

6 CONFRONTO TRA ALGORITMI DI “PIXEL LEVEL PROCESSING” PER VSS

6.1 Introduzione

Nel capitolo precedente sono stati descritti alcuni algoritmi di basso livello per un generico sistema di videosorveglianza (VSS). In questo capitolo sono presentati i risultati di una serie di confronti che ne evidenziano le caratteristiche. E' stata effettuata una “preselezione” degli algoritmi (rettangoli “Confronto...” a mezza altezza in Figura 6-1) che ha permesso di scartare quelli con scarse prestazioni.

Le prove hanno messo a confronto algoritmi con caratteristiche omogenee: sono stati comparati tra loro, con le stesse sorgenti di “rumore”, quelli di tipo derivativo; la stessa cosa è stata fatta per i moduli di detection degli algoritmi con riferimento di tipo statistico e per i moduli di detection di quelli con background di tipo “immagine”. Sono state infine misurate le prestazioni dei moduli di aggiornamento applicandoli ai migliori moduli di detection, così come risultati dai confronti precedenti.

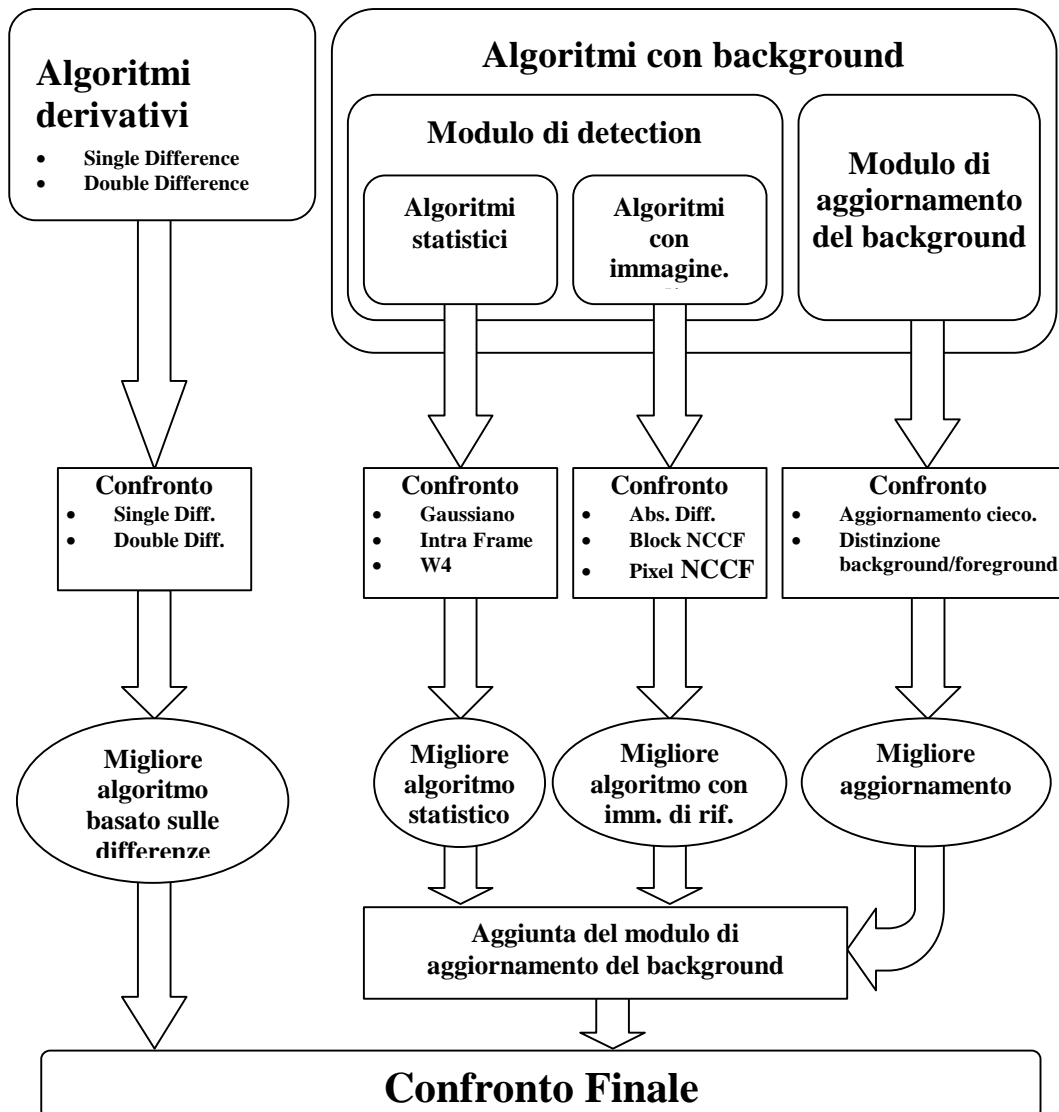


Figura 6-1 Schema seguito per le misure

Gli algoritmi emergenti dalla preselezione sono Double Difference (derivativo), W4 (background statistico), Absolute Difference e Block Based NCCF (background-immagine). Ognuno presenta delle caratteristiche particolari adatte a risolvere problematiche diverse. Ad esempio la presenza di rumore costante *waveing trees*, favorirebbe l'algoritmo W4. Una sequenza con variazioni repentine di illuminazione favorirebbe invece Block Based NCCF, mentre con variazioni luminose e oggetti privi di texture l'algoritmo Absolute Difference presenterebbe il miglior output.

Per eseguire un confronto obiettivo va studiato, per quanto possibile, l'effetto combinato di tutti i fenomeni [TOY99] che possono perturbare l'elaborazione: *waveing trees*, *mimetizzazione* e *light changes*.

In questo capitolo verrà dato conto del “confronto finale” in cui i migliori algoritmi risultanti dai confronti preliminari saranno messi a confronto per individuare quale tra questi rappresenta la risposta migliore all'elaborazione di basso livello.

6.2 Confronto finale

6.2.1 Definizione dei parametri di valutazione

Data la natura binaria degli output le misure valuteranno quanto correttamente gli algoritmi associano a ciascun pixel l'appartenenza a oggetto o a sfondo sulla base di maschere di "ground-truth" segmentate con algoritmi di inseguimento dei contorni su immagini selezionate nelle sequenze (Figura 6-1)

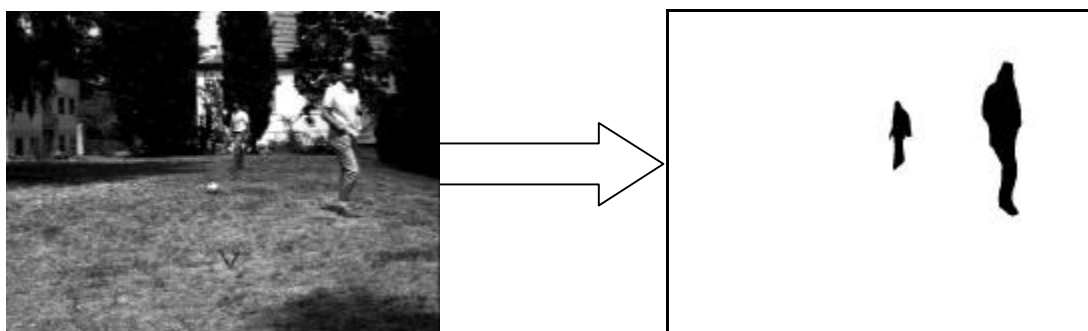


Figura 6-1 Immagine estratta da una sequenza e relativa maschera di *ground-truth*.

