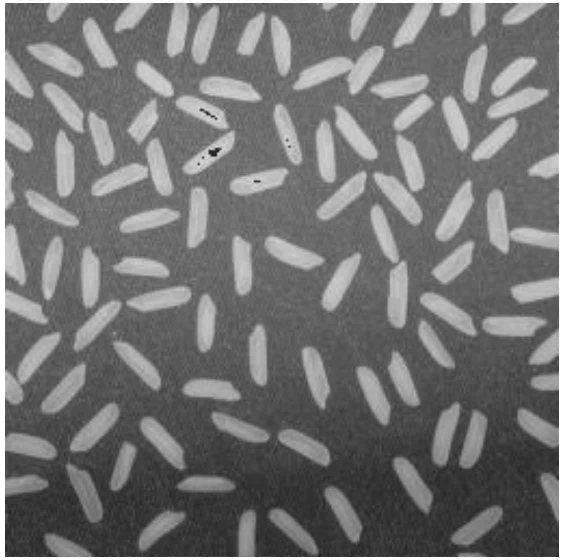
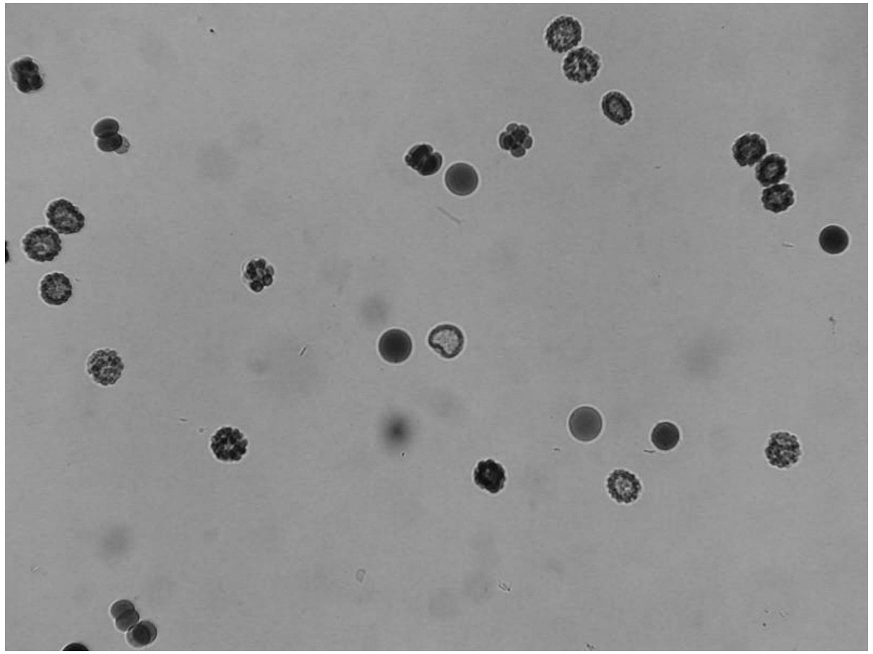
Problema:

Abbiamo alcune immagini che contengono oggetti simili e vogliamo contarli. In generale: calcolo e individuazione delle componenti connesse (per esempio nel senso della 8-connessione).
Occorre considerare:

- Tipo di immagini
- Rumore presente
- Binarizzazione (che strategia adottare)
- Operatore morfologico per le componenti connesse
- Rappresentazione risultati





Nel nostro caso:

- Immagine a livelli di grigio
- Rumore Gaussiano additivo
- Procederemo trovando i bordi:
    1. utilizzare proprio codice con operatori discreti per derivate parziali con ***immagine regolarizzata***, norma-1 per il gradiente, sogliatura sulla norma (sperimentare diversi valori di soglia)
    2. utilizzare il metodo di Canny implementato in Matlab (Image processing toolbox): funzione **edge**
- Utilizzare metodo per il conteggio delle componenti connesse.
- Rappresentare con colori differenti (funzione Matlab **pcolor**)

Utilizzare **help edge** e **help pcolor** per i parametri e l'utilizzo delle due funzioni.

