

Marco Tarini - Computer Graphics 2025/2026  
Università degli Studi di Milano

### La sequenza di trasformazioni nel rendering

Oggetto      Mondo      Vista      Clip

1

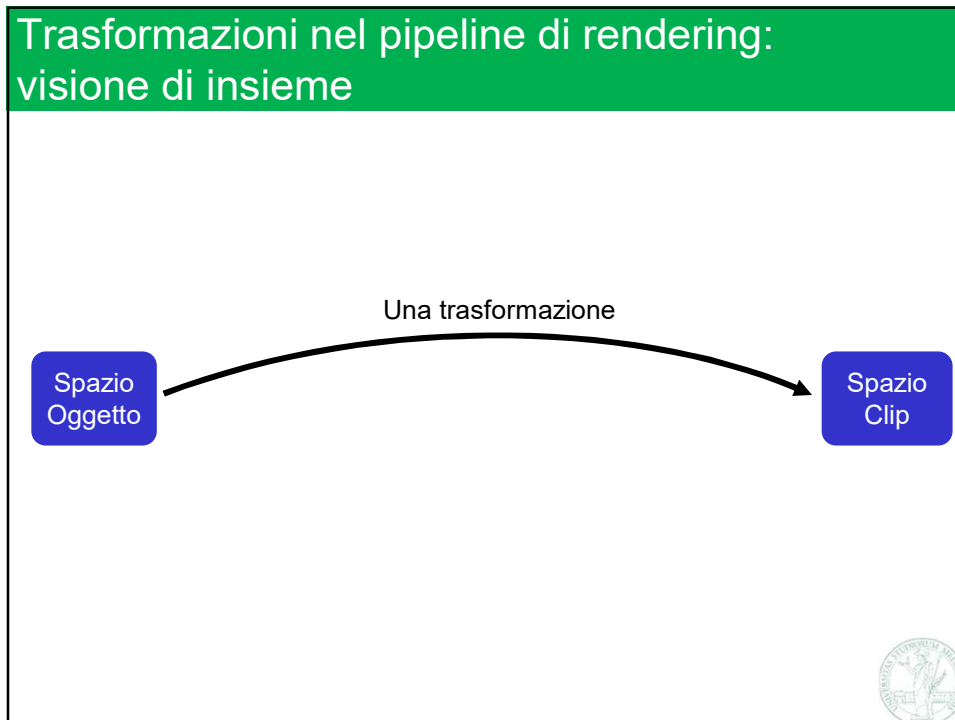
### La transform usata duante il rendering (con approccio rasterization)

- 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 modulo di rasterizzazione si aspetta triangoli da rasterizzare definiti in spazio clip!
- ✓ Vediamo questi due spazi iniziali e finali
  - ⇒ Per definire uno spazio occorre definire: l'origine e 3 assi
  - ⇒ Nota: ciascuno spazio può essere «a mano destra» o «a mano sinistra»
  - ⇒ Gli spazi non sono necessariamente orto-normali



4

## Spazio oggetto

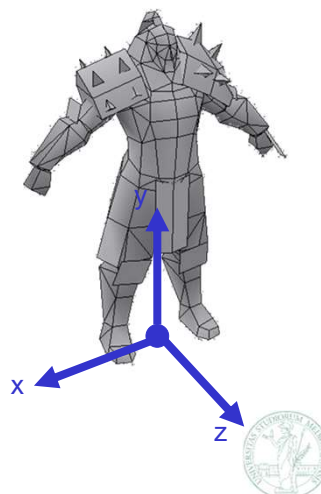
- ✓ E' il sistema di riferimento in cui è definito ciascun modello 3D
- ✓ Cioè lo spazio in cui esprimo, per esempio:
  - ⇒ In una mesh: le posizioni dei vertici, le normali (vettori unitari),
  - ⇒ I punti di controllo di un bezier patch,
  - ⇒ L'input della funzione che definisce un modello implicito,
  - ⇒ Le posizioni (intere) dei voxel in un modello voxellizzato
  - ⇒ 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 dei vettori normali dentro ad un file mesh (in formato OBJ o OFF o ...)
- ✓ 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 Clip (le coordinate in questo spazio sono dette «Normalized Device Coordinates», o «NDC»)

