

Impronte digitali

Matching basato su minuzie

Annalisa Franco
annalisa.franco@unibo.it

Dario Maio
dario.maio@unibo.it

2

Formulazione del problema (1)

- Siano \mathbf{T} e \mathbf{I} , rispettivamente, la rappresentazione di un template e di un'immagine in input.
- Nel matching basato su minuzie la rappresentazione di un'impronta è un **vettore di feature di lunghezza variabile** contenente informazioni sulle minuzie rilevate.
- Nel caso più semplice una minuzia è rappresentata dalla terna $\mathbf{m}=\{x,y,\theta\}$ che specifica sia le **coordinate** della minuzia sia la sua **orientazione**; nel caso più generale è possibile far ricorso ad altri attributi (es. tipo di minuzia, qualità, ecc.):

$$\mathbf{T} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_m\} \quad \mathbf{m}_i = \{x_i, y_i, \theta_i\} \quad i=1, \dots, m$$

$$\mathbf{I} = \{\mathbf{m}'_1, \mathbf{m}'_2, \dots, \mathbf{m}'_n\} \quad \mathbf{m}'_j = \{x'_j, y'_j, \theta'_j\} \quad j=1, \dots, n$$

dove m e n indicano, rispettivamente, il numero di minuzie in \mathbf{T} e in \mathbf{I} .

Formulazione del problema (2)

Una minuzia \mathbf{m}'_j in \mathbf{I} e una minuzia \mathbf{m}_i in \mathbf{T} si considerano “**corrispondenti**” se la loro **distanza spaziale** (sd) è inferiore a una soglia r_0 e la **differenza tra le orientazioni** (dd) è inferiore a una soglia θ_0 :

$$sd(\mathbf{m}'_j, \mathbf{m}_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_0,$$

$$dd(\mathbf{m}'_j, \mathbf{m}_i) = \min(|\theta'_j - \theta_i|, 360^\circ - |\theta'_j - \theta_i|) \leq \theta_0$$

Le **tolleranze** r_0 e θ_0 sono necessarie per compensare possibili errori di estrazione delle feature e per gestire in parte le distorsioni della pelle che determinano uno spostamento delle minuzie.

Formulazione del problema (3)

- L'**allineamento** delle due impronte è un passo fondamentale per massimizzare il numero di minuzie corrispondenti.
- Un corretto allineamento richiede certamente una **traslazione** (in x e y) e una **rotazione**, ma può comportare anche altre trasformazioni geometriche:
 - **cambiamento di scala** (quando la risoluzione delle due impronte è diversa a causa di una diversa risoluzione dello scanner);
 - altre **trasformazioni geometriche tolleranti alla distorsione**.
- **Nota:** tollerare un numero elevato di trasformazioni introduce ulteriori gradi di libertà nell'algoritmo di matching, rendendo possibile un numero molto superiore di allineamenti e aumentando di conseguenza anche la possibilità di commettere errori.

Formulazione del problema (4)

- Sia $map(.)$ la funzione che mappa una minuzia \mathbf{m}'_j di \mathbf{I} nella minuzia \mathbf{m}''_j sulla base di una determinata **trasformazione geometrica**; per esempio si consideri una traslazione di $[\Delta x, \Delta y]$ e una rotazione in senso antiorario θ attorno all'origine:

$$map_{\Delta x, \Delta y, \theta}(\mathbf{m}'_j = \{x'_j, y'_j, \theta'_j\}) = \mathbf{m}''_j = \{x''_j, y''_j, \theta'_j + \theta\}$$

$$\begin{bmatrix} x''_j \\ y''_j \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

- Indichiamo con $mm(.)$ una funzione che restituisce **1 se due minuzie corrispondono** in base alle soglie fissate:

$$mm(\mathbf{m}''_j, \mathbf{m}_i) = \begin{cases} 1 & sd(\mathbf{m}''_j, \mathbf{m}_i) \leq r_0 \quad \text{e} \quad dd(\mathbf{m}''_j, \mathbf{m}_i) \leq \theta_0 \\ 0 & \text{altrimenti.} \end{cases}$$

Formulazione del problema (5)

Il problema del matching può essere formulato come:

$$\underset{\Delta x, \Delta y, \theta, P}{\text{maximize}} \sum_{i=1}^m mm(map_{\Delta x, \Delta y, \theta}(\mathbf{m}'_{P(i)}), \mathbf{m}_i)$$

dove $P(i)$ è una funzione ignota che determina gli **“accoppiamenti”** tra le minuzie di \mathbf{I} e \mathbf{T} ; in particolare **ogni minuzia ha una sola compagna** nell'altra impronta o non ha nessuna corrispondenza:

- $P(i)=j$ indica che la compagna di \mathbf{m}_i in \mathbf{T} è la minuzia \mathbf{m}'_j in \mathbf{I} ;
- $P(i)=\text{null}$ indica che \mathbf{m}_i in \mathbf{T} non ha corrispondenze in \mathbf{I} ;
- Una minuzia \mathbf{m}'_j in \mathbf{I} , tale che $\forall i = 1..m, P(i) \neq j$ non ha compagne in \mathbf{T} ;
- $\forall i = 1..m, k = 1..m, i \neq k \Rightarrow P(i) \neq P(k)$ or $P(i) = P(k) = \text{null}$

dunque P è una **funzione biettiva**, ovvero una minuzia in \mathbf{T} è accoppiata al massimo con una sola in \mathbf{I} .

