TouchCU

Trenton Bishop, Blake LaFleur, Thales Lessa, Yizhou Hu, Matthew Spector

Texas Christian University - Computer Science Department

2013-2014 Senior Capstone Project

Agenda

- Project Overview
- Microsoft Kinect
- Gestures and Voice Commands
- System Design
- Obtaining Windows Screen Position
- Challenges
- Current State
- Q&A

Preview

Project Overview

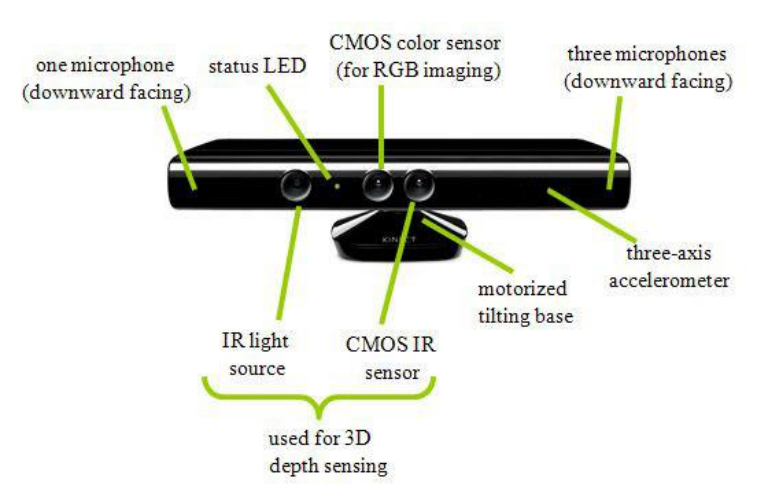
- Easy way to interact with devices.
- Growth of touch technology integration.
- Increased demand by users wanting new/innovative ways to interact.
- TouchCU was born.

Create a standalone application for the Windows 8 Operating System that will turn any flat surface into a multi-touch screen.

Minimal hardware requirements

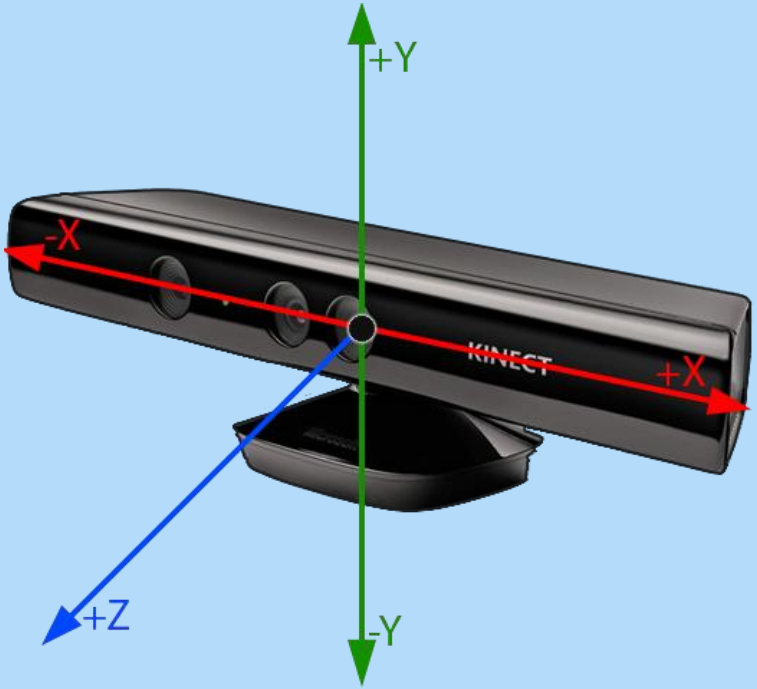
- Windows 8 PC
- Standard Projector
- Microsoft Kinect for Windows
- Flat, non-reflective surface

