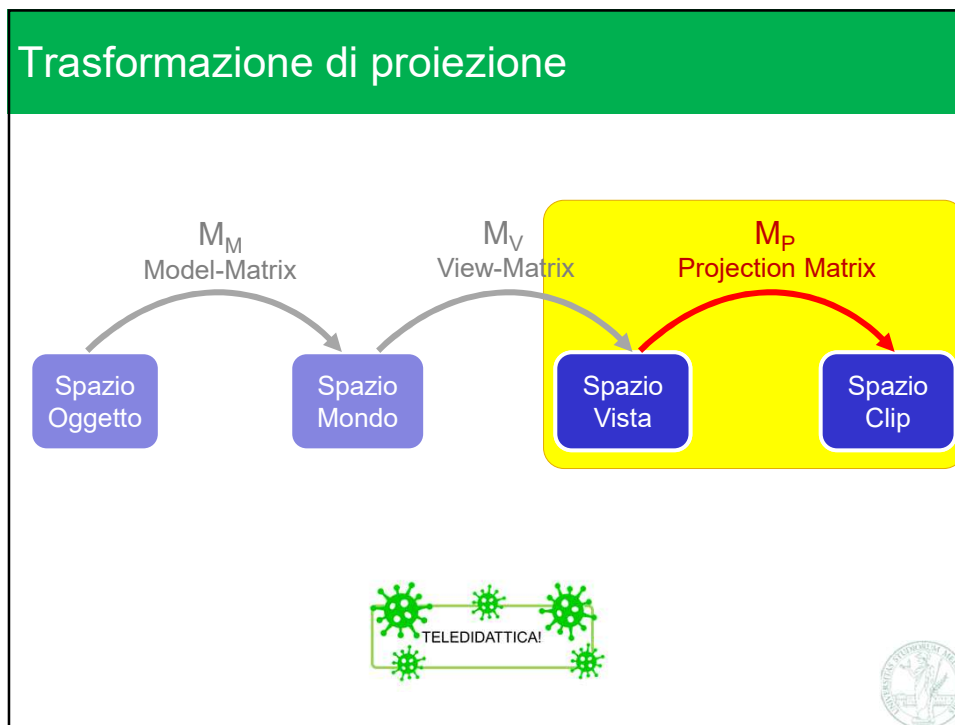


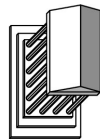
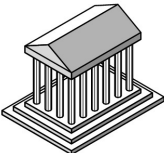
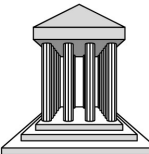


70

Trasformazione di proiezione

- ✓ Vecchio problema:
 - ⇒ (in arte, architettura progettazione)
- ✓ come riportare
 - ⇒ oggetti 3D
 - ⇒ su un piano 2D

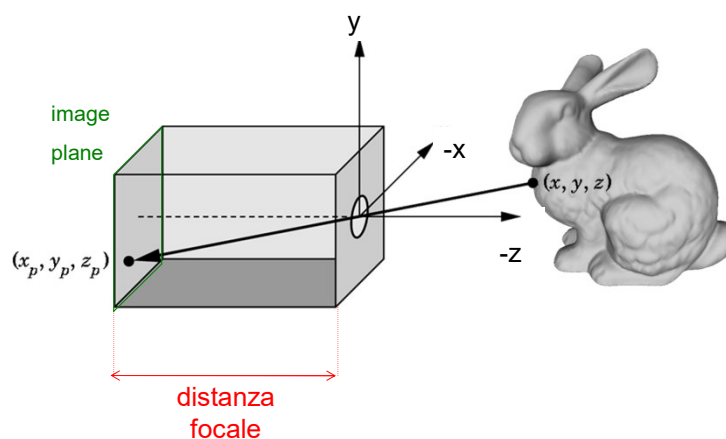
Alcune soluzioni classiche:

		
Prospetto Frontale	Assonometria cavaliera	Pianta Obliqua
		
Assonometria Isometrica	Prospettiva a punto di fuga singolo	Prospettiva a punto di fuga triplo

