Rendering Tutorial 2

Priv.-Doz. Dr. Ing. Martin Lambers

Assigment 1

Implement Motion Blur for the brute force path tracer, as discussed in the lecture.

Code that implements Quaternions and Transformations is given in the tutorial material.

It is convenient to be able to skip animations in the definition of surfaces and cameras because usually only a few things are animated and most are static. For example:

```
class SurfaceSphere : public Surface
{
    ...;
    const Animation* animation;

    SurfaceSphere(..., const Animation* anim = nullptr) :
        ..., animation(anim)
    {}

    virtual HitRecord hit(const Ray& ray, float amin, float amax) override {
        ...;
        if (animation) {
            Transformation T = animation->at(ray.time);
            // apply transformation...
        }
        ...;
    }
};
```

Reproduce the animation examples from the lecture:

- The falling sphere animation is a translation by offset $(0, -0.15 - t \cdot 0.1, -3.5)$.
- The camera animation defines two key frame animations using the same approach as `gluLookAt()`, with an eye position and a position that defines the view direction, and then interpolates between the two:

  ```
  vec3 center = vec3(0.0f, 0.0f, -5.0f);
  vec3 eye0 = vec3(0.0f, 0.0f, 0.0f);
  vec3 eye1 = vec3(0.0f, 3.0f, -3.0f);
  Transformation T0 = Transformation(eye0, center);
  Transformation T1 = Transformation(eye1, center);
  return mix(T0, T1, t);
  ```

  Instead of creating an image with motion blur, we create a set of images at fixed time points (e.g. $t \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$) using different camera poses.

Assignment 2

Implement support for Textures for the brute force path tracer, as discussed in the lecture.

Code that implements `TextureImage` and several other things is given in the tutorial material.

Reproduce the texture example from the lecture with the `earth.jpg` texture on a sphere with center $(0, 0.3, -4)$ and radius 0.5. Add an animation to the sphere that rotates it a few degrees around the y or x axis.

Note: add the following two lines to `pathtracer.cpp` (*after* including the project-specific headers) in order to get the function definitions of `stb_image.h`:

```
#define STB_IMAGE_IMPLEMENTATION
#include "stb_image.h"
```