

# Elaborazione e Analisi delle Immagini

Giuseppe Bellisano

30 novembre 2016

# Indice

<b>1</b>	<b>Image segmentation</b>	<b>2</b>
1.1	Cosa è l'immagine segmentation? . . . . .	2
1.2	Quanti tipi di segmentazione abbiamo? . . . . .	2
1.3	In quali campi viene utilizzata la segmentazione? . . . . .	2
1.4	Quali sono i metodi utilizzati per la segmentazione? . . . . .	3
1.5	Cosa si intende con Approccio Globale: Gray level thresholding?	3
1.6	Quale è l'idea alla base del gray level thresholding con due elementi (mode) da separare? . . . . .	3
1.7	Quale è l'idea alla base del gray level thresholding con tre elementi (mode) da separare? . . . . .	4
1.8	Nel gray level thresholding quanti e quali tipi di soglia possiamo avere? . . . . .	4
1.9	Il thresholding applicato ad immagini con rumore e variazioni di luminosità: cosa cambia? . . . . .	4
1.10	Cosa è il single global threshold e quando si applica? . . . . .	5
1.11	Quali sono le caratteristiche del single global threshold? . . . . .	5
1.12	Qual è l'algoritmo Matlab per il global thresholding? . . . . .	5
1.13	Cosa è l'Optimum global thresholding o metodo Otsu? . . . . .	6
1.14	Come funziona il metodo Otsu? . . . . .	6
1.15	Spiegare le sogliature multiple . . . . .	8
1.16	Illustra il metodo Otsu in Matlab . . . . .	9
1.17	Tecniche per migliorare il global thresholding . . . . .	9
<b>2</b>	<b>Segmentazione</b>	<b>10</b>
2.1	Formulazione matematiche di base . . . . .	10
<b>3</b>	<b>Image processing con Matlab</b>	<b>11</b>
3.1	Ridimensionamento . . . . .	11
3.2	Trasformazioni geometriche . . . . .	11
3.3	Rumore . . . . .	12
3.4	Smoothing con thresholding . . . . .	12

# Capitolo 1

## Image segmentation

### 1.1 Cosa è l'image segmentation?

La segmentazione è uno dei passi più importanti nell'analisi delle immagini processate

Il suo scopo è quello di **dividere** l'immagine in parti che hanno una forte correlazione con oggetti o aree che si trovano nel mondo reale.

### 1.2 Quanti tipi di segmentazione abbiamo?

Abbiamo due tipi di segmentazione:

1. **Segmentazione Completa** in cui le regioni corrispondono direttamente con gli oggetti dell'immagine in ingresso. In pratica ogni elemento dell'immagine viene riconosciuto e isolato. Per ottenere questo risultato è necessario avere informazioni di processing di alto livello (high level processing) da utilizzare per comprendere il dominio del problema da risolvere (che ci permette di riconoscere gli oggetti/regioni)
2. **Segmentazione Parziale** in cui le regioni/segmenti di immagini non corrispondono direttamente con gli oggetti reali rappresentati nell'immagine. In questo caso l'immagine è divisa in regioni omogenee secondo una determinata proprietà scelta (colore, texture, riflettività).

La segmentazione parziale è seguita da un ulteriori processi sino ad arrivare all'immagine finale di segmentazione che può essere trovata grazie all'aiuto di informazioni di alto livello.

### 1.3 In quali campi viene utilizzata la segmentazione?

Ecco alcune delle problematiche più comuni che possono essere risolte con la segmentazione:

- ottenere dati da immagini ambigue

- rumore nelle informazioni
- ottenere oggetti con discreto contrasto su di uno sfondo uniforme
- semplici lavori di raggruppamento come cellule del sangue, caratteri stampati

In generale è difficile ottenere una corretta e completa segmentazione; per questo spesso si utilizza una segmentazione parziale come input in un processo di alto livello.

