4 Studio del moto basato su Block Matching

4.1 Introduzione

La ricerca condotta ha per scopo lo sviluppo di un sistema di monitoraggio del traffico basato sulla visione che sia in grado di misurare i parametri fondamentali del traffico come la lunghezza delle code agli incroci e parametri di più alto livello semantico come il riconoscimento degli stati del traffico (congestione, blocco dovuto a incidente o rallentamenti a causa di condizioni meteo critiche...). In questo quadro un problema fondamentale consiste nello sviluppo di un algoritmo veloce ed affidabile per la individuazione e l'inseguimento (tracking). Si è scelto di concentrare gli sforzi sulla estrazione del moto dalle immagini per segmentare i veicoli rispetto agli oggetti fermi (strada ed edifici, ad esempio).
In particolare, l'approccio si basa sull'algoritmo di "Block Matching" (BMA) [MUS94], già utilizzato per la stima del moto anche nello standard di compressione MPEG [BAG97]. Il BMA è stato impiegato per la rilevazione dei veicoli in applicazioni aventi per obiettivo la stima delle percentuali di svolta agli incroci urbani [BAR96]. Diversamente dalle tecniche basate sulle derivate spazio-temporali [KOL94] [BAR98] o sulla sottrazione di un background [MIC91], il BMA non solo individua i veicoli, ma fornisce direttamente la stima quantitativa del loro moto (cioè associa vettori di moto ai punti dell'immagine). Ciò aumenta il potere discriminante in caso di occlusione parziale tra veicoli dovuta a situazioni di traffico intenso. Il fatto che il Block Matching sia computazionalmente costoso non costituisce un problema insormontabile, dato che al giorno d'oggi è disponibile hardware specializzato per la computazione

real-time del BMA [DEV89]. E' quindi possibile ipotizzare un sistema per il monitoraggio del traffico in cui la stima dei campi di moto può essere svolta da un hardware veloce di front-end, lasciando alla CPU il solo carico di analizzare le modeste quantità di dati così estratti.

4.2 L'algoritmo di Block Matching

Il Block Matching (BMA) agisce dapprima suddividendo ciascuna immagine della sequenza (frame) in blocchi quadrati di dimensione predefinita (normalmente da 6×6 a 8×8 pixel) e successivamente misurando lo spostamento dei blocchi tra il frame attuale (al tempo *t*) e quello precedente (al tempo *t-1*). Tale stima avviene cercando il blocco all'interno di un'area di scansione (scan area) all'interno della quale si presume possano essere contenuti i massimi spostamenti degli oggetti della scena tra due frame successivi. Ciascun blocco racchiude una parte dell'immagine ed è contenuto in una variabile matrice. La Figura 4-1 mostra la suddivisione del frame corrente ed il blocco la cui posizione di provenienza sarà ricercata nell'area di scansione del frame precedente.

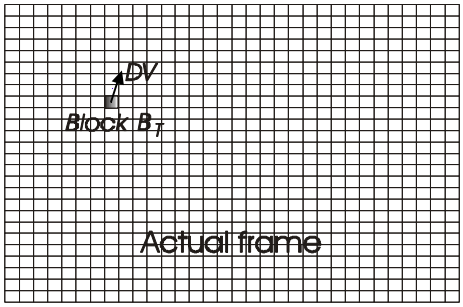

Figura 4-1 BMA: suddivisione del frame corrente e vettore di moto *DV*

Il vettore di spostamento (o di moto) *DV* originato nel blocco, rappresenta il risultato della ricerca. Durante la fase di ricerca, ad ogni blocco B_t risultante dal partizionamento del frame attuale è associata un'area di scansione nel frame precedente sulla quale è fatto scorrere il blocco B_t di pixel in pixel e confrontato ogni volta con il blocco B_{t-1} definito nell'area di scansione. Il matching ha lo scopo di individuare il blocco più simile a B_t tra tutti i B_{t-1}, come si nota in Figura 4-2

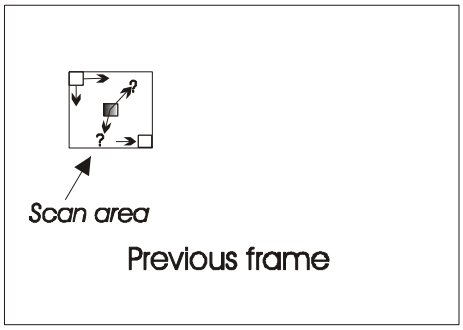

Figura 4-2 Processo di matching

La dimensione della scan area deve prevedere il massimo spostamento di un blocco tra due frame consecutivi, tuttavia, dato che il carico computazionale dipende ampiamente da questa dimensione è conveniente operare con aree di scansione di dimensioni il più possibile contenute.

