



# Animazione Digitale Morphing

Prof. N. Alberto Borghese



## Sommario

**Il morphing.**

Feature Based morphing.

Grid Based morphing.

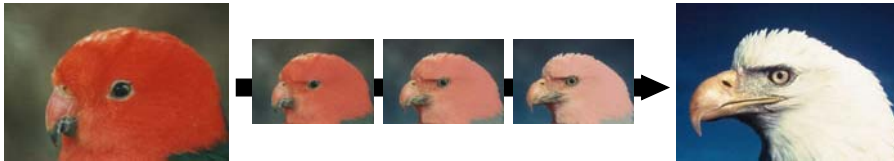


## Cosa fa il morphing?



## Morphing

Trasformazione di un'immagine in un'altra con continuità attraverso immagini intermedie.



- **Criteri da considerare:**
  - velocità di elaborazione;
  - facilità d'uso.
  - Qualità del risultato.





## Morphing: origine

- La crescita dei peli sul corpo dell' "Uomo-lupo".
- Dr. Jackill and Mr. Hide.

Ma anche:

- Willow (1988) introdusse Grid Morphing (Smythe e Wolberg, 1990).
- Indiana Jones e l'Ultima Crociata (1990, per movimento).
- Micheal Jackson: Black or White introdusse il feature based morphing (**Beier e Neely, 1992**).
- Terminator 2 (1991).

Negli anni 70 le immagini intermedie venivano generate mediante ritocco fotografico e fotografate in sequenza (stop motion).





## Cross-dissolvenza

Date 2 immagini, vengono generate immagini "intermedi", in cui il colore dei pixel e' una media pesata dei colori delle due immagini:

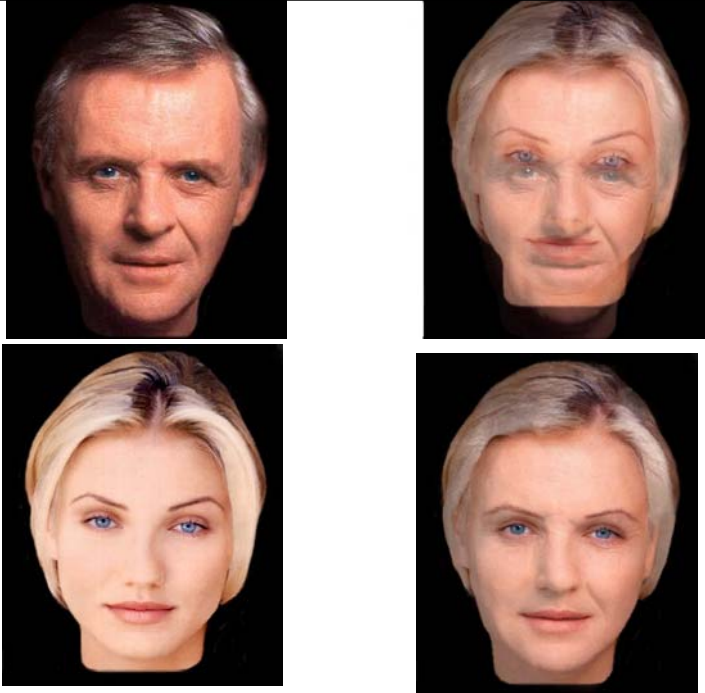
$$\begin{array}{l} I_1 = [R_1, G_1, B_1] \\ I_2 = [R_2, G_2, B_2] \end{array} \quad \rightarrow \quad I_m = [R_m, G_m, B_m]$$

$$\begin{aligned} R_m &= \alpha R_1 + (1 - \alpha) R_2 \\ G_m &= \alpha G_1 + (1 - \alpha) G_2 \\ B_m &= \alpha B_1 + (1 - \alpha) B_2 \end{aligned}$$



$\alpha$  pesa il relativo contributo delle due immagini all'immagine destinazione.

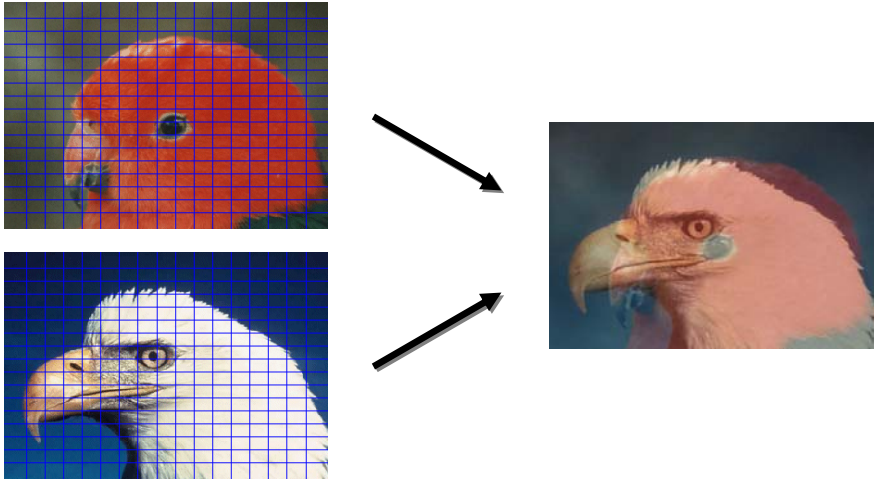
## Effetti e problemi



A.A. 2004-2005 7/47 http://homes.dsi.unimi.it/~borghese/

## Cross-dissolvenza



I punti di un'immagine intermedia saranno costituiti da una media pesata di punti non corrispondenti sulle due immagini sorgenti (fade-in / fade-out).

