

Corso di Percezione Robotica (PRo)



C. Modulo di Percezione Attiva

Visione robotica

Cecilia Laschi

ARTS Lab, Scuola Superiore Sant'Anna

cecilia@arts.sssup.it

050-883486

Sommario della lezione (1/2)

- Immagini digitali
 - formazione dell'immagine
 - definizioni di immagine digitalizzata, connettività e distanza
 - operatori puntuali, locali e globali
- Pre-elaborazione (early processing):
 - filtraggio
 - rilevamento di bordi
 - sogliatura

Riferimenti bibliografici:

Fu, Gonzalez, Lee, "Robotica", McGraw-Hill

Ballard & Brown, "Computer vision", Prentice Hall

Sommario della lezione (2/2)



- Segmentazione dell'immagine
 - Rilevamento dei contorni
 - Rappresentazione dei contorni
- Visione stereoscopica
 - Principi fondamentali
 - L'algoritmo di Marr & Poggio

Riferimenti bibliografici:

Fu, Gonzalez, Lee, "Robotica", McGraw-Hill

Ballard & Brown, "Computer vision", Prentice Hall

Principali classi di tecniche di elaborazione delle immagini digitali

PRE-ELABORAZIONE (EARLY PROCESSING)

- Elaborazione dei valori dei pixel dell'immagine a livello puntuale o locale

Es:

- FILTRAGGIO
- RILEVAMENTO DEI BORDI

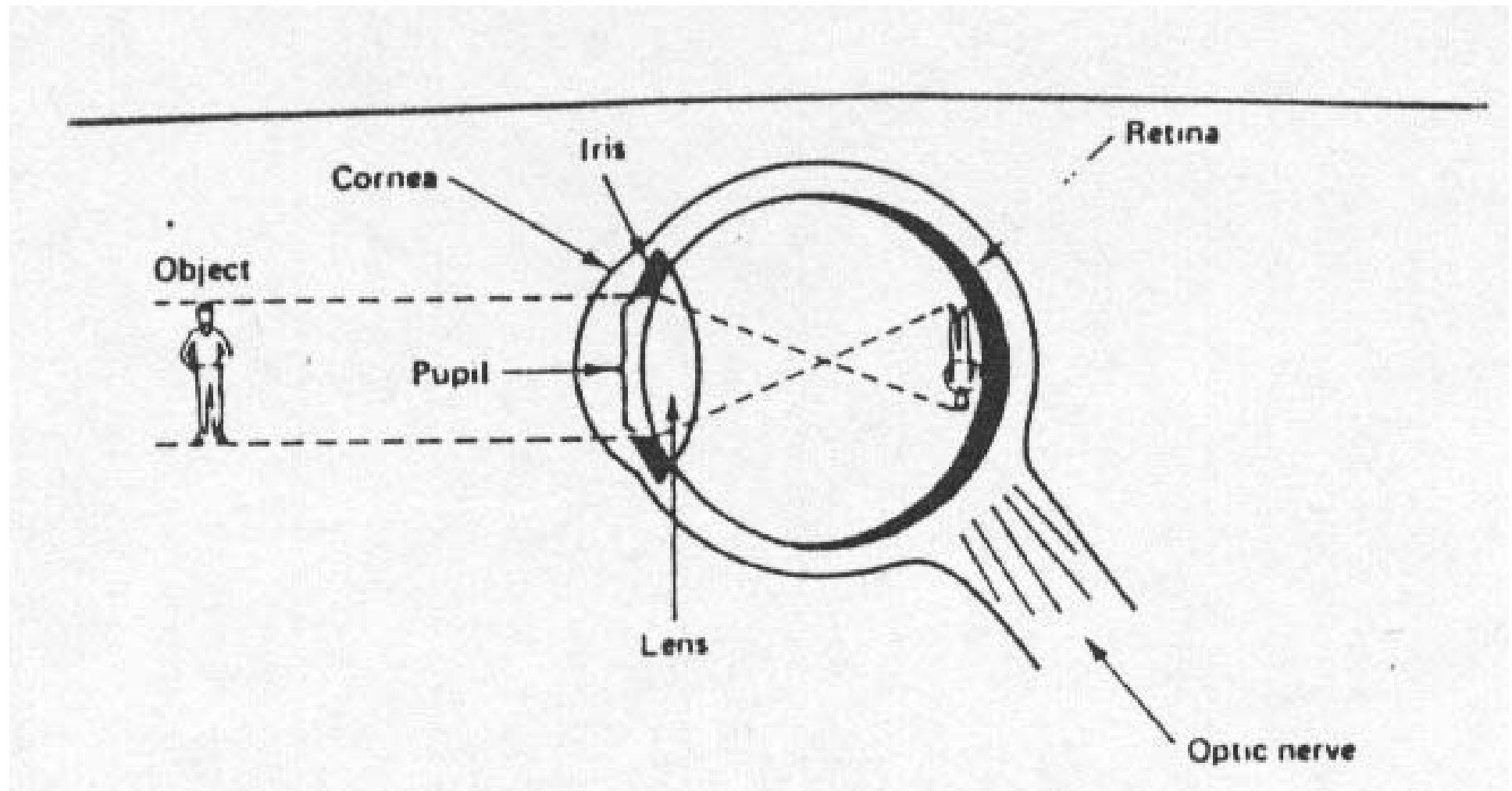
SEGMENTAZIONE

- Rilevamento delle parti che costituiscono una scena
- Elaborazione dell'immagine a livello globale

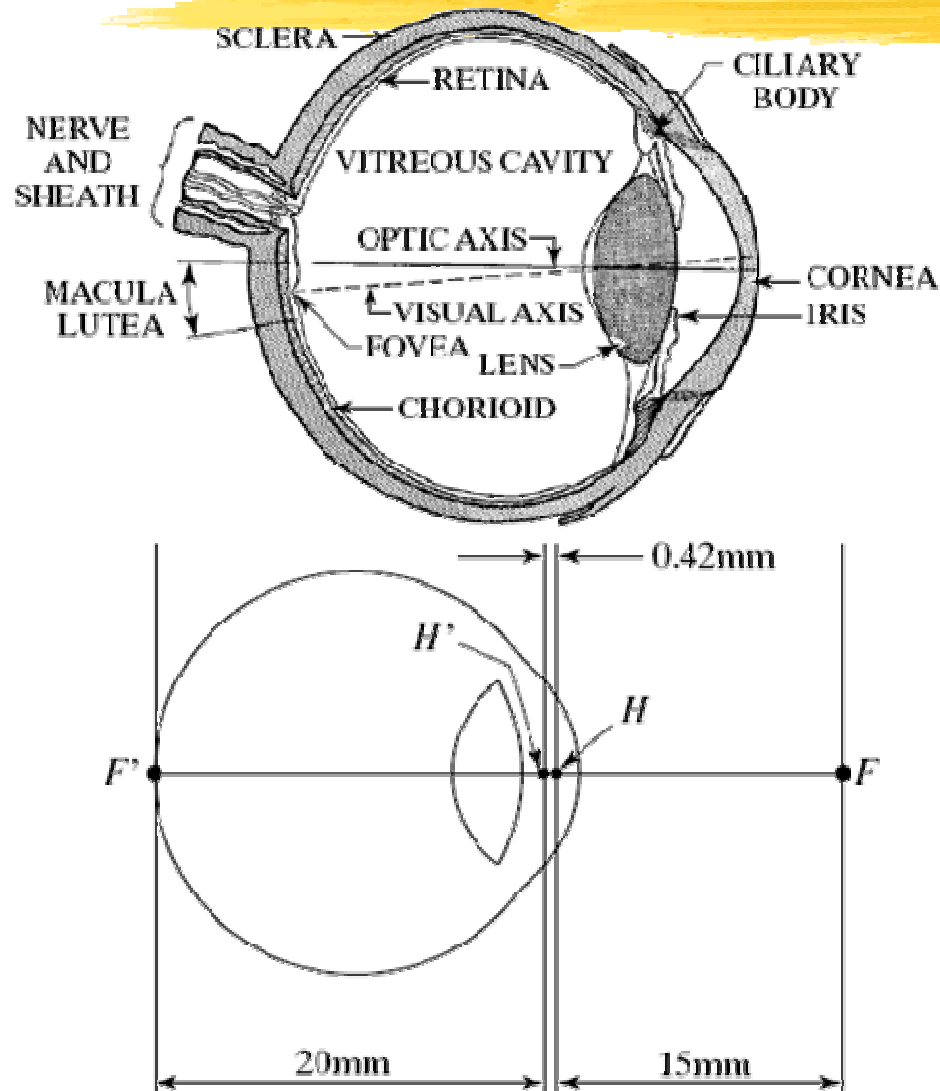
Es:

- RILEVAMENTO CONTORNI:
elementi basati sulla **discontinuità**
- RILEVAMENTO REGIONI:
elementi basati sulla **uniformità**

Formazione dell'immagine nell'occhio

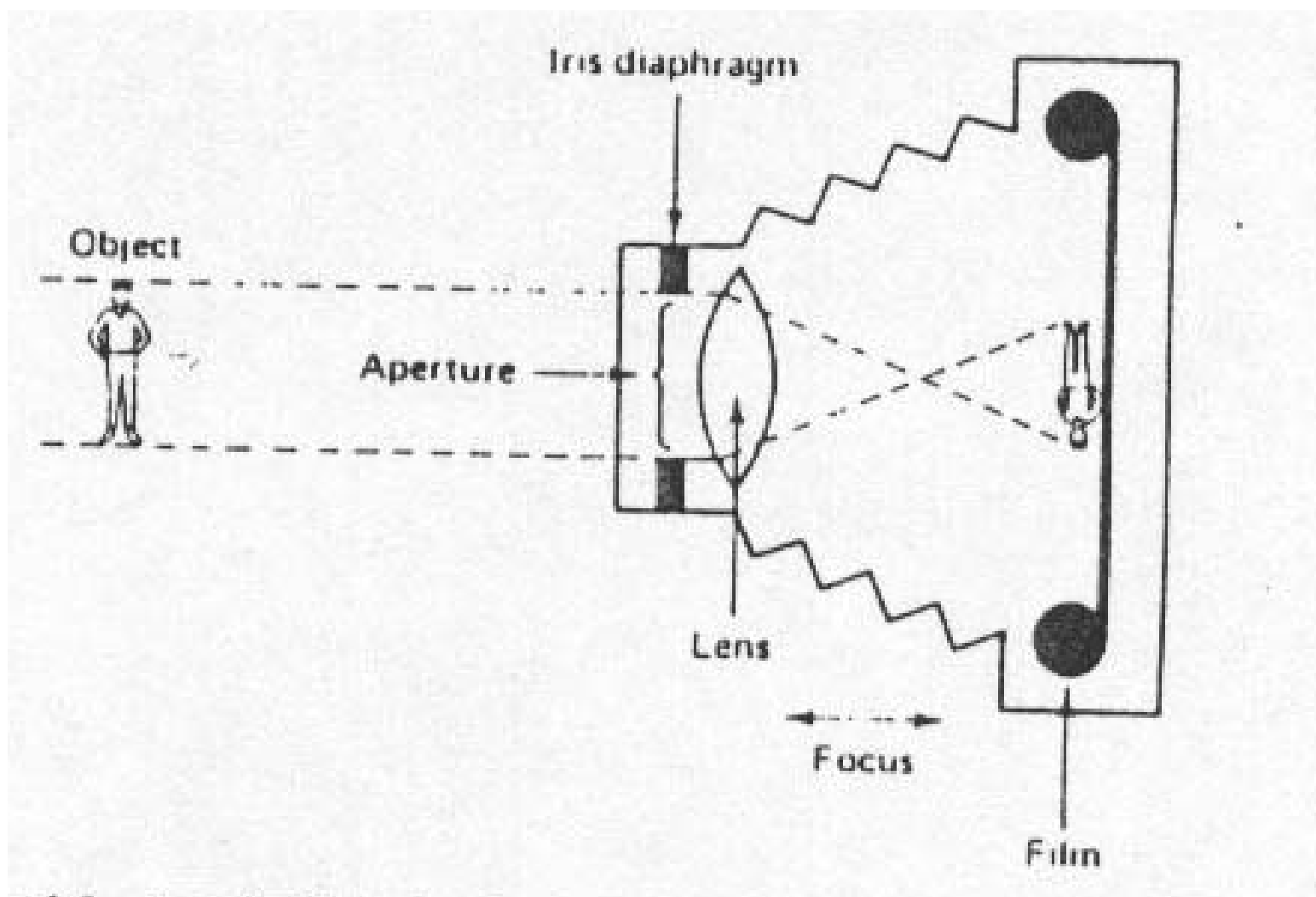


Sensore biologico

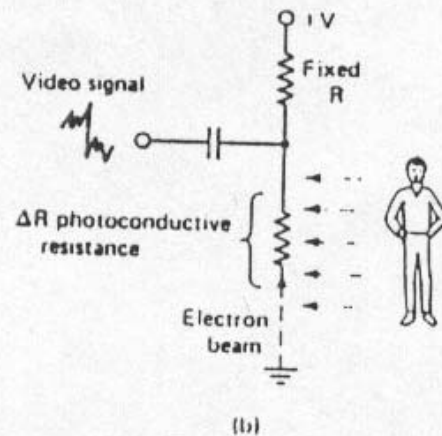
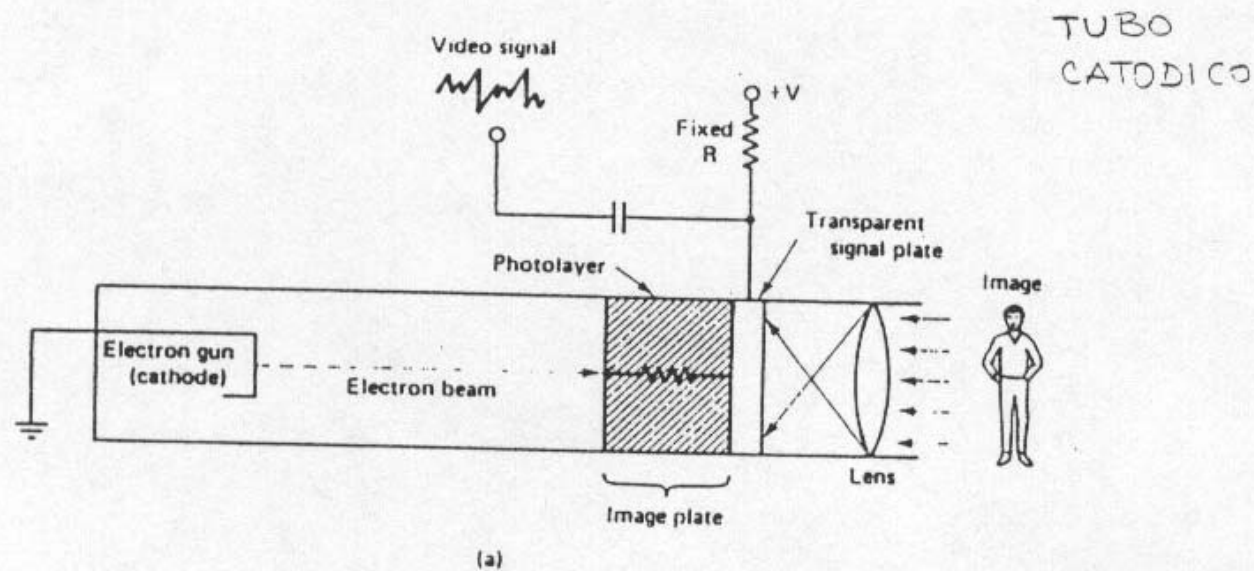


- Controllo dell'esposizione
 - Iride, pupilla (1-8 mm)
- Rifrazione della luce
 - Cornea, cristallino
- Formazione immagine
 - Retina
- Campo visivo
 - $160^\circ \times 135^\circ$
 - 330-730 nm

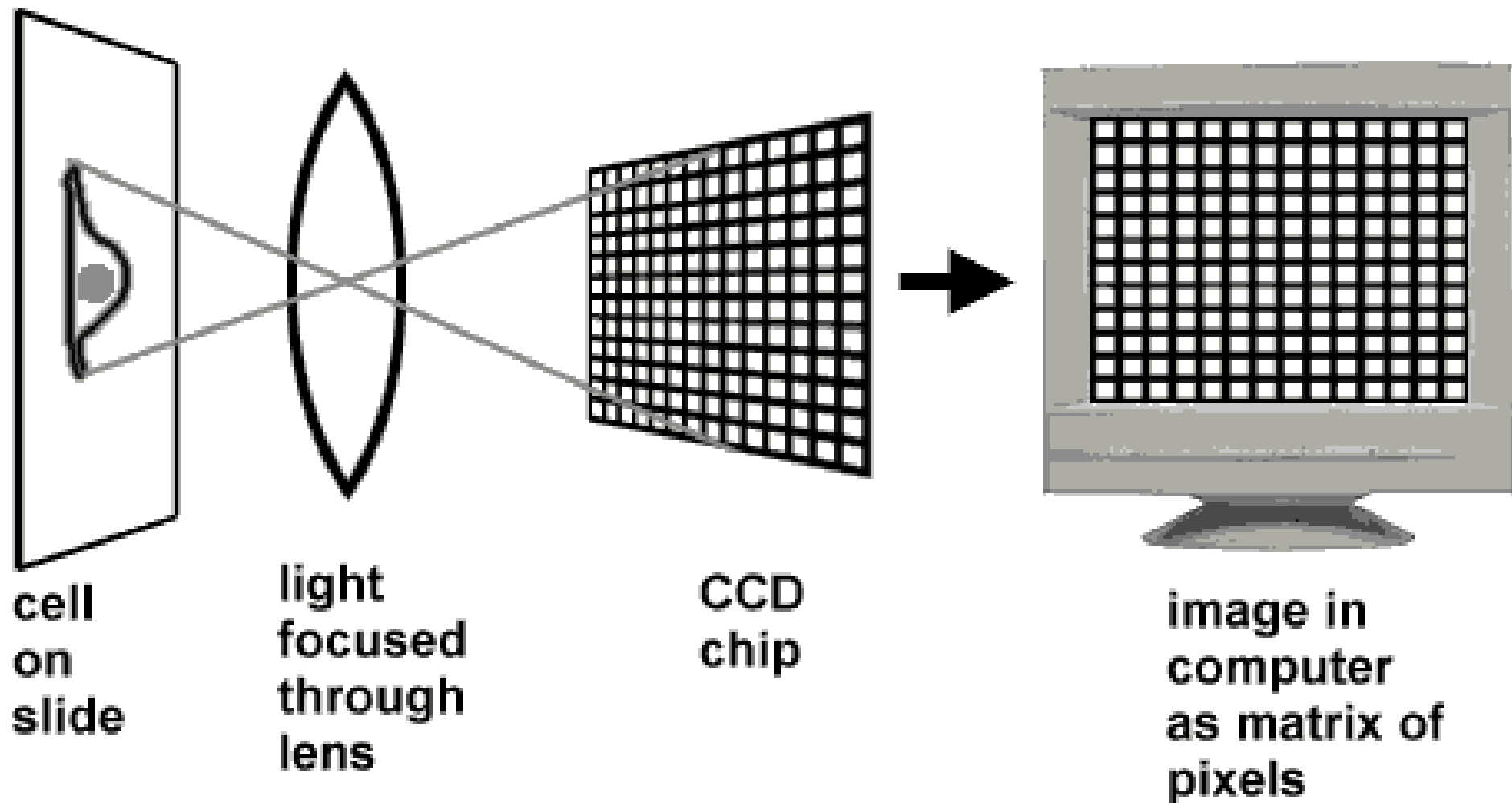
Formazione dell'immagine nella macchina fotografica



Formazione dell'immagine nella telecamera

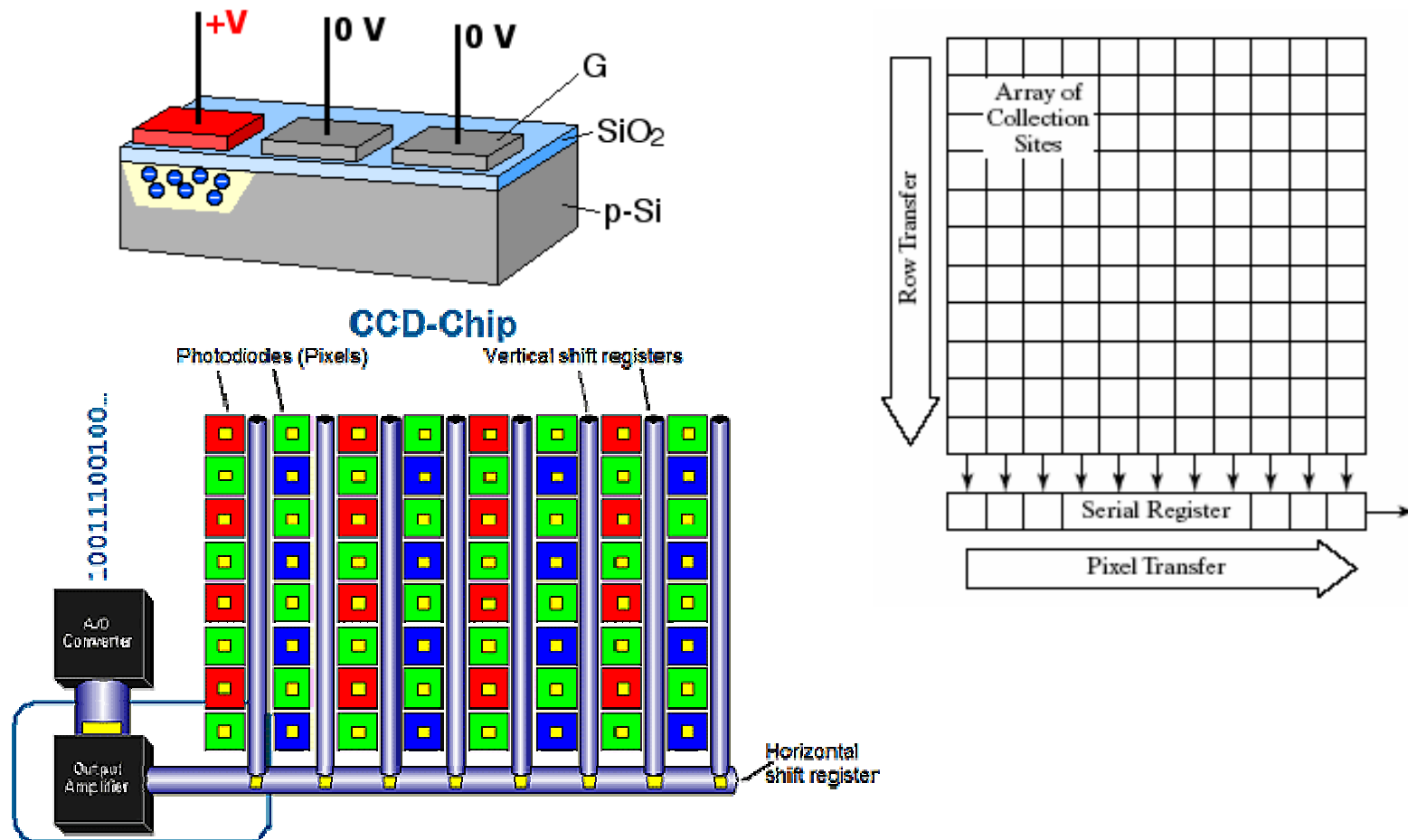


Formazione dell'immagine nella telecamera



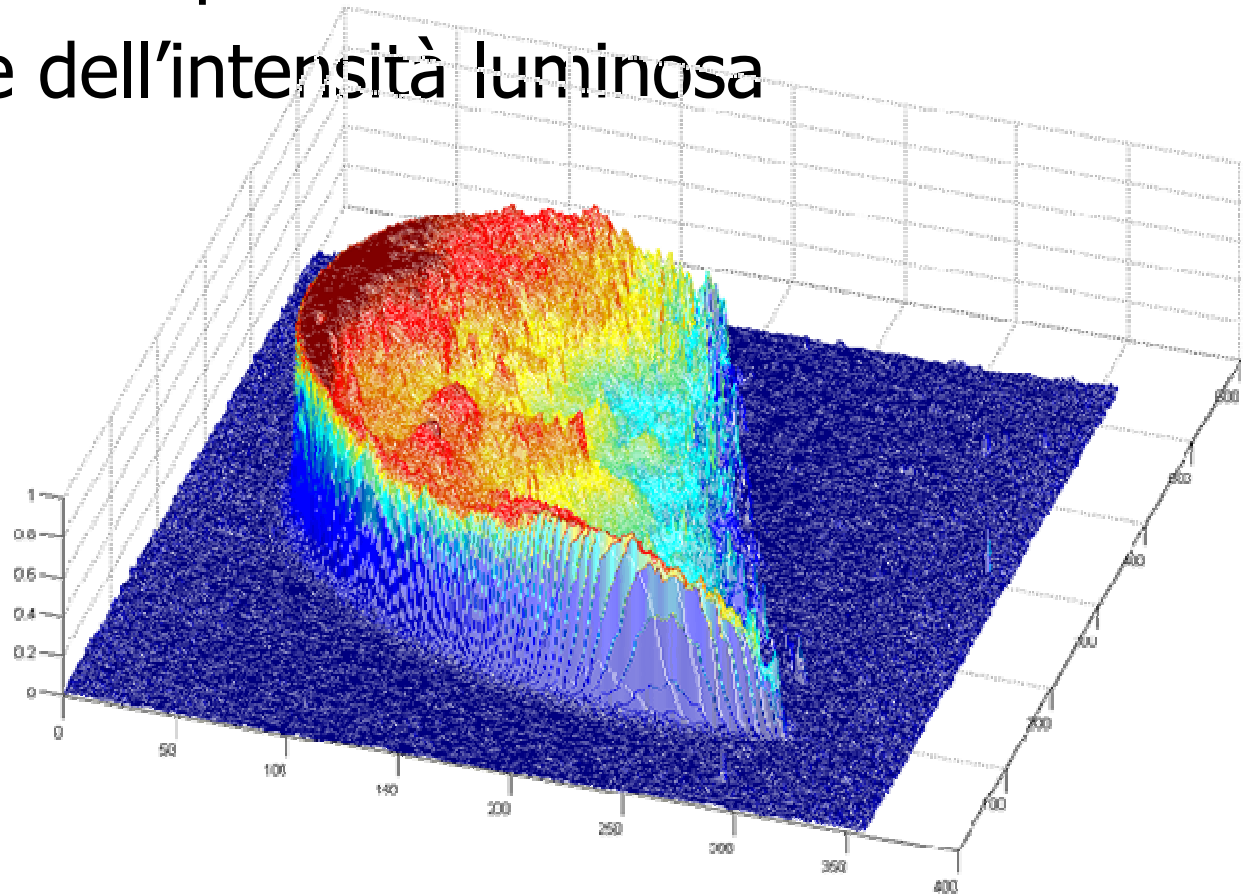
Sensori CCD - Charge-Coupled-Device

- 1969 Bill Boyle e George Smith



Funzione immagine

- $f(x,y) \in \mathbb{R}^2 \times \mathbb{R}$
 - (x,y) : coordinate spaziali
 - $f(x,y)$: valore dell'intensità luminosa



Un esempio di immagine digitale

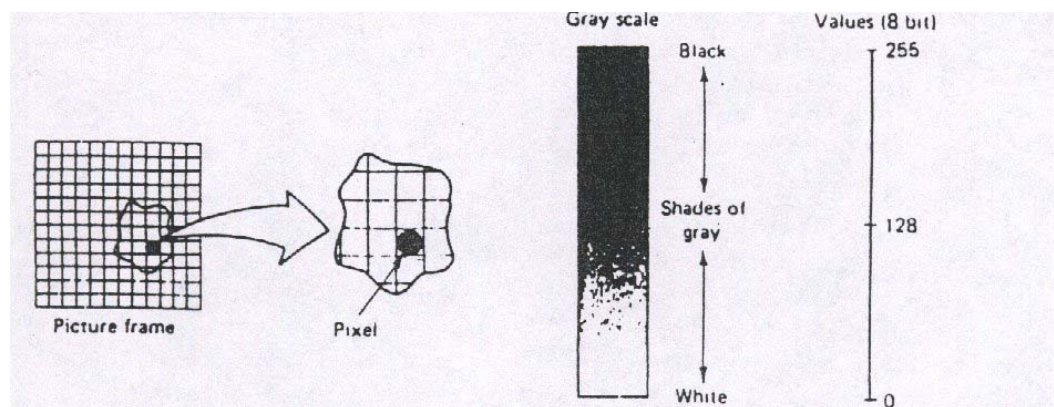
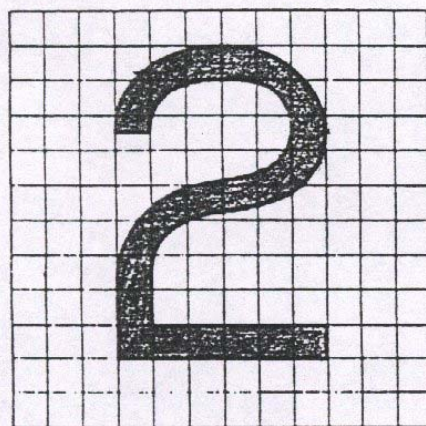


Figure 6-12 A picture frame is divided into picture elements, called pixels, for conversion to a gray-scale value.



(a)

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	50	128	240	255	180	0	0	0	0
0	0	0	200	128	0	0	128	128	0	0	0
0	0	0	128	0	0	0	0	255	0	0	0
0	0	0	0	0	15	50	200	225	0	0	0
0	0	0	25	200	225	175	128	0	0	0	0
0	0	0	200	56	0	0	0	0	0	0	0
0	0	0	255	0	0	0	0	0	0	0	0
0	0	0	255	0	0	0	0	0	0	0	0
0	0	0	255	255	255	255	255	255	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

(b)

Figure 6-13 (a) a 12 x 12 pixel grid and (b) matrix for the number 2 (Example 6-1).

