

# Computação Gráfica

## Coordenadas, Projeções e Câmera

Prof. Gabriel M. Alves

2024-10-14

versão: b66e7e-dirty

- Há diferentes sistemas de coordenadas para descrever os objetos modelados em um sistema.
- Um sistema de coordenadas serve como referência em termos de medidas do tamanho e posição dos objetos na área de trabalho.
- Exemplos de sistemas:
  - coordenadas polares: raio e ângulo
  - coordenadas esféricas: raio e dois ângulos
  - coordenadas cilíndricas: raio, ângulo e um comprimento
- Há diferentes sistemas de referências.

# Sistemas de coordenadas

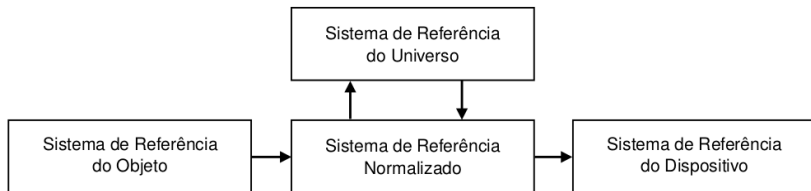


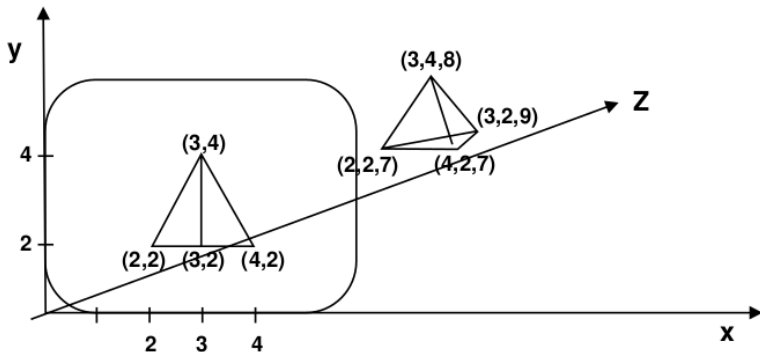
Figure: Diferentes sistemas de referências.

- **Sistema de Referência do Universo (SRU)**: é chamado de coordenadas do universo, ou *espaço do mundo* (*world space coordinates*). É o sistema de referência utilizado para descrever objetos em termos de coordenadas utilizadas em uma determinada aplicação. Exemplo: Aviação (coordenadas polares).
- **Sistema de Referência do Objeto (SRO)**: trata cada objeto tem suas particularidades descritas em função do seu sistema (“mini universo”), muitas vezes coincidindo com o centro do sistema de coordenadas com o seu centro de gravidade. Na modelagem este centro é conhecido como **pivô**. Também pode ser entendido como *espaço local* ou *espaço do objeto*.

- **Sistema de Referência Normalizado (SRN)**: trabalha com coordenadas normalizadas, por exemplo:  $-1 \leq x \leq 1$  e  $-1 \leq y \leq 1$ . Serve como intermediário entre o SRU e o SRD e torna a geração de imagens independente do dispositivo. Também pode ser entendido como coordenadas normalizadas do dispositivo (*Normalized Device Coordinates*).
- **Sistema de Referência do Dispositivo (SRD)**: coordenadas fornecidas para um dado dispositivo de saída. Nos hardwares, o sistema de coordenadas depende geralmente da resolução.

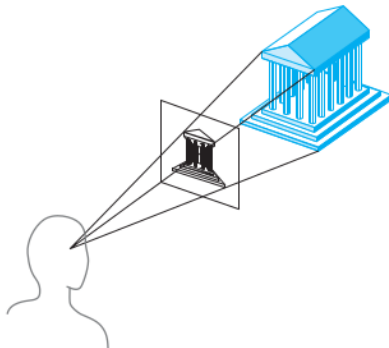
# Projeções

- **Projeções** permitem a visualização 2D de objetos 3D.



