Unreal Fest 2024 Seoul

UE5 Graphics Features Insights from Japan

Noriaki Shinoyama
Developer Relations, Lead
Epic Games Japan

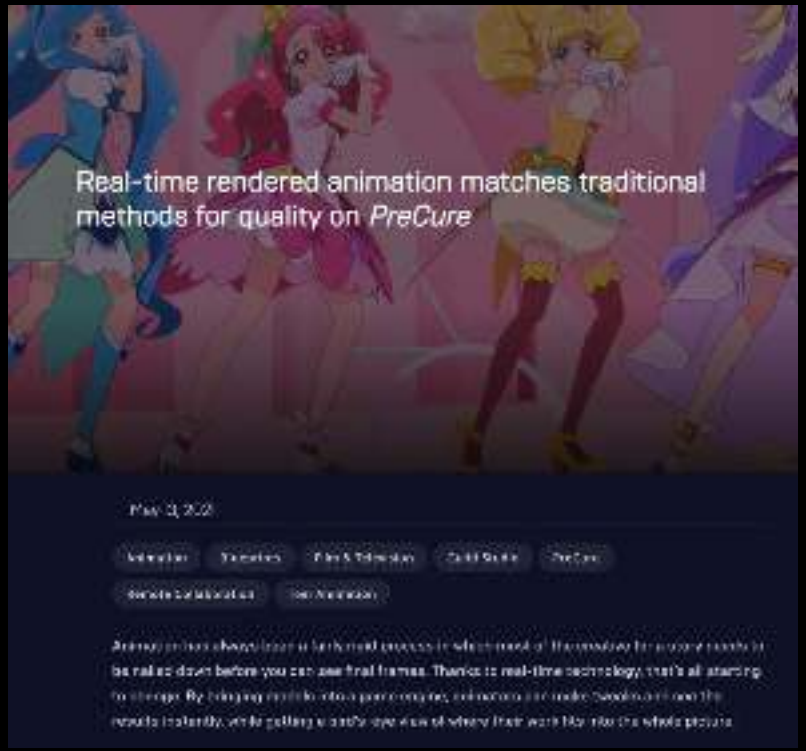

LIKE A DRAGON:
ISHIN!


PIKMIN4


P3
PERSONA3
RELOAD


HI-FI RUSH

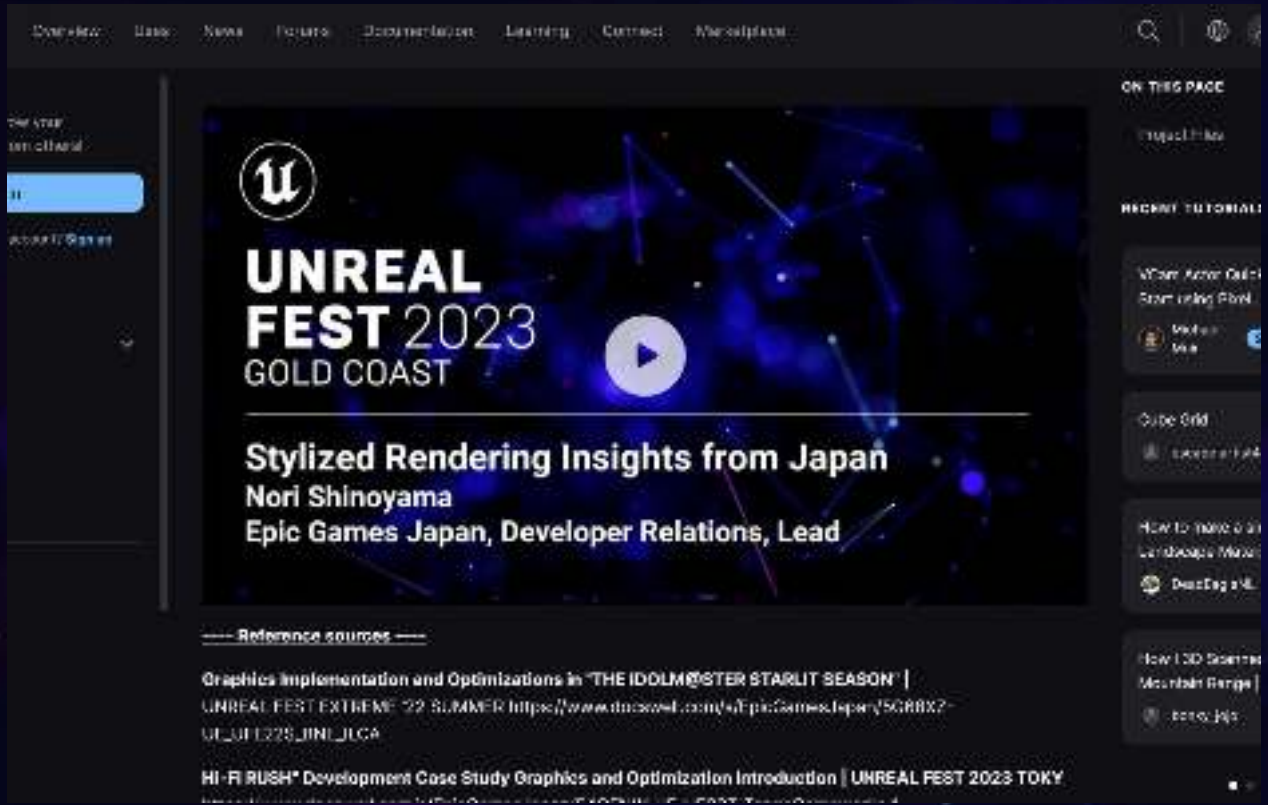
Anime



[Real-time rendered animation matches traditional methods for quality on PreCure]

Comic / Manga

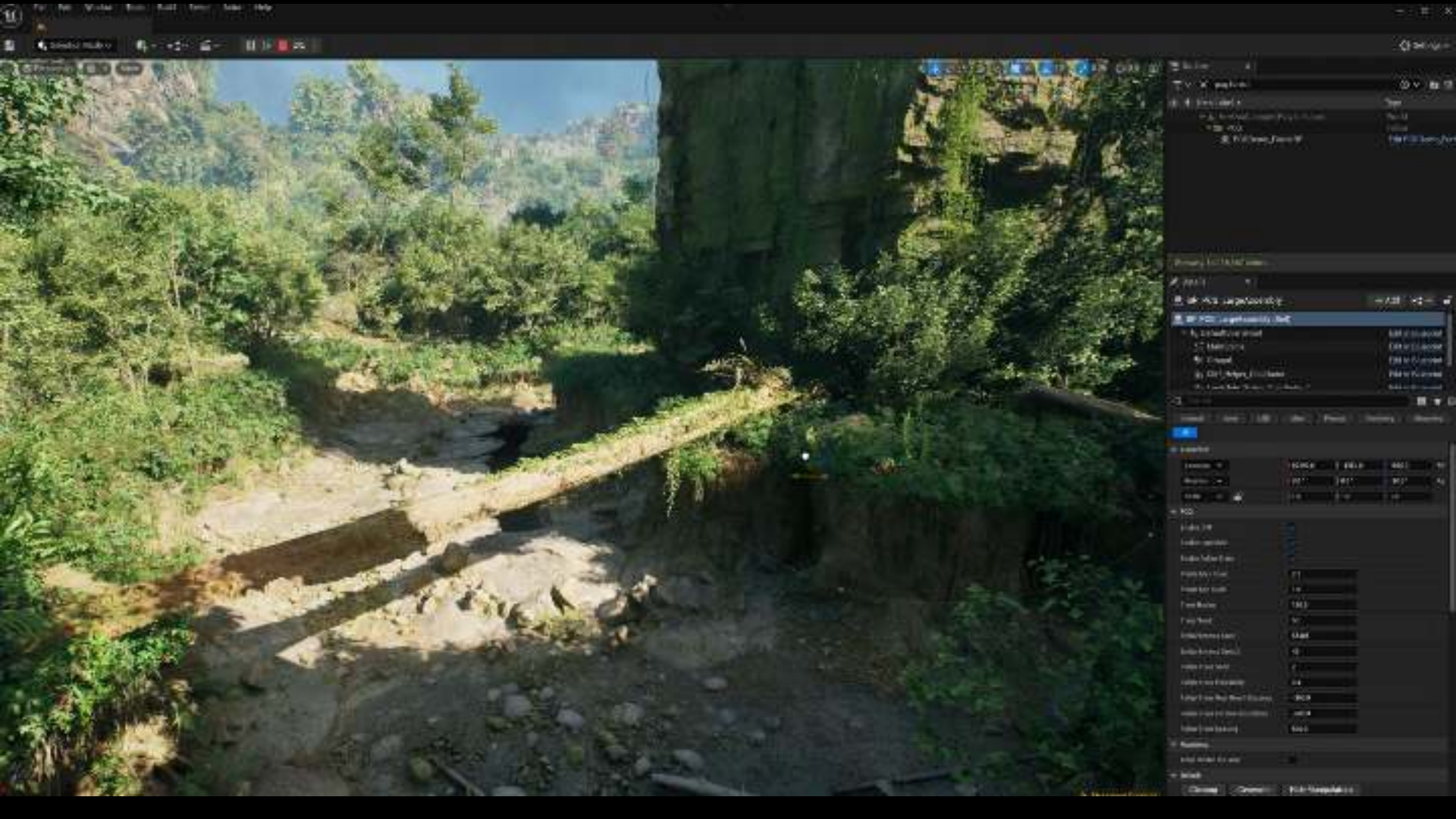
週刊少年ジャンプの読切作品『キルアオハル』はUnreal Engineを漫画制作にどのように活かしたか


UNREAL
FEST 2023
GOLD COAST
Stylized Rendering Insights from Japan
Nori Shinoyama
Epic Games Japan, Developer Relations, Lead
ON THIS PAGE
Reference sources
Graphics Implementation and Optimizations in "THE IDOLM@STER STARLIT SEASON" |
HI-FI RUSH" Development Case Study Graphics and Optimization Introduction | UNREAL FEST 2023 TOKY

Unreal Engine is used in various fields in Japan!

- Unique point
  - Market in Japan is dominated by Console rather than PC or Mobile.
    Strict optimization is required in many places
  - Very large number of Custom Licensees

What have DevRels in Japan faced since UE5 was released?



Introduction

Growing Expectations for Unreal Engine 5

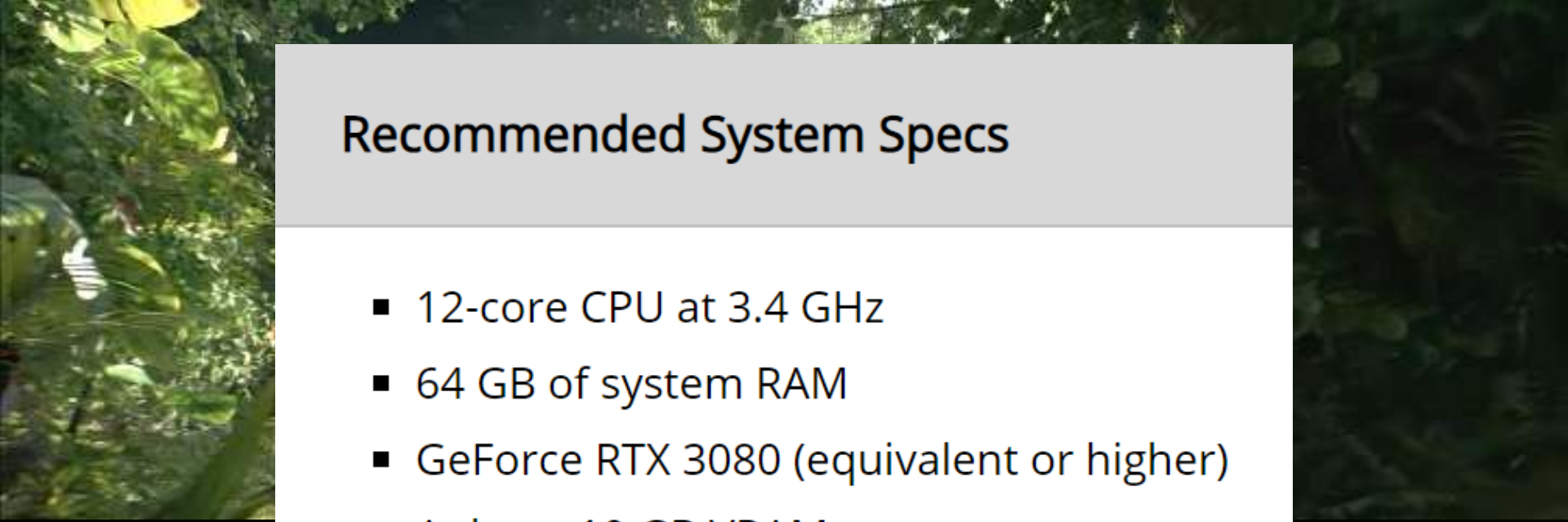
- **Infinite polygons** can be displayed in real time!
- **Realistic lighting** can be achieved by simply placing lights!
- You can use a variety of high-quality assets from **Megascans** for free!
- There are also mechanisms for **procedural workflows** and **open worlds!**

On the other hand,
some opinions like…

No grass

No leaves

Oh, grass and leaves!



