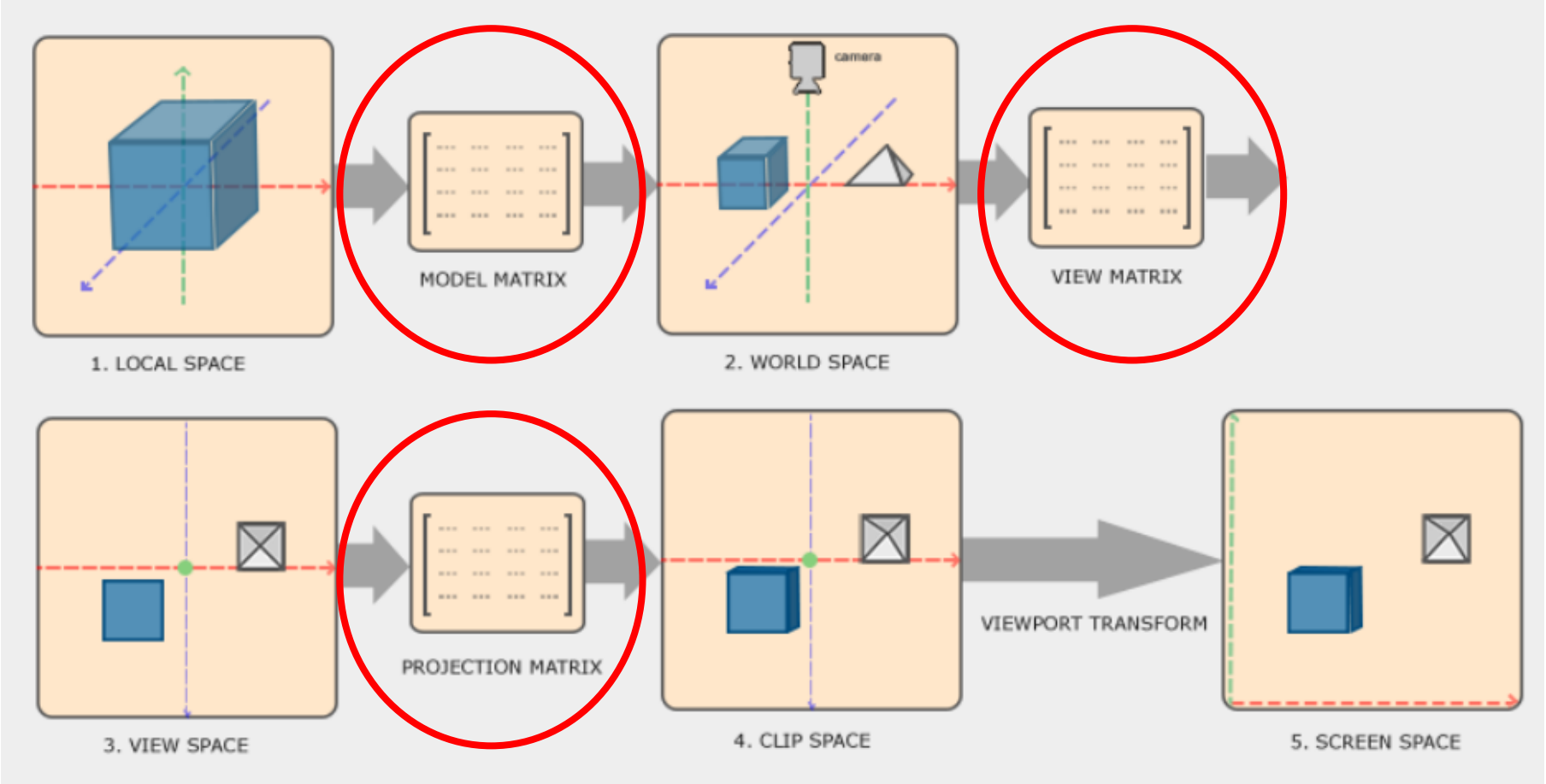
Computação Gráfica

Aula 10 – *Projection* e *viewport*

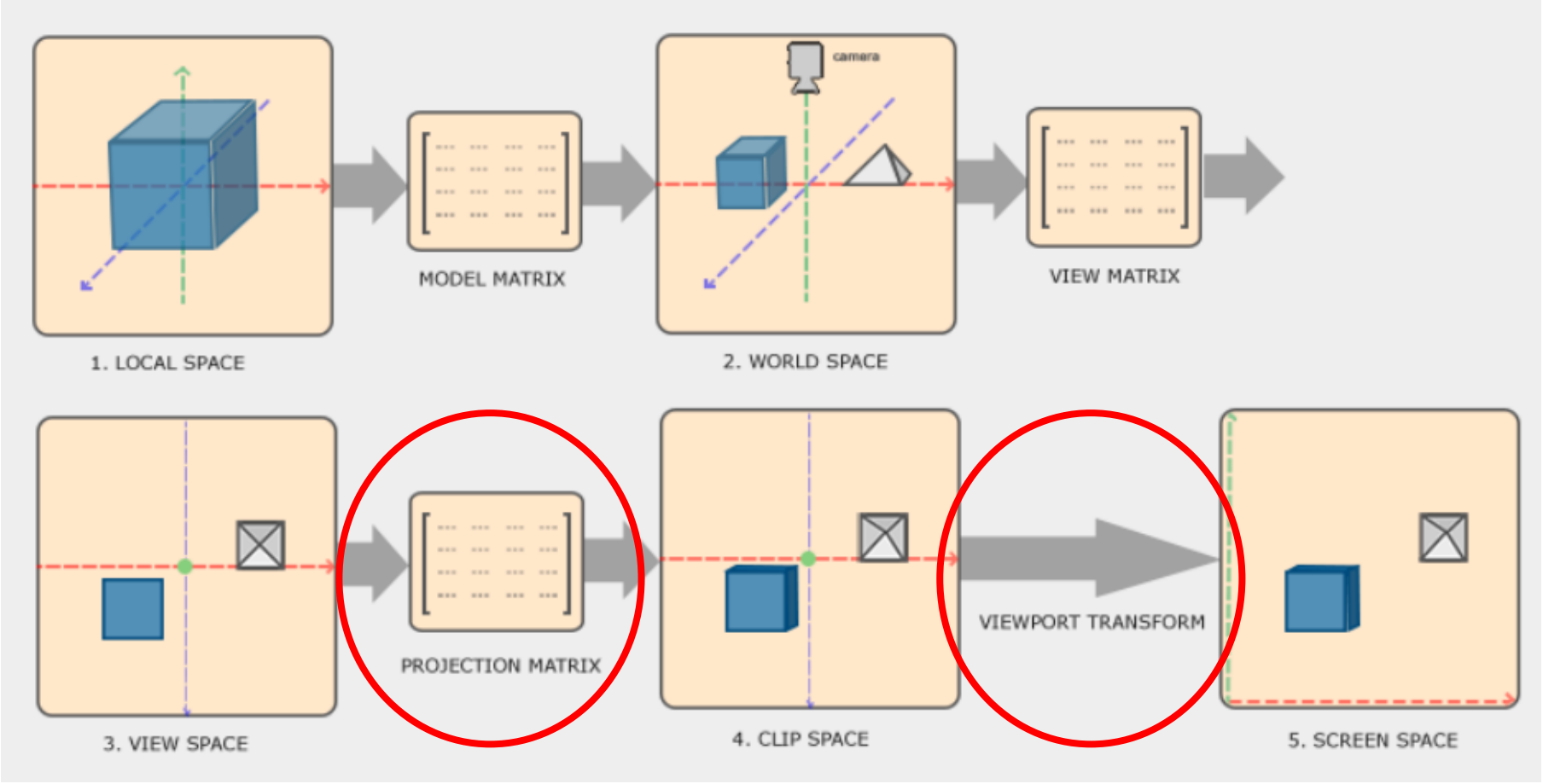
Prof. Jean R. Ponciano

Pipeline de transformações

P' = Projection x View x Model x P

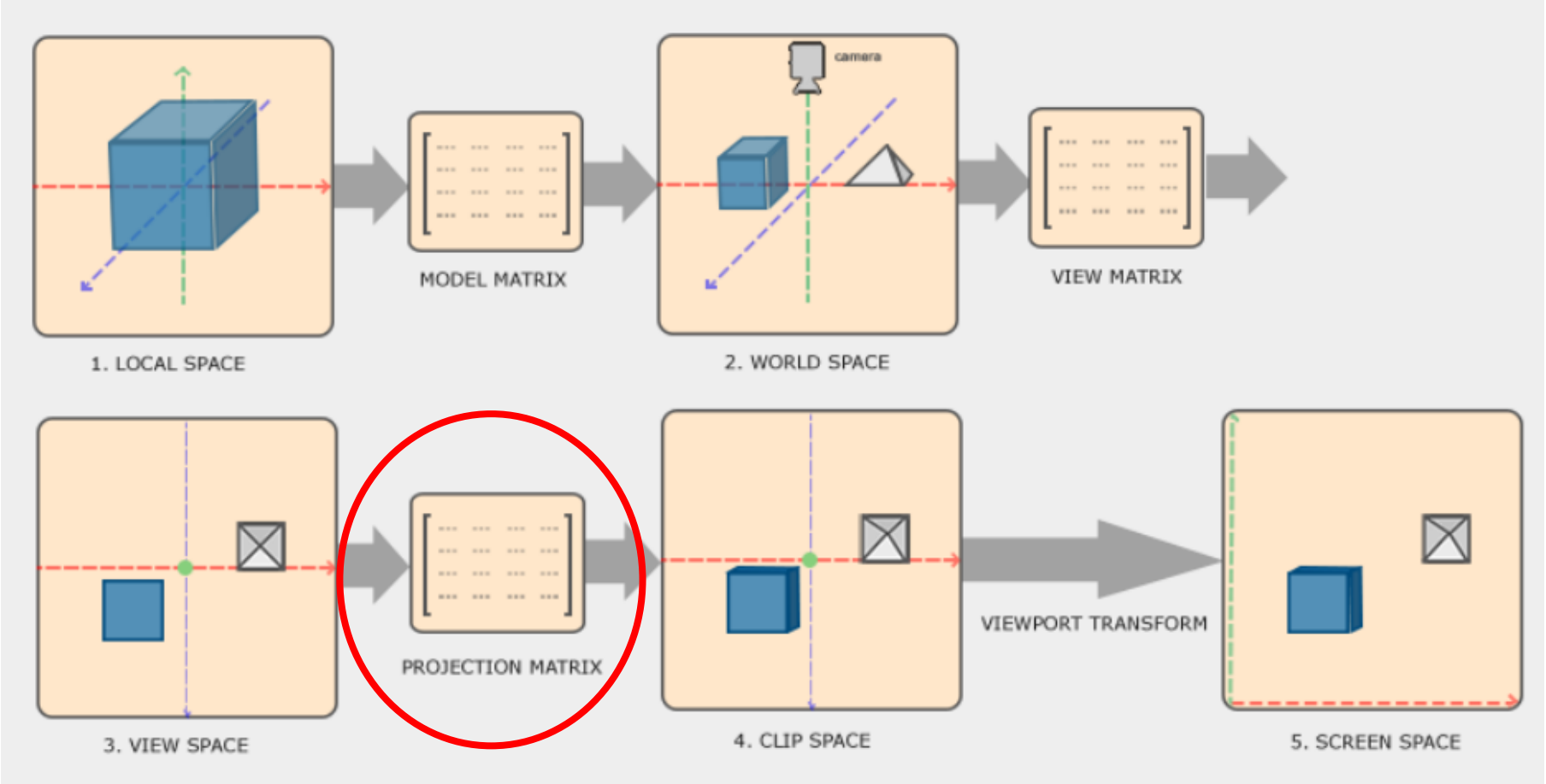


Pipeline de transformações



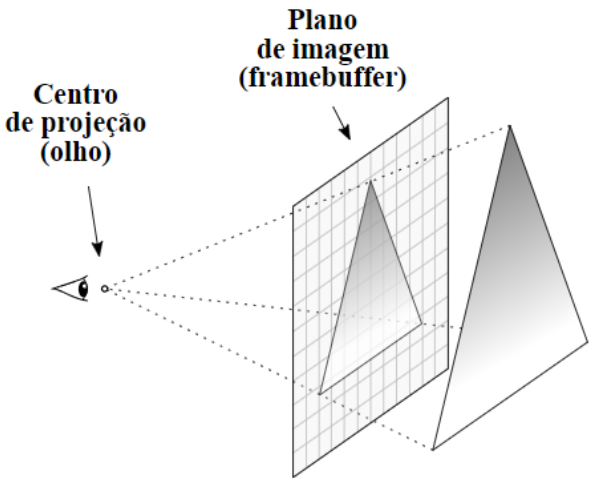
Pipeline de transformações

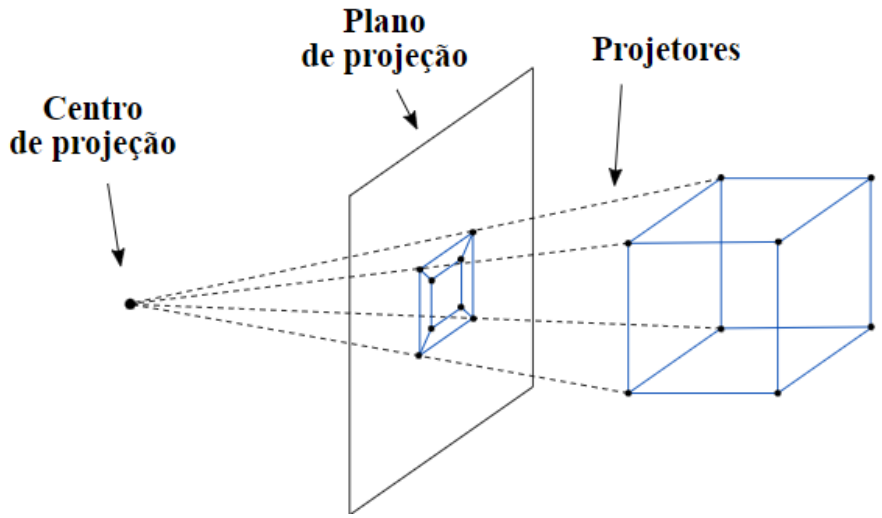
P' = Projection x View x Model x P



Projeções

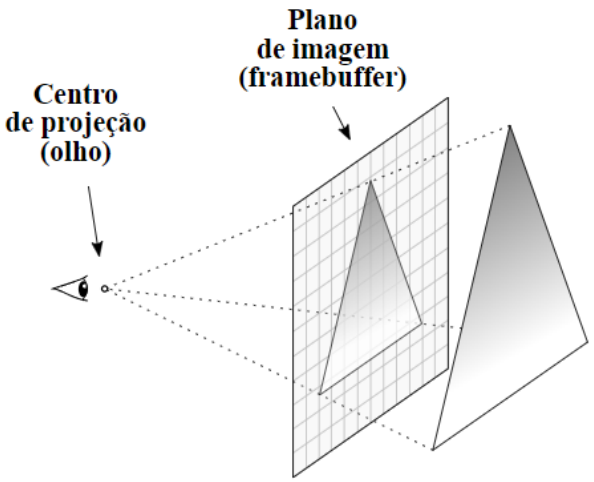
- Projeções permitem converter objetos nD em representações mD (m < n)
  - No nosso caso, objetos 3D em representações 2D
    - Focaremos em projeções planares, já que queremos projetar objetos em um plano de imagem (“fotografia tirada pela câmera virtual”)

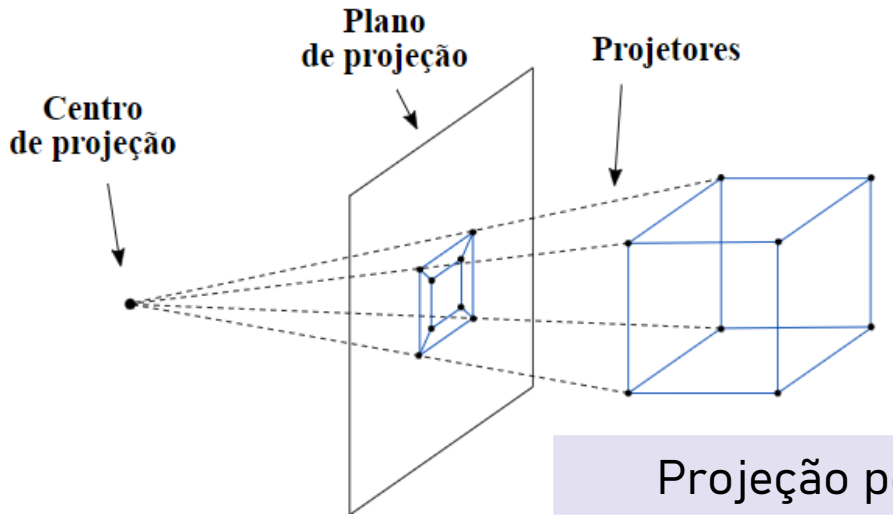




Projeções

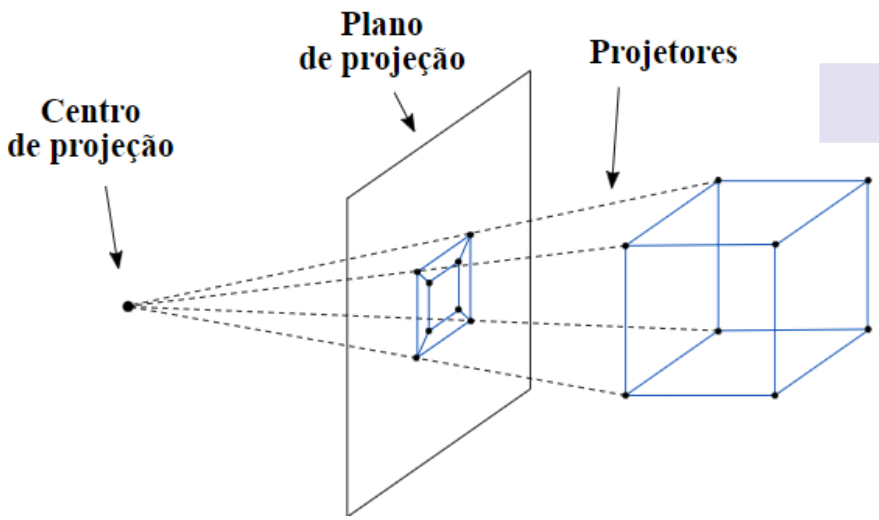
- Projeções permitem converter objetos nD em representações mD (m < n)
  - No nosso caso, objetos 3D em representações 2D
    - Focaremos em projeções planares, já que queremos projetar objetos em um plano de imagem (“fotografia tirada pela câmera virtual”)





Projeção perspectiva

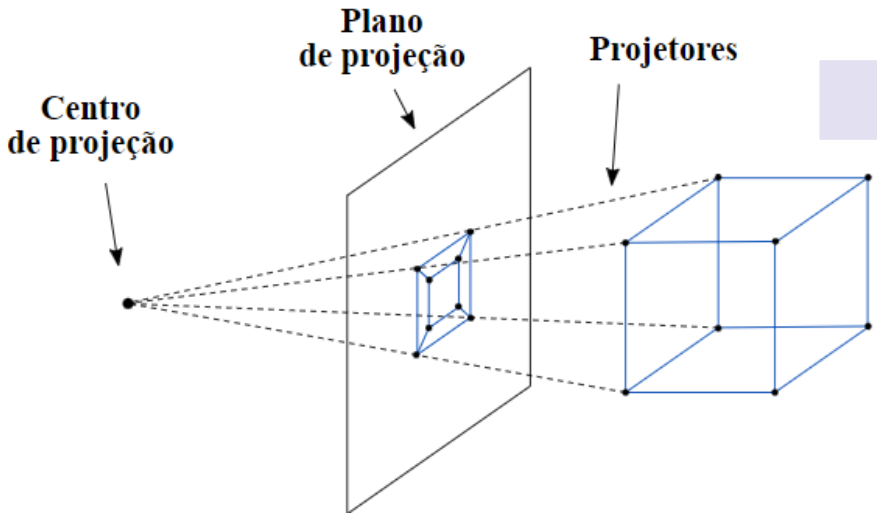
Projeções



Projeção perspectiva

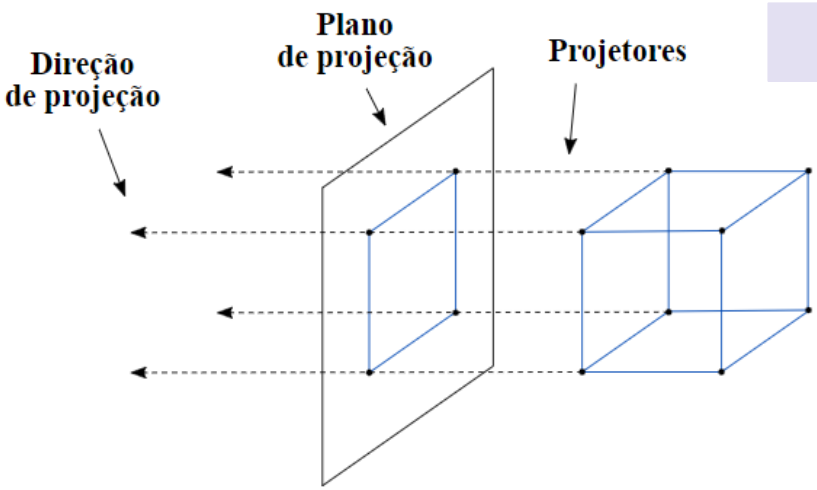
E se o centro de projeção estiver a uma distância infinita do plano de projeção?

Projeções



Projeção **perspectiva**

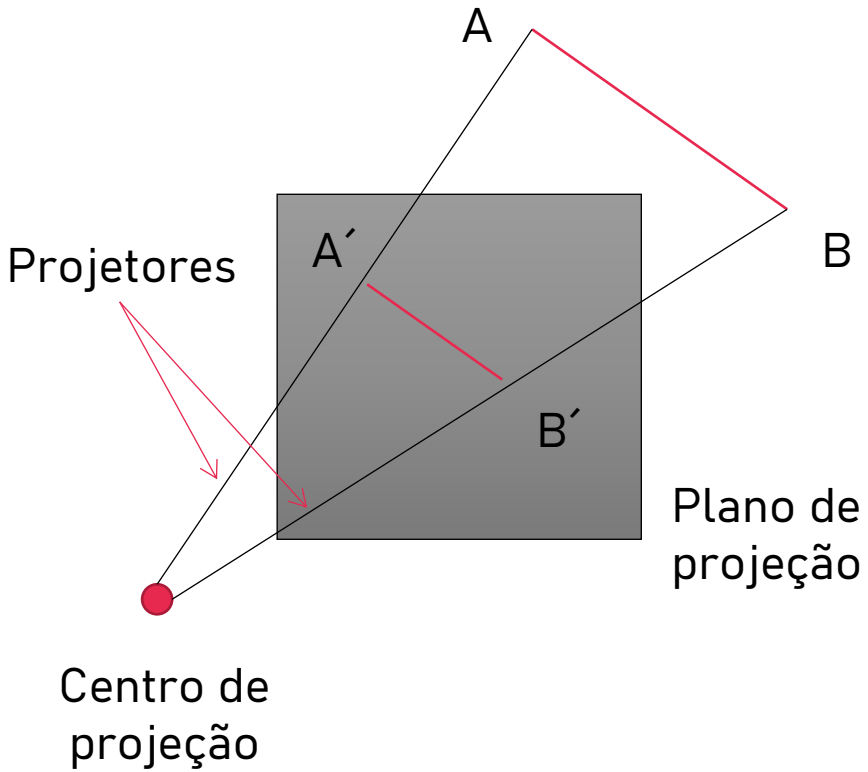
E se o centro de projeção estiver a uma distância infinita do plano de projeção?