A.A. 2004-2005 8/47 http://homes.dsi.unimi.it/~borghese/



## Morphing



Opera su una coppia di immagini.

Non è solo cross-dissolvenza.

**Image warping (deformazione dell'immagine) +  
Cross-dissolvenza tra gli elementi dell'immagine.**

Viene definita una trasformazione che deforma la prima immagine, trasformandola nella seconda in modo continuo. La trasformazione inversa, opera dalla seconda immagine alla prima.

Le immagini intermedie della sequenza sono ottenute deformando la prima immagine in modo parziale.



## I due problemi del Morphing



- 1) Deformazione della prima immagine in modo che la **forma** degli oggetti contenuti si trasformi nella forma che gli oggetti hanno nella seconda immagine.
- 2) **Cross-dissolvenza** dei punti appartenenti ad oggetti che si corrispondono sulle due immagini.

La cross-dissolvenza è “semplice”.

Il problema grosso è definire una trasformazione che deformi la prima immagine nella seconda.

Non esistono strumenti automatici e l'intervento manuale deve essere minimo e “facile” per l'animatore.



## Sommario



Il morhing.

Field morphing.

Mesh morping.



## Field morphing



E' stato introdotto da Beier e Neely nel 1992 ed è chiamato anche feature-based morphing.

E' basato sulla definizione di primitive (punti o linee) di controllo.

Le linee di controllo devono corrispondersi nelle 2 immagini.

Viene definita una regione di influenza per ogni primitiva di controllo.



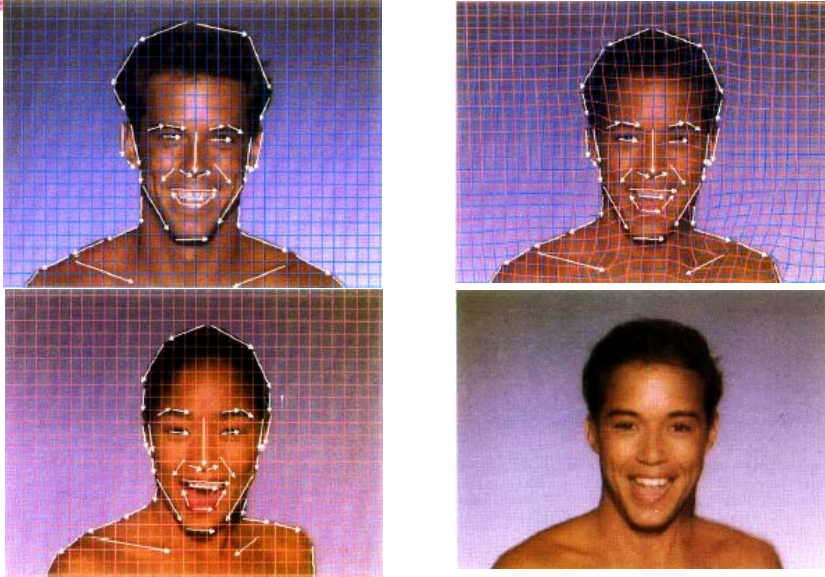
I punti all'interno della regione di controllo si spostano (deformando l'immagine) secondo una trasformazione geometrica.

I punti che cadono all'interno di più regioni di controllo, vengono deformati combinando le diverse trasformazioni geometriche.

Sarebbe bello potere identificare le feature in modo automatizzato.....



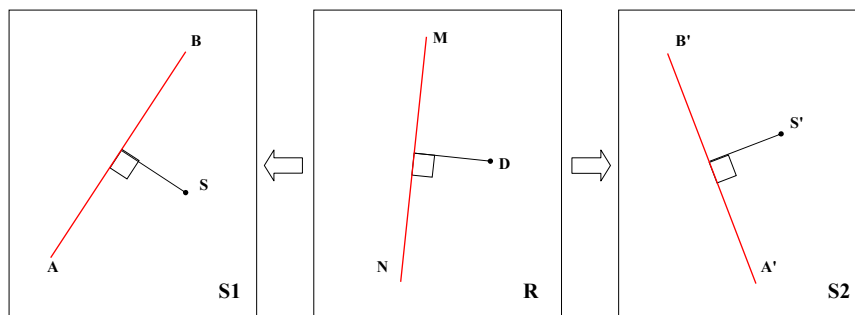
## Field morphing



Beier and Neely, Proc. Siggraph, 1992.



## Costruzione della deformazione



Identificata una coppia di linee (feature), viene determinata la **posizione**, **lunghezza** ed **orientamento** della linea nell'immagine intermedia.

I punti vicini (all'interno della **regione di influenza**) vengono descritti in un sistema di riferimento locale alla linea:  $D = D(A, B; S | A', B'; S')$ .

La trasformazione consisterà in una traslazione + scala.



## Vantaggi e problemi del field morphing



E' una tecnica espressiva ed intuitiva per l'animatore. Le feature sono dei tratti caratteristici nella coppia di immagine (e.g. occhi, bocca...).

Possibilità di seguire naturalmente linee curve di interesse.

### Problema computazionale

Per ogni pixel, occorre calcolare la quantità di spostamento per tutte le feature → Complessità computazionale notevole.

Possibili soluzioni:

Regioni di influenza di dimensioni limitate: questo può non consentire un corretto spostamento di tutti i pixel (e.g. nel caso di traslazione).

