

Segmentazione per regioni

1 Segmentazione per regioni

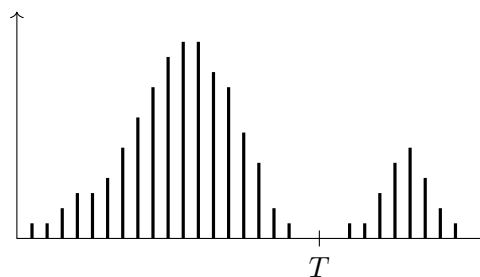
La segmentazione basata sull'edge detection lavora solo in modo indiretto sulle regioni connesse, in quanto considera le discontinuità tra una regione omogenea e lo sfondo (o un'altra regione). Di conseguenza, se l'interno di un oggetto non è sufficientemente omogeneo (ad esempio, a causa di rumore, illuminazione non uniforme, ecc.), si introducono dei *falsi contorni*, che rendono difficile/impossibile la segmentazione.

In questi casi, è più indicato l'uso di tecniche di segmentazione basate sulla determinazione di regioni.

2 Thresholding

Il **thresholding** (sogliatura) è un insieme di tecniche basate sull'omogeneità dei livelli di grigio all'interno delle regioni. Siccome non considerano la distribuzione spaziale dei valori di grigio, queste tecniche non danno garanzie di continuità spaziale delle regioni individuate.

Lo strumento fondamentale per la segmentazione mediante sogliatura è l'istogramma. Infatti, se oggetti e sfondo hanno livelli di grigio distinti, nell'istogramma si formano due mode ben distinte.



Allora, l'immagine può essere segmentata ponendo una soglia T che separi le due mode: i pixel con valore $f(x, y) > T$ appartengono agli oggetti, mentre gli altri allo sfondo (o viceversa).

È anche possibile definire più soglie, se nell'istogramma sono presenti tre o più mode, corrispondenti a oggetti diversi. In questo caso, chiamato **multilevel thresholding**, può essere difficile individuare delle soglie che diano risultati affidabili, quindi si preferisce spesso adottare altre tecniche.

2.1 Tipi di thresholding

In generale, un'operazione di sogliatura è rappresentata da una funzione

$$T[(x, y), f(x, y), p(x, y)]$$

i cui argomenti sono:

- (x, y) : le coordinate del pixel;
- $f(x, y)$: il livello di grigio del pixel;
- $p(x, y)$: una proprietà locale (come, ad esempio, il livello medio dei pixel di un determinato intorno).

Le caratteristiche di T determinano il tipo di soglia utilizzata per il thresholding:

- se T dipende solo da $f(x, y)$, si ha una **soglia globale**;
- se T dipende sia da $f(x, y)$ che da $p(x, y)$, si ha una **soglia locale**;
- se T dipende da $f(x, y)$, $p(x, y)$, e (x, y) , si ha una **soglia dinamica, adattiva**, cioè tarata per ogni pixel (che è molto complessa, e perciò poco usata).

2.2 Rumore e illuminazione

Il rumore può compromettere l'applicabilità del thresholding, introducendo variazioni dei livelli di grigio degli oggetti e dello sfondo che impediscono la distinzione delle mode nell'istogramma, e quindi la definizione delle soglie.

La soluzione è applicare un operatore di smoothing prima del thresholding, per cercare di rimuovere il rumore e "ripulire" l'istogramma, separando nuovamente le mode corrispondenti alle varie regioni.

Un altro fattore che rende difficile l'identificazione di una soglia è la presenza nell'immagine di illuminazioni non omogenee: in tal caso, parti degli oggetti e dello sfondo possono avere valori di grigio simili/uguali, a causa di una diversa illuminazione.

2.3 Sogliatura globale

La sogliatura globale è adeguata per immagini semplici con pochi oggetti uniformi. Un esempio di applicazione è il controllo industriale, per il quale si ha un'illuminazione controllata.

Uno dei modi per definire la soglia globale è un semplice algoritmo iterativo:

1. si seleziona un primo valore per la soglia T (ad esempio, il valore medio di tutti i pixel nell'immagine);
2. si segmenta l'immagine in base a T , ottenendo così due gruppi di pixel: G_1 , con valori sopra la soglia, e G_2 , con valori minori o uguali alla soglia;
3. si calcolano la media m_1 dei valori in G_1 e la media m_2 dei valori in G_2 ;
4. si definisce il nuovo valore di soglia, $T = \frac{1}{2}(m_1 + m_2)$;
5. si ripete il procedimento dal passo 2, finché la variazione di T tra un'iterazione e la successiva è inferiore a un certo valore T_0 (solitamente, per garantire la terminazione, si impone anche un numero massimo di iterazioni).

In alcuni casi in cui la sogliatura globale non è efficace (ad esempio, a causa di un'illuminazione non uniforme), si può invece ottenere un buon risultato suddividendo l'immagine originale in sottoimmagini, e usando una soglia diversa per ciascuna di esse. Se la prima suddivisione non è sufficiente, si suddividono ulteriormente le regioni problematiche, cercando di raggiungere per ogni sottoimmagine le condizioni nelle quali è efficace una sogliatura globale.

