



Fondamenti di Elaborazione di Immagini

Template Matching

Raffaele Cappelli
raffaele.cappelli@unibo.it

Contenuti

■ Introduzione

- Confronto diretto fra immagini e template matching
- Panoramica approcci di template matching

■ Correlazione

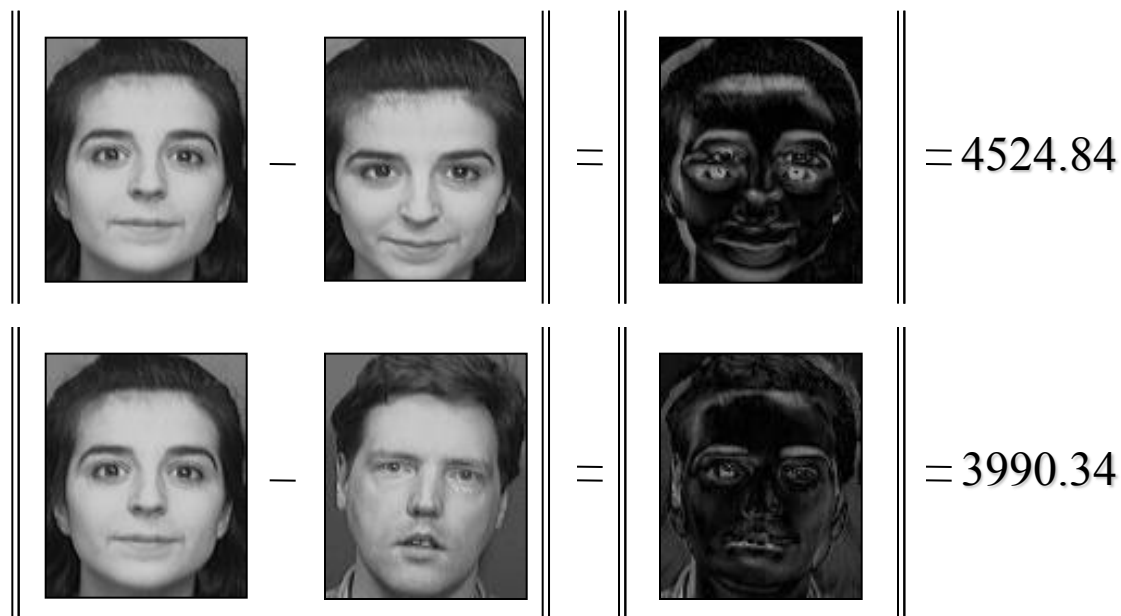
- Definizione
- Misure di correlazione normalizzate
- Analisi della complessità computazionale
- Approccio multi-risoluzione
- Alcuni esempi pratici

■ Tecniche avanzate di template matching

- Cenni ad argomenti trattati in Visione Artificiale e Riconoscimento

Confronto diretto fra immagini

- Ogni immagine potrebbe essere semplicemente considerata come un punto in uno spazio multidimensionale
 - Un'immagine grayscale corrisponderebbe quindi a un vettore con $n=W \times H$ dimensioni
- Il **confronto diretto di immagini**, mediante distanza fra tali vettori nello spazio, in generale **non funziona** per una serie di ragioni, fra cui:
 - Differenze di traslazione, rotazione, scala e prospettiva
 - Deformazione e variabilità dei pattern
 - Cambiamenti di illuminazione
 - Presenza di rumore nelle immagini e utilizzo di tecniche di acquisizione diverse

$$\begin{array}{c} \left\| \begin{array}{c} \text{img}_1 - \text{img}_2 \end{array} \right\| = \left\| \begin{array}{c} \text{img}_{\text{diff}} \end{array} \right\| = 4524.84 \\ \left\| \begin{array}{c} \text{img}_3 - \text{img}_4 \end{array} \right\| = \left\| \begin{array}{c} \text{img}_{\text{diff}} \end{array} \right\| = 3990.34 \end{array}$$


Template matching

- Anziché tentare di confrontare direttamente due immagini, si costruiscono **uno o più pattern modello (template)** e li si “**ricerca**” all’interno dell’immagine, misurandone il grado di “**somiglianza**” (**matching**) **in tutte le possibili posizioni**