Formulazione del problema (6)

- Una volta trovato un corretto allineamento diventa semplice risolvere il problema

$$\text{maximize}_{\Delta x, \Delta y, \theta, P} \sum_{i=1}^m mm(\text{map}_{\Delta x, \Delta y, \theta}(\mathbf{m}'_{P(i)}), \mathbf{m}_i)$$

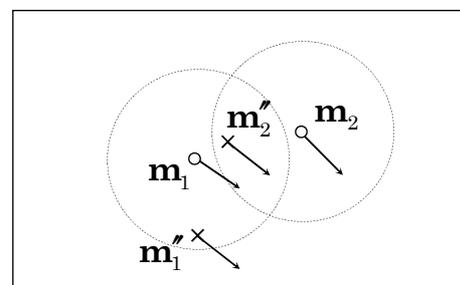
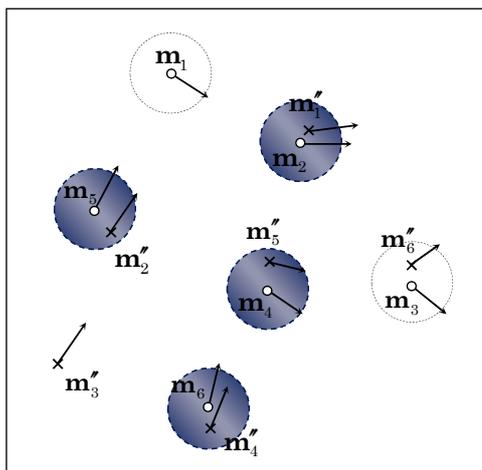
- Infatti per determinare la funzione P è sufficiente porre per ogni $i = 1..m$

$P(i) = j$ se $\mathbf{m}''_j = \text{map}_{\Delta x, \Delta y, \theta}(\mathbf{m}'_j)$ è la minuzia più vicina a \mathbf{m}_i tra le minuzie $\{\mathbf{m}''_k = \text{map}_{\Delta x, \Delta y, \theta}(\mathbf{m}'_k) \mid k = 1..n, mm(\mathbf{m}''_k, \mathbf{m}_i) = 1\}$;
 $P(i) = \text{null}$ se $\forall k = 1..n, mm(\text{map}_{\Delta x, \Delta y, \theta}(\mathbf{m}'_k), \mathbf{m}_i) = 0$

Problemi nel mapping di minuzie

Minuzie di I mappate nelle coordinate di T per un dato allineamento

- La minuzia \mathbf{m}_1 di T e la minuzia \mathbf{m}''_3 di I non hanno una compagna.
- Le minuzie \mathbf{m}_3 e \mathbf{m}''_6 non possono essere abbinate a causa dell'elevata differenza di orientazione.



Se \mathbf{m}_1 fosse abbinata a \mathbf{m}''_2 (la minuzia più vicina) \mathbf{m}_2 non potrebbe essere abbinata; abbinare \mathbf{m}_1 con \mathbf{m}''_1 permette di abbinare \mathbf{m}_2 con \mathbf{m}''_2 .

Calcolo dello score

- I sistemi automatici di riconoscimento di impronte devono restituire in output uno **score normalizzato** tra 0 e 1.
- Dato il numero k di minuzie corrispondenti:

$$\text{score} = \frac{k}{(n+m)/2}$$

- Si possono adottare alcuni accorgimenti:
 - Usare la qualità delle minuzie per pesare in modo diverso il contributo di minuzie di alta qualità e di bassa qualità.
 - Per non penalizzare eccessivamente le impronte che presentano una scarsa sovrapposizione con il template, come fattore di normalizzazione anziché $(n+m)/2$ si può usare il numero di minuzie che si trovano nell'intersezione delle due impronte dopo aver determinato l'allineamento ottimale.

Approcci al Point Pattern Matching (1)

- **Algebraic geometry**
 - Sono stati sviluppati diversi algoritmi ad hoc sia per $n=m$ (exact matching) sia per $n \leq m$ (inexact matching). Es. Bishnu et al. (2006) hanno proposto un metodo che opera con complessità in tempo $O(m^2 \cdot n^2 \cdot \log m)$
 - **Algoritmi general purpose non si adattano alle peculiarità delle impronte.**
- **Relaxation**
 - Questo approccio procede in modo iterativo regolando il livello di confidenza di ogni coppia di minuzie corrispondenti sulla base della consistenza con gli accoppiamenti delle altre minuzie, fino al raggiungimento di un criterio d'arresto. Ad ogni iterazione r , il metodo calcola $m \times n$ **probabilità di corrispondenza** p_{ij} tra il punto i e il punto j :

$$p_{ij}^{(r+1)} = \frac{1}{m} \sum_{h=1}^m \left[\max_{k=1..n} \{c(i, j; h, k) \cdot p_{ij}^{(r)}\} \right], \quad i = 1..m, \quad j = 1..n$$

dove $c(i, j; h, k)$ è una misura di compatibilità per l'accoppiamento che può essere determinata sulla base degli allineamenti necessari per mappare i in j e h in k .

- **Algoritmi lenti a causa della loro natura iterativa.**

Approcci al Point Pattern Matching (2)

- **Hough Transform**
 - Il problema del Point Pattern Matching viene convertito nel problema di individuare il picco più alto nello spazio di Hough dei parametri di trasformazione.
 - **Problemi in presenza di un numero basso di minuzie.**
- **Energy minimization**
 - Si definisce una funzione di costo sulla base di un insieme iniziale di possibili corrispondenze e si utilizzano appositi algoritmi di ottimizzazione (es. algoritmi genetici o simulated annealing) per trovare la soluzione ottimale.
 - **Metodi molto lenti e non adatti per riconoscimento on-line.**
- **Operations Research solutions**
 - Approcci **tree-pruning** che tentano di trovare la corrispondenza tra due set di punti ricercando all'interno di un albero di possibili confronti facendo ricorso a varie tecniche (es. **branch and bound**) per ridurre la dimensione dello spazio di ricerca. Altri approcci sono stati proposti come casi particolari di problemi noti (es. **assignment problem**, **bipartite graph matching**,..).

Matching globale: approccio basato su trasformata di Hough (1)

- L'applicazione della trasformata di Hough prevede la definizione di uno **spazio di parametri** in cui ogni punto è associabile a una soluzione del problema. Lo spazio è discretizzato e gestito come una matrice multidimensionale (denominata **array di accumulatori**) le cui entry sono dette celle.
- Per individuare la presenza di particolari forme in un'immagine il calcolo della trasformata implica una scansione dell'immagine:
 - ogni pixel **vota per una o più celle** dello spazio di parametri incrementandone la probabilità. Al termine della scansione le celle che hanno accumulato i **valori più elevati** indicano le **soluzioni più probabili** del problema.
- La definizione dello spazio di parametri e dei meccanismi di voto dipendono dal tipo di oggetto che si vuole identificare.

