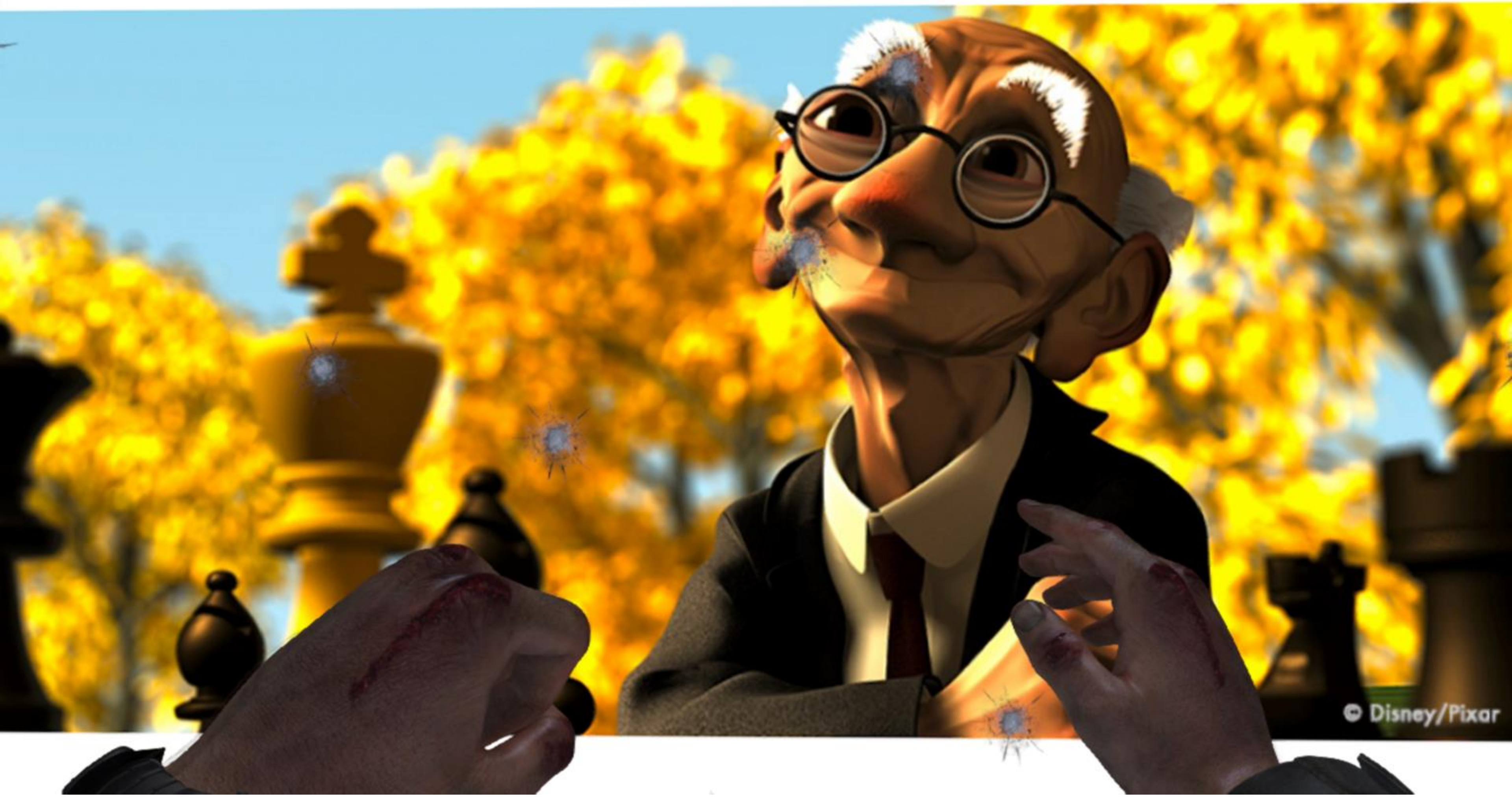
vancouver SIGGRAPH2014 NATURALLY DIGITAL

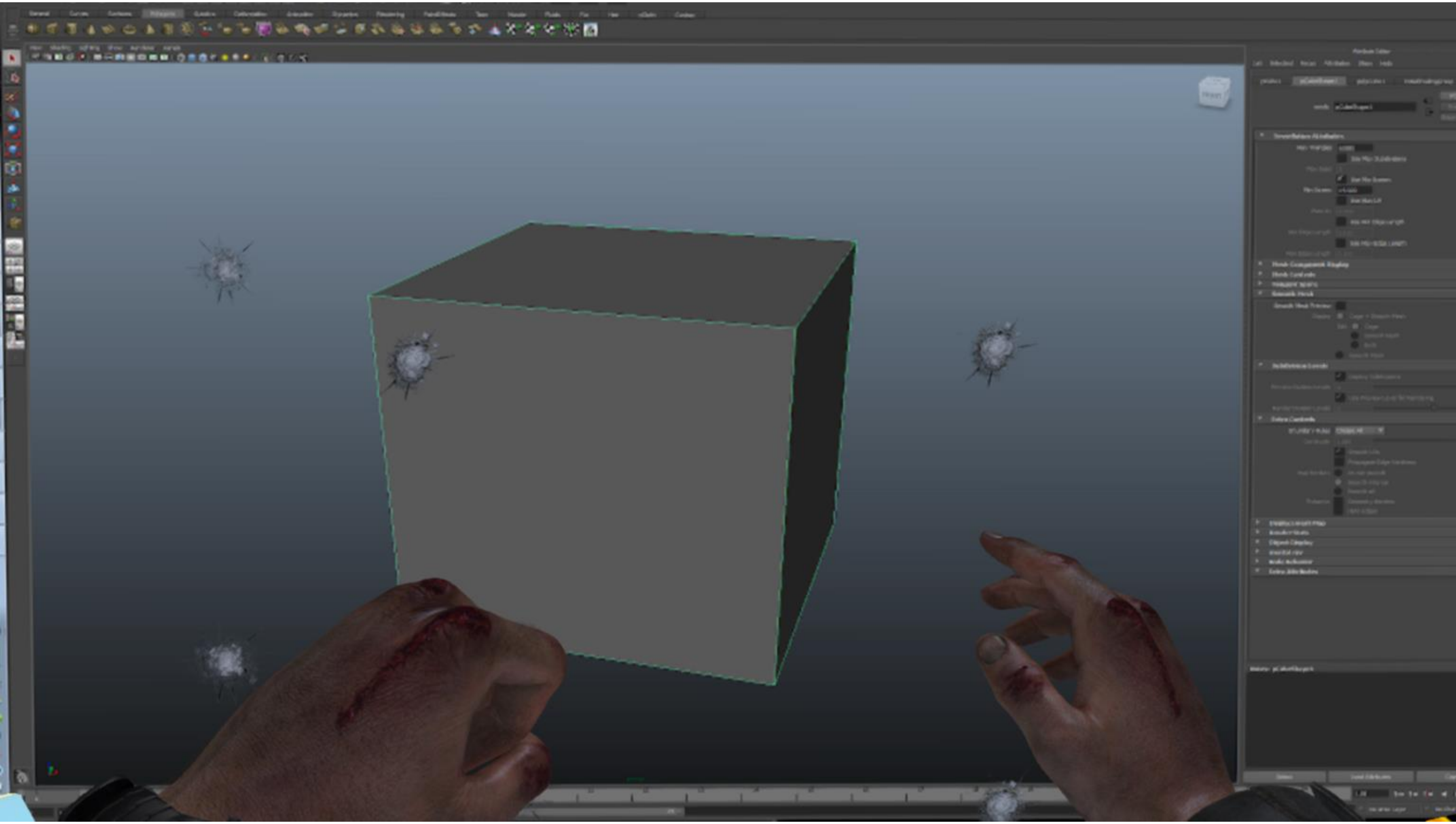
The 41st International Conference and Exhibition on Computer Graphics and Interactive Techniques

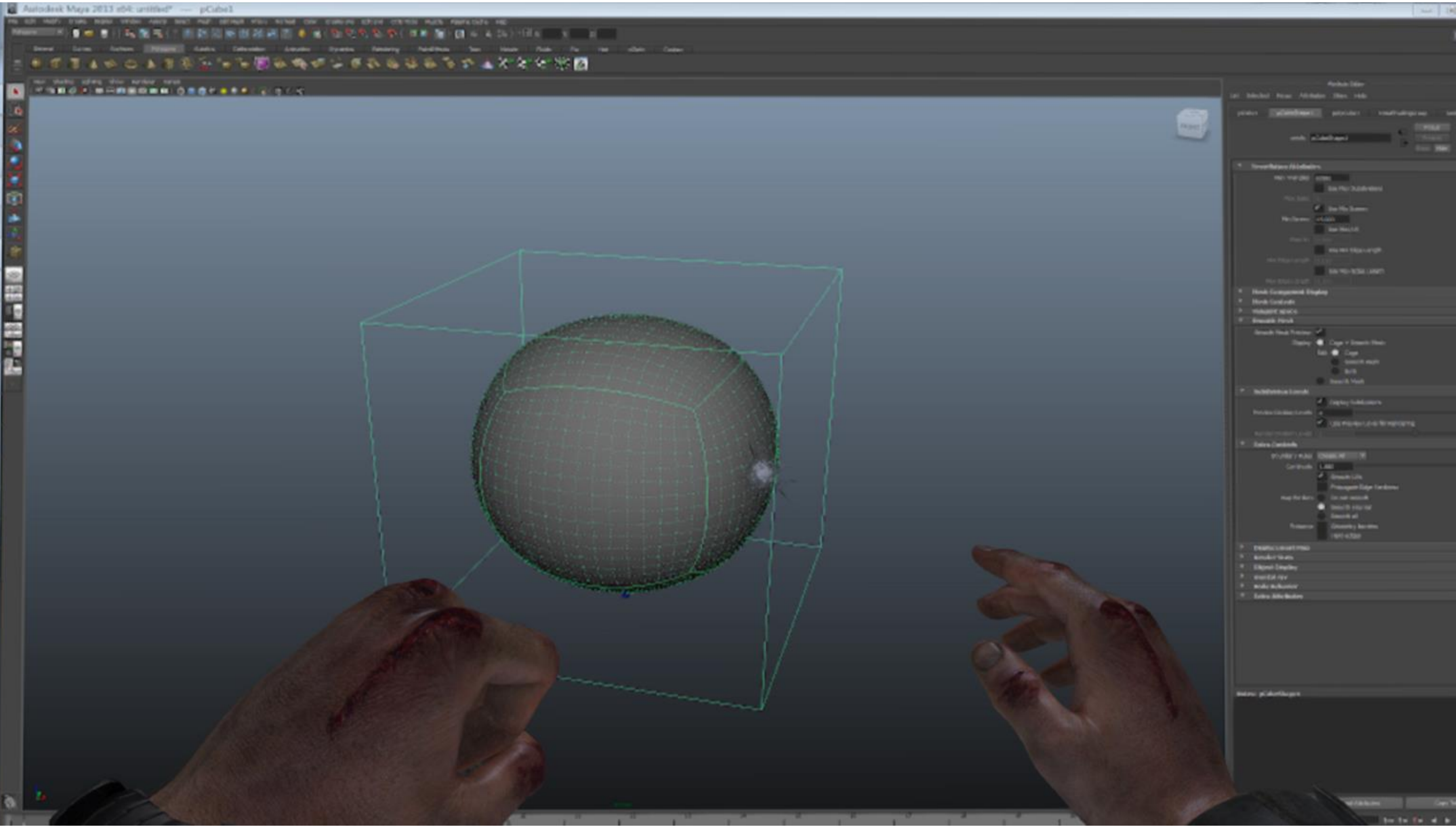
Tessellation in Call of Duty: Ghosts

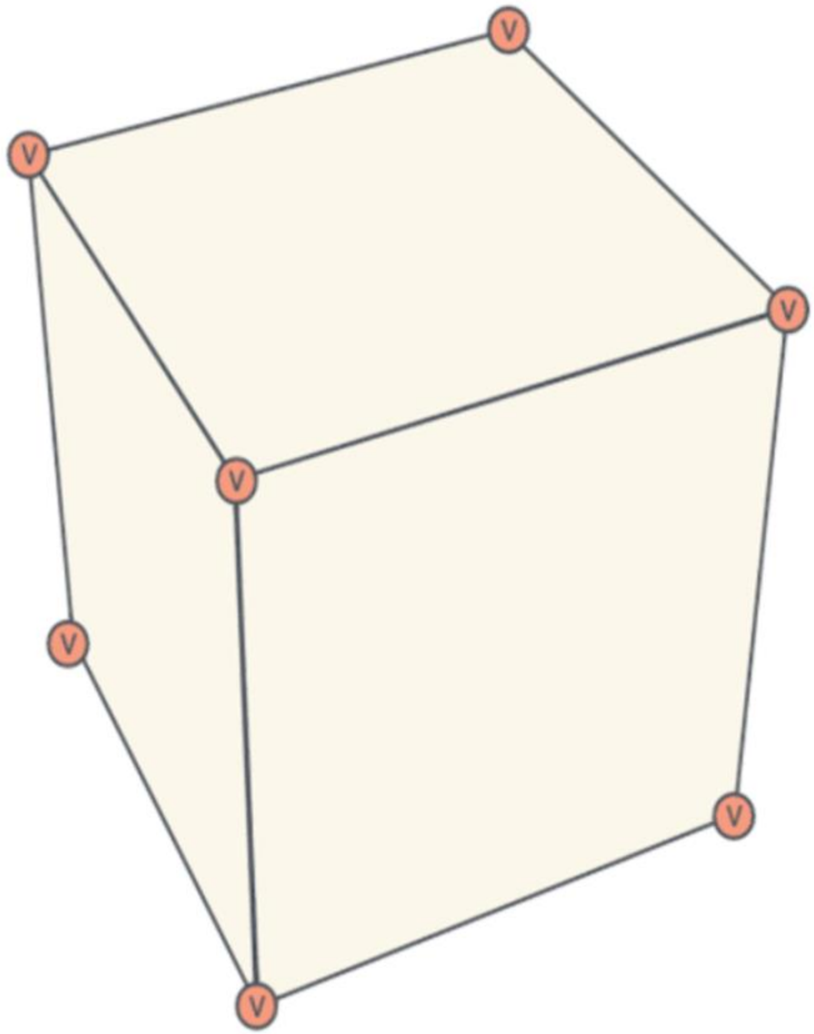
Wade Brainerd
Activision

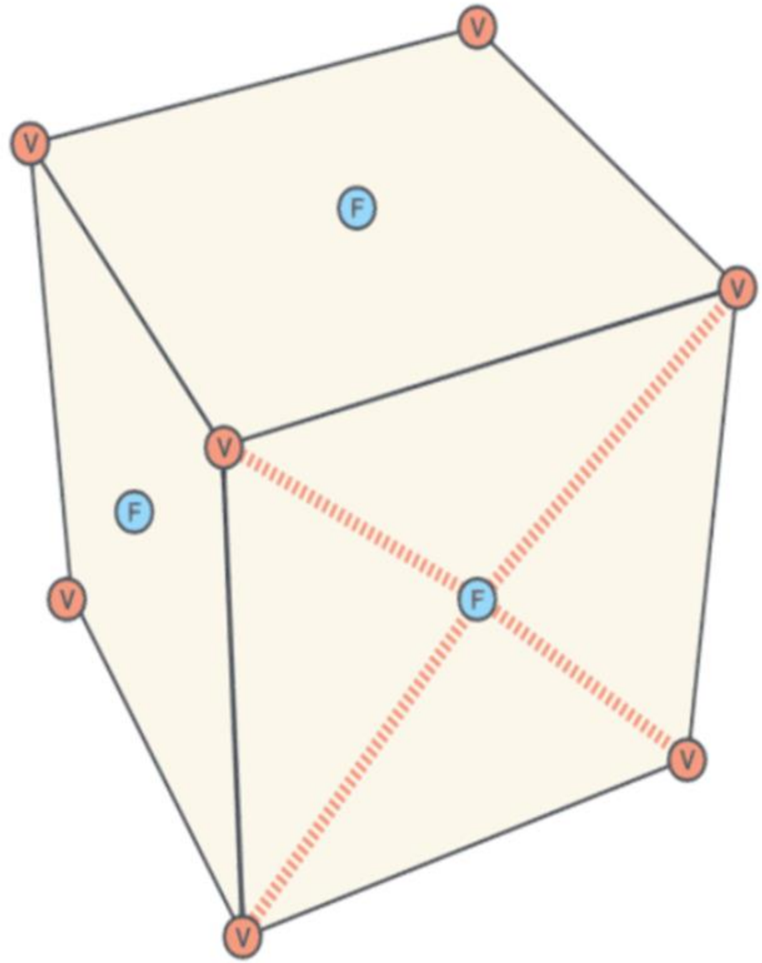

May 1978
My birth
November 1978
E. Catmull & J. Clark:
Recursively generated B-spline surfaces on arbitrary topological meshes
Com er-Aided Design
6): 350-355
OCTOBER
Point

