

Marco Tarini - Computer Graphics 2022/2023  
Università degli Studi di Milano

**La sequenza di trasformazioni nel rendering:  
visione di insieme**

1

**La transform usata duante il rendering  
(con approccio rasterization)**

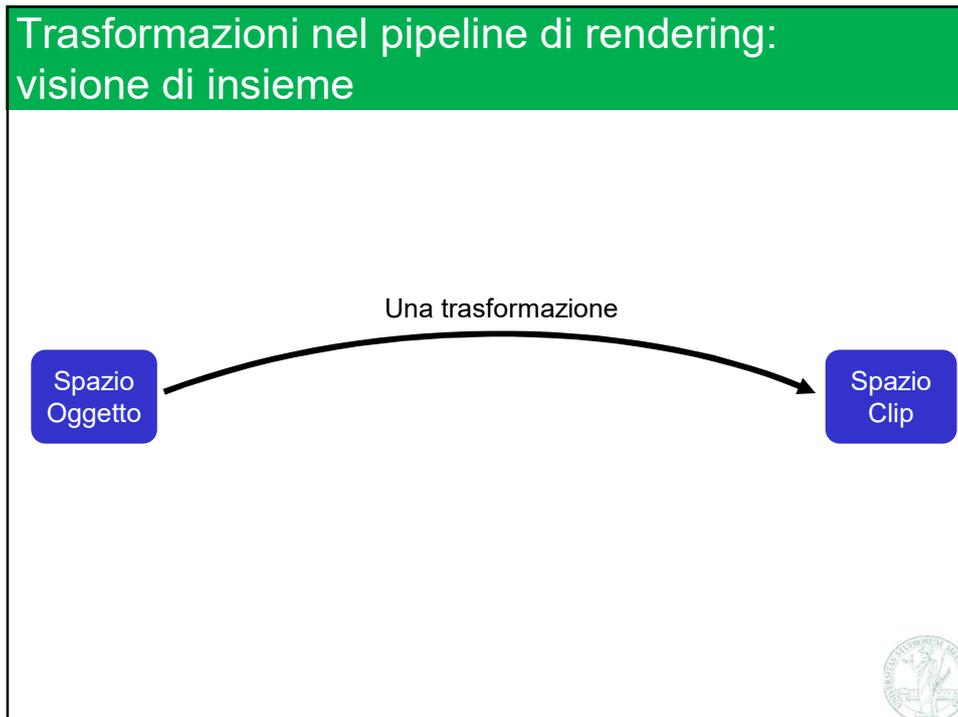
- Per ogni vertice di un modello:

coordinate in cui  
sono definiti i vertici  
dell'oggetto  
("object coords")

➔

Coordinate  
sullo schermo

2



3

### Trasformazioni nel pipeline di rendering: visione di insieme

- ✓ La trasformazione che stiamo cercando va dallo spazio in cui sono definite inizialmente le nostre mesh: lo «spazio oggetto» (iniziale) ...
- ✓ ...ad uno spazio definito sullo schermo in cui vengono proiettate: lo spazio clip (finale)
  - ⇒ Il rasterizzatore si aspetta triangoli da rasterizzare definiti nello spazio clip!
- ✓ Vediamo questi due spazi iniziali e finali
  - ⇒ Per definire uno spazio occorre definire: l'origine e 3 assi
  - ⇒ Nota: ciascuno spazio può a mano destra o sinistra,
  - ⇒ Gli spazi non sono necessariamente orto-normali



4

## Spazio oggetto

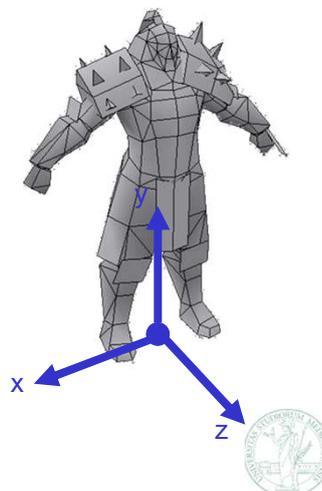
- ✓ E' il sistema di riferimento in cui è definito ciascun modello 3D
- ✓ Cioè lo spazio in cui esprimo, per esempio:
  - ⇒ In una mesh: le posizioni dei vertici, le normali (vettori unitari),
  - ⇒ I punti di controllo di un bezier patch,
  - ⇒ L'input della funzione che definisce un modello implicito,
  - ⇒ Le posizioni (intere) dei voxel in un modello voxelizzato
  - ⇒ Le coord. (x,y: implicite; z: memorizzata) dei punti in un campo di altezza
  - ⇒ etc.
- ✓ Ogni modello è provvisto del suo spazio oggetto
  - ⇒ Il suo sistema di riferimento «personale» in cui il modellatore / lo scanner etc ha definito i punti e i vettori che lo costituiscono
  - ⇒ E' il sist. di rif. in cui sono espressi, ad esempio, le coordinate dei vertici e delle normali in un file OBJ
- ✓ E' lo spazio di partenza della sequenza di trasformazioni nel rendering basato su rasterizzazione



