

Elaborazione di Segnali e Immagini (ESI) LABORATORIO

Lezione 4

Manuele Bicego

Corso di Laurea in Informatica

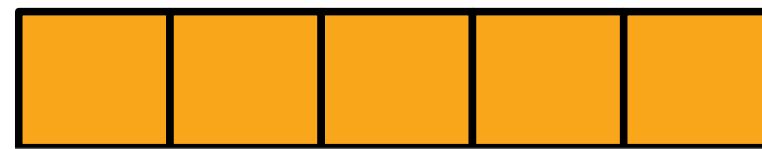
Dipartimento di Informatica - Università di Verona

Cross correlazione 2D

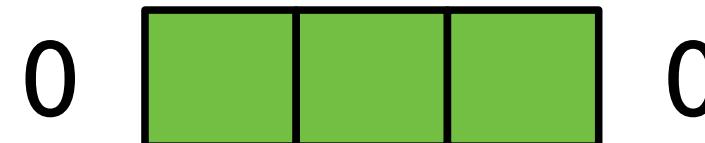
Cross correlazione 1D

$$x_1 \otimes x_2(n) = \sum_{k=-\infty}^{+\infty} x_1^*(k)x_2(k-n)$$

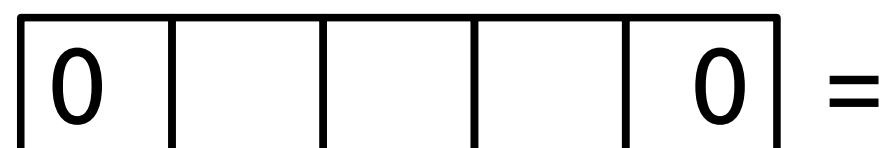
- Cross correlazione per lag =0: nessuno spostamento, moltiplicazione punto a punto e somma



Moltiplicazione punto a punto



= = = = =

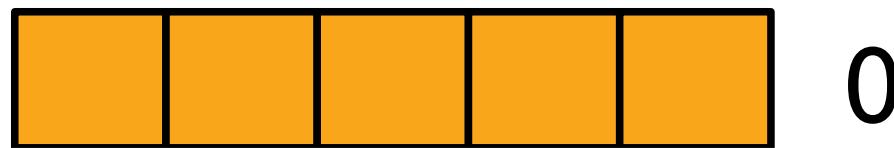


Cross
correlazione
per lag = 0

$$x_1 \otimes x_2(n) = \sum_{k=-\infty}^{+\infty} x_1^*(k)x_2(k-n)$$

- Cross correlazione per un certo lag (o offset): si sposta f2, si moltiplica punto a punto e si somma

