

4.Elaborazione di Immagini

4.1 Individuazione delle necessità

In questo capitolo viene fatto un accenno a quella che è la teoria sugli algoritmi di elaborazione dell'immagine in seguito utilizzati.

Pensando ad una qualsiasi ripresa di un bancale di cartone, è evidente che come prima operazione si debba eseguire un ritaglio (segmentazione) della parte di immagine utile, infatti solo dopo può essere effettuato il conteggio del numero fogli da cui è composta ogni fila. Il calcolo del numero di file non è necessario in quanto tale dato ci è fornito in modo esatto dal sistema gestionale.

Quindi gli algoritmi utilizzati possono essere divisi in due categorie:

- algoritmi per la segmentazione dell'immagine
- algoritmi per il conteggio

Per quanto riguarda gli algoritmi di conteggio, questi sono solitamente progettati ad hoc per le applicazioni che li utilizzano. Quindi in questo capitolo verranno presentati i soli algoritmi per la segmentazione.

4.2 Algoritmi di segmentazione dell'immagine

In gergo si parla di segmentazione dell'immagine come dell'individuazione in una certa immagine dei contorni chiusi, delle linee e più in generale delle discontinuità, allo scopo di partizionare l'immagine in regioni omogenee, ognuna delle quali corrisponderà ad un oggetto reale.

Capitolo 4 – Elaborazione di immagini

Accade infatti molto spesso che il contenuto informativo che interessa estrarre da un'immagine risieda nei contorni; ciò porta ad una riduzione drastica dello spazio di memoria occupato dalle informazioni utili. (invece di memorizzare tutta l'immagine, si memorizzano solo i contorni)

Sono molte le applicazioni di queste tecniche: dalla robotica industriale (riconoscendo il contorno l'end effector interagisce meglio con l'oggetto), alla fabbricazione di circuiti stampati (dove si usano metodi di segmentazione per evidenziare eventuali interruzioni non volute nelle piste metalliche di collegamento) a applicazioni di conteggio o di misurazioni geografiche.

Tuttavia il campo nel quale forse più che in tutti gli altri queste tecniche hanno trovato utili applicazioni è quello biomedico. Il riconoscimento di contorni infatti consente di identificare gli organi e di analizzarli per scopi diagnostici.

I metodi per la segmentazione dell'immagine si possono suddividere in due categorie:

- metodi di analisi globale: cercano di riconoscere in un solo passo tutti i contorni presenti in un'immagine
- metodi di analisi locale: consentono l'individuazione di un contorno per volta.

I risultati migliori si ottengono utilizzando entrambe le tecniche.

Un bordo è quell'insieme di pixel appartenenti alla frontiera tra due regioni omogenee in cui, nella transizione da una regione all'altra, la funzione intensità subisce una brusca variazione, sottintendendo un picco nel suo gradiente. Sotto tale ipotesi, se prendiamo la derivata dei valori di intensità sull'immagine e valutiamo i punti dove tale derivata è massima (o minima) riusciamo in prima approssimazione a trovarne i contorni, per questo vengono raggruppati come **metodi del primo ordine**. Altri metodi, classificati come **metodi del secondo ordine**,

valutano invece gli attraversamenti dello zero della derivata seconda.

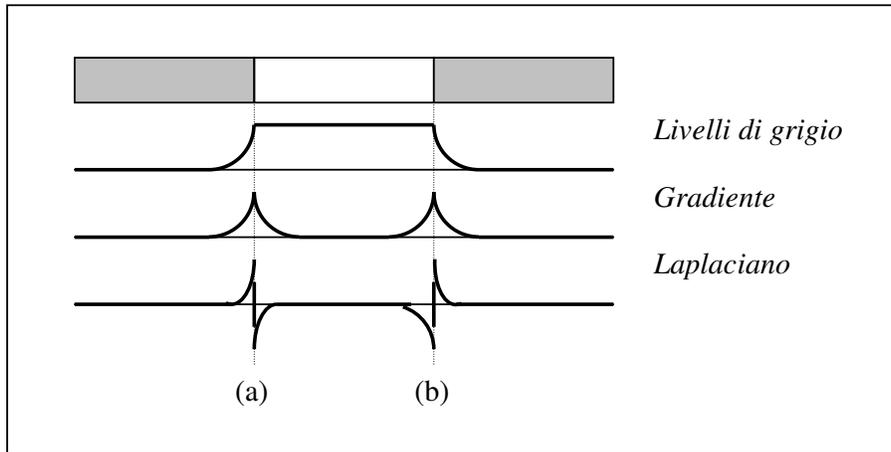


Figura 4.1 – Rivelazione di bordi

In (a) si vede che la derivata prima del profilo del livello di grigio è positiva nella zona di transizione dal nero al bianco, negativa nella transizione dal bianco al nero e zero nella zona con livello di grigio costante, mentre in (b) si ha la situazione opposta. La derivata seconda mostra invece una doppia transizione ad ogni discontinuità, essendo zero nel punto in cui la derivata prima è massima.