Recommended System Specs

- 12-core CPU at 3.4 GHz
- 64 GB of system RAM
- GeForce RTX 3080 (equivalent or higher)
- At least 10 GB VRAM

Oh, grass and leaves!

Character reflection is black

can NOT enter
the buildings

What have DevRels in Japan faced since UE5 was released?

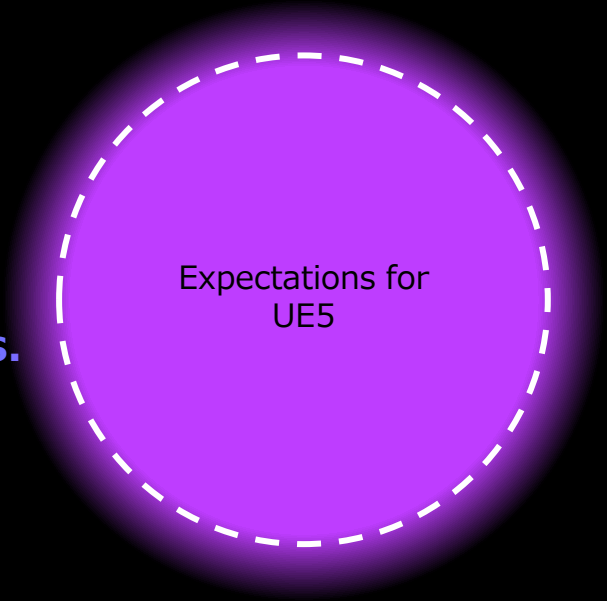
Vague expectations for UE5

Agenda

- Introduction
1. Multi-frame processing
2. Virtualization and Streaming
3. Utilization of Existing Assets
- Summary

Disclaimer

In this presentation,
there will be a lot of talk about
“UE doesn’t support it" and "this is difficult"



But, **I don't mean to imply that UE5 is useless.**
I just want to talk about the current limitations
so that you can understand and take advantage
of what UE5 is good at!

UE5 evolves at a tremendous pace!
Most of the current limitations may be eliminated
near future.

Introduction

Who I am?

Nori Shinoyama

Epic Games Japan,
Developer Relations, Lead
nori.shinoyama@epicgames.com

International Sake Sommelier

X(Twitter): @tempkinder

Agenda

- Introduction
1. Multi-frame processing
2. Virtualization and Streaming
3. Utilization of Existing Assets
- Summary

Multi-frame processing

Temporal / Progressive processing

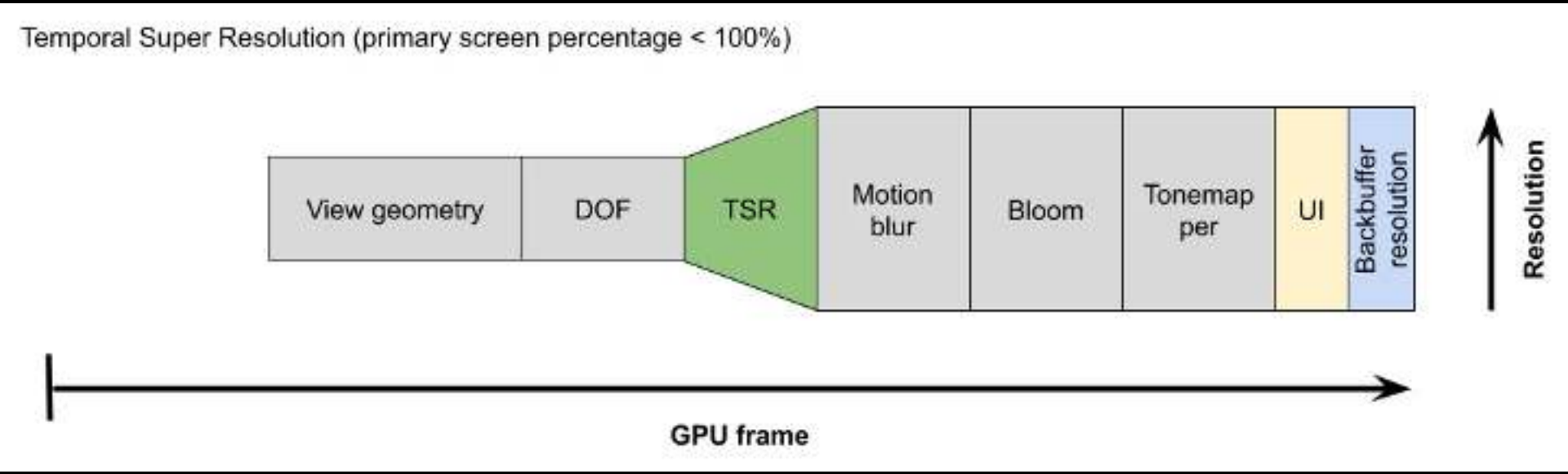
4K or 8K?
High resolution always required


LB Speed Down
RB Speed Up
LT Altitude Down
RT Altitude Up
Menu
Y Walk
X Dismiss Controls
KORNER CUP
OCEAN AND FIELD
THEATER


LB Speed Down
RB Speed Up
LT Altitude Down
RT Altitude Up
Menu
Y Walk
X Dismiss Controls
KORNER CUP
OCEAN AND FIELD
Frame: 57.50 ms
Game: 14.54 ms
Draw: 18.36 ms
GPU: 57.33 ms
DynRes: 100.0% x 100.0%
Draws: 5527
Prims: 414.3K

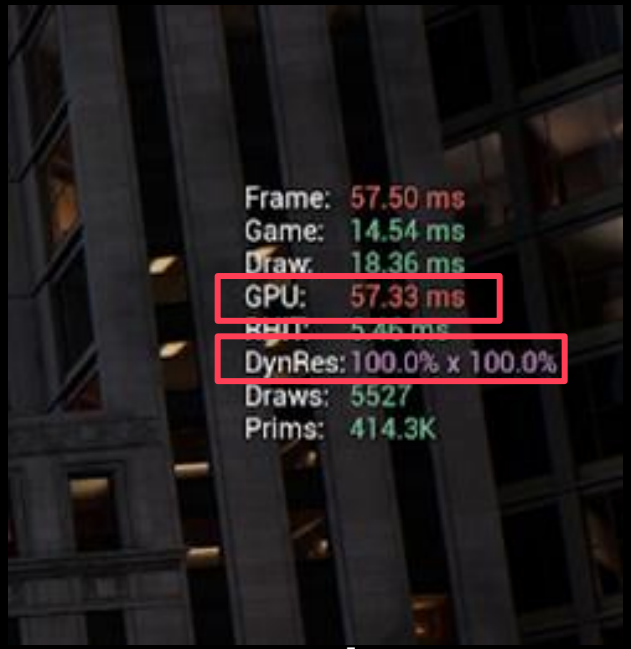
Multi-frame processing

Temporal Super Resolution (TSR)

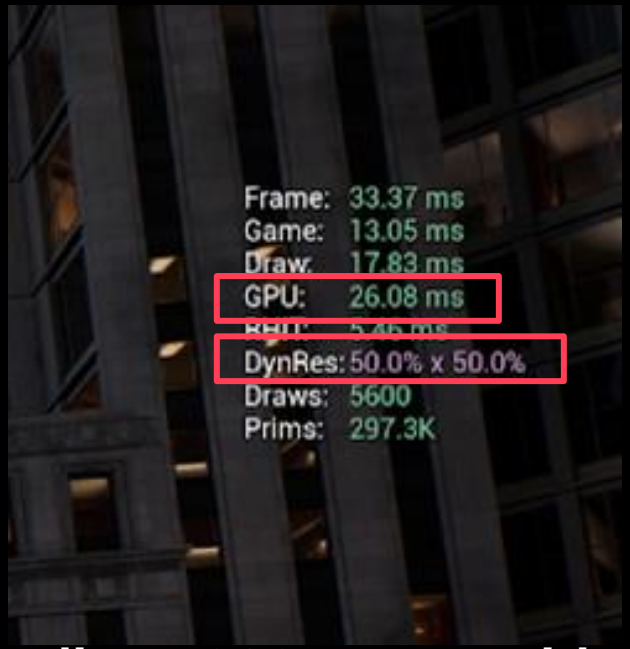



LB Speed Down
RB Speed Up
LT Altitude Down
RT Altitude Up
Menu
Y Walk
X Dismiss Controls
KORNER CUP
OCEAN AND FIELD
THEATER


LB Speed Down
RB Speed Up
LT Altitude Down
RT Altitude Up
Menu
Y Walk
X Dismiss Controls
KORNER CUP
OCEAN AND FIELD
THEATER



4K native
57.33ms

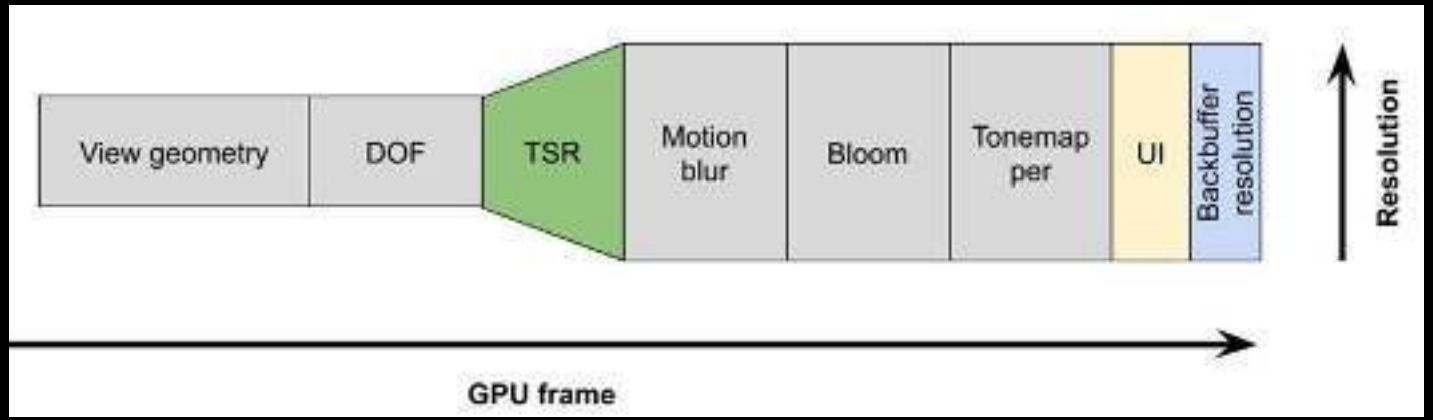


Upscaling 1080p to 4K with TSR
26.08ms

Multi-frame processing

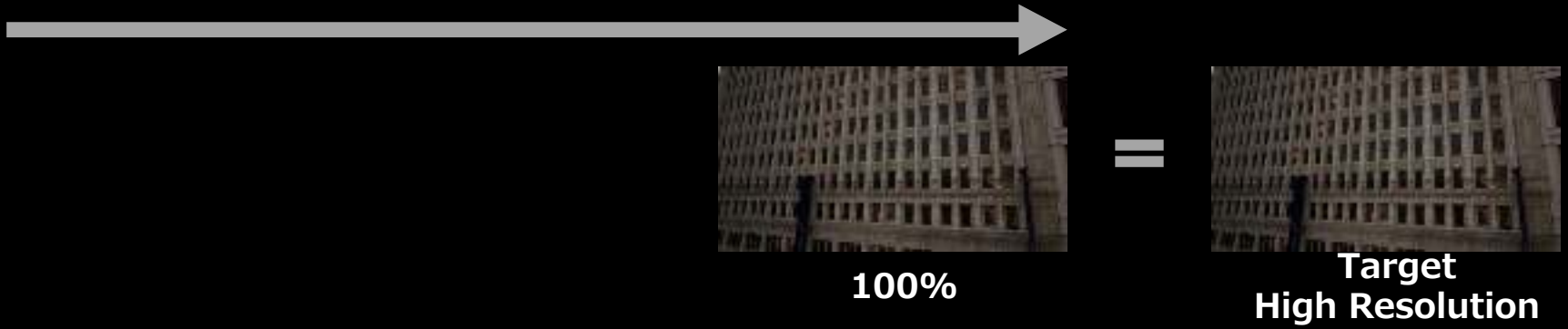
Temporal Super Resolution (TSR)