Immagine digitale



- CAMPIONAMENTO DELL'IMMAGINE:
discretizzazione delle coordinate spaziali
- QUANTIZZAZIONE DELL'INTENSITA' (o
DEI LIVELLI DI GRIGIO): discretizzazione
in ampiezza
- PIXEL: elemento dell'immagine digitale

Risoluzione spaziale



Profondità di colore

- Binario – 1 bit
- Livelli di grigio – 8 bit
- True color – 24 bit





Geometria delle immagini

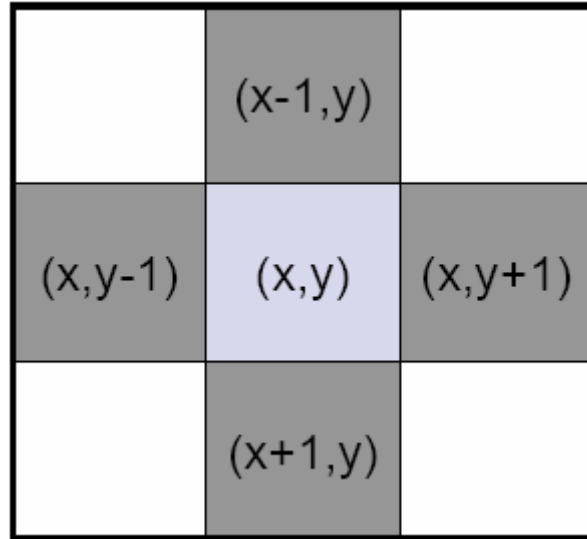
Pixel vicini



- $p = (x, y)$ è **4-VICINO** di:
 $(x+1, y) \ (x-1, y) \ (x, y+1) \ (x, y-1) \in N_4(p)$
- $p = (x, y)$ è **VICINO DIAGONALE** di
 $(x+1, y+1) \ (x+1, y-1) \ (x-1, y+1) \ (x-1, y-1)$
 $\in N_D(p)$
- $p = (x, y)$ è **8-VICINO** di $N_4(p) \cup N_D(p)$

Punti 4-vicini

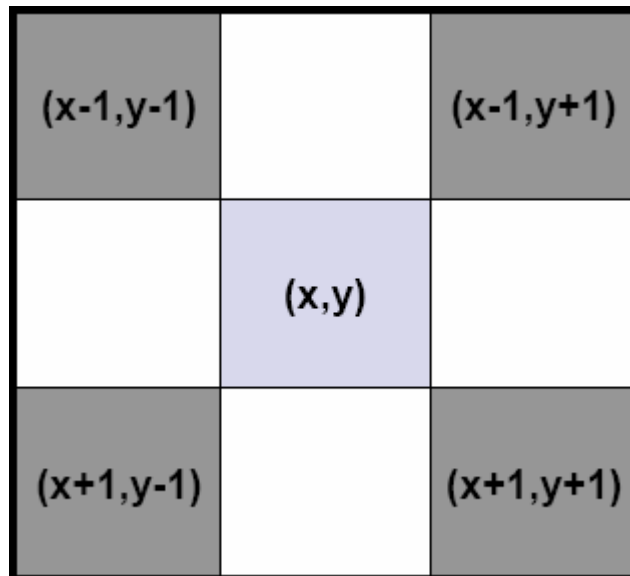
- Un punto $P=(x,y)$ appartenente ad un'immagine digitale ha 4 punti vicini in orizzontale e in verticale



L'insieme dei punti 4-vicini di P si indica con $N_4(P)$

Punti vicini diagonali

- Un punto $P=(x,y)$ appartenente ad un'immagine digitale ha 4 punti vicini diagonali



L'insieme dei punti vicini diagonali di P si indica con $N_D(P)$

Punti 8-vicini

- Un punto $P=(x,y)$ appartenente ad un'immagine digitale ha 8 punti 8-vicini

$(x-1,y-1)$	$(x-1,y)$	$(x-1,y+1)$
$(x,y-1)$	(x,y)	$(x,y+1)$
$(x+1,y-1)$	$(x+1,y)$	$(x+1,y+1)$

L'insieme dei punti 8-vicini di P si indica con $N_8(P)$

$$N_8(P) = N_4(P) \cup N_D(P)$$

Distanza



$p=(x,y)$, $q=(s,t)$ e $z=(u,v)$

D è una funzione della distanza o metrica se

1. $D(p,q) \geq 0$
2. $D(p,q) = D(q,p)$
3. $D(p,z) \leq D(p,q) + D(q,z)$

DISTANZA EUCLIDEA: $D_e(p,q)=((x-s)^2+(y-t)^2)^{1/2}$

Distanza sulle immagini digitali

- La *distanza euclidea* tra due punti $P(x,y)$ e $Q(u,v)$ è definita come:

$$d_e(P,Q) = \sqrt{(x-u)^2 + (y-v)^2}$$

- Sulle immagini digitali sono definite anche altre distanze, "più semplici":
 - distanza *city block* (*a blocchi*)
 - distanza *chessboard* (*a scacchiera*)

Distanza

- DISTANZA A BLOCCHI:

$$D_4 = |x-s| + |y-t|$$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	

- DISTANZA A SCACCHIERA:

$$D_8 = \max(|x-s|, |y-t|)$$

		2		
1	1	2	1	1
1	1	1	1	1
2	1	0	1	2
1	1	1	1	1
1	1	2	1	1

Distanza city block

- E' definita come:

$$d_4(P, Q) = |x - u| + |y - v|$$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

- "Cerchio" identificato dai punti X tali che $d_4(P, X) \leq 2$
- I punti a distanza 1 da P sono proprio i 4-vicini di P.
- $d_4(P, Q)$ è uguale alla lunghezza del più breve 4-percorso da P a Q.

Distanza chessboard

- E' definita come:

$$d_8(P, Q) = \max(|x - u|, |y - v|)$$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

- "Cerchio" identificato dai punti X tali che $d_8(P, X) \leq 2$
- I punti a distanza 1 da P sono proprio i 8-vicini di P.
- $d_8(P, Q)$ è uguale alla lunghezza del più breve 8-percorso da P a Q.

Proprietà metriche



- Si noti come tutte queste distanze siano *metriche*
- Infatti, per tutte valgono le seguenti proprietà:
 - $d(P,Q) \geq 0$; $d(P,Q) = 0 \Leftrightarrow P = Q$
 - $d(P,Q) = d(Q,P)$
 - $d(P,R) \leq d(P,Q) + d(Q,R)$

Connettività

- Siano P e Q due punti dell'immagine digitale allora:

- P e Q sono **4-CONNESSI** se $Q \in N_4(P)$

- P e Q sono **8-CONNESSI** se $Q \in N_8(P)$

- P e Q sono **m-CONNESSI** se:

- $Q \in N_4(P)$ oppure

- $Q \in N_D(P) \wedge N_4(P) \cap N_4(Q) = \emptyset$

0 1 1

0 2 0

0 0 1

0 1—1

0 2 0

0 0 1

0 1—1

0 2 0

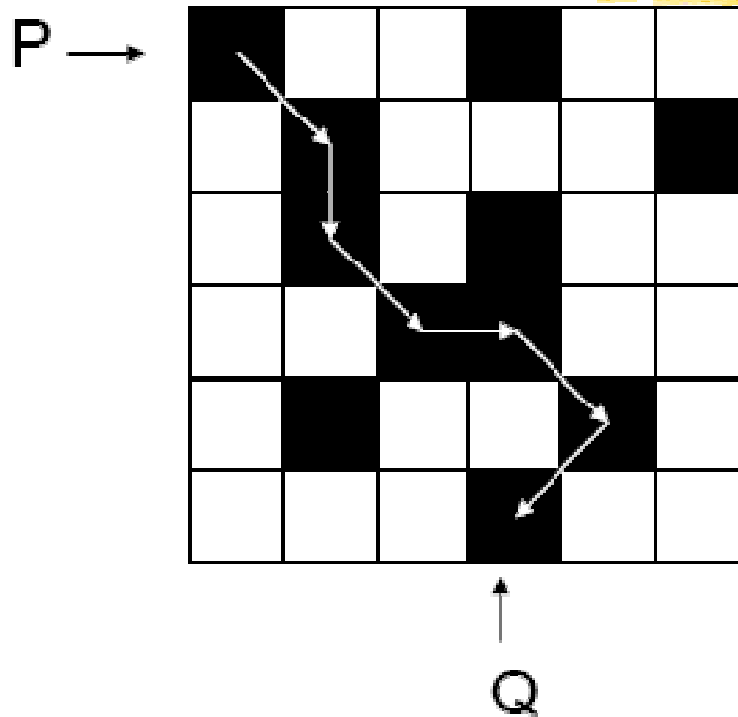
0 0 1

Connettività

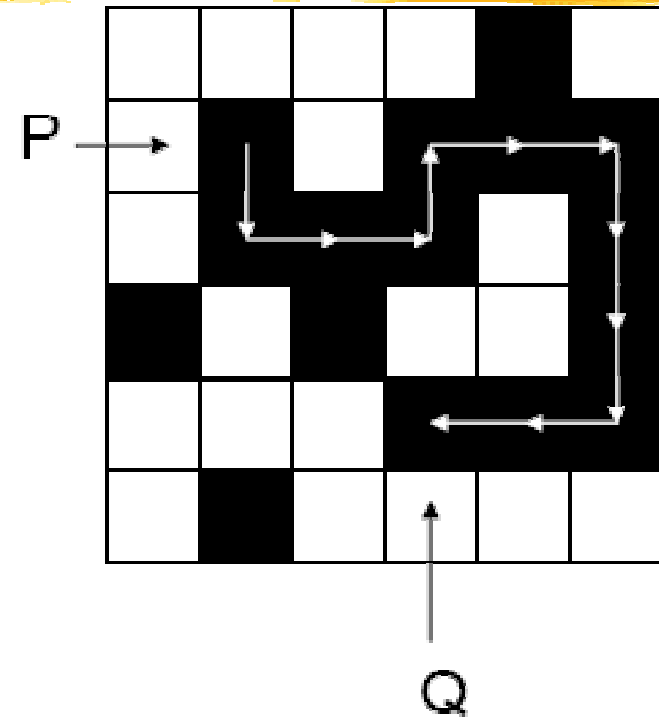


- Un *percorso* (*path*) di lunghezza n da P a Q è una sequenza di punti $P=P_1, P_2, \dots, P_n=Q$ tale che P_i è un k -vicino di $P_{i-1} \forall i=1, \dots, n$
- A seconda del valore di k , si parla di *4-percorso* o di *8-percorso*
- P si dice *k-connesso* a Q se esiste un k -percorso da P a Q formato interamente da punti appartenenti ad S .

Esempio



P è 8-connesso a Q



P è 4-connesso a Q

Se P è 4-connesso a Q allora è anche 8-connesso a Q

Marcatura dei componenti connessi



- In numerose applicazioni si presenta il problema di marcare ogni componente connesso di un'immagine binaria con un'etichetta diversa (*component labeling*).
- La connessione si intende con riferimento alla relazione di adiacenza scelta.
- Ciò permette di individuare gli oggetti distinti presenti in un'immagine, operazione preliminare spesso necessaria nell'analisi e nell'interpretazione di immagini.

Marcatura dei componenti connessi

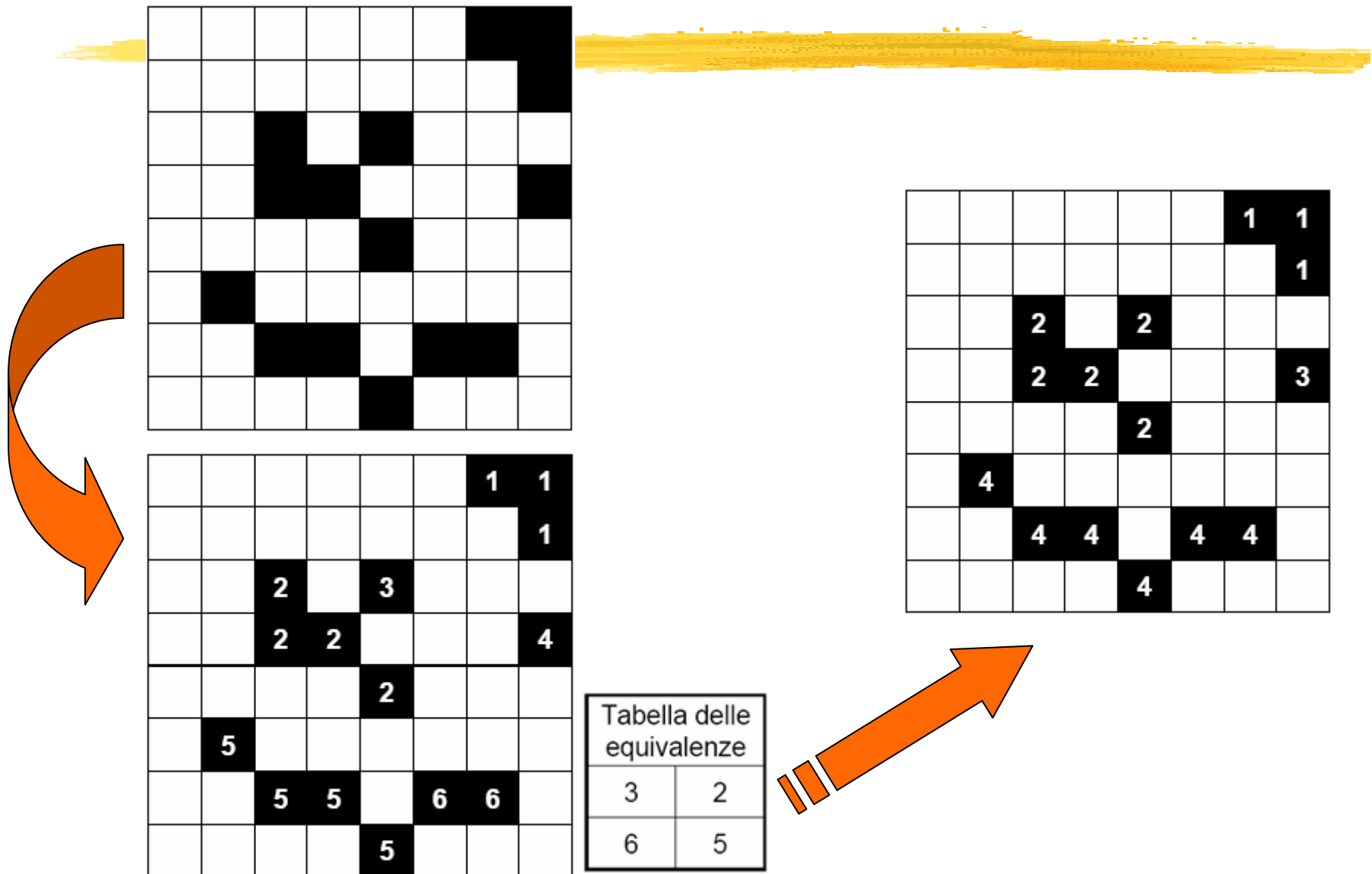


- E' un algoritmo sequenziale che elabora l'immagine punto per punto procedendo dall'alto verso il basso e da sinistra verso destra.
- Per ogni punto in esame si accede a due righe dell'immagine: quella che contiene il punto e quella precedente. E' quindi necessario gestire in modo particolare i punti della prima riga.
- Una gestione simile deve essere assicurata anche per i punti della prima colonna

Algoritmo di marcatura

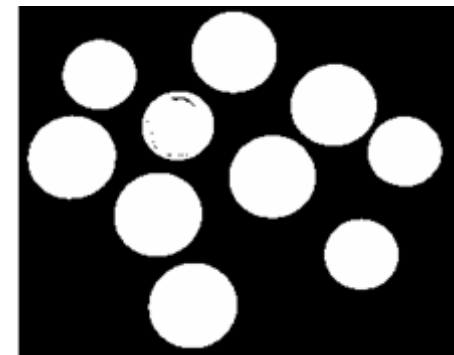
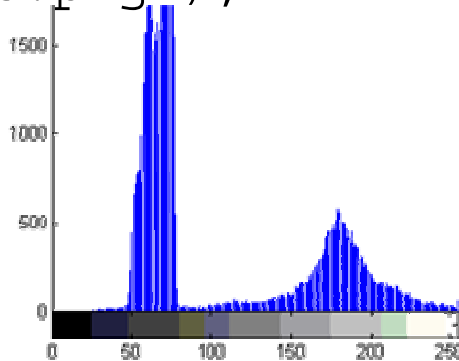
- Per ogni $P \in I$
- Si controlla se uno dei suoi vicini già visitati è già stato marcato con un'etichetta R_i ($i=1,2,\dots$).
 - Se questo non è verificato, viene assegnata a P l'etichetta col valore più basso non ancora assegnato.
 - Se, invece, qualcuno dei vicini è già stato marcato con etichette R_1, R_2, \dots, R_k , P viene marcato con l'etichetta $R_P = \min\{R_1, R_2, \dots, R_k\}$ e in una tabella viene registrato che l'etichette R_1, R_2, \dots, R_k sono equivalenti perché si riferiscono allo stesso oggetto.
- Si prosegue fino alla fine dell'immagine

Esempio: Marcatura 8-connessi



Esempio MATLAB

- `I = imread('coins.png');`
- `imshow(I);`
- `imhist(I);`
- `IBW = im2bw(I, 98/255);`
- `imshow(IBW);`
- `[lab,nObj] = bwlabel(IBW,4);`
 - `numObjects = 10`
- `p_color = label2rgb(labeled, @jet);`
- `imshow(p_color);`



Elaborazione delle immagini digitali

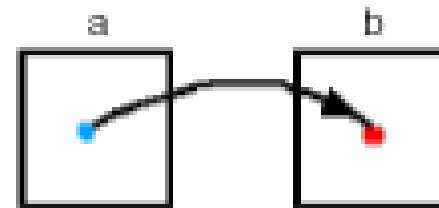


- I tipi di operazioni che si possono realizzare per trasformare un'immagine in ingresso $a[M,N]$ in un'immagine di uscita $b[M,N]$ possono essere classificate in tre categorie:
 - **Operatori puntuali**
 - **Operatori locali**
 - **Operatori globali**

Operazioni su immagini digitali

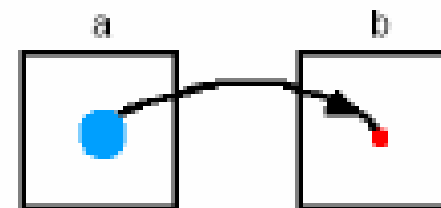
- **Operatori puntuali**

il valore di uscita nel punto (i,j) dipende solo dal valore di ingresso nel punto (i,j)



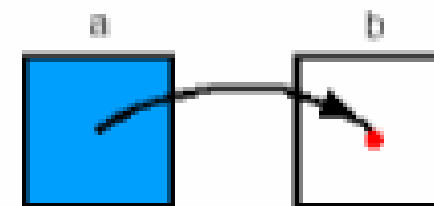
- **Operatori locali**

il valore di uscita nel punto (i,j) dipende solo dai valori di ingresso in un intorno del punto (i,j)



- **Operatori globali**

il valore di uscita nel punto (i,j) dipende da tutti i valori dell'immagine di ingresso





Operazioni puntuali

Trasformazioni puntuali



- Sono trasformazioni tipicamente orientate al miglioramento della qualità dell'immagine (image enhancement).
- Generalmente si realizzano tramite una funzione $y=y(x)$, che ad un livello di grigio x dell'immagine in ingresso, fa corrispondere il valore y per l'immagine in uscita.
- La trasformazione si può realizzare tramite delle Look-up Table (LUT) che permettono un'implementazione hardware efficiente della trasformazione.

Trasformazioni di base



- Binarizzazione
- Inversione dei livelli di grigio
- Compressione potenza
- Espansione di contrasto
- Equalizzazione istogramma

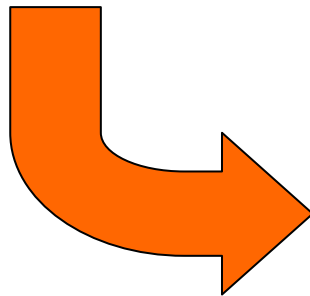
Binarizzazione - sogliatura

Trasformazione dell'immagine in IMMAGINE BINARIA, attraverso il confronto con una soglia T:

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

- T può essere costante o variabile
- T può essere determinata empiricamente o con tecniche statistiche

Esempio



Inversione dei livelli di grigio

- Semplice trasformazione del tipo:

$$g(x,y)=255-f(x,y)$$

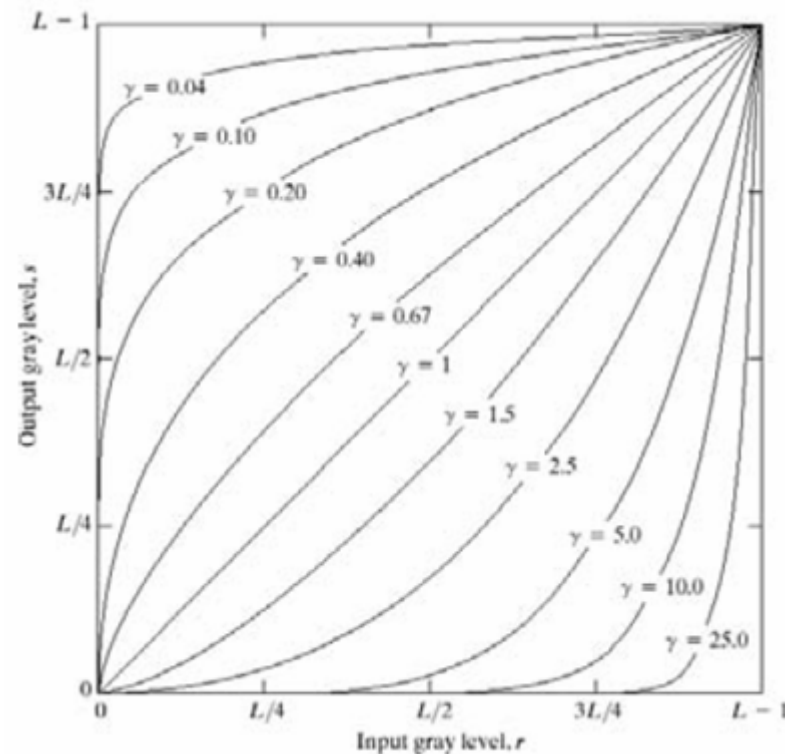
Fornisce la "negativa" dell'immagine originale



Compressione di potenza

- Trasformazione del tipo:

$$g(x,y)=c f(x,y)^\gamma$$



Esempio

originale



$\gamma = 3.0$



$\gamma = 4.0$



$\gamma = 5.0$



Espansione di contrasto

- Si realizza per aumentare la dinamica di un'immagine il cui istogramma è concentrato in un intervallo limitato dei valori possibili
- Esempio – clipping a 150

$$g(x, y) = \begin{cases} 0 & \text{per } f(x, y) < 150 \\ \frac{255 * [f(x, y) - 150]}{105} & \text{per } f(x, y) \geq 150 \end{cases}$$

Pre-elaborazione (early processing): tecniche di FILTRAGGIO



- Media degli intorni
- Filtraggio mediano
- Media dell'immagine
- Sottrazione dello sfondo
- Trasformazione dell'istogramma

Metodi del dominio spaziale

$$g(x,y) = h[f(x,y)]$$

Maschere di Convoluzione

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Correlazione di immagini

- Misure di similitudine, per confrontare una funzione immagine $f(x)$ con un template $t(x)$

$$R_{ft}(y) = \sum_x f(x)t(x-y)$$

Template

1	1	1
1	1	1
1	1	1

Immagine

1	1	0	0	0
1	1	1	0	0
1	0	1	0	0
0	0	0	0	0
0	0	0	0	8

Correlazione

7	4	2	x	x
5	3	2	x	x
2	1	9	x	x
x	x	x	x	x
x	x	x	x	x

Operatori locali

- Nei filtri lineari l'uscita è una combinazione lineare dei valori dei pixel di ingresso.
- I coefficienti della combinazione sono disposti su una sottoimmagine delle stesse dimensioni del vicinato del punto, in modo da corrispondere ai punti che vanno a pesare.
- La sottoimmagine viene definita maschera o filtro (filter, mask, kernel). Perciò si parla di *filtraggio spaziale*.

$w_{-1,-1}$	$w_{-1,0}$	$w_{-1,+1}$
$w_{0,-1}$	$w_{0,0}$	$w_{0,+1}$
$w_{+1,-1}$	$w_{+1,0}$	$w_{+1,+1}$

$$b(i,j) = w_{-1,-1} \cdot a(i-1,j-1) + w_{-1,0} \cdot a(i-1,j) + w_{-1,+1} \cdot a(i-1,j+1) + \\ w_{0,-1} \cdot a(i,j-1) + w_{0,0} \cdot a(i,j) + w_{0,+1} \cdot a(i,j+1) + \\ w_{+1,-1} \cdot a(i+1,j-1) + w_{+1,0} \cdot a(i+1,j) + w_{+1,+1} \cdot a(i+1,j+1)$$

Operatori locali

- Anche per i filtri non lineari, il valore risultante in un punto (i,j) dipende dal valore dei pixel nel vicinato di (i,j) definito tramite una maschera.
- La valutazione anche in questo caso richiede che la maschera "scorra" su tutta l'immagine (convoluzione)

Esempio di filtro non lineare:

$$\mathbf{b(i,j) = \max(a(i+h,j+k), h \in [-1,1], k \in [-1,1])}$$

Filtri di smoothing



- Sono filtri per il miglioramento della qualità dell'immagine
- Hanno l'effetto di diminuire il contrasto locale dell'immagine
- Sono usati per:
 - eliminare i dettagli inutili (blurring)
 - ridurre alcuni tipi di rumore (noise cleaning)

Caratteristiche dei filtri di smoothing

- Se l'immagine risultante è destinata alla visualizzazione, i valori dei suoi pixel devono restare entro la gamma dei livelli di grigio rappresentabili (es. 0-255)
- A questo scopo, i coefficienti del filtro devono soddisfare alcune condizioni:
 - $w_i \geq 0 \quad \forall i = 1 \dots n$
 - $\sum_i w_i = 1$
- In queste condizioni, una zona a valore di grigio costante entro la maschera del filtro resta immutata dopo il filtraggio e l'effetto del filtro resta limitato ai dettagli dell'immagine (zone ad alta freq. spaziale)

Filtri di smoothing

- Tipicamente, calcolano la media dei valori dei pixel in un intorno simmetrico (3x3, 5x5, 7x7,...)

1/9

1	1	1
1	1	1
1	1	1

1/25

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Media degli intorni



$$g(x,y) = 1/P \sum_{n,m \in S} f(n,m)$$

S: intorno di (x,y)

P: numero di punti di S

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Esempio: Blurring



Esempio: Blurring



Filtri di smoothing

- Sono utilizzate anche altre maschere che realizzano una media pesata (es. filtro gaussiano)

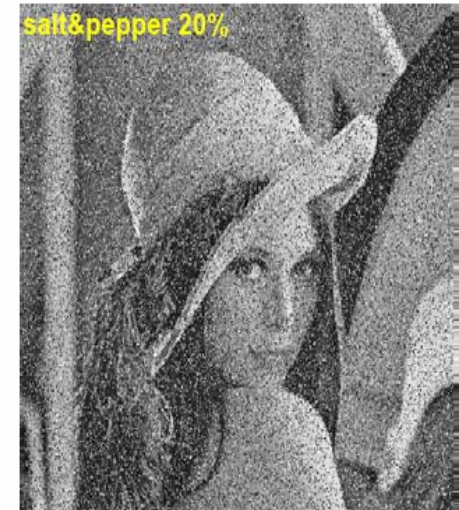
0.0113	0.0838	0.0113
0.0838	0.6193	0.0838
0.0113	0.0838	0.0113

Discretizzazione su una maschera 3x3 di una gaussiana con media nulla e varianza pari a 0.5

Noise cleaning

- **rumore impulsivo**, detto anche "sale e pepe" (*salt & pepper*)

Viene caratterizzato dalla frazione (in %) dell'immagine modificata



- **rumore gaussiano bianco**

Viene caratterizzato dalla media e dalla varianza

media=0 varianza=0.01



media=0 varianza=0.1



Rumore impulsivo e filtri di media

Filtro di media 3x3

salt&pepper 10%



salt&pepper 20%



Filtro di media 5x5

salt&pepper 10%



salt&pepper 20%



Filtraggio mediano

$$g(x,y) = \text{mediana di } (x,y)$$

Mediana M di un insieme di valori: valore tale che metà dei valori dell'insieme sono minori di M e metà sono maggiori

7	10	12
6	38	11
9	11	6

6 6 7 9 10 11 11 12 38



valore
mediano

Applicazione filtro mediano

Filtro mediano 3x3



Filtro mediano 5x5





Effetti di una media su
un intorno 5x5

Effetti di un filtraggio mediano
su un intorno 5x5

Esempi di applicazione di un filtro mediano

Immagine con rumore



Filtro mediano 3x3



Immagine con rumore



Filtro mediano 5x5



Media o mediana?



