

Elaborato di Robotica

La VISIONE ROBOTICA



Redatto da:

Zanotti	Daniele	Mat: 024970	E-mail: danizan@libero.it
Bettinsoli	Stefano	Mat: 022357	E-mail: bettinsoli@libero.it

ACQUISIZIONE IMMAGINI

Com'è possibile che un computer “veda” o meglio “acquisisca” delle immagini?

Oggigiorno esistono tre famiglie di dispositivi per l'acquisizione d'immagini dal mondo reale; essi sono:

1. Scanner
2. Telecamere digitali
3. Fotocamere digitali

Tutti questi strumenti hanno un particolare in comune: l'acquisizione dell'immagine avviene attraverso un dispositivo elettronico di silicio in grado di tradurre segnali elettromagnetici (la luce) emessi dagli oggetti in segnali elettrici facilmente manipolabili, esattamente come l'occhio umano fa attraverso la retina. Questi piccoli sensori di silicio sono chiamati CCD (charge coupled device); essi variano in base alle loro caratteristiche.

Di una periferica per l'acquisizione d'immagini bisogna tener conto di vari parametri:

1. Risoluzione ottica: ciò che in realtà il sensore riesce a vedere; si misura in dpi (dot per inch), punti per pollice che riesce a distinguere e ciò rappresenta la digitalizzazione dell'immagine. Doverosa è una precisazione: esiste anche una risoluzione di tipo interpolato, ma è una risoluzione che si raggiunge attraverso delle funzioni matematiche e i punti aggiunti all'immagine digitalizzata non appartengono realmente all'immagine originaria.
2. Profondità di colore: questa caratteristica dipende dalla qualità del CCD, maggiore è la profondità di colore migliore sarà la capacità di distinzione tra sfumature di uno stesso colore quindi maggiore la precisione del valore numerico assegnatogli. Il numero di bit che viene usato per rappresentare il colore di un pixel è detto *Profondità*. Per esempio se il numero di bit è 16, il CCD sarà in grado di distinguere 2^{16} colori.
3. Per telecamere e fotocamere è importante sapere il numero di pixel che il sensore ottico fisicamente ha. Questo perché per quanto riguarda l'acquisizione d'immagini attraverso uno scanner, il sensore scansiona l'immagine un poco per volta fino a che la scansione non è completa. Invece per gli altri due dispositivi si deve avere un sensore in grado di catturare in maniera istantanea e alla risoluzione voluta l'intera l'immagine. Per informazione, oggi esistono fotocamere con CCD che arrivano fino a 3.3 milioni di pixel per una risoluzione massima di 2100*1710.

Quando un'immagine viene acquisita da un dispositivo digitale le linee non sono più linee continue ma la loro continuità dipende dalla risoluzione alla quale il dispositivo sta lavorando.

Una linea geometrica precisa è convertita in un insieme di pixel quindi bisogna decidere quanti e quali pixel colorare.

A parità di immagine da acquisire, una frequenza di campionamento elevata genera un elevato numero di pixel, mentre una frequenza di campionamento ridotta ne genera un numero minore .

Nella visione naturale tipicamente umana è preferibile avere maggiore profondità di colore, nella visione robotica, invece, è meglio adottare delle acquisizioni che diano un numero maggiore di pixel.

In generale più un'immagine acquisita è ricca di informazioni migliore sarà il risultato degli algoritmi che ne ricercano le caratteristiche (come l'*edge detection*).

IMMAGINE COME RAPPRESENTAZIONE DEL MONDO

Nella robotica classica si parla di visione quando si utilizzano delle immagini digitalizzate per acquisire informazioni sull'ambiente esterno.

La "visione robotica" raggruppa le varie metodologie usate per elaborare tali informazioni.

In questo ambiente le immagini sono una *rappresentazione approssimata della realtà* (campionamento e quantizzazione).

Un'immagine acquisita è rappresentata in memoria come un insieme organizzato di pixel raggruppati secondo una *retinatura*.

Esistono diversi tipi di retinatura ed in figura sono rappresentati i più diffusi.

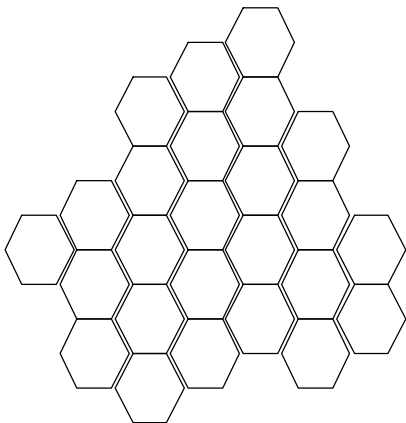


Figura 1: Retinatura a nido d'ape

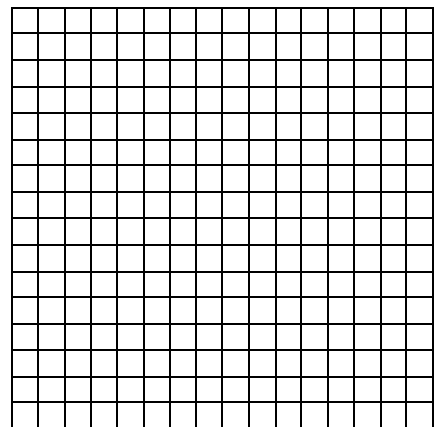


Figura 2: Retinatura a griglia quadrata

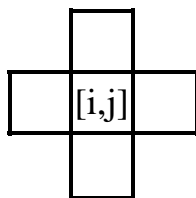
COMPUTAZIONE DELLE IMMAGINI

Bisogna dare alcune definizioni

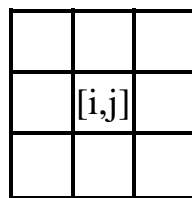
- *Vicinato:*

un'immagine è formata da una griglia di pixel, se uno di questi si trova nella posizione $[i,j]$ viene definito *vicinato* N4 l'insieme di pixel che si trovano rispettivamente nella posizione $[i+1,j]$, $[i-1,j]$, $[i,j+1]$, $[i,j-1]$;

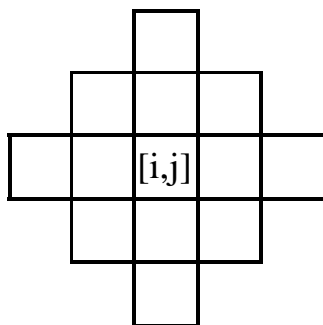
Viene definito vicinato N8 l'insieme di pixel dell'N4 più quelli nelle posizioni $[i+1,j+1]$, $[i+1,j-1]$, $[i-1,j+1]$, $[i-1,j-1]$. Continuando in questo modo si possono definire i vicinati N12, N20, N24, ecc



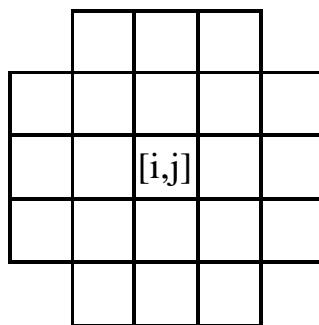
Vicinato di tipo N4



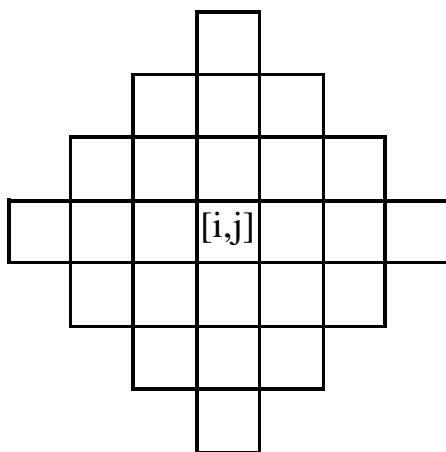
Vicinato di tipo N8



Vicinato di tipo N12



Vicinato di tipo N20



Vicinato di tipo N24

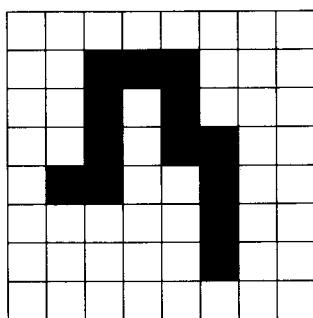
- *Connessione o connettività:*

Un pixel $p \in S$ si dice connesso a $q \in S$ se vi è un path da p a q costituito tutto da pixel $\in S$.

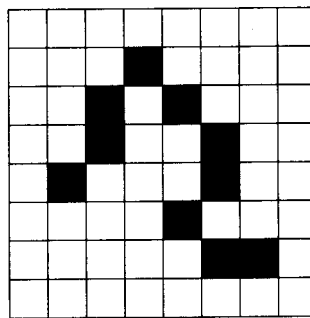
La connessione rappresenta una relazione con proprietà riflessiva, commutativa e transitiva

- *Path:*

Un path da un pixel $[i_0, j_0]$ ad un pixel $[i_n, j_n]$ è una sequenza di pixel $[i_0, j_0], [i_1, j_1], \dots, [i_n, j_n]$ tale che un pixel in $[i_k, j_k]$ risulta essere un “vicino” del pixel $[i_{k+1}, j_{k+1}]$ per ogni k compreso tra zero e $n-1$. Se il vicinato considerato è un N4 si ha un 4-path, per un N8 si ha un 8-path e così via.



(a) 4-path



(b) 8-path

- *Foreground:*

In un immagine in bianco e nero il “primo piano” è l’insieme dei pixel a cui è stato attribuito il valore 1, cioè quelli neri, questo insieme si denota con S .

- *Componenti connessi e background :*

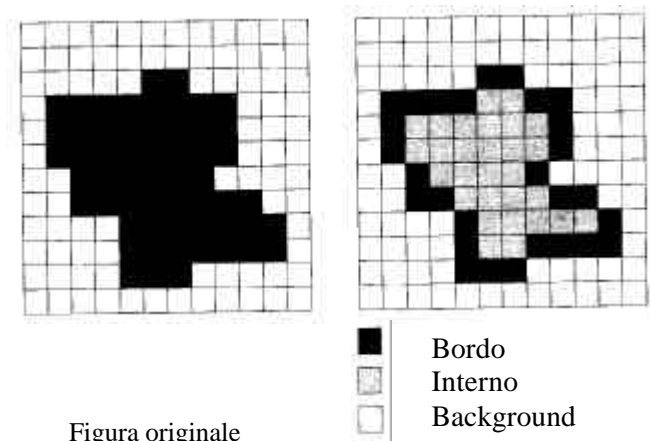
Considerato l’insieme S^* complementare di S , tutti i pixel connessi di S^* se sono adiacenti a S sono definiti background.

- *Bordo :*

Il bordo di S è l’insieme di pixel appartenenti a S e che hanno alcuni pixel del vicinato N4 che appartengono a S^* . Questo insieme viene definito con S' .

- *Interno :*

E’ l’insieme di pixel $\in S$ ma non a S' .



- **Size filter**

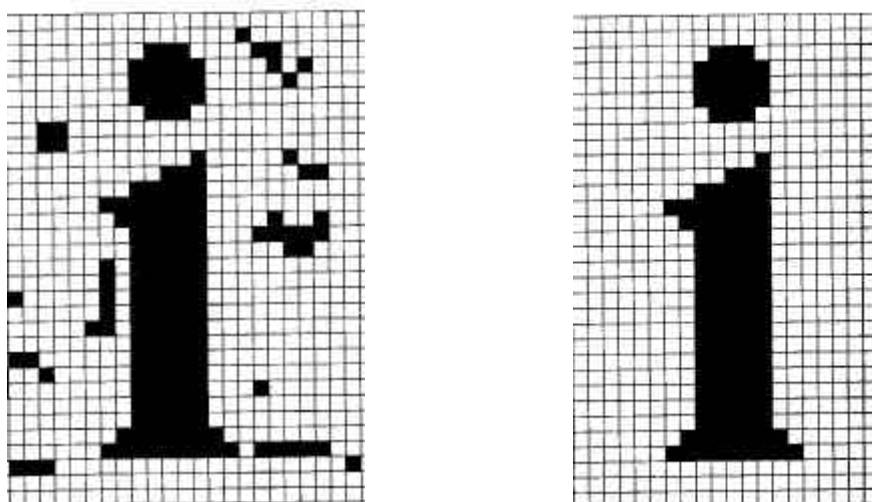
Per ricavare alcune informazioni da un'immagine a volte si utilizza un semplice algoritmo di "soglia".

Il *size filter* è appunto un esempio di questi algoritmi.

Spesso succede che durante l'acquisizione oltre all'immagine originaria viene acquisito anche del rumore. Questo rumore, rispetto all'immagine, aggiunge un numero casuale ma limitato di pixel e per rimuoverli è sufficiente far sì che l'algoritmo che elabora l'immagine ricerchi in quest'ultima delle figure contenenti un numero minore o uguale a k pixel dove k è una soglia da noi prefissata e una volta selezionate le elimini.

Chiaramente occorre fare molta attenzione quando si pone il valore di soglia perché può accadere che se viene fissato troppo alto vengano tolti particolari che in realtà appartenevano all'immagine originaria.

Nelle figure sotto riportate si può osservare come agisce il Size filter con k uguale a 12. Se avessimo posto k uguale a 22 avremmo perso anche il puntino della *i* poiché occupa 21 pixel !



Le zone che contengono un numero di pixel inferiore ad una certa soglia k vengono eliminate.

Naturalmente la soglia k può riguardare anche altri parametri quali il colore la luminosità ecc e non solo il numero di pixel contenuti in una regione chiusa.

- **Distanza**

In molte applicazioni è necessario conoscere la distanza tra due pixel o tra due figure in un'immagine. Tuttora non è stato definito un metodo unico per misurare tale distanza in un'immagine digitale. Quindi è possibile farlo in diversi modi. In ogni caso per ogni situazione vi è un metodo che risulta essere migliore rispetto agli altri.

Dati tre pixel p, q e r la distanza D gode di queste proprietà:

$$D(p, q) \geq 0 \quad \text{e} \quad D(p, q) = 0 \quad \text{se e solo se} \quad p = q$$

Cioè la distanza tra due punti è sempre una quantità positiva ed è nulla solo nel caso che i due punti abbiano coordinate equivalenti.

$$D(p, q) = D(q, p)$$

Cioè la distanza tra due punti non cambia invertendo il loro ordine.

$$D(p, r) \leq D(p, q) + D(q, r)$$

Cioè la distanza tra i punti p e r è minore od uguale alla somma delle distanze parziali che si possono calcolare introducendo un terzo punto q . Saranno uguali solo se il punto q giace sulla retta che passa per p e r .

Si hanno inoltre tre dei possibili modi per misurare la distanza :

a) Distanza euclidea

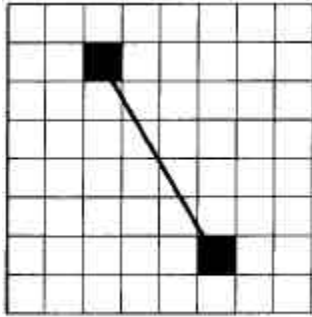
$$D([i_1, j_1], [i_2, j_2]) = \sqrt{(i_1 - i_2)^2 + (j_1 - j_2)^2}$$

b) Distanza di Manhattan

$$D([i_1, j_1], [i_2, j_2]) = |i_1 - i_2| + |j_1 - j_2|$$

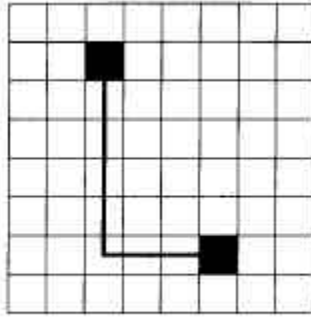
c) Distanza scacchiera

$$D([i_1, j_1], [i_2, j_2]) = \max(|i_1 - i_2|, |j_1 - j_2|)$$



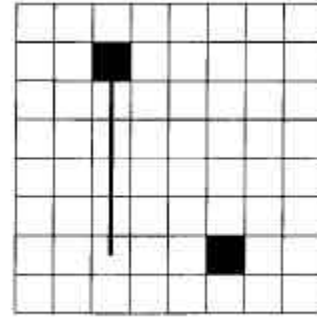
(a)

Distanza euclidea



(b)

Distanza di Manatthan

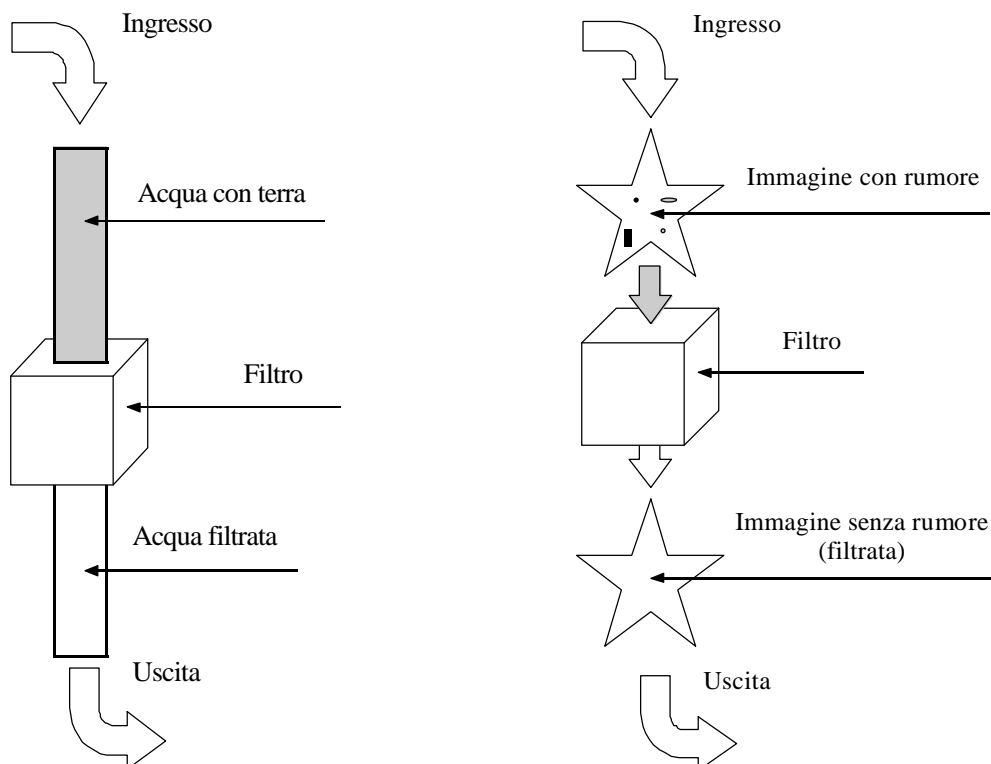


(c)

Distanza scacchiera

SISTEMI LINEARI E FILTRI

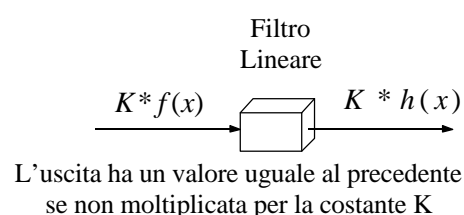
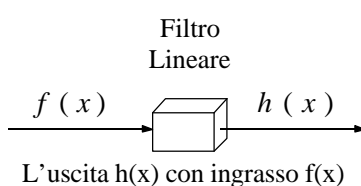
Un filtro è una ‘scatola’ che accetta qualcosa in ingresso per restituirne un'altra in uscita. L'operazione più comune compiuta da un filtro è l'eliminazione di una parte del contenuto dell'oggetto che ha ricevuto in ingresso. L'oggetto in ingresso può essere molte cose: acqua, luce, suoni e immagini digitali. La maggior parte delle volte il filtro cerca di eliminare delle impurità che si sono aggiunte all'oggetto che esamina: nel caso dell'acqua potrebbe essere della terra, nel caso di immagini digitali invece cercherà di eliminare del rumore aggiuntosi durante il processo di acquisizione.



Esempi di cosa fa un filtro

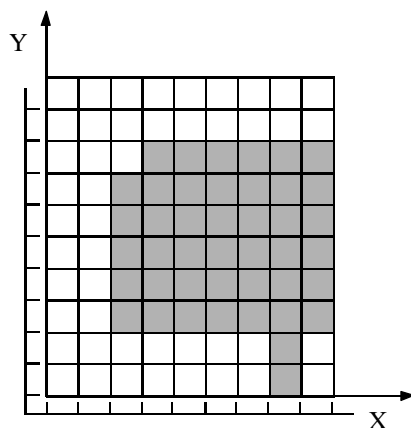
Un filtro può avere diverse caratteristiche e può essere:

- **Lineare:** analiticamente significa che se noi moltiplichiamo per una costante l'ingresso otteniamo un uscita che è anch'essa moltiplicata per la stessa costante, cioè:



Trattando di immagini significa che se si ingrandisce/diminuisce un immagine l'effetto del filtro non varia.

- Tempo invariante: significa che se il filtro viene applicato in tempi diversi a un oggetto il risultato che ne otteniamo è lo stesso, ovviamente in tempi diversi
- Spazio invariante: questa proprietà si traduce nel fatto che se noi applichiamo un filtro ad un immagine scandendola da punti diversi otteniamo sempre lo stesso risultato
- Tempo/spazio continuo/discreto: queste coppie di proprietà, che possono combinarsi le une con le altre indicano appunto che il filtro può agire su tempo o spazio dell'oggetto in esame e che questi ultimi possono essere continui o discreti. Le immagini digitali sono oggetti a spazio discreti, cioè nel piano x,y che le contiene (spazio) esiste una discretizzazione lungo gli assi cartesiani cioè questi ultimi vengono divisi in passi di dimensione uguale a quella dei pixel.



Esempio di discretizzazione spaziale

Ora senza perdere in generalità possiamo affermare che l'uscita di un filtro lineare a tempo continuo e spazio discreto che viene applicato ad un immagine da come risultato la seguente espressione:

$$h[i,j]=A \cdot p_1+B \cdot p_2+C \cdot p_3+D \cdot p_4+E \cdot p_5+F \cdot p_6+G \cdot p_7+H \cdot p_8+I \cdot p_9$$

dove $h[i,j]$ è il valore che il pixel assume dopo che il filtro gli è stato applicato. Nel caso specifico della visione possiamo immaginare i valori A, B, C, \dots, I come valori costanti di una matrice il cui centro, scorre pixel dopo pixel su tutta l'immagine, mentre i valori $p_1, p_2, p_3, \dots, p_9$ sono i valori dei pixel dell'immagine adiacenti al pixel filtrato.

Cosa molto importante è che ogni filtro ha una matrice che lo caratterizza e viceversa, cosicché è sufficiente variare i valori presenti in essa per cambiare il filtro adottato, non solo, essendo che la banda passante dipende dalla dimensione della matrice, possiamo aumentare quest'ultima ottenendo una banda maggiore, cioè invece che una matrice 3x3 si può usare una 4x4 o 5x5 e così via.

