MEDICAL RAY-TRACING IN VR

Jeroen Stinstra, Kevin Parker, Ian Williams, November 2019, Washington DC

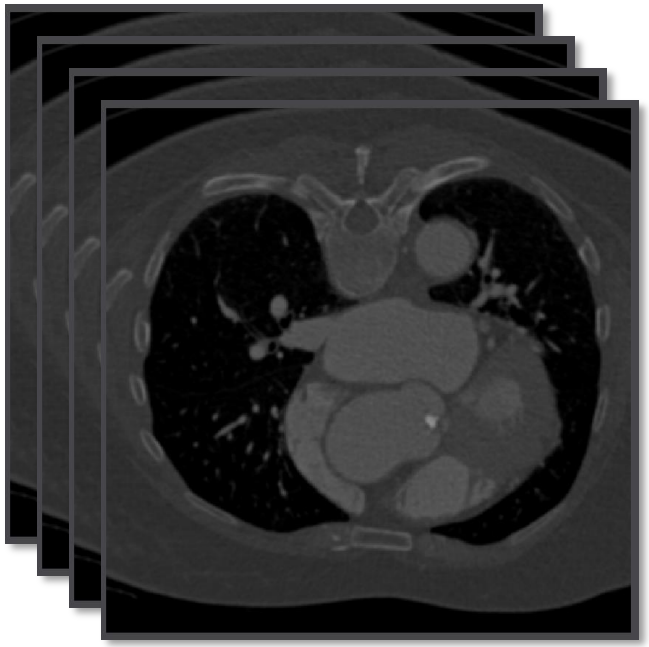
CONTENT

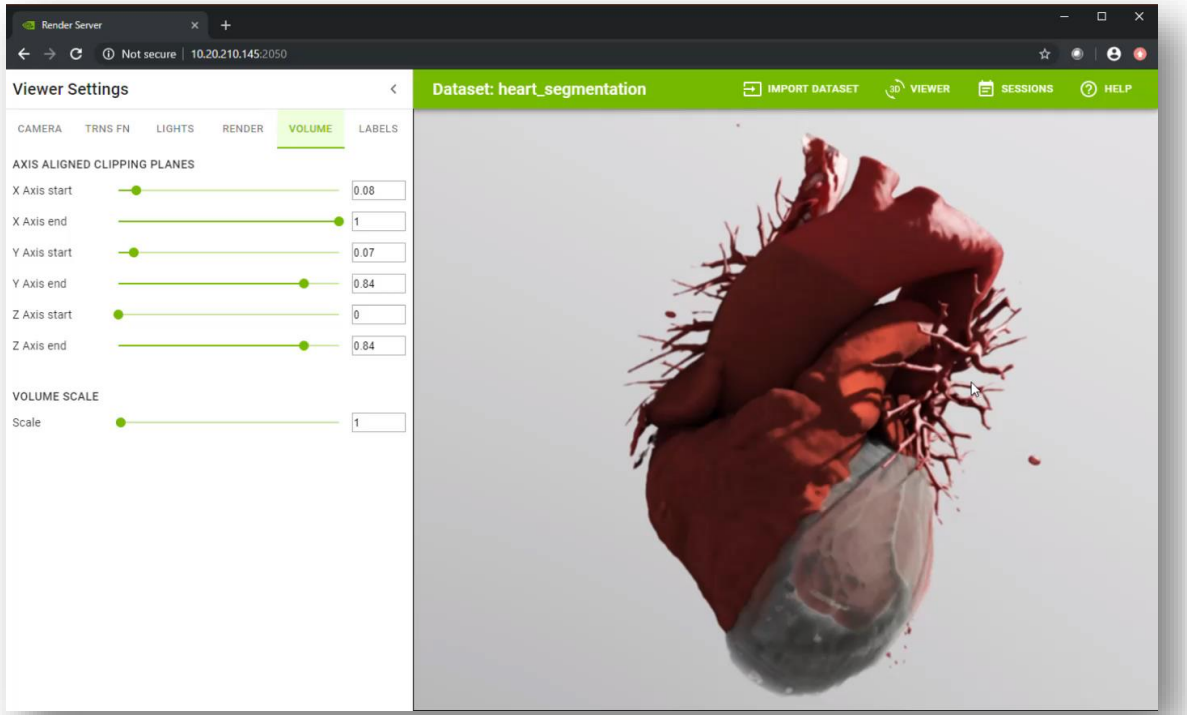
Quick overview of **Medical Ray-tracing**

