

Teorie e tecniche del riconoscimento



<http://www.adherents.com/lit/comics/img/p/PlasticMan.jpg>

Parte 10 – Descrittori di forma

Dr. Marco Cristani

e-mail: marco.cristani@univr.it

ufficio: Stanza 47, Ca Vignal 2



Introduzione

- **Descrittore di forma:**
 - insieme di numeri prodotti per descrivere una forma
- l'importante è che i descrittori di forma siano **discriminativi** nei confronti della forma che individuano
- Un buon descrittore riesce a distinguere fortemente forme differenti e rende simili forme simili
 - (intuitivamente! → non esiste un unico concetto di similarità tra forme)



Descrittori elementari

- Alcuni semplici descrittori di forma:
 - **Area**: il numero di pixel contenuti nella forma
 - **Perimetro**: il numero di pixel sul contorno della forma
 - **(Non)Compattezza o (non)circularità**: $\text{perimetro}^2 / \text{area}$
 - la forma più compatta è il cerchio (4π), tutte le altre forme hanno compattezza $> 4\pi$
 - **Eccentricità**: il rapporto tra la corda più lunga nella forma e la più lunga corda perpendicolare ad essa
 - **Elongazione**: il rapporto tra altezza e larghezza del minor rettangolo che racchiude la forma (bounding-box)
 - **Rettangolarità**: quanto rettangolare è una forma: $\text{area della forma} / \text{area della bounding-box}$
 - **Orientazione**: angolo dell'asse maggiore della forma



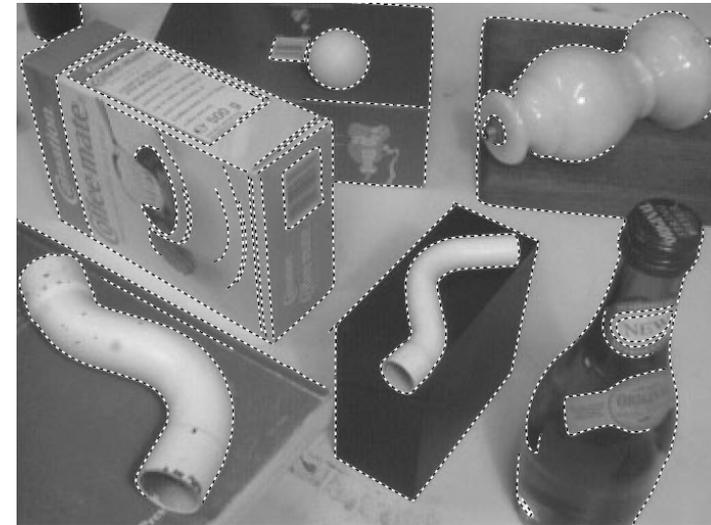
Tassonomia

- Esistono 2 tipologie di descrittori di forma:
 - **Region-based**
 - basati sull'area della forma
 - **Boundary-based**
 - basati sugli edge della forma 
- Avvertenze: come contare i pixel dei contorni?
 - 1 per i pixel 4-connessi
 - $\sqrt{2}$ per i pixel 8-connessi (i pixel diagonali)



Descrittori di forma basati sul contorno –spline

- Spline: curve parametriche
 - nello spazio, parametro s , crescente, reale
 - s determina lo sviluppo della curva
 - $x(s), y(s) = \langle x(s), y(s) \rangle$, con x, y particolari funzioni spline
 - nel tempo, parametro t
 - $r(s,t) = \langle x(s,t), y(s,t) \rangle$
- Spline di ordine d
 - funzione polinomiale a tratti
 - ogni tratto (= span) polinomio di ordine d



Descrittori di forma basati sul contorno –spline

- Ordine = numero di coefficienti necessari a definire un polinomio
- Grado = massimo esponente del polinomio

$$a + bx + cx^2$$

- Ordine = 3, grado = 2
- Di solito si usano polinomi di ordine 3 o 4
- Per approssimare una curva qualunque
 - Si aumenta l'ordine
 - si aumenta il numero di span (scelta migliore computazionalmente)



Descrittori di forma basati sul contorno – B-spline

- Una spline $x(s)$ puo' essere vista come somma pesata di funzioni base (le B-spline, appunto) $B_n(s)$, $n=0, \dots, N_B-1$ – da cui si parla di rappresentazione tramite B-spline, o semplicemente B-spline

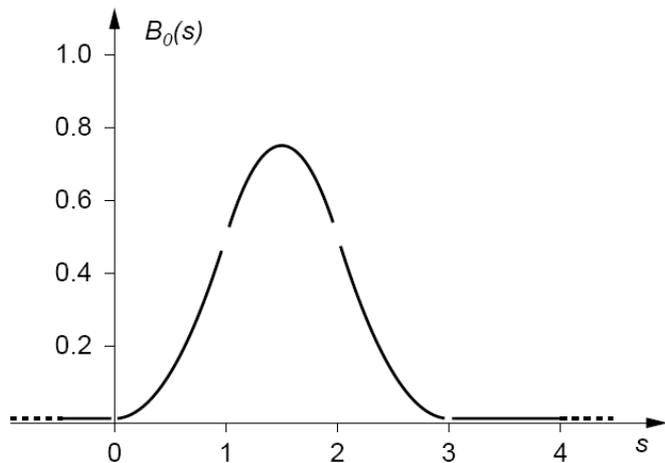
$$x(s) = \sum_{n=0}^{N_B-1} \overset{\text{pesi}}{x_n} B_n(s) \quad \left| \quad x(s) = \mathbf{B}(s)^T \mathbf{Q}^x \right. \begin{cases} \mathbf{Q}^x = \begin{pmatrix} x_0 \\ \vdots \\ x_{N_B-1} \end{pmatrix} \\ \mathbf{B}(s) = (B_0(s), B_1(s), \dots, B_{N_B-1}(s))^T \end{cases}$$

- Nel caso più semplice (caso regolare) ogni funzione base è rappresentata da d polinomi ognuno definito su uno span
- Per semplicità, definiamo uno span avente lunghezza unitaria
- Tutti gli span sono collegati in corrispondenza di nodi (*knots*)
- In questo caso, la curva risultante ha continuità $d-2$



Funzione di base

- Esempi:

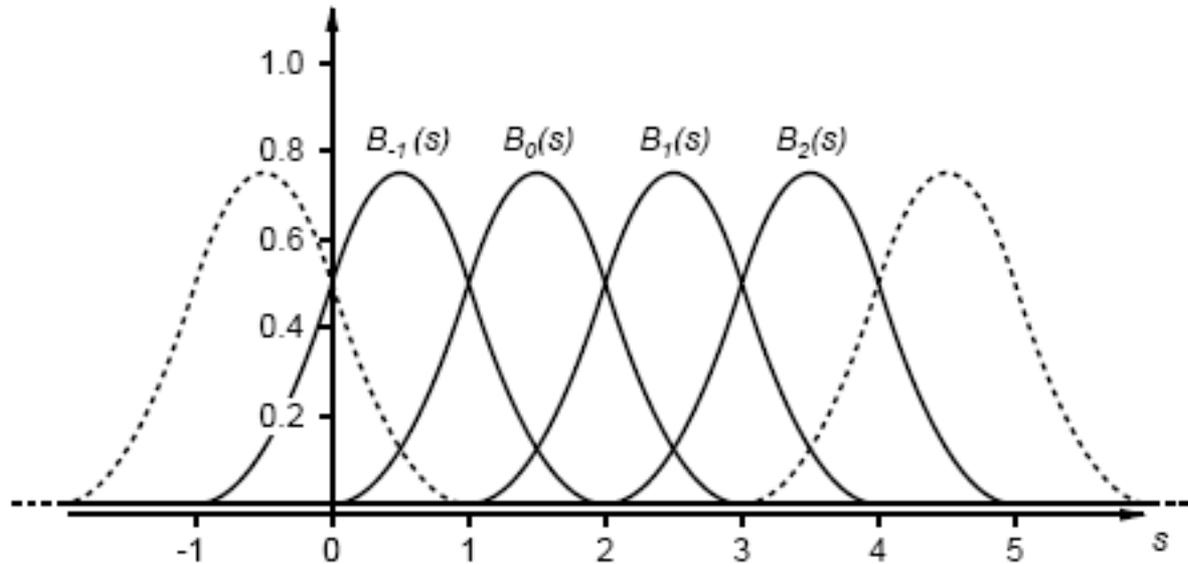


$$B_0(s) = \begin{cases} s^2/2 & \text{if } 0 \leq s < 1 \\ \frac{3}{4} - (s - \frac{3}{2})^2 & \text{if } 1 \leq s < 2 \\ (s - 3)^2/2 & \text{if } 2 \leq s < 3 \\ 0 & \text{otherwise} \end{cases}$$

- Funzione base $B_0(s)$ di ordine 3
- definita su 3 span
- parte da $s=0$



Funzioni di base – basi equispaziate



- Esempi:

- Le spline si possono localizzare $B_n(s) = B_0(s - n)$
- Le basi in teoria si definiscono anche all'infinito, in pratica $0 \leq s \leq L$



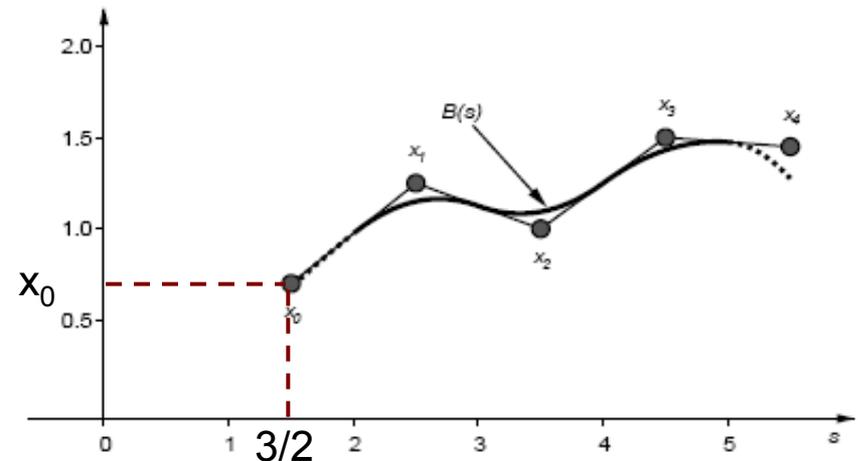
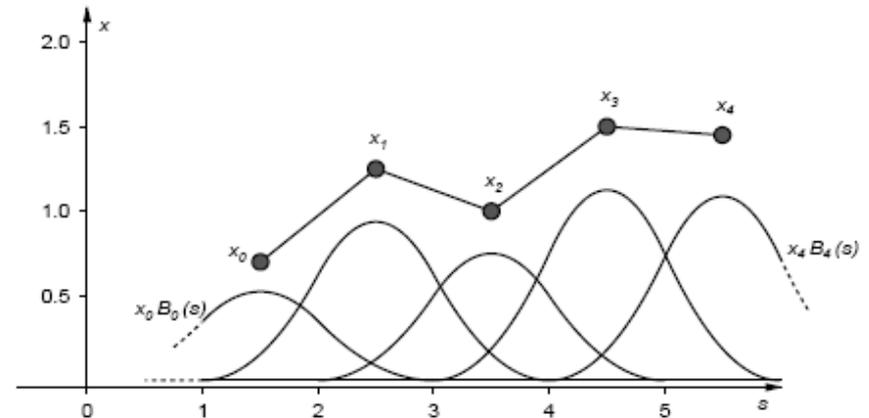
Creazione di una B-spline

- Esempi:

$$Q^x = \begin{pmatrix} x_0 \\ \vdots \\ x_{N_B-1} \end{pmatrix}$$

- Q^x individua la *forma* della curva
- la spline approssima i punti dati da

<ampiezzaspan/2,peso>



Forma spazio-temporale di una B-spline

- Generalizzando, nel caso di una curva in 2D, tempo-variante, avro' $\mathbf{r}(s, t)$

