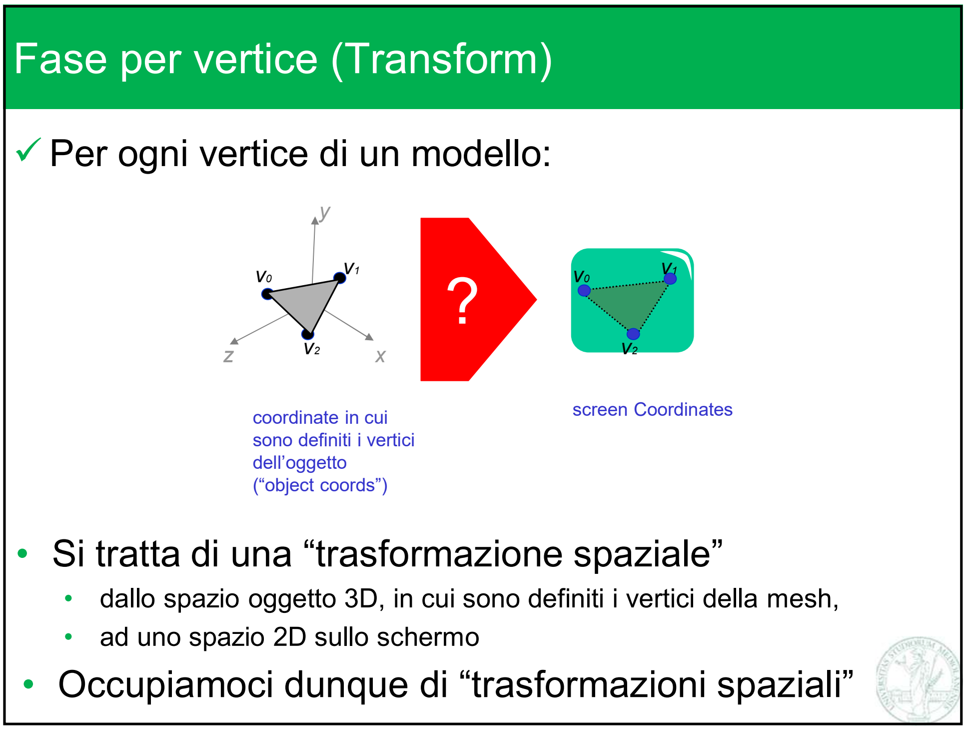
Marco Tarini - Computer Graphics 2020/2021
Università degli Studi di Milano

trasformazioni spaziali

TELEDIDATTICA!



Fase per vertice (Transform)

✓ Per ogni vertice di un modello:

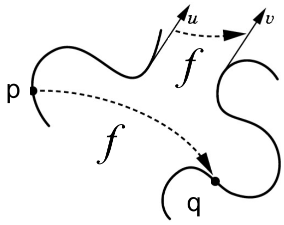
coordinate in cui
sono definiti i vertici
dell’oggetto
(“object coords”)

screen Coordinates

- Si tratta di una “trasformazione spaziale”
  - dallo spazio oggetto 3D, in cui sono definiti i vertici della mesh,
  - ad uno spazio 2D sullo schermo
- Occupiamoci dunque di “trasformazioni spaziali”

Trasformazioni geometriche (o spaziali)

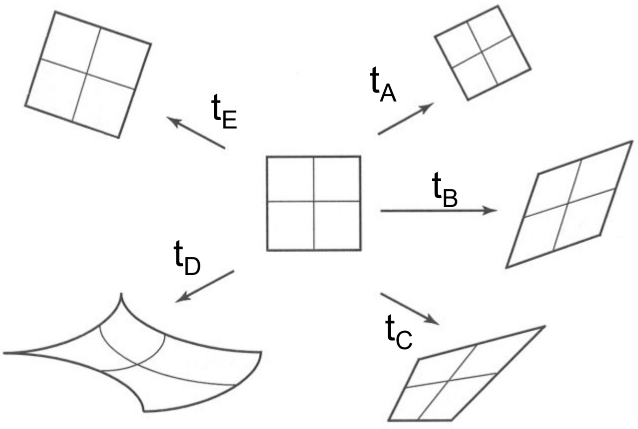
- ✓ Funzioni che
  - ⇒prendono un punto / vettore
  - ⇒lo mappano in un altro punto / vettore



$$q = f(p)$$
$$v = f(u)$$

Se input è un punto, output è un punto.
Se input è un vettore, output è un vettore

