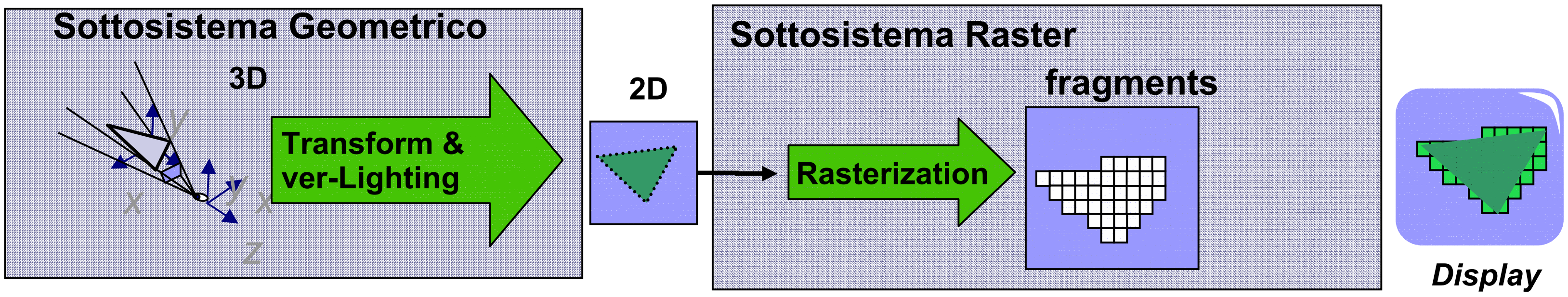
Grafica Computazionale

Trasformazioni

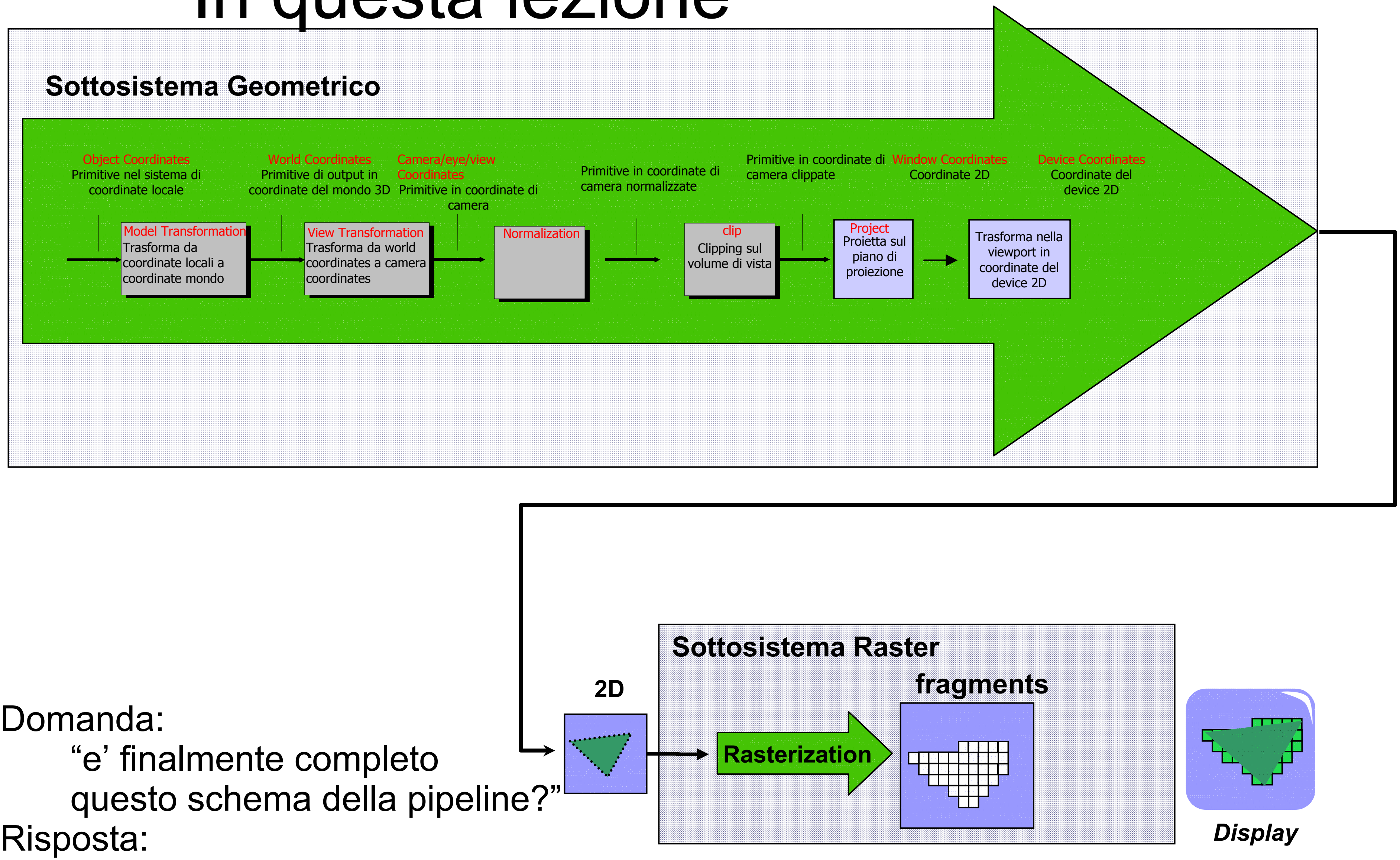
Fabio Ganovelli
fabio.ganovelli@isti.cnr.it
a.a. 2005-2006

Riepilogo



- Cosa sappiamo fare:
  - Rasterizzare segmenti, riempire poligoni
    - In 2D su una finestra rettangolare
  - Fare *clipping* delle primitive, visualizzare solo le superfici visibili (HSR)
    - In 2D su una finestra rettangolare + un valore di profondita' (distanza dal piano) delle primitive (e dei pixel)

In questa lezione



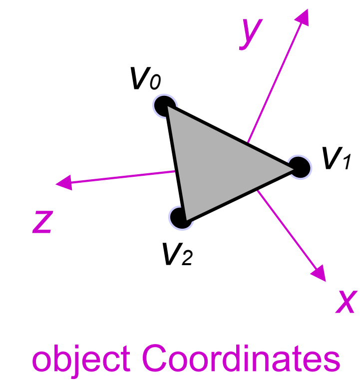
Domanda:

“e’ finalmente completo questo schema della pipeline?”

Risposta:

“No no…”

In questa lezione

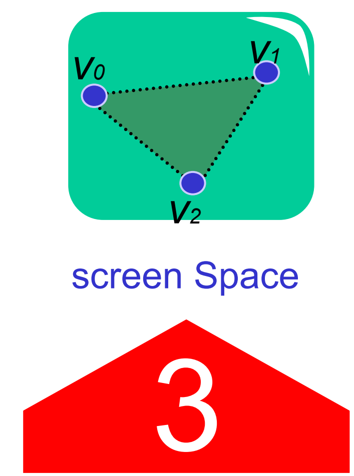


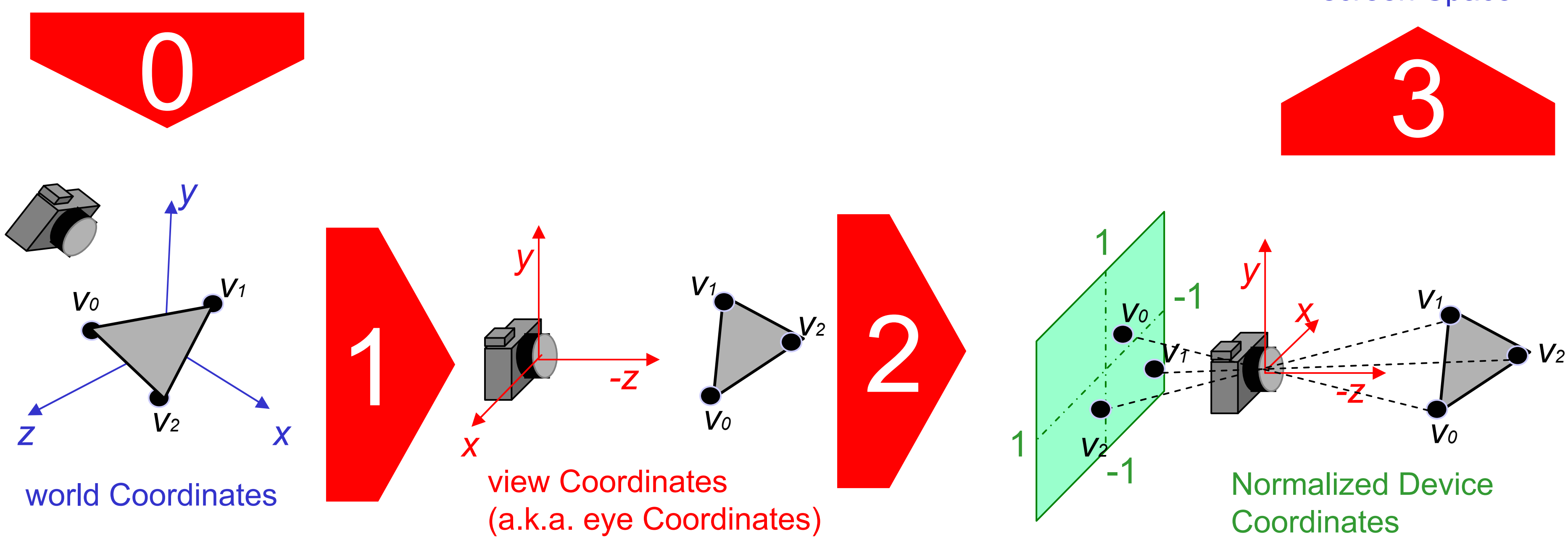
0) transformazione di modellazione

1) transformazione di vista

2) transformazione di proiezione

3) transformazione di viewport



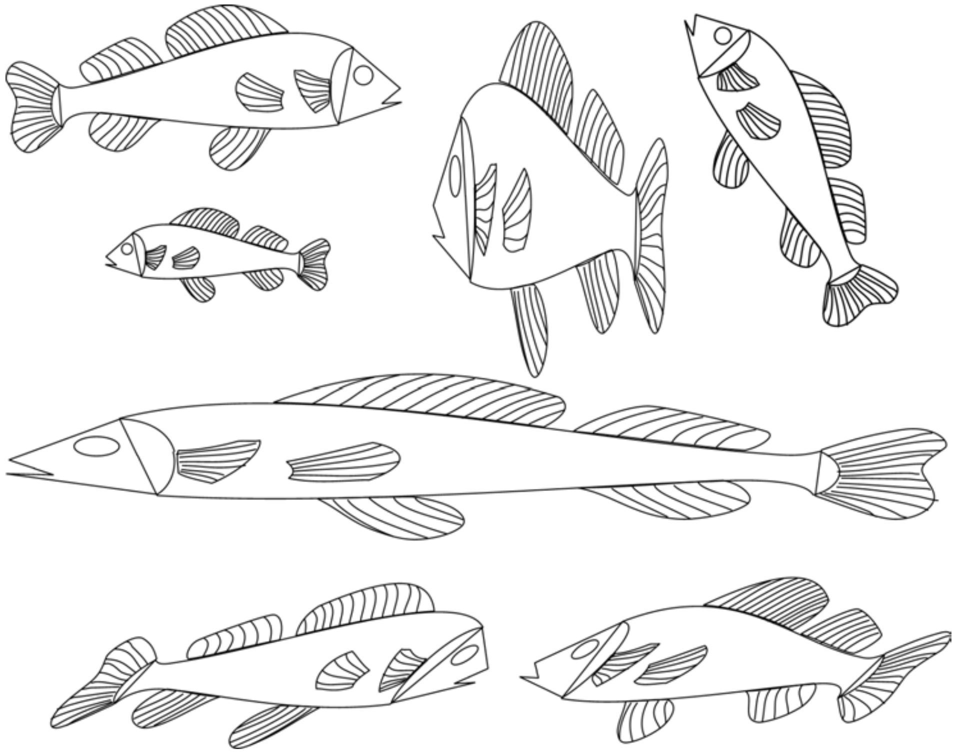


Prima parte: argomenti trattati …

- Trasformazioni geometriche e matrici
  - Entità geometriche e trasformazioni affini;
  - Trasformazioni geometriche nel piano (traslazione, scalatura e rotazione);
  - Coordinate omogenee e rappresentazione matriciale;
  - Altre trasformazioni geometriche nel piano: riflessione e deformazione;
  - Composizione di trasformazioni;
  - Trasformazioni geometriche nello spazio.

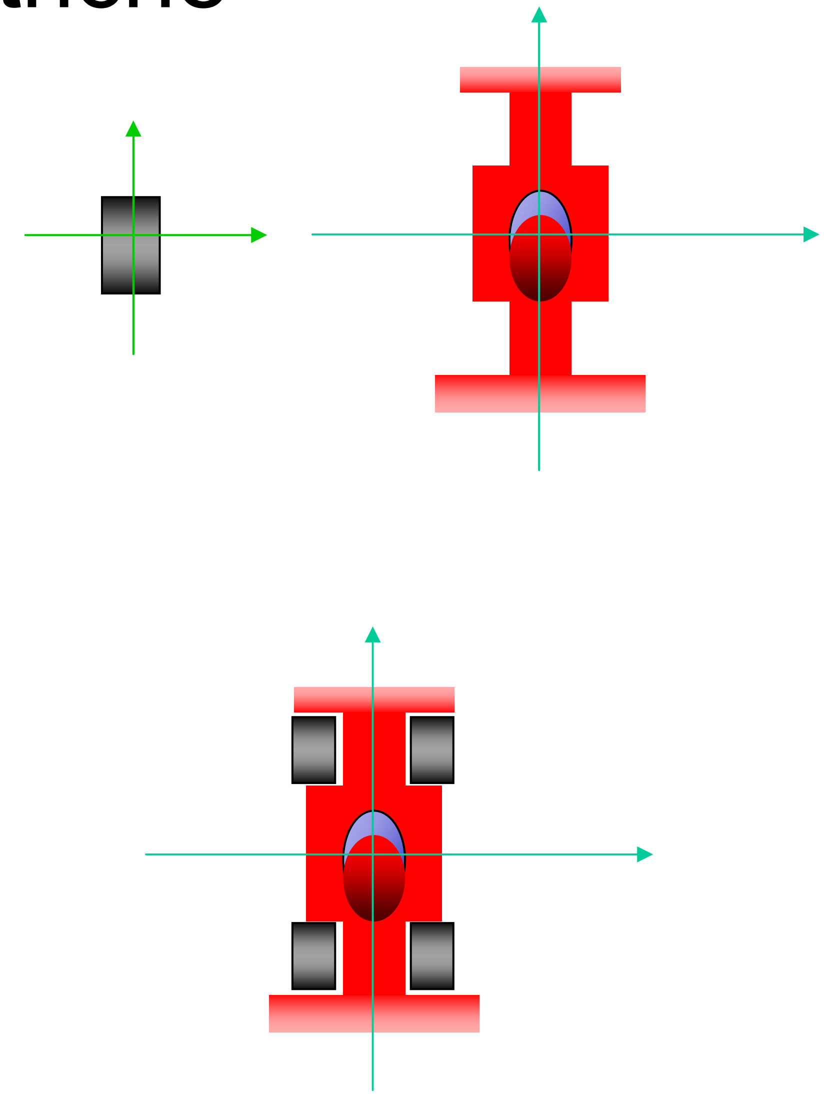
Trasformazioni geometriche

- Le *trasformazioni geometriche* permettono di istanziare una stessa geometria con attributi (posizione, orientamento, fattori di scala) diversi.



Trasformazioni geometriche

- Le trasformazioni geometriche permettono di definire ogni oggetto in un proprio sistema di riferimento. L'oggetto eredita gli attributi geometrici dal "padre" (il sistema "ruota" eredita gli attributi dal sistema "macchina" ad ogni istanza)



Entità geometriche e trasformazioni affini

- Entità geometriche
  - **Punto** - entità geometrica caratterizzata da un solo attributo: la posizione rispetto ad un sistema di riferimento;
  - **Vettore** - entità geometrica caratterizzata da due attributi: lunghezza e direzione.

  - Lunghezze, angoli, etc. sono espresse mediante **scalari.**

Entità geometriche e trasformazioni affini

- Spazio vettoriale
  - Due entità:
    - Scalari (*a, b, c,* …)
    - Vettori (**T, U, V,** …).
  - Operazioni:
    - Somma e moltiplicazione di scalari;
    - Somma vettore-vettore;
    - Moltiplicazione scalare-vettore.

Entità geometriche e trasformazioni affini

- Spazio affine
  - Tre entità:
    - Scalari (*a, b, c,* …)
    - Vettori (**T, U, V,** …)
    - Punti (*P, Q, R,* …)
  - Operazioni:
    - Le operazioni di uno spazio vettoriale,
    - Somma punto-vettore (restituisce un punto),
    - Sottrazione punto-punto (restituisce un vettore).

