

# Path Tracing in DOOM: The Dark Ages

Billy Khan, Director of Engine Technology (id Software)  
Matt Stack, Senior DevTech Engineer (NVIDIA)



# Speakers

---

## **Billy Khan**

Director of Engine Technology - id Software

X : @billykhan

## **Matt Stack**

Senior DevTech Engineer - NVIDIA



**NVIDIA**

# Agenda

- Motivation
- Path Tracing Integration
- Denoising
- Opacity Micro-Maps
- Performance and VRAM Strategies
- DLSS4
- Closing

# Motivation



# idTech8 in DOOM: The Dark Ages



## • idTech8 Goals

- Performance  $\geq$  60hz on most platforms
- Iteration / Iteration / Iteration
  - WYSIWYG Development
  - No More Baked Processes
  - Faster Cross Departmental Collaboration
  - Reduce Time Consuming Custom Markups
- Better Pixels – Not More Pixels
- Fully Dynamic Lighting
- 5x -10x Increase in World Size and World Detail compared to idTech7
- Extremely Fast Load Times
- Many more characters on screen



# idTech8 in DOOM: The Dark Ages

- Building on the foundations of idTech7
  - DOOM Eternal / idTech7 RayTracing Update
    - June 2021
  - Simplify / Unify previous systems
  - Can we get close to Path Tracing Results?
  - How do we speed up workloads?
- Hardware Raytracing
  - Almost a decade old
  - All current generation consoles and handhelds have raytracing support
  - Can be used for more than just cool imagery
    - Gameplay, Material Queries (sound/decals), Particle Collisions, streaming, mesh on surface spawning, tools, etc.
- Exciting and Extensive ongoing practical research in the industry
  - DDGI [Majercik2019]
  - Surfels [Halen2021]
  - Screen-Space Radiance Caching [Wright2021]
  - ReSTIR GI [Ouyang2021]



# idTech8 in DOOM: The Dark Ages



- **idTech8**

- Native Ray Tracing
  - Global illumination
  - Reflections
  - Material hit detection for dynamic gameplay features
  - Ray tracing productivity tooling to enhance content creation iteration times
- Multi-Composite materials
- Sector Streaming
- Vegetation
- Realtime Light Scattering Volumetrics
- Destructible Environments
- Cinematic Grade Tonemapping ( Aces 1.3 ) for SDR and HDR
- GPU Fluid Wind Simulation
- Dynamic Cloth
- Hair Strands
- Enhanced Job Scheduler
- Enhanced GPU Particles



# Show Surface Info Debugging



# Show Surface Info Debugging



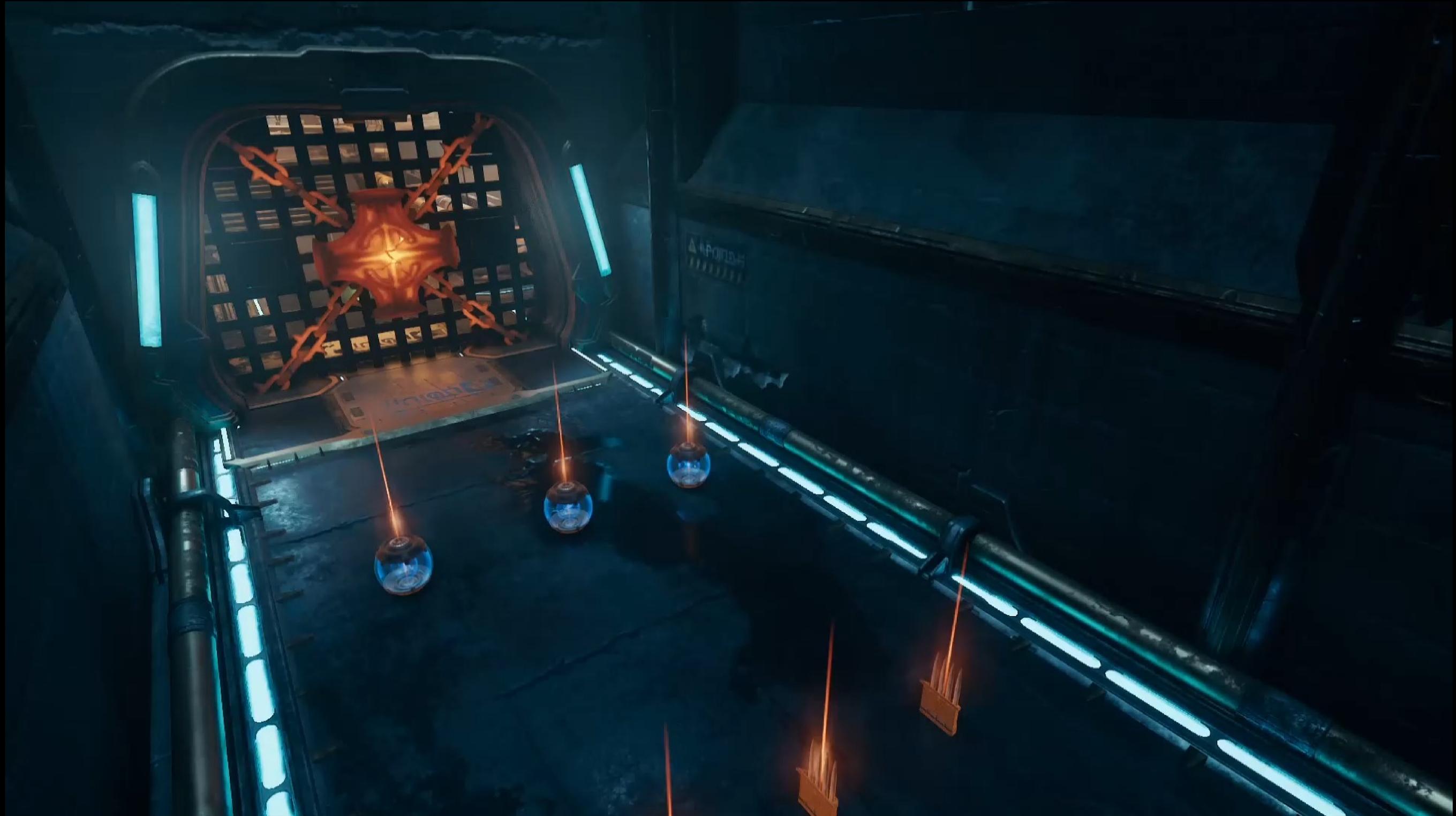
# Pixel Light Debugging



# Pixel Light Debugging



# Path Trace Ray Debugging



# Path Trace Ray Debugging



# idTech8 in DOOM: The Dark Ages

- **Fast As Hell : idTech8 Global Illumination**
  - Tiago Sousa, id Software @ Siggraph 2025
- **Later Today**
  - **Visibility Buffer and Deferred Rendering in DOOM : The Dark Ages**
    - Dominik Lazarek, id Software
    - Philip Hammer, id Software
  - **Variable Rate Shading in DOOM : The Dark Ages**
    - Martin Fuller, ATG Microsoft
    - Philip Hammer, id Software



# Why Path Tracing for DOOM?

- **Push the envelope in visuals and lighting for high end GPUs**
  - Understand the practical utility, return-of-investment, and validity of techniques in a shipping product for players
- **Provide more options to players to customize their experience**
  - Add flexibility and more options for players' personal preferences
- **Natural next step in image quality improvements**
  - Builds directly on DOOM's existing ray tracing renderer, extending it to full path tracing for unified lighting
- **More realistic lighting**
  - Accurate simulation of light transport accounts for color bleeding, caustics, and indirect effects
- **Multi-bounce lighting for GI**
  - Captures light interreflections between surfaces for believable room-scale illumination
- **Higher accuracy for diffuse + specular indirect lighting**
  - Correctly resolves rough diffuse bounce and glossy specular reflections in the same framework
- **Nice physically based shadow penumbras**
  - Soft shadows naturally emerge from light size and geometry, no need for ad-hoc filters

# Starting from the Great Circle

- **Started with porting path tracing in Indiana Jones and the Great Circle**
  - Reused the path tracing integrator and debugging features that the MachineGames team had developed
  - Added some of our own path tracing tools to help debug complex lighting situations
  - Major Kudos to the team at MachineGames for their early efforts
- **Same engine, different versions**
  - Refactor code that had diverged between the version of Motor (idTech7 Fork) that Indiana Jones used and idTech8 in DOOM
  - Many code merges! It was important to make sure incoming code from DOOM's progress didn't blow up path tracer
    - Or vice versa
  - Refined and tuned various technologies (SER, OMMs, SHaRC cadence) and memory budgets for DOOM's content
- **Artistically different games**
  - Lighting and Shading is quite different between DOOM and Indiana Jones
    - DOOM's gritty, darker, cinematic environment is very different than Indy's interior and exterior environments
  - IdTech8's multi composite material system is considerably more advanced, so path tracer needed updates to deal with that

# Path Tracing End Goals

- **Path tracing should improve image quality while maintaining existing aesthetic**
  - There were some caveats, but it was mostly straightforward thanks to idTech's engine architecture and renderer
- **60 FPS or bust**
  - Players must experience path traced DOOM: The Dark Ages at 60 fps (or faster) for all hardware requirement levels
- **Performance considerations**
  - **Reduce ray counts**
    - Employ SHaRC to get more bounces for higher quality GI without tracing additional rays for every path
  - **Reduce shader invocations**
    - Use Opacity Micro Maps (OMM) to reduce Any Hit Shader (AHS) invocations when alpha-testing
  - **Reduce divergence**
    - Use Shader Execution Reordering (SER) and Opacity Micro Maps (OMM) to reduce divergence in thread execution and data access
- **Optimize VRAM usage**
  - Tune all the knobs available for path tracing to make sure VRAM usage did not approach eviction danger zone
  - Make sure path tracing's settings were in memory budget for all hardware requirement levels

