

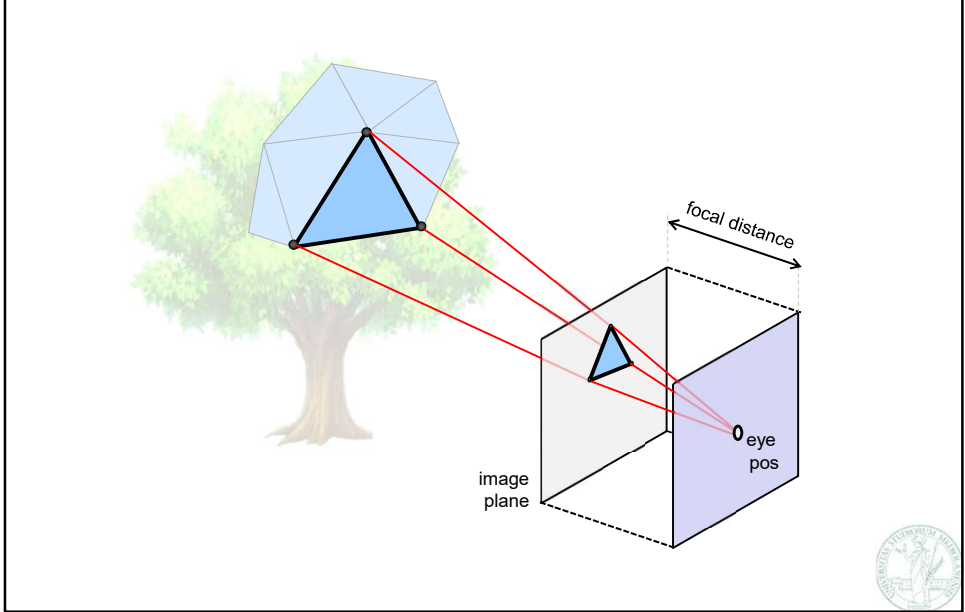
Marco Tarini - Computer Graphics 2024/2025
Università degli Studi di Milano

trasformazioni spaziali

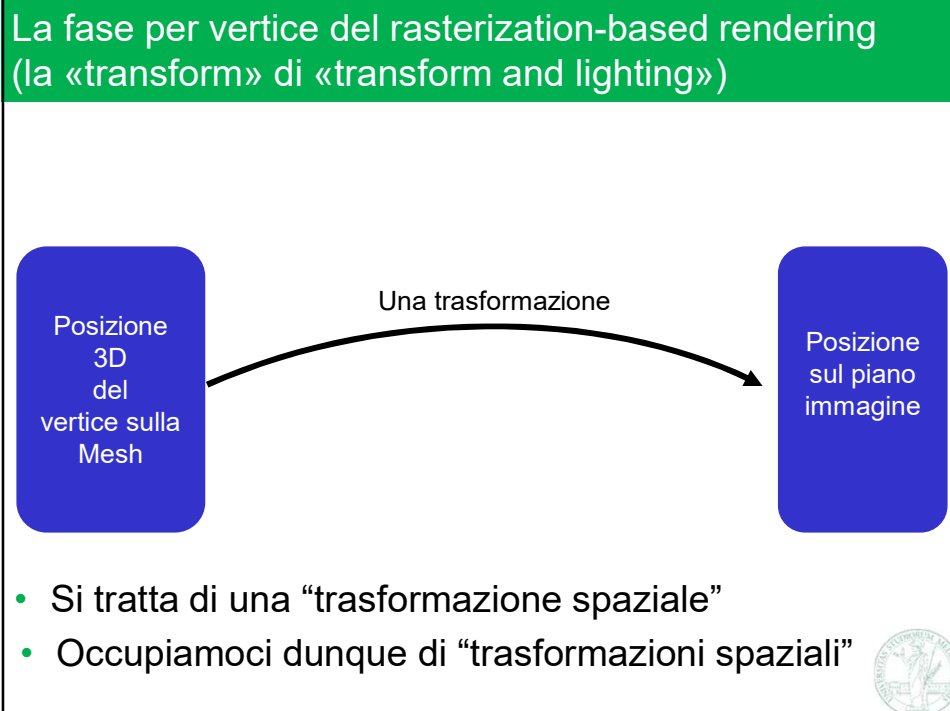


1

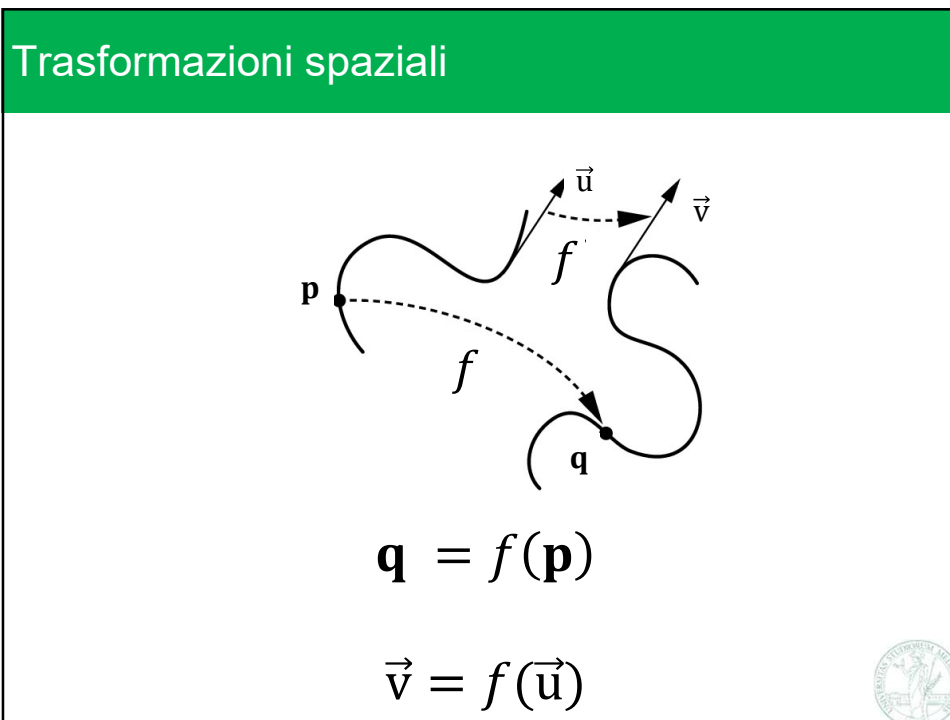
Rasterization based rendering



2



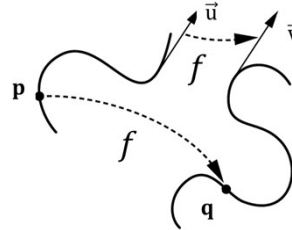
3



5

Trasformazioni spaziali

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



$$\mathbf{q} = f(\mathbf{p})$$

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

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

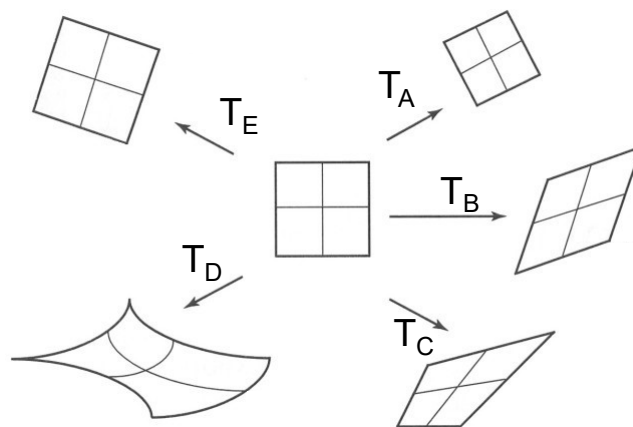
Se l'input di f è un punto, allora l'output è un punto.

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



6

Trasformazioni spaziali: in generale



Vediamo alcuni semplici esempi

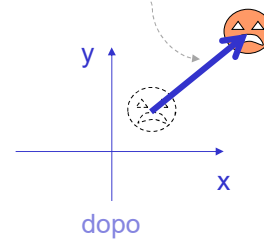
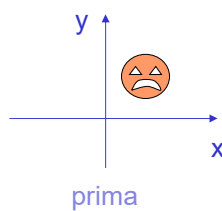


7

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



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

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

per punti

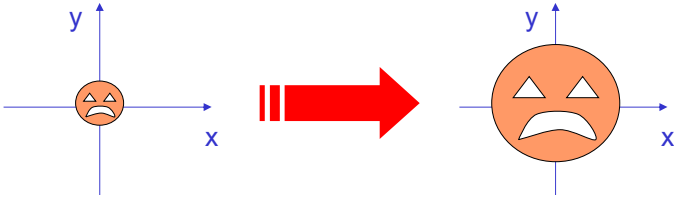
$$f \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

per vettori

9

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

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



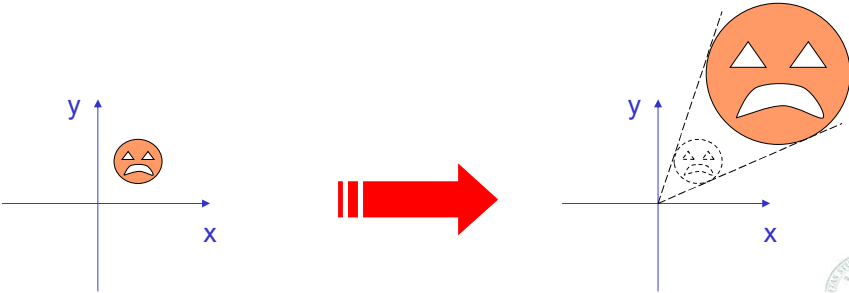


10

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**» 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 questo o altri contesti):
 - Invarianza rispetto alla direzione = «**isotropia** »
 - Dipendenza dalla direzione = «**anisotropia** »



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 $(0,0,0)$

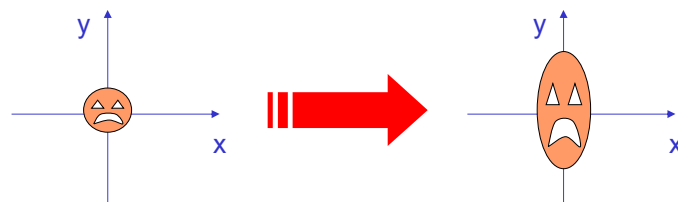


13

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)