71

Nostro modello semplificato:

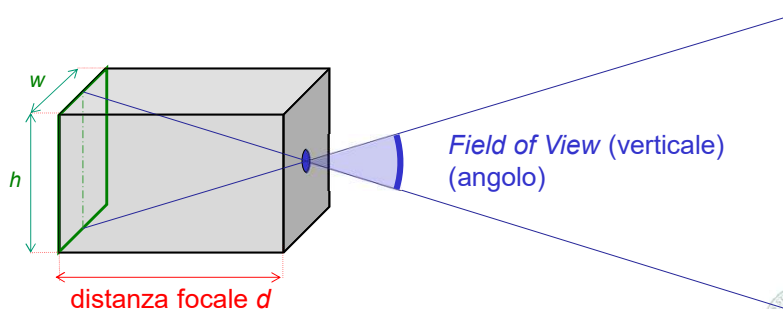
✓ pin-hole camera



77

Parametri *intrinseci* della camera: (solo i principali)

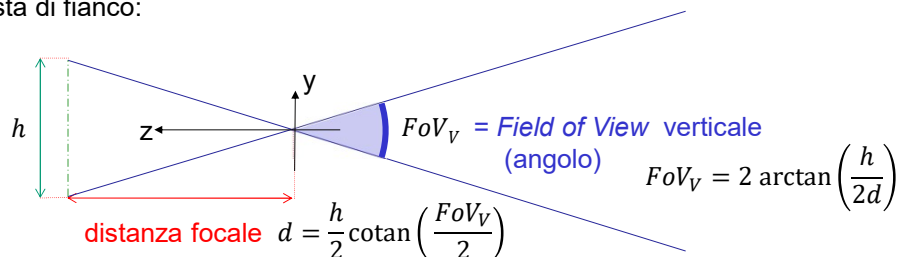
- ✓ dimensioni image plane (w, h) (in spazio vista)
- ✓ distanza focale (d) (in spazio vista) oppure Field of View (FoV)
 - ⇒ si possono ottenere uno dall'altro (come?)
 - ⇒ il valore d è facile da usare, ma FoV è intuitivo da settare:
 - FoV grande $>60^\circ$ (dist foc. piccola): «grandangolo»
 - FoV piccolo $<45^\circ$ (dist foc. grande): «teleobiettivo»



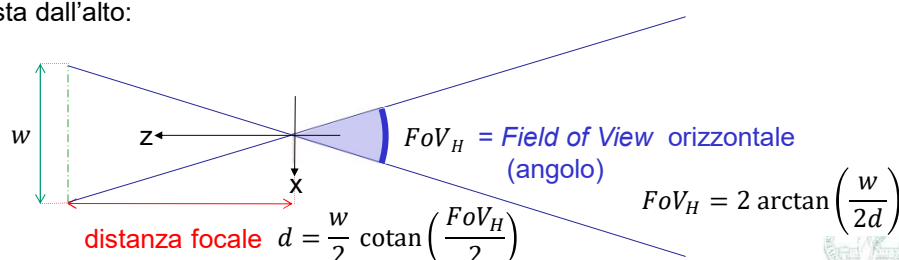
78

Da Field of View a Distanza focale

Vista di fianco:



Vista dall'alto:

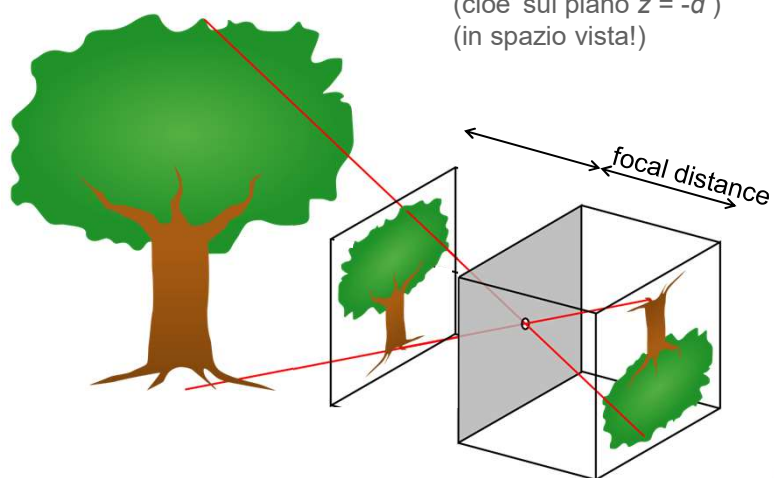


79

Pin Hole camera

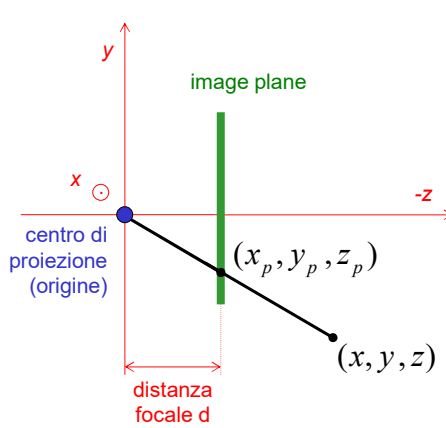
per semplicità, immaginiamoci il piano immagine *davanti* alla camera (e nn ribaltato), piuttosto che *dietro* (e ribaltato).

(cioe' sul piano $z = -d$)
(in spazio vista!)



80

Matematicamente



$$\begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = k \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \text{con } k \text{ t.c. } z_p = -d$$

quindi...

$$k = -d / z \quad \text{e} \quad \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} = \begin{pmatrix} -d \cdot x / z \\ -d \cdot y / z \\ -d \end{pmatrix}$$

Nota:
 non è affine;
 Per es, non mantiene parallelismo
 (ma mantiene: colinearità)

81

Estendiamo la rappresentazione di punti e vettori in coordinate **omogenee**

Punti:

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

1 →

Vettori:

$$\vec{v} = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

0 →

83

Estendiamo la rappresentazione di punti e vettori in coordinate **omogenee**

Punti: $\neq 0 \rightarrow \mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ * \end{bmatrix}$

Vettori: $0 \rightarrow \vec{v} = \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$



84

Estendiamo la notazione

✓ Esprimo i *punti* anche con la notazione

$$\mathbf{p} = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} \quad \text{con } w \neq 0$$

$$\begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix} \xrightarrow{\text{divisione per 4ta comp}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

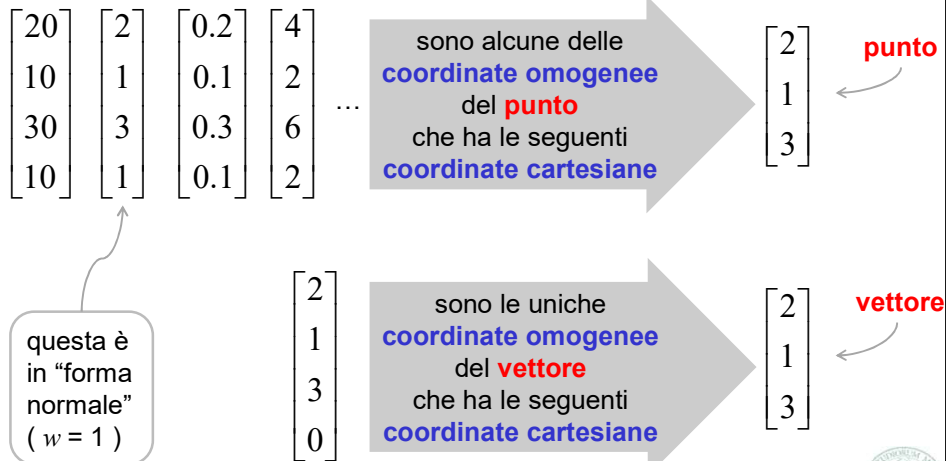
anche detta
normalizzazione affine



85

Estendiamo la notazione

✓ Per es:



86

Coordinate omogenee di punti

- ✓ Il punto P di coordinate **cartesiane** (x, y, z) è rappresentato in coordinate **omogenee** come (xw, yw, zw, w) , con w qualunque (ma non 0)
- ✓ Set di coordinate omogenee diversi (x, y, z, w) e (x', y', z', w') possono rappresentare lo stesso punto;
⇒ quando?
- ✓ Quando $w = 1$ (forma **canonica**) le coord cartesiane del punto coincidono con le prime tre coord omogenee.
- ✓ Con $(x, y, z, w \neq 0)$ si rappresentano **punti**,
con $(x, y, z, 0)$ si rappresentano **vettori**.
- ✓ *Nota: tutte le matrici di trasformazione viste fin'ora funzionano anche con questa notazione generalizzata! Es: VERIFICA.*

87

Coordinate omogenee (o affini)

✓ Tutte le matrici di trasformazione che abbiamo visto fin'ora continuano a funzionare anche con i punti espressi in coordinate omogenee non canoniche ($w \neq 1$)

✓ Esempio, traslazione:

$$\begin{bmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} 20 \\ 15 \\ 50 \\ 10 \end{pmatrix} = \begin{pmatrix} 70 \\ 35 \\ 80 \\ 10 \end{pmatrix}$$

Matrice di
traslazione
del vett
(+5, +2, +3)

Punto
di coord
cartesiane
(2, 1.5, 5)

Punto
di coord
cartesiane
(7, 3.5, 8)

(+5, +2, +3)

✓ Ma ora possiamo esprimere anche la proiezione prospettica...

88

Matrice di proiezione prospettica variante (equivalente)

matrice di trasformazione
per la
proiezione prospettica
(nota: non è affine)

$$\mathbf{P} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$\mathbf{P} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} d \cdot x \\ d \cdot y \\ d \cdot z \\ -z \end{pmatrix}$$

...che esprime il
punto di coordinate
cartesiane...
(divido per w)

$$\begin{pmatrix} -x \cdot d/z \\ -y \cdot d/z \\ -d \end{pmatrix}$$

90

Proiezione Prospettica: che effetto fa

d piccolo

d grande

d infinito
(diventa una proiezione ortogonale)

Più distorsione prospettica.
Effetto "fish-eye" (grandangolo)

Proporzioni più mantenute
Effetto "zoom" (eg. vista dal satellite)

$$P = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

92

Un problema con questa matrice di proiezione: schiaccia la scena su un piano a Z costante

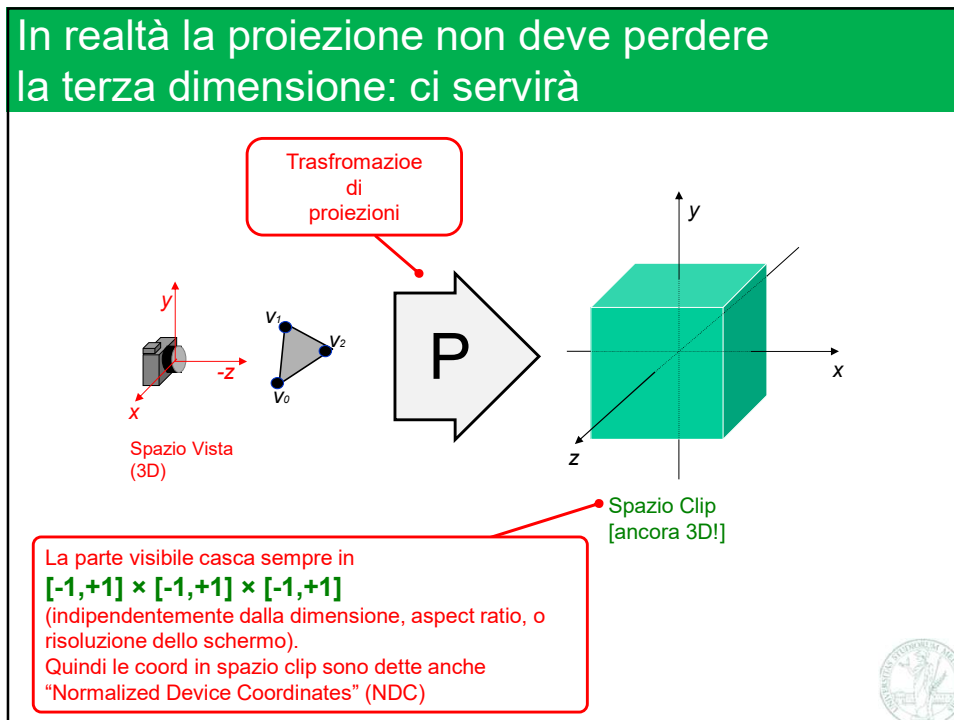
normalizzazione affine
(x estrazione coordinate Cartesiane)

