NVIDIA RESEARCH TALK:

THE MAGIC BEHIND GAMEWORKS' HYBRID FRUSTUM TRACED SHADOWS

Chris Wyman

July 28, 2016

MARCH 2016: 1ST RAY-TRACED SHADOWS IN GAMES

Now available as GameWorks module; shipped in ***Tom Clancy's The Division***

Left: **Hybrid Frustum Traced Shadows (HFTS)**

Right: **Percentage Closer Soft Shadows (PCSS)**

WHO?

Joint work:

- Chris Wyman, NVIDIA Research
- Jon Story, NVIDIA DevTech
- UbiSoft's Massive, developers of The Division

From *Tom Clancy's The Division*

WHO?

Joint work:

- Chris Wyman, NVIDIA Research
- Jon Story, NVIDIA DevTech
- UbiSoft's Massive, developers of The Division

An NVIDIA success story of transitioning research to product

From *Tom Clancy's The Division*

WHO?

Joint work:

- Chris Wyman, NVIDIA Research
- Jon Story, NVIDIA DevTech
- UbiSoft's Massive, developers of The Division

An NVIDIA success story of transitioning research to product

From *Tom Clancy's The Division*

May not know:

- NVIDIA has research division of 100+ researchers
- Covering graphics, VR, machine learning, AI, compilers, vision, circuits, etc.

WHO?

Joint work:

- Chris Wyman, NVIDIA Research (2+ years effort)
- Jon Story, NVIDIA DevTech (6+ months effort)

NVIDIA enables researchers and engineers to spend time addressing important graphics problems

- UbiSoft's Massive, developers of The Division

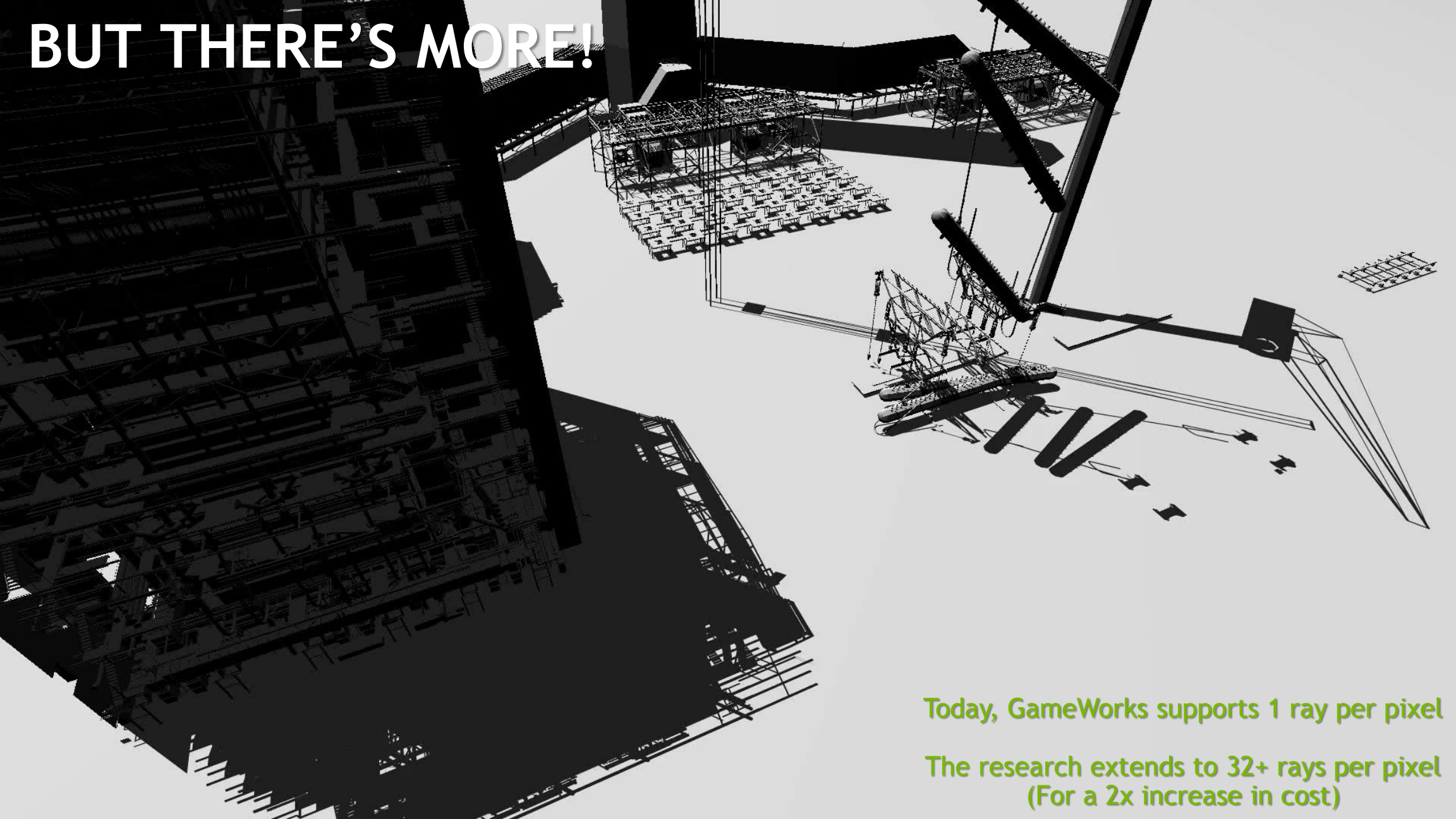
From *Tom Clancy's The Division*

An NVIDIA success story of transitioning research to product

May not know:

- NVIDIA has research division of 100+ researchers
- Covering graphics, VR, machine learning, AI, compilers, vision, circuits, etc.

BUT THERE'S MORE!



Today, GameWorks supports 1 ray per pixel

The research extends to 32+ rays per pixel
(For a 2x increase in cost)

STORY

Today: talk about the road to productization and research tech transfer

STORY

Today: talk about the road to productization and research tech transfer

Up to 5 billion shadow rays/sec in fully dynamic scenes, incl. data structure build

- On GeForce GTX Titan X (2015)
- Specialized algorithm for ray traced hard shadows
- Fits in raster pipeline; no extra ray tracing library

STORY

Today: talk about the road to productization and research tech transfer

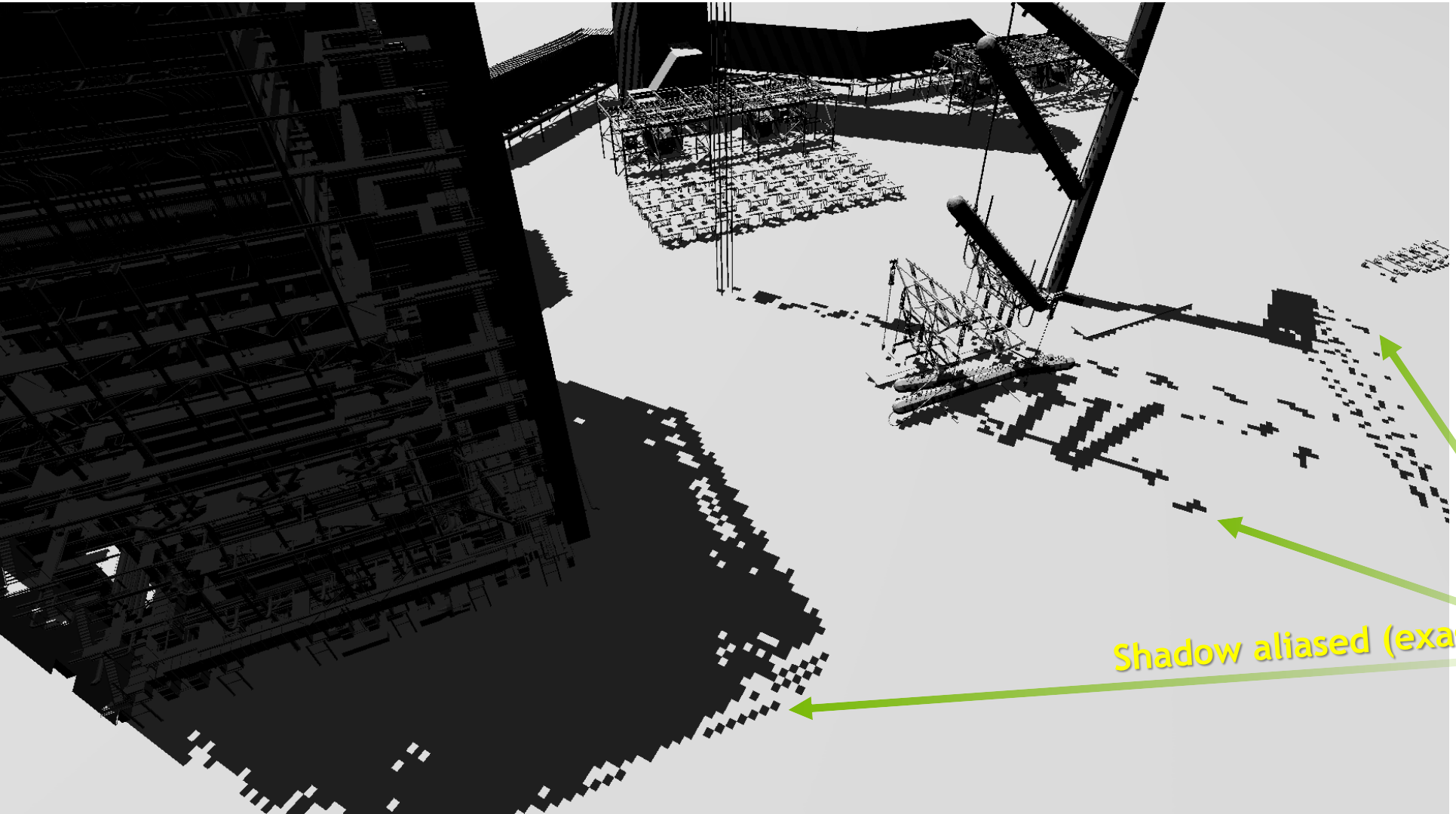
Up to 5 billion shadow rays/sec in fully dynamic scenes, incl. data structure build

- On GeForce GTX Titan X (2015)
- Specialized algorithm for ray traced hard shadows
- Fits in raster pipeline; no extra ray tracing library

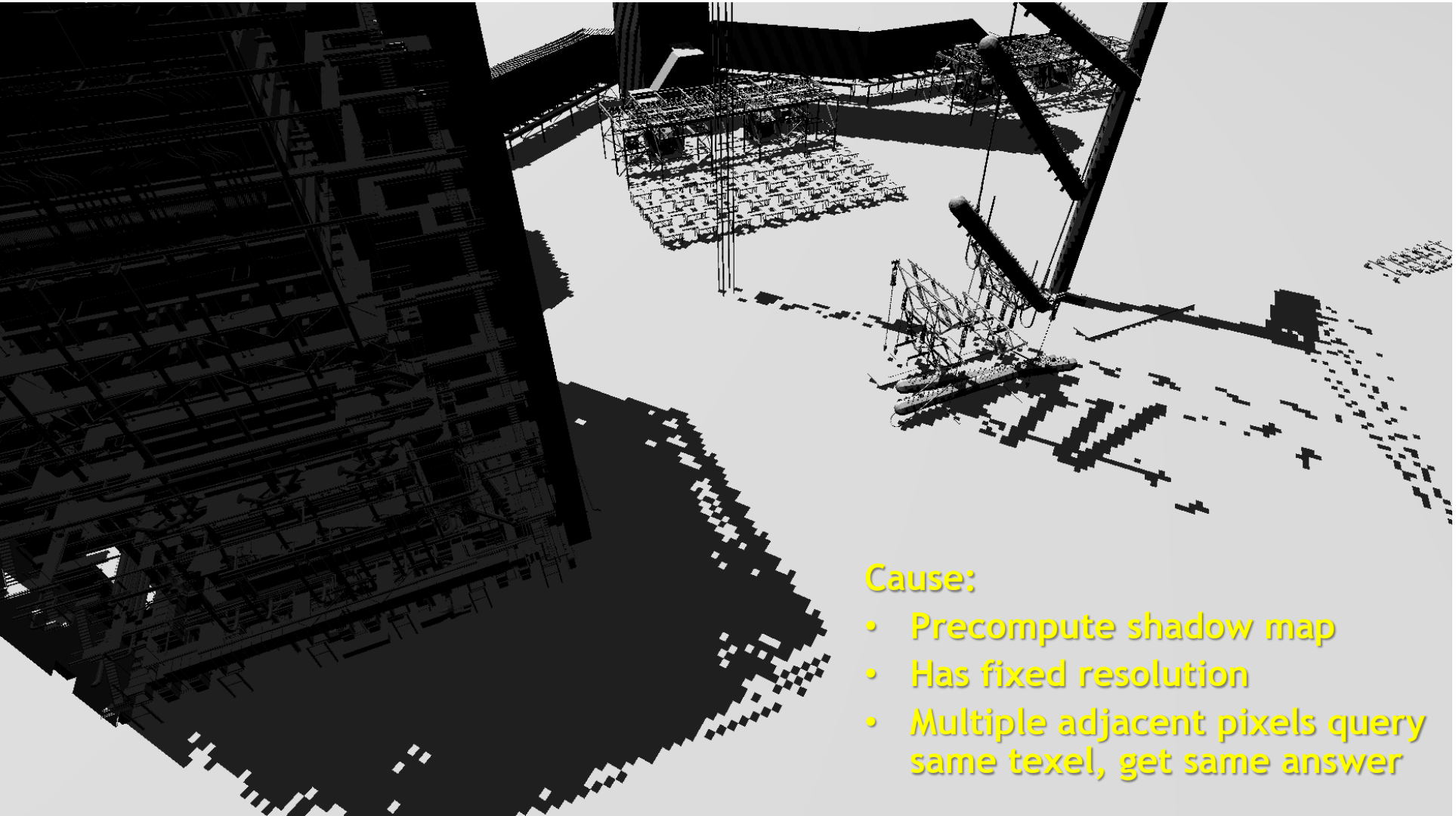
Builds on a “irregular z-buffer” for ray acceleration

- Not a traditional BVH or kd-tree
- Irregular z-buffers regarded as a dead end 3 years ago

WHY IS THIS WORTH INVESTIGATING?