- A method of upscaling
  by **referencing multiple previous low-resolution frames**
- Reduces the total amount of pixels computed in a single frame,
  thus **dramatically lowering the processing load on many**,
  including Nanite/Lumen



Note on TSR

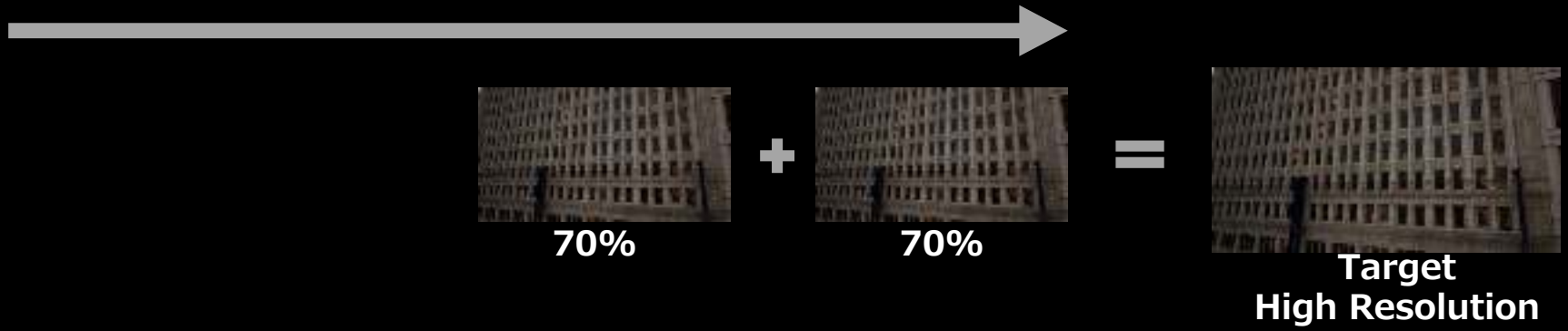
Need to **collect enough samples** to reach (converge) to the target high resolution



If you render at 100% target resolution, no need to upscale
Just output the 100% image as is.

Note on TSR

Need to **collect enough samples** to reach (converge) to the target high resolution



The lower the internal resolution, the more frames are required

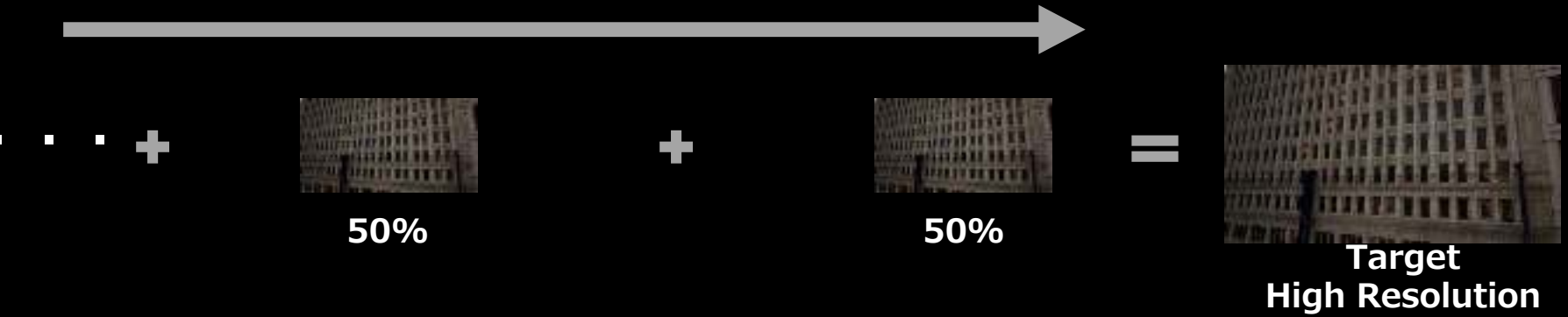
Note on TSR

Need to **collect enough samples** to reach (converge) to the target high resolution

The lower the internal resolution, the more frames are required

Note on TSR

Need to **collect enough samples** to reach (converge) to the target high resolution

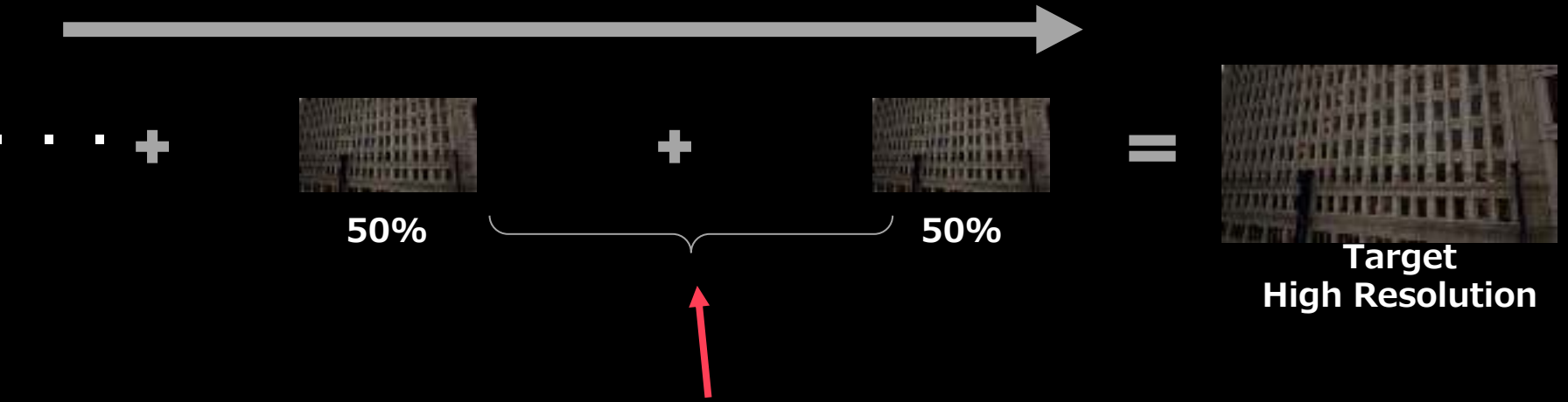


Convergence time also doubles when going from 60fps to 30fps

When the internal resolution is too low or the frame rate is too slow, the screen can get blurred when the camera moves.

Note on TSR

Need to **collect enough samples** to reach (converge) to the target high resolution

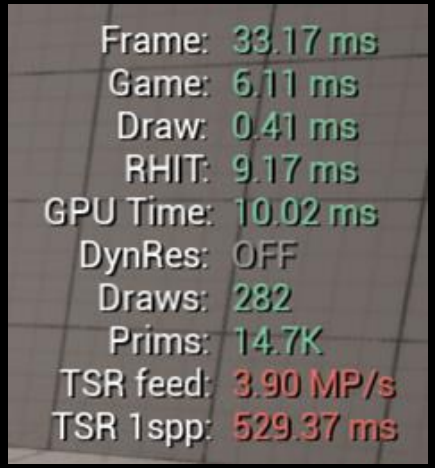


Even if low fps occurs in a process not related to rendering, the image may be blurred because of the time it takes for the TSR to converge.

TSR convergence time

- “stat tsr” command
  - use with “stat unit”
  - Good indicator for TSR flicker, delay, etc.

- The official UE5 documentation describes in detail the time to convergence under various conditions.

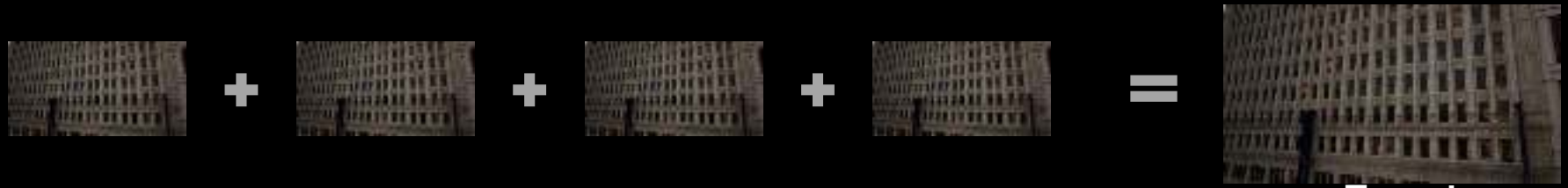


Screen Percentage	Framerate	TSR Convergence Rate
50%	60hz	1000 / ((50/100)^2 * 60) = 66.6 ms
58%	60hz	49.5 ms
66%	60hz	38.2 ms
100%	60hz	16.6 ms
50%	30hz	133.3 ms

