

Using the Stencil Buffer for Visualization



Oregon State University

Mike Bailey

mjb@cs.oregonstate.edu

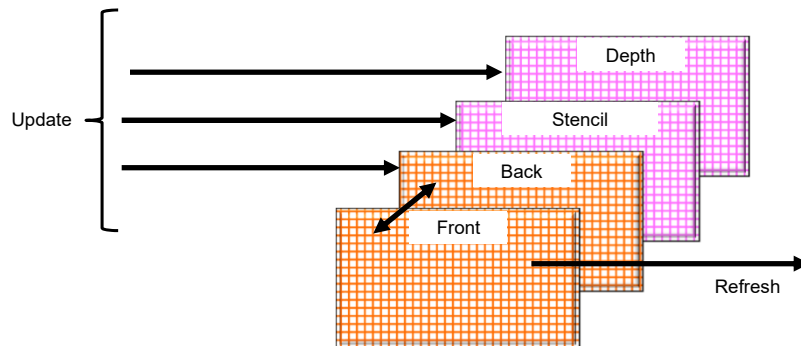


This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/)



Oregon State University
Computer Graphics

The Framebuffers



Here's how the Stencil Buffer works:

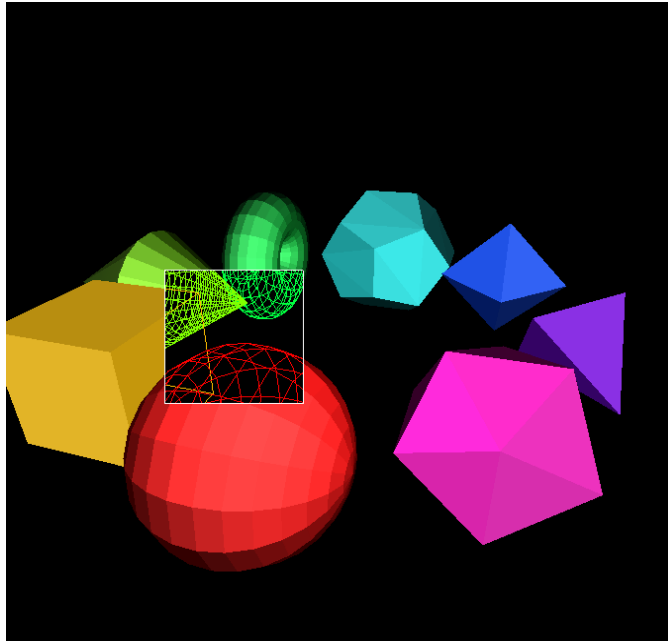
1. While drawing into the Back Buffer, you can write values into the Stencil Buffer at the same time.
2. While drawing into the Back Buffer, you can do arithmetic on values in the Stencil Buffer at the same time.
3. When drawing into the Back Buffer, you can write-protect certain parts of the Back Buffer based on values that are in the Stencil Buffer



Oregon State University
Computer Graphics

Using the Stencil Buffer to Create a *Magic Lens*

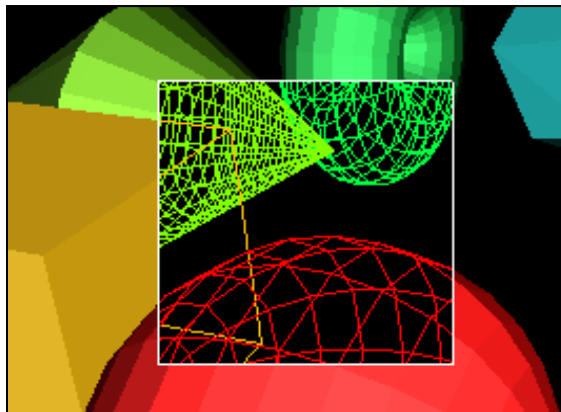
3



Using the Stencil Buffer to Create a *Magic Lens*

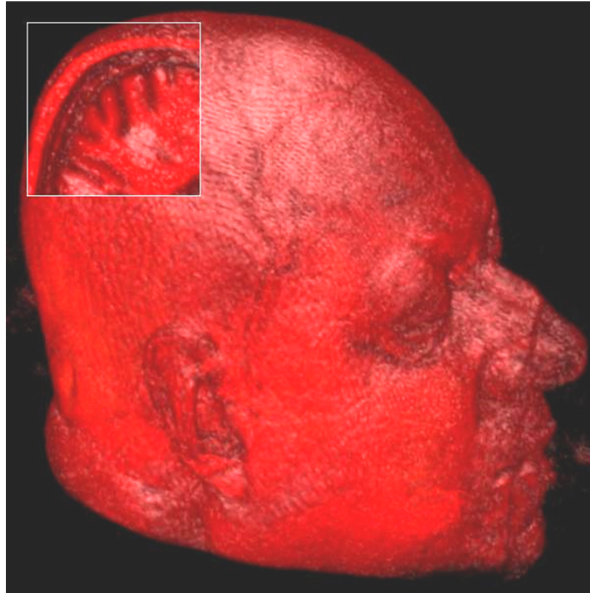
4

1. Clear the SB = 0
2. Write protect the color buffer
3. Fill a square, setting SB = 1
4. Write-enable the color buffer
5. Draw the solids wherever SB == 0
6. Draw the wireframes wherever SB == 1