6

## Spazio oggetto (convenzioni)

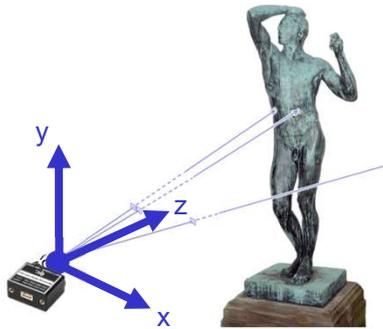
- ✓ E' scelto arbitrariamente dal modellatore (o dal software) che produce il modello 3D
- ✓ Per esempio, una convenzione per una mesh 3D che rappresenti un personaggio di un videogioco è spesso:
  - ⇒ Origine: un punto sul terreno localizzato fra i piedi del personaggio
  - ⇒ Asse X: l'asse dalla sinistra alla destra del personaggio
  - ⇒ Asse Y: l'asse dal basso all'alto del personaggio
  - ⇒ Asse Z: la direzione avanti del personaggio



7

## Spazio oggetto (convenzioni)

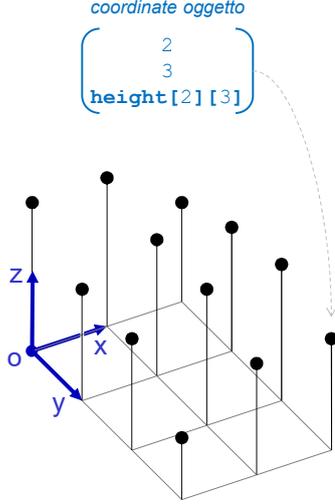
- ✓ E' scelto arbitrariamente dal modellatore (o dal software) che produce il modello 3D
- ✓ Per esempio, una convenzione per una nuvola di punti descritta da una range scan:
  - ⇒ Origine: punto di partenza dei raggi (posizione dello scanner)
  - ⇒ Asse X: l'asse dalla sinistra alla destra dello scanner
  - ⇒ Asse Y: l'asse dal basso all'alto dello scanner
  - ⇒ Asse Z: l'asse centrale del cono dei raggi che vengono mandati dello scanner




8

## Spazio oggetto (convenzioni)

- ✓ E' scelto arbitrariamente dal modellatore (o dal software) che produce il modello 3D
- ✓ Per esempio, una possibile convenzione per un campo di altezze:
  - ⇒ Origine: angolo 0,0, del array di altezze
  - ⇒ Asse X: la direzione orizzontale dell'array di altezze
  - ⇒ Asse Y: la direzione verticale dell'array di altezze
  - ⇒ Asse Z: l'altezza espressa nell'array




9

## Spazio Clip (o «Normalized Device Coordinates», «NDC»)

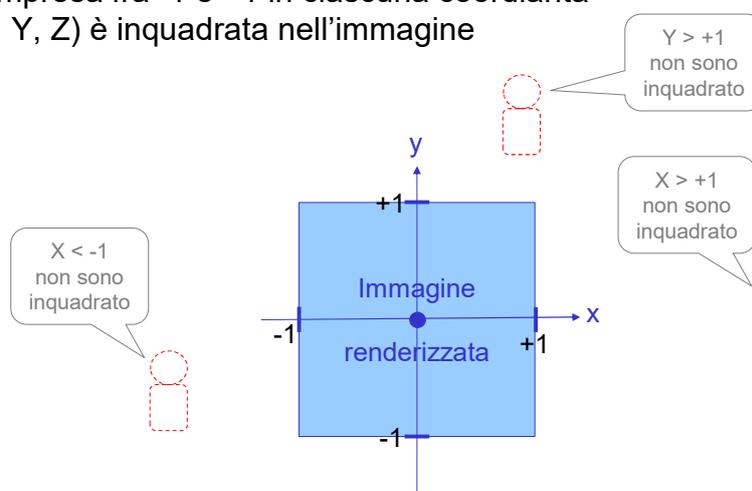
- ✓ Uno spazio allineato allo schermo / all'immagine prodotta dal rendering
  - ⇒ Origine: centro dello schermo / dell'immagine
  - ⇒ Asse X: orizzontale allo schermo / all'immagine, da sx a dx
  - ⇒ Asse Y: verticale allo schermo / all'immagine, verso l'alto
- ✓ In realtà, anche questo spazio è tridimensionale
  - ⇒ Come vedremo, questo è utile per computare correttamente gli effetti di occlusione (gli oggetti vicini coprono quelli lontani, e non viceversa)
  - ⇒ Asse Z: ortogonale allo schermo
- ✓ E' lo spazio finale in cui dobbiamo portare tutte le primitive, nel rendering basato su rasterizzazione



