

Transformações Projetivas

Prof. Antonio L. Apolinário Jr.
Estagiária Docente: Rafaela Alcantara

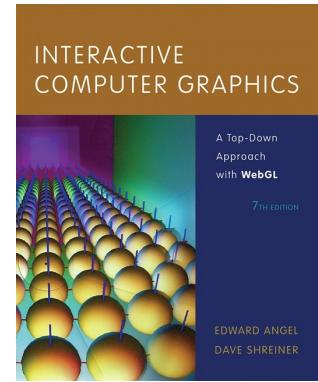
UFBA/IME/DCC/BCC - 2018.1

Roteiro

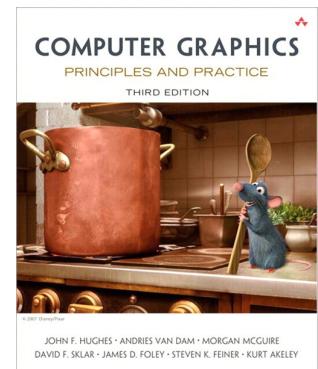
- Transformações Projetivas
 - Paralela
 - Perspectiva
- Sistema de Visualização e Projeção
- Transformação *Window Viewport*
- Aplicações em *Three.JS/WebGL*

Leitura de referencia

- Capítulo 5
Interactive Computer Graphics - A top-down approach with OpenGL
6th Edition
Angel, Edward.
Addison-Wesley. 2014.

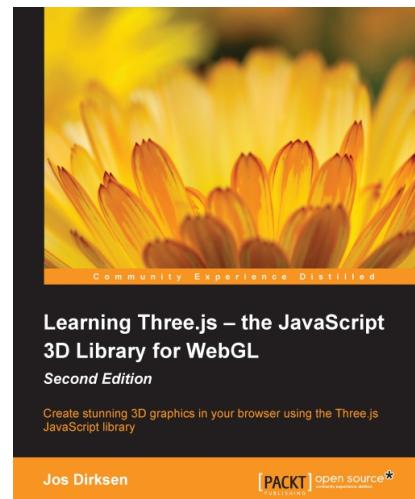


- Capitulos 13
Computer Graphics : Principles and Practice
3rd Edition
John F. Hughes / Andries van Dam
Morgan McGuire / David F. Sklar
James D. Foley / Steven K. Feiner
Addison-Wesley. 2013.



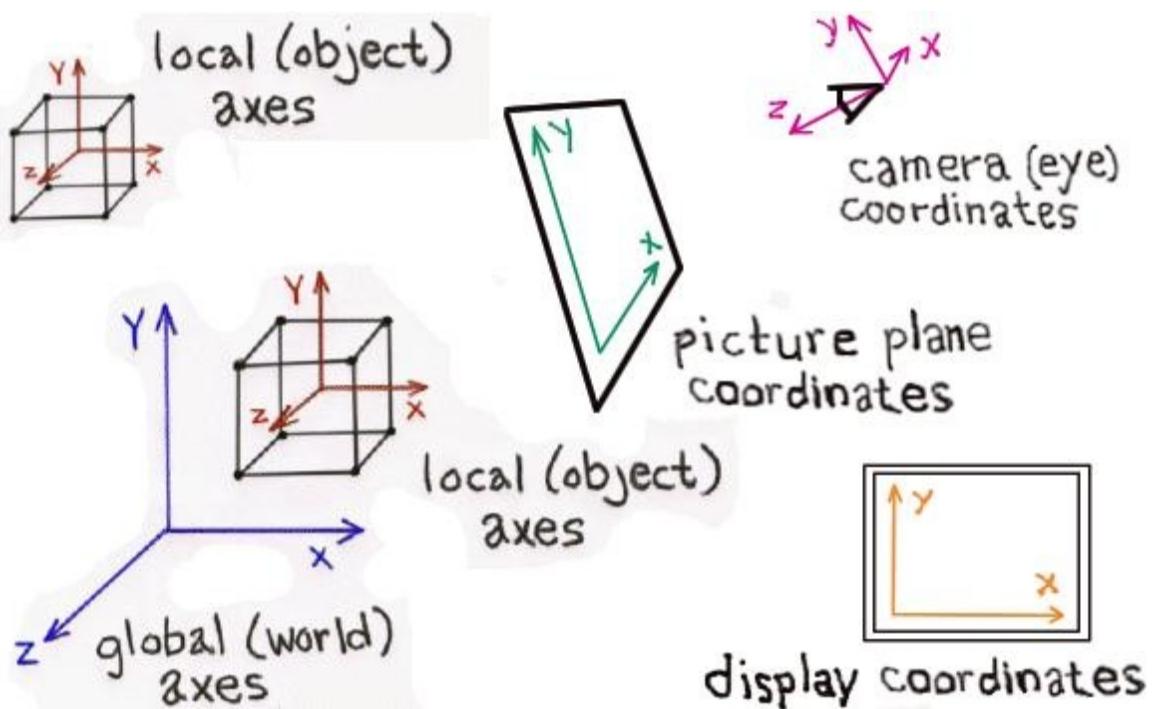
Leitura de referencia

- Capítulo 2 e 9
Learning Three.js: The JavaScript 3D Library for WebGL
Jos Dirksen
2nd Edition.
Packt Publishing - 2015.

