

Hidden Surface Removal

CMSC 161: Interactive Computer Graphics

1st Semester 2014-2015

Institute of Computer Science

University of the Philippines – Los Baños

Lecture by James Carlo Plaras

Hole



6:16



2

+ 86

92

Hidden Surface Removal

Discovery of what part of the object is
visible/invisible to the viewer

Polygon-level determination

3D Objects composed of triangles

Hidden Surface Removal

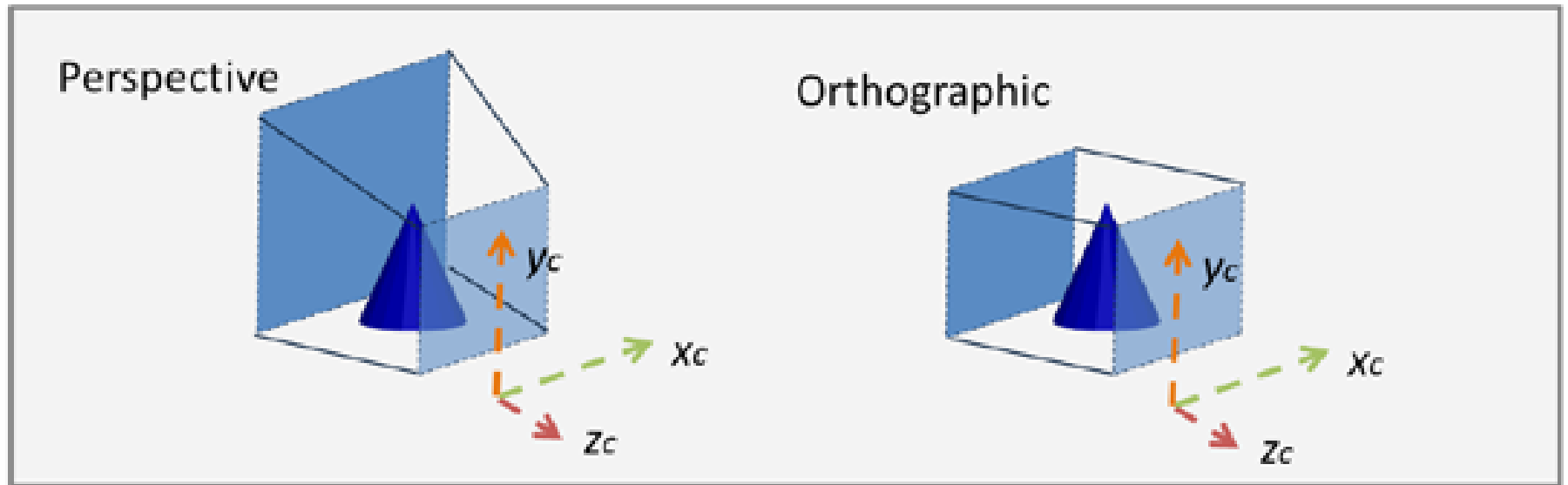
Outside the view frustum

Clipping and Culling Algorithms

Inside the view frustum

Hidden Surface Removal Algorithms

Frustum shape



The extent and shape of the frustum determines how much of the 3D view space is mapped to the screen and the type of 3D to 2D projection that takes place.

Hidden Surface Removal Approaches

Object Space Approach

Image Space Approach

Object-Space Approach

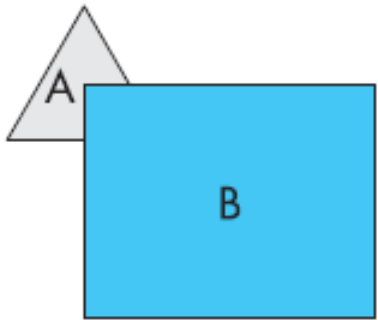
Works on geometry

Vertex-level approaches

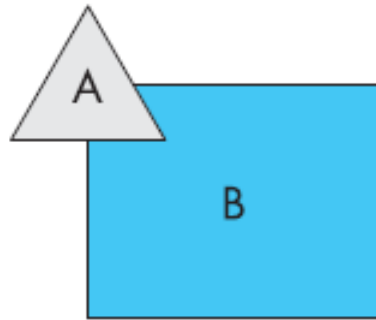
Applied before vertices are mapped to pixels

Works on 4 cases of polygon occlusion

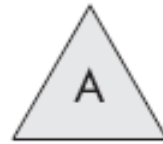
Object-Space Approach



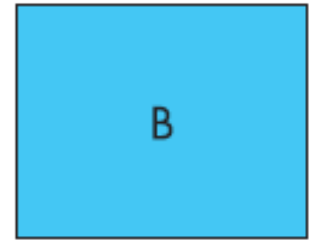
(a)



(b)



(c)