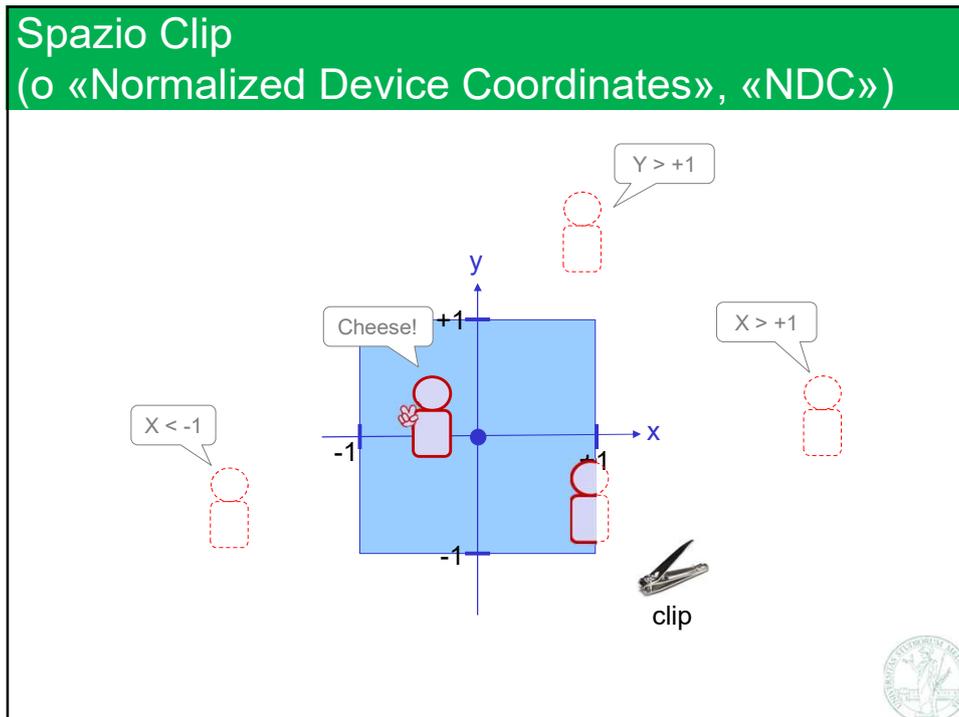
11

## Spazio Clip (Normalized Device Coordinates)

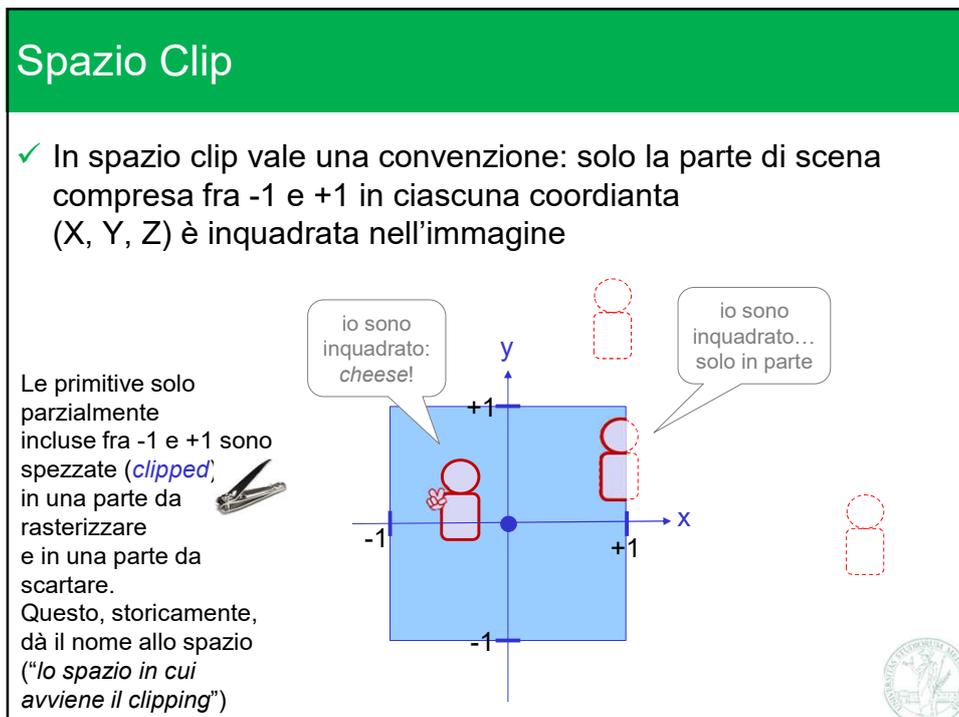
- ✓ In spazio clip vale una convenzione: solo la parte di scena compresa fra -1 e +1 in ciascuna coordinata (X, Y, Z) è inquadrata nell'immagine