Lag = 2



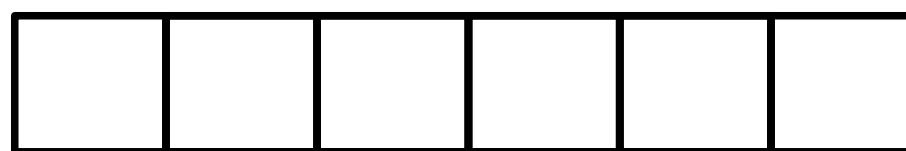
Moltiplicazione punto a punto



0 0 0



= = = = = =

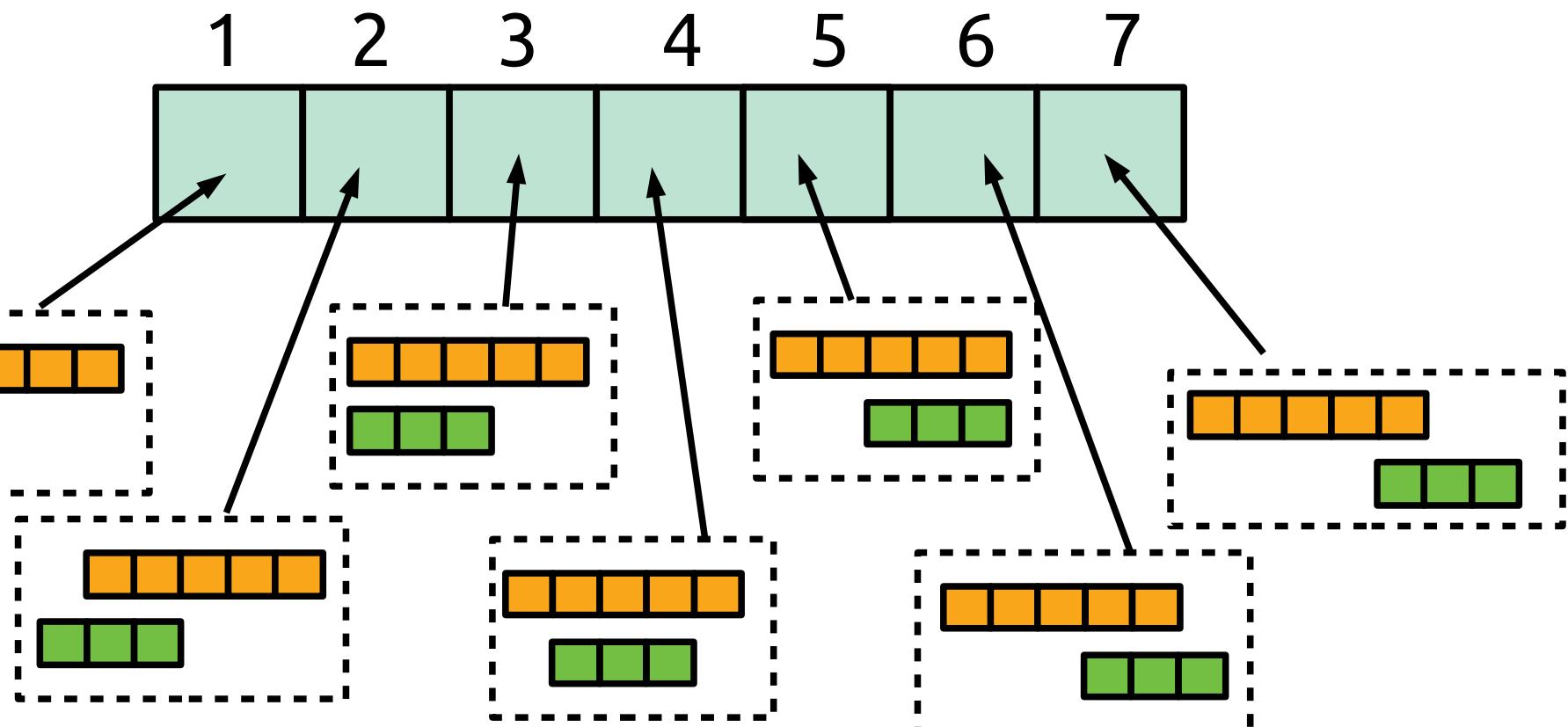


=

Cross
correlazione
per lag = 2

Vettore di cross correlazione:

- Dimensione: $M+N-1$ ($5+3-1 = 7$)



Estensione al 2D

- Nel 2D ci sono 2 lags
 - Lag di riga (spostamento nelle righe)
 - Lag di colonna (spostamento nelle colonne)