# Projeções

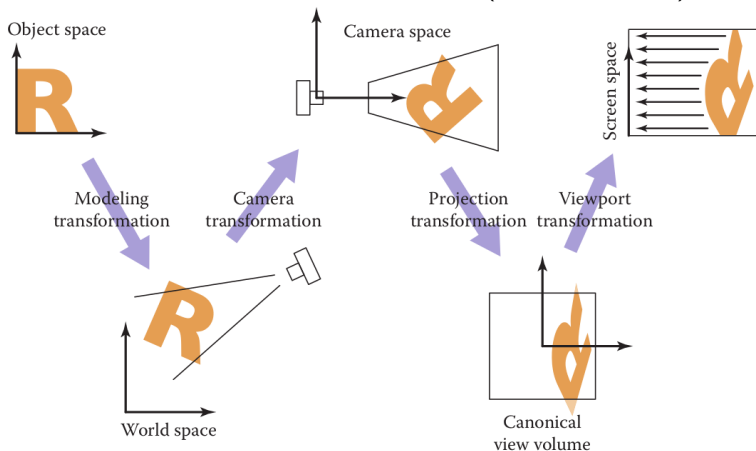
- Para representação de um objeto 3D em um plano 2D devem ser considerados os elementos básicos: plano de projeção, raios projetantes e centro de projeção.



- Há diferentes tipos de projeções

# Viewing Transformation

- Mapear um objeto 3D para uma imagem (plano 2D) é uma tarefa complicada, pois depende da posição e orientação da câmera, o tipo de projeção, o campo de visão, a resolução da imagem.
- O processo é então dividido em três etapas (transformações).





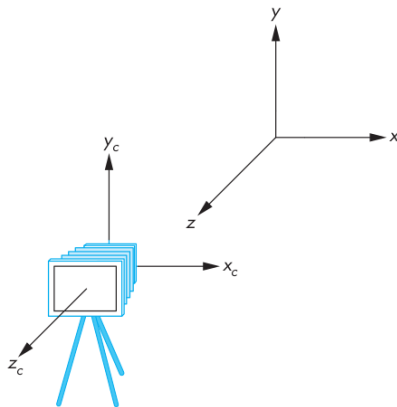
# Viewing Transformation

- *camera transformation* ou *eye transformation* : modela a transformação do sistema de coordenadas do mundo para o sistema de coordenadas da câmera. É uma transformação de corpo rígido. Depende apenas da posição e orientação (pose) da câmera.
- *projection transformation* : modela a transformação das coordenadas 3D para coordenadas de imagens 2D para um sistema padrão. Depende apenas do tipo de projeção adotado.
- *viewport transformation* ou *windowing transformation* : transforma a representação 2D padronizada para uma representação da imagem digital. Leva em consideração o dispositivo, o tamanho e posição da imagem gerada.

# Viewing Transformation

## Camera transformation

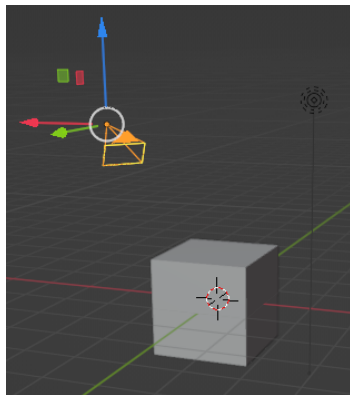
- A câmera pode ser entendida como um objeto inserido na cena, possuindo portanto localização e orientação no mundo virtual.



# Viewing Transformation

## Camera transformation

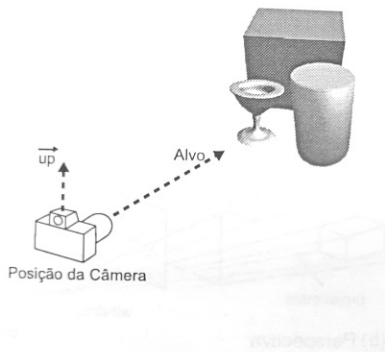
- A câmera possui parâmetros extrínsecos que a descrevem no *world space*:
  - A posição  $p_c = (x_c, y_c, z_c)$ , também chamado de *eye position*
  - três vetores que definem sua orientação



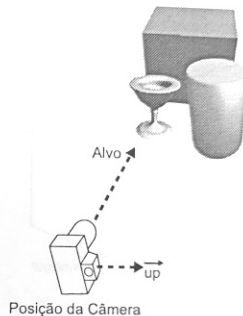
# Viewing Transformation

## Camera transformation

- Além disso, precisamos de dois vetores:
  - um vetor  $\vec{n}$  que determina a direção normal (perpendicular) ao plano onde será formada a imagem (alvo ou *gaze direction*),
  - um vetor normal  $\vec{up}$  que indica a direção a ser usada como direção vertical da imagem (*view-up direction*).



