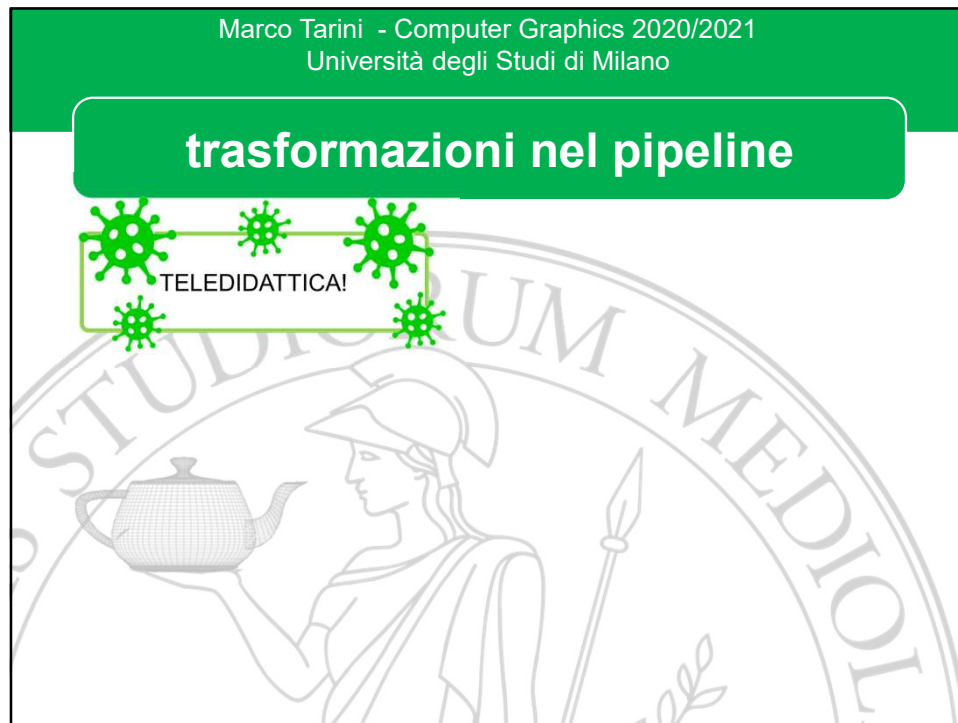




1

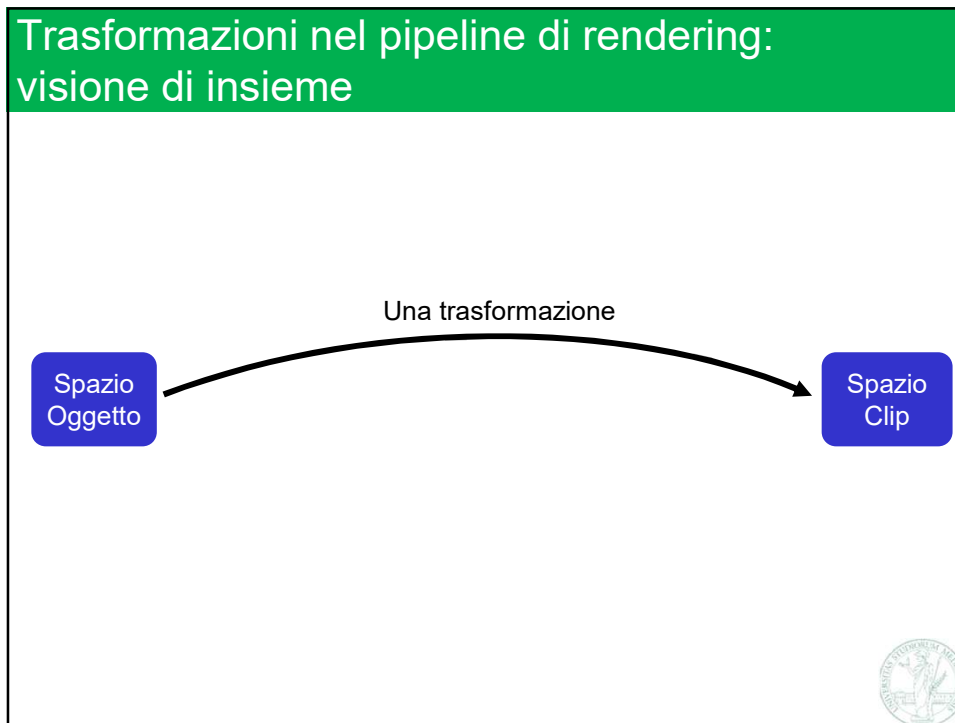
Transform durante il rendering (con approccio rasterization)

- Per ogni vertice di un modello:

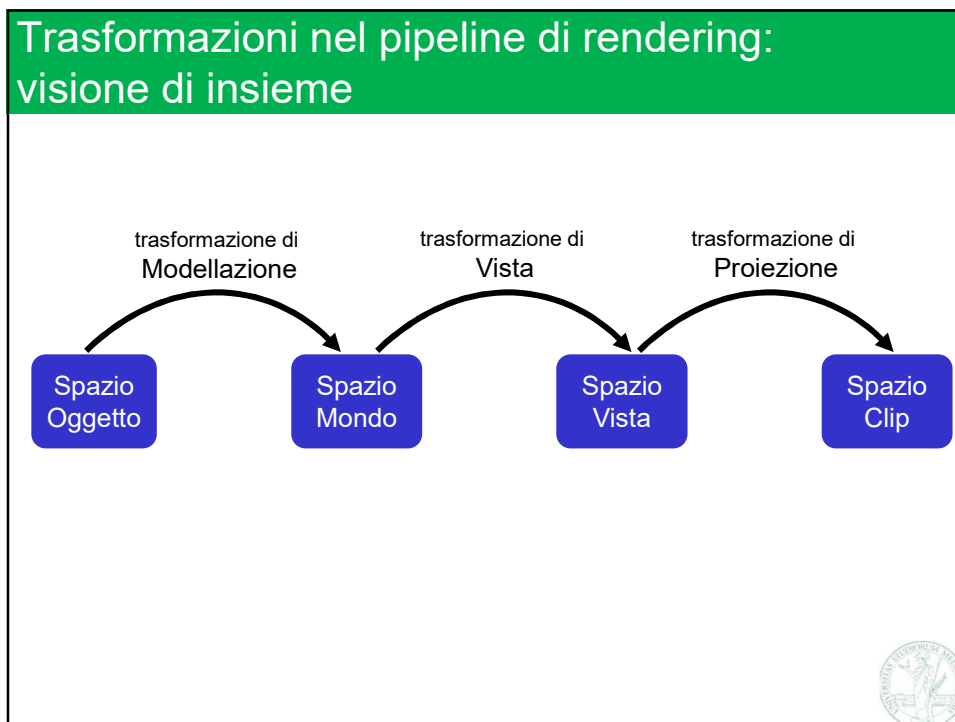
coordinate in cui
sono definiti i vertici
dell'oggetto
("object coords")

