

# Progetto di Computer Vision and Image Processing M

Gardini Erika (0000814361), Fucili Mattia (0000820900), Scalzotto Luca (0000814469)

Anno accademico 2017/2018

## ***Change Detection***

Per realizzare l'algoritmo change detection si possono utilizzare due tipi di approcci (anche in combinazione tra loro):

- differenza temporale tra frames (two-frame difference);
- background subtraction (modeling, differencing).

## ***Differenza temporale tra frames***

Questa tecnica mette in evidenza le differenze tra due 2 frames consecutivi. In questo modo è possibile capire quali parti della scena sono ferme, quindi appartenenti al background e quali sono in movimento, cioè appartenenti al foreground. Tuttavia, il limite di questo approccio è la mancanza di "memoria", cioè se l'oggetto in movimento rimane fermo per due frames consecutivi non viene più rilevato.

Risulta evidente, quindi, che questa tecnica non può essere utilizzata singolarmente.

## ***Background subtraction e segmentazione dell'immagine***

Per poter applicare questa tecnica è necessario, innanzitutto, acquisire un modello del background.

A tal fine, si considerano i valori assunti da ciascun pixel nei primi 100 frames (questo valore è stato mantenuto il più piccolo possibile, come da specifica) e si calcola una stima del background.

Il calcolo della stima del background può essere effettuato in 4 modi diversi:

- media
- moda
- mediana
- apprendimento adattivo

I risultati ottenuti applicando i primi tre metodi sono i seguenti:

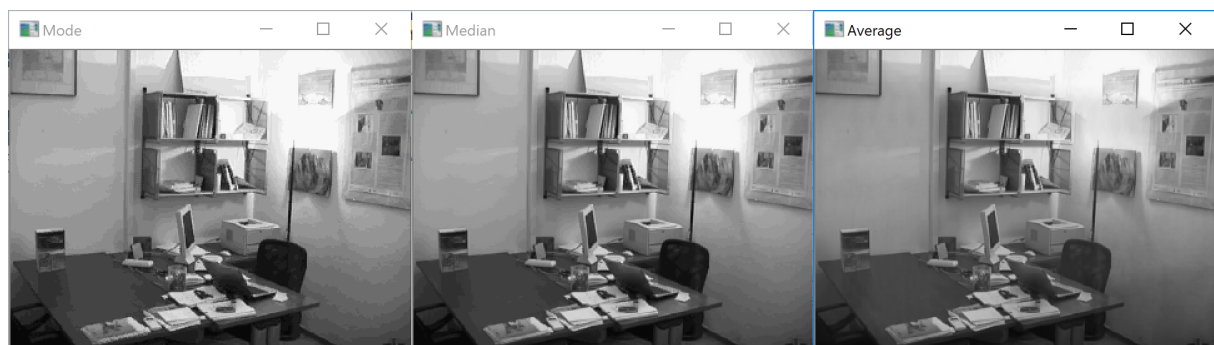


Figura 1: moda, mediana e media

Per il calcolo dei vari background è stato necessario definire degli intervalli di confidenza sui 256 valori assumibili dai pixel.

Per quanto riguarda il metodo di apprendimento adattivo, i risultati ottenibili sono diversi a seconda del fattore di apprendimento che si sceglie di utilizzare.

Infatti, la formula utilizzata per l'aggiornamento del valore dei pixel è la seguente:

$$\mathbf{background.at}(i,j) = \mathbf{background.at}(i,j) + \alpha * (\mathbf{frame.at}(i,j) - \mathbf{background.at}(i,j))$$

dove:

- $\alpha$  è il fattore di apprendimento.

