Virtual Room Optical Markerless Gesture Interface

Mentor: Todd Margolis

Group B
Jose Castillo
Jordan Clarno
Samuel Ha
Kevin Pham

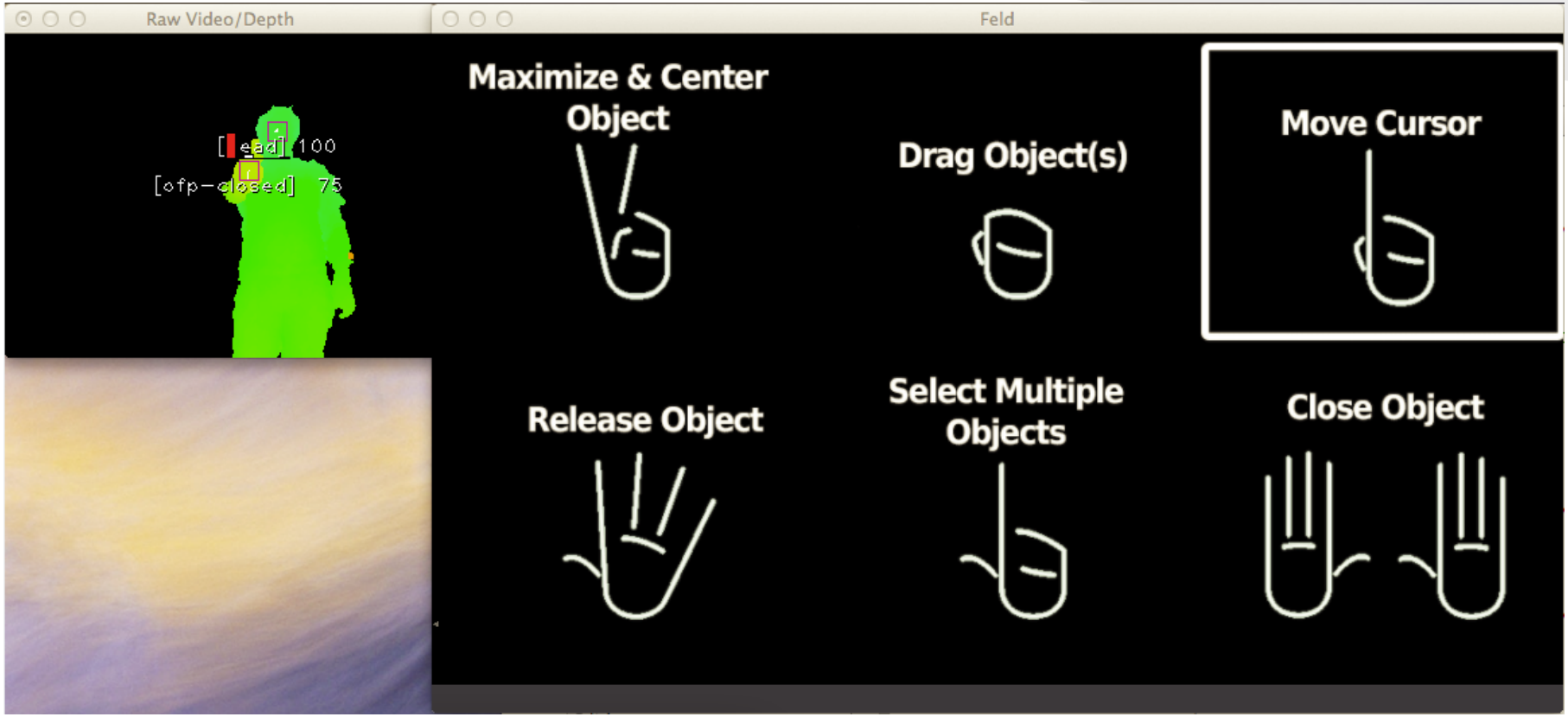
Overview

- Goal:
  - To create a user interface based on simple hand gestures that can manipulate various media objects on a large tiled display wall.
  - The interface uses a relatively inexpensive and commercially-available IR depth camera, similar to Microsoft Kinect.