Matching globale: approccio basato su trasformata di Hough (2)

- Nel caso delle impronte lo spazio delle trasformazioni è costituito da **quadruple** $(\Delta x, \Delta y, \theta, s)$, dove Δx e Δy rappresentano i parametri di traslazione, θ è la rotazione e s corrisponde al cambiamento di scala. Ciascun parametro è discretizzato in un insieme finito di valori:

$$\Delta x^+ \in \{\Delta x_1^+, \Delta x_2^+, \dots, \Delta x_a^+\}, \quad \Delta y^+ \in \{\Delta y_1^+, \Delta y_2^+, \dots, \Delta y_b^+\}$$

$$\theta^+ \in \{\theta_1^+, \theta_2^+, \dots, \theta_c^+\}, \quad s^+ \in \{s_1^+, s_2^+, \dots, s_d^+\}$$

N.K. Ratha, K. Karu, S. Chen, A.K. Jain, "A Real-Time Matching System for Large Fingerprint Databases", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 799-813, 1996.

Matching globale: approccio basato su trasformata di Hough (3)

- Si crea un array a 4 dimensioni (una per ogni parametro) e i voti sono accumulati con il seguente algoritmo:

```

for each  $m_i, i = 1..m$ 
  for each  $m'_j, j = 1..n$ 
    for each  $\theta^+ \in \{\theta_1^+, \theta_2^+, \dots, \theta_c^+\}$ 
      if  $dd(\theta'_j + \theta^+, \theta_i) < \theta_0$ 
        // le direzioni delle minuzie dopo la rotazione sono abbastanza vicine
        for each  $s^+ \in \{s_1^+, s_2^+, \dots, s_d^+\}$ 
          { // la funzione map include cambiamenti di scala
             $\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} - s^+ \cdot \begin{bmatrix} \cos \theta^+ & -\sin \theta^+ \\ \sin \theta^+ & \cos \theta^+ \end{bmatrix} \begin{bmatrix} x'_j \\ y'_j \end{bmatrix}$ 
             $\Delta x^+, \Delta y^+ =$  approssimazione di  $\Delta x, \Delta y$  al bin più vicino
             $A[\Delta x^+, \Delta y^+, \theta^+, s^+] = A[\Delta x^+, \Delta y^+, \theta^+, s^+] + 1$ 
          }
  
```

Computational Complexity: $O(n \times m \times c \times d)$

$$(\Delta x^*, \Delta y^*, \theta^*, s^*) = \arg \max_{\Delta x^+, \Delta y^+, \theta^+, s^+} A[\Delta x^+, \Delta y^+, \theta^+, s^+]$$

Un approccio alternativo al Point Pattern Matching (1)

- L'algoritmo prevede i seguenti passi:
 1. Determinazione della coppia di minuzie (*coppia principale*) che riceve il massimo *Matching Pair Support (MPS)* e dei parametri di allineamento (θ , s) che permettono di abbinare il maggior numero di minuzie tra **T** e **I**.
 2. Le rimanenti coppie di minuzie (ovvero la funzione P) sono determinate una volta che le due impronte sono state allineate in modo da sovrapporre le minuzie corrispondenti alla coppia principale.
 3. L'allineamento esatto è calcolato con il **metodo dei minimi quadrati** una volta che la funzione di mapping è nota.

S.H. Chang, F.H. Cheng, W.H. Hsu, G.Z. Wu, "Fast Algorithm for Point Pattern-Matching Invariant to Translations, Rotations and Scale Changes", *Pattern Recognition*, vol. 30 no. 2, pp. 311-320, 1997.

Un approccio alternativo al Point Pattern Matching (2)

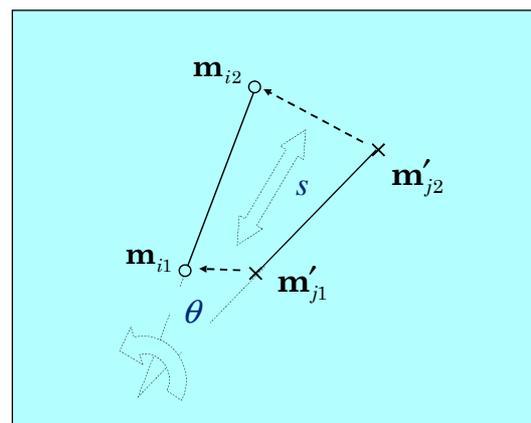
- Nel passo 1 l'algoritmo considera i segmenti definiti da coppie di minuzie $\overline{\mathbf{m}_{i2}\mathbf{m}_{i1}}$ in **T** e $\overline{\mathbf{m}'_{j2}\mathbf{m}'_{j1}}$ in **I** e determina, per ogni coppia di segmenti, i parametri θ e s come:

$$\theta = \text{angle}(\overline{\mathbf{m}_{i2}\mathbf{m}_{i1}}) - \text{angle}(\overline{\mathbf{m}'_{j2}\mathbf{m}'_{j1}})$$

Rotazione pari alla differenza di angolazione tra i due segmenti

$$s = \frac{\text{length}(\overline{\mathbf{m}_{i2}\mathbf{m}_{i1}})}{\text{length}(\overline{\mathbf{m}'_{j2}\mathbf{m}'_{j1}})}$$

Cambiamento di scala pari al rapporto tra le lunghezze dei due segmenti.



Matching con pre-allineamento

- Effettuare l'allineamento in fase di matching rende l'algoritmo robusto, ma risulta computazionalmente molto costoso.
- Memorizzare template pre-allineati e pre-allineare l'immagine in input può notevolmente velocizzare il matching in caso di identificazione (confronto 1:N).
- Esistono due principali approcci per il pre-allineamento:
 - **Assoluto**: ogni template è pre-allineato indipendentemente dagli altri, prima della memorizzazione nel database.
 - Il matching di un'impronta **I** con un insieme di template richiede che **I** sia registrata una sola volta e che la sua rappresentazione riallineata sia confrontata con tutti i template.
 - L'allineamento avviene solitamente traslando l'impronta in base alla posizione del core (difficile da localizzare correttamente).
 - **Relativo**: l'impronta in input **I** è pre-allineata con ogni template **T** nel database; il confronto 1:N richiede N pre-allineamenti.
 - Il pre-allineamento relativo permette certamente di migliorare l'efficienza ma in misura inferiore rispetto al pre-allineamento assoluto; consente in generale di ottenere risultati di accuratezza migliori, poiché le caratteristiche specifiche del template **T** possono essere usate per guidare l'allineamento.

