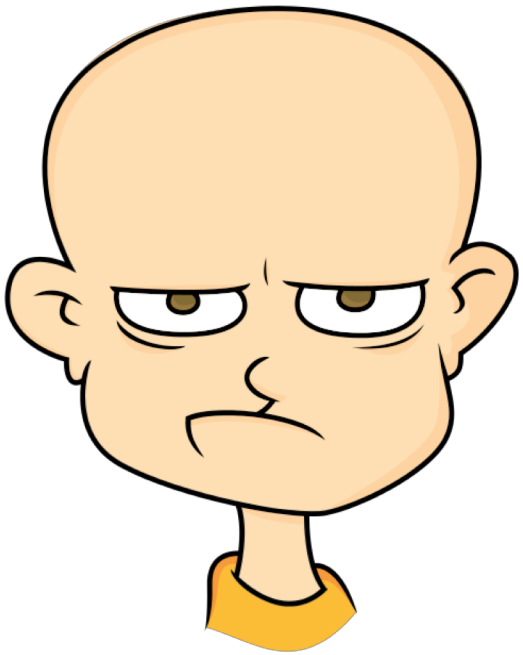
Touchless Interfaces

Working with sensor data
Gesture-based systems
Speech and Conversational Agents

Touchless Interfaces

Today, we have a wide array of sensors available.



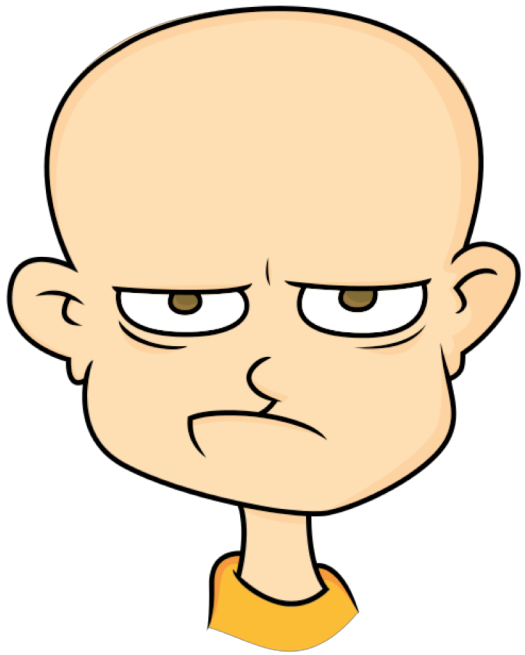
OpenClipArt.org

"Frustration Meter"

- microphone to detect users muttering and yelling at the computer
- pressure sensor to detect whether user is typing hard or squeezing the mouse
- webcam to detect frowning
- chair sensors to detect user agitation

Challenges with Sensors

The frustration meter is not easy to build



OpenClipArt.org

- need to learn the mapping between the output of the sensors and the presumed state of frustration in the user
- sensor output is noisy
- may need a combination of sensors
- our preconceived notions of frustration may not be what the sensors observed
- the way users expresses frustration is task and context dependent
- need to balance the cost of our system making mistakes and the benefit that the system provides

What might we want to sense?

Occupancy & Motion: senses motion or a person's presence

- infrared motion detectors, pressure mats, computer vision w. cameras

Range: calculate distance to a given object

- stereo computer vision system (triangulation), infrared cameras (time-of-flight), Polaroid ultrasound (time-delay)

Position: geo-location of the device

- GPS, motion capture system (with corresponding body model)

Movement and Orientation: sense spatial motion (e.g., translation, rotation)

- inertial sensors like gyroscopes, accelerometers

Gaze: determining where a person is looking (e.g. determining target on-screen)

- camera, eye-tracking systems

Speech: detect and process voice input or commands (e.g. Siri)

- array microphone combines audio from multiple sources for filtering

Brain-wave activity: low-fidelity input, potential for accessibility

- EEG (requires extensive training).

What can we do with this data?

Support In-Air Gestures: hand pose, spatial trajectory of hands or stylus

- tracking and movement devices (e.g. smartphone), hand-tracking systems (cameras, using markers etc.)