Il conteggio dei pixel per le misure è realizzato automaticamente da una funzione presente nell'ambiente "Sentinel". Gli errori possono quindi essere di due tipi:

- Pixel *falsi negativi*
- Pixel *falsi positivi*

(i primi sono pixel-oggetto non rilevati dall'algoritmo. I secondi sono invece pixel-sfondo rilevati come pixel-oggetto).

Il parametro *ErrorePixel%* valuta i *falsi negativi* o i *falsi positivi* esprimendo in percentuale il numero di pixel errati in rapporto all'area di interesse:

$$\text{ErrorePixel}\% = \frac{\text{PixelErrati}}{\text{DimeAreaDiInteresse}} \cdot 100$$

I *falsi negativi* sono relativi alla sagoma degli oggetti, mentre i *falsi positivi* sono riferiti alla restante area dell'immagine. Nello specifico avremo:

$$ErroreFP\% = \frac{PixelFP}{B} \cdot 100 \qquad ErroreFN\% = \frac{PixelFN}{M} \cdot 100$$

dove M e B sono rispettivamente le aree della maschera e del resto dell'immagine. Quindi per ottenere un parametro in grado di stimare globalmente la qualità dell'output sommando falsi positivi e falsi negativi scriveremo:

$$ErroreAssoluto\% = \frac{PixelErratiTotali}{DimImmagine} \cdot 100$$

Esprimendo questo parametro in funzione degli errori percentuali dei *falsi negativi* e *falsi positivi* si ottiene:

$$\begin{aligned} ErroreAssoluto\% &= ErroreFP\% \cdot \frac{B}{B+M} + ErroreFN\% \cdot \frac{M}{B+M} \\ &= ErroreFP\% \cdot B\% + ErroreFN\% \cdot M\% \end{aligned}$$

dove $B\%$ e $M\%$ indicano le percentuali di pixel appartenenti allo sfondo e all'oggetto. Dalla relazione ottenuta si nota come l'errore assoluto tenga conto delle dimensioni degli oggetti rispetto allo sfondo.

Può accadere tuttavia che nella scena siano presenti oggetti di dimensioni particolarmente ridotte; in questo caso il giudizio dell'occhio umano è influenzato maggiormente dalla quantità di errori compiuti nella regione (sfondo o oggetto) più grande. Se gli oggetti presenti costituiscono una minima parte dell'immagine, allora il giudizio è influenzato dai *falsi positivi*. Viceversa, se gli oggetti sono grandi, l'utente tende a dare meno importanza allo sfondo e di più ai *falsi negativi*. Per ottenere un parametro comunque corretto ma più aderente alla valutazione visiva è stato introdotto l'*errore visivo* definito come:

$$ErroreVisivo\% = \frac{(PixelFN + PixelFP)}{2 \cdot \min\{M; B\}} \cdot 100,$$

dove M , B , $PixelFN$ e $PixelFP$ hanno lo stesso significato descritto poc'anzi. Esaminando il legame che c'è tra questo parametro e il precedente si trova:

$$\text{ErroreVisivo}\% = \frac{\text{Min} + \text{Max}}{2 \cdot \text{Min}} \text{ErroreAssoluto}\%$$

$$\begin{aligned}
 \text{Min} &= \text{minimo}\{M; B\} \\
 \text{Max} &= \text{massimo}\{M; B\}
 \end{aligned}$$

Quindi i due parametri differiscono per un fattore moltiplicativo (il termine frazionario) che è sempre > 1 , cioè nell'*errore visivo* tale fattore “amplifica” l'*errore assoluto* in funzione della differenza tra le dimensioni di oggetto e sfondo. In questo modo si ottiene un parametro che risulta più vicino alla percezione umana. Un esempio è rappresentato dalla elaborazione del frame mostrato in Figura 6-2.



Figura 6-2 Immagine elaborata e relativa maschera binaria.

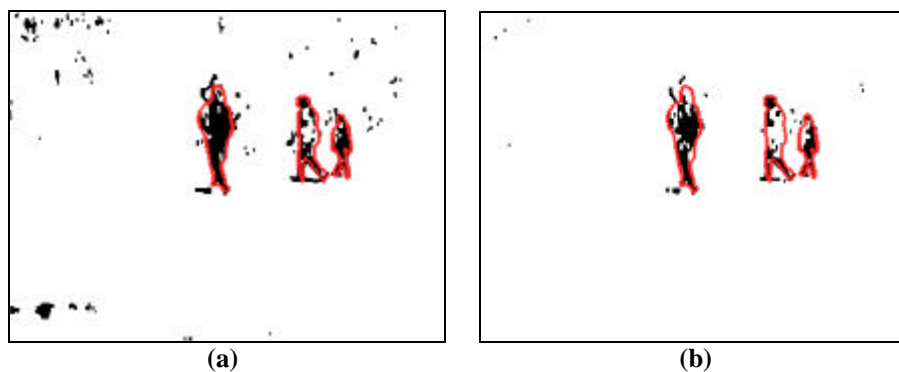


Figura 6-3 Output di prova di due algoritmi diversi.

Esaminando gli output relativi a due algoritmi diversi (Figura 6-3), si percepisce come migliore la Figura 6-3b a causa del minore numero di *falsi positivi*.

	Figura 6-3(a)	Figura 6-3(b)
<i>Errore Assoluto</i>	2,2%	1,5%
<i>Errore visivo</i>	41,7%	28,7%

Tabella 6-1 Valutazione di un output mediante l'*errore assoluto* e l'*errore visivo*.

Confrontando i parametri relativi alle due elaborazioni in Tabella 6-1 si nota come l'*errore assoluto* dia poco risalto alla “differenza” delle prestazioni percepita osservando le figure, al contrario di quanto accade per l'*errore visivo* che la esprimere in maniera più efficace.

6.2.2 Analisi della sequenza di test.

Per le misure del confronto finale si è scelta una tra le 15 sequenze filmate in esterna mediante il sistema di acquisizione di video stereo descritto nel Capitolo 5. Essa ha le proprietà di seguito descritte.

Nome sequenza : **Parco Sc**

Durata: 500 frames

Set: parco Facoltà Ingegneria Bologna

Durata training : 119 frames

Descrizione: Dopo una fase di training con illuminazione costante si verificano variazioni di illuminazione (*light changes*) con ampia escursione mentre gli attori sono in movimento nella scena, dando luogo al fenomeno della *mimetizzazione*. Sullo sfondo sono presenti *waveing trees*.

Questa scelta è motivata dalla presenza in questa sequenza di più elementi di disturbo. In Figura 6-1 è mostrato l'andamento della luminosità media nel corso della durata della sequenza (500 frame).

