

News

29.3.2023.

Rasterization: clipping

- Can't we use whole triangle instead of clipping it if it is partially inside view volume?

“As long as our triangle coverage test is reliable, we don't need to clip against the left, right, top and bottom planes at all!”

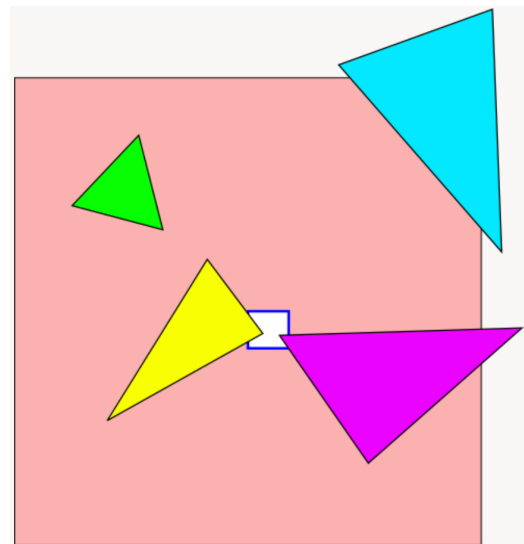
...

“That test is usually done in integer arithmetic with some fixed precision. And eventually, as you move say one triangle vertex further and further out, you'll get integer overflows and wrong test results.”

“[Therefore,] clip triangles eventually, just as they're about to go outside the safe range where the rasterizer calculations can't overflow”

...

“clip against the guard-band clip planes which are chosen so that after the projection and viewport transforms, the resulting coordinates are in the safe range”



Rasterization: clipping and more

- **Clipping:**

- Intuition: <https://fgiesen.wordpress.com/2011/07/05/a-trip-through-the-graphics-pipeline-2011-part-5/>
- Intuition: <https://www.gamedeveloper.com/business/in-depth-software-rasterizer-and-triangle-clipping>
- OpenGL clipping: https://www.khronos.org/opengl/wiki/Vertex_Post-Processing
- Line clipping: https://en.wikipedia.org/wiki/Cohen%E2%80%93Sutherland_algorithm
- Polygon clipping: https://en.wikipedia.org/wiki/Sutherland%E2%80%93Hodgman_algorithm

- **Rasterization pipeline:**

- <https://fgiesen.wordpress.com/2013/02/17/optimizing-sw-occlusion-culling-index/>
- OpenGL: https://www.khronos.org/opengl/wiki/Rendering_Pipeline_Overview
- OpenGL: <https://registry.khronos.org/OpenGL/specs/gl/glspec46.core.pdf#page=475&zoom=100,168,741>

Camera: look at

- What if `rand_vector == forward`?

`Forward = normalize(From - To)`

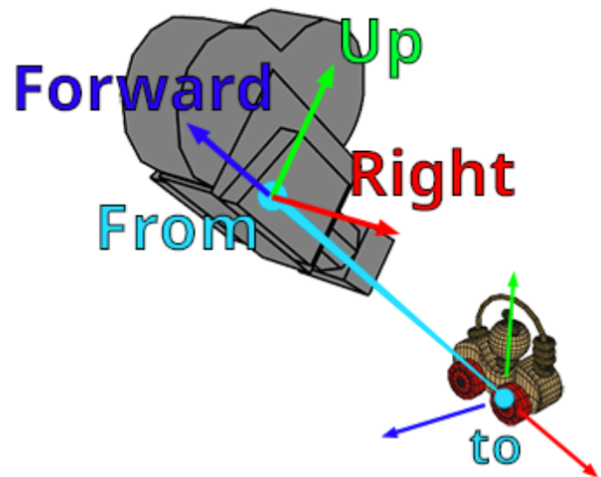
`Right = crossProduct(randomVec, Forward)`

- **RandomVec = (0,1,0) - not completely random!**

`Up = crossProduct(forward, right)`

Solutions:

