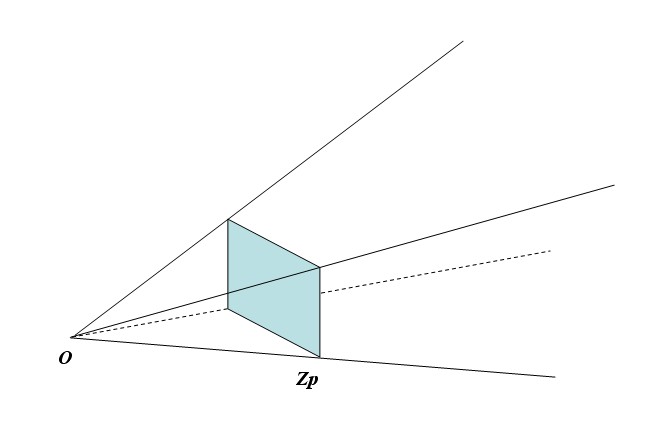
**透视投影：**

为了进行透视投影我们需要一个投影面和一个灭点。不失一般性的，我们假设灭点在世界坐标的原点，投影面为z=Zp。



那么根据透视原理，我们可以相应的计算出世界中的任意一点（x，y，z）经过投影之后的在屏幕上的坐标（xp,yp）：

xp=Zp(x/z), yp=Zp(y/z).

为了更加方便，我们引入了四维矩阵和四维矢量。实际空间中的点都是3维的。但是为了进行平移和投影变换，我们额外引进了一个维度。此时空间中的任一点可以表示成如下形式：

(x,y,z,1)

之所以将最后的一个维度设为1，一方面方便做平移变换，另一方面也方便透视变换。

此时根据透视原理可以构造出一个透视投影矩阵：

Zp 0 0 0

0 Zp 0 0

0 0 1 0

0 0 1 0

任意点经过他变换之后变成

(x\*Zp,y\*Zp,z,z) （\*）

有了这个矢量之后，我们将它的每个分量都除以w分量，就得到屏幕上的点了。

**裁剪空间 和 归一化设备坐标（clipping space & NDC）**

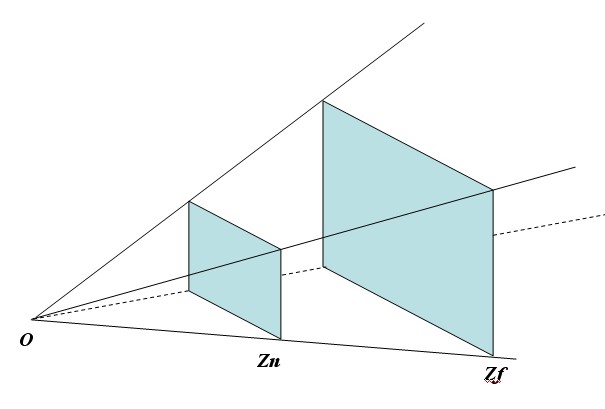
以上都是没有考虑具体设备的理论。当考虑到具体的设备时，由于一些限制，我们需要一些额外的工作。

不管这些限制是什么，最后做出的规定是，经过透视变换之后的矢量（\*），在除以w分量之后，只有在z∈(0,1),x,y属于(-1,1)时，他们才会被设备渲染。

我们现在知道了设备的渲染范围是有限的而且告诉了我们这个范围，那么我们现在要做的是在世界中挑选我们想要呈现的部分并经过一定的变换将这部分给变换到设备能够渲染的范围之内。

可以看出x，y方向的限制其实就是给投影面划定了大小（2×2），而z方向的限制也要求我们在z轴上截取一个范围。但我们发现限制当中并没有对投影面的位置做具体的规定。这个位置由我们来定，不失一般性地，我们仍将投影面的位置设为Zp。

如果我们想要划定下面示意图中所划定的范围作为想要呈现的区域（图中未标出投影面的位置）：



这部分在z轴上为(Zn,Zf).那么为了让这部分显示出来，我们需要进行一种映射变换，经过变换之后Zn点变到0，Zf变到1，这样我们的矩阵变成：

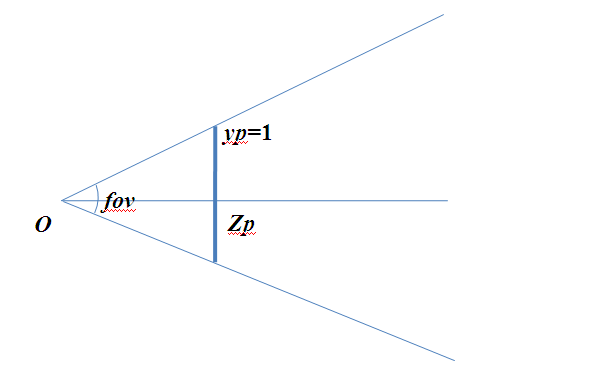
Zp 0 0 0

0 Zp 0 0

0 0 Zf/(Zf-Zn) -Zn\*Zf/(Zf-Zn)

0 0 1 0

现在可以看到在这个矩阵中只有Zp是一个未知量，这个是由我们自由设定的，



由上图可以知道他和fov有关：

Tan(fov/2)=1/Zp

Or

Zp=cot(fov/2)

从而矩阵变成

cot(fov/2) 0 0 0

0 cot(fov/2) 0 0

0 0 Zf/(Zf-Zn) -Zn\*Zf/(Zf-Zn)

0 0 1 0

考虑到设备的宽高比，需要在x，或者y上进行调整缩放，假设x/y=alpha,则有：

cot(fov/2)/ alpha 0 0 0

0 cot(fov/2) 0 0

0 0 Zf/(Zf-Zn) -Zn\*Zf/(Zf-Zn)

0 0 1 0

关于z轴拣选

对空间中任意点(x,y,z,w)，其中w=1，在透视变换之后我们得到裁剪空间中的点为(xc,yc,zc,wc)：

zc=Zf(z-Zn)/(Zf-Zn)

而wc=z;

再进行透视除法：

zc/wc=k (1-Zn/z)，其中k=(Zf/(Zf-Zn))。

对于Zf>Zn>0,Zf>0>Zn和0>Zf>Zn三种情况，上述函数有如下函数图像：