Coordinates
sullo schermo

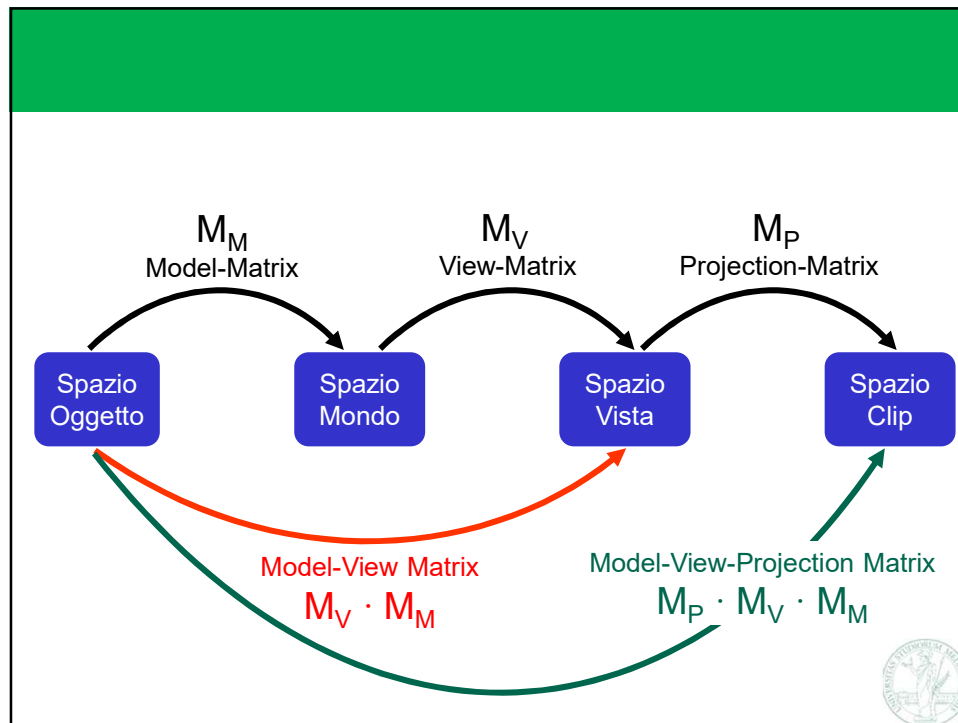
2



3



14



15

Trasformazioni nel pipeline di rendering: visione di insieme

- ✓ La trasformazione a cui è soggetto ogni punto e vettore che costituisce il modello 3D da renderizzare viene scomposta in una piccola sequenza di passaggi intermedi dal significato standard
 - ⇒ Ciascun passaggio è una sotto-trasformazione che può essere vista come un cambio di sistema di riferimento (o spazio)
 - ⇒ Ogni passaggio cattura alcune scelte da effettuare durante il rendering
 - ✓ Vediamo questi spazi 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
- A small circular logo is visible in the bottom right corner of the text area.

16

Spazio oggetto

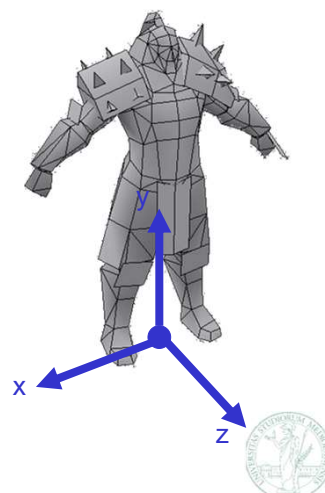
- ✓ E' il sistema di riferimento in cui è definito ciascun modello 3D
- ✓ Cioè lo spazio in cui esprimo, per esempio:
 - ⇒ Le posizioni dei vertici di una mesh, le sue normali (vettori unitari), i punti di controllo di un bezier patch, l'input della funzione che definisce un modello implicito, etc.
- ✓ Ogni modello è provvisto del suo spazio oggetto.
- ✓ E' lo spazio di partenza della sequenza di trasformazioni



17

Spazio oggetto

- ✓ Scelta arbitraria del modellatore (o del software) che produce il modello 3D
- ✓ Per esempio, una convenzione per una mesh 3D che rappresenti un personaggio di un videogioco può essere:
 - ⇒ 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: l'asse dal verso la direzione avanti **del personaggio**



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)
 - ⇒ anche questo spazio è definito in modo arbitrario
(da chi compone la scena)
- ✓ Ad esempio, per una scena «salone da ballo»:
 - ⇒ Origine: punto al centro del pavimento del salone
 - ⇒ Asse X: da Est a Ovest
 - ⇒ Asse Y: da Sud a Nord
 - ⇒ Asse Z: dal pavimento verso il soffitto



19

Trasformazione di modellazione

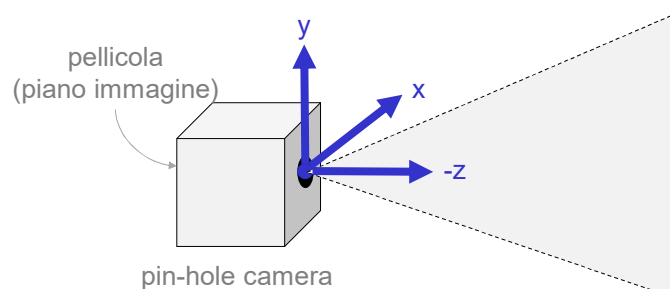
- ✓ Trasforma i punti / vettori da **spazio oggetto** a **spazio mondo**
- ✓ Riflette in che modo ciascun oggetto è disposto fisicamente nella scena
- ✓ Nota: ogni oggetto possiede la sua trasformazione di modellazione
- ✓ Quando un oggetto si sposta nella scena, cambia il suo spazio oggetto e quindi la sua trasformazione di modellazione