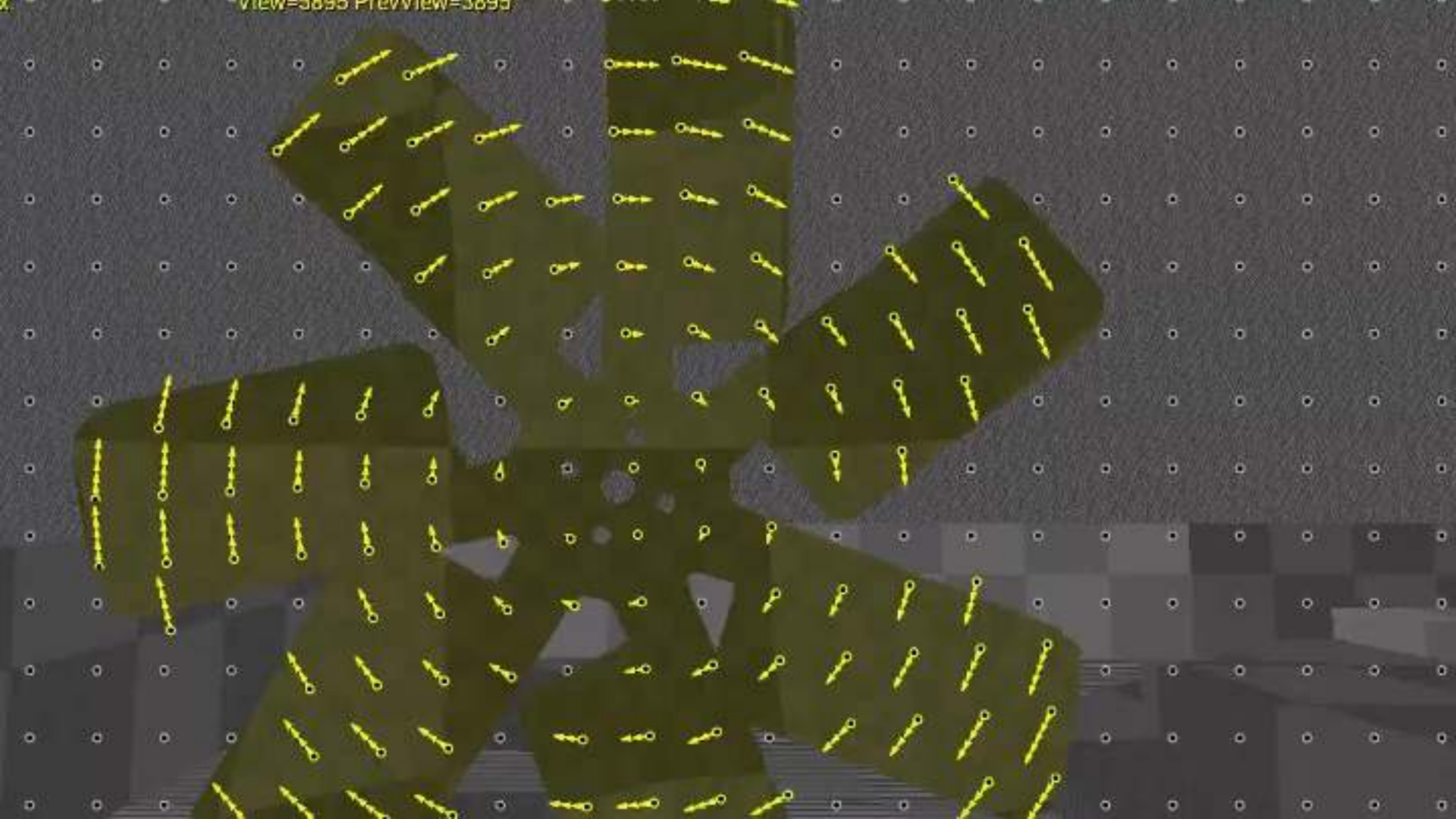
Another Note on TSR

need to output the correct velocity vector for each object in order for TSR to understand the change from the past frames

Incorrect output of velocity vectors may cause strong ghosting (blurring)



Target
High Resolution


View=3895 PrevView=3895

5.3

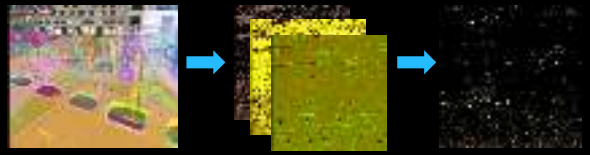
5.4

There is more

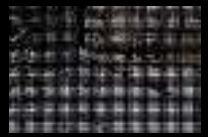

Lumen

Lumen Algorithm Overview

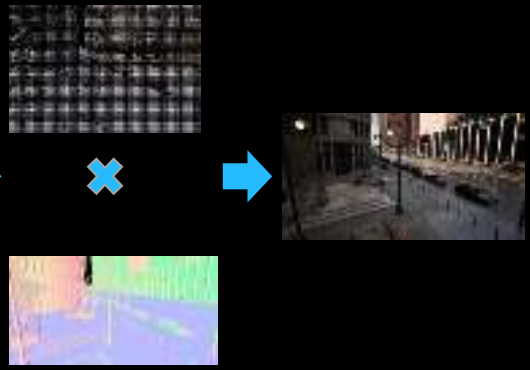
Lumen Card Generation



Construction of Lumen Scene and Lighting on Lumen Scene



On-screen Low resolution GI calculations

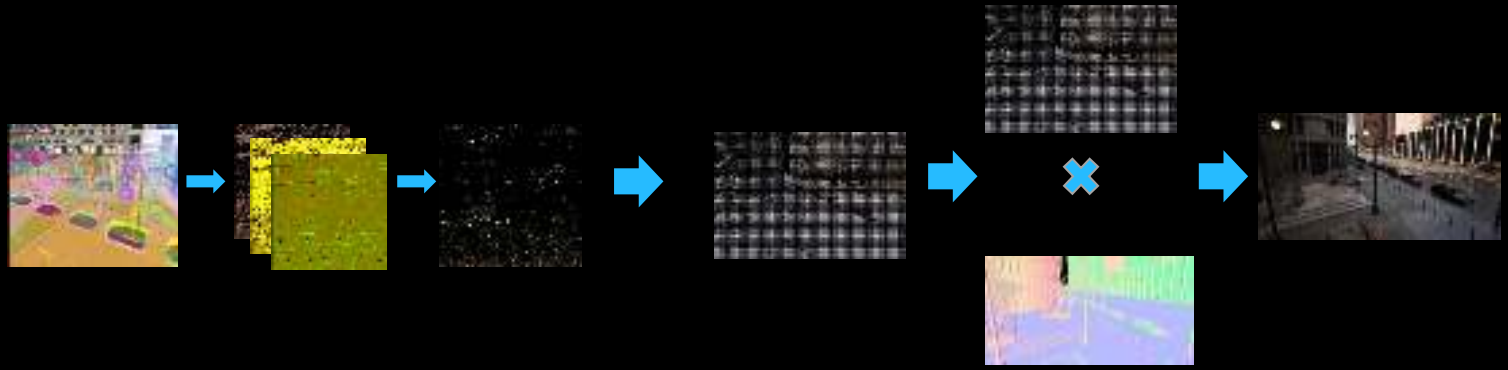


Upsampling on Lumen

Advance Preparation (Editor)

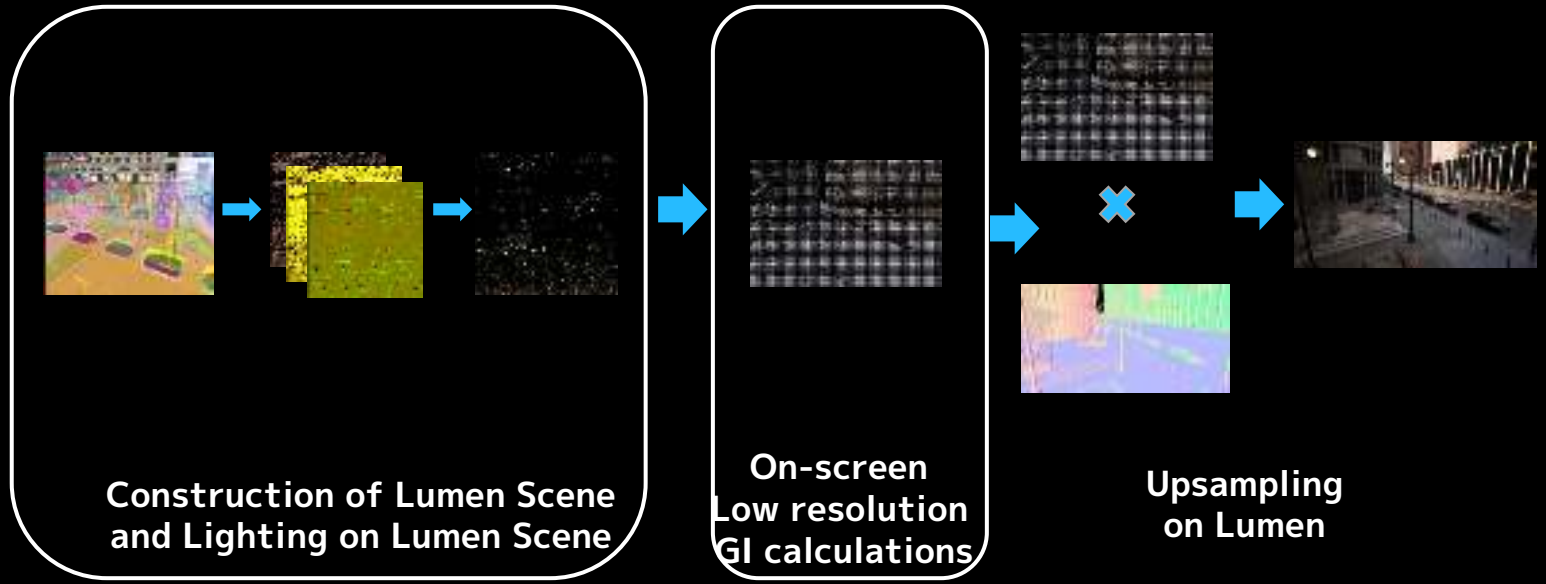
Runtime

Lumen Algorithm Overview



Can be calculated in one frame?

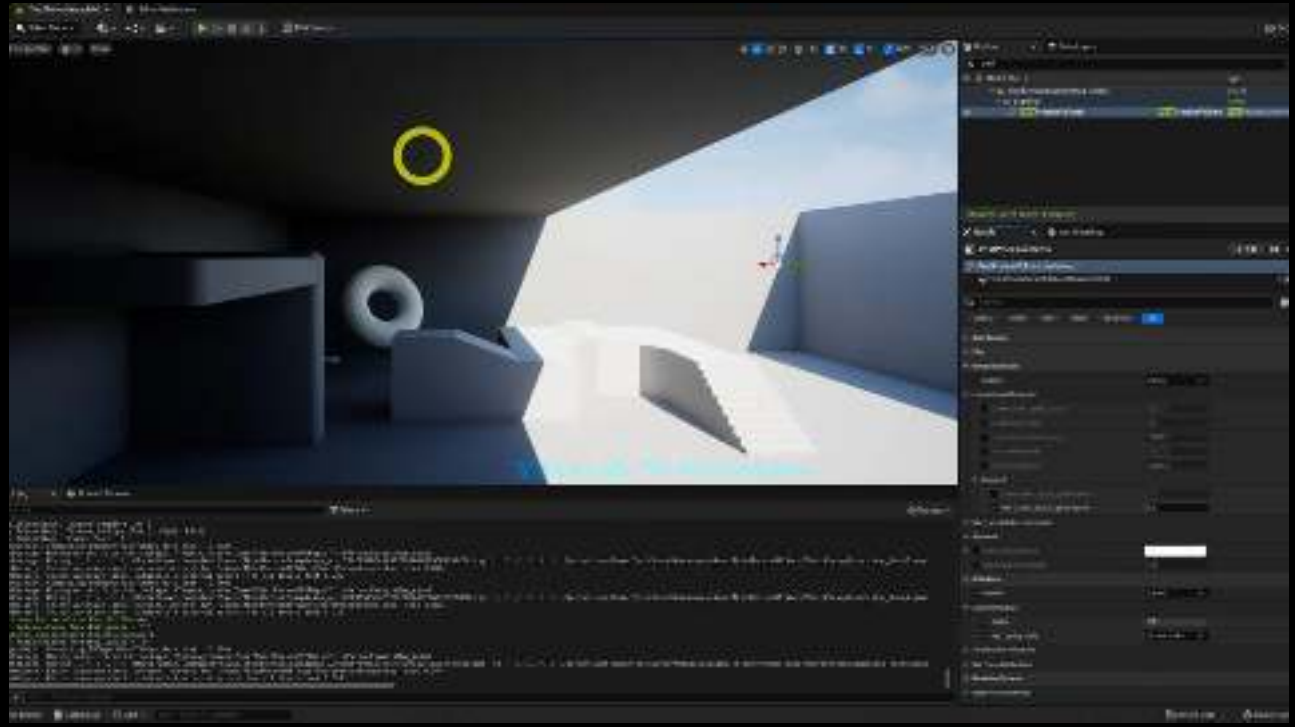
Lumen Algorithm Overview



Various internal processes are split into multiple frames in order to reduce the number of processes in a single frame

Splitting into multiple frames means
gradually and incrementally
converging on the correct lighting

