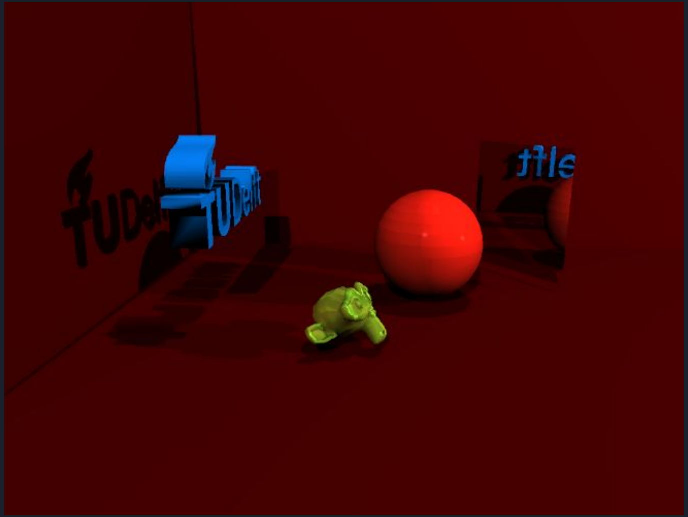
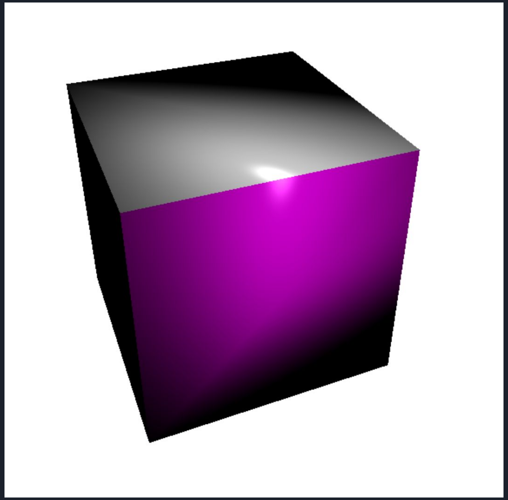
InstaTracer

Group: 48, CSE2215

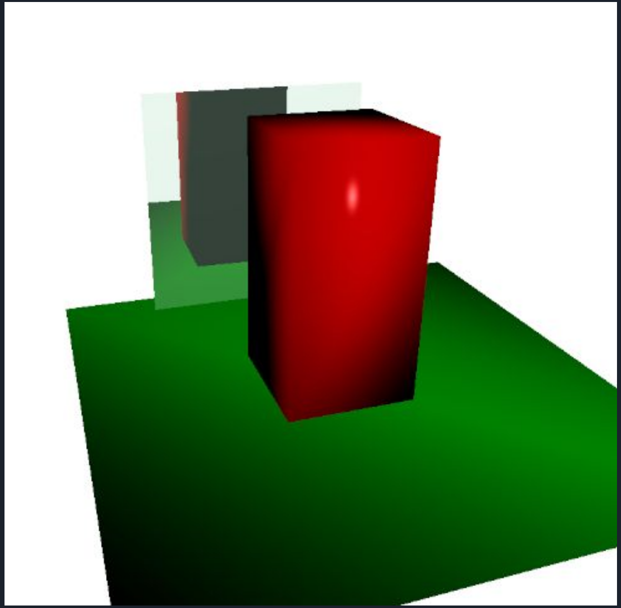
Sina Şen
Paco Pronk
Marilotte Koning
Joran Heemskerk
Julian Biesheuvel
Kevin Nanhekhan



Compute shading at the first impact point (diffuse use and specular).



Perform recursive ray-tracing for reflections to simulate specular materials.



Calculate hard shadows from a point light.