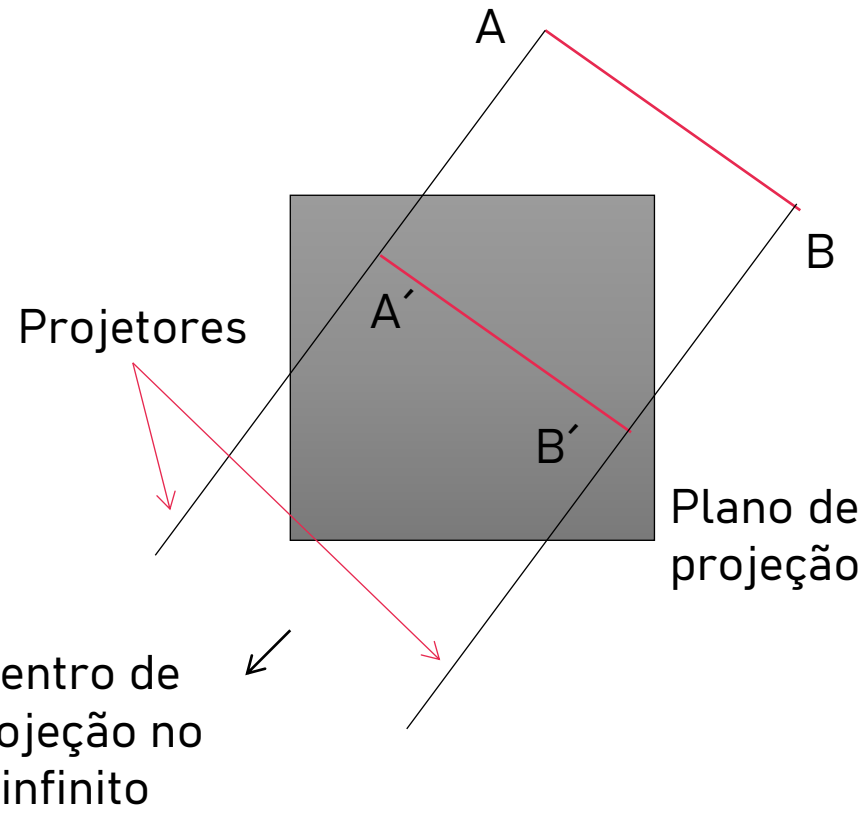
Projeção **paralela**

Projeções

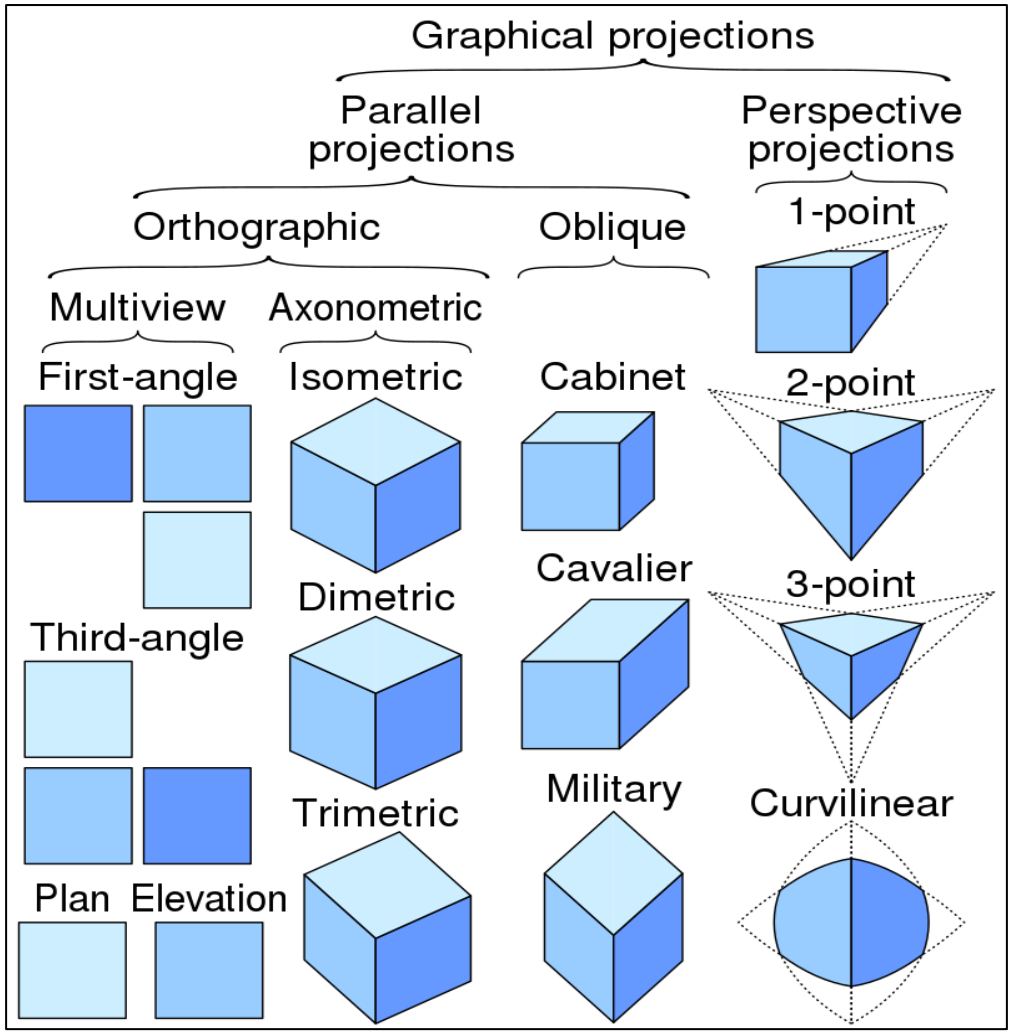
Exemplo 1:
Projeção **perspectiva** da reta AB



Exemplo 2:
Projeção **paralela** da reta AB

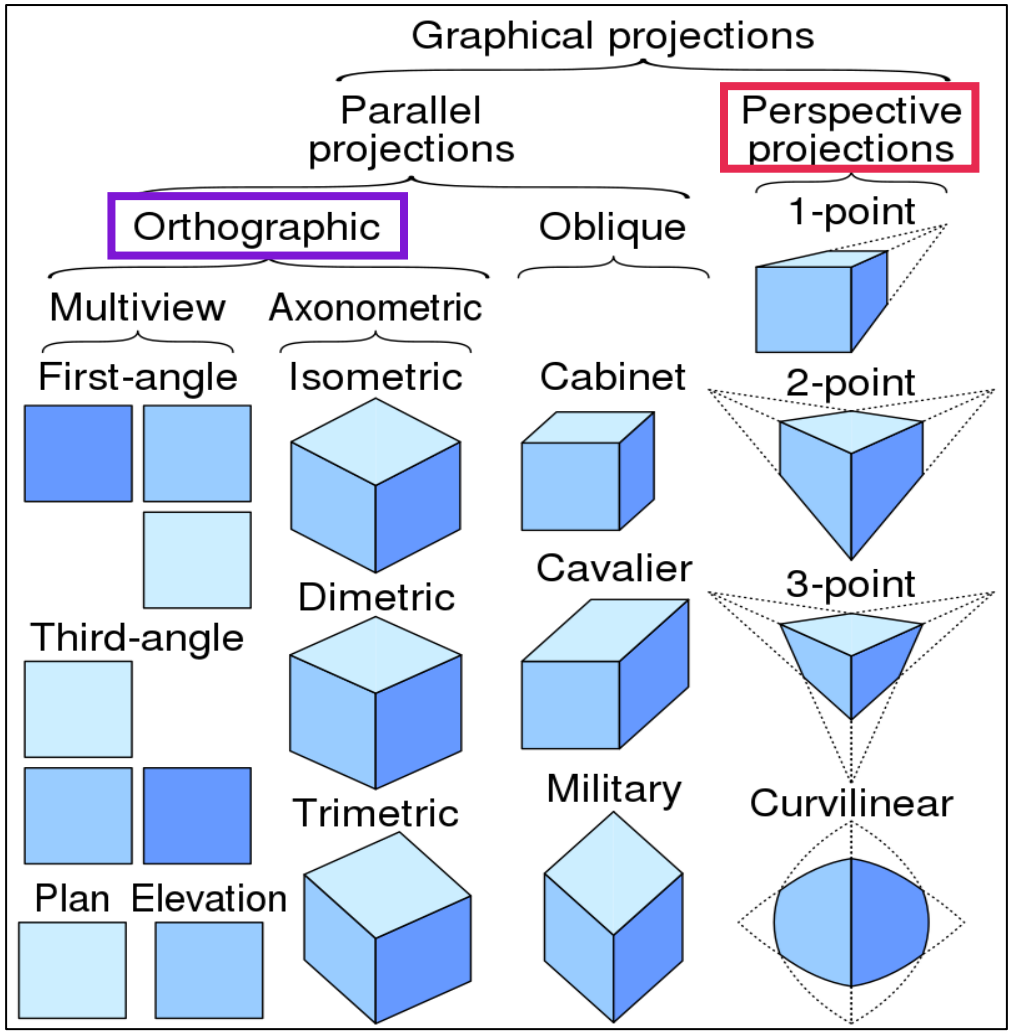


Hierarquia de Projeções



https://en.wikipedia.org/wiki/Parallel_projection

Hierarquia de Projeções



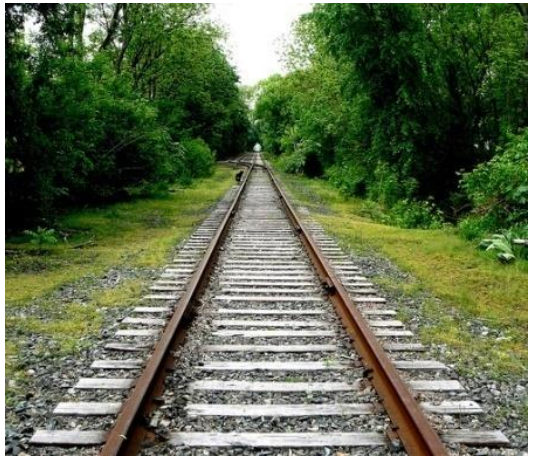
https://en.wikipedia.org/wiki/Parallel_projection

- Nosso foco:
  - Projeção perspectiva
  - Projeção paralela ortográfica
- É possível gerar qualquer outra a partir dessas duas

Projeção Perspectiva

Projeção perspectiva

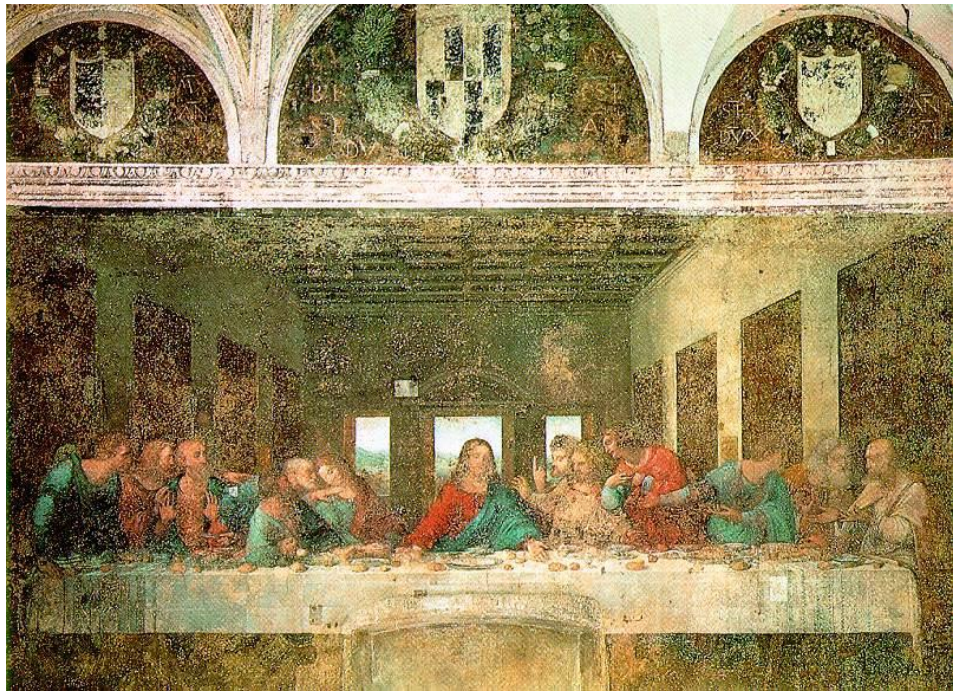
- Encurtamento perspectivo
  - Efeito de objetos distantes
  - Tamanho da projeção de um objeto varia inversamente com a distância dele ao centro de projeção
- Objetos tendem a parecer mais realistas

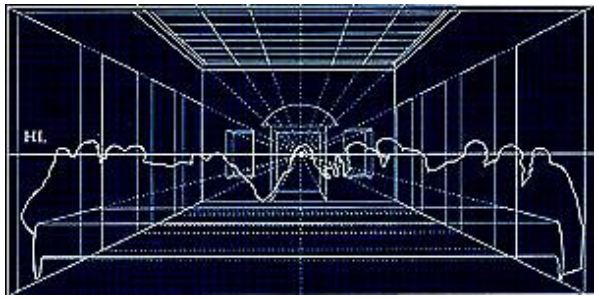


https://www.tomdalling.com/blog/modern-opengl/explaining-homogenous-coordinates-and-projective-geometry/

Projeção perspectiva

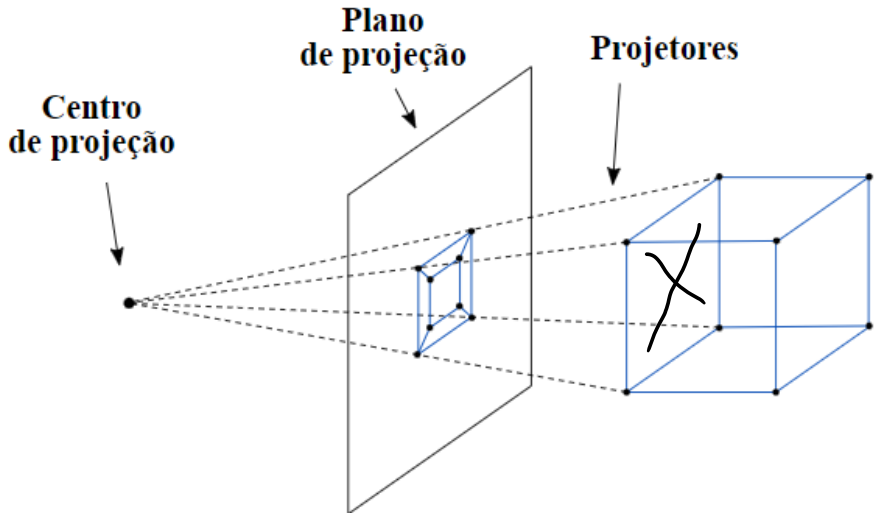
- A última ceia (Leonardo da Vinci, 1495)





Projeção perspectiva

- Não é adequada para aferição de medidas no objeto
  - Distâncias não podem ser obtidas da projeção
- Os ângulos são preservados somente nas faces do objeto que são paralelas ao plano de projeção



Projeção perspectiva

- Não é adequada para aferição de medidas no objeto
  - Distâncias não podem ser obtidas da projeção
- Os ângulos são preservados somente nas faces do objeto que são paralelas ao plano de projeção
- Linhas paralelas normalmente não são projetadas como linhas paralelas

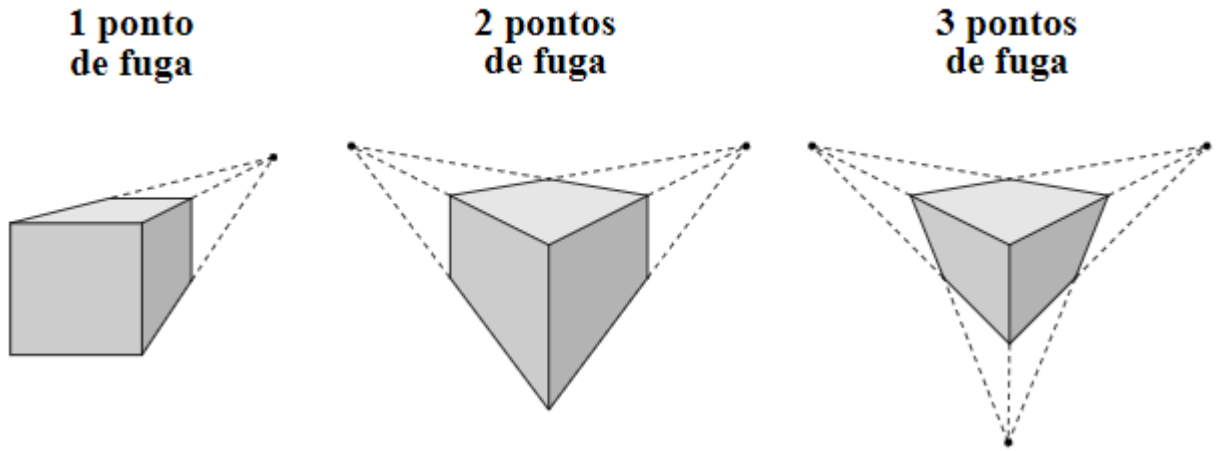
Projeção perspectiva

- Pontos de fuga
  - Locais de convergência de linhas paralelas (exceto para as paralelas ao plano de projeção, que continuam paralelas).

  - Uma cena pode ter diversos pontos de fuga, dependendo do número de linhas paralelas.

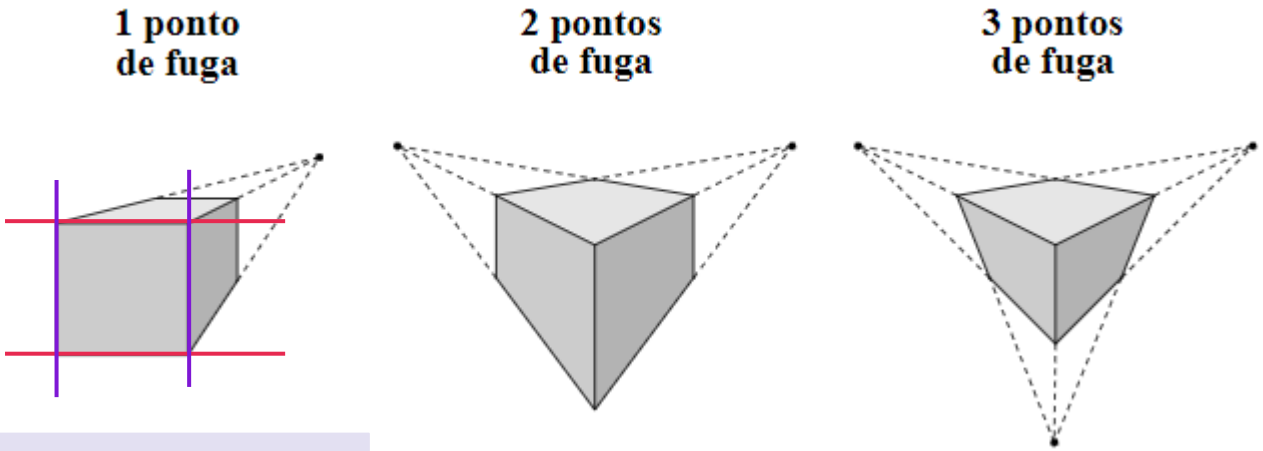
Projeção perspectiva

- Pontos de fuga



Projeção perspectiva

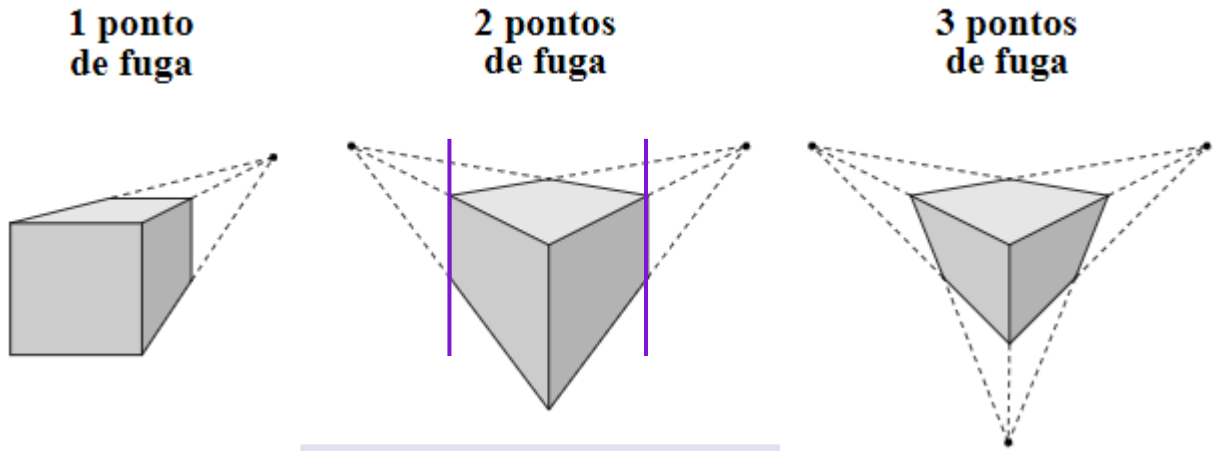
- Pontos de fuga



Arestas paralelas
aos eixos x e y

Projeção perspectiva

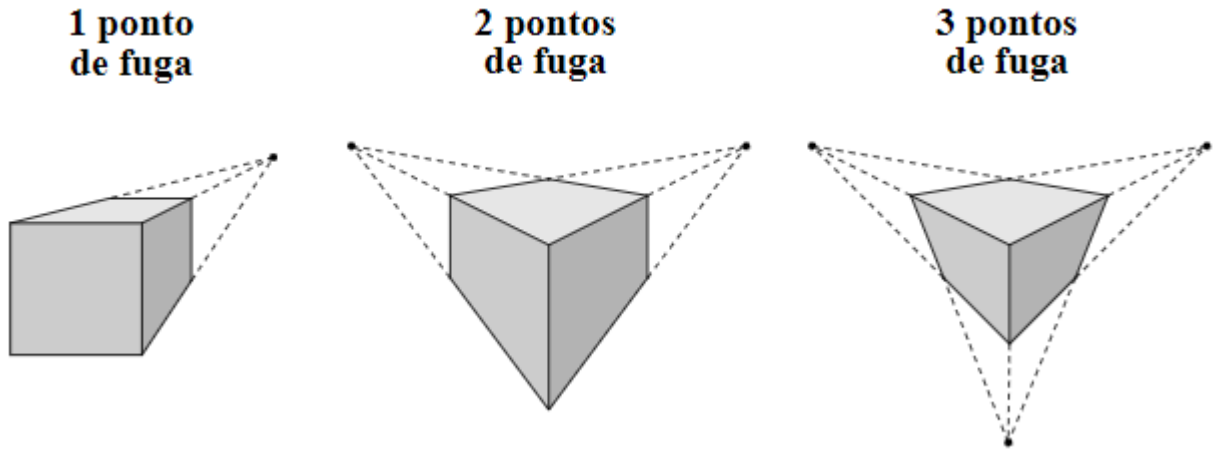
- Pontos de fuga



Arestas paralelas a um eixo (x ou y)

Projeção perspectiva

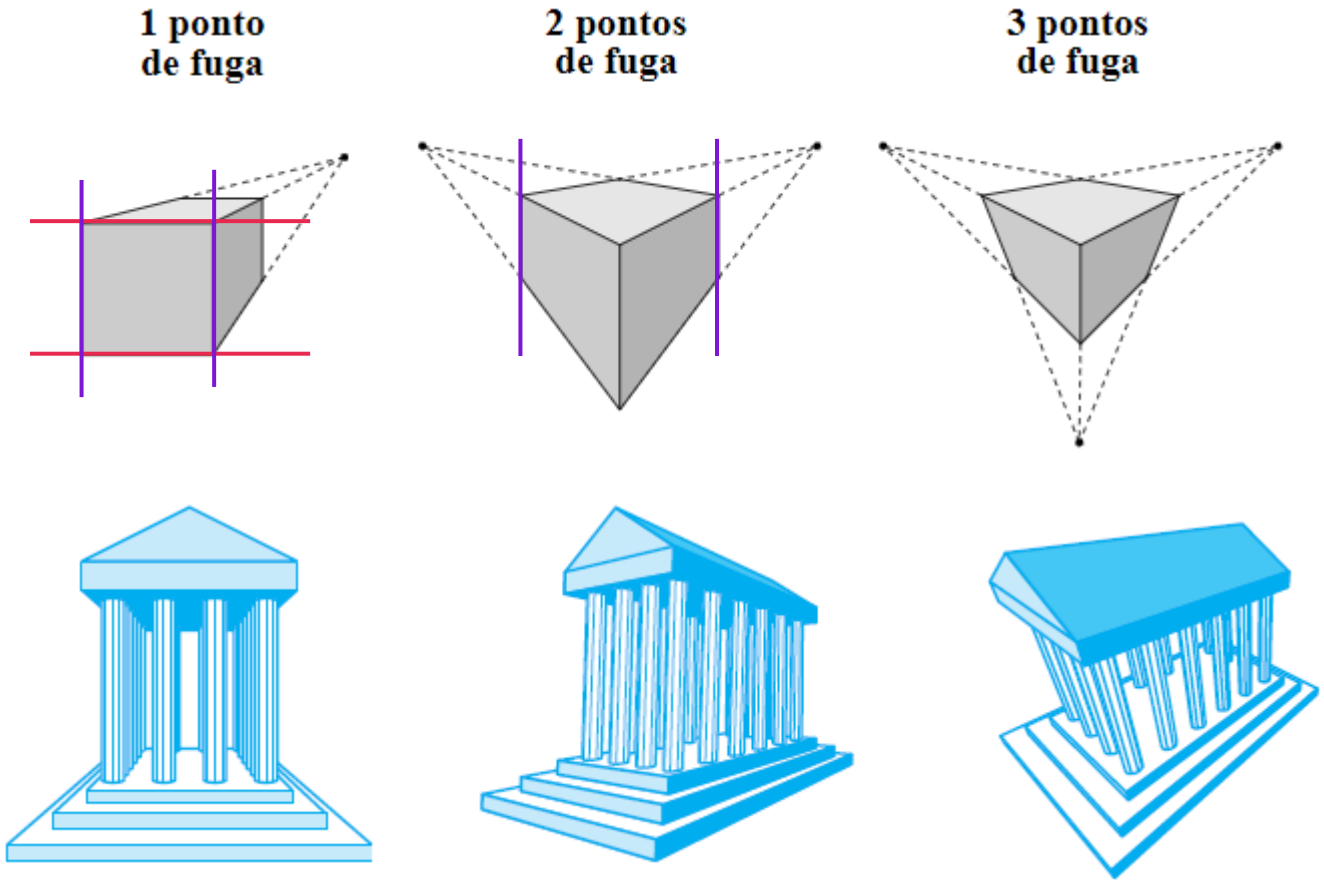
- Pontos de fuga



Sem arestas paralelas
a qualquer eixo

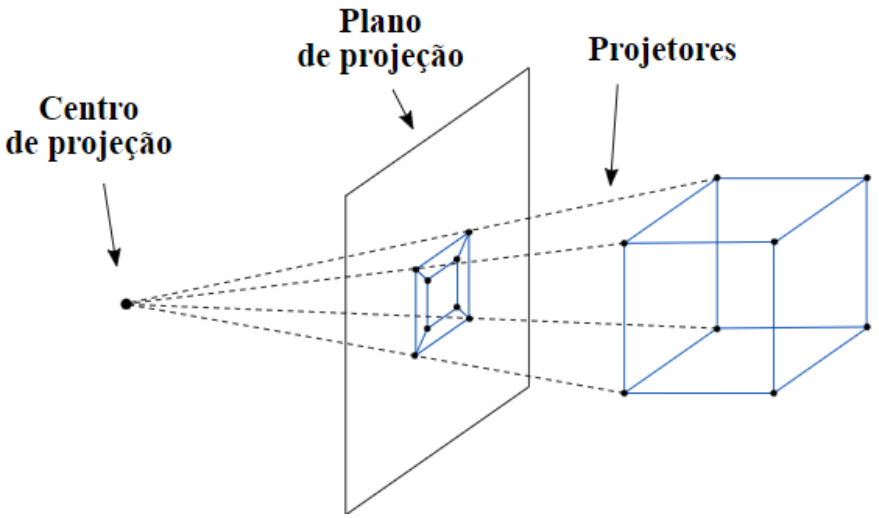
Projeção perspectiva

- Pontos de fuga

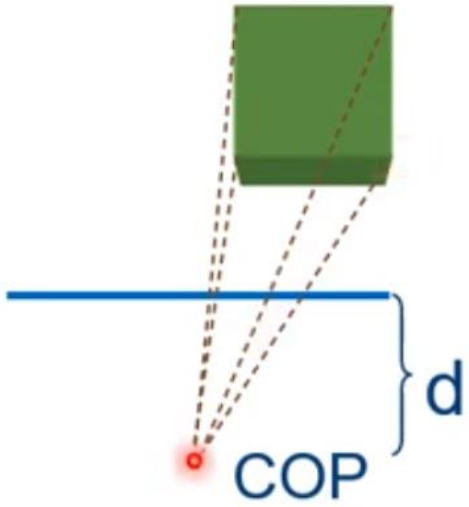


Projeção perspectiva

- Cada coordenada (x, y e z) pode ser projetada de forma independente.

