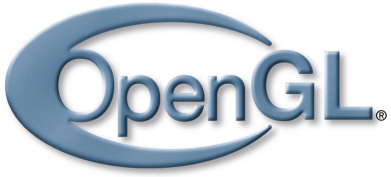
Computação Gráfica

TCC-00291

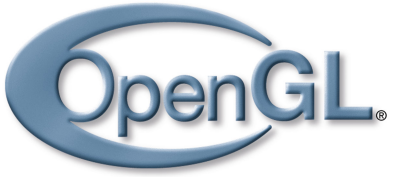
Assunto: Câmera



Introdução

Etapas do processo

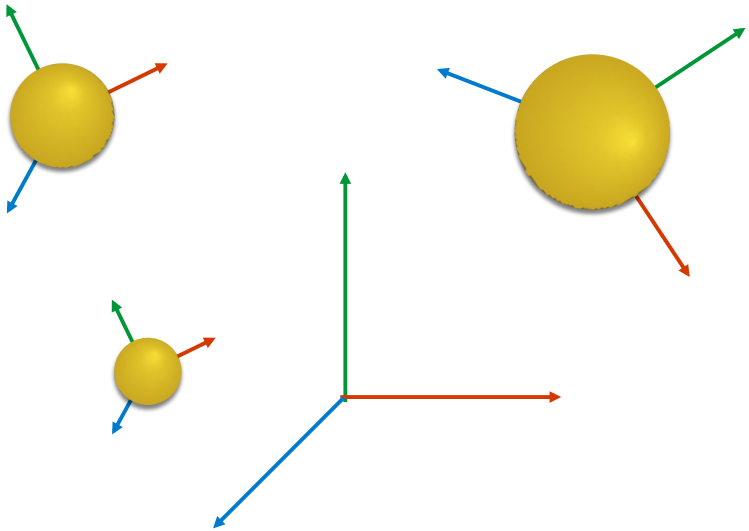
Vimos nas aulas passadas como usar transformações afins para transladar, escalar e rotacionar objetos 3D.



Introdução

Etapas do processo

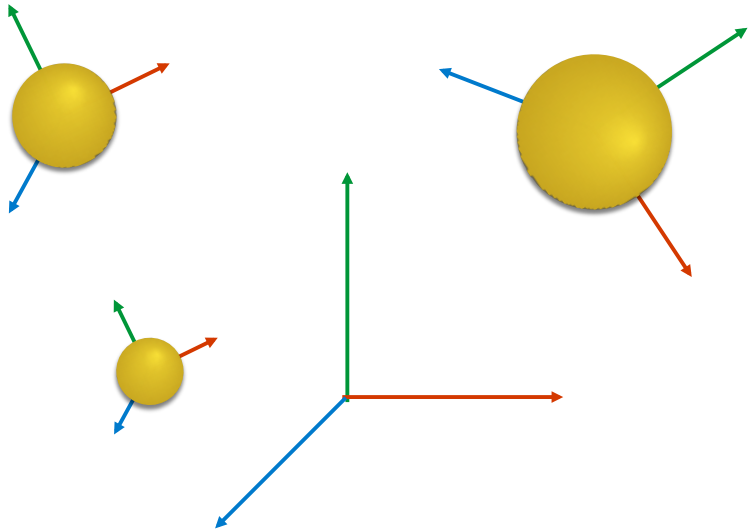
Vimos nas aulas passadas como usar transformações afins para transladar, escalar e rotacionar objetos 3D.



Introdução

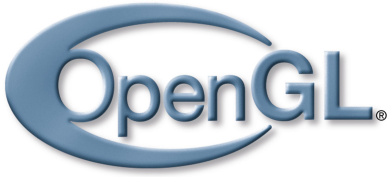
Etapas do processo

Vimos nas aulas passadas como usar transformações afins para transladar, escalar e rotacionar objetos 3D.



Na pratica, alteramos o modelo do sistema de coordenadas

do objeto para o sistema de coordenadas do mundo.



Introdução

Etapas do processo

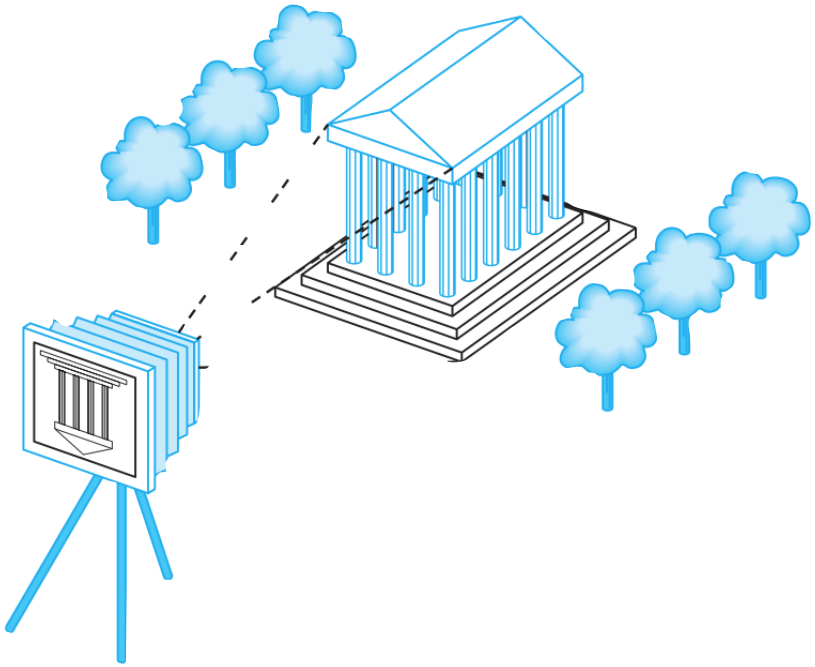
Estas transformações não são suficientes!

Ainda precisamos considerar:
Ponto de vista, tipo de projeção, tela …

Introdução

Etapas do processo

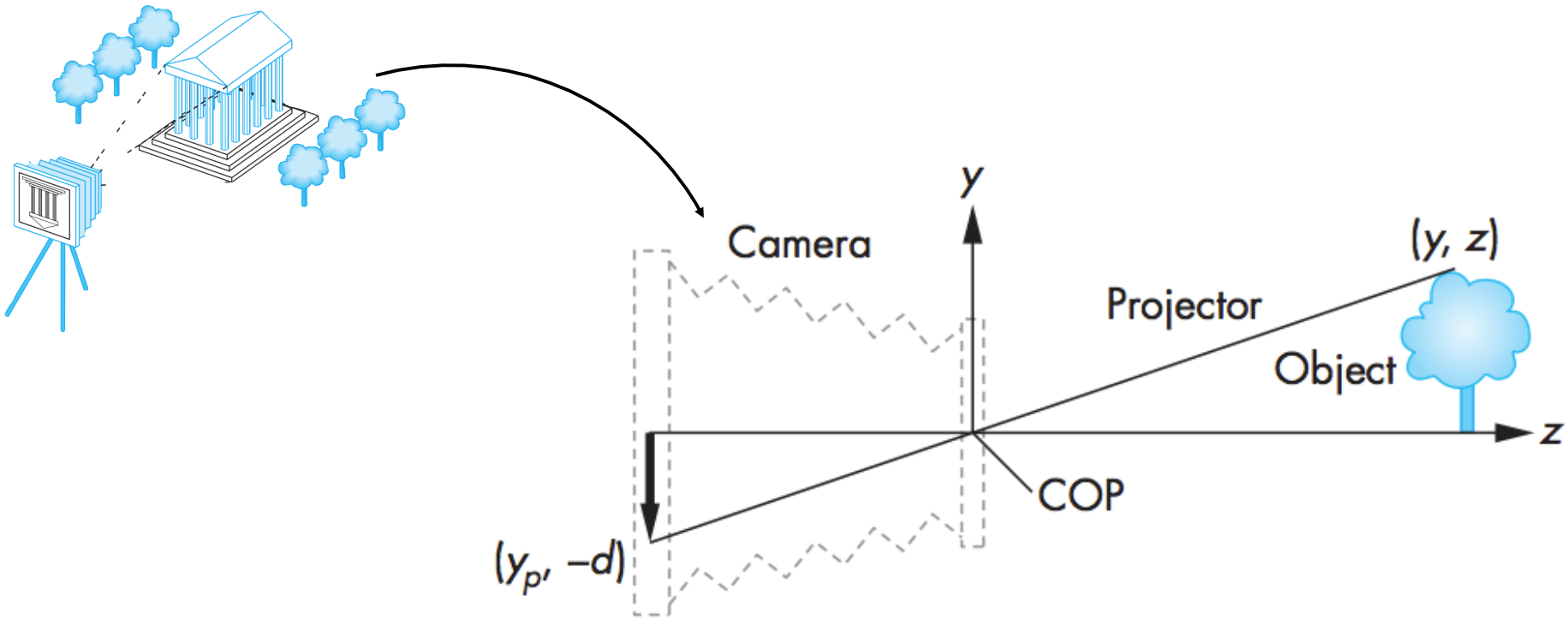
Vamos estudar um modelo de câmera.



