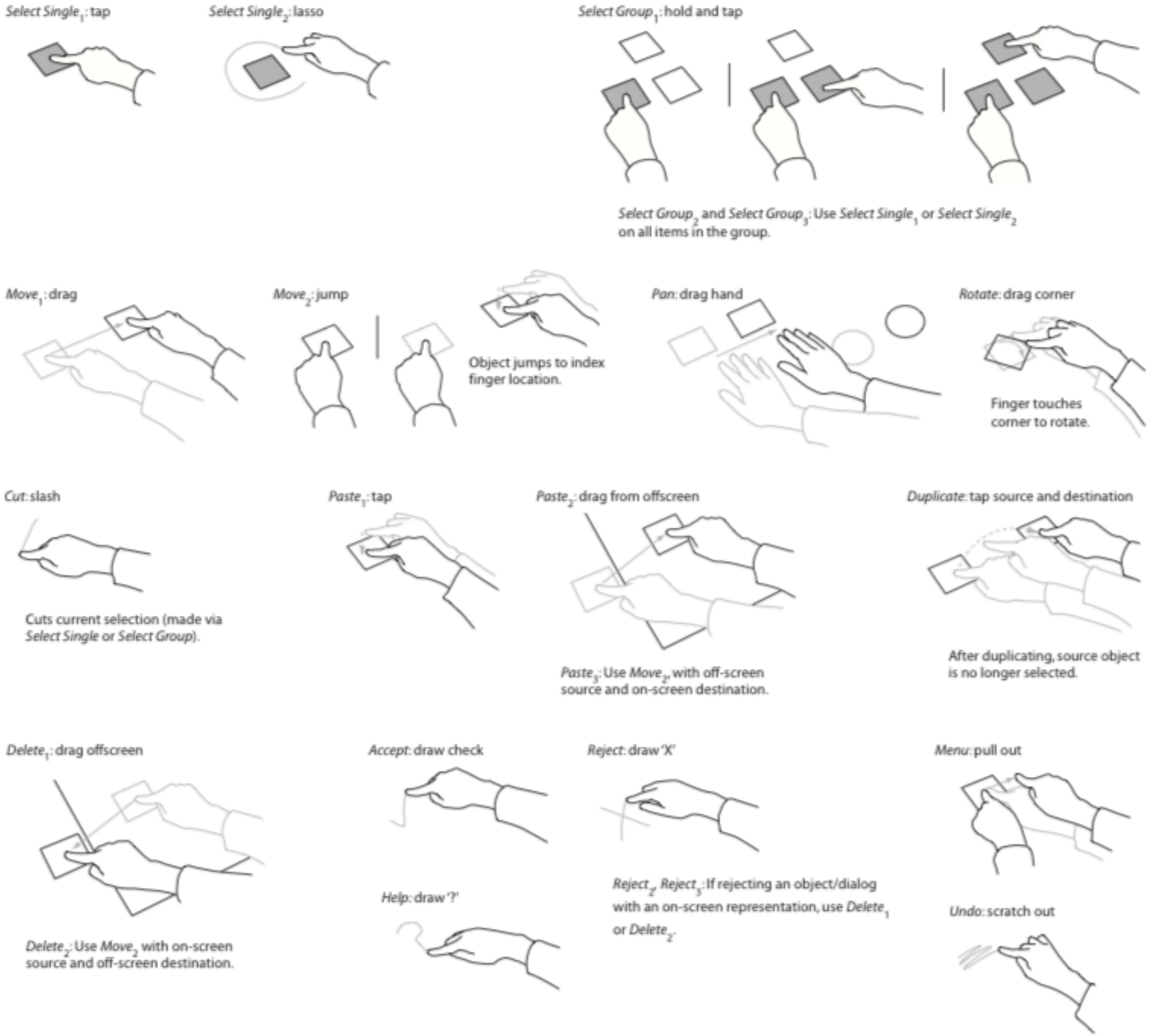
Gestural Interaction

CSCI 4849 Fall 2019



Working with gestures

- What is a gesture?
- Sensing gestures
- Recognizing gestures
- Designing gestures
- Evaluating gestures

What is a gesture anyway?

What is a gesture anyway?

- Interaction involving some **body movement** – usually hands, but not necessarily
- Often considered a discrete command: e.g., shake device to undo
- May occur on a surface, in air, on body
- May represent directional movements (1D), 2D or 3D shapes

Why this is important

- Gestures are increasingly becoming a core part of our user interfaces
- New input technologies lead to new gestures

Sensing gestures

Sensing gestures

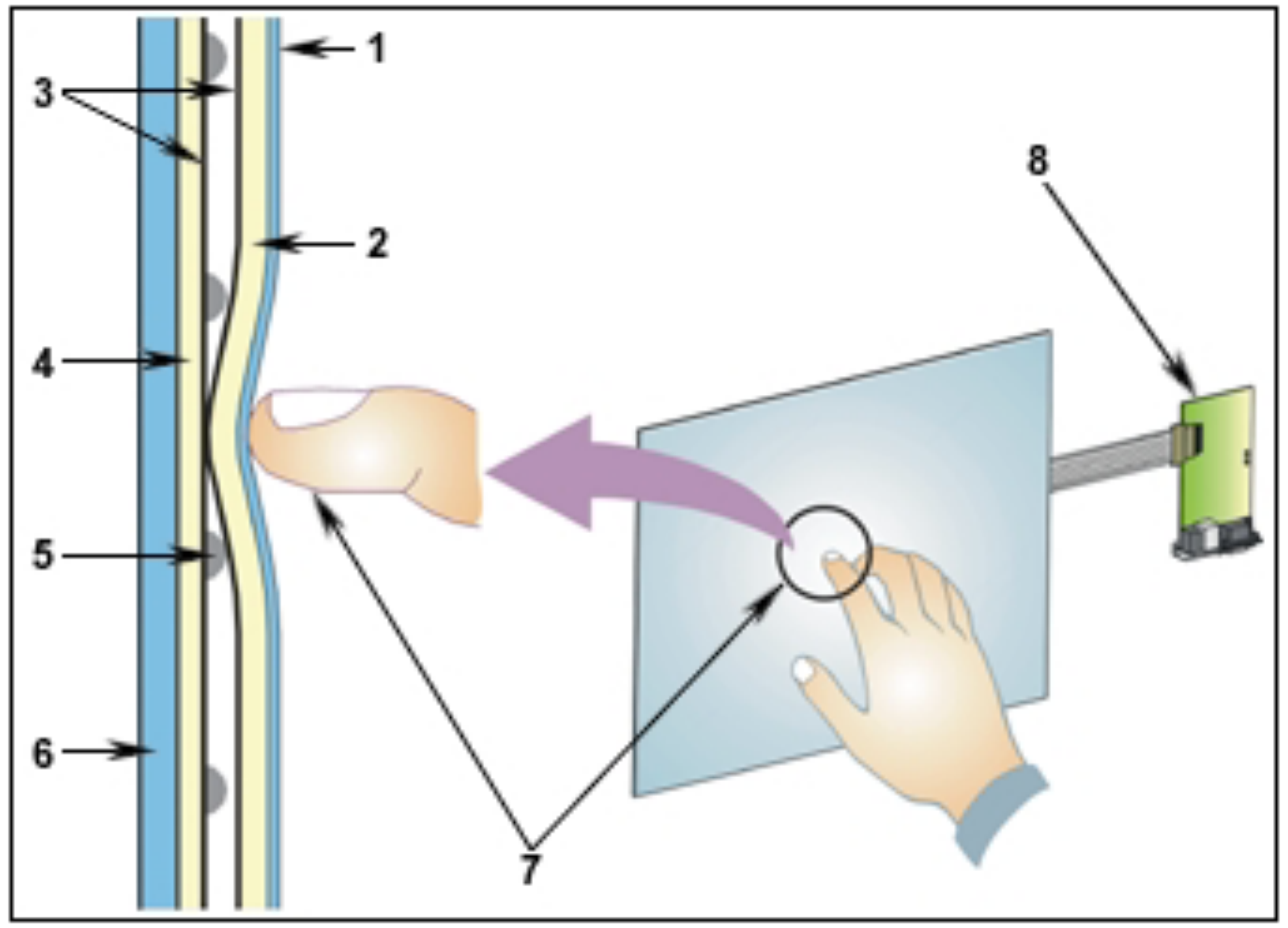
- How we sense gestures reflects what gestures are possible
- Common gesture input methods
  - Touch screen
  - Mouse/touchpad
  - 2D camera
  - Depth camera
  - Accelerometer/gyroscope
  - Pen gestures

Touch screens

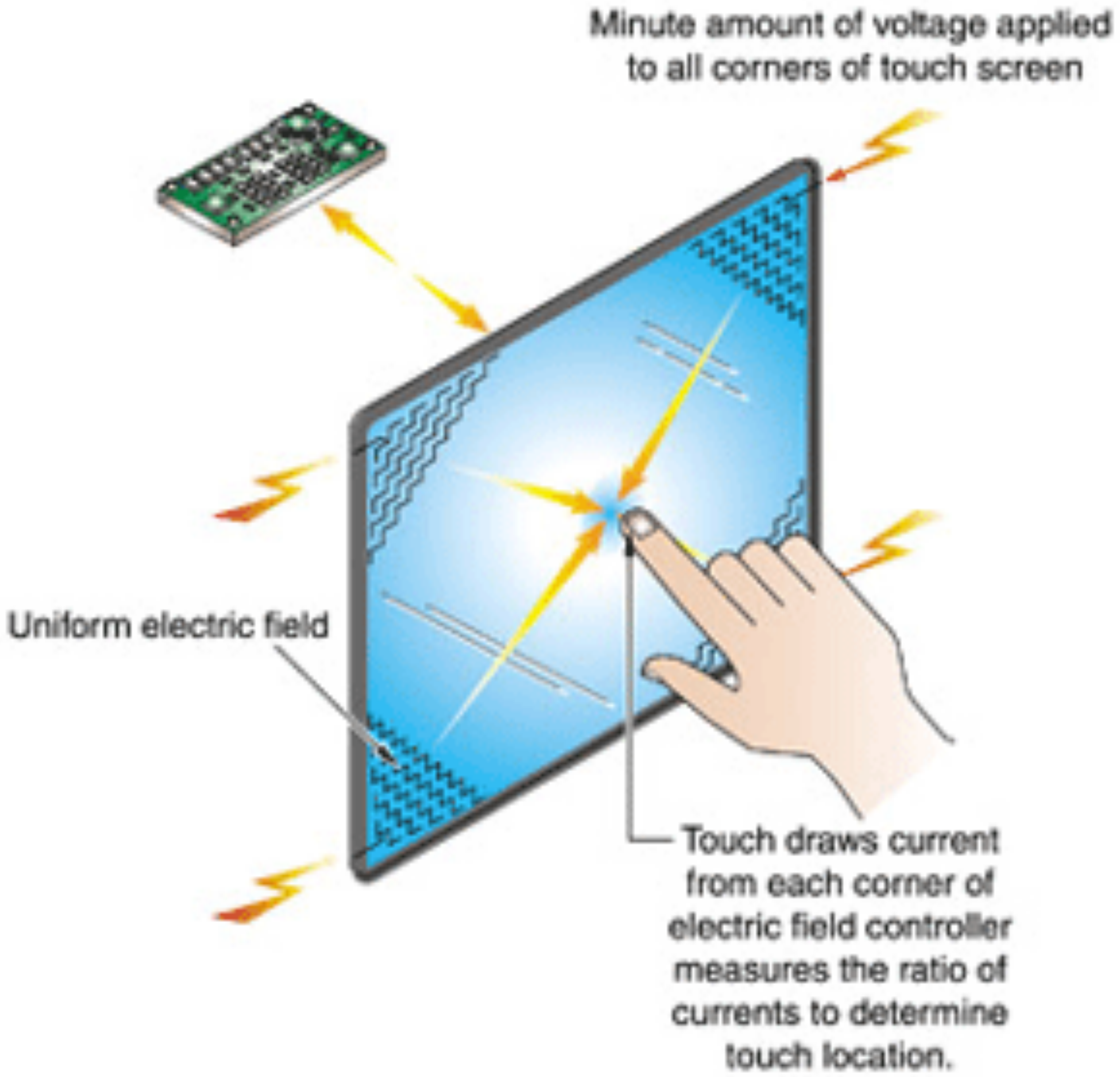
- Two main sensing technologies: resistive and capacitive

- **Resistive:** activated when user presses layers together
  - Requires force to activate
  - Typically averages multiple contacts to center point
  - Examples: airplane screens, Nintendo DS

- **Capacitive:** activated when contact made with human skin (or other electrically conductive object)
  - Requires electrically conductive contact
  - Can support gestures, multiple touches
  - Examples: most smartphones and tablets



Resistive touch screen technology [Image Credit: Chassis Plans]