Microsoft Kinect for Windows



- Connects to a Windows PC
- Maximum capture rate of 30fps

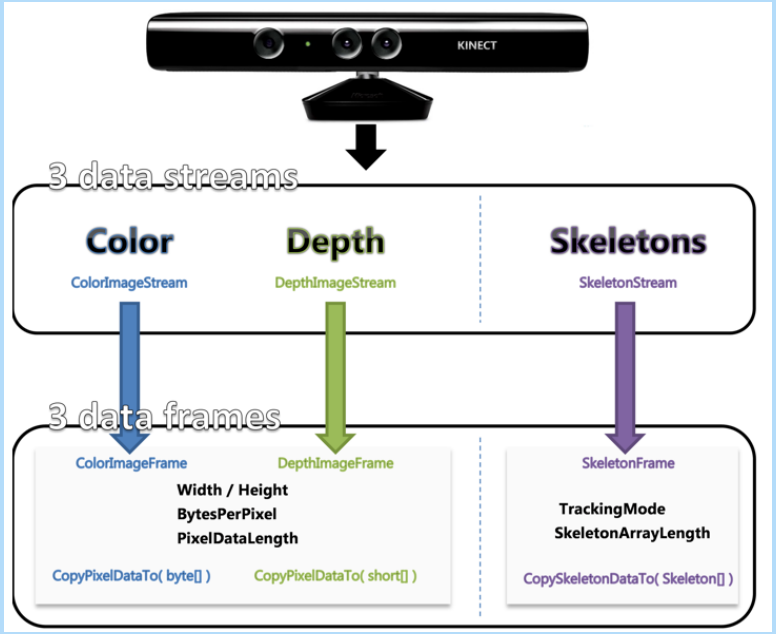
Kinect for Windows



- Joint data represented as (X,Y,Z) coordinates
- Distance in meters from Kinect origin

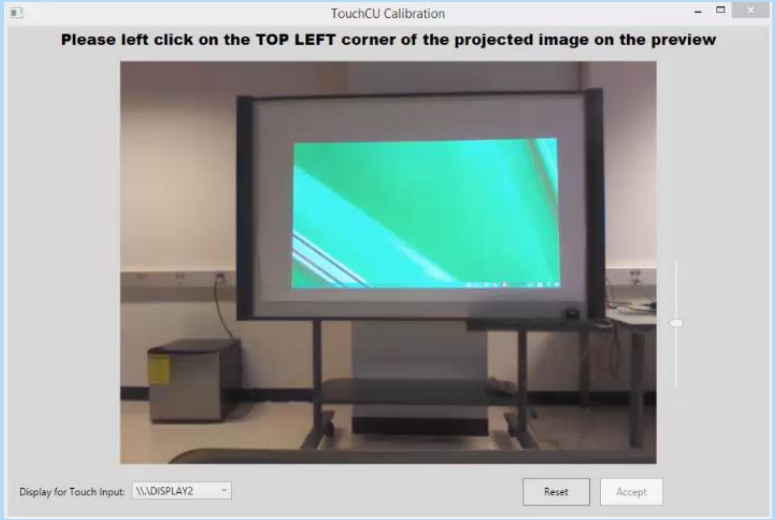
Kinect Data Streams

- 3 visual data streams
  - Color
  - Depth
  - Skeleton

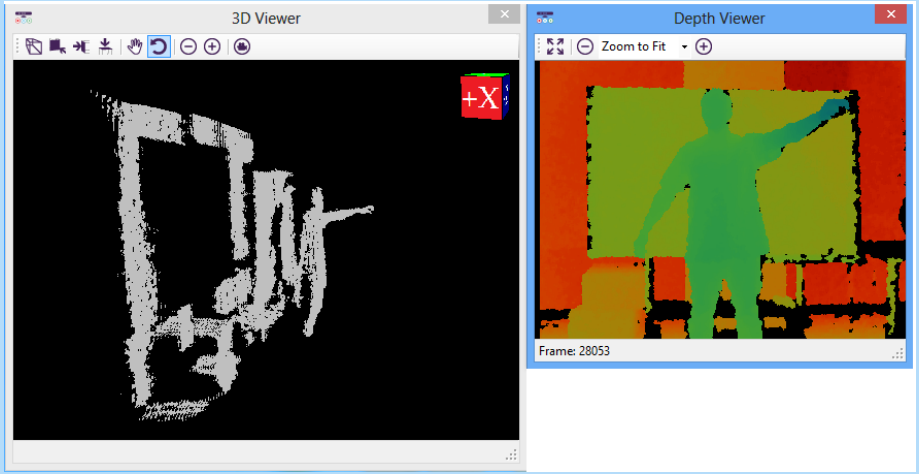


Kinect Color Stream

- Used for calibration
- Mapped to Depth Stream
- Projected image seen from Kinect view