- Il filtro di media tende a creare nuovi livelli di grigio prima non esistenti
- Inoltre, attenua non solo il rumore, ma anche tutte le alte frequenze spaziali in maniera indiscriminata, causando così sfocatura, perdita di dettaglio fine e smussatura dei fronti di salita nelle transizioni chiaro/scuro
- Il filtro mediano non deteriora i fronti di salita, ma elimina i picchi con base sufficientemente piccola rispetto all'ampiezza della maschera

Sottrazione dello sfondo



La tecnica della sottrazione dello sfondo tenta di rimuovere le leggere variazioni dei livelli di grigio dello sfondo, approssimandole con una funzione e sottraendo tale funzione dalla funzione immagine

$$f_n(x,y) = f(x,y) - f_b(x,y)$$

$$f_b(x,y) = c \text{ (costante)} \quad \text{oppure}$$

$$f_b(x,y) = m_1x + m_2y + c \text{ (lineare)}$$

Istogramma di un'immagine



- I pixel di una immagine sono una "popolazione" sulla quale possiamo calcolare tutte le quantità statistiche descrittive che si usano normalmente:

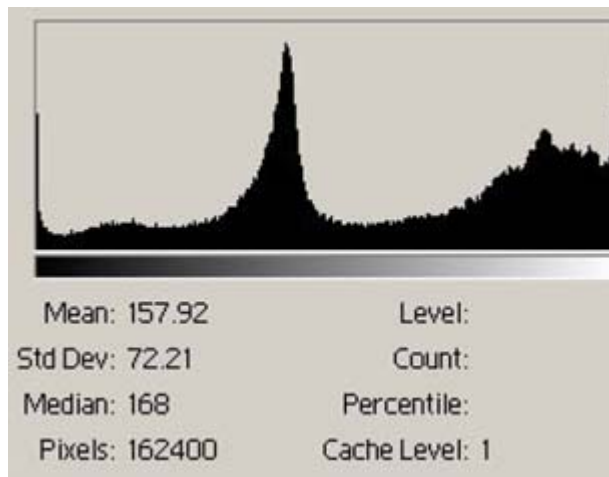
Media, mediana, varianza, deviazione standard, quartili, percentili, ecc.

- Particolarmente importante è la conoscenza della distribuzione delle frequenze dei toni di grigio:

l'istogramma

Istogramma

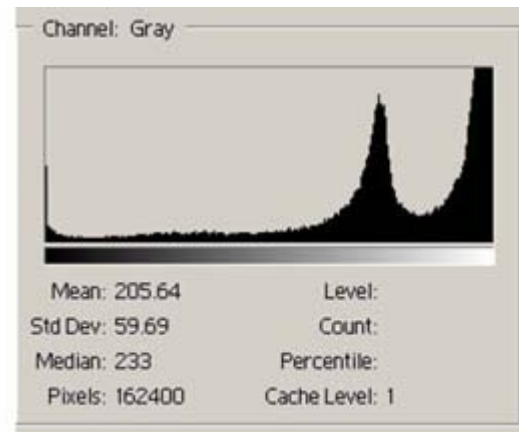
- Per ogni livello di grigio, riporta il numero di pixel di quel valore
- Per una immagine $I[m,n]$ si ha:
$$H(k) = \text{numero di pixel di valore } k$$
- La somma di tutti gli H è esattamente $m \times n$



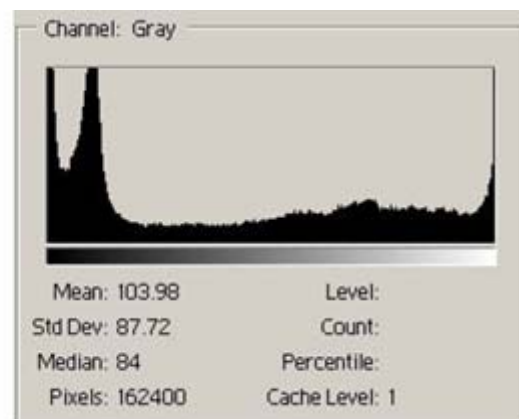
- L'istogramma è utile a comprendere in maniera immediata le caratteristiche dell'immagine

Caratteristiche dell'istogramma

- Immagine "chiara" più denso a destra

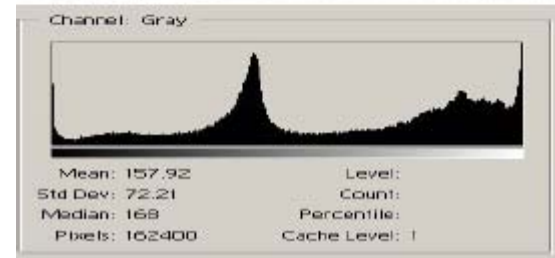
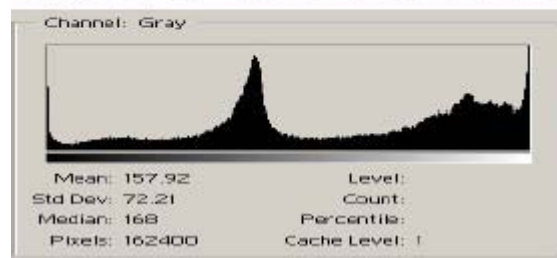


- Immagine "scura" più denso a sinistra



Limite dell'istogramma

- Non tiene conto della distribuzione spaziale



Equalizzazione istogramma



- E' una tecnica che mira a modificare la forma dell'istogramma redistribuendo i valori dei livelli di grigio in modo che l'istogramma sia quanto più uniforme possibile
- L'obiettivo è quello di migliorare immagini a debole contrasto
- L'equalizzazione non porta necessariamente ad un miglioramento dell'immagine (Es. immagine con istogramma bimodale)

Trasformazione (equalizzazione) dell'istogramma

ISTOGRAMMA DI UN'IMMAGINE: funzione che dà la frequenza di apparizione di ogni livello di grigio nell'immagine

$h(p)$ = numero di pixel con valore p ($0 \leq p \leq n$)

EQUALIZZAZIONE DELL'ISTOGRAMMA: mappatura dei livelli di grigio p in livelli di grigio q t.c. la loro distribuzione è uniforme

$$g(q) dq = h(p) dp$$

$$g(q) = N^2/M$$

N^2 : numero di pixel, M : numero dei livelli di grigio

$$g(p) = M/N^2 \int_0^p h(s) ds$$

Equalizzazione istogramma

- Supponiamo inizialmente di lavorare nel continuo e sia $h(x)$ l'istogramma dell'immagine di partenza
- Per realizzare l'equalizzazione è necessaria una trasformazione monotona

$$y=y(x)$$

- tale che l'istogramma $g(y)$ dell'immagine trasformata sia costante

$$g(y)=C$$

- Si impone che aree elementari dell'istogramma originale si trasformano in aree corrispondenti dell'istogramma modificato, si ha: $h(x)dx = g(y)dy = Cdy$

Equalizzazione istogramma

- Ricavo $y(x)$:

$$\frac{dy}{dx} = \frac{1}{C} h(x) \quad \Rightarrow \quad y(x) = \frac{1}{C} \int_0^x h(x) dx$$

- Nel dominio discreto diventa:

$$y(x) = \frac{1}{C} \sum_{k=0}^x H(k) = \frac{l}{n \cdot m} \sum_{k=0}^x H(k)$$

- Per un'immagine a livelli di grigio (8-bit):

$$y(x) = \frac{1}{C} \sum_{k=0}^x H(k) = \frac{256}{\sum_{k=0}^{255} H(k)} \sum_{k=0}^x H(k)$$

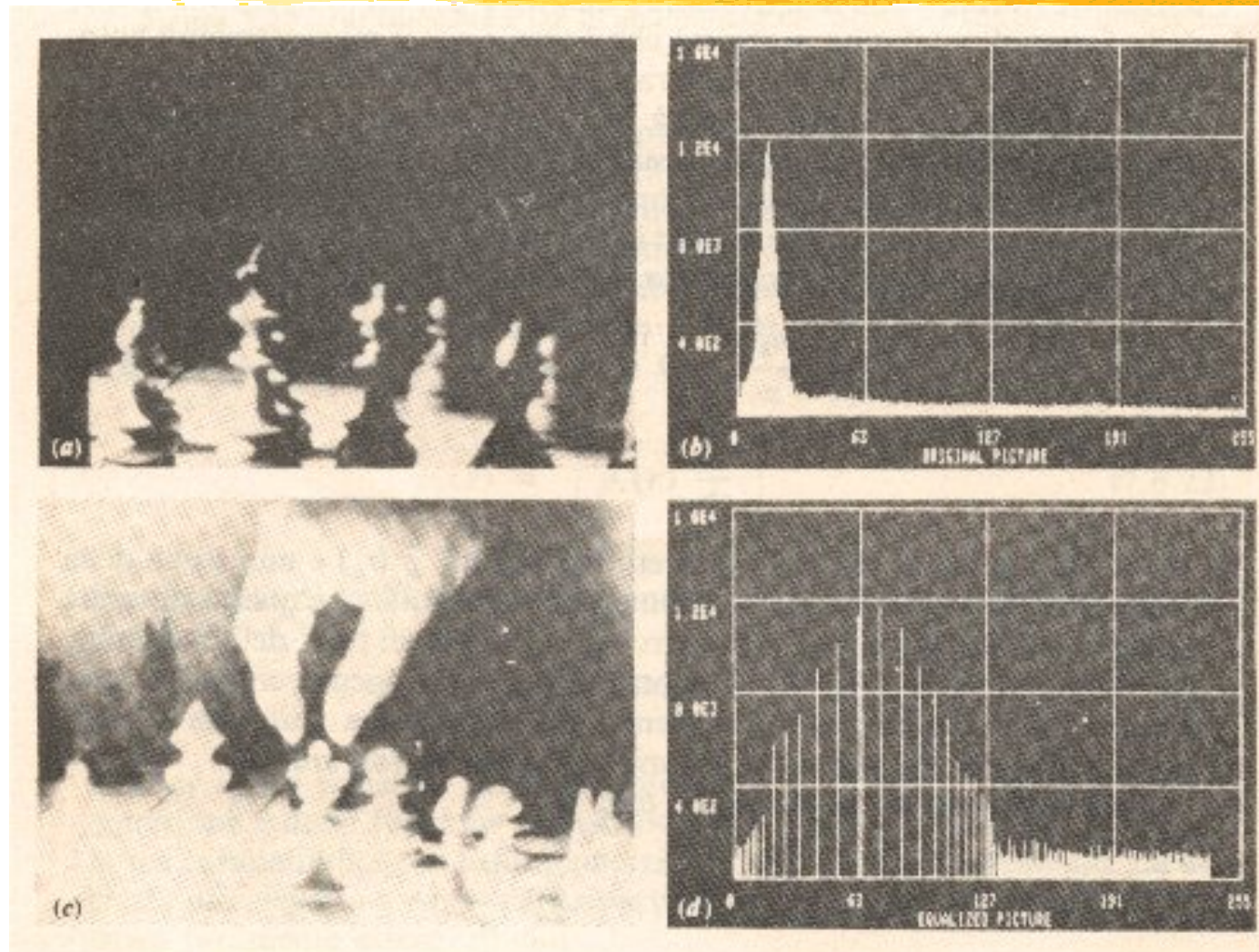
Algoritmo equalizzazione

1. Valutare istogramma $H(k)$

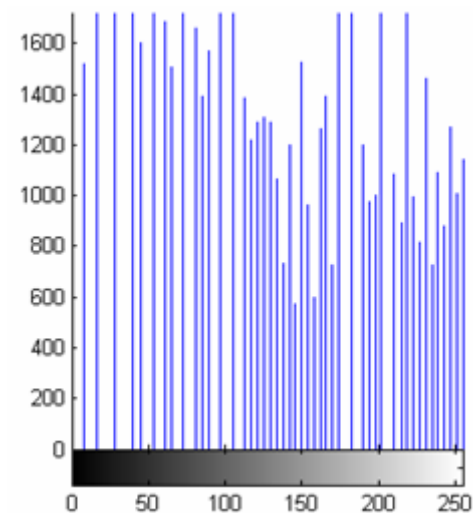
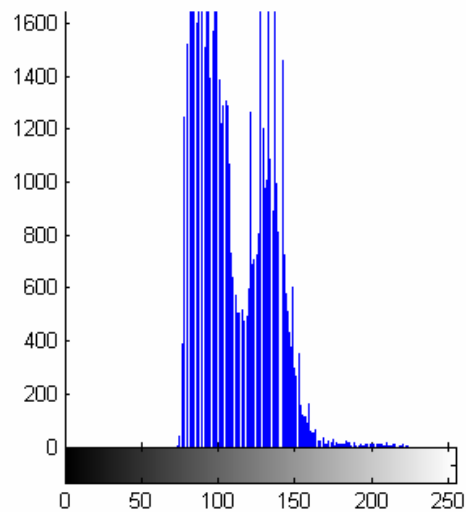
2. Calcolare la $y(x) = \frac{l}{n \cdot m} \sum_{k=0}^x H(k)$

3. Eseguire la trasformazione tramite la $y(x)$

Esempio di equalizzazione dell'istogramma



Esempio di equalizzazione dell'istogramma



Pre-elaborazione (early processing): tecniche di RILEVAMENTO DEI BORDI



BORDO (EDGE): area in cui i livelli di grigio variano rapidamente

OPERATORE DI BORDO (EDGE OPERATOR): operatore matematico in grado di rilevare la presenza di un bordo

- Operatori basati sul gradiente: operatore di differenza, operatore incrociato di Roberts, operatori di Prewitt e Sobel, metodo dei crack edge, metodo del laplaciano
- Tecniche di confronto con un modello (template matching): operatori di Kirsch
- Tecniche basate su modelli parametrici: operatore di Hueckel

Rilevamento di bordi basato sul gradiente - definizioni

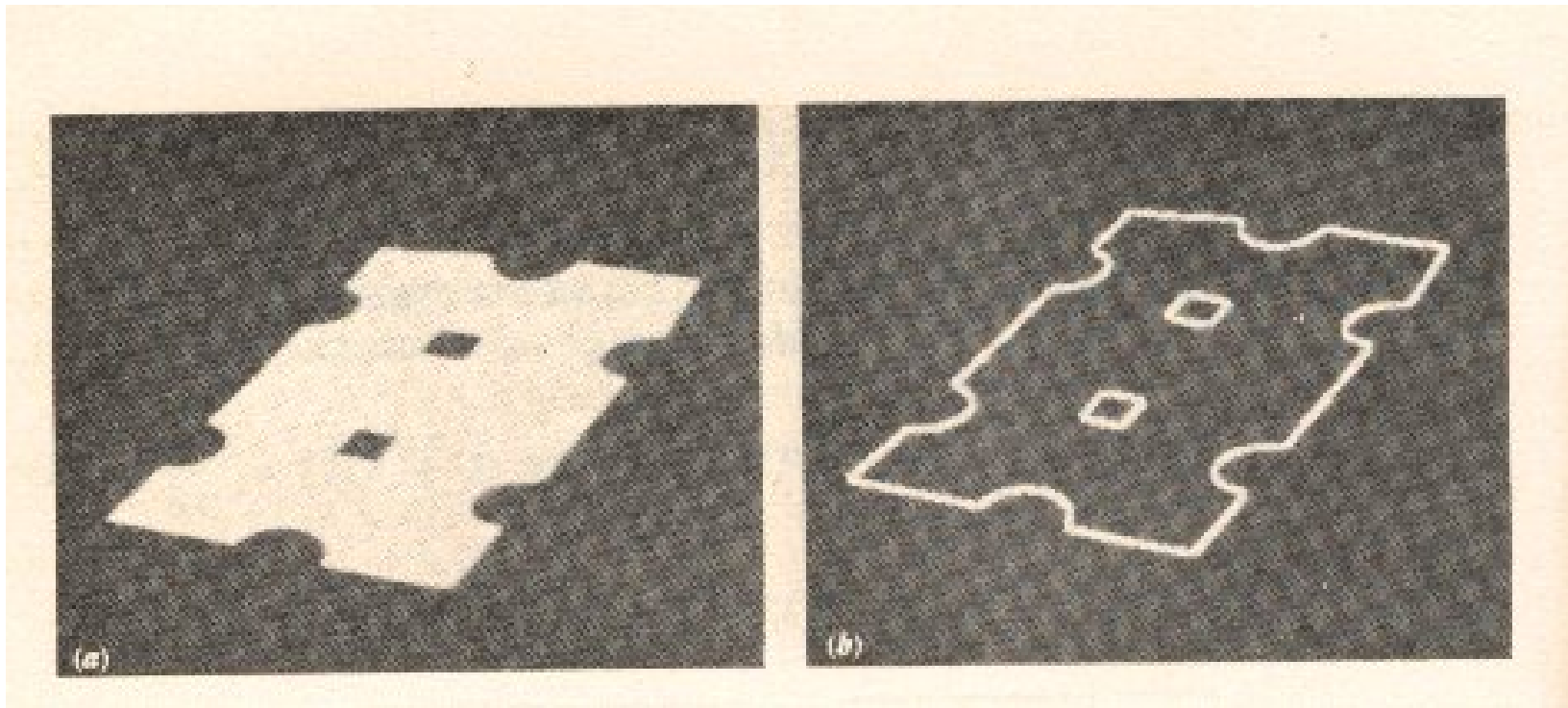


GRADIENTE: misura della discontinuità in un punto dell'immagine

DIREZIONE DEL GRADIENTE ($\phi(x,y)$):
direzione della massima variazione dei livelli di grigio

INTENSITA' DEL GRADIENTE ($s(x,y)$):
intensità della variazione dei livelli di grigio

Rilevamento di bordi basato sul gradiente



Filtri derivativi del primo ordine

- Per implementare le derivate del primo ordine, un operatore utilizzato di frequente è il gradiente che rappresenta la derivata di una $f(x,y)$ nella direzione di massima variazione
- Il gradiente è definito come un vettore colonna:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

DIREZIONE DEL GRADIENTE $\phi(x,y)$:
direzione della massima variazione dei
livelli di grigio

INTENSITA' DEL GRADIENTE $s(x,y)$:
intensità della variazione dei livelli di grigio

Implementazione filtri derivativi

- Di fatto, non si usa il gradiente così come è definito, in quanto le derivate parziali non sono isotrope.
- Si considera invece il modulo del gradiente, anche se l'operatore risultante non è lineare

$$\sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

- La valutazione del modulo del gradiente comporta un'elevata complessità computazionale.
- Per ridurre la complessità, si può approssimare il modulo con la somma dei valori assoluti delle componenti, anche se, a rigore, si perde l'isotropia dell'operatore

$$\left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|$$

Rilevamento di bordi basato sul gradiente



$$G[f(x,y)] = \left| \begin{array}{c} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{array} \right| = \left| \begin{array}{c} \Delta_1 \\ \Delta_2 \end{array} \right|$$

$$S(x,y) = (\Delta_1^2 + \Delta_2^2)^{1/2}$$

$$\phi(x,y) = \tan^{-1}(\Delta_2 / \Delta_1)$$

Rilevamento di bordi basato sul gradiente e sogliatura

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

Rilevamento dei bordi e sogliatura

OPERATORE DI DIFFERENZA:

$$\Delta_1 = f(x+a, y) - f(x, y)$$

$$\Delta_2 = f(x, y+a) - f(x, y)$$

$$S_{i,j} = \sqrt{\Delta_1^2 + \Delta_2^2}$$

$$\Delta_1 = I_{i-a,j} - I_{i,j}$$

$$\Delta_2 = I_{i,j+a} - I_{i,j}$$

$$E_{i,j} = \begin{cases} 0 & \text{se } S_{i,j} \leq \text{soglia} \\ 1 & \text{se } S_{i,j} > \text{soglia} \end{cases}$$

$I_{i-a,j}$		
\vdots		
$I_{i,j}$	\dots	$I_{i,j+a}$

$$S_{i,j} = \Delta_1^2 + \Delta_2^2$$

$$E_{i,j} = \begin{cases} 0 & \text{se } S_{i,j} \leq \text{soglia}^2 \\ 1 & \text{se } S_{i,j} > \text{soglia}^2 \end{cases}$$

Approssimazione derivata prima

$$\frac{\partial f}{\partial x} \cong f(x+1, y) - f(x, y)$$

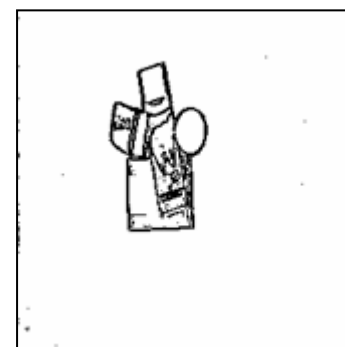
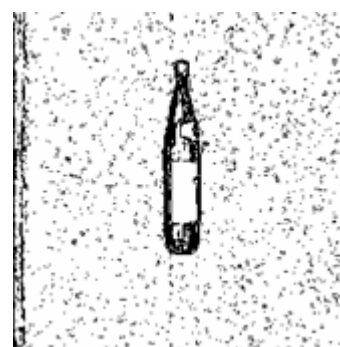
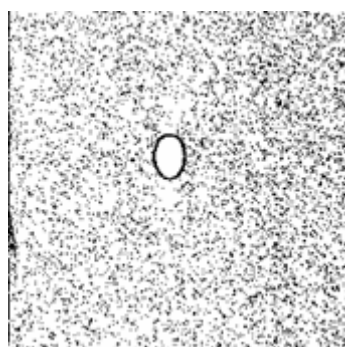
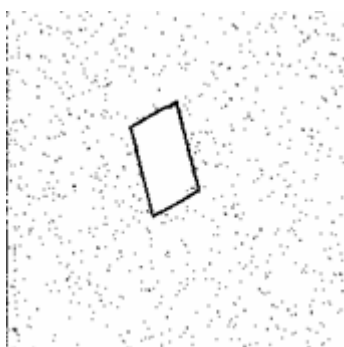
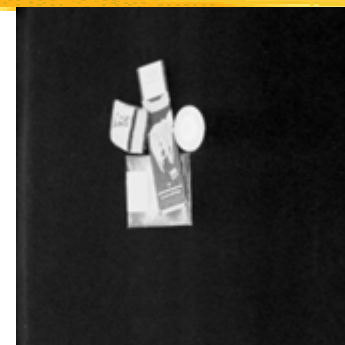
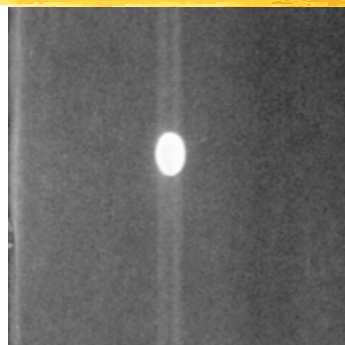
0	0	0
0	-1	0
0	+1	0

$$\frac{\partial f}{\partial y} \cong f(x, y+1) - f(x, y)$$

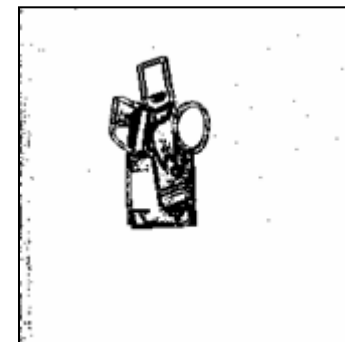
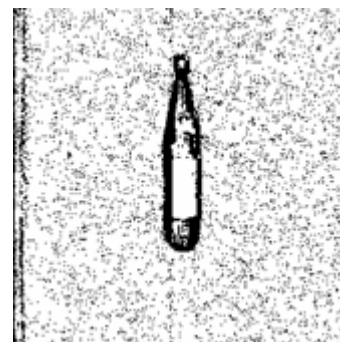
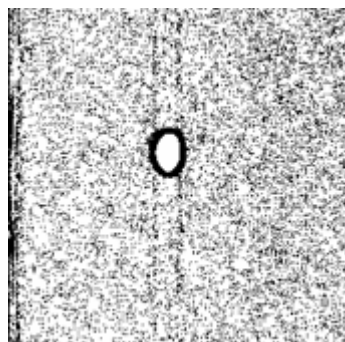
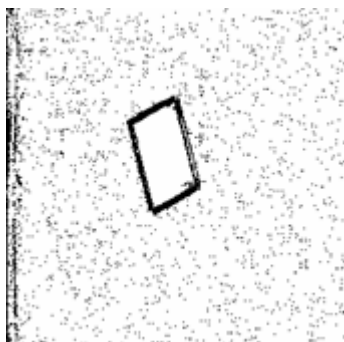
0	0	0
0	-1	+1
0	0	0

Rilevamento dei bordi e sogliatura

$A = 1$



$A = 2$



Rilevamento di bordi basato sul gradiente

OPERATORE INCROCIATO DI ROBERTS:

$$\Delta_1 = f(x, y+a) - f(x+a, y) \quad \Delta_2 = f(x, y) - f(x+a, y+a)$$

 Δ_1

0	1
-1	0

 Δ_2

1	
	-1

OPERATORE DI PREWITT:

$$\Delta_1 = f(x-1, y+1) + f(x, y+1) + f(x+1, y+1) \\ - f(x-1, y-1) - f(x, y-1) - f(x+1, y-1)$$

$$\Delta_2 = f(x-1, y-1) + f(x-1, y) + f(x-1, y+1) \\ - f(x+1, y-1) - f(x+1, y) - f(x+1, y+1)$$

 Δ_1

-1	-1	-1
0	0	0
1	1	1

 Δ_2

1	0	-1
1	0	-1
1	0	-1

Operatori di Prewit

OPERATORI DI PREWITT:

$$\nabla_1 = f(x-1, y+1) + f(x, y+1) + f(x+1, y+1) \\ - f(x-1, y-1) - f(x, y-1) - f(x+1, y-1)$$

$$\nabla_2 = f(x-1, y-1) + f(x-1, y) + f(x-1, y+1) \\ - f(x+1, y-1) - f(x+1, y) - f(x+1, y+1)$$

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

Rilevamento di bordi basato sul gradiente

OPERATORE DI SOBEL:

$$\Delta_1 = f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1) \\ - f(x-1, y-1) - 2f(x, y-1) - f(x+1, y-1)$$

$$\Delta_2 = f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1) \\ - f(x+1, y-1) - 2f(x+1, y) - f(x+1, y+1)$$

Δ_1

-1	-2	-1
0	0	0
1	2	1

Δ_2

-1	0	1
-2	0	2
-1	0	1

Approssimazione derivata seconda

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &\cong f(x+1, y) - f(x, y) - (f(x, y) - f(x-1, y)) = \\ &= f(x+1, y) + f(x-1, y) - 2f(x, y)\end{aligned}$$

0	+1	0
0	-2	0
0	+1	0

$$\begin{aligned}\frac{\partial^2 f}{\partial y^2} &\cong f(x, y+1) - f(x, y) - (f(x, y) - f(x, y-1)) = \\ &= f(x, y+1) + f(x, y-1) - 2f(x, y)\end{aligned}$$

0	0	0
+1	-2	+1
0	0	0

Isotropia dell'operatore: Il laplaciano

- Nel definire un operatore differenziale del secondo ordine, una caratteristica da garantire è che la risposta sia indipendente dalla direzione della discontinuità nell'immagine (operatore *isotropo*).
- L'operatore derivativo isotropo più semplice è il laplaciano:

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

- L'implementazione del laplaciano per immagini digitali si realizza utilizzando le implementazioni delle derivate seconde

Implementazione del Laplaciano

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

$$\frac{\partial^2 f}{\partial x^2}$$

0	+1	0
0	-2	0
0	+1	0

$$\frac{\partial^2 f}{\partial y^2}$$

0	0	0
+1	-2	+1
0	0	0


$$\nabla^2 f$$

0	+1	0
+1	-4	+1
0	+1	0

Filtro di sharpening mediante Laplaciano

- Per ottenere l'immagine migliorata, è necessario combinare l'immagine originale con il laplaciano



$$f(x, y)$$



$$\nabla^2 f(x, y)$$



$$f - \nabla^2 f$$

Maschere filtri di sharpening

- La maschera risultante è:

$$f - \nabla^2 f \quad \longrightarrow$$

0	-1	0
-1	5	-1
0	-1	0

- Spesso si usano anche altre due maschere di sharpening:

-1	-1	-1
-1	9	-1
-1	-1	-1

1	-2	1
-2	5	-2
1	-2	1

Metodo del Laplaciano

Approssimazione della derivata del secondo ordine

$$L(x,y) = [f(x,y+1) + f(x,y-1) + f(x-1,y) + f(x+1,y)] - 4f(x,y)$$

0	1	0
1	-4	1
0	1	0

Operatore di Kirsch

$$S(x) = \max[1, \max_k \sum_{k=1}^{K+1} |f(x_k) - f(x)|]$$

$$X_k \subseteq N_8(x)$$

-1	0	1
-1	0	1
-1	0	1

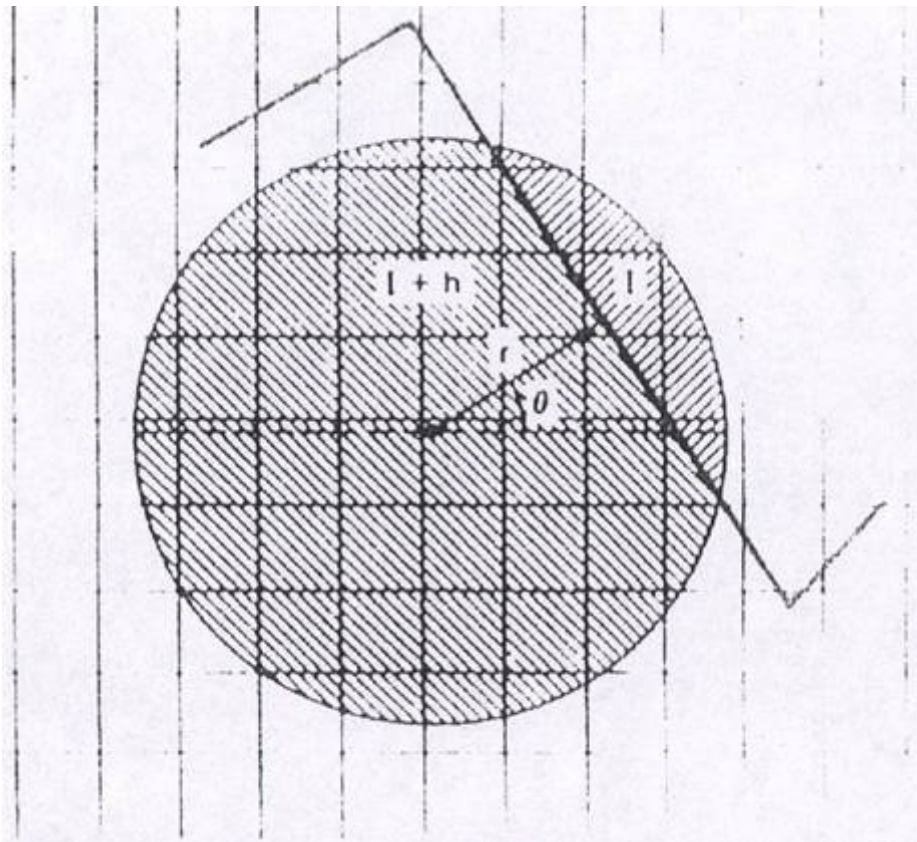
1	1	1
0	0	0
-1	-1	-1

0	1	1
-1	0	1
-1	-1	0

1	1	0
1	0	-1
0	-1	-1

Modello di Hueckel

Modello di un bordo ideale, rappresentato attraverso parametri che vengono variati per trovare il miglior matching sull'immagine