Trasformazioni spaziali: in generale

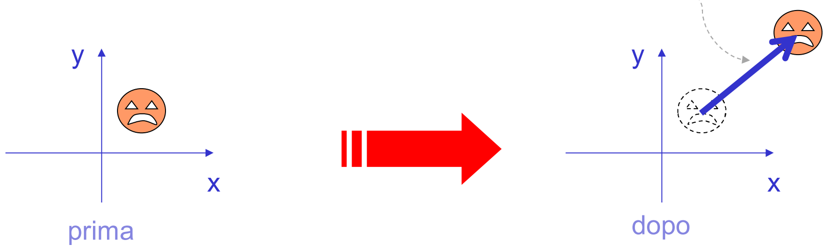


- ✓ Vediamo alcune classi utili di trasformazioni

Esempio: Trasformazione di **Traslazione**

$$f\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{pmatrix} = \begin{pmatrix} x+\alpha_x \\ y+\alpha_y \\ z+\alpha_z \end{pmatrix}$$

vettore di traslazione



Esempio: Trasformazione di **Traslazione**

✓ Osservazioni:

⇒ i punti vengono traslati

- es: le posizioni dei vertici della mesh

⇒ ma i vettori devono rimanere invariati

- es: le normali alla superficie
  la direzione: «da che parte guarda la faccia»
  il vettore «distanza fra gli occhi»
  non subiscono cambiamenti dopo la traslazione

$$f\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x+\alpha_x \\ y+\alpha_y \\ z+\alpha_z \end{pmatrix}$$

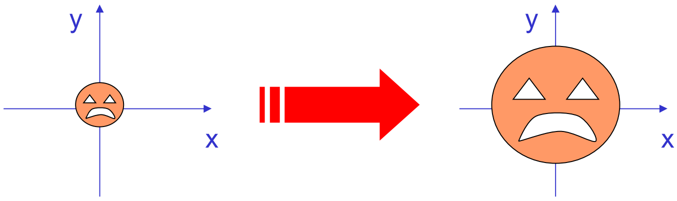
per punti

$$f\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

per vettori

Altro Esempio:
Trasformazione di **Scalatura** (uniforme)

$$f\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \gamma \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \gamma \cdot x \\ \gamma \cdot y \\ \gamma \cdot z \end{pmatrix}$$

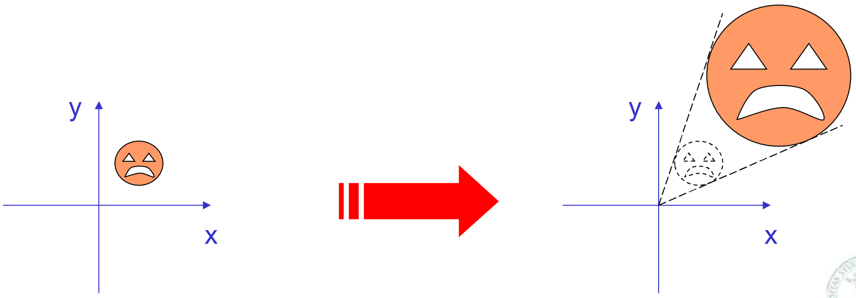


Esempio: trasformazione di **scalatura** (uniforme)

nota: stesso effetto su punti e vettori

(es: il vettore “distanza fra gli occhi” viene scalato)

nota: applicata ai punti
"scala" anche la distanza dall'origine



Esempio: trasformazione di **scalatura** uniforme

✓ Detta «**uniforme**» o «**isotropica**» perché applica lo stesso rapporto di scala a tutte e tre le coordinate

⇒quindi i vettori vengono scalati di una stessa quantità, indipendentemente dalla loro direzione

⇒vale non solo per vettori allineati all’asse X, Y o Z, ma anche per vettori orientati in qualsiasi direzione

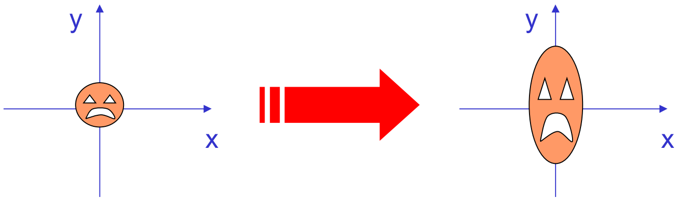
⇒Terminologia (in generale):