# Path Tracing Integration

**DOOM**  
THE  
DARK AGES



# Integration Goals



- **No major changes to the existing renderer**
  - Path tracing was in development in parallel to the finishing touches on the game
  - Path tracing code needs to be siloed, can't introduce bugs that would stall dev on the art or game
- **Protect artist intention!**
  - Need to make sure the lighting tools that the artists use are represented in path traced lighting
- **Must maintain performance**
  - SHaRC covered in this section, SER and OMMs covered in a later section



# Lighting in Path Tracing



- **Point lights, spot lights, directional lights, and area lights**
  - The classics, now with RT shadows
- **Emissive**
  - Emissive sources get new life with path tracing
- **Invisible emissives: Artist placed geometry that is used to light the RT Probes**
  - Developed for the real time RT GI probe system, required some care to handle for path tracing
- **Non-shadowing lights**
  - Need special attention when shadowing bounces. We used the SSDO output to scale the radiance of these particular lights. This was a nice compromise that gave good results.
- **SHaRC**
  - Radiance caching for performance



# PT Emissive

Showcase of path traced environmental reflections on first person models

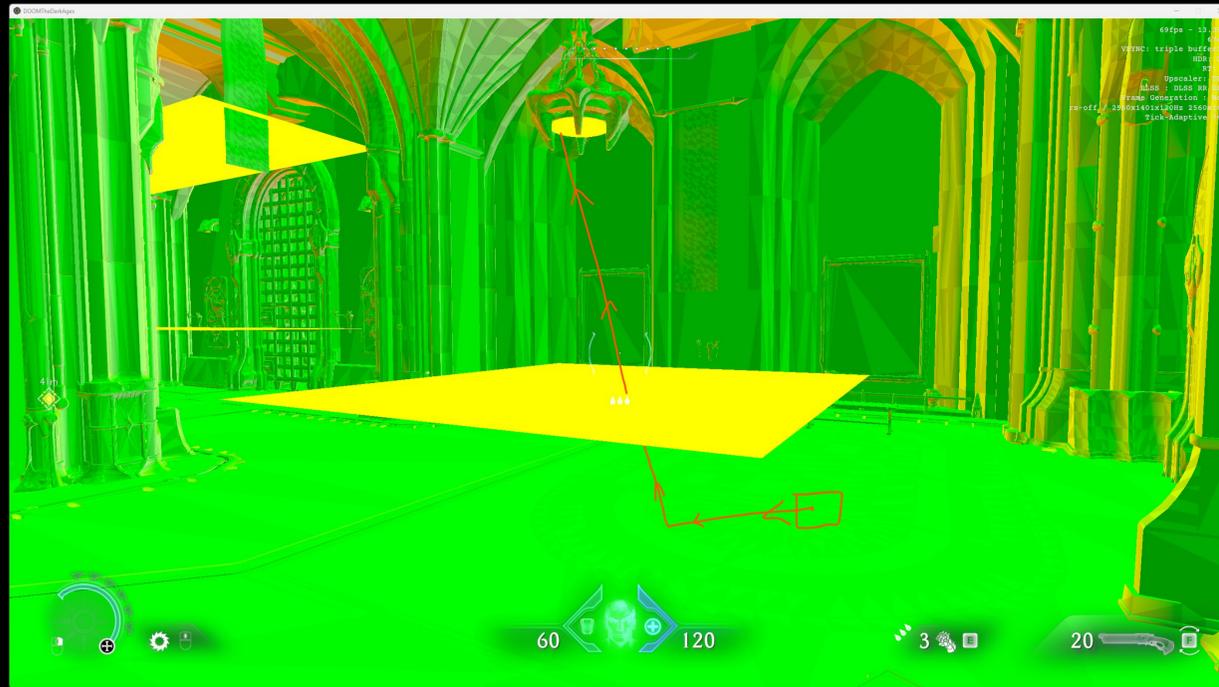


Standard Emissive

PT Emissive

# Invisible Emissives

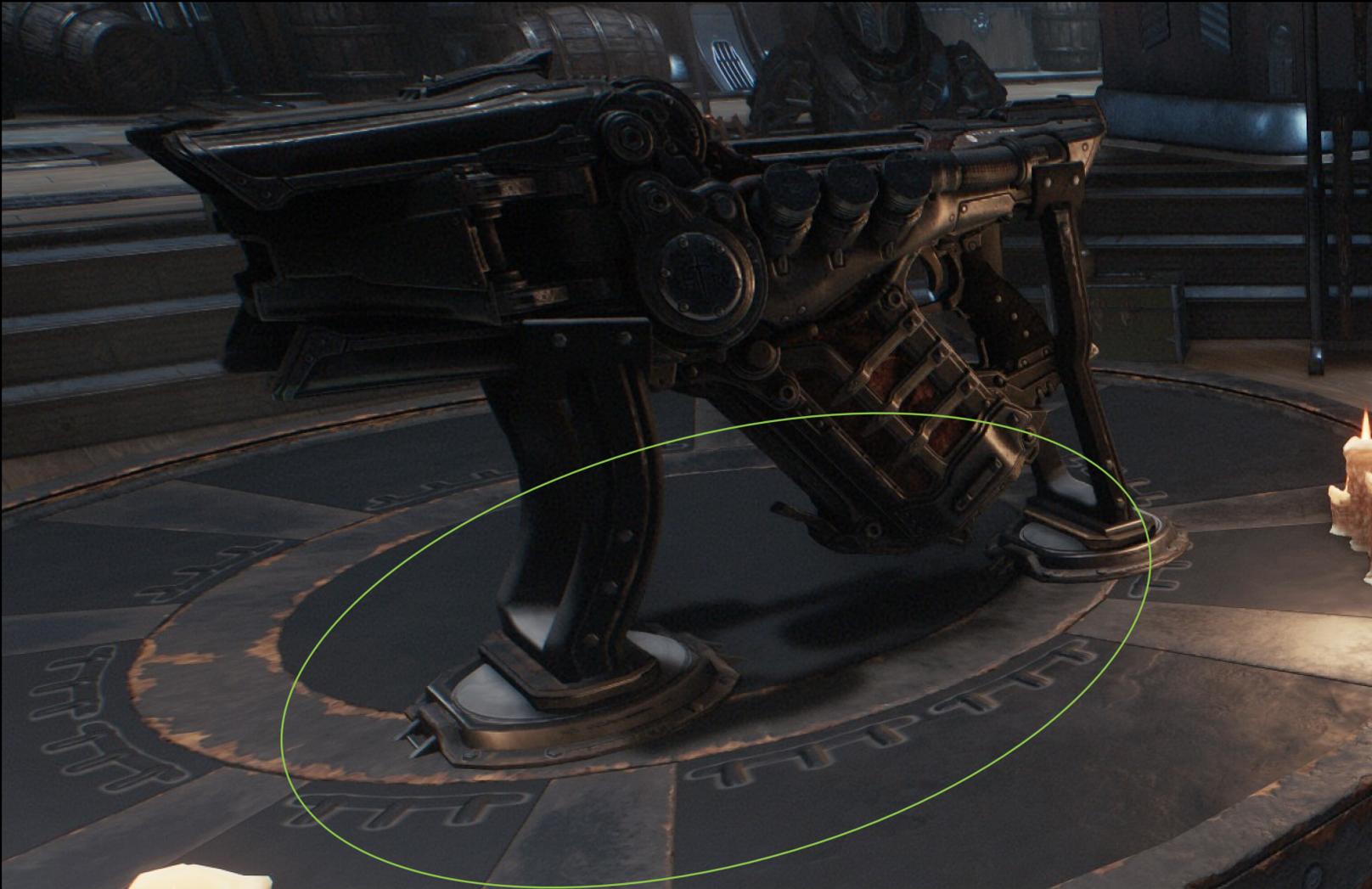
Tricky, but fundamental, to get just right with path tracing



