TOOLKIT SUPPORT FOR GESTURAL INPUT TECHNIQUES

LU HAN
What is a gesture?

- “A gesture is a motion of the body that contains information. Waving goodbye is a gesture. Pressing a key on a keyboard is not a gesture because the motion of a finger on its way to hitting a key is neither observed nor significant. All that matters is which key was pressed”.

- Kurtenbach and Hulteen (1990)
What is gesture recognition?

• A topic in computer science and language technology with the goal of interpreting human gestures via mathematical algorithms.

• Can originate from body motion, but mostly associated with movement of the hand
Our obsession/fantasy with gestural inputs as seen in movies:

- Iron Man
- Minority Report
Benefit of gestural based interfaces

- Higher degree of freedom
- Natural to humans compared to traditional GUI
- Control or interact with devices without physically touching them
- A way for computers to understand our language – bridging human and machine interactions
- Faster/easier interaction: example – slide to unlock smart phones
Major application areas of gesture recognition currently trending

- Automotive sector
- Consumer Electronics sector
- Transit sector
- Gaming sector
Gesture types:

• Offline gestures: Those gestures that are processed after the user interaction with the object. An example is the gesture to activate a menu.

• Online gestures: Direct manipulation gestures. They are used to scale or rotate a tangible object.
Gesture types:

• Offline gestures: A $1 Recognizer
• Online gestures: Smartphones, 3D gestural recognition
Gestures without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes

• Created by students and researcher at University of Washington and Microsoft
• 2-D single-stroke recognizer designed for rapid prototyping of gesture-based user interfaces
• Wanted to offer tool for rapid prototyping in design oriented environment
• present a “$1 recognizer” that is easy, cheap, and usable almost anywhere in about 100 lines of code.
$1$ Goals

- be resilient to variations in sampling due to movement speed or sensing
- support optional and configurable rotation, scale, and position invariance;
- require no advanced mathematical techniques (e.g., matrix inversions, derivatives, integrals);
- be easily written in few lines of code;
- be fast enough for interactive purposes (no lag);
- allow developers and application end-users to “teach” it new gestures with only one example;
http://depts.washington.edu/madlab/proj/dollar/index.html
$1 – four step algorithm

- 1. Resample
- 2. rotate once
- 3. scaled and translate
- 4. Find optimal angle for the best score
Step 1: resample

![Raw gesture comparison with different resolutions](image)
Step 2: rotate once
Step 3: scale and translate

- scaled to a reference square.
Step 4: Find Optimal Angle for the Best Score

• Recognition based on pre-defined shapes
Defining Multiple Instances
Limitations

• recognizer is a geometric template matcher

• cannot distinguish gestures whose identities depend on specific orientations, aspect ratios, or locations. For example, separating squares from rectangles, circles from ovals, or up-arrows from down-arrows is not possible without modifying the algorithm.

• does not use time, so gestures cannot be differentiated on the basis of speed
Online gestures: Smartphones

- **Tap**: Briefly touch surface with fingertips.
- **Double tap**: Rapidly touch surface twice with fingertip.
- **Drag**: Move fingertip over surface without losing contact.
- **Flick**: Quickly brush surface with fingertip.
- **Pinch**: Touch surface with two fingers and bring them closer together.
- **Spread**: Touch surface with two fingers and move them apart.
- **Press**: Touch surface for extended period of time.
- **Press and tap**: Press surface with one finger and briefly touch surface with second finger.
Gestures on the IOS

• Apple provides sets of HCI guidelines and standards for reference and best practice

Gestures on the IOS

• Apple provides framework called UIKit
• UIKit responds to user interactions and system events, access various device features, enable accessibility, and work with animations, text, and images.
Gestures on the IOS

• The UIGestureRecognizer class as an abstract class

• Series of sub classes for common gestures...
  • UITapGestureRecognizer
  • UISwipeGestureRecognizer
  • UIPanGestureRecognizer
  • UIPinchGestureRecognizer
  • UIRotationGestureRecognizer
  • UILongPressGestureRecognizer
  • UIScreenEdgePanGestureRecognizer
override func viewDidLoad() {
    super.viewDidLoad()

    let gestureRecognizer = UITapGestureRecognizer(target: self, action: "handleTap:"
    self.view.addGestureRecognizer(gestureRecognizer)
}

func handleTap(gestureRecognizer: UIGestureRecognizer) {
    let alertController = UIAlertController(title: nil, message: "You tapped at \(gestureRecognizer.locationInView(self.view))", preferredStyle: .Alert)
    alertController.addAction(UIAlertAction(title: "Dismiss", style: .Cancel, handler: { _ in })))
    self presentViewController(alertController, animated: true, completion: nil)
}
Gestures on the Android

• Similar to IOS, Android provides GestureDetector class for detecting common gestures.
• Some of the gestures it supports include onDown(), onLongPress(), onFling(), and so on.
• You can use GestureDetector in conjunction with the onTouchEvent() method for event handling
3D gesture handling
3D gesture handling - inputs

• The kinetic user interfaces (KUIs) are an emerging type of user interfaces that allow users to interact with computing devices through the motion of objects and bodies.
3D gesture handling - algorithm

• 3D model-based algorithms
• Skeletal-based algorithms
• Appearance-based models
3D model-based algorithms
Skeletal-based algorithms
Appearance-based models
wired gloves
Gesture-based controllers
Wiimote uses IR triangulation to determine pointer position.
Depth-aware cameras

**Kinect 2 - Specs**

**Hardware:**
- **Depth resolution:** 512 × 424
- **RGB resolution:** 1920 × 1080 (16:9)
- **FrameRate:** 60 FPS
- **Latency:** 60 ms
Kinect Software Development kit

• designed for Windows applications (but many on the web have developed IOS friend wrappers and libraries)

• could lead to innovative Kinect-ready software for industries like education, healthcare and transportation

• Kinect for Windows SDK commercial license for app release
Kinect Software Development kit
Conclusion

• Exciting new field with a lot of research and development
• Becoming more popular with increasing popularity of smart phones and VR/AR applications
• There are still some limitations to gesture-based UI
  • Noise
  • Difficulty of recognizing the right thing
  • Tiredness?