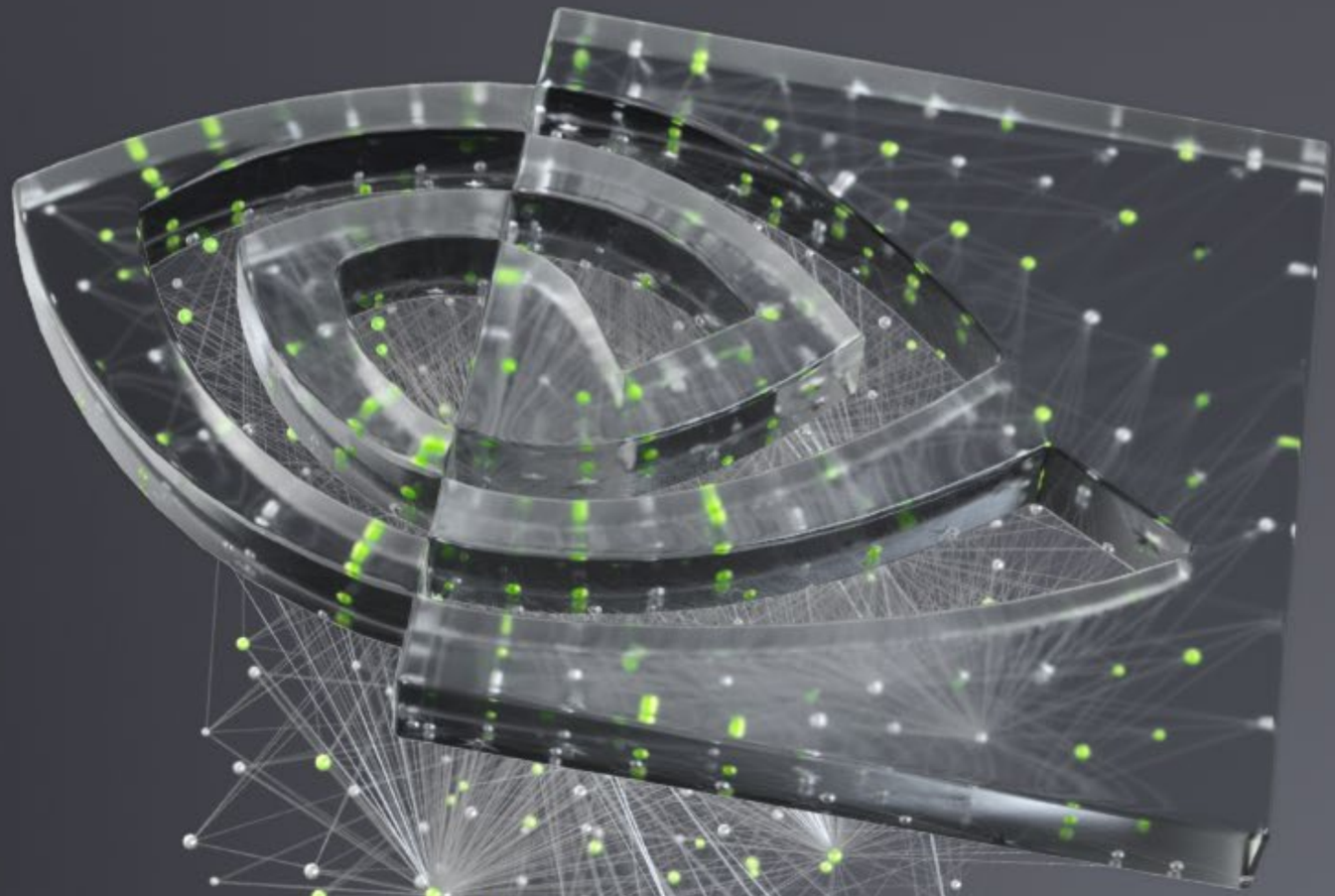
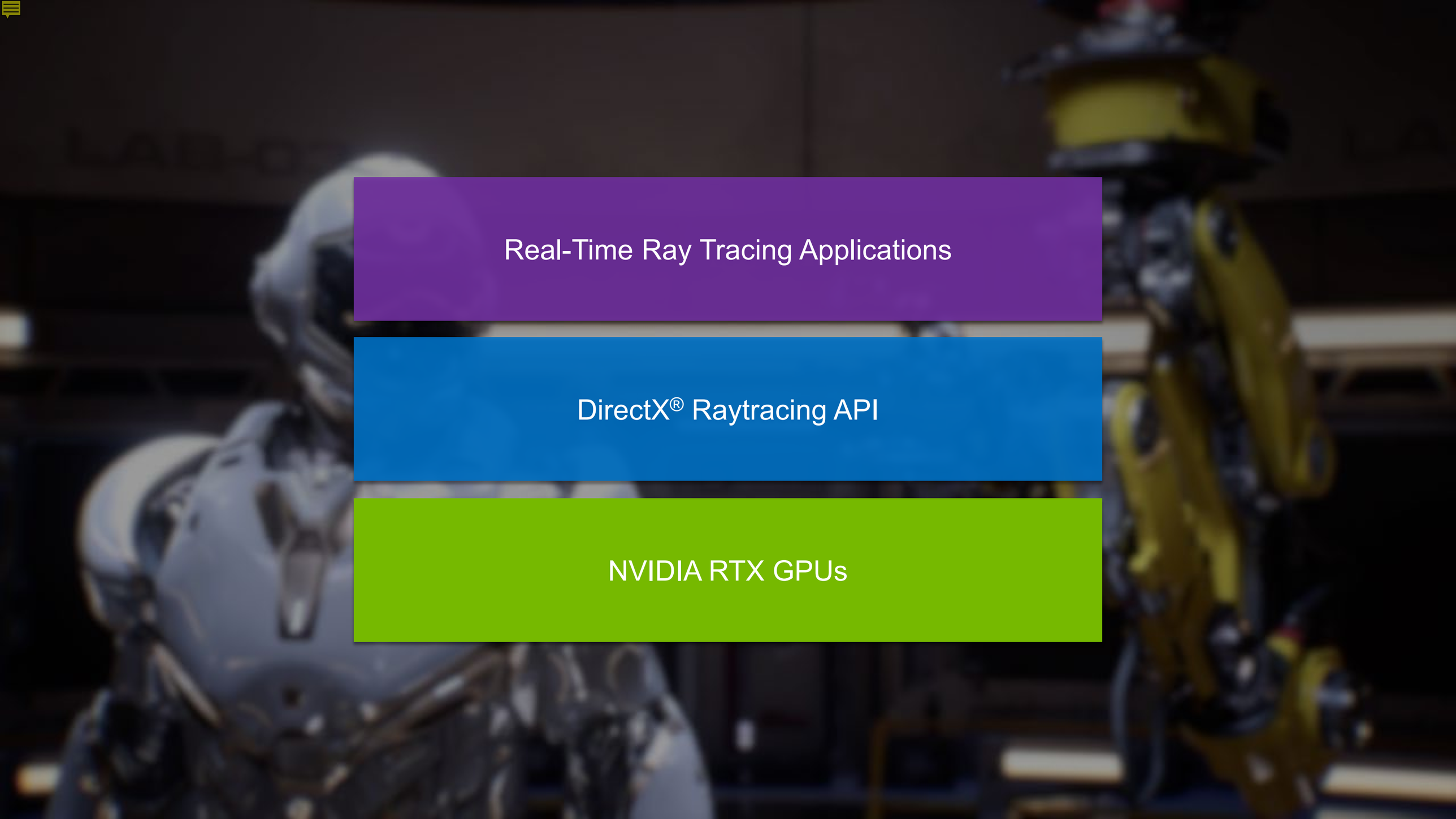


