



RACECAR

Modern real-time car rendering in Vulkan

Charles Wang, Anthony Ge, Saahil Gupta, Aaron Jiang
CIS 5650 Fall 2025

Meet the (Race) Engineers



Charles Wang
DMD & CGGT '26

charleszw.com
zwcharl@gmail.com



Anthony Ge
DMD & CGGT '26

geant.pro
geant@seas.upenn.edu



Saahil Gupta
DMD '27

saahil-gupta.com
saahil@seas.upenn.edu



Aaron Jiang
DMD & CGGT '26

aaron-jiang.com
aajiang@seas.upenn.edu

Introducing RACECAR

- High fidelity real-time car rendering engine in C++/Vulkan
 - ◆ Inspired by games like *Gran Turismo 7*, *Forza*, Formula 1
- RACECAR aims for:
 - ◆ High quality reflections
 - ◆ Realistic, physically-based car material rendering
 - ◆ Photorealistic atmospheric lighting for outdoor scenes





Breaking down the problem

The four main subproblems:



Atmospherics



Ray tracing

Breaking down the problem

The four main subproblems:



Car materials

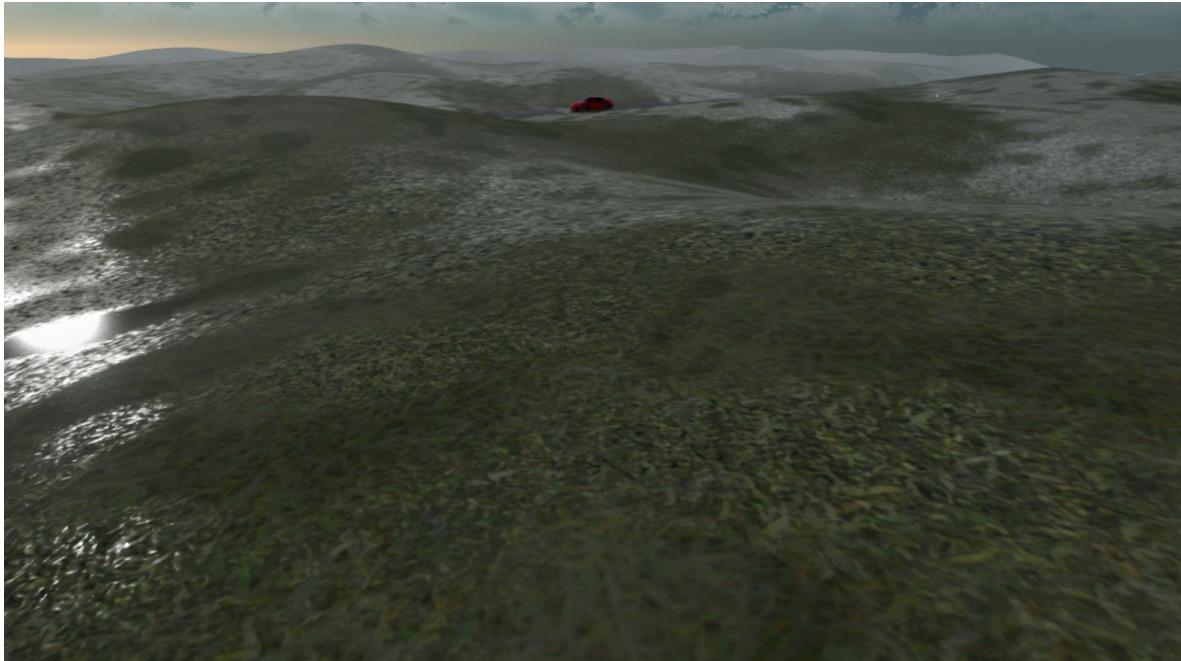


Post-processing

Furthermore... Terrain

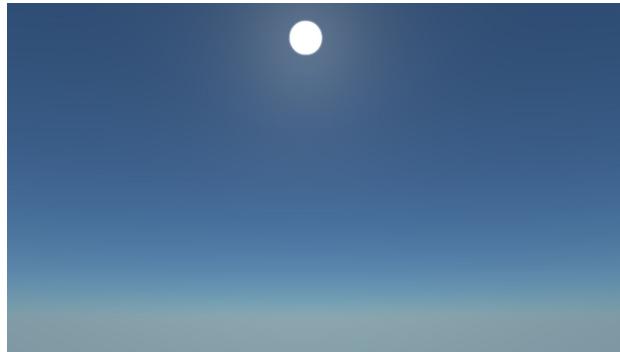
What good is a car if it doesn't have anything to drive on