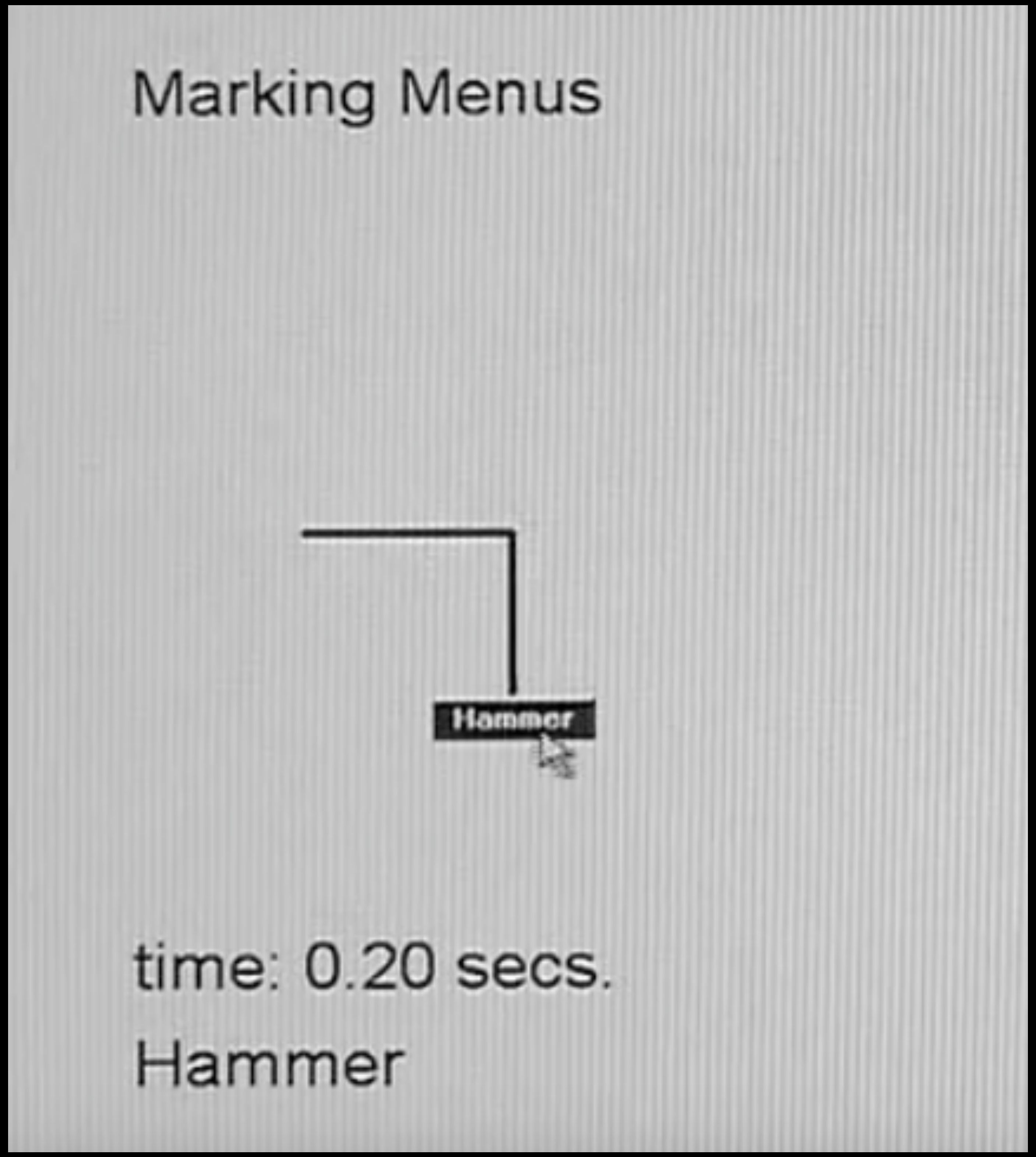
Capacitive touch screen technology [Image credit: Electrotest]

Touch screen gestures

- **Can typically measure:**
  - Movement, direction, speed, number of contacts
- **Can sometimes measure (depending on hardware):**
  - Pressure*, interactions off the surface*
- **Usually cannot measure:**
  - which fingers are being used, orientation of hand,

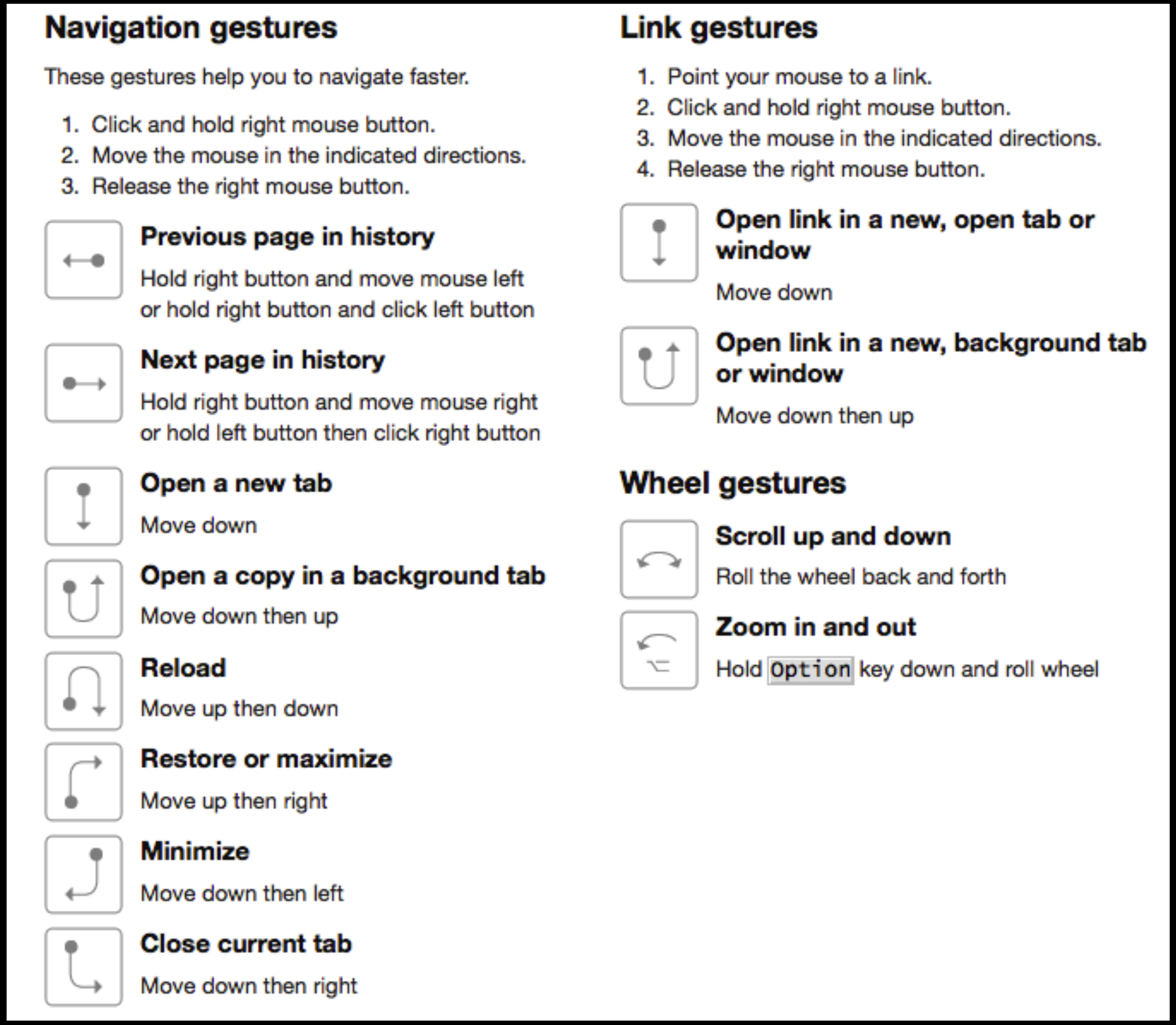
Mouse/touchpad

- Typically requires some mode switch to distinguish gestures from regular cursor movement (segmentation)
- Often involves “indirect” gestures – general movement rather than gesturing on a specific on-screen object
- Can be useful for expert use; e.g. marking menus



Mouse gestures

- Recall our three-state model
- Issues with accidental activation; segmentation
- Add another state (in this case, right mouse button)



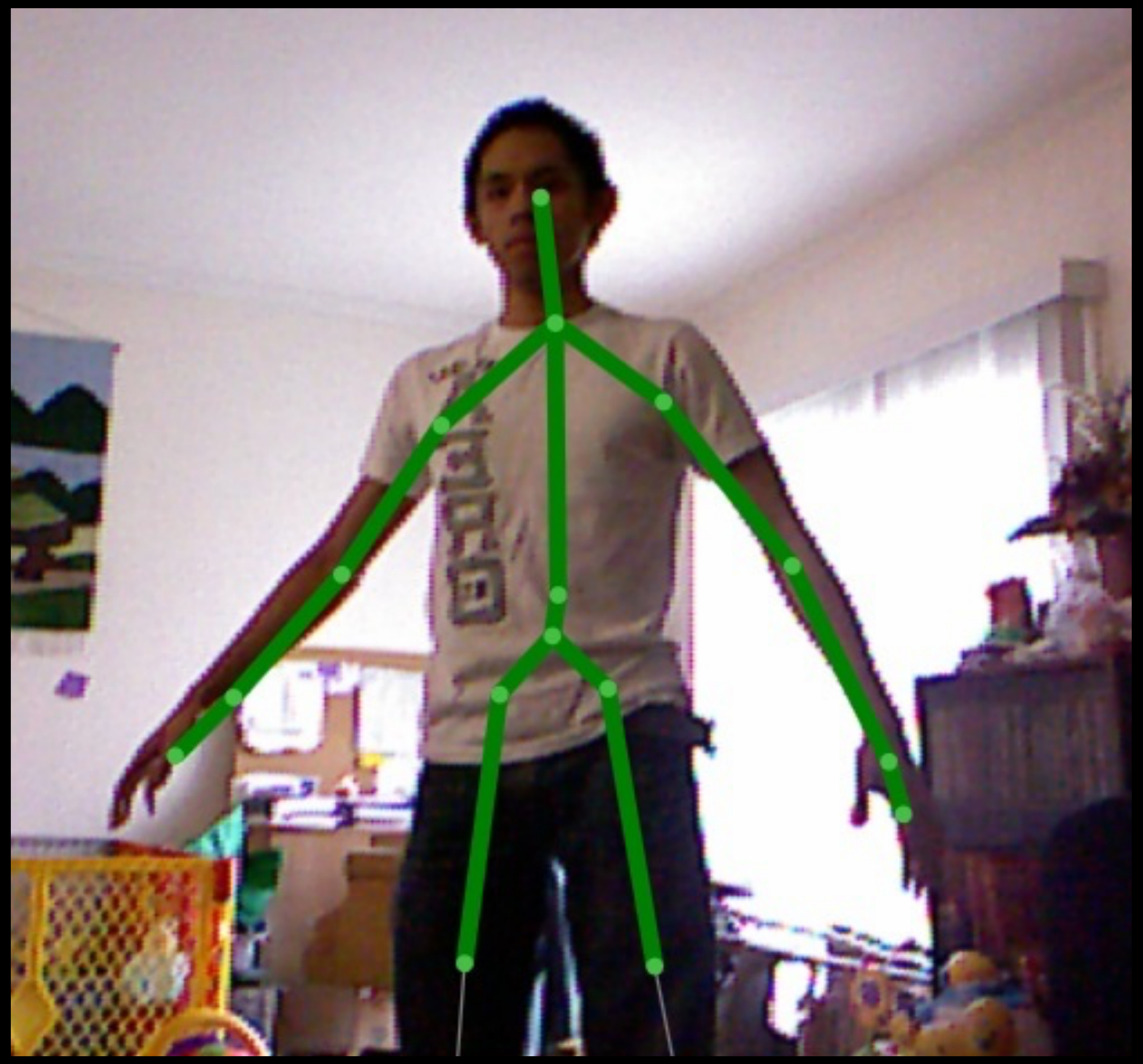
2D cameras

- Mostly used in research prototypes; hasn't taken off in commercial devices
- Can track hands and other objects using computer vision
- Requires marking hands or performing (sometimes noisy) image segmentation
- Difficult to determine **surface contact** (another segmentation problem)

Kane et al., Bonfire

Depth cameras (3D)

- Examples: Microsoft Kinect, Intel RealSense, LeapMotion
- Tracks 3D movement
- Can sometimes infer body parts
- Cons: may be less precise, not very portable, 3D gestures are difficult to perform, **gorilla arm**



Microsoft Kinect

Leap Motion

Accelerometer/gyroscope

- Use motion of device itself as gesture input
- **Accelerometer:** detect acceleration along device x,y,z axes
  - Can infer orientation (by detecting gravity)
  - Can detect thumps / whacks / collisions
  - Often requires big movements due to gravity, noise
  - Theoretically, can integrate acceleration to track velocity/position, in practice you can't
- **Gyroscope:** can detect angular movement
- **Inertial Measurement Unit (IMU):** may include accelerometer, gyroscope, and mafgnetometer; reports orientation

Case study: Wiimote

- Uses accelerometer to detect gestural movements
- Later augmented by MotionPlus (w/ gyroscope)
- Not accurate enough for pointing, supplemented by infrared pointing input via sensor bar

Pen/stylus gestures

- Can augment pen with sensors to detect position, orientation, etc.