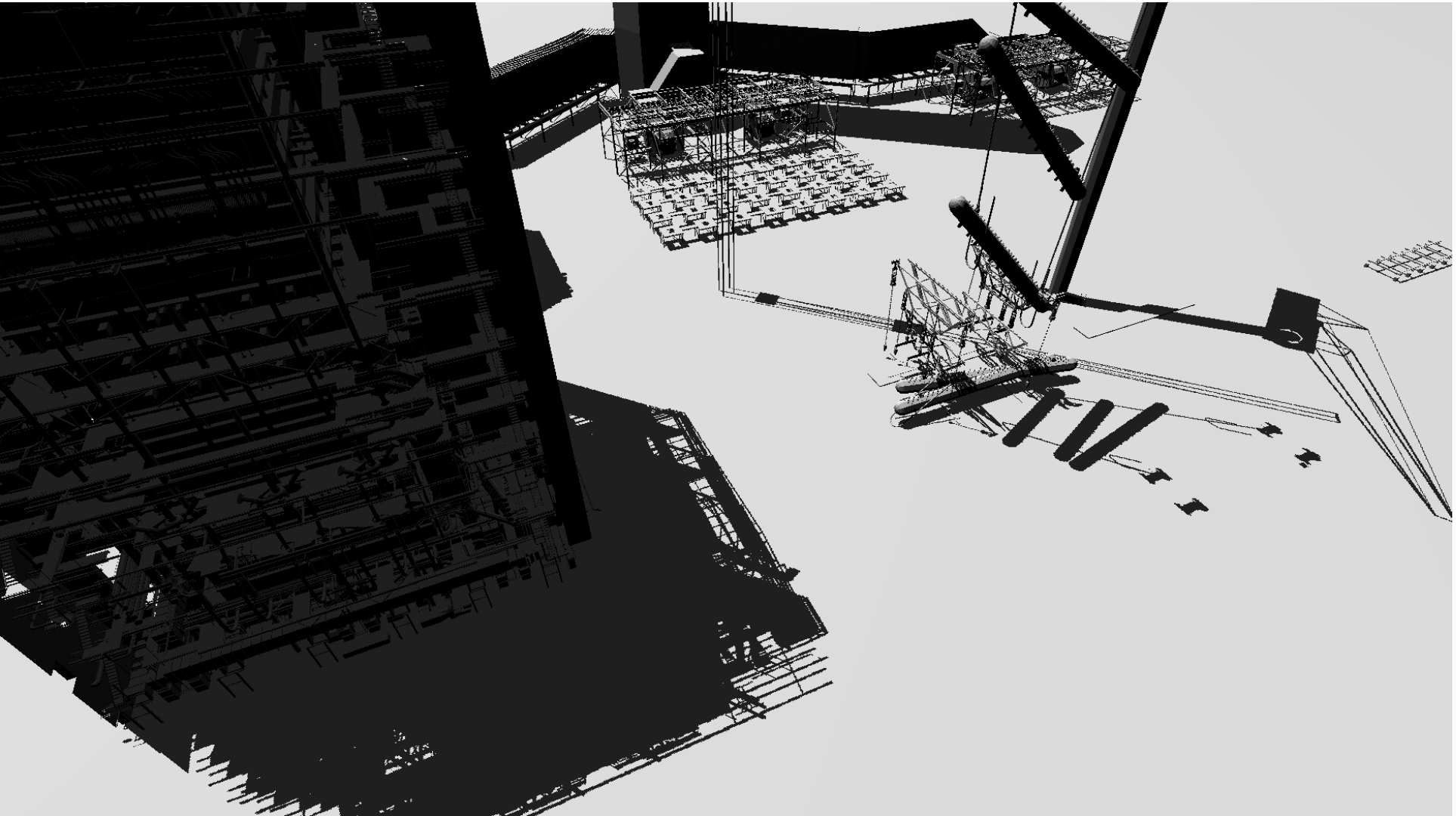
WHY IS THIS WORTH INVESTIGATING?



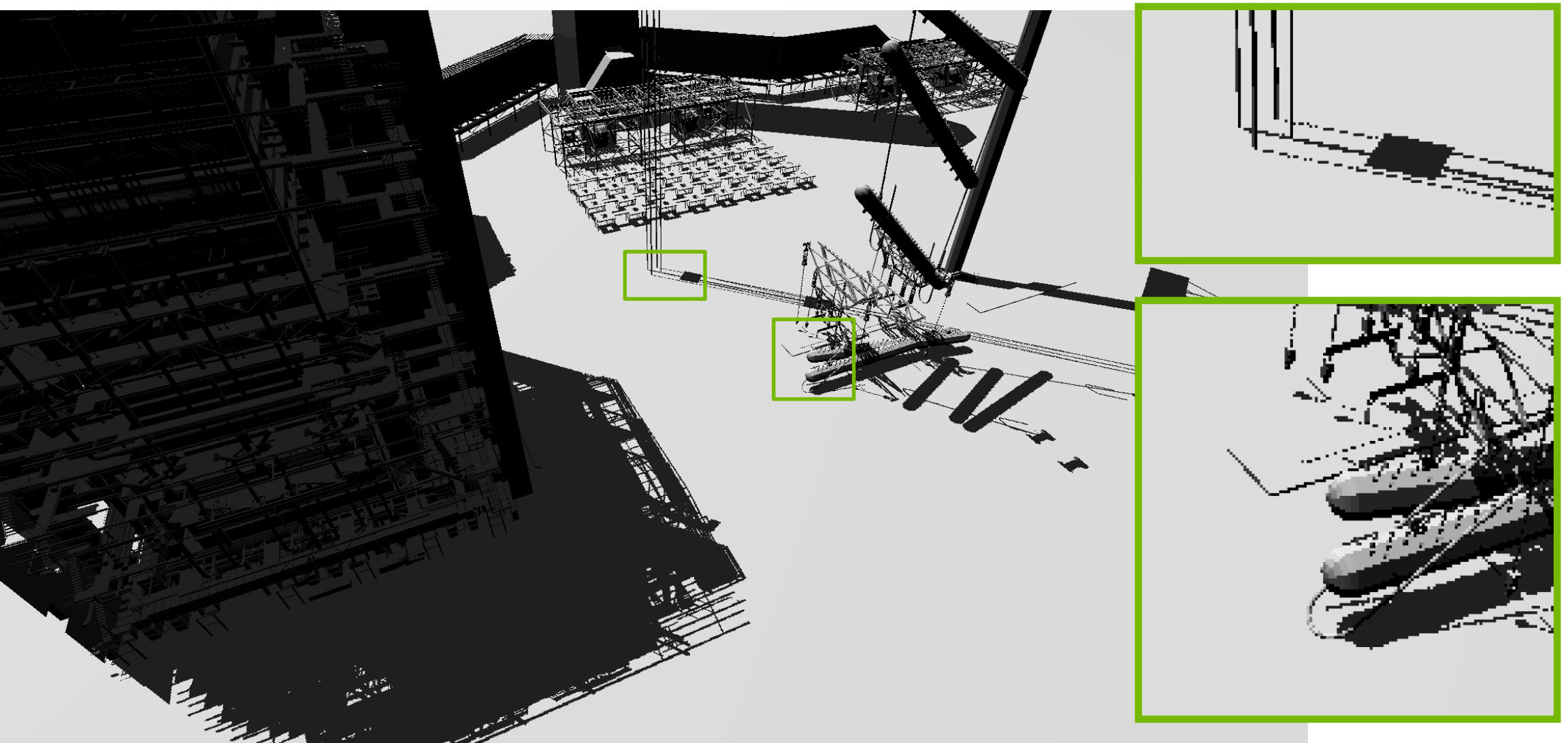
Cause:

- Precompute shadow map
- Has fixed resolution
- Multiple adjacent pixels query same texel, get same answer