- Procedural heightmap
- Hardware tessellation for LODs
- Terrain "wetness" factor and "snow" materials

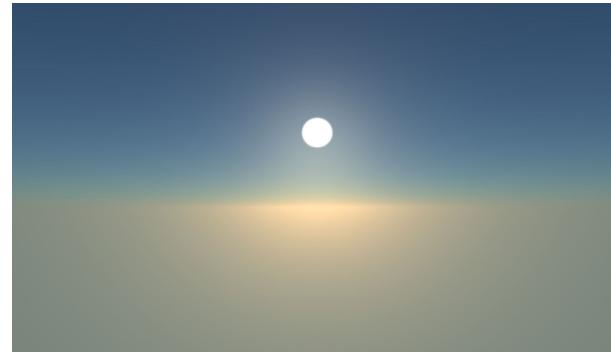


Atmospherics

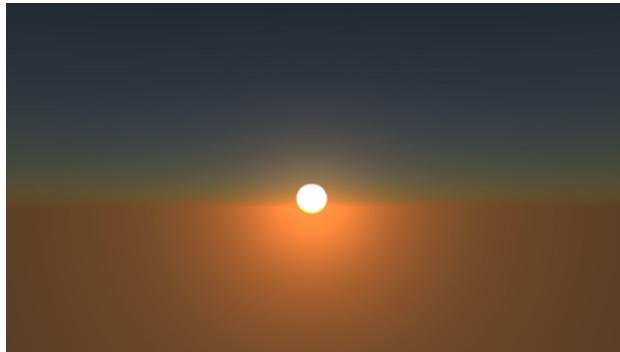
Physically-based atmospheric scattering



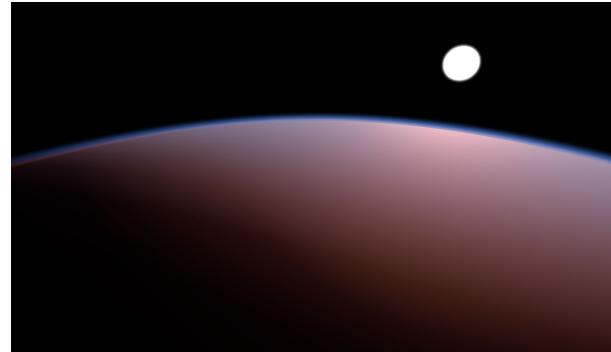
Sun is high in the sky



Approaching sunset



Sunset

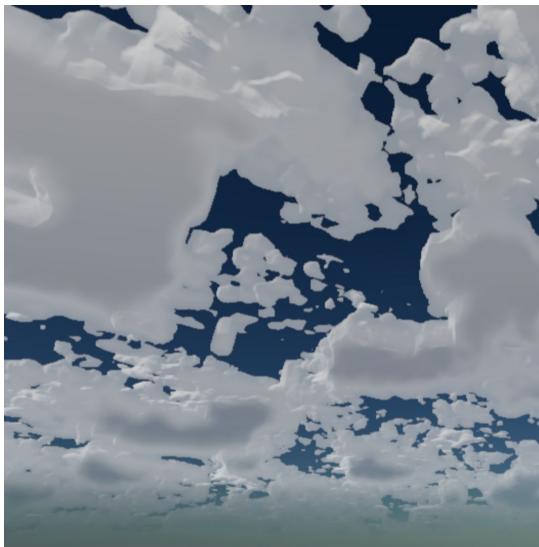


Outside Earth's atmosphere

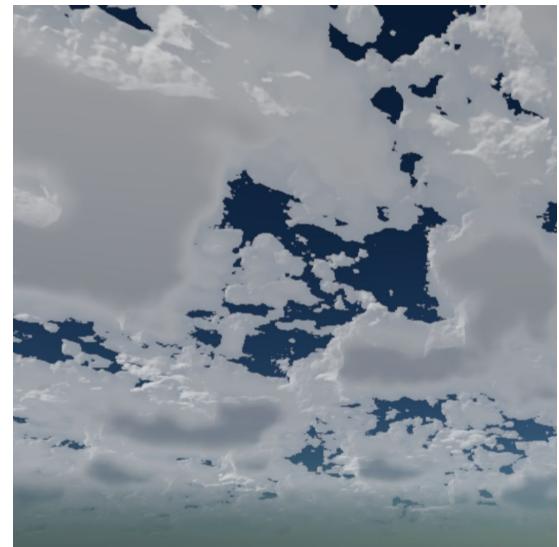
Volumetric clouds



Cumulus map