$$\begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} d \cdot x \\ d \cdot y \\ d \cdot z \\ -z \end{pmatrix} \rightarrow \begin{pmatrix} -x \cdot d/z \\ -y \cdot d/z \\ -d \\ 1 \end{pmatrix}$$

perspective matrix view coords clip coords

valore di z costante
formalmente è corretto
(il punto finisce sempre sul piano immagine)
ma perdiamo una dimensione inutilmente

93

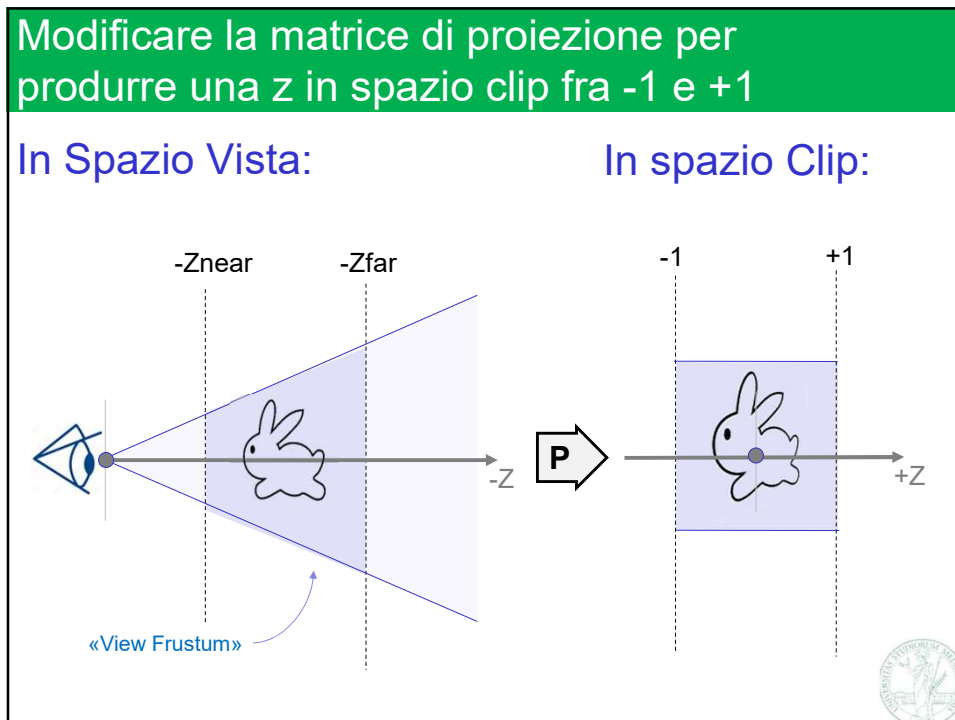


95

Modificare la matrice di proiezione per rimappare la Z correttamente

- ✓ Scelgo arbitrariamente
 - ⇒ z_near : la distanza della cosa più *vicina* inquadrata nell'immagine
 - ⇒ z_far : la distanza della cosa più *lontana* inquadrata nell'immagine
 - ⇒ sono distanze definite in spazio vista!
 - sono due reali > 0
- ✓ La trasf di proiezione deve portare l'intervallo Z (spazio vista) $[-z_near .. -z_far]$ nell'intervallo Z (spazio clip) $[-1..+1]$

