CPSC 314
Computer Graphics

Dinesh K. Pai

PBR and MeshStandardMaterial
File Formats

Preliminaries

- Today
  - Physically Based Rendering wrap up
  - Mesh Standard Material
  - File Formats (obj, glTF)

Physically Based Rendering

- Details of computing the intensity of light leaving the fragment (luminance) are bit complex. Some resources
  - https://marmoset.co/posts/basic-theory-of-physically-based-rendering/
  - https://substance3d.adobe.com/tutorials/courses/the-pbr-guide-part-1
- For this course we mainly need to
  - Understand the general principles
  - Know what “roughness” and “metalness” are
  - How these are used in Three.js
- Switch to tablet

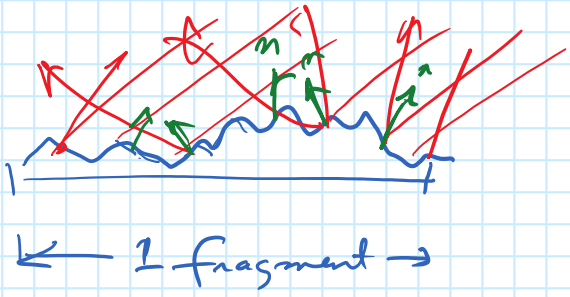
3

PBR support in Three.js

- Three.js offers two built-in PBR materials
- MeshStandardMaterial
  - Supports basic roughness and metalness
  - https://threejs.org/docs/#api/en/materials/MeshStandardMaterial
- MeshPhysicalMaterial
  - Adds clearCoat, better transparency, etc.
  - https://threejs.org/docs/#api/en/materials/MeshPhysicalMaterial

4

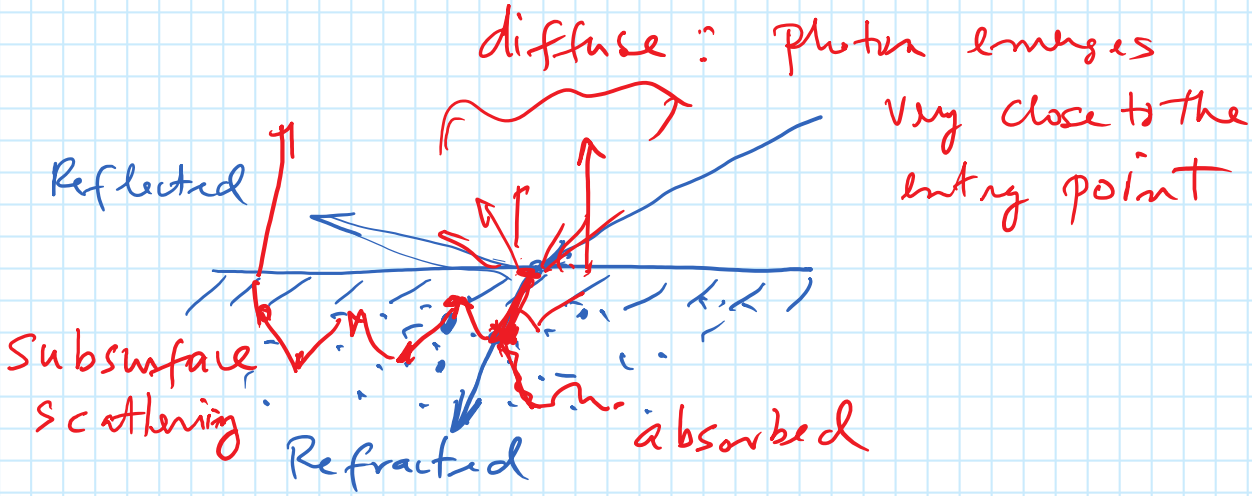
(1) Account for microgeometry



"Roughness

← 1 fragment →

(2) Account for optical properties of a surface



Metalness

Using spatially varying PBR materials

- Can be packed into a single texture
  Pass these as with other maps
- .roughnessMap : Texture
  - The **green** channel of this texture is used to alter the roughness of the material.
- .metalnessMap : Texture
  - The blue channel of this texture is used to alter the metalness of the material.