Nel seguito si utilizzano alcune slide del Prof. Raffaele Cappelli – Ingegneria e scienze informatiche – Università di Bologna

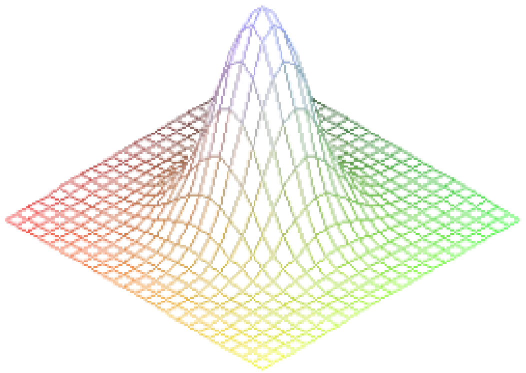
Canny edge detector

- Il metodo di Canny produce edge connessi che possono essere efficacemente utilizzati per le successive fasi di elaborazione.
- L’approccio prevede le seguenti fasi:
  - 1) Smoothing gaussiano dell’immagine
  - 2) Calcolo del gradiente
  - 3) Soppressione dei non-massimi in direzione ortogonale all’edge
  - 4) Selezione degli edge significativi mediante isteresi
- I risultati dipendono da alcuni parametri:
  - σ – ampiezza della gaussiana nella prima fase
  - Dimensione del filtro nella prima fase
  - T1 e T2 – soglie per l’isteresi nell’ultima fase

Canny – 1) Smoothing Gaussiano

- Gli elementi sono pesati secondo una funzione gaussiana.
  - Il parametro σ controlla l'ampiezza della gaussiana e quindi l'entità della regolarizzazione.
  - Il filtro è separabile: conviene effettuare la convoluzione con due filtri 1D (identici fra loro)
- Approssimazione con valori interi (per maggiore efficienza)
  - Il termine (1/sqrt(...)) può essere trascurato, in quanto dopo il calcolo è comunque necessario normalizzare gli elementi rispetto alla somma dei pesi
  - Esempio (σ=1) di una possibile soluzione:

$$G_{2D}(x,y,\sigma) = \frac{1}{2\pi\sigma^2} \cdot \exp^{-\frac{x^2+y^2}{2\sigma^2}}$$



$$G_{1D}(t,\sigma) = \frac{1}{\sqrt{2\pi\sigma}} \cdot \exp^{-\frac{t^2}{2\sigma^2}}$$

$$G_{2D}(x,y,\sigma) = G_{1D}(x,\sigma) \cdot G_{1D}(y,\sigma)$$

0.135	0.607	1	0.607	0.135

Filtro 1D (ignorando termine moltiplicativo)