4.2.1 Operatore Gradiente – 1° ordine

Il gradiente di un'immagine $f(x,y)$ nel punto (x,y) è il vettore:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (4.1)$$

È noto che il vettore gradiente punta nella direzione di massima velocità di variazione di f nei punti (x,y) . Pertanto nel problema della rivelazione dei bordi è importante l'ampiezza di questo vettore data da:

$$\nabla f = [G_x^2 + G_y^2]^{1/2} \quad (4.2)$$

Di solito nella pratica si approssima la (3.2) con la seguente:

$$\nabla f = |G_x| + |G_y| \quad (4.3)$$

che è molto più semplice da implementare hardware. Anche la direzione del gradiente $\alpha(x,y)$ è una quantità importante:

$$\alpha(x,y) = \tan^{-1} \left[\frac{G_y}{G_x} \right] \quad (4.4)$$

dove l'angolo è misurato rispetto all'asse x. Notare che il calcolo del gradiente di un'immagine è basato sul calcolo delle derivate $\delta f/\delta x$ e $\delta f/\delta y$ per ogni pixel dell'immagine. Quindi il calcolo del gradiente di un'immagine deve essere fatto in due passi utilizzando due maschere: una per la direzione x ed una per la direzione y.

Dovendo lavorare su immagini digitalizzate, i cui valori sono numeri discreti, sono state introdotte approssimazioni sia sulle derivate, le quali vengono approssimate tramite i loro rapporti incrementali, sia sull'operazione di differenziazione che viene eseguita tramite convoluzione con una maschera; in tal modo, il calcolo della derivata avviene in un intorno del punto in esame ed il valore finale viene ottenuto da una media pesata delle singole derivate.

Gli operatori in questa classificazione che abbiamo preso in considerazione sono stati :

- *Operatore di Roberts*
- *Operatore di Prewitt*
- *Operatore di Sobel*

Tra questi operatori le differenze risiedono nella dimensione delle maschere di convoluzione, nei pesi dati a ciascuna derivata all'interno della maschera e nel numero di derivate utilizzate.

E' di fondamentale importanza la scelta di una soglia che indichi quali pendenze considerare come discontinuità. Tale soglia può rivelarsi decisiva per regolarizzare eventuali errori di disturbo presenti nelle immagini e questo molto spesso può costituire motivo di rigidità di funzionamento di questi metodi.

4.2.1.1 Operatore di Roberts

L' operatore di Roberts effettua un controllo incrociato su due direzioni: +45° e +135°

$$\Delta_{45^\circ} = f(i+1, j+1) - f(i, j) \quad (4.5)$$

$$\Delta_{135^\circ} = f(i, j+1) - f(i+1, j) \quad (4.6)$$

(l'operatore usa le direzioni diagonali per calcolare i vettori gradiente)

A cui corrispondono le seguenti maschere di convoluzione:

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Figura 4.2 – Maschere di Roberts

e quindi il modulo del gradiente sarà:

$$|\nabla f| \approx |G_x| + |G_y| \quad (4.7)$$

(dove G_x e G_y sono i gradienti calcolati nelle due direzioni)

Limiti:

- Vengono estratte ed esaltate tutte le componenti ad alta frequenza quali il rumore
- I contorni estratti non sono a spessore unitario
- Non è garantita in alcun modo la continuità del contorno
- Immagine con Disturbo Uniforme

Conclusione:

Il risultato è piuttosto buono per immagini pulite; bisogna invece scegliere una soglia ottimale per elaborare immagini disturbate. Il metodo utilizzato prevede la scelta per l'elaborazione del risultato più alto: spesso il rumore influisce pesantemente su questo calcolo e si rischia di considerare di più il disturbo che l'immagine stessa.