Identify objects or people: object recognition, person recognition (face recognition)

- lots of options: on-body devices/wearables (RFID, BT), fingerprints, retina-scanning, heart-rate monitoring, EEG

Determine context: figure out the context of the user (e.g. in-car)

- environmental sensor that detect air temperature, lighting quality, air pressure; cameras that detect location or environment

Determine Affect: detect an emotion or subjective feeling in a person

- respiration, heart-rate monitors, blood-pressure sensors, EMG (muscle contractions)

Design Considerations

Computation cost

- Computer vision algorithms are still computationally intensive, and some analysis cannot easily be performed in real-time (e.g. head-tracking on a phone)
- Approaches may require aggregating data from multiple sensors.
- High-volume of continuous data!
- Sensor data is really, really noisy and requires work to cleanup

Traditional or non-traditional interfaces

- How do we integrate these sensors and this type of data into common, real-world applications?
- Sensors: suggest that apps are more data-driven than task-driven (recent approaches of speech and facial recognition)

Implementing Signal to Command Systems

Step 1: Pre-processing

- compress
- smooth
- thresholding (discretize continuous quantity)
- downsample
- handle latency

Step 2: Feature Selection

- e.g., face detection
  - distances between eye, nose and mouth
  - eye-blinking patterns

Step 3: Classification

- determining which class a given observation belongs to
- e.g., recognize which gesture is performed

Example: “Computer, Turn on the Lights”

(Brumitt et al., 2000) Users try controlling the light using these options:

- a traditional GUI list box
- graphical touch screen display depicting a plan view of the room with lights
- two speech only-based systems
- a speech and gesture-based system.

Users prefer to use speech to control the lights, but the vocabulary used to indicate which light to control is highly unpredictable.

Example: “Computer, Turn on the Lights”

Insight: Users look at the light that they are controlling while speaking.

XWand system is a hand-held device that can be used to select objets in the room by pointing, and a speech recognition system for simple command and control grammar. (turn on, turn off, roll to adjust volume)

http://www.youtube.com/watch?v=bf3mQRmzA4k#t=146

Key Challenge: Balancing Usage Patterns

Wilson, 2007

Explicit Interaction

- user takes action and expects a timely responsive from the system
- e.g. Siri, Kinect-driven games

Implicit Interaction

- based on user's existing pattern of behaviour
  - e.g., frustration meter
    - e.g., think "personalized google search"
    - e.g., a smart home that observe an inhabitants daily pattern of coming and going to determine an optimal thermostat schedule

We can use sensor data to drive interaction in both scenarios.

Key Challenge: Errors is a Balancing Act

Wilson, 2007

False Positive Error

- user does not intend to perform a gesture but the system recognizes one
- making the system seem erratic or overly sensitive.

False Negative Error

- user believes he/she has performed a gesture but the system does not recognize it
- making system seem unresponsive, like the user is doing something wrong

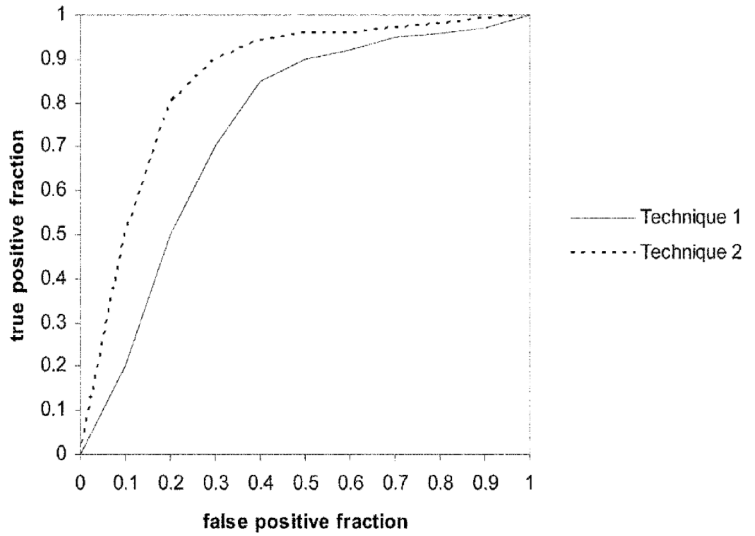


FIGURE 10.1. ROC curves illustrate the trade-off between the rate of true positives and false positives, and can be useful in comparing recognition techniques. Here we see that for a given tolerable rate of false positives, Technique 2 yields better recognition performance than Technique 1.