Di seguito si riportano alcuni dei risultati ottenuti:

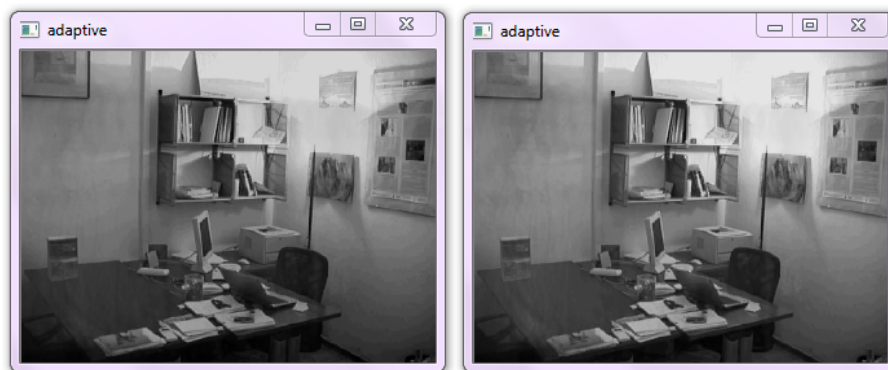


Figura 2: fattore di apprendimento 0,03 a sinistra e 0,05 a destra

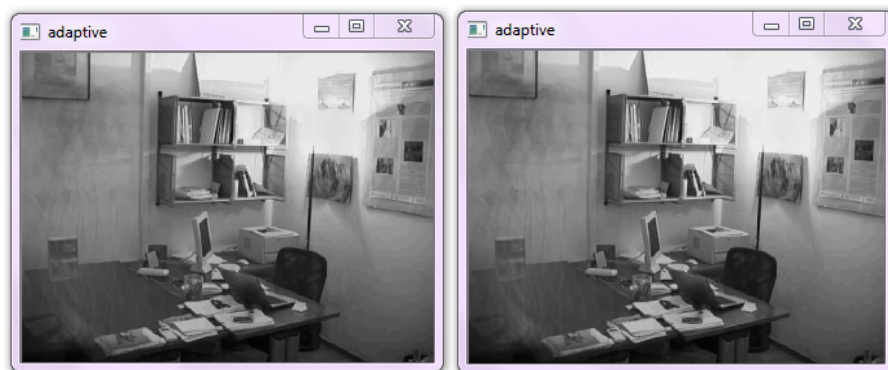


Figura 3: fattore di apprendimento 0,07 a sinistra e 0,09 a destra

Come si può notare, all'aumentare di  $\alpha$ , aumenta l'influenza che ciascun frame ha sul background, quindi si formano delle scie causate dal movimento.

Gli approcci appena illustrati sono definiti "blind" in quanto considerano i valori assunti dai pixel su tutti i 100 frames di training, invece di considerare solo un numero ristretto di pixel.

Si è deciso, in questo caso, di non utilizzare una tecnica "selective" su determinate zone dell'immagine in quanto i risultati ottenuti erano già sufficientemente soddisfacenti.

Per il calcolo del background si è scelto di utilizzare la moda, in quanto il modello di background ottenuto è stato ritenuto migliore rispetto agli altri.

Una volta ricavato il modello del background della scena è possibile procedere con il rilevamento degli oggetti. A tal fine si utilizza la tecnica di background subtraction (two frame difference fra il background ed il frame corrente) per mettere in risalto i cambiamenti dei frames rispetto al modello.

In questo modo, il risultato che si ottiene è già segmentato. La soglia utilizzata per la segmentazione è globale (scelta accuratamente attraverso un tuning pratico) ed è quella che minimizza il rumore causato dal procedimento di background subtraction. Valori più bassi provocano un grado di rumore più elevato, ma la persona risulta più nitida, viceversa per valori più alti.

Altre tecniche di segmentazione viste a lezione non erano applicabili in quanto l'istogramma non risulta essere bimodale. Metodi di segmentazione mediante estrazione dei contorni (algoritmo di Canny, ...) non hanno portato a risultati significativi.

