



TouchCU

Senior Capstone Project 2013 – 2014

Computer Science Department

Texas Christian University



The Team

- Trenton Bishop: Documentation Lead
- Yizhou Hu: Algorithm Design Lead
- Blake LaFleur: Technical Lead
- Thales Lessa: Testing Lead
- Matthew Spector: Project Lead



Preview





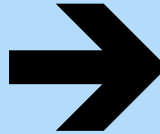
Agenda

- Project Overview
- Microsoft Kinect
- Fall and Spring Semester Work
- Project Support Environment
- System Design
- Gestures
- Using TouchCU
- Demos
- Challenges
- Results and Future Work
- Q&A



Project Overview

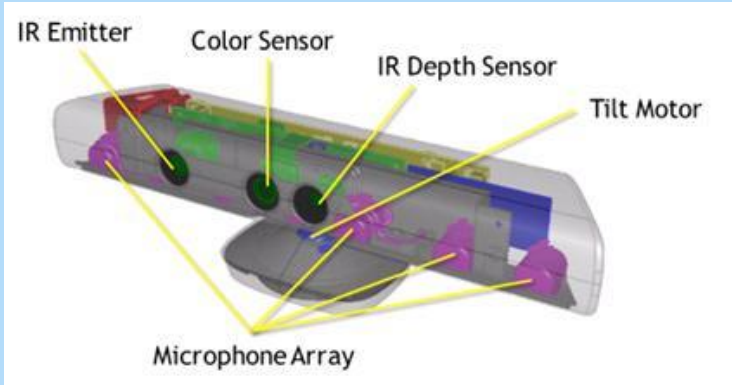
- Easy way to interact with devices.
- Growth of touch technology integration.
- Increased demand by users wanting new/innovative ways to interact.
- TouchCU was born.



Create a standalone application for the Windows 8 Operating System that implements voice and touch interaction on any flat surface.

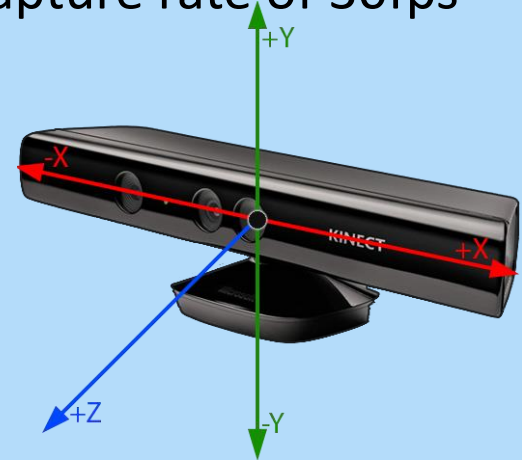
Minimal hardware requirements
Windows 8 PC
Standard Projector
Flat, non-reflective surface
Microsoft Kinect for Windows

Microsoft Kinect



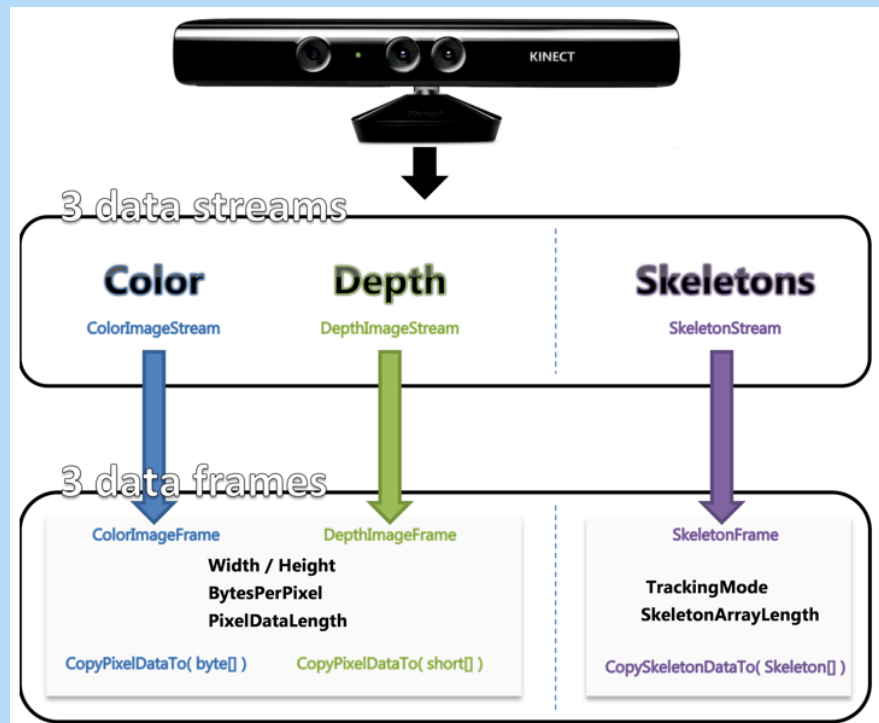
- Motion sensing input device for Xbox 360 and Windows PC
- Maximum capture rate of 30fps

