Normalized device coordinates

bmild

April 2020

1 NDC ray space derivation

The standard 3D perspective projection matrix for homogeneous coordinates looks like this:

$$M = \begin{pmatrix} \frac{n}{r} & 0 & 0 & 0 \\ 0 & \frac{n}{t} & 0 & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \tag{1}$$

where n, f are the near and far clipping planes and r and t are the right and top bounds of the scene at the near clipping plane. [1] (Note that this is in the convention where the camera is looking in the $-z$ direction.) To project a homogeneous point $(x, y, z, 1)^\top$, you left-multiply it by M and then divide out the fourth coordinate:

$$\begin{pmatrix} \frac{n}{r} & 0 & 0 & 0 \\ 0 & \frac{n}{t} & 0 & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{n}{r}x \\ \frac{n}{t}y \\ \frac{-(f+n)}{f-n}z - \frac{-2fn}{f-n} \\ -z \end{pmatrix} \tag{2}$$

$$\text{project} \rightarrow \begin{pmatrix} \frac{n}{r}\frac{x}{-z} \\ \frac{n}{t}\frac{y}{-z} \\ \frac{(f+n)}{f-n} - \frac{2fn}{f-n}\frac{1}{-z} \end{pmatrix} \tag{3}$$

The projected point is now in normalized device coordinate (NDC) space, where the original viewing frustum has been mapped to the cube $[-1, 1]^3$.

Our goal is to take a ray $\mathbf{o}+t\mathbf{d}$ and calculate a ray origin $\mathbf{o}'$ and direction $\mathbf{d}'$ in NDC space such that for every t, there exists a t' such that $\pi(\mathbf{o}+t\mathbf{d}) = \mathbf{o}'+t'\mathbf{d}'$ (π is projection using the above matrix). In other words, the projected original ray and the NDC space ray trace out the same points (but not necessarily at the same rate).

[1] `http://www.songho.ca/opengl/gl_projectionmatrix.html`

We'll rewrite the projected point as $(a_x x/z, a_y y/z, a_z + b_z/z)^\top$ to keep things less messy. Writing out all the constraints we need to satisfy:

$$\begin{pmatrix} a_x \frac{o_x+td_x}{o_z+td_z} \\ a_y \frac{o_y+td_y}{o_z+td_z} \\ a_z + \frac{b_z}{o_z+td_z} \end{pmatrix} = \begin{pmatrix} o'_x + t'd'_x \\ o'_y + t'd'_y \\ o'_z + t'd'_z \end{pmatrix} \tag{4}$$

For convenience, we'll decide that $t' = 0$ and $t = 0$ should map to the same point. This gives us our NDC space $\mathbf{o}'$

$$\mathbf{o}' = \begin{pmatrix} o'_x \\ o'_y \\ o'_z \end{pmatrix} = \begin{pmatrix} a_x \frac{o_x}{o_z} \\ a_y \frac{o_y}{o_z} \\ a_z + \frac{b_z}{o_z} \end{pmatrix} = \pi(\mathbf{o}) \tag{5}$$

which is just the projection $\pi(\mathbf{o})$ of the original ray origin. Now we can figure out what t' and $\mathbf{d}'$ should be.

$$\begin{pmatrix} t'd'_x \\ t'd'_y \\ t'd'_z \end{pmatrix} = \begin{pmatrix} a_x \frac{o_x+td_x}{o_z+td_z} - a_x \frac{o_x}{o_z} \\ a_y \frac{o_y+td_y}{o_z+td_z} - a_y \frac{o_y}{o_z} \\ a_z + \frac{b_z}{o_z+td_z} - a_z - \frac{b_z}{o_z} \end{pmatrix} \tag{6}$$

$$= \begin{pmatrix} a_x \frac{o_z(o_x+td_x) - o_x(o_z+td_z)}{(o_z+td_z)o_z} \\ a_y \frac{o_z(o_y+td_y) - o_y(o_z+td_z)}{(o_z+td_z)o_z} \\ b_z \frac{o_z - (o_z+td_z)}{(o_z+td_z)o_z} \end{pmatrix} \tag{7}$$