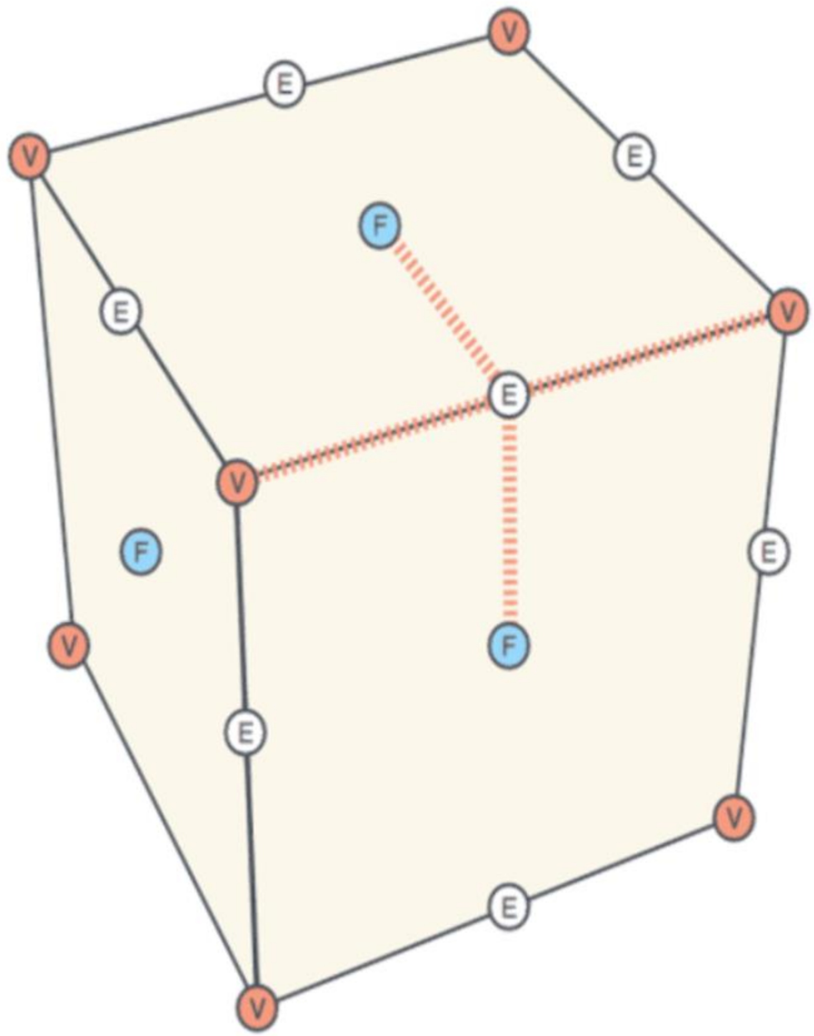

© Disney/Pixar

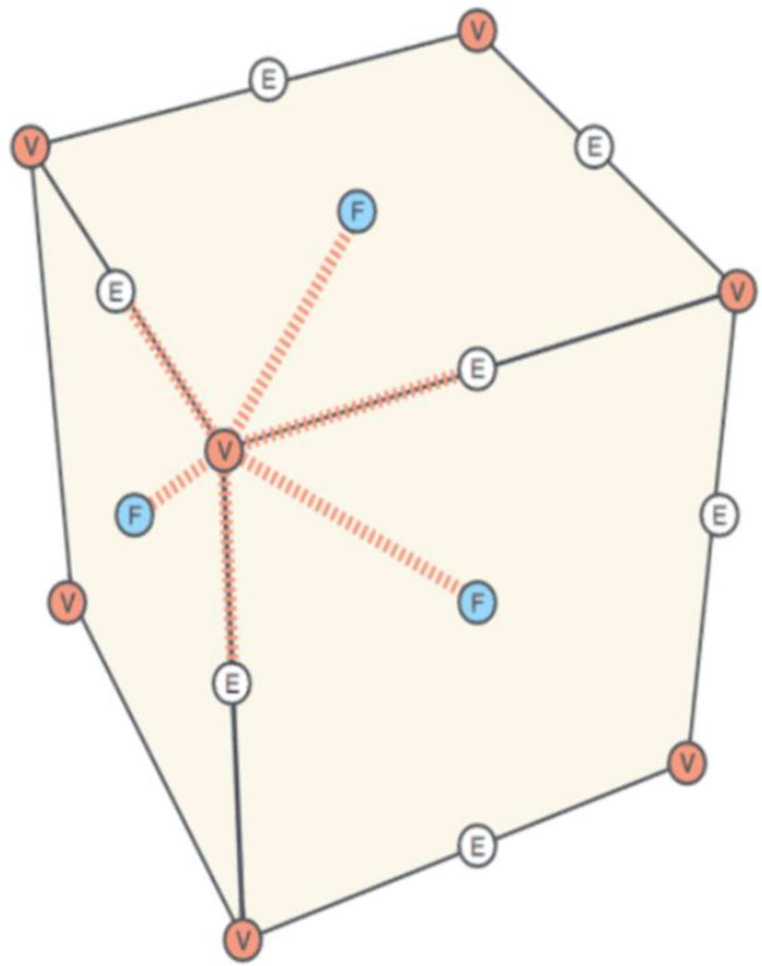


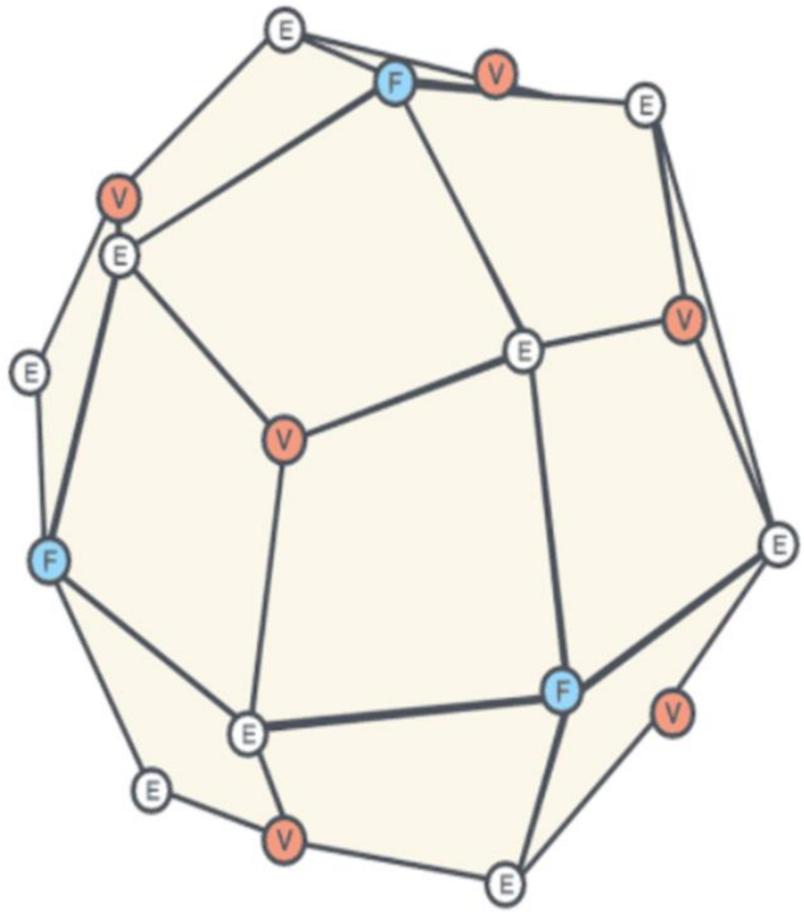


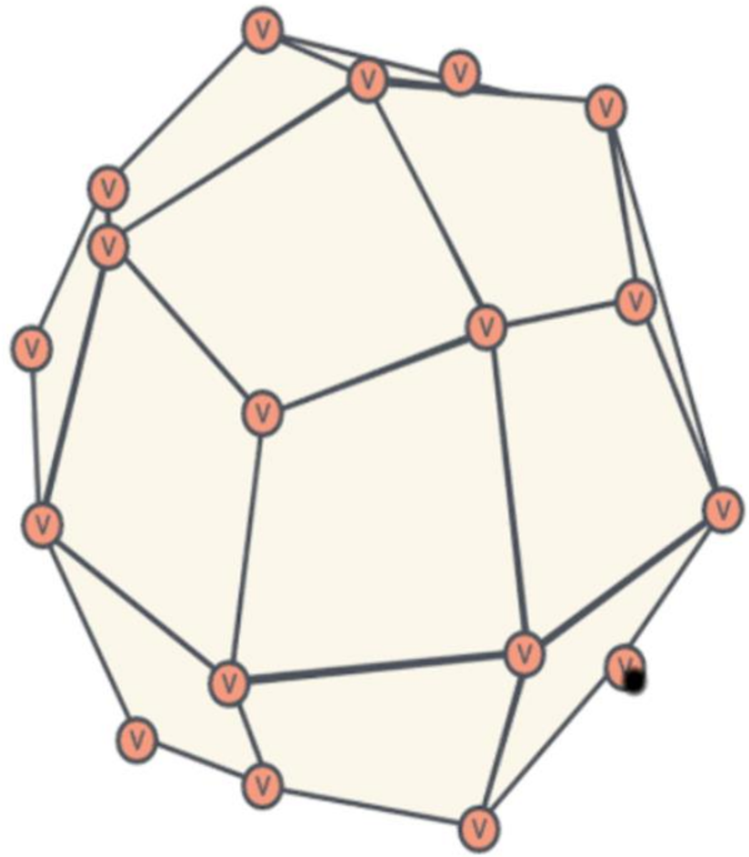

V
V
V
V
V
V
V

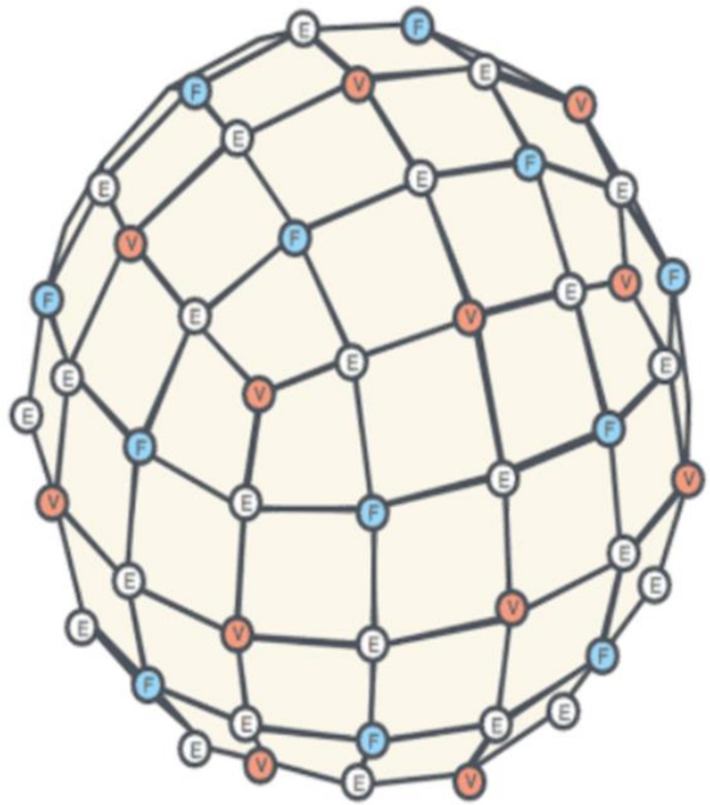

V
V
F
V
V
F
V
F
V
V

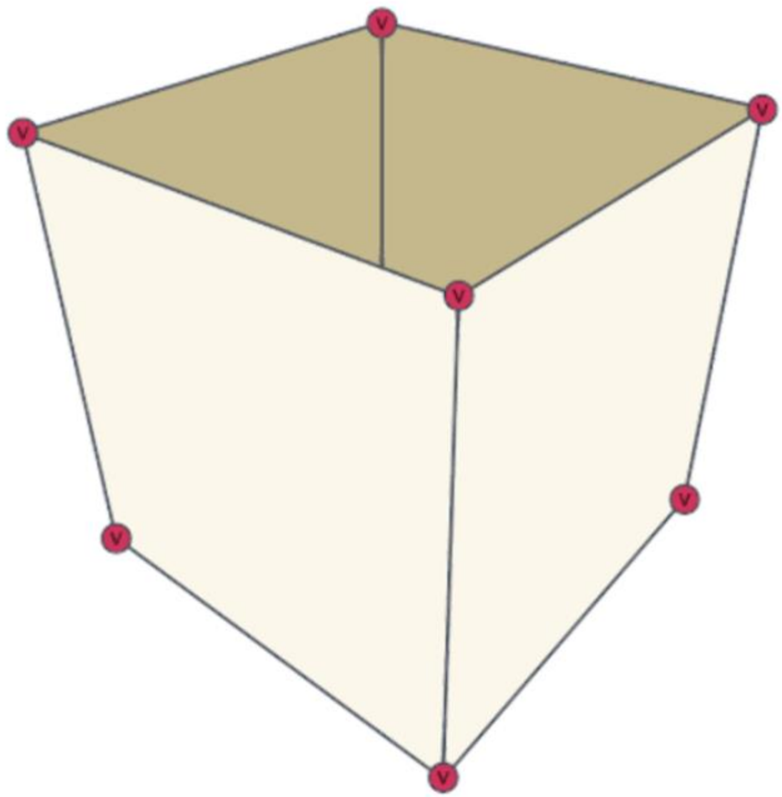

V
E
V
E
F
E
V
E
V
F
E
V
E
F
V
E
V
E
V

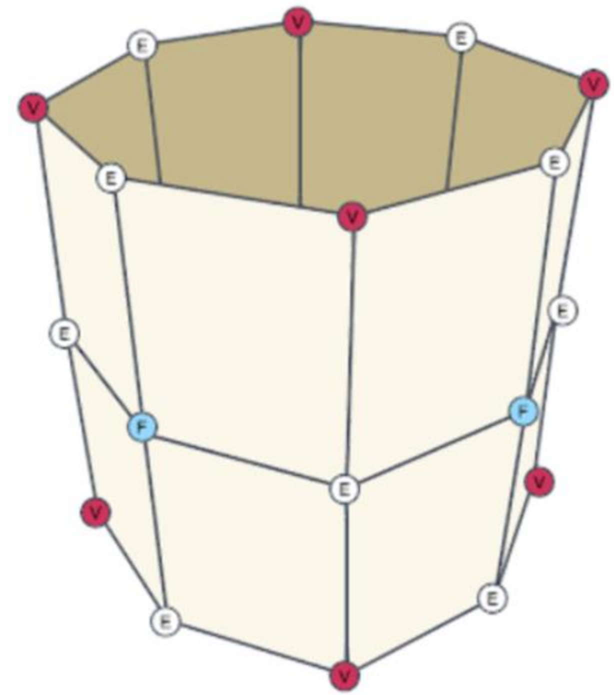

V
E
V
E
V
F
E
V
E
V
F
E
F
V
E
E
V
V

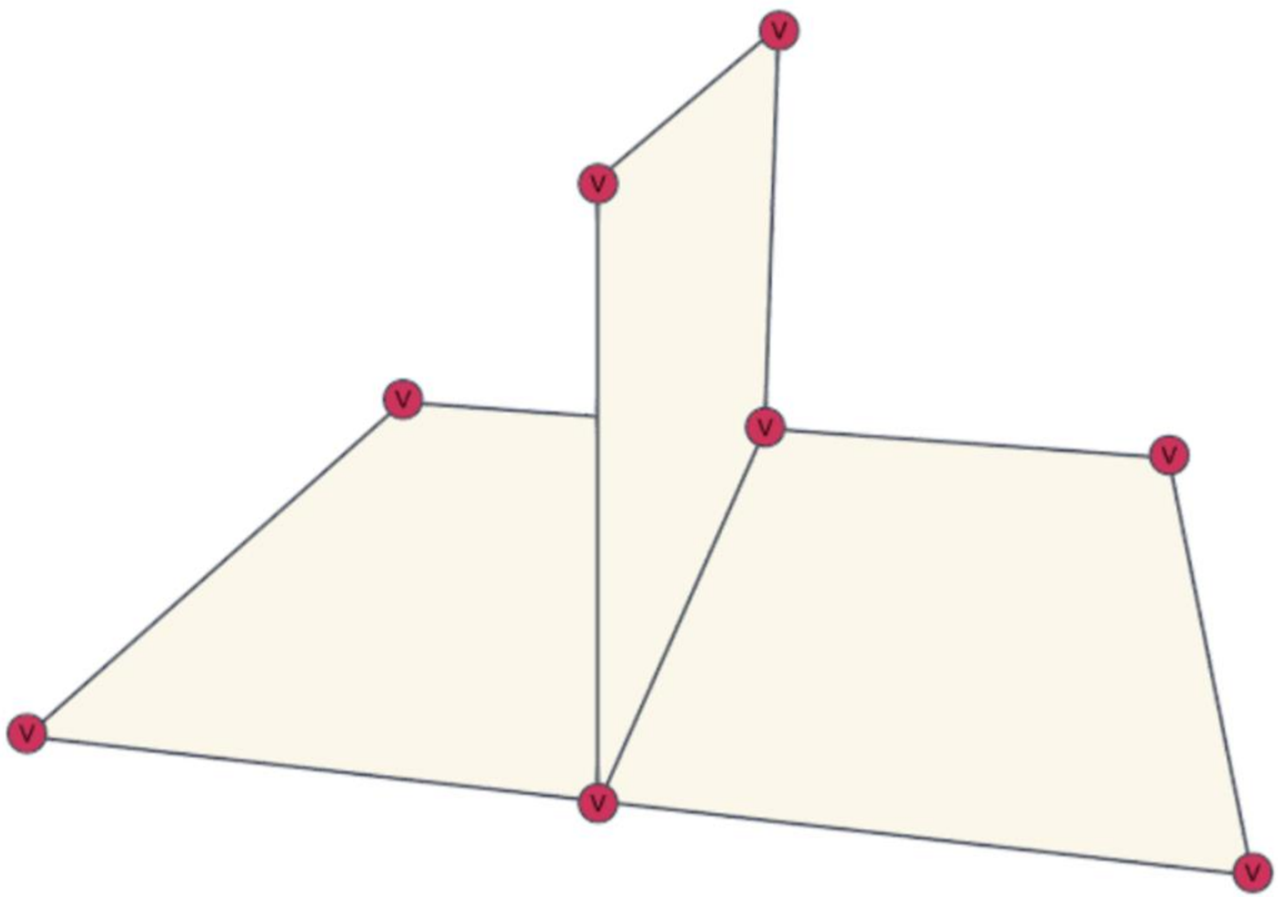

E
F
V
E
V
E
E
E
V
V
F
E
F
V
E
V
E



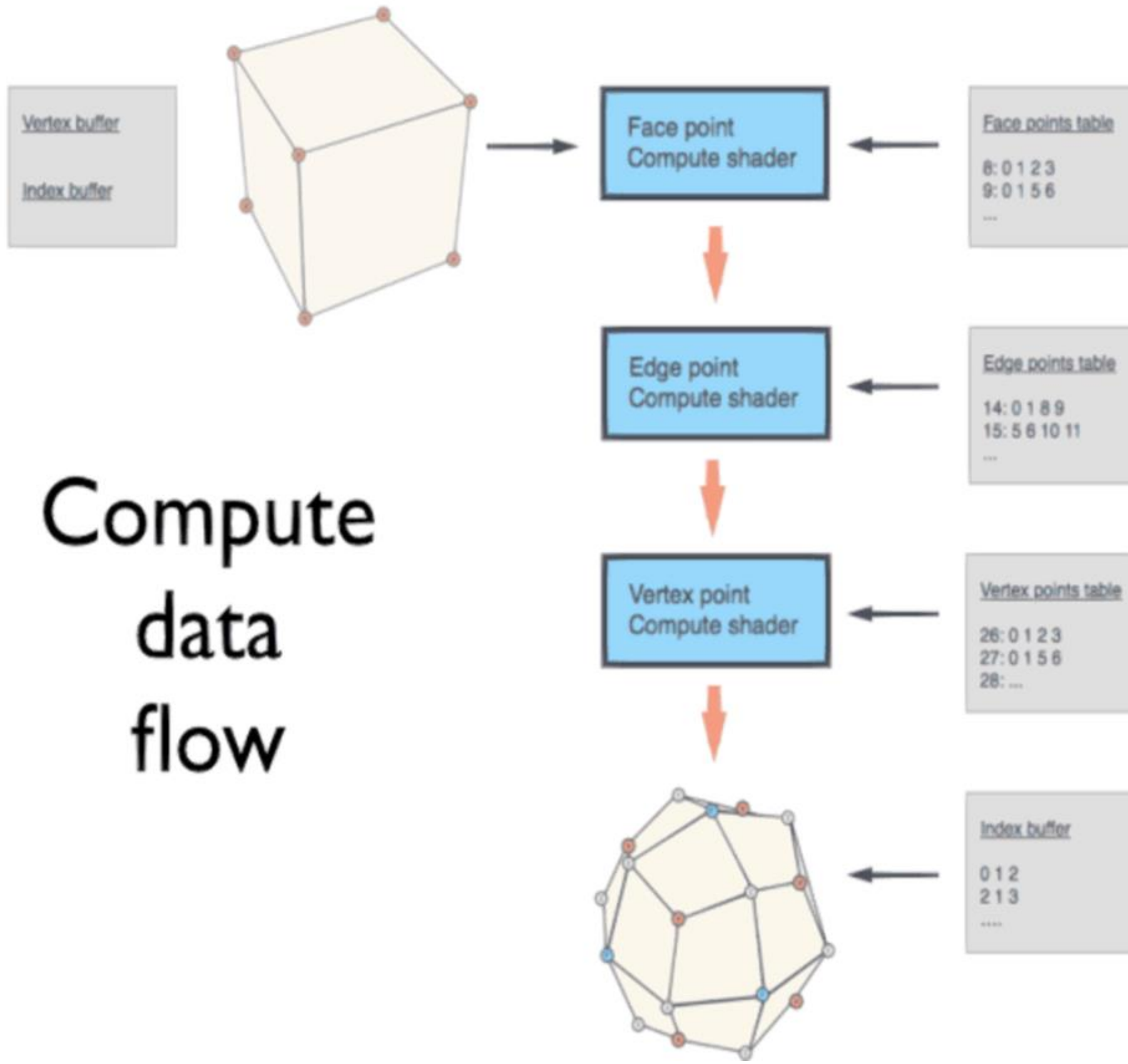



V
V
V
V
V
V
V

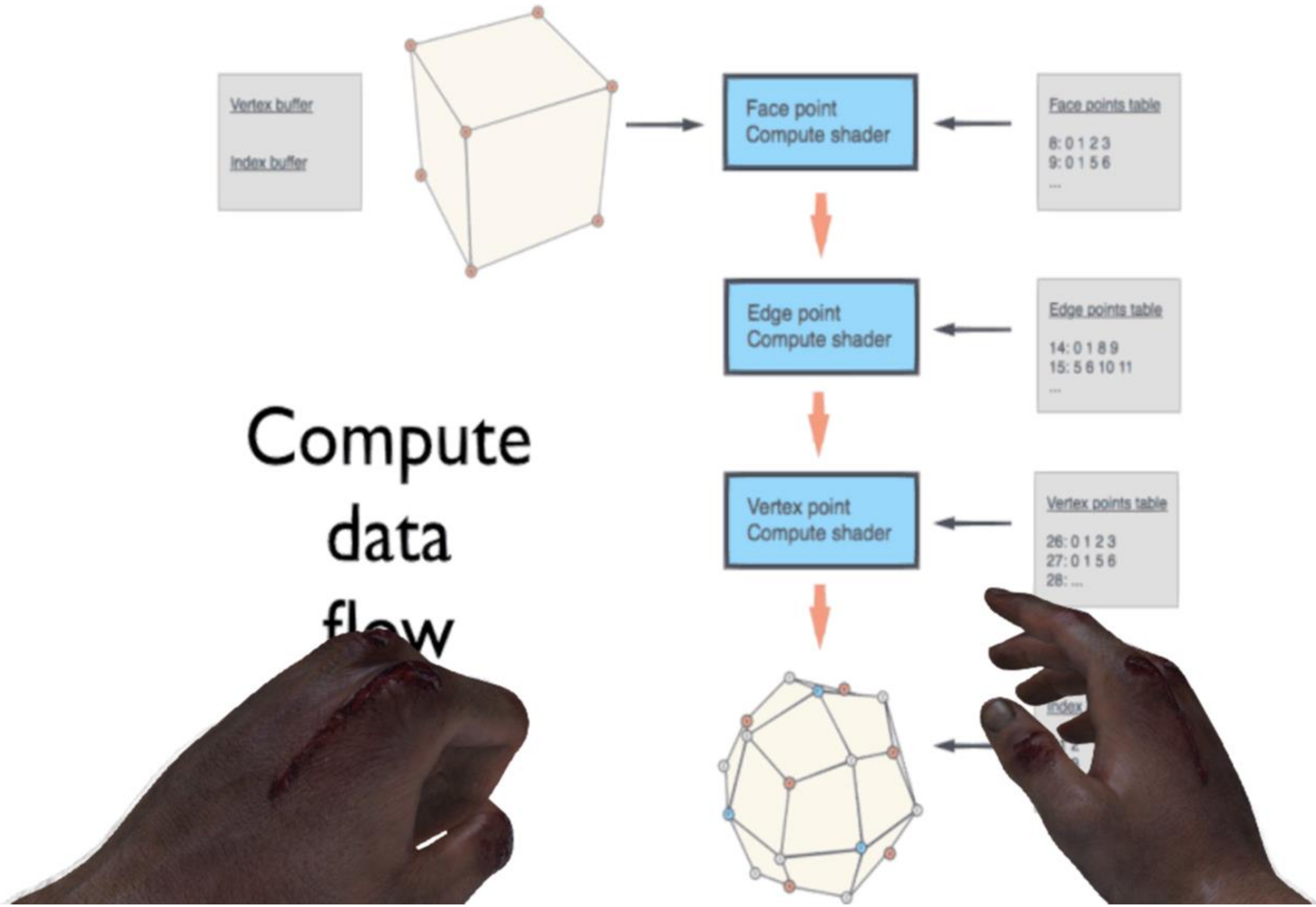

V
E
E
V
V
E
E
V
E
E
F
F
E
V
V
E
E
V


V
V
V
V
V
V
V
V

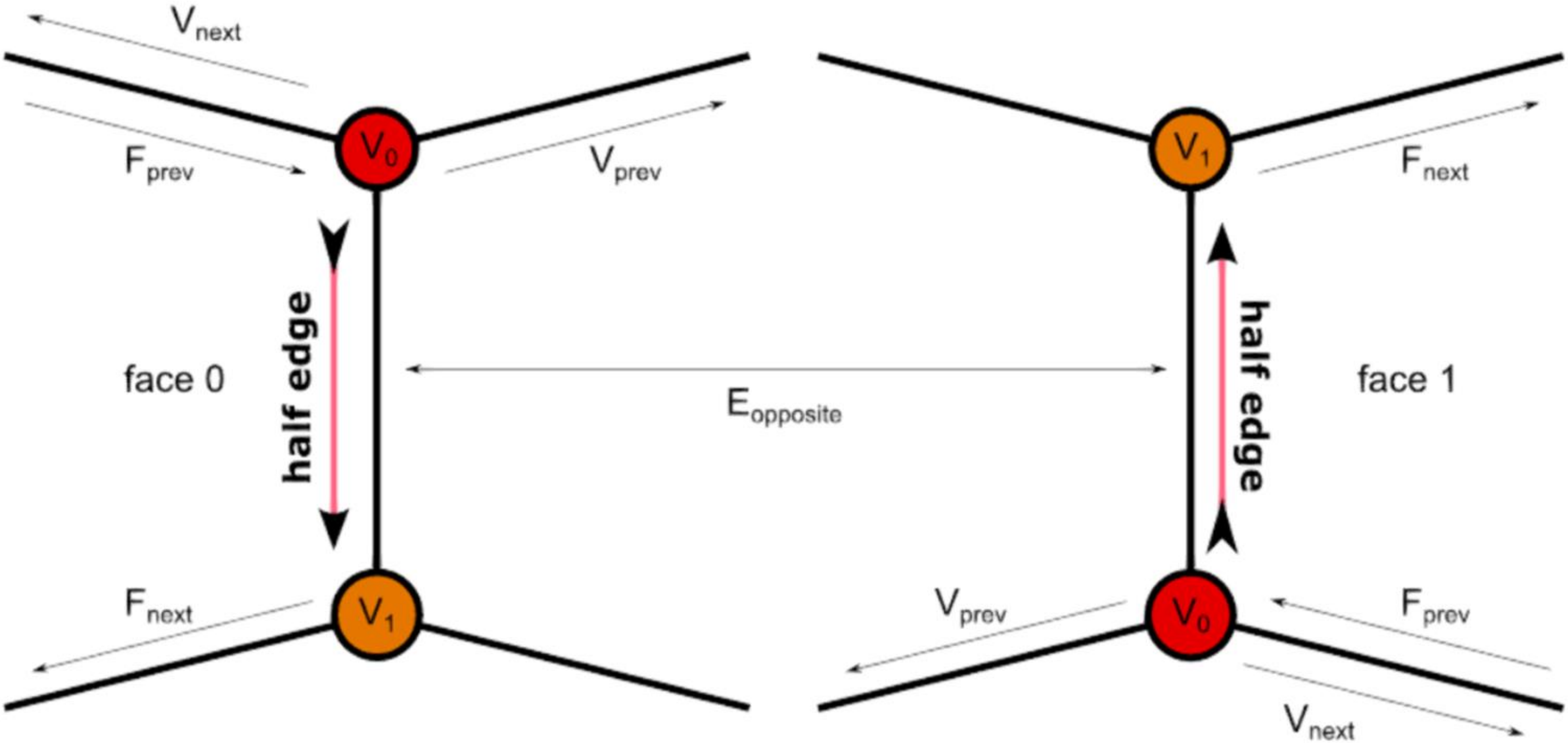
Compute data flow



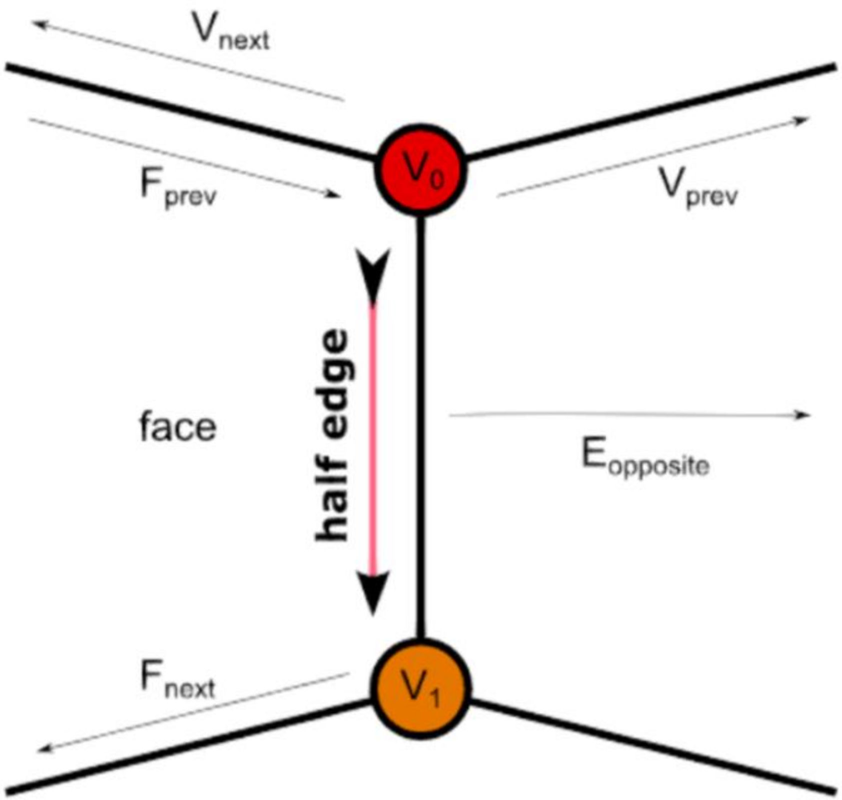
Compute data flow



Quad Edge Mesh



Quad Edge Mesh

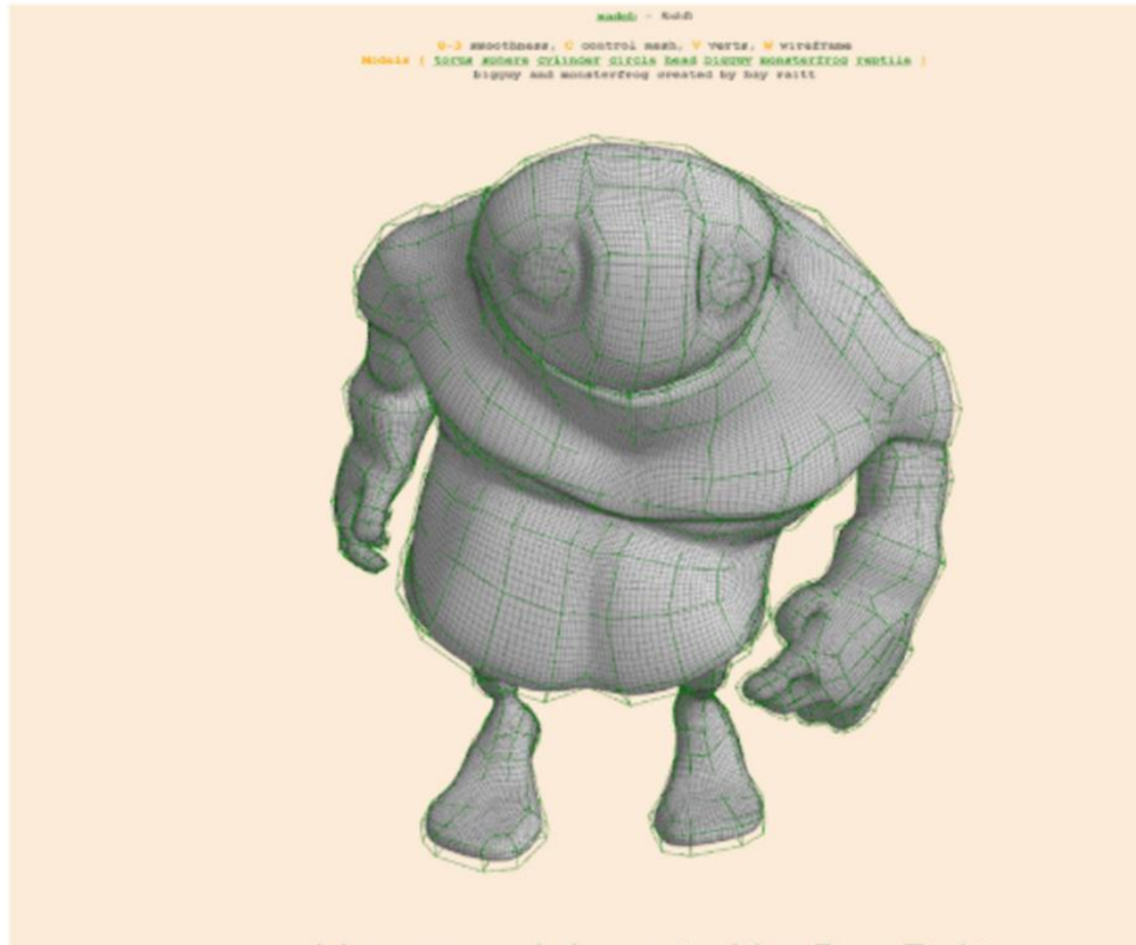



```
struct HalfEdge
{
        HalfEdge *opposite;
        HalfEdge *facePrev;
        HalfEdge *faceNext;
        HalfEdge *vertPrev;
        HalfEdge *vertNext;
        int face;
        int vert0;
        int vert1;
};
```

