Corso
Grafica Computazionale

Trasformazioni Geometriche

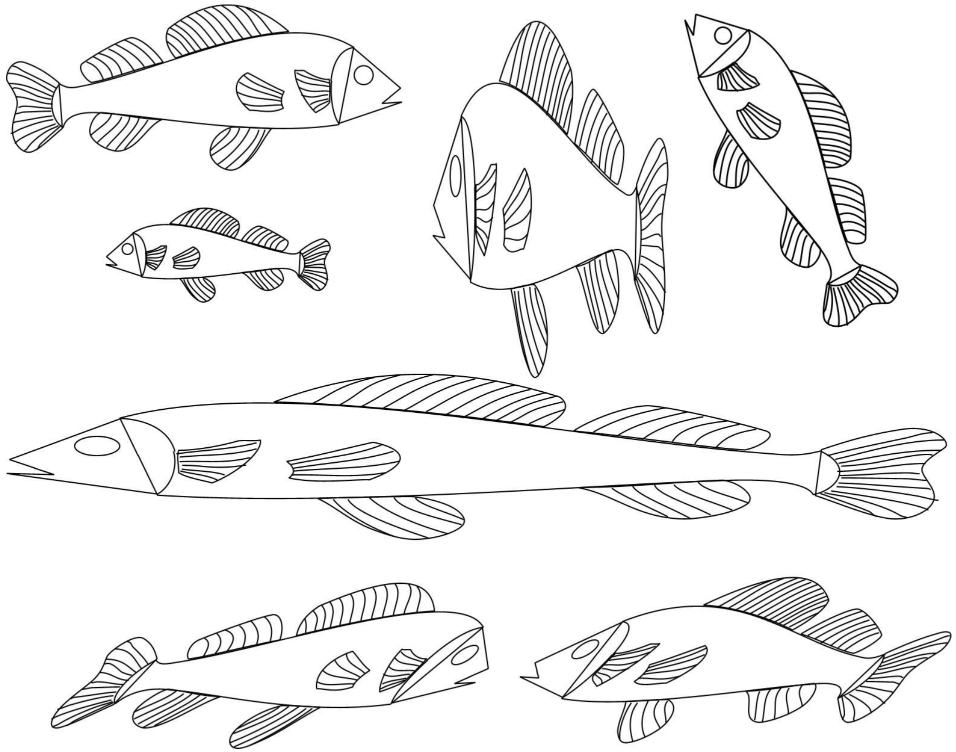
Docente:
Massimiliano Corsini

Laurea Specialistica in Ing. Informatica

Facoltà di Ingegneria
Università degli Studi di Siena

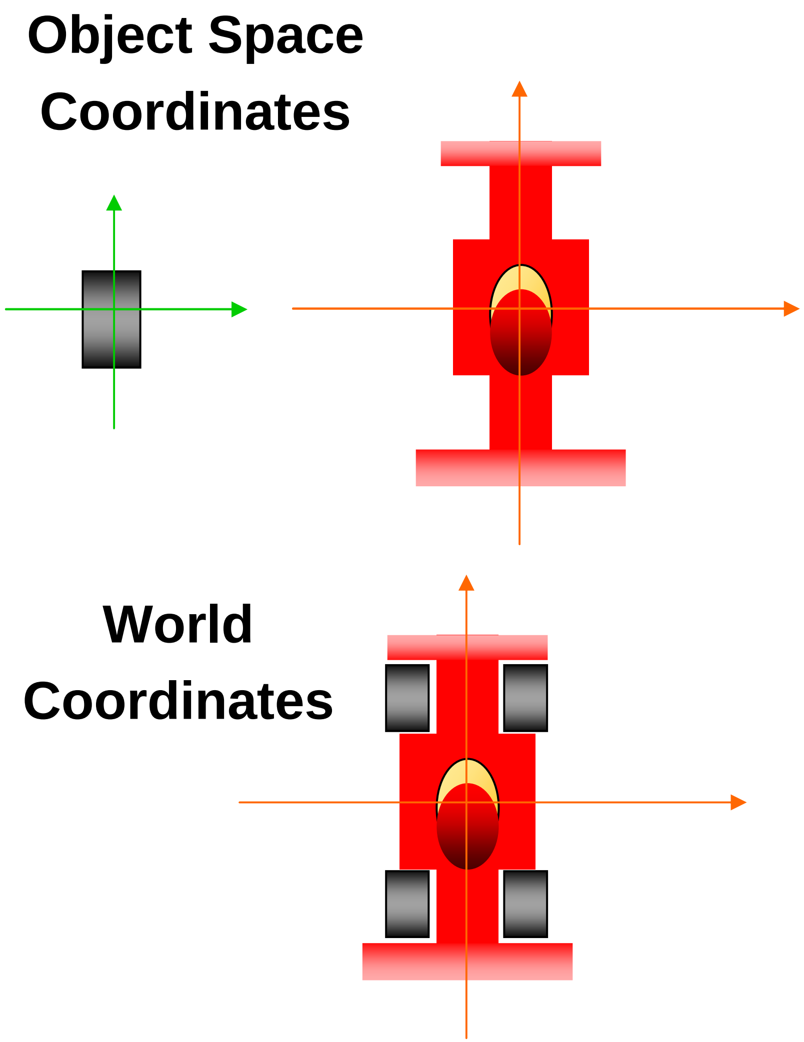
Trasformazioni Geometriche

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



Trasformazioni Geometriche

- Le trasformazioni geometriche ci permettono, ad esempio, di definire un oggetto tridimensionale componendolo con altri oggetti. Ogni oggetto, a partire dal proprio sistema di riferimento ***(object space),*** viene trasformato opportunamento in un sistema di riferimento comune ***(world space)*** per andare a far parte dell'oggetto finale.



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

- La trasformazione di una mesh poligonale si riduce alla trasformazione dei vertici che la compongono nel rispetto della connettività originale. Questo grazie al fatto che trattiamo trasformazioni affini …

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).

- 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
  - Deformazioni di tipo *shear*

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:

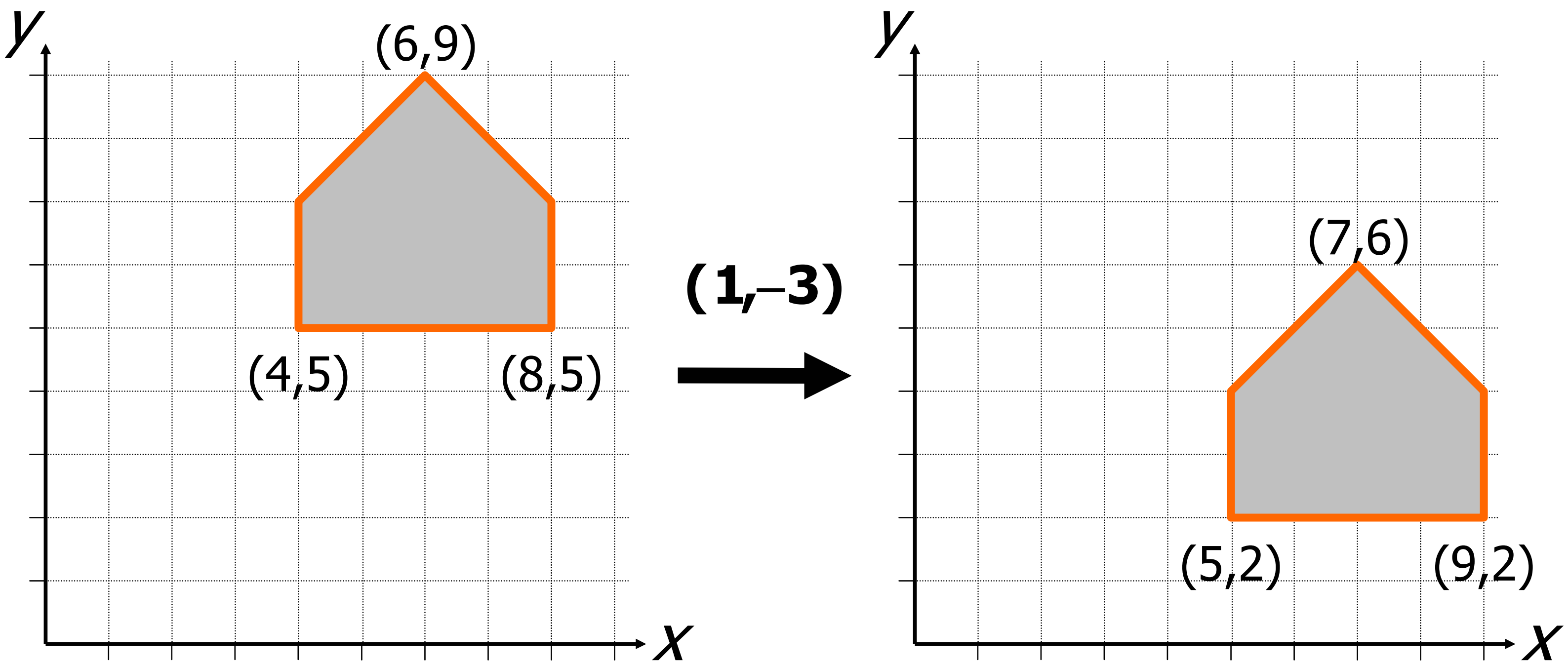
$$P = \begin{bmatrix} x \\ y \end{bmatrix}; \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix}; \quad \mathbf{T} = \begin{bmatrix} d_x \\ d_y \end{bmatrix};$$

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

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

Traslazione (esempio)

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



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:

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

Scalatura

- In notazione matriciale:

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

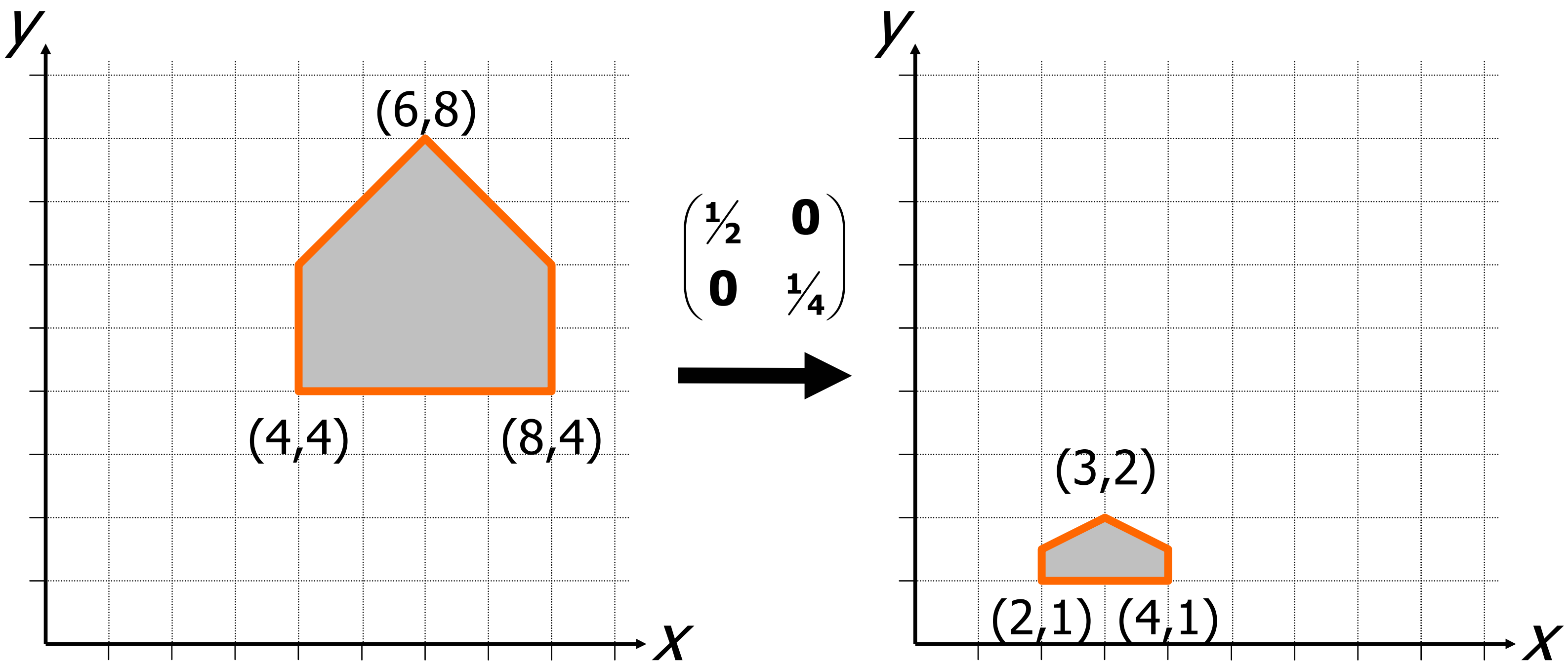
- dove

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

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

Scalatura (esempio)

