Localizzazione e tracking tramite “sensor fusion”

Marco De Rocchi
matr. 566841
marco.derocchi@gmail.com

Andrea Manente
matr. 566877
andrea.mane@alice.it

Alessandro Mucciardi
matr. 566547
antoina@libero.it

Giorgio Paccagnella
matr. 566759
giorgiopaccagnella@libero.it

***Sommario*—In questo lavoro viene presentato un testbed virtuale per la verifica di algoritmi di tracking e sensor fusion. Sulla base di specifiche reali è stato implementato un sistema virtuale di telecamere pan-tilt-zoom (PTZ) che insieme a una rete di sensori in radiofrequenza costituisce un ottima soluzione per testare soluzioni efficienti e robuste per la localizzazione e l’inseguimento di un oggetto in movimento. In particolare, in questo lavoro sono stati presi in considerazione aspetti strettamente legati al tracking, come per esempio l’inseguimento di un target in fase di manovra, e problematiche di data fusion e coordinazione tra videocamere che sono senza dubbio una delle sfide maggiori per il futuro nel campo della videosorveglianza.**

***Index Terms*—Sensor fusion, cooperative tracking, PTZ camera networks, videosurveillance systems.**

I. Introduzione

Negli ultimi anni è cresciuta la domanda relativa alla videosorveglianza sia in ambito civile che militare. Non si tratta semplicemente di telecamere fisse il cui unico scopo è quello di inquadrare una zona di interesse, ma di sistemi spesso molto complessi ideati per assolvere i più svariati compiti. L’idea principale che sta alla base di questo tipo di sistemi è la possibilità di poter monitorare a distanza oggetti o persone in movimento in un’area che spesso ha dimensioni molto ampie, riducendo di fatto il controllo diretto da parte degli addetti alla sorveglianza. La possibilità, infatti, di poter localizzare ed eventualmente inseguire un bersaglio (in inglese *target*) all’interno dell’area monitorata, si traduce in un risparmio diretto di tempo e dispiegamento di forze (come p.es. nel caso delle telecamere impiegate nelle zone a traffico limitato ZTL). In quest’ottica si inserisce un vasto campo di ricerca, caratterizzato da tendenze fortemente multidisciplinari, che si pone come obiettivo la messa a punto di soluzioni sempre più efficaci e robuste ai problemi di videosorveglianza.

Tra tutte le problematiche che contraddistinguono questi sistemi complessi, una delle più importanti è sicuramente la gestione intelligente dell’informazione proveniente da tutti i sensori che compongono il sistema: questa pratica va sotto il nome di *information fusion* (o *sensor fusion* a seconda del livello di applicazione) ed è trattata approfonditamente in [12]. Il sensor fusion è un processo che combina due o più rilevazioni provenienti da diversi sensori per migliorare la descrizione di un certo fenomeno. Per assicurare, infatti, che la stima del sistema sia particolarmente accurata, è possibile e preferibile utilizzare in maniera consona tutte le informazioni disponibili. La fusione delle misure da più sensori porta numerosi vantaggi relativi alla precisione delle stime, in particolare:

- si riduce l’incertezza delle singole misurazioni;
- aumenta la reiezione del rumore;
- possono essere tollerati malfunzionamenti (*failure*) dei sensori, come per esempio il mancato arrivo di dati;
- può essere estesa la copertura della rete.

I sensori da cui provengono i dati possono avere le stesse caratteristiche e stesse modalità di acquisizione (si parla allora di *sensori competitivi*), oppure possono essere eterogenei e prensentare diverse specifiche (*sensori complementari*). Un’interessante applicazione in questo senso è la localizzazione di un target per mezzo di sensori dello stesso tipo ma che si comportano diversamente (per esempio alcuni sono fissi mentre altri sono in movimento): una soluzione a questo problema è presente nell’articolo [9] e passa attraverso l’implementazione di un filtro di Kalman esteso per ciascun sensore. Successivamente viene operata una fusione dei dati per ottenere una stima globale della posizione del target migliore rispetto alle stime dei singoli sensori prese separatamente. Tale problema è di forte interesse in ambito militare, e viene spesso associato all’utilizzo di radar.

In ambito civile, invece, e in particolare nello studio della movimentazione automatica di un veicolo, come viene spiegato in [13], si tratta la fusione di dati derivanti da sensori di tipo eterogeneo e non sincronizzati, che portano informazioni relative a diverse variabili. In questo caso si ha ancora l’utilizzo di filtri di Kalman distribuiti e una successiva fusione dei dati che potrebbe essere anche una semplice combinazione lineare di essi. Invece, in [10], si prende in considerazione un sistema distribuito di telecamere fisse e una rete wifi per la localizzazione; in tale articolo si fondono insieme le posizioni ottenute attraverso la rete wifi (*wifi location estimation*) e la rete di telecamere (*particle filter estimation*).

Per quanto riguarda invece il tracking, in particolare di veicoli aerei e navali, in [14] viene esposto un modo alternativo al filtro di Kalman basato su una rete Bayesiana. Si tratta di un approccio che prende in considerazione molteplici aspetti tra cui la selezione di modelli Bayesiani, sistemi ad eventi discreti, modelli semi-markoviani e stima progressiva dello stato. Un approccio simile è stato trattato in [15], sempre legato alle reti Bayesiane e ai filtri particellari (*particle filter*).

Rimanendo in ambito civile, ma spostandoci verso la sicurezza di aree protette o impianti, ci si imbatte in un problema che ha avuto notevole sviluppo negli ultimi anni: la videosorveglianza. Questo è l’argomento trattato in [16] e [17], dove inizialmente si pone il problema dell’identificazione del target che dovrà essere inseguito, e successivamente viene affrontato il problema relativo all’*handover* (o *handoff*) tra le telecamere per un corretto inseguimento. In questi due casi la conoscenza della zona di lavoro è data a priori (si parla comunemente di *ambiente strutturato*), pertanto il meccanismo di handover si basa su questa conoscenza. Il problema dell’identificazione del target si basa su algoritmi di *features extraction* tra cui il riconoscimento di colori nella scansione dell’immagine.

Il nostro lavoro cerca di mettere insieme gli aspetti principali appena esposti, prescindendo tuttavia dalla specificità dell’applicazione. Abbiamo sviluppato un sistema di localizzazione e tracking di un agente mobile all’interno di una rete di sensori eterogenei quali videocamere e sensori in radiofrequenza. A partire dalla strumentazione presente in laboratorio, composta da un piano di lavoro su cui si muove un robot, da una rete di sensori RF e da una videocamera fissa che inquadra frontalmente il piano, è stato realizzato un testbed virtuale per la verifica degli algoritmi di *tracking* e *sensor fusion*. Il sistema consiste di N telecamere PTZ virtuali poste sopra al piano in grado di inquadrare e inseguire il veicolo, oltre che ovviamente della rete di sensori wireless per la localizzazione.

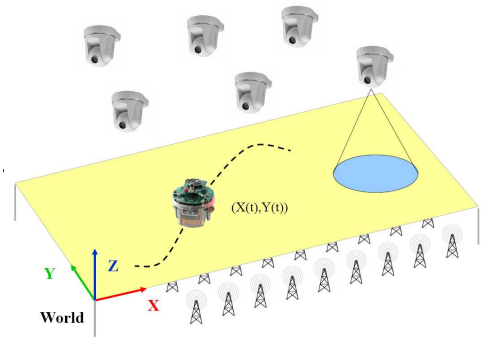

Figura 1. Sistema virtuale creato in simulazione

La rete di sensori entra in gioco nell'inizializzazione del tracking e viene utilizzata solo in assenza di misure provenienti dalle telecamere. Questo perchè l'incertezza delle misure provenienti dai sensori è troppo grande rispetto a quella dei dati delle telecamere e perciò una volta localizzato e agganciato il veicolo, da una qualsiasi telecamera, sarà sufficiente basarsi sulle immagini delle videocamere per effettuare un buon inseguimento. Esse possono così inseguire il veicolo quando questo si trova nel corrispettiva zona di azione e possono predisporsi ad agganciarlo quando si trova in prossimità della stessa. Per la sola rete di telecamere, infine, abbiamo sviluppato alcuni algoritmi di *sensor fusion* per migliorare l'accuratezza delle stime.

L'articolo si svilupperà in questo modo: nella sezione II verrà spiegato come è stato possibile "virtualizzare" il sistema reale, e cioè come partendo da una singola immagine sono state ricavate le immagini relative alle singole telecamere virtuali. Successivamente in III verranno esposti gli algoritmi di *tracking* utilizzati per garantire l'inseguimento del target. Nella sezione IV si passerà alla trattazione dei metodi di *sensor fusion* per poi concludere in V con alcune questioni realizzative e con l'esposizione dei risultati.

II. Layout del sistema

Il sistema preso in considerazione per i nostri esperimenti è costituito da un piano di lavoro, caratterizzato da una terna di riferimento che indicheremo come sistema mondo, da una rete di sensori in radiofrequenza che garantiscono un servizio di GPS virtuale sul target e da una serie di telecamere PTZ (Pan-Tilt-Zoom) poste sopra il piano.

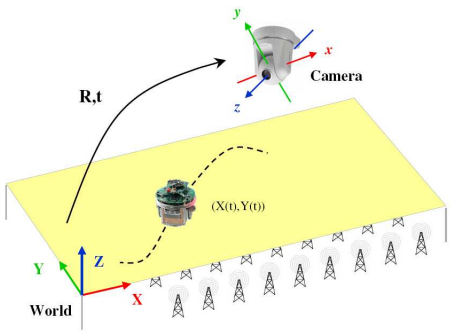

Figura 2. Layout del sistema

A differenza della realtà, dove il target può occupare un volume all'interno dello spazio tridimensionale in cui agiscono le telecamere, nel nostro caso si è operata una semplificazione drastica, vincolando il movimento al solo piano di lavoro e considerando come oggetto da inseguire una figura bidimensionale (più precisamente un triangolo) immersa in questo piano. Un'ulteriore semplificazione adottata è l'utilizzo di immagini in bianco e nero[1] al posto di immagini a colori o in scala di grigi: questo sia per semplificare le operazioni di *features extraction* che per velocizzare le operazioni di calcolo. Ovviamente queste semplificazioni potrebbero sembrare alquanto restrittive, soprattutto in virtù del fatto che nelle applicazioni reali non si può prescindere da quanto detto sopra. Tuttavia, dal momento che il nostro sistema virtuale, pur riproducendo fedelmente il funzionamento di un sistema reale, *non* è strutturato per la sperimentazione di algoritmi *application oriented* quanto piuttosto per testare strategie più generali, tutte le questioni legate alle semplificazioni messe in atto assumono un'importanza relativa.

Veniamo ora alla descrizione di come l'apparato sperimentale reale, illustrato in figura 2, è stato tradotto in un sistema virtuale che ne riproduce l'esatto funzionamento.

A. Virtualizzazione

L'idea della virtualizzazione del sistema nasce dalla necessità di avere a disposizione le immagini provenienti dalle singole telecamere che compongono la rete. Nel caso di un sistema reale, come per esempio un sistema di videosorveglianza, le viste relative a ogni telecamera giungono dalle telecamere stesse e sono pertanto utilizzabili direttamente. Se però non si hanno a disposizione queste immagini reali, è possibile ricostruirle in modo approssimato a partire da un'unica vista, attraverso quella che viene definita *rettificazione ortogonale*.

Entriamo più nel dettaglio: supponiamo di piazzare una telecamera fissa sopra al piano di lavoro, con l'asse ottico perfettamente ortogonale a esso. L'immagine che ne ricaviamo, trascurando gli effetti di distorsione prospettica e radiale, è l'esatta rappresentazione del piano a meno di un fattore di scala. Supponiamo inoltre di avere un'altra telecamera, questa volta PTZ, posta anch'essa sopra al piano, con una certa orientazione data dagli angoli di pan e tilt. Essa vedrà una porzione del piano di lavoro che corrisponde alla proiezione prospettica del suo CCD attraverso il centro ottico.

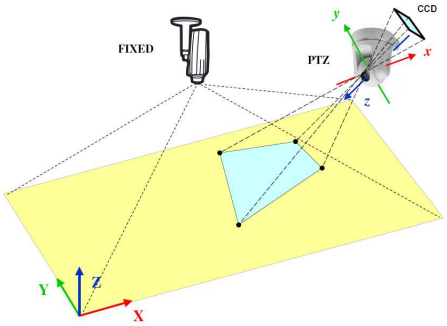

Figura 3. Layout del sistema per la virtualizzazione e corrispondenze tra punti

[1]Attenzione: le immagini in bianco e nero non sono immagini in scala di grigi, ma matrici di soli 0 e 1.

