Advanced Graphics – Physically Based Rendering

Jainesh Pathak

June 2023

School of Computing Science

Newcastle University

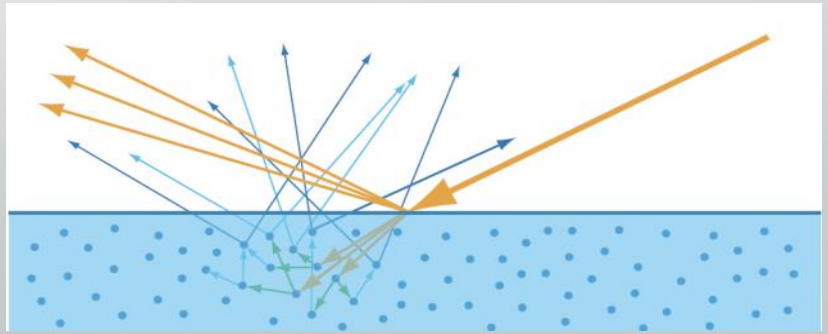
J.P.Pathak2@newcastle.ac.uk

Background

- Lighting plays an important role to bring realism in computer graphics.
- 2.5D games like Doom introduced “Sector-based” lighting and “Light Diminishing”:
  - Sector-Based Lighting:
    - Every sector had a light level with range 0-255. 0 being complete dark and 255 being very bright.
    - Light Attenuation were done where light levels of neighbour sectors gradually decreases to show light is traveling over distance.
  - Light Diminishing:
    - Sector area from player’s point of view is bright enough for player to see and slowly decreases as the distance between area and player increases. Also used in simulating fog.
- As GPUs progressed, lighting and overall video game graphics also progressed.
- 3D Games like Unreal, Doom 3 showcased more dynamic lighting with different light sources: Point, Directional, Spot Lights.
- Blinn-Phong light model became widely popular and used in computer graphics.
  - Uses sum of three components: Ambient, Diffuse and Specular.
- Blinn-Phong shading disadvantages:
  - Violates conservation of energy law. Energy is lost as specular intensity is increased.
  - Doesn’t take the metallic and roughness surface properties on a microfacet level in account.

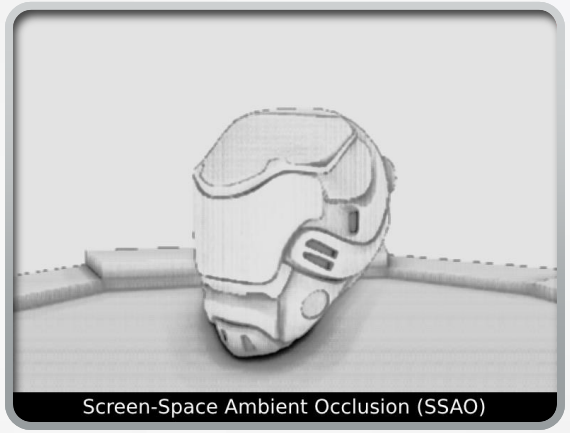
Introduction

- Light Terminologies: Diffuse (Refraction) and Specular (Reflection)
- Physically-based Rendering (PBR):
  - Lighting equations in real-life are too complex and computationally expensive.
  - PBR is collection of techniques to bring light interactions in the real world physics approximation.
- To do physical approximations, it needs to follow three rules:
  - Based on microfacet surface model.
  - Energy conserving – outgoing light reflected should not exceed incoming light.
  - Uses a physically based BRDF.
- Bidirectional reflective distribution function (BRDF):
  - Function that describes the reflectance properties of a the surface on a microfacet level.
  - Takes the incoming light L, view direction V, surface normal N and surface roughness A as inputs.
  - It approximates how much light ray gets reflected based on the surface properties.
- The most famous is the Cook-Torrance BRDF:
  - Fr = Kd * fLambert + Ks * fCook-Torrance.
  - fLambert is the diffusion part.
  - fCook-Torrance is the specular part.