With low frequency noise



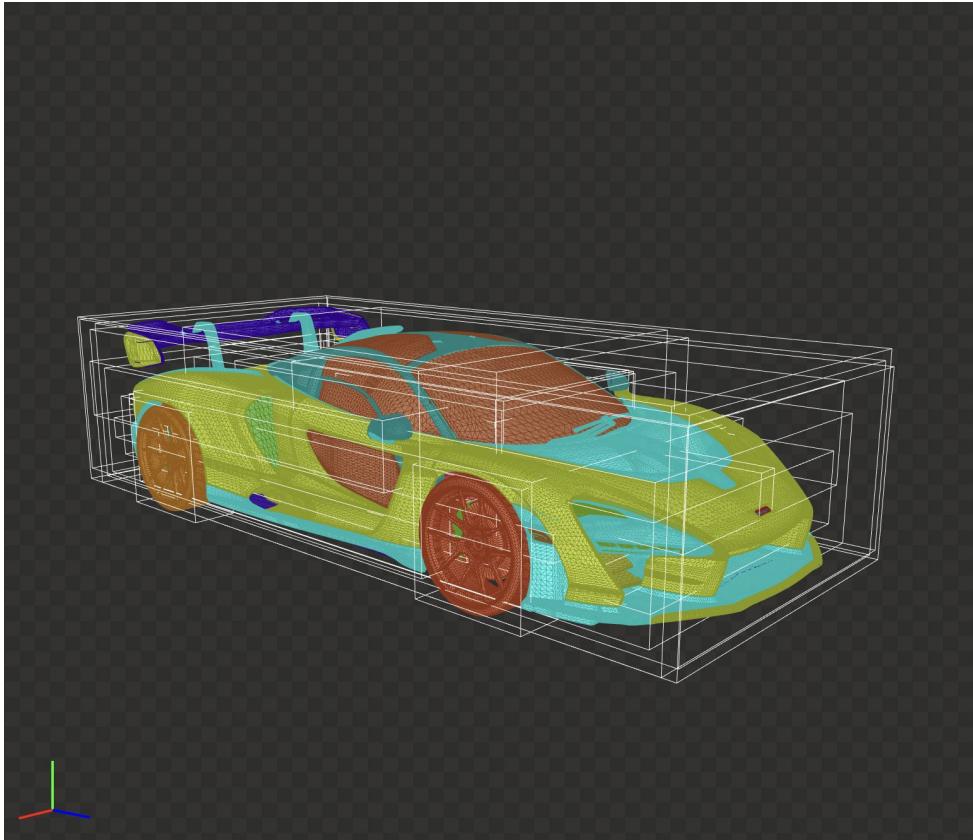
With low + high frequency noise

Final result



Shadows and Reflections

Hardware Acceleration Structures



- Features a full ray tracing pipeline and material system
- Making use of hardware acceleration structures



No shadows

A blue sports car is shown from a front-three-quarter perspective, driving on a dark, textured road. The background features a vast landscape with rolling hills under a cloudy sky.