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

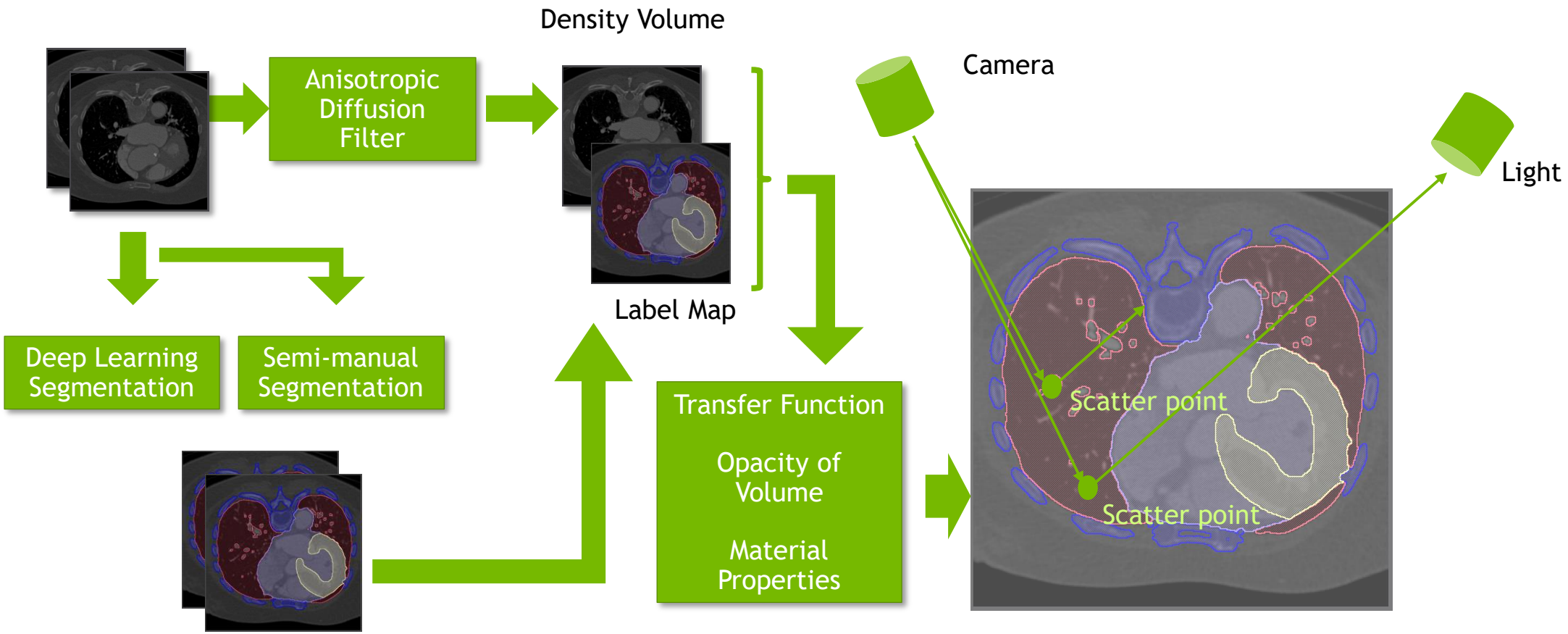
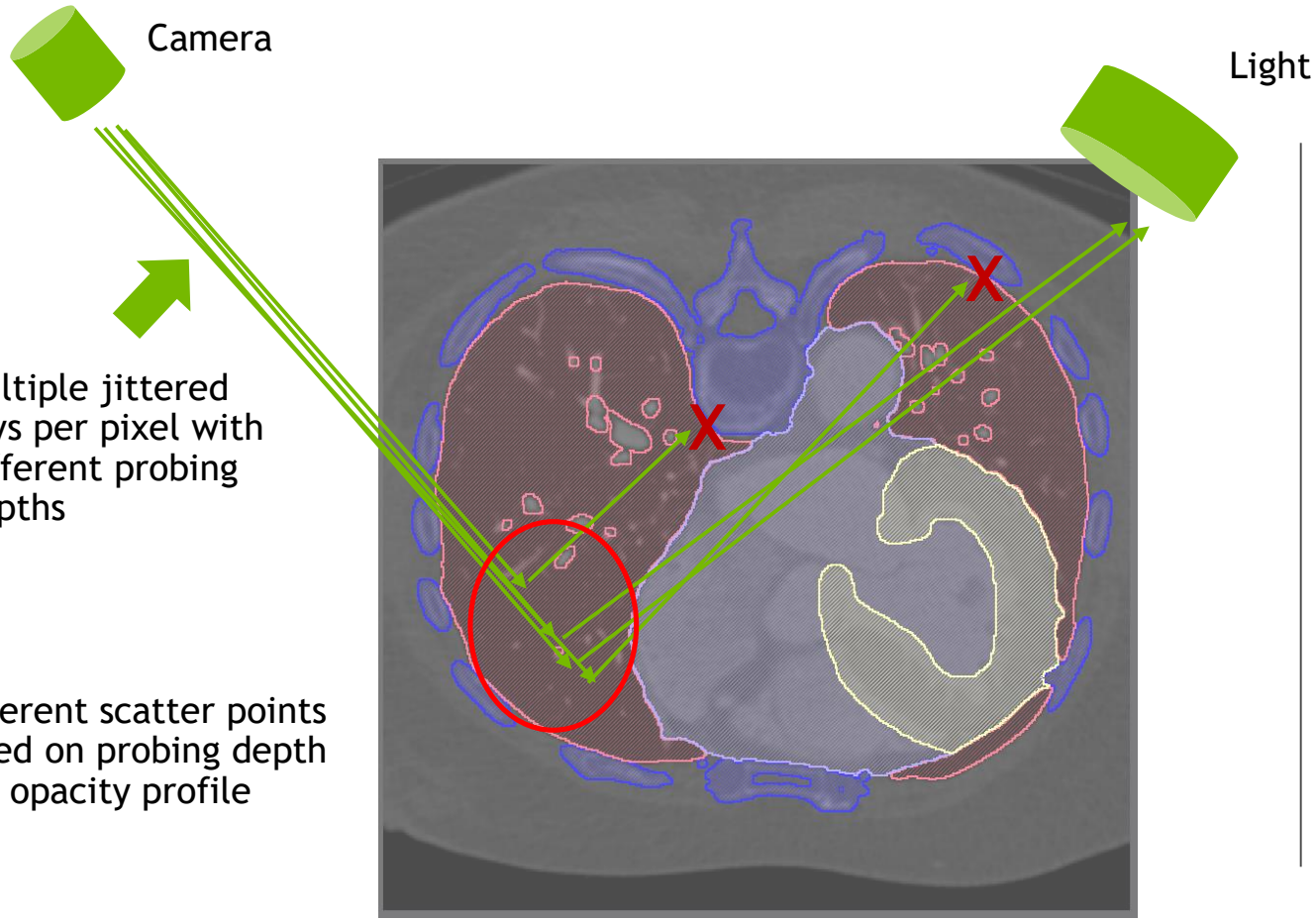