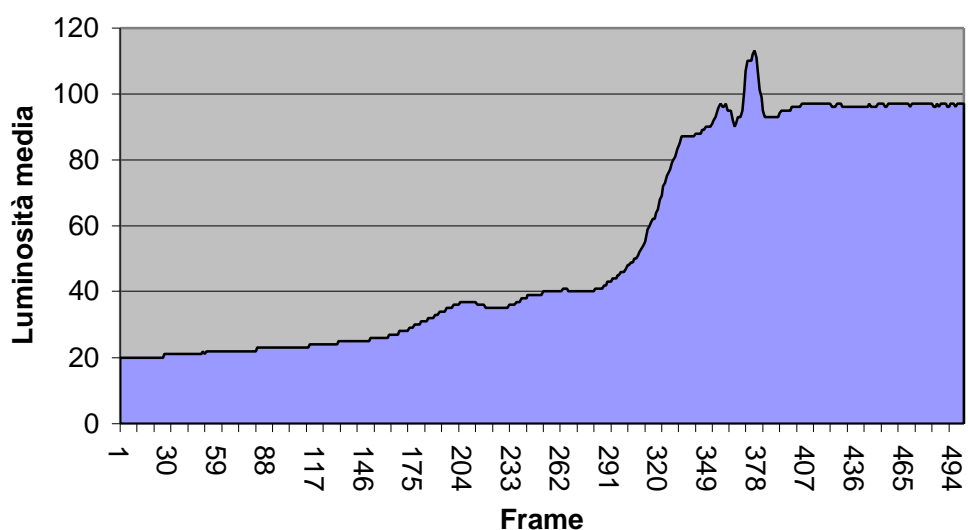


Figura 6-1 Parco Sc - Andamento della luminosità media.

In tale grafico si osserva come l'andamento sia caratterizzato da quattro fasi. La prima (frame 1-150) ha luminosità media stabile: questo garantisce la possibilità di effettuare un periodo di training corretto a favore dell'algoritmo "W4". La presenza di persone in movimento in una zona poco illuminata permette di valutare la sensibilità dell'algoritmo in condizioni di *mimetizzazione*.

La seconda fase (frame 151-290) è caratterizzata da un aumento dei fenomeni perturbatori *waveing trees* dovuta ad una maggiore intensità del vento, mentre la luminosità media subisce un primo aumento. Questa parte ci permetterà di capire come gli algoritmi reagiscono a cambi di illuminazione rapidi ma contenuti. La terza fase (frame 290-400) è quella più critica in quanto si verifica cospicuo incremento di illuminazione (dovuto ad una nube che scopre il sole che determina una variazione media di circa 60 toni sulla scala dei grigi). Questo metterà a dura prova gli algoritmi con background che dovrà essere aggiornato senza generare *falsi positivi* né *falsi negativi*. Inoltre l'incremento di illuminazione ha l'effetto di portare alla luce nuovi particolari relativi agli alberi sullo sfondo, evidenziando in questo modo i fenomeni di *waveing trees*. Il picco presente nel grafico intorno al frame 380 è dovuto al passaggio di un oggetto in corrispondenza dei pixel presi in esame (non è da imputare ad una fonte di rumore).

L’ultima fase (frame 401-500) è caratterizzata luminosità costante e da *waveing trees*. Questa parte sarà usata per valutare eventuali effetti negativi del cambio di illuminazione negli algoritmi con background; in particolare si cercherà di stimare l’attitudine al recupero di eventuali errori dovuti ad un aggiornamento del background durante la penultima fase.

6.2.3 Misure sperimentali.

Le impostazioni dei parametri per ciascun algoritmo sono state eseguite in modo da garantire la migliore detection possibile degli oggetti, minimizzando gli errori. La valutazione dei *falsi negativi* nel frame n -esimo è stata effettuata calcolando in percentuale il numero dei pixel dell’oggetto non individuati (*PixelFN*) su tutta la sua area (*PixelOggetto*):

$$FalsiNegativiErrore\%_n = \frac{PixelFN_n}{PixelOggetto_n} \cdot 100$$

I *falsi positivi* sono espressi conteggiando (sempre in un generico frame n), il numero dei pixel oggetto (*PixelFP*) nell’area dell’immagine non occupata da oggetti (*PixelSfondo* come da maschera di “ground-truth”):

$$FalsiPositiviErrore\%_n = \frac{PixelFP_n}{PixelSfondo_n} \cdot 100$$

Per valutare globalmente gli algoritmi sono stati usati i parametri utilizzati anche per i confronti preliminari ovvero *errore assoluto%* ed *errore visivo%*:

$$ErroreAssoluto\%_n = \frac{PixelFN_n + PixelFP_n}{DimImmagine_n} \cdot 100$$

$$ErroreVisivo\%_n = \frac{(PixelFN_n + PixelFP_n)}{2 \cdot \min\{PixelOggetto_n; PixelSfondo_n\}} \cdot 100$$

I frame per la valutazione estratti dalla sequenza “Parco Sc” sono mostrati in Figura 6-1. Il 135 e il 423 servono per valutare la prima e l’ultima fase della sequenza. I frame 170, 215, 271, 290, 310, 323, e 365 servono a valutare le fasi centrali durante le quali avvengono i cambiamenti di illuminazione.



(a) Frame 1



(b) Frame 135



(c) Frame 170



(d) Frame 215



(e) Frame 271



(f) Frame 290



(g) Frame 310



(h) Frame 323



(i) Frame 365



(j) Frame 423

Figura 6-1 Sequenza “Parco Sc”: (a) frame iniziale ; (b)-(j) frame di valutazione.

Le elaborazioni degli algoritmi sono mostrati in Figura 6-2, Figura 6-3, Figura 6-4, Figura 6-5, Figura 6-6, Figura 6-7, Figura 6-8, Figura 6-9, e Figura 6-10; nero = pixel oggetto risultato dell’elaborazione; rosso = contorno ground-truth.

