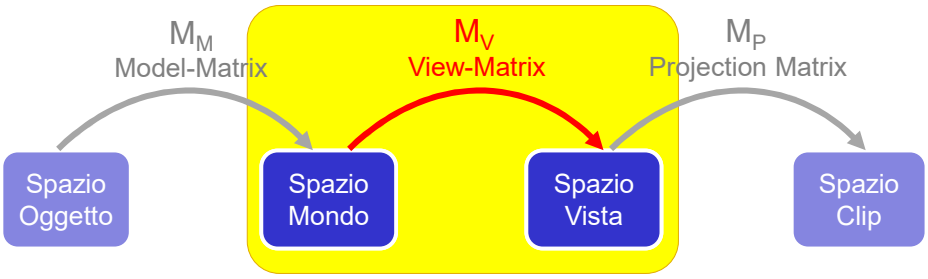
Trasformazioni nel pipeline di rendering visione di insieme

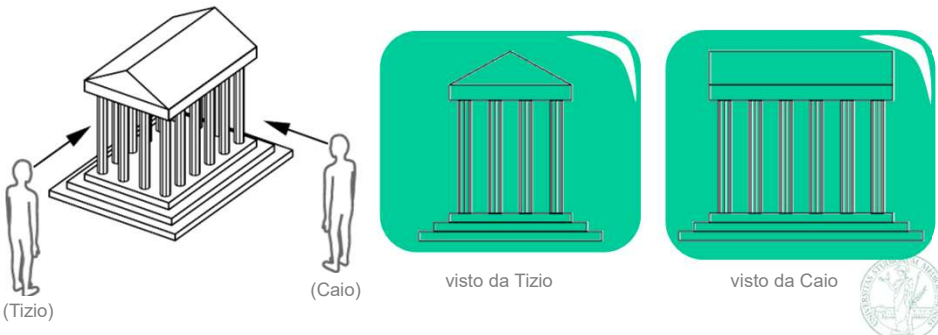


3

Matrice di vista

- dipende da dov'è e come è orientata la *camera*
  - cioè dai sui *parametri estrinseci*

la macchina fotografica



4

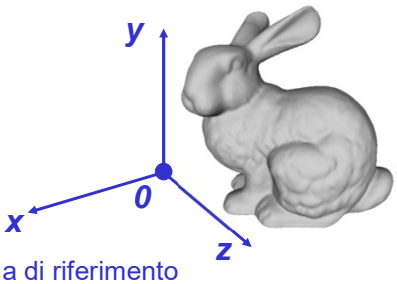
Trasformazione di “Vista”

- ✓ Da: **World Frame**
  A: **View Frame**
- ✓ La Trasformazione di Vista porta tutti gli oggetti da renderizzare in uno spazio di riferimento riferito alla telecamera

«un computer graphicist preferisce portare la montagna davanti alla macchina fotografica, piuttosto che la macchina fotografica davanti alla montagna»

5

Spazio (sistema di riferimento) Vista



sistema di riferimento del mondo (*World Space*)

sistema di riferimento Vista (*View Space*)

6

Trasformazione di "Vista"

- ✓ Da: **World Frame**
  A: **View Frame**
- ✓ Dipende interamente dai «**parametri estrinseci**» della macchina fotografica (virtuale)
  - ⇒ Cioè da *dove è, e come è orientata* (nel mondo)
  - ⇒ (per es: un tipico task di Computer Vision: «registrare una foto» = evincere i parametri estrinseci della camera al momento del suo scatto)
- ✓ E' un cambio di sistema di riferimento, cioè una trasformazione affine…
  - ⇒ Matrice di Vista = la Matrice che fa questa trasformazione

7

Descrizione dei parametri estriseci

Un modo per esprimere i parametri estrinseci:

1. Posizione dell'osservatore (POV)
   - ⇒ Cioè, dell'occhio (la pupilla), della macchina fotografica (il punto di fuoco) etc
2. Posizione di un punto target osservato
   - ⇒ Si richiede che questo punto compaia in mezzo alla foto
   - ⇒ Oppure, equivalentemente, una **direzione** di vista
3. Vettore "alto" ("up-vector")
   - ⇒ Descrive una direzione che, nella foto, deve apparire come direzione verticale, verso l'alto
   - ⇒ Distingue, ad es, una foto "portrait" da una "landscape" (e da un "campo obliquio")

- ✓ Nota: si tratta di punti e vettori espressi nel Sistema di coordinate Mondo
  - ⇒ Descrivono la posizione / orientamento della camera NELLA SCENA