Image processing pipeline

Ray-tracing pipeline

MEDICAL VOLUME RAY-TRACING PIPELINE 2

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

Mipmapped stack of bit masks

Accelerate traversal through empty spaces

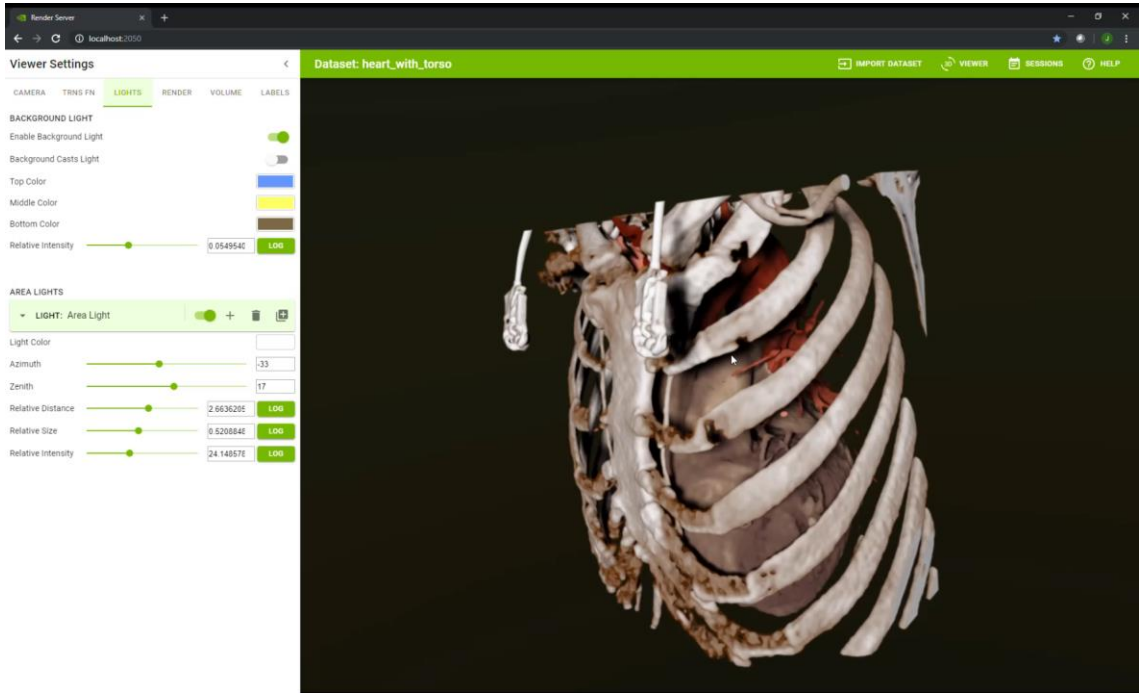
HYPER REALISM VOLUME RENDERING

Tricubic interpolation of density to avoid voxel artifacts

Single bounce path tracing for **soft shadows**

Material model with diffuse and specular, roughness

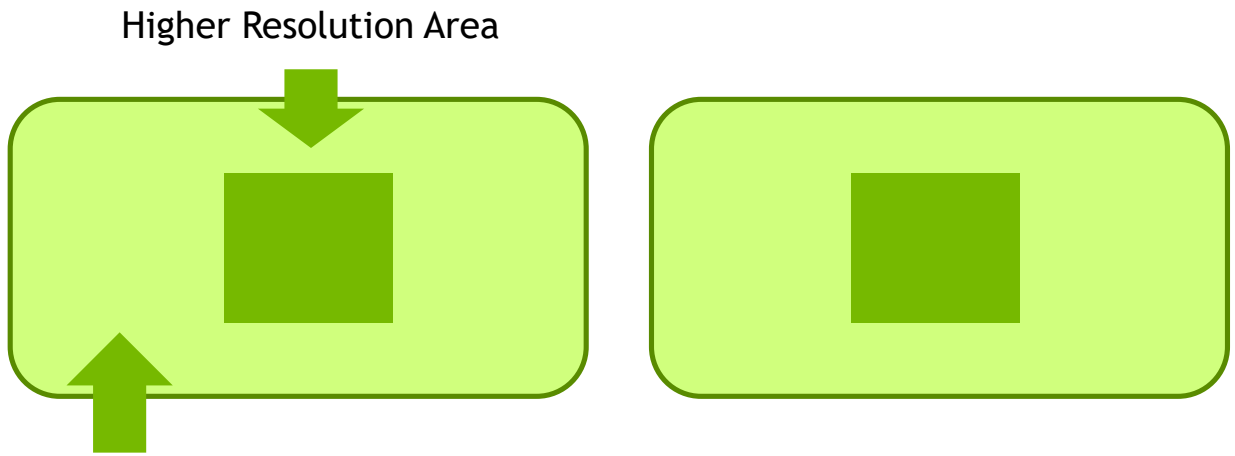
Use of labelmap to map different material properties per region



