


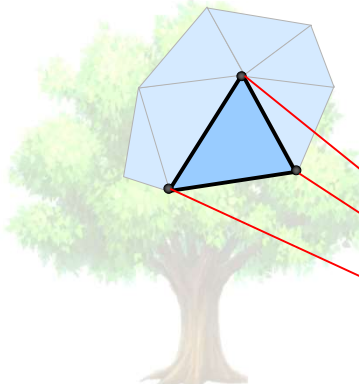
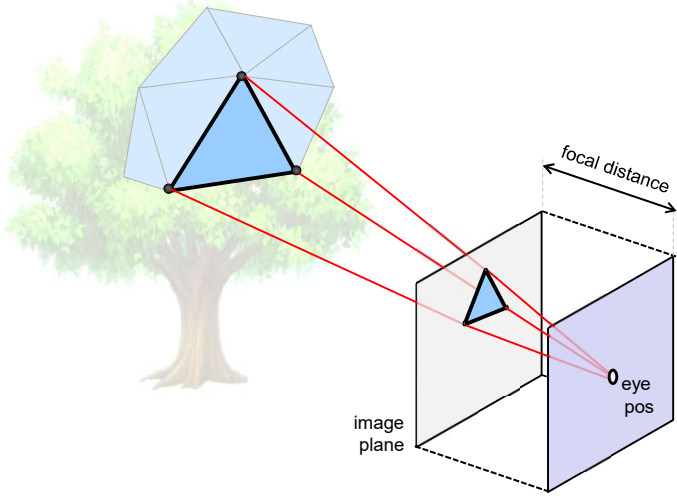
Marco Tarini - Computer Graphics 2025/2026
Università degli Studi di Milano

trasformazioni spaziali



1

Rasterization based rendering



2

La fase per vertice del rasterization-based rendering (la «transform» di «transform and lighting»)

- Si tratta di una “trasformazione spaziale”
- Occupiamoci dunque di “trasformazioni spaziali”

3

Trasformazioni spaziali

$$\mathbf{q} = f(\mathbf{p})$$
$$\vec{\mathbf{v}} = f(\vec{\mathbf{u}})$$

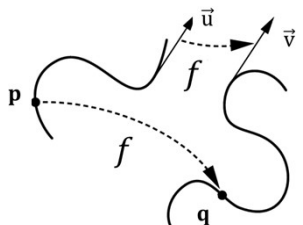
5

Trasformazioni spaziali

✓ Funzioni che


- ⇒ prendono un punto / vettore
- ⇒ lo mappano in un altro punto / vettore

Trasformando tutti i punti e i vettori della rappresentazione di un modello 3D, (come: le posizioni dei vertici di una mesh, i punti di controllo di una superficie parametrica, la mesh di controllo di una sup. di suddivisione...) trasformo spazialmente l'oggetto rappresentato



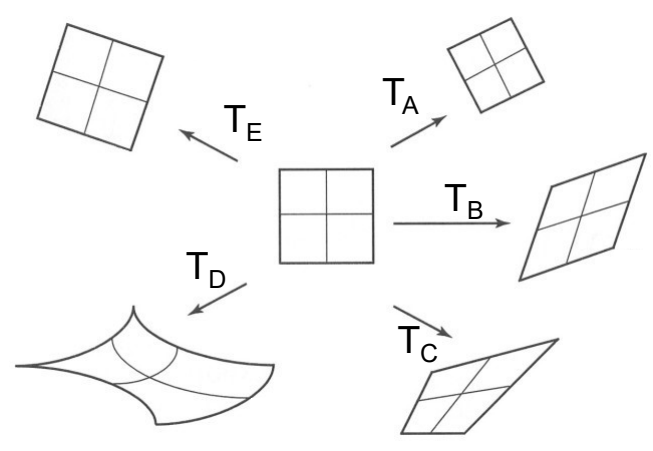
$\mathbf{q} = f(\mathbf{p})$
 $\vec{\mathbf{v}} = f(\vec{\mathbf{u}})$

Se l'input di f è un punto, allora l'output è un punto.
Se l'input di f è un vettore allora l'output è un vettore




6

Trasformazioni spaziali: in generale



Vediamo alcuni semplici esempi



7

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

prima dopo

8

Esempio: Trasformazione di Traslazione

✓ Osservazioni:


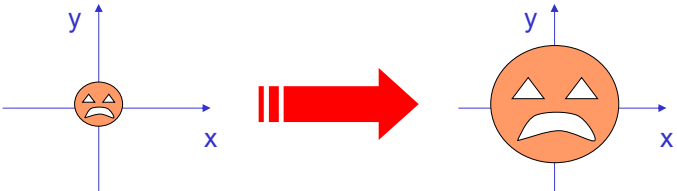
- ⇒ i punti vengono traslati
 - es: le posizioni dei vertici della mesh
- ⇒ **ma i vettori devono rimanere invariati**
 - es: le normali alla superficie
 - la direzione: «da che parte guarda la faccia»
 - il vettore «distanza fra gli occhi»
 - non subiscono cambiamenti dopo la traslazione