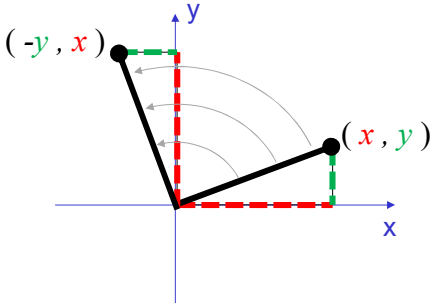
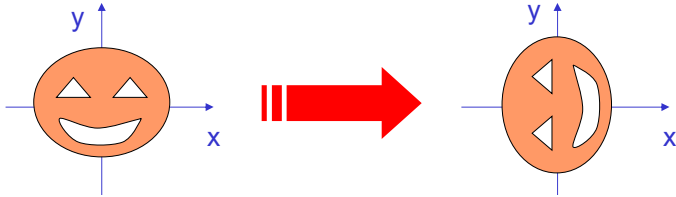
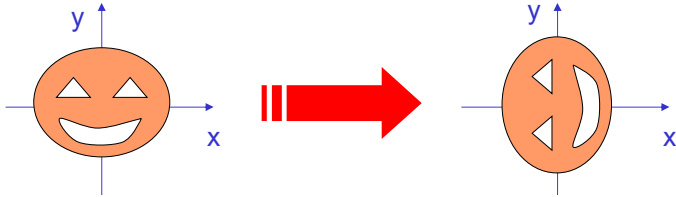
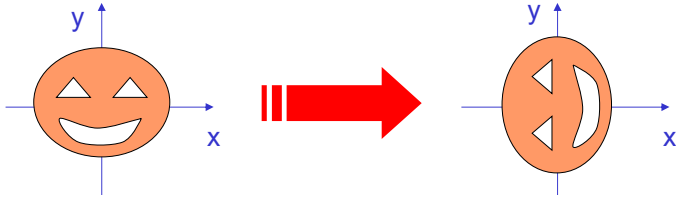
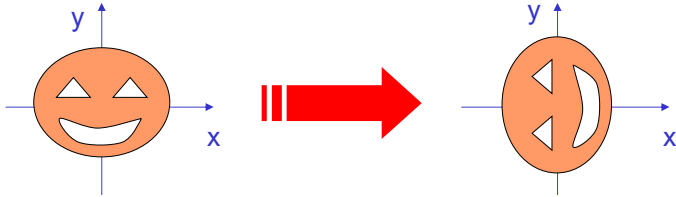
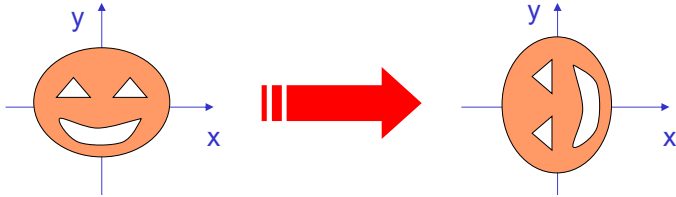
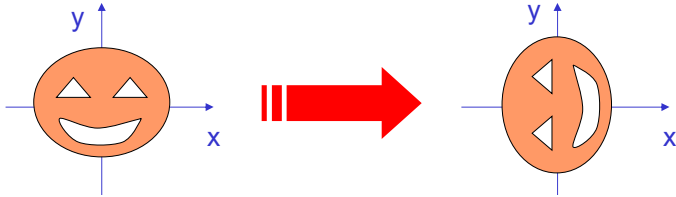
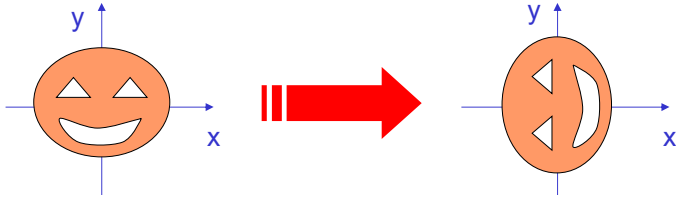
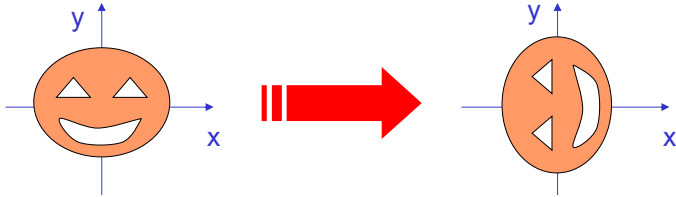
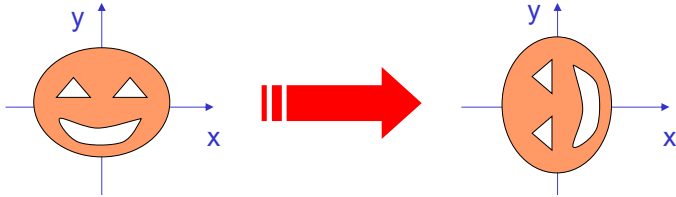
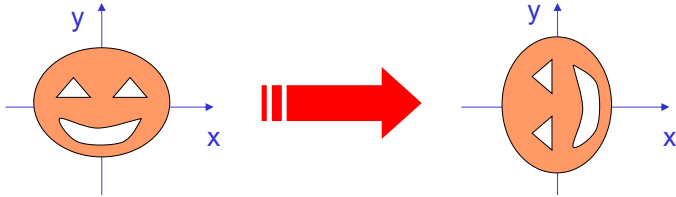
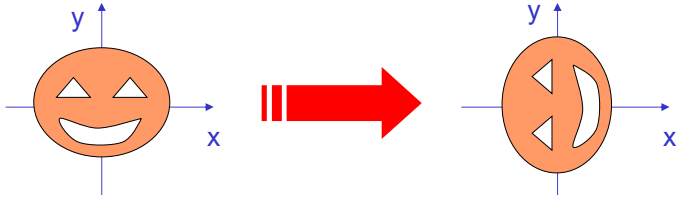
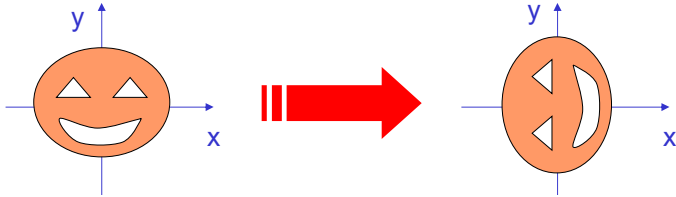
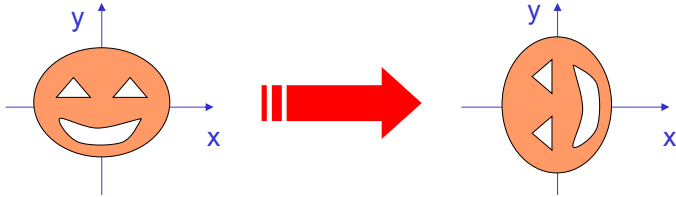
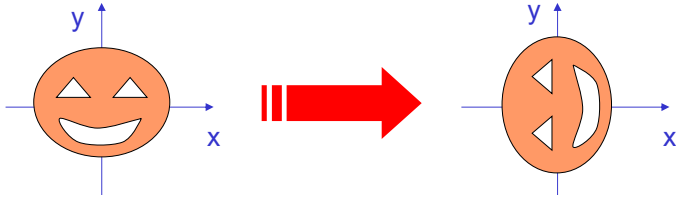
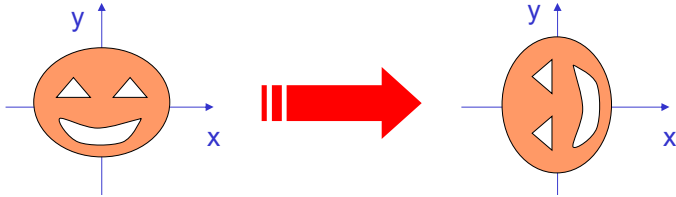
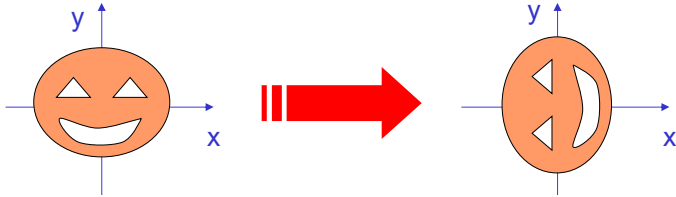
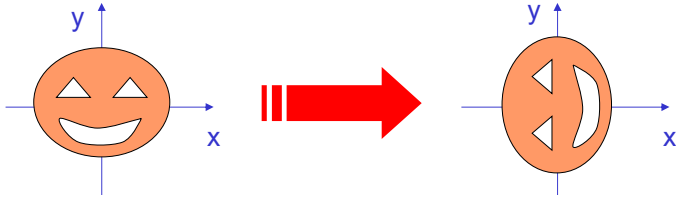
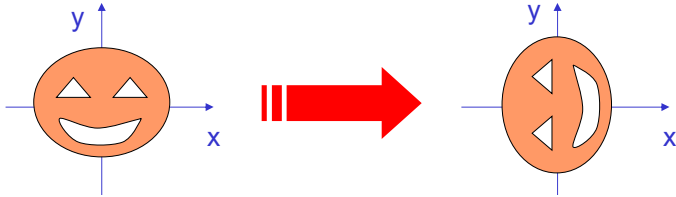
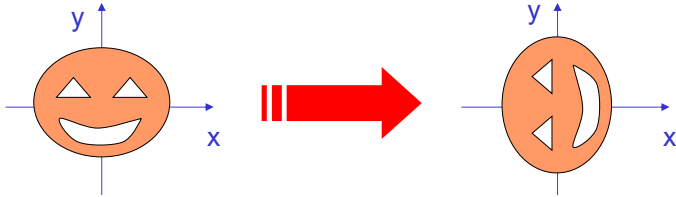
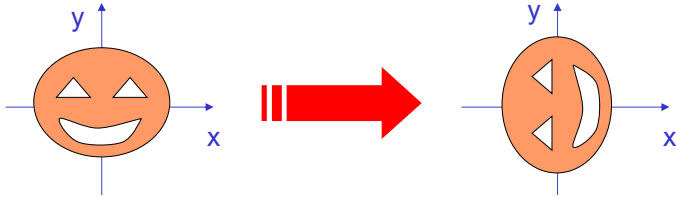
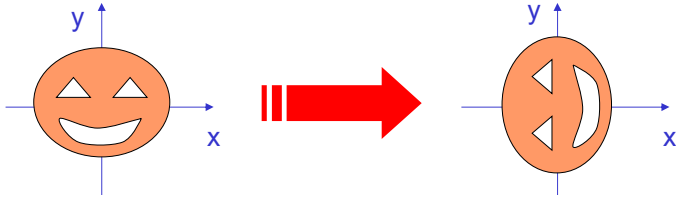
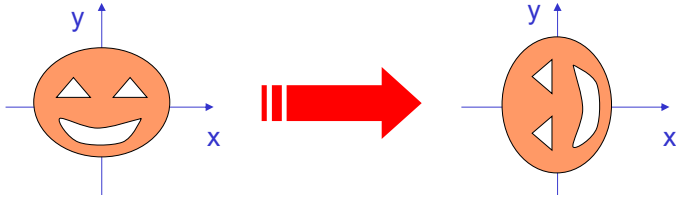
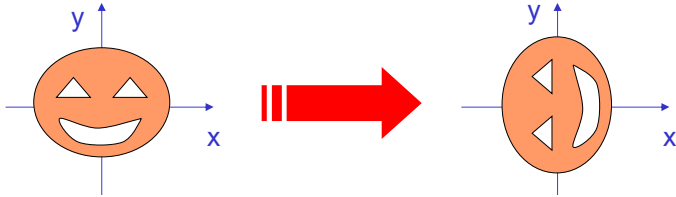
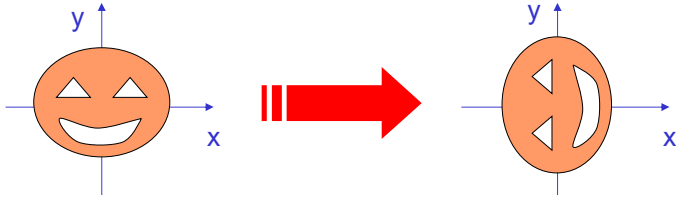
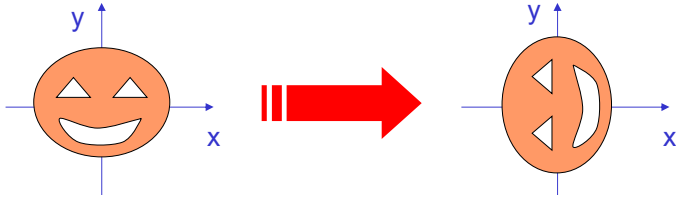
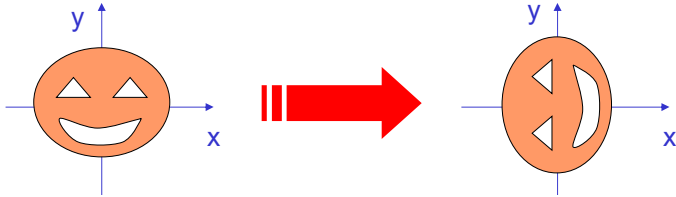
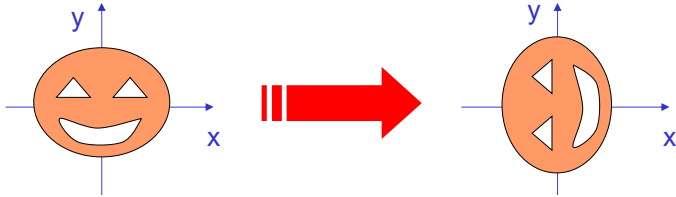