Example of delay due to frame distribution



Delay for fast changes in lighting

Example of delay due to frame distribution

Delay when switching cameras

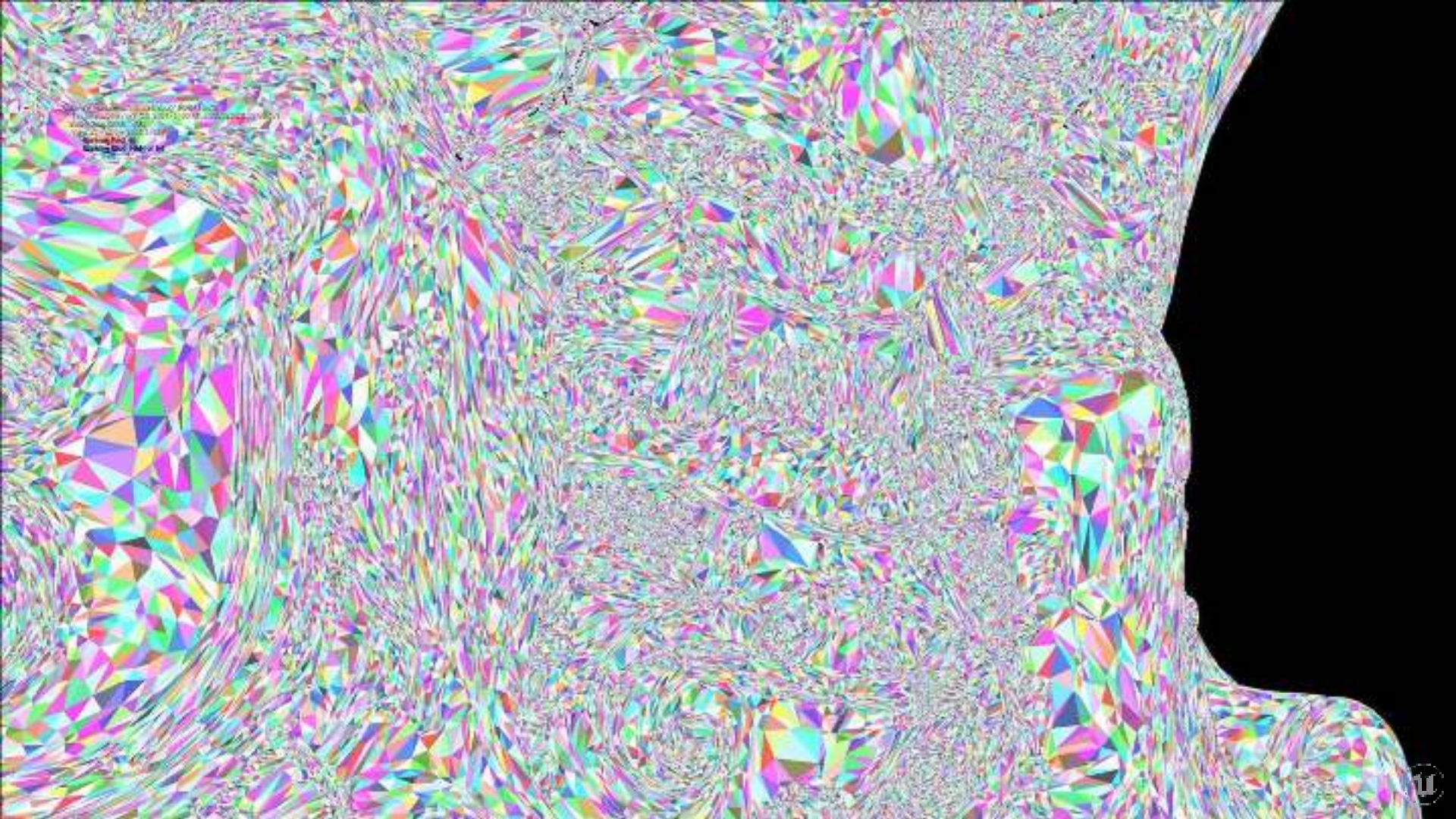
Multi-frame processing Summary

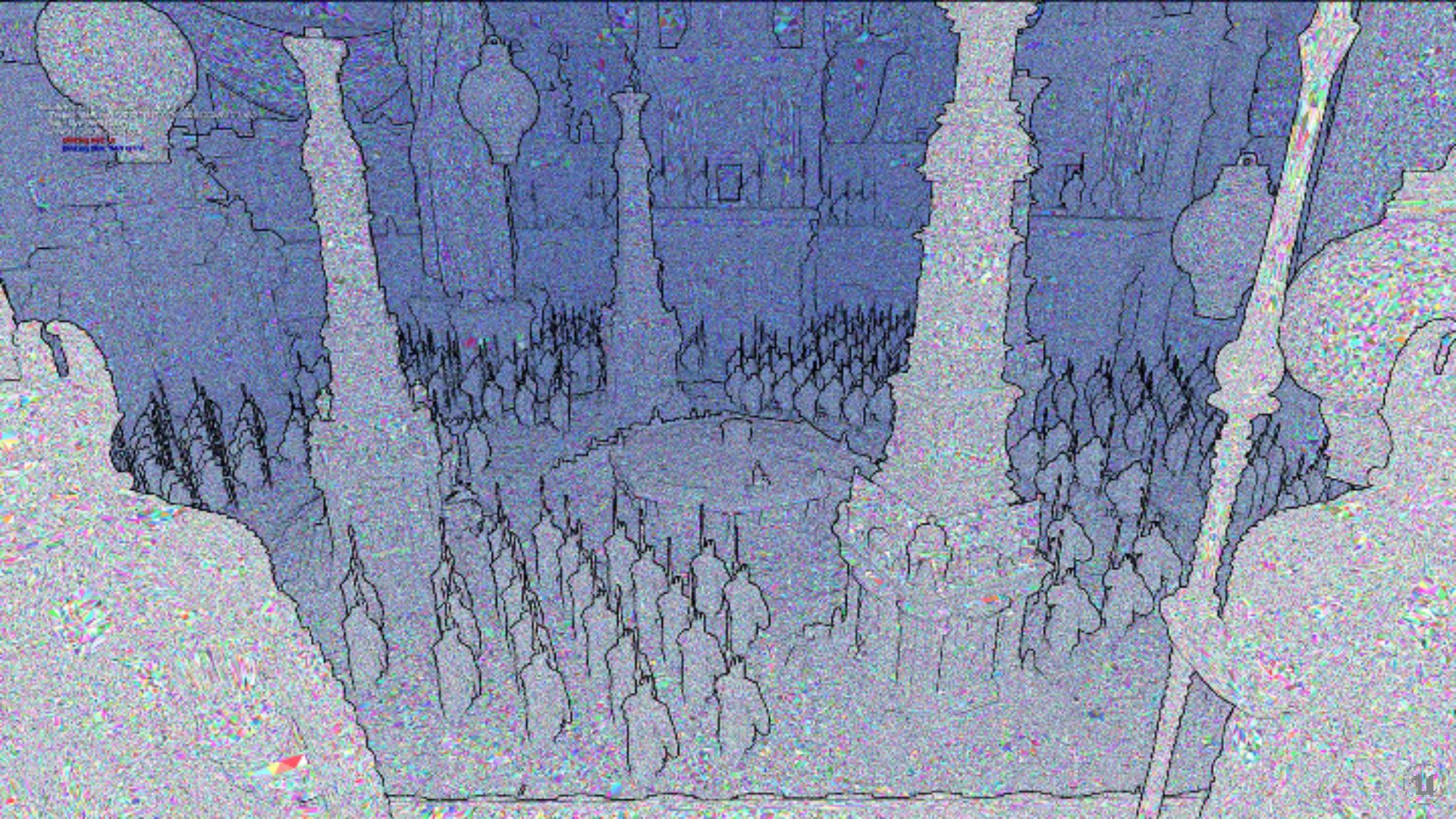
- To maintain real-time performance,
  UE5 has a process that uses multiple frames
- **Temporal Super Resolution**
  - Low internal resolution or low FPS will take more time to create a high-resolution image, making strong ghosting
  - Ghosting may occur in areas where the velocity vector is not correct
- **Lumen**
  - Lumen GI have some multi frame processings
    Lumen's GI lighting only scenes easily cause noticeable lighting delay
    In other words,
    Emissive-based lighting is still Experimental

Agenda

- Introduction
1. Multi-frame processing
2. Virtualization and Streaming
3. Utilization of Existing Assets
- Summary

Virtualization and Streaming

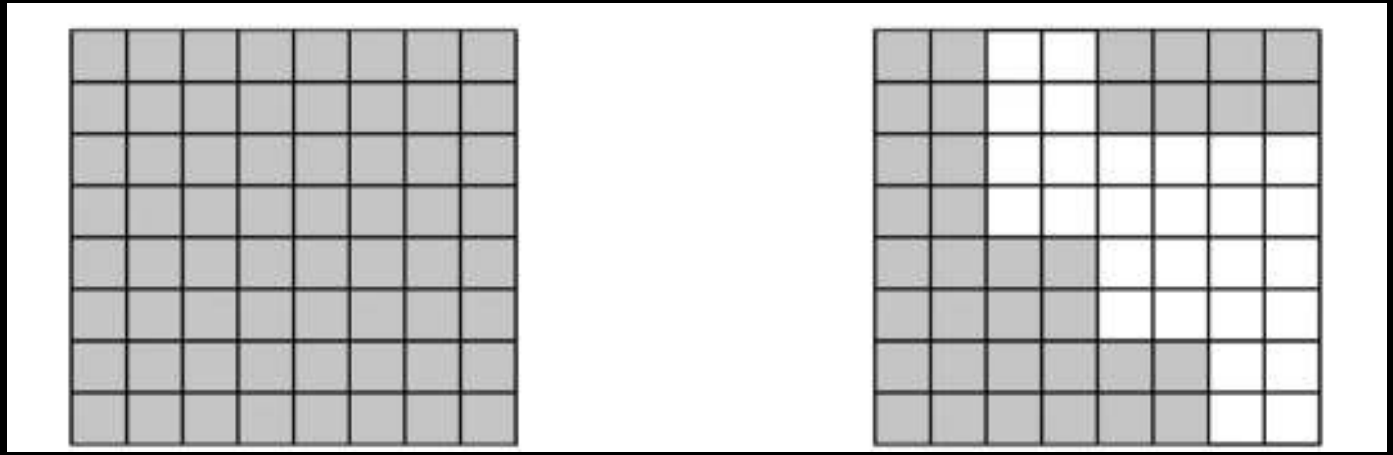




Nanite = Virtualized Geometry

- High-Density polygons
  - Data size is too large to fit in memory
  - Takes time to render

Virtual Texture

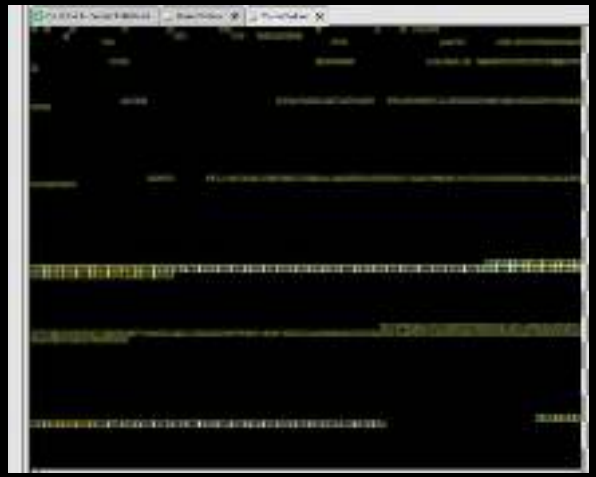


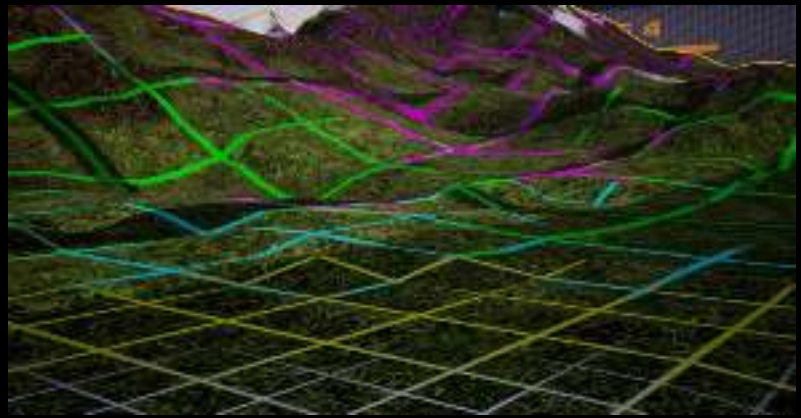
non-Virtual Texture

Virtual Texture

Virtual Texture

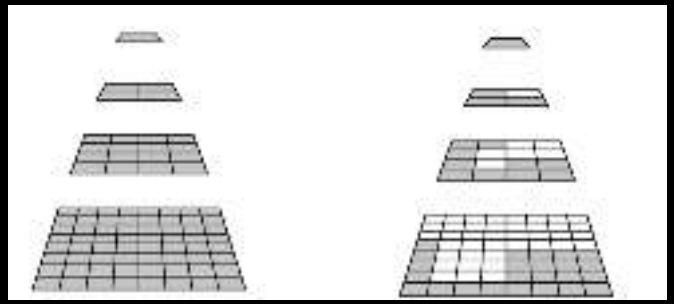
- Pros
  - Only what is actually visible from the field of view can be read!
    - This reduces memory consumption by a large amount
- ↓Visualization of a Virtual Texture memory



