Analisi di Immagini e Video (Computer Vision)

Giuseppe Manco

Outline

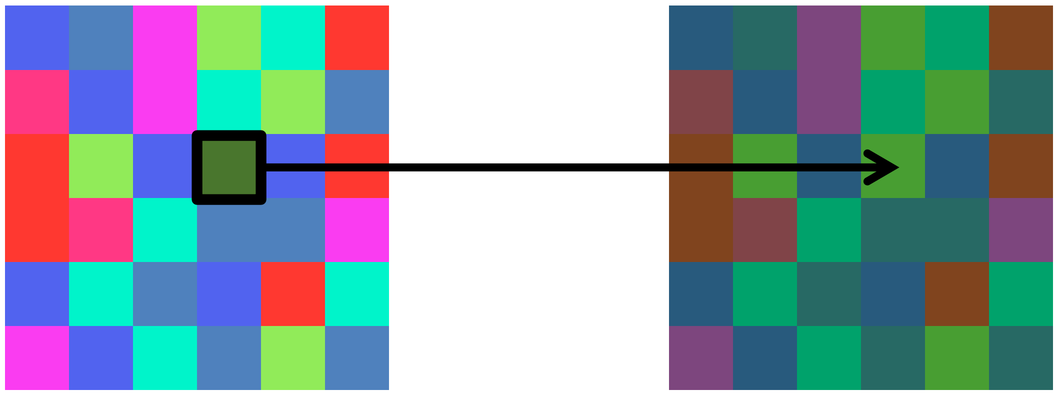
- Filtri e normalizzazione
- Image Processing avanzato
  - Edge detection

Crediti

- Slides adattate da vari corsi
  - Analisi di Immagini (F. Angiulli) – Unical
  - Intro to Computer Vision (J. Tompkin) – CS Brown Edu
  - Computer Vision (I. Gkioulekas), CS CMU Edu

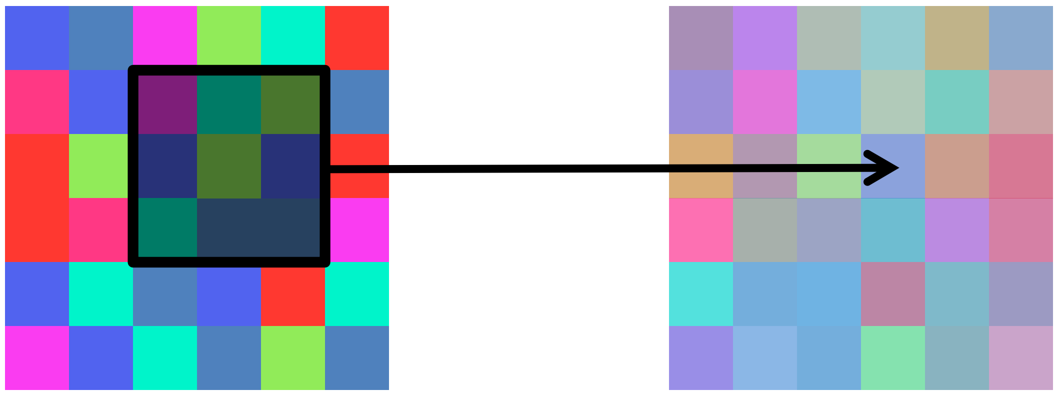
Filtri

Point Operation



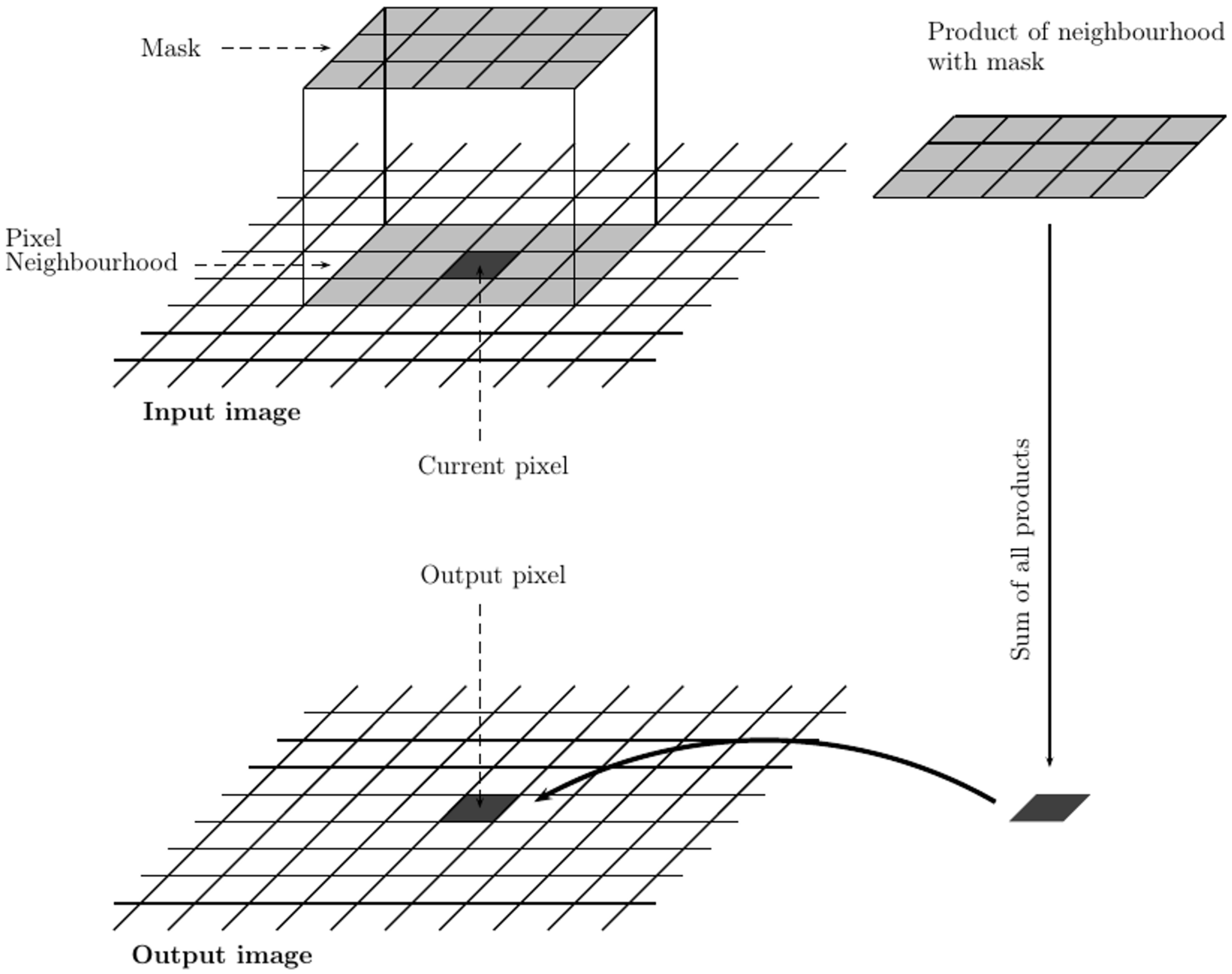
point processing

Neighborhood Operation



“filtering”

Filtraggio spaziale lineare



Filtraggio spaziale lineare

$$\mathrm{h}[i,j] = \sum_{k=-a}^{a} \sum_{l=-b}^{b} f[k+a, l+b] \times I[i+k, j+l]$$

- *f*, matrice dei coefficienti (maschera):
  - detta filter, mask, filter mask, kernel, template, window
- Maschera di dimensione *m* x *n* (in genere dispari):
  - *m = 2a+1*, *n = 2b+1*

Esempio: box (average) filter

$$f[\cdot\,,\cdot\,]$$

$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

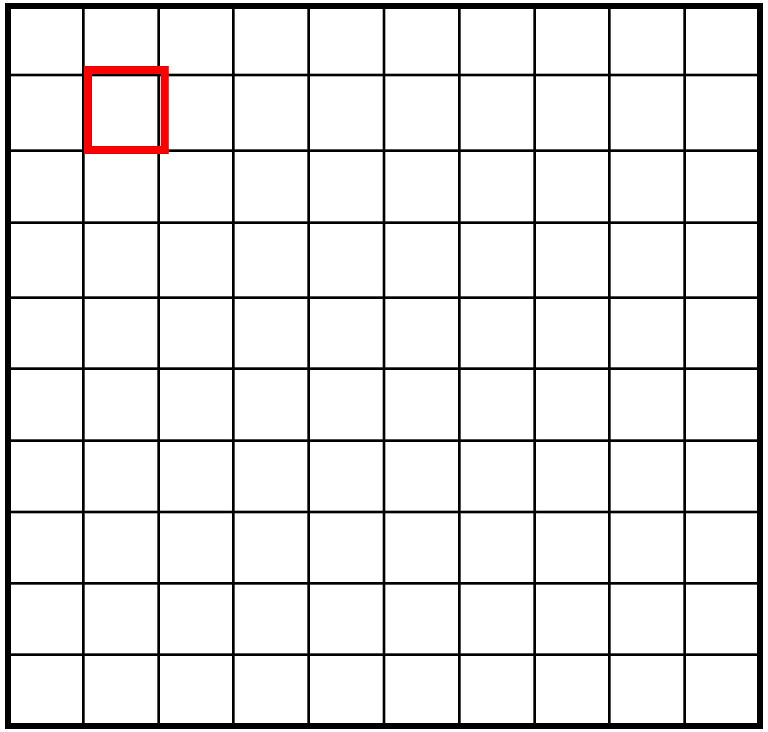
Image filtering

$$f[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$I[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$



$$\mathrm{h}[i,j] = \sum_{k=-a}^{a}\sum_{l=-b}^{b} f[k+a, l+b] \times I[i+k, j+l]$$

