



# Corso di Percezione Robotica

## C. Modulo di Percezione Attiva

### Visione robotica

Eliseo Stefano Maini

ARTS Lab, Scuola Superiore Sant'Anna

[es.maini@ieee.org](mailto:es.maini@ieee.org)

[s.maini@arts.sssup.it](mailto:s.maini@arts.sssup.it)

050-883486



# Sommario della lezione

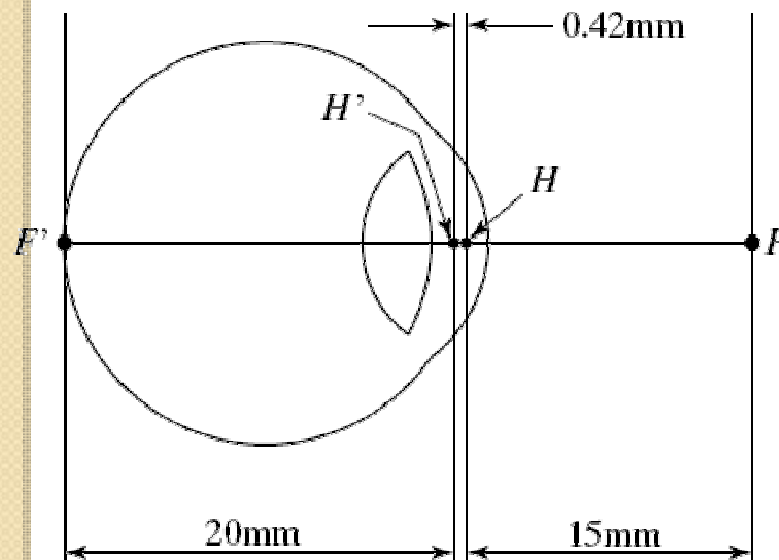
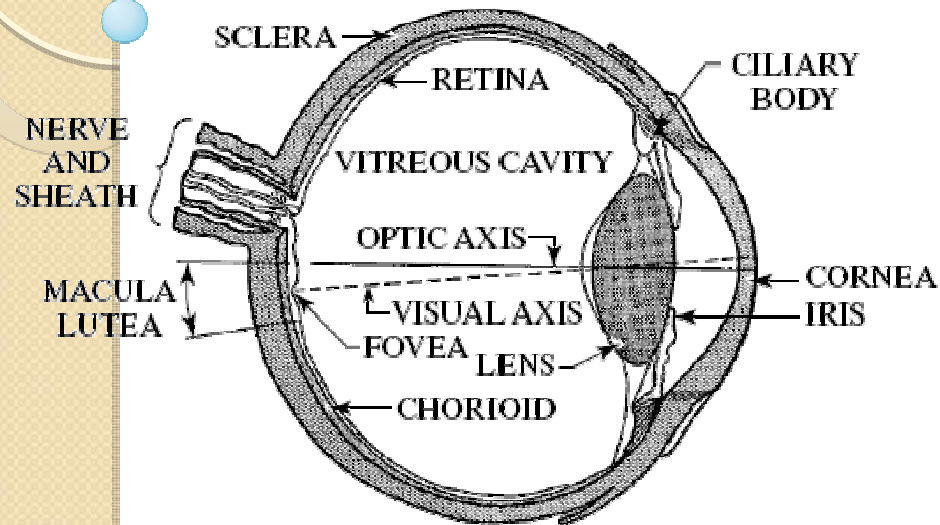
- **Formazione dell'immagine**
  - Modello della telecamera
  - Funzione immagine
- **Geometria delle immagini (binarie)**
  - Adiacenza e connettività
  - Componenti connessi
  - Distanze
- **Pre-elaborazione (early processing):**
  - Operazioni puntuali
  - Operazioni globali



# Materiale didattico

- *Forsyth, Ponce, “Computer vision: A modern approach”, Prentice Hall*
- *Fu, Gonzalez, Lee, “Robotica”, McGraw-Hill*
- *Ballard & Brown, “Computer vision”, Prentice Hall*
- *Lucidi lezione*

# Sensore biologico



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





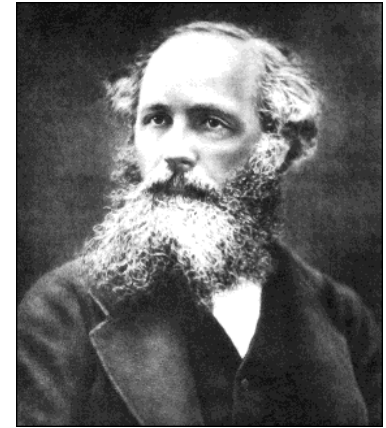
# Formazione dell'immagine

# La table servie



Nicéphore Niépce 1822

# The Tartan Ribbon



James Clerk Maxwell - 1862



Autochrome – Lumiere 1907

# La Sortie des Usines



**LE CINÉMATOGRAPHE**  
**SALON INDIEN**  
Grand Café  
*14, Boulevard Des Capucines, 14*  
**PARIS**

*Cet appareil, inventé par MM. Auguste et Louis Lumière, permet de recueillir, par des séries d'épreuves instantanées, tous les mouvements qui, pendant un temps donné, se sont succédé devant l'objectif, et de reproduire ensuite ces mouvements en projetant, grandeur naturelle, devant une salle entière, leurs images sur un écran.*

**SUJETS ACTUELS**

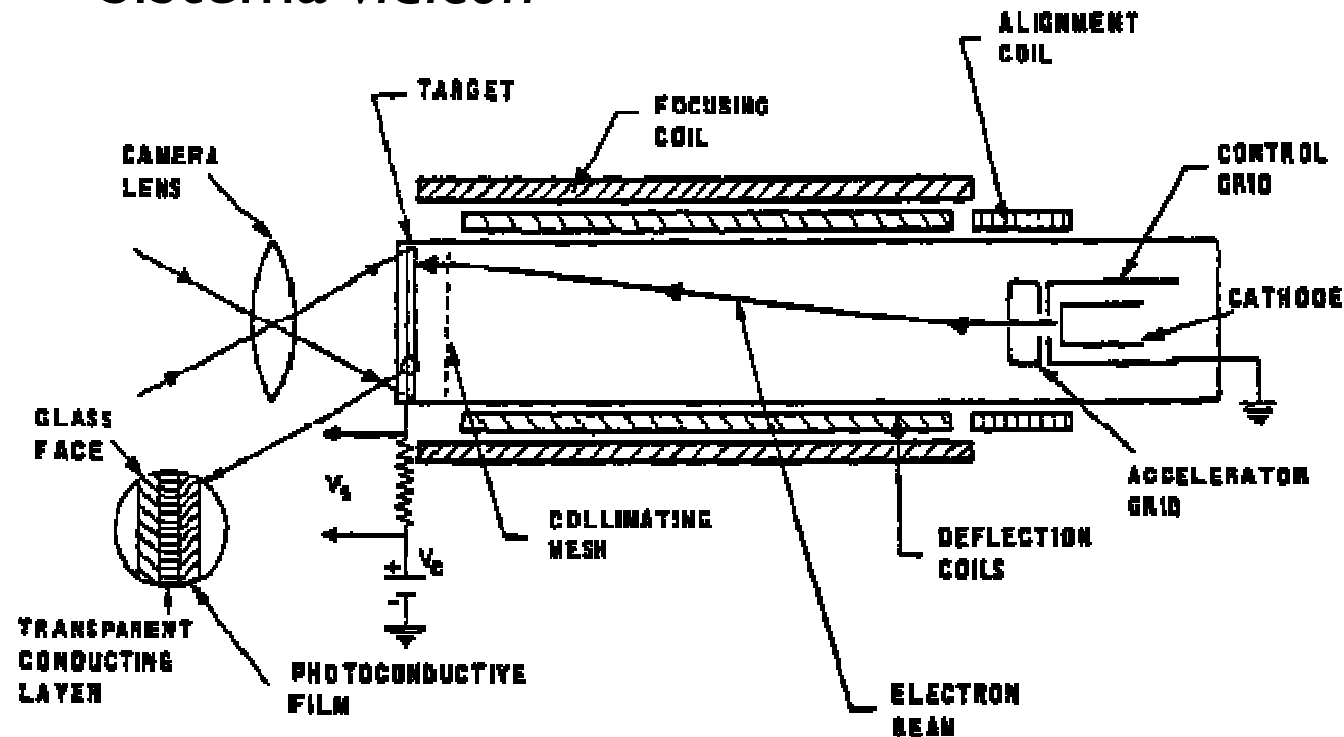
1. La Sortie de l'Usine LUMIÈRE à Lyon.	5. Les Forgerons.
2. La Pêche.	6. Le Jardinier.
3. La Pêche aux Poissons Rouges.	7. Le Regain.
4. Le Débarquement du Congrès de Photographie à Lyon.	8. Le Saut à la Couverture.
	9. La Place des Cordeliers à Lyon.
	10. La Mer.



