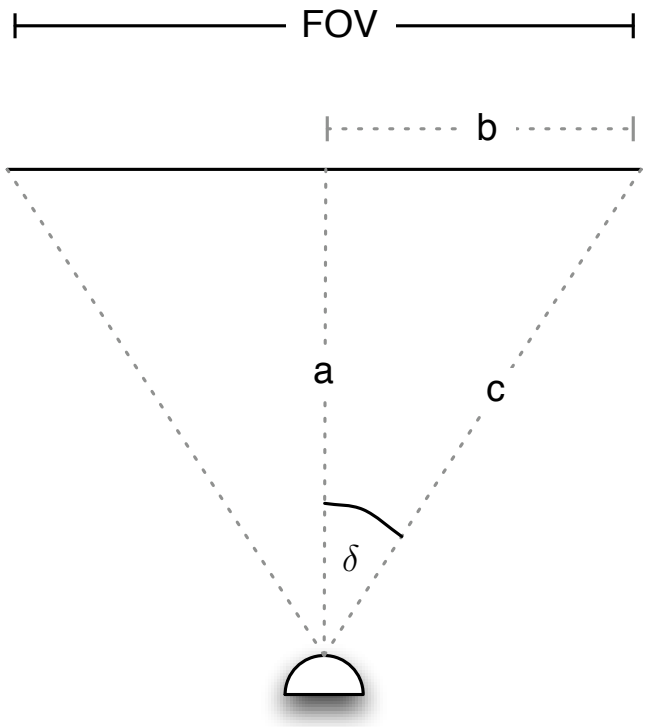
Tutorial on writing a Raytracer in Common Lisp

Part 2.1

Alexander Lehmann <lehmanna@in.tum.de>

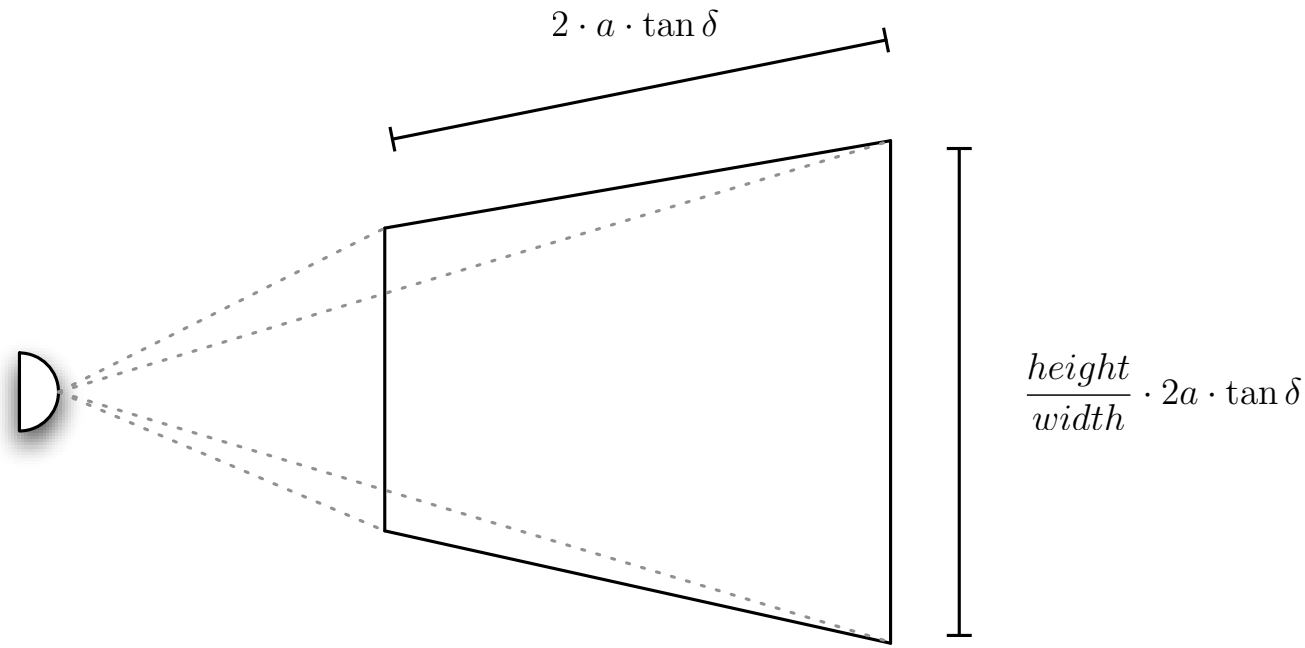
- Minor changes to the linear algebra package,
- the camera and the image plane,
- world- and view-coordinates,
- intersection of a ray and a sphere,
- intersection of a ray and a cube,
- asdf, zpng and
- the actual implementation ☺.

