Project plan

Participants

Students

- Sudiksha Sharma

Supervisors

- Dr.-Ing. Petra Gospodnetić
- Juraj Fulir, mag. Ing.

Goal

Construct a rasterization-based image generator for industrial surfaces and explore benefits and shortcomings of such an approach.

Tasks:

- Load 3D models in a PBR rasterization engine (e.g. Blender, Godot, **UE4/5**)
- Compose rasterization materials from existing procedural path tracing materials
- Compare and incrementally improve the result quality through comparison to the ground-truth results from path tracing
- Explore usage of anisotropic material shaders for more consistent results
- Explore usage of pre-computable light simulation techniques for more realistic results
- (reflection captures, GI probes, RTRT)
- Encapsulate the pipeline into a configurable image generator

Timeline

21.11.2022. - 31.3.2023. ~ 19 weeks

1. **Get acquainted with basics of UE (*2 weeks: 21.11. - 2.12.*)**
    a. Scene hierarchy and management
    b. Object loading
    c. Light setup and building lightmaps
    d. Material definition and usage
    e. Blueprint for object and scene programming
    f. Finding resources to help with solving specific problems
2. **Prepare the inspection scene (*3 weeks: 2.12. - 23.12.*)**
    ~~a. Load and set material for inspection light source~~
    b. Setup a simple light source
        ~~i. Spot light with doughnut material function and large source radius~~
        ii. Area light with material function is probably better
    c. Load the target mesh and place it into the scene
    d. Write a simple target materials using existing OSL shader as a reference
3. Setup the appropriate anti-aliasing settings for raytracing (*3 weeks: 26.12. - 13.1.*)
    a. Investigate anti-aliasing techniques used in UE's new hybrid pipeline
    b. Experiment with techniques and parameters (MSAA/spp) to remove aliasing as much as possible
    c. Try finding good parameters to reduce the cost of rendering while keeping it close to real-time
4. Program the image acquisition blueprint (*3 weeks: ~~26.12. - 13.1.~~ 16.1. - 3.2.*)
    a. Define several camera poses as array for testing purposes
    b. Program a method for traversing camera poses
    c. Export each image into a structured directory (ExportRenderTarget)

~~5. Parse viewpoints from a file (*3 weeks: 16.1. - 3.2.*)~~
    ~~a. Setup C++ environment for UE~~
    ~~b. Define viewpoint struct and proper trans. matrix calculation~~
    ~~c. Write file loader and parser~~
    ~~d. Expose modules to blueprints~~
    ~~e. Update image acquisition to use the new modules~~
    ~~f. Generate images~~

6. Translate more materials into UE (*1 week: 6.2. - 10.2.*)
7. Compare images to path traced targets (*1.5 weeks: 13.2. - 22.2.*)
    a. Write a small script in Python to load pairs of images and compares them
    b. Describe where are the differences most prominent
8. Increase image quality (*4.5 weeks: 23.2. - 24.3.*)
    a. Reflection capsules
    b. GI nodes
    c. Material anisotropy
    d. RTRT
    e. ...

9. Collect results and write the final project report (*1 week: 27.3. - 31.3.*)