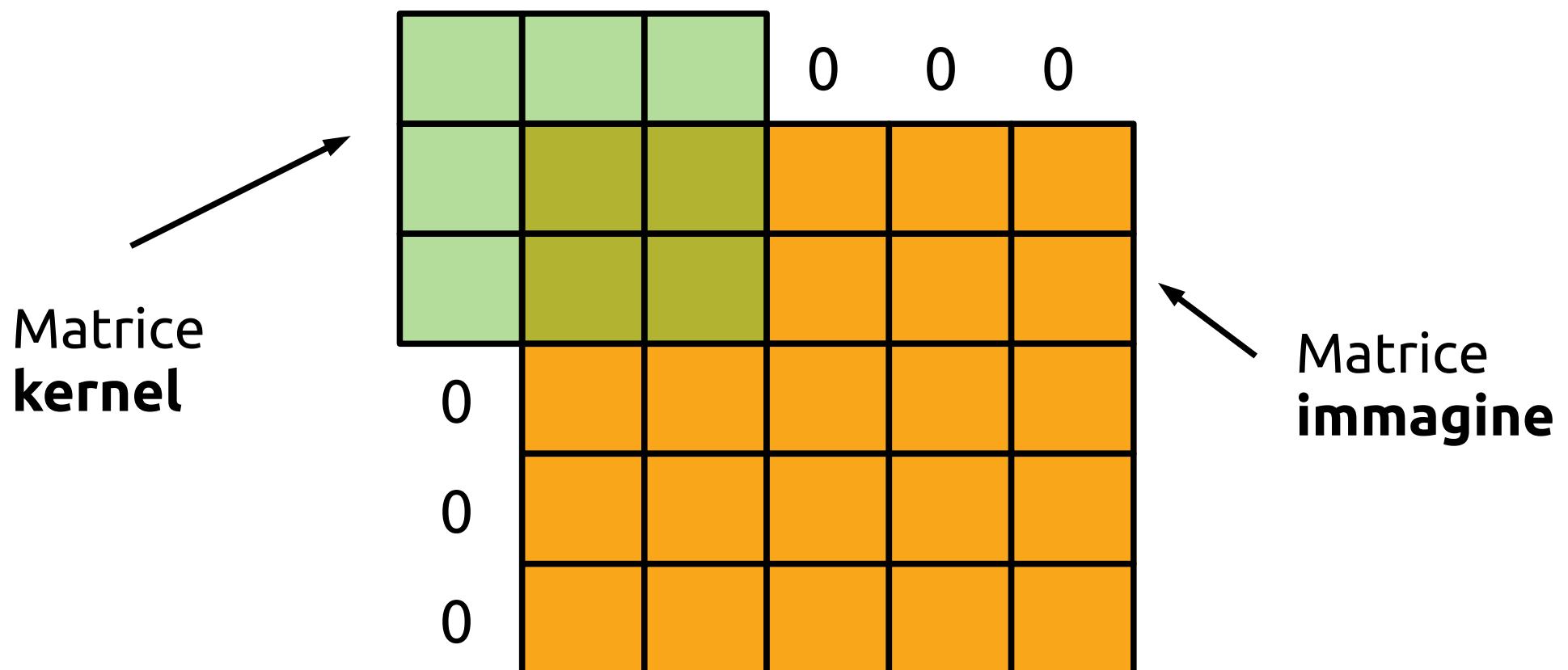
$$x_1 \otimes x_2(m, n) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} x_1(u, v)x_2(u - m, v - n)$$

- Tipicamente la matrice con dimensione più piccola viene chiamata **kernel**, mentre l'altra viene chiamata **immagine**

