

Tutorial on writing a Raytracer in Common Lisp

Part 2.1 ...continued

Alexander Lehmann <lehmann@in.tum.de>

So far...

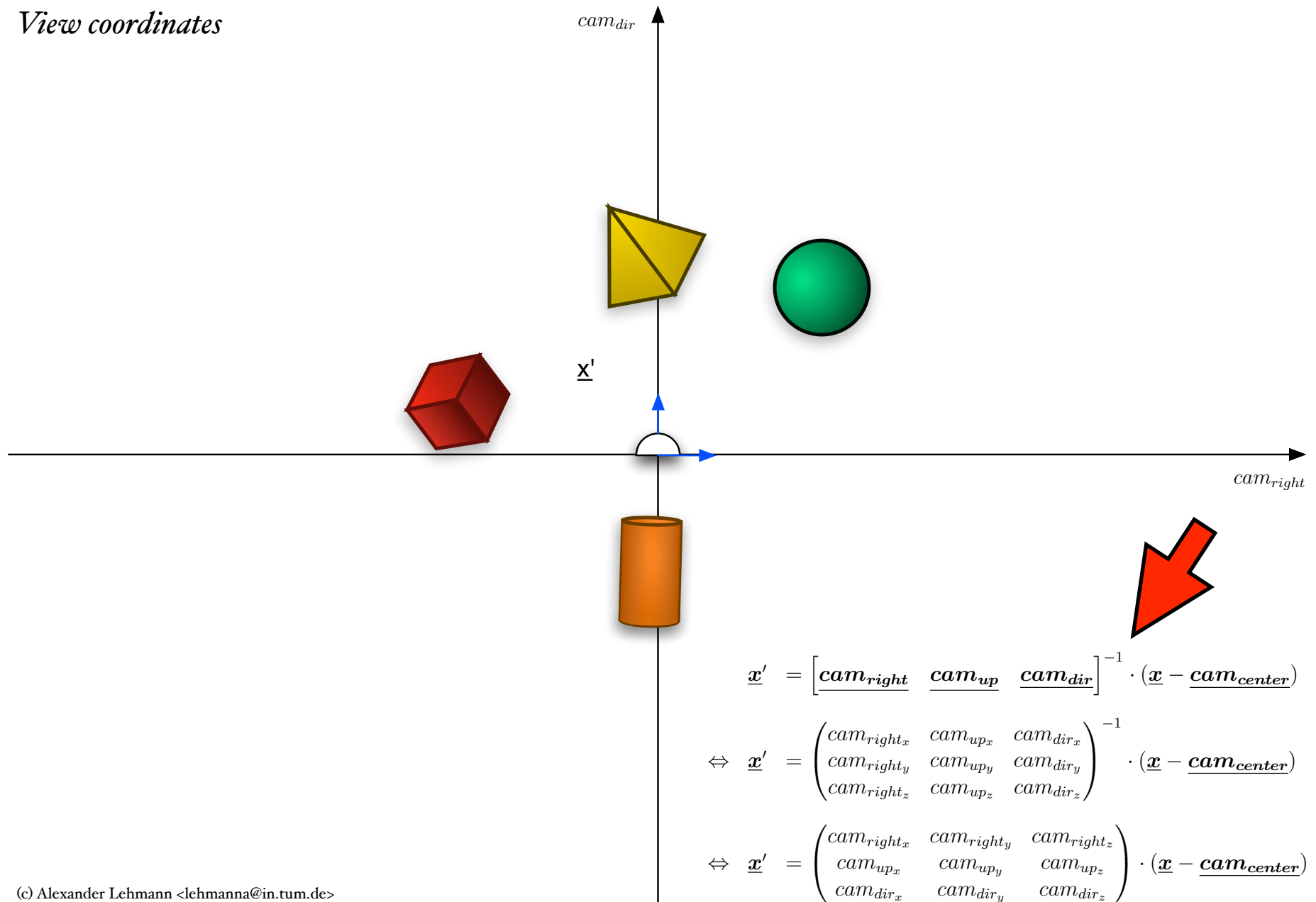
- Minor changes to the linear algebra package,
- discussed
 - the camera and the image plane as well as
 - world- and view-coordinates,
- asdf'ed and also
- implemented parts of the camera, the scene and the object superclass

Yet to do...