96



97

Modificare la matrice di proiezione per rimappare la Z correttamente

✓ Voglio modificare **P** in modo che la **Z** in output sia una qualche funzione della **Z** in input

$$\begin{pmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} d \cdot x \\ d \cdot y \\ Az + B \\ -z \end{pmatrix} \rightarrow \begin{pmatrix} -x \cdot d/z \\ -x \cdot d/z \\ -A - B/z \\ 1 \end{pmatrix}$$

view coords clip coords

che valori usare qui?
(per **A** e **B**)

98

Modificare la matrice di proiezione per produrre una z in spazio clip fra -1 e +1

- ✓ Troviamo i valori per A e B
- ✓ Vogliamo che:
 - ⇒ se $z_{\text{vista}} = -z_{\text{Near}}$, z_{clip} deve essere -1
 - ⇒ se $z_{\text{vista}} = -z_{\text{Far}}$, z_{clip} deve essere $+1$
 - ⇒ Quindi

$$-A + \frac{B}{z_{\text{Near}}} = -1 \quad -A + \frac{B}{z_{\text{Far}}} = +1$$

- ✓ Risolvo per A e B ottenendo... (verificare!)

$$A = \frac{z_{\text{Far}} + z_{\text{Near}}}{z_{\text{Far}} - z_{\text{Near}}} \quad B = \frac{2 \cdot z_{\text{Far}} \cdot z_{\text{Near}}}{z_{\text{Far}} - z_{\text{Near}}}$$



99

Matrice di Proiezione Prospettica ottenuta

$$P = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & \frac{(z_F + z_N)}{(z_F - z_N)} & \frac{2 \cdot z_F \cdot z_N}{(z_F - z_N)} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

a aspect ratio

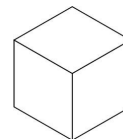
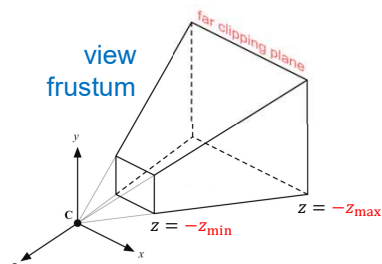
d lunghezza focale



100

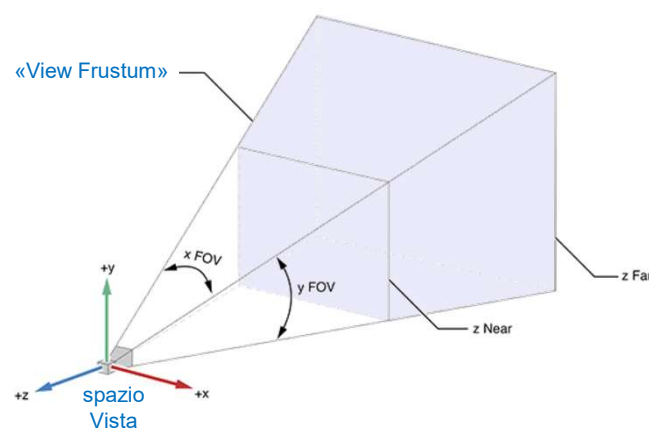
Limiti dello spazio visibile (inquadrato)

	Spazio Vista	Spazio Clip
x	«left clipping plane»	> -1
	«right clipping plane»	$< +1$
y	«bottom clipping plane»	> -1
	«top clipping plane»	$< +1$
z	$-z_{\min}$	> -1
	$-z_{\max}$	$< +1$