8

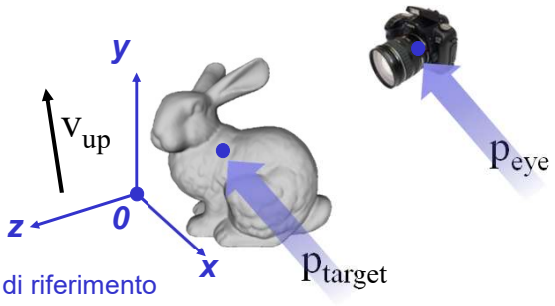
Esempio tipico di costruzione transformazione di vista

Input:

1) camera position: p_{eye}
2) target position: p_{target}
3) vettore di alto: v_{up}

nb: punti e vettori espressi in spazio mondo!

un esempio di descrizione esaustiva dei *parametri estrinseci* della camera



sistema di riferimento mondo *(world space)*

9

Esempio tipico di costruzione transformazione di vista

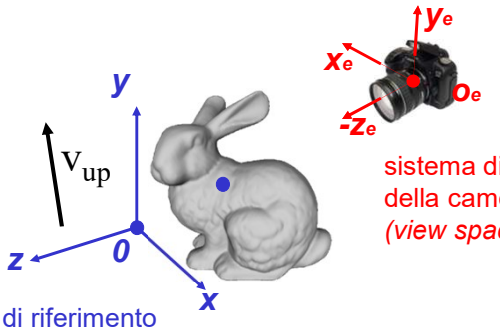
Input:

1) camera position: p_{eye}
2) target position: p_{target}
3) vettore di alto: v_{up}

Output:

Matrice di Trasformazione

world space ⟶ *view space*



sistema di riferimento della camera *(view space)*

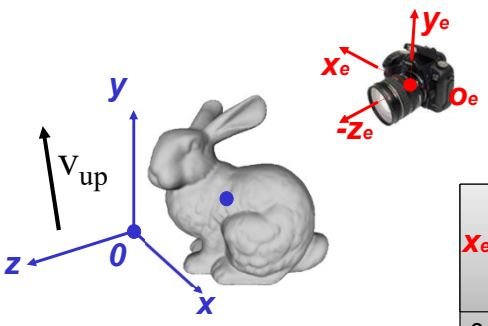
sistema di riferimento mondo *(world frame)*

10

Esempio tipico di costruzione transformazione di vista

Input:

1) camera position: p_{eye}
2) target position: p_{target}
3) vettore di alto: v_{up}




```
vec3 oe;
vec3 xe, ye, ze;

ze = p_target - p_eye;
ze = -ze;
ze = normlize( ze );

xe = cross( vup , ze );
xe = normalize( xe );

ye = cross(ze, xa);
```

x_e	y_e	z_e	o_e
0	0	0	1

matrice che va da spazio vista a spazio mondo.
E' l'inversa di quella che volevamo.
Ergo, va invertita.

11

Esempio tipico di costruzione transformazione di vista

Origine e assi del sistema vista espressi nelle coord del sistema mondo

L'asse zeta va verso l'osservatore

Deve essere reso unitario

"completamento di base"

normalizzaz non necessaria (perché?)

nb: quando si fallisce? le due normalizz possono essere div by 0? quando?

```
vec3 oe;
vec3 xe, ye, ze;

ze = p_eye - p_target;
ze = normlize( ze );

xe = cross( vup , ze );
xe = normalize( xe );

ye = cross(ze, xa);
```

x_e	y_e	z_e	o_e
0	0	0	1

matrice che va da **spazio vista** a **spazio mondo**.
E' l'inversa di quella che cerchiamo.
Ergo, va invertita.

12

Inversione di una rotazione (ripetiamoci)

$$\begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R^T & 0 \\ 0 & 1 \end{bmatrix}$$

rotazione 4x4 generica
(asse passante per origine)

dove:
R rotazione 3x3,
cioè ortonormale a det 1,
cioè $v_0\ v_1\ v_2$:
- unit sized
- ortogonali a due a due

$$R = \begin{bmatrix} v_0 & v_1 & v_2 \end{bmatrix}$$

14

Inversione di una rotazione (ripetiamoci)

$$\begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} R^T & 0 \\ 0 & 1 \end{bmatrix}$$

rotazione 4x4 generica
(asse passante per origine)

dimostrazione (traccia):

$$R^T * R = \begin{bmatrix} v_0 \\ v_1 \\ v_2 \end{bmatrix} * \begin{bmatrix} v_0 & v_1 & v_2 \end{bmatrix} = I$$

15

Inversione di una traslazione (ripetiamoci)

$$\begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} I & -t \\ 0 & 1 \end{bmatrix}$$

matrice 4x4 di traslazione

matrice 4x4 di traslazione del vettore opposto

16

Rototraslazione (isometria) (tutte e sole le trasformazioni rigide)

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix}$$

roto-traslazione (4x4) (o *isometria*)

traslazione

rotazione (asse passante per origine)