## 1.4 Quali sono i metodi utilizzati per la segmentazione?

- Thresholding (Approccio Globale) che si basa sull'istogramma di alcune caratteristiche dell'immagine come il **gray level thresholding** o **soglia del livello di grigio**. Si ricercano i toni soglia che separano oggetti differenti nell'immagine. E' il metodo più semplice e veloce
- Segmentazione basata sugli Edge in cui si cercano i contorni degli oggetti basandosi sulla discontinuità dei toni, cioè sugli edge
- Segmentazione basata sulle Regioni in cui si ricercano le regioni, cioè degli oggetti uniformi secondo un certo criterio (stesso tono di grigio, o tono di grigio che si differenzia al massimo di una certa soglia)
  - region growing
  - region splitting and merging

## 1.5 Cosa si intende con Approccio Globale: Gray level thresholding?

E' il sistema più semplice a livello concettuale di segmentazione e per questo è anche il sistema più efficiente computazionalmente e veloce.

Esso viene applicato alle immagini che contengono regioni o oggetti con un costante livello di riflettività o assorbimento di luce. Si sfrutta questa caratteristica per partizionare l'immagine grazie ai valori di intensità e/o alle proprietà di questi valori.

## 1.6 Quale è l'idea alla base del gray level thresholding con due elementi (mode) da separare?

1. Si supponga di avere un istogramma di intensità di una immagine  $f(x,y)$  che è composta da un oggetto bianco (level 255) e di uno sfondo scuro (level 0) e che si voglia estrarre l'oggetto dallo sfondo.

2. Inizialmente si seleziona una soglia  $T$  con cui si ottiene l'immagine segmentata  $g(x, y)$ :

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > T \\ 0 & \text{se } f(x, y) \leq T \end{cases}$$

3. Se il valore di intensità dell'immagine è superiore a quello della soglia, si assegna a  $g(x, y)$  il valore 1, altrimenti 0

### 1.7 Quale è l'idea alla base del gray level thresholding con tre elementi (mode) da separare?

- Maggiore è il numero di elementi (mode), più complessa sarà la segmentazione. Esaminiamo il caso di una immagine con due oggetti chiari posti su uno sfondo scuro.
- L'immagine segmentata risultante  $g(x, y)$  sarà:

$$g(x, y) = \begin{cases} a & \text{se } f(x, y) > T_2 \\ b & \text{se } T_1 < f(x, y) \leq T_2 \\ c & \text{se } f(x, y) \leq T_1 \end{cases}$$

dove  $a$ ,  $b$ ,  $c$  sono tre diversi valori di intensità e  $T_1$  e  $T_2$  sono due differenti valori di soglia.

### 1.8 Nel gray level thresholding quanti e quali tipi di soglia possiamo avere?

Possiamo avere due tipi di soglia:

1. Soglia Globale
2. Soglia Variabile, che si divide in soglia locale o regionale

### 1.9 Il thresholding applicato ad immagini con rumore e variazioni di luminosità: cosa cambia?

Esaminando l'istogramma dell'immagine si nota che questo subisce delle variazioni a seconda che l'immagine si affetta da rumore o una intensità variabile.

### 1.10 Cosa è il single global threshold e quando si applica?

Il single global threshold si applica a quelle immagini in cui l'intensità degli oggetti e dello sfondo rende questi facilmente distinti. L'algoritmo che si applica all'intera immagine si basa sui seguenti passi:

1. Si seleziona una soglia  $T$
2. Si effettua la segmentazione dell'immagine usando la soglia  $T$  producendo due gruppi di pixel:
  - $G_1$  dato da tutti i pixel che hanno una intensità  $\geq T$
  - $G_2$  dato da tutti i pixel che hanno una intensità  $< T$
3. Si calcolano i valori di **intensità medi**  $m_1$  e  $m_2$  rispettivamente per i gruppi di pixel  $G_1$  e  $G_2$
4. Si determina un nuovo livello di soglia  $T$  secondo la formula:

$$T = \frac{1}{2}(m_1 + m_2)$$

5. Si ripetono i passi dal 2 al 4 sino a quando la differenza tra i valori di  $T$  (delle successive iterazioni) è più piccolo rispetto al valore predefinito  $T_0$

### 1.11 Quali sono le caratteristiche del single global threshold?

- E' un algoritmo semplice.
- Funziona bene nei casi in cui l'istogramma dell'immagine presenta delle campane (le mode) corrispondenti agli oggetti e allo sfondo separate da una valle.
- Il valore di soglia iniziale deve essere maggiore rispetto al minimo valore di intensità e minore rispetto al massimo valore di intensità nell'immagine.