Entità geometriche e trasformazioni affini

- Le trasformazioni geometriche sono lo strumento che consente di manipolare **punti** e **vettori** all’interno del mondo dell’applicazione grafica;
- Le trasformazioni geometriche sono **funzioni** che mappano un punto (vettore) in un altro punto (vettore);

- La trasformazione di una primitiva geometrica **si riduce alla trasformazione dei punti caratteristici** (vertici) che la identificano nel rispetto della connettività originale. Questo grazie al fatto che trattiamo di trasformazioni **affini** …

Entità geometriche e trasformazioni affini

- Le **trasformazioni geometriche affini** sono trasformazioni *lineari*

$$f(aP + bQ) = af(P) + bf(Q)$$

- Esse preservano:
  - *colinearità* (i punti di una linea giacciono ancora su di una linea dopo la trasformazione);
  - *rapporto tra le distanze* (Il punto medio di un segmento rimane il punto medio di un segmento anche dopo la trasformazione).

Trasformazioni geometriche nel piano

- Le trasformazioni geometriche di base sono:
  - Traslazione;
  - Scalatura;
  - Rotazione.

- Altre trasformazioni geometriche comuni (ma derivabili dalle precedenti) sono:
  - Riflessione rispetto ad un asse;
  - Riflessione rispetto ad un punto;
  - Deformazione.

Trasformazione di Traslazione

- Traslare una primitiva geometrica nel piano significa muovere ogni suo punto $P(x,y)$ di d_x unità lungo l'asse x e di d_y unità lungo l'asse y fino a raggiungere la nuova posizione $P'(x', y')$ dove:

$$x' = x + d_x, \quad y' = y + d_y$$

- In notazione matriciale:

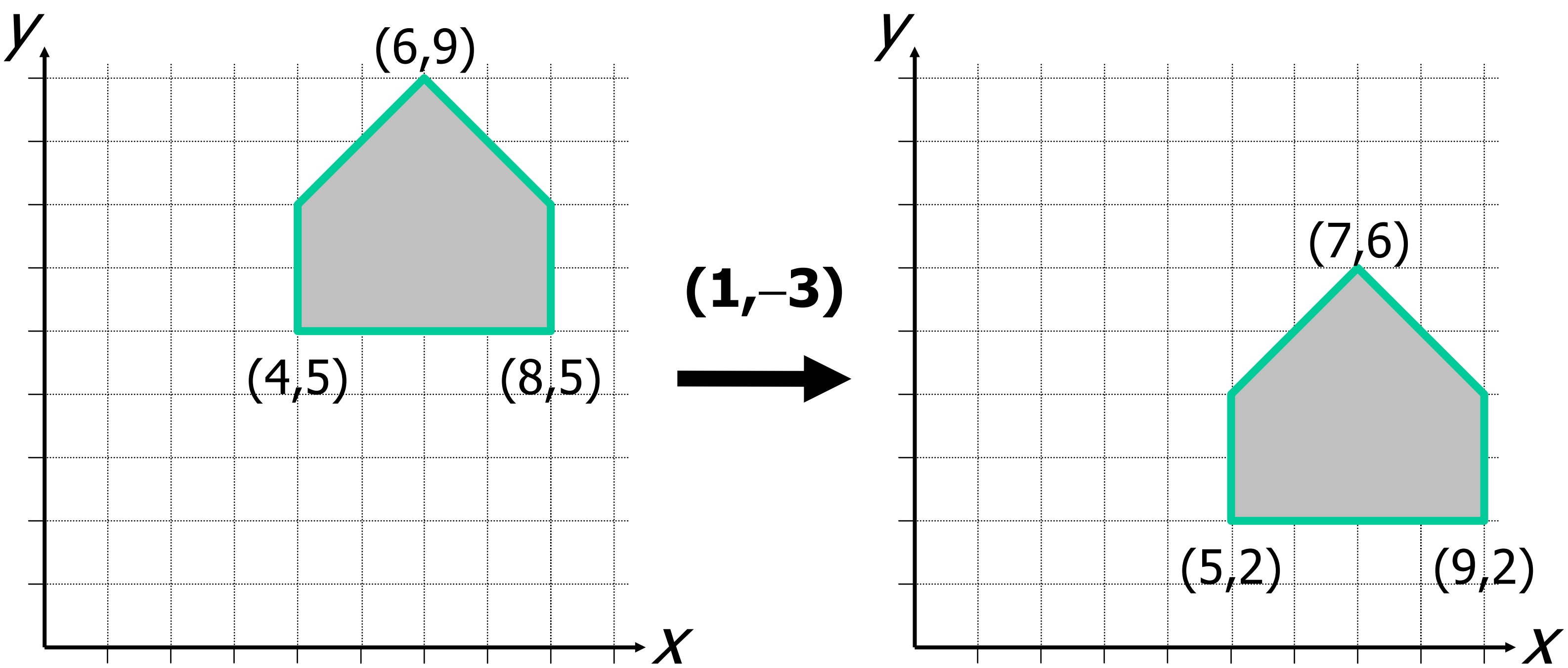
$$P = \begin{bmatrix} x \\ y \end{bmatrix}; \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix}; \quad \mathbf{T} = \begin{bmatrix} d_x \\ d_y \end{bmatrix};$$

$$P' = P + \mathbf{T}$$

- Con $\mathbf{T}$ *vettore traslazione*

Trasformazione di Traslazione

- Esempio di traslazione con vettore di traslazione $\mathbf{T}=(1,-3)$



Trasformazione di Scalatura

- Scelto un punto C (punto fisso) di riferimento, scalare una primitiva geometrica significa riposizionare rispetto a C tutti i suoi punti in accordo ai fattori di scala s_x (lungo l'asse x) e s_y (lungo l'asse y) scelti.

- Se il punto fisso è l'origine O degli assi, la trasformazione di P in P' si ottiene con:

$$x' = s_x \cdot x, \quad y' = s_y \cdot y$$

Trasformazione di Scalatura

- In notazione matriciale:

$$P' = \mathbf{S} \cdot P$$

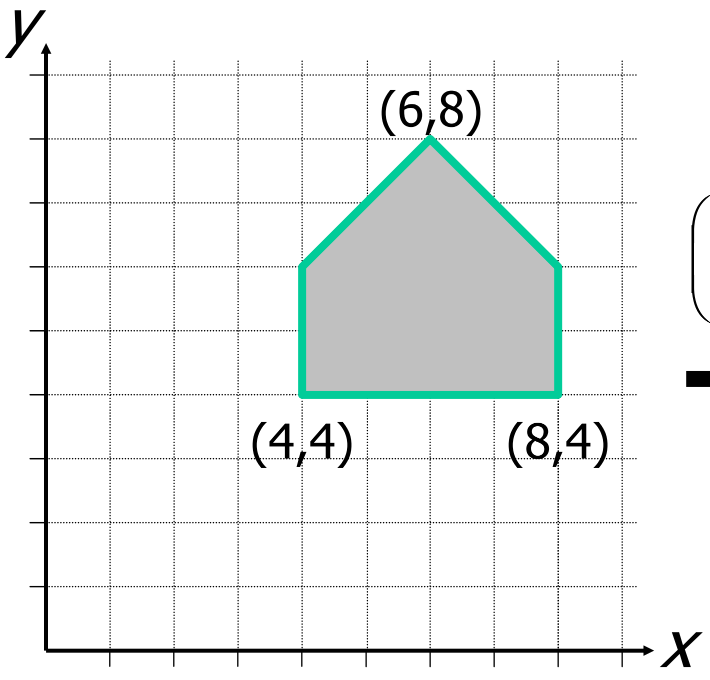
- dove

$$P = \begin{bmatrix} x \\ y \end{bmatrix}; \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix}; \quad \mathbf{S} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}.$$

- $\mathbf{S}$ pre-moltiplica P in quanto P è definito come vettore colonna

Trasformazione di Scalatura

- Esempio di scalatura di ½ lungo l'asse x e di ¼ lungo l'asse y



$$\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{4} \end{pmatrix}$$

Trasformazione di Scalatura

- Osservazioni:
  - Fattori di scala inferiori a 1 avvicinano l'oggetto al punto fisso di riferimento (origine);
  - Fattori di scala maggiori di 1 lo allontanano;
  - Se $s_x \neq s_y$ le proporzioni dell'oggetto non sono mantenute e si parla di *scalatura non uniforme;*
  - Se $s_x = s_y$ le proporzioni sono mantenute e si ha una *scalatura uniforme;*

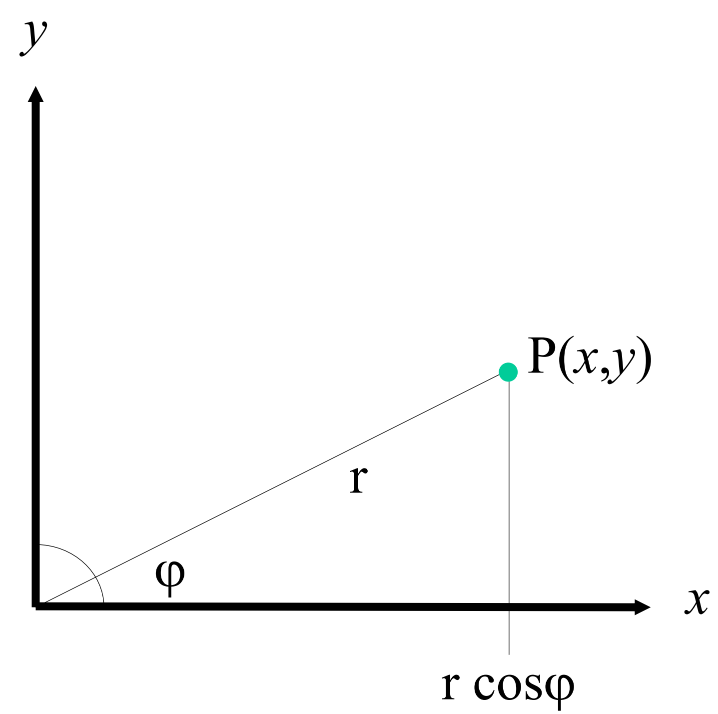
Trasformazione di Rotazione

- Fissato un punto C (pivot) di riferimento ed un verso di rotazione (orario, antiorario), ruotare una primitiva geometrica attorno a C significa muovere tutti i suoi punti nel verso assegnato in maniera che si conservi, per ognuno di essi, la **distanza** da C;
- Una rotazione di θ attorno all'origine O degli assi è definita come:

$$x' = x \cdot \cos\theta - y \cdot \operatorname{sen}\theta, \quad y' = x \cdot \operatorname{sen}\theta + y \cdot \cos\theta$$

Trasformazione di Rotazione

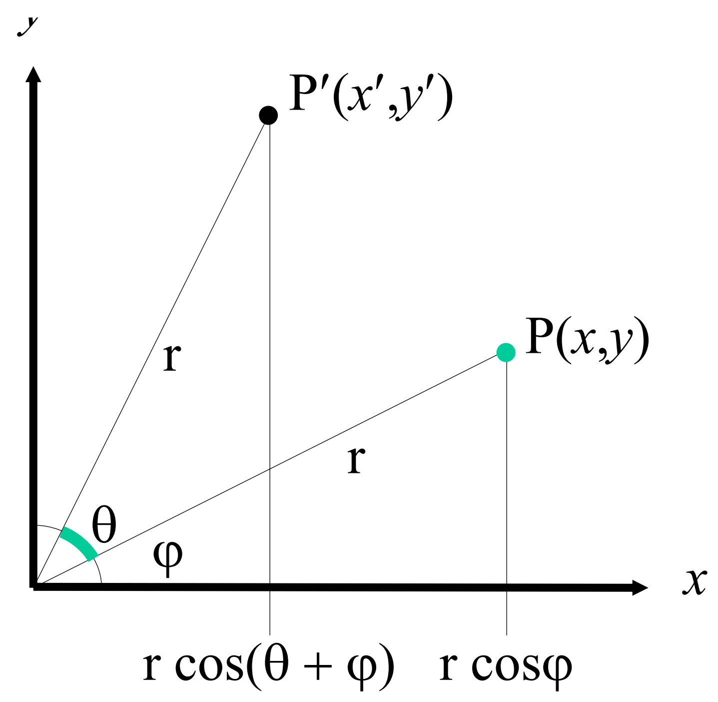
- La relazione tra P' e P si ricava trigonometricamente;
- Le coordinate di P possono essere espresse in coordinate polari:



$$x = r \cdot \cos\phi; \qquad y = r \cdot \sin\phi.$$

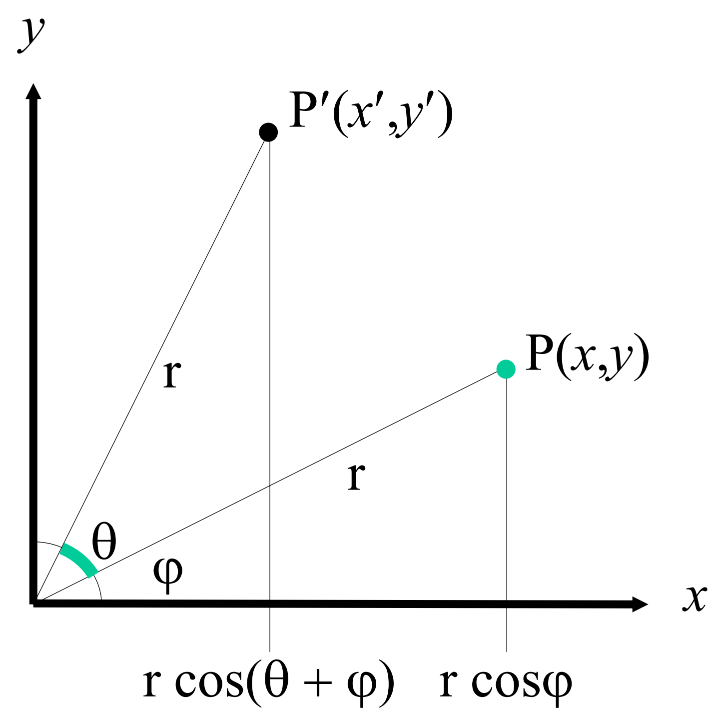
Trasformazione di Rotazione

$$
\begin{aligned}
x' &= r \cdot \cos(\theta + \phi) \\
&= r \cdot \cos\theta \cdot \cos\phi - r \cdot \sin\theta \cdot \sin\phi \\
&= r \cdot \cos\theta \cdot \frac{x}{r} - r \cdot \sin\theta \cdot \frac{y}{r} \\
&= x \cdot \cos\theta - y \cdot \sin\theta;
\end{aligned}
$$



Trasformazione di Rotazione

$$
\begin{aligned}
y' &= r \cdot \sin(\theta + \phi) \\
&= r \cdot \sin\theta \cdot \cos\phi + r \cdot \cos\theta \cdot \sin\phi \\
&= r \cdot \sin\theta \cdot \frac{x}{r} + r \cdot \cos\theta \cdot \frac{y}{r} \\
&= x \cdot \sin\theta + y \cdot \cos\theta.
\end{aligned}
$$



Trasformazione di Rotazione

- In notazione matriciale abbiamo:

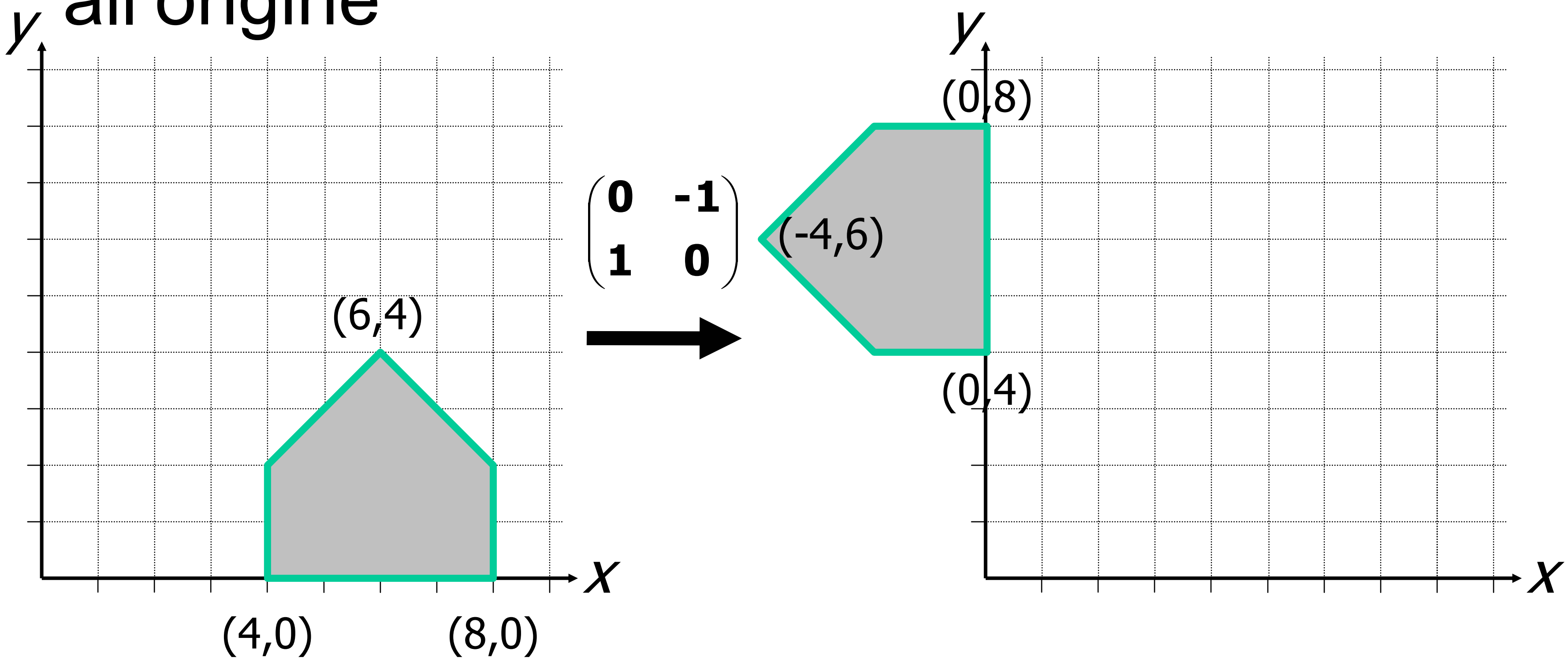
$$P' = \mathbf{R} \cdot P$$

- dove:

$$P = \left[\begin{array}{c} x \\ y \end{array} \right]; \quad P' = \left[\begin{array}{c} x' \\ y' \end{array} \right]; \quad \mathbf{R} = \left[\begin{array}{cc} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{array} \right].$$