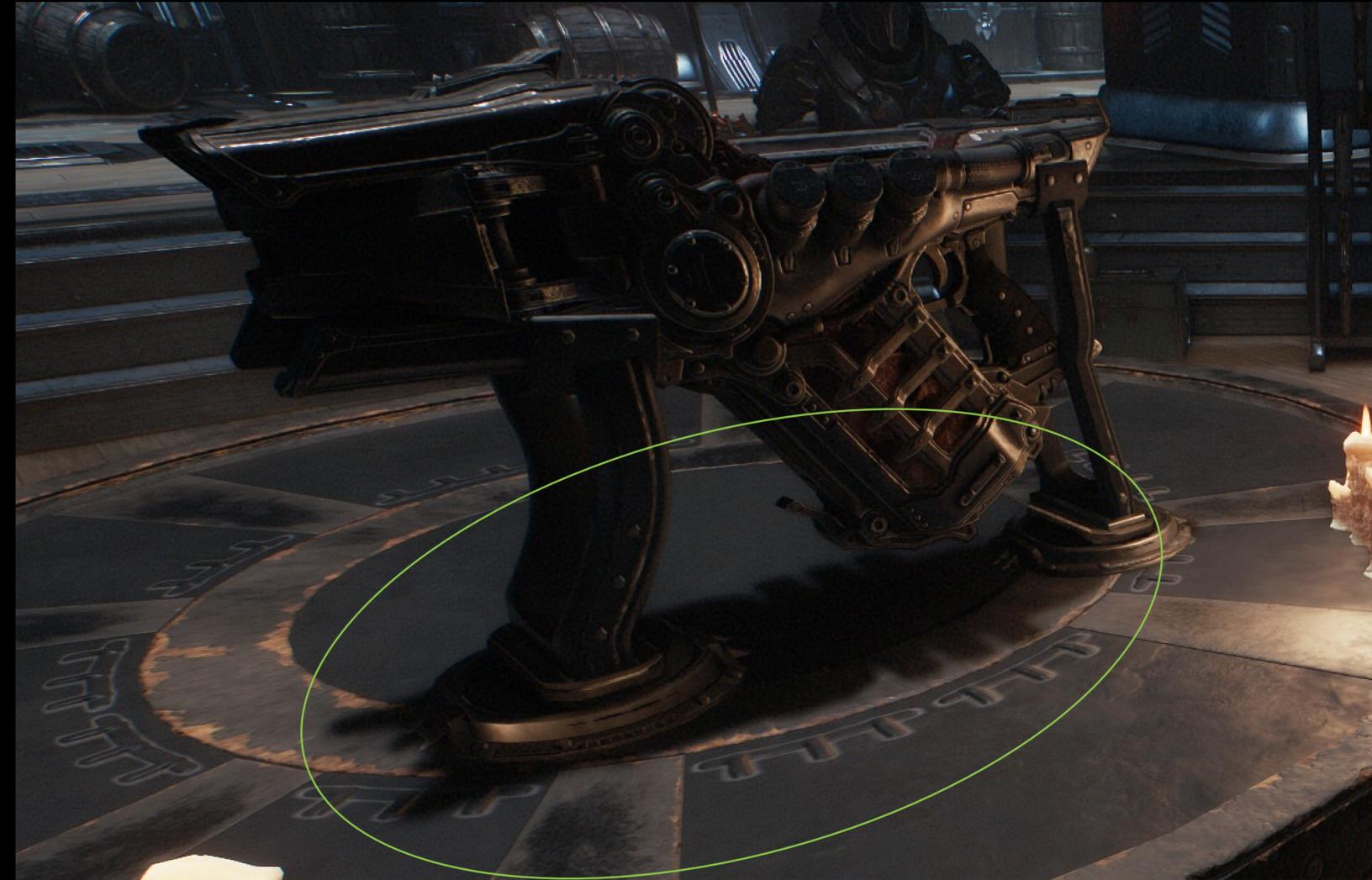
Various configurations of invisible emissives (green planes).

# Shadow Maps vs RT Shadows

Ray traced shadows when path tracing is enabled



Shadow Maps



RT Shadows

# Non-Shadowing lights

“reverse AO”



- **Non-shadowing lights require special attention in path tracing**

- Ignore them, and the scene will be too dark
- Use them, and the scene will be too bright

- **Solution:**

- Add a flag to non-shadowing lights to have option to shadow and light in PT, and only do it on the first bounce



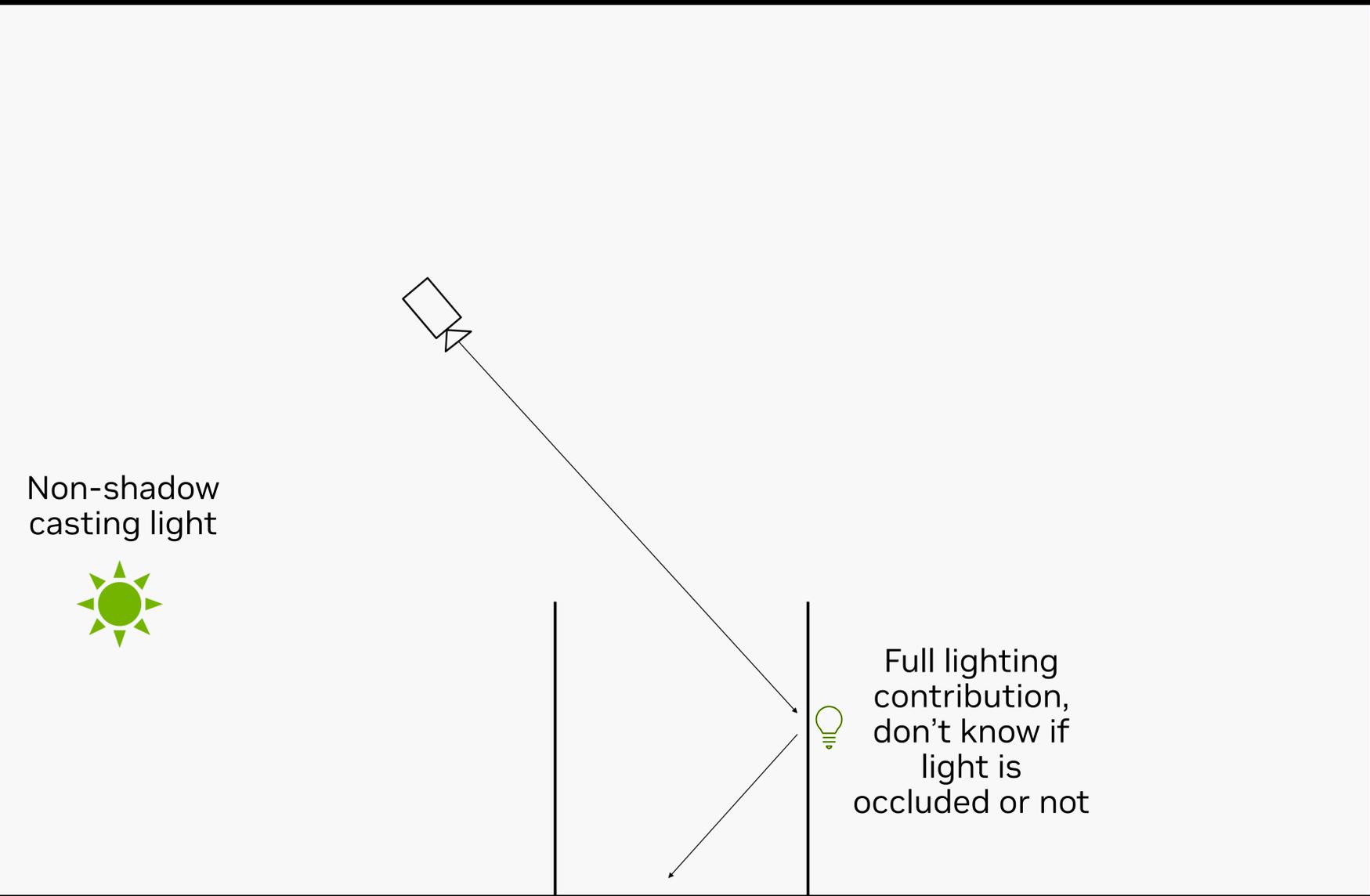
PT before fix



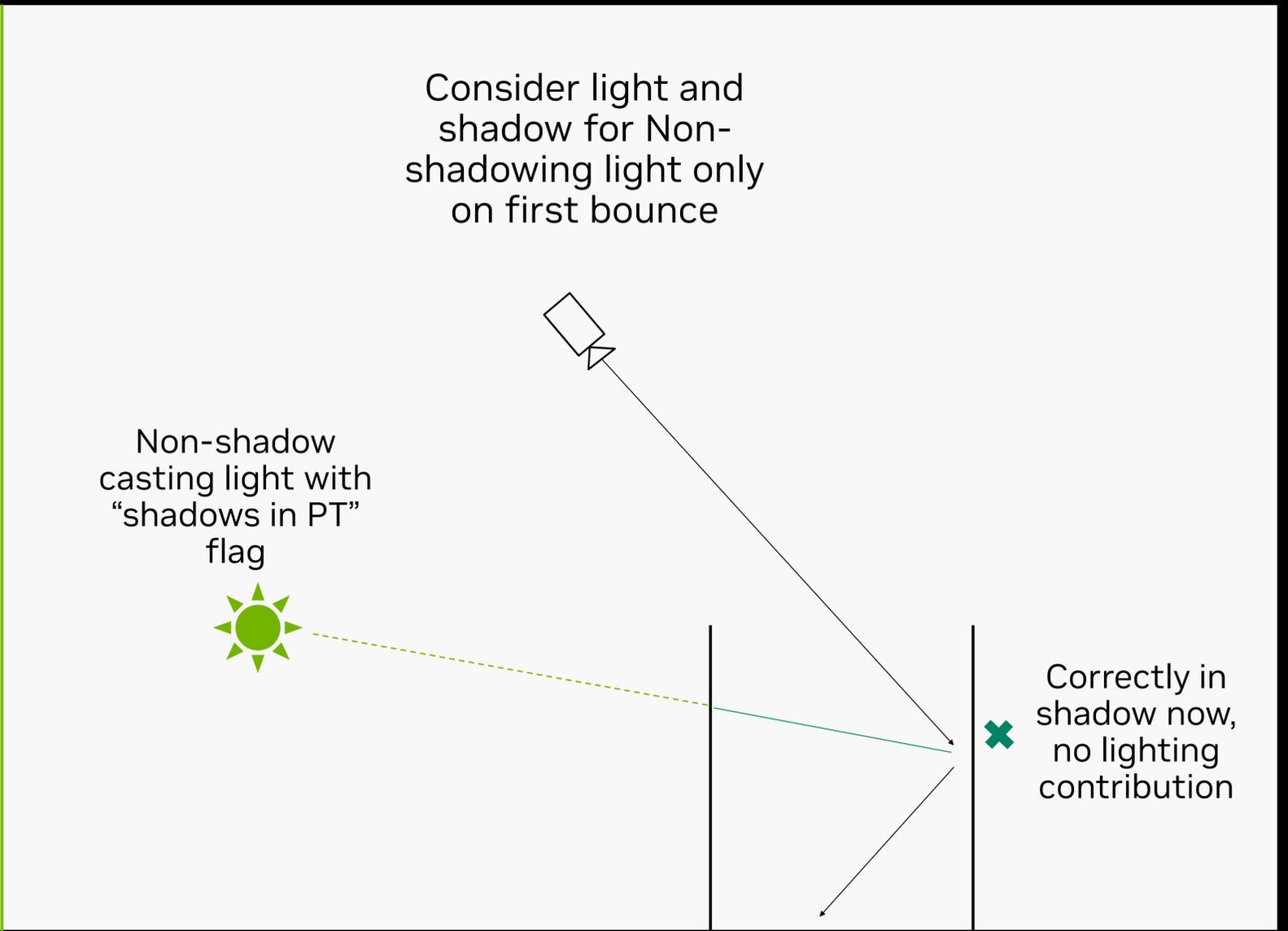
PT after fix

# Non-Shadowing lights

“reverse AO”