Luis e Auguste Lumiere 28 Dicembre 1895

# Verso la modernità...

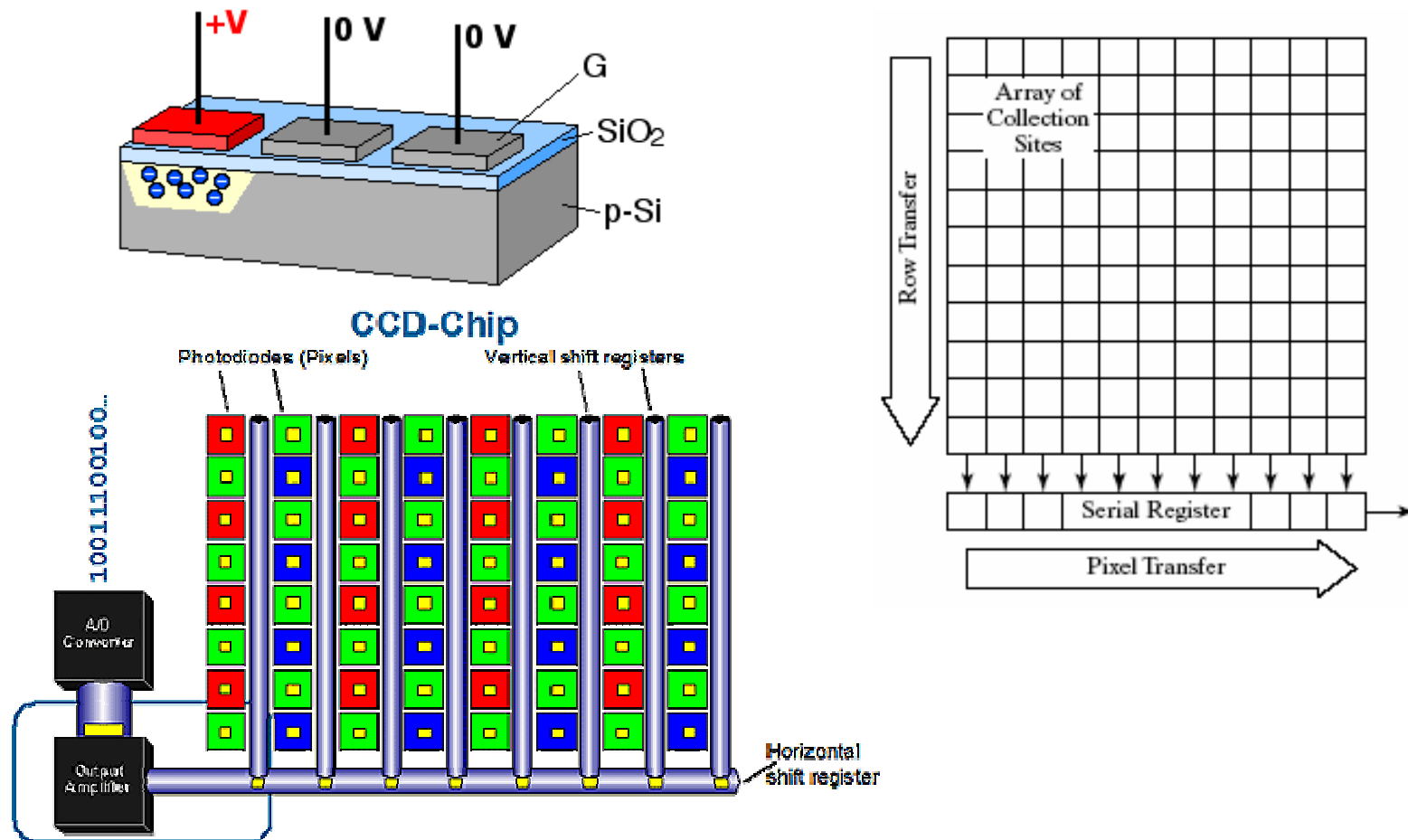
- Invenzione della televisione – circa 1920
  - Baird, Farnsworth, Zworykin
  - Sistema *vidicon*





# Sensori CCD - Charge-Coupled-Device

- 1969 Bill Boyle e George Smith



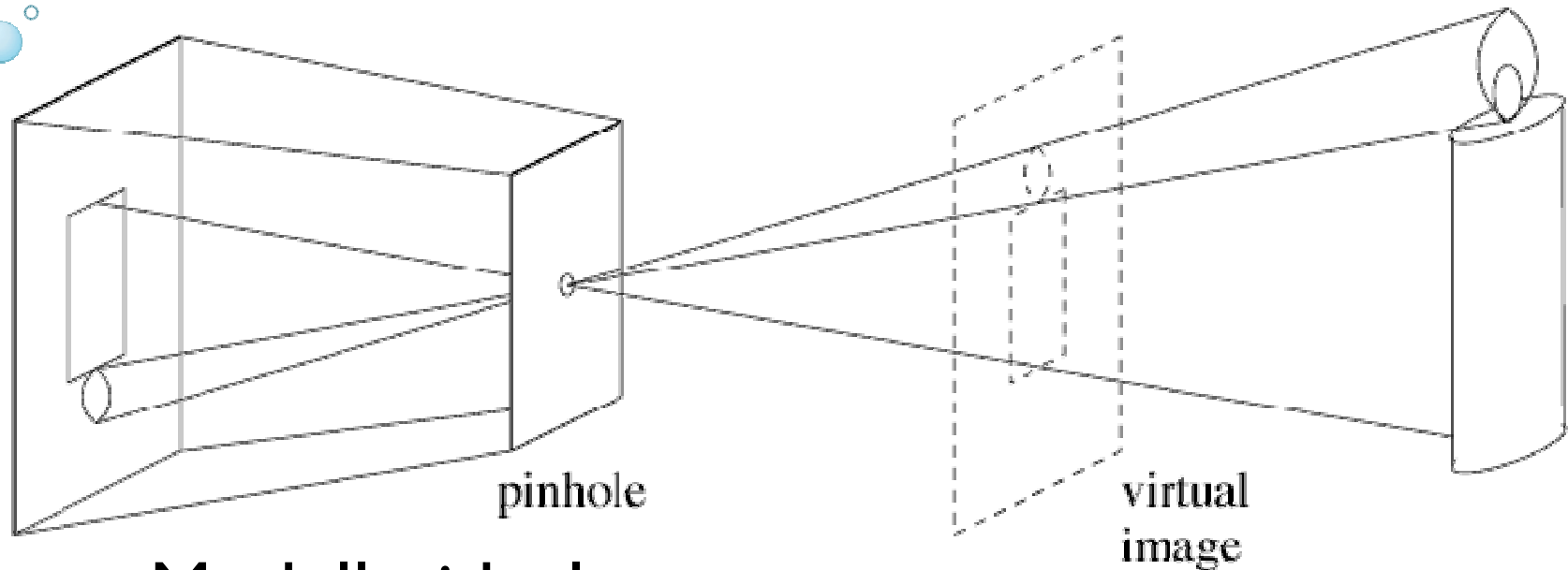


# Modelli di telecamera

- Pinhole camera (Brunelleschi XV secolo)
  - Modello di proiezione prospettica
  - Modelli di proiezione affine
    - Weak-perspective
    - Proiezione ortografica
- Telecamere con lenti
  - Modello delle lenti sottili
  - Modello delle lenti reali

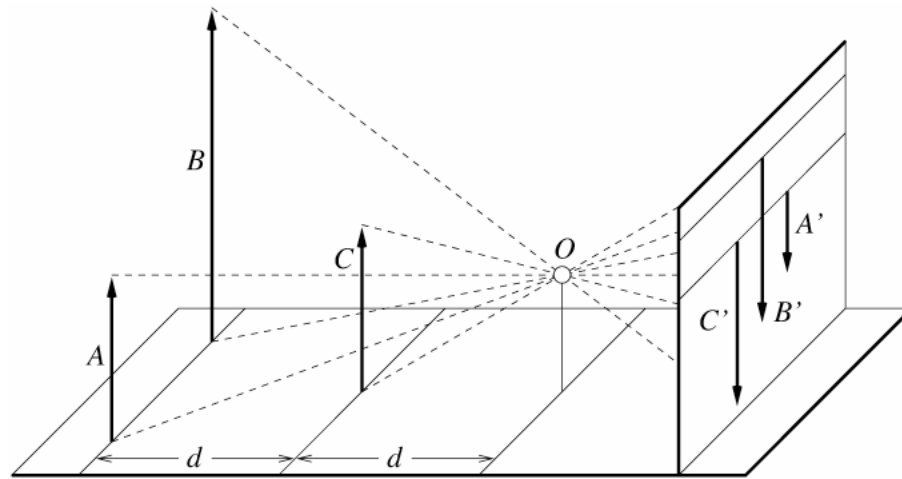


# Il modello pinhole



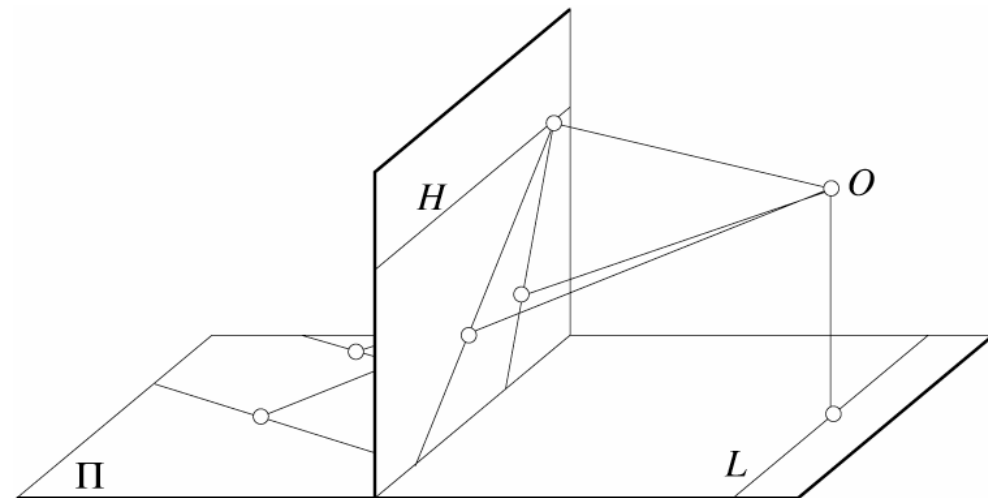
- **Modello ideale astratto**
  - Scatola con buco puntiforme
    - formazione immagine capovolta
    - piano virtuale
- **Buona approssimazione**

# Proiezione prospettica



- Riduzione della dimensione apparente all' aumentare della distanza

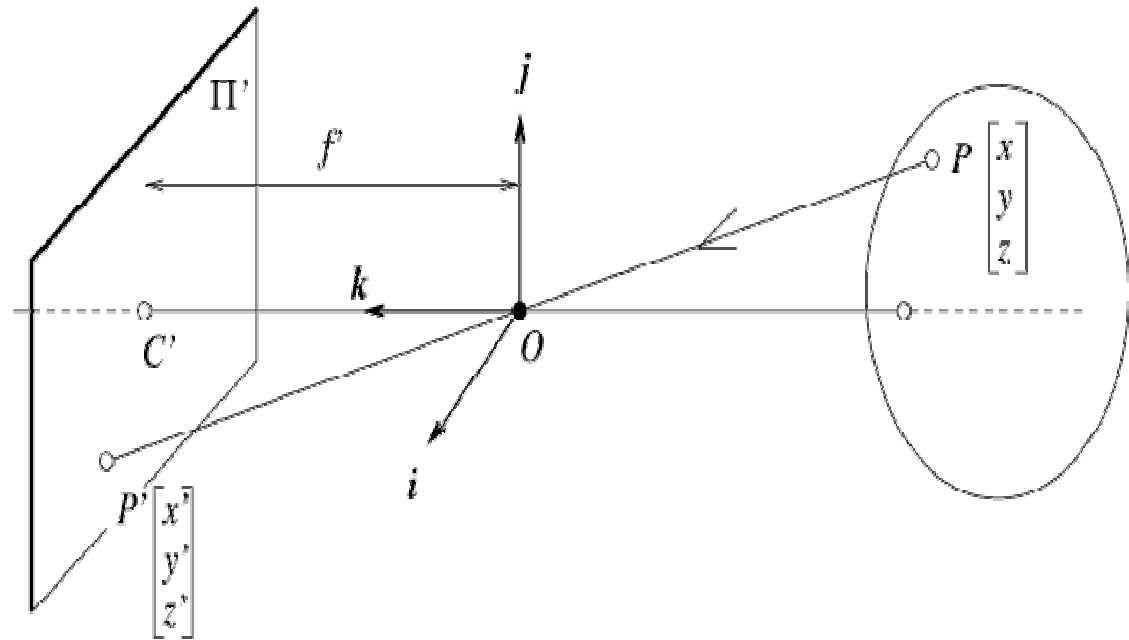
- Convergenza di rette parallele – linea di orizzonte