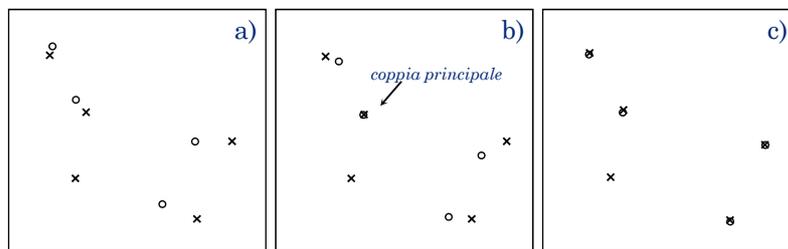
Pre-allineamento relativo M82 (1)

- Il metodo **M82** è stato sviluppato per il sistema di matching nell'Integrated AFIS dell'FBI (IAFIS).
- Si effettua dapprima un allineamento globale approssimativo in base alla posizione del core e all'orientazione media delle due regioni che si trovano ai lati del core.



Pre-allineamento relativo M82 (2)

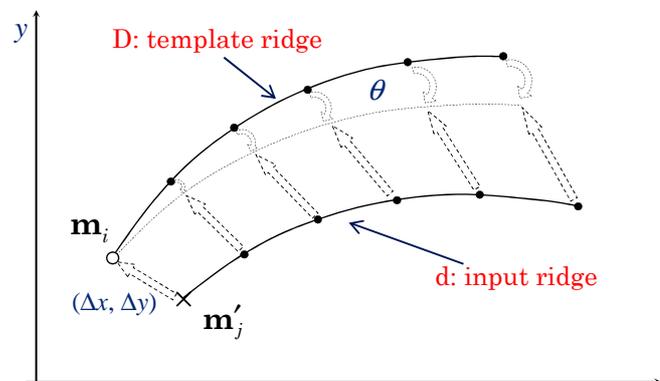
- Dopo un pre-allineamento approssimativo delle minuzie del template **T** e dell'impronta **I**, il metodo M82 determina un elenco di possibili coppie di minuzie considerando le minuzie che distano meno di una distanza prefissata.
- Il grado di matching di ogni possibile coppia è consolidato in base alla compatibilità con altre coppie.
- La lista di possibili coppie è ordinata rispetto al grado di matching; la coppia in prima posizione è selezionata come **coppia principale** e le restanti minuzie sono traslate di conseguenza.
- In un secondo passo è determinato un tensore di deformazione che permette di tollerare piccole distorsioni della pelle.



Pre-allineamento basato su ridge-line (1)

- **Idea di base:** le minuzie possono essere allineate allineando le corrispondenti ridge-line (ogni minuzia è associata a una ridge-line)
- La ridge è rappresentata come una curva, con origine corrispondente alla minuzia, e coordinata x nella stessa direzione della minuzia.
- Confrontando le ridge-line è possibile risalire ai parametri $(\Delta x, \Delta y, \theta)$.

Questo metodo procede iterativamente confrontando coppie di ridge finché non s'individua una coppia che supera un prefissato grado di matching.