Câmera

Pinhole

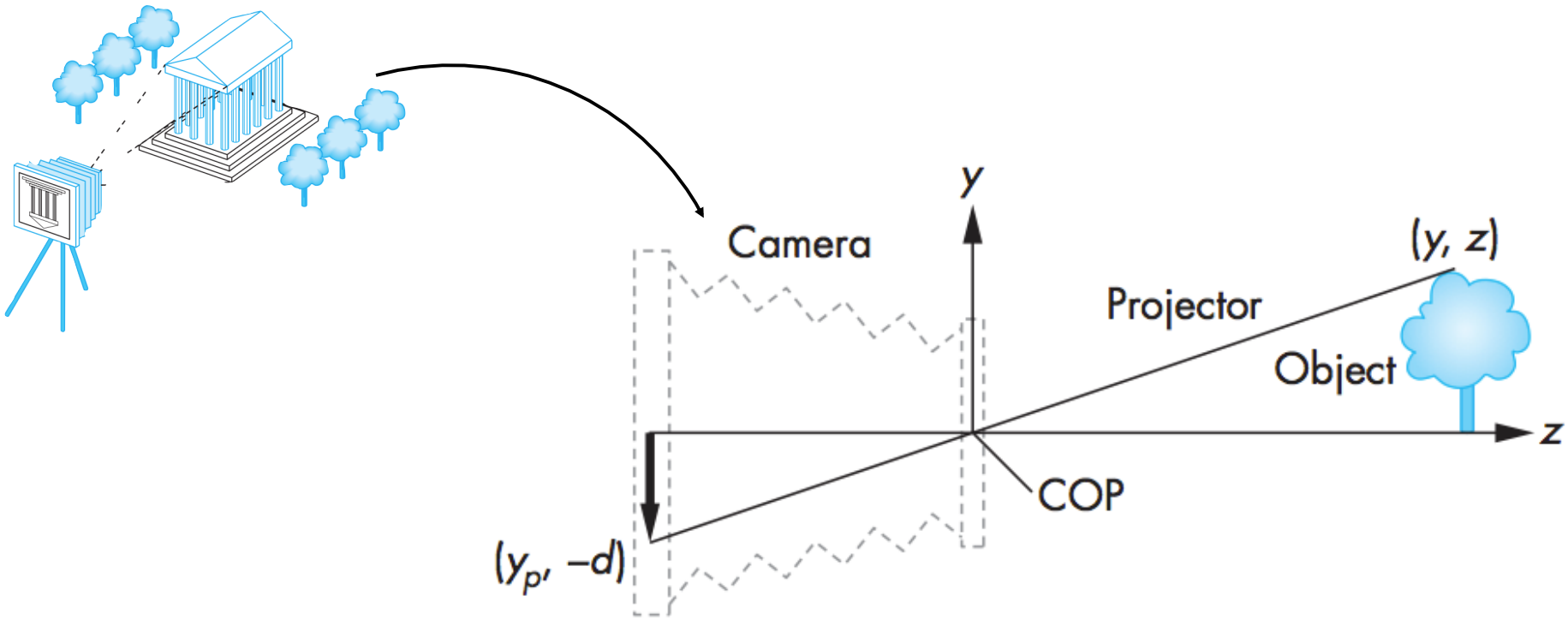
A imagem é formada no plano de projeção e o centro de projeção fica entre o objeto e o plano.



Câmera

Pinhole

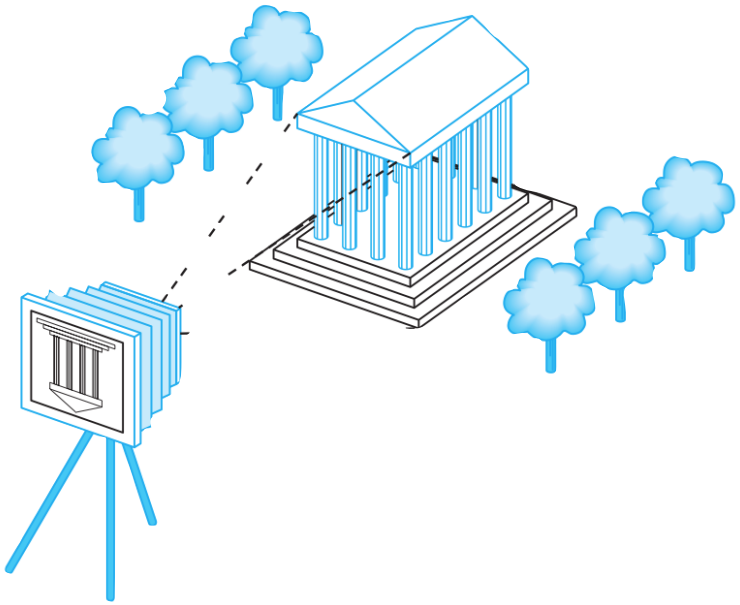
Como adaptar esta idéia para definir um modelo de câmera virtual ?



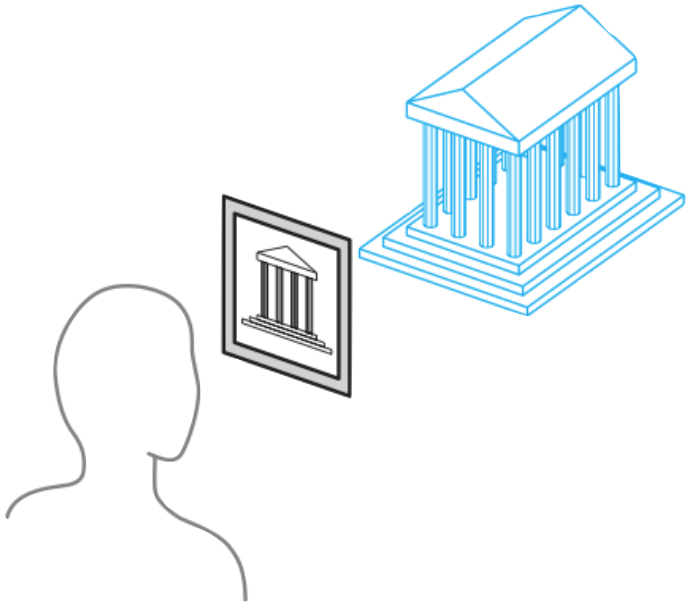
Câmera

Pinhole Vs Virtual

Como adaptar esta idéia para definir um modelo de câmera virtual ?