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

per punti per vettori

9

Esempio:
Trasformazione di Scalatura (uniforme)


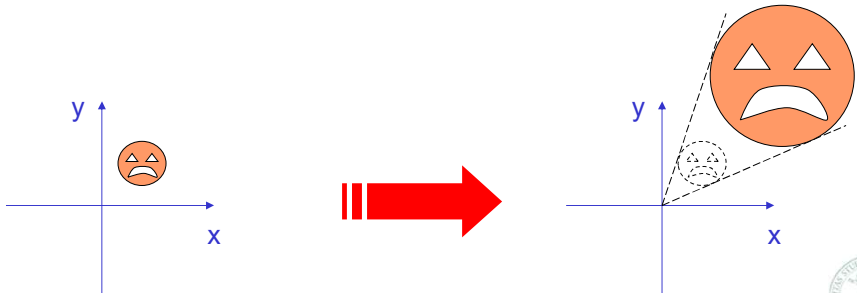
$$f \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \gamma \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \gamma \cdot x \\ \gamma \cdot y \\ \gamma \cdot z \end{pmatrix}$$


10

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

nota: stesso effetto su punti e vettori
(es: il vettore "distanza fra gli occhi" viene scalato)

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



11

Esempio: trasformazione di scalatura uniforme

- ✓ Detta «**uniforme**»
 - ⇒ perché applica lo stesso rapporto di scala a X,Y,Z
 - ⇒ anche a qualsiasi direzione, oblique comprese
- ✓ oppure «**isotropica**»
 - ⇒ perché 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 questo o altri contesti):
 - Invarianza rispetto alla direzione = «**isotropia**»
 - Dipendenza dalla direzione = «**anisotropia**»
- ✓ oppure «**conformale**»
 - ⇒ Perché mantiene gli angoli, cioè le forme



12

Trasformazione di Scalatura (di qualsiasi tipo)

- ✓ 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 all'origine del sistema di riferimento
 - ⇒ Fattori di scala maggiori di 1 lo allontanano
 - ⇒ Fattore di scala uguale ad 1: funzione identità
 - ⇒ L'unico punto che viene sempre mappato su se stesso (cioè che rimane invariato) è l'origine, il punto (0,0,0)
 - ⇒ E l'unico vettore che viene sempre mappato su se stesso (cioè che rimane invariato) è il vettore degenere (0,0,0)



13

Esempio: trasformazione di scalatura (anisotropica)

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

prodotto **componente per componente**
 "component-wise product"
 (non un'operazione canonica)

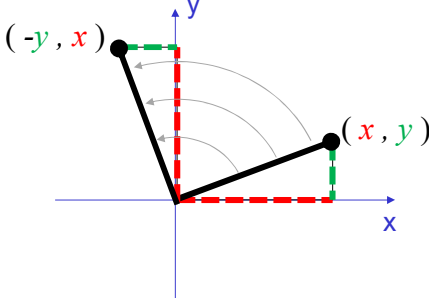

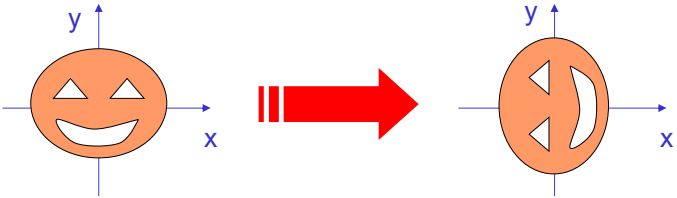
14

Trasformazione di Scalatura: terminologia

- ✓ Se i fattori di scala sono uguali, cioè se $\gamma_x = \gamma_y = \gamma_z$, allora le proporzioni dell'oggetto sono mantenute.
 E la scalatura viene detta
 - ⇒ **uniforme** (x, y e z sono soggetti a fattori uniformi)
 - ⇒ o **conformale** (mantiene la «forma»)
 - ⇒ o **isotropica** («qualsiasi direzione viene trattata nello stesso modo»)
 - e non solo i vettori lungo gli assi x, y, e z subiscono un allungamento di s_x , ma anche quelli in qualsiasi altra direzione obliqua (verificare!)
- ✓ Altrimenti, le proporzioni **non** sono mantenute.
 La scalatura viene detta
 - ⇒ **non uniforme**
 - ⇒ o **non conformale** (non mantiene la forma (delle mesh, etc), deforma)
 - ⇒ o **anisotropica** («dipende dalla direzione» (del vettore trasformato))

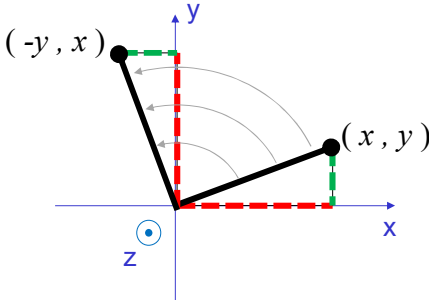

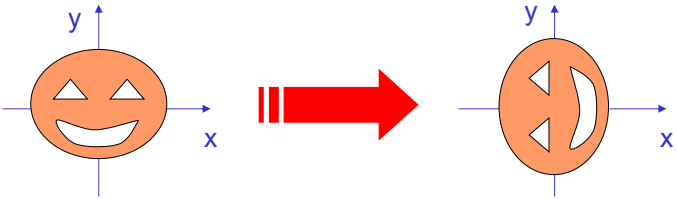
15

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


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


16

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


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


17

**In 3D:
trasf. di rotazione di 90 gradi senso orario**

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

18

Rotazioni di 90° (note)