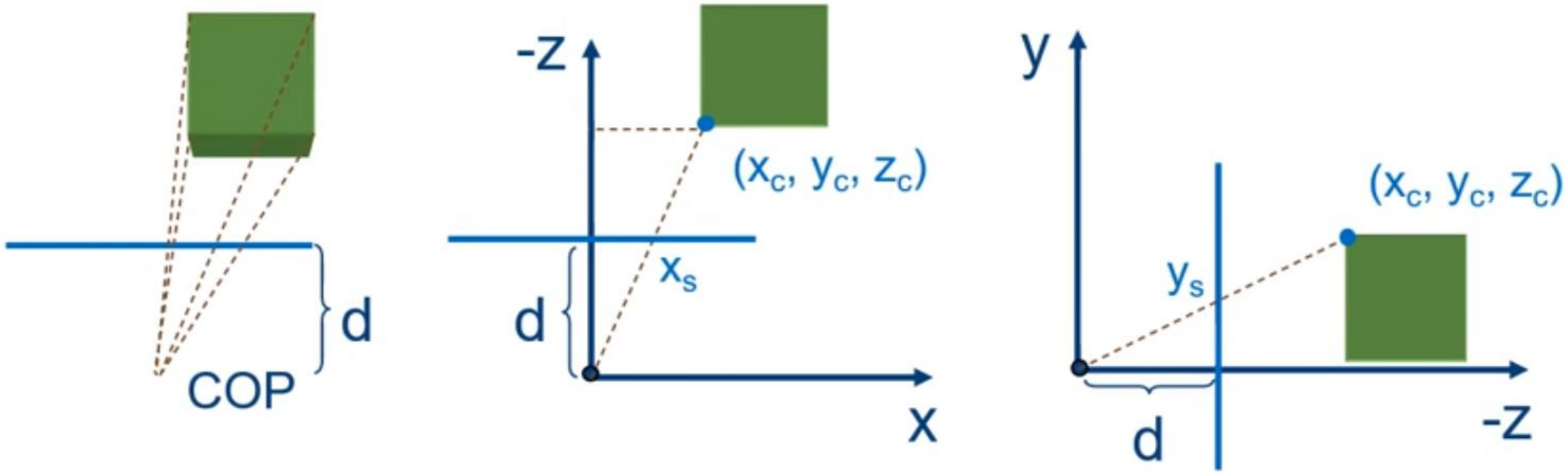


=



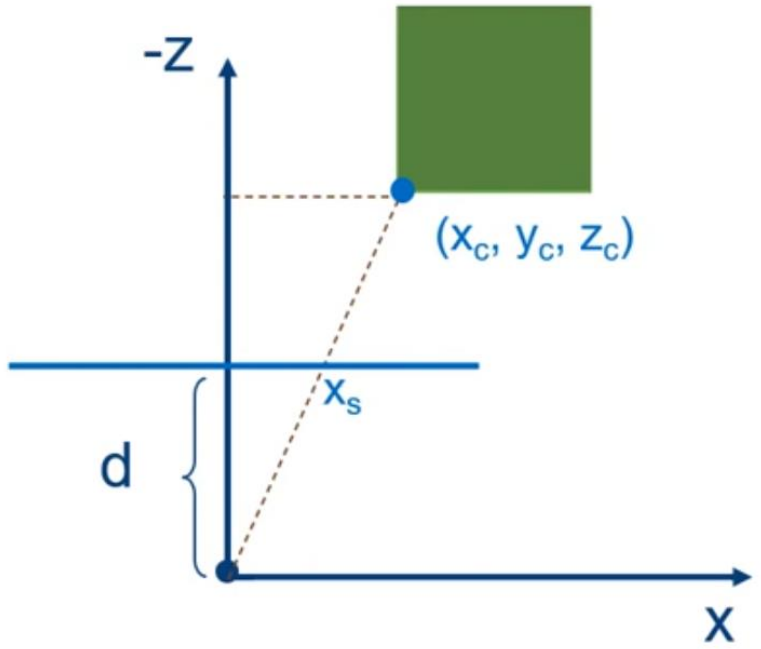
Projeção perspectiva

- Cada coordenada (x, y e z) pode ser projetada de forma independente.



Projeção perspectiva

- Cada coordenada (**x**, y e z) pode ser projetada de forma independente.

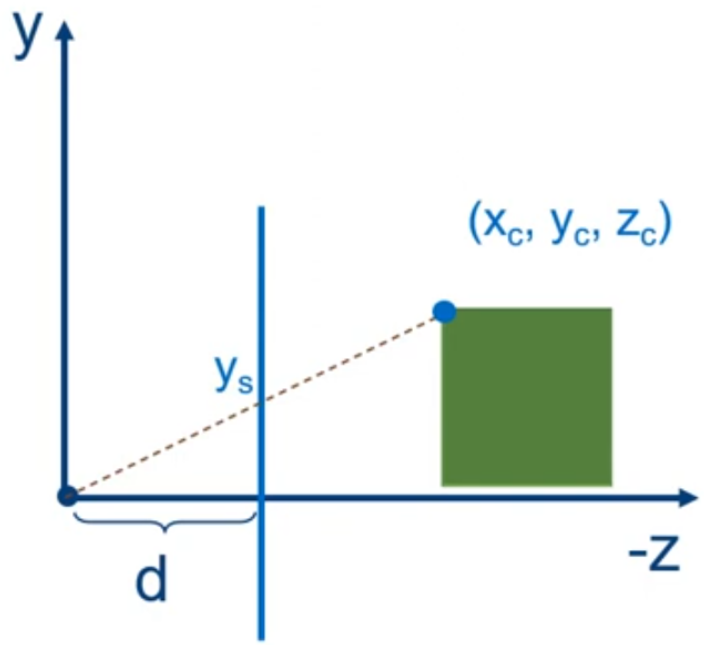


$$\frac{x_s}{d} = \frac{x_c}{-z_c}$$

$$x_s = \frac{x_c d}{-z_c} = \frac{x_c}{-z_c / d}$$

Projeção perspectiva

- Cada coordenada (x, **y** e z) pode ser projetada de forma independente.

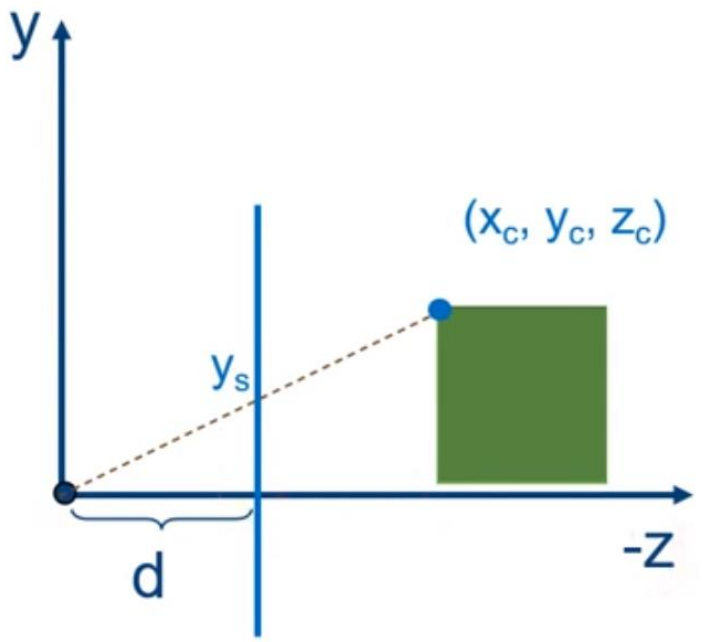


$$\frac{y_s}{d} = \frac{y_c}{-z_c}$$

$$y_s = \frac{y_c d}{-z_c} = \frac{y_c}{-z_c / d}$$

Projeção perspectiva

- Cada coordenada (x, y e **z**) pode ser projetada de forma independente.



$$\frac{z_s}{d} = \frac{z_c}{-z_c}$$

$$z_s = \frac{z_c d}{-z_c} = \frac{z_c}{-z_c / d} = -d$$

Projeção perspectiva

- Matriz de projeção (M_{persp})

$$x_s = \frac{x_c d}{-z_c} = \frac{x_c}{-z_c / d}$$

$$y_s = \frac{y_c d}{-z_c} = \frac{y_c}{-z_c / d}$$

$$z_s = \frac{z_c d}{-z_c} = \frac{z_c}{-z_c / d} = -d$$

$$w'_s = -z_c / d$$

Projeção perspectiva

- Matriz de projeção (M_{persp})

Multiplicando Mpersp pelo ponto P (em coordenadas da câmera), temos um ponto em coordenadas homogêneas no sistema de coordenadas da tela

$$\begin{bmatrix} x'_s \\ y'_s \\ z'_s \\ w'_s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

$$p_{scs} = M_{persp} p_{ccs}$$

$$x_s = \frac{x_c d}{-z_c} = \frac{x_c}{-z_c / d}$$

$$y_s = \frac{y_c d}{-z_c} = \frac{y_c}{-z_c / d}$$

$$z_s = \frac{z_c d}{-z_c} = \frac{z_c}{-z_c / d} = -d$$

$$w'_s = -z_c / d$$

Projeção perspectiva

- Matriz de projeção (M_{persp})

Multiplicando Mpersp pelo ponto P (em coordenadas da câmera), temos um ponto em coordenadas homogêneas no sistema de coordenadas da tela

$$\begin{bmatrix} x'_s \\ y'_s \\ z'_s \\ w'_s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & & \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ \\ 1 \end{bmatrix}$$

Regra da mão direita

$$p_{scs} = M_{persp} p_{ccs}$$

$$x_s = \frac{x_c d}{-z_c} = \frac{x_c}{-z_c / d}$$

$$y_s = \frac{y_c d}{-z_c} = \frac{y_c}{-z_c / d}$$

$$z_s = \frac{z_c d}{-z_c} = \frac{z_c}{-z_c / d} = -d$$

$$w'_s = -z_c / d$$

Projeção perspectiva

- Matriz de projeção (M_{persp})

Multiplicando Mpersp pelo ponto P (em coordenadas da câmera), temos um ponto em coordenadas homogêneas no sistema de coordenadas da tela

$$\begin{bmatrix} x'_s \\ y'_s \\ z'_s \\ w'_s \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/d & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

$$p_{scs} = M_{persp} p_{ccs}$$

$$x_s = \frac{x_c d}{-z_c} = \frac{x_c}{-z_c / d}$$

$$y_s = \frac{y_c d}{-z_c} = \frac{y_c}{-z_c / d}$$

$$z_s = \frac{z_c d}{-z_c} = \frac{z_c}{-z_c / d} = -d$$

$$w'_s = -z_c / d$$

Notem que w'$_s$!= 1. Para obter as coordenadas cartesianas (x_s, y_s, z_s), precisamos dividir por w's

$$(x_s, y_s, z_s) = \left(\frac{x'_s}{w'_s}, \frac{y'_s}{w'_s}, \frac{z'_s}{w'_s}\right) = \left(\frac{x_c}{-zc/d}, \frac{y_c}{-zc/d}, -d\right)$$

Projeção perspectiva

- Matriz de projeção (M_{persp})

$$x_s = \frac{x_c d}{-z_c} = \frac{x_c}{-z_c / d}$$

$$y_s = \frac{y_c d}{-z_c} = \frac{y_c}{-z_c / d}$$

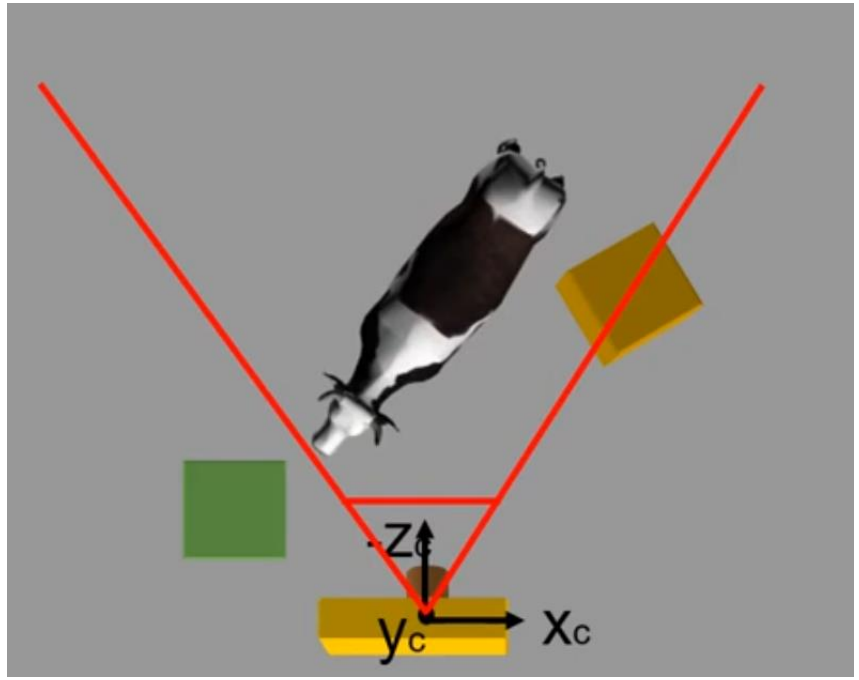
$$z_s = \frac{z_c d}{-z_c} = \frac{z_c}{-z_c / d} = -d$$

$$w'_s = -z_c / d$$

w's é o valor a ser usado para definir qual objeto está na frente de qual na cena (z-buffering ou depth-buffering)

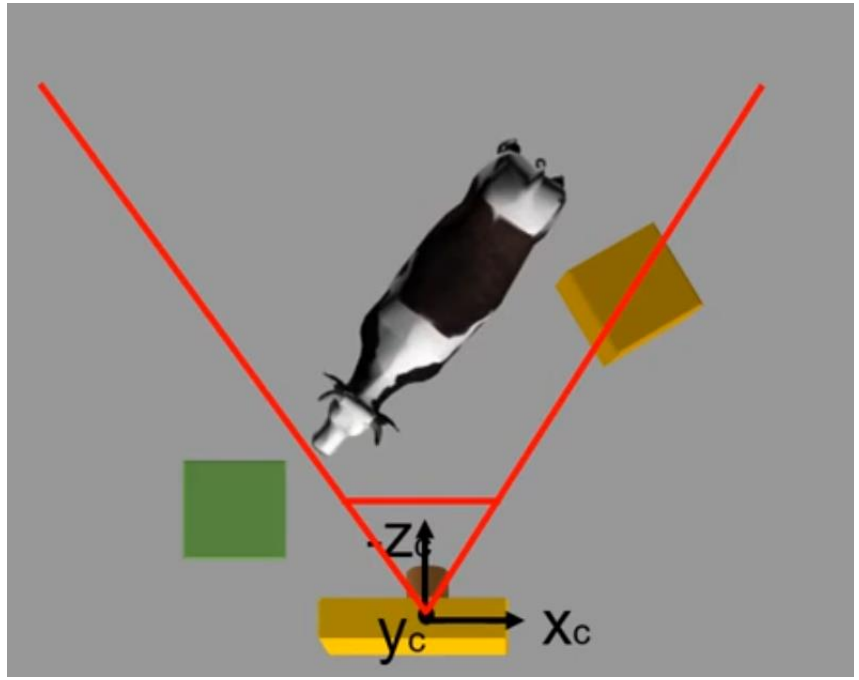
Projeção perspectiva

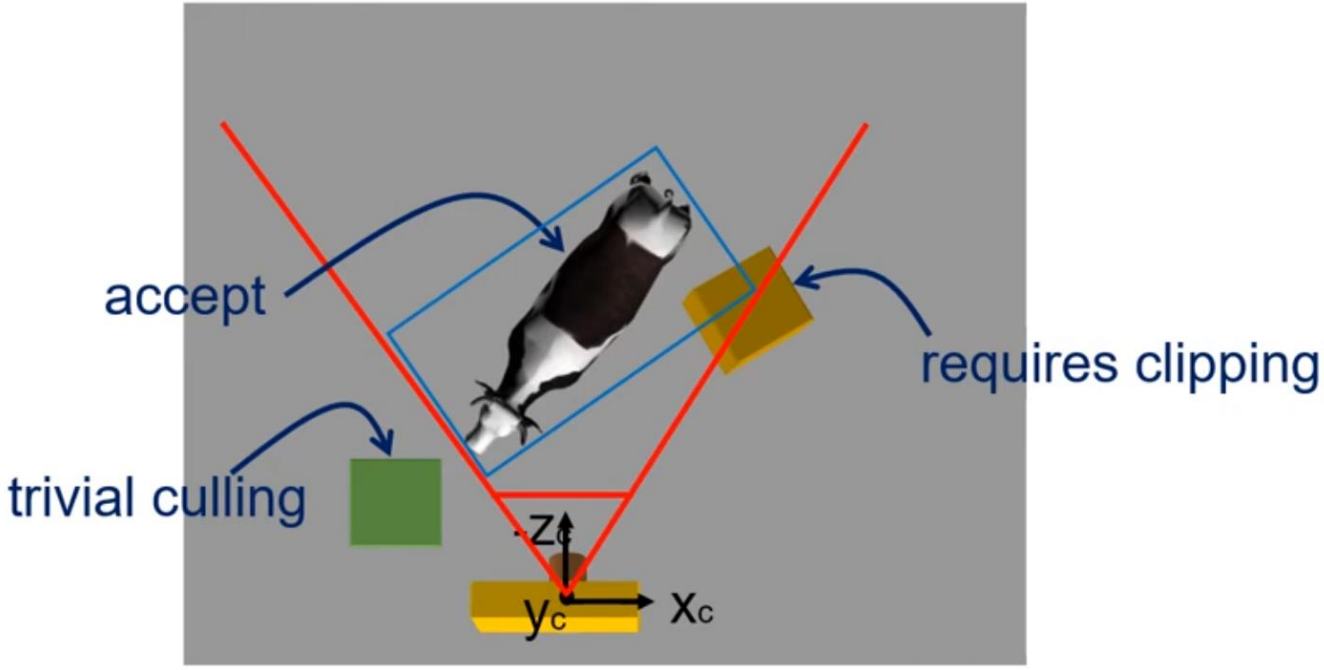
- Podemos aproveitar a matriz projection para fazer mais uma coisa...



Projeção perspectiva

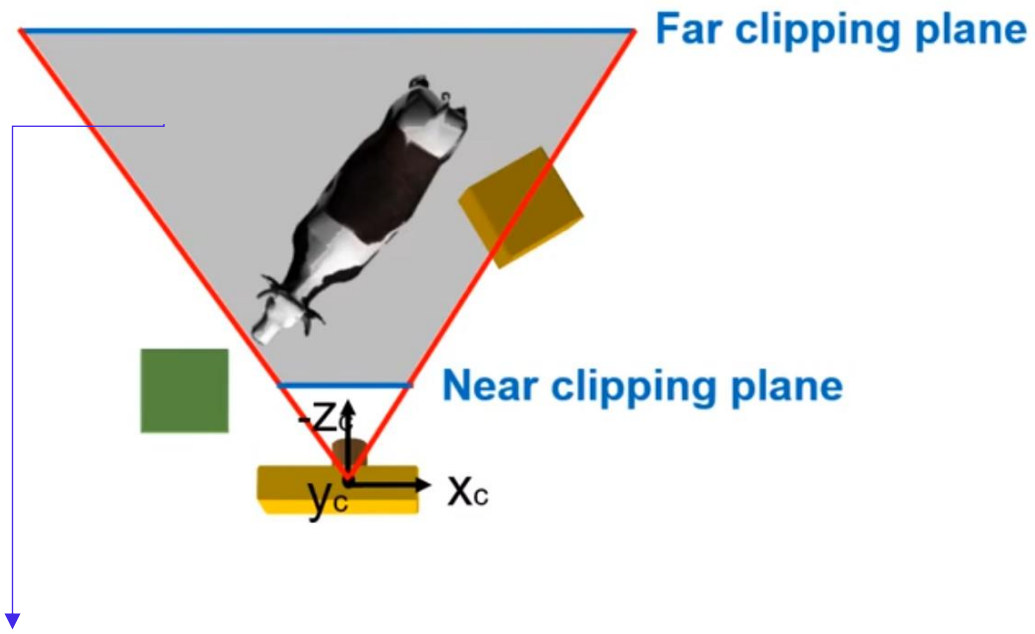
- Podemos aproveitar a matriz projection para fazer mais uma coisa...





Projeção perspectiva

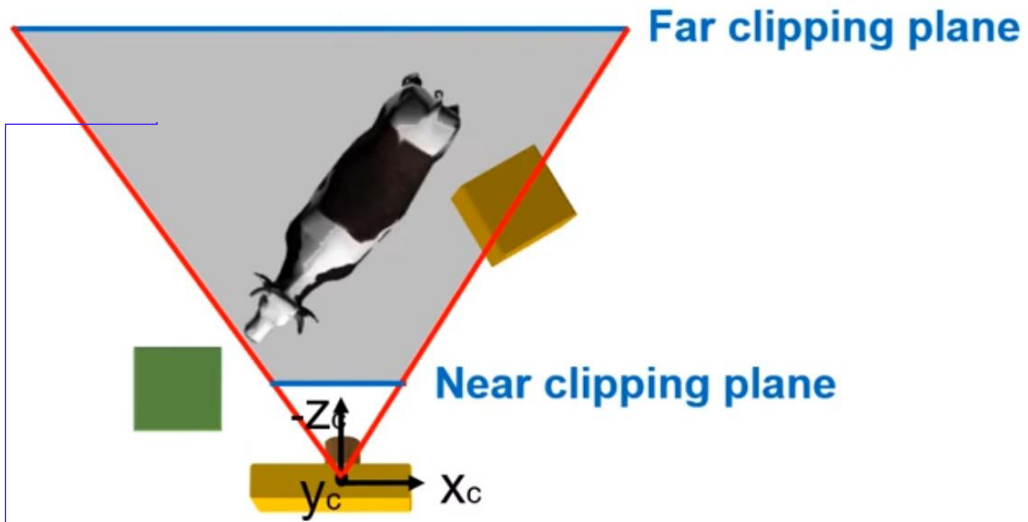
- Podemos aproveitar a matriz projection para fazer mais uma coisa...