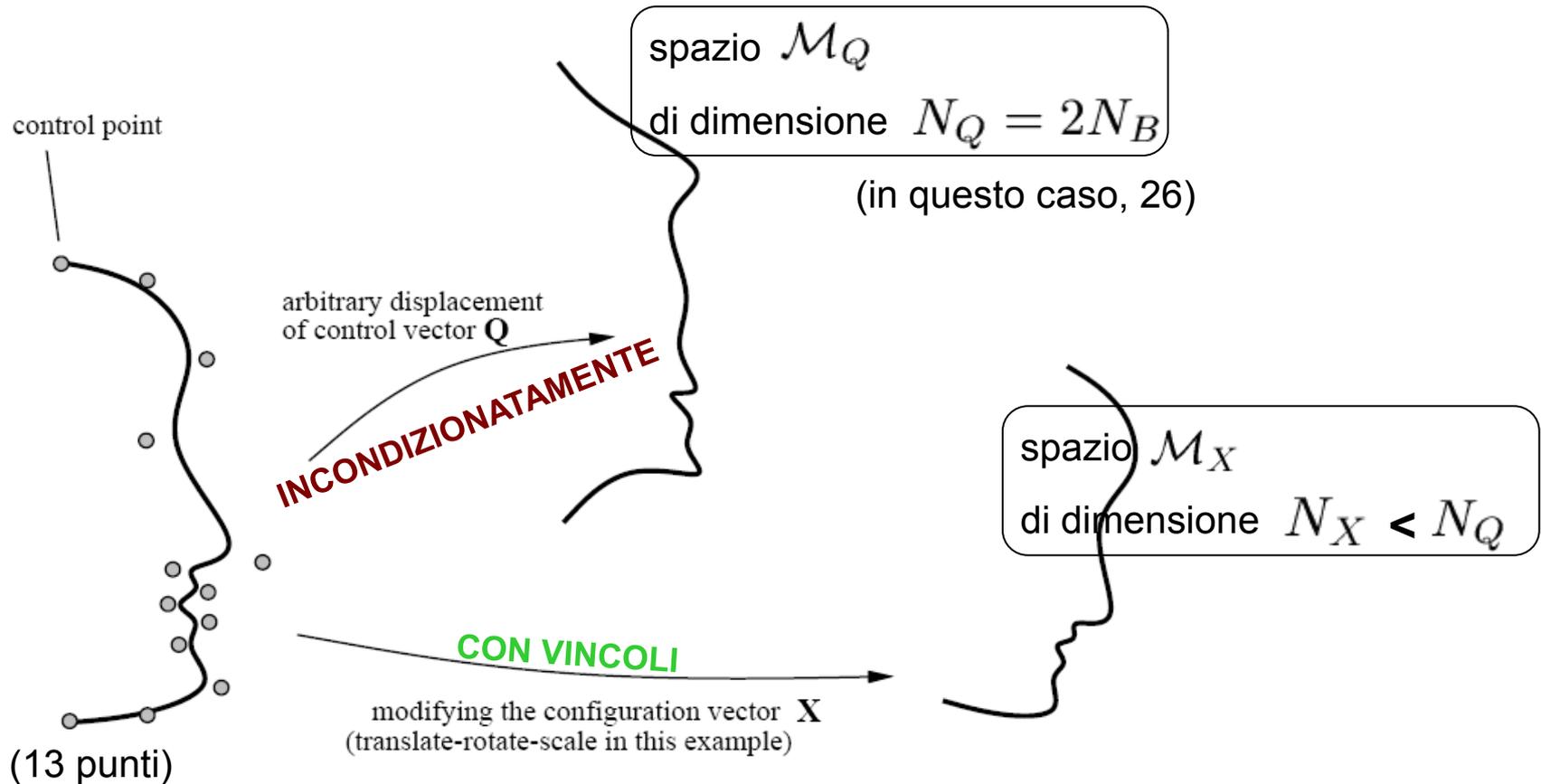
$$\mathbf{r}(s, t) = (\mathbf{B}(s) \cdot \mathbf{Q}^x(t), \mathbf{B}(s) \cdot \mathbf{Q}^y(t)) \quad \text{for } 0 \leq s \leq L$$

con $\mathbf{Q} = \begin{pmatrix} \mathbf{Q}^x \\ \mathbf{Q}^y \end{pmatrix}$



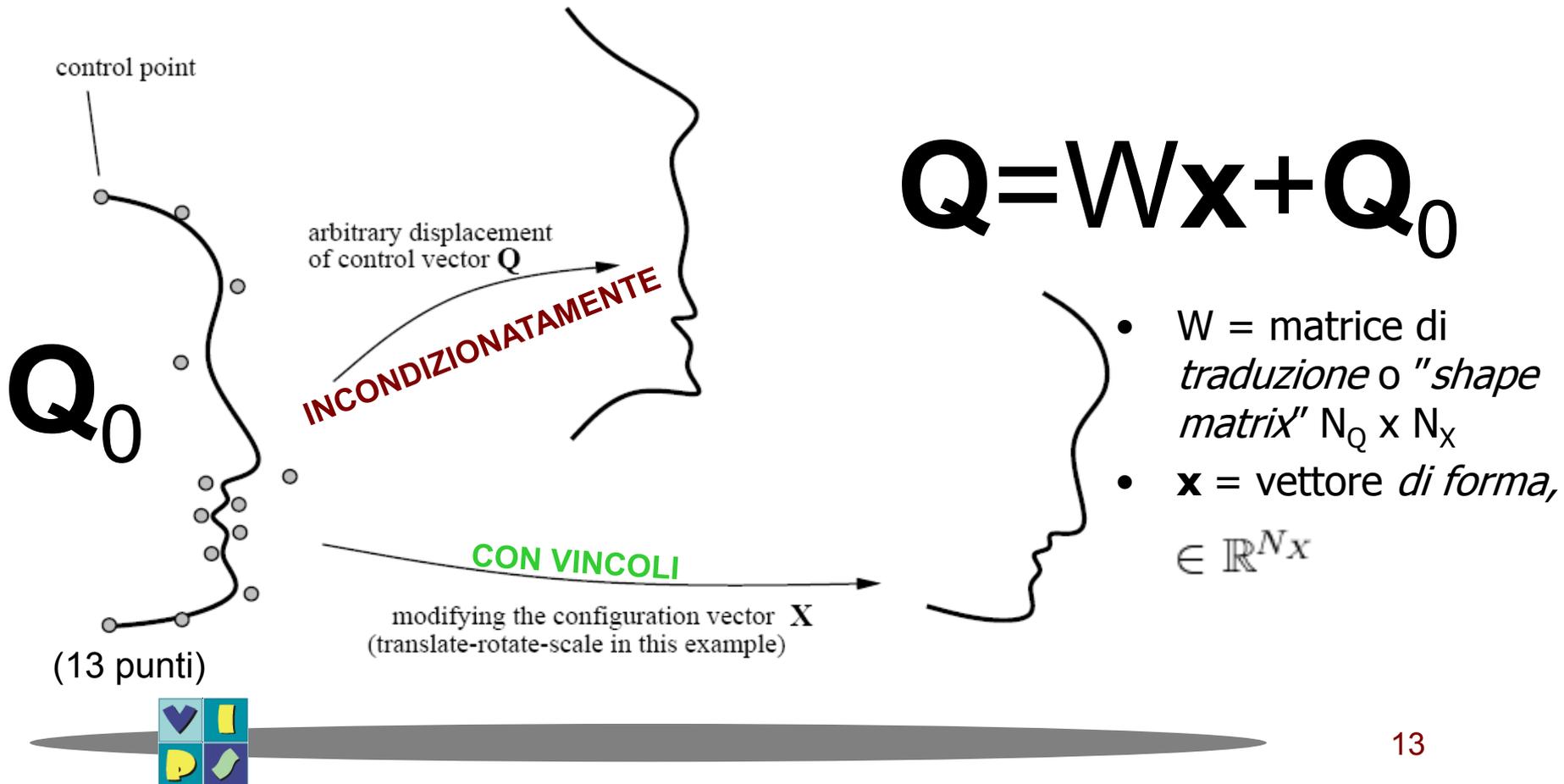
Evoluzione di una B-spline

- Come posso modificare \mathbf{Q} ? Due scelte:



Evoluzione vincolata di una B-spline

- Come posso modificare \mathbf{Q} ? Due scelte:



Trasformazioni

- Voglio avere trasformazioni affini
 - Classe di trasformazioni 2D lineari geometriche che mappano variabili (valori di intensità di pixel localizzati in un'immagine) in nuove variabili (in un'immagine di output), applicando una **combinazione lineare** di
 - **traslazioni**,
 - **rotazioni**,
 - **scalatura**,
 - **shearing**, ossia scalatura non uniforme in determinate direzioni
- E' possibile definire una trasformazione affine di un punto generico appartenente alla curva bspline come

$$\mathbf{r}(s) = \underset{2 \times 1}{\mathbf{u}} + \underset{2 \times 2}{M} \mathbf{r}_0(s)$$



Trasformazioni affini

$$\mathbf{r}(s) = \mathbf{u} + M\mathbf{r}_0(s)$$

- Traslazione $u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- Rotazione $u = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

θ : angolo positivo che indica rotazioni clockwise

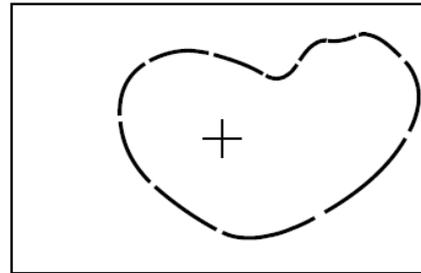
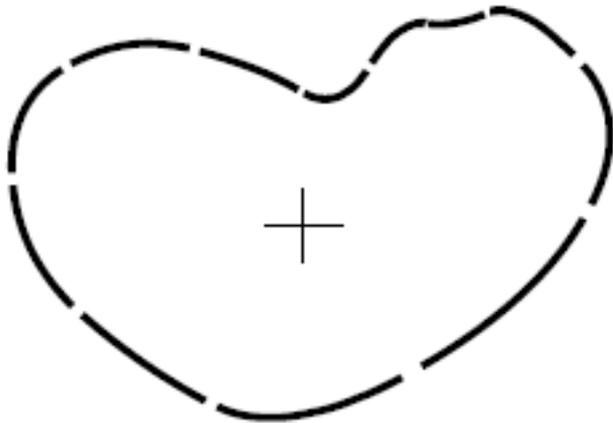
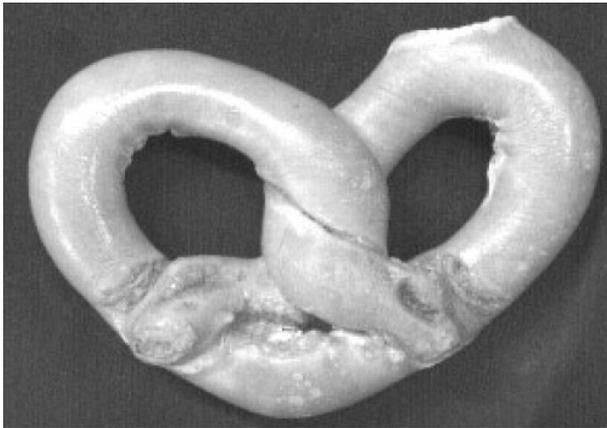
- Scalatura $u = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$M = \begin{bmatrix} a_{11} & 0 \\ 0 & a_{22} \end{bmatrix}$$

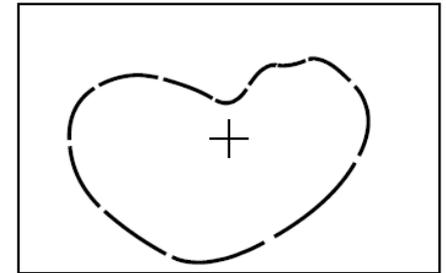


Esempi di trasformazioni

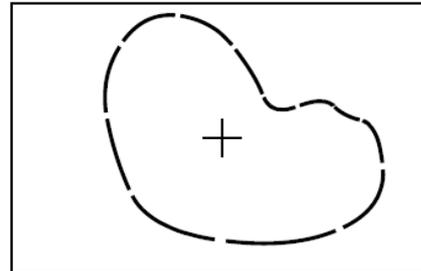
$$\mathbf{r}(s) = \mathbf{u} + M\mathbf{r}_0(s)$$



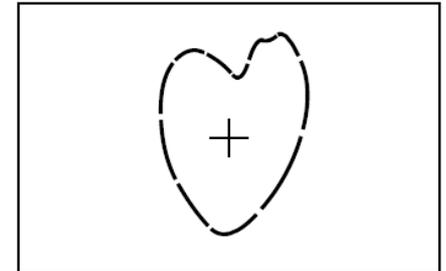
translate horizontally



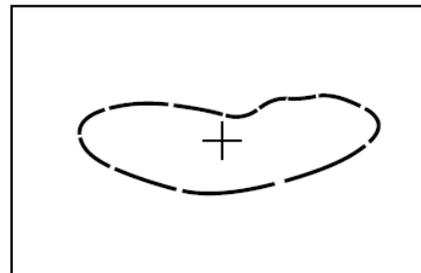
translate vertically



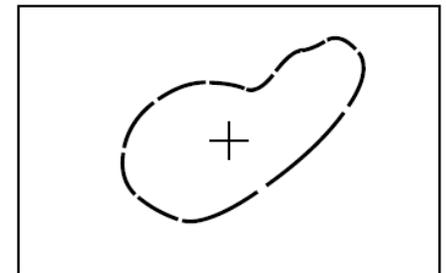
rotate



scale horizontally



scale vertically



scale diagonally



Identificatore di una curva

- Questo significa che ho 6 parametri da settare = 6 *gradi di libertà*
- Questi gradi di libertà possono essere linerizzati su un unico vettore $\mathbf{x} \in \mathbb{R}^{N_{\mathbf{x}}=6}$

$$\mathbf{x} = (u_1, u_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})^T$$


$$\mathbf{Q} = \mathbf{W}\mathbf{x} + \mathbf{Q}_0$$



Matrice di traduzione-*shape matrix*

- La matrice di traduzione è

$$W = \begin{pmatrix} \mathbf{1} & \mathbf{0} & \mathbf{Q}_0^x & \mathbf{0} & \mathbf{0} & \mathbf{Q}_0^y \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{Q}_0^y & \mathbf{Q}_0^x & \mathbf{0} \end{pmatrix}$$