Trasformazione di Rotazione

- Esempio di rotazione di $\pi/2$ attorno all'origine



Trasformazione di Rotazione

- Osservazioni:
  - Gli angoli sono considerati positivi quando misurati in senso antiorario;
  - Per le rotazioni di angoli negativi (senso orario) si ricorre alle identità:

$$\cos(-\theta) = cos(\theta); \quad \sin(-\theta) = -sin(\theta).$$

Coordinate omogenee

- Le trasformazioni geometriche possono essere applicate in sequenza; la presenza di una *somma di vettori* (traslazione):

$$P' = P + \mathbf{T}$$

- e di *moltiplicazioni* (scalatura e rotazione):

$$P' = \mathbf{S} \cdot P$$

$$P' = \mathbf{R} \cdot P$$

- rende disomogenea la concatenazione di trasformazioni. P`=S(R(P+T))+T’

Coordinate omogenee

- Il punto P di coordinate (x,y) è rappresentato in coordinate omogenee come (x_h,y_h,w), dove:

$$x = x_h/w; \quad y = y_h/w; \quad \text{con} \quad w \neq 0.$$

- Due punti di coordinate (x, y, w) e (x', y', w') rappresentano lo stesso punto del piano se e solo se le coordinate di uno sono multiple delle corrispondenti coordinate dell'altro;
- Almeno uno dei valori x, y, o w deve essere diverso da 0;
- Quando $w = 1$ (forma canonica) coordinate cartesiane ed omogenee coincidono.
- Con $(x, y, w \neq 0)$ si rappresentano *punti*, con $(x, y, 0)$ si rappresentano *vettori.*

Trasformazioni e coordinate omogenee

- Nella notazione in coordinate omogenee possiamo riscrivere le trasformazioni geometriche di base come:
- *Trasformazione di traslazione*:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Concatenazione di due traslazioni

$$\begin{bmatrix} 1 & 0 & d_{x1} \\ 0 & 1 & d_{y1} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & d_{x2} \\ 0 & 1 & d_{y2} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_{x1} + d_{x2} \\ 0 & 1 & d_{y2} + d_{y2} \\ 0 & 0 & 1 \end{bmatrix}$$

Trasformazioni e coordinate omogenee

- *Trasformazione di scalatura*:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- *Trasformazione di rotazione*:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Altre trasformazioni: riflessione

- Riflessione rispetto all'asse *x*:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Riflessione rispetto all'asse *y*:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Riflessione rispetto all'origine degli assi:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Altre trasformazioni: deformazione (shear)

- Deformazione rispetto all'asse x:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Deformazione rispetto all'asse y:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Riflessione rispetto entrambi gli assi:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Trasformazione di deformazione (shear)

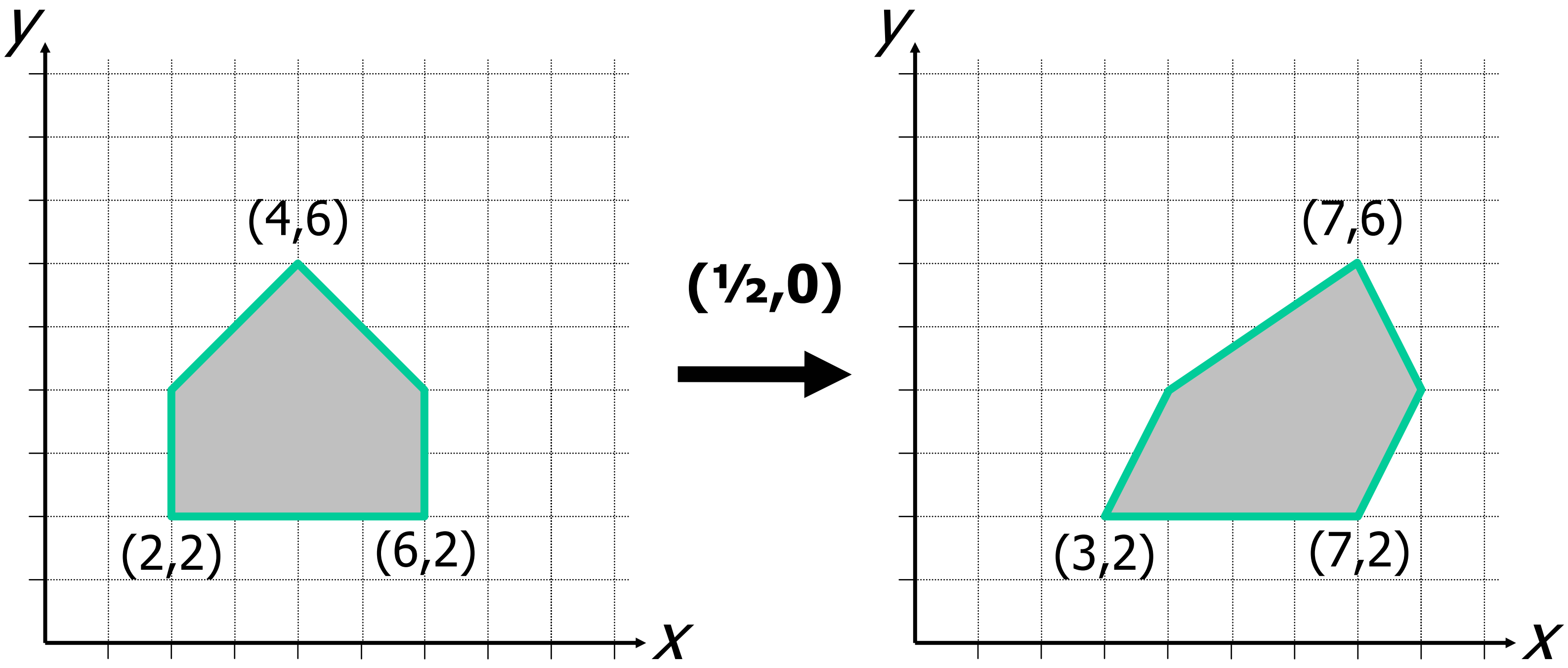
- Dalle relazioni:

$$x' = x + ay \quad y' = y + bx$$

risulta evidente come la deformazione lungo l'asse x sia linearmente dipendente dalla coordinata y e la deformazione in y sia strettamente correlata all'ascissa del punto.

Trasformazione di deformazione (shear)

- Esempio di deformazione con: $a=\frac{1}{2}$ e $b=0$



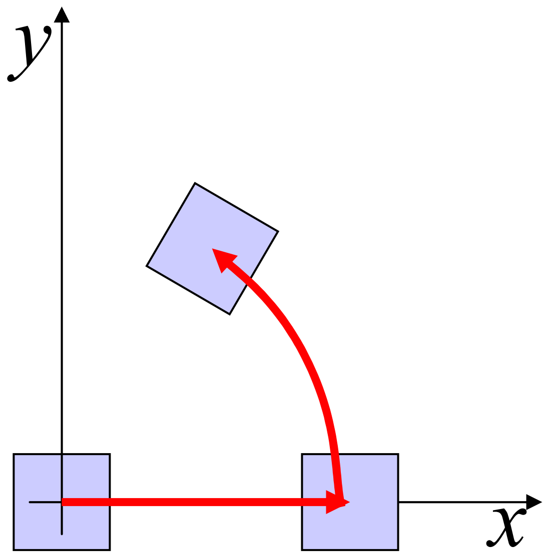
Composizione di trasformazioni

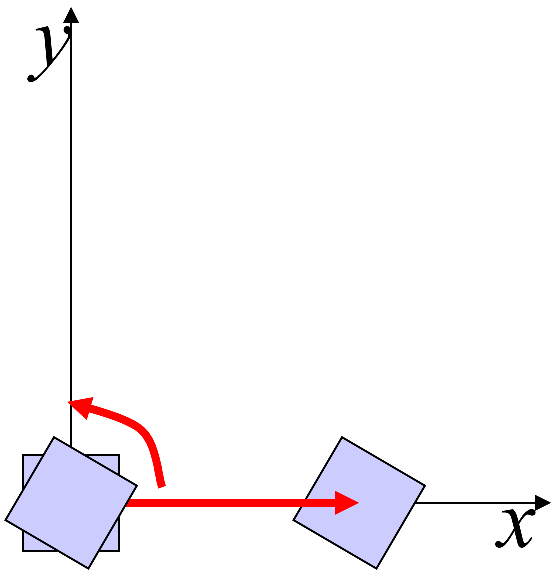
- La rappresentazione in coordinate omogenee permette la concatenazione di trasformazioni;
- L’ordina di concatenazione è importante perché le trasformazioni geometriche sono associative ma non sono (in generale) commutative;
- La corretta sequenza delle trasformazioni $\mathbf{T_1}$, $\mathbf{T_2}$, $\mathbf{T_3}$ e $\mathbf{T_4}$ si ottiene componendo $\mathbf{T}$ come:

$$\mathbf{T} = \mathbf{T_4} \cdot \mathbf{T_3} \cdot \mathbf{T_2} \cdot \mathbf{T_1}$$

Composizione di trasformazioni

- **Non commutatività** della composizione di trasformazioni: traslazione seguita da rotazione attorno all'origine (sinistra) e rotazione intorno all'origine seguita da traslazione (destra).





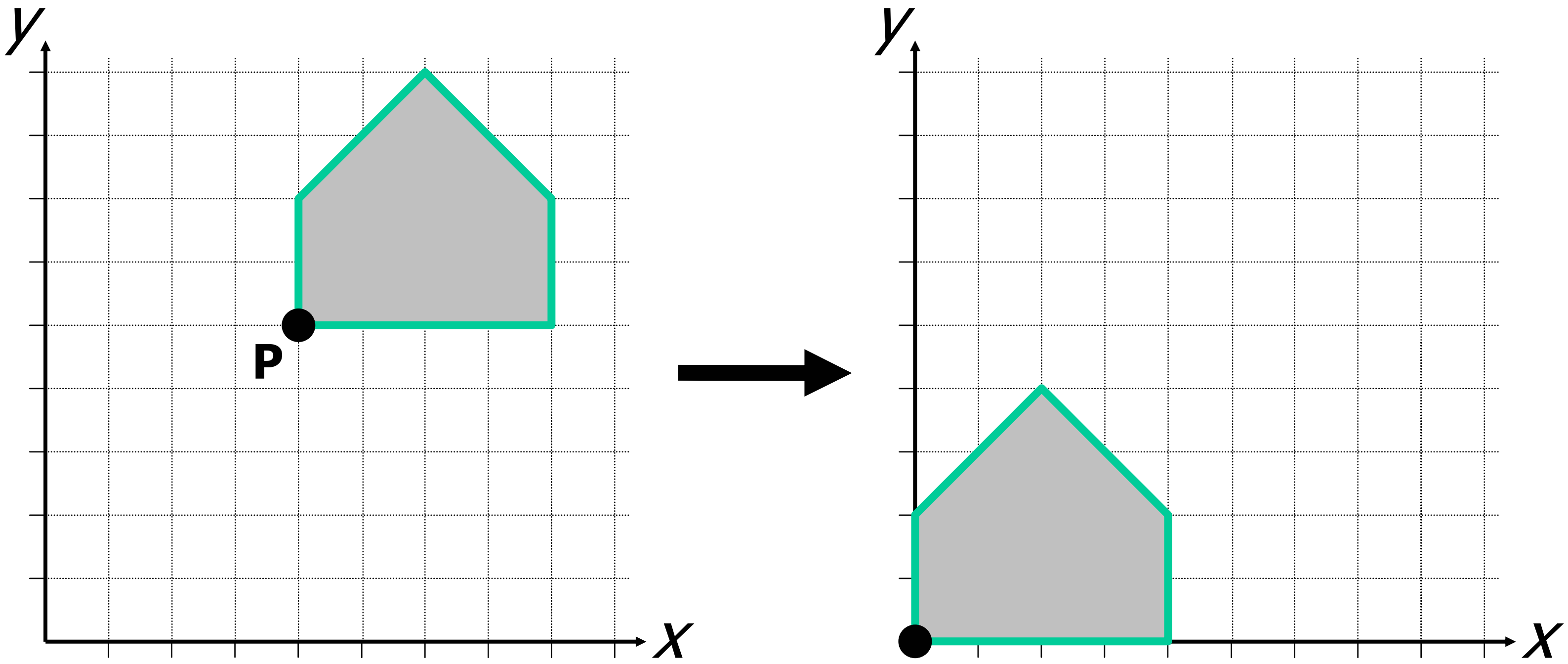
Composizione di trasformazioni

- Rotazione oraria di un angolo θ attorno ad un punto P generico:
  - Traslazione che muove P nell'origine degli assi;
  - Rotazione attorno all'origine;
  - Traslazione opposta alla precedente che riporta P nella sua posizione originale.

$$\mathbf{R}_{g} = \begin{bmatrix} 1 & 0 & \mathrm{P}_x \\ 0 & 1 & \mathrm{P}_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -sen\,\theta & 0 \\ sen\,\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -\mathrm{P}_x \\ 0 & 1 & -\mathrm{P}_y \\ 0 & 0 & 1 \end{bmatrix}$$

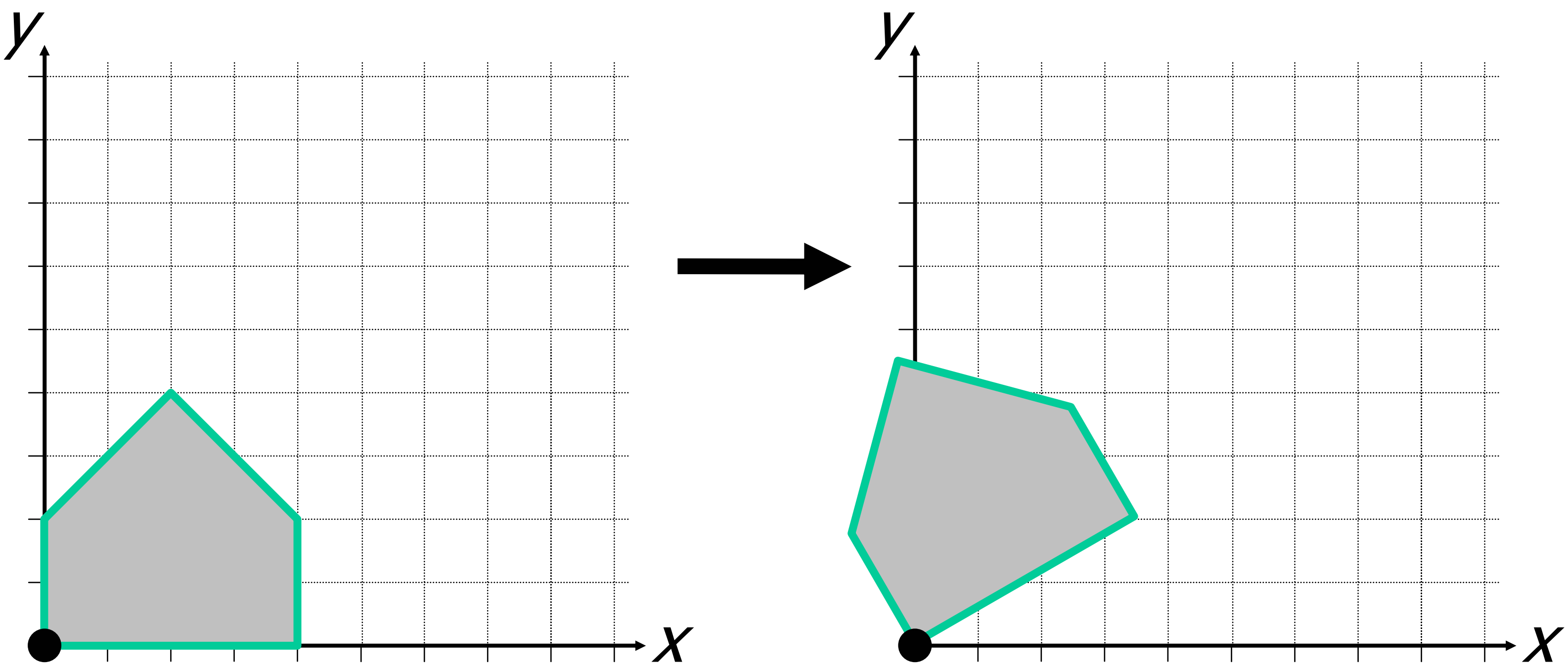
Composizione di trasformazioni

- Rotazione θ attorno ad un punto P.
  - *Passo 1:* Traslazione di P nell'origine degli assi