(a) Câmera na orientação normal

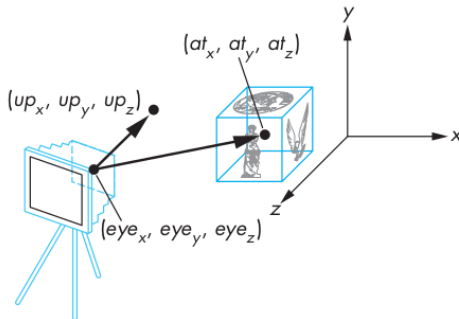


(b) Câmera inclinada para a direita

# Viewing Transformation

## Camera transformation

- Definir os parâmetros de maneira mais intuitiva:
  - uma posição  $eye = (e_x, e_y, e_z)$ , dita como posição do observador (centro da câmera);
  - uma posição que define para onde a câmera está olhando, um alvo  $at = (a_x, a_y, a_z)$ ;
  - um vetor que indica a “direção para cima” de uma cena 3D, ou seja, o vetor  $\vec{up}$



# Viewing Transformation

## Camera transformation

- Seja  $\vec{n}$  o vetor normal que aponta para a posição  $at = (a_x, a_y, a_z)$ .
- É ideal que  $\vec{up}$  seja perpendicular a  $\vec{n}$  para evitar distorções indesejadas e garantir uma visualização mais natural.
- É possível definirmos um sistema de coordenadas para a câmera, tendo como origem a posição  $eye = (e_x, e_y, e_z)$  e os dois vetores.

$$\vec{u} = \vec{up} \times \vec{n}$$

# Viewing Transformation

## Camera transformation

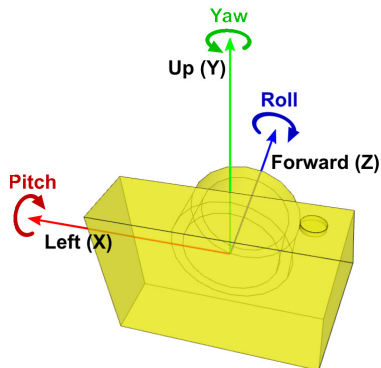
- Podemos definir a transformação entre a representação de um ponto qualquer  $p$  do sistema de coordenadas do mundo para sua representação  $p'$  no sistema de coordenadas da câmera como:
  - operação de transladar a origem para posição da câmera
  - seguida de uma rotação para reorientação dos eixos para o novo sistema.

$$V = RT = \begin{bmatrix} u_x & u_y & u_z & 0 \\ up_x & up_y & up_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -e_x \\ 0 & 1 & 0 & -e_y \\ 0 & 0 & 1 & -e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

# Viewing Transformation

## Camera transformation

- Tipos de rotações da câmera

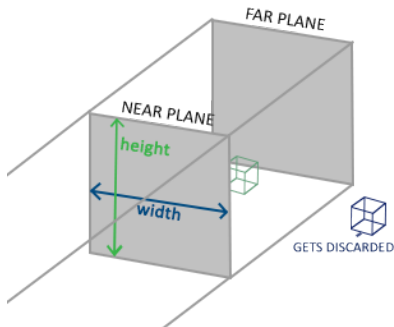




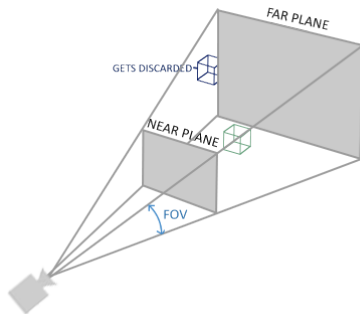
# Viewing Transformation

## Projection transformation

- O tipo de projeção usado controla como a cena é mostrada no plano de imagem.



Projeção ortográfica



Projeção perspectiva (ou cônica)

# Viewing Transformation

## Projection transformation

- Após os objetos serem representados no sistema de coordenadas da câmera, a próxima tarefa consiste em transformar as coordenadas 3D para coordenadas 2D por meio da projeção.
- Assume-se que nesta etapa a projeção é realizada para um sistema padrão (não se preocupa com o dispositivo).
- Durante a projeção pode ocorrer sobreposição, além disso a imagem digital é limitada.
- Não é possível produzir uma imagem de todo o plano de imageamento (infinito).