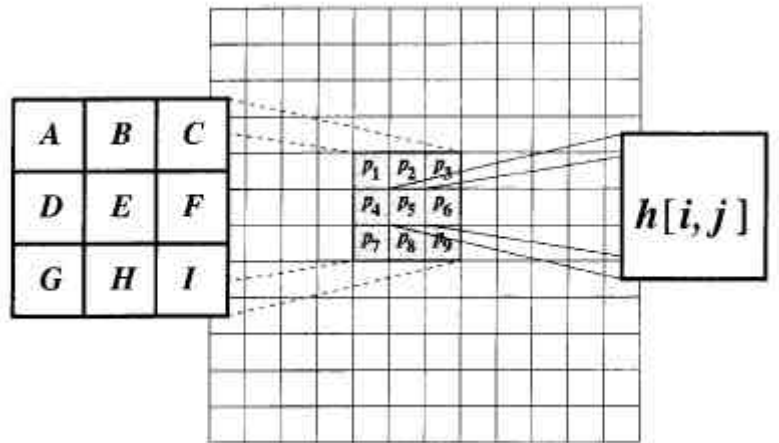


Immagine a cui si riferisce la formula sopra citata

FILTRI LINEARI

Si ha la necessità di filtrare le immagini perché spesso durante la loro acquisizione viene a sommarsi del rumore, oppure perché interessa evidenziarne alcune caratteristiche. I più classici e tipici casi di rumore sono quello granuloso, impulsivo e gaussiano. Il primo apporta granuli di pixel bianchi o neri, il secondo apporta solo valori impulsivi bianchi, mentre il terzo apporta una variazione gaussiana all'intensità di tutti i pixel che compongono l'immagine.



Figura originale di un particolare di un drago

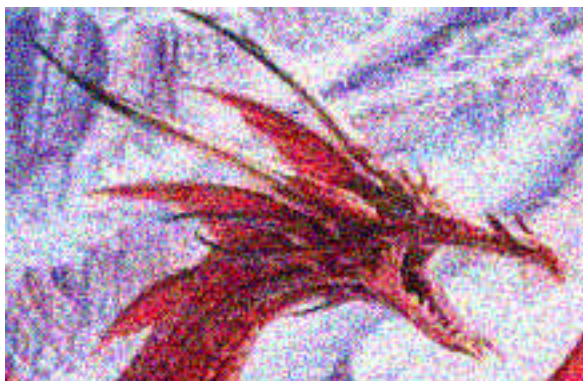


Figura con rumore di tipo granuloso

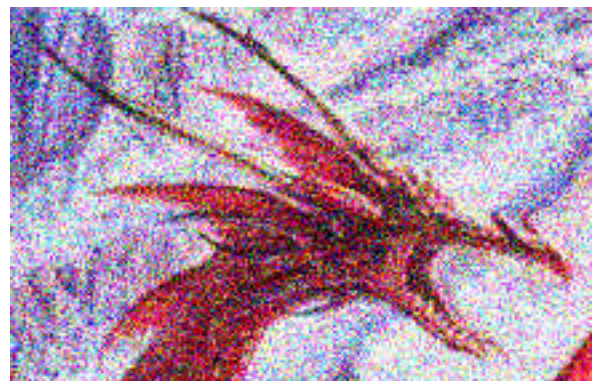


Figura con rumore di tipo gaussiano



Figura originale



Figura con rumore granuloso monocromatico



Figura con rumore gaussiano monocromatico

- **Smoothing filter**

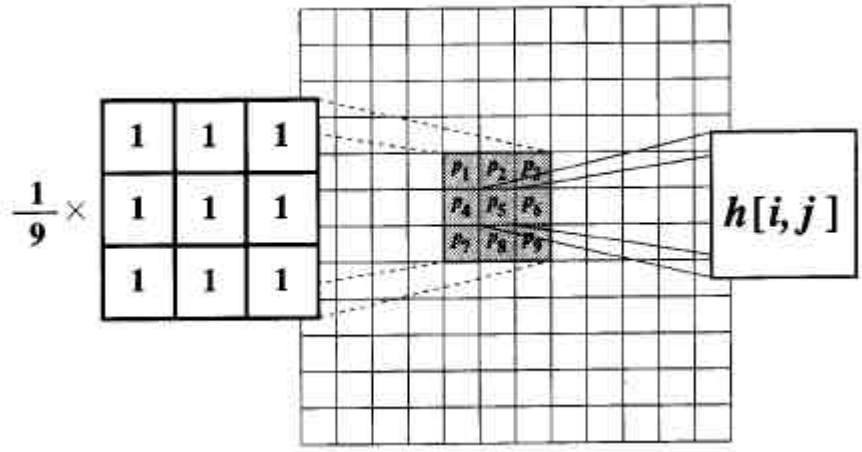
Questo filtro risulta essere molto efficace quando ci si trova di fronte alla presenza di rumore di tipo gaussiano.

Tale filtro può essere usato anche per gli altri tipi di rumore.

Un filtro lineare è implementato facendo la somma pesata dei pixel adiacenti (vedi figura) e questo vuol dire che è possibile adottare le maschere di convoluzione viste prima.

Bisogna precisare che se un filtro non esegue la somma pesata del valore dei pixel risulta essere un filtro non lineare.

Un esempio di filtro spazio invariante ma non lineare è il filtro mediano



Maschera di smoothing filter

- **Mean filter**

Questo filtro somma il valore dei pixel adiacenti a quello preso in esame per poi dividere per il totale dei pixel utilizzati.

La formula che lo definisce è :

$$h[i, j] = \frac{1}{M} \sum_{(k,l) \in N} f[k, l]$$

dove $h[i, j]$ rappresenta il nuovo valore del pixel mentre i valori $f[k, l]$ sono i valori dei pixel adiacenti a quello filtrato

Per esempio se avessimo una maschera 3x3

$$h[i, j] = \frac{1}{9} \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} f[k, l]$$

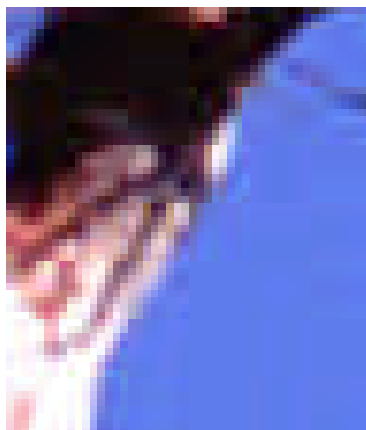


Immagine originale

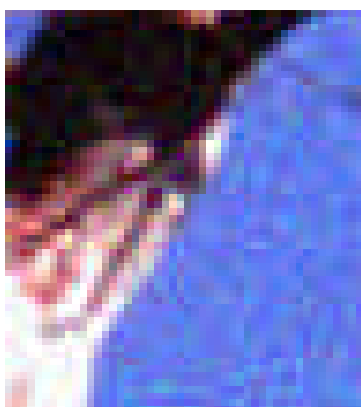


Immagine con dist. Gaussiano

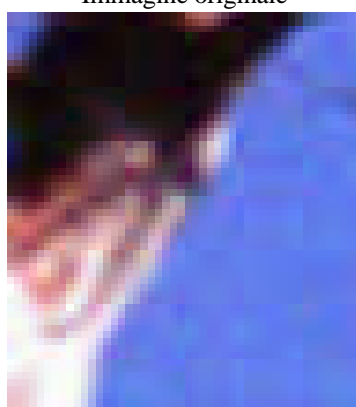


Immagine filtrata con *mean* 3X3

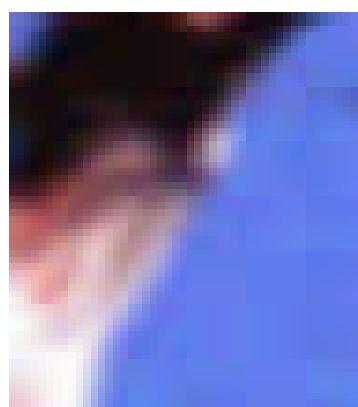


Immagine filtrata con *mean* 5X5



Immagine originale

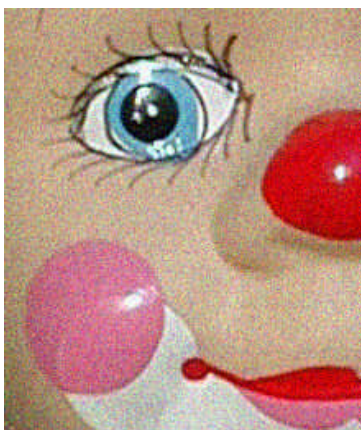


Immagine con dist. Gaussiano

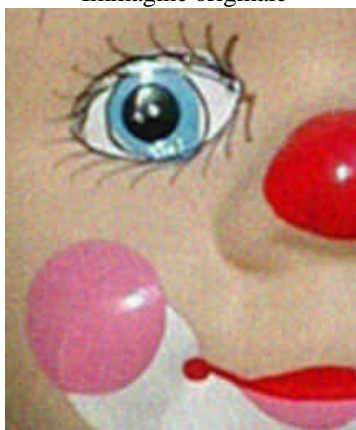


Immagine filtrata con *mean* 3X3

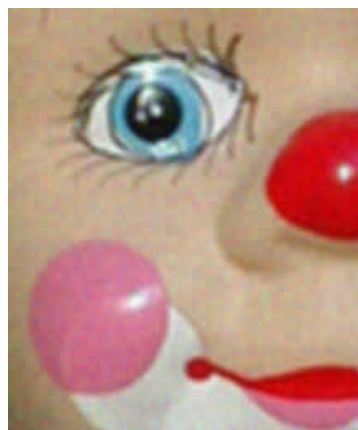


Immagine filtrata con *mean* 5X5

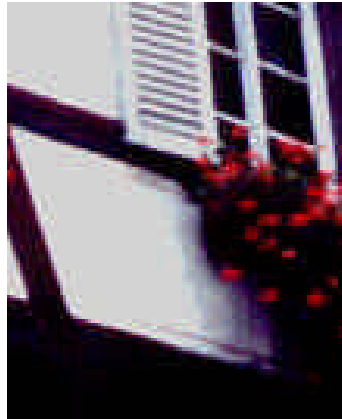


Immagine originale



Immagine con dist. Gaussiano

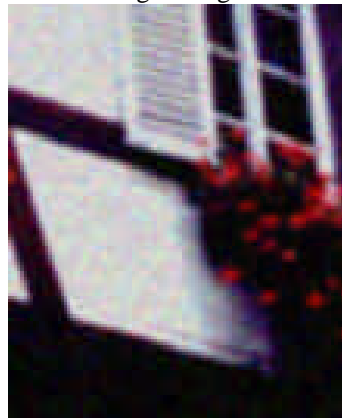


Immagine filtrata con *mean* 3X3

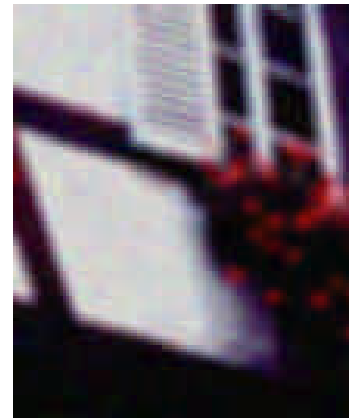


Immagine filtrata con *mean* 5X5

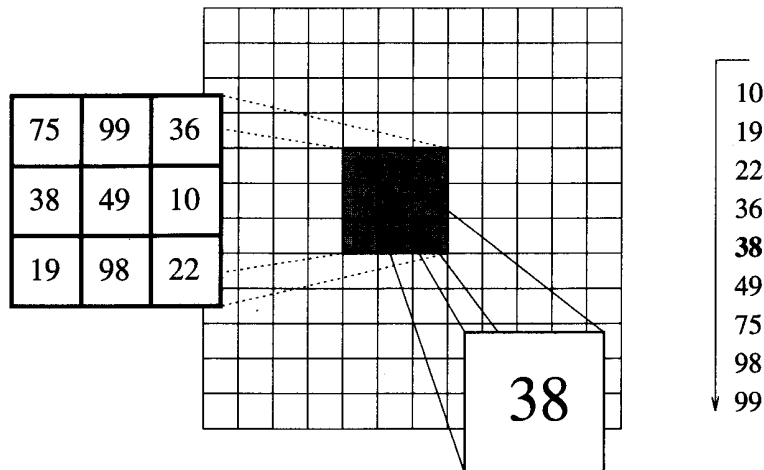
Nelle immagini precedenti è possibile notare la differenza tra l'immagine originale e la stessa con l'aggiunta di un disturbo gaussiano ed in fine quelle filtrate con un *mean filter* rispettivamente 3x3 e 5x5.

E' possibile adottare anche diversi altri pesi per i vari elementi della maschera, l'importante è che la somma dia 1.

- **Il filtro mediano**

Questo filtro fa una cosa abbastanza semplice: raccoglie in una lista tutti i valori dei pixel interessati dalla maschera ponendoli in ordine crescente e successivamente assegna al pixel in esame il valore che si trova a metà della lista stessa.

Nell'esempio sotto riportato 38 è il valore che si trova a metà lista (cioè il mediano) ed è quello che viene assegnato al pixel filtrato.



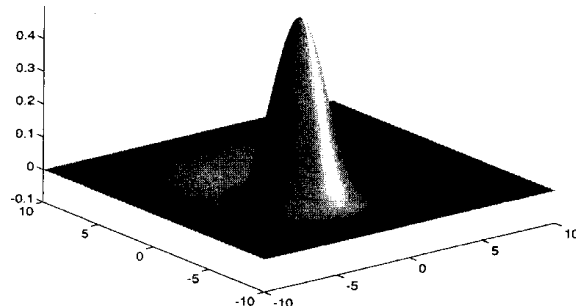
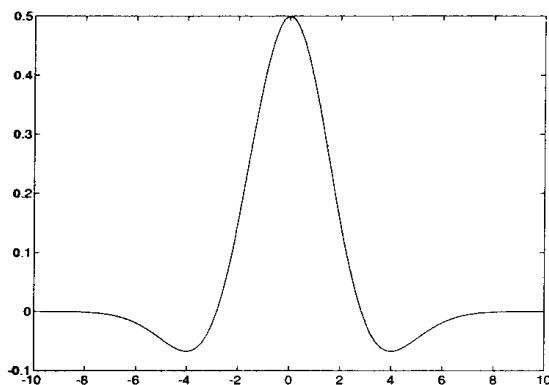
38 è il valore mediano della lista di valori ed è quello assegnato al pixel in esame

- **Filtro gaussiano**

Il filtro gaussiano come dice il nome stesso si basa sulla distribuzione gaussiana. Questa è data dalla formula in due dimensioni

$$g[i, j] = e^{-\frac{(i^2 + j^2)}{2\sigma^2}}$$

i e j sono lo scostamento dal valore 0 mentre σ è la varianza rispetto allo 0



Le 5 proprietà principali sono:

1. La funzione gaussiana in 2 dimensioni risulta essere totalmente simmetrica, questo vuol dire che è ininfluente la direzione verso la quale si applica il filtro, dunque la presenza di bordi di un oggetto del quale non si conosce l'orientamento non influenza l'applicazione del filtro.
2. La funzione gaussiana ha un solo massimo e decresce monotonicamente verso gli estremi tendendo ad azzerarsi.
3. La sua trasformata di Fourier è ancora una gaussiana perciò ha la proprietà del

punto 2.

4. Potendo normalizzare la funzione gaussiana attraverso il parametro sigma si può modificare il valore del picco.
5. La funzione gaussiana in \mathfrak{R}^2 è il risultato di un operazione matematica chiamata convoluzione in due dimensioni di due gaussiane e questo porta a semplificare i calcoli perché è possibile ottenere la matrice del filtro semplicemente convolvendo i due vettori riga che rappresentano l'andamento di ogni singola gaussiana in ciascuna dimensione

ISTOGRAMMI

Per fattori esterni può capitare che durante l'acquisizione di un'immagine si abbia una predominanza di una tonalità che va a coprire in parte le altre, la stessa cosa si può avere anche per quello che riguarda la luminosità di un'immagine, tanto che a volte risulta troppo scura o troppo chiara (è anche possibile trovare immagini in scala di grigio dove però vi è un piccolo range di variazione di intensità). Per risolvere questo problema si può lavorare sull'istogramma dell'immagine stessa.

Un istogramma è la rappresentazione grafica della presenza di un colore.

L'asse delle ascisse viene suddivisa in tante parti tante quanti sono gli elementi della scala del colore che si vuole diagrammare (per esempio se si utilizzano 8 bit per fornire la gradazione di rosso, l'ascissa verrà suddivisa in $2^8=256$ punti), poi si scansiona l'immagine e per ogni pixel che apporta un'informazione del colore scelto si incrementa il valore delle ordinate di 1 in corrispondenza del valore di colore che apporta.

Perciò si avrà che un'immagine con poca variazione di intensità avrà un istogramma concentrato in una zona (cioè il colore prevalente nell'immagine).

Ad un'immagine è associato uno e un solo istogramma, mentre ad un istogramma posso associare più immagini differenti.

Importante è notare che traslazioni e rotazioni di un'immagine non ne modificano l'istogramma.

L'equalizzazione di un'immagine invece fa perdere informazioni e ne varia l'istogramma.

Il problema di una tonalità dominante può essere risolto "stirando" l'asse delle ascisse in maniera tale da ridistribuire i valori del colore in maniera proporzionale.

La formula che permette di equalizzare l'istogramma ed espandere i valori concentrati nell'intervallo $[a, b]$ all'intervallo espanso $[z_k, z_l]$ è la seguente

$$z' = \frac{z_k - z_l}{b - a} (z - a) + z_l = \frac{z_k - z_l}{b - a} z + \frac{z_l b - z_k a}{b - a}$$

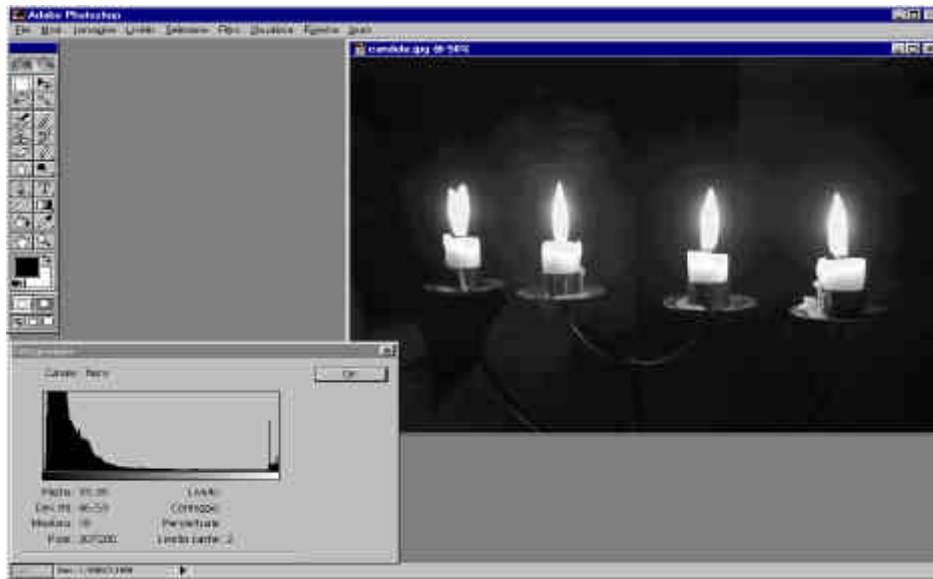


Fig1: Immagine con forte prevalenza di valori scuri; l'istogramma risulta 'schiacciato' su valori bassi

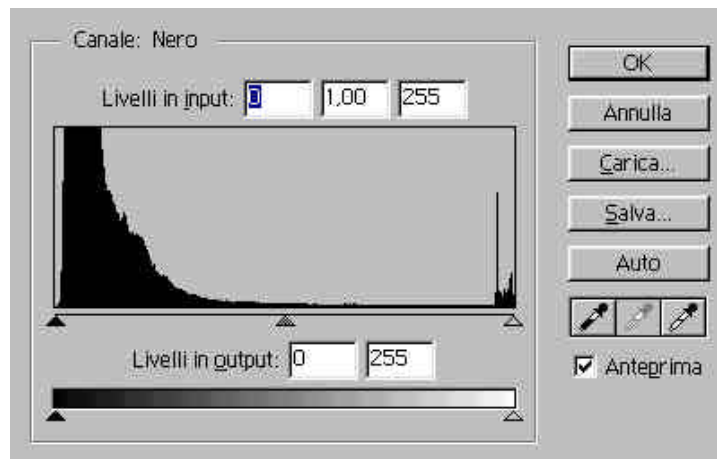


Fig 2: Particolare dell'istogramma

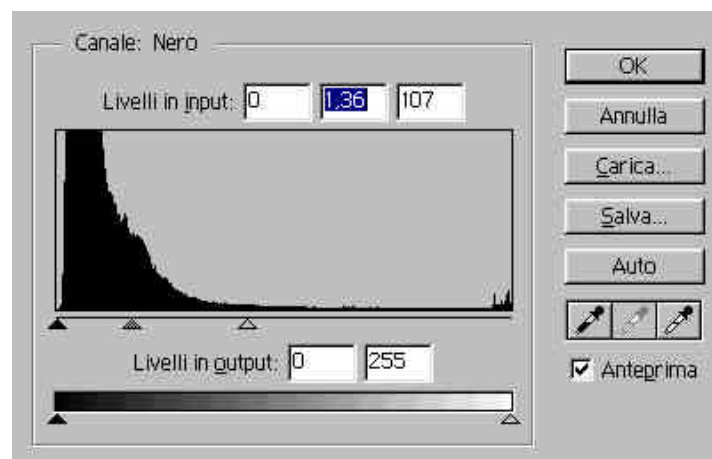


Fig 3: Si reimpostano i valori sull'asse

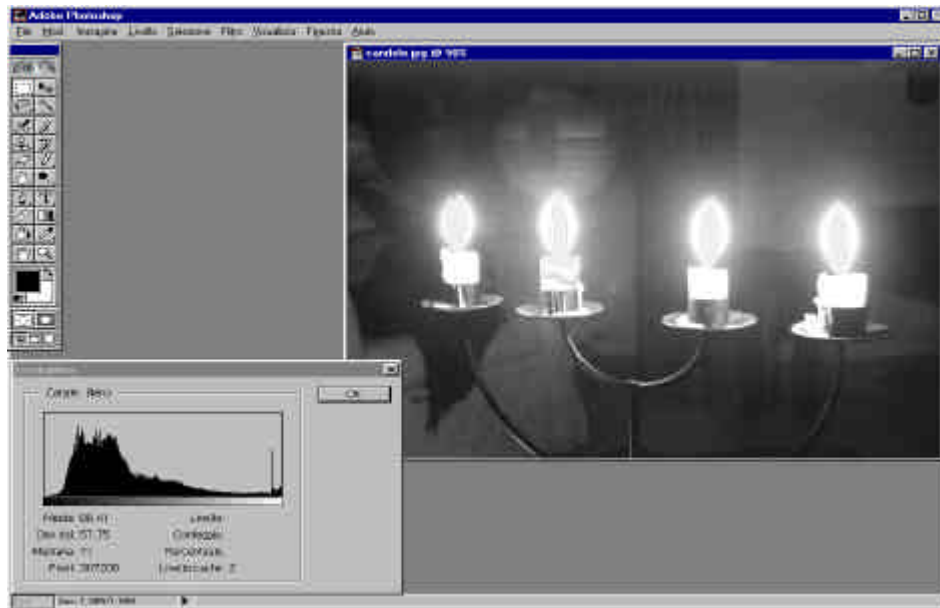


Fig 4: Dopo aver 'stirato' l'istogramma la figura appare più nitida e si scorgono oggetti che prima non apparivano

In fig 1 si può notare l'immagine con il suo istogramma concentrato su valori scuri. In fig 2 e 3 è stato reimpostata la scala delle ascisse il cui valore massimo sarà 107. Nella fig 4 si noti come l'immagine si sia schiarita e l'istogramma si sia dilatato.