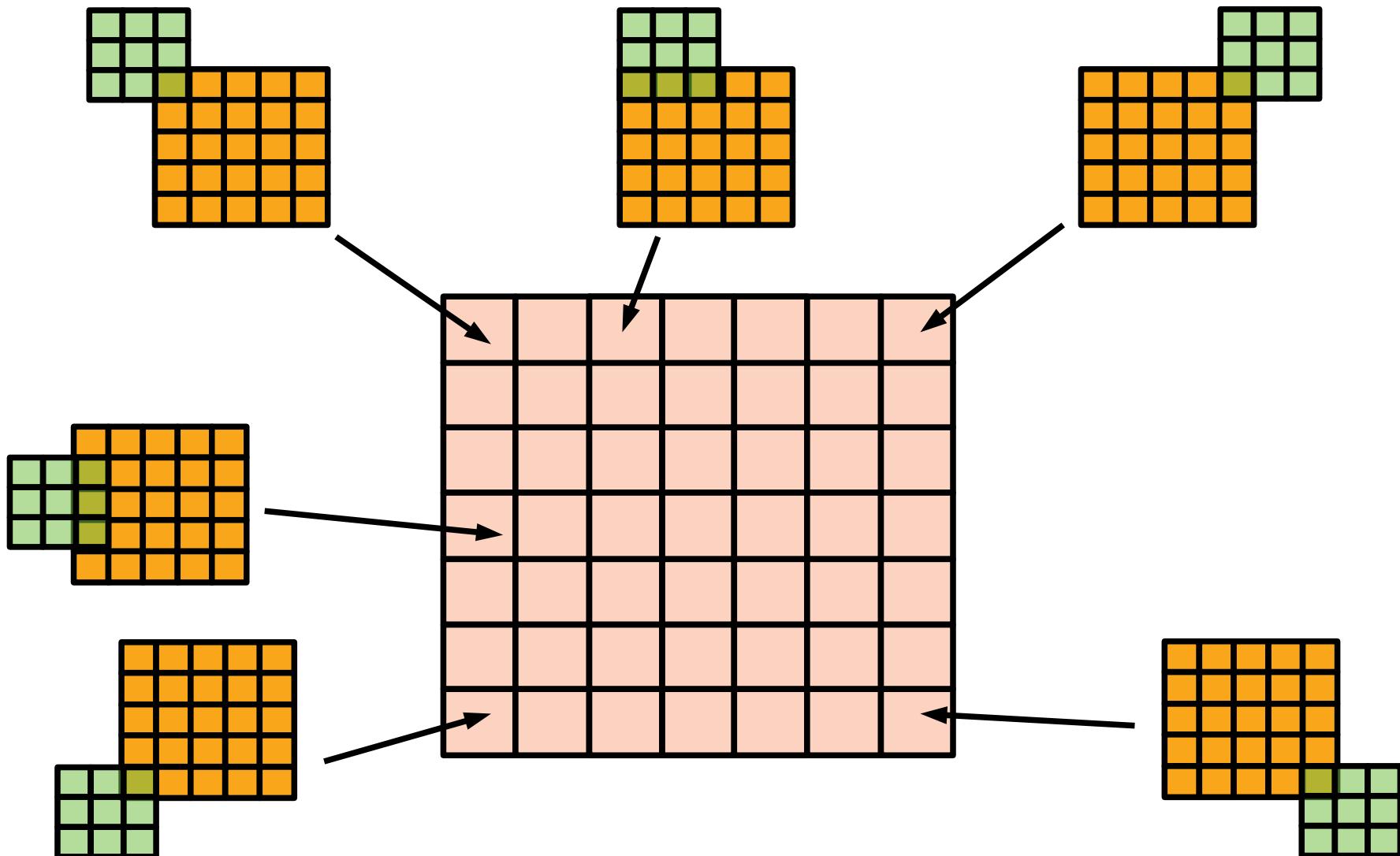
Estensione al 2D

- Come si calcola: è un'estensione del caso 1D
 - Si sposta la matrice kernel di un offset di riga **m** e di un offset di colonna **n**
 - Si fa zero padding
 - Si moltiplica punto a punto e si somma: questo rappresenta il valore di cross-correlazione 2D per il valore **(m,n)**

Si sposta la matrice **kernel**, si fa zero padding, si moltiplica punto a punto e si somma: valore di cross correlazione per quel dato lag