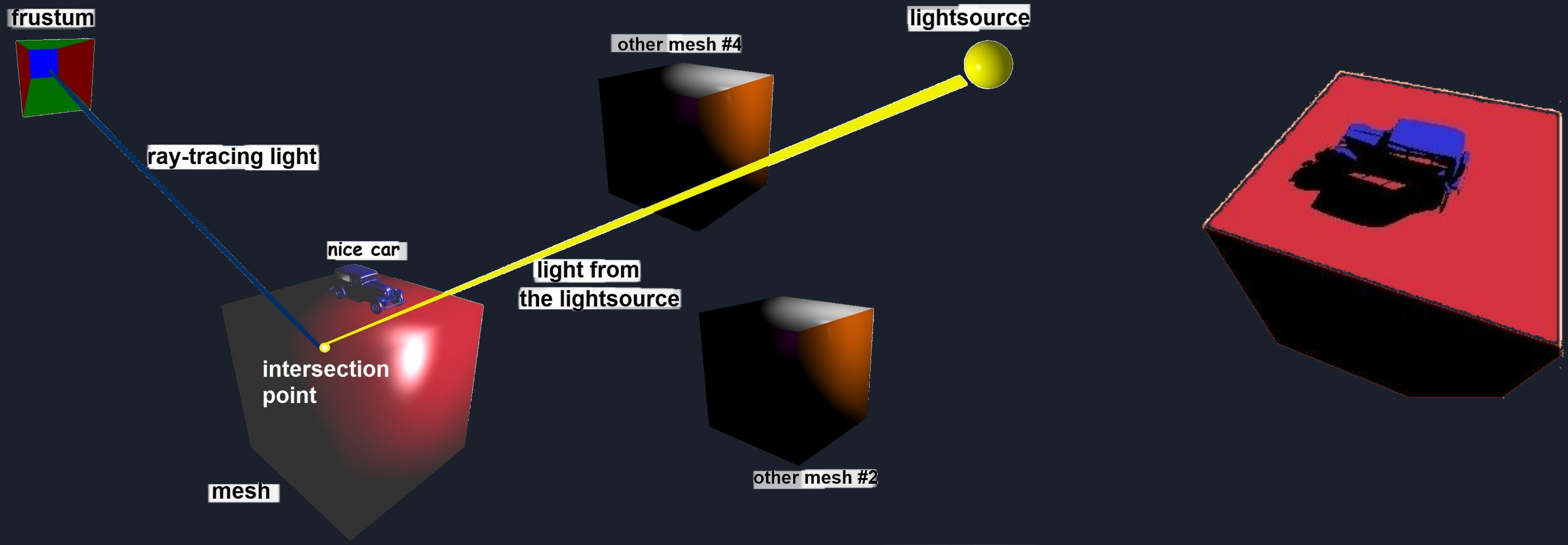
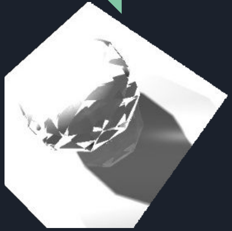
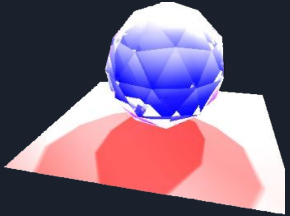


image created by us

Calculate soft shadows from a spherical light centered at a point light.