2.4 Sogliatura locale

Nella sogliatura locale, il valore di soglia T viene calcolato per ciascun pixel, in base, ad esempio, a una qualche misura statistica del suo intorno:

- la media dei valori dei pixel nell'intorno;
- il valore medio tra il minimo e il massimo livello di grigio nell'intorno ($T = \frac{\min + \max}{2}$);
- la mediana dei valori dei pixel nell'intorno;
- ecc.

Spesso, inoltre, è utile tarare la soglia sommando/sottraendo una costante alla misura statistica scelta. Ad esempio, se si considera la media dei valori nell'intorno, tutti i pixel nelle zone più omogenee di un'immagine tendono a essere molto vicini alla soglia, quindi il thresholding risulta essere molto sensibile alle piccole variazioni di valore in queste

aree, e di conseguenza può produrre dei pessimi risultati. Per risolvere questo problema, è sufficiente sottrarre una costante C alla media nel calcolo della soglia:

$$T = \text{media} - C$$

A differenza della sogliatura globale, quella locale può essere efficace anche per immagini il cui sfondo presenta un forte gradiente (ad esempio, a causa di un'illuminazione molto variabile).

3 Region growing

Le tecniche di **region growing** si basano sul concetto di *accrescimento delle regioni*: si parte con un insieme di **semi**, e a questi si aggregano pixel dei loro intorni che rispettano un determinato criterio di omogeneità.

Con questo metodo, il numero massimo di regioni che possono essere individuate è uguale al numero di semi da cui si parte. È invece possibile che il numero di regioni sia minore del numero di semi, se delle regioni “nate” da semi diversi si uniscono.

Per individuare i semi, è possibile adottare delle procedure euristiche, e/o basate su conoscenze a priori relative all'immagine. Anche con lo stesso criterio di omogeneità, semi diversi possono portare a risultati diversi. Per questo, la scelta dei semi è molto importante, ed è spesso tanto complessa quanto la fase di accrescimento delle regioni.

Il criterio di omogeneità può essere basato, ad esempio, sull'uniformità dei livelli di grigio, sul colore, sulla tessitura, ecc. Per determinarlo, può essere utile riferirsi all'istogramma (ma, rispetto all'uso dell'istogramma per il thresholding, con il region growing si ha la garanzia di continuità spaziale).

I vantaggi del region growing sono la flessibilità e l'accuratezza, mentre i punti critici sono la scelta dei semi e la complessità computazionale degli algoritmi utilizzati.

3.1 Esempio

Si considera l'immagine:

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

Come semi, vengono scelti due pixel i cui valori (1 e 7) sono alcuni dei più frequenti nell'immagine:

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

Si effettua poi l'accrescimento, considerando l'adiacenza a 8, e stabilendo, come criterio di omogeneità, che la differenza tra il livello di grigio del pixel corrente e del seme deve essere minore di 3. L'immagine viene così segmentata in due regioni, corrispondenti ai due semi iniziali:

0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

Se, invece, il criterio utilizzasse una soglia pari a 8, anziché 3, si otterrebbe una singola regione.

4 Split and merge

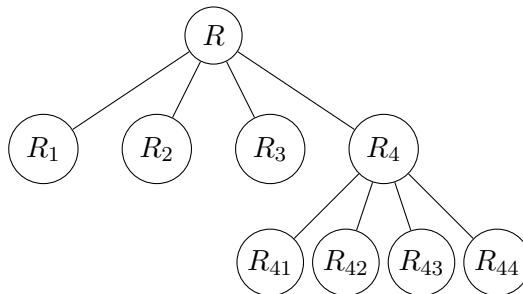
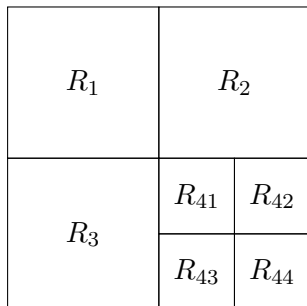
Il metodo di segmentazione **split and merge** è composto da due fasi:

1. **split**: suddivisione top-down dell'immagine in regioni elementari;
2. **merge**: raggruppamento bottom-up delle regioni elementari in regioni complesse.

Il processo di splitting avviene mediante una procedura ricorsiva, che costruisce un albero quaternario:

1. si suddivide l'immagine in quattro regioni, e si controlla se ciascuna di esse soddisfa la condizione di omogeneità P ;

- le regioni che non soddisfano tale condizione vengono nuovamente suddivise in quattro, e il processo si ripete finché la condizione di omogeneità non è soddisfatta per tutte le regioni.



Al termine dello splitting, l'immagine è segmentata in molte regioni di piccole dimensioni. È quindi probabile che ci siano regioni adiacenti con le stesse proprietà (sovra-segmentazione). Allora, nella fase di merge, si fondono le regioni adiacenti R_i e R_j per le quali $P(R_i \cup R_j) = \text{True}$. Quando non è possibile eseguire altre operazioni di merge, questa fase termina, e si ottiene l'immagine segmentata.

L'algoritmo può essere implementato con alcune varianti:

- si può fissata una dimensione minima che le sottoregioni devono avere (ad esempio 16×16 , 8×8 , ecc.);
- nella fase di merge, invece di valutare la condizione $P(R_i \cup R_j) = \text{True}$, si può valutare il predicato separatamente per ciascuna delle due regioni.

In questo modo, si ottiene un algoritmo più veloce.