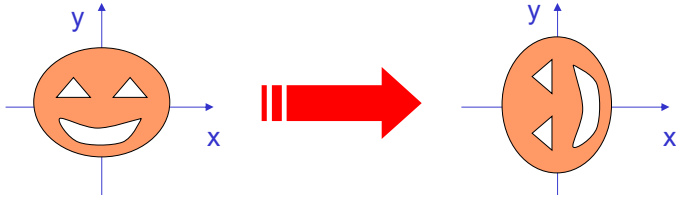
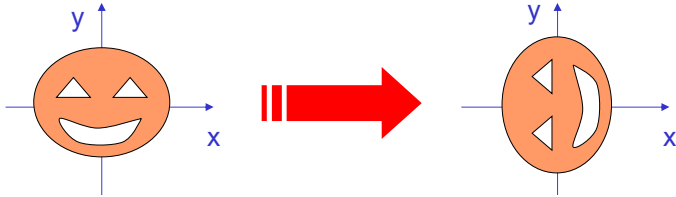
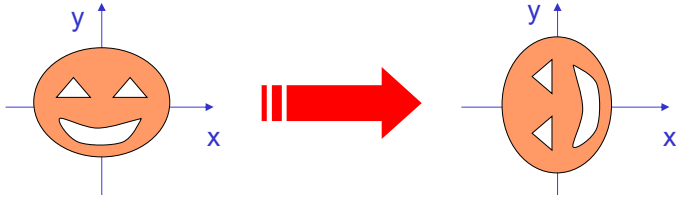
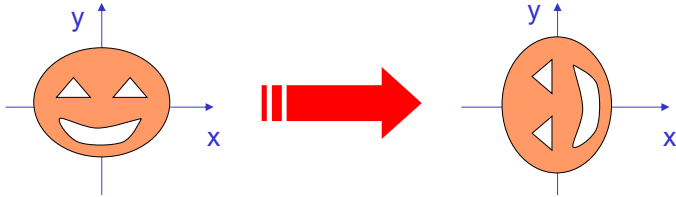
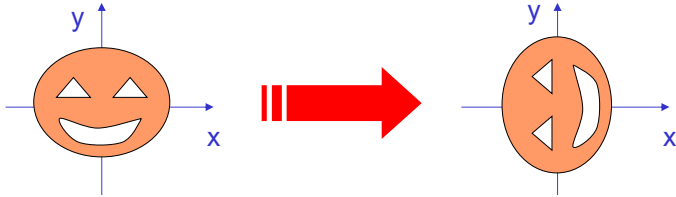
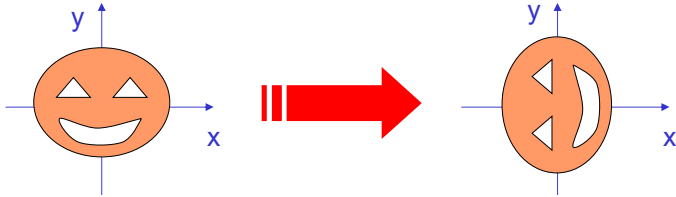
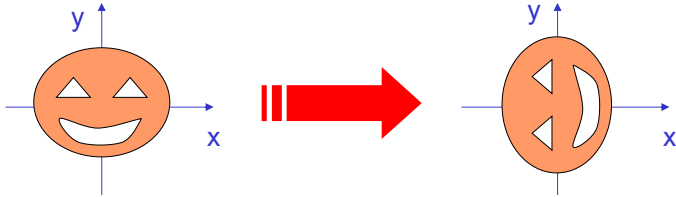
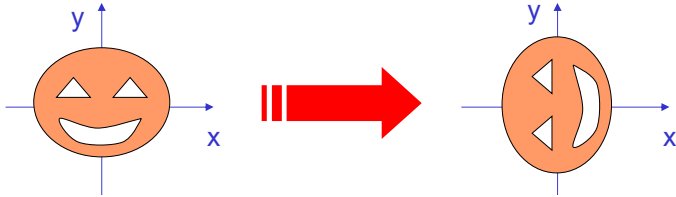
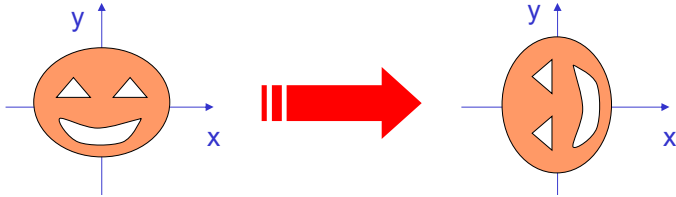
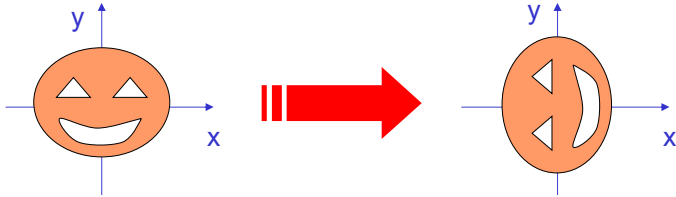
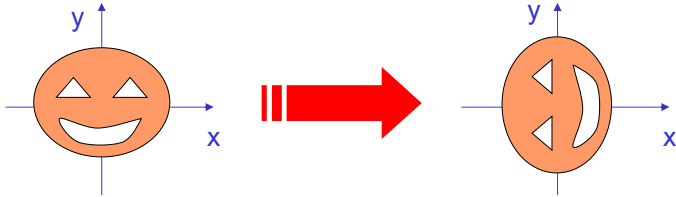
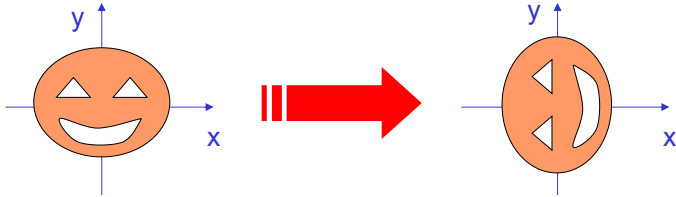
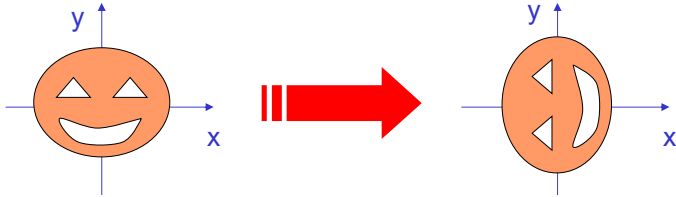
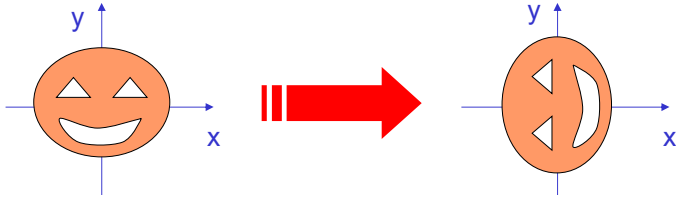
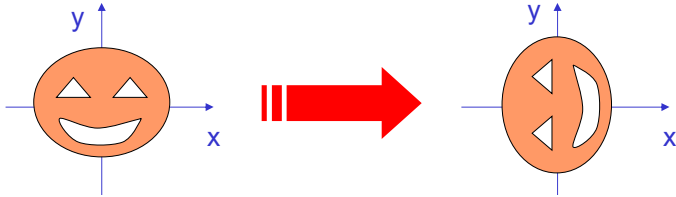
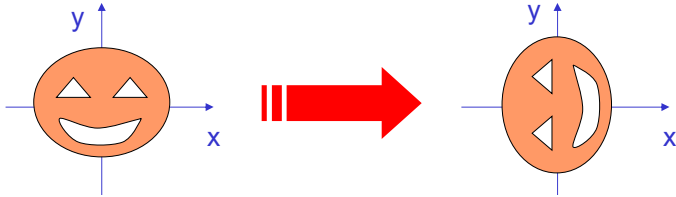
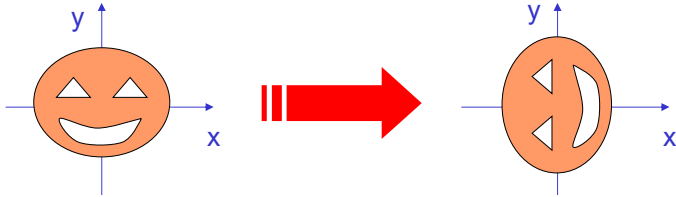
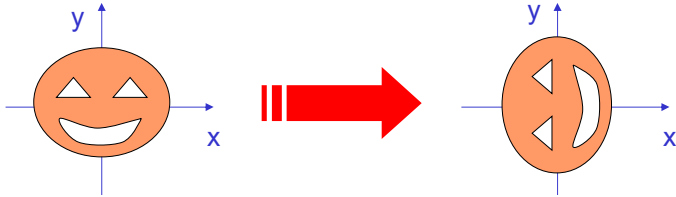
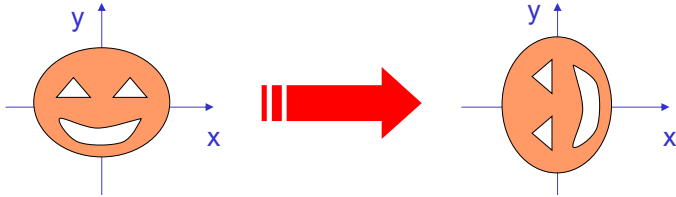
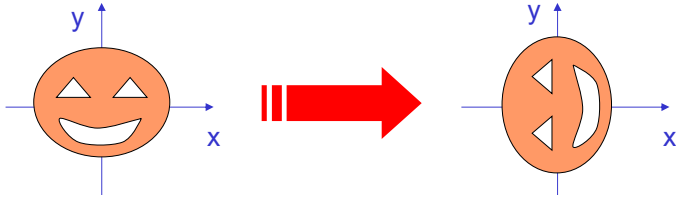
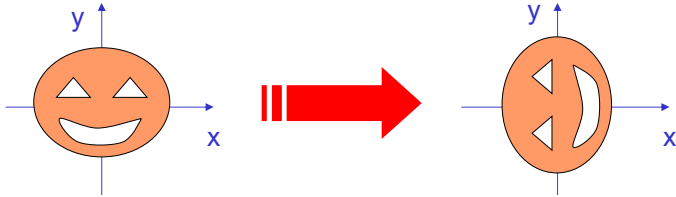
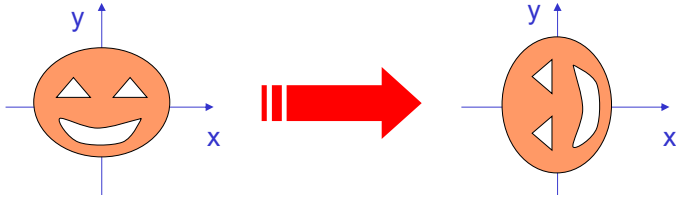
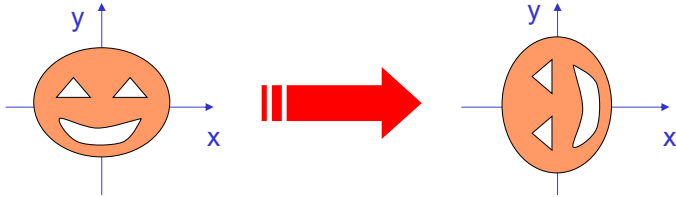
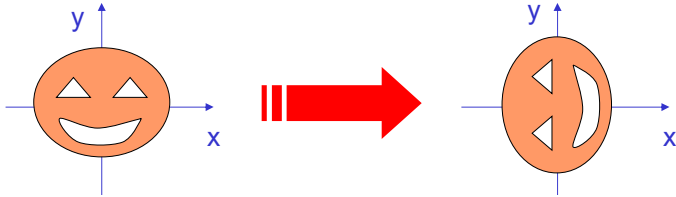
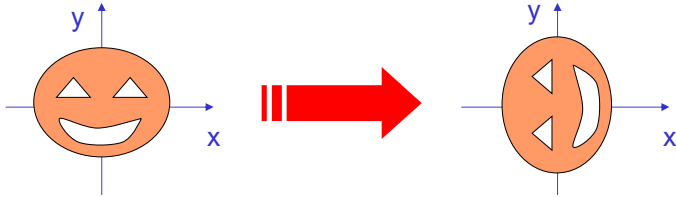
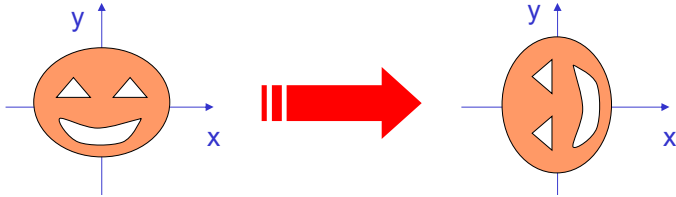
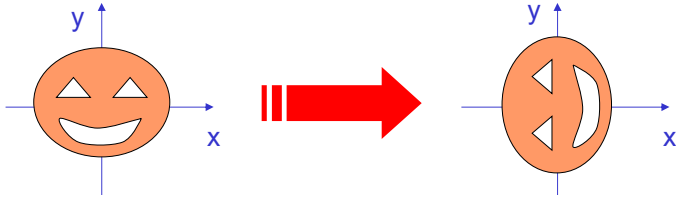
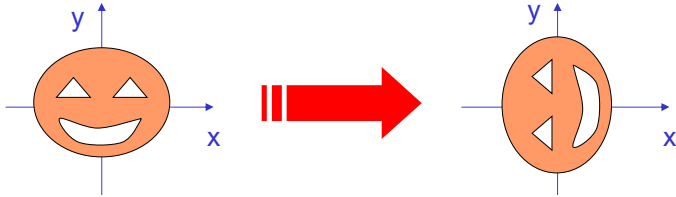