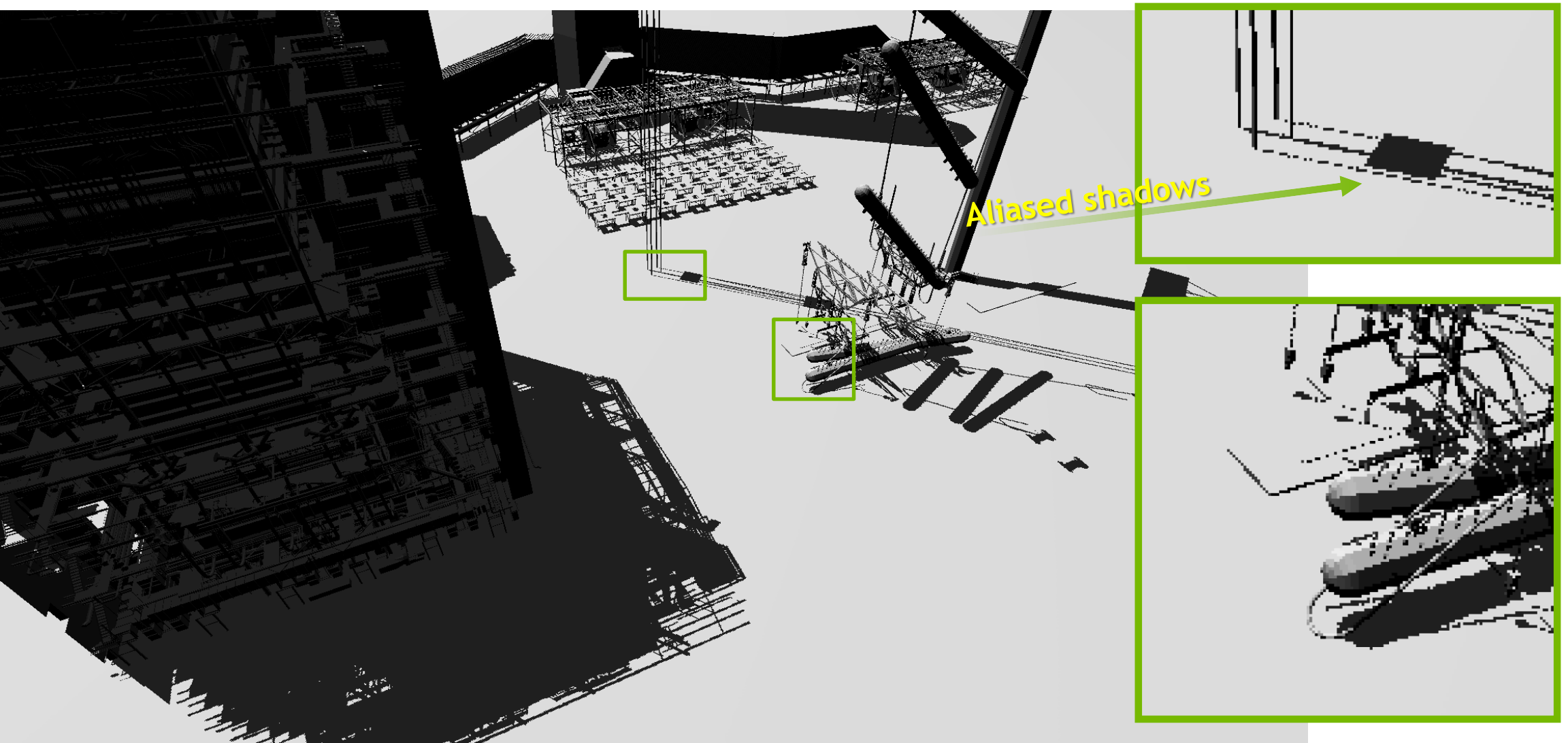
ALIASES EVEN WITH HIGH RESOLUTION



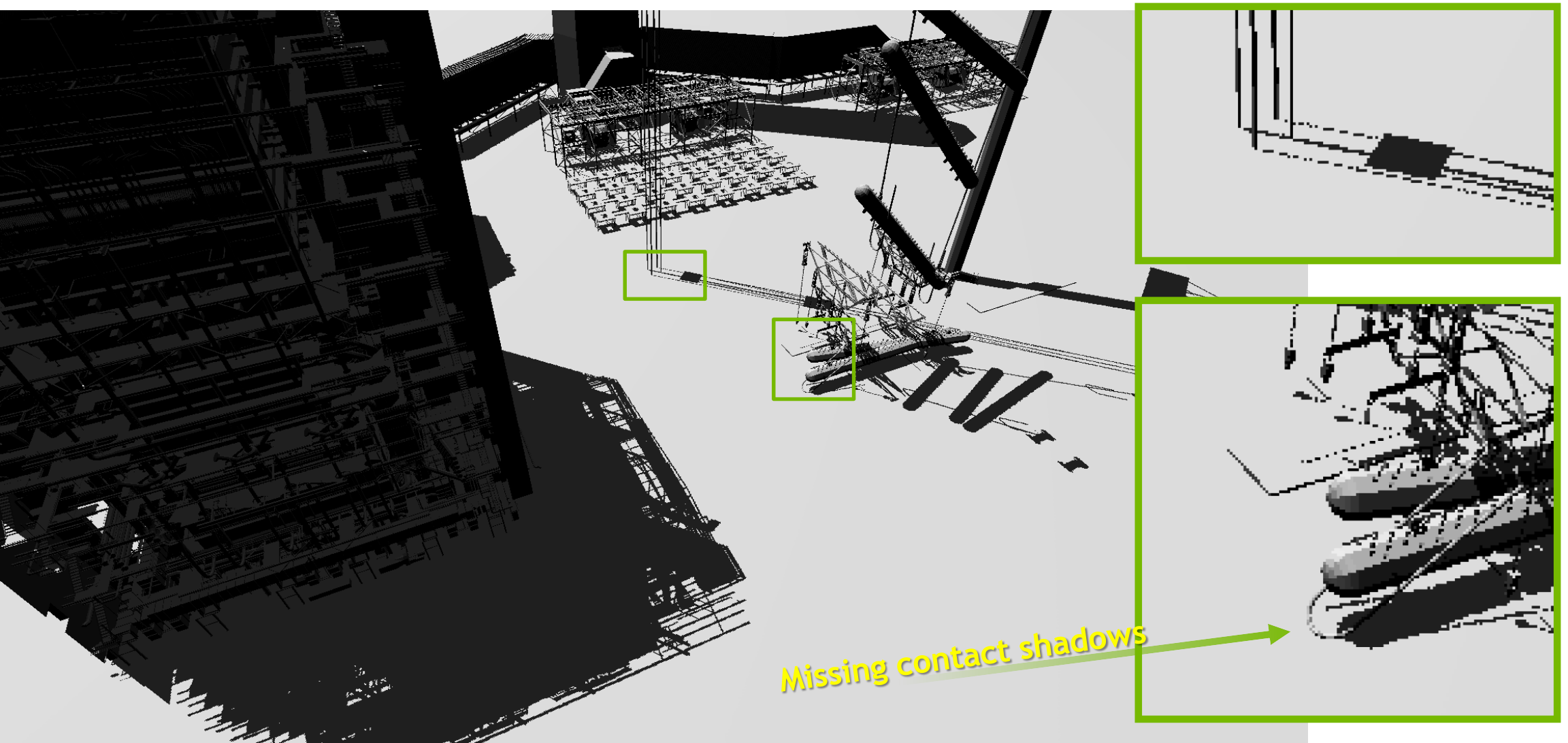
ALIASES EVEN WITH HIGH RESOLUTION



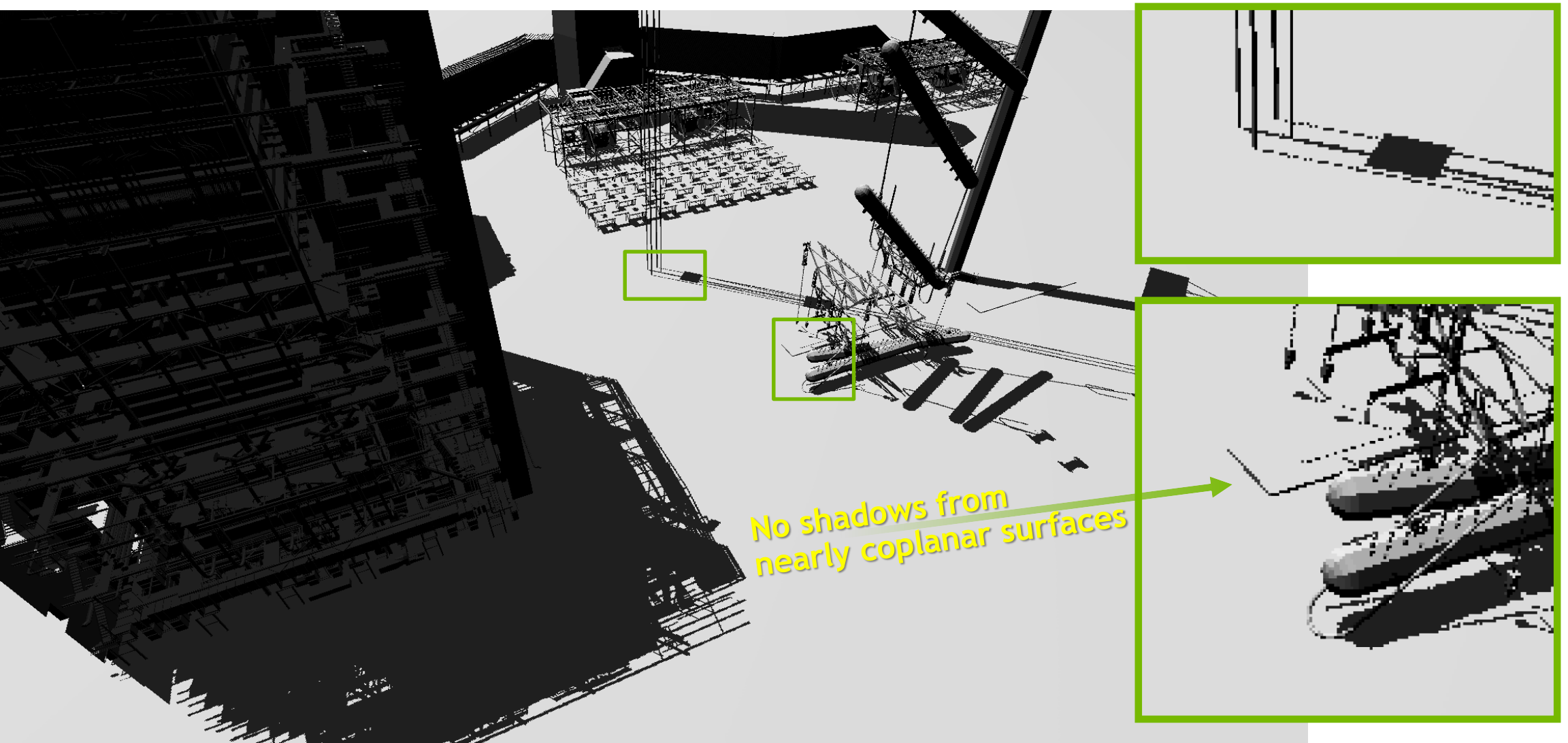
ALIASES EVEN WITH HIGH RESOLUTION



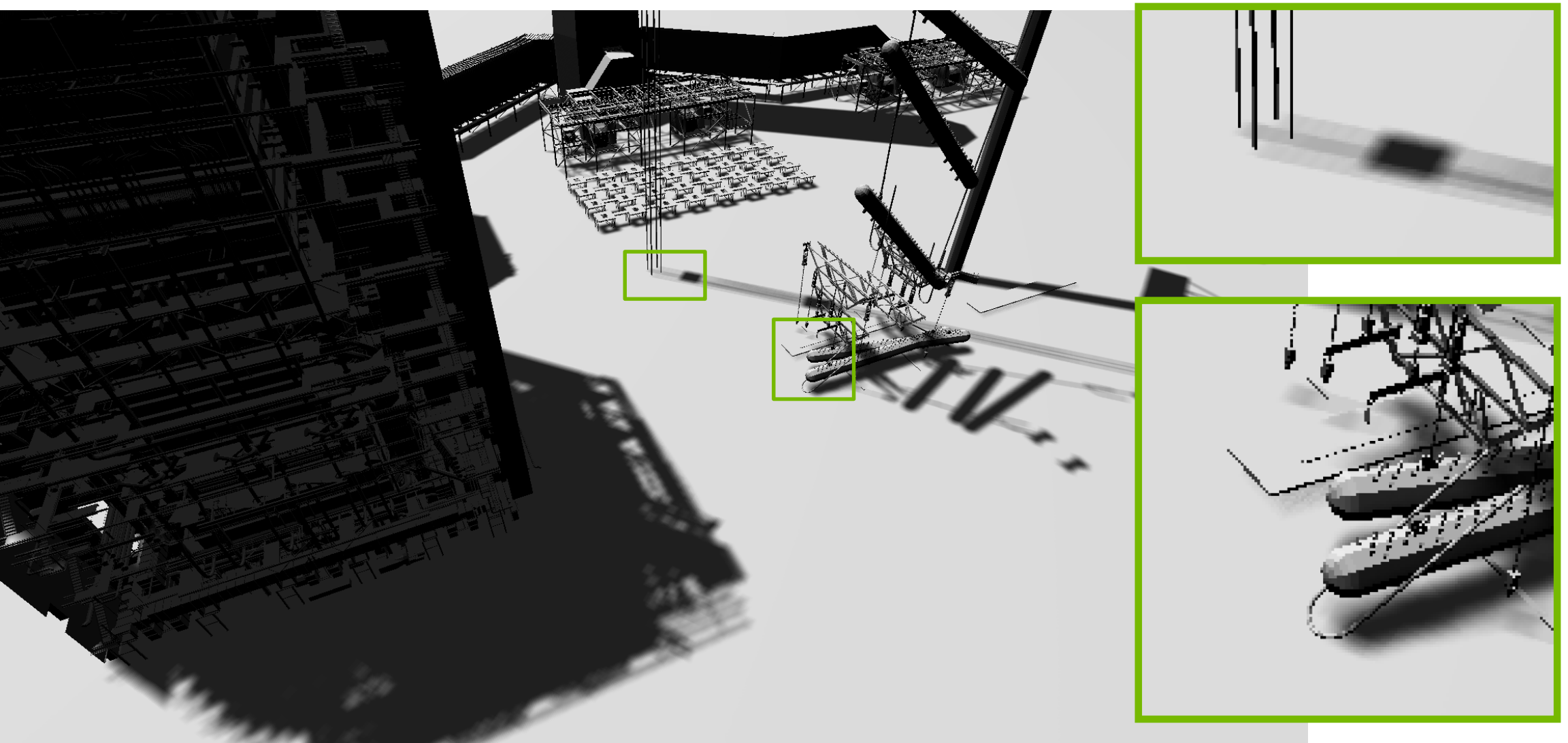
ALIASES EVEN WITH HIGH RESOLUTION



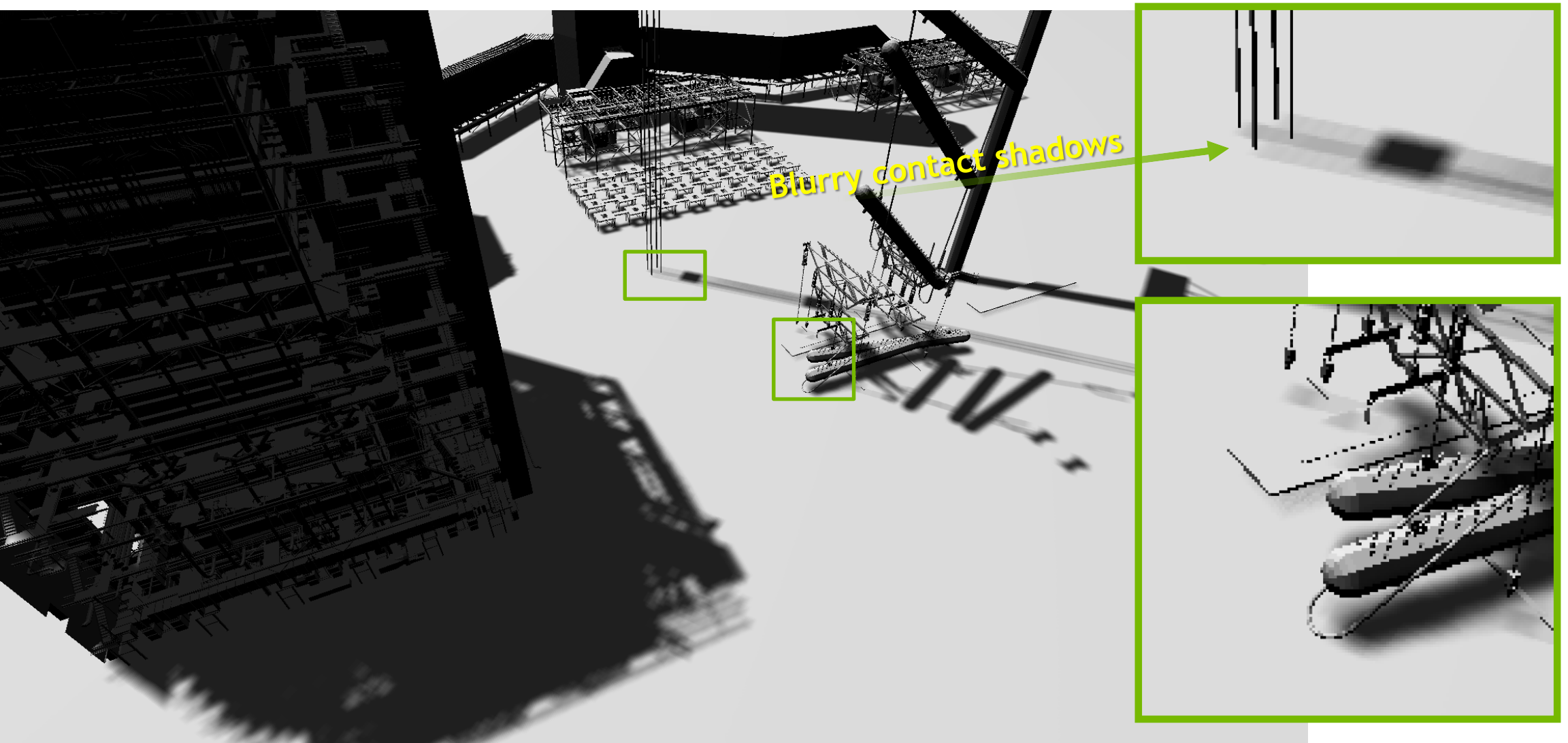