Questo metodo, inoltre, marca solo i punti interessati, senza restituire alcuna informazione circa la loro orientazione.

4.2.1.2 Operatore di Prewitt

L'operatore di Prewitt è il più semplice degli operatori 3x3. Utilizza due maschere di convoluzione ortogonali e con somma dei pesi uguale a 0.

Le matrici sono:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ per } \frac{\partial f}{\partial x}; \quad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \text{ per } \frac{\partial f}{\partial y}$$

Figura 4.3 - MASCHERE DI PREWITT

La presenza di rumore sull'immagine originaria può alterare i risultati, specialmente nelle fasi, anche se con l'applicazione

di questo filtro 3x3 i risultati sono migliori rispetto a quelli ottenuti tramite l'operatore di Roberts (2x2).

Ogni maschera è convoluta con l'immagine. Per ogni pixel dell'immagine sono calcolati due numeri p_1 e p_2 , uno per ciascuna maschera; con i quali vengono poi calcolati due parametri:

$$\text{Edge Magnitude: } \sqrt{p_1^2 + p_2^2} \quad (4.8)$$

$$\text{Edge Direction: } \tan^{-1}\left(\frac{p_1}{p_2}\right) \quad (4.9)$$

4.2.1.3 Operatore di Sobel

L'operatore di Sobel e' uno degli operatori 3x3 che, a differenza di Prewitt, tiene conto della distanza dei pixel dal centro in modo approssimativo.

Le matrici ortogonali da applicare sono:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ per } \frac{\partial f}{\partial x}; \quad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \text{ per } \frac{\partial f}{\partial y}$$

Figura 4.4- MASCHERE DI SOBEL

Ognuna delle due maschere è convoluta con l'immagine, alla fine delle convoluzioni ad ogni pixel sono associati due numeri s_1 e s_2 , uno per ciascuna maschera. Questi numeri sono quindi utilizzati per elaborare due metriche, l'edge magnitude e l'edge direction, così definite:

$$\text{Edge Magnitude: } \sqrt{s_1^2 + s_2^2} \quad (4.10)$$

$$\text{Edge Direction: } \tan^{-1}\left(\frac{s_1}{s_2}\right) \quad (4.11)$$

Capitolo 4 – Elaborazione di immagini

anche qui il rumore può influire negativamente sull'analisi, anche in questo caso specialmente sulle fasi. I risultati che solitamente si ottengono sono comunque migliori rispetto a quelli che si possono ottenere tramite l'operatore di Roberts (2x2).

Consideriamo ora le seguenti sezioni di codice, semplificate nei dettagli implementativi:

Robert

```
// prima maschera di Roberts
G0=img[r][c]; [1 0]
G2=img[r+1][c+1]; [0 -1]
//seconda maschera di Roberts
G1=img[r][c+1]; [0 1]
G3=img[r+1][c]; [-1 0]
// calcolo del gradiente e della fase
pixel_Gradiente = sqrt((G2-G0)^2+(G1-G3)^2);
if( pixel_Gradiente > 30 ){
    pixel_Fase=atan2((G2-G0), (G1-G3));
    pixel_Fase=(pixel_Fase + pi/2)*255/pi;
}
else pixel_Fase=0;
```

Prewitt

```
// Maschera Verticale
pixel_mv =
(1 * img[r-1][c+1] -1 * img[r-1][c-1]
+ 1 * img[r][c+1] -1 * img[r][c-1]
+ 1 * img[r+1][c+1] -1 * img[r+1][c-1])/3;
// Maschera Orizzontale
pixel_mo =
(1 * img[r-1][c+1] + 1 * img[r-1][c-1]
+ 1 * img[r-1][c] - 1 * img[r+1][c]
- 1 * img[r+1][c+1] - 1 * img[r+1][c-1])/3;
// calcolo del gradiente e della fase
pixel_Gradiente = sqrt(pixel_mv^2+pixel_mo^2);
if( pixel_Gradiente > 30 ){
    pixel_Fase=atan2(pixel_mv, pixel_mo);
    pixel_Fase=(pixel_Fase + pi/2)*255/pi;
}
else pixel_Fase=0;
```