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



- 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;*

Rotazione

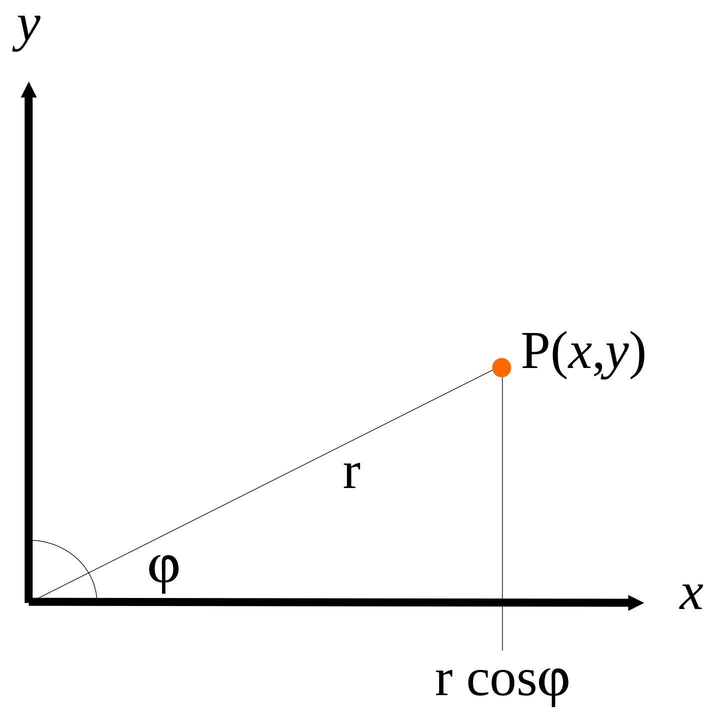
- Fissato un punto C (*pivot*) di riferimento ed un verso di rotazione (*orario* o *antiorario*), ruotare una primitiva geometrica attorno a C significa muovere tutti i suoi punti nel verso assegnato in maniera che si conservi, per ognuno di essi, la distanza da C;
- Una rotazione di θ attorno all'origine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

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

Come si ricava la trasformazione di rotazione

- La relazione tra P' e P si ricava trigonometricamente;

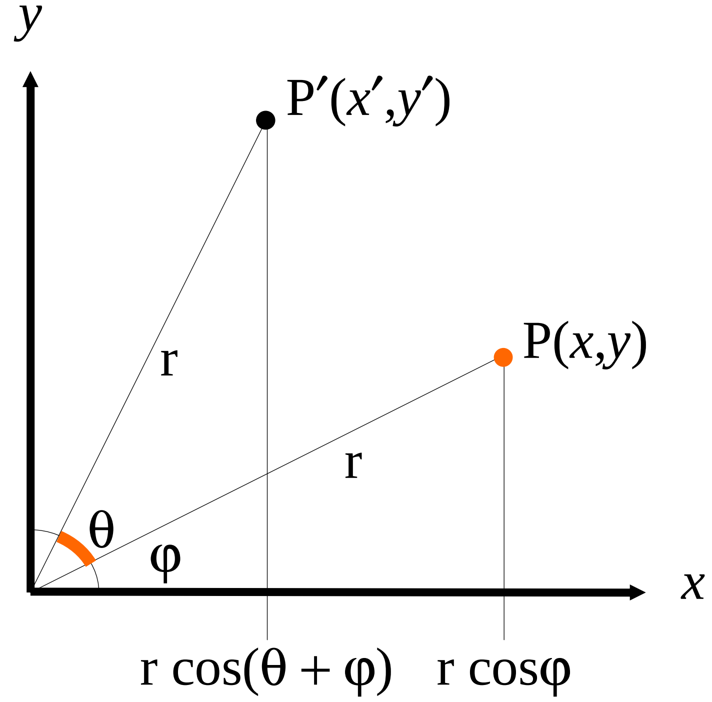
- Le coordinate di P possono essere espresse in coordinate polari:



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

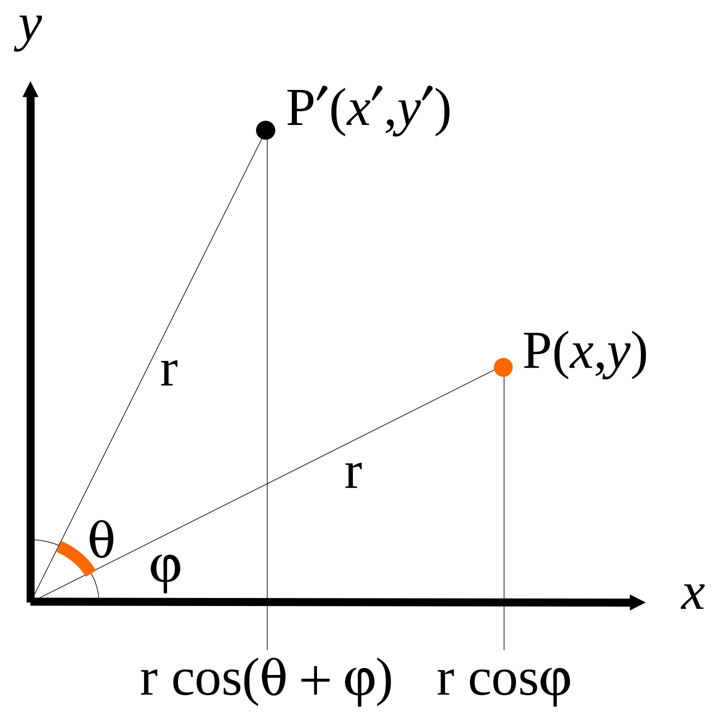
Come si ricava la trasformazione di rotazione

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



Come si ricava la trasformazione di rotazione

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



Rotazione (notazione matriciale)

- In notazione matriciale abbiamo:

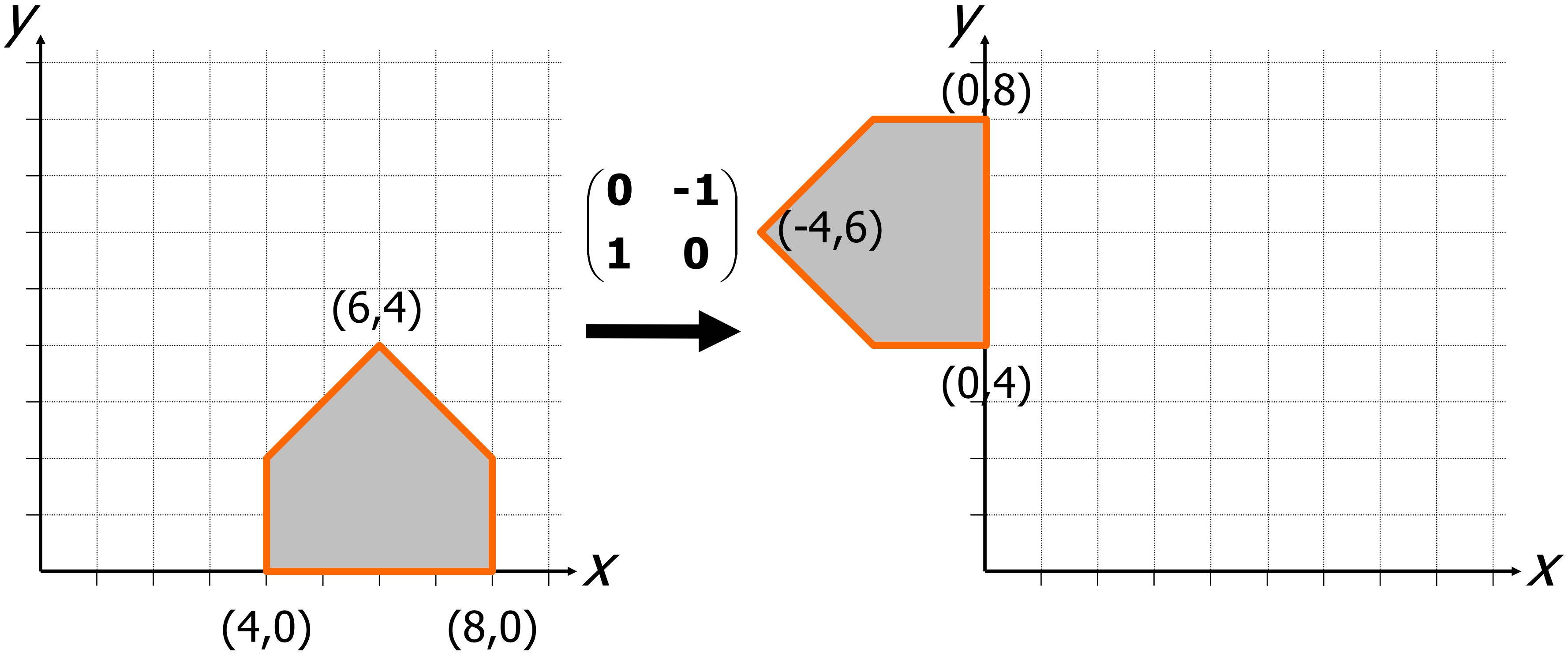
$$P' = \mathbf{R} \cdot P$$

- dove:

$$P = \begin{bmatrix} x \\ y \end{bmatrix}; \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix}; \quad \mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

Rotazioni (esempio)

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



$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

- 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

- 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 punti all'infinito e quindi *direzioni.*

Trasformazioni di base in 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 di base in 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*:

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

Trasformazione di 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:

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

Trasformazione di Shear

- Shear rispetto all'asse *x*:

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

- Shear rispetto all'asse *y*:

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

- Shear 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 Shear

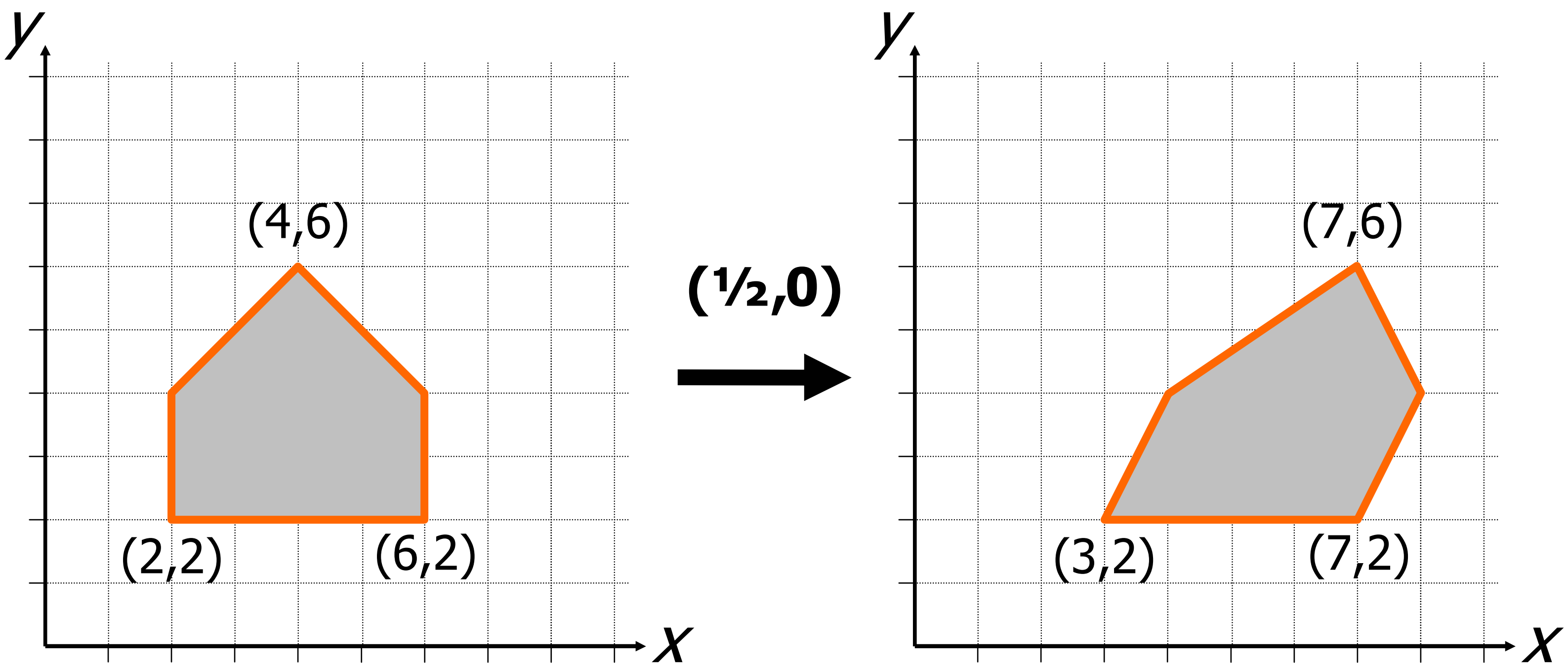
- Usando le matrici prima mostrate si ottengono le relazioni:

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

da cui risulta evidente come la deformazione lungo l'asse x sia linearmente dipendente dalla coordinata y e viceversa.