20

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



21

Trasformazione di vista

- ✓ Trasforma i punti / vettori da spazio mondo a spazio vista
- ✓ Riflette 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 (quindi la loro model-matrix) non è cambiata.
 - ⇒ Sarà invece cambiata la matrice di ModelView, che combina la model e la view matrix

22

Spazio Clip

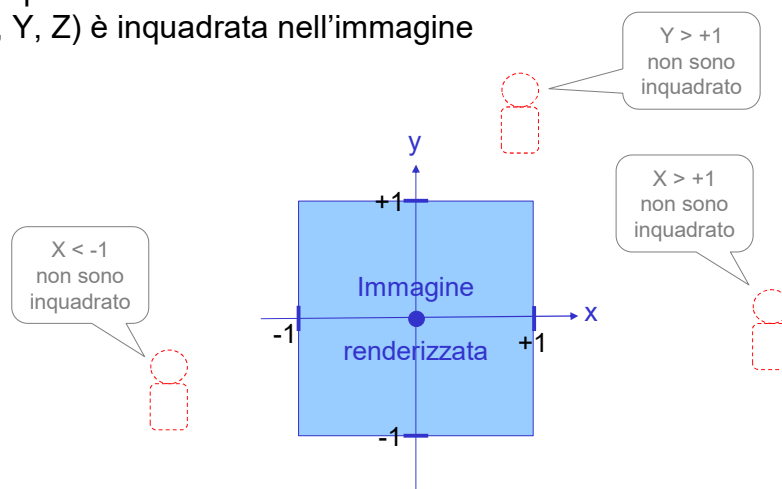
- ✓ Uno spazio allineato allo schermo / all'immagine prodotta dal rendering
 - ⇒ Origine: centro dello schermo / dell'immagine
 - ⇒ Asse X: orizzontale allo schermo / all'immagine, da sx a dx
 - ⇒ Asse Y: verticale allo schermo / all'immagine, verso l'alto
- ✓ In realtà, anche questo spazio è tridimensionale
 - ⇒ 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



23

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

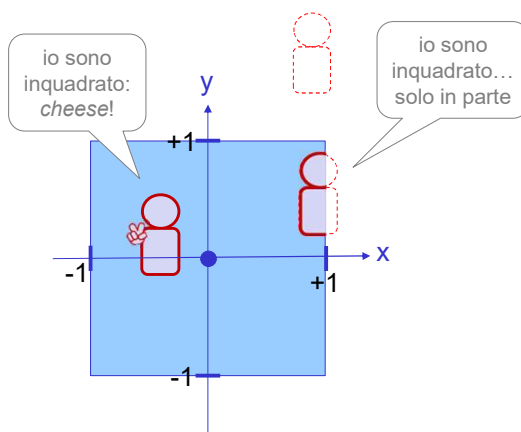


24

Spazio Clip

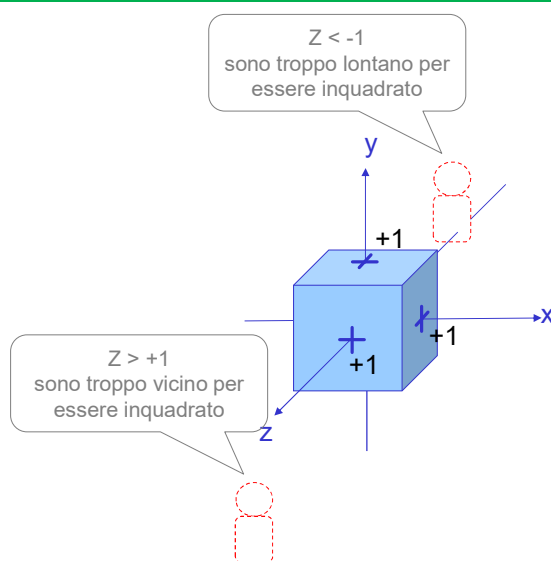
- ✓ In spazio clip vale una convenzione: solo la parte di scena compresa fra -1 e +1 in ciascuna coordinata (X, Y, Z) è inquadrata nell'immagine

Gli modelli solo parzialmente inclusi fra -1 e +1 sono spezzati (*clipped*) in una parte da rasterizzare e in una parte non rasterizzata. Questo, storicamente, dà il nome allo spazio ("lo spazio in cui avviene il clipping")



25

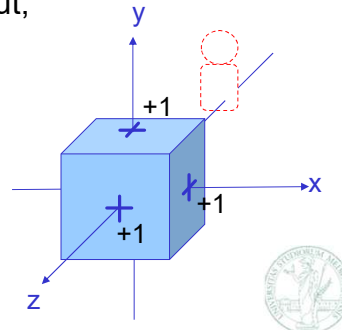
Spazio Clip è uno spazio 3D



26

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



27

Trasformazione di proiezione

- ✓ Trasforma lo Spazio Vista in Spazio Clip
- ✓ E' determinata da 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
- ✓ Ricordare: le trasformazioni si possono combinare!
 - ⇒ Es: Model-View-Projection matrix («MVP» matrix): da spazio oggetto a spazio clip
 - ⇒ Si ottiene come :
$$\text{Matrice di Proiezione} \cdot \text{Matrice di Vista} \cdot \text{Matrice di Modellaz.}$$



28

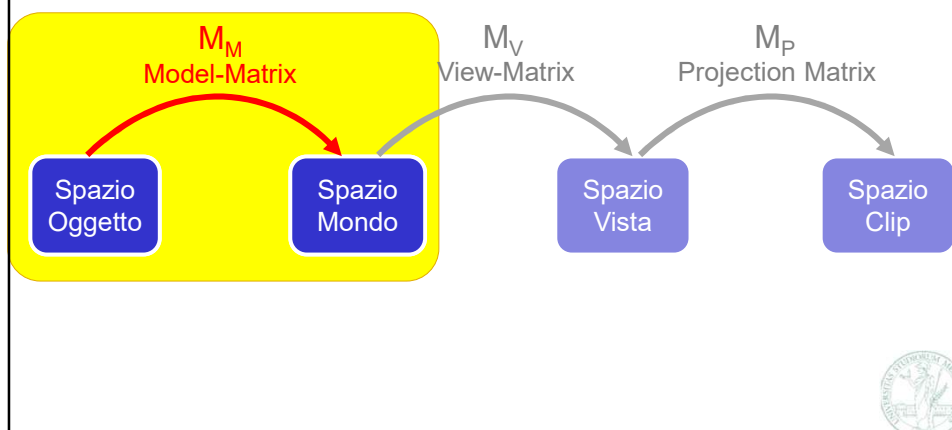
Domande (esercizio)

- ✓ 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?