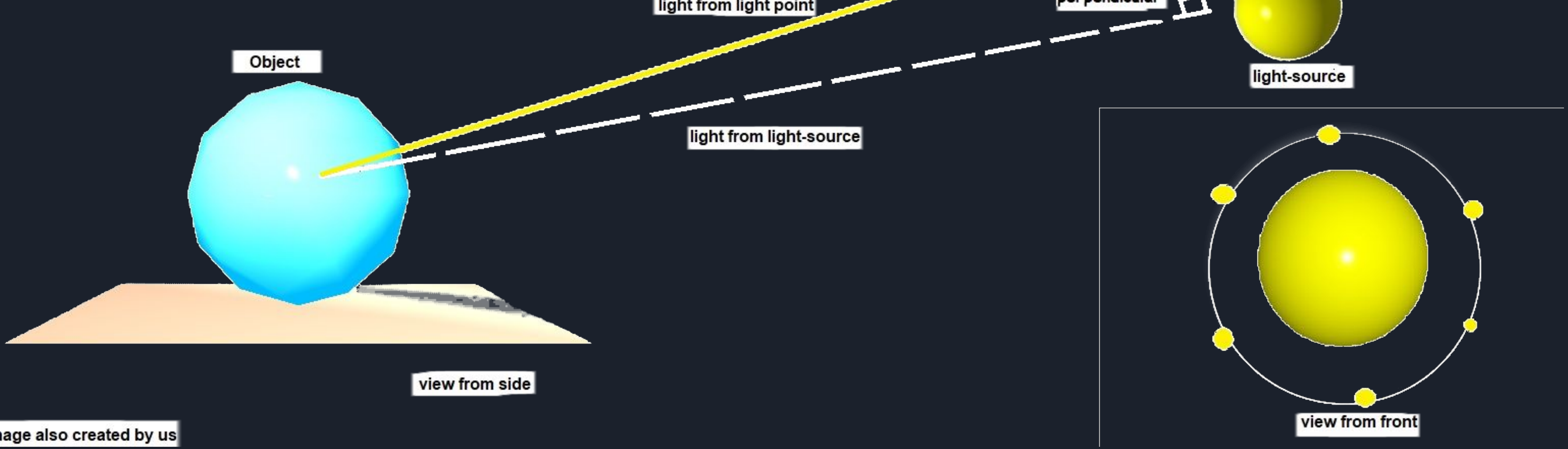
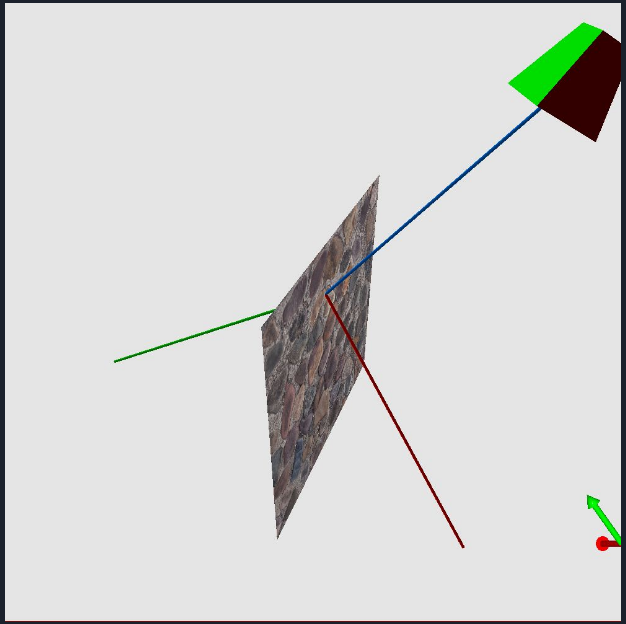
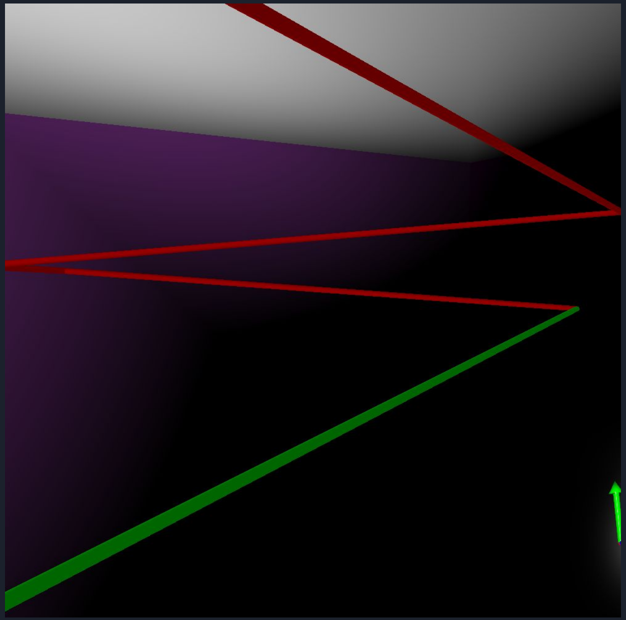