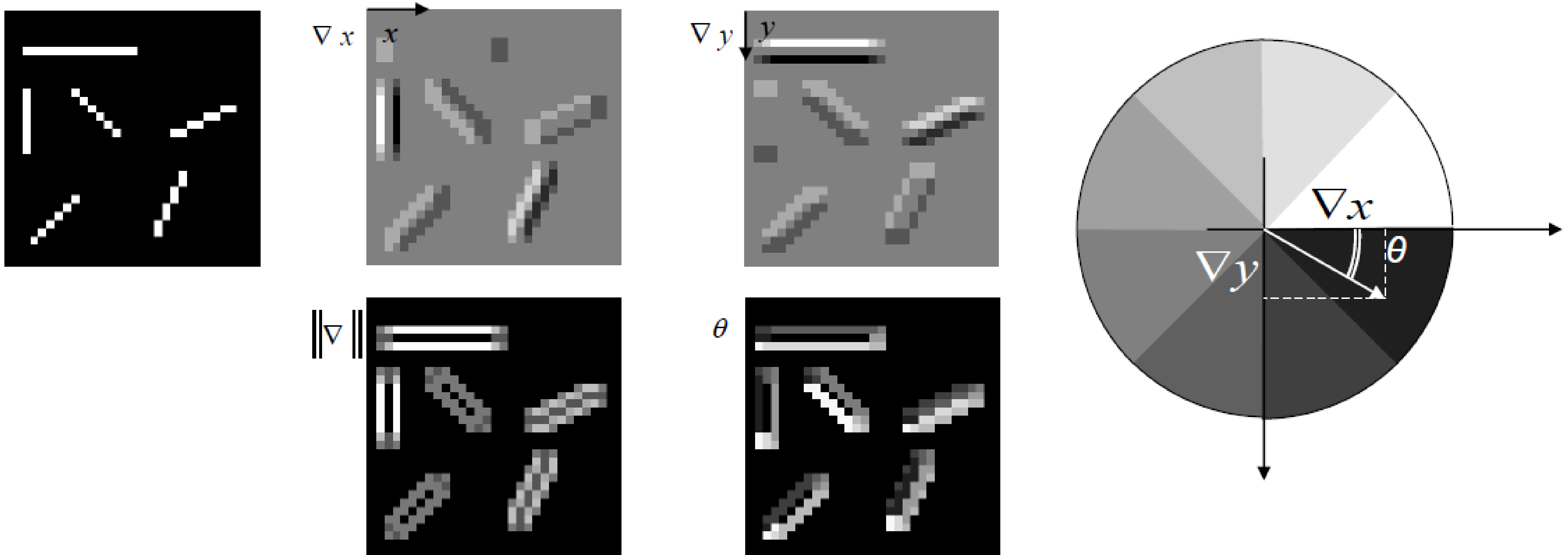
$\frac{1}{17} \cdot$

1	4	7	4	1

Approssimazione intera e normalizzazione

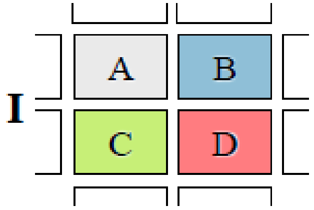
Canny – 2) Calcolo del gradiente

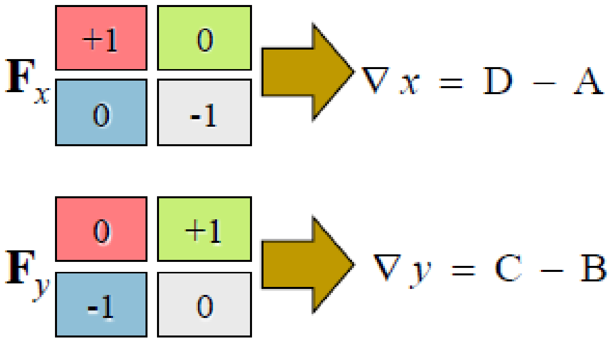
- Operatori da utilizzare
  - Dato che lo smoothing dovrebbe aver rimosso la maggior parte del rumore, l'implementazione più efficiente si avvale degli operatori di Roberts
  - Risultano tuttavia di più semplice applicazione gli operatori di Prewitt, in quanto non ci si deve preoccupare della rotazione degli assi di 45°
- Esempio del risultato della fase 2 utilizzando gli operatori di Prewitt:

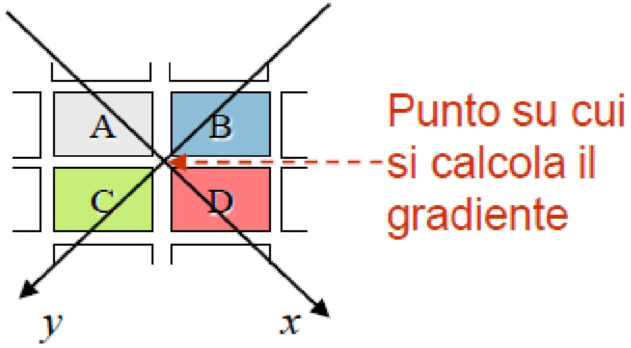


Calcolo del gradiente – Operatori di Roberts

- Convoluzione con una coppia di filtri 2x2
  - Misurano il gradiente lungo assi ruotati di 45° rispetto agli assi dell'immagine (con origine in alto a sinistra)
    - Questo consente di calcolare le due componenti del gradiente nel medesimo punto (esattamente al centro di quattro pixel adiacenti)
- Caratteristiche
  - Pro: possono essere calcolati in modo rapido ed efficiente
  - Contro: sono molto sensibili al rumore
- Calcolo del gradiente
  - Per maggiore efficienza il modulo è spesso approssimato come somma dei moduli delle due componenti
  - L'orientazione va riportata alle coordinate canoniche (ruotando di 45°)