\uparrow \uparrow
 \mathbf{x} \mathbf{y}
traslazioni

(riguardano)

rotazioni/scaling

$$\mathbf{1} = (1 \ 1 \ \dots \ 1)^T$$
$$\mathbf{0} = (0 \ 0 \ \dots \ 0)^T$$

sono vettori con N_B componenti
(= numero di funzioni base localizzate
~ numero di pesi = numero punti di controllo)

$$\mathbf{Q} = \mathbf{W}\mathbf{x} + \mathbf{Q}_0$$



Esempi di identificatori

- Dati:
$$W = \begin{pmatrix} 1 & 0 & Q_0^x & 0 & 0 & Q_0^y \\ 0 & 1 & 0 & Q_0^y & Q_0^x & 0 \end{pmatrix}$$
$$\mathbf{x} = (u_1, u_2, M_{11} - 1, M_{22} - 1, M_{21}, M_{12})^T$$

in pratica:

$\mathbf{x} = (0, 0, 0, 0, 0, 0)^T$ represents the original template shape Q_0

$\mathbf{x} = (1, 0, 0, 0, 0, 0)^T$ represents the template translated 1 unit to the right,

$\mathbf{x} = (0, 0, 1, 1, 0, 0)^T$ represents the template doubled in size

$\mathbf{x} = (0, 0, \cos \theta - 1, \cos \theta - 1, -\sin \theta, \sin \theta)^T$ represents the template rotated through angle θ

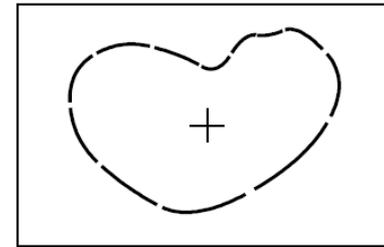
$\mathbf{x} = (0, 0, 1, 0, 0, 0)^T$ represents the template doubled in width



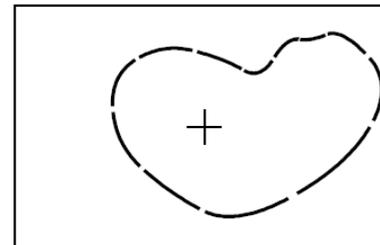
Trasformazioni euclidee

- Semplificando la modellazione, e restringendosi a trasformazioni euclidee ho una diversa matrice di traduzione

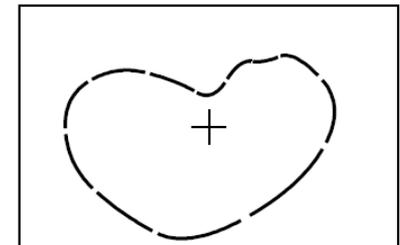
$$W = \begin{pmatrix} 1 & 0 & Q_0^x & -Q_0^y \\ 0 & 1 & Q_0^y & Q_0^x \end{pmatrix}$$



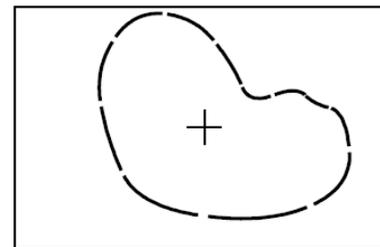
Original



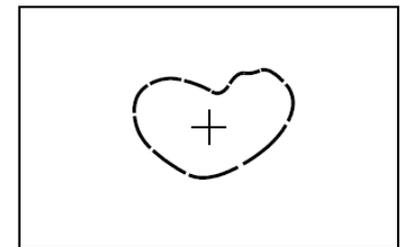
translate horizontally



translate vertically



rotate



scale isotropically



Ancora identificatori...

$\mathbf{x} = (0, 0, 0, 0)^T$ represents the original template shape Q_0

$\mathbf{x} = (1, 0, 0, 0)^T$ represents the template translated 1 unit to the right, so that, from (4.1),

$$Q = Q_0 + \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$\mathbf{x} = (0, 0, 1, 0)^T$ represents the template doubled in size

$$Q = 2Q_0$$

$\mathbf{x} = (0, 0, \cos \theta - 1, \sin \theta)^T$ represents the template rotated through angle θ :

$$Q = \begin{pmatrix} \cos \theta Q_0^x - \sin \theta Q_0^y \\ \sin \theta Q_0^x + \cos \theta Q_0^y \end{pmatrix}.$$



Esercizio

Esercizio: a cosa corrisponde questa trasformazione, in termini di fattore di scala e rotazione?

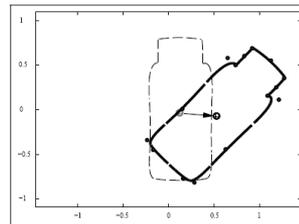
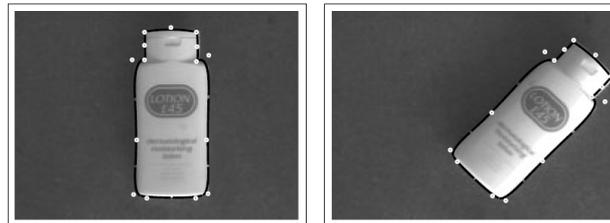
$$\mathbf{x} = (0.465, 0.047, -0.282, -0.698)^T$$

scalatura

$$\sqrt{(1 - 0.282)^2 + 0.698^2} = 1.001$$

rotazione

$$\arctan(1 - 0.282, -0.698) = -44.2^\circ$$



Trasformazioni non rigide

- Le trasformazioni affini/euclidee sono convenienti per modellare l'aspetto del moto 3D di un oggetto rigido
- Per le situazioni di deformazione (moto non rigido) questi strumenti non sono sufficienti
- Due soluzioni:
 1. Utilizzo dei **key-frames**
 - semplice ma poco espressivo
 2. **Learning** dello spazio delle deformazioni
 - si utilizza **PCA**