The Image Plane

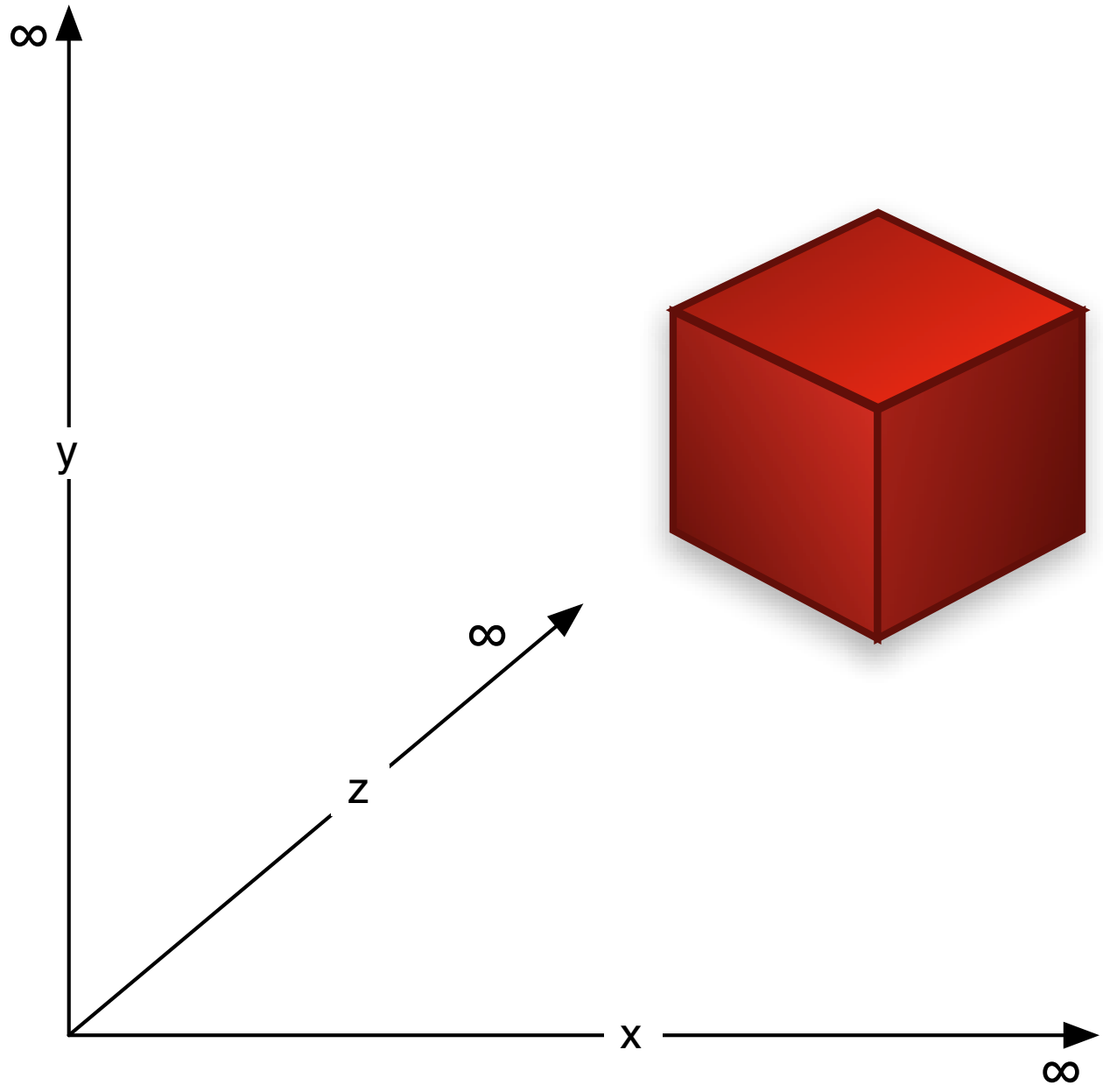


$$\sin\delta = \frac{b}{c} \qquad \cos\delta = \frac{a}{c}$$

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

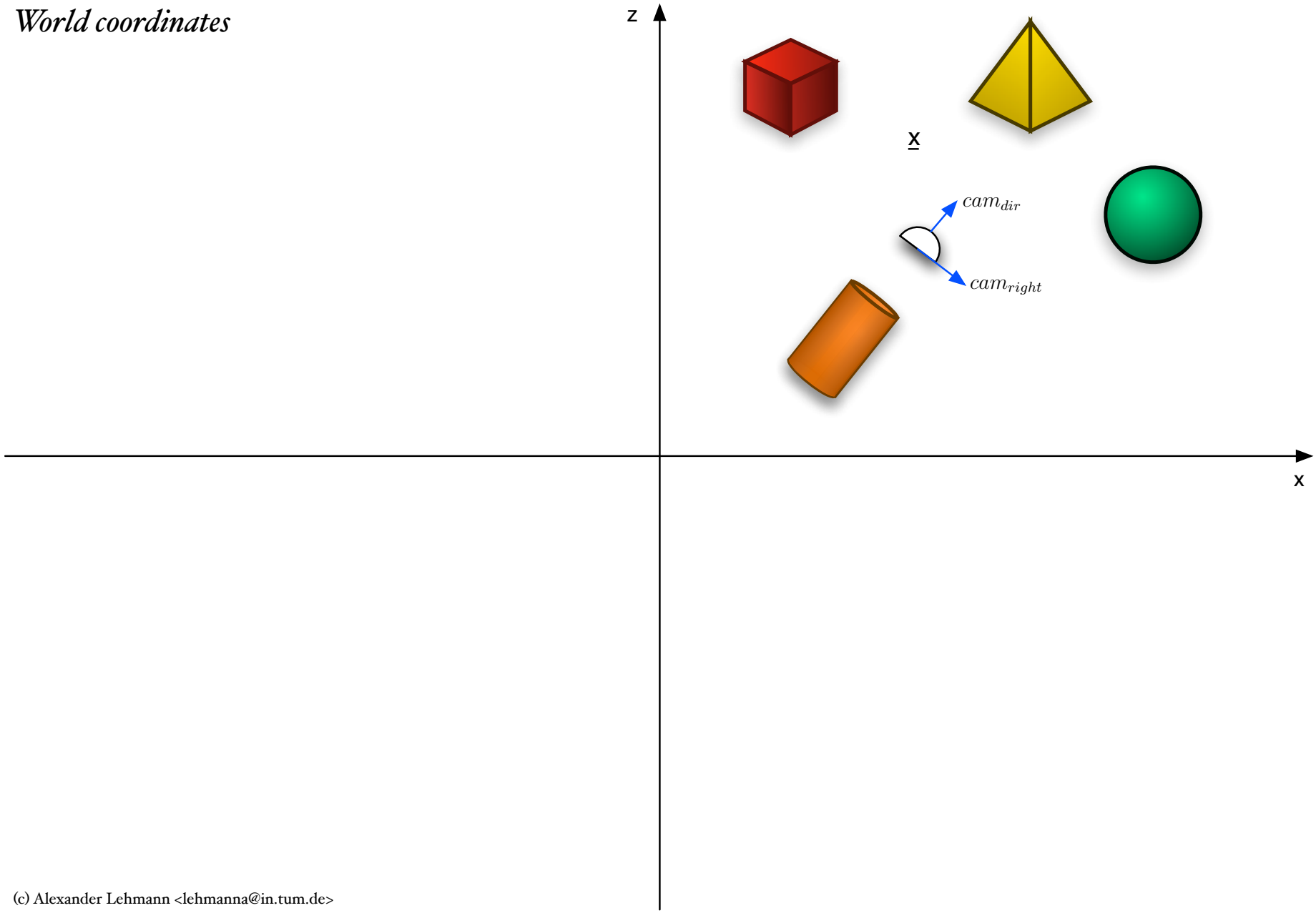


Coordinate System



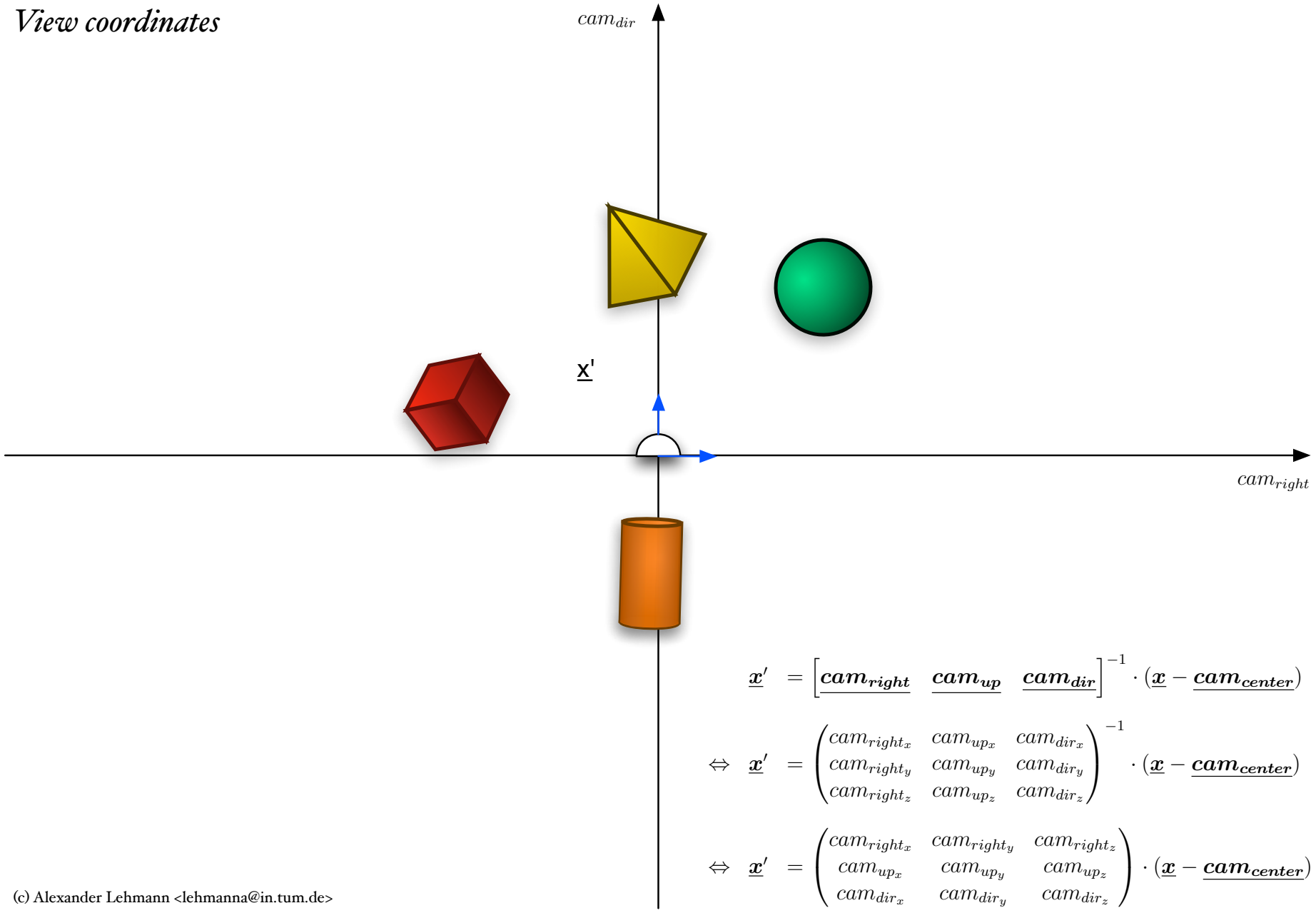
World- vs. View-Coordinates

