Raymarching

```
vec4 traverseRay(vec3 first, vec3 last) {
  vec4 result = vec4(0.0);
  vec3 direction = normalize(last - first);
  float stepIncr = length(last - first) / numSteps;          <- uniform float numSteps
  float t = 0.0;
  for (int i = 0; i < numSteps; ++i) {
    vec3 sampleCoord = first + t * direction;
    float intensity = texture(volume, sampleCoord).a;        <- uniform sampler3D volume
    vec4 color = classify(transferFunction, intensity);      <- uniform sampler1D transferFunction
    if (color.a > 0.0) {
      result.rgb = result.rgb + (1.0 - result.a) * color.a * color.rgb;
      result.a   = result.a + (1.0 - result.a) * color.a;
    }
    t += stepIncr;
  }
  return result;
}
```

Demo

WebGL