Regole diverse per ogni feature => complessità di definizione.



## Problemi del field morphing



### Ghost-busting

In alcune regioni si genera una deformazione inaspettata. Si manifesta con la comparsa di parti dell'immagine originale, in posizioni sbagliate dell'immagine intermedia (da cui ghost).

Soluzione:

Inserire o modificare features in queste regioni.

### Influence region

Legato al ghost-busting.

Come definisco le regioni di influenza?

Quale regione dovrà essere influenzata da una feature?

Quale metrica utilizzo (e.g. le labbra)?





## Sommario



Il morhing.

Field morphing.

**Mesh morping.**



## Mesh warping

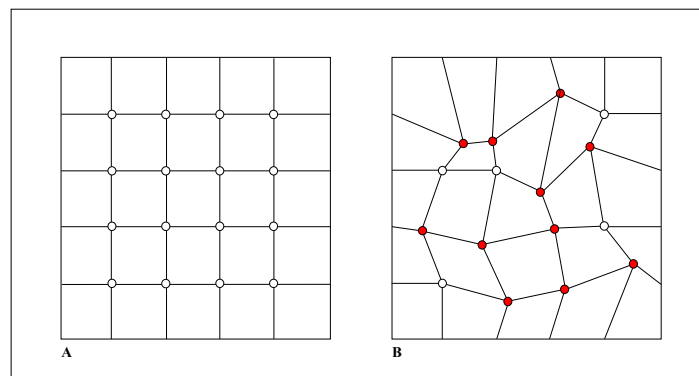


Immagine suddivisa in tante piccole aree rettangolari. Ciascun rettangolo viene trasformato in un quadrilatero. I punti all'interno di ciascuna area seguono la trasformazione dettata dall'area di appartenenza.



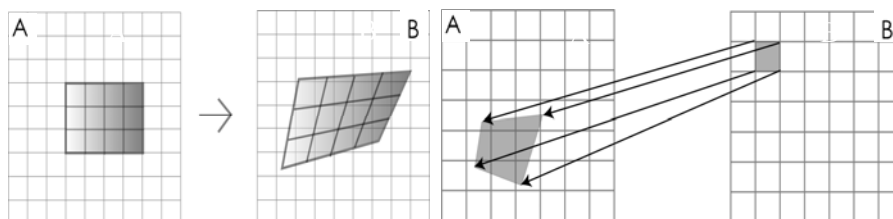
## Due approcci all'image warping.



- 1) Forward mapping. Ogni pixel dell'immagine sorgente viene copiato nella posizione corrispondente sull'immagine destinazione (intermedia).
- 2) Reverse mapping. Per ogni pixel dell'immagine destinazione, viene individuato il pixel corrispondente dell'immagine sorgente.

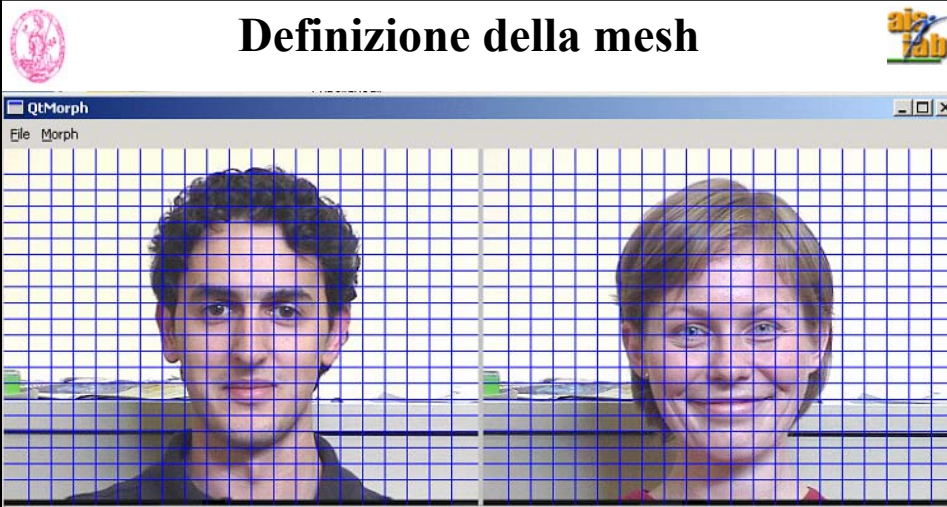


## Problemi nel warping



- La trasformazione (warping mediante mappatura diretta) di un pixel dell'immagine di partenza, A sull'immagine di destinazione B pone, in generale, tre problemi:
  - Il pixel cambia la sua forma da rettangolare a quadrilatero scaleno.
  - Il pixel di A trasformato non ricopre in generale un unico pixel di B: quale colore assegnerò ai pixel di B?
  - Esistono pixel di B che non vengono ricoperti da pixel di A.
- **Per risolvere questi problemi invertiamo il procedimento. Per ogni pixel dell'immagine di destinazione B, cerchiamo i corrispondenti pixel di A.**