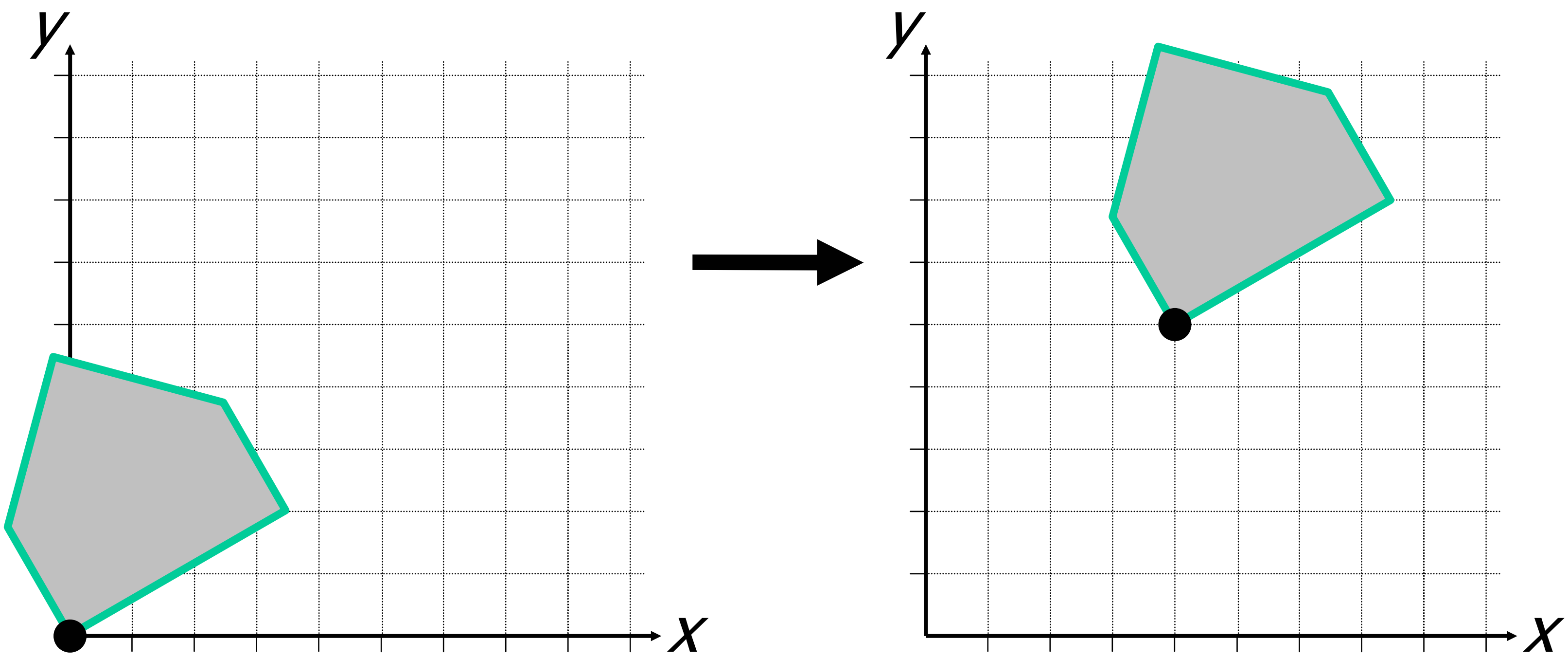
Composizione di trasformazioni

- Rotazione θ attorno ad un punto P.
  - *Passo 2:* Rotazione attorno all'origine ($\theta = \pi/6$)



Composizione di trasformazioni

- Rotazione θ attorno ad un punto P.
  - *Passo 3:* Traslazione opposta alla precedente



Composizione di trasformazioni

- Trasformazione di scalatura attorno ad un punto P generico:
  - Traslazione che muove P nell'origine degli assi;
  - Trasformazione di scala attorno all'origine;
  - Traslazione opposta alla precedente che riporta P nella sua posizione originale.

$$\mathbf{R}_g = \begin{bmatrix} 1 & 0 & P_x \\ 0 & 1 & P_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -P_x \\ 0 & 1 & -P_y \\ 0 & 0 & 1 \end{bmatrix}$$

Cosa preservano le trasformazioni

trasformazione		Lunghezze	Angoli	Rapporti tra lunghezze	Collinearita’
traslazione		SI	SI	SI	SI
rotazione		SI	SI	SI	SI
scalatura uniforme		NO	SI	SI	SI
scalatura non uniforme		NO	NO	SI	SI
shear		NO	NO	SI	SI
proiezione ortigonale		NO	NO	SI	SI
trasf. affine generica		NO	NO	SI	SI
proiezione prospettica		NO	NO	NO	SI

Sistema di riferimento (*frame*)

- Definito da
  - un punto base (origine) p_0
  - e una base vettoriale $\{ \overbrace{v_0 , v_1 , v_2}^{\text{lin indip}} \}$

- Posso esprimere (univocamente) ogni **punto** p come:

$$p = v_0 \, \eta_0 + v_1 \, \eta_1 + v_2 \, \eta_2 + p_0$$

- cioè: $$p = \underbrace{[\ \eta_0 \ , \ \eta_1 \ , \ \eta_2 \ , \ 1]}_{\text{coordinate omogenee di } p} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ p_0 \end{bmatrix}$$

Sistema di riferimento (*frame*)

- Definito da
  - un punto base (origine) p_0
  - e una base vettoriale $\{ v_0 , v_1 , v_2 \}$

- Posso esprimere (univocamente) ogni **vettore** v come:

$$v = v_0\, \eta_0 + v_1\, \eta_1 + v_2\, \eta_2 + \cancel{p_0}$$

- cioè:

$$v = \underbrace{[\, \eta_0 \;,\; \eta_1 \;,\; \eta_2 \;,\; 0 \,]}_{\text{coordinate omogenee di } v} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ p_0 \end{bmatrix}$$

Cambio di frame (cambio di sistema di riferimento)

- Dati due sistemi di riferimento: $\{v_1, v_2, v_3, p_0\}$ $\{u_1, u_2, u_3, q_0\}$

$$p = [a_1, a_2, a_3, 1] \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ p_0 \end{bmatrix} = [b_1, b_2, b_3, 1] \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ q_0 \end{bmatrix}$$

coordinate di p nel primo sist. di rif.:

coordinate di p nel sec. sist. di rif.:

- Esprimo uno in termini dell'altro:

$$u_1 = \gamma_{11} v_1 + \gamma_{12} v_2 + \gamma_{13} v_3$$

$$u_2 = \gamma_{21} v_1 + \gamma_{22} v_2 + \gamma_{23} v_3$$

$$u_3 = \gamma_{31} v_1 + \gamma_{32} v_2 + \gamma_{33} v_3$$

$$q_0 = \gamma_{41} v_1 + \gamma_{42} v_2 + \gamma_{43} v_3 + p_0$$

matrice di cambio di frame

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{21} & \gamma_{31} & \gamma_{41} \\ \gamma_{12} & \gamma_{22} & \gamma_{32} & \gamma_{42} \\ \gamma_{13} & \gamma_{23} & \gamma_{33} & \gamma_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 1 \end{bmatrix}$$

Cambio di frame (cambio di sistema di riferimento)

caso particlare: arrivo a sistema di riferimento canonico

- Dati due sistemi di riferimento: $\{v_1, v_2, v_3, p_0\}$ $\{u_1, u_2, u_3, q_0\}$

$$p = [a_1, a_2, a_3, 1] \begin{bmatrix} 1,0,0 \\ 0,1,0 \\ 0,0,1 \\ 0,0,0 \end{bmatrix} = [b_1, b_2, b_3, 1] \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ q \end{bmatrix}$$

coordinate di p
nel primo sist. di rif.:

coordinate di p
nel sec. sist. di rif.:

- Esprimo uno in termini dell'altro:

$$u_1 = (\gamma_{11}, \gamma_{12}, \gamma_{13})$$
$$u_2 = (\gamma_{21}, \gamma_{22}, \gamma_{23})$$
$$u_3 = (\gamma_{21}, \gamma_{22}, \gamma_{23})$$
$$q_0 = (\gamma_{21}, \gamma_{22}, \gamma_{23})$$

matrice di cambio
di frame

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ 1 \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & u_3 & q \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ 1 \end{bmatrix}$$

Cambio di frame

- In realtà tutte le transf. affini lineari si possono vedere come un *cambio di frame*
  - compresse quelle viste:
    - traslazione
    - scaling (uniforme o no)
    - shearing
    - rotazioni

Prodotto scalare

- Prodotto Scalare ("dot-product", "internal product"):

vettore x vettore → scalare

$$(\alpha_x, \alpha_y, \alpha_z) \cdot (\beta_x, \beta_y, \beta_z) = \alpha_x \beta_x + \alpha_y \beta_y + \alpha_z \beta_z$$

Proprietà

commuta $u \cdot v = v \cdot u$

lineare 1/2 $(u + v) \cdot w = u \cdot w + v \cdot w$

lineare 2/2 $(\alpha u) \cdot v = u \cdot (\alpha v) = \alpha (u \cdot v)$

Prodotto scalare

- Prodotto Scalare ("dot-product", "internal product"):

vettore x vettore → scalare

$$(\alpha_x, \alpha_y, \alpha_z) \cdot (\beta_x, \beta_y, \beta_z) = \alpha_x \beta_x + \alpha_y \beta_y + \alpha_z \beta_z$$

Proprietà

per il modulo: $|v| = \sqrt{v \cdot v}$

quindi, per calcolare una distanza tra punti: $|P - Q| = \sqrt{(P - Q) \cdot (P - Q)}$

e anche: $v \cdot v = 0 \Rightarrow v = (0,0,0)$

Prodotto scalare

- Prodotto Scalare ("dot-product", "internal product"):

vettore x vettore → scalare

$$(\alpha_x, \alpha_y, \alpha_z) \cdot (\beta_x, \beta_y, \beta_z) = \alpha_x \beta_x + \alpha_y \beta_y + \alpha_z \beta_z$$

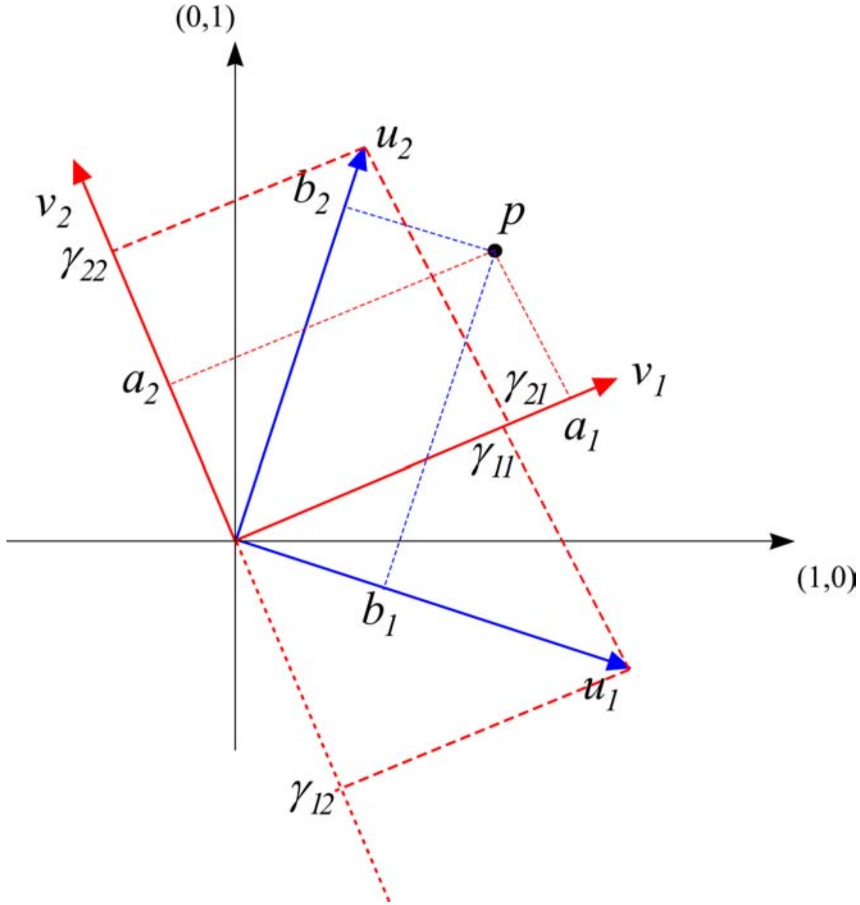
Proprietà

molto utilmente: $u \cdot v = |u||v| \cos\theta$

quindi se u e v non sono nulli: $u \cdot v = 0 \Leftrightarrow u$ e v ortogonali

e, se u e v sono normalizzati: $u \cdot v = \cos\theta$

Cambio di frame: esempio



$$u_1 = \gamma_{11} v_1 + \gamma_{12} v_2$$

$$u_2 = \gamma_{21} v_1 + \gamma_{22} v_2$$

$$dove:$$

$$\gamma_{11} = u_1 \cdot v_1 \quad \gamma_{12} = u_1 \cdot v_2$$

$$\gamma_{21} = u_2 \cdot v_1 \quad \gamma_{22} = u_2 \cdot v_2$$

$$quindi:$$

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \gamma_{11} & \gamma_{21} \\ \gamma_{12} & \gamma_{22} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

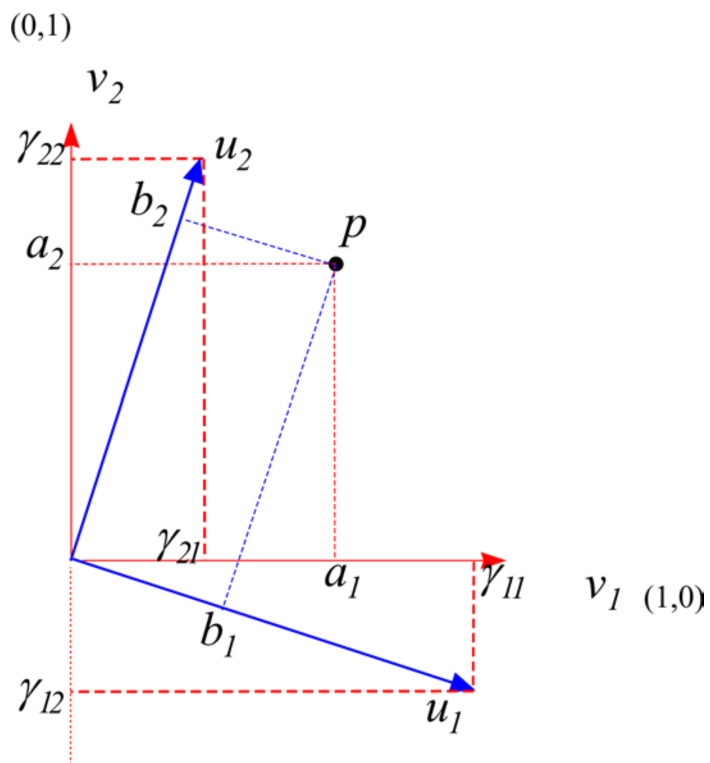
$$a = R \cdot b$$

$$ma\ anche$$

$$b = R^T a$$

$$poiche'\ R^{-1} = R^T$$

Cambio di frame: esempio



$$u_1 = \gamma_{11} v_1 + \gamma_{12} v_2$$

$$u_2 = \gamma_{21} v_1 + \gamma_{22} v_2$$

$dove:$

$$\gamma_{11} = u_1 \cdot (1,0) \quad \gamma_{12} = u_1 \cdot (0,1)$$

$$\gamma_{21} = u_2 \cdot (1,0) \quad \gamma_{22} = u_2 \cdot (0,1)$$

$quindi:$

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} u_{1x} & u_{2x} \\ u_{1y} & u_{2y} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