### 1.12 Qual è l'algoritmo Matlab per il global thresholding?

Indicando con:

- $f$  per l'immagine di input
- $g$  per l'immagine di output
- $T$  per la soglia

```

1 function [g]=iter_thresh(f)
2 T = 0.5 * (double(min(f(:)))+ double(max(f(:))));
3 flag = false;
4 while ~flag
5     g = f >= T;
6     Tnext = 0.5 * (mean(f(g)) + mean(f(~g)));
7     flag = abs (T - Tnext) < 0.5;
8     T = Tnext;
9 end

```

### 1.13 Cosa è l'Optimum global thresholding o metodo Otsu?

Meglio conosciuto come Metodo Otsu è un è un metodo di sogliatura automatica dell'istogramma nelle immagini digitali.

Viene definito come "ottimo" perché **massimizza la varianza tra le classi**. Un algoritmo di sogliatura (threshold) che fornisce la migliore separazione tra le classi in termini dei loro valori di intensità è il miglior threshold.

### 1.14 Come funziona il metodo Otsu?

- Si considerino:
  - $\{0, 1, 2, \dots, L-1\}$  siano gli  $L$  distinti livelli di intensità in una immagine
  - $M \times N$  sia la dimensione dell'immagine di  $M$  righe X  $N$  colonne
  - $n_i$  indica il numero di pixel con intensità  $i$
  - $MN = n_0 + n_1 + n_2 + \dots + n_{L-1}$
- l'istogramma normalizzato ha componenti  $p_i = \frac{n_i}{MN}$  da cui segue:

$$\sum_{i=0}^{L-1} p_i = 1 \quad p_i \geq 0$$

- Selezionando e utilizzando una soglia  $T(k) = k$  con  $0 < k < L - 1$  con l-immagine di input, si ottengono le due classi  $c_1$  e  $C_2$  dove:
  - $C_1$  è la classe di tutti i pixel con valori di intensità compresi nel range  $[0, k]$
  - $C_2$  è la classe di tutti i pixel con valori di intensità compresi nel range  $[k + 1, L - 1]$
- Usando questa soglia, si può calcolare la *probabilità*  $P_1(k)$  che un pixel appartenga alla classe  $C_1$ :

$$P_1(k) = \sum_{i=0}^k p_i$$

- La *probabilità* che il pixel invece appartenga alla classe  $C_2$  è data da:

$$P_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - P_1(k)$$

- Il valore di *intensità medio* di tutti i pixel della classe  $C_1$  è:

$$m_1(k) = \sum_{i=0}^k iP\left(\frac{i}{C_1}\right) = \sum_{i=0}^k iP\left(\frac{C_1}{i}\right) \frac{P(i)}{P(C_1)} = \frac{1}{P_1(k)} \sum_{i=0}^k ip_i$$

- Il valore di *intensità medio* di tutti i pixel della classe  $C_2$  è invece:

$$m_2(k) = \sum_{i=k+1}^{L-1} iP\left(\frac{i}{C_2}\right) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} ip_i$$

- La *media complessiva* sino al livello  $k$  è data da:

$$m(k) = \sum_{i=0}^k ip_i$$

- Invece l'*intensità media* dell'intera immagine (**la media globale**) è data da:

$$m_G = \sum_{i=0}^{L-1} ip_i$$

- Inoltre valgono le seguenti:

$$P_1 m_1 + P_2 m_2 = m_G$$

$$P_1 + P_2 = 1$$

- Per valutare la bontà della soglia al livello  $k$  si usa la *metrica normalizzata*:

$$\eta = \frac{\sigma_B^2}{\sigma_G^2}$$

- dove  $\sigma_G^2$  è la **varianza globale** ed è *costante*:

$$\sigma_G^2 = \sum_{i=0}^{L-1} (i - m_G)^2 p_i$$

- dove  $\sigma_B^2$  è la **varianza tra le classi** che indica la misura della *separabilità* tra le classi:

$$\sigma_B^2 = P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2 = P_1 P_2 (m_1 - m_2)^2 = \frac{(m_G P_1 - m)^2}{P_1(1 - P_1)}$$