# Proiezione prospettica - Modello

## Definizioni

- $\Pi'$  = piano immagine
- $f'$  = lunghezza focale
- $OC'$   $k$  = asse ottico principale
- $C'$  = centro dell'immagine



$$z' = f'$$

P, O, P' collineari

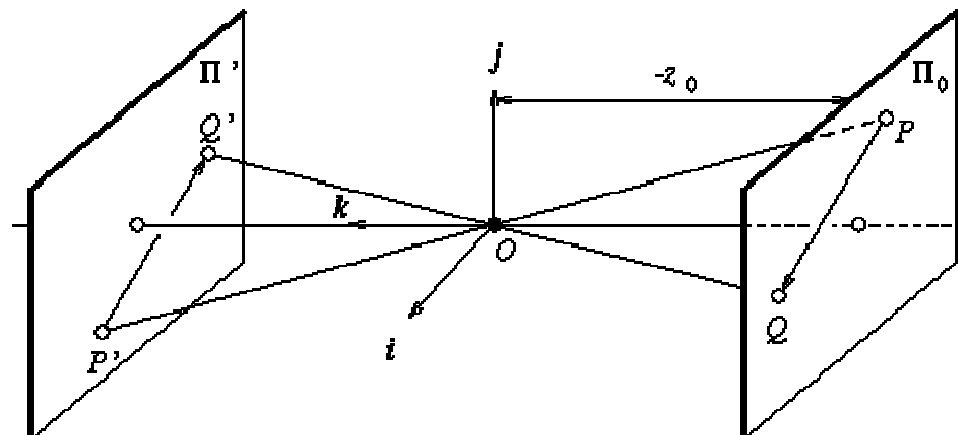
$$\Rightarrow \exists \lambda \text{ t.c. } \vec{OP'} = \lambda \vec{OP}$$

$$\begin{cases} x' = \lambda x \\ y' = \lambda y \\ f' = \lambda z \end{cases} \Leftrightarrow \lambda = \frac{x'}{x} = \frac{y'}{y} = \frac{f'}{z};$$

$$\begin{cases} x' = f' \frac{x}{z} \\ y' = f' \frac{y}{z} \end{cases}$$

# Proiezioni affini I

- Weak perspective

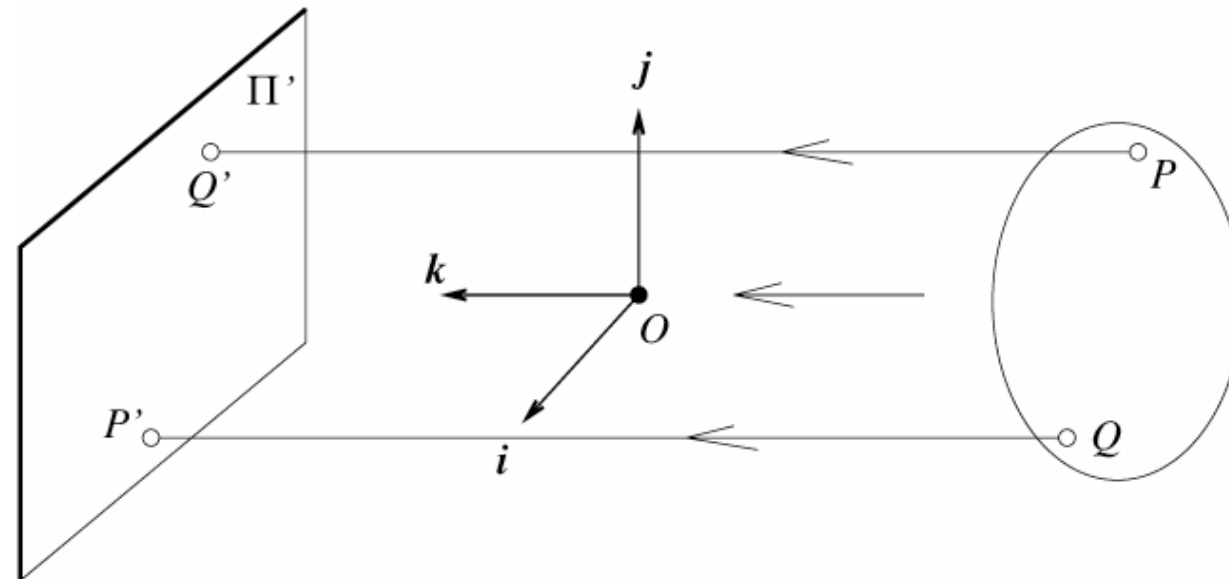


- HP: Profondità della scena piccola rispetto alla distanza media dalla camera
- Vantaggio: semplice
- Svantaggio: ipotesi restrittive

$$\begin{cases} x' = -mx \\ y' = -my \\ m = -f' / z_0 \end{cases}$$

# Proiezioni affini II

- Proiezione ortografica



- Magnificazione unitaria ( $m=1$ )
- Vantaggio: semplice
- Svantaggio: molto irrealistico

$$\begin{cases} x' = x \\ y' = y \end{cases}$$

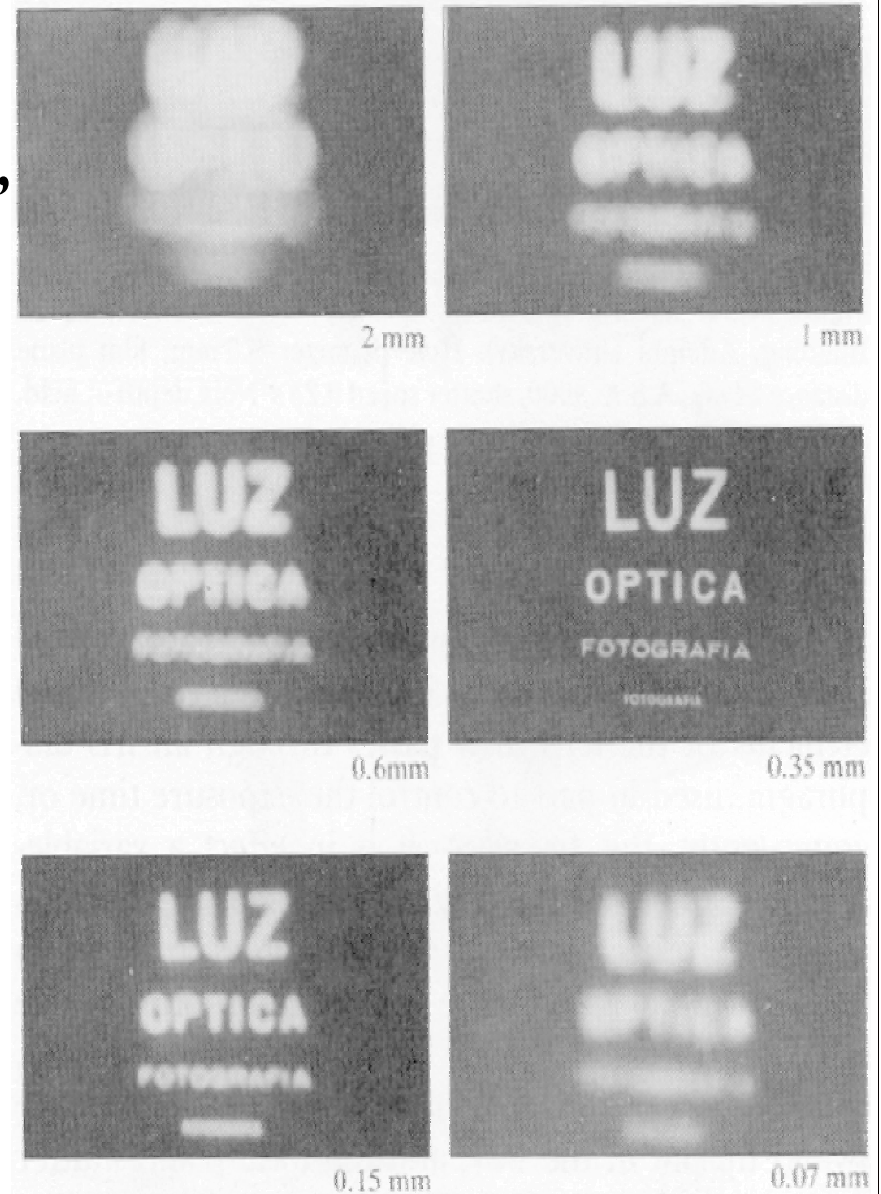
# Criticità del pinhole

- Sono telecamere “**scure**”