Le corrispondenze tra i vertici del CCD e le loro proiezioni sul piano di lavoro codificano una trasformazione lineare non singolare del piano proiettivo in sè stesso detta *omografia* (o *collineazione*). Nel nostro caso l'omografia è rappresentata da una matrice $\mathbf{H}$ non singolare 3×3 che lega le coordinate dei punti nel piano CCD con le coordinate delle loro proiezioni prospettiche sul piano di lavoro. In formule:

$$\lambda \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix} = \begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} \\ H_{2,1} & H_{2,2} & H_{2,3} \\ H_{3,1} & H_{3,2} & H_{3,3} \end{bmatrix} \begin{bmatrix} x^{ccd} \\ y^{ccd} \\ 1 \end{bmatrix} \quad (1)$$

I punti sono espressi in coordinate omogenee, cioè si denotano i punti 2D sul piano immagine (CCD) come $(x^{ccd}, y^{ccd}, z^{ccd})$ dove $(x^{ccd}/z^{ccd}, y^{ccd}/z^{ccd})$ sono le coordinate cartesiane corrispondenti[2]. La matrice $\mathbf{H}$ è definita a meno di una fattore di scala e ha 8 gradi di libertà: pertanto è necessario trovare almeno quattro corrispondenze di punti (purchè tre di questi non siano collineari) per definire univocamente la matrice di collineazione. Per trattare il fattore di scala incognito si possono utilizzare due metodi: fissare a piacimento uno dei valori della matrice (p. es. $H_{3,3} = 1$) oppure risolvere il sistema omogeneo $\mathbf{Ah} = \mathbf{0}$ sfruttando la SVD di $\mathbf{A}$.[3]

Torniamo ora al problema originale, cioè ottenere le immagini che vedrebbero le singole telecamere PTZ partendo da un'unica vista, per esempio quella di una telecamera fissa posizionata sopra al piano di lavoro. La rettificazione ortogonale serve a "raddrizzare" un'immagine prospettica di un piano preso di scorcio (v. fig 4). Questa rettificazione si basa sul fatto che la trasformazione tra il piano della scena e la sua immagine prospettica è un'omografia. Il modo più semplice per vederlo è scegliere il sistema di riferimento mondo in modo che il piano abbia equazione $z = 0$, come nel nostro caso. Allora la matrice di proiezione prospettica (MPP) $\mathbf{P}$ della telecamera PTZ che sta inquadrando la scena di scorcio si riduce a una matrice 3×3 invertibile e dunque rappresenta una trasformazione proiettiva del piano (omografia). L'equazione di proiezione prospettica, che lega le coordinate immagine a quelle cartesiane nel sdr mondo, risulta:

$$k \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} P_{1,1}\ P_{1,2}\ P_{1,3}\ P_{1,4} \\ P_{2,1}\ P_{2,2}\ P_{2,3}\ P_{2,4} \\ P_{3,1}\ P_{3,2}\ P_{3,3}\ P_{3,4} \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} P_{1,1}\ P_{1,2}\ P_{1,4} \\ P_{2,1}\ P_{2,2}\ P_{2,4} \\ P_{3,1}\ P_{3,2}\ P_{3,4} \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix} \quad (2)$$

dove (u, v) sono le coordinate immagine della telecamera PTZ e sono in corrispondenza biunivoca con le coordinate cartesiane della matrice CCD (x^{ccd}, y^{ccd}) a meno di un fattore moltiplicativo[4]. Pertanto la MPP della telecamera si riduce a una matrice 3×3 invertibile che rappresenta l'omografia tra il piano immagine e quello di lavoro.

Dunque, la trasformazione tra un piano della scena (il nostro piano di lavoro) e una sua immagine prospettica (quella vista dalla telecamera PTZ) è un'omografia, completamente definita da quattro punti dei quali si conosca la posizione nel piano della scena; nel nostro caso, come già accennato in precedenza, possiamo prendere le proiezioni dei vertici del CCD sul piano di lavoro. Una volta determinata tale omografia, l'immagine può essere proiettata all'indietro nel piano della scena. Questo è equivalente a sintetizzare un'immagine da una vista fronto-parallela del piano. Tale metodo è conosciuto col nome di *rettificazione ortogonale* [Liebowitz e Zisserman (1998)] di un'immagine prospettica.

Tuttavia noi siamo interessati all'operazione inversa rispetto alla rettificazione, cioè ricostruire la vista prospettica a partire da quella fronto-parallela derivante da una telecamera posta sopra al piano. Dal momento che l'omografia è rappresentata da una matrice 3×3 non singolare e quindi invertibile, possiamo usare la sua inversa per proiettare l'immagine fronto-parallela sul piano immagine della telecamera PTZ. In questo modo, da un'unica immagine del piano di lavoro, possiamo ricostruire virtualmente ciò che vedrebbe una telecamera PTZ i cui parametri geometrici sono rappresentati dalla matrice $\mathbf{H}$.

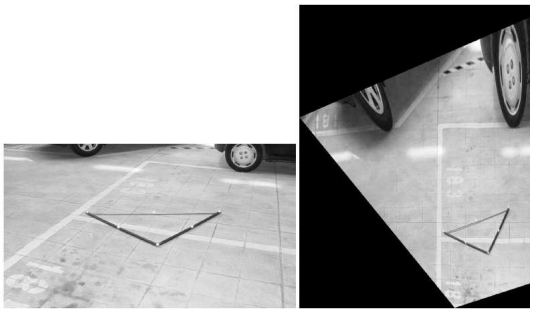

Figura 4. Esempio di immagine prospettica e immagine orto-rettificata del piano del pavimento. Da notare come risultano distorti gli oggetti non appartenenti al piano (le ruote delle automobili ad esempio). Immagine presente in [2] pag. 117.

In formule tutto ciò equivale a invertire la (1), in modo da ottenere le coordinate immagine (CCD) a partire da quelle espresse nel sdr mondo.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} x^{ccd} \\ y^{ccd} \\ 1 \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} \\ H_{2,1} & H_{2,2} & H_{2,3} \\ H_{3,1} & H_{3,2} & H_{3,3} \end{bmatrix}^{-1} \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix} \quad (3)$$

Sostituendo (2) in (3) risulta:

$$\frac{1}{k} \begin{bmatrix} P_{1,1}\ P_{1,2}\ P_{1,4} \\ P_{2,1}\ P_{2,2}\ P_{2,4} \\ P_{3,1}\ P_{3,2}\ P_{3,4} \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} H_{1,1}\ H_{1,2}\ H_{1,3} \\ H_{2,1}\ H_{2,2}\ H_{2,3} \\ H_{3,1}\ H_{3,2}\ H_{3,3} \end{bmatrix}^{-1} \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix} \quad (4)$$

Pertanto risulta chiaro come l'informazione che codifica le trasformazione tra il piano immagine e il piano di lavoro sia presente, oltre che nella matrice $\mathbf{H}$, anche nella MPP della telecamera. Ponendo infatti $\mathbf{P} = [\mathbf{p}_1 \mid \mathbf{p}_2 \mid \mathbf{p}_3 \mid \mathbf{p}_4]$, dove $\mathbf{p}_i = [P_{1,i}\ P_{2,i}\ P_{3,i}\ P_{4,i}]^T$ è l'i-esima colonna di $\mathbf{P}$, si ha che $[\mathbf{p_1} \mid \mathbf{p_2} \mid \mathbf{p_4}] = \mathbf{H}^{-1}$.

La rettificazione ortogonale inversa dell'immagine di partenza può dunque essere effettuata attraverso il calcolo esplicito della matrice $\mathbf{H}$ a partire dalle corrispondenze tra punti, oppure sfruttando direttamente la matrice di proiezione prospettica nel caso in cui questa sia nota. Ovviamente la seconda strada è quella più conveniente nel nostro caso; pertanto opteremo per una modellizzazione completa

[2] Per una migliore comprensione si veda [2] appendice A.

[3] $\mathbf{A}$ è una matrice che deriva dalle equazioni fornite dalle corrispondenze di punti e $\mathbf{h}$ è il vettore di dimensione 9 contenente gli elementi di $\mathbf{H}$. Per una trattazione più completa consultare [2] capitolo 10.

[4] Il fattore moltiplicativo deriva dal fatto che se una matrice CCD di $n \times m$ elementi fotosensibili viene convertita in un'immagine di $N \times M$ pixel, allora le coordinate u sono legate dalla relazione $u = u_{pix} = N/n \cdot x_{ccd}$ (lo stesso vale per v con M al posto di N e m al posto di n). Nel seguito si farà riferimento sempre a un rapporto 1:1 tra pixel ed elementi CCD.

delle telecamere che tenga in considerazione ogni aspetto necessario alla definizione della MPP.

B. Modello di telecamera PTZ e sistemi di riferimento

La matrice di proiezione prospettica rappresenta il modello geometrico della telecamera, codificando "l'essenza" della trasformazione sia nei suoi aspetti intrinseci che in quelli estrinseci. Un modello realistico di telecamera, che descriva la trasformazione da coordinate 3D a coordinate pixel, oltre che della trasformazione prospettica, deve tener conto di

- trasformazione rigida tra la telecamera e la scena;
- pixelizzazione, cioè la forma e la dimensione della matrice CCD e la sua posizione rispetto al centro ottico.

Cominciamo analizzando il primo aspetto: dotiamo la telecamera di una terna di riferimento destrorsa, spesso definita sistema di riferimento (abbreviato sdr) *standard* della telecamera. Prendiamo come origine il suo centro ottico, l'asse z coincidente con l'asse ottico e l'asse x sempre parallelo al piano di lavoro come mostra la figura 5.

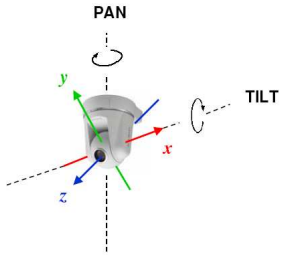

Figura 5. Terna di riferimento della telecamera PTZ e assi per il movimento di pan e tilt.

Il movimento della telecamera è consentito da una coppia di brandeggi motorizzati che assicurano le rotazioni attorno a due assi specifici: nel nostro caso abbiamo identificato il movimento di *pan* con la rotazione attorno a un'asse verticale perpendicolare al piano di lavoro, mentre il movimento di *tilt* è associato alla rotazione attorno all'asse x della telecamera come illustrato in figura 5. Nel caso più semplice (ma solo ideale) in cui questi due assi di rotazione si intersecano perfettamente nel centro ottico della telecamera, l'equazione che descrive la movimentazione risulta:

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \mathbf{R_{pan}}\mathbf{R_{tilt}} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{5}$$

dove $\mathbf{R_{pan}}$ e $\mathbf{R_{tilt}}$ sono le martici di rotazione per i movimenti di pan e tilt rispettivamente, (x', y', z') sono le coordinate di un punto rispetto al sdr della telecamera *prima* della movimentazione, mentre (x'', y'', z'') sono le coordinate dello stesso punto rispetto al sdr della telecamera *dopo* la movimentazione.

Tuttavia la condizione di rotazioni ideali attorno ad assi perpendicolari passanti per il centro ottico spesso viene violata (specialmente se si ha a che fare con il setup di un sistema reale che utilizza telecamere commerciali). In questo caso basta tenere in considerazione gli offset tra gli assi di rotazione reali e quelli ideali, come mostrato nel paragrafo 3.2 di [11], modificando la relazione precedente nel seguente modo:

$$\begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \mathbf{T_{pan}}\mathbf{R_{pan}}\mathbf{T_{pan}^{-1}}\mathbf{T_{tilt}}\mathbf{R_{tilt}}\mathbf{T_{tilt}^{-1}} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{6}$$

dove $\mathbf{R_{pan}}$ e $\mathbf{R_{tilt}}$ sono matrici di rotazione 3×3 che ruotano gli angoli di pan e tilt attorno ai veri assi di rotazione. $\mathbf{T_{pan}}$ e $\mathbf{T_{tilt}}$ rappresentano invece gli offset di questi assi rispetto a quelli ideali passanti per il centro ottico della telecamera.

Infine, per tenere conto del fatto che, in generale, il sistema di riferimento mondo non coincide con quello standard della telecamera, è necessario introdurre una trasformazione rigida che leghi i due sistemi di riferimento come illustrato in figura 6.

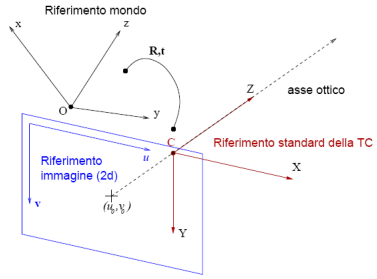

Figura 6. Sistemi di riferimento. Immagine presente in [2] pag. 28.

La trasformazione che porta da coordinate mondo a coordinate telecamera risulta:

$$\begin{bmatrix} x^{tc} \\ y^{tc} \\ z^{tc} \end{bmatrix} = \mathbf{R} \begin{bmatrix} x^{w} \\ y^{w} \\ z^{w} \end{bmatrix} + \mathbf{t} \tag{7}$$

dove $\mathbf{R}$ è una matrice di rotazione 3×3 e $\mathbf{t}$ rappresenta la traslazione tra i due sdr. Nel nostro caso, dal momento che $z^w = 0$ l'equazione può essere riscritta più semplicemente in questo modo:

$$\begin{bmatrix} x^{tc} \\ y^{tc} \\ z^{tc} \end{bmatrix} = \left[\begin{array}{cc|c} R_{1,1} & R_{1,2} & t_1 \\ R_{2,1} & R_{2,2} & t_2 \\ R_{3,1} & R_{3,2} & t_3 \end{array}\right] \begin{bmatrix} x^{w} \\ y^{w} \\ 1 \end{bmatrix} \tag{8}$$

in cui la terza colonna di $\mathbf{R}$ è stata rimpiazzata da $\mathbf{t}$ (che moltiplicata per 1 è come se venisse semplicemente sommata). Abbiamo così catturato attraverso la coppia $(\mathbf{R}, \mathbf{t})$ tutti gli aspetti cosiddetti *estrinseci* del modello della telecamera.

Rivolgiamo ora la nostra attenzione agli aspetti relativi alla pixelizzazione dell'immagine. Come mostra la (8), dalle coordinate mondo possiamo risalire alle coordinate standard della telecamera: il passo mancante per la conoscenza totale del modello sta nel trovare la trasformazione che porta dal sdr della telecamera a quello dell'immagine. Per far questo dobbiamo introdurre alcune grandezze che sono le dirette responsabili del processo di formazione e discretizzazione dell'immagine. Queste grandezze vanno sotto il nome di *parametri intrinseci* del modello e sono codificate in una matrice 3×3

$$\mathbf{A} = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

dove f è la distanza focale, u_0 e v_0 sono le coordinate del punto principale (vedi figura 6), k_u e k_v sono l'inverso della dimensione efficace del pixel lungo le direzioni u e v.