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

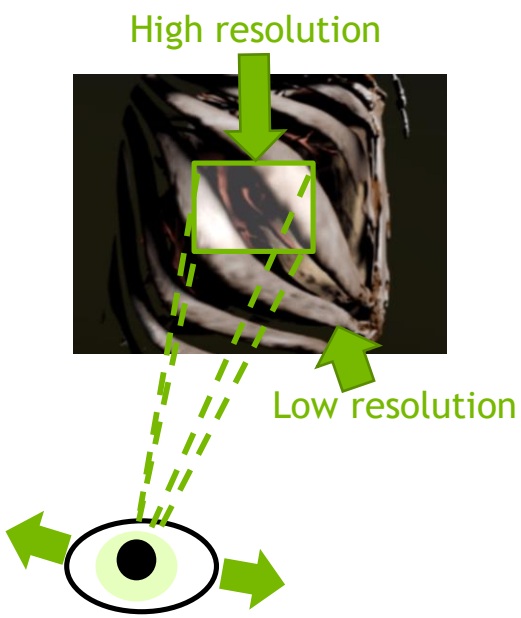
CONS:

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



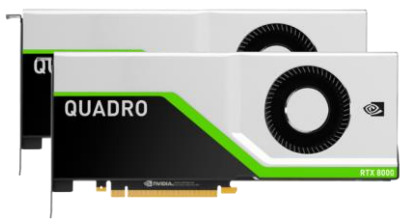
Track eye movements

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

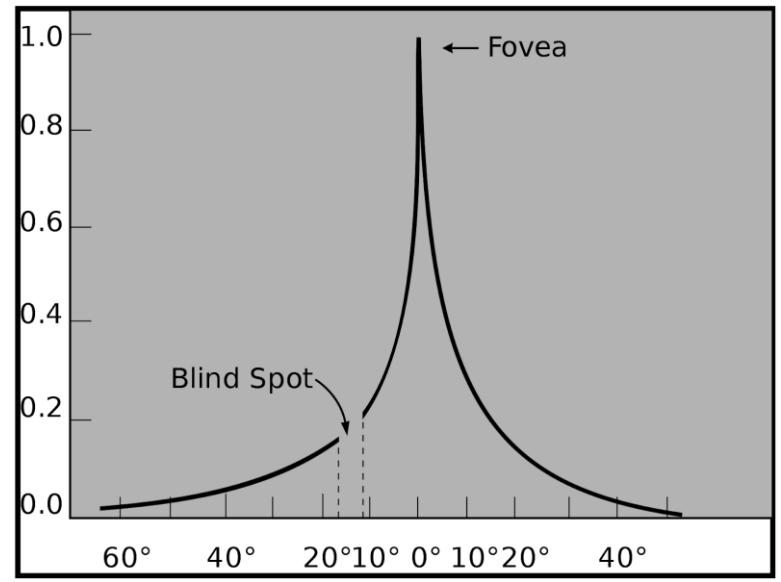
Use CUDA to transfer images quickly Between cards over **NVlink**

FOVEATED/WARPED RENDERING

Unique VR feature:

Only **one eye** is looking at the image

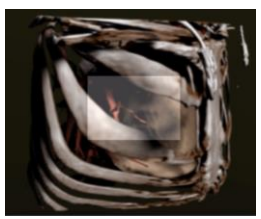
We only see **high details in a very small area** of the screen



https://en.wikipedia.org/wiki/Fovea_centralis
mage 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