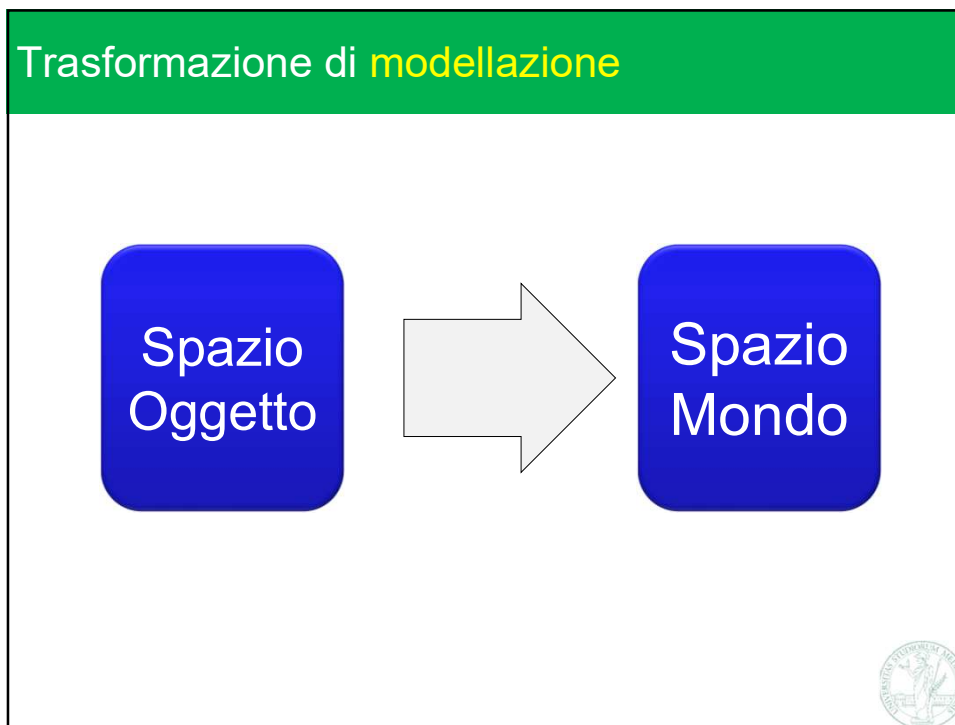


29

Pipeline di trasformazioni



30

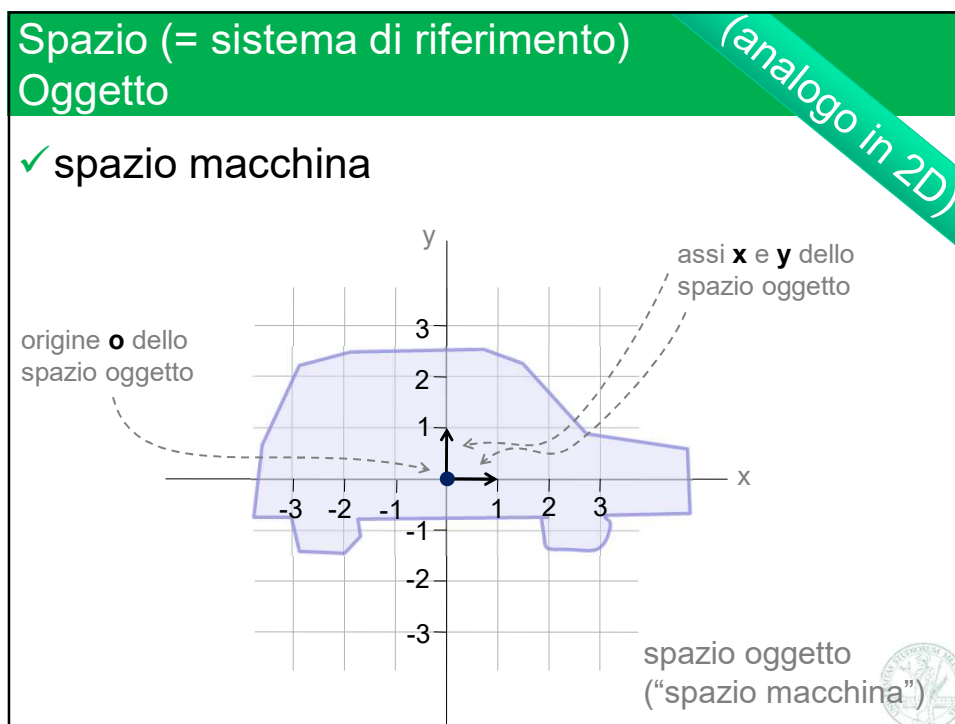


31

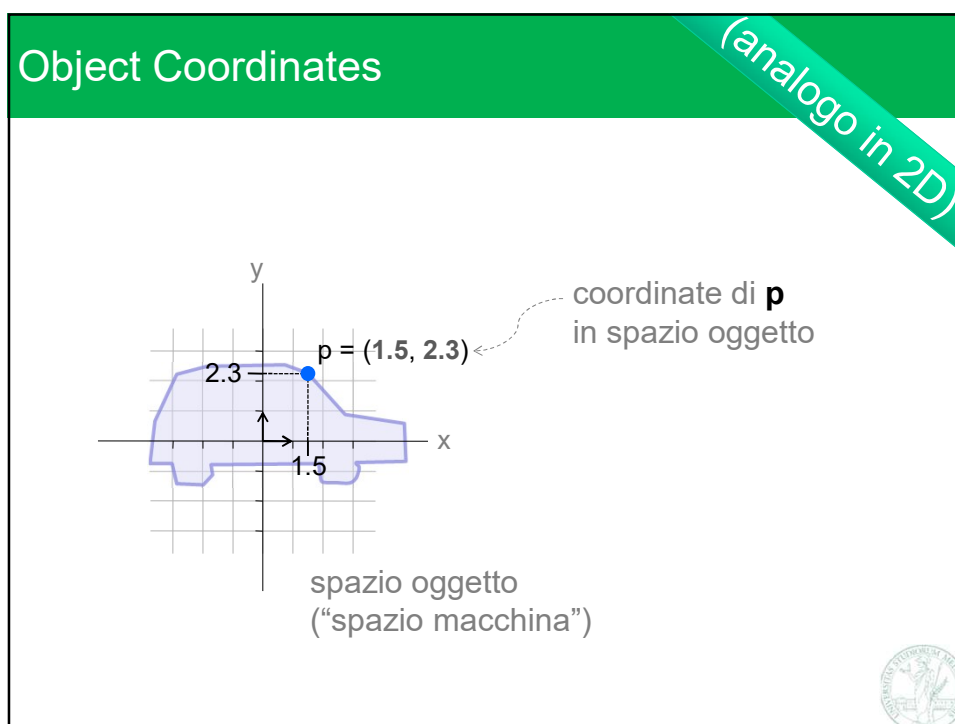
Trasformazione di "Modellazione"

