

Computação Gráfica e Interfaces

2016-2017
Fernando Birra

LookAt

2015-2016

Fernando Birra

Objetivos

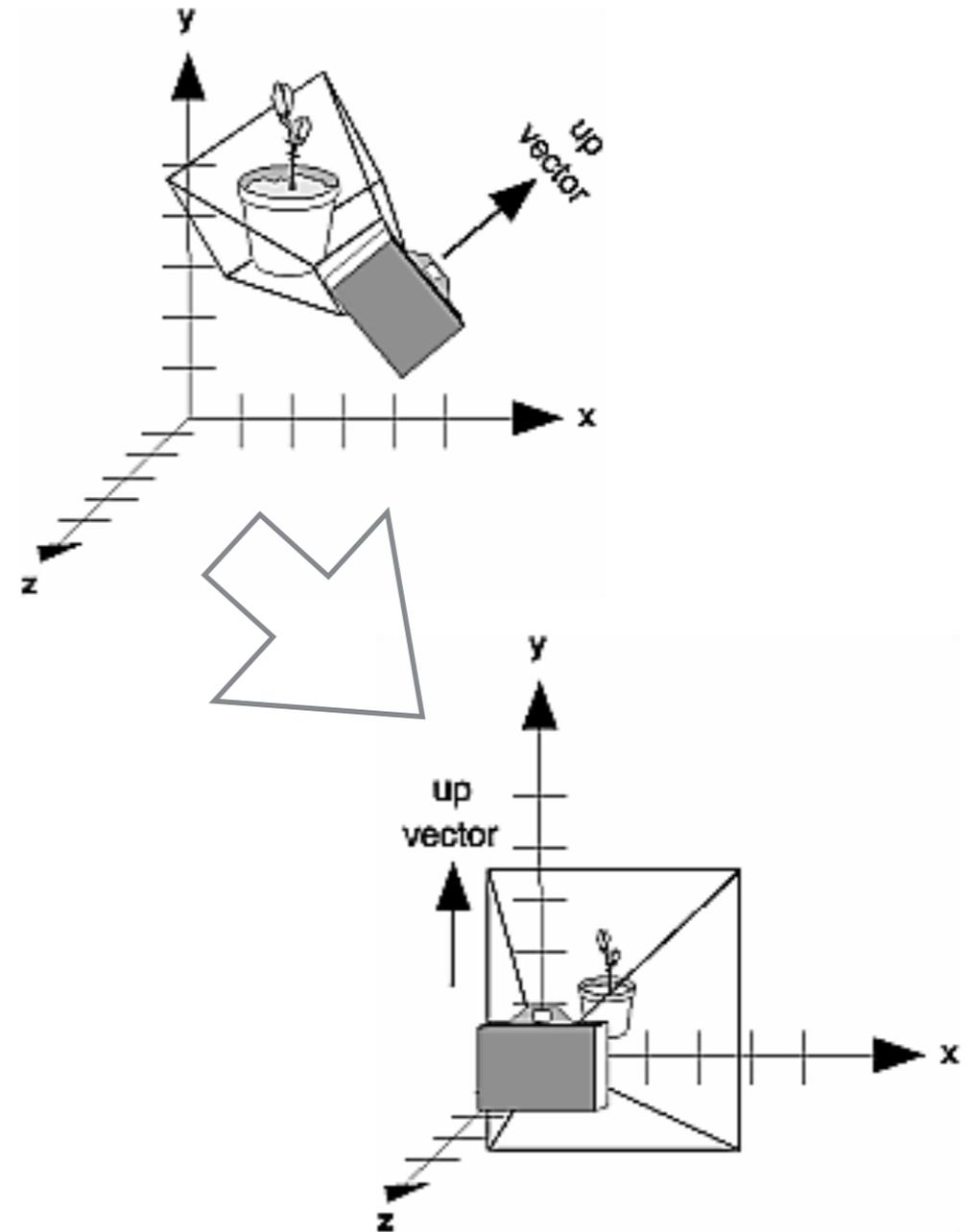
- Saber implementar uma orientação genérica da câmara
- Conhecer a função `lookAt()` da biblioteca `MV.js`
- Deduzir a matriz que transforma do referencial do mundo para o referencial da câmara

Problema

- Nas (maior parte das) matrizes de projeção estudadas assume-se que o plano de projeção é o plano $Z=0$, e que o observador está a olhar para o lado negativo do eixo Z .
- Como conciliar isso com uma orientação arbitrária da câmara?