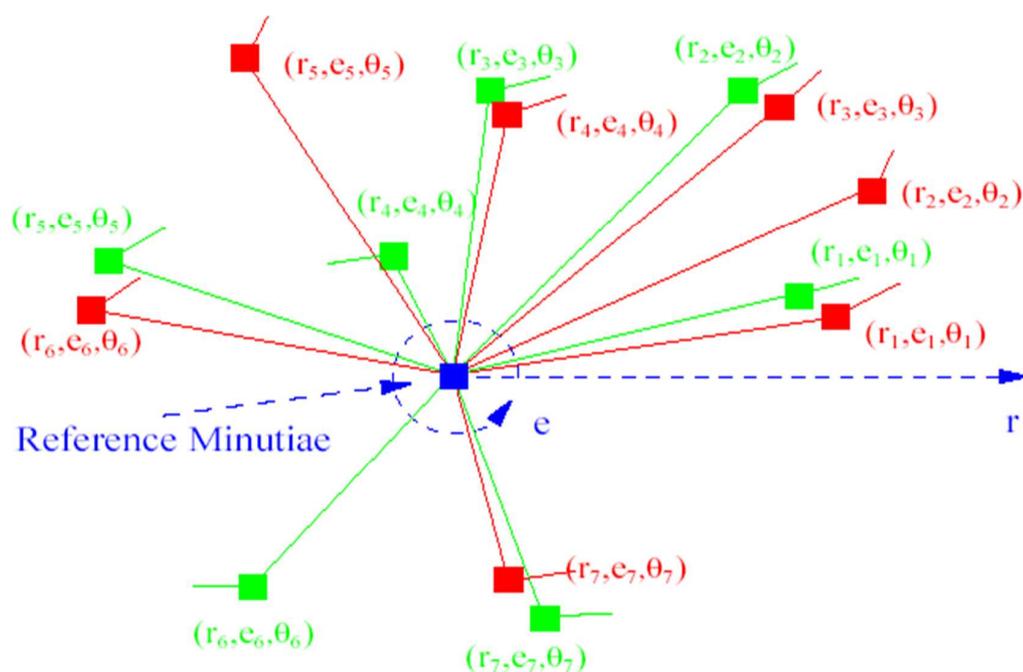


$$S = \frac{\sum_{i=0}^L d_i D_i}{\sqrt{\sum_{i=0}^L d_i^2 D_i^2}}$$

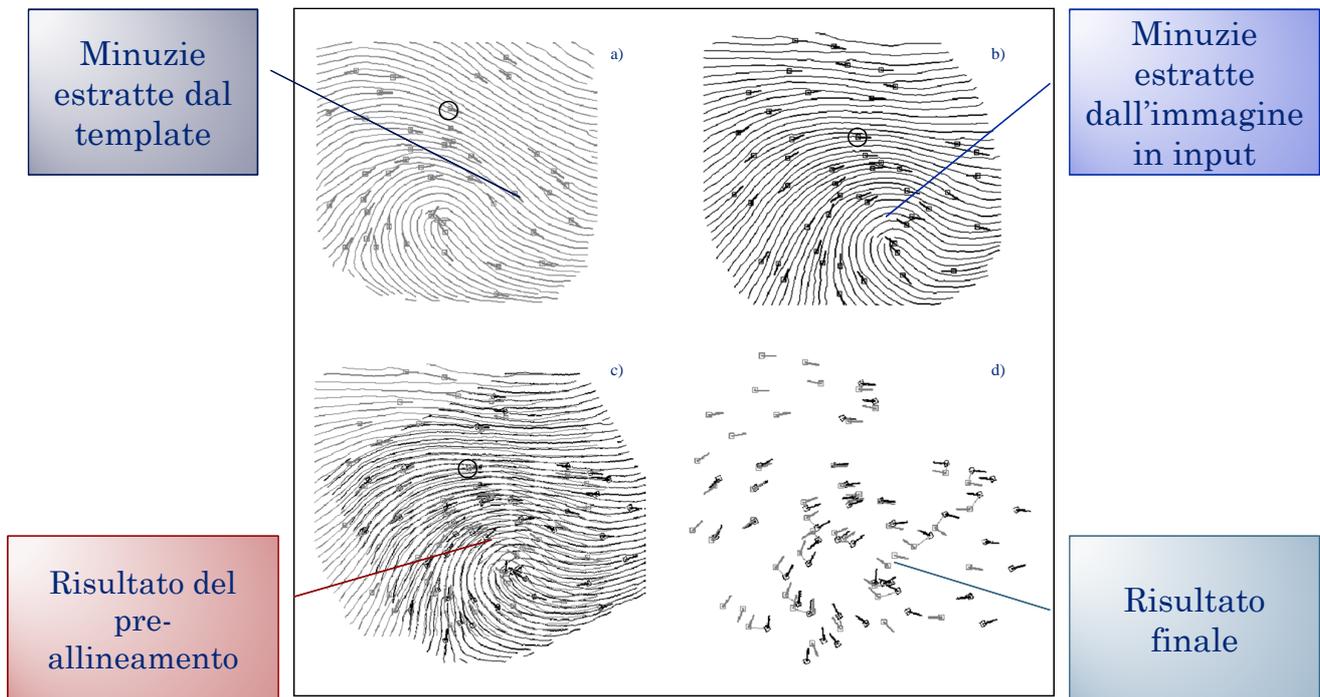
L : lunghezza minima delle due ridge;
 D_i, d_i : distanze dall'asse x del punto i sulle ridge D e d

Pre-allineamento basato su ridge-line (2)

- Per tollerare possibili errori in fase di estrazione delle minuzie è possibile adottare un algoritmo di **adaptive matching**:
 - Si converte ogni minuzia in **T** e **I** in un **sistema di coordinate polari** definito rispetto alla minuzia di riferimento (associata alla ridge selezionata durante il pre-allineamento).
 - T e I sono trasformate in **stringhe simboliche** concatenando ogni minuzia in ordine crescente di angolo radiale.
 - Le due stringhe sono confrontate con una tecnica di programmazione dinamica per trovare la loro **edit distance**.
 - Edit distance: numero di inserimenti, cancellazioni e scambi necessari per rendere identiche due stringhe.
 - Inserimenti e cancellazioni permettono di gestire minuzie mancanti e false minuzie.
 - Gli errori (**radiali e angolari**) possono essere gestiti fissando il costo delle modifiche proporzionale alla differenza delle posizioni.



Pre-allineamento basato su ridge-line (3)



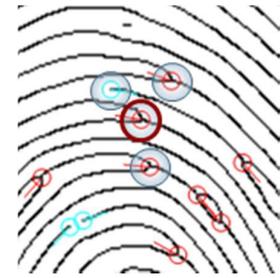
Matching globale vs. matching locale

- Il matching locale di minuzie consiste nel confronto tra due impronte sulla base della loro **struttura locale**, caratterizzata da attributi invarianti rispetto a trasformazioni globali (traslazione, rotazione, ecc.).
- Il **matching locale non richiede pre-allineamento** delle due impronte.
- Nel matching locale non sono considerate le **relazioni spaziali globali** tra minuzie che hanno un **elevato potere discriminante**.
- È possibile adottare **strategie ibride** nelle quali il matching locale di minuzie è seguito da un passo di **consolidamento** che ha lo scopo di validare il matching locale sulla base di un'analisi globale delle minuzie.

Tipologie di strutture locali

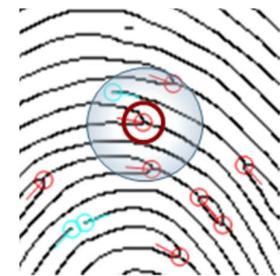
- **Nearest neighbor**

- L'intorno di una minuzia è costituito dalle K minuzie a essa più vicine.
- I descrittori locali hanno lunghezza fissa e possono essere confrontati in modo molto efficiente.



- **Raggio fisso**

- L'intorno di una minuzia è rappresentato da tutte le minuzie che si trovano a una distanza inferiore a un raggio prefissato.
- La lunghezza dei descrittori locali è variabile e dipende dalla densità locale.
- Il matching locale diventa più complesso, ma queste tecniche sono in grado di tollerare meglio minuzie false o mancanti
- Problemi possibili in prossimità del confine.