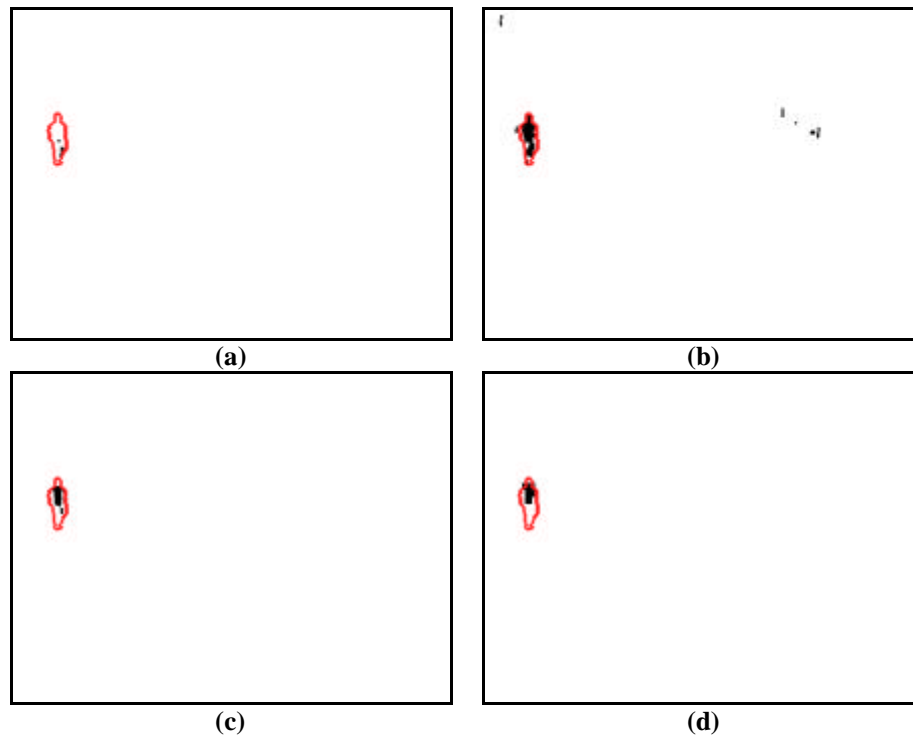


Figura 6-2 Parco Sc - Frame 135. (a) Risultato Double Difference. (b) Risultato W4. (c) Risultato Absolute Difference. (d) Risultato Block Based Nccf.

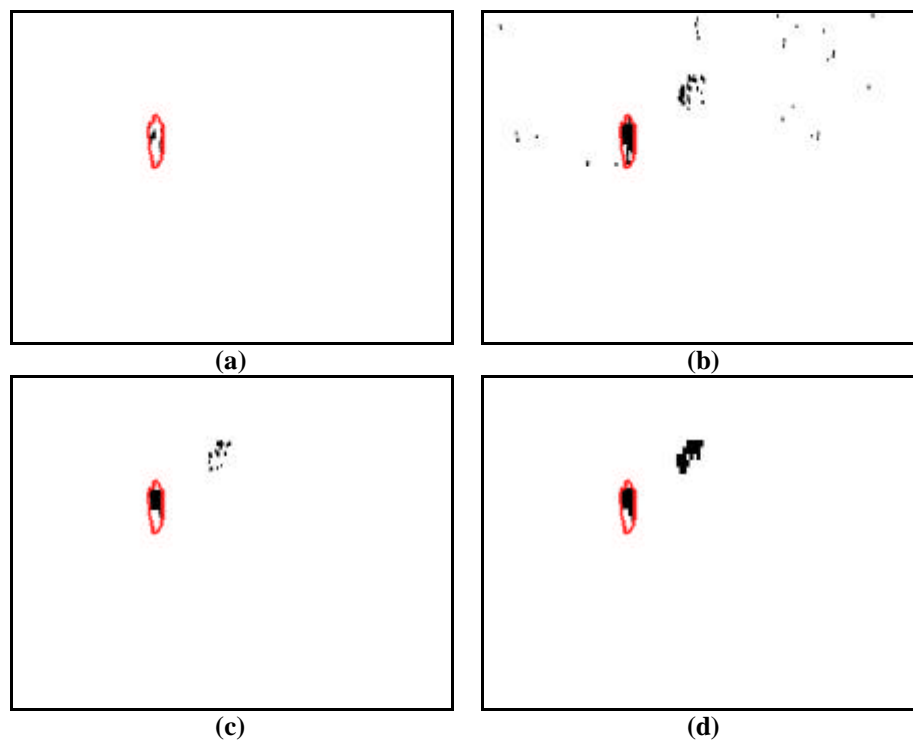


Figura 6-3 Parco Sc - Frame 170. (a) Risultato Double Difference. (b) Risultato W4. (c) Risultato Absolute Difference. (d) Risultato Block Based Nccf.

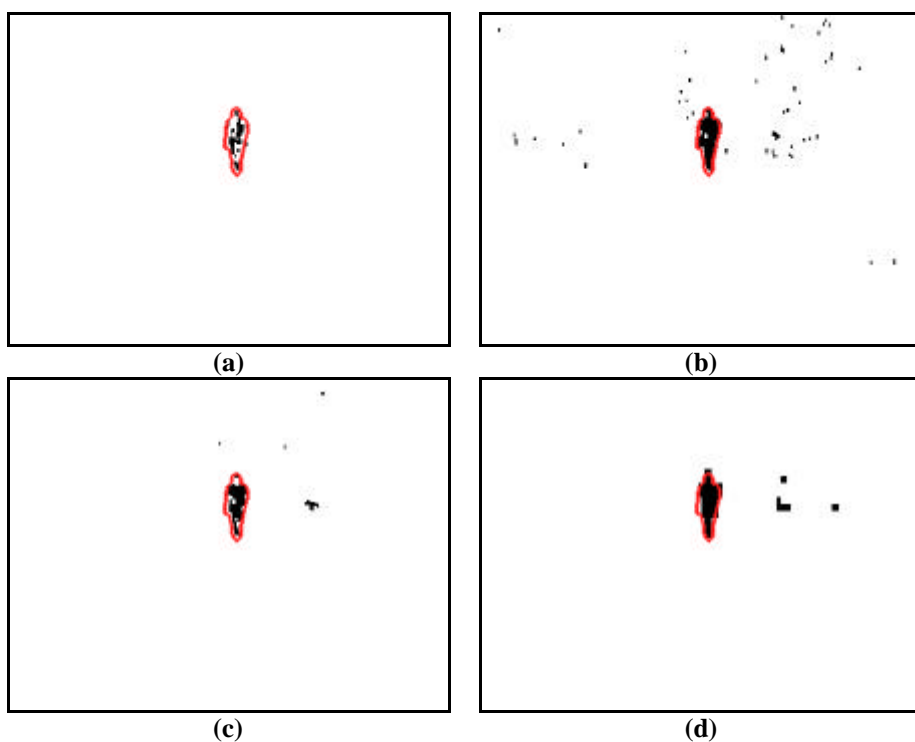


Figura 6-4 Parco Sc - Frame 215. (a) Risultato Double Difference. (b) Risultato W4. (c) Risultato Absolute Difference. (d) Risultato Block Based Nccf.