$i = 1, j = 1$

$a, b = 1$

Image filtering

$$f[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

$I[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10							

$$\mathrm{h}[i,j] = \sum_{k=-a}^{a}\sum_{l=-b}^{b} f[k+a,l+b]\times I[i+k,j+l]$$

$$i = 1, j = 2$$
$$a, b = 1$$

Image filtering

$f[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$I[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20						

$$h[i,j] = \sum_{k=-a}^{a} \sum_{l=-b}^{b} f[k+a, l+b] \times I[i+k, j+l]$$

$$i = 1, j = 3$$
$$a, b = 1$$

Image filtering

$f[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$I[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30					

$$\mathrm{h}[i,j] = \sum_{k=-a}^{a} \sum_{l=-b}^{b} f[k+a, l+b] \times I[i+k, j+l]$$

$$i = 1, j = 4$$
$$a, b = 1$$

Image filtering

$f[\cdot,\cdot]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$I[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				

$$h[i,j] = \sum_{k=-a}^{a} \sum_{l=-b}^{b} f[k+a, l+b] \times I[i+k, j+l]$$

$$i = 1, j = 5$$
$$a, b = 1$$

Image filtering

$f[\cdot,\cdot]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$I[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
				?					

$$h[i,j] = \sum_{k=-a}^{a}\sum_{l=-b}^{b} f[k+a, l+b] \times I[i+k, j+l]$$

$$i = 6, j = 4$$
$$a, b = 1$$

Image filtering

$f[\cdot\,,\cdot\,]\ \frac{1}{9}$

1	1	1
1	1	1
1	1	1

$I[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30				
						?			
				50					

$$h[i,j] = \sum_{k=-a}^{a} \sum_{l=-b}^{b} f[k+a, l+b] \times I[i+k, j+l]$$

$$i = 4, j = 6$$
$$a, b = 1$$

Image filtering

$f[\cdot\,,\cdot\,]\ \frac{1}{9}\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$

$I[.,.]$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$h[.,.]$

	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$$\mathrm{h}[i,j] = \sum_{k=-a}^{a}\sum_{l=-b}^{b} f[k+a,l+b]\times I[i+k,j+l]$$

$i = 1, j = 1$
$a, b = 1$

Smoothing mediante filtraggio spaziale

- **Average filter**
  - Sostituisce l'intensità del pixel col valore medio del suo vicinato
  - **Smoothing**: Le transizioni brusche (sharp) d'intensità vengono attenutate

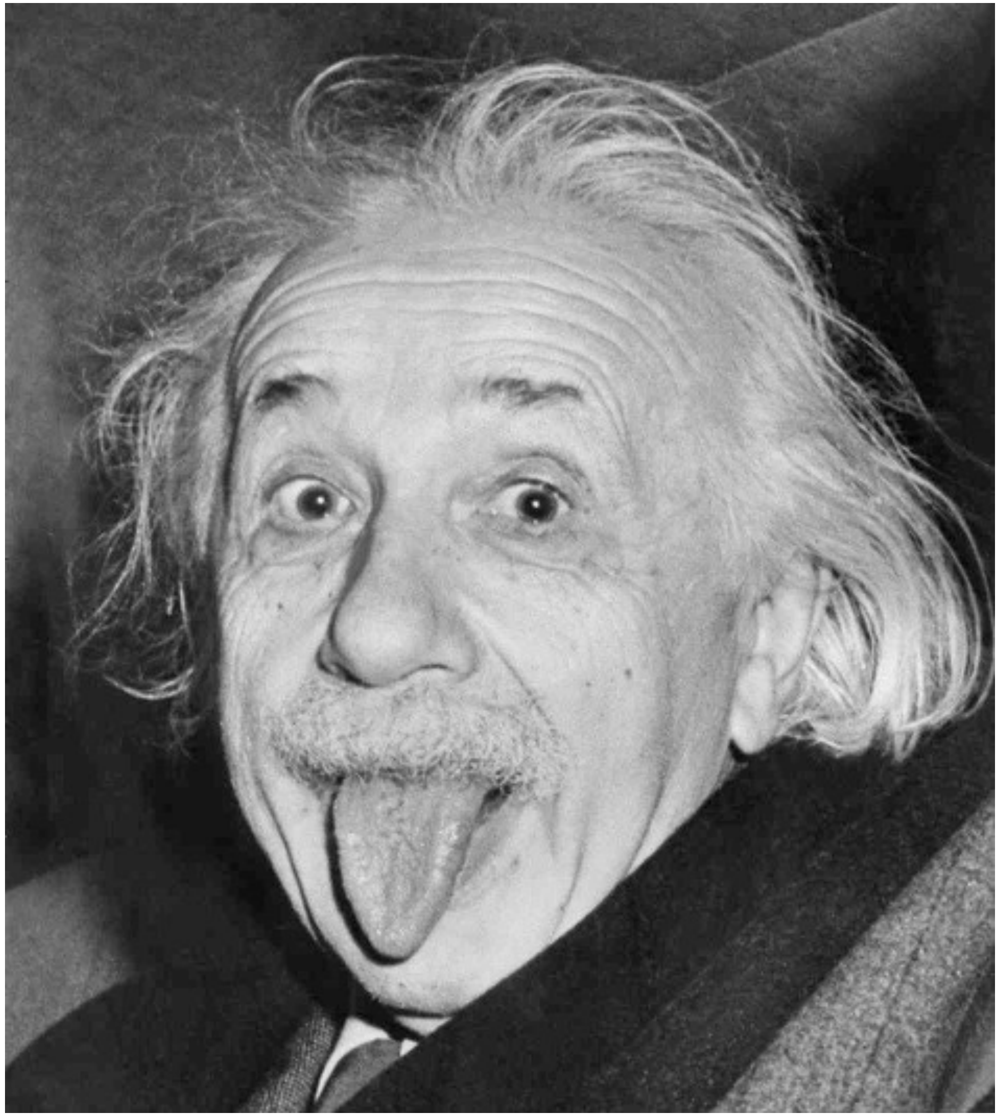
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

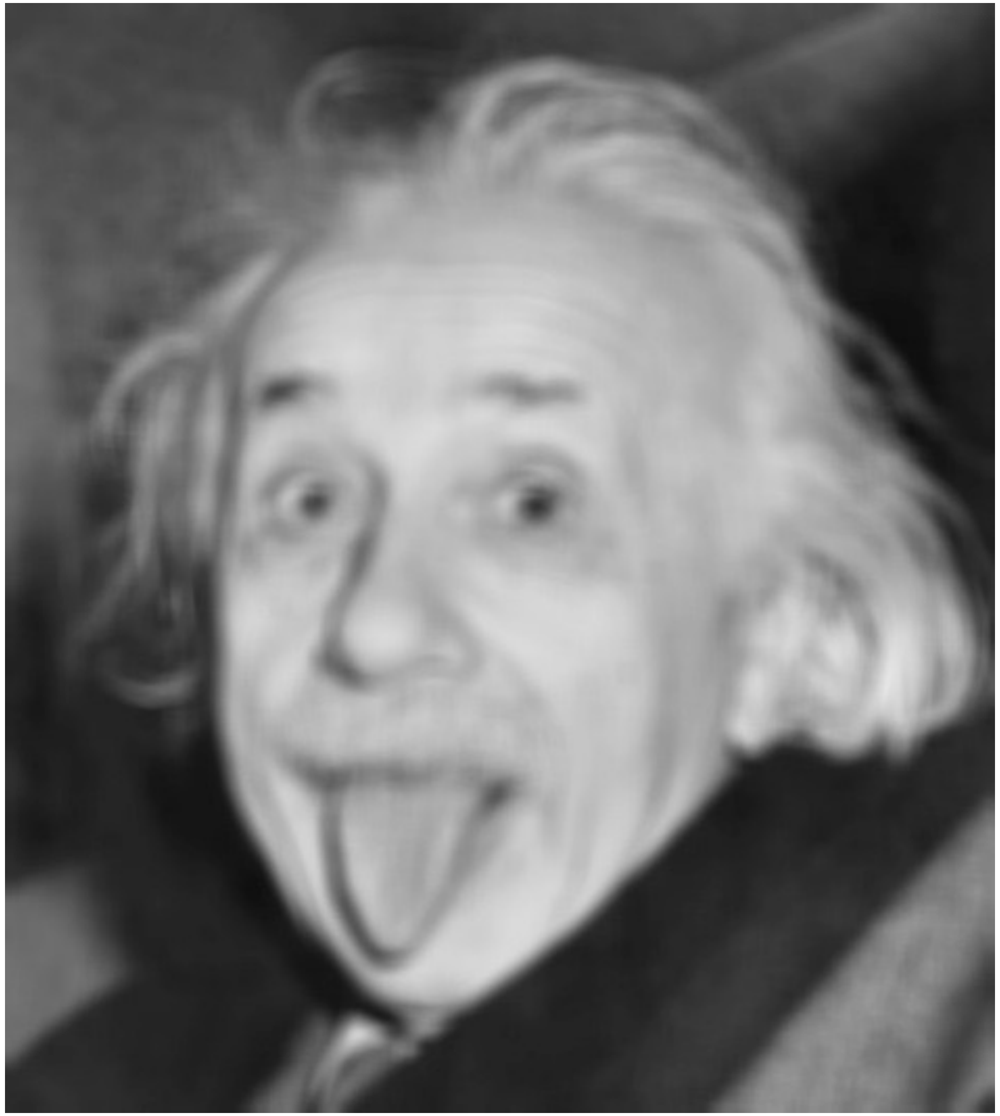
Smoothing mediante filtraggio spaziale con average filter