Rilevamento dei crack edge



CRACK EDGE: bordo considerato tra 2 pixel

Ogni pixel ha 4 crack edge:

$$\phi(x,y) = k \pi/2$$

$$0 \leq k < 4$$

$$s(x,y) = |f(x,y) - f(x+h,y-h)|$$

$$h = 1, -1$$

Algoritmo di rilassamento di Prager

Migliora l'immagine dei bordi (crack edge), rafforzando quelli più plausibili ed indebolendo quelli più dubbi, rispetto al valore dei bordi vicini

- 0.** Calcola il valore di fiducia iniziale di ogni bordo $C^0(e)$ come il gradiente normalizzato rispetto al massimo gradiente
- 1.** $k=1$
- 2.** Calcola il tipo di ogni bordo in base alla fiducia dei vicini
- 3.** Modifica la fiducia di ogni bordo $C^k(e)$ sulla base del tipo e del valore di fiducia precedente $C^{k-1}(e)$
- 4.** Verifica se c'è stata la convergenza di tutti i $C^k(e)$ a 0 o 1. Se sì allora stop, altrimenti incrementa k e vai al passo 2

Algoritmo di rilassamento di Prager

Un **bordo** è di **tipo i-j** se i suoi vertici sono di tipo i e di tipo j, rispettivamente

Un **vertice** è di **tipo i** se i massimizza $\text{conf}(i)$

$$\text{conf}(0) = (m-a)(m-b)(m-c)$$

$$\text{conf}(1) = a(m-b)(m-c)$$

$$\text{conf}(2) = ab(m-c)$$

$$\text{conf}(3) = abc$$

$$m = \max(a, b, c, q)$$

a, b, c gradienti normalizzati con $a \geq b \geq c$

q costante (es: $q=0.1$)

Algoritmo di rilassamento di Prager

Un bordo è di tipo i - j se i suoi vertici sono di tipo i e di tipo j , rispettivamente

INCREMENTA i bordi di tipo: 1-1, 1-2, 1-3

DECREMENTA i bordi di tipo: 0-0, 0-2, 0-3

INVARIATI i bordi di tipo: 0-1, 2-2, 2-3, 3-3

INCREMENTO: $C^{k+1}(e) = \min(1, C^k(e) + \delta)$

DECREMENTO: $C^{k+1}(e) = \max(0, C^k(e) - \delta)$

NO VARIAZIONE: $C^{k+1}(e) = C^k(e)$

δ costante (tipicamente $0.1 \leq \delta \leq 0.3$)

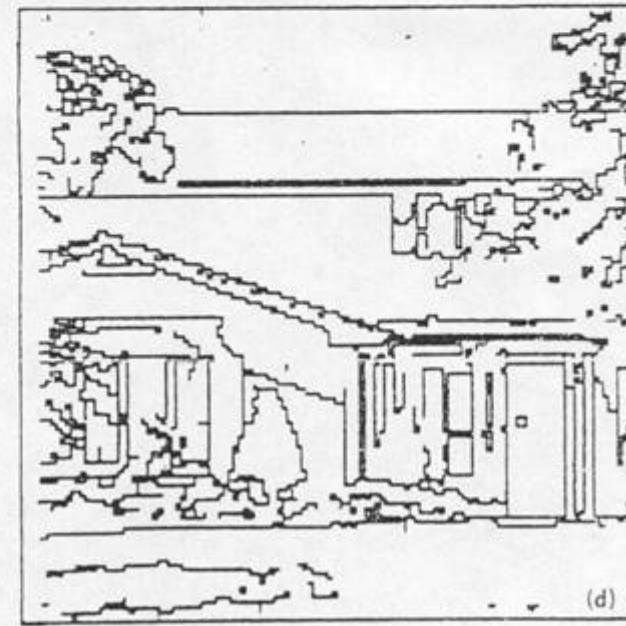
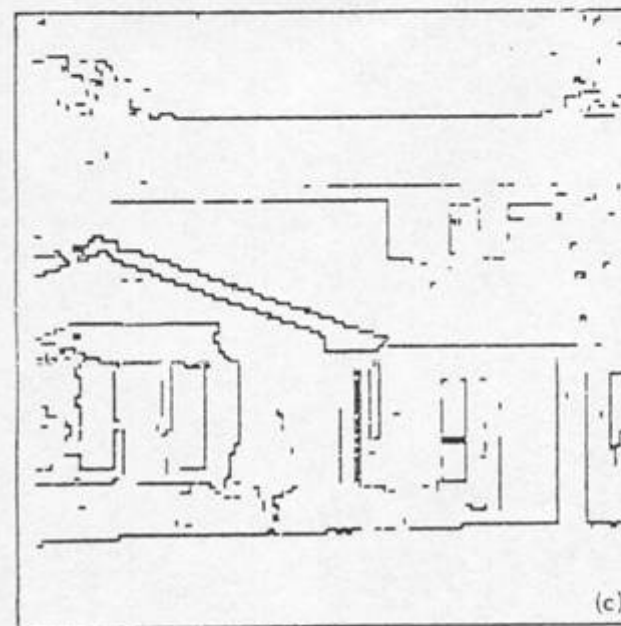
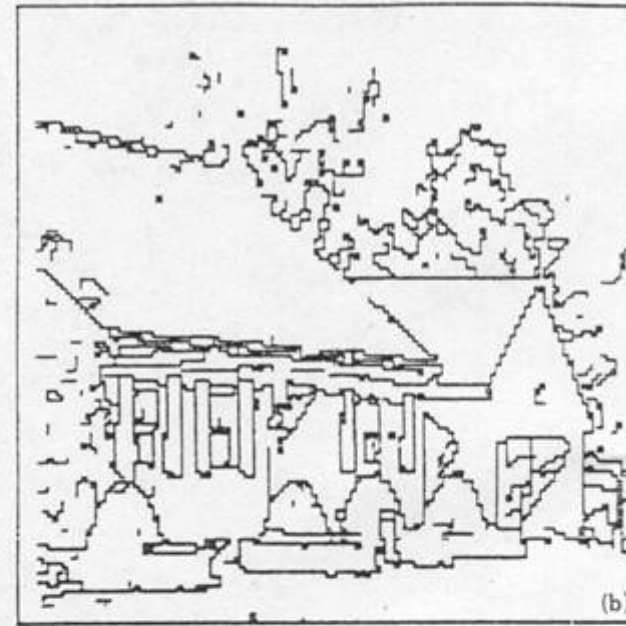
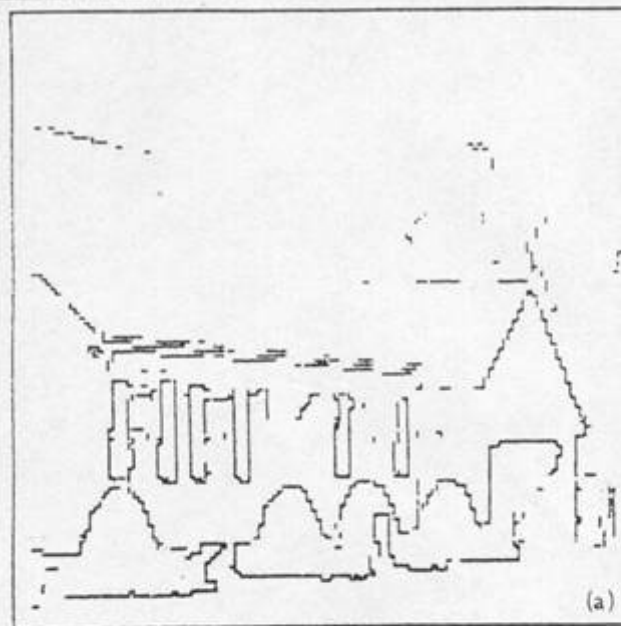


Fig. 3.22 Edge relaxation results. (a) Raw edge data. Edge strengths have been thresholded at 0.25 for display purposes only. (b) Results after five iterations of relaxation applied to (a). (c) Different version of (a). Edge strengths have been thresholded at 0.25 for display purposes only. (d) Results after five iterations of relaxation applied to (c).

Sogliatura

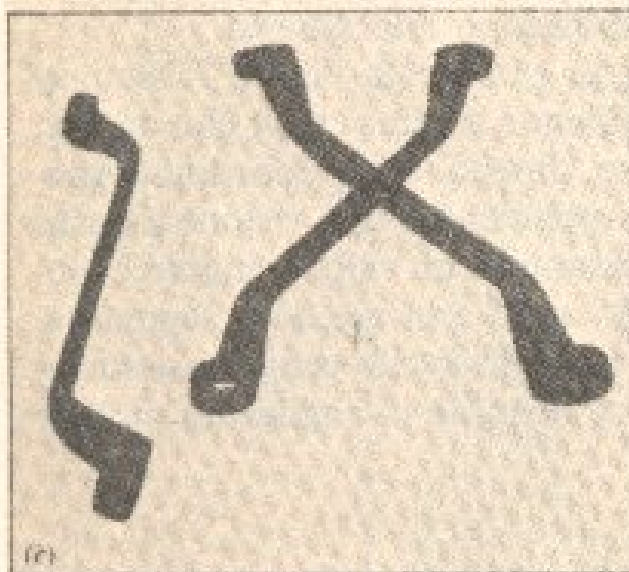
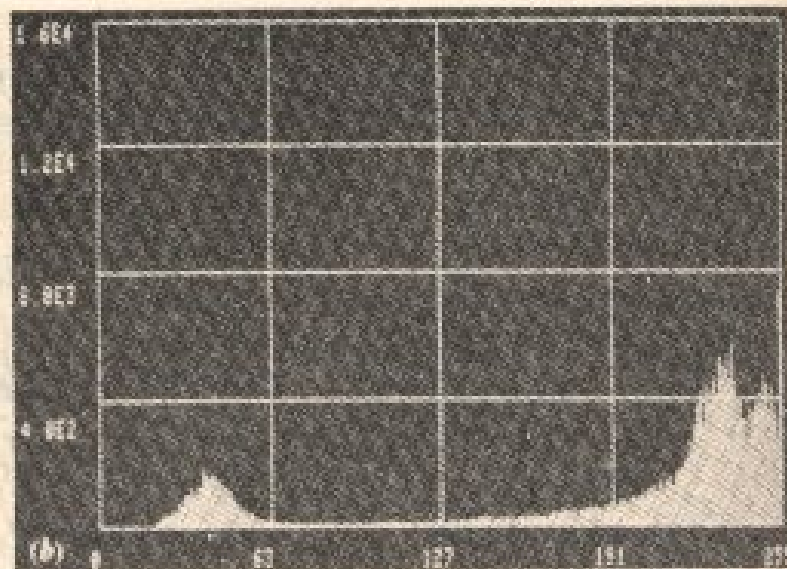
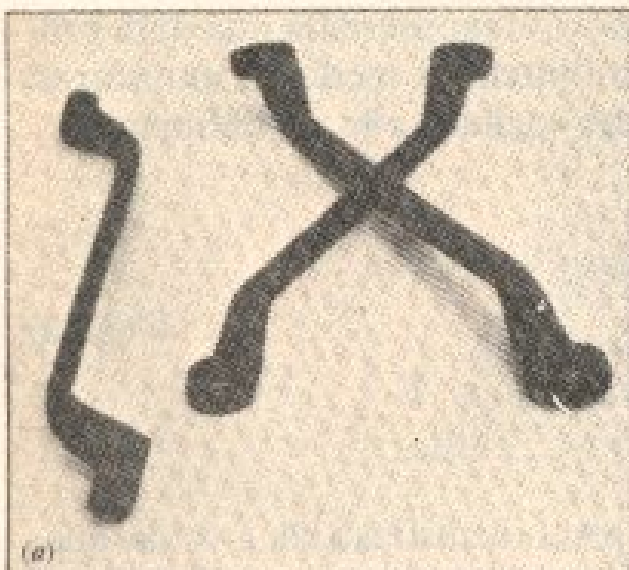


Trasformazione dell'immagine in IMMAGINE BINARIA, attraverso il confronto con una soglia T:

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

T può essere costante o variabile rispetto a x , y , $f(x,y)$ o altre proprietà locali

T può essere determinata empiricamente o con tecniche statistiche



Segmentazione



- Individuazione delle parti costituenti una scena
- CONTORNI: elementi di un'immagine segmentata basati sulla discontinuità
- REGIONI: elementi di un'immagine segmentata basati sulla uniformità

Rilevamento dei contorni



- Ricerca di un contorno noto
 - verifica del bordo sulle normali
 - verifica del bordo sul contorno
 - divide et impera
- confronto con un modello
- confronto con un esempio
- inseguimento
 - scansione e inseguimento
 - metodo del punto finale
 - analisi locale
- metodo di Hough
- ricerca sul grafo

Verifica del bordo sulle normali

4.2.1 Adjusting A Priori Boundaries

This idea was described by [Bolles 1977] (see Fig. 4.2). Local searches are carried out at regular intervals along directions perpendicular to the approximate (a priori) boundary. An edge operator is applied to each of the discrete points along each of these perpendicular directions. For each such direction, the edge with the highest magnitude is selected from among those whose orientations are nearly parallel to the tangent at the point on the nearby a priori boundary. If sufficiently many elements are found, their locations are fit with an analytic curve such as a low-degree polynomial, and this curve becomes the representation of the boundary.

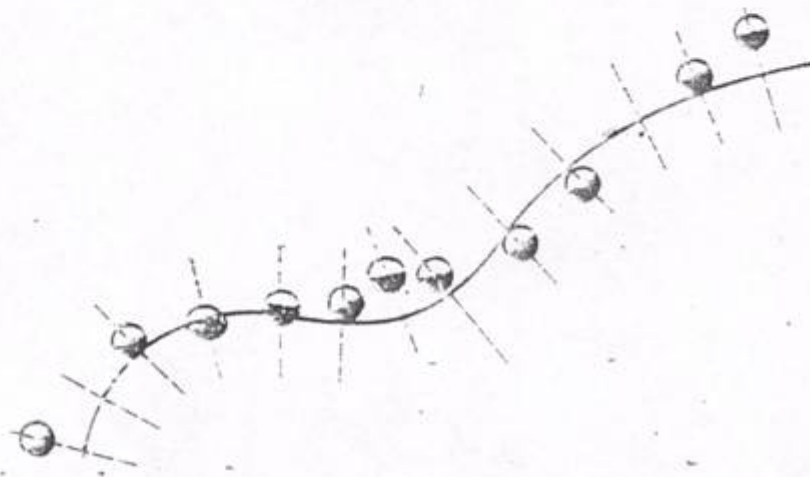


Fig. 4.2 Search orientations from an approximate boundary location.

Verifica del bordo sul contorno

4.2.2 Non-linear Correlation in Edge Space

In this correlation-like technique, the a priori boundary is treated as a rigid template, or piece of rigid wire along which edge operators are attached like beads. The a priori representation thus also contains relative locations at which the existence of edges will be tested (Fig. 4.3). An edge element returned by the edge-operator application "matches" the a priori boundary if its contour is tangent to the template and its magnitude exceeds some threshold. The template is to be moved around the image, and for each location, the number of matches is computed. If the number of matches exceeds a threshold, the boundary location is declared to

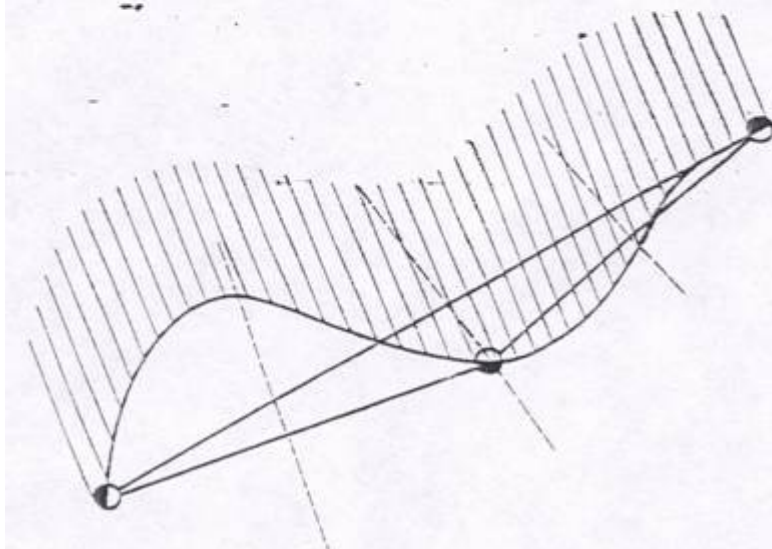


Fig. 4.3 A template for edge-operator application.

be the current template location. If not, the template is moved to a different image point and the process is repeated. Either the boundary will be located or there will eventually be no more image points to try.

Divide et impera

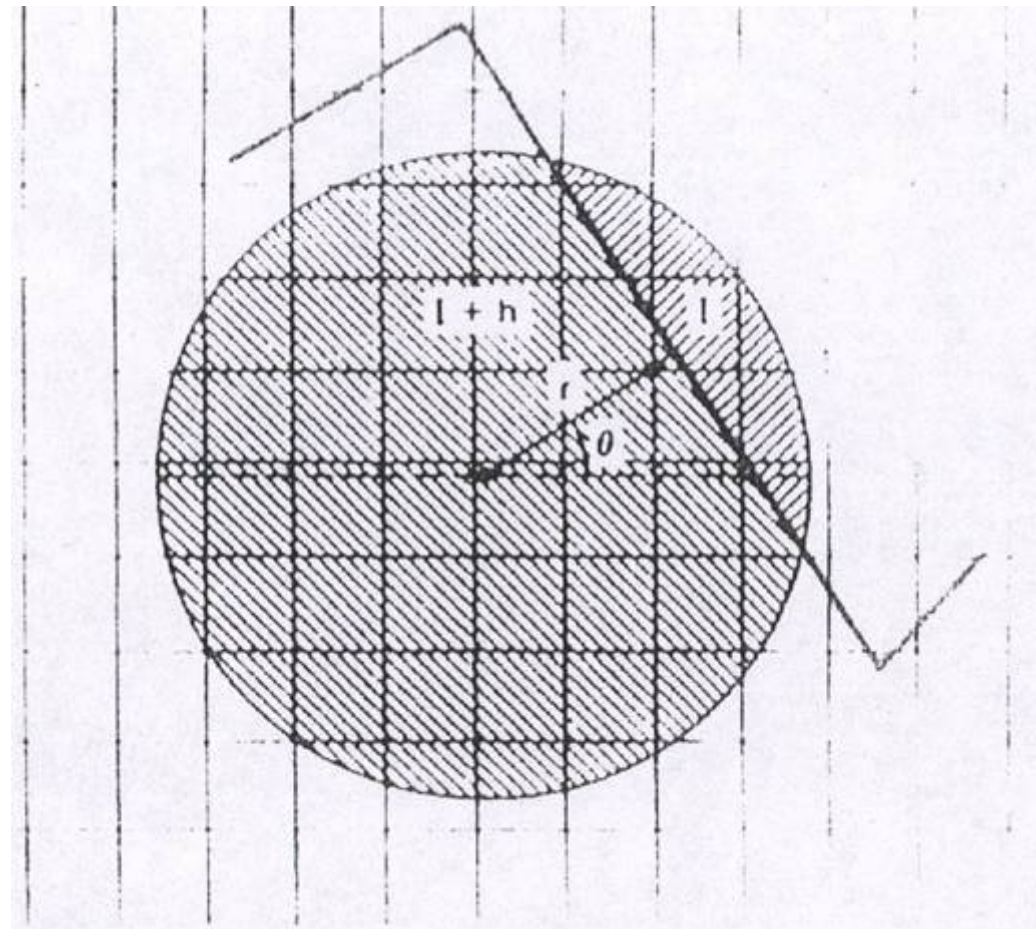
This is a technique that is useful in the case that a low-curvature boundary is known to exist between two edge elements and the noise levels in the image are low (Algorithm 8.1). In this case, to find a boundary point in between the two known points, search along the perpendiculars of the line joining the two points. The point of maximum magnitude (if it is over some threshold) becomes a break point on the boundary and the technique is applied recursively to the two line segments formed between the three known boundary points. (Some fix must be applied if the maximum is not unique.) Figure 4.4 shows one step in this process. Divide-and-conquer boundary detection has been used to outline kidney boundaries on computed tomograms (these images were described in Section 2.3.4 [Selfridge et al. 1979]).



Confronto con un modello

Modello di Hueckel

Modello di un bordo ideale, rappresentato attraverso parametri che vengono variati per trovare il miglior matching sull'immagine



Confronto con un esempio (template)

0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0
0	0	1	0	0

Contorno verticale

0	0	0	0	0
0	0	0	0	0
1	1	1	1	1
0	0	0	0	0
0	0	0	0	0

Contorno orizzontale

0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0

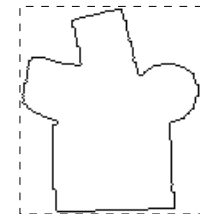
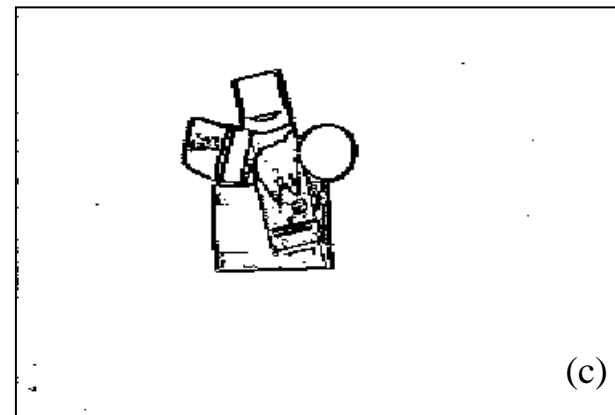
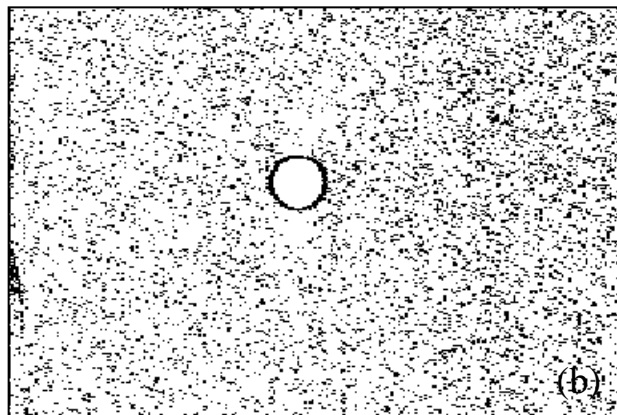
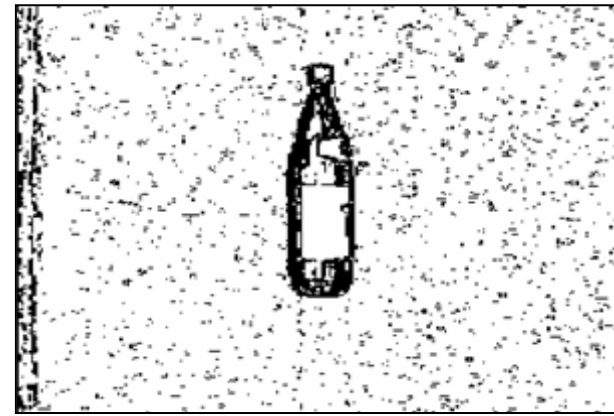
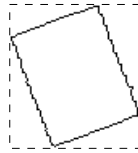
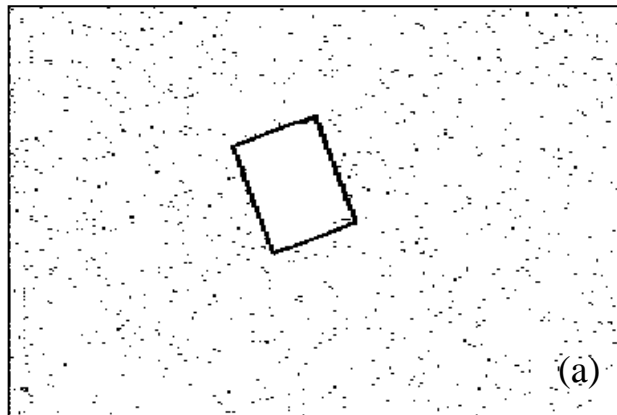
Contorno diagonale

Scansione e inseguimento

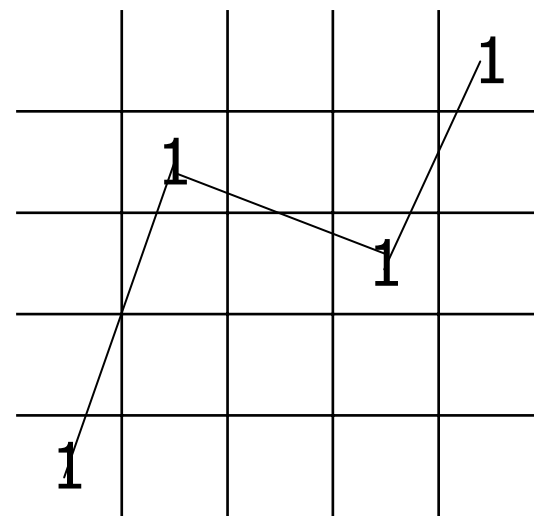
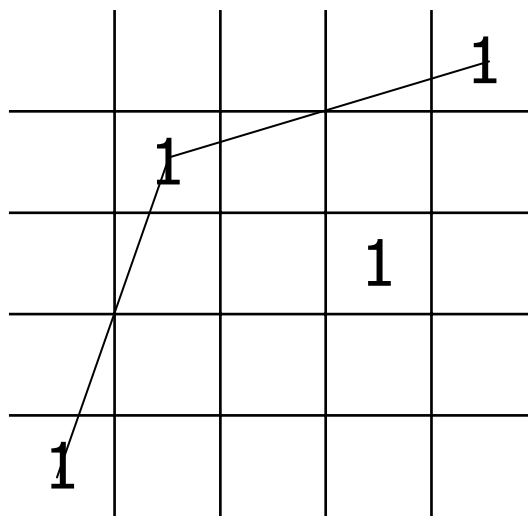
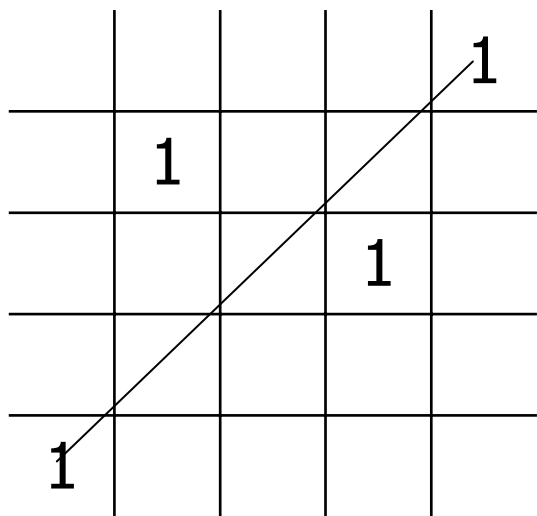


1. Scandisci l'immagine da sinistra a destra dall'alto verso il basso fino a trovare un edge
2. Collegalo al precedente e ricerca altri edge in un suo intorno
3. (a) se c'è un solo edge: vai a 2.
(b) se ce ne sono 2 o più: scegline uno e vai a 2; memorizza gli altri
(c) se non ce ne sono, l'edge considerato è un terminale di contorno. Se ci sono edge memorizzati scegline uno e vai a 2, altrimenti vai a 1.

Un esempio di rilevamento dei contorni con una tecnica di inseguimento



Metodo del punto finale



Analisi locale



Il metodo consiste nel collegare gli edge di un intorno che hanno caratteristiche simili

$$|S(x,y) - S(x',y')| < T \text{ (soglia)}$$

$$|\phi(x,y) - \phi(x',y')| < A \text{ (soglia)}$$

Metodo di Hough



Rileva ogni contorno che possa essere espresso da una curva parametrica, confrontando l'immagine dei bordi con tutte le curve ottenute variando i valori dei parametri

Trasformata di Hough



- E' una tecnica che permette di riconoscere particolari configurazioni di punti presenti nell'immagine, come segmenti, curve o altre forme prefissate.
- E' un tipico *operatore globale*.
- Il principio fondamentale è che la forma cercata può essere espressa tramite una funzione nota che fa uso di un insieme di parametri.
- Una particolare istanza della forma cercata è quindi completamente precisata dal valore assunto dall'insieme di parametri.