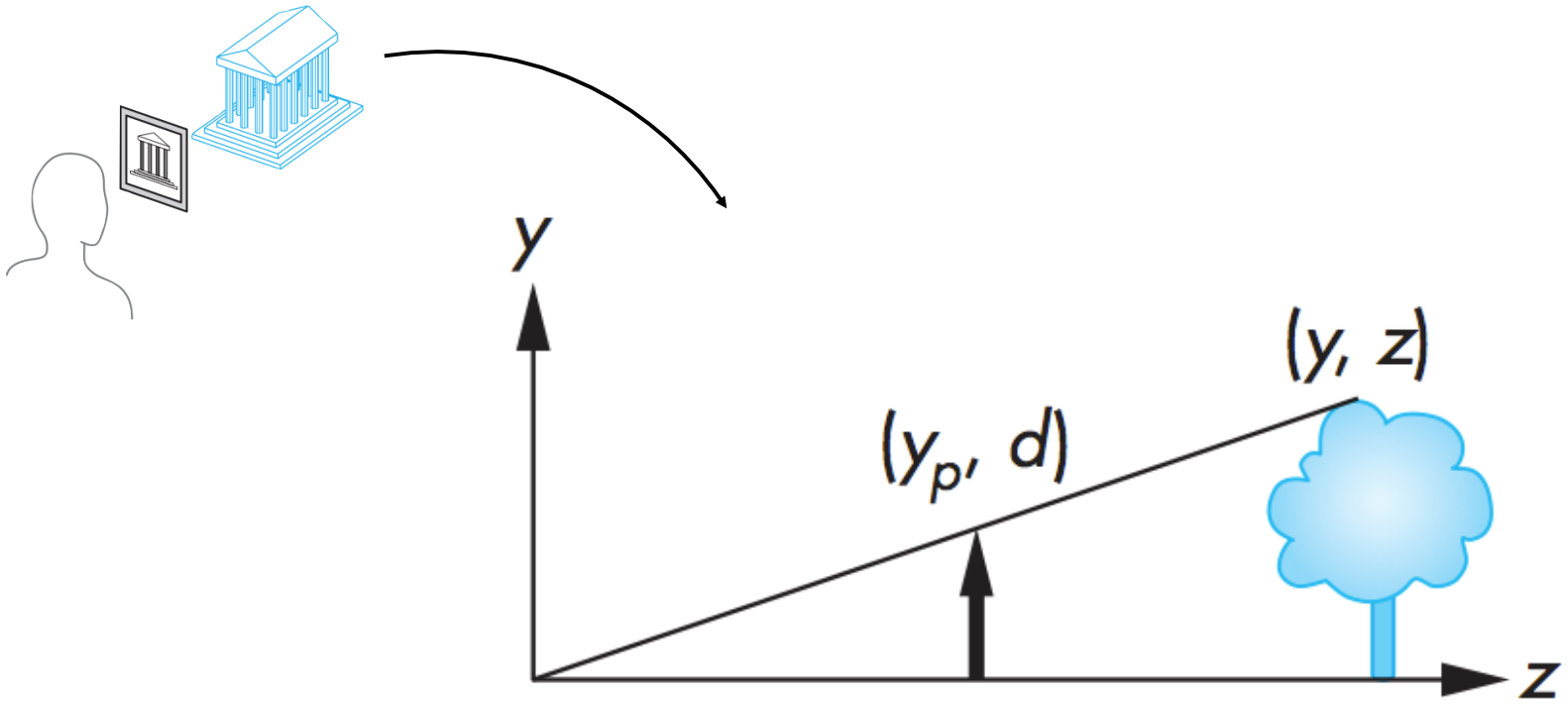
câmera pinhole



câmera virtual

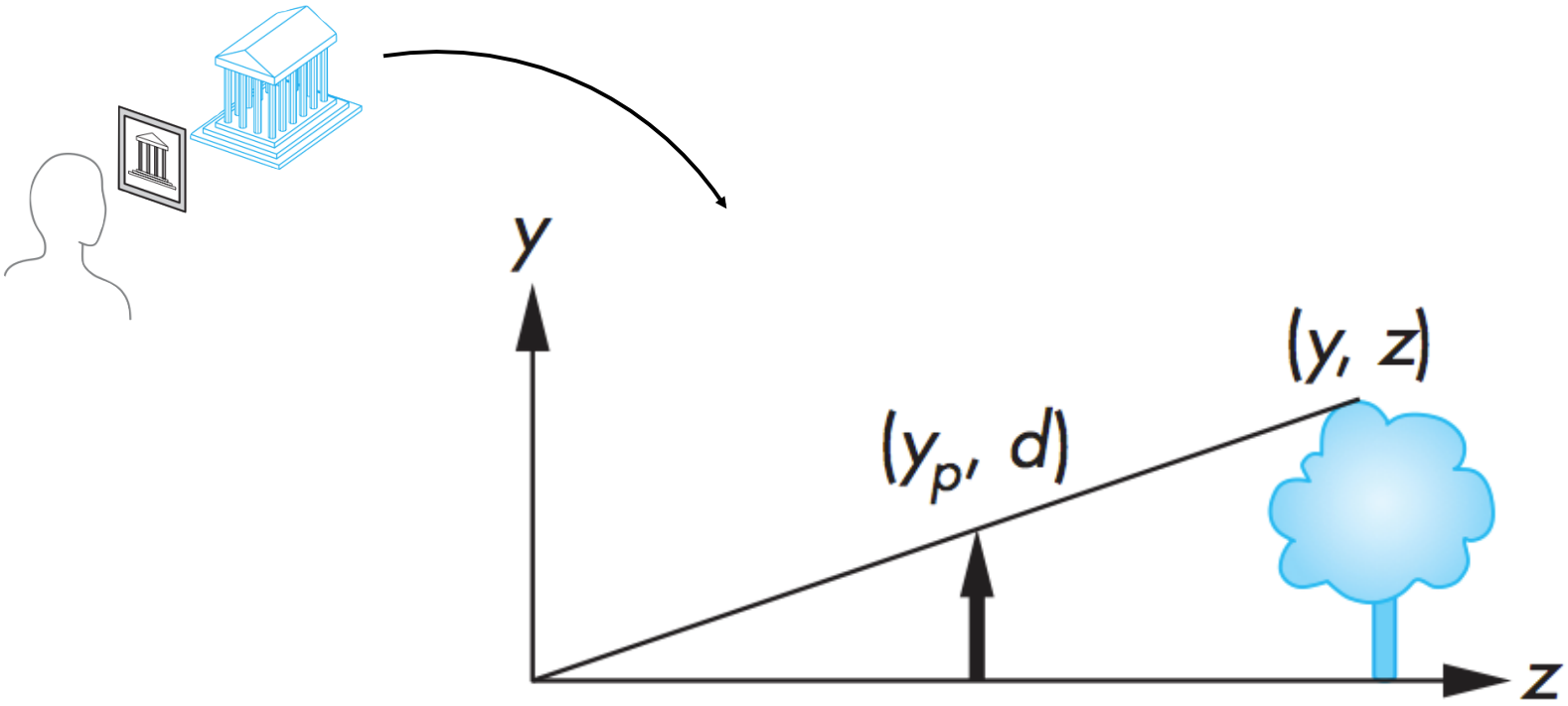
Câmera
Virtual

Movemos o plano de projeção!



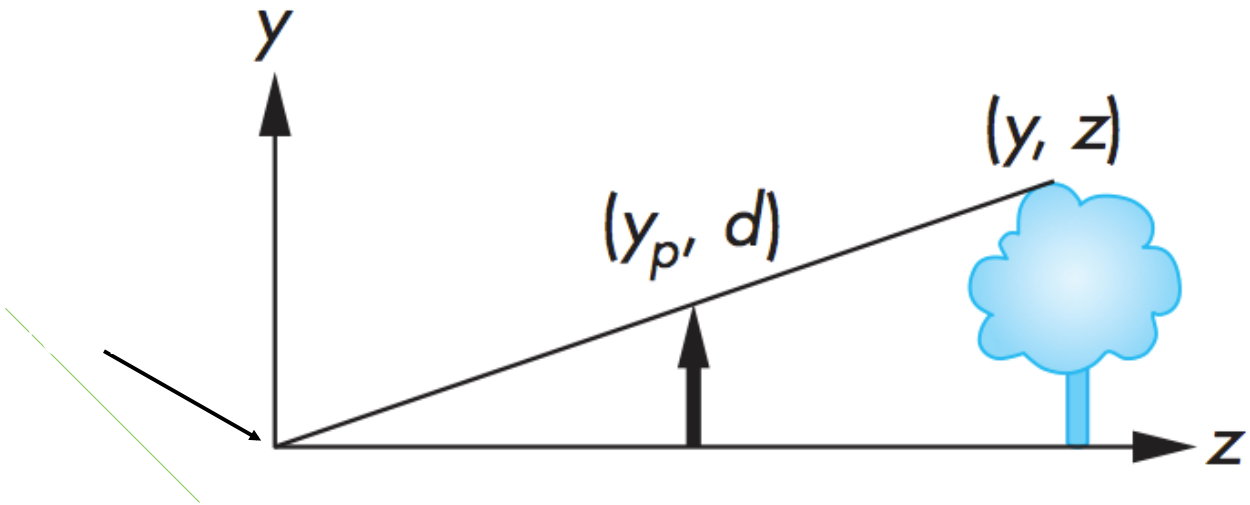
Câmera
Virtual

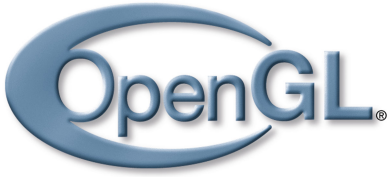
Como representar este modelo matematicamente?



Câmera
Virtual

O posicionamento da câmera é feito através de transformações geométricas que fazem da câmera a origem do sistema de coordenadas…





Câmera
Virtual

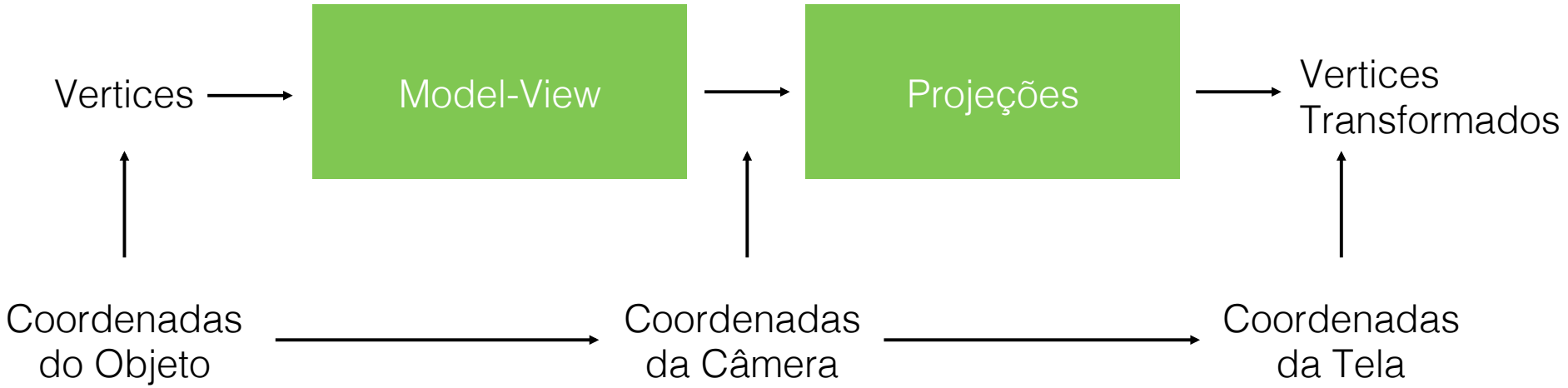
Quando concatenada com as transformações geométricas vistas anteriormente, dá origem a matriz Model-View.

Câmera

Sistemas de coordenadas

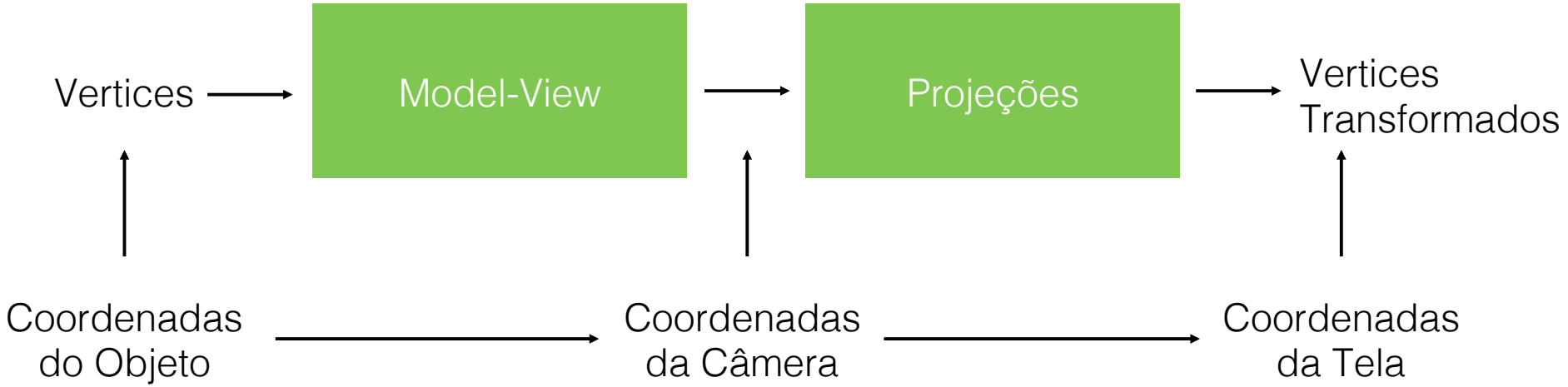
Quando concatenada com as transformações geométricas vistas anteriormente, dá origem a matriz Model-View.