- *Riduzione del rumore* (noise removal)
  - Side-effect: i bordi (edge) vengono attenuati (blur)

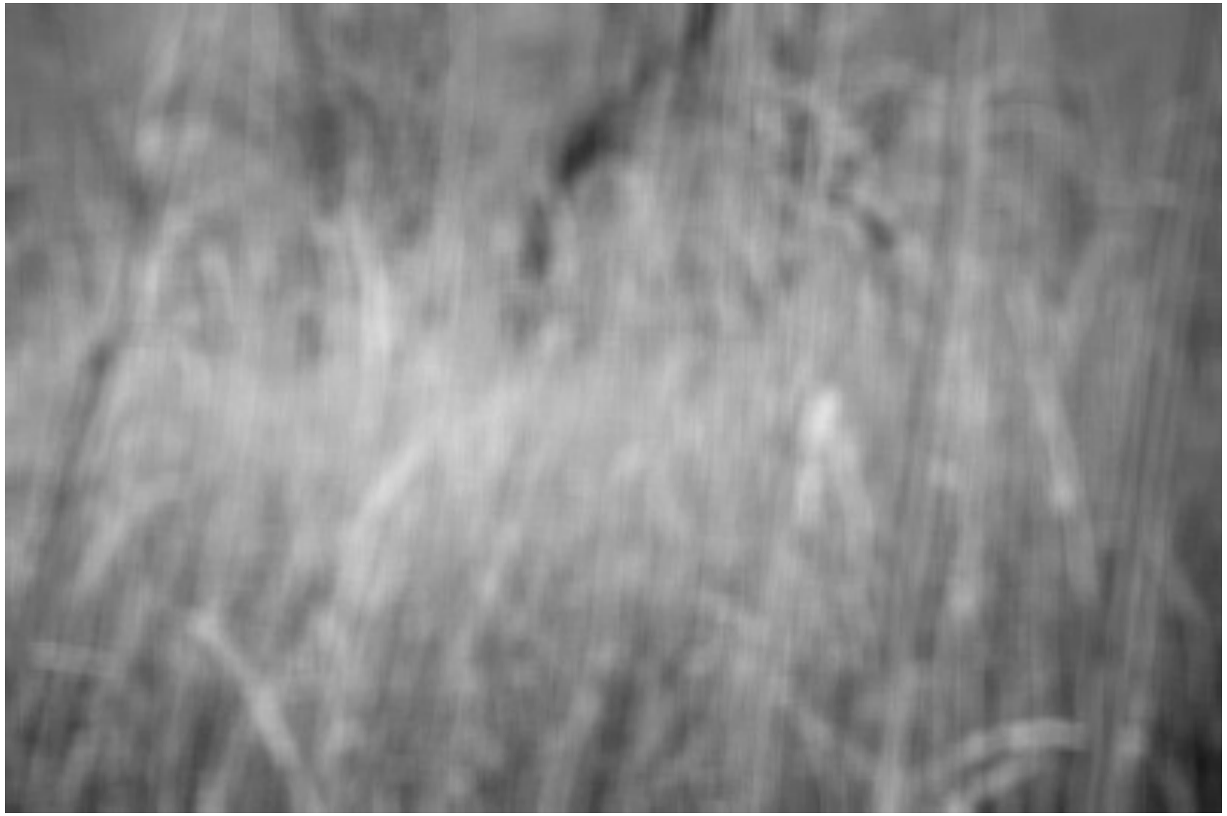
- *Riduzione dei dettagli "irrilevanti"* (image blurring)
  - Offuscare l'immagine per ottenerne una rappresentazione grossolana
  - Gli oggetti più piccoli si confondono con lo sfondo
  - Gli oggetti più grandi diventano "bloblike" e facili da individuare

Esempi





Esempi

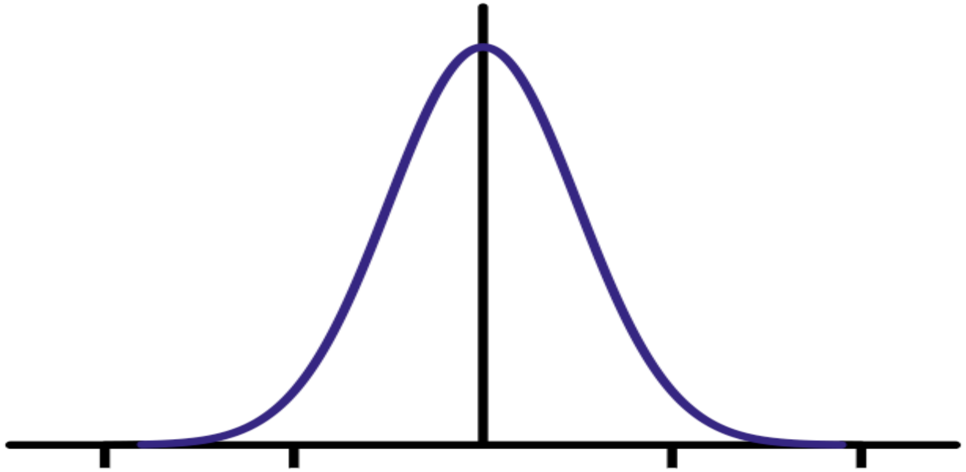


Esempi

Filtro Gaussiano

- Campiona I valori del kernel sulla base della funzione

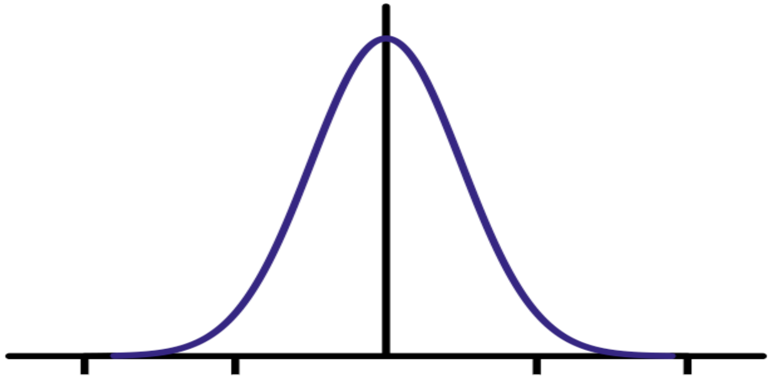
$$f(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$