Shear (esempio)

- Esempio di deformazione con: a=½ e b=0



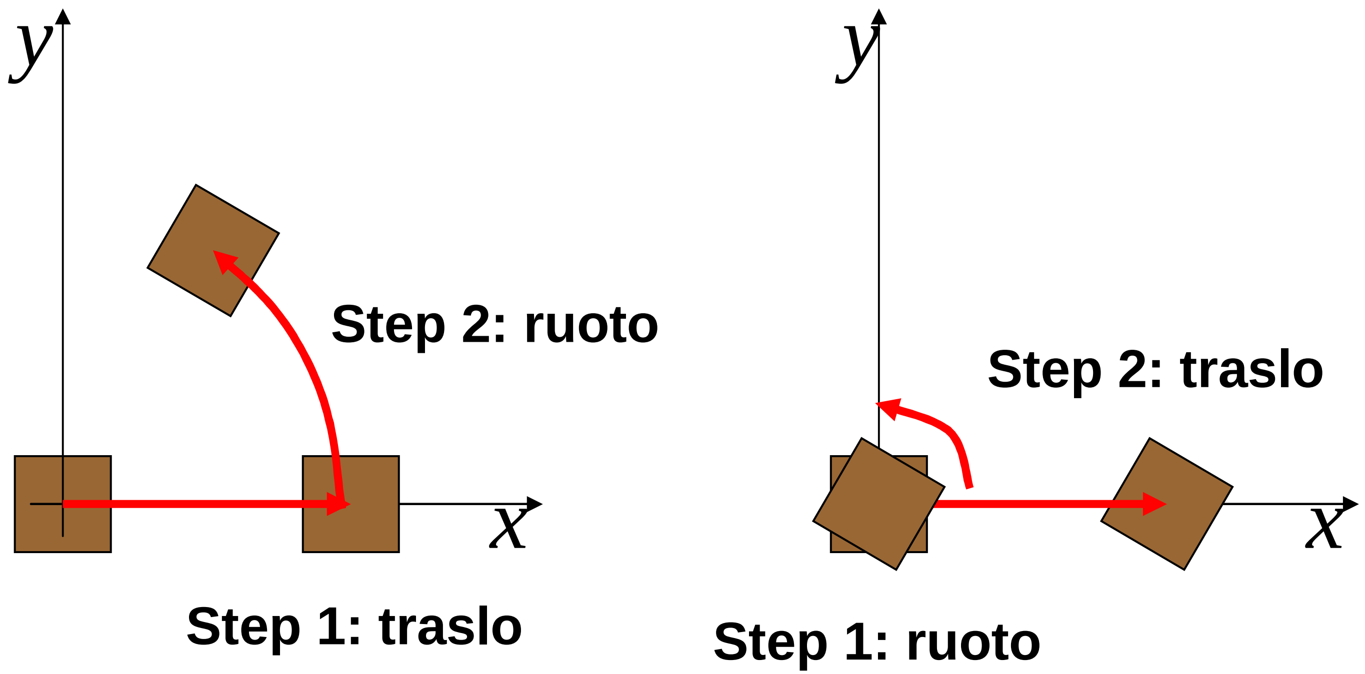
Composizione di Trasformazioni

- La rappresentazione in coordinate omogenee permettono di gestire facilmente la concatenazione di trasformazioni;
- L'ordina di concatenazione è importante perché le trasformazioni geometriche sono associative ma non sono (di solito) 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:

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

Non commutatività(!)

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



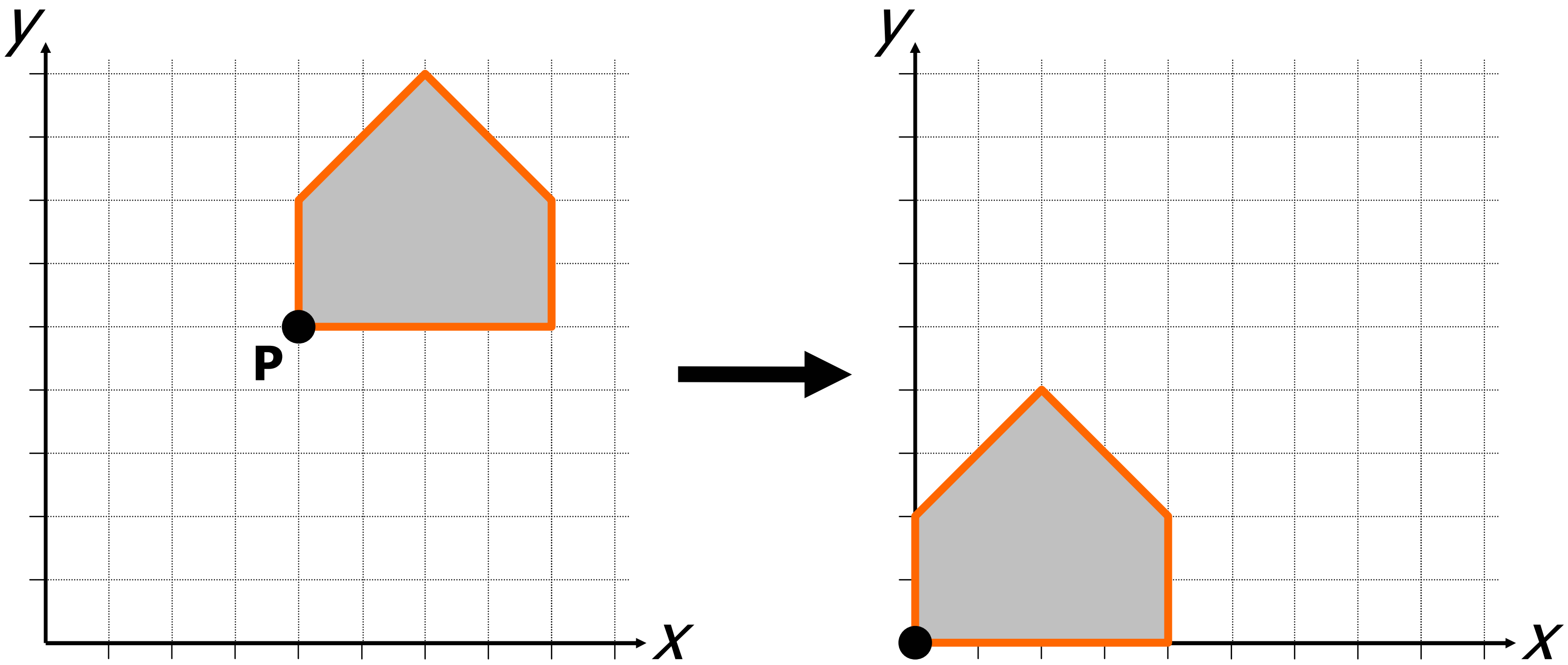
Rotazione attorno ad un punto dato P

- La rotazione antioraria di un angolo θ attorno ad un punto P generico si ottiene componendo le seguenti trasformazioni:
  1. Traslazione che muove P nell'origine degli assi;
  2. Rotazione attorno all'origine;
  3. Traslazione opposta alla precedente che riporta P nella sua posizione originale.

$$\mathbf{R}_g = \begin{bmatrix} 1 & 0 & P_x \\ 0 & 1 & P_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -sen\theta & 0 \\ sen\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -P_x \\ 0 & 1 & -P_y \\ 0 & 0 & 1 \end{bmatrix}$$

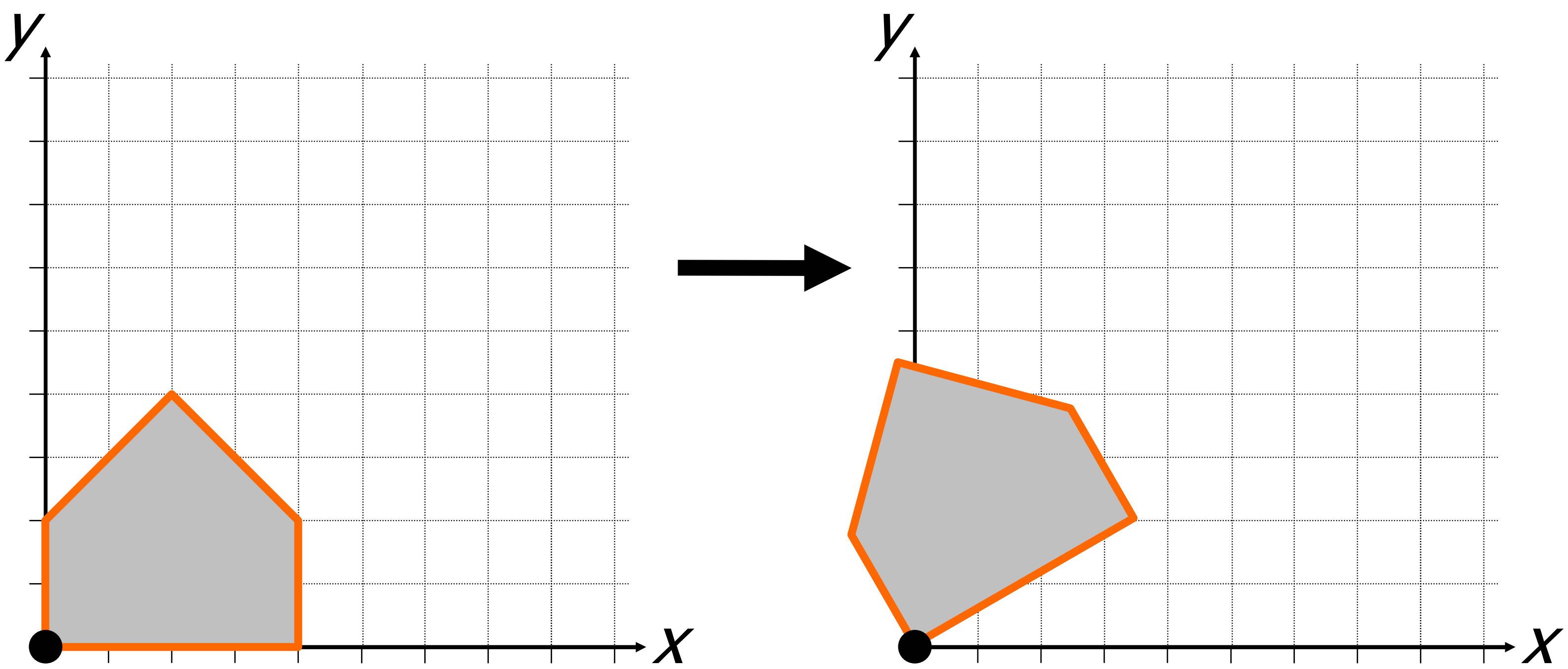
Rotazione attorno ad un punto dato P (esempio)