## Definizione della mesh

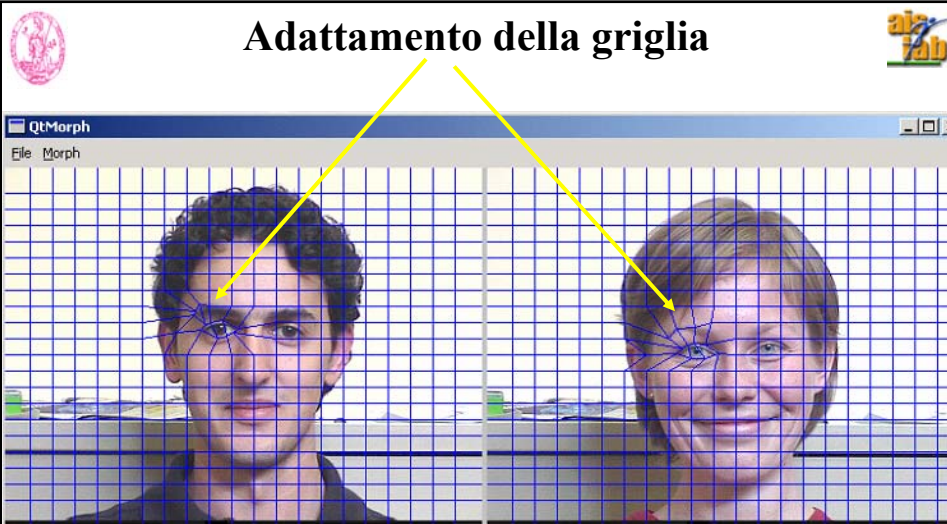


Click near a point of a mesh and drag it elsewhere in the image space

Si parte stendendo sulla coppia di immagini una griglia regolare.  
La griglia si può infittire in un secondo tempo.

A.A. 2004-2005 21/47 <http://homes.dsi.unimi.it/~borgnese/>

## Adattamento della griglia



Click near a point of a mesh and drag it elsewhere in the image space

- Identificazione dei punti corrispondenti (e dei quadrati corrispondenti)
- Deformazione della griglia (i rettangoli divengono quadrilateri)
- La corrispondenza diventa tra coppie di quadrilateri.

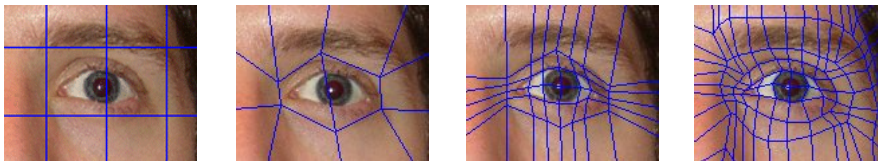
A.A. 2004-2005 22/47 <http://homes.dsi.unimi.it/~borgnese/>



## Deformazione delle griglie



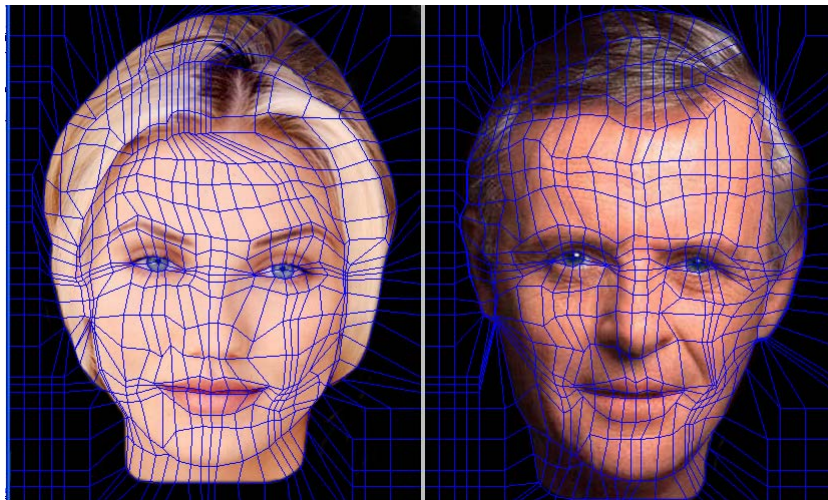
- **Spostamento punti:** le griglie vengono deformate trascinando con il mouse i punti generati dall'incrocio di linee verticali ed orizzontali della griglia (vertici dei quadrilateri).
- **Costruzione incrementale:** possibilità di inserire nuove linee per aggiungere ulteriori punti di controllo.



Più fitta sarà la griglia, maggiore sarà il livello di dettaglio... ed il tempo per impostarla.



## La coppia di mesh deformate



I lati dei quadrilateri seguono le feature.



## Mesh morphing



- Per ogni pixel dell'immagine intermedia:
  - Vengono calcolate le sue coordinate implicite.
  - Vengono identificati i pixel corrispondenti nella coppia di immagini di partenza.
  - Viene letto il colore dei pixel corrispondenti nella coppia di immagini di partenza.
  - Viene calcolato il colore del pixel di destinazione mediante cross-dissolvenza.

Definizione della griglia intermedia.

Per ogni quadrilatero, identificazione della corrispondenza tra pixel.

Velocità: circa 500ms per immagini di 1Mpixel.

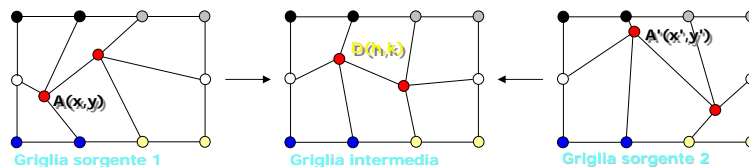


## Set-up della mesh sull'immagine destinazione



Per ogni maglia del reticolo:

- Combinazione lineare delle coordinate relative ai vertici dei quadrilateri omologhi.