I

T₁T₂

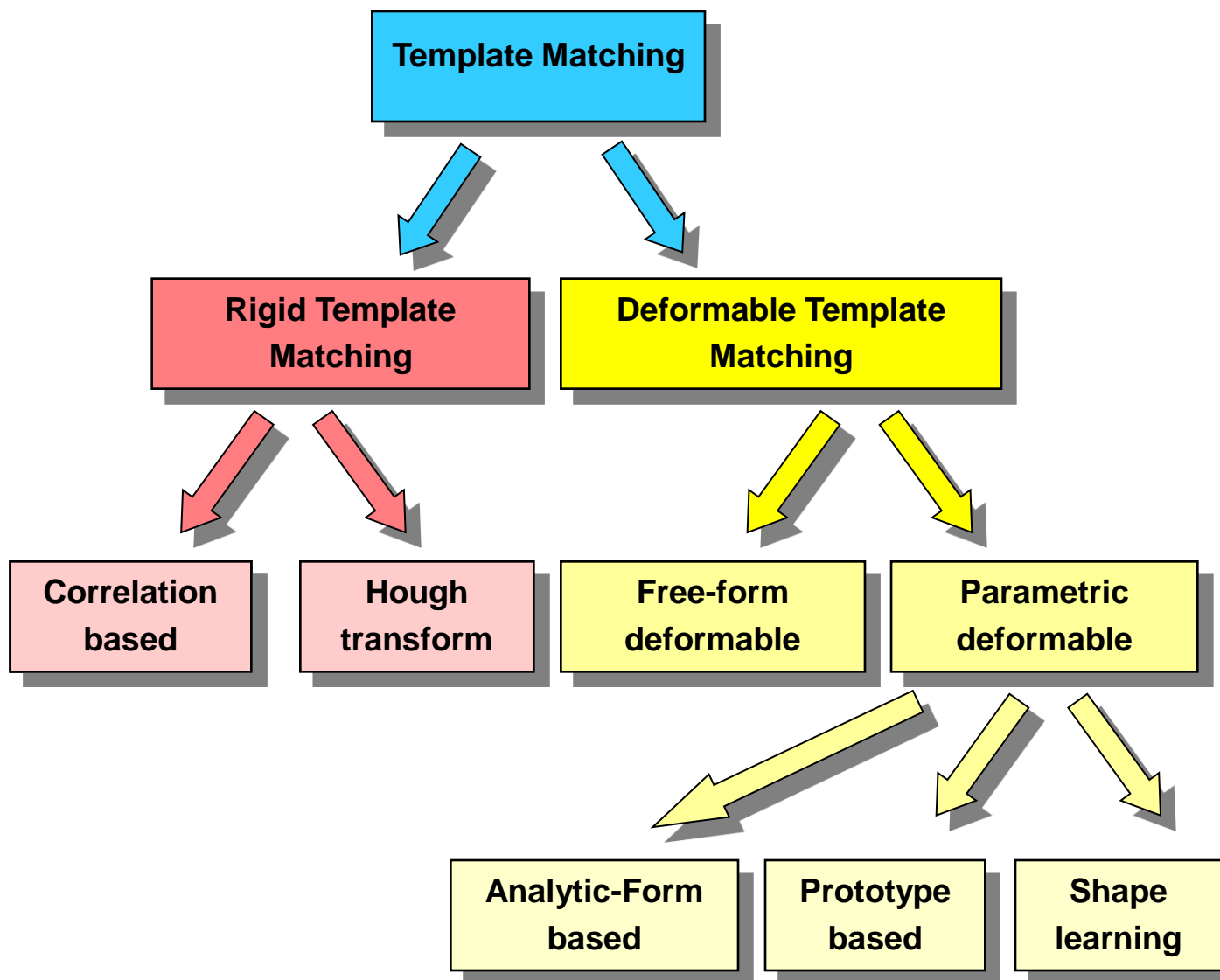
2 template

Immagine di riferimento

la posizione (ideale) di massima
somiglianza per i due template
all'interno dell'immagine



