Capitolo 10
SEGMENTAZIONE DI IMMAGINI

La **segmentazione** suddivide un'immagine nelle regioni o negli oggetti che la compongono. Il dettaglio della segmentazione dipende da ciò che si vuole ottenere dall'immagine, cioè il processo deve terminare quando gli oggetti o le regioni di interesse sono stati individuati.
Ci sono due possibili situazioni: si conoscono a priori le condizioni di contorno, quindi si conosce a priori la struttura dell'immagine; oppure non si conosce nulla a priori e si utilizzano dei metodi specifici per individuare le regioni di interesse.
Ci sono due approcci alla segmentazione:

- **Discontinuità** (*segmentazione dipendente dagli edge*): si partiziona un'immagine basandosi sui bruschi cambiamenti di intensità locali (ad esempio gli edge), quindi i bordi delle regioni devono essere sufficientemente diversi l'uno dall'altro e dallo sfondo.

- **Similarità** (*segmentazione dipendente dalle regioni*): ci si basa sulle similarità tra regioni. Si utilizzano le tecniche di *thresholding* e *region growing.*

Se *R* indica l'intera regione occupata da un'immagine, la segmentazione è quel processo che partiziona *R* in *n* sottoregioni, $R_1, R_2, \ldots, R_n$ rispettando le seguenti proprietà:

a) $\bigcup_{i=1}^{n} R_i = R$
cioè la segmentazione deve essere completa, ogni pixel deve appartenere a una qualche regione

b) R_i, è un insieme connesso, *i = 1,2,…,n*
cioè i punti in una regione devono essere 4 o 8 connessi

c) $R_i \cap R_j = \emptyset$ per tutti i valori *i* e *j*, $i \neq j$
cioè le regioni devono essere disgiunte

d) $Q(R_i) = VERO$ per *i = 1,2,…,n*
indica le proprietà che devono essere soddisfatte dai pixel in una regione segmentata

e) $Q(R_i \cup R_j) = FALSO$ per ogni coppia di regioni adiacenti R_i e R_j
due regioni adiacenti R_i e R_j devono essere diverse nel senso del predicato *Q*. Ricordiamo che due regioni sono adiacenti se la loro unione forma un insieme connesso.

Individuazione di edge, linee e punti

Le tre caratteristiche dell'immagine a cui siamo interessati sono:

- **Edge** (o segmenti di edge):
  sono insiemi di pixel di edge connessi, dove i *pixel di edge* sono pixel in cui l'intensità dell'immagine cambia all'improvviso. I pixel di edge vengono individuati tramite dei metodi di elaborazione locali chiamati *individuatori di edge* (edge detector)
- **Linee**:
  segmento di edge in cui l'intensità dello sfondo su entrambi i lati della linea è maggiore o minore dell'intensità dei pixel della linea.
- **Punti**:
  linea con lunghezza e larghezza pari a 1 pixel.

Gli strumenti che permettono di individuare le brusche variazioni locali di intensità sono le derivate prima e seconda.
Le caratteristiche matematiche della *derivata prima* sono:

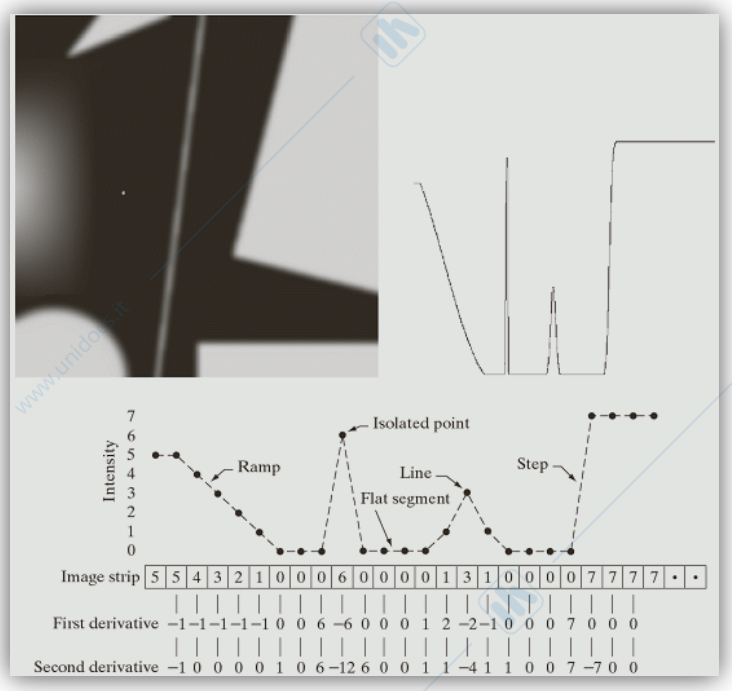
- Deve essere nulla nelle aree di intensità costante
- Deve essere non nulla all'inizio di una rampa o di un gradino di intensità
- Deve essere non nulla nei punti lungo una rampa di intensità

Inoltre la derivata prima produce edge spessi e mette in rilievo i dettagli fini meno rispetto alla derivata seconda.

Le caratteristiche matematiche della *derivata seconda* invece sono:

- Deve essere nulla in aree ad intensità costante
- Deve essere non nulla all'inizio e alla fine di una rampa o di un gradino di intensità
- Deve essere nulla lungo le rampe di intensità

Inoltre la derivata seconda evidenzia meglio i cambiamenti bruschi e quindi mette in rilievo i dettagli sottili (incluso il rumore).
Produce una *risposta doppia* lungo le transizioni di intensità a rampa e a gradino, questo avviene perché abbiamo visto che è diversa da zero sia all'inizio che alla fine del gradino, mentre la derivata prima è diversa da zero solo all'inizio.
Il segno della derivata seconda può essere utilizzato per determinare il tipo di transizione di un edge, cioè se c'è una transizione da chiaro a scuro (*derivata seconda negativa*) o da scuro a chiaro (*derivata seconda positiva*) dove il segno si mantiene mentre si attraversa l'edge.
Questa immagine mostra un esempio dei valori appena specificati che assumono le derivate lungo un certo profilo di intensità di un'immagine:



Siccome abbiamo a che fare con immagini digitali, quindi valori finiti, la massima variazione di intensità possibile è finita e la più piccola distanza in cui può avvenire un cambiamento è tra pixel adiacenti.
Per calcolare le derivate prime e seconde in ogni pixel dell'immagine si utilizzano i filtri spaziali con opportune maschere.

Individuazione di punti isolati

Da quanto detto finora è chiaro che i punti isolati possono essere individuati utilizzando la derivata seconda, nello specifico utilizzando il *laplaciano*:

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

Dopo aver calcolato le derivate parziali quindi otteniamo:

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

Questa espressione è implementata tramite la seguente maschera:

0 1 0
1 -4 1
0 1 0

Se vogliamo includere anche i termini diagonali allora la maschera diventa:

1 1 1
1 -8 1
1 1 1

Come al solito la somma dei coefficienti è zero, in quanto vogliamo che queste maschere diano risposta nulla nelle aree ad intensità costante.
Vengono utilizzate queste maschere per individuare il punto perché l'idea di base è che l'intensità di un punto isolato è abbastanza diversa dalle intensità vicine.
Utilizzando la seconda maschera, un punto viene individuato nella posizione (*x, y*) su cui la maschera è centrata se il valore assoluto della risposta della maschera in quel punto supera uno specifico valore di soglia. Tali punti vengono denotati con 1 nell'immagine di output, mentre tutti gli altri vengono posti a 0, producendo così un'immagine binaria.
In formula:

$$g(x,y) = \begin{cases} 1 & \text{se } |R(x,y)| \geq T \\ 0 & \text{altrimenti} \end{cases}$$

dove *g* è l'immagine di output e *T* è la soglia non negativa.
Quello che avviene è che vengono misurate le differenze pesate tra un pixel e i suoi 8 vicini.

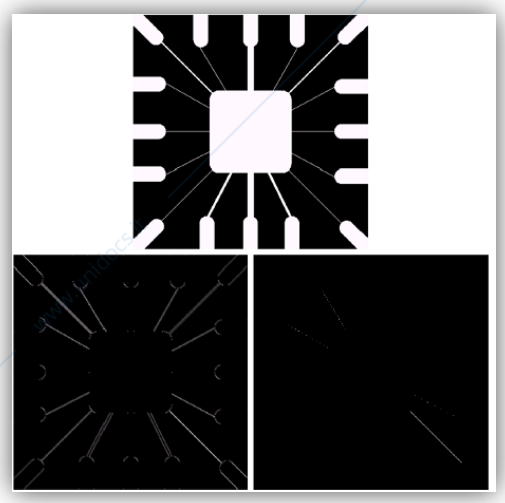
Individuazione di linee

Anche per l'individuazione di linee conviene utilizzare la derivata seconda, in quanto da una risposta più forte e produce linee più sottili rispetto alla derivata prima. Utilizziamo ancora la maschera laplaciana vista per l'individuazione dei punti isolati, stando attenti alla doppia risposta della derivata seconda.
L'immagine laplaciana che otteniamo contiene dei valori negativi, quindi è necessario effettuare un'operazione di *scaling*.
La maschera utilizzata è isotropica, quindi la sua risposta è indipendente dalla direzione; un problema che non avevamo con l'individuazione dei punti isolati è che le linee possono avere differenti direzioni, quindi dobbiamo utilizzare maschere diverse per poter avere risposta massima nelle differenti direzioni.
Ad esempio possiamo utilizzare le seguenti maschere:

-1 -1 -1	2 -1 -1	-1 2 -1	-1 -1 2
2 2 2	-1 2 -1	-1 2 -1	-1 2 -1
-1 -1 -1	-1 -1 2	-1 2 -1	2 -1 -1
orizzontale	*+45°*	*verticale*	*-45°*

Dove la direzione preferita di ogni maschera è pesata con un coefficiente più grande (in questo caso 2) rispetto ad altre possibili direzioni.
Quindi se ci interessa trovare tutte le linee in una certa direzione in un'immagine, dobbiamo spostare la maschera (che meglio individua quella direzione) lungo l'immagine e sogliare il valore assoluto del risultato. I punti che rimangono sono le risposte più forti, che corrispondono meglio alla direzione definita dalla maschera.