Esempio: Considerando due vertici di coordinate  $A(x,y)$  e  $A'(x',y')$

$$h = x \cdot \alpha + (1 - \alpha) \cdot x'$$

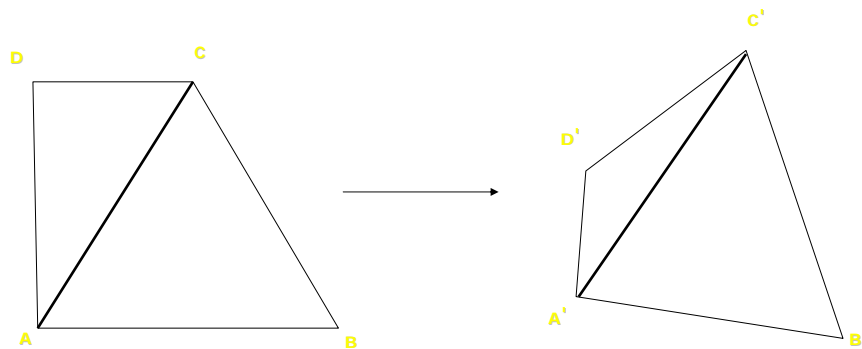
$$k = y \cdot \alpha + (1 - \alpha) \cdot y'$$

Con  $\alpha$  compreso tra 0 e 1: parametro di morphing.

Otengo la mesh disegnata sull'immagine destinazione



## Dai quadrilateri ai triangoli



I triangoli sono più facilmente gestibili.



## Base di uno spazio vettoriale (da SI)



N vettori linearmente indipendenti costituiscono una base,  $B$ , dello spazio vettoriale  $n$ -dimensionale,  $X$ .

$$B = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_N]$$

Linearmente indipendenti:

$$\lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 + \dots + \lambda_N \mathbf{b}_N = \mathbf{0}, \text{ Iff } \lambda_1 = \lambda_2 = \dots = \lambda_N = 0.$$

Qualsiasi vettore  $\mathbf{x} \in X$ , può essere rappresentato come combinazione lineare della base.

$$\mathbf{x} = \lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 + \dots + \lambda_N \mathbf{b}_N$$



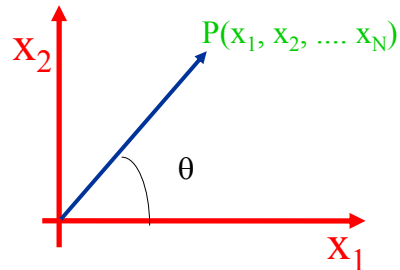
## Spazio vettoriale: esempio (da SI)



Il vettore  $P(x_1, x_2, \dots, x_N)$  e' esprimibile mediante le sue coordinate.  
Cosa sono le sue coordinate?

Consideriamo dei vettori "unitari" allineati con gli assi (unità di misura).

Le coordinate sono la misura della posizione del punto lungo gli assi.

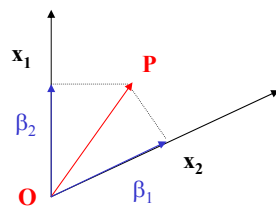


$$\text{Possiamo scrivere: } \mathbf{P} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 = [(\mathbf{P}-\mathbf{O}) \cdot \mathbf{x}_1] \mathbf{x}_1 + [(\mathbf{P}-\mathbf{O}) \cdot \mathbf{x}_2] \mathbf{x}_2$$

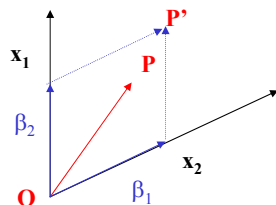
Negli spazi Hilbertiani



## Base non ortogonale dello spazio Euclideo



$$\beta_1 = (\mathbf{P}-\mathbf{O}) \cdot \mathbf{x}_1$$
$$\beta_2 = (\mathbf{P}-\mathbf{O}) \cdot \mathbf{x}_2$$



$$\mathbf{P}' = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2$$
$$\mathbf{P} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2$$

$$\alpha_i \neq \beta_i$$



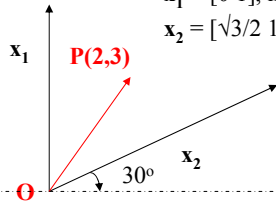
## Calcolo delle coordinate intrinseche ( $\alpha_1$ e $\alpha_2$ )



Consideriamo la base costituita da:

$x_1 = [0 \ 1]$ , norma unitaria.

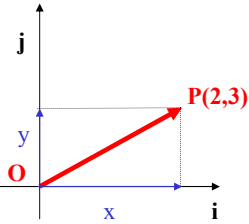
$x_2 = [\sqrt{3}/2 \ 1/2]$ , norma unitaria.



**Base ortogonale:**

$i = [1 \ 0]$ , norma unitaria.

$j = [0 \ 1]$ , norma unitaria.



$AX = B$

$$\begin{aligned} 2 &= \alpha_1 \cdot 0 + \alpha_2 \cdot \sqrt{3}/2 & A &= \begin{bmatrix} 0 & \sqrt{3}/2 \\ 1 & 1/2 \end{bmatrix} & 2 &= \alpha_1 \cdot 1 + \alpha_2 \cdot 0 \\ 3 &= \alpha_1 \cdot 1 + \alpha_2 \cdot 1/2 & & & 3 &= \alpha_1 \cdot 0 + \alpha_2 \cdot 1 \end{aligned} \quad A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$\det(A) = (-\sqrt{3}/2)(1) \neq 1$

Quando  $\det(A) = 0$ ?