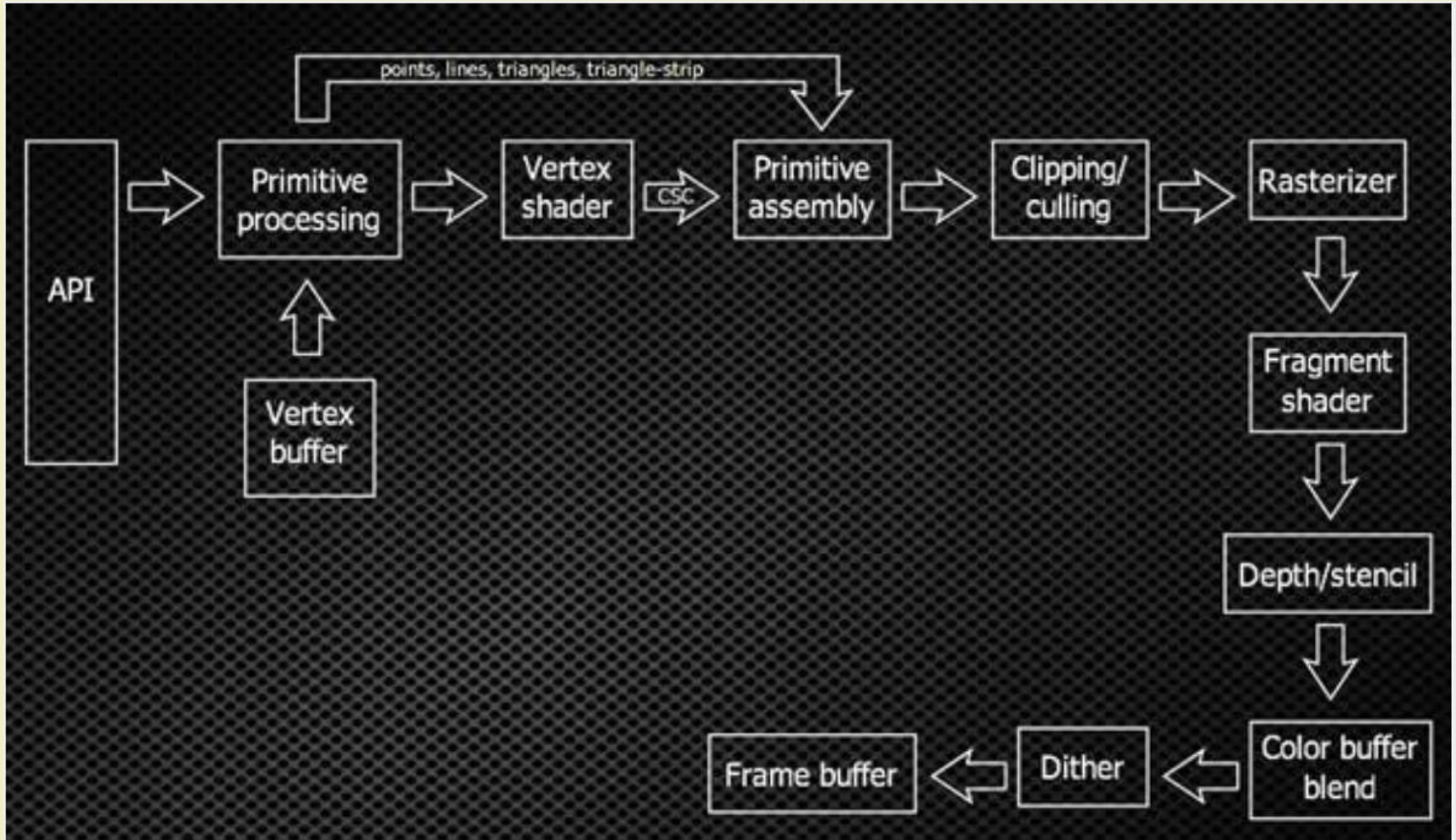
(d)

Image-Space Approach

Works on fragment level

Applied after rasterization

Hidden Surface Removal in WebGL



HIDDEN SURFACE REMOVAL ALGORITHMS

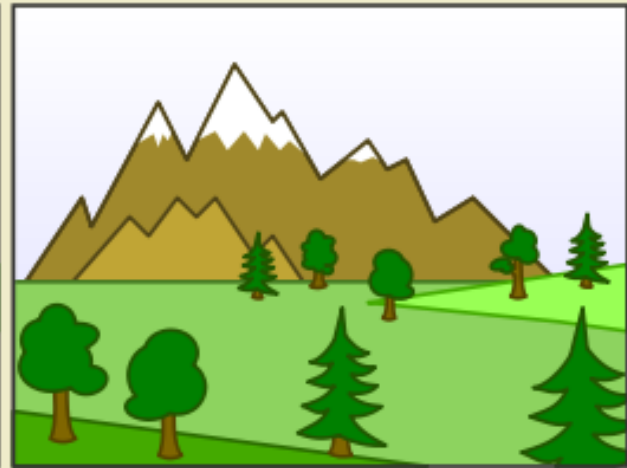
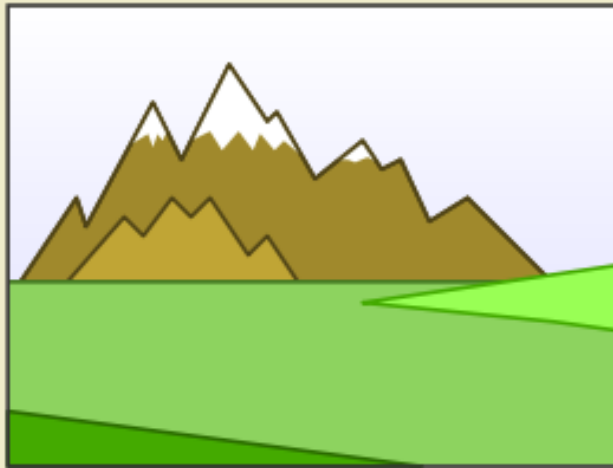
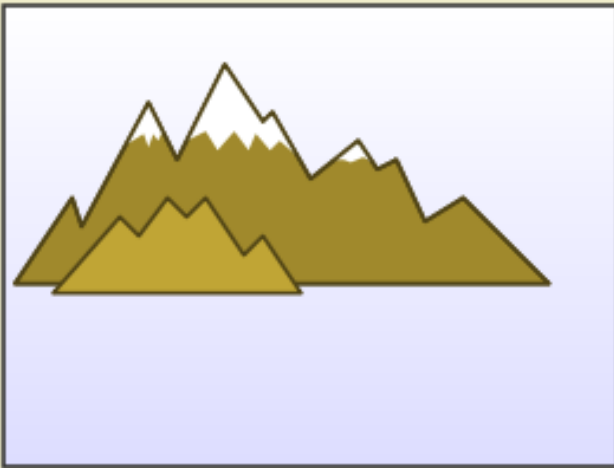
Painter's Algorithm