Trasformata di Hough – Caso della retta

- Per esempio, assumendo come rappresentazione della retta la forma:

$$y=ax+b$$

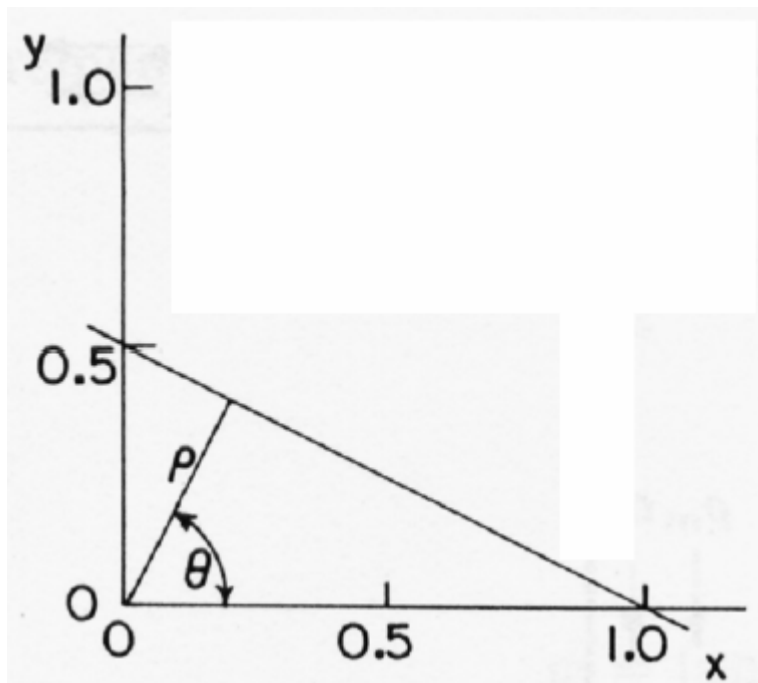
- qualunque retta è completamente specificata dal valore dei parametri (a,b).

- Se si assume un tipo di rappresentazione diversa, quale la forma normale di Hesse

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

- la retta è completamente specificata dalla coppia (ρ, ϑ) .

Esempio



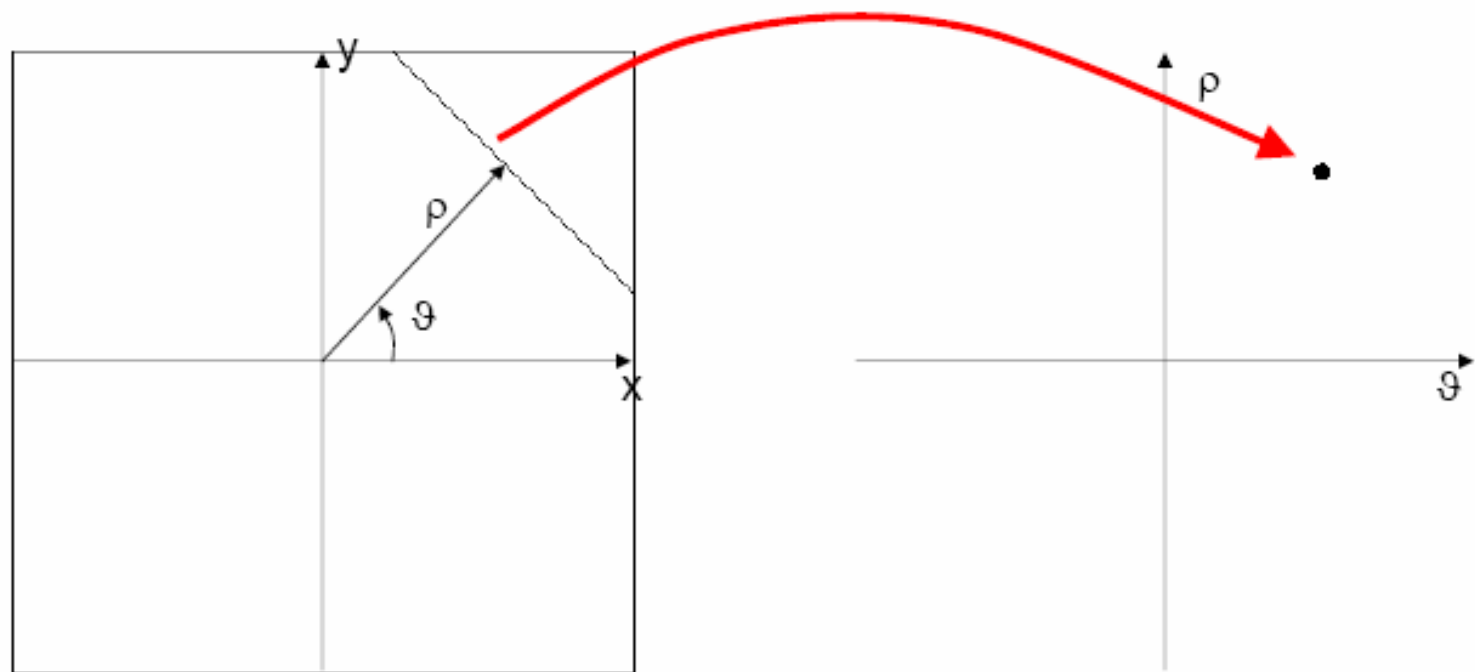
- La retta in figura è identificata dalla coppia:

$$(a,b)=(-0.5,0.5)$$

- o dalla coppia:

$$(\rho, \theta)=(0.447, 1.107)$$

Il piano trasformato



Domande

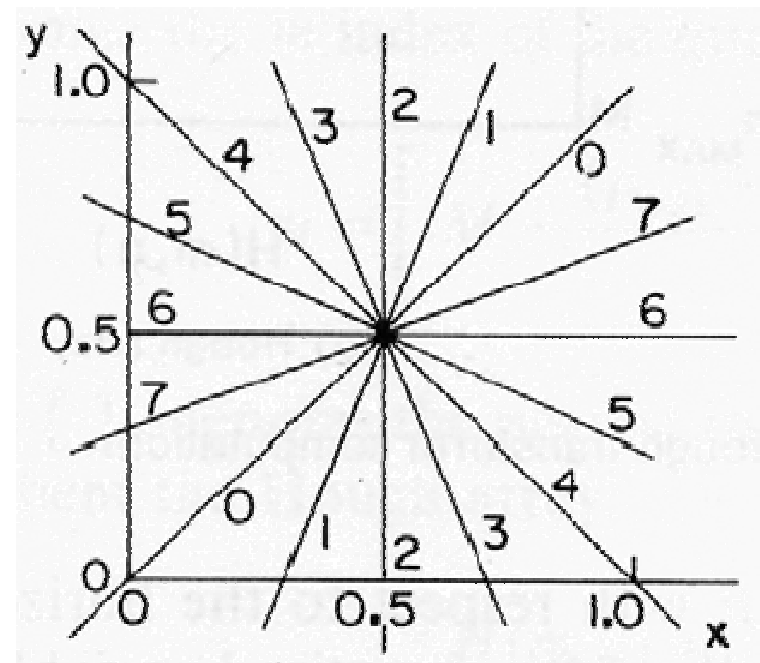


- Nell'immagine in analisi, l'unica informazione disponibile è costituita dall'insieme di punti che appartiene al foreground.

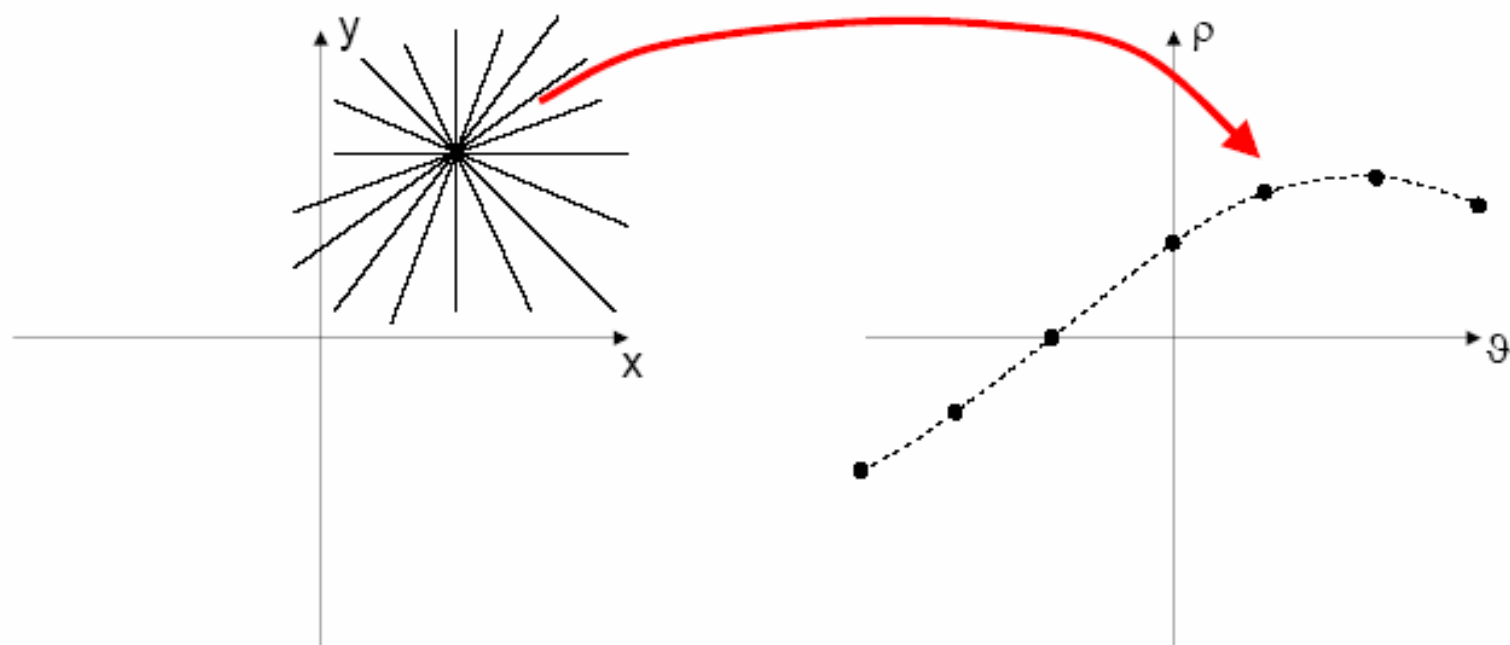
- 1. Come poter sfruttare questa trasformata ai fini della individuazione di segmenti in un'immagine ?
- 2. Qual è la trasformata di un punto nell'immagine ?

Trasformata di un punto

- Nel piano dell'immagine, un punto è identificato dall'intersezione di più rette.
- Quindi, ad ogni punto P corrisponde, nel piano dei parametri, la curva formata dai punti immagine delle rette passanti per P .



Trasformata del punto

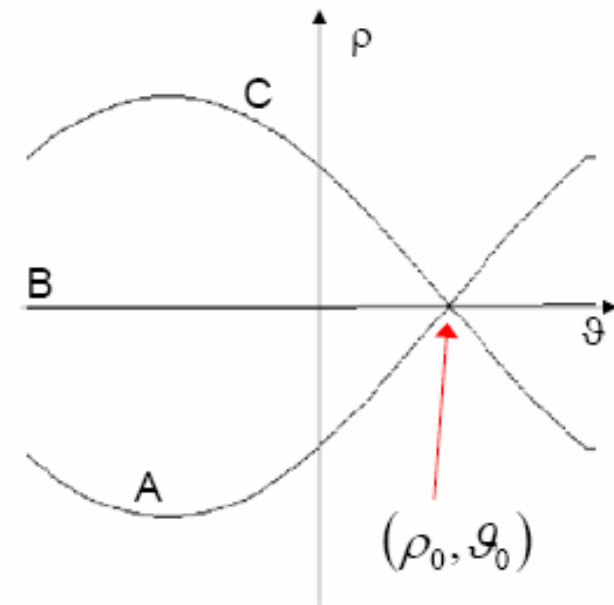
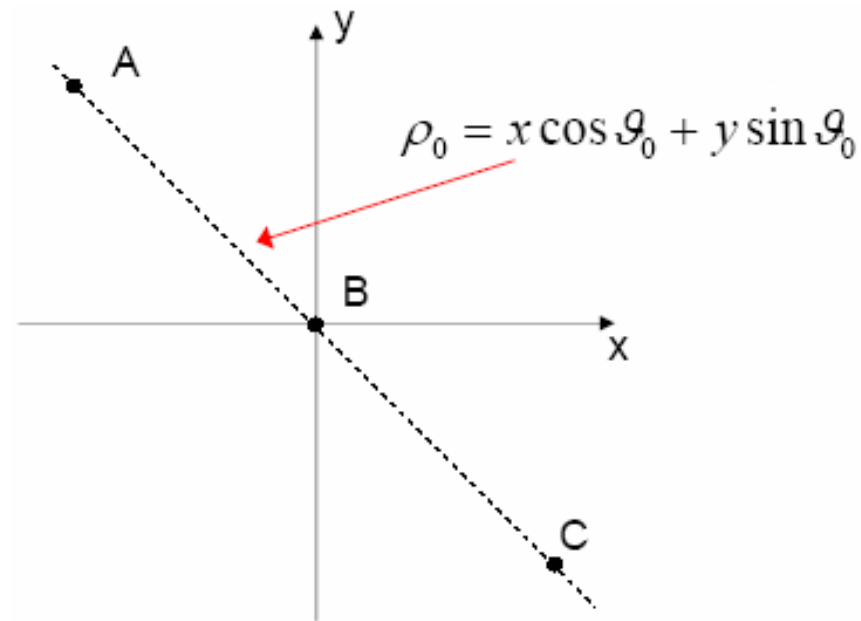


Domanda



- Che cosa succede se nell'immagine ci sono dei punti allineati su una stessa retta ?
- Sul piano dei parametri, le curve che corrispondono alle trasformazioni dei vari punti si intersecano in un punto del piano trasformato che è l'immagine della retta su cui giacciono i punti.

Individuazione della retta sul piano trasformato



Implementazione trasformata di Hough

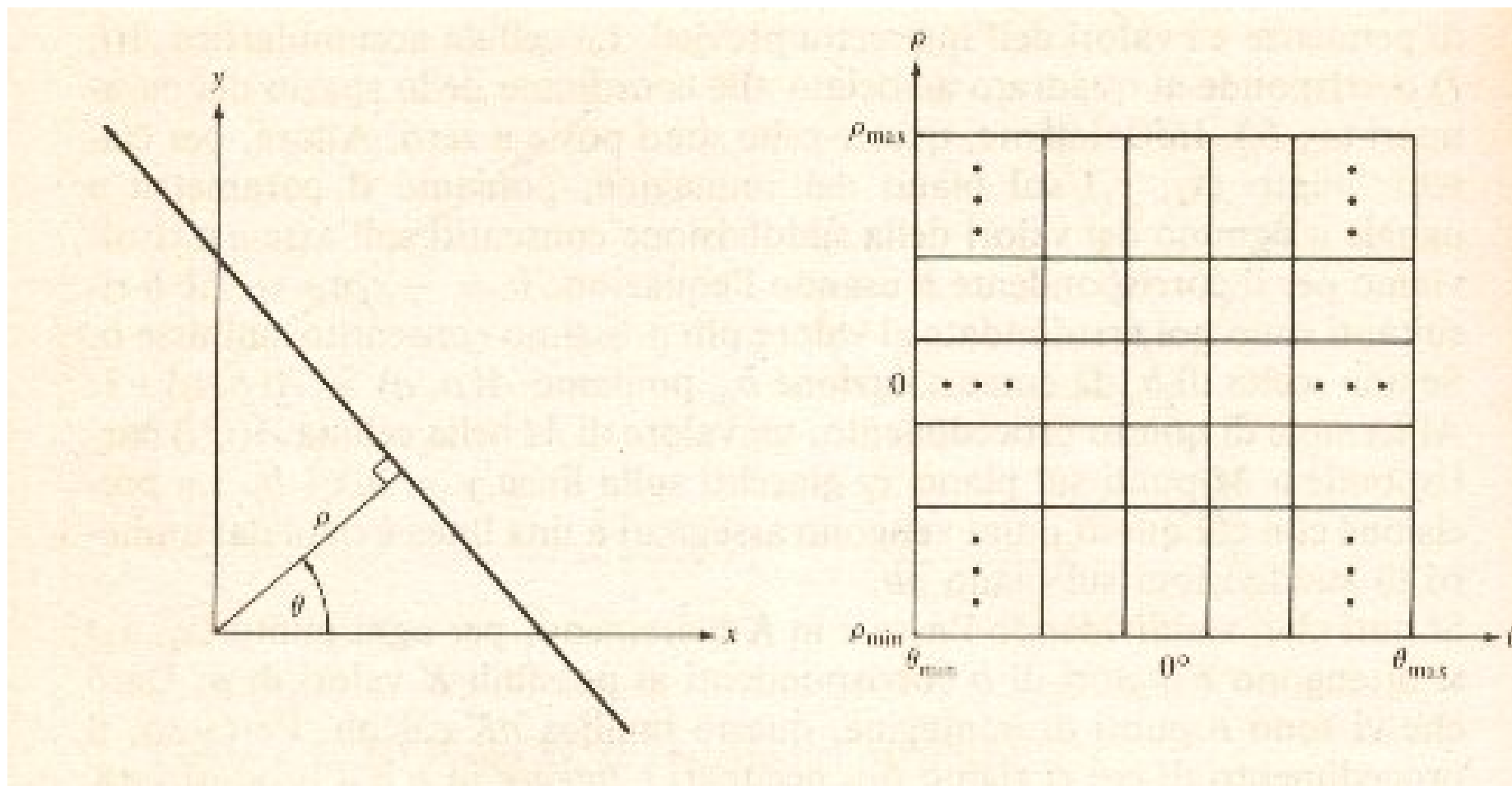
– caso della retta



- Si consideri una discretizzazione del piano dei parametri (ρ, ϑ) .
- Ciò permette di rappresentare tale piano su una matrice $H(m, n)$ i cui indici di riga e di colonna corrispondono ai valori quantizzati di ρ e ϑ .
- Gli intervalli di variazione di ρ e ϑ sono fissati sulla base delle caratteristiche dell'immagine originale.
 - Tipicamente $-\rho_{\max} \leq \rho \leq \rho_{\max}$, $-\pi/2 \leq \vartheta \leq \pi/2$,
 - dove $\rho_{\max} = 0.5 * (\mathbf{NR}^2 + \mathbf{NC}^2)^{1/2}$ e $(\mathbf{NR}, \mathbf{NC})$ sono le dimensioni dell'immagine originale.
 - Il numero dei livelli di quantizzazione va poi scelto in base all'accuratezza desiderata.
 - Una scelta quasi sempre soddisfacente è $\max(\mathbf{NR}, \mathbf{NC})$.

Metodo di Hough: rappresentazione polare di una retta

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



Algoritmo di Hough



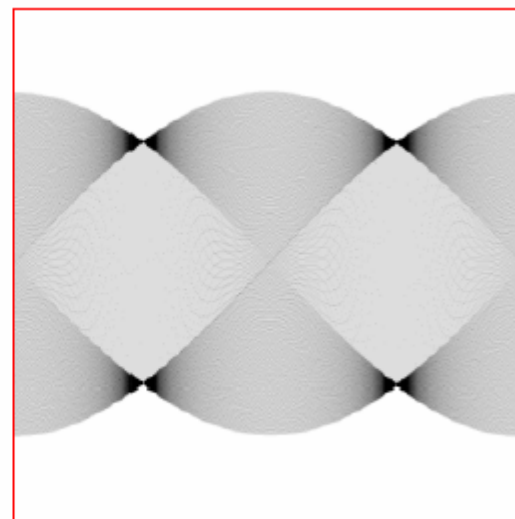
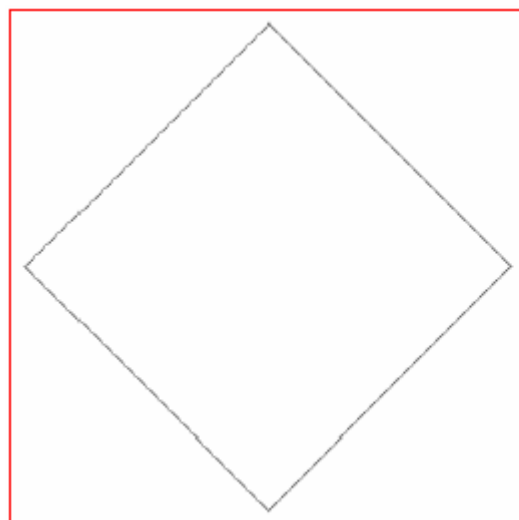
1. Quantizza lo spazio dei parametri tra valori appropriati di massimo e minimo
2. Crea una matrice di accumulazione di dimensioni pari al numero dei parametri, inizializzata a 0
3. Per ogni punto di edge dell'immagine, incrementa tutti i punti della matrice di accumulazione corrispondenti ai valori dei parametri delle curve su cui l'edge giace
4. I massimi locali nella matrice rappresentano i valori dei parametri che individuano le curve che meglio approssimano il contorno

Algoritmo

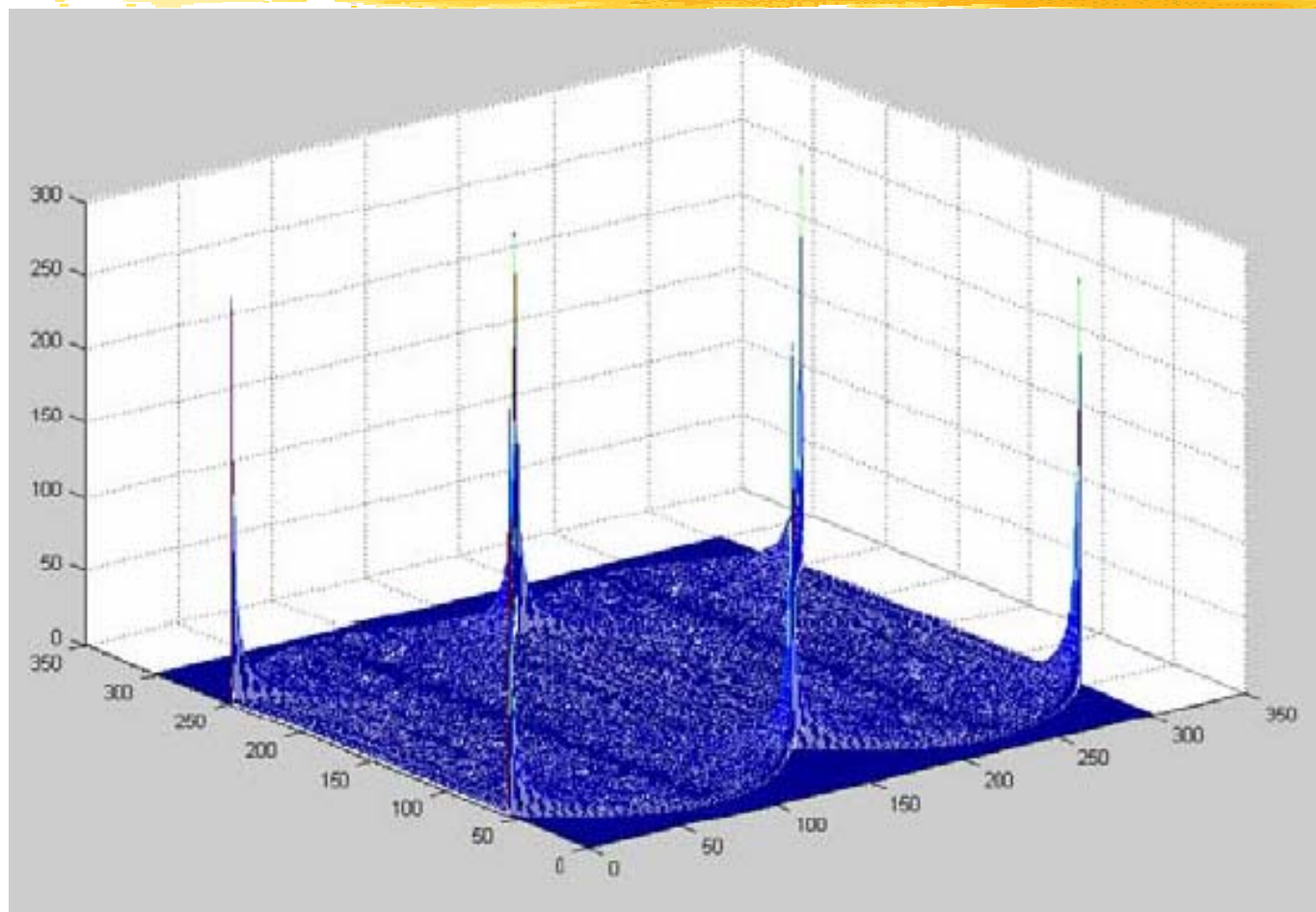


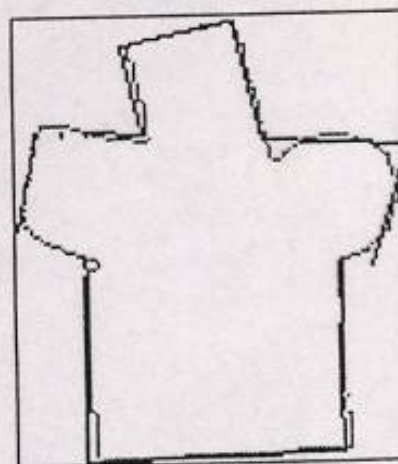
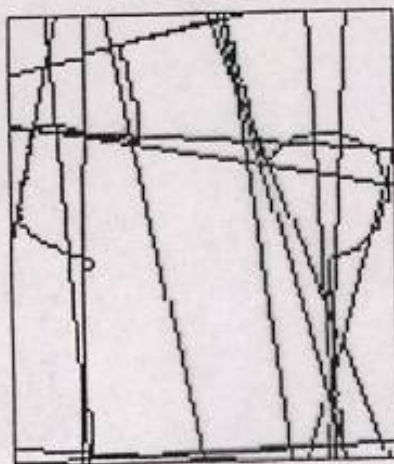
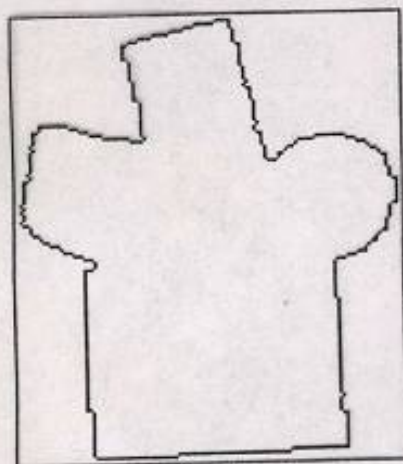
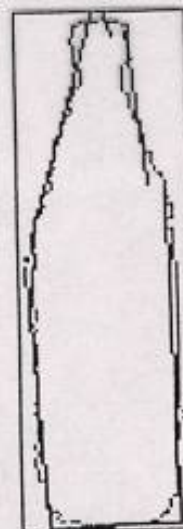
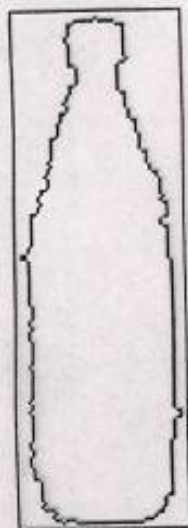
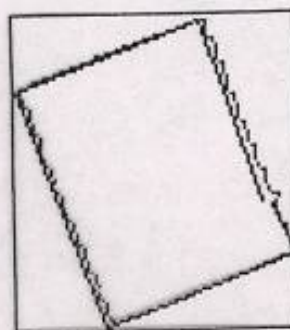
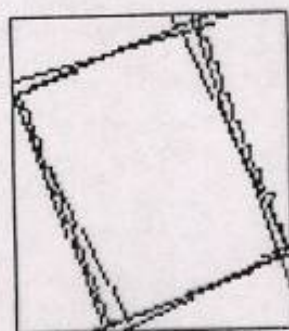
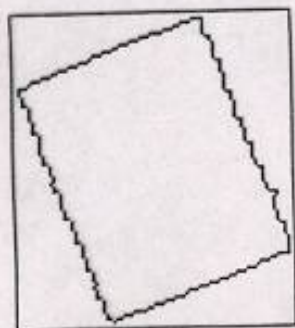
1. Si azzeri la matrice $H(.,.)$;
2. Per ogni punto $P \in F$, $P=(x,y)$
 1. per ϑ_n che varia tra $-\pi/2$ e $\pi/2$ con passo $d\vartheta$
 1. si valuta $p(n)=x*\cos(\vartheta_n)+y*\sin(\vartheta_n)$
 2. si ricava l'indice m corrispondente a $p(n)$
 3. si incrementi $H(m,n)$
 2. end
3. End
4. Si individuino i massimi locali su $H(.,.)$ corrispondenti ai parametri dei segmenti individuati

