MEDICAL RAY-TRACING IN VR

Jeroen Stinstra, Kevin Parker, Ian Williams, November 2019, Washington DC
Quick overview of Medical Ray-tracing

Challenges with Ray-tracing in Virtual Reality

Solutions to make ray-tracing achievable in Virtual Reality
MEDICAL VOLUME RAY-TRACING

CT image (DICOM) → Ray-traced volume rendering
MEDICAL VOLUME RAY-TRACING PIPELINE

Image processing pipeline

Deep Learning Segmentation
Semi-manual Segmentation

Anisotropic Diffusion Filter

Density Volume
Label Map

Transfer Function
Opacity of Volume
Material Properties

Camera
Light

Ray-tracing pipeline
Scatter point
Stochastic Path-tracing pipeline in CUDA

Multiple jittered rays per pixel with different probing depths

Different scatter points based on probing depth and opacity profile

Acceleration Structure
Opacity mask
Mipmapped stack of bit masks
Accelerate traversal through empty spaces
HYPER REALISM VOLUME RENDERING

**Tricubic interpolation** of density to avoid voxel artifacts

Single bounce path tracing for **soft shadows**

**Material model** with diffuse and specular, roughness

**Use of labelmap** to map different material properties per region
RAY-TRACED VOLUME RENDERING IN VR
**VR CHALLENGE #1 - FRAMERATE**

**NEEDED:** VR Recommended Resolution: 2 x 2460x2740 @90FPS

To match resolution inside center area one needs to render higher resolution than physical screen resolution when assuming regular pixel layout.

Image is warped down to match screen resolution.
TRADITIONAL RASTER SOLUTIONS

VRWorks

- VARIABLE RATE SHADING
- LENS MATCHED SHADING
- VR SLI
- SINGLE PASS STEREO
- MULTI-VIEW RENDERING
- MULTIRES SHADING
RAY TRACING SOLUTIONS

PROS:

In ray-tracing / compute shaders one can steer the direction of rays per pixel

CONS:

Ray tracing requires a lot of rays to be fired or at least enough for denoisers to remove noise from the images.
**SOLUTION: MEDICAL RAY-TRACING IN VR**

**Render in Warped Space**
- In ray-tracing/compute shader: per pixel control of ray direction
- Track eye movements

**Denoise in warped space**
- Keep discrete history buffers to accommodate frequent camera movement for reprojection
- Spatial denoise in warped space

**Render on multiple RTX cards**
- Target 2 GPUs
- Use CUDA to transfer images quickly Between cards over NVLink
FOVEATED/WARPED RENDERING

Unique VR feature:

Only one eye is looking at the image

We only see high details in a very small area of the screen

https://en.wikipedia.org/wiki/Fovea_centralis
Image by Vanessa Ezekowitz - Hand-drawn based on File:AcuityHumanEye.jpg by Hans-Werner Hunziker, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=7327065
FOVEATED/WARPED RENDERING

Ray-traced image in warped space

Unwarped image used in VR

Original Ray traced volume

Actual Pixel Size

Virtual Reality Rendering

Flatland Rendering
WARPED RENDERING

Ray-tracing advantage:
We can send rays in arbitrary directions per each pixel

- We can even add jitter per pixel

Matched to HMD screen resolution

Render Density
EYE TRACKING

Some modern HMDs can track where you are looking with eye tracking.

We can adjust warp *per frame* and *per eye* to match where you are looking by moving the foveated region to where you are looking.
FOVEATED/ RAY-TRACED EYE TRACKING IN VR

- Ray-traced VR with eye tracking
- Running on 2 x RTX 6000 cards at 90Hz

Unwarped imaged resolution:
- 2 x 2460 x 2740 pixels

Warped resolution:
- 2 x 615 x 685 pixels

High resolution area:
- 2 x 369 x 411 pixels
DENOISING WARPED IMAGES

Current frame + 7 previous frames

Record Warp Parameters + camera Position per frame per eye

About 13 samples per pixel per frame

Temporal filter

Find 3D location pixel in Current frame

Reproject location into previous frame and find pixel

Mix in previous pixel if depth range matches

Spatial Filter

Use weighted neighboring pixels (radius 3) to reduce noise

Unwarp Filter

Submit to HMD
In VR a lot of noise on hard flat surfaces was the most distracting.

Reprojection depth (num images)

Reproject previous frames onto latest every frame.
DENOISING WARPED IMAGES

Number of rays per pixel for each frame at 90Hz: 10 - 20 samples
Number of potential pixels used from reprojection: 1 - 8 samples
Number of pixels in spatial filter: 7 x 7 grid of samples
500 - 5000 samples of information
RAY SCHEDULING

Multiple path-traces per pixel

Stochastically different paths per pixel scheduled on neighboring threads

<table>
<thead>
<tr>
<th>Samples in parallel</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.75</td>
</tr>
<tr>
<td>2</td>
<td>4.05</td>
</tr>
<tr>
<td>4</td>
<td>4.80</td>
</tr>
<tr>
<td>8</td>
<td>6.75</td>
</tr>
<tr>
<td>16</td>
<td>10.20</td>
</tr>
</tbody>
</table>
ADAPTIVE CONTROL

To deal with varying degrees of complexity in scenes:

- Determine render time for previous frame
- **Estimate optimal rays per pixel** by adjusting rays per pixel
- Schedule rays for all pixels in batches for entire frame of 11ms
- Draw frame
SCHEDULING CUDA AND OPENVR

Problem:

Scheduling CUDA ray tracing as one big kernel is fastest

Render times in ray tracing vary depending on scene complexity

When render times exceed 11ms by too much, OpenVR cannot interrupt CUDA

Result:

VR Experience has hiccups
SMOOTH VR SCHEDULING

Render Batch 1

Render Batch 2

Render Batch 3

Render Batch 4

Render Batch 5

Render Batch 6

Render Batch 7

Render Batch 8

Render Batch 9

Stream 1

Stream 2

Stream 3
Most rays are generated in central image where foveated region moves

We use CUDA to schedule rendering in a 16 x 16 grid of tiles and use a lookup table to figure out rays per pixel when rendering.
CONCLUSIONS

Using rendering/denoising in warped space with eye tracking we need to render about \(1/16\) of the amount of pixels.

The gained speedup can be used to render more paths per pixel to reduce the noise introduced by the stochastic ray tracing.

There is even budget for features like higher-order interpolation kernels to reduce voxel artifacts to get closer to hyper-realistic rendering.

This technique can be used for common dimensions of medical images.