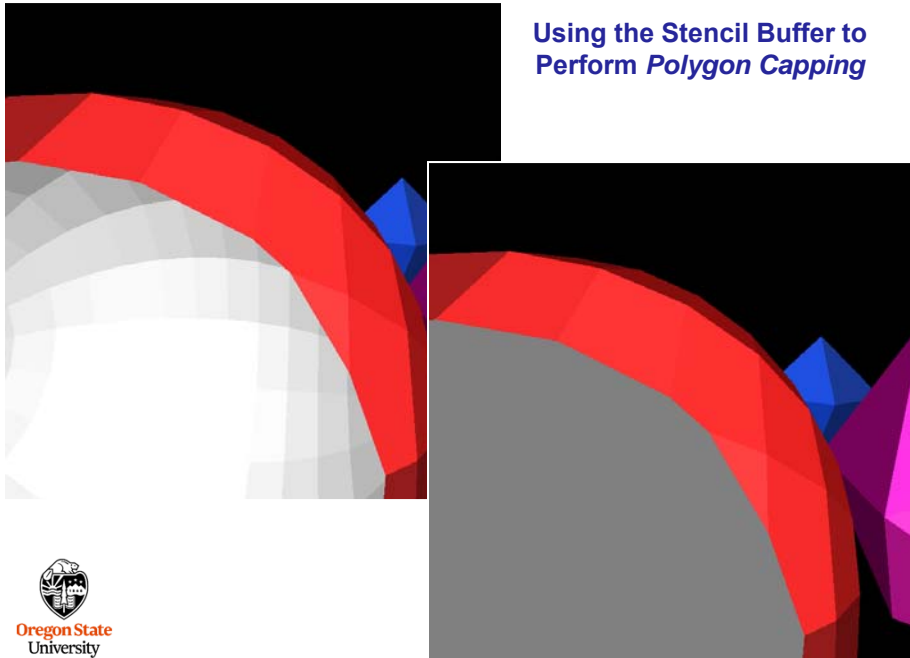
Using the Stencil Buffer to Create a *Magic Lens* for Volume Data

5



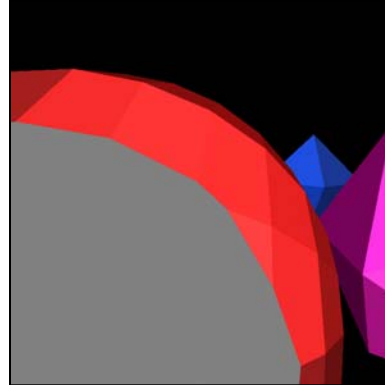
Using the Stencil Buffer to Perform *Polygon Capping*

6



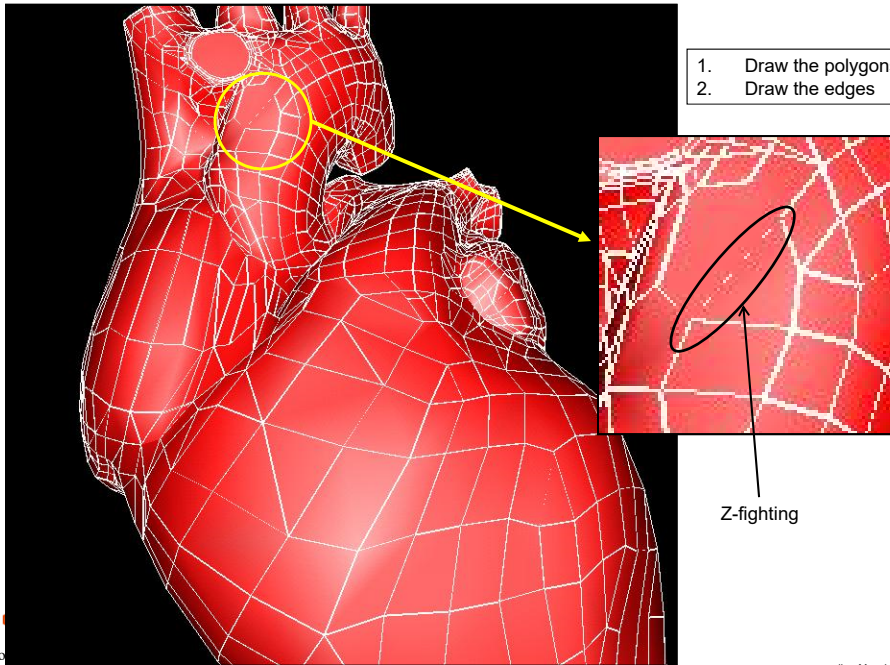
Using the Stencil Buffer to Perform *Polygon Capping*