- solo una piccola frazione di raggi luminosi colpisce il sensore

- Dimensione del pinhole

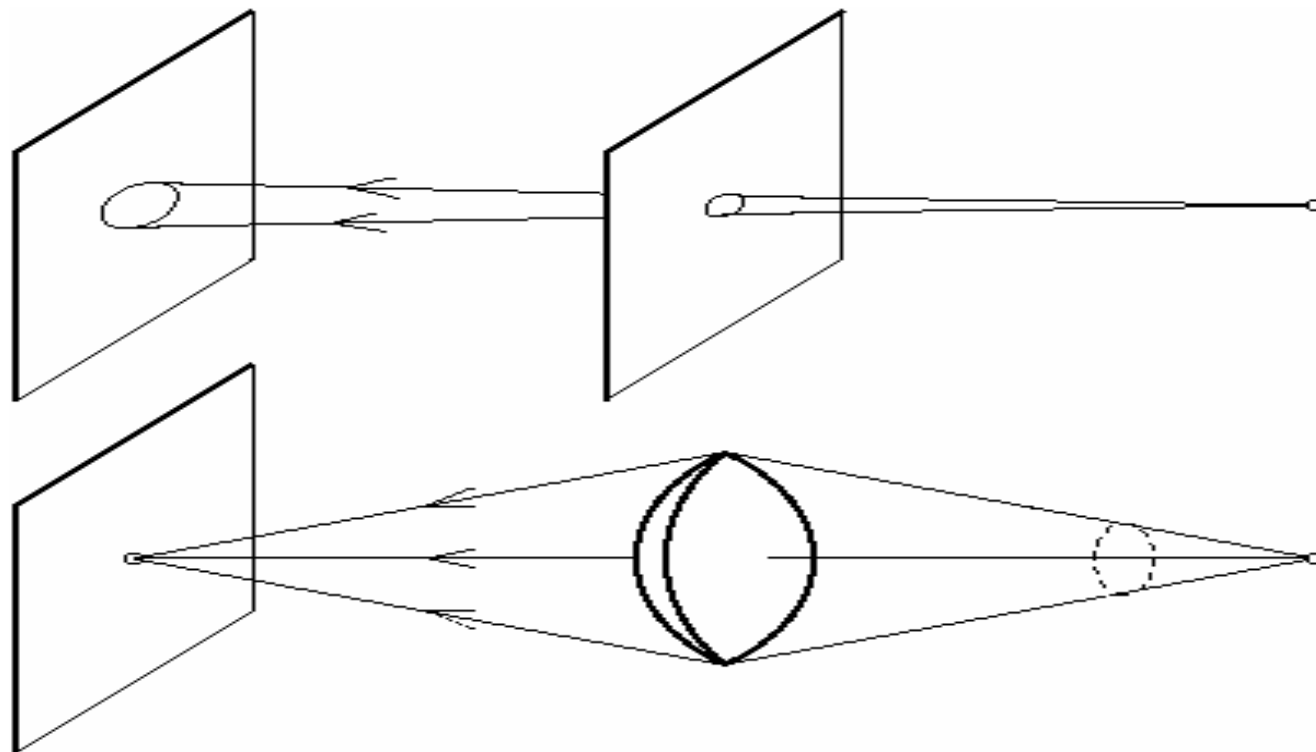
- Pinhole troppo grande
  - Sfocatura dell'immagine per l'averaging di raggi provenienti da troppe direzioni
- Pinhole troppo piccolo
  - Sfocatura dell'immagine per effetto della diffrazione





# Necessità delle lenti

- Aumentare il numero di raggi che colpisce il sensore mantenendo l'immagine a fuoco

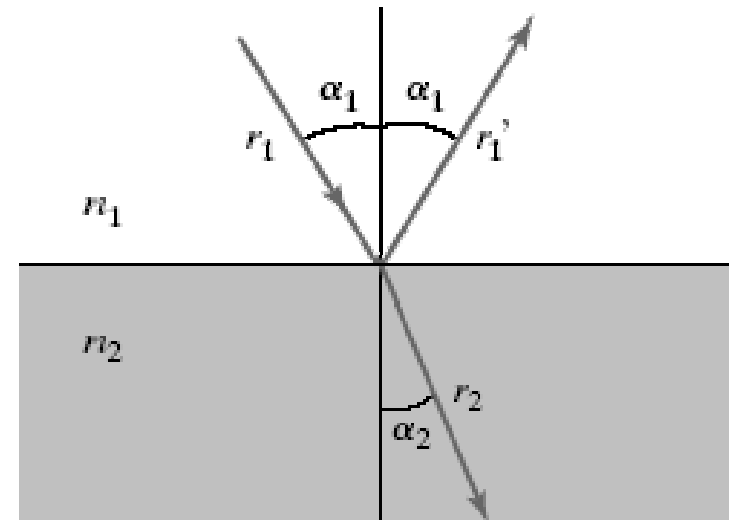




# Legge di Snell

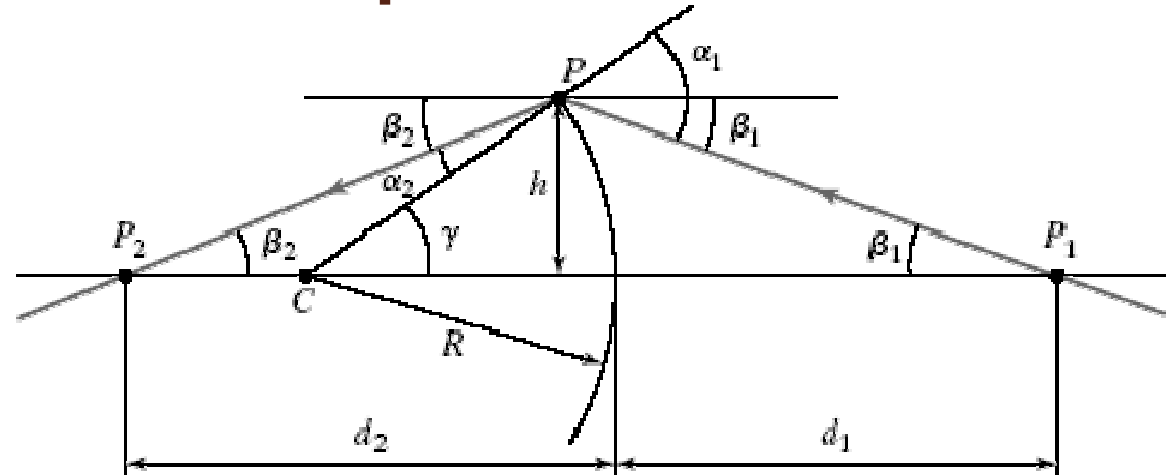
- Riflessione e rifrazione all'interfaccia tra due mezzi omogenei

$$n_1 \sin \alpha_1 = n_2 \sin \alpha_2$$



- $n_1$  indice di rifrazione mezzo 1
- $n_2$  indice di rifrazione mezzo 2

# Rifrazione parassiale

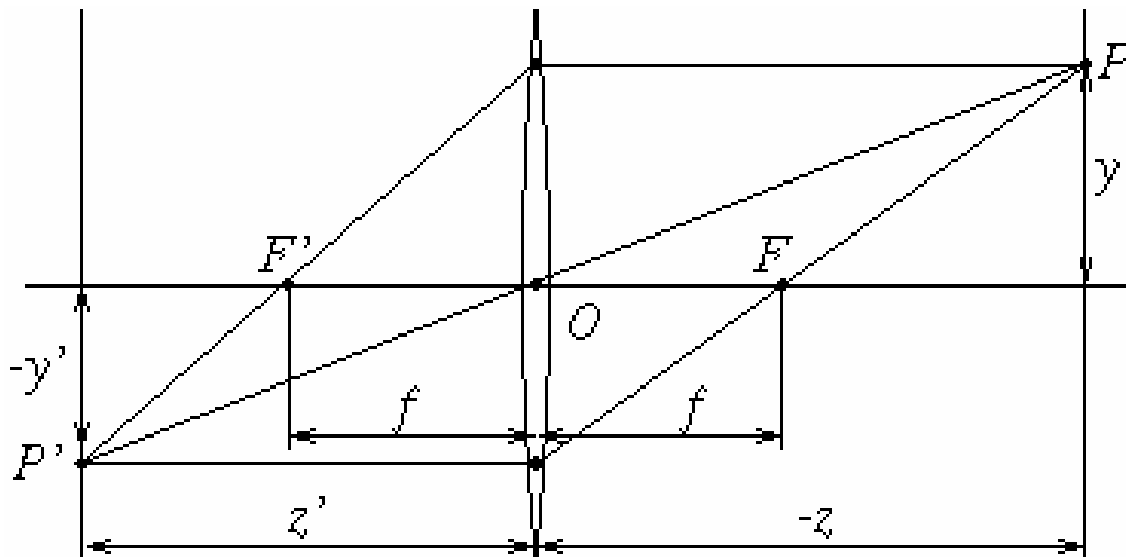


- Approssimazione valida per piccoli angoli di incidenza (approssimo il seno e la tangente con l'angolo)

$$\frac{n_1}{d_1} + \frac{n_2}{d_2} = \frac{n_2 - n_1}{R}$$

# Camera con lenti sottili

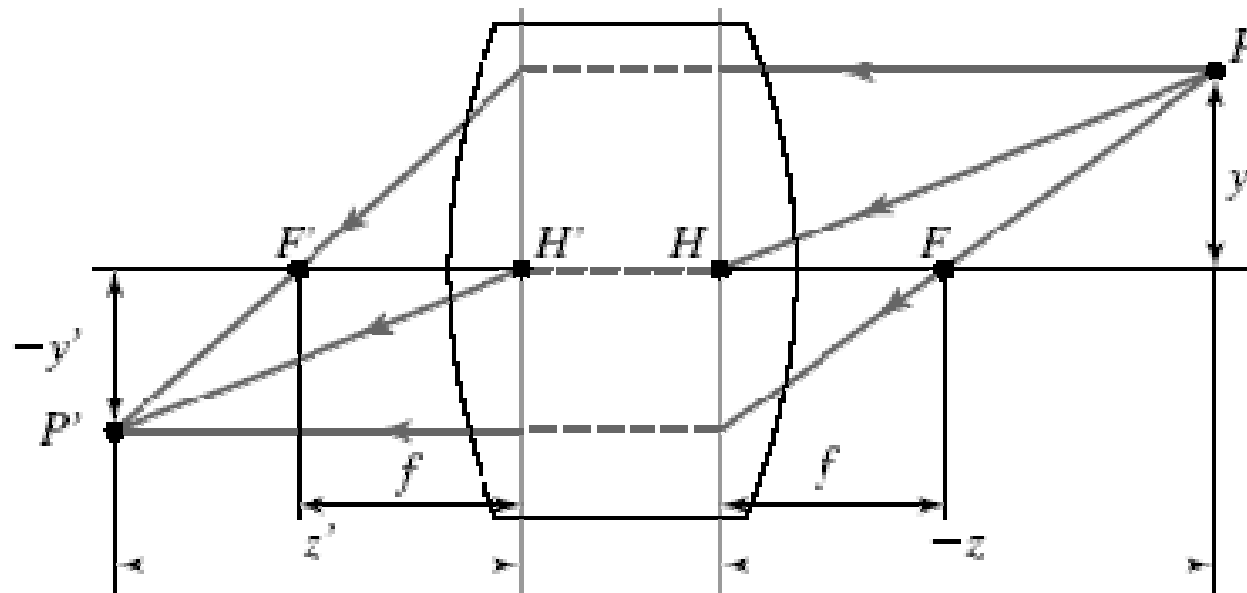
- Modello delle lenti sottili
  - Un raggio che entra nella lente ed è rifratto dall'interfaccia viene immediatamente nuovamente rifratto dalla seconda superficie della lente
  - Indice di rifrazione uguale ad 1



