# Assignment 3 - Projections

Mikkel Willén

December 22, 2022

## Contents



## 1 Introduction

This is the third assignment in computer graphics and is about projections in a vector space. In this assignment we will go over how to transform any view volume to a canonical perspective view volumen and then from this to a canonical parallel view volume. We will start by taking a look at what a projection is and what types exists.

## 2 Theory

Projections is to tranform points in a vector space with n dimensions to points in a vector space with less than n dimensions. The projections we will look at is what we call planar geometric projections, since we are working with projections on a plane and not on a curved surface. There are two different types of planar geometric projections, parallel and perspective. The difference between these two types are their relation to the center of projection. If the distance to the center of projection is finite it is a perspective projection. If the distance is infinite it is a parallel projection.

Parallel projections are categorized in two further types, which is defined by their relation to the direction of the projection and the norm to the projection plane. The first one is orthographic parallel projections. In this one the directions, as mentioned before, is either the same or the reverse of each other. The the other parallel projection, this is not the case. We will look at the most common parallel projection, which orthographic projections. [**?**]

When we have to transform projections, it is done in a projection plane or a eye coordinate system. We need two vectors, VPN (view plane normal) and VUP (view up vector), to create the eye coordinate system.

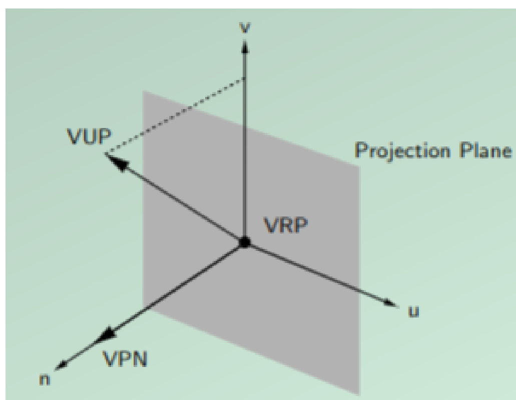


Figure 1: Eye coordinate system

The figure above shows how the eye coordinate system looks. The point VRP (view reference point) is the reference point in the system. The vector space $(u, v, n)^T$ is defined by the three variables $(u, v, n)$. $n$ is the unit vector in the $VPN$ direction. It kan be written as

$$n = \frac{VPN}{||VPN||_2}$$

$u$ is also a unit vector, and is perpendicular on $VUP$ and $VPN$. It can be written as

$$u = \frac{VUP \times VPN}{||VUP \times VPN||_2}$$

$v$ is also a unit vector in the same direction as $VUP$'s porjection along $VPN$ on the projection plane. $v$ is defined as

$$v = \frac{VPN \times (VUP \times VPN)}{||VPN \times (VUP \times VPN)||_2} = n \times u$$

This is how we get the eye coordinate system from $VPN$ and $VUP$.

To make the tranformation from the perspective view volume to the canonical perspective view volumen, we need to do the following:

1. Tranlate $VRP$ to origo
2. Rotate the eye coordinate system, so it is aligned with the worlds coordinate system. This means that the u-axis becomes the x-axis, the v-axis becomes the y-axis and the n-axis becomes the z-axis
3. Translate $PRP$ to origo
4. Shear so the center line of out view volume becomes our z-axis
5. Scale it to the canonical perspective view volume.

The first thing we do it to translate $VRP$ to origo. We do this with the translation matrix

$$T(-VRP) = \begin{bmatrix} 1 & 0 & 0 & -VRP_x \\ 0 & 1 & 0 & -VRP_y \\ 0 & 0 & 1 & -VRP_s \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The next thing to do is to rotate $(u, v, n)$ so it alignes with the world coordinate system $(x, y, z)$. We use the rotation matrix to do this.

$$R = \begin{bmatrix} r_{1,x} & r_{2,x} & r_{3,x} & 0 \\ r_{1,y} & r_{2,y} & r_{3,y} & 0 \\ r_{1,z} & r_{2,z} & r_{3,z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where the first row corrosponds to

$$R_x^T = (r_{1,x}, r_{2,x}, r_{3,x}) = \frac{VUP \times R_z}{||VUP \times R_z||_2}$$
$$R_y^T = (r_{1,y}, r_{2,y}, r_{3,y}) = \frac{R_z \times R_x}{||R_z \times R_x||_2}$$
$$R_x^T = (r_{1,}, r_{2,x}, r_{3,x}) = \frac{VPN}{||VPN||_2}$$