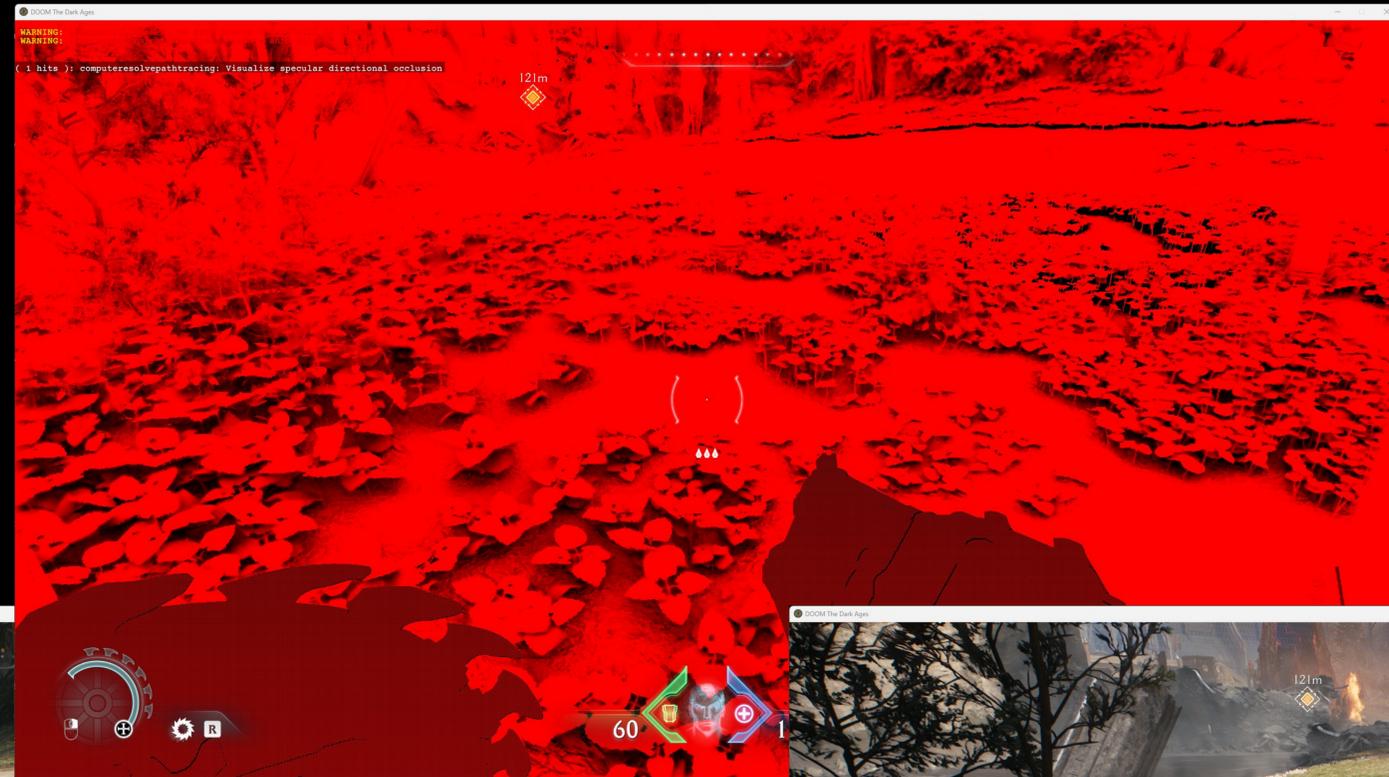
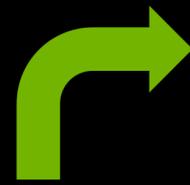
Full lighting contribution, don't know if light is occluded or not



✗ bounce 2+ doesn't consider non-shadowing lights for lighting contribution

# Devil in the Details

Shadowing geometry with no BLAS model – SSDO Mask Scaling Radiance  
[ Sousa2025 ]



# Devil in the Details

Shadowing geometry with no BLAS model – SSDO Mask Scaling Radiance  
[ Sousa2025 ]



Light leak

Hybrid shadows



Light leak

Hybrid shadows

# Particles in Path tracing

What's included



- **Particles**

- Weapon shots (player, enemy, dragon), fire, lightning, explosives

- **One TLAS for particles/blended, one for everything else**

- Keeps tracing clean and efficient for our use case
- Fast updates for the TLAS, don't want lag

- **Performance tracking**

- Having a second trace does have a perf cost
- Separation of TLAS is worth it

- **Can denoisers keep up with fast moving particles?**

- Spoiler: Yes!

```
Pathtrace() {
    Float3 outColor;
    RayDesc PTRay;
    PTRay.tMax = maxDistance;
    traceRay( PrimaryTlas, PTRay ); // with SER
    if ( PTRay.hitDistance < someCullingAmount ) {
        RayDesc particleRay;
        particleRay.tMax = PTRay.hitDistance;
        traceRay( BlendedTlas, particleRay ); // with SER
        outColor = Blend results of shading from PTRay and
        particleRay
    } else {
        outColor = only result from original ray
    }
}
```

# Particles in Path tracing

## What's NOT Included

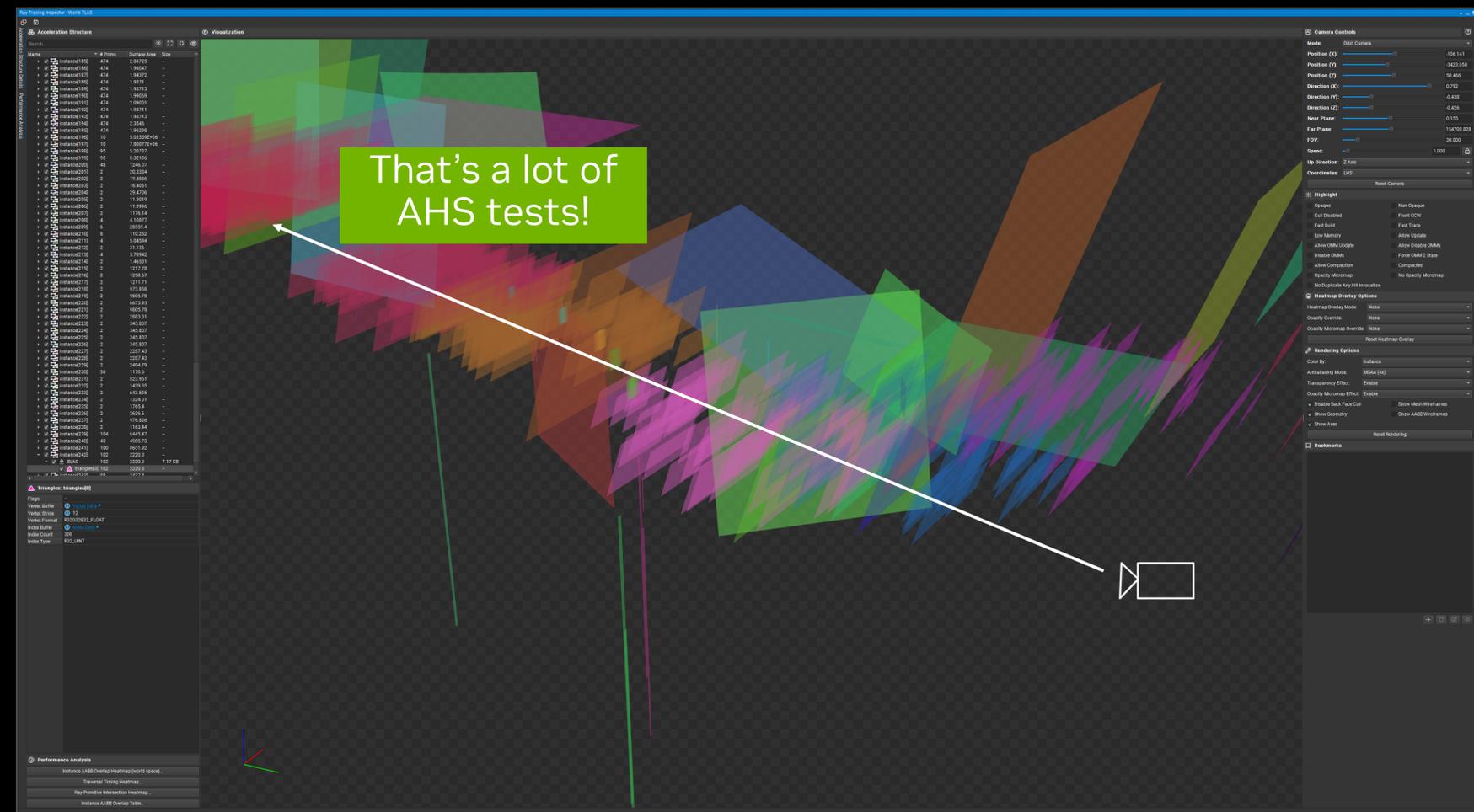


- **Exclude Fog and Smoke**

- One ray would intersect with hundreds of fog cards..
- High performance hit, low impact to final pixel
- Ran out of time, but potentially OMMs could have helped massively here? Warrants some future investigation.