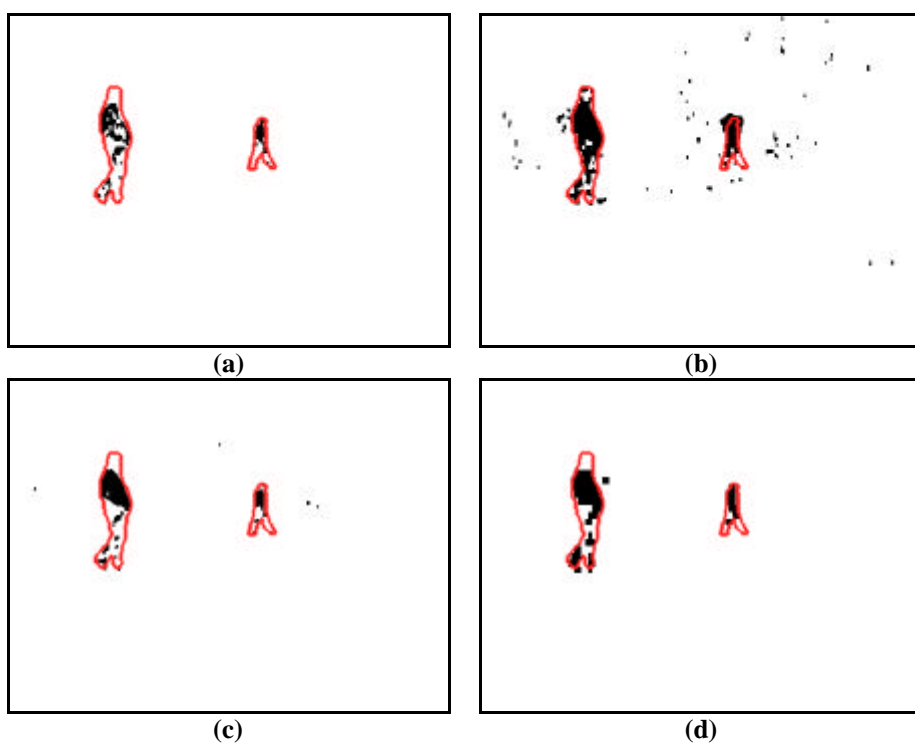


Figura 6-5 Parco Sc - Frame 271. (a) Risultato Double Difference. (b) Risultato W4. (c) Risultato Absolute Difference. (d) Risultato Block Based Nccf.

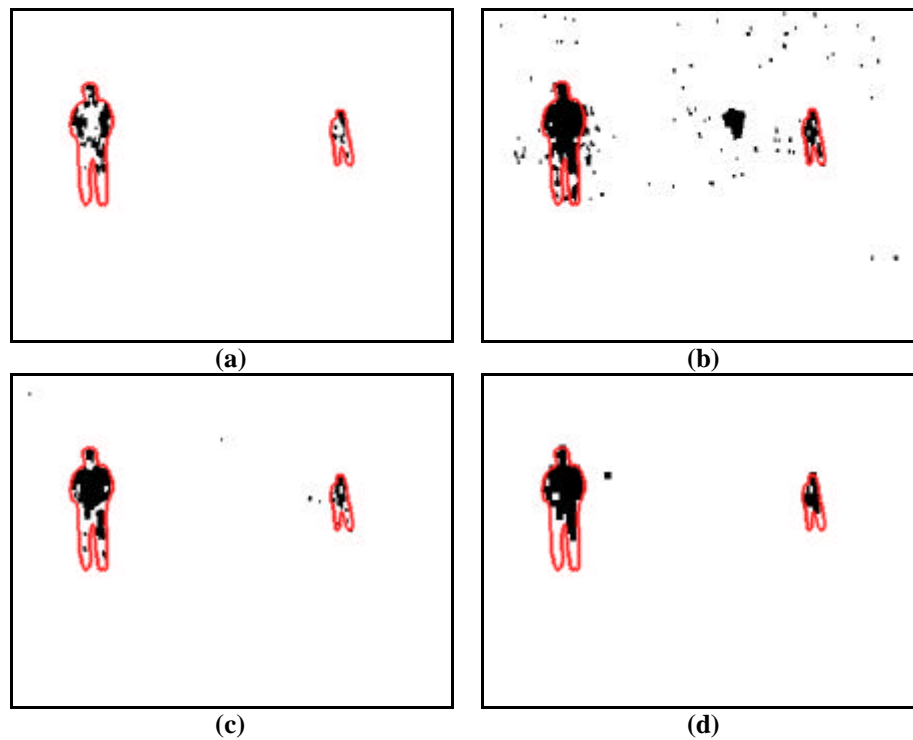


Figura 6-6 Parco Sc - Frame 290. (a) Risultato Double Difference. (b) Risultato W4. (c) Risultato Absolute Difference. (d) Risultato Block Based Nccf.

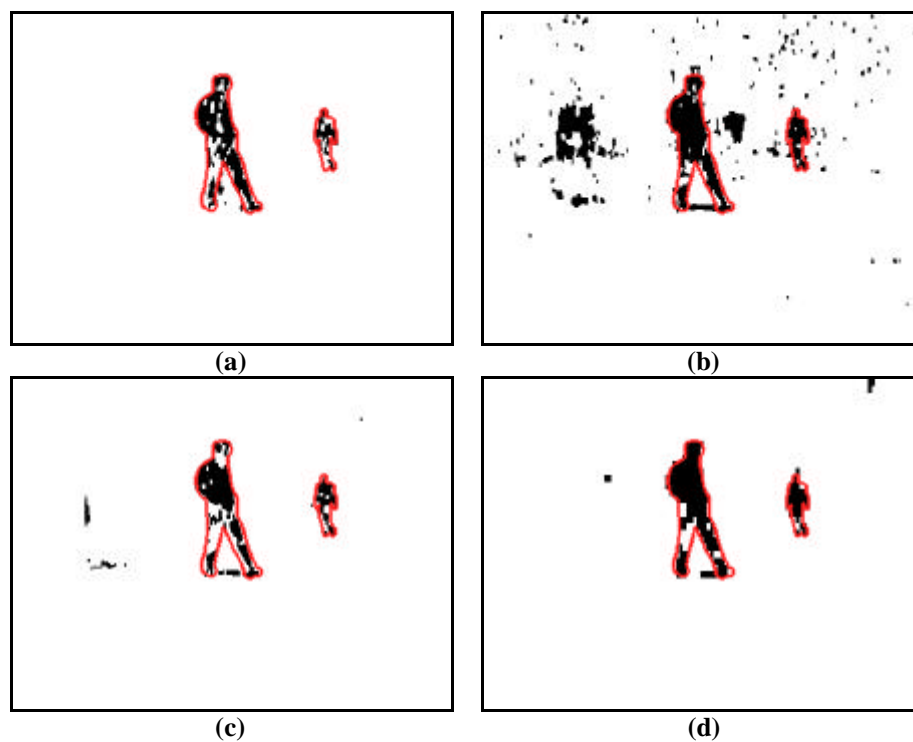


Figura 6-7 Parco Sc - Frame 310. (a) Risultato Double Difference. (b) Risultato W4. (c) Risultato Absolute Difference. (d) Risultato Block Based Nccf.

