Project 1: Ray Tracing

# 1 Goal

Implement the ray-tracing algorithm covered in class. Your programs should render triangles and spheres illuminated by a single white light source (with no attenuation).

Ideally, your programs have to compile and run in the Alamode Lab. If you are very uncomfortable with the lab's LINUX, feel free to use another environment. In any case, please follow the submission instructions in Section 6.

This is an individual project. You are free to discuss high level implementation ideas with others but the code has to be your own. In particular, this means that you have to implement all the vector calculations and intersection routines yourself. In addition, you are not allowed to share the output images for any of the test input files. We will follow the MACS Department programming project guidelines outlined in the last section of the Syllabus.

# 2 Input file format

The input file (assume its name is 'input.txt' and it's placed in the current directory) will specify the view, window size, light source and the primitives (triangles and spheres), in the following order:

**1.** The desired resolution of the output image (two integers, $m$ and $n$ if the output is to be $m \times n$)

**2.** The coordinates of the viewpoint $e$ (three floats)

**3.** The screen data: $l$ (lower left corner), $\vec{h}$ and $\vec{v}$ (vectors running along the horizontal and vertical edges)

**4.** The light source ($b$, the coordinates and $I$, the intensity)

**5.** The ambient light intensity

**6.** Number of primitives

**7.** List of primitives (spheres and triangles)

The list of primitives will consist of a number of primitive records. A primitive record will start with a letter, $S$ for a sphere and $T$ for a triangle. For a sphere, we shall then list the coordinates of the center and the radius (4 floats) in one line and the material description (8 floats): $k_{dr}$, $k_{dg}$, $k_{db}$, $k_{ar}$, $k_{ag}$, $k_{ab}$, $k_s$ and $n_{\text{spec}}$. The first three are the diffuse reflection coefficients for red green and blue components, then come the ambient reflection coefficients and the specular coefficient and exponent. Note that we won't have separate RGB components for the specular coefficient: we'll just use the same value for all of them (i.e. set $k_{sr} = k_{sg} = k_{sb} = k_s$). A triangle record will start with a letter $T$ and contain a list of the coordinates of each of the three vertices (9 floats) and the material properties (in the same form as for spheres).

The input files will be designed so that the intensity of a pixel should never exceed 1 (except possibly for a tiny amount due to numerical error).

Thus, starting portion of the file will look like this:

$m$ $n$
$e_x$ $e_y$ $e_z$
$l_x$ $l_y$ $l_z$
$\vec{h}_x$ $\vec{h}_y$ $\vec{h}_z$
$\vec{v}_x$ $\vec{v}_y$ $\vec{v}_z$
$b_x$ $b_y$ $b_z$ $I$
$I_a$
$N$

Then, there will follow the description of the primitives. For a triangle, we will have:
T
$a_{1x}$ $a_{1y}$ $a_{1z}$
$a_{2x}$ $a_{2y}$ $a_{2z}$
$a_{3x}$ $a_{3y}$ $a_{3z}$
$k_{dr}$ $k_{dg}$ $k_{db}$ $k_{ar}$ $k_{ag}$ $k_{ab}$ $k_s$ $n_{\text{spec}}$

And for a sphere:
S
$c_x$ $c_y$ $c_z$ $r$
$k_{dr}$ $k_{dg}$ $k_{db}$ $k_{ar}$ $k_{ag}$ $k_{ab}$ $k_s$ $n_{\text{spec}}$

# 3 Output

A binary PPM file named 'output.ppm'. Use 255 as the maximum color value. See the sample code or http://netpbm.sourceforge.net/doc/ppm.html for more details.

# 4 Grading

100 points maximum. Things we will look at:

**1.** Visibility (do correct object show up at the right pixels): 40

**2.** Shadows: 20

**3.** Illumination: 20

**5.** Two mid-project reports: 20 (10 each)

# 5 Mid-project reports

The main purpose of the reports is to help you stay on track with the project and to enforce a step-by-step approach. The deadline for the mid-project reports is sharp: late submissions will get no credit.

You are required to submit two reports and each will be worth 10 points. Each of the reports should contain:

- a short paragraph (a few sentences) describing your progress

- the output of your code for the provided test data (all 19 files)

- source code with instructions on how to compile and run it; please follow the directions in Section 6.

Minimum for full credit is as follows:

**Report#1.** Rendering objects with no illumination: use the diffuse coefficients $(k_{dr}, k_{dg}, k_{db})$ of the primitive visible through a pixel as the red, green and blue components for that pixel.

**Report#2.** Implementing illumination formula without shadows: use the illumination formula with the material properties in the input file to compute color, but ignore shadows.

If you are ahead of time, you can submit output images with more features than required (e.g. with illumination with Report #1).

# 6 Turning in your code

Submit a compressed tarfile containing:

**(i)** everything we need to compile your code (source, makefile, project files) and

**(ii)** the output images for the test input files.

If you use the Alamode lab, submit a file named `YourFamilyName.tgz`, where YourFamilyName is, of course, your family name. Here is what we are going to do:
`$ tar xfz YourFamilyName.tgz`
This should create a directory 'YourFamilyName' with all your files. Then, we'll do
`$ cd YourFamilyName`
`$ make`
This should compile your code. The executable should be named 'raytrace'. Now, we'll place the input file in the currenty directory (and name it `input.txt`) and run your code with no command line arguments:
`$ ./raytrace`
Finally, we'll view your output with gimp:
`$ gimp output.ppm`

To prepare your code for submission, place all the source files, makefile(s) and output images in the `YourFamilyName` subdirectory and run
`$ tar cfz YourFamilyName.tgz YourFamilyName`

If you use a different environment, include a README file explaining how to compile and run your code and make sure you submit all the required files. Be prepared to personally demo your code on your laptop or easily accessible campus machine. For example, if you use Windows environment, submit a Visual Studio Express project. Make compiling and running your code as simple as possible. Happy Grader will give better grades :)

Late submissions will be accepted, but late penalty of $3^{\#\text{dayslate}-1}$ points will be applied.