- Make sure to add check `rand_vector` close to `forward`. If this is true, pick another vector by hand. Otherwise `crossProduct()` will fail.
- Use quaternions → stable orientation description
- Require additional up vector as input next to `from` and `to`
 - <https://registry.khronos.org/OpenGL-Refpages/gl2.1/xhtml/gluLookAt.xml>
 - <https://glm.g-truc.net/0.9.5/api/a00176.html#ga454fdf3163c2779eeeeeb9d75907ce97>



www.scratchapixel.com

<i>Right_x</i>	<i>Right_y</i>	<i>Right_z</i>	0
<i>Up_x</i>	<i>Up_y</i>	<i>Up_z</i>	0
<i>Forward_x</i>	<i>Forward_y</i>	<i>Forward_z</i>	0
From _x	From _y	From _z	1

Hair modeling

- Practical:

- NVIDIA:

- <https://developer.nvidia.com/gpugems/gpugems2/part-iii-high-quality-rendering/chapter-23-hair-animation-and-rendering-nalu-demo>

- NVIDIA: <https://developer.nvidia.com/content/hairworks-11-release>

- Cem Yuksel: <http://www.cemyuksel.com/courses/conferences/siggraph2010-hair/>

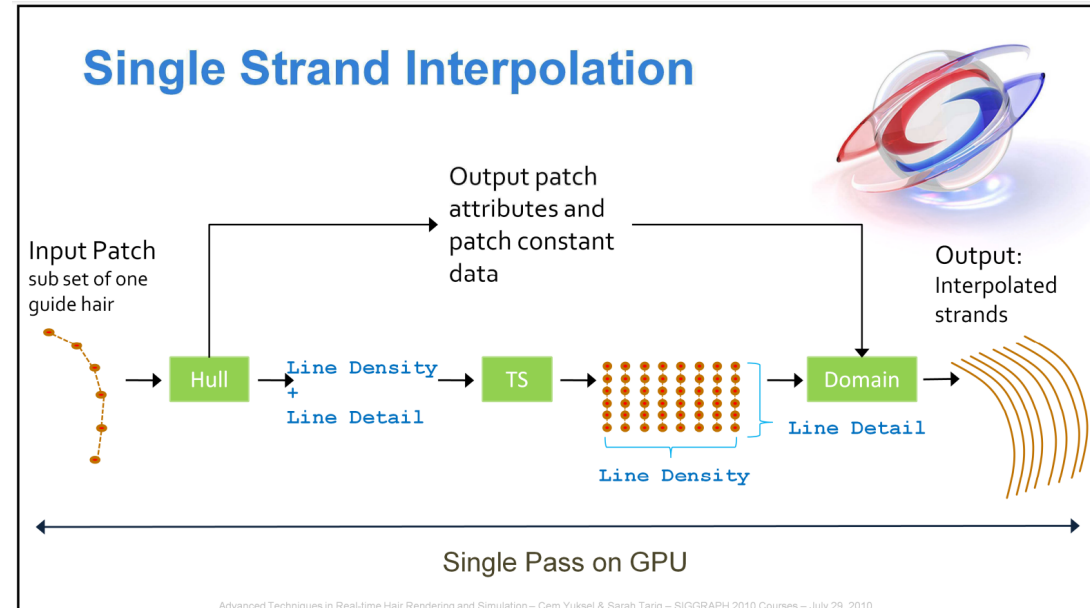
- Research:

- <https://web.engr.oregonstate.edu/~mjb/cs557/Projects/Papers/HairRendering.pdf>

- <https://gamma.cs.unc.edu/HAIRSURVEY/WBKMCL07.pdf>

- <http://www.cemyuksel.com/research/gihair/>

- <https://diglib.eg.org/bitstream/handle/10.2312/sr20191215/001-008.pdf>



Key hair strands (guide hair) is generated on application stage. Then it is interpolated and generated on GPU and finally rendered.

Lecture evaluation

- Personal lecture evaluation process
 - Link: <https://forms.gle/sf5U3qyWgh7h9dra7>
 - Deadline: 8.4.2023.
- DHBW lecture evaluation process has started (link was sent by the system)
 - Deadline: 8.4.2023.

Project

- **Deadline: 8.4.2023., 23:59h**
- Mail:
 - lovro.bosnar@itwm.fraunhofer.de
 - lovro.bosnar1@gmail.com
 - If you don't receive confirmation of your mail/submission under 24h please check mail address again and send again. I will at least confirm each mail/submission that I receive.
- Questions?