

# **WebGL Shader Editor with Kinect Data**

CIS565 Final Project  
Alpha Presentation

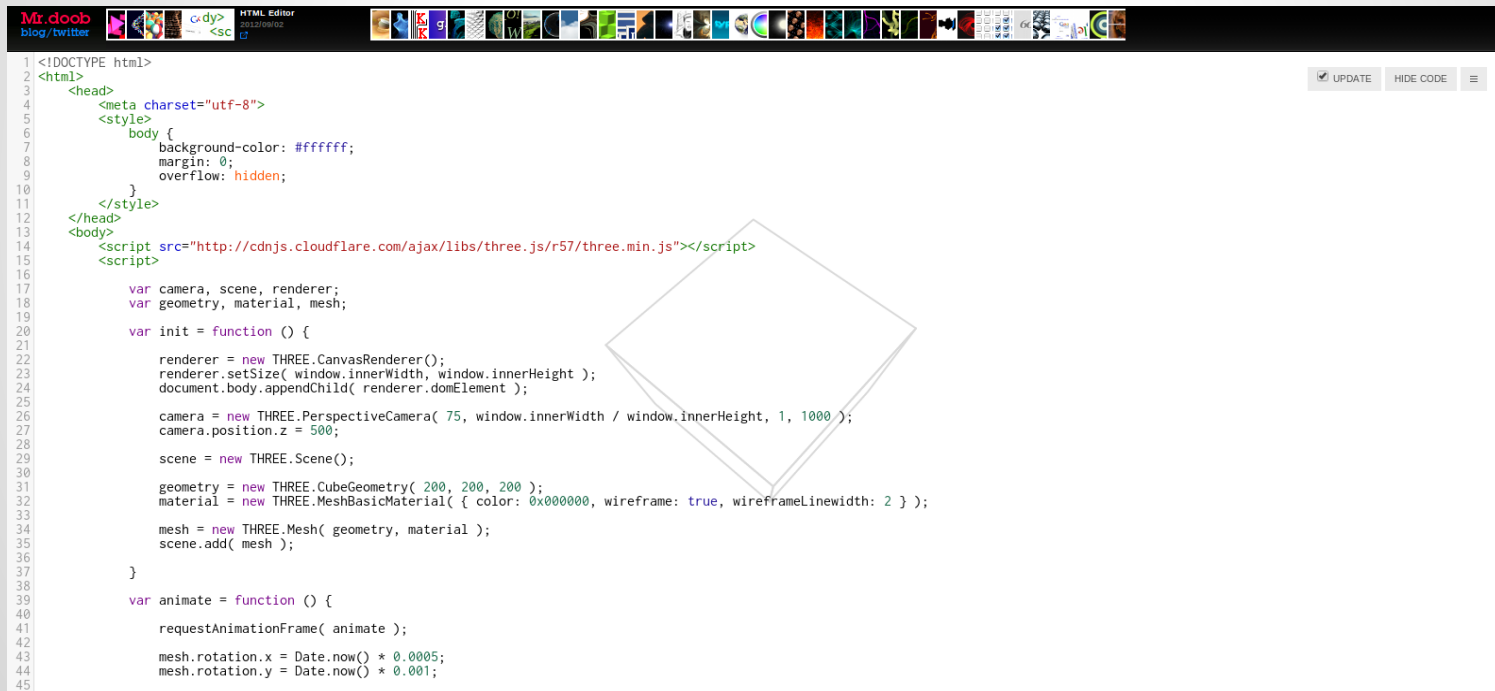
Takashi Furuya, Qiong Wang

# Objective

- Create an online shader editor (like Shader Toy) with input from Kinect.
- Create 3 demos

# Demo 1

Vertex morphing: [http://www.mrdoob.com/#/146/html\\_editor](http://www.mrdoob.com/#/146/html_editor)