Sobel

Capitolo 4 – Elaborazione di immagini

```
// Maschera Verticale
pixel_mv =
  (1 * img[r-1][c+1] -1 * img[r-1][c-1]
+ 2 * img[r][c+1]   -2 * img[r][c-1]
+ 1 * img[r+1][c+1] -1 * img[r+1][c-1])/4;
// Maschera Orizzontale
pixel_mo =
  (1 * img[r-1][c+1] + 1 * img[r-1][c-1]
+ 2 * img[r-1][c]   - 2 * img[r+1][c]
- 1 * img[r+1][c+1] - 1 * img[r+1][c-1])/4;
// calcolo del gradiente e della fase
pixel_Gradiente = sqrt(pixel_mv^2+pixel_mo^2);
if( pixel_Gradiente > 30){
  pixel_Fase=atan2(pixel_mv,pixel_mo);
  pixel_Fase=(pixel_Fase + pi/2)*255/pi;
}
else pixel_Fase = 0;
```

Il primo passo e' stato quello di calcolare la convoluzione "discreta" nelle due distinte direzioni, calcolarne la somma vettoriale ed assegnare tale risultato al pixel dell'immagine elaborata; questo processo viene poi ripetuto per ogni pixel facendo scorrere la maschera su tutta l'immagine originale. Il calcolo della fase ha comportato in primo luogo un'applicazione di soglia, che abbiamo preferito mantenere costante in tutti e tre i metodi per valutarne le differenze, ed una normalizzazione (espansione) di valori per riportarla alla corretta scala tra 0 e 255 presente nelle nostre immagini a gradazione di grigi. La soglia applicata alla fase ha inoltre permesso di evidenziarne solo le parti significative, poiché il calcolo nelle zone omogenee non avrebbe prodotto alcuna informazione. Dalle immagini prodotte in Figura1 e da alcune prove da noi effettuate possiamo poi osservare quanto segue. I tre operatori hanno risposto bene amplificando le rapide variazioni dell'immagine ma, per loro natura, sono stati scarsamente sensibili ai tratti con variazioni gradualmente. La dimensione ristretta delle maschere (2x2 e 3x3) favorisce l'elaborazione di quasi tutti i pixel, trascurando solo quelli del bordo (spessore di 1 pixel) e quindi si rivela efficace per elaborare praticamente tutta l'immagine; d'altra parte però un

intorno così ristretto risulta limitativo nella ricerca di contorni nei quali le differenze significative dei valori di luminosità non sono riscontrabili prendendo in esame pixel adiacenti. Questo è evidente applicando i metodi su contorni sfumati o degradati presenti ad esempio nelle nostre immagini alle quali sono stati aggiunti rumori sia uniformi che random (immagini `im*nu.pgm` e `im*nr.pgm`). I tre metodi hanno poi saputo produrre significative immagini di fase; dai risultati abbiamo potuto constatare quanto la fase sia un indice direzionale ovvero come questa indichi la direzione che assume il pixel considerato. A direzioni uguali corrispondono stessi colori. Il fatto che nelle immagini prodotte stesse angolazioni hanno colore diverso indica che le direzioni sono si parallele ma sfasate di 180 gradi. Poiche' in un angolo sono presenti 360 gradi ma avendo noi a disposizione soltanto 256 colori la rappresentazione e' risultata poco sensibile tra angoli vicini.

4.2.2 Operatore Laplaciano – II° Ordine

L'operatore Laplaciano di una funzione 2D $f(x,y)$ è una derivata seconda definita nel modo seguente:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (4.12)$$

Come per il gradiente, la (4.12) può essere implementata in forma digitale in diversi modi, ma il più frequente è quello basato su di una maschera 3x3, tale che:

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8) \quad (4.13)$$

dove la funzione su cui si applica il Laplaciano è rappresentata dai livelli di grigio dell'immagine. La condizione

necessaria per definire un filtro Laplaciano è che il coefficiente al centro del pixel sia positivo e quelli al di fuori siano negativi, come descritto nella figura successiva.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Figura 4.5 – Maschera per Laplaciano

Come mostrato nella figura 4.1, il Laplaciano produce variazioni positive e negative in corrispondenza del bordo. Pertanto per isolare il bordo sarà necessario compiere ulteriori operazioni sull'immagine di gradiente. Per esempio si possono ricercare i punti di massimo e di minimo, oppure l'attraversamento della linea di zero (zero crossing). Un operatore Laplaciano molto usato è ottenuto applicando l'operatore ∇^2 alla seguente funzione gaussiana 2D:

$$h(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.14)$$

dove σ indica la deviazione standard. Definiamo $r^2 = x^2 + y^2$. La scelta della funzione gaussiana è dovuta al fatto che essa è limitata alle alte frequenze. Applicando il Laplaciano alla (3.10) si ha:

$$\nabla^2 h = \left(\frac{r^2 - \sigma^2}{\sigma^4} \right) e^{-\frac{r^2}{2\sigma^2}} \quad (4.15)$$