- Can add buttons on pen barrel

- Easy to detect contact on surface

- Can sometimes detect “hover” state

Microsoft Surface Pen

Sensing via wireless radio

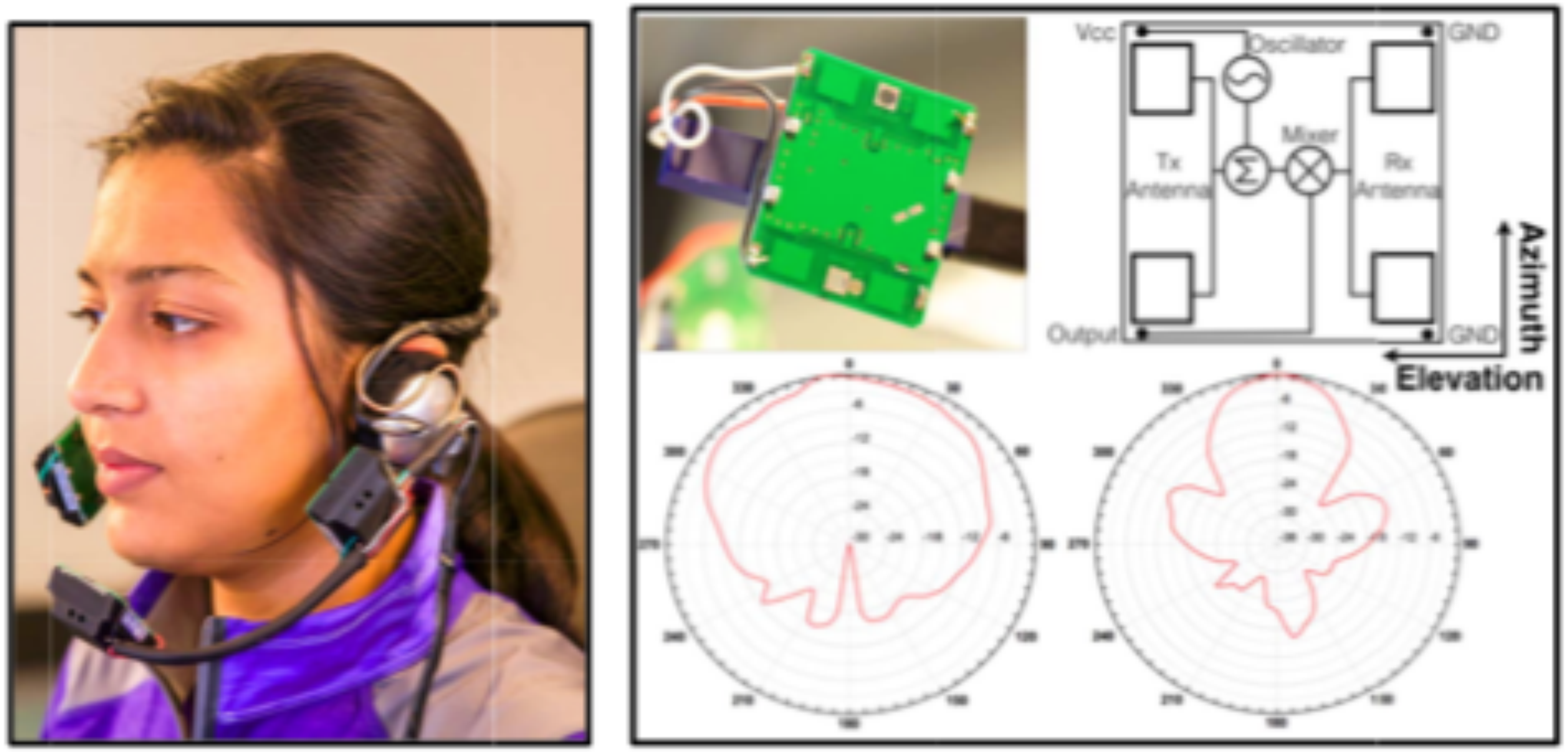


Figure 1. ***(Left)*** Tongue-in-Cheek is a non-contact facial gesture detection system. It is integrated into a pair of off-the-shelf headphones. ***(Right)*** The motion sensor module and Azimuth (bottom-left) and elevation (bottom-right) of the antenna beam pattern.

Augmenting the body

- Saponas et al., Optically sensing tongue gestures for computer input

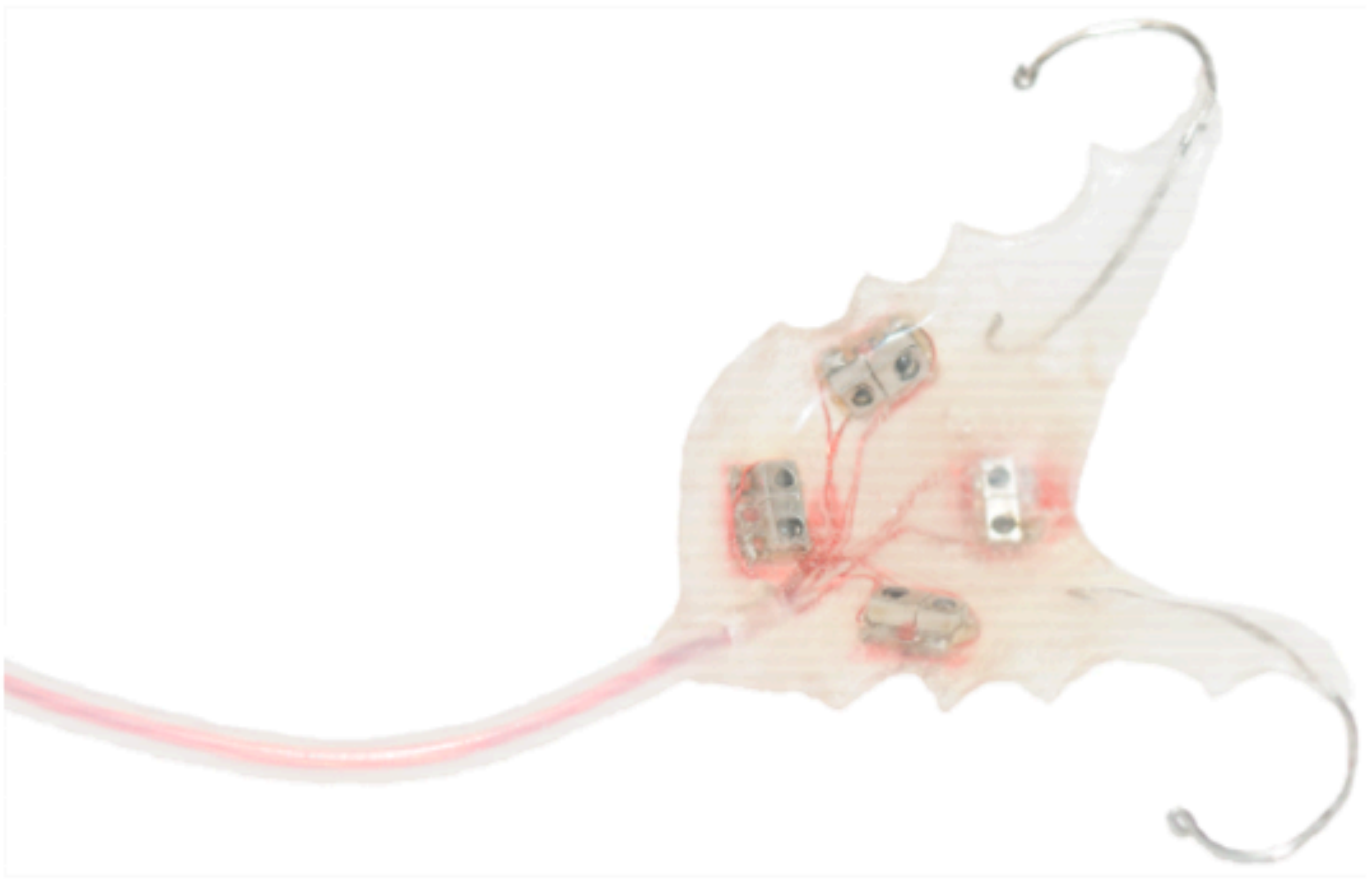


Figure 1. Our prototype optical tongue sensing retainer.