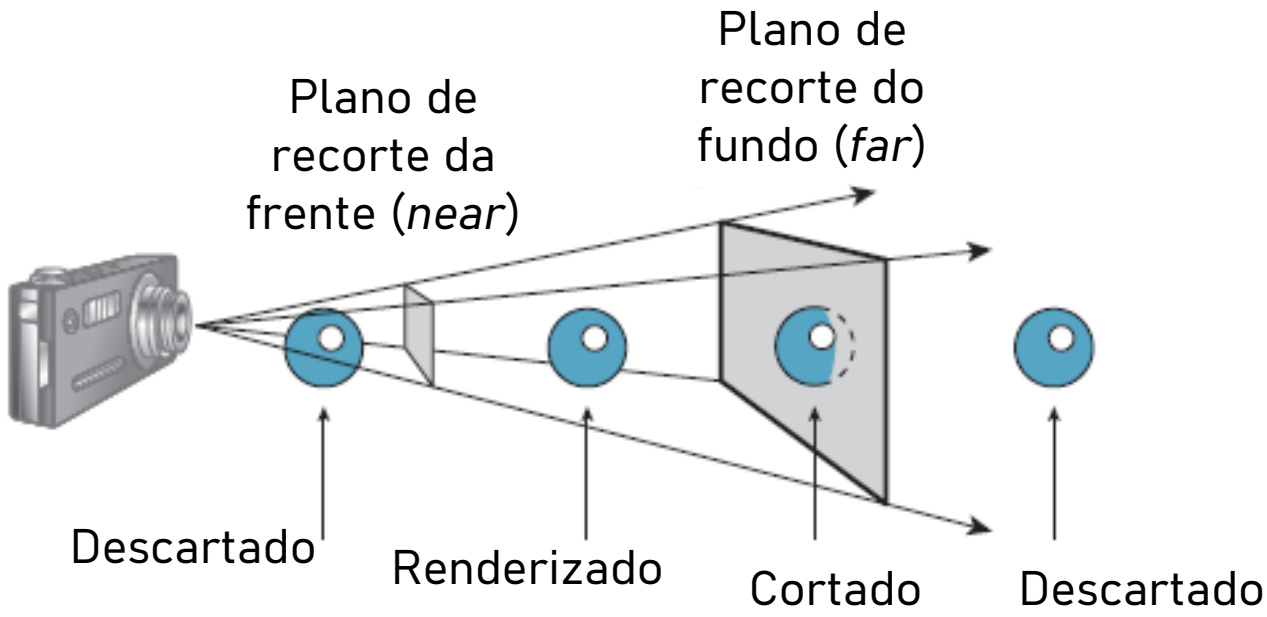


Frustum (volume de visão)

Projeção perspectiva

- Podemos aproveitar a matriz projection para fazer mais uma coisa...

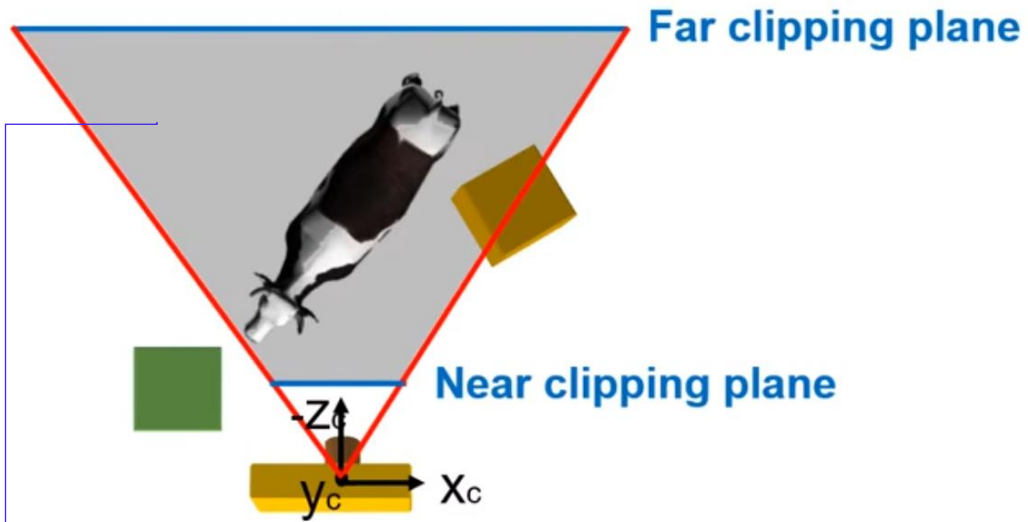


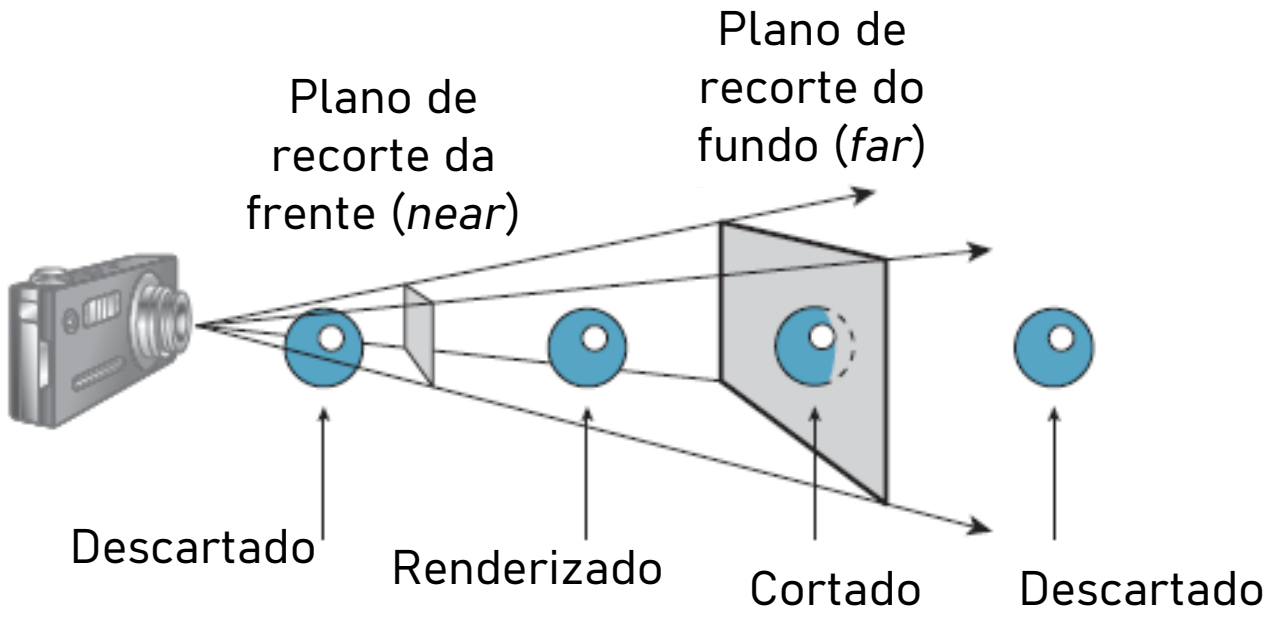


Frustum (volume de visão)

Projeção perspectiva

- Podemos aproveitar a matriz projection para fazer mais uma coisa...

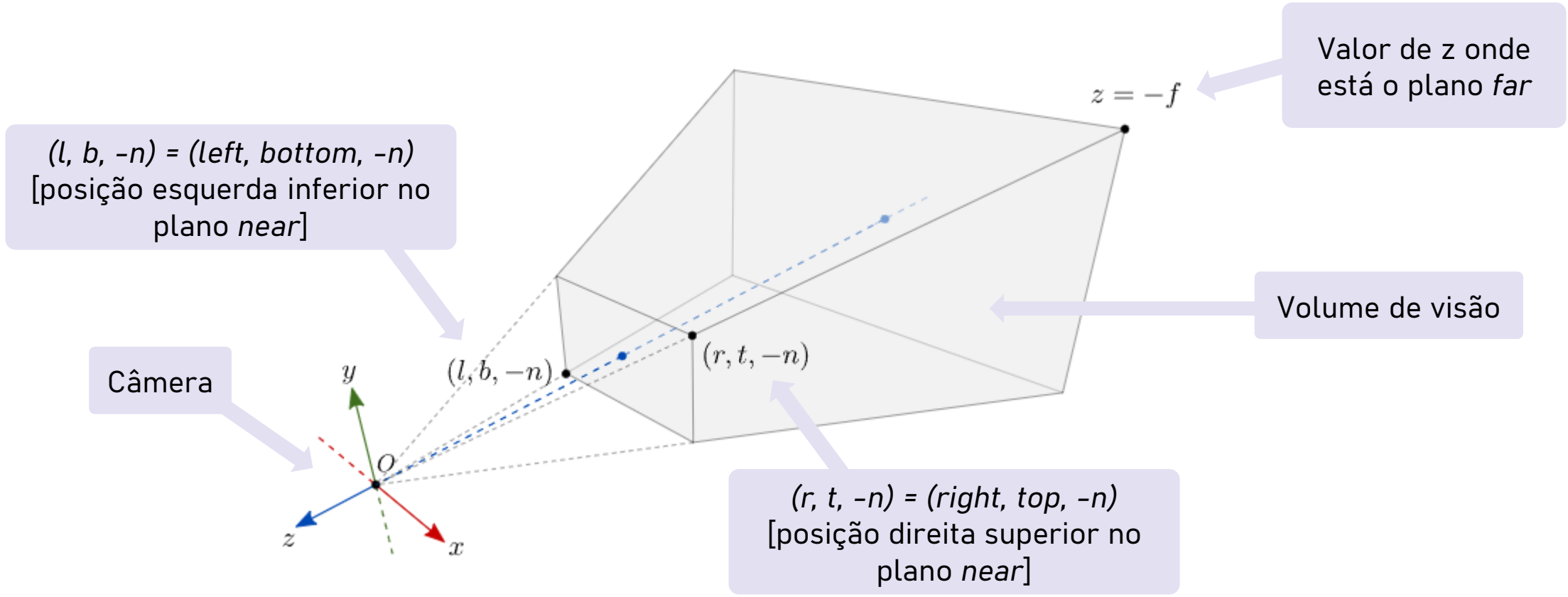




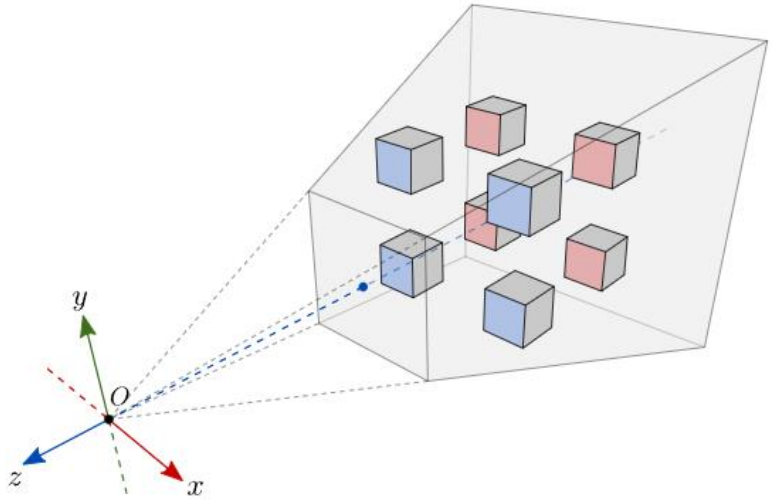
Frustum (volume de visão)

https://webglfundamentals.org/webgl/frustum-diagram.html

Volume de visão (espaço da câmera)

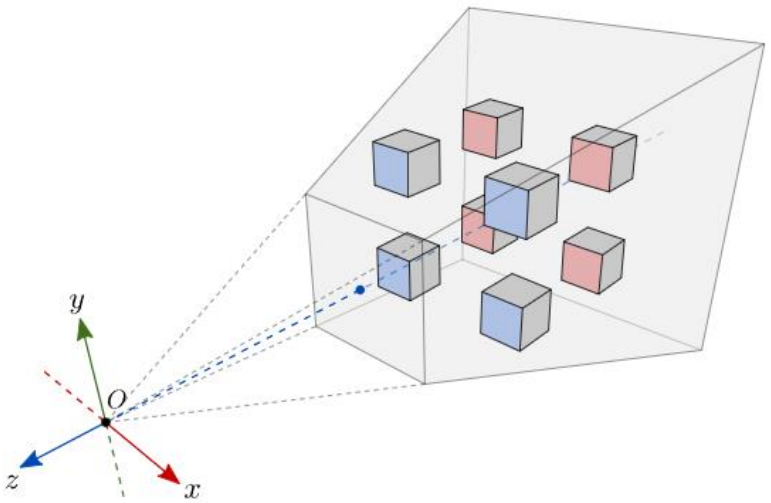


Projeção perspectiva

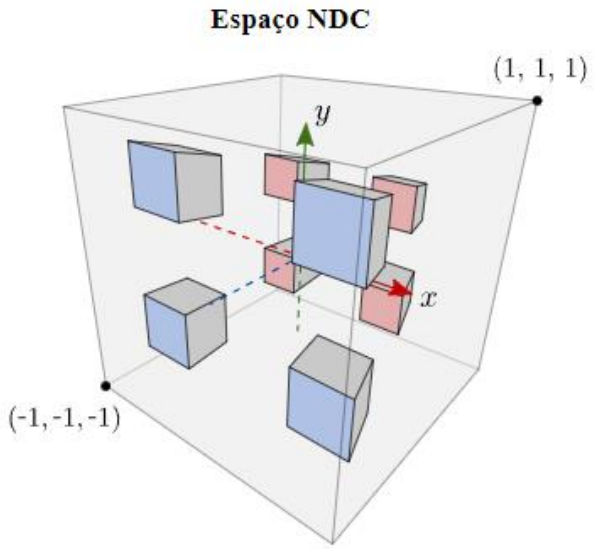


Volume de visão (espaço da câmera)

Projeção perspectiva



Volume de visão (espaço da câmera)



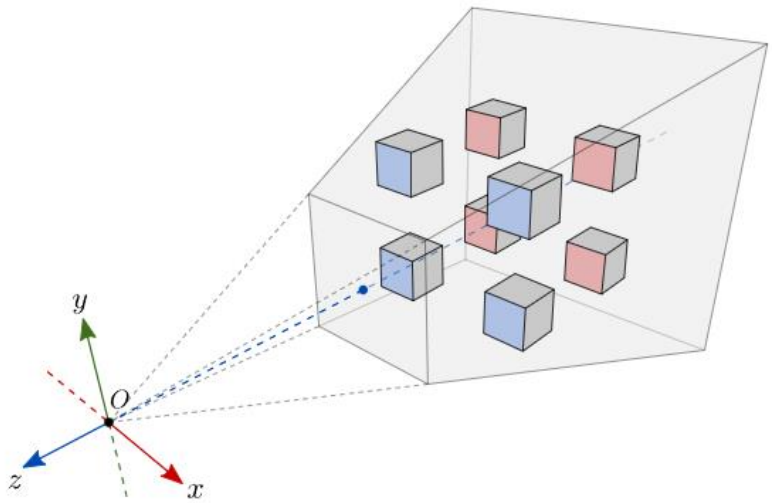
Volume de visão normalizado (NDC = espaço normalizado do dispositivo)

Note:

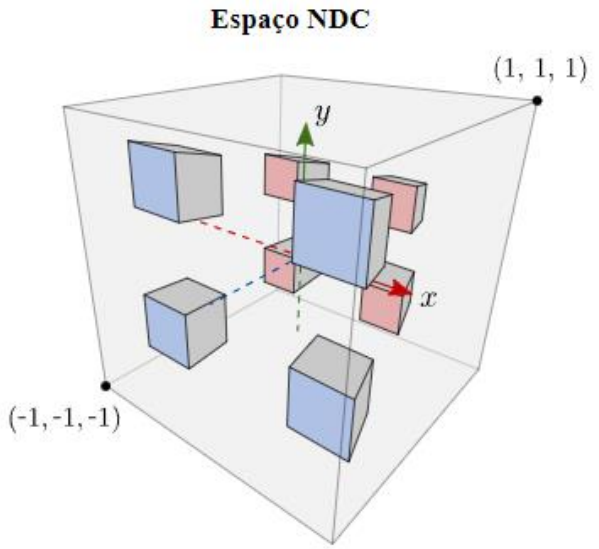
1) Geometria da cena distorcida (objetos distantes são menores)

2) Arestas laterais não são mais paralelas

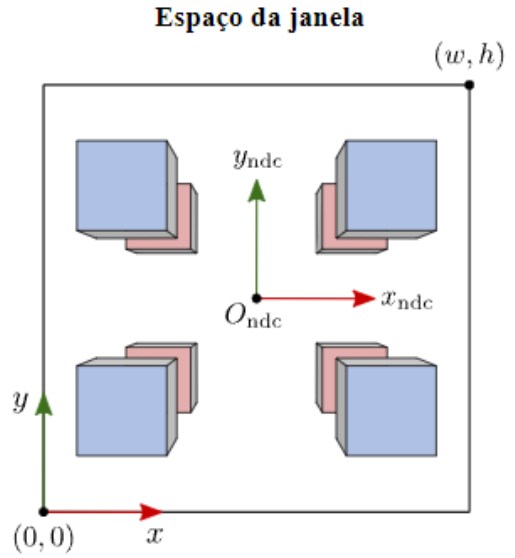
Projeção perspectiva



Volume de visão (espaço da câmera)

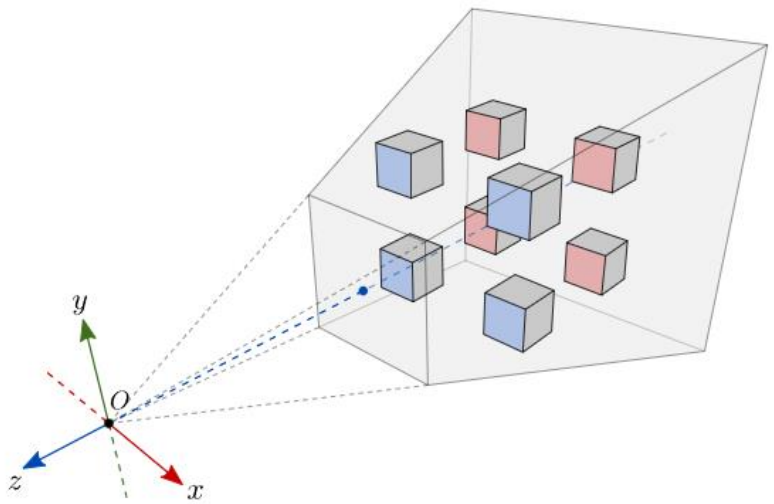


Volume de visão normalizado (NDC = espaço normalizado do dispositivo)

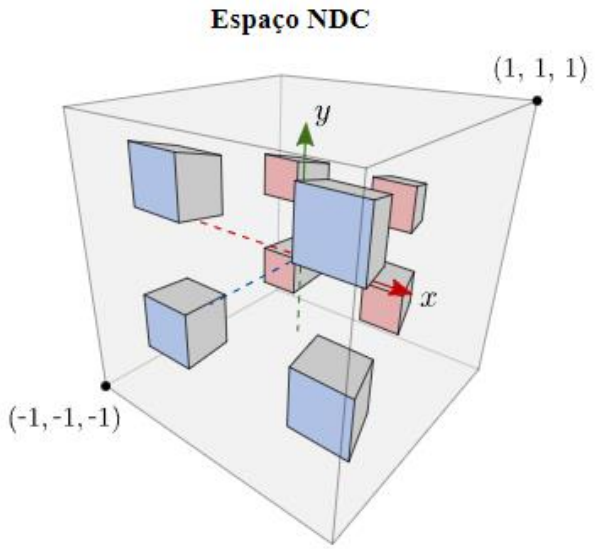


Resultado de uma projeção paralela ortográfica (veremos daqui a pouco)

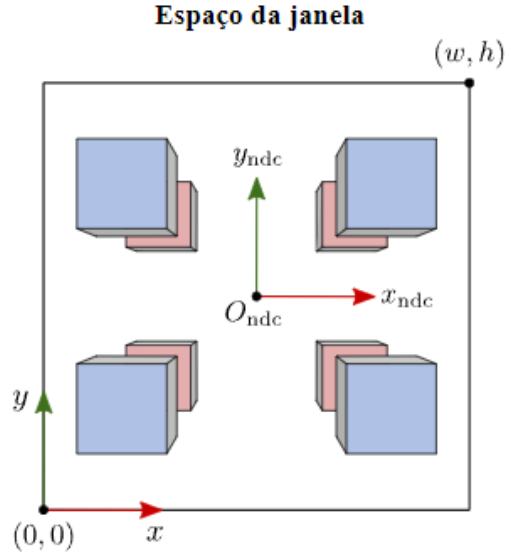
Projeção perspectiva



Volume de visão (espaço da câmera)

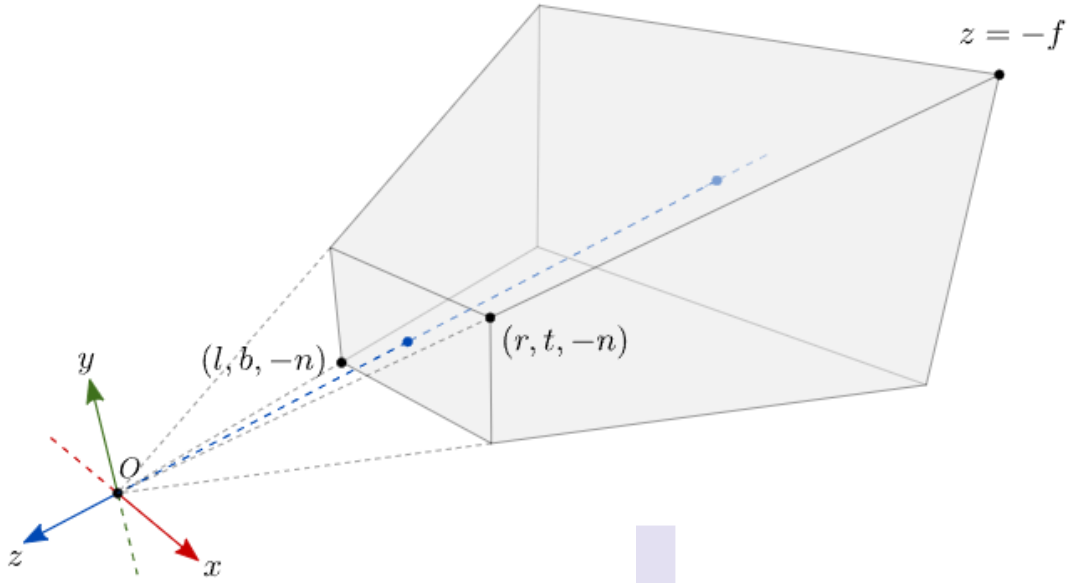


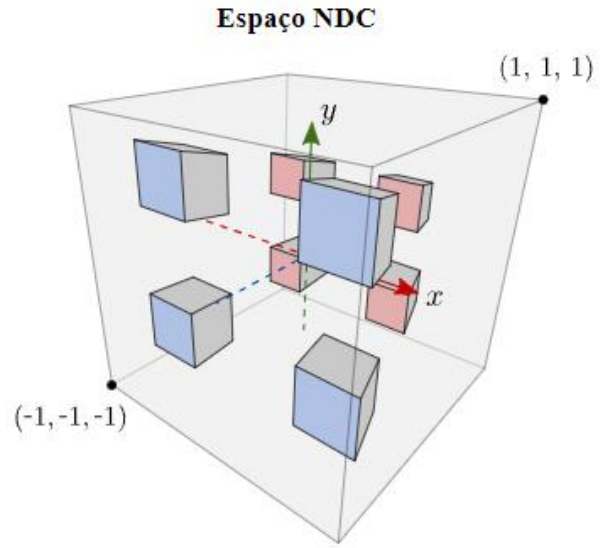
Volume de visão normalizado (NDC = espaço normalizado do dispositivo)