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



Kinect Data Streams

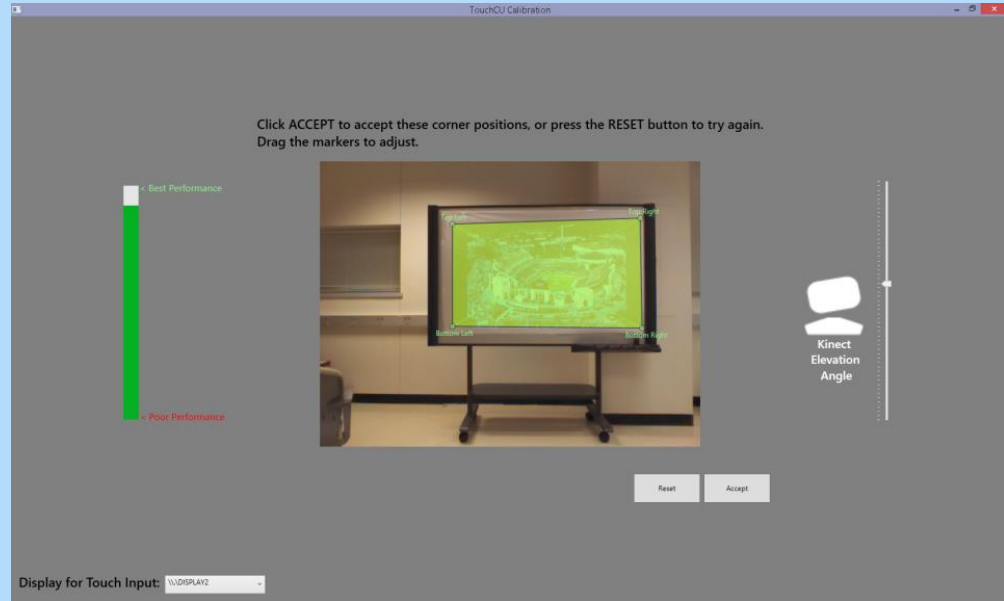
- 3 visual data streams
 - Color
 - Depth
 - Skeleton



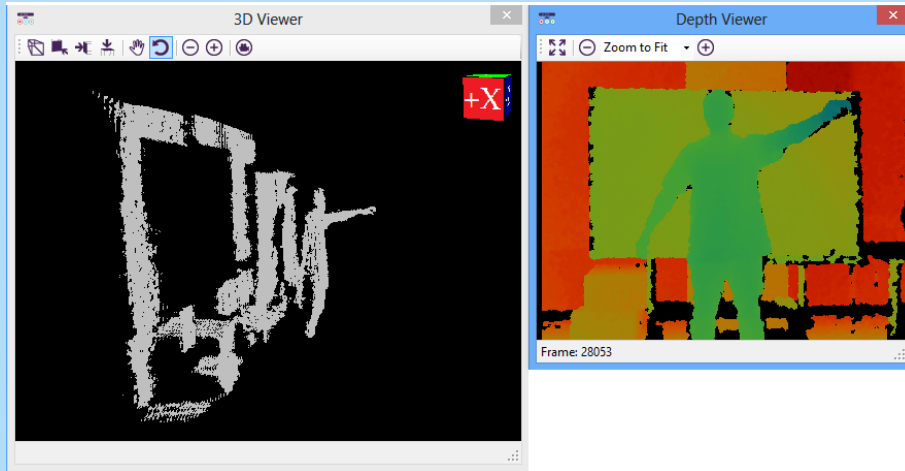


Kinect Color Stream

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



Kinect Depth Stream

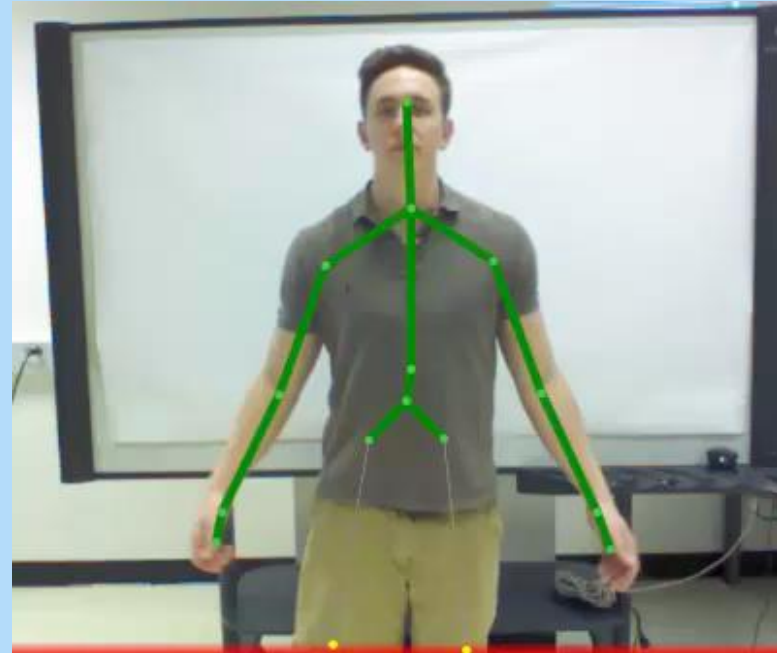
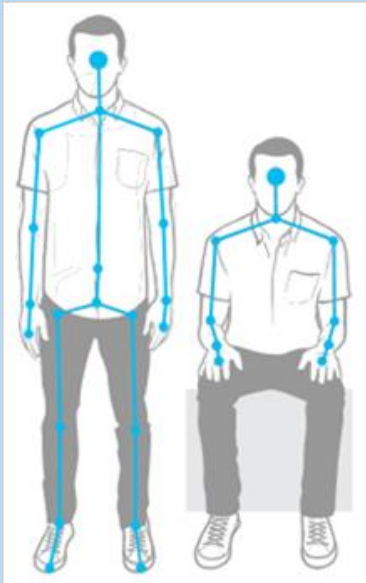