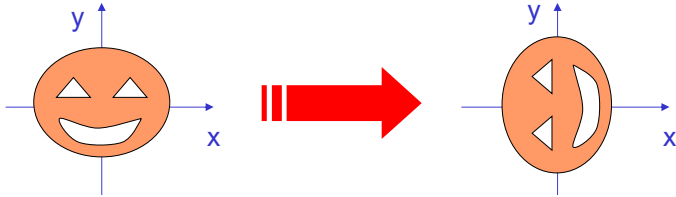
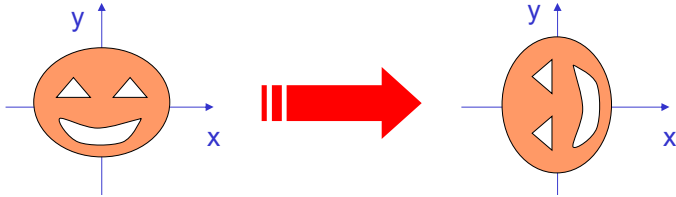
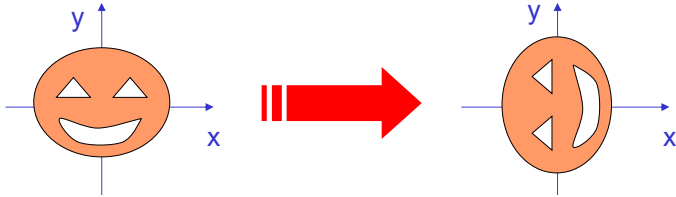
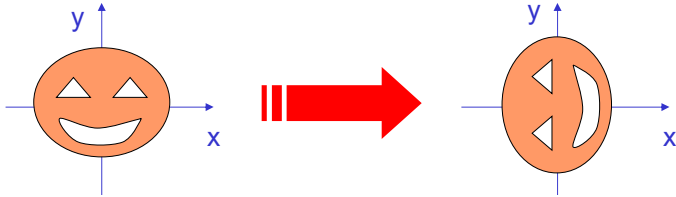
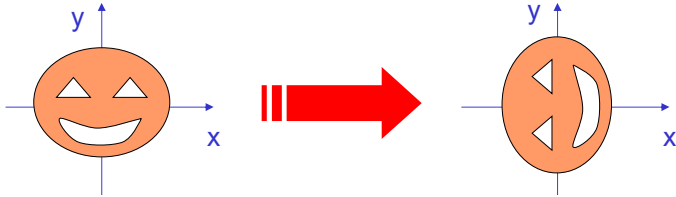
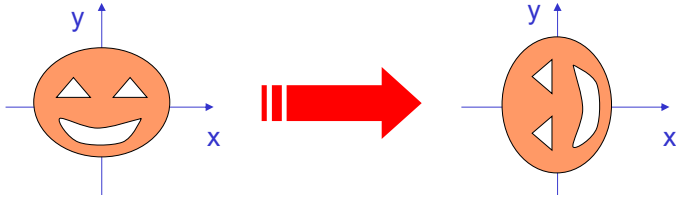
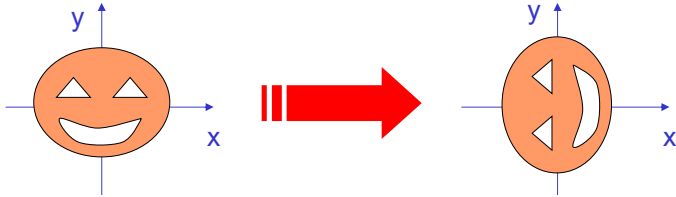
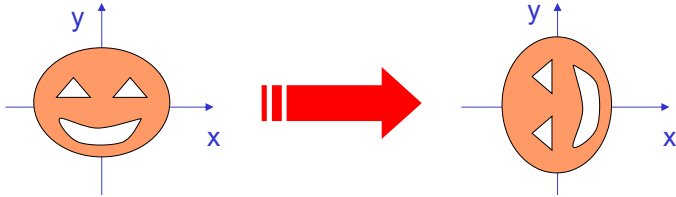
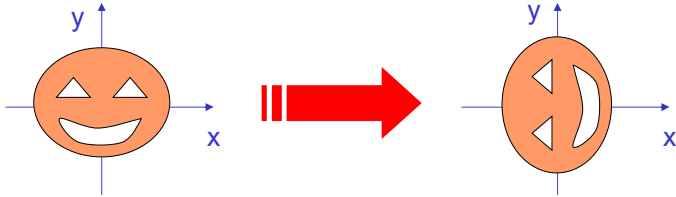
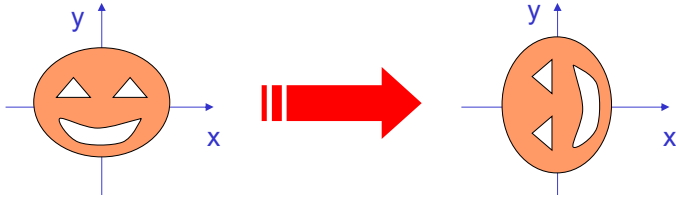
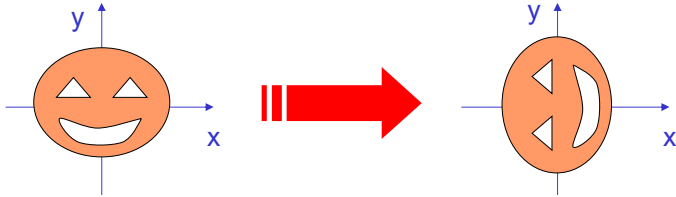
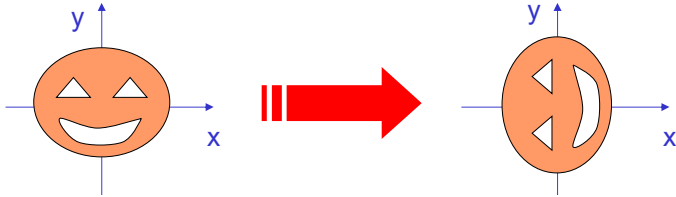
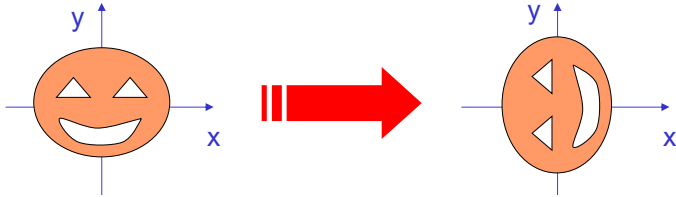
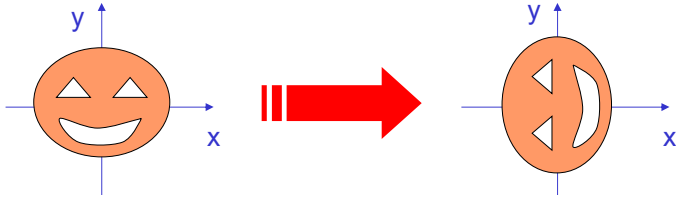
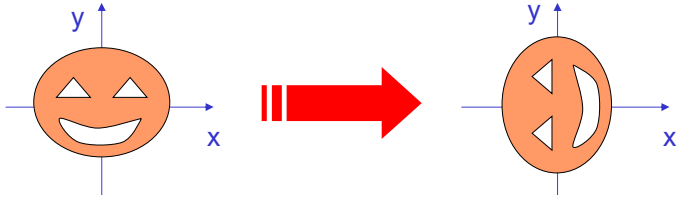
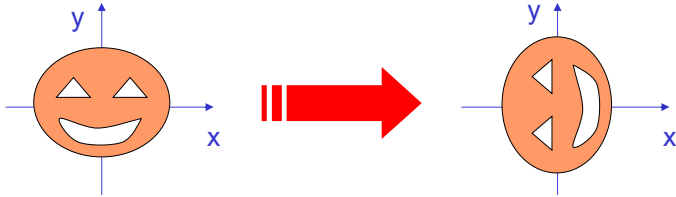
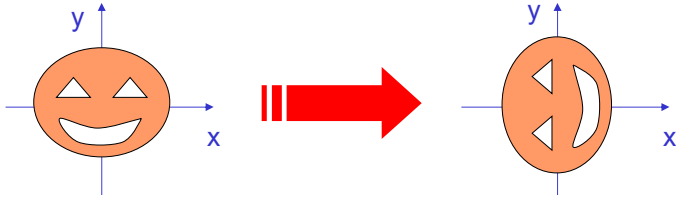
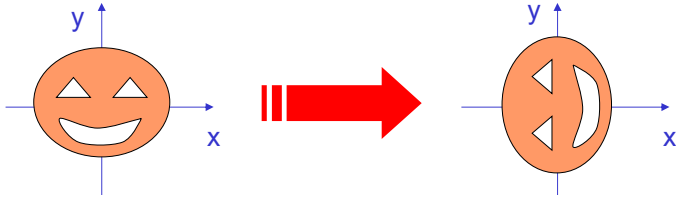
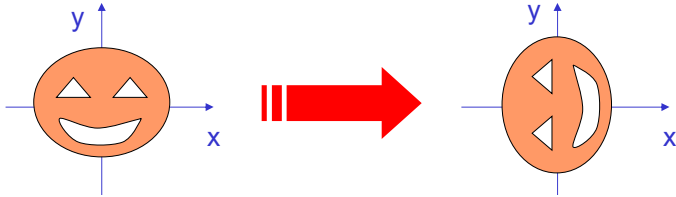
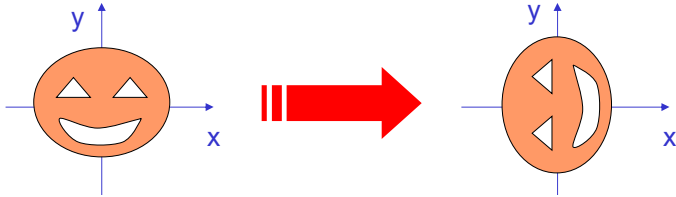
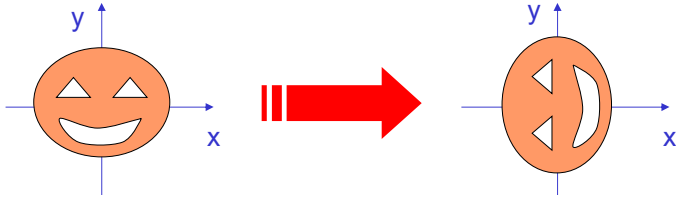
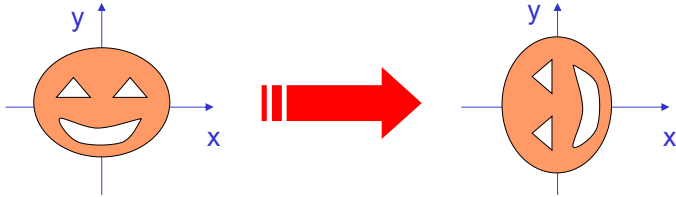
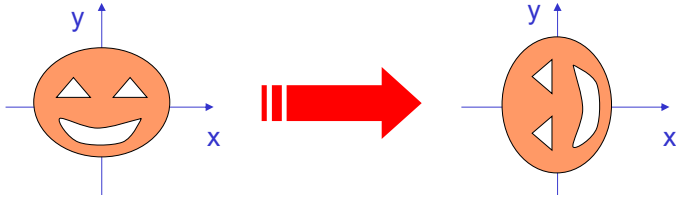
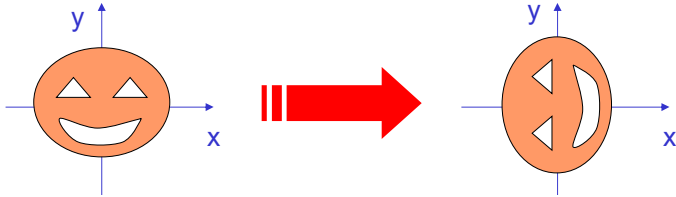
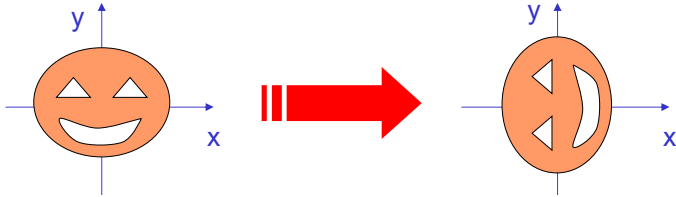
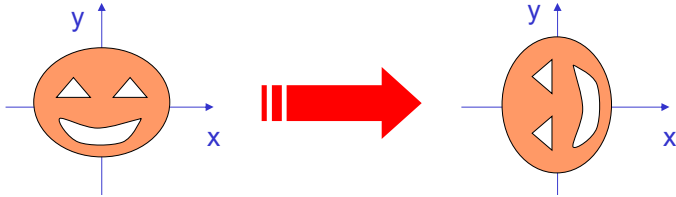
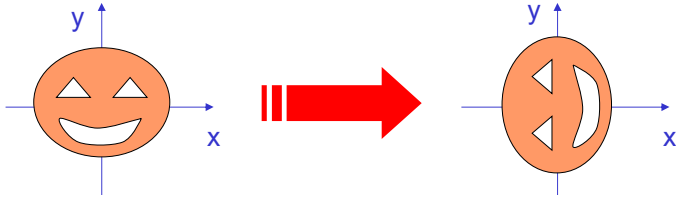
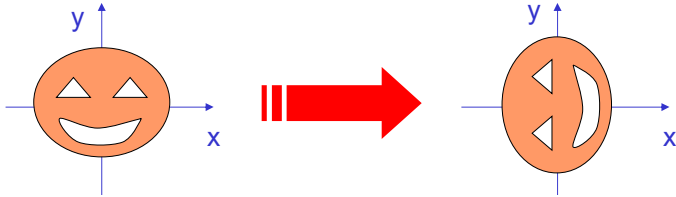
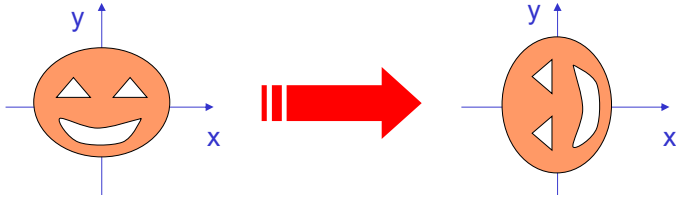