Resultado de uma projeção paralela ortográfica (veremos daqui a pouco)

Projeção perspectiva



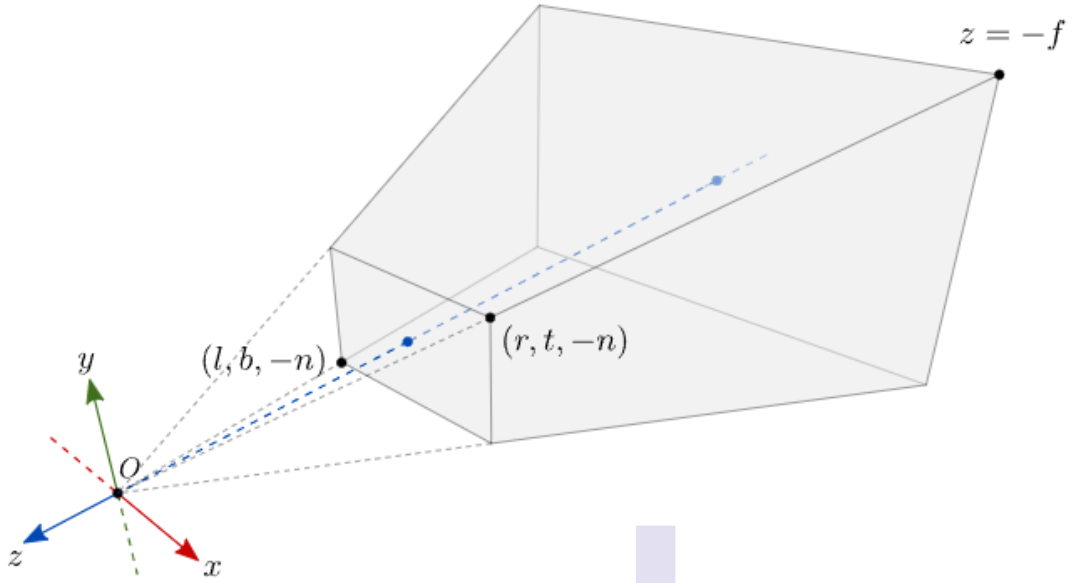


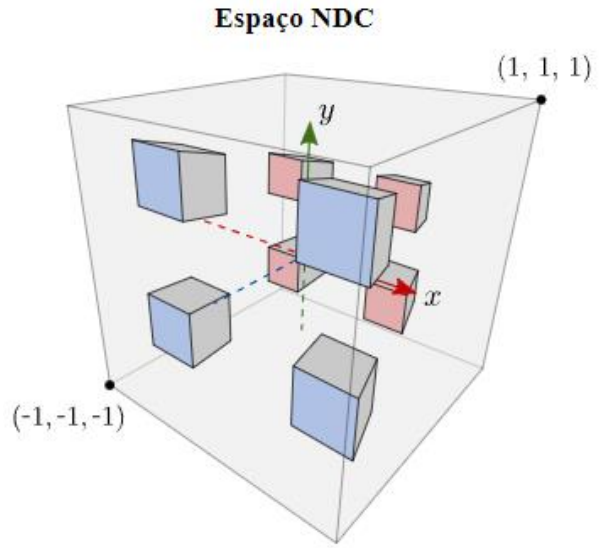
Projeção perspectiva

Em x: [l,r], no espaço da câmera, para [−1,1] em NDC;

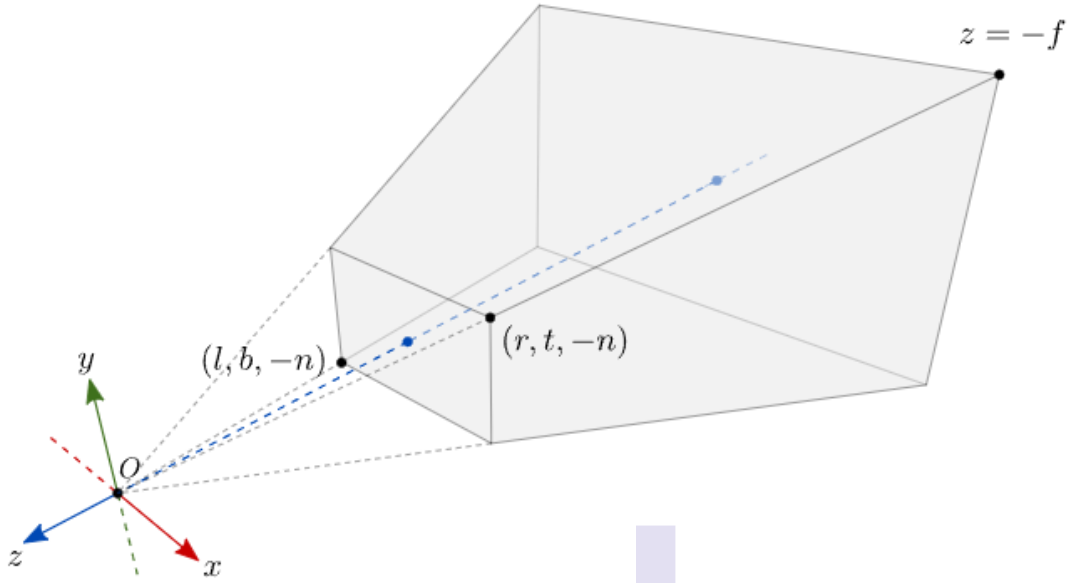
Em y: [b,t], no espaço da câmera, para [−1,1] em NDC;

Em z: [−n,−f], no espaço da câmera, para [−1,1] em NDC.





Projeção perspectiva

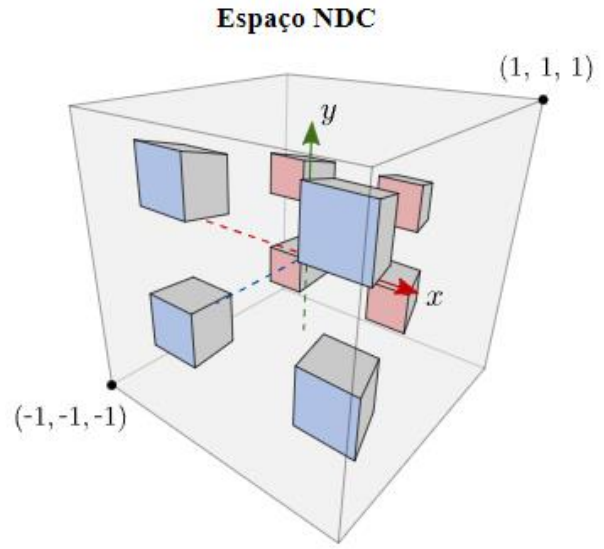


Em x: [l,r], no espaço da câmera, para [−1,1] em NDC;

Em y: [b,t], no espaço da câmera, para [−1,1] em NDC;

Em z: [−n,−f], no espaço da câmera, para [−1,1] em NDC.

Normalização do espaço de visão

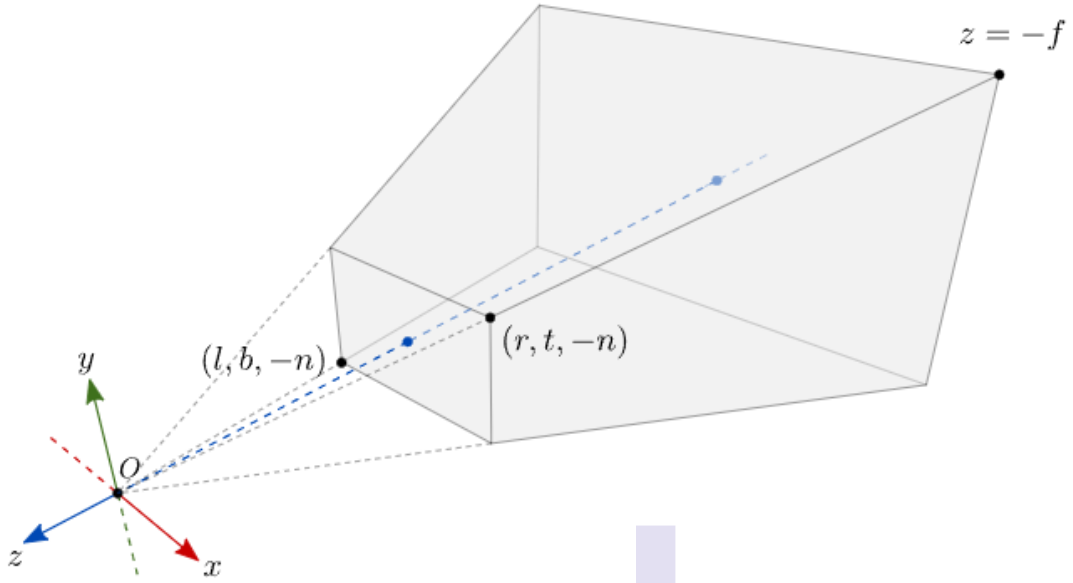


1. **Translação** do volume de visão de modo a centralizá-lo na origem.

2. **Escala** do volume de visão de modo a deixá-lo com tamanho 2 em cada direção.

3. **Reflexão** para inverter a coordenada z.

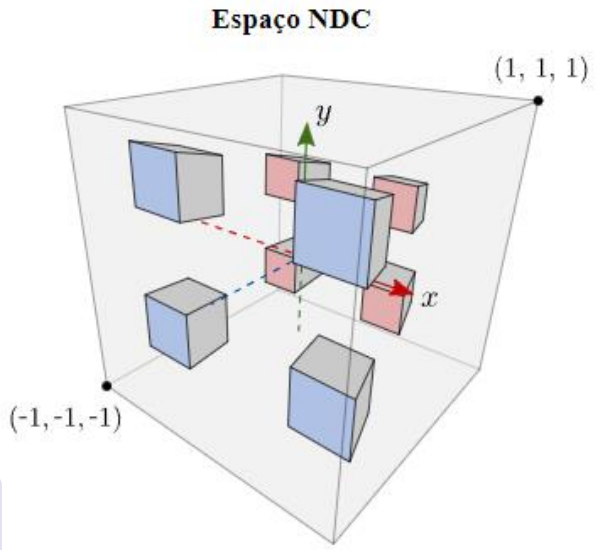
Projeção perspectiva



Em x: [l,r], no espaço da câmera, para [−1,1] em NDC;

Em y: [b,t], no espaço da câmera, para [−1,1] em NDC;

Em z: [−n,−f], no espaço da câmera, para [−1,1] em NDC.



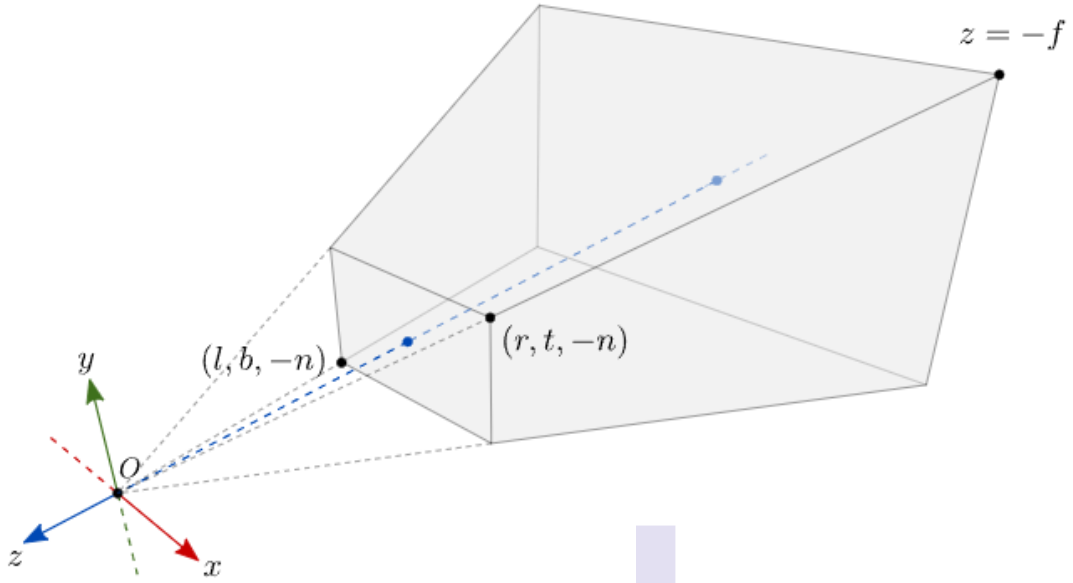
1. **Translação** do volume de visão de modo a centralizá-lo na origem.

2. **Escala** do volume de visão de modo a deixá-lo com tamanho 2 em cada direção.

3. **Reflexão** para inverter a coordenada z.

Estamos mudando de regra (mão direita -> mão esquerda)

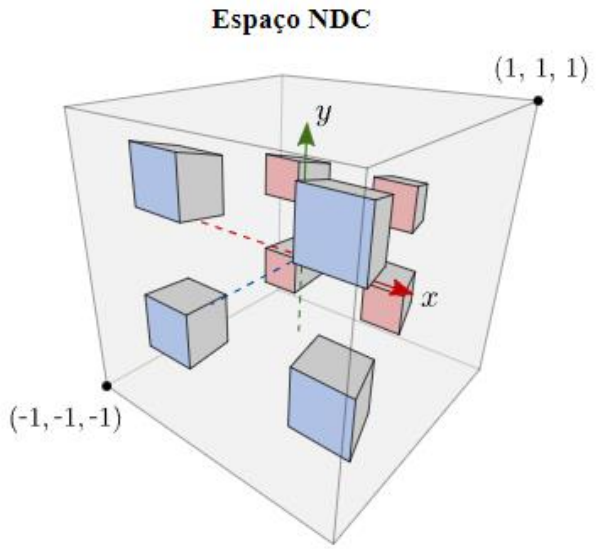
Projeção perspectiva



Em x: [l,r], no espaço da câmera, para [−1,1] em NDC;

Em y: [b,t], no espaço da câmera, para [−1,1] em NDC;

Em z: [−n,−f], no espaço da câmera, para [−1,1] em NDC.



1. **Translação** do volume de visão de modo a centralizá-lo na origem.

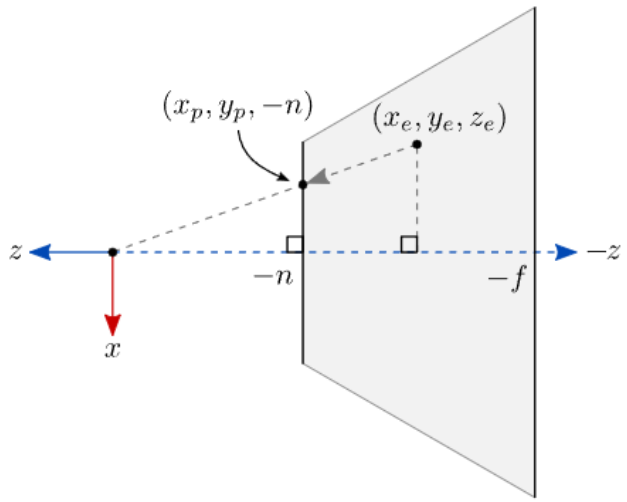
2. **Escala** do volume de visão de modo a deixá-lo com tamanho 2 em cada direção.

3. **Reflexão** para inverter a coordenada z.

Reflexão é uma escala invertendo o sinal, então (2) e (3) podem ser feitas juntas.

Projeção perspectiva

- Antes de aplicar as transformações, vamos revisitar nossa matriz de projeção
  - Agora considerando o plano *near* como o plano de projeção



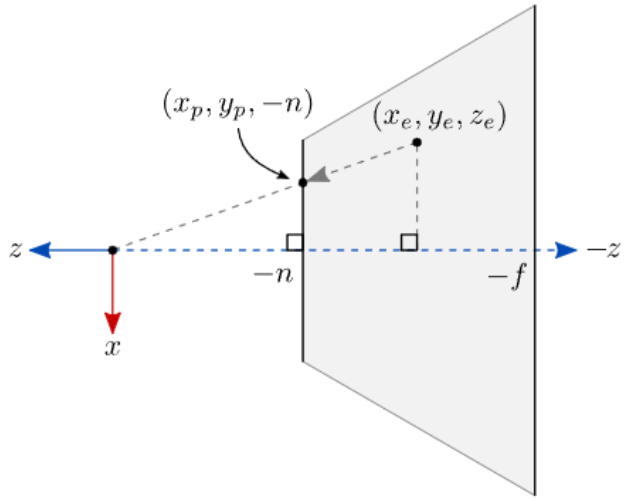
Como um ponto no volume de visão com coordenadas (x_e, y_e, z_e) é projetado para o ponto (x_p, y_p, $-n$) no plano near [onde $z_e = -n$] ?

Obs 1: *e* remete ao *eye space* (os slides 24-32 usavam *c* (câmera) ao invés de *e*). É a mesma coisa.

Obs 2: *p* e *n* equivalem ao *s* e *d* dos slides 24-32. A mudança de termos é porque agora estamos no contexto do plano *near*, e não de um plano de projeção genérico.

Projeção perspectiva

- Antes de aplicar as transformações, vamos revisitar nossa matriz de projeção
  - Agora considerando o plano *near* como o plano de projeção



Como um ponto no volume de visão com coordenadas (x_e, y_e, z_e) é projetado para o ponto (x_p, y_p, -n) no plano near [onde z_e = -n] ?

Semelhança de triângulos!

$$\frac{x_p}{-n} = \frac{x_e}{z_e} \qquad x_p = \frac{-nx_e}{z_e} = \frac{nx_e}{-z_e}$$

$$\frac{y_p}{-n} = \frac{y_e}{z_e} \qquad y_p = \frac{-ny_e}{z_e} = \frac{ny_e}{-z_e}$$

Projeção perspectiva

$$\frac{x_p}{-n} = \frac{x_e}{z_e} \qquad x_p = \frac{-nx_e}{z_e} = \frac{nx_e}{-z_e}$$

$$\frac{y_p}{-n} = \frac{y_e}{z_e} \qquad y_p = \frac{-ny_e}{z_e} = \frac{ny_e}{-z_e}$$

Note que tanto x_e quanto y_e são divididos por $-z_e$. Então...

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$w_c = -z_e$

Para obter coordenadas cartesianas, divida cada componente por w_c.

Projeção perspectiva

$$\frac{x_p}{-n} = \frac{x_e}{z_e} \qquad x_p = \frac{-nx_e}{z_e} = \frac{nx_e}{-z_e}$$

$$\frac{y_p}{-n} = \frac{y_e}{z_e} \qquad y_p = \frac{-ny_e}{z_e} = \frac{ny_e}{-z_e}$$

Note que tanto x_e quanto y_e são divididos por $-z_e$. Então...

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$w_c = -z_e$

Atenção: o c subscrito não remete a "câmera", mas sim a "corte". Ele nada tem a ver com aquele usado nos slides 24-32.

Para obter coordenadas cartesianas, divida cada componente por w_c.

Projeção perspectiva

$$\frac{x_p}{-n} = \frac{x_e}{z_e} \qquad x_p = \frac{-nx_e}{z_e} = \frac{nx_e}{-z_e}$$

$$\frac{y_p}{-n} = \frac{y_e}{z_e} \qquad y_p = \frac{-ny_e}{z_e} = \frac{ny_e}{-z_e}$$

Como é o restante da matriz? Aqui entram as transformações que mencionamos.

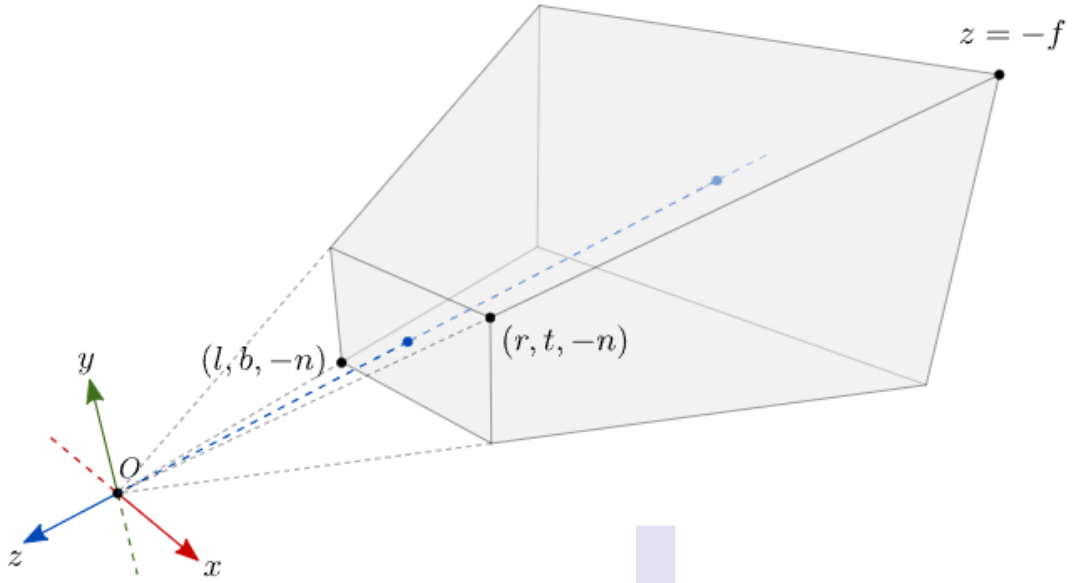
$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

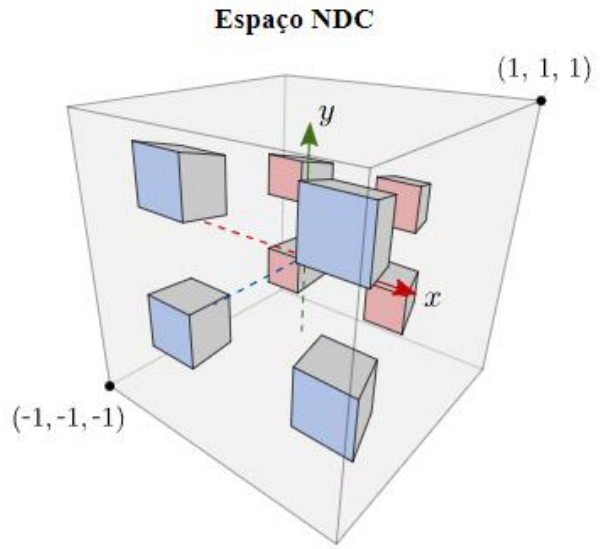
Relembrando...

Em x: [l,r], no espaço da câmera, para [−1,1] em NDC;

Em y: [b,t], no espaço da câmera, para [−1,1] em NDC;

Em z: [−n,−f], no espaço da câmera, para [−1,1] em NDC.





1. **Translação** do volume de visão de modo a centralizá-lo na origem.

2. **Escala** do volume de visão de modo a deixá-lo com tamanho 2 em cada direção.

3. **Reflexão** para inverter a coordenada z.

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

1. **Translação** do volume de visão de modo a centralizá-lo na origem.

O centroide $C = (c_x, c_y, c_z)$ do volume de visão no espaço da câmera é

$$c_x = \frac{r+l}{2}, \qquad c_y = \frac{t+b}{2}, \qquad c_z = -\frac{f+n}{2}.$$

Logo, a matriz de translação que desloca o volume de visão para a origem é a matriz de translação por $-C$:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

2. **Escala** do volume de visão de modo a deixá-lo com tamanho 2 em cada direção.

3. **Reflexão** para inverter a coordenada z.