$$\|\nabla\| = \sqrt{\nabla x^2 + \nabla y^2} \cong |\nabla x| + |\nabla y|$$

$$\theta = \arctan_q(\nabla y, \nabla x) + \frac{\pi}{4}$$

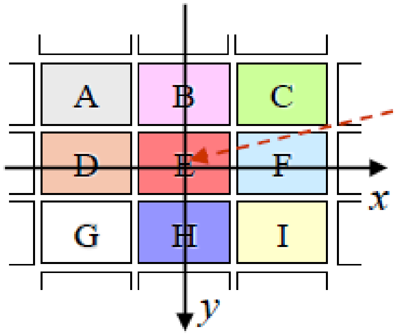
Operatori di Prewitt e di Sobel

- Prewitt: due filtri 3x3
  - Meno sensibili a variazioni di luce e rumore
  - Calcolo del gradiente lungo una direzione e media locale (smooth) lungo la direzione ortogonale
  - Simmetrici rispetto al punto di applicazione
  - Assi x e y orientati in modo tradizionale
    - Origine in alto a sinistra (se si lavora con l'origine in basso è sufficiente invertire il filtro y)
- Sobel: due filtri 3x3
  - Peso maggiore al pixel centrale

$$\mathbf{I} = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix}$$

$$\mathbf{F}_x = \frac{1}{3} \cdot \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \Rightarrow \nabla x = \tfrac{1}{3}\left[(C + F + I) - (A + D + G)\right]$$

$$\mathbf{F}_y = \frac{1}{3} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \Rightarrow \nabla y = \tfrac{1}{3}\left[(G + H + I) - (A + B + C)\right]$$



Punto su cui si calcola il gradiente

$$\|\nabla\| = \sqrt{\nabla x^2 + \nabla y^2}$$

$$\theta = \arctan_q(\nabla y, \nabla x)$$

$$\mathbf{F}_x = \frac{1}{4} \cdot \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \mathbf{F}_y = \frac{1}{4} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Canny – 3) Soppressione dei non-massimi

- Obiettivo
  - Eliminare dall'immagine modulo-gradiente i pixel che non sono massimi locali rispetto all'orientazione del gradiente
- Diversi approcci possibili
  - Il più semplice consiste nell'analizzare l'intorno 3x3 di ogni pixel, eliminando i pixel che non rispettano la condizione di massimo locale lungo la direzione del gradiente (ortogonale all'edge)