Esempio: figura piana



Esempio: matrice accumulazione

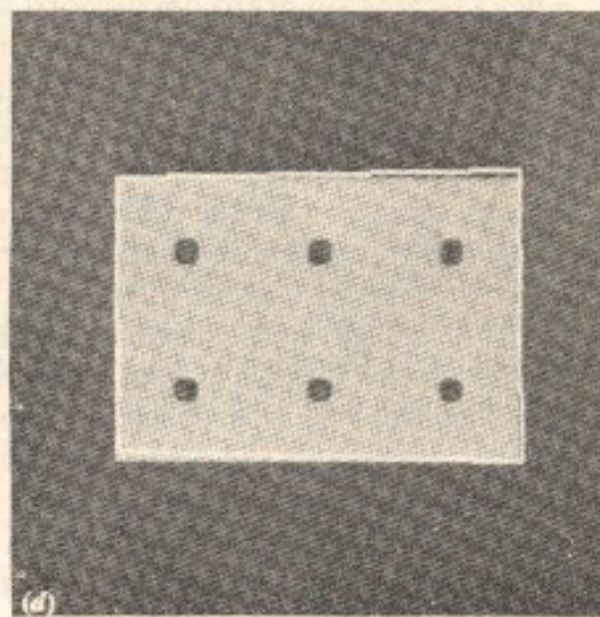
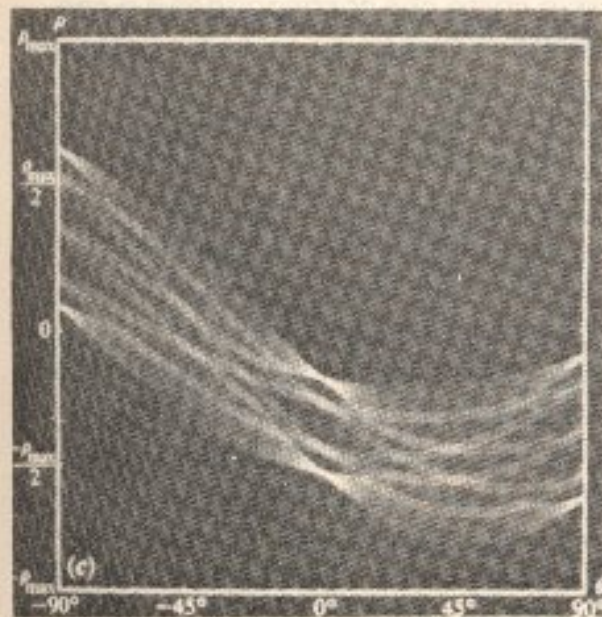
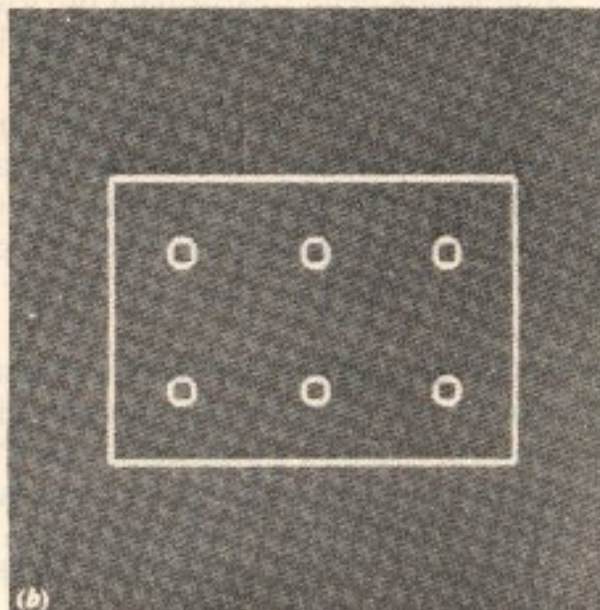
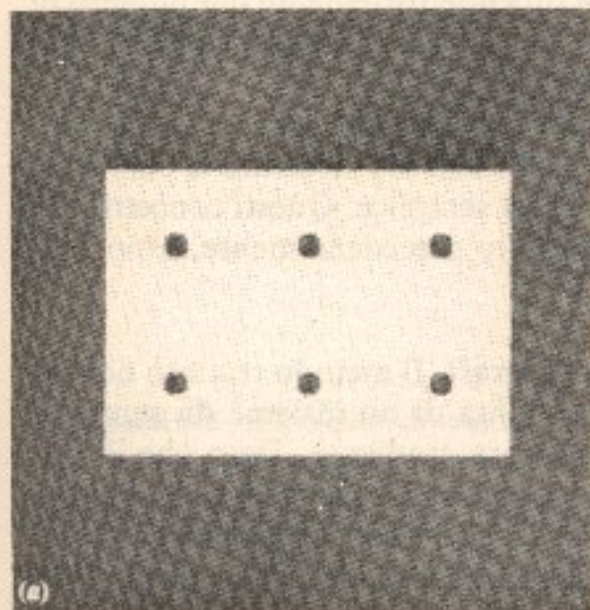




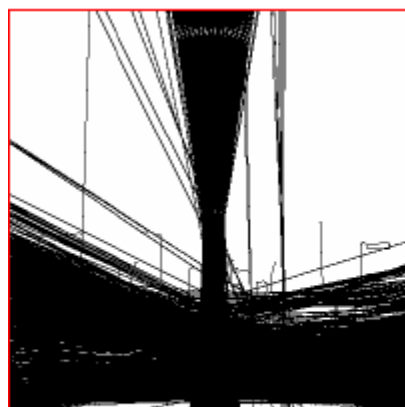
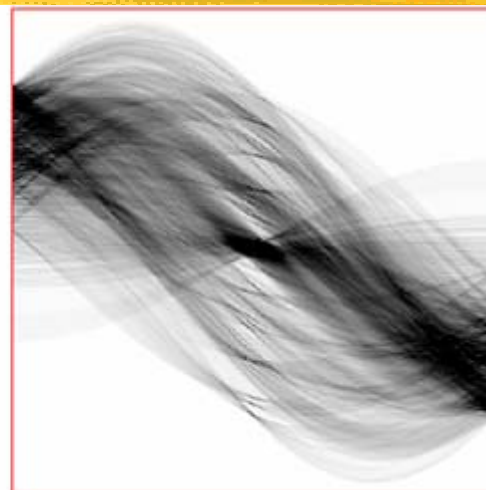
(a)

(b)

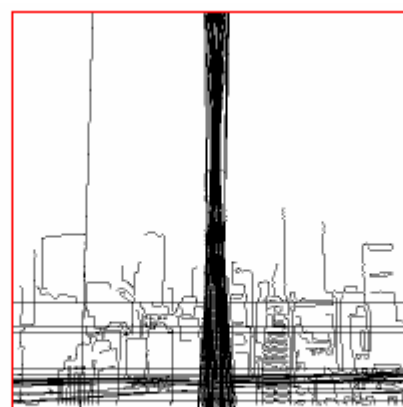
(c)



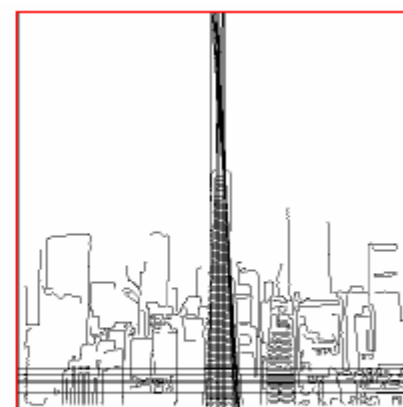
Sogliatura immagini reali



Soglia: 101



Soglia: 140



Soglia: 160

Conclusioni



- E' possibile – **ma molto oneroso** - utilizzare la trasformata di Hough per individuare cerchi, tenendo conto dell'equazione $(x-a)^2+(y-b)^2=c^2$
 - In questo caso è possibile lavorare su:
 - un piano dei parametri (a,b) , fissando il raggio c dei cerchi da individuare
 - uno spazio dei parametri (a,b,c) , facendo variare c in un intervallo finito.
- E' stata inoltre proposta (Ballard) una generalizzazione della trasformata di Hough che permette di individuare oggetti di forma qualunque.
- **Non è adatta a applicazioni real-time**

Metodo della ricerca su grafo

NODI: crack edge (p,q)

con p,q pixel di intensità $f(p)$ e $f(q)$

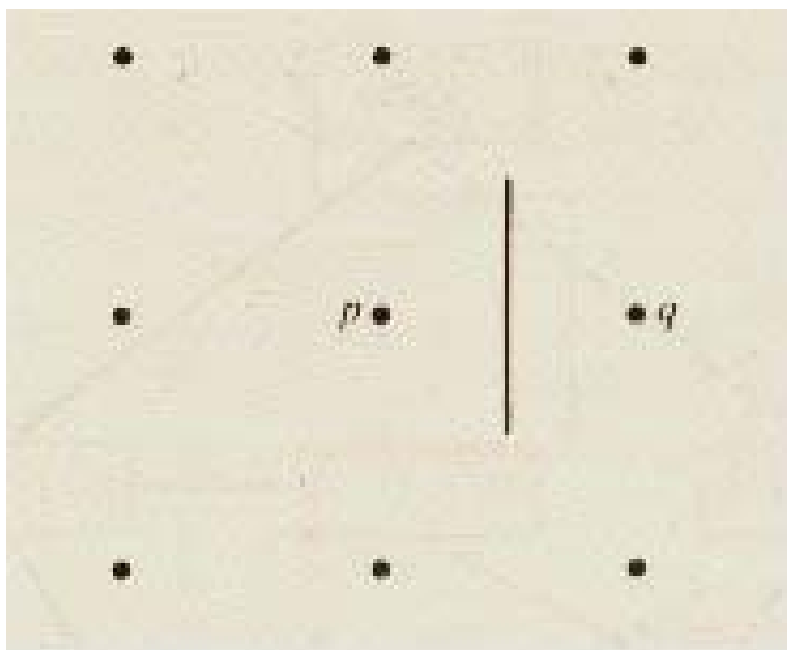
ARCHI: collegamenti tra crack edge che in sequenza possono far parte di un contorno

COSTO (associato all'arco entrante in (p,q)):

$$c(p,q) = H - [f(p) - f(q)]$$

con H massima intensità nell'immagine

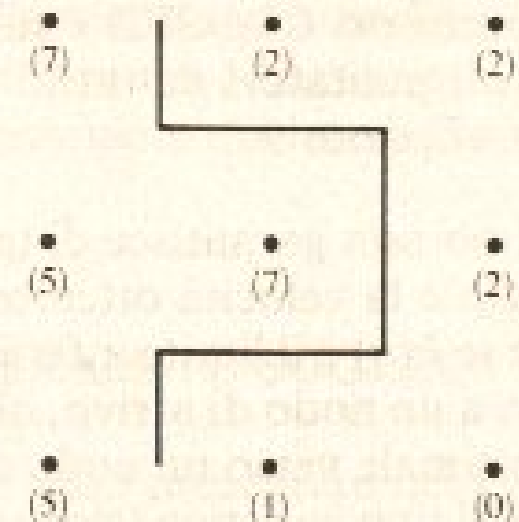
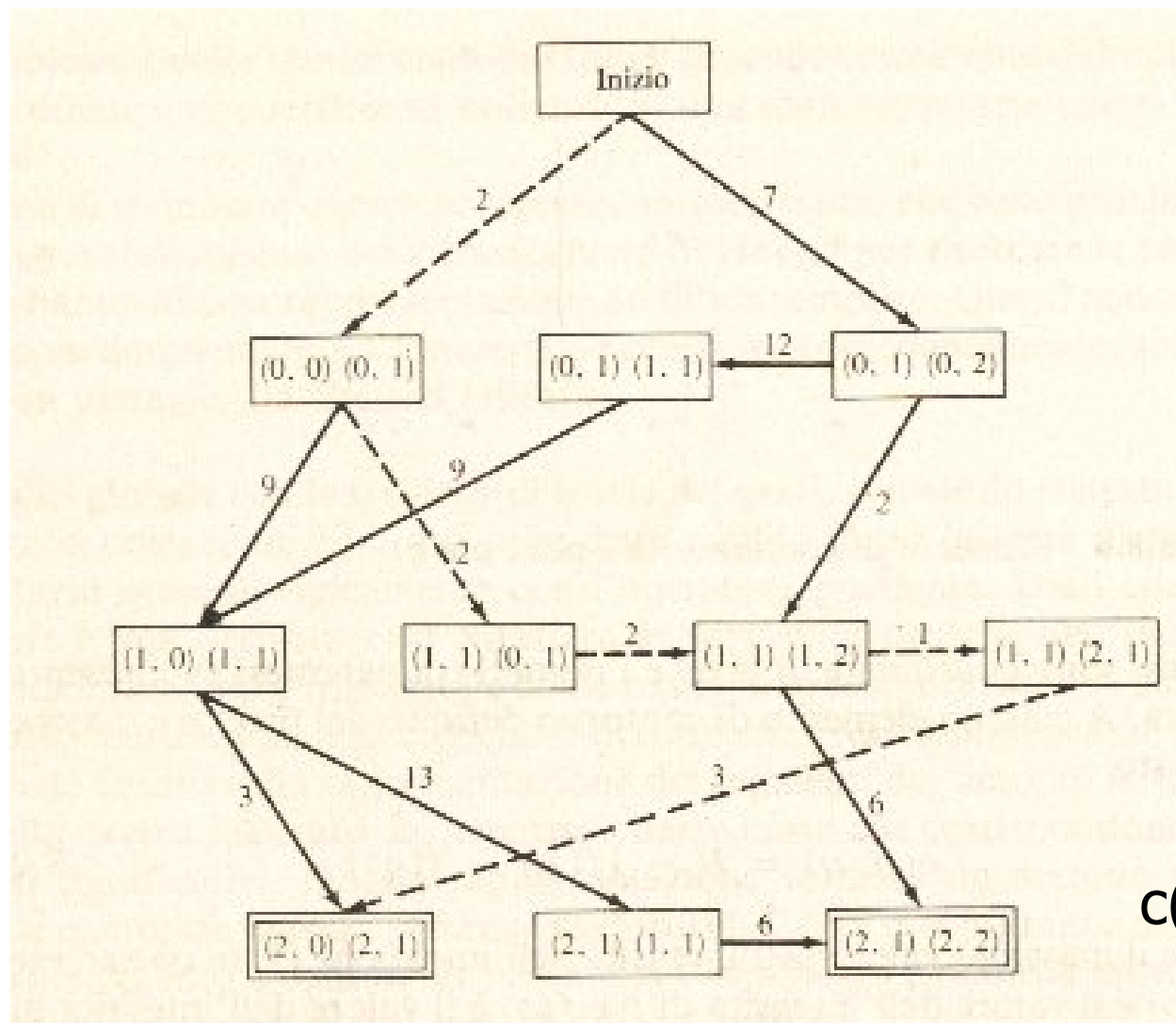
Metodo della ricerca su grafo



	0	1	2
0	(7)	(2)	(2)
1	(5)	(7)	(2)
2	(5)	(1)	(0)

$$c(p,q) = H - [f(p) - f(q)]$$

Metodo della ricerca su grafo



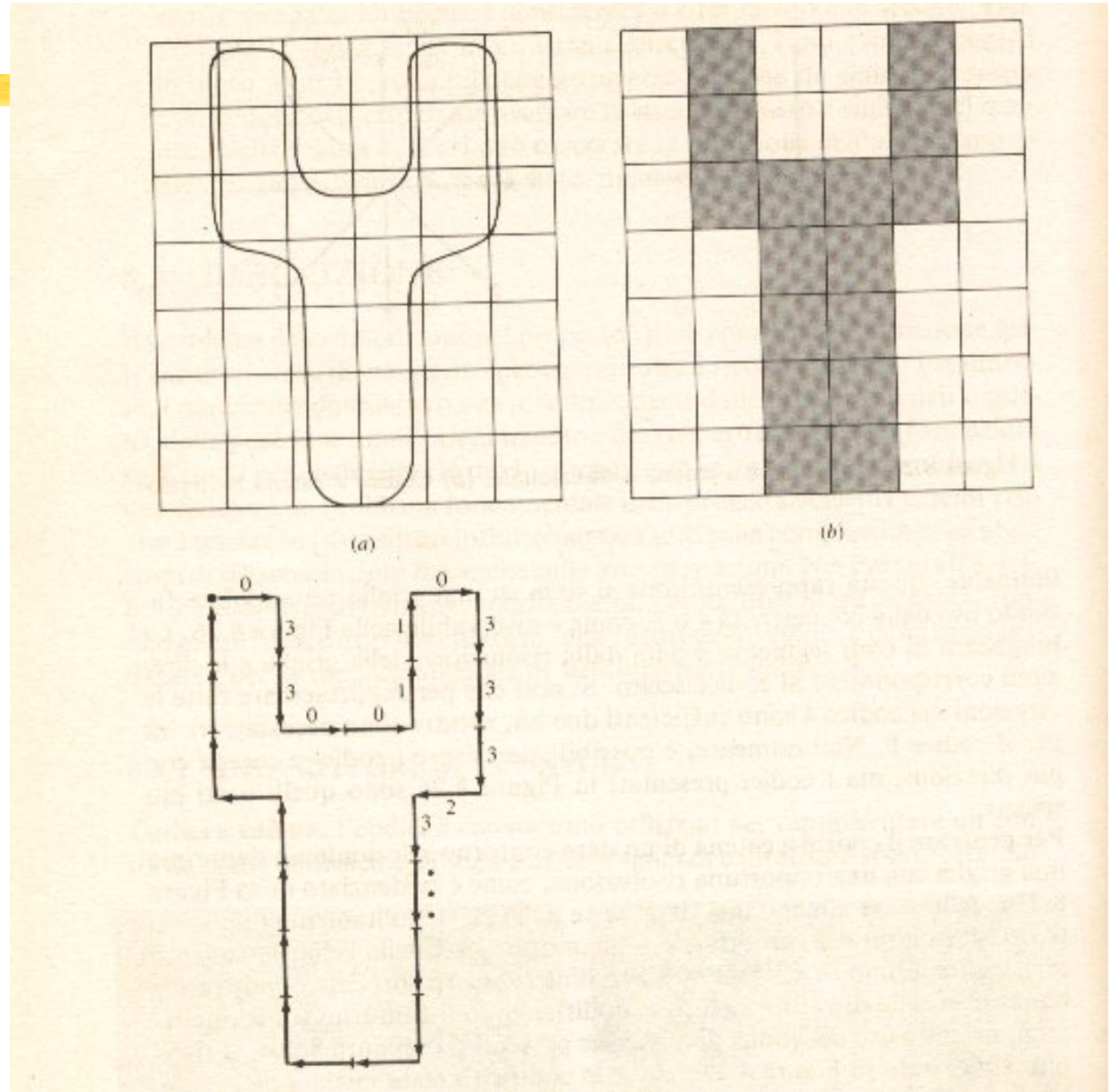
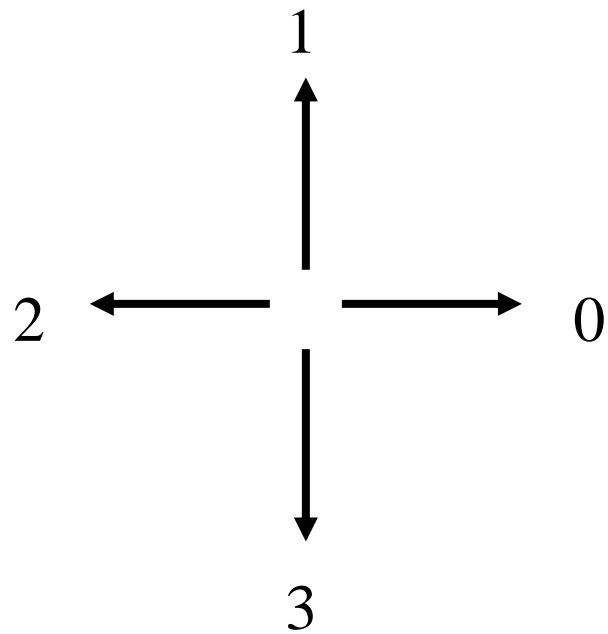
$$c(p,q) = H - [f(p) - f(q)]$$

Rappresentazione dei contorni



- Codici a catena e numeri di forma
- Forme caratteristiche

Codici a catena e numeri di forma





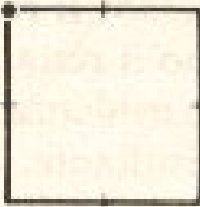
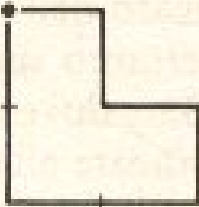

Codici a catena e numeri di forma



Normalizzazione rispetto alla rotazione:
utilizzare la differenza del codice a catena
contando in senso antiorario il numero delle
direzioni che separano due elementi adiacenti

Es: 10103322
3133030
33133030

Codici a catena e numeri di forma

	Ordine 4	Ordine 6	
			
Codice catena:	0 3 2 1	0 0 3 2 2 1	
Differenza:	3 3 3 3	3 0 3 3 0 3	
Numero delle forme:	3 3 3 3	0 3 3 0 3 3	
	Ordine 3		
			
Codice catena:	0 0 3 3 2 2 1 1	0 3 0 3 2 2 1 1	0 0 0 3 2 2 2 1
Differenza:	3 0 3 0 3 0 3 0	3 3 1 3 3 0 3 0	3 0 0 3 3 0 0 3
Numero di forma:	0 3 0 3 0 3 0 3	0 3 0 3 3 1 3 3	0 0 3 3 0 0 3 3

Calcolo dell'area con i codici a catena

1. Area=0;
2. $y=0$;
3. per ogni elemento del codice a catena

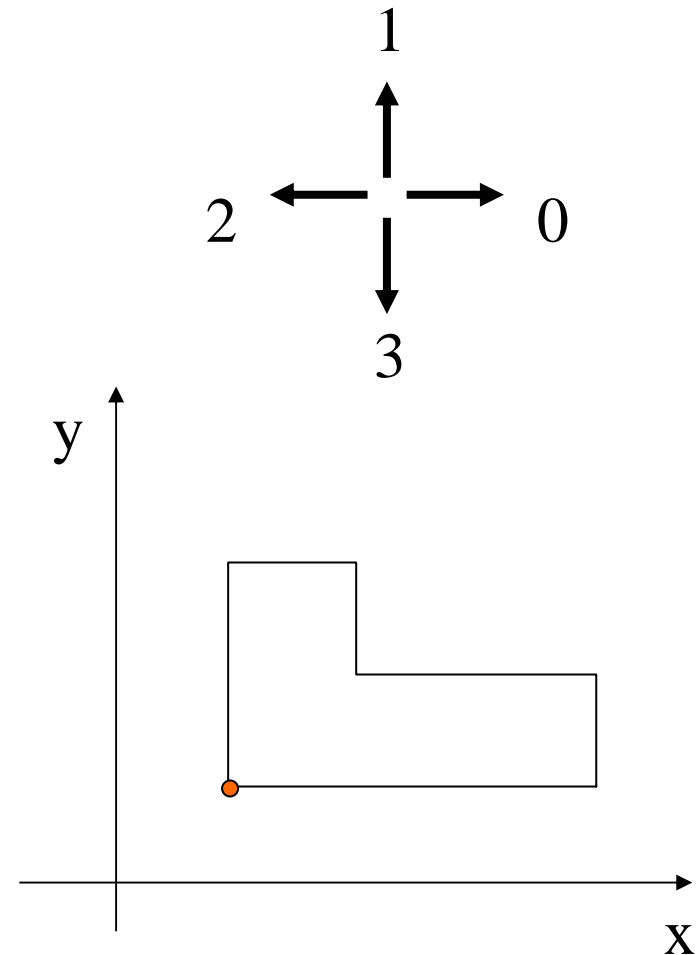
case direzione

0: $\text{area} += y$;

1: $y++$;

2: $\text{area} -= y$;

3: $y--$;



Forme caratteristiche

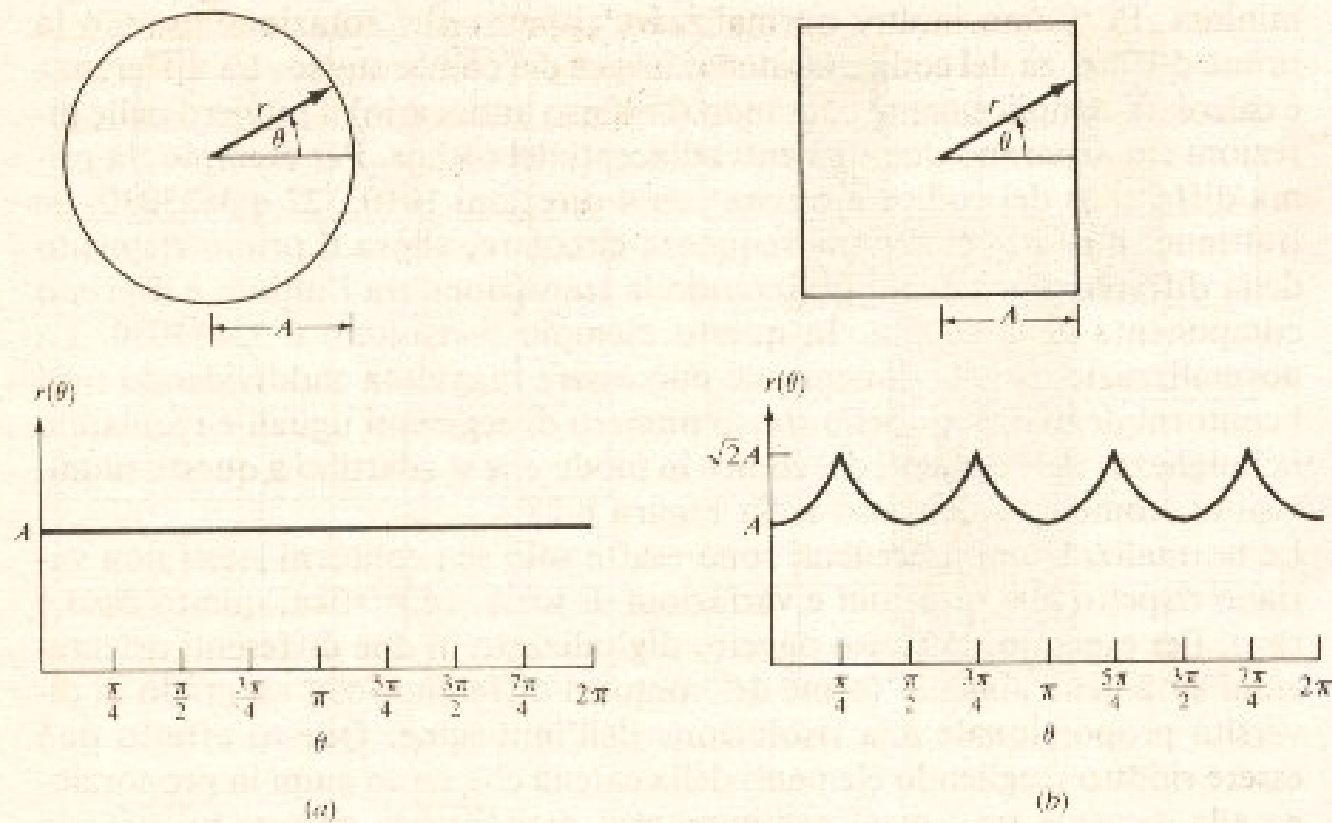



Figura 8.29 Due semplici forme di contorno e le rispettive forme caratteristiche con la distanza in funzione dell'angolo. In (a), $r(\theta)$ è costante, mentre in (b), $r(\theta) = A \sec \theta$.

Rilevamento delle regioni



- Tecniche locali:
 - colorazione della mappa
- Tecniche globali:
 - sogliatura
 - region growing by splitting & merging

Colorazione della mappa (blob coloring)

Algorithm 5.1: Blob Coloring

Let the initial color, $k = 1$. Scan the image from left to right and top to bottom.

If $f(x_C) = 0$ then continue

else

begin

if ($f(x_U) = 1$ and $f(x_L) = 0$)
then color (x_C) := color (x_U)

if ($f(x_L) = 1$ and $f(x_U) = 0$)
then color (x_C) := color (x_L)

if ($f(x_L) = 1$ and $f(x_U) = 1$)

then begin

color (x_C) := color (x_L)

color (x_L) is equivalent to color (x_U)

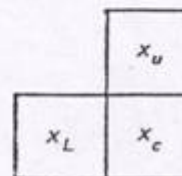
end

comment: two colors are equivalent.

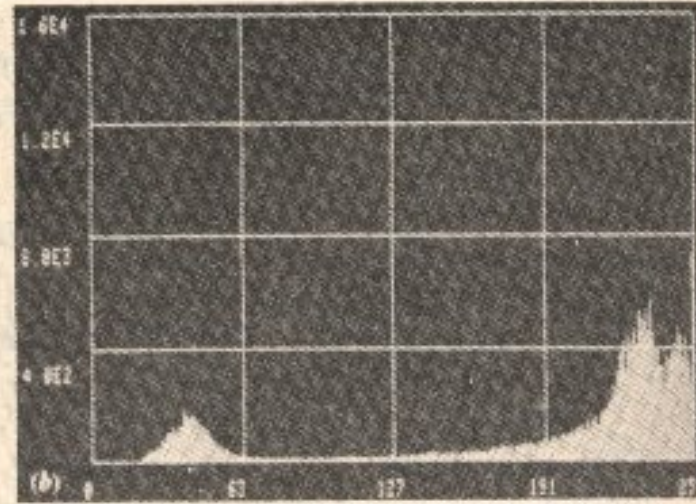
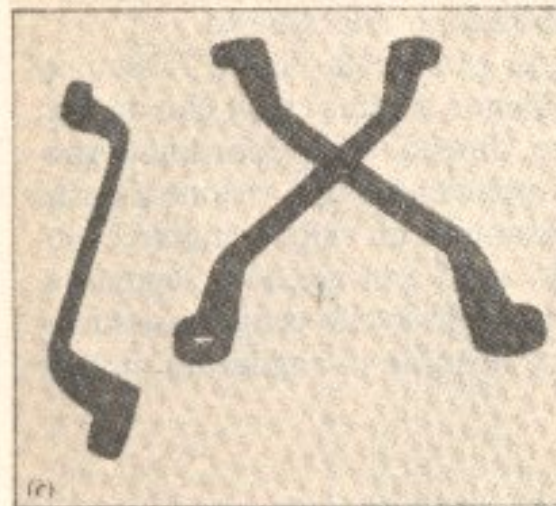
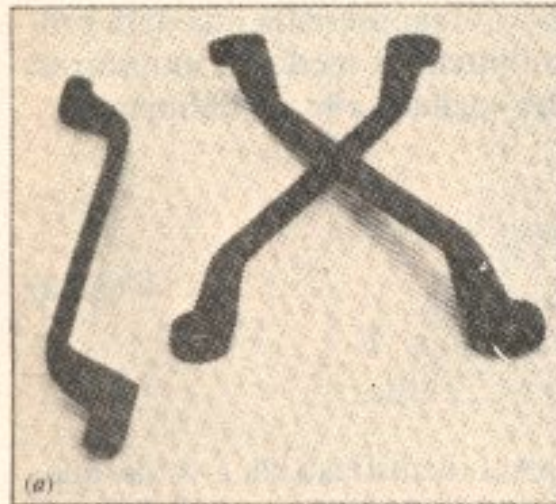
if ($f(x_L) = 0$ and $f(x_U) = 0$)

then color (x_C) := k , $k := k + 1$

comment: new color



Sogliatura



Splitting & merging

The method of [Brice and Fennema 1970] uses three criteria for merging regions, reflecting a transition from local measurements to global measurements. These criteria use measures of boundary strength s_{ij} and w_{ij} defined as

$$s_{ij} = |f(x_i) - f(x_j)| \quad (5.9)$$

$$w_{ij} = \begin{cases} 1 & \text{if } s_{ij} < T_1 \\ 0 & \text{otherwise} \end{cases} \quad (5.10)$$

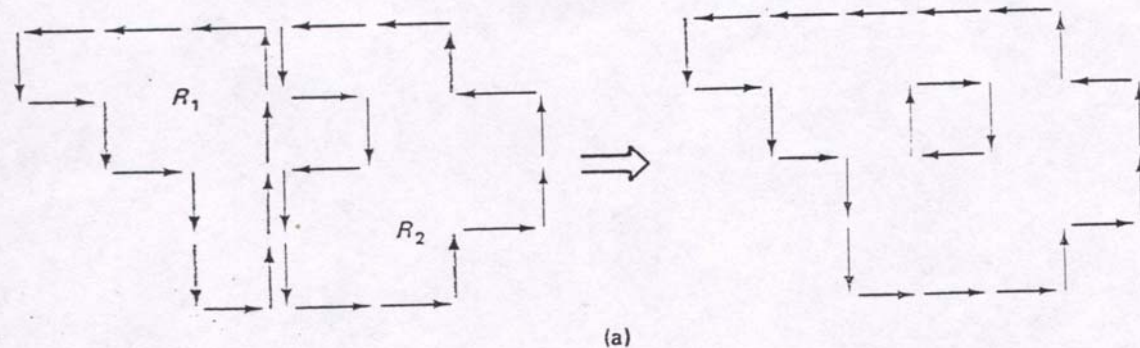


Fig. 5.7 Region operations on the grid structure of Fig. 5.6.