SSAO

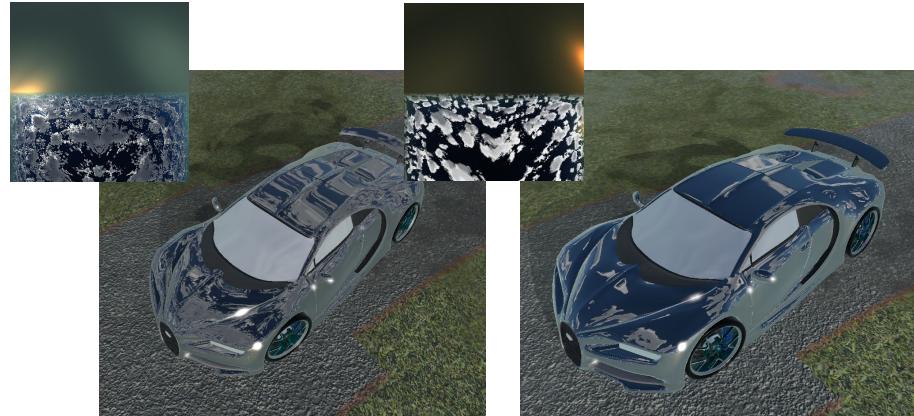


SSAO + Ray Traced Shadows

Reflections and image based lighting



Porsche using IBL reflections



High-frequency
detail, visually messy

Low-frequency,
cheaper and nicer

Ray traced reflections

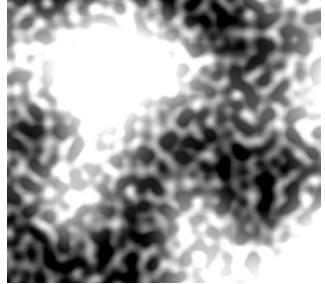


Total Specular Reflection
(Puddles)

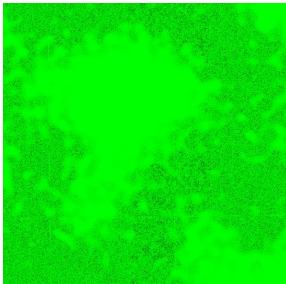


Rough Normal Reflection
(Wet ground)