La stima della somiglianza può essere valutata sulla base di vari criteri [MUS94]; come nel caso del VIL (capitolo precedente) si è scelto di utilizzare la funzione di Cross Correlazione normalizzata (NCCF):

$$NCCF = \frac{\sum_{i,j}[B_{t-1}(i,j) \cdot B_t(i,j)]}{\sqrt{\left[\sum_{i,j} B_{t-1}{}^2(i,j)\right] \cdot \left[\sum_{i,j} B_t{}^2(i,j)\right]}}$$

Equazione 4-1 Funzione di Cross Correlazione Normalizzata (NCCF)

Nell'Equazione 4-1 il simbolo "$\cdot$" rappresenta il prodotto tra elementi omologhi delle matrici B_t e B_{t-1} e gli indici i,j scorrono lungo l'intera area della matrice. Il processo di match identifica gli estremi di un vettore nelle posizioni di B_t e del miglior match di B_{t-1}. Tale vettore è considerato applicato nel centro di B_t e rappresenta la stima della posizione del blocco per l'immagine successiva al tempo $t+1$. L'insieme dei vettori di spostamento è chiamato campo di moto (DVF, displacement vector field) del frame.

Purtroppo, a causa delle fluttuazioni dei toni di grigio imputabili a variefonti di rumore, il BMA genera molti vettori errati nelle aree corrispondenti a blocchi statici che si trovano ad esempio nelle porzioni di asfalto con poca texture e non occupate da veicoli. Questo problema è tipicamente trattato favorendo la scelta del vettore nullo nella stima del miglior match durante la scansione dell'area. Ad

esempio in [ITU95], per ogni blocco, il valore costante del match calcolato nella medesima posizione del frame precedente è sottratto alle misure di similarità effettuate nel resto della sua scan area, in modo da favorire il risultato di "spostamento nullo" del blocco. Così, per quei blocchi che non hanno un match ragionevolmente buono all'interno della scan area, la stima dello spostamento tende a dare risultato nullo. Tuttavia, con questo approccio il processo di match continua ad essere eseguito sull'intera area nonostante vi sia spostamento nullo. Esiste una soluzione molto più conveniente che consente anche un notevole risparmio computazionale. Si può procedere calcolando prima la correlazione di un blocco nella stessa posizione del frame precedente (cioè si stima la correlazione di un eventuale spostamento nullo) e si confronta questa con una soglia fissa: se la correlazione non supera la soglia (match scarso) allora si sceglie il vettore di spostamento nullo, altrimenti si prosegue con la ricerca del massimo calcolando tutte le correlazioni della scan area. L'uso di una soglia relativamente alta equivale a favorire i vettori nulli per le aree particolarmente prive di texture e per questo potenzialmente più "rumorose".

Figura 4-3 Scena di traffico contenente meno del 5% di blocchi in movimento

I nostri risultati sperimentali mostrano che questo approccio è efficace nella prevenzione degli errori di match nelle aree statiche e garantisce al tempo stesso un notevole risparmio computazionale che nel caso del BMA è normalmente dell'ordine del 90%. Si consideri, ad esempio, un'area di scansione di 20×20

pixel ed un blocco di 8×8 pixel. Il matching del blocco sull'intera area richiede $(20)^2 = 400$ iterazioni della cross correlazione. Nell'immagine di 512×512 pixel mostrata in figura Figura 4-3 ci sono circa 180 vettori non nulli su 4096 blocchi totali. Confrontando il metodo standard con l'idea descritta si ottengono

```
4096*400 =          1638400- (BMA standard)
(180*400)+4096-180= 75916= (BMA con soglia)
--------------------------
                    1562484,
```

iterazioni di cross correlazione risparmiate per ogni frame. Così, con riferimento al caso di Figura 4-3, calcolando una sola correlazione per i blocchi in quiete si ottiene un risparmio computazionale del 95.6% [DI299].