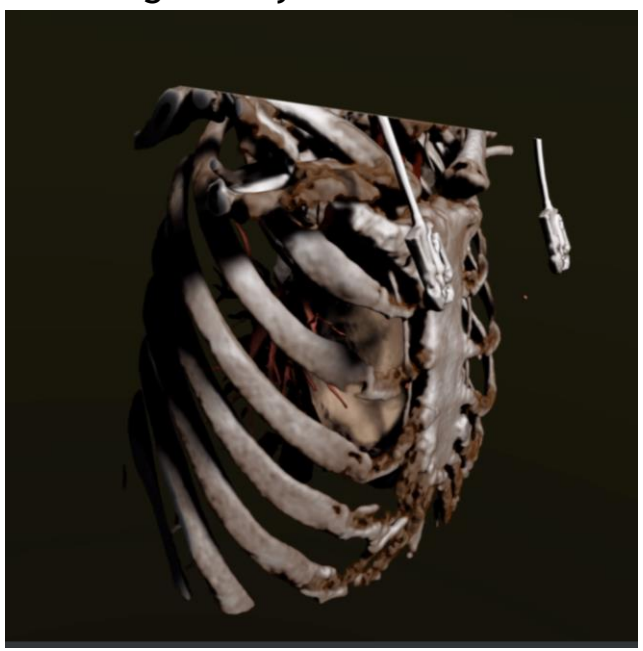


Actual Pixel Size

Unwarped image used in VR

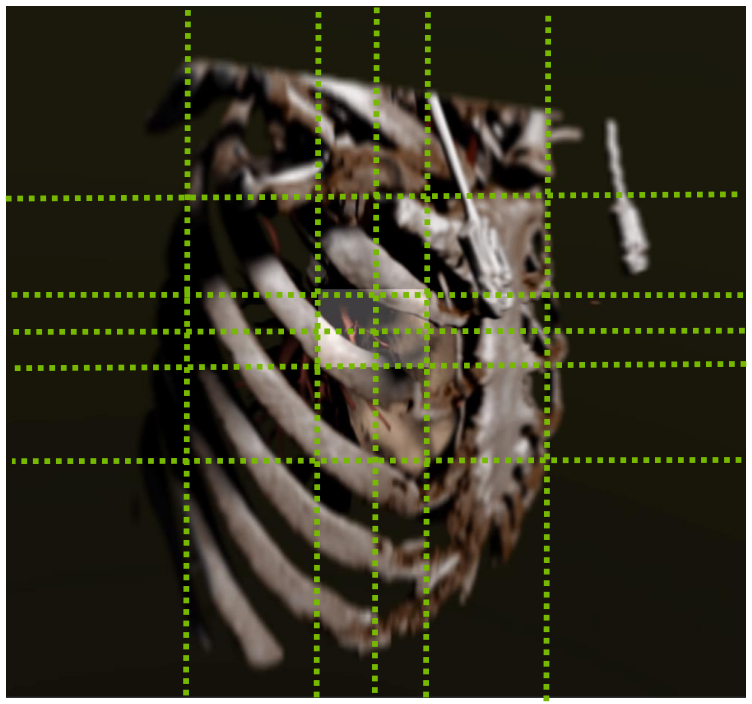
Virtual Reality Rendering

Original Ray traced volume

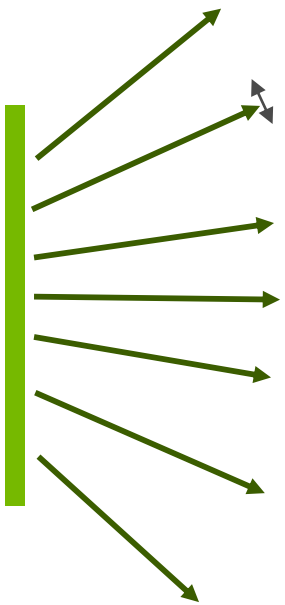


Flatland Rendering

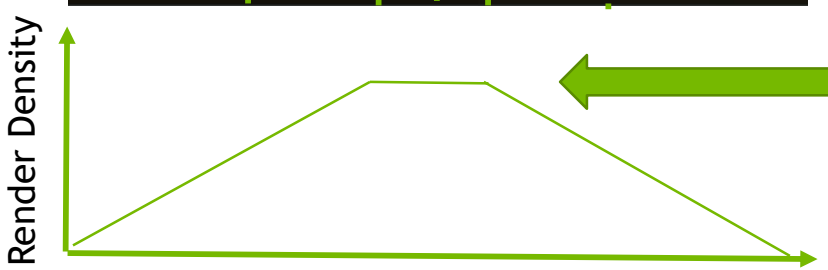
WARPED RENDERING



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

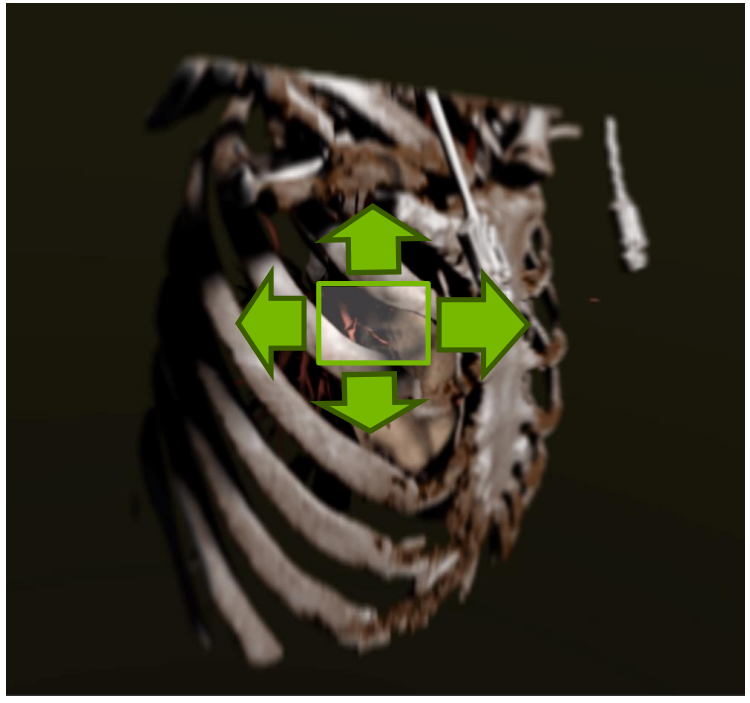


We can even add jitter per pixel



Matched to HMD screen resolution

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

About 13 samples
per pixel per frame

Current frame +
7 previous frames

Record Warp Parameters +
camera Position per frame
per eye

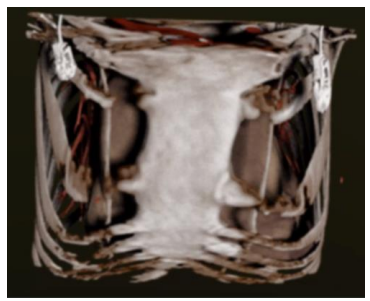
Temporal filter

Find 3D location pixel in
Current frame

Reproject location into
previous frame and find pixel

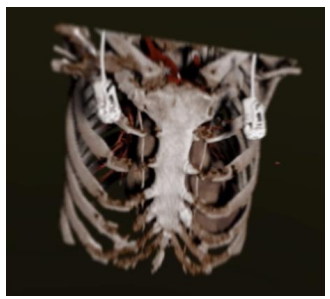
Mix in previous pixel if depth
range matches