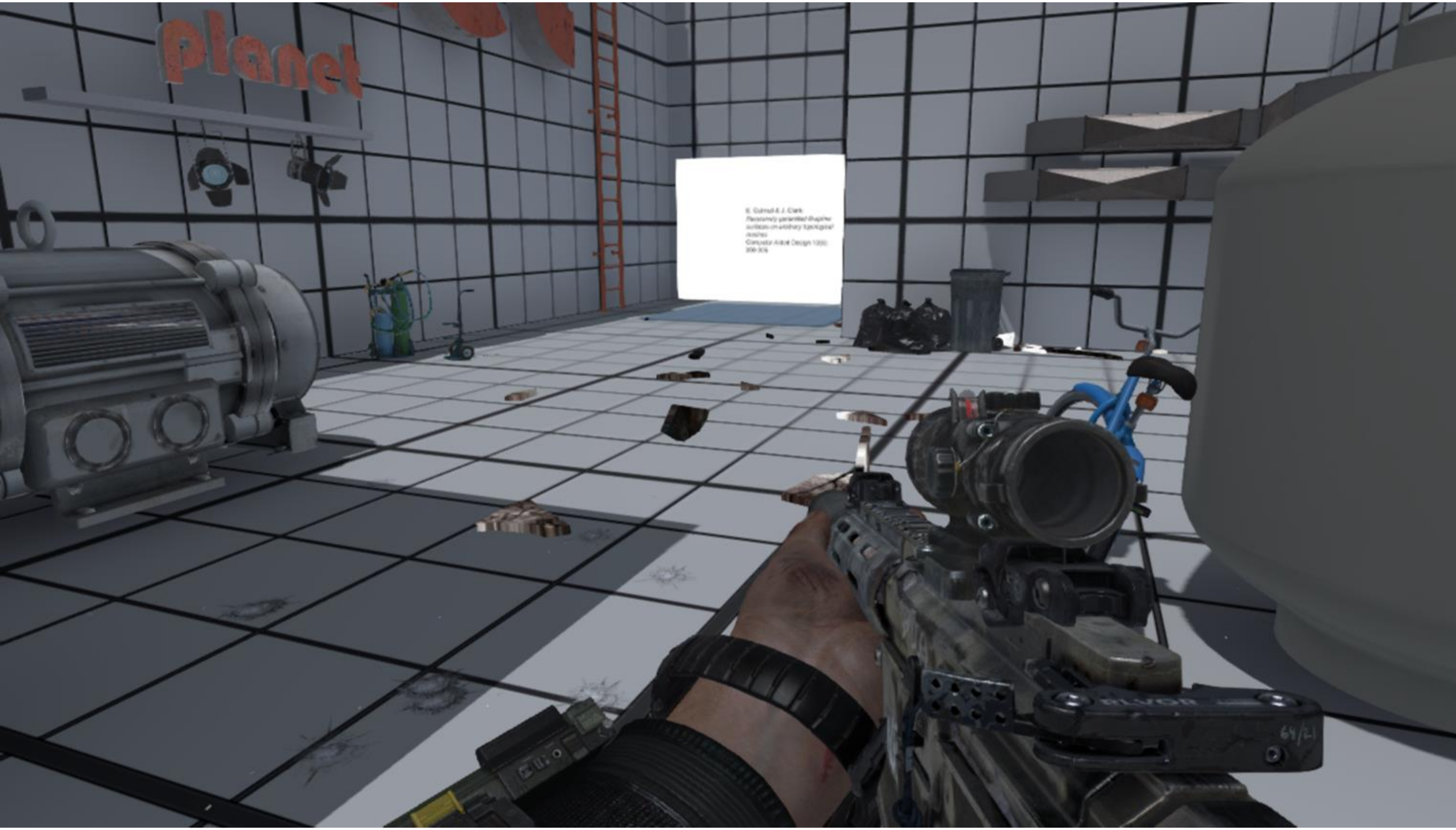
Quad Edge Mesh

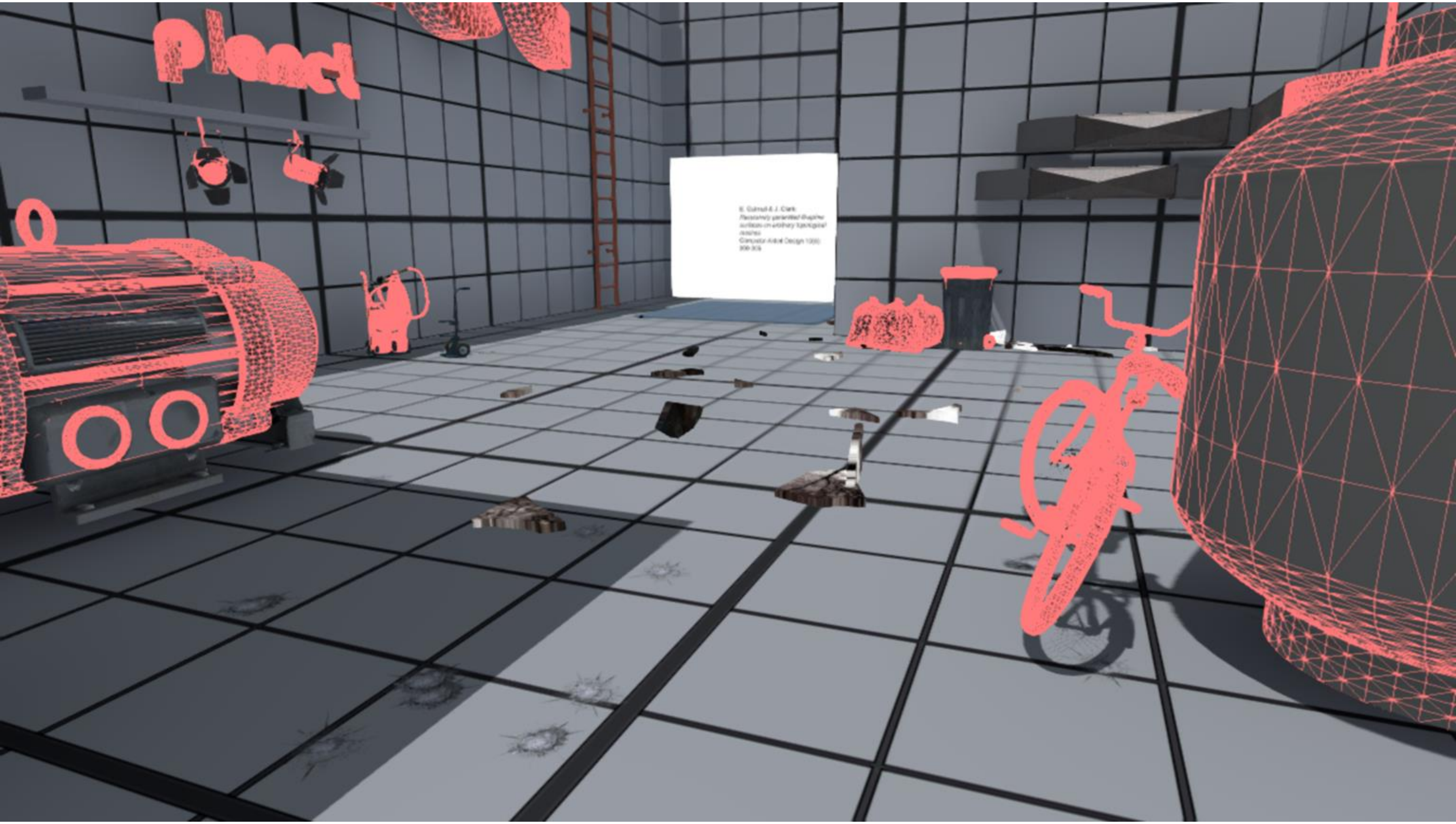
Sample implementation
in JavaScript + WebGL

http://wadeb.com/subd

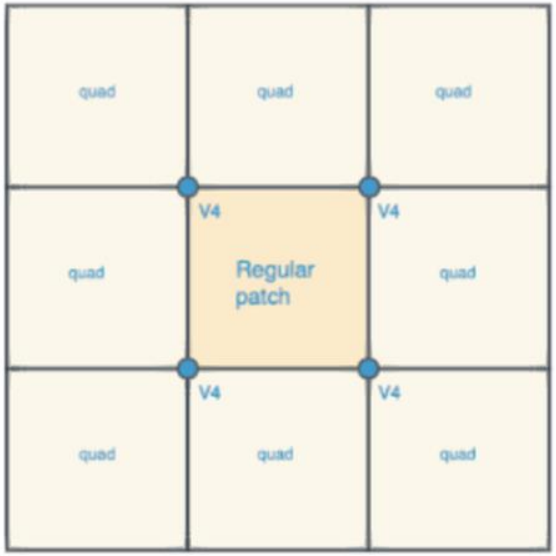



Error: Missing soundalias "weap_hb_lift_plr".
Error: Missing soundalias "weap_hb_stock_plr".
Error: Missing soundalias "weap_hb_grab_plr".

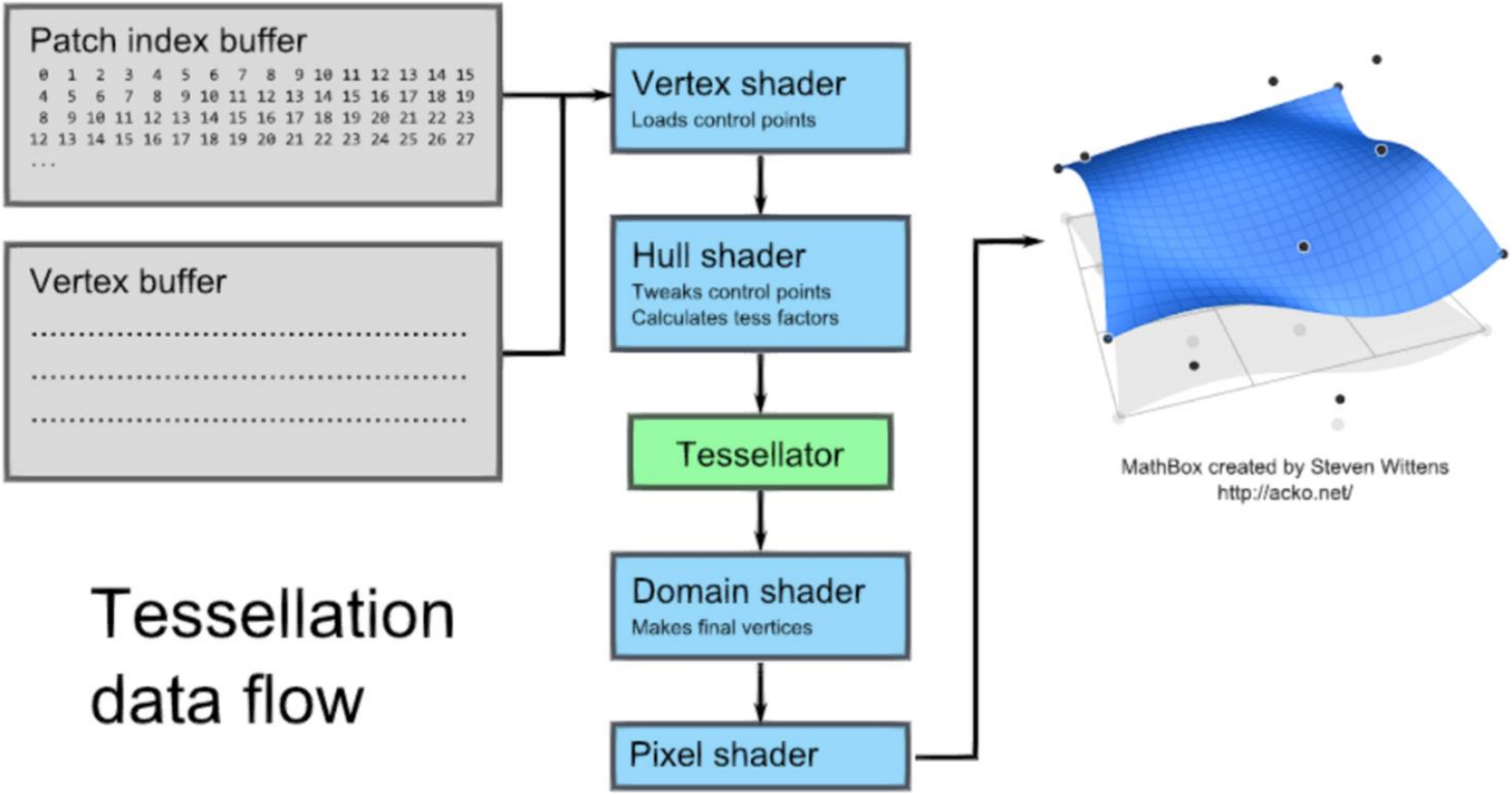

Planet

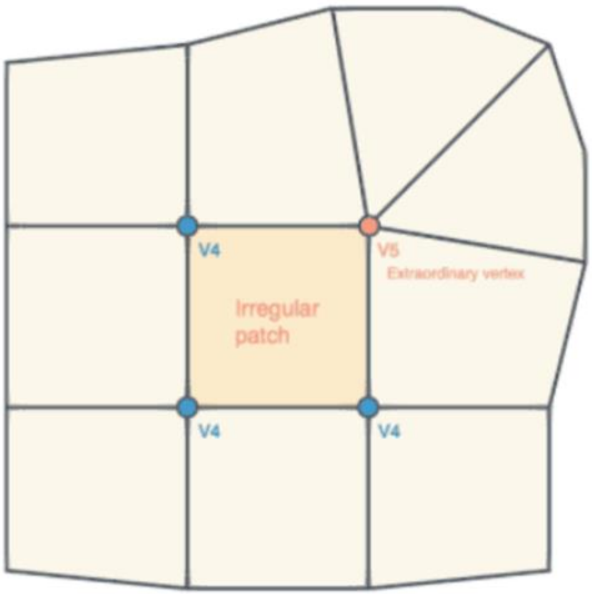


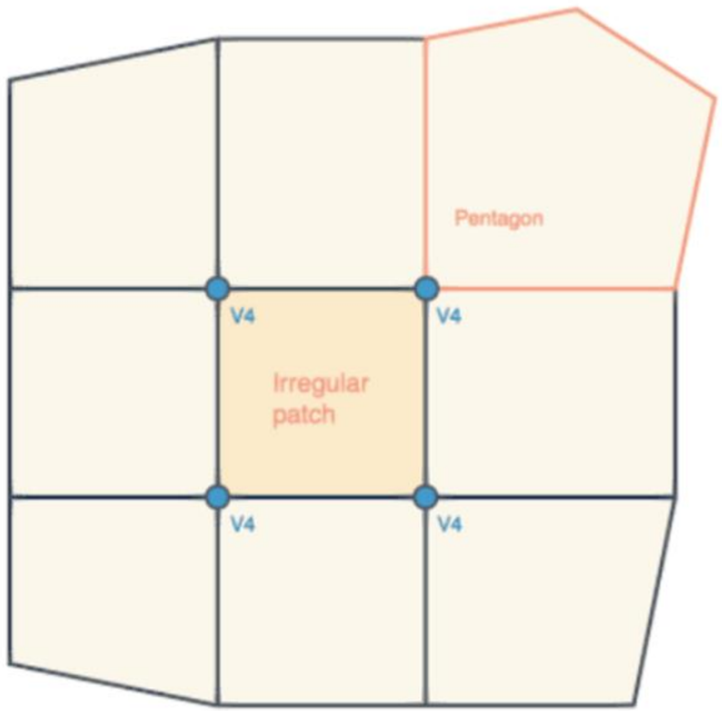
E. Catmull & J. Clark:
Recursively generated B-spline surfaces on arbitrary topological meshes
Computer-Aided Design 10(6): 350-355

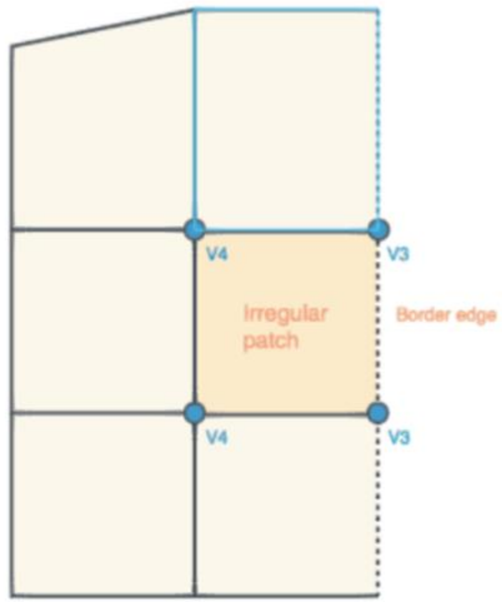

quad
quad
quad
V4
V4
quad
Regular patch
quad
V4
V4
quad
quad
quad

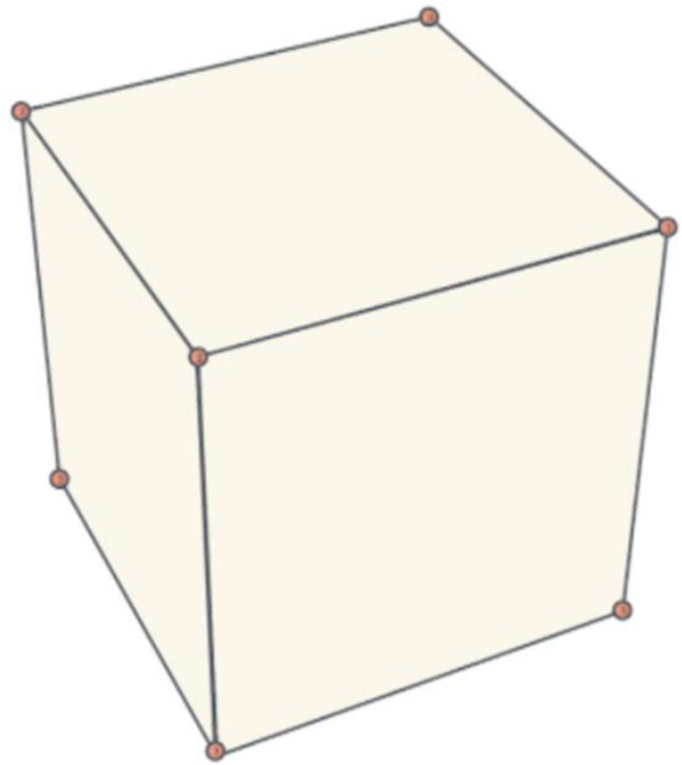
Tessellation data flow

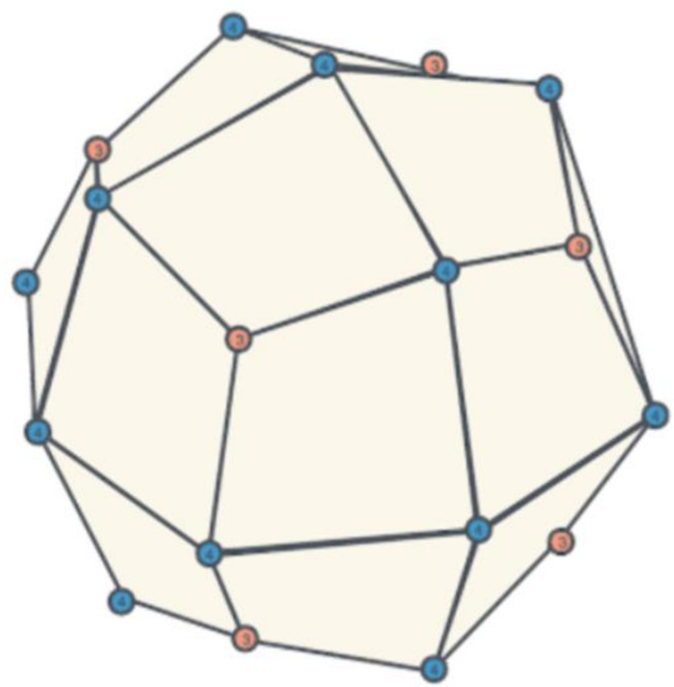


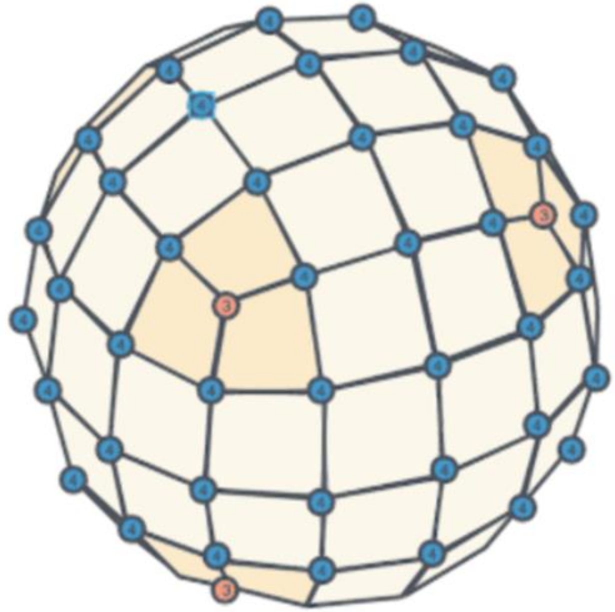

V4
V5
Extraordinary vertex
Irregular
patch
V4
V4

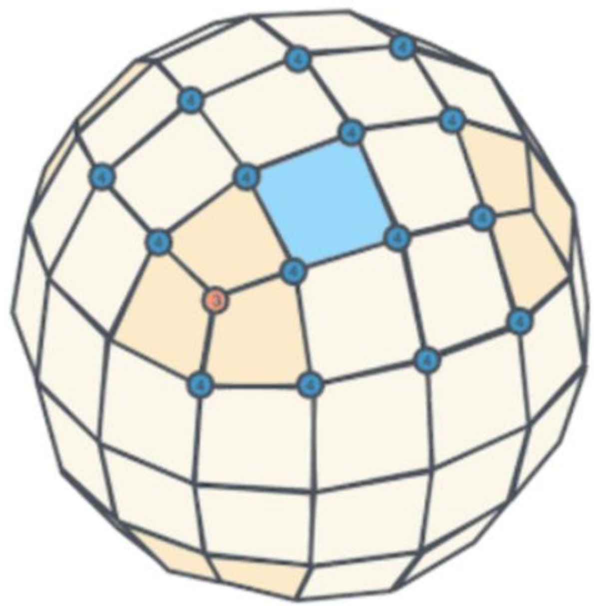

Pentagon
V4
V4
Irregular patch
V4
V4


V4
V3
Irregular patch
Border edge
V4
V3





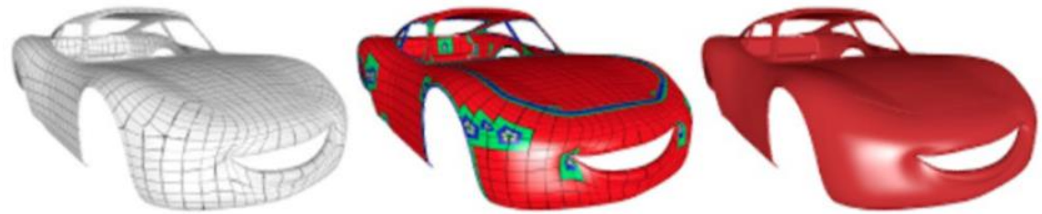




Feature Adaptive Subdivision

Feature Adaptive GPU Rendering of Catmull-Clark
Subdivision Surfaces

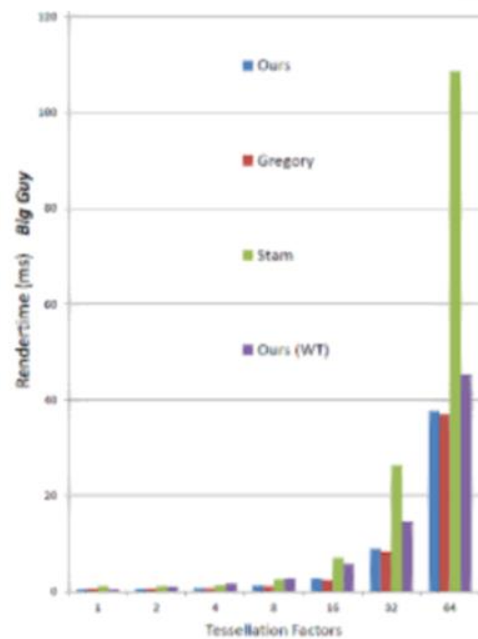
Matthias Nießner
University of Erlangen-Nuremberg
and
Charles Loop
Microsoft Research
and
Mark Meyer and Tony DeRose
Pixar Animation Studios

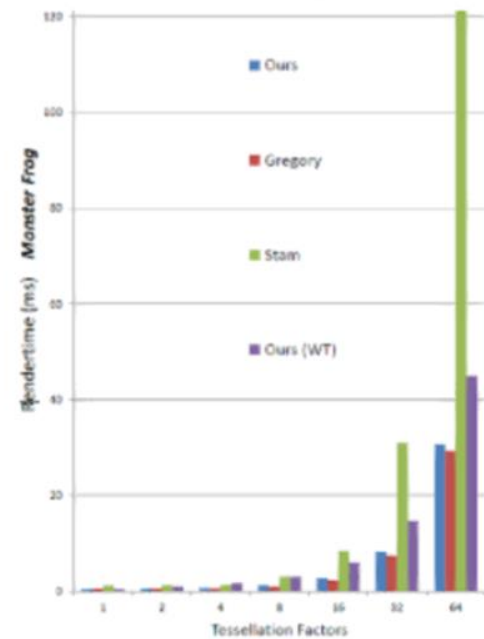


8.1 Comparison to Global Mesh Refinement

Table I. Timing using the Big Guy model for our scheme (feature adaptive patching) compared against our global table driven subdivision method and the previously published GPU subdivision algorithm by Shiue et al.. Note that all timings include final rendering, while we additionally break out draw time for our global subdivision scheme.