- Used for calibration and depth tracking
- Mapped to Skeleton Stream



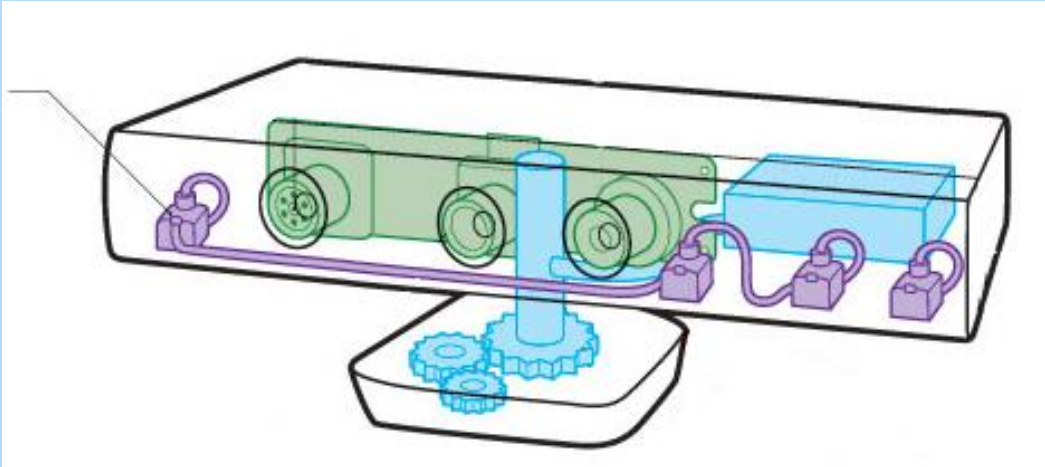
Kinect SDK Skeleton Stream

- Two skeletal tracking modes
 - Default – 20 joints
 - Seated – 10 joints



Kinect Audio Stream

- Used for voice commands



Voice

Addie



Open / Close



Start Menu
Window
My Documents
Settings
Debug



Milestones

- Iteration 1 12 Dec, 2013
- Iteration 2 30 Jan, 2014
- Iteration 3 28 Feb, 2014
- Iteration 4 01 Apr, 2014
- NTASC 05 Apr, 2014
- SRS 11 Apr, 2014
- Final Presentation 01 May, 2014



Fall 2013 Semester

- Project Selection
- Familiarization with project support environment
- Project Defined
 - Project Plan
 - Initial Requirements
 - Initial Design
- Iteration 1
 - Test constraints on the Kinect's ability to capture hand locations
 - Identify supported gestures - single, two-handed and air gestures
 - Implement manual screen calibration
 - Functional implementation of "Drag" gestures to OS



Spring 2014 Semester

- Iteration 2
 - Optimize screen calibration
 - Define voice commands
 - Implement debugging overlay and settings menu
- Iteration 3
 - Implement all pointer movements from Kinect to Windows
 - Implement touch-specific gestures
 - Implement voice commands



Spring 2014 Semester

- Iteration 4
 - Implement air gestures
 - Final system testing and bug fixing
 - Developer and User Manuals
 - User Acceptance Testing (UAT)
- Each Iteration
 - Planning
 - Requirements and Design adjustment
 - Documentation, Documentation, Documentation



Presentations

North Texas Area Student Conference



TCU Student Research Symposium





Project Support Environment

Programming Environment

- Kinect Developer Toolkit Software Development Kit (SDK)
- Kinect Studio for Windows v1.8.0
- Microsoft Windows 8 x64 Professional
- Visual Studio Pro 2012

Other Software

- Adobe Photoshop CS 6
- Camtasia Studio 8
- Core FTP Lite
- Google Drive
- GroupMe
- Handbrake
- iMovie
- Microsoft Office 2010
- Microsoft Visio 2013
- Subversion, Tortoise SVN



Kinect Studio

The screenshot displays the Kinect Studio software interface. At the top is the main menu with options: File, View, Timeline, Tools, and Help. Below the menu is a playback control bar with various icons for play, stop, and navigation. A 'Stream' section contains checkboxes for 'Color' and 'Depth', both of which are checked. The status bar at the bottom left indicates 'Connected: Stopped Injecting | Live Preview'.

The interface is divided into three main viewports:

- 3D Viewer:** Shows a 3D point cloud of a person standing in a room. A red wireframe bounding box is visible around the person. A small 3D coordinate system with axes labeled -Z, -Y, and X is shown in the top right corner.
- Depth Viewer:** Displays a depth map of the scene, where the person is represented in shades of green and yellow against a red background. The frame number '4731133' is shown at the bottom.
- Color Viewer:** Shows a standard color video feed of the person. The frame number '4635700' is shown at the bottom.