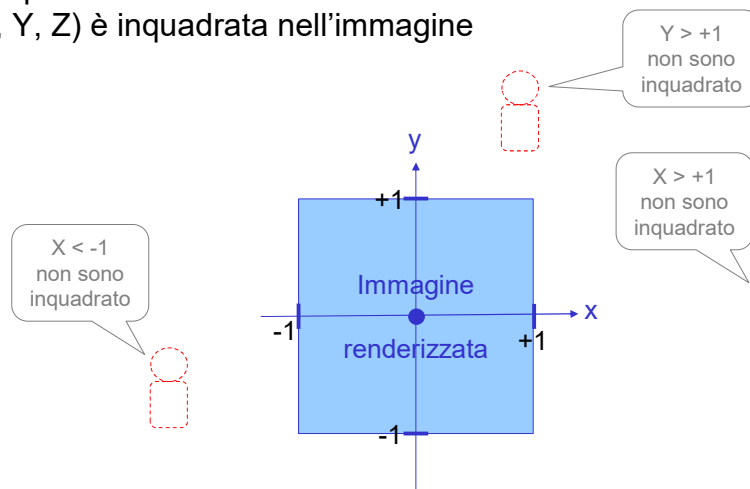
- ✓ Uno spazio allineato allo schermo o comunque all'immagine prodotta dal rendering
  - ⇒ Origine: centro dello schermo / dell'immagine
  - ⇒ Asse X: orizzontale allo schermo / all'immagine, da sx a dx
  - ⇒ Asse Y: verticale allo schermo / all'immagine, verso l'alto
- ✓ In realtà, anche questo spazio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




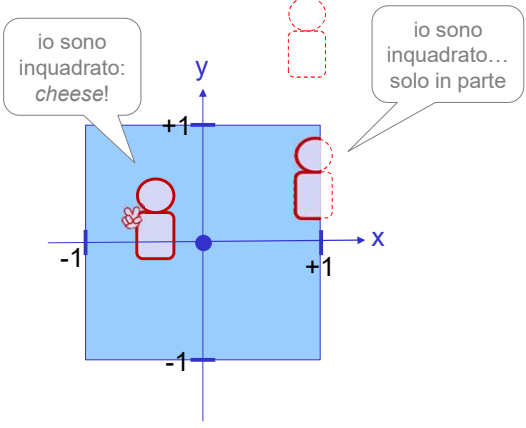
12


## Spazio Clip

✓ 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

Le primitive solo parzialmente incluse fra -1 e +1 sono spezzate (*clipped*), in una parte da rasterizzare e in una parte da scartare. Questo, storicamente, dà il nome allo spazio ("lo spazio in cui avviene il clipping")



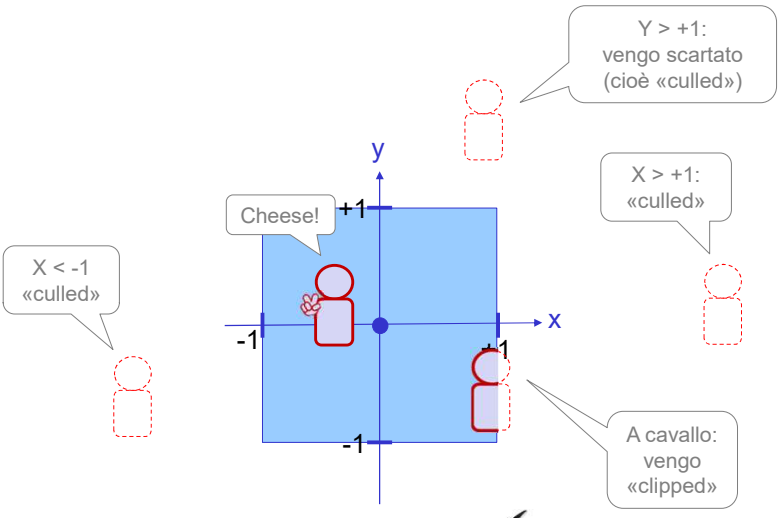





13

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

$X < -1$   
«culled»



this is a clipper



14

## Anche lo Spazio Clip è uno spazio 3D

Lo spazio che contiene le cose visualizzate è un **volume**

$Z > +1$   
sono troppo lontano per essere inquadrato

$Z < -1$   
sono troppo vicino per essere inquadrato

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 (il *device*) 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

