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

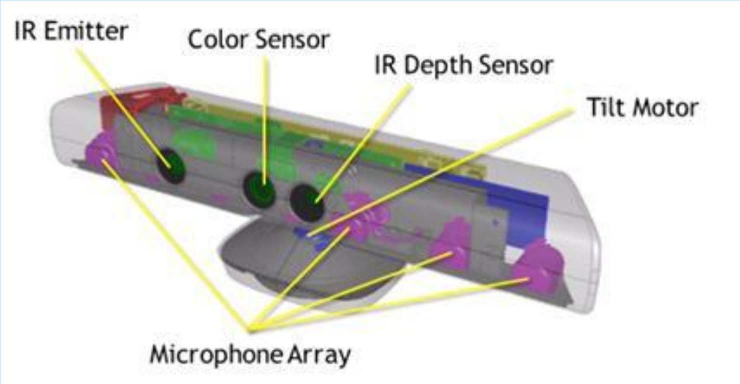
Windows 8 PC

Standard Projector

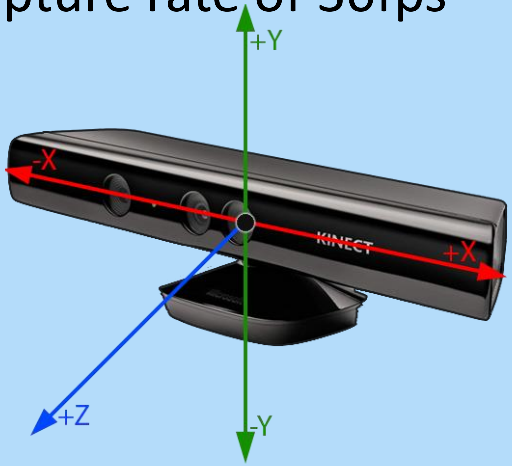
Flat, non-reflective surface

Microsoft Kinect for Windows

Microsoft Kinect



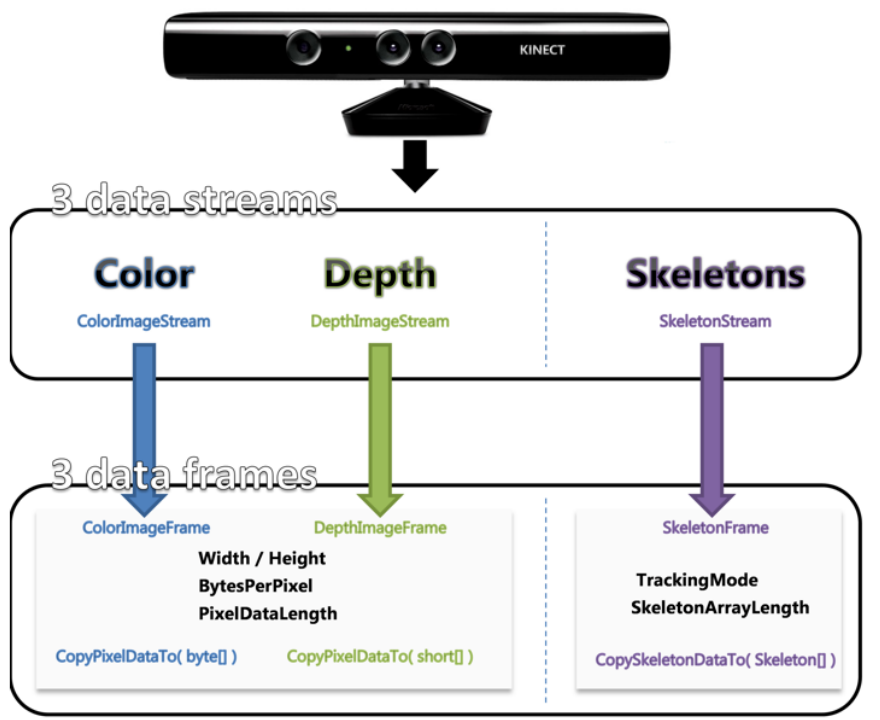
- Motion sensing input device for Xbox 360 and Windows PC
- Maximum capture rate of 30fps