Subdivision Level	0	1	2	3	4
Feature Adaptive Patching	0.10	0.20	0.34	0.81	2.30
Shiue Subdivision	0.62	7.26	13.97	21.42	34.93
Global Table Subdivision	0.06	0.18	0.79	3.07	12.05
Draw Time (Table Subd.)	0.04	0.06	0.37	1.45	5.78





Open
Subdiv
graphics.pixar.com

Feature Adaptive Subdivision

Nießner's implementation

http://research.microsoft.com/en-us/downloads/aae4b28d-bcc7-46b5-b179-718f1ead28fb/

OpenSubdiv

https://github.com/PixarAnimationStudios/OpenSubdiv

Implementation steps

Offline Global Subdivision

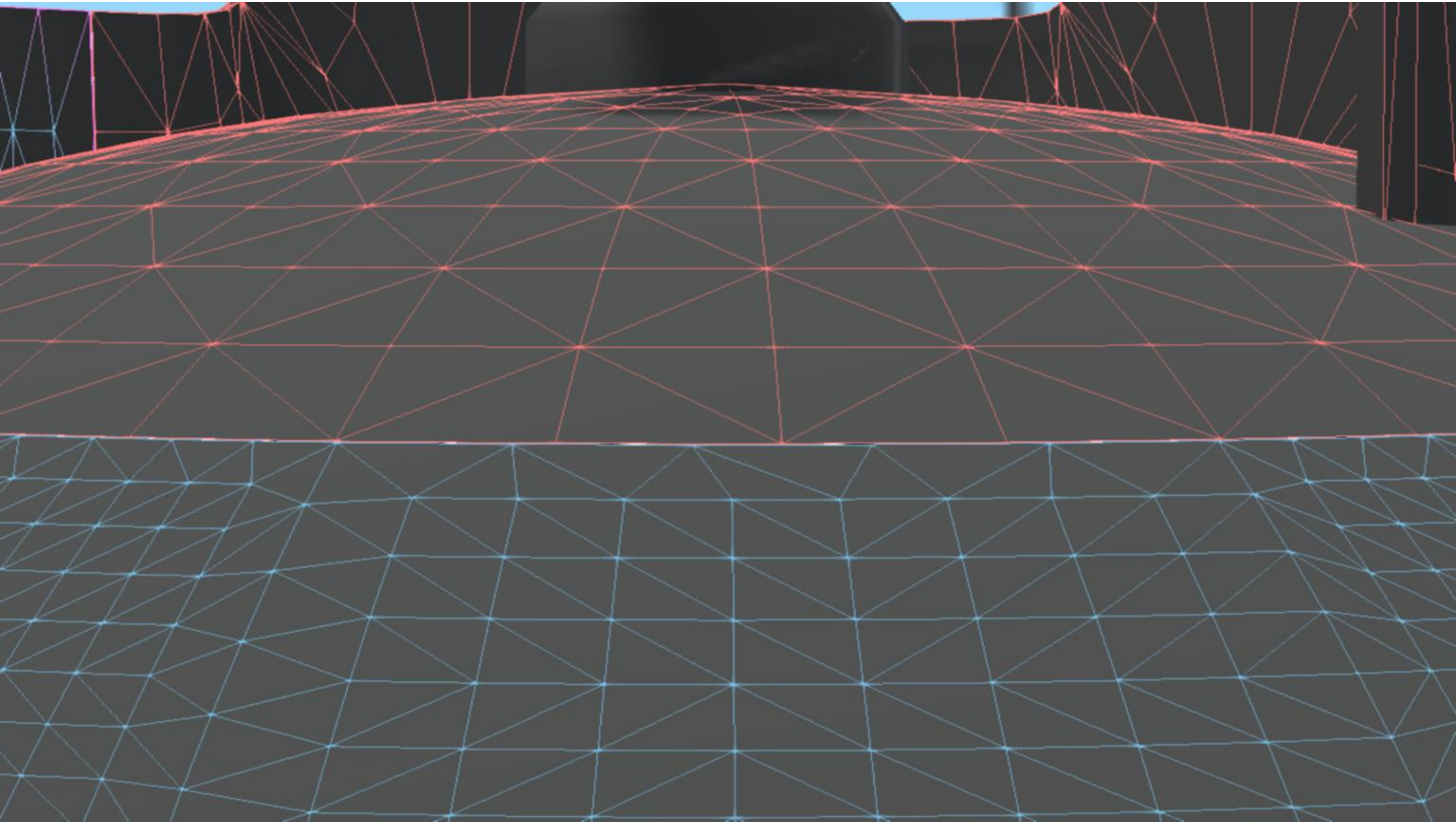
In the model converter
Rendered as triangles
Test DCC parity

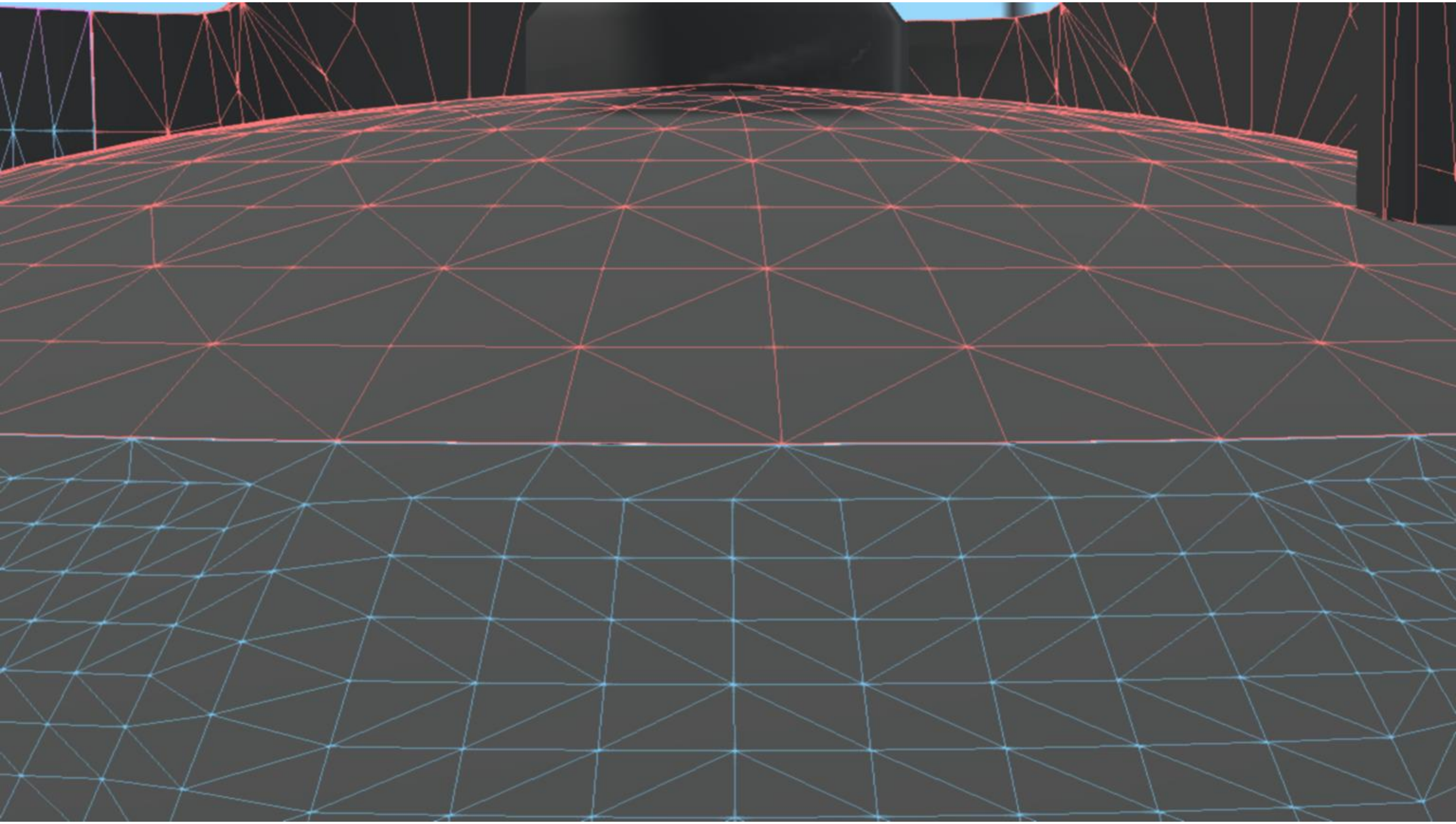
Runtime Global Subdivision

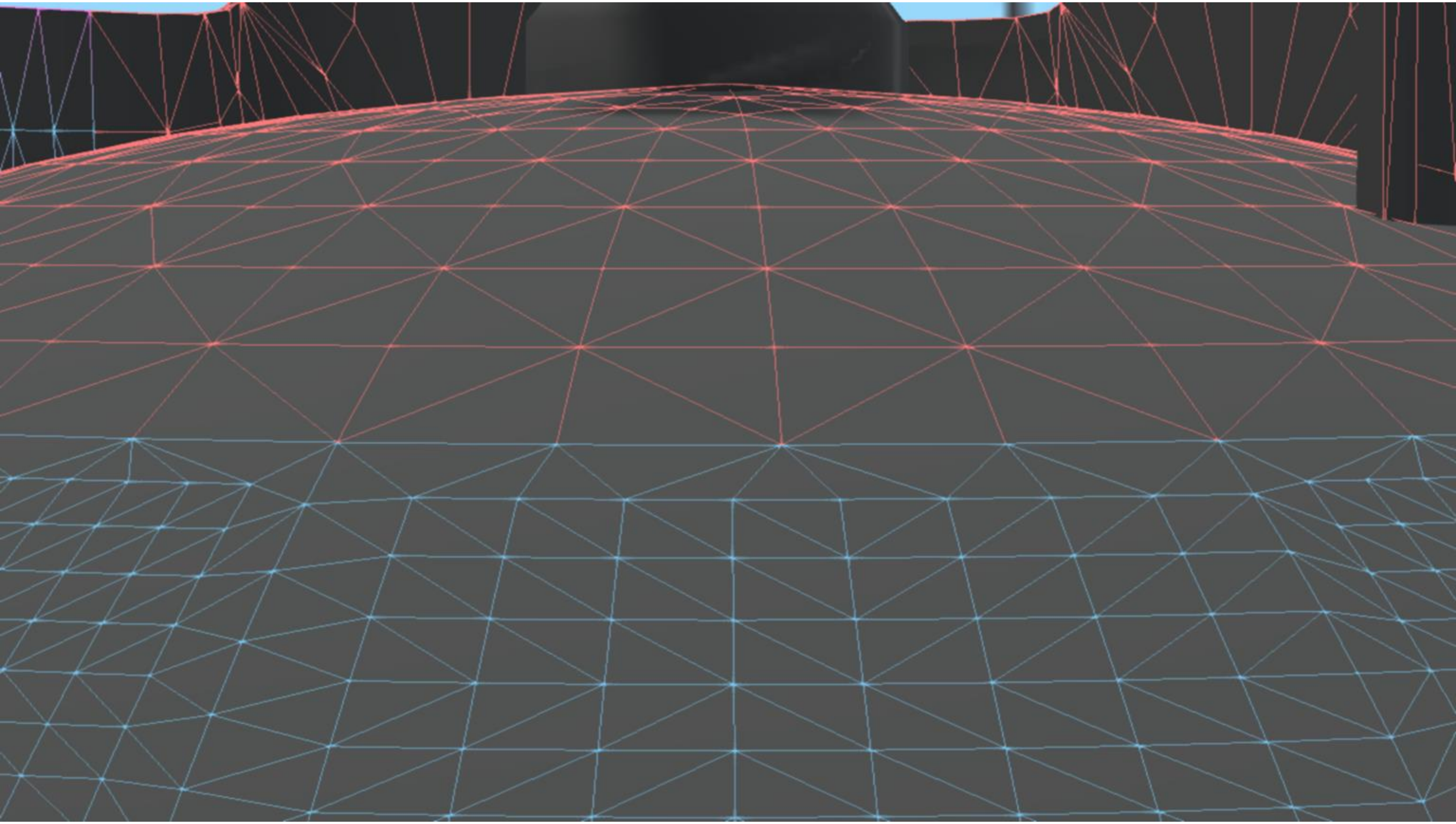
Model converter calculates influence tables
Vertices evaluated by compute shaders
Still rendered as triangles

Feature Adaptive Subdivision

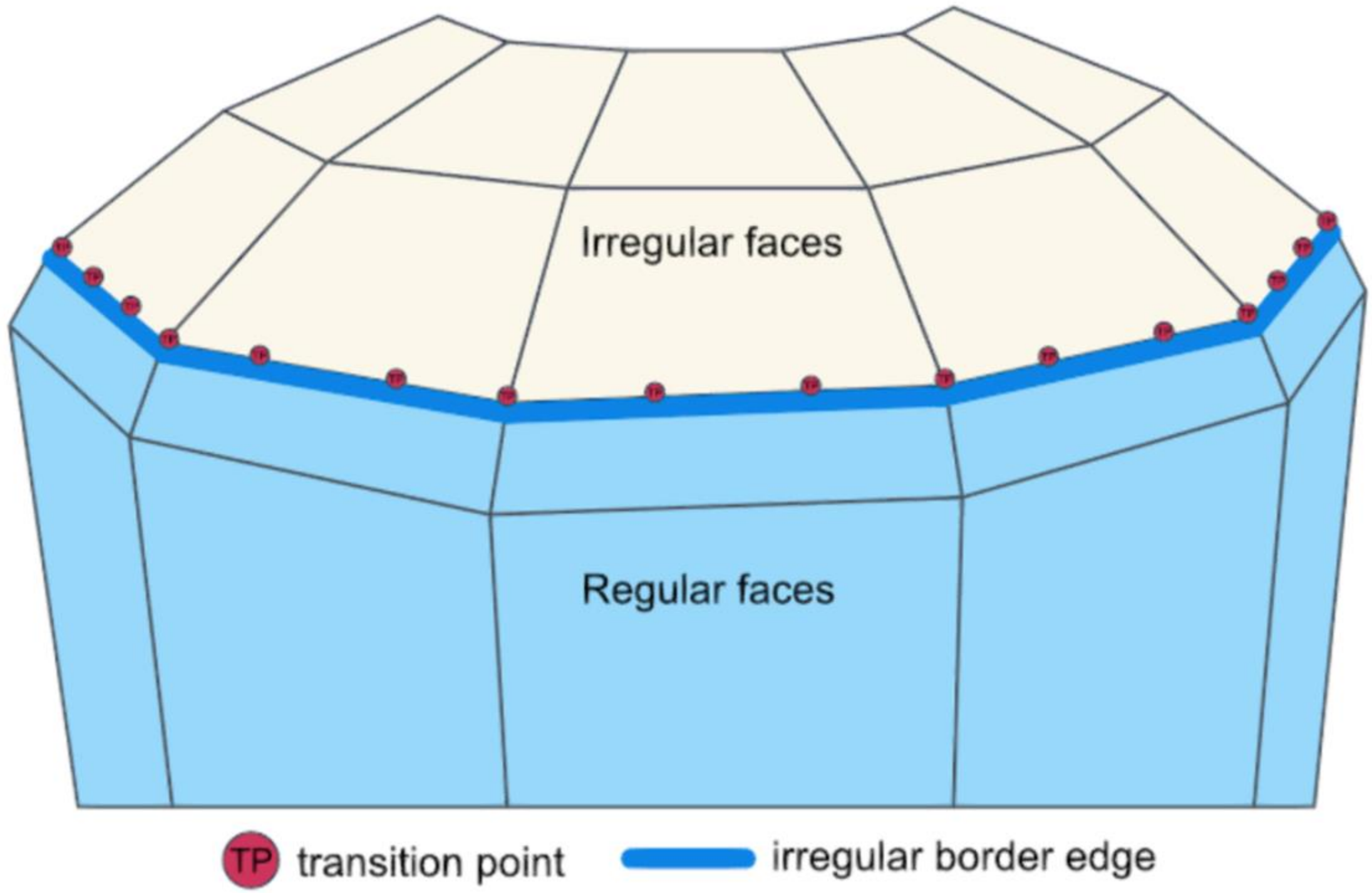
Regular faces rendered by tessellator
Irregular faces rendered as triangles w/compute