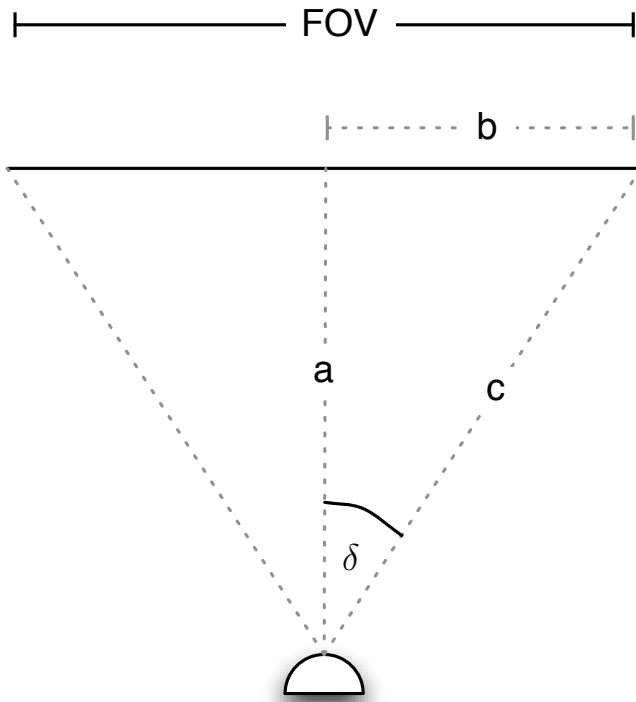
- small corrections regarding the camera and the scene,
- discuss intersections of
 - a ray and a sphere as well as those of
 - a ray and a cube and
- eventually finish the actual implementation of this tutorial's part 2.1.