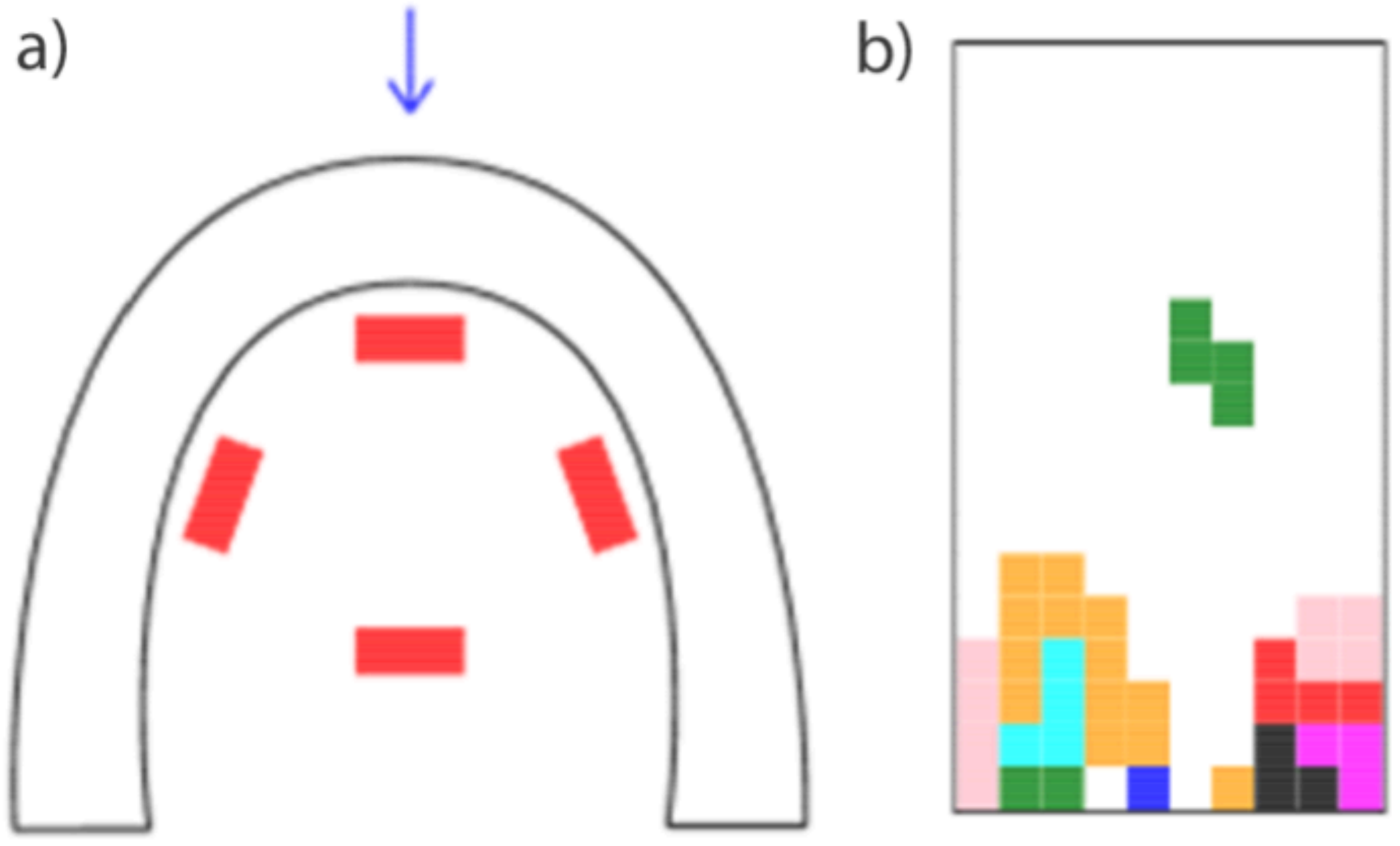


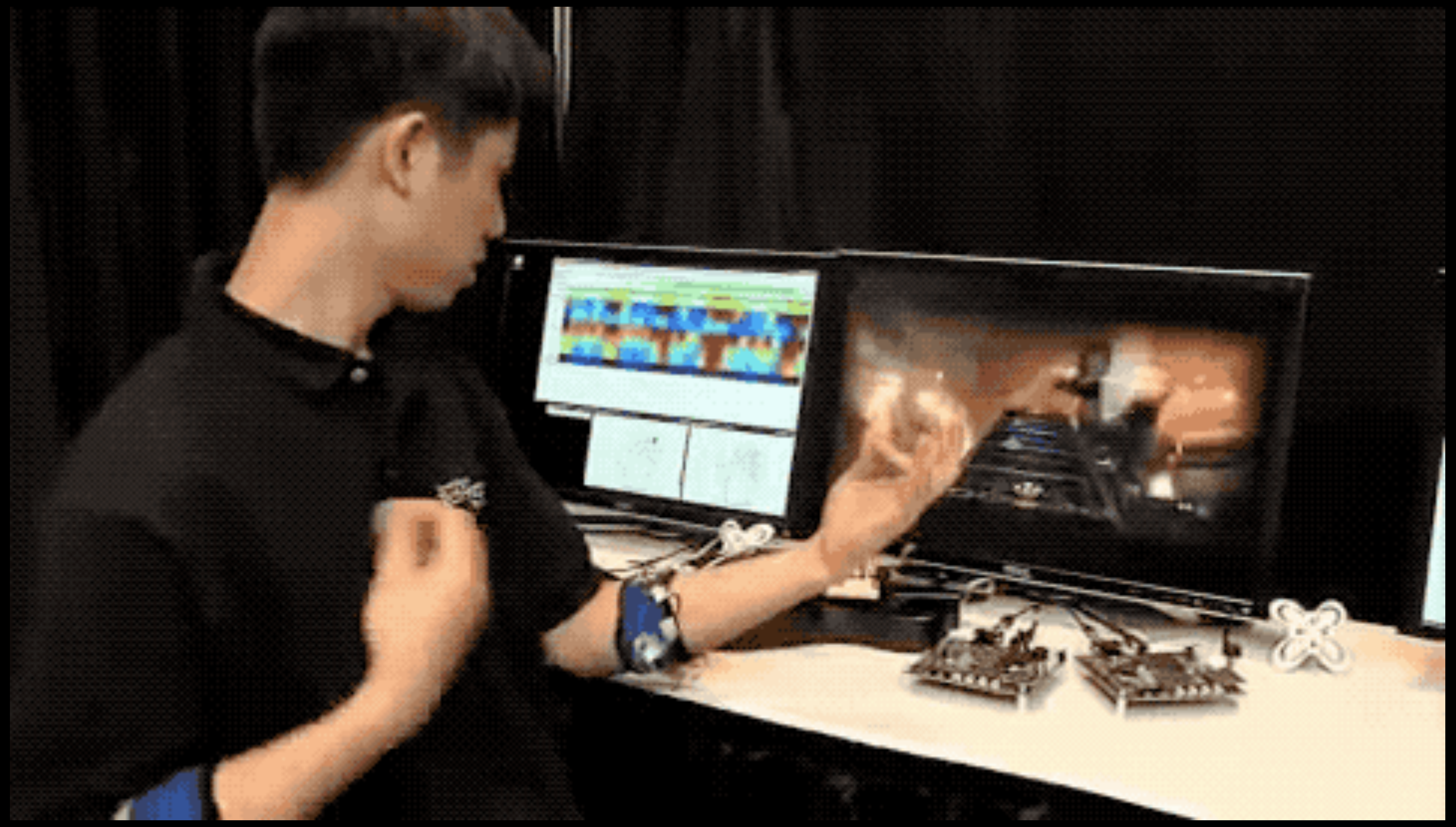
Figure 3. a) Sensor state visualization and recognized gesture feedback b) Tetris game

Designing gestures for weird devices

- Gesture-sensing retainer supports tongue swipes (left, right, front, back)
- Must balance sensing capability, reliability, usability

Other sensing technology

- Device sensors: camera, ambient light sensor, face tracker, eye tracker
- Body sensing: EEG, EMG
- Research opportunities to use new sensors in interesting ways



Saponas et al., [Muscle-Controlled User Interfaces]

NeuroPhone (Campbell et al. 2010)

- EEG-based mobile phone dialer
- Relatively limited signal from brain without going inside the skull
- So, uses scanning interface

Brain to Mobile Phone Interface

Andrew T. Campbell, Tanzeem Choudhury, Shaohan Hu, Hong Lu,
Matthew K. Mukerjee*, Mashfiqui Rabbi, Rajeev D. S. Raizada

"NeuroPhone"

* Contact Author

Recognizing gestures

Recognizing gestures

- Typical methods
  - Heuristic methods
  - Shape-based recognition
  - Machine learning-based methods

- Of course, the correct recognition method may depend on the problem, sensing method, gesture set, etc.

Heuristic gesture recognizers

- Can wire up gesture recognizer using existing UI events
- Specify gesture in terms of substeps
- Useful for prototyping

```
// swipe detector
function onTouchDown(float x, float y) {
  // save where we touched down
  Point startPoint = new Point(x,y);
}

function onTouchUp(float x, float y) {
  // where did we touch up?
  Point endPoint = new Point(x,y);
  // measure the difference
  diffX, diffY = endPoint - startPoint;

  // we have a heuristic for what constitutes a swipe
  if (diffX < -30 && -10 < diffY < 10)
    handleSwipeLeft();
}
```

Let’s try it

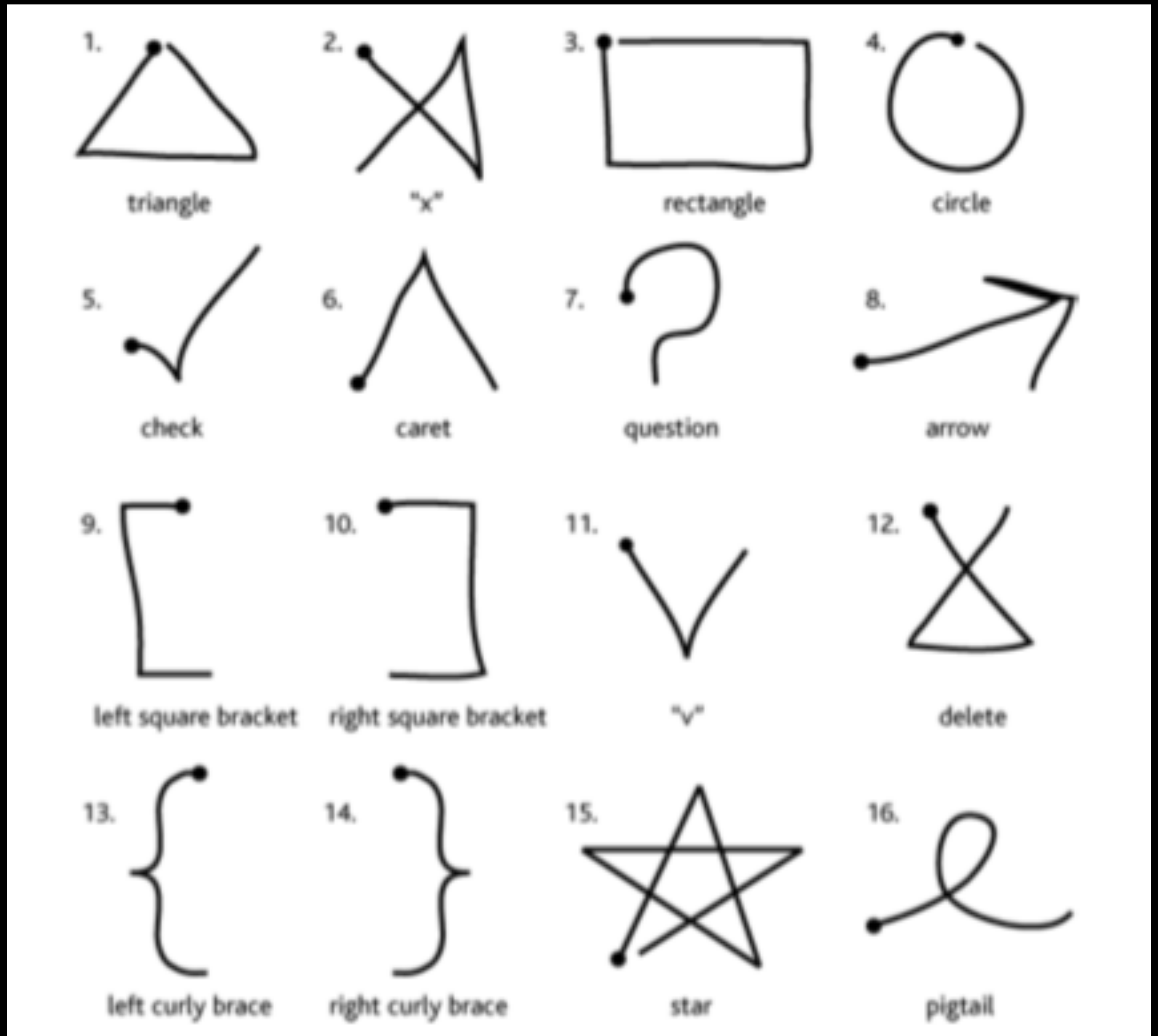
- Sketch out pseudocode for double tap

Limitations of heuristic approach

- Hard coded values may not represent what users actually do
- No capability to adapt behavior to user or context
- Clunky, but often our best option

Shape-based recognition

- Compare shape of gesture to set of reference gestures
- Common algorithms: Dynamic time warping, Rubine (1991), $1 recognizer
- **Limitations:** can be confused by direction of movement (clockwise or CCW), multi-stroke gestures



Wobbrock et al., $1 recognizer

Machine learning approaches

- Various algorithms: neural networks, SVMs, AdaBoost
- Can be more robust to individual differences, can be trained by example
- But, models are sometimes opaque, must choose appropriate features, more trial and error
- Good starting point: Fiebrink et al., Wekinator

Recognition challenges

- **Segmentation:** how do we know when a gesture begins or ends?
- Rest position / “off state”
- Assumptions of training data set
  - Size, speed, orientation, etc.
  - Does training data reflect all valid ways to perform the gesture?

Choosing gesture recognition methods

- Sense data as directly as possible
  - Touch screens are very precise; almost everything else is noisier
  - Measurement noise is cumulative (e.g., accelerometer -> position)
- Identify segmentation rules
  - Time-based, speed-based, auxiliary sensor

Adapting to different parts of the body

- Gestures can be performed by different body parts
- May be especially useful for accessibility
- But, consider variations in dexterity
  - Range of movement, speed, force, **conductivity**
  - May not always be symmetric
  - Decide whether to support ambidextrous use (or other body parts)

Next time

- Designing gestures
- Evaluating gestures

Evaluating gestures

Evaluating gestures

- How should we evaluate whether a gesture is good or bad?

Evaluation criteria

- Recognition accuracy
- Ergonomics / reachability
- Intuitiveness / learnability
- Social acceptability

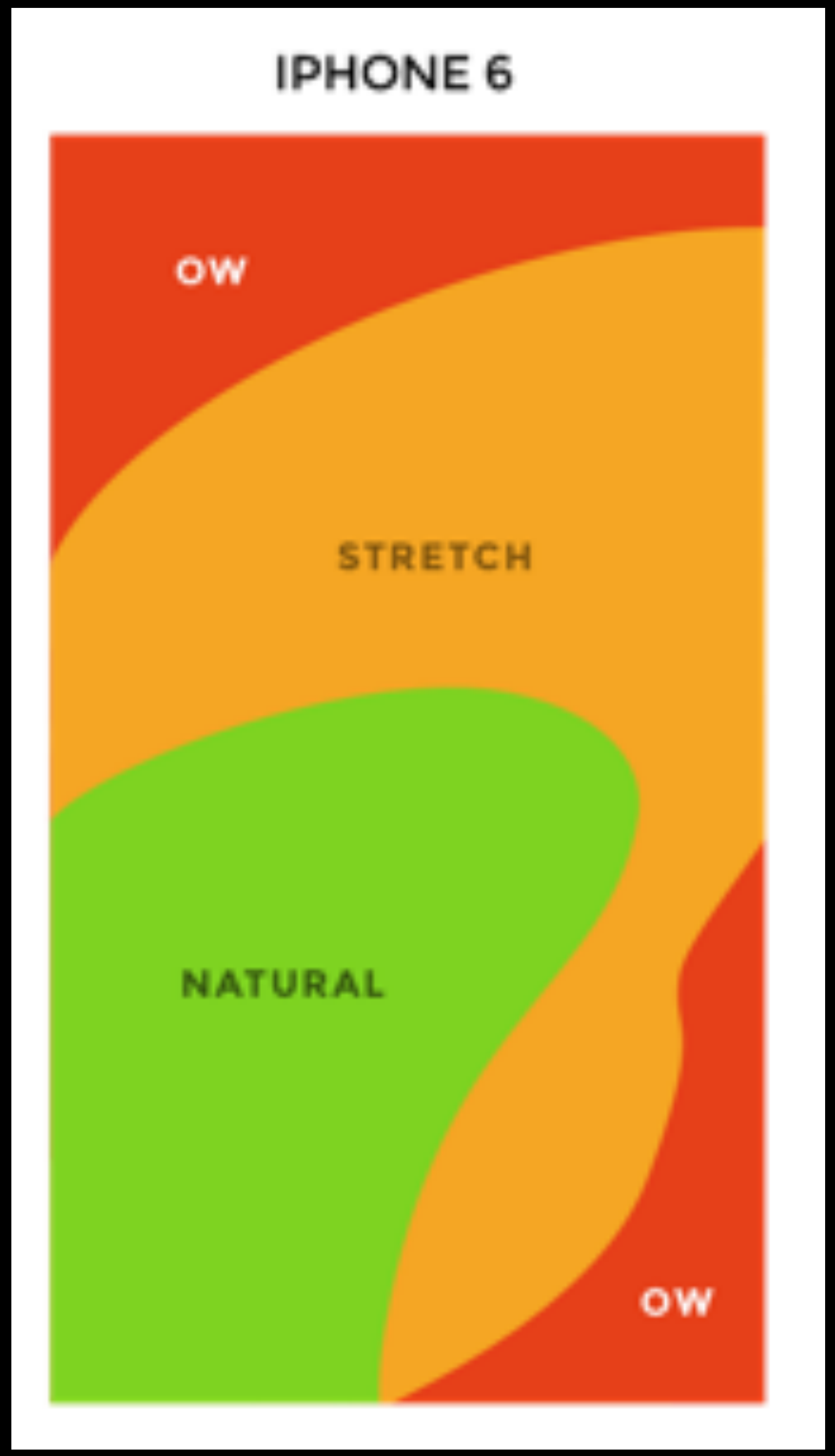
Recognition accuracy

- Types of errors: false positive, false negative, matched to other gesture
  - Costs/effects differ by error
- Users will expect high accuracy: 95-99%
- Identify commonly confused gestures and change them (or change recognizer)