- **Particles are distance culled**

- In general, all particles are distance culled for a simple performance improvement without sacrificing quality



# Particle Emissive Lighting

Fire



RT Particles Emissive



PT Particles Emissive

# Particle Emissive Lighting In Action



Base Game



PT

# Particles

## Dragon Emissive Shot



Standard Emissive

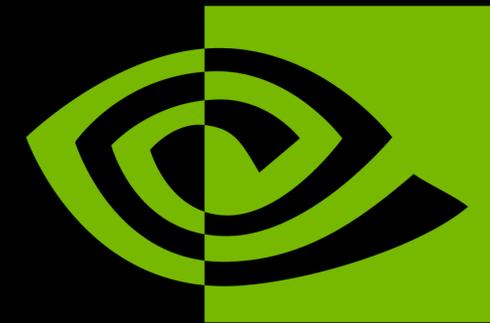


PT Emissive

# Shared vision, shared success

NVIDIA and id Software collaboration

- id Software and NVIDIA teams worked together to bring path tracing to DOOM: The Dark Ages
- Long standing partnership in driving innovation
  - DOOM (2016) - idTech6
    - GTX 1080 Launch (2 weeks later)
    - Vulkan Post Launch Patch (July 2016)
  - DOOM Eternal (2020) - idTech7
    - 3080 Launch Event – DOOM Eternal running 4K @ 120hz
    - Ray Tracing + DLSS 2 Patch (June 2021)
  - DOOM: The Dark Ages – idTech8
    - RTX 50 Series Launch
    - Path Tracing
    - DLSS4

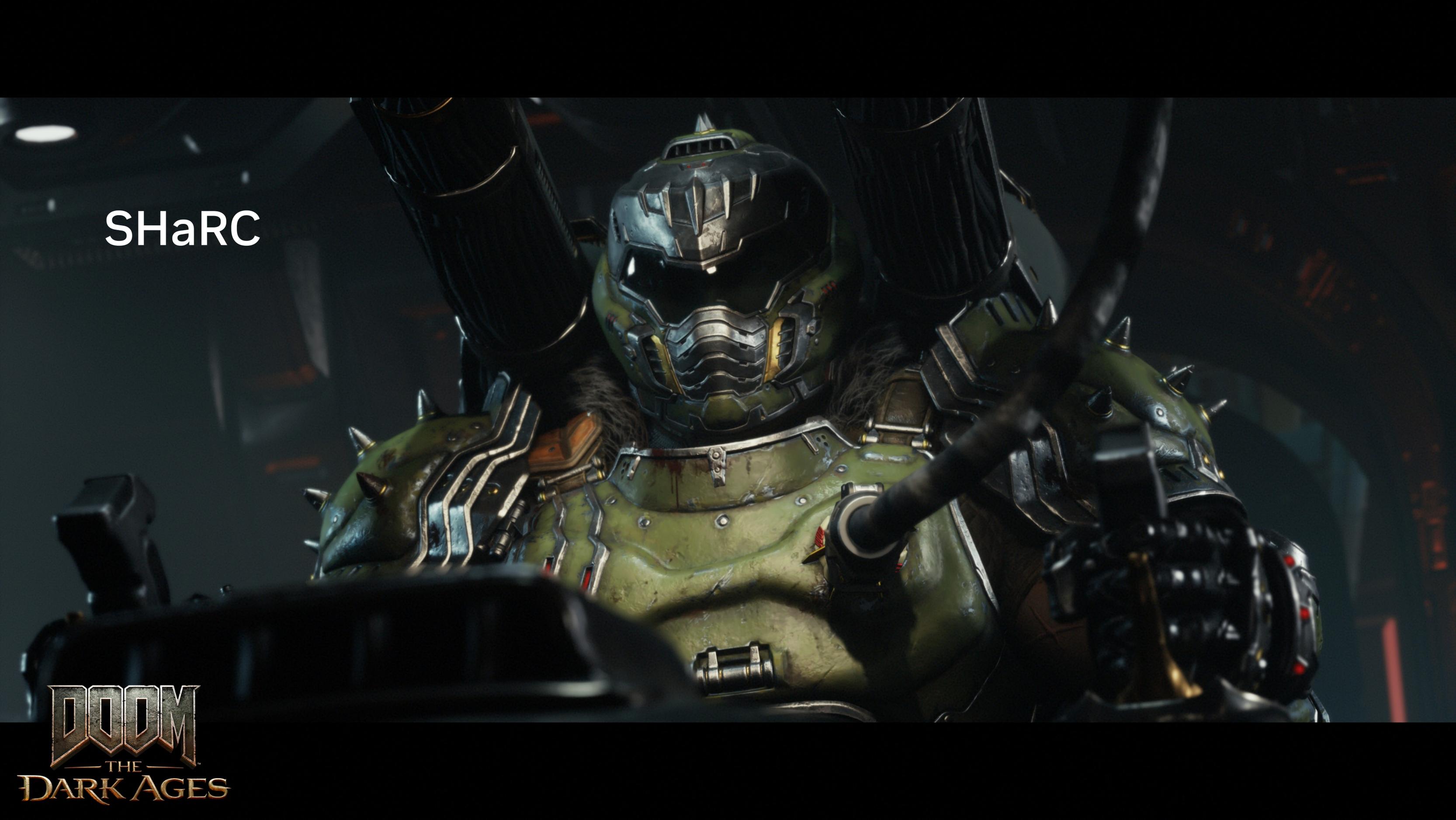


**NVIDIA**



SHaRC

**DOOM**  
— THE —  
**DARK AGES**



# Radiance Caching

## SHaRC

- **Performance technique**

- SHaRC- Spatial Hash Radiance Cache
- Use existing path tracing loop to update cache with a sparse grid, then potentially query that cache to exit path early
- 4 ray bounces for update, 2 for query (with valid cache)

- **SHaRC is an elegant software solution that is hardware agnostic**

- 3 headers files, see integration guide QR code (and GPU Zen 3 Cyberpunk article) for further details



Denoising



**DOOM**  
— THE —  
DARK AGES

# Denoising

## DLSS-RR and NRD



- **Ray Reconstruction in DLSS4 (DLSS-RR)**

- DLSS-RR uses a new transformer model that improves upon image quality over the previously used CNN model
  - Improved temporal stability, less ghosting, higher detail of objects in motion
  - Preserves details in specular areas and thin objects
- Transformer model uses 2x as many inputs as the CNN, capturing more details per pixel

- **NVIDIA Real-time Denoiser (NRD)**

- Has 3 flavors of denoisers, each taking different inputs
  - REBLUR - recurrent blur based denoiser
  - RELAX - A-trous based denoiser (designed for RTXDI)
  - SIGMA - shadow-only denoiser
- Not AI driven, loses out to DLSS-RR transformer model in terms of image quality



(a) CNN model vs transformer model



(b) NRD vs DLSS-RR

# Denoising



NRD vs DLSS-RR



NRD



DLSS-RR

# Denoising

NRD vs DLSS-RR: Fast Moving Particles



NRD



DLSS-RR

# Opacity Micro-Maps

**DOOM**  
— THE —  
**DARK AGES**



# What are OMMs?

## Alpha Testing in a Material World

- **Problem**

- How do we get from figure (a) to figure (b) efficiently when ray tracing?

- **Opacity Micro Maps**

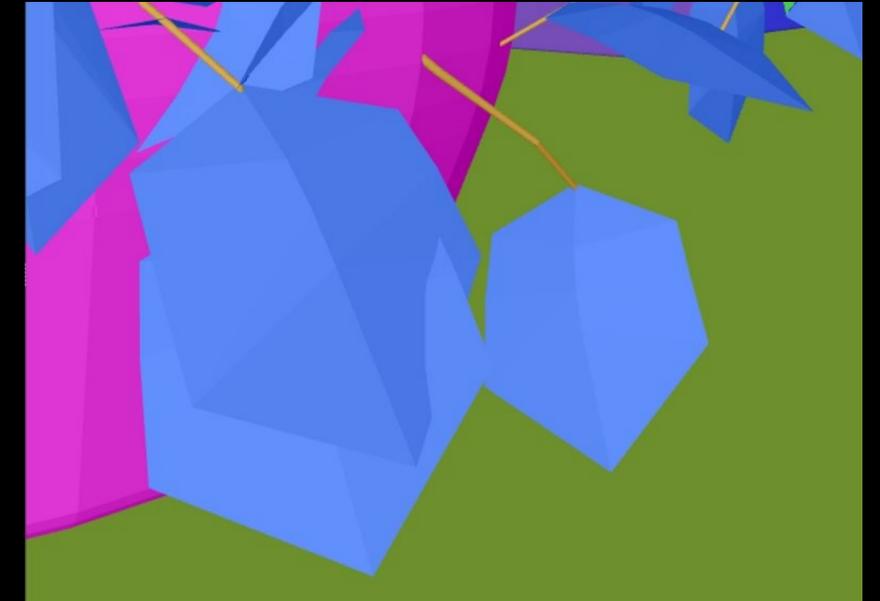
- Acceleration structure used by ray tracing unit to accelerate alpha-testing
- Subdivides a triangle's alpha-tested texture regions into a micro-triangle bitmask grid
- Allows ray traversal to quickly skip fully opaque or transparent micro-triangle
- Only evaluate rays in partially covered areas (aka unknowns)

