CS 498 VR

Lecture 18 - 4/4/18

go.illinois.edu/VRlect18

Review and Supplement for last lecture

1. What is aliasing? What is Screen Door Effect?
2. How image-order rendering works?
3. If there are several devices (mobile phone, computer screen, and VR headset, for example), how to compare the designated visual resolutions?

Object-order Rendering

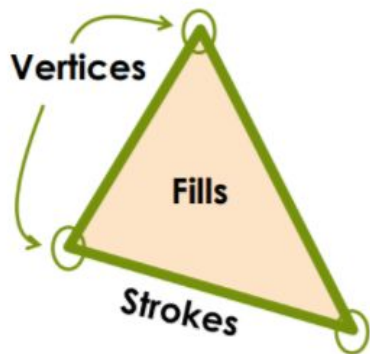1. Rasterization
2. Depth Order
3. Assign RGB values

(Compare to) Image-order Rendering

1. Ray Generation
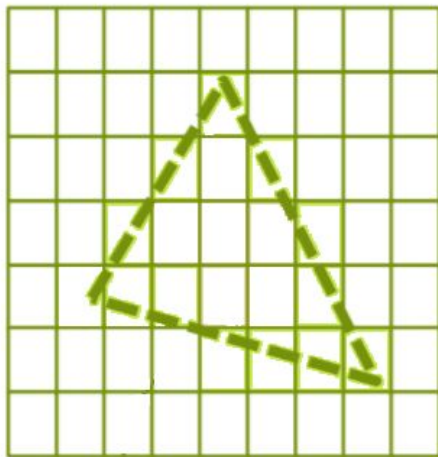2. Ray Intersection
3. Assign RGB values

Object-order Rendering

1. Rasterization

From primitive geometries to visuals shown on pixels.

- Triangles
- Lines
- Polygons
- ...

But, will ALL resulting pixels be shown on the screen?

Only those triangles in the "front" will be shown, so:

2. Depth Order

Determine the order of the geometries in the scene.
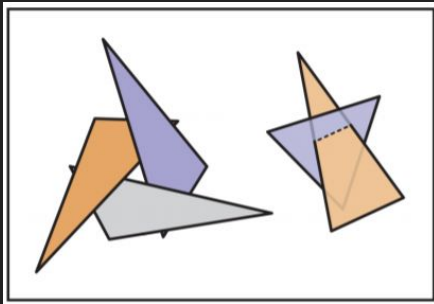To do hidden surface removal.

Solution 1. Painter's Algorithm:
Render the contents from back (farthest) to front (nearest)

2. Depth Order - Painter's Algorithm

What problems does this method have?

- Not efficient: we have tried to render unnecessary contents, and we have sorted all polygons - O(nlogn), not efficient enough. Also, hard to pick representative vertex to sort the polygons.

- Not adequate: we cannot handle occlusion cycles and intersecting triangles:

2. Depth Order - Z-buffer

Key Observation: Each pixel displays color of only one triangle. We can ignore everything behind it.
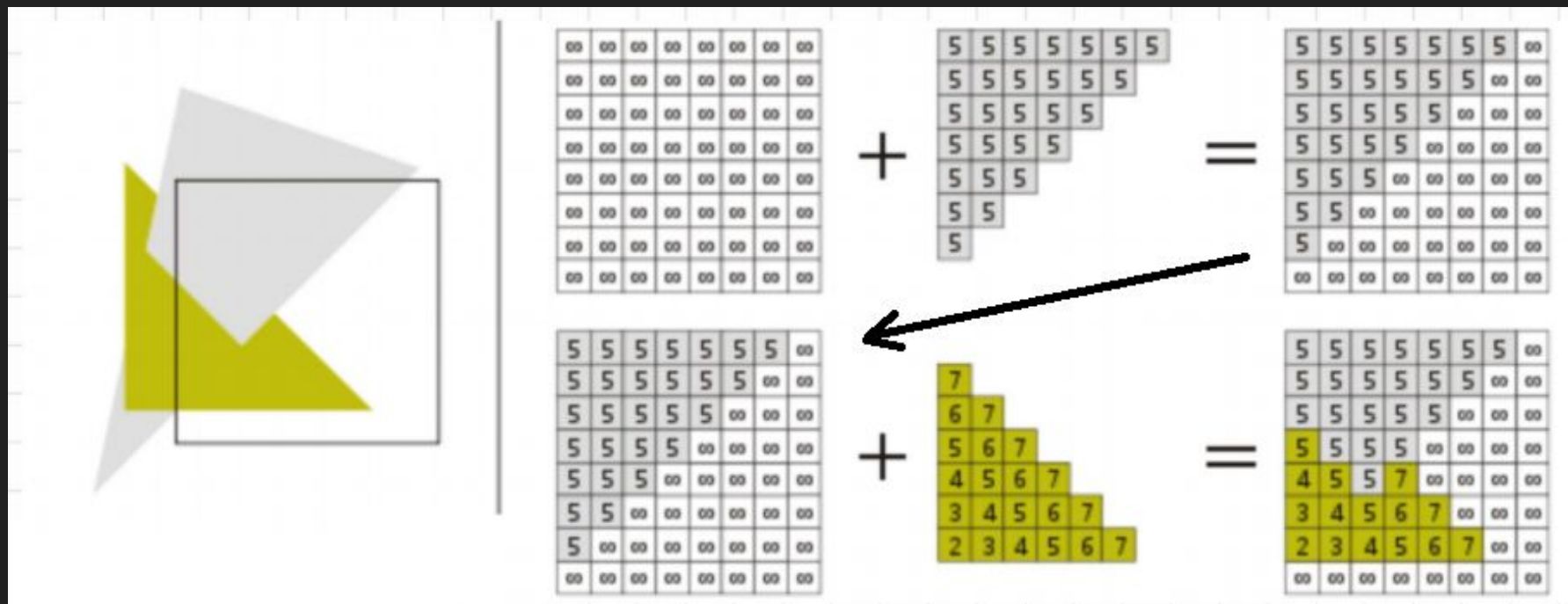- Don't need to sort triangles, just find for each pixel the closest triangle.
- Z-buffer: one fixed or floating point value per pixel.
- Algorithm:
  For each rasterized fragment (x,y)
      If z < zbuffer(x,y) then // if it is closer than the previously handled ones
          framebuffer(x,y) = fragment color // change the color
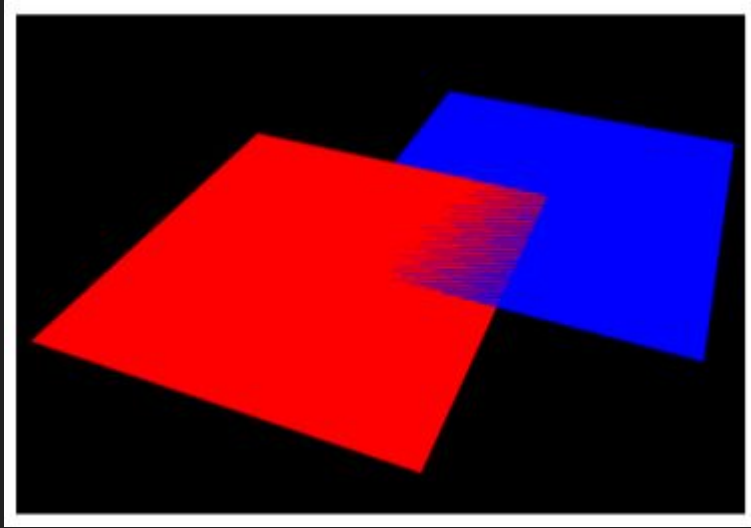   zbuffer(x,y) = z // overwrite the z buffer

2. Depth Order - Z-buffer

Example:

2. Depth Order - Z-buffer

Still have some flaws/difficulties: Z-fighting
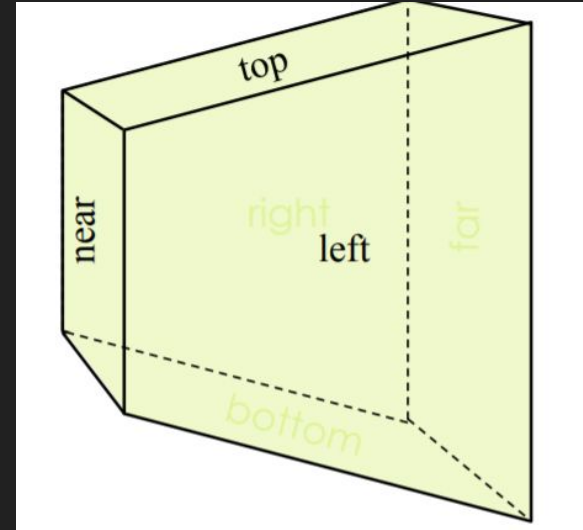

Solutions:
1. Move co-planar polygons slightly away from each other
2. Move near and far clipping planes as close together as you can
(More in CS 418)

Step before complete: clipping/culling

Clipping: Remove polygons outside of the view.

Culling: Remove triangles outside of the (View Frustum):
Remove hidden triangles (Occlusion)
Remove triangles on the back of objects (Backface)

3. Assign RGB Value - Interpolation

In the last step of Object-Order Rendering, interpolation is needed.
Quick Question: why not necessary in Image-Order Rendering?

Interpolation (in Graphics): Extend from discrete set of values to larger scales.

Example: Barycentric Coordinate (for triangle)

Input: Triangle (vertices, corresponding color values)

Output: The colors of all regions in this triangle.