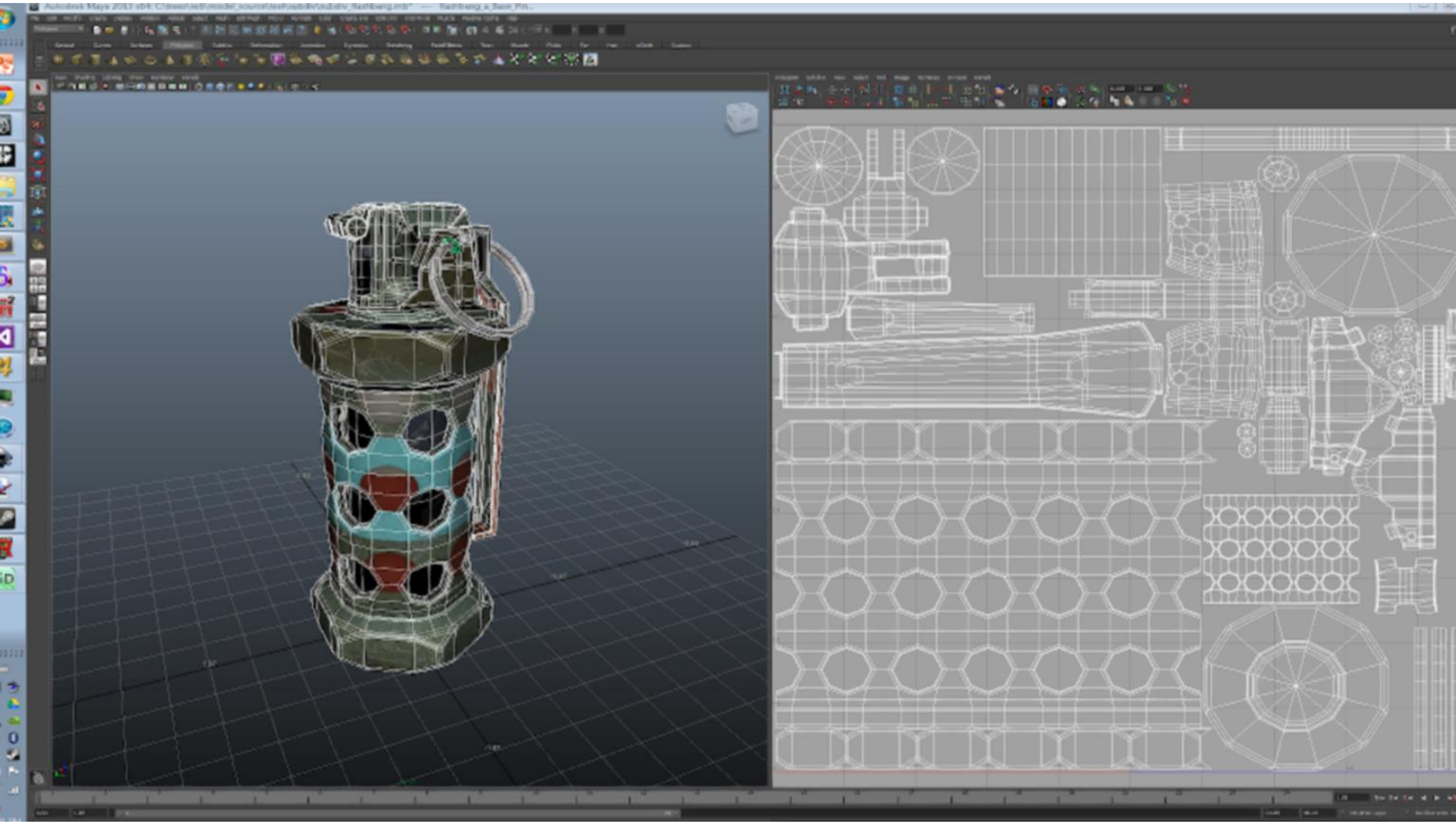


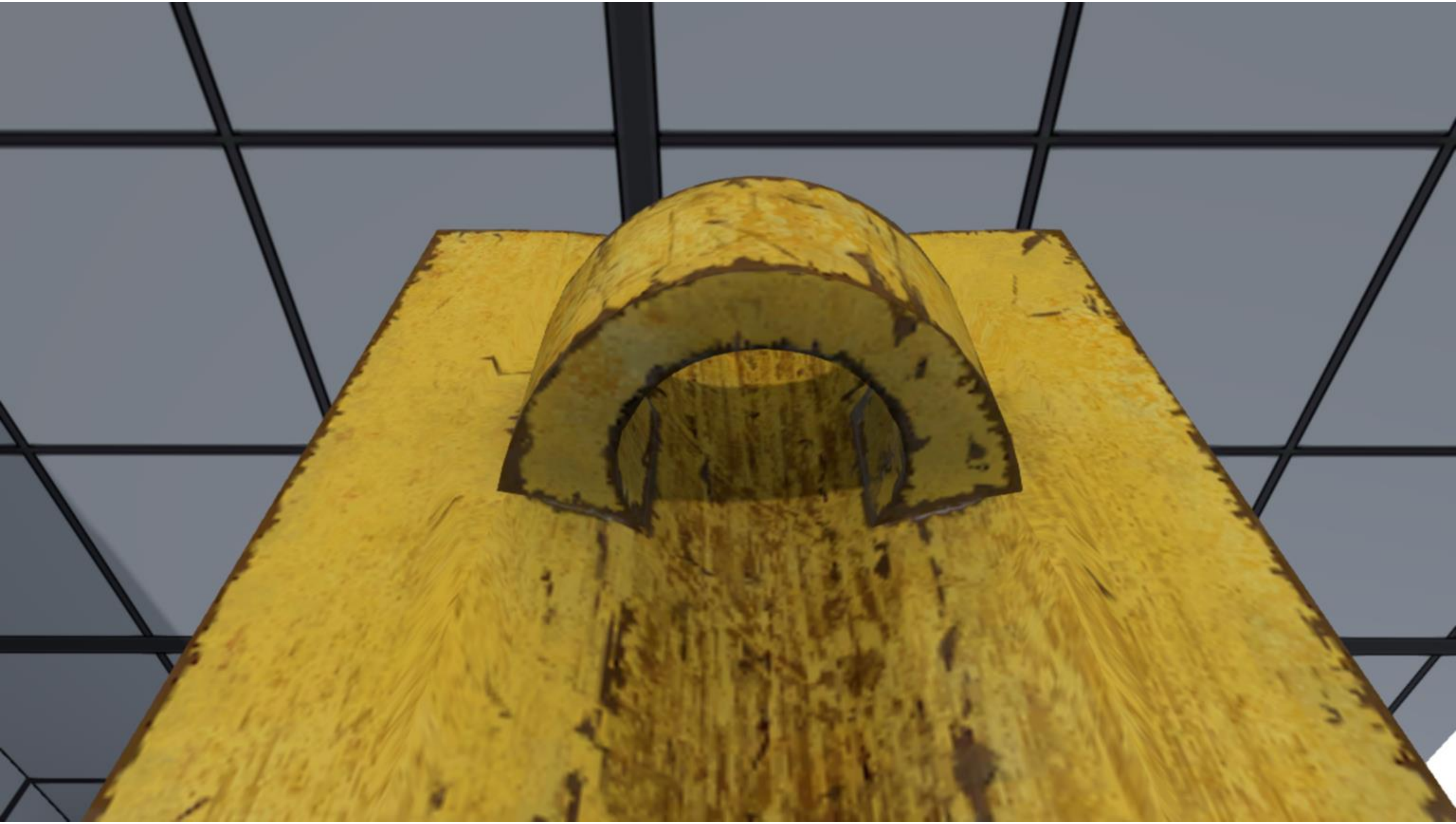


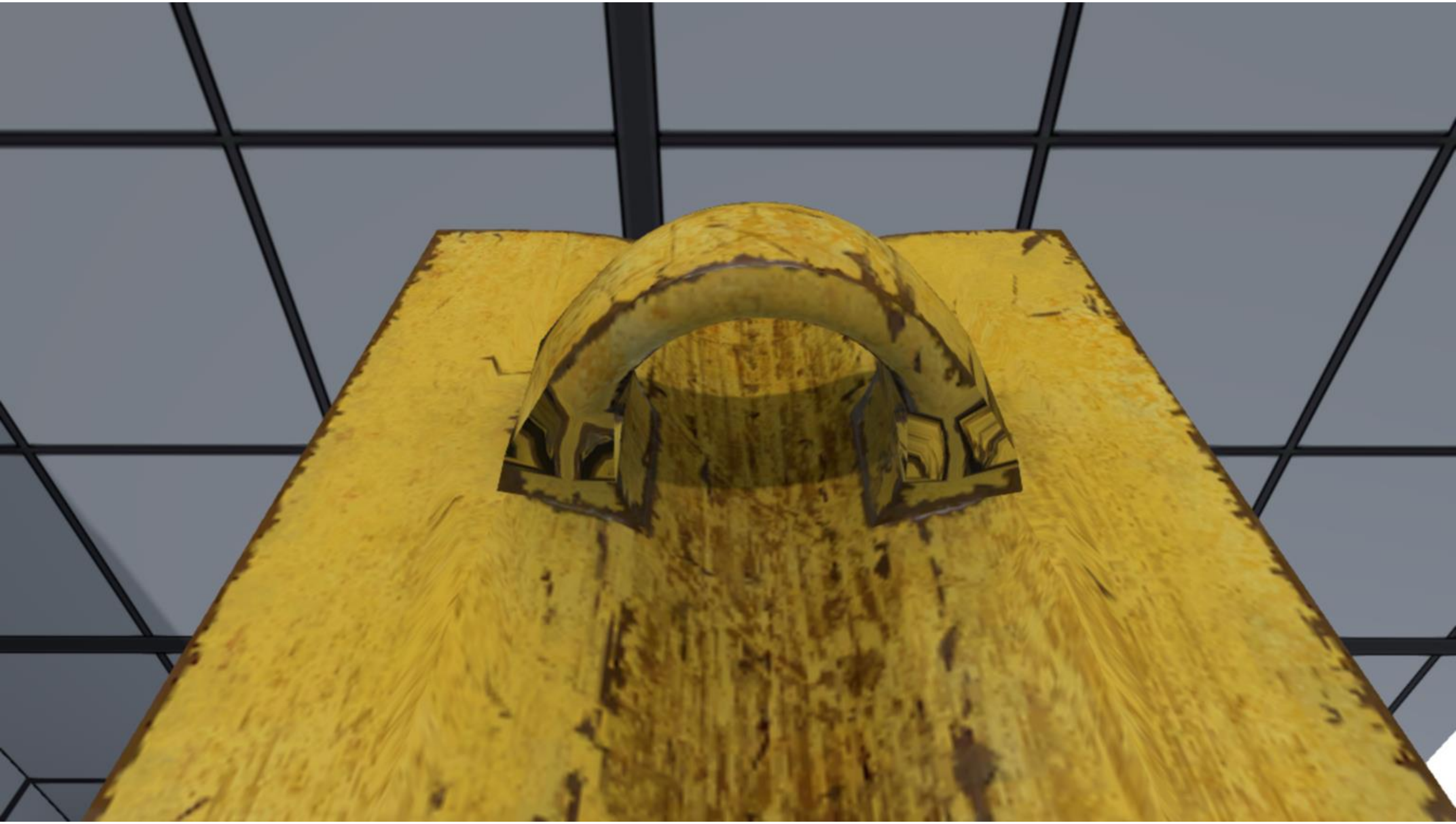


Cracks

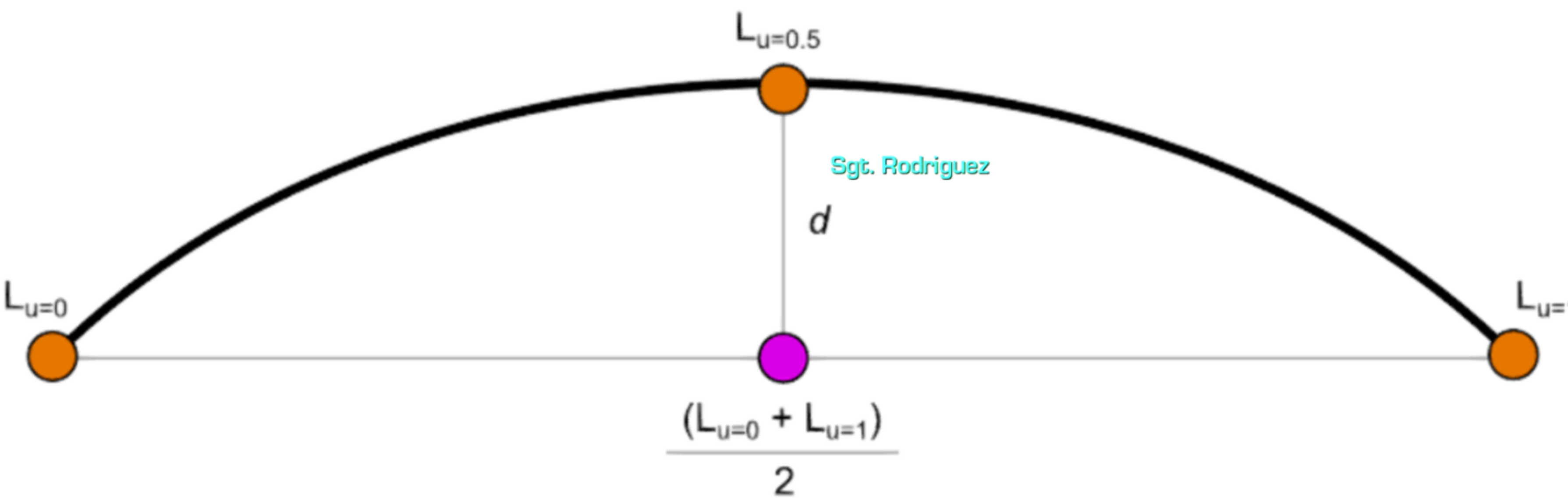

Irregular faces
Regular faces
TP transition point
irregular border edge





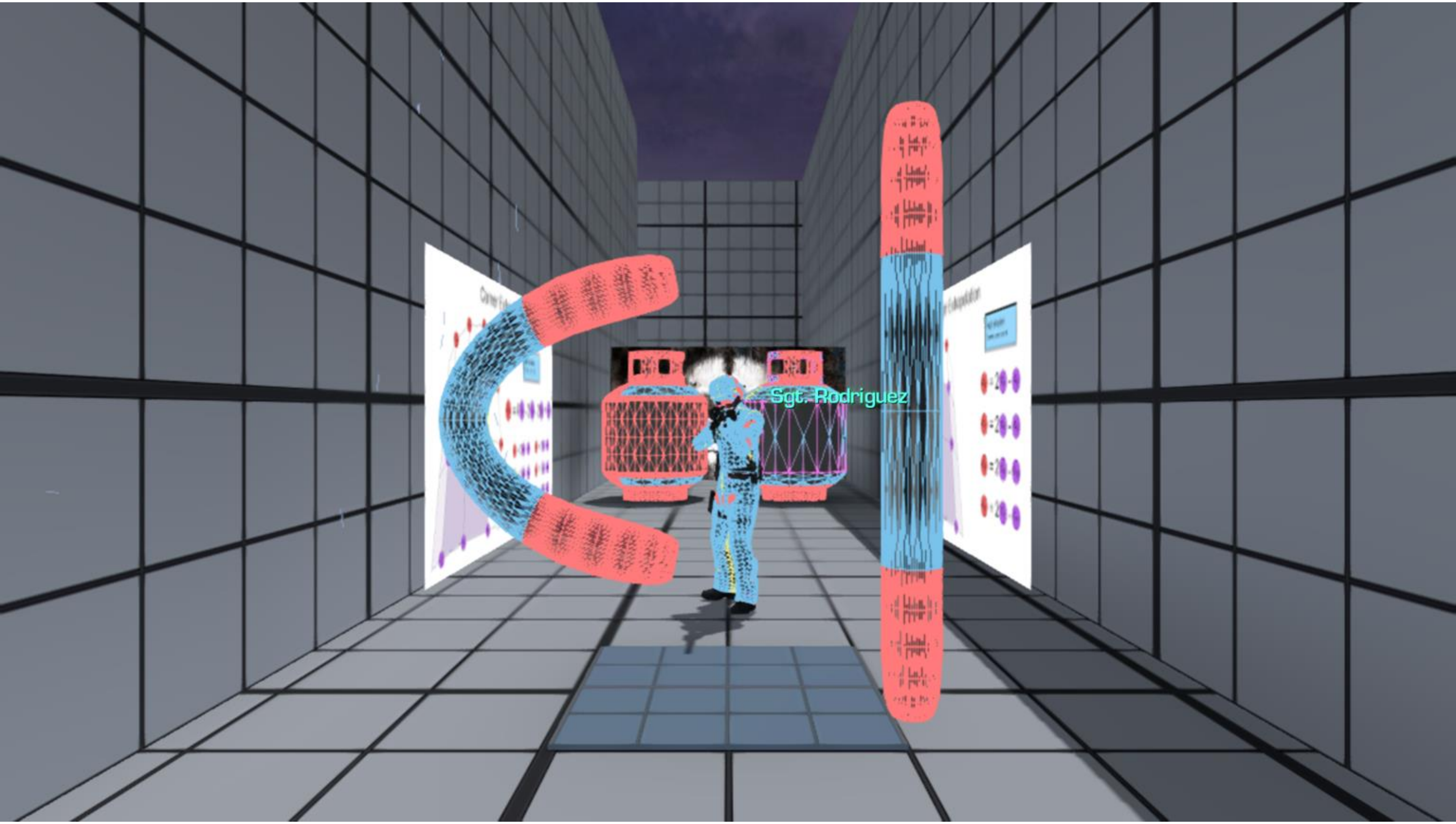


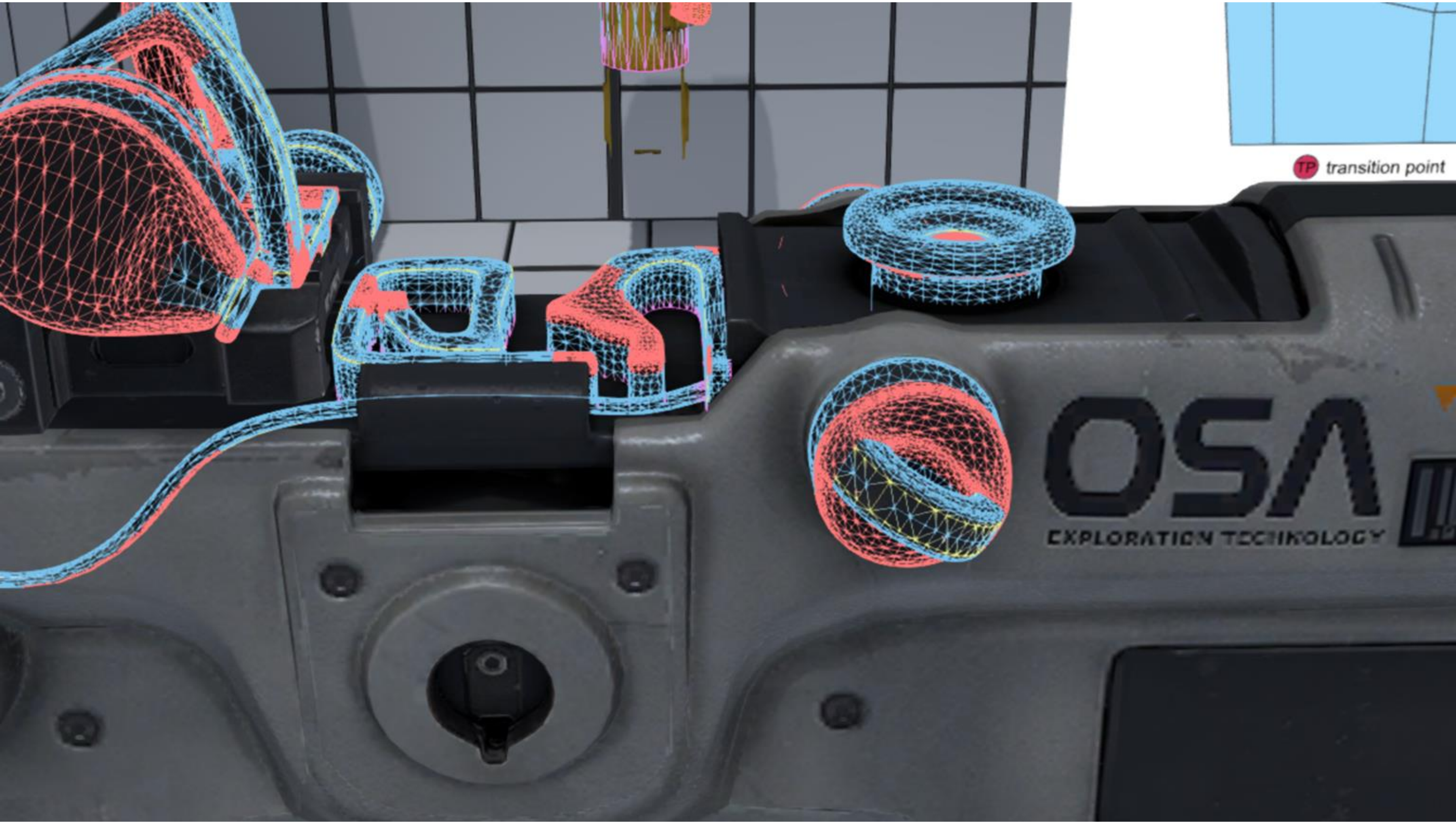
Screen Space Adaptive

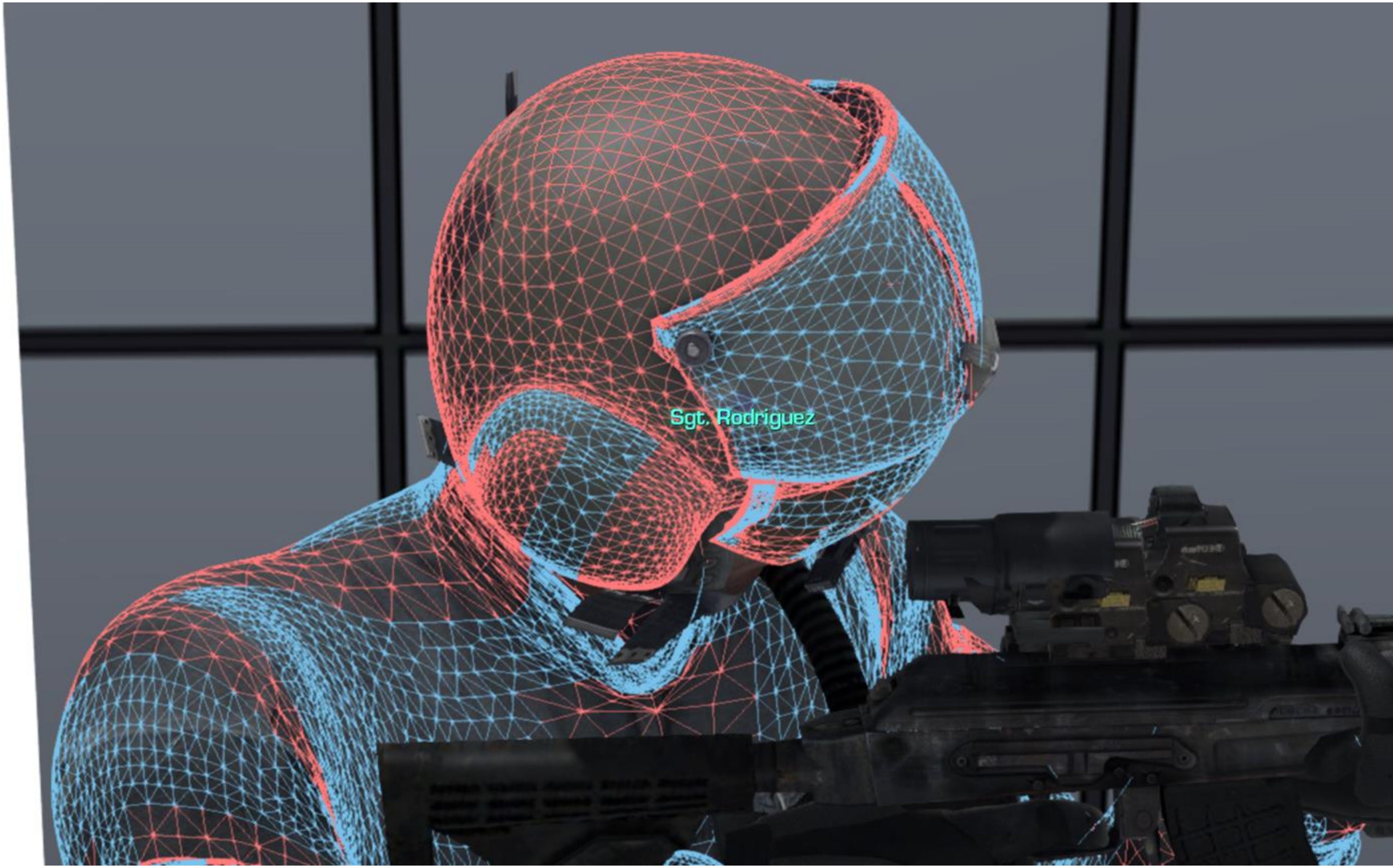


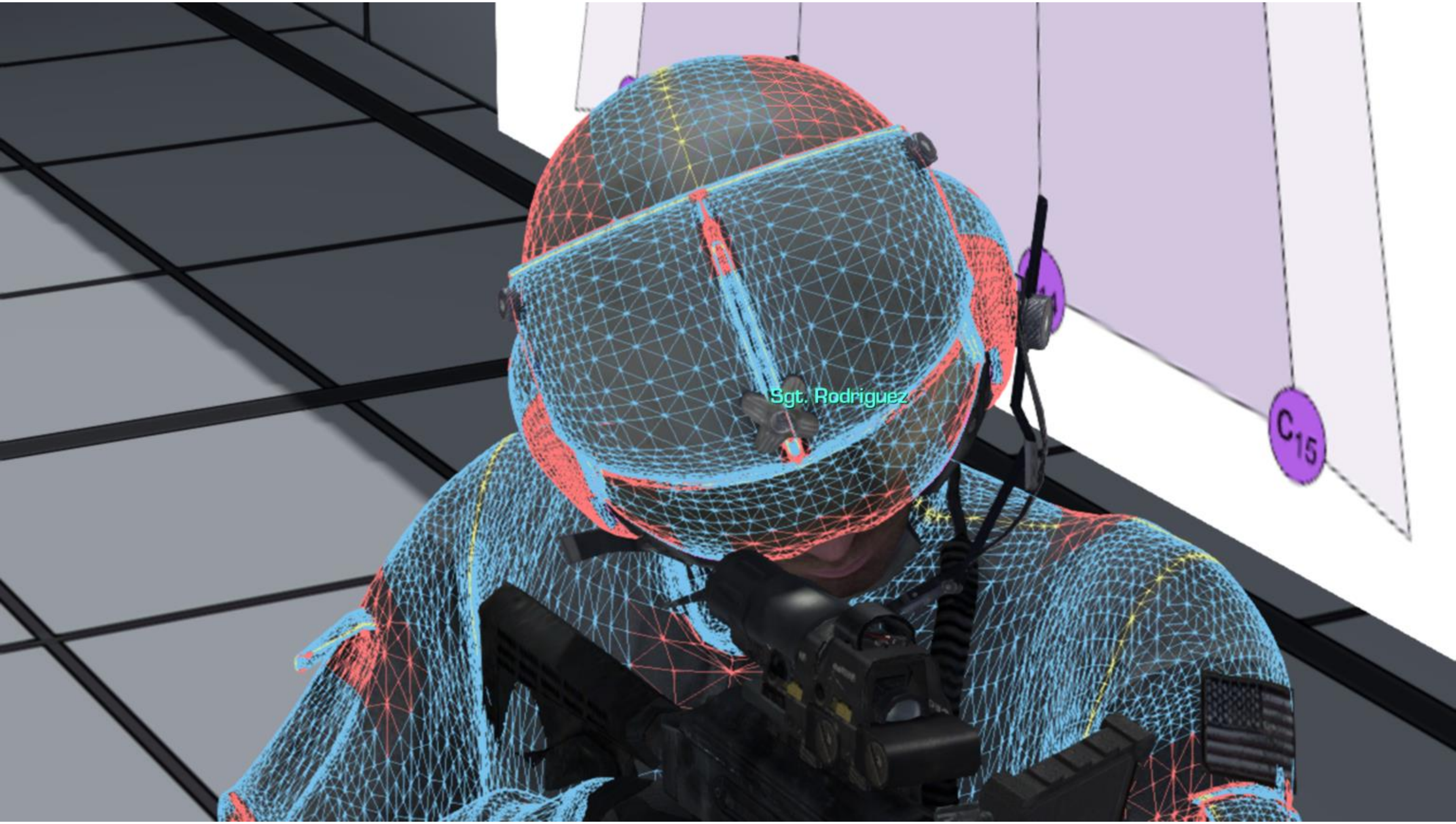
L = projected limit surface

d = curvature metric

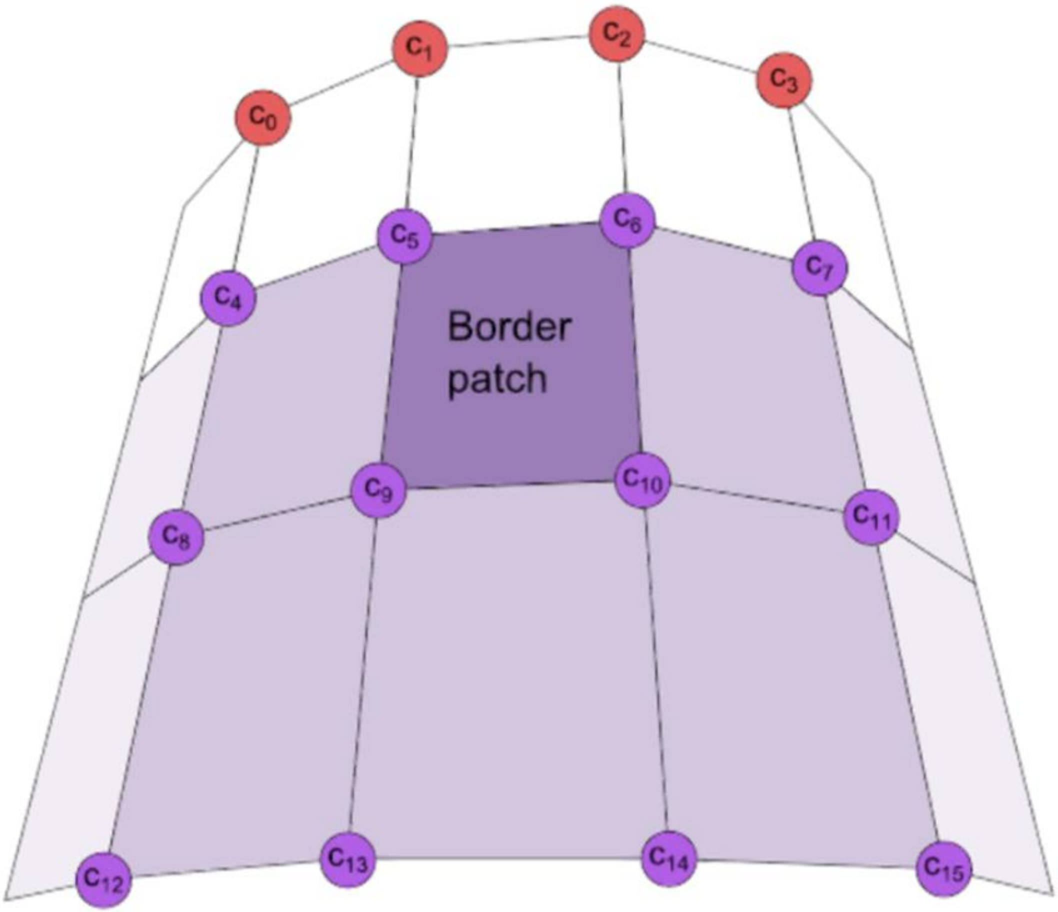

Sgt. Rodriguez


TP transition point
OSA
EXPLORATION TECHNOLOGY


Sgt. Rodriguez


Sgt. Rodriguez
C15

Edge Extrapolation



Hull shader
Tweaks control points

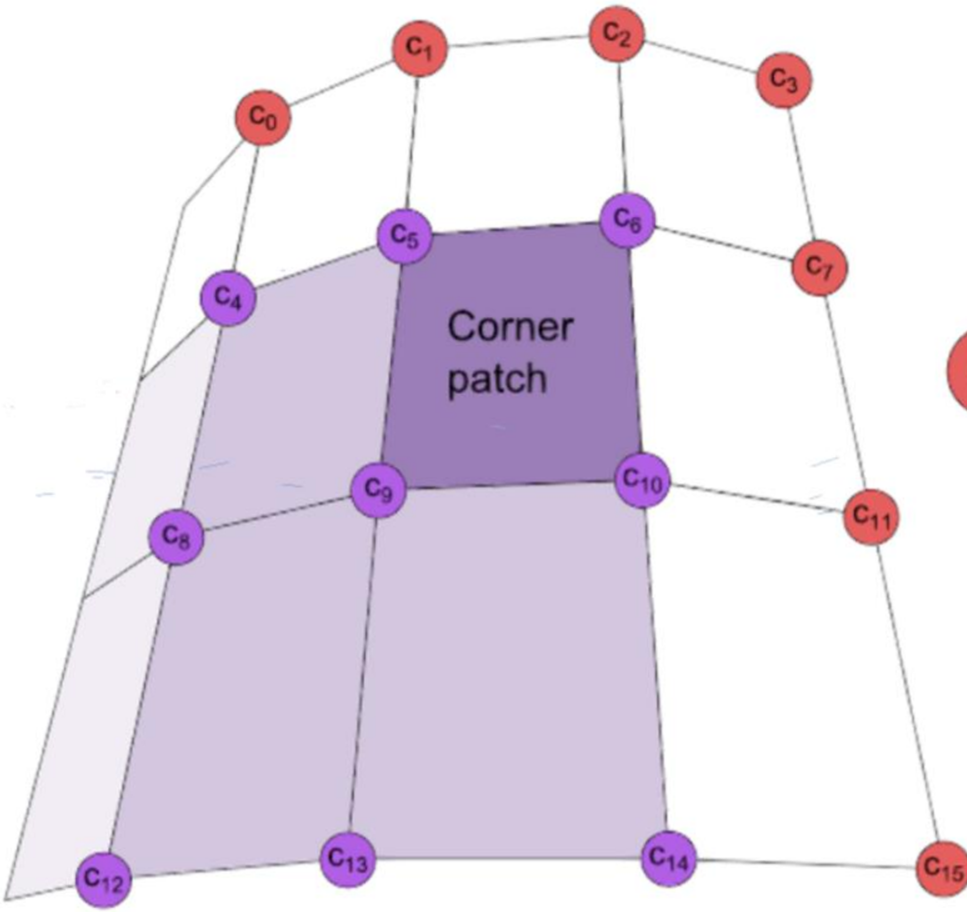
$$c_0 = 2c_4 - c_8$$

$$c_1 = 2c_5 - c_9$$

$$c_2 = 2c_6 - c_{10}$$

$$c_3 = 2c_7 - c_{11}$$

Corner Extrapolation



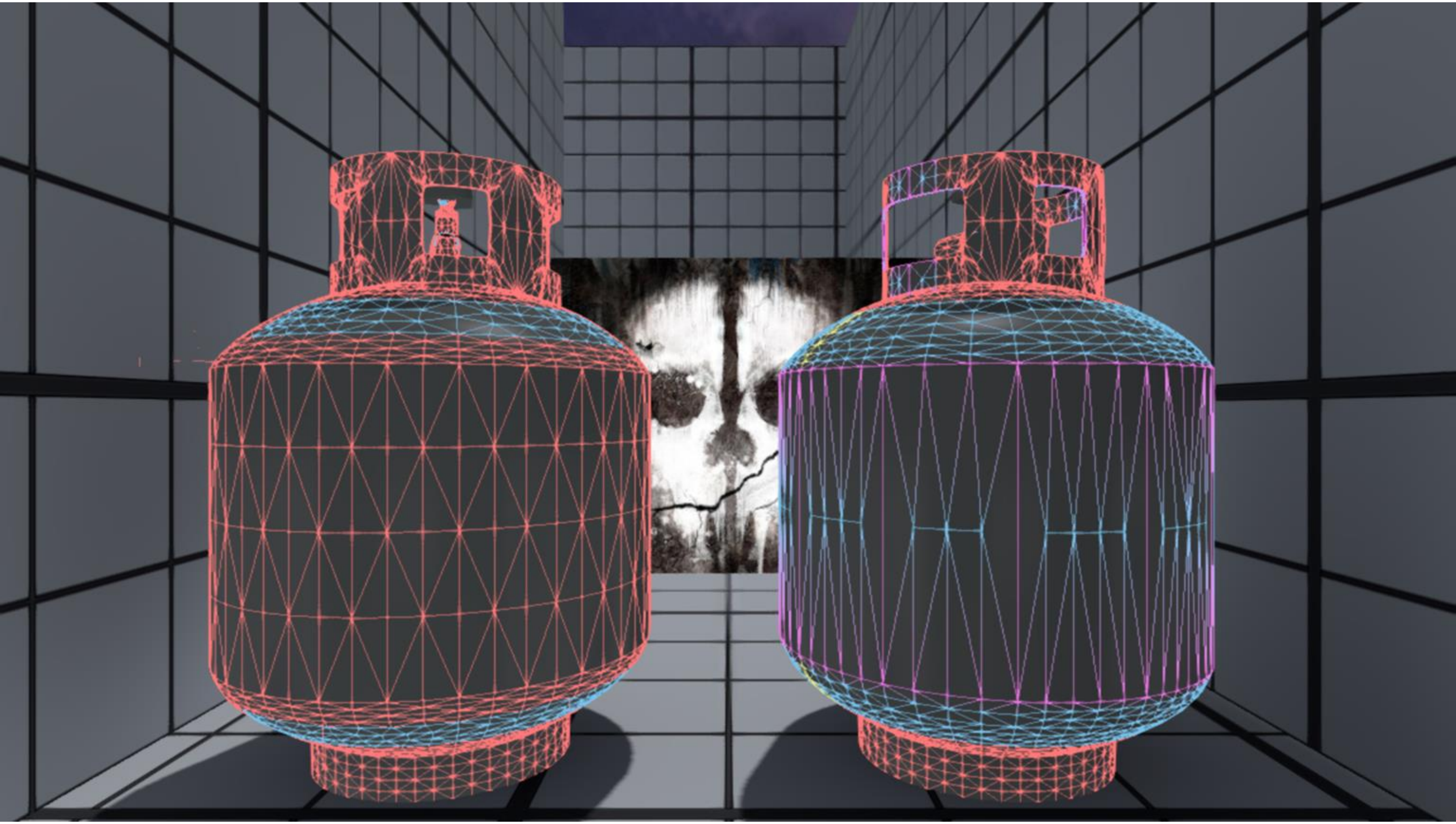
Hull shader
Tweaks control points

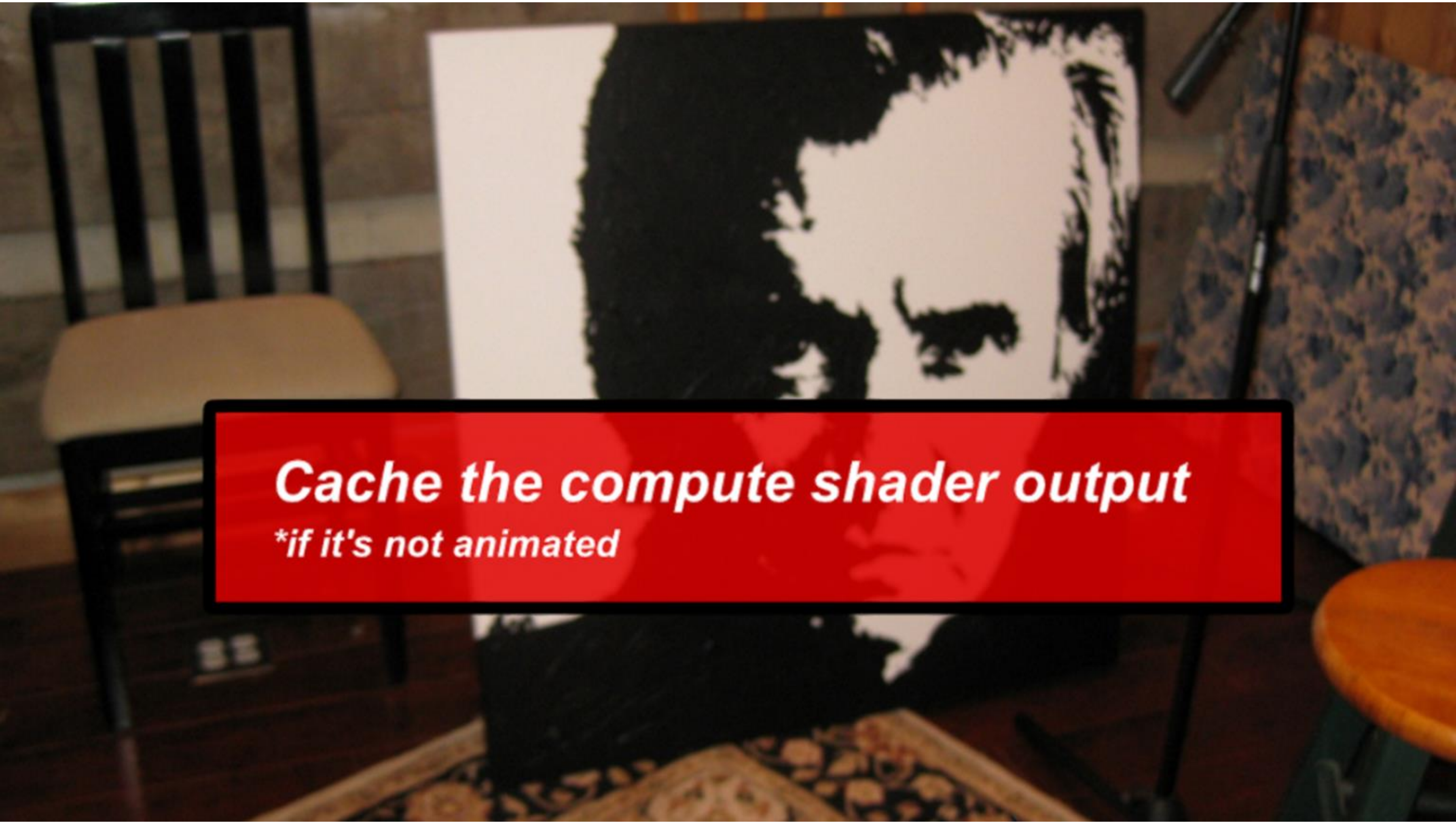
$$c_3 = 4c_6 - 2c_5 - 2c_{10} + c_9$$

$$c_0 = 2c_4 - c_8 \qquad c_7 = 2c_6 - c_5$$

$$c_1 = 2c_5 - c_9 \qquad c_{11} = 2c_{10} - c_9$$

$$c_2 = 2c_6 - c_{10} \qquad c_{15} = 2c_{14} - c_{13}$$

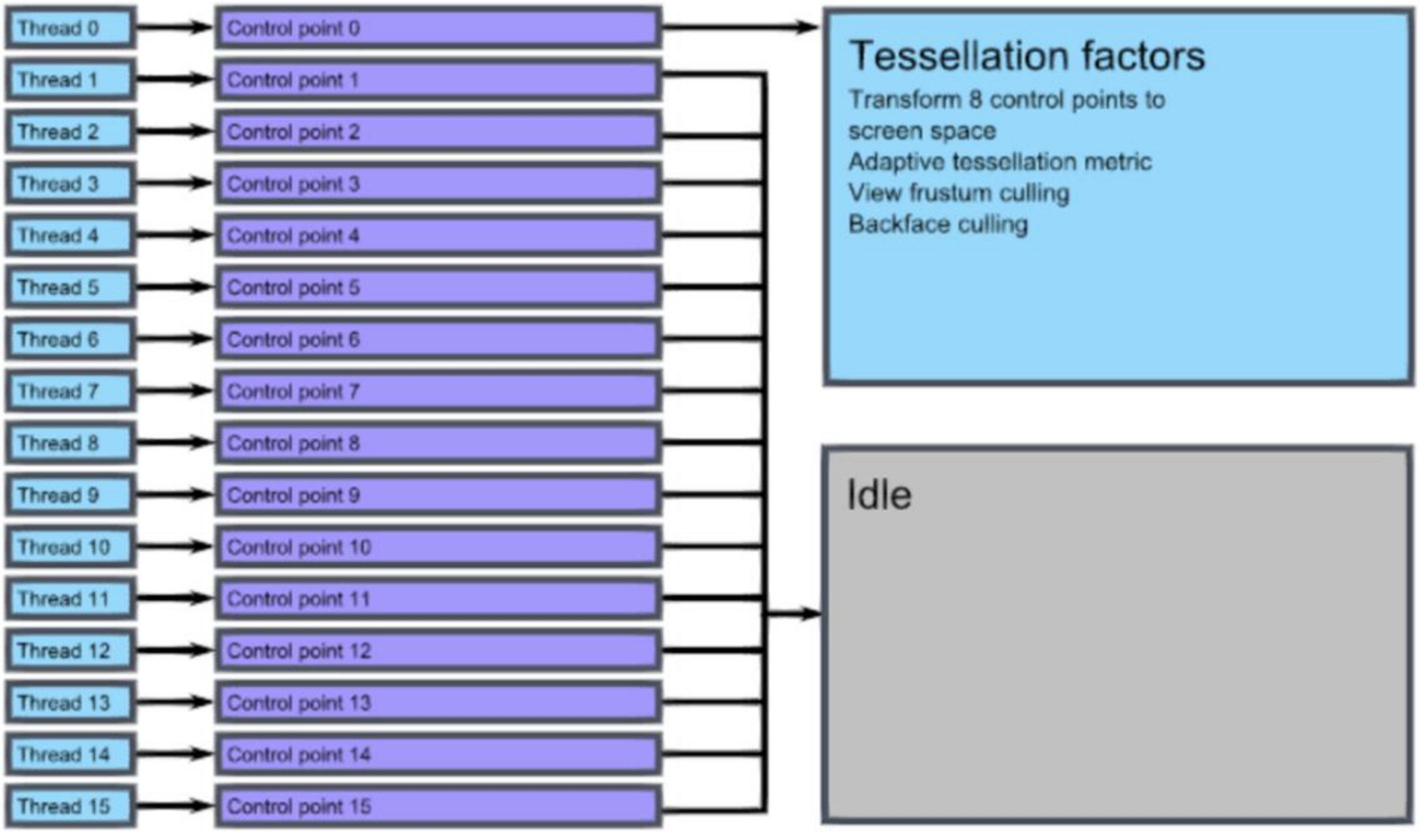



Cache the compute shader output
*if it's not animated