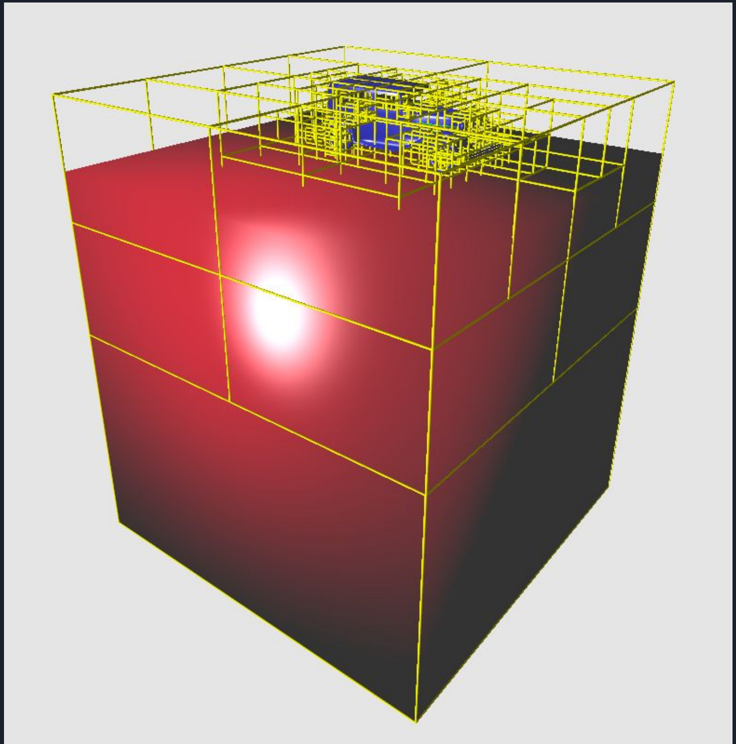
Image also created by us

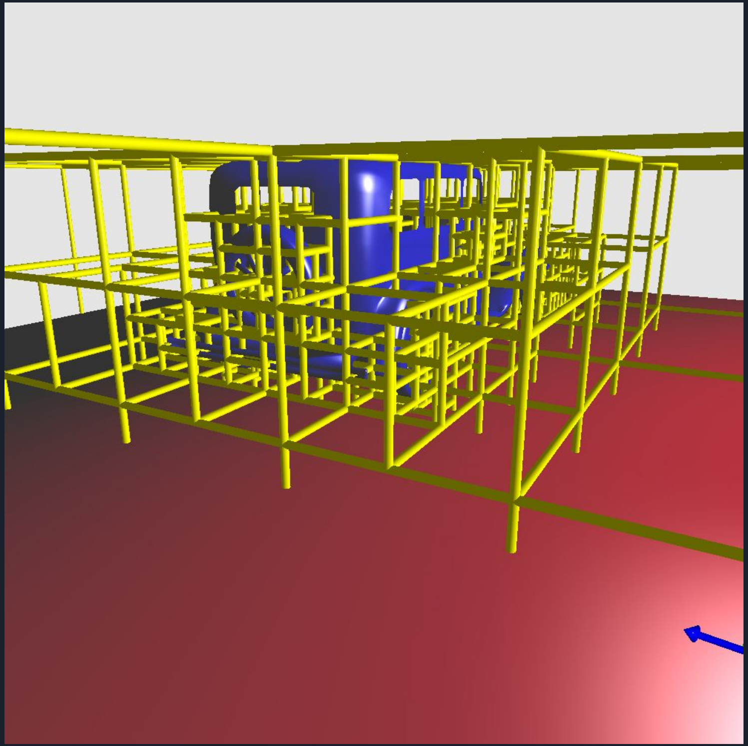
Show an interactive display in OpenGL of the 3D scene and a debug ray tracer.