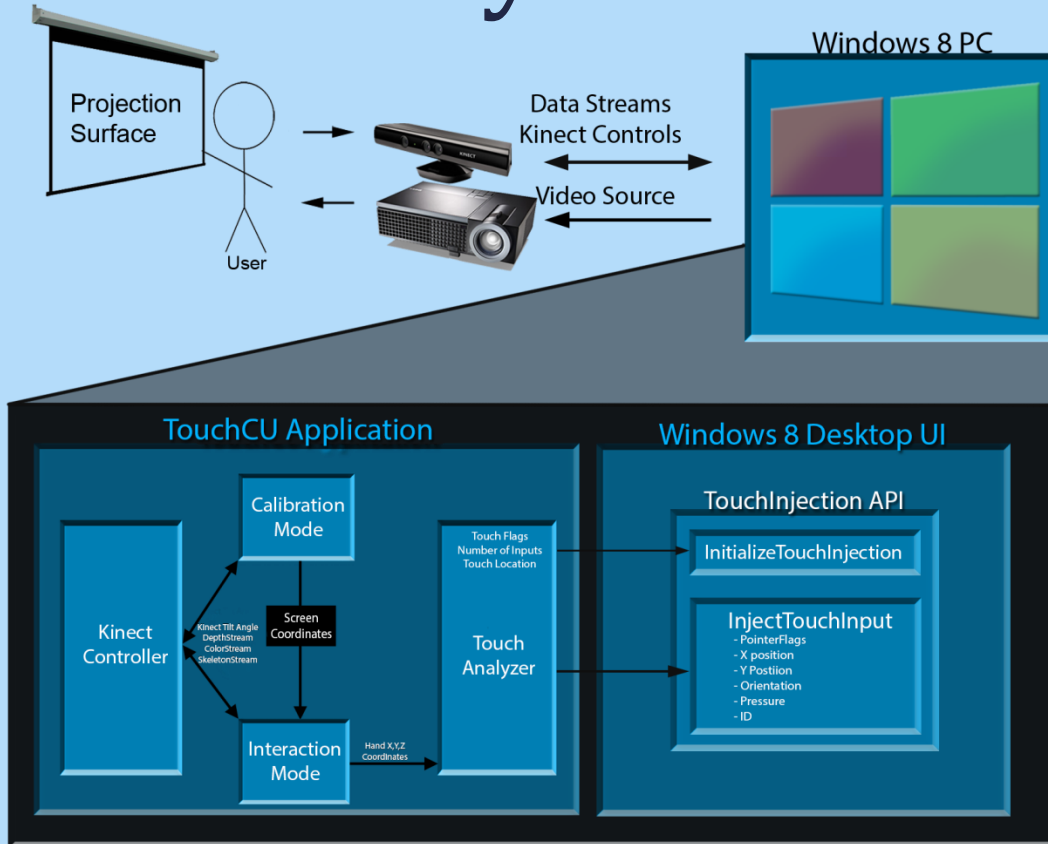
Code snippets are visible in the background of the viewports:

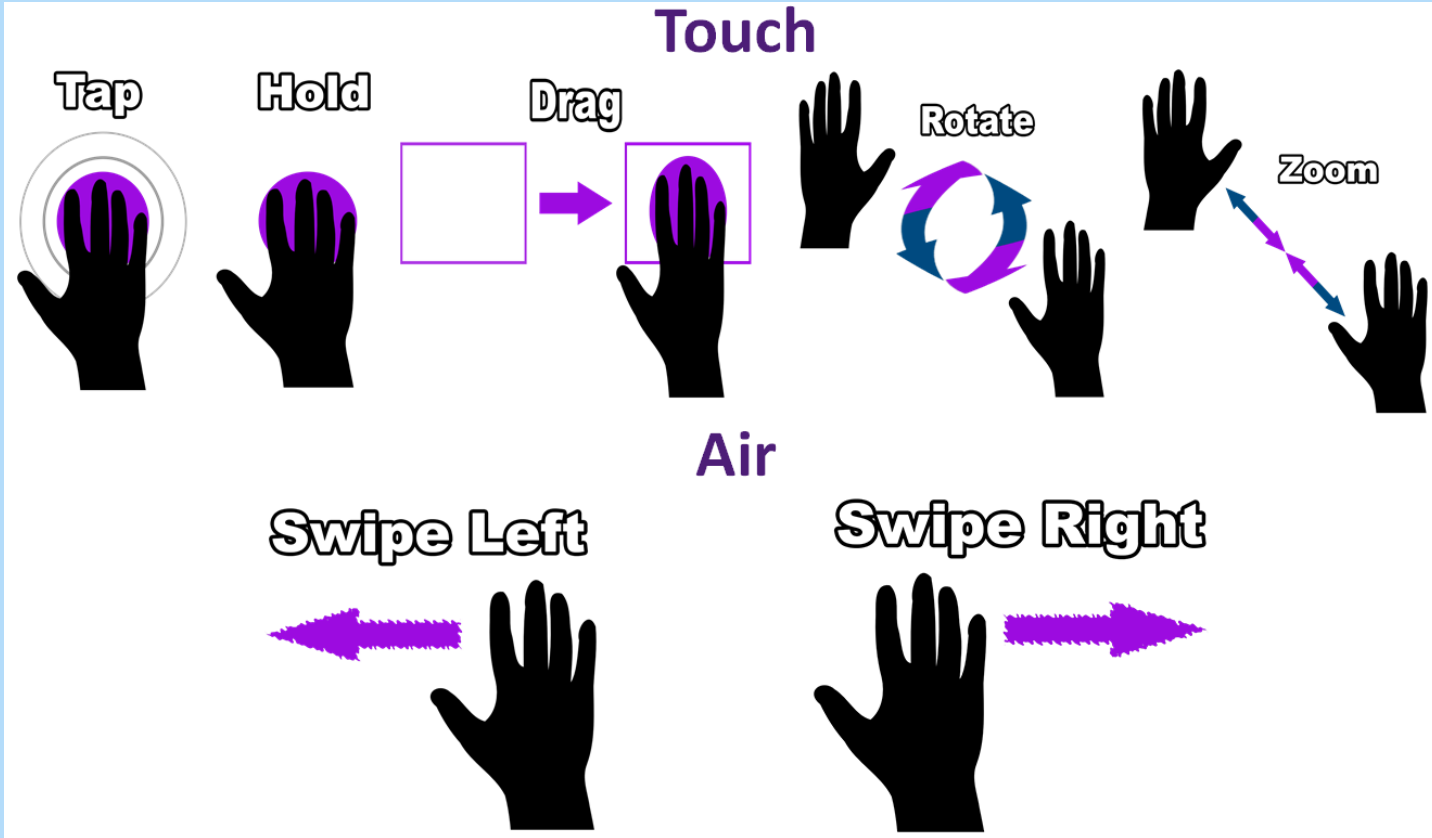
```
reen.Y + p.Y);  
/ + p.Y), 10, (uint)tIndex);  
  
Y + p.Y);  
| PointerFlags.INCONTACT;
```

```
gs.INRANGE | PointerFlags.INCONTACT;  
ON | TouchMask.PRESSURE;
```