RTXGI:
SCALABLE RAY TRACED GLOBAL ILLUMINATION IN REAL TIME

Adam Marrs, 3/23/2020


Real-Time Ray Tracing Applications
DirectX® Raytracing API
NVIDIA RTX GPUs

REINVENTING REAL-TIME

GLOBAL ILLUMINATION, IN REAL TIME

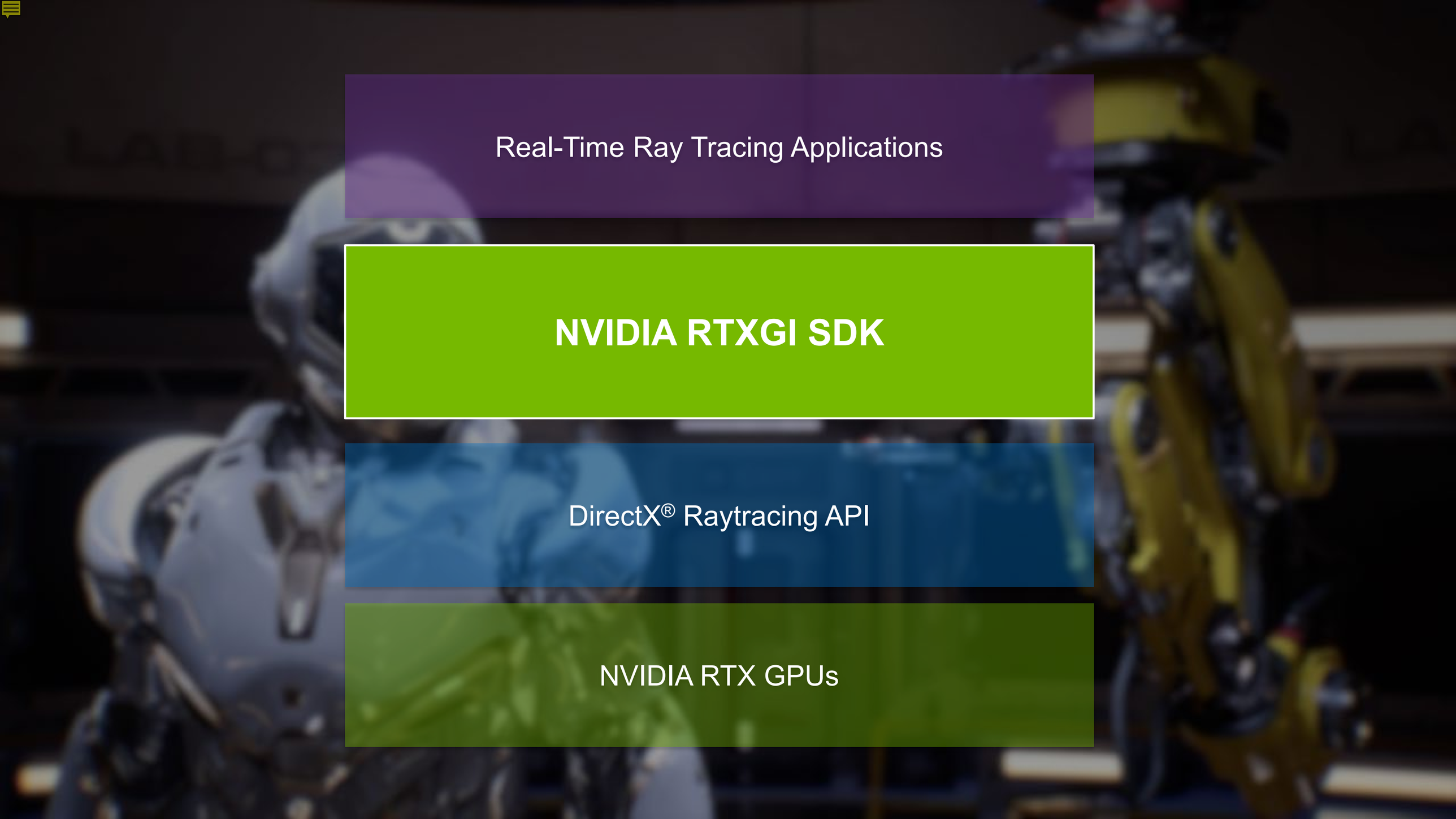
Light maps	[Quake97, Mitchell06]
Virtual point lights	[Keller97, Kaplanyan10, Ding14, Xu16, Luksch19]
Reflective shadow maps	[Kaplanyan10, Billeter12, Ding14, Malmros17, Xu16]
Light propagation volumes	[Kaplanyan09, Kaplanyan10, Boeckmann19]
Sparse voxel cone tracing	[Crassin11, McLaren16]
Denoised ray tracing	[Mara17, Schied17, Metro19, Archard19]
Irradiance probes/voxels	[Greger98, Ramamoorthi01, Tatarchuk05, Gilabert12, McGuire17, Majercik19]