Trasformazioni non rigide con key frames

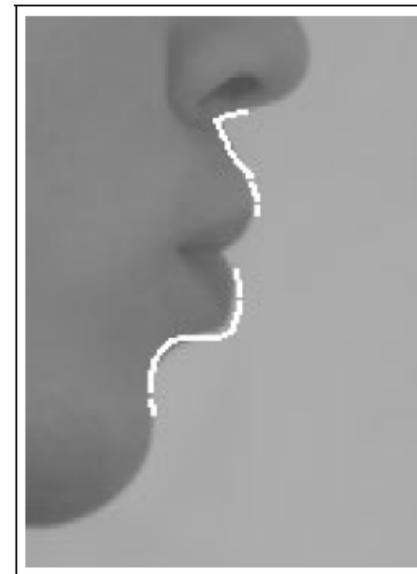
- Una curva viene vista come **combinazione lineare di forme elementari** (NB: non modello ora alcuna trasformazione)



Template Q_0



Key-frame: opening Q'_1



Key-frame: protrusion Q'_2



Trasformazioni non rigide con key frames (2)

- Una curva viene vista come **combinazione lineare** di **forme elementari** (NB: non modello ora alcuna trasformazione)
- Parto dalla solita espressione vincolante la forma della curva

$$\mathbf{Q} = \mathbf{W}\mathbf{x} + \mathbf{Q}_0$$

- Sfrutto i due k-frames $\mathbf{Q}'_1, \mathbf{Q}'_2$ e costruisco la matrice di traduzione

$$\mathbf{W} = \begin{pmatrix} \mathbf{Q}_1^x & \mathbf{Q}_2^x \\ \mathbf{Q}_1^y & \mathbf{Q}_2^y \end{pmatrix}$$

dove

$$\mathbf{Q}_i = \mathbf{Q}'_i - \mathbf{Q}_0$$



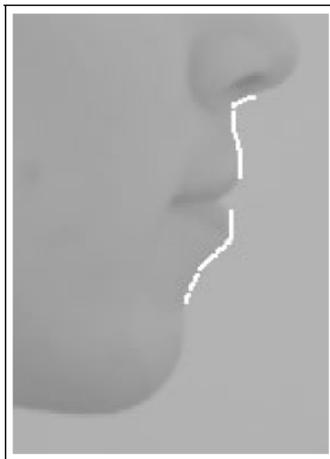
Trasformazioni non rigide con key frames (3)

- A questo punto risulta intuitivo formare nuove configurazioni

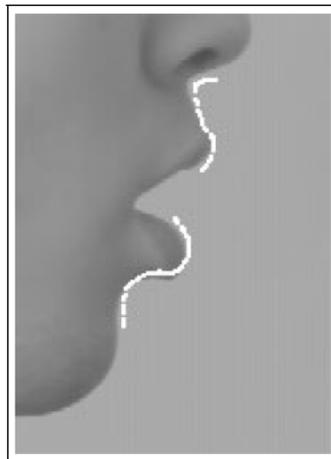
$\mathbf{x} = (0, 0)^T$ represents the closed mouth;

$\mathbf{x} = (1/2, 0)^T$ represents the half-open mouth;

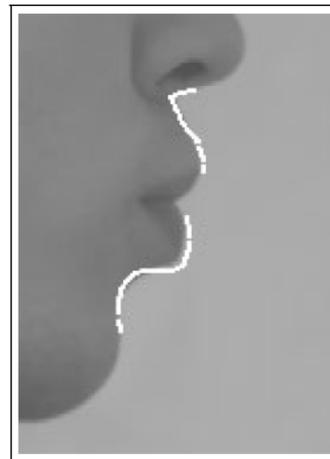
$\mathbf{x} = (1/4, 1/2)^T$ represents the mouth, half-protruding and slightly open.



Template Q_0



Key-frame: opening Q_1'



Key-frame: protrusion Q_2'



Trasformazioni non rigide con key frames (4)

- Per includere le trasformazioni (euclidee), la matrice utile risulta essere

$$W = \begin{pmatrix} 1 & 0 & Q_0^x & -Q_0^y & Q_1^x & -Q_1^y & Q_2^x & -Q_2^y \\ 0 & 1 & Q_0^y & Q_0^x & Q_1^y & Q_1^x & Q_2^y & Q_2^x \end{pmatrix}$$

- quindi:

$\mathbf{X} = (u, 0, 0, 0, 1, 0, 0, 0)^T$ represents the fully open mouth, shifted to the right by u ;

$\mathbf{X} = (0, 0, \cos \theta - 1, \sin \theta, 0, 0, \frac{1}{2} \cos \theta, \frac{1}{2} \sin \theta)^T$ represents the closed mouth, half-protruding and rotated through an angle θ .



Trasformazioni non rigide con key frames - limiti

- Problema:
 - come scegliere i k-frames?
 - quanti k-frames?
 - Se ho una sequenza di M frames, avro' $\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_M$ forme: sono tutte indicative di forme differenti?

$$W = \begin{pmatrix} 1 & 0 & Q_0^x & -Q_0^y & Q_1^x & -Q_1^y & Q_2^x & -Q_2^y \\ 0 & 1 & Q_0^y & Q_0^x & Q_1^y & Q_1^x & Q_2^y & Q_2^x \end{pmatrix}$$

- Soluzione: N_x
 - uso PCA → ...out of the scope!!!



Studio della dinamica di una B-spline

- Dato una forma in movimento, voglio dedurre la dinamica (da inserire in nel tracking, per esempio)
- Uso un modello auto regressivo (*auto-regressive process, ARP*) del secondo ordine

$$\mathbf{x}_t = A_2 \mathbf{x}_{t-2} + A_1 \mathbf{x}_{t-1} + \mathbf{D}_0 + B_0 \mathbf{w}_t$$

\mathbf{w}_t = valori estratti da distribuzioni normali

\mathbf{D}_0 = matrice di offset deterministica

A_2, A_1 = matrici, componenti deterministiche del sistema

B_0 = matrice, componente stocastica del sistema



Studio della dinamica di una B-spline (2)

$$\mathbf{x}_t = A_2 \mathbf{x}_{t-2} + A_1 \mathbf{x}_{t-1} + \mathbf{D}_0 + B_0 \mathbf{w}_t$$

- Per semplicità notazionale possiamo vedere anche

$$\mathbf{X}_t = \begin{pmatrix} \mathbf{x}_{t-1} \\ \mathbf{x}_t \end{pmatrix}$$

quindi

$$\mathbf{X}_t = A \mathbf{X}_{t-1} + \mathbf{D} + B \mathbf{w}_t$$

dove

$$A = \begin{pmatrix} 0 & I \\ A_2 & A_1 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} 0 \\ \mathbf{D}_0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 0 \\ B_0 \end{pmatrix}$$