Di seguito si riportano alcune immagini dei risultati ottenuti:

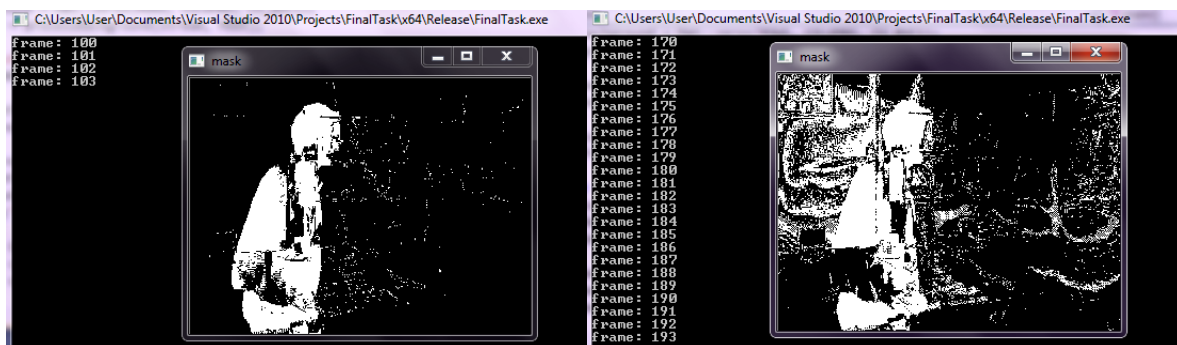


Figura 4: background subtraction frame 103 e frame 193

Come si può notare, il risultato ottenuto non è molto disturbato per la prima parte del video. Le aperture dell'uomo sono dovute a zone del foreground con un colore simile a quelle corrispondenti del background, mentre il rumore è dato in parte dall'approssimazione utilizzata per modellare il background. Entrambe le imperfezioni rilevate possono essere eliminate con degli operatori morfologici.

Nella seconda parte del video, quando avviene la variazione di luce, il rumore diventa tanto elevato da non poter essere più eliminato e compromettendo quindi il risultato.

È pertanto necessario utilizzare una tecnica per l'aggiornamento del background. A tal scopo si è scelto di utilizzare la tecnica di apprendimento adattivo. Il fattore di apprendimento deve essere in questo caso molto elevato (nel caso specifico 0,20) in modo tale che ogni singolo frame modifichi in modo sostanziale il modello del background in modo tale da rimanere robusti alla variazione di luminosità. Ovviamente, però, non si è utilizzata una tecnica "blind" (causerebbe effetti fantasma e scomparsa di oggetti stazionari), ma una tecnica "selective" sulle zone in cui non sono stati rilevati dei movimenti (ottenute selezionando i pixel uguali a 0 dell'immagine calcolata come differenza tra currentFrame e background).

Il background, di volta in volta aggiornato, ha il seguente aspetto nell'ultimo frame:

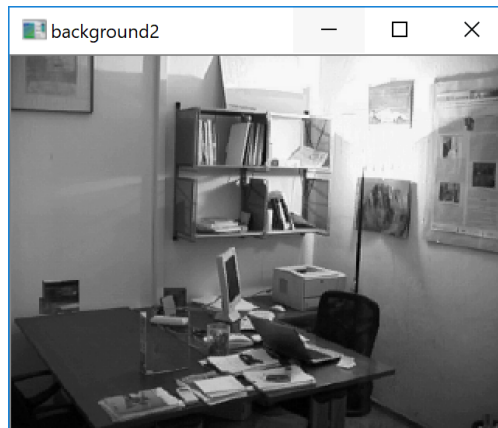


Figura 5: background aggiornato con apprendimento adattivo

Come si può notare, il libro di sinistra (che viene spostato a metà del video e non è quindi più visibile) viene parzialmente eliminato. La parte superiore non viene correttamente eliminata in quanto la differenza di colore tra il frame corrente ed il background è sempre più elevata della soglia. Questo causerà la rilevazione di un falso positivo (come previsto dal task), che verrà eliminato nella seconda parte del progetto.

Per quanto riguarda il secondo oggetto (poco più a destra), il contorno viene parzialmente rilevato come background, poiché la differenza tra il frame corrente ed il background è meno elevata della soglia. I pixel all'interno del contorno, però, hanno ancora il colore del background (differenza tra frame corrente e background maggiore della soglia) e questo consente la corretta rilevazione del blob.

