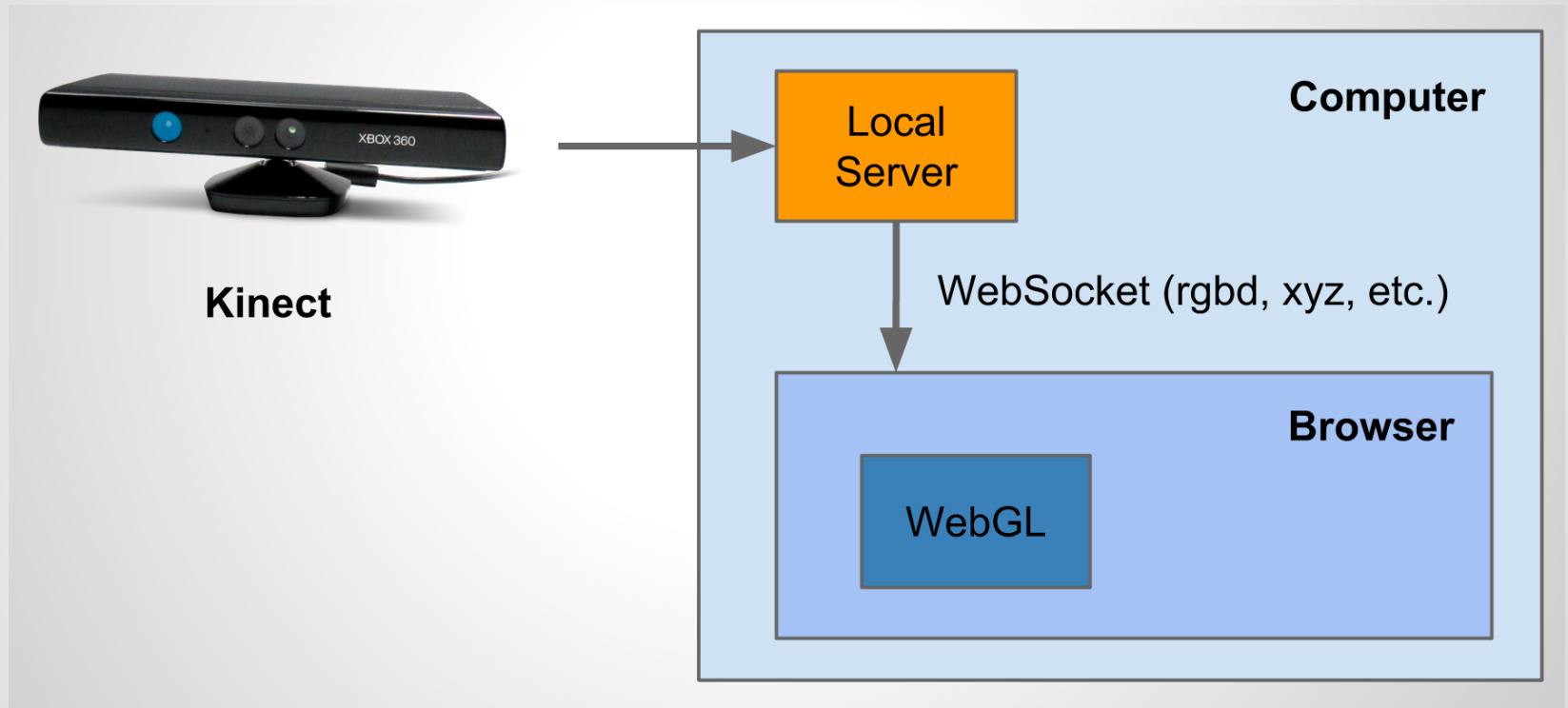


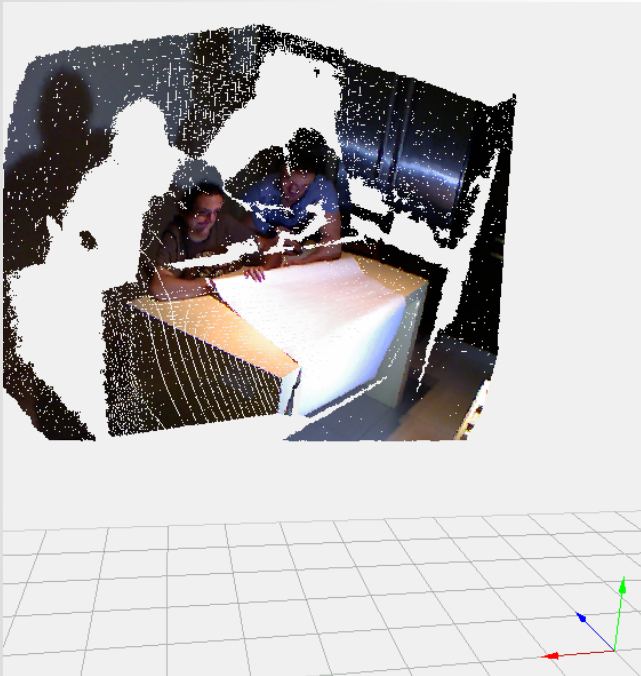
WebGL Realtime 3D Point Cloud Visualization