GAME DEVELOPERS CONFERENCE

GDC

March 18–22, 2019
San Francisco, CA

https://www.gdcvault.com/play/1026182/


Real-Time Ray Tracing Applications
NVIDIA RTXGI SDK
DirectX® Raytracing API
NVIDIA RTX GPUs

RTX GLOBAL ILLUMINATION (RTXGI) SDK

High Level Goals

- Flexibility
  - Customizable, so you can tailor it to your specific needs
- Scalability
  - Effective solutions for a wide range of target hardware
- Convenience
  - Implement and optimize global lighting algorithms, so you don’t have to

RTX GLOBAL ILLUMINATION (RTXGI) SDK

Scalable Ray Traced Global Illumination in Real Time

- Full C++ and HLSL source code
- Sample application with full C++ and HLSL source
- Runs on *all* DXR enabled GPUs: NVIDIA Turing, NVIDIA Pascal, other vendors
- Scalable quality for GTX 1060 6GB through RTX 2080 Ti
- Available now, for free (on GitHub)

http://developer.nvidia.com/rtxgi

RTX GLOBAL ILLUMINATION (RTXGI) SDK

Dynamic Diffuse Global Illumination (DDGI)

- Based on irradiance probes
  - A common solution already used in many game engines today
- Fixes light and shadow leaking issues caused by lack of visibility information

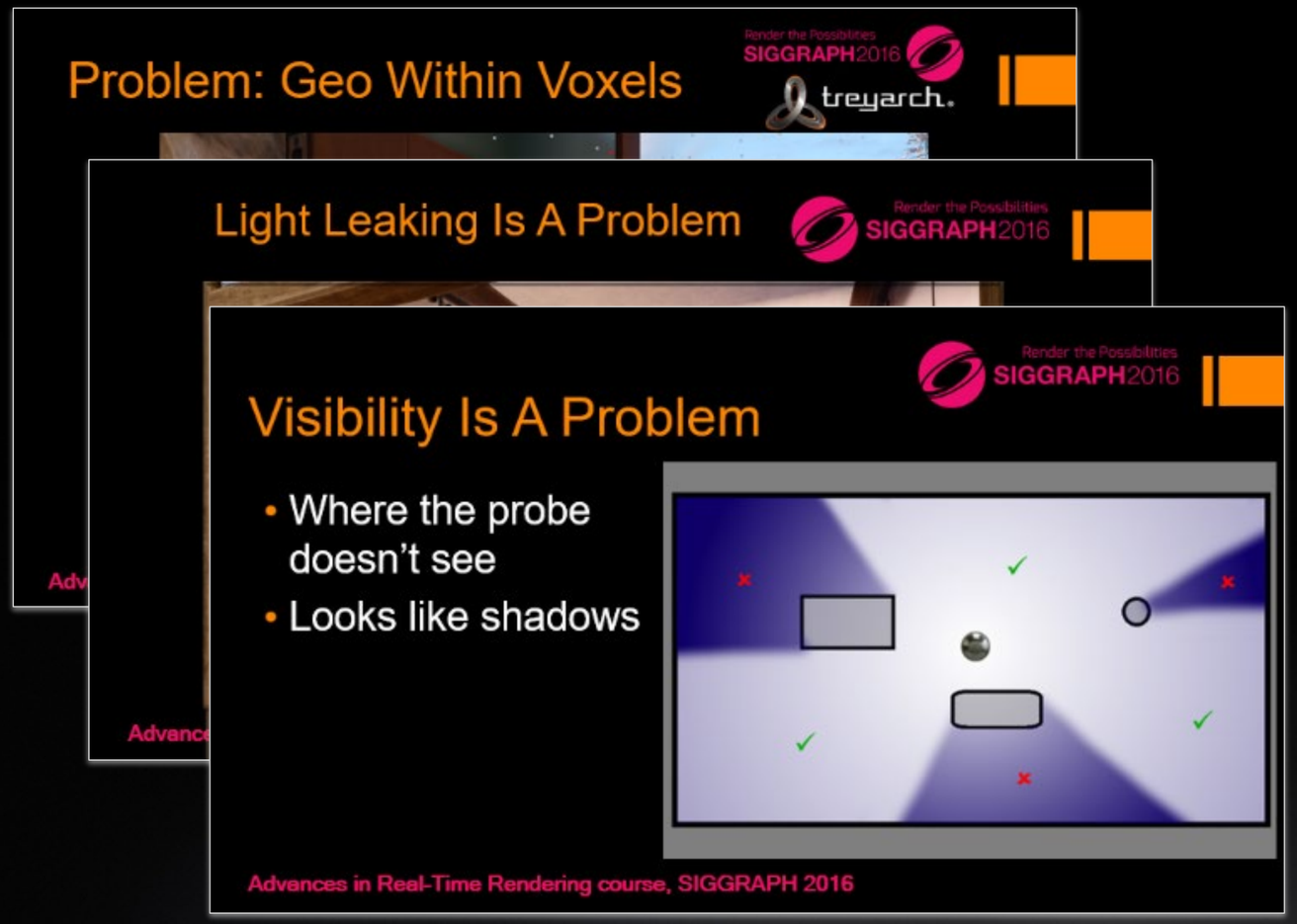

Unreal Engine


Dunia (Far Cry engine)