$$a = R \cdot b$$

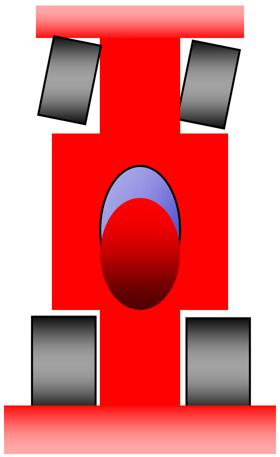
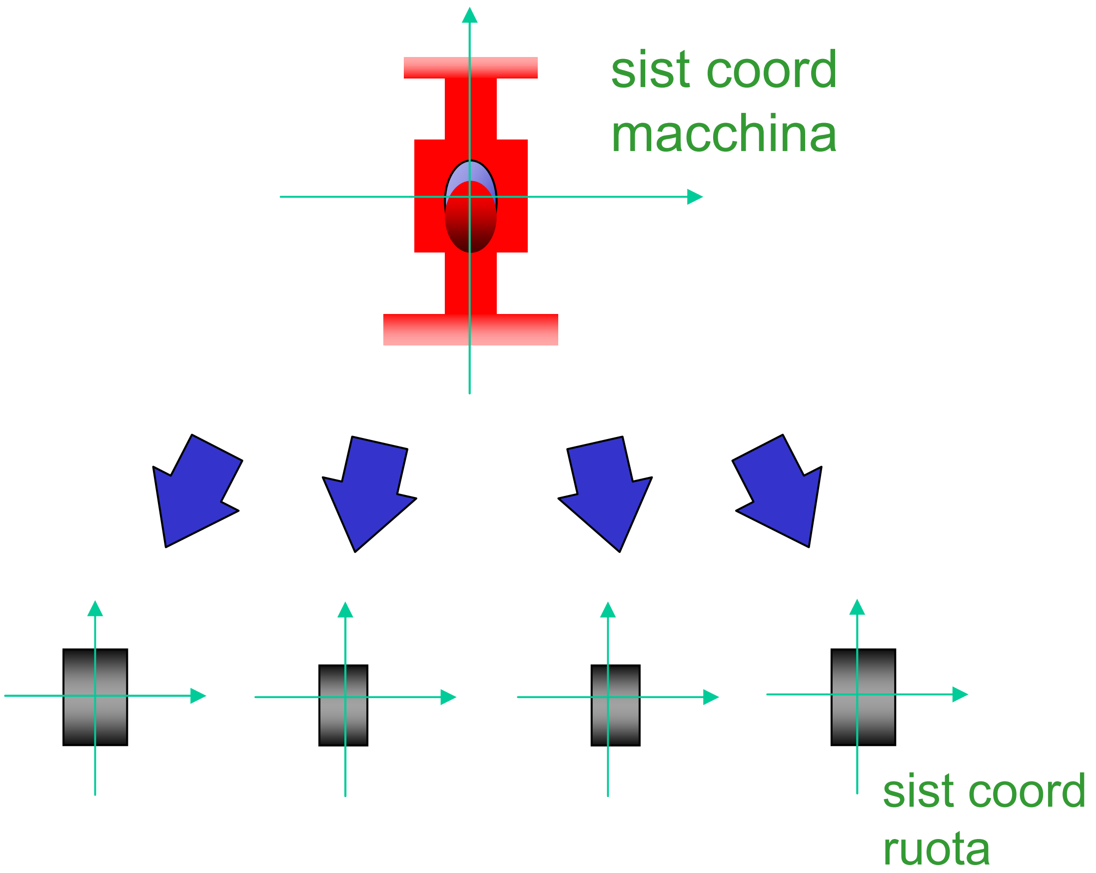
$ma\ anche$

$$b = R^T a$$

$$poiche'\ R^{-1} = R^T$$

Nota: R e’ la rotazione che porta il frame U a coincidere con il frame V

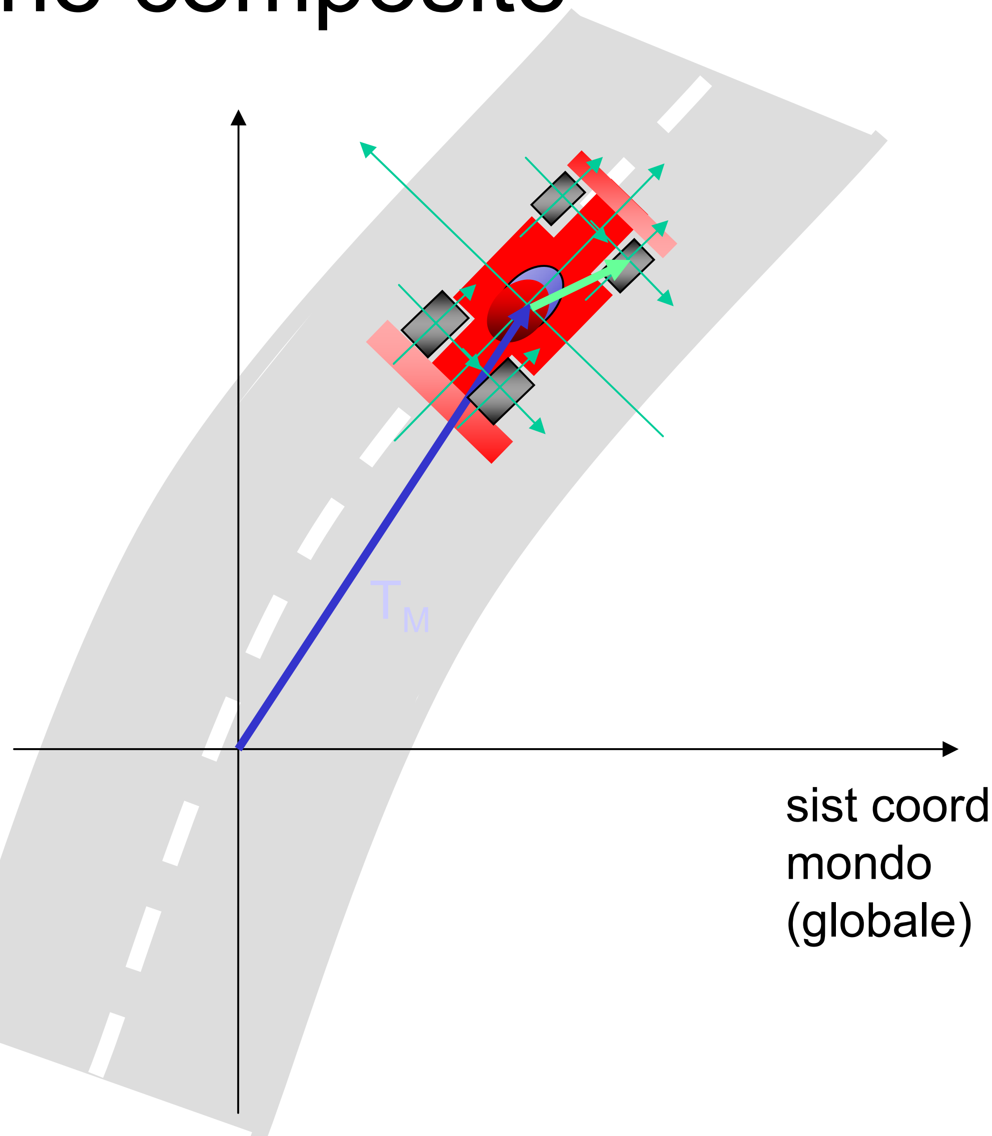
Usiamo I cambi di frame: scene composite

Rendering di scene composite

M matrice di modelling usata

T_M matr traslazione macchina
R_M matr rotazione macchina
T_{Ri} matr traslazione
R_{Ri} matr rotazione ruota i

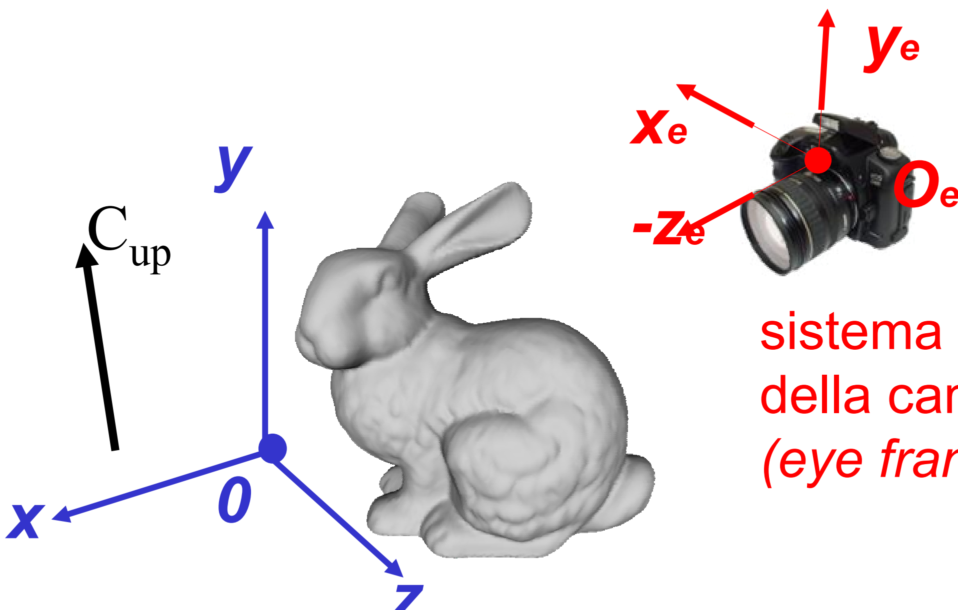


Transformazione di vista: esercizio!

- Input:
  1) camera position C_{pos}
  2) direzione di vista C_{dir}
  3) vettore di alto C_{up}
- Output:

  Matrice di Trasformazione

  world frame $\longrightarrow$ *eye frame*



sistema di riferimento della camera *(eye frame)*

sistema di riferimento globale *(wolrd frame)*

Prodotto vettoriale

- Prodotto Vettoriale ("cross-product", "external product"):

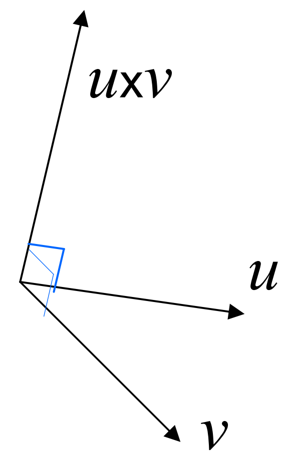
vettore x vettore → vettore

$$(\alpha_x, \alpha_y, \alpha_z) \times (\beta_x, \beta_y, \beta_z) = \begin{bmatrix} \alpha_y \beta_z - \alpha_z \beta_y \\ \alpha_z \beta_x - \alpha_x \beta_z \\ \alpha_x \beta_y - \alpha_y \beta_x \end{bmatrix}$$

Proprietà

non commuta: $u \times v = -(v \times u)$

il risultato è ortogonale ad entrambi gli operandi: $(u \times v) \cdot v = (u \times v) \cdot u = 0$



Prodotto vettoriale

- Prodotto Vettoriale ("cross-product", "external product"):

vettore x vettore → vettore

$$(\alpha_x, \alpha_y, \alpha_z) \times (\beta_x, \beta_y, \beta_z) = \begin{bmatrix} \alpha_y \beta_z - \alpha_z \beta_y \\ \alpha_z \beta_x - \alpha_x \beta_z \\ \alpha_x \beta_y - \alpha_y \beta_x \end{bmatrix}$$

Proprietà

molto utilmente: $|u \times v| = |u||v| \operatorname{sen}\theta$

quindi se u e v non sono nulli: $|u \times v| = 0 \Leftrightarrow u$ e v allineati

e, se u e v sono normalizzati: $|u \times v| = \operatorname{sen}\theta$

Esercizio: transformazione di vista

- Input:

1) camera position C_{pos}

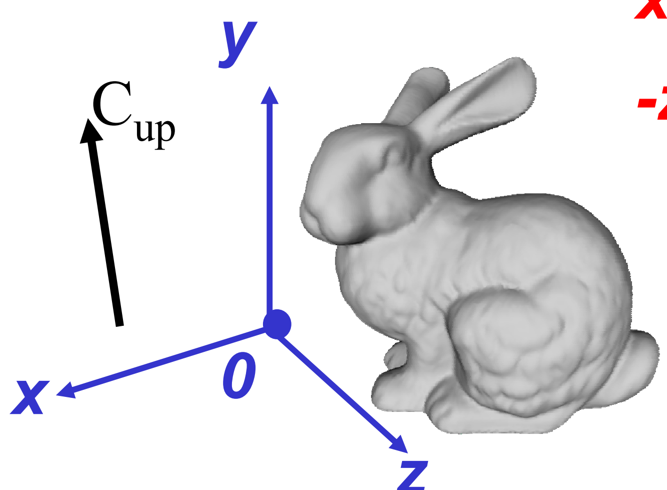
2) direzione di vista C_{dir}

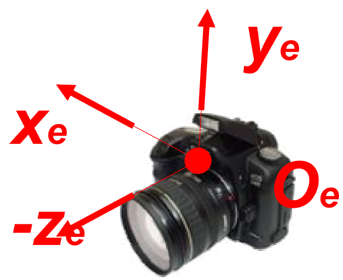
3) vettore di alto C_{up}

- Output:

Matrice di Trasformazione

world frame ⟶ *eye frame*





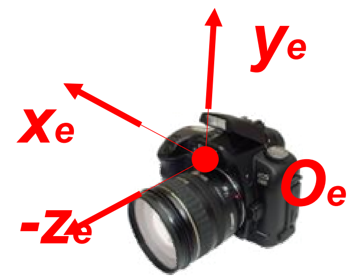
sistema di riferimento
della camera
(eye frame)

sistema di riferimento
globale *(wolrd frame)*

Transformazione di vista: costruire il sistema di riferimento

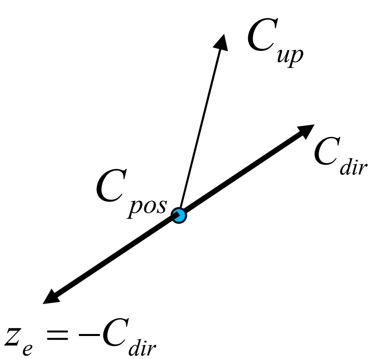
- Input:

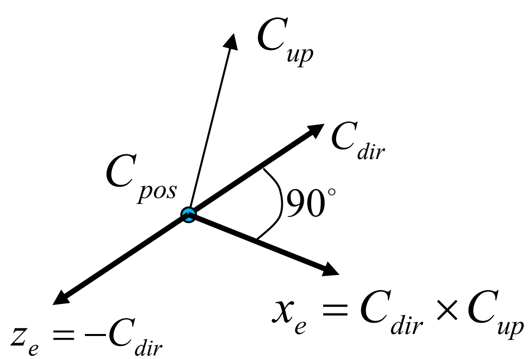
1) camera position (punto) C_{pos}
2) direzione di vista (vettore) C_{dir}
3) vettore di alto (vettore) C_{up}

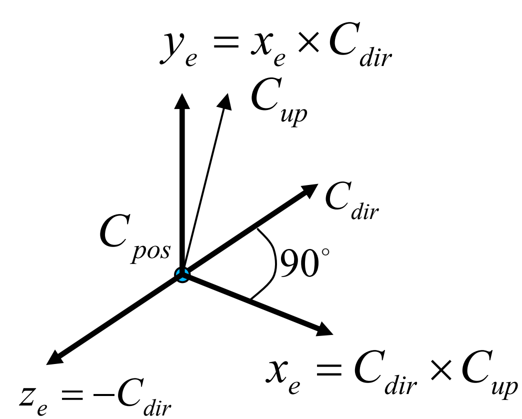


sistema di riferimento della camera
(eye frame)

- Per comodita' dell'utente, non si richiede la specifica del frame della camera
- Solo il punto di vista, la direzione di vista e dov'e' "su"
- Per la stessa ragione "su" non si richiede che sia perpendicolare alla direzione

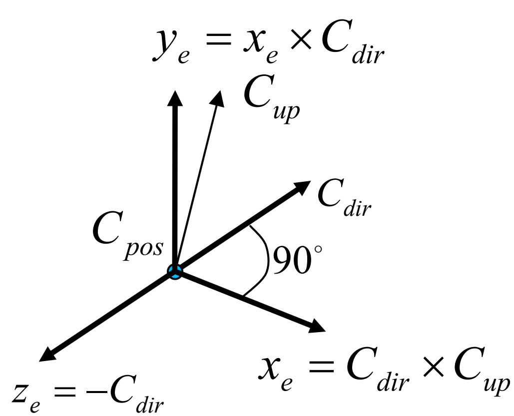


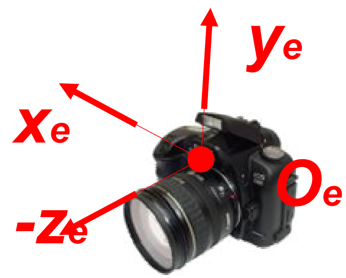




Transformazione di vista: costruire il sistema di riferimento

$$M_{eye} = \begin{bmatrix} C_{dir} \times C_{up} & 0 \\ (C_{dir} \times C_{up}) \times C_{dir} & 0 \\ -C_{dir} & 0 \\ \multicolumn{2}{c}{C_{pos}} \end{bmatrix}$$





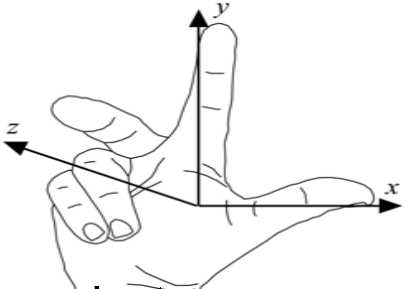
sistema di riferimento della camera *(eye frame)*

Funziona sempre?