Verificare eseguendo il prodotto «riga per colonna» fra i sottoblocchi

17

Inversione di una roto-traslazione

nb: $(A\,B)^{-1} = B^{-1}\,A^{-1}$

$$\begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}^{-1} = \left(\begin{bmatrix} I & t \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix} \right)^{-1} =$$

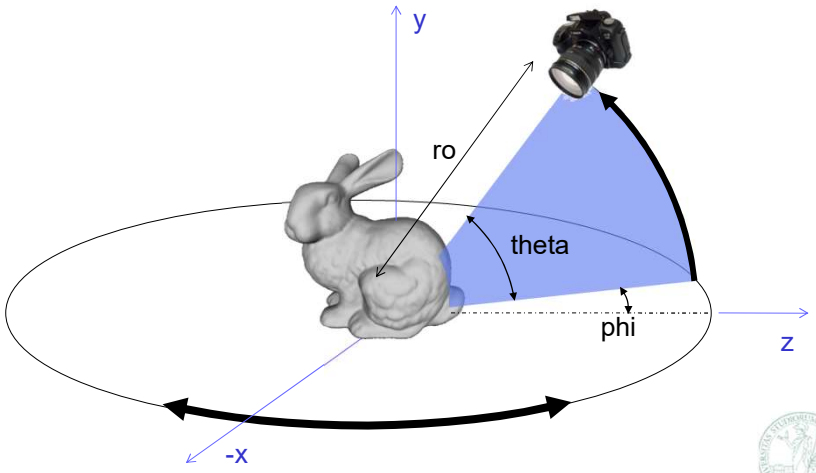
$$= \begin{bmatrix} R^T & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} I & -t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R^T & \\ 0 & 1 \end{bmatrix}$$

$$R^T * -t$$

18

Alternativa: definire la *matrice di vista* combinando rotazioni e traslazioni

Esempio: una semplice "Trackball":



19

Osservazione

Matrice di modellazione M di un oggetto qualsiasi:
da spazio di quel dato oggetto, a spazio mondo

Matrice di Vista V:
da spazio mondo a spazio Vista
(cioè lo spazio oggetto… della camera)

Quindi, se l'oggetto in questione è la *camera*: V è l'inversa di M

la trasformaz di **modellazione** che localizza
un **oggetto** in una certa posizione & orientamento

è l'inversa

della trasformaz di **vista** necessaria per piazzare la
camera nella stessa posizione & orientamento

20

Trackball (elemento di una interfaccia grafica)

- ✓Trackball = interfaccia
  usata per consentire all'utente selezionare in modo semplice i parametri estrinseci
  - ⇒e dunque la matrice di vista
- ✓Una semplice trackball: posizione & orientamento della camera controllata da soli tre parametri:
  - ⇒angoli (`phi` e `theta`) + distanza (`ro`)
  - ⇒es mappati su assi X Y mouse + (mousewheel)
- ✓Utile per visualizzare un piccolo oggetto
  - ⇒permettere alla camera di orbitare intorno all'origine

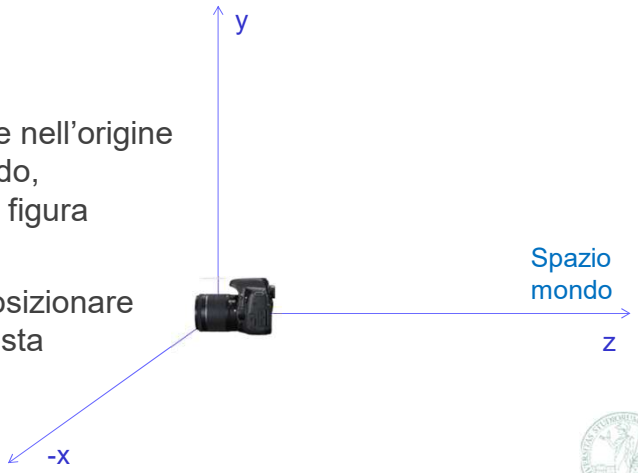
21

Passo 0

Se la matrice di vista è l'identità, allora lo spazio vista coincide con lo spazio mondo

La camera è dunque nell'origine dello spazio mondo, orientate come in figura

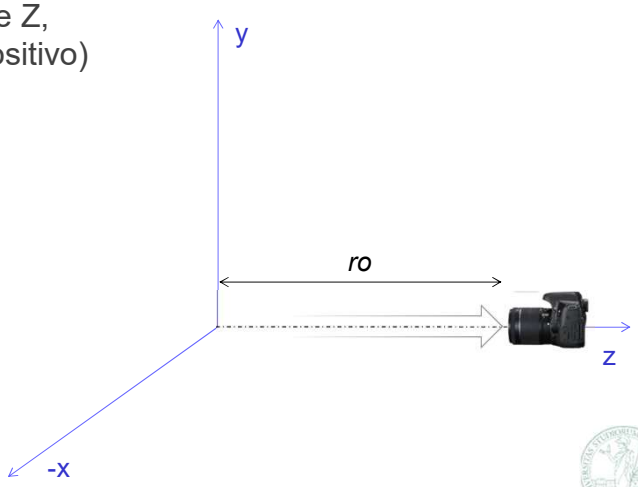
Immaginiamo di riposizionare la camera da questa situazione.



