

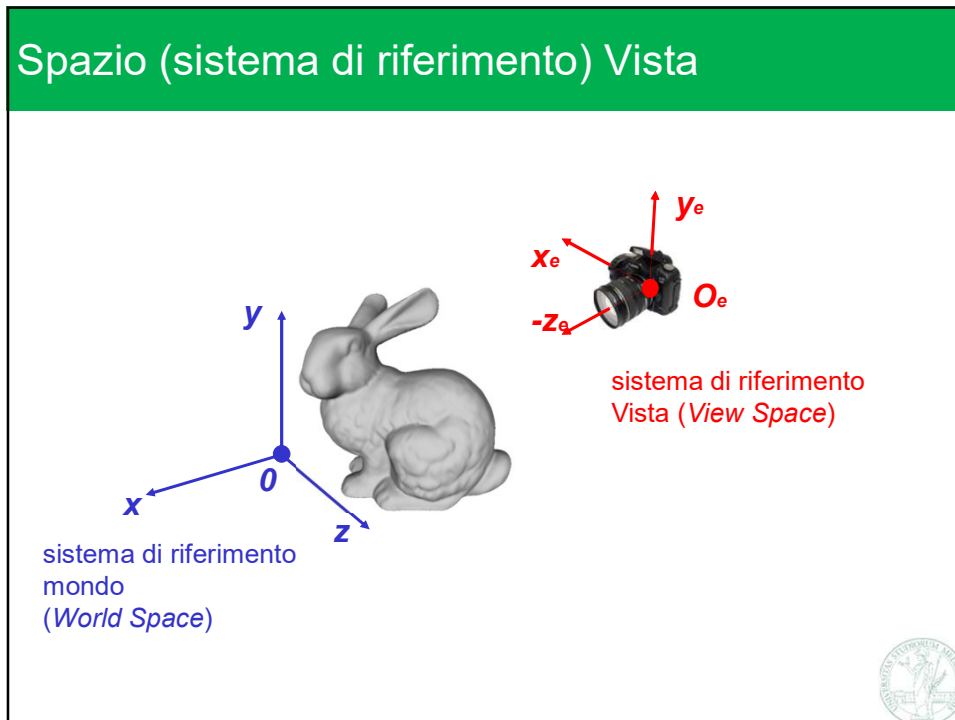
3

La 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



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 modo (completo) per descrivere i 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



10

Un descrizione (completa) dei parametri estrinseci

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

} nb: punti e vettori espressi in spazio mondo!

sistema di riferimento mondo (world frame)

11

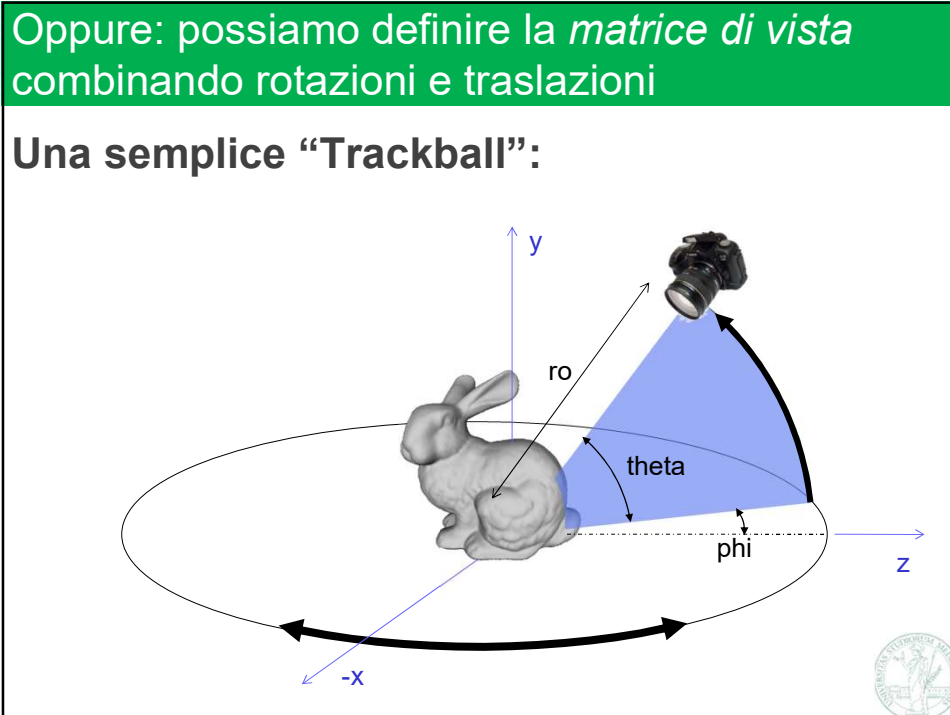
Esercizio: come calcolare la matrice di vista V a partire dai parametri estrinseci p_{eye} , p_{target} e \vec{v}_{up} ?

✓ Traccia della soluzione: attraverso l'algebra di punti e vettori

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

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

12



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

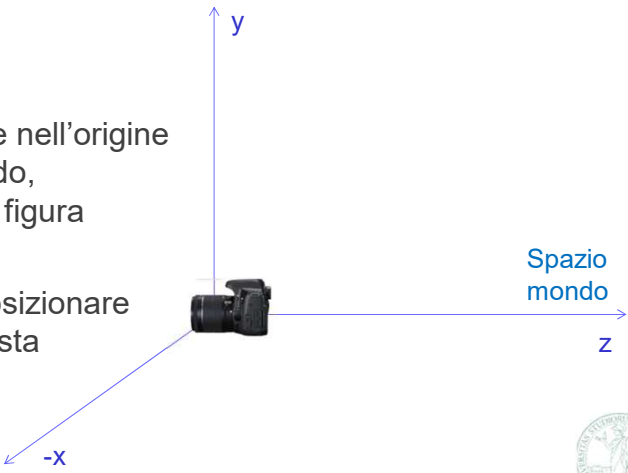
23

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.