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

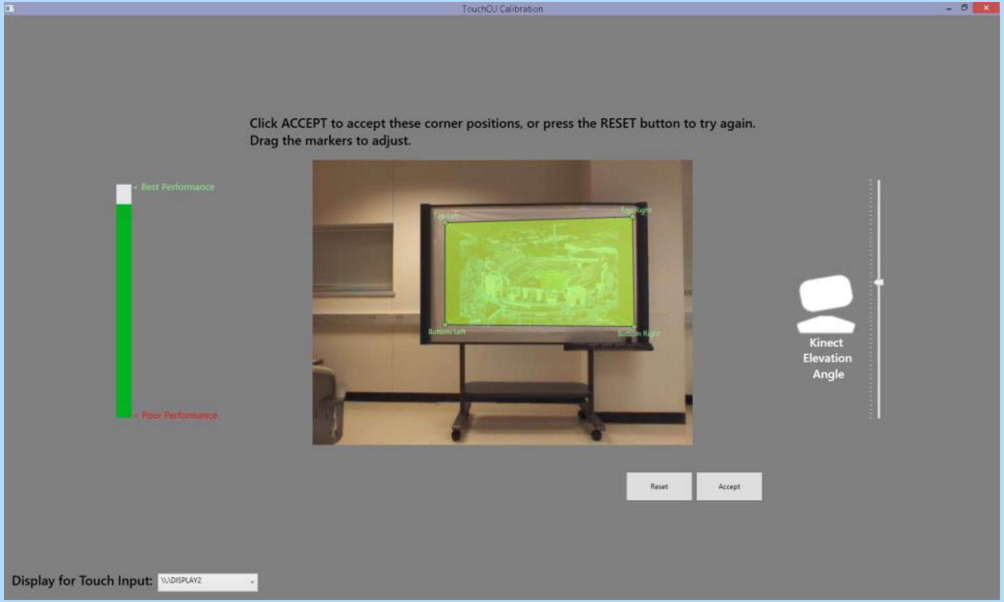
Kinect Data Streams

- 3 visual data streams
  - Color
  - Depth
  - Skeleton

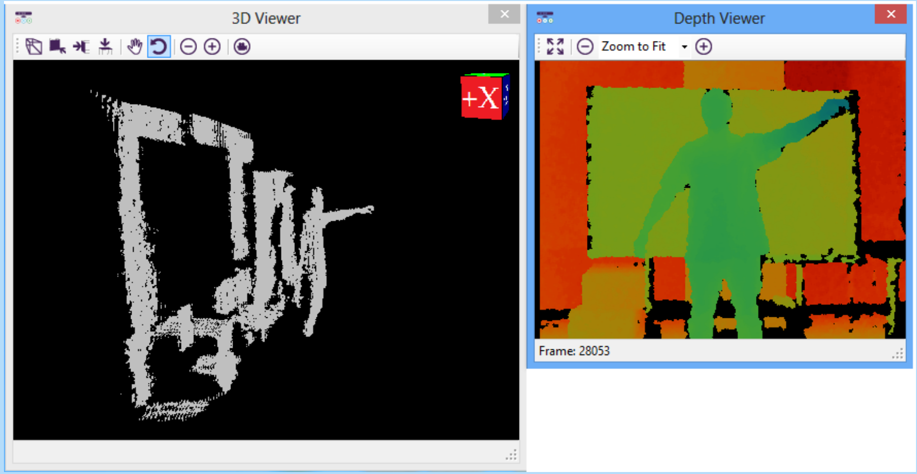


Kinect Color Stream

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



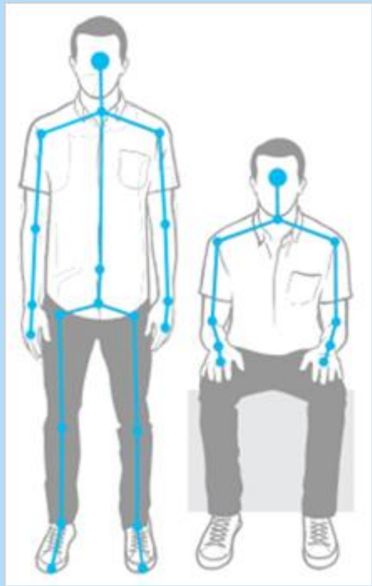
Kinect Depth Stream

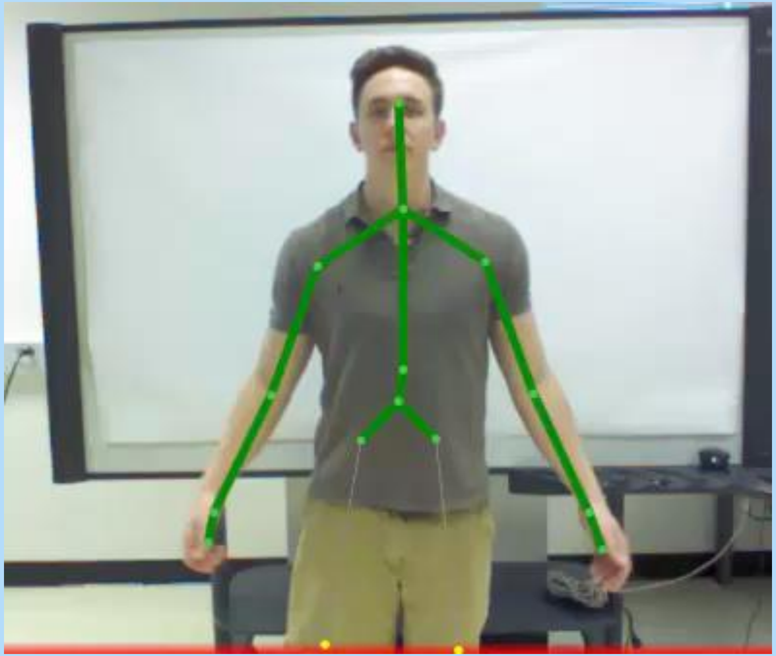


- Used for calibration and depth tracking
- Mapped to Skeleton Stream

Kinect SDK Skeleton Stream

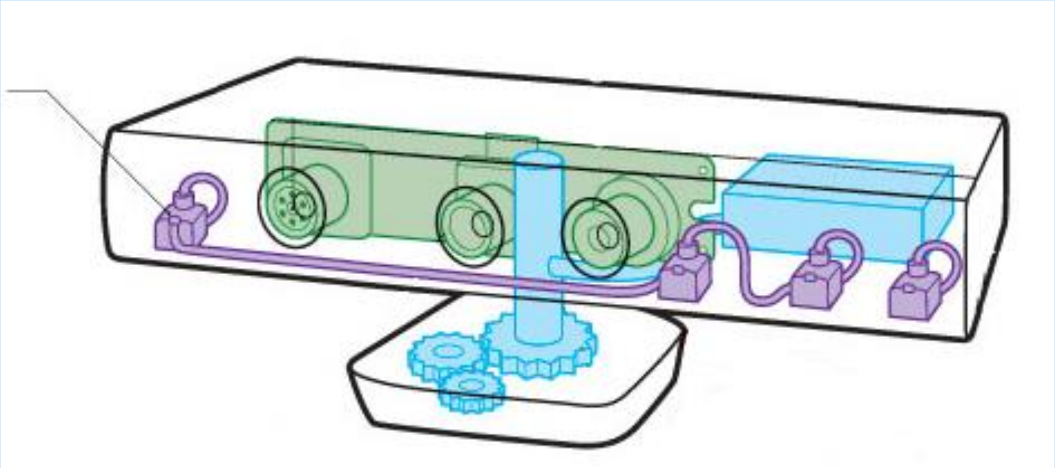
- Two skeletal tracking modes
  - Default – 20 joints
  - Seated – 10 joints