Key Challenge: Errors is a Balancing Act

Users need to feel a sense of control

- may feel unable to correct a mistake made by the system, or unable to exert more explicit control in an exception (e.g., a party)

Users may be very intolerant of errors

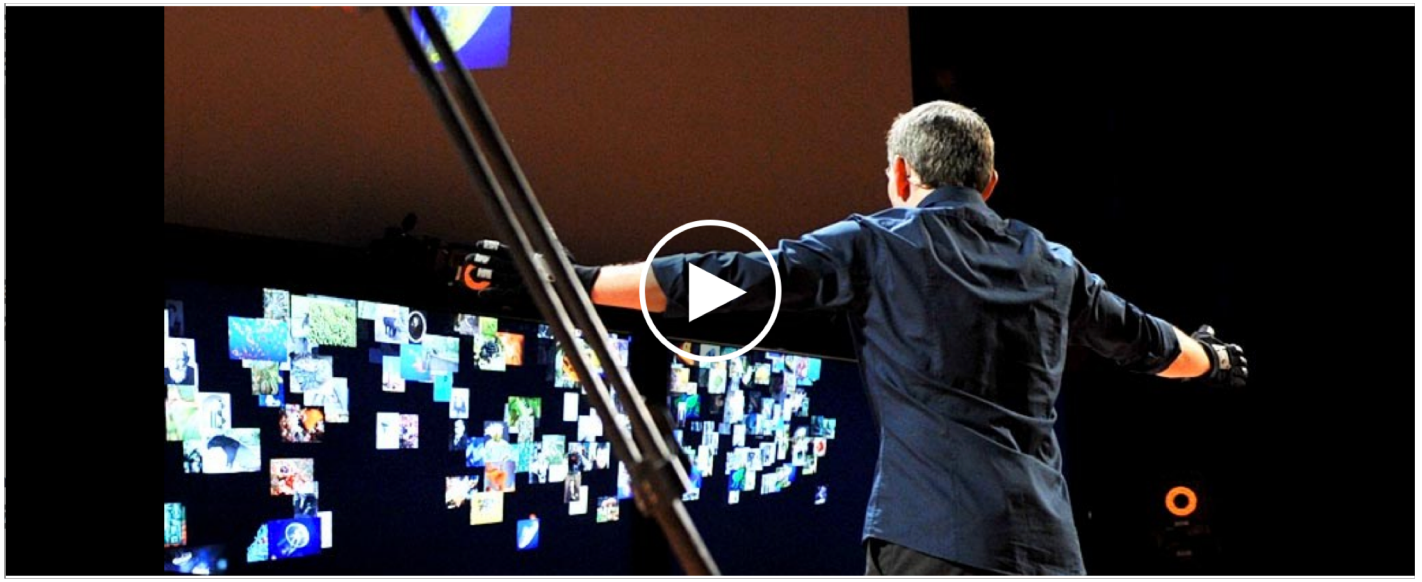
- in speech recognition, only users that are unable to use a regular keyboard may accept a dictation system that fails 3 times out of 100 words

Strategies

- graceful degradation, i.e., return a results similar to desired results
- avoid false positive by seeking deliberate confirmation from users
- give control: allow users to query the system for why it took a given action, fix errors, and revert to manual mode.

