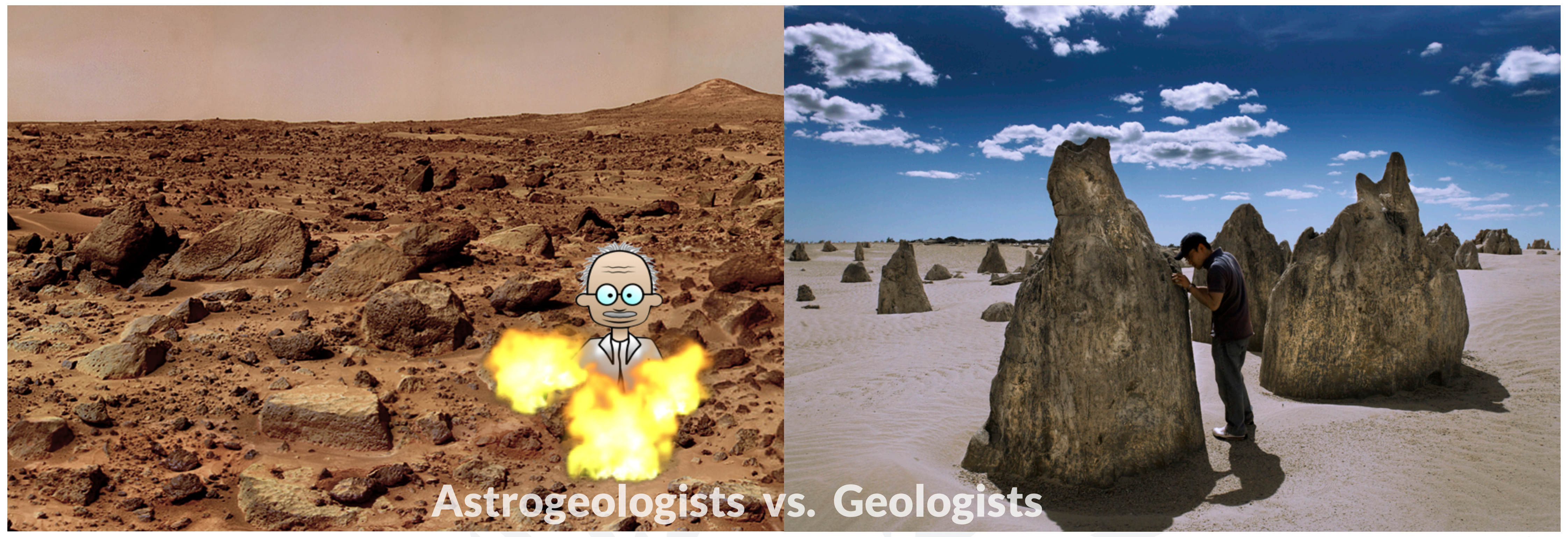
UCSB Computer Science Capstone 2013 The Hex Pistols

Kinects in Unity

TEAM MEMBERS Boyang **Peng** LEAD Sea **Pong** Alexander **Scarlett** Anthony **Narsi** Kevin **Sheridan**

SPECIAL THANKS TO
Chandra **Krintz** UCSB
Tim **Sherwood** UCSB
Janet **Kayfetz** UCSB
Victor **Luo** JPL
Garrett **Johnson** JPL