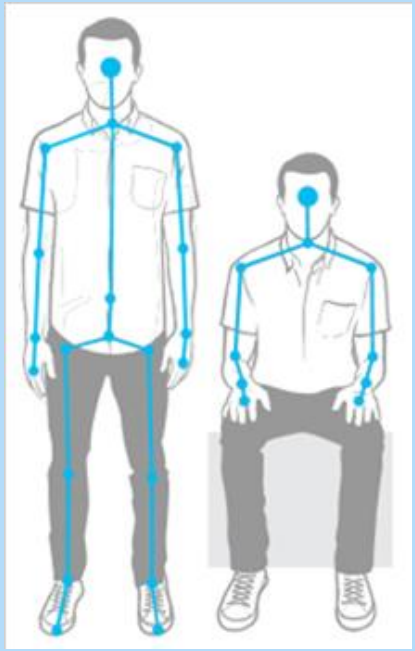
Kinect Depth Stream

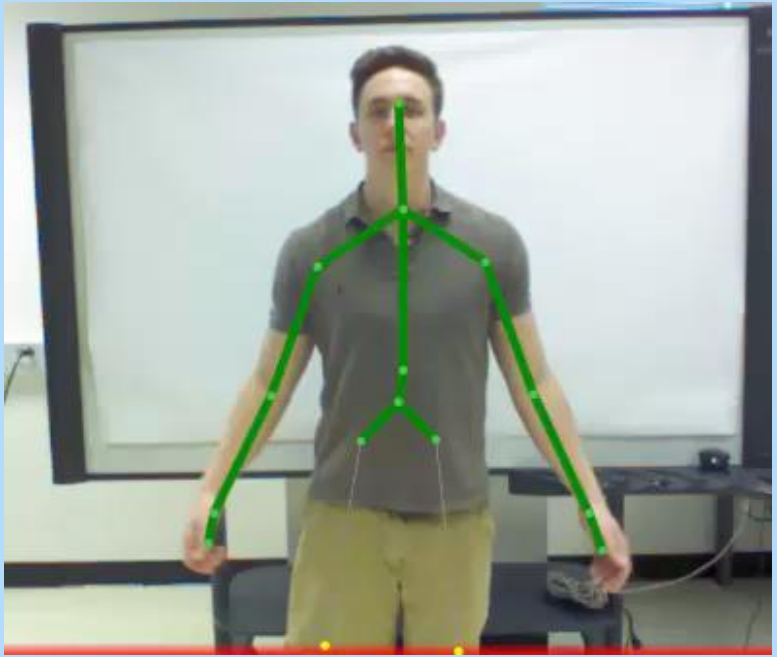


- Used for calibration and depth tracking
- Mapped to Skeleton Stream

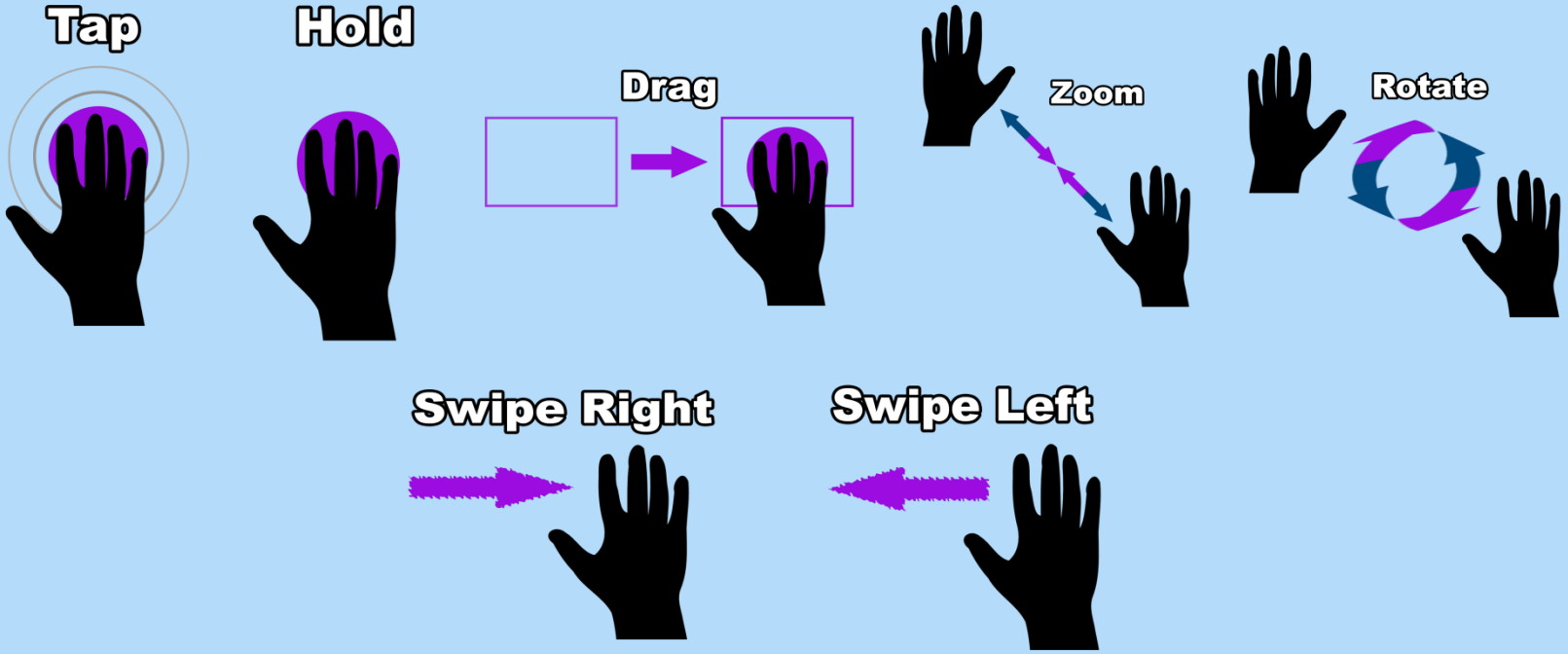
Kinect SDK Skeleton Stream

- Tracks up to 20 joints (default), 10 (seated)