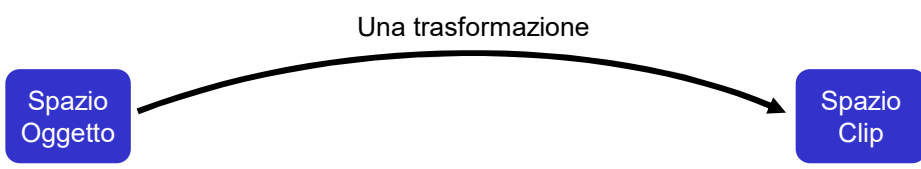
### Trasformazioni nel pipeline di rendering: visione di insieme

- ✓ La trasformazione dallo spazio oggetto (iniziale) allo spazio clip (finale) viene scomposta (almeno, concettualmente) in una piccola sequenza di passaggi (di trasformazioni spaziali) 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



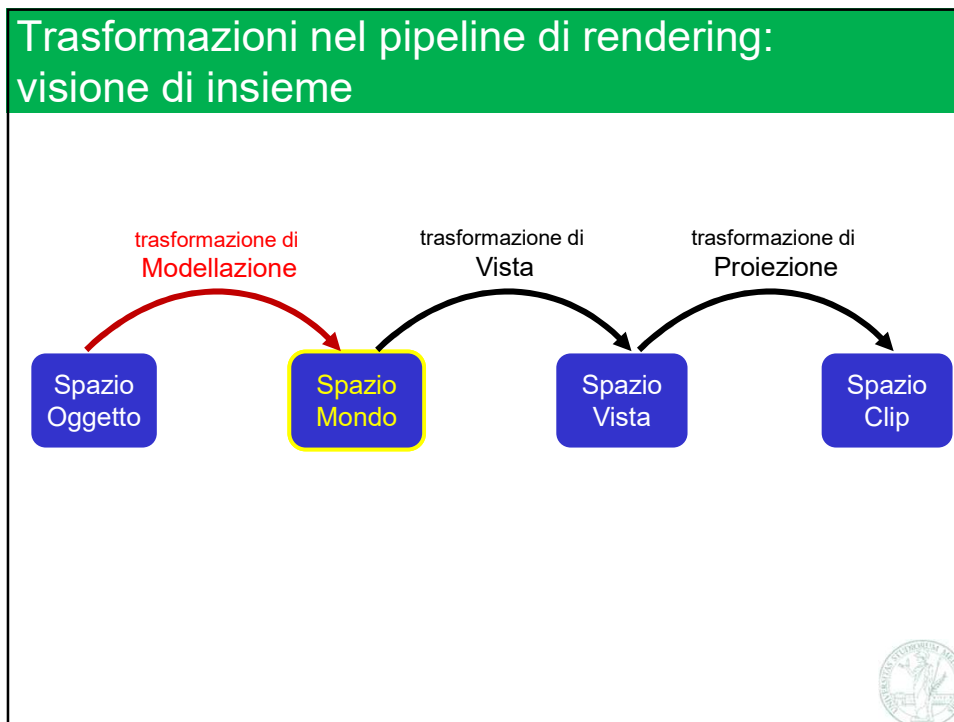
Una trasformazione

Spazio Oggetto

Spazio Clip



18



19

### 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 (arrangiando insieme diversi modelli 3D)
  - ⇒ 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

In basso a destra del contenuto è presente il logo dell'Università degli Studi di Milano.

20

## Trasformazione di modellazione

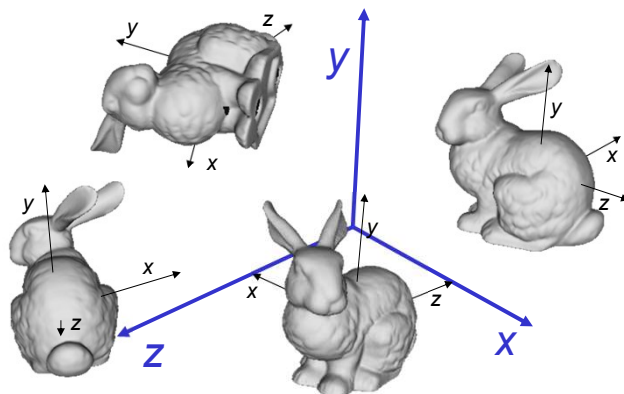
- ✓ Trasforma i punti / vettori da **spazio oggetto** a **spazio mondo**
- ✓ Riflette in che modo ciascun oggetto è disposto fisicamente nella scena
- ✓ Per comporre la scena, dotiamo ogni oggetto che la compone della sua trasformazione di modellazione
  - ⇒ Per es: per modellare un interno, dovremmo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