Example 1: In-Air Gestures

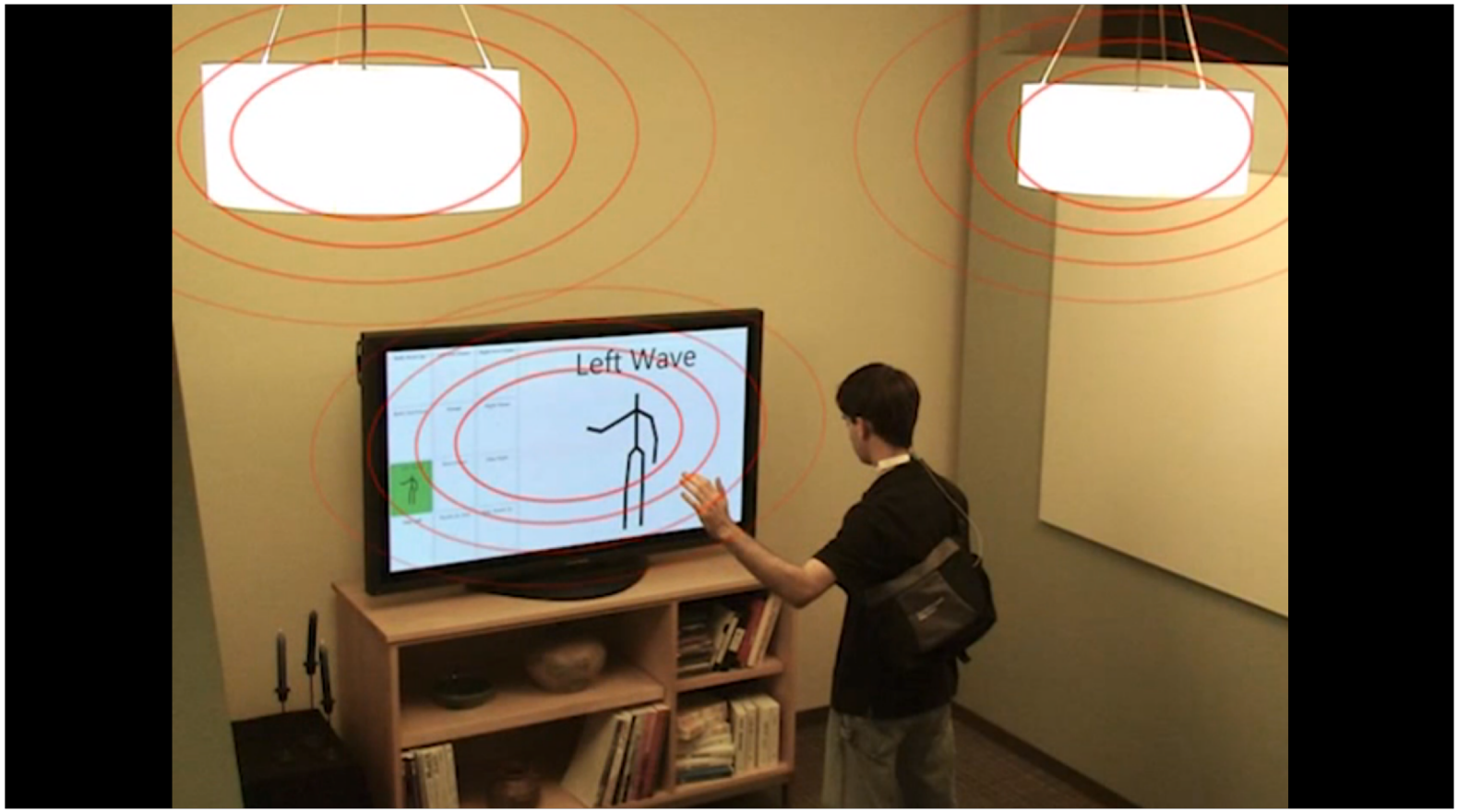
Example: Pointing to the future of UI



http://www.ted.com/talks/john_underkoffler_drive_3d_data_with_a_gesture (5:30-9:15)

https://vimeo.com/49216050

Example: Humantenna



http://www.youtube.com/watch?v=em-nvzxzC68
http://www.youtube.com/watch?v=7lRnm2oFGdc

Trouble with In-Air Gestures

"My fellow Americans, I'm pleased to tell you I just signed legislation which outlaws Russia forever. The bombing begins in five minutes."