Funziona, a meno che…..

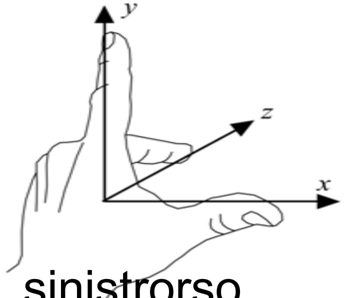
Il sistema di riferimento nello spazio

- Per capire quale sara' un'immancabile fonte di errori
- Un sistema di riferimento (*frame*) puo' essere destrorso o sinistrorso



destrorso

$$z = x \times y$$



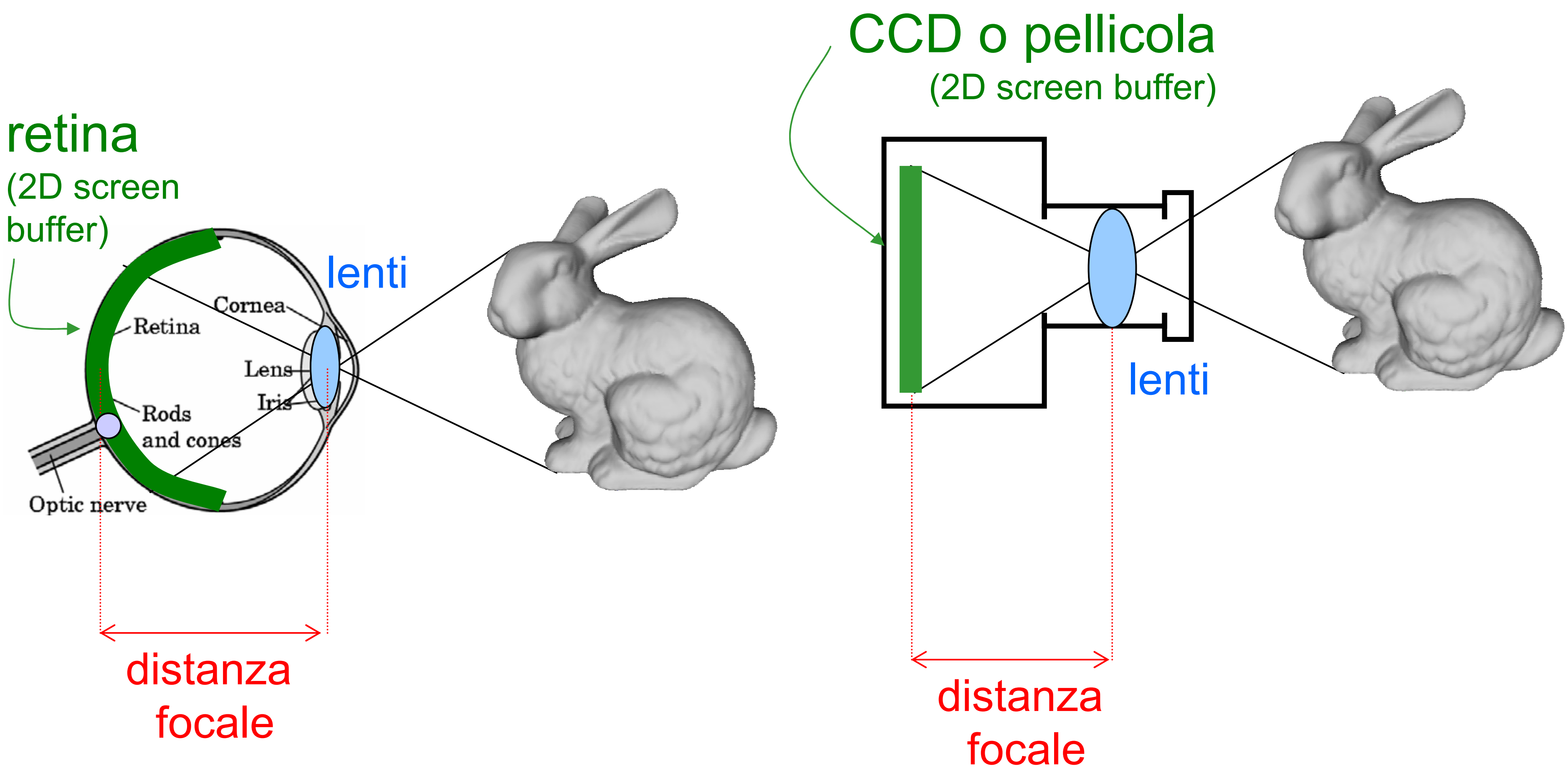
sinistrorso

$$z = y \times x$$

Convenzione: il sistema di riferimento adottato e' destrorso

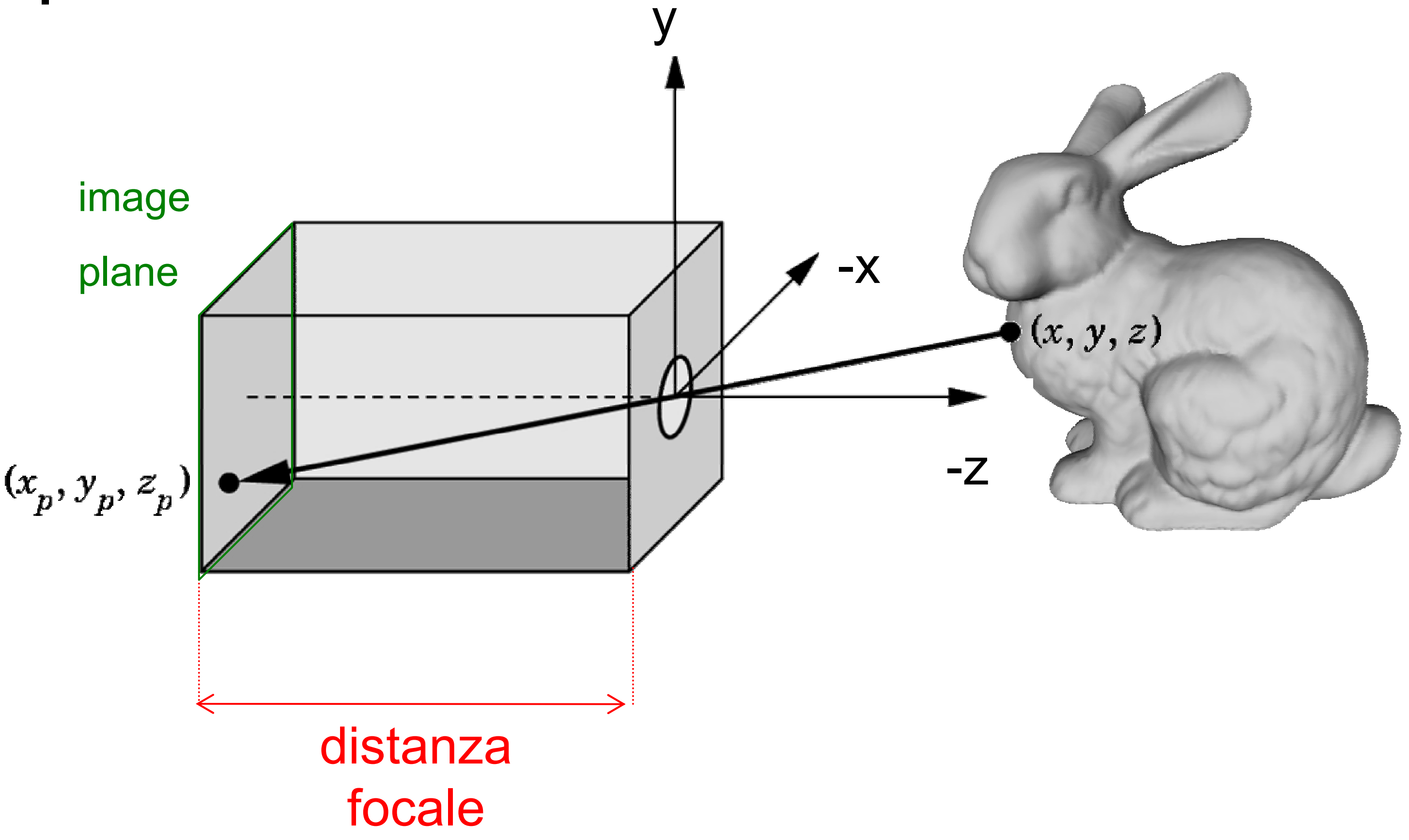
Visione prospettica: occhio e macchina fotografica

- il concetto è lo stesso:



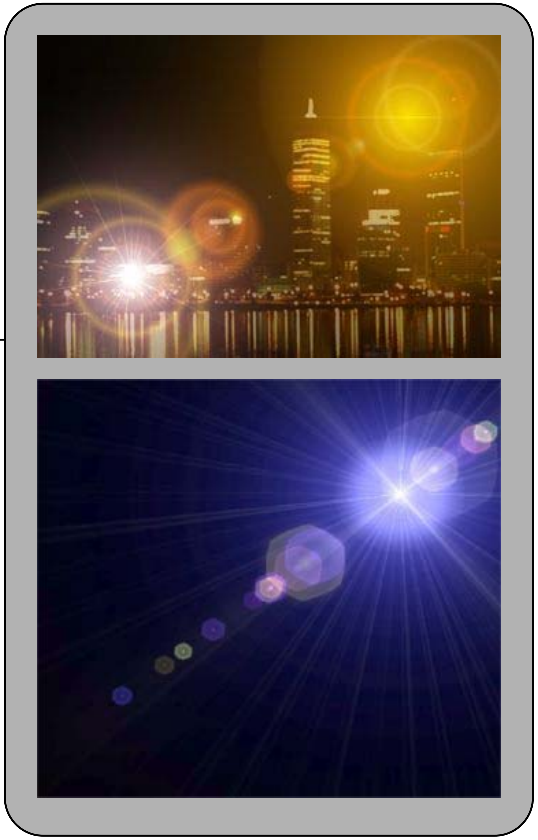
Modello semplificato usato in CG:

- pin-hole camera

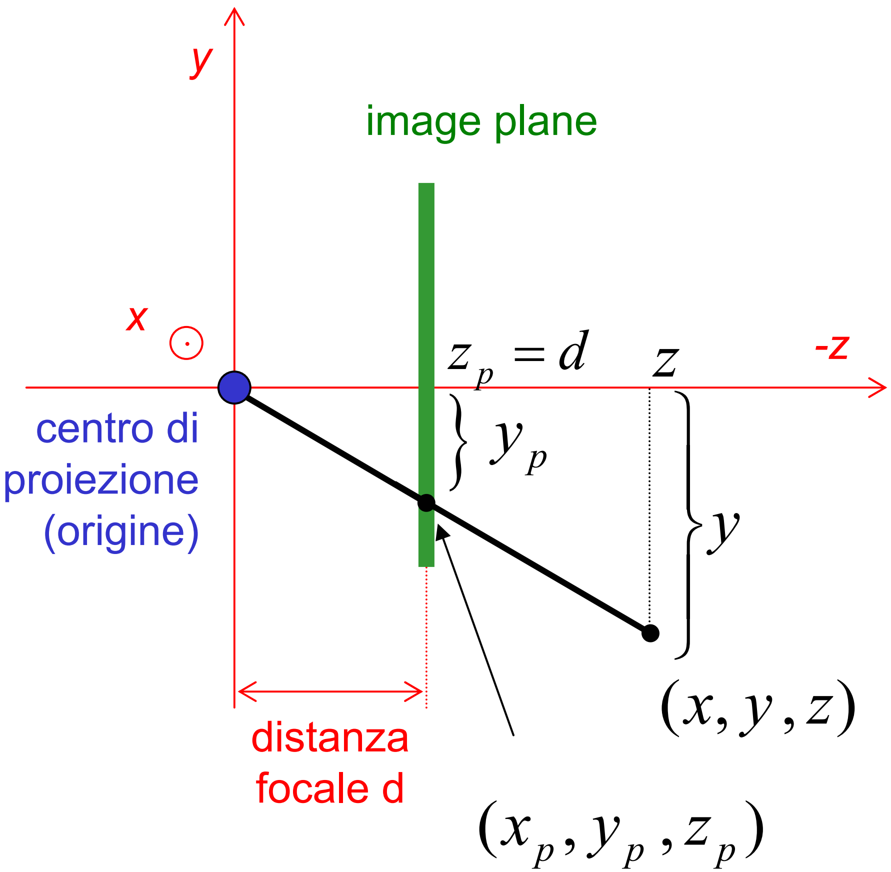


Modello semplificato usato in CG: nota: **niente lenti**

- le lenti servivano a "simulare" una pin-hole camera
- non modellandole,
  non abbiamo
  i "difetti" di questa simulazione:
  - range di fuoco finito
  - flares
  - distorsioni radiali



Proiezione prospettica



triangoli simili

$$\frac{y_p}{d} = \frac{y}{z} \Rightarrow y_p = \frac{y}{z/d}$$

$$\frac{x_p}{d} = \frac{x}{z} \Rightarrow x_p = \frac{x}{z/d}$$

$$z_p = d$$

Nota:

non è lineare né affine;

non è neanche reversibile.

Proiezione prospettica: forma matriciale

matrice di trasformazione
per la proiezione prospettica: $\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$

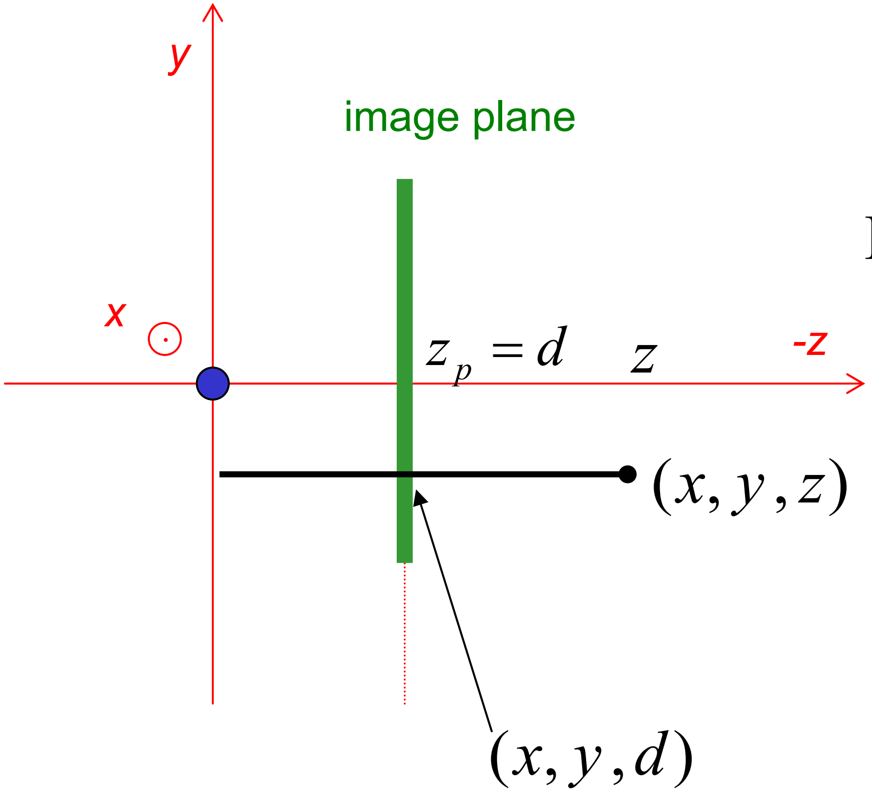
questa operazione
si fa per ultima.
La 3 e 4
componente
ci saranno utili !