Ergonomics

- Consider reach, physical effort, repetitive movements

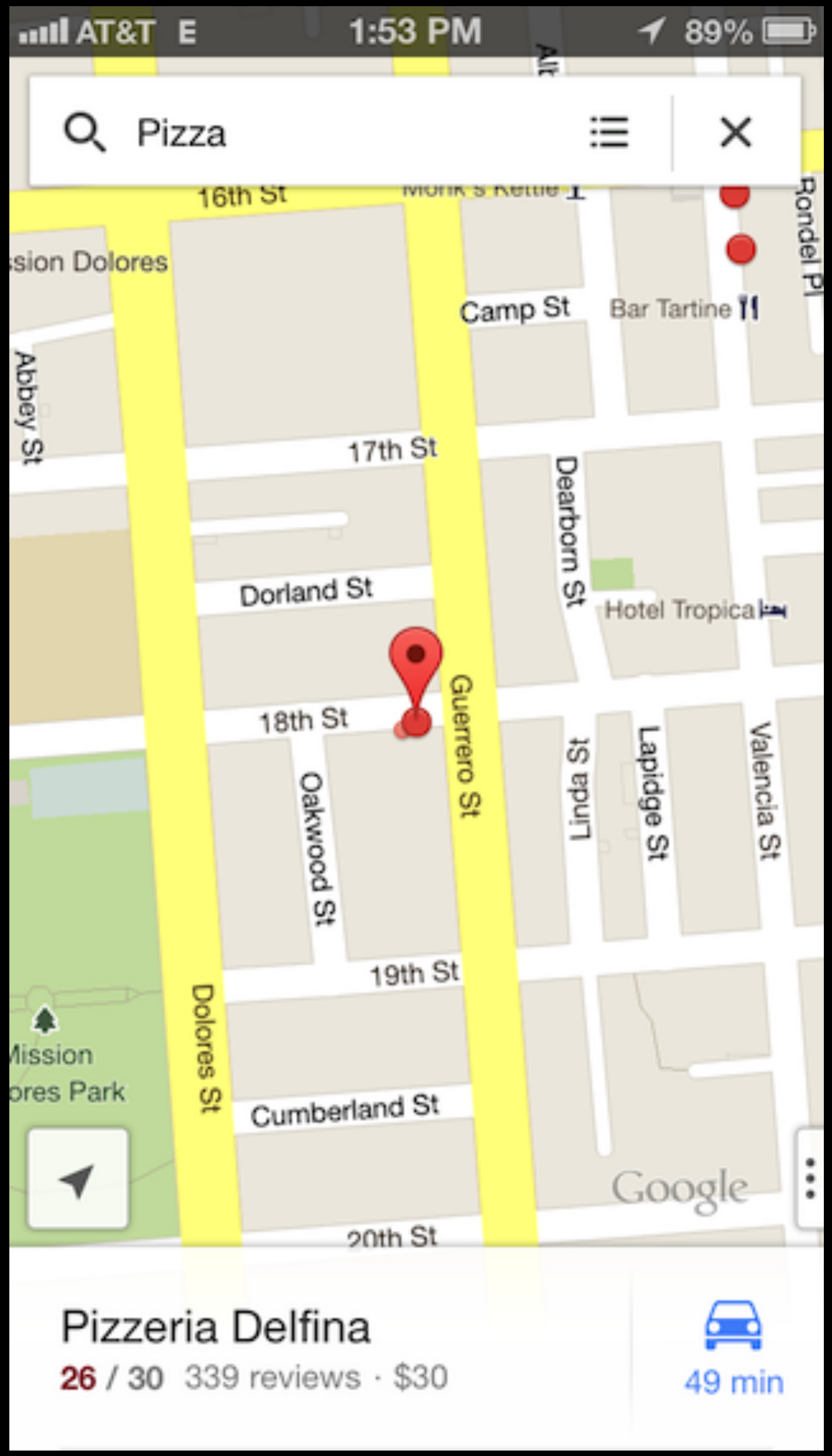
- "Gorilla arm" — humans are very bad at holding arms out for extended periods



Minority Report → bad for gorilla arm!

Considering contextual and physical limitations

- Users can be constrained by their abilities or context
  - Free hands?
  - Holding something?
  - Sitting/standing/walking?
- Example: Google Maps supports two finger pinch to zoom, also double tap + slide up/down



Intuitiveness

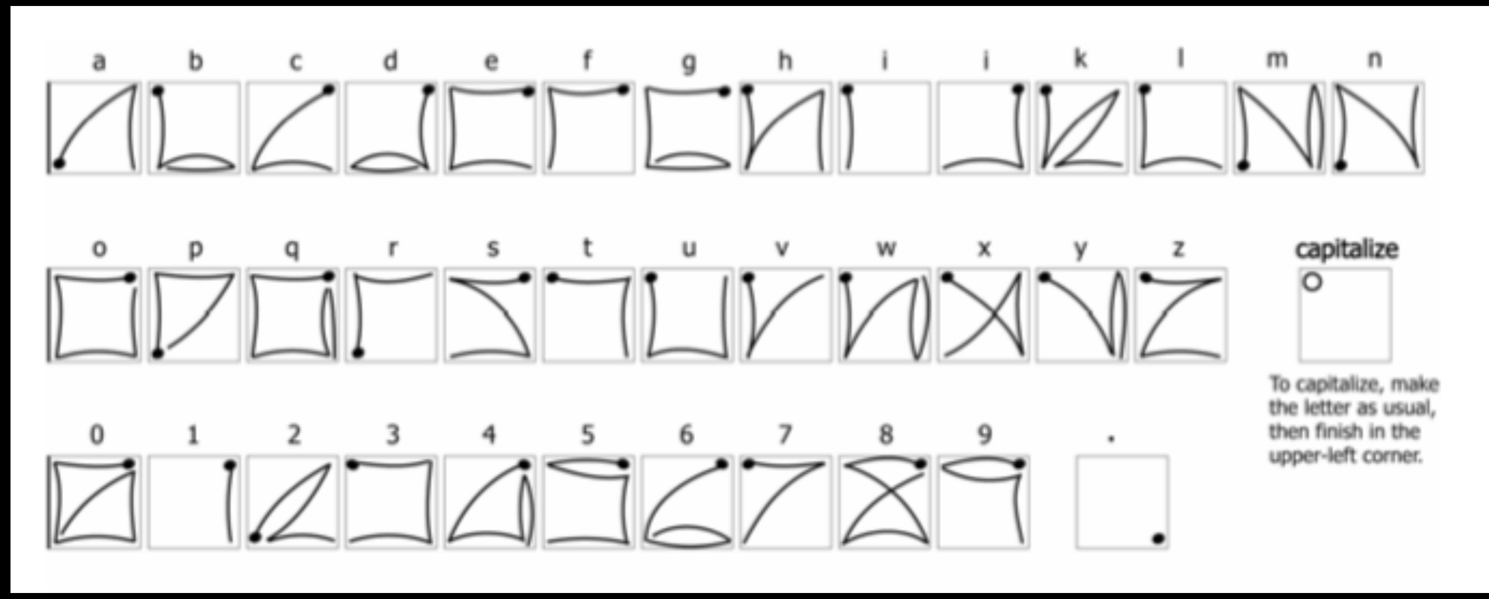
- How could we measure intuitiveness of a gesture?

What is intuitiveness anyway?

- Understandable
- Feedback
- Predictable outcome
- Can be explained concisely, “makes sense”
- Using what you know / no need for specialized knowledge
- Guessable

Testing intuitiveness

- **Learnability:** how long does it take someone to learn?
- **Memorability:** do users remember gestures after some time away?
- **Guessability:** tell user what the goal is, ask them to guess what the gesture will be



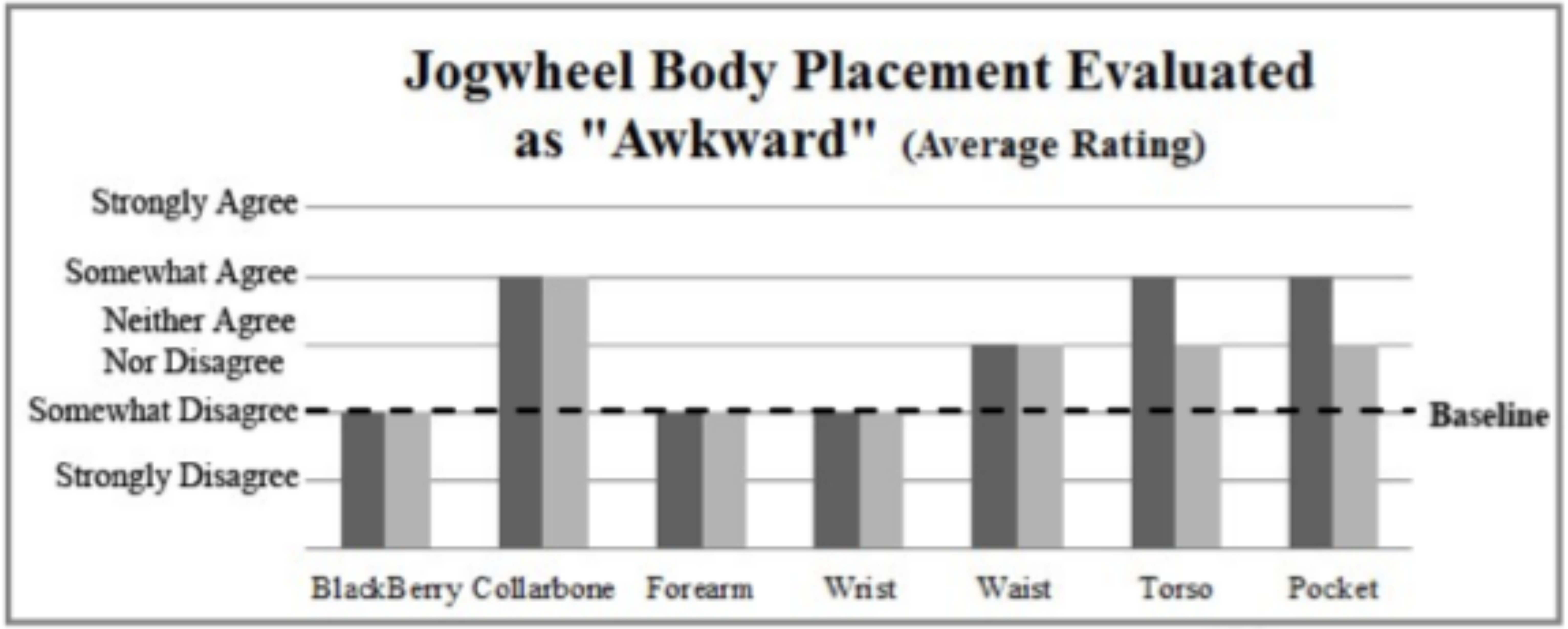
Wobbrock et al., [EdgeWrite]

Are obscure gestures useful?

- Not a great idea to hide dangerous features behind gestures
    - May activate by accident
    - Unclear what gesture does
- May be some social benefit in learning obscure gestures
    - “Why Snapchat’s Design is Deliberately Confusing”

Social acceptability

- Some gestures may be intuitive, memorable, highly recognizable, and <u>embarrassing</u>
- Must consider (and test) whether users are willing to perform gestures
- Consider type of movement, location on body, etc.
- May vary based on context and cultural factors