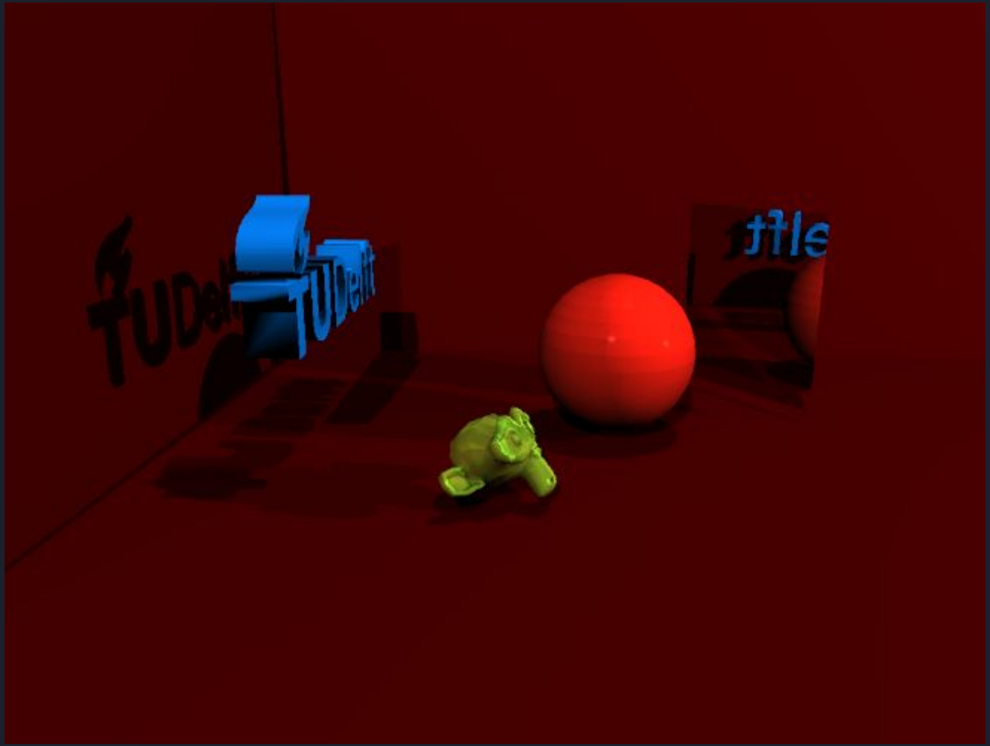


Implement a (simple) acceleration structure

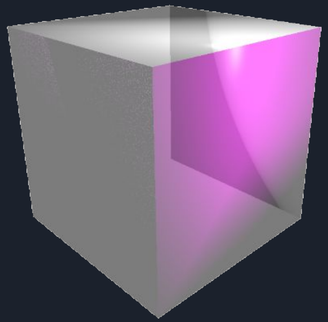




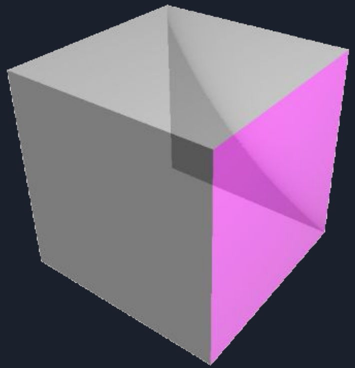
Show a scene created by the group, and directly loaded into the application.



Utilizing interpolated normals to smooth objects.