ALIASES EVEN WITH HIGH RESOLUTION



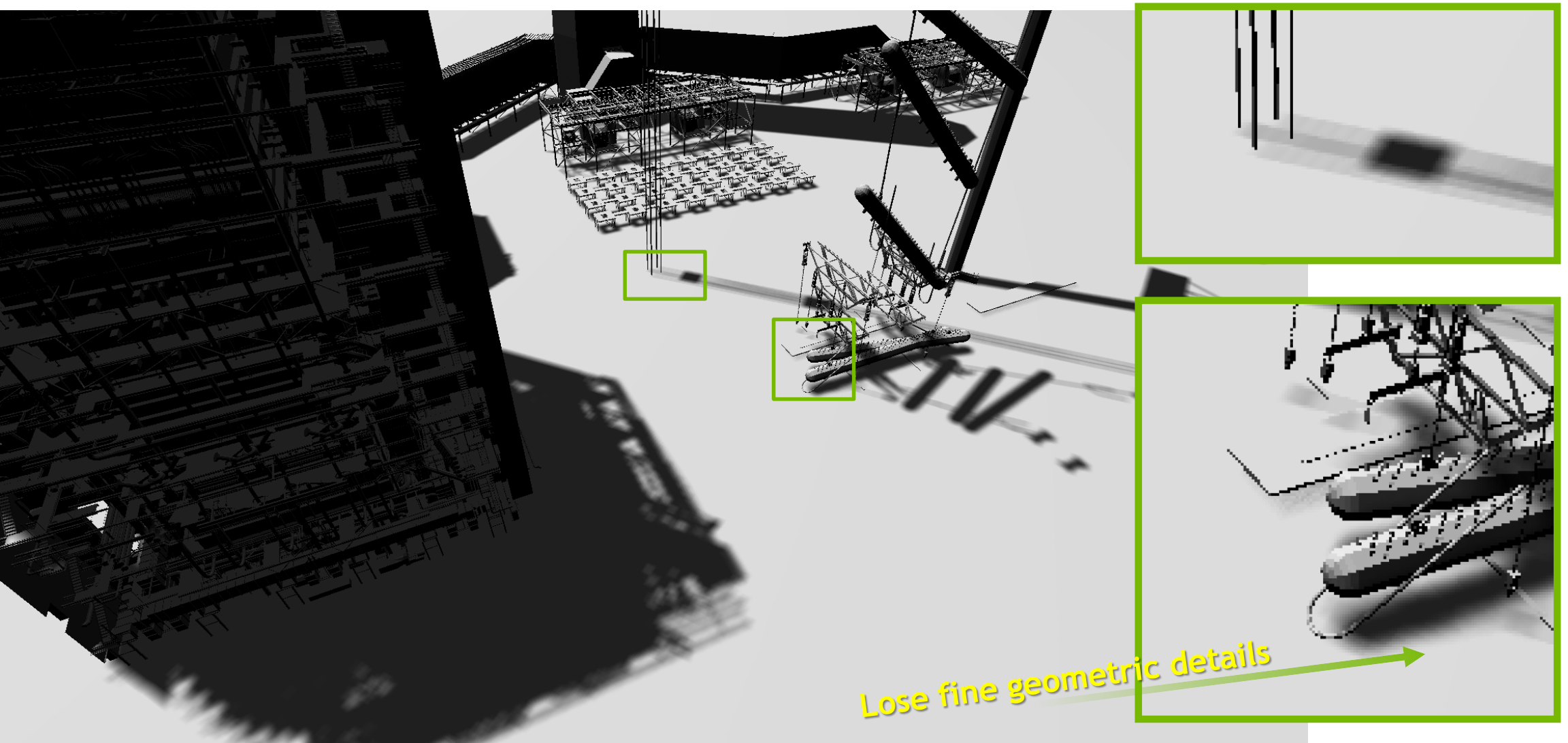
FILTERING SHADOW MAPS HELP



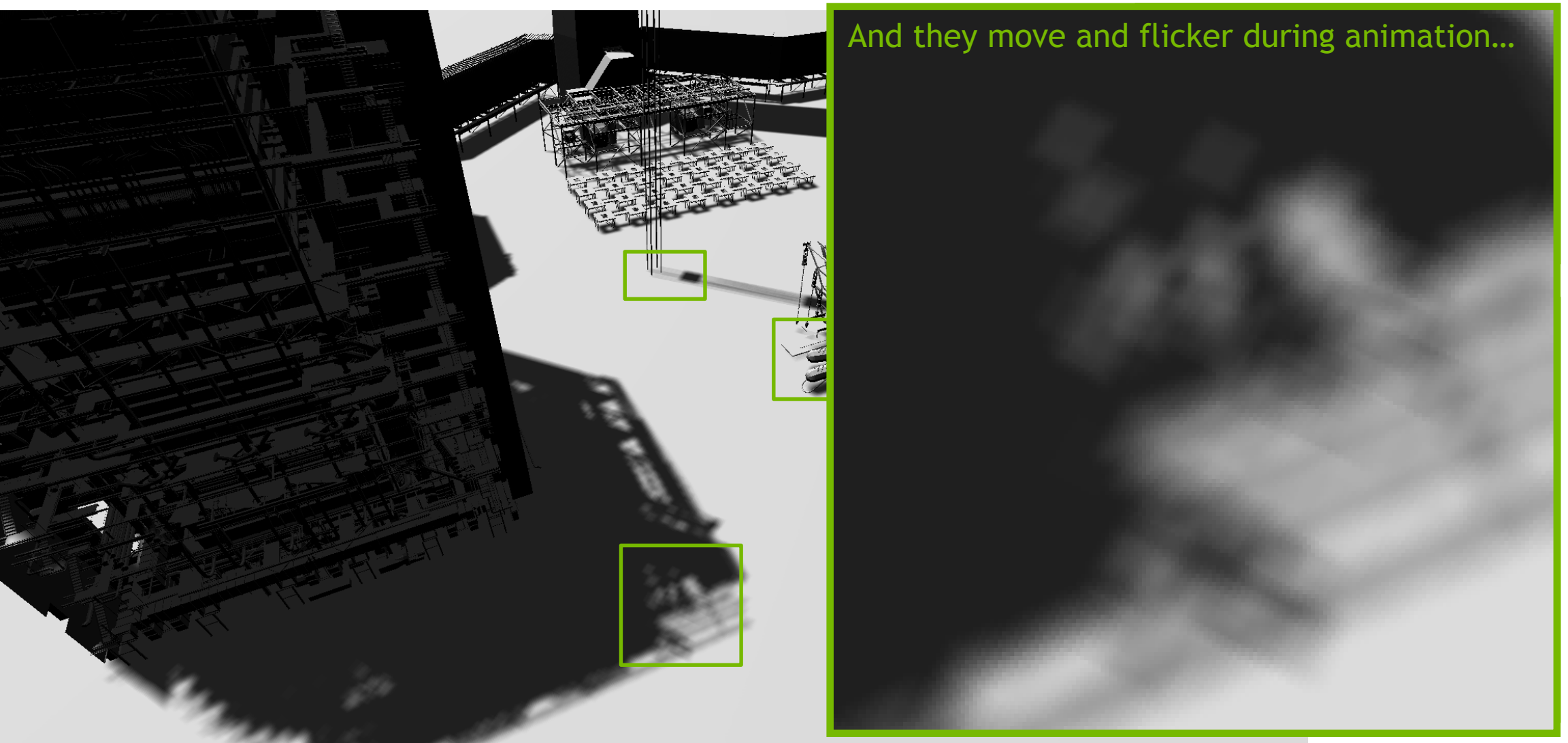
FILTERING SHADOW MAPS HELP



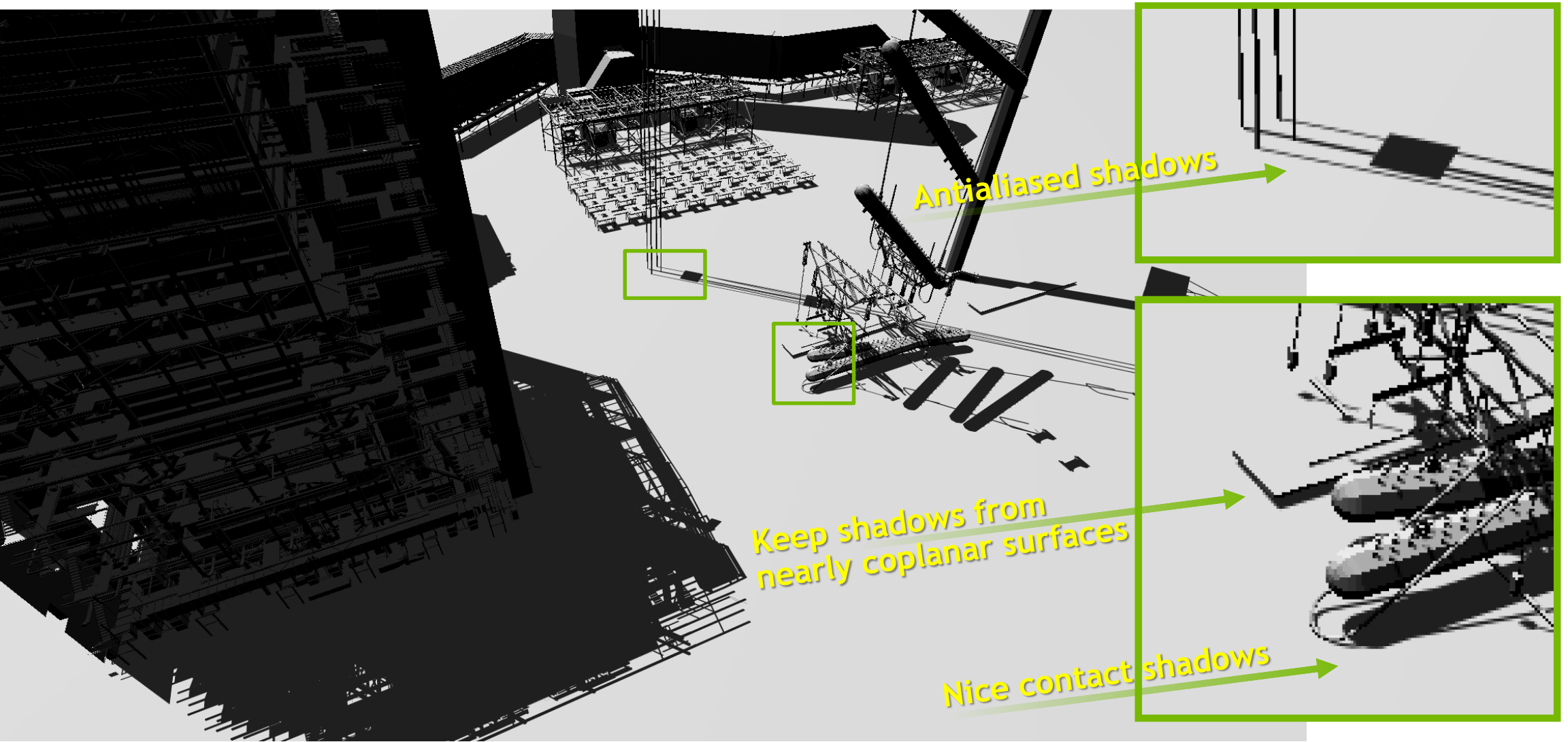
FILTERING SHADOW MAPS HELP



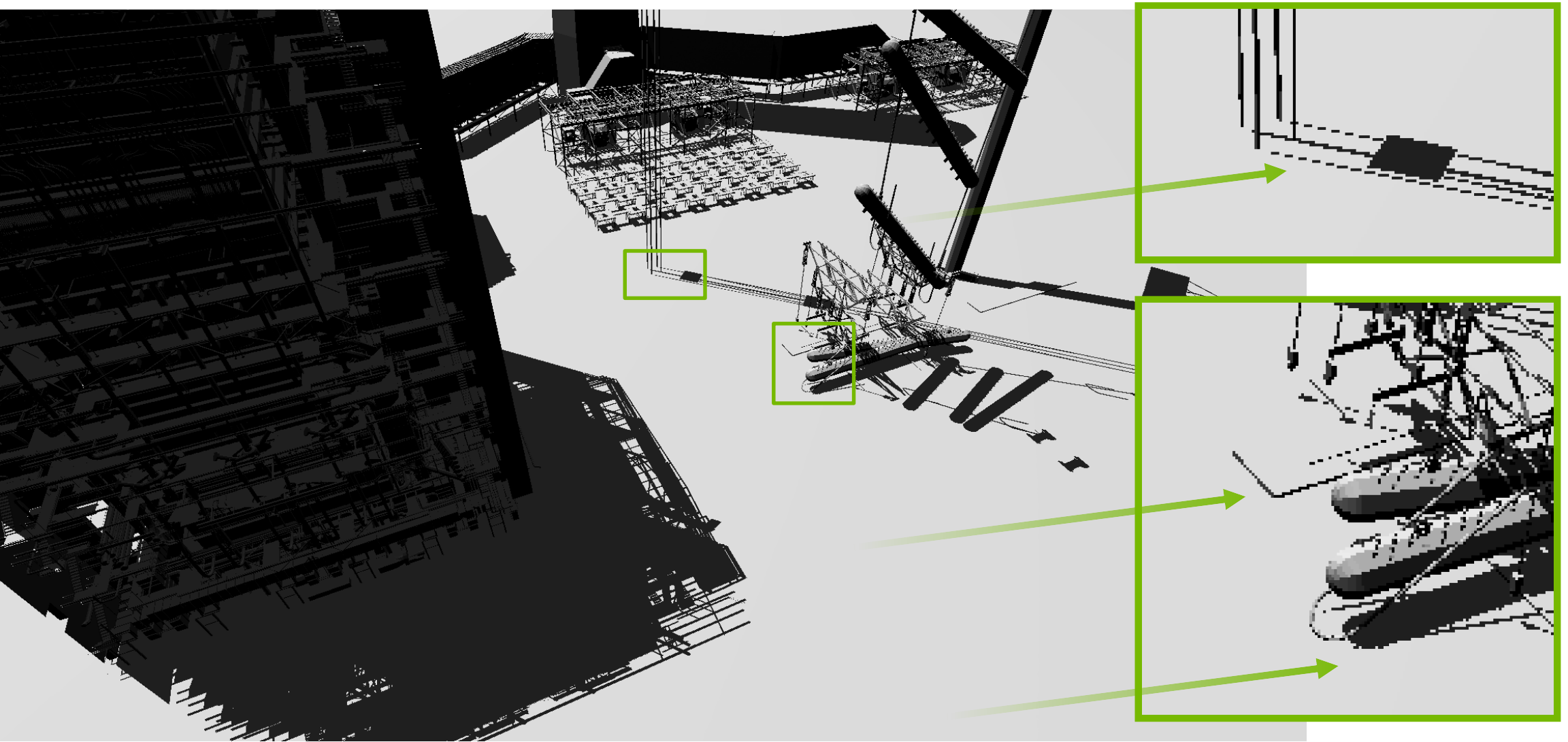
AND BLOCKS STILL VISIBLE AFTER FILTERING!