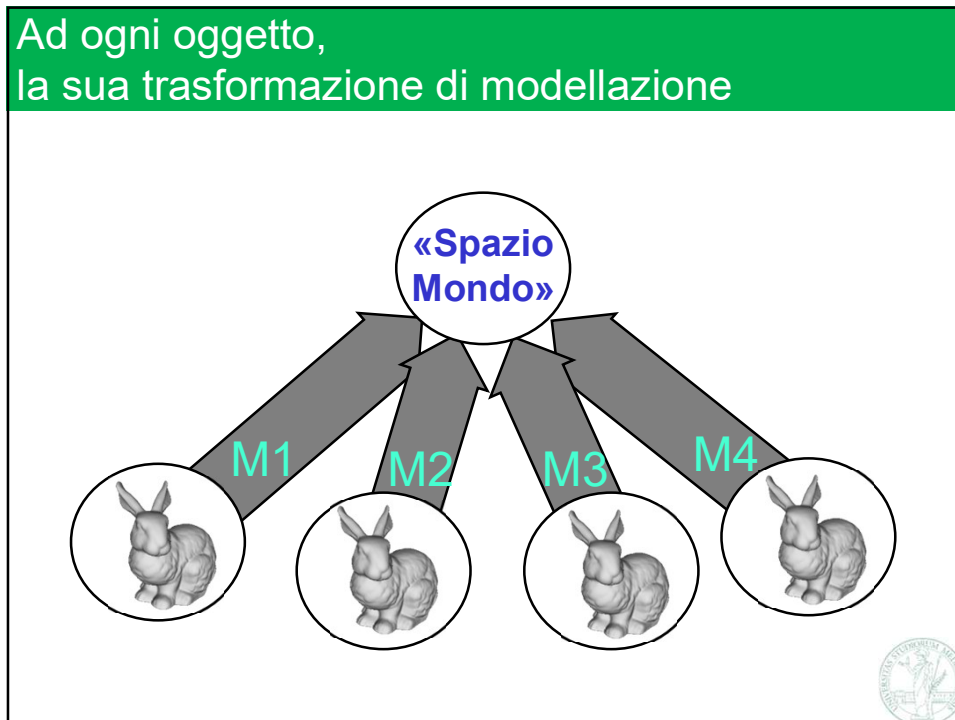
21

## Spazio oggetto VS Spazio mondo

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



22



23

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
- ✓ che riflettono
  - ⇒ l'orientamento,
  - ⇒ la posizione,
  - ⇒ e la dimensione

desiderata per quell'oggetto nella scena  
(cioè in spazio mondo)

matrix multiplication!

matrici

The slide explains how to determine the modeling transformation for an object. It lists two main methods: defining it as a sequence of transformations (rotation, translation, scaling) or as properties that reflect the object's orientation, position, and dimension. Both methods lead to the use of matrices. A note indicates that matrix multiplication is used to combine these transformations.

24

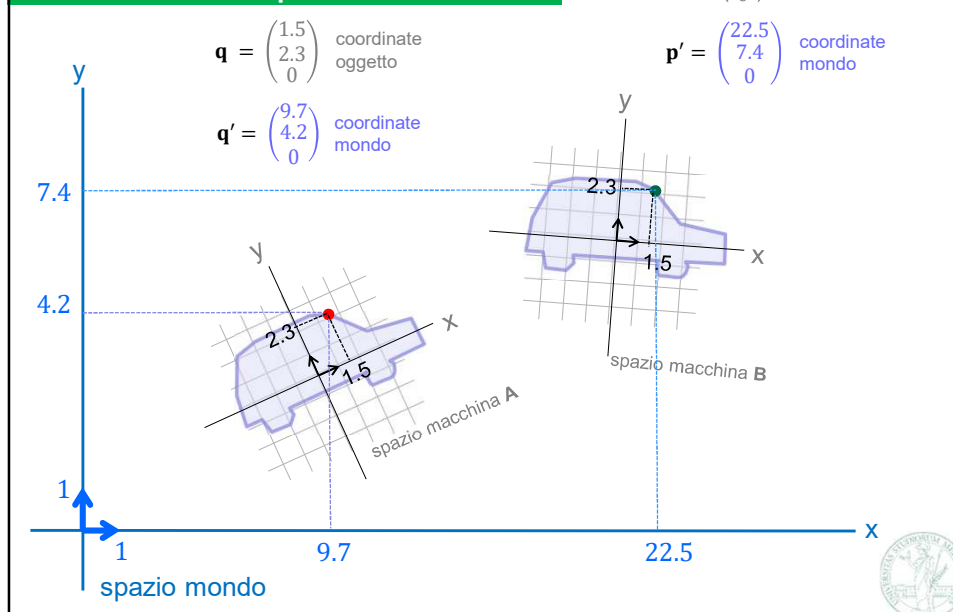
## Trasformazione di modellazione e animazioni