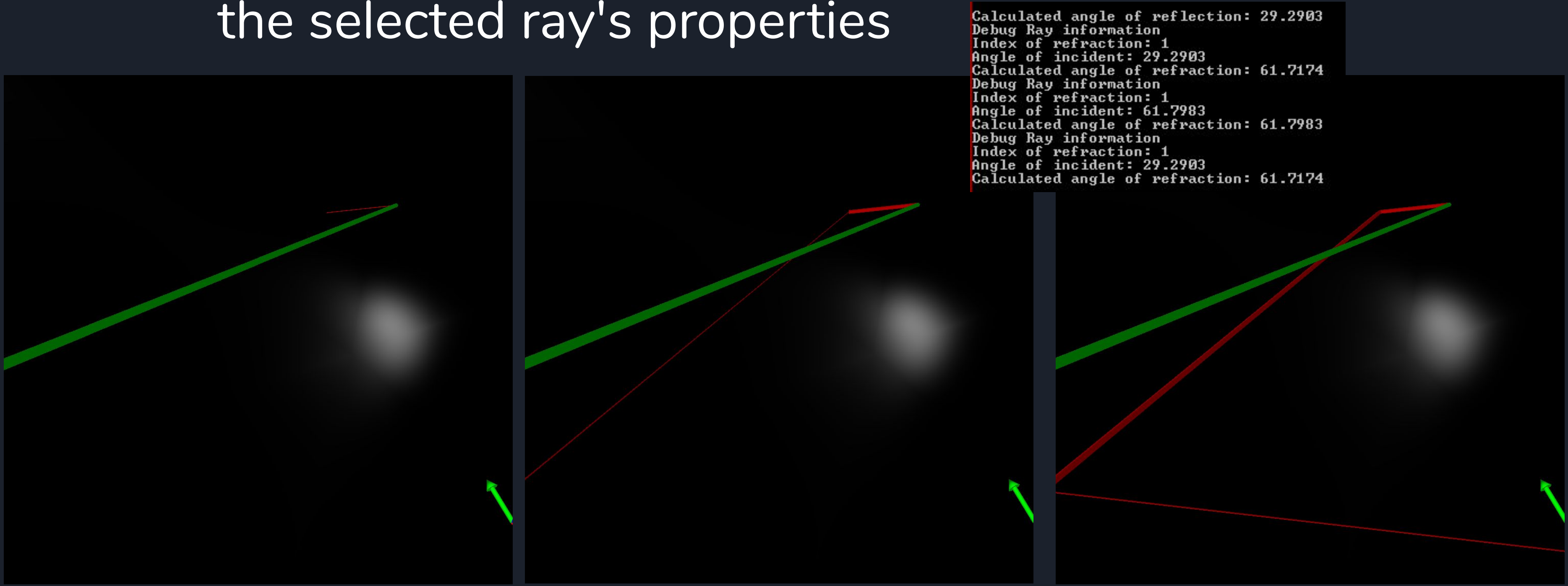


Interpolated Normals

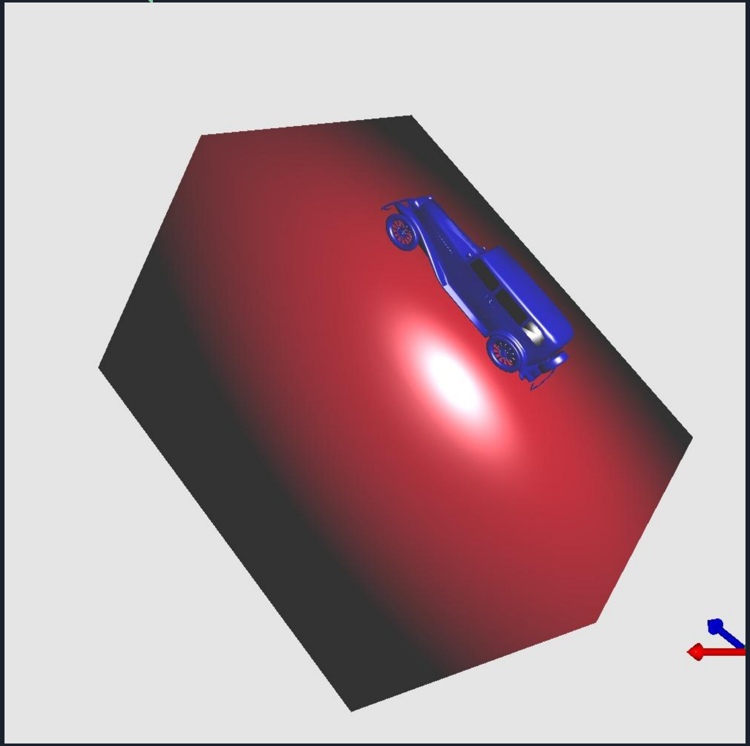


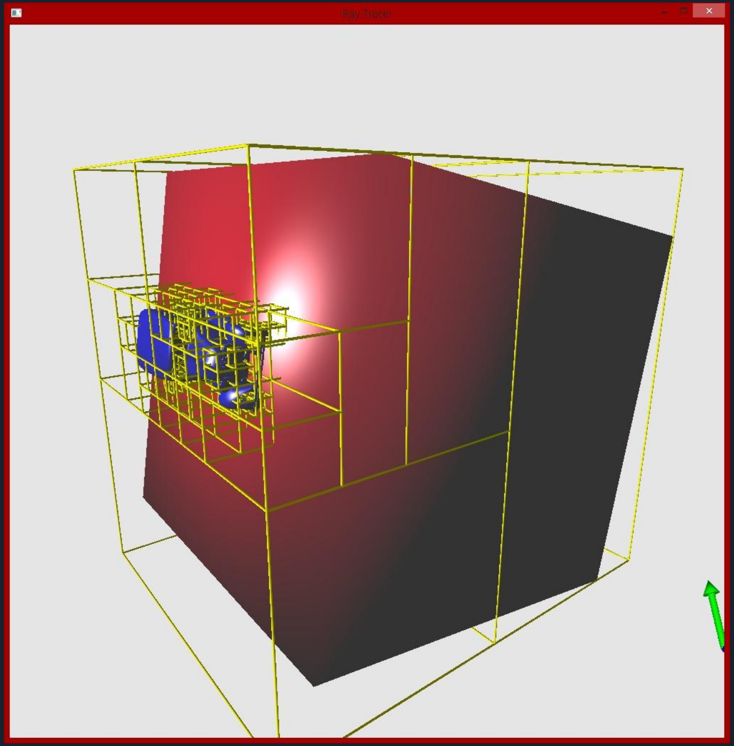
Not Interpolated Normals

Extending the debugger to show the nth reflection of a ray via the key-board, or triggering a ray highlighting and showing command line output of the selected ray's properties