# Viewing Transformation

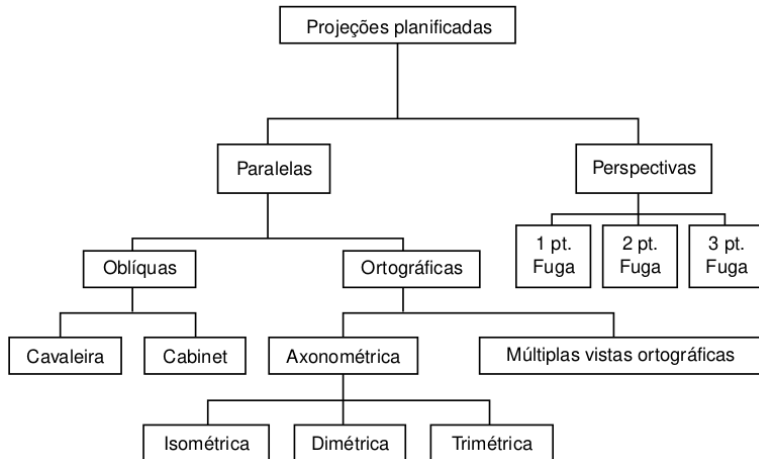
## Projection transformation

- É comum a definição de um **volume de visualização** (*viewing box*) que restringe quais objetos de fato serão projetados sobre o plano da imagem.
- A forma do volume de visualização depende do tipo de projeção e das configurações da câmera.
- As projeções geométricas são classificadas de acordo com as relações entre o centro de projeção (COP), o plano de projeção e as direções das linhas ou raios de projeção.

# Viewing Transformation

## Projection transformation

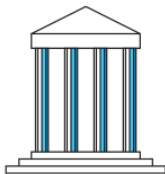
- Classificações das projeções geométricas



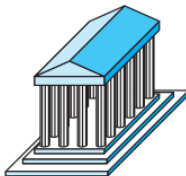
# Viewing Transformation

## Projection transformation

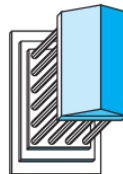
- Projeções clássicas



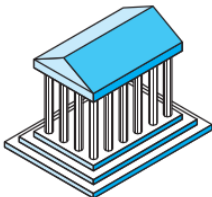
Front elevation



Elevation oblique



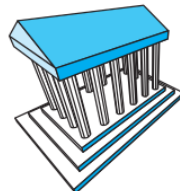
Plan oblique



Isometric



One-point perspective

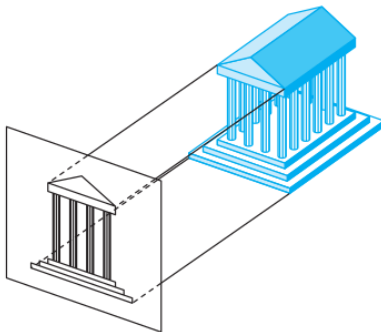


Three-point perspective

# Viewing Transformation

## Projection transformation

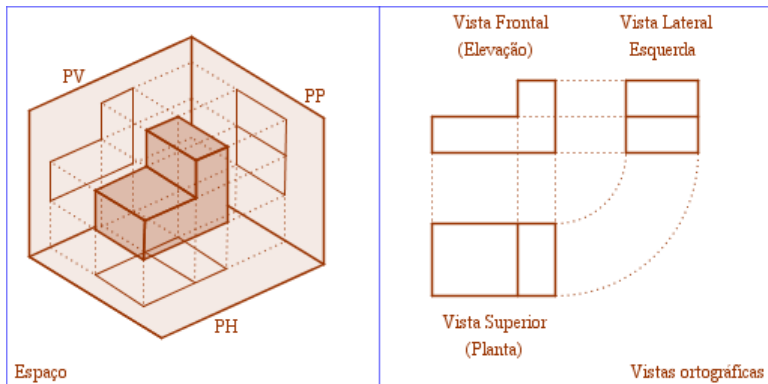
- **Projeção ortográficas:** é um tipo de projeção paralela em que as linhas de projeção são ortogonais ao plano de projeção.
- Nas projeções paralelas o centro de projeção é localizado no infinito e todas as linhas de projeção são paralelas entre si.
  - as linhas paralelas à própria direção de projeção são transformadas em pontos.



# Viewing Transformation

## Projection transformation

- Diferentes planos de projeção produzidos utilizando projeções ortográficas.