- ✓ 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!



25

## Due istanze nello stesso spazio mondo



28

## Due istanze nello stesso spazio mondo

- ✓ Nota che i due punti indicati  $\mathbf{p}$  e  $\mathbf{q}$ , nelle due automobili, hanno le stesse coordinate in **spazio oggetto** ...
  - ⇒ sono lo stesso vertice della mesh
- ma hanno coordinate diverse  $\mathbf{p}'$  e  $\mathbf{q}'$  in **spazio mondo**
  - ⇒ perché le due istanze dell'automobile sono associate a due **matrici di modellazione** diverse



29

## 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 è definito da punti e vettori con le stesse coordinate in spazio oggetto, ma con coordinate diverse in spazio mondo
  - ⇒ 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
  - ⇒ ...

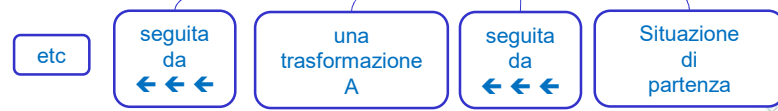


30

## Costruire la Matrice di Modellazione

- ✓ Come ogni matrice di trasformazione spaziale affine, posso vedere (e costruire!) la matrice di modellazione  $\mathbf{M}$  in due modi equivalenti:
  - ⇒ Come una descrizione dello **spazio oggetto**: la sua origine e i suoi assi, espressi in **spazio mondo**, costituiscono le colonne di  $\mathbf{M}$
  - ⇒ Una sequenza di trasformazioni spaziali  $A, B, C \dots$  (come traslazioni, rotazioni, etc) che modificano una situazione "di partenza" in cui spazio oggetto = spazio mondo:

$$\mathbf{M} = \dots \cdot \mathbf{T}_C \cdot \mathbf{T}_B \cdot \mathbf{T}_A \cdot \mathbf{I}$$



31

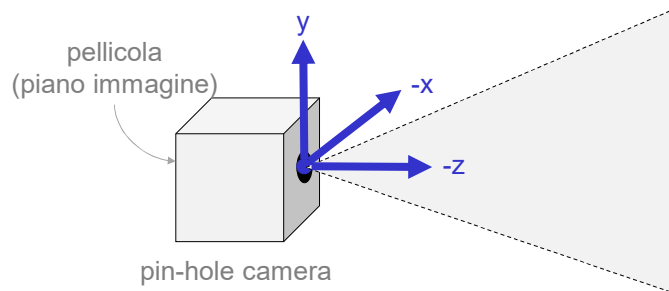
## Trasformazioni nel pipeline di rendering: visione di insieme



32

## 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



34

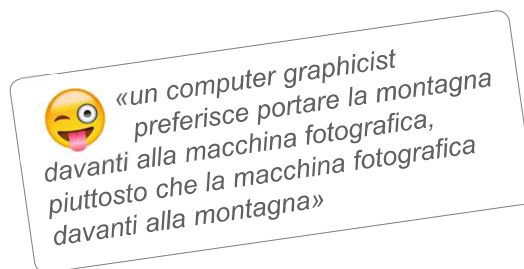
## Trasformazione di vista

- ✓ 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

35

## 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



36

## 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



37

## Osservazione

Lo spazio vista è lo spazio oggetto dell'oggetto  
«macchina fotografica virtuale»

Matrice di Vista V:  
da spazio mondo a spazio Vista  
(cioè lo spazio oggetto dell'oggetto *camera*)