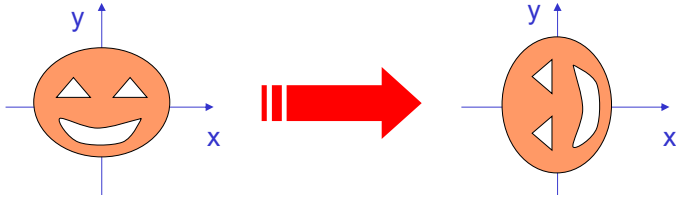
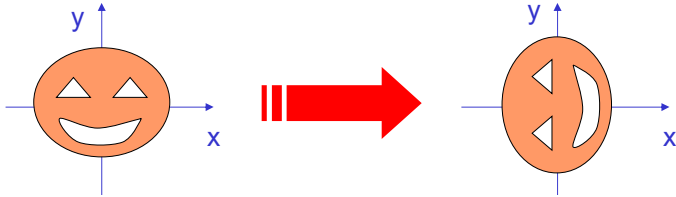
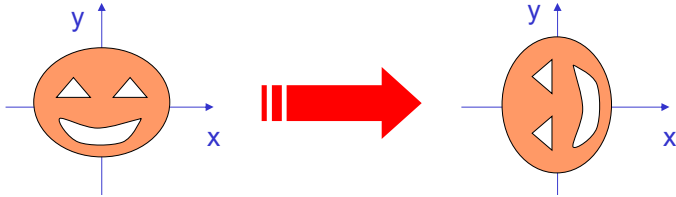
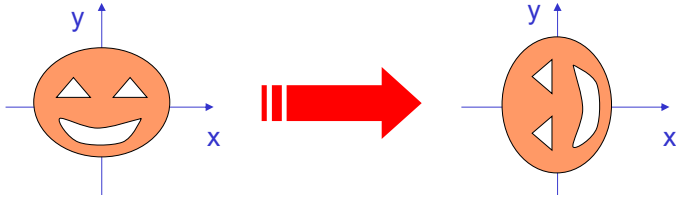
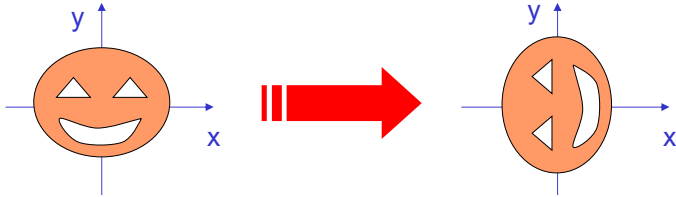
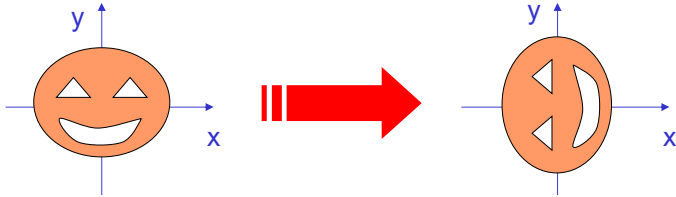
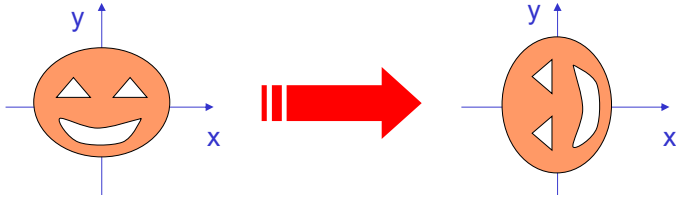
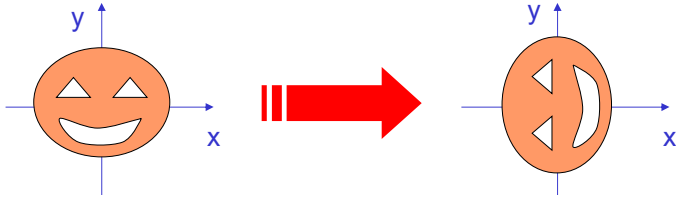
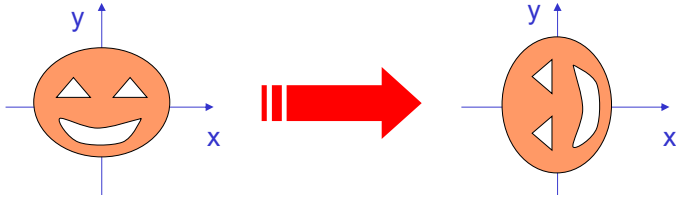
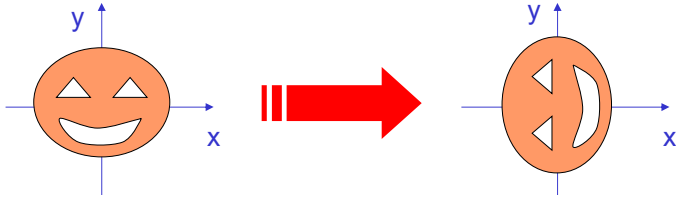
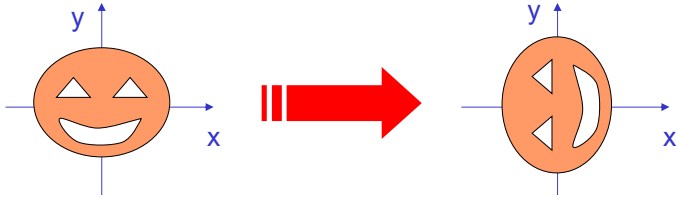
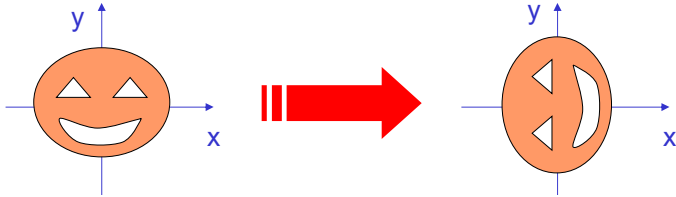
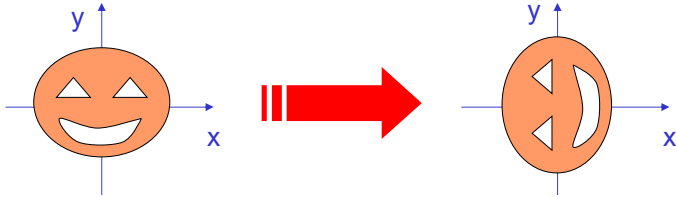
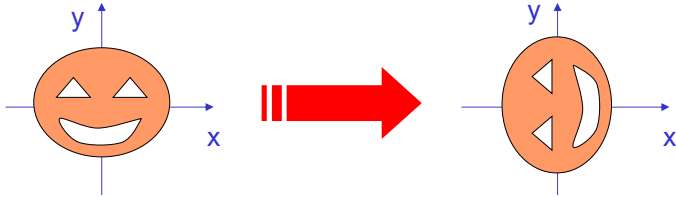
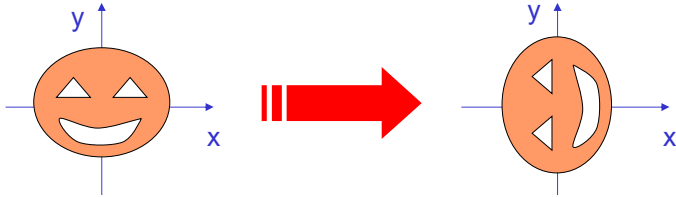
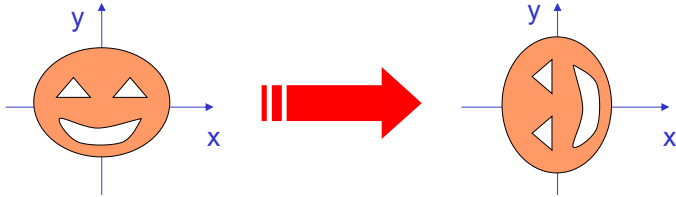
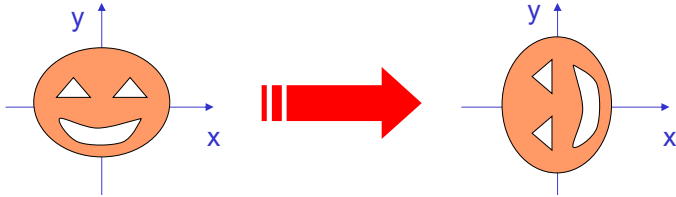
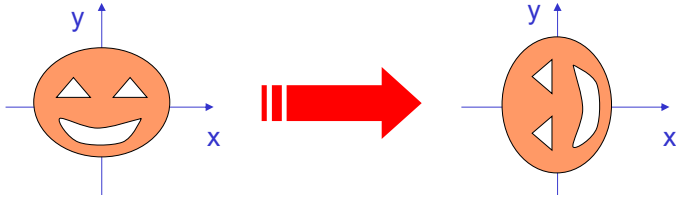
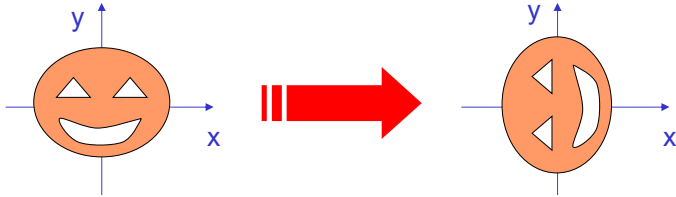
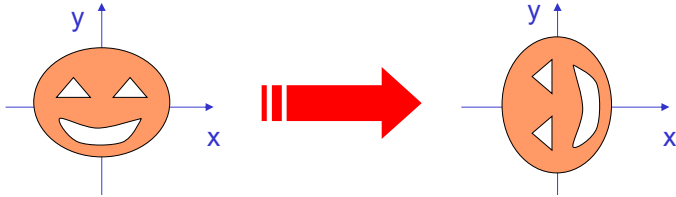
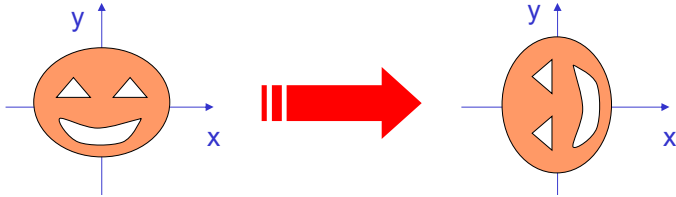
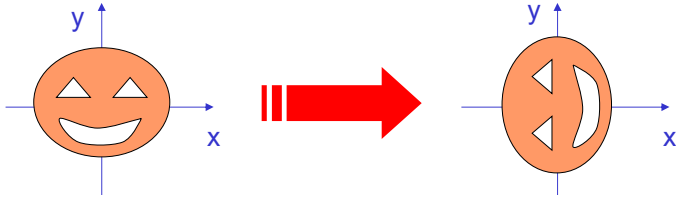
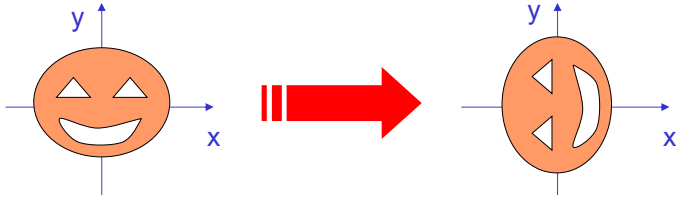
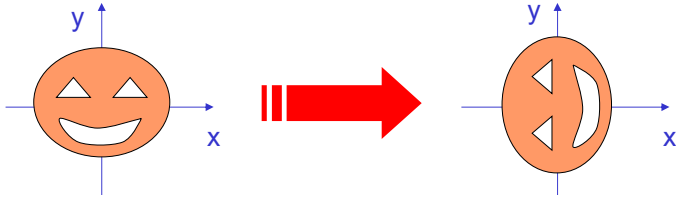
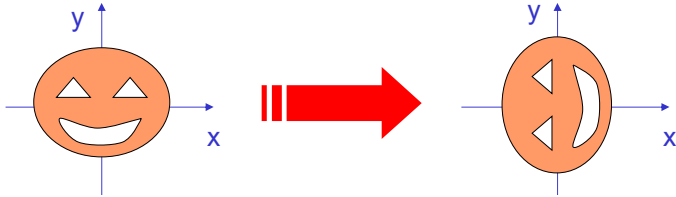
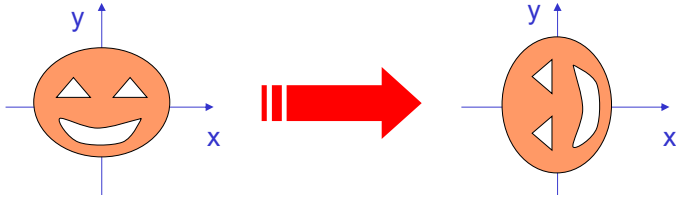
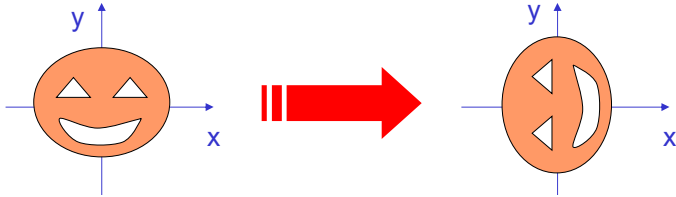
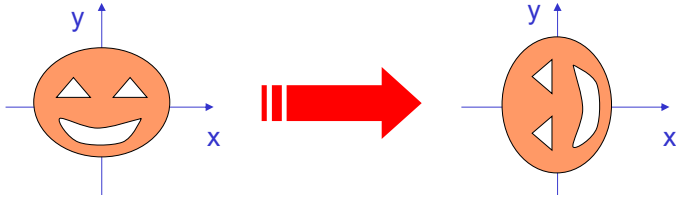