Profita et al., Don’t mind me touching my wrist

Designing gestures

Designing gestures

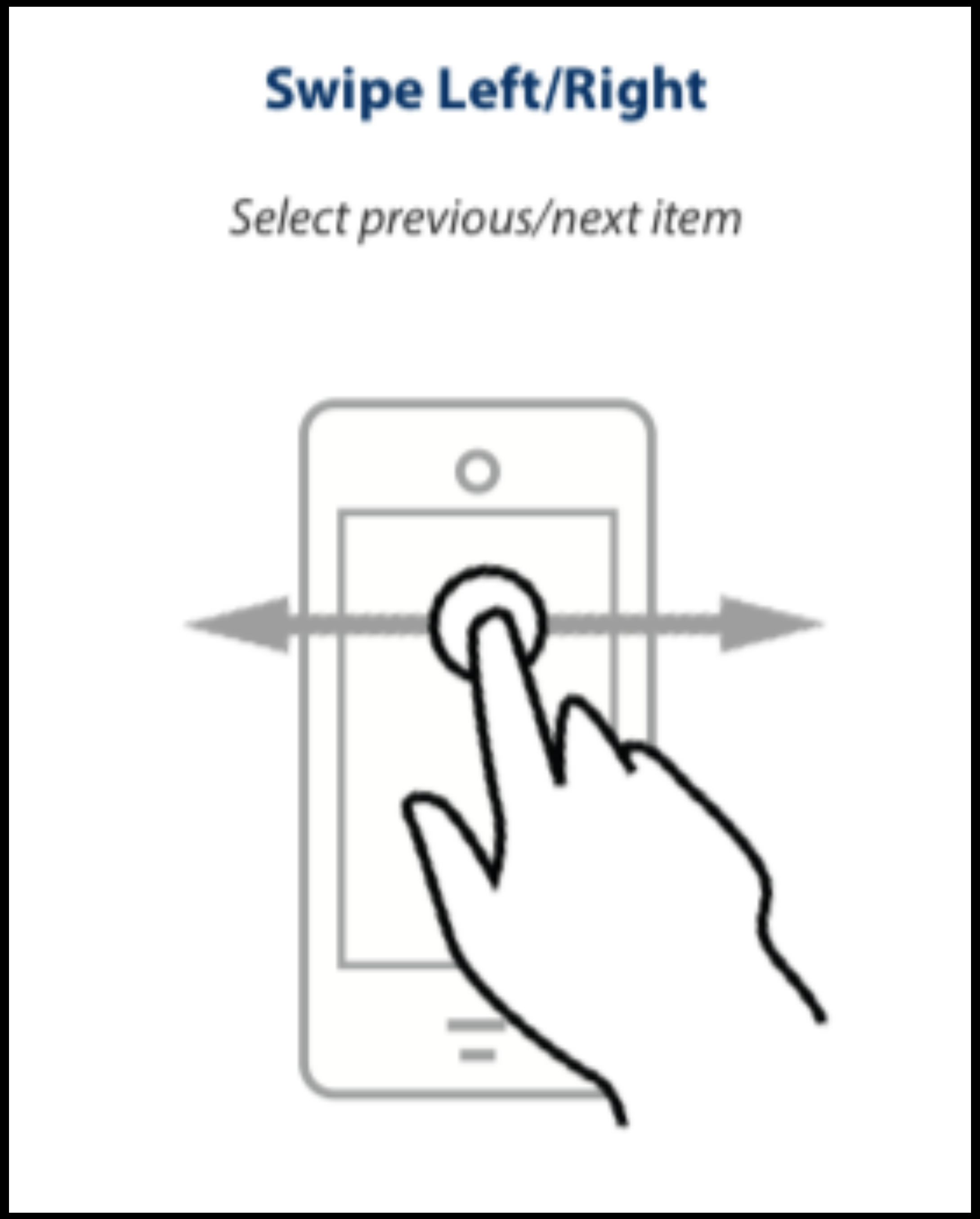
- Approaches for designing gestures
- Handling complexity
- Having a system of gestures
- Determining what’s important

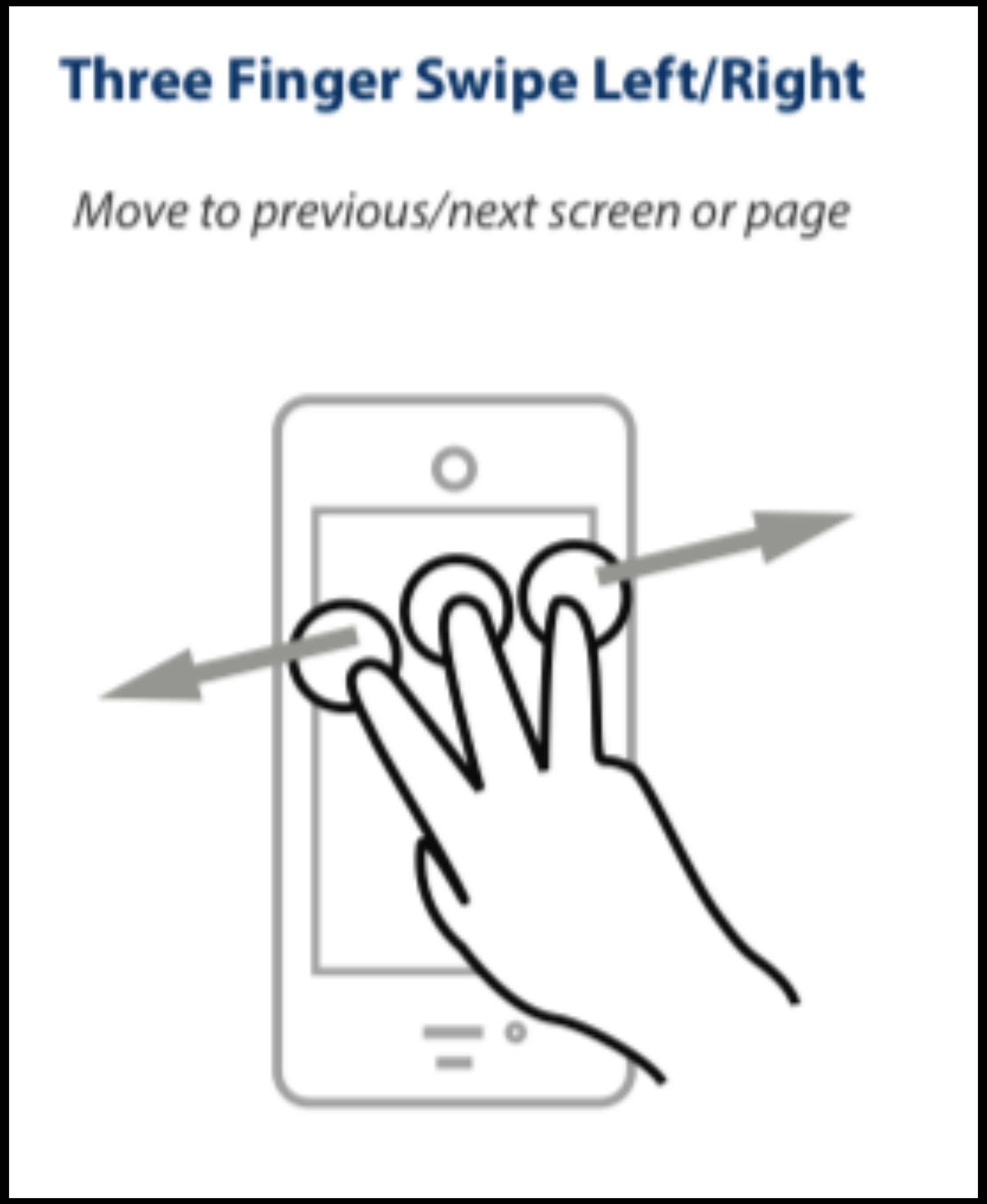
Approaches to designing gestures

- Developing conventions
- Using physical metaphors
- User-defined elicitation methods
- Dealing with complexity

Developing conventions

- This is an issue of **mappings**
- How do characteristics of gesture (direction, shape, number of fingers) map to effect?
- Users may have difficulty remembering complex mappings





Using metaphors

- Gestures can map to metaphors
- At a basic level, **direct manipulation** of items on screen
    - Drag left, item goes left
- Developing vocabulary of gestures can help in adoption

Figure 9: The horizontal hand physically blocks others from seeing the box of furniture properties below it.

Figure 8: Vertical hand sweeping. (a) Initial position. (b) When the hand makes contact with furniture, the pieces move with it. (c) Final position after sweeping.

Wu and Balakrishnan, [Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays]()

Gesture elicitation

(Wobbrock et al., 2009)

- Big idea: allowing users to pick their own gestures will lead to more intuitive gestures
  - But, users may pick gestures that conflict, or are difficult to memorize
- Solution: elicit gestures from a set of representative users, maximize agreement
- This method has become extremely popular and has been used for many types of gestures: mobile devices, body gestures, drone interactions…

Eliciting gestures

- Show participants **outcome** of action; allow them to demonstrate a gesture for that action

- Pick gestures with largest # of agreeing participants

- Can reverse it; show gestures to guess outcome

- Can't just pick top gestures, must deal with overloading, conventions

REFERENTS	Mean	SD	REFERENTS	Mean	SD
1. Move a little	1.00	0.00	15. Previous	3.00	0.00
2. Move a lot	1.00	0.00	16. Next	3.00	0.00
3. Select single	1.00	0.00	17. Insert	3.33	0.58
4. Rotate	1.33	0.58	18. Maximize	3.33	0.58
5. Shrink	1.33	0.58	19. Paste	3.33	1.15
6. Delete	1.33	0.58	20. Minimize	3.67	0.58
7. Enlarge	1.33	0.58	21. Cut	3.67	0.58
8. Pan	1.67	0.58	22. Accept	4.00	1.00
9. Close	2.00	0.00	23. Reject	4.00	1.00
10. Zoom in	2.00	0.00	24. Menu access	4.33	0.58
11. Zoom out	2.00	0.00	25. Help	4.33	0.58
12. Select group	2.33	0.58	26. Task switch	4.67	0.58
13. Open	2.33	0.58	27. Undo	5.00	0.00
14. Duplicate	2.67	1.53	**MEAN**	**2.70**	**0.47**

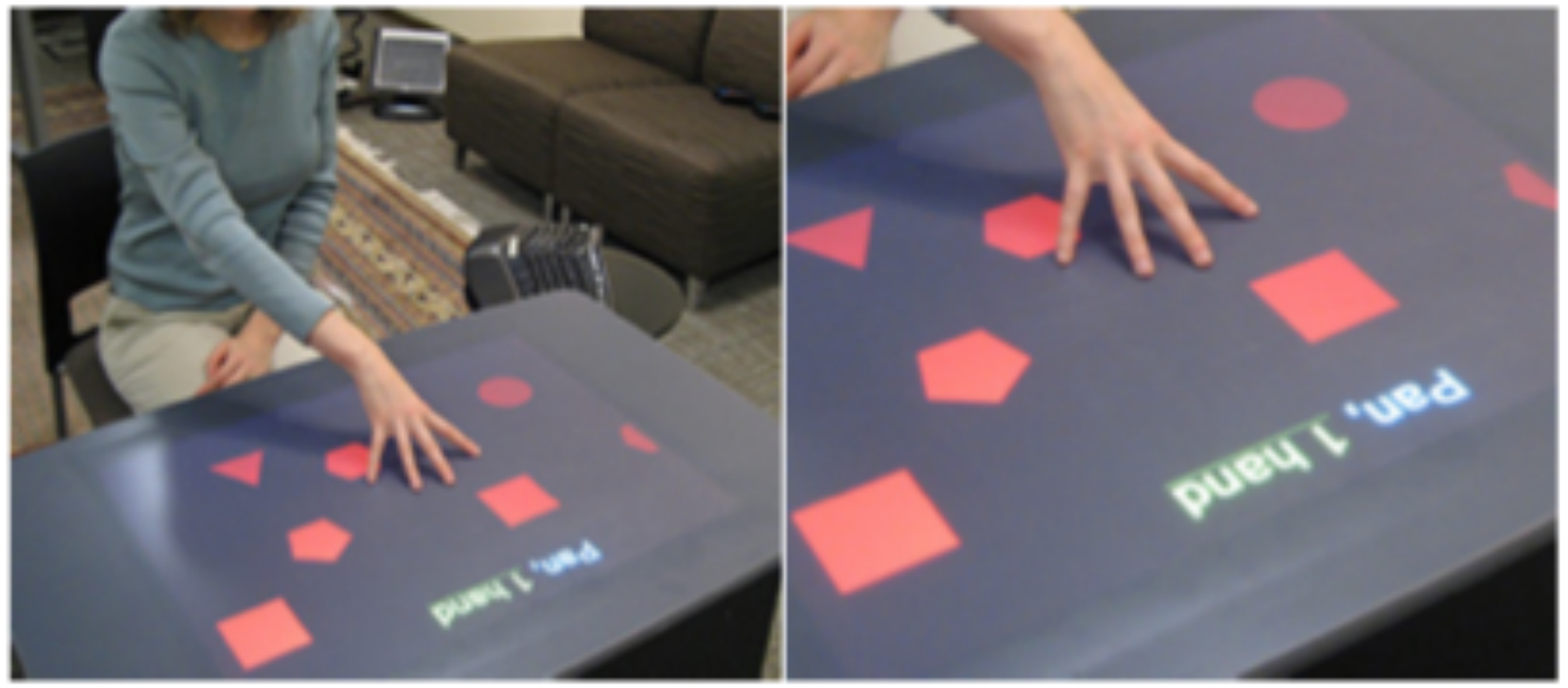


Figure 1. A user performing a gesture to pan a field of objects after being prompted by an animation demonstrating the panning effect.