Implementations

- Using OpenGL and NCLGL framework as a base.
- Dear ImGui framework for debugging, tweaking parameters at runtime and profiling.
- Different types of BRDFs are made for comparison:
  - PBR Cook-Torrance
  - PBR Disney
  - Blinn-Phong
  - Oren-Nayar + Beckmann/GGX/Gaussian
- Post Processing Effects:
  - Bloom
  - Vignette
  - SSAO
  - Invert Colour
- Gamma Correction and Tone Mapping:
  - Reinhard Tone Mapping
- Shadows:
  - Basic Shadow using PCF filtering with directional light as a light source.


Screen-Space Ambient Occlusion (SSAO)


Vignette


Invert Colour


Bloom

Implementations

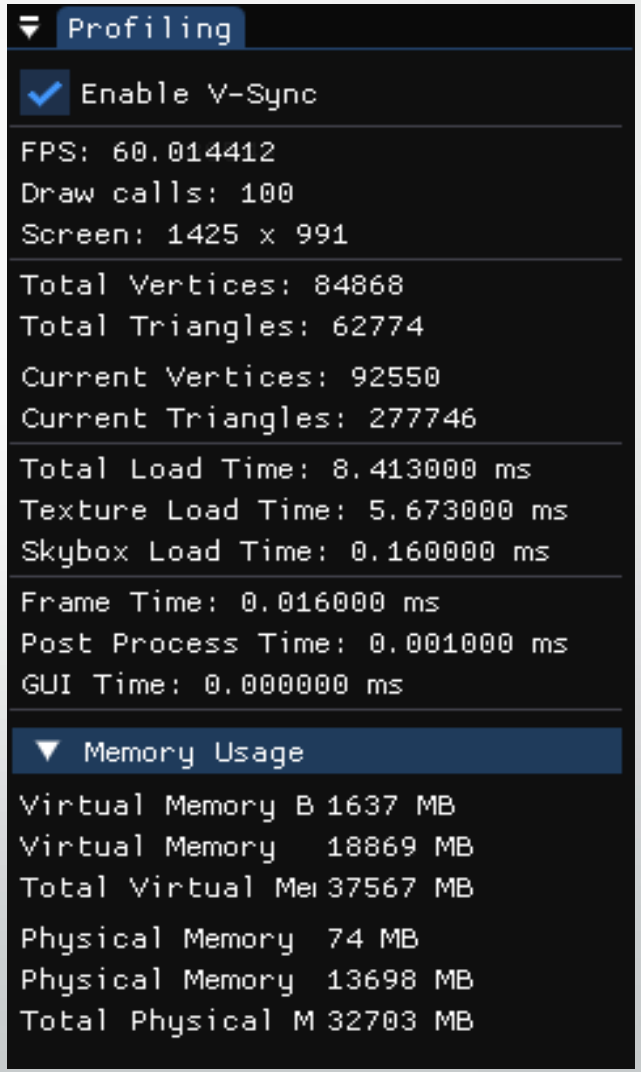
- Image-based Lighting (IBL):
  - Using HDR textures to convert to Cubemaps.
  - Exposure controllable from ImGui
  - Five different types of skybox used to show ambient lighting on surfaces
  - Diffuse Irradiance
  - Specular Prefiltering and Lookup table
- For PBR, different textures are being used:
  - Albedo: Base colour
  - Normal: Surface details
  - Metallic: Metallic or Dielectric
  - Roughness: Controls the smoothness
  - Emission: Emitting light from surface
  - Ambient Occlusion: Enhances shading
- Lights:
  - Billboard effect to show different types of light sources.
- Material System:
  - Basic material system which applies to whole mesh instead of sub-meshes.