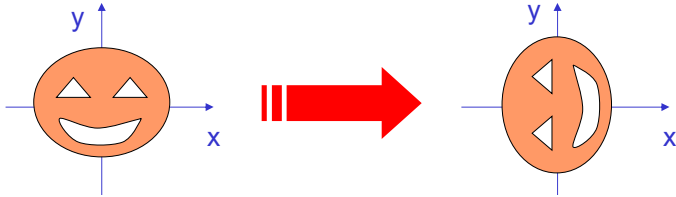
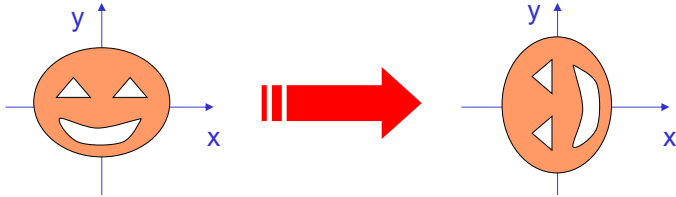
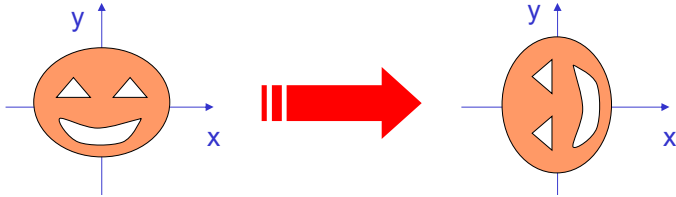
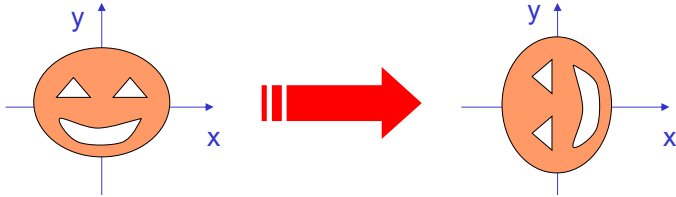
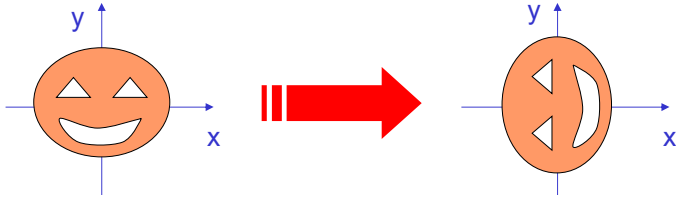
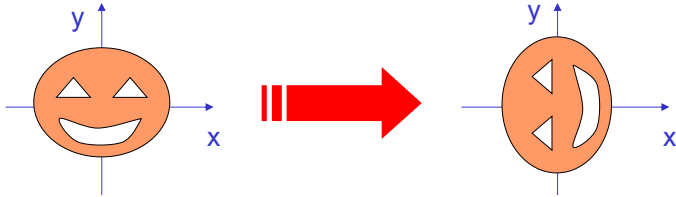
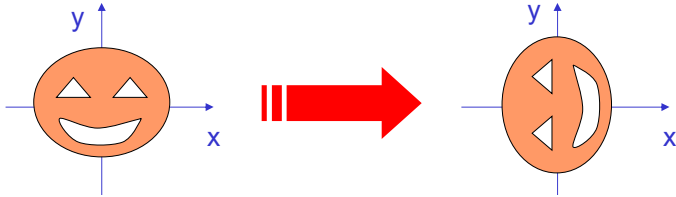
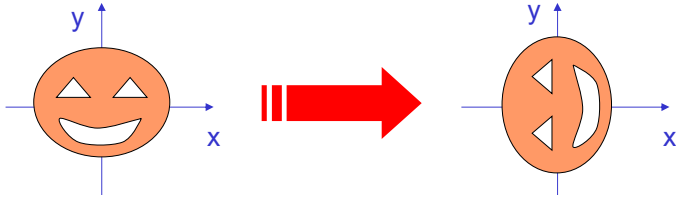
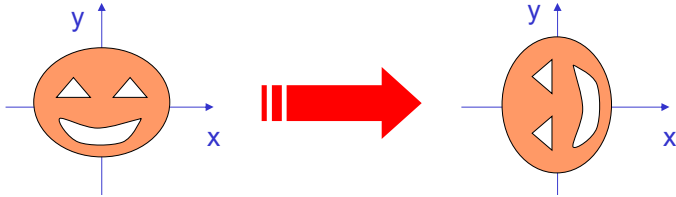
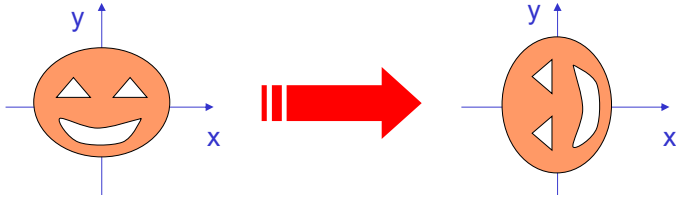
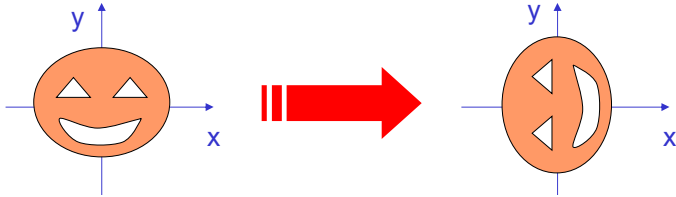
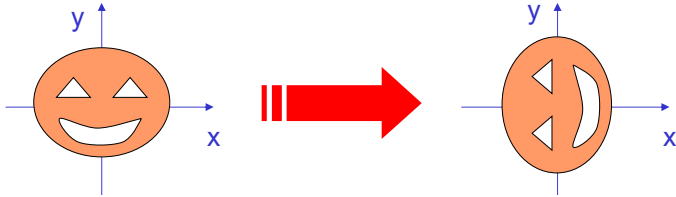
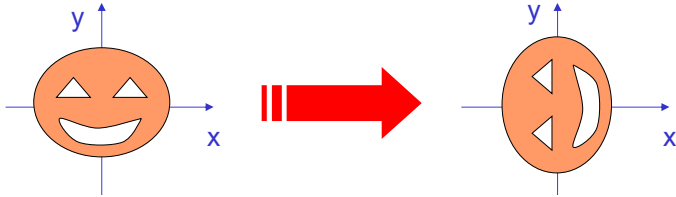
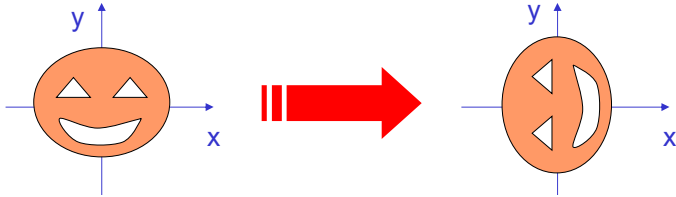
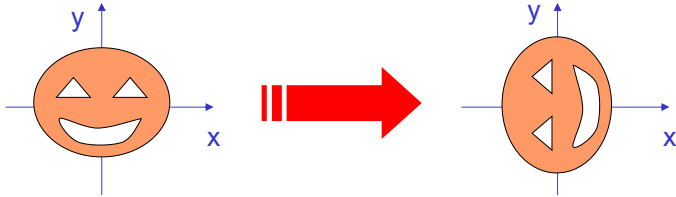
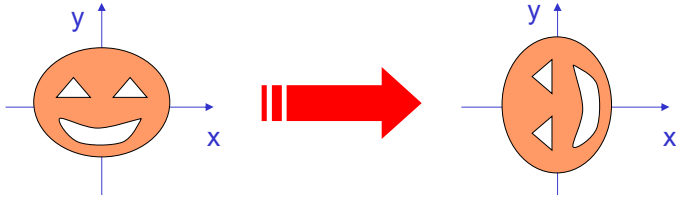
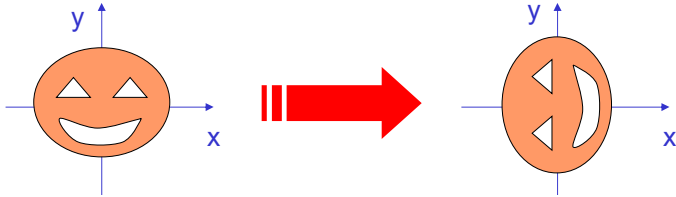
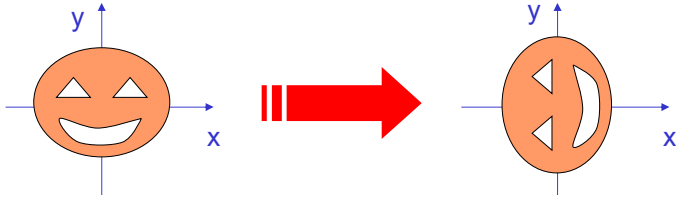
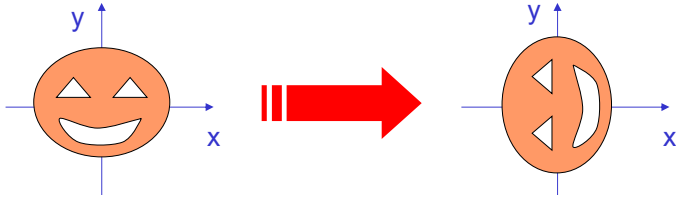
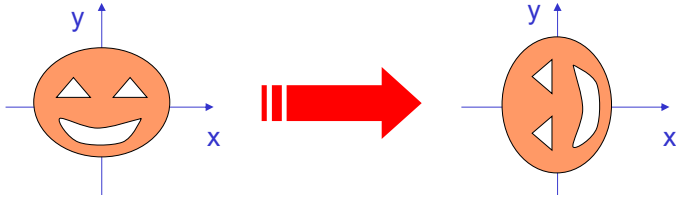
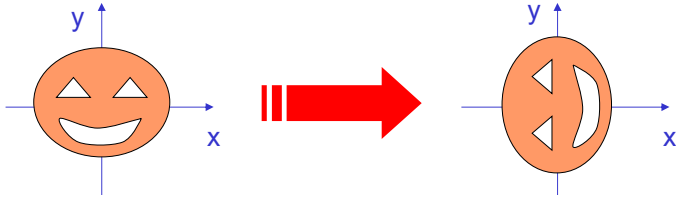
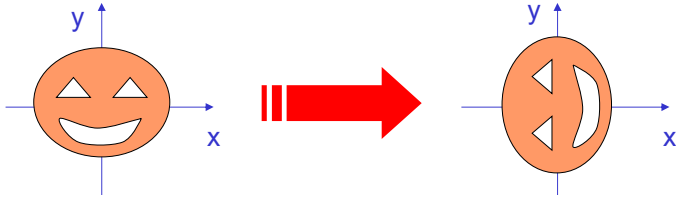
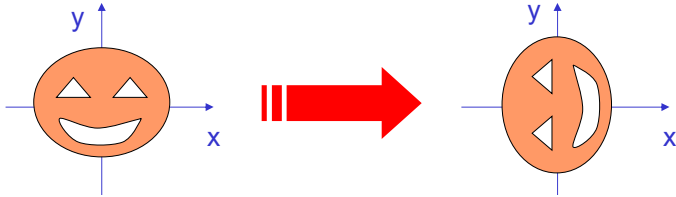
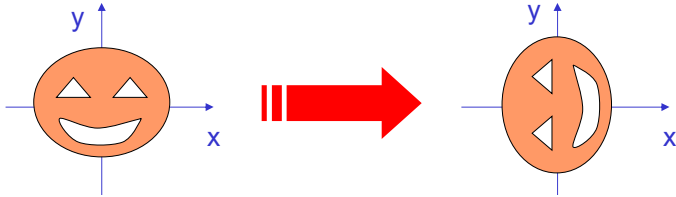
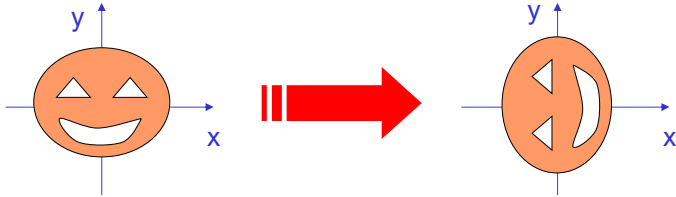
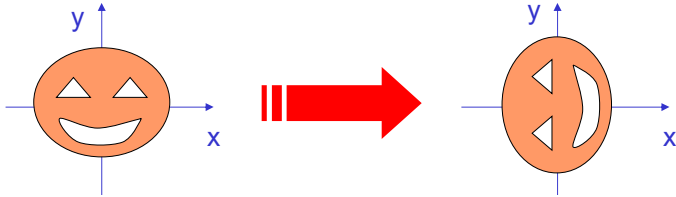
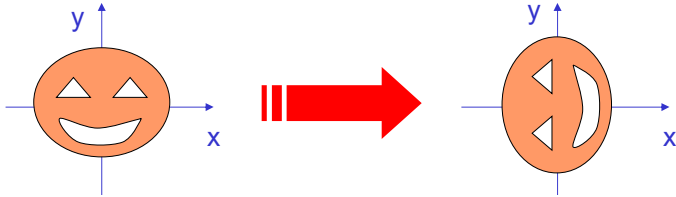
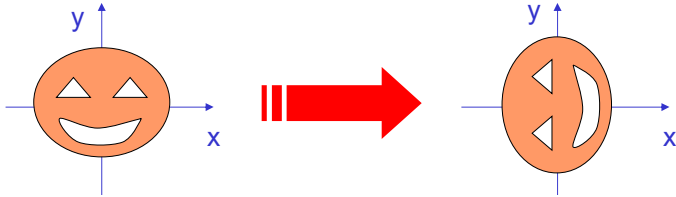
14