O pipeline de transformações é composto por:



Câmera

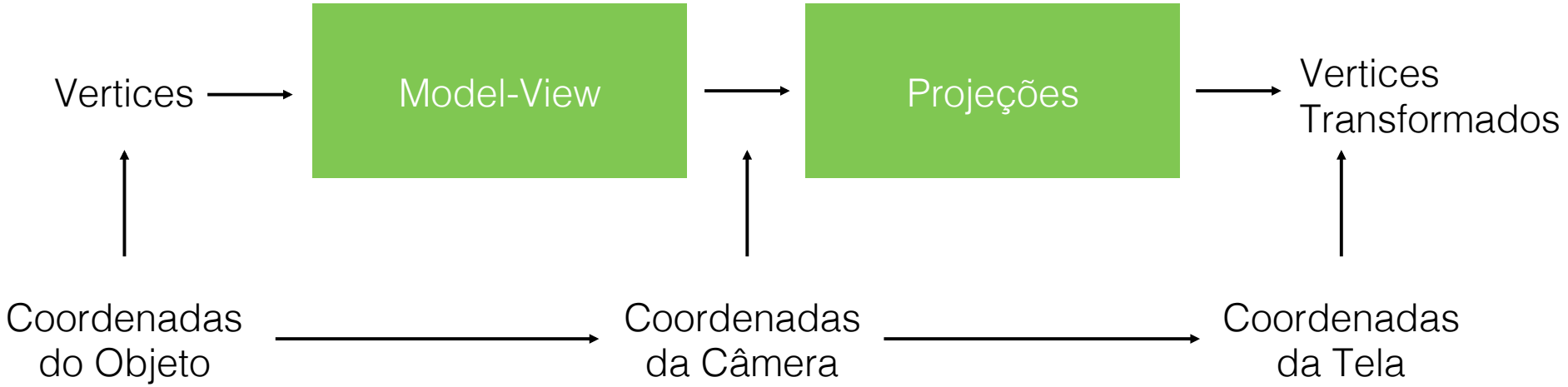
Sistemas de coordenadas



Na pratica, raramente trabalhamos nas coordenadas do mundo. Os sistemas de coordenadas mais importantes são os sistemas do objeto e da câmera.

Câmera

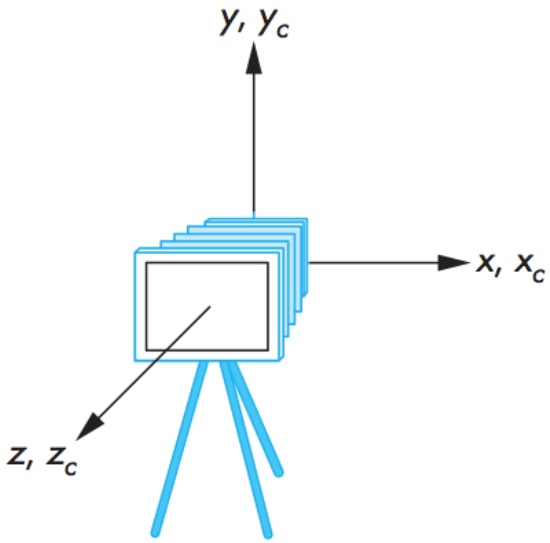
Sistemas de coordenadas



Vamos definir a matriz Model-View a partir de parâmetros que definem a câmera.

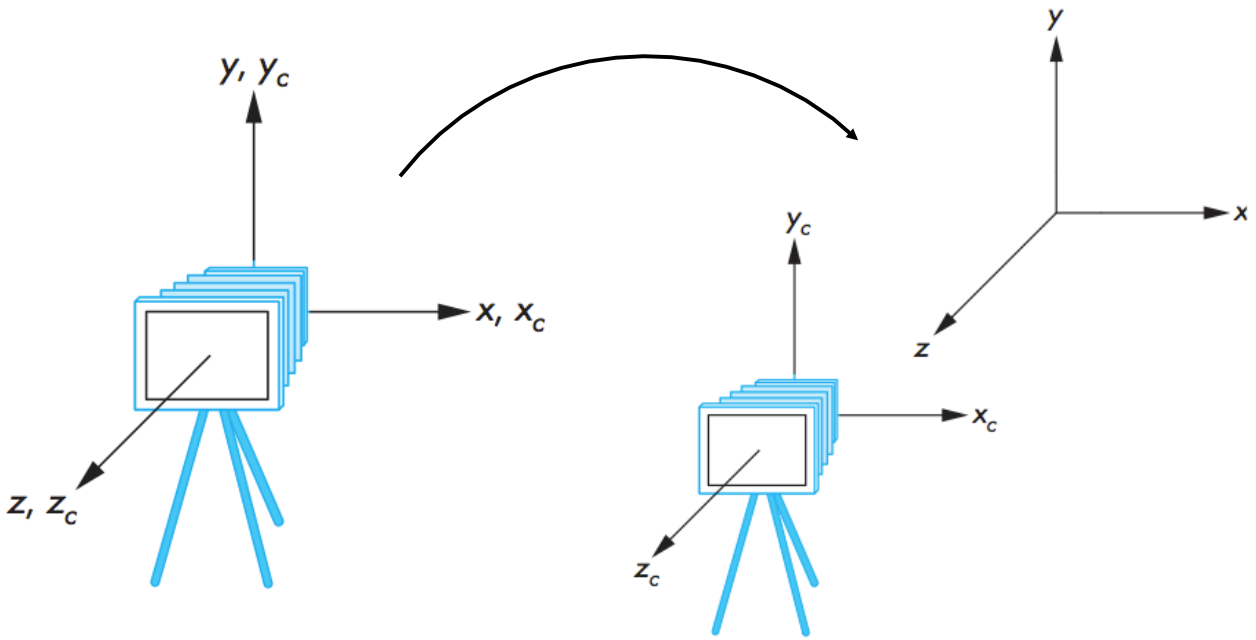
Câmera

Parâmetros



Câmera

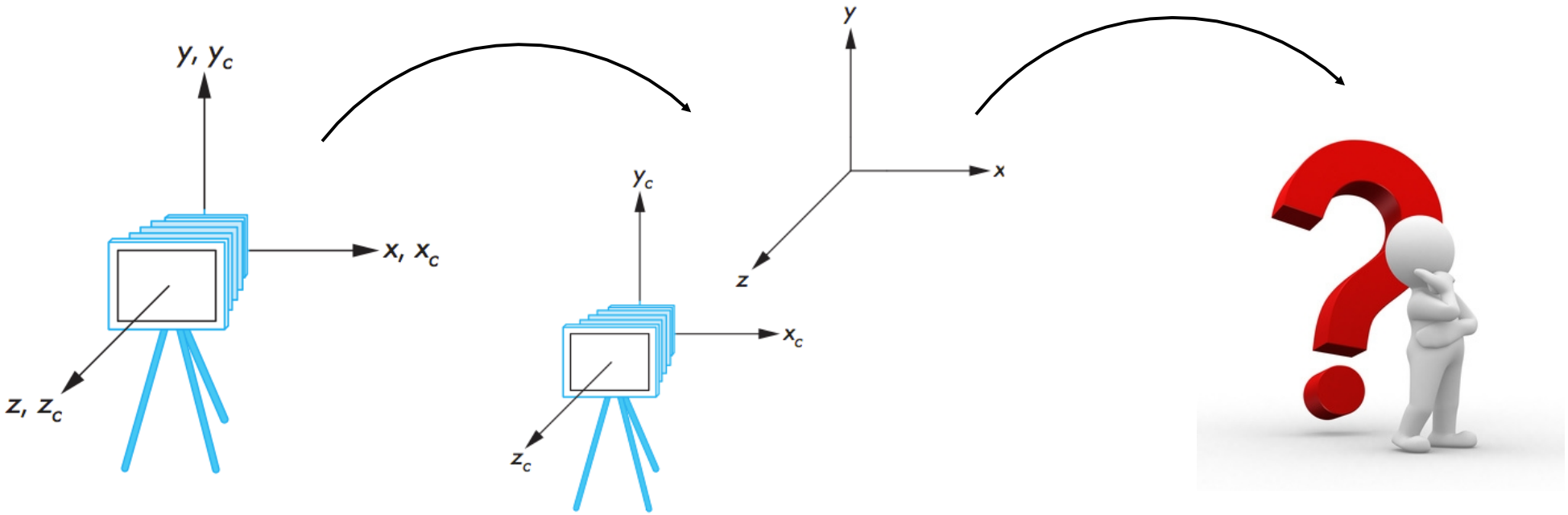
Parâmetros



Posição da câmera
(óbvio ;) …)