$$\frac{1}{z'} - \frac{1}{z} = \frac{1}{f}$$

# Camera con lenti reali

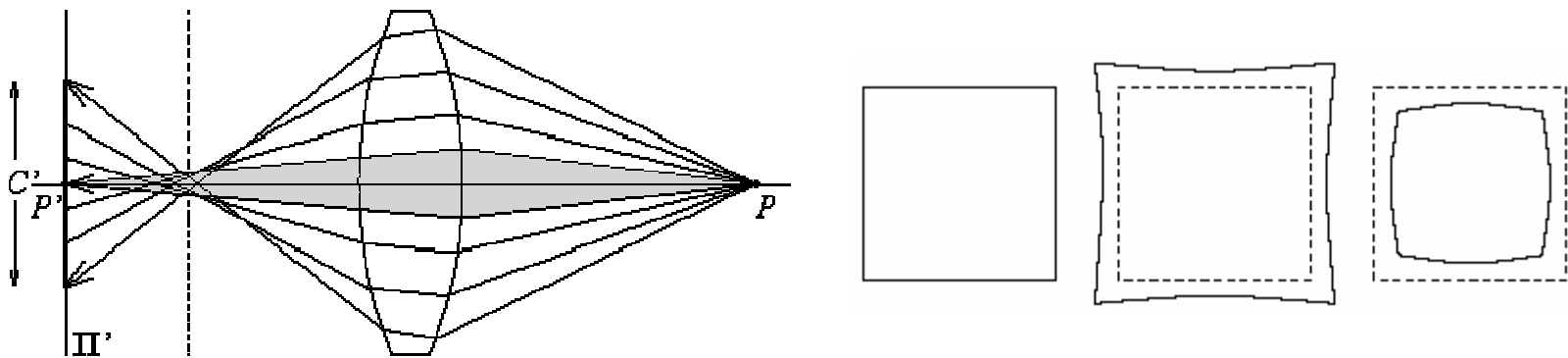
- Modello della lente reale
  - Non vale l'approssimazione parassiale



$$\frac{n_1}{d_1} + \frac{n_2}{d_2} = \frac{n_2 - n_1}{R} + h^2 \left[ \frac{n_1}{2d_1} \left( \frac{1}{R} + \frac{1}{d_1} \right)^2 + \frac{n_2}{2d_2} \left( \frac{1}{R} - \frac{1}{d_2} \right)^2 \right]$$

# Aberrazioni delle lenti reali

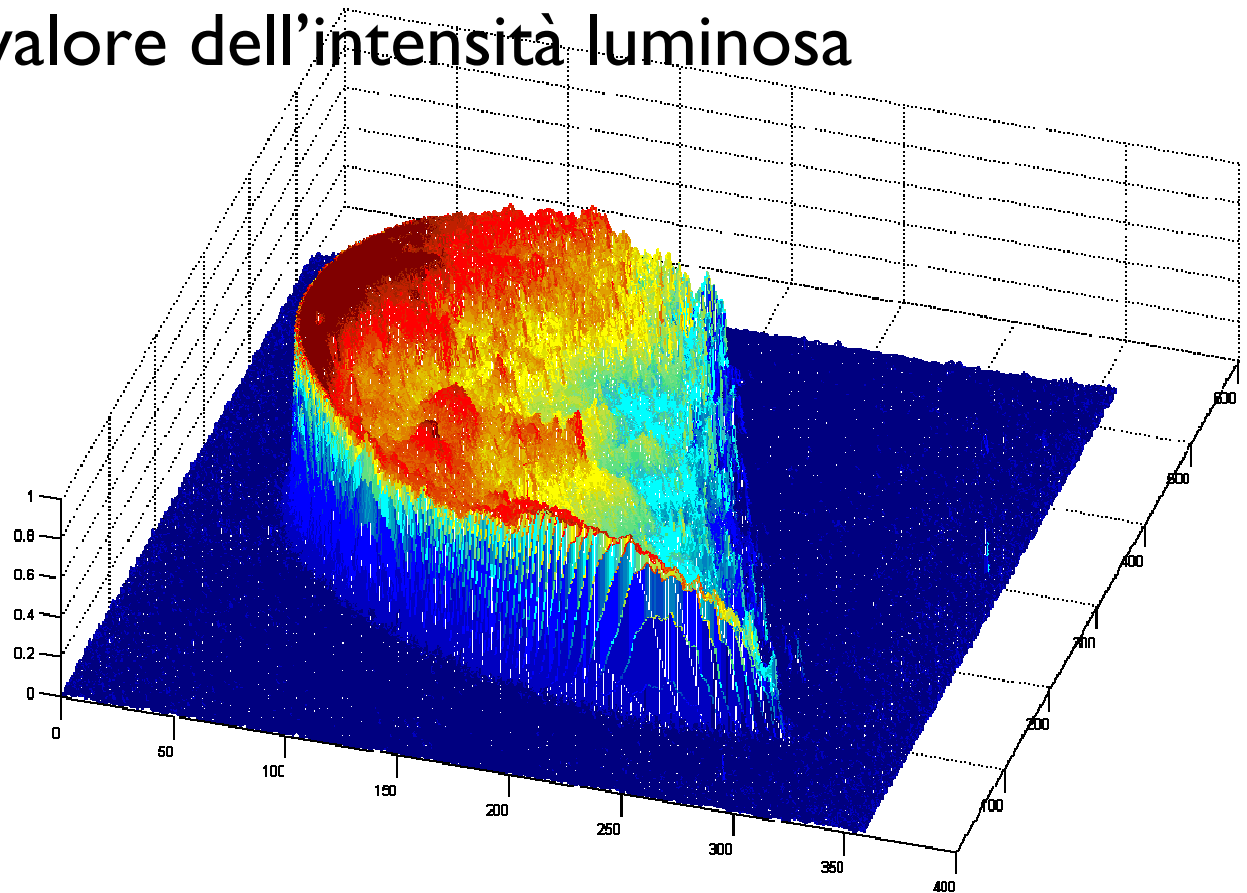
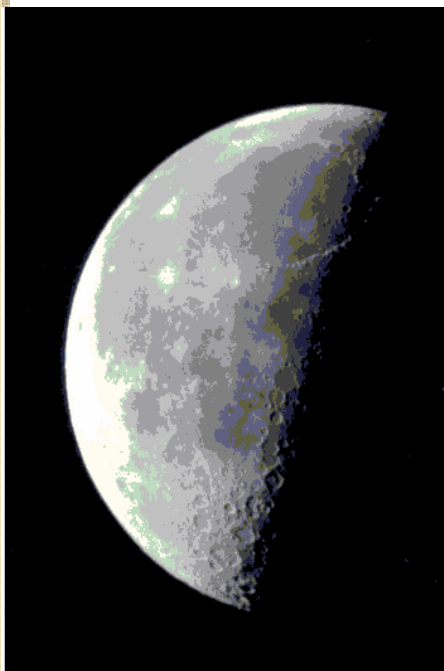
- **Aberrazione sferica** (inadeguatezza del modello parassiale)
  - Longitudinale e trasversale



- **Distorsione** (differenti lunghezze focali in differenti aree della lente)
- **Aberrazione cromatica** (indice di rifrazione dipende dalla lunghezza d'onda)
  - Longitudinale e trasversale

# Funzione immagine

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





# Immagine digitale

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



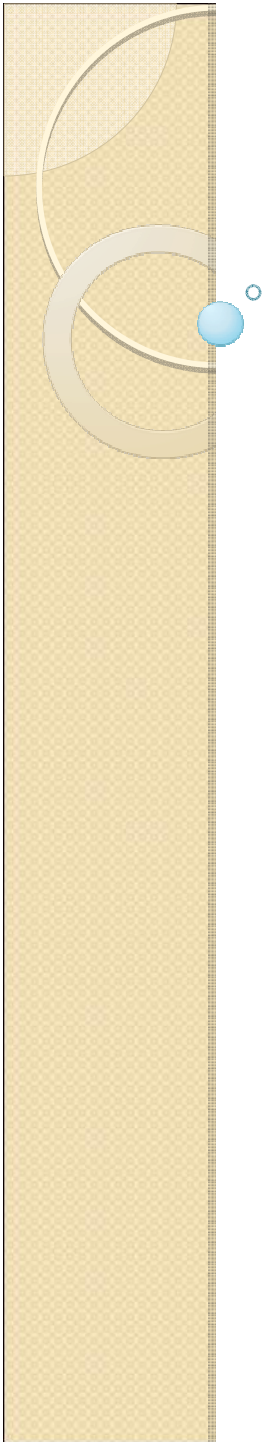
# Risoluzione spaziale



# Profondità di colore

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

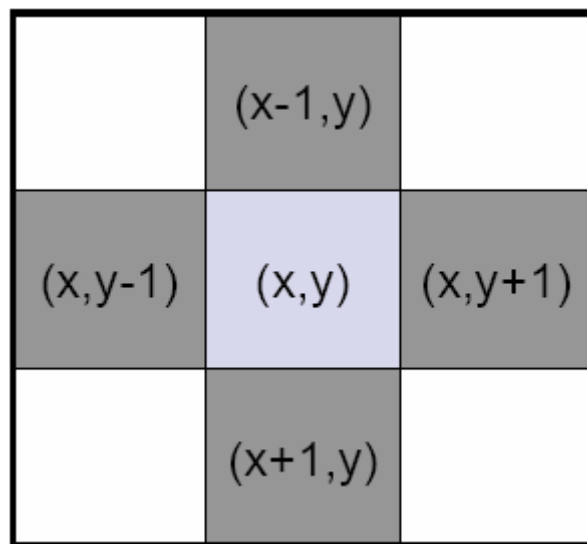




# Geometria delle immagini

## Punti 4-vicini

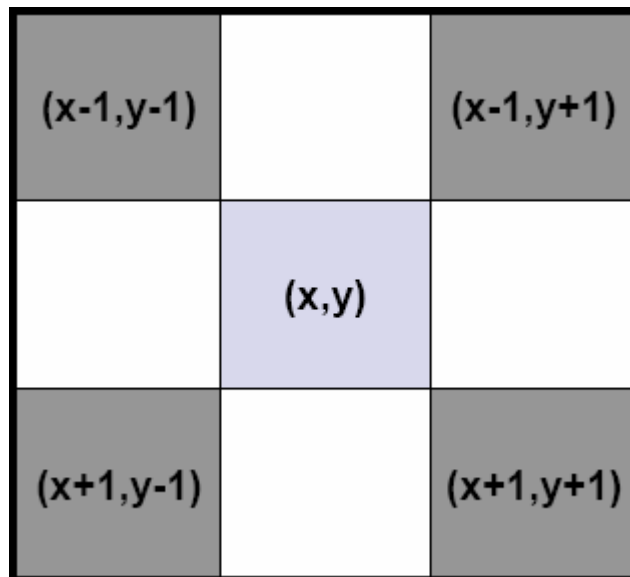
- Un punto  $P=(x,y)$  appartenente ad un'immagine digitale ha 4 punti vicini in orizzontale e in verticale



