Global illumination

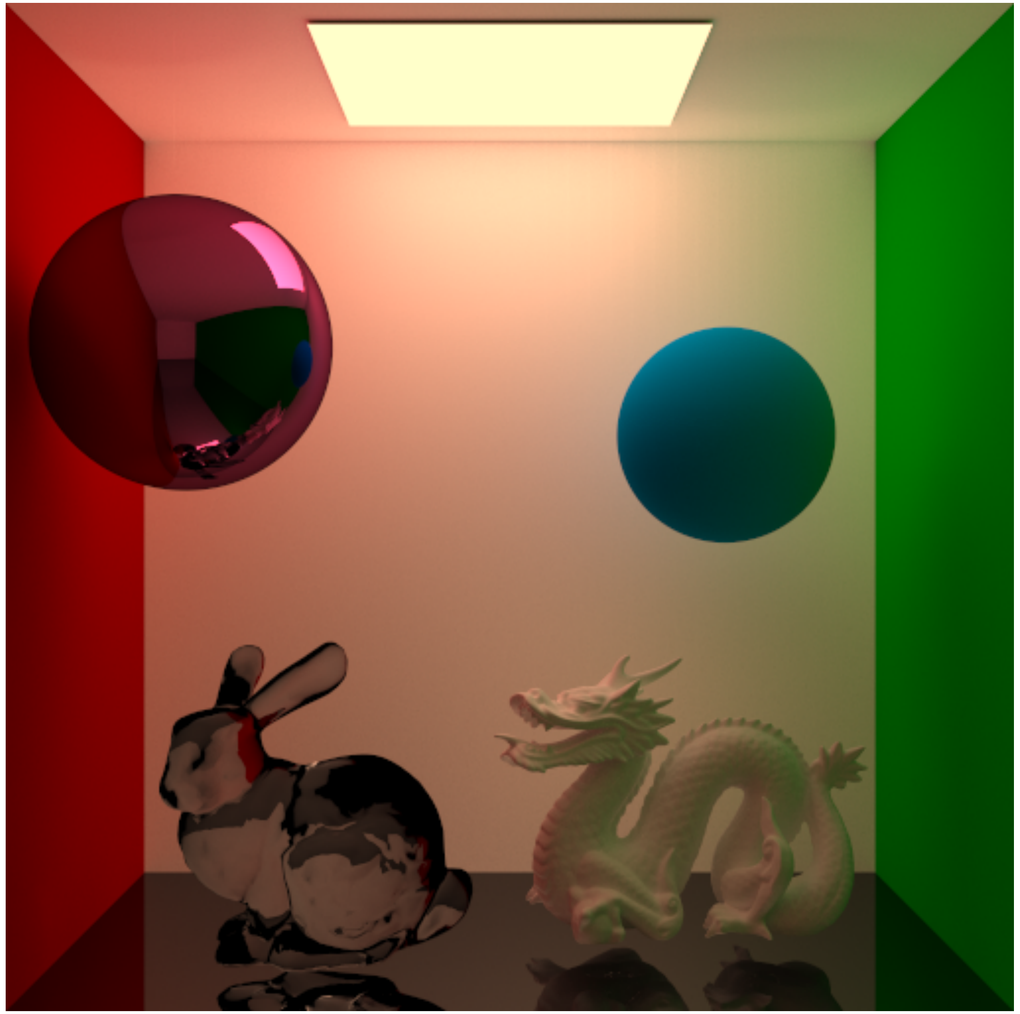
Homework 4 Code Skeleton & Implementation