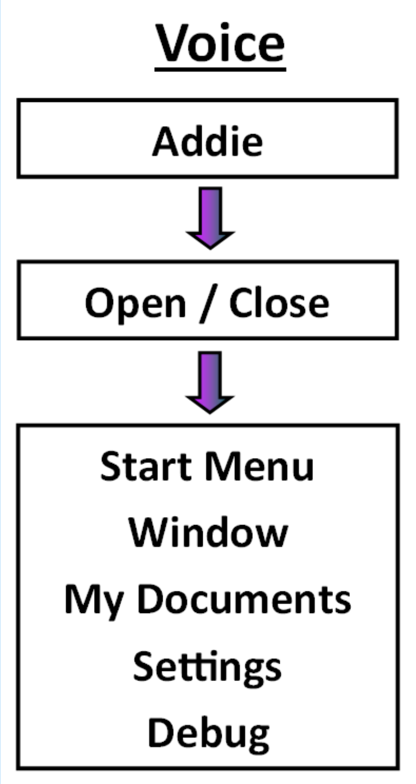




Kinect Audio Stream

- Used for voice commands





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

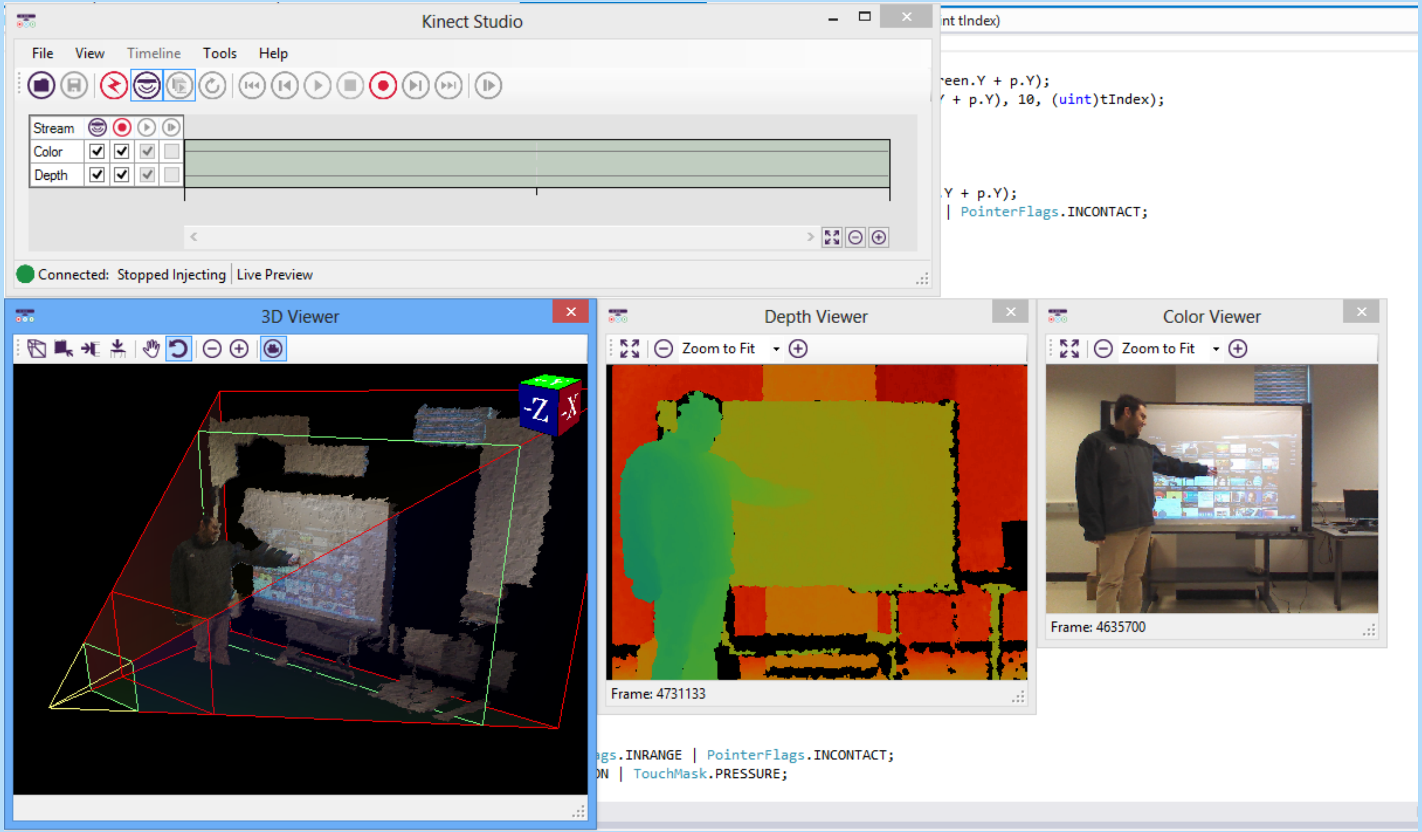
Programming Environment

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

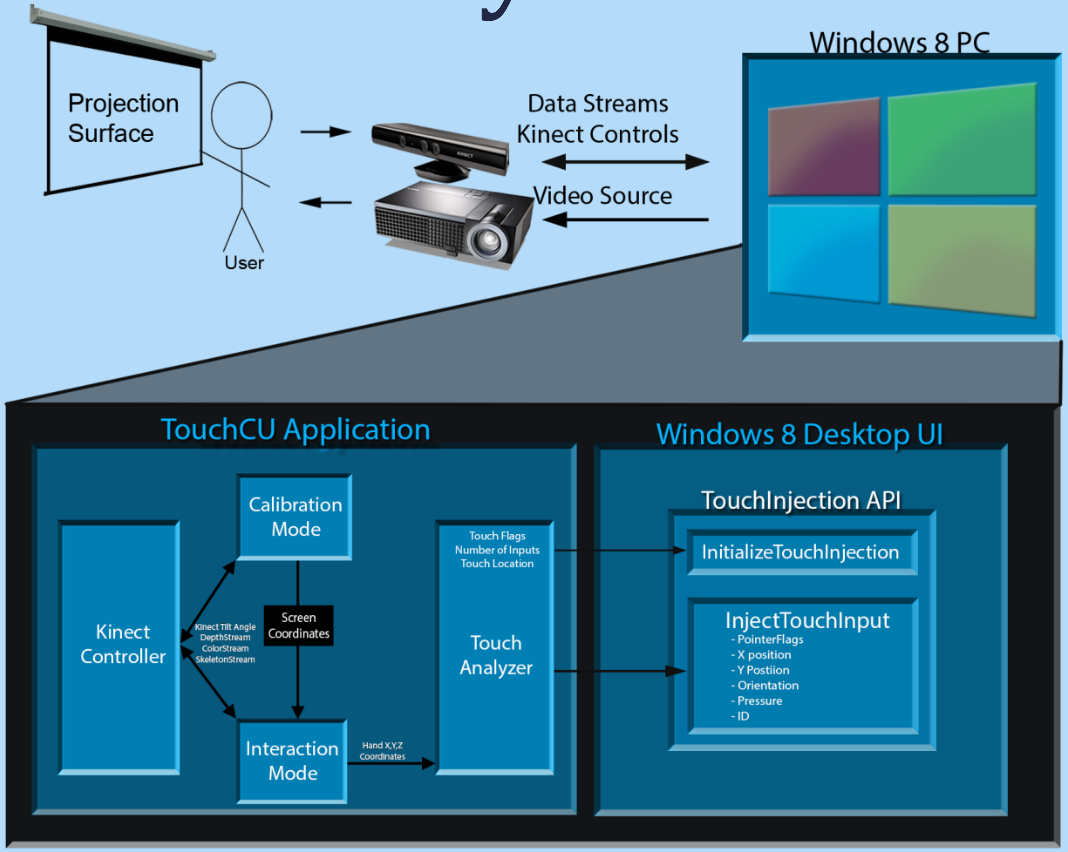
Other Software