Allowing modification of triangles within the ray tracer. Press G (translate), H (scale), P (rotate)





Numerical Evaluation of the ray tracer

Numerical Evaluation of Ray Tracer of Group 48

Sina Sen

***Abstract*—In this document, we explored how our ray tracer performs given different scenes. Our primary criterion is the render time that our ray tracer takes to render the scene we give as the input. We use different scenes for this. (*Abstract*)**

***Keywords*—component, formatting, style, styling, insert** (*key words)*

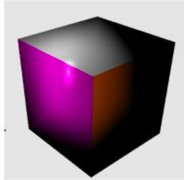
I. Introduction

In the research paper published by Utah University[1], it is said that "We observe that even for smaller scenes, that can essentially fit into cache, memory still is the highest contributor in energy and latency, suggesting that even in a case of balanced compute workload, compute remains inexpensive". So primary thing that affects the rendering speed of the raytracer is apparently the memory of the device which run our ray tracer on. To make sure we will get the best result, we ran all of our scenes in the best laptop our group had. So, the results of this paper will probably not be the same on any other computer. For the evaluation, the main parameter we are using is the relationship between the rendering time of the scenes vs. the number of faces our scene has in total. We will also sketch some graphs to evaluate this relationship more visually and provide the rendered image of the scene for accuracy. We are also running all of the scenes in 1000x1000 resolution

II. Computer Specs

This is the original scene created by us, it contains multiple light sources, hard and soft shadows, refraction and reflection and shading that slows down the rendering.

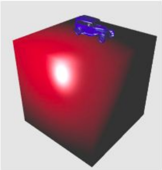
Scene 2



Number of faces: 12, Rendering Time 6.839 seconds

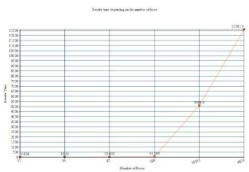
Scene 3

Scene 4



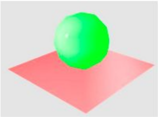
Number of Faces: 16311, Rendering Time: 5086.4 seconds

Graph 1: Rendering time vs. number of faces



(error, point 4823 should be 1258.23, not 12582.3)

Scene 5



Number of Faces: 82, Rendering Time: 33.403 seconds

Number of Faces	*Rendering Time*
12	6.839
16	13.34
82	33.403
184	61.194
4823	1258.23
16311	5086.4

Fig. 1. Number of Faces and Rendering Time

III. Evaluation

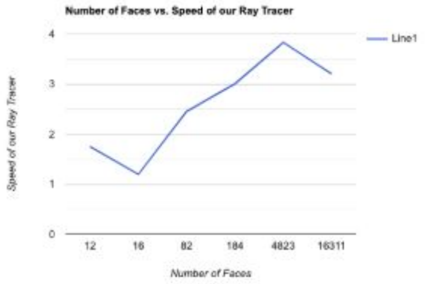
As one can see from the graph and from the table, it is apparent that the rendering time increases as the number

Let's try this for our other data points as well:

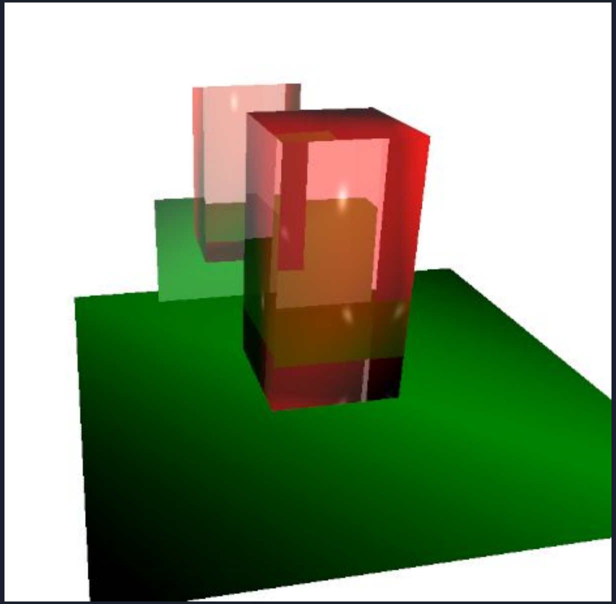
12/6.839 = 1.754
16/13.34 = 1.199
82/33.403 = 2.454
184/61.194 = 3.006
16311/5086.4 = 3.206