SEGMENTAZIONE

L'obiettivo di questa operazione è dividere un'immagine acquisita da una telecamera o altro dispositivo in modo tale da creare zone i cui pixel vengono ad avere una particolare proprietà in comune.

Questa divisione viene eseguita poiché ogni pixel che appartiene ad un'oggetto presente in un'immagine ha caratteristiche in comune con quelli ad esso adiacenti se anch'essi appartengono all'oggetto ritratto.

La segmentazione di un'immagine gode di alcune proprietà.

Sia S l'insieme dei pixel che compongono l'immagine e P_i l'insieme dei sottospazi che si sono venuti a creare le proprietà della segmentazione sono:

1. Completezza: $\bigcup_{i=0}^n P_i = S$, cioè l'intera immagine, $\{P_i\}$ rappresenta una partizione esaustiva.
2. Unicità: $P_i \cap P_j = 0$ con $i \neq j$ $\{P_i\}$ è una partizione esclusiva.
3. Omogeneità: in ogni regione P_i i pixel che vi appartengono hanno qualche proprietà in comune.
4. Connessione: pixel che appartengono a regioni adiacenti se presi insieme non soddisfano la condizione precedente.

Esistono due approcci per la segmentazione di un'immagine:

- Suddivisione dei pixel in zone omogenee
- Ricerca dei contorni – *edge detection*

Vediamo di seguito alcune tecniche classiche per la suddivisione di un'immagine.

• Sogliatura

Abbiamo detto prima che per eseguire una operazione di *segmentazione* di un'immagine si devono marcare con un valore specifico ed univoco i pixel aventi una proprietà in comune, l'algoritmo di sogliatura ha lo scopo di dividere l'immagine in due zone: la figura e lo sfondo. Eseguendo quest'operazione in maniera molto semplice, dato un valore di soglia k associata ad un parametro da noi scelto, divide i pixel dell'immagine in due insiemi, quelli con valore inferiore a k e quelli con valore superiore. La segmentazione per sogliatura dipende fortemente dal contrasto presente in un'immagine, per esempio se si avesse da riconoscere un cubo bianco su di un pavimento bianco, sia l'uomo sia il robot avrebbero notevoli difficoltà perché il confine che delimita la fine di un oggetto (il cubo) con l'inizio di un altro (il pavimento) risulta molto flebile, cosa invece che non accadrebbe se il colore del cubo fosse per esempio nero.

Affinché questo algoritmo risulti robusto, è meglio lasciare all'algoritmo stesso

la scelta del valore di confine k .

Ma come si fa a far scegliere questo valore all'algoritmo stesso ?

Uno dei metodi utilizzati sfrutta le informazioni che l'istogramma dell'immagine può fornire. Una volta ricavato l'istogramma si impone come soglia il valore medio che divide in due parti uguali l'istogramma. Può capitare che un istogramma abbia più picchi di intensità ed in questo caso le operazioni da compiere sono leggermente più complesse. Come prima cosa si divide l'immagine in zone tali che gli istogrammi appartenenti a tali zone abbiano un solo picco, successivamente si lavora su queste singole parti di immagini procedendo come nel caso precedente.

Nel caso di *backlighting* (illuminazione dal fondo) risulta più facile trovare la soglia di divisione.

- **Split and merge**

Riconsideriamo una immagine digitale che deve essere suddivisa in regioni. L'algoritmo di split and merge ci permette di suddividere iterativamente un'immagine in regioni che hanno in comune alcune caratteristiche. Chiamiamo I l'immagine di partenza. Supponiamo che non tutti i pixel dell'immagine I siano simili. L'algoritmo di split and merge genera una prima suddivisione in regioni chiamate I_1, I_2, I_3, I_4 , e supponiamo poi che I_1, I_2, I_3 , siano caratterizzate da pixel simili mentre I_4 no. A questo punto l'algoritmo genera una ulteriore suddivisione dividendo l'immagine I_4 in quattro regioni. Supponiamo a questo punto che solo le regioni I_{43} e I_{44} siano simili: l'algoritmo le trasformerà in una unica regione accorpata.

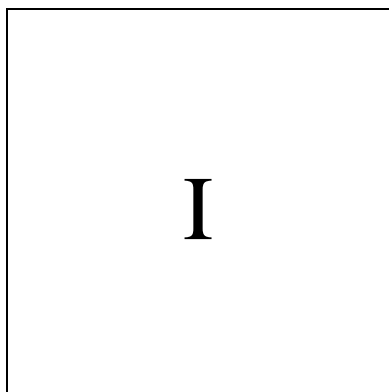


Fig 1: Immagine di partenza

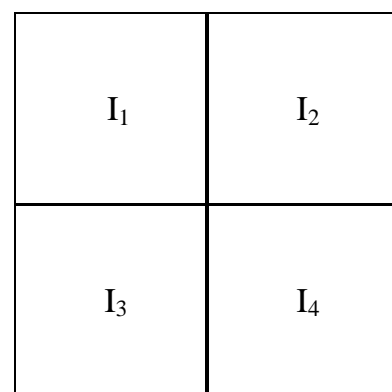


Fig 2: prima divisione

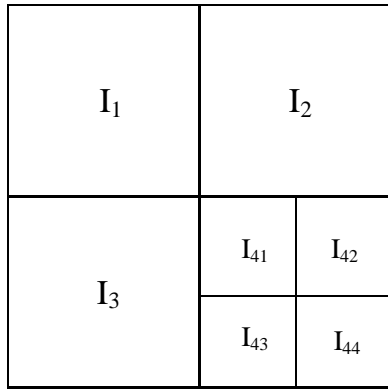


Fig 3: Seconda divisione

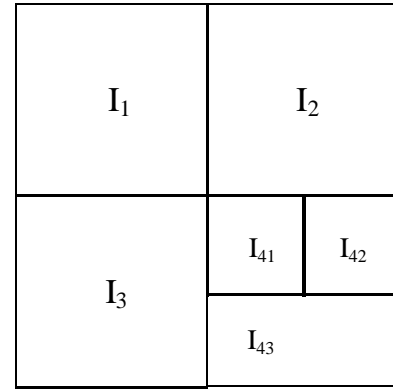
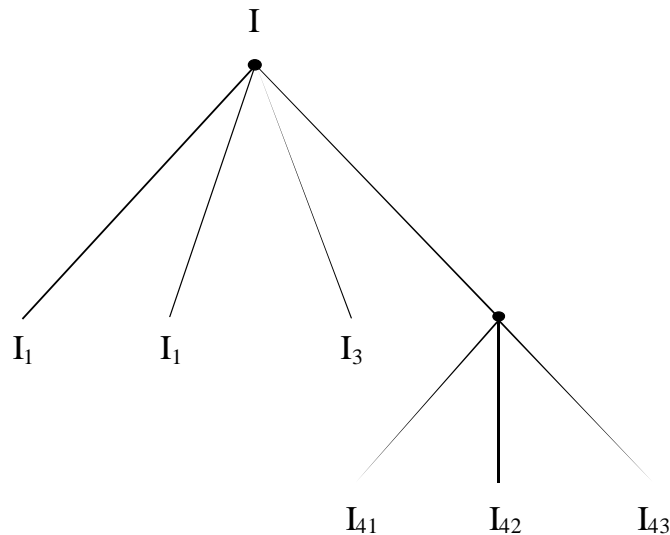


Fig 4: Accorpamento

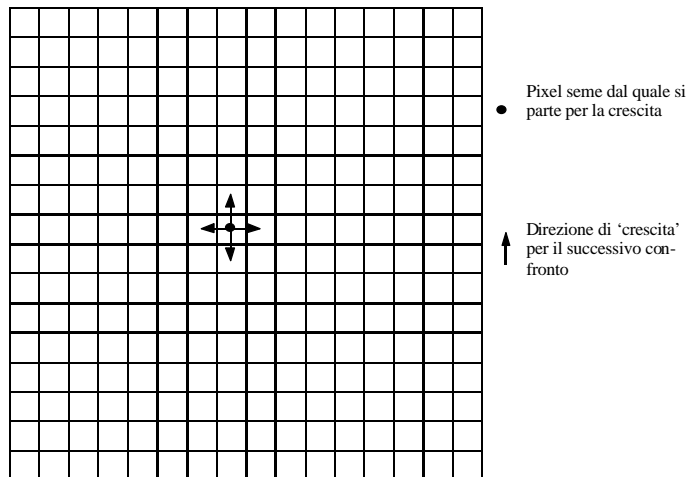
Per evidenziare l'aspetto ricorsivo dell'algoritmo è possibile rappresentare il suo svolgimento con una struttura ad albero :



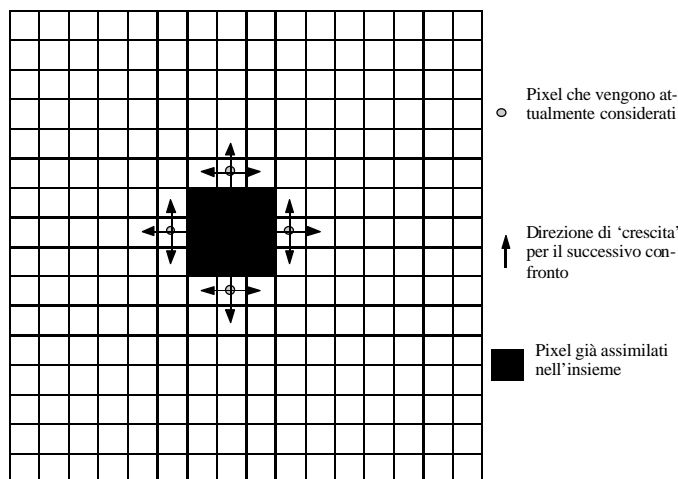
Andamento del processo di divisione in regioni dell'immagine originaria

- **Region Growing**

L'approccio con il metodo di *Region growing* lavora in modo opposto all'approccio con metodo di split and merge. Dapprima viene formato un insieme iniziale di piccole aree raggruppate in funzione delle loro caratteristiche, poi si comincia scegliendo un *pixel seme* arbitrario che viene comparato con i pixel vicini. A questo punto la regione di crescita relativa a questo pixel seme viene formata aggiungendo all'insieme i pixel che risultano essere simili a quello di partenza. Quando l'allargamento della regione si ferma si sceglie un'altro pixel seme che non sia stato ancora considerato in nessuna regione già controllata. Questo processo continua fino a quando non è stata coperta tutta l'immagine. Il processo di region growing solitamente porta ad una buona segmentazione dell'immagine ed è molto utile nell'edge detection.



Esempio di come parte la ricerca della regione



Esempio di come parte la ricerca della regione

Partendo da pixel particolari e forzando l'espansione della regione è possibile influenzare il processo di segmentazione in favore dell'area che per prima viene coperta, introducendo alcuni effetti indesiderati:

- La regione che si sta espandendo domina il processo di region growing e alcuni edge di regioni adiacenti non vengono risolti correttamente.
- Scelte differenti di pixel di partenza portano a differenti stati di segmentazione.
- Ci possono essere dei problemi di risoluzione se vengono scelti seed pixel su linee di edge.

Per ovviare a questi inconvenienti è stata studiata una procedura leggermente diversa che si basa sempre sul region growing.

Questo approccio è chiamato *simultaneous region growing*.

In questa tecnica si tiene conto delle regioni adiacenti e dei problemi che esse possono generare. Infatti la simultaneous region growing permette una elaborazione parallela delle regioni accorpando in regioni uniche quelle regioni che presentano caratteristiche analoghe.

Non viene permesso ad un processo di growing di dominare l'intera procedura e viene stabilito un limite sulle regioni che possono essere elaborate contemporaneamente.

Una implementazione su sistemi di calcolo "paralleli" può portare ad un aumento sostanziale delle prestazioni.

EDGE DETECTION

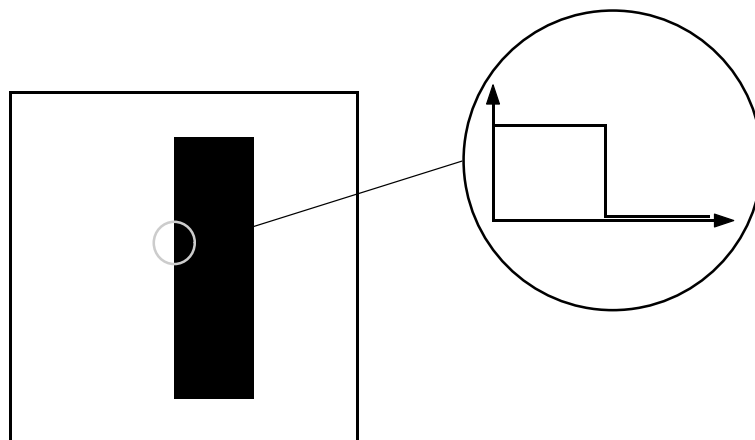
Con l'Edge detection è possibile riconoscere i bordi di un oggetto all'interno di un immagine.

L'idea base che sostiene gli algoritmi di ricerca dei bordi di oggetti all'interno di un immagine è racchiusa nel principio che in un immagine ogni oggetto apporta dei cambiamenti locali. Questi cambiamenti delle proprietà di ogni pixel sono molto netti e coinvolgono quelli che fanno parte dei contorni.

Quindi il contorno in un immagine risulta essere un cambio locale e significativo nell'intensità dei pixel.

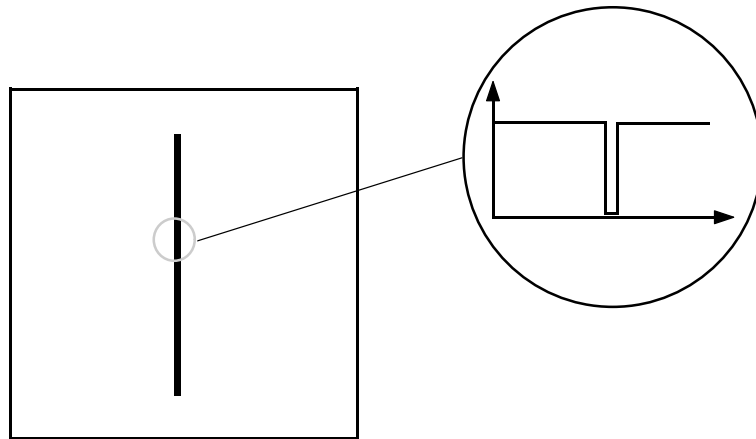
Esistono due tipi di discontinuità:

- *Step discontinuities* discontinuità ad un passo cioè si ha un cambiamento repentino del valore dell'intensità dei pixel. Immaginiamo per esempio una lastra nera su sfondo bianco se si scorre l'immagine da sinistra a destra ad un certo punto si noterà un salto nell'intensità dei pixel e questo salto permarrà fin tanto che la lastra non finisce. Si ha appunto un salto (passo) nell'intensità di colore dei pixel.



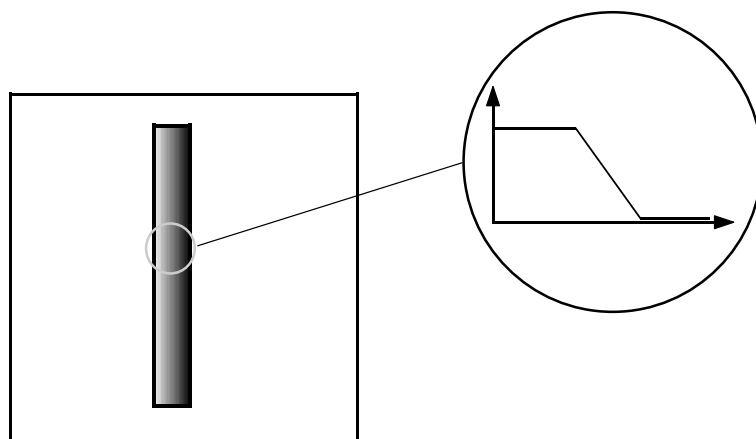
Graficando l'andamento dell'intensità si ha un salto

- *Line discontinuities* dove si ha un repentino cambio di valore dell'intensità ma si ritorna al valore precedente dopo una breve distanza. Si può immaginare un filo nero su di uno sfondo bianco, se si scorre l'immagine da sinistra a destra si otterrà un salto nell'intensità del pixel ma subito dopo (pochi pixel dopo) si ritornerà al valore che contraddistingue lo sfondo. Si ha appunto una linea all'interno dell'immagine.



Graficando l'andamento dell'intensità si ha un salto ma si ritorna quasi subito al valore iniziale

Bisogna comunque ricordare che sia l'uno sia l'altro risultano essere casi rari nella realtà poiché pochi oggetti apportano questo tipo di cambiamento all'interno di un immagine. Molto più spesso si ha il caso di *line and roof edge* dove il cambio di intensità degli oggetti che compongono l'immagine risulta essere graduale. Ciò significa che il confine di un oggetto può confondersi con lo sfondo, per esempio potrebbe essere poco illuminato, o il colore dell'oggetto è molto simile a quello che gli sta attorno o per altri motivi ancora, ma man mano che si scansiona l'immagine la distinzione con lo sfondo si fa sempre più netta e ad un certo punto il divario diviene così sensibile che l'algorithmo rilevata un confine, anche se in realtà il confine dell'oggetto era già stato superato. Questo fatto può portare a male interpretazioni di ciò che l'immagine contiene.

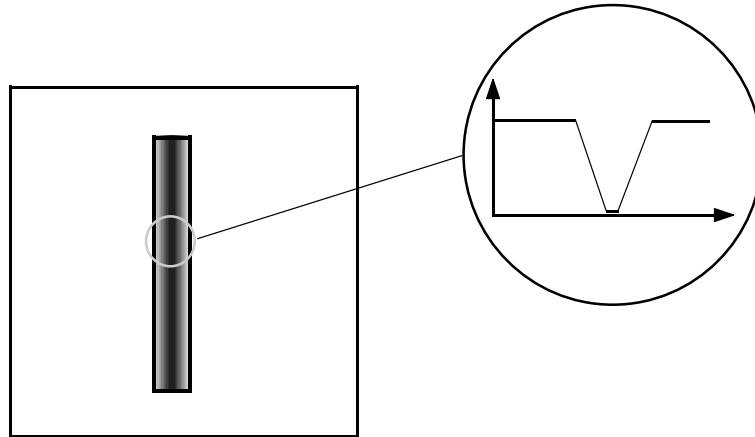


Graficando l'andamento dell'intensità si ha una rampa in discesa (o salita)

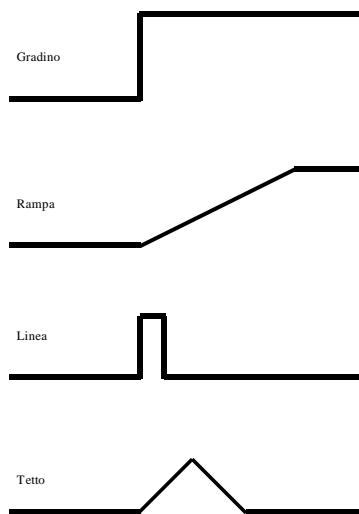
Quello appena presentato era il *line edge*, si ha anche un quarto caso ed è quello dell'*roof edge*. Quest'ultimo è simile al precedente se non per il fatto che non si raggiunge il valore di soglia che fa rilevare il confine, ma arrivato in prossimità di

esso si ha un'inversione di intensità nei pixel e man mano che si avanza nella scansione si ritorna ai valori dello sfondo.

Nelle figure sotto sono riportate le variazioni di gradazione che si hanno nei singoli casi di confine appena illustrati.



Graficando l'andamento dell'intensità si ha una rampa in discesa subito seguita da una in salita



Alcuni esempi di confine presenti in immagini

1. A gradino
2. A rampa
3. Linea
4. A tetto

OPERATORI DEL PRIMO ORDINE

- **Gradiente**

L'operazione fondamentale nella ricerca di confini è, come già detto, trovare i bruschi cambiamenti nell'intensità dell'immagine. Questa ricerca si limita a cercare delle discontinuità nella derivata prima della funzione di intensità dell'immagine. Il gradiente è appunto la misura di quanto la derivata prima varia e, potendo ipotizzare un'immagine come una funzione continua dell'intensità dei pixel che la compongono, si avrà che elevati sbalzi della funzione gradiente identificano un punto di discontinuità.

Il gradiente è la funzione bidimensionale delle derivate parziali ed è definito da

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Il vettore $G[f(x, y)]$ indica la direzione della massima variazione incrementale della funzione $f(x, y)$, la cui ampiezza è data dalla formula

$$G[f(x, y)] = \sqrt{(G_x^2 + G_y^2)}$$

Risulta comunque più semplice approssimare il gradiente con i valori assoluti cioè

$$G[f(x, y)] \cong |G_x| + |G_y|$$

La direzione del vettore gradiente è data da

$$\alpha(x, y) = \arctan \left(\frac{G_y}{G_x} \right)$$

dove l'angolo α è misurato rispetto all'asse delle ascisse x , mentre per la lunghezza del vettore gradiente si assume il massimo dei valori rispetto a X e Y .

$$G[f(x, y)] \cong \max(|G_x|, |G_y|)$$

- **Approssimazione digitale**

Quanto si è visto fino ora è perfetto trattando immagini in formato analogico, ma nella gestione digitale non si ha un passaggio continuo di valori e perciò non è possibile applicare la funzione di derivazione. Per questo è opportuna l'introduzione dell'approssimazione digitale. Questa non fa altro che sostituire la funzione derivata continua con quella discreta che, pur essendo più approssimata, meglio si adatta alla trattazione delle immagini in formato digitali.

