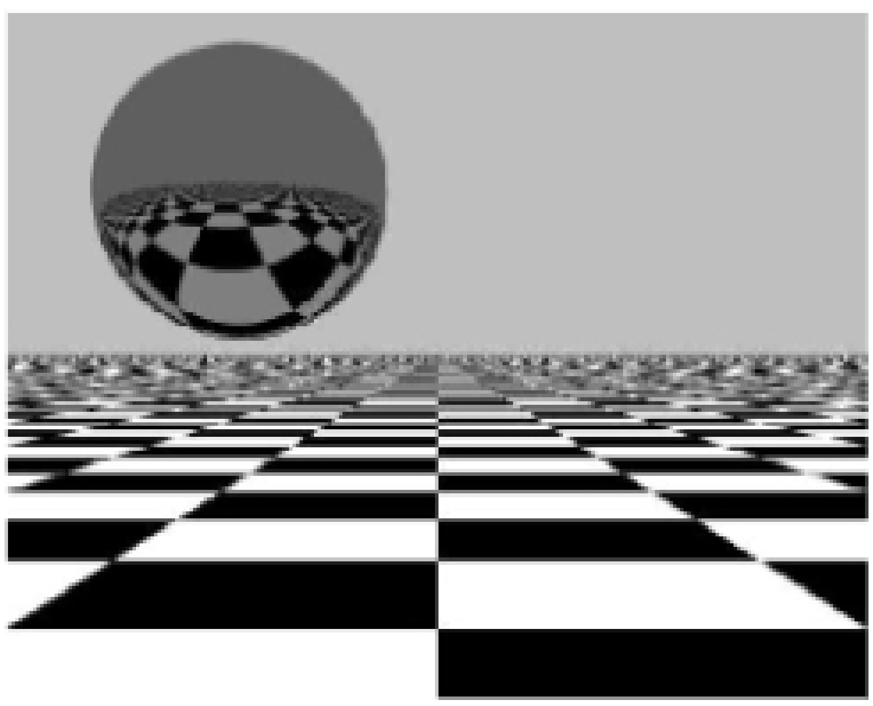
Homework CS116 due Friday 03/05/01

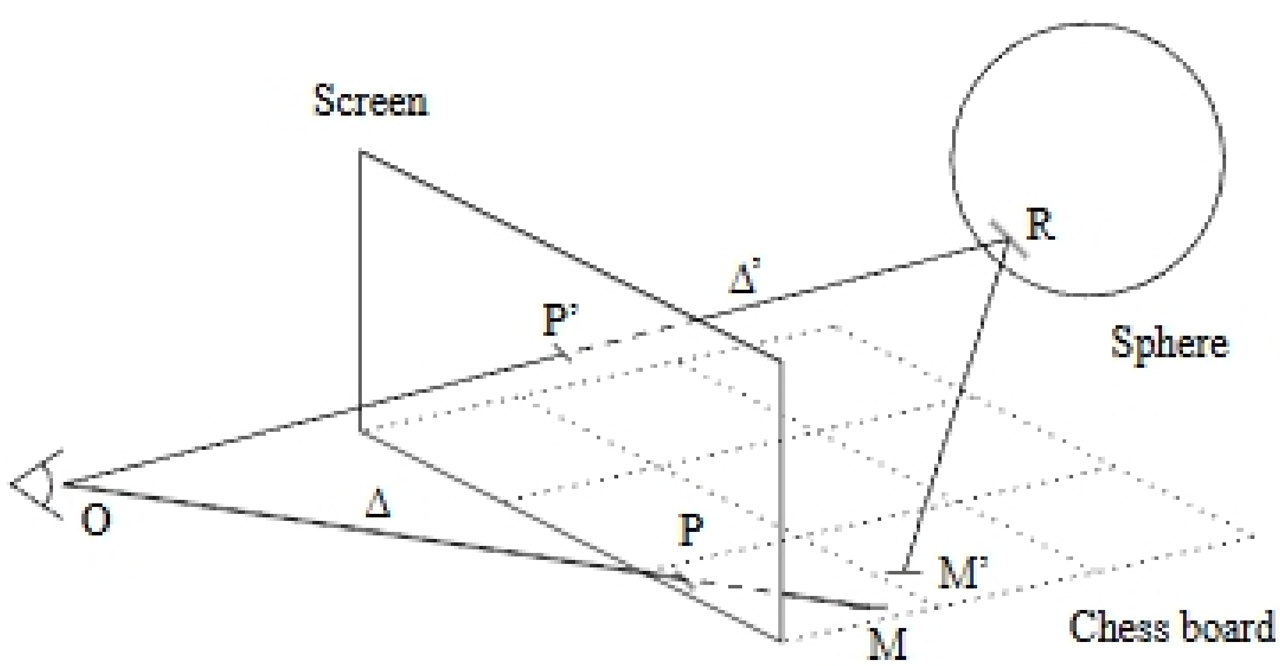
# 1 Ray-tracing

## 1.1 Introduction

The goal of this project is to implement a simple version of the well known ray-tracing algorithm. This technic is widely used to generate synthetic pictures and allow to simulate lot of very complex light, reflection and refraction effects (cf. figure 1).



The main idea is to associate to each pixel of the screen a virtual *ray* and to compute which objects in the scene intersect this ray (cf. figure 2).



## 1.2 Description of the algorithm

For the first version, we will not implement reflections or refractions, just visualizing opaque objects.

1. Open a window ;
2. loop thought all pixels :
   (a) compute the associated ray $\Delta$ ;
   (b) compute the first intersection with an object of the scene ;
   (c) draw the color ;
3. wait for a key-press.

The objects will have to be either : a sphere of a given color, location and size, or a "infinite chess board", which is horizontal, and is defined by its height, the two colors and the size of the squares.