Il risultato ottenuto con l'aggiornamento del background è resistente alle variazioni di luminosità, come si può notare dal seguente risultato:

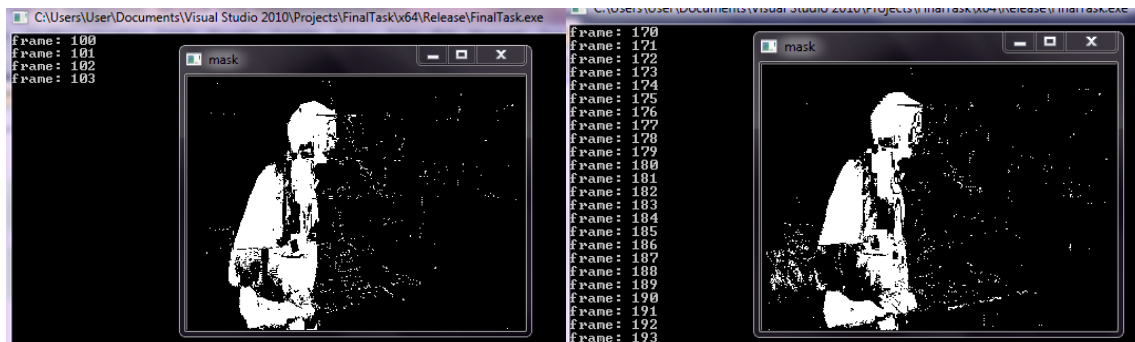


Figura 6: background subtraction frame 103 e 193 con aggiornamento del background

## Operazioni morfologiche

Ridotto il rumore, il risultato è ulteriormente migliorabile attraverso l'applicazione di operatori morfologici; in particolare, come suggerito nelle specifiche, sono state applicate le seguenti "correzioni":

- area opening: con questa tecnica vengono eliminati i blob di piccole dimensioni (in particolare si eliminano tutti i blob di area inferiore a 50);
- closing: con questa tecnica si chiudono tutte le zone aperte del foreground;
- opening: con questa tecnica si va ad eliminare il rumore rimasto, convertendolo in background.

I risultati ottenuti sono i seguenti:



Figura 7: risultato dopo area opening, chiusura ed apertura

Gli operatori morfologici di apertura e chiusura utilizzati sono quelli già forniti dalla libreria di OpenCV, mentre l'area opening è stata realizzata attraverso il metodo *connectedComponentsWithStats* di OpenCV, che individua i blob e restituisce le statistiche associate a ciascuno di essi (informazioni sull'area, sul perimetro, ...).

Le operazioni morfologiche vengono effettuate prima dell'aggiornamento del background in modo tale che quest'ultimo possa essere effettuato sulla maschera morfologicamente migliorata, al fine di ottenere un risultato ancora più soddisfacente.

### ***Blob detection ed eliminazione dei "falsi positivi"***

Infine, viene effettuata la blob detection utilizzando, come per l'area opening, il metodo *connectedComponentsWithStats*. A ciascun blob individuato viene assegnato un colore differente e viene stabilito se un oggetto può essere classificato come "persona" (se l'area è superiore a 1500) o come "other" (se l'area è inferiore a 1500).

Il risultato ottenuto è il seguente:



Figura 8: blob detection

Come previsto, viene erroneamente identificato un oggetto, che non è realmente presente. Per risolvere questo problema è stato calcolato il contorno attraverso l'algoritmo di Canny sia sul frame corrente che sulla maschera ottenuta attraverso la background subtraction. A questo punto, andando ad effettuare la differenza delle due immagini, si otterranno dei contorni nitidi in concomitanza dei "falsi oggetti" (infatti, un contorno presente nella maschera ma non nel

frame corrente, non viene mai eliso), mentre si avranno dei contorni frastagliati in concomitanza degli “oggetti reali” (tanto più frastagliati quanto più i contorni fra la maschera ed il frame corrente coincidono).