Dealing with complexity

- Gestures have many variables
    - Example: for one-handed touch screen gestures, speed, size, number of figures, direction, location
- It’s important to be clear about which variables change how the gesture is interpreted, and which don’t
- Users may feel more strongly about certain variables
    - e.g., users do not feel strongly about the number of fingers in a gesture (Wobbrock et al., 2009)

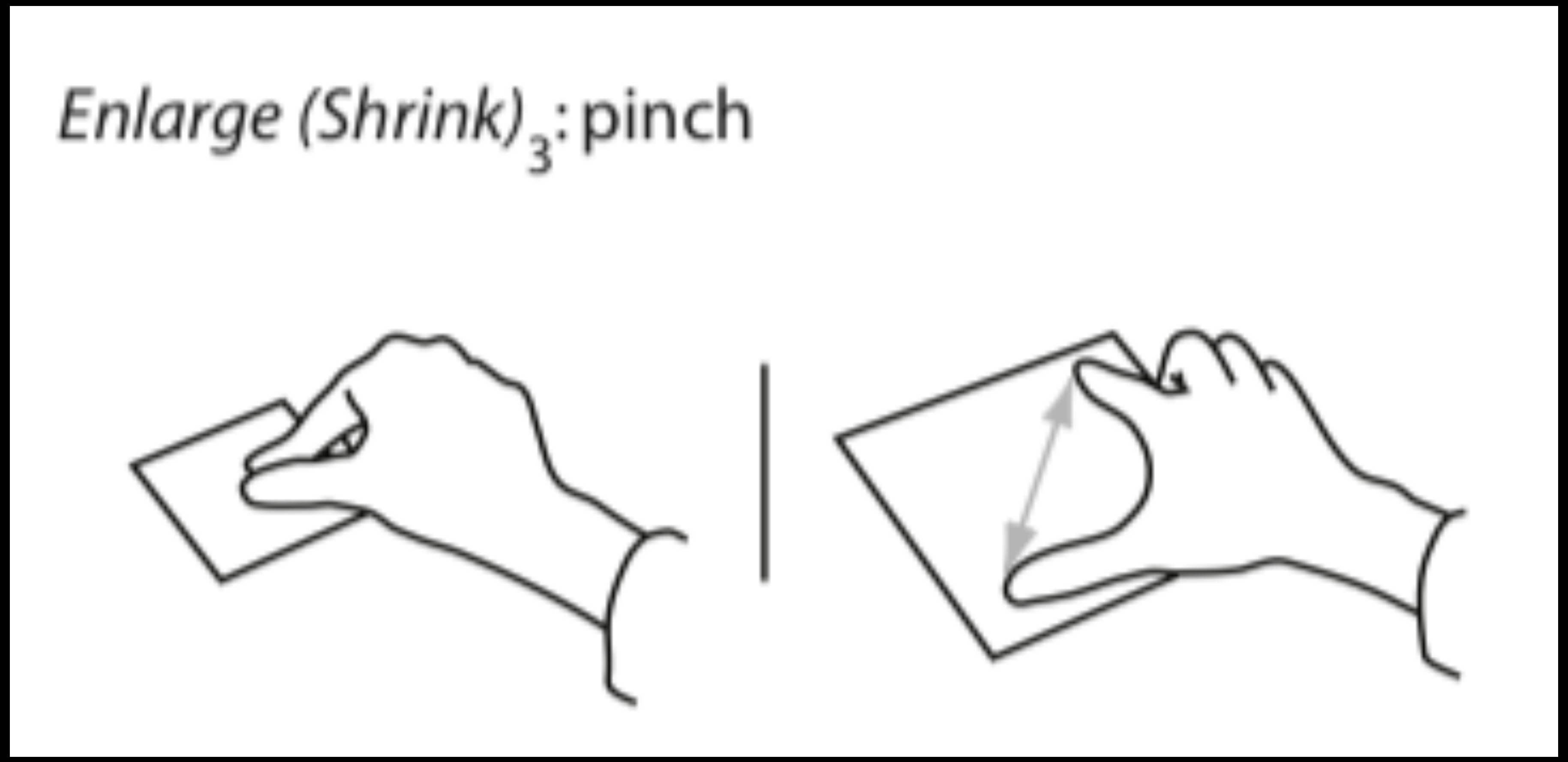
Teaching gestures

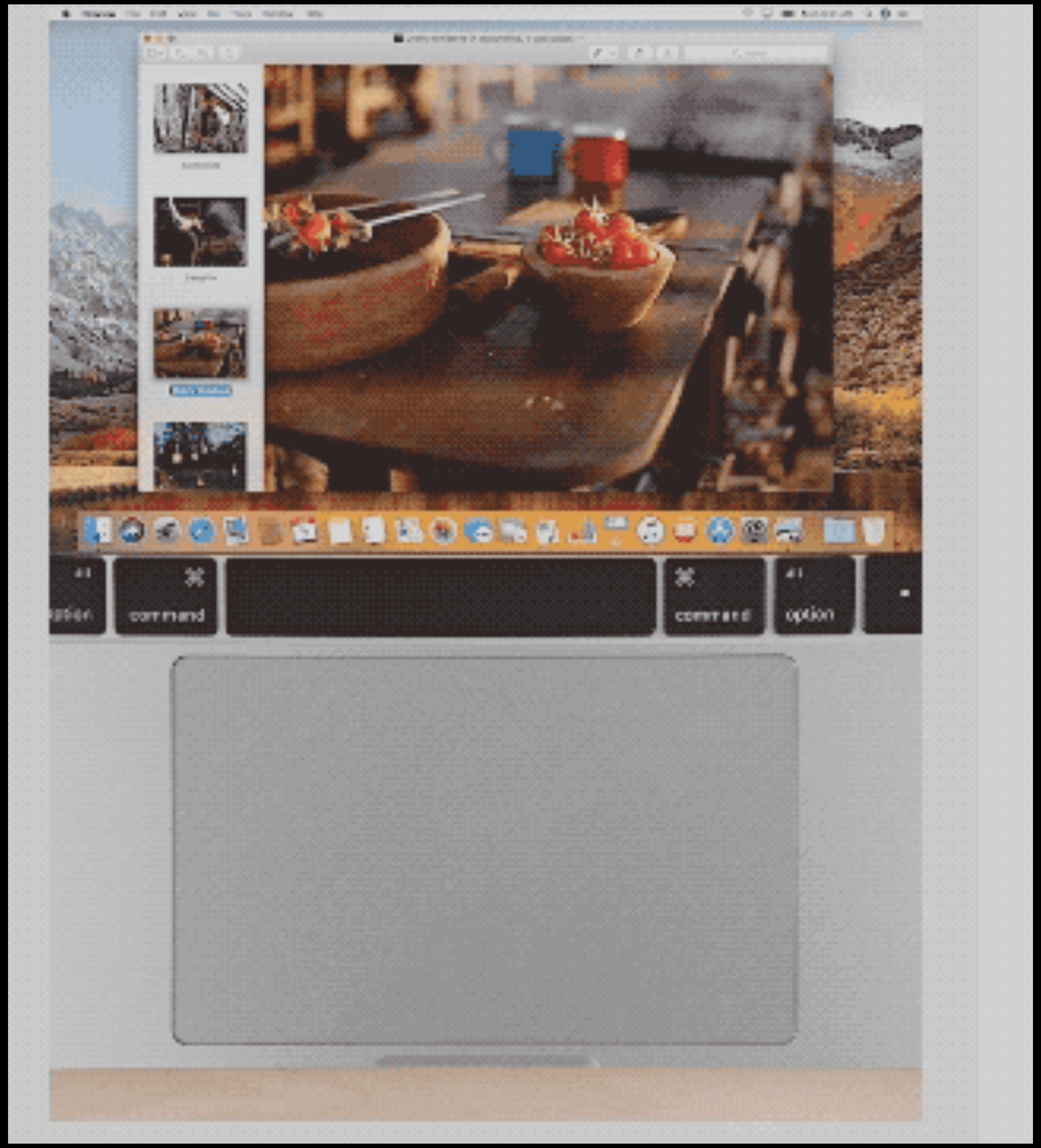
How to teach gestures effectively?

- Show, don’t tell
- Provide feedforward and feedback
- Hint at possible gestures

Show, don't tell

- "To zoom, pinch your fingers"
  vs.



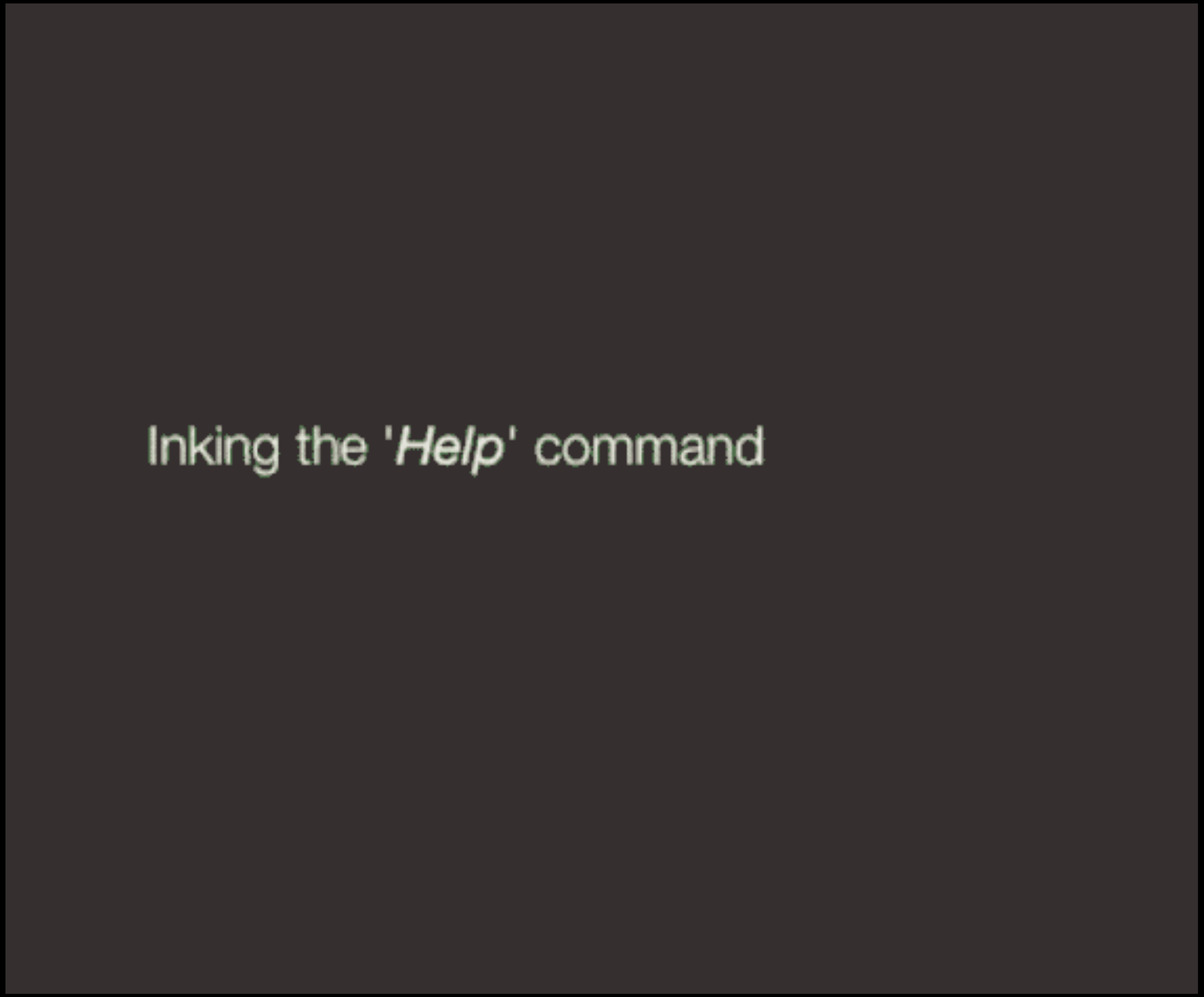


Feedforward and feedback

- Feedforward?

Feedforward and feedback

- Feedback: make it clear what I did
- Feedforward: show possible actions from here
- One (complex) example: Octopocus by Bau et al. (paper, video)



Hinting at possible gestures

- Show that there is more content offscreen
- Or show which objects can move
- A simple form of feedforward

Gesture miscellany

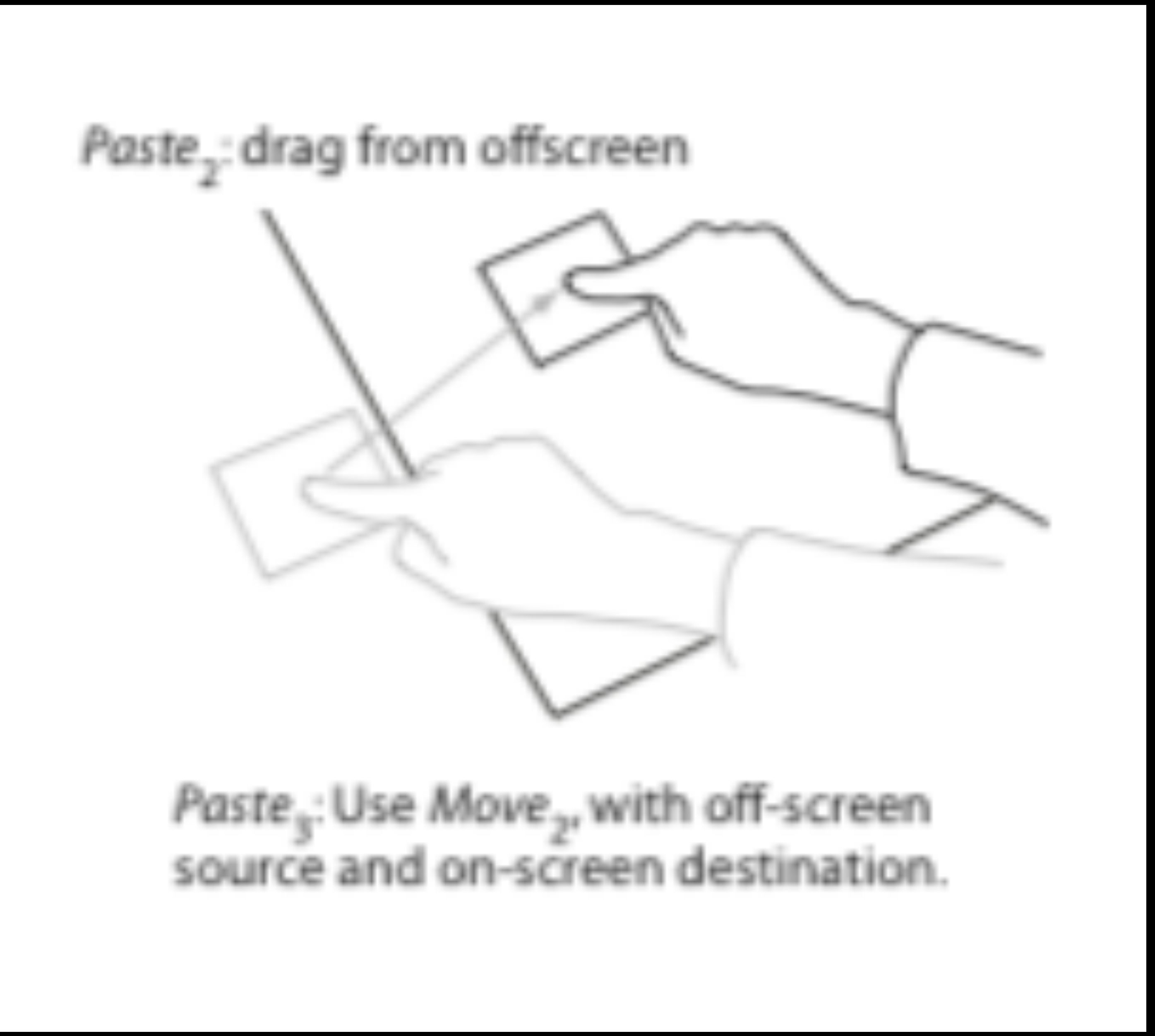
- Documenting gestures
- User tests with gestures
- Gesture accessibility
- Downsides of gestures