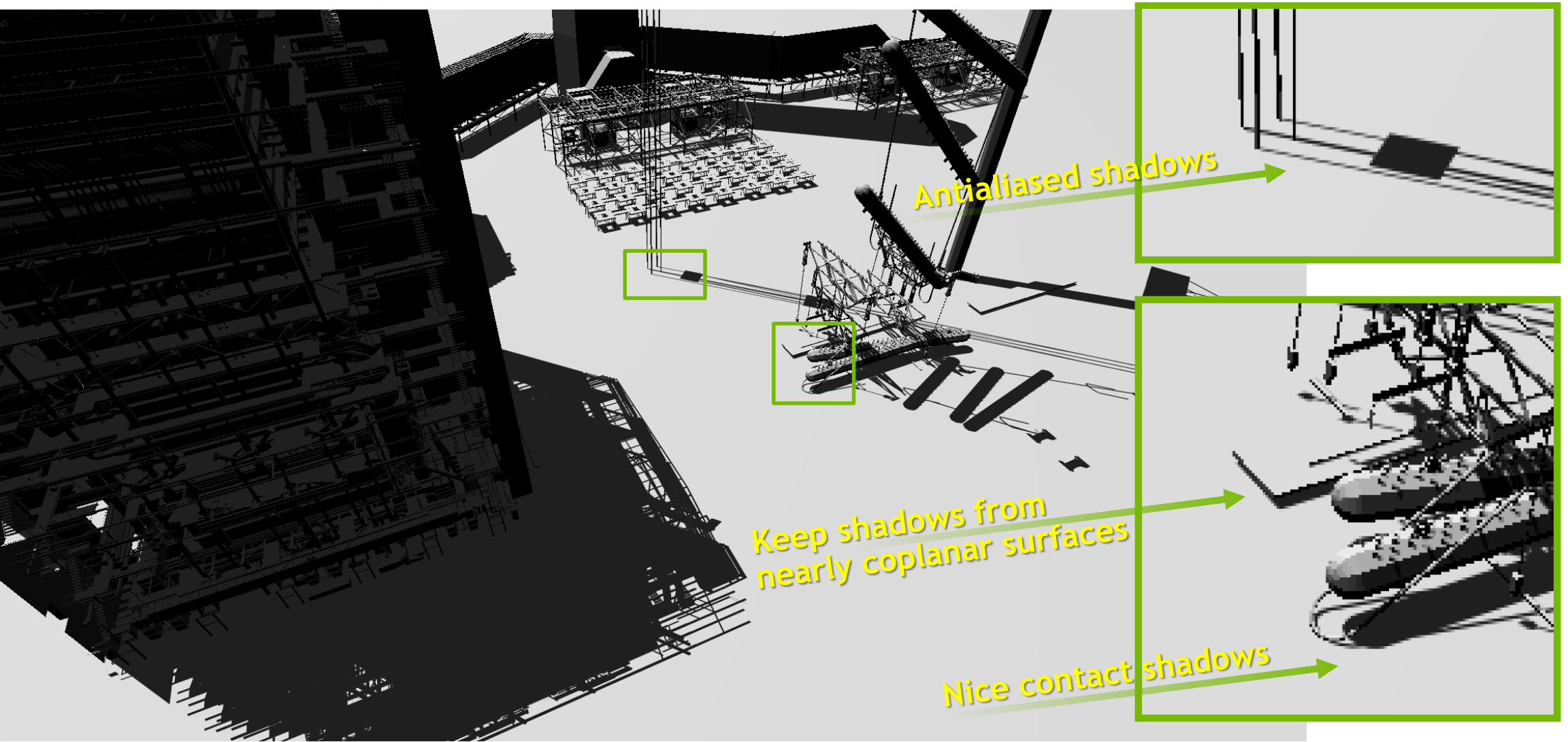
HIGH QUALITY RAY TRACING



HIGH QUALITY SHADOW MAP



HIGH QUALITY RAY TRACING



USING RAY TRACING TODAY

Requires separate ray tracing libraries, APIs, and acceleration structures:

- May need separate geometric representation
- Data structure rebuild traditionally costly (for dynamic scenes)

USING RAY TRACING TODAY

Requires separate ray tracing libraries, APIs, and acceleration structures:

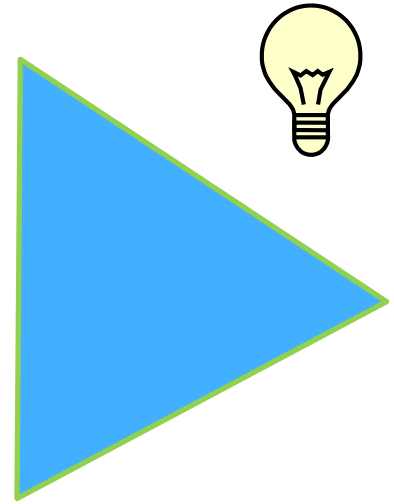
- May need separate geometric representation
- Data structure rebuild traditionally costly (for dynamic scenes)

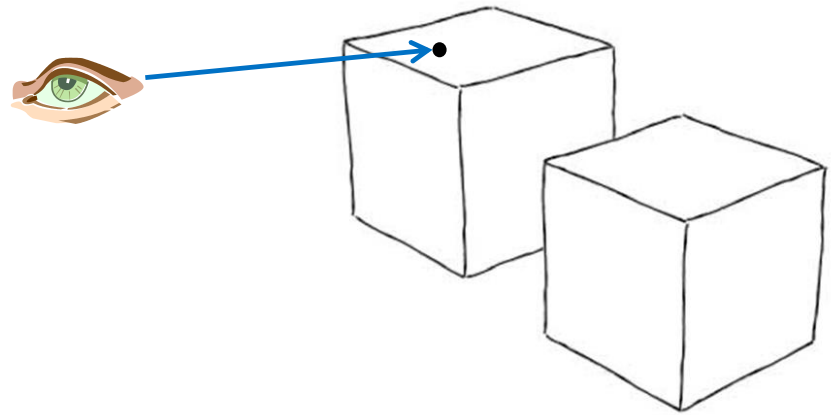
Our goals:

- **Specialize ray tracing for hard shadows**
- **Build on existing APIs (DirectX, OpenGL, Vulkan) and geometric representations**
- **Quickly build a new data structure each frame**

WHAT IS RAY TRACING?

Query visibility along arbitrary rays



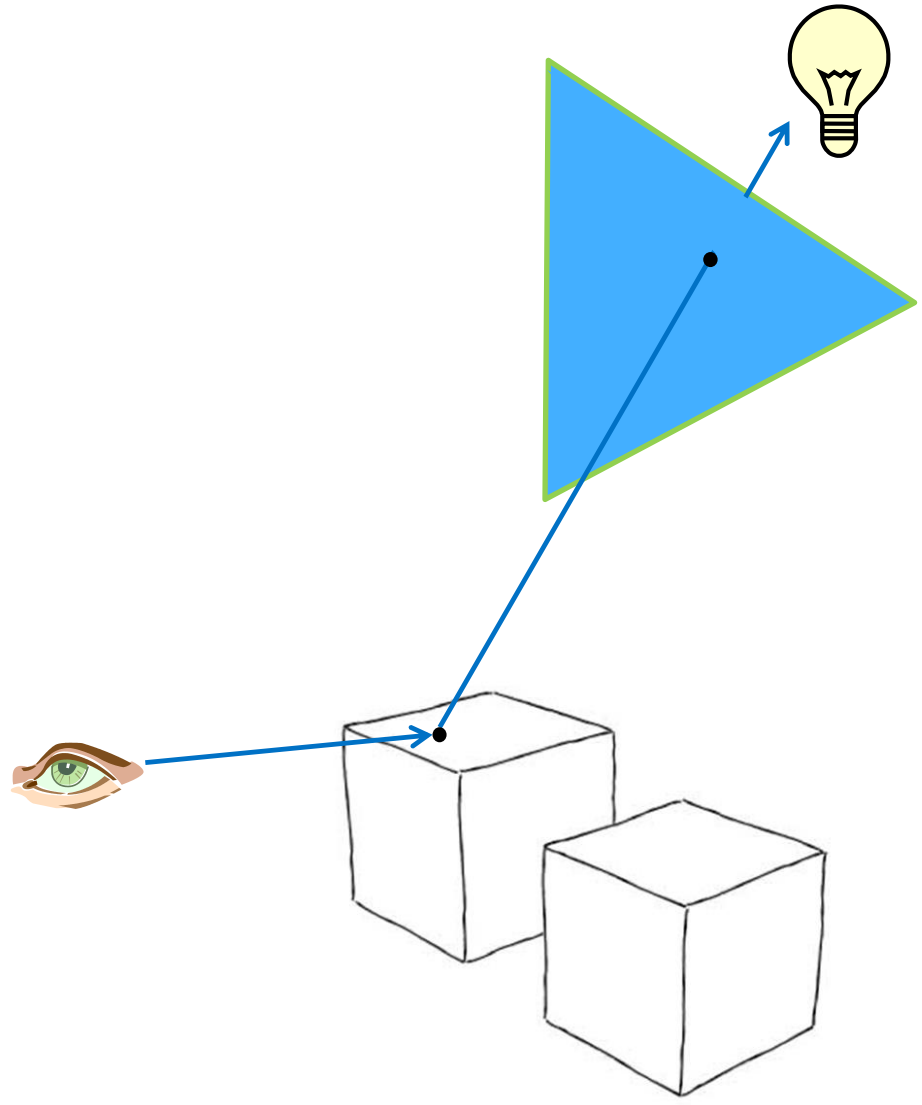


WHAT IS RAY TRACING?

Query visibility along arbitrary rays

To shadow each pixel, test ray to light

- If occluded, pixel shadowed
- If unoccluded, pixel lit



WHAT IS RAY TRACING?

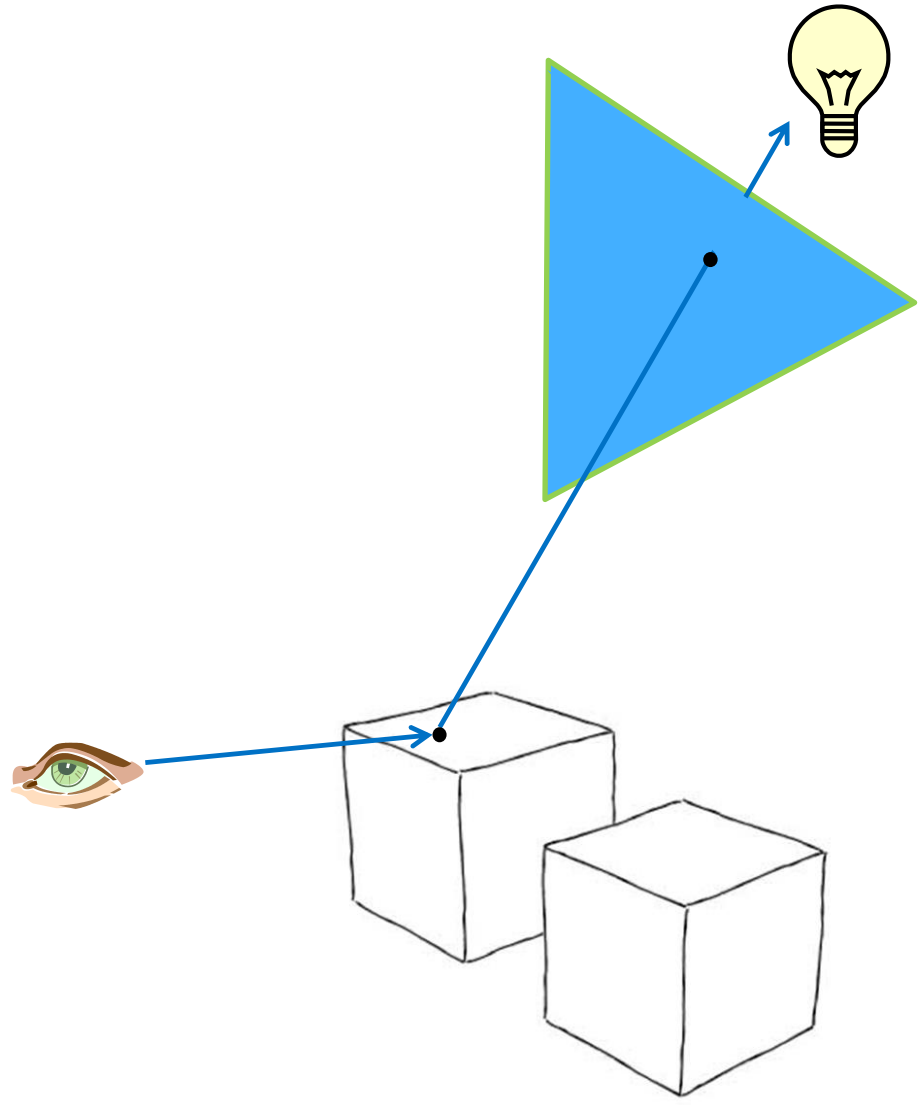
Query visibility along arbitrary rays

To shadow each pixel, test ray to light

- If occluded, pixel shadowed
- If unoccluded, pixel lit

Avoids problems with shadow maps

- Light visibility not precomputed
- Computations exactly match pixel locations



MAKING SHADOW RAY TRACING FAST

Typical ray tracer is extremely general

- 10s, 100s, or 1000s of rays per pixel
- Incoherent memory access
- Unknown reflectance of surfaces in scene

From WikiPedia

MAKING SHADOW RAY TRACING FAST

Typical ray tracer is extremely general

- 10s, 100s, or 1000s of rays per pixel
- Incoherent memory access
- Unknown reflectance of surfaces in scene

Specializing for shadows helps

- Only care about binary visibility per ray

From WikiPedia

MAKING SHADOW RAY TRACING FAST

Typical ray tracer is extremely general

- 10s, 100s, or 1000s of rays per pixel
- Incoherent memory access
- Unknown reflectance of surfaces in scene

Specializing for shadows helps

- Only care about binary visibility per ray

Specializing for ***hard*** shadows helps even more

- Know all rays go to same location (i.e., the point light)
- Starts to look like raster, with irregular samples