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

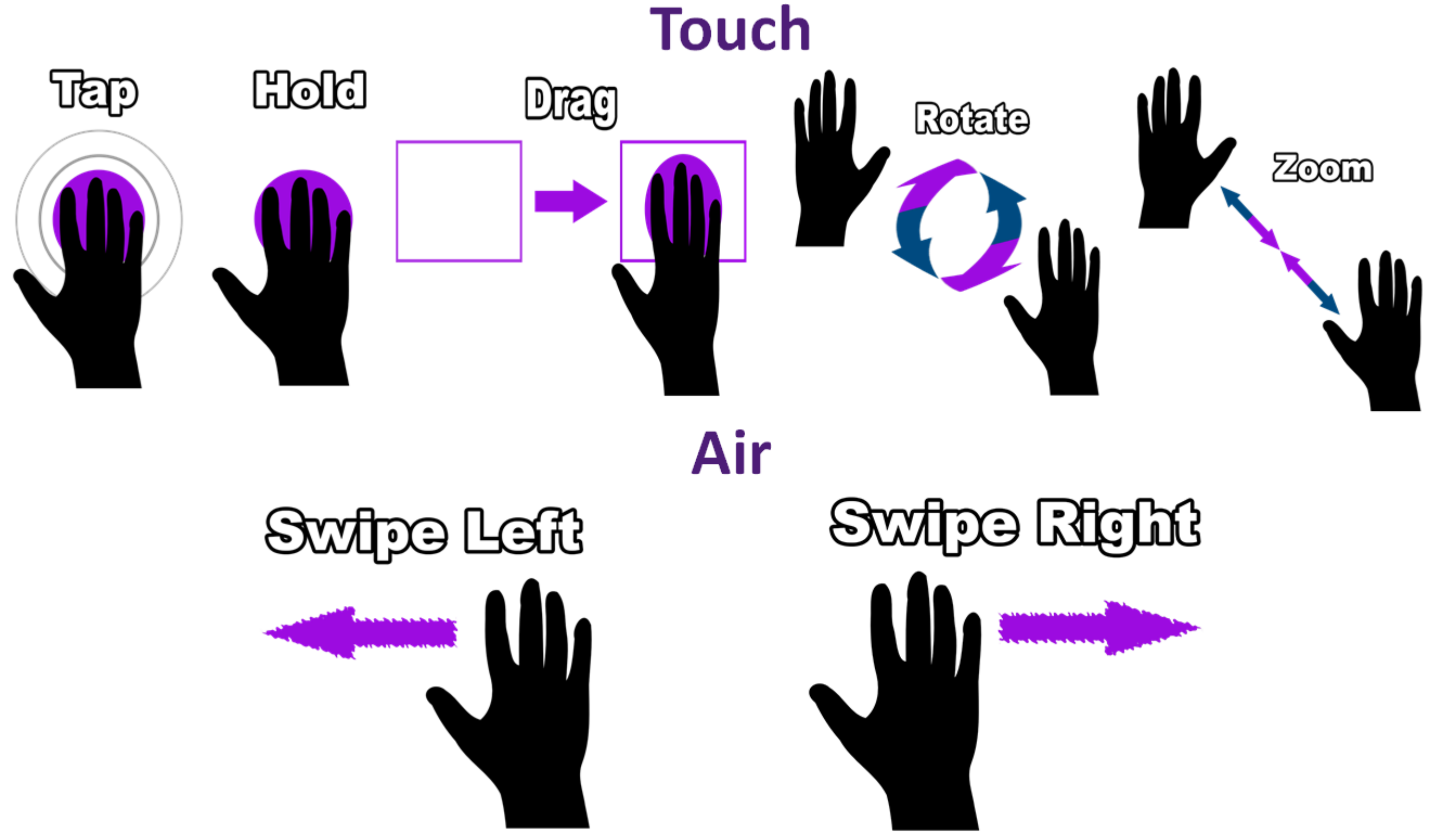
Kinect Studio



System Architecture

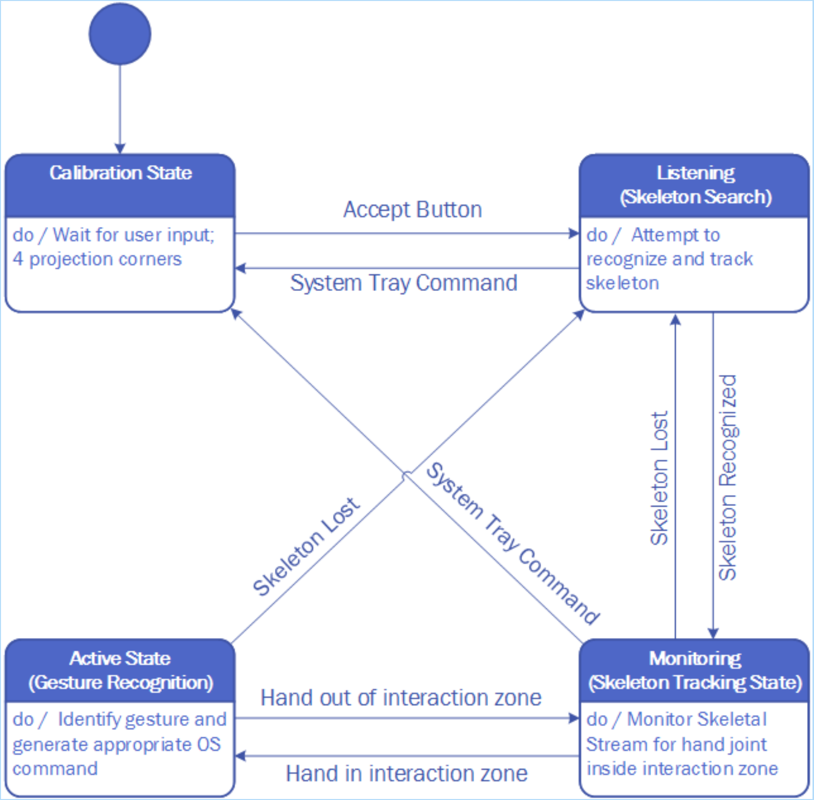


Gestures