Spatial Filter



Use weighted neighboring
pixels (radius 3) to reduce
noise

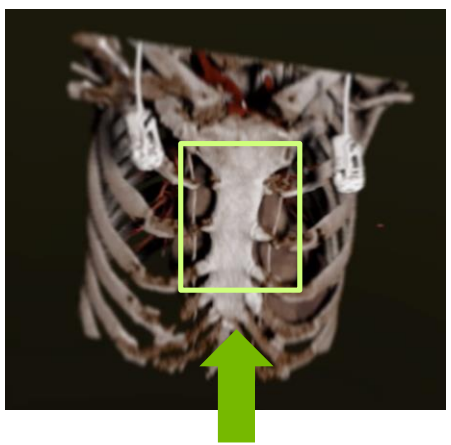
Unwarp Filter



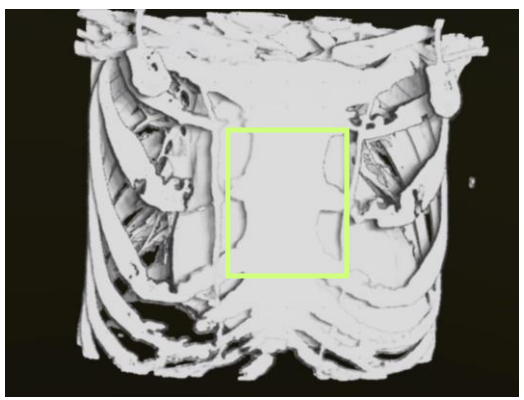
Submit to HMD

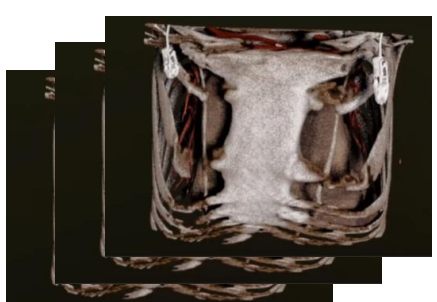
TEMPORAL WARPED REPROJECTION

Reprojection depth (num images)



In VR a lot of noise on hard flat surfaces was the most distracting





Reproject previous frames onto latest every frame

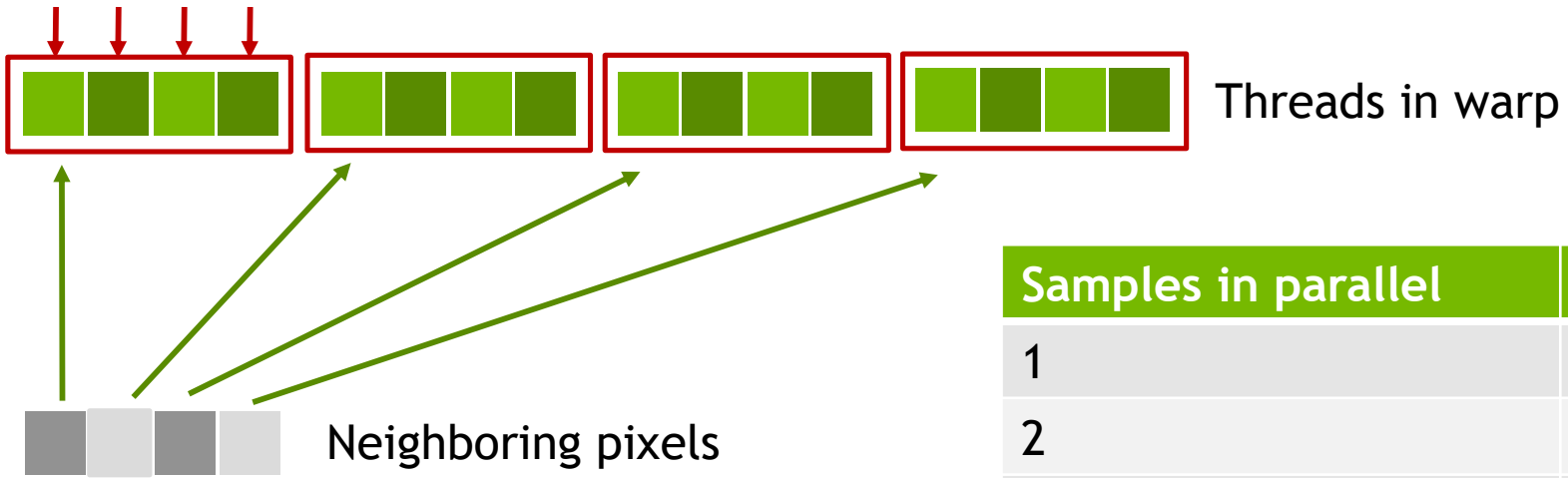
DENOISING WARPED IMAGES

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
	x
	500 - 5000 samples of information