the Problem

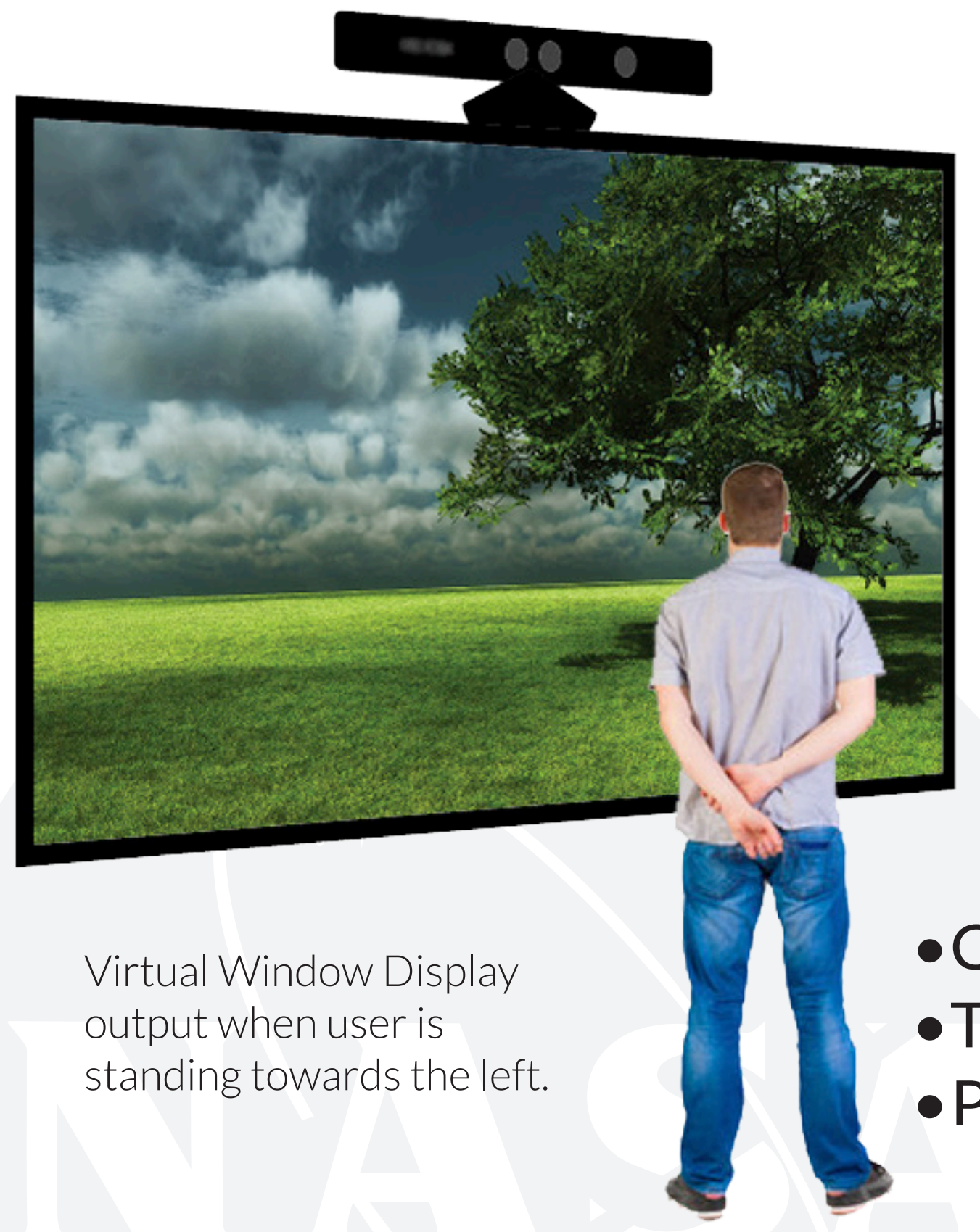


- Field scientists need to be able to safely and conveniently interact with the environments they study.
- Unlike earth geologists, it is not easy for planetary geologists to travel beyond our planet to study other celestial bodies.
- The Varrier Autostereoscopic Virtual Reality Display is an example of a virtual reality system that displays a first-person motion-tracked perspective via multiple displays.
- JPL currently has an equivalent infrastructure working, but employs extremely expensive equipment and bulky devices.
- Our goal was to develop an ***inexpensive*** and ***intuitive*** way to present a virtual window perspective of a 3D environment.

