**RTGI TODO lists**

**Remember guys, our top priorities are reading papers and doing research work for this project. Not game engine programming☺!!!**

**Pick up whichever task you are interested in and feel free to add new research topics! Happy doing GI research!**

**Research TODO list** **(Priority, lower number is higher):**

|  |  |  |
| --- | --- | --- |
| **Technique** | **Description** | **Priority** |
| **~~Temporal Coherence~~** | **~~[Tokuyoshi 2012] uses it. If we want to compare our result with theirs, we have to implement it in our rendering framework.~~** | **Cancelled** |
| **~~Interleaved Sampling~~** | **~~Is G-buffer splitting necessary? Maybe we can follow SSDO’s technique, in which they do interleaved sampling in place. See SSDO for details.~~** | **Done** |
| Scene BV computation | We only want to voxelize and create global ray-bundles for the visible part of the scene, which means we can compute the visible scene BV by using G-buffer’s world position texture and compute shader. | 1 |
| **~~Global ray-bundles~~** | **~~Investigate if it is possible generating ray-bundles using just voxel grid instead of the original scene.~~** | **Cancelled** |
| **~~Efficient GPU ray marching~~** | **~~Currently I am using a naïve ray marching, try to use an efficient 3D-mipmap or octree-based data structure to accelerate the ray marching procedure.~~** | **Done** |
| **~~SVO for voxel grid~~** | **~~Cool technique worth studying~~** | **Done** |
| Specular indirect sampling | Do we follow SVO’s way sampling indirect illumination?  Or can we come up with a new idea using our framework? | 2 |
| VPL Shadow Maps | [Tokuyoshi 2012] uses it. If we want to compare our result with theirs, we have to implement it in our rendering framework. | 1 |
| **~~HDR tone mapping~~** | **~~In order to create beautiful images, we should store intermediate data (colors) in HDR format, just as the SSDO demo. Then in the final image composition, we have to do a high quality tone mapping.~~** | **Done** |
| **~~Geometric-aware filtering~~** | **~~Since we use interleaved sampling, we must implement a filter that removes the noises.~~** | **Done** |
| Adaptive Caustic Maps | Can we integrate an improved version of ACM into the framework using GPU per-pixel link list? | 3 |
| Sub-surface Scattering | Can we integrate an SSS into the framework? | 3 |
| **~~Sub-rendering pass timing~~** | **~~We want to time all techniques mentioned above.~~** | **Done** |

**Engineering TODO list (Priority, lower number is higher):**

|  |  |  |
| --- | --- | --- |
| **Functionality** | **Description** | **Priority** |
| **~~Support complex scenes~~** | **~~In our final demo, we want to render not only cornel box, but also a complex indoor scene with several dynamic objects and lights.~~** | **Done** |
| **~~Rendering API wrapper~~** | **~~Currently being carried out by Che Sun. The goal is to hide OpenGL API as much as possible from application level. Should be finished by the end of December.~~** | **Done** |
| Design material script and uber shader system. | Apparently this is a rendering engine problem | 4 |
| **~~Support mesh animation~~** | **~~Say, a running elephant using GPU matrix blending.~~** | **Done** |
| Shader uniform reflection | When the shader loader parses a program, exam each used shader uniform or uniform buffer then separate them into system and user groups. Uniforms in system group can then be set automatically. This is not an issue for now. | 4 |
| Uniform buffer management | Manage shader uniforms based on per-frame, per-object-per-frame, per-object-per-pass level, reduce the cost of upating shader uniforms. This is not an issue for now. | 4 |