Os fatores de escala $S = (s_x, s_y, s_z)$ para redimensionar o volume de visão em um cubo com tamanho 2 em cada direção são:

$$s_x = \frac{2}{r-l}, \qquad s_y = \frac{2}{t-b}, \qquad s_z = \frac{2}{f-n}.$$

Entretanto, precisamos refletir o cubo na direção z para a conversão da regra da mão direita para mão esquerda. Assim, precisamos inverter o sinal de s_z:

$$s_z = -\frac{2}{f-n}.$$

$$\mathbf{S} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$\mathbf{M}_{\text{orth}} = \mathbf{ST} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{M}_{\text{orth}} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$\mathbf{M}_{\text{orth}} = \mathbf{ST} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{M}_{\text{orth}} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fatores de translação

Fatores de escala

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$\mathbf{M}_{\text{orth}} = \mathbf{ST} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{M}_{\text{orth}} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fatores de translação

Fatores de escala

$$x_{\text{ndc}} = a_x x_p + b_x$$

x do ponto no NDC (entre -1 e 1)

$$x_p = \frac{n \cdot x_e}{-z_e}$$

x do ponto no plano projetado (Slide 50)

$$a_x = \frac{2}{r-l}$$

$$b_x = -\frac{r+l}{r-l}$$

Fatores para x na matriz ao lado

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$x_{\text{ndc}} = a_x x_p + b_x$$

$$x_p = \frac{n \cdot x_e}{-z_e}$$

$$a_x = \frac{2}{r-l}$$

$$b_x = -\frac{r+l}{r-l}$$

$$\begin{aligned} x_{\text{ndc}} &= a_x x_p + b_x \\ &= \frac{2x_p}{r-l} - \frac{r+l}{r-l} \\ &= \frac{2 \cdot \frac{n \cdot x_e}{-z_e}}{r-l} - \frac{r+l}{r-l} \\ &= \frac{2n \cdot x_e}{-z_e(r-l)} - \frac{r+l}{r-l} \\ &= \frac{\frac{2n}{r-l} \cdot x_e}{-z_e} - \frac{r+l}{r-l} \\ &= \frac{\frac{2n}{r-l} \cdot x_e}{-z_e} + \frac{\frac{r+l}{r-l} \cdot z_e}{-z_e} \\ &= \frac{x_c}{-z_e}, \end{aligned}$$

onde

$$x_c = n \cdot a_x \cdot x_e - b_x \cdot z_e$$

$$= \frac{2n}{r-l} \cdot x_e + \frac{r+l}{r-l} \cdot z_e$$

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$x_{\text{ndc}} = a_x x_p + b_x$$

$$x_p = \frac{n \cdot x_e}{-z_e}$$

$$a_x = \frac{2}{r-l}$$

$$b_x = -\frac{r+l}{r-l}$$

$$\begin{aligned} x_{\text{ndc}} &= a_x x_p + b_x \\ &= \frac{2x_p}{r-l} - \frac{r+l}{r-l} \\ &= \frac{2 \cdot \frac{n \cdot x_e}{-z_e}}{r-l} - \frac{r+l}{r-l} \\ &= \frac{2n \cdot x_e}{-z_e(r-l)} - \frac{r+l}{r-l} \\ &= \frac{\frac{2n}{r-l} \cdot x_e}{-z_e} - \frac{r+l}{r-l} \\ &= \frac{\frac{2n}{r-l} \cdot x_e}{-z_e} + \frac{\frac{r+l}{r-l} \cdot z_e}{-z_e} \\ &= \frac{x_c}{-z_e}, \end{aligned}$$

onde

$$x_c = n \cdot a_x \cdot x_e - b_x \cdot z_e$$

$$= \frac{2n}{r-l} \cdot x_e + \frac{r+l}{r-l} \cdot z_e$$

De forma análoga para o y_{ndc}

$$y_{\text{ndc}} = \frac{y_c}{-z_e},$$

onde

$$y_c = n \cdot a_y \cdot y_e - b_y \cdot z_e$$

$$= \frac{2n}{t-b} \cdot y_e + \frac{t+b}{t-b} \cdot z_e$$

$$a_y = \frac{2}{t-b}$$

$$b_y = -\frac{t+b}{t-b}$$

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$x_c = n \cdot a_x \cdot x_e - b_x \cdot z_e$$

$$= \frac{2n}{r-l} \cdot x_e + \frac{r+l}{r-l} \cdot z_e$$

$$y_c = n \cdot a_y \cdot y_e - b_y \cdot z_e$$

$$= \frac{2n}{t-b} \cdot y_e + \frac{t+b}{t-b} \cdot z_e$$

Acabamos de encontrar as duas primeiras linhas da nossa matriz acima

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ . & . & . & . \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

- Ainda falta a terceira linha da matriz
  - Seus elementos correspondem à transformação de z_e (z em coordenadas de câmera/eye) em z_c (z em coordenadas homogêneas de corte)
  - O valor de z_c não depende de x_e e y_e, então esses valores são zero. Vamos chamar os outros dois valores de α e β.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ . & . & . & . \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$z_c = \alpha z_e + \beta w_e$$

Após a divisão por w_c , que vale $-z_e$ (slide 51):

$$z_{ndc} = \frac{\alpha z_e + \beta w_e}{-z_e}$$

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ . & . & . & . \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$z_c = \alpha z_e + \beta w_e$$

Após a divisão por w_c , que vale $-z_e$ (slide 51):

$$z_{ndc} = \frac{\alpha z_e + \beta w_e}{-z_e}$$

Obs: w_e = 1 no espaço da câmera =)

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ . & . & . & . \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$z_c = \alpha z_e + \beta w_e$$

Após a divisão por w_c:

$$z_{ndc} = \frac{\alpha z_e + \beta w_e}{-z_e}$$

Obs: w_e = 1 no espaço da câmera =)

Sabemos que [-n,-f] deve ser mapeado para [-1,1], então:

$$\frac{-\alpha n + \beta}{n} = -1 \qquad \frac{-\alpha f + \beta}{f} = 1 \qquad \rightarrow \qquad -\alpha n + \beta = -n \qquad -\alpha f + \beta = f \qquad \rightarrow \qquad \alpha = -\frac{f+n}{f-n} \qquad \beta = -\frac{2fn}{f-n}$$

Projeção perspectiva

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \alpha & \beta \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$

$$\mathbf{M}_{\text{persp}} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

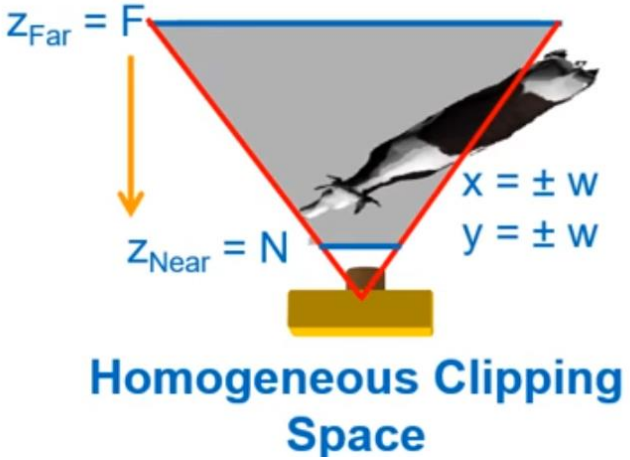
Relembrando:
left, right, top, bottom, near, far

No OpenGL: glm::frustum(float left, float right, float bottom, float top, float zNear, float zFar)

Projeção perspectiva

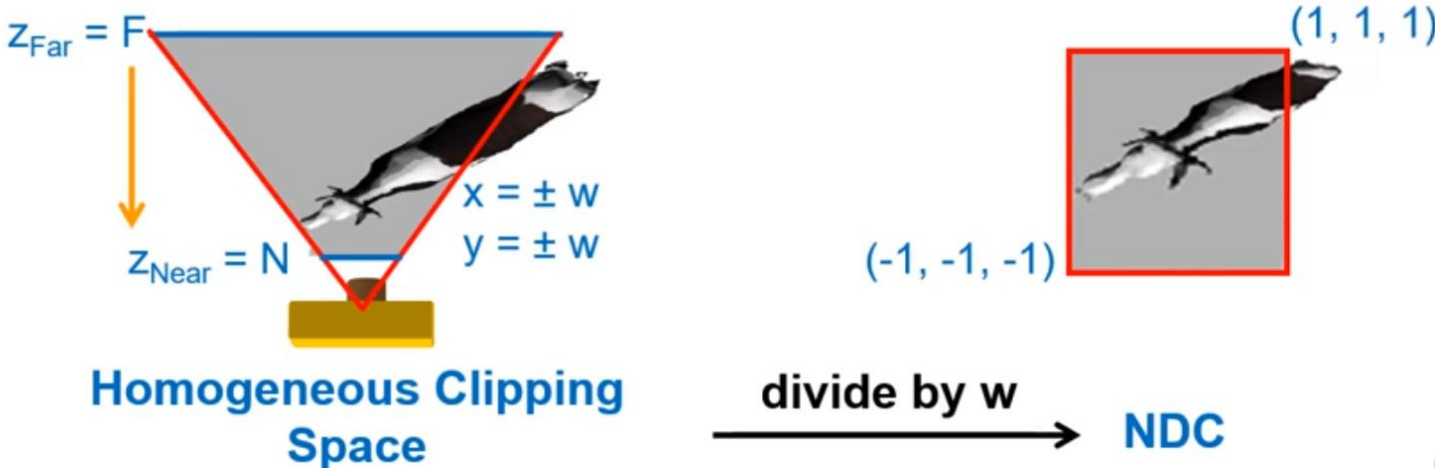
- Após multiplicar os vértices pela matriz de projeção, chegaremos no espaço de corte (*Homogeneous Clipping Space)*
  - Cubo com dimensões [-w,w]

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ w_c \end{bmatrix} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} x_e \\ y_e \\ z_e \\ w_e \end{bmatrix}$$



- Ainda falta normalizar, ou seja, dividir as coordenadas por w, para chegar no NDC.
  - Cubo com dimensões [-1,1]

Exibição, clipping e culling



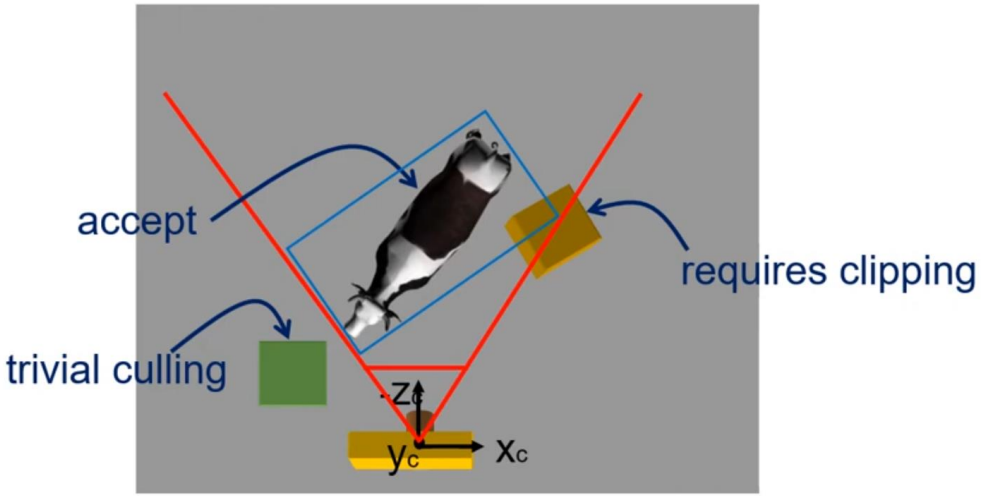
Em NDC, os valores de x e y serão usados para exibição da imagem (projeção paralela ortográfica)

É trivial decidir os vértices que serão renderizados e os que serão descartados (basta ver se as coordenadas do vértice estão entre [-w,w]).

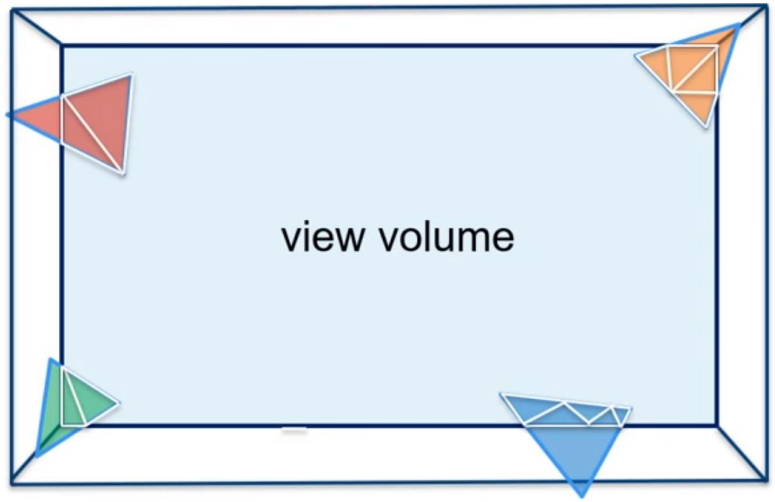
Exibição, clipping e culling

É trivial decidir os vértices que serão renderizados e os que serão descartados (basta ver se as coordenadas do vértice estão entre [-w,w]).

- *Culling* trivial



- *Clipping* não-trivial (exige formar novos triângulos)



Projeção perspectiva – forma alternativa

- Existe outra forma de obter a matriz projection
  - Sem depender de left, right, top, bottom
  - Eles podem ser obtidos por meio de:
  1. O ângulo de visão vertical da câmera (*field of view* – FOV) e
  2. O *aspect ratio* (largura/altura) do plano de projeção

No OpenGL: glm::perspective(float fovy, float aspect, float zNear, float zFar)

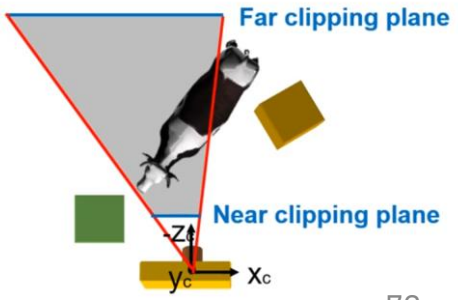
Detalhes sobre como isso é feito e sobre FOV vertical e horizontal aqui.

Projeção perspectiva – forma alternativa

- Existe outra forma de obter a matriz projection
  - Sem depender de left, right, top, bottom
  - Eles podem ser obtidos por meio de:
  1. O ângulo de visão vertical da câmera (*field of view* – FOV) e
  2. O *aspect ratio* (largura/altura) do plano de projeção

No OpenGL: glm::perspective(float fovy, float aspect, float zNear, float zFar)

- Essa forma só funciona se o volume de visão for **simétrico**
  - Não funciona para a situação da figura ao lado



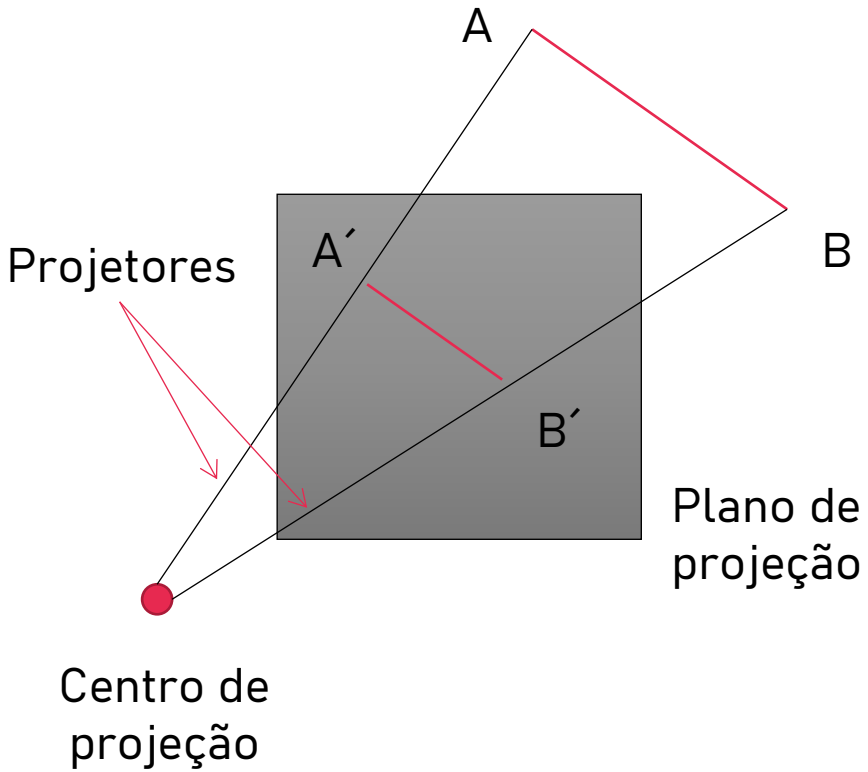
Projeção perspectiva – forma alternativa

Jupyter Notebook

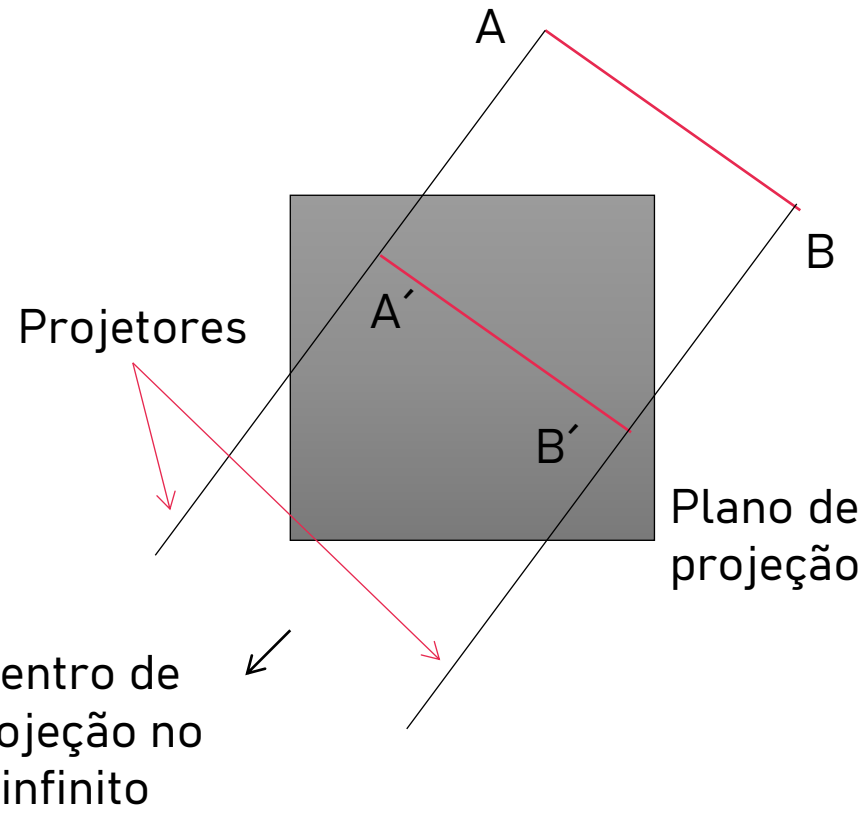
Projeção Paralela Ortográfica

Relembrando...

Exemplo 1:
Projeção **perspectiva** da reta AB



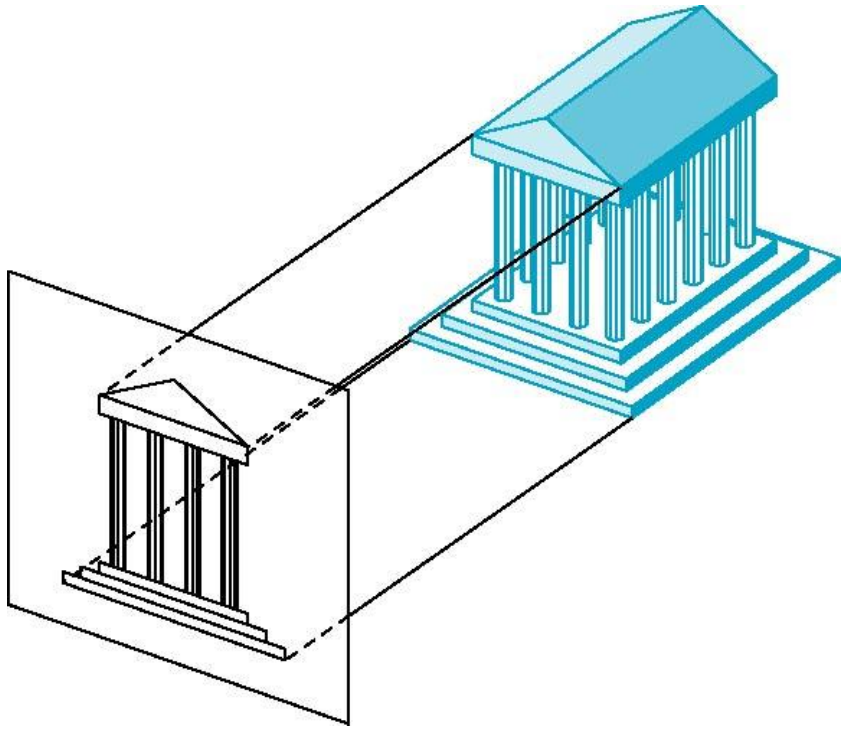
Exemplo 2:
Projeção **paralela** da reta AB



Projeção paralela ortográfica

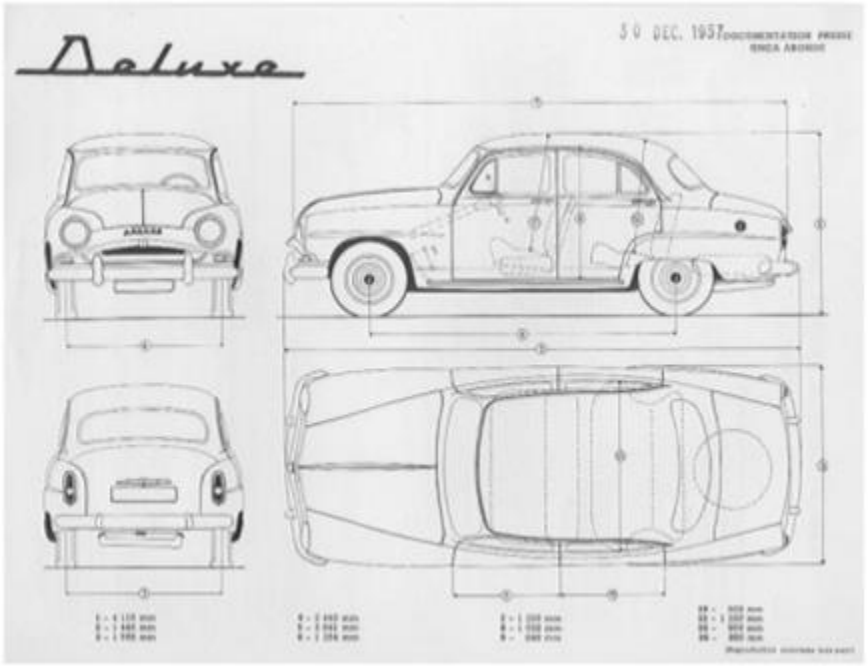
- Preserva distâncias e ângulos
  - Formas são preservadas
  - Podem ser usadas para o cálculo de medidas

- Não é possível visualizar como o objeto realmente aparenta pois muitas superfícies ficam escondidas da projeção

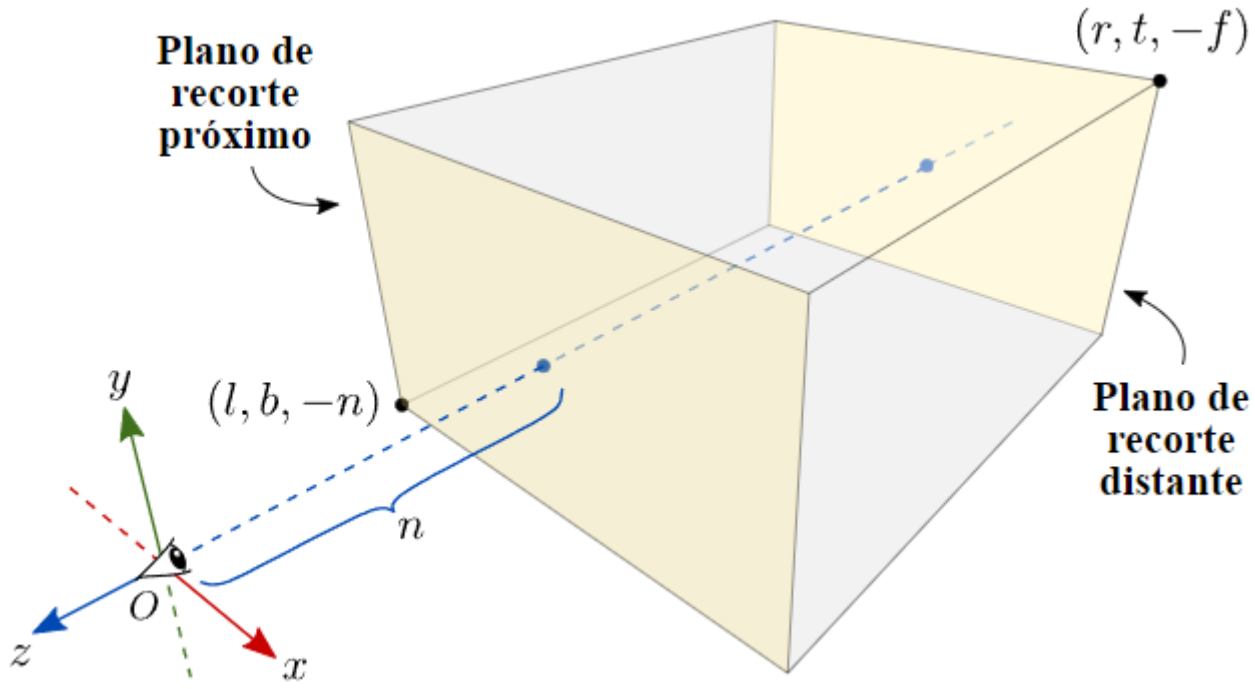


Projeção paralela ortográfica

- É uma forma de projeção importante em desenhos técnicos de engenharia, arquitetura e outros.