Filtro Gaussiano

- Campiona i valori del kernel sulla base della funzione

$$f(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$



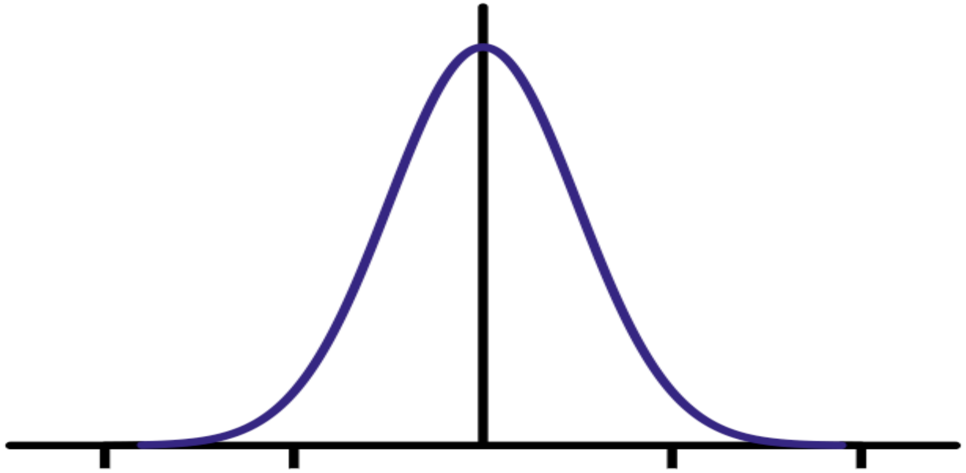
- I pesi decadono con la distanza dal centro
- Ridurre l'effetto di blurring quando si effettua l'operazione di smoothing
- Coefficienti inversamente proporzionali alla distanza dal pixel centrale
- Con maschera piccola non vi sono grandi differenze

kernel $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

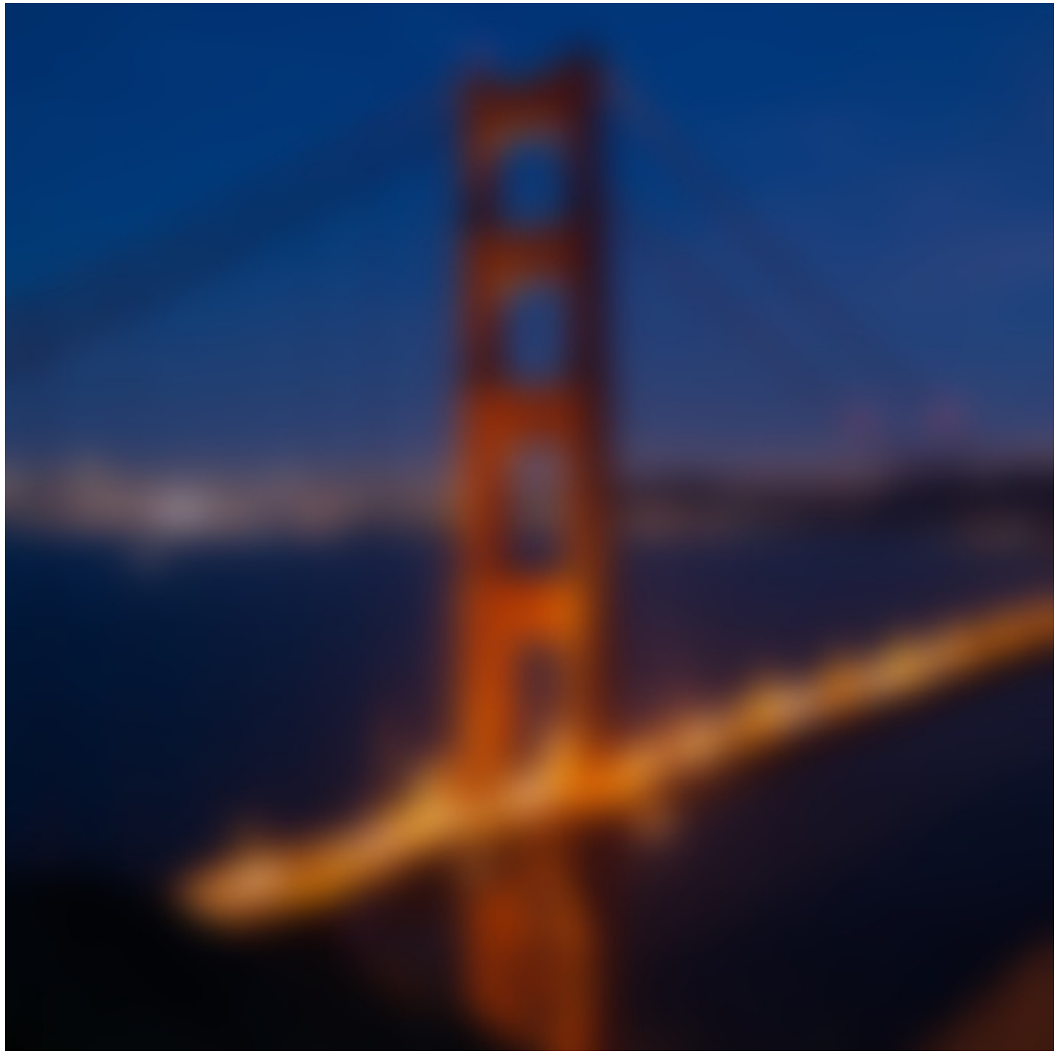
Dimensione ottimale?

- Il filtro gaussiano è potenzialmente infinito…
- Regola empirica (Gaussian): settiamo l'ampiezza a 6 σ

$$f(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}$$



Esempio

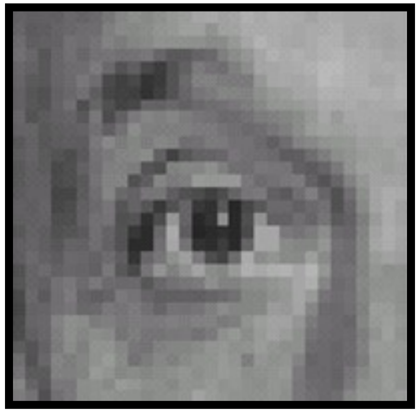


Filtraggio spaziale lineare

$$\mathrm{h}[i,j] = \sum_{k=-a}^{a}\sum_{l=-b}^{b} f[k+a, l+b] \times I[i+k, j+l]$$

- Fondamentale!
  - Migliora l’immagine
    - Denoise, ridimensiona, aumenta il contrasto, etc.
  - Estrae informazioni dall’immagine
    - Texture, edges, distinctive points, etc.
  - Trova patterns
    - Template matching

Esempi



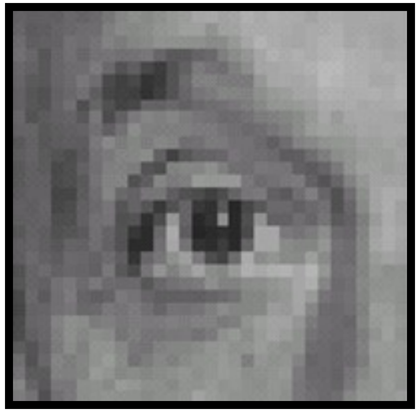
1. $$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

2. $$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ \hline \end{array}$$

3. $$\begin{array}{|c|c|c|}\hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array}$$

4. $$\begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \frac{1}{9}\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

1

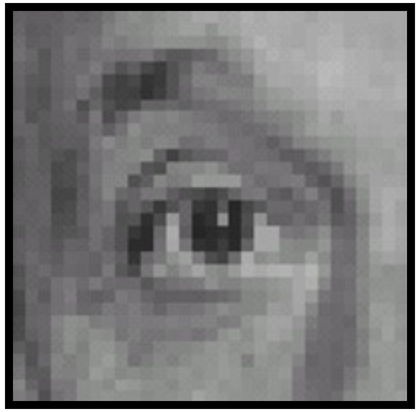


Original

0	0	0
0	1	0
0	0	0

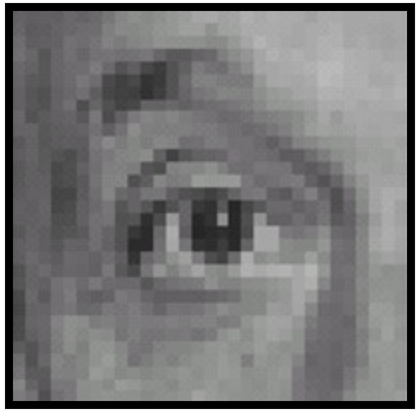
?

1



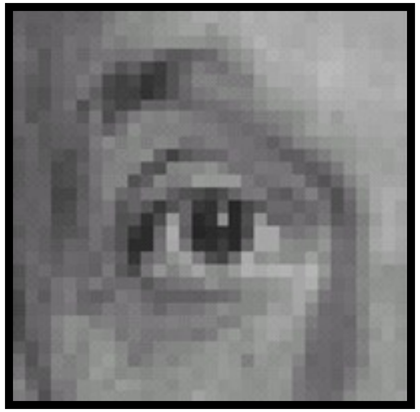
Original

0	0	0
0	1	0
0	0	0



Filtered
(no change)

2

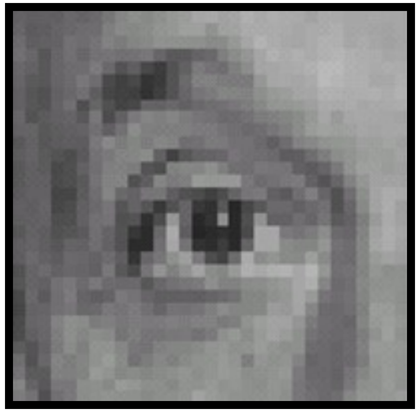


Original

0	0	0
0	0	1
0	0	0

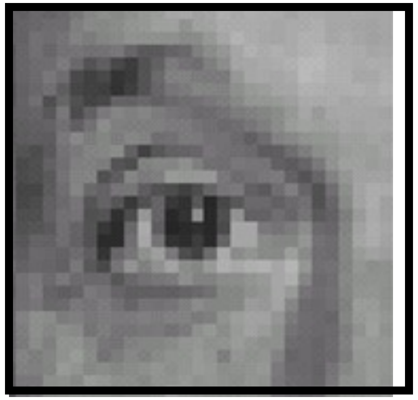
?

