

Implementation Details

- Graphics API: Direct3D 12
- Programming Languages: C++ for CPU and HLSL for GPU

Successfully run the application on these GPU Models:

- Nvidia RTX 3070
- AMD Radeon RX 9070XT
- Nvidia RTX 3090 (Vislab)

Scene

- From:
<https://sketchfab.com/3d-models/the-tavern-under-the-falling-pigeon-f893c00f9fc34dde9e92870b3aea12f5>
- License: <https://creativecommons.org/licenses/by/4.0/>
- Changes: None made

Music (Played during the demo)

- <https://www.youtube.com/watch?v=eGypRmFHIcNs&list=PLfP6i5T0-DkJmc3oV1NM1lu5fXVi4FSy&index=23>
- License: <https://creativecommons.org/licenses/by/3.0/>
- Changes: None made

Creative Commons BY 3.0 license

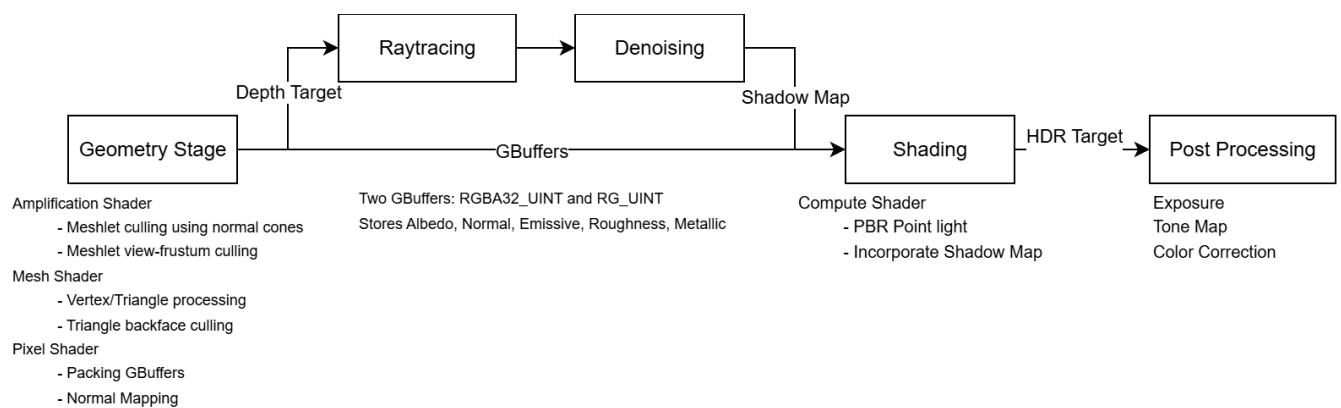
Additional libraries we used are:

- ImGui: basic gui for debugging <https://github.com/ocornut/imgui> (created a fork)
- SpdLog: basic logging: <https://github.com/gabime/spdlog>. Used an extended version from TheCherno and modified it
- Glm: math library: <https://github.com/g-truc/glm>
- Meshoptimizer: Preprocessing Step for data used in the Mesh Shader. Creating and optimizing meshlets <https://github.com/zeux/meshoptimizer>
- DirectX-Headers: Needed for Direct3D12 and some utility functions, which take some work off of the otherwise very strenuous task of setting up mesh shader pipelines in D3D12 (no clue why it is so difficult compared to other pipelines like compute or graphics. These extension headers from Microsoft make the setup process similar to graphic pipelines).
<https://github.com/microsoft/DirectX-Headers>
- cgltf: Loading gltf scene file <https://github.com/jkuhlmann/cgltf>
- WinAPI: Basic Window handling

Controls:

- Rotation of the camera
 - Left click with the mouse in the window and then move it over the screen to look around
- Translation of the camera
 - W - Moving camera forward
 - A - Moving camera to the left
 - S - Moving camera to the right
 - D - Moving camera backward
 - Left Shift - Moving camera down
 - Space - Moving camera up
- Switch between Demo and Manual Camera: F1

Features:



Scene is stored in glTF format and loaded using cgltf. Textures are pre-converted into DDS format. Albedo, MetallicRoughness and Emissiveness BC7 compressed, normal map BC5 compressed. Meshes are processed using meshoptimizer to get meshlets. Raytracing acceleration structures are built from mesh data.

The scene is rendered using a hybrid renderer, using mesh shaders for the geometry and raytracing for soft shadows.

The raytracing starts from the depth image produced by the geometry stage of the deferred shading pipeline. With the depth image we reconstruct the 3d-world position. From there, multiple shadow-rays are shot with slightly different angles (for soft shadows) to the light source to check if some occluder is between the light and the scene position. That will then be written into the shadow-map. This procedure will result in a noisy shadow-map if there are not a lot of shadow rays used, but increasing that can lead to a significant performance issue. Instead of doing that, a denoiser is used to remove most of the noise. That's implemented via a compute shader that applies a

bilateral filter over the whole image. That results in the final shadow map which is used in the further pip

The scene is rendered to a float16 texture in the shading pipeline. The postprocessing shader then reads from the texture, allows for color correction operations like exposure, saturation, brightness and contrast, but is there mainly for the tonemapping writing to the swapchain image.

Using the ImGui Debug menus, you can change light position, strength and radius, the fov, color correction settings, ambient factor, and, most importantly, there is a toggle to visualize the meshlets. The “Fixate Viewpos” checkbox, makes it such that the camera position used for culling is fixed. This can be used to check that the view frustum culling is working. Also, the number of primitives currently rendered are also displayed, showing that the culling is working. The “Disable Backface Culling” option is for the normal cone culling. The normal cone backface culling can also be checked using the fixate viewpos checkbox, or by disabling it and noticing the mesh shader innovations going down.

What has changed from initial proposal: omitted culling on a per-model basis before doing meshlet culling. Also, the setting changed a little bit. We wanted to make sure to have an indoor scene, but we couldn’t find a suitable, free to use scene that matched our cozy description.

2 Complex Effects

- Meshlet Rendering and Culling (Nikolas Kaipel)
- Raytracing Soft Shadows (Fabian Schick)

Other effects

- Simple normal mapping
- Deferred Shading
- PBR Rendering