp; sketching tasks.</td>
<td>Nominal target size for rapid acquisition via touch is about 10-18 mm² (Vogel and Baudisch 2007) (Sears 1993) (Lewis, Potosnak, and Magyar 1997)</td>
</tr>
<tr>
<td>Hand</td>
<td><strong>Preferred</strong> hand</td>
<td><strong>Either</strong> hand / <strong>Both</strong> hands</td>
</tr>
<tr>
<td>Elementary Inputs</td>
<td>Tap, Drag, Draw Path</td>
<td>Tap, Hold, Drag Finger, Pinch</td>
</tr>
<tr>
<td>Intermediary</td>
<td><strong>Mechanical Intermediary</strong></td>
<td><strong>None</strong>: Bare-Handed Input</td>
</tr>
<tr>
<td></td>
<td>Takes time to unsheathe the pen.</td>
<td>Nothing to unsheathe, nothing to lose.</td>
</tr>
<tr>
<td></td>
<td>Pen can be forgotten.</td>
<td>No lever arm.</td>
</tr>
<tr>
<td>Acquisition Time</td>
<td><strong>High</strong> (first use: unsheathe the pen)</td>
<td><strong>Low</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Moderate</strong> on subsequent uses: pen tucked between fingers.</td>
<td>No mechanical intermediary to acquire.</td>
</tr>
<tr>
<td>Buttons</td>
<td><strong>Barrel Button, Eraser</strong> <em>(some pens)</em></td>
<td><strong>None</strong></td>
</tr>
<tr>
<td>Activation Force</td>
<td><strong>Non-Zero</strong></td>
<td><strong>Zero</strong> (capacitive touch).</td>
</tr>
<tr>
<td>False Inputs</td>
<td>Tip switch or minimum pressure.</td>
<td>Note that resistive touch requires some force.</td>
</tr>
<tr>
<td></td>
<td><strong>Palm Rejection</strong>: Palm triggers accidental inputs, fingers drag on screen while writing, etc.</td>
<td><strong>“Midas Touch Problem”</strong> Fingers brush screen, finger accidentally rests on screen while holding device, etc.</td>
</tr>
<tr>
<td></td>
<td>This is a difficult problem. Designs must accommodate incidental palm contact when it inevitably occurs.</td>
<td><strong>“Chess Player’s Syndrome”</strong> Device senses touch when none occurred. Common problem on optical touch-screens.</td>
</tr>
</tbody>
</table>

“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.”

Challenge #1: The Fat Finger Problem

Occlusion:
• the user’s finger occludes the target before touching the display
• a common technique 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
Challenge #1: The Fat Finger Problem

“Informing the Design of Direct Touch Tabletops” by Chen et al., 2006
Challenge #1: The Fat Finger Problem

Apple:
recommended 44x44 points,
44 points = 15mm  (hmmm….)

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

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

“Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input” by Hrvoje Benko and Daniel Wigdor
Challenge #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?
Challenge #3: Lack of Hover State

Hover state is nice 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
Challenge #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
Challenge #5: Physical Manipulation 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
SimPress

http://www.youtube.com/watch?v=EIPWkh0xaG8
Touch Interfaces

Display

Input

**Interaction**

Design

Implementation
The traditional interaction model for the desktop is WIMP, which stands for “Windows, Icons, Menus and Pointing.”

In a nutshell:

- Application objects are displayed in document windows
- Objects can be selected and sometimes dragged and dropped between different windows
- Commands are invoked through menus or toolbars, often bringing up a dialog box that must be filled in before the command’s effect on the object is visible.

Many commands are invoked indirectly, pulling users away from objects of interest.
Direct Manipulation

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 (or instruments) to manipulate objects of interest in the physical world.

Modern GUIs and Touch-input devices both rely on the principle of direct manipulation

“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: Affordances

Affordance:

- a quality of an object or of an environment which allows one to perform an action
- an affordance should *suggest* its uses; uses should be “discoverable”

— Norman, The Design of Everyday Things

Direct Manipulation:

- attempting to make *affordances* in the interface like affordances for *analogous* actions in the real world.
Direct Manipulation: Examples

Dragging a document to the trash

Changing the size of a shape by dragging a “handle”

Inserting characters in a document by pointing to where they should go (with a mouse/cursor/insertion point) and then typing

“Dialing” a phone number by pushing numbers on a virtual keypad

Playing a song using controls like a physical CD/DVD player

What is not direct manipulation in a GUI?
Direct Manipulation in a GUI

• WIMP GUIs don’t leverage Direct Manipulation consistently
  • Many commands are invoked indirectly
    • e.g. menus, dialog boxes, toolbars
      • Not direct manipulation
      • They are mediators that pull users away from objects of interest
  • Lots of objects in the interface are not objects of interest and can’t be manipulated
    • e.g. toolbar pallets, containers

— Beaudoin-Lafon [2000]
Direct Manipulation on Touch Interfaces

https://www.youtube.com/watch?v=QKh1Rv0PlOQ&noredirect=1
Bret Victor, Inventing on Principle (talk from CUSEC 2012)
http://vimeo.com/36579366
Below is a simplified digital adaptation of the analog state variable filter.