Quindi la nuova funzione derivata viene così calcolata:

$$G_x \cong f[i, j+1] - f[i, j] \quad G_y \cong f[i, j] - f[i+1, j]$$

Bisogna tener presente che la j corrisponde alla direzione delle x e la i a quella delle y .

Questa operazione può essere implementata con le maschere di convoluzione di seguito riportate

$$G_x = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

C'è però un problema nell'approssimare in questo modo. Il rumore potrebbe ingannare gli algoritmi che cercano i contorni dell'oggetto. Questo problema diminuisce utilizzando delle maschere che selezionano più pixel, infatti se è presente del rumore si può assumere che questo coinvolga 1 pixel ma non gli adiacenti ed elaborando più pixel alla volta il rumore presente in uno di essi viene distribuito anche sugli altri e viene così a pesare di meno. Utilizzando più pixel è come se si calcolasse la derivata utilizzando un passo più piccolo di quello che può fornire il pixel, cioè G_x in $[i, j_{+1/2}]$ e G_y in $[i_{+1/2}, j]$. Per ottenere questo si adottano, invece che maschere di convoluzione di dimensione 2×1 , delle maschere con dimensione 2×2 come quelle sotto riportate

$$G_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Con lo stesso ragionamento è possibile utilizzare maschere 3×3 per calcolare il gradiente del pixel centrale.

- **Fasi**

Gli algoritmi per l'edge detection seguono tre passi:

- *Filtraggio*
L'operatore di gradiente calcola risultati considerando l'intensità di due soli punti, e può essere affetto da rumore, si è soliti applicare un filtro per eliminarlo. L'utilizzo di queste tecniche deve però essere limitato, infatti l'uso troppo pesante di filtri può diminuire eccessivamente le componenti in alta frequenza, che caratterizzano i bordi.
- *Enhancement*
Attraverso l'equalizzazione dell'istogramma si cerca di elevare il contrasto nell'immagine in maniera tale da far risaltare i confini presenti in essa
- *Rilevazione*
A questo punto è opportuno scegliere solo quei punti che possiedono un elevato valore di gradiente in quanto esisteranno sicuramente dei punti il cui gradiente non è 0 e questo può trarre in inganno l'algoritmo di rilevazione utilizzato.

OPERATORI DEL SECONDO ORDINE

L'approccio appena visto funziona ed è semplice da implementare, tuttavia può essere forviante in quanto restituisce molti possibili punti candidati a rappresentare il contorno dell'oggetto presente nell'immagine. Un migliore approccio al problema sarebbe quello di considerare solo quei punti il cui gradiente locale risulta essere il massimo. Dall'analisi matematica sappiamo che presa una funzione $f(x)$ per ottenere le coordinate dei punti con valore massimo (o massimi locali) è sufficiente applicare l'operatore di derivata ed eguagliare a 0 la nuova equazione ottenuta, cioè $f'(x)=0$. Le radici della nuova equazione saranno quei punti da noi cercati.

A questo punto basta applicare lo stesso metodo alla funzione gradiente per ottenere così quei punti il cui gradiente assume valori di massimo locale.

Ci sono due operatori che svolgono questa funzione il Laplaciano e la derivata seconda direzionale:

- **Il laplaciano**
Il laplaciano è una funzione che fornisce i punti di zero equivalentemente alla derivata seconda.
La sua formula è la seguente

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

Come detto precedentemente questo metodo vale per il caso continuo, ma in caso di immagini digitali si dovranno effettuare delle approssimazioni ridefinendo le derivate seconde come

$$\begin{aligned}
 \frac{\partial^2 f}{\partial x^2} &= \frac{\partial Gx}{\partial x} \\
 &= \frac{\partial f[i, j+1] - f[i, j]}{\partial x} \\
 &= \frac{\partial f[i, j+1]}{\partial x} - \frac{\partial f[i, j]}{\partial x} \\
 &= (f[i, j+2] - f[i, j+1]) - (f[i, j+1] - f[i, j]) \\
 &= f[i, j+2] - 2f[i, j+1] + f[i, j]
 \end{aligned}$$

Essendo l'approssimazione fatta nei confronti del pixel [i,j+1], basterà riportarla al pixel [i,j] e, scalando di 1 il valore j nella formula appena trovata, si otterrà

$$\frac{\partial^2 f}{\partial x^2} = f[i, j+1] - 2f[i, j] + f[i, j-1]$$

che rappresenta la derivata seconda approssimata rispetto al pixel [i,j]. Similmente per quanto riguarda la derivata seconda nell'altra direzione si avrà

$$\frac{\partial^2 f}{\partial y^2} = f[i+1, j] - 2f[i, j] + f[i-1, j]$$

Come fatto per il gradiente, anche in questo caso è possibile utilizzare una maschera di convoluzione.

Per il laplaciano si avrà

$$\nabla^2 \approx \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Per enfatizzare maggiormente i pixel che stanno a confine fra un oggetto ed un altro, si utilizza una diversa maschera di laplaciano:

$$\nabla^2 \approx \begin{array}{|c|c|c|} \hline 1 & 4 & 1 \\ \hline 4 & -20 & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array}$$

Come detto anche prima la funzione Laplaciano può segnalare la presenza di un confine se esistono radici dell'equazione, cioè se quest'ultima passa per lo zero.

Questo fatto fa sì che se l'algoritmo che sfrutta questo sistema trova nell'immagine una discontinuità a gradino, l'equazione che ne risulta, applicando il Laplaciano, non ammette soluzioni e l'algoritmo non è quindi in grado di identificare un confine, come per esempio in figura dove si ha un passaggio brusco di intensità da un valore 2 ad un valore 8.

2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8
2	2	2	2	2	8	8	8	8	8

Nella tabella è riportato il valore dell'intensità dei pixel in esame

E' presente un confine....

0	0	0	6	0	0	0	0
0	0	0	6	0	0	0	0
0	0	0	6	0	0	0	0
0	0	0	6	0	0	0	0

Ma non viene rilevato perché si passa da 0 a 6 a poi ancora 0, non avendo valori negativi non si passerà per lo 0

Un caso invece favorevole all'utilizzo del laplaciano è quello di confini tipo *line and roof edge*, dove il passaggio è più graduale e, utilizzando un algoritmo che sfrutta il laplaciano, si otterrà un passaggio per lo 0 che indicherà la presenza di un confine.

2	2	2	2	2	5	2	2	2	2
2	2	2	2	2	5	2	2	2	2
2	2	2	2	2	5	2	2	2	2
2	2	2	2	2	5	3	3	3	3
2	2	2	2	2	5	2	2	2	2
2	2	2	2	2	5	2	2	2	2

Nella tabella è riportato il valore dell'intensità dei pixel in esame

0	0	0	3	-3	0	0	0
0	0	0	3	-3	0	0	0
0	0	0	3	-3	0	0	0
0	0	0	3	-3	0	0	0

Il confine viene rilevato perché si passa da 3 a -3 e questo passaggio deve essere fatto sicuramente passando per lo 0, il che rileva un punto di massimo.

- **La derivata seconda direzionale**

La derivata seconda direzionale non è altro che la derivata seconda applicata nella direzione del gradiente. Viene così implementata:

$$\frac{\partial^2}{\partial n^2} = \frac{f_x^2 f_{xx} + 2f_x f_y f_{xy} + f_y^2 f_{yy}}{f_x^2 + f_y^2}$$

Che in digitale viene implementata così

X

Entrambe questi operatori vanno comunque usati con cautela in quanto, dovendo operare due derivate, del rumore presente potrebbe disturbare i risultati finali.

E' quindi sempre consigliato applicare dei filtri per eliminare il rumore o comunque ridurlo prima di applicare questi operatori.

- **Laplaciano del gaussiano (LoG)**

Come detto in precedenza, la presenza di rumore rovina i risultati degli operatori appena descritti, quindi risulta importante filtrare le immagini prima di elaborarle. E' per questo motivo che si è arrivati a sviluppare il laplaciano del gaussiano (LoG) che combina il filtro gaussiano con la ricerca dei bordi utilizzando il laplaciano.

Le caratteristiche fondamentali del LoG sono:

1. Il filtro per il livellamento è gaussiano.
2. La ricerca dei confini viene fatta attraverso la derivata seconda (il laplaciano è in due dimensioni).
3. Si ricercano gli zero di passaggio nella derivata seconda che corrispondono ad elevati salti nella derivata prima.
4. La ricerca di confini può essere stimata anche se vengono aggiunti all'immagine "subpixel" derivati dalla interpolazione lineare per aumentarne la definizione.

Le operazioni matematiche da svolgere sono qui rappresentate

$$h(x, y) = \nabla^2 [g(x, y) * f(x, y)]$$

cioè si applica la funzione laplaciano (∇^2) alla convoluzione tra il nostro segnale di ingresso ($f(x,y)$) e il filtro gaussiano ($g(x,y)$). Utilizzando la regola delle derivate nelle convoluzioni otteniamo

$$h(x, y) = [\nabla^2 g(x, y)] * f(x, y)$$

dove

$$\nabla^2 g(x, y) = \left(\frac{x^2 + y^2 - 2s^2}{s^4} \right) e^{-\frac{(x^2+y^2)}{2s^2}}$$

è la funzione del filtro gaussiano

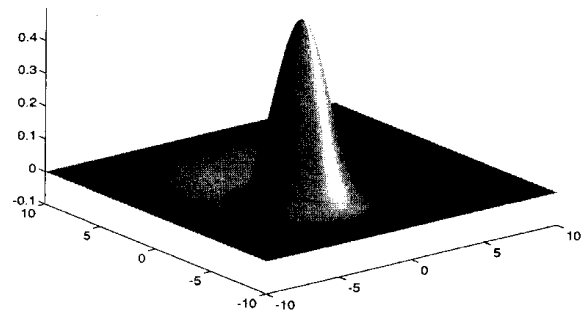
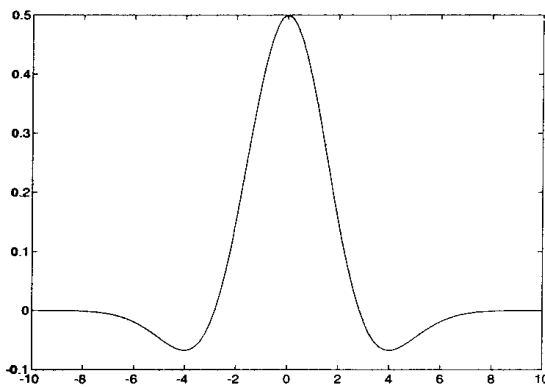


Figure che rappresentano il filtro Gaussiano in una e due dimensioni

Anche qui come nei casi precedenti esistono delle maschere di convoluzione $n*n$ che semplificano notevolmente il lavoro degli algoritmi

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

Maschera 5x5 del filtro Gaussiano

0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-3	-2	-1	0
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	0
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1
-1	-1	-3	-3	-3	4	12	21	24	21	12	4	-3	-3	-3	-1
-1	-1	-3	-3	-2	2	10	18	21	18	10	2	-2	-3	-3	-1
-1	-1	-3	-3	-3	0	4	10	12	10	4	0	-3	-3	-3	-1
0	-1	-2	-3	-3	-3	0	2	4	2	0	-3	-3	-3	-2	0
0	-1	-1	-2	-3	-3	-3	-2	-3	-2	-3	-3	-3	-2	-1	0
0	0	-1	-1	-2	-3	-3	-3	-3	-3	-3	-3	-2	-1	-1	0
0	0	-1	-1	-1	-2	-3	-3	-3	-3	-3	-2	-1	-1	-1	0
0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0
0	0	0	0	0	0	-1	-1	-1	-1	-1	0	0	0	0	0

Maschera 17x17 del filtro Gaussiano

TRASFORMAZIONE DI HOUGH

Trovare i confini di un oggetto è molto diverso dal tracciarne il contorno. Per fare questo si può usare il metodo che utilizza la trasformata di Hough.

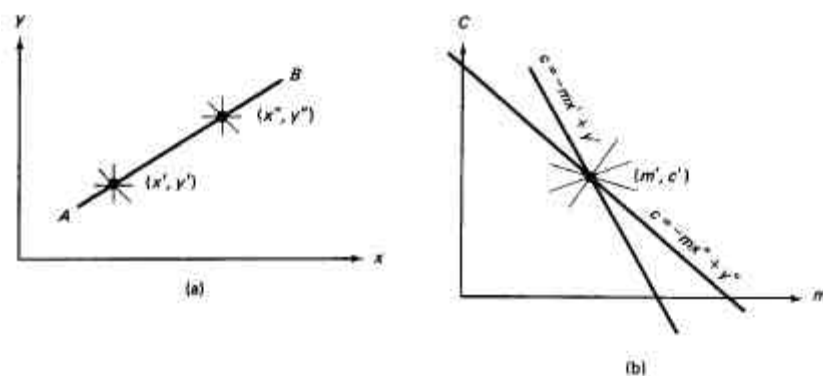
Questo sistema è tale da permettere di trovare il contorno di un oggetto, anche con una scarsa conoscenza di dati, a patto che il contorno stesso possa essere descritto da curve parametriche (per esempio rette o curve coniche). Tale metodo risulta anche essere immune al rumore o alla mancanza di punti che descrivono la curva dell'oggetto stesso.

Prima di introdurre questo metodo si consideri il problema di determinare una singola linea in un'immagine supponendo che l'immagine stessa sia già stata elaborata da qualche algoritmo che ha trovato i pixel che delineano i confini dell'oggetto.

La tecnica di Hough organizza questi pixel in linee rete considerando tutte le possibili linee rette che si ottengono tracciando solo quelle che meglio rappresentano i pixel dell'insieme di partenza.

Ora considerando una coppia di punti $[(x',y'),(x'',y'')]$ sul piano x,y vi sarà una sola retta di equazione $y=mx+c$ che intercetterà i due punti dati.

Data questa equazione perciò è possibile definire univocamente la coppia (m,c) che, in un piano cartesiano m,c , identifica univocamente un punto. È possibile anche dire che in questo nuovo piano, tutte le rette $c=-mx+y$ che passano per questo punto (m,c) hanno una coppia di coefficienti (x,y) che soddisfano la prima equazione della retta che era stata data.



Con l'Algoritmo di Hough e grazie a questa biunivocità tra i due spazi, è possibile seguire questi passi:

1. Suddividere lo spazio parametrico (m,c) in maniera che possa contenere il max e min valore di m e c .
2. Supporre che ciascuna cella in questo spazio sia un accumulatore numerico che viene inizializzato al zero.
3. Per ogni punto nello spazio (x,y) dell'immagine che soddisfa l'equazione di

- partenza incrementare di una unità il valore della cella di accumulazione.
4. Alla fine l'accumulatore che avrà il massimo valore sarà anche quello che modellerà meglio la ricerca dell'oggetto.

OPERATORI MORFOLOGICI

Le trasformazioni morfologiche derivano il loro nome dalle azioni principali che compiono, cioè intendono trasformare la forma di un'oggetto. In molte applicazioni per la visione robotica, per implementare un algoritmo che possa intervenire e modificare un'oggetto, risulta essere più naturale considerarne la "forma". La base fondamentale dell'approccio morfologico, nello studio ed elaborazione delle informazioni visive, è l'immagine binaria.

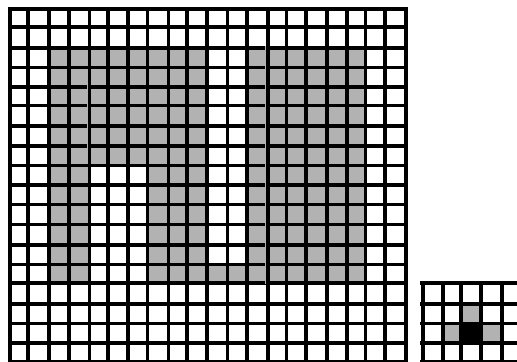
L'*intersezione* di due immagini binarie A e B, $(A \cap B)$ è un'immagine binaria rappresentata da tutti i pixel che risultano appartenere sia ad A sia a B.

L'*unione* di A e B $(A \cup B)$ è l'insieme di pixel presenti in almeno una delle due immagini.

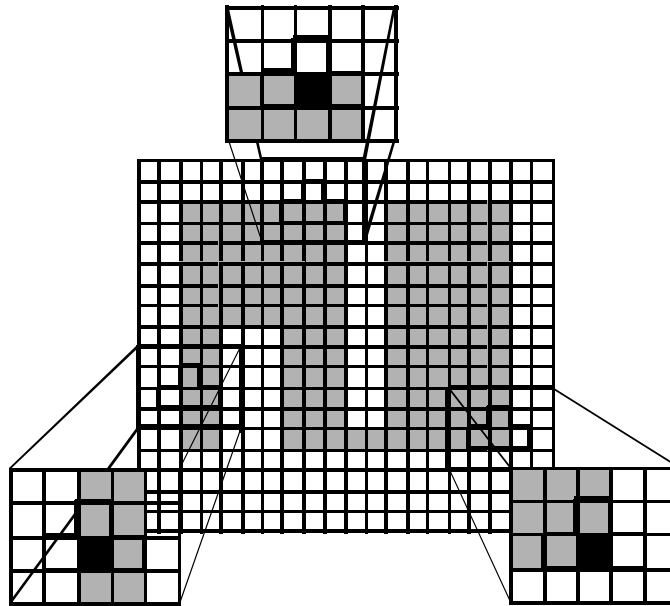
Il vettore somma di due pixel $p[i,j]$ e $q[k,l]$ è il pixel $p+q$ in posizione $[i+k,j+l]$, per quanto riguarda la differenza tra due vettori-pixel $p[i,j]$ e $q[k,l]$ avremo $p-q$ in posizione $[i-k,j-l]$.

Applicando ora queste funzioni ad un'immagine è possibile ottenere alcuni algoritmi.

Nella figura sotto riportata sono mostrate rispettivamente: la figura A (la più grande) sulla quale saranno applicati i vari operatori morfologici, la figura B (la più piccola) che funge da maschera. Il centro della figura B sarà preso come pixel di riferimento per gli operatori, ciò significa che quando saranno applicati questo pixel scorrerà l'intera figura senza uscire da essa.



Esempi di immagini originali A e B che utilizzeremo con gli operatori



Esempio di come è applicata l'immagine B sulla A

- **Dilation**

L'operatore *Dilatino* non fa altro che prendere i pixel dell'immagine originaria e metterli in **or** logico con i pixel presenti nella maschera. Questa operazione è svolta su tutta l'immagine e viene così scritta:

$$A \oplus B = \bigcup_{b_i \in B} A_{b_i}$$

Questa funzione gode della proprietà associativa e commutativa quindi l'ordine con il quale si opera non è particolarmente importante. Ciò vuol dire che posso considerare sia l'immagine A che la B come se fossero o l'originale o la maschera.

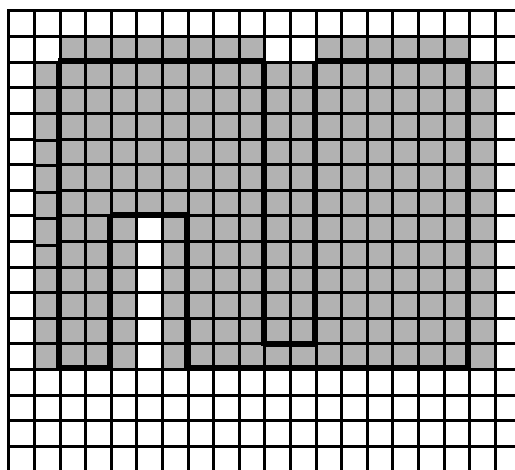


Immagine dopo l'applicazione dell'operatore di *dilation*

- **Erosion**

L'opposto della dilatazione è l'erosione. L'operatore di erosione lascia inalterati quei pixel risultanti da un'operazione di **and** logico tra l'immagine originale e maschera. Questa operazione si scrive con:

$$A \ominus B = \{p \mid B_p \subseteq A\}$$

Anche quest'operatore gode della proprietà commutativa.

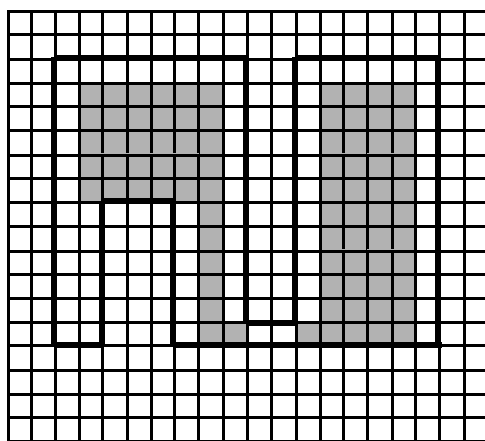


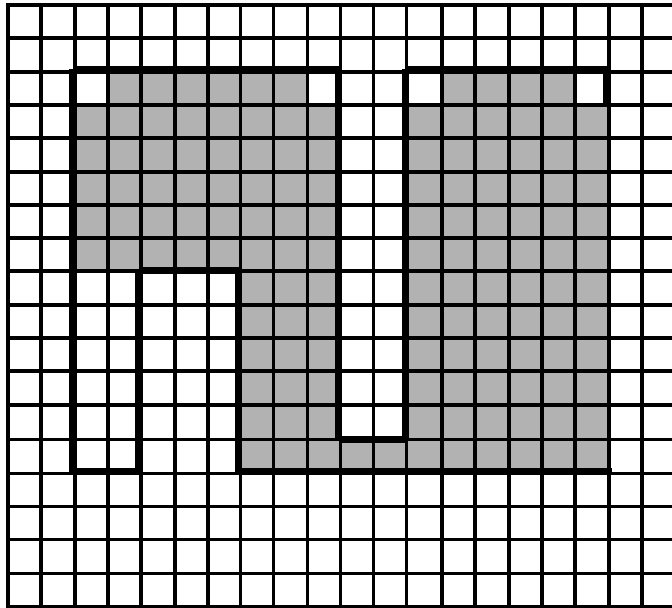
Immagine dopo l'applicazione dell'operatore di *erosion*

Erosione e dilatazione sono spesso usati per filtrare le immagini tanto che, se la natura del rumore percepito durante l'acquisizione dell'immagine è conosciuta, è possibile applicarne una sequenza predefinita in maniera tale da poterlo eliminare.