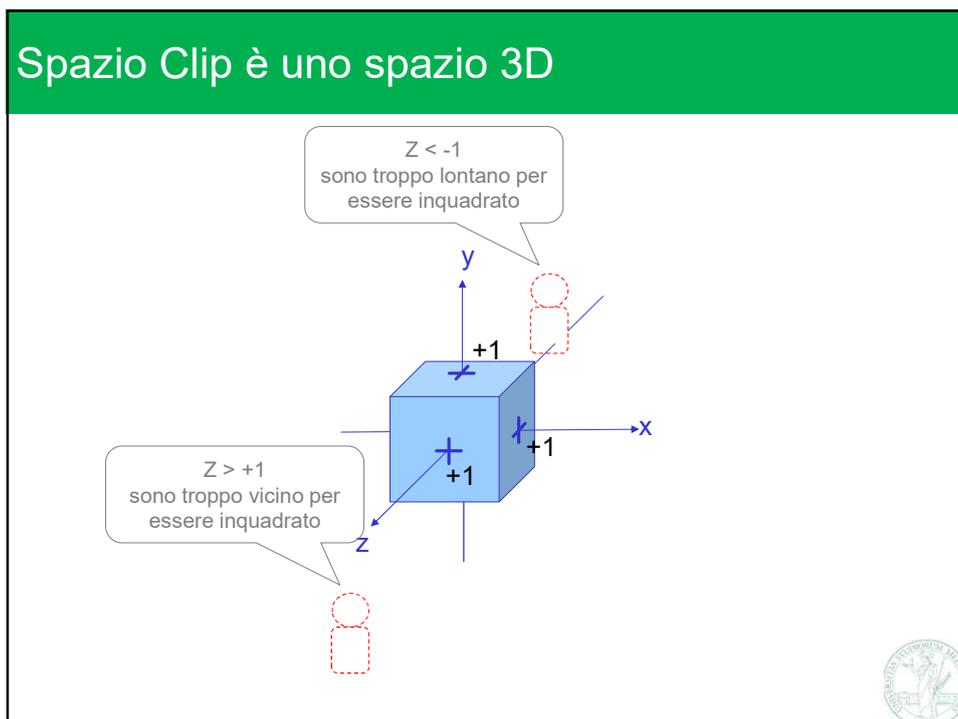
12



13



14



15

## Spazio Clip

- ✓ Nota: in spazio clip l'immagine renderizzata / lo schermo ha dimensione 2 (da -1 a +1) sia verticalmente che orizzontalmente
- ✓ Questo è indipendente dalla risoluzione in pixel o dall'aspect ratio dell'immagine rasterizzata (o dello schermo)
- ✓ Per questo motivo, le coordinate in spazio clip sono anche dette NDC (Normalized Device Coordinates) («non dipendono dal dispositivo di output, cioè dal monitor»)
- ✓ Anche nella direzione Z, vale che la scena viene renderizzata solo nella porzione che va da -1 a +1
- ✓ Anche le parti della scena troppo lontane o troppo vicine vengono scartate