- *Passo 1:* Traslazione di *P* nell'origine degli assi



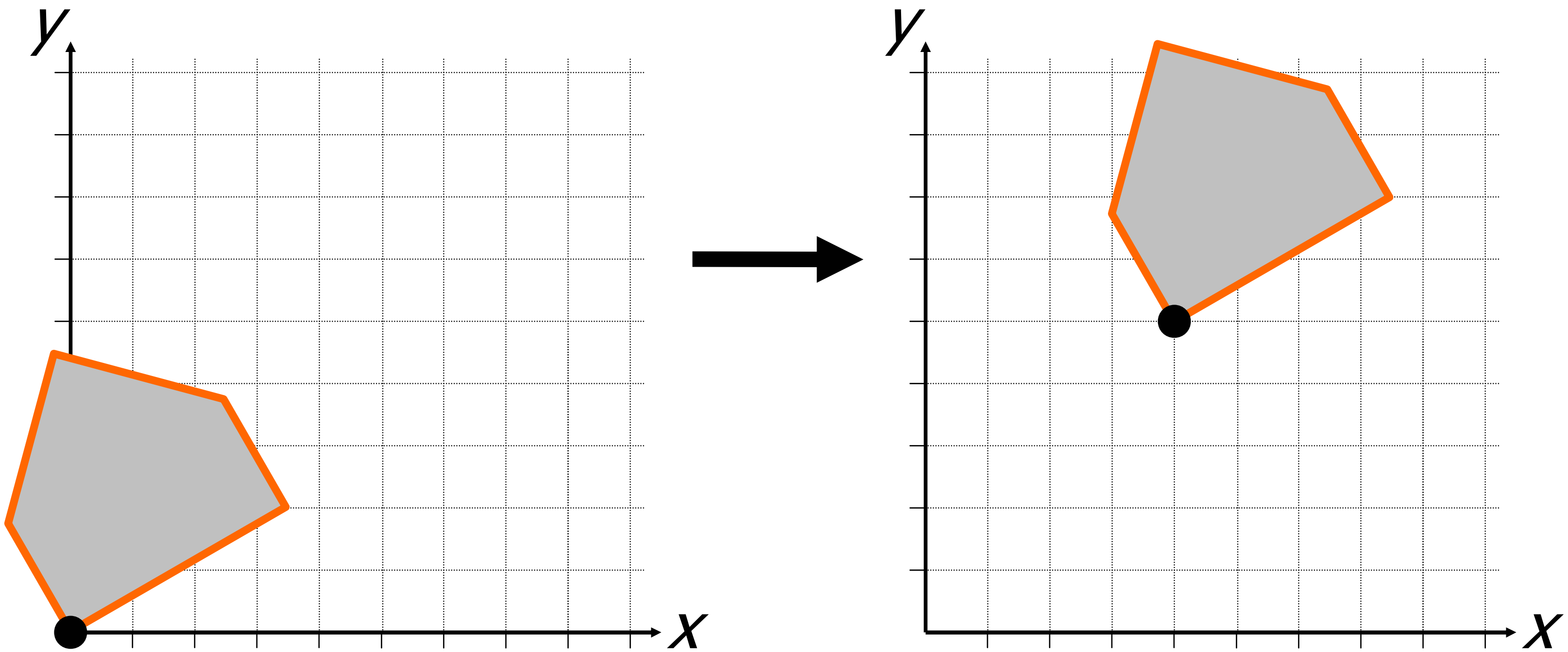
Rotazione attorno ad un punto dato P (esempio)

- *Passo 2:* Rotazione attorno all'origine ($\theta = \pi/6$)



Rotazione attorno ad un punto dato P (esempio)

- *Passo 3:* Traslazione opposta alla precedente



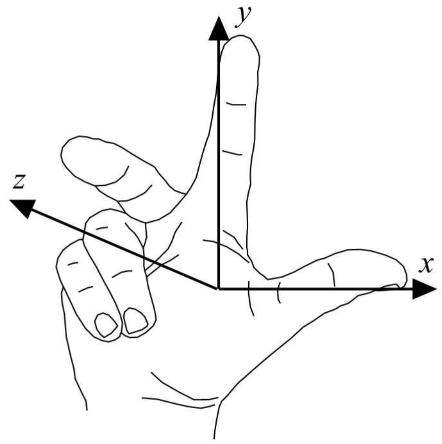
Scalatura rispetto ad un punto dato P

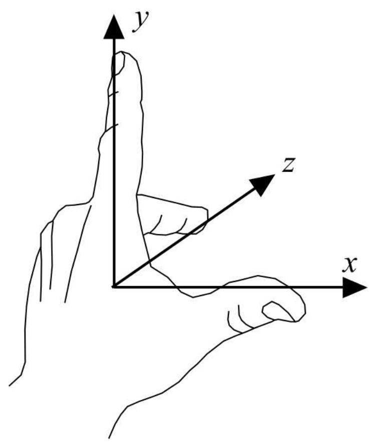
- Trasformazione di scalatura attorno ad un punto P generico:
  1. Traslazione che muove P nell'origine degli assi;
  2. Trasformazione di scala attorno all'origine;
  3. Traslazione opposta alla precedente che riporta P nella sua posizione originale.

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

Sistema di riferimento nello spazio

- Il passaggio dal piano allo spazio introduce una ambiguità per quanto concerne la scelta del sistema di riferimento cartesiano;
- Sistema destrorso (in alto, *right-handed system*) oppure sinistrorso (in basso, *left-handed system*);





Trasformazioni nello spazio

- Le trasformazioni geometriche nel piano possono essere rappresentate, in coordinate omogenee, mediante matrici 3×3;
- In modo analogo, le trasformazioni geometriche nello spazio possono essere rappresentate da matrici 4×4;
- Nello spazio, un punto in coordinate omogenee è rappresentato dalla quadrupla (x, y, z, w).

Traslazione (3D)

- Trasformazione di traslazione

$$\mathbf{T}(\mathrm{d}_x, \mathrm{d}_y, \mathrm{d}_z) = \begin{bmatrix} 1 & 0 & 0 & \mathrm{d}_x \\ 0 & 1 & 0 & \mathrm{d}_y \\ 0 & 0 & 1 & \mathrm{d}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Scalatura (3D)

- Trasformazione di scalatura

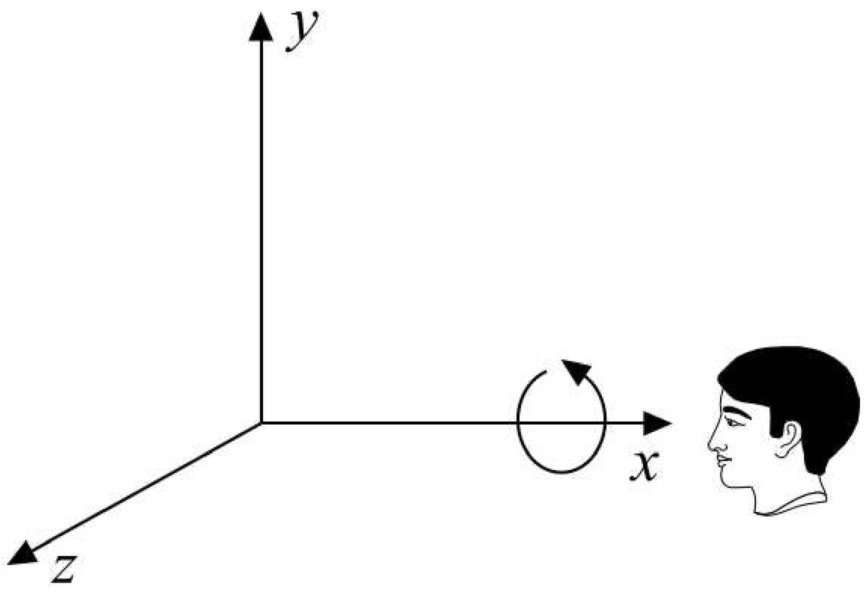
$$\mathbf{S}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotazione (3D)

- Trasformazione di rotazione
  - La trasformazione di rotazione generica nello spazio (cioè attorno ad un asse qualsiasi) è invece complessa e non direttamente estendibile dal caso 2D (in cui l'asse di rotazione è perpendicolare al piano xy);
  - Una generica rotazione nello spazio può essere ottenuta come composizione di 3 rotazioni attorno agli assi cartesiani.

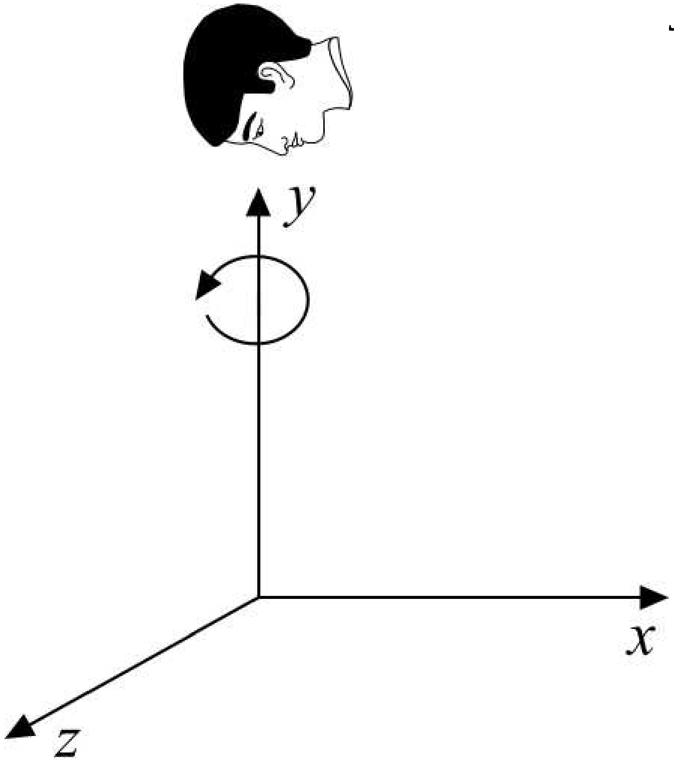
Rotazione rispetto asse X

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -sen\theta & 0 \\ 0 & sen\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



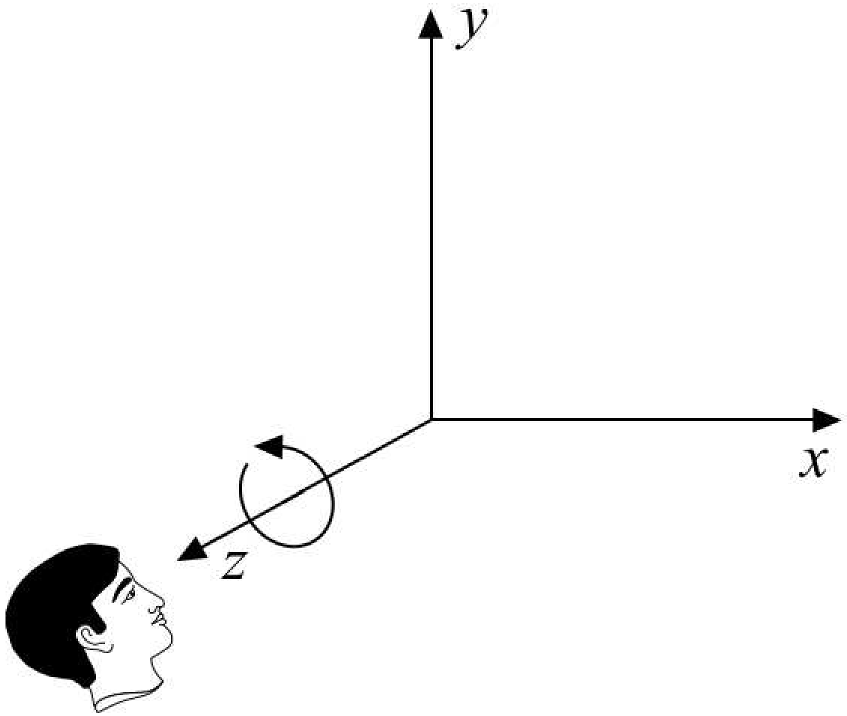
Rotazione rispetto asse Y

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \operatorname{sen}\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\operatorname{sen}\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Rotazione rispetto asse Z