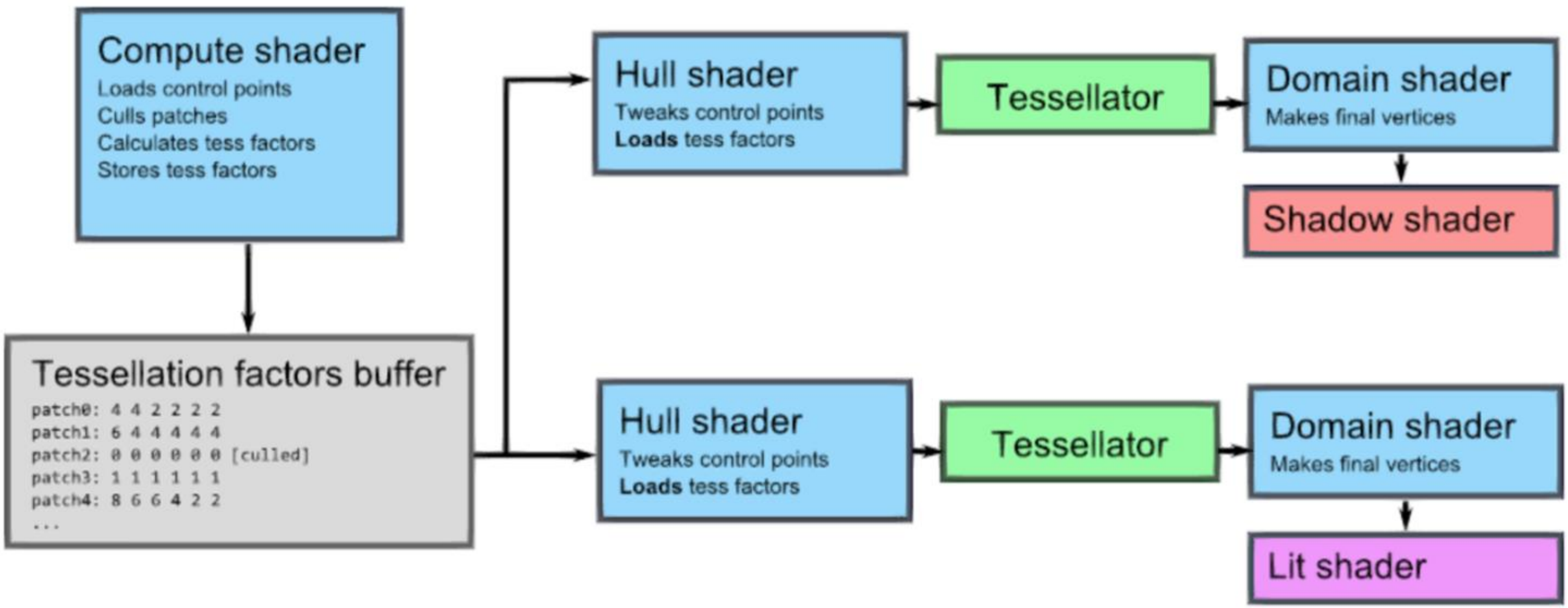

ulling
WARNING: unknown dvar 'r_glowRadius1' in file 'vision/wade_gdc.vision'
WARNING: unknown dvar 'r_glowBloomIntensity1' in file 'vision/wade_gdc.vision'
WARNING: unknown dvar 'r_glowSkyBleedIntensity0' in file 'vision/wade_gdc.vision'
WARNING: unknown dvar 'r_glowSkyBleedIntensity1' in file 'vision/wade_gdc.vision'


output

HS thread flow

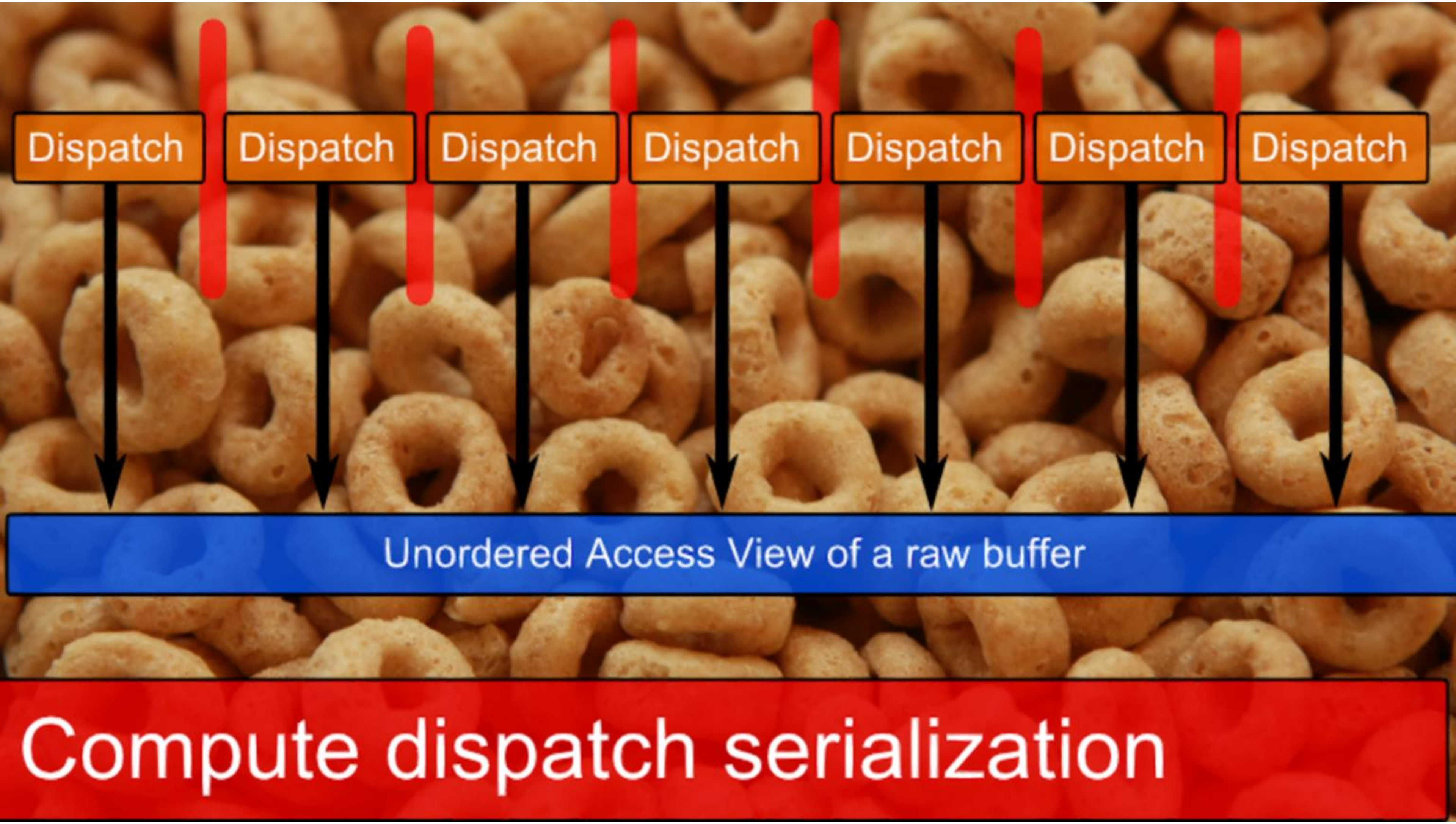


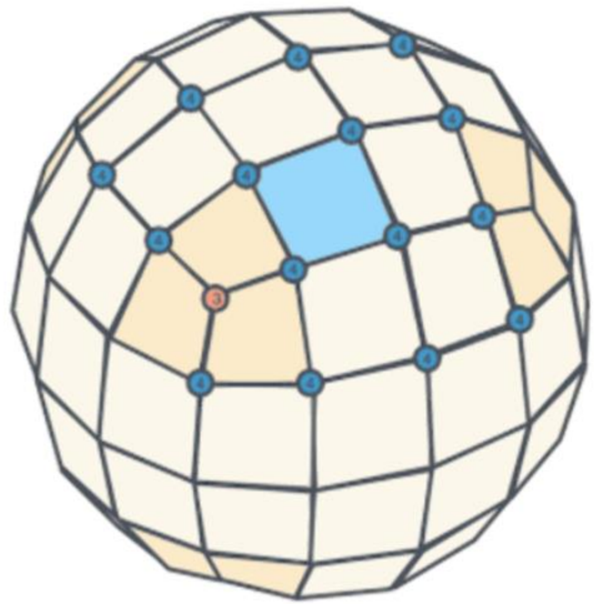
CS as HS

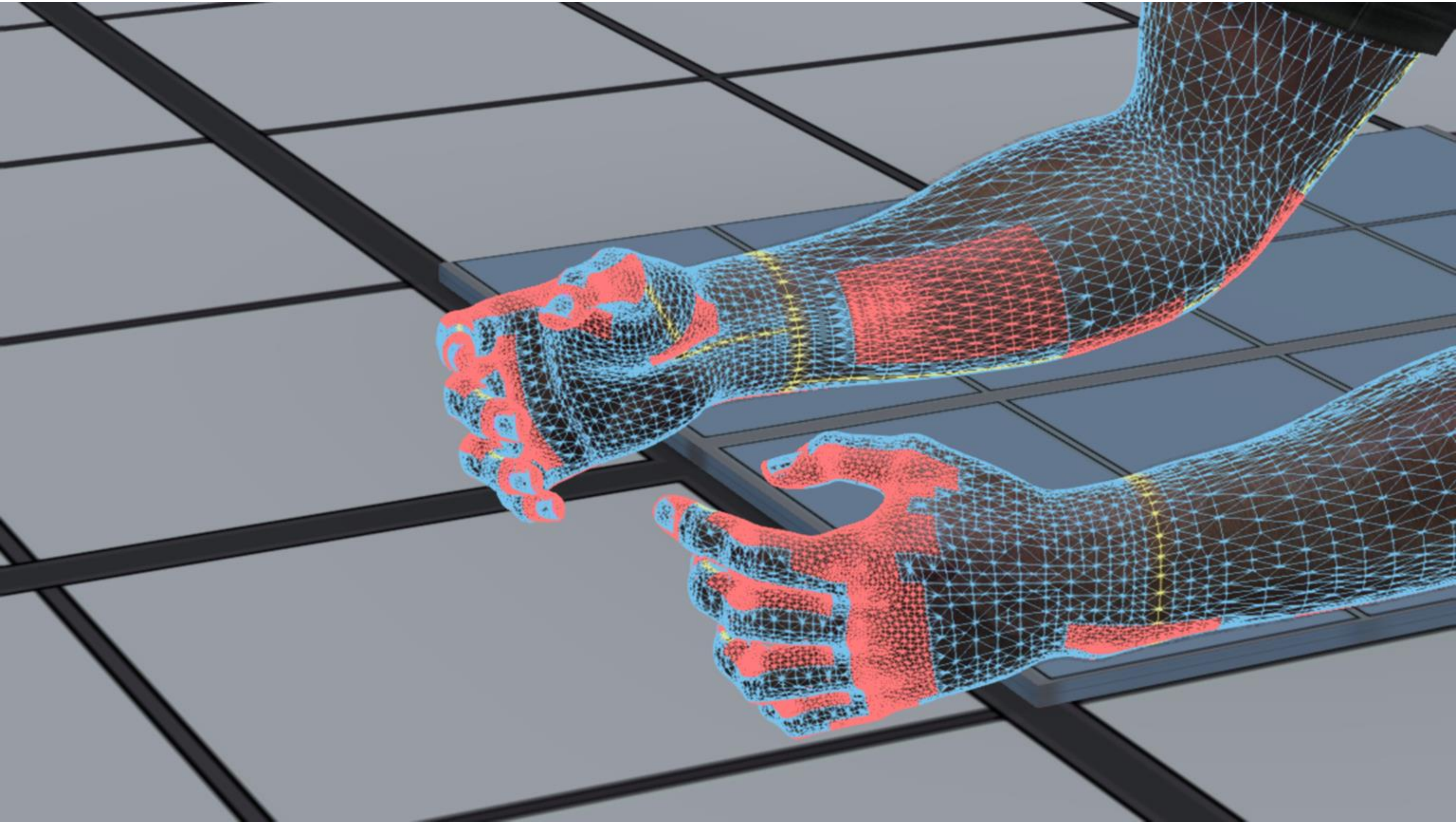


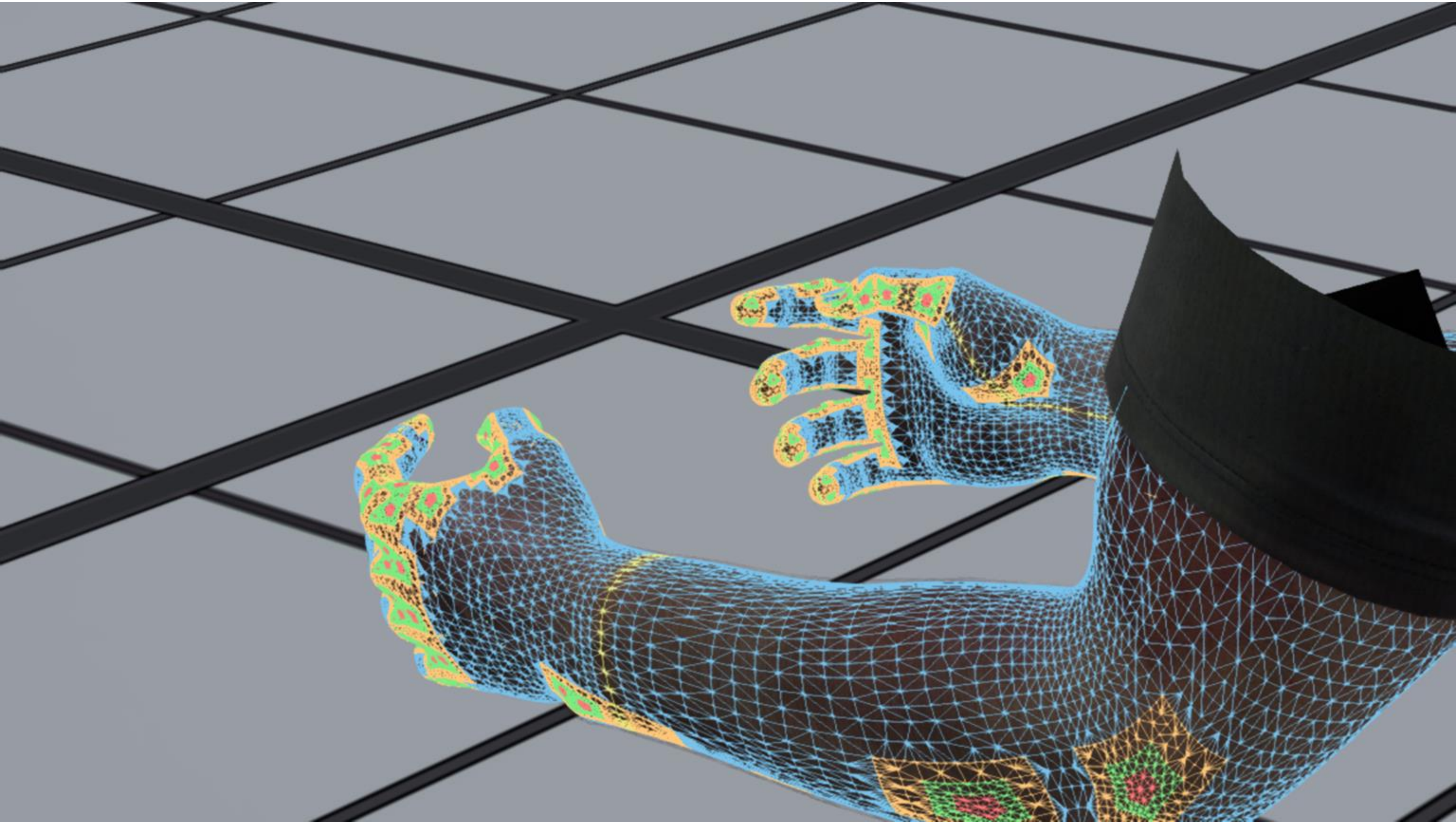
Compute shader
Loads control points
Culls patches
Calculates tess factors
Stores tess factors
Tessellation factors buffer
patch0: 4 4 2 2 2 2
patch1: 6 4 4 4 4 4
patch2: 0 0 0 0 0 0 [culled]
patch3: 1 1 1 1 1 1
patch4: 8 6 6 4 2 2
...
Hull shader
Tweaks control points
Loads tess factors
Tessellator
Domain shader
Makes final vertices
Shadow shader
Hull shader
Tweaks control points
Loads tess factors
Tessellator
Domain shader
Makes final vertices
Lit shader

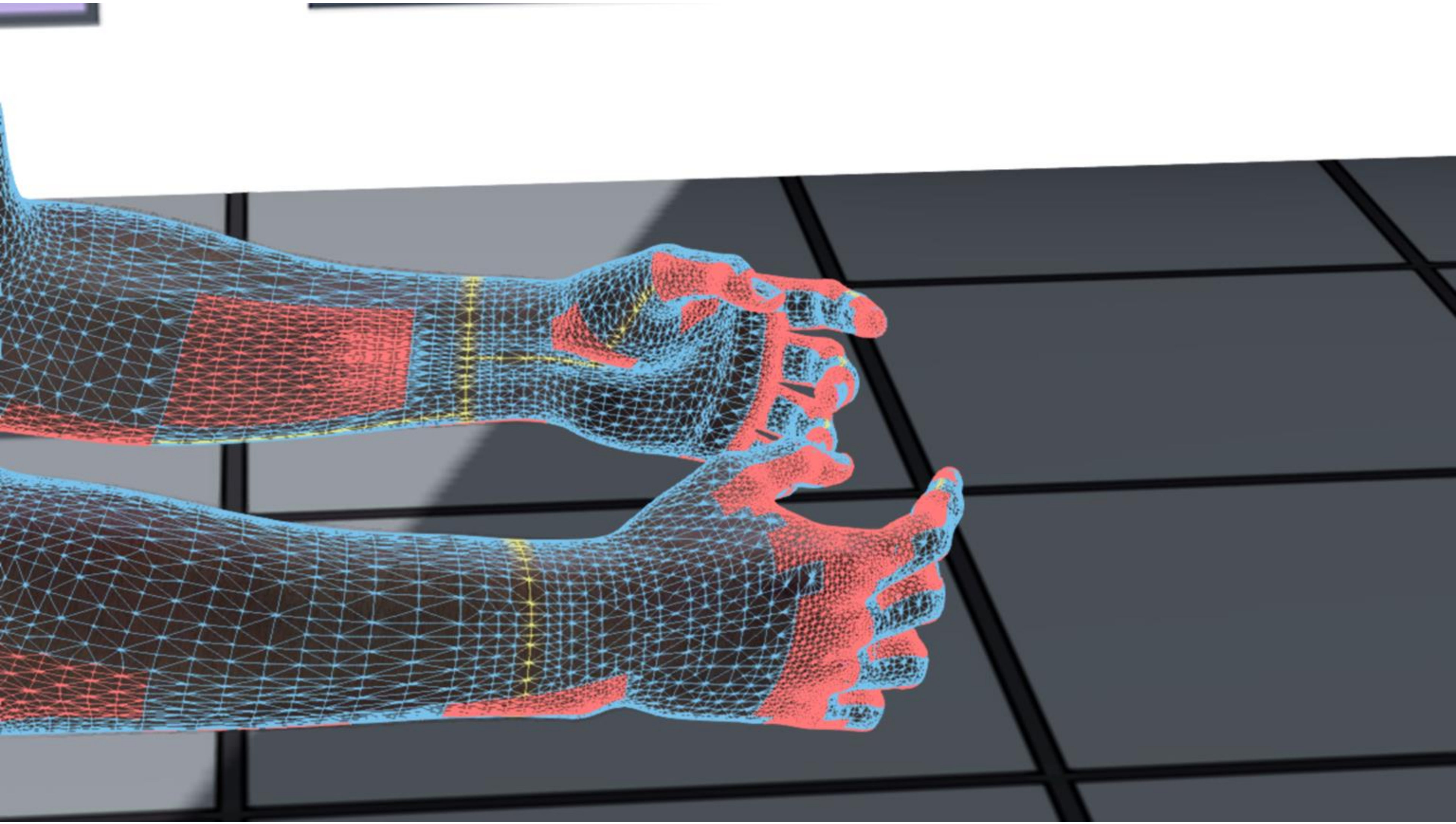
CS thread flow

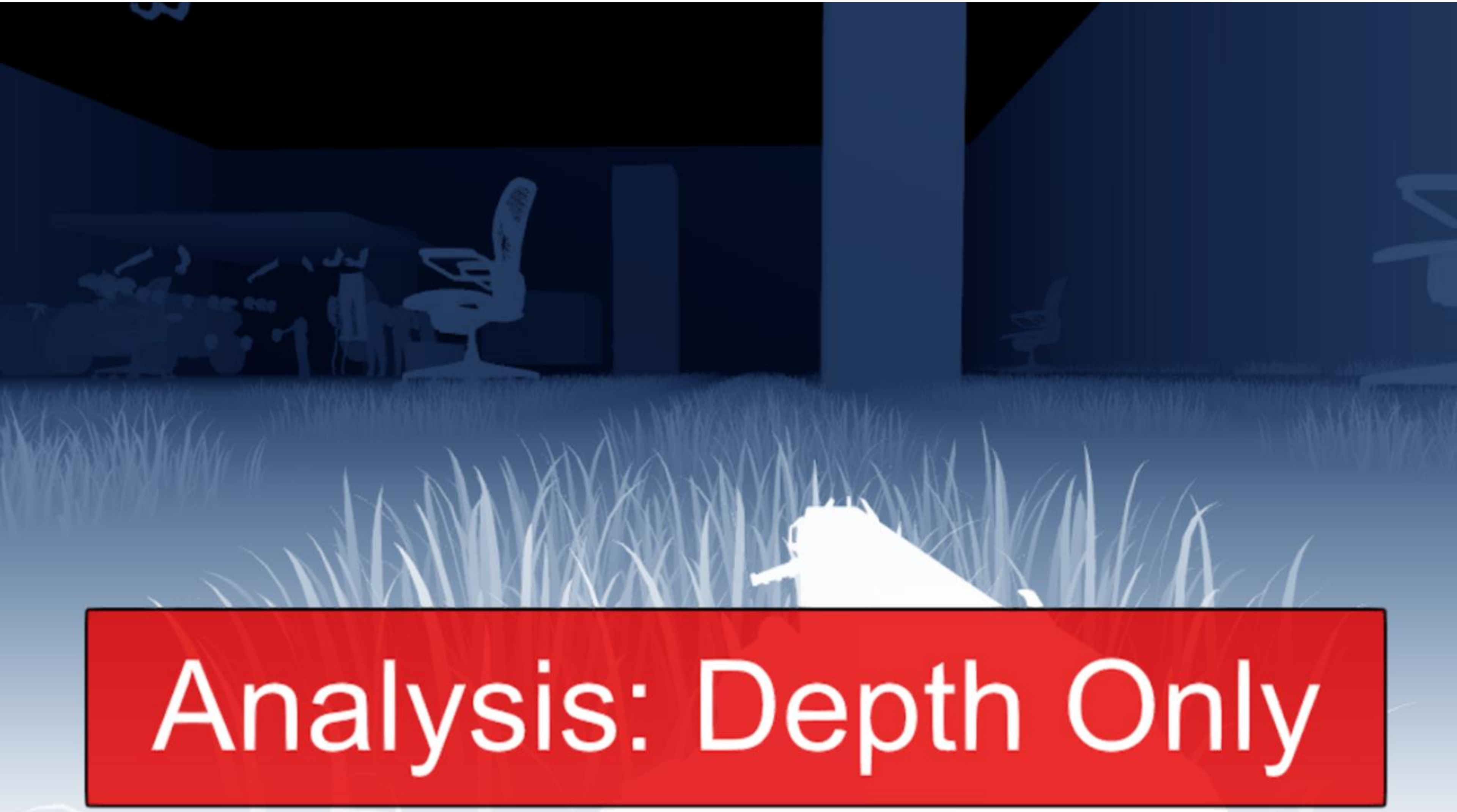

Dispatch
Dispatch
Dispatch
Dispatch
Dispatch
Dispatch
Dispatch
Unordered Access View of a raw buffer
Compute dispatch serialization