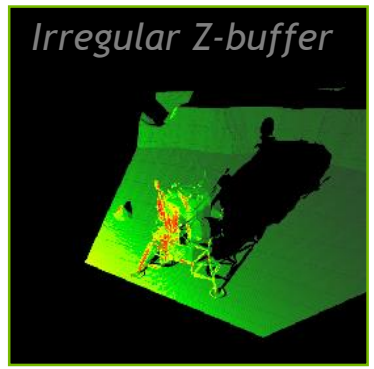
From WikiPedia

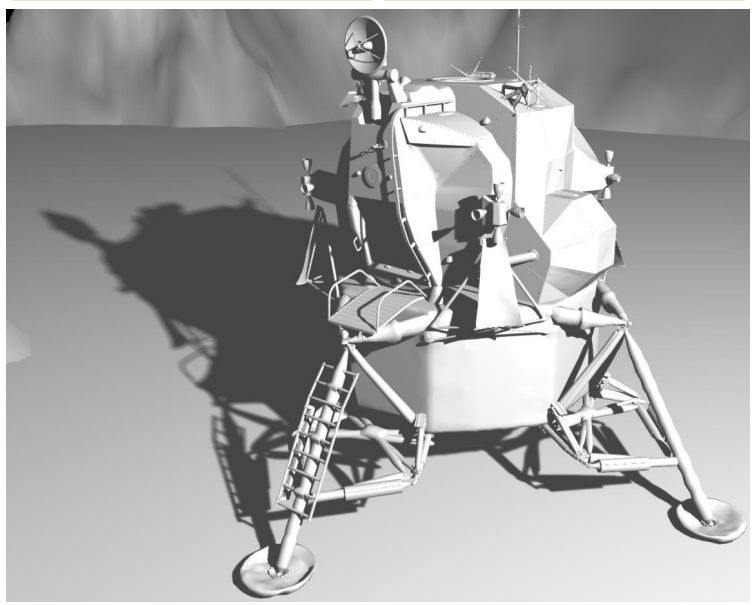
DATA STRUCTURE: IRREGULAR Z-BUFFER

Accelerates queries emanating from a point

Can efficiently build and traverse in parallel

- Fully rebuilds in < 1 ms per frame





DATA STRUCTURE: IRREGULAR Z-BUFFER

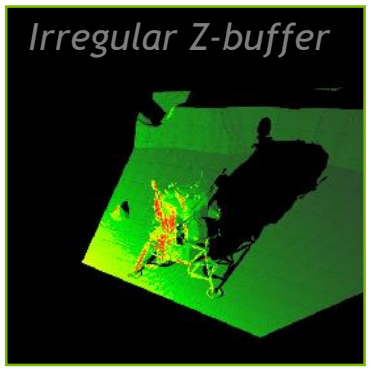
Accelerates queries emanating from a point

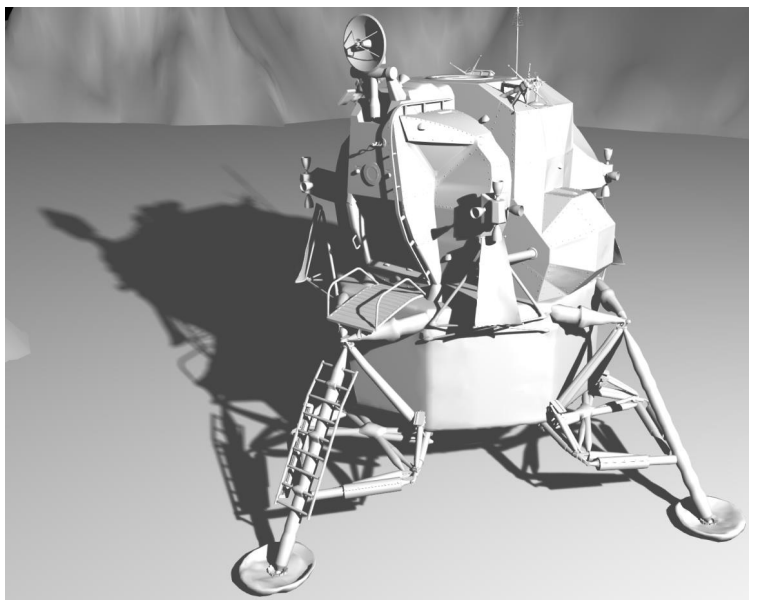
Can efficiently build and traverse in parallel

- Fully rebuilds in < 1 ms per frame

A type of ray caching

- Stores ray endpoints rather than triangles
- Reorders rays; allows ray tracing via raster hardware
- Leverage shadow map techniques for more perf wins

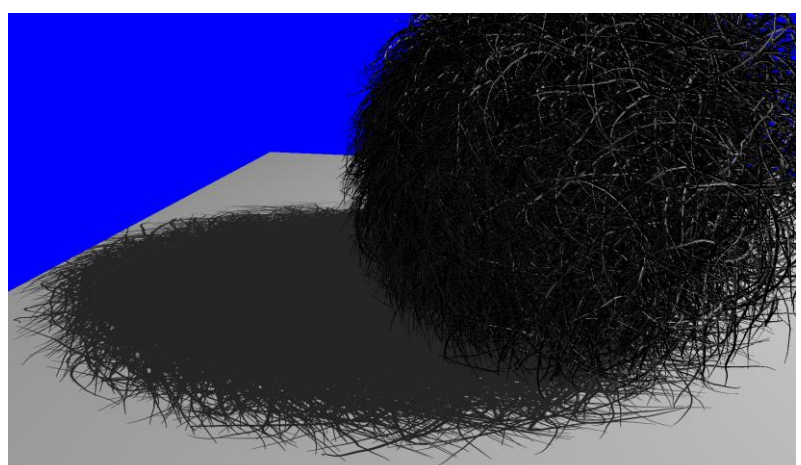




WHY HAS NOBODY ELSE DONE THIS?

Irregular z-buffering is ***hard***

- 3 years ago, was a “dead end” in academic research
- Our 1st prototype cost >2 sec for this frame (now <5 ms; a 400x speedup)

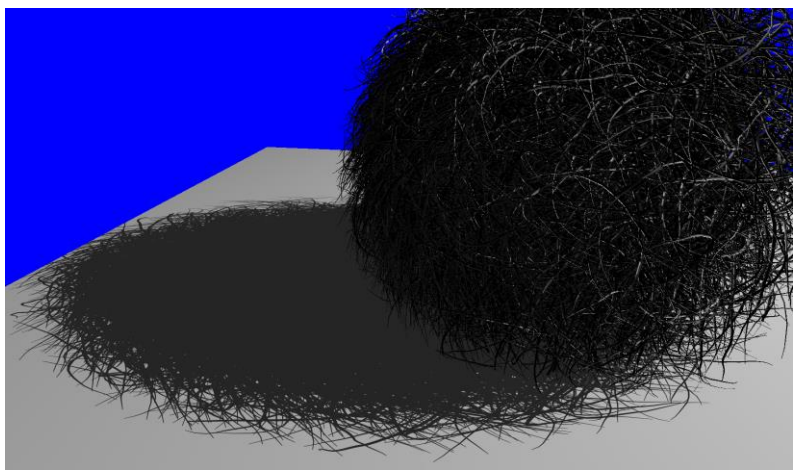


WHY HAS NOBODY ELSE DONE THIS?

Irregular z-buffering is ***hard***

- 3 years ago, was a “dead end” in academic research
- Our 1st prototype cost >2 sec for this frame (now <5 ms; a 400x speedup)

Bad: Costs increased linearly with # pixels & polygons



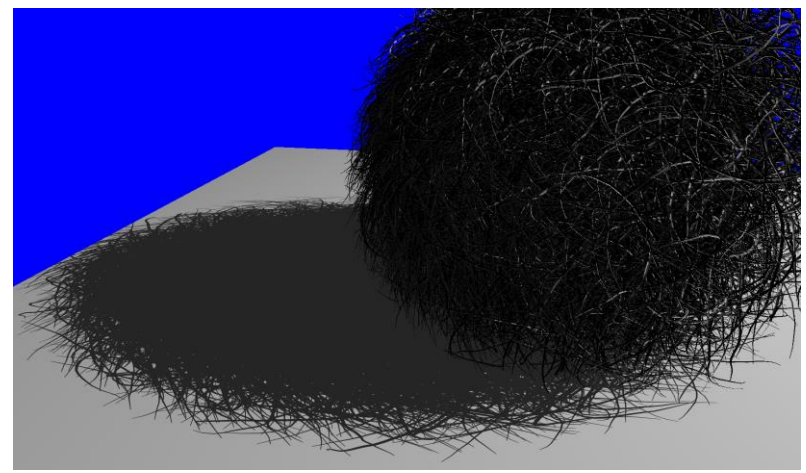
WHY HAS NOBODY ELSE DONE THIS?

Irregular z-buffering is ***hard***

- 3 years ago, was a “dead end” in academic research
- Our 1st prototype cost >2 sec for this frame (now <5 ms; a 400x speedup)

Bad: Costs increased linearly with # pixels & polygons

Worse: Performance could vary 100:1 between frames



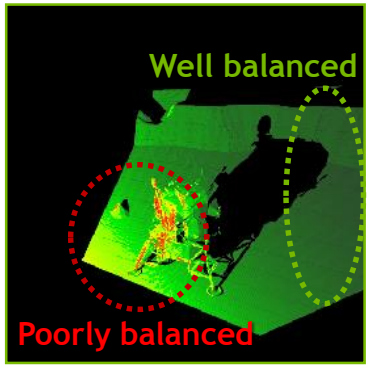
MAKING IRREGULAR Z-BUFFERS USABLE

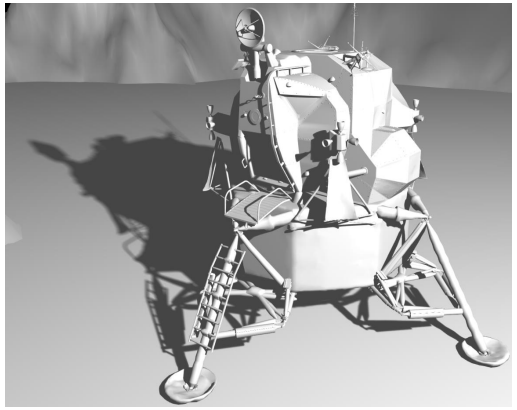
IZBs *eliminate* aliasing, *converting* it to performance variability

MAKING IRREGULAR Z-BUFFERS USABLE

IZBs *eliminate* aliasing, *converting* it to performance variability

- If shadow maps alias, many pixels correspond to one texel
- IZBs have to enumerate, cache, and reorder these pixels
- Coverts aliasing into a parallel load balancing problem
- Poor load balancing = poor GPU performance





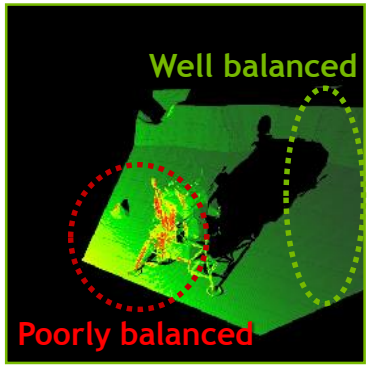
MAKING IRREGULAR Z-BUFFERS USABLE

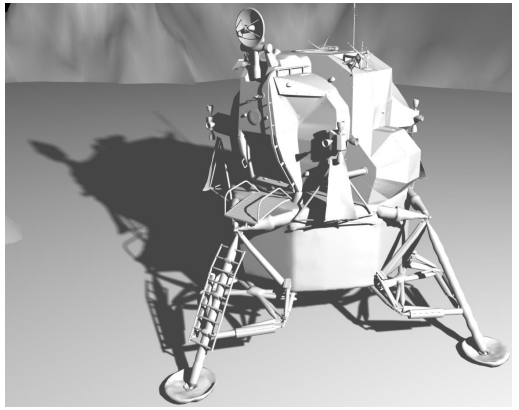
IZBs *eliminate* aliasing, *converting* it to performance variability

- If shadow maps alias, many pixels correspond to one texel
- IZBs have to enumerate, cache, and reorder these pixels
- Coverts aliasing into a parallel load balancing problem
- Poor load balancing = poor GPU performance

Our research:

- **First, identified this problem**
- **Second, proposed a simple solution implementable today**