Projeção paralela ortográfica



Sistema de coordenadas da câmera

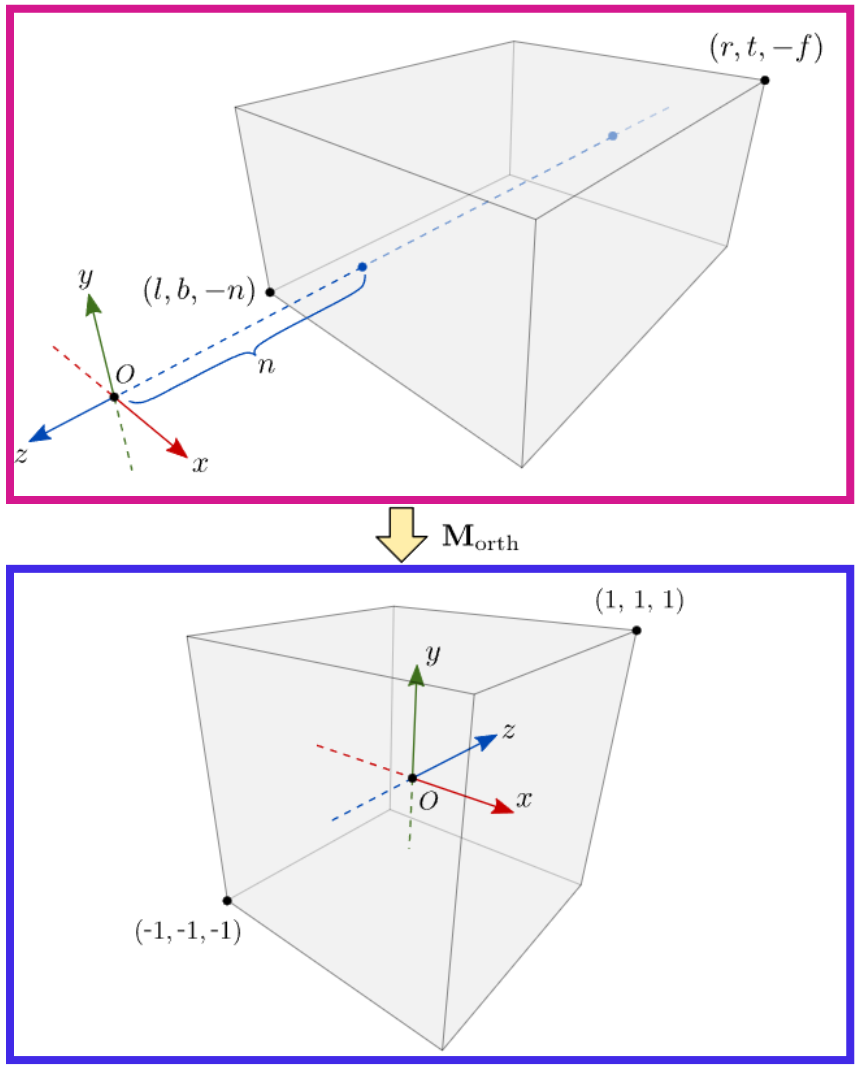
Projeção paralela ortográfica

Mesmo objetivo de antes:

Obter uma matriz (M_{orth}) que transforma

o volume de visão, do espaço da câmera,

em um cubo [-1,1] em NDC.

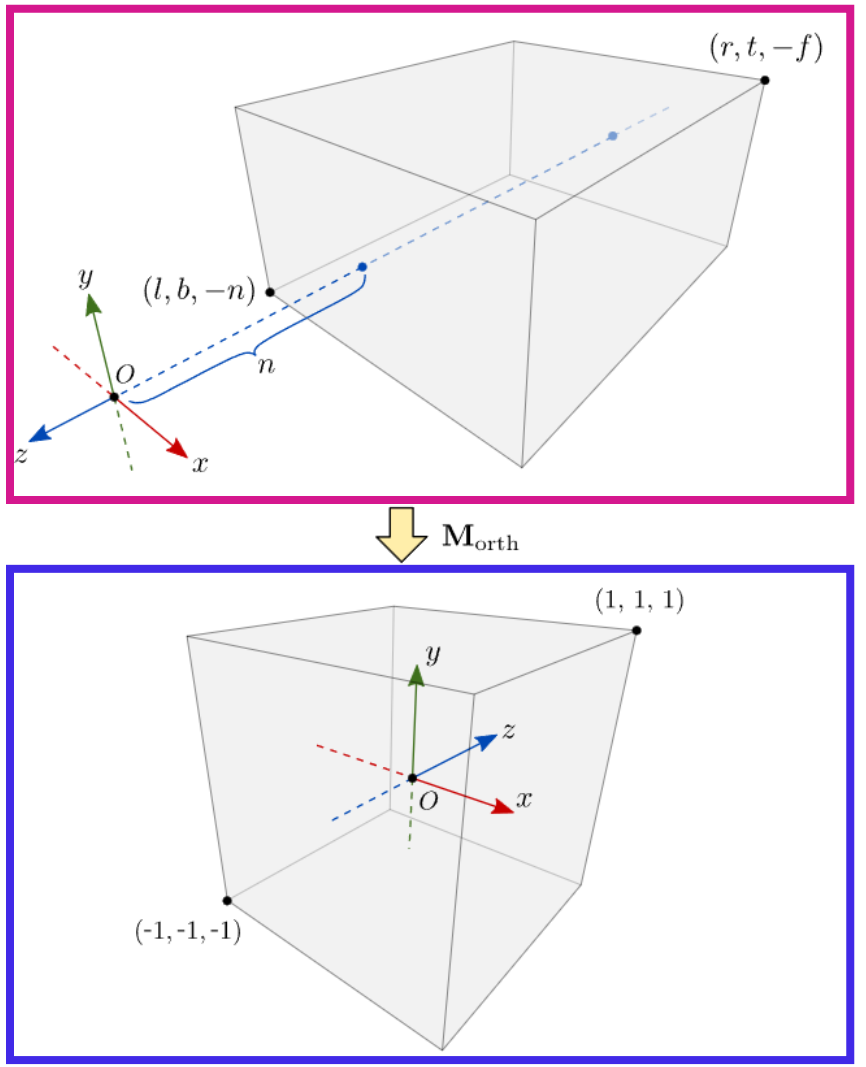


Projeção paralela ortográfica

Essa transformação precisa fazer com que:

(l,b,-n) se torne (-1,-1,-1)

(r,t,-f) se torne (1,1,1)



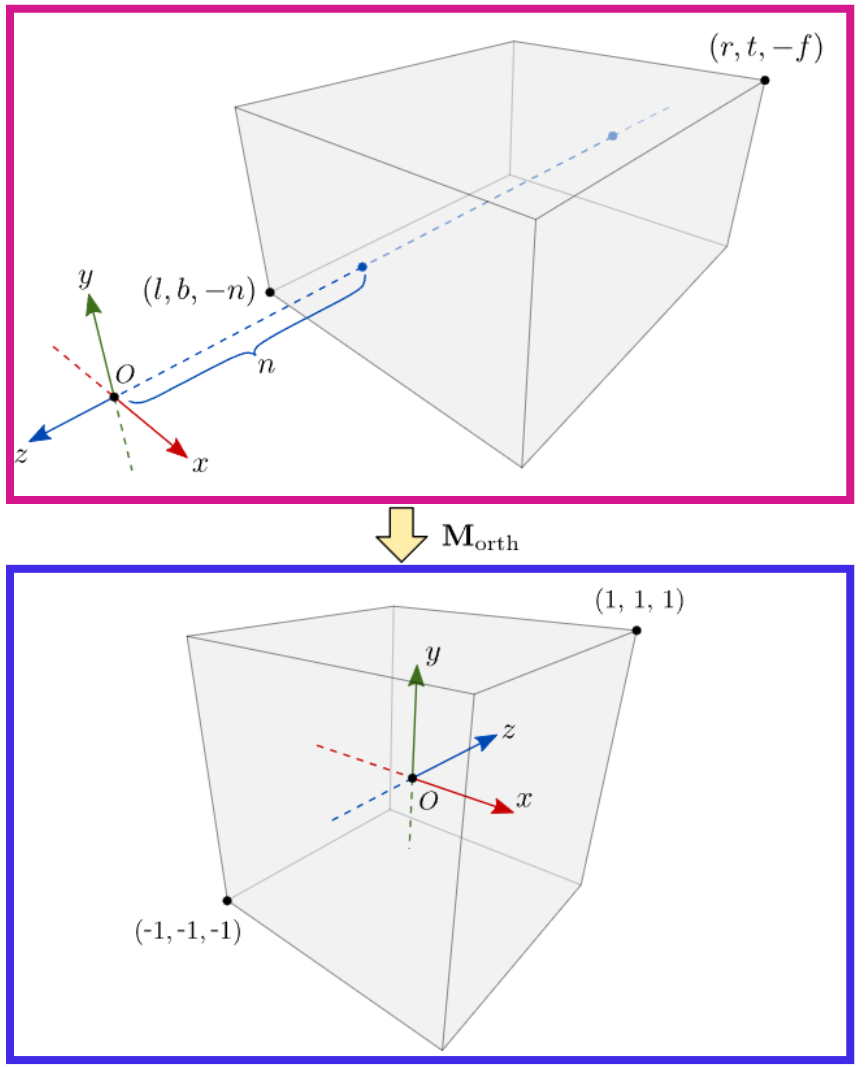
Projeção paralela ortográfica

Essa transformação precisa fazer com que:

(l,b,-n) se torne (-1,-1,-1)

(r,t,-f) se torne (1,1,1)

Já vimos como fazer isso =)

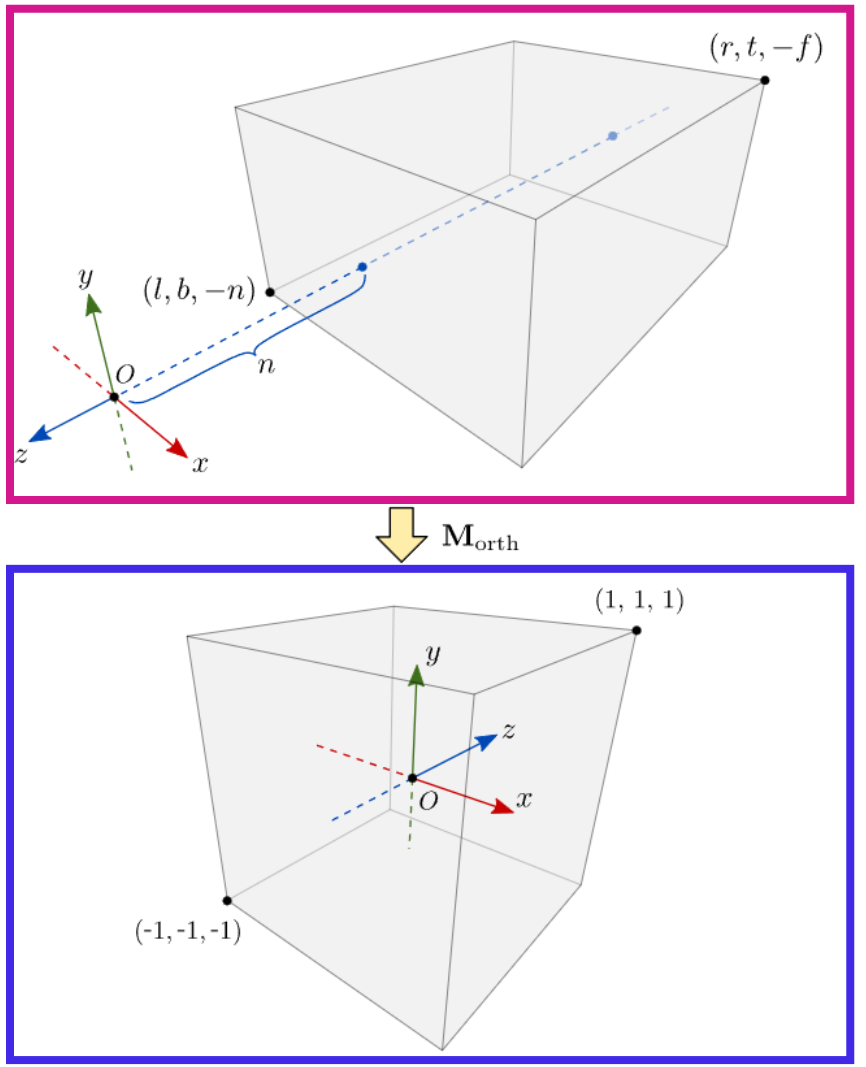


Projeção paralela ortográfica

1. **Translação** do volume de visão de modo a centralizá-lo na origem.

2. **Escala** do volume de visão de modo a deixá-lo com tamanho 2 em cada direção.

3. **Reflexão** para inverter a coordenada z.



Revisitando o slide 55...

1. **Translação** do volume de visão de modo a centralizá-lo na origem.

O centroide $C = (c_x, c_y, c_z)$ do volume de visão no espaço da câmera é

$$c_x = \frac{r+l}{2}, \qquad c_y = \frac{t+b}{2}, \qquad c_z = -\frac{f+n}{2}.$$

Logo, a matriz de translação que desloca o volume de visão para a origem é a matriz de translação por $-C$:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Revisitando o slide 56...

2. **Escala** do volume de visão de modo a deixá-lo com tamanho 2 em cada direção.

3. **Reflexão** para inverter a coordenada z.

Os fatores de escala $S = (s_x, s_y, s_z)$ para redimensionar o volume de visão em um cubo com tamanho 2 em cada direção são:

$$s_x = \frac{2}{r-l}, \qquad s_y = \frac{2}{t-b}, \qquad s_z = \frac{2}{f-n}.$$

Entretanto, precisamos refletir o cubo na direção z para a conversão da regra da mão direita para mão esquerda. Assim, precisamos inverter o sinal de s_z:

$$s_z = -\frac{2}{f-n}.$$

$$\mathbf{S} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Revisitando o slide 57...

$$\mathbf{M}_{\text{orth}} = \mathbf{ST} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{M}_{\text{orth}} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Essa é nossa matriz de projeção paralela ortográfica.

Matriz de projeção ortográfica

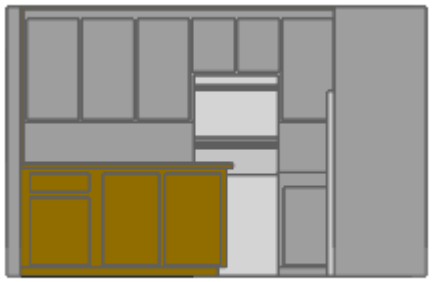
$$\mathbf{M}_{\text{orth}} = \mathbf{ST} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & -\frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{M}_{\text{orth}} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & -\frac{2}{f-n} & -\frac{f+n}{f-n} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

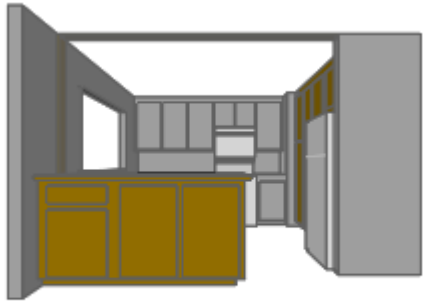
Essa é nossa matriz de projeção paralela ortográfica.

No OpenGL, glm::ortho(float left, float right, float bottom, float top, float zNear, float zFar)

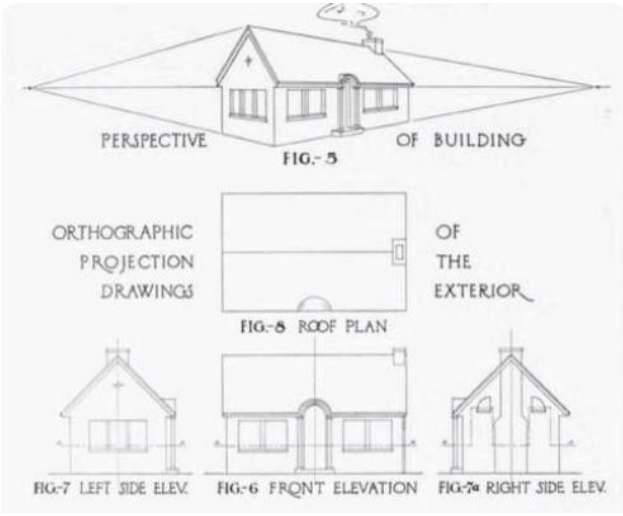
Paralela vs Perspectiva

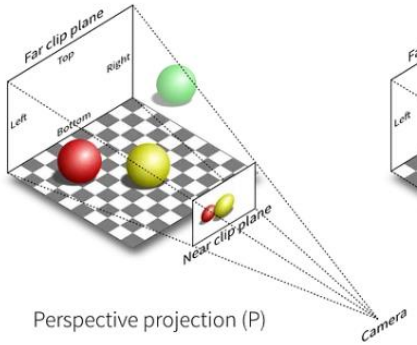


Parallel projection

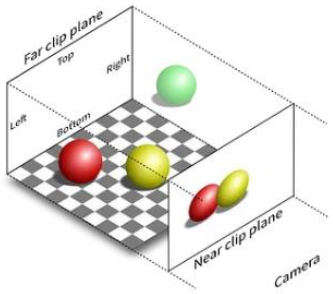


Perspective projection





Perspective projection (P)



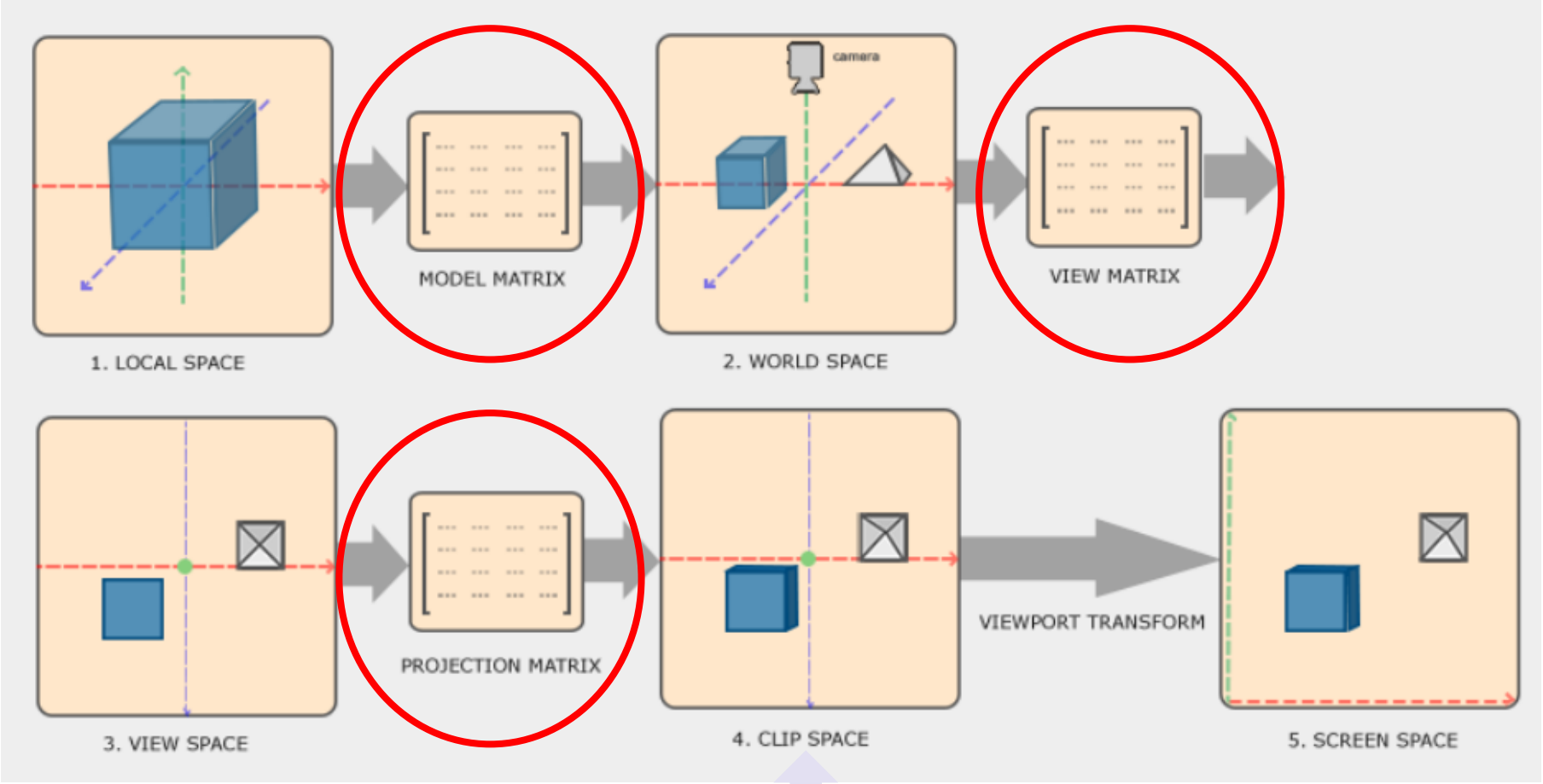
Orthographic projection (O)