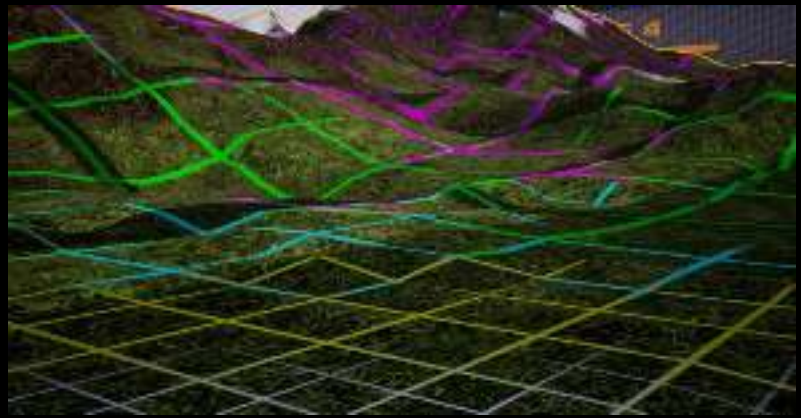


Virtual Texture

- MipMap can also be read partially for each Mip tile
  - Load a detailed MipMap for nearby tiles
- Potential to significantly reduce memory consumption



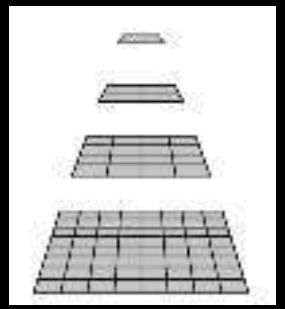
Virtual texture can also partially load MipMaps.

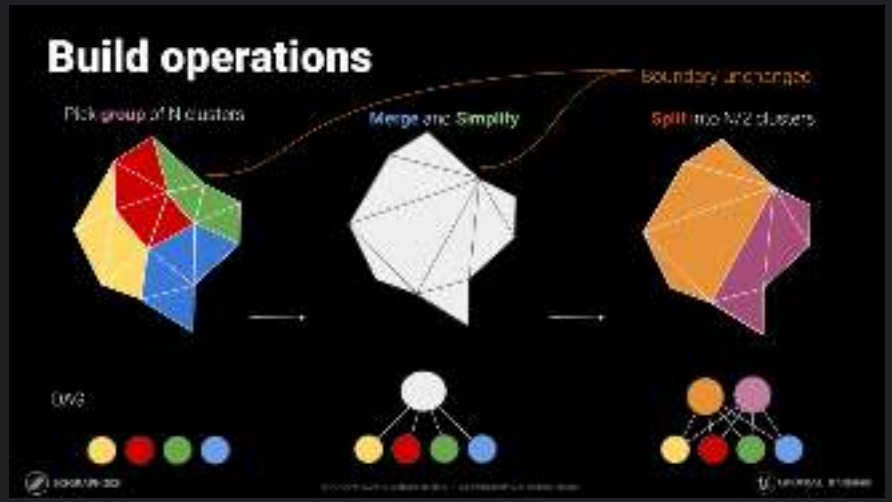


Nanite

- Nanite is a polygon object version of Virtual Texture

- In advance
  - Clustering
  - Into a hierarchical structure





SIGGRAPH2021 - Nanite Deep Dive

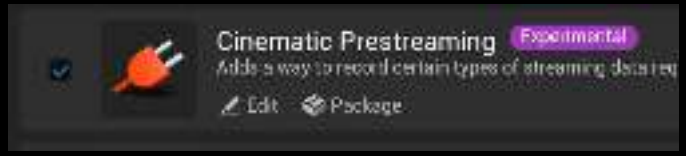

Perspective
Lit
Show

Nanite has the same constraints as Virtual Texture

Nanite streaming may not be completed in time
when camera cuts are abruptly switched

Countermeasure: Cinematic Prestreaming

- When Camera cuts and objects are predetermined
  - Record and save what Nanite and Virtual Texture to load
  - By pre-loading that data during actual Sequencer playback, the delay can be reduced to zero



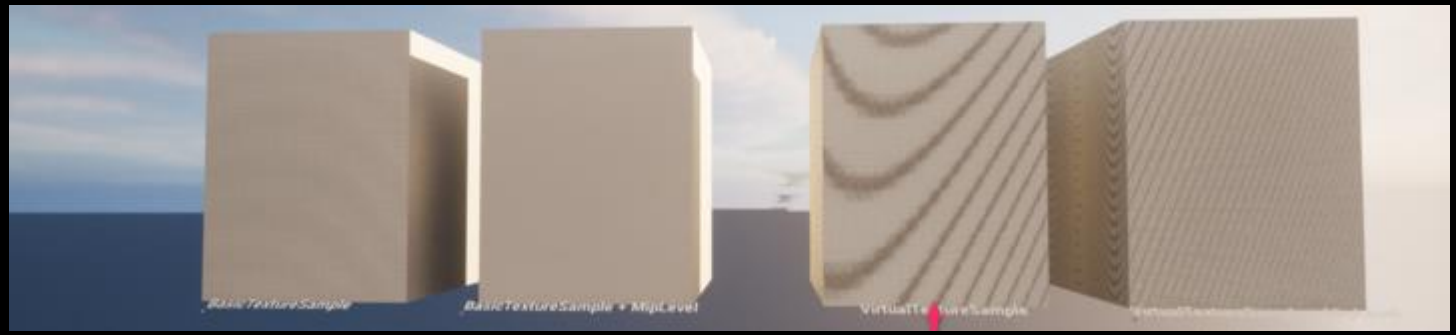
Sudden camera movement or teleportation

- Streaming delay may cause low-resolution objects to be momentarily visible
  - Non-continuous camera movement
  - Transition between levels

- In such cases, it may be necessary to take measures such as fade-in

Caution: Virtual texture’s min mipmap size

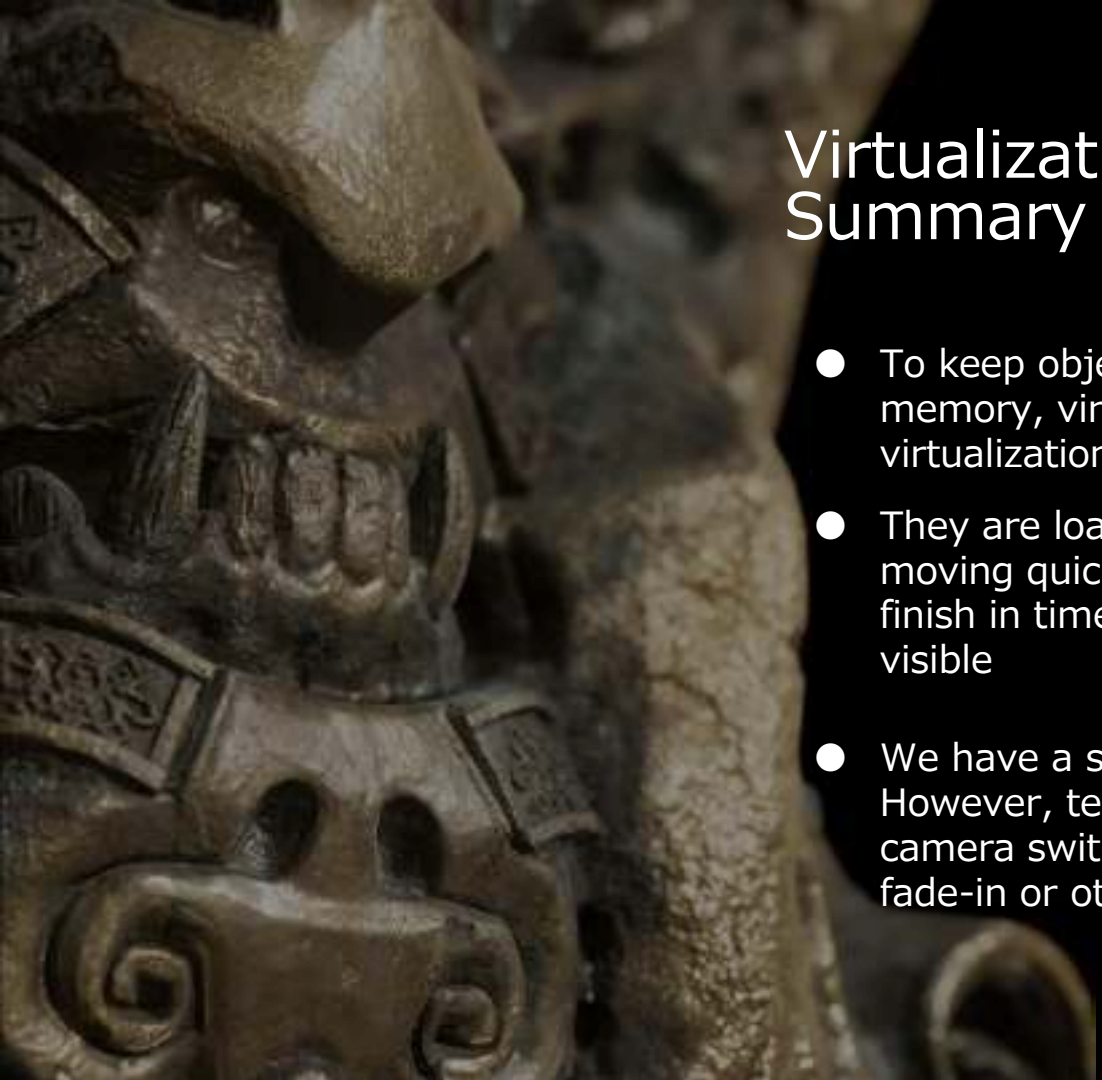
- **Min mipmap size = tile size (default 128*128)**
  - High frequency virtual textures can cause terrible moare



Normal Texture **Virtual Texture**

Caution: Virtual texture's min mipmap size

- **Min mipmap size = tile size (default 128*128)**
  - High frequency virtual textures can cause terrible moare
  - TSR can mitigate the issue but cannot eliminate it completely



Virtualization and Streaming Summary

- To keep objects with huge data sizes in constant memory, virtual texture and Nanite utilize data virtualization technology
- They are loaded by streaming, so if the camera is moving quickly, the streaming may not be able to finish in time and low-resolution objects may be visible
- We have a solution for cutscenes.
  However, teleportation and other instantaneous camera switching is not yet supported and requires fade-in or other countermeasures

Agenda

- Introduction
1. Multi-frame processing
2. Virtualization and Streaming
3. Utilization of Existing Assets
- Summary

Utilization of Existing Assets

Megascans / Marketplace


Image courtesy of Quixel

You can use many existing assets for free!