- ✓ In 2D, la rotazione è attorno ad un punto:
nel nostro esempio sopra, l'origine.
 - ⇒ L'origine è l'unico punto che rimane invariato
 - ⇒ Osserva un disegno su un foglio a quadretti:
per ruotare attorno all'origine, basta:
scambiare fra loro le coordinate x e y, e cambiare il segno ad una delle due (quale? Questo determina il verso della rotazione, se in senso orario o antiorario)
- ✓ In 3D, la rotazione è attorno ad una retta
(l'asse di rotazione):
nel nostro esempio sopra, la retta che coincide con l'asse delle Z (in generale, una qualsiasi retta)
 - ⇒ I punti sull'asse di rotazione (qui: la Z) rimangono invariati


19

Rappresentazione di punti e vettori in coordinate **omogenee** (o **affini**)

Punti:

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


Vettori:

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


20

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


- ✓ Un **punto** di **coordinate cartesiane** (x, y, z) è rappresentato in **coordinate omogenee** come $(x, y, z, \mathbf{1})$
- ✓ Un **vettore** di **coordinate cartesiane** (x, y, z) è rappresentato in **coordinate omogenee** come $(x, y, z, \mathbf{0})$
- ✓ La coordinata affine viene spesso denotata dalla lettera w
- ✓ Nota (verificalo!) questo è consistente con l'algebra di punti e vettori (applicando l'operazione a x, y, z, w):
 - punto – punto = vettore
 - punto + vettore = punto
 - vettore \pm vettore = vettore
 - mix(punto, punto, t) = punto
 - mix(vettore, vettore, t) = punto, etc



21

Trasformazioni spaziali 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 «matrice di trasformazione»
 - ⇒ Questo varrà anche per tutte le altre trasformazioni che vedremo
 - ⇒ per questo motivo, in CG «matrice 4x4» è in pratica un sinonimo di «trasformazione spaziale»
 - ⇒ (Una qualsiasi trasformazione che io possa esprimere in questo modo è detta «affine»)



25

Trasformazioni Affini

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

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


sempre

↑

punto di partenza
in coordinate affini

↑

punto di arrivo
in coordinate affini



26

Matrice di trasformazione: scaling isotropico

$$f \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \gamma x \\ \gamma y \\ \gamma z \\ 1 \end{pmatrix} \quad \begin{bmatrix} \gamma & 0 & 0 & 0 \\ 0 & \gamma & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma x \\ \gamma y \\ \gamma z \\ 1 \end{bmatrix}$$

S_γ
matrice di scaling isotropico

Cosa succede se la applico S_γ ad un vettore?

27

Matrice di trasformazione: scaling anisotropico

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

$S_{\gamma_x, \gamma_y, \gamma_z}$
matrice di scaling anisotropico

29

Matrice di trasformazione: rotazione attorno all'asse delle Z

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

$R_{z,90^\circ}$
matrice di rotazione
di 90° attorno all'asse delle Z

30

Matrice di trasformazione: traslazione

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

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

32

Matrice di trasformazione: traslazione (applicate ai vettori)

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

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

T_{*t_x, t_y, t_z*}
 matrice di traslazione
 del vettore **t** = (*t_x, t_y, t_z*)

33

Matrice di trasformazione: traslazione

cosa succede quando la applico ad un *vettore* ?

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

coerentemente con il comportamento atteso,
 “traslare un vettore”
 lascia il vettore rimane invariato

34

Trasformazioni Affini

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


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

per i vettori, conta solo questo

sempre

vettore di partenza
in coordinate affini

vettore di arrive
in coordinate affini




35

Trasformazioni Affini (o lineari)

- ✓ Le trasformazioni esprimibili come una moltiplicazione per una matrice M 4x4 al vettore di coordinate omogenee sono dette «affini»
 - ⇒ L'ultima riga della M è sempre 0,0,0,1, in modo che i punti vengano mappati su punti, e i vettori in vettori
- ✓ E' classe di trasformazioni spaziali particolarmente utile in CG
 - ⇒ (quasi) tutte le trasformazioni spaziali che ci interessano appartengono a questa classe
 - ⇒ Al punto che, in CG, «matrice» (4x4) è praticamente sinonimo di «trasformazione» (spaziale)
- ✓ Comprende tutte le trasformazioni che abbiamo visto fin'ora
 - ⇒ traslazione, scalature uniformi e non, rotazione di 90 gradi attorno a Z, ribaltamenti...

e altre, che vedremo la prossima volta

- ⇒ Rotazioni in generale, shearing...



36