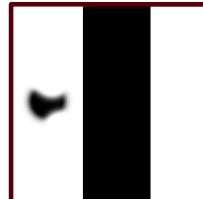
Procedural terrain and lighting



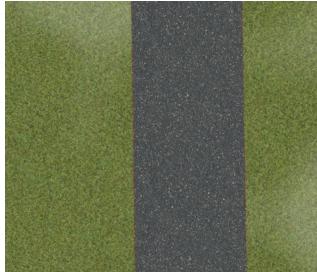
Overall glossiness



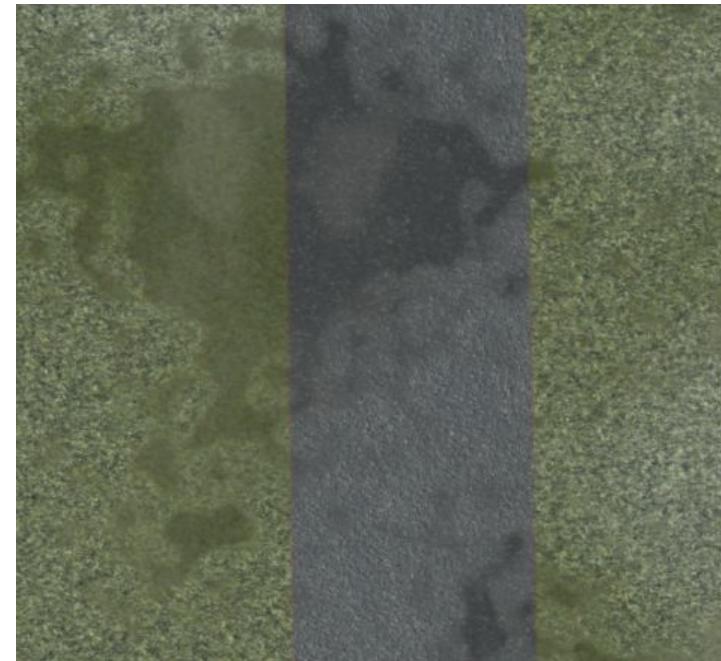
Overall normal



Layer mask



Overall albedo



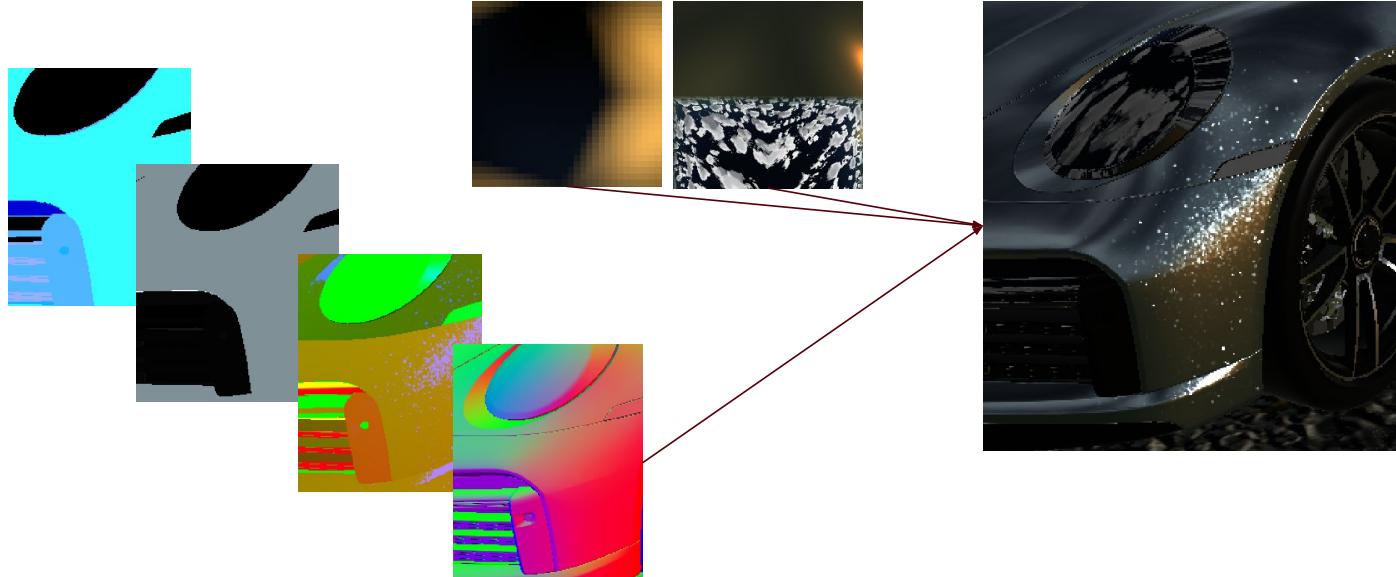
Final terrain

For puddles, we use Uncharted 4's wet surface shaders.

Car material rendering

Car materials

- Car materials are complex - often made of many different layers stacked together
- Approximate realistic car-material rendering via PBR
- Support multi-layer finishes for microflakes and clearcoat



Microflakes

- Microflakes are rendered using Deliot's 2023 paper on anisotropic glint rendering
 - ◆ Controllable log-density and other params help simulate microflakes