A questo punto, avendo a disposizione sia i parametri intrinseci che quelli estrinseci, il modello della telecamera PTZ è completo e l'equazione di proiezione prospettica assume la forma

$$k\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}-fk_u & 0 & u_0\\0 & -fk_v & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}x^{tc}\\y^{tc}\\z^{tc}\end{bmatrix} \tag{9}$$

Sostituendo la (8) nella (9), l'equazione può essere riscritta mettendo in evidenza la matrice 3×3 (prodotto di $\mathbf{A}$ per la matrice di rototraslazione) che porta dalle coordinate mondo alle coordinate immagine.

$$k\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}-fk_u & 0 & u_0\\0 & -fk_v & v_0\\0 & 0 & 1\end{bmatrix}\left[\begin{array}{cc|c}R_{1,1} & R_{1,2} & t_1\\R_{2,1} & R_{2,2} & t_2\\R_{3,1} & R_{3,2} & t_3\end{array}\right]\begin{bmatrix}x^w\\y^w\\1\end{bmatrix} \tag{10}$$

É importante notare come questa matrice *non* sia la MPP della telecamera, in quanto le matrici di proiezione prospettica hanno sempre dimensione 3×4 essendo associate alle coordinate omogenee. Tuttavia, dato che ai fini della rettificazione ortogonale ci interessano solo le colonne $1, 2$ e 4 della MPP, e poichè la MPP si può scrivere come $\mathbf{P}=\mathbf{A}[\mathbf{R}|\mathbf{t}]$, si dimostra facilmente come questa matrice 3×3 sia effettivamente la MPP privata della terza colonna.

$$\begin{bmatrix}H_{1,1} & H_{1,2} & H_{1,3}\\H_{2,1} & H_{2,2} & H_{2,3}\\H_{3,1} & H_{3,2} & H_{3,3}\end{bmatrix}^{-1}=\begin{bmatrix}-fk_u & 0 & u_0\\0 & -fk_v & v_0\\0 & 0 & 1\end{bmatrix}\left[\begin{array}{cc|c}R_{1,1} & R_{1,2} & t_1\\R_{2,1} & R_{2,2} & t_2\\R_{3,1} & R_{3,2} & t_3\end{array}\right] \tag{11}$$

Abbiamo concluso, pertanto, che la nostra matrice di proiezione porta in sè tutta l'informazione necessaria per effettuare la *rettificazione ortogonale inversa*.

Passiamo ora ad analizzare gli aspetti relativi all'inseguimento del target nel piano di lavoro.

III. Tracking

La soluzione più immediata a problemi di inseguimento (in inglese *tracking*) e di navigazione automatica è rappresentata dall'impiego del filtro di Kalman. Il filtro di Kalman è, come noto, un filtro ricorsivo che valuta lo stato di un sistema dinamico a partire da una serie di misure soggette a rumore. Una delle prime applicazioni del filtro risale alla fine degli anni '60, quando la routine di calcolo ideata da R.E. Kalman venne utilizzata per la stima delle traiettorie nel programma di voli spaziali Apollo della NASA.

Le problematiche di fondo legate al tracking e alla navigazione automatica sono molto simili in quanto consistono nella stima in linea delle variabili di stato di un sistema dinamico in base alle misure affette da rumore provenienti da uno o più sensori. La differenza fondamentale tra le due sta nel fatto che, mentre nel tracking la ricostruzione della traiettoria di stato viene effettuata da una stazione esterna (tipicamente un sensore) senza conoscere gli ingressi di comando e le forze agenti sul target, nella navigazione automatica, invece, la stima dev'essere fatta *on board* per decidere quali comandi applicare in seguito alla pianificazione del percorso. Nel seguito la nostra attenzione sarà rivolta esclusivamente alle problematiche di inseguimento (*tracking*).

A. Modellizzazzione del sistema

Punto di partenza per la costruzione di un algoritmo di inseguimento è la modellizzazione del sistema dinamico su cui si vuole effettuare il tracking. Cominciamo allora col descrivere il sistema nel suo complesso e fissare le variabili di stato.

Come già accennato nell'introduzione, il nostro obiettivo è quello di inseguire un veicolo mobile che si muove lungo il piano di lavoro attraverso una serie di telecamere PTZ (*visual tracking*) coadiuvate da una rete di sensori in radiofrequenza. Per fare questo è stato introdotto un sistema di riferimento globale, detto *sistema mondo*, nel quale risulta comodo esprimere posizione e velocità del target durante il suo percorso nel piano (vedi fig. 2). Partendo da un'accezione il più generale possibile si può supporre che la coppia veicolo-sistema di misura possa essere descritta da un modello non lineare a tempo continuo del tipo:

$$\begin{cases}\dot{\mathbf{x}}(t)=\mathbf{f}(\mathbf{x}(t),\mathbf{u}(t))+\mathbf{w}(t)\\\mathbf{z}(t)=\mathbf{h}(\mathbf{x}(t))+\mathbf{v}(t)\end{cases} \tag{12}$$

in cui la prima equazione descrive la dinamica del veicolo soggetto a ingressi (o comandi) sconosciuti $\mathbf{u}(t)$, mentre la seconda rappresenta l'equazione di osservazione e descrive le misure provenienti dal sistema di rilevamento. Nel nostro caso, non essendo noto a priori il comando $\mathbf{u}(t)$, i suoi effetti vengono conglobati nel termine di rumore additivo $\mathbf{w}(t)$, eliminando di fatto la dipendenza di $\mathbf{f}$ da $\mathbf{u}(t)$. Nel seguito vedremo come sia possibile trattare questo tipo di problema in altro modo (vedi sezione III-C).

Rimane ora da determinare la forma di $\mathbf{f}$ e $\mathbf{h}$. Queste due funzioni possono essere ricavate sfruttando la fisica del sistema, sia per quanto riguarda il veicolo che per quanto riguarda il dispositivo di misura. In problemi di tracking, tuttavia, sarebbe necessario conoscere troppi parametri (ad esempio inerzia, attriti, etc.) per ottenere un modello realistico del veicolo da inseguire: pertanto in molti casi ci si accontenta di un modello semplificato che concentra tutta la parte relativa alle forze agenti (compresi i comandi) nel termine di rumore additivo $\mathbf{w}(t)$. Si ottiene così un *modello cinematico lineare* che ben si presta a descrivere un bersaglio il cui moto nel piano sia approssimativamente rettilineo uniforme.

Nel nostro caso, fissiamo come stati del sistema le coordinate x e y associate alla posizione del target nel piano rispetto al sdr mondo, x^w e y^w, e le rispettive velocità lungo le direzioni x e y, $\dot{x}^w$ e $\dot{y}^w$. Dunque il vettore di stato $\mathbf{x}$ risulta

$$\mathbf{x}=\begin{bmatrix}x^w & \dot{x}^w & y^w & \dot{y}^w\end{bmatrix}^T$$

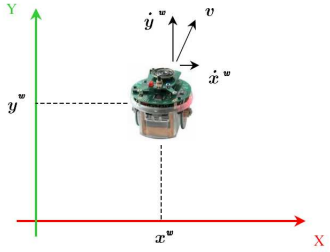

Figura 7. Modello cinematico.

L'equazione di transizione di stato del *modello cinematico lineare* assumerà pertanto la forma

$$\dot{\mathbf{x}}(t)=\mathbf{A}\mathbf{x}(t)+\mathbf{w}(t)$$

che può essere riscritta esplicitando $\mathbf{A}$ e $\mathbf{w}(t)$ in questo modo

$$\frac{d}{dt}\begin{bmatrix} x^w \\ \dot{x}^w \\ y^w \\ \dot{y}^w \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x^w \\ \dot{x}^w \\ y^w \\ \dot{y}^w \end{bmatrix} + \begin{bmatrix} 0 \\ f_x \\ 0 \\ f_y \end{bmatrix} \quad (13)$$

dove f_x e f_y indicano le componenti delle forze agenti. Discretizzando la (13) si ottiene un'equazione di stato del tipo

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{w}(k) \quad (14)$$

dove $\mathbf{v}(k)$ rappresenta la discretizzazione delle forze agenti. La matrice di transizione $\mathbf{F}$ assume la forma

$$\mathbf{F} = e^{\mathbf{A}T} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

in cui T è il periodo di campionamento, ovvero l'intervallo di tempo che intercorre tra due osservazioni consecutive. É importante notare come la matrice $\mathbf{F}$ sia diagonale a blocchi, conseguenza del fatto che le dinamiche lungo le direzioni x e y sono separate; grazie a questa proprietà è possibile suddividere il sistema di dimensione 4 in due sottosistemi indipendenti di dimensione 2.

La seconda equazione, quella che descrive le osservazioni, nel nostro caso sarà semplicemente

$$\mathbf{z}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \quad (15)$$

dove

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Pertanto, mettendo insieme la (14) e la (15) si ottiene il modello completo del veicolo che va sotto il nome di **modello a velocità costante** (vedi a proposito [3] par. 2.7).

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{w}(k) \\ \mathbf{z}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{v}(k) \end{cases} \quad (16)$$

Il rumore di modello $\mathbf{w}(k)$ serve quindi a tener conto di tutte le approssimazioni che vengono fatte in fase di modellizzazione. L'errore che si commette viene così schematizzato utilizzando un rumore bianco additivo. Occorre quindi fornire un'espressione ragionevole della covarianza $\mathbf{Q}$ del rumore di modello. Con il modello a velocità costante la covarianza di $\mathbf{w}(k)$ ha la forma:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q_x} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_y} \end{bmatrix}$$

con

$$\mathbf{Q_x} = \mathbf{Q_y} = q * \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix}$$

dove q è un valore costante che andrebbe stimato sulla base di quanto si conosce a priori sulla dinamica del target. Nel nostro caso ci potrebbe venire in aiuto avere un'idea dell'entità delle deviazioni standard della velocità angolare e di quella tangenziale del bersaglio. Un'altro modo per settare il valore di q è quello di operare un *test di bianchezza* [5] in linea in modo da mantenere il filtro di Kalman sempre "accordato" (in inglese *tuned*): questo potrebbe portare a una scelta di q diversa ad ogni istante di campionamento.

B. Filtro di Kalman

Abbiamo quindi utilizzato il modello a velocità costante per implementare un filtro di Kalman discreto a guadagno costante che dall'informazione corrente sulla posizione del veicolo calcola una predizione della posizione al passo successivo. Le equazioni ricorsive che lo caratterizzano sono le seguenti:

- *Inizializzazione*
$$\hat{\mathbf{x}}(k_0|k_0-1) = \mu_0$$
- *Aggiornamento*
$$\hat{\mathbf{x}}(k|k) = \hat{\mathbf{x}}(k|k-1) + \mathbf{K}[\mathbf{z}(k) - \mathbf{C}\hat{\mathbf{x}}(k|k-1)], \quad k \geq k_0$$
- *Predizione*
$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}\hat{\mathbf{x}}(k|k), \quad k \geq k_0$$

L'inizializzazione dell'algoritmo, cioè la determinazione del valore μ_0 da associare a $\hat{\mathbf{x}}(k_0|k_0-1)$, può essere risolta in pratica utilizzando le prime due misure disponibili $\mathbf{z}(k_0)$ e $\mathbf{z}(k_0+1)$. Posto per semplicità $k_0 = 0$ e $\boldsymbol{\xi} = [x^w y^w]^T$, si effettua un'inizializzazione in due passi di tipo differenziale (*two steps initialization*) in questo modo:

$$\hat{\boldsymbol{\xi}}(1|1) = \mathbf{z}(1)$$
$$\dot{\hat{\boldsymbol{\xi}}}(1|1) = \frac{\mathbf{z}(1) - \mathbf{z}(0)}{T}$$

Dopo l'arrivo di $\mathbf{z}(1)$ (seconda misura) abbiamo dunque a disposizione $\hat{\mathbf{x}}(1|1)$, che è costituito dalle componenti di $\hat{\boldsymbol{\xi}}(1|1)$ e $\dot{\hat{\boldsymbol{\xi}}}(1|1)$ prese nel giusto ordine. Di conseguenza prima dell'arrivo della terza misura si cacola la predizione $\hat{\mathbf{x}}(2|1)$ che verrà poi filtrata per mezzo dell'osservazione $\mathbf{z}(2)$.

Un'altra considerazione importante è quella relativa al calcolo della matrice di covarianza dell'errore di misura $\mathbf{R}$. In molte applicazioni questa matrice può essere ricavata, come già detto in precedenza, a partire dalle specifiche del dispositivo che effettua le misure. Nel nostro caso la telecamera fornisce delle misure abbastanza accurate dell'effettiva posizione del veicolo quando lo sta inquadrando: possiamo quindi optare per un modello a osservazioni perfette o quasi, ovvero un modello caratterizzato da una matrice $\mathbf{R} \cong \mathbf{0}$. Per una trattazione approfondita del problema della singolarità di $\mathbf{R}$ si rimanda a [4] cap. 9.4.

Come mostra le figura 8, nel caso di osservazioni perfette o quasi il filtro di Kalman dà un'ottima ricostruzione del percorso del veicolo nel piano. Allo stesso tempo si nota però un certo scostamento tra le due traiettorie nei tratti in cui il veicolo curva (vedi fig. 9). Questo può essere spiegato in relazione al modello che è stato scelto per

[5] Per determinare la scelta di q si cerca quel valore per cui la matrice $\mathbf{Q}$ rende minima la differenza tra il predittore vero e quello approssimato con il nostro modello. Questo corrispone a trovare la matrice $\mathbf{Q}$ per cui l'errore di predizione $\epsilon_{\mathbf{Q}}(k) = \mathbf{z}(k) - \mathbf{C}\hat{\mathbf{x}}_{\mathbf{Q}}(k|k-1)$ è un rumore bianco. Per verificare questa condizione, ovvero la bianchezza dell'errore di predizione (condizione sufficiente e necessaria per l'ottimalità del filtro) si possono utilizzare dei criteri statistici che permettono di stabilire se una sequenza dell'errore di predizione ottenuto con un dato $\mathbf{Q}$ può essere interpretabile come tratto di realizzazione di un rumore bianco. Questi criteri vanno sotto il nome di *test di bianchezza* ([5] cap. 11). Nel nostro caso abbiamo implementato un test di bianchezza fuori linea al fine di trovare la matrice varianza dell'errore di aggiornamento di stato $\mathbf{Q}$ che minimizza l'errore di predizione e si avvicina pertanto a quella del modello vero.