## 1.3 Some maths

### 1.3.1 Parameterization of a ray

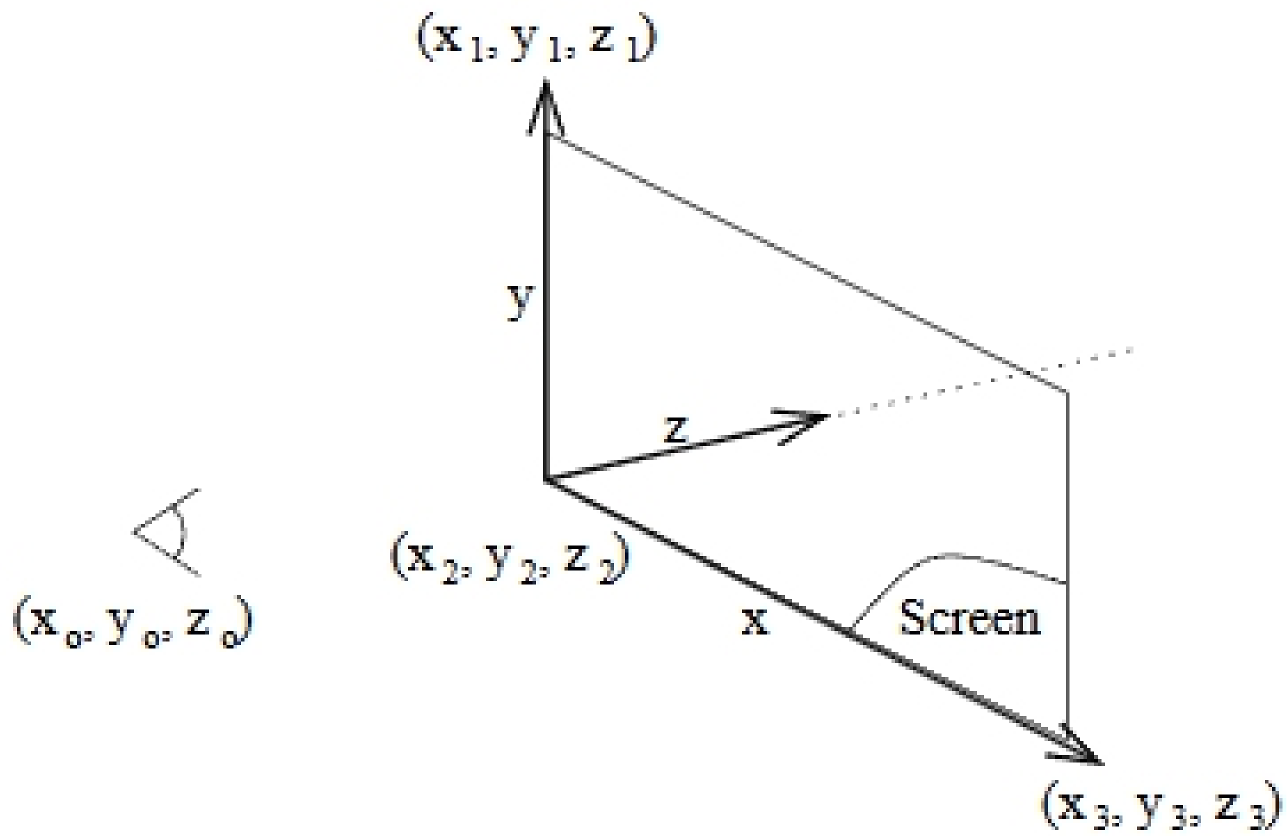
A ray is defined by its origin $(x_0, y_0, z_0)$ and its direction $(v_x, v_y, v_z)$. The coordinates of the points that belong to it are of the form $(x_0 + \lambda v_x, y_0 + \lambda v_y, y_0 + \lambda v_y)$ with $\lambda \in R_+$.

Given the location of the observer $(x_o, y_o, z_o)$, and the location of three corners of the screen : upper-left at $(x_1, y_1, z_1)$, lower-left at $(x_2, y_2, z_2)$ and lower-right at $(x_3, y_3, z_3)$, the size of the screen $w \times h$ and the pixel $(x_p, y_p)$, we want to estimate the ray's parameter.

The pixel's $P$ coordinates in the scene $(x, y, z)$ are estimated with linear interpolation. Let's define $\alpha = \frac{x_p}{w}$ and $\beta = 1 - \frac{y_p}{h}$, we have :

$$\begin{cases} x &= x_2 + \alpha(x_3 - x_2) + \beta(x_1 - x_2) \\ y &= y_2 + \alpha(y_3 - y_2) + \beta(y_1 - y_2) \\ z &= z_2 + \alpha(z_3 - z_2) + \beta(z_1 - z_2) \end{cases}$$

Thus, the ray as for origin the observer's location $(x_o, y_o, z_o)$ and for direction $(x - x_o, y - y_o, z - z_o)$.




#### 1.3.2 Sphere

A sphere is defined by the location of its center $(x_c, y_c, z_c)$, its radius $r$ and its color. The pixels that belongs to it verify $(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2$.

A ray has either zero, one or two intersections with a sphere. By substituting the coordinates of the point of the ray into the sphere's equation, we obtain a quadratic equation in $\lambda$.

#### 1.3.3 Chess board

A "infinite" chess board is defined by its height $y_{cb}$ the size of the squares $l$ and two colors $c_1$ and $c_2$. A ray meets such an object if its direction goes down (i.e. $v_y < 0$). In such a case, the coordinates of the intersection points can be estimated by computing $\lambda$ such that $y_o + \lambda v_y = y_{cb}$. The color of the met point will be $c_1$ if $sin(\pi \frac{x}{l}) sin(\pi \frac{z}{l}) \geq 0$ and $c_2$ if not.