RAY SCHEDULING

Multiple path-traces per pixel

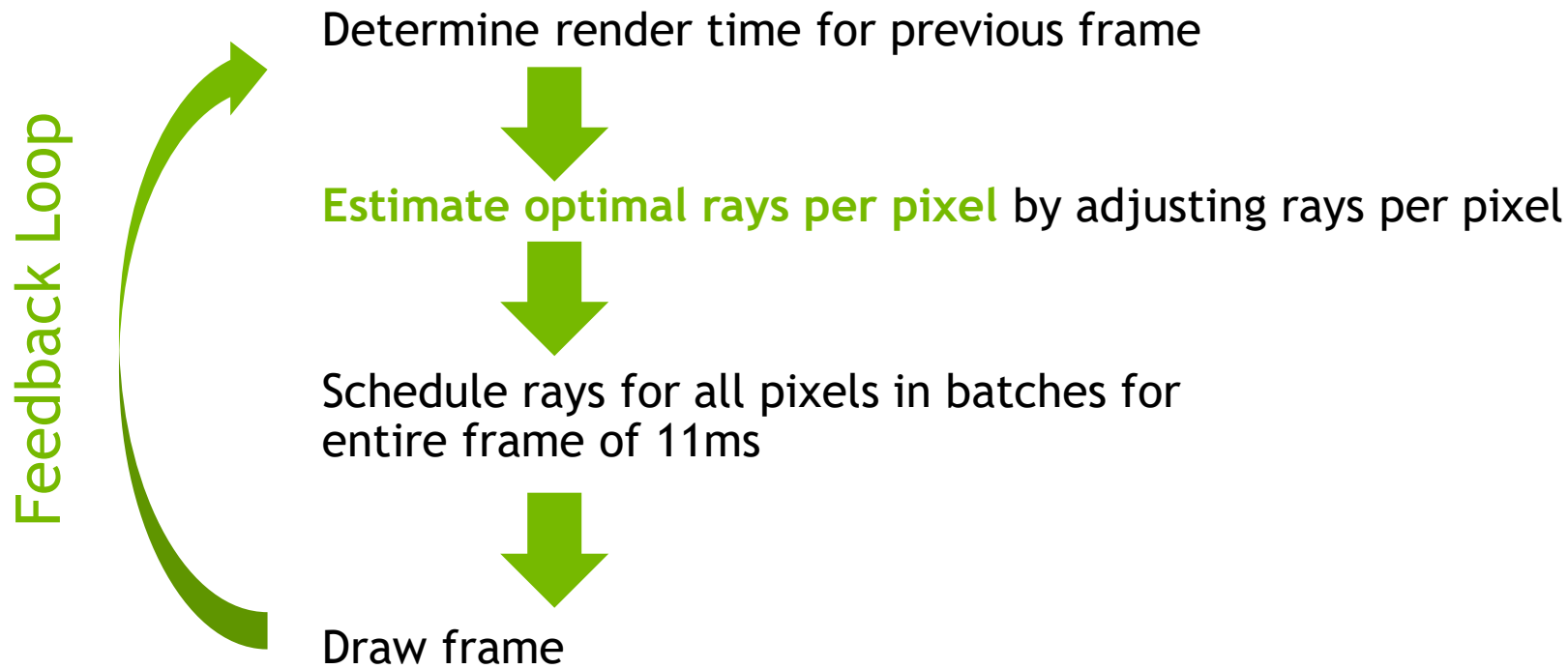
Stochastically different paths per pixel scheduled on neighboring threads



Samples in parallel	Time (ms)
1	3.75
2	4.05
4	4.80
8	6.75
16	10.20

ADAPTIVE CONTROL

To deal with varying degrees of complexity in scenes:



SCHEDULING CUDA AND OPENVR

Problem:

Scheduling CUDA ray tracing as **one big kernel** is **fastest**

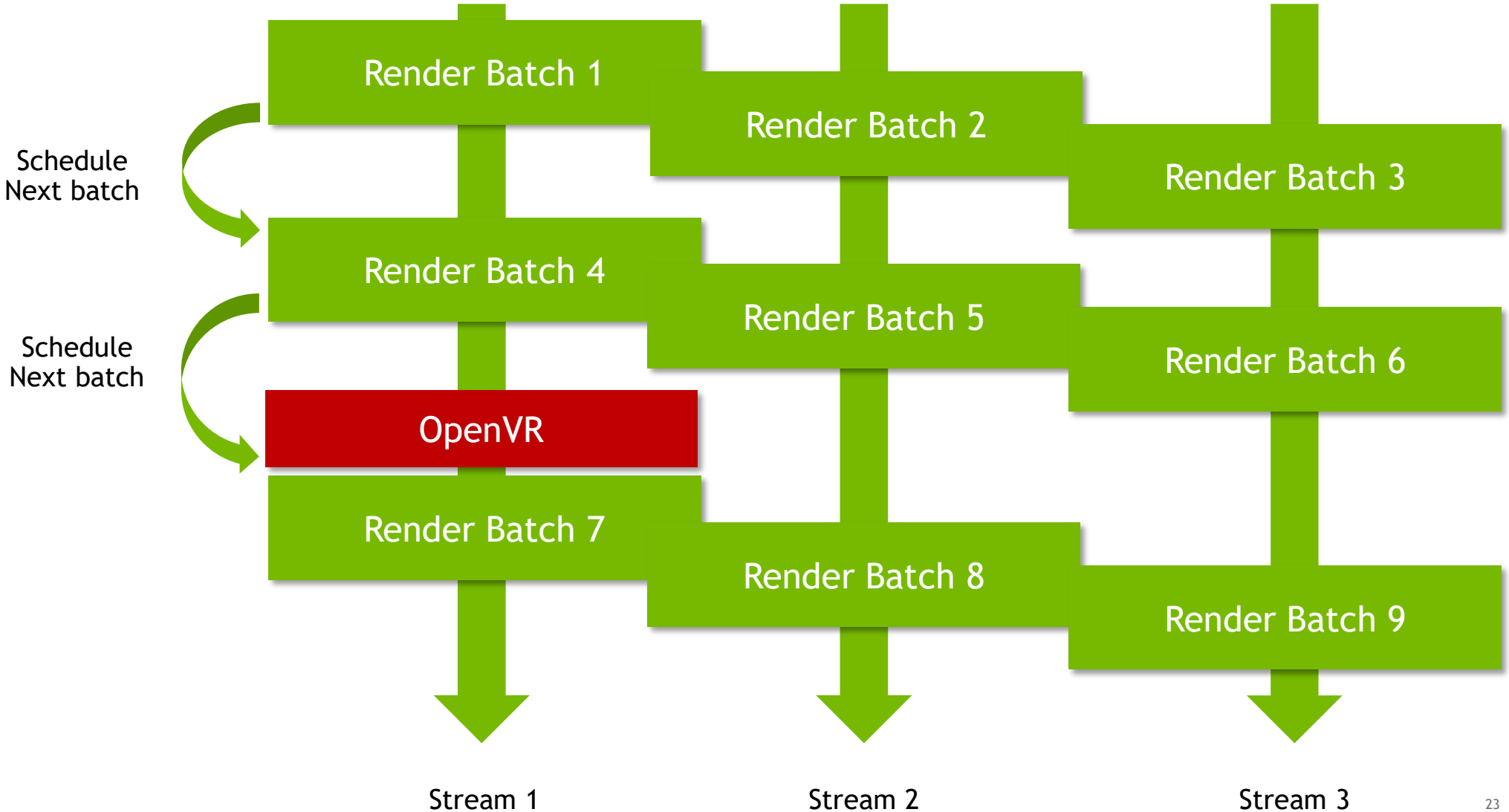
Render times in ray tracing **vary depending** on scene complexity