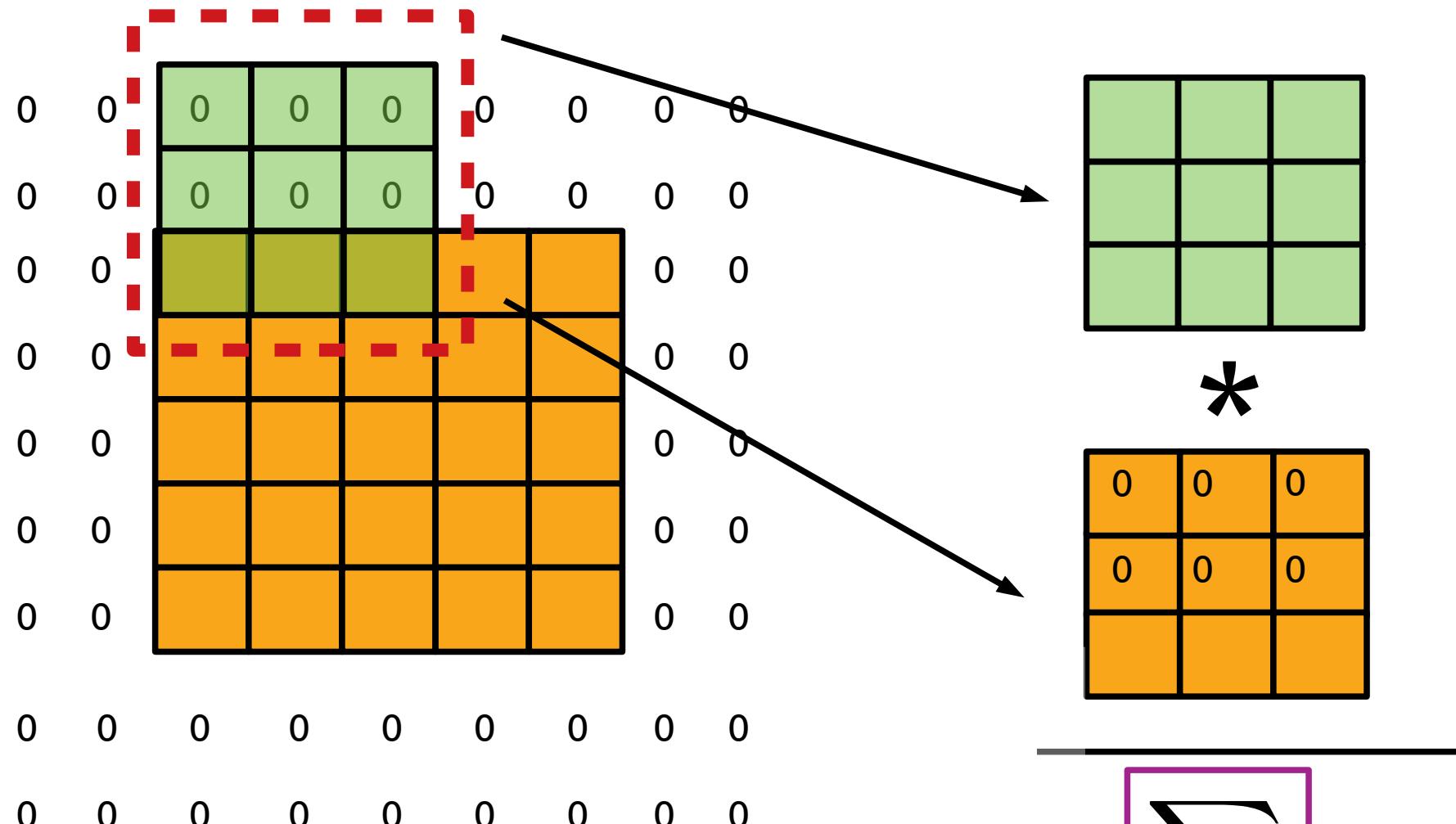


- La matrice di correlazione risultante ha dimensione $(R1+R2-1) \times (C1+C2-1)$
- $R1, C1$ sono le dimensioni della matrice immagine
- $R2, C2$ sono le dimensioni del kernel



- Versione ottimizzata: si fa zero padding della matrice **immagine**
 - Si aggiungono $(R_2 - 1)$ colonne di zeri a dx e a sx e $(C_2 - 1)$ righe di zeri sopra e sotto

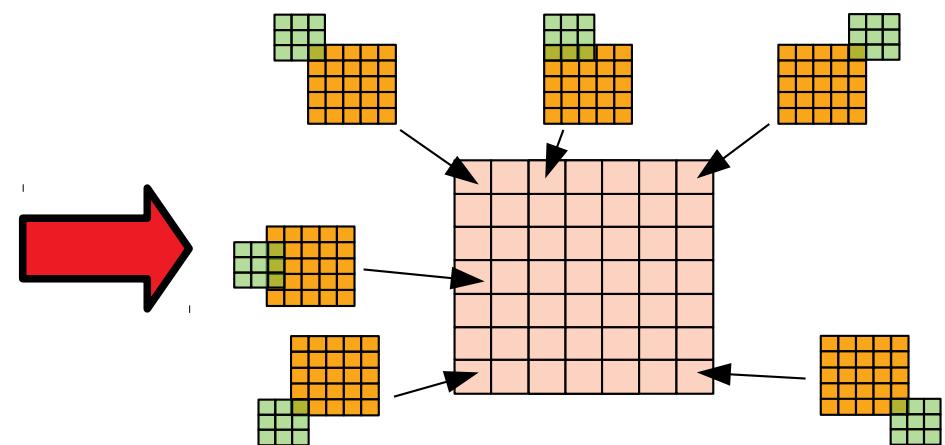
- Per un determinato lag: si posiziona la matrice di kernel, si estrae la corrispondente parte della matrice **immagine** con lo zero padding, si moltiplica punto a punto e si somma



Somma

- Ripetendo queste operazioni per tutti i possibili lag si ottiene la matrice di cross correlazione