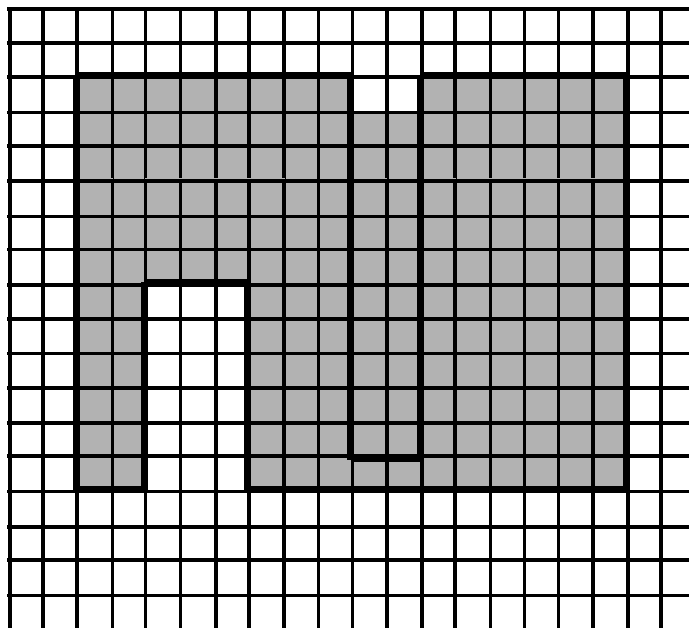
Le operazioni basilari della matematica morfologica si possono combinare in complesse sequenze. Per esempio un'erosione seguita da una dilatazione utilizzando la stessa figura di prova rimuoverà tutti i pixel dell'immagine stessa che saranno contenuti in una piccola regione mentre lascerà inalterati i restanti. Questa sequenza è chiamata *opening*.

La sequenza opposta a quella appena descritta, cioè dilatazione seguita da erosione riempirà i buchi e le concavità più piccole dell'immagine di prova. Quest'altra sequenza si chiama *closing*.

Di seguito sono riportati i risultati di queste due sequenze applicati alla figura A con l'ausilio della maschera B.



L'operatore di opening



L'operatore di closing

SISTEMI DI RISCrittURA

Attraverso i sistemi di riscrittura è possibile esplicitare e memorizzare agevolmente alcune importanti caratteristiche di un' immagine .

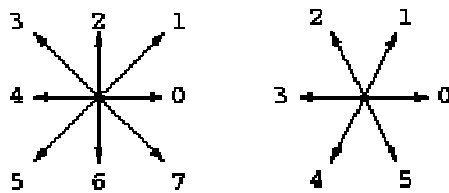
- **Tecnica di CHAIN CODE o analisi di Freeman**

L'algoritmo di chain code è uno dei metodi più popolari ed efficienti per la codifica dei contorni di un oggetto. L' idea base fu introdotta da Freeman nel 1961 (da cui il nome analisi di Freeman).

Questo algoritmo permette di codificare una arbitraria configurazione geometrica.

Ad ogni punto della curva viene associato un vettore di lunghezza unitaria (versori) che può essere disposto secondo un insieme limitato di direzioni prestabilite.

Il contorno viene rappresentato come una sequenza di vettori



Esempio di direzioni possibili.

In funzione del tipo di retinatura dell' immagine, due punti adiacenti nel contorno dell' oggetto, possono essere uniti da uno di questi vettori.

Il contorno questo oggetto può essere codificato memorizzando un punto di partenza e la serie dei numeri relativi alle direzioni dei vettori elencate in ordine.

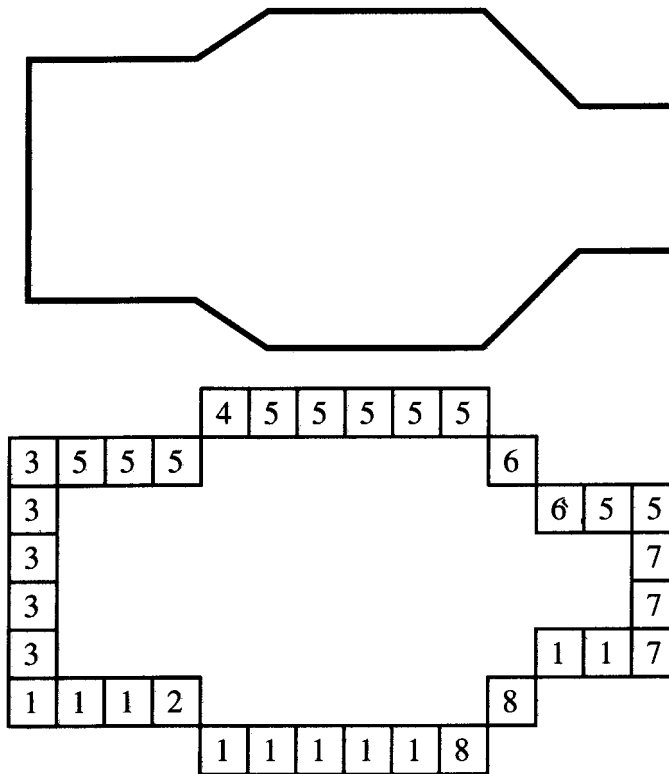
L' elenco può essere compilato in senso orario od antiorario.

Utilizzando la seguente serie di direzioni

2	3	4
1	*	5
8	7	6

Griglia di direzioni possibili

dove l' asterisco rappresenta il vettore e il numero rappresenta la sua direzione, è possibile codificare un' immagine nel modo seguente:



Sopra: contorno originario. Sotto elenco delle direzioni dei vettori

Questa rappresentazione gode di alcune proprietà:

- Le direzioni rappresentate da un numero dispari sono orizzontali o verticali (rettangolari).
- Le direzioni rappresentate da numeri pari sono inclinate di $k\delta + \delta/4$ con k intero.
- L'insieme del punto di partenza e della lista ordinata delle direzioni individuano univocamente un contorno.
- Quando nella lista delle direzioni c'è un cambiamento di valore il contorno presenta un'angolo (*corner*).
- Volendo ruotare di $\pm 45^\circ$ l'oggetto (o di un suo multiplo intero $n \cdot 45^\circ$) basta sommare ad ogni direzione n modulo 8.

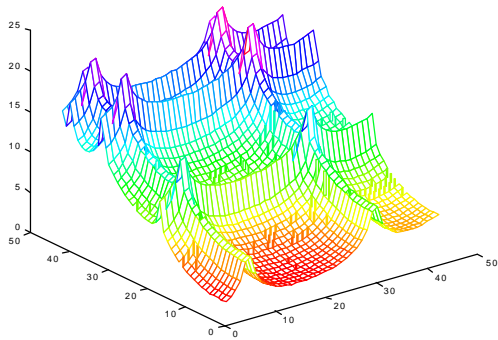
- **Tecnica di SLOPE CODE**

La tecnica di riscrittura denominata *slope code* fu introdotta le prime volte per l'analisi delle immagini provenienti dai satelliti o da telescopi terrestri.

Queste immagini raffiguravano porzioni di spazio con un numero variabile di stelle.

L'algorithmo era in grado di riconoscere diversi *patterns* in modo automatico, i pixel venivano ridisegnati in funzione di quelli a lui vicini. Di seguito è riportato un esempio di Slope Code.

CDI rappresentata come superficie



1	2	3
0	P _x	4
7	6	5

$$sc = \sum_{j=0}^7 2^j + |c \leq c_j|$$

a ₁₁
.
.
.
.	.	.	.	a _{nn}

$$a_{ij} \in (0, 255)$$

Immagine codificata con il codice SC

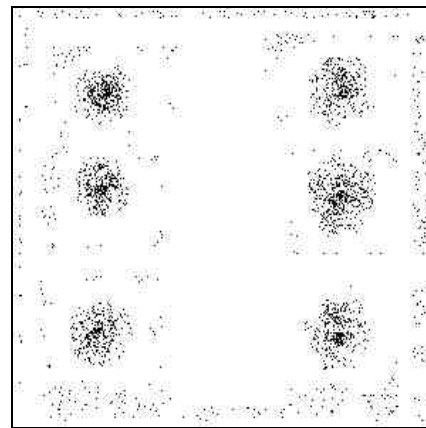
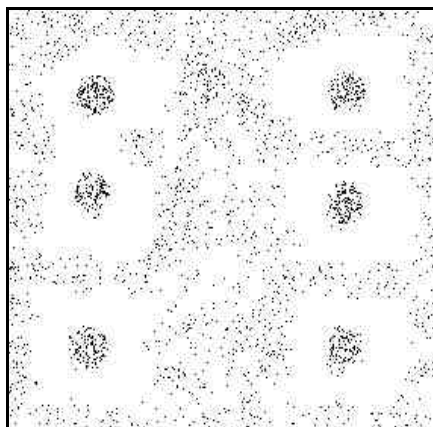


Immagine segmentata

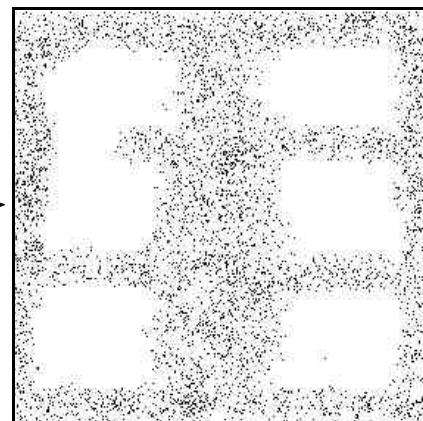
Codice di pendenza
Minimi, massimi,
flessi

Bordi zone non strutturate
colli non rilevanti



Cime degli eccessi

Valli
Selle



Punti di fondo

TRASFORMAZIONI AFFINI 2D

Le trasformazioni affini sono funzioni matematiche che permettono di operare delle trasformazioni sull'oggetto rappresentato. Ne sono diversi tipi tra i quali:

- **Traslazione:**

E' un operazione che permette di spostare un oggetto da una posizione all'altra. Si tratta di uno spostamento rigido dove vengono mantenute inalterate sia le misure dell'oggetto stesso sia i rapporti tra esse. Gli spostamenti che l'oggetto subisce sono comunque paralleli agli assi coordinati.

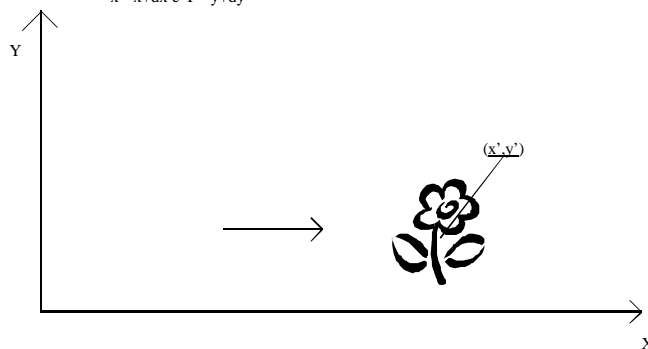
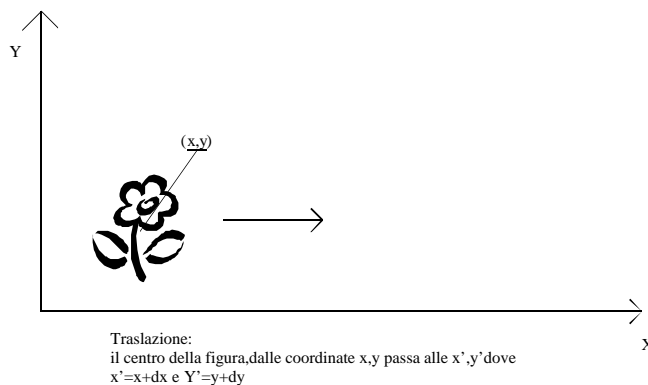
Matematicamente la traslazione si traduce in semplici equazioni quali

$$x' = x + \Delta x$$

$$y' = y + \Delta y$$

Che tradotte in forma matricale vengono scritte come

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 + \Delta x & 0 \\ 0 & 1 + \Delta y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$



Esempio di *traslazione*

- Scalatura:

Opera solo sulle dimensioni dell'oggetto in esame. Ognuna di esse viene moltiplicata per un fattore fisso, se questo fattore è < 1 l'oggetto che ne risulta è più piccolo dell'originale altrimenti se > 1 risulta più grande.

In forma matematica si ottiene:

$$x' = M_x \cdot x$$

$$Y' = M_y \cdot y$$

che tradotte in formula matricale risultano

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} M_x & 0 \\ 0 & M_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Scalatura rispetto al centro degli assi



Esempio di *scaling*

- Rotazione:

Come dice la parola stessa si tratta di una rotazione su un punto che viene fissato.

Anche in questo caso della traslazione ci si riferisce ad una rotazione rigida, in maniera tale da mantenere inalterate le dimensioni dell'oggetto ruotato.

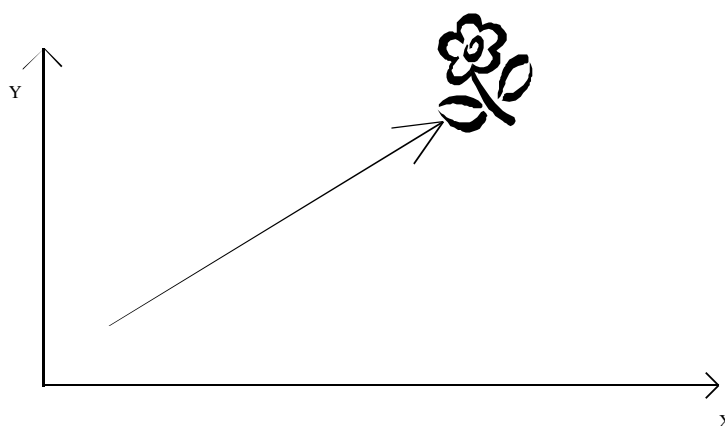
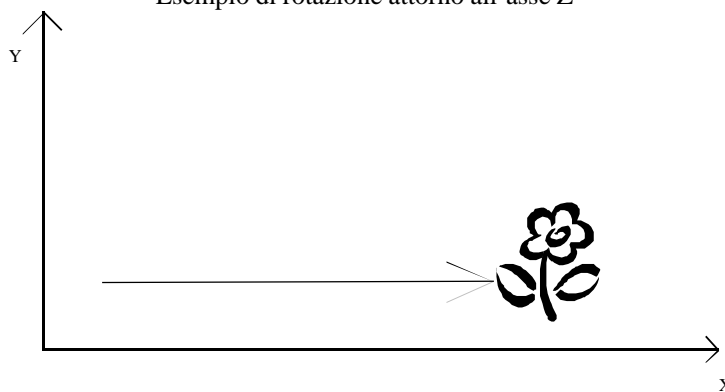
$$x' = x \cdot \cos(\mathbf{q}) - y \cdot \sin(\mathbf{q})$$

$$y' = x \cdot \sin(\mathbf{q}) + y \cdot \cos(\mathbf{q})$$

Con l'uso delle matrici posso sintetizzare ulteriormente il problema così:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\mathbf{q}) & -\text{sen}(\mathbf{q}) \\ \text{sen}(\mathbf{q}) & \cos(\mathbf{q}) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Esempio di rotazione attorno all'asse Z



Esempio di rotazione (notare che il punto origine è al di fuori dell'oggetto)

- Shear (o Sghembatura)

Si tratta sempre di una trasformazione affine ed il suo effetto è di 'piegare' l'immagine come se avesse preso una botta. Di seguito viene riportata la matrice della trasformazione *Share* sia rispetto all'asse delle ascisse (x) che quello delle ordinate (y).

$$SH_x = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix}$$

$$SH_y = \begin{bmatrix} 1 & 0 \\ a & 1 \end{bmatrix}$$

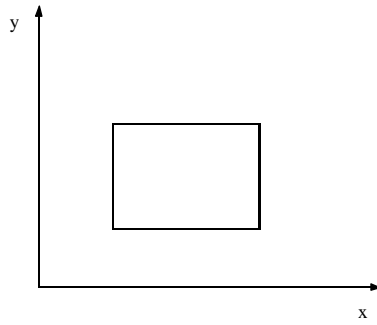


Fig 1: Il rettangolo prima della sghembatura

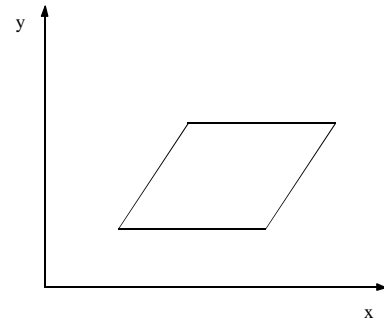


Fig 2: Dopo la sghembatura

Composizione delle trasformazioni

Le trasformazioni bi-dimensionali hanno la caratteristica importante di godere della proprietà di linearità. La proprietà di linearità ci permette di trattare le trasformazioni da un punto di vista matematico in maniera molto efficiente: usando il prodotto fra matrici (classico delle trasformazioni lineari).

Le varie trasformazioni sono così componibili. Infatti se ad esempio si vuole effettuare una rotazione e poi una traslazione si potrà scrivere le equazioni in questo modo:

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 1 + \Delta x & 0 \\ 0 & 1 + \Delta y \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \end{bmatrix}$$

Questa è la matrice della traslazione

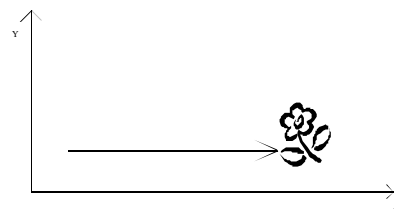
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\mathbf{q}) & -\text{sen}(\mathbf{q}) \\ \text{sen}(\mathbf{q}) & \cos(\mathbf{q}) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Questa quella della rotazione

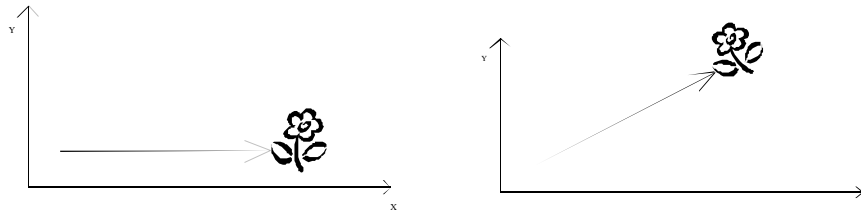
Nel primo sistema, mettendo al posto del vettore colonna $[x' \ y']$ il secondo, otteniamo la composizione prima descritta:

$$\begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 1 + \Delta x & 0 \\ 0 & 1 + \Delta y \end{bmatrix} \cdot \begin{bmatrix} \cos(\mathbf{q}) & -\text{sen}(\mathbf{q}) \\ \text{sen}(\mathbf{q}) & \cos(\mathbf{q}) \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

Una cosa molto importante da considerare è che queste trasformazioni non godono della proprietà commutativa perché anche algebricamente il prodotto tra matrici non è commutativo.



Prima effettuo una rotazione e poi una traslazione



In questo esempio prima una traslazione e poi una rotazione

Come si può notare dalla differente posizione del fiore nello stadio finale le due composizioni non godono della proprietà commutativa.

Le trasformazioni 2D (tranne la shearing) conservano quindi la proprietà nativa degli oggetti nelle immagini, cioè un quadrato se trasformato rimarrà un quadrato e non diverrà un cerchio oppure una linea retta verrà trasformata in un'altra retta diversa in posizione e dimensione, ma sempre retta.

Questo implica che per trasformare un poligono (che è una poliretta) basta operare sui suoi vertici e poi unirli con rette come erano prima.

COORDINATE OMOGENEE

Il sistema a coordinate omogenee è un altro metodo di rappresentazione utilizzato nella *computer vision*. Il sistema in coordinate omogenee è un sistema ridondante, infatti mentre un punto viene rappresentato con tre parametri nel classico sistema cartesiano tridimensionale, nel sistema a coordinate omogenee lo stesso viene a possedere quattro parametri ed il punto è rappresentato da una linea retta.

Lo stesso discorso vale per la rappresentazione 2D. Anche qui un punto rappresentato da due coordinate (x,y) passa ad essere espresso in tre coordinate (x,y,w) e diviene quindi una retta.

Coordinate cartesiane in 3D

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Coordinate omogenee

$$\begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix}$$

normalizzando su w si otterrà

$$\begin{bmatrix} x/w \\ y/w \\ z/w \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

E' quindi evidente che la posizione di un punto può essere assegnata mediante un infinità di quaterne poiché non è significativo tanto il valore delle singole coordinate quanto il rapporto delle prime tre (x,y,z) con la quarta (w).

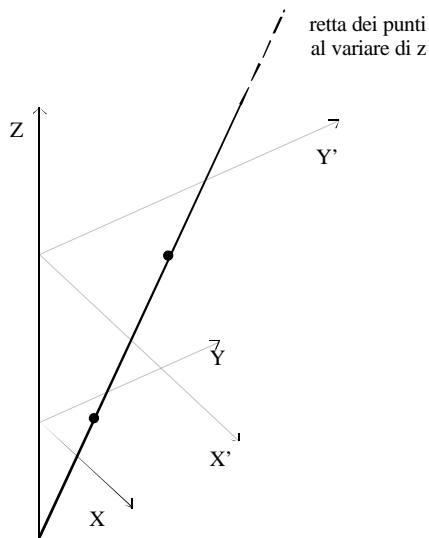
La coordinata w può assumere pertanto il significato di fattore di scala. L'introduzione di questa ultima coordinata permette inoltre di poter rappresentare più facilmente i punti all'infinito (punti impropri) ponendo w=0. Il valore di w è spesso scelto unitario nel caso di punti propri e necessariamente nullo nel caso di punti impropri. Per w=1 abbiamo una corrispondenza immediata tra coordinate cartesiane ed omogenee. Per w=0 le coordinate x,y,z assumono il significato di parametri direttori di tutte le rette che ammettono come punto improprio il punto P₀(x,y,z,0).