Takashi Furuya
Qiong Wang

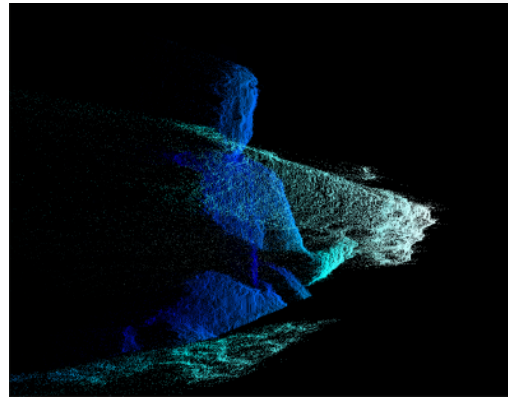
Recap



Related Work

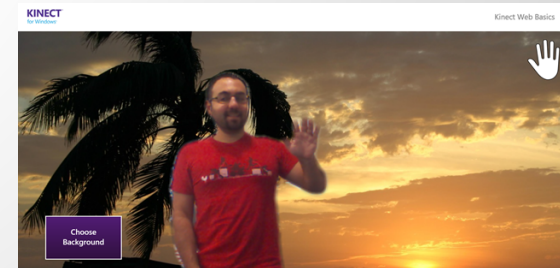


Mozilla: OpenDepth



Mr.doob
Chrome Experiments

Kinect SDK



Implemented on Linux & Windows

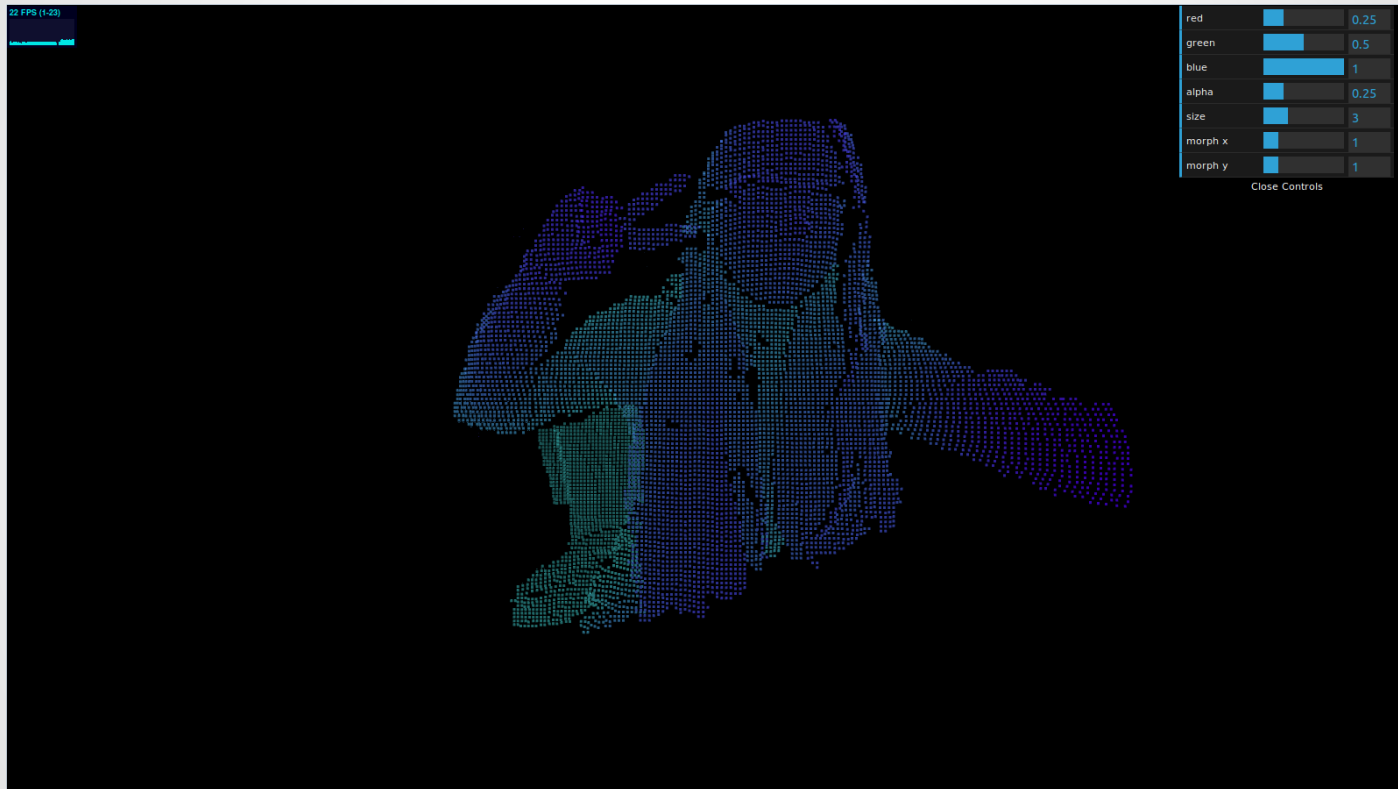
Linux

- Python server
- libfreenect
- Downsampling & compression
- Demo

Windows

- C# server
(Alchemy
WebSocket
library)
- Kinect SDK
- Performance
analysis

Demo



Performance Analysis

→ Want high res, real-time Kinect data.