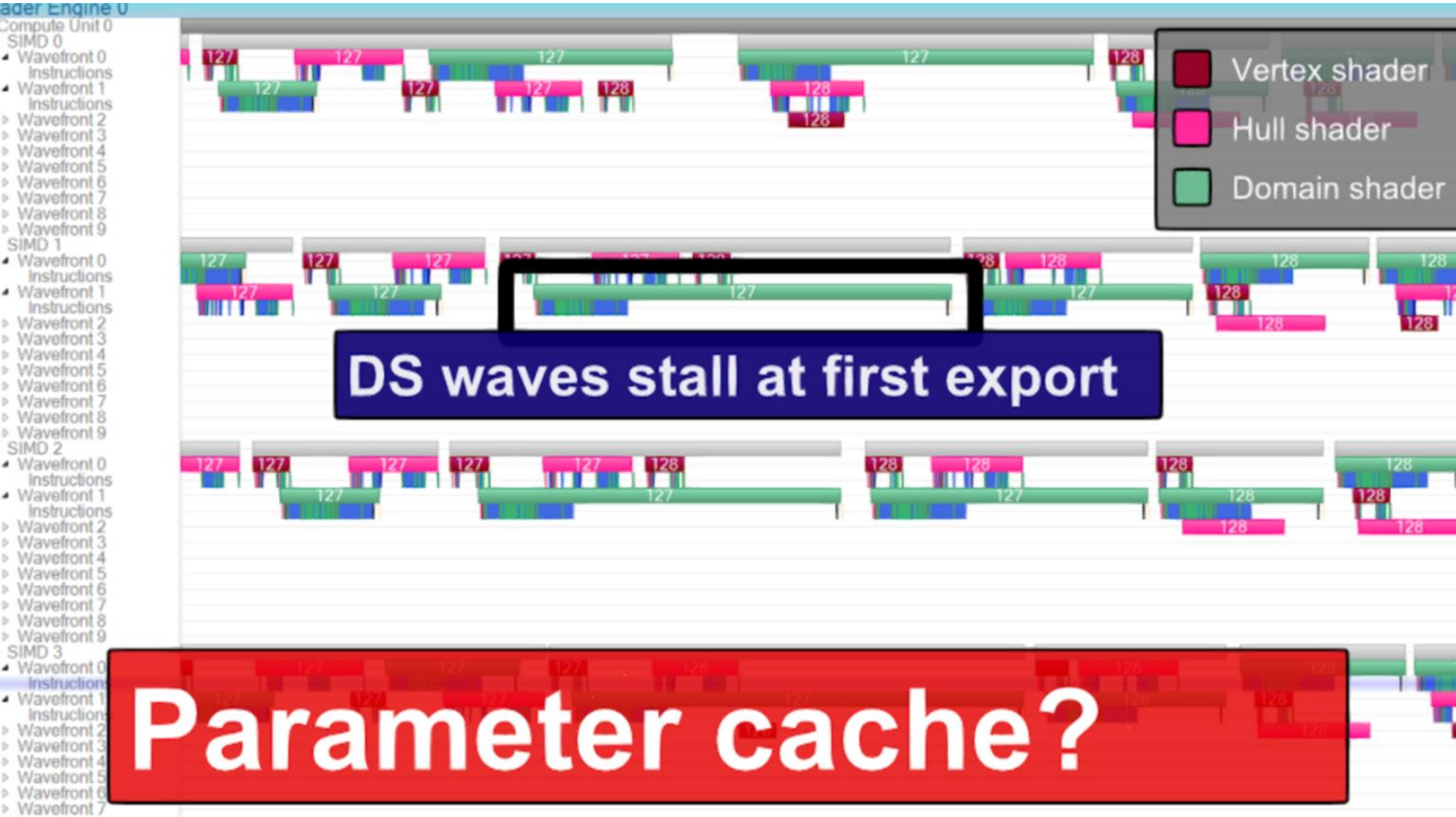




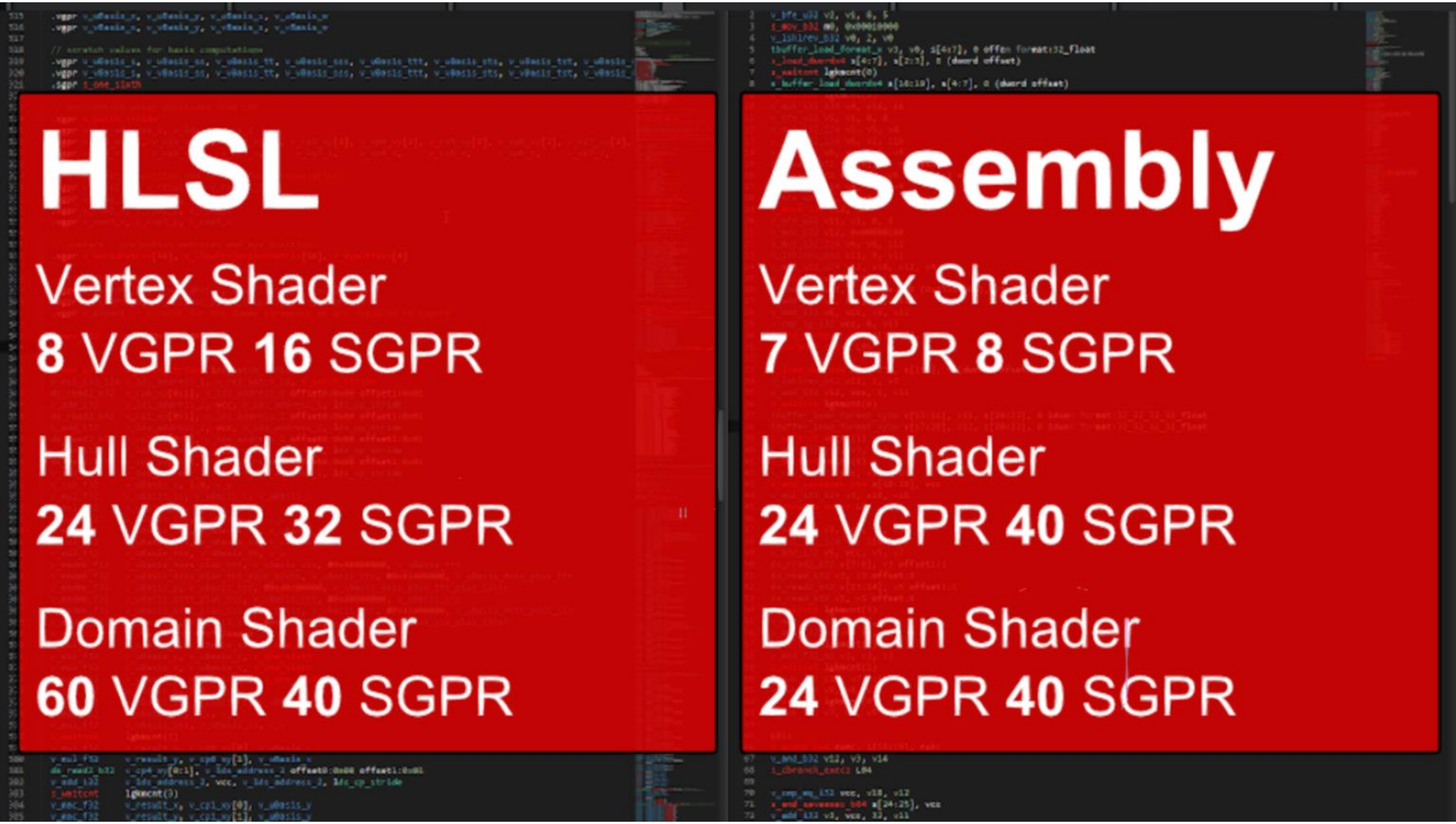
Analysis: Depth Only


Bottleneck mitigation


Vertex shader
Hull shader
Domain shader
Low utilization



Compute Unit 0
SIMD 0
Wavefront 0
Instructions
Wavefront 1
Instructions
Wavefront 2
Wavefront 3
Wavefront 4
Wavefront 5
Wavefront 6
Wavefront 7
Wavefront 8
Wavefront 9
SIMD 1
SIMD 2
SIMD 3
Vertex shader
Hull shader
Domain shader
DS waves stall at first export
Parameter cache?


HLSL
Vertex Shader
8 VGPR 16 SGPR
Hull Shader
24 VGPR 32 SGPR
Domain Shader
60 VGPR 40 SGPR
Assembly
Vertex Shader
7 VGPR 8 SGPR
Hull Shader
24 VGPR 40 SGPR
Domain Shader
24 VGPR 40 SGPR

HLSL

Domain Shader

60 VGPR

4 waves per SIMD

Assembly

Domain Shader

24 VGPR

10 waves per SIMD

GCN VGPR Count	<=24	28	32	36	40	48	64	84	<= 128	> 128
Max Waves/SIMD	10 ☺	9	8	7	6	5	4	3	2 ☹	1 ☹

table by Layla Mah, AMD


Shader Engine 0
Compute Unit 0
SIMD 0
SIMD 1
SIMD 2
SIMD 3
Vertex shader
Hull shader
Domain shader
Assembly utilization

Wave launch rate

Shader Engine

SIMD SIMD SIMD SIMD | SIMD SIMD SIMD SIMD | SIMD SIMD SIMD SIMD | SIMD SIMD SIMD SIMD | SIMD SIMD SIMD SIMD | SIMD SIMD SIMD SIMD | SIMD SIMD SIMD SIMD | SIMD SIMD SIMD SIMD | SIMD SIMD SIMD SIMD

VS/HS/DS wave rate is 1 CP / clock

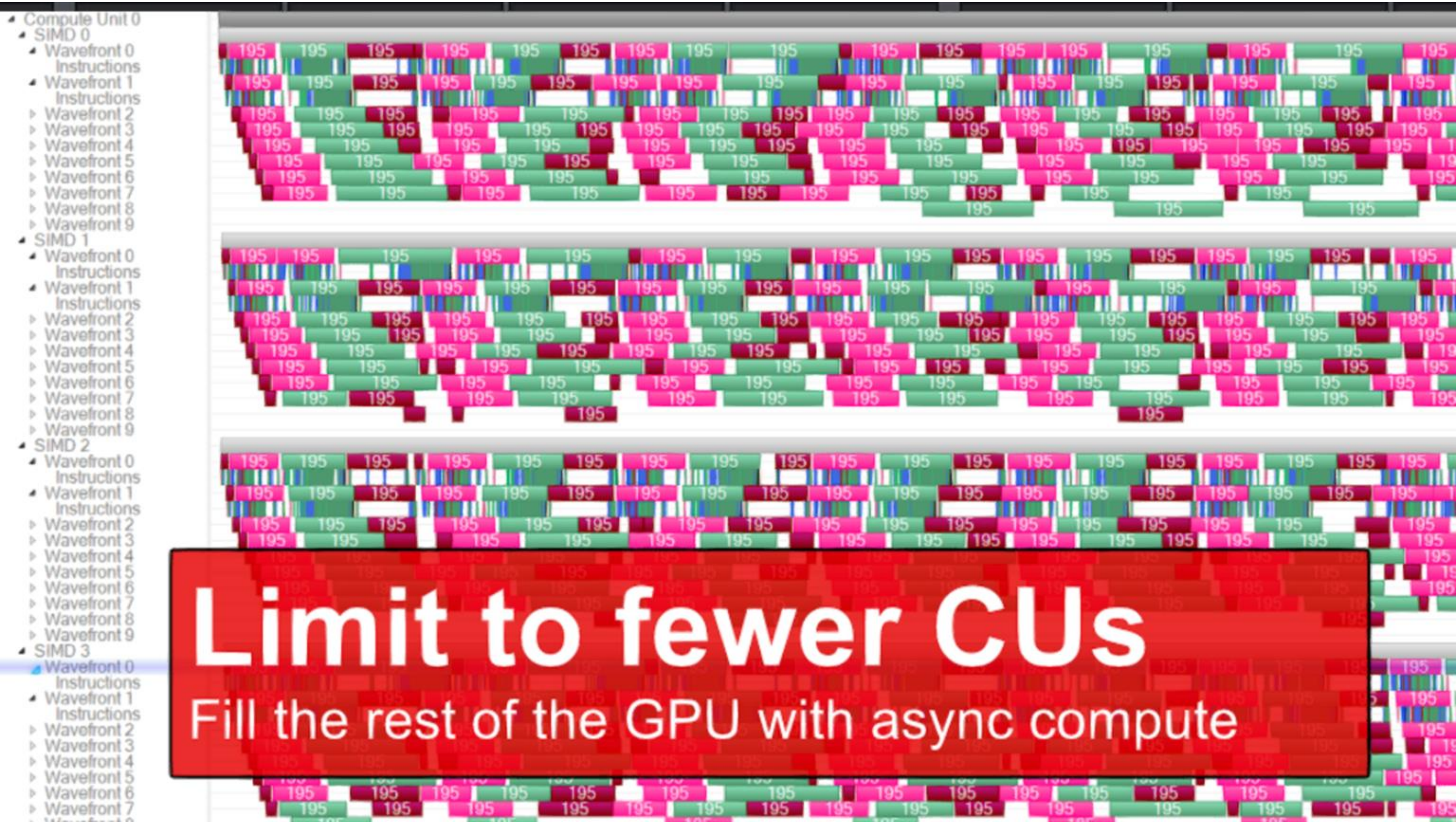
36 SIMDs @ 64 clocks / wave

64 clocks / wave

2304 clocks latency

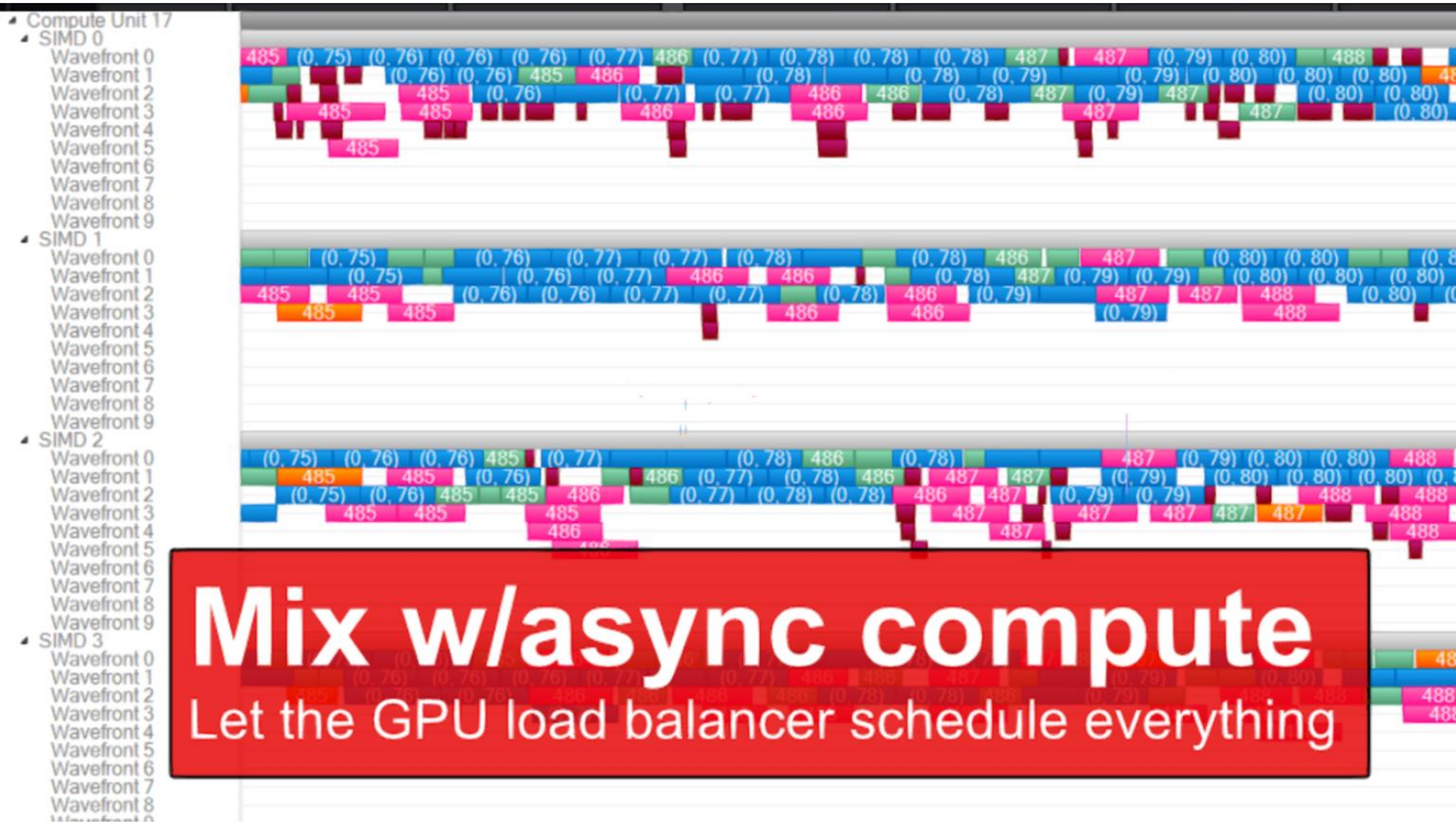
Our waves run in **< 2000** clocks

Occupancy limit is 1



Limit to fewer CUs

Fill the rest of the GPU with async compute



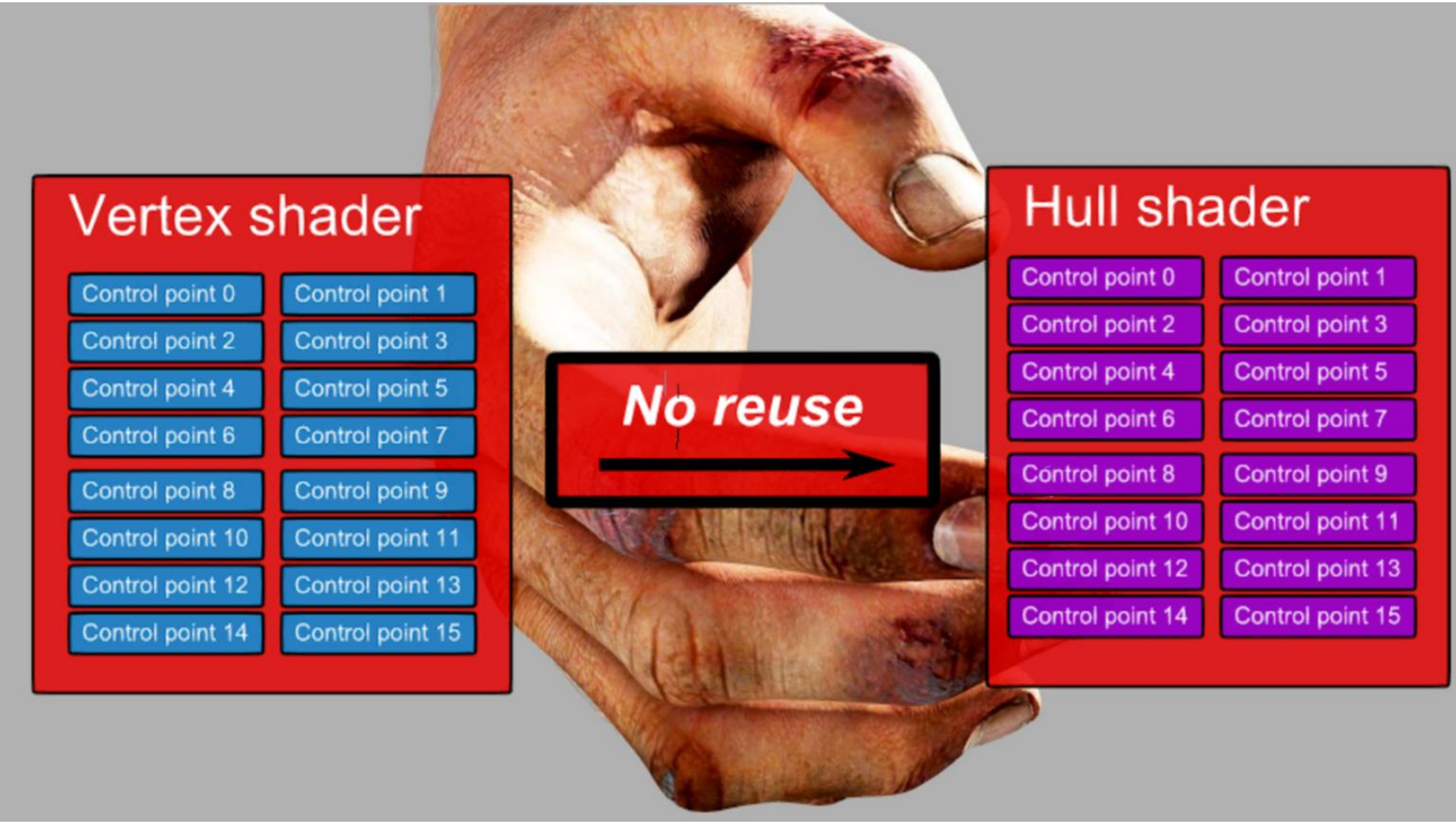
Mix w/async compute

Let the GPU load balancer schedule everything

Async compute

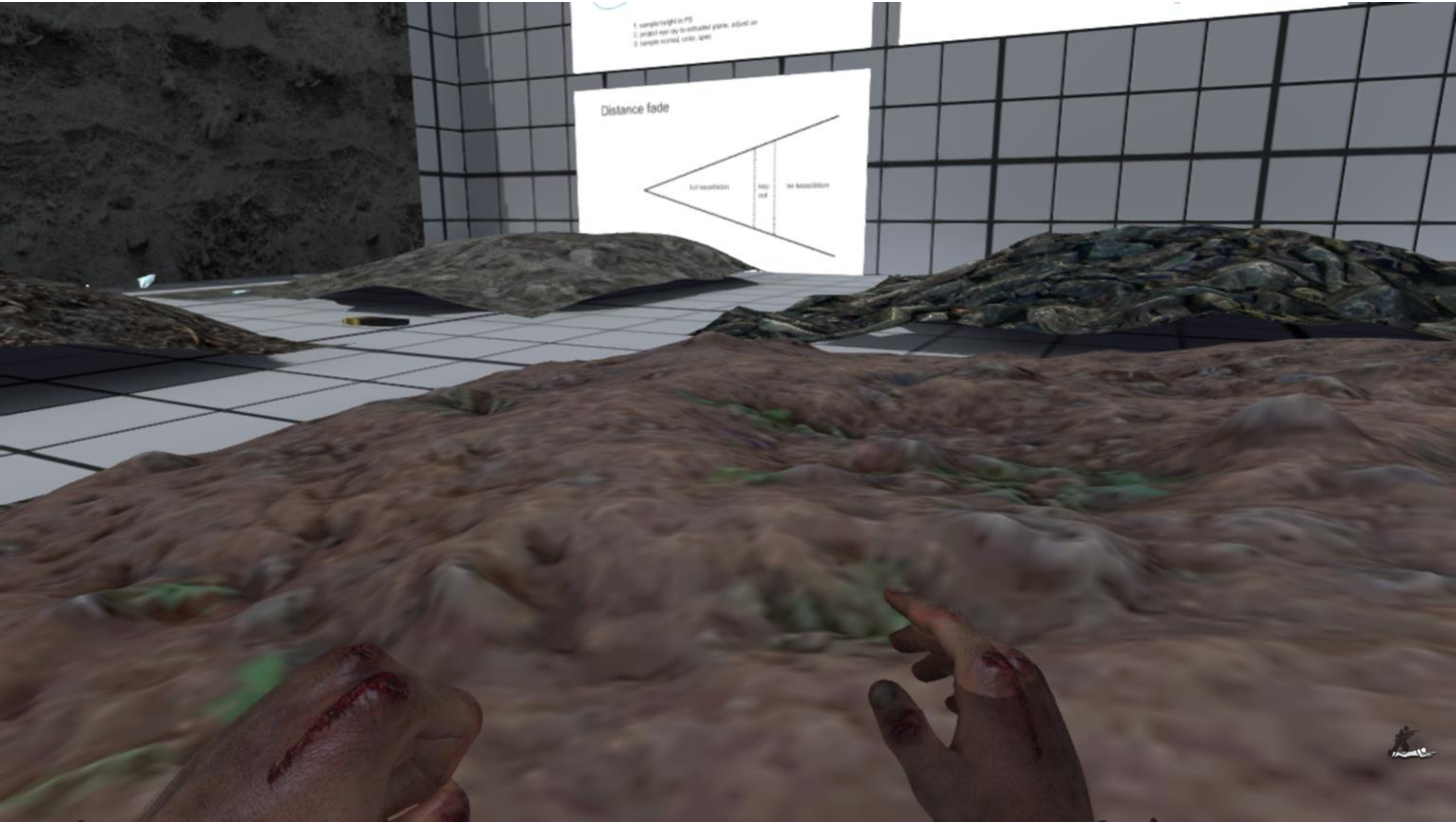
- Skinning
- Tension mapping
- Blend shapes
- Hull shaders
- Ambient occlusion
- Depth decompress

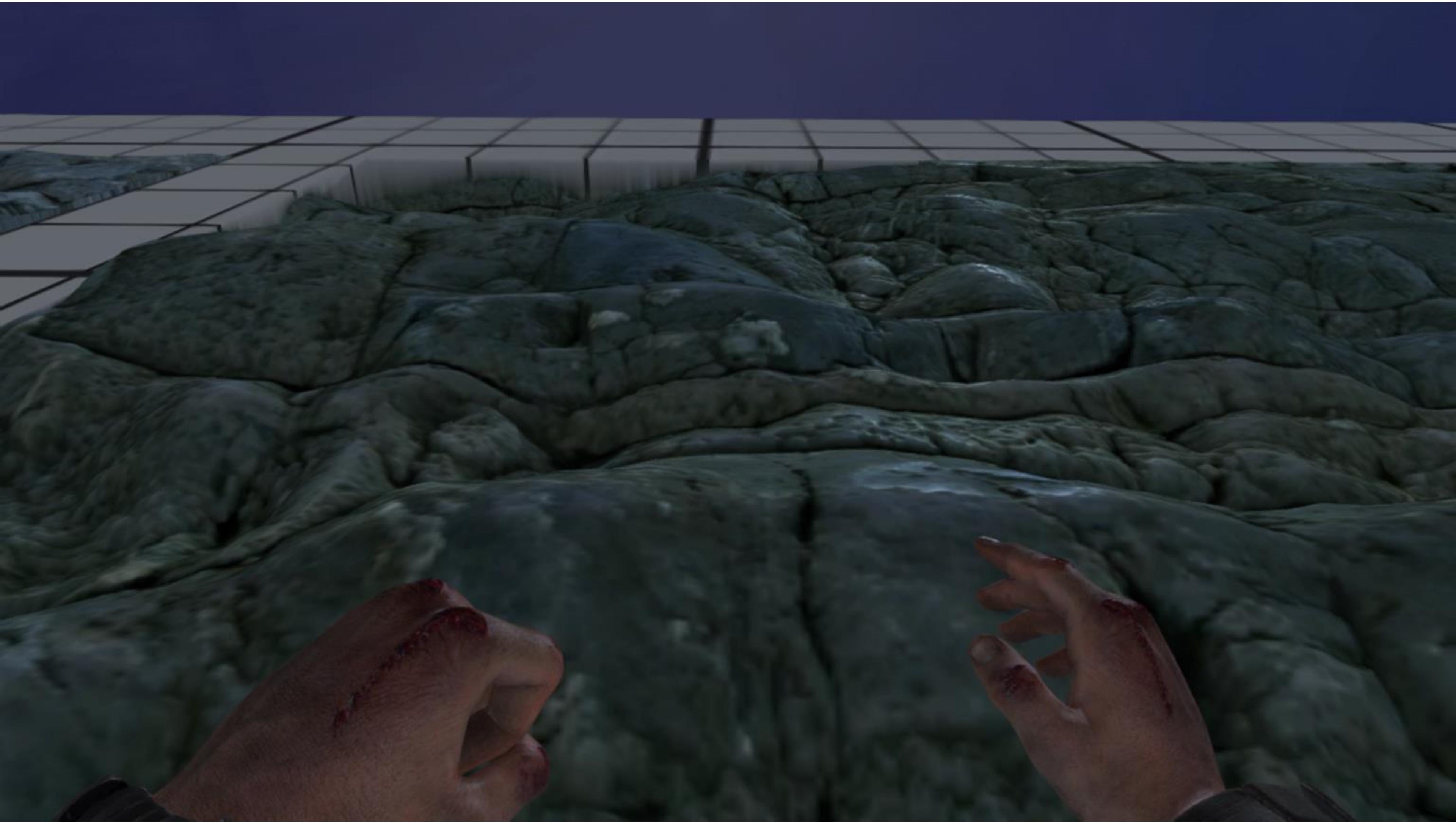
Post FX?