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


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


**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 rimangono invariati

19

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

Punti:

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

1 →

Vettori:

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

0 →



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



21

Breve ripasso: prodotto Matrice $\times$ Vettore
(«riga per colonna»)

✓ Posso scriverlo
come 4 prodotti dot
con i vettori riga

A →

B →

C →

D →

$\cdot$

$=$

A →

B →

C →

D →

$\cdot$

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 $\mathbf{M}$
 - ⇒ $\mathbf{M}$ è detta la «matrice di trasformazione»
 - ⇒ Questo varrà anche per tutte le altre trasformazioni che vedremo
 - ⇒ Una trasformazione esprimibile in questo modo è detta «affine»
 - ⇒ In CG, «matrice 4x4» è in pratica sinonimo di «trasformazione spaziale»



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

Trasformazioni Affini

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

per i vettori,
conta solo questo

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

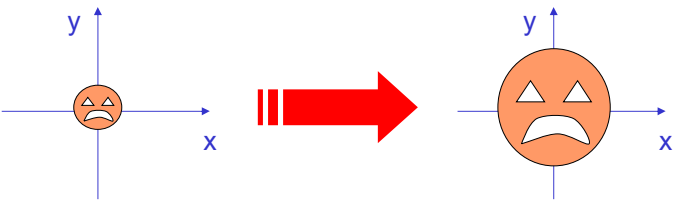
sempre

vettore di partenza
in coordinate affini

vettore di arrivo
in coordinate affini

27

Matrice di trasformazione:
scaling isotropico



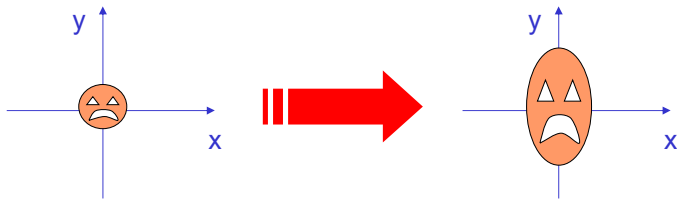
$$f \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \gamma x \\ \gamma y \\ \gamma z \\ 1 \end{pmatrix}$$
$$\begin{bmatrix} \gamma & 0 & 0 & 0 \\ 0 & \gamma & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} \gamma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

S_γ
matrice di scaling
isotropico

Cosa succede se la applico S_γ ad un vettore?