Object Space Approach

Also known as **Priority Fill**

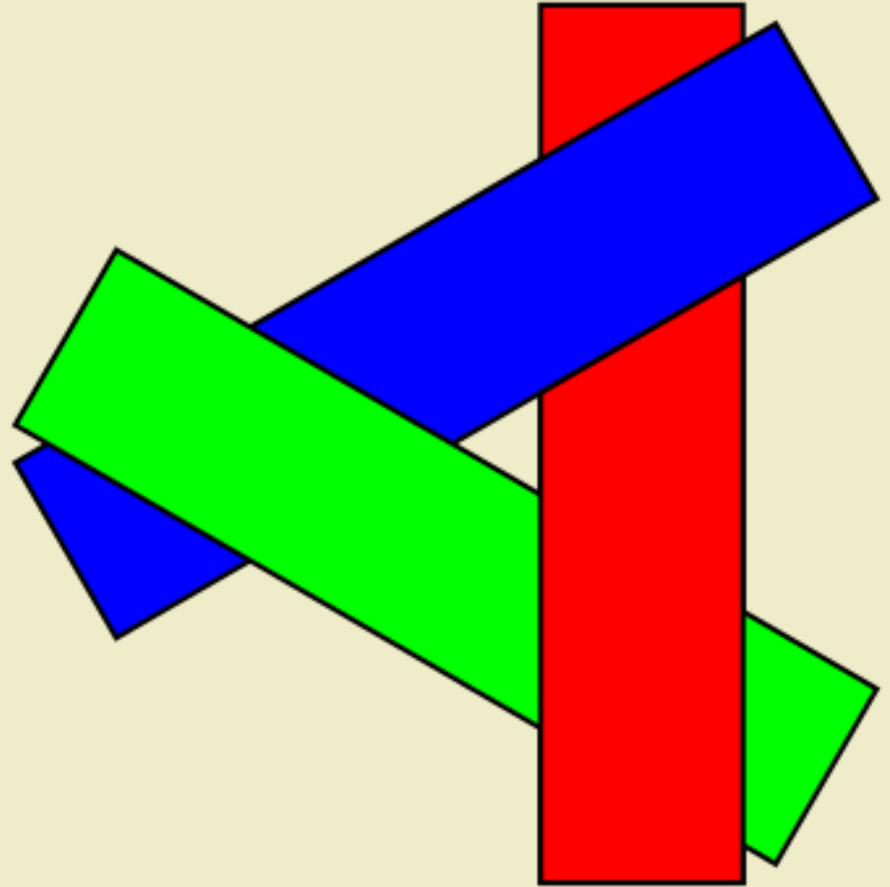
Based on the technique of scene painters