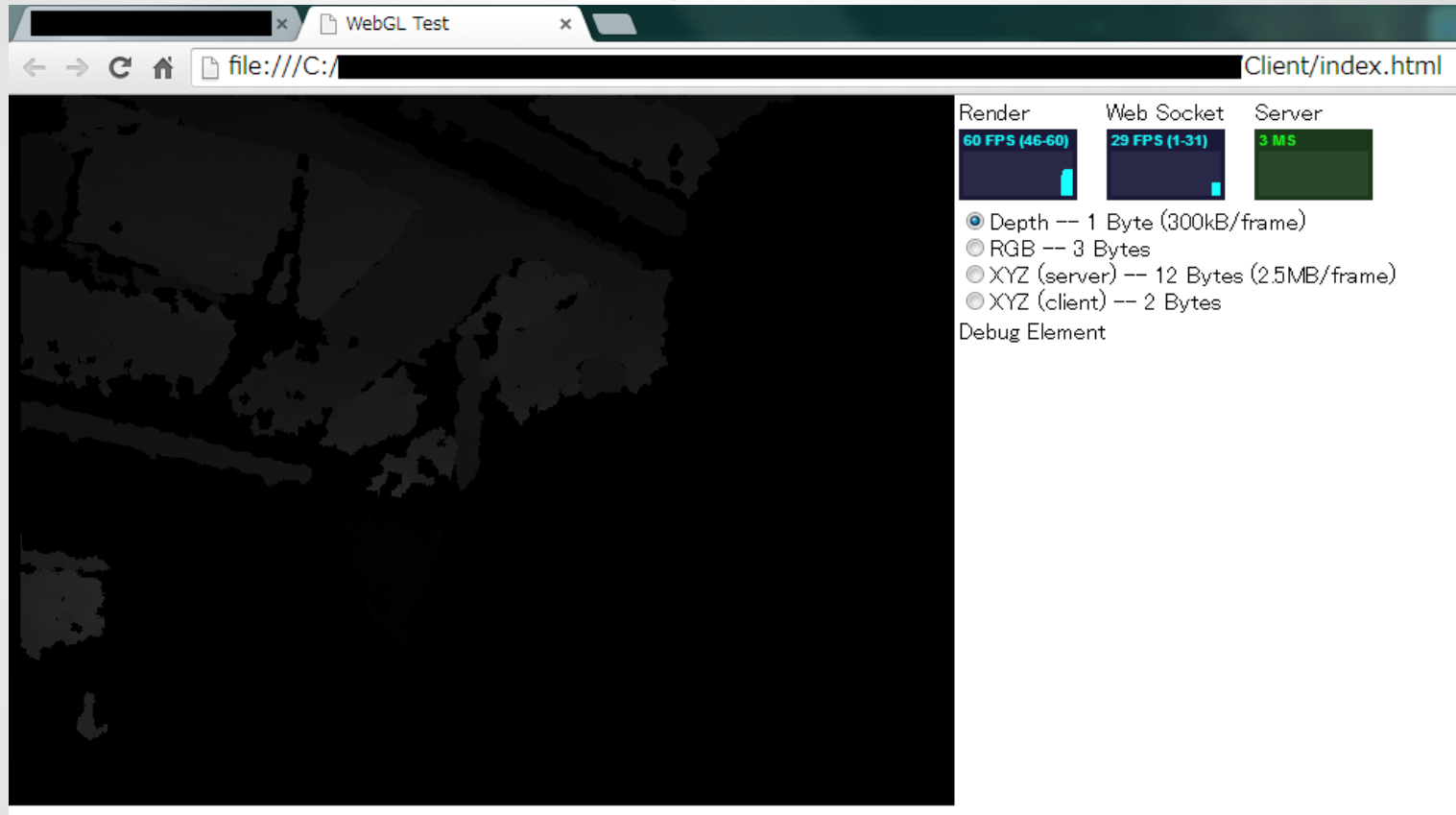
- Use little GL calls
- Efficient buffer update
- No loops or manual copy/moving of data