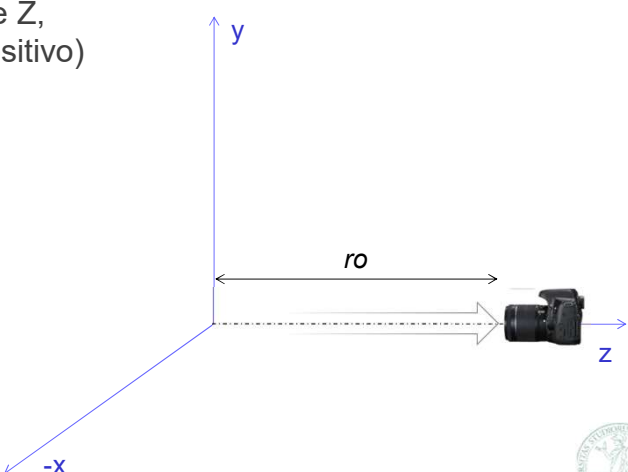


The diagram shows a 3D coordinate system with axes labeled y (vertical), z (horizontal to the right), and $-x$ (diagonal down-left). A camera is positioned at the origin (0,0,0). The text 'Spazio mondo' is written above the z -axis. A small circular logo of the University of Milan is in the bottom right corner.

24

Passo 1

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

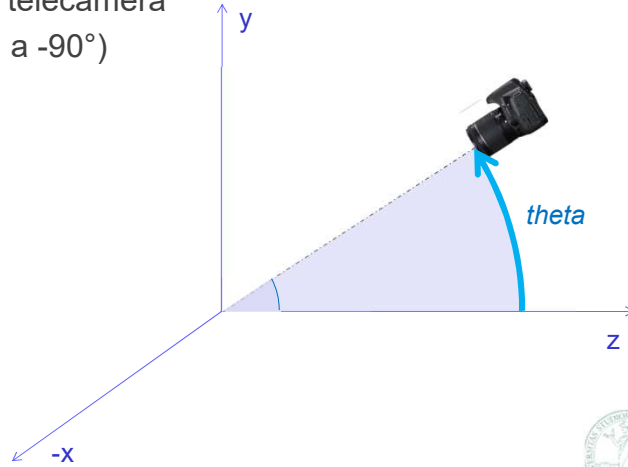


The diagram shows the same 3D coordinate system as in Step 0. The camera is now moved along the z -axis to a position labeled ro . A double-headed arrow indicates the distance from the origin to the camera. The text 'Spazio mondo' is not present in this diagram. A small circular logo of the University of Milan is in the bottom right corner.

25

Passo 2

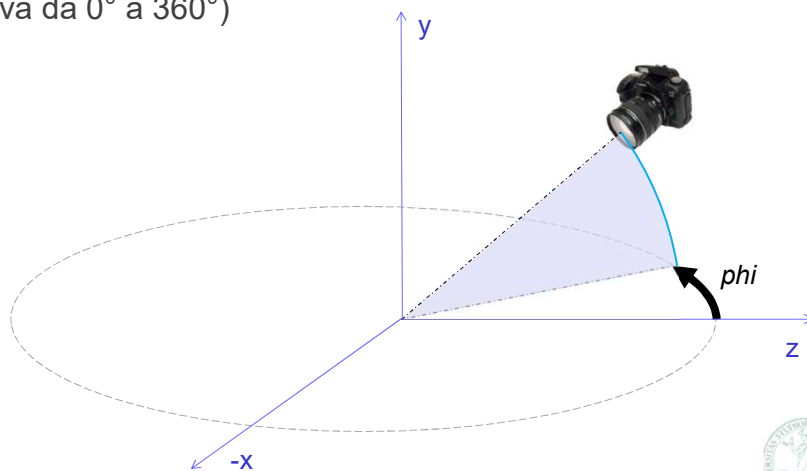
Ruotare di θ gradi
attorno all'asse delle X,
quindi alzando la telecamera
(θ va da $+90^\circ$ a -90°)



26

Passo 3

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

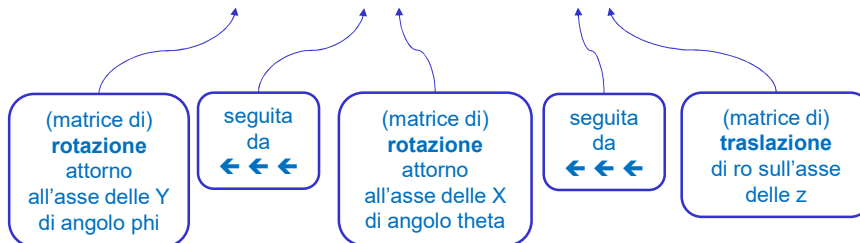


27

La matrice di Vista controllata dalla Trackball

- ✓ Matrice che sposta la telecamera nel posto voluto:

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



- ✓ 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

28

Nel nostro progetto (vedi `cgLab03.html` e `.js`)

- ✓ La trackball come oggetto JavaScript (o JSON):

```
var miaTrackball = {
  ro: 3,
  phi: 45, // in gradi!
  theta: -45, // in gradi!
}
```

- ✓ Per costruire la matrice di vista
(in un apposito metodo):

```
var v = new THREE.Matrix4();
v.identity();
v.multiply( matriceDiTraslazione( 0, 0, -this.ro ) );
v.multiply( matriceDiRotazioneX( -this.theta ) );
v.multiply( matriceDiRotazioneY( -this.phi ) );
```

(vedi progetto nella pagina del corso per altri dettagli)

30