— Ronald Reagan

In-Air Gestures suffer from the "Live Mic" problem.

https://en.wikipedia.org/wiki/We_begin_bombing_in_five_minutes

Mouse -> Touch-> Touchless

Mouse is a three-state input device.

Touch Interface is a two-state input device.

Touchless Interface is a one-state input device.

- The challenge is that in-air gesture system is always on, and there is no mechanism that will differentiate gestures from non-gestures.

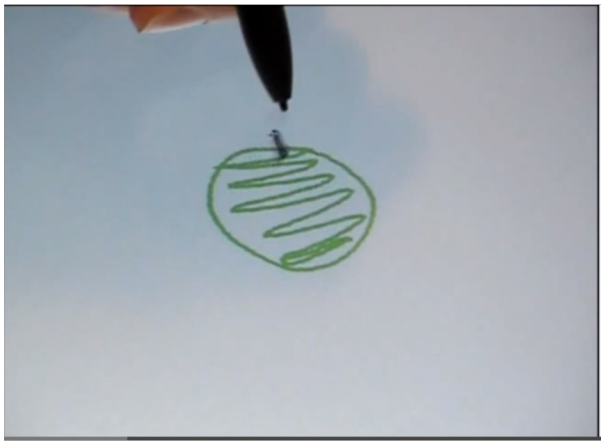
Consider a user who needs to sneeze, scratch his head, stretch, gesture to another person in the room, what would this mean for the three input devices?

Solution: Reserved Actions

Wigdor, 2011

Take a particular set of gestural actions and reserve them so that those actions are always used for navigation or commands.

pigtail gesture



http://www.youtube.com/watch?v=WPbiPn1b1zQ

hover widget

Question: for hover widget, which type error is more common (false negative or false positive) and why?

- user might accidentally lift their pen and move it
- user might exit the zone without realizing it *

Solution: Reserving a Gesture

A **delimiter** is an indicator that you want to start or stop the recognition of a gesture.

Example: to engage, user pushes past an invisible plane in front

Advantages?
- provides an a priori indicator to the recognizer that the action to follow is intended to be treated as a gesture or not.
- enables the complete gestural space to be used

Disadvantages?
- where should this invisible plane be? This may be different for different users, or different for the same user over time

Solution: Reserving a Gesture

Alternative: to engage, user pinches finger and thumb together

Advantages:

- less likely to be subject to false positive errors (Why?)
  - the action is unlikely to occur outside the context of the app
- less likely to be subject to false negative errors (Why?)
  - the action is easy to recognize and differentiate from other actions

Disadvantages:

- cannot use pinching in your gestural set

Solution: Multi-Modal Input

Wigdor, 2011

iPhone is a touch input device, so … Why does have a button?

The key problem is that users need to be able to exit their application and return to the home screen in a reliable way.

What's the alternative?

- a reserved action for exit

The multi-modal solution enable touch input to be sent to the application, while hardware buttons control universal parameters (e.g., volume) and navigation (e.g., go home, use Siri).

Why can we **now** design without buttons? e.g. iPhone X

Solution: Multi-Modal Input

Wigdor, 2011

Advantage over Reserved Action or Clutch:

- does not reduce the vocabulary of the primary modality

Some other examples

- CTRL-drag becomes copy instead of move
- speech + gesture (e.g., the “put that there” system)

Put That There (MIT, 1980)
http://www.youtube.com/watch?v=-bFBr11Vq2s

Feedback is Useful

Wilson, 2007

Provide feedback on all recognition results and the nature of failure so that users can modify the behaviour to meet the system's expectation.