- Invarianza rispetto alla direzione = «**isotropia** »
- Dipendenza dalla direzione = «**anisotropia** »

Esempio: trasformazione di **scalatura** (generica)

$$f\begin{pmatrix}x\\y\\z\end{pmatrix} = \begin{pmatrix}\gamma_x\\\gamma_y\\\gamma_z\end{pmatrix}\cdot\begin{pmatrix}x\\y\\z\end{pmatrix} = \begin{pmatrix}\gamma_x\cdot x\\\gamma_y\cdot y\\\gamma_z\cdot z\end{pmatrix}$$

prodotto **componente per componente**
“component-wise product”
(non un’operazione canonica)

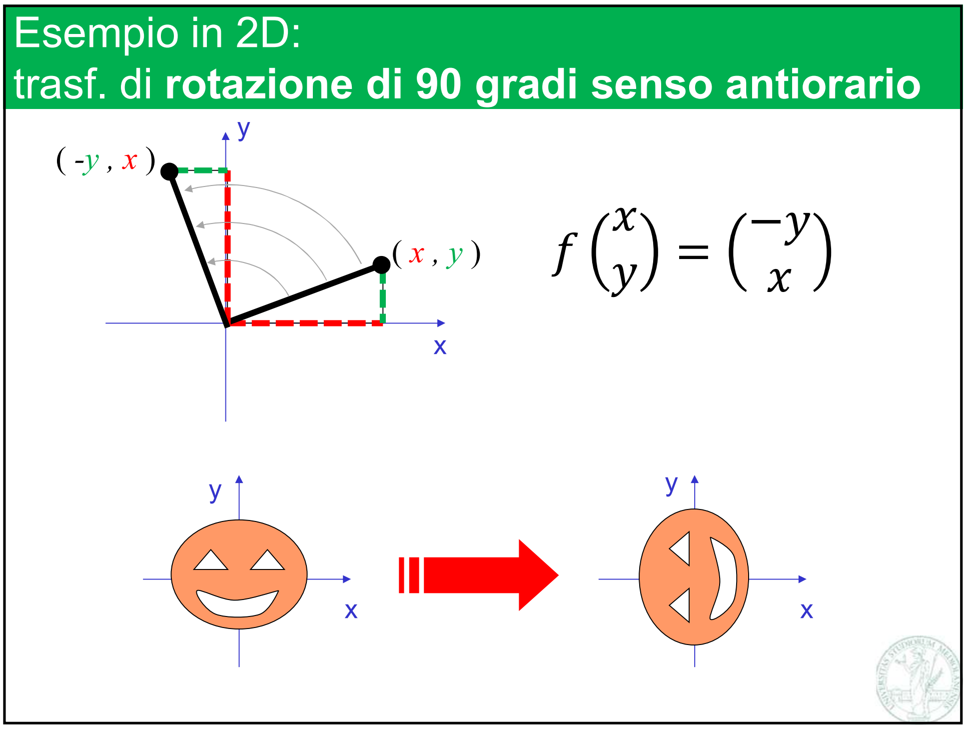


Trasformazione di Scalatura generica

- ✓ Osservazioni:
  - ⇒ Si applica tanto a punti che a vettori
    - (es: se ingrandisco un dinosauro – tutti i punti che lo definiscono, allora ingrandisco anche il vettore che connette la sua coda alla sua testa)
  - ⇒ Fattori di scala inferiori a 1 avvicinano i punti al origine del sistema di riferimento
  - ⇒ Fattori di scala maggiori di 1 lo allontanano
  - ⇒ L'unico punto che viene mappato su se stesso è l'origine