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

# Punti vicini diagonali

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



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

# Punti 8-vicini

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

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

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

Ovviamente :

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



# Connettività

- Siano P e Q due punti dell'immagine digitale allora:
  - P e Q sono **4-CONNESSI** se  $Q \in N_4(P)$
  - P e Q sono **8-CONNESSI** se  $Q \in N_8(P)$
  - P e Q sono **m-CONNESSI** se:
    - $Q \in N_4(P)$  oppure
    - $Q \in N_D(P) \wedge N_4(P) \cap N_4(Q) = \emptyset$

0 1 1

0 2 0

0 0 1

0 1—1

0 2 0

0 0 1

0 1—1

0 2 0

0 0 1

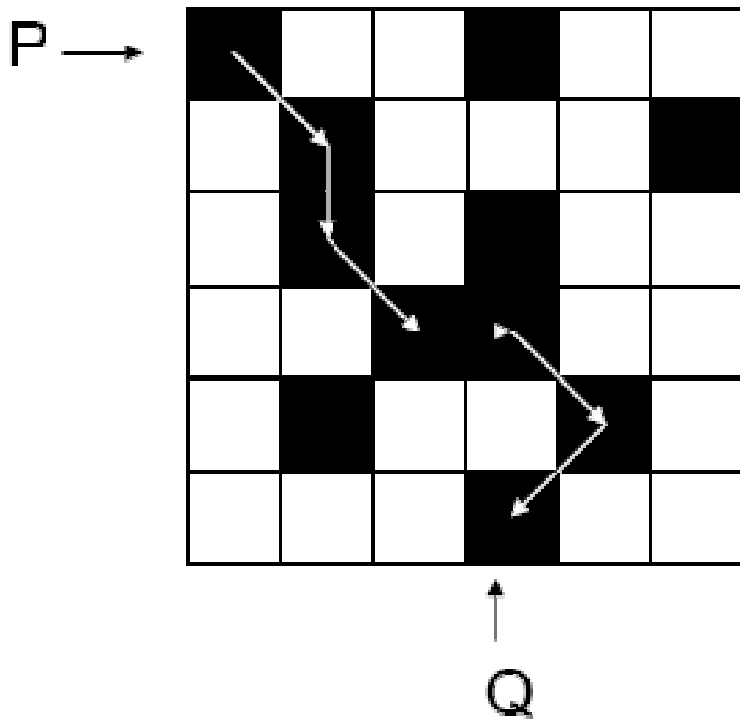




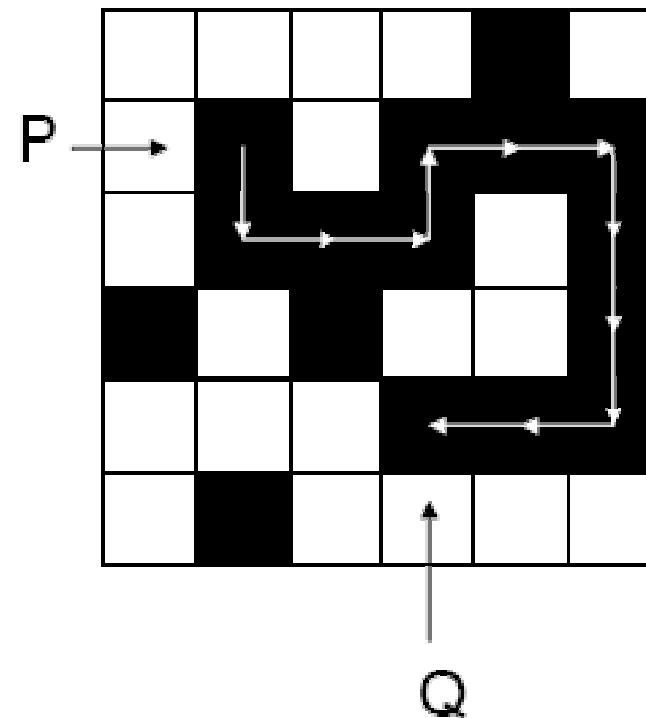
# Connettività

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

# Esempio



P è 8-connesso a Q



P è 4-connesso a Q

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



# Distanza sulle immagini digitali

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

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

- Sulle immagini digitali sono definite anche altre distanze, “più semplici”:
  - distanza *city block*
  - distanza *chessboard*

# Distanza city block

- E' definita come:

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

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

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

# Distanza chessboard

- E' definita come:

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

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

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



# Proprietà metriche

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





# Marcatura dei componenti connessi

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



# Marcatura dei componenti connessi

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



# Algoritmo di marcatura

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

# Esempio: Marcatura 8-connessi

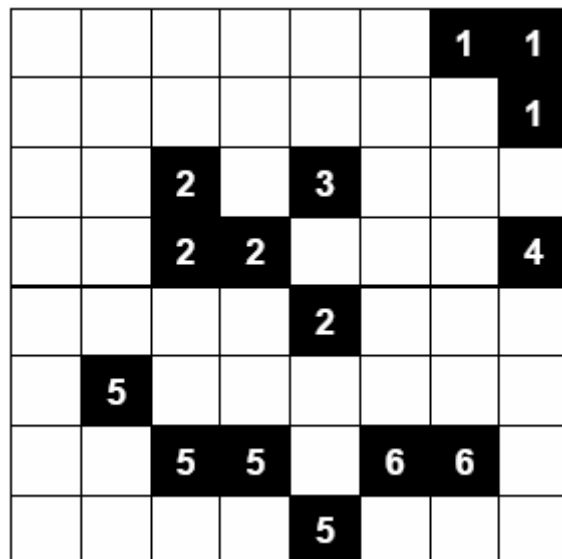
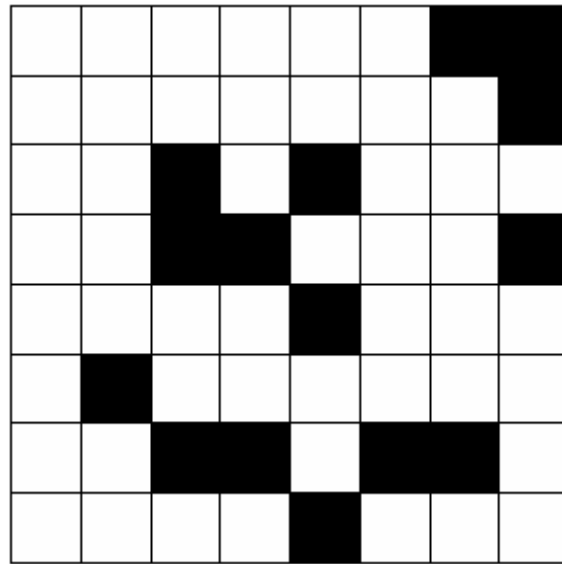
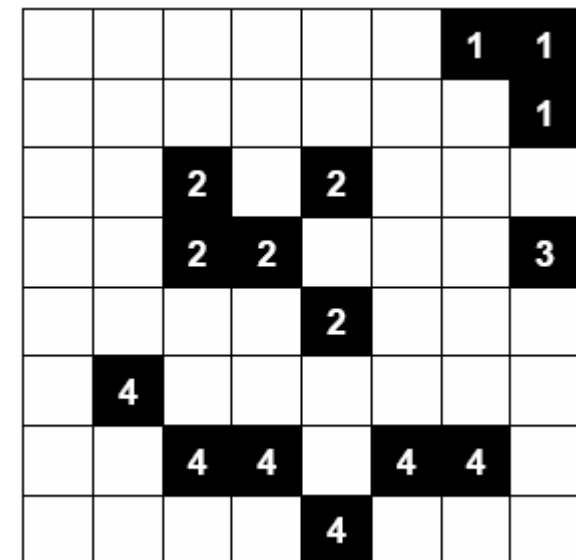


Tabella delle equivalenze	
3	2
6	5

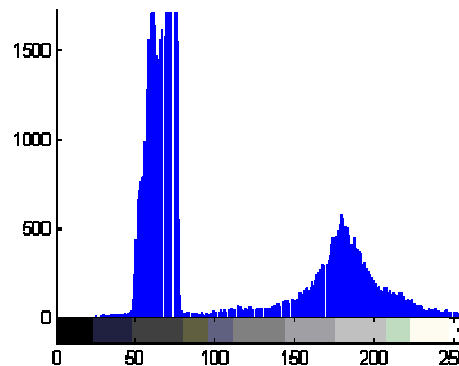


# Esempio MATLAB

- `I = imread('coins.png');`

- `imshow(I);`

- `imhist(I);`



- `IBW = im2bw(I, 98/255);`

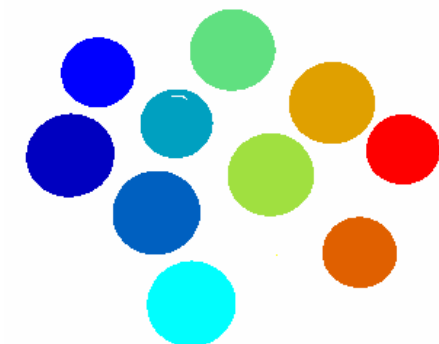
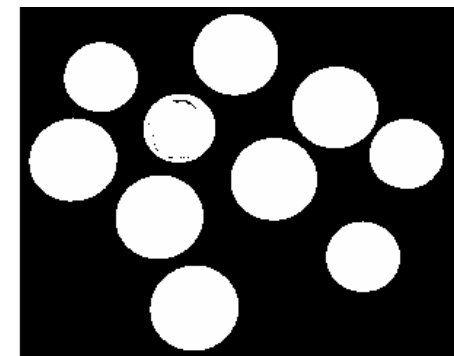
- `imshow(IBW);`

- `[lab,nObj] = bwlabel(IBW,4);`

- `numObjects = 10`

- `p_color = label2rgb(labeled, @jet)`

- `imshow(p_color);`

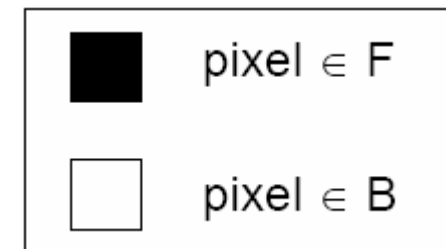
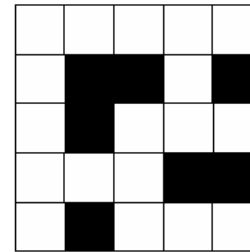


# Contorni nelle immagini binarie

- In un'immagine digitale binaria  $I$ , sono individuabili due sottoinsiemi disgiunti di pixel (chiamiamoli  $F$  e  $B$ ), individuati dai due diversi valori che questi possono assumere.