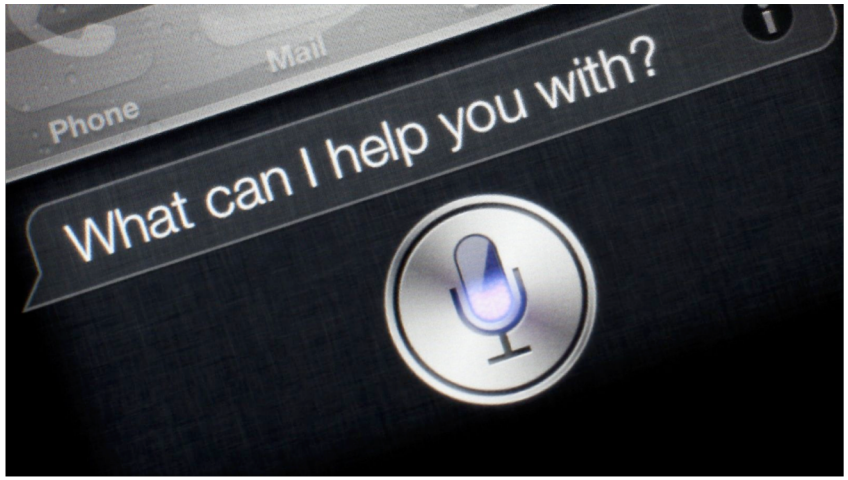
EyeToy

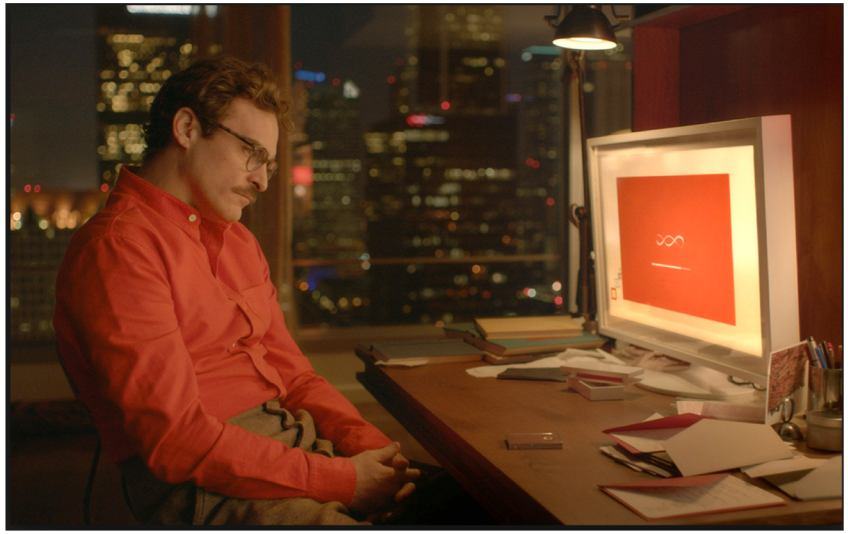
- image of player on screen
- by watching themselves on screen, players are able interact with the onscreen elements without relying on the sophisticated (but more failure prone and computationally intensive) hand-tracking algorithms
- this feedback also keeps player staying in camera's field of view