4.3 Filtraggio mediano vettoriale

Sebbene la soglia per la cross correlazione del tipo descritto nel paragrafo precedente sia stata impostata su un valore molto selettivo, si ottengono di norma DVF rumorosi a causa delle inevitabili fluttuazioni dei toni di grigio tra i frame successivi e a causa degli spostamenti "fuori scan area" degli oggetti in rapido movimento. Nel nostro caso abbiamo realizzato una segmentazione dei veicoli che si basa sulla somiglianza dei vettori che gli appartengono. Per questa ragione i vettori spuri possono creare una notevole sovra-segmentazione come nel caso del veicolo racchiuso nel rettangolo della Figura 4-3 per il quale lo stadio di segmentazione ("labelling", descritto nel prossimo paragrafo) produce il risultato di Figura 4-1.

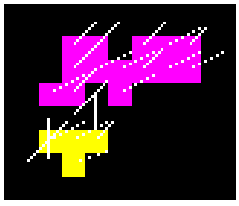

Figura 4-1 Eccessiva segmentazione di un veicolo dovuta a vettori spuri

Qui si vede chiaramente che, a causa di alcuni vettori spuri, il campo di moto associato al veicolo dà origine a due oggetti distinti che sono "etichettati" con due colori diversi.

Dato che il rumore riscontrato nel DVF può essere modellato con una statistica piuttosto regolare contenente solo alcuni valori fuori "moda", si è scelto di affidare la regolarizzazione del campo di moto ad un filtro mediano (vettoriale). Lo stesso approccio è stato seguito per la regolarizzazione della velocità [AST90], di campi di flusso ottico [BAR93] e dei campi di moto stessi [BAR96]. Il mediano di un insieme di valori *n* scalari è definito come il *(n/2)-esimo* elemento dell'insieme ordinato. Evidentemente questa definizione non può essere applicata direttamente al caso vettoriale. Tuttavia Astola et. Al. [AST90] hanno introdotto l'operatore mediano per grandezze vettoriali derivandolo da una funzione bi-dimensionale di densità di probabilità bi-esponeziale utilizzando il principio di massima somiglianza. Questo operatore vettoriale ha proprietà molto simili a quelle del mediano scalare per grandezze monodimensionali. L'operatore mediano vettoriale di Astola et Al. è quello che identifica l'elemento dell'insieme di vettori che minimizza la somma delle distanze da tutti gli altri elementi. Tali distanze sono calcolate sulla base della norma L_2.
Nello spazio $\Re^2$ la norma L_2 di un vettore $\boldsymbol{v}(x_v,y_v)$ è espressa dalla Equazione 4-1

$$\left\|\vec{v}\right\|_{L_2} = \sqrt{x_v^{\,2} + y_v^{\,2}}$$

Equazione 4-1 Norma L_2 di un vettore *v*

e la distanza tra due vettori $\boldsymbol{u}(x_u,y_u)$ e $\boldsymbol{v}(x_v,y_v)$ in base a L_2 si esprime con l'Equazione 4-2

$$\left\|\vec{u} - \vec{v}\right\|_{L_2} = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}$$

Equazione 4-2 Distanza tra 2 vettori

Nella nostra implementazione dell'operatore mediano vettoriale ciascun vettore di moto è rappresentato dal vettore mediano dell'insieme di vettori formato dal vettore stesso e dai suoi 8-vicini. La Figura 4-2 mostra un esempio di operatore mediano vettoriale applicato come filtro per regolarizzare un campo di moto.

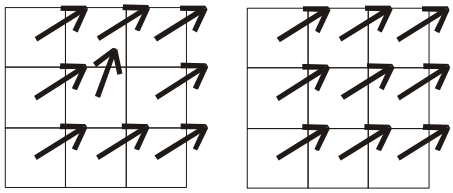

Figura 4-2 Filtro mediano vettoriale basato su intorno 3´3

La regolarizzazione attraverso il filtro mediano riduce significativamente le componenti di rumore e genera insiemi di blocchi con vettori vettori di moto più simili tra loro.

4.4 Lo stadio di raggruppamento dei blocchi

Dopo aver applicato il filtro mediano al DVF ottenendone la regolarizzazione i veicoli possono essere identificati. Ciò avviene accorpando aree di blocchi tra loro connessi ed aventi vettori simili (e non nulli). La Figura 4-1 mostra gli otto vicini 8-connessi di un dato blocco ***B***.

