

Università degli Studi di Salerno



FACOLTÀ DI INGEGNERIA

Tesi di laurea
in
INGEGNERIA INFORMATICA

Un metodo per il tracking di persone in
ambienti affollati basato sulla predizione del
moto e l'apparenza visuale

Relatore:

Ch.mo Prof. Mario Vento

Candidato:

Rosario Di Lascio

Matr. 062/0700004

ANNO ACCADEMICO 2009 – 2010

Abstract

Nel corso dell'ultimo decennio, l'importanza sempre maggiore attribuita al tema della sicurezza in ambienti pubblici è stata d'impulso per un incremento dello sviluppo delle applicazioni di videosorveglianza intelligente. Le modalità e gli ambiti di utilizzo sono tra i più svariati, ad esempio nelle stazioni e negli aeroporti si controllano le persone per poter riconoscere eventuali comportamenti scorretti o anomali. L'obiettivo dei sistemi di videosorveglianza è generalmente il seguente: attraverso operazioni di image processing, vengono riconosciuti gli oggetti in movimento (object detection); vengono quindi calcolate le traiettorie di questi oggetti per tutto il tempo in cui questi sono presenti sulla scena (object tracking) ed infine, attraverso lo studio delle traiettorie e quindi dell'interazione tra gli oggetti presenti in un dato momento, è possibile modellare tutti i possibili eventi che si verificano nel filmato (behavioral analysis). Il processo di riconoscimento di un'immagine avviene su tre distinti livelli: Pixel Processing, Frame Processing e Tracking. Tra i vari livelli che compongono una generica applicazione di videosorveglianza, la nostra attenzione si è soffermata sul modulo di tracking. Infatti, l'obiettivo di questo lavoro di tesi è la progettazione e la realizzazione di un modulo di tracking, ponendo particolare importanza alla risoluzione delle problematiche dovute ad una stima errata del vettore di moto, alla classificazione robusta di un oggetto presente all'interno della scena ed all'evoluzione delle metriche di somiglianza per l'associazione di un box rilevato in fase di detection con un oggetto già presente nella sequenza; tali scelte sono avvalorate valutando sperimentalmente il sistema attraverso metriche standard del settore.

Il primo obiettivo affrontato è stato quello di cercare delle metriche che potessero valutare oggettivamente le prestazioni dell'attuale algoritmo di tracking noto in letteratura come "Matrix Tracking". Una volta analizzato il sistema "Matrix Tracking" e comprese appieno le sue problematiche, ci siamo apprestati a realizzare una nuova architettura dell'intero modulo. Questa fase è stata soggetta a vincoli stringenti, imposti dal contesto reale, nonché dal più complesso sistema di cui il modulo di tracking entrerà a far parte. Le esigenze di fronte alle quali ci siamo trovati sono tipiche di tutte le applicazioni di videosorveglianza intelligente; in primo luogo, infatti, l'applicazione deve funzionare in real-time. Per questo motivo è stato necessario trovare un giusto compromesso tra robustezza ed onere computazionale del sistema. Ulteriore specifica riguarda l'utilizzo di un paradigma di programmazione sequenziale in sostituzione dell'attuale paradigma orientato agli oggetti; ciò ha determinato la scelta di diagrammi IDEF0 per avviare la fase di progettazione. Tale scelta è stata avvalorata dall'approccio di tipo top-down utilizzato, il che ha reso possibile svincolarci dalle reali implementazioni delle singole funzioni. Inoltre, l'utilizzo di un approccio sequenziale ha reso indispensabile una reingegnerizzazione di tutte le strutture dati. Terminata la fase di progettazione dell'intero sistema di tracking, ci siamo soffermati su un sottoinsieme dei moduli che lo compongono; i moduli implementati sono i seguenti: Filtro di Kalman, Classificatore per gli oggetti rilevati, Matrice di somiglianza basata su tre categorie di misure (posizione, forma e apparenza visiva). Tali moduli sono stati scritti utilizzando il linguaggio C++ e le librerie grafiche OpenCV.

Non avendo a disposizione l'intera piattaforma, abbiamo realizzato un simulatore per testare il nostro modulo. Questo è in grado di simulare l'intero processo di tracking, estraendo le informazioni dalla fase di detection precedentemente effettuata in maniera sincrona rispetto al frame-rate impostato e producendo un output grafico per avere un immediato riscontro, oltre al reale output richiesto dal sistema attuale. Tramite il simulatore è stato possibile verificare il funzionamento in modalità real-time dell'applicazione. Per ogni singolo modulo è stata eseguita una fase di sperimentazione e di testing; infine, per validare il nostro sistema, sono state eseguite varie fasi di verifiche che permettessero un confronto con la precedente versione eseguendo i medesimi test e confrontando i risultati delle metriche ottenuti.

Per quanto riguarda la fase di sperimentazione delle singole unità, i risultati ottenuti ci hanno portato a configurare, indipendentemente l'uno dall'altro, ogni singolo modulo. In merito al filtro di Kalman, ne sono stati configurati i parametri utilizzando un vettore di stato ad otto dimensioni. Come metrica basata sulla posizione è stata utilizzata la distanza tra il centro di un oggetto presente all'istante precedente, di cui ne sono state predette dimensioni e posizioni utilizzando il filtro di Kalman, ed il centro dei box rilevati dalla fase di detection. La metrica basata sulla forma, invece, è stata calcolata considerando come parametri l'altezza media reale e l'area del blob. Infine, per quanto riguarda la metrica basata sull'apparenza visiva, è stata calcolata la distanza Chi-Quadro tra gli istogrammi di colore, dopo aver suddiviso l'oggetto in esame in due regioni utilizzando la tecnica dell'Image Partitioning (una superiore per testa e busto ed una inferiore per le gambe) ed averne estratto le tre componenti di colore (r, g e b). Per il modulo di classificazione è stato realizzato un classificatore non parametrico Nearest-Neighbor con tre classi (persona, gruppo, bagaglio).

Ultimo passo è stato il testing dell'intero sistema e la sua valutazione prestazionale. Le metriche utilizzate sono sei e definite in letteratura dagli acronimi VACE e CLEAR, ormai divenute standard de facto per questo tipo di applicazioni. Queste ci permettono di confermare la bontà del nostro metodo, grazie ad un miglioramento medio delle prestazioni che si assesta sul 20%.