Vediamo un esempio:



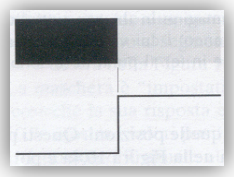
Lo scopo in questo esempio è quello di trovare tutte le linee di un pixel orientate a +45°

(a) Immagine binaria di un circuito elettronico.
(b) Risultato dell'elaborazione con una maschera di Individuazione di linee a +45°. In questa immagine i valori negativi sono impostati a zero, infatti ricordiamo che come risultato del laplaciano otteniamo un'immagine con sfondo grigio.
(c) Risultato dell'applicazione della soglia sull'immagine (b), quindi i punti bianchi sono quelli che soddisfano la condizione.

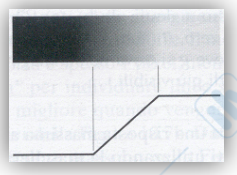
Modelli di edge

L'**edge detection**, cioè l'individuazione dei bordi, è il metodo più utilizzato per la segmentazione di immagini e si basa sui bruschi cambiamenti locali di intensità.
Esistono tre tipi di edge, classificati secondo i loro profili di intensità:

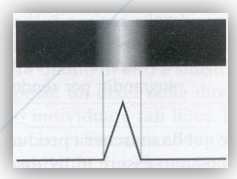
1. *a gradino*:
   sono edge ideali che separano due livelli di intensità alla distanza di *1* pixel; sono ideali perché è più probabile trovarli in un'immagine generata in laboratorio che non in un'immagine reale in quanto queste sono solitamente sfocate e rumorose.



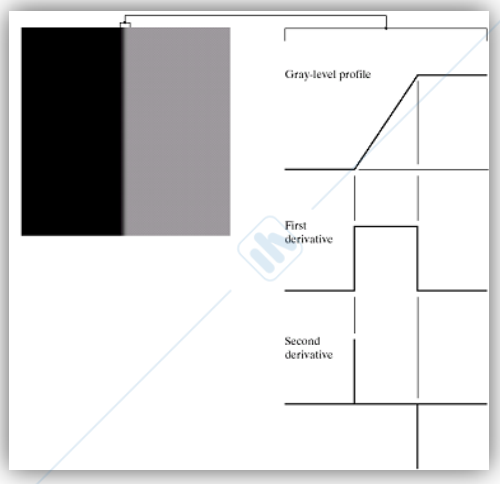
2. *a rampa*:
   sono edge caratterizzati dalla sfocatura e dal rumore presenti nelle immagini presenti a causa del dispositivo di acquisizione. L'ampiezza della rampa è inversamente proporzionale alla sfocatura nell'edge.



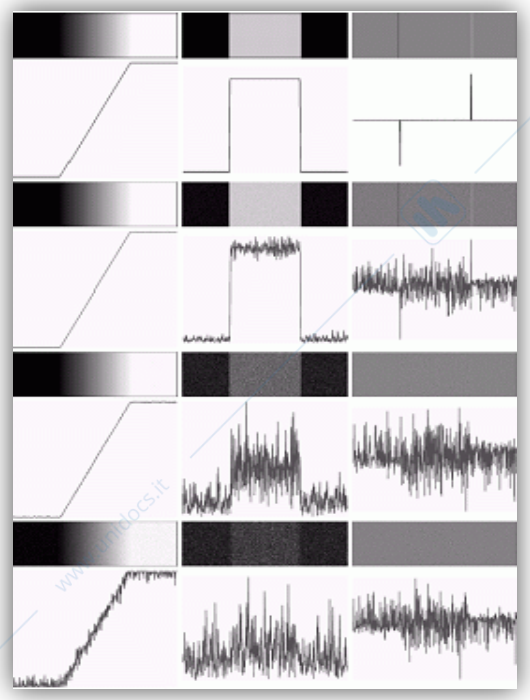
3. *roof*:
   è associato al bordo di una qualche regione e la base (la larghezza) di questo tipo di bordo è determinata dallo spessore e dalla sfocatura della linea.



Comunque sia nelle immagini possono essere presenti tutti e tre questi tipi di edge, ovviamente non identici ai modelli ideali, ma simili in base alla quantità di rumore e sfocatura presenti.
In base alle caratteristiche dei tre modelli di edge risulta evidente che per poterli individuare si possono utilizzare le caratteristiche della derivata prima e della derivata seconda. Infatti la *magnitudo* della derivata prima può essere utilizzata per individuare la presenza di un edge in un punto, mentre il *segno* della derivata seconda può essere utilizzato per determinare se un pixel di un edge giace sul lato scuro o chiaro di un edge. Inoltre l'intersezione tra l'asse di intensità zero e una linea che passa per gli estremi della derivata seconda segna un punto chiamato *zero crossing*, questo punto può essere utilizzato per localizzare i centri di un edge.



Il problema delle derivate è che in presenza di rumore è quasi impossibile ricavare qualsiasi informazione dall'istogramma, come vediamo nella figura sottostante.



Prima colonna: immagini e profili di intensità di un edge a rampa corrotto da rumore gaussiano casuale a media nulla e deviazione standard a livelli di intensità rispettivamente 0.0, 0.1, 10.0.

Seconda colonna: immagini della derivata prima e profili di intensità.

Terza colonna: immagini della derivata seconda e profili di intensità.

Individuazione degli edge

I **tre passi fondamentali** per individuare gli edge sono:

1. applicare lo smoothing all'immagine per ridurre il rumore presente
2. individuare i potenziali punti di edge
3. localizzare gli edge, cioè selezionare tra i punti candidati quelli che lo sono veramente

Le tecniche utilizzate per questo scopo sono:

- operatori gradiente (utilizza la derivata prima)
- combinazione del gradiente con il thresholding (utilizza la derivata prima)
- individuatore di Marr-Hildreth (utilizza la derivata seconda)

Operatori gradiente

Il gradiente viene utilizzato per definire l'*intensità* e la *direzione* di un edge in un certo punto *(x,y)*.
Tutto ciò è possibile perché il gradiente è un vettore bidimensionale che ha la proprietà geometrica di puntare nella direzione di massima variazione di *f* nel punto *(x,y)*.
Il gradiente e' definito:

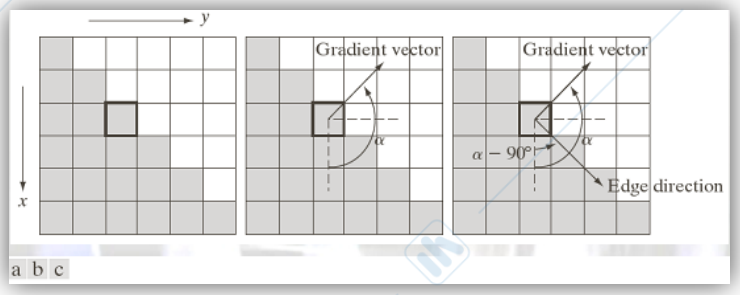
$$\nabla f \equiv grad(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

La direzione del gradiente è data dall'angolo:

$$\alpha(x, y) = \tan^{-1}\left[\frac{g_y}{g_x}\right]$$

calcolato rispetto all'asse *x*. La direzione di un edge in un punto qualsiasi *(x, y)* è *ortogonale* alla direzione di *α(x,y)* del vettore gradiente in quel punto.

Vediamo un esempio:



Utilizzo del gradiente per determinare direzione e forza di un edge in un punto. Si noti che l'edge è perpendicolare alla direzione del vettore gradiente nel punto in cui il gradiente viene calcolato.

Per ottenere le componenti del gradiente serve calcolare le derivate parziali $\partial f/\partial x$ e $\partial f/\partial y$ per ogni pixel dell'immagine e queste possono essere implementate utilizzando le maschere (o operatori) di Roberts, di Prewitt e di Sobel.