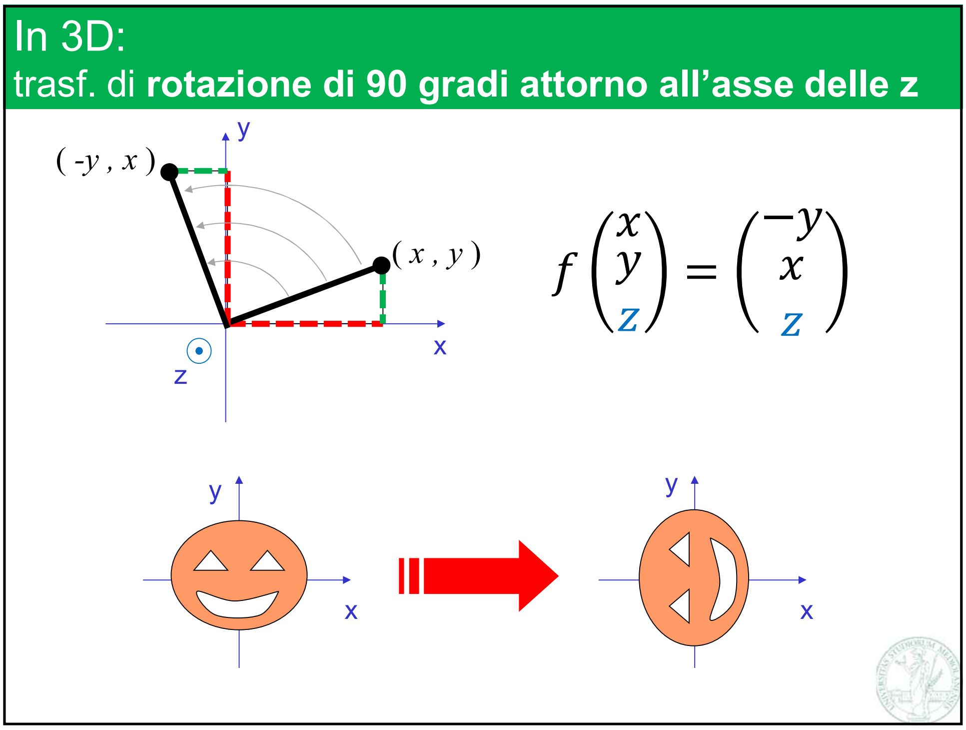
Trasformazione di Scalatura generica

- ✓ Se $s_x = s_y = s_z$ le proporzioni dell'oggetto sono mantenute. La scalatura viene detta
  - ⇒ uniforme (*x, y* e *z* sono soggetti a fattori uniformi)
  - ⇒ o conformale (mantiene la «forma»)
  - ⇒ o isotropica («una direzione vale l'altra»)
    - infatti non solo i vettori in dir x, y, e z subiscono un allungamento di s_x, ma anche quelli in qualsiasi altra direzione (verificare!)
- ✓ Altrimenti, le proporzioni **non** sono mantenute. La scalatura viene detta
  - ⇒ non uniforme
  - ⇒ o non conformale (non mantiene la forma, deforma)
  - ⇒ o anisotropica («dipende dalla direzione»)



Esempio in 2D:
trasf. di **rotazione di 90 gradi senso antiorario**

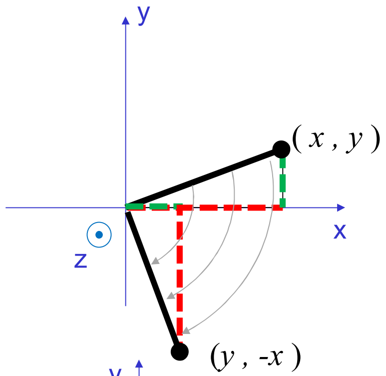
$$f\begin{pmatrix}x\\y\end{pmatrix}=\begin{pmatrix}-y\\x\end{pmatrix}$$



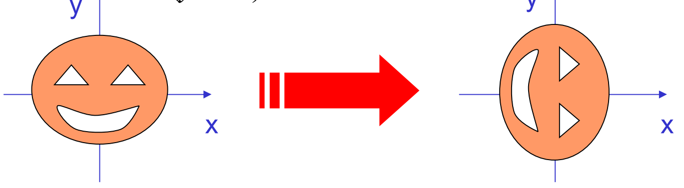
In 3D:
trasf. di **rotazione di 90 gradi attorno all’asse delle z**

$$f\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}-y\\x\\z\end{pmatrix}$$

Esempio in 2D: trasf. di **rotazione di 90 gradi senso orario**



$$f\begin{pmatrix}x\\y\\z\end{pmatrix}=\begin{pmatrix}y\\-x\\z\end{pmatrix}$$



Rappresentazione di punti e vettori in coordinate **omogenee**

Punti:

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

Vettori:

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

Rappresentazione di punti e vettori in coordinate **omogenee**

- ✓ Un **punto** di **coordinate cartesiane** (x,y,z)
  è rappresentato
  in **coordinate omogenee** come (x,y,z,**1**)

- ✓ Un **vettore** di **coordinate cartesiane** (x,y,z)
  è rappresentato
  in **coordinate omogenee** come (x,y,z,**0**)

- ✓ Nota: (da verificare!)
  questo è coerente con l'algebra di punti e vettori :
    punto – punto = vettore
    punto + vettore = punto

Punti e vettori in coordinate **omogenee** (note in GLSL)

- ✓ In GLSL:

x y z **w**

```
vec4 foo = vec4(3.0,2.0,4.1,1.0);
```

Passare da coord. omogenee a cartesiane:
(cioè estrarre le prime tre componenti di un vec4)

```
vec3 pippo = foo.xyz;
```

Viceversa (per un vettore):
(cioè aggiungere la coord. w)

x,y,z w

```
vec4 boo = vec4( pippo , 0);
```

Trasformazioni come moltiplicazioni per matrici

- ✓ Esprimendo punti e vettori in coordinate omogenee (in un vettore colonna), possiamo esprimere le trasformazioni viste come una moltiplicazione di una matrice 4x4 M
  - ⇒M è detta la sua matrice di trasformazione
  - ⇒Questo varrà anche per tutte le altre trasformazioni che vedremo
  - ⇒Una trasformazione esprimibile in questo modo è detta «affine»

Trasformazioni Affini

- ✓ Quelle che posso esprimere come moltiplicazione con matrice 4x4 :

$$f\begin{pmatrix}\gamma_1\\ \gamma_2\\ \gamma_3\\ 1\end{pmatrix} = \begin{bmatrix}\alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14}\\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24}\\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34}\\ \underbrace{0 \quad 0 \quad 0 \quad 1}_{\text{sempre}}\end{bmatrix}\begin{bmatrix}\gamma_1\\ \gamma_2\\ \gamma_3\\ 1\end{bmatrix} = \begin{pmatrix}\delta_1\\ \delta_2\\ \delta_3\\ 1\end{pmatrix}$$

punto di partenza
in coordinate affini

punto di arrivo
in coordinate affini

Trasformazioni Affini

- ✓ Quelle che posso esprimere come moltiplicazione con matrice 4x4 :

conta solo questo

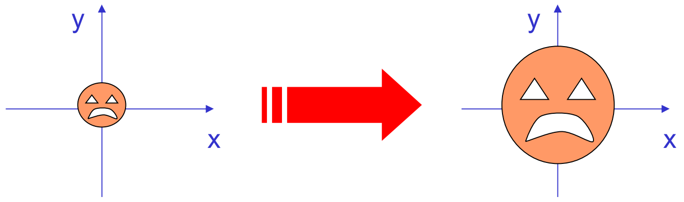
$$f\begin{pmatrix}\gamma_1\\ \gamma_2\\ \gamma_3\\ 0\end{pmatrix}=\begin{bmatrix}\alpha_{11}&\alpha_{12}&\alpha_{13}&\alpha_{14}\\ \alpha_{21}&\alpha_{22}&\alpha_{23}&\alpha_{24}\\ \alpha_{31}&\alpha_{32}&\alpha_{33}&\alpha_{34}\\ 0&0&0&1\end{bmatrix}\begin{bmatrix}\gamma_1\\ \gamma_2\\ \gamma_3\\ 0\end{bmatrix}=\begin{pmatrix}\delta_1\\ \delta_2\\ \delta_3\\ 0\end{pmatrix}$$

sempre

vettore di partenza in coordinate affini

vettore di arrive in coordinate affini

Matrice di trasformazione: scaling isotropico



$$f\begin{pmatrix}x\\ y\\ z\\ 1\end{pmatrix}=\begin{pmatrix}\gamma\, x\\ \gamma\, y\\ \gamma\, z\\ 1\end{pmatrix}$$