1	2	3
4	**B**	5
6	7	8

Figura 4-1 Blocco *B* e suoi vicini 8-connessi

Il processo di raggruppamento corrisponde fondamentalmente ad un algoritmo di labelling per componenti connesse, sequenziale e iterativo [HAR92], in cui sono presi in considerazione i blocchi in movimento. Due blocchi sono considerati "vicini" se sono 8-connessi e se la norma L_2 dei loro vettori resta al di sotto di una certa soglia *Th* come descritto nella Equazione 4-1

$$\left\|\vec{u}-\vec{v}\right\|_{L_2} = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2} < Th$$

Equazione 4-1 Condizione di "vicinanza" basata sulla norma L_2

Tale processo è conosciuto con il nome di *d*labelling. In aggiunta al filtraggio mediano dei vettori abbiamo scelto di eliminare i gruppi formati da un esiguo numero di blocchi, allo scopo di aumentare maggiormente la robustezza

dell'algoritmo nei confronti del rumore. Il valore della soglia sul numero di blocchi di un gruppo dipende dal rapporto tra le dimensioni in pixel dei blocchi e degli oggetti.

L'algoritmo di labelling è composto da un ciclo "`main`" ed una subroutine chiamata "`Mark-8`". Esso si esegue su ciascun frame e genera una matrice chiamata "`label_map`" avente le stesse dimensioni della matrice del DVF. Nello pseudo-codice riportato di seguito, il ciclo principale scandisce la `label_map` per trovare il primo blocco di cui non sia ancora stata determinata l'appartenenza (il ciclo termina quando tutti i blocchi sono stati etichettati con la rispettiva appartenenza). Per ciascun blocco non etichettato trovato dal ciclo `main` si istanzia una nuova etichetta chiamata seme (`seed`) incrementando il contatore di etichette `new_label` ed assegnandone il valore al blocco nella `label_map`. Quindi il ciclo principale effettua la chiamata a `mark-8` che annette al blocco-seme tutti i suoi vicini 8-connessi aventi vettore non nullo e soddisfacente l'Equazione 4-1. L'annessione è si ottiene in pratica assegnando a questi blocchi il valore di `new_label`.

PSEUDO CODICE

```
Per each frame of the sequence:
BEGIN MAIN
{
  FOR i,j scan the whole DVF's block
  matrix:
  {
    IF the block(i,j) is yet unlabelled AND
    it has a non null vector THEN
    REM (i,j) is a seed for the new label

      new_label = new_label + 1;

      label_map(i,j) = new_label;

      CALL "Mark-8"(i,j,new_label);

  }END FOR i,j scan

}END MAIN
*****************************************
PROCEDURE "Mark-8"(i,j,new_label)
BEGIN PROCEDURE
{
  WHILE scanning on (r,c), there are
  8-connected to (i,j) blocks
  DO BEGIN{
```

```
    IF (r,c) is an un-labelled block
    AND has a non-null associated vector
    AND L2_norm[vector(i,j)-vector(r,c)]
      LOWER THAN Threshold
    THEN
      label_map(r,c) = new_label;

  }END WHILE;
}END PROCEDURE
RETURN;
```

La nuova `label_map` così ottenuta dà origine ad un'altra matrice detta `resized_map`. Questa si ottiene scalando `label_map`, ingrandendola fino alla dimensione del frame originale di 512×512 pixel, così da avere una matrice con oggetti etichettati con le stessa risoluzione dell'immagine originale. La matrice `resized_map` sarà utilizzata nell'algoritmo di inseguimento descritto in seguito (Paragrafo 4-6).

4.5 Adattatività del filtro mediano

Abbiamo riscontrato che il filtraggio mediano vettoriale causa un fenomeno di "erosione" a danno delle componenti connesse presenti nel DVF. Quale conseguenza di questo fenomeno molti veicoli risultano frammentati, identificati da un esiguo numero di blocchi o addirittura persi, al termine del processo di raggruppamento. La Figura 4-1 mostra il DVF risultante dal processo di raggruppamento per il veicolo rappresentato in Figura 4-3 dopo l'applicazione del filtraggio mediano vettoriale.