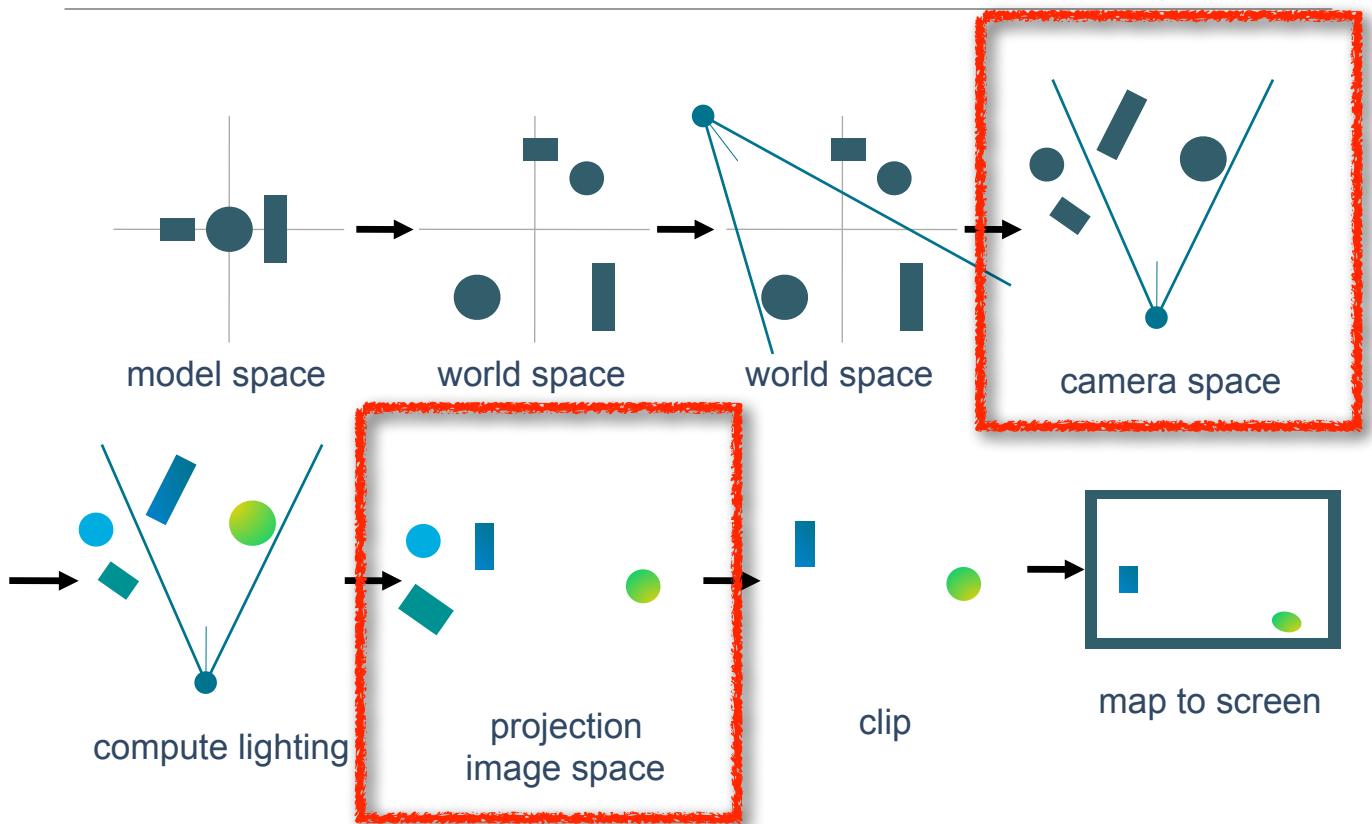


Transformações Projetivas

Projeções no contexto do *Pipeline Gráfico*



Projeções no contexto do *Pipeline Gráfico*



Projeções

- Noção intuitiva



The Invention of Drawing (1830) - Karl Friedrich Schinkel

Projeções

- Interferem na nossa percepção de relações espaciais



Projeções

- Noção de profundidade pode ser simulada



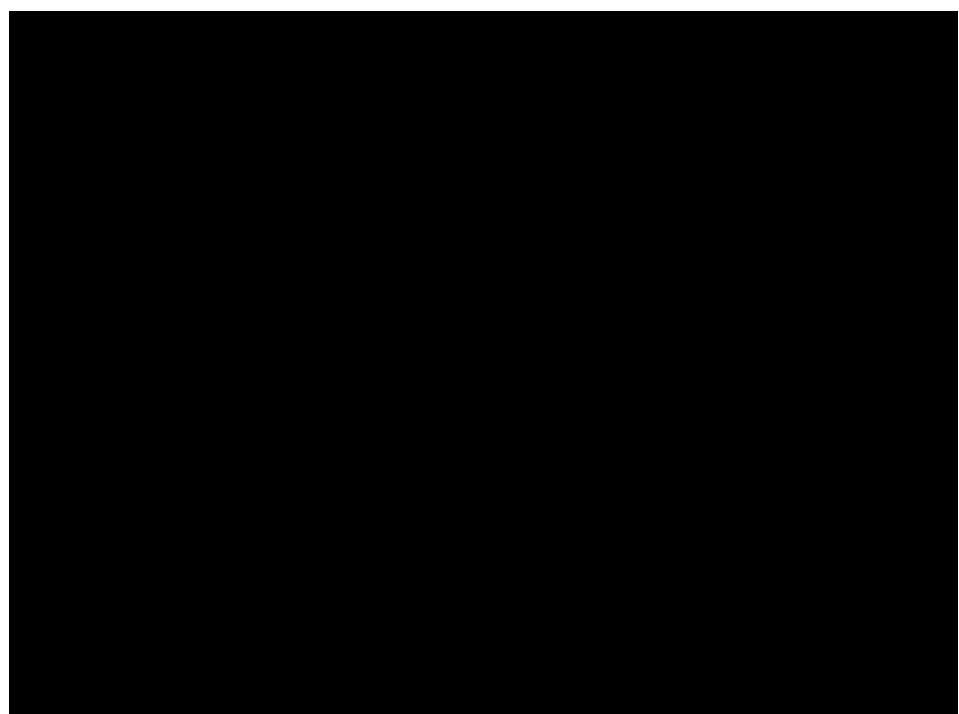
Projeções

- Noção de profundidade pode ser simulada



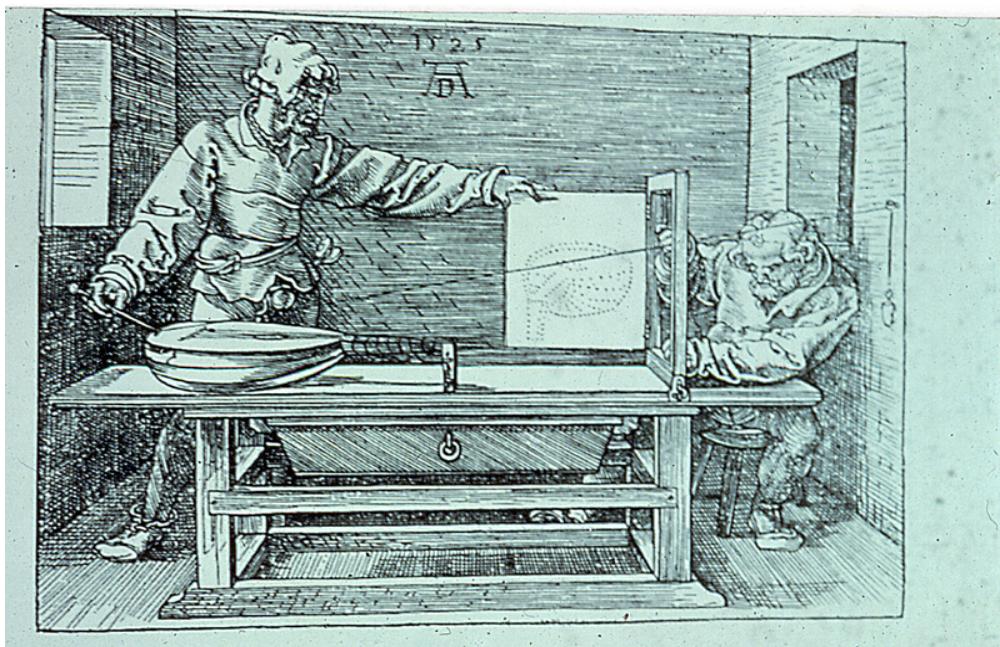
Projeções

- A noção de tamanho é relativa e pode nos enganar.



Projeções

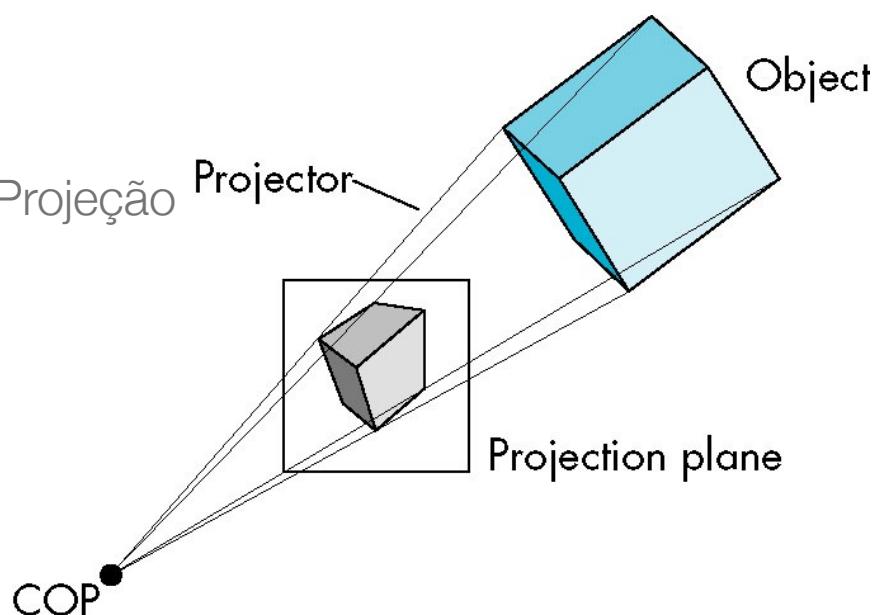
- Processo analógico de registro de objetos 3D em superfícies 2D



Projeções Planares

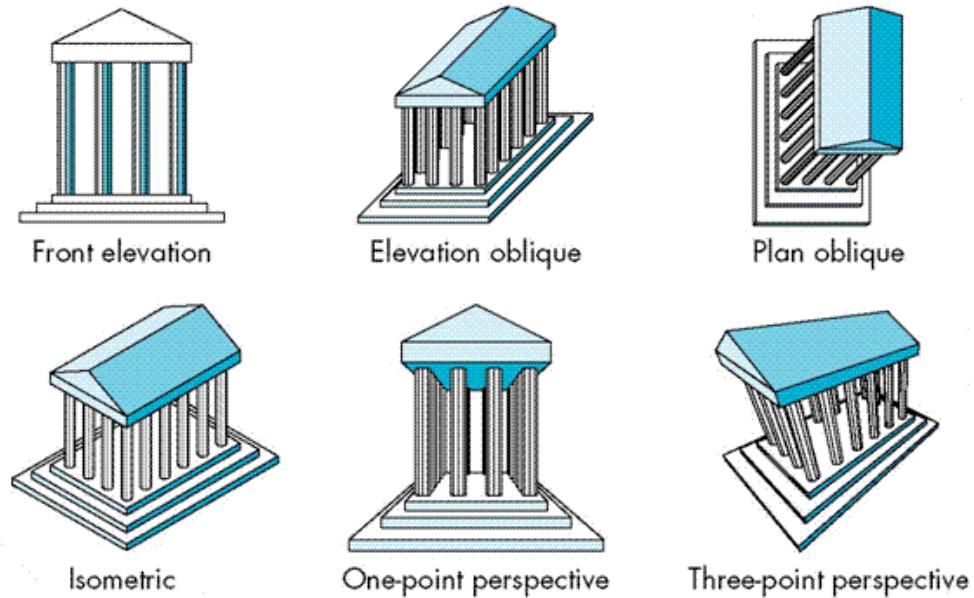
- Componentes:

- Objeto
- Superfície de Projeção
 - Plano
- Projetores
- Projeção



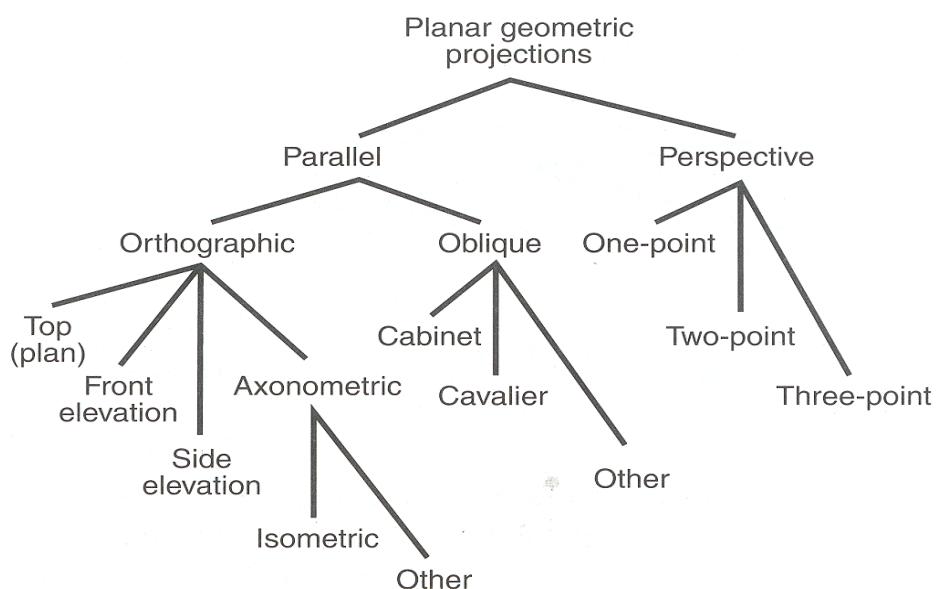
Projeções

- Diversos tipos de projeção



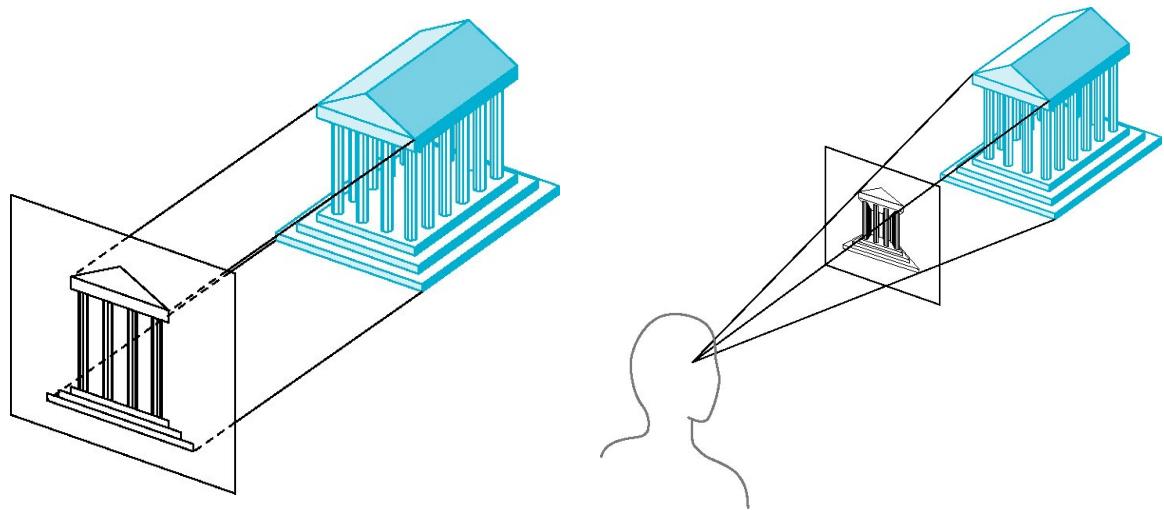
Projeções

- Taxonomia das projeção:



Projeções Planares

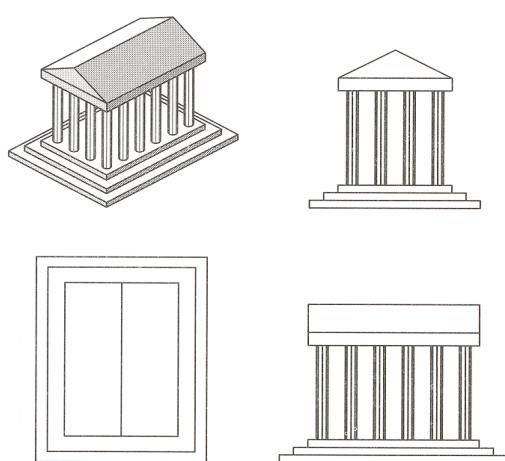
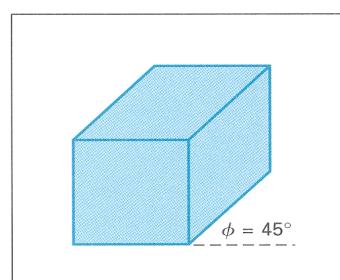
- Dois grandes grupos:
 - Forma como os projetores são definidos