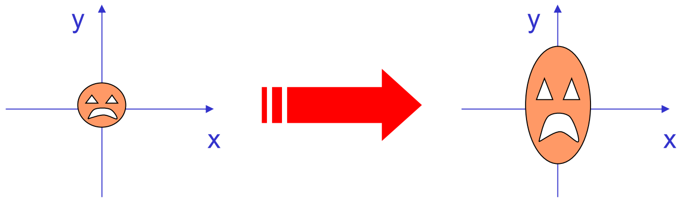
$$\begin{bmatrix}\gamma&0&0&0\\ 0&\gamma&0&0\\ 0&0&\gamma&0\\ 0&0&0&1\end{bmatrix}\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}=\begin{bmatrix}\gamma\, x\\ \gamma\, y\\ \gamma\, z\\ 1\end{bmatrix}$$

$\mathbf{S}_\gamma$

matrice di scaling isotropico

Cosa succede se la applico $\mathbf{S}_\gamma$ ad un vettore?

Matrice di trasformazione: scaling anisotropico



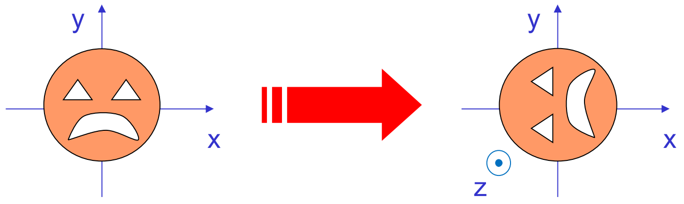
$$f\begin{pmatrix}x\\y\\z\\1\end{pmatrix}=\begin{pmatrix}\gamma_x\ x\\\gamma_y\ y\\\gamma_z\ z\\1\end{pmatrix}$$

$$\begin{bmatrix}\gamma_x & 0 & 0 & 0\\0 & \gamma_y & 0 & 0\\0 & 0 & \gamma_z & 0\\0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}=\begin{bmatrix}\gamma_x\ x\\\gamma_y\ y\\\gamma_z\ z\\1\end{bmatrix}$$

$\mathbf{S}_{\gamma_x,\gamma_y,\gamma_z}$

matrice di scaling anisotropico

Matrice di trasformazione: rotazione attorno all'asse delle Z



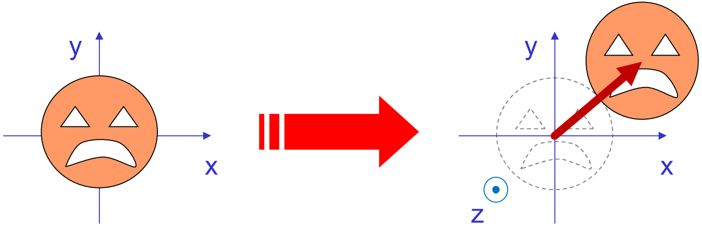
$$f\begin{pmatrix}x\\y\\z\\1\end{pmatrix}=\begin{pmatrix}-y\\x\\z\\1\end{pmatrix}$$

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

$\mathbf{R}_{z,90°}$

matrice di rotazione di 90° attorno all'asse delle Z

Matrice di trasformazione: traslazione



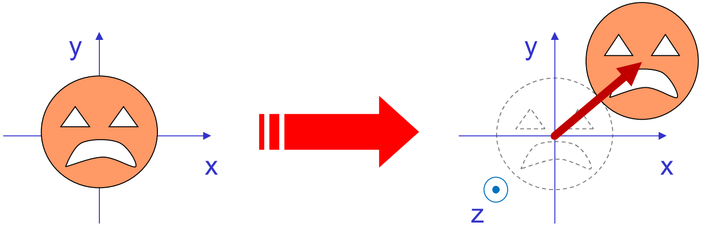
$$f\begin{pmatrix}x\\y\\z\\1\end{pmatrix}=\begin{pmatrix}x+t_x\\y+t_y\\z+t_z\\1\end{pmatrix}$$

$$\begin{bmatrix}1&0&0&t_x\\0&1&0&t_y\\0&0&1&t_z\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}=\begin{bmatrix}x+t_x\\y+t_y\\z+t_z\\1\end{bmatrix}$$