16

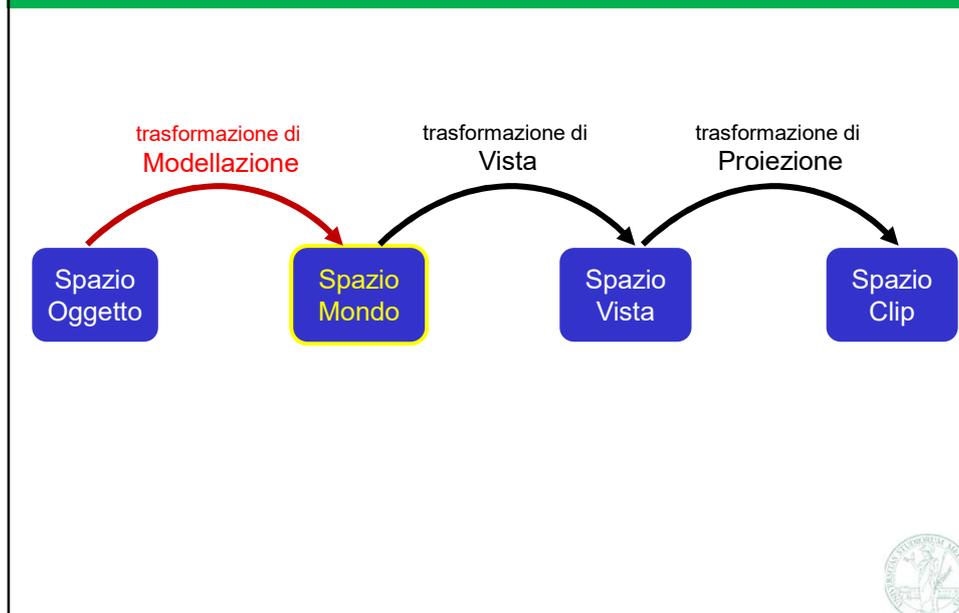
## Trasformazioni nel pipeline di rendering: visione di insieme

- ✓ La trasformazione dallo spazio oggetto (iniziale) allo spazio clip (finale) viene scomposta (almeno, concettualmente) in una piccola sequenza di passaggi dal significato standard
  - ⇒ Anche ciascun passaggio può essere visto come un cambio di sistema di riferimento (o spazio)
  - ⇒ Ogni passaggio cattura alcune scelte da effettuare durante il rendering
- ✓ Vediamo gli spazi intermedi, in sequenza
  - ⇒ Definire uno spazio occorre definire: l'origine e 3 assi
  - ⇒ Nota: ciascuno spazio può a mano destra o sinistra,
  - ⇒ Gli spazi non sono necessariamente orto-normali



17

## Trasformazioni nel pipeline di rendering: visione di insieme



18

## Spazio Mondo

- ✓ Una scena 3D è spesso costituita da *molti* modelli 3D
  - ⇒ ciascuno provvisto del suo spazio oggetto
- ✓ Lo spazio mondo è il sistema di riferimento globale della scena:
  - ⇒ è **comune a tutta la scena** (a differenza dello spazio oggetto, che è «personale» di ogni oggetto)
  - ⇒ questo spazio è definito (in modo arbitrario) da chi compone la scena (mettendo insieme, ad esempio, diverse mesh)
  - ⇒ Ad esempio, un level designer di un video-game
- ✓ Ad esempio, per una scena «salone da ballo» potremmo scegliere:
  - ⇒ Origine: punto al centro del pavimento del salone
  - ⇒ Asse X: da Est a Ovest
  - ⇒ Asse Z: da Sud a Nord
  - ⇒ Asse Y: dal pavimento verso il soffitto



19

## Trasformazione di modellazione

- ✓ Trasforma i punti / vettori da **spazio oggetto** a **spazio mondo**
- ✓ Riflette in che modo ciascun oggetto è disposto fisicamente nella scena
- ✓ Per comporre la scena, dotiamo ogni oggetto che la compone della sua trasformazione di modellazione
  - ⇒ Per es: per modellare un interno, dovremmo mettere insieme modelli 3D separati di: mobili, pareti, lampadari...
- ✓ Determinare queste trasformazioni per ciascun oggetto, è il modo di **modellare** (costruire un modello de-) **la scena**
  - ⇒ Da qui, il nome delle trasformazioni



20

### Spazio oggetto VS Spazio mondo

✓ Quattro **Object Space** in uno **Spazio Mondo**

The diagram illustrates four separate object spaces (bunny models) within a single world space. Each bunny model has its own local coordinate system with axes labeled x, y, and z. These local systems are positioned and oriented differently within a larger world coordinate system with axes labeled X, Y, and Z. The world space axes are shown in blue, while the object space axes are in black.

21

### Ad ogni oggetto, la sua trasformazione di modellazione

The diagram shows a hierarchical structure where a central node labeled «Spazio Mondo» is connected to four separate transformation nodes labeled M1, M2, M3, and M4. Each transformation node contains a bunny model, representing the application of a specific transformation matrix to the object in the world space.