- **2-state OMMs**

- Uses 1 bit to store if a micro-triangle is opaque or transparent
- No shader invocations

- **4-state OMMs**

- Uses 2 bit to store if a micro-triangle is opaque, transparent, unknown opaque, or unknown transparent
- Unknowns requires any-hit shader invocation of resolve transparency state



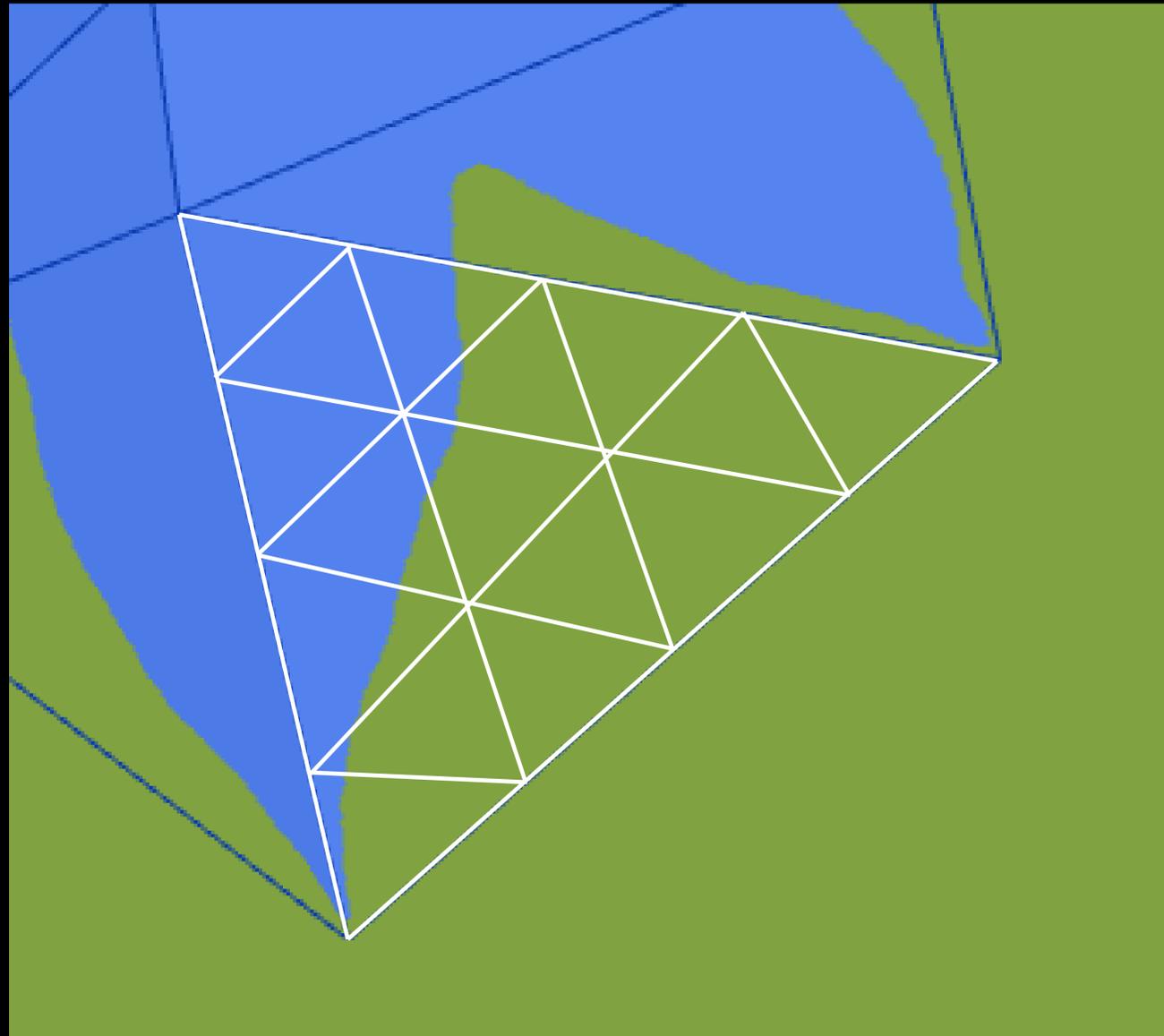
(a) Foliage geometry



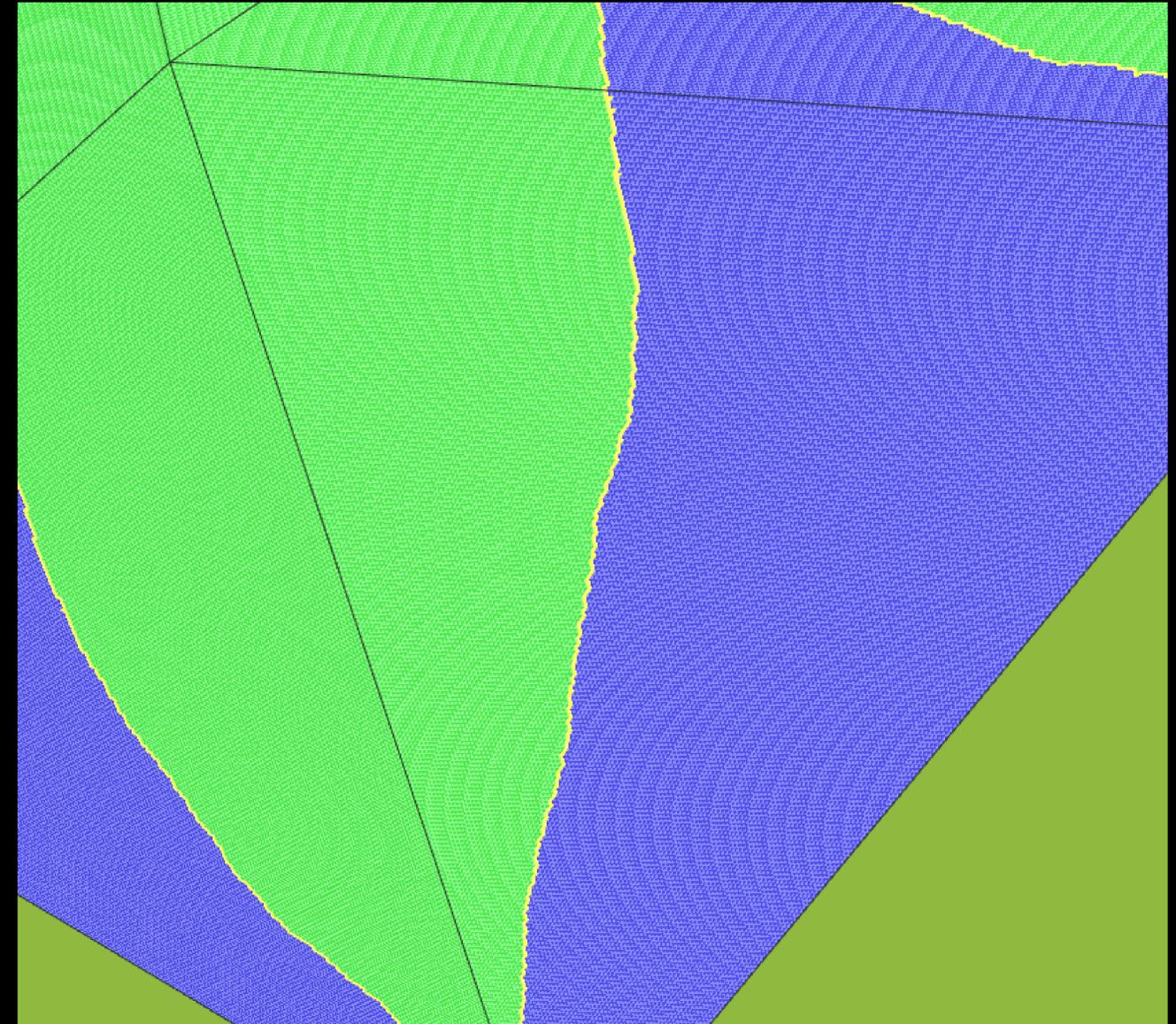
(b) Alpha-tested result

# What are OMMs?

## Subdivided Micro-Triangles



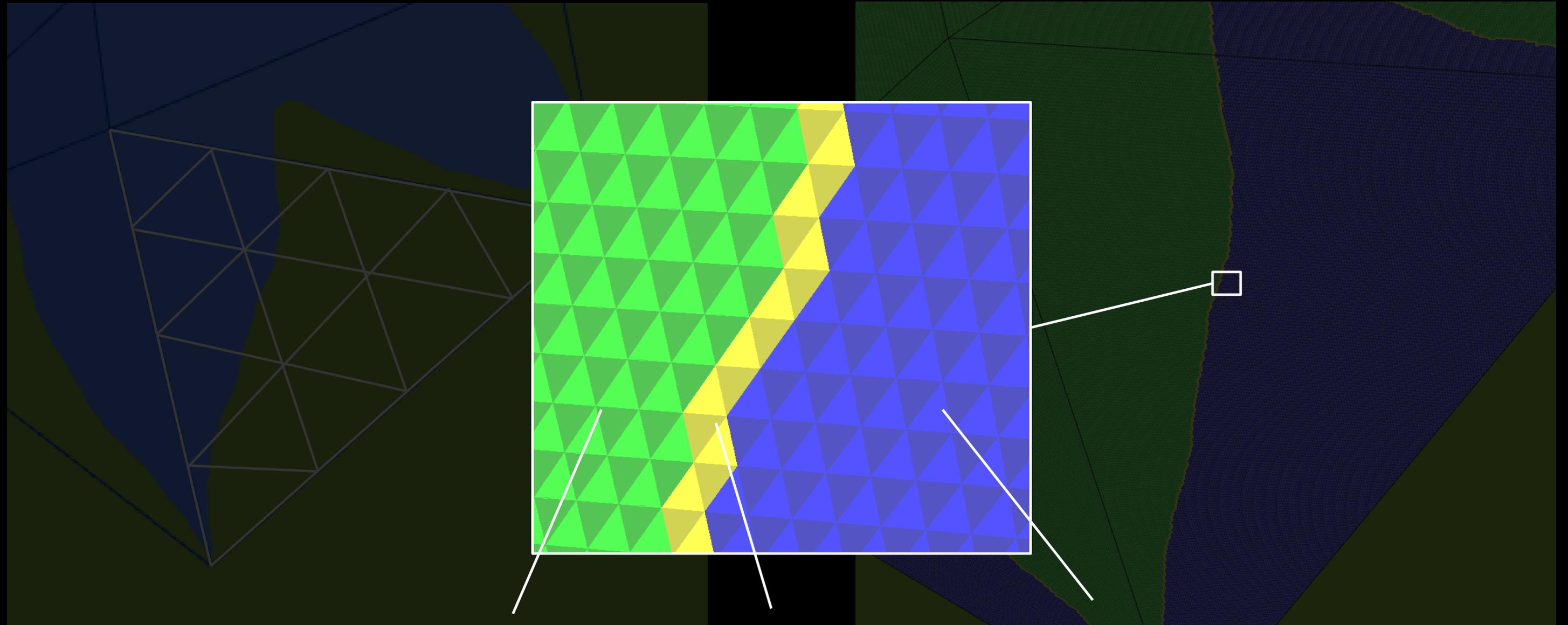
Low resolution subdivision



12 subdivision levels (20736 micro triangles)

# What are OMMs?

4-State Opacity Micro-Maps



Opaque

Partially Transparent

Transparent

Example subdivision

12 subdivision levels (20736 micro triangles)

# Why OMMs Matter In Path Tracing

## Alpha Testing in a Material World



- **Reduces wasted calls to any-hit invocation and texture reads**

- Without OMMs alpha-testing requires a shader invocation per hit to evaluate the transparency – see Figure (a)
- With OMMs alpha-testing only requires shader invocation when a micro-triangle is not fully opaque or fully transparent – see Figure (b)

- **Lowers divergence in data access**

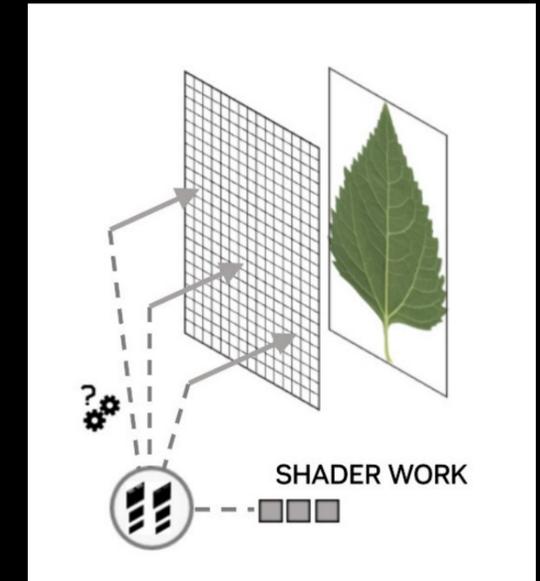
- Regions of opacity (alpha) maps are usually the same
  - Enables en-masse hit or miss evaluations of rays that intersect the same micro-triangles without potentially reading from disparate texture memory regions

- **Memory-performance tradeoff control**

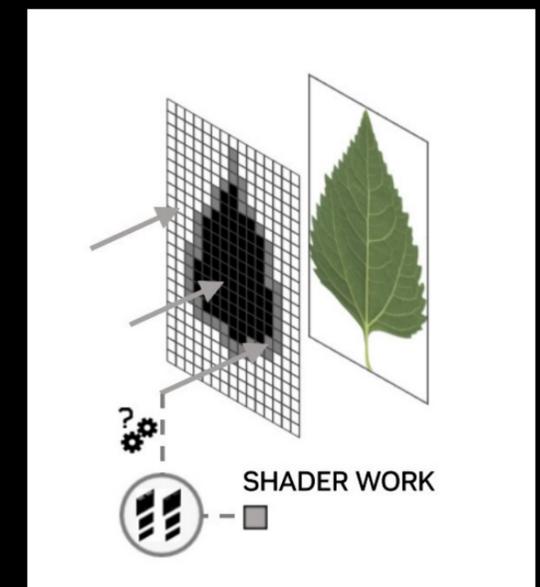
- OMM subdivisions are scalable allowing developers to adjust resolution to fit memory budgets or to boost performance
  - Helpful on GPUs with less VRAM where every byte counts

- **GPU budget that was once wasted can be spent elsewhere in path tracer**

- Increase ray counts in areas that can improve overall image quality
- More head room for denoiser to user higher settings
- Or spent on nothing at all: clawed back GPU savings helps maintain steady FPS



(a) Without OMMs  
3 ray hits / 3 shader invocations



(b) With OMMs  
3 ray hits / 1 shader invocation

# OMMs in DOOM: The Dark Ages

- **DOOM: The Dark Ages has very diverse environments**
  - Lots of areas with foliage on trees, scattered leaves on the ground, and dangling vines
  - Not as much as Indiana Jones and the Great Circle but still a lot
- **OMMs clawed back about 2~3 ms (depending on GPU)**
  - More noticeable on slower GPUs where performance is in much more of a crunch
- **Links below for in depth information about OMMs**



OMM SDK



IJ&GC OMM



# Performance and VRAM Strategies

**DOOM**  
— THE —  
**DARK AGES**



# DOOM THE DARK AGES

## PATH TRACING HARDWARE REQUIREMENTS

	MINIMUM	RECOMMENDED	ULTRA 4K