You can use many existing assets for free!

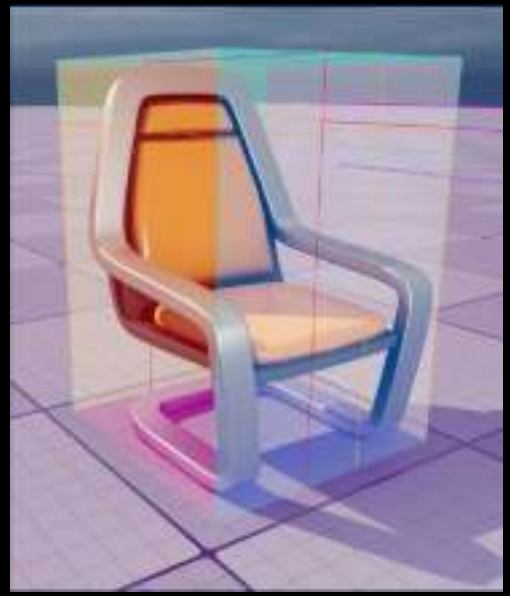
Something is wrong with the rendering...
GPU costs jump up…

Modular Assets vs Lumen Cards

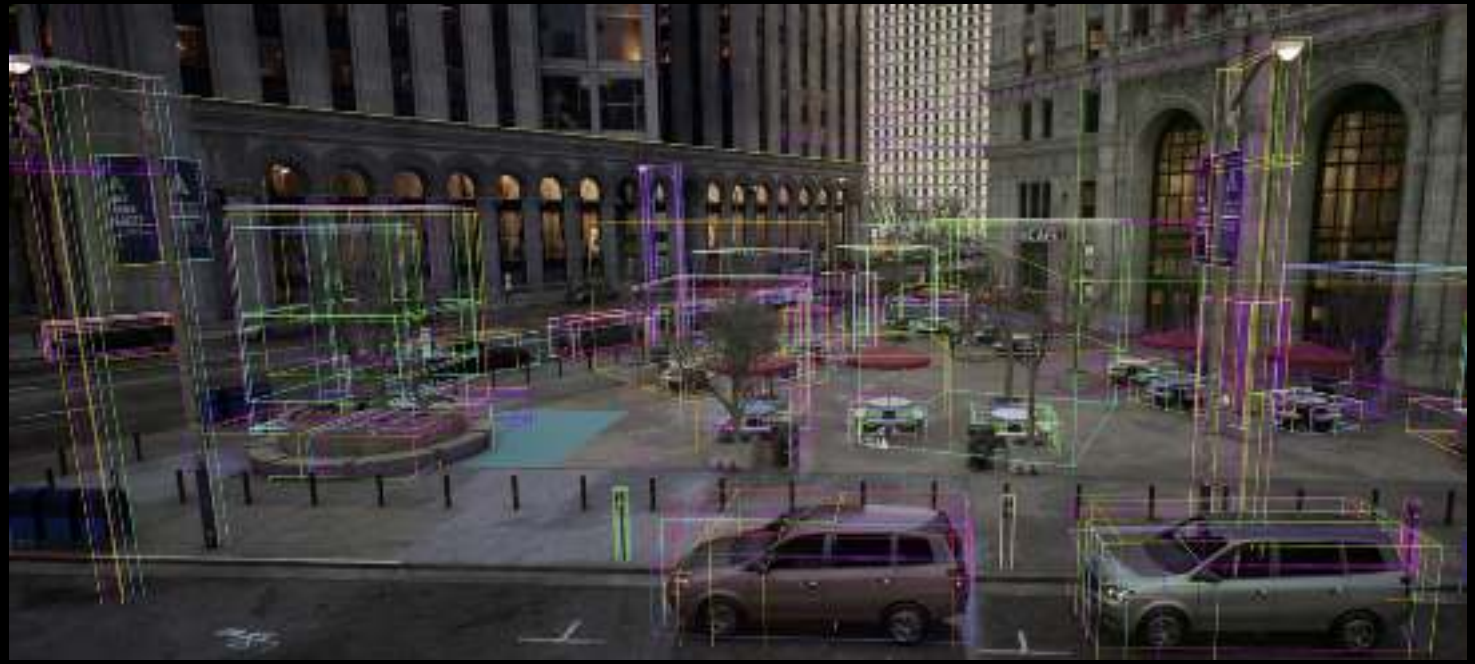
Modular Assets: not made as a single unit, broken down into parts, which are then combined to create an object

Lumen Cards

In order to calculate GI,
Lumen creates Lumen cards covering each object to calculate GI



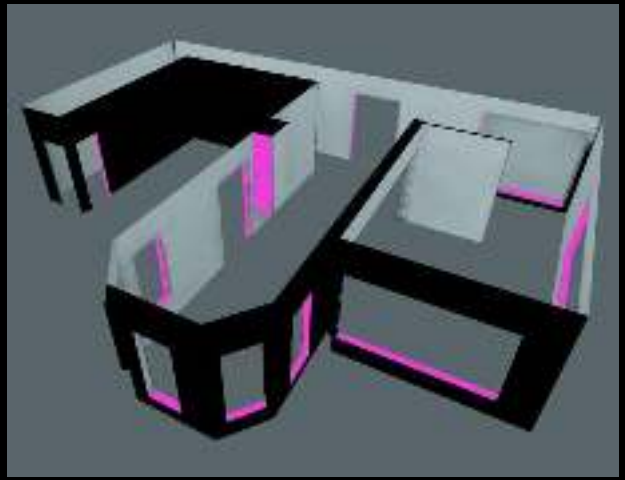
Modular Assets vs Lumen Cards



Lumen Cards in City Sample

Modular Assets vs Lumen Cards

- Basic rule
  - To get Lumen working right, create your objects with modular assets
  - Lumen Card can only create simple shapes, so if you create a complex object in one piece, there will be more areas that Lumen Card cannot consider, resulting in inaccurate GI calculations
  - Pink is areas where Lumen cannot be calculated

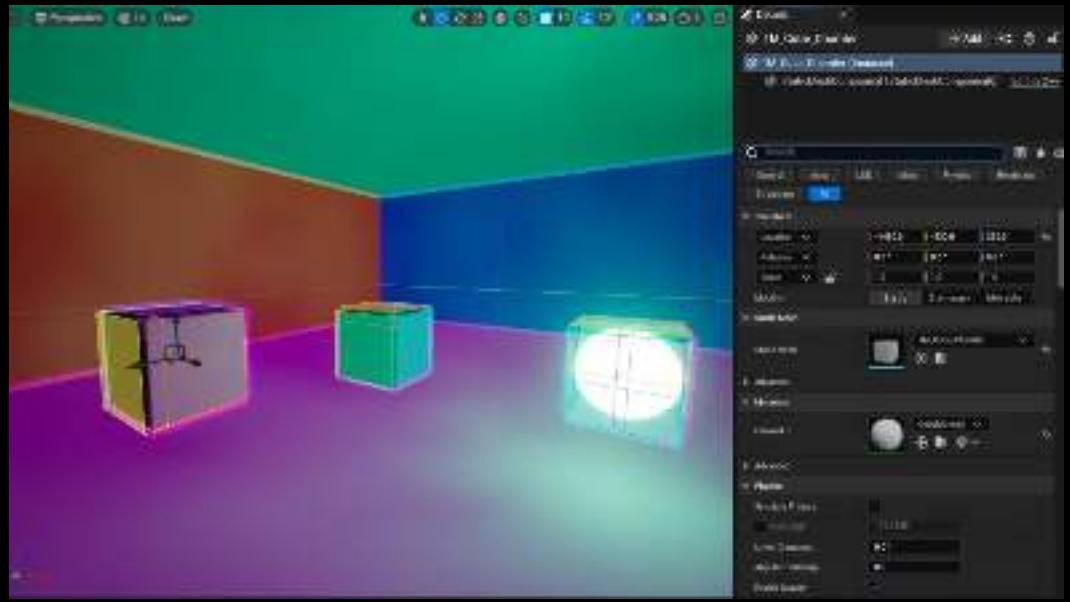


Modular Assets vs Lumen Cards

- But… Too many modular assets increase Lumen computation

RayTracingGroupID

- Lumen cards of objects with the same RayTracingGroup ID can be marged

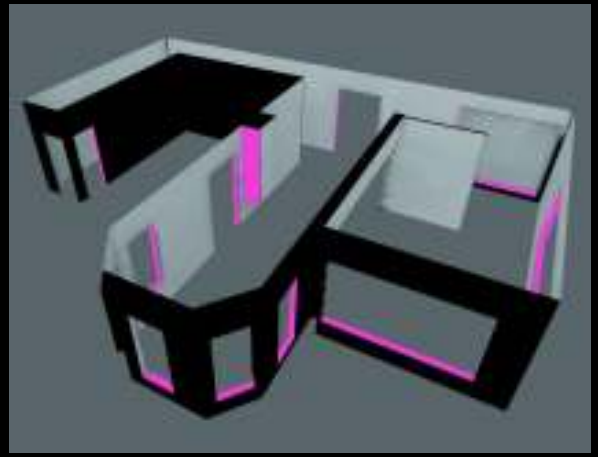


RayTracingGroupID

- Lumen cards of objects with the same RayTracingGroup ID can be combined. This drastically reduces processing load and memory usage

Modular Assets vs Lumen Cards

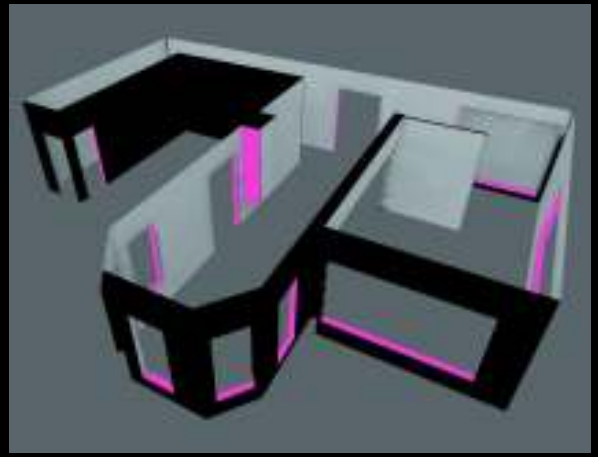
Lumen Cards covering an entire building cannot cover complex objects



Can not enter the buildings

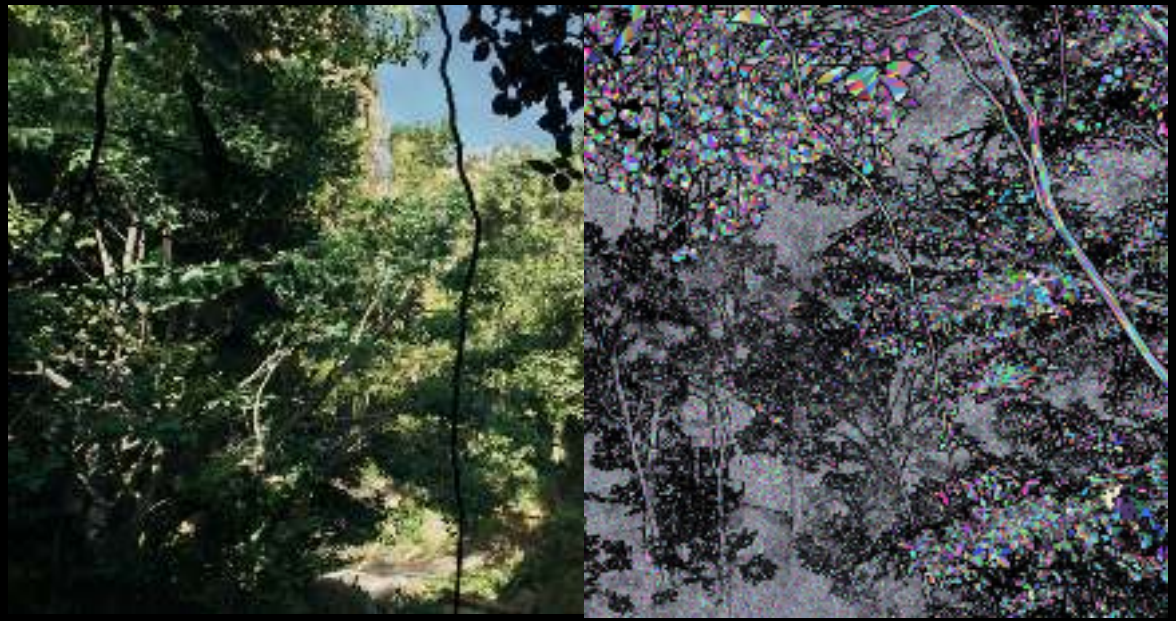
Modular Assets vs Lumen Cards

Currently, there is no system that satisfies both of these requirements at the same time.
It might be necessary to separate the exterior and interior walls, etc.