- (The farther the two means  $m_1$  and  $m_2$  are from each other the larger will be.)
- Poiché  $\sigma_G^2$  è una costante ne consegue che  $\eta$  è una *misura della separabilità*. Inoltre massimizzare questa metrica, è identico a massimizzare  $\sigma_B^2$
- l'obiettivo principale è determinare il valore  $k$  di soglia che massimizza la varianza di classe.
- Introducendo nuovamente  $k$  si ha:

$$\eta(k) = \frac{\sigma_B^2}{\sigma_G^2}$$

e

$$\sigma_B^2(k) = \frac{[m_g P_1(k - m(k))]^2}{P_1(k)[1 - P_1(k)]}$$

- La **soglia ottimale** è il valore  $k^*$  che massimizza  $\sigma_B^2(k)$ :

$$\sigma_B^2(k^*) = \max_{0 \leq k \leq L-1} \sigma_B^2(k)$$

- Per trovare  $K^*$  si deve valutare questa equazione per tutti i valori interi di  $k$  e scegliere il valore che produce il massimo  $\sigma_B^2(k)$
- Se il massimo esiste *per più di un valore di  $k$*  è consuetudine mediare i diversi valori di  $k$  per cui  $\sigma_B^2(k)$  è massima.
- La metrica normalizzata  $\eta$  valutata per il valore di soglia ottimale  $\eta(k^*)$  può essere utilizzata per ottenere una stima quantitativa della separabilità delle classi, che a sua volta fornisce un'idea della facilità di sogliatura di una data immagine. Il suo valore oscilla tra 0 e 1:
  - il valore più basso è ottenibile solo da immagini con un singolo livello di intensità costante
  - Il limite superiore invece è ottenibile solo dalle immagini 2-valori con intensità uguali a 0 e  $L - 1$

## 1.15 Spiegare le sogliature multiple

Il metodo di Otsu può essere esteso ad un numero arbitrario di soglie in modo da valutare un diverso numero di classi.

Generalmente i metodi che utilizzano più di due soglie vengono risolte con più valori di intensità (colore).

Un altro metodo è quello di suddividere l'immagine in *rettangoli sovrapposti*

- questo sistema viene utilizzato per compensare i casi di illuminazione non uniforme
- i rettangoli vengono scelti di piccole dimensioni in modo che l'illuminazione da essi coperta, sia uniforme

## 1.16 Illustra il metodo Otsu in Matlab

La funzione *graythresh* di Matlab permette di calcolare una soglia utilizzando il metodo di Otsu.

## 1.17 Tecniche per migliorare il global thresholding

- Applicare lo smoothing all'immagine prima del thresholding: è una tecnica che si applica quando il rumore non può essere ridotto e come metodo di sogliatura si seleziona il global thresholding. Di seguito un esempio di codice Matlab.

```
1 f=imread('large_septagon_gaussian_noise_mean_0_std_50_added.  
   tif');  
   figure , imshow(f);  
3 figure , imhist(f);  
   T=graythresh(f);  
5 T=T*255  
   g=f>=T;  
7 figure , imshow(g);  
   filt=fspecial('average', [5 5]);  
9 ff=imfilter(f, filt);  
   figure , imshow(ff);  
11 figure , imhist(ff);  
   ylim([0 16000])  
13 T1=graythresh(ff);  
   T1=T1*255  
15 g1=ff>=T1;  
   figure , imshow(g1);
```

- Thresholding variabile attraverso il partizionamento dell'immagine. Di seguito un esempio di codice Matlab.