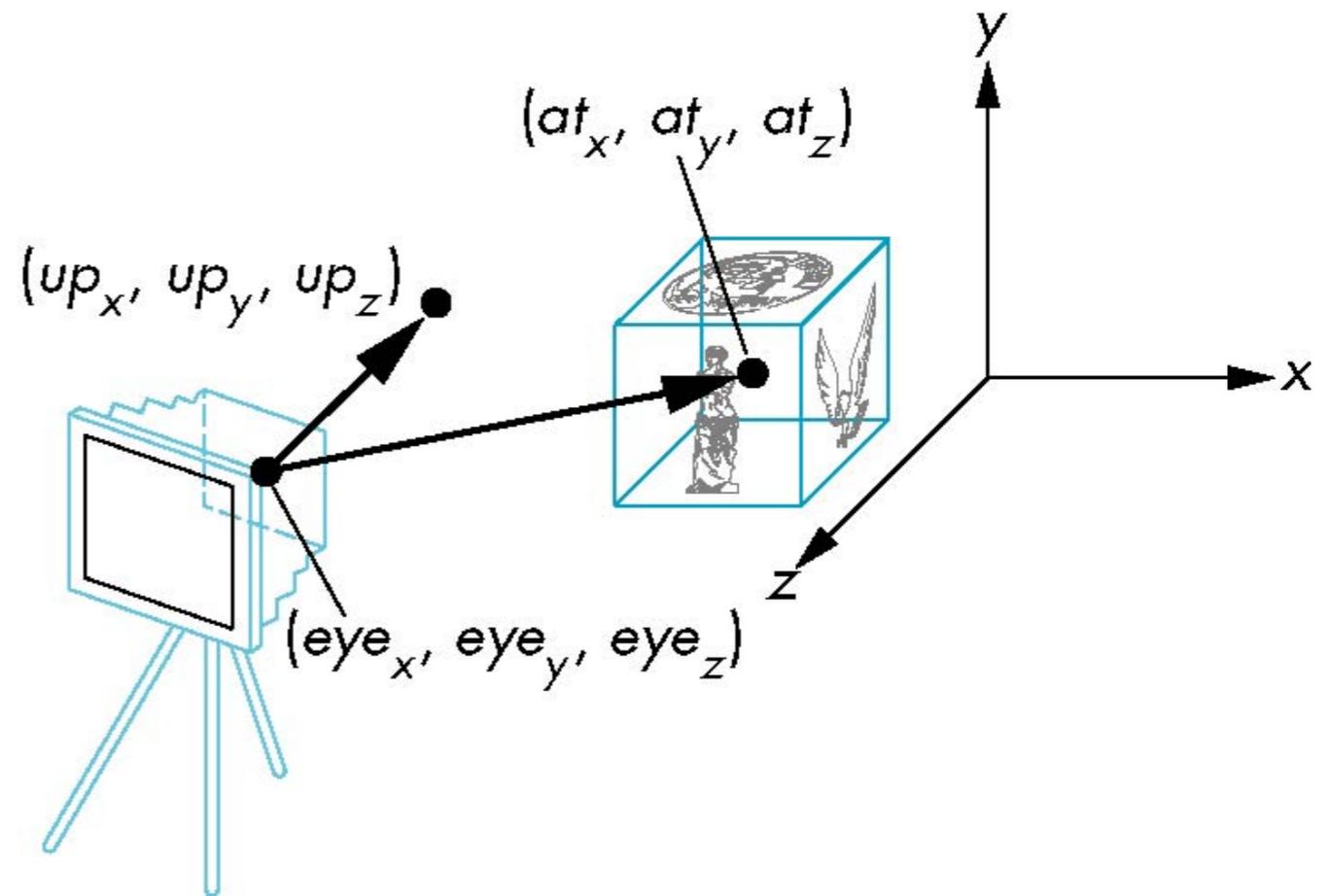
Solução

- Transformar coerentemente (da mesma forma) todos os objetos da cena e a câmara de tal modo que a câmara passe a estar na origem, apontada para o lado negativo do eixo Z e com a direção vertical coincidente com o eixo Y .