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

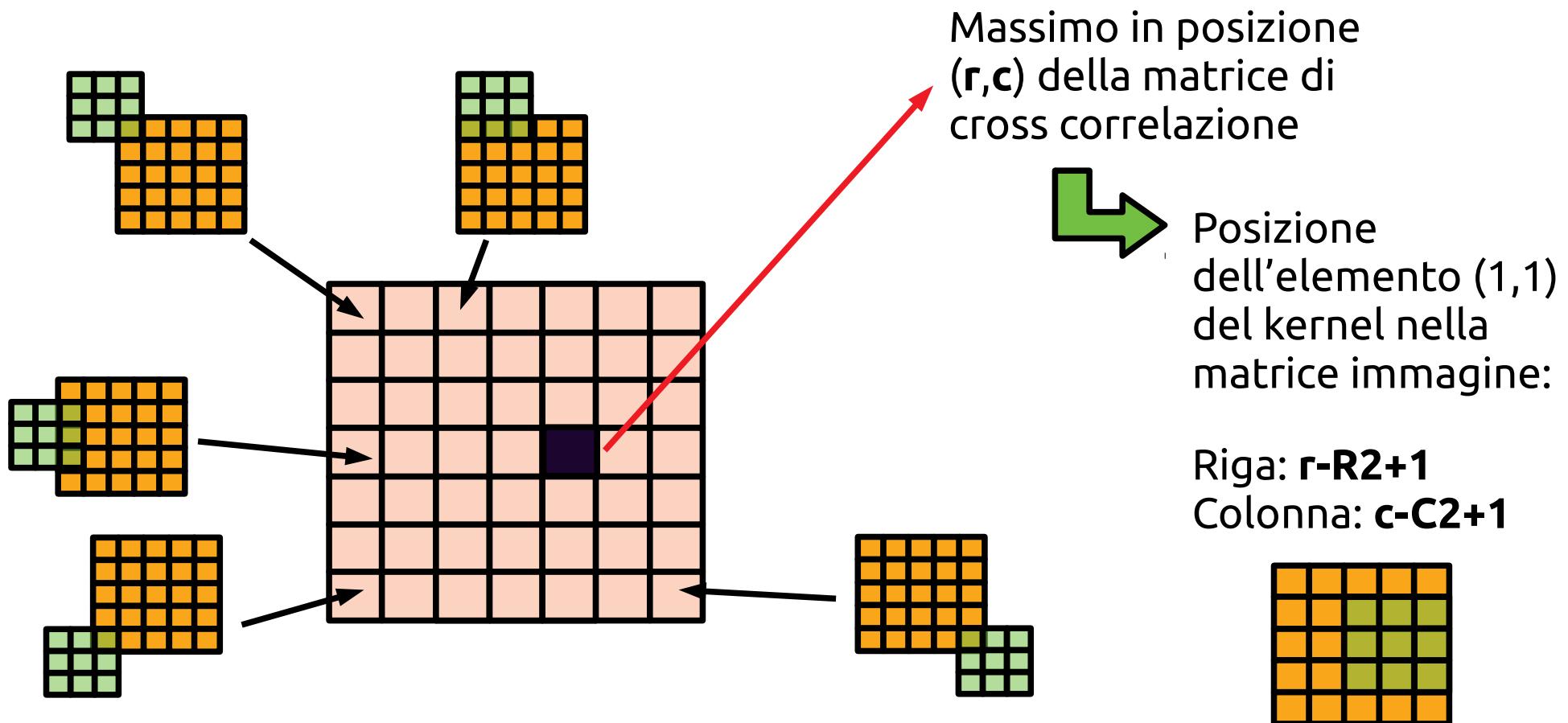


dalla teoria

$$\begin{array}{c}
 \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline
 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ \hline
 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ \hline
 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ \hline
 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline
 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ \hline
 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline
 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline
 \end{array} \times \begin{array}{|c|c|c|} \hline
 1 & 0 & 1 \\ \hline
 0 & 1 & 0 \\ \hline
 1 & 0 & 1 \\ \hline
 \end{array} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|} \hline
 1 & 4 & 3 & 4 & 1 \\ \hline
 1 & 2 & 4 & 3 & 3 \\ \hline
 1 & 2 & 3 & 4 & 1 \\ \hline
 1 & 3 & 3 & 1 & 1 \\ \hline
 3 & 3 & 1 & 1 & 0 \\ \hline
 \end{array}
 \end{array}$$

Massimo della cross-correlazione

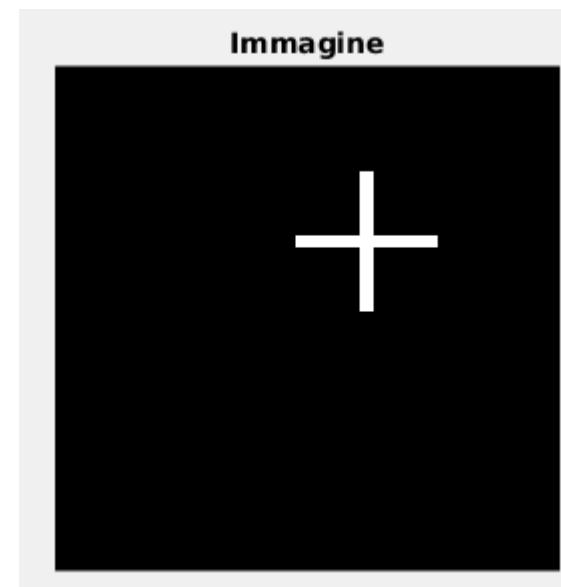
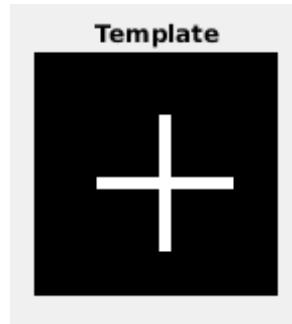
- Se si calcola il massimo del vettore di cross-correlazione, come si determina la corrispondente posizione?



