

# Simple Ray-Tracing Algorithm

## Introduction

This program reads an input file defining a simple scene description and creates a ppm file describing the scene. The input file must be in the form: “keyword ” + item vector + “\n” (ex. “eye 0 0 0\n”) for each keyword. Keywords include: eye, viewdir, updir, vfov, imsize, bkgcolor, mtlcolor, and sphere. Each sphere is defined by the coordinate of its center and its radius. Each sphere must be defined on its own line and its color will correspond to the preceding mtlcolor. Color keywords are represented by their rgb value on the scale 0-1 ([0, 0, 0] would result in black while [1, 1, 1] would result in white).

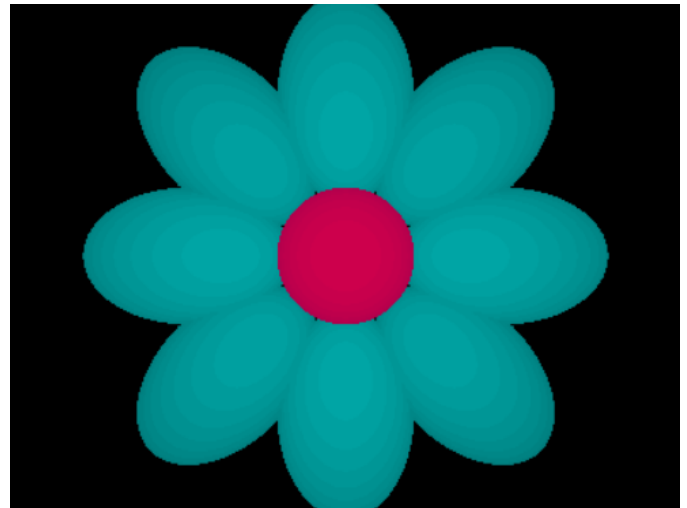


Image 1

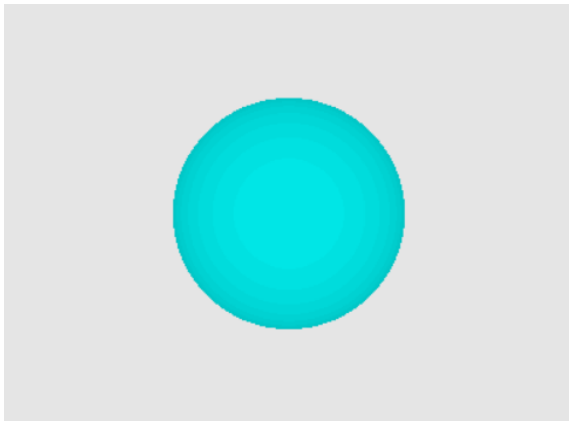


Image 2

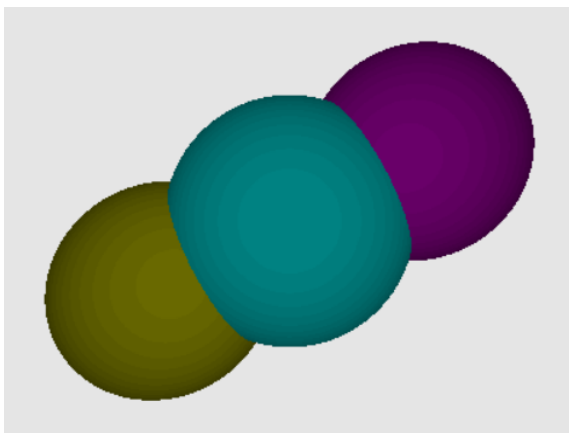


Image 3

## Shading

A simple shading algorithm was also implemented to better express the sphere's figure. This was done simply by decrementing each pixel's maximum rgb value by its distance from the eye (scaled).

## Adjusting Input Parameters

Adjusting the parameters of the input scene had various effects on the final appearance of the scene. It was found that the vfov had the greatest effect on the final image. Any vfov greater than 70 would greatly distort spheres not directly in the viewdir of the eye (image 1 had a vfov of 120 while image 3 had a vfov of 70). Adjusting the updir vector would rotate the image. For example, image 3 had an initial updir of [0, 1, 0] and a viewdir of [1, 0, 0]. Reversing the updir to [0, -1, 0] rotated the image 180 degrees (the yellow and purple spheres swapped positions). Image 4 showed the least amount of distortion by both decreasing the vfov to 50 and moving the eye position much farther away from the spheres.



Image 4