$\det(A) = (1)(1) = 1$

$\alpha_1 = 1.8453$

$\alpha_2 = 2.3094$

$P = \alpha_1 x_1 + \alpha_2 x_2 = 1.8453 \cdot 0 + 2.3094 \cdot 0.866$   
 $1.8454 \cdot 1 + 2.3094 \cdot 1/2$

$\alpha_1 = 2$

$\alpha_2 = 3$

$P = \alpha_1 i + \alpha_2 j$

Coordinate intrinseche  $\neq$  proiezione ortogonale di P su  $x_1$  e  $x_2$



## Identificazione di un punto all'interno della mesh



Dato un triangolo ABC, non degenere, consideriamo i vettori  $(B-A)$  e  $(C-A)$ , i quali costituiscono una base per lo spazio vettoriale considerato. In particolare, ogni punto P del piano risulta individuato da una coppia ordinata  $(\lambda_1, \lambda_2)$  definita in modo tale che:

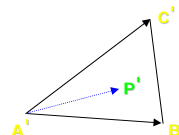
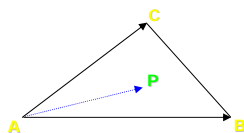
$$P = A + \lambda_1(B - A) + \lambda_2(C - A).$$

Possiamo riscrivere quanto affermato anche nel seguente modo:

$$P = A(1 - \lambda_1 - \lambda_2) + B\lambda_1 + C\lambda_2 = A\mu_1 + B\mu_2 + C\mu_3$$

Per determinare le coordinate baricentriche di un punto basta risolvere il sistema:

$$\begin{cases} (x_B - x_A)\lambda_1 + (x_C - x_A)\lambda_2 = x_P - x_A \\ (y_B - y_A)\lambda_1 + (y_C - y_A)\lambda_2 = y_P - y_A \end{cases}$$

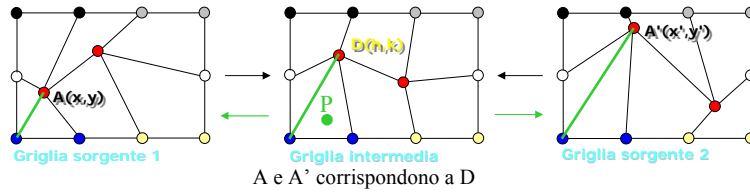


In questo modo identifico la posizione di un qualsiasi punto all'interno dei triangoli appartenenti alla mesh di partenza e di arrivo.





## Inverse mapping



Per ogni punto P dell'immagine intermedia:

- 1) Viene identificato il quadrilatero e poi il triangolo a cui appartiene.
- 2) Vengono calcolate le sue coordinate implicite al triangolo.
- 3) Vengono individuati sulle due immagini di origine e destinazione i punti che hanno le stesse coordinate.

Si ottiene quindi la corrispondenza tra tutti i punti dell'immagine intermedia ed i punti delle due immagini di partenza.

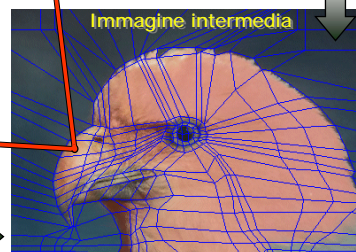
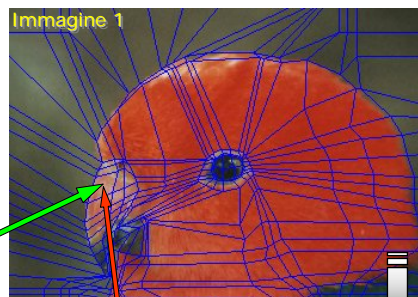
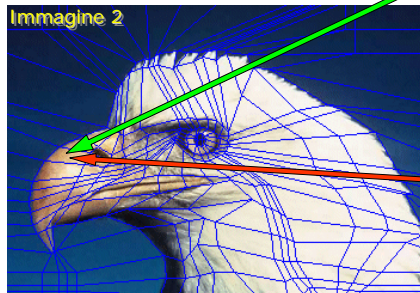


## Cross-dissolving



- Dalla corrispondenza dei punti si passa alla corrispondenza dei pixel.
- Vengono identificati i colori corrispondenti sulle due immagini sorgenti.
- Il colore finale viene ottenuto come combinazione lineare dei 2 colori:

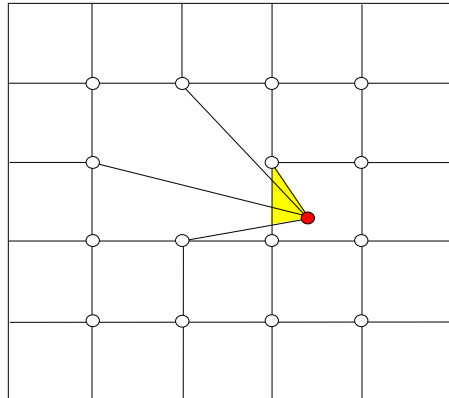
$$\text{Colore finale} = \text{RGBpixel}_1 \cdot \alpha + (1 - \alpha) \cdot \text{RGBpixel}_2$$





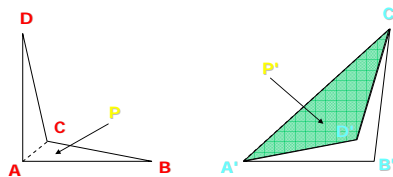
## Problema 1: Sovrapposizione di quadrilateri

Spostando i vertici della mesh, occorre evitare situazioni degeneri.