System Architecture

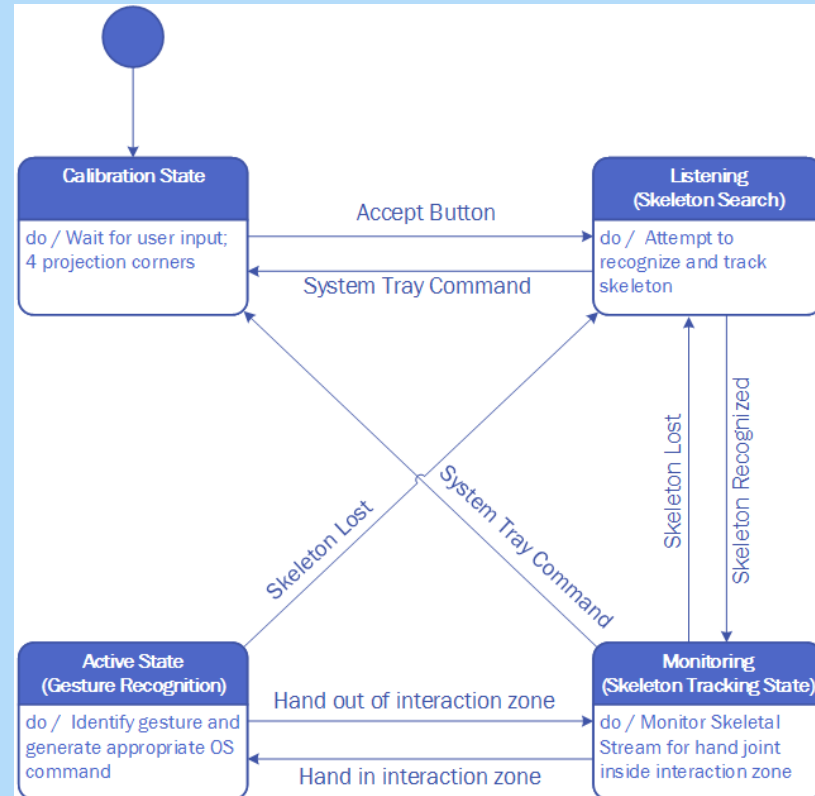






Application States

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

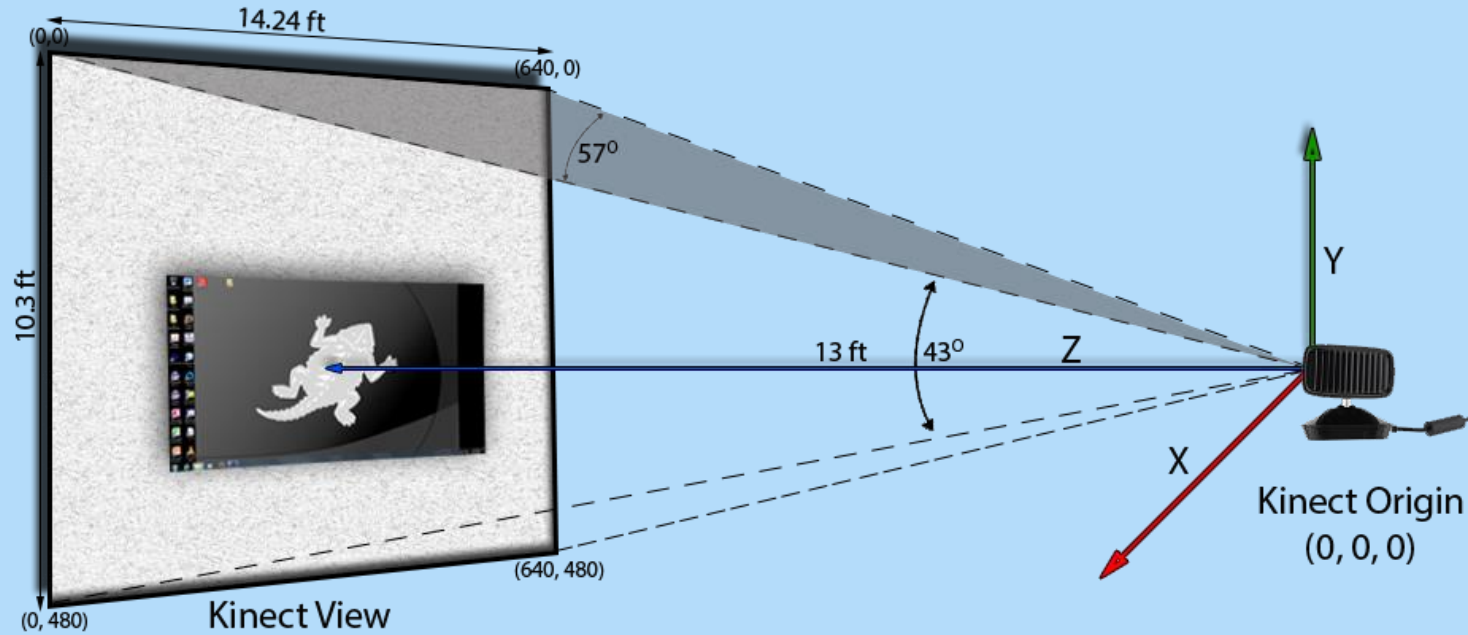




Using TouchCU



Step 1: Setup





Step 2: Calibration

TouchOU Calibration

Click ACCEPT to accept these corner positions, or press the RESET button to try again.
Drag the markers to adjust.

< Best Performance

< Poor Performance

Bottom Left Bottom Right

Kinect Elevation Angle

Reset Accept

Display for Touch Input: \\DISPLAY2

P1 P2
P3 P4

MapColorStreamToDepthStream

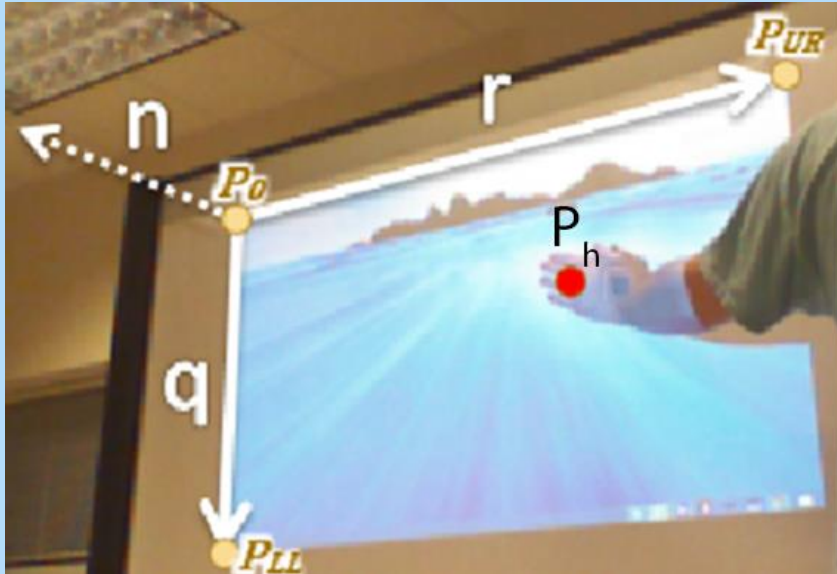
$P_1 = (x, y, z)$
 $P_2 = (x, y, z)$
 $P_3 = (x, y, z)$
 $P_4 = (x, y, z)$
X and Y in pixels
Z in millimeters

MapDepthPointToSkeletonPoint

$P_1 = (x, y, z)$
 $P_2 = (x, y, z)$
 $P_3 = (x, y, z)$
 $P_4 = (x, y, z)$
X, Y and Z in meters