$$\mathbf{P}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ z/d \end{bmatrix}$$

divisione per
4ta comp

$$\begin{bmatrix} \dfrac{x}{z/d} \\ \dfrac{y}{z/d} \\ d \\ 1 \end{bmatrix}$$

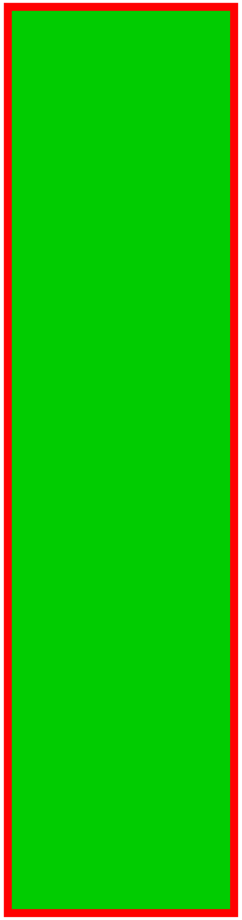
Proiezione ortogonale

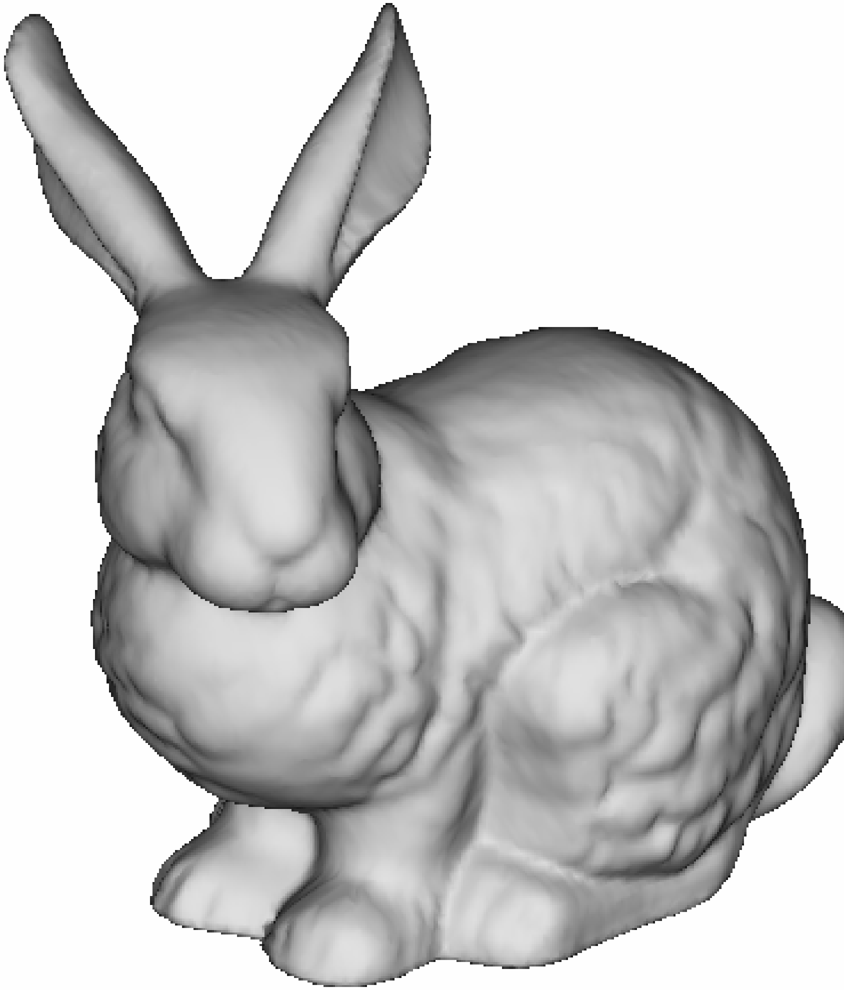


$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix}$$

La proiezione prospettica

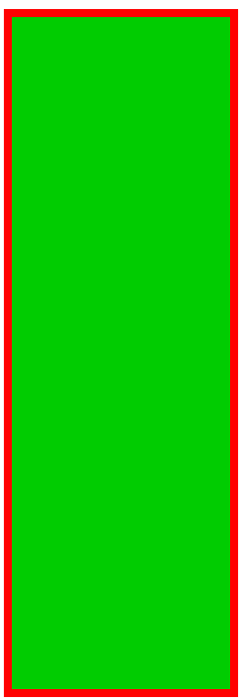
- Al variare della distanza focale (d)

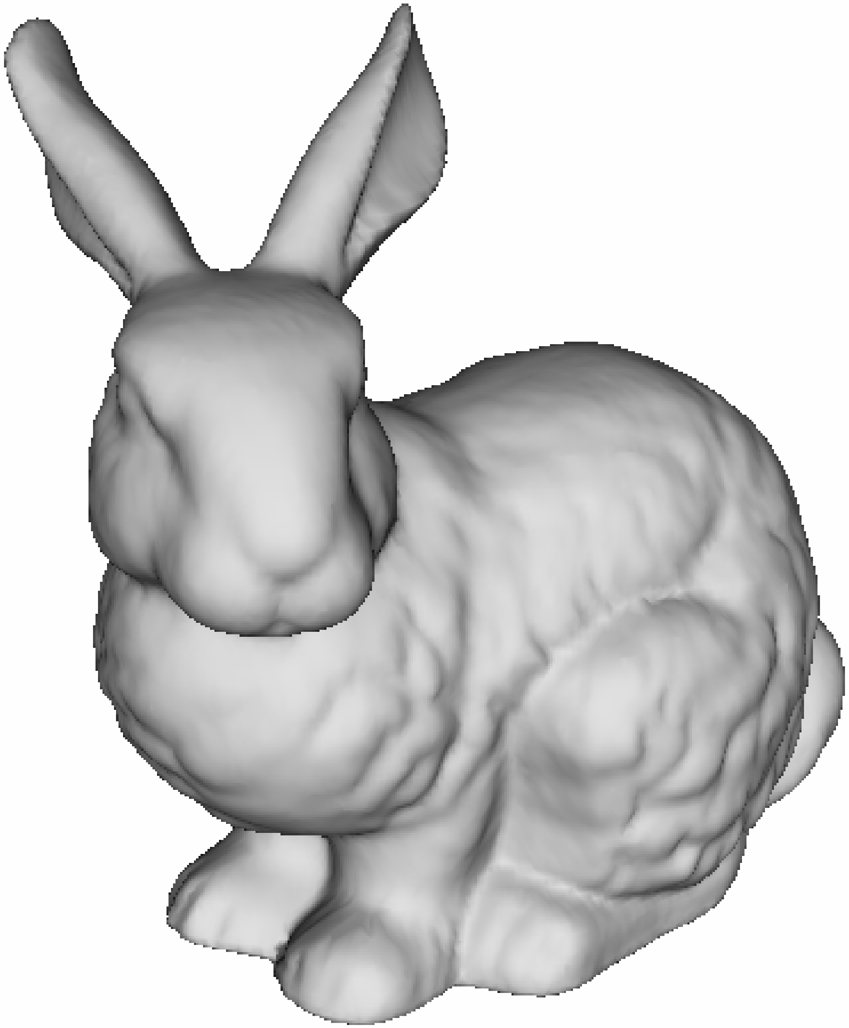




La proiezione prospettica

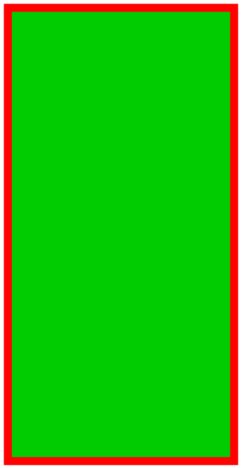
- Al variare della distanza focale (d)

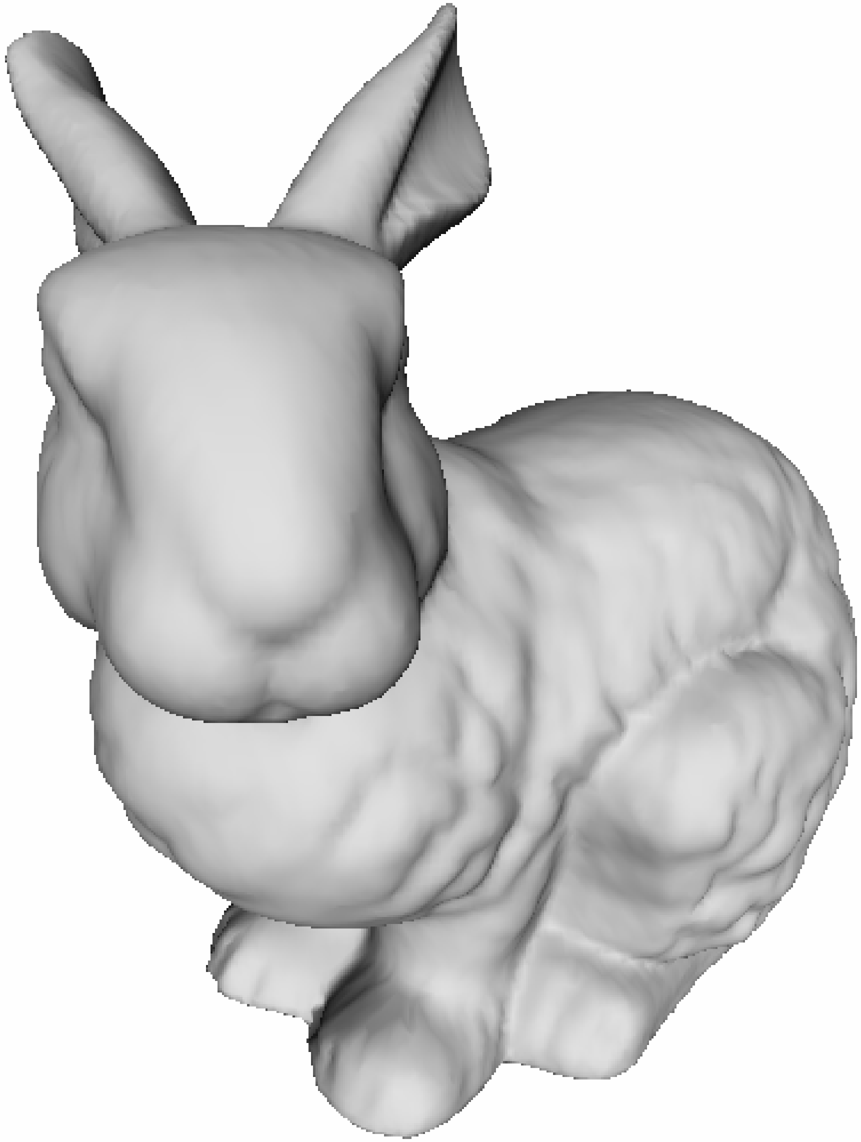




La proiezione prospettica

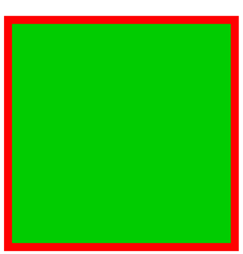
- Al variare della distanza focale (d)

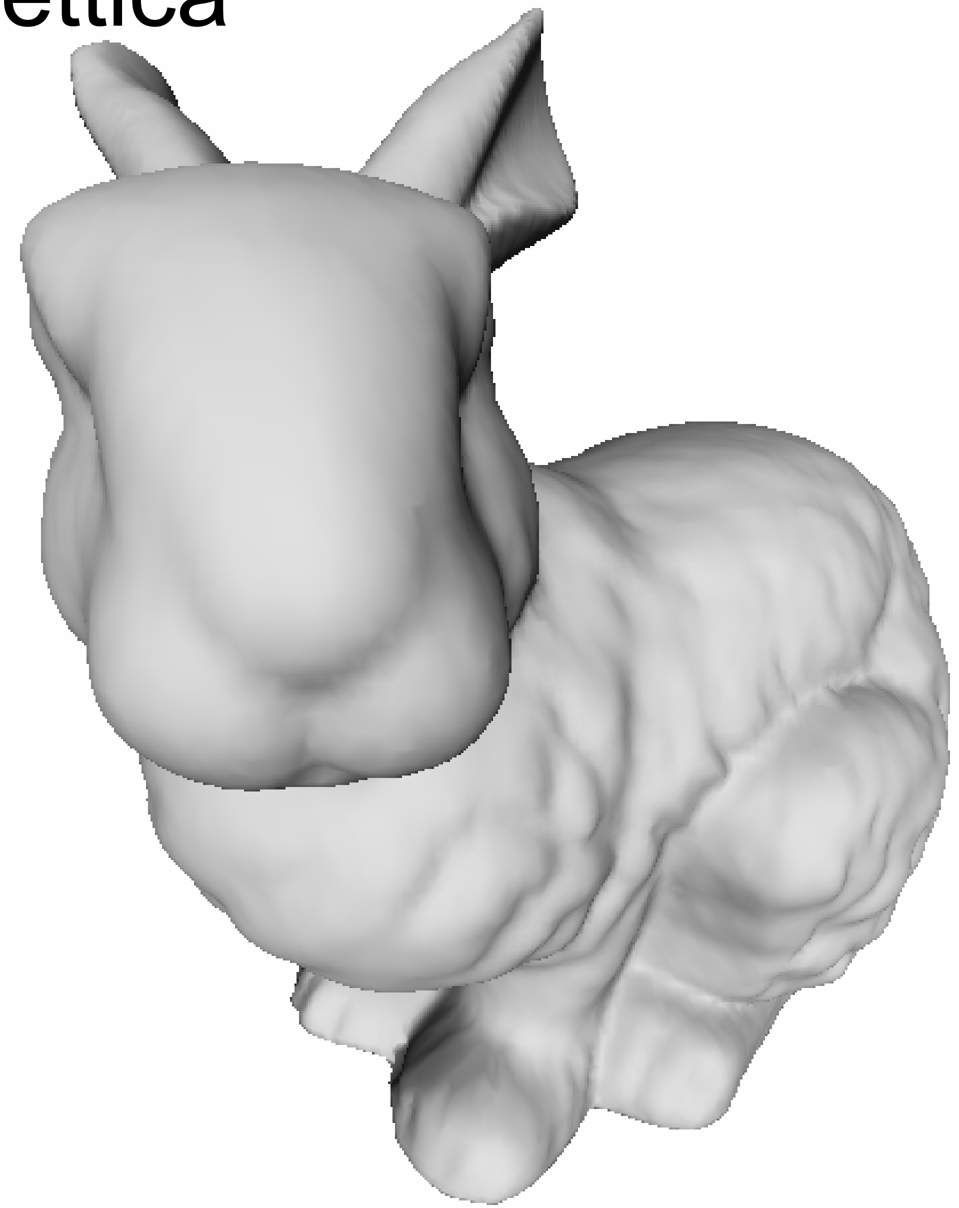




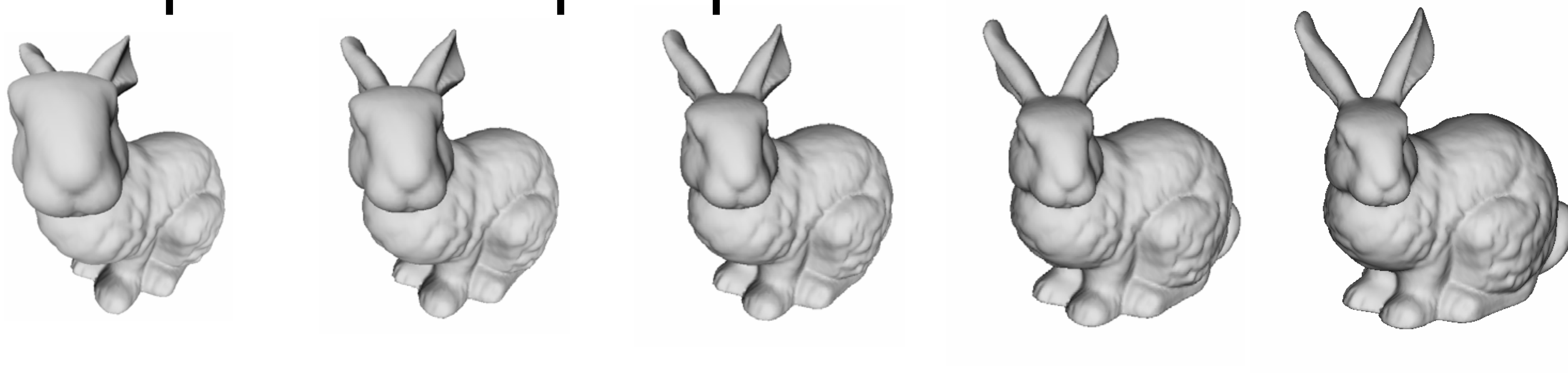
La proiezione prospettica

- Al variare della distanza focale (*d*)





La proiezione prospettica



d piccolo

d grande

d infinito (p. parallela)

Più distorsione prospettica.