Unity

Image Credits: Epic Games, Ubisoft, and Unity. https://unity3d.com/learn/tutorials/topics/graphics/probe-lighting, https://docs.unrealengine.com/en-us/Engine/Rendering/LightingAndShadows/IndirectLightingCache, https://unity3d.com/learn/tutorials/topics/graphics/probe-lighting

LIGHT & SHADOW LEAKS

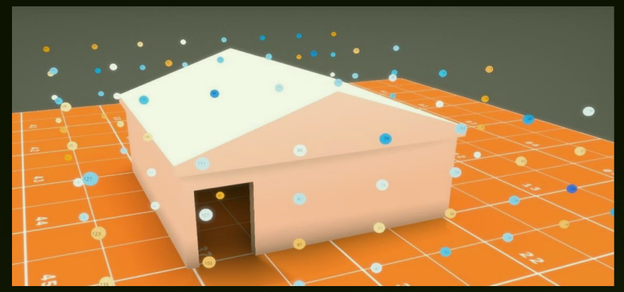


Hooker 2016

DYNAMIC DIFFUSE GLOBAL ILLUMINATION

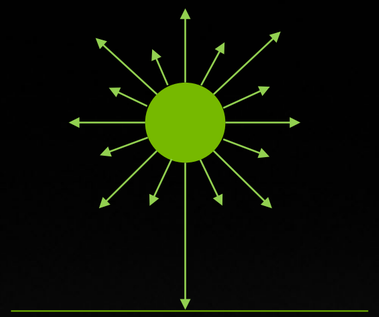
Algorithm

Scene Authoring



Place volumes of probes in the 3D world.

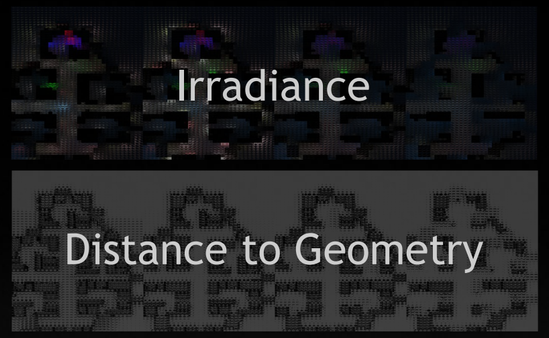
Ray Trace & Shade



Trace and shade rays cast from active probes in relevant volumes.

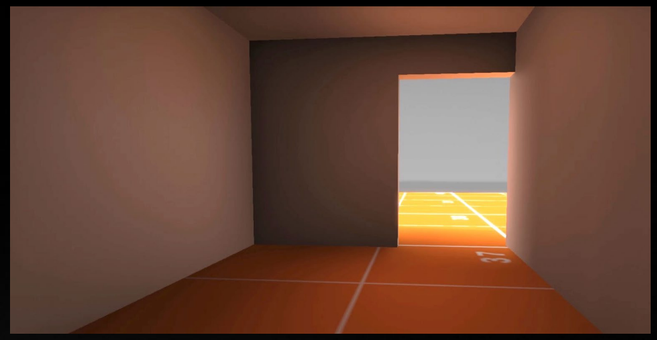
Use previous probe data during shading for infinite bounce GI.

Update Probes



Blend ray traced results into probes, storing irradiance and the distance to geometry.

Render Diffuse GI



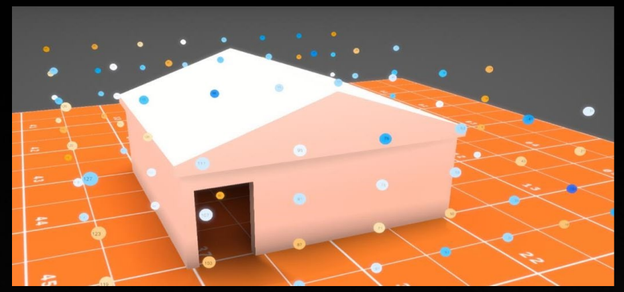
Compute indirect lighting and visibility from ray traced probes. No leaks.

Offline ← → Runtime

DYNAMIC DIFFUSE GLOBAL ILLUMINATION

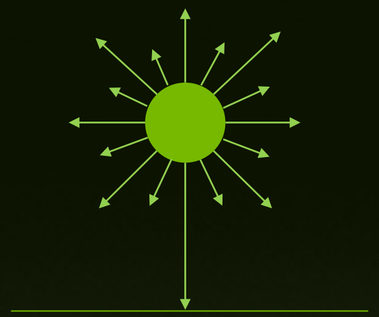
Algorithm

Scene Authoring



Place volumes of probes in the 3D world.

Ray Trace & Shade



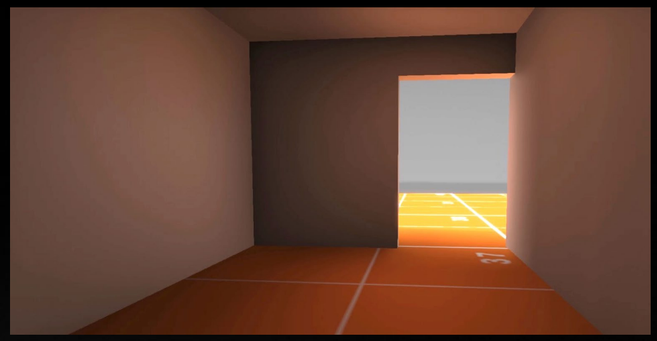
Trace and shade rays cast from active probes in relevant volumes.

Use previous probe data during shading for infinite bounce GI.

Update Probes

Blend ray traced results into probes, storing irradiance and the distance to geometry.