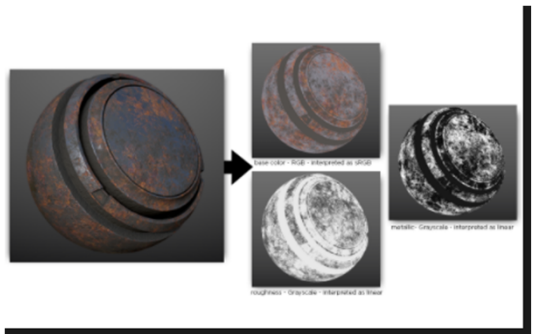


Figure 17: Metallic/Roughness workflow

https://substance3d.adobe.com/tutorials/courses/the-pbr-guide-part-2

5

Graphics File Formats from OBJ to GLTF 2.0

6

Obj Format

- Developed by Wavefront Technologies for 3D geometry
- Simple, easy to read and write. Widely used
- Old (introduced ~1990), many limitations
- Geometry only
  - Vertex position, UV, normal
  - Faces
- There have been dozens of other file formats proposed to address the shortcomings (e.g., FBX, PLY, STL, VRML, X3D, etc.)

7

.obj example https://en.wikipedia.org/wiki/Wavefront_.obj_file

```
# List of geometric vertices, with (x, y, z [,w]) coordinates, w is optional and defaults to 1.0.
v 0.123 0.234 0.345 1.0
v ...
...
# List of texture coordinates, in (u, [,v ,w]) coordinates, these will vary between 0 and 1. v, w
vt 0.500 1 [0]
vt ...
...
# List of vertex normals in (x,y,z) form; normals might not be unit vectors.
vn 0.707 0.000 0.707
vn ...
...
# Parameter space vertices in ( u [,v] [,w] ) form; free form geometry statement ( see below )
vp 0.310000 3.210000 2.100000
vp ...
...
# Polygonal face element (see below)
f 1 2 3
f 3/1 4/2 5/3
f 6/4/1 3/5/3 7/6/5
f 7//1 8//2 9//3
f ...
...
# Line element (see below)
l 5 8 1 2 4 9
```

8

glTF

- gl Transmission Format
- Introduced by Khronos – Industry consortium supporting OpenGL, WebGL, etc.
- Version 1.0 (2015) had some limitations
- Version 2.0 (2017) is much better, and is getting a lot of traction
  - Lot of industry support
  - I encourage you to start moving to it

9

- See https://www.khronos.org/gltf/ for resources
- The following are slides from Khronos's presentations https://www.khronos.org/assets/uploads/developers/library/2017-web3d/glTF-2.0-Launch_Jun17.pdf

(Switch to pdf)

10

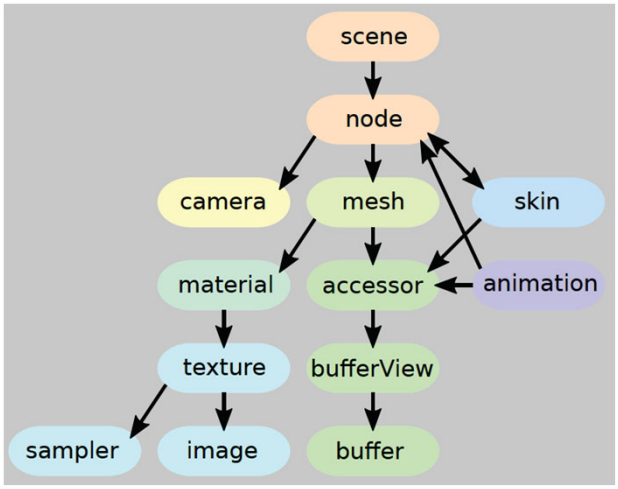
Support for glTF

- Microsoft PowerPoint (demo)
- Facebook
- Three.js can load and export https://threejs.org/examples/#webgl_loader_gltf
- VSCode
  (demo and file structure)

11

glTF structure

- A high level introduction
- Stores scene graph + textures + animation assets in a JSON file
- You now most of the pieces to understand it
- Very nice reference: https://www.khronos.org/files/gltf20-reference-guide.pdf



12