- ✓ Da: **Object Frame**
A: **World Frame**
- ✓ Detta di modellazione perchè
serve a modellare il contenuto della scena
⇒decidendo la posizione / dimensione / etc di
ciascuno oggetto che la compone

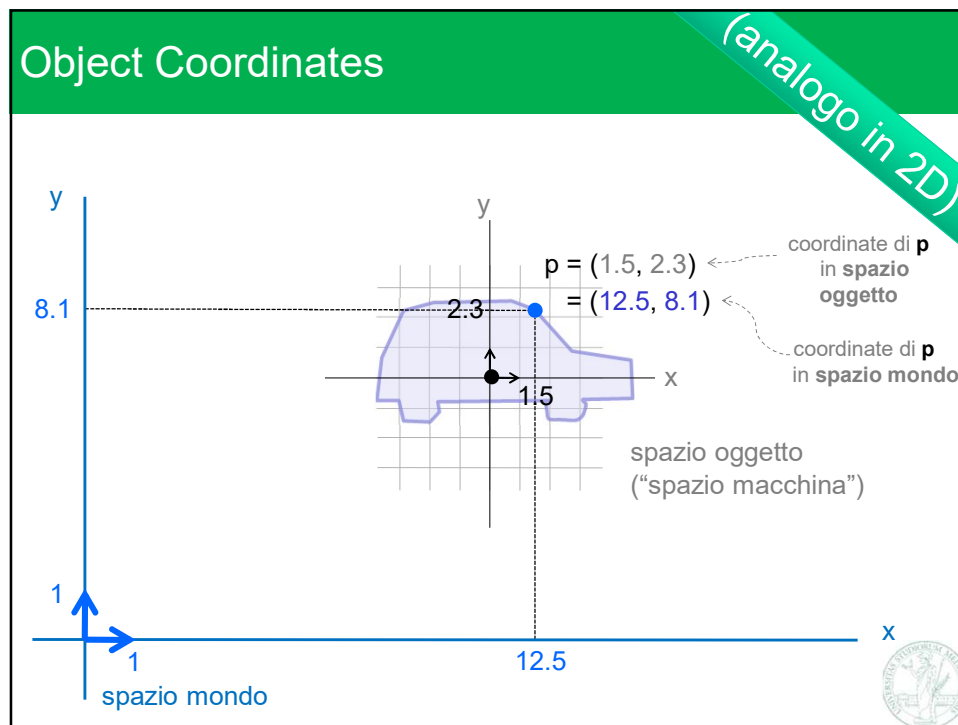
32



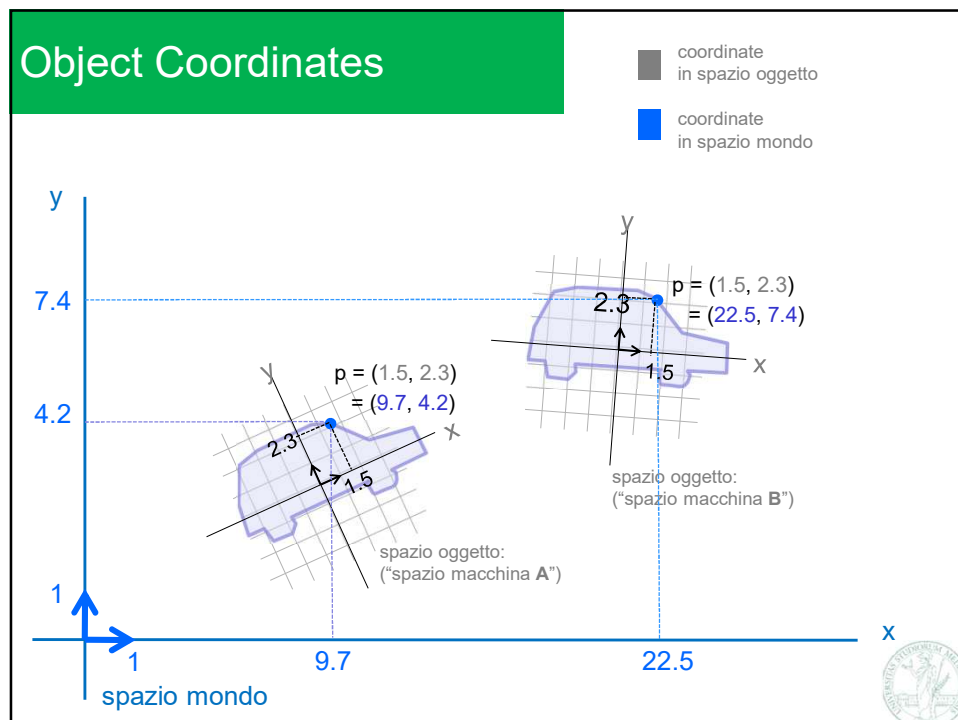
33



34



35



36

Spazio Oggetto: note

- ✓ Spazio riferito ad (ogni istanza di) un oggetto specifico
 - ⇒ Spazio casa, Spazio gatto, Spazio casa
- ✓ In un modello 3D, punti e vett. sono definiti in *questo* spazio
 - ⇒ es: posizione dei vertici (punti) e normali (vett) di una mesh
- ✓ Il modellatore dell'oggetto decide questo spazio
- ✓ Esempio di scelta tipica (ma non obbligata):
 - ⇒ Origine: centro dell'oggetto
 - oppure, centro della base (es di una statua), pos al suolo (di un personaggio)
 - ⇒ Asse X: da sinistra a destra (dell'oggetto)
 - ⇒ Asse Y: verso l'alto (dell'oggetto)
 - ⇒ Asse Z: verso il davanti (dell'oggetto) (o il dietro, se frame mano dx)