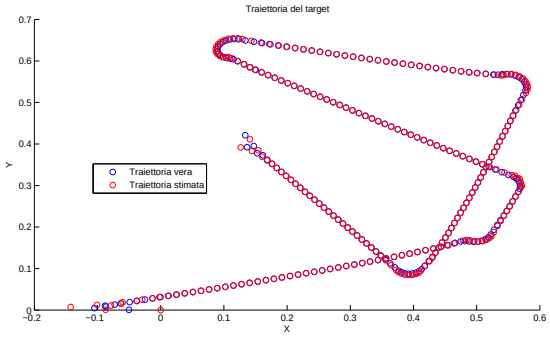

Figura 8. Tracking con osservazioni perfette

descrivere la dinamica del target. Il modello descritto da (16), infatti, non considera possibili accelerazioni del veicolo, o meglio, le tollera fino a una certa soglia determinata dalla matrice $\mathbf{Q}$. Nel momento in cui queste accelerazioni compaiono, come per esempio quando il target percorre un tratto curvilineo, il modello non è abbastanza accurato per tenerne conto e gli errori di inseguimento aumentano necessariamente.

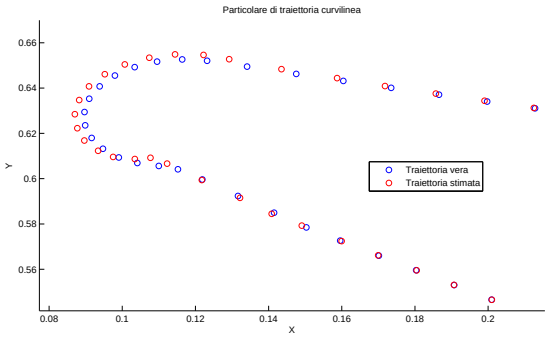

Figura 9. Particolare di un pezzo di traiettoria curvilinea

In questo caso si può migliorare il tracking utilizzando alcune tecniche che vanno sotto il nome di **maneuvering tracking** (tracking in presenza di manovre, si veda in proposito [3]).

C. Maneuvering target

Come già anticipato, utilizzando un modello dinamico semplificato ci sono molte variabili che vengono trascurate e di cui si cerca di tener conto nel rumore di modello. Quando però sul veicolo vengono ad agire forze esterne sconosciute (manovre), si possono utilizzare delle tecniche per migliorare le prestazioni del filtro.

Un target in fase di manovra è caratterizzato da un'equazione che è una naturale estensione della (14)

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{u}(k) + \mathbf{w}(k) \tag{17}$$

in cui $\mathbf{w}(k)$ è un rumore bianco a media nulla con matrice di covarianza $\mathbf{Q}(k)$ (vale lo stesso discorso visto in III-A) e gli ingressi $\mathbf{u}(k)$ (comandi di manovra) non sono noti.[6] Per ovviare alla mancanza di informazione relativa agli ingressi del modello, si possono percorrere due strade

1) modellare l'ingresso sconosciuto come un processo casuale;
2) assumere l'ingresso come non casuale e stimarlo in tempo reale.

Per il primo approccio elencheremo solo alcune idee, mentre per il secondo metteremo in pratica la tecnica del filtro a dimensione variabile.

1) Ingresso modellato come processo casuale: Il comando di manovra, in accordo alle proprietà statistiche del processo, può venir classificato come:

A. rumore bianco
B. rumore markoviano autocorrelato

Inoltre possono essere considerati diversi livelli di rumore a seconda della dinamica del target e il sistema può passare da un livello a un altro (tramite switch) a seconda della soglia raggiunta da un certo indice di manovra. Questi metodi rappresentano comunque sempre delle approssimazioni perchè i comandi del target in genere non sono mai processi stocastici ma segnali deterministici.

2) Ingresso ricavato o stimato in tempo reale: Anche in questo approccio si possono distinguere sostanzialmente due casi, ovvero:

A. lo stato stimato viene corretto dalla stima dell'ingresso (*input estimation*)
B. si aumenta la dimensione del vettore di stato aggiungendo l'ingresso che quindi diventa una nuova componente da stimare (*variable dimension*)

In generale la scelta del metodo da applicare dipende dalla durata della manovra in relazione al tempo di campionamento. Se le misure sono sufficientemente frequenti durante la manovra, tali cioè da mantenere l'inseguimento durante questa, gli approcci di tipo B sono preferibili. Nel caso invece la manovra si verifichi tra due istanti di campionamento successivi, con caratteristica essenzialmente istantanea, è meglio trascurare i dettagli della manovra e concentrarsi sulla sua rilevazione correggendo la stima successiva in accordo con i metodi di tipo A. Nel nostro caso la frequenza di campionamento è tale da garantire l'inseguimento del veicolo durante le fasi di manovra e pertanto opteremo per la soluzione 2.B.

VARIABLE DIMENSION FILTERING

Il concetto alla base di questo metodo è considerare la manovra del target come parte interna alla sua dinamica e quindi compresa nello stato. Si utilizzano pertanto due modelli del sistema, uno che agisce in assenza di manovre (*modello normale*) e l'altro, di dimensione maggiore, che viene utilizzato solo quando agiscono dei comandi sul veicolo (*maneuvering model*). Il modello con stato aumentato viene dunque utilizzato per la sola durata della manovra, per poi tornare a considerare il modello a velocità costante (vedi [3] par. 2.7). Il vettore di stato $\mathbf{x}$ per il nuovo modello assume la forma

$$\mathbf{x}^m = \begin{bmatrix} x^w & \dot{x}^w & y^w & \dot{y}^w & \ddot{x}^w & \ddot{y}^w \end{bmatrix}'$$

dove l'apice m sta per "maneuvering". Questo modello ad accelerazione (quasi) costante si può ricavare facilmente da (16); la matrice di aggiornamento di stato diventa

[6] Riferendoci sempre al modello a velocità costante rimane da definire solo la matrice G:

$$G = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \tag{18}$$

$$\mathbf{F}^m = \begin{bmatrix} 1 & T & 0 & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & 0 & T & 0 \\ 0 & 0 & 1 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

e quella di uscita

$$\mathbf{C}^m = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{20}$$

Nello stesso modo si ha la matrice varianza dell'errore di modello:

$$\mathbf{Q}^m = \begin{bmatrix} \frac{T^5}{20} & \frac{T^4}{8} & 0 & 0 & \frac{T^3}{6} & 0 \\ \frac{T^4}{8} & \frac{T^3}{3} & 0 & 0 & \frac{T^2}{2} & 0 \\ 0 & 0 & \frac{T^5}{20} & \frac{T^4}{8} & 0 & \frac{T^3}{6} \\ 0 & 0 & \frac{T^4}{8} & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^3}{6} & \frac{T^2}{2} & 0 & 0 & T & 0 \\ 0 & 0 & \frac{T^3}{6} & \frac{T^2}{2} & 0 & T \end{bmatrix} \tag{21}$$

Resta ora da considerare il problema di come rilevare una manovra e passare così dal modello normale a quello aumentato. La soluzione più semplice è utilizzare un valore scalare, detto *indice di manovra*, e una soglia: quando l'indice supera il valore di soglia si passa dal modello normale a quello aumentato. Un buon indice potrebbe essere una media *fading-memory* dell'innovazione relativa allo stimatore basato sul modello normale. In formule

$$\rho(k) = \alpha\rho(k-1) + \epsilon_{\boldsymbol{\nu}}(k) \tag{22}$$

con

$$\epsilon_{\boldsymbol{\nu}}(k) = \boldsymbol{\nu}^T(k)\mathbf{S}^{-1}(k)\boldsymbol{\nu}(k)$$

dove $\alpha \in (0,1)$ è il cosiddetto fattore di sconto (*discount factor*), $\boldsymbol{\nu}(k)$ è l'innovazione[7] e $\mathbf{S}(k)$ la sua covarianza. Dal momento che $\epsilon_{\boldsymbol{\nu}}(k)$, sotto ipotesi di Gaussianità dei segnali in gioco, è distribuita come una χ^2 con $n_z = 2$ gradi di libertà (n_z è la dimensione del vettore delle osservazioni), a regime si ottiene che la memoria effettiva su cui viene testata la presenza di manovre è pari a

$$s = \frac{1}{1-\alpha}$$

Il parametro s indica pertanto il ritardo con cui viene rilevata una variazione significativa nel modello.

Se $\rho(k)$ supera un certo valore stabilito[8], allora significa che sul veicolo è attiva una qualche manovra e perciò lo stimatore dall'istante successivo prende in considerazione il *maneuvering model*.

Essenso ρ un parametro a memoria finita, l'errore iniziale dovuto alla variazione di accelerazione (picco iniziale) influisce anche sugli istanti successivi se il *discount factor* α è prossimo a uno: di conseguenza la curva ottenuta con il maneuvering model impiega un certo tempo a portarsi sotto quella risultante dal modello normale (vedi fig. 11). La diminuzione dell'errore di stima si può vedere meglio se si considera un indice senza memoria ($\alpha = 0$) che valuta solo l'errore istante per istante (vedi fig. 12).

[7] L'innovazione è pari a $\boldsymbol{\nu}(k) = \mathbf{z}(k) - \mathbf{C}\hat{\mathbf{x}}(k|k-1)$.

[8] La soglia può essere calcolata a partire da test statistici dal momento che $\epsilon_{\boldsymbol{\nu}}(k)$ a regime è distribuita come una χ^2 con due gradi di libertà.

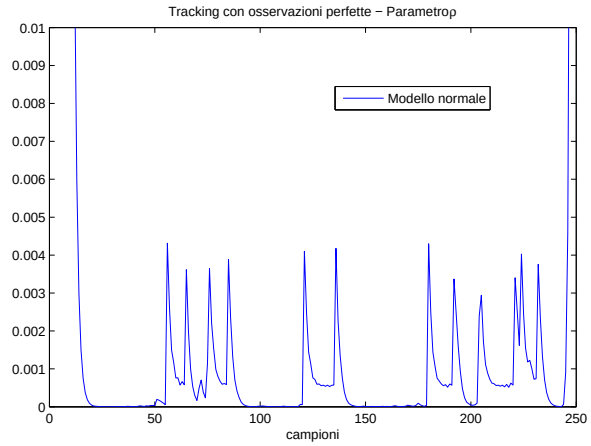

Figura 10. Andamento del parametro ρ basato sul modello normale.

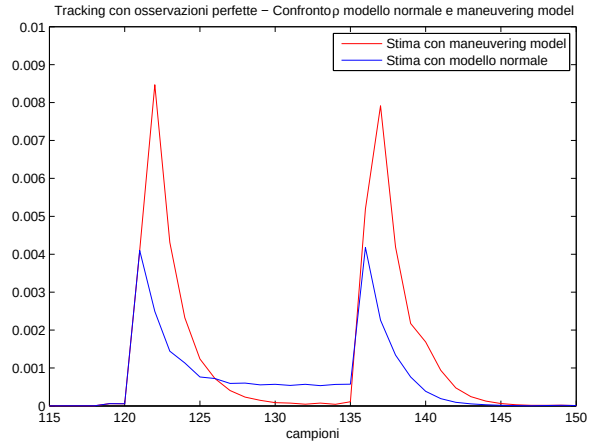

Figura 11. Confronto tra il ρ calcolato in base al modello normale e al modello aumentato con discount factor $\alpha = 0.5$.

Sorge ora il problema di come inizializzare il nuovo modello, in particolare le componenti di accelerazione. Se il maneuvering model viene attivato all'istante k in seguito alla rilevazione di una manovra, si può assumere che la manovra sia iniziata all'istante $k-s$, cioè esattamente s campioni prima. La stima dell'accelerazione all'istante $k-s$ diventa:

$$\hat{x}^m_{4+i}(k-s|k-s) = \frac{2}{T^2}[z_i(k-s) - \hat{z}_i(k-s|k-s-1)]$$

dove $i = 1, 2$ e $4+i$ indica la posizione nel vettore di stato $\mathbf{x}$. Le componenti relative alle posizioni, invece, vengono eguagliate alle misure corrispondenti prese sempre all'istante $k-s$:

$$\hat{x}^m_{2i-1}(k-s|k-s) = z_i(k-s)$$

Infine le componenti della velocità vengono corrette con la stima dell'accelerazione:

$$\hat{x}^m_{2i}(k-s|k-s) = \hat{x}_{2i}(k-s|k-s-1) + T\hat{x}^m_{4+i}(k-s|k-s)$$

Abbiamo quindi risolto il problema di come passare dal modello normale a quello aumentato. Resta ora da risolvere il problema

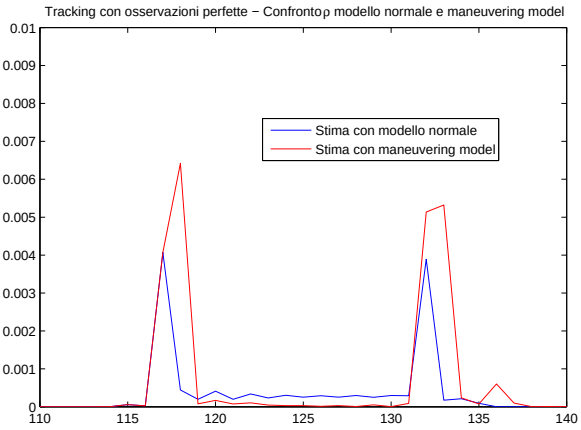

Figura 12. Confronto tra il ρ calcolato in base al modello normale e al modello aumentato con discount factor $\alpha = 0$.

opposto. Quando la manovra finisce e il veicolo ritorna ad avere accelerazione nulla, infatti, sarebbe preferibile utilizzare nuovamente il modello normale. Per far questo si considera un altro parametro, questa volta legato allo stato aggiunto:

$$\delta_a(k) = \hat{\mathbf{a}}^T(k|k)[\mathbf{P}_a^m(k|k)]^{-1}\hat{\mathbf{a}}(k|k) \tag{23}$$

dove $\hat{\mathbf{a}}(k|k) = [\hat{x}_5^m(k|k), \hat{x}_6^m(k|k)]^T$ sono le due componenti di accelerazione del vettore $\hat{\mathbf{x}}(k|k)$ e $\mathbf{P}_a^m(k|k)$ è il corrispondente blocco della matrice di covarianza del maneuvering model.