Template matching – Panoramica approcci



Template matching “rigido”

- Il template T è costituito da un oggetto rigido
 - Normalmente una piccola immagine in formato raster: il **confronto** avviene **direttamente fra i pixel**; tuttavia, a seconda dell'applicazione, *potrebbe essere più efficace eseguire il template matching dopo aver estratto determinate feature* (es. gli edge, oppure l'orientazione del gradiente)
- T viene sovrapposto a I in tutte le possibili posizioni (rispetto agli assi x e y), ma, a seconda dell'applicazione, può essere anche necessario operare delle trasformazioni (ad es. ruotarlo e/o scalarlo).
 - Nel seguito si indicherà con T_i una generica istanza di T ottenuta con una data trasformazione.
- Per ogni istanza T_i , il grado di similarità viene solitamente calcolato massimizzando la **correlazione** con la porzione di immagine I “coperta” da T_i .
 - Normalmente i template hanno dimensione inferiore all'immagine
 - Analogamente alla convoluzione, è necessario decidere come comportarsi sui pixel di bordo (ossia dove T_i non è completamente contenuto in I)

Correlazione

■ Definizioni

- Data un'immagine I e un'istanza T_i del template, sia I_{xy} la porzione di immagine con le stesse dimensioni di T_i centrata nel pixel (x,y)
- Nel seguito, per semplicità di notazione, I_{xy} e T_i indicheranno i **vettori** ottenuti *concatenando le righe* delle due immagini rettangolari corrispondenti

■ SSD e CC

- Una misura intuitiva di diversità fra I_{xy} e T_i è la **Sum of Squared Differences (SSD)**:

$$SSD(I_{xy}, T_i) = \|I_{xy} - T_i\|^2 = (I_{xy} - T_i)^T (I_{xy} - T_i) = \|I_{xy}\|^2 + \|T_i\|^2 - 2I_{xy}^T T_i$$

- se le norme di I_{xy} e T_i fossero costanti, *minimizzare* la SSD equivarrebbe a *massimizzare* la Cross Correlation (CC):

$$CC(I_{xy}, T_i) = I_{xy}^T T_i = \sum_k T_i[k] \cdot I_{xy}[k]$$

Correlazione (2)

■ Misure di correlazione normalizzate

□ Necessarie, quando I_{xy} e T_i non sono costanti:

- Diverse regioni della stessa immagine raramente lo sono
- Istanze dello stesso template diverse tra loro per numero di pixel e luminosità media

■ *Normalized SSD (NSSD) e Normalized CC (NCC)*

□ Sono *indipendenti dal contrasto* di immagine e template.

Infatti pattern a più elevato contrasto (caratterizzati da ampio range di livelli di grigio) vengono ritenuti dalla semplice *SSD* più dissimili rispetto a pattern con scarso contrasto.

$$NSSD(\mathbf{I}_{xy}, \mathbf{T}_i) = \frac{\|\mathbf{I}_{xy} - \mathbf{T}_i\|^2}{\|\mathbf{I}_{xy}\| \cdot \|\mathbf{T}_i\|}$$

$$NCC(\mathbf{I}_{xy}, \mathbf{T}_i) = \frac{\mathbf{I}_{xy}^T \mathbf{T}_i}{\|\mathbf{I}_{xy}\| \cdot \|\mathbf{T}_i\|}$$

■ *Zero-mean NSSD (ZNSSD) e Zero-mean NCC (ZNCC)*

□ Rispetto a *NSSD* e *NCC* sono invarianti anche per pattern che, a parità di contrasto (stesso range dinamico), presentano *luminosità medie* diverse.

$$ZNSSD(\mathbf{I}_{xy}, \mathbf{T}_i) = NSSD(\mathbf{I}_{xy} - \bar{\mathbf{I}}_{xy}, \mathbf{T}_i - \bar{\mathbf{T}}_i)$$

$$ZNCC(\mathbf{I}_{xy}, \mathbf{T}_i) = NCC(\mathbf{I}_{xy} - \bar{\mathbf{I}}_{xy}, \mathbf{T}_i - \bar{\mathbf{T}}_i)$$