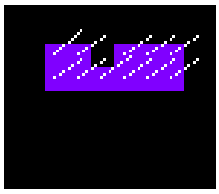

Figura 4-1 Eerosione di un veicolo causata dal filtro mediano

Se confrontiamo questo risultato con la Figura 4-1 risulta evidente che il filtro mediano causa una notevole erosione della forma del veicolo. Il problema è accentuato dal fatto che abbiamo a che fare con campi di moto contenenti un gran numero di vettori nulli ed uno relativamente basso di vettori non nulli (tra i quali sono presenti vettori spuri). Considerata poi la difficoltà del BMA di

identificare vettori di moto su aree senza texture e dato che queste aree sono piuttosto diffuse (oggetti a tinta uniforme come l'esempio mostrato in Figura 4-2)

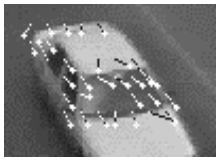

Figura 4-2 DVF carente su un oggetto con scarsa texture

è importante evitare di perdere vettori che potrebbero risultare determinanti per mantenere la connessione del DVF.

Se applichiamo semplicemente il filtraggio vettoriale la gran parte dei vettori sono trasformati in vettori nulli dal momento che, negli intorni dei vettori non nulli ai bordi delle forme, i vicini tendono ad essere dominati da quelli nulli. La Figura 4-3 mostra un esempio per quanto appena detto. Il vettore al centro dell'intorno sarà cancellato invece di essere trasformato in uno simile a un suo vicino non nullo, perché possiede più vicini "nulli" che vicini "non nulli".

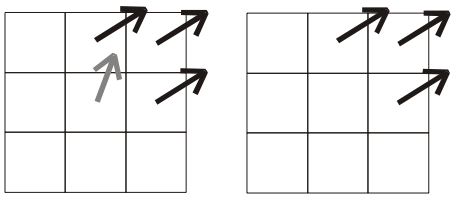

Figura 4-3 Azione del mediano su un DVF povero di vettori

Per questo motivo si è studiato un filtro adattativo in cui fossero presi in considerazione solo i vettori non nulli per il calcolo del mediano durante il processo di regolarizzazione. Ciascun vettore non nullo è trasformato nel vettore mediano sulla base dei soli vettori non nulli del suo intorno. Questo operatore può essere visto come un filtro adattativo statistico (o filtro "rank") [SHA89] in cui l'ordine cambia in funzione delle caratteristiche locali del segnale.

Figura 4-4 Veicolo della Figura 4-3 dopo l'azione del filtro adattativo

La Figura 4-4 mostra il veicolo della Figura 4-3 dopo il filtraggio adattativo vettoriale ed il raggruppamento dei blocchi. Il DVF è stato regolarizzato senza

causare l'erosione del veicolo che ora è rilevato correttamente come un singolo oggetto in movimento. Come detto nella introduzione il BMA è efficace nella identificazione di oggetti che si occludono parzialmente: nella Figura 4-5 è mostrato l'esempio di due veicoli le cui etichette sono a contatto: il semplice algoritmo di labelling non sarebbe in grado di distinguere i due oggetti. In questo caso la corretta rivelazione è invece garantita dalla stima del DVF dei due veicoli che evidentemente procedono a velocità diverse (si noti che i vettori del veicolo in alto sono leggermente più lunghi di quelli del veicolo in basso).

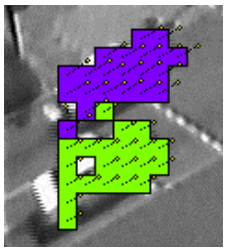

Figura 4-5 Due veicoli con etichette in contatto segmentati correttamente

4.6 Tracking

L'algoritmo di inseguimento (tracking) si basa sulla seguente idea: il BMA calcola gli spostamenti dei blocchi tra due frame; dato che i blocchi corrispondono a dettagli dell'immagine noi consideriamo ciascun vettore non nullo di una immagine come la informazione di tracking associata al blocco. In pratica, il BMA è sufficiente di per sé a inseguire ciascuna porzione dell'immagine contenuta in un blocco. L'algoritmo di labelling fornisce le informazioni necessarie ad inseguire gli oggetti (che si possono considerare come sottoparti dell'immagine costituite da blocchi aventi caratteristiche di moto tra loro simili). Più precisamente lo stadio di raggruppamento fornisce le informazioni circa l'appartenenza di ciascun blocco ad un oggetto. Integrando le informazioni provenienti dal BMA e dal labelling saremo in grado di inseguire interi oggetti [DIS99]. In altri approcci di tracking si cerca di prevedere la posizione del veicolo nel frame a seguire per poi assegnargli un collegamento all'oggetto corrispondente del frame attuale sulla base della vicinanza dei