Third step is to translate $PRP$ to origo.

$$T(-PRP) = \begin{bmatrix} 1 & 0 & 0 & -PRP_u \\ 0 & 1 & 0 & -PRP_n \\ 0 & 0 & 1 & -PRP_v \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The pyramide will now be a bit off. To correct this, we will use shear on the center of view, $CW$, so it is aligned with the z-axis, as seen on the figure below. We got direction of projection defined as

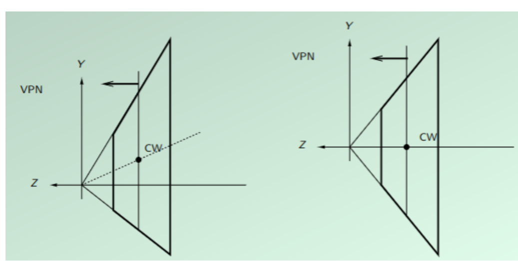


Figure 2: Figure before and after shear

$$DOP = PRP - CW = (DOP_u, DOP_v DOP_n, 0)^T$$

and the shear matrix is

$$sh_{per} = \begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where we have to find $sh_x$ and $sh_y$. We now also have

$$DOP = sh_{per} \cdot DOP = (0, 0, DOP_n, 0)^T$$

This can be written out as

$$\begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} DOP_u \\ DOP_v \\ DOP_n \\ 0 \end{bmatrix} = \begin{bmatrix} DOP_u + sh_x \\ DOP_v + sh_y \\ DOP_n \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ DOP_n \\ 0 \end{bmatrix}$$

This gives us the following two equations for $sh_x$ and $sh_y$

$$sh_x = -\frac{DOP_u}{DOP_n}$$
$$sh_y = -\frac{DOP_v}{DOP_n}$$

If we insert this is the $sh_{per}$ matrix, we get

$$sh_{per} = \begin{bmatrix} 1 & 0 & -\frac{DOP_u}{DOP_n} & 0 \\ 0 & 1 & -\frac{DOP_v}{DOP_n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In the fifth and final step, we have to scale the canonical projection, so we get the correct size. The figure below shows a figure and how we want to change the size. To get the desired figure, we have to place the front clipping plane and the back clipping plane the right places, so the pyramid ends up having a height of 1 and have an angle of 45°. Then we have to scale the length to our $CW$, which is $-PRP_n$. We only take half the length to $CW$, since both lengths have to be the same on either side. We have new values of $VRP$ after the translation and sheared, and we have

$$VRP' = sh_{per} \cdot T(-PRP) = \begin{bmatrix} -PRP_u + \frac{DOP_u}{DOP_n} PRP_n \\ -PRP_v + \frac{DOP_v}{DOP_n} PRP_n \\ -PRP_n \\ 1 \end{bmatrix}$$

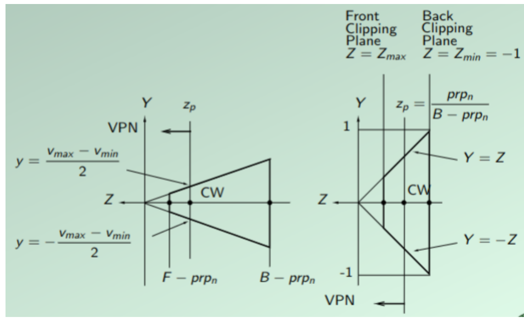


Figure 3: Figure before and after scaling

Now we scale in the x and y directions, so half of CW is equal to $VRP'_z = -PRP_n$ and so we get a 45° line. This is done by

$$s_x = \frac{-2VRP'_z}{u_{max} - u_{min}} = \frac{2PRP_n}{u_{max} - u_{min}}$$
$$s_y = \frac{-2VRP'_z}{v_{max} - v_{min}} = \frac{2PRP_n}{u_{max} - u_{min}}$$