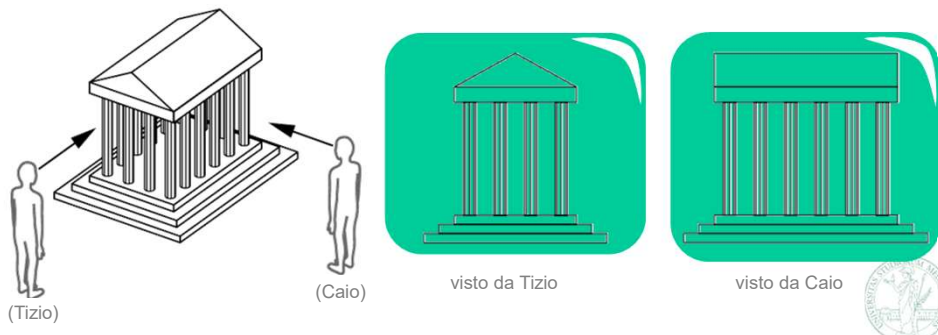
la trasformazione di **vista**  
è *l'inversa*  
della trasformazione di **modellazione** dell'oggetto  
«macchina fotografica virtuale»



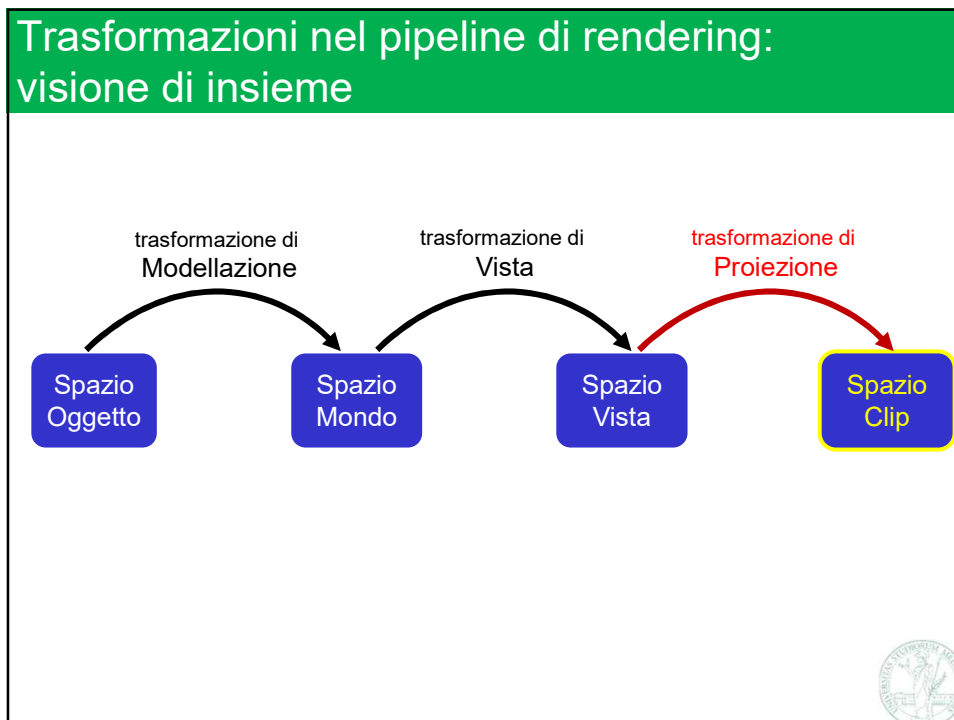
38

## Trasformazione di vista

- dipende da dov'è  
e da come è orientata la "**camera**" (la macchina fotografica)
  - Cioè dai suoi "parametri estrinseci"