lookAt()

- A biblioteca MV.js oferece uma função que permite especificar uma orientação genérica para a câmara:
- `lookAt(eye, at, up)`
- `eye`, `at` e `up` são fornecidos em coordenadas do mundo (**W**orld **C**oordinates)



lookAt: estratégia para implementação

- Estabelecer um referencial ortonormado para a câmara
- Mover o ponto eye para a origem do referencial

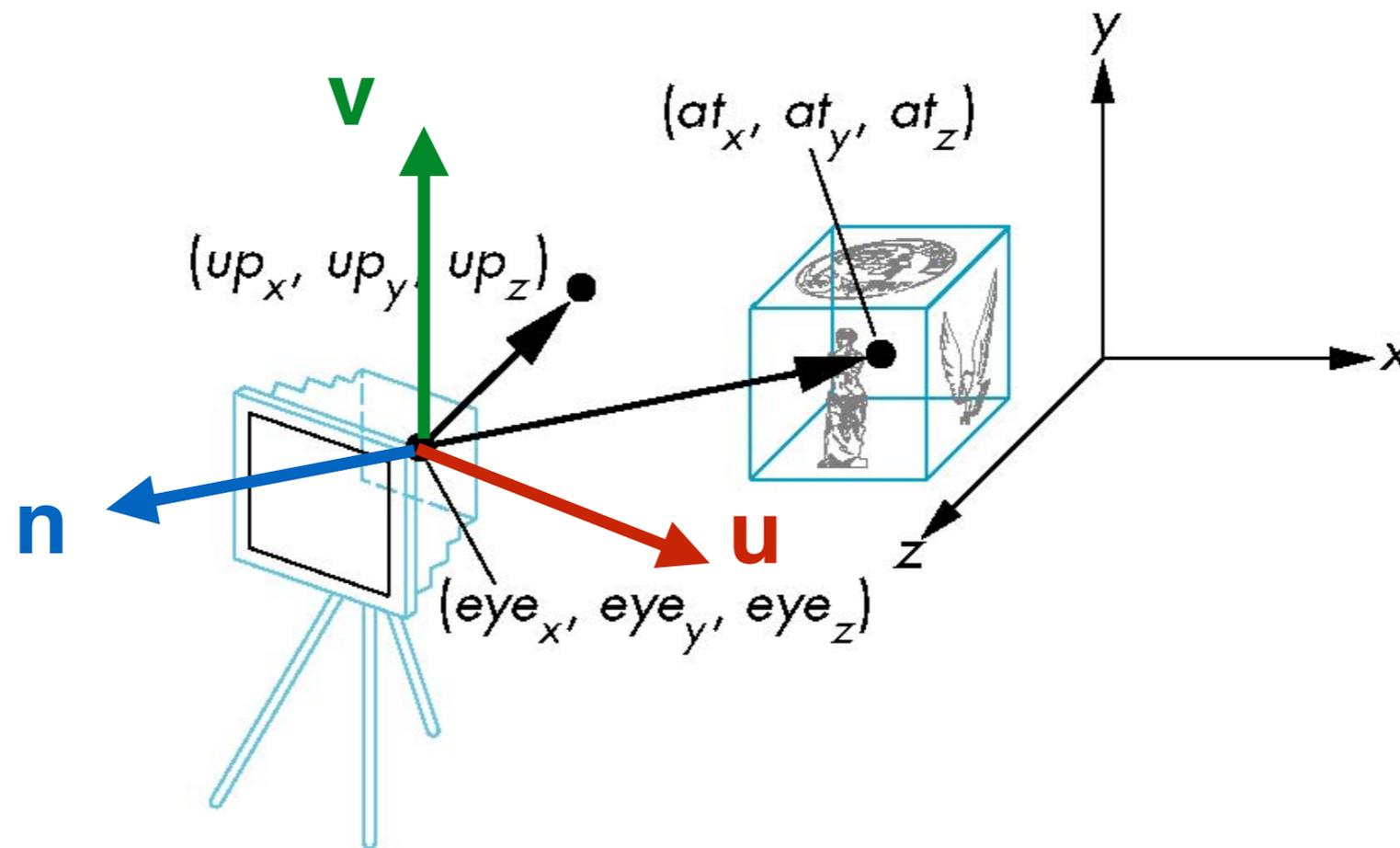
$$T(-eye_x, -eye_y, -eye_z)$$

- Rodar de forma a que o referencial da câmara se alinhe com o referencial do mundo

Seja R a matriz respetiva

- A transformação final será: $R T(-eye_x, -eye_y, -eye_z)$

Referencial da câmara



Determinação de u , v e n

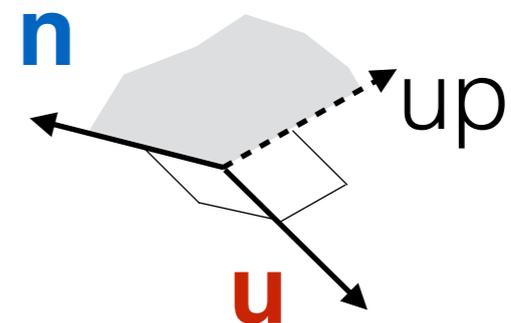
- u , v e n são perpendiculares entre si e formam um sistema de coordenadas direito.
- n está alinhado com o vetor que define a direção para onde a câmara está apontada, mas com o sentido inverso

$$\mathbf{n} = (\text{eye} - \text{at}) / \|\text{eye} - \text{at}\|$$

Determinação de u , v e n

- O vetor up poderá, por mero acaso, ser logo perpendicular a n e, nesse caso, v estaria logo determinado. Não estando isso garantido, é preferível determinar primeiro o vetor u .