2



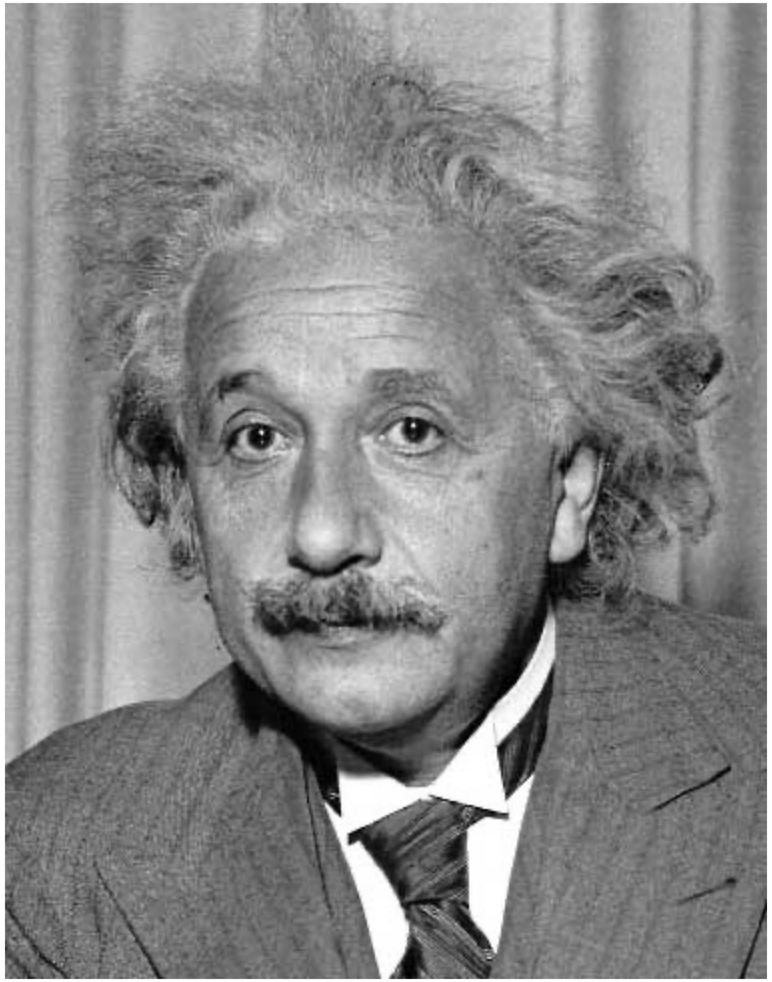
Original

0	0	0
0	0	1
0	0	0



Shifted right
By 1 pixel

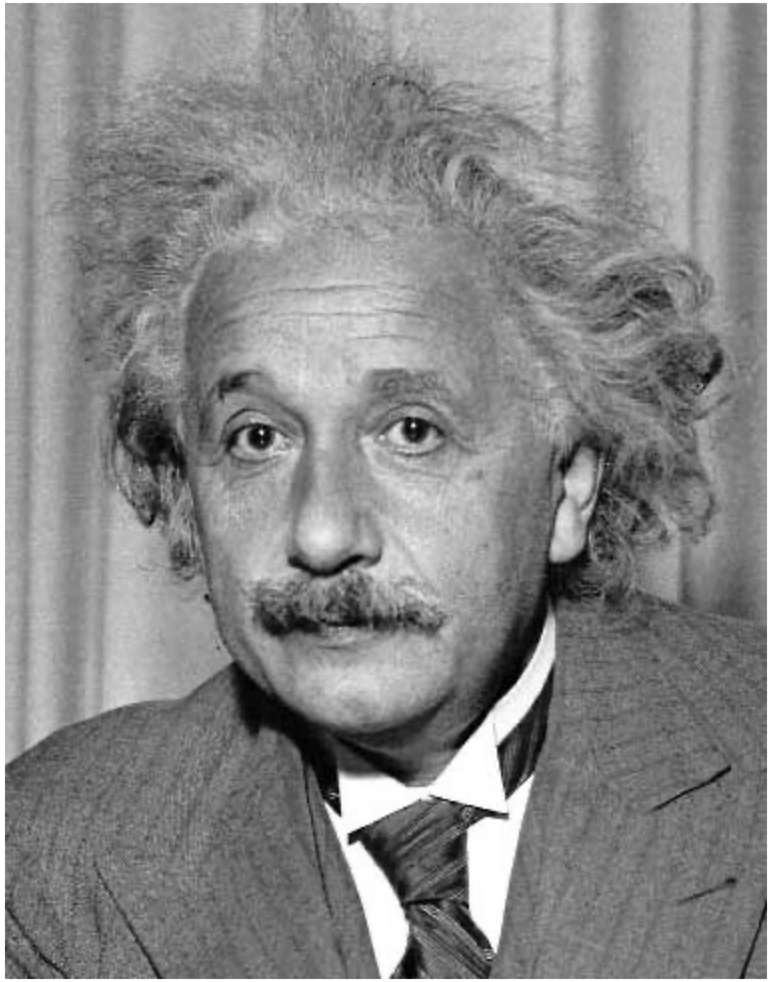
3



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

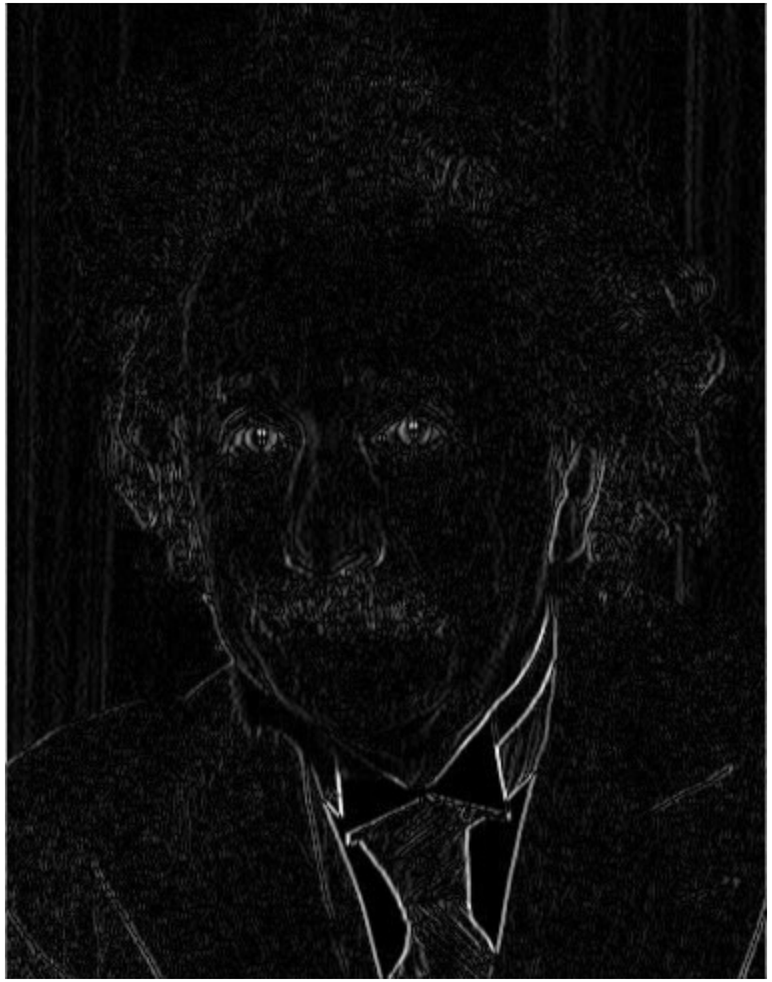
?