La figura 4.6 mostra una sezione di tale funzione. Quando un'immagine viene convoluta con l'operatore $\nabla^2 h$, l'entità del filtraggio è proporzionale a σ . Inoltre, dato che il valore

medio di $\nabla^2 h$ è zero, si può dimostrare che anche il risultato ottenuto convolvendo $\nabla^2 h$ con l'immagine è a valore medio nullo.

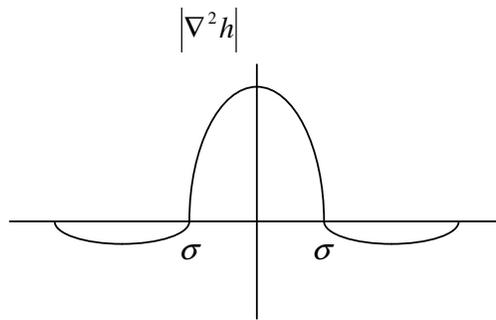


Figura 4.6 – Sezione della funzione 4.15

L'implementazione del filtro basato sulla (4.15) è possibile mediante una maschera 5x5 o superiore per sfruttare meglio la forma del filtro.

Un altro metodo è quello Dog, il cui nome significa "differenza di gaussiane" poiché le maschere che vengono utilizzate si calcolano effettuando questa differenza. Otteniamo così una figura a forma di sombrero come quella mostrata in Figura 4.7 che è molto simile ai valori di intensità che i ricettori dell'occhio umano percepiscono sulla retina.

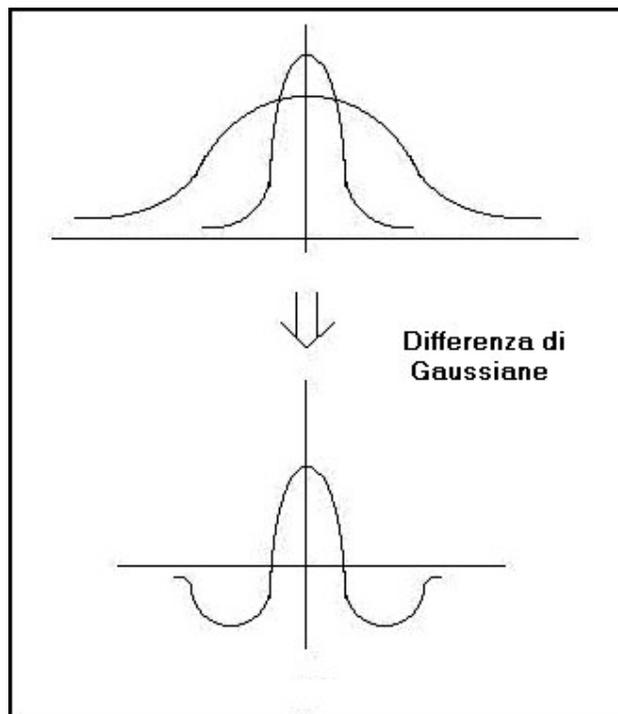


Figura 4.7 – Metodo DoG

Ogni ricevitore quantifica l'intensità ricevuta dalla zona di immagine di sua competenza e lo confronta con i valori dei ricevitori adiacenti. Questa operazione è simulata dal nostro algoritmo mediante delle convoluzioni.