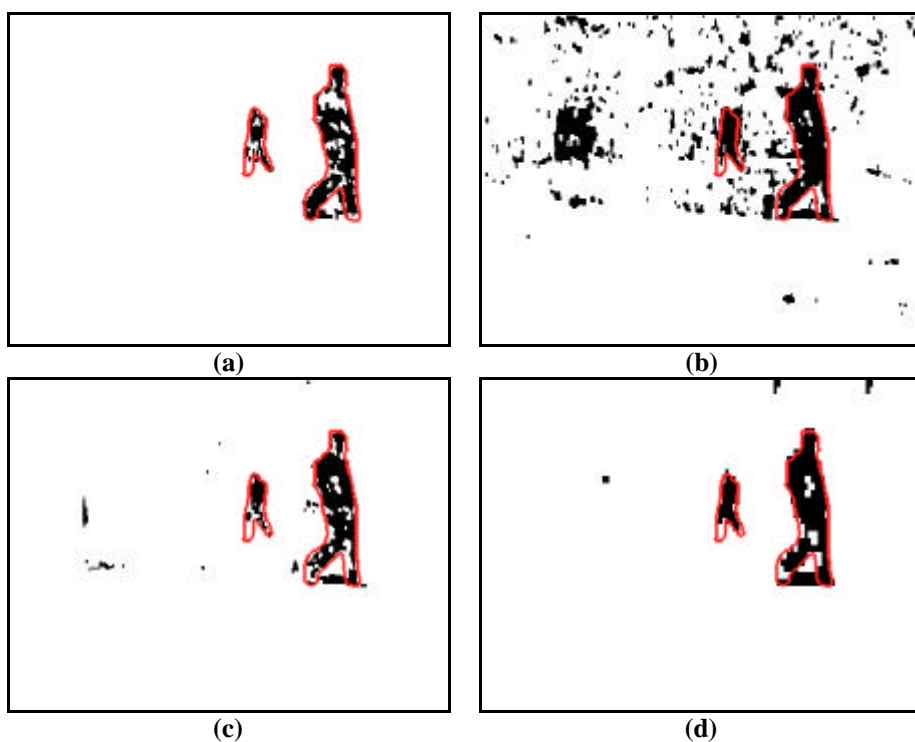


Figura 6-8 Parco Sc - Frame 323. (a) Risultato Double Difference. (b) Risultato W4. (c) Risultato Absolute Difference. (d) Risultato Block Based Nccf.

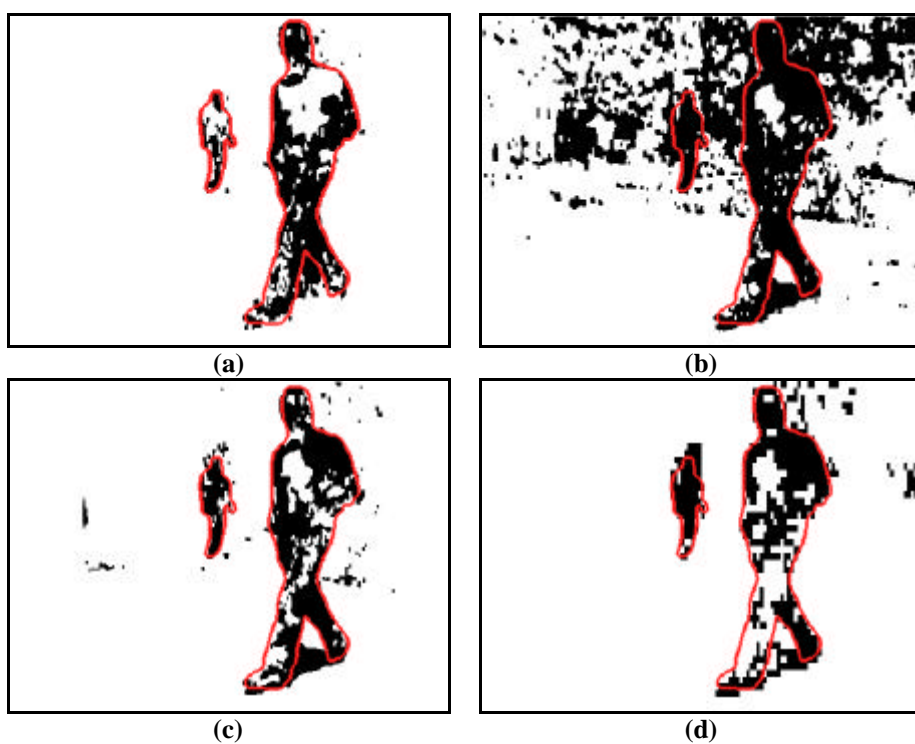


Figura 6-9 Parco Sc - Frame 365. (a) Risultato Double Difference. (b) Risultato W4. (c) Risultato Absolute Difference. (d) Risultato Block Based Nccf.

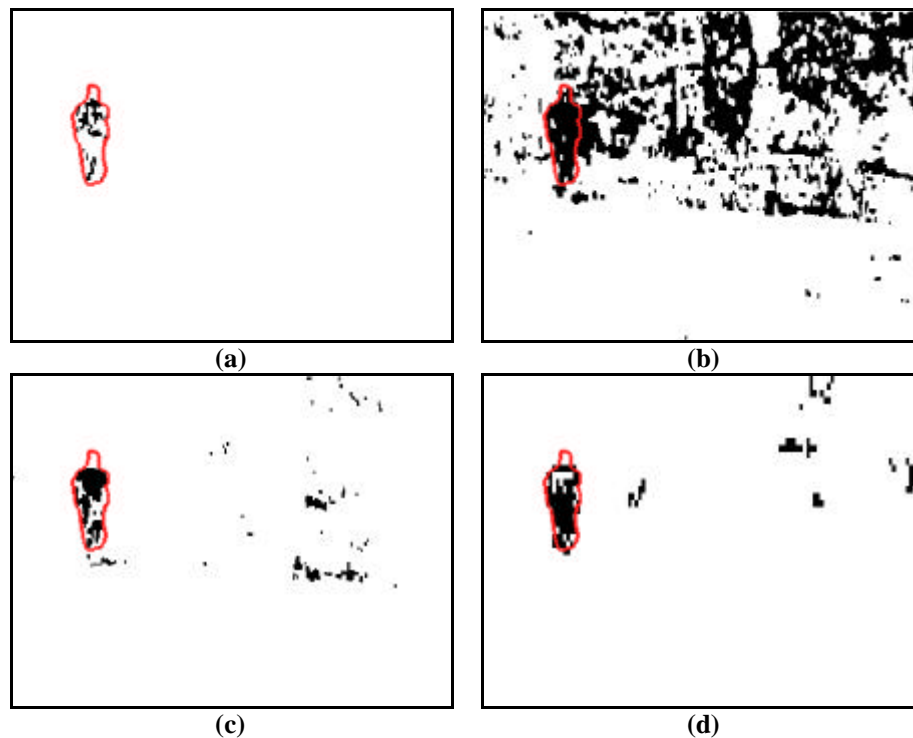


Figura 6-10 Parco Sc - Frame 423. (a) Risultato Double Difference. (b) Risultato W4. (c) Risultato Absolute Difference. (d) Risultato Block Based Nccf.

I risultati delle misure dei *falsi positivi* e *falsi negativi* sono riportati rispettivamente in Tabella 6-1 e Tabella 6-2, mentre quelli relativi all'*errore assoluto* in Tabella 6-3 e quelli relativi all'*errore visivo* in Tabella 6-4.