Esercizi principali

Esercizio 1

- Usare la cross correlazione 2D per trovare la posizione del **template** nell'immagine
 - In particolare si richiede di calcolare di quanto (righe-colonne) il template è stato traslato rispetto all'angolo in alto a sinistra dell'immagine



Esercizio 1

- Suggerimento:
 - calcolare la cross correlazione (xcorr2) tra l'immagine e il template (kernel)
 - estrarre le coordinate del massimo
 - recuperare la posizione del kernel
- Controllare anche l'help della funzione xcorr2

Esercizio 2

- Calcolare manualmente la cross correlazione 2D tra le matrici X1 e X2 definite nel file
“Lezione4_EserciziPrincipali.m”
- Confrontare con il risultato del comando matlab
`xcorr2(X1,X2)`
 - Usare la versione ottimizzata descritta nelle diapositive precedenti

Esercizi extra

Esercizio 3

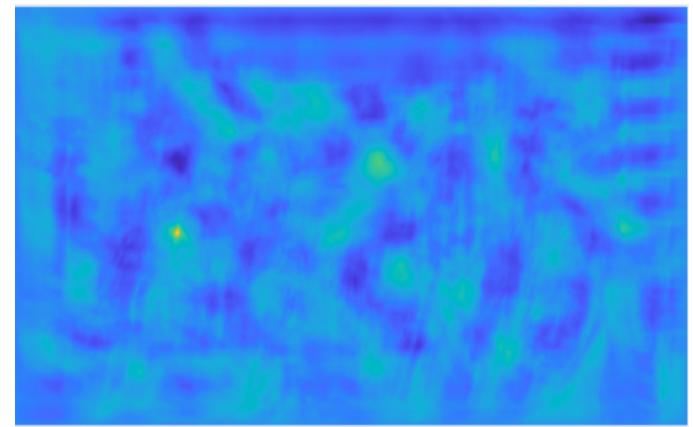
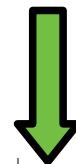
- Caricare l'immagine "puzzle.jpg"
- Ritagliare un pezzo dell'immagine (suggerimento: funzione `imcrop`)
- Utilizzare la cross-correlazione per ritrovare la posizione corretta del pezzo tagliato
- Visualizzare l'immagine originale, il pezzo ritagliato, la matrice di cross correlazione e la figura “risultato”
 - Figura risultato: immagine originale a toni di grigio “attenuata” con sovrapposto il pezzo estratto (a colori) nella posizione corretta



Immagine originale



Porzione



Cross correlazione

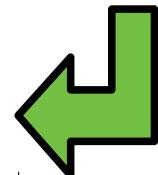


Immagine risultato

Esercizio 3

- Suggerimenti:
 - Per la cross correlazione:
 - convertire l'immagine e il pezzo ritagliato in scala di grigi, la cross correlazione funziona su matrici -- funzione `rgb2gray`
 - usare la cross correlazione normalizzata – funzione `normxcorr2` dell'Image processing toolbox (attenzione all'ordine dell'input)
 - Per la visualizzazione:
 - creare un'immagine con 3 canali (uguali all'immagine originale in scala di grigio)
 - per attenuare moltiplicare tutti i valori dell'immagine per 0.6

Esercizio 4

- Utilizzo avanzato della cross-correlazione 2D normalizzata: trovare difetti su tessuti
 - Caricare l'immagine “tex.jpg”
 - Estrarre alcuni pattern in zone che non contengono difetti
 - Calcolare la cross correlazione dei pattern con l'immagine originale
 - Mediare le matrici di cross correlazione ottenute con i diversi pattern (per una maggiore robustezza)
 - le zone a bassa cross correlazione indicano i difetti

