

3

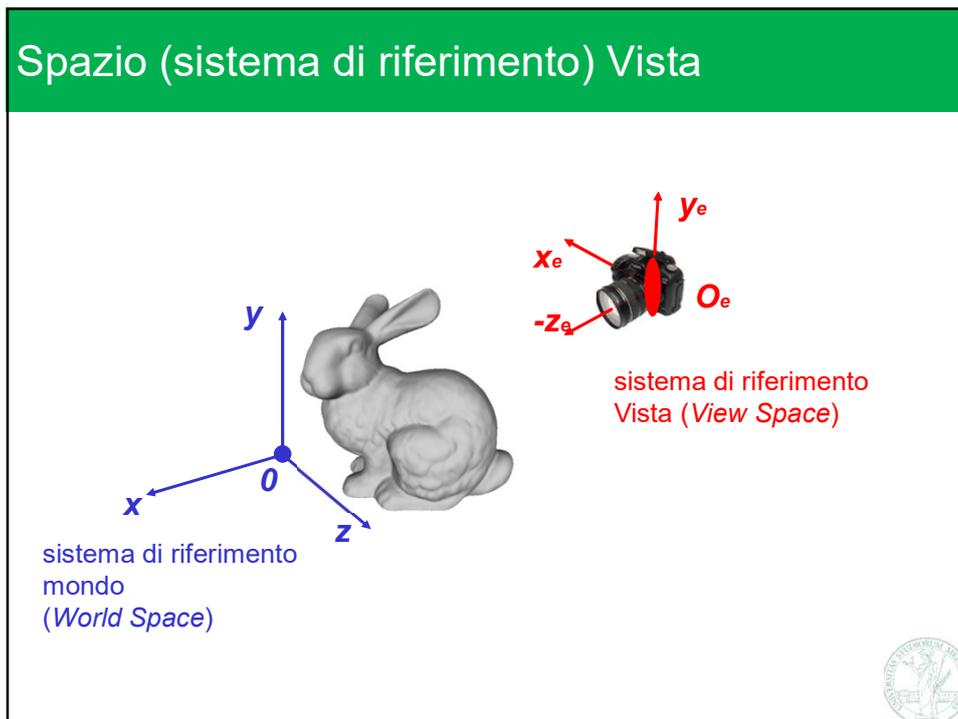
Trasformazione di "Vista"

- ✓ 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»

A small logo of the University of Milan is visible in the bottom right corner.

5



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

Una descrizione dei parametri estrinseci

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*" (o da un "campo obliquo")
- ✓ Nota: si tratta di punti e vettori tutti espressi nel Sistema di coordinate Mondo
- ⇒ Descrivono la posizione / orientamento della camera NELLA SCENA



8

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... dell'oggetto *camera*)

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

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

è l'*inversa*

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



9

Da parametri estrinseci a matrice di vista

✓ Piano:

1. Usare i parametri per trovare il set { punto origine + tre vettori asse }, dello spazio vista (lo spazio delle camera) espressi in spazio mondo
2. La matrice M le cui colonne sono quei vettori e punto rappresenta la trasformazione che porta lo spazio vista nello spazio mondo
3. L'inversa di questa matrice è quindi la matrice di vista

attraverso l'algebra di punti e vettori

nota: se la telecamera fosse un qualsiasi oggetto da renderizzare nella scena, allora M sarebbe la sua matrice di modellazione



10

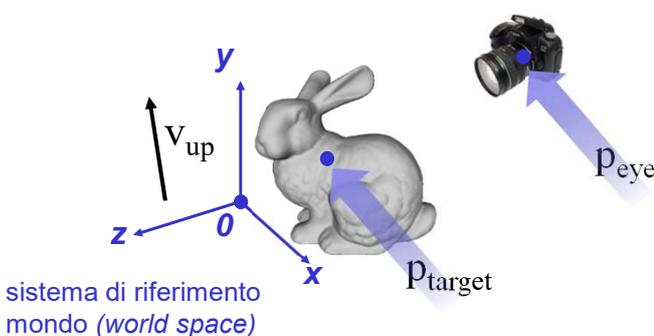
Da parametri estrinseci a matrice di vista

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*)



11

Esempio tipico di costruzione trasformazione di vista

Input:

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

Output:
 Matrice di Trasformazione
 $world\ space \rightarrow view\ space$

sistema di riferimento mondo (world frame)

sistema di riferimento della camera (view space)

12

Esempio tipico di costruzione trasformazione 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 = normalize( 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.