28

Matrice di trasformazione: scaling anisotropico



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

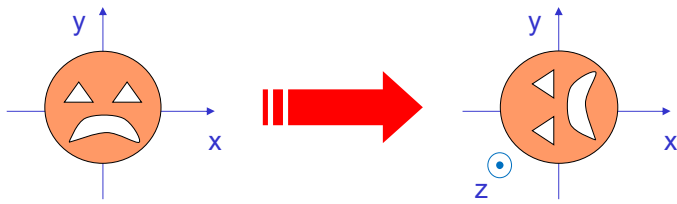
$$\begin{bmatrix} \gamma_x & 0 & 0 & 0 \\ 0 & \gamma_y & 0 & 0 \\ 0 & 0 & \gamma_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} \gamma_x x \\ \gamma_y y \\ \gamma_z z \\ 1 \end{bmatrix}$$

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



30

Matrice di trasformazione: rotazione attorno all'asse delle Z



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

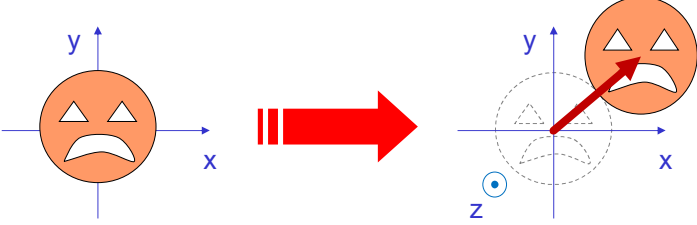
$$\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} -y \\ x \\ z \\ 1 \end{bmatrix}$$

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



31

Matrice di trasformazione:
traslazione



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

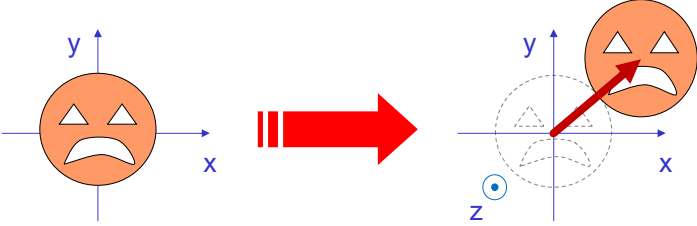
$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix}$$

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



33

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 $\mathbf{t} = (t_x, t_y, t_z)$



34

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



35

Trasformazioni Affini (o lineari)

- ✓ Le trasformazioni esprimibili come una moltiplicazione per una matrice M 4×4 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» (4×4) 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

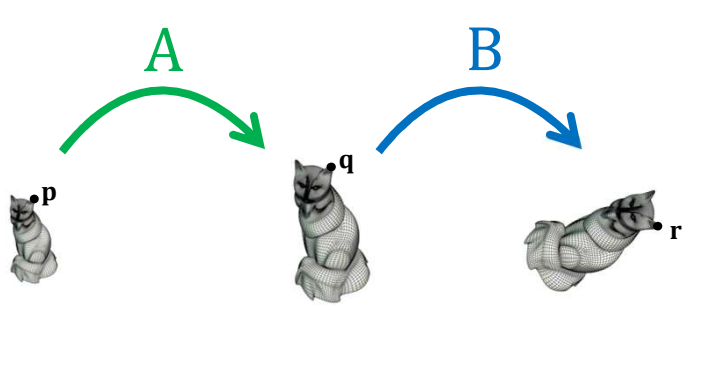
- ✓ Le caratteristiche spaziali di una trasformazione si riflettono nelle caratteristiche matematiche della matrice corrispondente.
- ✓ Ad esempio, il **determinante** della matrice ha una interpretazione geometrica:
è il fattore di scala dei volumi della trasformazione
 - ⇒ Se prendo un modello 3D di una bottiglia da 2L e gli applico una trasformazione M (cioè trasformo tutti i suoi vertici), la bottiglia risultante sarà avrà una capienza $2 \cdot \det(M)$ litri
 - ⇒ Se il determinante è negativo, l'oggetto si «ribalta» specularmente (le cose in *sensu orario* diventano antiorario, le scritte disegnate sopra si leggeranno *inverted*, etc
 - ⇒ Esercizio: verificare, trovando il determinante delle matrici viste fin'ora
- ✓ Il vantaggio principale di questa rappresentazione delle trasformazioni spaziali consiste nella loro composizione
 - ⇒ Vediamo in che modo

37

✓ Cioè: due (oppure n) trasformazioni al prezzo di una!
 ⇒ Moltiplicando per la matrice $M_C = M_B \cdot M_A$, da sola, ottengo l'effetto che otterrei trasformando con M_A e M_B in cascata

38

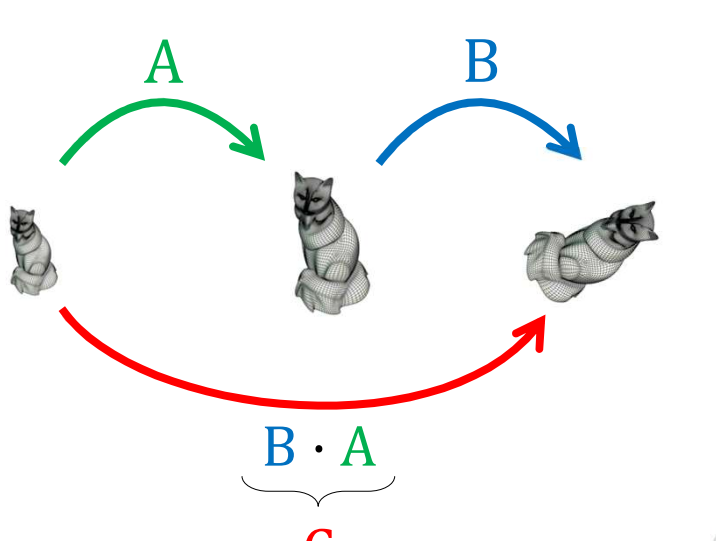
Composizione di trasformazioni affini



$\mathbf{q} = \mathbf{A} \cdot \mathbf{p}$
 $\mathbf{r} = \mathbf{B} \cdot \mathbf{q} = \mathbf{B} \cdot (\mathbf{A} \cdot \mathbf{p}) = \underbrace{(\mathbf{B} \cdot \mathbf{A})}_{\mathbf{C}} \mathbf{p}$

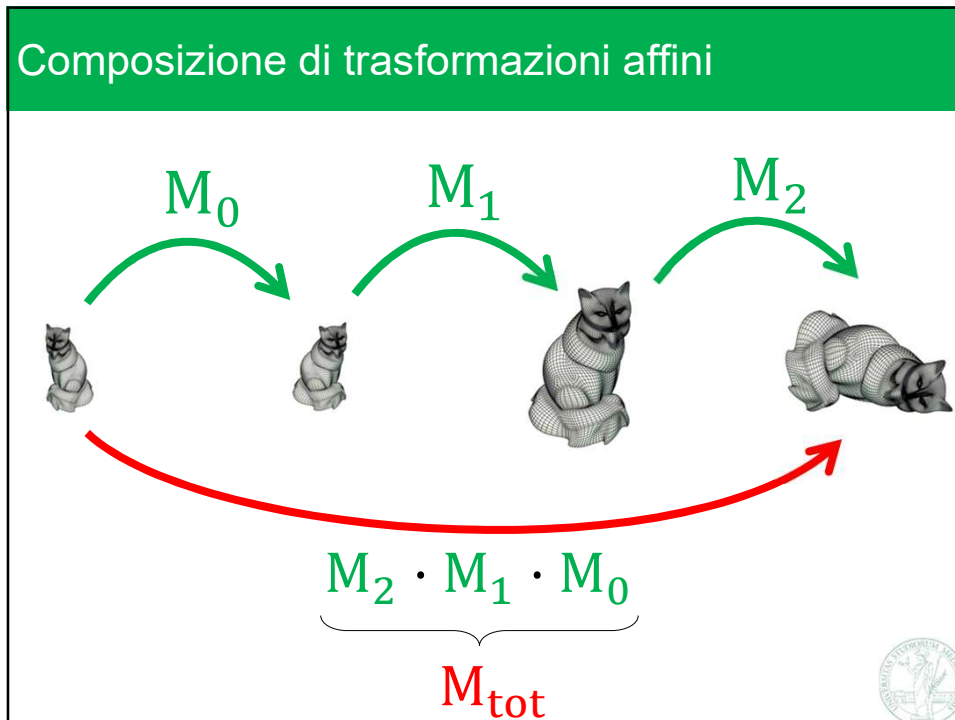
39

Composizione di trasformazioni affini



$\mathbf{B} \cdot \mathbf{A}$
 $\mathbf{C}$

40



41

Ripasso: moltiplicazione fra matrici (cioè composizione di trasformazioni)

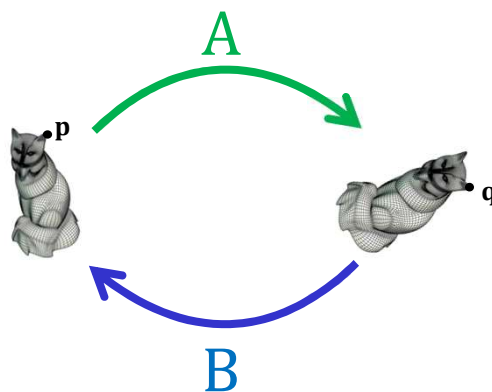
- ✓ Associativa: $A(BC) = (AB)C$
- ✓ Ma non commutativa! $AB \neq BA$

⇒ previsione:
determinare il corretto ordine delle trasformazioni non sarà intuitivo

The diagrams illustrate the non-commutativity of transformations. The left diagram shows a translation (RT) followed by a rotation. The right diagram shows a rotation (TR) followed by a translation.

42

Inversa di una trasformazione



$$B = A^{-1}$$

$$q = A \cdot p$$

$$p = B \cdot q$$



44

Calcolo dell'inversa di una matrice di trasformazione affine

- ✓ Potremmo applicare un algoritmo generico di inversione di matrici 4×4
- ✓ molte delle matrici di trasformazioni utili sono però casi particolari che ci consentono di ottenere in modo semplificato (ed efficiente) l'inversa
- ✓ Vediamo per esempio l'inversa delle matrici di: traslazione, scalatura, rotazione, e di composizione di matrici



45

Inversa della matrice di scalatura

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

l'inversa è, logicamente:

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



46

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:

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



47

Inversa di matrice di simmetria speculare

matrice di
rotaz di 90°
in senso **antiorario**
attorno
all'asse delle z:

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

l'inversa è, logicamente...

matrice di
rotaz di 90°
in senso **orario**
attorno
all'asse delle z:

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

Nota: la matrice inversa è anche la trasposta.

Come vedremo, questo varrà per tutte le matrici di rotazione.

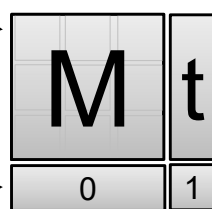


48

La matrice 4x4 corrispondente ad una trasformazione affine: sommario

Questa
sottomatrice **M** 3x3
si applica alle coord
cartesiane sia dei
punti che dei **vettori**

Ultima riga: (0,0,0,1)
Così che i **punti**
vengano mappati
sempre in **punti**,
e i **vettori** sempre
in **vettori**



Matrice 4x4

Questo vettore **t**
si somma al risultato
quando trasformato
punti
ma viene annullato
(moltiplicato per zero)
quando trasformato
vettori.



49

Alcune caratteristiche numeriche delle matrici 4x4 e loro interpretazione geometrica

- ✓ Matrice inversa = trasformazione inversa
 - ⇒ se esiste (se $\det \neq 0$)
- ✓ Determinante della matrice 4x4 =
(che è anche il determinante della sottomatrice 3x3)
 - ⇒ Moltiplicatore del volume
 - ⇒ Per es, determinante = 0.5: questa matrice riduce i volumi degli oggetti trasformati del 50%
- ✓ Deficit di rango della matrice:
 - ⇒ cioè rango massimo (4) meno rango della matrice
 - ⇒ È la perdita di dimensionalità
 - ⇒ Per es: rango 3 (invece di 4): la trasf. riduce oggetti 3D a oggetti piatti.
 - ⇒ Per es: rango 2 (invece di 4): la trasf. riduce oggetti 3D ad una linea
 - ⇒ Per es: rango 1 (invece di 4): la trasf. riduce oggetti 3D ad un punto
 - ⇒ Se rango < 4: $\det = 0$ (il volume scompare) e la matrice non è invertibile (è «singolare»)



50

Esercizio

- ✓ Comporre (moltiplicare, nell'ordine giusto) una matrice di rotazione di 90 gradi attorno all'asse z con una matrice di traslazione del vettore (1,2,3) per ottenere la matrice che "ruota-poi-trasla"
 - ⇒ Hai trovato così una matrice di "roto-traslazione"
- ✓ Trovare poi la matrice che ottegni se invece *traslo poi ruota*
 - ⇒ È la stessa matrice? In cosa differisce dalla prima?



51