NDC para coordenadas de tela

A matriz *Viewport*

Pipeline de transformações

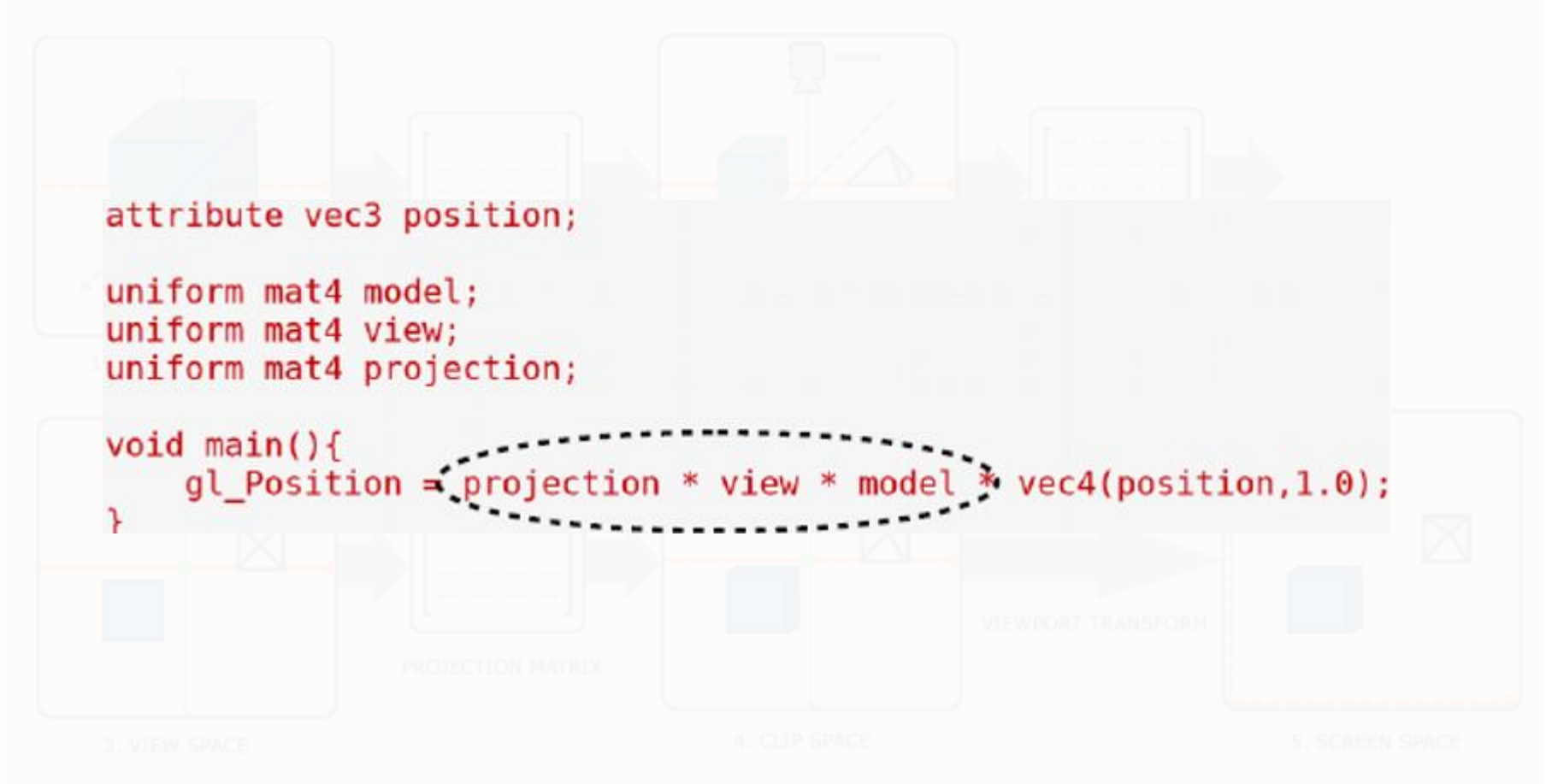
P' = Projection x View x Model x P



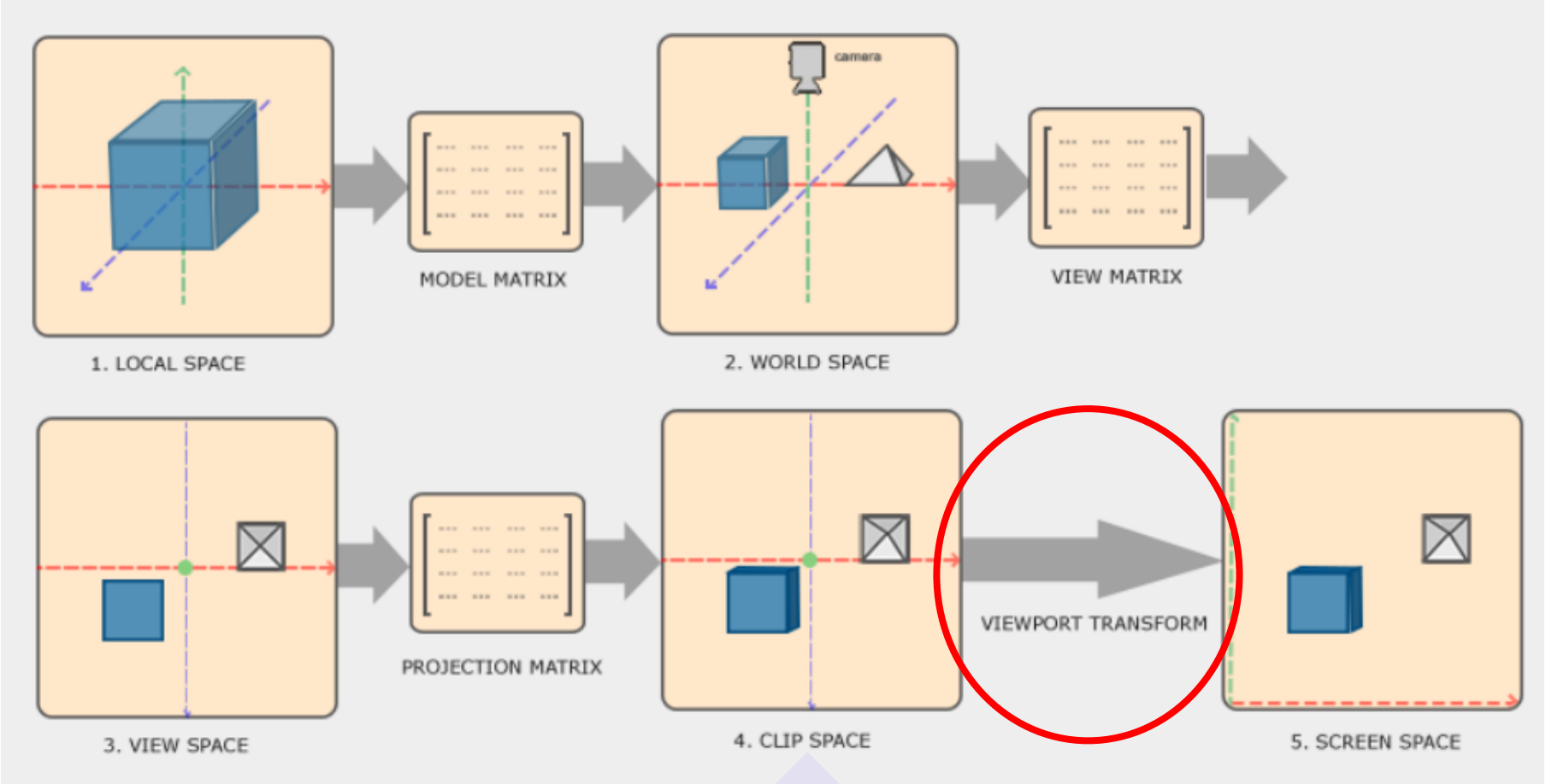
NDC (cubo entre [-1,1])

Pipeline de transformações

P' = Projection x View x Model x P

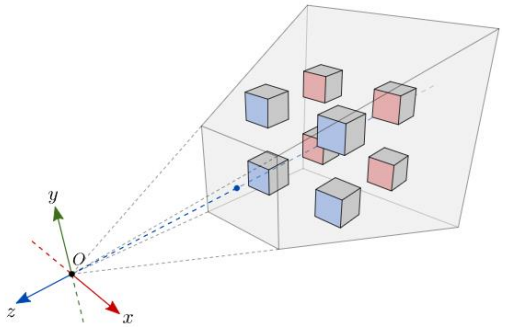


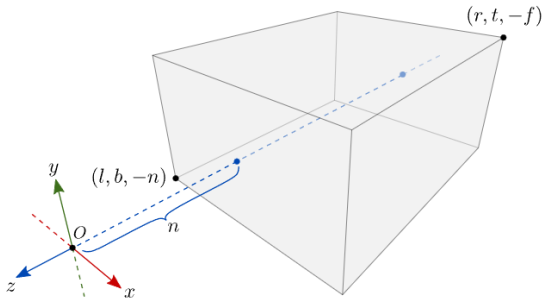
Pipeline de transformações



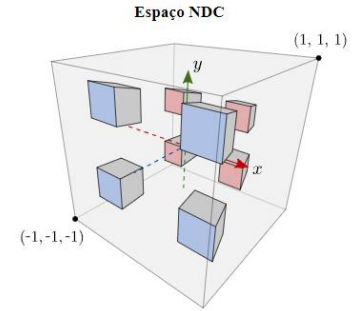
NDC (cubo entre [-1,1])

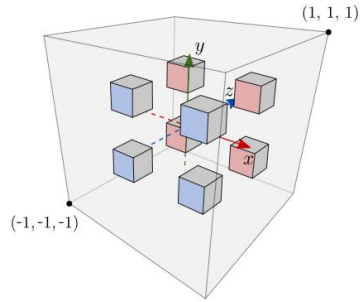
Matriz *viewport*



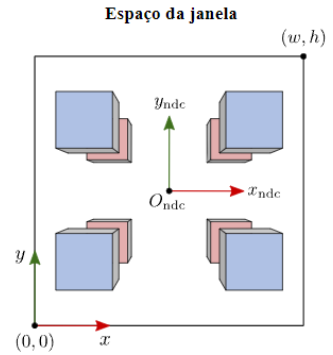


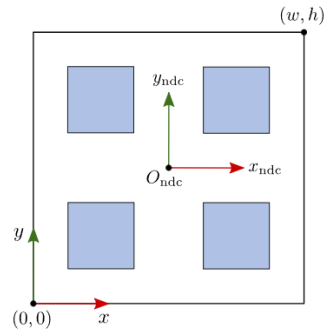
Volume de visão (espaço da câmera)





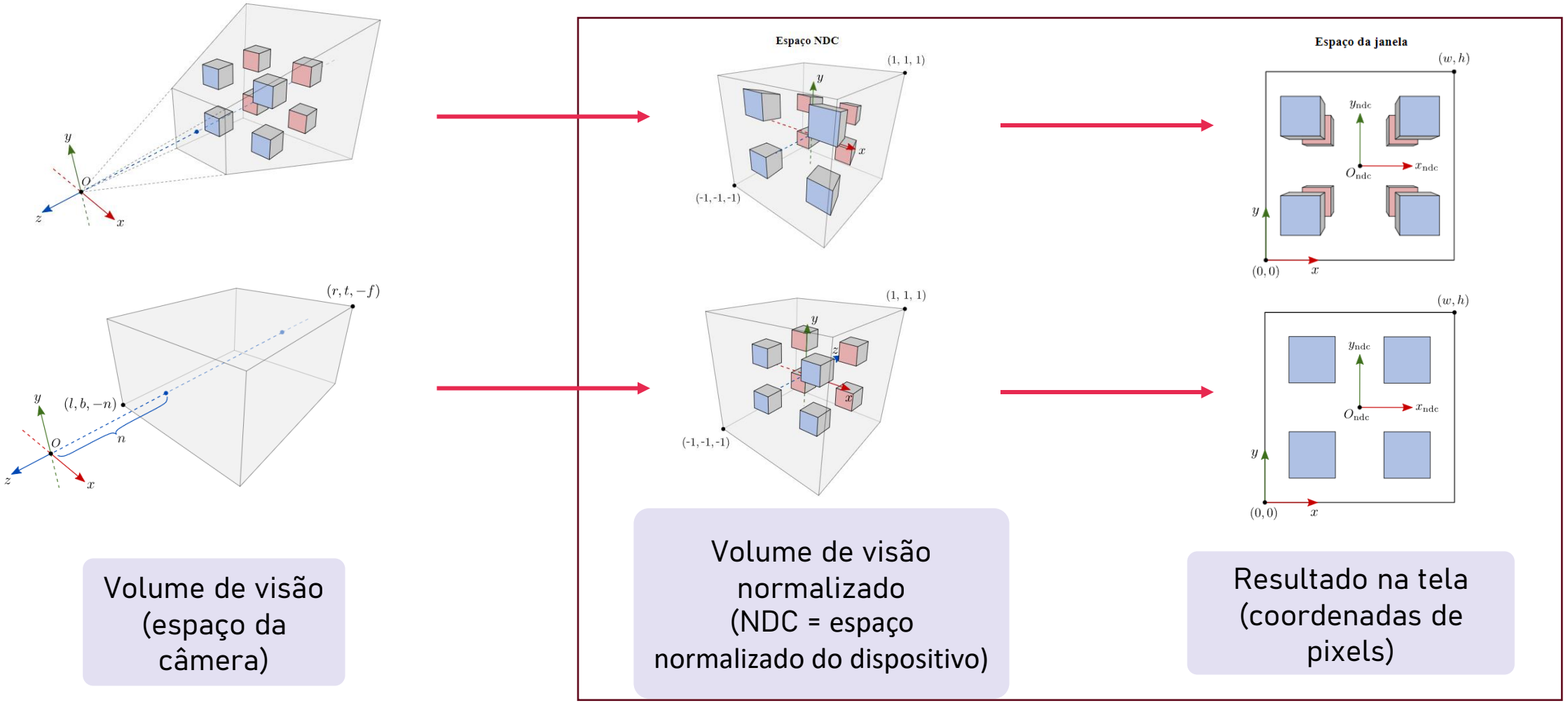
Volume de visão normalizado (NDC = espaço normalizado do dispositivo)





Resultado na tela (coordenadas de pixels)

Matriz *viewport*



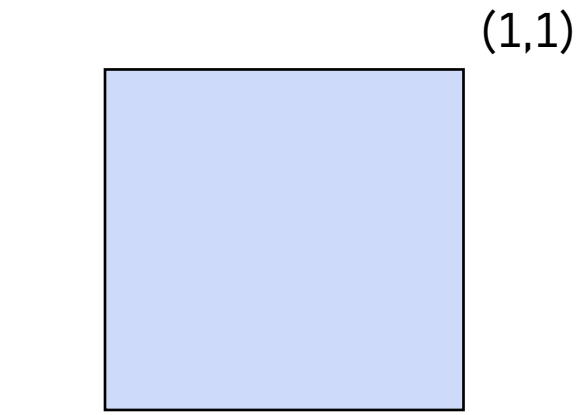
Matriz *viewport*

- Fizemos a **projeção**, que incluiu o clipping/culling e divisão perspectiva (divisão por w)

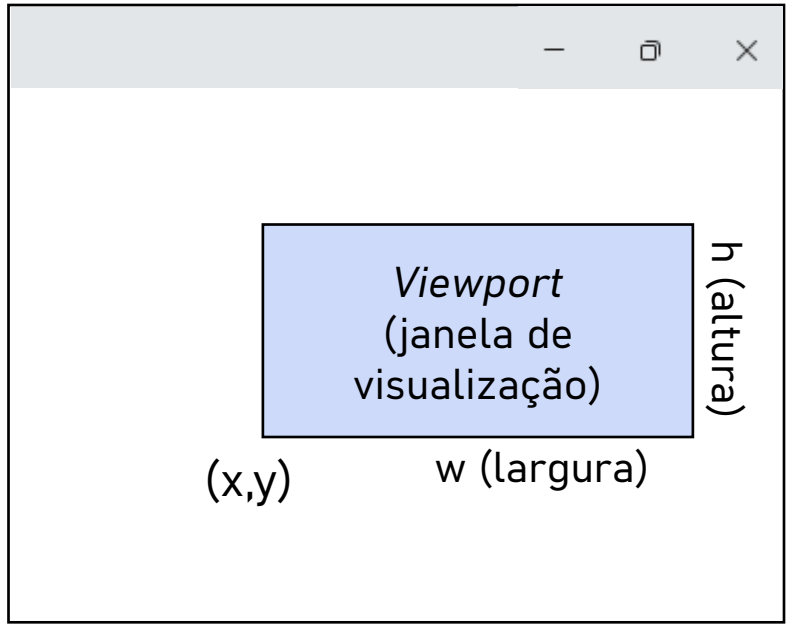
- Estamos prontos para fazer a **rasterização**
  - Ela ocorre em coordenadas de pixel
  - Precisamos então converter do NDC (cubo [-1,1]) para coordenadas de pixel (espaço da janela)

Matriz *viewport*

- Transformação do NDC (cubo [-1,1]) para coordenadas de pixel
  - Vamos usar só as coordenadas (x,y) do NDC aqui.
  - A coordenada z será usada para *depth-buffering*, i.e., para determinar o que será visível.

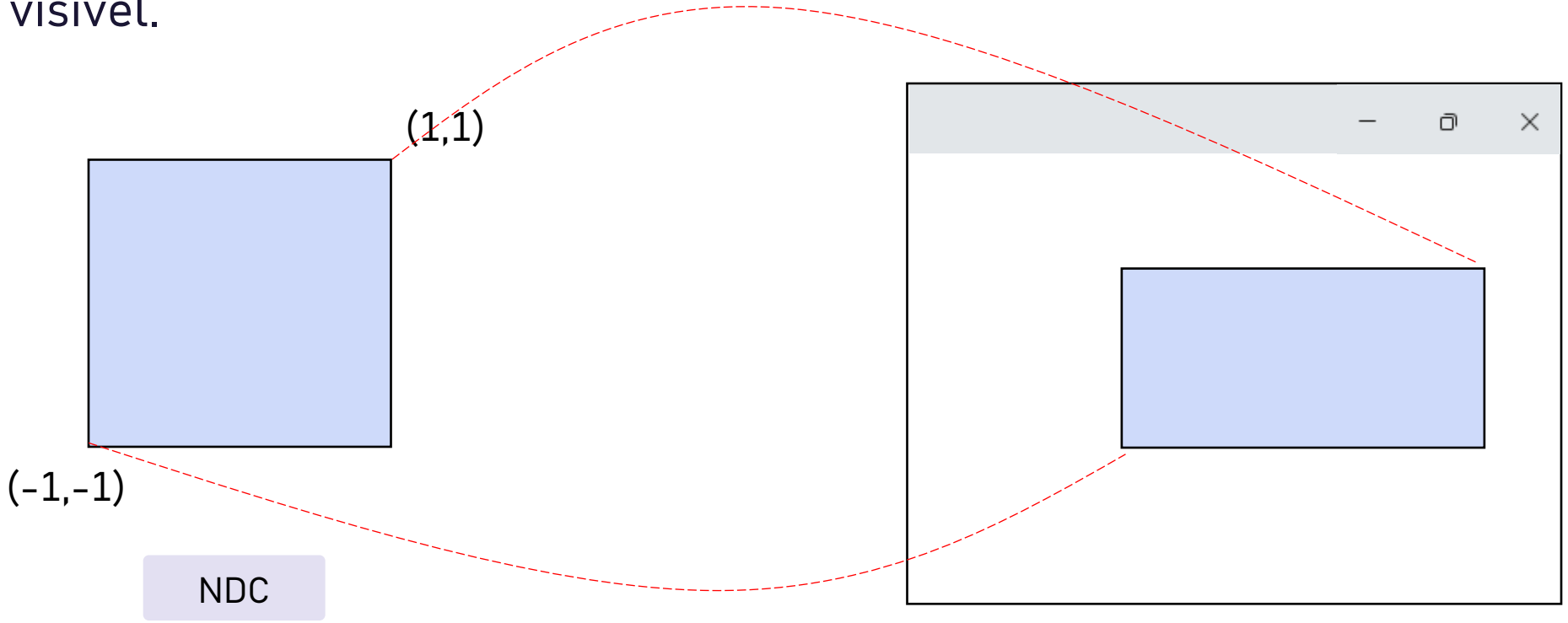


NDC



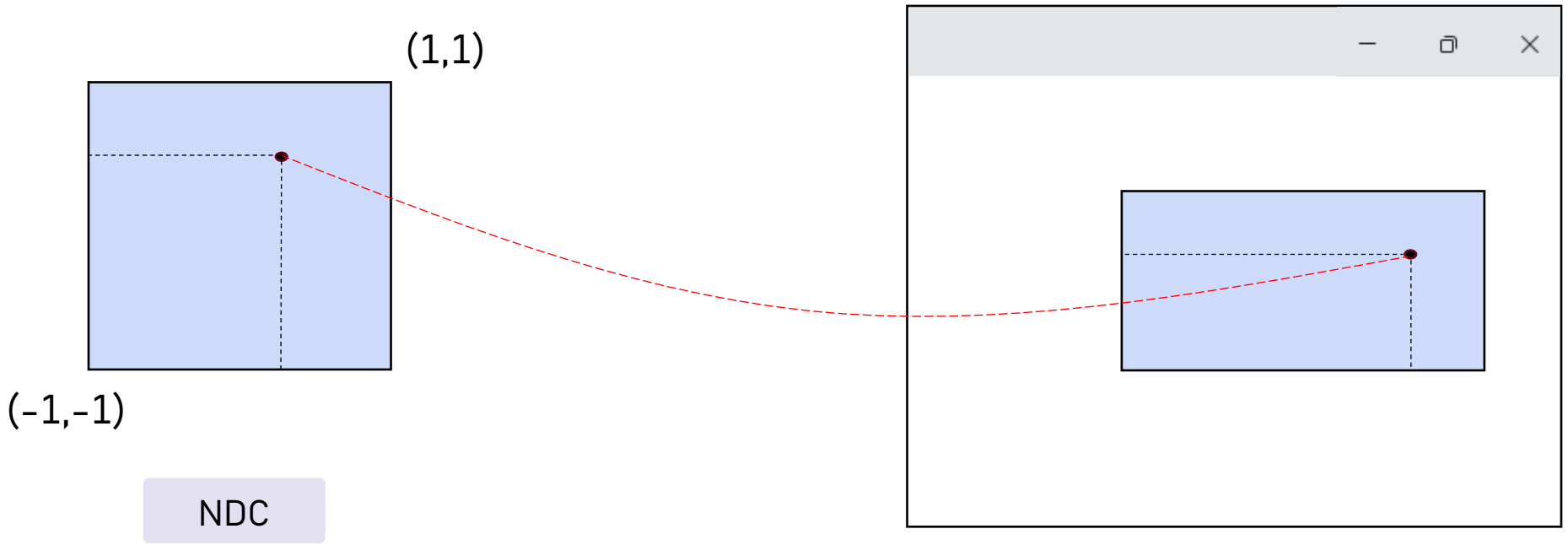
Matriz *viewport*

- Transformação do NDC (cubo [-1,1]) para coordenadas de pixel
  - Vamos usar só as coordenadas (x,y) do NDC aqui.
  - A coordenada z será usada para *depth-buffering*, i.e., para determinar o que será visível.

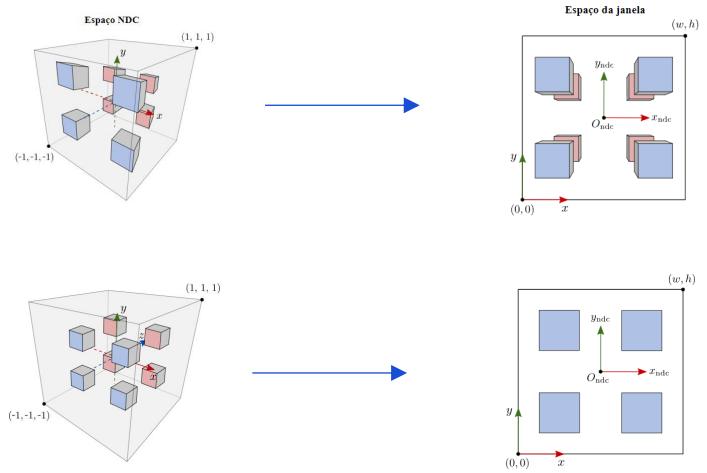


Matriz *viewport*

- Transformação do NDC (cubo [-1,1]) para coordenadas de pixel
  - Vamos usar só as coordenadas (x,y) do NDC aqui.
  - A coordenada z será usada para *depth-buffering*, i.e., para determinar o que será visível.



Matriz *viewport*



Considerando w = largura, h = altura, f = far, n = near:

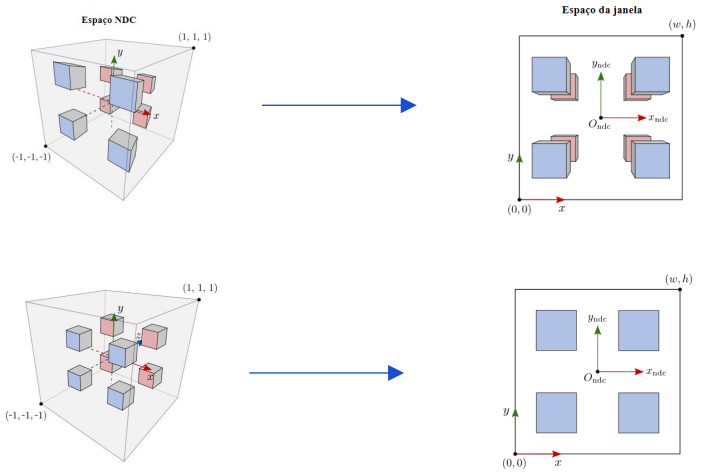
$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} \frac{w}{2} & 0 & 0 & x + \frac{w}{2} \\ 0 & \frac{h}{2} & 0 & y + \frac{h}{2} \\ 0 & 0 & \frac{f-n}{2} & \frac{f+n}{2} \end{bmatrix} \begin{bmatrix} x_{\text{ndc}} \\ y_{\text{ndc}} \\ z_{\text{ndc}} \\ 1 \end{bmatrix}$$

Espaço de janela
(x,y) coordenadas de pixel
z = profundidade (*depth-buffering*)

Matriz viewport

Espaço NDC (cubo [-1,1])
w = 1 pois houve a divisão por w na projeção

Matriz *viewport*



Considerando w = largura, h = altura, f = far, n = near:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} \frac{w}{2} & 0 & 0 & x + \frac{w}{2} \\ 0 & \frac{h}{2} & 0 & y + \frac{h}{2} \\ 0 & 0 & \frac{f-n}{2} & \frac{f+n}{2} \end{bmatrix} \begin{bmatrix} x_{\text{ndc}} \\ y_{\text{ndc}} \\ z_{\text{ndc}} \\ 1 \end{bmatrix}$$

No OpenGL:

`glViewport(X, Y, W, H)` -> configura viewport (default: X = 0, Y = 0 (canto inferior esquerdo da janela))
`glDepthRangef(N, F)` -> mapeia valores de profundidade (default: N = 0, F = 1)

Cronograma (revisitado)

- Veremos Rasterização após a primeira prova.

- O conteúdo dessa prova abrange até o conteúdo que vimos hoje.

S	09/set	Projection Matrix
Q	11/set	Projection (pt 2) + Exercícios de Revisão
S	**16/set**	**Apresentação do Trabalho 1**
Q	18/set	Prática Model-View-Projection **(lab 1-004)**
S	23/set	Plantão de dúvidas
Q	**25/set**	**Prova 1**

Bibliografia

- Essa aula foi baseada e utiliza exemplos/figuras que foram retirados do seguinte material:

- https://www.brunodorta.com.br/cg/projections [Capítulo 8]

- Computação Gráfica, Slides de Ricardo Marcacini, Maria Cristina, Alaor Cervati, Agma Traina, Mirela Cazzolato. ICMC/USP.

- Computação Gráfica, Slides de Bruno Travençolo, FACOM/UFU.

- Vídeos do Prof. Manuel Menezes de Oliveira Neto, Instituto de Informática, UFRGS
  - https://www.youtube.com/watch?v=9opn8G8pScs&list=PLQ7b--ynwIXhP-MrKHdEKDu3T4plN5jYe&index=8&ab_channel=ManuelM.Oliveira
  - https://www.youtube.com/watch?v=lHrFhympQvQ&list=PLQ7b--ynwIXhP-MrKHdEKDu3T4plN5jYe&index=8&ab_channel=ManuelM.Oliveira