1. Clear the SB = 0
2. Draw the polygons, setting SB = ~ SB
3. Draw a large gray polygon across the entire scene wherever SB != 0



Outlining Polygons the Naïve Way

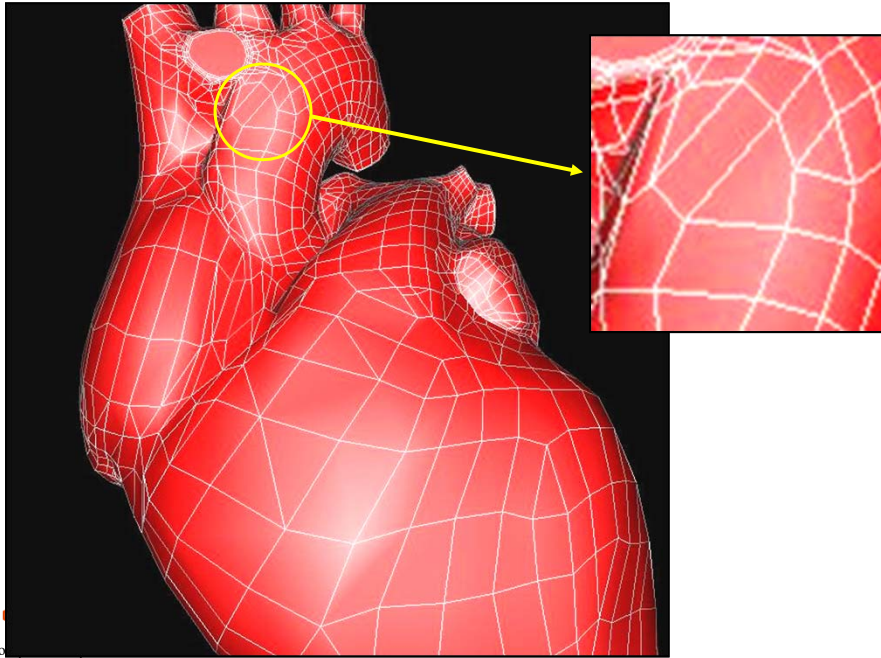
1. Draw the polygons
2. Draw the edges



Z-fighting

Using the Stencil Buffer to Better Outline Polygons

9



Using the Stencil Buffer to Better Outline Polygons

10

```
Clear the SB = 0
```

```
for( each polygon )
```

```
{
```

```
    Draw the edges, setting SB = 1
```

```
    Draw the polygon wherever SB != 1
```

```
    Draw the edges, setting SB = 0
```

```
}
```

Before

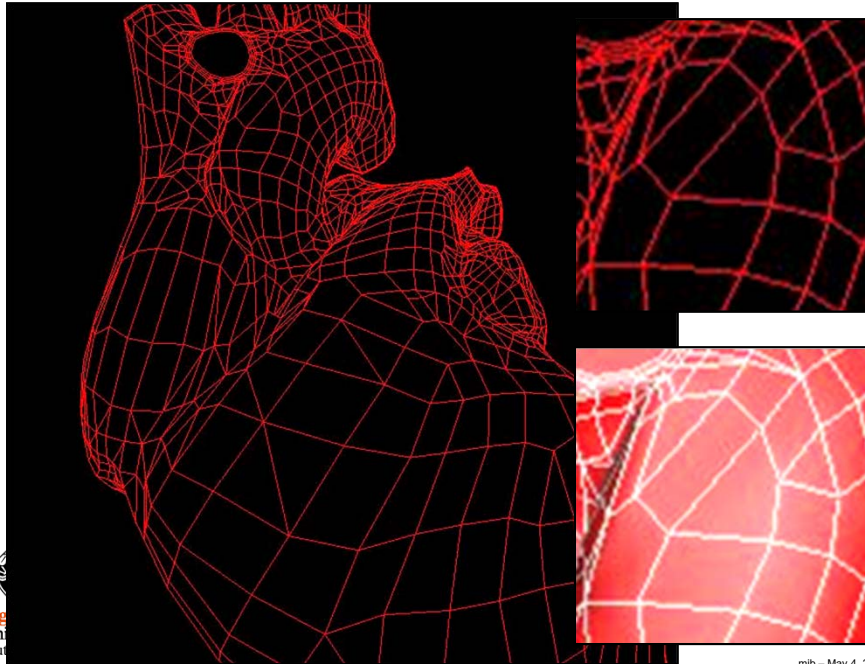


After



Using the Stencil Buffer to Perform *Hidden Line Removal*

11



12