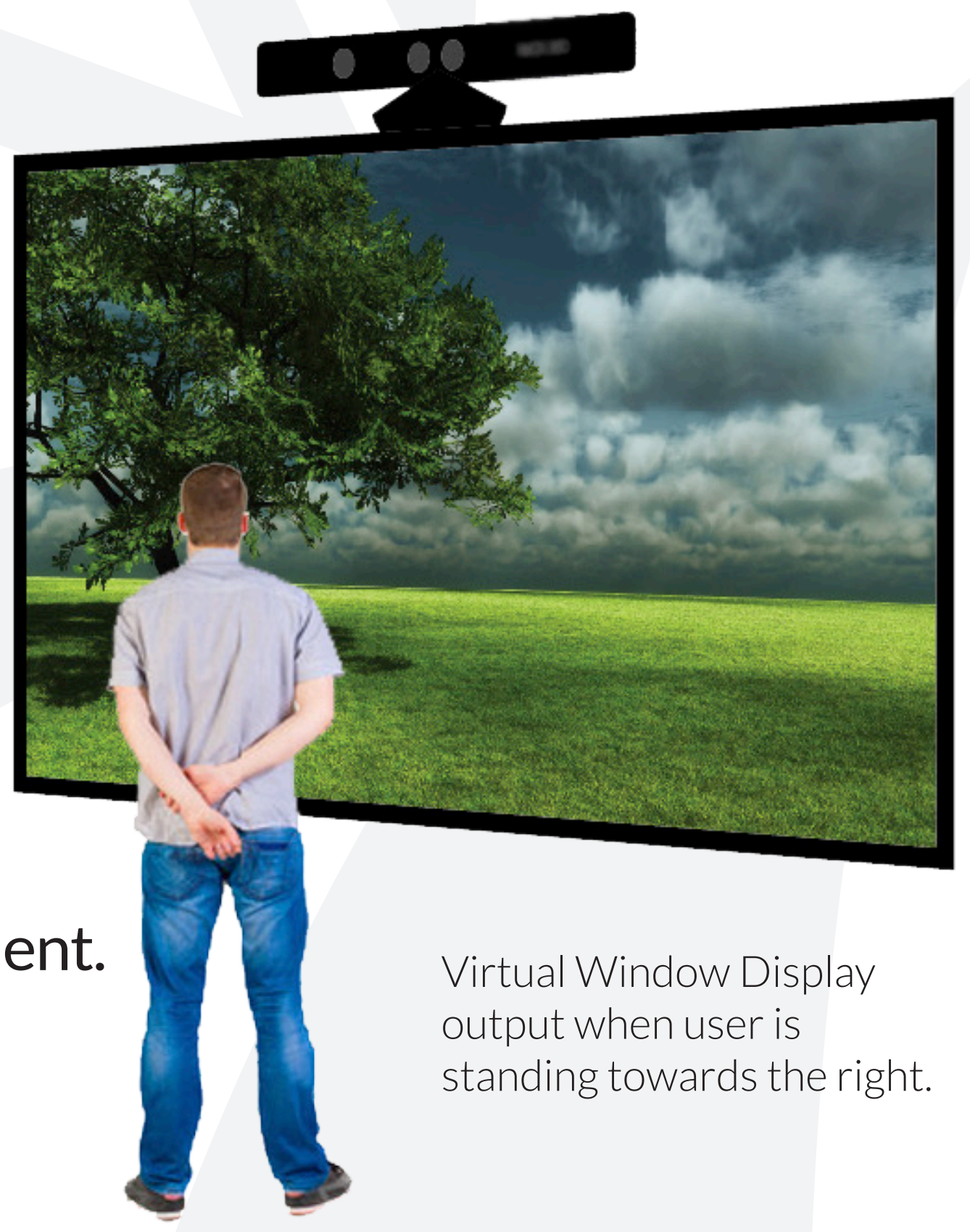
the Team

the Team (from left to right): Anthony Narsi, Sea Pong, Kevin Sheridan, Boyang (Jerry) Peng, Alex Scarlett