Real life reference



Our rendered glints

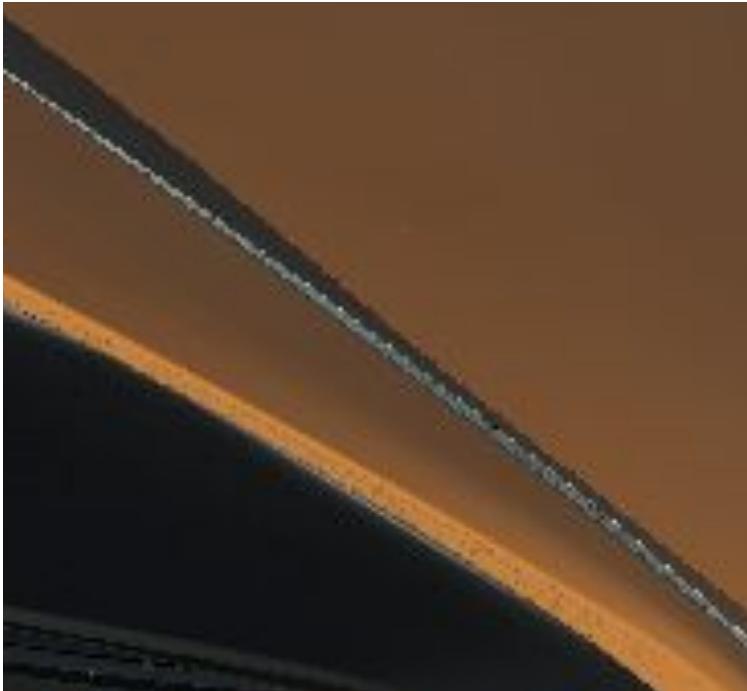
Renders



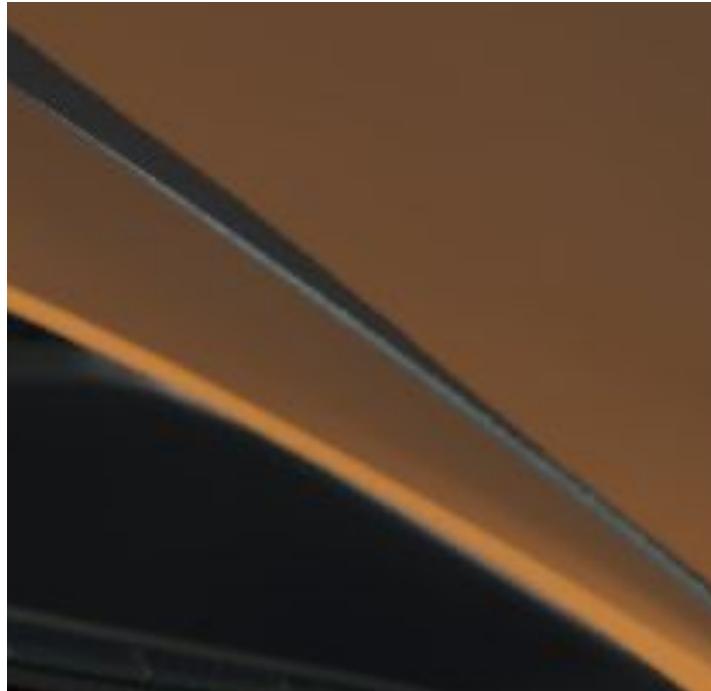
Compute-based post-processing

Temporal anti-aliasing (TAA)