Studio della dinamica di una B-spline (3)

RICORDA: $\mathbf{x}_t = A_2 \mathbf{x}_{t-2} + A_1 \mathbf{x}_{t-1} + \mathbf{D}_0 + B_0 \mathbf{w}_t$

- Lo studio dei parametri puo' essere formulato da un punto di vista statistico
- Voglio massimizzare la likelihood dei dati, dati i parametri

$$L(\mathbf{x}_1, \dots, \mathbf{x}_M | A_1, A_2, B_0, \mathbf{D}_0) \equiv \log p(\mathbf{x}_1, \dots, \mathbf{x}_M | A_1, A_2, B_0, \mathbf{D}_0)$$

$$p(\mathbf{x}_1, \dots, \mathbf{x}_k | A_1, A_2, B_0, \mathbf{D}_0) \propto \prod_{k=3}^M p_{B_0 \mathbf{w}_k}(\mathbf{x}_k - A_2 \mathbf{x}_{k-2} - A_1 \mathbf{x}_{k-1} - \mathbf{D}_0)$$

Per l'indipendenza tra i vettori di forma e il rumore

La modello come una normale multivariata



Studio della dinamica di una B-spline (4)

RICORDA: $\mathbf{x}_t = A_2 \mathbf{x}_{t-2} + A_1 \mathbf{x}_{t-1} + \mathbf{D}_0 + B_0 \mathbf{w}_t$

- Pertanto la forma della log-likelihood diventa

$$L(\mathbf{x}_1 \dots \mathbf{x}_M | A_1, A_2, B_0, \mathbf{D}_0) = -\frac{1}{2} \sum_{k=3}^M |B_0^{-1} (\mathbf{x}_k - A_2 \mathbf{x}_{k-2} - A_1 \mathbf{x}_{k-1} - \mathbf{D}_0)|^2 - (M - 2) \log \det B_0$$

- I parametri buoni danno alta likelihood. Quindi la devo massimizzare. E' una Normale multivariata, ed esiste forma chiusa per la massimizzazione.



Given a training set $\{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ of shapes from an image sequence, learn the parameters A_1, A_2, B_0 and \mathbf{D}_0 for a second-order AR process that describes the dynamics of the moving shape.

Algorithm

matrice



1. First, sums R_i , $i = 0, 1, 2$ and auto-correlation coefficients R_{ij} and R'_{ij} , $i, j = 0, 1, 2$ are computed:

$$R_i = \sum_{k=3}^M \mathbf{x}_{k-i}, \quad R_{ij} = \sum_{k=3}^M \mathbf{x}_{k-i} \mathbf{x}_{k-j}^T, \quad R'_{ij} = R_{ij} - \frac{1}{M-2} R_i R_j^T.$$

2. Estimated parameters \hat{A}_1 , \hat{A}_2 and $\hat{\mathbf{D}}_0$ are given by

$$\begin{aligned} \hat{A}_2 &= \left(R'_{02} - R'_{01} R'_{11}^{-1} R'_{12} \right) \left(R'_{22} - R'_{21} R'_{11}^{-1} R'_{12} \right)^{-1} \\ \hat{A}_1 &= \left(R'_{01} - \hat{A}_2 R'_{21} \right) R'_{11}^{-1} \\ \hat{\mathbf{D}}_0 &= \frac{1}{M-2} \left(R_0 - \hat{A}_2 R_2 - \hat{A}_1 R_1 \right). \end{aligned}$$

3. The covariance coefficient B_0 is estimated as a matrix square root $\hat{B}_0 = \sqrt{\hat{C}}$ where

$$\hat{C} = \frac{1}{M-2} \left(R_{00} - \hat{A}_2 R_{20} - \hat{A}_1 R_{10} - \hat{\mathbf{D}}_0 R_0^T \right).$$



Fitting di una B-spline sull'immagine

- Manualmente,
 - determino i punti sul contorno della forma, calcolo la B-spline
 - aumento gli span fino a quando sono soddisfatto del fitting
- Automaticamente
 - A partire da una forma binaria (ottenuta per esempio via sottrazione del background)
 - estraggo il perimetro
 - campiono il perimetro con una frequenza a piacere, ottengo i nodi
 - calcolo la B-spline



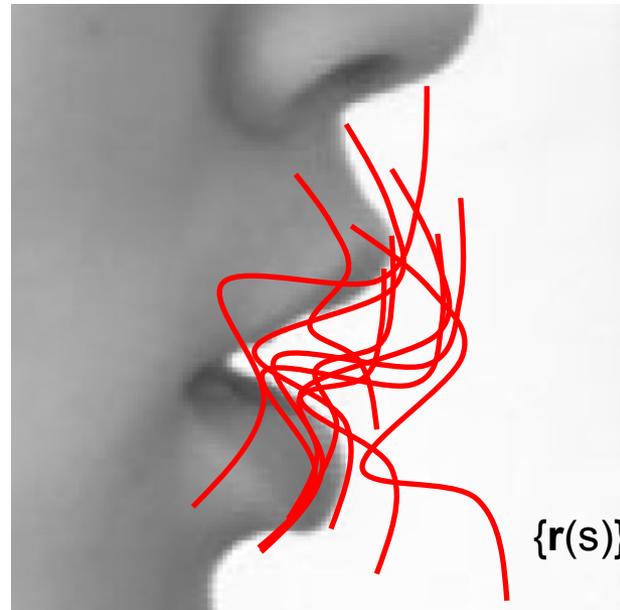
Applicazione di una B-spline a Condensation

- Parto da uno stato piu' probabile (la posterior) \mathbf{x}_{t-2}
- Ogni particella è un vettore \mathbf{x}_{t-1} che determina \mathbf{Q} , da cui origina una B-spline $\mathbf{r}(s)$
- La fase di **selezione o campionamento** produce un numero alto di campioni $\{\mathbf{x}_{t-1}\}$
- Ad essi applico la dinamica, ottenendo $\{\mathbf{x}_t\}$

$$\mathbf{x}_t = A_2 \mathbf{x}_{t-2} + A_1 \mathbf{x}_{t-1} + \mathbf{D}_0 + B_0 \mathbf{w}_t$$