Al contrario dell'indice ρ, il passaggio dal modello aumentato a quello normale avviene quando δ_a scende sotto una certa soglia, indicando che contributo dell'accelerazione è praticamente trascurabile.

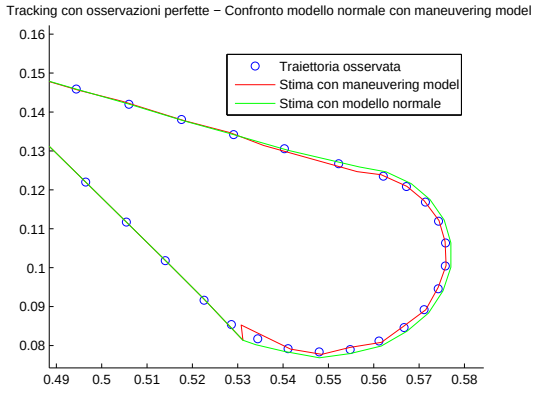

Figura 13. Confronto tra le traiettorie ottenute con il modello normale e con il maneuvering model.

Il motivo per cui conviene utilizzare i due modelli e non solo, ad esempio, il maneuvering model, è che quando il target non ha accelerazione (assenza di manovre) l'utilizzo di un modello di ordine maggiore aumenta l'errore di stima sia per la posizione che per la velocità. Alternando i due sistemi, invece, si riescono ad avere buone prestazioni sia in assenza che in presenza di manovre. I miglioramenti che porta questa tecnica sono però davvero minimi (vedi fig. 13) quindi nel seguito per alleggerire l'onere computazionale utilizzeremo il semplice modello a velocità costante.

IV. Sensor fusion

Il tracking può venire però migliorato utilizzando le informazioni provenienti da due o più fonti. Come mostra [9], la possibilità di fondere insieme le stime o le misure generate da sensori anche eterogenei dà un notevole contributo al miglioramento delle stime stesse. Nel nostro caso, tuttavia, si è deciso di sviluppare algoritmi di *sensor fusion* solo tra telecamere, per motivi legati all'enorme varianza d'errore che caratterizza le misure provenienti dai sensori in radiofrequenza.

A. Switching mode sensori-telecamere

Come appena accennato le informazioni provenienti dalle telecamere sono molto più accurate di quelle provenienti dai sensori. In figura 14 questo aspetto è dimostrato in maniera molto chiara: infatti, quando il target ricade sotto la zona coperta dalla telecamera (nel caso della figura questo succede in due occasioni), l'errore di misura subisce una drastica riduzione. Pertanto, invece di fondere le informazioni dei sensori con quelle delle telecamere, abbiamo optato per un funzionamento *switching* del sistema di inseguimento. Quando si hanno a disposizione misure provenienti da telecamera si usano quelle, altrimenti, se nessuna telecamera sta inquadrando il target, si utilizzano le informazioni provenienti dai soli sensori. La rete di sensori viene dunque adoperata solo in fase di inizializzazione per localizzare il target.

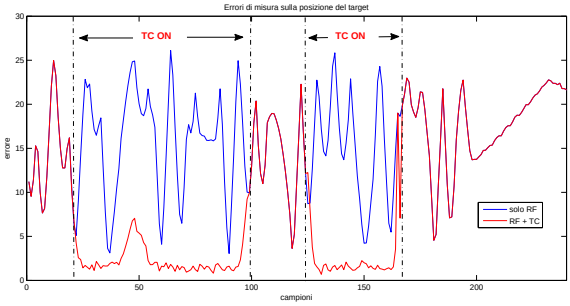

Figura 14. Confronto tra gli errori di misura dei sensori e delle telecamere.

B. Tracking con 2 telecamere

Consideriamo dapprima un sistema costituito da due sole telecamere PTZ che chiameremo A e B. Le corrispondenti sequenze di osservazioni $\mathbf{z}_k^{(A)}$ e $\mathbf{z}_k^{(B)}$ sono disponibili ad ogni istante di campionamento k. Lo scopo del tracking cooperativo tra queste due videocamere è quello di ottenere una stima dello stato che massimizzi la probabilità *a posteriori* $P[\mathbf{x}_k|\mathbf{z}_{1:k}^{(A)}, \mathbf{z}_{1:k}^{(B)}]$. Per far questo si può pensare di utilizzare le informazioni provenienti dalle singole telecamere in maniera congiunta, pesando opportunamente il contributo fornito da ciascuna telecamera, in modo che le prestazioni di entrambe aumentino.

Dal momento che come soluzione al problema del tracking è stato scelto il filtro di Kalman, risulta naturale modificarlo leggermente secondo quelle che sono le nostre esigenze. Definiamo quindi un nuovo modello, a partire da quello ricavato nella (16), su cui calcolare successivamente il filtro. Per la derivazione del nuovo modello distinguiamo però due casi, che corrispondono ai due passi successivi dell'algoritmo di Kalman (il passo di aggiornamento e il passo di predizione).

1) Metodo I - fusione al passo di predizione: Per prima cosa definiamo il vettore di stato del nuovo modello come il vettore esteso le cui componenti sono i vettori di stato delle singole telecamere.

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}^{(A)} \\ \mathbf{x}^{(B)} \end{bmatrix}$$

Ricordando (16) otteniamo un modello esteso del tipo

$$\begin{aligned} \mathbf{x}_{k+1} &= \begin{bmatrix} \mathbf{x}_{k+1}^{(A)} \\ \mathbf{x}_{k+1}^{(B)} \end{bmatrix} = \begin{bmatrix} (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}^{(A)} & \mathbf{\Lambda}\mathbf{F}^{(B)} \\ \mathbf{\Lambda}\mathbf{F}^{(A)} & (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}^{(B)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k^{(A)} \\ \mathbf{x}_k^{(B)} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_k^{(A)} \\ \mathbf{w}_k^{(B)} \end{bmatrix} \\ \mathbf{z}_k &= \begin{bmatrix} \mathbf{z}_k^{(A)} \\ \mathbf{z}_k^{(B)} \end{bmatrix} = \begin{bmatrix} \mathbf{H}^{(A)} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}^{(B)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k^{(A)} \\ \mathbf{x}_k^{(B)} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_k^{(A)} \\ \mathbf{v}_k^{(B)} \end{bmatrix} \end{aligned} \tag{24}$$

dove $\mathbf{F}^{(A)} = \mathbf{F}^{(B)} = \mathbf{F}$ e $\mathbf{H}^{(A)} = \mathbf{H}^{(B)} = \mathbf{H}$ dal modello valido per un singolo sensore (telecamera nel nostro caso), $\mathbf{w}_k$ e $\mathbf{v}_k$ rappresentano i rumori di processo e di misura, con media zero e matrici di covarianza rispettivamente $\mathbf{Q}$ e $\mathbf{R}$, e $\mathbf{\Lambda}$ è un peso predefinito che serve a mediare i contributi delle singole telecamere.

Se si scrive la matrice covarianza dell'errore di stima come

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{k|k}^{1,1} & \mathbf{P}_{k|k}^{1,2} \\ \mathbf{P}_{k|k}^{2,1} & \mathbf{P}_{k|k}^{2,2} \end{bmatrix}$$

allora la covarianza dell'errore di predizione può essere calcolata a partire da (24) e risulta

$$\begin{aligned} \mathbf{P}_{k+1|k}^{1,1} &= \mathbf{Q}^{(A)} + (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T \\ &\quad + \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T\mathbf{\Lambda}^T + \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{2,1}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T \\ &\quad + (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{1,2}\mathbf{F}^T\mathbf{\Lambda}^T \\ \mathbf{P}_{k+1|k}^{2,2} &= \mathbf{Q}^{(B)} + (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T \\ &\quad + \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T\mathbf{\Lambda}^T + \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{2,1}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T \\ &\quad + (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{1,2}\mathbf{F}^T\mathbf{\Lambda}^T \\ \mathbf{P}_{k+1|k}^{1,2} &= (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T\mathbf{\Lambda}^T + \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T \\ &\quad + \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{2,1}\mathbf{F}^T\mathbf{\Lambda}^T + (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{1,2}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T \\ \mathbf{P}_{k+1|k}^{2,1} &= \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T + (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T\mathbf{\Lambda}^T \\ &\quad + \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{2,1}\mathbf{F}^T\mathbf{\Lambda}^T + (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{1,2}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T \end{aligned}$$

Poichè le due telecamere A e B costituiscono due sistemi di tracking indipendenti, i termini che non sono sulla diagonale sono sufficientemente piccoli rispetto a quelli sulla diagonale affinchè si possano trascurare. Per ridurre i calcoli e per partizionare i processi A e B delle singole telecamere, mantenendo comunque lo scambio di informazioni tra le due, appare dunque ragionevole porre

$$\begin{aligned} \mathbf{P}_{k+1|k}^{1,1} &= \mathbf{Q}^{(A)} + (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T \\ &\quad + \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T\mathbf{\Lambda}^T \\ \mathbf{P}_{k+1|k}^{2,2} &= \mathbf{Q}^{(B)} + (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T(\mathbf{I}-\mathbf{\Lambda})^T \\ &\quad + \mathbf{\Lambda}\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T\mathbf{\Lambda}^T \end{aligned}$$

Il sistema complessivo può allora essere separato in due processi indipendenti, cioè

$$\begin{cases} \mathbf{x}_{k+1}^{(A)} = (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{x}_k^{(A)} + \mathbf{\Lambda}\mathbf{F}\mathbf{x}_k^{(B)} + \mathbf{w}_k^{(A)} \\ \mathbf{z}_k^{(A)} = \mathbf{H}\mathbf{x}_k^{(A)} + \mathbf{v}_k^{(A)} \end{cases} \tag{25}$$

e

$$\begin{cases} \mathbf{x}_{k+1}^{(B)} = (\mathbf{I}-\mathbf{\Lambda})\mathbf{F}\mathbf{x}_k^{(B)} + \mathbf{\Lambda}\mathbf{F}\mathbf{x}_k^{(A)} + \mathbf{w}_k^{(B)} \\ \mathbf{z}_k^{(B)} = \mathbf{H}\mathbf{x}_k^{(B)} + \mathbf{v}_k^{(B)} \end{cases} \tag{26}$$

Risulta chiaro da (25) e (26) come lo scambio di informazioni tra le due telecamere avvenga al passo di predizione per mezzo della matrice $\mathbf{\Lambda}$.

2) Metodo II - fusione al passo di aggiornamento: Un altro modo di procedere è quello di utilizzare la matrice dei pesi $\mathbf{\Lambda}$ nell'equazione di osservazione. In quest'ottica il modello (16) si modifica in

$$\begin{aligned} \mathbf{x}_{k+1} &= \begin{bmatrix} \mathbf{x}_{k+1}^{(A)} \\ \mathbf{x}_{k+1}^{(B)} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^{(A)} & \mathbf{0} \\ \mathbf{0} & \mathbf{F}^{(B)} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k^{(A)} \\ \mathbf{x}_k^{(B)} \end{bmatrix} + \begin{bmatrix} \mathbf{w}_k^{(A)} \\ \mathbf{w}_k^{(B)} \end{bmatrix} \\ \mathbf{z}_k &= \begin{bmatrix} \mathbf{z}_k^{(A)} \\ \mathbf{z}_k^{(B)} \end{bmatrix} = \begin{bmatrix} (\mathbf{I}-\mathbf{\Lambda})\mathbf{H} & \mathbf{\Lambda}\mathbf{H} \\ \mathbf{\Lambda}\mathbf{H} & (\mathbf{I}-\mathbf{\Lambda})\mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k^{(A)} \\ \mathbf{x}_k^{(B)} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_k^{(A)} \\ \mathbf{v}_k^{(B)} \end{bmatrix} \end{aligned} \tag{27}$$

In maniera analoga al primo approccio, per l'indipendenza delle osservazioni delle due telecamere, si possono partizionare i processi A e B in questo modo

$$\begin{cases} \mathbf{x}_{k+1}^{(A)} = \mathbf{F}\mathbf{x}_k^{(A)} + \mathbf{w}_k^{(A)} \\ \mathbf{z}_k^{(A)} = (\mathbf{I}-\mathbf{\Lambda})\mathbf{H}\mathbf{x}_k^{(A)} + \mathbf{\Lambda}\mathbf{H}\mathbf{x}_k^{(B)} + \mathbf{v}_k^{(A)} \end{cases} \tag{28}$$

e

$$\begin{cases} \mathbf{x}_{k+1}^{(B)} = \mathbf{F}\mathbf{x}_k^{(B)} + \mathbf{w}_k^{(B)} \\ \mathbf{z}_k^{(B)} = \mathbf{\Lambda}\mathbf{H}\mathbf{x}_k^{(A)} + (\mathbf{I}-\mathbf{\Lambda})\mathbf{H}\mathbf{x}_k^{(B)} + \mathbf{v}_k^{(B)} \end{cases} \tag{29}$$

Anche in questo caso si nota abbastanza chiaramente come il passaggio di informazione tra le due telecamere avvenga al passo di aggiornamento, modificando di fatto l'equazione di osservazione il cui legame con la stima filtrata è ben noto.

3) Scelta dei pesi: Il peso $\mathbf{\Lambda}$ viene usato per bilanciare i contributi delle differenti telecamere. Il criterio di assegnazione dei pesi non è univoco e molte considerazioni possono essere fatte a riguardo. Spesso vengono utilizzate delle tecniche di tipo statistico, come per esempio assegnare il peso in proporzione inversa rispetto al corrispondente livello di rumore (a tal proposito si veda [18]).

Nel nostro caso si è scelto di assegnare i pesi in modo deterministico: la scelta più semplice è quella di assegnare lo stesso peso a tutte le misure (*pesi uniformi*). In formule:

$$\lambda_k^i = \frac{1}{n} \tag{30}$$

dove n è il numero di telecamere che stanno inquadrando l'oggetto. Un'altra possibilità è quella di assegnare un peso maggiore alle telecamere che stanno inquadrando un target che si trova molto vicino alla loro posizione. É più probabile, infatti, che la misura effettuata sull'immagine da una telecamera vicina sia più precisa e meno corrotta da rumore rispetto a quella di una telecamera che sta inquadrando la scena da lontano. I pesi allora possono essere assegnati secondo la formula:

$$\lambda_k^i = \frac{1}{n-1}\left(1 - \frac{d_k^i}{\sum_{j=1}^n d_k^j}\right) \tag{31}$$

dove n è sempre il numero di telecamere, d_k^i e' la distanza del target dalla telecamera i-esima all'istante k e $\sum_{j=1}^n d_k^j$ e' la somma di tutte le distanze sempre all'istante k. Nella formula compare il fattore moltiplicativo $\frac{1}{n-1}$ che è un'indice di normalizzazione.