39



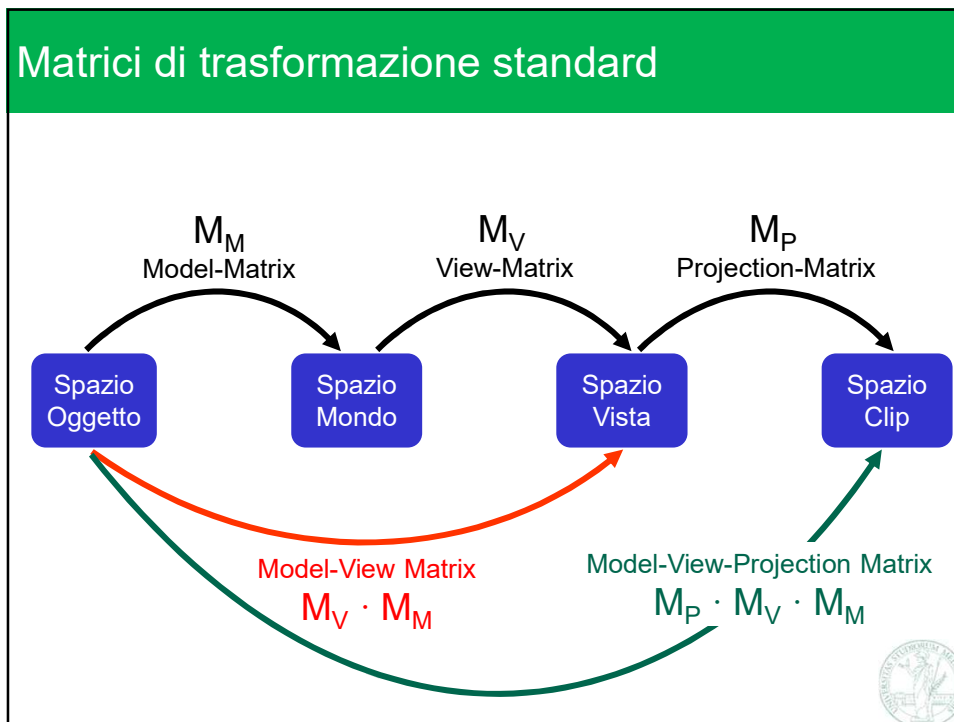
41

### Trasformazione di proiezione

- ✓ Trasforma da Spazio Vista a Spazio Clip
- ✓ E' determinata dalle caratteristiche *interne* della camera
  - ⇒ come le dimensioni del box della pin-hole camera
  - ⇒ soprattutto dalla lunghezza focale, cioè la profondità della pin-hole camera (determina se grandangolo, o tele-obiettivo etc...)
  - ⇒ la posizione del «buco di spillo» della pin-hole camera
  - ⇒ questi sono detti i **parametri INTRINSECI** della camera
- ✓ Responsabile anche degli effetti prospettici
  - ⇒ Il fenomeno in cui gli oggetti più vicini ci 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



42



44

### 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> ”

45

## Una metafora... cinematografica

- ✓ Matrice di **modellazione** (di ogni oggetto):  
scelta dello *scenografo*
  - ⇒ «Dove posiziono questo candelabro nella scena?»
  - ⇒ «Come lo oriento? Adagiato sul tavolo o dritto?»
- ✓ Matrice di **vista**:  
scelta del *regista*
  - ⇒ «Da quale punto di vista inquadro la scena?»
  - ⇒ «Cioè: come posiziono e oriento la telecamera?»
- ✓ Matrice di **proiezione**:  
scelta del *direttore della fotografia*
  - ⇒ «Uso uno zoom, o un grandangolo, o altro?»
  - ⇒ «Filmo in 4:3 o in 16:9?»

Tutti e tre concorrono a determinare dove appare, nel fotogramma, il candelabro (e se viene inquadrato o no)



46

## 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?



47

## 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 il piede di un personaggio è in un dato punto  $p$ , (nel suo spazio oggetto!) come posso determinare se tale piede è inquadrato oppure no nello schermo?

48

## Domande (esercizio)

Faccio una foto col cellulare ad un gatto (una mesh di un gatto).

Siano date le seguenti matrici:

A = matrice di modellazione-vista,

B = matrice di vista

C = inversa di A

D = inversa di B

E = matrice di model-view-projection

✓ Come determineresti...

- ⇒ La distanza dal PIANO del cellulare (che contiene l'obiettivo) dell'ORIGINE dello spazio gatto
- ⇒ Nota: è la z dello spazio vista! Anzi: il suo opposto (-z), se adottiamo la nostra convenzione sul verso dell'asse Z in spazio vista (verso l'osservatore)
- ⇒ La pos della macchina fotografica (o meglio, del punto di fuoco) in spazio mondo
- ⇒ Nota: in spazio vista, sarebbe l'origine!
- ⇒ Dove finisce in spazio CLIP l'occhio del gatto, che, nella geometria della mesh, è un vertice in posizione nel punto 3D  $p$

49