baricentri [BAR96] o della somiglianza della texture degli oggetti [BAR98]. Il nostro approccio, al contrario guarda alla posizione dalla quale proviene l'oggetto poiché grazie all'inseguimento basato sulla consistenza multipla dei blocchi abbiamo una ragionevole speranza di trovarlo in un'area ben delimitata dell'immagine precedente. Dato che la posizione, lo spostamento, la velocità e le dimensioni del veicolo fornite dal BMA sono le uniche caratteristiche stimate per il nostro algoritmo di tracking, possiamo considerare quest'ultimo come completamente basato sul BMA.

L'algoritmo di tracking si basa sui dati forniti dagli algoritmi di block matching e di labelling. I parametri calcolati dal labelling consistono infatti in un insieme di etichette "temporanee" che saranno aggiornate dallo stadio di tracking in accordo con le informazioni di tracking già note a livello di blocco fornite da BMA e contenute nel DVF.

Dato l'insieme dei blocchi appartenenti ad un oggetto etichettato nel frame corrente si copia ciascun blocco sulla `resized_map` del frame precedente (vedere Paragrafo 4.4) traslato di una quantità pari all'opposto del corrispondente vettore del DVF (***V'*** **= -*V***) valutando la sovrapposizione della etichetta temporanea assegnata all'oggetto su quelle degli oggetti del frame precedente.

Il processo è descritto in Figura 4-1 la cui parte destra mostra l'etichettatura temporanea di un oggetto nel frame corrente e quella sinistra le etichette "stabilite" per gli oggetti fino al frame precedente.

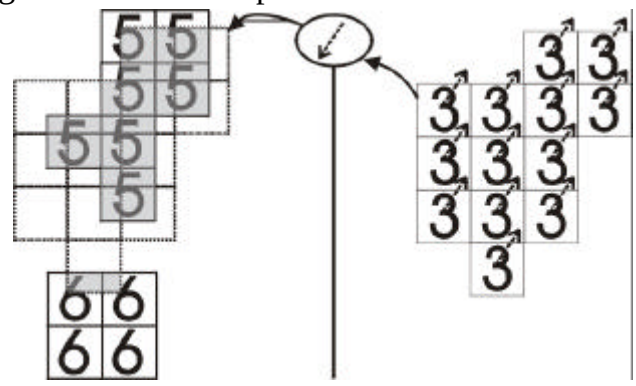

Figura 4-1 Inseguimento di un oggetto basato sulle componenti dei suoi blocchi

Per inseguire l'oggetto temporaneamente etichettato con "3" se ne copiano i blocchi sul frame precedente traslati dell'opposto del suo DVF. Ogni blocco

traslato si sovrapporrà ad *NxN* pixel della `resized_map` precedente. Questi pixel rappresentano i valori delle etichette per i blocchi ricoperti che saranno utilizzati per stabilire il valore finale dell'etichetta per l'oggetto considerato. Come mostrato in Figura 4-1 i blocchi possono sovrapporsi a pixel etichettati (tratteggiati in grigio) ed a pixel senza alcuna etichetta (bianchi). Scorrendo l'intera area occupata dall'oggetto traslato sul frame precedente si contano le quantità di pixel per ogni diverso valore di etichetta ricoperto. Si ottiene così un vettore di valori dove ogni cella contiene il numero dei pixel conteggiati per il valore dell'etichetta espresso dall'indice della cella stessa. L'indice della cella che contiene il numero più alto di pixel conteggiati sarà la nuova etichetta per tutti i blocchi dell'oggetto temporaneamente etichettato con "3". Con riferimento alla parte sinistra della Figura 4-1 è evidente che il massimo valore conteggiato si troverà in corrispondenza dell'indice "5" del vettore menzionato. Conseguentemente la Figura 4-2 mostra il risultato dell'inseguimento dell'oggetto temporaneamente etichettato con "3", al quale viene definitivamente assegnata l'etichetta "5".