Correlazione – Implementazione di base

```
[...]
if (T.Width * T.Height > int.MaxValue / (255 * 255))
    throw new Exception("Template troppo grande per calcolo intero a 32 bit");
[...]
res = new Image<int>(I.Width, I.Height);
int tw = T.Width;
int th = T.Height;
int mw2 = T.Width / 2;
int mh2 = T.Height / 2;
int y1 = mh2;
int y2 = I.Height - mh2 - 1;
int x1 = mw2;
int x2 = I.Width - mw2 - 1;
// Esegue la correlazione (i bordi restano inizializzati a zero)
for (int y = y1; y <= y2; y++)
    for (int x = x1; x <= x2; x++)
    {
        int val = 0;
        for (int yt=0; yt<th; yt++)
            for (int xt=0; xt<tw; xt++)
                val += (int)I[y-mh2+yt, x-mw2+xt] * T[yt, xt];
        res[y, x] = val;
    }
```

N.B. Un'implementazione più efficiente dovrebbe evitare l'accesso ai pixel con [y,x] (utilizzando indici lineari) e cercare di evitare calcoli inutili, ad esempio precalcolando gli offset del template.

Complessità computazionale

■ Un esempio: riconoscimento di targhe

□ Immagine: 512x256



□ Template: 10 cifre e 26 caratteri

- Risoluzione 13x20 pixel.
- Per ogni cifra e per ogni carattere consideriamo 9 istanze dovute a variazioni di scala (diverse distanze dalla telecamera) e rotazione.
- Ogni istanza (324 istanze = 36×9) deve essere sovrapposta all'immagine in tutte le possibili posizioni e genera quindi ulteriori 512x256 istanze (se si trascurano i bordi).
- Pertanto occorre stimare circa $42.467.328 = 324 \times 512 \times 256$ correlazioni, ciascuna richiedente almeno 13x20 moltiplicazioni e altrettante somme (nel caso di semplice CC). In totale circa 11×10^9 moltiplicazioni (interi) e altrettante somme.



□ Quanto tempo occorre per processare un'immagine?

- Almeno 44 secondi su una macchina capace di eseguire 500 milioni di operazioni intere al secondo.

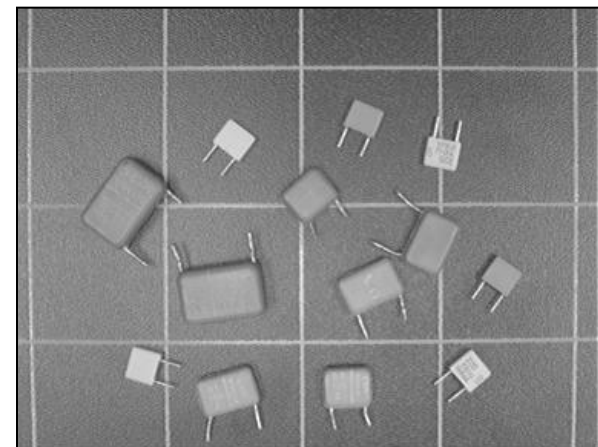
Correlazione – Problematiche

■ Complessità computazionale

- Il numero di operazioni richieste cresce linearmente con il numero di istanze e con il numero di pixel di I e T (e quindi quadraticamente rispetto al lato di I e di T).
- In pratica, per applicazioni real-time, l'approccio di base è raramente applicabile.

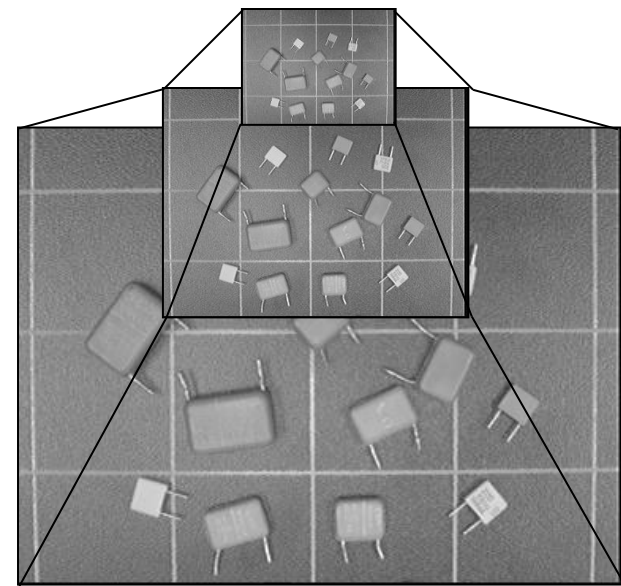
■ Difficile gestione di pattern deformabili