Fonte: <https://ifcequixada.files.wordpress.com/2016/04/vistas-ortogracc81ficas.png>

# Viewing Transformation

## Projection transformation

- Considerando as projeções ortogonais tal que o vetor normal ao plano de projeção esteja alinhado ao eixo-z do sistema de coordenadas da câmera temos que: as coordenadas  $x$  e  $y$  permanecem inalteradas e a coordenada  $z$  é desprezada.
- Neste caso, a projeção paralela ortogonal é representada pela matriz:

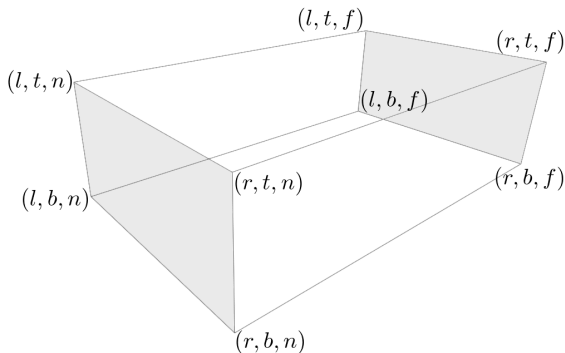
$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



# Viewing Transformation

## Projection transformation

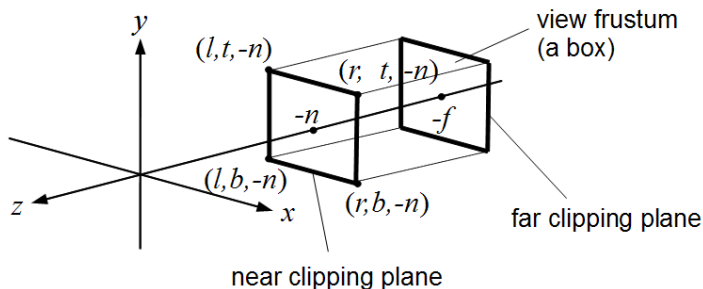
- O volume de visualização em projeções ortográficas é um paralelepípedo de ângulos retos.



# Viewing Transformation

## Projection transformation

- São definidos seis planos de referência no sistema de coordenadas da câmera: esquerdo ( $l$ ), direito( $r$ ), superior( $t$ ), inferior( $b$ ) além de um par de distâncias para limitar o eixo de profundidade (próximo( $n$ ) e longe( $f$ )).



Fonte: [https://en.wikipedia.org/wiki/Viewing\\_frustum#/media/File:Orthographic\\_view\\_frustum.png](https://en.wikipedia.org/wiki/Viewing_frustum#/media/File:Orthographic_view_frustum.png)

# Viewing Transformation

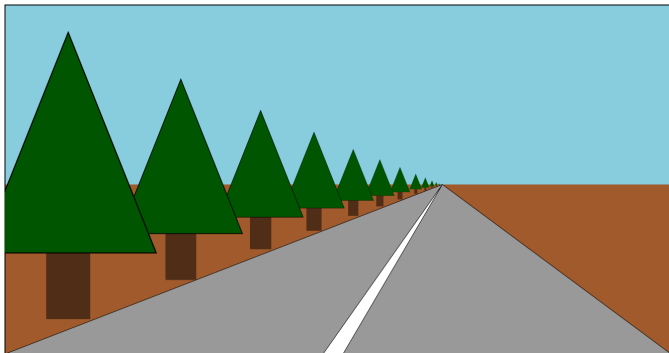
## Projection transformation

- A projeção ortográfica é mais simples, pois ela simplesmente despreza a coordenada  $z$  do espaço 3D.
- A **projeção perspectiva** é mais realista do que a projeção ortográfica, pois imita como as imagens são formadas na retina do olho por raios de luz viajando em direção a um ponto, ou seja, é o *foreshortening* que nos leva a ter a noção de distância de um objeto.

# Viewing Transformation

## Projection transformation

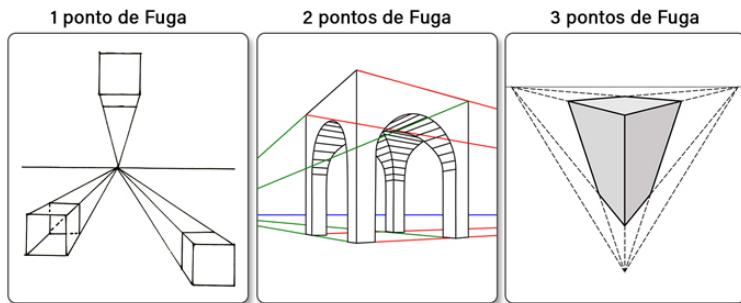
- Na **projeção perspectiva**, os objetos mais distantes ficam menores. As linhas paralelas não são paralelas, mas convergem para um único ponto.