Step 3: Determine Hand Position on Projection Screen



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

$$\text{ScreenLocationX} = \text{Width} * x_0$$

$$\text{ScreenLocationY} = \text{Height} * y_0$$

$$P_s = \text{InjectTouch}(\text{ScreenLocationX}, \text{ScreenLocationY})$$

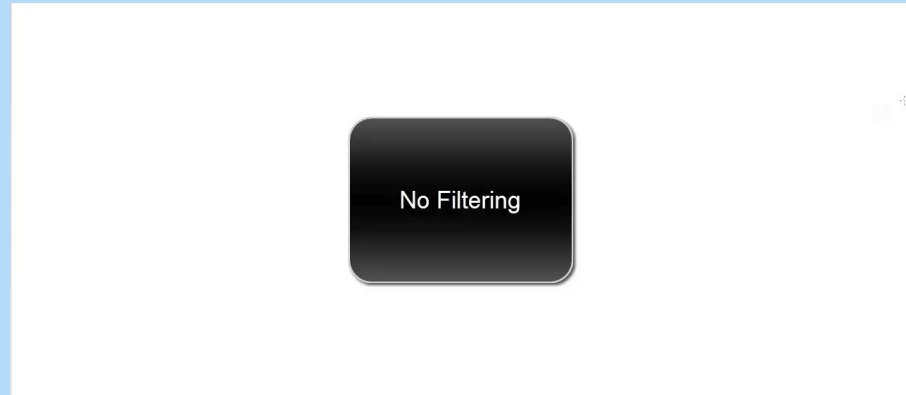




Data Filtering

Built-in Kinect filtering parameters

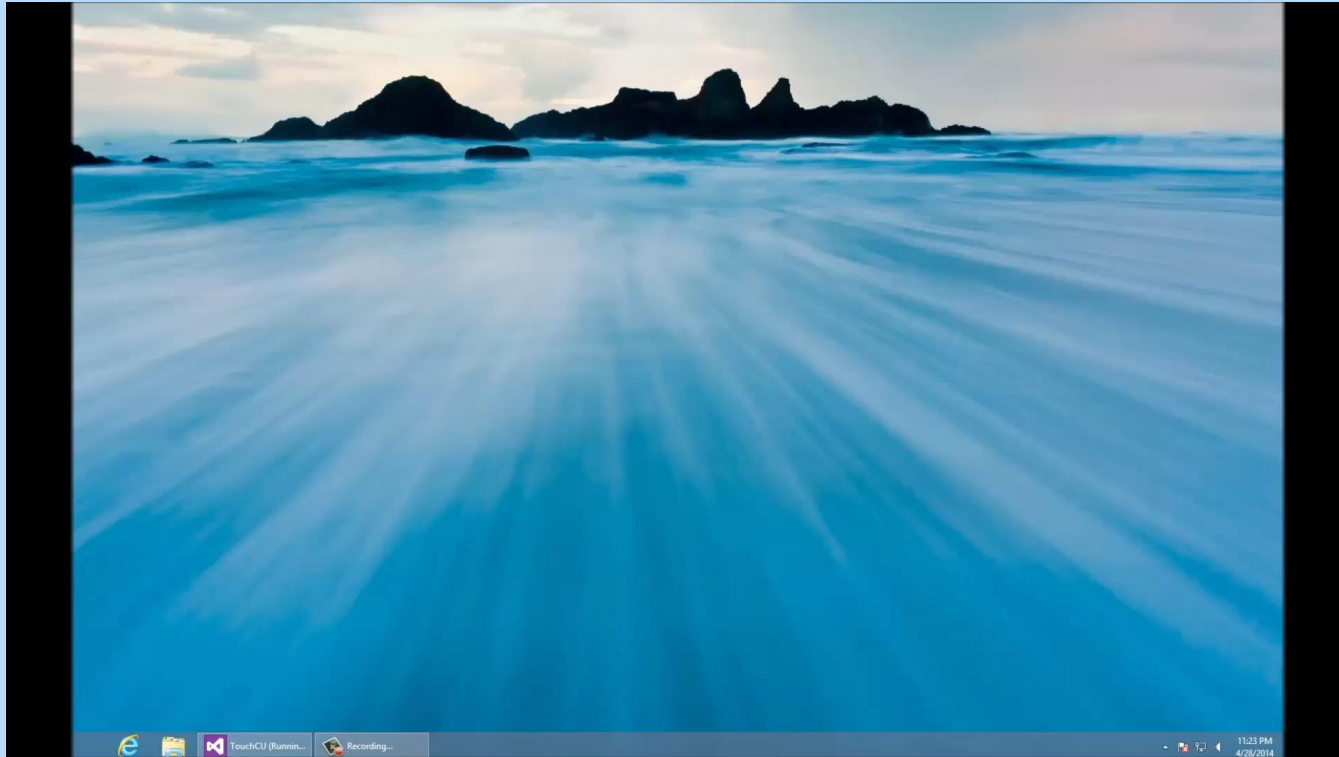
- Smoothing
- Correction
- Prediction
- Jitter Radius
- Max Deviation Radius