37

Spazio Mondo: note

- ✓ Spazio comune a tutta la scena 3D
- ✓ Esempio di scelta tipica (ma non obbligatoria)
 - ⇒ origine: centro della stanza / del mondo
 - ⇒ asse delle X: verso Est
 - ⇒ asse delle Y: verso il cielo
 - ⇒ asse delle Z: verso Nord (o Sud, se mano dx)



38

Trasformazione (matrice) di modellazione

- ✓ Da spazio Oggetto a Spazio Mondo
- ✓ Definisce come è composta la scena
 - ⇒ in che modo (istanze di) modelli 3D la compongono
- ✓ E' diversa su ogni istanza di modello
 - ⇒ es: 10 sedie di una stanza:
stesse coordinate oggetto, diverse matrici di modellazione,
diverse coordinate mondo
- ✓ Definisce in che modo ogni (istanza di) oggetto è dimensionato / posizionato nella scena 3D
- ✓ E' competenza del modellatore della scena



39

Trasformazione (matrice) di modellazione Note

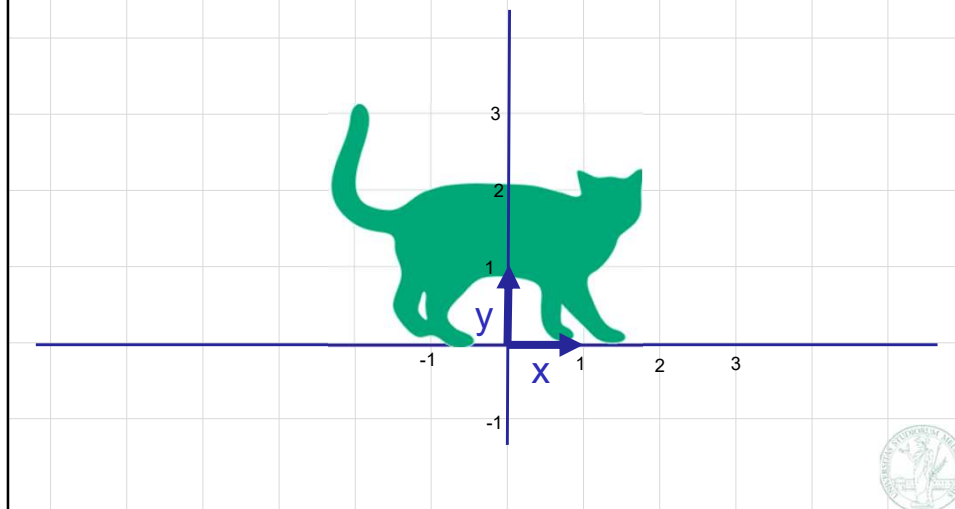
- ✓ Nelle applicazioni che prevedono il rendering di un unico oggetto, la matrice di modellazione può essere l'identità
 - ⇒ Lo spazio oggetto e lo spazio mondo coincidono
- ✓ Come tutte le matrici di trasformazione, può essere ottenuta moltiplicando diverse matrici ciascuna responsabile di una trasformazione. Per es:
 - ⇒ Una data scalatura, *seguita da*
 - ⇒ Una data rotazione, *seguita da*
 - ⇒ Una data traslazione
 - ⇒ Nota: se $C = AB$, C equivale a "fare prima B, poi A"
- ✓ Equivalentemente, la matrice di modellazione può essere interpretata (e/o costruita) come la descrizione del Sistema di riferimento Oggetto
 - ⇒ Le sue colonne rappresentano gli assi X, Y, Z e il punto di origine, espresso nel Sistema di riferimento Mondo



40

Esercizio

Spazio Oggetto (oggetto gatto)

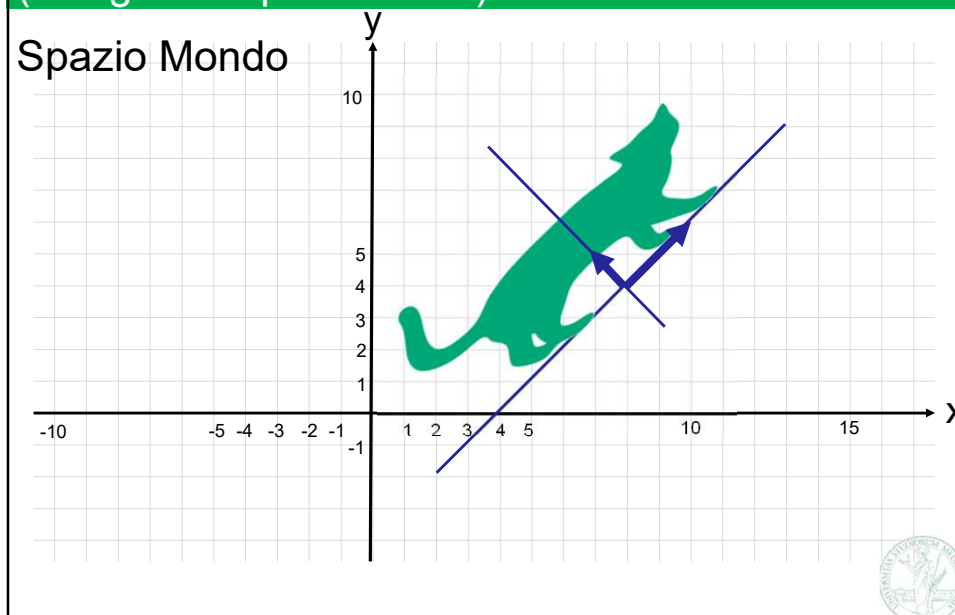


41

Esercizio

(disegno del piano $Z = 0$)

Spazio Mondo



44

Esercizio: le domande (parte A)

1. Determinare, guardando il secondo disegno, la matrice di modellazione del gatto (la Z non cambia)
2. Le coordinate oggetto della punta della coda sono ... (rispondere guardando il primo disegno)
3. Quindi le coordinate mondo dello stesso punto del gatto sono... rispondere moltiplicando per la matrice di modellazione, poi verificare la risposta nel secondo disegno
4. Computare l'inverso della matrice di modellazione del gatto
5. Una mosca si poggia nel punto di coordinate mondo (7,3,0): quali sono le coordinate della mosca rispetto al gatto, cioè nel sistema di coordinate gatto? Calcolarlo, moltiplicando con la matrice opportuna, poi verificarlo paragonando i due disegni
6. Volendo, ripetere per altri punti in spazio mondo e oggetto