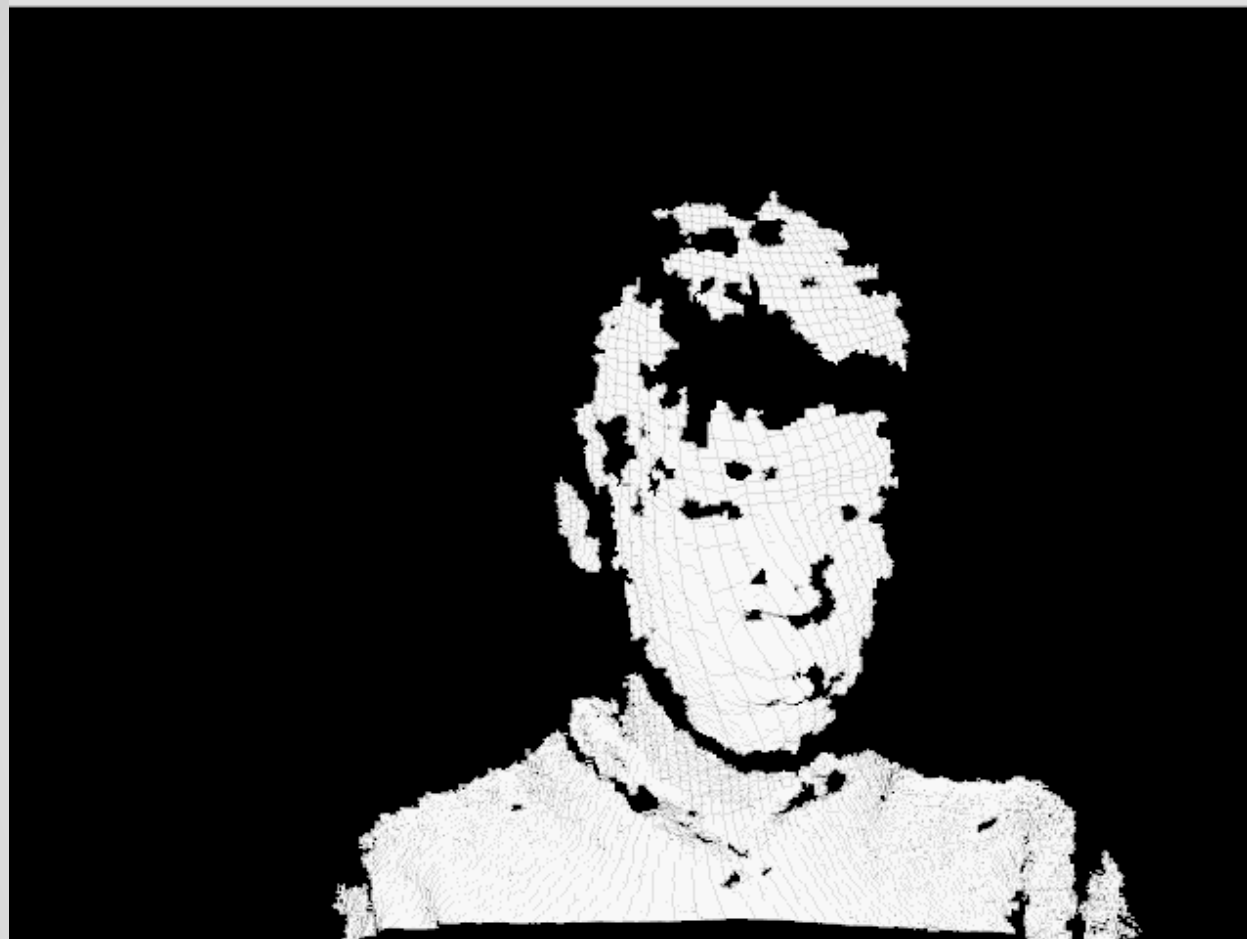
→ Code from scratch without relying on Three.js

Performance Analysis

1. Send 8 bit depth, render depth
2. Send 24 bit color, render color
3. Send 32x3 bit position, render point cloud
4. Send 16 bit depth, render point cloud