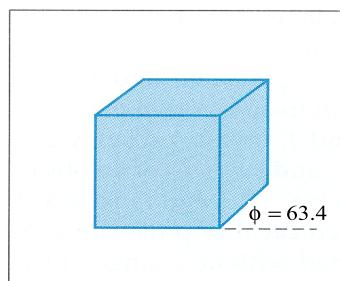
Projeções Planares

- Dois grandes grupos:
 - Paralelas

Cavalier
(DOP $\alpha = 45^\circ$)
 $\tan(\alpha) = 1$

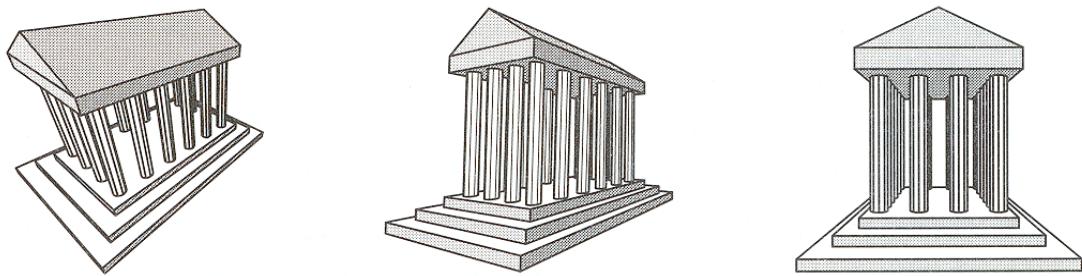


Cabinet
(DOP $\alpha = 63.4^\circ$)
 $\tan(\alpha) = 2$



Projeções Planares

- Dois grandes grupos:
 - Perspectiva
 - com 1, 2 ou 3 pontos de fuga



Projeção Perspectiva

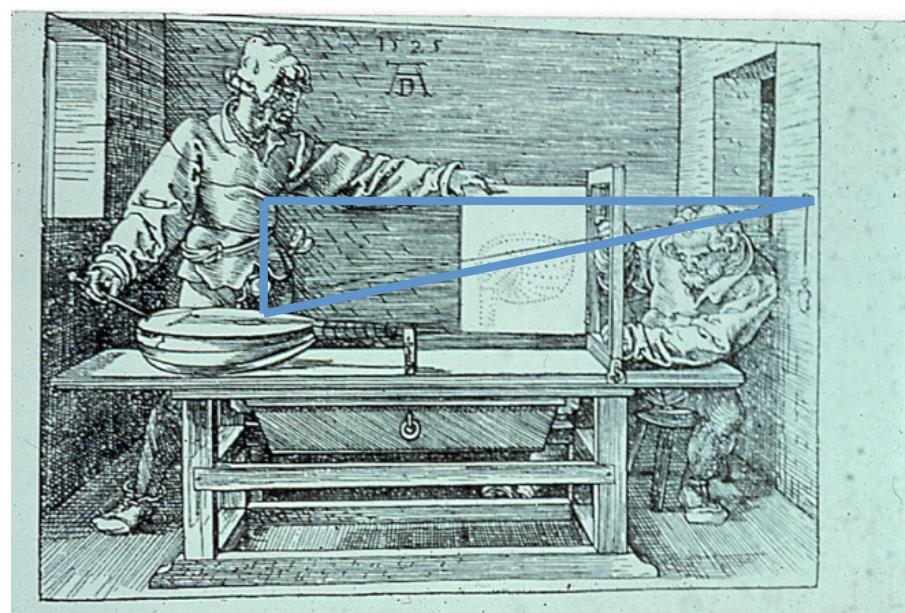
Projeção Perspectiva

- Utilizada pelo sistema visual humano
 - maior realismo



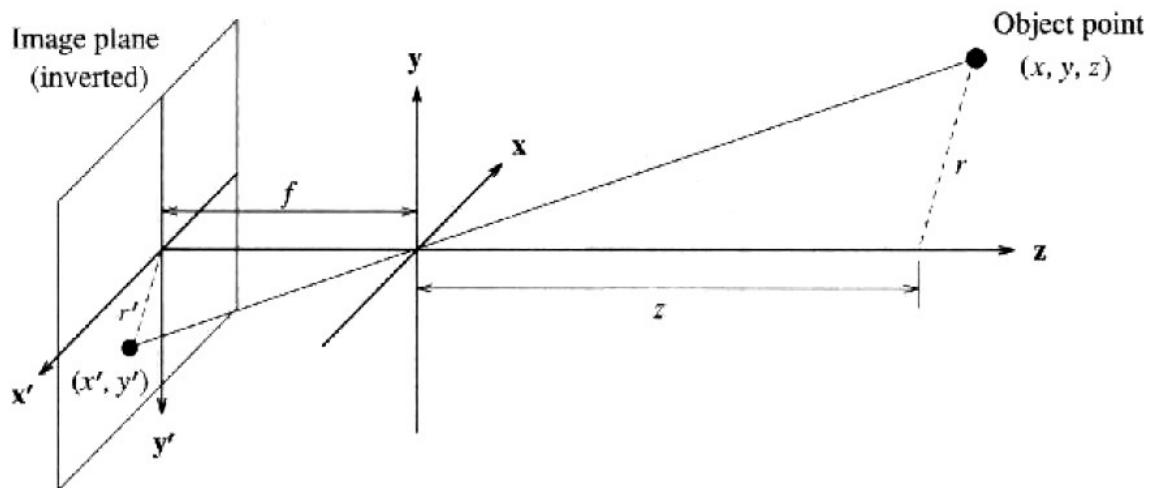
Projeção Perspectiva

- Processo de acompanhamento de um raio luminoso



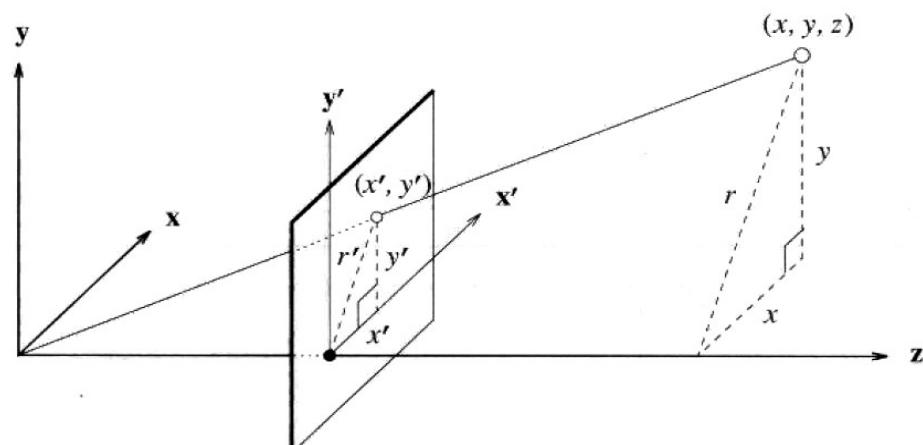
Projeção Perspectiva

- Utilizando o processo de formação de imagem de uma camera *pinhole*



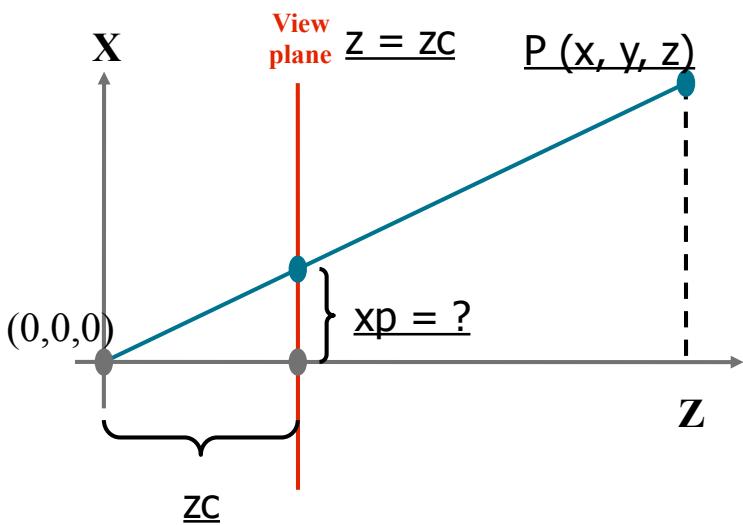
Projeção Perspectiva

- Utilizando o processo de formação de imagem de uma câmera *pinhole*
 - Reflexão do plano de projeção para obtenção de uma imagem direita



Projeção Perspectiva

- Considerando o sistema de coordenadas da câmera temos:

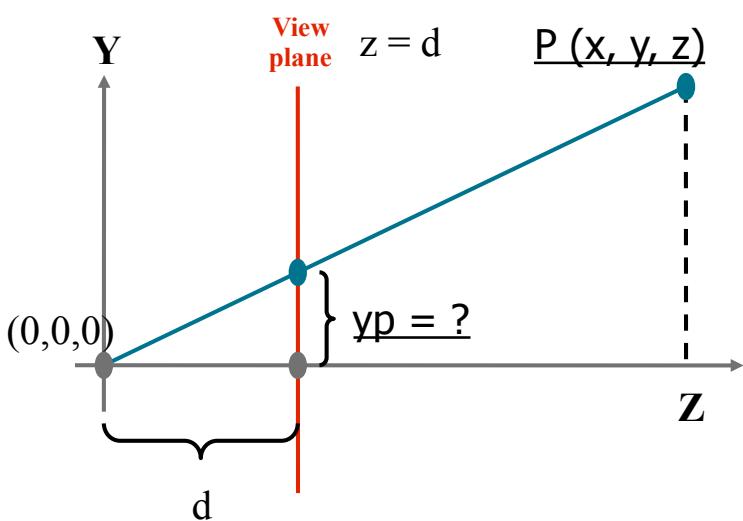


$$\frac{x_p}{d} = \frac{x}{z}$$

$$x_p = \frac{x \cdot d}{z} = \frac{x}{z/d}$$

Projeção Perspectiva

- Considerando o sistema de coordenadas da câmera temos:



$$\frac{y_p}{d} = \frac{y}{z}$$

$$y_p = \frac{y \cdot d}{z} = \frac{y}{z/d}$$

Projeção Perspectiva

- Representando com uma transformação em coordenadas homogêneas

$$\begin{bmatrix} x_p & y_p & z_p & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} M = \begin{bmatrix} x & y & d & 1 \\ \cancel{z/d} & \cancel{z/d} & 1 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1/d \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1/d \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} x & y & z & z/d \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \\ \cancel{z/d} & \cancel{z/d} & \cancel{z/d} & 1 \end{bmatrix} = \begin{bmatrix} x & y & d & 1 \\ \cancel{z/d} & \cancel{z/d} & 1 & 1 \end{bmatrix}$$

Projeção Perspectiva

- Resumindo:
 - Projetores se encontram no Centro de Projeção;
 - Preserva linhas;
 - Não preserva ângulos;
 - Não preserva medidas;
 - Equações não lineares;
 - Não inversível;
 - Mais realista
 - Tamanho da imagem inversamente proporcional a distância do objeto ao plano de projeção