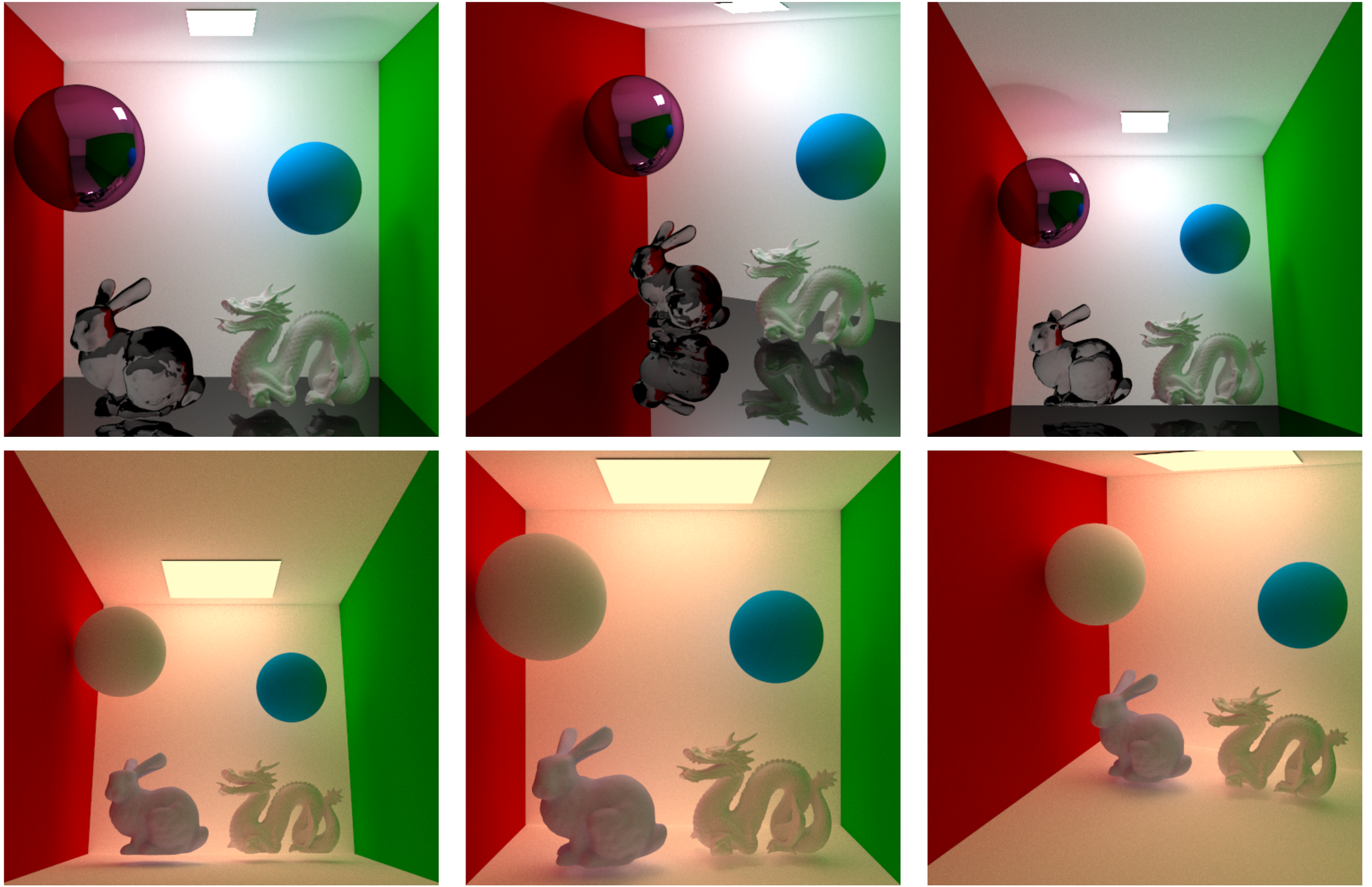
Yuehao Wang; May 6

Homework 4

- Deadline: 22:00, May 18, 2021
- Requirements:
  - Implement ray-mesh intersection test with uniform grids for acceleration.
  - Implement the diffusion BRDF with area light source.
  - Implement Monte-Carlo path tracing algorithm for global illumination: direct + indirect lighting.
- http://faculty.sist.shanghaitech.edu.cn/faculty/liuxp/course/cs171.01/assignment/4/assignment4.html

Homework 4 Gallery





Code Structures

- accel_struct.hpp: defines some data structures for acceleration, e.g., AABB
- brdf.hpp: defines classes and interfaces for BRDFs, e.g., IdealDiffusion
- geometry.hpp: TriangleMesh (New), Sphere
- integrator.hpp: PathTracingIntegrator (New)
- interaction.hpp: define a data structure used for recording intersection info
  - New properties: wi (direction of incoming radiance), wo (direction of outcoming radiance), material (BRDF at the intersecting point);
- light.hpp: defines classes for modeling lights, e.g., AreaLight
- utils.hpp: define utility functions (samplers, transformation)

Interfaces to Implement

- TriangleMesh::raySingleTriangleIntersection
  - Test whether the given ray is intersected with the triangle (v0_idx, v1_idx, v2_idx).
  - bool raySingleTriangleIntersection(Interaction& interaction, const Ray& ray, int v0_idx, int v1_idx, int v2_idx) const;

Interfaces to Implement

- TriangleMesh::buildUniformGrid
  - Build a uniform grid of the triangle mesh: create a data structure to store cells, add triangles to the corresponding cell.
  - void buildUniformGrid();
- TriangleMesh::rayIntersection
  - Check whether the triangle mesh is intersected with the given ray: find cells that the ray will go through, perform intersection test between the ray and triangles in the cells.
  - virtual bool rayIntersection(Interaction& interaction, const Ray& ray) override;