- Per tali insiemi valgono:

- $F = \text{not } B$
- $I = F \cup B$

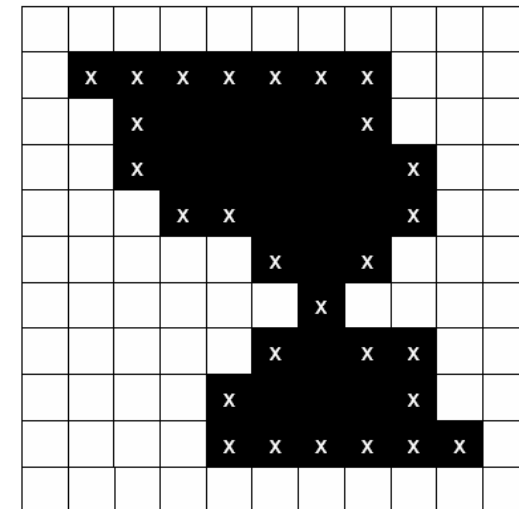


- Il **contorno** di un insieme  $T$  di punti di  $F$  è formato dall'insieme di punti di  $T$  che sono adiacenti a  $B$ .
- I punti di  $T$  che non appartengono al contorno si dicono **punti interni**; il loro insieme forma il *core* di  $T$



# Contorni immagini binarie

- Avere una rappresentazione del contorno può essere utile per:
  - esigenze di rappresentazione
  - esigenze di analisi
  - esigenze di descrizione
- P è un punto di contorno se:
  - $P \in F$
  - $\exists$  almeno un k-vicino di P appartenente a B.
- **Quindi, il contorno dipende dalla scelta delle relazioni di adiacenza.**





# Algoritmo per l'inseguimento del contorno di immagini binarie

- Dato un componente connesso  $S$ , l'algoritmo identifica la sequenza di punti che ne costituisce il contorno.
  - L'algoritmo parte da un punto  $P$  del contorno e individua i punti successivi, arrestandosi quando viene raggiunto il punto di partenza (**immagine binaria  $\Rightarrow$  contorno chiuso**).
- 1. Siano  $P$  e  $Q$  un punto del contorno di  $S$  ed un punto del background 4-adiacente a  $P$ .
- 2. Viene inizializzata la lista dei punti del contorno:  $LC=\{P\}$ .
- 3. Siano  $R_1=Q, R_2, R_3, \dots, R_8$  gli 8-vicini di  $P$  ordinati in senso antiorario a partire da  $Q$ .
- 4. Sia  $R_i$  il primo degli  $R$  che appartiene a  $S$ ;
  - si ponga  $P'=R_i$  e  $Q'=R_{i-1}$ .
- 5. Sia  $U$  l'ultimo punto nella lista  $LC$ .
  - IF la sequenza  $U-P'$  è già presente nella lista  $LC$  THEN stop
  - ELSE  $LC=LC \cup \{P'\}$ ;  $P=P'$ ;  $Q=Q'$ ; GOTO 3



# Pre-elaborazione



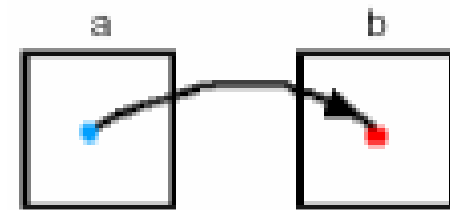
# Pre-elaborazione delle immagini

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

# Operazioni su immagini digitali

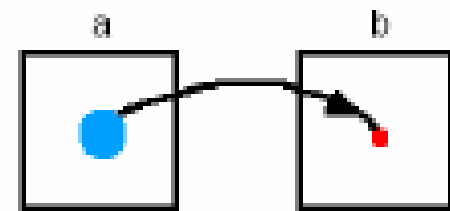
- **Operatori puntuali**

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



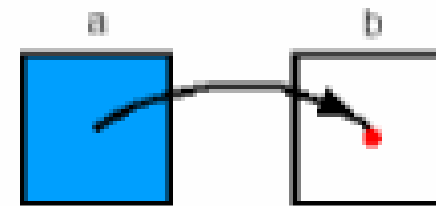
- **Operatori locali**

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



- **Operatori globali**

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





# Operazioni puntuali





# Trasformazioni puntuali

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



# Trasformazioni di base

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



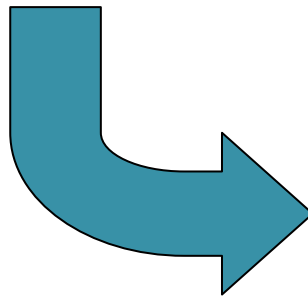
# Binarizzazione - sogliatura

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

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

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

# Esempio



# Inversione dei livelli di grigio

- Semplice trasformazione del tipo:

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

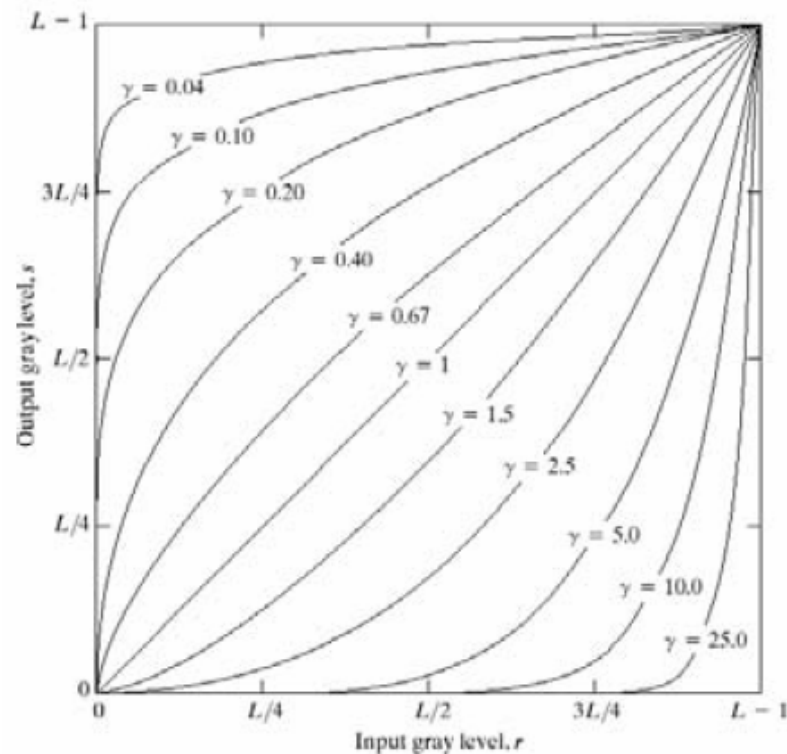
Fornisce la “negativa” dell’immagine originale



# Compressione di potenza

- Trasformazione del tipo:

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





# Esempio

originale



$\gamma = 3.0$



$\gamma = 4.0$



$\gamma = 5.0$





# Espansione di contrasto

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

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



# L'Istogramma

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

**Media, mediana, varianza, deviazione standard, quartili, percentili eccetera.**

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

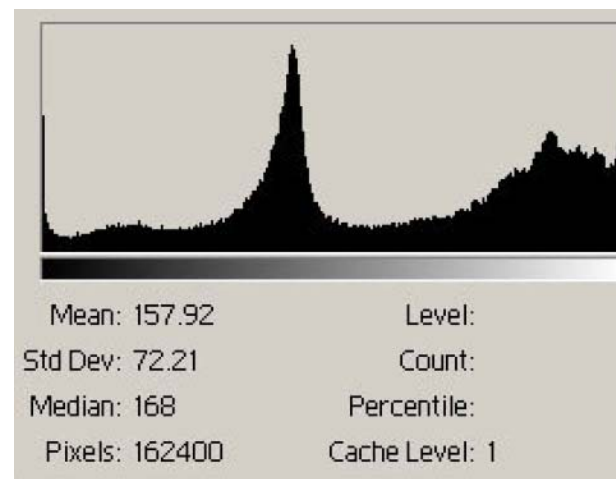
**l'istogramma.**

# Istogramma

- Per ogni livello di grigio, riporta il numero di pixel di quel colore.
- Per una immagine  $I[m,n]$  si ha:

$H(k)$  = numero di pixel di valore  $k$

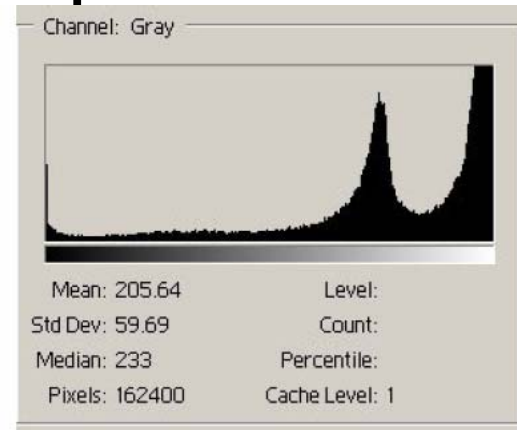
- La somma di tutti gli  $H$  è esattamente  $m \times n$



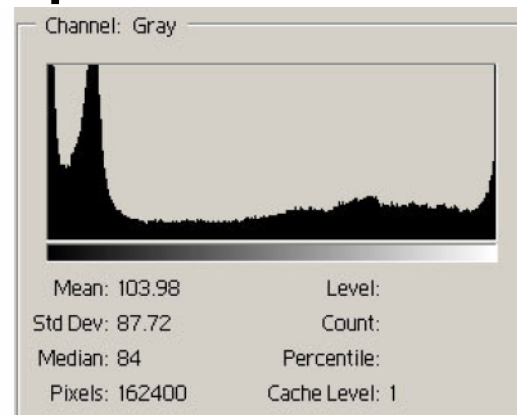
- L'istogramma è utile a comprendere in maniera immediata le caratteristiche dell'immagine.