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

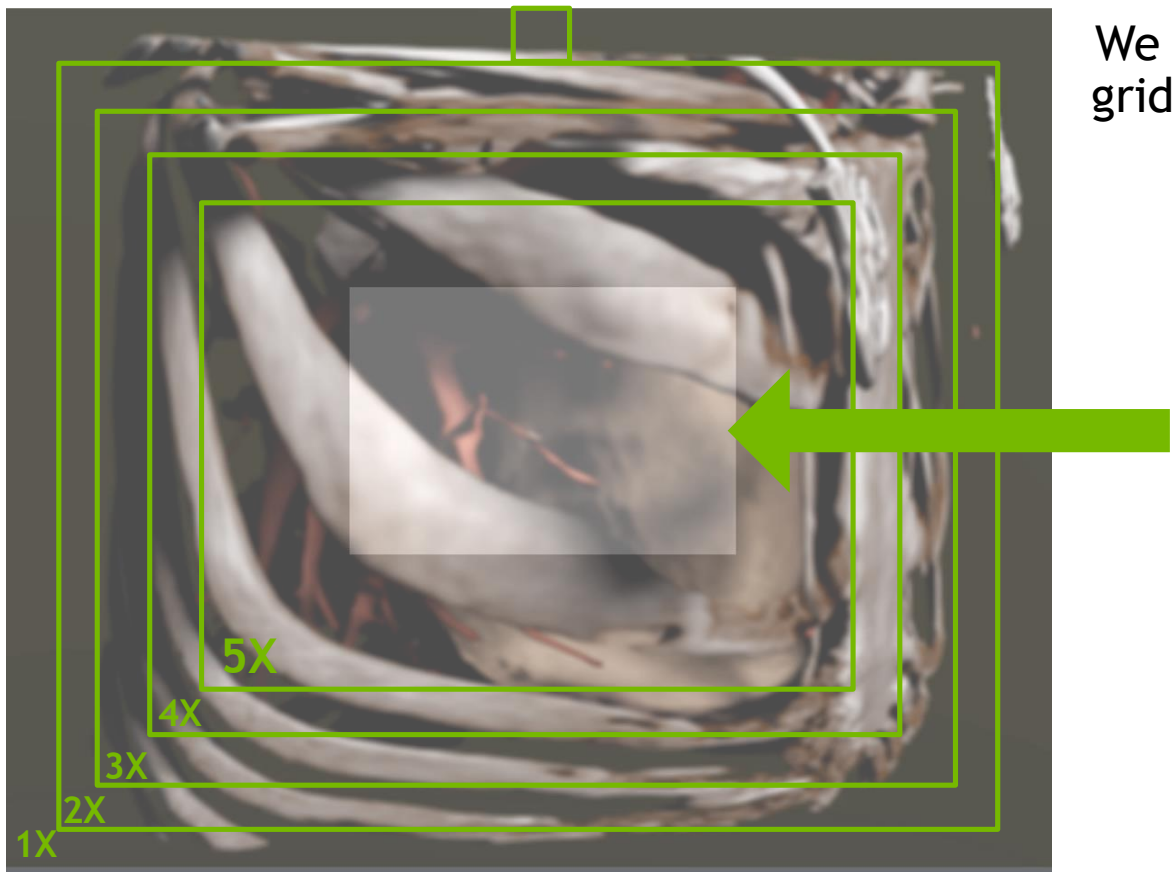
Result:

VR Experience has hiccups

SMOOTH VR SHEDULING



RAYS PER PIXEL IN VR



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

Most rays are generated in central image where foveated region moves

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.