Render Diffuse GI



Compute indirect lighting and visibility from ray traced probes. No leaks.

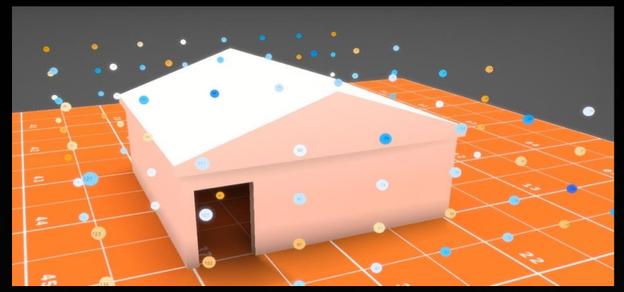
Offline

Runtime

DYNAMIC DIFFUSE GLOBAL ILLUMINATION

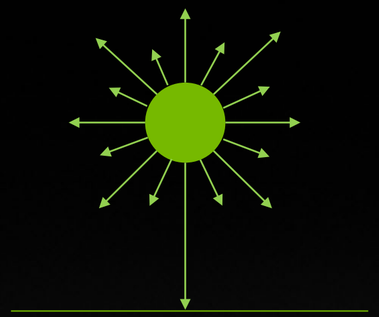
Algorithm

Scene Authoring



Place volumes of probes in the 3D world.

Ray Trace & Shade



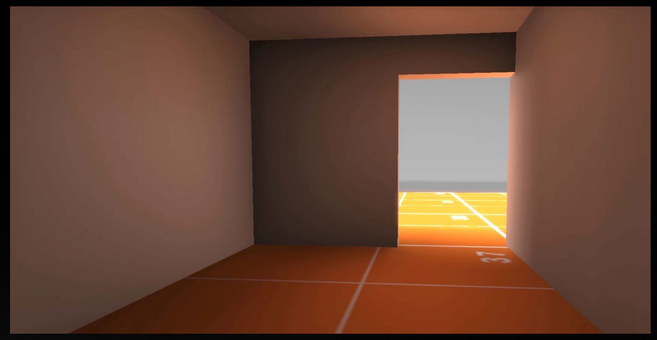
Trace and shade rays cast from active probes in relevant volumes.

Use previous probe data during shading for infinite bounce GI.

Update Probes

Blend ray traced results into probes, storing irradiance and the distance to geometry.

Render Diffuse GI



Compute indirect lighting and visibility from ray traced probes. No leaks.

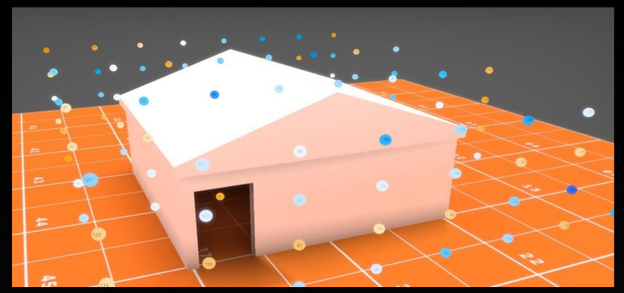
Offline

Runtime

DYNAMIC DIFFUSE GLOBAL ILLUMINATION

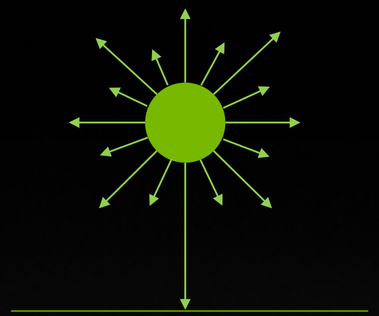
Algorithm

Scene Authoring



Place volumes of probes in the 3D world.

Ray Trace & Shade



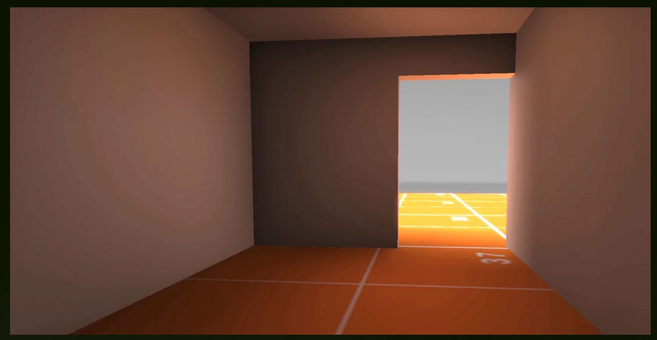
Trace and shade rays cast from active probes in relevant volumes.

Use previous probe data during shading for infinite bounce GI.

Update Probes

Blend ray traced results into probes, storing irradiance and the distance to geometry.

Render Diffuse GI



Compute indirect lighting and visibility from ray traced probes. No leaks.

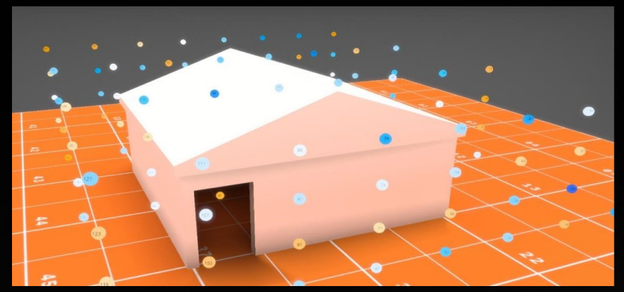
Offline

Runtime

DYNAMIC DIFFUSE GLOBAL ILLUMINATION

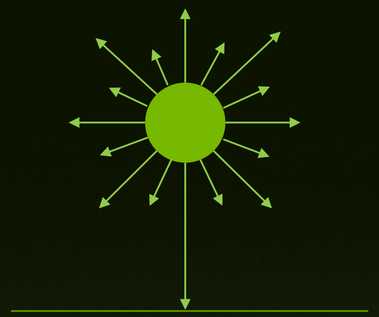
Algorithm

Scene Authoring



Place volumes of probes in the 3D world.