<b>PLAYER EXPERIENCE</b>	1080P/60 FPS/MEDIUM VIDEO SETTINGS DLSS SUPER RESOLUTION PRESET: BALANCED	1440P/60 FPS/HIGH VIDEO SETTINGS DLSS SUPER RESOLUTION PRESET: BALANCED	2160P/60 FPS/ULTRA VIDEO SETTINGS DLSS SUPER RESOLUTION PRESET: PERFORMANCE
<b>PATH TRACE SETTINGS</b>	<ul style="list-style-type: none"> <li>• DENOISER: DLSS RAY RECONSTRUCTION</li> <li>• RAY TRACED SHADOWS: OFF</li> <li>• RAY TRACED TRANSPARENCY: ON</li> <li>• RAY WATER REFLECTIONS: ON</li> <li>• OPTIONAL : DLSS FRAME GENERATION 2X</li> </ul>	<ul style="list-style-type: none"> <li>• DENOISER: DLSS RAY RECONSTRUCTION</li> <li>• RAY TRACED SHADOWS: ALL LIGHTS</li> <li>• RAY TRACED TRANSPARENCY: ON</li> <li>• RAY WATER REFLECTIONS: ON</li> <li>• OPTIONAL : DLSS FRAME GENERATION 2X, 3X, 4X</li> </ul>	<ul style="list-style-type: none"> <li>- DENOISER: DLSS RAY RECONSTRUCTION</li> <li>- RAY TRACED SHADOWS: ALL LIGHTS</li> <li>- RAY TRACED TRANSPARENCY: ON</li> <li>- RAY WATER REFLECTIONS: ON</li> <li>- OPTIONAL : DLSS FRAME GENERATION 2X, 3X, 4X</li> </ul>
<b>CPU</b>	AMD ZEN 3 OR INTEL 12TH GENERATION @3.2GHZ WITH 8 CORES / 16 THREADS OR BETTER (EXAMPLES: AMD RYZEN 7 5700X OR BETTER, INTEL CORE I7 12700K OR BETTER)	AMD ZEN 3 OR INTEL 12TH GENERATION @3.2GHZ WITH 8 CORES / 16 THREADS OR BETTER (EXAMPLES: AMD RYZEN 7 5700X OR BETTER, INTEL CORE I7 12700K OR BETTER)	AMD ZEN 3 OR INTEL 12TH GENERATION @3.2GHZ WITH 8 CORES / 16 THREADS OR BETTER (EXAMPLES: AMD RYZEN 7 5700X OR BETTER, INTEL CORE I7 12700K OR BETTER)
<b>GPU</b>	NVIDIA RTX 4070 WITH 12GB DEDICATED VRAM OR BETTER	NVIDIA RTX 5080 WITH 16GB DEDICATED VRAM OR BETTER	NVIDIA RTX 5090 WITH 24GB DEDICATED VRAM OR BETTER
<b>SYSTEM RAM</b>	32GB	32GB	32GB
<b>STORAGE CAPACITY</b>	512GB OR HIGHER NVME SSD (100GB AVAILABLE)	512GB OR HIGHER NVME SSD (100GB AVAILABLE)	512GB OR HIGHER NVME SSD (100GB AVAILABLE)

# Performance and VRAM Challenges

60+ FPS or Bust

- **Path tracing in DOOM: The Dark Ages pushes the performance envelope on modern GPUs**
  - Players experience is 60+ FPS @ 1080P (Min), 1440P (Rec), and 2160P (Ultra)
  - Each hardware requirement level required finding path tracing settings and DLSS-RR preset that provides the best experience
- **Path tracing increases trace depth and adds divergence**
  - More rays on top of the ray tracing workloads that the game was already doing
  - Going beyond primary bounce can lead to execution divergence and memory access divergence
- **Path tracing increases VRAM usage**
  - Component used by path tracer that incurred notable VRAM increase
    - Denoiser, SHaRC, and OMMs
  - Increased VRAM utilization makes game susceptible to OS eviction
  - Eviction can mean a possible dramatic drop in performance and the likelihood of never recovering in the session
- **Goals**
  - Leverage GPU features like SER and OMMs that can yield improved performance in path tracing scenarios
  - Ensure that VRAM usage is kept out of the OS eviction danger zone