Projeção Paralela

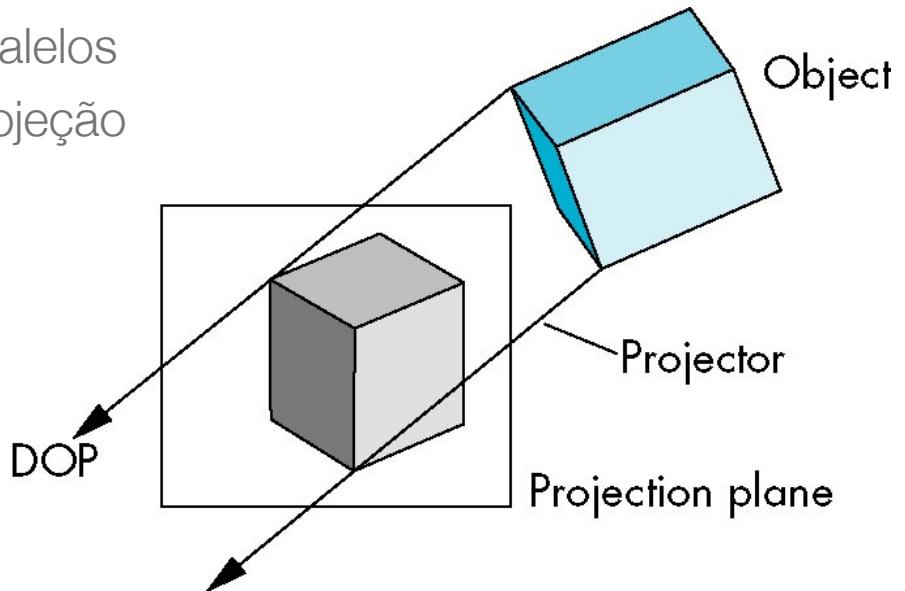
Projeção Paralela

- Utilizada em:
 - Desenho Industrial
 - Arquitetura



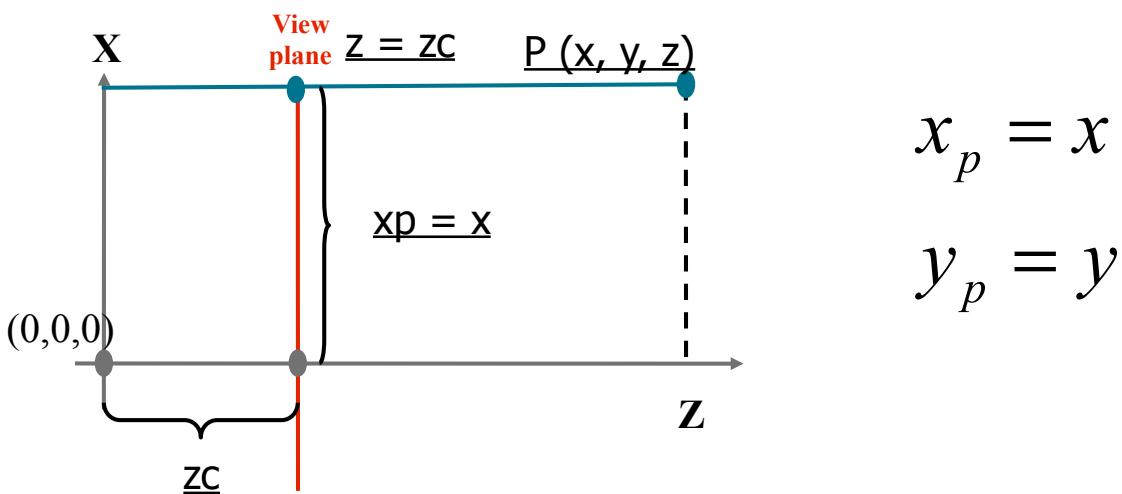
Projeção Paralela

- Projetores se encontram em um centro de projeção no infinito
 - Projetores paralelos
 - Direção de projeção



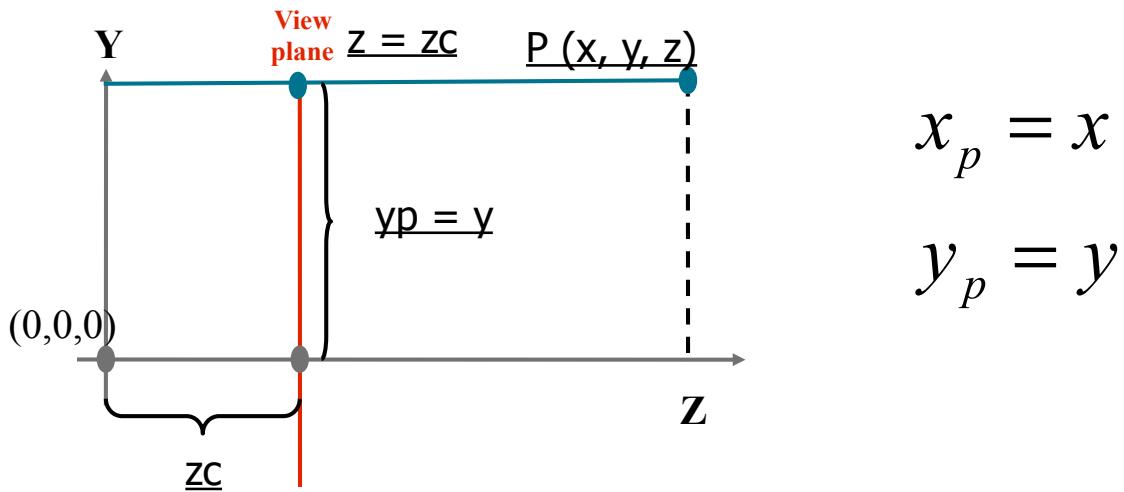
Projeção Paralela Ortogonal

- Considerando o sistema de coordenadas da câmera
 - DOP \Rightarrow perpendicular ao plano XY
 - Paralelo ao eixo Z



Projeção Paralela Ortogonal

- Considerando o sistema de coordenadas da câmera
 - DOP \Rightarrow perpendicular ao plano XY
 - Paralelo ao eixo Z



Projeção Paralela Ortogonal

- Representando com uma transformação em coordenadas homogêneas

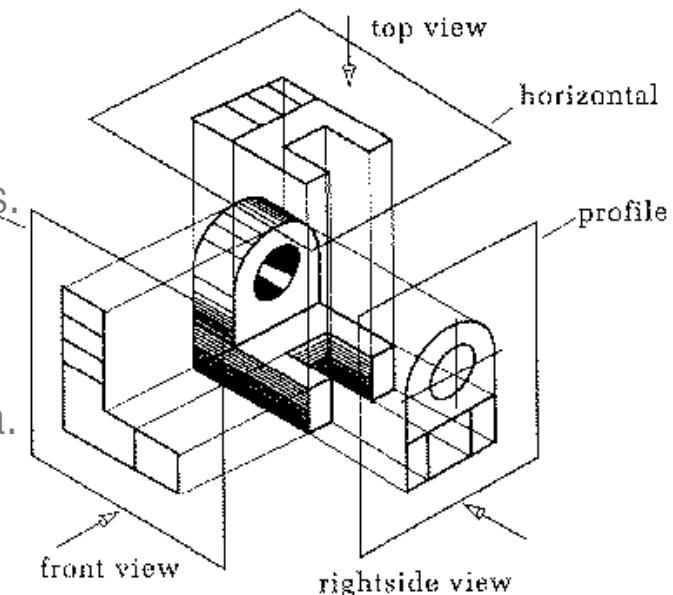
$$\begin{bmatrix} x_p & y_p & z_p & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} M = \begin{bmatrix} x & y & 0 & 1 \end{bmatrix}$$

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & 0 & 1 \end{bmatrix}$$

Projeção Paralela Ortogonal

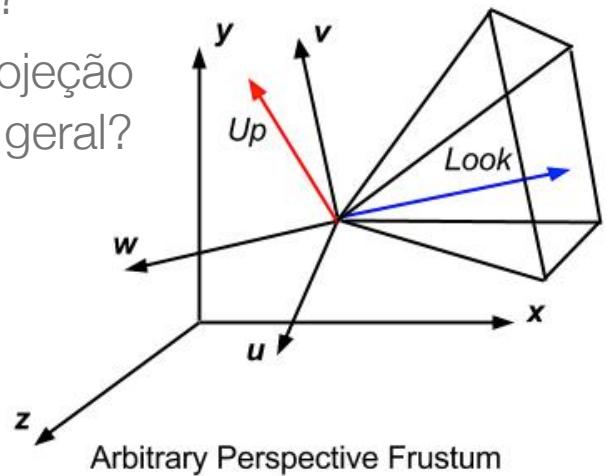
- Resumindo:
 - Projetores se encontram no infinito -> Direção de Projeção;
 - Preserva linhas;
 - Pode preservar ângulos;
 - Pode preservar medidas.
 - Equações lineares;
 - Não inversível;
 - Resultado pouco realista.



Sistema de Visualização e Projeções

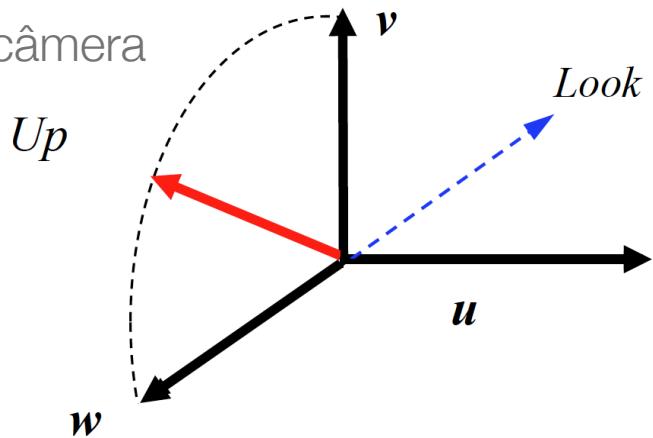
Sistema de Visualização e Projeções

- Em uma situação geral:
 - Como definir o sistema de coordenadas da camera a partir dos parâmetros da camera?
 - Como definir a matriz de projeção a partir do volume de visão geral?



Sistema de referencia da camera

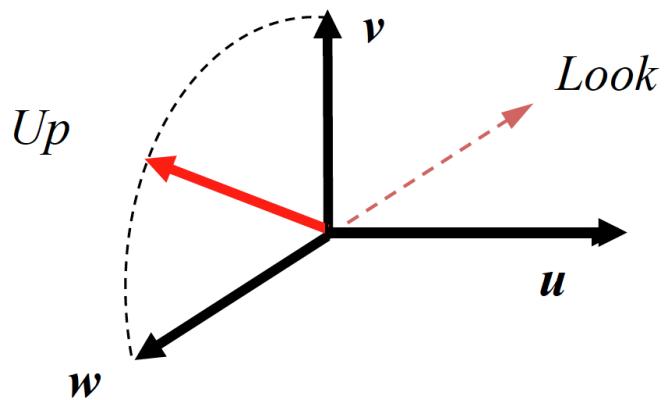
- Os parâmetros da camera:
 - **Eye-Center** => **Look**
 - **Eye-Up** => **Up**
- Devem definir os eixos do Sistema de referencia da câmera
 - **u , v , w .**