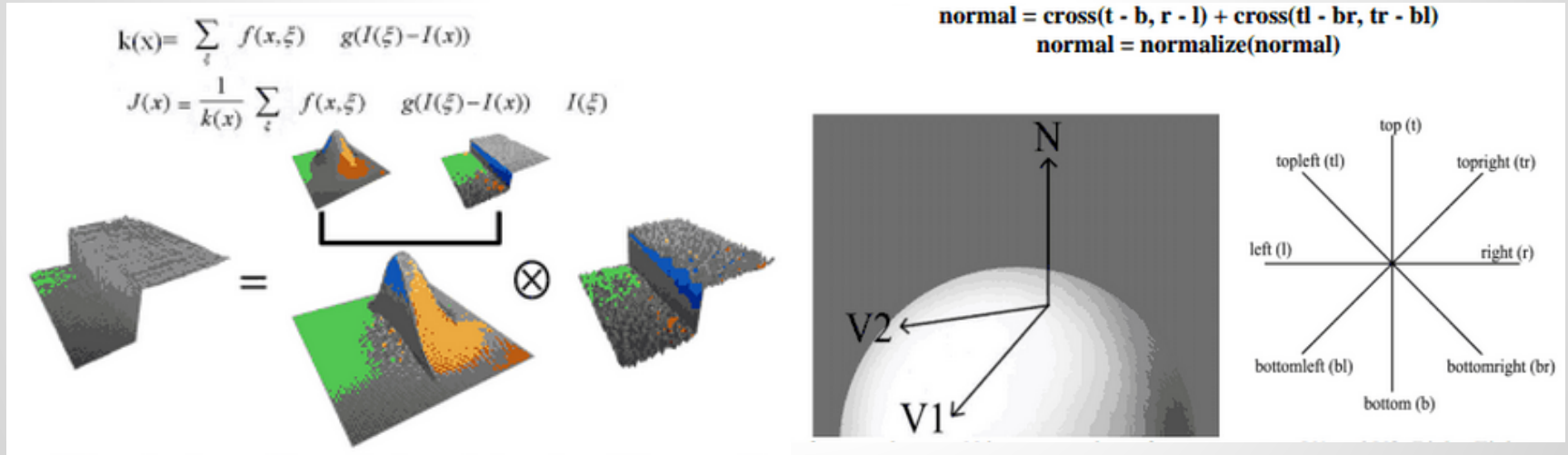


The screenshot shows the Mr. Doob HTML Editor interface. The top bar includes the Mr. Doob logo, social media links, and a toolbar with icons for file operations and editing. The main area is split into a code editor on the left and a 3D preview window on the right. The code editor displays a complete HTML document with a JavaScript script that initializes a Three.js scene and animates a wireframe cube. The 3D preview window shows a white wireframe cube centered in a 3D space.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <style>
6       body {
7         background-color: #ffffff;
8         margin: 0;
9         overflow: hidden;
10      }
11    </style>
12  </head>
13  <body>
14    <script src="http://cdnjs.cloudflare.com/ajax/libs/three.js/r57/three.min.js"></script>
15    <script>
16
17      var camera, scene, renderer;
18      var geometry, material, mesh;
19
20      var init = function () {
21
22        renderer = new THREE.CanvasRenderer();
23        renderer.setSize( window.innerWidth, window.innerHeight );
24        document.body.appendChild( renderer.domElement );
25
26        camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 1, 1000 );
27        camera.position.z = 500;
28
29        scene = new THREE.Scene();
30
31        geometry = new THREE.CubeGeometry( 200, 200, 200 );