Painter's Algorithm



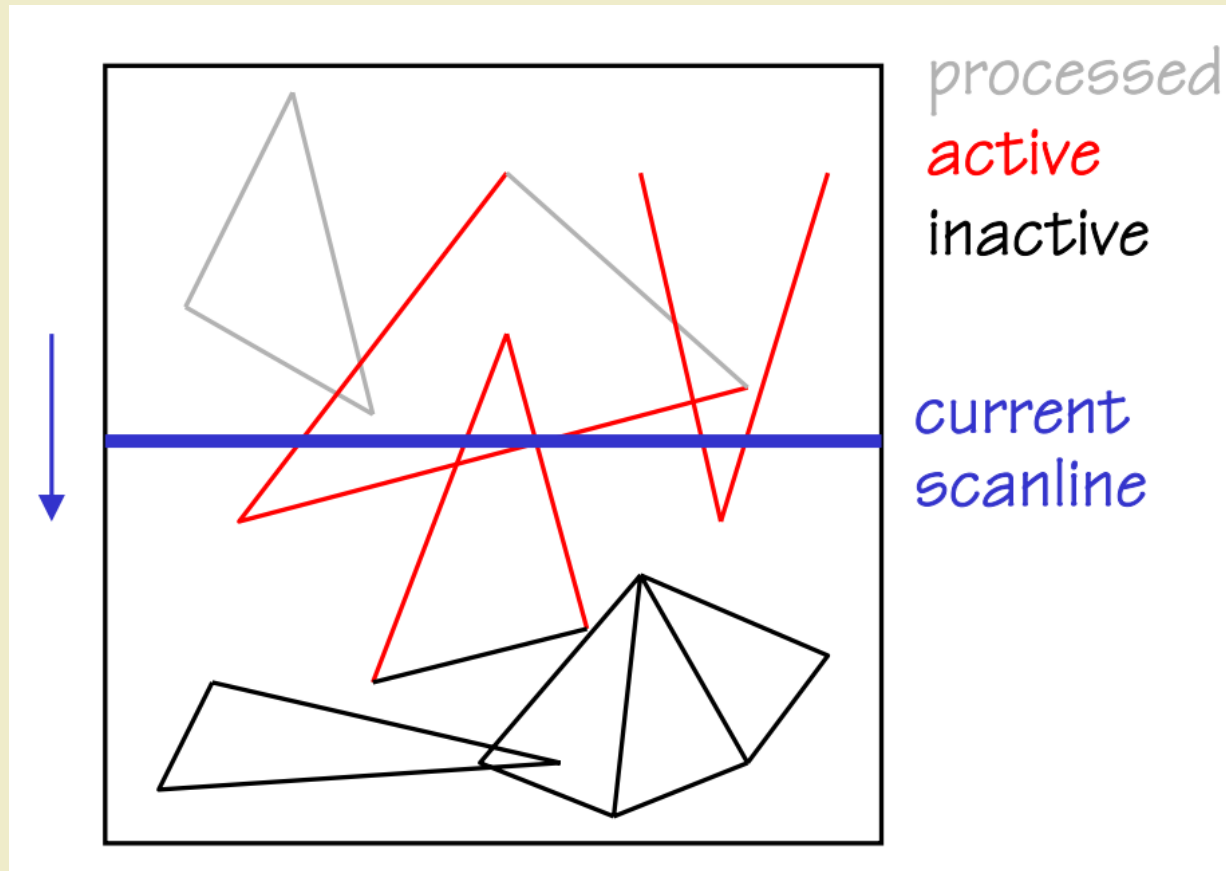
Painter's Algorithm Disadvantages

Cannot handle cyclic overlaps

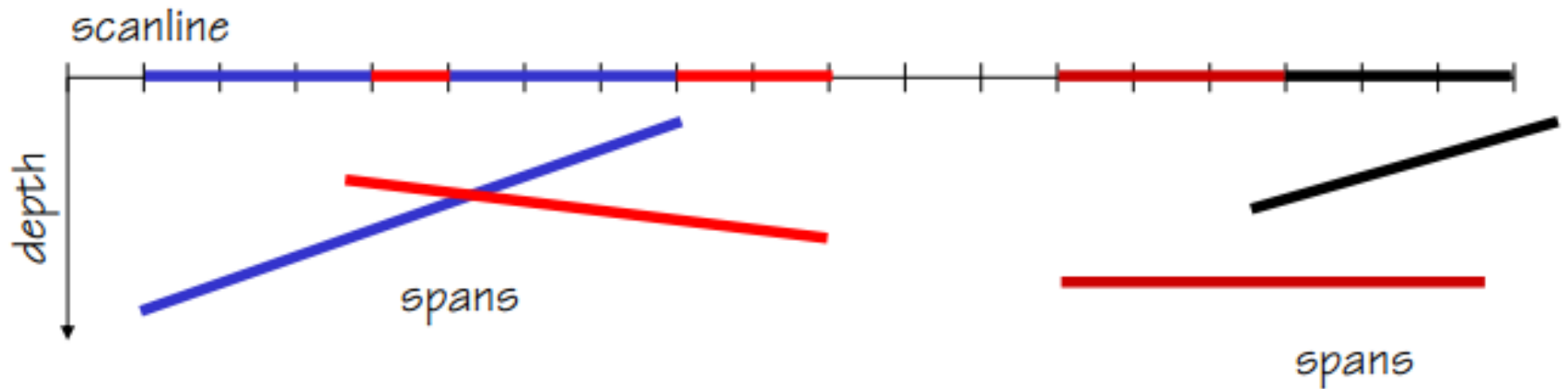


Scanline Algorithm

Using an **extended** Odd-Even Fill version



Scanline Algorithm



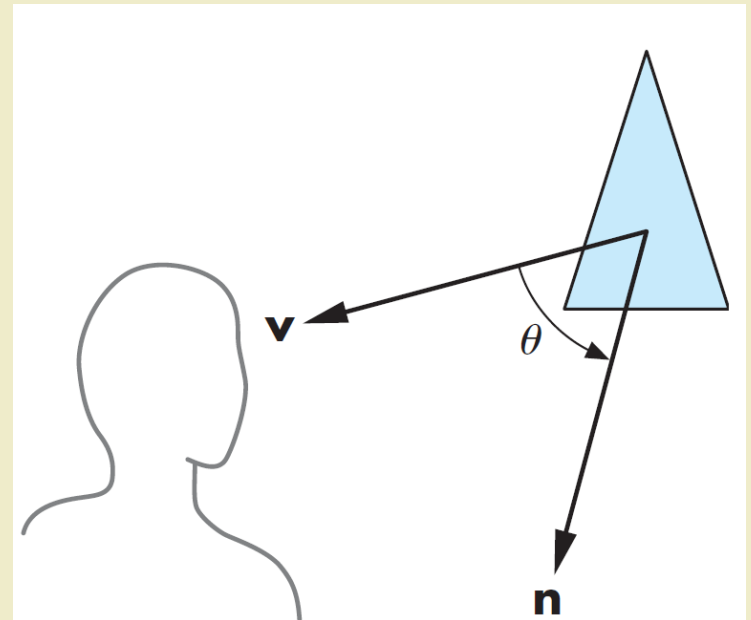
Back-Face Removal

Culling of polygon facing away from the viewer

Based on the normal vector of the polygon

– Facing forward if:

- $-90 \leq \theta \leq 90$
- $\cos \theta \geq 0$
- $n \cdot v \geq 0$



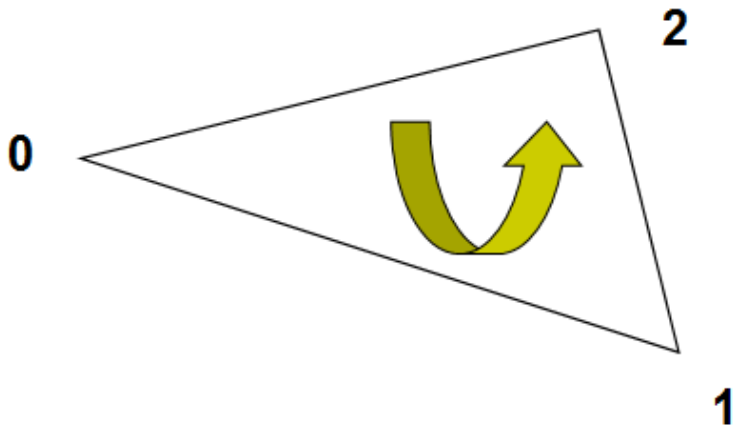
Back-Face Removal : Normal Direction

OpenGL/WebGL Exploits the property of a
polygon normal

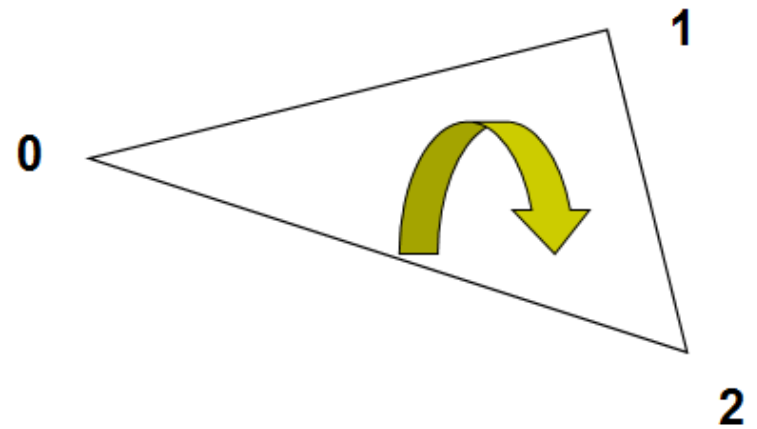
Vertices counterclockwise: Front-facing

Vertices clockwise: Back-facing

Back-Face Removal: Normal Direction



Front facing



Back facing

Z-Buffer Algorithm

Image-Space approach

Most used hidden-surface removal algorithm