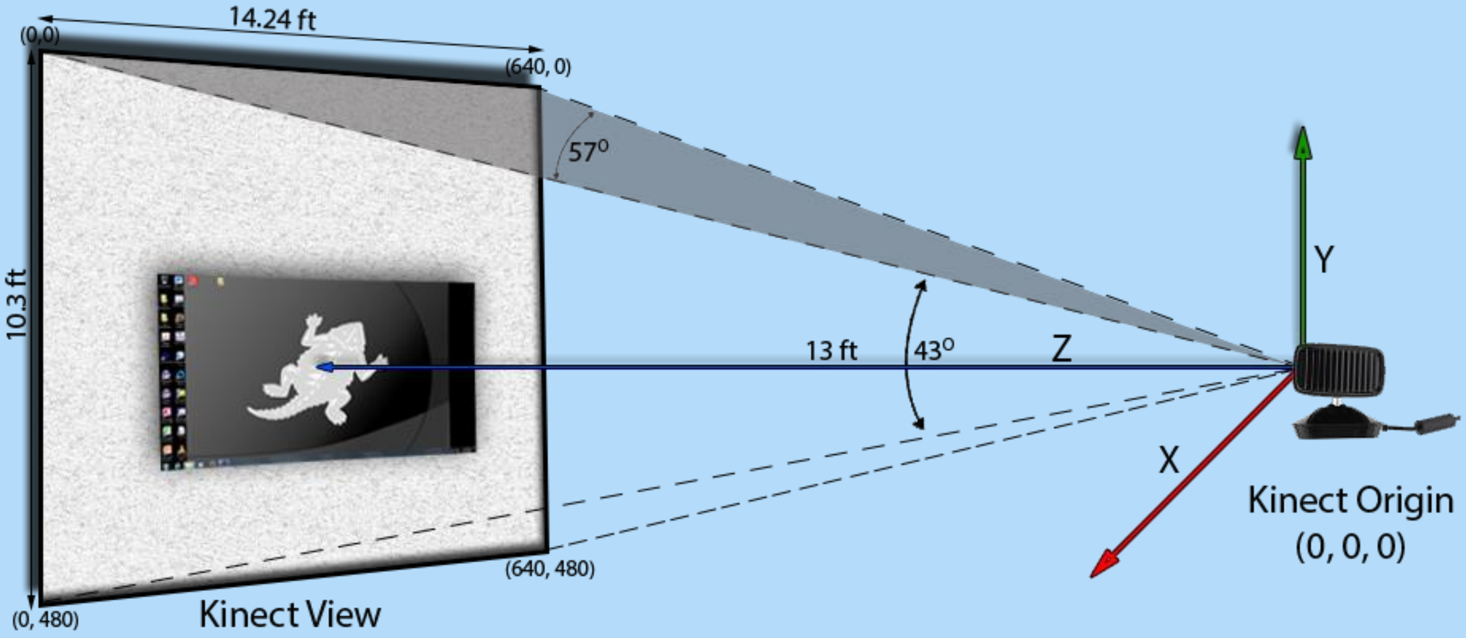
Application States

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

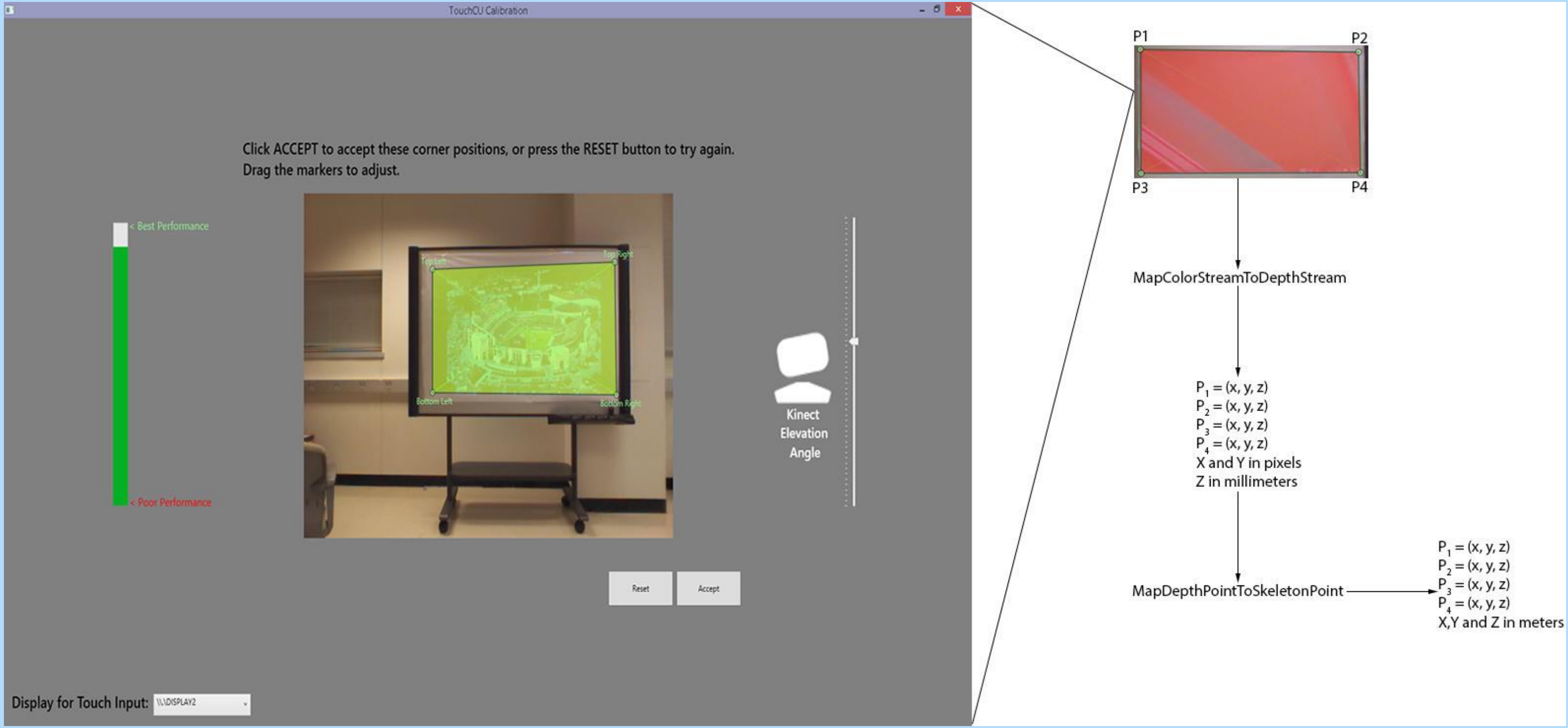


Using TouchCU