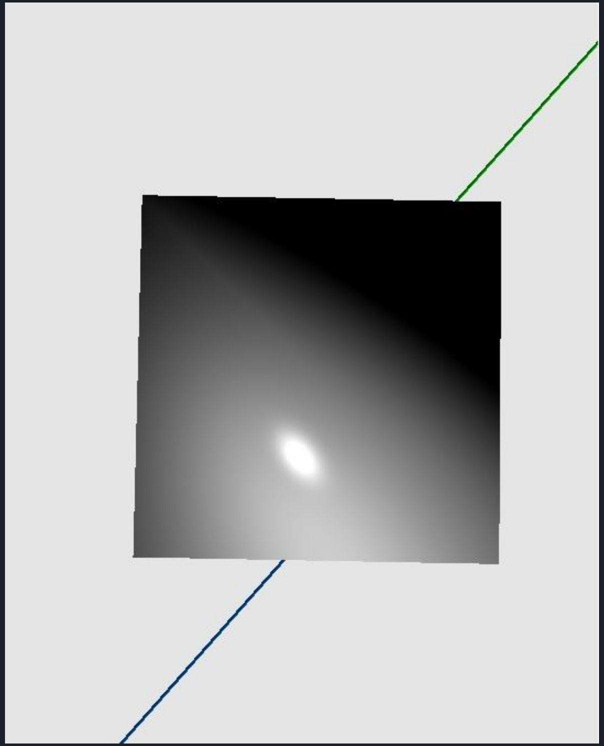
Average speed = 2.575

So, there's a easily recognizable trend going on in these calculations. As the number of faces (triangles) increase, the speed of our ray tracer (rendered faces per second) increase, with only one exception, which can be interpreted as it is a consequence of the complexity of the scene . To visualize this:



Supporting refraction and the display of transparent objects.

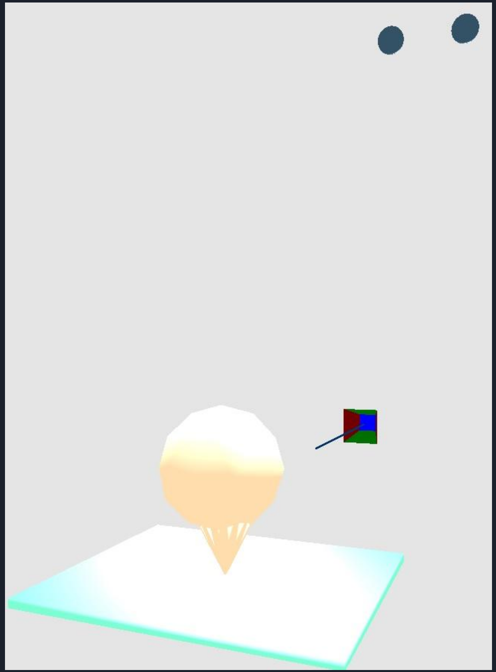




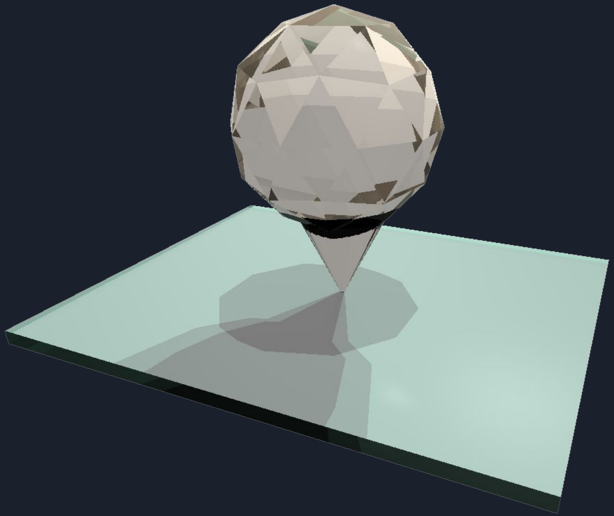
Supporting soft shadows and other types of light sources.

In previewer

Multiple light sources



result



Bonus Slide: Amazing sketch

Made in paint