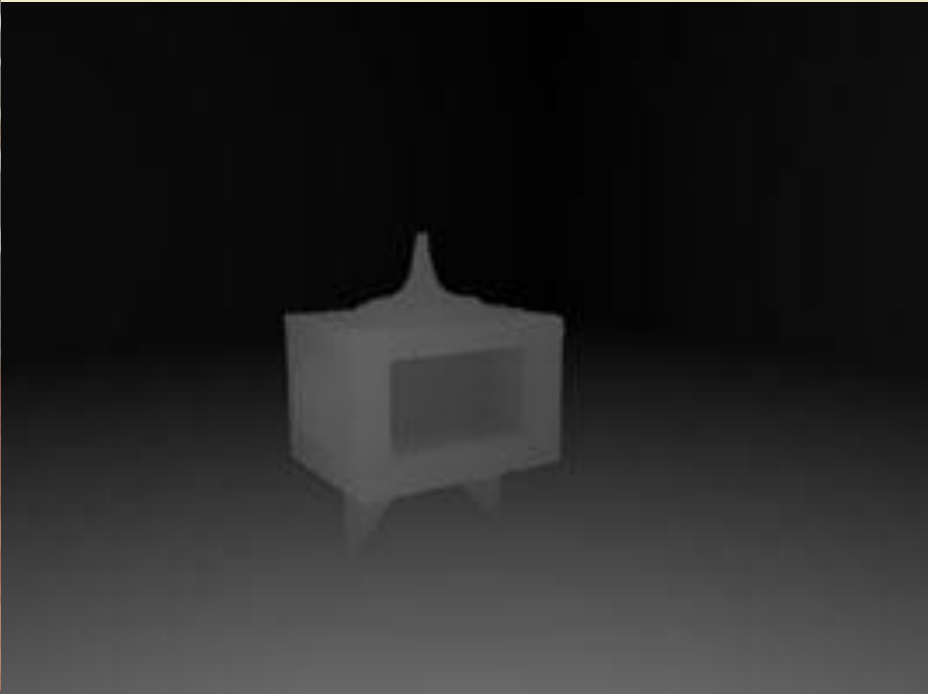
Easy to implement

Used by OpenGL/WebGL

Z-Buffer Algorithm



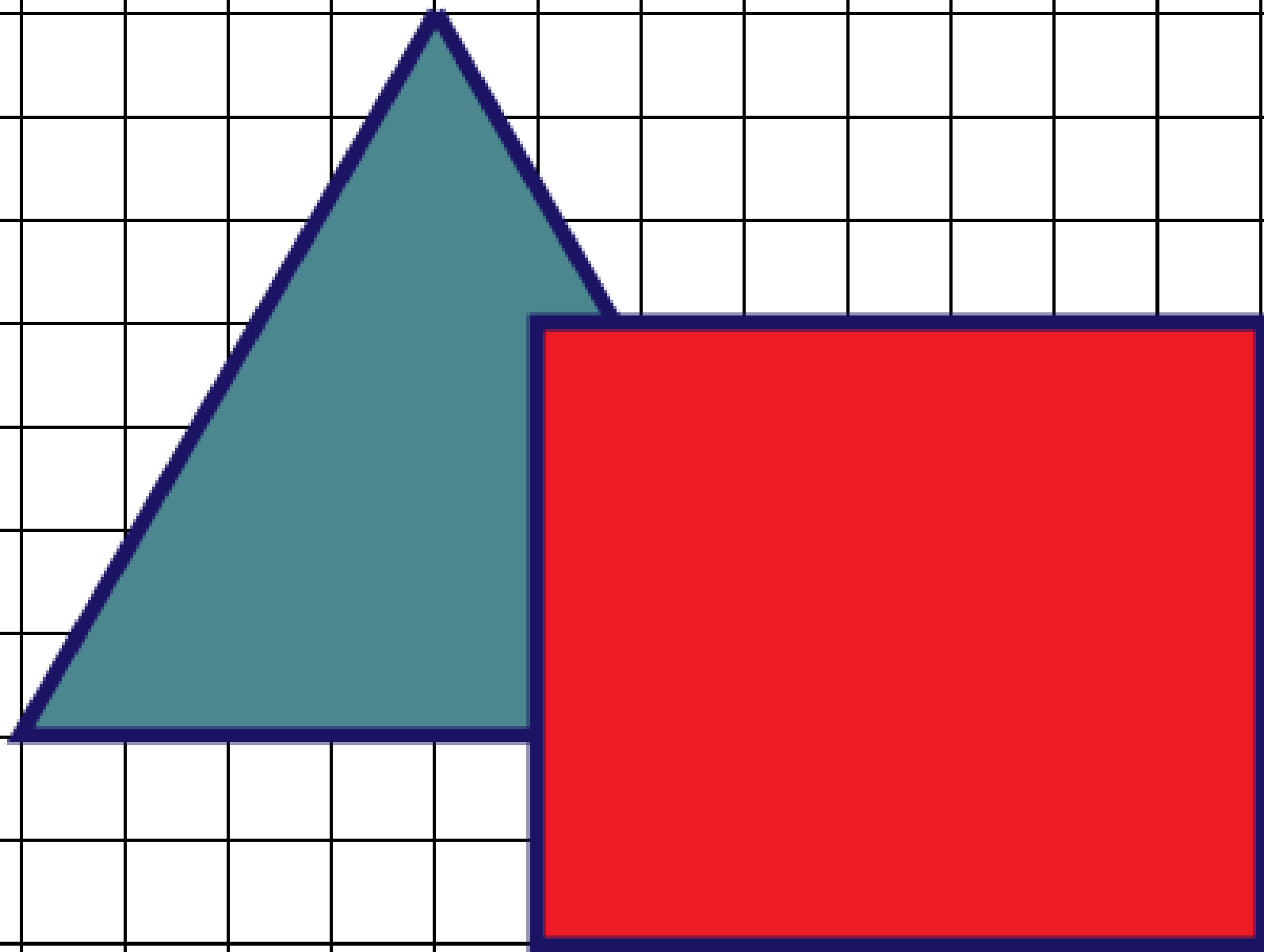
COLOR BUFFER



DEPTH BUFFER

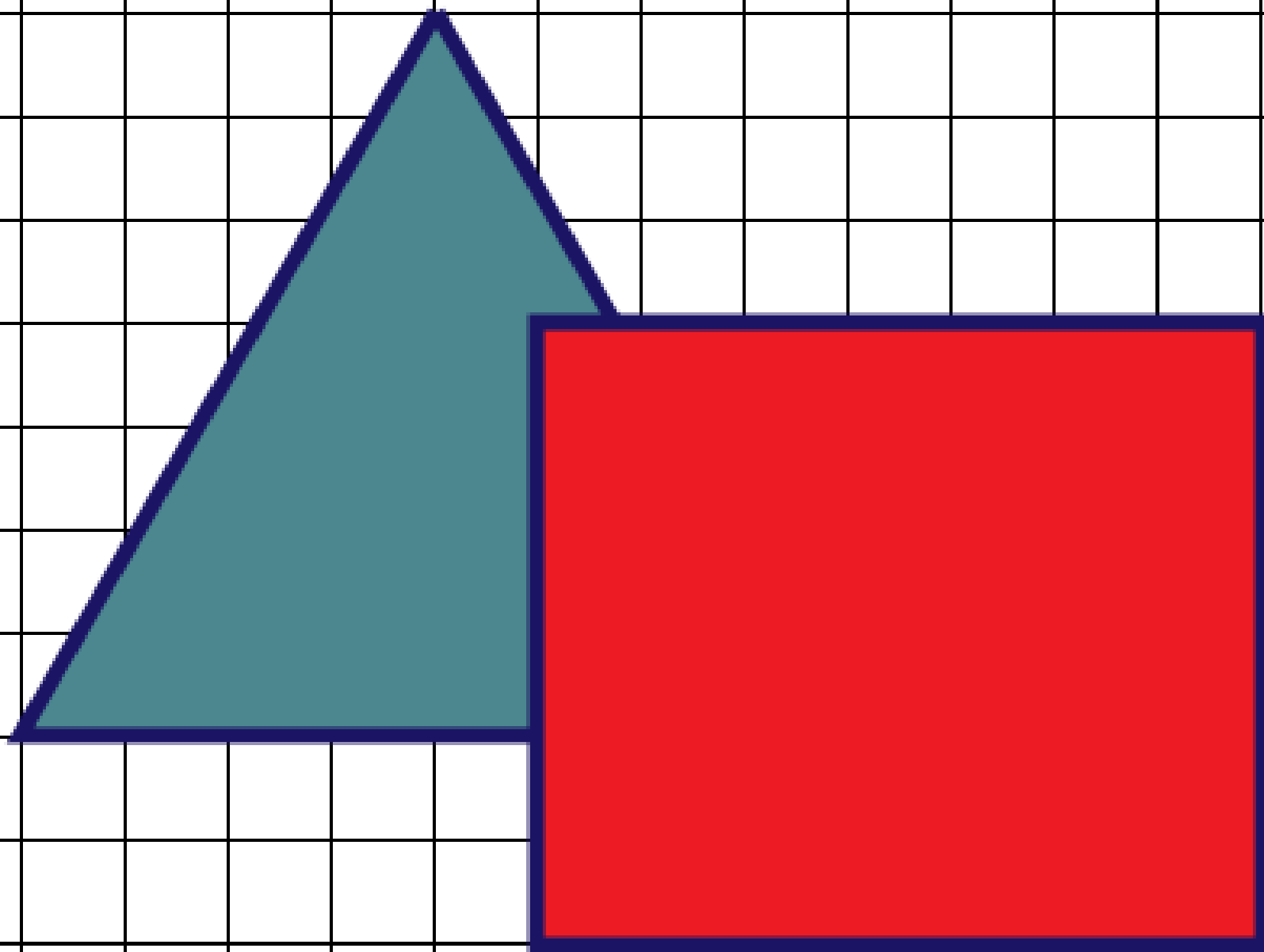
Z-Buffer Algorithm

```
For each face F in the scene {  
    For each fragment P in face F {  
        if (depthBuffer[P.x][P.y] < P.depth) {  
            colorBuffer[P.x][P.y] = P.color  
        }  
    }  
}
```



[illegible]

[illegible]



[illegible]

[illegible]