# Viewing Transformation

## Projection transformation

- Os desenhos em perspectiva são caracterizados pela mudança do comprimento e pelos **pontos de fuga**.
- A mudança de comprimento é uma ilusão que mostra objetos cada vez menores à medida que sua distância do centro de projeção aumenta.
- Pontos de fuga são uma ilusão de que conjuntos de linhas paralelas convergem para um ponto, denominado de *fuga*.



# Viewing Transformation

## Projection transformation

- Duas descrições são utilizadas para definir o volume de visão (*frustum*):
  - seis valores: esquerdo, direito, superior, inferior, longe e perto (são planos inclinados)
  - quatro valores: ângulo de visualização ou *fov* ( $\theta$ ), *aspect ratio*, longe e perto

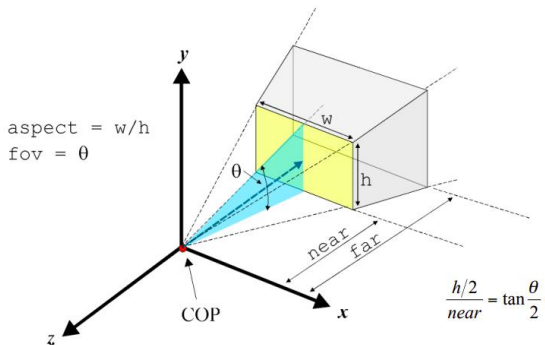
Saiba mais!

[Frustum - Wikipedia](#)

# Viewing Transformation

## Projection transformation

- Utilizamos *frustum* simétrico (sistema simplificado de visão):
  - linha perpendicular que liga o centro de projeção (COP) ao meio do plano de visão
  - O centro de projeção está na origem
  - o plano de visão está sobre o plano de recorte
  - a direção de observação é paralela a  $-z$



# Viewing Transformation

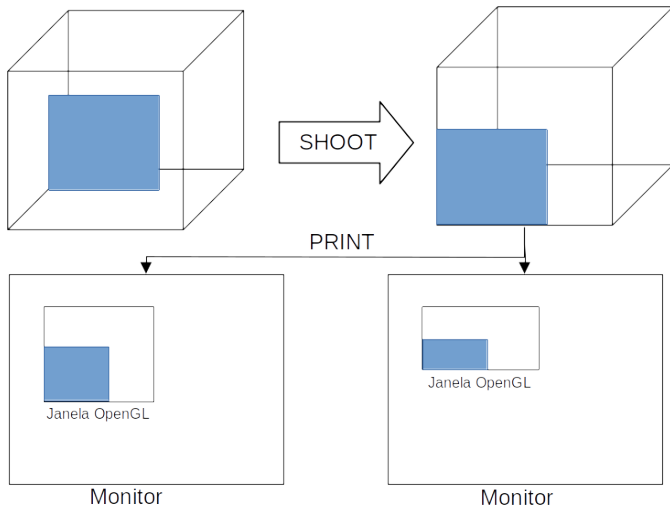
## Viewport transformation

- Consiste em mapear as coordenadas do sistema de referência normalizado (NDC) para o dispositivo em que a imagem será exibida.
- O processo de renderização ocorre em duas etapas:
  - 1 **shoot**: os objetos são projetados perpendicularmente na face frontal (visor).
  - 2 **print**: o visor é proporcionalmente redimensionado para se ajustar a tela da janela do dispositivo. Leva em consideração o *aspect ratio* (largura / altura).



# Viewing Transformation

## Viewport transformation



- Eduardo Azevedo, Aura Conci, Cristina Vasconcelos (2007) Computação Gráfica: Teoria e Prática - Geração de Imagens, Elsevier.
- WebGL2 Fundamentals. Disponível em: <https://webgl2fundamentals.org>
- Ginsburg, D.; Purnomo, B. OpenGL ES 3.0: Programming Guide. Second Edition. Addison-Wesley. 2014.
- Bailey, M.; Cunningham S. Graphics Shaders: Theory and Practice. Second Edition. CRC Press. 2012.
- Dave Shreiner, Graham Sellers, John M. Kessenich, Bill Licea-Kane. *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, Version 4.3, Addison-Wesley. 2013.
- Song Ho Ahn. OpenGL: Transformation. Disponível em: [http://www.songho.ca/opengl/gl\\_transform.html](http://www.songho.ca/opengl/gl_transform.html). Acesso em outubro de 2024.

- Dúvidas?
- Comentários?

## Contato

Gabriel Marcelino Alves  
gabriel.marcelino@ifsp.edu.br



*This work is licensed under Attribution-NonCommercial-ShareAlike 4.0 International. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>*