World- vs. View-Coordinates

View coordinates

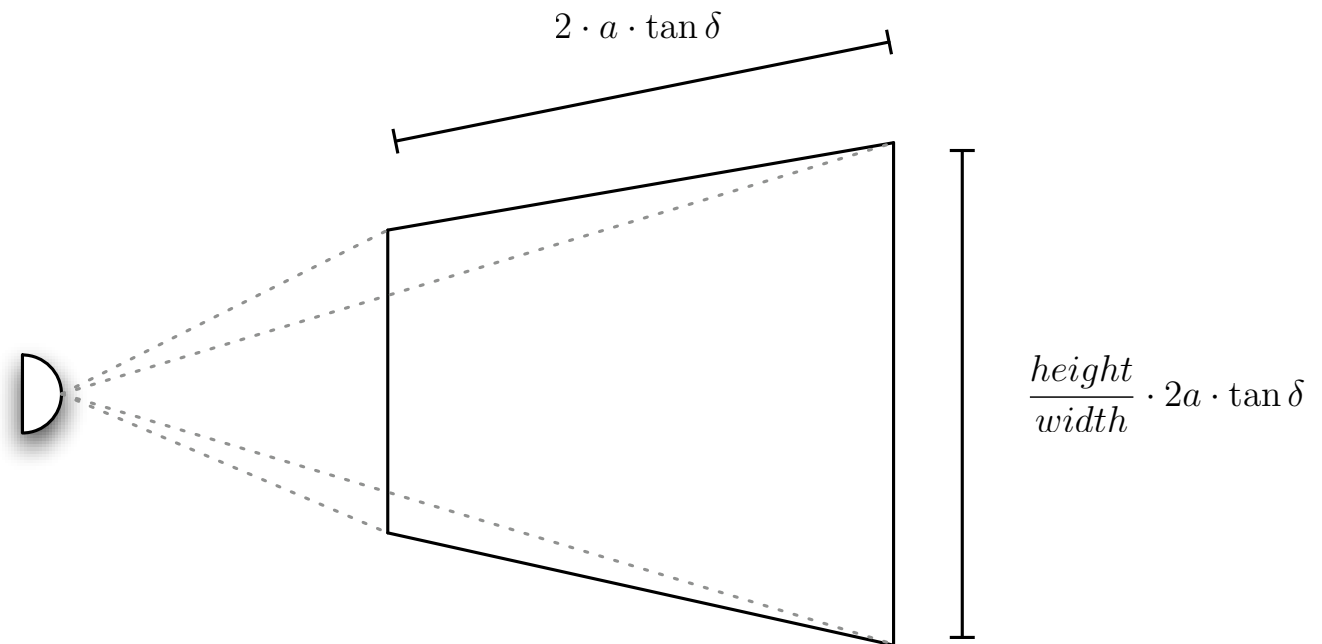


The Image Plane



$$\sin \delta = \frac{b}{c} \quad \cos \delta = \frac{a}{c}$$

$$\Rightarrow b = a \cdot \frac{\sin \delta}{\cos \delta} = a \cdot \tan \delta$$



Ray-Sphere-Intersection

A sphere is described by the equation

$$x^2 + y^2 + z^2 - r^2 = 0 \Leftrightarrow \underbrace{\begin{pmatrix} x & y & z \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}}_{=: \underline{\mathbf{x}}^2} - r^2 = 0.$$

Let $\underline{\mathbf{c}}$ be the center of the sphere.

$$\Rightarrow (\underline{\mathbf{x}} - \underline{\mathbf{c}})^2 - r^2 = 0$$

Also, let $\underline{\mathbf{r}}_o$ be the origin of the intersecting ray and $\underline{\mathbf{r}}_d$ be it's direction.

Replacing $\underline{\mathbf{x}}$ with $\underline{\mathbf{r}}_o + t \cdot \underline{\mathbf{r}}_d$ then yields

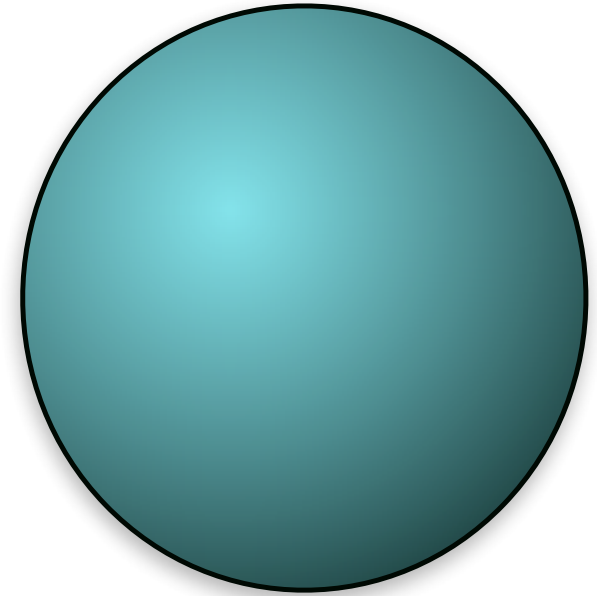
$$\begin{aligned} & ((\underline{\mathbf{r}}_o + t \cdot \underline{\mathbf{r}}_d) - \underline{\mathbf{c}})^2 - r^2 = 0 \\ \Rightarrow & \underbrace{\underline{\mathbf{r}}_d^2}_{=:a} t^2 + 2 \cdot \underbrace{(\underline{\mathbf{r}}_o \underline{\mathbf{r}}_d - \underline{\mathbf{c}} \underline{\mathbf{r}}_d)}_{=:b} t + \underbrace{\underline{\mathbf{r}}_o^2 - 2 \underline{\mathbf{r}}_o \underline{\mathbf{c}} + \underline{\mathbf{c}}^2 - r^2}_{=:c} = 0 \end{aligned}$$

Solving for t by using the *quadratic formula*

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

finally provides

$$t_{1,2} = \frac{-2 \cdot (\underline{\mathbf{r}}_o \underline{\mathbf{r}}_d - \underline{\mathbf{c}} \underline{\mathbf{r}}_d) \pm \sqrt{4 \cdot (\underline{\mathbf{r}}_o \underline{\mathbf{r}}_d - \underline{\mathbf{c}} \underline{\mathbf{r}}_d)^2 - 4 \cdot \underline{\mathbf{r}}_d^2 \cdot (\underline{\mathbf{r}}_o^2 - 2 \cdot \underline{\mathbf{r}}_o \underline{\mathbf{c}} + \underline{\mathbf{c}}^2 - r^2)}}{2 \cdot \underline{\mathbf{r}}_d^2}.$$