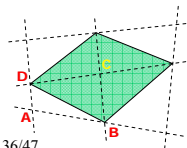


## Problema 2: quadrilateri convessi

- Dovendo trasformare un quadrilatero  $ABCD$  nell'omologo  $A'B'C'D'$ , può capitare una situazione come quella mostrata in figura:  
ai triangoli costituenti  $ABC$  e  $ACD$  del quadrilatero  $ABCD$ , corrispondono i triangoli  $A'B'C'$  e  $A'C'D'$  i quali, oltre ad essere uno dentro l'altro, sono parzialmente o totalmente esterni al quadrilatero  $A'B'C'D'$ . Col risultato che  $P$  apparterrà al quadrilatero  $ABCD$  mentre  $P'$  risulterà esterno ad  $A'B'C'D'$ .

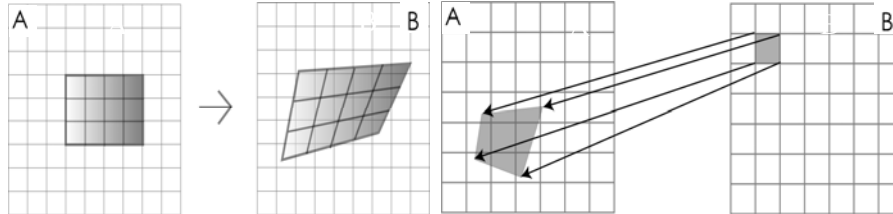


- **Soluzione:** limitazione nella manipolazione dei vertici.





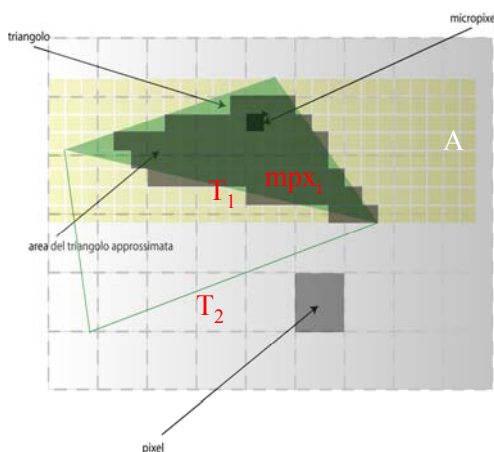
## Problemi nel warping



- La trasformazione (warping mediante mappatura diretta) di un pixel dell'immagine di partenza, A sull'immagine di destinazione B pone, in generale, tre problemi:
  - Il pixel cambia la sua forma da rettangolare a quadrilatero scaleno.
  - Il pixel di A trasformato non ricopre in generale un unico pixel di B: quale pixel viene messo in corrispondenza?
- Il colore (medio) del pixel nell'immagine destinazione, B sarà dato dalla media pesata (in base all'area) dei pixels su cui il pixel si "distende" sull'immagine originale A.



## Soluzione: Aliasing



Un triangolo, A conterrà diversi pixel.

Ciascun pixel può essere suddiviso a sua volta in aree quadrate, *micropixels*,  $mpx_i$

Il colore di ogni micropixel è quello associato al punto centrale.

Il colore finale di ogni pixel sarà dato dalla somma pesata:

$$\frac{\sum_i C(mpx_i) A_{mpx_i}}{\sum_i A_{mpx_i}}$$

**Tecniche di super sampling**



# Morphing non-lineare



■ Tipi di morphing (deformazione griglia):

$$h = x \cdot f(\alpha) + (1 - f(\alpha)) \cdot x'$$

$$k = y \cdot f(\alpha) + (1 - f(\alpha)) \cdot y'$$

- Lineare

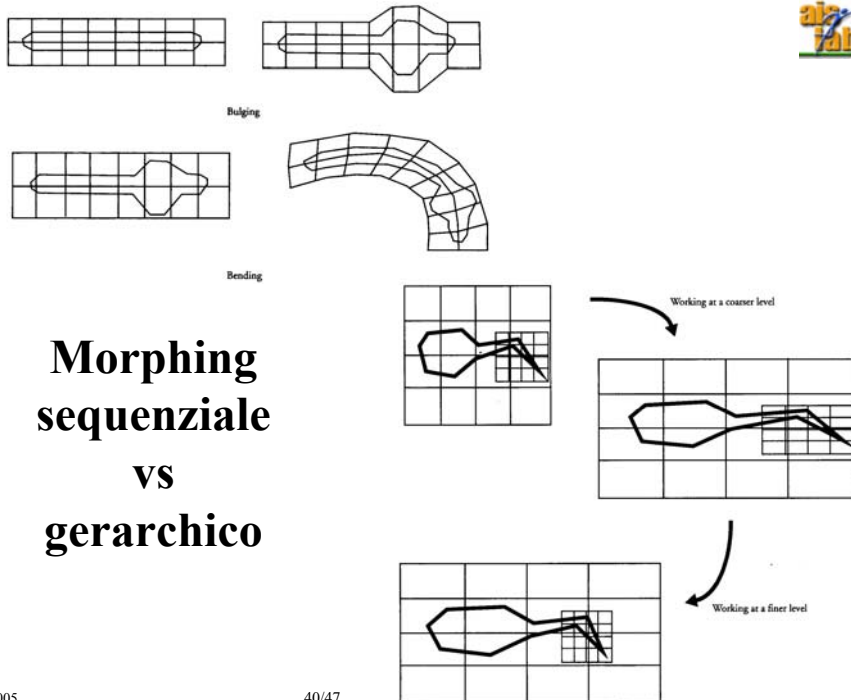
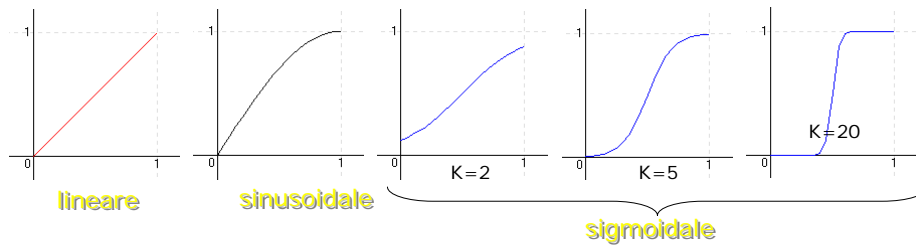
$$f(\alpha) = \alpha$$