Osservando le elaborazioni del frame 135 (Figura 6-2) si nota come nella prima fase l'algoritmo W4 presenti la minore percentuale di *falsi negativi* (28,4%) e una buona soppressione di *waveing trees* (0,091%). Sebbene gli altri algoritmi presentino percentuali minori di *falsi positivi* (Tabella 6-1, colonna 1), essi sono caratterizzati da una minore sensibilità come mostrano le alte percentuali di *falsi negativi* (93,1% per la Double Difference, 64,6% per l'Absolute Difference e 68,3% per la Block Based NCCF). I dati relativi all'*errore assoluto* e all'*errore visivo* (colonna 1 della Tabella 6-3 e Tabella 6-4) confermano il buon risultato, percepibile osservando la Figura 6-2.

Osservando i dati relativi alla seconda parte della sequenza (frame 170, 215 e 271) si nota come W4 mantenga il primato della più bassa percentuale di *falsi*

negativi (mediamente 18%, Tabella 6-2, riga 2); la stessa sensibilità che porta questo buon risultato influenza negativamente le prestazioni di W4 durante il cambio di illuminazione evidenziato in Figura 6-3(b), Figura 6-4(b) e Figura 6-5(b) dalla quantità di *falsi positivi* che risultano distribuiti su tutta l'immagine. Dal canto suo la Double Difference risulta immune ai cambi di illuminazione mostrando percentuali bassissime di *falsi positivi* (intorno al 3%), ma mostra limiti nella individuazione di oggetti a causa della *foreground aperture* (Tabella 6-2, riga 1).

Osservando gli output (c) e (d) in Figura 6-3, Figura 6-4 e Figura 6-5 si nota come gli algoritmi Absolute Difference e Block Based NCCF fronteggino con successo i cambi di illuminazione mostrando solo una minima quantità di *falsi positivi*, dovuti principalmente al movimento degli alberi sullo sfondo. Osservando i dati relativi ai *falsi positivi* e *falsi negativi* (Tabella 6-1 e Tabella 6-2, righe 3 e 4) si nota che i due algoritmi hanno un comportamento opposto: l'Absolute Difference risulta più robusto di Block Based NCCF rispetto a *waveing trees*; viceversa quest'ultimo presenta *falsi negativi* in misura mediamente inferiore (28,5% contro 44,8% dell'algoritmo precedente). Osservando l'*errore assoluto* e l'*errore visivo* (Tabella 6-3 e Tabella 6-4, righe 3 e 4) si nota come i due algoritmi abbiano prestazioni simili, con un leggero vantaggio a favore dell'Absolute Difference.

Il frame 290, si trova a cavallo tra la fase appena analizzata e la successiva (caratterizzata da un forte cambio di illuminazione). In questo caso si nota che l'Absolute Difference e la Block Based NCCF presentano il miglior risultato, mentre Double Difference, pur immune ai cambiamenti di illuminazione, presenta un'elevata percentuale di *falsi negativi*.

L'algoritmo W4 è caratterizzato dalla migliore sensibilità che però lo rende più vulnerabile ai cambi di illuminazione.

<i>Falsi Positivi</i>	Frame sequenza Parco Sc								
	135	170	215	271	290	310	323	365	423
Double Difference	0%	0,004%	0,02%	0,06%	0,083%	0,34%	0,48%	1,51%	0,039%
W4	0,091%	0,335%	0,345%	0,812%	1,2%	3,43%	7,1%	22,4%	21%
Absolute Difference	0,004%	0,101%	0,090%	0,068%	0,073%	0,383%	0,586%	2,11%	0,795%
Block Based NCCF	0,028%	0,361%	0,270%	0,274%	0,240%	0,664%	1,003%	3,051%	1,113%

Tabella 6-1 Errore percentuale di *falsi positivi* rispetto all'area dell'immagine non contenente oggetti.

<i>Falsi Negativi</i>	Frame sequenza Parco Sc								
	135	170	215	271	290	310	323	365	423
Double Difference	93,1%	77,9%	60,5%	60,2%	64,2%	32,5%	30,7%	34,7%	72,2%
W4	28,4%	21,4%	11,6%	21,5%	19,5%	14,7%	12,5%	13,4%	9,3%
Absolute Difference	64,6%	47,9%	32,8%	53,8%	40,1%	36,5%	26,9%	28,4%	40,5%
Block Based NCCF	68,3%	35,4%	11,8%	38,3%	32,7%	16,6%	16,6%	35,2%	27,6%

Tabella 6-2 Errore percentuale di *falsi negativi* sugli oggetti di ciascuna immagine.

<i>Errore assoluto</i>	Frame sequenza Parco Sc								
	135	170	215	271	290	310	323	365	423
Double Difference	0,348%	0,227%	0,337%	1,033%	1,52%	1,22%	1,52%	5,4%	1,06%
W4	0,197%	0,409%	0,41%	1,15%	1,61%	3,74%	7,29%	21,3%	20,8%
Absolute Difference	0,245%	0,269%	0,28%	0,938%	0,969%	1,37%	1,49%	5,2%	1,35%
Block Based NCCF	0,283%	0,484%	0,337%	0,89%	0,968%	1,1%	1,54%	6,8%	1,49%

Tabella 6-3 Errore assoluto.

<i>Errore visivo</i>	Frame sequenza Parco Sc								
	135	170	215	271	290	310	323	365	423
Double Difference	46,5%	32,4%	29%	31,9%	33,9%	22,3%	22,1%	22,8%	35,5%
W4	26,3%	58%	35,5%	35,5%	35,9%	68,2%	106%	89,9%	730%
Absolute Difference	32,7%	38,4%	24,1%	29%	21,6%	25%	21,6%	20,1%	48%
Block Based NCCF	37,8%	69,1%	30%	27,5%	21,6%	20,6%	22,3%	28,7%	52,9%

Tabella 6-4 Errore visivo.

L'output di W4 relativo al frame 290 e mostrato in Figura 6-11(b), evidenzia il fenomeno di "printing" del background che si verifica quando un oggetto si muove liberando una porzione di sfondo, subito dopo una variazione di illuminazione: a causa della presenza dell'oggetto non è possibile aggiornare lo sfondo che quindi resta allo stato anteriore alla variazione. Quando la persona si sposta, la zona non aggiornata dà origine a falsi positivi poiché interpretata da W4 quale oggetto. Questo fenomeno è comune a tutti gli algoritmi con il background, ma in questo caso genera errori solo con W4 in quanto esso è caratterizzato da una sensibilità maggiore rispetto agli altri algoritmi con background.

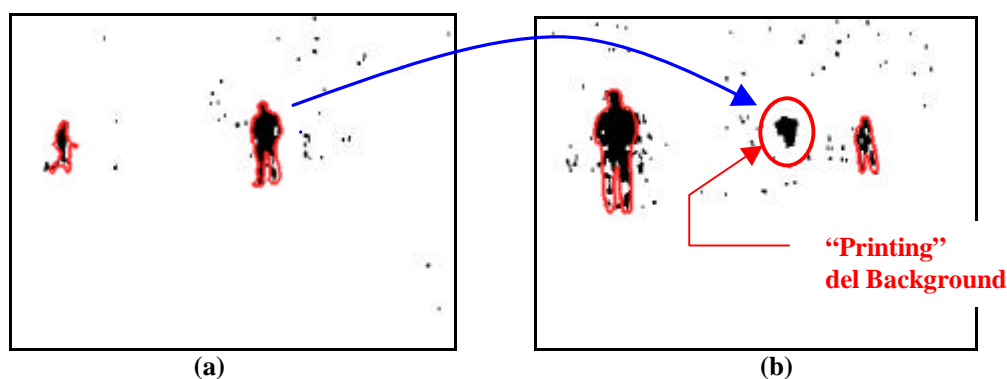


Figura 6-11 W4 - "Printing" della sagoma dell'oggetto nel background.