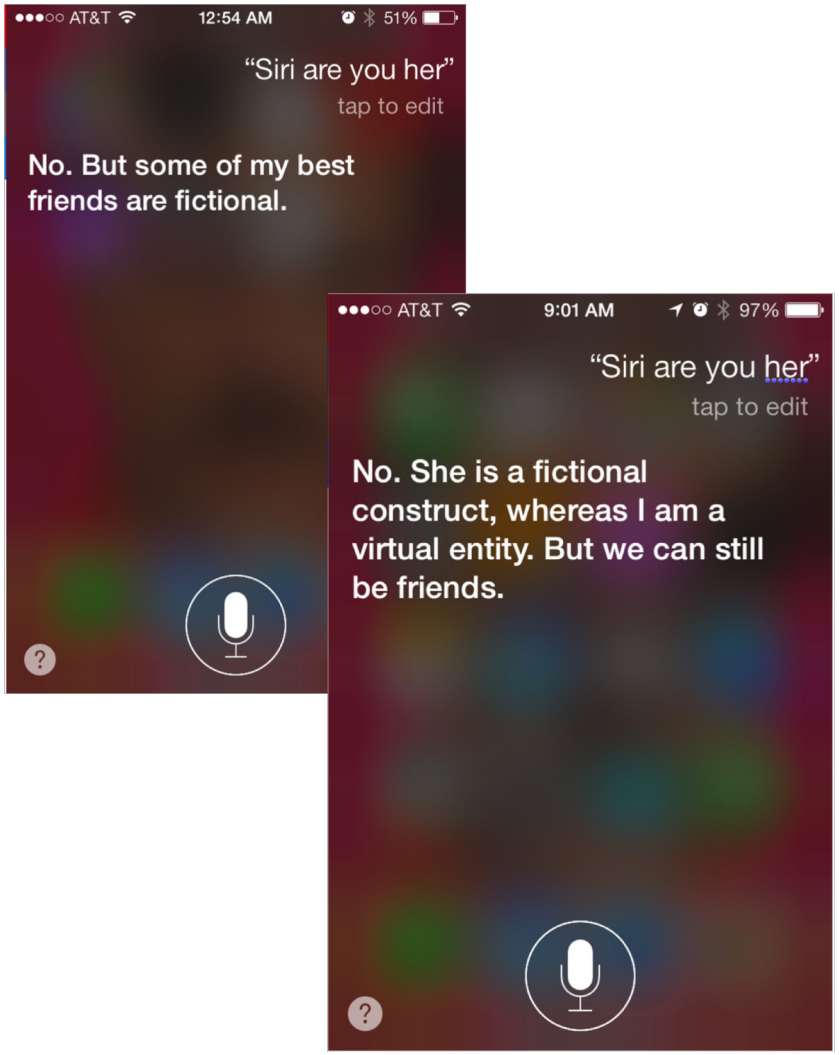
http://blog.us.playstation.com/2010/11/03/eyetoy-innovation-and-beyond/

Example 2: Speech Interfaces

Speech Interfaces (SUI)







SpeechActs - early example

Yankelovich, 1995

SpeechActs provides interfaces to a number of applications, including email, calendar, weather and stock quotes.

https://www.youtube.com/watch?v=OzNRsGaXyUA

To make interaction feel conversational, the system avoids explicitly prompting users for input whenever possible (e.g., use a prompt tone)

Found that recognition rates were a poor indicator of satisfaction

> “Users bring many and varying expectations to a conversation, and their satisfaction will depend on how well the system fulfills those expectations”

GUI to SUI: Challenges

Yankelovich, 1995

Discourse segment pop-up / navigation / context
- complete a sub-dialog and reveal context
- e.g., after replying to a message, return to reading messages
- how do we simulate “closing a pop-up window”?

Users want to use their “human” language
- Manager: Next Monday — Can you get into John’s calendar?
- Assistant: Gosh, I don’t think I can get into his calendar.
- no mention of “browse”, “user ID”; use of “relative date”

Dialog Boxes
- when to ask for confirmation?
- what if the user did not say “yes” or “no”?

GUI to SUI: Recognition Errors

Yankelovich, 1995

Recognition errors are frequent (e.g., user speaks before system is ready, background noise, words spoken by passerby, out-of-vocabulary utterances)

- Users often say something once and have it recognized, then say it again and have it mis-recognized
- Lack of predictability means that users cannot have a useful conceptual model of how the system works

Types of Errors

- Rejection error: system has no hypothesis about what user has said, so must reject ambiguous input
- Substitution error: mistaking utterances (“Seventh” for “send”)
- Insertion error: recognize noise as a legal utterance

GUI to SUI

Yankelovich, 1995

Rejection error

- "brick wall effect": keep saying "I don't understand"
- progressive assistance
  - What did you say?
  - Sorry?
  - Sorry. Please rephrase.
  - I don't understand. Speak clearly, but don't overemphasize.
  - Still no luck. Wait for the prompt tone before speaking.