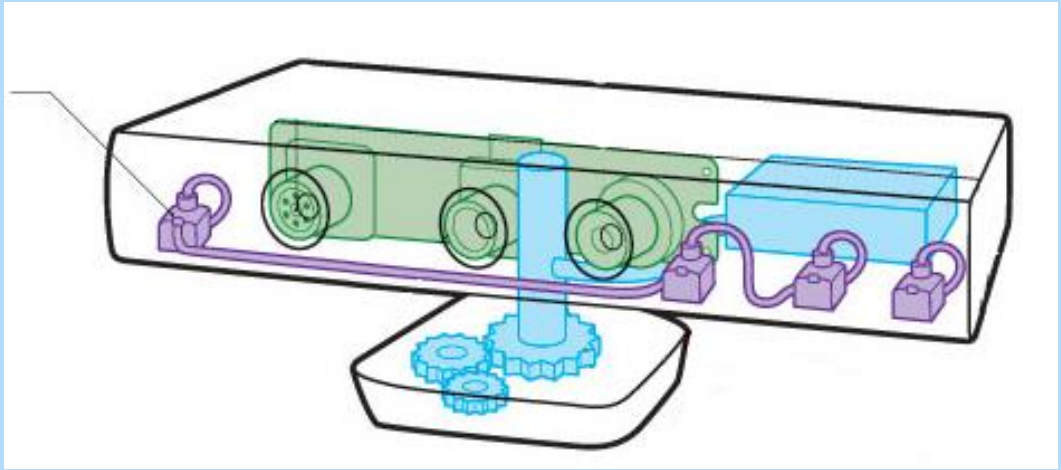


Gestures



Kinect Audio Stream

- Used for voice commands



Voice

Addie