## Capitolo 2

# Segmentazione

### 2.1 Formulazione matematiche di base

- $R$  rappresenta l'intera regione dell'immagine
- la segmentazione è il processo che divide la regione  $R$  nelle sottoregioni  $R_1, R_2, \dots, R_n$  per cui vale
  1.  $\bigcup_{i=1}^n R_i = R$  ovvero la somma di tutte le sottoregioni è uguale all'intera regione  $R$  dell'immagine
  2.  $R_i$  è un insieme connesso dove  $i = 1, 2, \dots, n$
  3.  $R_i \cap R_j = \emptyset \quad \forall i, j \quad i \neq j$
  4.  $Q(R_i) = TRUE \quad i = 1, 2, \dots, n$
  5.  $Q(R_i \cup R_j) = FALSE$  per ogni regione adiacente  $R_i$  e  $R_j$

## Capitolo 3

# Image processing con Matlab

### 3.1 Ridimensionamento

**comando:** `imresize`

**possibili domande:**

- Leggere un'immagine e rimpicciolirla o farne lo zoom di un valore intero
- Leggere un'immagine e rimpicciolirla o farne lo zoom di un valore decimale con interpolazione lineare

### 3.2 Trasformazioni geometriche

In Matlab è possibile effettuare una trasformazione geometrica affine specificando la matrice di trasformazione  $T$  attraverso il comando *maketform*. Per effettuare la trasformazione si usa il comando *imtransform*.

**comando:** `maketform`, `imtransform`, `imrotate`, `impixelinfo` (`pixval`),

**possibili domande:**

- Effettuare la rotazione di un'immagine qualsiasi
- Confrontare il risultato con quello ottenuto mediante la funzione *imrotate*
- Leggere l'immagine di 'lena' e realizzare l'ingrandimento di una zona dell'immagine usando la matrice di trasformazione  $T$  la sezione da ingrandire è intorno all'occhio di lena. Per individuare la sezione e, quindi, avere informazioni sulla posizione dei pixel potete usare il comando *impixelinfo* o *pixval* (in base alla versione di Matlab più o meno recente).
- Fare degli esperimenti modificando il tipo di interpolazione e notate l'effetto di blocchettatura causa
- La combinazione di diverse trasformazioni affini è ancora una trasformazione affine, che può essere ottenuta tramite il prodotto (matriciale) delle matrici che le definiscono.

- Scrivere una funzione dal prototipo `function g=rot_dist(f, alfa, c)` per realizzare prima una rotazione e poi una distorsione verticale.
- Creare l'immagine di ingresso usando il seguente comando `f = checkerboard(40);` in modo da generare una scacchiera su cui le modifiche risultano essere più facilmente visibili.

### 3.3 Rumore

- Aggiungere del rumore gaussiano bianco ad un'immagine `f` con il comando `noisy = f + n` con `n = d*randn(size(f))` dove **d** è la **deviazione standard del rumore**.
- Rimuovere il rumore dall'immagine con i filtri a media mobile (al variare della dimensione della finestra).
- Valutare l'efficacia del filtraggio sia visivamente sia calcolando l'errore quadratico medio tra `f` e l'immagine "ripulita".
- L'errore quadratico medio rappresenta una misura quantitativa per stabilire quanto l'immagine elaborata sia simile all'originale.
- L'MSE (Mean Squared Error) tra due immagini si definisce come:

$$MSE = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} |f(m, n) - g(m, n)|^2$$

### 3.4 Smoothing con thresholding

- Consideriamo l'immagine 'telescopio.jpg', proveniente dal telescopio Hubble, in orbita intorno alla terra. Rilevare gli oggetti grandi realizzando le seguenti operazioni:
  - Visualizzare l'immagine
  - Applicare il filtro che effettua la media aritmetica su una finestra di dimensioni 15x15 e visualizzare il risultato
  - Applicare un'operazione a soglia per eliminare gli oggetti piccoli (considerare una soglia pari al 25 per cento del valore massimo presente nell'immagine filtrata)
  - Visualizzare il risultato dell'elaborazione

# Indice analitico

Image segmentation, 1

Otsu, 5, 8

# Glossario

**connesso** pulsazione naturale.

**trasformazione geometrica affine** Esempi di affinità sono rotazioni, omote-  
tie, traslazioni, rototraslazioni, riflessioni. Le affinità non sono necessaria-  
mente isometrie, non preservano cioè angoli e distanze, mentre mantengono  
sempre il parallelismo tra le rette..