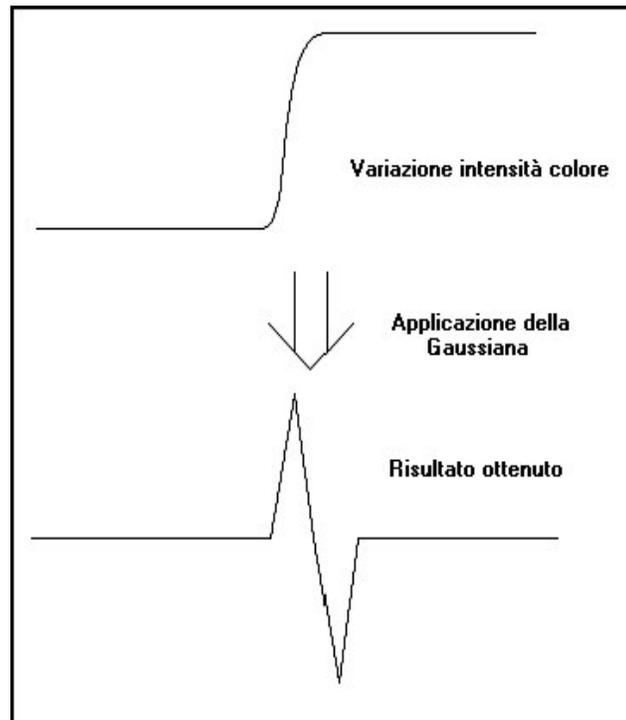


Figura 4.8 – Risultati Metodo DoG

Questa ultima operazione fa sì che per valori ad intensità costante il valore restituito sia zero mentre per zone in cui l'intensità varia (contorni) il valore restituito sia diverso da zero come mostrato in Figura 4.8.

$$Dog = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}} \quad (4.16)$$

Questo filtro permette di ottenere immagini esclusivamente in bianco e nero quindi il contorno è messo bene in evidenza; inoltre quest'ultimo è di dimensione un pixel a differenza degli altri metodi che erano soggetti al problema dei fronti di salita e dei fronti di discesa delle maschere.

Altra caratteristica importante del metodo è quella di avere sempre dei contorni con linee chiuse e anche nel caso in cui il metodo venisse applicato più volte sulla stessa immagine

con valori di delta delle varie gaussiane differenti i contorni comuni nelle due rilevazioni sarebbero esattamente gli stessi.

4.2.3 - Metodi Template-Matching

In questa categoria di metodi ne sono stati scelti 3:

4.2.3.1 Operatore Tre su Nove

Per tale operatore si procede inizialmente attraverso la convoluzione con le seguenti otto maschere:

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

Figura 4.9 – Metodo Tre su Nove: maschere 1..8

dalle quali ottenere gli otto valori corrispondenti; di questi ultimi si sceglie quello con maggiore valore e lo si divide per quello ottenuto elaborando un'ulteriore convoluzione attraverso una maschera (di media) :

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figura 4.10 – Metodo Tre su Nove: maschera 9

e procedendo infine con una normalizzazione dei dati, tipicamente :

$$\text{Pixel_finale} = 1.5 * (\text{valore} - 0.333) \quad (4.17)$$

4.2.3.2 Metodo di Kirsch Compass

Le maschere di Kirsch sono definite maschere compass perché sono definite da una singola maschera che viene ruotata nelle otto maggiori orientazioni. Le maschere sono definite come segue:

$$\begin{array}{ll} k0 \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & k1 \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \\ k2 \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & k3 \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\ k4 \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & k5 \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \\ k6 \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & k7 \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \end{array}$$

Figura 4.11 – Maschere di Kirsch

L'edge magnitude è definito come il massimo valore trovato dalla convoluzione di ogni maschera con l'immagine. L'edge direction è individuata dalla maschera che produce il massimo magnitude: per esempio k0 corrisponde ad un edge verticale, mentre k5 corrisponde ad un edge diagonale nella direzione nord-ovest/sud-est. Si può notare che le ultime 4 maschere sono come le prime quattro, ma ruotate rispetto agli assi centrali.

4.2.3.3 Metodo di Robinson Compass

Le maschere di Robinson sono utilizzate in modo simile a quelle di Kirsch, ma sono più semplici da realizzare perché rappresentano solamente coefficienti pari 0,1 e 2 e sono simmetriche rispetto agli assi direzionali (cioè gli assi con gli zeri).

Nell'utilizzazione ci sarà bisogno di calcolare i risultati su quattro delle maschere, i risultati delle altre quattro potrà essere ottenuto semplicemente negando i risultati delle prime quattro.

Le maschere sono ottenute come segue:

$$\begin{array}{ll} r0 \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & r1 \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \\ r2 \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} & r3 \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} \\ r4 \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} & r5 \begin{bmatrix} 0 & -1 & 2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \\ r6 \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} & r7 \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \end{array}$$

Figura 4.12 – Maschere di Robinson

L'edge magnitude è definito come il massimo valore trovato dalla convoluzione di ogni maschera con l'immagine. L'edge direction è dato dalla maschera che produce il massimo magnitude.