22

## Determinare la trasformazione di modellazione (di ogni oggetto)

- ✓ Un modo facile ed intuitivo di determinare la trasf di modellazione è definirla come una sequenza arbitraria di trasformazioni in cascata di
    - ⇒ rotazione,
    - ⇒ traslazione,
    - ⇒ e scaling
- matrix multiplication!
- matrici
- ✓ che riflettono
    - ⇒ l'orientamento,
    - ⇒ la posizione,
    - ⇒ e la dimensione
- desiderata per quell'oggetto nella scena  
(cioè in spazio mondo)
- 

23

## Trasformazione di modellazione e animazioni

- ✓ In una computer animation, posso spostare un oggetto nella scena, ridefinendo in ogni fotogramma la sua **trasformazione di modellazione**
    - ⇒ Cambiando così in un colpo solo le coordinate dei suoi punti e vettori in spazio mondo, ma **non** in spazio oggetto!
    - ⇒ Quindi (crucialmente): la struttura dati della mesh – che è memorizzata in VRAM in spazio oggetto - non cambia durante l'animazione!
- 

24

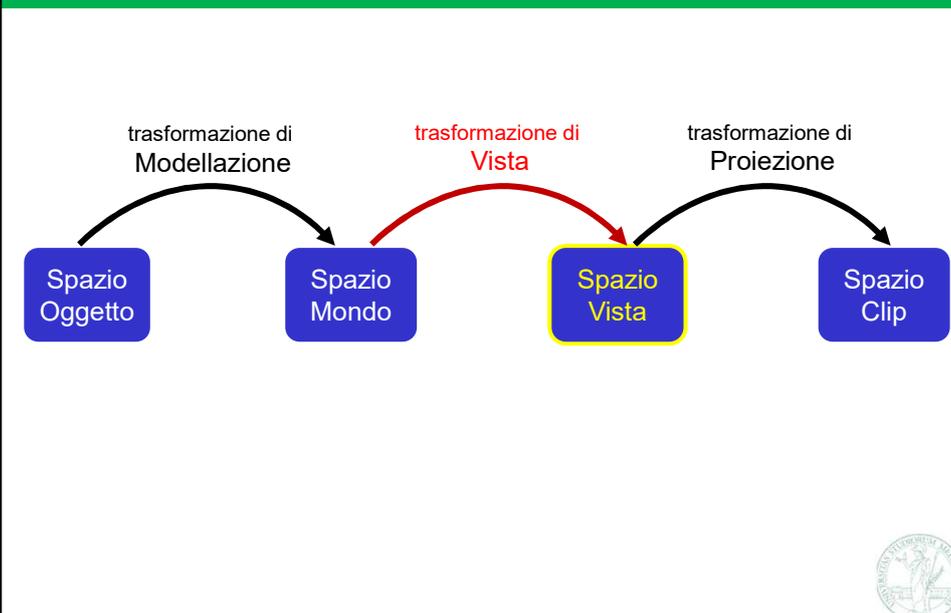
## Trasformazione di modellazione e multi-instancing

- ✓ Disegnando, in uno stesso screen buffer, una stessa mesh più volte, ogni volta con una diversa matrice di modellazione, (cioè: disegnando diverse *istanze* dello stesso oggetto), ottengo diverse copie dello stesso oggetto nella scena
  - ⇒ Nota: ogni istanza ha punti e vettori di coordinate diverse, in spazio mondo, ma identiche, in spazio oggetto
  - ⇒ Mantenendo in memoria una sola mesh (con coordinate oggetto univoche), mostro diversi oggetti a schermo (ciascuno con coordinate mondo diverse)
- ✓ Ad esempio, possono essere istanze di una stessa mesh:
  - ⇒ tutte le sedie della mia scena di interno
  - ⇒ tutte le forchette sul tavolo
  - ⇒ le 4 ruote del modello di una macchina
  - ⇒ Etc.



25

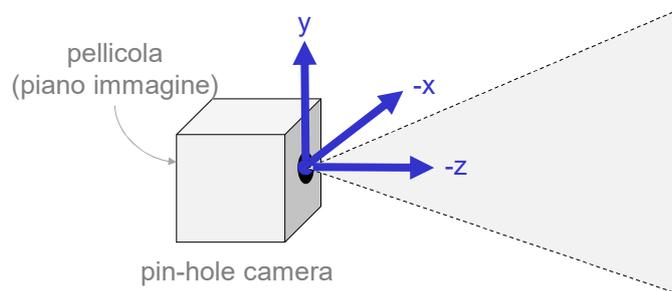
## Trasformazioni nel pipeline di rendering: visione di insieme



26

## Spazio Vista

- ✓ Lo spazio vista è posizionato in modo standard rispetto alla pin-hole camera che inquadra la scena
  - ⇒ Origine: point of view (l'hole della pin-hole camera)
  - ⇒ Asse X: asse orizzontale della camera
  - ⇒ Asse Y: dal basso all'alto della camera
  - ⇒ Asse Z: dal davanti al dietro della camera