$$= \begin{pmatrix} a_x \frac{td_z}{o_z+td_z} \left(\frac{d_x}{d_z} - \frac{o_x}{o_z} \right) \\ a_y \frac{td_z}{o_z+td_z} \left(\frac{d_y}{d_z} - \frac{o_y}{o_z} \right) \\ -b_z \frac{td_z}{o_z+td_z} \frac{1}{o_z} \end{pmatrix} \tag{8}$$

$$\tag{9}$$

We can factor out a common expression that depends only on t to get:

$$t' = \frac{td_z}{o_z + td_z} = 1 - \frac{o_z}{o_z + td_z} \tag{10}$$

$$\mathbf{d}' = \begin{pmatrix} a_x \left(\frac{d_x}{d_z} - \frac{o_x}{o_z} \right) \\ a_y \left(\frac{d_y}{d_z} - \frac{o_y}{o_z} \right) \\ -b_z \frac{1}{o_z} \end{pmatrix} \tag{11}$$

Note that, as we wanted, $t' = 0$ when $t = 0$. Additionally, we see that $t' \to 1$ as $t \to \infty$.

Going back to the original projection matrix, we see our constants are

$$a_x = -\frac{n}{r} \tag{12}$$

$$a_y = -\frac{n}{t} \tag{13}$$

$$a_z = \frac{f+n}{f-n} \tag{14}$$

$$b_z = \frac{2fn}{f-n} \tag{15}$$

By standard pinhole camera math, we can reparameterize as

$$a_x = -\frac{f_{cam}}{W/2} \tag{16}$$

$$a_y = -\frac{f_{cam}}{H/2} \tag{17}$$

where W, H are the width and height of the image and f_{cam} is the focal length of the pinhole camera.

In NeRF, we assume that the far scene bound is infinity (this costs us very little since NDC uses the z dimension to represent *inverse* depth, i.e., disparity). In this limit the z constants simplify to

$$a_z = 1 \tag{18}$$

$$b_z = 2n \tag{19}$$

Combining everything together, we get the expressions found in the `ndc_rays()` function in the NeRF code:

$$\mathbf{o}' = \begin{pmatrix} -\frac{f_{cam}}{W/2}\frac{o_x}{o_z} \\ -\frac{f_{cam}}{H/2}\frac{o_y}{o_z} \\ 1 + \frac{2n}{o_z} \end{pmatrix} \tag{20}$$

$$\mathbf{d}' = \begin{pmatrix} -\frac{f_{cam}}{W/2}\left(\frac{d_x}{d_z} - \frac{o_x}{o_z}\right) \\ -\frac{f_{cam}}{H/2}\left(\frac{d_y}{d_z} - \frac{o_y}{o_z}\right) \\ -2n\frac{1}{o_z} \end{pmatrix} \tag{21}$$

One final trick we play in NeRF is that we move $\mathbf{o}$ to the ray's intersection with the near plane at $z = -n$ (before this NDC conversion) by taking $\mathbf{o}_n = \mathbf{o} + t_n\mathbf{d}$ for $t_n = -(n + o_z)/d_z$. Once we're in NDC, this allows us to simply sample t' linearly from 0 to 1 in order to get a linear sampling in disparity from n to ∞ in the original space!

2 NeRF code

```
def ndc_rays(H, W, focal, near, rays_o, rays_d):

    # Shift ray origins to near plane
    t = -(near + rays_o[...,2]) / rays_d[...,2]
    rays_o = rays_o + t[...,None] * rays_d

    # Projection
    o0 = -1./(W/(2.*focal)) * rays_o[...,0] / rays_o[...,2]
    o1 = -1./(H/(2.*focal)) * rays_o[...,1] / rays_o[...,2]
    o2 = 1. + 2. * near / rays_o[...,2]

    d0 = -1./(W/(2.*focal)) * (rays_d[...,0]/rays_d[...,2] - \
                               rays_o[...,0]/rays_o[...,2])
    d1 = -1./(H/(2.*focal)) * (rays_d[...,1]/rays_d[...,2] - \
                               rays_o[...,1]/rays_o[...,2])
    d2 = -2. * near / rays_o[...,2]

    rays_o = tf.stack([o0,o1,o2], -1)
    rays_d = tf.stack([d0,d1,d2], -1)

    return rays_o, rays_d
```