Câmera

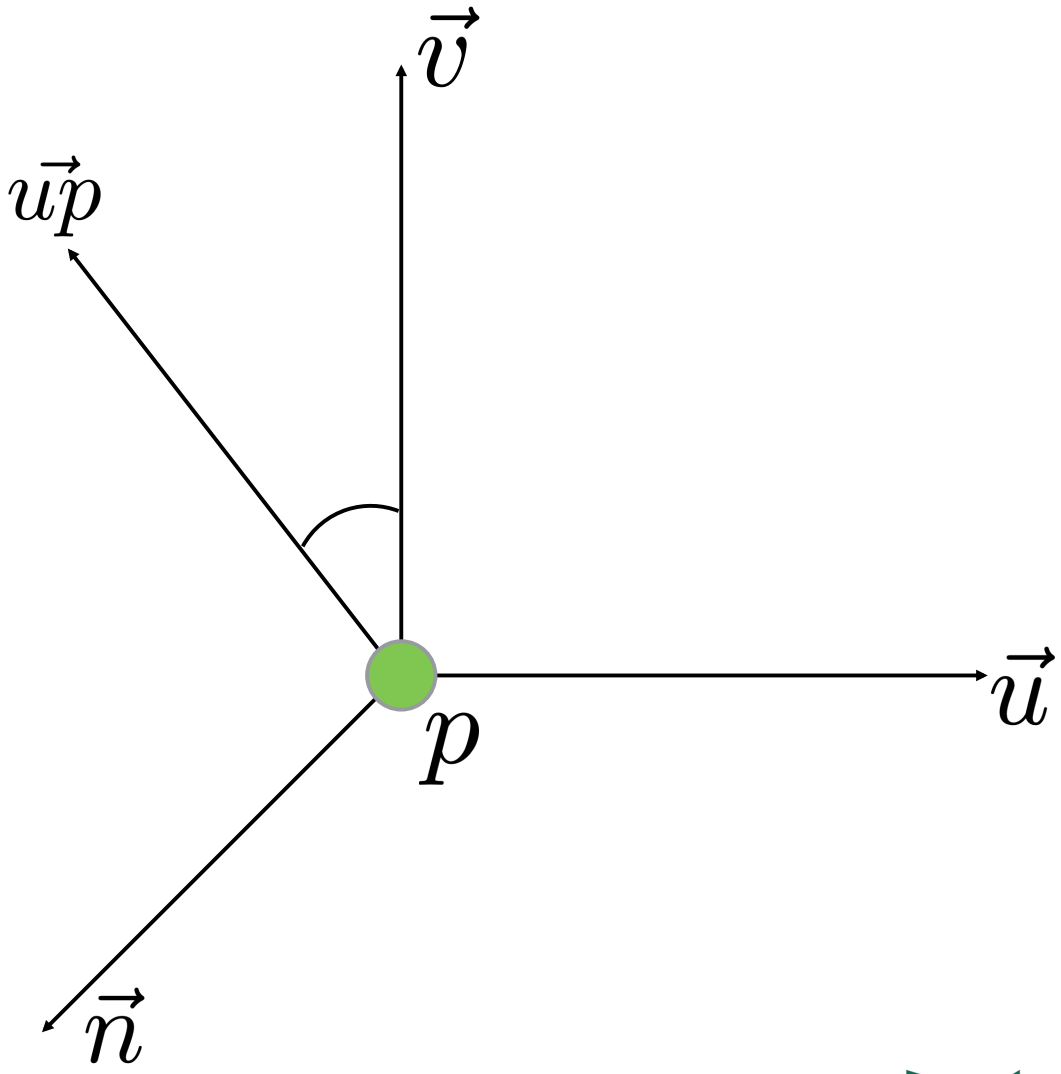
Parâmetros



Quais os outros parâmetros?