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\operatorname{sen}\theta & 0 & 0 \\ \operatorname{sen}\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



View Transformation

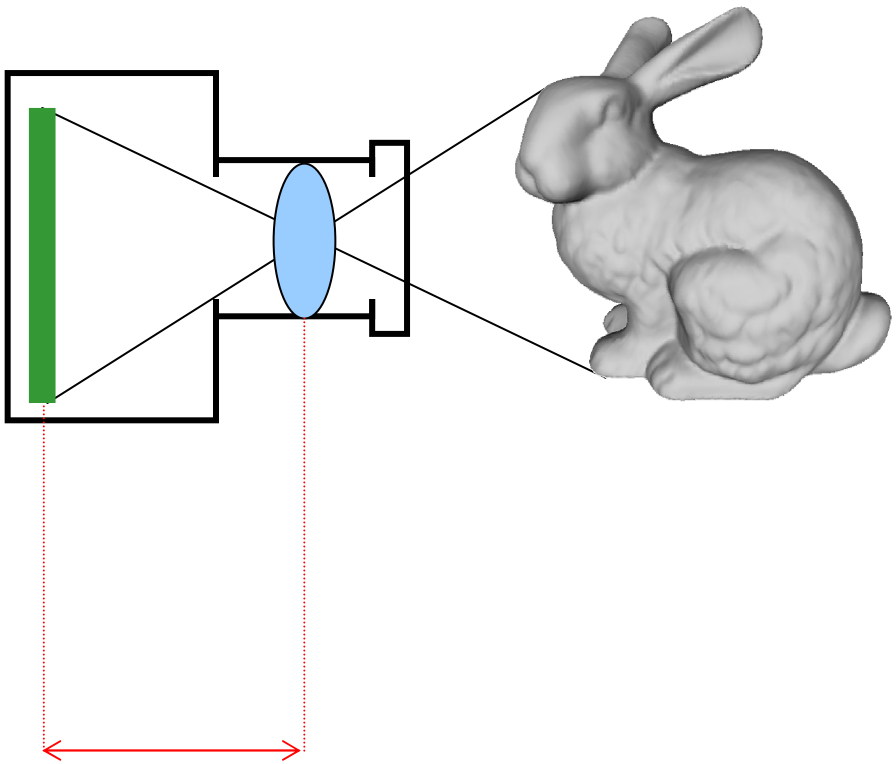
- Trasformazioni di vista
  - Il processo di visione in tre dimensioni;
  - Le trasformazioni di proiezione;
  - I parametri della vista 3D;
  - I sistemi di coordinate

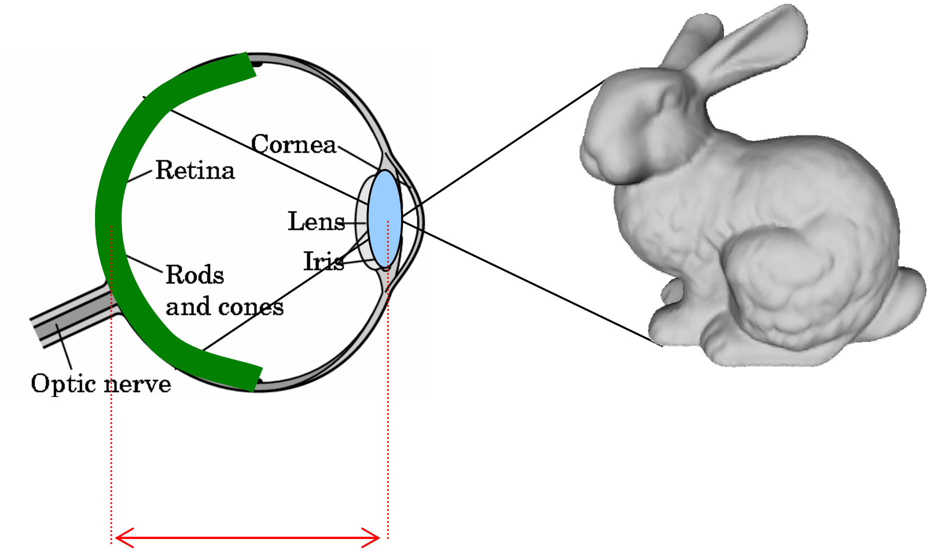
Dal 3D al 2D

- Il processo di *rendering* (visualizzazione) nello spazio *2D* si riduce alla definizione di una *window* nello spazio dell'applicazione grafica, una *viewport* nello spazio delle coordinate del dispositivo di output ed alla applicazione di una trasformazione "window-to-viewport" dopo aver effettuato il *clipping* (rimozione) delle primitive (o parte di esse) esterne alla *window.*
- Se il dispositivo di output è 2D, il processo di *rendering 3D* è assimilabile al processo di formazione di un'immagine da parte di un sistema ottico, quale ad esempio una macchina fotografica. La visualizzazione consiste nel creare una particolare *vista* della scena 3D (relazione scena/osservatore).

Formazione dell'immagine

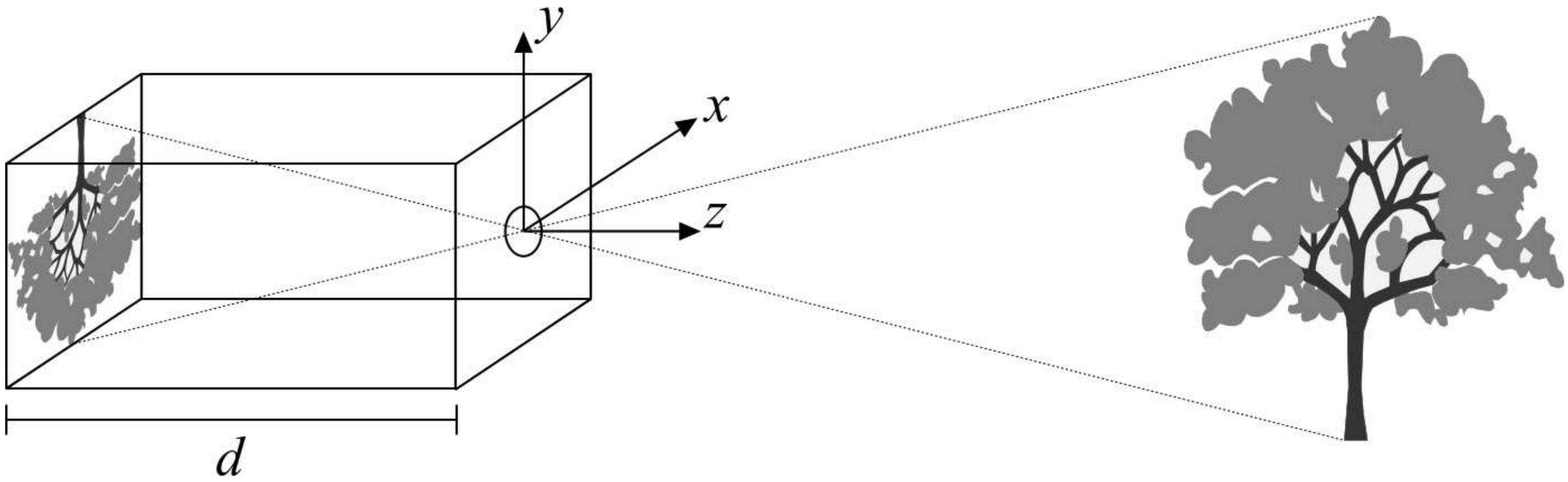
- Il processo di formazione dell'immagine in un sistema ottico.





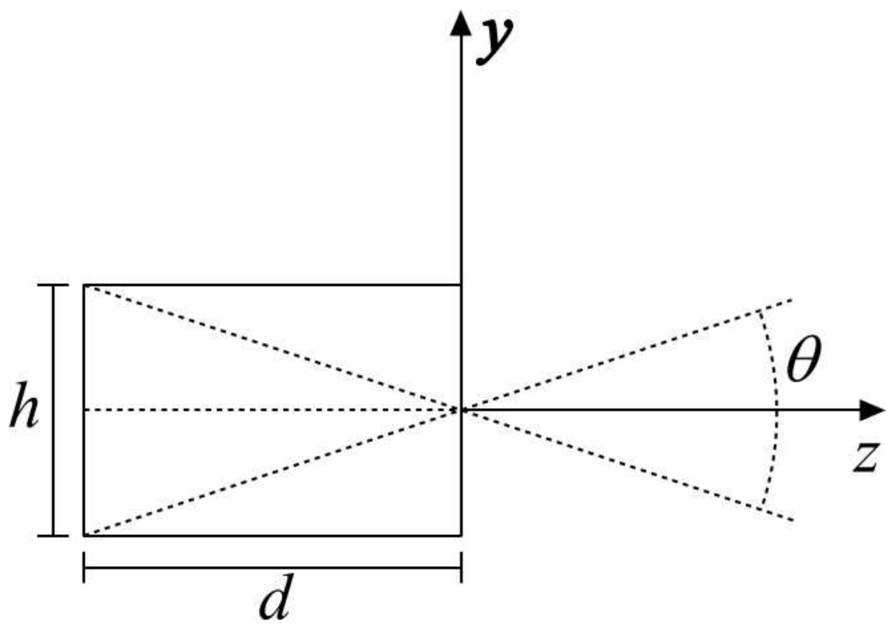
Macchina fotografica virtuale

- La metafora utilizzata per descrivere le relazioni scena/osservatore è quella della macchina fotografica virtuale (*synthetic camera*).



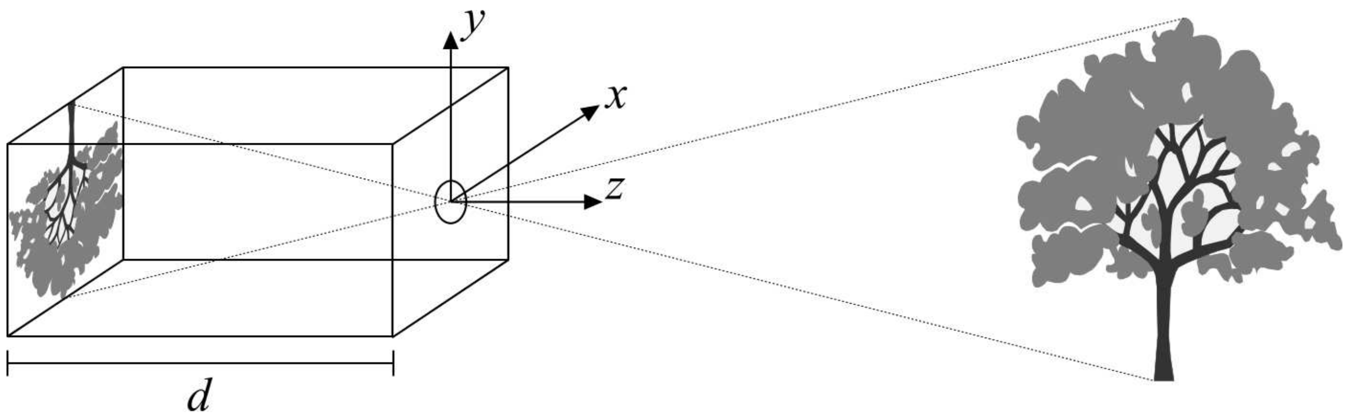
Pinhole Camera

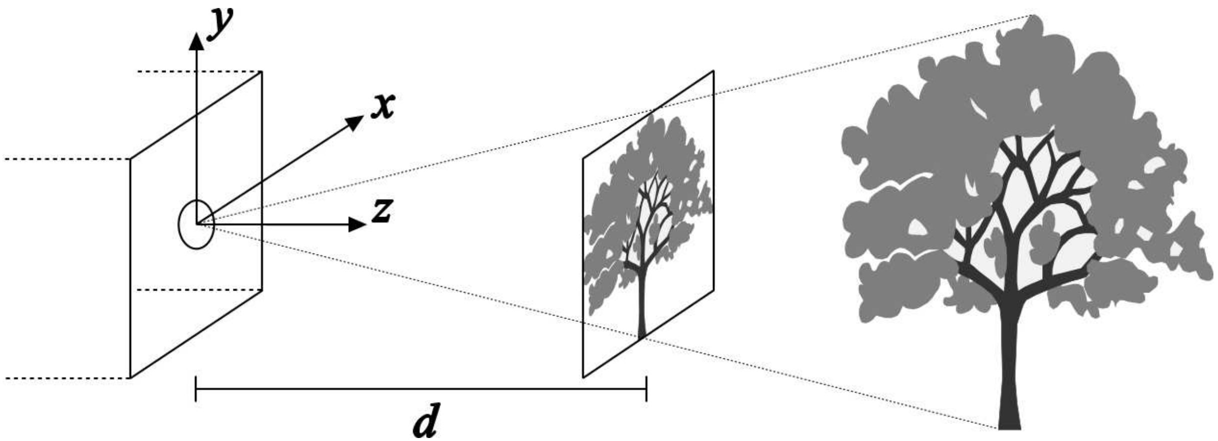
- La macchina fotografica virtuale è modellata considerando un parallelepipedo in cui la faccia anteriore presenta un foro di dimensioni infinitesime (*pinhole camera*) e sulla faccia posteriore si formano le immagini;
- Immagini nitide, nessun problema di luminosità, l’angolo θ di vista può essere modificato variando il rapporto tra la distanza focale (d) e la dimensione del piano immagine.