Next thing is to scale $VRP'_z + B = B - PRP_n \to -1$ where $B$ is the back clipping plane, which becomes equal to $Z = -1$. We divide by $-1$ and we get

$$s_z = \frac{-1}{VRP'_z + B} = \frac{-1}{B - PRP_n}$$

This gives us the correct height of the pyramid, and we now also have all our scaling factors for $S_{per}$

$$s_x = \frac{-2PRP_n}{(u_{max} - u_{min})(B - PRP_n)}$$
$$s_y = \frac{-2PRP_n}{((v_{max} - v_{min})(B - PRP_n)}$$
$$s_z = \frac{-1}{B - PRP_n}$$

which we insert in the scale matrix

$$S_{per} = \begin{bmatrix} \frac{-2PRP_n}{(u_{max}-u_{min})(B-PRP_n)} & 0 & 0 & 0 \\ 0 & \frac{-2PRP_n}{((v_{max}-v_{min})(B-PRP_n)} & 0 & 0 \\ 0 & 0 & \frac{-1}{B-PRP_n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

and we now have the canonical perspective view volume from any perspective view volume.

All the steps for the transformation can be written as

$$N_{per} = S_{per} \cdot sh_{per} \cdot T(-PRP) \cdot R \cdot T(-VRP)$$

After scaling the values $Z_{min}$, $Z_{max}$ and $z_p$ have changed, and are now

$$\begin{aligned} Z_{min} &= -1 \\ Z_{max} &= \frac{F - PRP_n}{B - PRP_n} \\ z_p &= \frac{PRP_n}{B - PRP_n} \end{aligned}$$

These values needs to be tranformed to the canonical parallel view volume. On the figure below, we see the canonical perspective view volume and the transformation to the canonical parallel view volume. We use the parallel

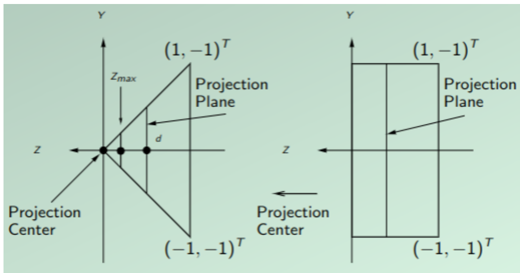


Figure 4: Canonical perspective view volume and canonical parallel view volume

perspective matrix to do the transformation.

$$M_{perpar} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1+Z_{max}} & \frac{-Z_{max}}{1+Z_{max}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

We get the matrix by the following steps. First thing we do is to look at equations for perspective projections for $[x, y]^T$ which are

$$x = d\frac{X}{Z}$$

$$y = d\frac{Y}{Z}$$

The two equations have to be in the interval

$$[x, y]^T \in [-|d|, |d|] \times [-|d|, |d|]$$

We then transform the coordinate to be in the interval

$$[-1, 1] \times [-1, 1]$$

by scaling with the factor $-\frac{1}{d}$, and we get

$$x = -\left(\frac{1}{d}\right) d\frac{X}{Z} = -\frac{X}{Z}$$
$$y = -\left(\frac{1}{d}\right) d\frac{Y}{Z} = -\frac{Y}{Z}$$

Now the equations are within the interval, which also corrosponds to the black clipping plane.

Next step is to get the $z$ value. We start by making a function $f$, which can transform the $z$ coordinate. $f(Z) = aZ + b$. We put some restrictions on our function

$$f(Z_{max} = aZ_{max} + b = 0$$
$$f(-1) = a + b = -1$$

which gives us the equation system

$$\begin{bmatrix} Z_{max} & 1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

If we solve for $a$ and $b$ we get

$$a = \frac{\begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}}{\begin{bmatrix} Z_{max} & 1 \\ -1 & 1 \end{bmatrix}} = \frac{1}{Z_{max} + 1}$$

$$b = \frac{\begin{bmatrix} Z_{max} & 0 \\ -1 & -1 \end{bmatrix}}{\begin{bmatrix} Z_{max} & 1 \\ -1 & 1 \end{bmatrix}} = \frac{-Z_{max}}{Z_{max} + 1}$$

This also gives us the function

$$z = f(Z) = \frac{Z}{Z_{max}+1} - \frac{-Z_{max}}{Z_{max}+1}$$

We use these equations to create the perspective parallel matrix

$$m_{perpar} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1+Z_{max}} & \frac{-Z_{max}}{1+Z_{max}} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