Sistema de referencia da camera

- Vetor w
 - Baseado no vetor Look
 - unitário

$$w = \frac{-Look}{\|Look\|}$$



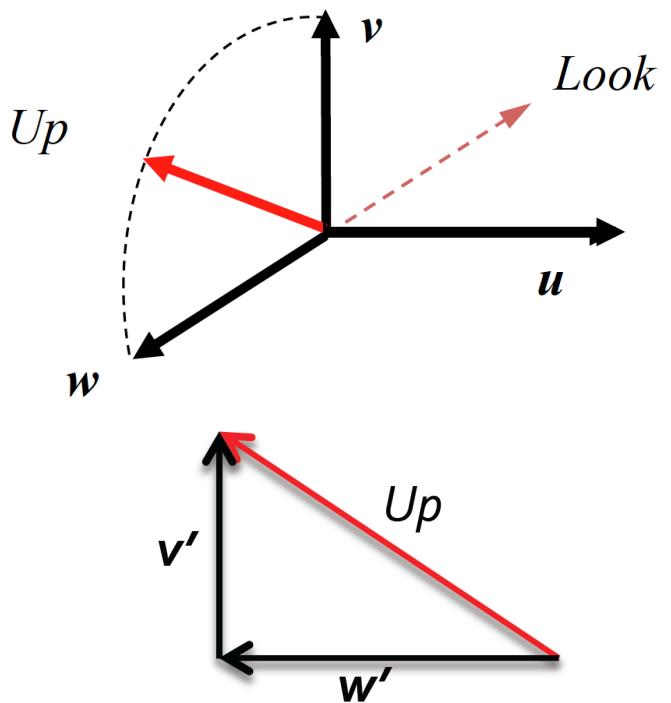
Sistema de referencia da camera

- Vetor v
 - Perpendicular a w
 - Unitário

$$Up = w' + v'$$

$$v' = Up - (Up \cdot w)w$$

$$v = \frac{v'}{\|v'\|}$$

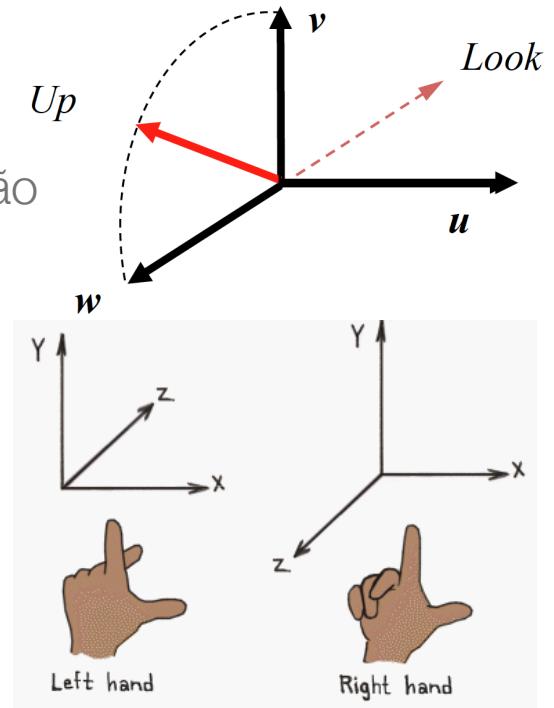


Sistema de referencia da camera

- Vetor \mathbf{u}

- Perpendicular ao plano \mathbf{vw}
- Unitário
- Definido pela orientação padrão

$$\mathbf{u} = \mathbf{v} \times \mathbf{w}$$



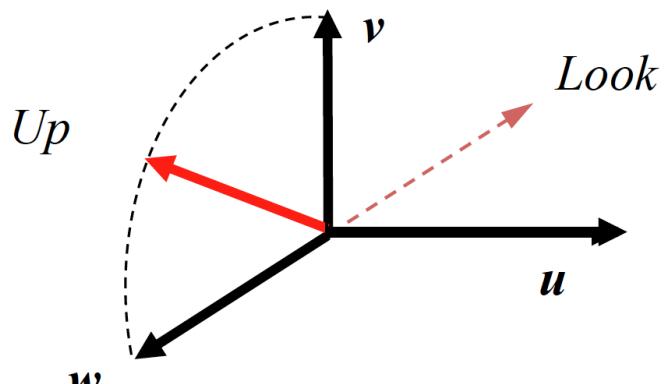
Sistema de referencia da camera

- Resumindo:

$$\mathbf{w} = \frac{-\mathbf{Look}}{\|\mathbf{Look}\|}$$

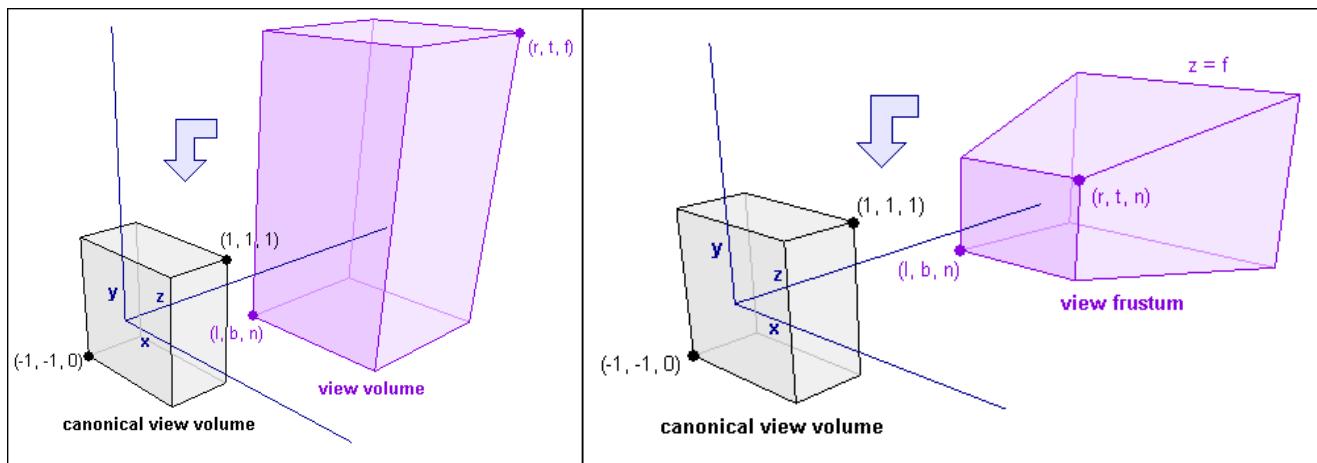
$$\mathbf{v} = \frac{\mathbf{Up} - (\mathbf{Up} \cdot \mathbf{w})\mathbf{w}}{\|\mathbf{Up} - (\mathbf{Up} \cdot \mathbf{w})\mathbf{w}\|}$$

$$\mathbf{u} = \mathbf{v} \times \mathbf{w}$$



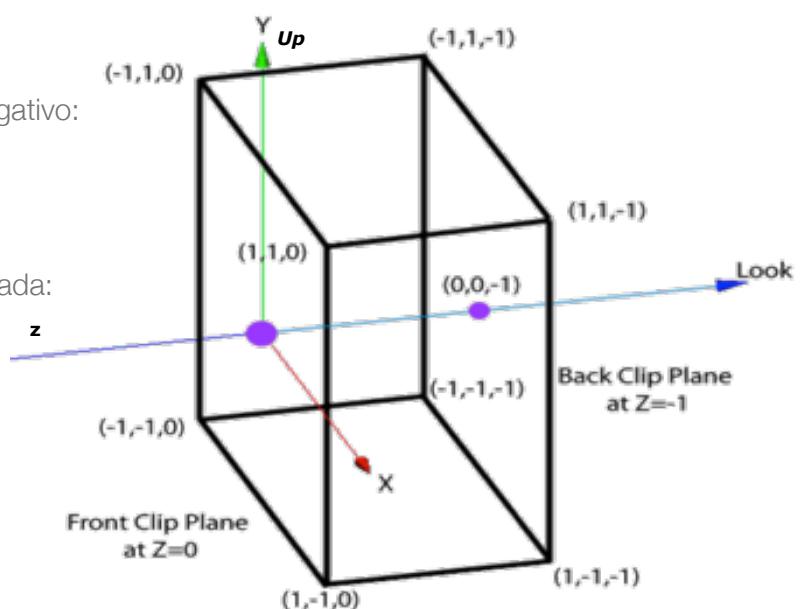
Volume de Visão

- Volumes de visão genéricos dificultam as contas a serem feitas no processo de recorte
 - Uso de volumes de visão canônicos / padrão / unitários



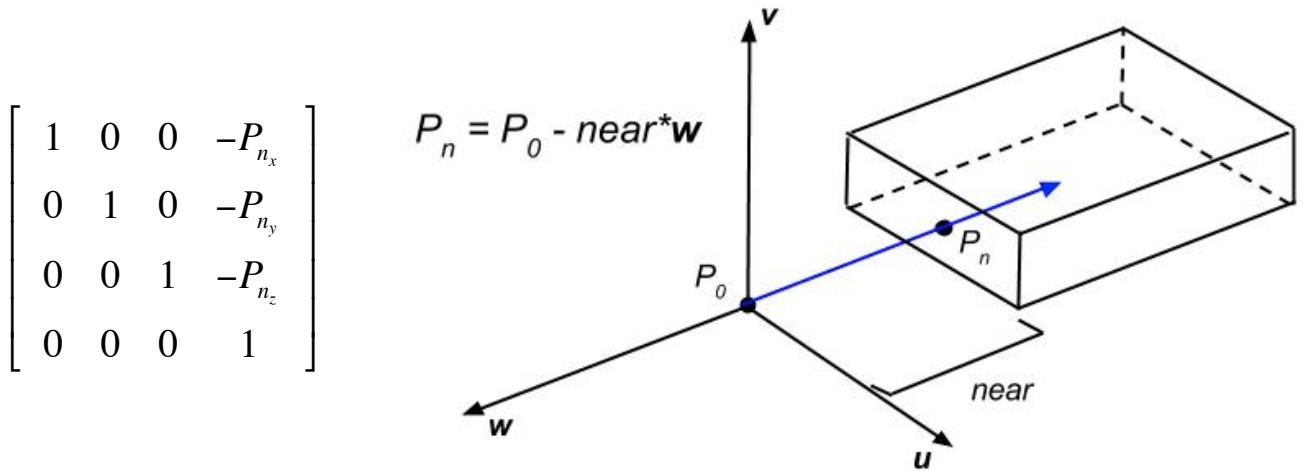
Volume de Visão Canônico - Projeção paralela

- Volume de visão canônico
 - Centroado na origem:
 - *near clipping plane* = $(0,0,0)$
 - Direção de Observação eixo z-negativo:
 - **Look vector** = $(0,0,-1)$
 - Orientado para cima:
 - **Up vector** = $(0,1,0)$
 - Limite da janela de visão normalizada:
 - -1 a 1 nas direções **x** e **y**
 - Profundidade de campo:
 - **near plane** em $z = 0$
 - **far plane** em $z = -1$



Normalizando o Volume de Visão Projeção Paralela

1. Levar o centro do near plane para a origem



Normalizando o Volume de Visão Projeção Paralela

2. Alinhar os eixos u , v e w com os eixos x , y e z do volume de visão Canônico

- Encontrar rotações tais que:

$$R_{rot}u = e_1 \quad R_{rot}v = e_2 \quad R_{rot}w = e_3$$

- onde

$$e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad e_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad e_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

- Ou seja:

- $u = R_{rot}^{-1} e_1$

- $v = R_{rot}^{-1} e_2$

- $w = R_{rot}^{-1} e_3$

Normalizando o Volume de Visão Projeção Paralela

2. Alinhar os eixos \mathbf{u} , \mathbf{v} e \mathbf{w} com os eixos \mathbf{x} , \mathbf{y} e \mathbf{z} do volume de visão Canônico