Câmera

Parâmetros

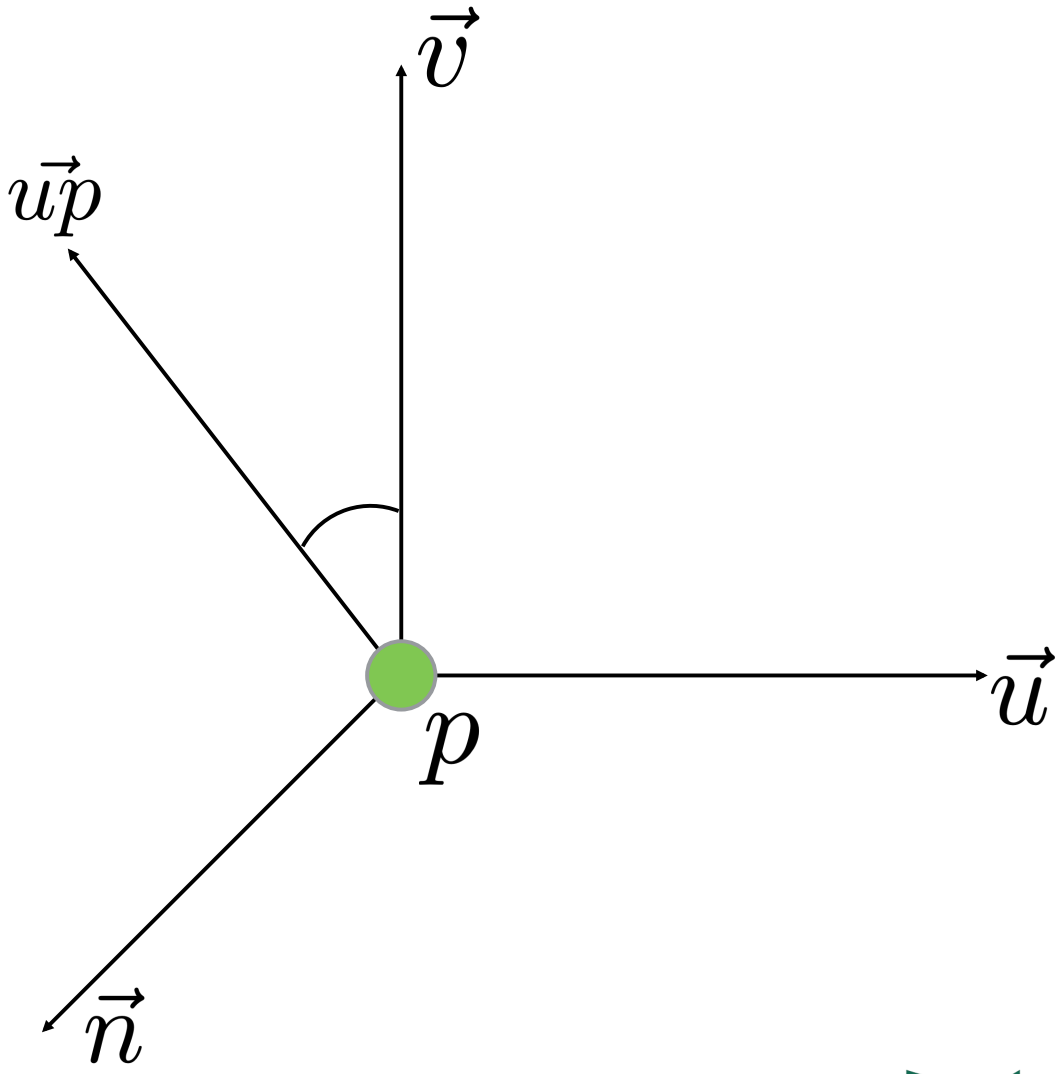


p

Posição da Câmera

Câmera

Parâmetros



p

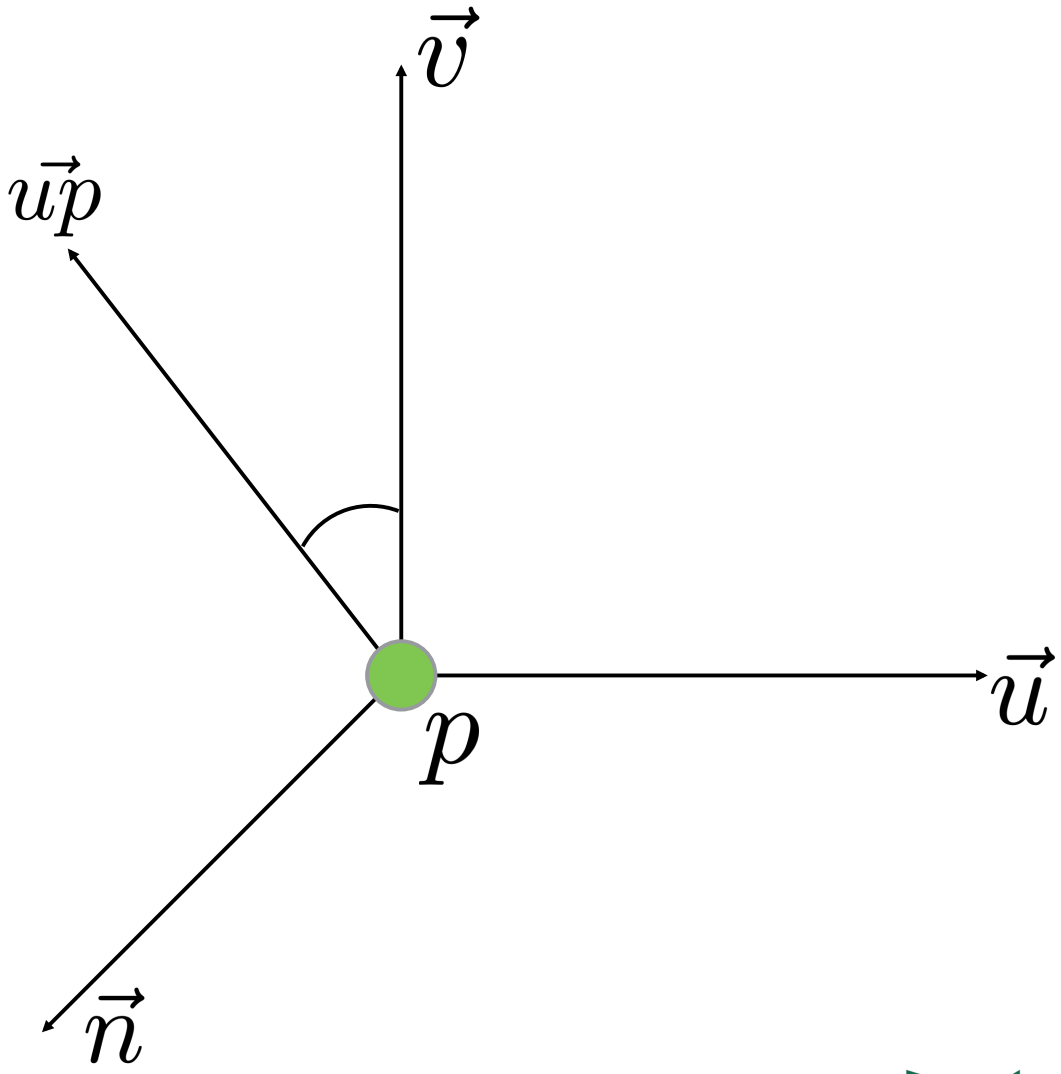
Posição da Câmera

$\vec{n}$

Direção normal ao plano de projeção

Câmera

Parâmetros



p

Posição da Câmera

$\vec{n}$

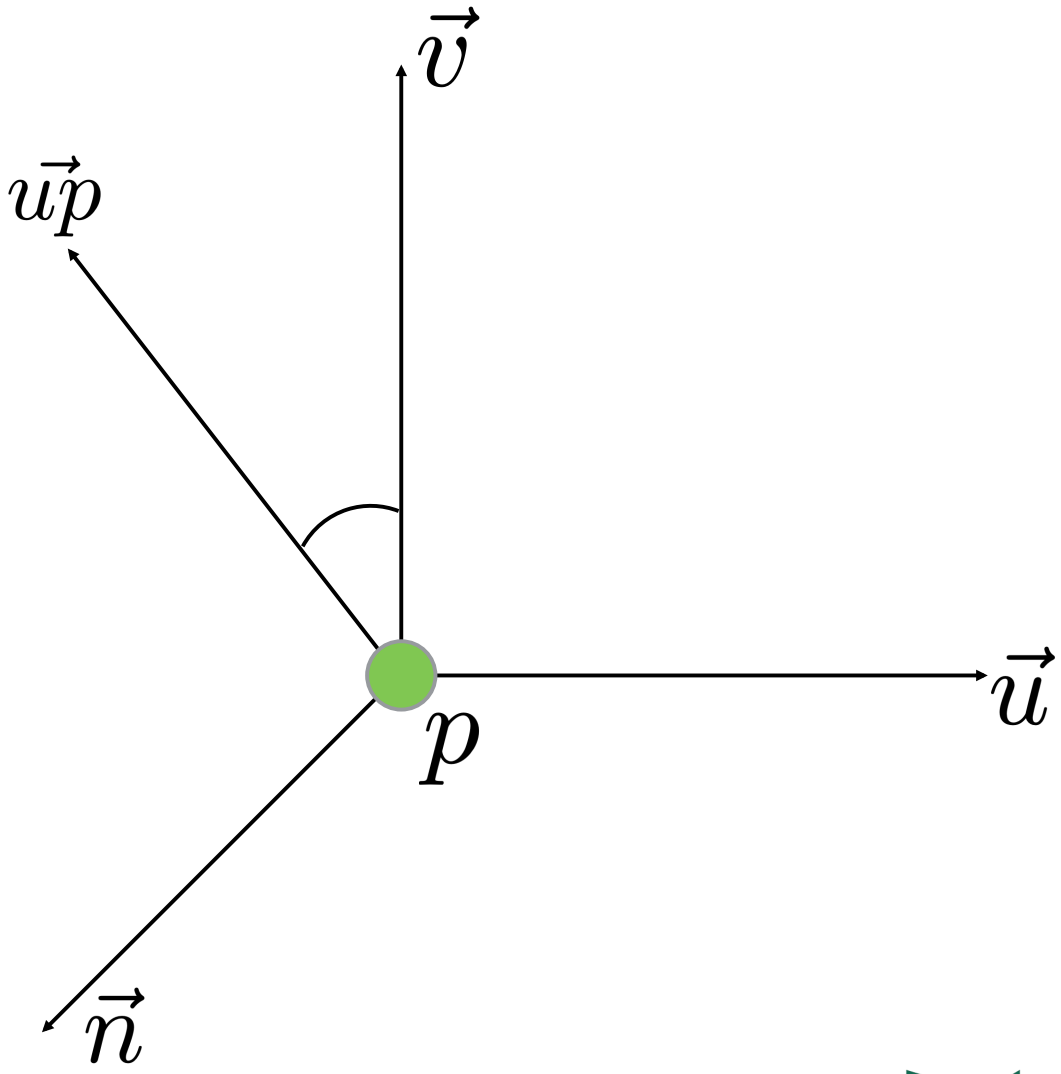
Direção normal ao plano de projeção

$\vec{up}$

Orientação da câmera

Câmera

Parâmetros



$$\vec{v}$$

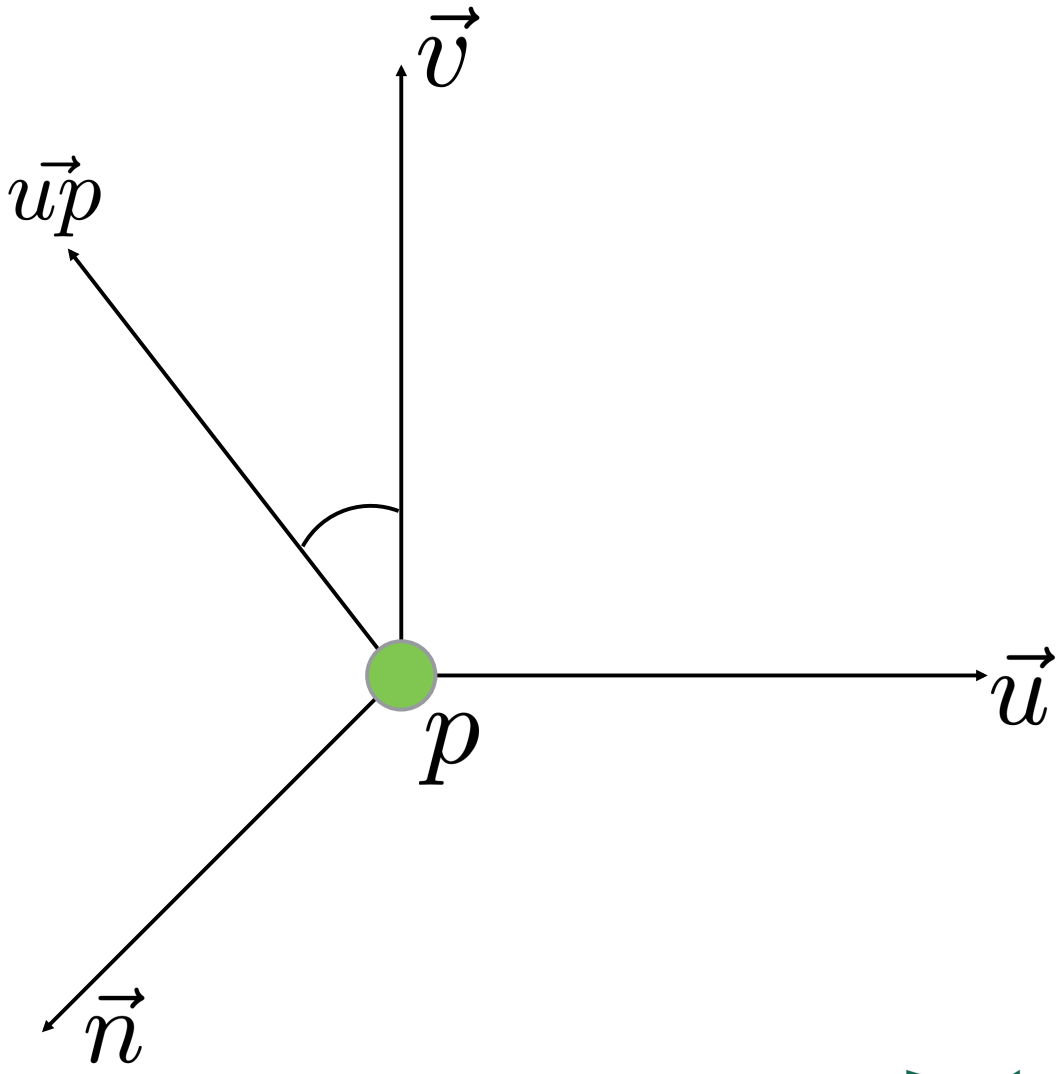
Vetor ortogonal a n.