Demo

http://www.youtube.com/watch?v=ugCn7LS2qFo&feature=youtu.be

Gesture Recognition



What the user will see when they are performing gestures in front of the sensor.

Motivation

- Existing user interfaces may require the user to hold a device and confine the user to a small area

- This project has:
  - Intuitive gestures instead of abstract buttons
  - No motion tracking hardware held by user
  - Ability to control software from large area

Specifications

- ASUS Xtion motion sensor
- Oblong's Greenhouse (coding tool kit for spatial references)
  - Gesture recognition code (C)
  - Client server for socket communication (Ruby)
- Scalable Adaptive Graphics Environment [SAGE] platform for graphical object interaction
  - Hardware Capture and Device plugins (Python)

High-Level Objectives

- Support 6 gestures that perform various functions
  - Each gesture may support multiple functions (i. e., performing a left click and dragging objects)
- Uses hand gestures that are natural, intuitive for new users to learn, and easy to perform

Gesture Capture Diagram

Image Processing

IR Dot Pattern

Reflected Light to Sensor

Scene

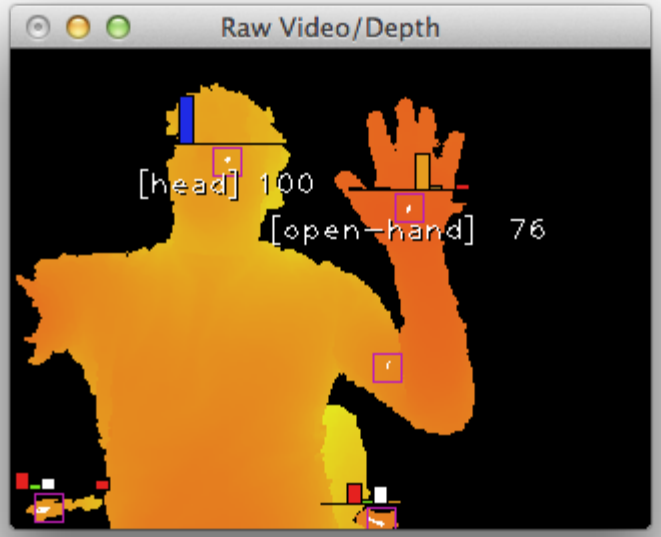
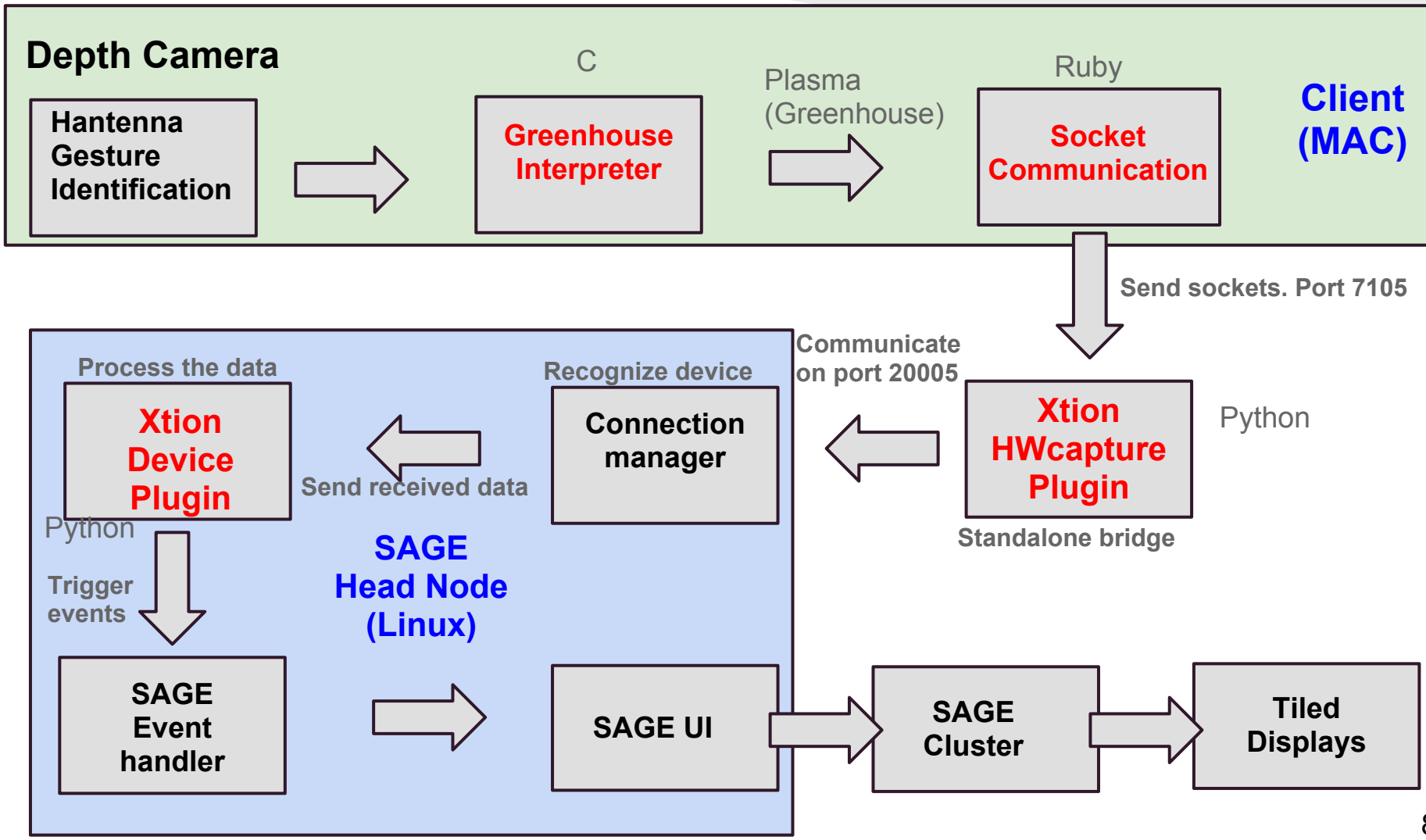