- **Operatore di Roberts**:
  è l'operatore a croce più semplice che serve per individuare la direzione diagonale di un edge. E' un operatore *2X2* semplice da implementare, ma scomodo (per calcolare la direzione) per via del fatto che non è simmetrico rispetto al punto centrale.
  Le forme possibili sono:
  -1 0 0 -1
  0 1 1 0

- **Operatore di Prewitt**:
  è un operatore *3X3* che approssima le derivate nella direzione *x* e *y* e permette anche di individuare gli edge diagonali.
  Le forme possibili sono:
  -1 -1 -1 -1 0 1 *orizzontale e verticale*
  0 0 0 -1 0 1
  1 1 1 -1 0 1

  0 1 1 -1 -1 0 *diagonale*
  -1 0 1 -1 0 1
  -1 -1 0 0 1 1

- **Operatore di Sobel**:
  è un operatore *3X3* che approssima le derivate nella direzione *x* e *y* e permette anche di individuare gli edge diagonali.
  Viene preferito a Prewitt perché è più performante nella rimozione del rumore e nelle prestazioni.
  Le forme possibili sono:

```
-1 -2 -1     -1 0 1   orizzontale e verticale
 0  0  0     -2 0 2
 1  2  1     -1 0 1

 0  1  2     -2 -1  0  diagonale
-1  0  1     -1  0  1
-2 -1  0      0  1  2
```

Per poter individuare gli edge in modo più selettivo e mantenendo il più alto grado di connettività solitamente conviene effettuare uno smoothing prima del calcolo del gradiente e quindi il thresholding all'immagine gradiente. Utilizzare questo metodo può portare ad eliminare tutti gli edge i cui valori di intensità sono stati radicalmente attenuati dalla sfocatura.

L'individuatore di edge di Marr-Hildreth

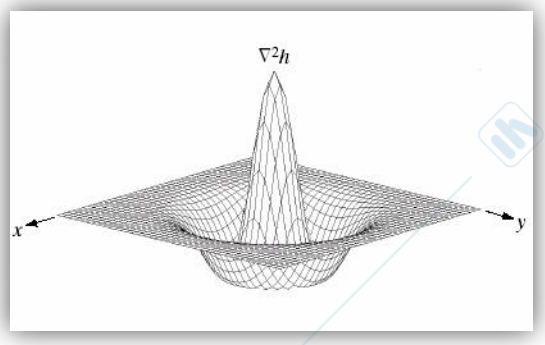
Una *tecnica avanzata* per l'individuazione degli edge che tiene conto (a differenza dei metodi visti sopra) della presenza di rumore e della natura degli edge è l'individuatore di Marr-Hildreth.
Le *ipotesi* di partenza di questi tizi sono:

- le variazioni di intensità sono dipendenti dalla scala dell'immagine e quindi la loro individuazione richiede l'uso di operatori di dimensioni diverse.
- un cambiamento di intensità improvviso da origine a un picco o lungo la derivata prima o a uno zero crossing nella derivata seconda.

Quindi le **caratteristiche principali** di un operatore utilizzato per l'individuazione degli edge dovrebbero essere:

- essere un operatore differenziale capace di calcolare un'approssimazione delle derivate prima e seconda in ogni punto dell'immagine
- poter essere regolato per agire a ogni scala selezionata, in modo tale che gli operatori più grandi possano essere utilizzati per individuare gli edge sfocati, mentre gli operatori più piccoli per individuare i dettagli più piccoli scarsamente visibili.

L'operatore che ha le seguenti caratteristiche è il **LoG**, il **Laplaciano del Gaussiano** o per via della sua forma **operatore a sombrero**.



L'espressione che lo caratterizza è la seguente:

$$\nabla^2 G(x,y) = \left[\frac{x^2+y^2+2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Questo operatore ha due **aspetti fondamentali**:

- la parte *gaussiana* dell'operatore sfoca l'immagine, quindi riduce il rumore sia nel dominio spaziale che in quello della frequenza ed è quindi meno probabile che vengano introdotti artefatti non presenti nell'immagine originale.
- il *laplaciano* ha il vantaggio di essere isotropico, cioè invariante per rotazione, e quindi ha le caratteristiche del sistema visivo umano e risponde in egual modo alle variazioni di intensità in ogni direzione della maschera, senza quindi la necessità di dover utilizzare maschere multiple per calcolare la risposta più forte.

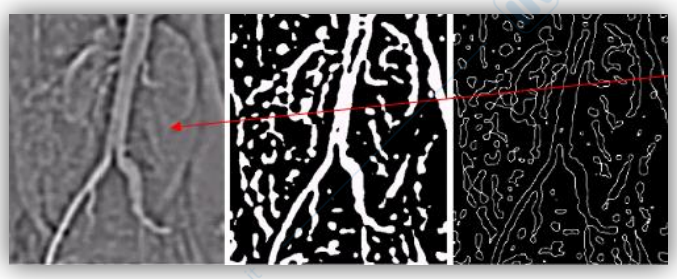
Per concludere l'algoritmo di Marr-Hildreth può essere riassunto nei seguenti passi:

1) filtrare l'immagine di input con un filtro passa basso gaussiano
2) calcolare il laplaciano dell'immagine ottenuta nel passo precedente
3) trovare gli zero crossing dell'immagine del passo precedente per determinare le posizioni degli edge.
   Gli zero crossing, in ogni pixel *p* dell'immagine filtrata, vengono individuati utilizzando un intorno *3x3* centrato in *p*; uno zero crossing in *p* implica che i segni di almeno dei due suoi pixel vicini opposti sono diversi. Ci sono quattro casi possibili*: sinistra/destra*, *sopra/sotto* e le due *diagonali*.
   Una caratteristica fondamentale di questo metodo è che utilizzare gli zero crossing per l'individuazione degli edge comporta che gli edge risultanti sono spessi *1* pixel.

Vediamo in un esempio le differenze tra un risultato ottenuto con Sobel e col LoG:

Immagine originale

Gradiente di Sobel



Rispettivamente: LoG; LoG sogliato; LoG zero crossing

Edge-linking e individuazione dei bordi

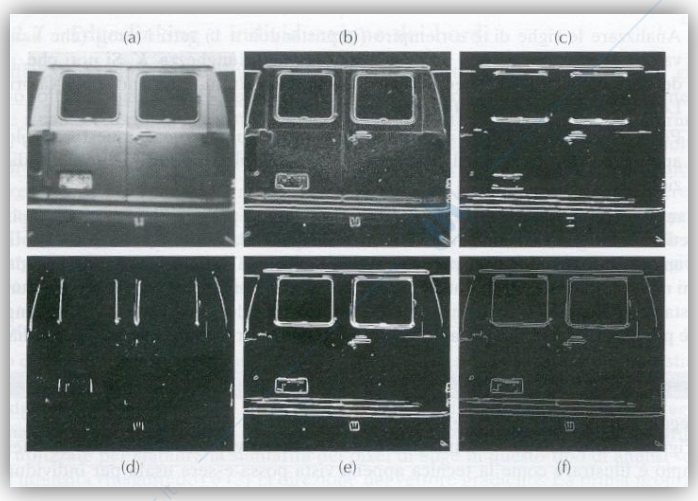
I metodi appena visti servono per individuare gli edge e idealmente dovrebbero restituire insiemi di pixel che giacciono soltanto sugli edge. In pratica, questi pixel non caratterizzano completamente gli edge a causa del rumore, di rotture negli edge dovute a illuminazione non uniforme e altre discontinuità. Risulta necessario quindi utilizzare, dopo aver individuato gli edge, un algoritmo di linking per poter collegare tra loro i pixel di edge.
Vediamo i tre metodi principali per effettuare edge-linking.

Elaborazione locale:

consiste nell'analizzare le caratteristiche dei pixel in un piccolo intorno su ogni punto (*x, y*) che è stato riconosciuto come punto di edge. Tutti i punti che soddisfano determinate proprietà di similarità vengono collegati, formando un edge di pixel.
Le due proprietà che servono per stabilire la similarità dei pixel di edge sono:

- *Intensità* (magnitudo)
- *Direzione del vettore gradiente*

Per collegare i pixel entrambi i criteri devono essere soddisfatti.
Vediamo un esempio:



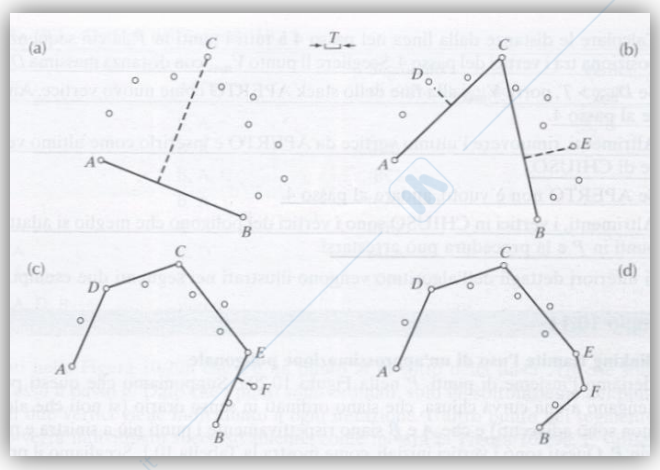
(a) Immagine del retro di un veicolo
(b) Immagine della magnitudo del gradiente
(c) Pixel di edge connessi in orizzontale
(d) Pixel di edge connessi in verticale
(e) L'OR logico delle due immagini precedenti
(f) Risultato finale ottenuto utilizzando l'assottigliamento morfologico

Elaborazione nelle regioni:

si utilizza quando si ha una conoscenza a priori delle regioni di interesse e dell'appartenenza di un pixel ad una determinata regione, ottenendo come risultato l'approssimazione del bordo della regione. I pixel vengono collegati tra loro perché si sa che fanno parte dei contorni di una regione.
In questo tipo di elaborazione si utilizzano delle tecniche che permettono di ottenere un'approssimazione delle caratteristiche essenziali della forma della regione, mantenendo la rappresentazione del bordo relativamente semplice.
In questa procedura ci sono due requisiti fondamentali: devono essere specificati due punti iniziali; tutti i punti devono essere ordinati. Questo ci permette di capire se ci troviamo davanti a una curva aperta o chiusa analizzando le distanze tra i punti:

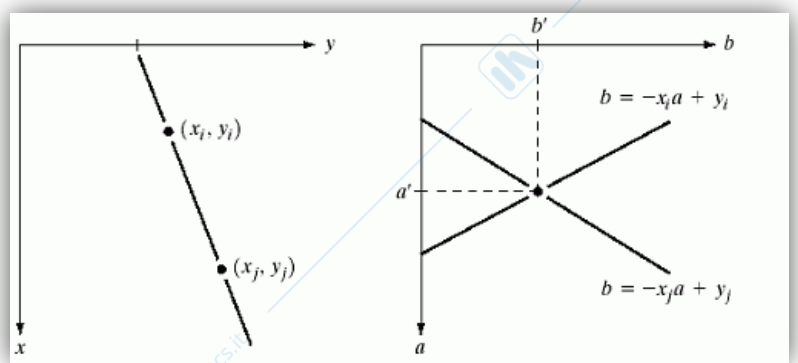
- *curva chiusa*: se la separazione tra punti tende ad essere uniforme
- *curva aperta*: se c'è una grande distanza tra due punti consecutivi nella sequenza ordinata rispetto alle distanze tra altri punti nella sequenza

Vediamo un esempio:



In questo esempio abbiamo un insieme di punti che rappresentano una curva aperta in cui gli estremi sono indicati con A e B; questi due punti sono i vertici del poligono.
Si comincia calcolando i parametri di una retta che passa da A a B. Ora calcoliamo la distanza perpendicolare da tutti gli altri punti della curva a questa linea e scegliamo il punto di massima distanza; se tale distanza supera una soglia T, il punto corrispondente, C, diventa un nuovo vertice (come si vede in figura).
Successivamente vengono definite le linee da A a C e da C a B e si calcolano le distanze da tutti i punti tra A e C alla linea AC. Il punto corrispondente alla distanza massima diventa vertice, D, se tale distanza supera la soglia T; altrimenti non viene dichiarato alcun nuovo vertice per quel segmento.
Lo stesso procedimento viene applicato ai punti tra C e B. Questa procedura iterativa viene ripetuta finché nessun punto soddisfa il test di sogliatura.
(d) Mostra il risultato finale, che è un'approssimazione ragionevole della forma di una curva che si adatta ai punti dati.

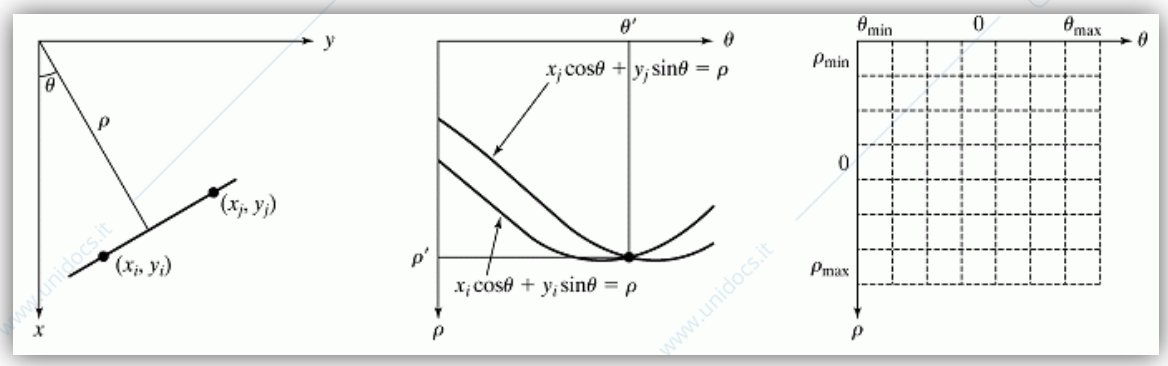
<u>*Elaborazione globale utilizzando la trasformata di Hough:*</u>
Quando non si ha nessuna conoscenza a priori sugli oggetti di interesse (sulle regioni e i bordi) e si ha soltanto un'immagine di edge, allora tutti i pixel sono possibili candidati per il linking e possono essere accettati o rifiutati sulla base delle loro proprietà *globali*, infatti ci si basa sulla direzione delle rette.
La **trasformata di Hough**, dati *n* punti in un'immagine, permette di trovare dei sottoinsiemi che si trovano su linee rette.
Per un punto (x_i, y_i), del piano *xy*, passano infinite rette e tutte soddisfano l'equazione generale della retta $y_i = ax_i + b$ al variare di *a* e *b*.
Possiamo considerare la retta $b = -ax_i + y_i$ considerando il piano *ab* (chiamato *spazio dei parametri*) in modo da ottenere l'equazione di una singola retta passante per (x_i, y_i).
Anche per un secondo punto (x_j, y_j) passa una retta nello spazio dei parametri che interseca (a meno che sono parallele) la retta associata a (x_i, y_i) nello stesso punto (a', b') dove a' è la pendenza e b' l'intersezione della linea che contiene sia (x_i, y_i) che (x_j, y_j) nel piano *xy*.
Per *tutti* i punti di questa retta (quella del piano *xy*) passano infinite rette nello spazio dei parametri che si intersecano in (a', b').
In questa immagine sono illustrati i concetti appena descritti:



In generale, le rette dello spazio dei parametri che corrispondono a tutti i punti (x_k, y_k) nel piano *xy* possono essere identificate, tirando fuori le linee principali nel piano, individuando i punti nello spazio dei parametri in cui si interseca il maggior numero di rette. Il problema è che all'avvicinarsi della retta alla direzione verticale, *a* (la pendenza) tende all'infinito. Per poter risolvere questo problema si utilizza la *rappresentazione normale della retta*, cioè:

$$x\cos\Theta + y\sin\Theta = \rho$$

L'interpretazione geometrica dei parametri Θ e ρ è illustrata nelle seguenti figure:



Notiamo dalla *figura (a)* che una linea verticale ha $\Theta = 90°$, mentre ρ è uguale all'intersezione positiva rispetto all'asse y; una linea orizzontale ha $\Theta = 0°$ mentre ρ è uguale all'intersezione positiva rispetto all'asse x.
Ogni curva sinusoidale in *figura (b)* rappresenta l'insieme delle linee che passano per un particolare punto (x_k, y_k) nel piano *xy*. Il punto di intersezione (ρ', Θ') corrisponde alla retta che passa sia per (x_i, y_i) sia per (x_j, y_j).
La *figura (c)* mostra il piano $\rho\Theta$ suddiviso nelle celle di accumulazione, dove (ρ_{min}, ρ_{max}) (Θ_{min}, Θ_{max}) sono gli intervalli consentiti per i valori dei parametri: $-90° \le \Theta \le 90°$ e $-D \le \rho \le D$, in cui D è la distanza massima tra gli angoli opposti di un'immagine. La cella con coordinate (*i,j*), con un valore di accumulazione A(*i,j*), corrisponde al quadrato associato alle coordinate dello spazio dei parametri (ρ_i, Θ_j). Queste celle vengono inizialmente poste a zero.
Per ogni punto (x_k, y_k) nel piano *xy* si suppone che Θ sia uguale a ciascuno dei valori di suddivisione permessi sull'asse Θ e si risolve per i ρ corrispondenti utilizzando la seguente equazione:

$$\rho = x_k\cos\Theta + y_k\sin\Theta$$

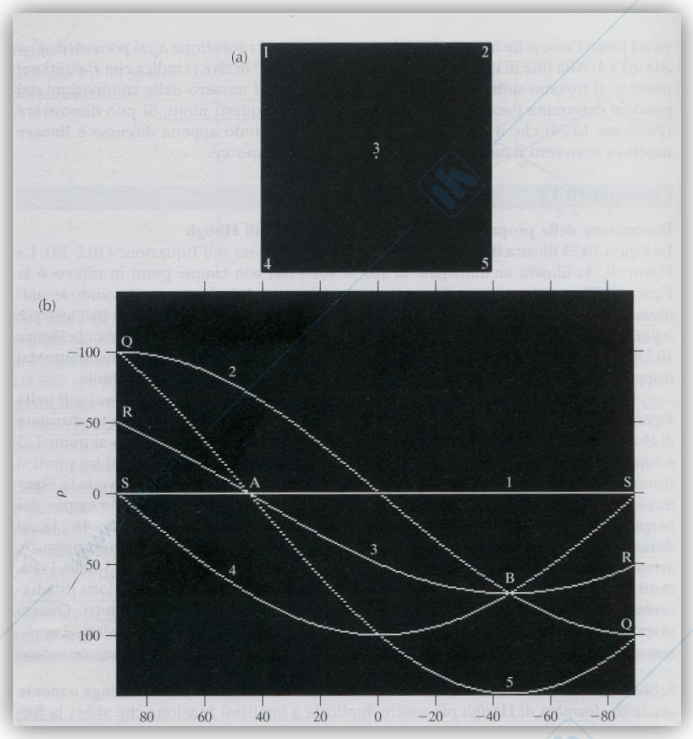
I valori di ρ risultanti vengono successivamente arrotondati al valore della cella più vicina lungo l'asse ρ. Se la scelta di Θ_p ha come risultato la soluzione ρ_q, si pone $A(p, q) = A(p, q) + 1$.
Alla fine della procedura, un valore di *P* in *A(i, j)* indica che *P* punti nel piano *xy* si trovano sulla retta $x\cos\Theta_j + y\sin\Theta_j = \rho_i$.
Il numero delle suddivisioni nel piano $\rho\Theta$ determina l'accuratezza della collinearità di questi punti.