World coordinates



World- vs. View-Coordinates

View coordinates



$$\underline{\boldsymbol{x}}' = \left[\underline{\boldsymbol{cam_{right}}} \quad \underline{\boldsymbol{cam_{up}}} \quad \underline{\boldsymbol{cam_{dir}}}\right]^{-1} \cdot (\underline{\boldsymbol{x}} - \underline{\boldsymbol{cam_{center}}})$$

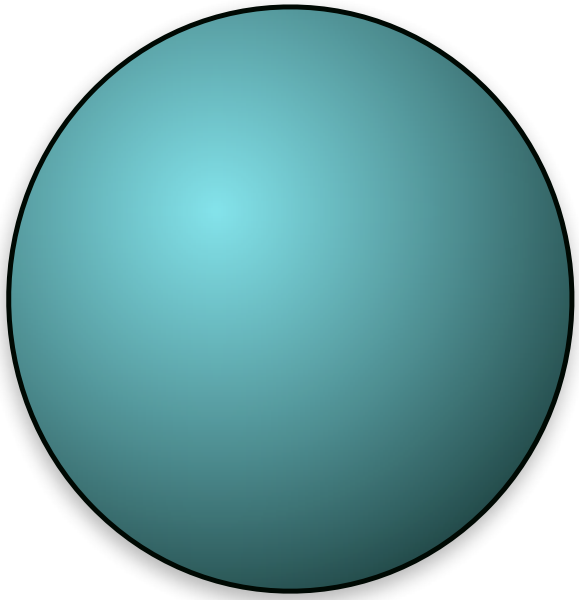
$$\Leftrightarrow \quad \underline{\boldsymbol{x}}' = \begin{pmatrix} cam_{right_x} & cam_{up_x} & cam_{dir_x} \\ cam_{right_y} & cam_{up_y} & cam_{dir_y} \\ cam_{right_z} & cam_{up_z} & cam_{dir_z} \end{pmatrix}^{-1} \cdot (\underline{\boldsymbol{x}} - \underline{\boldsymbol{cam_{center}}})$$

$$\Leftrightarrow \quad \underline{\boldsymbol{x}}' = \begin{pmatrix} cam_{right_x} & cam_{right_y} & cam_{right_z} \\ cam_{up_x} & cam_{up_y} & cam_{up_z} \\ cam_{dir_x} & cam_{dir_y} & cam_{dir_z} \end{pmatrix} \cdot (\underline{\boldsymbol{x}} - \underline{\boldsymbol{cam_{center}}})$$