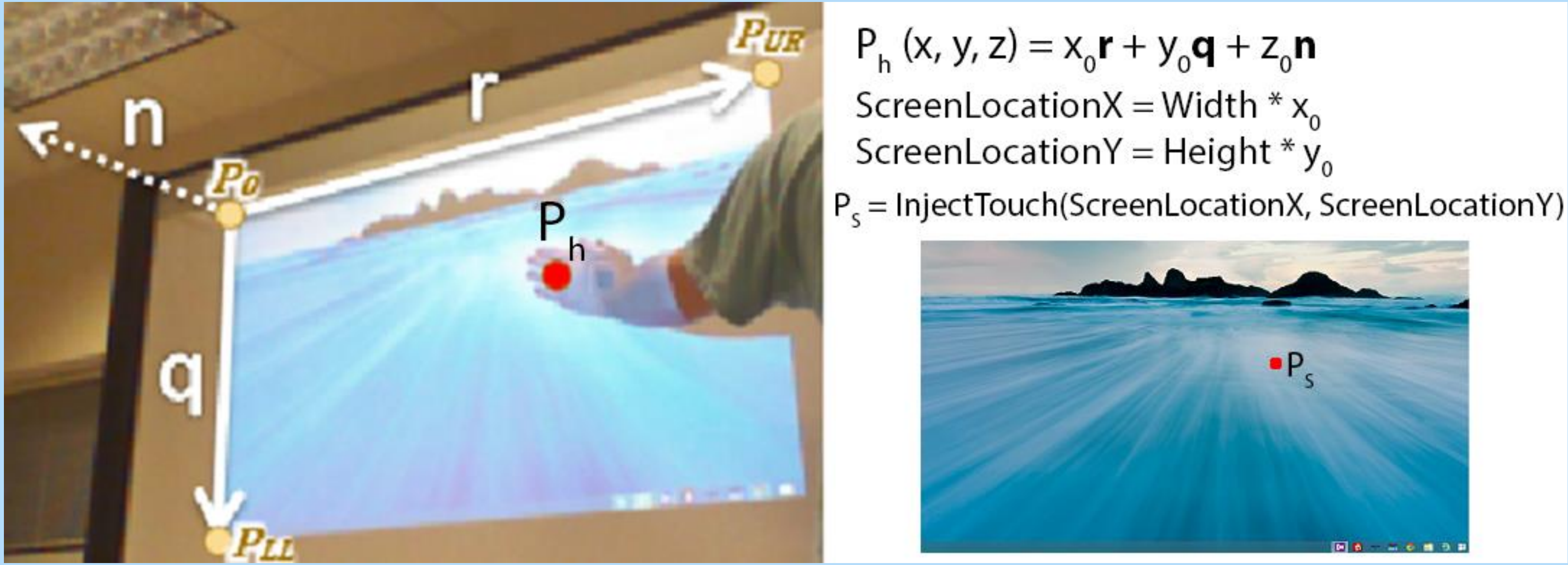
Step 1: Setup


14.24 ft
(0,0)
(640, 0)
57°
10.3 ft
13 ft
43°
Z
Y
X
(640, 480)
(0, 480)
Kinect View
Kinect Origin
(0, 0, 0)