Image courtesy of http://geeksrespawn.com/see-how-kinect-works-night-vision-camera/

Design Block Diagram

RED = Our Project



What is Greenhouse?

- Designed to manipulate data (images, tweets, etc.) in 3D space, using various interfaces including depth camera
- Can communicate with other Greenhouse applications natively using "Plasma" protocol
- Has libraries for C and Ruby, though documentation/features vary for either language

Greenhouse Interpreter

- Receives raw data from Greenhouse:
  - All Gesture types
  - Absolute Position of each Gesture
- Filters out unwanted gestures
- Calculates displacement of gestures from initial location
- Records data in a shared location

Server Program

- Listens for data from Interpreter program
- Converts data from Greenhouse proprietary format to floating point values
- Transmits gesture identifier and location over network socket to SAGE

What is SAGE?

- SAGE = Scalable Adaptive Graphics Environment
- It is a graphical interface that is distributed across many screens
- It allows the user to manipulate various media objects (videos, images, etc.)

Hardware Capture Plugin

- Establish bridge connection between Client side and the SAGE head node.
- Pre-process data, pass received data and device identification to Connection Manager.

Device Plugin

- Perform various calculations on received data. (i.e., displacement, acceleration)
- Decide which and how the events will be triggered.
- Call different event handlers to perform desired function.

Gestures Interface Demo

http://www.youtube.com/watch?v=OdHIrf0Jrg4&feature=youtu.be

Accomplishments

- Able to support all six gestures with low latency
- Perform the essential functions of SAGE
- Reduce the jitter of the pointer movement (required filtering on both Greenhouse and SAGE ends)
- Simplify socket communication code (100+ lines in C vs. roughly 10 lines in Ruby)
- Pointer color changes in response to the gesture performed, for user feedback

Observations

- Two-handed gesture recognition is still in development by Oblong Industries

- Use of multiple sensors, tracking at different angles, may improve gesture recognition and user experience

Summary

- Our gestural user interface is functional and supports interaction with SAGE
- Recognition is almost perfectly synchronized with hand movement
- Any questions?