Quanto più il target sarà vicino alla telecamera i-esima, tanto più il suo peso λ_k^i sarà elevato. Quindi, nel caso di sole due telecamere sarà $n = 2$ e la precedente diventa

$$\lambda_k^i = 1 - \frac{d_k^i}{d_k^i + d_k^j}$$

Si ottiene così un modello (e di conseguenza un filtro) tempo variante dal momento che le distanze e quindi i pesi possono cambiare ad ogni istante. Seguendo il metodo illustrato in sezione IV-B1 si ottiene il seguente modello

$$\begin{aligned}\mathbf{x}_{k+1} &= \begin{bmatrix}\mathbf{x}_{k+1}^{(A)}\\ \mathbf{x}_{k+1}^{(B)}\end{bmatrix} = \begin{bmatrix}\lambda_k^{(A)}\mathbf{F} & \lambda_k^{(B)}\mathbf{F}\\ \lambda_k^{(A)}\mathbf{F} & \lambda_k^{(B)}\mathbf{F}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k^{(A)}\\ \mathbf{x}_k^{(B)}\end{bmatrix} + \begin{bmatrix}\mathbf{w}_k^{(A)}\\ \mathbf{w}_k^{(B)}\end{bmatrix}\\ \mathbf{z}_k &= \begin{bmatrix}\mathbf{z}_k^{(A)}\\ \mathbf{z}_k^{(B)}\end{bmatrix} = \begin{bmatrix}\mathbf{H} & \mathbf{0}\\ \mathbf{0} & \mathbf{H}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k^{(A)}\\ \mathbf{x}_k^{(B)}\end{bmatrix} + \begin{bmatrix}\mathbf{v}_k^{(A)}\\ \mathbf{v}_k^{(B)}\end{bmatrix}\end{aligned} \tag{32}$$

Come prima si può calcolare la matrice della covarianza dell'errore di predizione $\mathbf{P}_{k+1|k}$.

$$\begin{aligned}\mathbf{P}_{k+1|k}^{1,1} &= \mathbf{Q}^{(A)} + (\lambda_k^{(A)})^2\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T + (\lambda_k^{(B)})^2\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T\\ &\quad + \lambda_k^{(A)}\lambda_k^{(B)}\mathbf{F}\mathbf{P}_{k|k}^{2,1}\mathbf{F}^T + \lambda_k^{(A)}\lambda_k^{(B)}\mathbf{F}\mathbf{P}_{k|k}^{1,2}\mathbf{F}^T\\ \mathbf{P}_{k+1|k}^{2,2} &= \mathbf{Q}^{(B)} + (\lambda_k^{(B)})^2\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T + (\lambda_k^{(A)})^2\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T\\ &\quad + \lambda_k^{(A)}\lambda_k^{(B)}\mathbf{F}\mathbf{P}_{k|k}^{2,1}\mathbf{F}^T + \lambda_k^{(A)}\lambda_k^{(B)}\mathbf{F}\mathbf{P}_{k|k}^{1,2}\mathbf{F}^T\\ \mathbf{P}_{k+1|k}^{1,2} &= \lambda_k^{(A)}\lambda_k^{(B)}\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T + \lambda_k^{(A)}\lambda_k^{(B)}\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T\\ &\quad + (\lambda_k^{(B)})^2\mathbf{F}\mathbf{P}_{k|k}^{2,1}\mathbf{F}^T + (\lambda_k^{(A)})^2\mathbf{F}\mathbf{P}_{k|k}^{1,2}\mathbf{F}^T\\ \mathbf{P}_{k+1|k}^{2,1} &= \lambda_k^{(A)}\lambda_k^{(B)}\mathbf{F}\mathbf{P}_{k|k}^{1,1}\mathbf{F}^T + \lambda_k^{(A)}\lambda_k^{(B)}\mathbf{F}\mathbf{P}_{k|k}^{2,2}\mathbf{F}^T\\ &\quad + (\lambda_k^{(A)})^2\mathbf{F}\mathbf{P}_{k|k}^{2,1}\mathbf{F}^T + (\lambda_k^{(B)})^2\mathbf{F}\mathbf{P}_{k|k}^{1,2}\mathbf{F}^T\end{aligned}$$

e per l'indipendenza tra le due telecamere si può scrivere

$$\begin{aligned}P_{k+1|k}^{1,1} &= \mathbf{Q}^{(A)} + (\lambda_k^{(A)})^2\mathbf{F}P_{k|k}^{1,1}\mathbf{F}^T + (\lambda_k^{(B)})^2\mathbf{F}P_{k|k}^{2,2}\mathbf{F}^T\\ P_{k+1|k}^{2,2} &= \mathbf{Q}^{(B)} + (\lambda_k^{(B)})^2\mathbf{F}P_{k|k}^{2,2}\mathbf{F}^T + (\lambda_k^{(A)})^2\mathbf{F}P_{k|k}^{1,1}\mathbf{F}^T\\ P_{k+1|k}^{1,2} &= P_{k+1|k}^{2,1} = 0\end{aligned}$$

Come prima si può separare il processo complessivo in due processi indipendenti

$$\begin{cases}\mathbf{x}_{k+1}^{(A)} = \lambda_k^{(A)}\mathbf{F}\mathbf{x}_k^{(A)} + \lambda_k^{(B)}\mathbf{F}\mathbf{x}_k^{(B)} + \mathbf{w}_k^{(A)}\\ \mathbf{z}_k^{(A)} = H\mathbf{x}_k^{(A)} + \mathbf{v}_k^{(A)}\end{cases} \tag{33}$$

e

$$\begin{cases}\mathbf{x}_{k+1}^{(B)} = \lambda_k^{(B)}\mathbf{F}\mathbf{x}_k^{(B)} + \lambda_k^{(A)}\mathbf{F}\mathbf{x}_k^{(A)} + \mathbf{w}_k^{(B)}\\ \mathbf{z}_k^{(A)} = H\mathbf{x}_k^{(B)} + \mathbf{v}_k^{(B)}\end{cases} \tag{34}$$

C. Tracking con N telecamere

Si può estendere facilmente il discorso riguardante due telecamere al caso di N telecamere. Il modello di sistema corrispondente al metodo che si basa sulla fusione delle predizioni diventa:

$$\begin{aligned}\begin{bmatrix}\mathbf{x}_{k+1}^{(1)}\\ \mathbf{x}_{k+1}^{(2)}\\ \vdots\\ \mathbf{x}_{k+1}^{(N)}\end{bmatrix} &= \begin{bmatrix}\lambda_k^{(1)}\mathbf{F}^{(1)} & \lambda_k^{(2)}\mathbf{F}^{(2)} & \dots & \lambda_k^{(N)}\mathbf{F}^{(N)}\\ \lambda_k^{(1)}\mathbf{F}^{(1)} & \lambda_k^{(2)}\mathbf{F}^{(2)} & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \lambda_k^{(1)}\mathbf{F}^{(1)} & \dots & \dots & \lambda_k^{(N)}\mathbf{F}^{(N)}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k^{(1)}\\ \mathbf{x}_k^{(2)}\\ \vdots\\ \mathbf{x}_k^{(N)}\end{bmatrix} + \begin{bmatrix}\mathbf{w}_k^{(1)}\\ \mathbf{w}_k^{(2)}\\ \vdots\\ \mathbf{w}_k^{(N)}\end{bmatrix}\\ \begin{bmatrix}\mathbf{z}_k^{(1)}\\ \mathbf{z}_k^{(2)}\\ \vdots\\ \mathbf{z}_k^{(N)}\end{bmatrix} &= \begin{bmatrix}\mathbf{H}^{(1)} & 0 & \dots & 0\\ 0 & \mathbf{H}^{(2)} & 0 & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ 0 & \dots & \dots & \mathbf{H}^{(N)}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k^{(1)}\\ \mathbf{x}_k^{(2)}\\ \vdots\\ \mathbf{x}_k^{(N)}\end{bmatrix} + \begin{bmatrix}\mathbf{v}_k^{(1)}\\ \mathbf{v}_k^{(2)}\\ \vdots\\ \mathbf{v}_k^{(N)}\end{bmatrix}\end{aligned} \tag{35}$$

e, ricordando che tutte le osservazioni sono indipendenti, con un'accettabile approssimazione si ottengono N modelli del tipo

$$\begin{cases}\mathbf{x}_{k+1}^i = \lambda_k^i\mathbf{F}\mathbf{x}_k^i + \sum_{j\neq i}\lambda_k^j\mathbf{F}\mathbf{x}_k^j + \mathbf{w}_k^i\\ \mathbf{z}_k^i = \mathbf{H}\mathbf{x}_k^i + \mathbf{v}_k^i\end{cases} \tag{36}$$

I pesi corrispondenti alle diverse osservazioni possono venir presi dipendenti dalle distanze utilizzando la (31) oppure possono essere presi uniformi.

Nello stesso modo si può applicare anche il secondo approccio ottenendo questa volta il modello

$$\begin{aligned}\begin{bmatrix}\mathbf{x}_{k+1}^{(1)}\\ \mathbf{x}_{k+1}^{(2)}\\ \vdots\\ \mathbf{x}_{k+1}^{(N)}\end{bmatrix} &= \begin{bmatrix}\mathbf{F}^{(1)} & 0 & \dots & 0\\ 0 & \mathbf{F}^{(2)} & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ 0 & \dots & \dots & \mathbf{F}^{(N)}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k^{(1)}\\ \mathbf{x}_k^{(2)}\\ \vdots\\ \mathbf{x}_k^{(N)}\end{bmatrix} + \begin{bmatrix}\mathbf{w}_k^{(1)}\\ \mathbf{w}_k^{(2)}\\ \vdots\\ \mathbf{w}_k^{(N)}\end{bmatrix}\\ \begin{bmatrix}\mathbf{z}_k^{(1)}\\ \mathbf{z}_k^{(2)}\\ \vdots\\ \mathbf{z}_k^{(N)}\end{bmatrix} &= \begin{bmatrix}\lambda_k^{(1)}\mathbf{H}^{(1)} & \lambda_k^{(2)}\mathbf{H}^{(2)} & \dots & \lambda_k^{(N)}\mathbf{H}^{(N)}\\ \lambda_k^{(1)}\mathbf{H}^{(1)} & \lambda_k^{(2)}\mathbf{H}^{(2)} & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \dots & \dots & \dots & \dots\\ \lambda_k^{(1)}\mathbf{H}^{(1)} & \dots & \dots & \lambda_k^{(N)}\mathbf{H}^{(N)}\end{bmatrix}\begin{bmatrix}\mathbf{x}_k^{(1)}\\ \mathbf{x}_k^{(2)}\\ \vdots\\ \mathbf{x}_k^{(N)}\end{bmatrix} + \begin{bmatrix}\mathbf{v}_k^{(1)}\\ \mathbf{v}_k^{(2)}\\ \vdots\\ \mathbf{v}_k^{(N)}\end{bmatrix}\end{aligned} \tag{37}$$

e da questo N modelli indipendenti del tipo

$$\begin{cases}\mathbf{x}_{k+1}^i = \mathbf{F}\mathbf{x}_k^i + \mathbf{w}_k^i\\ \mathbf{z}_k^i = \mathbf{H}\mathbf{x}_k^i + \sum_{j\neq i}\mathbf{H}\mathbf{x}_k^j + \mathbf{v}_k^i\end{cases} \tag{38}$$

V. Simulazioni

Per poter stabilire la bontà delle scelte fatte in ambito di modellizzazione virtuale della rete di telecamere e degli algoritmi di tracking considerati, è stata generata una traiettoria per la movimentazione del target che però si assumerà sconosciuta da parte del sistema di monitoraggio e, da tale traiettoria, è stato ricavato un filmato per simulare le immagini relative alla telecamera fissa. Mentre la rete di sensori è stata considerata fissa come struttura ma variabile come numero di elementi, la rete di telecamere è stata scelta variabile sia dal punto di vista del posizionamento che del numero delle stesse. Si tratta quindi, prima di iniziare le simulazioni, di fissare il numero di sensori coinvolti nella rete e di fissare la posizione della telecamera: si tratta cioè di stabilire le sue coordinate x_{tc}^w e y_{tc}^w nel sistema di riferimento mondo e l'altezza rispetto al piano di lavoro. A questo punto ciascuna telecamera avrà un suo campo visivo, dentro il quale dovrà seguire il target secondo gli algoritmi di tracking scelti.

Analizziamo meglio ciò che accade durante una simulazione: il target (nel caso particolare un triangolo bianco) inizia a muoversi da una posizione nel piano di lavoro che non è nota; inizialmente tutte le telecamere risultano in posizione di riposo, cioè inquadrano la zona sottostante a loro, mentre la rete di sensori è pronta a ricevere le informazioni. A meno che il target non si trovi subito all'interno di una immagine di una telecamera, non si possono utilizzare le informazioni relative ad esse; inizialmente quindi, nella maggior parte dei casi, si deve utilizzare la rete di sensori per stimare la posizione del target all'interno del piano di lavoro. Tale posizione viene stabilita con un errore che diminuisce all'aumentare del numero di sensori coinvolti nella rete per la srtuttura appunto della rete. Si ha infatti che la stima della posizione del target corrisponde alla posizione del sensore più vicino ad esso, cioè il sensore che segnala la presenza del target. In questo modo è possibile seguire il progressivo movimento del target anche se con delle posizioni nel campo di lavoro che risultano discrete. Durante la simulazione il target si muoverà in zone

che le telecamere possono inquadrare, cioè in zone in cui è possibile avere informazioni dalle telecamere riguardo la posizione dell'oggetto e poter passare al tracking del target. Si incontra a questo punto un problema di inizializzazione delle telecamere, si vuole cioè che le telecamere aggancino il prima possibile il target quando esso si trova dentro il campo visivo. Una volta che le telecamere hanno agganciato il target, il tracking viene effettuato con uno degli algoritmi sopra proposti. E' possibile che il target esca dal campo visivo della rete di telecamere, ma a questo punto si procede di nuovo all'inizializzazione delle telecamere fino alla prima acquisizione da parte di esse di informazioni relative al target, per il successivo tracking.