Ray-Plane-Intersection

All points on a plane can be expressed via a linear combination of

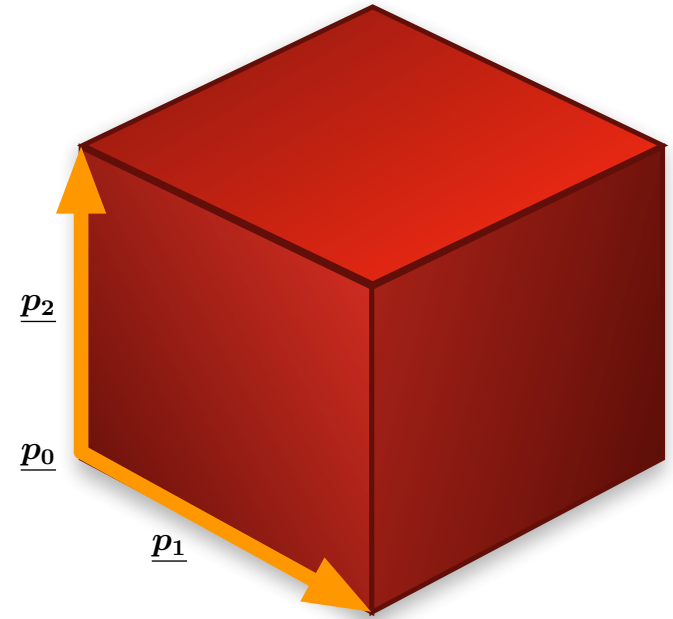
$$\begin{aligned} & \underline{p_0} + u \cdot \underline{p_1} + v \cdot \underline{p_2} \\ = & \underline{p_0} + [\underline{p_1} \quad \underline{p_2}] \cdot \begin{pmatrix} u \\ v \end{pmatrix}. \end{aligned}$$

Setting the plane's equation equal to the ray's equation gives

$$\begin{aligned} \underline{p_0} + [\underline{p_1} \quad \underline{p_2}] \cdot \begin{pmatrix} u \\ v \end{pmatrix} &= \underline{r_o} + t \cdot \underline{r_d} \\ \Leftrightarrow [\underline{p_1} \quad \underline{p_2} \quad -\underline{r_d}] \cdot \begin{pmatrix} u \\ v \\ t \end{pmatrix} &= \underline{r_o} - \underline{p_0}. \end{aligned}$$

Multiplying by the inverse of the matrix yields

$$\begin{pmatrix} u \\ v \\ t \end{pmatrix} = [\underline{p_1} \quad \underline{p_2} \quad -\underline{r_d}]^{-1} \cdot (\underline{r_o} - \underline{p_0}).$$



(c) Alexander Lehmann <lehmann@in.tum.de>

...suitable for (planar) rectangular and triangular faces...

Cramer's Rule

According to Cramer's rule, the inverse of a given matrix A can be determined as follows:

$$A^{-1} = \frac{\text{adj}(A)}{\det(A)}$$

Let $A \in \mathbb{R}^{3 \times 3}$ be a non-singular matrix

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}.$$

Its cofactor matrix C computes to

$$C = \begin{pmatrix} ei - fh & fg - di & dh - eg \\ ch - bi & ai - cg & bg - ah \\ bf - ce & cd - af & ae - bd \end{pmatrix},$$

hence:

$$\text{adj}(A) = C^T = \begin{pmatrix} ei - fh & ch - bi & bf - ce \\ fg - di & ai - cg & cd - af \\ dh - eg & bg - ah & ae - bd \end{pmatrix}$$

Preview of Part 2.2

- ▶ transformations (rotation, scaling, translation),
- ▶ axis-aligned bounding-boxes,
- ▶ the Blinn-Phong shading model and
- ▶ recursive raytracing.