our Solution

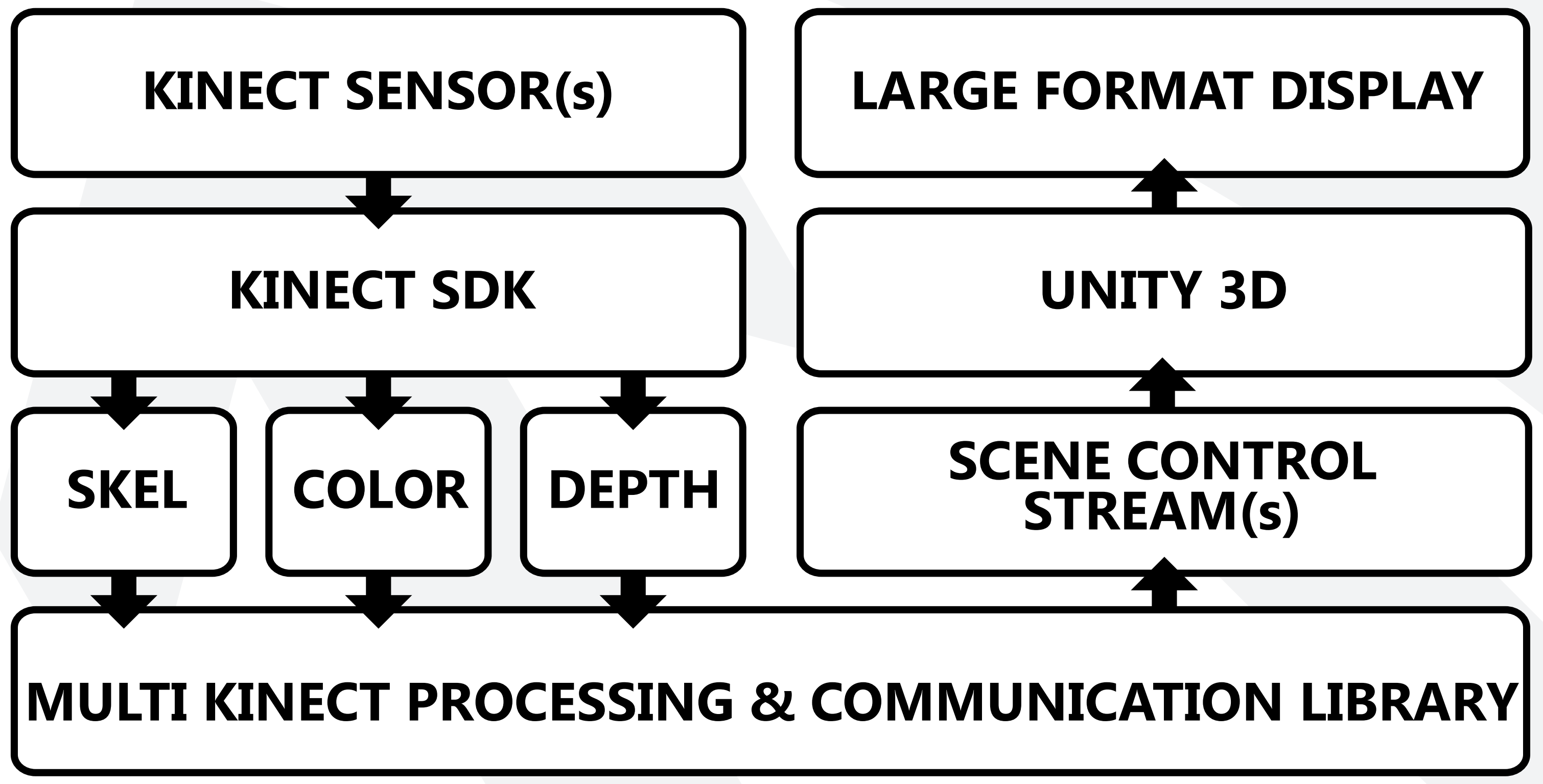


Virtual Window Display output when user is standing towards the left.



Virtual Window Display output when user is standing towards the right.

- Create a system that transforms displays into virtual windows that look onto a virtual enviornment.
- Take advantage of the Microsoft Kinect and it's included Software Development Kit (SDK).
- Provide a self-calibrating system that allows variable placement of displays (virtual windows).