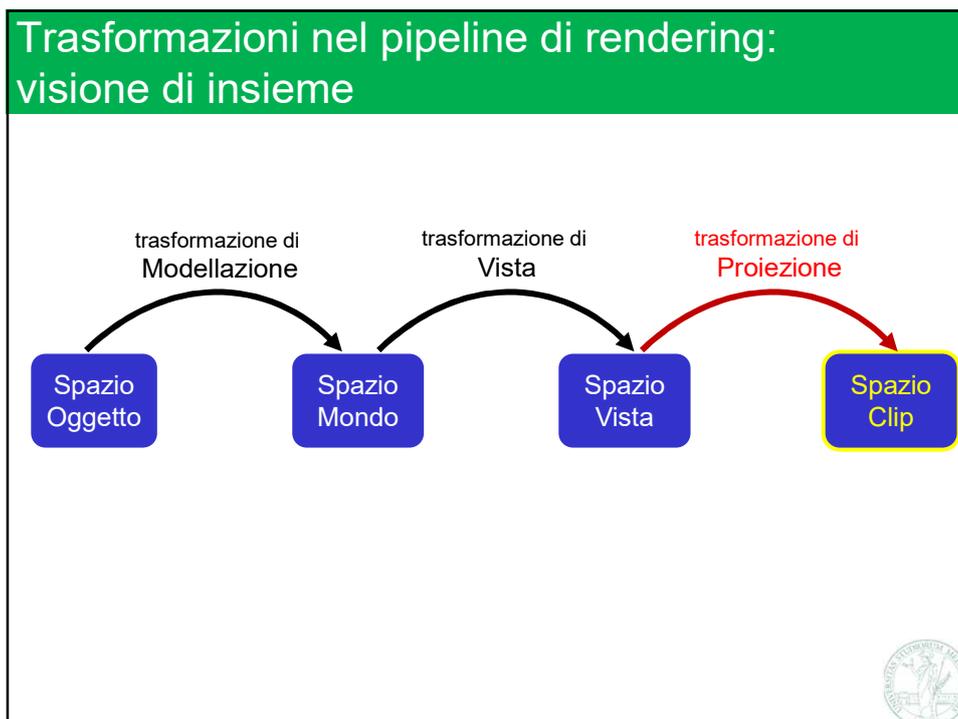


28

## Trasformazione di vista

- ✓ Trasforma i punti / vettori da spazio mondo a spazio vista
- ✓ Dipende la posizione e l'orientamento della camera che inquadra la scena
  - ⇒ Detti i **parametri estrinseci** della camera
- ✓ Se la telecamera si sposta, cambia la trasformazione (la matrice) di vista
  - ⇒ Questo causerà ovviamente uno spostamento nel piano immagine degli oggetti inquadrati, anche se la loro posizione nella scena (cioè in spazio mondo!) non è cambiata.
- ✓ Tipicamente è una trasformazione rigida: rotazione + traslazione