- $\mathbf{u} = \mathbf{R}_{rot}^{-1} \mathbf{e1}$
- $\mathbf{v} = \mathbf{R}_{rot}^{-1} \mathbf{e2}$
- $\mathbf{w} = \mathbf{R}_{rot}^{-1} \mathbf{e3}$

$$\mathbf{R}_{rot} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_{rot}^{-1} = \begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix}$$

$$\mathbf{R}_{rot} = \begin{bmatrix} u_x & u_y & u_z \\ v_x & v_y & v_z \\ w_x & w_y & w_z \end{bmatrix}$$

Normalizando o Volume de Visão Projeção Paralela

3. Escalar o volume para que fique no intervalo de -1 a 1 nas direções \mathbf{x} e \mathbf{y} e o **far plane** em $\mathbf{z} = 1$

$$\begin{bmatrix} \frac{2}{width} & 0 & 0 & 0 \\ 0 & \frac{2}{height} & 0 & 0 \\ 0 & 0 & \frac{1}{far} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Normalizando o Volume de Visão Projeção Paralela

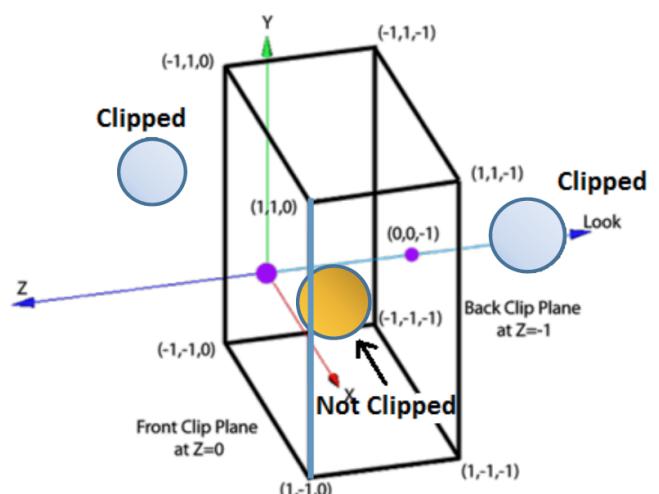
- Resumindo:
 - Translação de \mathbf{P}_n (centro do plano mais próximo) para a origem - \mathbf{T}_{trans}
 - Alinhamento dos eixos $\mathbf{u}, \mathbf{v}, \mathbf{w}$ com os eixos $\mathbf{x}, \mathbf{y}, \mathbf{z}$ - \mathbf{R}_{rot}
 - Escala do volume de visão para as dimensões $[-1,1] \times [-1,1] \times [0,-1]$ - \mathbf{S}_{xyz}

$$\left[\begin{array}{cccc} 2/\text{width} & 0 & 0 & 0 \\ 0 & 2/\text{height} & 0 & 0 \\ 0 & 0 & 1/\text{far} & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \left[\begin{array}{cccc} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \left[\begin{array}{cccc} 1 & 0 & 0 & -P_{n_x} \\ 0 & 1 & 0 & -P_{n_y} \\ 0 & 0 & 1 & -P_{n_z} \\ 0 & 0 & 0 & 1 \end{array} \right]$$

Volume de Visão Canônico - Projeção Paralela

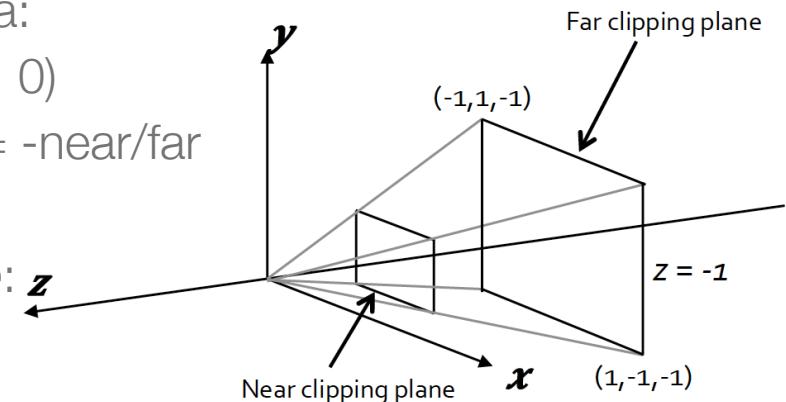
- Resumindo:
 - Projeção feita no Volume de Visão Canônico
 - Recorte feito no Volume de Visão Canônico

$$\left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} x \\ y \\ z \\ 1 \end{array} \right] = \left[\begin{array}{c} x \\ y \\ 0 \\ 1 \end{array} \right]$$



Volume de Visão Canônico - Projeção Perspectiva

- Centro na origem:
 - **Eye** = (0, 0, 0)
- Direção de visualização no eixo z negativo:
 - **Look vector** = (0, 0, -1)
- Orientação para cima:
 - **Up vector** = (0, 1, 0)
- **Near plane** : $z = c = -\text{near/far}$
- **Far plane** : $z = -1$
- Limites do **far plane**: $z \in [-1, 1]$



Normalizando o Volume de Visão Projeção Perspectiva

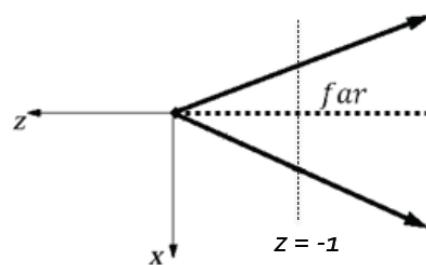
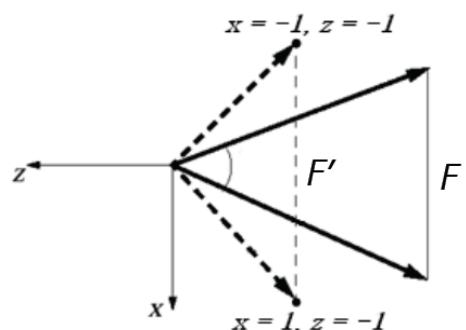
- Primeiras etapas comuns a normalização da projeção paralela
 1. Translação do centro de projeção (P_0) para a origem;
 2. Alinhamento dos eixos u, v, w aos eixos canônicos x, y, z

$$T_{trans} = \begin{bmatrix} 1 & 0 & 0 & -P_{0_x} \\ 0 & 1 & 0 & -P_{0_y} \\ 0 & 0 & 1 & -P_{0_z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_{rot} = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Normalizando o Volume de Visão Projeção Perspectiva

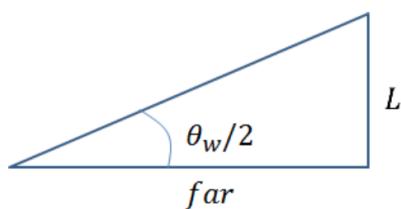
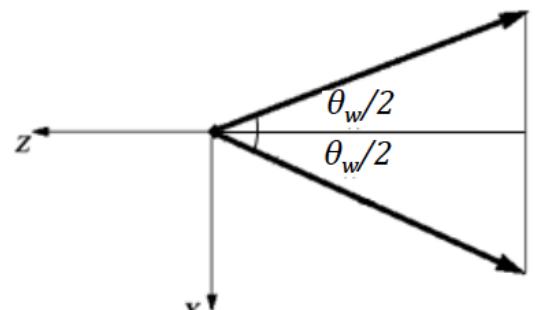
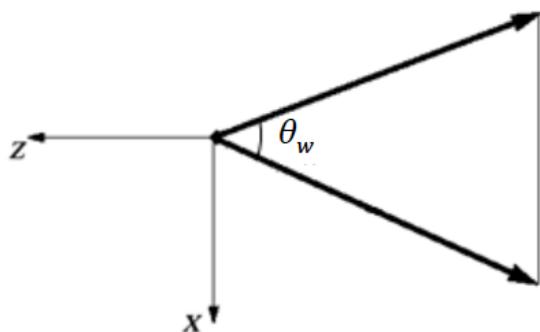
3. Escala do Frustum:

- Limites do **far plane** em $[-1, 1]$
- **Far plane** em $z=-1$



Normalizando o Volume de Visão Projeção Perspectiva

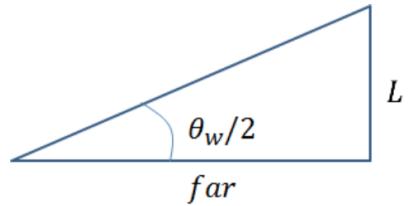
- Limites do **far plane** em $[-1, 1]$
 - Obtidos a partir do **fov_x/fov_y** /razão de aspecto



Normalizando o Volume de Visão Projeção Perspectiva

- Limites do **far plane** em [-1, 1]
 - Obtidos a partir do **fov_x/fov_y** / razão de aspecto

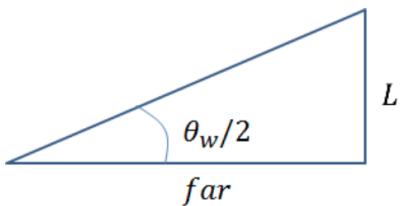
$$\frac{L}{far} = \tan\left(\frac{\theta_w}{2}\right) \rightarrow L = far \tan\left(\frac{\theta_w}{2}\right)$$



$$Scale_x = \frac{1}{far \tan\left(\frac{\theta_w}{2}\right)} \quad Scale_y = \frac{1}{far \tan\left(\frac{\theta_h}{2}\right)}$$

Normalizando o Volume de Visão Projeção Perspectiva

- **Far plane** em z = -1
 - **Scale_z = 1 / far**
- Matriz de Escala final:



$$S_{xyz} = \begin{bmatrix} 1 / far \tan\left(\theta_w / 2\right) & 0 & 0 & 0 \\ 0 & 1 / far \tan\left(\theta_h / 2\right) & 0 & 0 \\ 0 & 0 & 1 / far & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

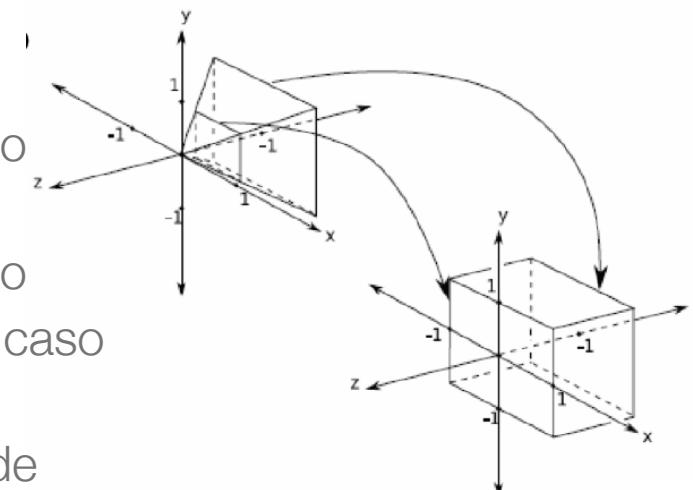
Normalizando o Volume de Visão Projeção Perspectiva

- Resumindo:
 - Translação de P_0 (centro de projeção) para a origem - T_{trans}
 - Alinhamento dos eixos u, v, w com os eixos x, y, z - R_{rot}
 - Escala do volume de visão para as dimensões $[-1,1] \times [-1,1] \times [0,-1]$ - S_{xyz}

$$S_{xyz} R_{rot} T_{trans} = \begin{bmatrix} 1/far \tan(\theta_w/2) & 0 & 0 & 0 \\ 0 & 1/far \tan(\theta_h/2) & 0 & 0 \\ 0 & 0 & 1/far & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -P_{0x} \\ 0 & 1 & 0 & -P_{0y} \\ 0 & 0 & 1 & -P_{0z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

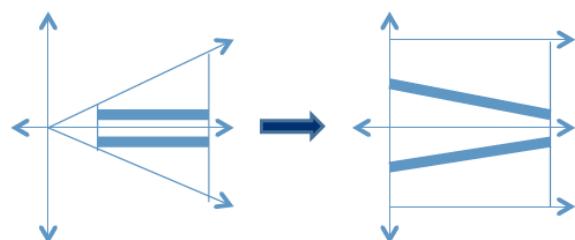
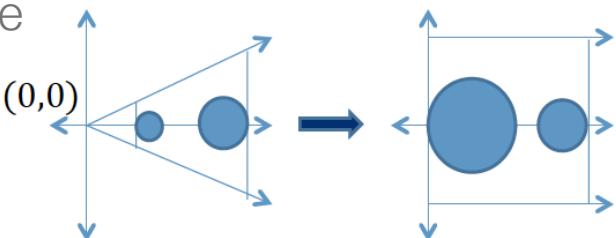
Volume de Visão Canônico - Projeção Perspectiva

- Resumindo:
 - Projeção feita no Volume de Visão Canônico
 - Recorte feito no Volume de Visão Canônico
 - Mais complexo que no caso paralelo
 - Transformar o Volume de Visão Canônico Perspectiva em Volume de Visão Canônico Paralelo



Volume de Visão Canônico Perspectiva em Paralelo

- Transformar o Volume de Visão Perspectiva em Paralelo
deforma os objetos em seu interior
 - Mudança de escala.
 - Linhas paralelas tornam-se convergentes.