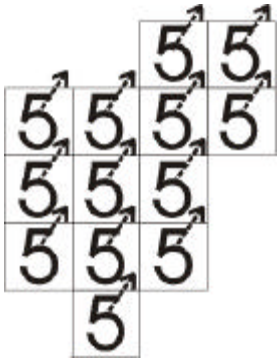

Figura 4-2 Risultato del tracking per l'oggetto di Figura 4-1

Il processo è iterato per ciascuna etichetta temporanea del frame corrente per inseguire tutti gli oggetti del frame. La Figura 4-3 mostra il tracking di due veicoli tra due frame contigui: tempo = *t* a sinistra e tempo = *t+1* a destra. I veicoli a destra mantengono la stessa etichetta del frame precedente (a sinistra).

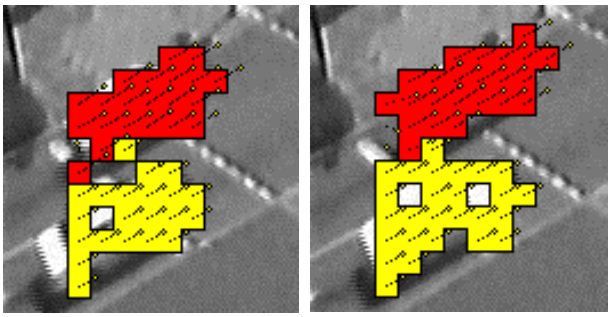

Figura 4-3 Tracking in due immagini consecutive di una sequenza di traffico

Se per una data etichetta temporanea la sovrapposizione non ricopre alcun pixel etichettato del frame precedente allora sarà ad essa assegnato un nuovo valore (si tratta di un “nuovo nato”).

Il moto degli oggetti assieme alla estrazione a bassa risoluzione della forma degli oggetti, al rumore ed alle conseguenti operazioni di filtraggio possono causare variazioni notevoli alla forma ed alla dimensione dello stesso oggetto frame dopo frame. Per queste ragioni può succedere che il massimo numero di pixel ricoperti sia dato da quelli non etichettati. Il problema cresce specialmente nel caso in cui gli oggetti siano costituiti da un esiguo numero di blocchi. Se ci limitassimo ad assegnare all’oggetto l’etichetta col maggior numero di pixel ricoperti potremmo etichettare erroneamente l’oggetto come “nuovo nato”. E’ invece opportuno scartare un eventuale massimo sul valore “0” se è stato ricoperto almeno un pixel con un altro valore di etichetta.

4.7 Conclusioni

Abbiamo condotto gli esperimenti utilizzando le sequenze del sito http:// i21www.ira.uka.de/image_sequences/ e sequenze provenienti da alcune telecamere installate lungo la rete stradale urbana di Bologna e gentilmente concesse dalla centrale operativa comunale per il controllo del traffico. I risultati [DIS99] mostrano che la procedura sviluppata (cioè l’unione del BMA con soglia di correlazione con il filtro adattativo e con l’algoritmo di raggruppamento) è efficace nell’identificazione dei veicoli. Tuttavia, quando le immagini sono affette da distorsione prospettica e quando gli oggetti sono troppo grandi rispetto

alle dimensioni dei blocchi, permangono effetti di sovra-segmentazione, come nel caso mostrato in Figura 4-1.

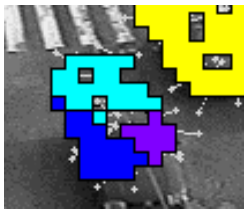

Figura 4-1 Veicolo frammentato in tre oggetti

Per trattare questo problema si può pensare di applicare una correzione della distorsione prospettica come menzionato nel capitolo precedente.

Per quanto concerne il tracking i risultati sperimentali mostrano che la procedura sviluppata (BMA con soglia, filtro adattativo, labelling e tracking) è efficace nell'inseguimento dei veicolo. Il problema maggiore del BMA sembra essere il carico computazionale. Per ovviare questo inconveniente si potrebbe sviluppare un nuovo algoritmo di BMA in cui avvenga una "pre-selezione" dei blocchi di interesse sulla base di differenze tra frame. Per il trattamento di fenomeni complessi del tracking (ad es. occlusione reciproca totale o ad opera di elementi del background) si prevede l'implementazione di un modulo logico di alto livello.