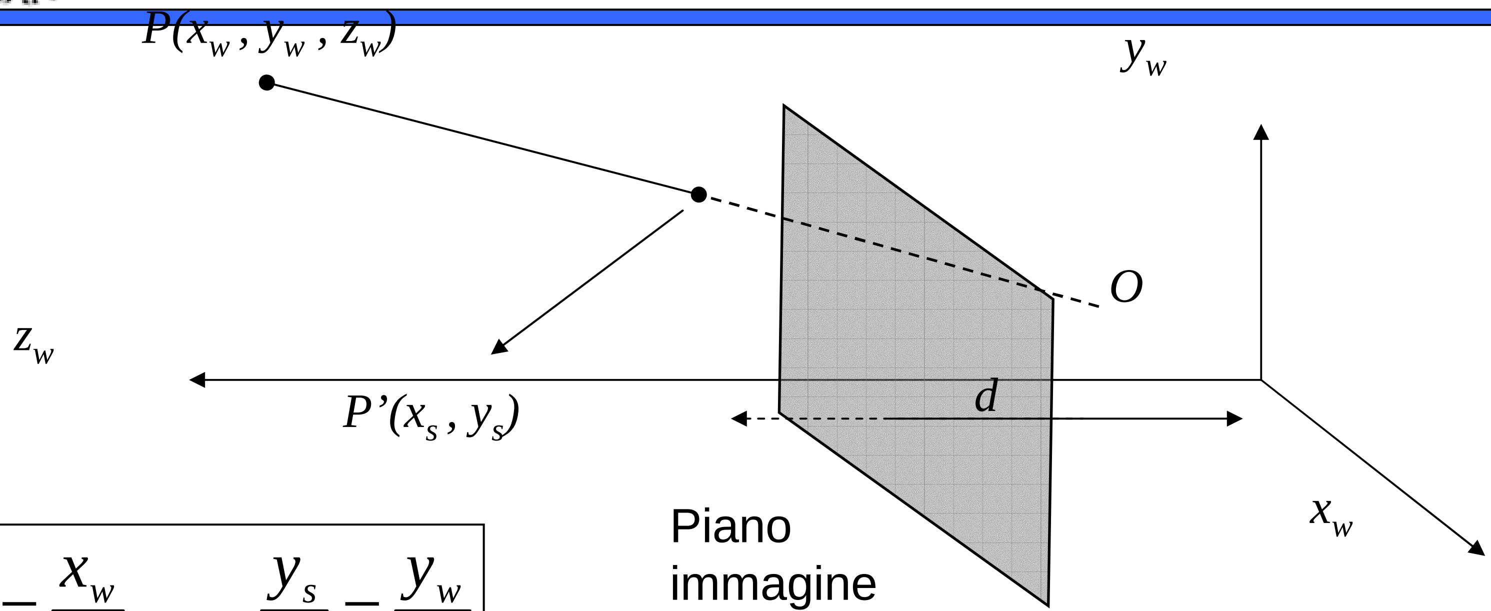
Pinhole Camera

- Per evitare l’effetto di ribaltamento si assume l’esistenza di un piano immagine tra la scena ed il centro di proiezione





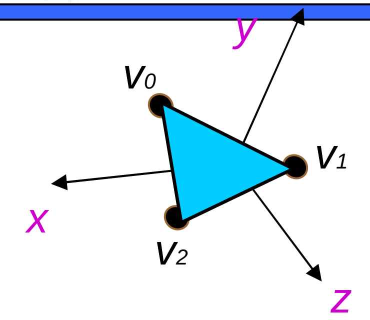
La matematica della Pinhole Camera



$$\boxed{\frac{x_s}{d} = \frac{x_w}{z_w} \quad , \quad \frac{y_s}{d} = \frac{y_w}{z_w}}$$

$$\begin{bmatrix} x_s \\ y_s \\ d \end{bmatrix} = \begin{bmatrix} (x_w d)/z_w \\ (y_w d)/z_w \\ d \end{bmatrix} \xrightarrow[\text{omogenee}]{\text{Coordinate}} \begin{bmatrix} x_s \\ y_s \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = T_{\text{persp}} P$$

Sistemi di coordinate



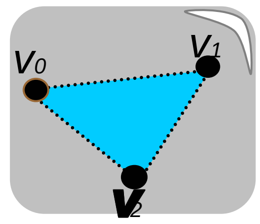
Object-space Coordinates

0) transformazione di modellazione

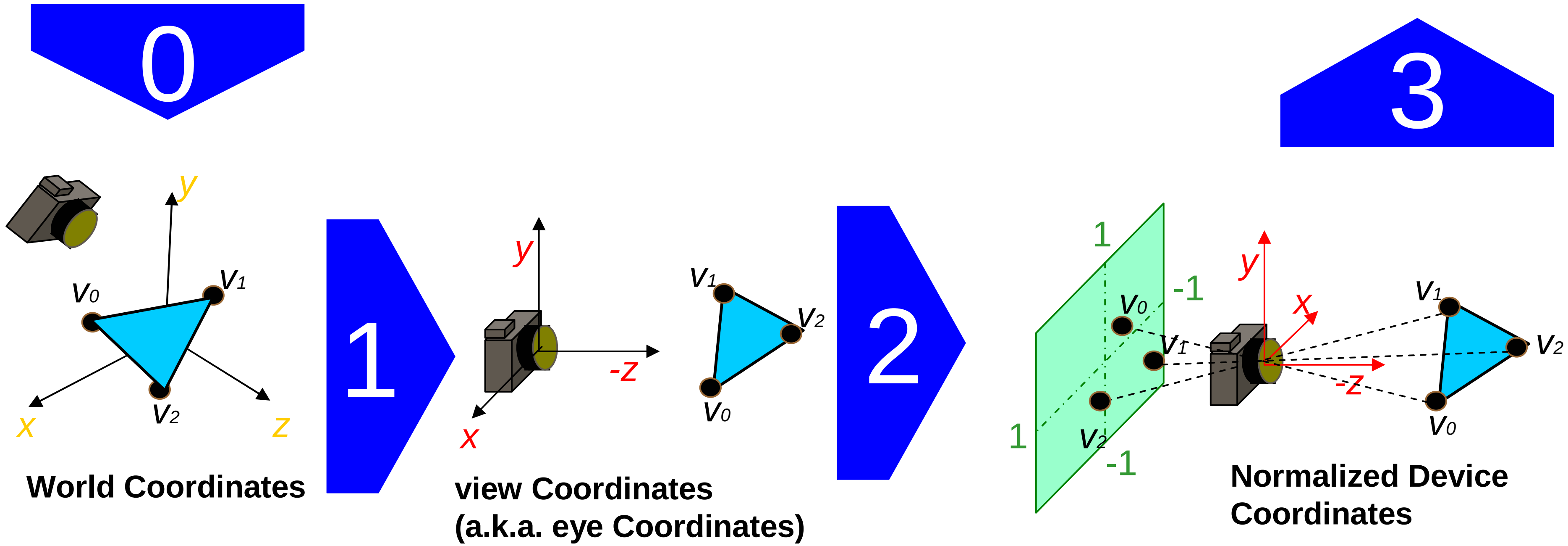
1) transformazione di vista

2) transformazione di proiezione

3) transformazione di viewport



Screen Space



World → view coordinates

- È sufficiente ruotare e traslare tutti i vertici del modello (in world coordinates) prima di fare la proiezione (con la pinhole):

$$P_{eye} = M_{wv} P_{world}$$

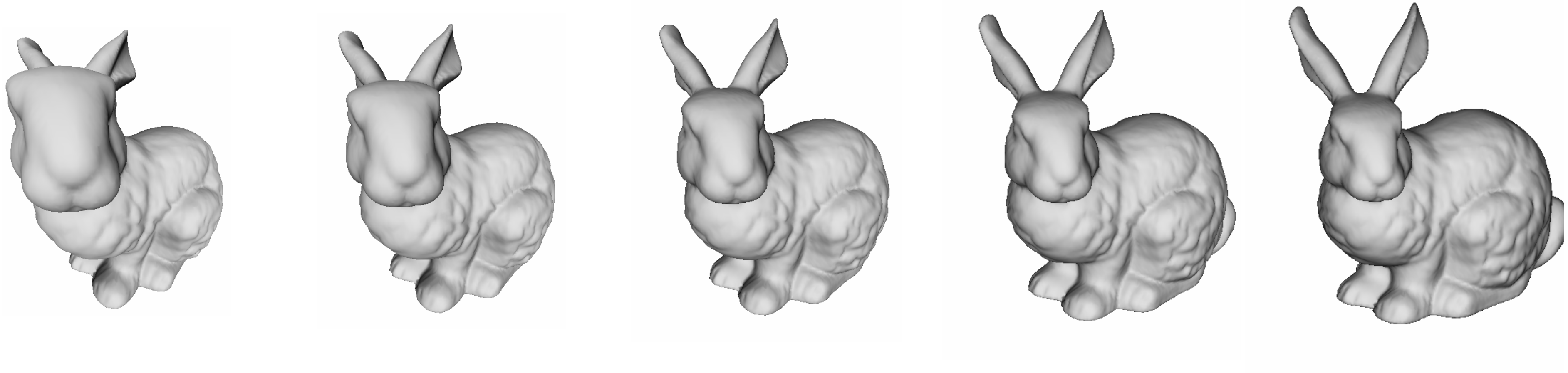
M_{wv} **È una matrice 4x4 che modella la roto-traslazione opportuna**

Non solo pinhole...

- Non soltanto la proiezione prospettica è utilizzata ma anche altri tipi di proiezione, ad esempio la ***proiezione ortogonale***
- **Per ottenerla basta rendere la z=0**

$$P_{ort} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad P_{prsp} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

Distanza focale



d **piccolo** → ***d*** **grande**

d **infinito**
**(diventa
una proiezione
ortogonale)**

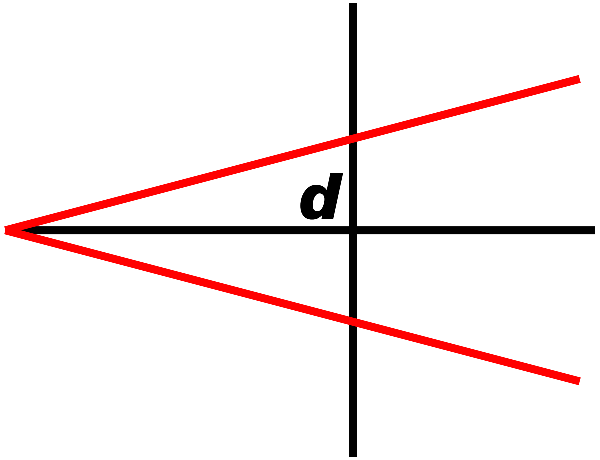
Più distorsione
prospettica.