Performance Analysis Demo





Render

60 FPS (30-60)



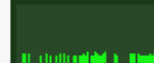
Web Socket

30 FPS (1-32)



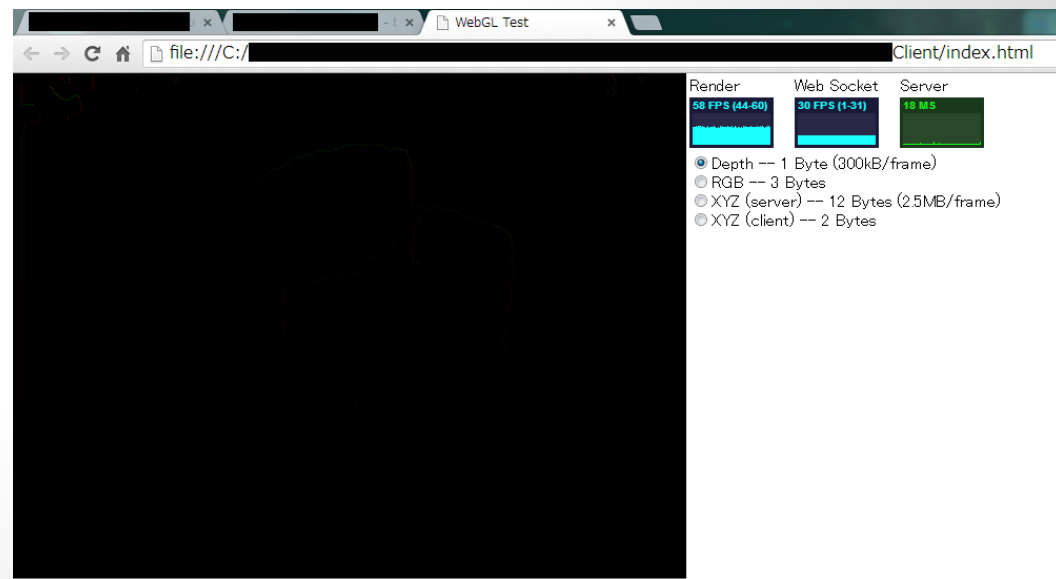
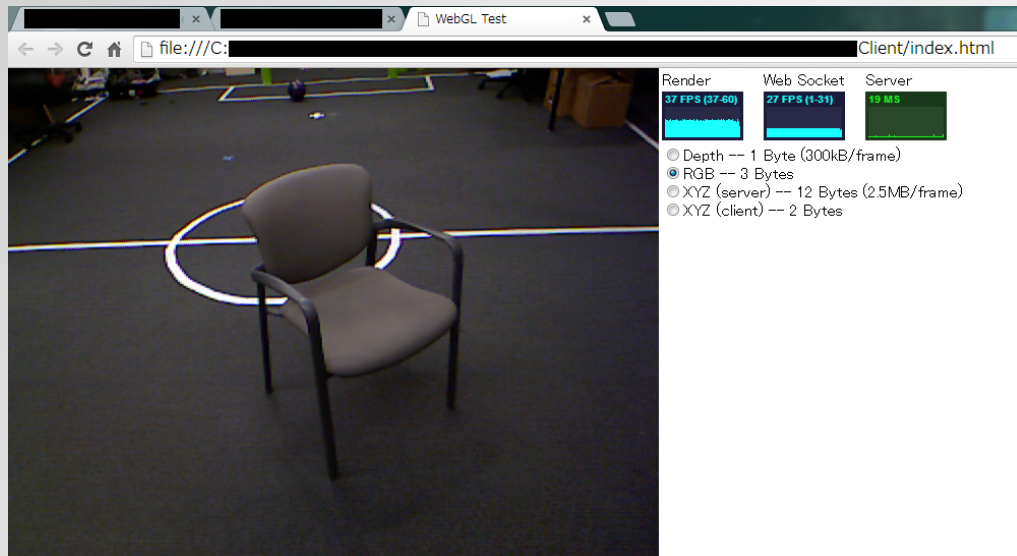
Server

0 MS



- ☐ Depth -- 1 Byte (300kB/frame)
- ☐ RGB -- 3 Bytes
- ☐ XYZ (server) -- 12 Bytes (2.5MB/frame)
- ☒ XYZ (client) -- 2 Bytes

Debug Element



Future Work

- Crash prevention using polling or web worker
- Investigate effectiveness of JavaScript decompression.
- Point cloud with RGB from camera.

Contacts

Takashi Furuya: tfu@seas.upenn.edu

Qiong Wang: qiong@seas.upenn.edu

GitHub

<https://github.com/GabriellaQiong/Kinect-based-Open-Source-WebGL-Library>