The coefficients and transfer function are:

\[
k_f = 0.26 \quad k_q = 0.159
\]

\[
H(z) = \frac{0.068}{1 - \frac{0.068}{1.891z^{-1}} - \frac{0.068}{0.959z^{-2}}}
\]

An example frequency response:

\[
F_c = 1.83 \text{ KHz} \quad Q = 6.31
\]
Direct Manipulation: Principles

Clear affordances: interaction is “intuitive” and obvious.

• There is a visible and continuous representation of the task objects and their actions. Consequently, there is little syntax to remember.

The task objects are manipulated by physical actions, such as clicking or dragging, rather than by entering complex syntax.

• Every operation or manipulation is syntactically legal.
• Operations are fast, incremental and self revealing.
• The effect of operations on task objects are immediately visible

(Almost) all actions are reversible; Users can explore without severe consequences.

(from User Interface Design & Evaluation, p. 213-214)
Direct Manipulation: Benefits

While interacting with DM interfaces, users feel as if they are interacting with the task object rather than with the interface, so they focus on the task rather than on the technology.

There is a feeling of direct involvement with a world of task objects rather than communication with an intermediary.

Clear affordances mean that users can discover through exploration; interfaces are learnable.
Direct Manipulation: Challenges

Accessibility issues:
• Visually impaired users can’t see the graphics
• no linear flow for screen readers (i.e. difficulty mapping)
• physically impaired may have difficulty with required movements

Switching between keyboard and pointer is time consuming

Analogies may not be clear
• Users need to learn meaning of visual representations
• Visual representations may be misleading

Not all interactions are valid!
• What does it mean to resize the trash-can?
• Users still need to determine which interactions are meaningful.
Direct Manipulation: Direct Touch Challenges

As mentioned before, touch input relies almost exclusively on the principle of direct manipulation.

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

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

The lack of hover state and a cursor means that we’re missing key visual feedback (i.e. knowing where the input was directed)

“Imprecision, Inaccuracy, and Frustration: The Tale of Touch Input” by Hrvoje Benko and Daniel Wigdor
Direct Manipulation on Touch-Enabled Devices

Interaction on a mobile device goes beyond clicking, hover, scrolling and keyboard shortcuts, to a full fledge set of gestures.
Direct Manipulation via Gestures

- **CUT**
- **UNDO**
- **PASTE**
- **REDO**
- **COPY**
- **TEXT POINT**
- **SCROLL**
- **TEXT SELECT**
- **BACKTAB**
- **TAB**
- **MIDDLE BUTTON**
- **RIGHT BUTTON**
- **RIGHT BUTTON**
- **POINT/CCLICK**
- **DRAG/DBLCLICK**
- **PAGEUP/PAGEDN**
- **GLOBAL SEARCH**
- **FIND**
- **FIND NEXT**
- **FIND PREVIOUS**
- **REPLACE**
- **GLOBAL REPLACE**
- **PAN**
- **BEGIN/END OF LINE**
What gesture would you use?

“Input Technologies and Techniques” by Hinckley and Wigdor.
Gestural Interaction

How should gestures map to various system functions?

Should gestures map to the most common tasks or the most complex?

Does each command require its own gesture—and if so, how many gestures can we reasonably expect users to learn?

How can hand gestures coexist with familiar point-based, mouse-like interaction?

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

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

Typically defined by system designers based on technical and recognition constraints.

Wobbrock et al., asked the question: 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
Designing Gestures

The software randomly presented 27 referents to 20 participants.

For each referent, participants performed 1-handed or 2-handed gestures while thinking aloud, then indicated whether they preferred 1 or 2 hands.

After each gesture, participants are asked to rate the gesture on “goodness” and “ease”.
User-Defined Gestures

- **Select Single:** tap
- **Select Single:** lasso

- **Select Group:** hold and tap

- **Select Group** and **Select Group**:
  - Use **Select Single** or **Select Single** on all items in the group.