- Esempio: localizzazione di un componente elettronico
 - Necessaria invarianza per posizione, rotazione, scala e aspect ratio (lunghezza/altezza). Ciò implica un enorme numero di istanze!
 - Come gestire le variazioni di colore? (si potrebbe operare sugli edge e non sui pixel)
 - Come gestire le deformazioni locali (ad esempio i piedini piegati o di diversa lunghezza)?



Correlazione: approccio multi-risoluzione

- Obiettivo: ridurre la complessità computazionale
 - Si segue la ricerca su una gerarchia crescente di risoluzioni
 - Viene creata una “piramide” di risoluzioni sia per I che per T (ad esempio dimezzando la risoluzione ad ogni livello)
 - La ricerca viene eseguita inizialmente sulla risoluzione più bassa, e ai livelli successivi vengono analizzate solo le istanze promettenti (la cui correlazione al livello inferiore eccedeva una data soglia)



- Vantaggi
 - Consente di eseguire una “scrematura” ai livelli iniziali e di perfezionare la localizzazione e filtrare “false somiglianze” ai livelli successivi
 - Esemplificando:
 - A metà risoluzione le operazioni si riducono di 16 volte.
 - A 1/4 quarto di risoluzione di 256 volte.
 - A $1/2^n$ di risoluzione di 2^{4n} volte, ma tipicamente oltre a 3, 4 livelli non è possibile operare per mancanza di dettagli

Approccio multi-risoluzione – Esempio

- Si consideri il caso del riconoscimento di targhe visto in precedenza (immagine 512x256, 36x9 istanze di template)
 - L'applicazione della ZNCC (con un'implementazione mediamente efficiente in C#) richiede **70.8 secondi** su una CPU a 2.0GHz
 - Applicando l'approccio multirisoluzione con 2 livelli e impostando la soglia in modo da considerare al secondo livello circa l'1% delle correlazioni più promettenti del primo livello, i tempi risultano essere:
 - **4.3 secondi** per la correlazione al primo livello (come era logico fosse!)
 - **1.5 secondi** per la correlazione al secondo livello (dimensioni originali)
 - **0.3 secondi** per selezionare le istanze da analizzare al secondo livello



Le posizioni selezionate al primo livello per una particolare istanza di un template



Tecniche avanzate di template matching

■ Template matching rigido

□ Correlazione nel dominio delle frequenze

- Analogamente al caso della convoluzione, è possibile utilizzare la trasformata di Fourier per calcolare la correlazione in maniera più efficiente [→ VAR]

□ Trasformata di Hough

- Nella sua versione originale, si tratta di un metodo piuttosto robusto per individuare linee rette in un'immagine [→ VAR]
- Può essere generalizzata per ricercare figure di forma arbitraria in un'immagine (es. circonferenze, ellissi, ...)

Tecniche avanzate di template matching (2)

■ Template matching deformabile

□ Free-form deformable

- Il template non è vincolato a forme precise. Si fa uso di un potential field, cioè di una funzione di energia prodotta dalle feature salienti dell'immagine, per guidare il processo verso le deformazioni maggiormente significative. [→VAR]

□ Parametric deformable

- Il template è parametrico (ad esempio formato da archi, curve spline, ...) e regolato da un numero limitato di parametri, agendo sui quali si ottengono deformazioni controllate. [→VAR]

□ Shape learning

- Le possibili variazioni della forma del template (e i corrispondenti gradi di libertà) vengono “appresi” in modo automatico a partire da un insieme di esempi rappresentativi delle possibili variazioni dell'oggetto che il template deve rappresentare. [→VAR]