A. Stati delle telecamere

Prima di parlare in modo dettagliato di come si può procedere all'inizializzazione di ciascuna telecamera è conveniente discutere gli stati che essa può assumere nel corso della simulazione. Il criterio usato per stabilire lo stato della telecamera si basa sulla distanza del target, in posizione (x^w, y^w), dal centro del campo visivo; la distanza d è data da:

$$d = \sqrt{(x^w_{tc} - x^w)^2 + (y^w_{tc} - y^w)^2}.$$

Poichè è possibile considerare il campo visivo (FOV - in inglese *field of view*) come una circonferenza, viene naturale suddividere il piano di lavoro in zone d'azione circolari, come illustrato in figura 15. Si tratta ora quindi di decidere quali siano i valori più adeguati per i due raggi d'azione r e R; considerando che all'interno del campo visivo la telecamera deve inseguire adeguatamente il target, è stato scelto che il valore del primo raggio r sia uguale al valore del raggio del campo visivo della telecamera. Per quanto riguarda il secondo raggio d'azione R, abbiamo pensato che fosse necessario prendere in considerazione la struttura della rete di sensori. Poichè ciascun sensore copre una zona rettangolare di diagonale D, si è preso come secondo raggio d'azione:

$$R = r + D.$$

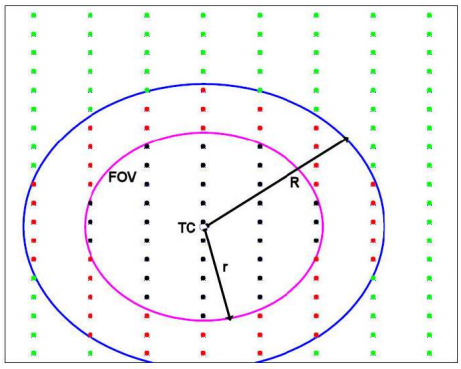

Figura 15. Raggi d'azione della telecamera.

Si possono a questo punto definire i quattro stati della telecamera che sono stati considerati: '*tracking*', '*alert*', '*standby*', '*nosigal*'.

- Lo stato '**tracking**' si ha nel caso in cui la distanza del target sia inferiore al raggio r e contemporaneamente ci sia acquisizione nell'immagine della telecamera del target o di parte di esso, con relativa informazione utile al tracking. Quando la telecamera si trova in questo stato si usa uno degli algoritmi considerati per inseguire il target nel suo movimento. Considerando la sola rete di sensori, i puntini neri in figura 15 indicano le possibili posizioni del target per cui la telecamera si trova in questo stato.
- Lo stato '**alert**' si ha quando la distanza del target risulta compresa tra i due raggi d'azione, la telecamera non riesce a inquadrare completamente il target o non lo inquadra affatto. Siamo nel caso in cui la telecamera deve procedere alla sua inizializzazione per evitare che il target entri nel campo visivo senza che la telecamera lo agganci nella sua immagine; può anche succedere che l'oggetto si stia allontanando dalla telecamera, cioè stia uscendo dal suo campo visivo, ma anche in questo caso si deve inizializzare la telecamera per evitare che il target torni subito dentro il campo visivo senza essere riagganciato nell'immagine. Considerando la sola rete di sensori, i puntini rossi in figura 15 indicano le possibili posizioni del target per cui la telecamera si trova in questo stato.
- Lo stato '**standby**' si verifica quando la distanza del target dal centro del campo visivo è maggiore del secondo raggio d'azione R. Il target risulta essere abbastanza distante, quindi la telecamera può rimanere a 'riposo', cioè può rimanere ferma inquadrando il centro del campo visivo. Considerando la sola rete di sensori, i puntini verdi in figura 15 indicano le possibili posizioni del target per cui la telecamera si trova in questo stato.
- Lo stato '**nosignal**' si verifica quando la distanza del target dal centro del campo visivo è tale per cui la telecamera dovrebbe essere in stato '*tracking*', ma l'acquisizione dell'immagine della telecamera non dà informazioni utili al tracking, cioè il target non è presente nell'immagine. Si verifica quindi un errore in uno dei livelli dell'inseguimento o si ha temporanea occlusione del target, pertanto si deve procedere di nuovo all'inizializzazione della telecamera attraverso la rete di sensori finchè il target non viene agganciato nuovamente.

B. Inizializzazione telecamere

Innanzitutto vogliamo definire cosa si intende con il termine 'inizializzazione' della telecamera: si tratta della movimentazione della telecamera prima di acquisire il target nell'immagine per permettere l'aggancio del target stesso quando entra nel campo visivo della telecamera. A questo scopo in base alla posizione del target nel piano di lavoro dovremmo scegliere la migliore zona del campo visivo da inquadrare per evitare di perdere l'aggancio del target; consideriamo inizialmente il caso in cui sia attiva la sola rete di sensori, si abbiano cioè informazioni solo da essa relative alla posizione del target. Possiamo quindi identificare la posizione del target con la posizione del sensore che ne sente la presenza; si avrà quindi:

$$\begin{bmatrix} x_{rf} \\ y_{rf} \end{bmatrix} = \begin{bmatrix} x^w \\ y^w \end{bmatrix}$$

Supponiamo inoltre che la telecamera sia in stato di alert, stato in cui si deve fare l'inizializzazione. Ci si trova quindi nella situazione schematizzata in figura 16.

La telecamera dovrà pertanto inquadrare la parte più esterna del campo visivo lungo la retta congiungente il centro del FOV con la posizione del sensore che ha comunicato la posizione del target. Dobbiamo quindi calcolare il punto centrale dell'inquadratura per poi muovere, attraverso la cinematica inversa, la telecamera. E' quindi necessario calcolare l'angolo α in figura 16, dato dalla seguente relazione:

$$\alpha = \arctan(\frac{Y}{X}) = \arctan(\frac{y_{rf} - y^w_{tc}}{x_{rf} - x^w_{tc}}).$$

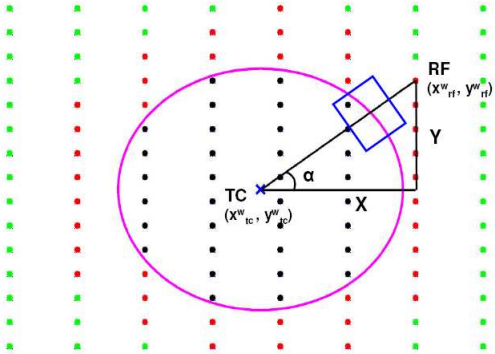

Figura 16. Inizializzazione della telecamera.

A questo punto conoscendo il raggio r del campo visivo è possibile calcolare le coordinate nel sistema di riferimento mondo del punto (x_{im}, y_{im}) che la telecamerà dovrà avere al centro della sua immagine:

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \begin{bmatrix} x^w_{tc} + r * \cos(\alpha) \\ y^w_{tc} + r * \sin(\alpha) \end{bmatrix}.$$

Questo tipo di calcolo e conseguente movimento dovrà essere fatto per ogni telecamera che si trova nello stato *'alert'*; è possibile che le informazioni relative alla posizione del target nel piano di lavoro arrivino dal tracking dello stesso ad opera di altre telecamere, questo caso specifico verrà trattato successivamente nella sezione dedicata all' *handover.*

Consideriamo ora un'altra inizializzazione sempre legata ad uno stato della telecamera, in questo caso lo stato *'nosignal'*; in tale situazione si ha che il target è all'interno del campo visivo della telecamera ma la sua visione è ostruita oppure c'è stato un errore di tracking. Si vuole che il target venga agganciato il prima possibile per continuare l'inseguimento, ma non avendo più le informazioni relative alla telecamera è necessario usare quelle che arrivano dalla rete di sensori. Si è deciso a questo punto di inquadrare al centro dell'immagine il sensore che ha comunicato la posizione del target, si pone cioè:

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \begin{bmatrix} x_{rf} \\ y_{rf} \end{bmatrix} = \begin{bmatrix} x^w \\ y^w \end{bmatrix}.$$

C. Handover

Per *handover* si intende la cooperazione tra due o più telecamere per continuare a inseguire il target senza perderlo mai di vista; è quindi necessario che vi sia un istante in cui le due telecamere possano vedere contemporaneamente il target, i due campi visivi devono pertanto sovrapporsi (*overlapping*); un esempio è mostrato in figura 17.

L' *handover* si verifica quando il target in movimento si sta dirigendo ai margini del campo visivo di una telecamera e, al tempo stesso, si sta avvicinando al campo visivo di un'altra telecamera, che è quindi in grado di continuare il tracing iniziato dalla prima. Mentre la telecamera che sta eseguendo il tracking continuerà il suo lavoro finchè le è possibile, è necessario inizializzare la telecamera che dovrà agganciare il target; per tale inizializzazione si ricorre ad un metodo

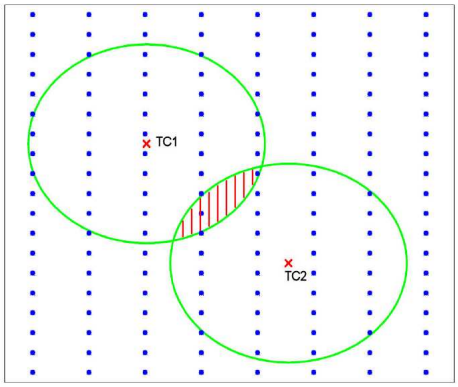

Figura 17. Overlapping

molto simile a quello visto in precedenza, con la sola variante che, in questo caso, la posizione del target è più accurata e arriva da una (o più) telecamere. Si dovrà quindi considerare:

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \begin{bmatrix} x^w \\ y^w \end{bmatrix}.$$

Dal momento che il tracking è già in funzione, anche se dovuto a un'altra telecamera, è possibile prendere in considerazione la predizione dovuta all'algoritmo scelto in modo da anticipare l'aggancio; si ha quindi:

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \begin{bmatrix} x^w_{pred} \\ y^w_{pred} \end{bmatrix}.$$

Nella zona del campo visivo in cui entrambe le telecamere inseguono il target, il contributo al tracking è dato da tutte e due le telecamere come spiegato nella sezione dedicata al *sensor fusion*. Si ha quindi che in tale zona entrambe rimangono agganciate al target finchè esso non esce definitivamente dal relativo campo visivo. Il passaggio da una telecamera ad un'altra risulta quindi naturale e senza comandi diretti, poichè è stato imposto che tutte le telecamere che sono in grado di osservare il target e di inseguirlo, lo devono fare.

D. Risultati

Abbiamo simulato il comportamento di una rete di telecamere formata da due o tre elementi, per verificare gli algoritmi precedentemente esposti, nell'inseguimento del target. Si è supposto che il target fosse già agganciato da almeno una telecamera, cioè i grafici si riferiscono ad una parte dell'intera simulazione da cui si possono ricavare delle utili considerazioni.

Iniziamo considerando il comportamento del classico filtro di Kalman (centralizzato) con pesi uniformi e con pesi dipendenti dalle distanze utilizzando una rete formata da tre telecamere.

Ciò che si nota subito in figura 18 è la sostanziale uniformità di comportamento, non si hanno quindi vantaggi o svantaggi nell'utilizzare un metodo per stabilire i pesi piuttosto che l'altro; l'errore di predizione è sostanzialmente uguale nei due casi. L'unica piccola differenza si osserva nel momento in cui anche la terza telecamera aggancia il target (*handover*), ottavo o nono campione del grafico. Una spiegazione può essere dovuta al fatto che la posizione iniziale della telecamera che dovrà agganciare il target è migliore nel filtro centralizzato con i pesi dipendenti dalle distanze, nel caso specifico. Infatti se si decide di confrontare il comporamento del filtro centralizzato in presenza di due telecamere si nota che tale differenza non sussiste più, come si può notare osservando le figure

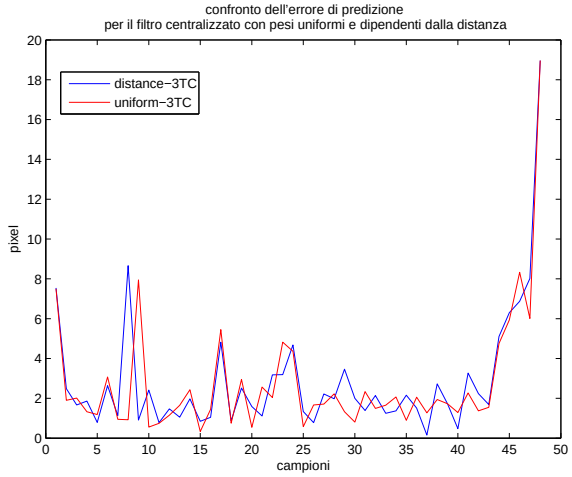

Figura 18. Confronto del filtro di Kalman centralizzato con pesi uniformi e dipendenti dalle distanze

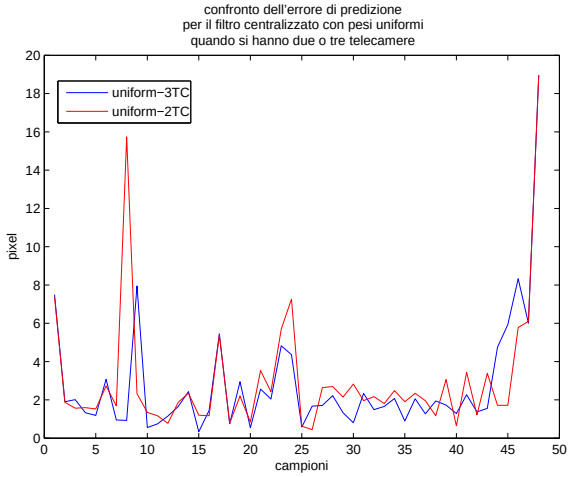

Figura 20. Confronto del filtro di Kalman centralizzato con pesi uniformi per due e tre telecamere

19 e 20. Sempre da queste ultime due figure però è possibile ossevare che, durante l'aggancio da parte di una telecamera del target, l'errore di predizione relativo alla rete composta da tre telecamere è minore rispetto alla rete formata da due elementi, si riducono infatti gli errori dovuti alla fase di acquisizione parziale dell'obbiettivo grazie ad una predizione più robusta. Tuttavia bisogna anche dire che la differenza tra i due errori non è molto elevata e in simulazione impercettibile durante la visione.