3



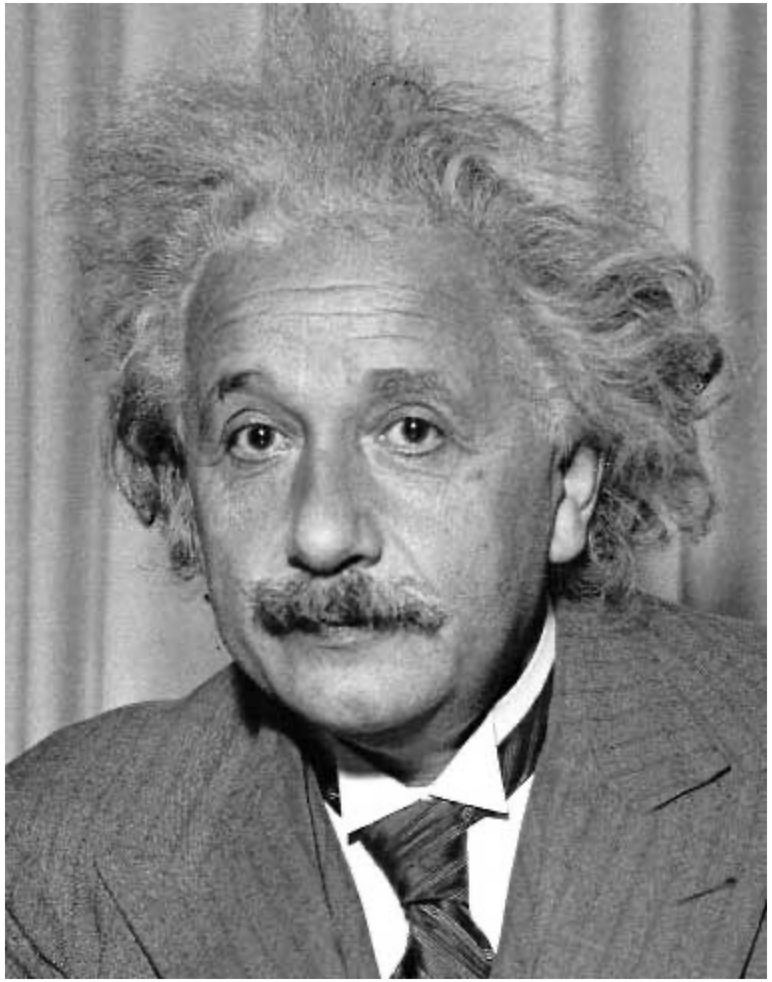
1	0	-1
2	0	-2
1	0	-1

Sobel



Vertical Edge
(absolute value)

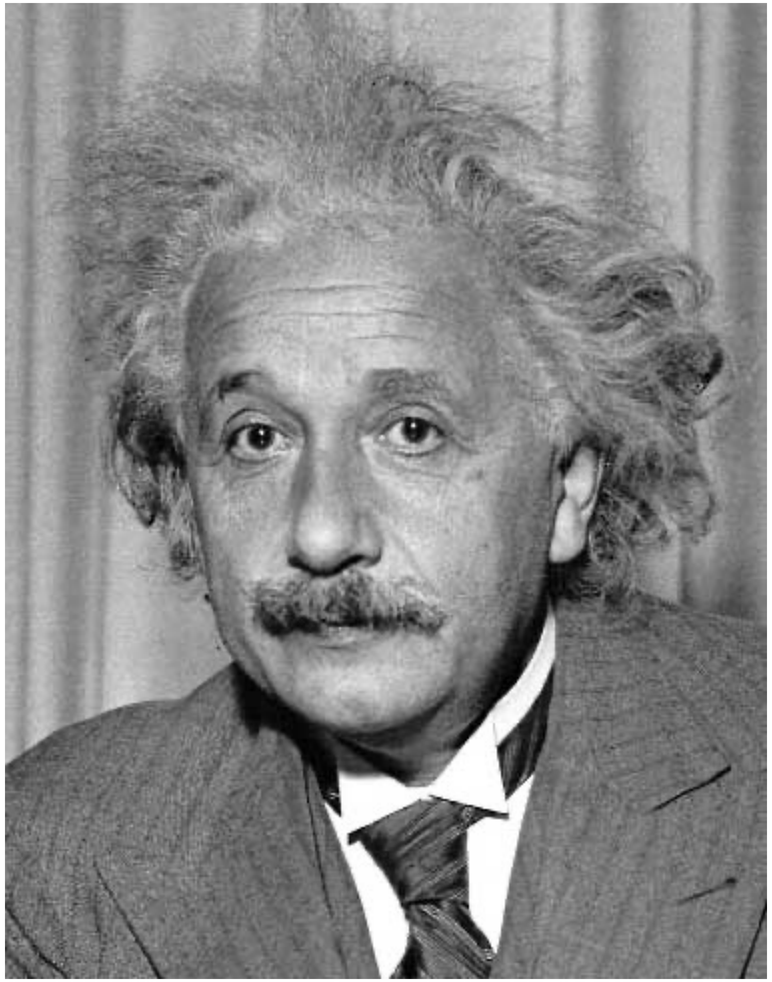
3



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

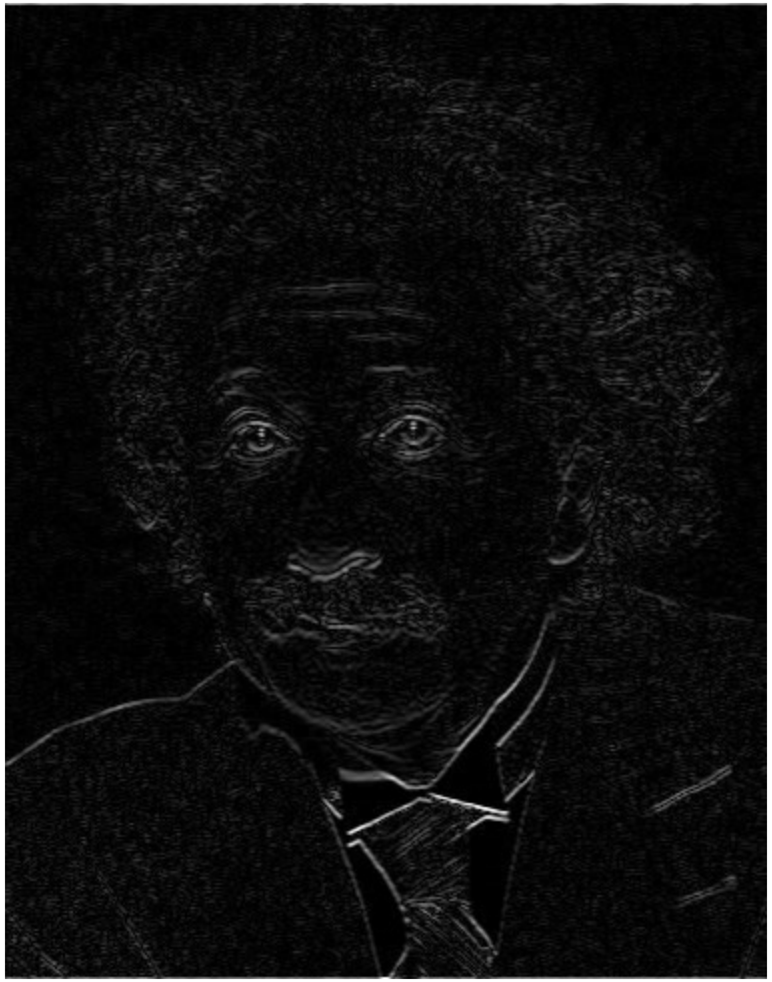
?

3



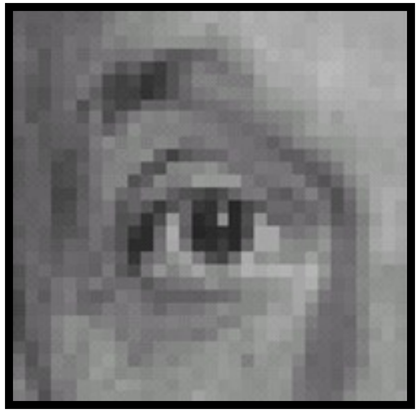
1	2	1
0	0	0
-1	-2	-1

Sobel



Horizontal Edge
(absolute value)