- Based on the Holt Double Exponential Smoothing method



Calibration + Settings



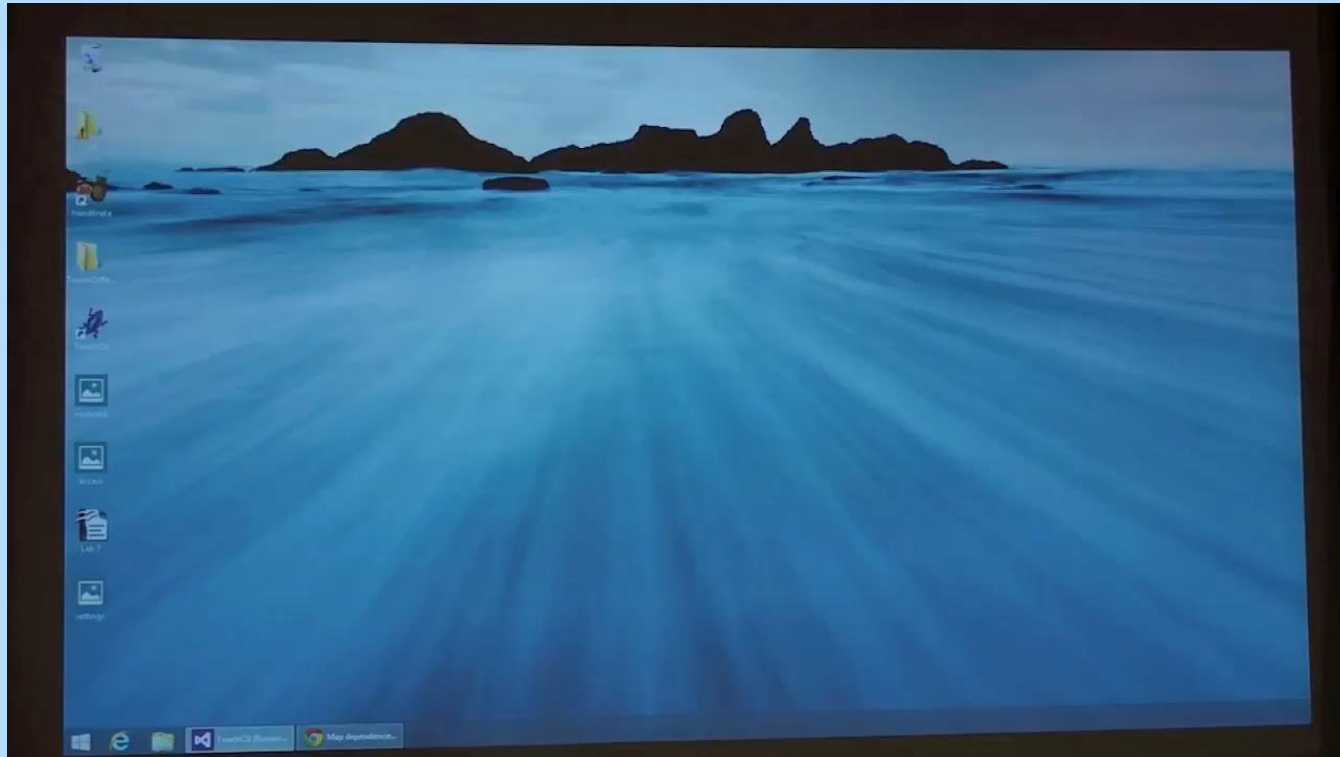


Voice Commands + Debug





Metro Screen + Maps





Air Gestures





Technical Challenges

- Kinect design and orientation
- Skeletal Tracking
- Accuracy vs. Responsiveness
- Windows 8 Touch Injection API
- Simulating gestures



General Challenges

- Scheduling and prioritizing
- Effective communication
- Accountability among team members
- Keeping documentation up-to-date with each iteration

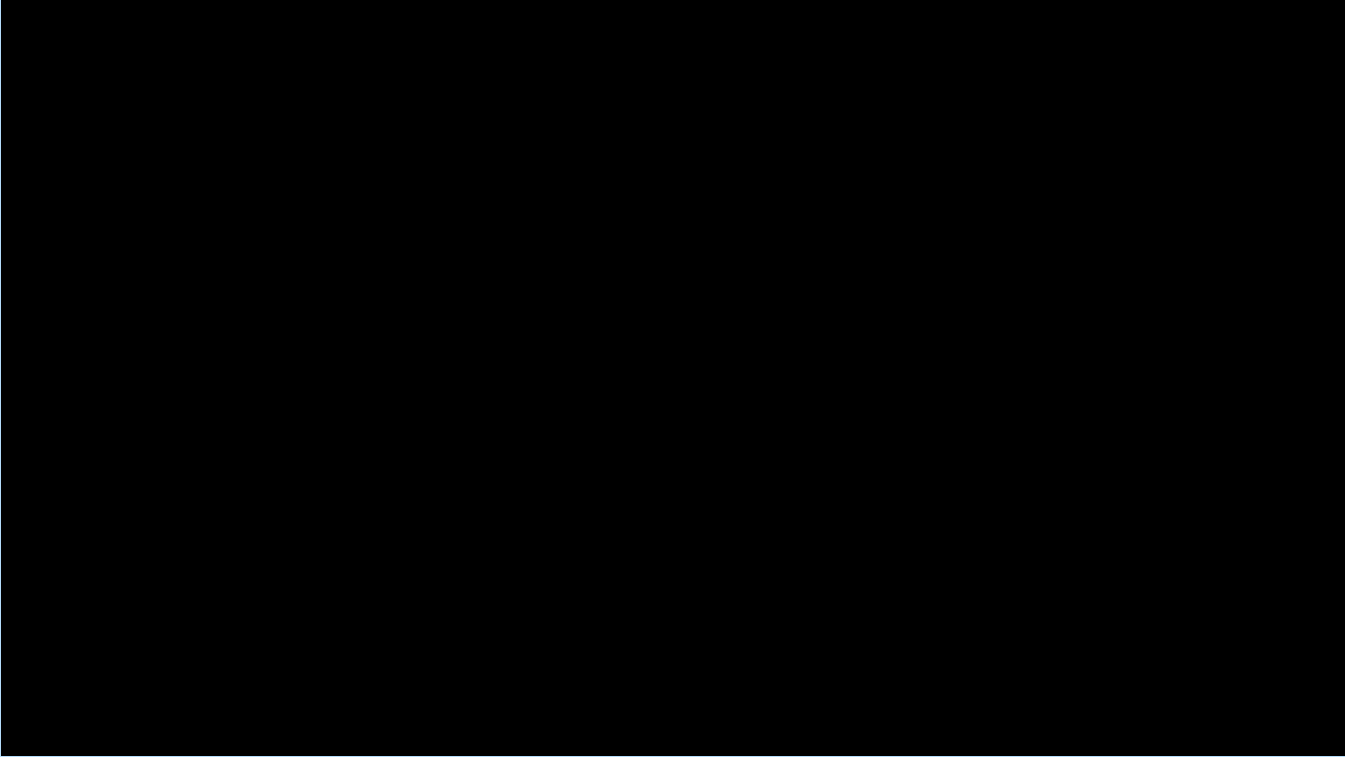


Results and Future Work

- Successfully allows user to interact with Windows 8
 - Response time meets Microsoft Touch Device standards of 500ms
 - Supports most Windows 8 compatible gestures
- Transition to the Kinect SDK 2.0 and the new Kinect from Microsoft
- Expand voice command/air gesture functionality
- Implement annotation mode
- Research use of open source SDK for cross-OS support



Fun with TouchCU





Thanks

- Project Advisor
 - Dr. Payne
- Equipment Donor
 - Patrick Anderson – Microsoft (TCU CS '93)
- Advice on Mathematics
 - Eric Elsken – TCU CS/Mathematics Double Degree Student

Thank you from Team



Questions?