Volume de Visão Canônico Perspectiva em Paralelo

- Primeiro passo:
 - Ajuste do centro do **near plane (P_n')** para a origem

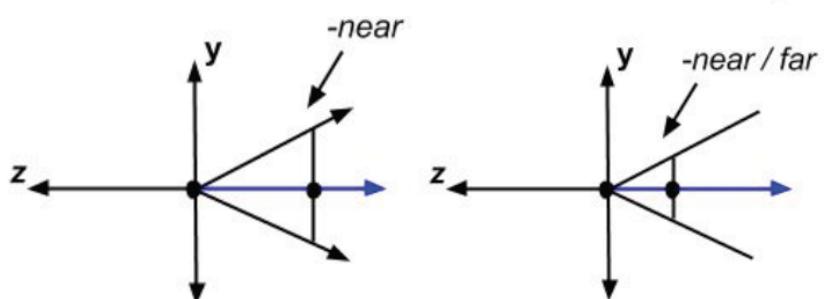
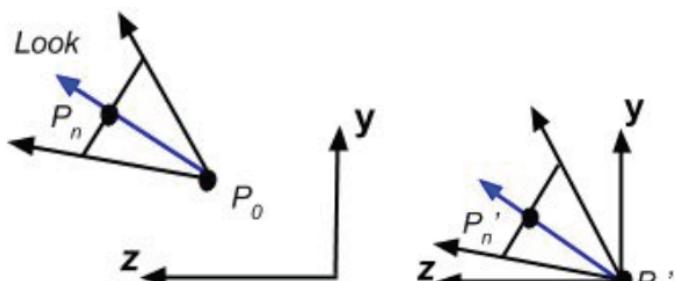
$$P_n = P_0 + near * Look'$$

$$P_n' = S_{xyz} R_{rot} T_{trans} P_n$$

$$P_n' = \begin{pmatrix} 0 & 0 & c \end{pmatrix}$$

- onde

- $c = -near/far$



Volume de Visão Canônico Perspectiva em Paralelo

- Segundo passo:
 - Deformação do Volume
- Matriz de transformação Perspectiva em Paralelo (M_{pp}):
 - onde $c = -\text{near/far}$

$$M_{pp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1+c} & \frac{-c}{1+c} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

Volume de Visão Canônico Perspectiva em Paralelo

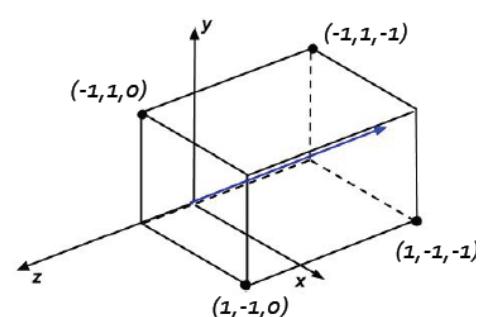
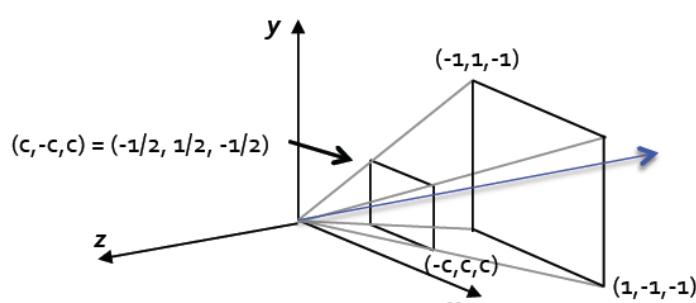
- Exemplo: $c = -1/2$

$$\left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & -1 & 0 \end{array} \right] \left[\begin{array}{c} -1 \\ 1 \\ -1 \\ 1 \end{array} \right] = \left[\begin{array}{c} -1 \\ 1 \\ -1 \\ 1 \end{array} \right]$$

$$\left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & -1 & 0 \end{array} \right] \left[\begin{array}{c} 1 \\ -1 \\ -1 \\ 1 \end{array} \right] = \left[\begin{array}{c} 1 \\ -1 \\ -1 \\ 1 \end{array} \right]$$

$$\left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & -1 & 0 \end{array} \right] \left[\begin{array}{c} -1/2 \\ 1/2 \\ -1/2 \\ 1 \end{array} \right] \xrightarrow{\text{hom.}} \left[\begin{array}{c} -1 \\ 1 \\ 0 \\ 1 \end{array} \right]$$

$$\left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & -1 & 0 \end{array} \right] \left[\begin{array}{c} 1/2 \\ -1/2 \\ -1/2 \\ 1 \end{array} \right] \xrightarrow{\text{hom.}} \left[\begin{array}{c} 1 \\ -1 \\ 0 \\ 1 \end{array} \right]$$



Normalizando o Volume de Visão Projeção Perspectiva

- Resumindo:
 - Translação de \mathbf{P}_0 (centro de projeção) para a origem - \mathbf{T}_{trans}
 - Alinhamento dos eixos $\mathbf{u}, \mathbf{v}, \mathbf{w}$ com os eixos $\mathbf{x}, \mathbf{y}, \mathbf{z}$ - \mathbf{R}_{rot}
 - Escala do volume de visão para as dimensões $[-1,1] \times [-1,1] \times [0,-1]$ - \mathbf{S}_{xyz}
 - Deformação do Volume Canonico Perspectiva em paralelo

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1+c} & \frac{-c}{1+c} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 1/far \tan(\theta_w/2) & 0 & 0 & 0 \\ 0 & 1/far \tan(\theta_h/2) & 0 & 0 \\ 0 & 0 & 1/far & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -P_{0_x} \\ 0 & 1 & 0 & -P_{0_y} \\ 0 & 0 & 1 & -P_{0_z} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Homogeneização das coordenadas deformadas
- Projeção Paralela

Efeitos Colaterais

- Compressão das coordenadas \mathbf{z} dos objetos:

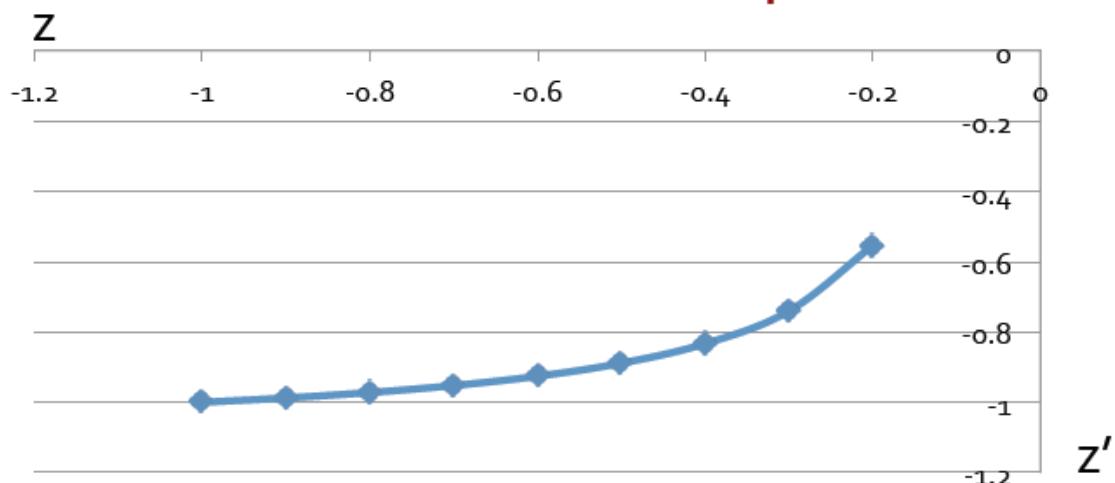
$$M_{pp} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1+c} & \frac{-c}{1+c} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \frac{z-c}{1+c} \\ -z \end{bmatrix} \xrightarrow{\text{hom.}} \begin{bmatrix} -x/z \\ -y/z \\ \frac{c-z}{z+zc} \\ 1 \end{bmatrix}$$

Efeitos Colaterais

- O que acontece se:
 - **Near plane** se aproximar de $z = 0$?
 - **Far plane** estiver bastante distante ($z = \infty$)?
 - Lembrando que $c = -\text{near}/\text{far}$
- **c** tende a **0** !!
 - Portanto se $z' = (c-z)/(z+zc)$, então z' tende a:
 - $z' = -z/z = -1$
 - Ou seja, todos os pontos no intervalo entre **znear** e **zfar** terão coordenada $z' = -1$!

Efeitos Colaterais

- Exemplo:
 - Considere **near = 0.1, far = 1**, logo $c = -0.1$



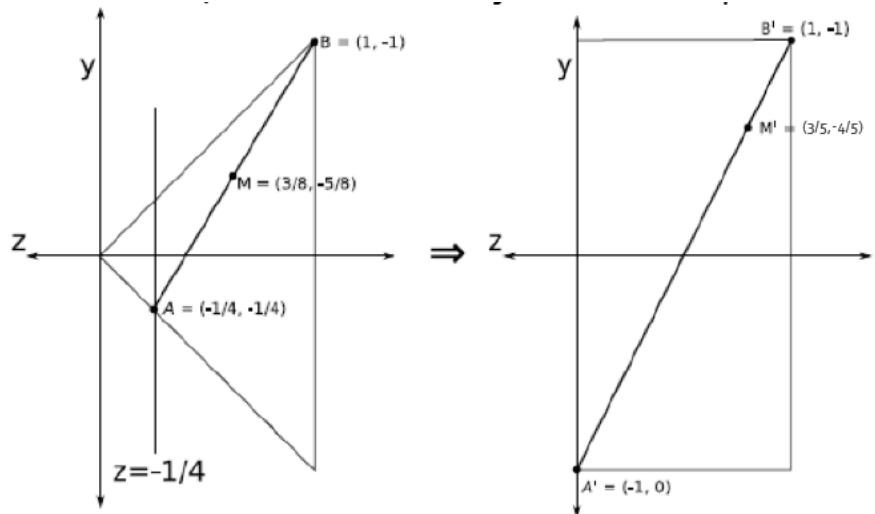
Efeitos Colaterais

- Calculo da interpolação linear dos vértices:

- Exemplo:

- M é ponto médio

- M' não é!