Si vanno quindi ad eliminare tutti i blob la cui percentuale di diversità fra i bordi è superiore al 90%.

Qui di seguito si riporta un’immagine esplicativa dell’algoritmo:

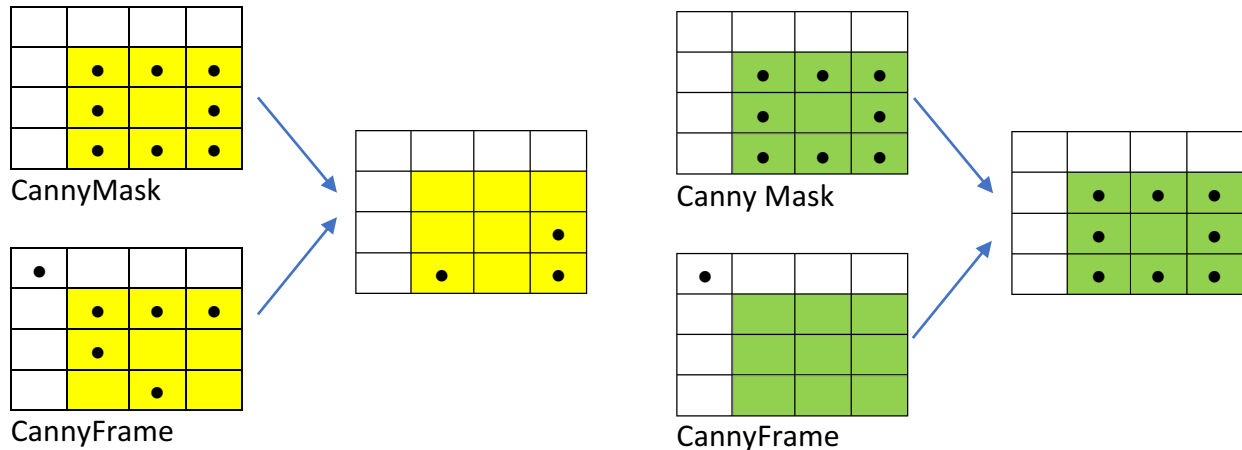


Figura 9: esempio 1 (sinistra), esempio 2 (destra)

$$\text{Percentuale di diversità} = \frac{\text{Numero di pixel diversi nelle due maschere}}{\text{Numero totale di pixel}}$$

Il numero di pixel diversi nelle due maschere viene calcolato solo nella regione in cui è stato identificato il blob (quindi si ha una percentuale di diversità per ciascun blob) e non su tutta l’immagine.

Maggiore è la percentuale di diversità tra i bordi, più è probabile che un oggetto sia falso.

Nel primo esempio la percentuale di diversità è pari a  $\frac{3}{9}$ , quindi l’oggetto è vero, mentre nel secondo esempio la percentuale di diversità vale  $\frac{9}{9}$ , quindi l’oggetto è falso.

Il risultato ottenuto è il seguente:

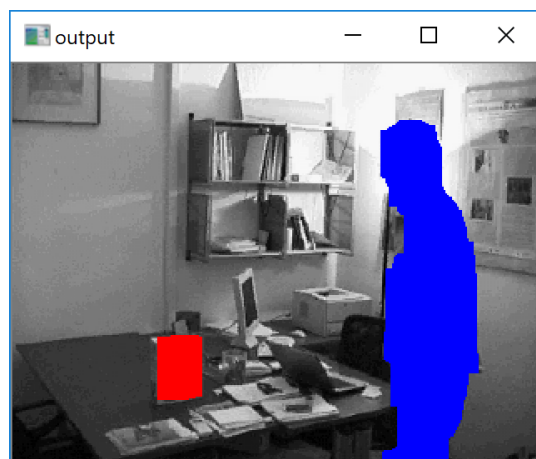


Figura 10: blob detection con eliminazione dei "falsi oggetti"