- Sinusoidale

$$f(\alpha) = \sin(\alpha \cdot \pi/2)$$

- Sigmoidale

$$f(\alpha) = \frac{1}{1 + e^{2k\alpha - k}}$$



## Morphing sequenziale vs gerarchico



## Sw disponibile per Mesh morphing



Le versioni ottimizzate utilizzano i quadrilateri come poligono di controllo per B-spline.

<http://www.colorado-research.com/~gourlay/software/Graphics/Xmorph/>

<http://xmorph.sf.net>

<http://homes.dsi.unimi.it/~borgnese/Research/Software/index.html>

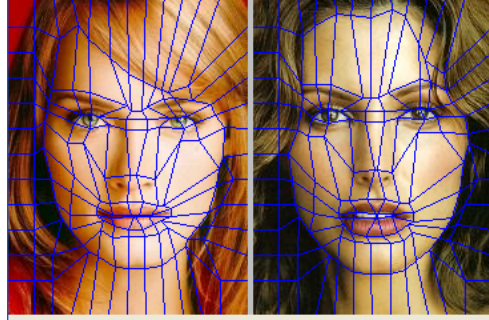
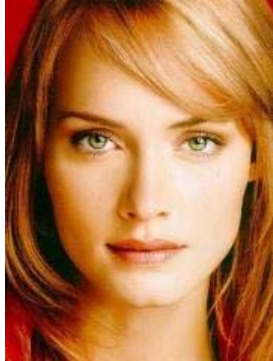


## Alcuni risultati



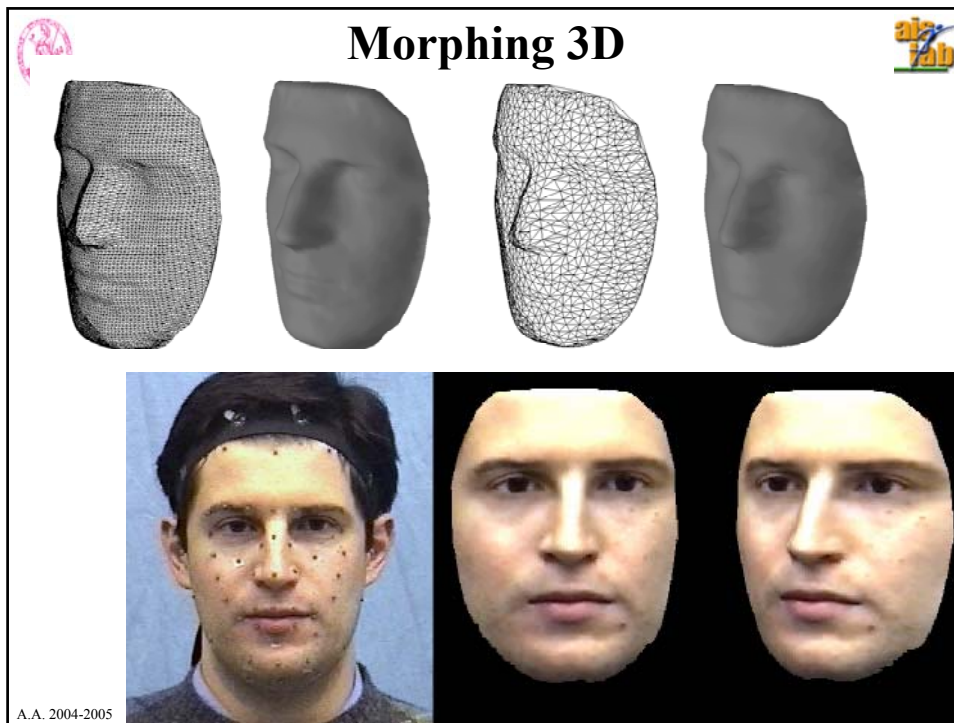


## Mesh insufficiente



## Altri morphing





## Morphing 3D

Mesh:

- Elenco di punti e della loro connettività
- Definizione del colore per ogni punto (Gouraud shading).
- Definizione di un'immagine e della corrispondenza tra pixel e vertici (Texture mapping).

Interpolazione delle mesh.

Requisito: stessa connettività.

A.A. 2004-2005 46/47 <http://homes.dsi.unimi.it/~borgese/>



## Sommario



Il morhing.

Field morphing.

Mesh morping.