Efeitos Colaterais

- Calculo da interpolação linear dos vértices:

- Solução:

- Aplicar a interpolação antes de homogeneização das coordenadas

$$M_{pp} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{1+c} & \frac{-c}{1+c} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \frac{z-c}{1+c} \\ -z \end{bmatrix} \xrightarrow{\text{hom.}} \begin{bmatrix} -x/z \\ -y/z \\ \frac{c-z}{z+zc} \\ 1 \end{bmatrix}$$

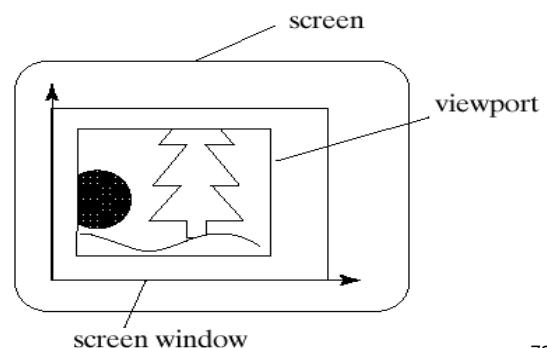
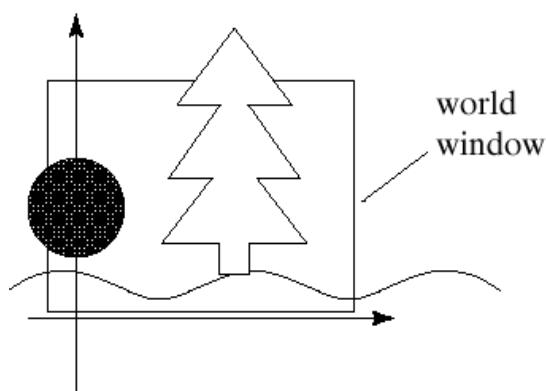
- $W = -Z$

Transformação *Window Viewport*

69

Transformação *Window Viewport*

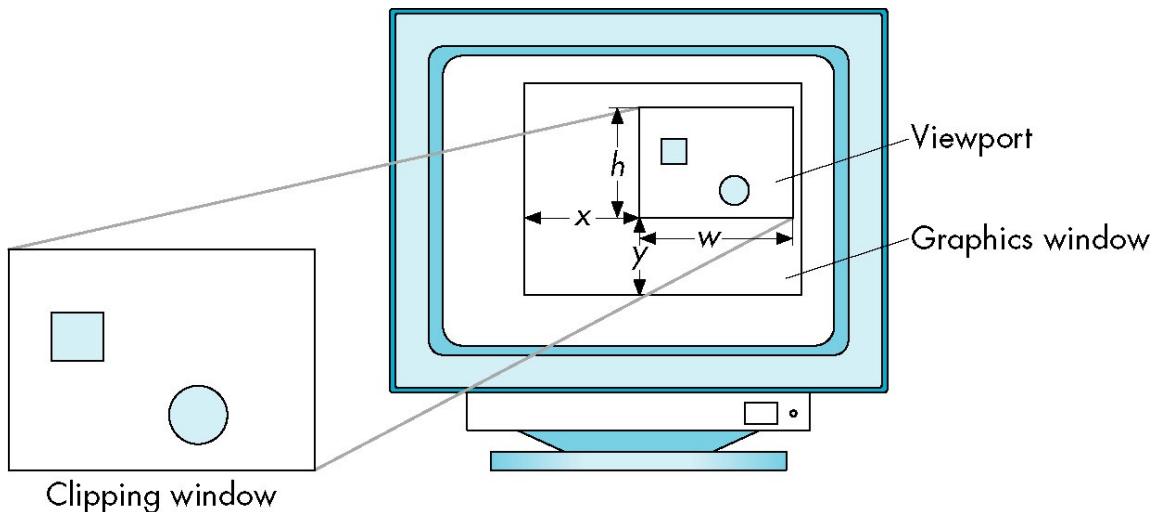
- *Window*
 - Região do espaço do “mundo” visível
 - Coordenadas reais
 - Independente de unidade
- *Viewport*
 - Área da janela física onde a imagem do mundo será mostrada
 - Coordenadas inteiras positivas
 - Unidade: pixels



70

Transformação Window Viewport

- Window x Viewport



71

Transformação Window Viewport

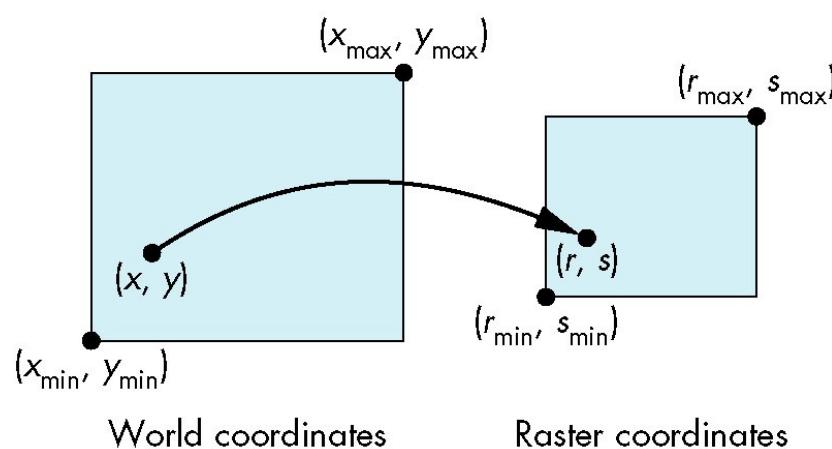
- Considerando:

- Window

$$(X_{\min}, Y_{\min}) - (X_{\max}, Y_{\max})$$

- Viewport

$$(r_{\min}, s_{\min}) - (r_{\max}, s_{\max})$$



72

Transformação *Window Viewport*

- Sequência de Transformações Geométricas

$$\begin{aligned}
 M_{jv} &= T(r_{\min}, s_{\min}) \cdot E \begin{pmatrix} \frac{r_{\max} - r_{\min}}{x_{\max} - x_{\min}} & \frac{s_{\max} - s_{\min}}{y_{\max} - y_{\min}} \\ 0 & 0 \end{pmatrix} \cdot T(-x_{\min}, -y_{\min}) = \\
 &= \begin{bmatrix} 1 & 0 & r_{\min} \\ 0 & 1 & s_{\min} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \frac{r_{\max} - r_{\min}}{x_{\max} - x_{\min}} & 0 & 0 \\ 0 & \frac{s_{\max} - s_{\min}}{y_{\max} - y_{\min}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_{\min} \\ 0 & 1 & -y_{\min} \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

73

Transformação *Window Viewport*

- Transformações Geométricas final

$$M_{jv} = \begin{bmatrix} \frac{r_{\max} - r_{\min}}{x_{\max} - x_{\min}} & 0 & -x_{\min} \cdot \frac{r_{\max} - r_{\min}}{x_{\max} - x_{\min}} + r_{\min} \\ 0 & \frac{s_{\max} - s_{\min}}{y_{\max} - y_{\min}} & -y_{\min} \cdot \frac{s_{\max} - s_{\min}}{y_{\max} - y_{\min}} + s_{\min} \\ 0 & 0 & 1 \end{bmatrix}$$

74

Transformação *Window Viewport*

- Transformações Geométricas final

$$P = M_{jv} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} (x - x_{\min}) \cdot \frac{r_{\max} - r_{\min}}{x_{\max} - x_{\min}} + r_{\min} \\ (y - y_{\min}) \cdot \frac{s_{\max} - s_{\min}}{y_{\max} - y_{\min}} + s_{\min} \\ 1 \end{bmatrix}$$

75

Aplicações com *Three.JS/WebGL*

76

Tipo de projeção no Three.JS

- Define o processo de formação da imagem do objeto
Camera
 - Propriedades
 - **projectionMatrix**

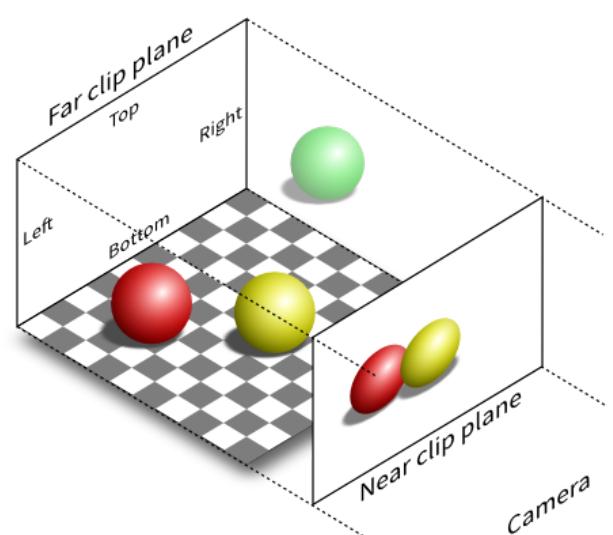
77

Tipo de projeção no Three.JS

- **OrthographicCamera**

- Propriedades
 - **left**
 - **right**
 - **top**
 - **bottom**
 - **near**
 - **far**

- Metodos
 - **updateProjectionMatrix()**



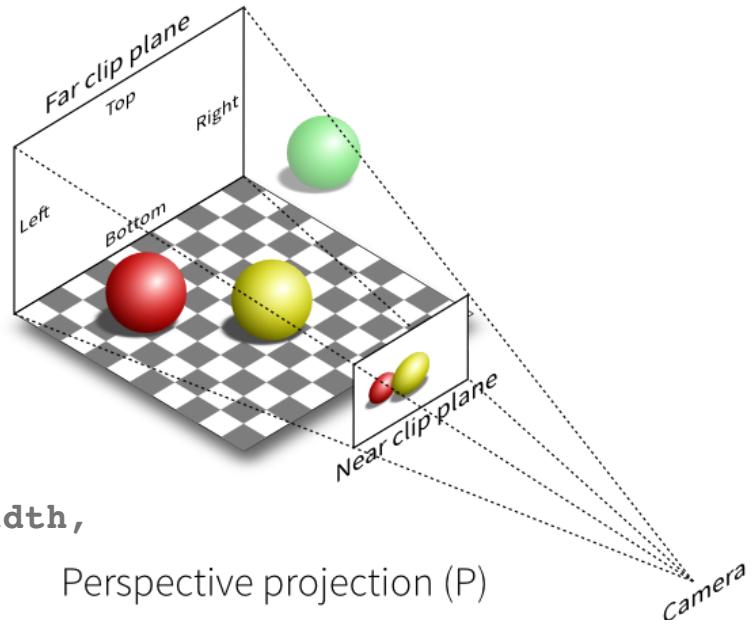
Orthographic projection (O)

78

Tipo de projeção no Three.JS

- **PerspectiveCamera**

- Propriedades
 - **fov**
 - **aspect**
 - **near**
 - **far**
- Métodos
 - **setLens(focalLength, frameSize)**
 - **setViewOffset(fullWidth, fullHeight, x, y, width, height)**
 - **updateProjectionMatrix()**



A seguir... Visibilidade