La trasformata di Hough non è applicabile soltanto alle rette, ma può essere applicata a qualsiasi funzione che abbia la forma $g(\mathbf{v}, \mathbf{c}) = 0$ dove **v** è un vettore di coordinate e **c** è un vettore di coefficienti.
Ovviamente la complessità della trasformata di Hough dipende dal numero di coordinate e coefficienti nella rappresentazione.

Riassumendo quindi i passi da seguire sono i seguenti:

1) Ottenere un'immagine binaria dei contorni
2) Individuare le suddivisioni nel piano $\rho\Theta$
3) Esaminare i contatori delle celle di accumulazione con alte concentrazioni di pixel
4) Esaminare la relazione (di continuità) tra i pixel delle celle scelte.
   La continuità si basa sul calcolo della distanza tra i pixel disconnessi corrispondenti alla data cella di accumulazione. L'interruzione in una retta associata con una cella viene colmata se la sua lunghezza è minore di una data soglia.

Vediamo un esempio che chiarisce questi concetti astrusi:
La figura *(a)* mostra un'immagine con cinque punti in rilievo; la figura *(b)* mostra ognuno di questi punti posizionato sul piano *ρΘ*, utilizzando le suddivisioni di un'unità per gli assi *ρ* e *Θ*.
Il range dei valori *Θ* è $\pm 90°$, mentre il range dell'asse *ρ* è $\pm\sqrt{2}D$, dove *D* è la distanza tra gli angoli opposti dell'immagine.
Nella figura *(b)* notiamo che ogni curva ha una forma sinusoidale diversa; la linea orizzontale risultante dal mapping del punto 1 è un caso particolare di una sinusoide con ampiezza nulla.
I punti marcati con *A* e *B* mostrano le proprietà della detection della collinearità della trasformata di Hough.
Il punto *A* denota l'intersezione delle curve che corrispondono ai punti 1, 3 e 5 nel piano *xy* dell'immagine. La posizione del punto *A* indica che questi tre punti si trovano sulla linea retta passante per l'origine (*ρ=0*) e orientata a *45°*.



In modo analogo, le curve che si intersecano nel punto *B* nello spazio dei parametri indicano che i punti 2, 3 e 4 si trovano sulla linea retta orientata a *-45°*, la cui distanza dall'origine è *ρ=71* (la metà della distanza tra l'origine all'angolo opposto, arrotondata al valore intero più vicino).
Infine i punti marcati con *Q*, *R* e *S* illustrano il fatto che la trasformata di Hough esibisce una relazione di adiacenza riflessiva nei contorni a destra e a sinistra dello spazio dei parametri. Questa proprietà è il risultato del cambiamento di segno di *Θ* e *ρ* nei contorni a $\pm 90°$.

Sogliatura

La sogliatura, per la sua semplicità di implementazione e velocità computazionale, è una delle tecniche più importanti per la segmentazione delle immagini.

Sogliatura dell'intensità
Questo metodo ci permette, tramite l'istogramma di un'immagine, di estrarre gli oggetti che ci interessano dallo sfondo. Se ad esempio abbiamo un istogramma dell'intensità di un'immagine composta di oggetti chiari su uno sfondo scuro (quindi i valori di intensità sono raggruppati in due mode), possiamo estrarre gli oggetti dallo sfondo scegliendo una determinata soglia *T* che separa le due mode presenti nell'istogramma.
Viene definito *punto dell'oggetto* ogni punto (*x, y*) nell'immagine in cui *f(x,y) > T*; al contrario viene definito *punto dello sfondo* ogni punto (*x, y*) nell'immagine in cui *f(x,y) < T*;
L'immagine segmentata *g(x, y)* perciò è data da:

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

Abbiamo diversi tipi di sogliatura in base al valore di T:

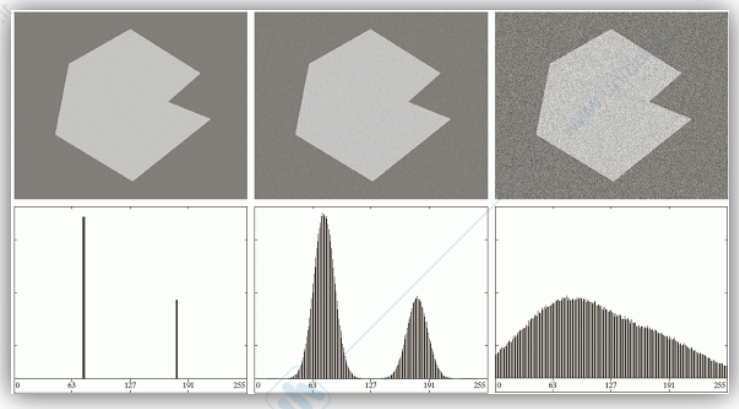
- *Sogliatura globale*:
  quando *T* è una costante applicabile all'intera immagine
- *Sogliatura variabile*:
  quando il valore di *T* cambia nel corso del processo

1. *Sogliatura regionale (o locale)*:
   sogliatura variabile in cui il valore di *T*, in un qualsiasi punto (*x, y*), dipende dalle proprietà dell'intorno del punto stesso
2. *Sogliatura dinamica (o adattativa)*:
   se *T* dipende dalle coordinate spaziali (*x, y*)

Il successo della sogliatura quindi dipende dall'altezza delle mode e dalle profondità delle valli che le separano. I *fattori* che influenzano le proprietà delle valli sono i seguenti:

1) La *distanza tra i picchi* (maggiore è la distanza tra i picchi, maggiore è la probabilità di riuscire a separare le mode)

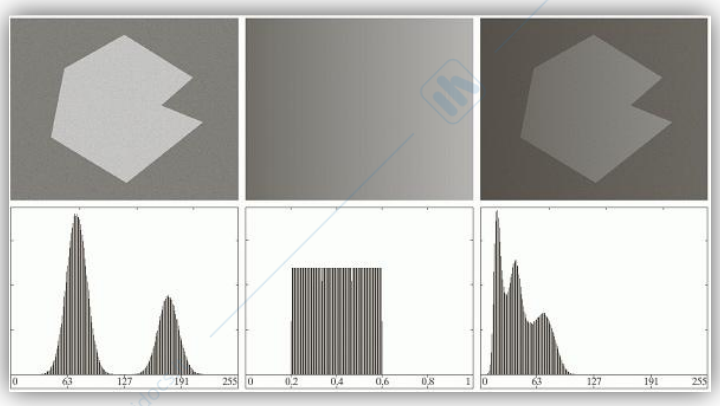
2) Il *rumore* contenuto nell'immagine (all'aumentare del rumore le mode si allargano).
   Per capire come il rumore modifichi le mode vediamo un'immagine di esempio:



   Nel terzo caso non è possibile trovare una soglia per segmentare l'immagine, se non applicando prima un'elaborazione aggiuntiva.
   L'elaborazione aggiuntiva che permette di risolvere il problema del rumore è lo smoothing dell'immagine, che va applicato quando non è possibile rimuovere il rumore alla sorgente. Maggiore è lo smoothing che applichiamo maggiore è la presenza di errori (distorsione) dei contorni.

3) Le relative *dimensioni* degli oggetti e dello sfondo.

4) L'uniformità della fonte di *illuminazione*.
   Anche in questo caso vediamo un esempio:



l'illuminazione non uniforme non permette di separare le mode.

5) L'*uniformità* delle proprietà di *riflettanza* dell'immagine.

L'illuminazione e la riflettanza sono importantissime per la riuscita della segmentazione ed è quindi necessario gestire questi fattori all'inizio del processo di segmentazione tramite tre approcci:

a) Correggere direttamente il problema
b) Provare a correggere il modello globale attraverso un processo di elaborazione
c) Lavorare attorno alle parti non uniformi attraverso una sogliatura variabile

Ora vediamo in dettaglio i due tipi di sogliatura accennati prima.

Sogliatura globale

Come abbiamo visto prima, la sogliatura è globale quando la soglia *T* è una costante applicabile all'intera immagine. Questo tipo di sogliatura possiamo applicarla quando le distribuzioni di intensità dei pixel dello sfondo e degli oggetti dell'immagine sono sufficientemente distinti.
Risulta quindi evidente che per ogni immagine è necessario stimare automaticamente il valore della soglia, per farlo si utilizza il seguente procedimento (*algoritmo di bisezione con media*):

1) Si stima un valore iniziale per la soglia globale *T*. Questo valore deve essere compreso tra il minore e il maggiore livello di intensità dell'immagine, meglio se si utilizza l'*intensità media*.
2) Si segmenta l'immagine utilizzando *T*. Questa operazione da come risultato due gruppi di pixel: G_1, che comprende i pixel i cui valori di intensità sono maggiore della soglia *T*; G_2 i cui pixel sono minori della soglia *T*
3) Si calcolano i valori di intensità media m_1 e m_2 per i rispettivi gruppi G_1 e G_2
4) Si calcola un nuovo valore di soglia, utilizzando la seguente espressione:
$$T = (m_1 + m_2)/2$$
5) Si ripetono i passi dal 2 al 4 fino a quando la differenza tra i valori assegnati a *T* non risulta minore di un parametro predefinito ΔT. Quindi maggiore è ΔT minori saranno i tempi di risposta.