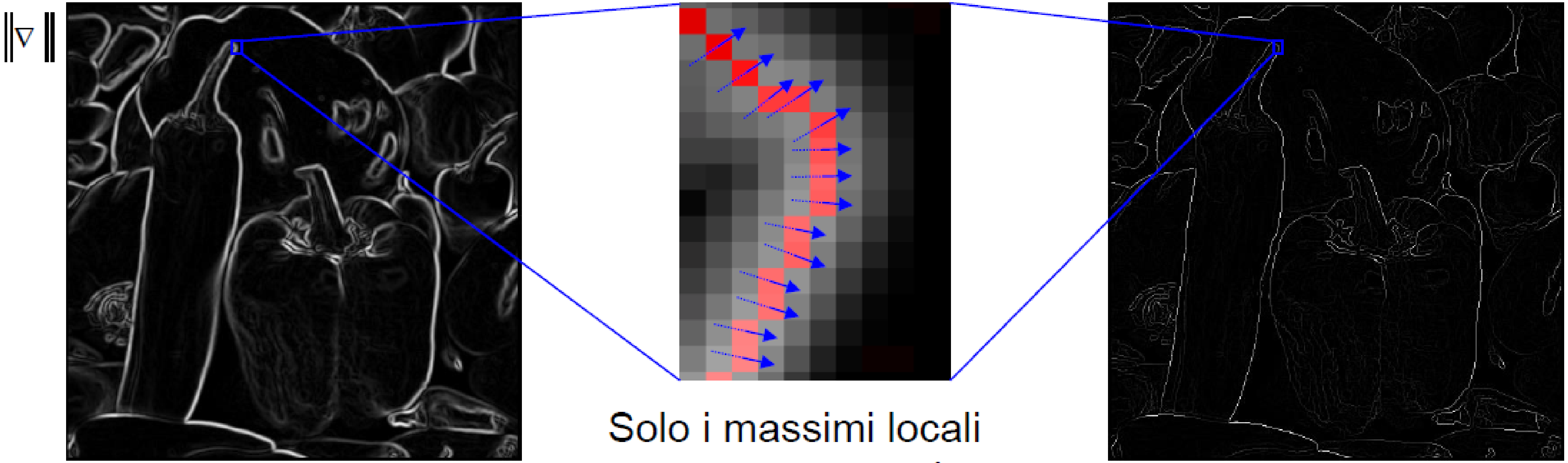


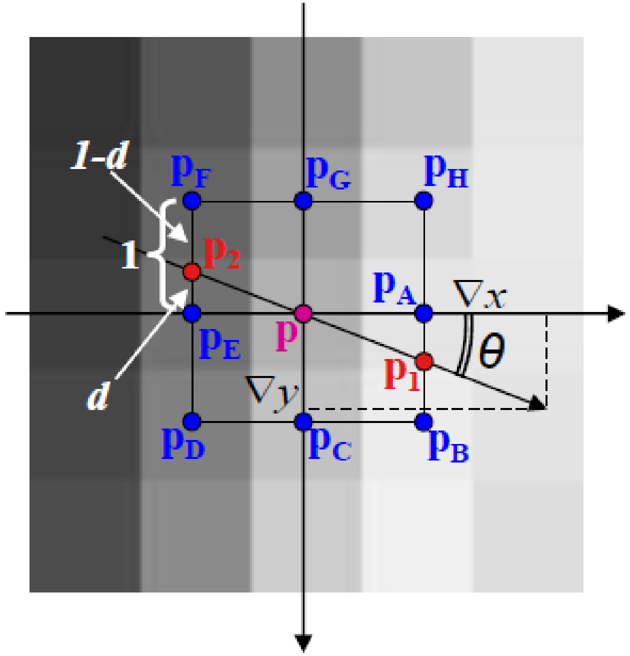
Immagine iniziale

Solo i massimi locali sono mantenuti

Risultato

Canny – 3) Soppressione dei non-massimi (2)

- Verifica della condizione di massimo locale nell'intorno 3x3
  - Si stima il modulo del gradiente nei punti $\mathbf{p}_1$ e $\mathbf{p}_2$ mediante interpolazione lineare
    - La figura mostra l'interpolazione nel caso l'orientazione del gradiente appartenga al primo ottante; gli altri casi sono analoghi
  - Il pixel p viene conservato solo se $\|\nabla[\mathbf{p}]\| \geq \|\nabla[\mathbf{p}_1]\| \wedge \|\nabla[\mathbf{p}]\| \geq \|\nabla[\mathbf{p}_2]\|$
  - Questo approccio non garantisce edge di spessore unitario (benché in genere lo siano)
    - A tale fine può essere utilizzata una procedura di thinning al termine dell'intero algoritmo



$$\|\nabla[\mathbf{p}_1]\| \cong d \cdot \|\nabla[\mathbf{p}_B]\| + (1-d)\|\nabla[\mathbf{p}_A]\|$$

$$\|\nabla[\mathbf{p}_2]\| \cong d \cdot \|\nabla[\mathbf{p}_F]\| + (1-d)\|\nabla[\mathbf{p}_E]\|$$