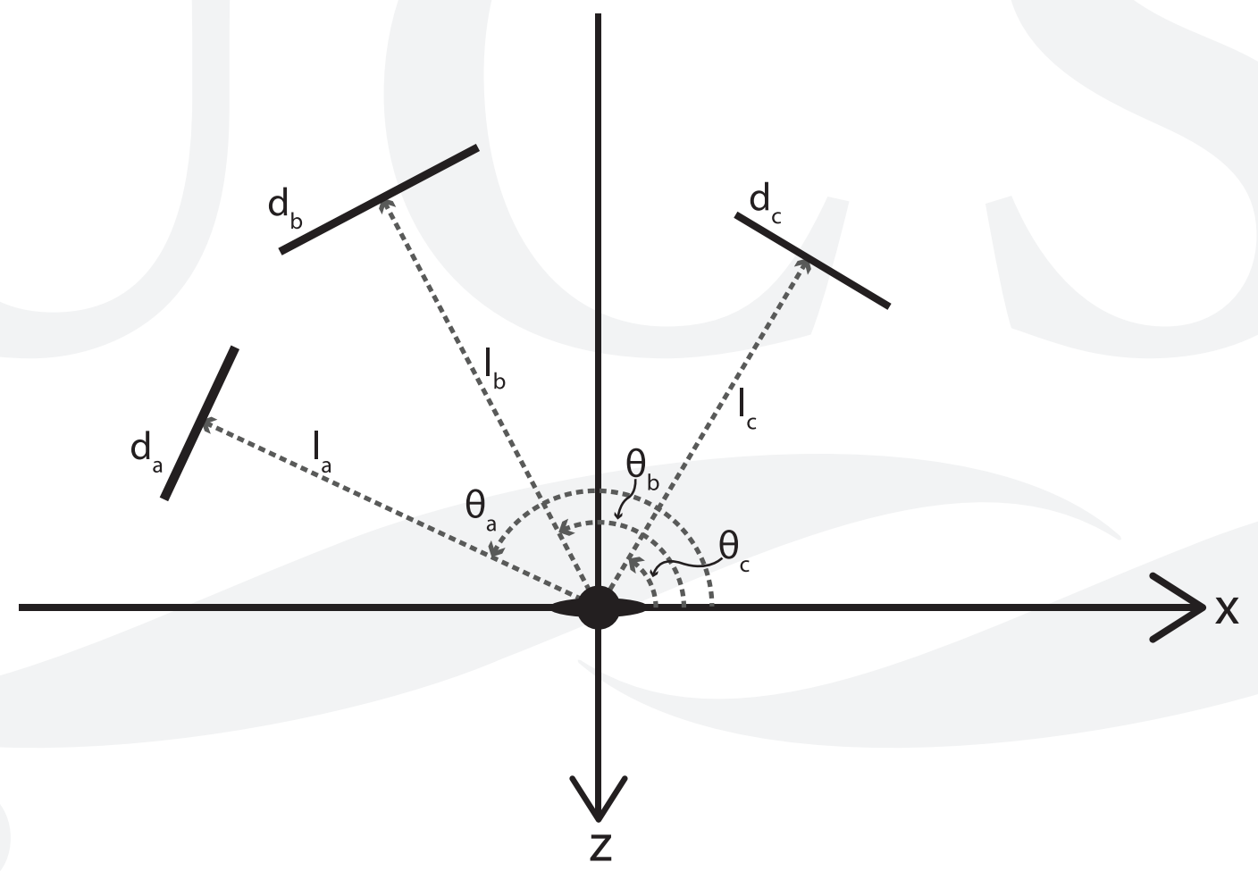

KINECT SENSOR(s)
KINECT SDK
SKEL
COLOR
DEPTH
LARGE FORMAT DISPLAY
UNITY 3D
SCENE CONTROL STREAM(s)
MULTI KINECT PROCESSING & COMMUNICATION LIBRARY