Ray Trace & Shade



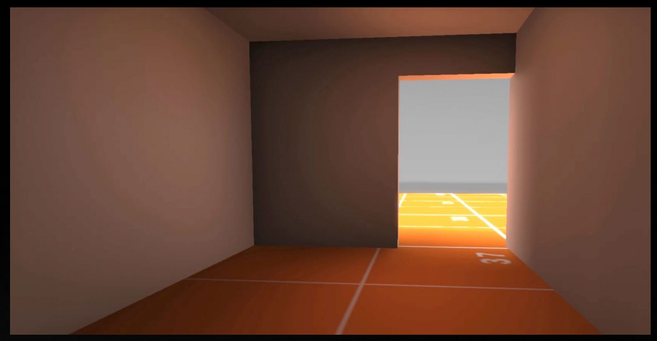
Trace and shade rays cast from active probes in relevant volumes.

Use previous probe data during shading for infinite bounce GI.

Update Probes

Blend ray traced results into probes, storing irradiance and the distance to geometry.

Render Diffuse GI



Compute indirect lighting and visibility from ray traced probes. No leaks.

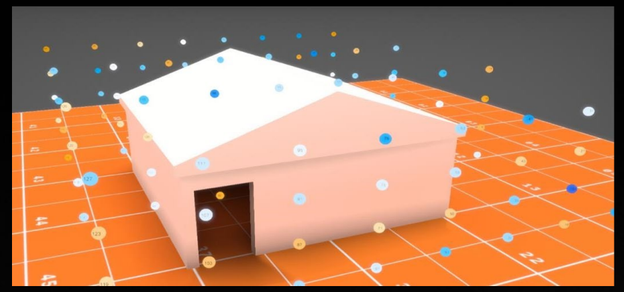
Offline

Runtime

DYNAMIC DIFFUSE GLOBAL ILLUMINATION

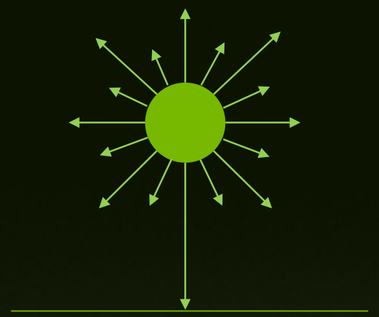
Algorithm

Scene Authoring



Place volumes of probes in the 3D world.

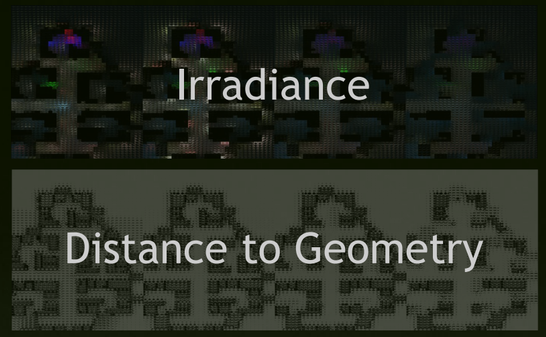
Ray Trace & Shade



Trace and shade rays cast from active probes in relevant volumes.

Use previous probe data during shading for infinite bounce GI.

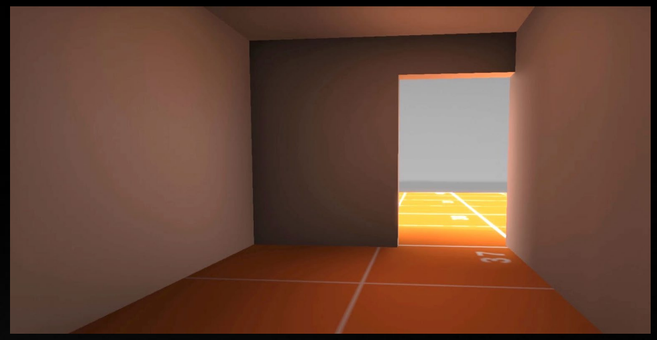
Update Probes



Blend ray traced results into probes, storing irradiance and the distance to geometry.

Independent of screen resolution *and* framerate

Render Diffuse GI



Compute indirect lighting and visibility from ray traced probes. No leaks.

Offline ← → Runtime

DDGI FEATURES & BENEFITS

Summary

- Infinite bounce indirect lighting, for forward or deferred renderers
- Ray traced quality, no denoising necessary
- Accelerated content creation
  - No baking, no leaks
  - No UV parameterization or probe blockers
  - In-game and in-editor lighting updates for fast iteration

RTXGI SDK: DDGI

Features and Improvements

- Flexible resource management
- Works with any material and lighting model
- Perceptual encoding to speed up convergence when large lighting changes occur
- Flexibility for programmers and artists to control performance and lighting quality
- Performance: fast probe updates using GPU shared memory
- Probe Relocation (early access)
- Probe State Classification (early access)

RTXGI SDK: DDGI

Flexible Resource Management

- Engines and renderers handle resources differently to solve different problems
- Providing a choice in how resources are managed
- SDK Managed Resources
  - SDK internally allocates/tracks/deallocates necessary GPU resources
  - Can’t manipulate resources directly, but black box design is easier to use
- Application Managed Resources
  - Application allocates/tracks/deallocates GPU resources, passes pointers into SDK
  - Better flexibility for advanced applications, but greater responsibility

RTXGI SDK: DDGI

Any Material or Lighting Model

- SDK does not force specific resource bindings or material properties
- Probe ray tracing step is owned and implemented by the application
- SDK provides utility functions, such as DDGIGetProbeRayDirection(…), to compute unique, low discrepancy, spherically distributed directions on the unit sphere
- Write ray traced results to the SDK’s radiance texture