Step 2: Calibration



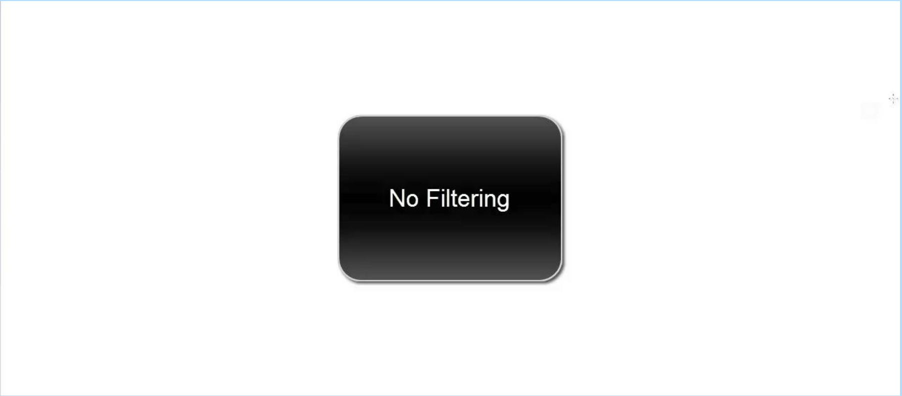
Step 3: Determine Hand Position on Projection Screen



Data Filtering

Built-in Kinect filtering parameters

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

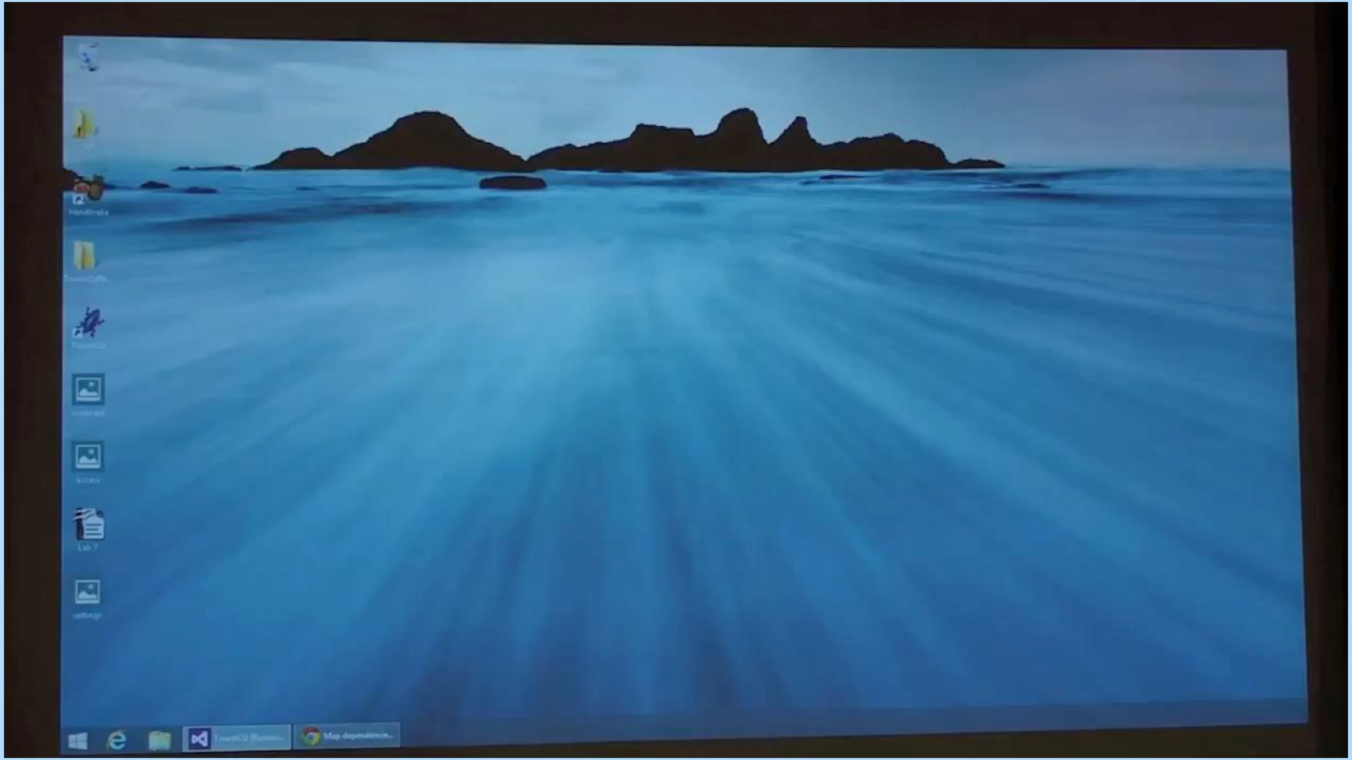


- Based on the Holt Double Exponential Smoothing method

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?