Data Flow Diagram

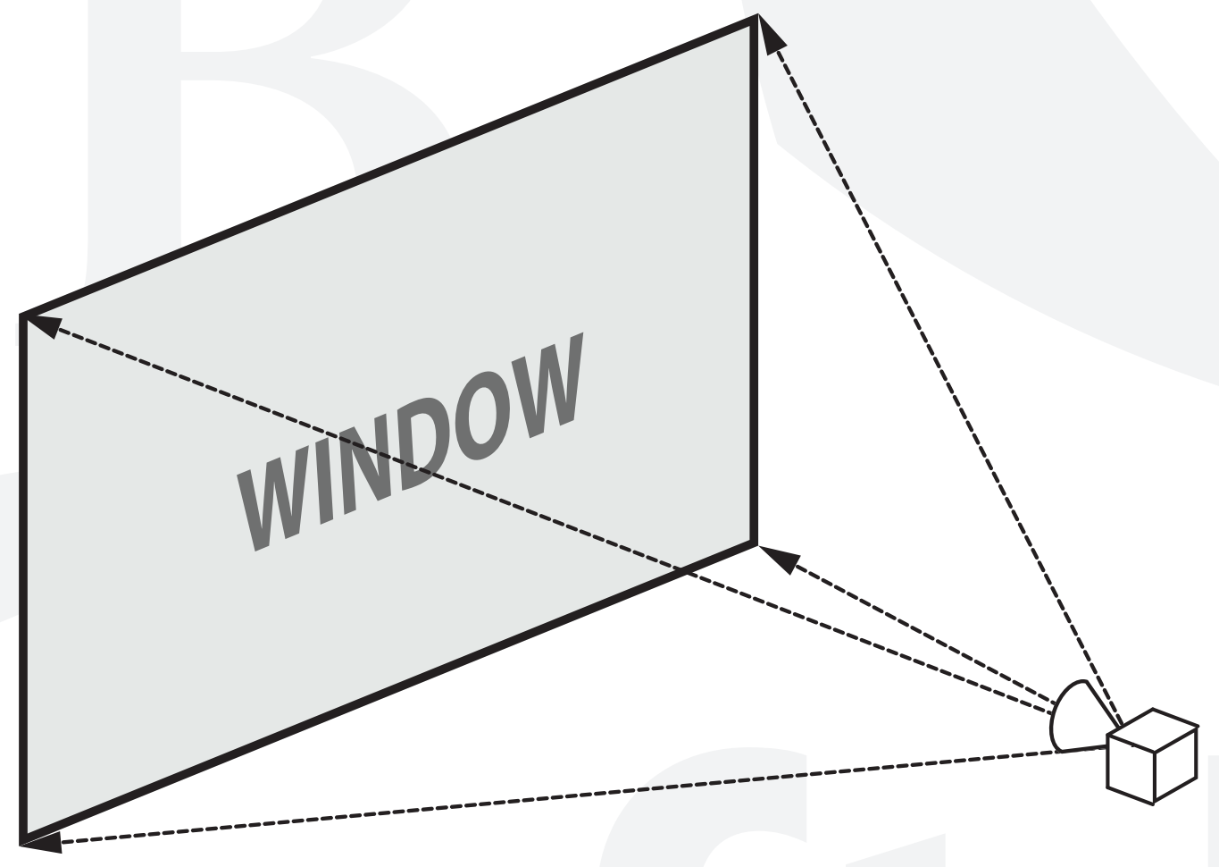
the Technology

the Data Flow Process

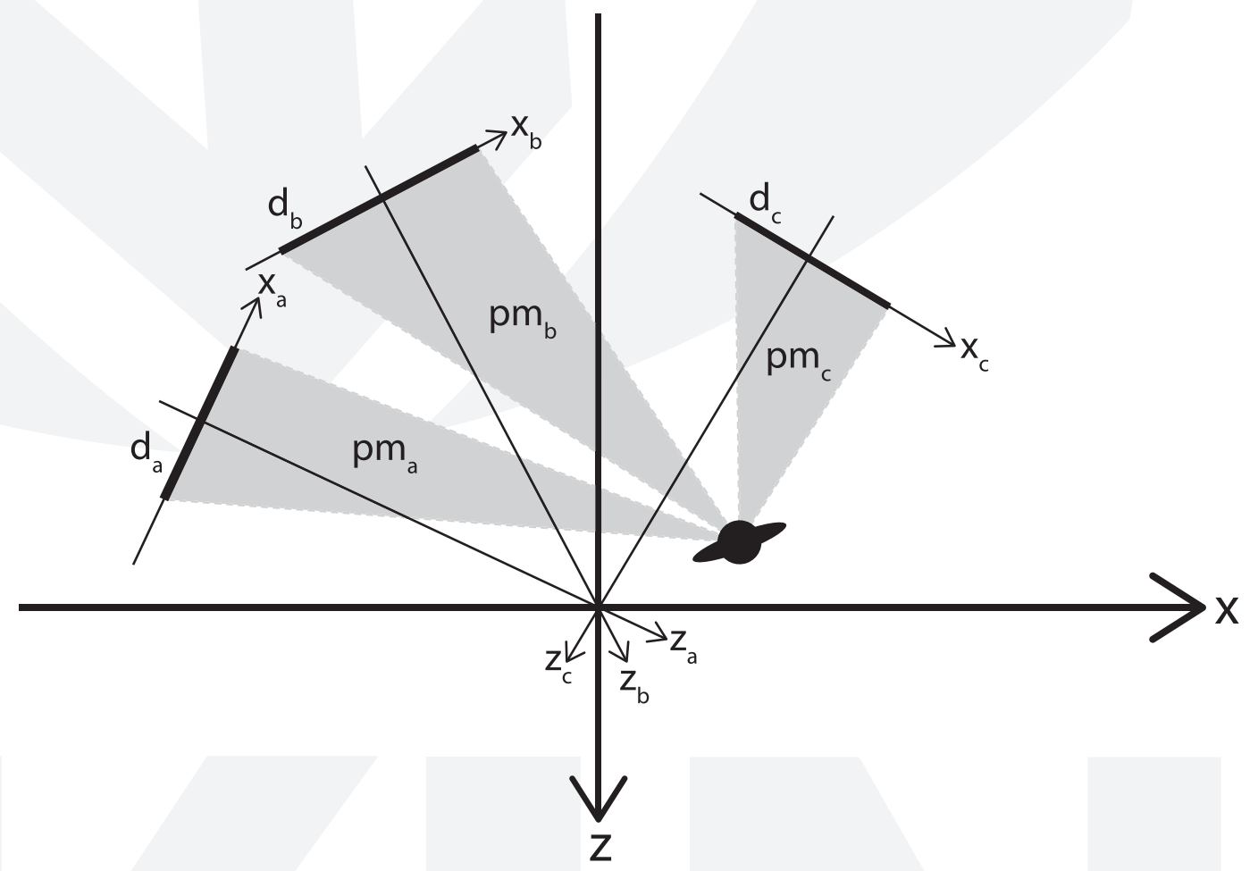
1. Using skeletal, depth, and color (S/D/C) data from the Kinect SDK, recognize and establish the placement of the screens and user in a unified coordinate system (UCS).
2. Establish communication through network interfaces (between the Kinect processor machine and each Unity 3D display rendering machine) to promote scalability of the VR system.
3. Load placement and user position data based on the UCS into our custom Unity3D game rendering engine scripts.
4. Using linear algebra (matrices and vectors), calculate projection matrix data set for each display at each frame.
5. Render the virtual window display as each S/D/C frame from the Kinect is updated (at 30 frames/second).