La posizione di un punto P può essere rappresentata matricialmente da una delle seguenti matrici colonna

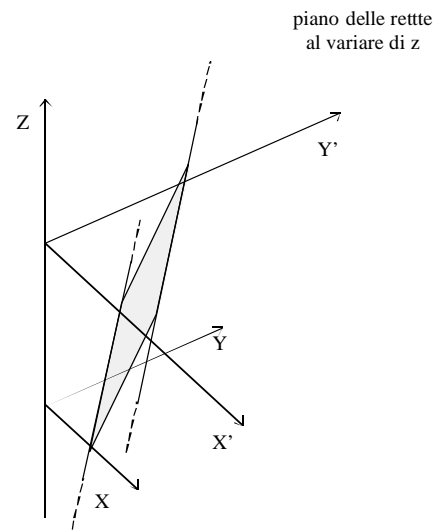
$$P = \begin{bmatrix} wx \\ wy \\ wz \\ w \end{bmatrix}$$

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Aggiungendo così questa quarta dimensione tutti gli oggetti che posseggono $n \leq 3$ dimensioni rappresentati in coordinate omogenee hanno dimensione $n+1$ cioè un punto diviene una retta, una retta un piano e lo spazio un iperspazio ($n=4$ difficilmente rappresentabile)



Da un punto ad una retta



Da una retta ad un piano

Lo scaling e la rotazione di immagini attraverso le coordinate omogenee vengono facilmente implementate con i seguenti sistemi che sono uguali a quelli visti precedentemente tranne per il fatto che posseggono una coordinata in più:

scaling:

$$(x', y', 1) = (x, y, 1) \begin{pmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

rotazione attorno all'asse Z:

$$(x', y', 1) = (x, y, 1) \begin{pmatrix} \cos q & \sin q & 0 \\ -\sin q & \cos q & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Se ad esempio si volesse operare su un'immagine una scalatura di un fattore 2, una traslazione di 1.5 e una rotazione di 45° potremmo sfruttare la proprietà associativa del prodotto fra matrici:

$$(x', y', 1) = (x, y, 1) \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 5 & 1 \end{pmatrix} \begin{pmatrix} 0.707 & 0.707 & 0 \\ -0.707 & 0.707 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$(x', y', 1) = (x, y, 1) \begin{pmatrix} 1.4142 & 1.4142 & 0 \\ -1.4142 & 1.4142 & 0 \\ -2.8284 & 4.2426 & 1 \end{pmatrix}$$

Nel caso tri-dimensionale le varie operazioni matriciali si trasformano in:

Traslazione

$$(x', y', z', 1) = (x, y, z, 1) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ dx & dy & dz & 1 \end{pmatrix}$$

Scaling

$$(x', y', z', 1) = (x, y, z, 1) \begin{pmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotazione antioraria di un angolo q attorno all'asse y

$$(x', y', z', 1) = (x, y, z, 1) \begin{pmatrix} \cos q & 0 & \sin q & 0 \\ 0 & 1 & 0 & 0 \\ -\sin q & 0 & \cos q & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotazione antioraria di un angolo q attorno all'asse x

$$(x', y', z', 1) = (x, y, z, 1) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos q & \sin q & 0 \\ 0 & -\sin q & \cos q & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Rotazione antioraria di un angolo \mathbf{q} attorno all'asse z

$$(x', y', z', 1) = (x, y, z, 1) \begin{pmatrix} \cos \mathbf{q} & \sin \mathbf{q} & 0 & 0 \\ -\sin \mathbf{q} & \cos \mathbf{q} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

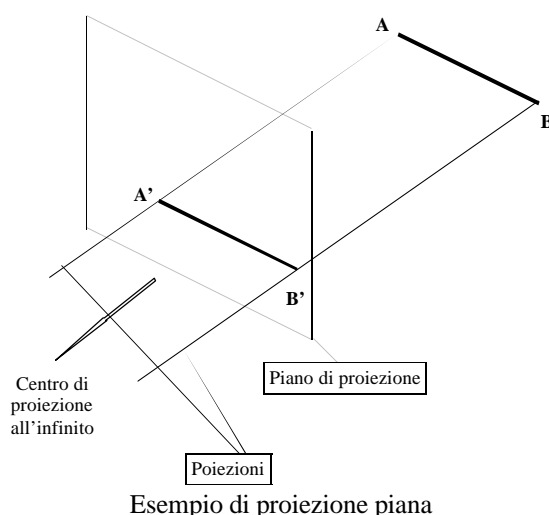
PROIEZIONI DI IMMAGINI

Una *proiezione* è una rappresentazione derivata da un'entità reale. Se l'entità reale è *n-dimensionale* la sua proiezione sarà *(n-1)-dimensionale* (quelle riportate di seguito nelle figure passano da una dimensione 3D ad un in 2D).

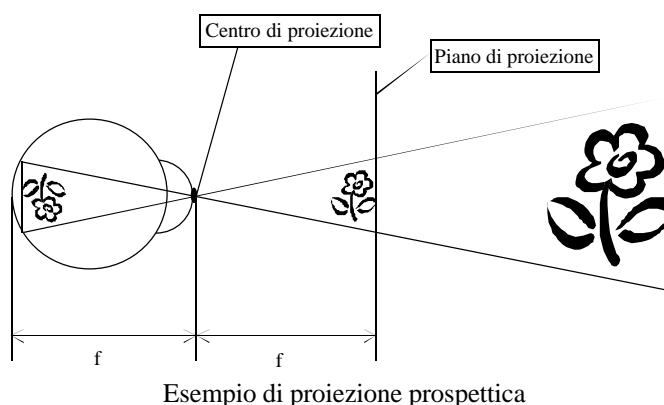
Esistono vari tipi di proiezioni tra le quali:

1. piana o ortografica
2. prospettica

La proiezione piana viene ottenuta supponendo l'osservatore ad una distanza infinita. In questo modo le rette proiettive sono rette parallele e l'immagine presente sul piano prospettico ha le dimensioni uguali a quelle dell'oggetto che si sta osservando



La proiezione prospettica è invece quella più vicina a quella umana. In questo caso l'immagine viene proiettata sulla retina all'interno dell'occhio umano attraverso il cristallino che funge da lente. Questo passaggio oltre che a capovolgere l'immagine e ridurne le dimensioni fa sì che si possa immaginare di vedere lo stesso oggetto "proiettato" su un piano prospettico a distanza f uguale alla focale dell'occhio



Come l'occhio umano anche quello artificiale del robot ha una visione prospettica del mondo.

Si supponga di avere un oggetto ad una certa distanza da un sistema di riferimento fisso x,y,z . Congiungendo con delle rette i vertici significativi dell'oggetto (con significativi intendiamo tutti i punti dell'oggetto che mi permettono di ricostruire l'oggetto stesso tramite la loro unione con linee rette o curve) con l'origine del sistema (x,y,z) è possibile ottenere delle proiezioni facendo incidere queste rette su di un secondo piano. Questo secondo piano può essere posto in tre posizioni distinte, più precisamente il piano può essere:

1. Tra l'oggetto e l'origine del sistema di riferimento. Fig 1
2. Dopo l'oggetto e l'origine del sistema di riferimento. Fig 2
3. Dalla parte opposta rispetto all'origine del sistema di riferimento. Fig 3

Le prime due proiezioni rimpiccioliscono/ingrandiscono rispettivamente la visione dell'oggetto, mentre la terza oltre a potere variare le dimensioni come le prime due capovolge l'immagine dell'oggetto rispetto all'asse o agli assi che stanno fra il piano e l'oggetto stesso.

Il fatto di variare le dimensioni dell'oggetto dipende dal rapporto tra f e z dove f è la distanza dell'origine del secondo piano (x',y') all'origine di quello di riferimento mentre z è la distanza tra l'oggetto e l'origine del piano (x,y) .

Se questo rapporto è minore di 1 allora l'immagine risulterà più piccola, altrimenti se è maggiore di 1 risulterà più grande.

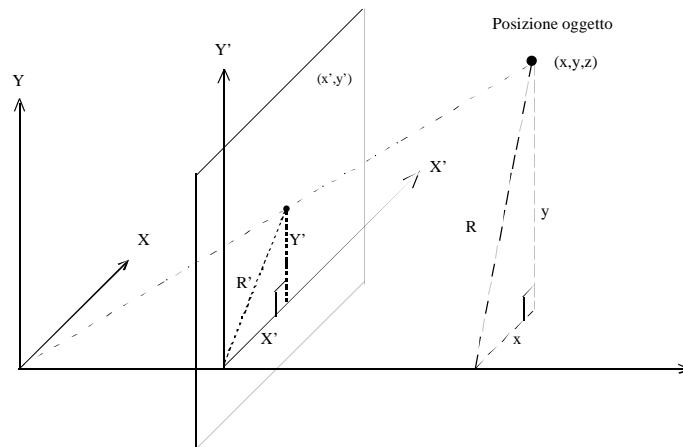


Fig 1 Piano tra l'oggetto e l'origine del sistema di riferimento.

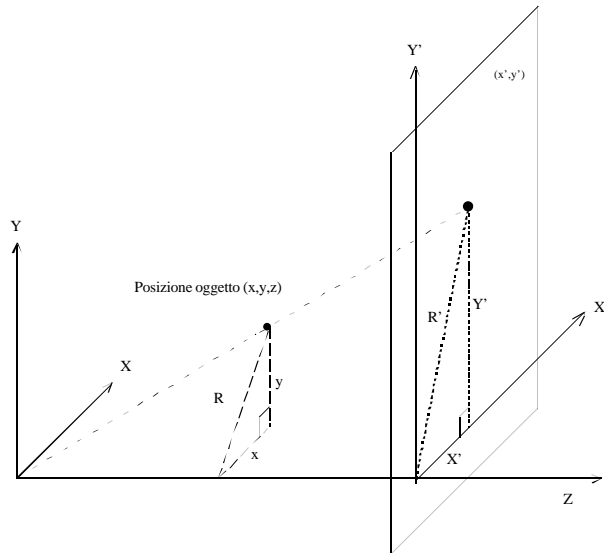


Fig 2 Piano dopo l'oggetto e l'origine del sistema di riferimento.

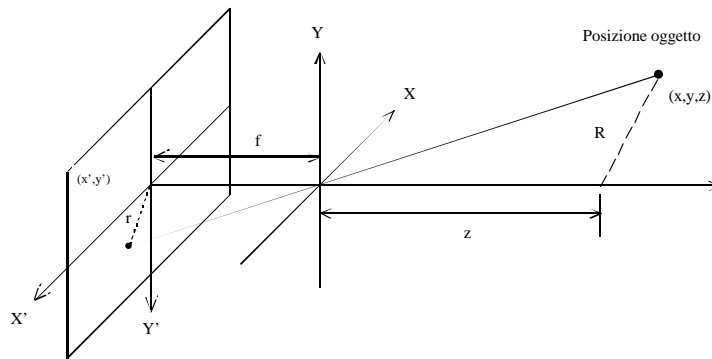


Fig 3 Piano Dalla parte opposta rispetto all'origine del sistema di riferimento.

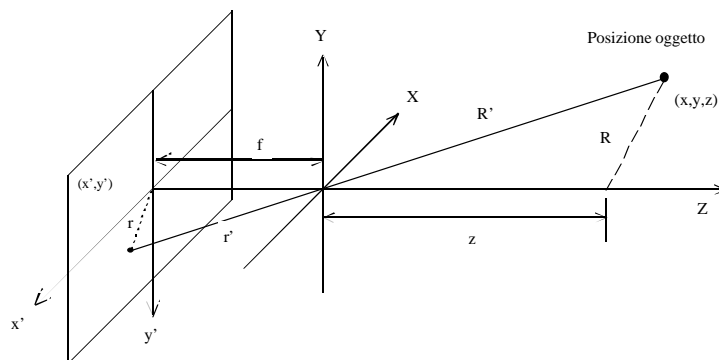


Fig 4

Ora prendendo una proiezione qualsiasi come in fig 4: è possibile osservare che i due triangoli R,Z,R' e r,f,r' sono simili avendo gli angoli uguali e pertanto si può fare la proporzione seguente

$$\frac{f}{z} = \frac{r'}{r}$$

Si può anche scrivere sempre per similitudine che

$$\frac{x'}{x} = \frac{y'}{y} = \frac{r'}{r}$$

Combinando queste due equazioni si ottiene

$$\frac{x'}{x} = \frac{f}{z} \quad \text{e} \quad \frac{y'}{y} = \frac{f}{z}$$

Utilizzando queste formule è possibile calcolare la nuova posizione sul piano (x', y')

$$x' = \frac{f}{z} x \quad \text{e} \quad y' = \frac{f}{z} y$$

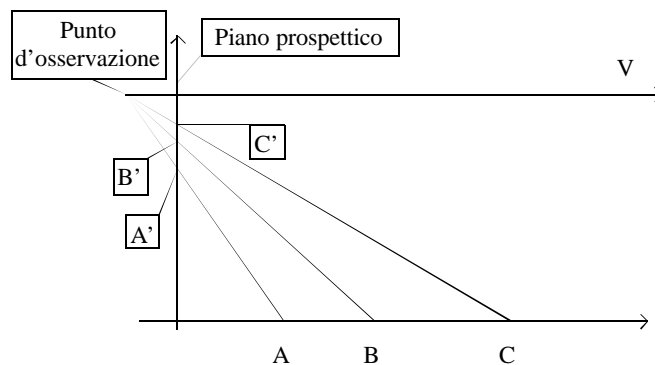
Definiti questi concetti si può supporre che l'origine del piano (x, y, z) coincida con l'osservatore (una telecamera o l'occhio umano).

- **Invarianti proiettivi**

Le proiezioni preservano alcune proprietà del oggetto in esame mentre ne perde altre.

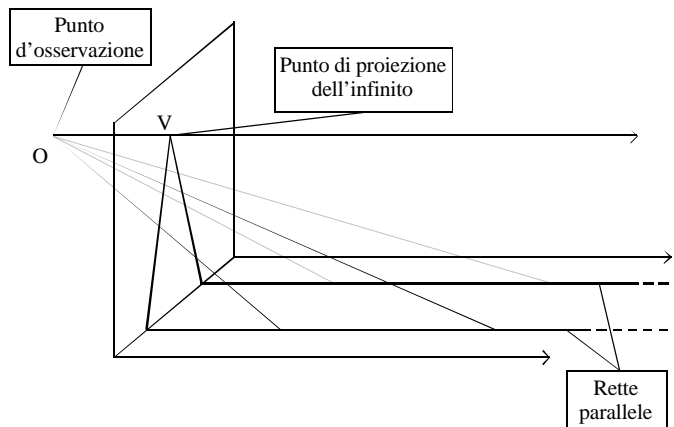
La proiezione prospettica NON preserva:

1. Le distanze e i rapporti tra esse:



Nella figura, i punti A, B, C vengono rimappati nei punti A', B', C' le cui rispettive distanze dall'origine e tra essi sono variate.

2. Il parallelismo

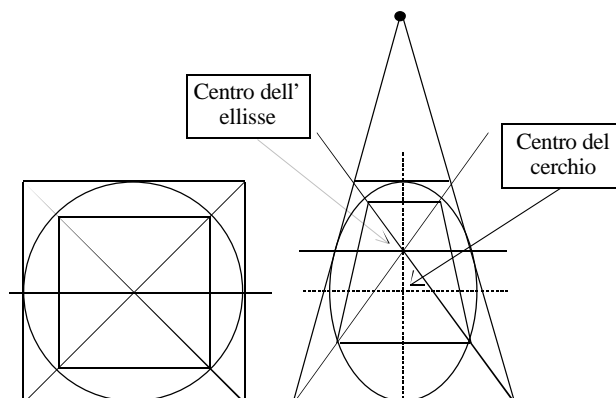


In questa seconda figura i punti delle rette parallele vengono rimappati sul piano prospettico in due rette che si incontrano in un punto V detto anche punto all'infinito in quanto rappresenta il punto dove si mapperebbero le rette se fossero prolungate all'infinito.

3. Le aree degli oggetti né i loro rapporti

Ovviamente se non vengono rispettate le misure monodimensionali (lunghezze e distanze) non potranno essere rispettate nemmeno le aree che altro non sono che prodotti tra lunghezze.

4. Il centro del cerchio



5. La simmetria

Come si vede nella figura, la parte sottostante del cerchio non è più uguale alla parte che sta sopra.

Mentre preserva altre proprietà quali:

1. La collinearità:

Due punti che stanno sulla stessa retta dopo una trasformazione continuano ad essere sulla stessa retta.

2. La tangenza:

I punti di tangenza tra due curve vengono mantenuti.

3. Le intersezioni:

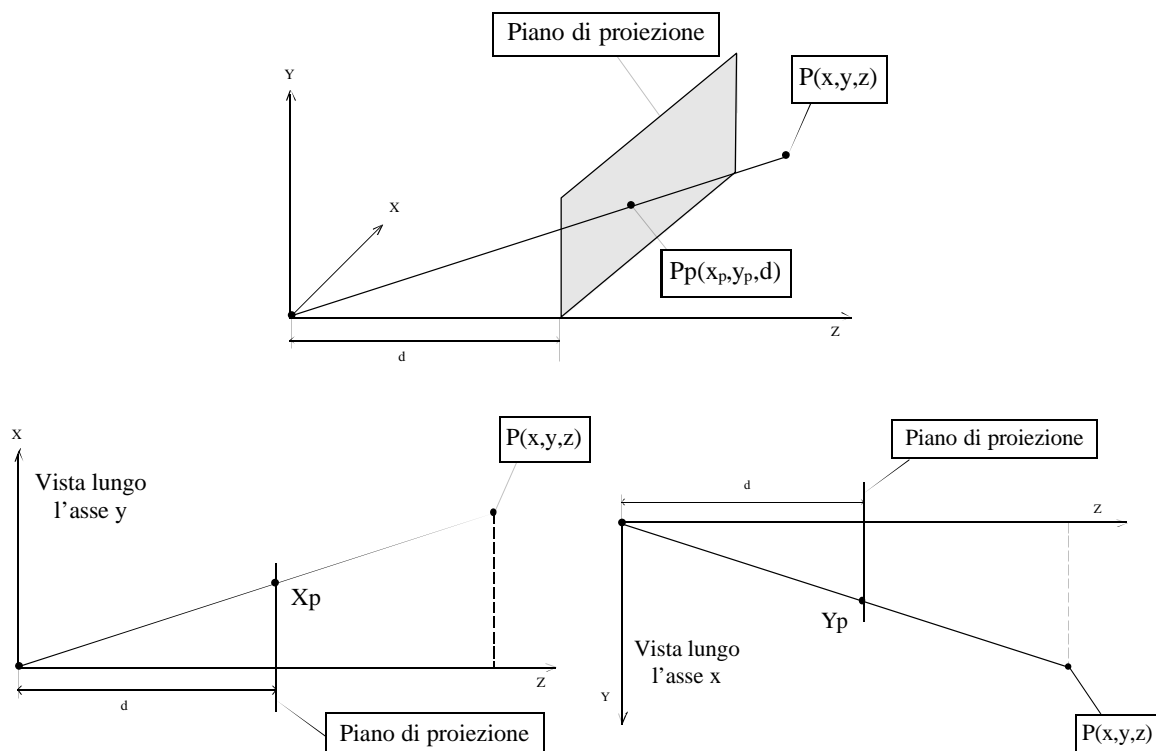
Le curve che si intersecano in un punto mantengono questo punto di intersezione.

LA MATEMATICA DELLE PROIEZIONI

In questo paragrafo vengono introdotte gli aspetti matematici delle proiezioni geometriche.

Ogni proiezione può essere definita da una matrice 4x4. Questo risulta conveniente perché in questo modo sarà possibile utilizzare l'analisi matriciale.

Si inizia col descrivere la proiezione di un punto P su di un piano a distanza d dall'origine.



Per calcolare le coordinate di $P_p=(X_p, Y_p, Z_p)$ si può considerare la similitudine fra triangoli e scrivere:

$$\frac{X_p}{d} = \frac{X}{Z} \qquad \frac{Y_p}{d} = \frac{Y}{Z}$$

Le quali sono riscrivibili nella forma :

$$X_p = \frac{d \cdot X}{Z} = \left(\frac{X}{Z/d} \right) \qquad Y_p = \frac{d \cdot Y}{Z} = \left(\frac{Y}{Z/d} \right)$$

La distanza d è il fattore di scala applicato a X_p e Y_p .

Tutti i valori di Z sono ammessi tranne quello di $Z=0$, tali equazioni possono essere sintetizzate dalla matrice 4×4

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

Moltiplicando il punto $P=[x,y,z,1]^T \cdot M$ si può riscrivere il punto in coordinate omogenee $[X,Y,Z,W]^T$

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = M \cdot P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Equivalentemente

$$[X \ Y \ Z \ W]^T = \left[x \ y \ z \ \frac{z}{d} \right]^T$$

Ora dividendo tutto per W e tornando quindi al 3D si ha:

$$\left(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W} \right) = (x_p, y_p, z_p) = \left(\frac{x}{z/d}, \frac{y}{z/d}, d \right)$$

Queste equazioni sono le stesse scritte in partenza dove d è la distanza sull'asse z del piano di proiezione.

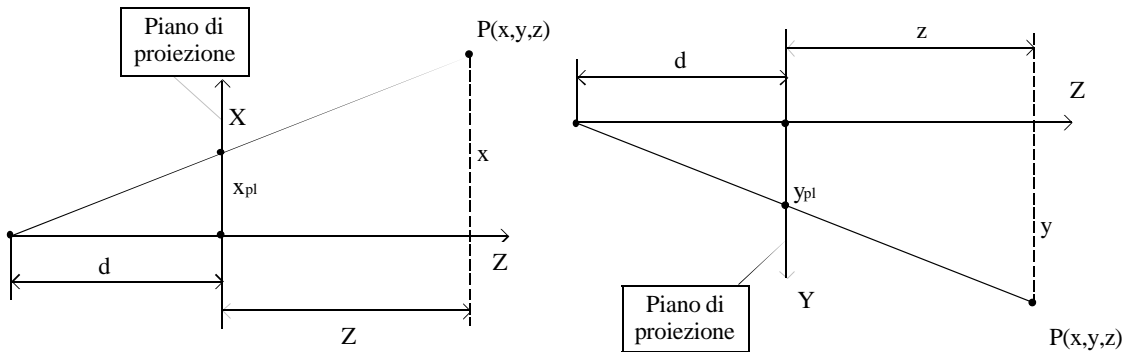
Alternativamente si può supporre che il piano di proiezione sia a $Z=0$ e il centro di proiezione sia a $Z=-d$.

Sempre per la similitudine fra triangoli si scrive:

$$\frac{x_p}{d} = \frac{x}{z+d} \quad \frac{y_p}{d} = \frac{y}{z+d}$$

Moltiplicando per d:

$$x_p = \frac{d \cdot x}{z + d} = \frac{x}{(z/d) + 1} \qquad y_p = \frac{d \cdot y}{z + d} = \frac{y}{(z/d) + 1}$$



la cui matrice diventa:

$$M' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}$$

Supponendo che la distanza d del centro di proiezione sia infinita, ed il piano di proiezione sia in Z=0, si ottiene una proiezione piana dove :

$$x_p = x \qquad y_p = y \qquad z_p = 0$$

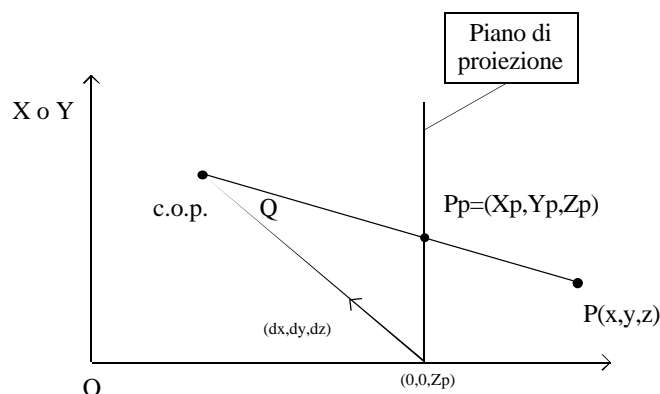
Il vettore di proiezione ha la terna dei parametri direttori con valori (0, 0, -1).