- u representa a direção horizontal da câmara e será perpendicular ao plano definido pelos vetores up e n .

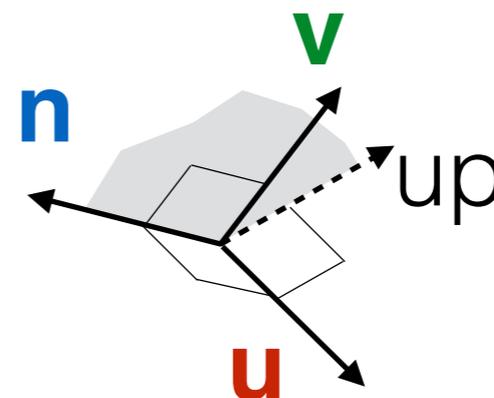
$$u = (up \times n) / \|up \times n\|$$



Determinação de u , v e n

- Finalmente, o vetor v pode agora ser facilmente determinado por forma a ser perpendicular quer a u , quer a n :

$$u = n \times v$$



Dedução de R

- A matriz R, será responsável por orientar o referencial da câmara (entretanto com a origem partilhada com o referencial do mundo via a translação entretanto efetuada), por forma a fazer coincidir os seguintes pares de eixos:
 - **u** com x; **v** com y e **n** com z
- Outra forma de interpretar o papel desta matriz é pensar que ela transforma pontos e vetores no referencial do mundo para o referencial **u,v,n** (com a origem em comum). Assim:
 - $R\mathbf{u} = (1,0,0,0)^T$; $R\mathbf{v} = (0,1,0,0)^T$; $R\mathbf{n} = (0,0,1,0)^T$

Dedução de R

- $R\mathbf{u} = (1,0,0,0)^T$; $R\mathbf{v} = (0,1,0,0)^T$; $R\mathbf{n} = (0,0,1,0)^T$