A fronte del cambiamento di illuminazione nei frame 310, 323 e 365 (Figura 6-7 e segg.) si nota come l'algoritmo W4 non riesca ad aggiornare correttamente il

riferimento: la percentuale di *falsi positivi* cresce dal 3,43% al 22,4% (Tabella 6-1, riga 2).

Double Difference mostra la sua immunità alle variazioni di illuminazione mantenendo le percentuali di *falsi positivi* molto basse (mediamente 0,8%). Ciò a causa del fatto che in questi frame (310, 323, 365) sono visibili i dettagli (texture) degli oggetti e grazie a questo scende il rumore *foreground* aperture. Così i *falsi negativi* medi scendono per Double Difference da 67% a 32% (Tabella 6-2, riga 1). Questo è dovuto all'avvicinarsi delle persone alla telecamera: i dettagli risultano riconoscibili ed aumenta la loro texture causando una riduzione di *foreground aperture* per gli algoritmi derivativi.

Osservando le immagini dei due algoritmi con immagine di riferimento si nota come sia Absolute Difference, che Block Based NCCF presentino output (Figura 6-7, Figura 6-8 e Figura 6-9(c)-(d)) di buona qualità. Analizzando i dati si vede che l'algoritmo basato sull'NCCF garantisce una minor quantità di *falsi negativi*, mediamente 22,8% contro 39,6% di Absolute Difference. Viceversa l'algoritmo Block Based NCCF presenta una maggiore quantità di *falsi positivi* (Tabella 6-1, riga 4, frame 310,323 e 365), dovuti all'aumento di luminosità che rende più evidenti i *waveing trees*, mettendo in evidenza la vulnerabilità di NCCF verso questi fenomeni. Rispetto a Block Based NCCF, l'Absolute Difference genera più *falsi positivi* a causa del fenomeno di "Printing" descritto in Figura 6-12.

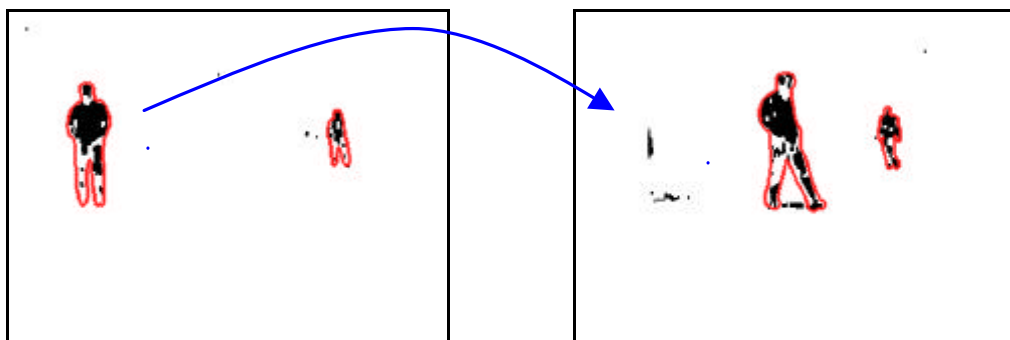


Figura 6-12 Absolute Difference - "Printing" della sagoma dell'oggetto nel background.

Analizzando globalmente le prestazioni degli algoritmi, si nota che W4 risulta in genere troppo sensibile al rumore e che il modello di background statistico male si adatta alle variazioni di illuminazione: dai dati sull'*errore assoluto* e

sull'*errore visivo* (Tabella 6-3 e Tabella 6-4, riga 1) si nota come le prestazioni (inizialmente buone) degradino rapidamente in corrispondenza della massima variazione di illuminazione.

I dati relativi alla Double Difference provano in maniera chiara la sua immunità a variazioni di illuminazione (Tabella 6-1, riga 1), ma anche come la qualità dell'output (*falsi negativi* - Tabella 6-2 riga 1) dipenda dalla *texture* dell'oggetto (a causa della *foreground aperture*).

Absolute Difference e Block Based Nccf hanno prestazioni molto simili (Tabella 6-3 e Tabella 6-4, righe 3 e 4). Per Absolute Difference riusciamo a ottenere un buon compromesso tra *falsi negativi* e *falsi positivi*, mentre vi sono problemi dovuti al mancato aggiornamento del background durante le variazioni di illuminazione. Block Based NCCF non risente di questo fenomeno, ma per contro è troppo sensibile *waveing trees* ed ha difficoltà nell'individuare oggetti privi di *texture* su sfondo uniforme.

Osservando i dati relativi a tutta la sequenza, l'algoritmo che ha prestazioni migliori risulta essere l'Absolute Difference con aggiornamento a due pesi.

6.3 Conclusioni

In questo capitolo sono riportati i risultati delle misure sperimentali effettuate per caratterizzare le prestazioni di alcuni algoritmi di basso livello per VSS. Sono inizialmente stati riprodotti alcuni tra i metodi di detection standard. Questo lavoro ha suggerito spunti per nuovi algoritmi anch'essi inseriti nel confronto.

Dai risultati ottenuti è emerso come algoritmi che sfruttano ipotesi particolari (in generale tutti gli algoritmi statistici) funzionino molto bene fintanto che le queste sono rispettate. Variando il contesto di applicazione di questi algoritmi sono emersi limiti derivanti dal fatto di dover fronteggiare fenomeni come i cambiamenti globali introdotti da variazioni di illuminazione, non contemplati nelle ipotesi di partenza.

Soluzioni che fanno uso di funzioni particolari come la NCC presentano pro e contro: la peculiarità di valutare la *texture* dell'immagine garantisce notevoli

robustezza alle variazioni di illuminazione e agli errori di aggiornamento del background; per contro vi è una eccessiva sensibilità ai fenomeni di *waveing trees*.

Esaminando il comportamento degli algoritmi in condizioni generiche ed in presenza di rumore di vario genere, l'algoritmo migliore risulta essere Absolute Difference con aggiornamento a due pesi. La sua semplicità deriva dalla assenza di ipotesi sul tipo di scena che risulta essere la scelta migliore.

Tuttavia dalle prove emerge che non esiste un algoritmo in grado di fornire una buona detection in qualsiasi condizione di ripresa. Questo ci permette di concludere che si può ottenere un sistema di videosorveglianza efficiente integrandovi più algoritmi di base: a seconda della condizioni della scena ripresa, un modulo di alto livello deciderà quale algoritmo utilizzare per ottenere le prestazioni ottimali. In questo caso la combinazione di due algoritmi con background, uno statistico e uno con immagine di riferimento, con uno basato sulle differenze (per es. la Double Difference) potrebbe rappresentare la scelta ideale.