4.2.4 Metodo basato sul colore

Il metodo in questione è basato sul colore è quindi ogni oggetto che vuole riconoscere nell'immagine ne deve avere un suo specifico. L'idea è quella di applicare un filtro all'immagine da elaborare con lo scopo di estrarre solo i punti il cui colore rientra nel filtro (punti significativi); si ottiene un'immagine binaria in cui tutti e soli i punti significativi sono posti ad 1.

Capitolo 4 – Elaborazione di immagini

Su queste immagini poi, tramite correlazioni, si individuano le aree con maggiore densità di colore cercato e si scartano le aree che contengono pochi punti di detto colore.

Il riconoscimento basato sul colore, richiede quindi una certa familiarità con i modelli per la loro rappresentazione. Il modello RGB, a causa della sua struttura orientata all'hardware, non è semplice da usare. Per poter riconoscere un oggetto P deve essere noto il suo colore (R_p, G_p, B_p) e si deve applicare un filtro all'immagine tarato su tale tonalità di colore in modo da estrarre solo i punti appartenenti a P.

Questo metodo risulterebbe adeguato, quindi efficace, se il colore di P nell'immagine fosse uniforme ma, di fatto, per la presenza di ombre e riflessi causati dalle risorse di luci non è vero. Il risultato in tale caso è che solo pochi punti vengano riconosciuti come appartenenti a P.

Si potrebbe allora utilizzare un filtro "allargato" e prendere come punti dell'oggetto P tutti quei punti caratterizzati da valori RGB tali che:

$$\begin{aligned} R_p - \text{delta} &\leq R \leq R_p + \text{delta} \\ G_p - \text{delta} &\leq G \leq G_p + \text{delta} \\ B_p - \text{delta} &\leq B \leq B_p + \text{delta} \end{aligned} \quad (4.18)$$

Anche se questo nuovo filtro può sembrare efficiente, in effetti ci sono diverse considerazioni da fare:

- colori simili come il primo ed il terzo in figura 3.15, hanno grandi variazioni dei valori di verde e blu; le stesse variazioni differenziano il primo ed il secondo colore che però sono visibilmente diversi;

		
R = 255	R = 255	R = 255
G = 100	G = 80	G = 120
B = 0	B = 40	B = 40

Figura 4.13 – Metodo basato sul colore

- tra colori simili, le variazioni dei colori fondamentali non sono uniformi; sarebbe più corretto usare tre delta distinti:

$$\begin{aligned}R_p - \text{delta}R &\leq R \leq R_p + \text{delta}R \\G_p - \text{delta}G &\leq G \leq G_p + \text{delta}G \\B_p - \text{delta}B &\leq B \leq B_p + \text{delta}B\end{aligned}\tag{4.19}$$

A questo punto risulta necessario stabilire quali siano i valori di delta corretti: questa determinazione non è facile in quanto essi dipendono dal colore di riferimento ed anche dalle condizioni di luce d'ambiente che, per un robot in movimento, cambiano continuamente.

Il modello HSV risulta essere più intuitivo e quindi più facile da utilizzare (rispetto al modello RGB) per il riconoscimento degli oggetti. Anche per il filtro HSV conviene usare dei delta distinti per ogni parametro. Tenendo conto della netta distinzione tra il colore (H e S) e la luminosità (V) è possibile creare un filtro robusto alle variazioni di luminosità.

Quindi abbiamo generato un'immagine binaria, l'immagine filtrata, nella quale ad ogni 1 corrisponde un pixel dell'immagine di partenza i cui valori HSV rientrano nel filtro scelto. Eseguiamo la correlazione tra quest'immagine ed una più piccola di colore uniforme, (tutti 1) rettangolo di correlazione. Tale rettangolo scorrendo sull'immagine binaria, calcola la funzione di correlazione; quando questa ha un valore maggiore di una prefissata soglia, allora in quell'area c'è un'alta percentuale del colore cercato e può essere considerata tutta di quel colore.

Capitolo 4 – Elaborazione di immagini

Notiamo che un singolo valore della funzione di correlazione si ottiene sommando il numero di 1 in un'area dell'immagine binaria di dimensioni pari al rettangolo di correlazione. Il successivo valore si calcola da un'area parzialmente sovrapposta o anche adiacente e così via fino a coprire tutta l'immagine.