$$\vec{v} \cdot \vec{n} = 0$$

$$\vec{v} = \vec{up} - \frac{\vec{up} \cdot \vec{n}}{\vec{n} \cdot \vec{n}}$$

Câmera

Parâmetros



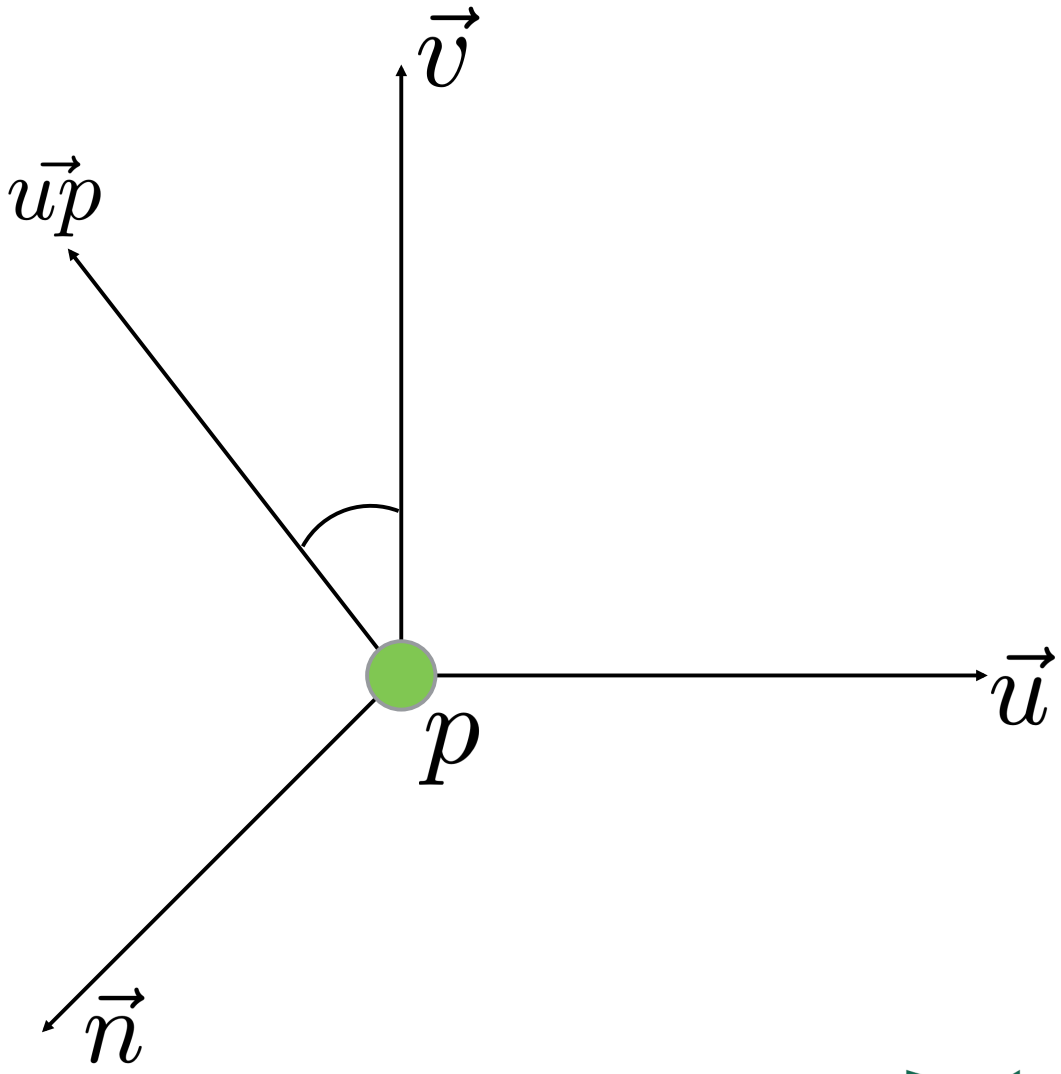
$\vec{u}$

Vetor ortogonal a n e v.

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

Câmera

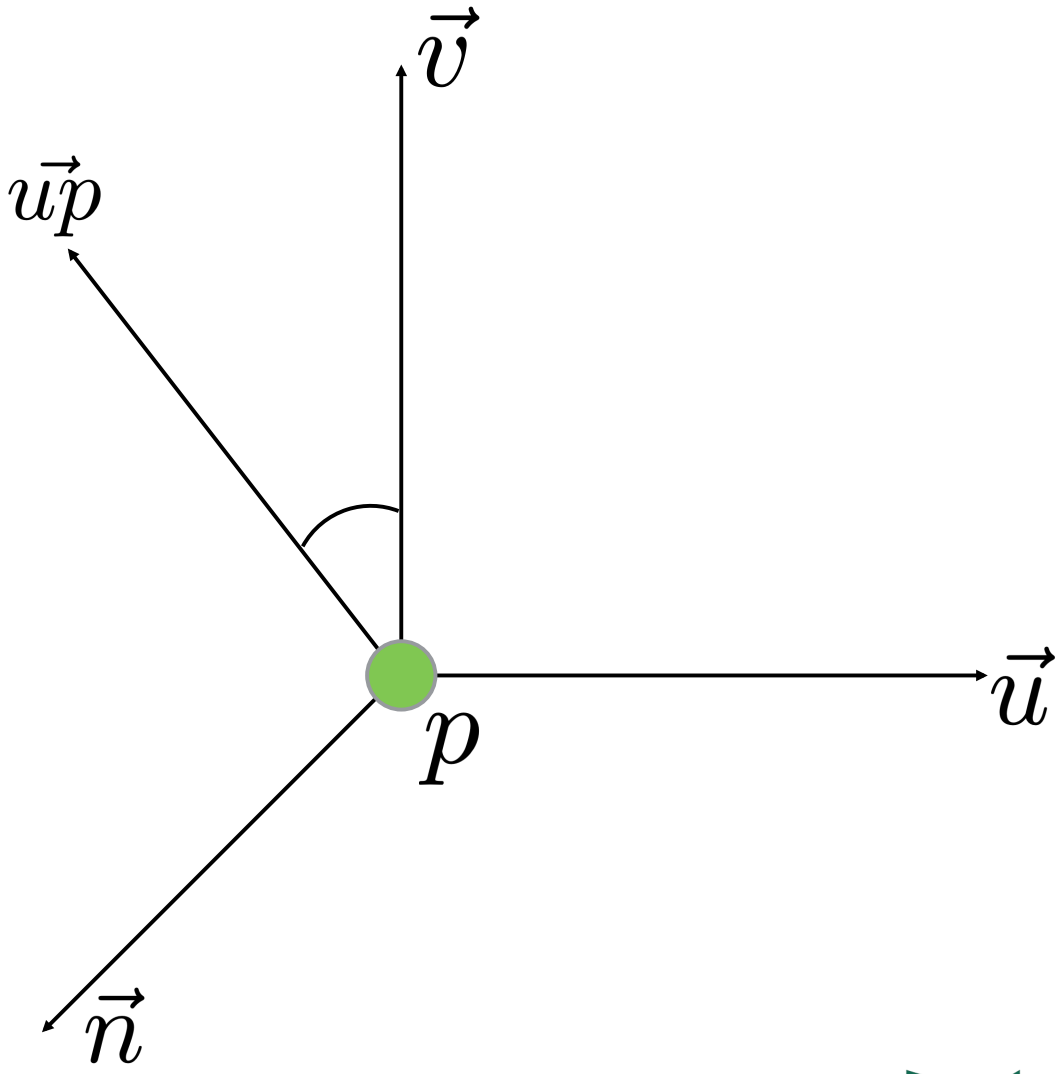
Parâmetros



Na prática, u, v e n devem ser normalizados

Câmera

Parâmetros



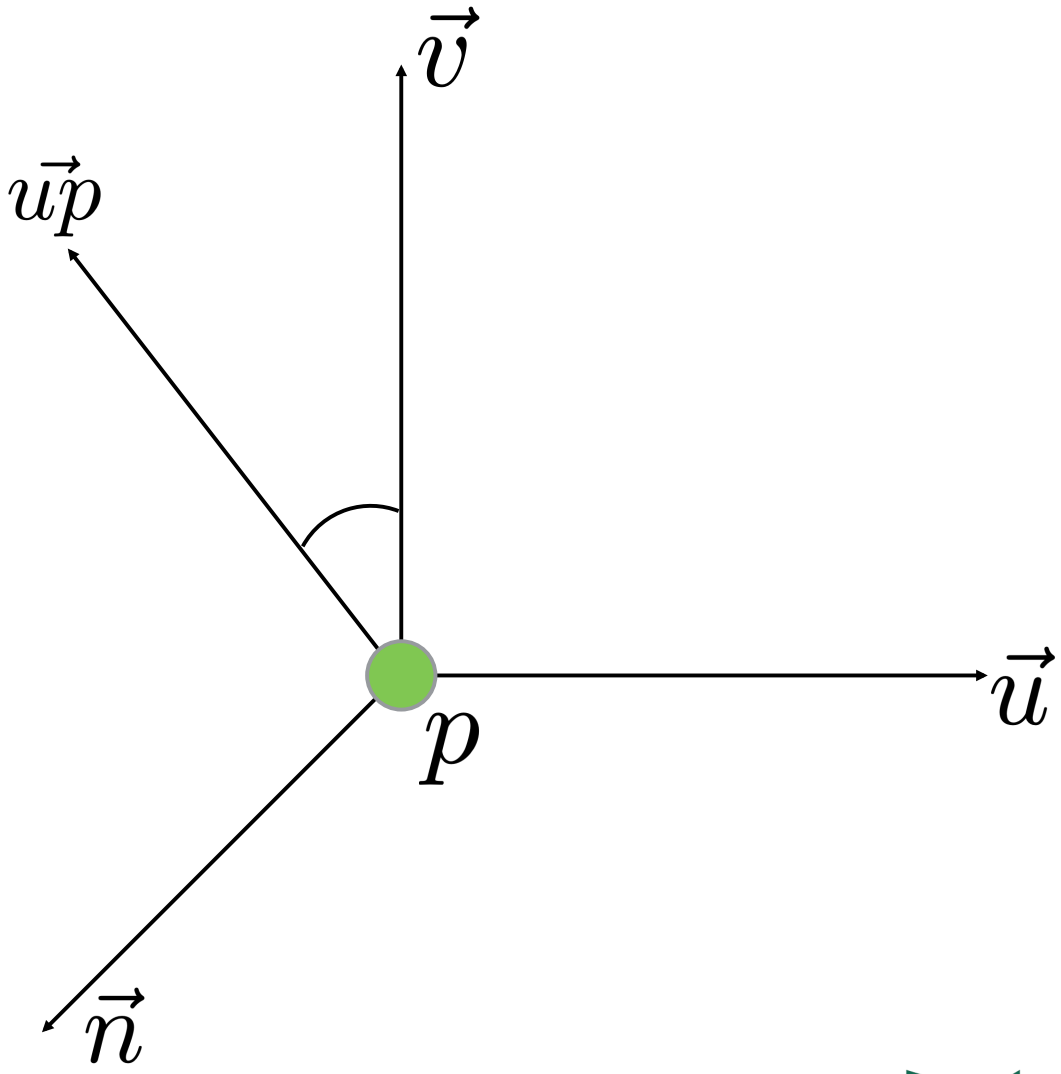
Na prática, u, v e n devem ser normalizados