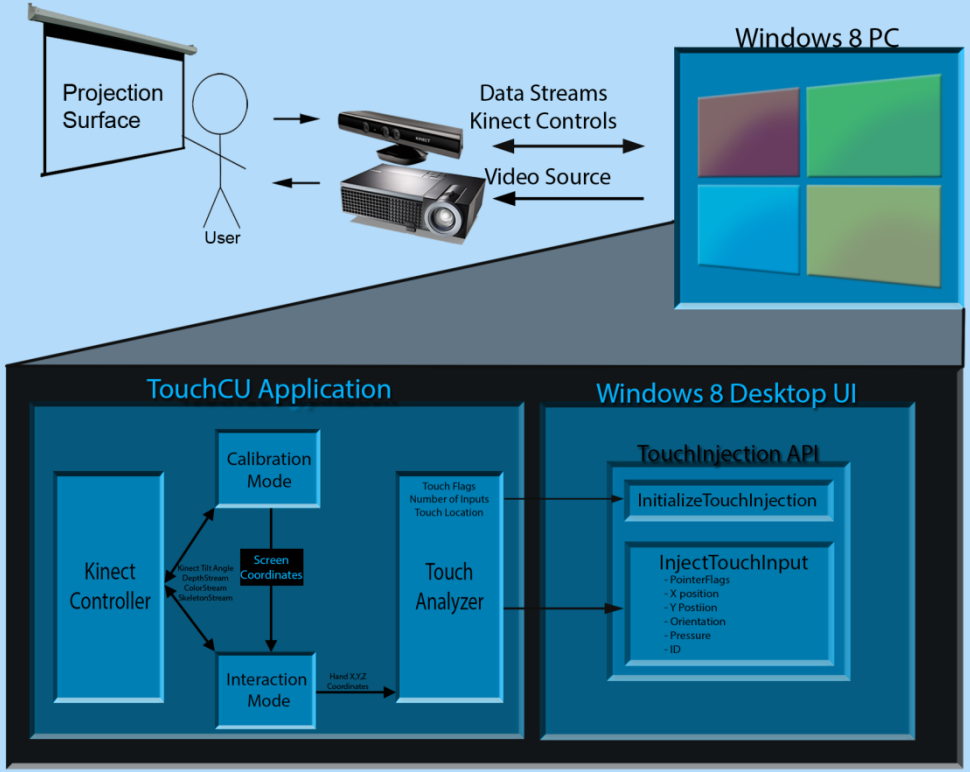
↓

Open / Close

↓

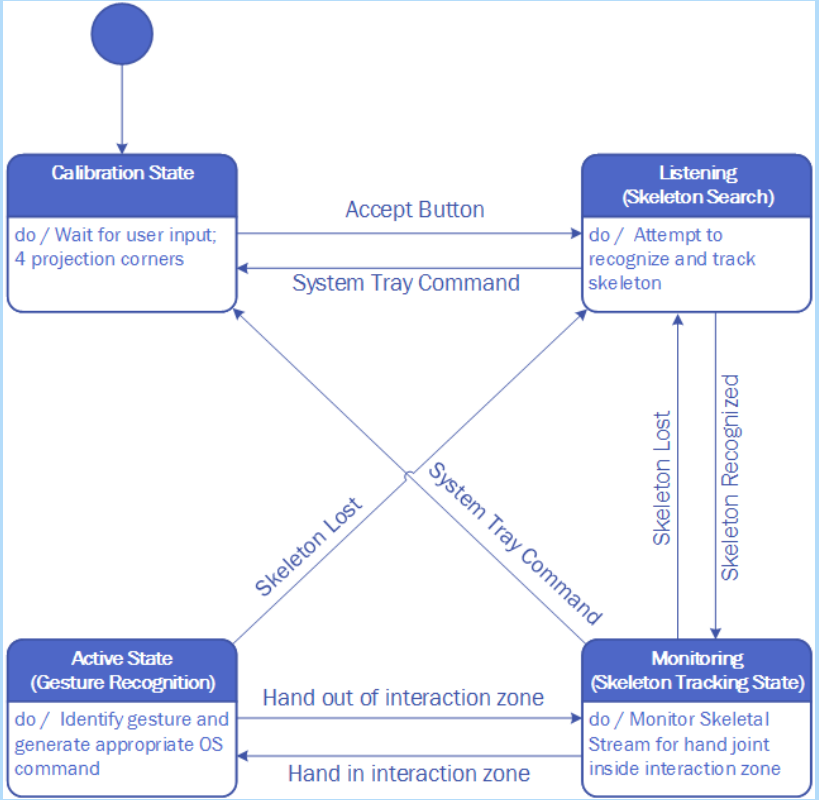
Start Menu
Window
My Documents
Settings
Debug