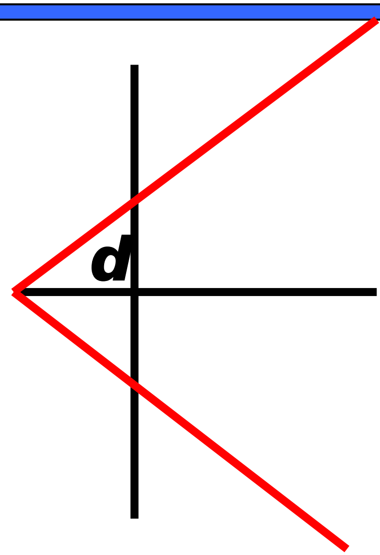
Effetto "fish-eye"
(grandangolo)

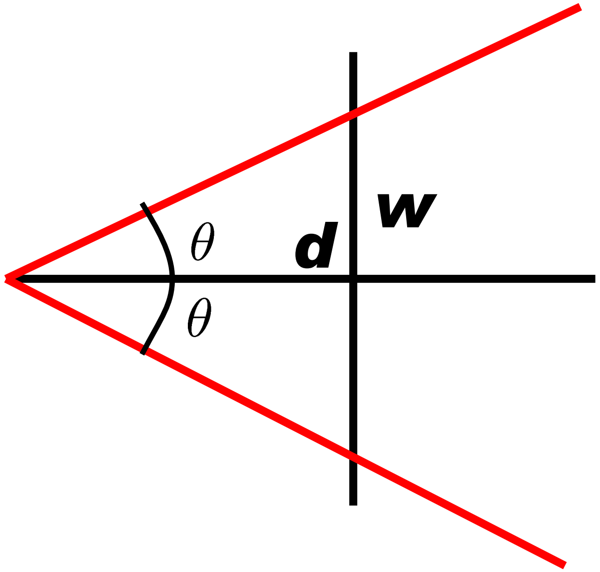
Proporzioni
più mantenute

Effetto "zoom"
(eg. vista dal
satellite)

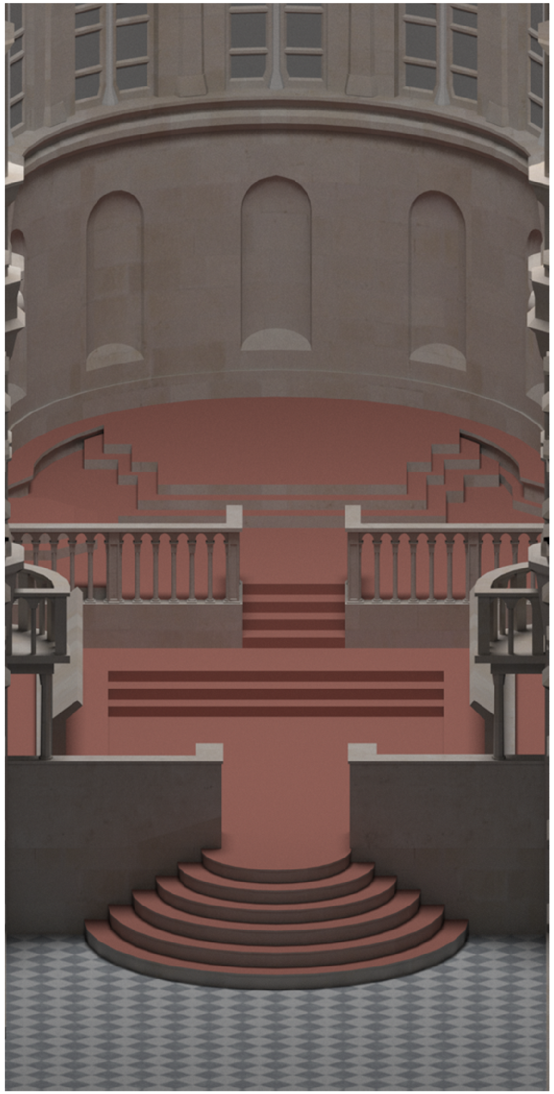
Field of View (fov)



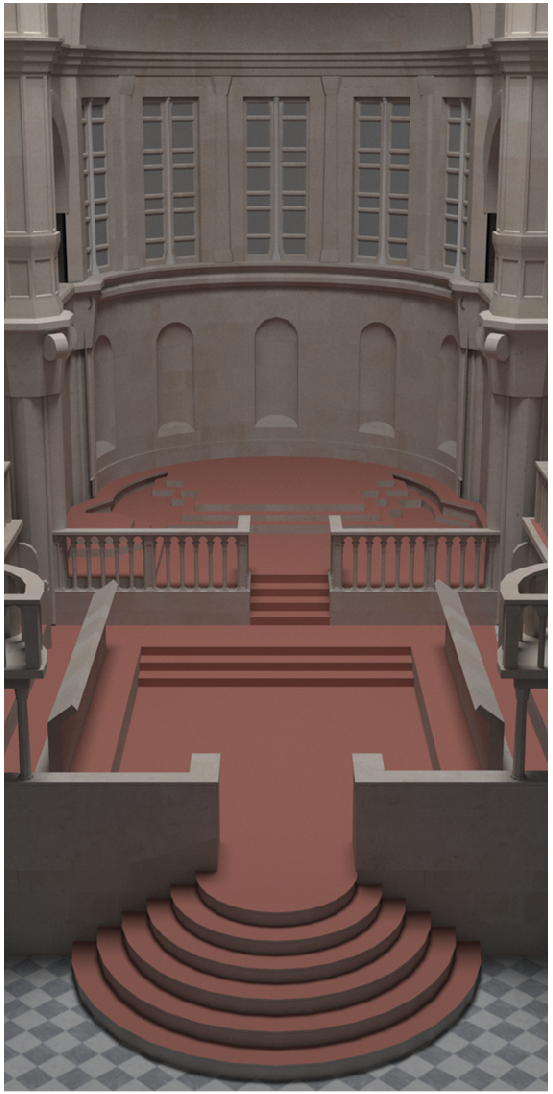




$$\phi = 2\theta \text{ (field of view)}$$
$$\theta = \arctan\left(\frac{w}{2d}\right)$$



Images from
Matt Pharr &
Greg Humpreys

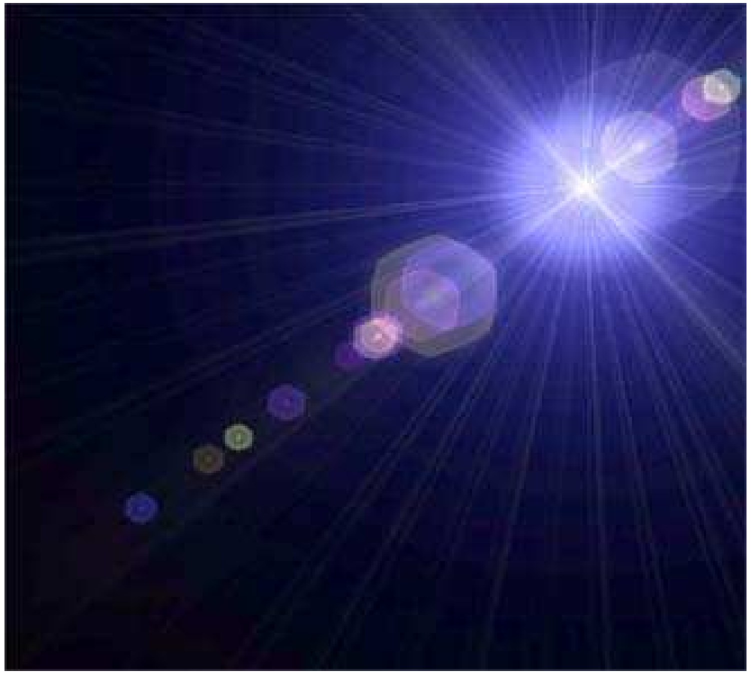


Proprietà delle trasformazioni geometriche

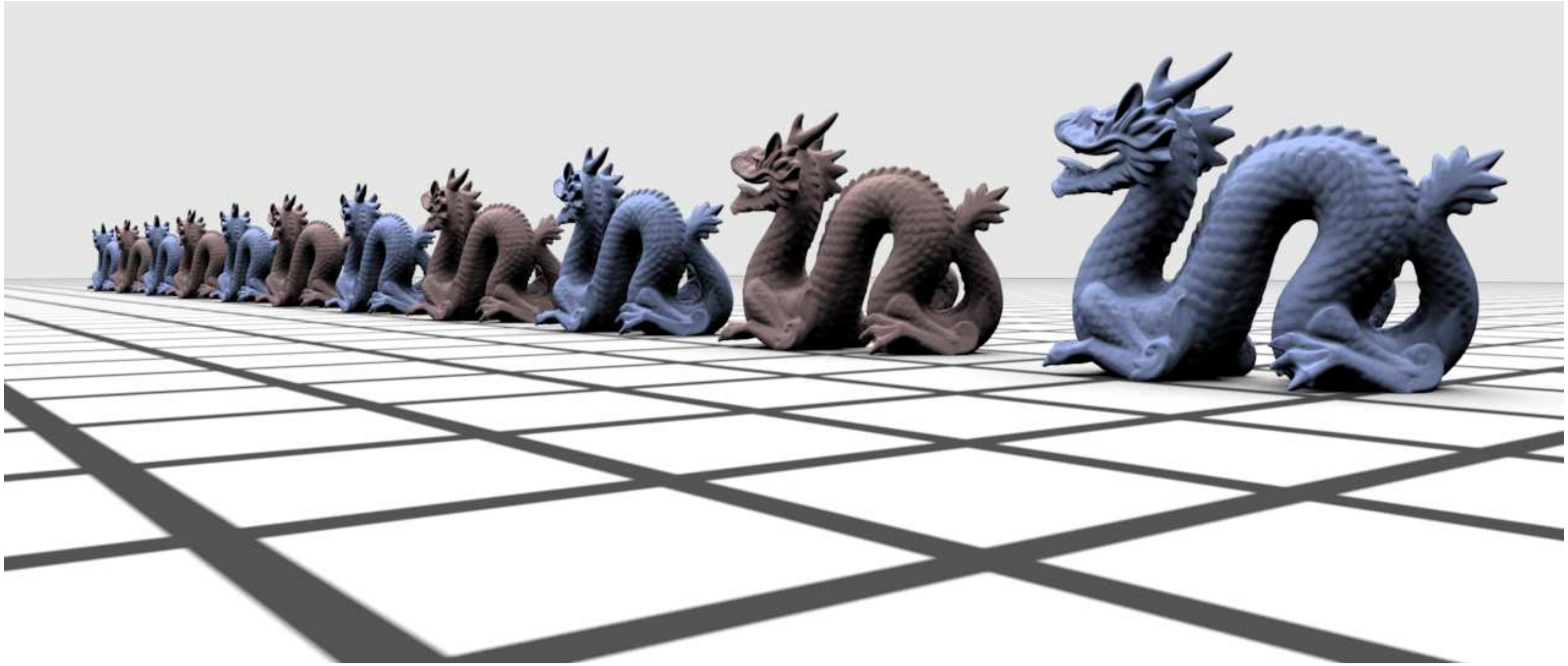
	Lunghezze	Angoli	Rapp. lunghezze	Colinearità
Traslazione	V	V	V	V
Rotazione	V	V	V	V
Scalatura uniforme	X	V	V	V
Scaltura non uniforme	X	X	V	V
Shearing	X	X	V	V
Proiezione Ortogonale	X	X	V	V
Proiezione Prospettica	X	X	X	V

- La pinhole è semplice ma poco realistica. Ad esempio la modellazione di eventuali lenti sarebbe utile per “simulare” una camera più realistica.
  - Con la pinhole abbiamo:
    - range di fuoco infinito (apertura infinitesima)
    - no flares (la luce non rimbalza nelle lenti…)
    - no distorsioni radiali (sempre dovute alle lenti)