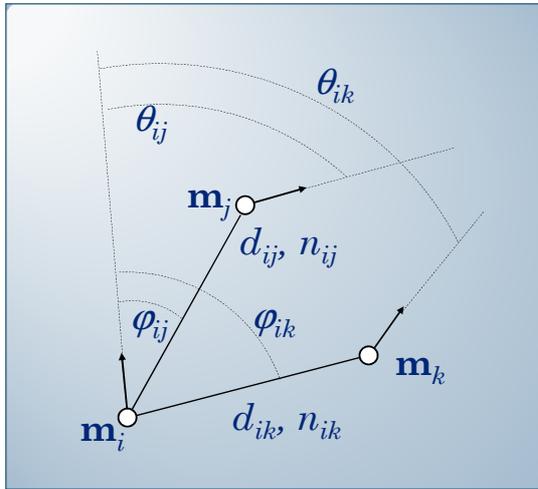


Matching locale: metodo di Jiang e Yau

- Le **strutture locali** sono definite da una minuzia centrale e dalle due minuzie a essa più vicine.
- Il vettore di feature \mathbf{v}_i associato alla minuzia \mathbf{m}_i con minuzie più vicine \mathbf{m}_j (la più vicina) e \mathbf{m}_k (la seconda più vicina) è dato da:

$$\mathbf{v}_i = [d_{ij}, d_{ik}, \theta_{ij}, \theta_{ik}, \varphi_{ij}, \varphi_{ik}, n_{ij}, n_{ik}, t_i, t_j, t_k],$$

Metodo Jiang e Yau: strutture locali



- d_{ab} è la distanza tra la minuzia \mathbf{m}_a e \mathbf{m}_b
- θ_{ab} è la differenza di direzione tra gli angoli θ_a e θ_b di \mathbf{m}_a e \mathbf{m}_b
- φ_{ab} è la differenza di direzione tra l'angolo θ_a di \mathbf{m}_a e la direzione dell'arco che connette \mathbf{m}_a e \mathbf{m}_b
- n_{ab} è il numero di ridge tra \mathbf{m}_a e \mathbf{m}_b
- t_a è il tipo associato alla minuzia \mathbf{m}_a .

Metodo Jiang e Yau: matching

- Il matching locale è effettuato calcolando, per ogni coppia di minuzie \mathbf{m}_i e \mathbf{m}'_j , $i = 1..m$, $j = 1..n$, una distanza euclidea pesata tra i rispettivi vettori di feature \mathbf{v}_i e \mathbf{v}'_j .
- Il calcolo delle $m \times n$ distanze è molto veloce.
- Lo score di similarità tra \mathbf{T} e \mathbf{I} si ottiene a partire dalle due strutture che ottengono il migliore risultato di matching (distanza minore).
- Per ottenere un'accuratezza migliore gli autori suggeriscono anche di effettuare un passo di consolidamento.

Matching locale: metodo di Ratha et al.

- Il concetto di struttura locale è definito in modo più formale tramite un **grafo** (**Minutiae Adjacency Graph, MAG**).
- La **stella** associata alla minuzia \mathbf{m}_i per una determinata distanza d_{\max} è il grafo $S_i = (V_i, E_i)$ costituito da:
 - l'**insieme dei vertici** V_i contenenti tutte le minuzie \mathbf{m}_j la cui distanza spaziale sd da \mathbf{m}_i è inferiore o uguale a d_{\max} : $V_i = \{ \mathbf{m}_j \mid sd(\mathbf{m}_i, \mathbf{m}_j) \leq d_{\max} \}$;
 - l'**insieme degli archi** $E_i = \{ \mathbf{e}_{ij} \}$, dove \mathbf{e}_{ij} è l'arco che connette la minuzia \mathbf{m}_i con la minuzia \mathbf{m}_j in V_i ; \mathbf{e}_{ij} è etichettato con una quintupla $(i, j, sd(\mathbf{m}_i, \mathbf{m}_j), rc(\mathbf{m}_i, \mathbf{m}_j), \varphi_{ij})$, dove $rc(\mathbf{m}_i, \mathbf{m}_j)$ è il numero di ridge tra \mathbf{m}_i e \mathbf{m}_j e φ_{ij} è l'angolo sotteso dall'arco con l'asse x .

N.K. Ratha, V.D. Pandit, R.M. Bolle, V. Vaish, "Robust Fingerprint Authentication Using Local Structural Similarity", *In Proc. Workshop on Applications of Computer Vision*, pp. 29-34, 2000.

Metodo di Ratha et al.: stella



$d_{\max} = 70$



$d_{\max} = 100$

Metodo di Ratha et al.: matching

Il matching tra due impronte avviene in tre passi:

1. Strict matching phase

Questa fase analizza tutte le possibili coppie di minuzie con l'obiettivo di individuare un numero sufficiente di coppie che presentano un buon grado di match, ovvero coppie di minuzie $u \in V_1$ e $v \in V_2$ caratterizzate da **strutture locali molto simili** in termini di distanza, angolo sotteso, numero e orientazione delle ridge line.

2. Consistency check

La seconda fase ha lo scopo di verificare la **consistenza degli accoppiamenti** individuati al passo precedente. La consistenza è verificata considerando che se (u_1, v_1) e (u_2, v_2) sono match corretti, allora l'arco (u_1, u_2) deve avere caratteristiche simili a (v_1, v_2) .

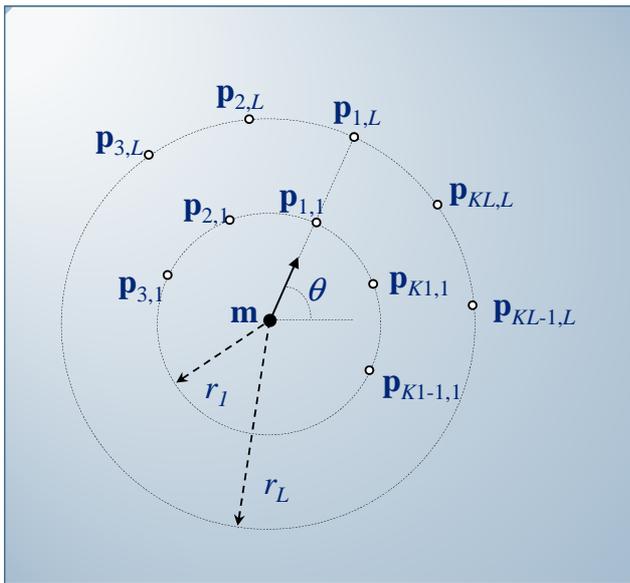
3. Extension phase

Nell'ultima fase sono **rilassati i vincoli** per l'accoppiamento di minuzie e si verifica quali nuovi abbinamenti possono essere realizzati, compatibilmente con quelli effettuati nella fase 1. Si calcola un costo di matching complessivo sulla base del quale viene presa la decisione finale.

