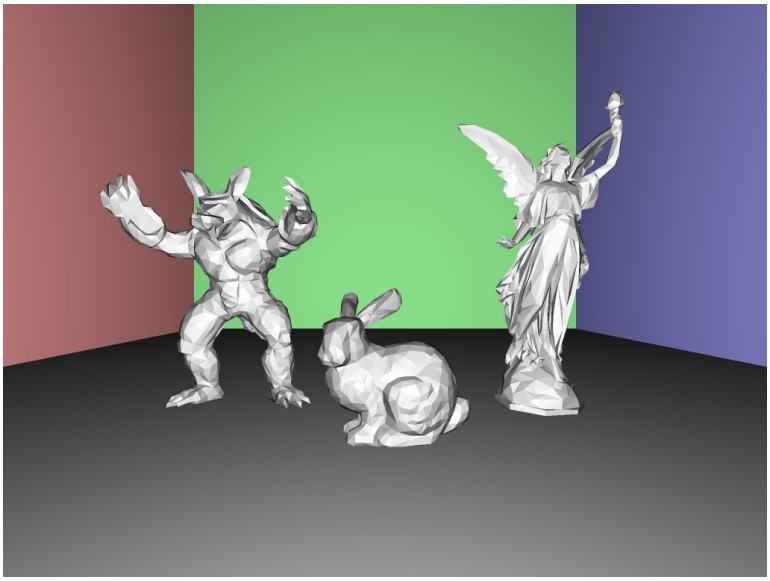
Università della Svizzera italiana

Faculty of Informatics

Computer Graphics (Fall 2023)

Bonus assignment : Triangle meshes

October 5, 2023



Let's render some iconic models from computer graphics! This is an extra assignment from which you can obtain bonus points and will start taking your renderer to an entirely new level. Note that some of the future bonus exercises will depend on this one. Also, what you implement will contribute to the Rendering Competition. While you are welcome to implement the functionalities listed here anytime, you will obtain points only if you submit the solution by the indicated deadline.

Updated framework

There is no update to the framework. Implement the required functionalities the way you want. **You are not allowed to use any external libraries or snippets of code downloaded from the internet**.

Exercise 1 [7 points]

Extend your raytracer to render meshes from OBJ[1] files. To this end, you will need to introduce a new object to the raytracer (triangle/mesh), implement a ray-triangle intersection, and mesh loader. OBJ files are very versatile and can store a lot of different information. You need to implement only the functionality that you need for this assignment.

You are given three meshes that you can use to demonstrate the new raytracer capabilities. They come from Stanford repository[2], but are decimated and adjusted to make your tasks slightly easier. There are two versions for each mesh, with and without per-vertex normal vectors. The meshes with normal vectors are for smooth shading (see the next exercise). You are also invited to look for other meshes and try to render them. Note that once you introduce meshes, your raytracer will become slow. For the reference, the above image was rendered in 2048×1536 resolution, and it took approximately one hour to render with our current raytracer. In the future, we will study how to make it faster. For now, we recommend that for testing and placing

[1] https://en.wikipedia.org/wiki/Wavefront_.obj_file

[2] http://graphics.stanford.edu/data/3Dscanrep/

objects, you render very low-resolution images, and only for the final result you increase it. The meshes are located at point $(0, 0, 0)$. Therefore, you will need to translate the objects away from the camera location. You can position objects as you want as long as the objects are well visible in the image.

Exercise 2 [3 points]

Implement smooth normal interpolation using barycentric coordinates and demonstrate the effect.

Submission

Your submission **must** contain only one ZIP-file containing:

- a readme file with information which exercises you solved, the authors of the solutions, and an explanation of encountered problems, if any,
- one `result.ppm` file generated by your code after solving the exercises,
- a folder named `code` which contains all the files required to run your code, in particular there should be `main.cpp` file which will be used to compile your ratracer with the command `g++ main.cpp`.

The ZIP-file you submit **must** be of a form `surname1_surname2_Assignment1.zip` for team of two, and `surname_Assignment1.zip` for a single submission. Only one person from the team should submit the solution. **You MUST follow the above rules when submitting the assignment.**

Solutions must be returned on October 31, 2023 via iCorsi3