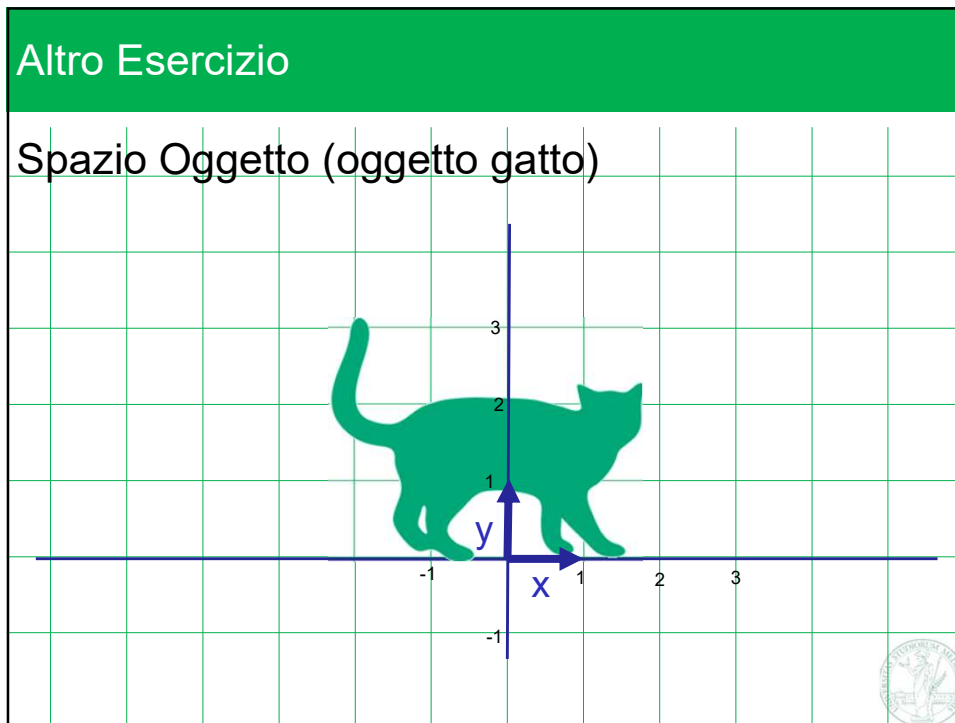
45

Esercizio: le domande (parte B)

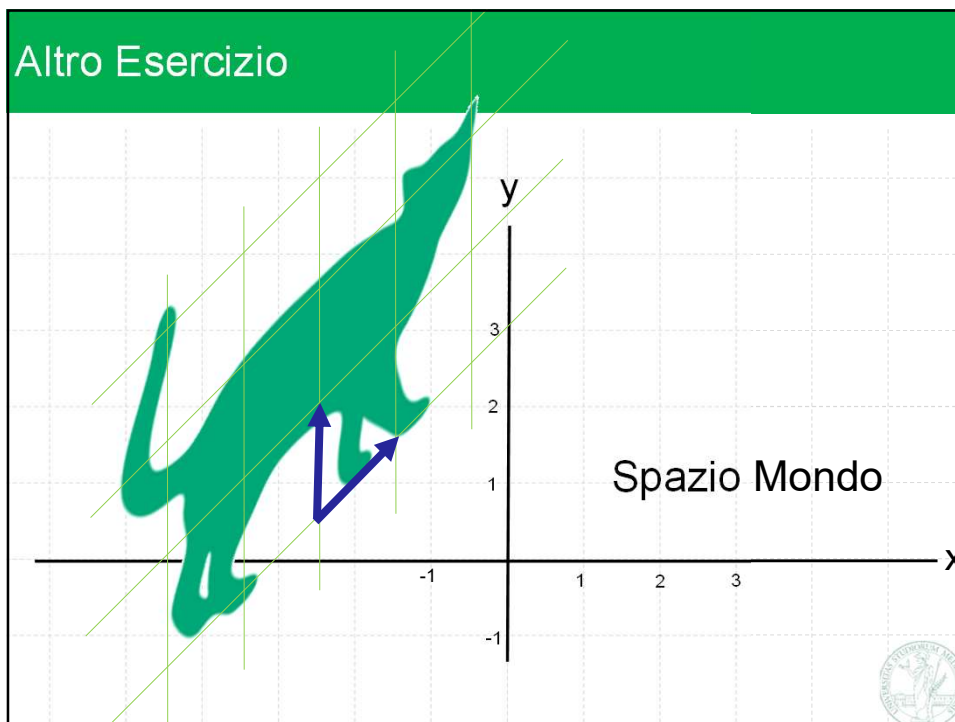
- ✓ Info: per portare il gatto nel mondo, abbiamo eseguito le seguenti trasformazioni, in quest'ordine:
 - ⇒ Scalato non uniformemente il gatto, con fattori... (nota: è stato allungato più di quanto sia stato alzato!)
 - ⇒ Ruotato il gatto, attorno all'asse delle... di gradi...
 - ⇒ Traslato il gatto di...
- 1. Completare le affermazioni sopra, riempiendo i puntini (guardando i disegni)
- 2. Scrivere le tre matrici corrispondenti
- 3. Calcolare la matrice di modellazione come prodotto delle matrici trovate (in che ordine vanno moltiplicate?)
- 4. Verificare che la matrice di modellazione evinta nella parte A (cioè guardando il disegno) corrisponda a quella calcolata
- 5. Scrivere le matrici delle tre trasformazioni inverse
- 6. Moltiplicarle per ottenere l'inversa della matrice di modellazione (in che ordine?).
- 7. Verificare che il risultato corrisponda a quello trovato nella parte A



46



47



48

Altro esercizio

- ✓ Ripetere alle stesse domande per i due disegni dati
- ✓ Per la parte B:
in questo caso, si ipotizzi che la trasf di modellazione sia ottenuta combinando, nell'ordine
 - ⇒ Una scalatura anisotropica solo sulle Y (di ...)
 - ⇒ Uno shearing (di ..., dell'asse delle ... rispetto a quello delle ...)
 - ⇒ Una traslazione (di ...)