Substitution error

- e.g., "Kuai" interpreted as "Good-bye" and hangup
- verification should commensurate with cost of action

Other Challenges

Yankelovich, 1995

Lack of Visual Feedback

- long pauses in conversations are perceived as embarrassing, so users feel a need respond quickly
- lack of think time can lead to false starts or “ums” and “ahs”
- less information is transmitted to users at a time

Speech is easy to produce, hard to absorb

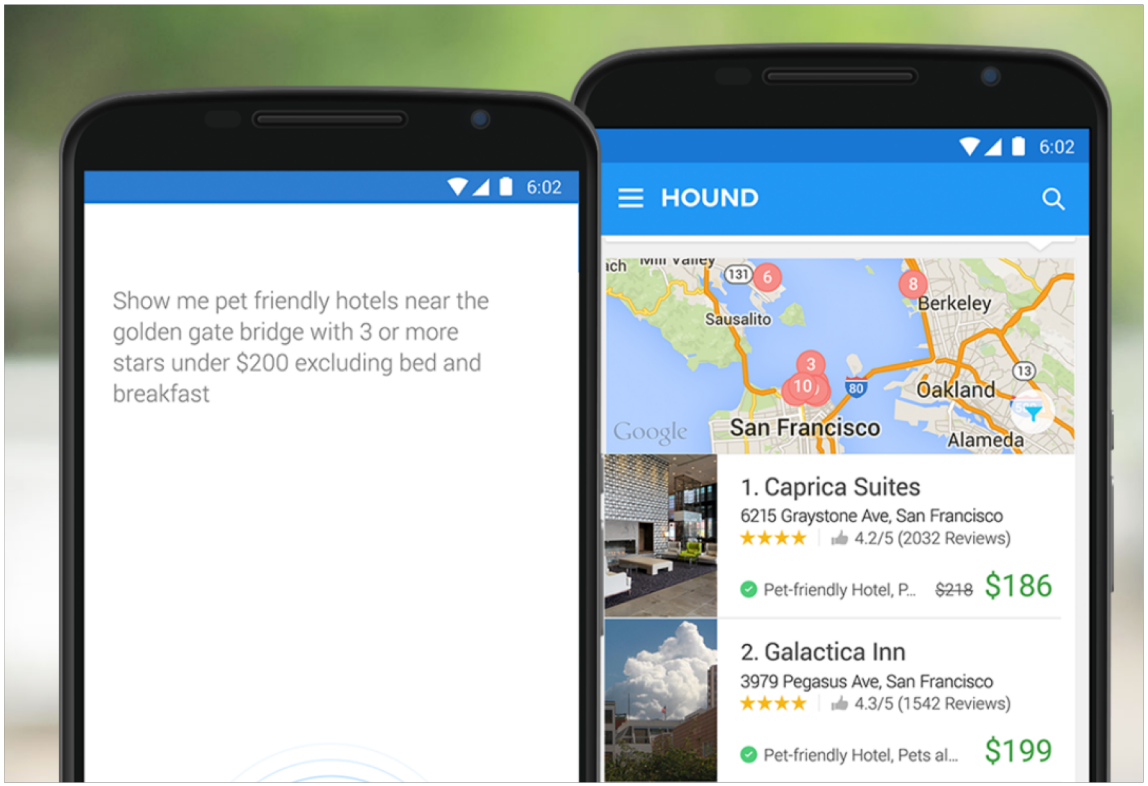
- we have short term memory
- eliminate repetitive word

```
Currently, Sun is trading at 32, up 1/2 since yesterday
           SGI is                 23, down 1/4
           IBM is                 69, up 1/8
```

Ambiguous Silence

- is speech recognizer working on something? did it not hear?
- users need immediate feedback if response is slow

SoundHound - modern example



http://www.soundhound.com/hound
https://www.youtube.com/watch?v=M1ONXea0mXg

Summary

Touchless interfaces is a one-state input device!

The mapping of sensor information to commands is computationally expensive, ambiguous and error prone.

In designing touchless interfaces, a key objective is to ensure that the system fails gracefully, and errors are managed properly.