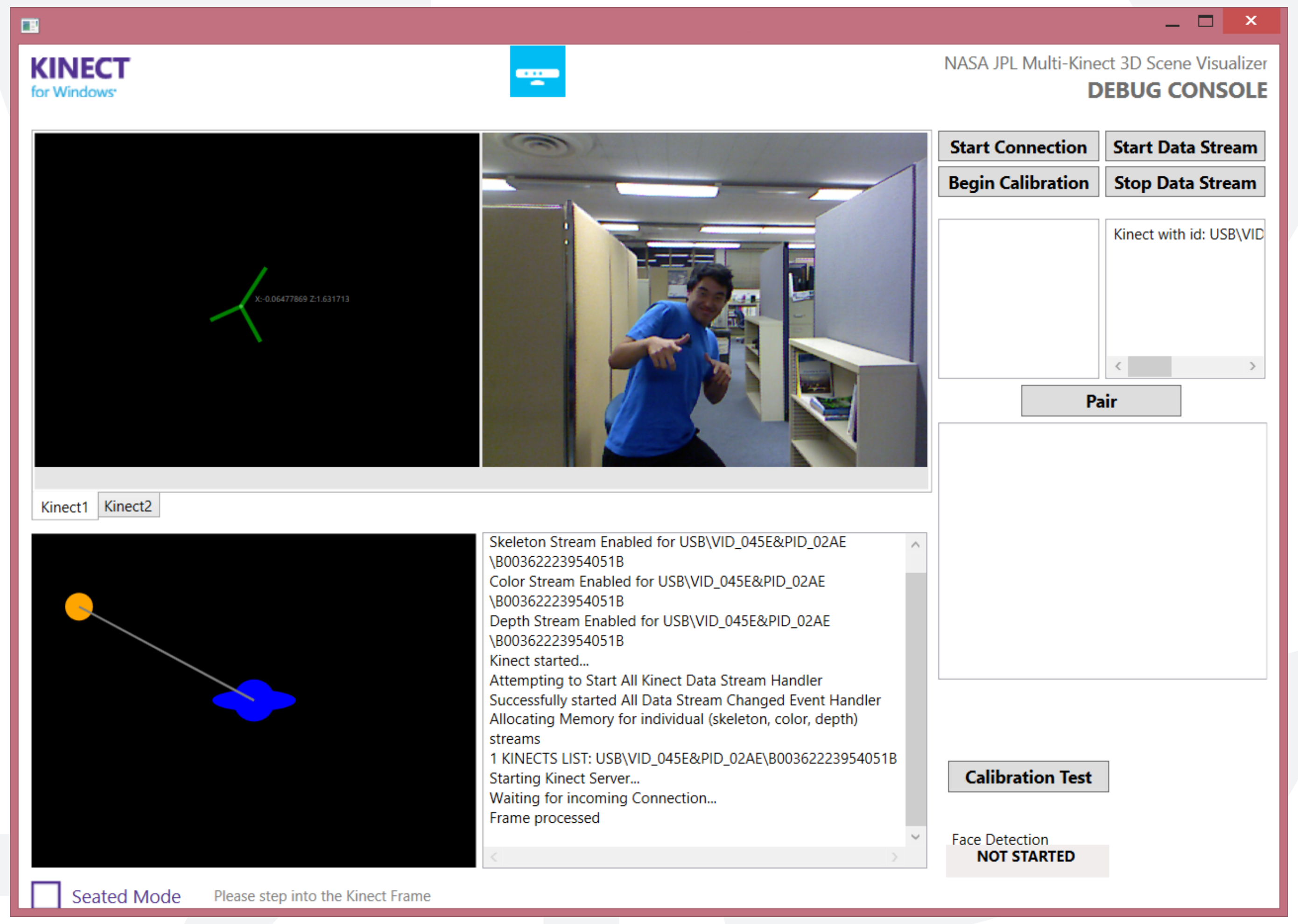
A top view of the unified coordinate system (UCS) during display calibration and placement