Per effettuare la sogliatura globale è possibile utilizzare il **metodo di Otsu**, in quanto risulta essere un metodo ottimale perché massimizza la *varianza interclasse* e si basa su calcoli eseguiti sull'istogramma dell'immagine.
Il principio su cui si basa è che classi ben sogliate possono essere distinte rispetto ai valori di intensità dei loro pixel, e viceversa, che una soglia riuscendo ad ottenere la migliore separazione tra le classi rispetto ai valori di intensità risulterà la migliore.

L'algoritmo di Otsu può essere riassunto nei seguenti passi:+

1) Calcolare l'istogramma normalizzato dell'immagine. Le componenti dell'istogramma vengono denotate con p_i, $i = 0,1,2,\ldots,L-1$
2) Calcolare le somme cumulative, $P_1(k)$, per $k = 0,1,2,\ldots,L-1$
3) Calcolare le medie cumulative, $m(k)$, per k = 0,1,2,…,L-1
4) Calcolare l'intensità globale media, m_G
5) Calcolare la varianza interclasse $\sigma^2_B(k)$, per $k = 0,1,2,\ldots,L-1$
6) Calcolare la *soglia di Otsu*, k^*, cioè il valore di *k* per il quale $\sigma^2_B(k)$è massimo.
   Se il massimo non è unico, ottenere la soglia facendo la media dei valori di k corrispondenti ai differenti massimi calcolati
7) Ottenere la misura di separabilità η^*, tramite la seguente equazione (con $k = k^*$):

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

Utilizzo dei contorni per migliorare la sogliatura globale

Un metodo che permette di migliorare le caratteristiche dei picchi nell'istogramma (ricordiamo che devono essere alti, simmetrici e separati da profonde valli) è quello di considerare solo quei pixel che si trovano sui contorni o in prossimità di essi, quindi tra gli oggetti e lo sfondo.

Questo metodo fa in modo che:

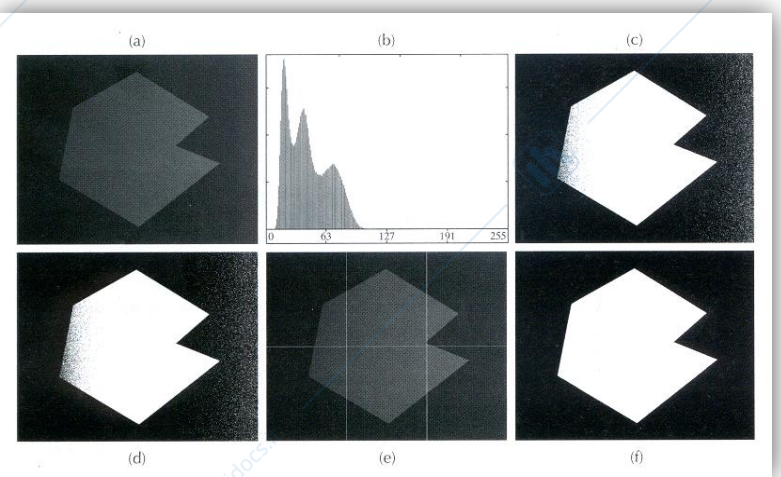
- l'istogramma dipenda di meno dalle dimensioni degli oggetti e dello sfondo
- l'istogramma risultante avrà dei picchi circa della stessa altezza
- la probabilità che un pixel appartenga a un oggetto è simile alla probabilità che appartenga allo sfondo (migliore simmetria delle mode)
- accentuazione della valle tra i picchi (utilizzando pixel che soddisfano determinate condizioni basate sul gradiente e sul laplaciano)

Il problema di questo metodo è che è necessario conoscere i contorni tra gli oggetti e lo sfondo. Un algoritmo che permette di fare ciò è il seguente:

1) calcolare un'immagine di edge *f(x,y)* o come magnitudo del gradiente o tramite il valore assoluto del laplaciano
2) individuare un valore di soglia *T*. Solitamente si utilizza un valore alto (90-esimo percentile) per fare in modo che dopo il calcolo del gradiente o dell'immagine laplaciana siano estratti solo pochi pixel
3) applicare la soglia all'immagine del passo 1 per ottenere un'immagine binaria $g_T(x,y)$
4) utilizzare l'immagine trovata nel passo 3 come *maschera* per poter individuare i pixel di *f(x, y)* che corrispondono agli edge forti. Nello specifico questo viene fatto calcolando l'istogramma utilizzando solo i pixel di *f(x,y)* che corrispondono alle posizioni dei pixel con valore 1 in $g_T(x,y)$
5) utilizzare l'istogramma calcolato nel passo 4 per segmentare *f(x, y)* in maniera globale, ad esempio tramite il metodo di Otsu

Sogliatura variabile

Come abbiamo visto prima, la sogliatura è locale quando il valore di *T* cambia nel corso del processo. Uno dei metodi più semplici di sogliatura variabile è quello di suddividere un'immagine in rettangoli non sovrapposti. Questo metodo permette di compensare l'illuminazione e la riflettenza non uniformi, infatti la dimensione dei rettangoli deve essere piccola abbastanza da fare in modo che l'illuminazione di ognuno di essi sia approssimativamente uniforme.
La debolezza della suddivisione dell'immagine consiste nel fatto che è efficace solo quando gli oggetti di interesse e dello sfondo occupano regioni di dimensioni confrontabili.
Vediamo un esempio:



(a) immagine rumorosa e ombreggiata
(b) istogramma
(c) segmentazione utilizzando l'algoritmo globale iterativo
(d) risultato ottenuto utilizzando il metodo di Otsu
(e) immagine suddivisa in sei sottoimmagni
(f) risultato dell'applicazione del metodo di Otsu in ogni singola sottoimmagine

Segmentazione basata sulle regioni

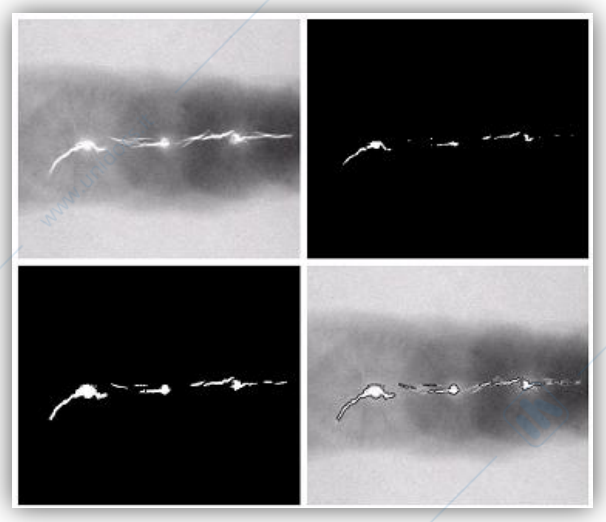
Questo tipo di segmentazione si basa sulla ricerca diretta delle regioni.
Vediamo due tecniche diverse:

Region growing (*crescita di regioni*)
è una procedura che raggruppa i pixel o le sottoregioni in regioni via via più grandi basandosi su criteri predefiniti di similarità e connettività. I pixel da cui si parte vengono chiamati *seed* (seme), e man mano si aggiunge a questi semi quei pixel dell'intorno che hanno delle proprietà predefinite simili a quelle del seme stesso.
Un problema di questo metodo è che bisogna stabilire una regola d'arresto, in quanto la crescita della regione dovrebbe cessare quando non ci sono più pixel che soddisfano i criteri per l'inserimento nella regione stessa.
Criteri aggiuntivi che aumentano la potenza dell'algoritmo utilizzano il concetto di *dimensione* e di *somiglianza* tra un pixel candidato e i pixel già implicati nel processo di crescita e la *forma della regione* stessa (implica che sia parzialmente disponibile un modello dei risultati previsti).

Un algoritmo di region growing può essere il seguente:
dove $f(x, y)$ matrice dell'immagine di input; $S(x,y)$ matrice seme che assegna il valore 1 a tutte le posizioni dei punti seme e 0 nelle posizioni rimanenti; Q predicato da applicare ad ogni posizione (x, y)

1) trovare tutte le componenti connesse in $S(x,y)$ ed erodere ogni componente connessa a un pixel. Assegnare quindi il valore 1 a tutti i pixel di questo tipo e ai rimanenti 0
2) formare un'immagine f_Q in modo che, nel punto (x,y), se l'immagine di input soddisfa il predicato Q si abbia $f_Q(x,y) = 1$, altrimenti $f_Q(x,y) = 0$
3) sia g l'immagine che si è formata dall'aggiunta a ogni punto seed in S di tutti i punti di valore 1 in f_Q che sono 8-connessi al dato seed
4) marcare ogni componente connessa in g con un'etichetta diversa per ogni regione