4

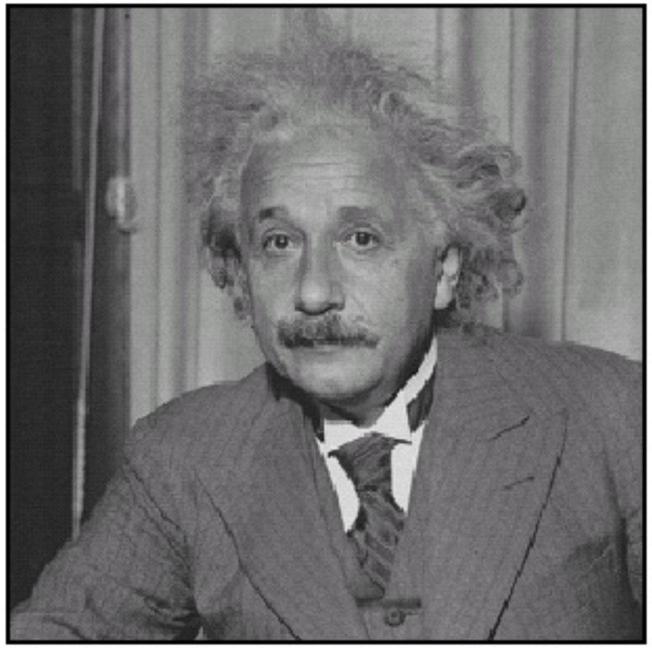


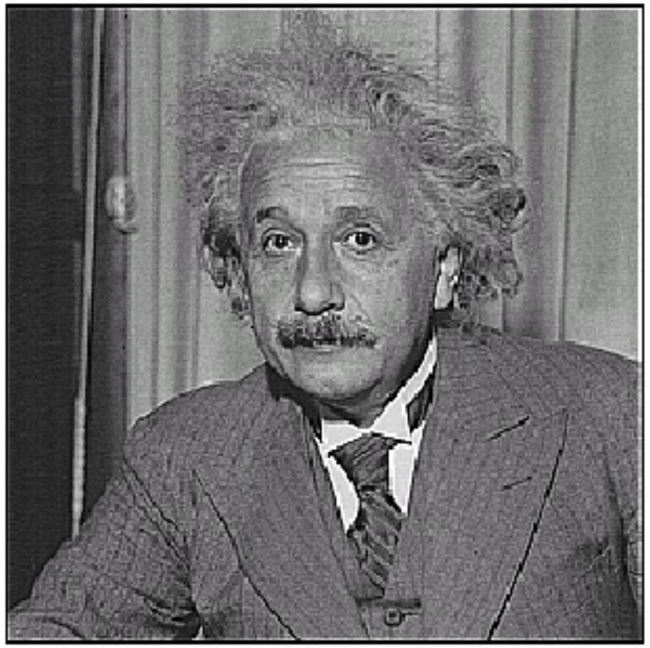
Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

?

4





before after

Convoluzione

- Definizione generale:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(u)g(x-u)du$$

Segnale filtrato

Filtro

Segnale di input

Convoluzione

- Il filtering come convoluzione

$$(f * I)(x,y) = \sum_{i,j=-\infty}^{\infty} f(i,j)I(x-i,y-j)$$

Segnale filtrato

Filtro

Segnale di input

Convoluzione e correlazione

- Convoluzione

$$(f * I)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i, j) I(x - i, y - j)$$

- Correlazione

$$(f \otimes I)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i, j) I(x + i, y + j)$$

Applicato al contrario

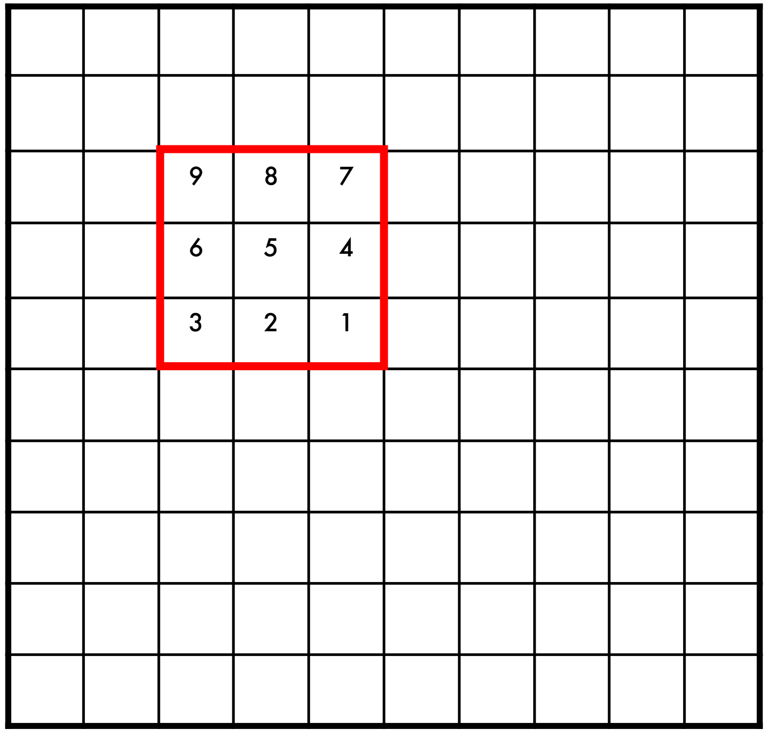
- Nessuna differenza se il filtro è simmetrico

Convoluzione e correlazione

$$(f * I)(x, y) = \sum_{i,j=-\infty}^{\infty} f(i,j) I(x-i, y-j)$$

- Convoluzione

1	2	3
4	5	6
7	8	9

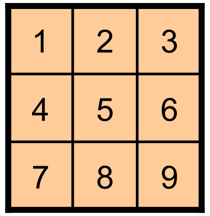


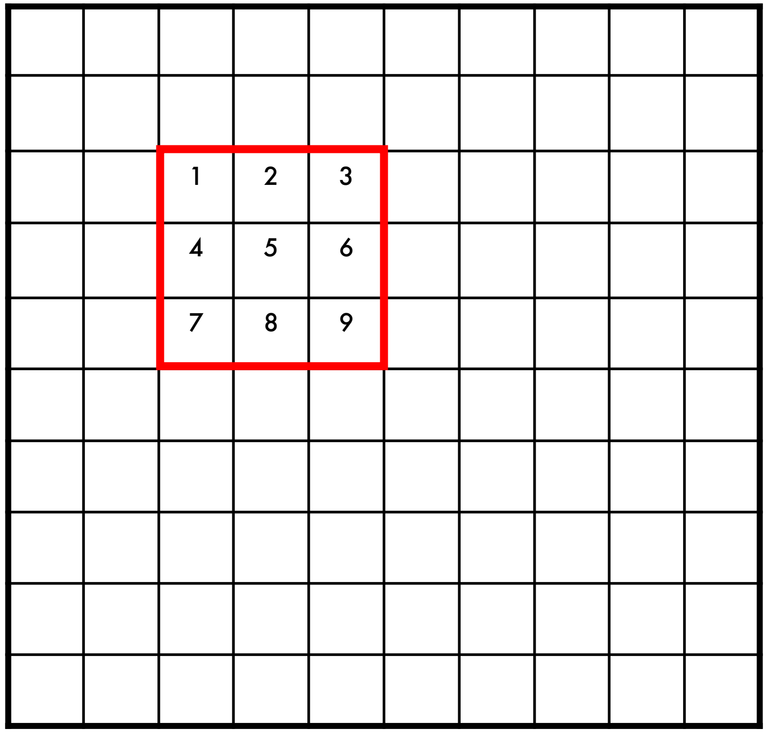
$$(f * I)(3,3) = f(-1,-1) * I(4,4) + f(-1,0) * I(4,3) + f(-1,1) * I(4,2)$$
$$+f(0,-1) * I(3,4) + f(0,0) * I(3,3) + f(0,1) * I(3,3)$$
$$+f(1,-1) * I(2,4) + f(1,0) * I(2,3) + f(1,1) * I(2,2)$$

Convoluzione e correlazione

$$(f \otimes I)(x,y) = \sum_{i,j=-\infty}^{\infty} f(i,j)I(x+i,y+j)$$

- Correlazione





$$\begin{aligned}(f \otimes I)(3,3) = {} & f(-1,-1) * I(2,2) + f(-1,0) * I(2,3) + f(-1,1) * I(2,4) \\ & + f(0,-1) * I(3,2) + f(0,0) * I(3,3) + f(0,1) * I(3,4) \\ & + f(1,-1) * I(4,2) + f(1,0) * I(4,3) + f(1,1) * I(4,4)\end{aligned}$$

Proprietà

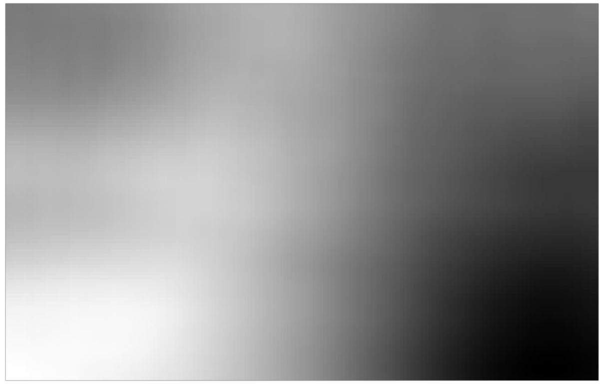
- Commutativa: $a * b = b * a$
- Associativa: $a * (b * c) = (a * b) * c$
  - La correlazione non lo è (effetto rotazione)
- Si distribuisce: $a * (b + c) = (a * b) + (a * c)$
- lineare: $ka * b = a * kb = k\,(a * b)$
- Identità: sull’impulso unitario $e = [0, 0, 1, 0, 0]$, $e * a = a$