System Architecture

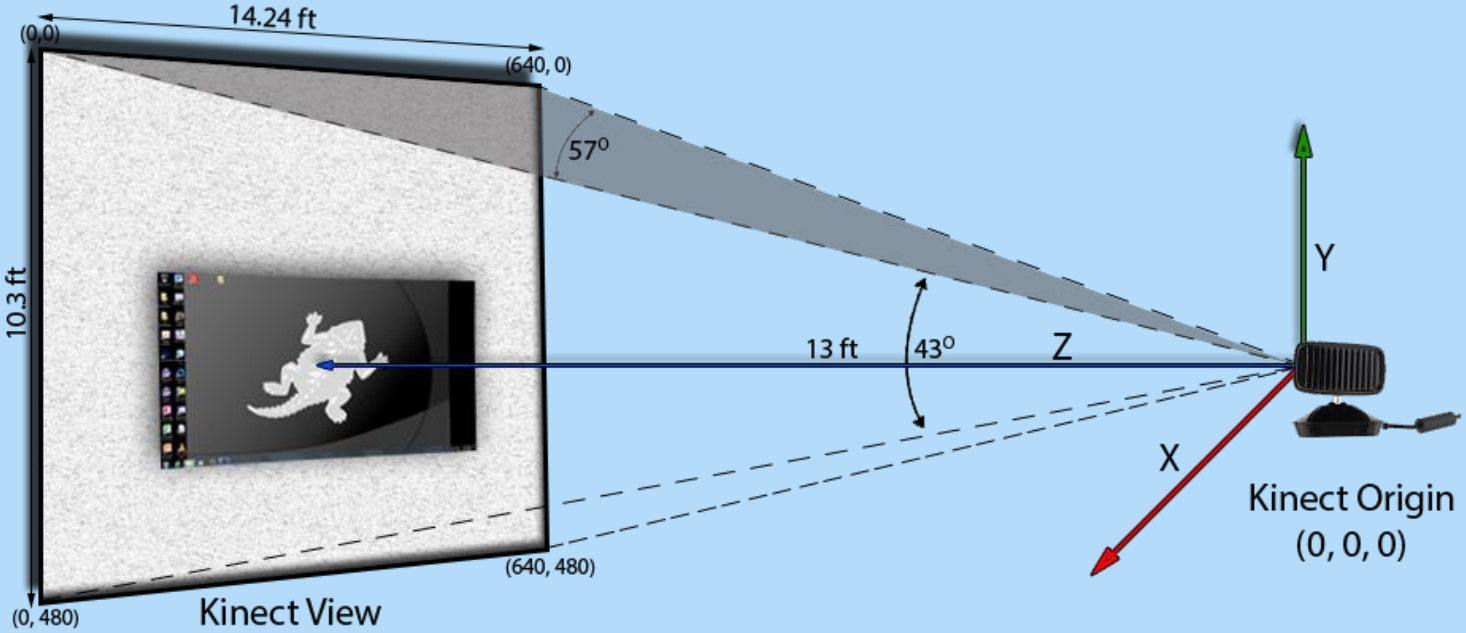


Application States

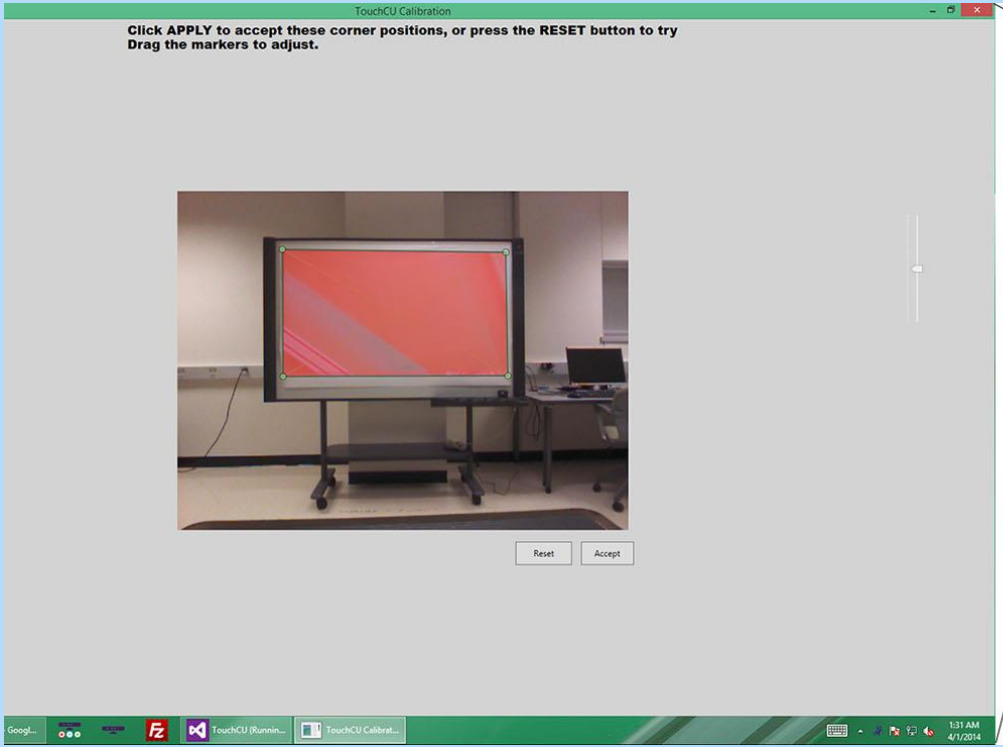
State Name	Description
Calibration	Calibrating the screen size
Listening	Looking for available skeletons to track
Monitoring	Looking for gestures
Active	Gesture recognized, sending input to OS