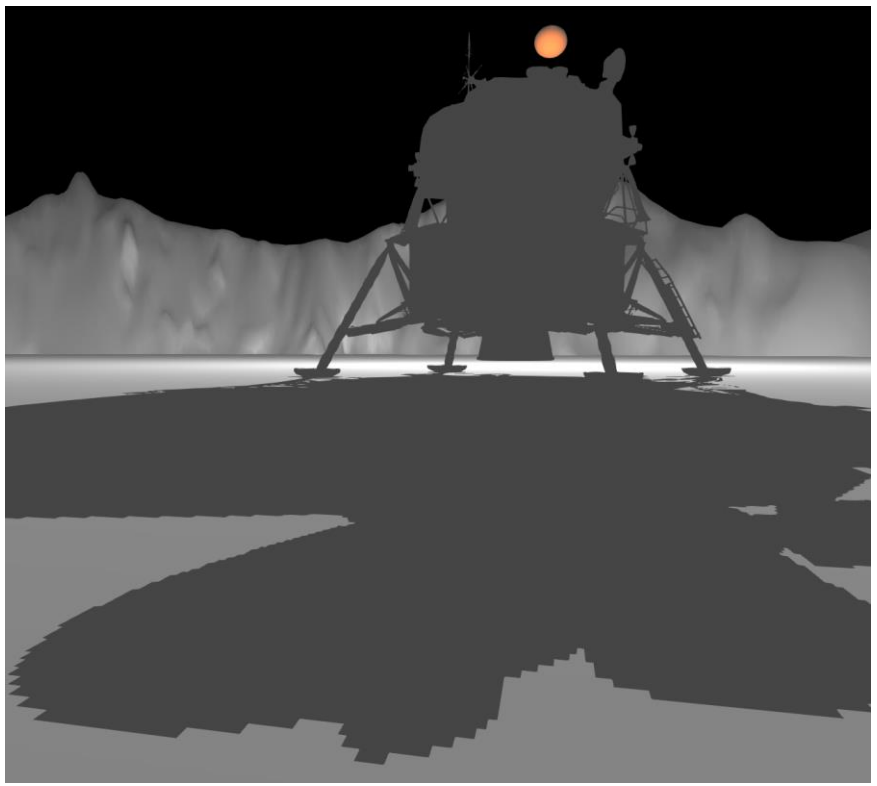




HOW TO LOAD BALANCE

Even well designed shadow map implementations alias badly from some views

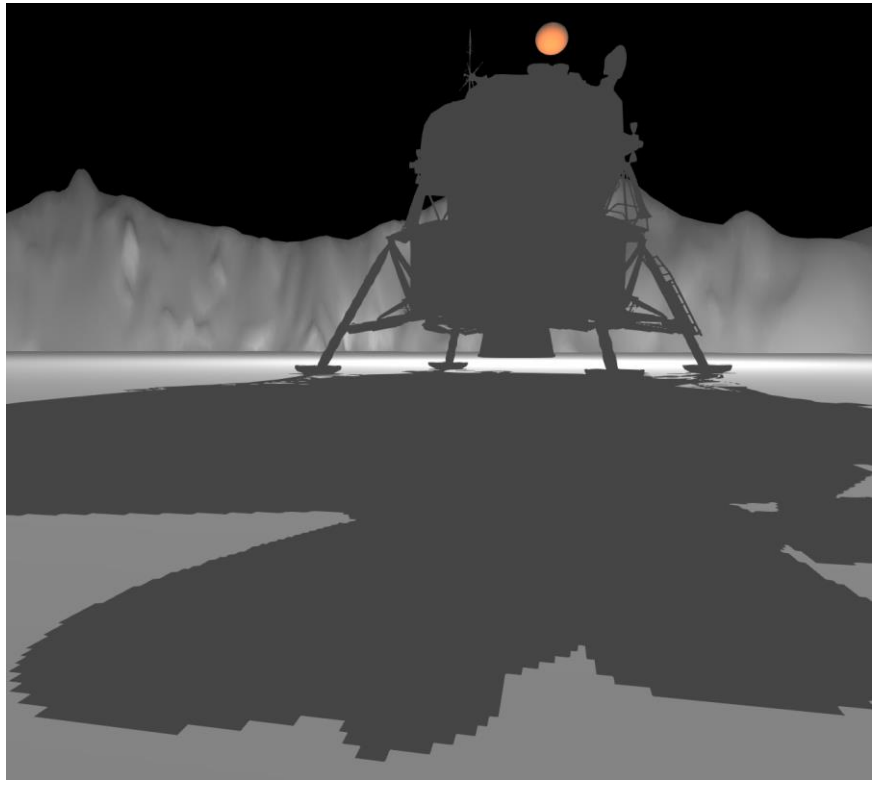
- Nearby texels here 100:1 larger than distant ones



HOW TO LOAD BALANCE

Even well designed shadow map implementations alias badly from some views

- Nearby texels here 100:1 larger than distant ones
- Hence the use of cascaded shadow maps
- Cascades reduce ***variability*** in aliasing



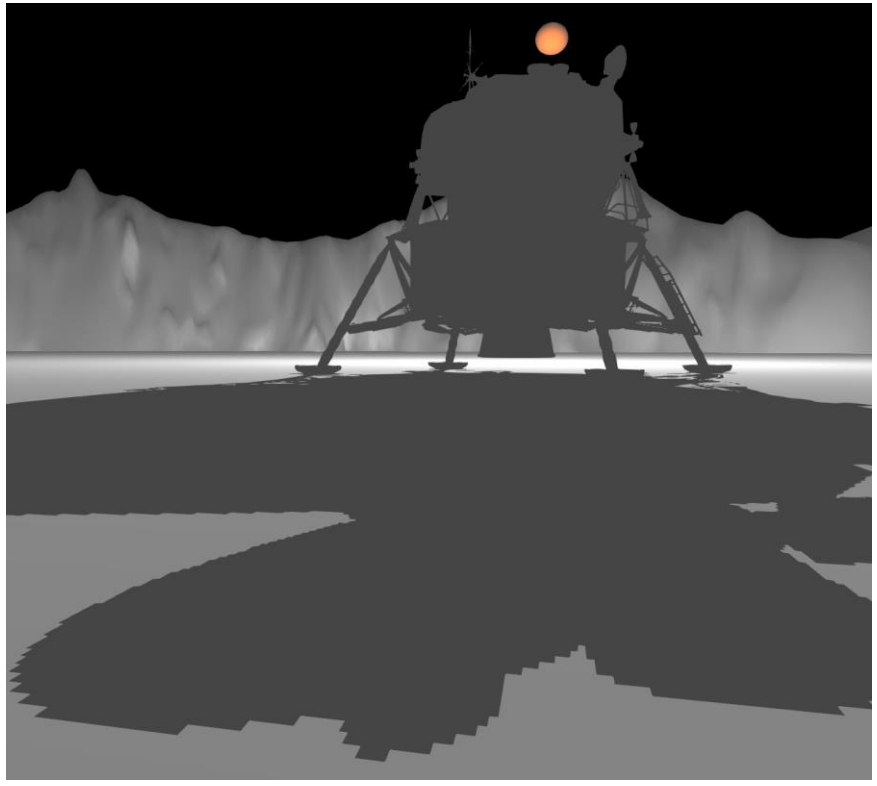
HOW TO LOAD BALANCE

Even well designed shadow map implementations alias badly from some views

- Nearby texels here 100:1 larger than distant ones
- Hence the use of cascaded shadow maps
- Cascades reduce ***variability*** in aliasing

IZBs convert aliasing to poor load balancing

- Some texels cost 100x more than others



HOW TO LOAD BALANCE

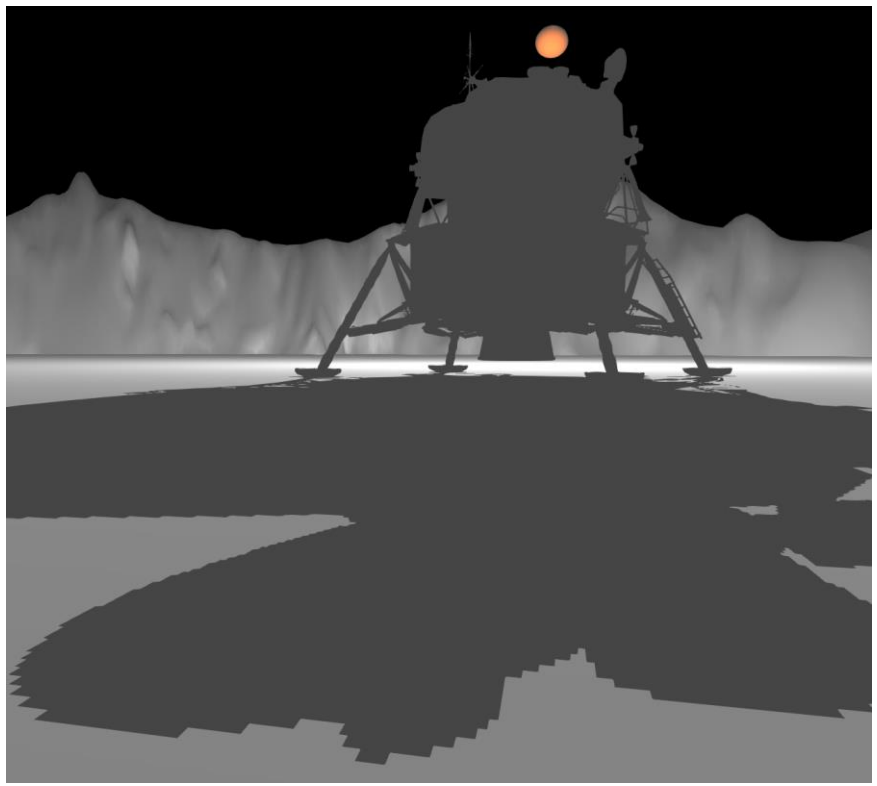
Even well designed shadow map implementations alias badly from some views

- Nearby texels here 100:1 larger than distant ones
- Hence the use of cascaded shadow maps
- Cascades reduce ***variability*** in aliasing

IZBs convert aliasing to poor load balancing

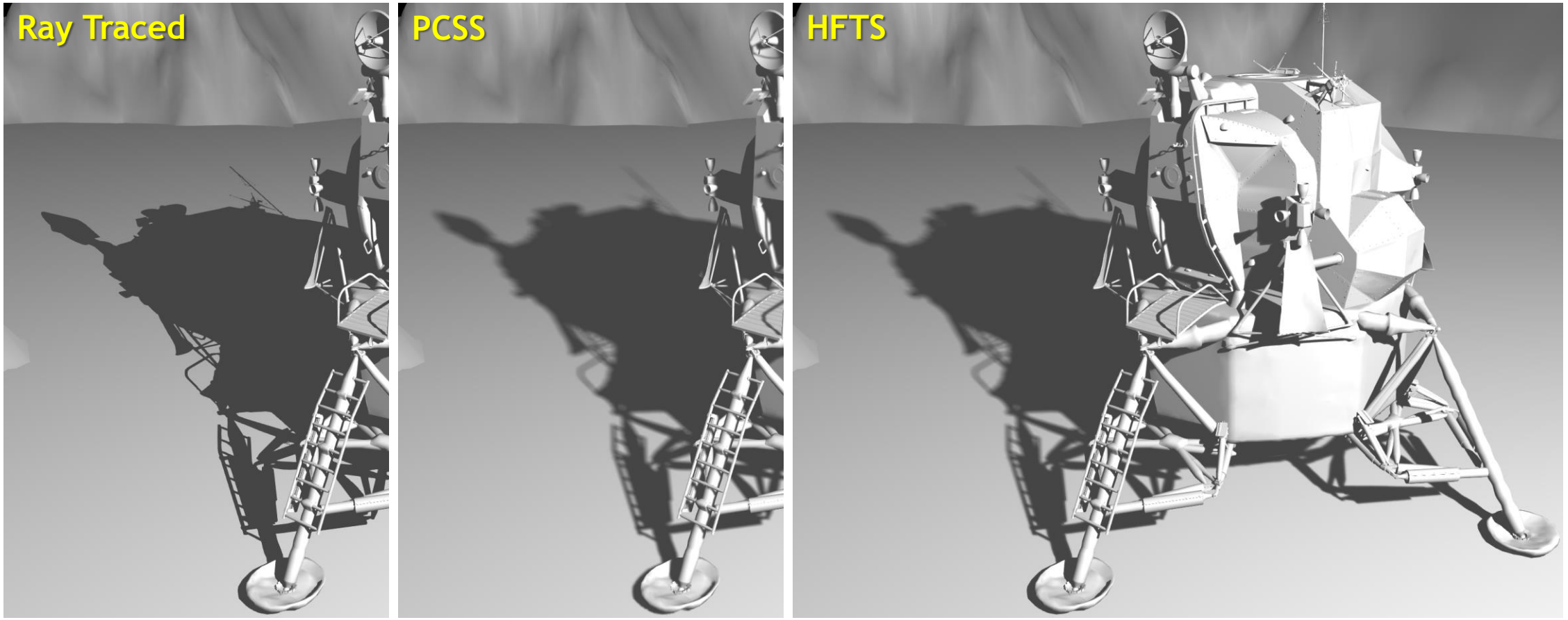
- Some texels cost 100x more than others
- Cascaded IZBs ***reduce this variability*** (to <<2x)
- Other shadow map techniques apply too

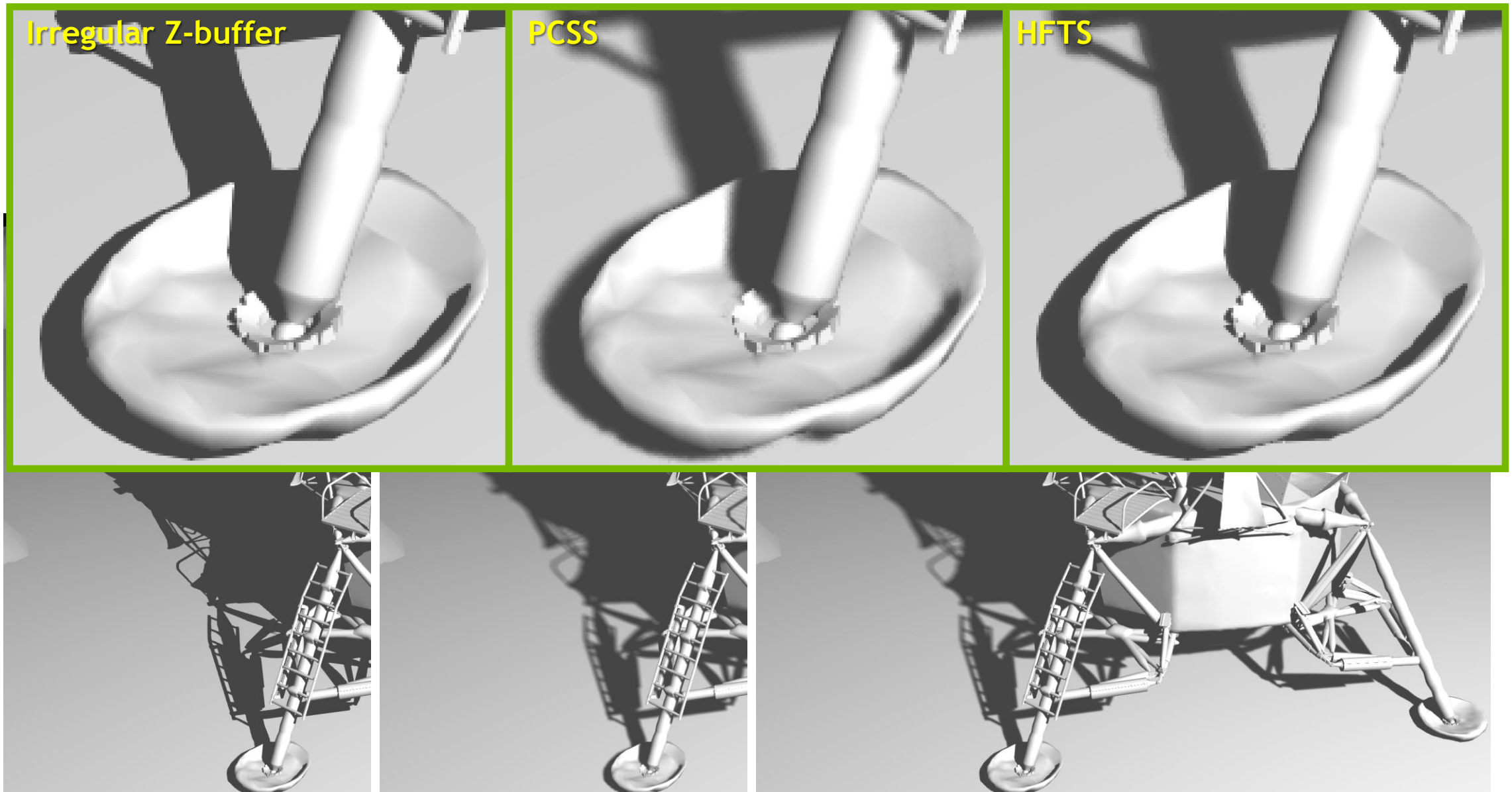
  (E.g., adaptive, perspective, logarithm, etc.)