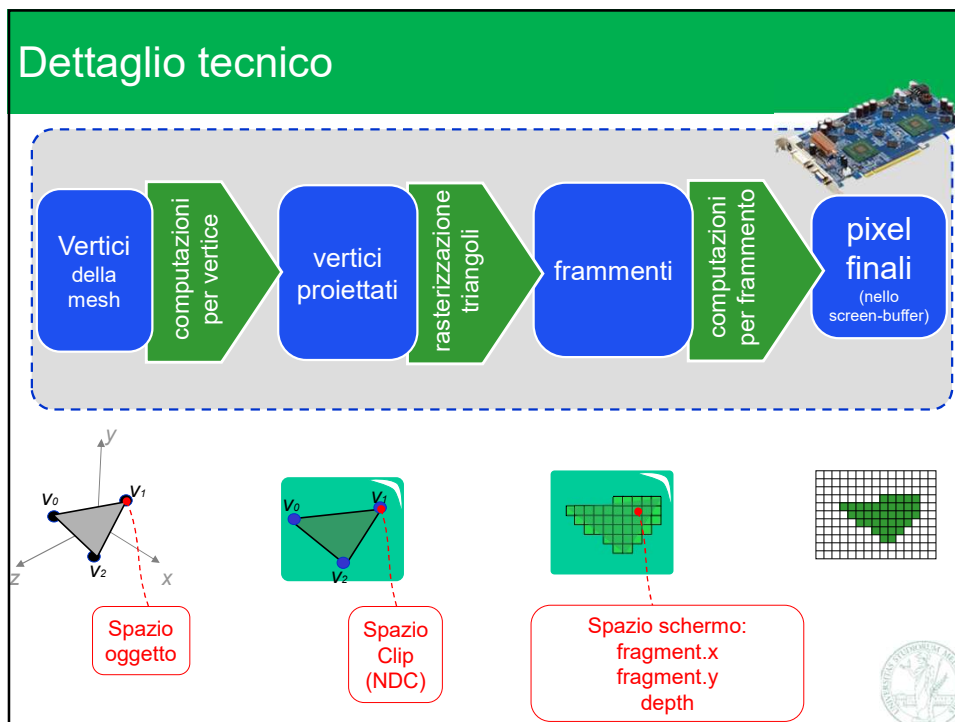


101

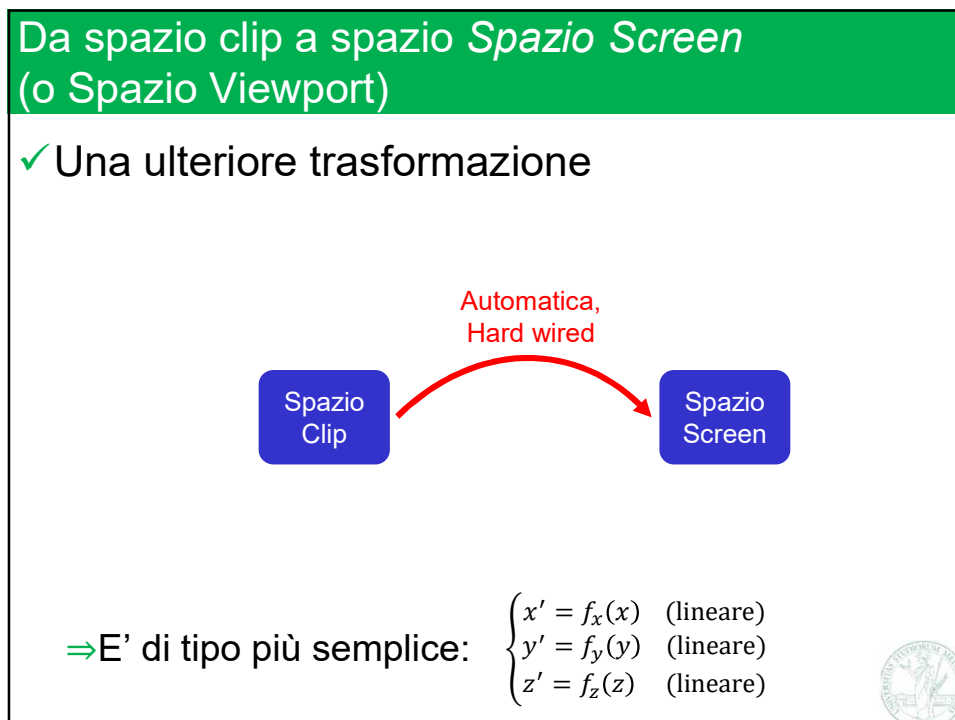
View Frustum (o viewing frustum) in 3D



102



105



106

Da Clip Space a Screen Space (o "Viewport")