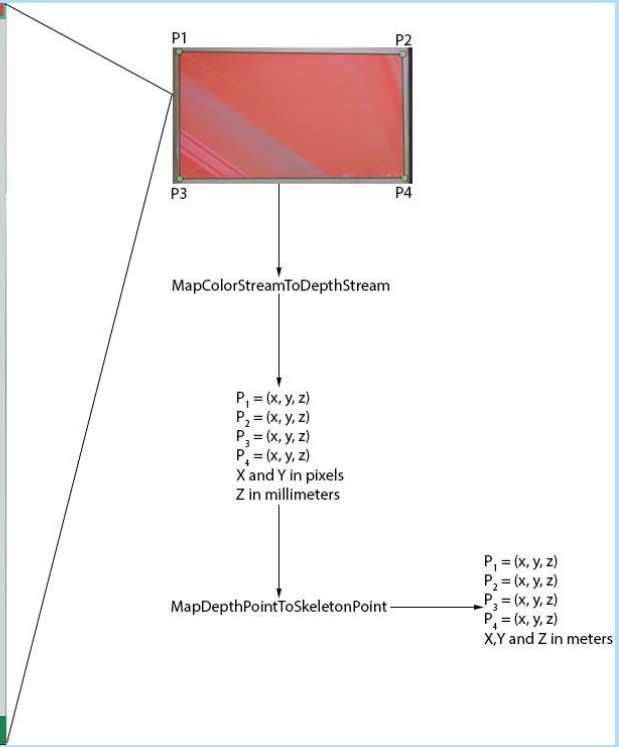


Step 1: Setup

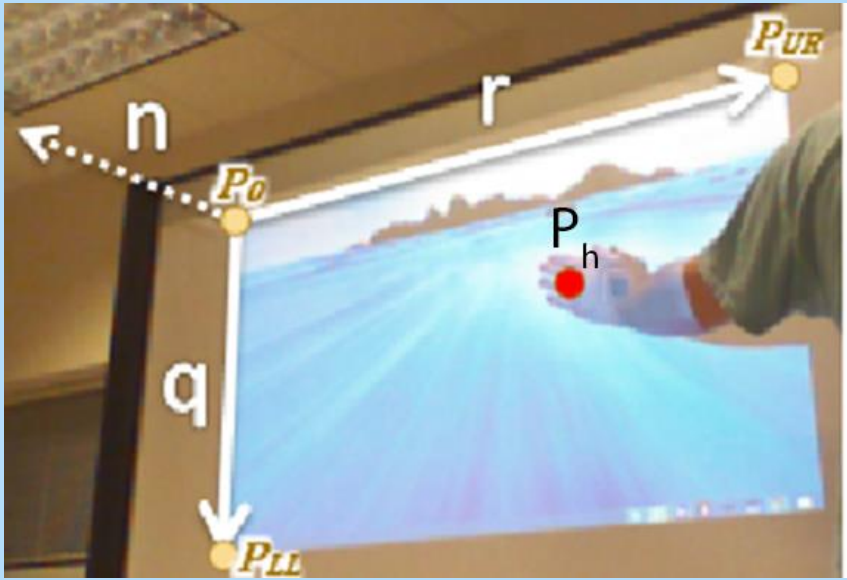


Step 2: Calibration





Step 3: Determine Hand Position on Projection Screen



$P_h(x, y, z) = x_0 r + y_0 q + z_0 n$

$ScreenLocationX = Width * X_0$

$ScreenLocationY = Height * Y_0$

$P_S = InjectTouch(ScreenLocationX, ScreenLocationY)$