Documenting gestures

- Video may be ideal: shows all aspects of how to perform gesture correctly

- For figures, it's often easiest to take a photo and then trace it

- Use arrows to show motion



User testing gestures

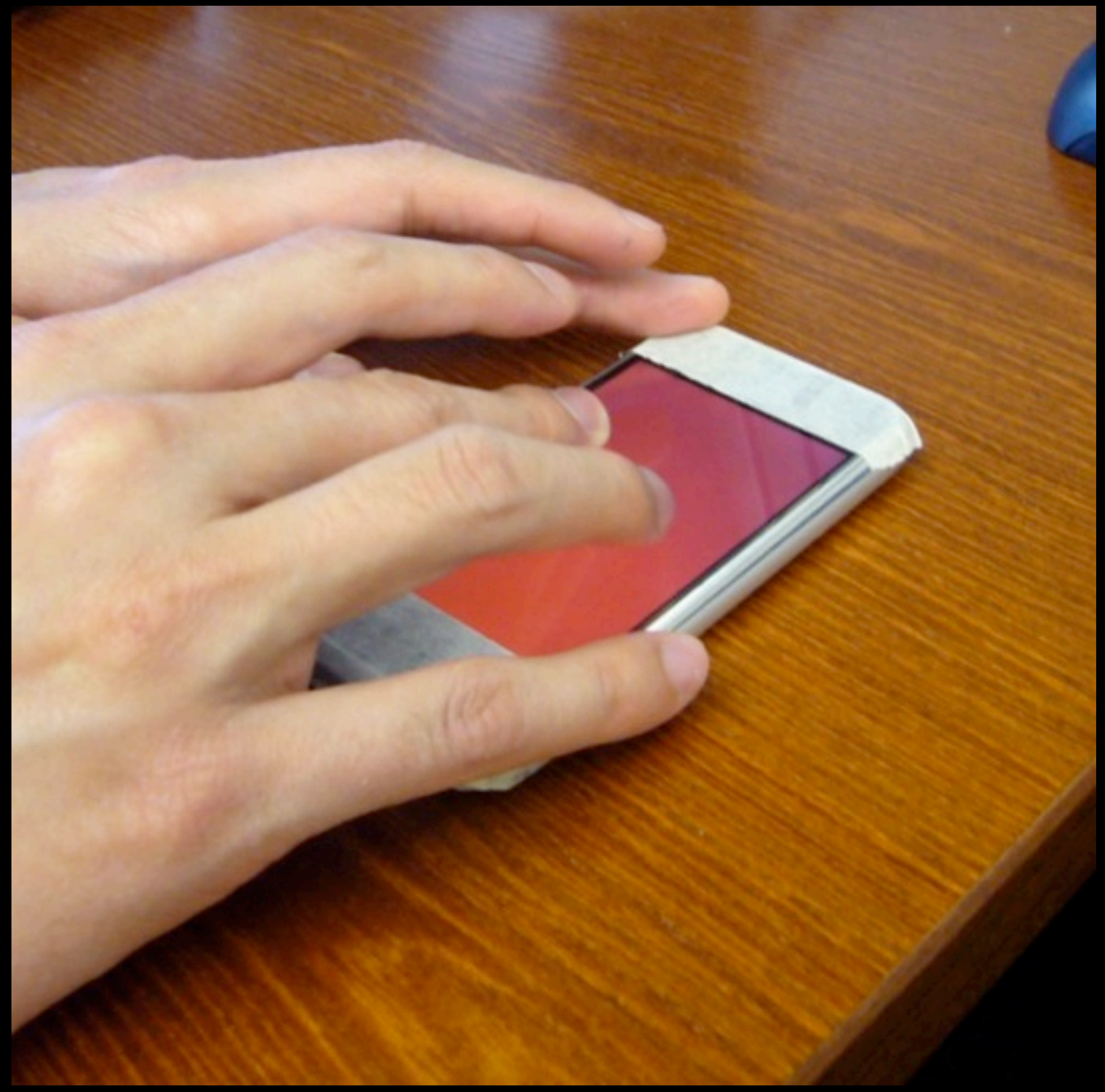
- Provide an introduction to the gesture set, then take away hints and documentation
    - Users will not have documentation in most cases
- Test recognizer for accuracy, errors, confusion
- Test gesture set usability by learnability, memorability, types of errors
    - Think aloud can help here

Gesture accessibility

- Many users may have difficulty performing certain gestures (due to reach, hand pose, tremor, ability to touch and lift-off)
- Many gestures rely on vision (to position on screen, to precisely control shape)

Eyes-Free Touch Screens

- How can we design touch screen user interfaces that can be operated without sight?



- Can we achieve this with off-the-shelf devices?

SK's publications:

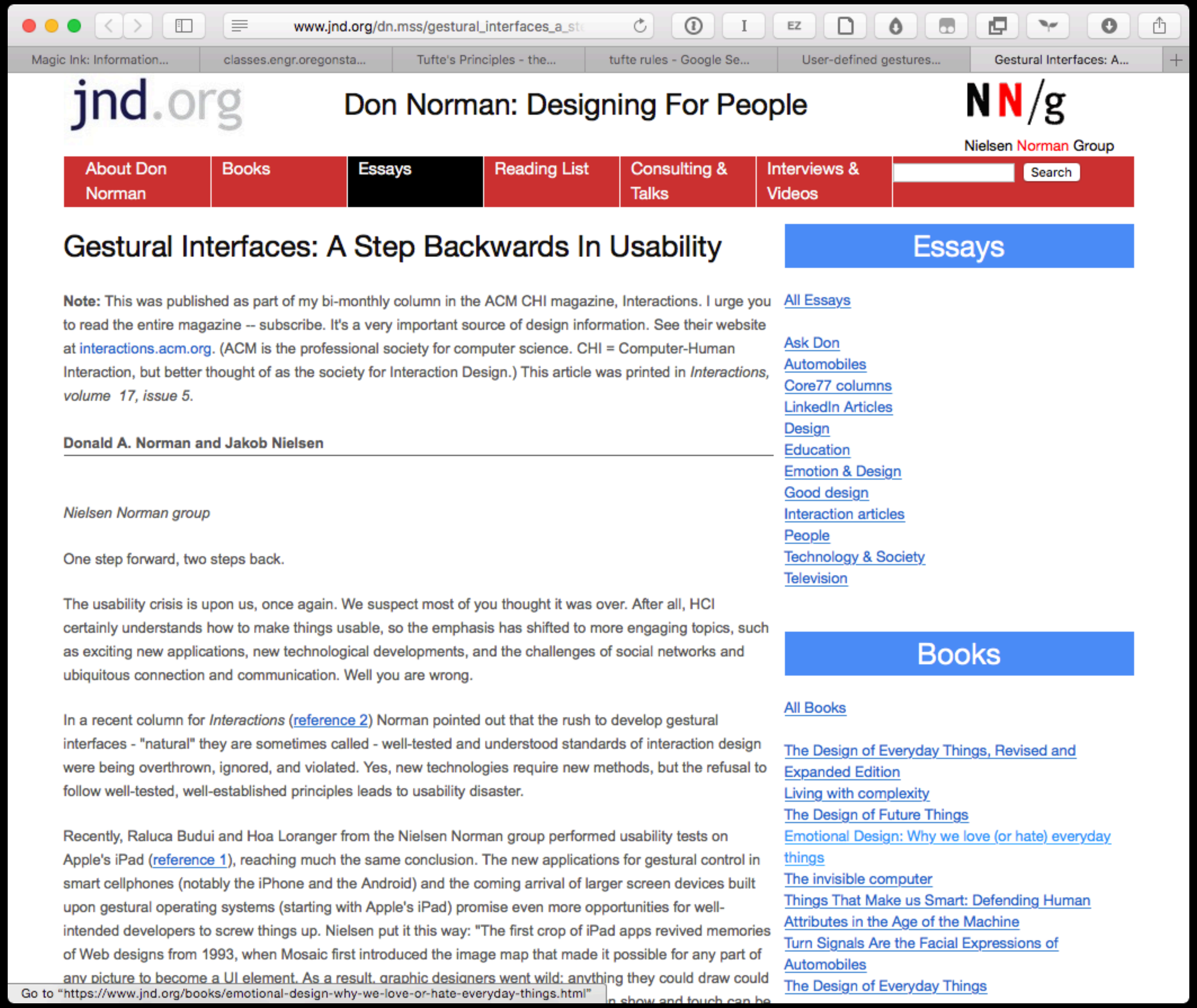
- **ASSETS 2008, 2009, 2014**
- **CHI 2011🏆, 2013, 2016**
- **DIS 2017**
- **TACCESS 2015**

Gesture accessibility: what to do

- Keep in mind that some users may be unable to perform some gestures, even standard gestures
- Provide alternatives for keyboard, mouse, etc.
- **For blind and vision impaired users:** Support system-wide accessibility systems (e.g. VoiceOver on iOS, Talkback on Android)

Downsides of gestures

- Very low discoverability
- Unclear what gesture requirements are
- No standard gesture set



www.jnd.org/dn.mss/gestural_interfaces_a_st

jnd.org Don Norman: Designing For People NN/g Nielsen Norman Group

About Don Norman | Books | Essays | Reading List | Consulting & Talks | Interviews & Videos | Search

Gestural Interfaces: A Step Backwards In Usability

Note: This was published as part of my bi-monthly column in the ACM CHI magazine, Interactions. I urge you to read the entire magazine -- subscribe. It's a very important source of design information. See their website at interactions.acm.org. (ACM is the professional society for computer science. CHI = Computer-Human Interaction, but better thought of as the society for Interaction Design.) This article was printed in *Interactions, volume 17, issue 5.*

Donald A. Norman and Jakob Nielsen

Nielsen Norman group

One step forward, two steps back.

The usability crisis is upon us, once again. We suspect most of you thought it was over. After all, HCI certainly understands how to make things usable, so the emphasis has shifted to more engaging topics, such as exciting new applications, new technological developments, and the challenges of social networks and ubiquitous connection and communication. Well you are wrong.

In a recent column for *Interactions* (reference 2) Norman pointed out that the rush to develop gestural interfaces - "natural" they are sometimes called - well-tested and understood standards of interaction design were being overthrown, ignored, and violated. Yes, new technologies require new methods, but the refusal to follow well-tested, well-established principles leads to usability disaster.

Recently, Raluca Budui and Hoa Loranger from the Nielsen Norman group performed usability tests on Apple's iPad (reference 1), reaching much the same conclusion. The new applications for gestural control in smart cellphones (notably the iPhone and the Android) and the coming arrival of larger screen devices built upon gestural operating systems (starting with Apple's iPad) promise even more opportunities for well-intended developers to screw things up. Nielsen put it this way: "The first crop of iPad apps revived memories of Web designs from 1993, when Mosaic first introduced the image map that made it possible for any part of any picture to become a UI element. As a result, graphic designers went wild: anything they could draw could

Essays

All Essays

Ask Don
Automobiles
Core77 columns
LinkedIn Articles
Design
Education
Emotion & Design
Good design
Interaction articles
People
Technology & Society
Television

Books

All Books

The Design of Everyday Things, Revised and Expanded Edition
Living with complexity
The Design of Future Things
Emotional Design: Why we love (or hate) everyday things
The invisible computer
Things That Make us Smart: Defending Human Attributes in the Age of the Machine
Turn Signals Are the Facial Expressions of Automobiles
The Design of Everyday Things

Norman and Nielsen, [Gestural Interfaces: A Step Backwards In Usability]

Discoverability

- Replacing explicit UI with gestures can negatively affect usability
- How can we address this problem?

Discoverability solutions

- Show help on screen
- Use gestures as an accelerator (with buttons as a backup)
- Hint at gestures in UI design