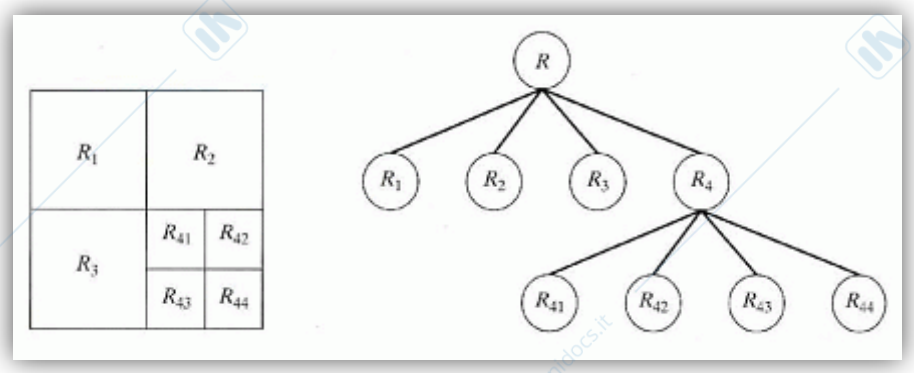
Vediamo un'immagine di esempio:



(a) immagine che mostra una saldatura difettosa
(b) immagine contenente i punti di seed iniziali
(c) risultato del region growing
(d) immagine che mostra in nero i contorni delle falle

Region splitting e merging
procedura che suddivide un'immagine in un insieme di regioni arbitrarie, ma disgiunte e successivamente le fonde e/o le separa in regioni che possano soddisfare le cinque condizioni di segmentazione viste all'inizio del capitolo.
In dettaglio considerando R la regione intera dell'immagine si sceglie un predicato Q. Si può segmentare R in regioni sempre più piccole in modo che per ogni regione R_i, $Q(R_i)$ = VERO. Quindi partendo dalla regione

intera, se *Q(R)* = FALSO si divide l'immagine in quadranti; se *Q* è falso per i quadranti che si ottengono questi a loro volta vengono suddivisi in ulteriori sottoquadranti e così via.
Per questo motivo la tecnica di scissione che si utilizza viene rappresentata in forma di *quadtrees*, cioè di alberi nei quali ogni nodo ha esattamente quattro ramificazioni. La radice dell'albero è l'immagine intera, mentre ciascun nodo corrisponde alla suddivisione di un nodo precedente.
Vediamo un esempio grafico:

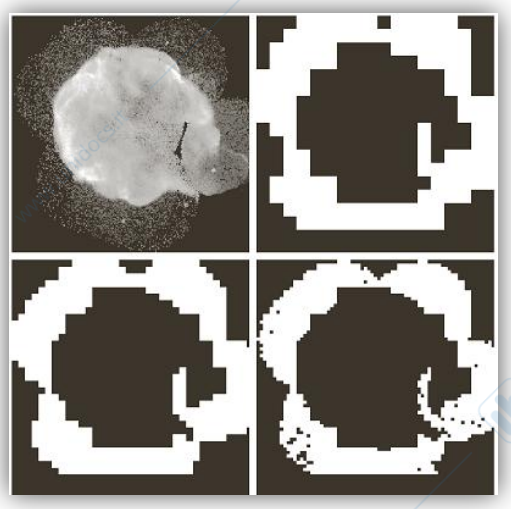


Se viene utilizzato soltanto il processo di *splitting* (scissione), la partizione finale conterrà soltanto regioni adiacenti con proprietà identiche. Per risolvere questo problema si utilizza il processo di *merging* (fusione) che unisce le regioni adiacenti i cui pixel soddisfano il predicato *Q*.

L'algoritmo quindi è composto dei seguenti passi:

1) scindere in quattro quadranti disgiunti ogni regione R_i per la quale $Q(R_i)$ = FALSO
2) quando non è più possibile separarle, applicare il processo di merging alle regioni adiacenti R_j e R_k per le quali $Q(R_j \text{ unione } R_k)$ = VERO
3) fermarsi quando il processo di merging non è più realizzabile

Vediamo un esempio:



L'obiettivo è quello di tagliare fuori l'anello di materia meno densa che circonda il centro dell'immagine.
Notiamo che la regione di interesse ha le seguenti caratteristiche: i dati hanno natura casuale, quindi la sua deviazione standard è maggiore di quella dello sfondo (che è quasi nulla) e della regione centrale (che è abbastanza sfocata); l'intensità media è maggiore rispetto allo sfondo scuro e minore della regione centrale più chiara.
Basandoci su queste caratteristiche possiamo definire un predicato Q, i pixel in una regione che soddisfano il predicato vengono etichettati di bianco, mentre tutti gli altri diventano neri.

(a) Immagine originale della supernova
(b) Risultato ottenuto con una regione quadrata 32x32
(c) Risultato ottenuto con una regione quadrata 16x16
(d) Risultato ottenuto con una regione quadrata 8x8

Notiamo che il risultato migliore si ottiene utilizzando delle regioni quadrate 16x16, perché nel caso 8x8 si introducono dei blocchi neri che non soddisfano il predicato, mentre nel caso 32x32 otteniamo una segmentazione "a blocchi".
In tutti i casi comunque le regioni segmentate separano del tutto la zona centrale dallo sfondo, ottenendo tre aree distinte: sfondo, zona densa, regione rada.
Questi risultati non possono essere ottenuti utilizzando i metodi più semplici quali la segmentazione basata su soglie o contorni.

Segmentazione mediante watershed morfologica

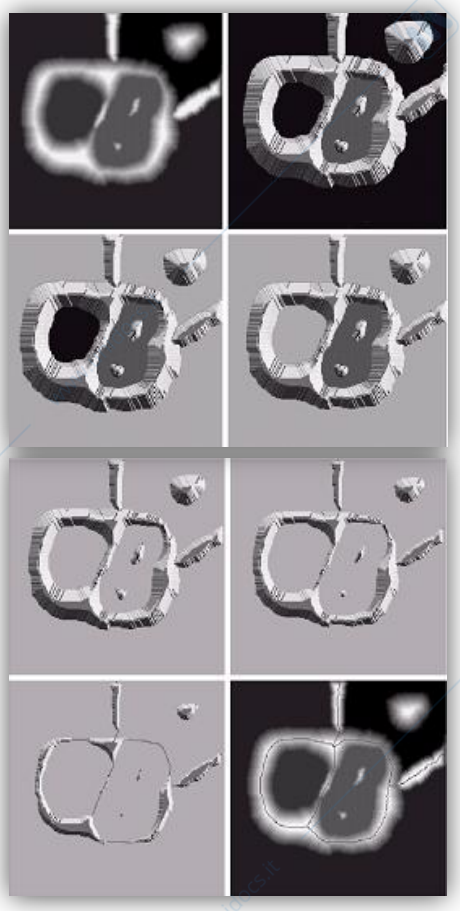
Il concetto di watershed si basa sulla visualizzazione di un'immagine in tre dimensioni: due coordinate spaziali e l'intensità. In questa interpretazione consideriamo tre punti chiave:

1. Punti che appartengono ad un minimo locale.
2. *Bacino di raccolta* (o watershed) del minimo: punti in cui una goccia d'acqua scorrerebbe verso uno di tali minimi.
3. *Linee di separazione* (linee di watershed): punti in cui l'acqua potrebbe egualmente cadere in più di un punto di minimo, che quindi formano le creste sulla superficie topografica.

L'obiettivo di questo metodo è quello di individuare le linee di watershed utilizzando il seguente sistema: ogni minimo viene perforato e l'intera topografia viene riempita d'acqua dal basso lasciando che l'acqua risalga uniforme attraverso questi fori (i minimi). Per prevenire la crescita del livello dell'acqua oltre i contorni dell'immagine, si chiude il perimetro dell'immagine con delle dighe che hanno un'altezza maggiore delle montagne, il cui valore viene determinato dal più alto valore di intensità presente nell'immagine. Quando l'acqua dei differenti bacini rischia di debordare, viene costruita una diga al fine di prevenire tale merging. Si continua con questo processo fino a raggiungere il massimo livello di allagamento (cioè il più alto valore di intensità dell'immagine). La diga finale corrisponde alle linee di watershed, che corrispondono al risultato della segmentazione desiderata, infatti le linee formano dei path connessi e di conseguenza dei *contorni continui tra le regioni*; questo metodo elimina il problema delle linee di segmentazione frammentate riscontrato in altri algoritmi.

Nota: tutto questo procedimento può essere fatto anche considerando l'acqua proveniente dall'alto, non solo dai minimi.

Vediamo un esempio:



(a) Immagine originale
(b) Visualizzazione topografica
(c) - (d) due fasi dell'allagamento
(e) Risultato dell'ulteriore allagamento
(f) Principio di fusione dell'acqua dei due bacini (è stata costruita una diga molto corta tra i due)
(g) Dighe più lunghe
(h) Linee di watershed finale (segmentazione)

Utilizzo dei marker

Questo algoritmo riscontra dei problemi in presenza di rumore e di irregolarità locali del gradiente, in quanto porta ad una *sovrasegmentazione* (oversegmentation). Per risolvere questo problema è necessaria una fase di pre-elaborazione nella quale si utilizzano i *marker*, cioè una componente connessa in un'immagine. La selezione dei marker è formata di due passi principali: pre-elaborazione; determinazione di un insieme di criteri che devono essere soddisfatti dai marker.

Ci sono due tipi di marker:

- *Marker interni*: sono associati agli oggetti della figura.
  Sono definiti come:
  a) una regione circondata da punti di altitudine elevata
  b) tale che i punti nella regione formino una componente connessa
  c) in cui tutti i punti nella componente connessa abbiano la stessa intensità.