22

Passo 1

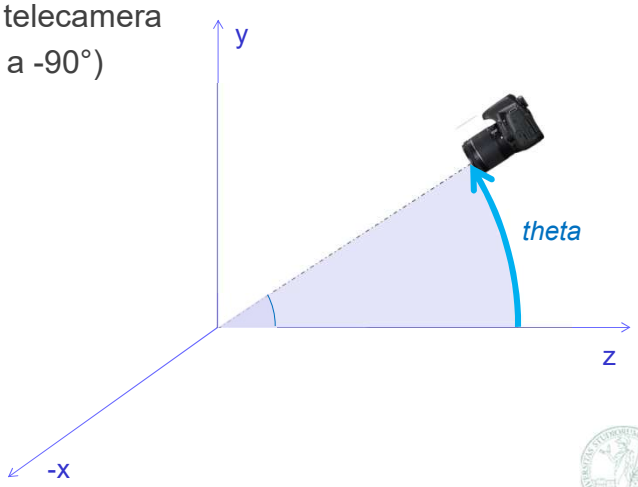
Spostare la macchina all'indietro di *ro* unità all'indietro (dunque, sull'asse Z, con *ro* numero positivo)



23

Passo 2

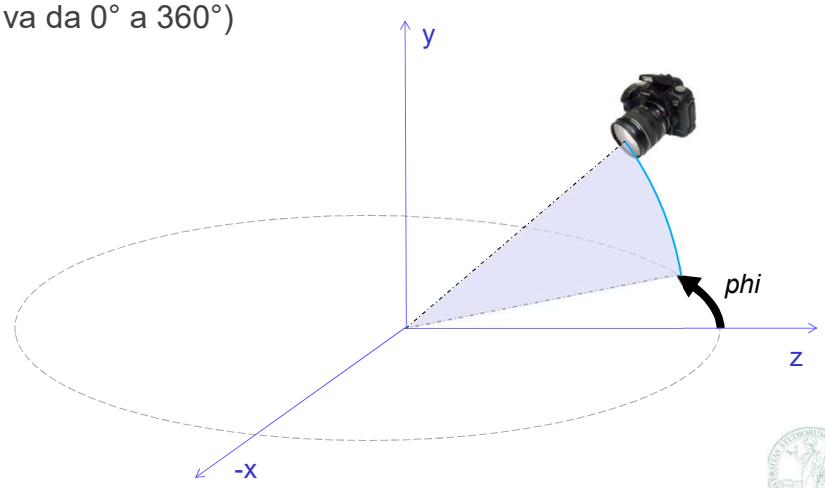
Ruotare di theta gradi attorno all'asse delle X, quindi alzando la telecamera (theta va da +90° a -90°)



24

Passo 3

Ruotare la camera di phi gradi attorno all'asse delle Y (phi va da 0° a 360°)



25

Trackball (elemento di una interfaccia grafica)

- ✓ Interfaccia base usata per selezionare in modo semplice i parametri estrinseci (matr vista)
- ✓ Posizione / orientamento della camera controllata da tre parametri:
  - ⇒angoli (`phi` e `theta`) + distanza (`ro`)
  - ⇒es mappati su assi X Y mouse + (mousewheel)
- ✓ Utile per visualizzare un piccolo oggetto
  - ⇒permettere alla camera di orbitare intorno all'origine

26

Realizzazoine della Trackball

- ✓ Matrice che sposta la telecamera nel posto voluto:

$$\mathbf{R}_{\mathrm{Y}}(phi) \cdot \mathbf{R}_{\mathrm{X}}(theta) \cdot \mathbf{T}(0,0,ro)$$

(matrice di) **rotazione** attorno all'asse delle Y di angolo phi

seguita da ← ← ←

(matrice di) **rotazione** attorno all'asse delle X di angolo theta

seguita da ← ← ←

(matrice di) **traslazione** di ro sull'asse delle z

- ✓ Questa è la matrice di modellazione dell'oggetto camera
  - ⇒ va da spazio camera a spazio mondo!
- ✓ La **matrice di vista** è dunque la sua *inversa*, e cioè:

$$\mathbf{T}(0,0,-ro) \cdot \mathbf{R}_{\mathrm{X}}(-theta) \cdot \mathbf{R}_{\mathrm{Y}}(-phi)$$

Le inverse, in ordine inverso

27