Challenges

- Lights Performance:
  - Sending light attributes to every frame is heavy on performance.
  - To counter this, Uniform Buffer Objects (UBO) are used.
  - OpenGL version 4.20 is used for easy setting the binding points of UBOs
  - Light attributes are stored in a struct and datatype of Vector4 are used for memory alignment and padding precisions.
  - Lights UBO Data are not sent every frame. Instead, they are sent when there is a change in any of the light attributes when using ImGui.
- Uniform Buffer Objects:
  - Also used to store the view projection matrix.
  - Skybox light data like Gamma and Exposure intensity.
- Texture Loading Time:
  - PBR Textures are quite heavy to load due to size of resolutions.
  - Loading the textures in main thread is not effective.
  - To solve this, Multi-threading is used.
  - The texture's raw data are loaded using threads and once loaded the usual OpenGL texture objects are created loaded with raw data.
  - This reduced the loading time from 17 seconds to 7-8 seconds.

Analysis and Evaluation

- Profiling:
  - Dear ImGui framework is used to profiling at runtime.
  - Frames Per Second (FPS) stayed at a stable 60 FPS with V-Sync on and approximately 350 FPS when V-Sync is off.
  - Draw calls count are shown.
  - Total vertices and triangles shows the total vertices and triangles of all meshes loaded.
  - Current vertices and triangles shows the stats of meshes drawn in current screen.
  - Total load time is time taken to load the whole program start to first frame.
  - Texture load time is time taken to load the textures. Using threads significantly reduced the load time.
  - Skybox time denotes the duration of capturing and processing all five skyboxes.
  - Frame time is the time between current frame and previous frame.
  - GUI time is time for rendering of all ImGui windows.
- Memory Usage:
  - Both virtual and physical memory usage is shown.



Analysis and Evaluation

- Disney PBR vs PBR:
  - Visually not much of difference.
  - Disney diffuse uses Burley BRDF.
  - Burley BRDF is a mix of dielectric and metallic BRDF based on metallic parameter.
  - Disney provides more parameters like Sheen, Anisotropic Specular, Clearcoat.
  - Anisotropic filtering used the Ward BRDF to simulate specular like on brushed off metals.
  - Disney uses GTR for Normal Distribution.
  - Additional parameters requires more computations.

  - PBR uses Lambert diffuse.
  - PBR uses Schlick-GGX for Normal Distribution.
  - Has less control over Specular part.
  - Both use the same Fresnel-Schlick for Fresnel effect.
  - PBR is generally more light-weight when compared to Disney BRDF and is widely used in video games.


PBR Disney


PBR Cook-Torrance

Analysis and Evaluation

- Blinn-Phong vs Oren-Nayar:
  - Blinn-Phong ignores energy conservation law which results in bright colours.
  - Blinn-Phong uses lambert diffuse.
  - Blinn-Phong also ignores the metallic and roughness properties of the surface.
  - Blinn-Phong is more suitable in terms of performance and simple graphics due to being light-weight.

  - Oren-Nayar is a diffuse BRDF for handling rough surfaces on microfacet level.
  - Oren-Nayar works well on surfaces with high roughness like brick wall, wood, etc.
  - Oren-Nayar can be merged with other specular model like Torrance-Sparrow Specular model.
  - Oren-Nayar BRDF is also merged with three additional specular effects i.e. Gaussian, GGX and Beckmann.

Conclusion

- Real world lighting physics can be simulated in computer graphics with the help of Physically based rendering (PBR).
- Use of PBR or Phong shading depends on the need of the application.
- PBR, Phong, Oren-Nayar all have their pros and cons.
- Phong is more suitable for performance and simple visuals.
- Use for PBR went up as hardware progressed and demand of real life graphics is high.
- Oren-Nayar is a diffuse BRDF and more widely used in animated movies than in video games.
- Future work:
  - Implement Cascaded Shadow Maps.
  - Proper material system which applies to sub-meshes.
  - Transparent materials apply to sub-meshes and sorting them.
  - More post-processing effects.
  - Proper multi-threading to load assets using multiple OpenGL contexts.
- Links:
  - Youtube: CSC8599 – Physically Based Rendering (PBR)
  - Github: CSC8599 - Github