Note: Nanite restrictions

- From UE5.3
  - Masked Material is now supported! However, the processing load will be very high



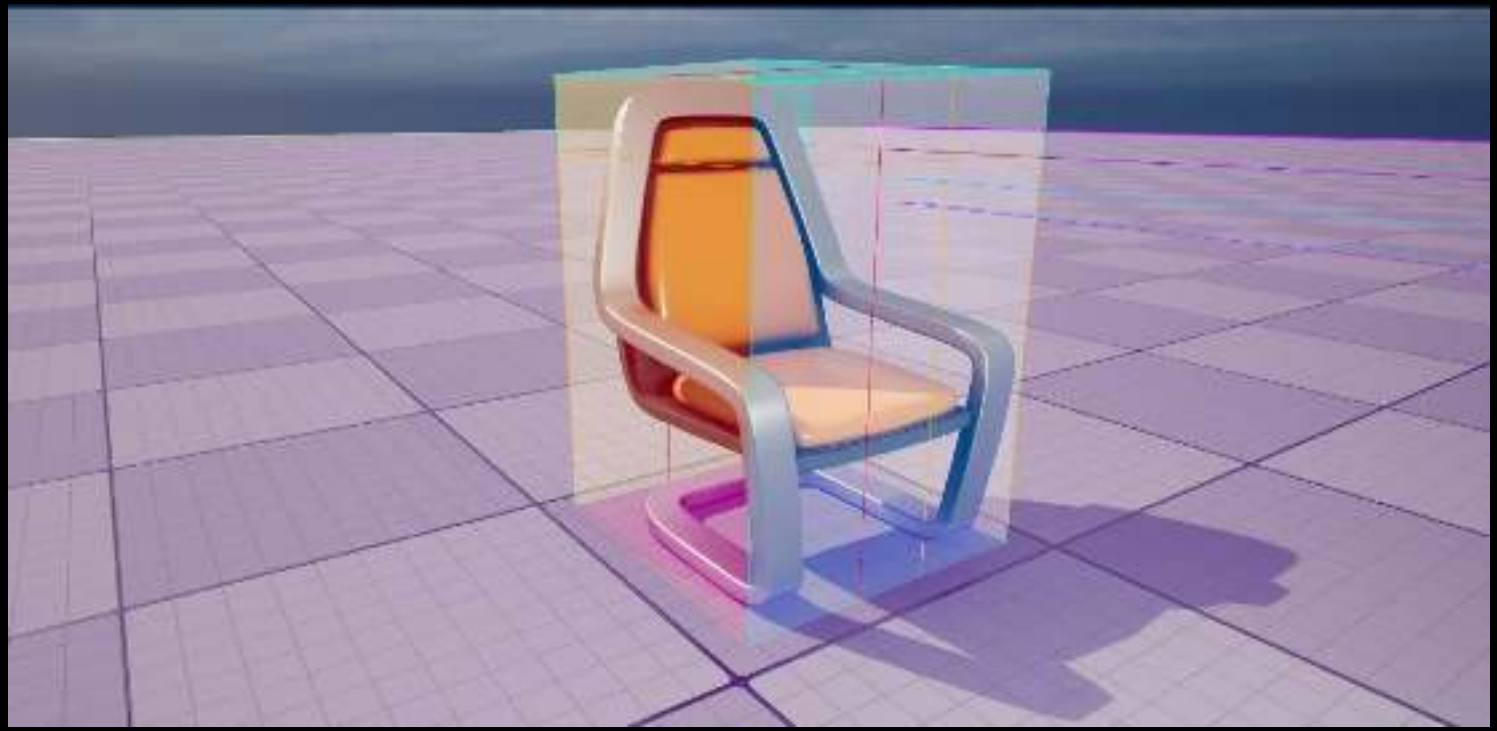
Nanite

Non-Nanite

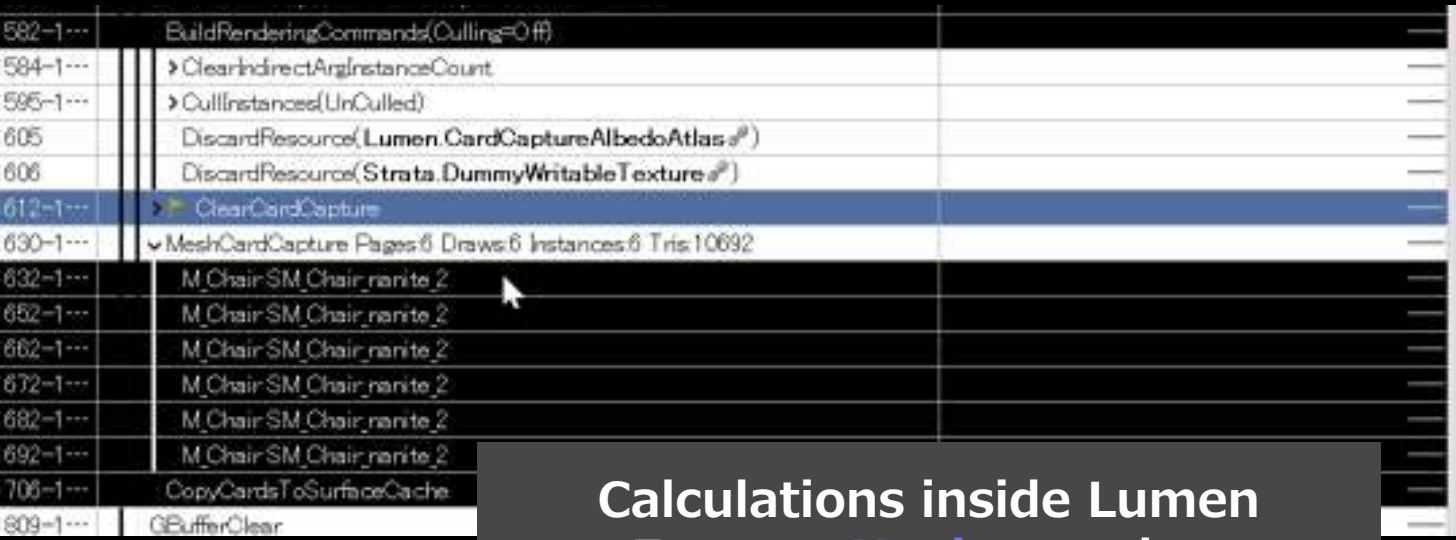
OK, I'll set all objects as Nanite except for grass and leaves!

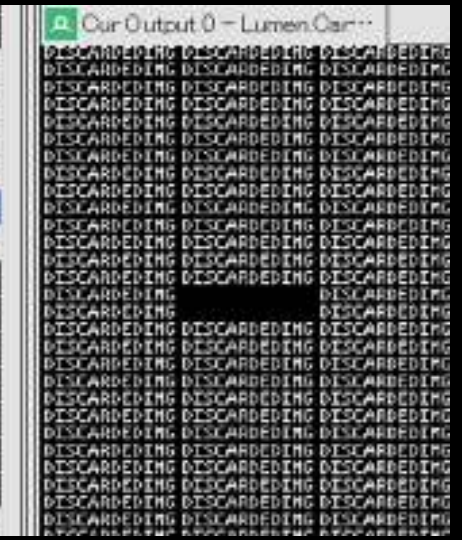
… Does this work well?

Note: There is a difference in processing load for Nanite and non-Nanite objects on Lumen/VSM



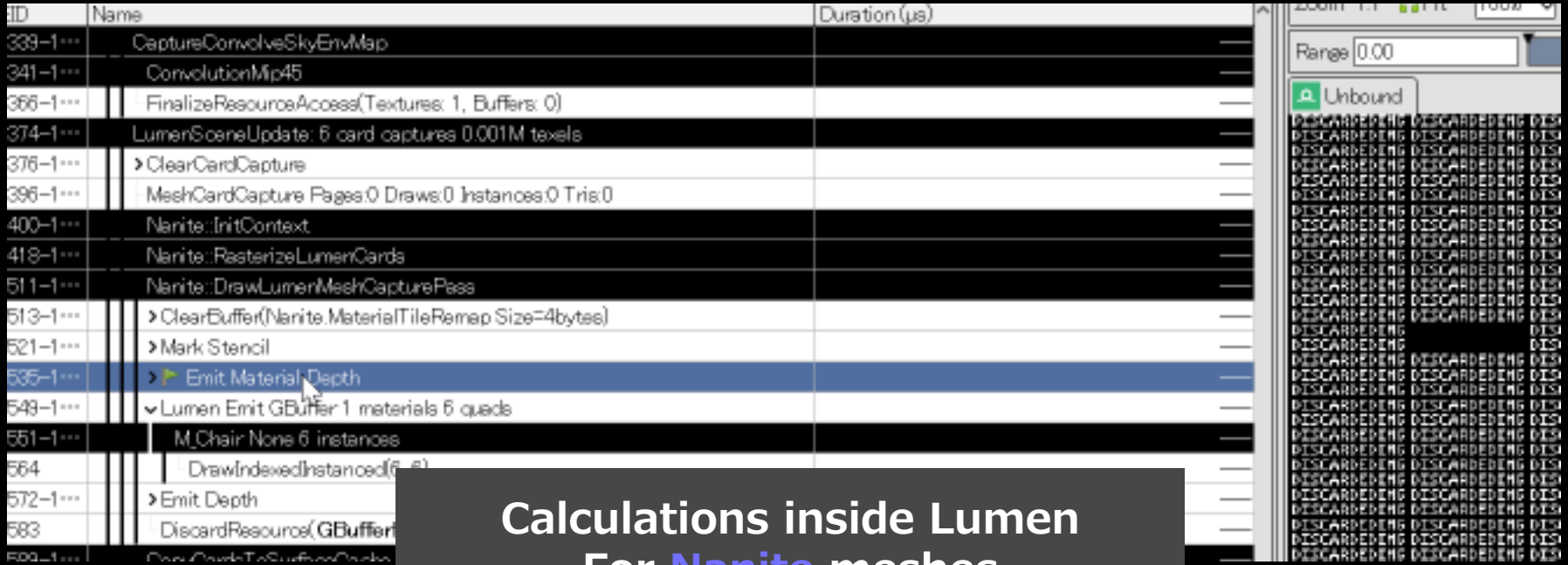
Note: There is a difference in processing load for Nanite and non-Nanite objects on Lumen/VSM





Calculations inside Lumen
For non-Nanite meshes

Note: Lumen/VSM can handle Nanite faster.



Calculations inside Lumen
For Nanite meshes

Nanite

Non-Nanite

OK, I'll set all objects as Nanite except for grass and leaves!

… Does this work well?

Utilization of existing assets

- Unreal Engine has a large number of assets available for free, including Megascans
- However, it is not always possible to make use of these assets as they are
- They may be overloaded or may not render the correct results
- You need to check the compatibility of your scene and the technology you wish to use

Agenda

- Introduction
1. Multi-frame processing
2. Virtualization and Streaming
3. Diversion of existing assets
- Summary

Vague expectations for UE5


Thank you!