49

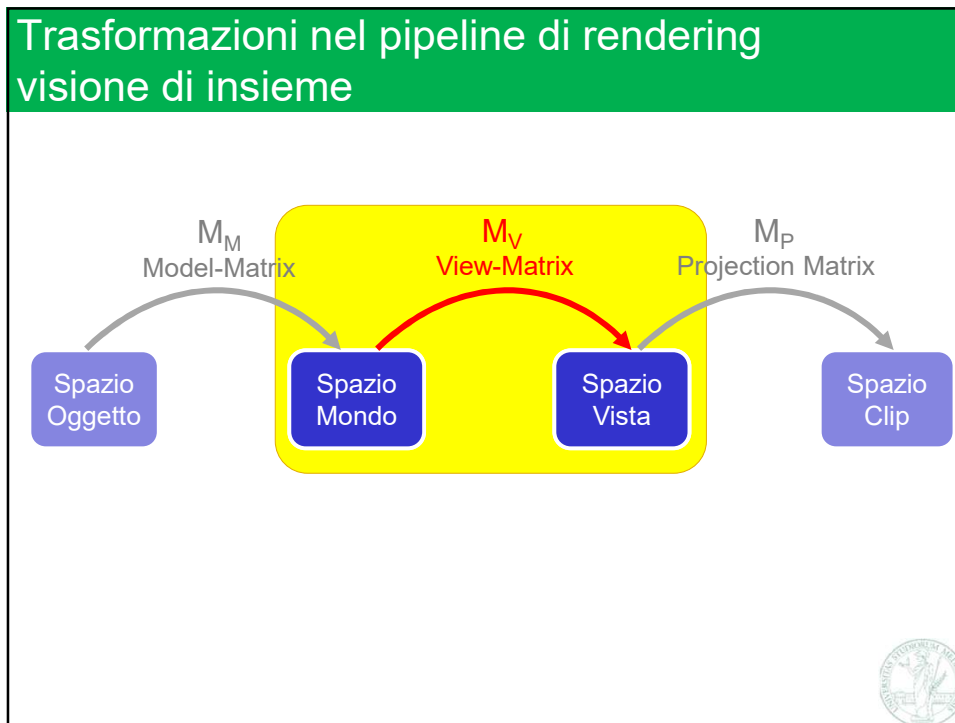
Multi-instancing di oggetti 3D

La divisione fra coordinate oggetto e mondo è utile anche per risparmiare memoria

- ✓ consente di riutilizzare lo stesso modello più volte nella stessa scena (istanze dell'oggetto)
- ✓ ogni istanza: stesse Object Coordinates dei vertici, ma una trasformazione (di "modellazione") diversa per arrivare a World Coordinates diverse
- ✓ Es:
 - ⇒ ruote di una macchina (4 volte l'istanza di una ruota)
 - ⇒ alberi, case, sedie in una stanza, pedoni su una scacchiera,
 - ⇒ etc, etc, etc



50



51



52

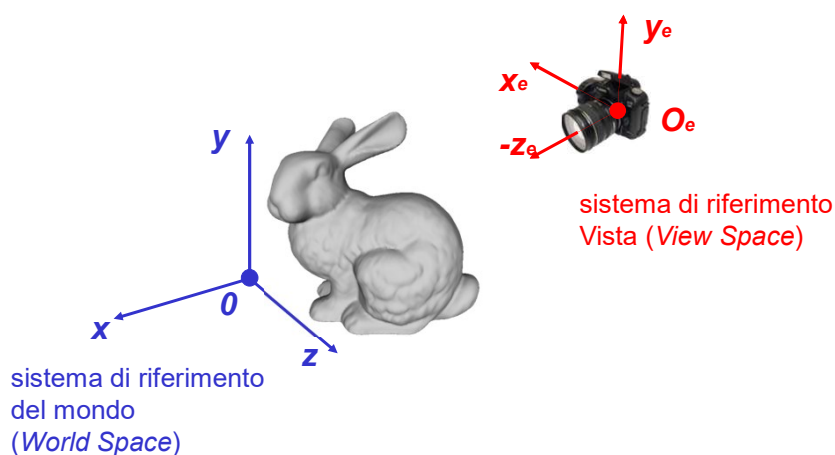
Trasformazione (matrice) di vista

- ✓ Da spazio Oggetto a Spazio Vista
- ✓ Definisce come viene *inquadrata* la scena
 - ⇒ cioè dove è piazzata e come è orientata la macchina fotografica virtuale (la «camera») che la riprende
 - ⇒ dipende cioè dai cosiddetti **parametri ESTRINSECI** della camera
- ✓ Tipicamente è una trasformazione rigida
- ✓ Ricordare: le trasformazioni si possono combinare!
 - ⇒ Model-View matrix: matrice di modellazione-vista da spazio oggetto a spazio a spazio Mondo
 - ⇒ Come si ottiene?



54

Spazio (sistema di riferimento) Vista



55

Trasformazione di "Vista"

- ✓ Da: **World Frame**
A: **View Frame**



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

- ✓ 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



56

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. Direzione di vista
 - ⇒ Oppure, equivalentemente, posizione di un oggetto target osservato, che si richiede compaia in mezzo alla foto
 3. Vettore "alto"
 - ⇒ Descrive una direzione che, nella foto, deve apparire come direzione target
 - ⇒ Distingue, ad es, una foto "portrait" da una "landscape" (e da un "campo obliquo")
- ✓ Nota: si tratta di punti e vettori espressi nel Sistema di coordinate Mondo



57

Esempio tipico di costruzione trasformazione 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)

58

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)

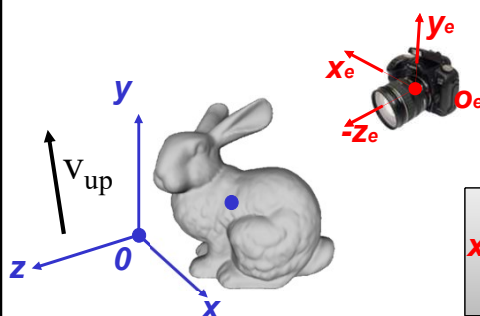
sistema di riferimento della camera (view space)

59

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, xe );
```

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.



60

Esempio tipico di costruzione trasformazione 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 = normalize( ze );

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

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

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.



61