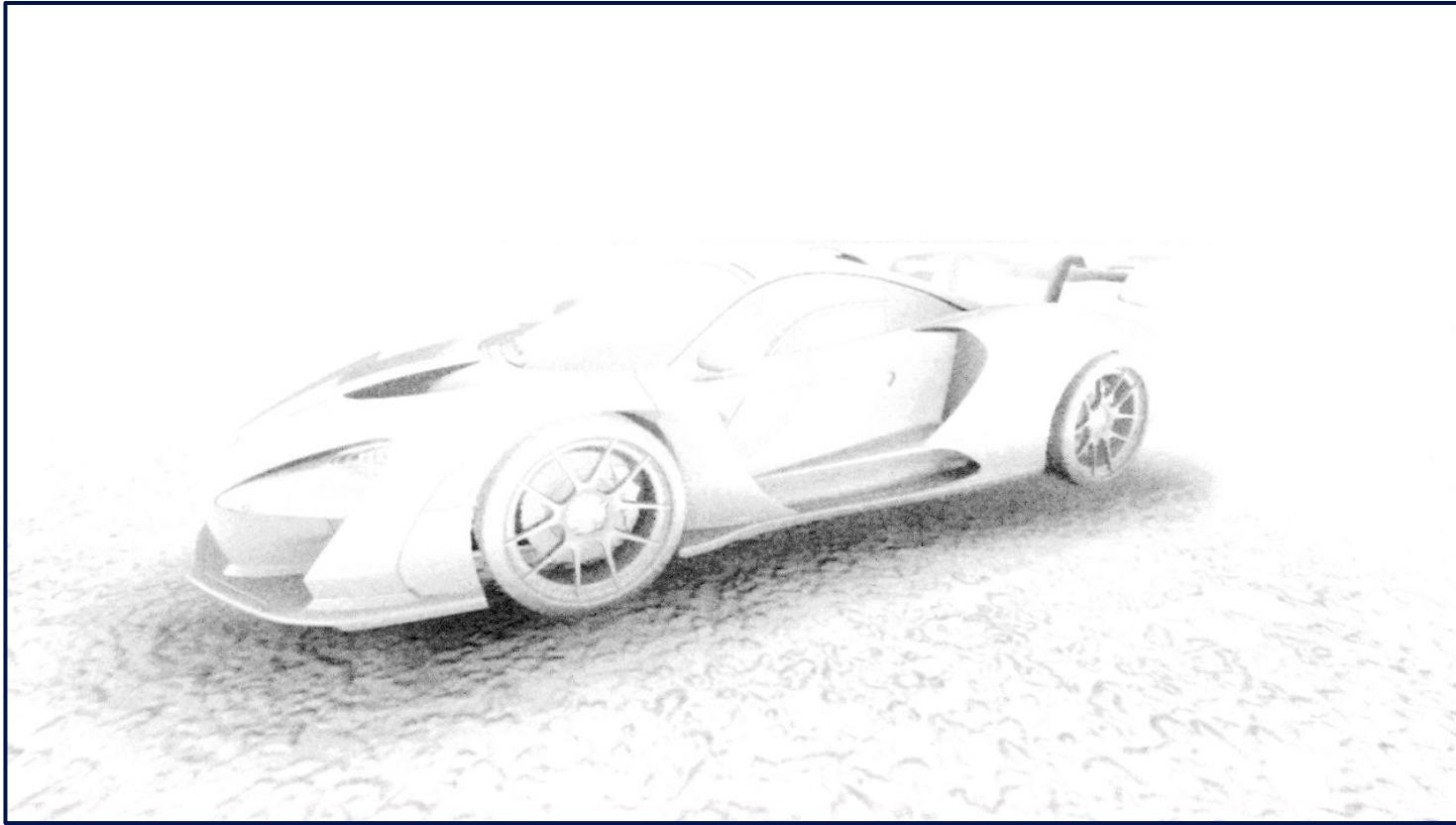
Before



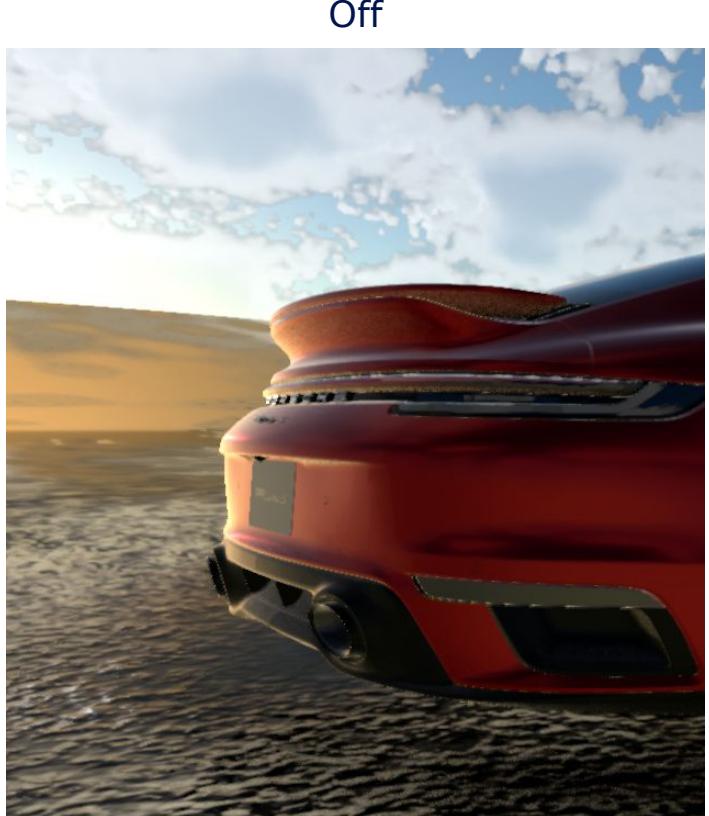
After



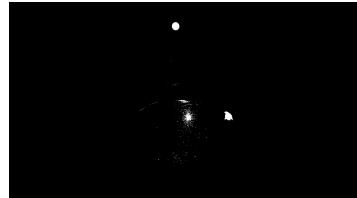
Screen space ambient occlusion (SSAO)



Bloom



Bloom



Brightness threshold



Downsampling



Upsampling



Final gather

Tonemapping



None



ACES 1



GT7 Physically Based Tonemapping
SIGGRAPH 2025





Charles Wang
zwcharl@gmail.com

Anthony Ge
geant@seas.upenn.edu

Saahil Gupta
saahil@seas.upenn.edu

Aaron Jiang
aajiang@seas.upenn.edu

Special thanks to Team Cumulus for providing valuable feedback and suggestions throughout the development of this project.