Lens flares



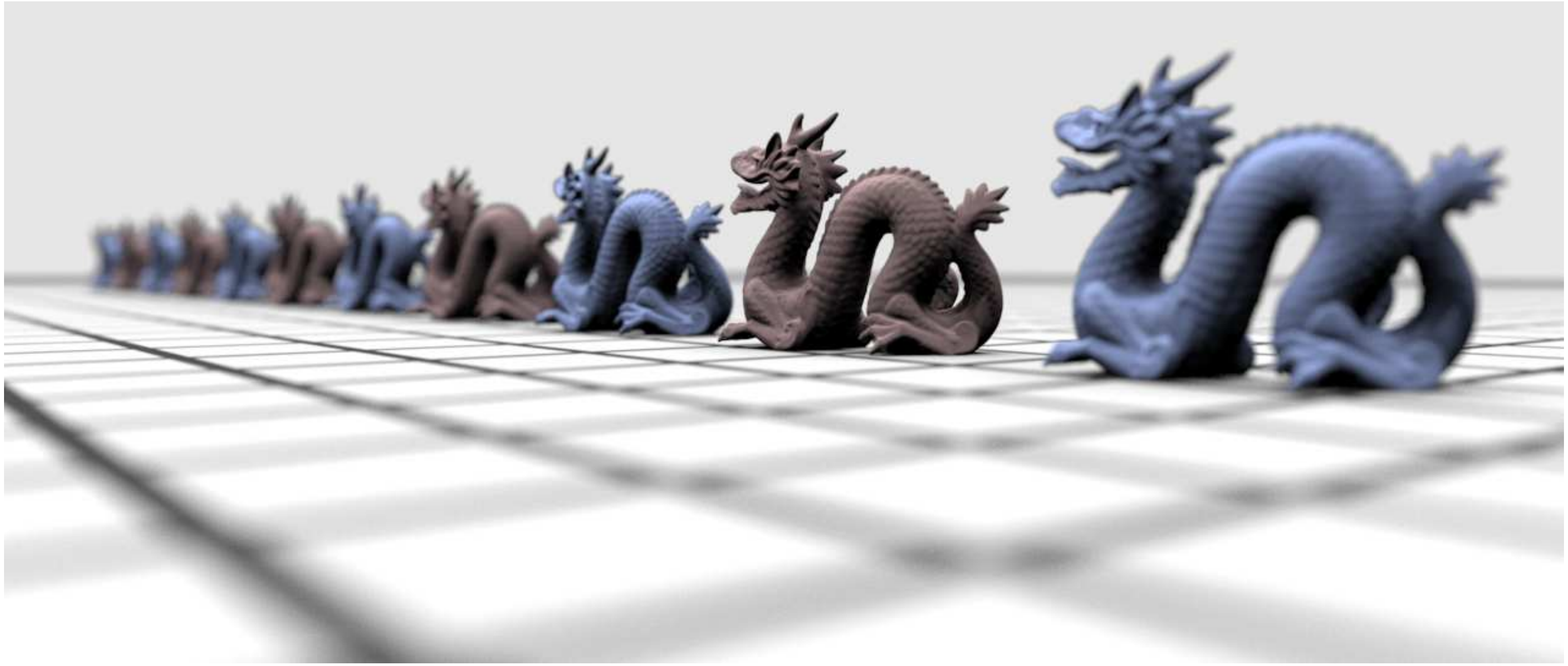
Profondità di Campo (Depth of Field)



Apertura otturatore puntiforme => fuoco infinito

Images from Matt Pharr & Greg Humpreys

Profondità di Campo (Depth of Field)



Apertura otturatore finita => soltanto alcuni oggetti a fuoco

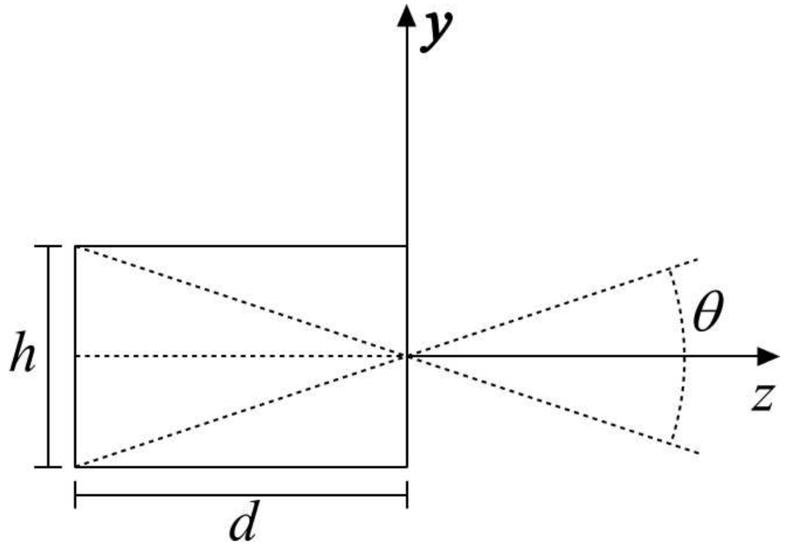
Images from Matt Pharr & Greg Humpreys

Profondità di Campo (Depth of Field)

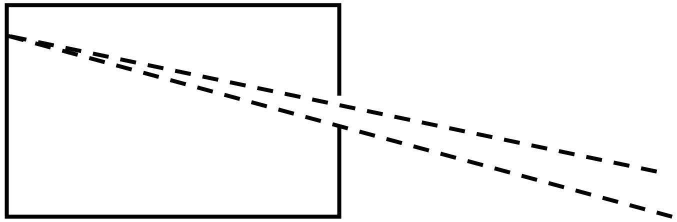
Apertura otturatore finita => soltanto alcuni oggetti a fuoco

Images from Matt Pharr & Greg Humpreys

Considerazioni sulla resa visiva della pinhole camera



Apertura otturatore infinitesima



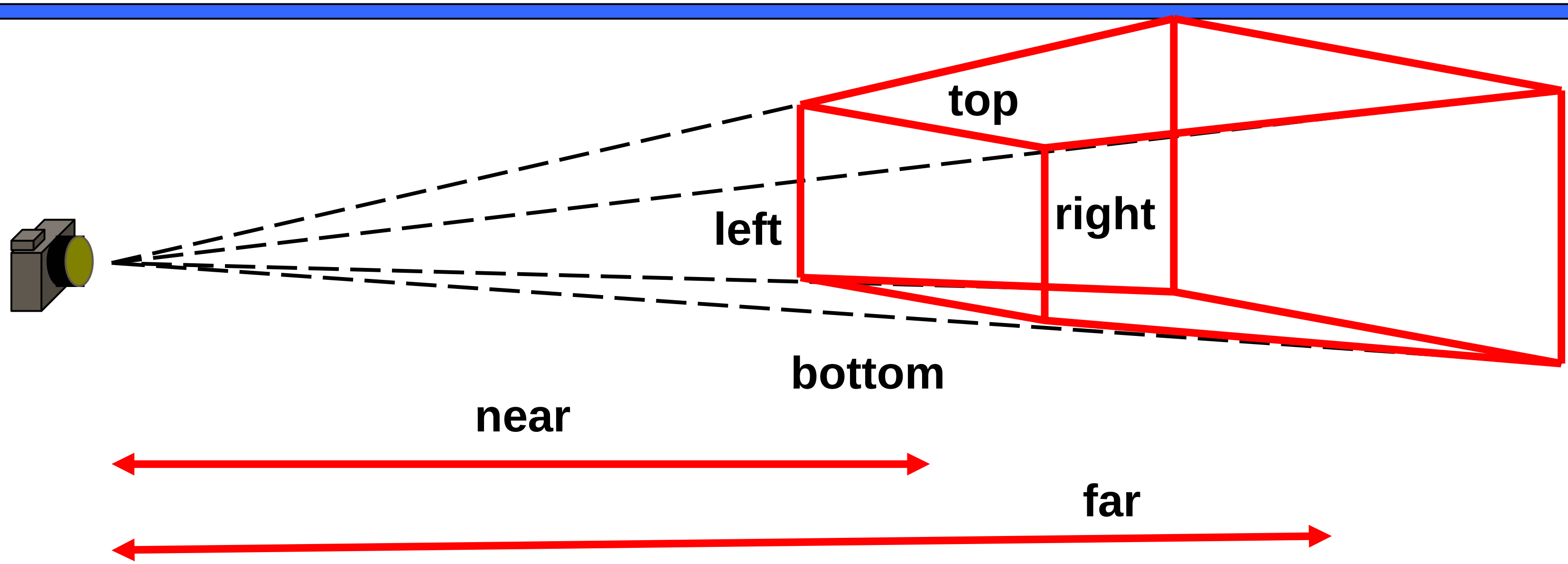
Apertura otturatore finita => più raggi colpiscono lo stesso punto della pellicola

NOTA: si tenga conto che la luce si propaga in maniera rettilinea, o più precisamente che la ***radianza*** di una fonte di luce è costante lungo una retta.

Normalized Device Coordinates

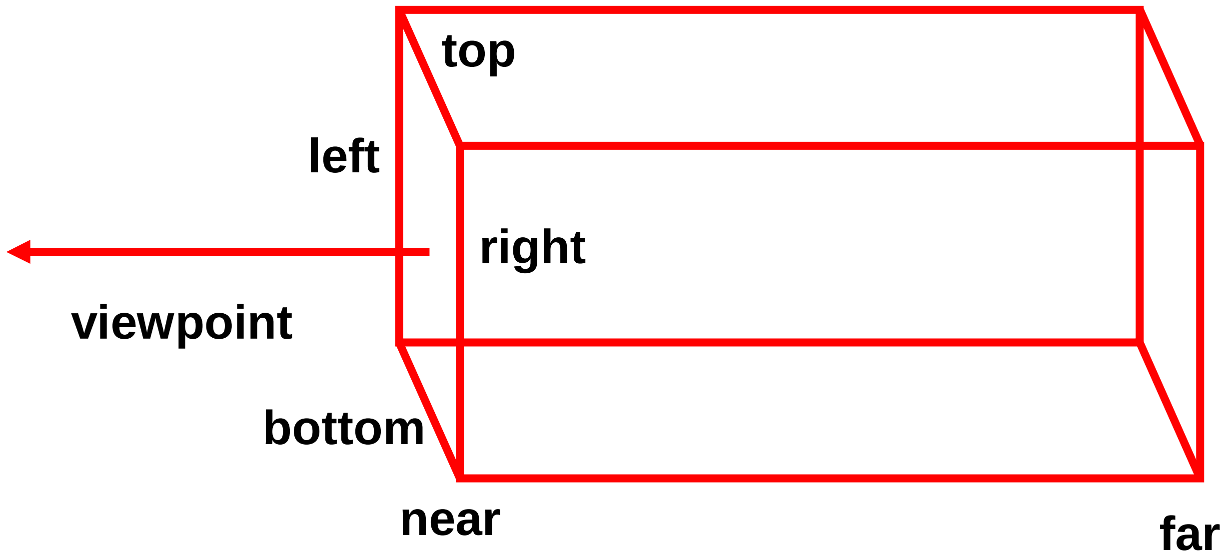
- Abbiamo visto la proiezione ortogonale e prospettica
- Queste devono essere tali da mappare il volume di vista (*view frustum*) relativo nelle cosiddette *Normalized Device Coordinates* ([-1 , 1] x [-1 , 1] x [-1 , 1])
- Le matrici P_{prsp} and P_{ort} devono essere modificate opportunamente

View Frustum



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

View Frustum



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

Screen Mapping

- La coordinata *z* non è interessata da questa trasformazione (è comunque passata allo stage di rasterizzazione)
- Le coordinate normalizzate *(x,y)* vengono mappate in *coordinate schermo* (x_{screen}, y_{screen}) essenzialmente attraverso un'operazione di traslazione e una di scalatura.
- Talvolta le *screen coordinates* vengono chiamate anche *window coordinates* (x_{window}, y_{window}).

$$\begin{pmatrix} x_{window} \\ y_{window} \end{pmatrix} = \begin{pmatrix} (w/2)x_{ndc} + o_x \\ (h/2)y_{ndc} + o_y \end{pmatrix}$$

w, h: larghezza ed altezza della viewport in pixel

o_x , o_y : centro della viewport in pixel

Domande?