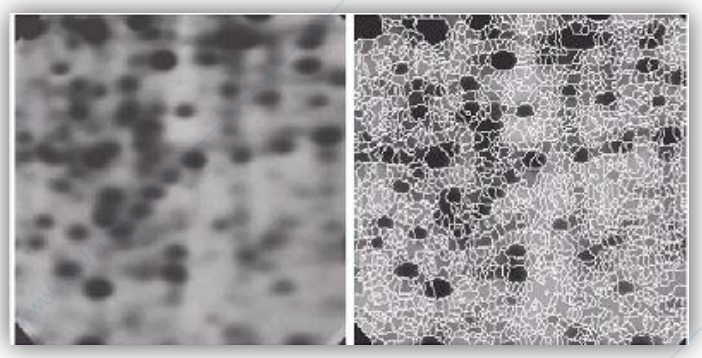
L'algoritmo watershed viene applicato come visto prima, ma con la condizione che solo i marker interni siano possibili minimi regionali. Le linee di watershed che si ottengono sono i marker esterni.

- *Marker esterni*: sono associati allo sfondo.
  Dividono efficacemente l'immagine in regioni, dove ogni regione contiene un singolo marker interno e parte dello sfondo.

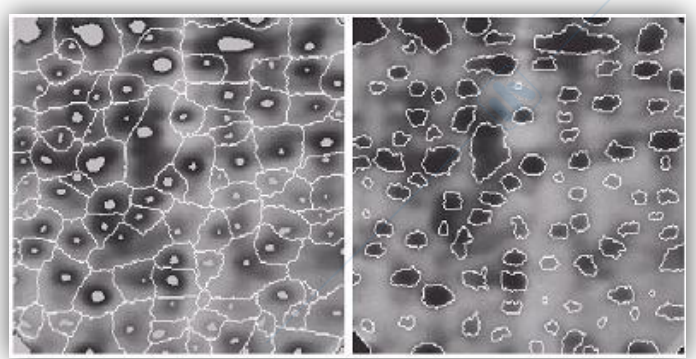
Ora per poter concludere la segmentazione è possibile applicare l'algoritmo di watershed a ogni singola regione.

La selezione dei marker che abbiamo visto si basa su descrizioni semplici, ma è possibile utilizzare descrizioni più complesse come dimensione, forma, posizione per poter aggiungere una conoscenza a priori nel processo di segmentazione simile ad alcune peculiarità del sistema visivo umano.

Vediamo un esempio:



(a) Immagine di elettroforesi
(b) Risultato dell'applicazione dell'algoritmo di segmentazione watershed sull'immagine gradiente. Otteniamo un numero eccessivo di regioni segmentate (oversegmentation)



(a) Immagine che mostra i marker interni (regioni in grigio chiaro) ed esterni (linee di watershed)
(b) Risultato della segmentazione. Molto migliore rispetto all'esempio precedente.

Uso del moto nella segmentazione

Nelle applicazioni di imaging, il moto si verifica quando si ha uno spostamento tra il sistema di sensori e la scena da visualizzare.

Tecniche spaziali

Di seguito vediamo l'uso del moto nella segmentazione nel dominio spaziale.

Approccio di base

Uno degli approcci più semplici per rilevare la variazione tra due scene $f(x, y, t_i)$ e $f(x, y, t_j)$, scattate in tempi diversi a t_i e t_j, è quello di confrontare le due immagini pixel per pixel, generando un'immagine differenza. Se abbiamo un'immagine di riferimento che contiene soltanto le componenti statiche e la confrontiamo con le immagini successive della stessa scena, ma che includono un oggetto in movimento, otteniamo la differenza in cui vengono rimossi gli elementi stazionari, mentre gli elementi non nulli corrispondono alle componenti in movimento dell'immagine.
Matematicamente parlando l'immagine differenza ottenuta dalle immagini scattate ai tempi t_i e t_j è può essere definita:

$$d_{ij}(x,y) = \begin{cases} 1 & \text{se } |f(x,y,t_i) - f(x,y,t_j)| > T \\ 0 & \text{altrimenti} \end{cases}$$

dove T è la soglia.
Quindi tutti i pixel di $d_{ij}(x, y)$ che hanno valore *1* sono gli *oggetti in movimento*. Questo metodo è applicabile soltanto se le due immagini vengono registrate spazialmente e se la componente di illuminazione è dentro i limiti stabiliti dalla soglia. Altrimenti può capitare che i valori *1* siano dovuti alla presenza di rumore e questo comporterebbe la presenza, nell'immagine differenza, di punti isolati; questi punti sono eliminabili ignorando le regioni che contengono un numero di elementi inferiore ad una certa quantità stabilita.

Differenze cumulative

Il caso trattato finora prevede la presenza di due immagini, ma possiamo avere anche una sequenza di immagini. In questo caso si parla di *immagine differenza cumulativa* (ADI, Accumulative Difference Image) che si ottiene confrontando l'immagine di riferimento con tutte le immagini successive della sequenza.
Il procedimento è il seguente:
per ogni posizione dei pixel nell'immagine cumulativa, ogni volta che ci si imbatte in una differenza tra l'immagine di riferimento e l'immagine in sequenza si incrementa un contatore. Quando si confronta il *k-esimo* frame con l'immagine di riferimento, ciascun valore dell'immagine cumulativa fornisce il numero di volte in cui l'intensità risulta, in quel punto, diversa rispetto al valore del pixel corrispondente nell'immagine di riferimento.
Ci sono tre tipi di ADI:

- *Assoluta*:

$$A_k(x,y) = \begin{cases} A_{k-1}(x,y) + 1 & \text{se } |R(x,y) - f(x,y,k)| > T \\ A_{k-1}(x,y) & \text{altrimenti} \end{cases}$$

  Contiene le regioni delle ADI negativa e positiva.
  E' possibile determinare la *direzione* e la *velocità* dell'oggetto in movimento.

- *Positiva*:

$$P_k(x,y) = \begin{cases} P_{k-1}(x,y) + 1 & \text{se } |P(x,y) - f(x,y,k)| > T \\ P_{k-1}(x,y) & \text{altrimenti} \end{cases}$$

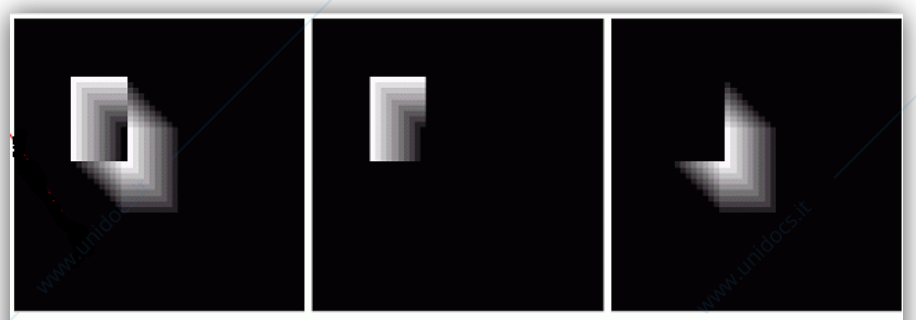
  L'area diversa da zero ha la stessa dimensione dell'oggetto in movimento, quindi i contatori non aumentano quando l'oggetto non è più sovrapposto rispetto alla sua posizione nel frame di riferimento.
  La posizione dell'ADI corrisponde alla *posizione* dell'oggetto in movimento nel frame di riferimento.

- *Negativa*:

$$N_k(x,y) = \begin{cases} N_{k-1}(x,y) + 1 & \text{se } |N(x,y) - f(x,y,k)| > T \\ N_{k-1}(x,y) & \text{altrimenti} \end{cases}$$

  E' possibile determinare la *direzione* e la *velocità* dell'oggetto in movimento.

Dove *R(x, y)* è l'immagine di riferimento; *k* denota t_k, quindi *f(x, y, k) = f(x, y, t_k)*.
Vediamo un'immagine chiarificatrice:



a b c
FIGURE 10.49 ADIs of a rectangular object moving in a southeasterly direction. (a) Absolute ADI. (b) Positive ADI. (c) Negative ADI.

Il concetto chiave in questo metodo è che bisogna avere un'immagine di riferimento (contenente solo elementi stazionari) attraverso la quale poter effettuare i confronti. In pratica però è difficile ottenere un'immagine di riferimento che contiene solo elementi stazionari, risulta quindi necessario generarla partendo dall'insieme delle immagini che contengono uno o più oggetti in movimento; questo viene fatto quando si ha a che fare con scene molto affollate.
Il procedimento da seguire per generare l'immagine di riferimento è il seguente:

- Si considera come immagine di riferimento la prima immagine della sequenza.
- Quando tutte le componenti non stazionarie si sono allontanate dalle loro posizioni nell'immagine di riferimento, il corrispondente sfondo nel frame può essere duplicato nella posizione che in origine era occupata dagli oggetti nel frame di riferimento.
- Quando tutti gli oggetti in movimento sono stati completamente rimossi dalle loro posizioni originali, si può creare un'immagine di riferimento che contiene solo le componenti stazionarie.

Nota: lo spostamento degli oggetti può essere stabilito osservando le variazioni dell'ADI positiva, in quanto abbiamo visto che permette di determinare la posizione iniziale di un oggetto in movimento.

Vediamo un esempio:

(a) - (b) due frame di una sequenza
(c) L'automobile è estratta da (a) e il background è stato restaurato attraverso la corrispondente area in (b).

Per ottenere questo risultato sono stati applicati esattamente i passi descritti sopra.