Matching locale: metodo di Tico e Kuosmanen

- Il metodo adotta una struttura locale centrata sulla minuzia e basata su **informazioni invarianti non direttamente legate alle minuzie circostanti**.
- I descrittori locali includono informazioni sull'**orientazione** stimata in alcuni punti campione attorno alla minuzia **m**.
- Ogni orientazione è calcolata come **angolo relativo rispetto all'orientazione θ della minuzia**, risultando così invariante rispetto alla posizione e all'orientazione della minuzia.

Metodo Tico e Kuosmanen: strutture locali



- I punti campione \mathbf{p} sono posizionati lungo **L cerchi concentrici di raggio r_l** , ($1 \leq l \leq L$): il primo punto di ciascun cerchio ha distanza angolare 0 rispetto alla direzione della minuzia; gli altri sono ***uniformemente distribuiti***.
- Il numero di punti campione è diverso per i diversi cerchi.

- Gli autori propongono una relazione per il numero di punti $K_l = \lceil 172 \cdot r_l / R \rceil$ dove R è la risoluzione dell'impronta in dpi.

Metodo Tico e Kuosmanen: matching

- La ***similarità*** tra due descrittori è data dalla ***differenza angolare media*** tra orientazioni corrispondenti.
- **Vantaggio** del metodo: l'uso di strutture locali che non codificano la relazione con minuzie circostanti permette di ***gestire meglio minuzie false o mancanti***.
- **Svantaggio**: è indispensabile un passo di ***consolidamento*** per sfruttare il potere discriminante dato dalla posizione e orientazione delle minuzie.

Consolidamento

- Il consolidamento permette di verificare se gli score ottenuti dal confronto di strutture locali sono validi anche a livello globale.
- Un metodo molto semplice consiste nell'usare un matching globale dove gli score ottenuti dal matching locale possono essere usati per escludere gli abbinamenti di determinate minuzie.
 - **Vantaggi:**
 - riduzione dello spazio di ricerca;
 - maggiore efficienza.
 - **Svantaggi:**
 - il matching globale non è robusto rispetto alla distorsione e potrebbe complessivamente ridurre l'accuratezza.

Consolidamento: trasformazione singola

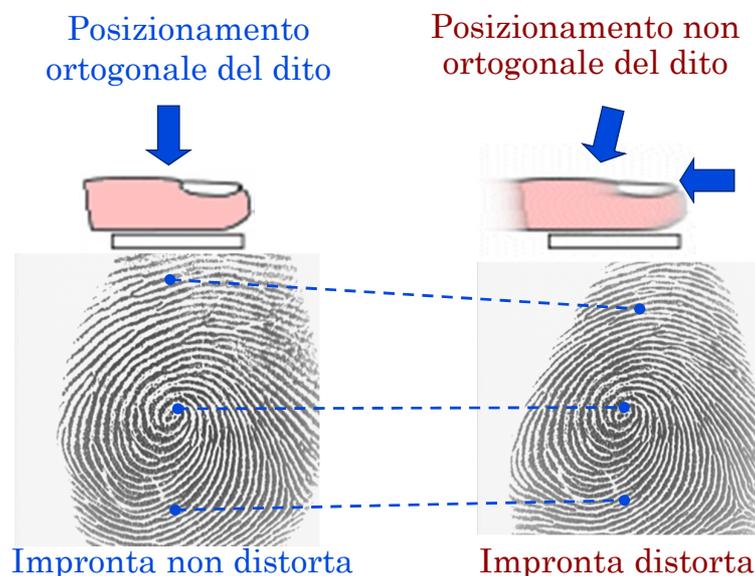
- Il metodo di consolidamento più semplice fa ricorso all'allineamento delle minuzie di **T** e **I** sulla base della migliore trasformazione risultante dal matching di strutture locali:
 1. La migliore trasformazione si ottiene allineando la minuzia centrale delle due strutture locali che ricevono lo score più alto:
 - Il vettore di traslazione è determinato dallo scostamento tra l'origine delle minuzie.
 - La rotazione è determinata dalla differenza angolare tra le direzioni delle due minuzie.
 2. La migliore trasformazione si determina tramite il matching simultaneo di due minuzie di **T** con due minuzie di **I**.
 - Complessità computazionale più elevata.
 - Maggiore robustezza.

Consolidamento: trasformazione multipla

- Il metodo basato su singola trasformazione può risultare inefficace in presenza di impronte di bassa qualità o affette da distorsione.
- Molti autori propongono di valutare più trasformazioni candidate per l'allineamento e poi:
 1. selezionare la trasformazione finale sulla base dello score più alto ottenuto nell'ultimo passo di abbinamento.
 2. limitare il matching globale alle regioni più vicine alla coppia di riferimento e/o fondere i risultati di più registrazioni.

La distorsione della pelle

Uno dei fattori principali che contribuiscono a rendere **molto diverse** due immagini della stessa impronta è la **distorsione della pelle**



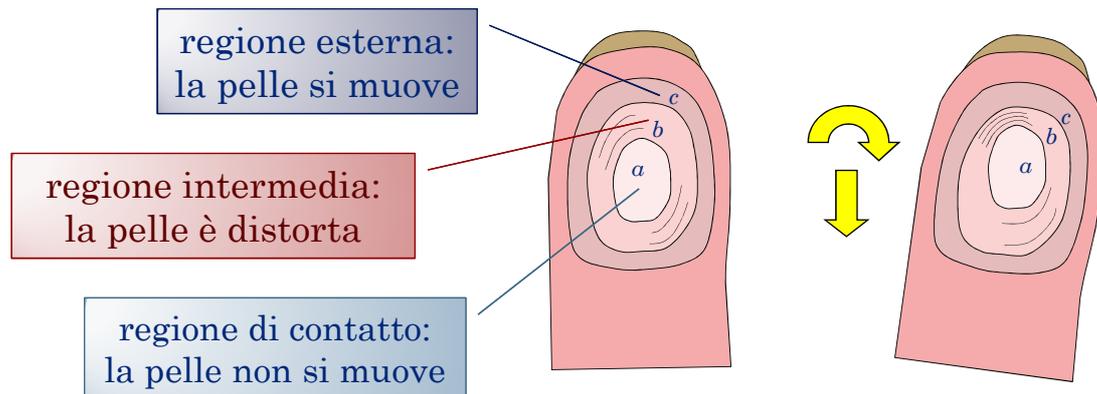
Gestire la distorsione (1)