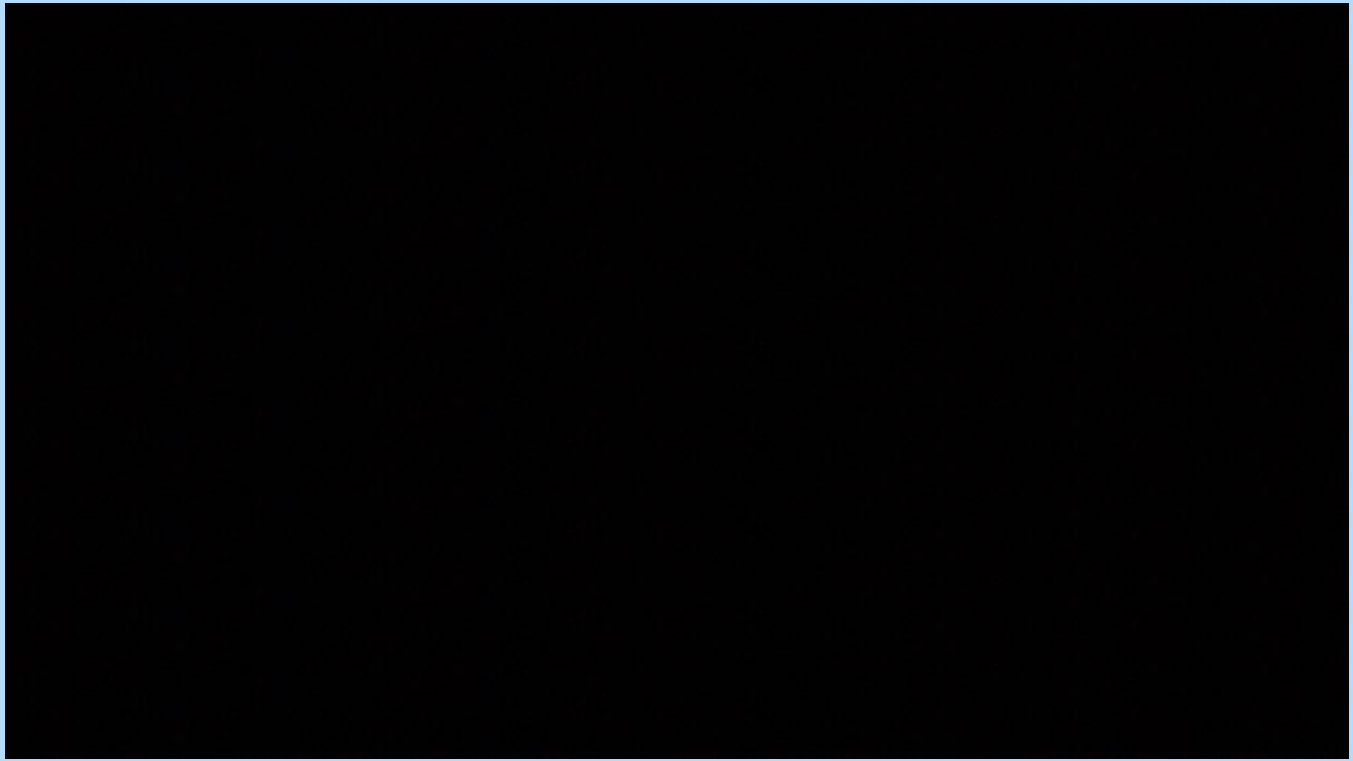
Data Filtering

- Built-in Kinect filtering
  - Holt double exponential smoothing method parameters:
    - Smoothing
    - Correction
    - Prediction
    - JitterRadius
    - MaxDeviationRadius

Challenges

- Accuracy of Kinect
- Accuracy vs. Responsiveness
- System Tray icon disposal
- Simulate a tap

Current State



Q&A