Sendo R uma matriz de rotação, com a forma ao lado, resulta que:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(r_{11}, r_{12}, r_{13}, 0)\mathbf{u}^T = 1$$

$$(r_{11}, r_{12}, r_{13}, 0)\mathbf{v}^T = 0$$

$$(r_{11}, r_{12}, r_{13}, 0)\mathbf{n}^T = 0$$

$$(r_{21}, r_{22}, r_{23}, 0)\mathbf{u}^T = 0$$

$$(r_{21}, r_{22}, r_{23}, 0)\mathbf{v}^T = 1$$

$$(r_{21}, r_{22}, r_{23}, 0)\mathbf{n}^T = 0$$

$$(r_{31}, r_{32}, r_{33}, 0)\mathbf{u}^T = 0$$

$$(r_{31}, r_{32}, r_{33}, 0)\mathbf{v}^T = 0$$

$$(r_{31}, r_{32}, r_{33}, 0)\mathbf{n}^T = 1$$

Dedução de R

$$(r_{11}, r_{12}, r_{13}, 0) \mathbf{u}^T = 1$$

$$(r_{11}, r_{12}, r_{13}, 0) \mathbf{v}^T = 0$$

$$(r_{11}, r_{12}, r_{13}, 0) \mathbf{n}^T = 0$$

$$(r_{21}, r_{22}, r_{23}, 0) \mathbf{u}^T = 0$$

$$(r_{21}, r_{22}, r_{23}, 0) \mathbf{v}^T = 1$$

$$(r_{21}, r_{22}, r_{23}, 0) \mathbf{n}^T = 0$$

$$(r_{31}, r_{32}, r_{33}, 0) \mathbf{u}^T = 0$$

$$(r_{31}, r_{32}, r_{33}, 0) \mathbf{v}^T = 0$$

$$(r_{31}, r_{32}, r_{33}, 0) \mathbf{n}^T = 1$$



$$(r_{11}, r_{12}, r_{13}) = (\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z)$$

$$(r_{11}, r_{12}, r_{13}) \perp (\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z)$$

$$(r_{11}, r_{12}, r_{13}) \perp (\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z)$$

$$(r_{21}, r_{22}, r_{23}) \perp (\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z)$$

$$(r_{21}, r_{22}, r_{23}) = (\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z)$$

$$(r_{21}, r_{22}, r_{23}) \perp (\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z)$$

$$(r_{31}, r_{32}, r_{33}) \perp (\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z)$$

$$(r_{31}, r_{32}, r_{33}) \perp (\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z)$$

$$(r_{31}, r_{32}, r_{33}) = (\mathbf{n}_x, \mathbf{n}_y, \mathbf{n}_z)$$

Conclusão

- Estando determinada a matriz R :

$$R = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- A matriz final construída pela função `lookAt()` será:

$$R \cdot T(-eye_x, -eye_y, -eye_z):$$

Conclusão

R T(-eye_x, -eye_y, -eye_z):

$$\begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ n_x & n_y & n_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -eye_x \\ 0 & 1 & 0 & -eye_y \\ 0 & 0 & 1 & -eye_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Desenvolvendo:

Matriz criada pela função
lookAt: transforma
coordenadas do mundo em
coordenadas da câmara

$$\begin{bmatrix} u_x & u_y & u_z & -eye \cdot u \\ v_x & v_y & v_z & -eye \cdot v \\ n_x & n_y & n_z & -eye \cdot n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

lookAt()

- A matriz M_{view} devolvida por `lookAt()` deverá ser aplicada após as transformações de modelação M_{model} , que convertem os objetos primitivos em instâncias na cena e antes da projeção, M_{proj}

$$P' = M_{proj} \cdot M_{view} \cdot M_{model} \cdot P$$

- As projeções axonométricas poderão, em alternativa, ser substituídas pela projeção ortogonal no plano XY, com recurso a `lookAt`.
- Exemplo (isometria):

`lookAt(vec3(1,1,1), vec3(0,0,0), vec3(0,1,0))`

eye

at

up