# 3 Implmentation

All the steps we have looked at in the theory section can be implemented with to equations.
Parallel projections:

$$M_{parTotal} = M_{wv} \cdot S_{par} \cdot T_{par} \cdot sh_{par} \cdot R \cdot T(-VRP)$$

where $S_{par} \cdot T_{par} \cdot sh_{par}$ is the view projection and $\cdot R \cdot T(-VRP)$ is view orientation.
Perspective projections:

$$M_{perTotal} = M_{wv} \cdot M_{perpar} \cdot S_{par} \cdot sh_{par} \cdot T_{par}(-PRP) \cdot R \cdot T(-VRP)$$

where $M_{perpar} \cdot S_{par} \cdot sh_{par} \cdot T_{par}(-PRP)$ ir the view projection and $R \cdot T(-VRP)$ is the view orientation.

The first thing we do in Camera::ComputeViewOrientation is to make the 4x4 matrix $T(-VRP)$. Then we create the eye coordinate system, and do a rotation on $(n, u, v)$. To get the view orientation matrix, we multiply the rotation matrix with the translation matrix, and find the inverse.

The first thing we do in Camera::ComputeViewProjection is to make the 4x4 matrix $(-PRP)$, and then we find the two vectors $CW$ and $DOP$. Then we calculate the shear x and shear y values, and insert them in the shearXY matrix. Next step is to compute the scaling factors, and insert them in the matrix S. The last step is to compute the inverse scale matrix and the inverse $M_{perpar}$ matrix. With these we have all we need to compute the view projection matrix and the inverse view projection matrix.

We then use the two matrixes we have found to compute the CurrentTransformationMatrix and InvCurrentTransformationMatrix().

# 4 Testing

We check the figures in the program and check if they look like they should. As seen on the picture, we get the right figures of the house.

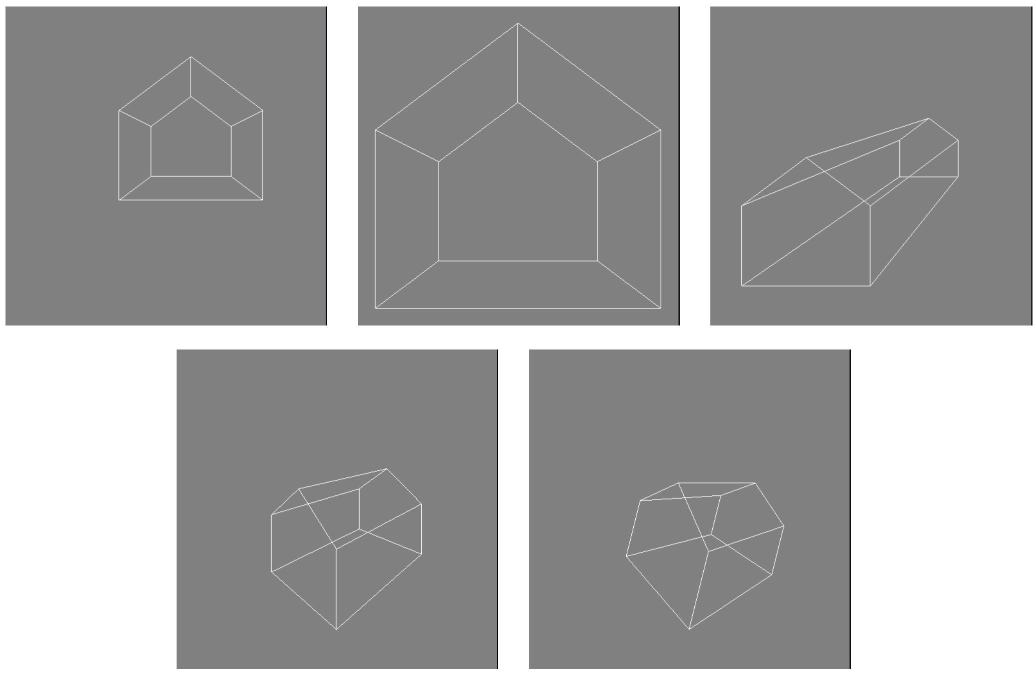

Figure 5: Figures of the house, looking like they should

# 5 Conclusion

We have describe how to implement transformation of any perspective view volume to the canonicall perspective view volume, and then to canonical parallel view volume. We have then implemented it, and tested that the program works as intended, and we get the desired figures, when running the program.