Os vetores normalizados formam uma matriz de rotação

$$\mathbf{A} = \begin{bmatrix} u'_x & v'_x & n'_x & 0 \\ u'_y & v'_y & n'_y & 0 \\ u'_z & v'_z & n'_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Câmera

Parâmetros



Denote:

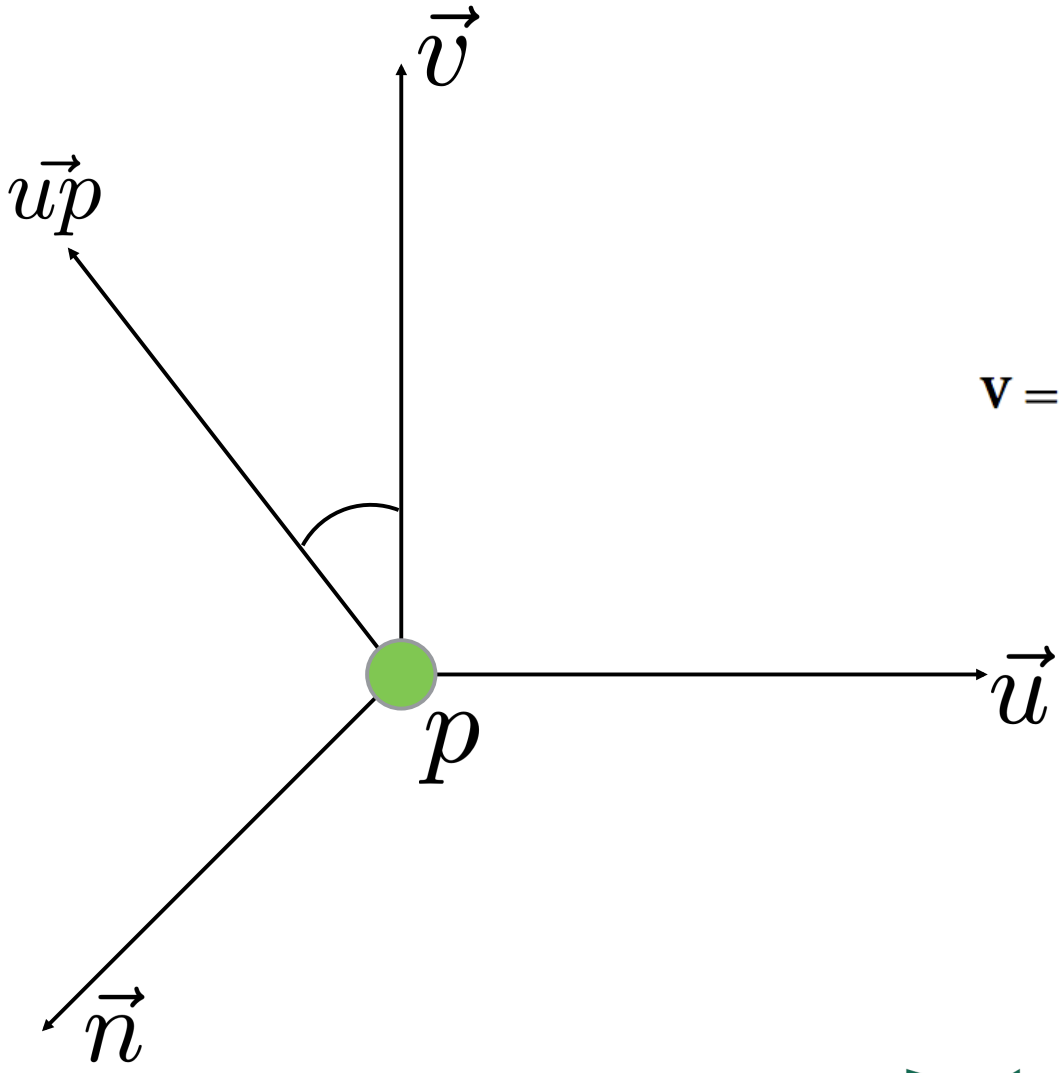
$$\mathbf{R} = \mathbf{A}^{-1} = \mathbf{A}^{T}$$

E ainda,

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

Câmera

Parâmetros

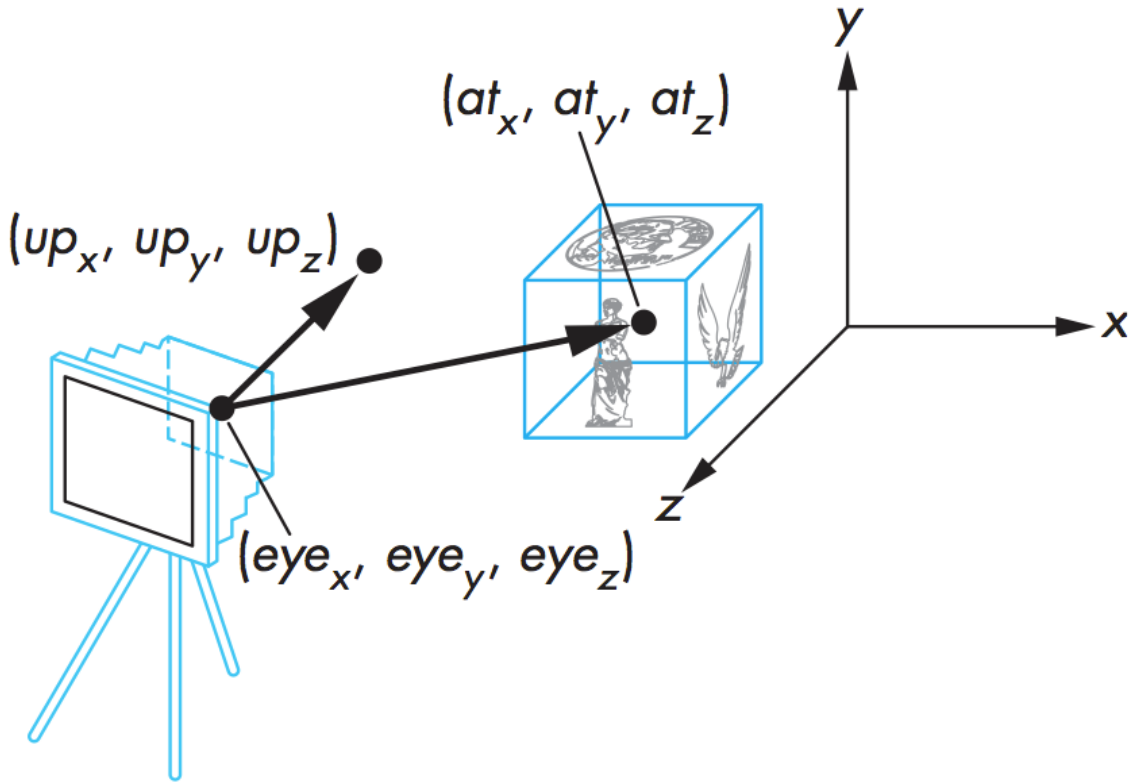


Por fim:

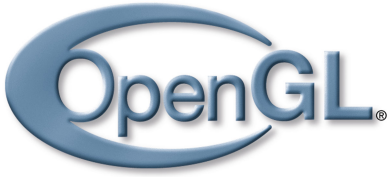
$$\mathbf{V} = \mathbf{RT} = \begin{bmatrix} u'_x & u'_y & u'_z & -xu'_x - yu'_y - zu'_z \\ v'_x & v'_y & v'_z & -xv'_x - yv'_y - zv'_z \\ n'_x & n'_y & n'_z & -xn'_x - yn'_y - zn'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Câmera

Parâmetros



Matrix4f lookAt(vector4f eye, vector4f at, vector4f up)



Câmera

Exemplo

Código fonte!

Computação Gráfica

TCC-00291

Assunto: Câmera