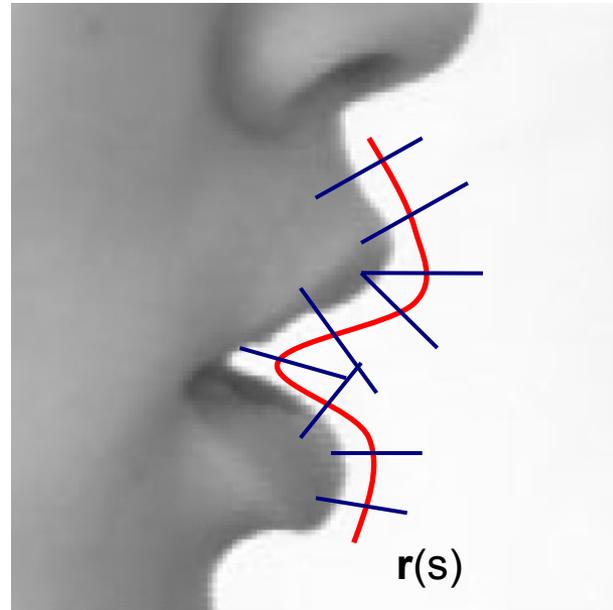
Applicazione di una B-spline a Condensation (2)



- La fase di **valutazione** valuta la spline che fitta meglio l'obiettivo



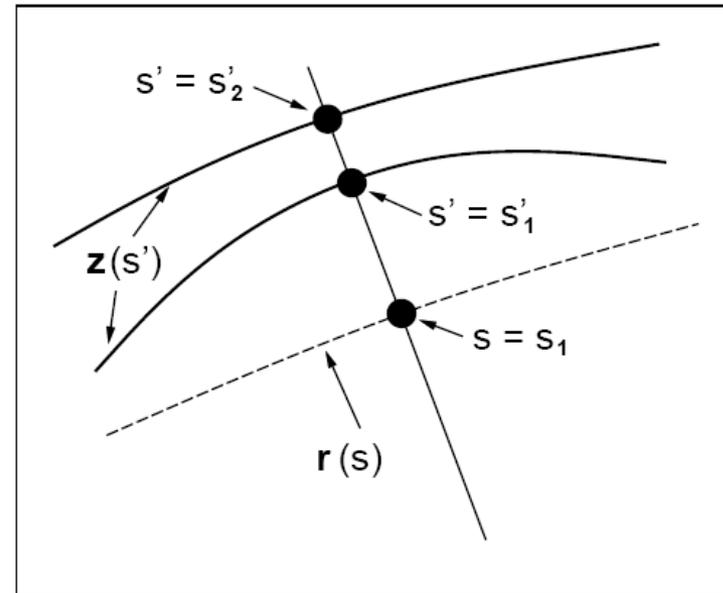
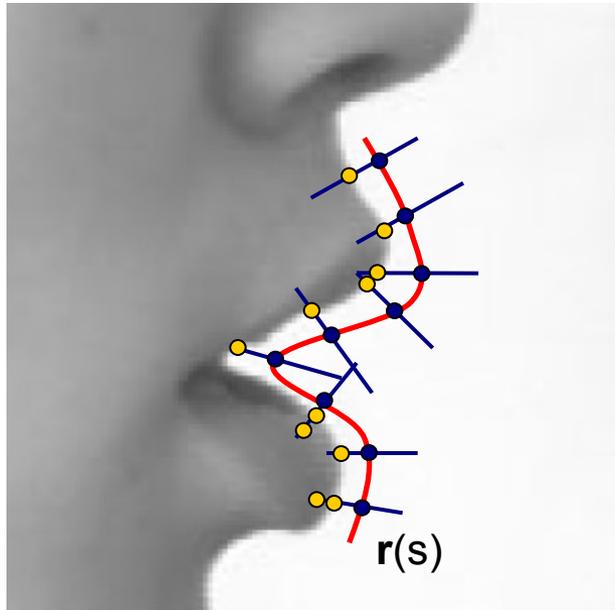
Applicazione di una B-spline a Condensation (3)



- in corrispondenza di M punti equispaziati sulla curva, costruisco dei segmenti normali, di lunghezza fissata a priori



Applicazione di una B-spline a Condensation (4)



- su ogni normale, calcolo gli edge
- noto che si puo' verificare la presenza di piu' edge in corrispondenza di una normale



Applicazione di una B-spline a Condensation (5)

- La funzione di pesatura o valutazione è

$$p(\mathbf{Z}|\mathbf{x}) \propto \exp \left\{ - \sum_{m=1}^M \frac{1}{2rM} f(\mathbf{z}_1(s_m) - \mathbf{r}(s_m); \mu) \right\}$$

dove

$$\mathbf{z}_1(s) = \mathbf{z}(s')$$

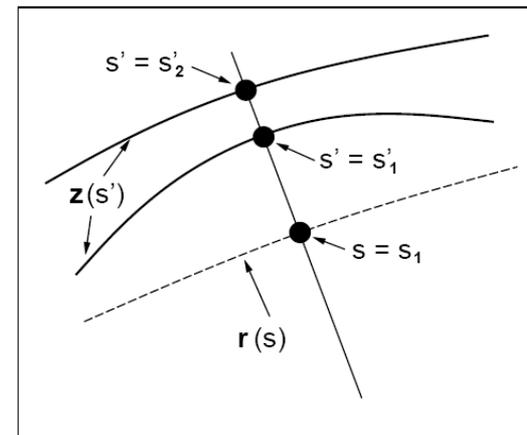
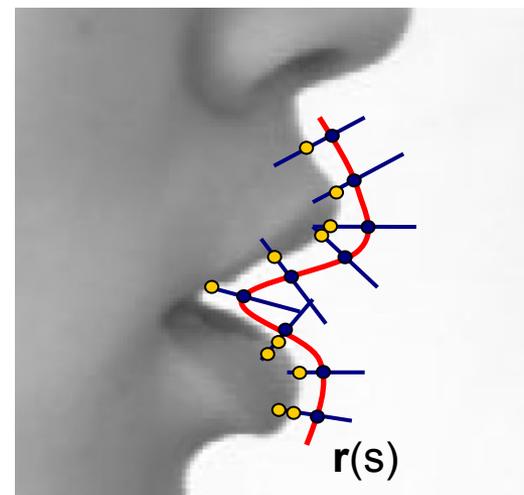
con

$$s' = \arg \min_{s' \in \text{normale}} |\mathbf{r}(s) - \mathbf{z}(s')|$$

e

- $f(\nu; \mu) = \min(\nu^2, \mu^2)$

- μ scelta opportunamente



Applicazione di una B-spline a Condensation (6)

- Nota che la sottrazione del background puo' essere inserita nel framework