La nuova matrice è :

$$M_{ort} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

che è la matrice M' con d → infinito.

In generale la proiezione di un generico punto P=(x,y,z) su un piano di proiezione perpendicolare all'asse Z e a distanza z_p dall'origine, con un centro di proiezione (COP) a distanza Q dal punto (0,0,z_p) genera P_p=(x_p , y_p , z_p) detto anche punto di proiezione.



La direzione dal punto $(0,0,z_p)$ al COP è data dal versore normalizzato (d_x, d_y, d_z) .
 Dopo alcuni passaggi matematici si arriva a definire la matrice di proiezione generale:

$$M_{generale} = \begin{bmatrix} 1 & 0 & -\frac{d_x}{d_z} & z_p \frac{d_x}{d_z} \\ 0 & 1 & -\frac{d_y}{d_z} & z_p \frac{d_y}{d_z} \\ 0 & 0 & -\frac{z_p}{Qd_z} & \frac{z_p^2}{Qd_z} + z_p \\ 0 & 0 & -\frac{1}{Qd_z} & \frac{z_p}{Qd_z} + 1 \end{bmatrix}$$

Per le varie proiezioni si può considerare la seguente tabella

	Z_p	Q	d_x	d_y	d_z
P.piana	0	Inf	0	0	-1
P.prospettica con centro in d	d	d	0	0	-1
P.prospettica con centro in 0	0	d	0	0	-1

RECUPERO DELLA TERZA DIMENSIONE

Uno dei più affascinanti problemi posti alla computer vision è quello di poter ricavare in maniera automatizzata, avendo come uniche fonti di informazione delle immagini, la distanza tra il punto di acquisizione e l'oggetto inquadrato.

Quando il mondo reale viene acquisito da un dispositivo meccanico (fotografato) si passa dalla realtà in tre dimensioni ad una sua proiezione in due dimensioni.

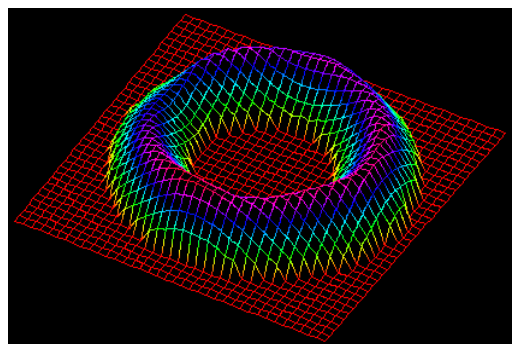
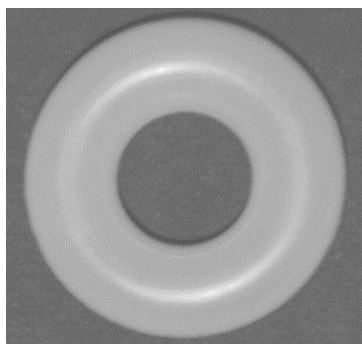
Il processo di acquisizione dunque passando ad una rappresentazione 2D causa la perdita di molte informazioni, tra le quali proprio la possibilità di ricavare la distanza o profondità, degli oggetti raffigurati.

Nella computer vision la ricerca della terza dimensione gioca un ruolo fondamentale, infatti oltre a stimare la distanza tra l'osservatore e l'oggetto permette di capirne le proprietà tridimensionali e in particolar modo la profondità. Questi dati sono molto utili anche nella visione robotica, se si vuole sapere la distanza a cui ci si trova un oggetto, per poterlo toccare, afferrare o evitare. Inoltre conoscendo le reali dimensioni, è possibile trovarne il volume ecc.

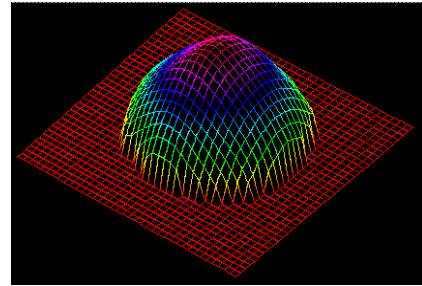
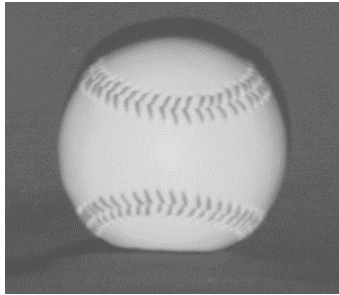
La computer vision però alcuni metodi per ovviare a questa perdita di informazioni.

- *Shape from shading* L'ombreggiatura è un elemento molto importante di una scena, in quanto favorisce la sua interpretazione tridimensionale, determinando la sensazione di profondità. La tecnica di "shape from shading" parte dall'immagine descritta in termini di livelli di intensità luminosa e utilizza questi ultimi per ricavare una rappresentazione delle superfici presenti sulla stessa scena. Le variazioni di intensità luminosa sono proporzionali all'angolo di orientazione della superficie. Questa metodologia è particolarmente utilizzata quando si tratta di esaminare la scena dal punto di vista della luminosità, come nel caso di immagini la cui superficie è completamente priva di elementi pittorici.

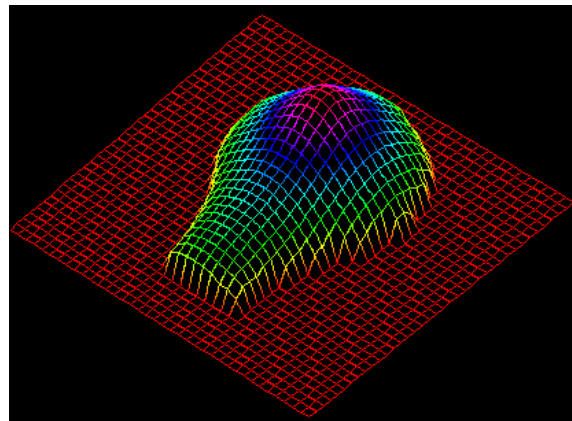
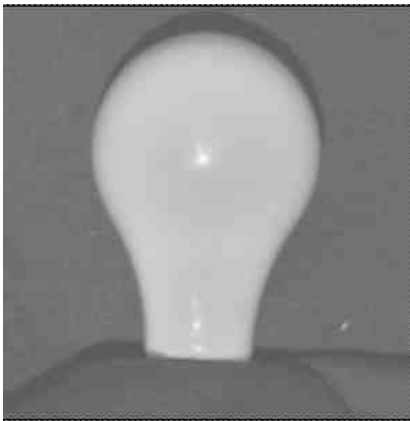
Di seguito verranno riportati alcuni esempi di elaborazione di immagini attraverso la tecnica di *shape from shading*.



esempio di elaborazione dell'immagine con shape from shading



esempio di elaborazione dell' immagine con shape from shading



esempio di elaborazione dell' immagine con shape from shading
(immagini tratte dal sito:www.cs.usask.ca/projects/vision/results/Shape_from_shading.html)

- *Shape from contour* La ricostruzione tridimensionale e l'interpretazione di una scena può essere fatta utilizzando semplicemente il disegno "a linee" della stessa scena, senza alcuna informazione pittorica, usando cioè i contorni bidimensionali ottenuti sul piano immagine per proiezione prospettica. Gli approcci adottati per ricostruire scene tridimensionali da informazioni bidimensionali, utilizzano in modo massiccio, la conoscenza a priori disponibile sulla scena e sono, di conseguenza, quasi simili al processo di interpretazione. E' stata sviluppata una teoria matematica che va sotto il nome di "inversione prospettica" per affrontare il problema della ricostruzione tridimensionale. Un altro approccio è la ricostruzione tramite i contorni "line drawing", ossia da disegni degli spigoli fisici degli oggetti. Limitandosi a scene costituite solo di oggetti con facce piane (il mondo a blocchi), è possibile estrarre informazioni geometriche, in quanto è possibile "etichettare" in modo opportuno le linee in corrispondenza dei tre tipi di contorni tridimensionali: concavi, convessi, di occlusione.
- *Shape from texture* Dal punto di vista concettuale, la ricostruzione tridimensionale basata sulla texture, è analoga alla ricostruzione attraverso i contorni. Per le texture regolari, le metodologie utilizzate per la ricostruzione, si basano sulla considerazione che l'orientamento e la forma della superficie, a cui la texture appartiene, possono essere ricavate

analizzando la deformazione geometrica che gli elementi della texture, detti "texel", hanno subito per prospettiva. L'analisi delle texture non regolari è molto più complessa. Per queste non è possibile fornire una descrizione in termini geometrici; i texel vengono descritti secondo parametri qualitativi quali il colore, le dimensioni e il valore di intensità luminosa. Le tecniche di ricostruzione, in questo caso, sono basate soprattutto su considerazione statistiche globali.

- *Shape from motion* Un'altra tecnica di ricostruzione è quella che ha a disposizione una telecamera che si muove rispetto alla scena inquadrata o viceversa ad un oggetto che si muove rispetto alla telecamera. Durante il movimento, le proiezioni dei punti della scena sul piano dell'immagine, si muovono descrivendo curve che vengono indicate come "curve di flusso ottico". Analizzando queste curve è possibile determinare le distanze dei punti della scena dalla telecamera. Tali distanze possono essere utilizzate per costruire il modello tridimensionale della scena. Se il movimento è una traslazione lungo l'asse z , le curve del flusso ottico saranno rette convergenti in un punto, che viene detto "fuoco di espansione" o "fuoco di contrazione" se il sensore si allontana dalla scena. Facendo fare alla telecamera un movimento traslativo lungo l'asse y , durante il quale vengono acquisite un insieme di immagini della scena, se si conosce il fuoco di espansione del movimento e se tutti i movimenti della telecamera sono noti, è possibile ricostruire la distanza tra i punti della scena e la telecamera stessa. Questa ultima tecnica si utilizza nella stereovisione di seguito riportata, anche se risulta essere meno precisa.
- *Shape from stereo*. Essendo la più usata ed interessante, anche perché è quella che più si avvicina alla visione umana, ne daremo di seguito un approfondimento.

STEREOVISIONE

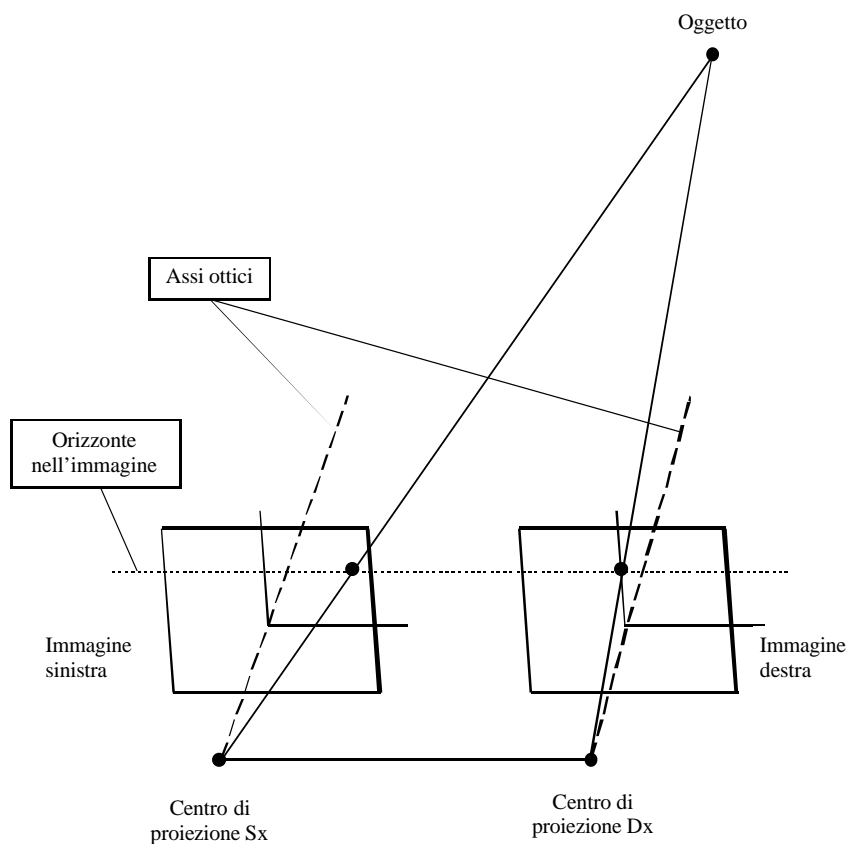
E' noto che la profondità di una scena ci risulta evidente in quanto i nostri occhi vedono da punti di vista leggermente diversi. Gli occhi sono leggermente convergenti e i loro assi visivi si incontrano in un punto. La proiezione di questo punto su ognuna delle due retine definisce i "centri di visione" delle stesse. Per cui, qualunque altro punto appartenente al campo visivo, sarà proiettato su ogni retina a una certa distanza dal centro di visione. La differenza di questa distanza per i due occhi viene detta disparità binoculare (nell'uomo è di circa 65 mm), ed è legata alla profondità del punto osservato.

Il processo di ricostruzione stereoscopica nella computer vision consiste nel ricavare la tridimensionalità della scena, a partire dalle disparità binoculare.

Questa tecnica di ricostruzione è ottenuta attraverso l'utilizzo di due identici elementi di acquisizione (fotocamere) o contemporaneamente o attraverso l'utilizzo di un solo dispositivo che effettui due acquisizioni consecutive da due posizioni diverse.

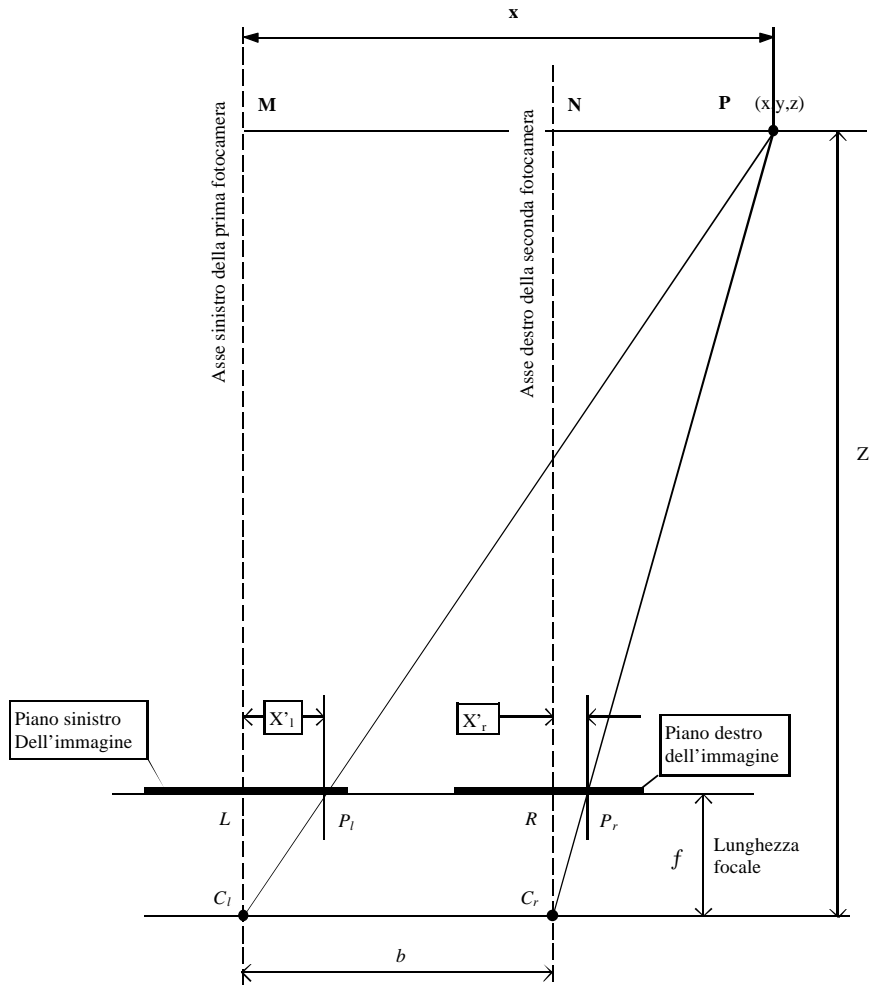
Per ricavare la profondità di un oggetto da due immagini che lo ritraggono si può procedere dando alcune definizioni relative al caso generico.

Consideriamo il caso di visione contemporanea come in figura:



Esempio di come un oggetto è visto in posizioni differenti da due punti di riferimento distinti

Gli assi ottici o focali sono quelle rette che partono dai centri di proiezione e si dirigono verso un punto all'infinito, restando ortogonali al piano dell'immagine destra o sinistra rispettivamente.



P_l e P_r sono le posizioni (l =sinistro r =destro) in cui il punto $P(x,y,z)$ viene proiettato sui due piani immagine.

Queste coppie di punti sono dette *punti coniugati*.

c'_l e c'_r sono rispettivamente le distanze tra gli assi ottici e le relative posizioni in cui si trova l'oggetto proiettato.

La proiezione di un punto sull'oggetto genera, nelle due immagini destra e sinistra, due punti che si trovano in posizioni diverse rispetto agli assi coordinati delle singole immagini.

Lo spiazzamento relativo a due punti coniugati è chiamato *disparità*. Tale disparità può essere misurata sovrapponendo le due immagini.

Nelle figure il punto P è osservato dai punti P_l e P_r rispettivamente nei piani sinistro e destro delle immagini. Senza perdere in generalità si può assumere che l'origine degli assi coordinati coincida con la lente sinistra e comparando i triangoli simili PMC_l e P_lC_l si ottiene che :

$$\frac{x}{z} = \frac{x_l'}{f}$$

dove f è la focale degli obbiettivi.

Similmente per i triangoli PNCr e PrRCr si ottiene che :

$$\frac{x-b}{z} = \frac{x_r'}{f}$$

combinando a questo punto le due equazioni si può ottenerne la distanza :

$$z = \frac{bf}{(x_l' - x_r')}$$

dove b è la distanza tra gli assi ottici

Per la natura del processo di acquisizione digitale, l'accuratezza della misura della disparità è vincolata al passo di campionamento dei pixel. Quindi per meglio misurare la distanza è possibile aumentare il valore di b distanziando maggiormente i due dispositivi di acquisizione. Questo atteggiamento introduce però altri problemi. Aumentando la distanza aumenta l'angolo di inclinazione tra l'asse ottico e l'asse che va dal centro della lente al centro dell'oggetto e di conseguenza diminuisce il numero di punti dell'oggetto che sono visibili su entrambe le immagini (punti coniugati). Inoltre le porzioni di immagine che contengono i punti coniugati essendo viste da posizioni diverse, appaiono differenti. In questo caso i punti coniugati risultano di difficile riconoscimento.

Queste difficoltà possono essere superate cercando di individuare, attraverso opportuni algoritmi, le porzioni di immagine che sono corrispondenti. Un primo approccio può essere l'utilizzo degli algoritmi di edge detection già visti, debitamente modificati. Questi algoritmi presentano però un drastico calo di prestazioni quando l'oggetto inquadrato non presenta contorni ben definiti o ben contrastati. Alla luce di queste problematiche sono stati sviluppati degli altri algoritmi per ricercare i cosiddetti punti interessanti (*interesting points*). Questi punti devono essere punti di facile individuazione, indipendentemente dalla direzione dalla quale è stata effettuata la acquisizione.

Punti siti in aree di colore e caratteristiche uniformi, chiaramente, non sono buoni candidati a diventare interesting points. Lo sono i punti caratterizzati da una elevata varianza nelle caratteristiche locali, quali angoli o punti di intersezione multipla.

Il problema della Corrispondenza

Note le coordinate in pixel di un punto di interesse in una delle due viste della scena, si vuole trovare nell'altra immagine la posizione del punto corrispondente, ovvero del punto che sia immagine dello stesso punto dello spazio.

Un solo pixel, la maggioranza delle volte, non contiene sufficiente informazione tale da permettere di trovarne la corrispondenza, e quindi, in generale viene considerato un intorno (rettangolare) di tale pixel (template), e si cerca la corrispondenza di quest'ultimo piuttosto che del singolo pixel. Una cosa da notare è che non tutti i pixel (e relativi intorni) sono adatti a risolvere il problema della corrispondenza: un'area a luminosità costante non contiene sufficiente informazione da permettere la risoluzione dell'ambiguità intrinseca al problema.

La ricerca della corrispondenza può essere vista in maniera più generale come un problema di *Signal Matching* ovvero di ricerca del segnale corrispondente (più simile) ad un dato segnale e in particolare di *Image Matching*.

Date due immagini, o porzioni di immagini, I' e I'' , siano P' e P'' due punti di coordinate (r', c') e (r'', c'') in I' e I'' rispettivamente. Tale coppia di punti può assumere solo uno di due possibili stati: possono essere punti corrispondenti, oppure non esserlo.

Se (r', c') e (r'', c'') sono le coordinate di punti corrispondenti, supponiamo soddisfino la seguente relazione:

$$(r', c') = TG(r'', c'', p_G)$$

dove TG è una particolare mappa funzionale che racchiude la conoscenza sulla relazione geometrica tra le due immagini, e p_G è l'insieme di parametri incogniti di tale mappa.

L'intensità di un'immagine può essere messa in relazione con quella dell'altra immagine da una relazione del tipo:

$$g' = T_1(g'', p_1)$$

dove T_1 è una mappa funzionale che esprime la conoscenza della relazione fra le intensità e p_1 è l'insieme di parametri incogniti di tale mappa.