3. Assign RGB Value - Barycentric Coordinate

Brute-force solution: for any point in this triangle, calculate the distance from this point to three vertices, taking inverse of them as weight and normalize. - Complex

Barycentric Coordinate: _____, s. t. _____
Let target point's Cartesian coordinate position (x, y, z) be p, three vertices'
Cartesian coordinate position be p1, p2, p3. If _____,
Then the Barycentric Coordinate of this point in this triangle is (a1, a2, a3)

a1, a2, a3 is between 0 and 1 -> _____
a1/a2/a3 is 0 -> _____
a1/a2/a3 = 1, i.e. two of a1, a2, a3 are 0 -> _____

3. Assign RGB Value - Barycentric Coordinate

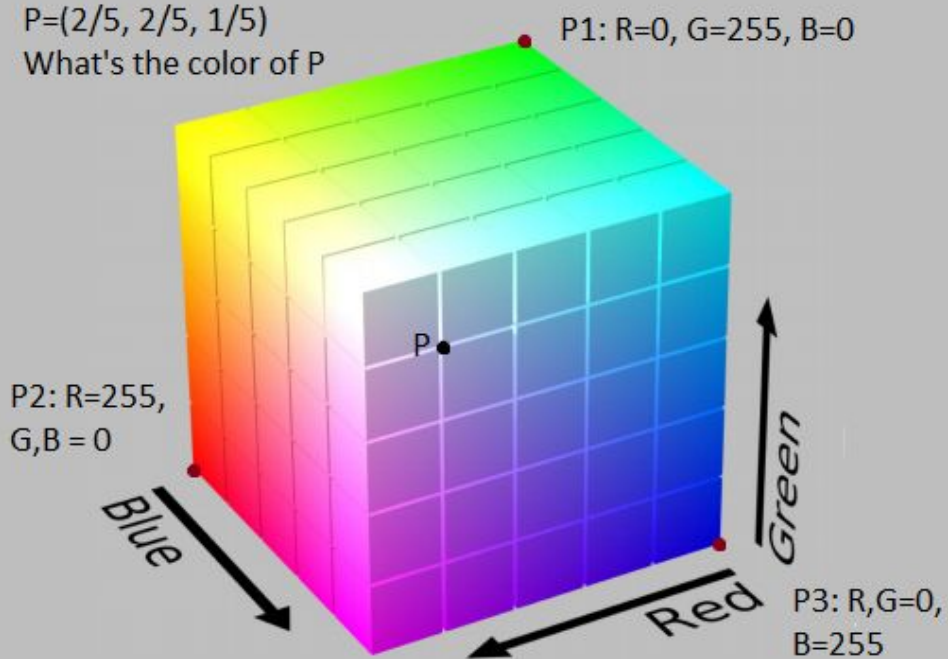Usage: Assigning RGB values: R = __R1+__R2+__R3. Similar for G and B.
Also can be used as a tool in depth order: z = __z1+__z2+__z3. And feed z to the z-buffer.

Many other ways to do interpolation in different conditions: See more in CS 418

RGB mapping

P=(2/5, 2/5, 1/5)
What's the color of P

P1: R=0, G=255, B=0

P2: R=255, G,B = 0

P3: R,G=0, B=255

P

Blue

Green

Red

Color of P:
P = ___*_____ +
___*____ +
___*____

R = ___, G = ___, B = ___

Review for this lecture:

We have covered the steps of Object-order Rendering:
    What are the steps?
    What's the key difference with Image-order Rendering?
    In what cases do O-o Rendering work better than I-o Rendering?
    In most cases which one performs more efficient? Higher quality?

Rasterization: What is it?

Depth Order: What are the methods we have discussed? Their algorithms? Their flaws? What is Clipping/Culling?

Assigning RGB Values: What is interpolation? What is Barycentric Coordinate? Could you do the transformation from/to Barycentric Coordinate to/from other coordinate?

Appendix: Ray Tracing

https://www.youtube.com/watch?v=jkhBlmKtEAk

Next lecture...

Mapping

- Texture mapping, bump mapping, normal mapping…
- More Challenges and Solutions for Graphics in VR

Start the topic of Tracking

Announcements/Reminders...More on next slide

Final Project Open House Time Posted on Piazza
- If you have exam in the 7-10 time slot of that day, post private posts to reserve 5pm-7pm slots for open house.

Announcements...

HCESC Jump ARCHES Summer Internship 2018

Jump Trading Simulation and Education Center (jumpsimulation.org) and the University of Illinois Health Care Engineering Systems Center (healtheng.illinois.edu) are pleased to announce the ARCHES summer internship opportunity in the area of Medical Simulation and Virtual Reality. This internship, located at the University of Illinois Urbana-Champaign, is a full time summer internship.

Interns will be working on VR based simulation projects related to healthcare and medicine during this 8 week internship program. The projects shall be discussed briefly during the interview process.

Skill Base required:

- **Undergraduate/ Graduate students in ECE / Computer science / Bio-engineering** programs with a strong academic record
- Strong programming skills in C#
- Experience working in UNITY; Using Steam VR plugin for HTC Vive is highly preferred
- Prior knowledge or experience in virtual reality
- Willingness to learn and explore new applications for virtual reality in healthcare
- Should be a self-starter and work effectively in self-directed and proactive manner
- Ability to collaborate and work within a team and adhere to the timeline of the project

Interested candidates, please send your resume to hcesc@illinois.edu

Website: http://healtheng.illinois.edu/