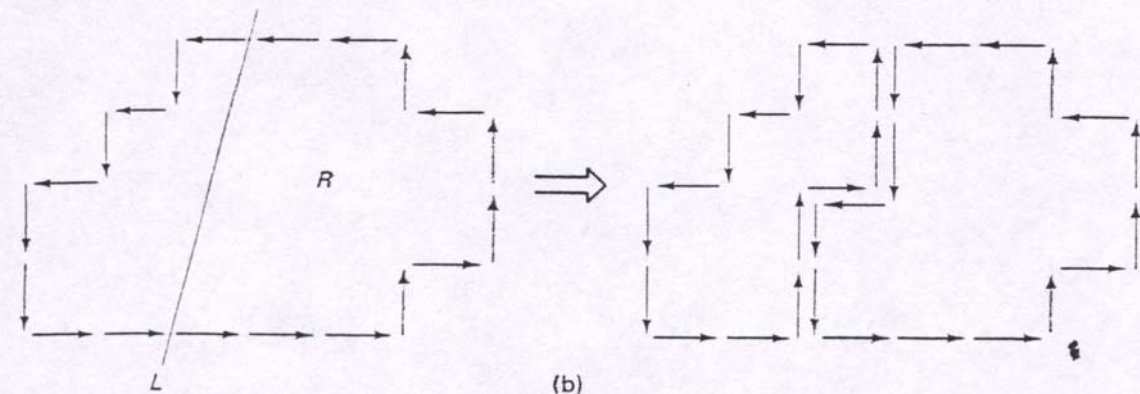


Fig. 5.7 (cont.)

where x_i and x_j are assumed to be on either side of a crack edge (Chapter 3). The three criteria are applied sequentially in the following algorithm:

Splitting and merging

Algorithm 5.4: Region Growing via Boundary Melting (T_k , $k = 1, 2, 3$ are reset thresholds)

For all neighboring pairs of points, remove the boundary between x_i and x_j if $i \neq j$ and $w_{ij} = 1$. When no more boundaries can be removed, go to step 2.

Remove the boundary between R_i and R_j if

$$\frac{W}{\min [p_i, p_j]} \geq T_2 \quad (5.11)$$

where W is the sum of the w_{ij} on the common boundary between R_i and R_j , that have perimeters p_i and p_j respectively. When no more boundaries can be removed, go to step 3.

Remove the boundary between R_i and R_j if

$$W \geq T_3 \quad (5.12)$$

Rappresentazione delle regioni



- Tessitura
- Scheletro

Tessitura

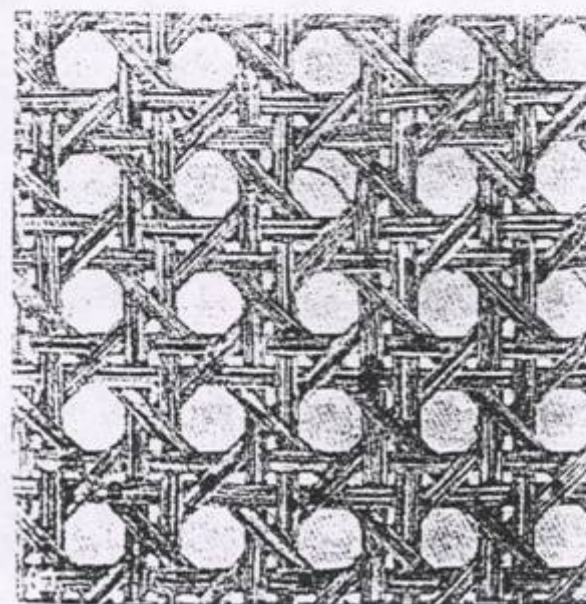
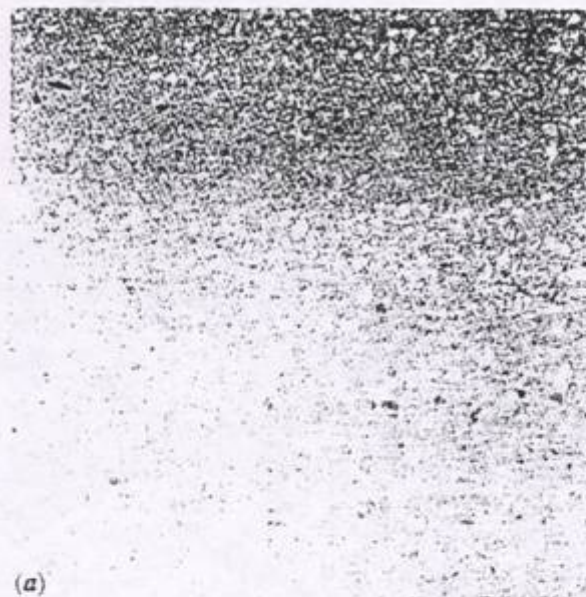
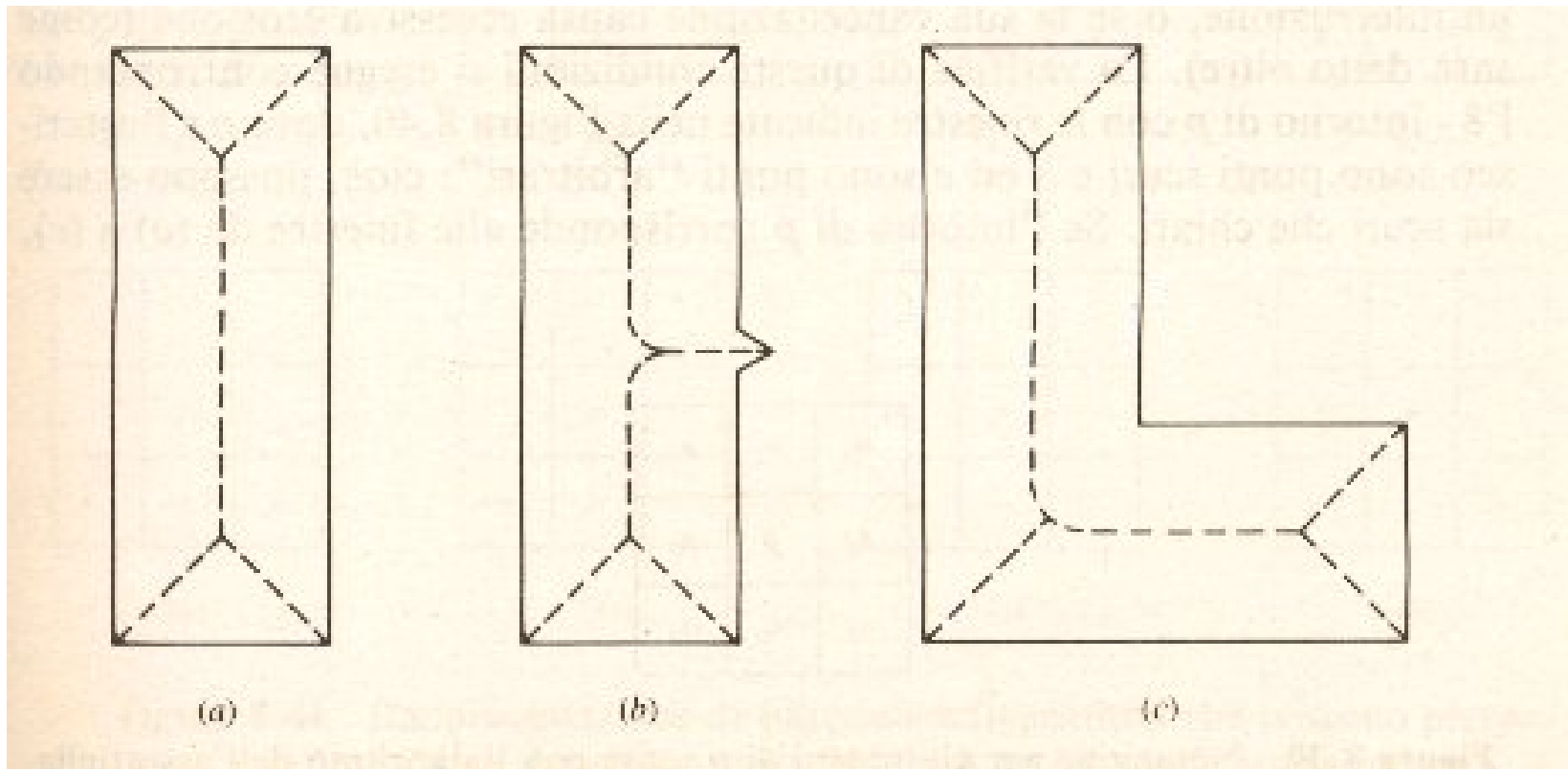


Figura 8.36 Esempi di tessitura (a) liscia, (b) ruvida e (c) regolare.

Rappresentazione delle regioni

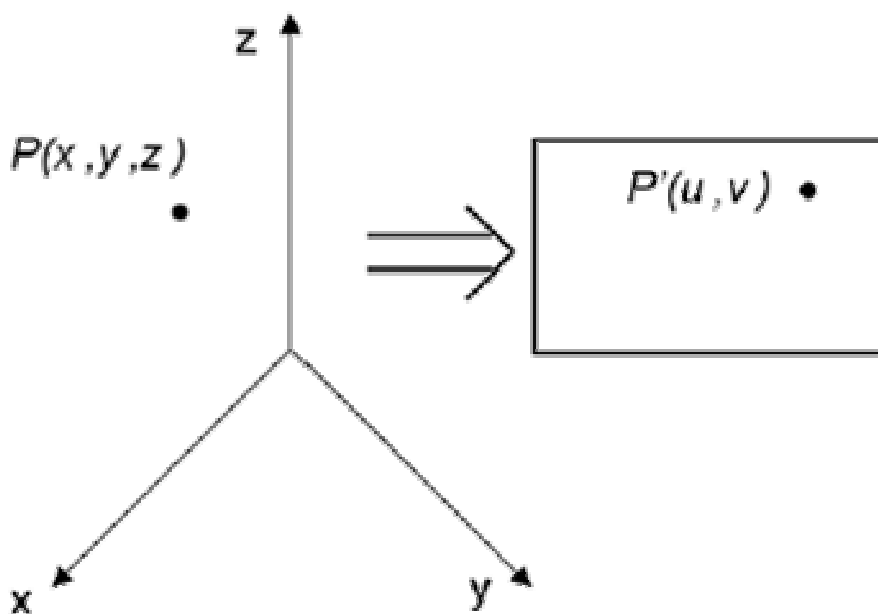
- Scheletro



Identificazione degli ASSI MEDIANI, come luogo dei punti a maggiore distanza dal contorno

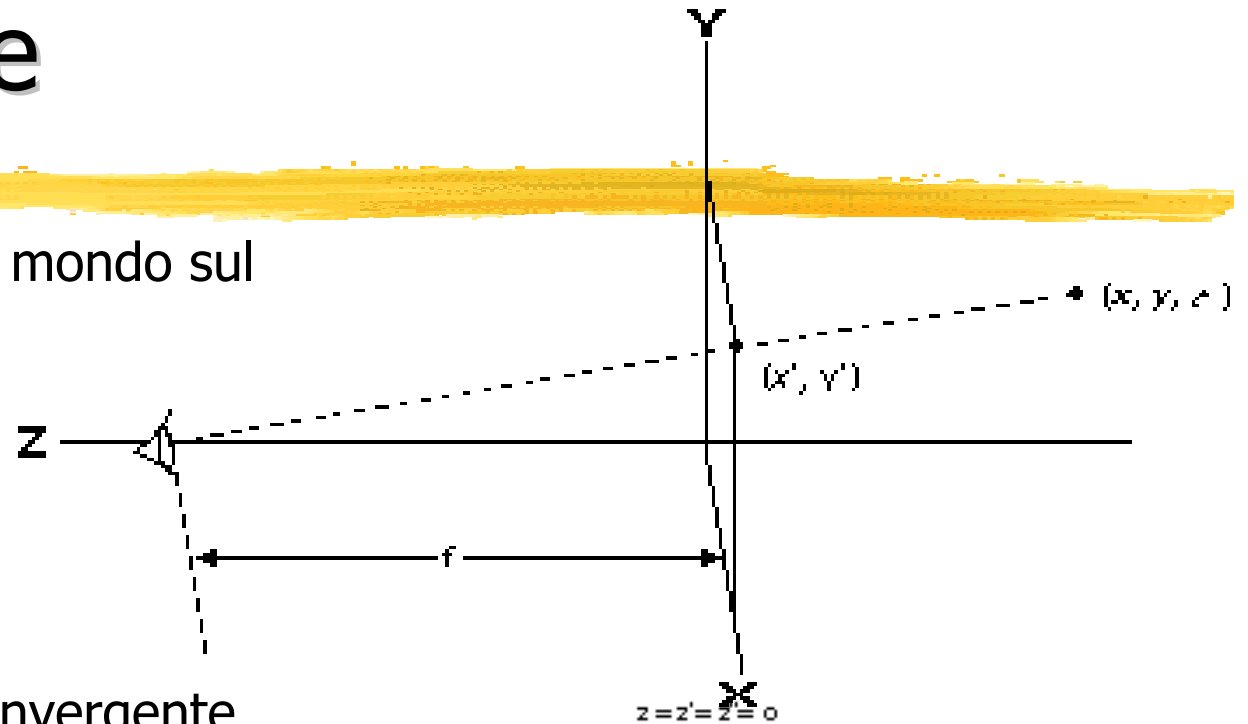
Problema fondamentale della visione stereoscopica

- trasformazione inversa, dalla proiezione 2D sul piano di un'immagine, ai punti corrispondenti nello spazio 3D
 - La proiezione operata da una telecamera comporta un'inevitabile perdita d'informazione: infatti il passaggio da un punto dello spazio tridimensionale ad un punto dell'immagine bidimensionale, prevede la perdita di una coordinata, quella che tiene conto della profondità
 - in un ipotetico passaggio inverso si ha il problema di recuperare l'informazione relativa a questa coordinata "smarrita"
 - Questo problema viene superato da una tecnica nota come *stereopsi*.



Visione stereoscopica principi di base

Proiezione di un punto del mondo sul
piano dell'immagine



Sistema binoculare non convergente

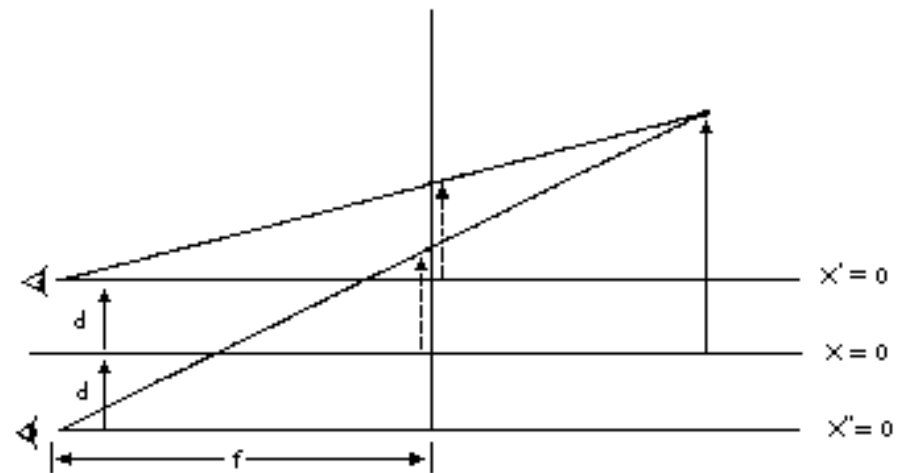
$$x_1 = \frac{(x-d)f}{f-z} \quad (f-z)x_1 = (x-d)f$$

$$x_2 = \frac{(x+d)f}{f-z} \quad (f-z)x_2 = (x+d)f$$

$$(f-z)(x_2 - x_1) = 2df$$

$$z = f - \frac{2df}{x_2 - x_1}$$

Disparità



[Tratta da: Ballard e Brown, 1982]

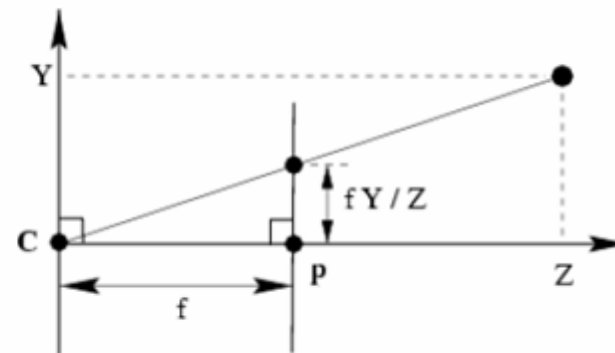
Visione stereoscopica



Passi fondamentali:

- Acquisire due immagini distinte, conoscendo la distanza tra le telecamere
- Ricavare le linee su cui i punti tridimensionali reali giacciono
- Intersecare le linee

Spazio 3D e piano immagine



Un punto nello spazio $\mathbf{X}=(x,y,z)$ è mappato in un punto sul piano dell'immagine corrispondente all'intersezione della retta passante per il punto \mathbf{X} ed il centro di proiezione con il piano dell'immagine.

$$(X, Y, Z)^T \text{ a } (fX/Z, fY/Z)$$

Ricostruzione 3D

$$x_1 = \frac{(x-d)f}{f-z} \quad (f-z)x_1 = (x-d)f$$

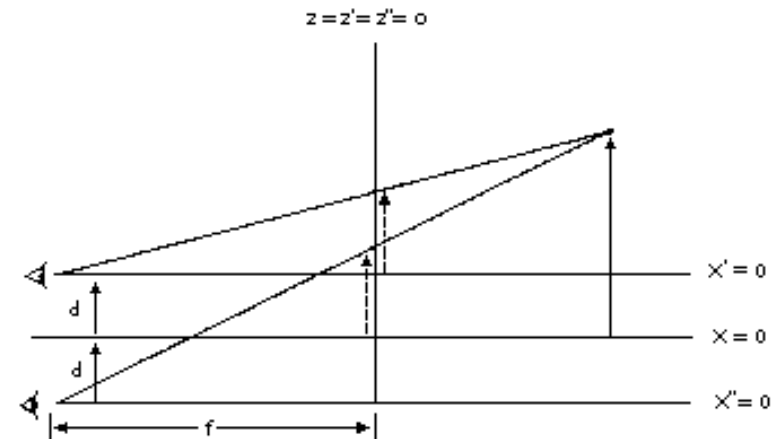
$$x_2 = \frac{(x+d)f}{f-z} \quad (f-z)x_2 = (x+d)f$$

$$(f-z)(x_2 - x_1) = 2df$$

Disparità

$$z = f - \frac{2df}{x_2 - x_1}$$

Sistema binoculare non convergente

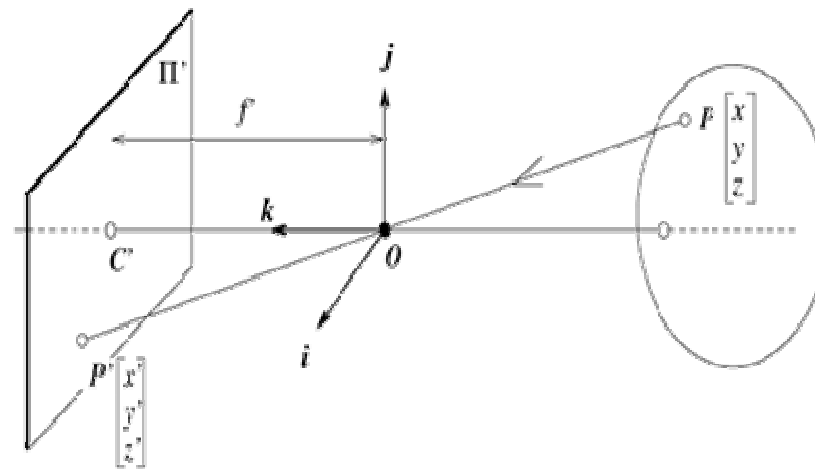


- la calibrazione è il punto di partenza obbligato, e quanto più preciso sarà il processo che ricava i parametri della telecamera, quanto più robusto ed efficiente sarà il funzionamento del sistema finale.

La calibrazione della telecamera

Tre diversi sistemi di riferimento:

1. il sistema di riferimento 3d detto anche *sistema mondo*;
2. il sistema di riferimento standard 3d della telecamera, centrato in C ;
3. il sistema di riferimento 2d per l'immagine.



- $X = (X; Y; Z)$ coordinate del punto P nel sistema mondo;
- $XC = (XC; YC; ZC)$ coordinate del punto P nel sistema della camera;
- $x = (x; y)$ coordinate del punto P nel sistema dell'immagine;
- $w = (u; v)$ coordinate del punto P nel sistema dell'immagine digitale.

Passi fondamentali



- Identificare la trasformazione:
Sistema mondo \Rightarrow sistema della telecamera
(rototraslazione)
- Identificare la trasformazione:
Sistema di della telecamera \Rightarrow sistema
dell'immagine (eq. prospettica)
- Identificare la trasformazione:
Coordinate dell'immagine $(x; y) \Rightarrow$ coordinate
in pixel $(u; v)$

Analiticamente

- **Passo 1** $\hat{\mathbf{X}}_C = \mathbf{P}_R \hat{\mathbf{X}}$
$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$
- **Passo 2** $\hat{\mathbf{x}} = \mathbf{P}_P \hat{\mathbf{X}}_C$
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{-f}{Z_C} & 0 & 0 & 0 \\ 0 & \frac{-f}{Z_C} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix}$$
- **Passo 3** $\hat{\mathbf{w}} = \mathbf{P}_C \hat{\mathbf{x}}$
$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} sx \\ sy \\ s \end{bmatrix}$$

- $\mathbf{X} = (X; Y; Z)$ coordinate del punto P nel sistema mondo;
- $\mathbf{X}_C = (X_C; Y_C; Z_C)$ coordinate del punto P nel sistema della camera;
- $\mathbf{x} = (x; y)$ coordinate del punto P nel sistema dell'immagine;
- $\mathbf{w} = (u; v)$ coordinate del punto P nel sistema dell'immagine digitale.

$$\hat{\mathbf{w}} = \mathbf{P}_C \mathbf{P}_P \mathbf{P}_R \hat{\mathbf{X}}$$

In pratica

$$\hat{\mathbf{w}} = \mathbf{P}_C \mathbf{P}_P \mathbf{P}_R \hat{\mathbf{X}}$$

Fattorizzazione $\mathbf{P}_C \mathbf{P}_P$

$$\hat{\mathbf{w}} = \mathbf{P}_{CP} \mathbf{P}_R \hat{\mathbf{X}} = \begin{bmatrix} -fk_u & 0 & u_0 & 0 \\ 0 & -fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \hat{\mathbf{X}}$$

Pongo inoltre

$$\mathbf{P}_{CP} = \mathbf{K} [\mathbf{I} \mid \mathbf{0}] \quad \text{con } \mathbf{K} = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{P}_R = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \quad \text{con } \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \text{e } \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

Allora risulta

$$\hat{\mathbf{w}} = \mathbf{K} [\mathbf{I} \mid \mathbf{0}] \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \hat{\mathbf{X}} = \mathbf{K} [\mathbf{R} \mid \mathbf{t}] \hat{\mathbf{X}} = \mathbf{P} \hat{\mathbf{X}}$$

\mathbf{P} è detta

MATRICE di PROIEZIONE PROSPETTICA

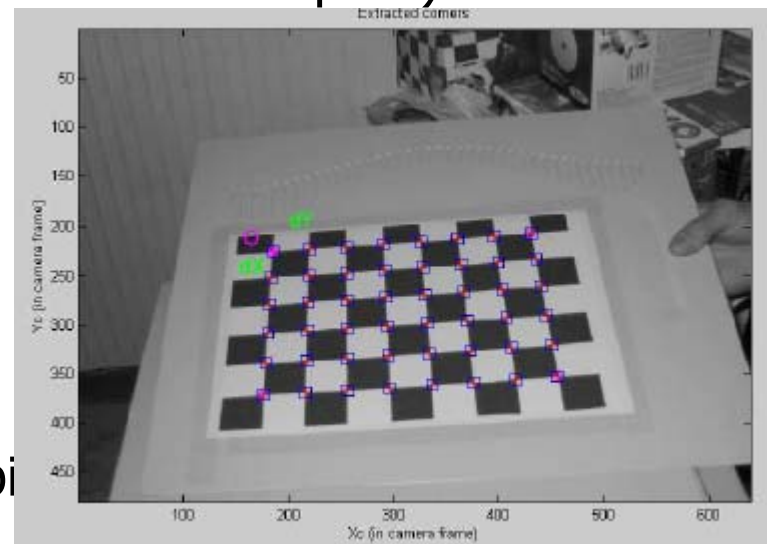
Matrice di Proiezione Prospettica (MPP)

- *MPP* rappresenta il modello geometrico della telecamera.
- E' una matrice 3x4, ma dipende da 11 parametri indipendenti.
- La matrice $[R|t]$ incorpora sia la rotazione che la traslazione del sistema mondo rispetto a quello della telecamera. Si dice che tale matrice contiene i **parametri estrinseci** della telecamera.
- Nella matrice K si può porre $a_u = -f k_u$ e $a_v = -f k_v$. Si tratta della **lunghezza focale espressa in pixel** orizzontali e verticali.
- Queste due grandezze, assieme a u_0 e v_0 rappresentano i **parametri intrinseci** della telecamera.
- Questi due insiemi di parametri caratterizzano completamente la telecamera.

La calibrazione è quel processo che permette di determinare le matrici dei parametri intrinseci ed estrinseci di una telecamera.

Calibrazione pratica

- E' necessario munirsi di un oggetto di calibrazione (geometria nota)
- una buona scelta è rappresentata da una scacchiera.
 - quadrati di lato noto
 - colore alternato (estrazione dei contorni semplice)
- In teoria, dato che *MPP* dipende da 11 parametri, sarebbero necessari 6 soli punti per poter calibrare la telecamera,
 - nella pratica ne sono disponibili molti di più.