Ciò porta al seguente modello completo e più generale per il problema di signal (image) matching:

$$g'(r', c') = T_1\{g''[T_G(r'', c'', p_G)]; p_1\}$$

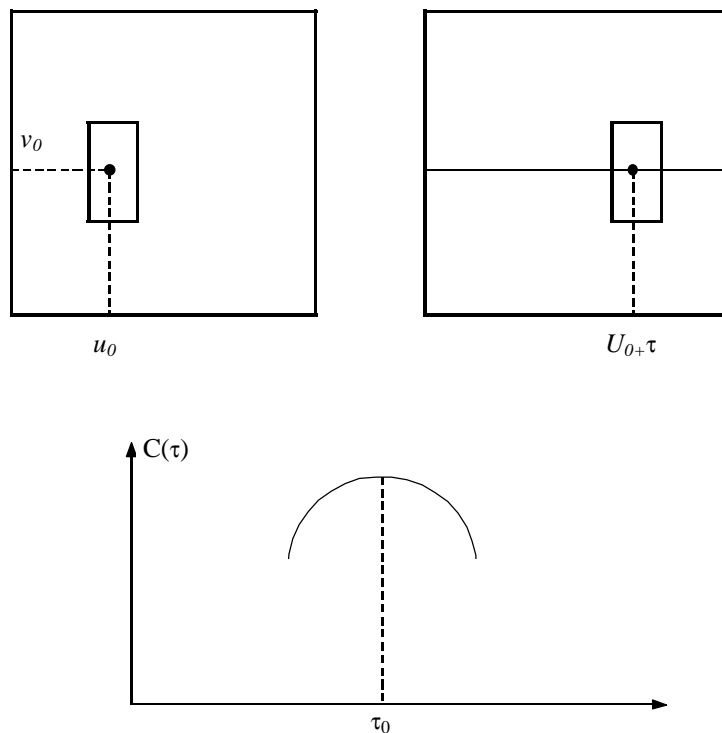
Il problema di image matching può quindi essere separato in due parti:

- a. trovare tutti i punti corrispondenti;
- b. determinare i parametri incogniti p_G e p_I delle due mappe T_G e T_I .

Si noti che la soluzione della prima parte solitamente non implica necessariamente la soluzione anche della seconda, mentre risolvendo la seconda può essere ricavata anche la soluzione del punto a, applicando la mappa T_G a tutte le posizioni. In letteratura sono presenti, varie tecniche che possono essere usate per la risoluzione del problema, noi vedremo la *Cross Correlation Matching*

Cross Correlation Matching

Il principio della tecnica di correlazione è mostrato nella figura sottostante. Si supponga di avere un pixel di coordinate u_0, v_0 nella prima immagine e di voler trovare lo stesso nella seconda immagine. Consideriamo ora un rettangolo di dimensione $(2P+1) \times (2N+1)$ centrato in (u_0, v_0) e eseguiamo la correlazione $C_{12}(\mathbf{t})$ con i valori d'intensità della seconda immagine sulla riga $v_2=v_0$.



La formula che ci permette di fare questo confronto è la seguente:

$$C_{12}(\mathbf{t}) = \frac{1}{K} \sum_{u_1=-N}^{+N} \sum_{v_1=-P}^{+P} (I_1(u_1 + u_0, v_1 + v_0) - \overline{I_1(u_0, v_0)}) (I_2(u_1 + u_0 + \mathbf{t}, v_1 + v_0) - \overline{I_2(u_0 + \mathbf{t}, v_0)})$$

dove

$$K = (2N + 1)(2P + 1) \mathbf{s}_1(u_0, v_0) \mathbf{s}_2(u_0 + \mathbf{t}, v_0)$$

In questa formula $\overline{I_1(u_0, v_0)}$ e $\mathbf{s}_1(u_0, v_0)$ sono l'intensità media e la deviazione standard della prima immagine nel punto (u_0, v_0) .

$$\overline{I_1(u_0, v_0)} = \frac{1}{(2N + 1)(2P + 1)} \sum_{u_1=-N}^{+N} \sum_{v_1=-P}^{+P} I_1(u_1 + u_0, v_1 + v_0)$$

$$\mathbf{s}_1^2(u_0, v_0) = \frac{1}{(2N + 1)(2P + 1)} \sum_{u_1=-N}^{+N} \sum_{v_1=-P}^{+P} (I_1(u_1 + u_0, v_1 + v_0) - \overline{I_1(u_0, v_0)})^2$$

In maniera analoga si procede per $\overline{I_2(u_0 + \mathbf{t}, v_0)}$ e $\mathbf{s}_2(u_0 + \mathbf{t}, v_0)$.

Cosicché la curva $C_{J_2}(\mathbf{t})$ normalmente ha un solo massimo e questo massimo si trova in corrispondenza del valore τ_0 di scostamento dalla posizione originale del pixel di partenza all'interno della seconda immagine.

Misura dell'Informazione di un Template

Come abbiamo già detto non tutti punti selezionabili in un'immagine sono buoni punti per poterne cercare la corrispondenza. Ad esempio, se si seleziona un punto su di una parete bianca è praticamente impossibile trovarne l'esatta corrispondenza nell'altra immagine in quanto, in linea di principio, qualsiasi punto della parete, o al più tutti i punti della parete che giacciono sulla retta epipolare associata al punto selezionato, sono ottimi candidati per il match in quanto il coefficiente di correlazione sarà prossimo ad 1 per ciascuno di essi.

Un punto deve essere sufficientemente distinguibile dal suo intorno per poter essere un buon template per la corrispondenza, ovvero il template associato (una finestra rettangolare centrata nel punto selezionato) deve contenere sufficiente informazione.

Una buona misura dell'informazione contenuta in una finestra si può estrapolare dall'algoritmo di Tomasi e Kanade. Nell'ambito della problematica di tracking¹ di punti attraverso sequenze di immagini, Tomasi e Kanade hanno trovato necessario introdurre il concetto di feature buona, nel senso che fosse facilmente inseguibile attraverso i vari fotogrammi. Solitamente in letteratura si considera una nozione a priori sulla bontà di una feature, indipendente dal metodo utilizzato per inseguirle: si considerano buone feature solitamente i punti associati ad angoli, bordi, ecc.

Come per la corrispondenza, neppure il tracking si può fare su semplici punti, ma si fa piuttosto usando delle finestre. Quello che si cerca è un modo per vedere se una finestra contiene sufficiente variazione di intensità (texture).

¹ Tracking cioè seguire un punto che si muove

Si può supporre che tra due frame successivi l'immagine cambi poco, e quindi detta I l'immagine all'istante t e J quella al tempo $t+1$, quest'ultima può essere modellata con un semplice modello lineare:

$$J(y) = I(x - d) = I(x) - g \cdot s \qquad g = \nabla I(x)$$

Ovvero come lo sviluppo in serie di Taylor troncato al primo ordine e centrato in x di $I(x)$. In tal modo il problema di tracking si riduce a trovare lo piazzamento bidimensionale s , che minimizza in norma quadratica la distanza tra J e il modello. La soluzione è data dal vettore d che risolve il sistema lineare di due equazioni in due incognite:

$$Gs = e$$

Con

$$G = \int_{w \in W} g \cdot g^T \cdot w \cdot dW \qquad e = \int_{w \in W} (I - J)g \cdot w \cdot dW$$

dove gli integrali sono calcolati su di una finestra W , centrata in x .

Affinché la soluzione delle equazioni sopra elencate sia stabile numericamente la matrice simmetrica G deve essere ben condizionata e i suoi coefficienti devono essere sufficientemente grandi rispetto al rumore sovrapposto all'immagine. Le suddette richieste su G si traducono nel criterio di selezione delle feature buone, ovvero si considerano buone feature quelle per cui il sistema è ben condizionato.

Dal punto di vista pratico queste condizioni sono equivalenti a richiedere che i due autovalori di G siano sufficientemente grandi (robustezza rispetto al rumore) e non troppo diversi (buon condizionamento numerico). Poiché l'intensità di un immagine è un segnale ad ampiezza limitata, generalmente quando l'autovalore minore di G è sufficientemente grande, G è anche ben condizionata.

Quindi, detti λ_1 e λ_2 i due autovalori di G , una finestra contiene sufficiente informazione, ovvero ne può essere fatto facilmente il tracking oppure, dal nostro punto di vista, la corrispondenza, se

$$\min(\lambda_1, \lambda_2) > \lambda$$

la soglia λ si può scegliere come valore medio tra gli autovalori misurati in una zona a colorazione uniforme ed una ad elevata tessitura.

L'Algoritmo di Tomasi-Kanade per il problema della corrispondenza

Si supponga di aver selezionato un punto m in una delle immagini. Si vuole determinare se di tale punto se ne può fare agevolmente la corrispondenza. Per quanto detto, ciò è equivalente al fatto che l'autovalore minimo della matrice G calcolata in un intorno W del punto sia sufficientemente maggiore della soglia λ .

Il realtà piuttosto di concentrarsi sul solo punto m appena selezionato, conviene considerare un intorno X di tale punto e controllare se gli altri punti contenuti in X soddisfano la condizione, magari in maniera migliore di m .

Si considera quindi una finestra rettangolare X centrata nel punto m , di dimensione $k \times k$. Per ognuno dei punti $m_{ij} \in X$, con $i, j = -k/2, \dots, k/2$ si calcola la matrice G_{ij} su di una finestra W di dimensione $s \times s$, con $s < k$. Di ognuna delle G_{ij} si calcola l'autovalore minimo², e si seleziona come migliore punto del quale cercare la corrispondenza il punto m_{ij} , per il quale il corrispondente autovalore minimo λ_{\min} di G_{ij} è massimo tra quelli maggiori della soglia λ .

Se nessuno dei punti X soddisfa al criterio di selezione, prima di marcare il punto come non adatto alla corrispondenza, è possibile usare una delle seguenti alternative:

- a. si può aumentare la dimensione della finestra X e ripetere il processo di selezione;
- b. si può sottocampionare l'immagine di partenza e ripetere il processo di selezione su un intorno X di m della stessa dimensione di X .

La prima tecnica se ripetuta molte volte può diventare piuttosto costosa in termini di tempo di esecuzione, e risulta quindi preferibile la seconda, anche se può portare alla selezione di punti piuttosto distanti da m . Ma di questo si può eventualmente tenerne conto limitando la distanza tra m ed il punto effettivamente selezionato per la ricerca della corrispondenza.

² Si noti che tale autovalore è esprimibile in forma chiusa in funzione degli elementi di G :

$$\lambda_{\min} = \frac{1}{2} \left\{ g_{11} + g_{22} - \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2} \right\}$$

Restrizioni per la ricerca della Corrispondenza

La ricerca della corrispondenza tramite il coefficiente di correlazione è, in generale, una ricerca esaustiva bidimensionale in tutta l'immagine. Si è notato come nel caso di due viste stereoscopiche normali, i punti corrispondenti giacciono sulle medesime righe dell'immagine ed è quindi sufficiente una ricerca solamente unidimensionale, riducendo notevolmente la complessità del problema. Tale situazione non è un caso particolare, ma è dovuta ad un vincolo intrinseco dei sistemi di visione stereoscopica, qualsiasi sia la posizione relativa delle telecamere.

Questo non è il solo vincolo presente in un sistema stereoscopico. In generale si possono distinguere tre classi di vincoli :

- a. Vincoli geometrici imposti dal sistema di visione. Sono probabilmente i più importanti, e grazie a questi si può ridurre una ricerca bidimensionale ad una ricerca unidimensionale (vincolo epipolare).
- b. Vincoli geometrici imposti dagli oggetti osservati. Si può assumere, ad esempio, che la loro distanza dal sistema di visione vari lentamente praticamente ovunque (vincolo sul gradiente della disparità). Un altro vincolo di questo tipo si ha quando l'oggetto che si osserva è assimilabile ad un poliedro.
- c. Vincolo fisici imposti dai modelli dell'interazione degli oggetti con l'illuminazione.

Tutti questi vincoli si possono usare per diminuire l'ambiguità intrinseca del problema della corrispondenza.

La restrizione Epipolare

Quella che vediamo ora è una delle restrizioni per la ricerca di punti coniugati più importante ed utilizzata nella stereovisione. Questa operazione serve per ridurre lo spazio di ricerca di un punto m_2 coniugato ad un punto m_1 preso in un immagine R_1 .

Per spiegare come funziona l'algoritmo ci serviremo della figura sottostante.

In questa figura possiamo vedere che m_1 , presente nel piano R_1 , è proiezione di un punto M . La posizione di m_1 è ottenuta facendo incidere sul piano R_1 la semiretta che parte dal punto M e passa per il centro di proiezione C_1 . Lo stesso discorso vale per il piano R_2 , la proiezione di M attraverso C_2 darà m_2 .

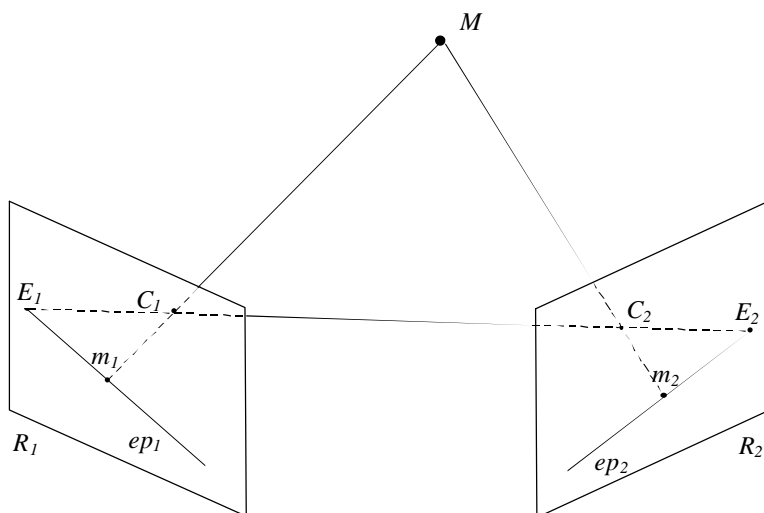


Figura che ritrae il caso di piani con assi ottici non paralleli

Quest'ultima proiezione sarà però un insieme infinito di punti che percorrerà una retta ep_2 sulla quale giace anche E_2 . E_2 è l'intersezione tra il piano R_2 e la retta che congiunge i due centri di proiezione $\langle C_1, C_2 \rangle$, questo punto viene chiamato punto epipolare della seconda camera rispetto alla prima, e la retta ep_2 è chiamata retta epipolare del punto m_1 sulla piano di proiezione R_2 della seconda camera. Questo vincolo, cioè la possibilità di trovare la proiezione m_2 del punto M solo su di una retta riduce lo spazio di ricerca da due dimensioni ad uno spazio in una sola dimensione.

Questa tecnica è simmetrica ciò vuol dire che si può partire prima da m_2 e poi fare al ricerca di m_1 sulla retta epipolare ep_1 . Le linee ep_1 ed ep_2 sono l'intersezione del piano C_1MC_2 chiamato piano epipolare definito da M con i piani di proiezione R_1 ed R_2 .

Lo stesso discorso si può fare nel caso che i due piani di proiezione abbiano gli assi ottici paralleli. In questo caso i punti E_1 E_2 risultano non appartenenti ne a R_1 ne a R_2 (in quanto sono all'infinito) e le rette ep_1 e ep_2 possono essere trovate calcolando l'intersezione del piano definito dai tre punti C_1 M C_2 con R_1 e R_2

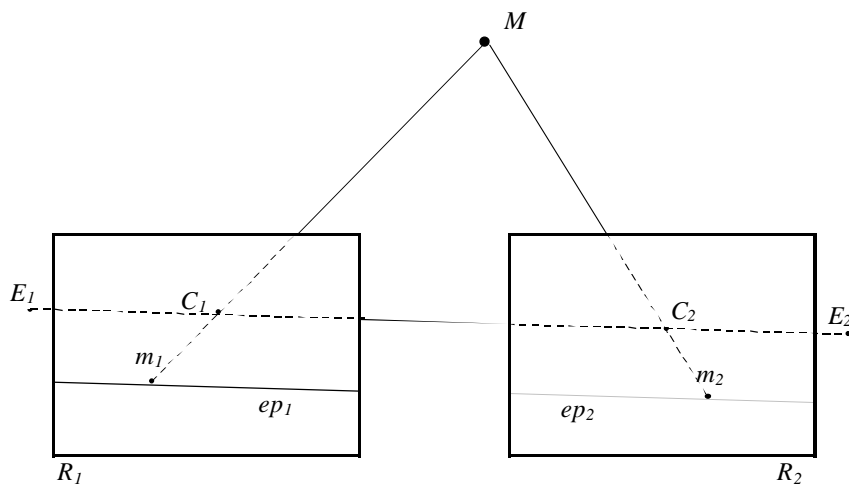
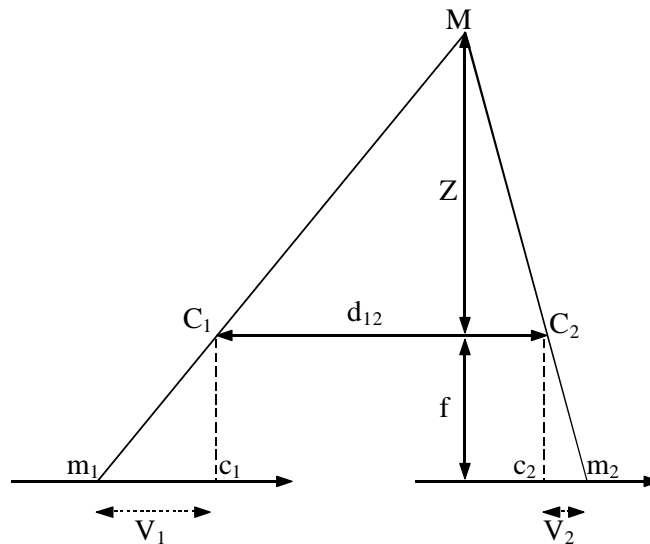


Figura che ritrae il caso di piani con assi ottici paralleli

Una definizione per la *Disparità*

Quando si ha a che fare con il sistema visivo umano (o un qualsiasi sistema stereoscopico equivalente, come le viste stereoscopiche normali), generalmente la disparità viene definita, con riferimento alla figura di seguito riportata, come $d=v_2-v_1$, dove (u_1,v_1) e (u_2,v_2) sono le coordinate nei rispettivi piani immagine, di m_1 e m_2 (si noti che $u_1=u_2$ perché i piani di proiezione sono paralleli). La geometria del sistema induce una semplice relazione tra la disparità d e la profondità z del punto M che si osserva data dall'equazione vista prima dove al posto di v_1 e v_2 sono stati usati X_1 e X_2 . L'equazione appena citata afferma che un qualsiasi piano di fronte alle camere è parallelo a quest'ultime è il luogo dei punti a disparità costante. Si noti che quando il punto M si avvicina ai punti di proiezione la disparità aumenta.



Relazione tra profondità e disparità con viste stereoscopiche normali.

Un Primo Vincolo sulla Disparità

L'idea alla base di questo vincolo è che il mondo (la scena che si osserva) è composta principalmente da oggetti descritti da superfici continue (o pressoché continue). Ciò significa che la cosiddetta funzione di ricostruzione che assegna ad una coppia di punti corrispondenti un punto M dello spazio 3D e storicamente definita come $z=f(d)$, deve essere smooth praticamente ovunque.

Quindi se M è un punto dello spazio 3D con proiezioni m_1 ed m_2 nelle due camere, e disparità d nel senso appena definito, un punto n_1 vicino a m_1 nella camera sinistra deve essere messo in corrispondenza con un punto n_2 nella camera destra tale che la loro disparità sia prossima a d . Ovvero se si assume che il mondo sia continuo, così deve essere anche la disparità dei punti osservati.

Il Gradiente della Disparità

Date due coppie di punti corrispondenti $(m_1; m_2)$ e $(n_1; n_2)$, il gradiente della disparità DG si definisce come:

$$DG = \left| \frac{\|m_1 - n_1\| - \|m_2 - n_2\|}{\frac{1}{2} [\|m_1 - n_1\| + \|m_2 - n_2\|]} \right|$$

Esperimenti di psicofisica hanno portato alla conclusione che generalmente il suo valore è limitato superiormente. La topologia delle immagini è ovunque preservata se le corrispondenze tra le due immagini danno origine ad un gradiente di disparità sempre inferiore a 2. Tale risultato è invariante per rotazioni, ingrandimenti e riduzioni, e vale in qualsiasi direzione e non solo lungo le rette epipolari (è isotropico).

Se $DG=2$ si è in presenza di una situazione degenera in cui un punto in uno dei piani immagine può essere messo in corrispondenza con due punti dell'altro piano immagine. In altre parole, si sta osservando una scena in cui due feature puntuali danno la stessa immagine in una delle camere (ovvero vengono proiettate nella medesima posizione, ad esempio $m_1=n_1$), mentre nell'altra camera producono immagini distinguibili. Tale situazione viola uno dei principi fondamentali della corrispondenza: un punto può partecipare solo ad una corrispondenza (Principio di Esclusione).

Il limite sul gradiente della disparità è spesso usato per risolvere l'ambiguità della corrispondenza in stereoscopia. Può essere formalmente dimostrato come l'imposizione $DG \leq 2k$, con $k < 1$, assicuri che non potranno essere fatte corrispondenze tra punti immagine di un punto nello spazio che è visibile ad una camera ma non all'altra. Si osservi inoltre, che in generale il contrario non è vero. Infatti l'imposizione di un limite sul gradiente della disparità si può considerare una politica conservativa: previene le false corrispondenze, ma può non permetterne alcune di vere.

Un valore minimo di $DG \approx 0$ è un vincolo molto forte permettendo il match solo di punti appartenenti ad un piano di fronte alle camere e parallelo a quest'ultime.

Il limite sul Gradiente della Disparità

Il limite sul gradiente della disparità ha origine da osservazioni di natura psicofisica sulla fusione binoculare³ operata dal sistema visivo umano.

E' stato dimostrato che, definito il gradiente della disparità tra due punti vicini come la differenza delle loro disparità divisa per l'angolo visivo di separazione, la fusione binoculare di almeno un punto falliva quando questo gradiente superava un valore critico (circa 1).

In Computer Vision non esiste il concetto di fusione binoculare ma, come si vedrà nel seguito, un vincolo analogo al precedente, definito in termini più generali, può essere usato in questo ambito per risolvere l'ambiguità del problema della corrispondenza in visione stereoscopica.

Questo materiale didattico è stato tratto principalmente dagli appunti redatti durante la porzione del corso di robotica riguardante la visione e da alcuni articoli pubblicati sulla rete internet.

Di questi articoli e web pages vengono di seguito riportati gli *url*:

Shape from shading:

- http://www.cs.usask.ca/projects/vision/results/Shape_from_shading.html
- <http://www.merl.com/people/freeman/sfs.html>
- <http://www.cssip.edu.au/~vision/shading.html>
- [http://www.dgp.toronto.edu/~jstewart/ Current Research on Shape-from-shading.htm](http://www.dgp.toronto.edu/~jstewart/Current%20Research%20on%20Shape-from-shading.htm)
- <http://www.psrw.com/~markc/Articles/SFS/sfs.html>

Shape from motion:

- <http://www.cs.cmu.edu/People/vaschelp/Seminars/Vascsem/031196.html>
- <http://cvs.anu.edu.au/zhang/shape.abs.html>
- <http://online.anu.edu.au/Physics/qt/perplengths.html>
- www.vision.caltech.edu/soatto/research.html

Shape from focus

- <http://computer.org/tpami/tp1995/i0266abs.htm>

³ La fusione binoculare è il fenomeno per il quale una coppia di immagini stereoscopiche appaiono come un'unica immagine