✓ In **coordinate screen**,
i pixel (e i frammenti)
sono quadratini
a coordinate intere
di dimensione 1×1



108

X e Y da Clip Space a Screen Space

$$f_{to_screen} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (x+1)/2 \cdot RES_X \\ (y+1)/2 \cdot RES_Y \\ (z+1)/2 \end{pmatrix}$$

in
[-1 , +1]³
in
[0 , 1]³
Se $RES_X \neq RES_Y$
questo
corrisponde
ad una scalatura
non uniforme



109

Limiti dello spazio visibile (sommario)

	Spazio Vista	Spazio Clip	Spazio Schermo	
x	«left clipping plane»	-1	0	pixel coordinates
	«right clipping plane»	+1	res_x	
y	«bottom clipping plane»	-1	0	pixel depth
	«top clipping plane»	+1	res_y	
z	$-z_{min}$	-1	0	
	$-z_{max}$	+1	1	

110

Dettaglio tecnico

- ✓ Il **processing per vertice** :
ha il compito di produrre
le **clip coords** del vertice
 - ⇒ sotto forma di coordinate omogenee $(x \cdot w, y \cdot w, z \cdot w, w)$
 - ⇒ se X, Y, Z sono in $[-1..+1]$, il vertice è nell'immagine
 - ⇒ quando X, Y , **oppure** Z fuori da $[-1..+1]$ → fuori da immagine!
 - ⇒ cioè: se $Z > +1$ troppo lontano, non disegno
cioè: se $Z < -1$ troppo vicino, non disegno
 - ⇒ La matrice di proiezione deve quindi occuparsi
anche di produrre la Z corretta in spazio clip
- ✓ Il rasterizzatore (hard wired) trasforma le coordinate
clip x, y, z in coordinate schermo x, y e *depth*

111

Dettaglio tecnico

- ✓ Il vertex processor (*programmabile!*)
 - ⇒ deve produrre clip coordinates del vert. processato (in coordinate affini, come un vec4) (anche non in forma normale, $w \neq 1$)
- ✓ Il rasterizzatore: (*hard-wired*)
 - ⇒ prende vertici in clip coordinates
 - ⇒ produce frammenti in screen coordinates
 - cioè «in pixel», e.s. in $[0..639] \times [0..479]$
 - ⇒ prima di rasterizzare, converte le clip coordinates in screen coordinates (e divide per w)
 - ⇒ applica cioè la trasformazione di viewport



112

Trasformazione di viewport : da Clip Space a Screen Space (o Viewport)

- ✓ Piano XY dello Spazio Clip: $[-1, +1] \times [-1, +1]$
Un quadrato (di lato 2)
- ✓ Viewport: $[0, \text{resX}] \times [0, \text{resY}]$
 - ⇒ Un rettangolo (quadrato solo se $\text{resX} = \text{resY}$)
- ✓ Il passaggio da Clip Space a Screen Space introduce uno scaling NON uniforme (se $\text{resX} \neq \text{resY}$)

quindi:

- ✓ nella Trasformazione di Proiezione
deve compensare applicando
uno scaling NON uniforme inverso



113

Pre-correzione dell'aspect ratio

✓ Matrice di proiezione prospettica con correzione dell'aspect ratio $a = \frac{RES_X}{RES_Y}$

$$P = \begin{pmatrix} d & 0 & 0 & 0 \\ 0 & da & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix} \text{ oppure } P = \begin{pmatrix} d/a & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & A & B \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

Nota: la trasformaz. di proiezione dipende dai parametri INTRINSECI della macchina fotografica... compreso l'aspect ratio del suo frame!



114

Matrice di Proiezione Prospettica finale

$$P = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & a \cdot d & 0 & 0 \\ 0 & 0 & \frac{(Z_F + Z_N)}{(Z_F - Z_N)} & \frac{2 \cdot Z_F \cdot Z_N}{(Z_F - Z_N)} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

in spazio vista

- a aspect ratio
- d lunghezza focale
- Z_N z-near
- Z_F z-far



115