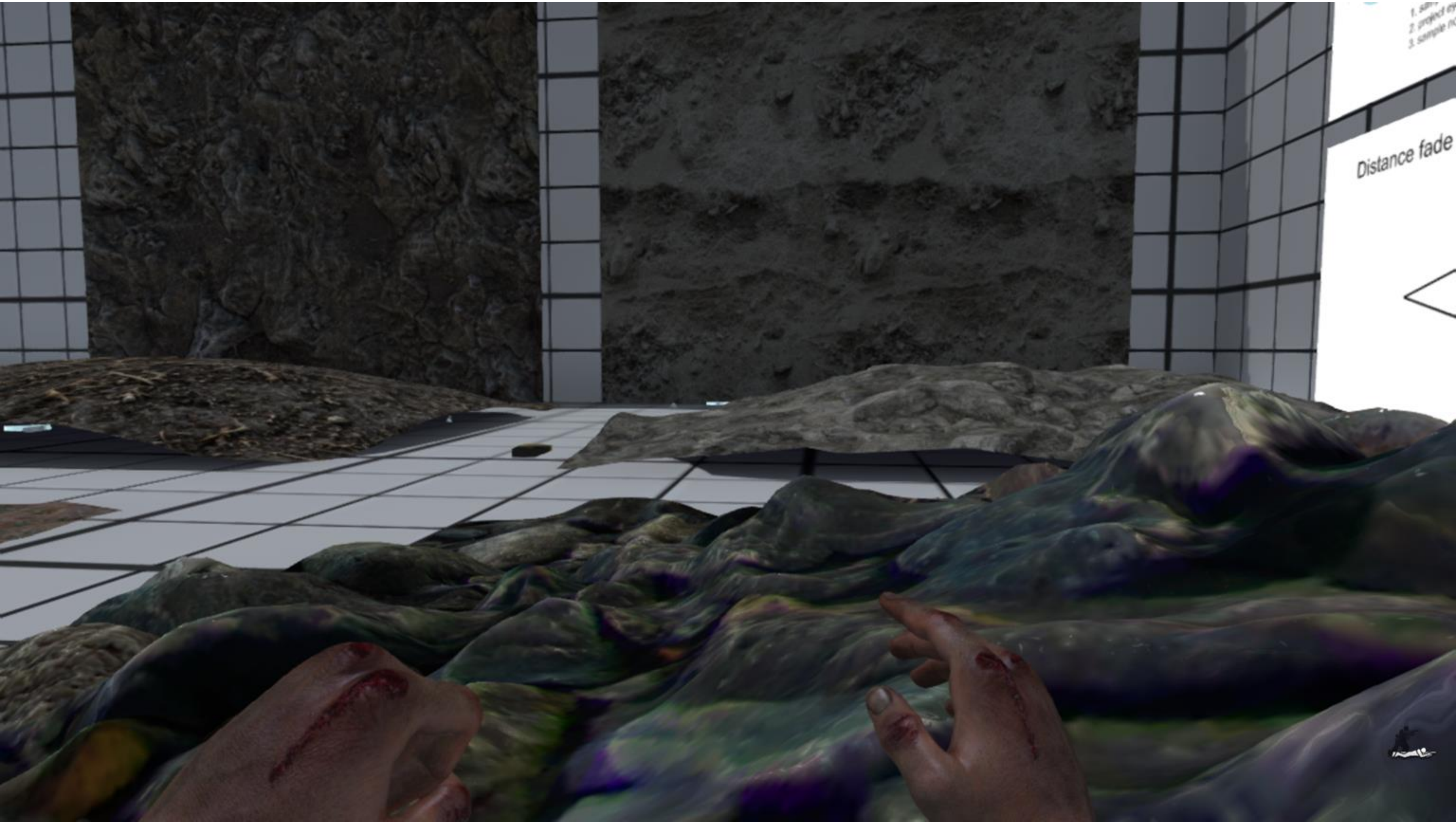


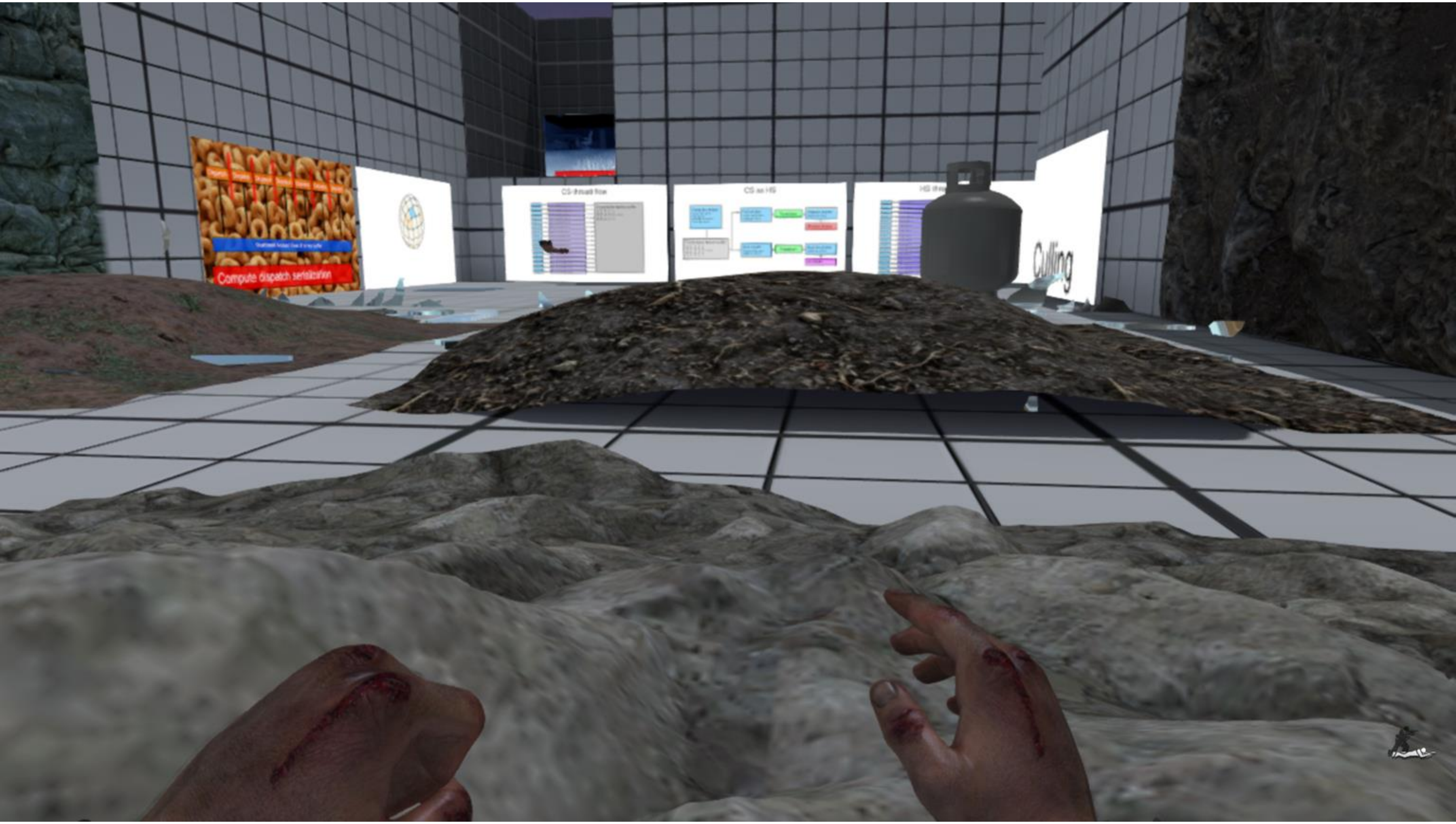
Vertex shader
Control point 0
Control point 1
Control point 2
Control point 3
Control point 4
Control point 5
Control point 6
Control point 7
Control point 8
Control point 9
Control point 10
Control point 11
Control point 12
Control point 13
Control point 14
Control point 15
No reuse
Hull shader
Control point 0
Control point 1
Control point 2
Control point 3
Control point 4
Control point 5
Control point 6
Control point 7
Control point 8
Control point 9
Control point 10
Control point 11
Control point 12
Control point 13
Control point 14
Control point 15


FPS
Xbox One
0.8ms
Xbox 360
1.2ms
FPS
DISTORTION
GLOW

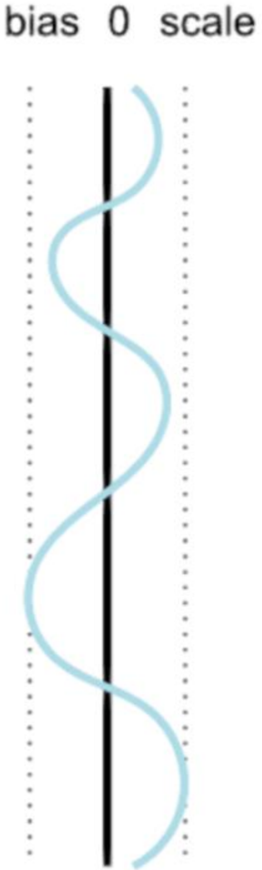

Distance fade

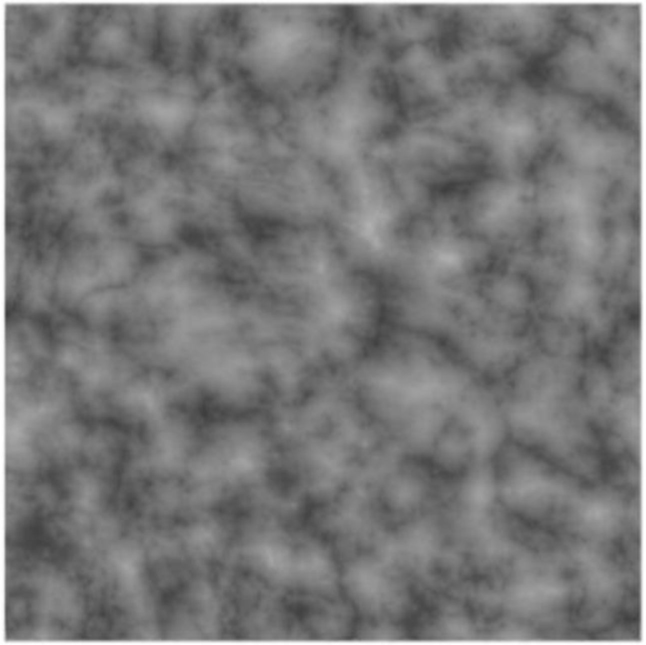


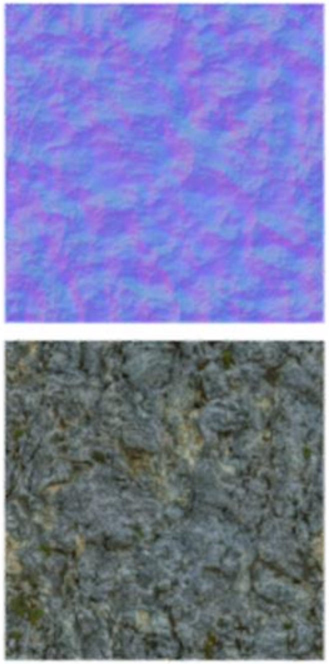

Distance fade


Compute dispatch serialization
CS threads flow
CS as HS
Culling

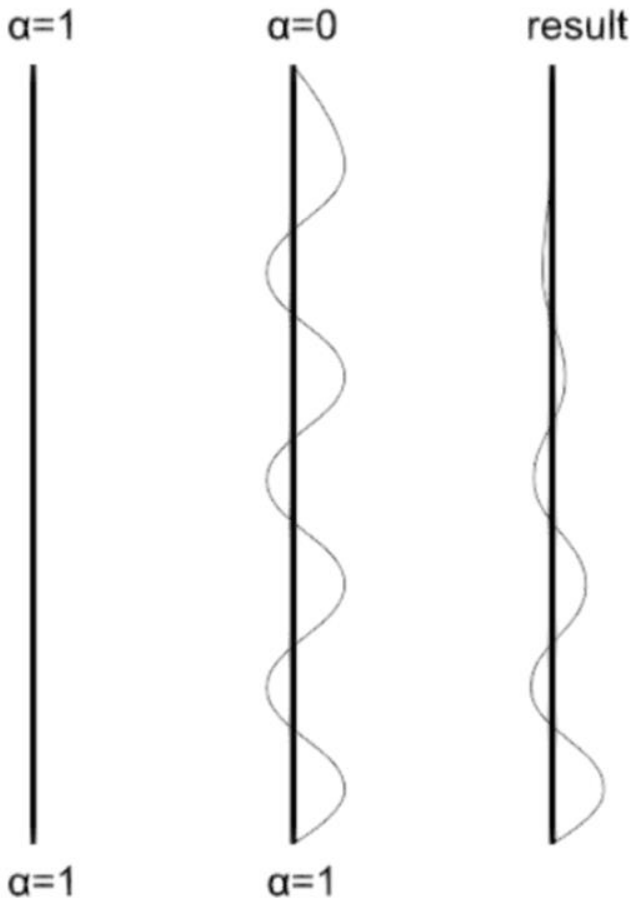
Displacement



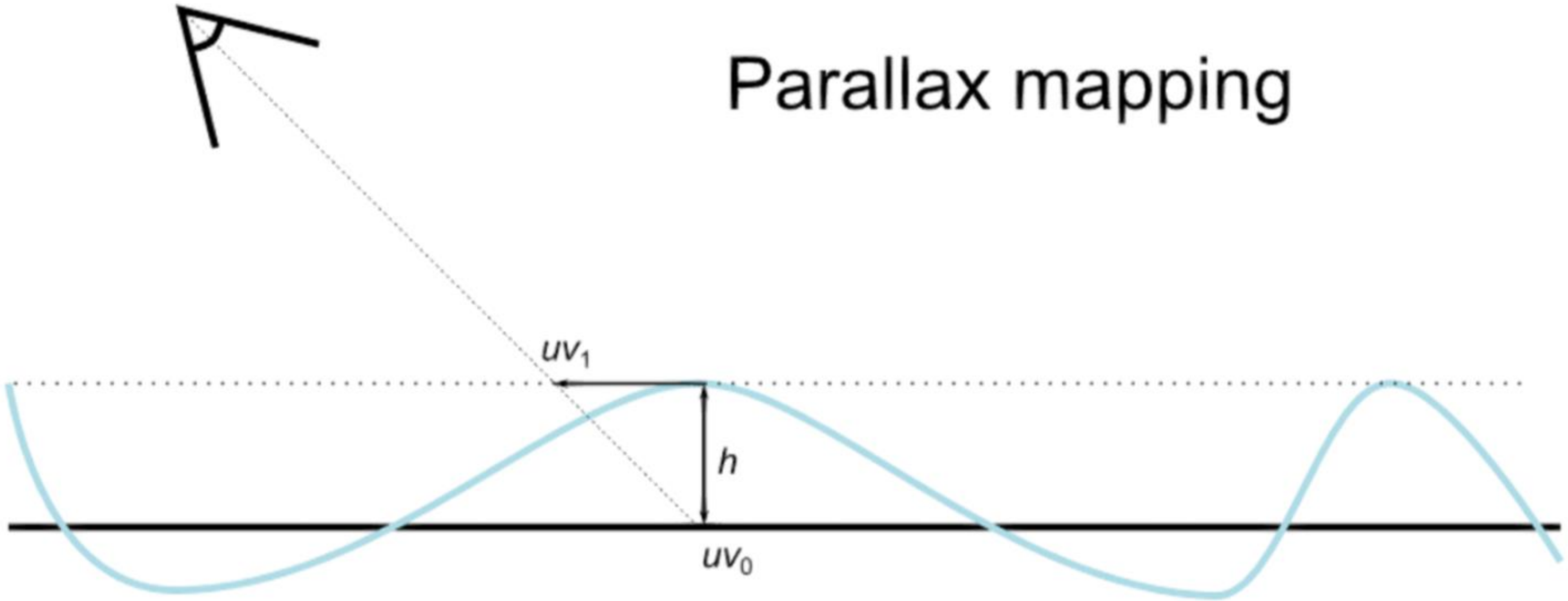




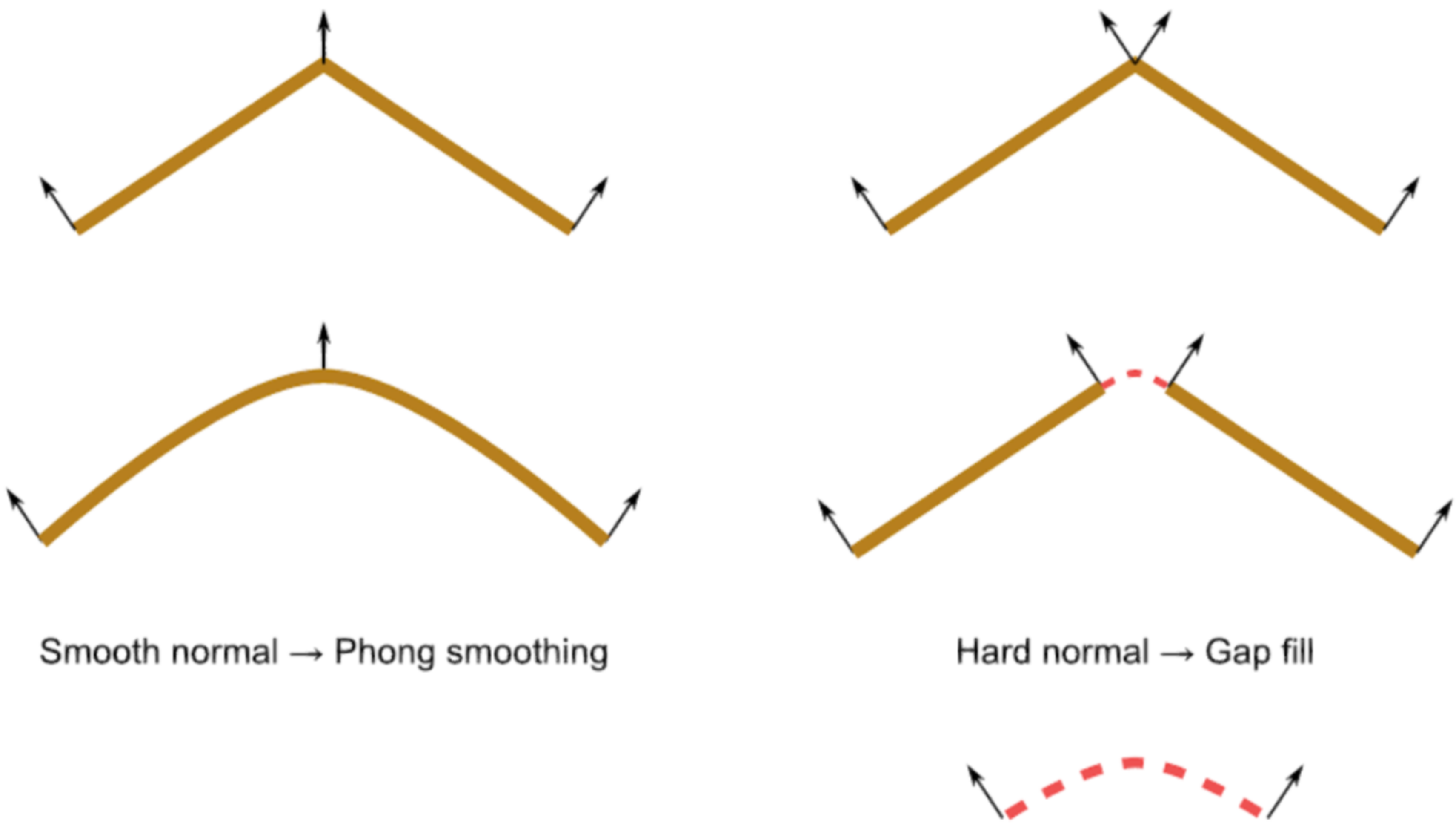
Layering



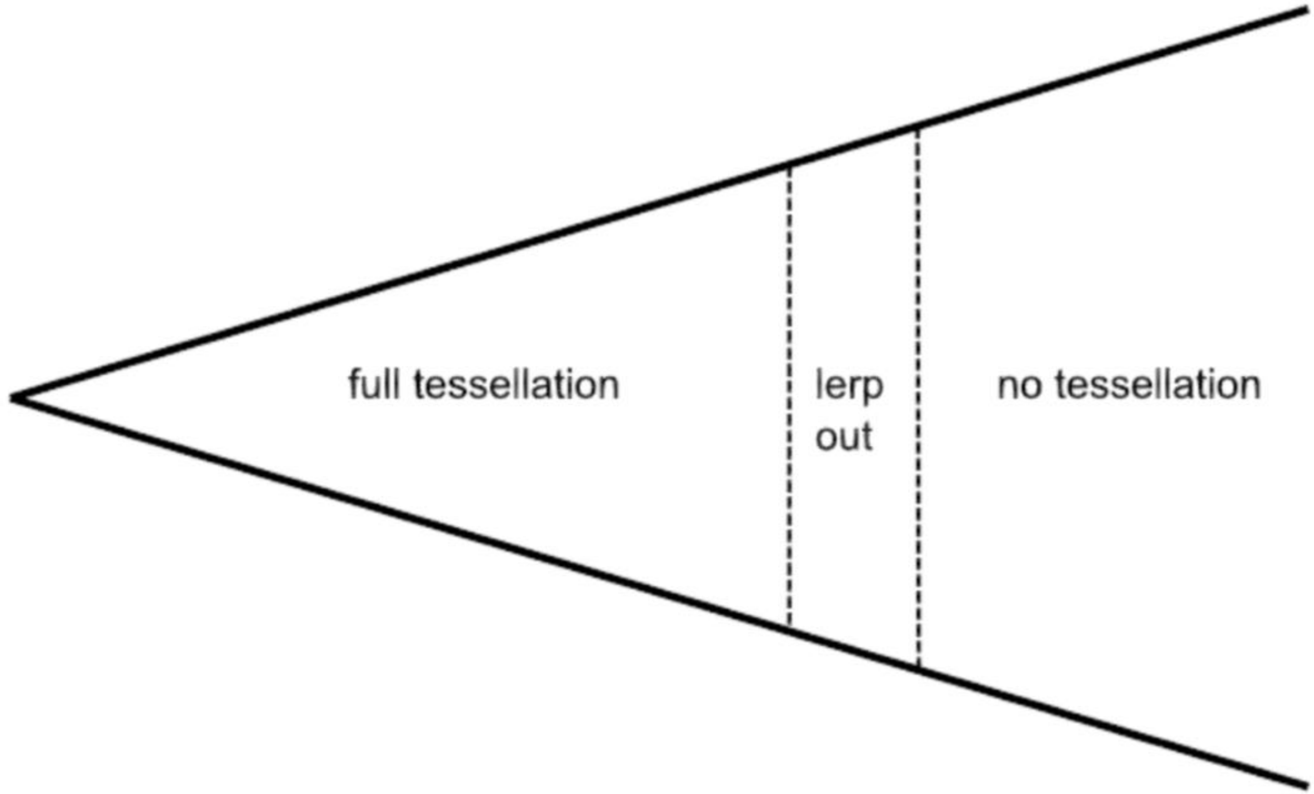
Parallax mapping



1. sample height in PS
2. project eye ray to extruded plane, adjust uv
3. sample normal, color, spec


Smooth normal → Phong smoothing
Hard normal → Gap fill

Distance fade




Evac to the river for exfil.
Stream
30
300

Thanks

Paul Edelstein
Andrew Aye
Michael Vance
Matthias Niessner
Jens Raab
Kefei Lei
Maceij Kot
Felipe Gomez-Frittelli
Jake Rowell
John Dobbie
Riccard Linde