29

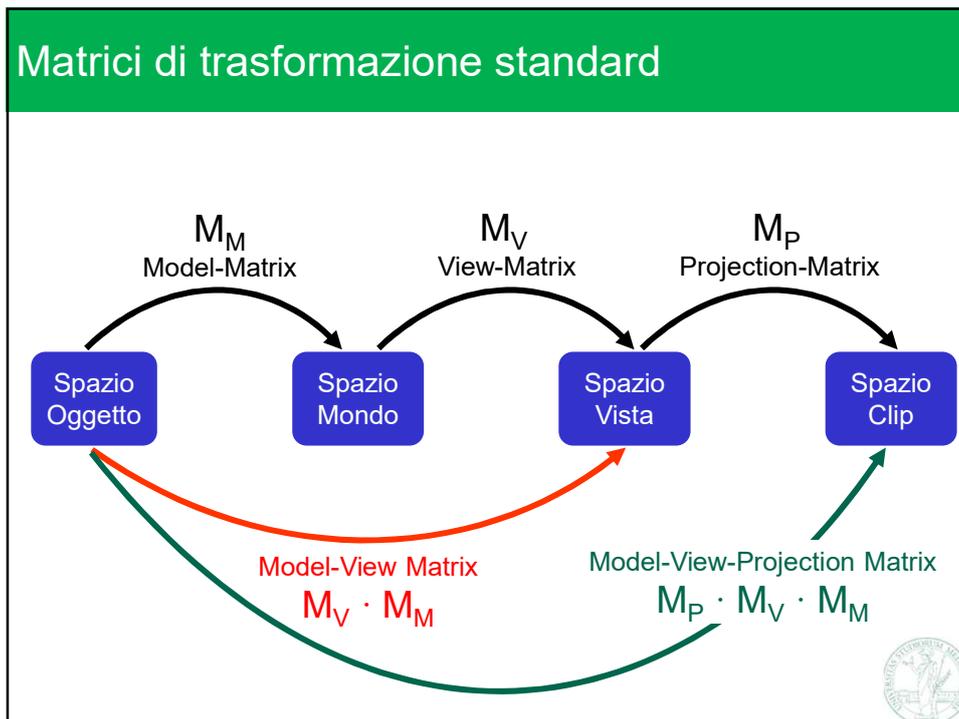


32

### Trasformazione di proiezione

- ✓ Trasforma da Spazio Vista a Spazio Clip
- ✓ E' determinata dalle caratteristiche *interne* della camera
  - ⇒ come le dimensioni del box della camera
  - ⇒ la posizione del «buco di spillo» della pin-hole camera
  - ⇒ soprattutto dalla lunghezza focale (determina se grandangolo, teleobiettivo...)
  - ⇒ Questi sono detti i **parametri INTRINSECI** della camera
- ✓ Responsabile anche degli effetti prospettici
  - ⇒ Oggetti più vicini appaiono più grandi.
  - ⇒ Cioè:
    - oggetti che hanno la stessa dimensione in spazio mondo o vista, ma sono ad una Z diversa in spazio vista, diverranno di dimensioni diverse in spazio clip (tanto maggiori, quanto più vicina allo 0 è la Z in spazio vista).
- ✓ Vedremo nelle prossime lezioni

33



35

### Note terminologiche: alcuni sinonimi (sommario)

	...anche detto/a...
matrice che effettua una trasformazione affine $X$	“matrice di $X$ ”
<i>per es:</i> matrice che effettua la trasformazione di traslazione $(x,y,z)$	“matrice di traslazione $(x,y,z)$ ” (“ <i>translation matrix</i> ”)
<i>per es:</i> matrice che effettua la trasformazione di vista	“Matrice di Vista” (“ <i>View Matrix</i> ”)
Sistema di Riferimento	“spazio” ( <i>space</i> )
<i>per es:</i> Sistema di Riferimento Mondo	“spazio mondo” ( <i>world space</i> )
Spazio <b>Oggetto</b> dell’oggetto $X$	“spazio $X$ ”
<i>per es:</i> Spazio <b>Oggetto</b> dell’oggetto gatto	“spazio gatto”
Coordinate espresse in uno spazio $X$	“coordinate $X$ ”
<i>per es:</i> Coordinate espresse nello Spazio Oggetto	“coordinate oggetto” (“ <i>object coordinates</i> ”)
Coordinate <b>Clip</b> (coordinate espresse nello spazio <b>Clip</b> )	“ <i>Normalized Device Coordinates (NDC)</i> ”

36

## Domande (esercizio)

- ✓ Quali sono le coordinate omogenee:
  - ⇒ Di un pt che appare in mezzo dallo schermo, in spazio clip?
  - ⇒ Della posizione del POV, in spazio vista?
  - ⇒ Del un punto in alto a destra del monitor, in spazio Clip?
  - ⇒ Della direzione di vista, in spazio vista?
- ✓ Date le matrici di Modellazione  $M$ , Vista  $V$ , e proiezione  $P$ , come si trovano le coord omogenee...
  - ⇒ Dell'origine dello spazio oggetto, in spazio mondo?
  - ⇒ Dell'origine dello spazio mondo, in spazio oggetto?
  - ⇒ Della direzione di vista, in spazio oggetto?
  - ⇒ Della posizione del POV, in spazio mondo?
  - ⇒ Di un punto che appare in mezzo allo schermo, in spazio oggetto?



37

## Domande (esercizio)

(usando le stesse matrici del lucido precedente)

- ✓ Ipotizziamo che in spazio mondo l'asse delle  $Y$  sia orientato verso il cielo:
- ✓ Quali sono le coordinate omogenee:
  - ⇒ Del vettore «direzione di gravità» (la direzione unitaria di caduta degli oggetti), in spazio mondo?
  - ⇒ Dello stesso vettore, in spazio oggetto? (nota: questo determina in che direzione quell'oggetto percepisce la forza di gravità, nel proprio sistema di rif.)
- ✓ Se la mano di un personaggio è in un dato punto  $\mathbf{p}$ , (nel suo spazio oggetto!) come posso determinare se tale mano è inquadrata oppure no nello schermo?



38