HOW TO GET SOFT SHADOWS

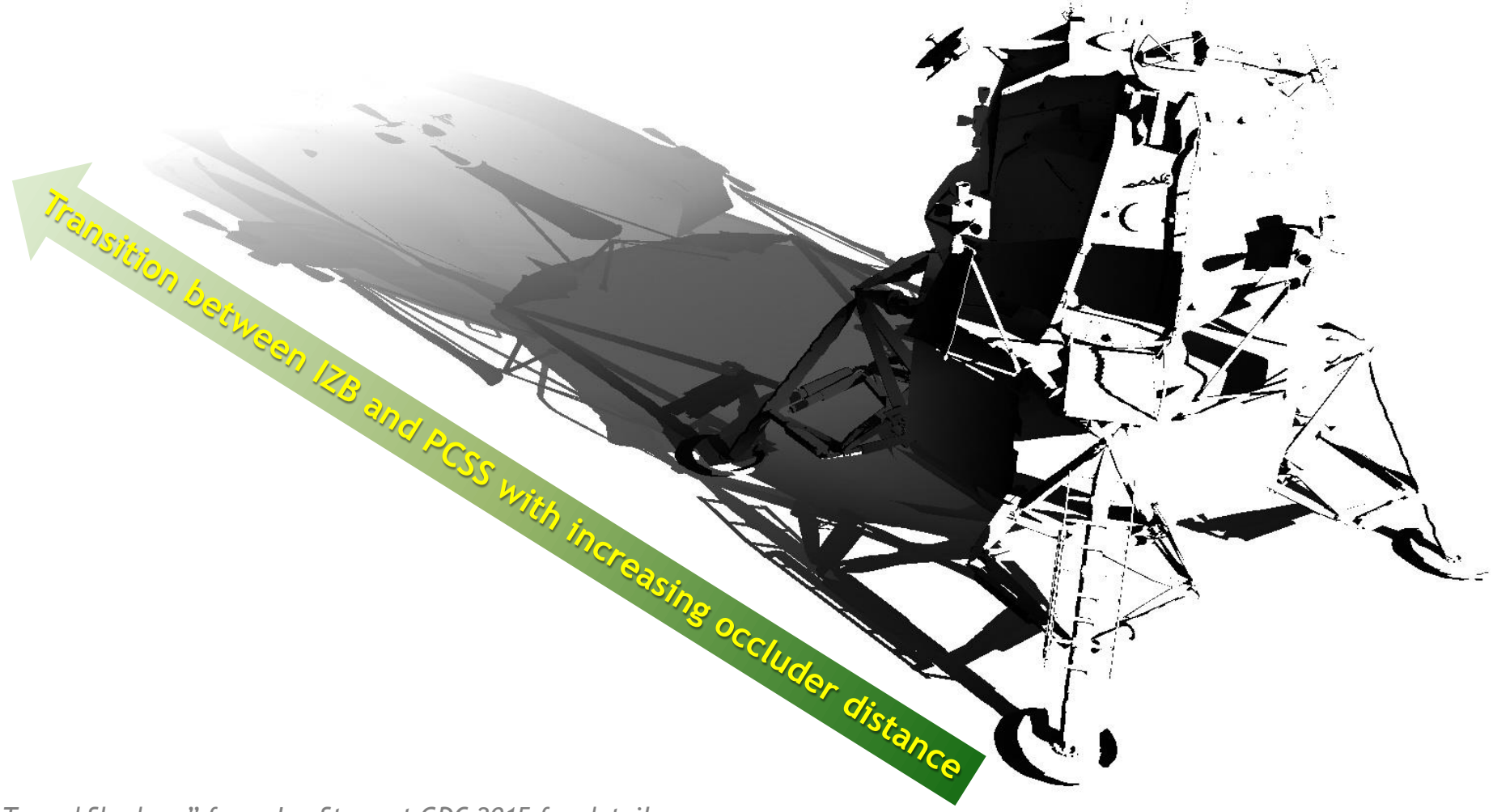
Unlike shadow maps, maintains high quality contact shadows when filtering




Irregular Z-buffer
PCSS
HFTS

HOW TO COMBINE?

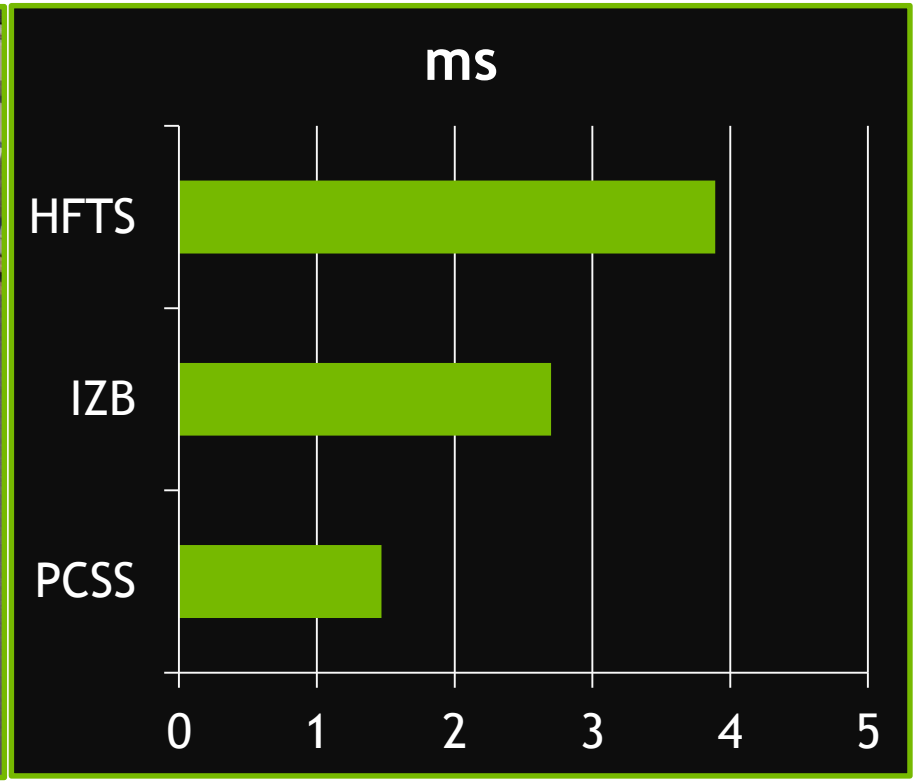
Multiple ways, but straightforward seems to work pretty well



See "Hybrid Ray Traced Shadows" from Jon Story at GDC 2015 for details

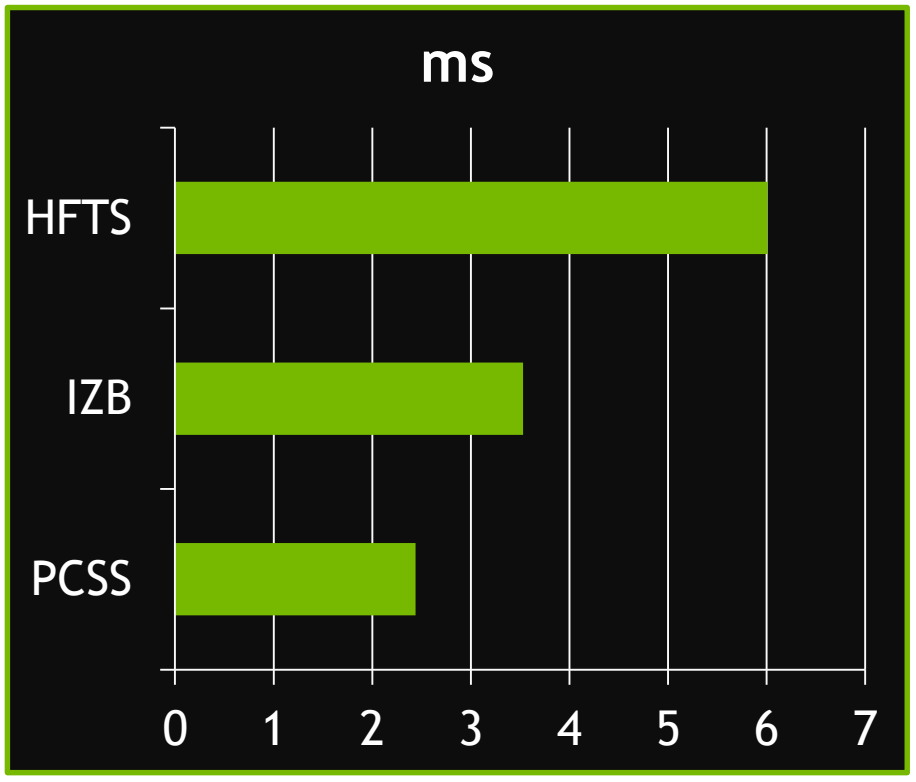
HFTS PERFORMANCE

GeForce GTX Titan X (2015) at Resolution: 1920x1080



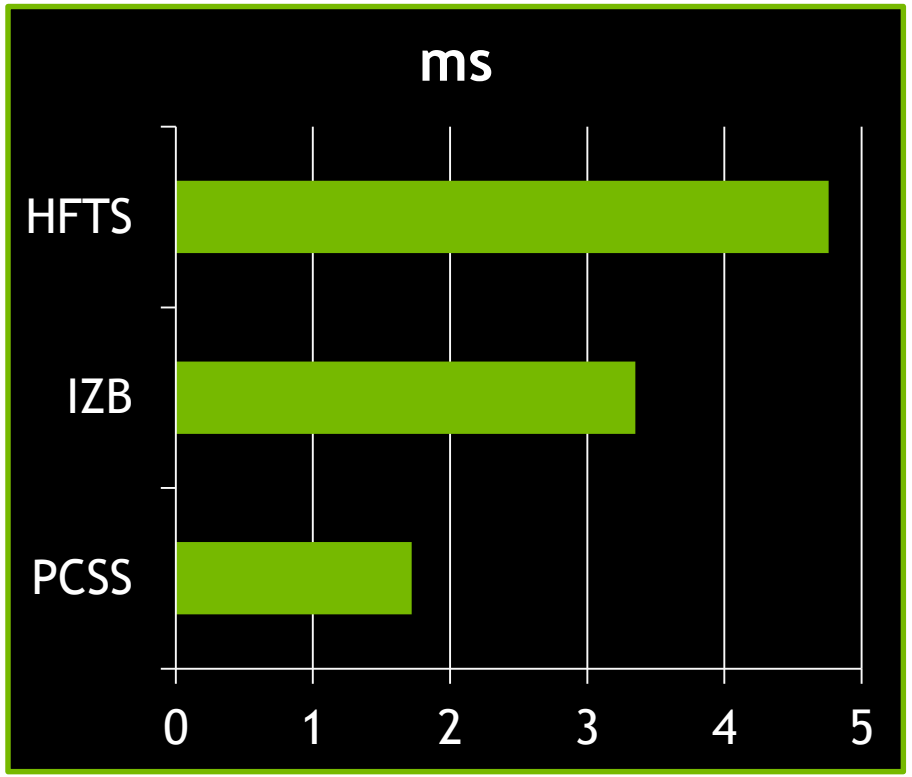
HFTS PERFORMANCE

GeForce GTX Titan X (2015) at Resolution: 1920x1080



HFTS PERFORMANCE

GeForce GTX Titan X (2015) at Resolution: 1920x1080



FURTHER TO GO

GameWorks version limited by in-game feasibility

- More advanced features available in budgets ~10-30 ms

Research prototype shows

- 32 samples per pixel ~2x cost of 1 sample
- Seemless shadows from transparent and alpha tested geometry
- Possibility of higher quality soft shadows

FURTHER TO GO

TAKEAWAYS

Can do fast ray traced shadows in games today

TAKEAWAYS

Can do fast ray traced shadows in games today

Eliminates shadow map aliasing larger than a pixel

TAKEAWAYS

Can do fast ray traced shadows in games today

Eliminates shadow map aliasing larger than a pixel

Introduces some variability to perf; cascades & other techniques reduce dramatically

TAKEAWAYS

Can do fast ray traced shadows in games today

Eliminates shadow map aliasing larger than a pixel

Introduces some variability to perf; cascades & other techniques reduce dramatically

Provides high quality input to shadow filters for approximate soft shadows

TAKEAWAYS

Can do fast ray traced shadows in games today

Eliminates shadow map aliasing larger than a pixel

Introduces some variability to perf; cascades & other techniques reduce dramatically

Provides high quality input to shadow filters for approximate soft shadows

HFTS available in NVIDIA GameWorks; shipped in Tom Clancy's The Division

QUESTIONS?

More on:
Hybrid Frustum Traced Shadows

More on:
Irregular Z-Buffers

Contact:

Chris Wyman
cwyman@nvidia.com
*@_cwyman_*

Jon Story
jons@nvidia.com

COME DO YOUR LIFE'S WORK
JOIN NVIDIA

We are looking for great people at all levels to help us accelerate the next wave of AI-driven computing in Research, Engineering, and Sales and Marketing.

Our work opens up new universes to explore, enables amazing creativity and discovery, and powers what were once science fiction inventions like artificial intelligence and autonomous cars.

Check out our career opportunities:

- www.nvidia.com/careers
- Reach out to your NVIDIA social network or NVIDIA recruiter at DeepLearningRecruiting@nvidia.com