32        material = new THREE.MeshBasicMaterial( { color: 0x000000, wireframe: true, wireframeLinewidth: 2 } );
33
34        mesh = new THREE.Mesh( geometry, material );
35        scene.add( mesh );
36
37      }
38
39      var animate = function () {
40
41        requestAnimationFrame( animate );
42
43        mesh.rotation.x = Date.now() * 0.0005;
44        mesh.rotation.y = Date.now() * 0.001;
45      }
```

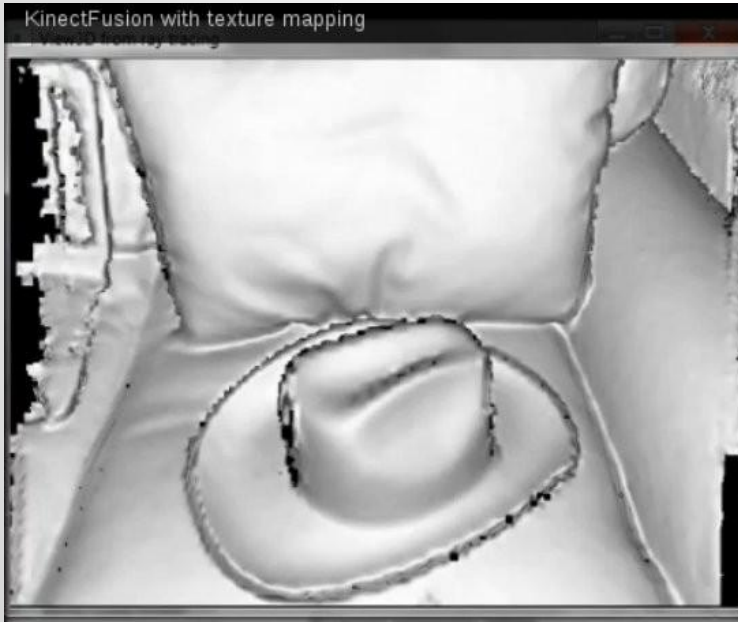
# Demo 2

Artificial Lights: [http://www.azerdev.com/wp-content/uploads/2013/02/IGS\\_Report.pdf](http://www.azerdev.com/wp-content/uploads/2013/02/IGS_Report.pdf)



# Demo 3

Material editing (colors, diffuse/specular) <http://www.youtube.com/watch?v=6o0wjCvKe5c>



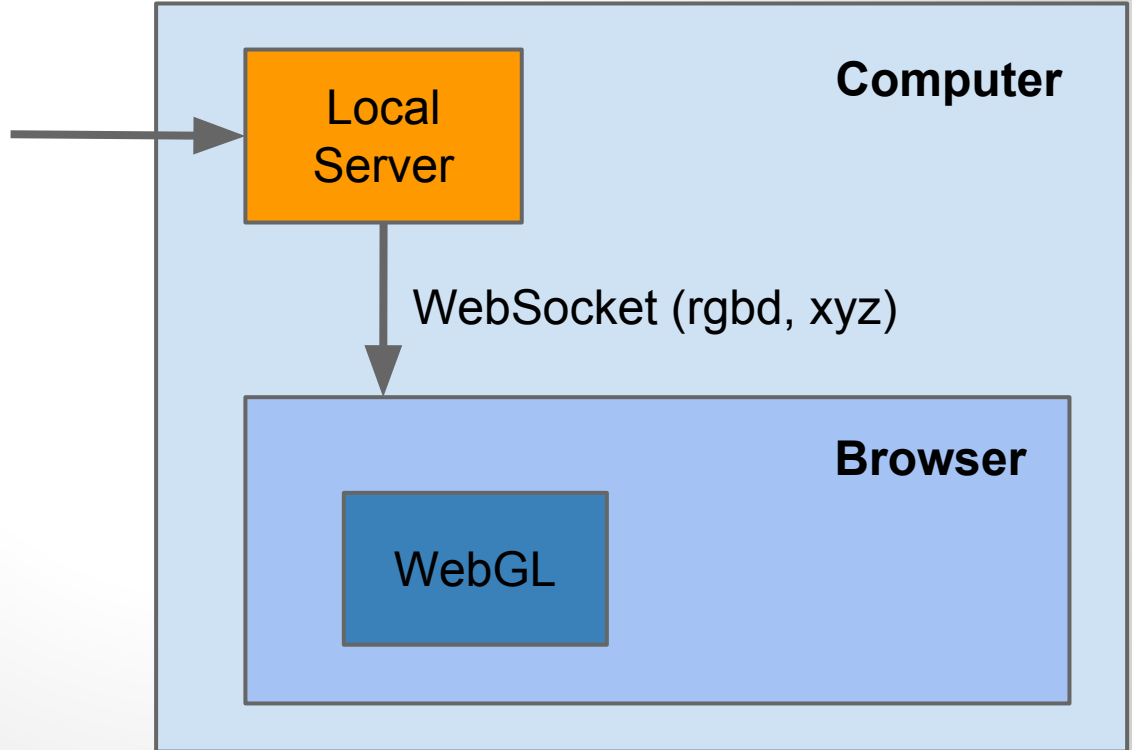
# Current Status - Live Demo



# Method



**Kinect**



# Plans & Challenges

- Challenges

- Process streaming data for WebGL usage efficiently.
- Remove noise from raw data.
- Update the shader.

- Plans

- 2D depth to 3D point cloud.
- Shader editor.
- Demos.



# Google Chrome Extension

## NPAPI

- Can launch external program
- Is used by
  - Silverlight
  - Unity
  - Google Earth
- Is being phased out  
(unavailable after Sept 2014)

## Alternatives

- NaCl  
Sandboxed (prohibits external device access)
- Chromium fork
  - Long dev. time
  - Users can't test
- Separate program
  - Requires installer to modify Windows registry (e.g. launch when Kinect is connected)