Calibrazione pratica

1. Ottenere le coordinate dei corner nell'immagine digitale

2. Creare una tabella di corrispondenze

Punto	Sistema mondo	Sistema immagine
1	(X_1, Y_1, Z_1)	(u_1, v_1)
2	(X_2, Y_2, Z_2)	(u_2, v_2)
...
n	(X_n, Y_n, Z_n)	(u_n, v_n)

3. Si sfrutta la relazione $\hat{\mathbf{w}}_i = \mathbf{P} \hat{\mathbf{X}}_i = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \hat{\mathbf{X}}_i$ da cui

$$u_i = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \quad \text{e} \quad v_i = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}}$$

4. Per ogni punto della tabella si scrive una coppia di equazioni.

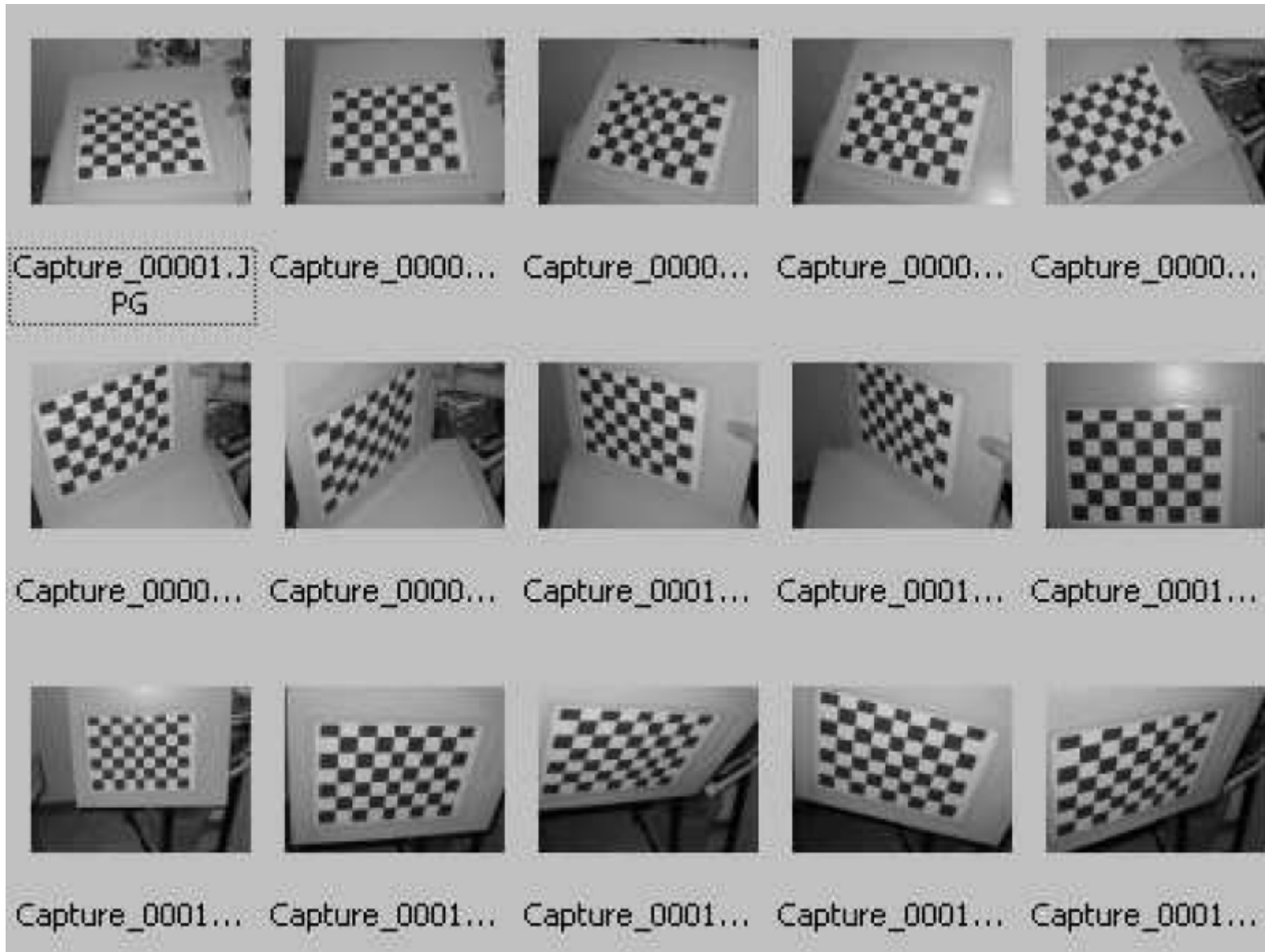
5. Dato che i punti sono in numero maggiore di 6 ci si trova a dover risolvere un sistema lineare sovradeterminato, (le equazioni sono in numero maggiore delle incognite).

6. La soluzione può essere ricavata utilizzando il metodo dei *minimi quadrati*.

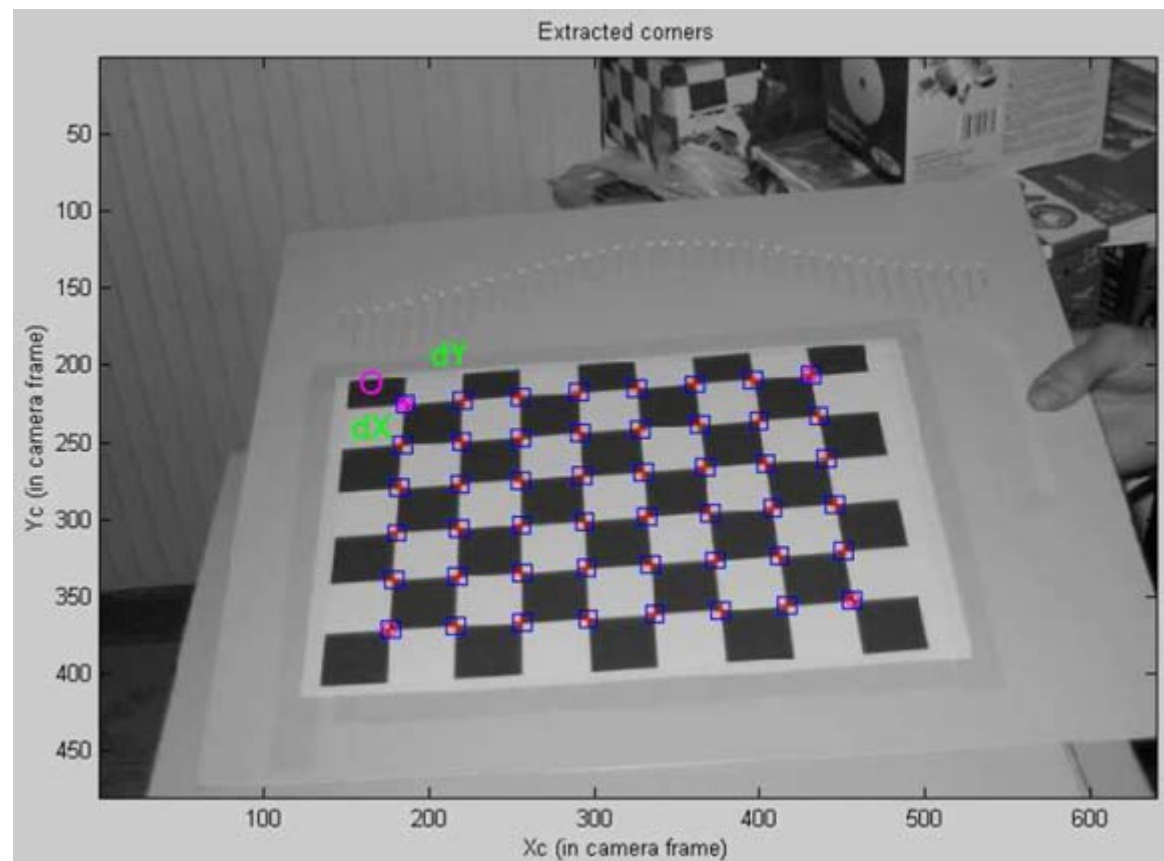
7. Metodi iterativi per il raffinamento della soluzione $\min_P \sum ((u_i - \bar{u}_i)^2, (v_i - \bar{v}_i)^2)$

8. Decomposizione QR della matrice (intrinseci ed estrinseci)

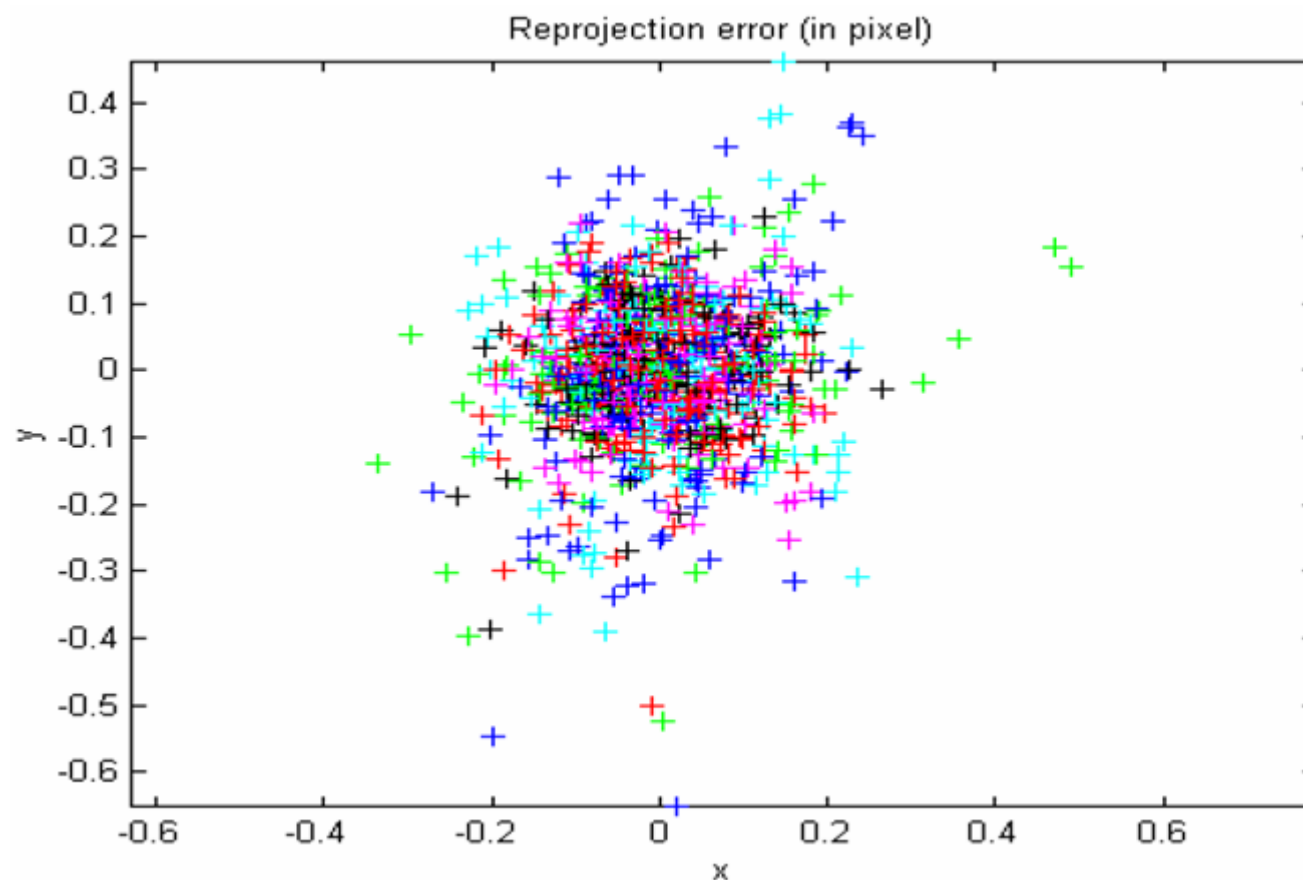
Sequenza immagini - Robustezza



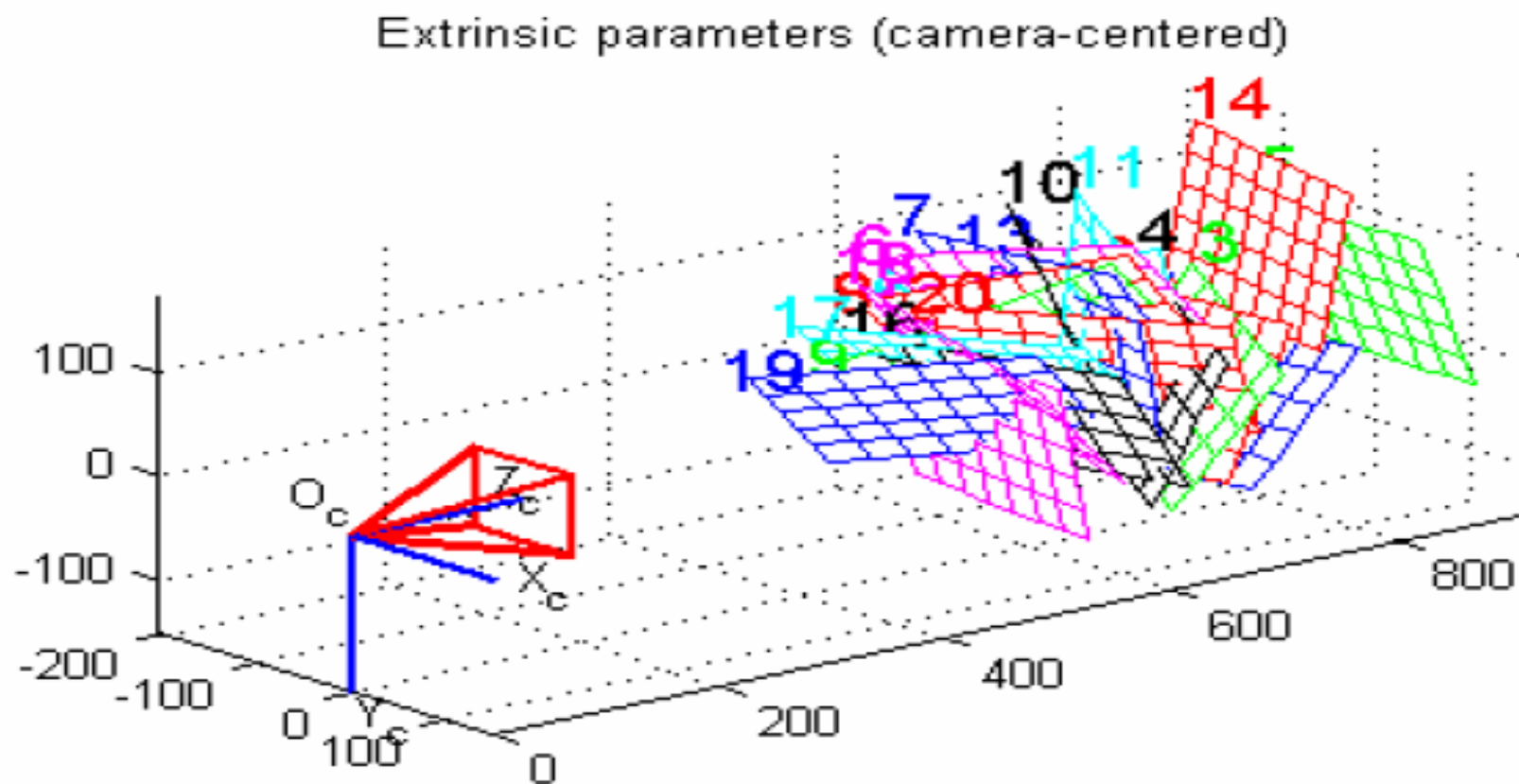
Estrazione contorni



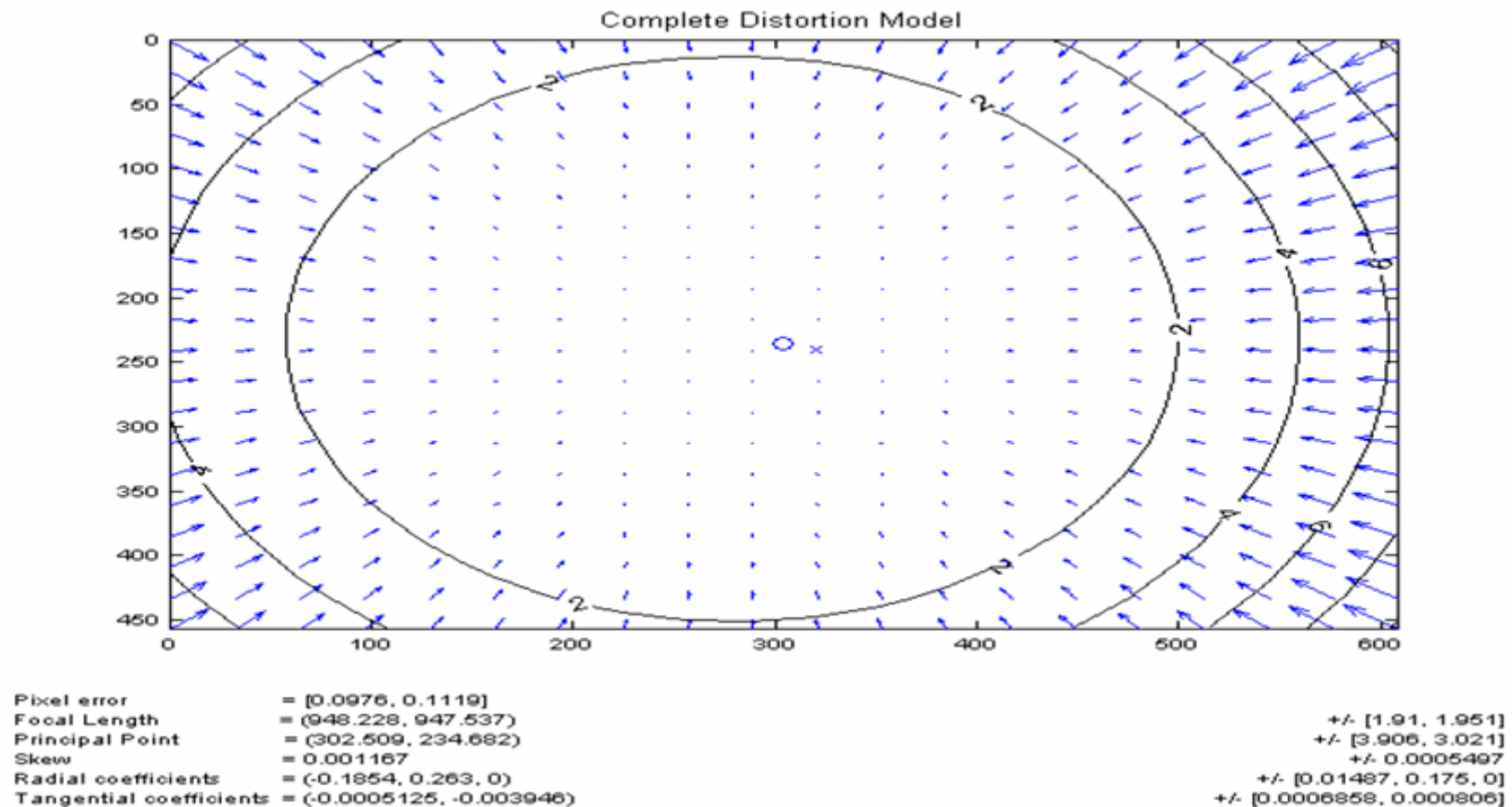
Errore di riproiezione



Parametri estrinseci



Stima distorsione

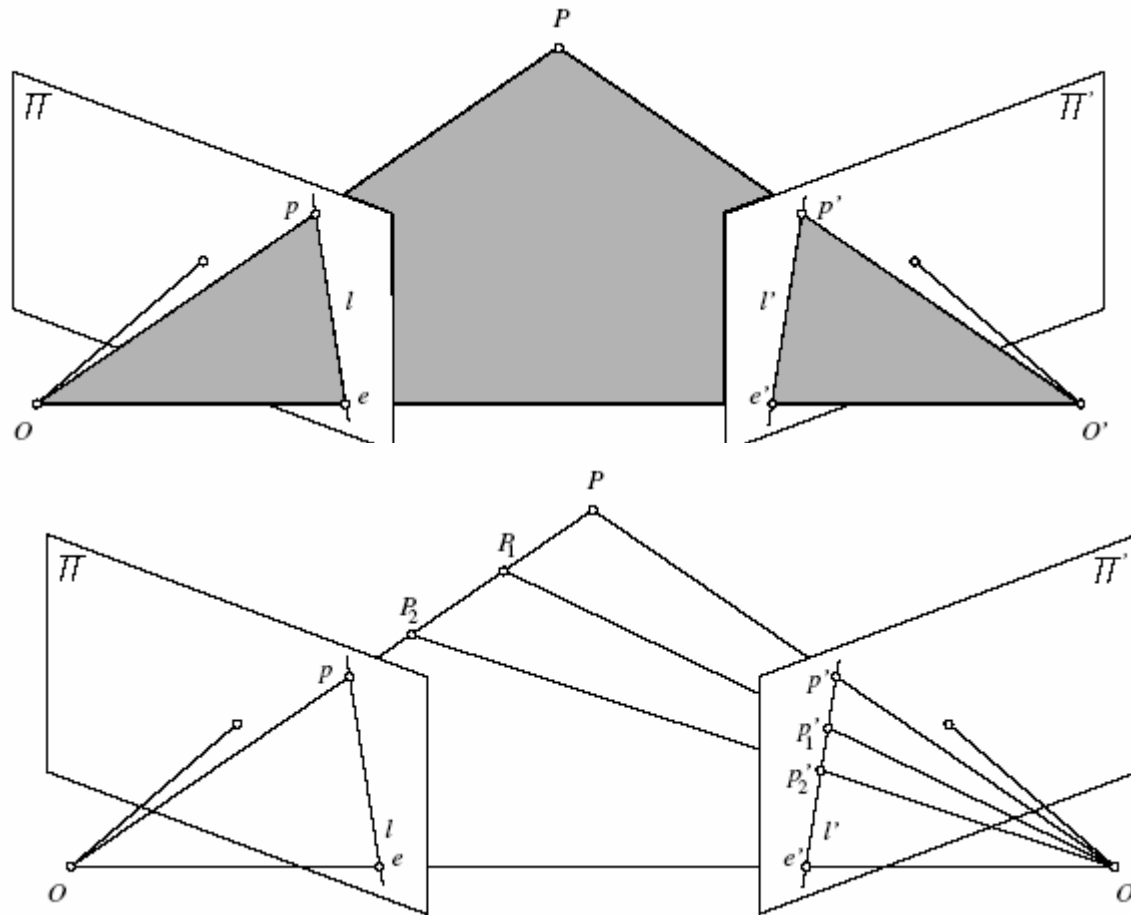


2 Telecamere: problema del matching

- Date due telecamere trovare la corrispondenza di punti omologhi su due diverse immagini

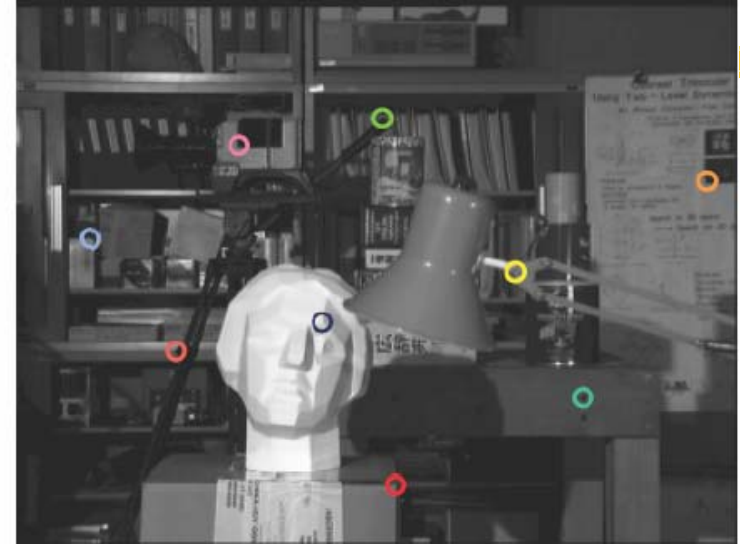


2 Telecamere – Geometria epipolare



Vincolo epipolare: dato un sistema stereo calibrato l'insieme dei punti di matching di p è vincolato a giacere sulla linea epipolare l'

Mappa delle disparità



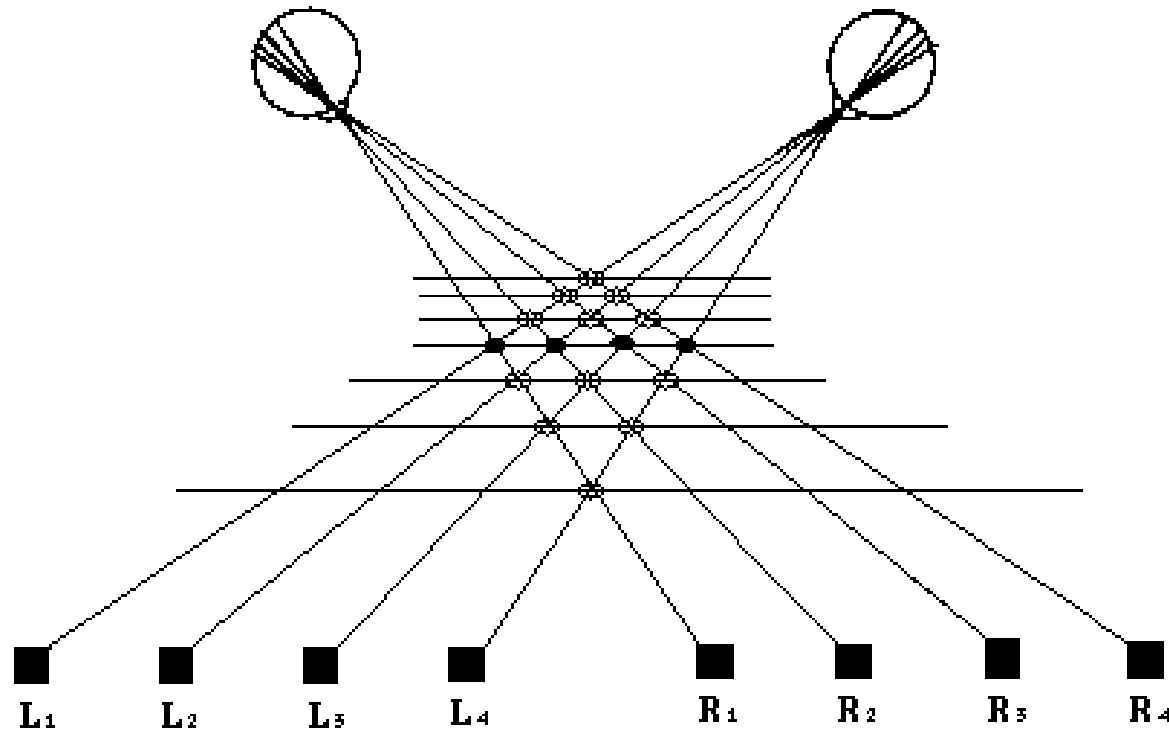
Ricostruzione stereo



1. Calibrare la 1 telecamera
2. Calibrare la II telecamera
3. Calcolare la rototraslazione tra i sistemi di riferimento delle 2 telecamere
4. Determinare le corrispondenze tra punti (vincolo epipolare)
5. Costruire una mappa delle disparità
6. Ricostruire la posizione 3D dei punti di interesse usando la disparità e i parametri delle telecamere

Algoritmo di Marr & Poggio

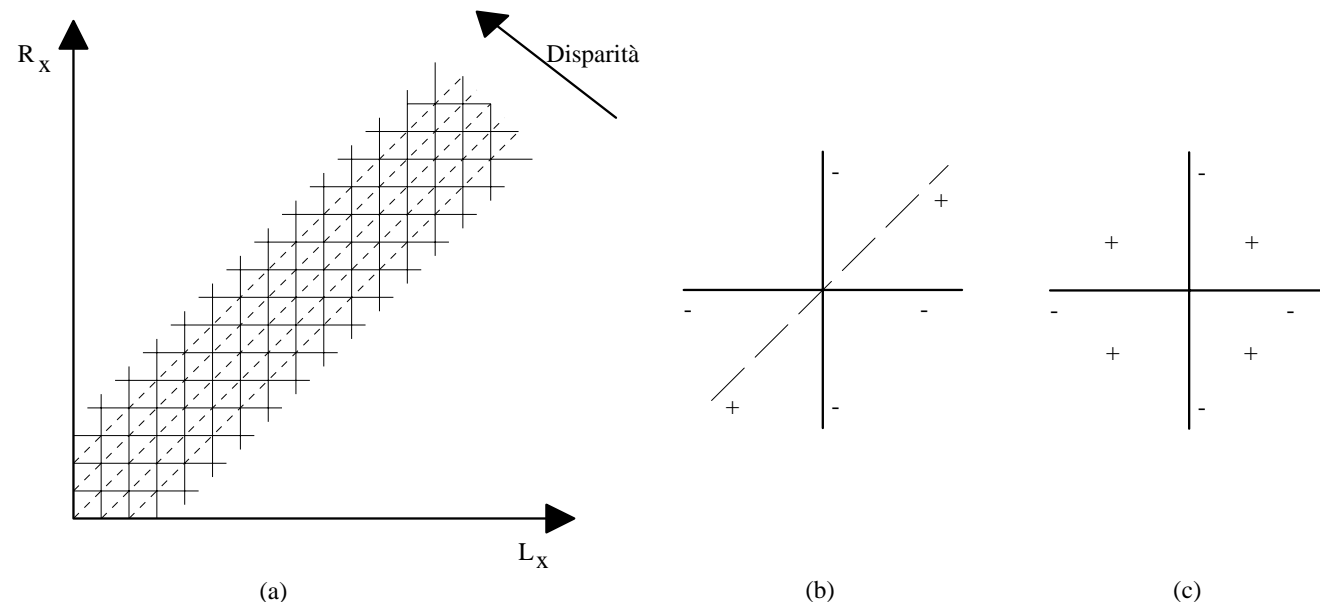
- Selezionare un punto su una superficie della scena in una delle due immagini
- Identificare lo stesso punto nell'altra immagine
- Misurare la disparità tra le due posizioni del punto



Algoritmo di Marr & Poggio

Vincoli:

- Compatibilità: i punti neri possono corrispondere solo a punti neri
- Unicità: quasi sempre un punto nero di un'immagine può corrispondere a non più di un punto nero dell'altra immagine
- Continuità: la disparità varia linearmente quasi ovunque nell'immagine, eccetto ai bordi



In (a) L_x e R_x rappresentano le posizioni degli elementi nelle immagini sinistra e destra rispettivamente. Le linee continue orizzontali e verticali rappresentano le linee di visione dell'occhio sinistro e destro. Le intersezioni di queste linee corrispondono a possibili valori di disparità. Le linee tratteggiate diagonali sono linee di disparità costante. Le linee tratteggiate rappresentano le interazioni eccitatorie, mentre le altre (orizzontali e verticali) quelle inibitorie. (b) mostra la struttura locale della rete ad ogni nodo. (c) mostra la struttura locale della rete corrispondente applicata ad immagini bidimensionali. [Tratta da: Marr, 1982]

Algoritmo di Marr & Poggio

$$C_{x,y,d}^{t+1} = \sigma \left\{ \sum_{x',y',d' \in S(x,y,d)} C_{x',y',d'}^t - \varepsilon \sum_{x',y',d' \in O(x,y,d)} C_{x',y',d'}^t + C_{x,y,d}^0 \right\}$$

: stato della cella (x,y) con disparità d al tempo t

$S(x,y,z)$: intorno locale eccitatorio

$O(x,y,z)$: intorno locale inibitorio

ε : costante di inibizione

σ : funzione di soglia