RTXGI SDK: DDGI

Perceptual Encoding, Hysteresis, and Artist Controls

- Exponential weighting when storing irradiance to improve light-to-dark convergence
  - Moves irradiance into a non-linear space that more closely matches human perception
- Hysteresis settings give you control over convergence time
- Probe Change and Brightness Thresholds give programmers and artists control over convergence speed, lighting quality, and performance tradeoffs

RTXGI SDK: DDGI

Hypothetical Frame

- New passes for DDGI are shown in green
- In practice, DDGI Lighting can be combined with an existing lighting pass
- Timeline is for illustration purposes, does not represent performance (that's next)

RTXGI SDK: DDGI

Performance

Two Rooms, GPU time in milliseconds Stress Test Config | 16,384 Probes | 144 Rays Per Probe | 1920x1080				
	Probe RT	Probe Update	Lighting	Total
RTX 2080 Ti	1.05	1.22	0.44	2.71
RTX 2060	2.34	2.81	0.93	6.08
GTX 1080 Ti	8.53	2.11	1.00	11.64

RTXGI SDK: DDGI

Performance

Two Rooms, GPU time in milliseconds Stress Test Config | 16,384 Probes | 144 Rays Per Probe | 1920x1080				
	Probe RT	Probe Update	Lighting	Total
RTX 2080 Ti	1.05	1.22	0.44	2.71
RTX 2060	2.34	2.81	0.93	6.08
GTX 1080 Ti	8.53	2.11	1.00	11.64

RTXGI SDK: DDGI

Performance

Two Rooms, GPU time in milliseconds Stress Test Config | 16,384 Probes | 144 Rays Per Probe | 1920x1080				
	Probe RT	Probe Update	Lighting	Total
RTX 2080 Ti	1.05	1.22	0.44	2.71
RTX 2060	2.34	2.81	0.93	6.08
GTX 1080 Ti	8.53	2.11	1.00	11.64

RTXGI: SDK DDGI

Fast Probe Updates

Two Rooms, GPU time in milliseconds Stress Test Config | 16,384 Probes | 144 Rays Per Probe			
	RTX 2080 Ti	RTX 2060	GTX 1080 Ti
Default Implementation	3.57	7.83	5.32
RTXGI SDK Implementation with Shared Memory Optimizations	1.22	2.81	2.11
Speedup	**2.92x**	**2.79x**	**2.52x**

RTXGI SDK: DDGI

Performance

Two Rooms, GPU time in milliseconds Stress Test Config | 16,384 Probes | 144 Rays Per Probe | 1920x1080				
	Probe RT	Probe Update	Lighting	Total
RTX 2080 Ti	1.05	1.22	0.44	2.71
RTX 2060	2.34	2.81	0.93	6.08
GTX 1080 Ti	8.53	2.11	1.00	11.64

RTXGI SDK: DDGI

Performance

Two Rooms, GPU time in milliseconds Stress Test Config | 16,384 Probes | 144 Rays Per Probe | 1920x1080				
	Probe RT	Probe Update	Lighting	Total
RTX 2080 Ti	1.05	1.22	0.44	2.71
RTX 2060	2.34	2.81	0.93	6.08
GTX 1080 Ti	8.53	2.11	1.00	11.64

RTXGI SDK: DDGI

Probe Relocation (early access)

- Any regular grid of probes has a hard time correctly handling all scenarios
- Probe Relocation
  - Maintains world-space offsets for every probe
  - Uses the ray tracing results to determine proximity of back facing geometry
  - Attempts to move probes to more effective locations based on surrounding geometry
- Sample application includes a linear descent optimizer that relocates probes over several iterations (frames)
- Early access feature, expect it to be updated and improved in future releases

RTXGI SDK: DDGI

Probe Classification (early access)

- Not all probes in a scene contribute to the final lighting
  - For example, probe stuck in walls or too far outside the scene to be useful
  - These probes don’t need to spend time ray tracing or updating textures and can be disabled
- Probe Classification
  - Maintains a state value (active, inactive) for every probe
  - Marks probes as active or inactive based on the results of ray tracing
- As much as 30% to 80% of the probes can be disabled in many scenarios
- Early access feature, expect it to be updated and improved in future releases

Download the RTXGI SDK today!

http://developer.nvidia.com/rtxgi

Request Access

Go to the link above and click “Get Started”. You’ll need to fill out a short survey to request access.

Clone the GitHub Repo

After filling out the survey and receiving an approval email, you can access the full source distribution on GitHub.

Dive in with the Sample Application

A working sample application is included as an example integration that demonstrates the SDK’s functionality.