Z-Buffer: Depth Computation

Based on your projection matrix

- Perspective

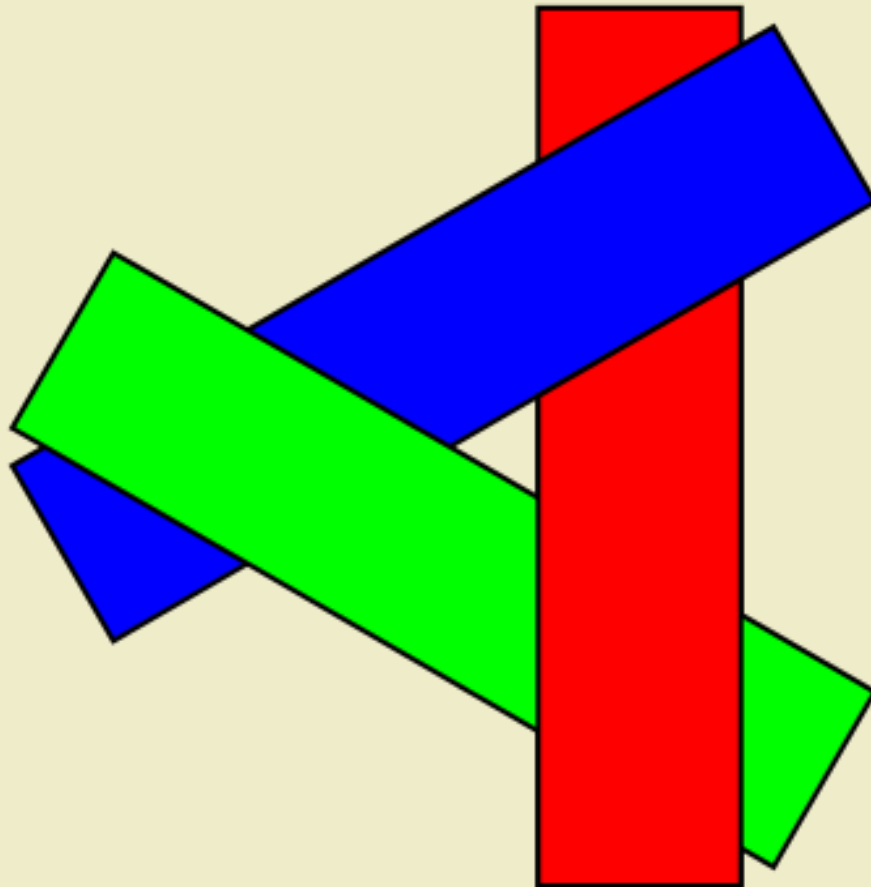
$$z' = \frac{far + near}{far - near} + \frac{1}{z} \frac{(-2 \times far \times near)}{far - near}$$

- Orthographic

$$z' = 2 \times \frac{z - near}{far - near} - 1$$

Z-Buffer Advantages

Easily handles cyclic overlaps



More Hidden Surface Removal Algorithms

Binary Space Partition Trees

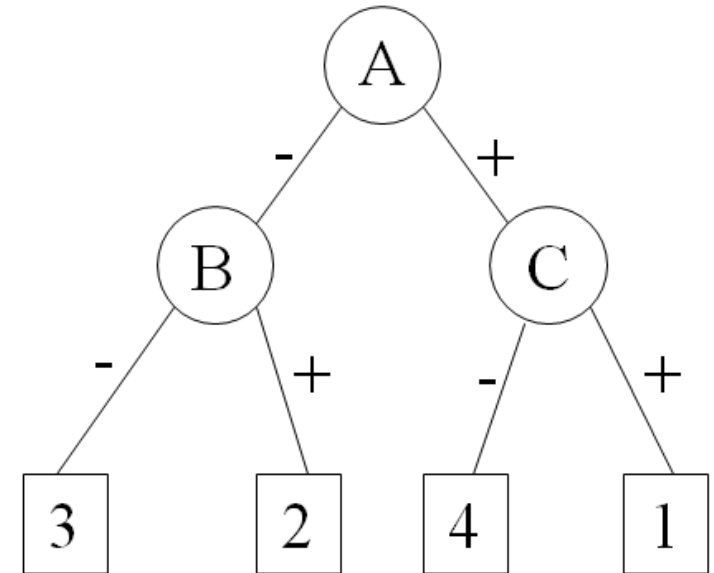
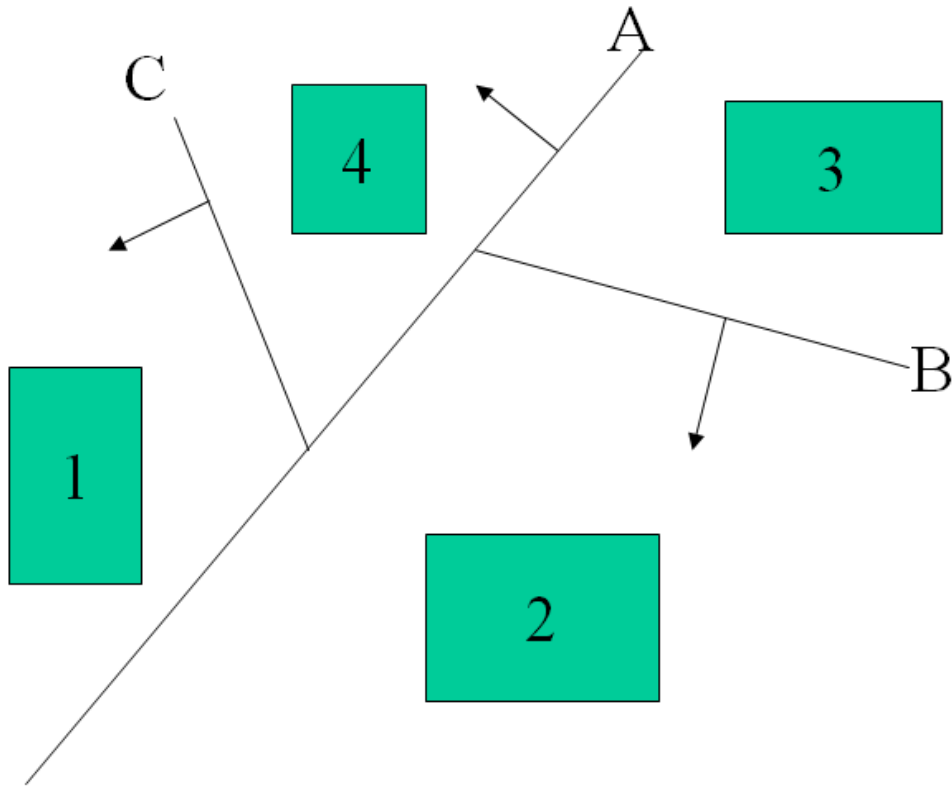
Warnock's Area Subdivision