Ray-Sphere-Intersection

A sphere is described by the equation

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



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

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

Also, let $\underline{\boldsymbol{r_o}}$ be the origin of the intersecting ray and $\underline{\boldsymbol{r_d}}$ be it's direction. Replacing $\underline{\boldsymbol{x}}$ with $\underline{\boldsymbol{r_o}} + t \cdot \underline{\boldsymbol{r_d}}$ then yields

$$\begin{aligned} & ((\underline{\boldsymbol{r_o}} + t \cdot \underline{\boldsymbol{r_d}}) - \underline{\boldsymbol{c}})^2 - r^2 && = 0 \\ \Rightarrow \quad & \underbrace{\underline{\boldsymbol{r_d}}^2}_{=:a}\, t^2 + \underbrace{2 \cdot (\underline{\boldsymbol{r_o}}\underline{\boldsymbol{r_d}} - \underline{\boldsymbol{c}}\underline{\boldsymbol{r_d}})}_{=:b}\, t + \underbrace{\underline{\boldsymbol{r_o}}^2 - 2\underline{\boldsymbol{r_o}}\underline{\boldsymbol{c}} + \underline{\boldsymbol{c}}^2 - r^2}_{=:c} && = 0 \end{aligned}$$

Solving for t by using the *quadratic formula*

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

finally provides

$$t_{1,2} = \frac{-2 \cdot (\underline{\boldsymbol{r_o}}\underline{\boldsymbol{r_d}} - \underline{\boldsymbol{c}}\underline{\boldsymbol{r_d}}) \pm \sqrt{4 \cdot (\underline{\boldsymbol{r_o}}\underline{\boldsymbol{r_d}} - \underline{\boldsymbol{c}}\underline{\boldsymbol{r_d}})^2 - 4 \cdot \underline{\boldsymbol{r_d}}^2 \cdot (\underline{\boldsymbol{r_o}}^2 - 2 \cdot \underline{\boldsymbol{r_o}}\underline{\boldsymbol{c}} + \underline{\boldsymbol{c}}^2 - r^2)}}{2 \cdot \underline{\boldsymbol{r_d}}^2}.$$

Ray-Plane-Intersection

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

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

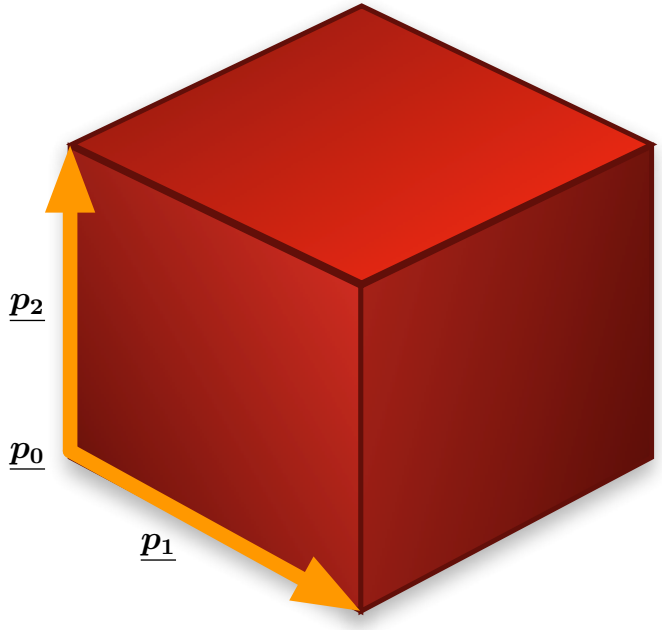
Setting the plane's equation equal to the ray's equation gives

$$\underline{\boldsymbol{p_0}} + [\underline{\boldsymbol{p_1}} \quad \underline{\boldsymbol{p_2}}] \cdot \begin{pmatrix} u \\ v \end{pmatrix} = \underline{\boldsymbol{r_o}} + t \cdot \underline{\boldsymbol{r_d}}$$

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

Multiplying by the inverse of the matrix yields

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



...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 $A \in \mathbb{R}^{3x3}$ be a non-singular matrix

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

It's 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.