- Range di tolleranza
 - La distorsione è gestita rilassando le relazioni spaziali tra le minuzie; in particolare i parametri r_0 and θ_0 visti nella formulazione del problema di matching sono incrementati rendendo i vincoli per il matching meno stringenti.
- Warping
 - Il problema della distorsione è affrontato in modo diretto, sottoponendo una delle due impronte a un processo di warping con l'obiettivo di massimizzare il numero di minuzie accoppiate.

Gestire la distorsione (2)

- Allineamenti multipli e clustering
 - Quando si confrontano due impronte distorte con un approccio globale, allineamenti diversi determinano corrispondenze tra coppie di minuzie diverse. Restringere il matching a sottoinsiemi (**cluster**) di minuzie allevia il problema della distorsione mantenendo al tempo stesso il potere discriminante degli approcci globali.
- Normalizzazione
 - L'idea di base è quella di operare a priori una rimozione della distorsione dalle immagini, per permettere anche a metodi che utilizzano un allineamento rigido di operare correttamente anche in presenza di distorsione.

Un modello di distorsione (1)



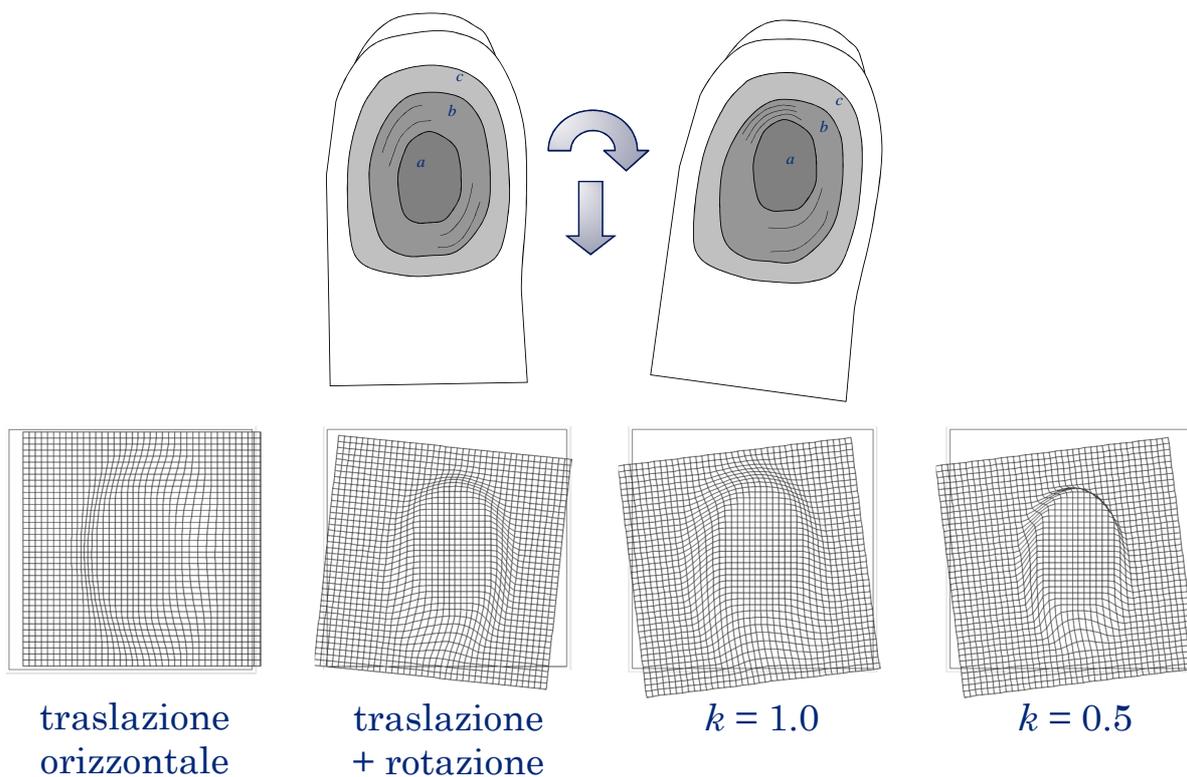
- regione di contatto: l'elevata pressione e l'attrito del sensore non permettono alcuno spostamento della pelle
- regione intermedia: regione di transizione in cui si produce una distorsione della pelle per raccordare la regione a e la regione c
- regione esterna: la bassa pressione permette alla pelle di essere trascinata dal movimento del dito

Un modello di distorsione (2)

- Il modello di distorsione è definito da un **mapping** $\mathcal{R}^2 \rightarrow \mathcal{R}^2$ che può essere visto come una trasformazione affine senza cambiamento di scala che si riduce gradualmente quando ci si sposta da c verso a .
- Ogni punto \mathbf{v} è mappato in $distortion(\mathbf{v})$:

$$distortion(\mathbf{v}) = \mathbf{v} + \Delta(\mathbf{v}) \cdot brake(shapedist_a(\mathbf{v}), k)$$
 - $\Delta(\cdot)$ specifica la trasformazione affine di un punto della regione esterna c ;
 - $shapedist(\cdot)$ è una funzione "forma" che descrive il confine della regione a ;
 - $brake(\cdot)$ è una funzione monotona crescente che regola la transizione graduale dalla regione a alla regione c ;
 - il parametro k regola la plasticità della pelle.

Un modello di distorsione (3)



Un modello di distorsione (4)