Interfaces to Implement

- IdeaDiffusion::sample
  - Sample a direction according to the BRDF, store the sampled direction in the given interaction. Also, the PDF of this sample should be returned.
  - virtual float sample(Interaction& interact);
- IdealDiffusion::samplePdf
  - Compute the PDF of the given BRDF sample at the specified interaction. You may need to use the Interaction.wi and Interaction.wo.
  - virtual float samplePdf(const Interaction& interact);
- IdealDiffusion::eval
  - Evaluate the BRDF, namely, return the BRDF value at the given interation
  - Eigen::Vector3f eval(const Interaction& interact);

Interfaces to Implement

- AreaLight::emission
  - Get the emission at the specified position along the given direction.
  - virtual Eigen::Vector3f emission(Eigen::Vector3f pos, Eigen::Vector3f dir);
- AreaLight::sample
  - Sample a position on the light and obtain the corresponding PDF.
  - virtual Eigen::Vector3f sample(Interaction& ref_it, float* pdf = nullptr);
- AreaLight::samplePdf
  - Compute the PDF of the given light sample.
  - virtual float samplePdf(const Interaction& ref_it, Eigen::Vector3f pos);

Interfaces to Implement

- PathTracingIntegrator::render
  - Interface of rendering the whole image. Go through each pixel on the film, generate corresponding rays, and compute the radiance by calling the PathTracingIntegrator::radiance method.
  - virtual void render() override;
- PathTracingIntegrator::radiance
  - Compute the radiance brought by the given ray.
  - virtual Eigen::Vector3f radiance(Ray ray) override;
- You may need to add your own functions.
- Recall the Tutorial 9!

Notes for Sampling

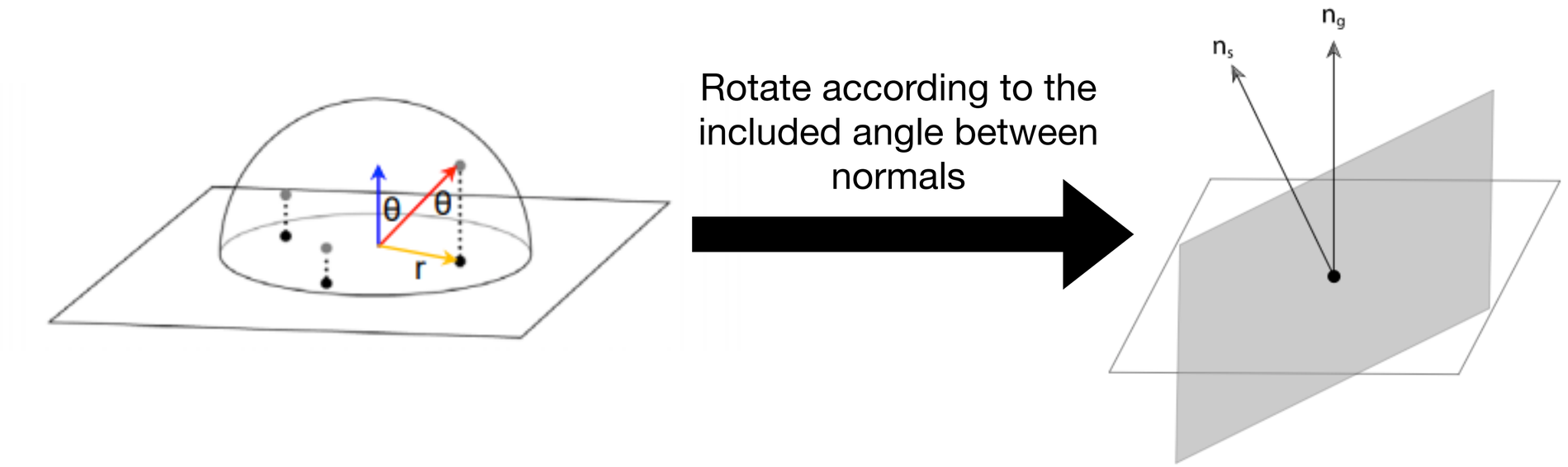
- Uniform sampling Unif(a, b) in modern C++:

```
#include <random>
#include <algorithm>


std::default_random_engine random_generator;
std::uniform_real_distribution<float> dis(a, b);
float random_val = dis(random_generator);
```

Notes for Sampling Directions

- The original sampled directions are relative to the local coordinate systems.
- You need to transform samples to the world coordinate systems.
- Useful function: Eigen::Quaternionf::FromTwoVectors(src, des)



Thanks!

Q&A