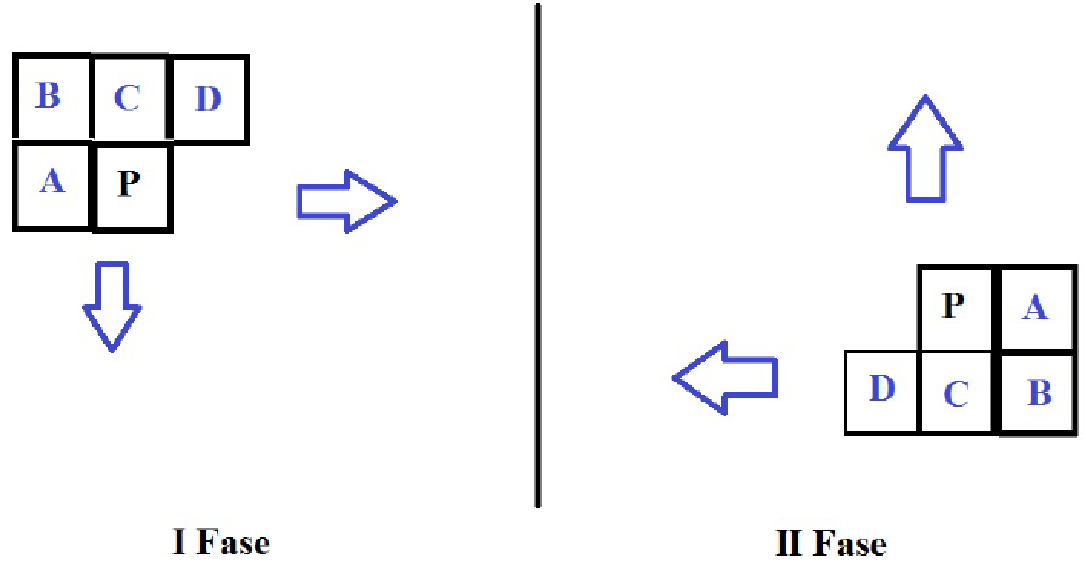
$$d = \frac{\nabla y[\mathbf{p}]}{\nabla x[\mathbf{p}]}$$

Canny – 4) Selezione finale degli edge

- Isteresi
  - Al fine di selezionare solo gli edge significativi (tralasciando edge "spuri"), ma evitando allo stesso tempo la frammentazione, si utilizza il concetto di isteresi: vengono impiegate due soglie T_1 e T_2, con $T_1 > T_2$ per scremare ulteriormente i massimi locali ottenuti nella fase precedente:
    - Sono inizialmente considerati validi solo i pixel in cui il modulo del gradiente è superiore a T_1.
    - I pixel il cui modulo è inferiore a T_1 ma superiore a T_2 sono considerati validi solo se adiacenti a pixel validi.
  - T_1 e T_2 sono tipicamente espresse come valori fra 0 e 1 (il modulo del gradiente va normalizzato nello stesso intervallo per permettere il confronto con le due soglie).
  - Una corretta scelta di T_1 e T_2, così come un'adeguata scelta di σ nella prima fase, sono molto importanti per ottenere gli effetti desiderati.
    - La scelta dipende solitamente dall'applicazione e sono tipicamente necessari vari esperimenti per giungere ai valori ottimali dei parametri.

Semplice algoritmo per estrarre componenti connesse

Consideriamo un semplice algoritmo per il calcolo delle componenti connesse all'interno di un'immagine binaria (oggetti pixel=1, sfondo pixel=0). Se si hanno N pixel non nulli si inizializza una matrice con le stesse dimensioni dell'immagine originale ed etichettando tutti i pixel progressivamente con interi da 1 fino ad N. Di seguito si alternano due passaggi per tutti i pixel:



il primo dall'alto al basso e da sinistra verso destra, il secondo dal basso verso lalto e da destra verso sinistra. Per ogni pixel in ogni passaggio si sostituisce all'attuale etichetta del pixel la più piccola presente in un opportuno intorno. Se per tutti i pixel nei due passaggi i valori rimangono inalterati ci si ferma.

In figura si mostra un esempio semplice: (a) immagine iniziale (attenzione 1 significa presenza di un oggetto); (b) inizializzazione; (c) dopo passata dall'alto verso il basso, da sinistra verso destra; (d) dopo passata dal basso verso l'alto, da destra verso sinistra.