A projection field in vector representation



A top view of the unified coordinate system (UCS) and it's sub coordinate systems (SCS) during tracking mode



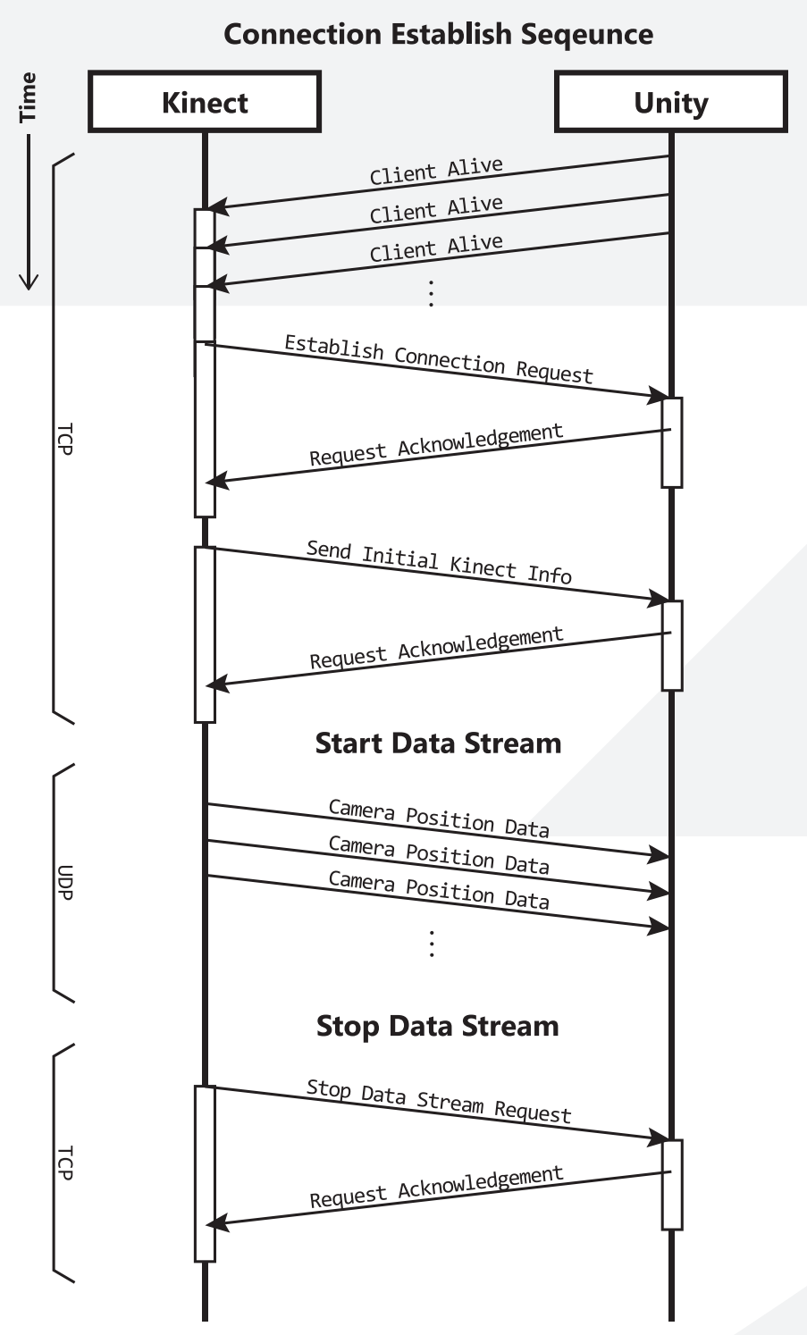
Kinect with id: USB\VID

```
Skeleton Stream Enabled for USB\VID_045E&PID_02AE
\B00362223954051B
Color Stream Enabled for USB\VID_045E&PID_02AE
\B00362223954051B
Depth Stream Enabled for USB\VID_045E&PID_02AE
\B00362223954051B
Kinect started...
Attempting to Start All Kinect Data Stream Handler
Successfully started All Data Stream Changed Event Handler
Allocating Memory for individual (skeleton, color, depth)
streams
1 KINECTS LIST: USB\VID_045E&PID_02AE\B00362223954051B
Starting Kinect Server...
Waiting for incoming Connection...
Frame processed
```

Face Detection NOT STARTED

Please step into the Kinect Frame

The Kinect Processor Control Panel



Network Time Seqeunce Diagram