# Caratteristiche istogramma

- Immagine “chiara” più denso a destra



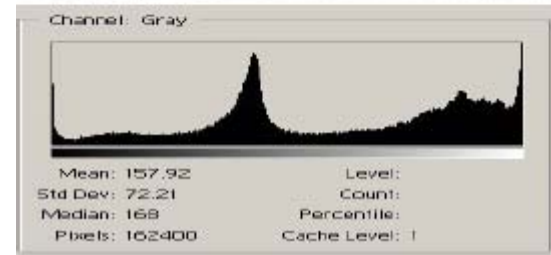
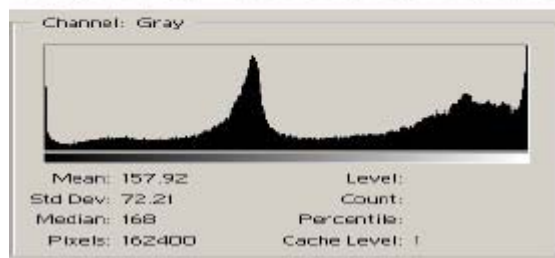
- Immagine “scura” più denso a sinistra





# Limite istogramma

- Non tiene conto della distribuzione spaziale







# Equalizzazione istogramma

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



# Equalizzazione istogramma

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

$$y=y(x)$$

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

$$g(y)=C$$

- Si impone che aree elementari dell'istogramma originale si trasformano in aree corrispondenti dell'istogramma modificato, si ha:

$$h(x)dx = g(y)dy = Cdy$$

# Equalizzazione istogramma

- Ricavo  $y(x)$ :

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

- Nel dominio discreto diventa:

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

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

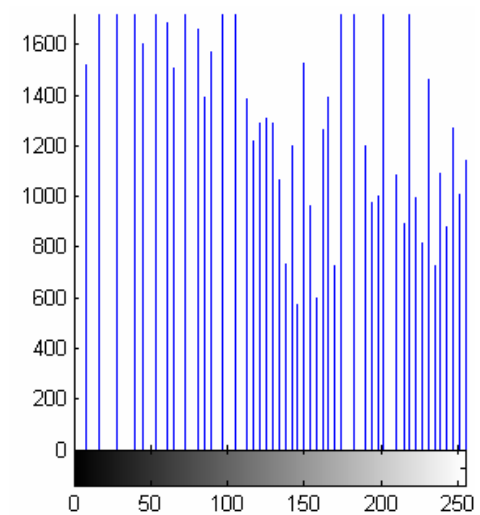
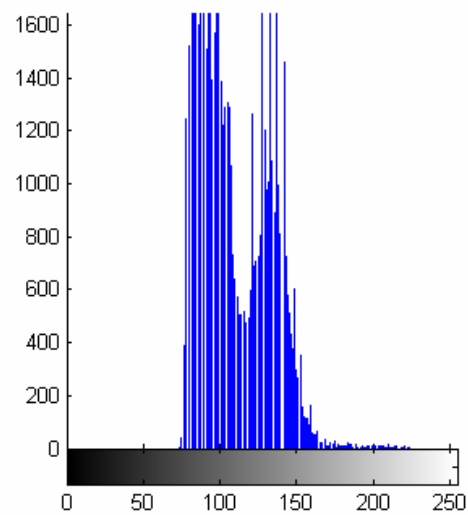
$$y(x) = \frac{1}{C} \sum_{k=0}^x H(k) = \frac{256}{\sum_{k=0}^{255} H(k)} \sum_{k=0}^x H(k)$$



# Algoritmo equalizzazione

1. Valutare istogramma  $H(k)$
2. Calcolare la 
$$y(x) = \frac{l}{n \cdot m} \sum_{k=0}^x H(k)$$
3. Eseguire la trasformazione tramite la  $y(x)$

# Esempio





# Operazioni locali



# Operatori locali

- Questi operatori sono usati per:
  - miglioramento della qualità di un'immagine (come per gli operatori puntuali)
  - estrazione di caratteristiche dell'immagine (immagine in ingresso  $\Rightarrow$  immagine delle caratteristiche)
- Il valore di uscita dell'operatore nel punto  $(i,j)$  dipende solo dai valori di ingresso in un vicinato del punto  $(i,j)$
- Il vicinato è di solito definito in maniera simmetrica rispetto al punto
- Gli operatori locali possono essere di tipo lineare o non lineare





# Principali operatori locali

- Filtri di smoothing
- Filtri di sharpening
- Filtri derivativi

# Operatori locali

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

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

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



# Operatori locali

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

Esempio di filtro non lineare:

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



# Filtri di smoothing

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



# Caratteristiche dei filtri di smoothing

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

# Filtri di smoothing

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

1/9

1	1	1
1	1	1
1	1	1

1/25

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

# Filtri di smoothing

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

0.0113	0.0838	0.0113
0.0838	0.6193	0.0838
0.0113	0.0838	0.0113

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



# Esempio: Blurring

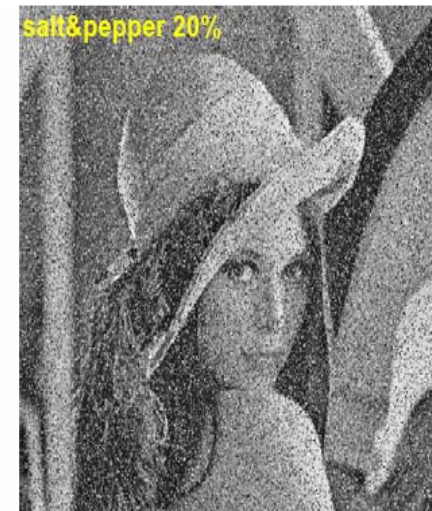


# Esempio: Blurring



# Noise cleaning

- **rumore impulsivo**, detto anche “sale e pepe” (*salt & pepper*). Viene caratterizzato dalla frazione (in %) dell'immagine modificata.



- **rumore gaussiano bianco**. Viene caratterizzato dalla media e dalla varianza.

media=0 varianza=0.01



media=0 varianza=0.1





# Rumore impulsivo e filtri di media

Filtro di media 3x3



Filtro di media 5x5



# Il filtro mediano

- Una tecnica alternativa è quella di usare un filtro mediano
- E' un filtro **non lineare** che fornisce in uscita il valore mediano dell'intorno del pixel

- Esempio:

7	10	12
6	38	11
9	11	6

6 6 7 9 10 11 11 12 38



valore  
mediano

# Applicazione filtro mediano

Filtro mediano 3x3

salt&pepper 10%



salt&pepper 20%



Filtro mediano 5x5

salt&pepper 10%



salt&pepper 20%





# Media o mediana?

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





# Filtri di sharpening

- Lo scopo di questo tipo di filtri è di incrementare la nitidezza dell'immagine aumentando il contrasto locale.
- Di conseguenza, vengono enfatizzati i dettagli fini e le regioni di bordo, al contrario dei filtri di smoothing.
- Tali filtri agiscono come filtri passa-alto rispetto alla frequenza spaziale



# Caratteristiche filtri di sharpening

- Vengono realizzati tramite operazioni di differenziazione spaziale.
- L'ampiezza della risposta di un operatore differenziale è proporzionale al grado di discontinuità dell'immagine nel punto in cui l'operatore è applicato.
- La differenziazione dell'immagine enfatizza i bordi e altre discontinuità (rumore) e deenfattizza le aree con livelli di grigio lentamente variabili.



# Implementazione filtri di sharpening

Consideriamo filtri di sharpening basati su derivate di primo e secondo ordine.

- Come si realizza una derivata su un'immagine digitale ?
  - Tramite differenze
- Proprietà per la derivata di primo ordine:
  - Nulla su aree con livelli di grigio costanti
  - Non nulla all'attacco di un gradino o di una rampa a livelli di grigio
  - Non nulla lungo una rampa a livelli di grigio
- Proprietà per la derivata di secondo ordine:
  - Nulla su aree con livelli di grigio costanti
  - Non nulla all'attacco di un gradino o di una rampa a livelli di grigio
  - Nulla lungo una rampa a livelli di grigio di inclinazione costante

# Approssimazione derivata prima

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

0	0	0
0	-1	0
0	+1	0

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

0	0	0
0	-1	+1
0	0	0

# Approssimazione derivata seconda

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

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

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

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



# Isotropia dell'operatore: Il laplaciano

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

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

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



# Implementazione del Laplaciano

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

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

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

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

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


$$\nabla^2 f$$

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

# Filtro di sharpening mediante Laplaciano

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



$$f(x, y)$$



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



$$f - \nabla^2 f$$

# Maschere filtri di sharpening

- La maschera risultante è:

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

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

- Spesso si usano anche altre due maschere di sharpening:

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

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

# Filtri derivativi del primo ordine

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

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

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

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



# Implementazione filtri derivativi

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

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

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

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

# Operatore di Roberts

- Storici

OPERATORE INCROCIATO DI ROBERTS:

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

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

-1	0
0	1

0	-1
1	0





# Operatori più “moderni”

- Esistono anche delle implementazioni con maschere  $3 \times 3$ , in cui è più chiaramente definito il punto al quale l'operatore è applicato.
- Implementazioni molto usate sono date da:
  - Operatori di Prewitt
  - Operatori di Sobel

# Operatori di Prewitt

OPERATORI DI PREWITT:

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

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

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

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

# Operatori di Sobel

OPERATORI DI SOBEL:

$$\nabla_1 = f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1) \\ - f(x-1, y-1) - 2f(x, y-1) - f(x+1, y-1)$$

$$\nabla_2 = f(x-1, y-1) + 2f(x-1, y) + f(x-1, y+1) \\ - f(x+1, y-1) - 2f(x+1, y) - f(x+1, y+1)$$

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

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