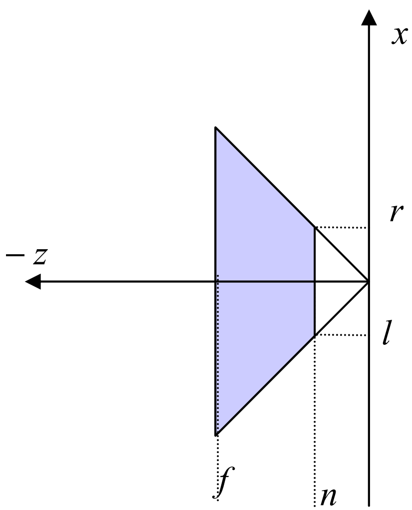
Effetto "fish-eye" (grandangolo)

Proporzioni più mantenute

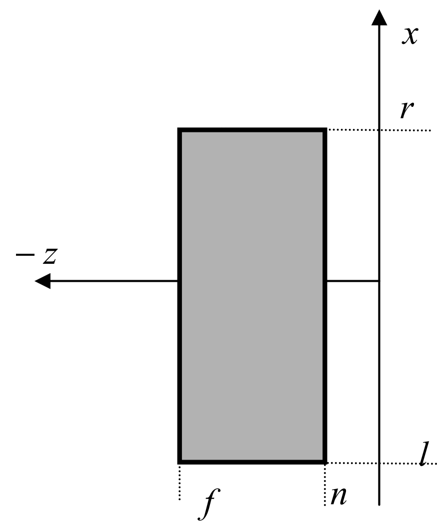
Effetto "zoom" (eg. vista dal satellite)

I volumi di vista (*viewing volumes*)

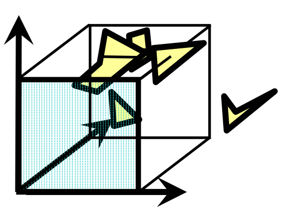
prospettico



ortografico



Qui i piani di clipping
sono piu' complicati

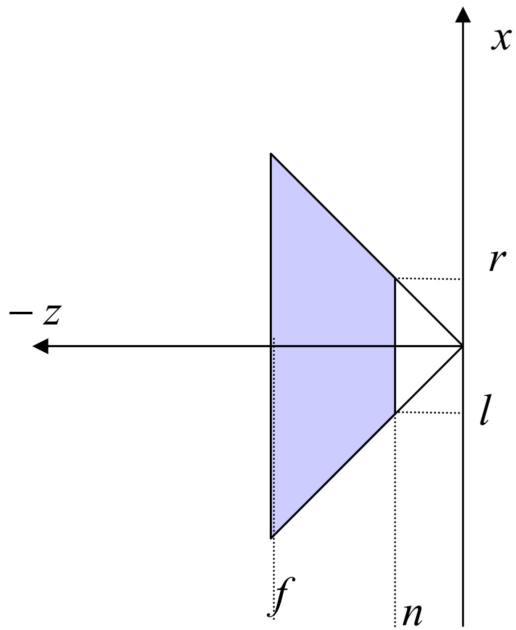


Ricordate?

Il volume di vista canonico

- Il volume di vista canonico e' definito dai piani $x=-1$, $x=1$, $y=-1$, $y=1$, $z=-1$, $z=1$, o piu' brevemente $[-1,1]\mathrm{x}[-1,1]\mathrm{x}[-1,1]$
- Sia che la proiezione sia prospettica che ortogonale, il volume di vista viene **trasformato** nel volume di vista canonico per effettuare il clipping piu' facilmente

da volume di vista prospettico a volume di vista *canonico*



$r = right$

$l = left$

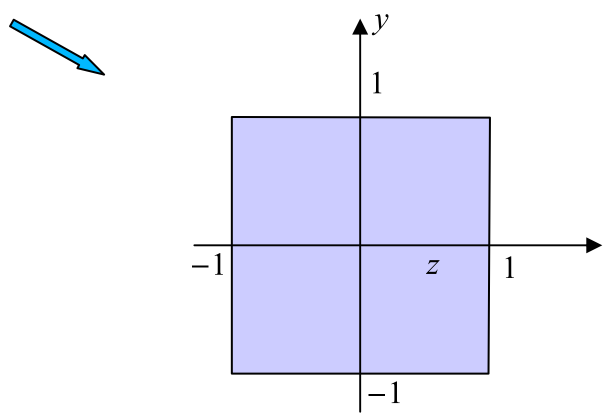
$b = bottom$

$t = top$

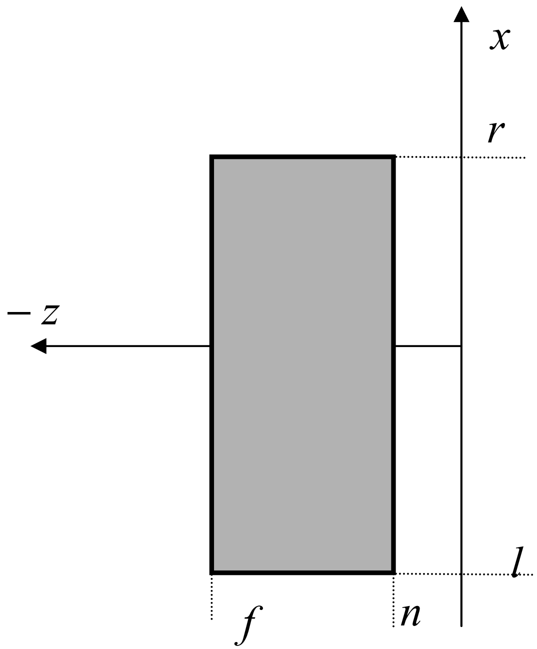
$n = near$

$f = far$

$$M_{per} = \begin{bmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & \frac{-(f+n)}{f-n} & \frac{-2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$



da volume di vista ortogonale a volume di vista canonico



$r = right$

$l = left$

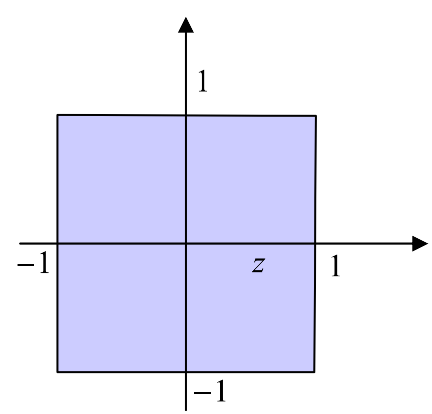
$b = bottom$

$t = top$

$n = near$

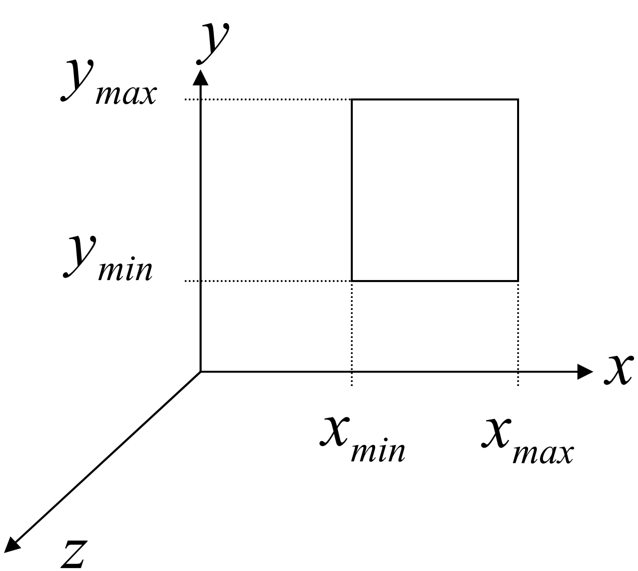
$f = far$

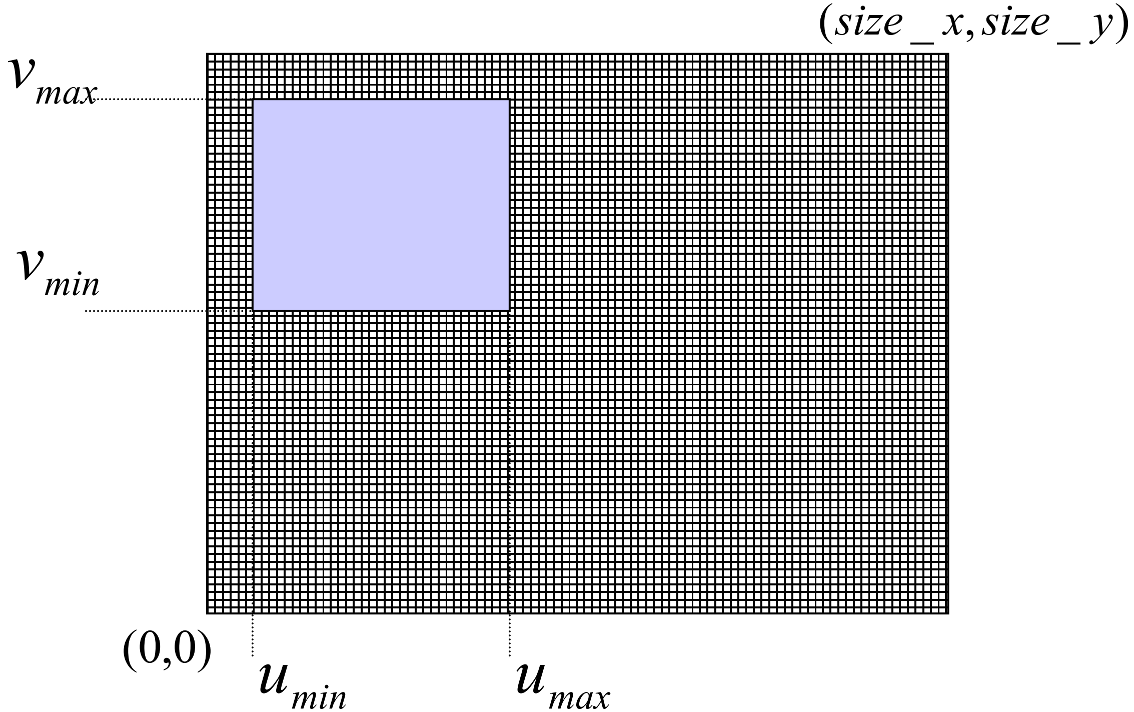
$$M_{ort} = \begin{bmatrix} \frac{2}{r-l} & 0 & 0 & -\frac{r+l}{r-l} \\ 0 & \frac{2}{t-b} & 0 & -\frac{t+b}{t-b} \\ 0 & 0 & \frac{2}{n-f} & -\frac{n+f}{n-f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Window to viewport

- Window: left, right,bottom,top. La nostra finestra di vista nel piano di proiezione
- Viewport: dove viene visualizzata sullo schermo





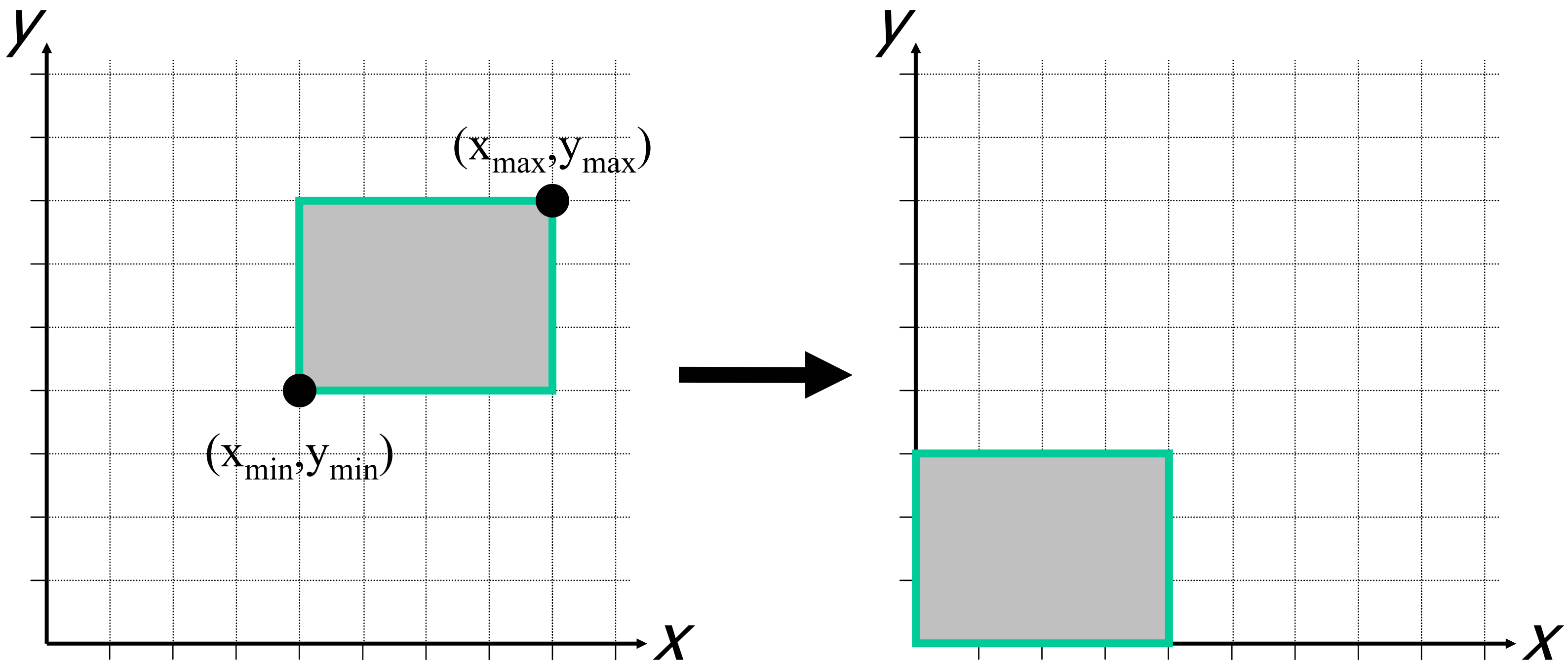
Trasformazione “window to viewport”

- La window è traslata nell’origine del sistema di coordinate;
- La dimensione della window è scalata sino ad essere uguale alla dimensione della viewport;
- La viewport è traslata nella posizione finale nel sistema di coordinate del dispositivo di output;

$$\mathbf{WV} = \begin{bmatrix} 1 & 0 & u_{\min} \\ 0 & 1 & v_{\min} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dfrac{u_{\max} - u_{\min}}{x_{\max} - x_{\min}} & 0 & 0 \\ 0 & \dfrac{v_{\max} - v_{\min}}{y_{\max} - y_{\min}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -x_{\min} \\ 0 & 1 & -y_{\min} \\ 0 & 0 & 1 \end{bmatrix}$$

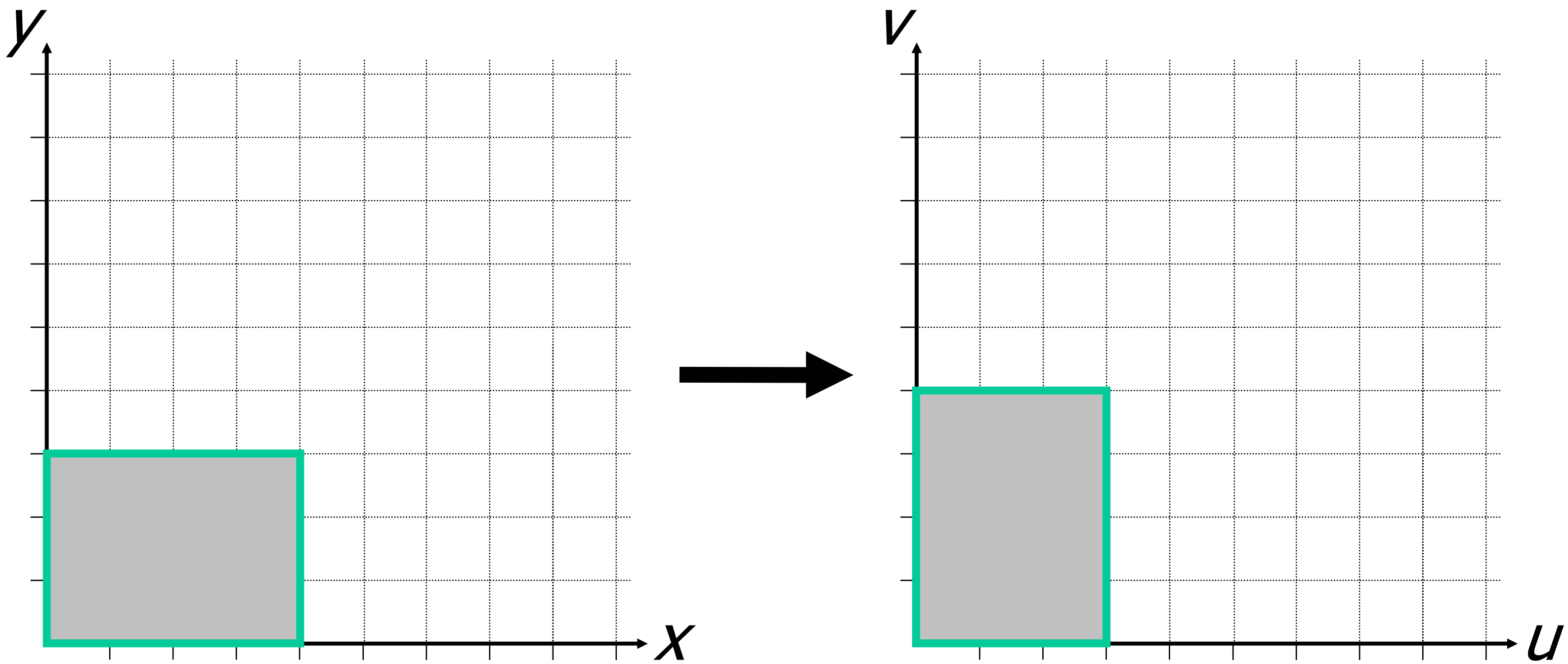
Trasformazione “window to viewport”

- 1 - La window è traslata nell’origine del sistema di coordinate;



Trasformazione “window to viewport”

- 2 - La dimensione della window è scalata sino ad essere uguale alla dimensione della viewport;



Trasformazione “window to viewport”

- 3 - La viewport è traslata nella posizione finale nel sistema di coordinate del dispositivo di output;