13

Esempio tipico di costruzione trasformazione di vista

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

Per de, l'asse zeta della vista va verso l'osservatore

Deve essere reso unitario (perché?)

Operazioni dette "completamento di base"

Normalizzaz non necessaria (perché?)

Osservaz:
è possibile fallire? le due normalizazioni possono essere div by 0? quando?

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

ze = p_eye - p_target;
ze = normalize( 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.

14

Esempi di pseudocodice per la costruzione della matrice di vista

```
mat4 view_matrix( vec3 p_eye, vec3 p_target, vec3 v_up )
{
    vec3 xe, ye, ze; // gli assi del sist. di rif. vista

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

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

    ye = cross( ze, xa );

    mat4 m; // l'inversa della mat di vista
    m[0] = vec4( xe, 0 ); // setta la lma colonna di m
    m[1] = vec4( ye, 0 );
    m[2] = vec4( ze, 0 );
    m[3] = vec4( p_pov, 1 );

    return invese(m); // inversione generica? (spreco!)
}
```

15

Inversione di una rotazione (ripetiamoci)

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

rotazione 4x4 generica
(asse passante per origine)

dove:

- R rotazione 3x3, cioè ortonormale a det 1, cioè $v_0 v_1 v_2$:
- unitari
- ortogonali a due a due

$$R = [v_0 \ v_1 \ v_2]$$

16

Inversione di una rotazione (ripetiamoci)

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

rotazione 4x4 generica
(asse passante per origine)

dimostrazione (traccia):

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

riga
x
colonna:

17

Inversione di una roto-traslazione

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

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

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

20

Altro esempio: definire la *matrice di vista* combinando rotazioni e traslazioni

Una semplice "Trackball":

21

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 (ϕ e θ) + distanza (r_0)
 - ⇒ es mappati su assi X Y mouse + (mousewheel)
- ✓ Utile per visualizzare un piccolo oggetto
 - ⇒ permettere alla camera di orbitare intorno all'origine



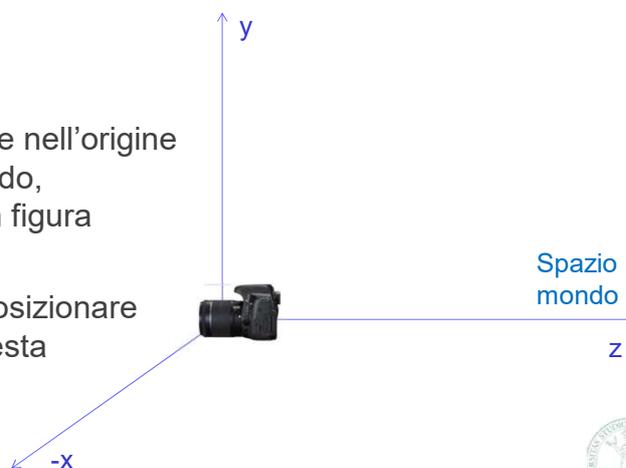
22

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.



23

Passo 1

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

The diagram illustrates a 3D coordinate system with axes labeled x , y , and z . The z -axis points to the right, the y -axis points upwards, and the x -axis points downwards and to the left. A camera is positioned on the positive z -axis. A horizontal double-headed arrow labeled ro indicates the distance from the origin to the camera. A dashed line shows the camera's position relative to the origin.

24

Passo 2

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

The diagram illustrates a 3D coordinate system with axes labeled x , y , and z . The z -axis points to the right, the y -axis points upwards, and the x -axis points downwards and to the left. A camera is positioned in the yz -plane, tilted upwards from the z -axis. A blue arc labeled $theta$ indicates the angle of rotation from the z -axis to the camera's position.

25

Passo 3

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

26

La matrice di Vista controllata dalla Trackball

✓ Matrice che sposta la telecamera nel posto voluto:

$$\mathbf{R}_Y(phi) \cdot \mathbf{R}_X(theta) \cdot \mathbf{T}(0,0,ro)$$

(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:

$$\mathbf{T}(0,0,-ro) \cdot \mathbf{R}_X(-theta) \cdot \mathbf{R}_Y(-phi)$$

Le inverse, in ordine inverso

27