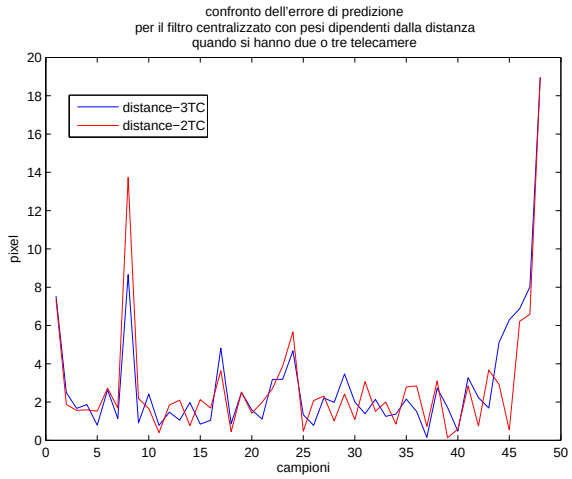

Figura 19. Confronto del filtro di Kalman centralizzato con pesi dipendenti dalle distanze per due e tre telecamere

Si può osservare come il passaggio da una rete a due telecamere a tre telecamere, per il caso dei pesi dipendenti dalle distanze, non fornisce miglioramenti nell'errore di predizione poichè si continuano ad avere errori pressochè simili e che oscillano attorno ad uno stesso valore. Nel caso invece di pesi uniformi la rete a tre telecamere migliora leggermente gli errori di predizione che oscillano attorno ad un valore più basso.

Passiamo ora a considerare il filtro di Kalman distribuito con pesi uniformi e dipendenti dalle distanze. Prima di passare al confronto diretto del comportamento con pesi differenti dei filtri distribuiti, analizziamo il comportamento dei singoli filtri con lo stesso criterio per i pesi, per una rete di tre telecamere. Anche in questo caso è semplice notare in figura 21 e 22 come gli errori di predizione abbiano un picco nel momento in cui la terza telecamera aggancia anche essa il target. Si vede invece da entrambe le figure come il picco relativo alla perdita di informazioni dalla telecamera 2 (ventiquattresimo campione) influisca meno nell'errore di predizione; questo perchè la sua predizione dipende, anche dopo aver perso il target, dalle immagini che arrivano dalle restanti telecamere attive nel tracking. Ulteriore osservazione va fatta sull'errore di predizione della telecamera 2 dopo aver perso il target, in quanto mancando ad essa l'informazione retroattiva della sua immagine non riesce a ridurre l'errore di predizione (come fanno invece le altre telecamere) dovuto al filtro di Kalman tempo variante che continua a dare la predizione ad ogni passo come fosse in catena aperta, pure ottenendo informazioni dalle altre telecamere.

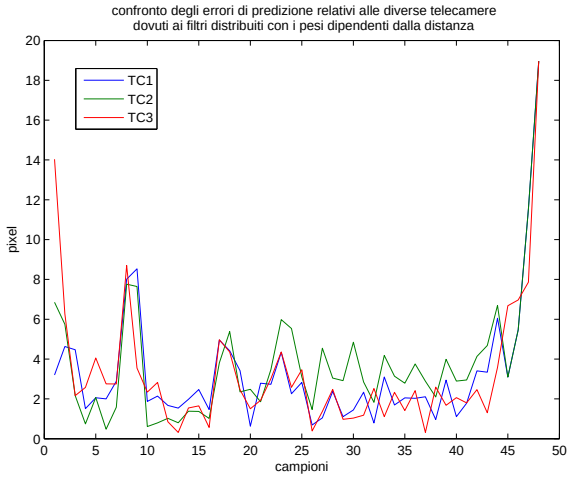

Figura 21. Confronto del filtro di Kalman distribuito con pesi dipendenti dalle distanze per ciascuna telecamera

Passiamo ora a considerare il confronto tra diversi tipi di pesi al livello della stessa telecamera. Poichè il comportamento è simile per tutte e tre le telecamere ne prendiamo in esame una sola, premettendo che le considerazioni saranno di validità generale. Come

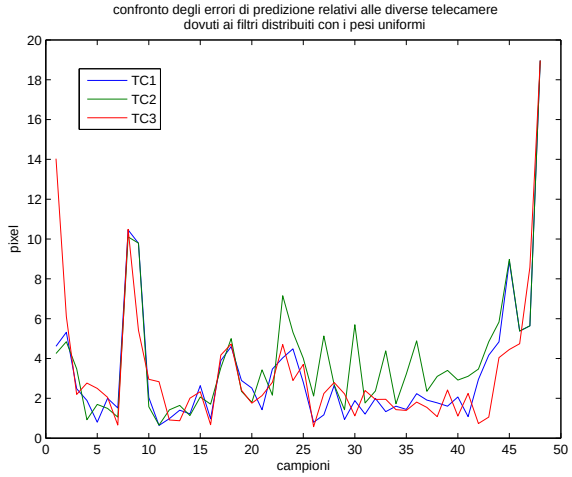

Figura 22. Confronto del filtro di Kalman distribuito con pesi uniformi per ciscuna telecamera

si osserva in figura 23, l'unica vera differenza apprezzabile si ha nel momento dell'*handover*, quando la terza telecamera aggancia il bersaglio parzialmente, ottavo campione. Questo è l'unico momento della simulazione in cui si può dichiarare che il comportamento del filtro distribuito con pesi dipendenti dalle distanze funziona meglio di quello con i pesi uniformi, tenendo sempre conto che il miglioramento effettivo nell'errore di predizione è di circa due pixel. Per il resto dela simulazione gli errori di predizione continuano ad oscillare e sovrapporsi non fornendo motivi di rilievo per preferire una scelta rispetto all'altra.

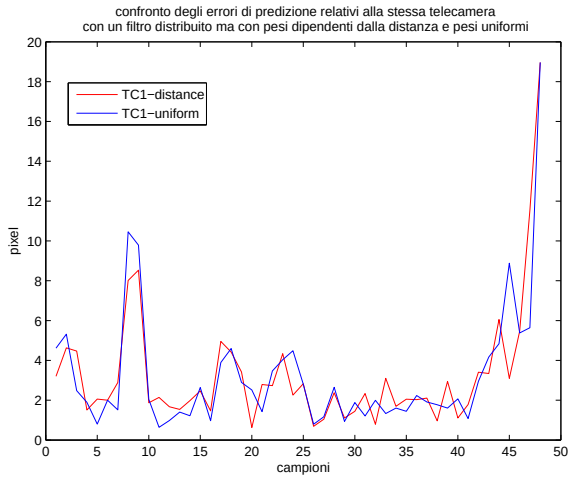

Figura 23. Confronto del filtro di Kalman distribuito con pesi uniformi e dipendenti dalle distanze per la telecamera 1

La cosa che si nota osservando tutte le figure riportate è che, quando tutte le telecamere perdono le informazoni provenienti dalle loro immagini e si passa quindi alla rete di sensori, gli errori di predizione aumentano molto velocemente, fatto che era stato già mostrato nella sezione IV in figura 14.

VI. Conclusioni e Sviluppi futuri

Come si evince dalla sezione V-D, gli errori di predizione rispetto alla traiettoria reale, in pixel, sono molto bassi (circa 7-8 in media) rispetto alle centinaia di pixel presenti in una immagine delle telecamere virtuali. Possiamo quindi affermare che tutti gli algoritmi considerati localizzano e inseguono in modo efficente il target dal punto di vista simulativo; rimane a questo punto aperto il discorso relativo alla prova sperimentale degli algoritmi, in base all'ambiente di lavoro considerato, che può essere senza dubbio uno degli aspetti futuri da sviluppare di questo progetto in quanto per via simulativa le osservazioni relative al filtro di Kalman sono state considerate quasi perfette, mentre le rilevazioni in laboratorio potrebbero presentare un rumore più elevato. Inoltre abbiamo considerato la rete di sensori sincrona a quella delle telecamere e sempre funzionante, mentre in fase sperimentale si possono avere problematiche relative a ritardi nelle informazioni o anche a perdite di pacchetto. Sarebbe quindi opportuno poter testare la bontà degli algoritmi considerati in situazioni reali, come può essere appunto la struttura presente in laboratorio e da cui abbiamo preso spunto per la modellizzazione dell'ambiente di lavoro virtuale.

Dal punto di vista degli sviluppi futuri in ambito pratico, una forte richiesta di questo tipo di sorveglianza è ultimamente venuta alla luce nelle *Art Gallery*. In questo settore viene richiesta una copertura totale degli spazi occupati dalle opere d'arte in modo da avere sempre a disposizione delle visuali utili per la sorveglianza; particolare aspetto da sviluppare in questo campo riguarda l'opportuna disposizione della rete di telecamere per assolvere il compito richiesto, conoscendo la struttura dello spazio di lavoro e minimizzando il numero di telecamere necessarie.

Diverso è il discorso relativo all'autocalibrazione delle telecamere, altro importante aspetto che si può sviluppare; in particolare sarebbe interessante cercare di utilizzare il percorso del target come oggetto tridimensionale, cioè fornito di tre componenti, per raffinare i parametri necessari per la calibrazione delle telecamere (nel nostro caso in particolare l'oggetto è da considerarsi bidimensionale perchè è stato considerato su un piano di lavoro, trascurando la profondità). Aspetto necessario da considerare è l presenza di molte regioni del campo visivo delle telecamere che si sovrappongono (overlapping), in modo da avere diverse informazioni da più telecamere.

Un ultimo problema da poter considerare riguarda il tracking di un oggetto in movimento con contemporanea azione di zoom su di esso. In questo caso si tratta di coordinare adeguatamente le due operazioni simultanee, usando correttamente le informazioni che giungono dalle telecamere.

RIFERIMENTI BIBLIOGRAFICI

[1] Trucco E., Verri A., 1998, *Introductory Techniques for 3-D Computer Vision*, Prentice-Hall.

[2] Fusiello A., 2006, *Visione Computazionale: appunti delle lezioni*.

[3] Bar-Shalom Y., Fortmann T. E., 1988, *Tracking and data Association*, Academic Press.

[4] Picci G., *Filtraggio statistico (Wiener, Levinson, Kalman) e applicazioni*, Edizioni Progetto Padova.

[5] Söderström T., Stoica P., 2001, *System Identification*, Prentice Hall.

[6] Arulampalam, M.S., Maskell, S., Gordon, N. & Clapp, T., 2002, *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*, Signal Processing, IEEE Transactions on, vol. 50, no. 2, pp. 174-188.

[7] Bar-Shalom, Y., 2002, *Update with out-of-sequence measurements in tracking: exact solution*, Aerospace and Electronic Systems, IEEE Transactions on, vol. 38, no. 3, pp. 769-777.

[8] Detmold, H., van den Hengel, Anton, Dick, A., Cichowski, A., Hill, R., Kocadag, E., Falkner, K. & Munro, D.S., 2007, *Topology Estimation for Thousand-Camera Surveillance Networks*, Distributed Smart Cameras, 2007. ICDSC '07. First ACM/IEEE International Conference on, pp. 195.

[9] Liu, Z., Wang, X. & PaIaniswami, M., 2005, *Improving tracking accuracy using information of dissimilar sensors*, Intelligent Sensing and Information Processing. Proceedings of 2005 International Conference on, pp. 94.

[10] Miyaki, T., Yamasaki, T. & Aizawa, K., 2007, *Multi-Sensor Fusion Tracking Using Visual Information and WI-Fi Location Estimation*, Distributed Smart Cameras, 2007. ICDSC '07. First ACM/IEEE International Conference on, pp. 275.

[11] Davis, J., Chen, X., 2003, *Calibrating pan-tilt cameras in wide-area surveillance networks*, Proceedings of the Ninth IEEE International Conference on Computer Visio, ICCV '03.

[12] Nakamura, E., Loureiro, A., Frery, A., 2007, *Information Fusion for Wireless Sensor Networks: Methods, Models and Classification*, ACM Computing Surveys, Vol. 39, No. 3, Article 9, Publication date: August 2007.

[13] Becker, Jan C., 2000, *Fusion of Heterogeneous Sensors for the Guidance of an Autonomous Vehicle*.

[14] Godsill, Simon J., Vermaak, Jako, William, Ng, & Li, Jack F., 2007, *Models and Algorithms for Tracking of Maneuvering Objects Using Variable Rate Particle Filters*, Vol. 95, IEEE No. 5, May 2007, Proceedings of the IEEE

[15] DU, Wei, Piater, Justus, *Data Fusion by Belief Propagation for Multi-Camera Tracking*

[16] Kang, Sangkyu, Paik, Joonki, Abidi, Besma R., Shelton, David, Mitckes, Mark , & Abidi, Mongi A., 2001, *Gate-to-gate automated video tracking and location*, The FAA's 3rd International Aviation Security Tecnology Symposyum, Atlantic City, NJ, November 2001.

[17] Kang, S.,Koschan, A., Abidi, B., & Abidi, M., 2004, *Video Surveillance of High Security Facilities*, Proc. of 10th Int. Conf. on Robotics & Remote Systems for Hazardous Environments, pp. 530-536, Gainesville, FL, March 2004.

[18] Yao, Y., Abidi, B., & Abidi, M., 2006, *Fusion of Omnidirectional and PTZ Cameras for Accurate Cooperative Tracking*, Proc. of IEEE International Conference on Video and Signal Based Surveillance (AVSS'06).