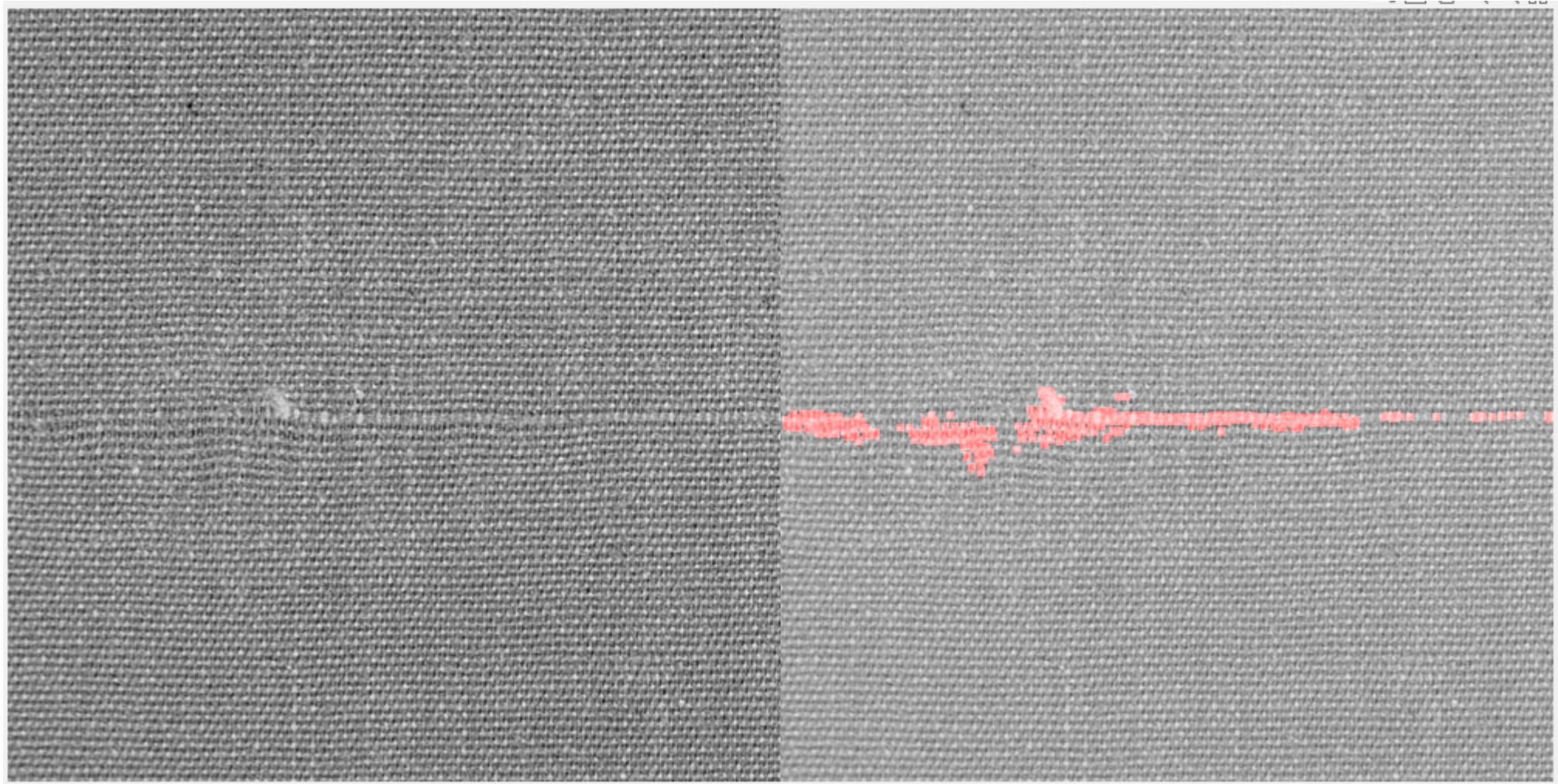


Immagine originale

Immagine con difetto evidenziato

Esercizio 4

- Suggerimento: usare nuovamente la cross correlazione normalizzata (normxcorr2) sulle immagini convertite in toni di grigio
- Partire dalla traccia presente in **Lezione4_EserciziExtra.m**
- Extra: provare la pipeline su altre immagini di difetti prese da internet, modificandola se necessario
 - Cambiare la dimensione, il numero e la posizione dei patterns
 - Cambiare la soglia per il rilevamento

Solo da guardare

- Esempio alla fine di **Lezione4_EserciziExtra.m**:
- Utilizzo della cross correlazione per fare video stabilizzazione
 - Idea: si crea un punto di ancoraggio (una porzione di un frame che si suppone rimanga “stabile”)
 - Si allineano tutti i frame rispetto al punto di ancoraggio
 - Miglior allineamento: cross correlazione 2D