# SER

## Shader Execution Reordering



- **A heavy lifter in DOOM: The Dark Ages path tracing performance is SER**

- Made hitting 60+ FPS possible when path tracing

- **What is Shader Execution Reordering?**

- GPU performance optimization feature that improves ray and memory coherency for ray tracing workloads
- On the fly reorders thread execution and data access to mitigate divergence
- Available on NVIDIA GPUs since the RTX 40 series
- Vulkan requires `VK_NV_ray_tracing_invocation_reorder` / D3D12 requires DXR 1.2

- **SER shader integration in GLSL is straight forward**

```
• #if defined( ENABLE_SER )
  hitObjectNV hitObject;
  traceRayHitObject( hitObject, rayFlags, instanceMask , ray, rayPayload, tlas );
  reorderThreadNV( hitObject, bounceNum == ( bounceId + 1 ) ? 1 : 0 , 1 );
  hitObjectExecuteShaderNV( hitObject, _hitPayloadIndex( rayPayload ) );
#else
  traceRayEXT( rayFlags, instanceMask, ray, rayPayload, tlas );
#endif
```

- **For real-time path tracing, SER can be the make or break performance feature**

- Depending on the workload, SER can deliver up to 20-40% performance improvements

- **Performance improvements are observable through Nsight Graphics**

- Top QR code is easy to follow introduction to SER
- Bottom QR code link deep dives into SER optimization for Indiana Jones and the Great Circle



SER INTRO



IJ&GC SER

# VRAM Utilization

- **Components of the path tracer that increased VRAM utilization**
  - DLSS-RR (Denoiser)
  - SHaRC
  - OMMs
- **DLSS-RR VRAM usage**
  - Additional images for the following
    - Albedo, normal+roughness, SSS guide, specular motion vectors, transparency guide, and depth of field guide
- **SHaRC VRAM usage**
  - Hash and voxel data combined uses 320 bits (40 bytes)
  - $2^{22}$  entries are used amounting to 160MB
- **OMMs VRAM usage**
  - Varies depending on the number of transparent objects
  - Total VRAM usage can be capped by controlling pools
- **DLSS-SR VRAM usage**
  - Changing DLSS-SR preset to higher quality increases VRAM usage
  - Typically, not a big deal but combined with changes in other settings can push VRAM usage closer to the red line

# Balancing Performance vs VRAM Tradeoffs

- **Approaching the VRAM red line**

- Minimum VRAM requirement for pathtracing in DOOM: The Dark Ages is 12GB
- About 9.6GB (roughly 80%) is available to the game
  - Empirical observation on when resource started to get evicted by OS
- Had to ensure total VRAM usage for largest map didn't put the game's resources in the danger zone for eviction by the OS
  - Evicted resources can easily crater the frame rate

- **Dialing in VRAM utilization**

- Finding the right DLSS-SR preset for each hardware requirement level
- Lowering SHaRC voxel resolution to reduce VRAM usage
- Lowering OMMs resolution to reduce VRAM usage
- Old school blood, sweat, and tears optimization efforts by team members to reduce overall VRAM usage

- **Performance Tweaking**

- OMMs can be forced to two state to eliminate all shader invocations for unknown states

DLSS4

**DOOM**  
— THE —  
**DARK AGES**



# DLSS4

## Giving Players More Options

- **Enable gamers to target their monitor's native resolution and refresh rates**
  - High resolution + high refresh rate monitors are common nowadays
  - High-end gaming monitors can display 4k @ 240hz and 1080P @ 480hz
  - Gamers want to know their investment in hardware was worth it
- **Challenging to sustain extreme frame rates**
  - GPUs can struggle to keep up with the pixel load for 4K @ 240hz or 1080P @ 480hz
  - CPUs can bottleneck on command generation and scheduling
  - Latency and frame pacing issues become more noticeable at sub 4ms
- **DLSS4 helps titles tackle these challenges**
  - Frame Gen boosts FPS by generating a frame between rendered ones on RTX 40 series and RTX 50 series GPUs
  - Multi Frame Gen pushes FPS envelope further by generating up to 3 additional frames on RTX 50 series GPUs
  - Reflex reduces latency by pacing work submissions from the CPU to minimize GPU-bound scenarios

# Frame Gen (FG)

DLSS4

- **Inputs for frame generation**

- Color buffer (no HUD or UI), motion vector buffer, and linear depth buffer for each frame
- Separate HUD/UI buffer for compositing after frame generation

- **Per-frame setup**

- Send constants such as camera matrices, jitter offsets, and frame index through API
- Must keep all tagged resources valid and unchanged until presentation completes

- **Frame graph placement**

- Insert Frame Gen pass after post-processing chain but before HUD/UI compositing
- Update the present loop to alternate between rendered and generated frames

- **Integration Considerations**

- RTX 40 series and RTX 50 series GPUs only
- Requires Reflex to control latency and prevent synchronization issues

# Multi-Frame Gen (MFG)

DLSS4

- **Inputs for frame generation (same as FG)**
  - Color buffer (no HUD or UI), motion vector buffer, and linear depth buffer for each frame
  - Separate HUD/UI buffer for compositing after frame generation
- **Per-frame setup**
  - Provide correct frame index and timing data to API
    - Correctness is critical since one frame now drives multiple presents
  - Must keep all tagged resources valid and unchanged until presentation completes for all frames
- **Frame graph placement**
  - Insert Frame Gen pass after post-processing chain but before HUD/UI compositing
  - Present loop must schedule 1 rendered frame followed by up to 3 generated frames in proper sequence
- **Integration Considerations**
  - RTX 50 series GPUs only
  - Requires Reflex to control latency and prevent synchronization issues

Closing

**DOOM**  
— THE —  
DARK AGES



# Thank You!

- **id Software**

- Allen Bogue, Bogdan Coroi, Carson Fee, Darin McNeil, Dominik Lazarek, Ian Malerich, Jean Geffroy, Johan Donderwinkel, John Roberts, Mel-Frederic Fidorra, Oliver Fallows, Philip Hammer, Regan Carver, Seth Hawkins, Thorsten Lange, Tiago Sousa, Yixin Wang
- Our amazing content teams for constantly pushing the limits.
- Special Thanks : Marty Stratton, Hugo Martin, and Laffy Taylor

- **NVIDIA**

- Mike Songy, Ryan Prescott, Eric Reichley, Cory Spencer, Hai Nguyen, Pawel Kozlowski, Peter Morley, Ray Batts, Rajeev Penmatsa, Jussi Rasanen, Ivan Povarov, Louis Bavoil, Sean Pelletier, Chris Lentini, Jiho Choi

- **MachineGames**

- Jim Kjellin, Magnus Auvinen, Markus Alind, Sergei Kulikov

**Most importantly: Thank you to all our fans!!!**

# References

1. [ Sousa2025 ] Fast as Hell: idTech8 Global Illumination
2. [ Vahl2025 ] Pushing 60hz. Shipping Indiana Jones and The Great Circle
3. [ Ouyang2021 ] ReSTIR GI: Path Resampling For Real-Time Path Tracing
4. [ Halen2021 ] Global Illumination Based on Surfels
5. [ Wright2021 ] Radiance Caching for Real-Time Global Illumination
6. [ Geffroy2020 ] Rendering The Hellscape of DOOM Eternal
7. [ Gautron2020 ] Real-Time Ray-Traced Ambient Occlusion of Complex Scenes using Spatial Hashing. <https://youtu.be/oza36AqcLW8>
8. [ Majercik2019 ] Dynamic Diffuse Global Illumination with Ray-Traced Irradiance Fields
9. [ Hobson2019 ] The Indirect Lighting of God of War
10. [ O'Donnell2018 ] Precomputed Global Illumination in Frostbite
11. [ Drobot2017 ] Improved Culling For Tiled And Clustered Rendering
12. [ Sousa2016 ] The Devil Is In The Details, idTech 666
13. [ Wronski2014] Assassin's Creed 4: Road To Next Gen Graphics
14. [ Lazarov2013 ] Getting More Physical in Call of Duty Black Ops 2
15. [ Olson2012 ] Clustered Deferred and Forward Shading

# References

16. [ Czuba2010 ] Box Projected Cube-Map Environment Mapping <http://devlog.behc.pl/> (<http://web.archive.org>)
17. [ Mittring2009 ] A bit more deferred – CryEngine 3
18. [ Ritschel2009 ] Approximating Dynamic Global Illumination In Image Space
19. [ Dachsbacher2008 ] Octahedron Environment Maps
20. [ Sloan2007 ] Image-Based Proxy Accumulation for Real-Time Soft Global Illumination
21. [ Tatarchuk2005 ] Parallax Occlusion Mapping: Self-shadowing, Perspective Correct Bump Mapping using Reverse Height Map Tracing
22. [ Landis2002 ] Production-Ready Global Illumination
23. [ Ramamoorthi2001 ] An efficient representation for irradiance environment maps
24. [ Greger1998 ] The Irradiance Volume
25. [ Ward1988 ] A Raytracing Solution For Diffuse Inter-reflections
26. [ SBT ] The Shader Binding Table Demystified, Will Usher, Ray Tracing Gems 2
27. [ Oodle Kraken ] <https://www.radgametools.com/oodlekraken.htm>