$\mathbf{T}_{t_x,t_y,t_z}$

matrice di traslazione
del vettore $\mathbf{t}=(t_x,t_z,t_z)$

Matrice di trasformazione: traslazione (applicate ai vettori)



$$f\begin{pmatrix}x\\y\\z\\0\end{pmatrix}=\begin{pmatrix}x\\y\\z\\0\end{pmatrix}$$

$$\begin{bmatrix}1&0&0&t_x\\0&1&0&t_y\\0&0&1&t_z\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\0\end{bmatrix}=\begin{bmatrix}x\\y\\z\\0\end{bmatrix}$$

la stessa matrice
applicata ad un **vettore**
lo lascia invariato
(come ci aspettiamo
da una traslazione)

$\mathbf{T}_{t_x,t_y,t_z}$

matrice di traslazione
del vettore $\mathbf{t}=(t_x,t_y,t_z)$

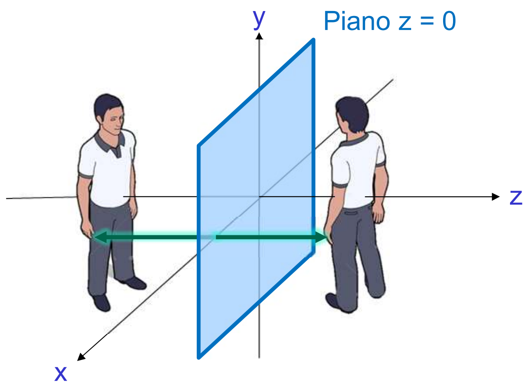
Matrice di trasformazione: traslazione

cosa succede quando la applico ad un *vettore* ?

$$f\begin{pmatrix}x\\y\\z\\0\end{pmatrix} = \begin{bmatrix}1&0&0&\alpha_x\\0&1&0&\alpha_y\\0&0&1&\alpha_z\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\0\end{bmatrix} = \begin{pmatrix}x\\y\\z\\0\end{pmatrix}$$

coerentemente con il comportamento atteso,
il vettore sottoposto alla traslazione
rimane invariato

Matrice di simmetria speculare (planare)



$$f\begin{pmatrix}x\\y\\z\end{pmatrix} = \begin{pmatrix}x\\y\\-z\end{pmatrix}$$

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

Alcune trasformazioni affini utili espresse attraveso le loro matrici

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{T}_{t_x,t_y,t_z}$

matrice di Traslazione
del vettore **t** $= (t_x, t_y, t_z)$

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

$\mathbf{M}_z$

matrice di simmetria (Mirroring)
del piano $z = 0$

$$\begin{bmatrix} \gamma_x & 0 & 0 & 0 \\ 0 & \gamma_y & 0 & 0 \\ 0 & 0 & \gamma_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{S}_{\gamma_x,\gamma_y,\gamma_z}$

matrice di Scaling
anisotropico

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

$\mathbf{R}_{z,90°}$

matrice di Rotazione
di 90° attorno all'asse delle z

Inversa della matrice di scalatura

matrice di traslazione:

$$\mathbf{S}_{\gamma_x,\gamma_y,\gamma_z} = \begin{bmatrix} \gamma_x & 0 & 0 & 0 \\ 0 & \gamma_y & 0 & 0 \\ 0 & 0 & \gamma_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

l'inversa è, logicamente:

$$(\mathbf{S}_{\gamma_x,\gamma_y,\gamma_z})^{-1} = \mathbf{S}_{1/\gamma_x,1/\gamma_y,1/\gamma_z} = \begin{bmatrix} 1/\gamma_x & 0 & 0 & 0 \\ 0 & 1/\gamma_y & 0 & 0 \\ 0 & 0 & 1/\gamma_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inversa della matrice di traslazione

matrice di traslazione: $$\mathbf{T}(\alpha_x,\alpha_y,\alpha_z)=\begin{bmatrix}1&0&0&\alpha_x\\0&1&0&\alpha_y\\0&0&1&\alpha_z\\0&0&0&1\end{bmatrix}$$

l'inversa è, logicamente:

$$\mathbf{T}^{-1}(\alpha_x,\alpha_y,\alpha_z)=\mathbf{T}(-\alpha_x,-\alpha_y,-\alpha_z)=\begin{bmatrix}1&0&0&-\alpha_x\\0&1&0&-\alpha_y\\0&0&1&-\alpha_z\\0&0&0&1\end{bmatrix}$$

Inversa di matrice di simmetria speculare

matrice di traslazione: $$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&-1&0\\0&0&0&1\end{bmatrix}$$

l'inversa è, logicamente, la matrice stessa