- **Move:** drag
- **Move:** jump

- **Paste:** tap
- **Paste:** drag from offscreen

- **Paste:** Use **Move** with off-screen source and on-screen destination.

- **Cut:** slash

- **Rotate:** drag corner

- **Duplicate:** tap source and destination

Cuts current selection (made via **Select Single** or **Select Single**).
User-Defined Gestures

- **Delete**: drag offscreen
- **Accept**: draw check
- **Reject**: draw “X”
- **Menu**: pull out
- **Undo**: scratch out

- **Delete**: Use Move, with on-screen source and off-screen destination.
- **Help**: draw “I”
- **Reject**: Reject. If rejecting an object/dialog with an on-screen representation, use Delete₁ or Delete₂.

- **Enlarge (Shrink)**: pull apart with hands
- **Enlarge (Shrink)**: pull apart with fingers
- **Enlarge (Shrink)**: pinch
- **Enlarge (Shrink)**: splay fingers

- **Zoom in (Zoom out)**: pull apart with hands
- **Open**: double tap
- **Minimize**: drag to bottom of surface
- **Next (Previous)**: draw line across object

- **Zoom in (Zoom out)**: Use Enlarge (Shrink), performed on background.
- **Open**: Use Enlarge (Shrink) atop an "openable" object.
- **Minimize**: Use Move to move object to the bottom of the surface (as defined by user's seating position).
Designing Gestures

The more simple the gesture, the more it was correlated highly or agreed-upon.

Old habits stick:
• 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.
Direct Manipulation on Tabletop

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
Touch Interfaces

Display

Input

Interaction

Design

Implementation
Desktop vs. Mobile
Mobile Interaction

Display is Smaller
- large variety of sizes (phone, phablet, tablet)
- orientation changes from portrait to landscape
- large variety of spatial resolutions
Mobile Interaction

Navigation
- one app in the foreground, with other apps suspended
- each app fills the entire screen
- interaction is a sequence of screens

Responsiveness
- variable bandwidth

Minimal help
- no hints via hover
- needs to be intuitive and easy to learn

“Clear”

Steve Krug (“Don’t Make Me Think Revisited”)
It’s all about tradeoffs

“One way to look at design — at any kind of design — is that it’s essentially about constraints (things you have to do and things you can’t do) and tradeoffs (the less-than-ideal choices you make to live within the constraints).”

- Steve Krug (“Don’t Make Me Think Revisited”)
Interface Guidelines


http://developer.android.com/design
Help Users to Enter Information Quickly

The Right Data Entry Tool

Anticipate and Predict Input

“Mobile UI Design Pattern” (Bank and Zuberi)
Help Users to Know What Actions to Take

Highlight New Content

Access to Most Frequent Actions

“Mobile UI Design Pattern” (Bank and Zuberi)
Help Users to Know What Actions to Take

Make Actions Obvious

Accordance: Control vs Content

“Mobile UI Design Pattern” (Bank and Zuberi)
Utilize Real Estate and Avoid Clutter

Expandable Controls

Morphing Controls

“Mobile UI Design Pattern” (Bank and Zuberi)
Utilize Real Estate and Avoid Clutter

Hide Metadata

Hide Secondary Menus

“Mobile UI Design Pattern” (Bank and Zuberi)
Touch Interfaces

Display

Input

Interaction

Design

Implementation
Touch Event API (iOS and Android)

Each touch event includes three lists of touches:

- **touches**: a list of fingers currently on the screen
- **targetTouches**: a list of fingers on the current DOM element
- **changedTouches**: a list of fingers involved in the current event (e.g., the finger that was removed in a touched event)

<table>
<thead>
<tr>
<th>Event Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>touchstart</td>
<td>triggered when finger is placed on a DOM element</td>
</tr>
<tr>
<td>touchmove</td>
<td>triggered when finger is dragged along a DOM element</td>
</tr>
<tr>
<td>touchend</td>
<td>triggered when finger is removed from a DOM element</td>
</tr>
</tbody>
</table>
Event Handling in Touch Interfaces

```javascript
var obj = document.getElementById('id');

obj.addEventListener('touchmove', function(event) {
  // If there's exactly one finger inside this element
  if (event.targetTouches.length == 1) {
    var touch = event.targetTouches[0];
    // Place element where the finger is
    obj.style.left = touch.pageX + 'px';
    obj.style.top = touch.pageY + 'px';
  }
}, false);
```

Mouse + Touch

Ideally, your web application should support both touch and mouse events. 

Chrome: “Emulate Touch Events”

http://www.creativebloq.com/javascript/make-your-site-work-touch-devices-51411644
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