THANK YOU

Alexander Majercik

Josef Spjut

Morgan McGuire

Alan Wolfe

Ben Boudaoud

Kelsey Blanton

Alex Hyder

Ethan Einhorn

John Spitzer

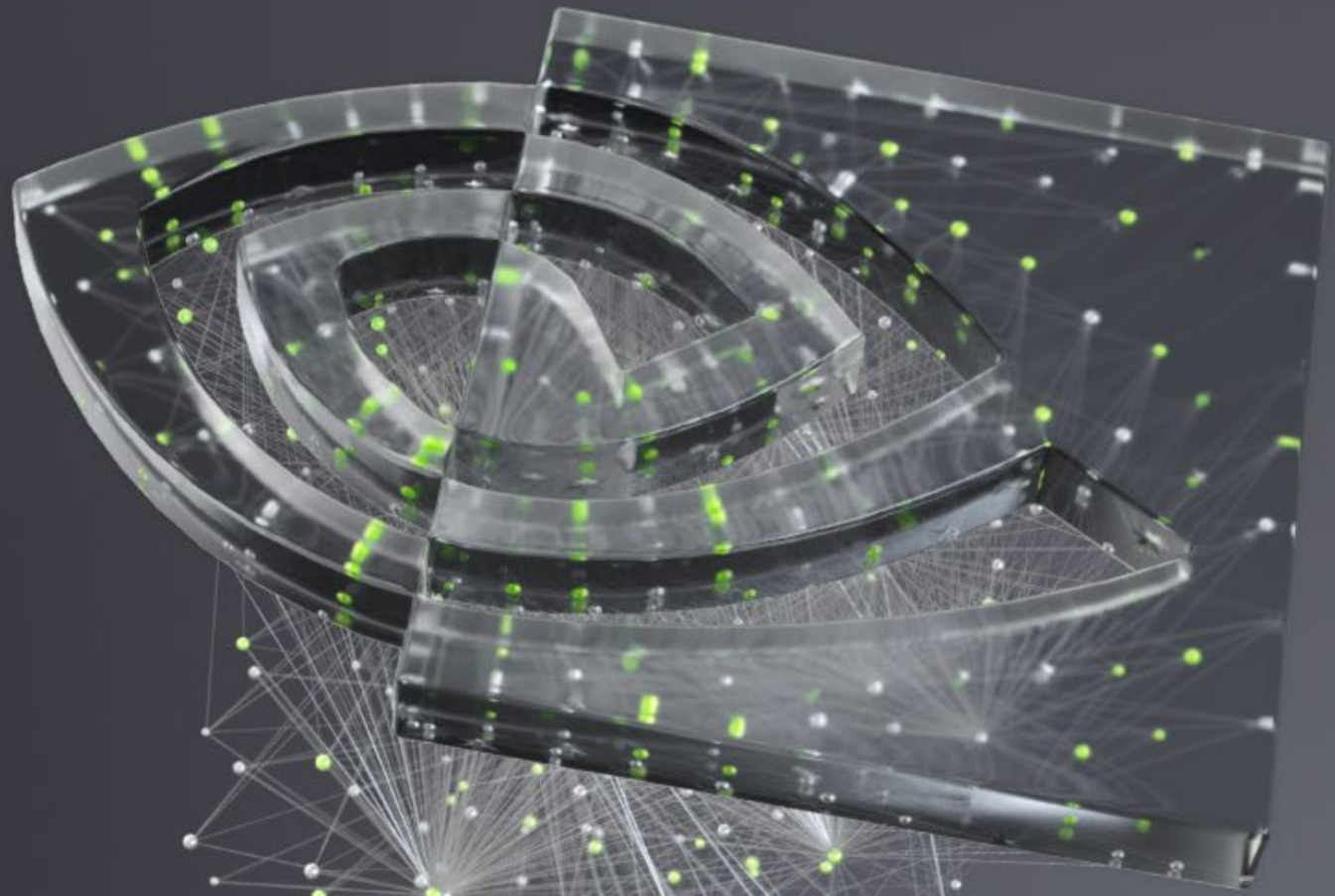
QUESTIONS?

@acmarrs

Issues

REFERENCES

[Archard19] Archard, Zhdan, Shyshkovtsov, and Karmalsky, Exploring Raytraced Future in Metro Exodus, GDC Talk, 2019
[Billeter12] Billeter, Sintorn, Assarsson. Real-time multiple scattering using light propagation volumes. I3D 2012.
[Crassin11] Crassin, Neyret, Sainz, Green, and Eisemann, Interactive Indirect Illumination Using Voxel Cone Tracing, Pacific Graphics, 2011
[Dachsbacher05] Dachsbacher and Stamminger, Reflective Shadow Maps, I3D, 2005
[Ding14] Ding, In-Game and Cinematic Lighting of The Last of Us, GDC Presentation, 2014
[Gilabert12] Gilabert and Stefanov, Deferred Radiance Transfer Volumes: Global Illumination in Far Cry 3, Talk at GDC, 2012
[Greger98] Greger et al., The Irradiance Volume, IEEE CG&A 1998
[Hooker16] Hooker, Volumetric Global Illumination at Treyarch, SIGGRAPH Advances in Real-Time Rendering, 2016
[Kaplanyan09] Kaplanyan, Light Propagation Volumes in CryEngine 3, SIGGRAPH Advances in Real-Time Rendering Course, 2009
[Kaplanyan10] Kaplanyan, Real-time Diffuse Global Illumination in CryENGINE 3, SIGGRAPH Presentation, 2010
[Keller97] Keller, Instant radiosity, SIGGRAPH, 1997
[Luksch19] Luksch et al., Incrementally baked Global Illumination, I3D 2019
[Majercik19] Majercik et al., Dynamic Diffuse Global Illumination with Ray Traced Irradiance Fields, JCGT 2019
[Malmros17] Malmros, Gears of War 4: custom high-end graphics features and performance techniques, SIGGRAPH Talk, 2017
[Mara17] Mara et al., An Efficient Denoising Algorithm for Global Illumination, HPG 2017
[McGuire17] McGuire, Mara, Nowrouzezahari, Luebke. Real-Time Global Illumination using Precomputed LightField Probes. 2017
[Metro19] Battaglia, Interview with Ben Archard on Metro: Exodus, Eurogamer, Feb 17, 2019
[McLaren16] McLaren, Graphics Deep Dive: Cascaded voxel cone tracing in The Tomorrow Children, Gamasutra, November 28, 2016
[Mitchell06] Mitchell, McTaggart, and Green, Shading in Valve's Source Engine, SIGGRAPH Advances in Real-Time Rendering Course, 2006
[Quake97] ID Software, Quake II, 1997
[Ramamoorthi01] Ramamoorthi and Hanrahan, An Efficient Representation for Irradiance Environment Maps, SIGGRAPH, 2001
[Schied17] Schied et al., Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path Traced Global Illumination, HPG 2017
[Tatarchuk05] Tatarchuk, Irradiance Volumes for Games, GDC 2005
[Xu16] Xu, Temporal Antialiasing In Uncharted 4, SIGGRAPH Advances in Real-Time Rendering Course, 2016



RTXGI:
SCALABLE RAY TRACED GLOBAL ILLUMINATION IN REAL TIME

Adam Marrs, 3/23/2020