Convoluzione e correlazione

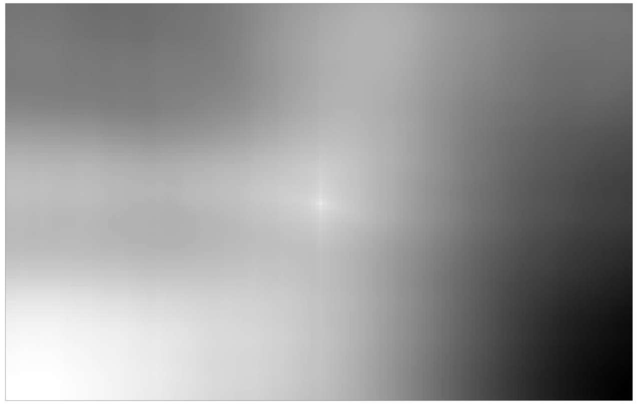
- $A = B * B$
- $C = B \otimes B$

B

A

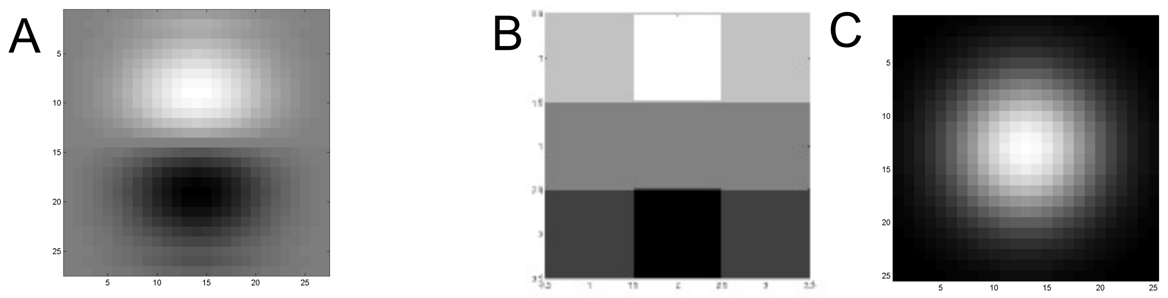


C



Convoluzione e correlazione

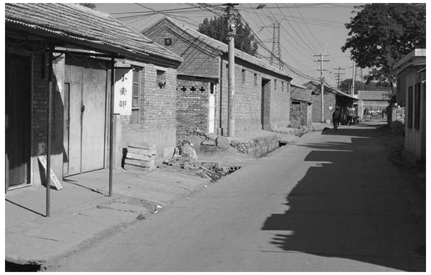
- $A = B \otimes C$
  - “because it kind of looks like it.”



C è un filtro Gaussiano

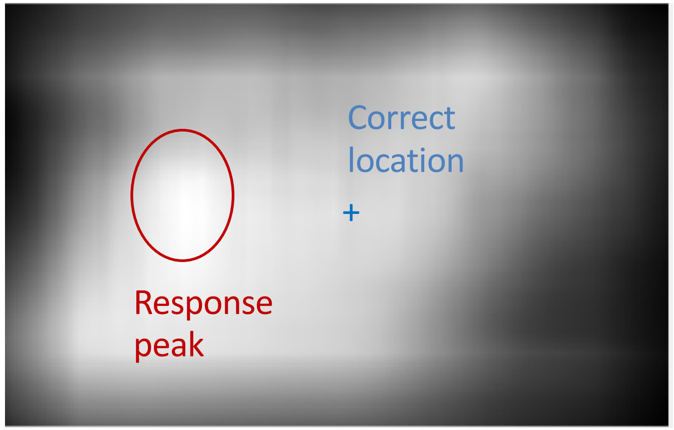
- Se il filtro ‘assomiglia’ all’immagine = ‘template matching’
  - Confronta un’immagine con quello che vuoi trovare, in tutte le regioni.
  - L’asimmetria acquista un senso

D (275 x 175 pixels)



f
61 x 61

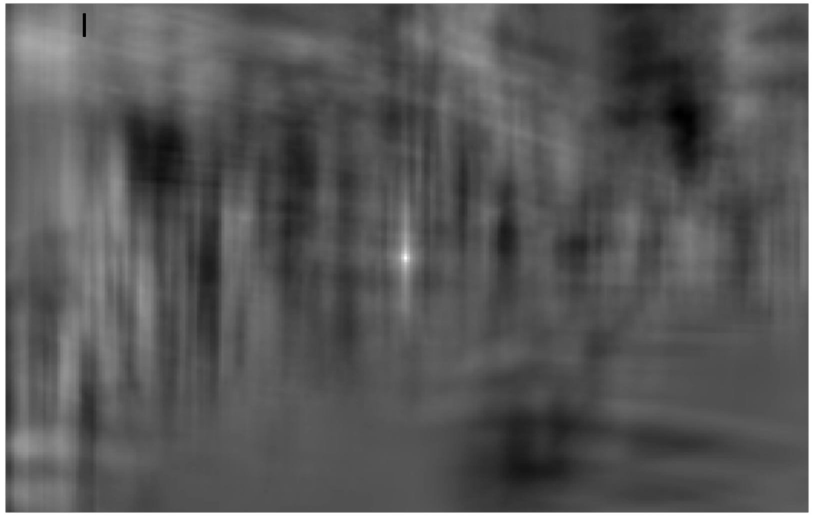
I



D (275 x 175 pixels)

f
61 x 61

I



Filtri separabili

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

- Un filtro è **separabile** se lo stesso effetto può essere ottenuto dall'applicazione in sequenza di due filtri più semplici

Filtri separabili

- Perché è utile?
  - Immagine MxN, filtro PxQ
  - 2D convolution: ~MNPQ addizioni/moltiplicazioni
  - Separable 2D: ~MN(P+Q) addizioni/moltiplicazioni
  - Speed up = PQ/(P+Q)

- Filtro 9x9 = ~4.5x più veloce

Componenti a bassa ed alta frequenza

- Informalmente, le ***frequenze*** di una immagine sono una misura di quanto l'intensità varia con la distanza
  - Le componenti ad *alta frequenza* sono associate a grandi cambiamenti dell'intensità entro piccole distanze (es. bordi e rumore)
  - Le componenti a *bassa frequenza* sono associate a piccoli cambiamenti dell'intensità (regioni uniformi)

- Terminologia utile per discutere gli effetti di un filtro e scegliere il filtro più appropriato al task

Filtri passa-basso e passa-alto

- **Filtro passa-alto**: fa “passare” le componenti ad alta frequenza e riduce o elimina le componenti a bassa frequenza

- **Filtro passa-basso**: fa “passare” le componenti a bassa frequenza e riduce o elimina le componenti ad alta frequenza

Filtri passa-basso e passa-alto nel dominio spaziale

- **Filtro passa-basso** (es., average filter):
  - La somma dei coefficienti vale 1 → regioni uniformi preservate e non uniformi tendono ad uniforme
  - Offusca sia i bordi che il rumore

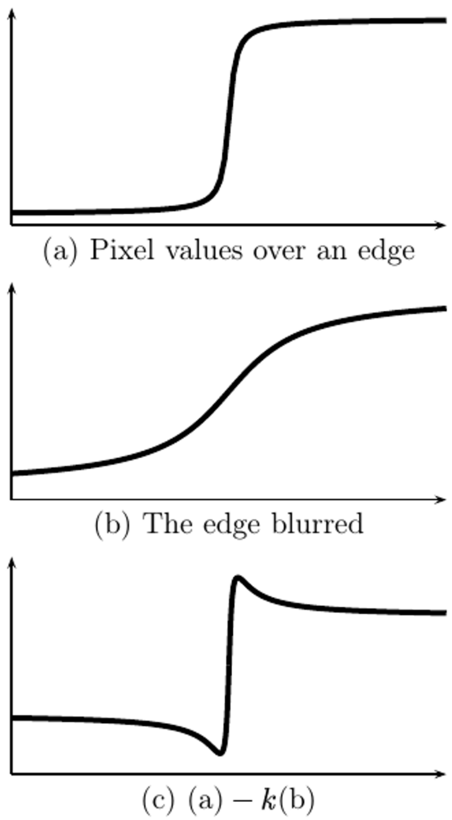
$$f = \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

- **Filtro passa-alto**:
  - La somma dei coefficienti 0 → la risposta sulle componenti a bassa frequenza è prossima a zero

$$f = \frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Sharpening: Unsharp masking

- *Sharpening*: evidenziare transizioni d'intensità
- *Unsharp masking*:
  - Appica average filter all'immagine
  - Sottrai l'immagine filtrata da quella originale (maschera)
  - Aggiungi la maschera opportunamente pesata all'immagine originale



Unsharp masking

$$I_m = I - h_{BLUR} * I = (h_{ID} - h_{BLUR}) * I = g_m * I$$

$$I_{res} = I + k \cdot I_m = I + k \cdot g_m * I = (h_{ID} + k \cdot g_m) * I$$

$g_m = (h_{ID} - h_{BLUR})$ è un filtro passa-alto

- Esempio:

$$w = h \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

h=2 → unsharp masking

- *k* = 1 → unsharp masking
- *k* < 1 → si riduce l'importanza della maschera
- *k* > 1 → highboost filtering

Gestire i risultati di un filtro

- L'applicazione di un filtro può produrre valori al di fuori dell'intervallo previsto per le intensità
  - *Clipping*
  - *Scaling*
  - Utilizzo diretto andando a sottrarre/sommare immagine di partenza
  - Dividere per una costante da determinare caso per caso