Depth Sorting (generalization of Painters Algorithm)

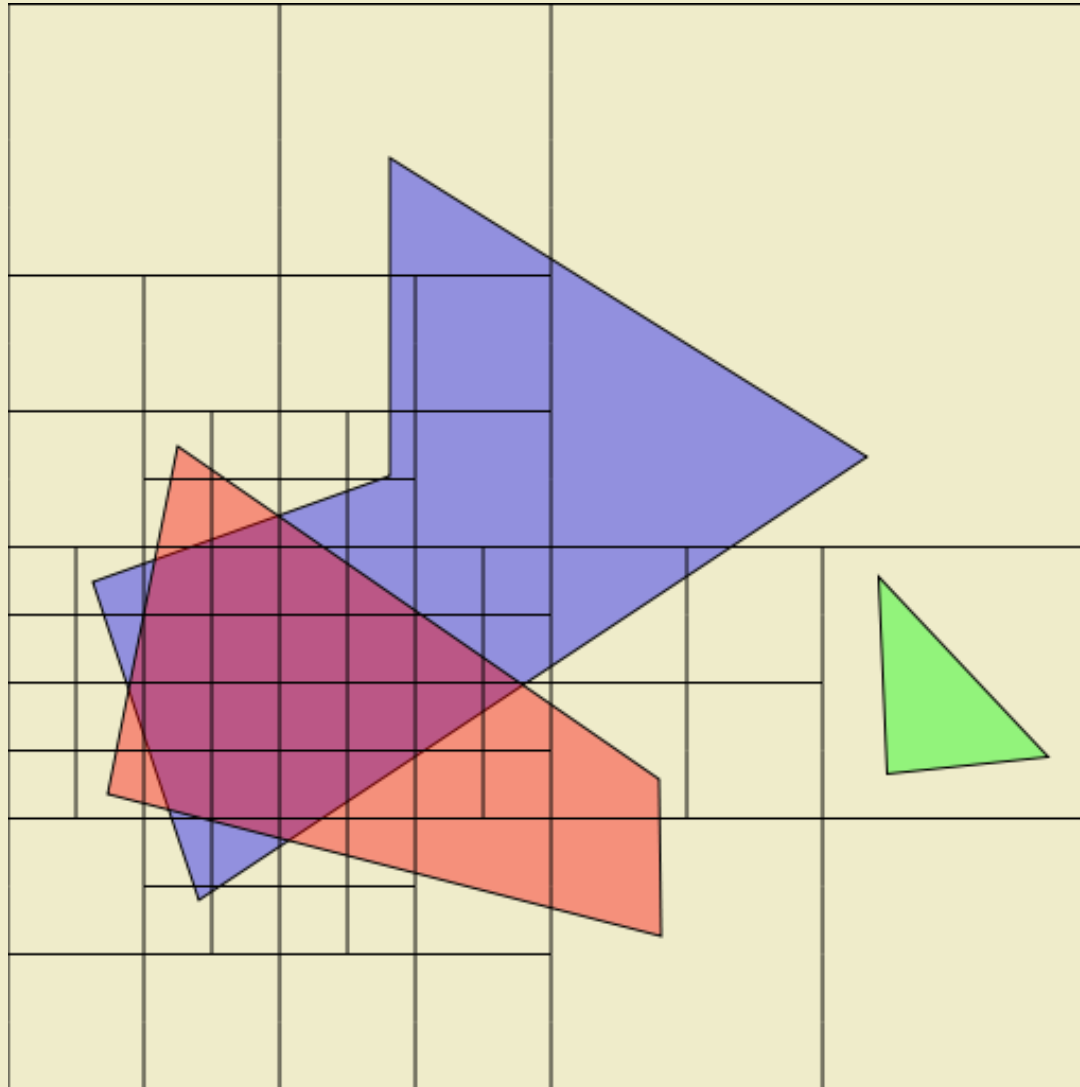
A-Buffer

Ongoing researches on faster HSR Algorithms...

Binary Space Partition Trees



Warnock's Area Subdivision



References

Books

- ANGEL, E. AND SHREINER, D. 2012. Interactive computer graphics : a top-down approach with shader-based OpenGL. Addison-Wesley. 6.ed. Boston, MA.
- CANTOR, D. AND JONES, B. 2012. WebGL Beginner's Guide. Packt Publishing. Birmingham, UK.
- MATSUDA, K. AND LEA, R. 2013. WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL.. Addison-Wesley. Upper Saddle River, NJ

Lecture Slides

- LINDEMAN, R. Hidden Surface Removal: CS 543: Computer Graphics

Images

- <http://i.imgur.com/ATIHeec.jpg>
- http://en.wikipedia.org/wiki/File:Painter%27s_algorithm.svg
- http://en.wikipedia.org/wiki/File:Painters_problem.svg
- http://en.wikipedia.org/wiki/File:Warnock_algorithm.svg