

Localizzazione e tracking tramite “sensor fusion”



De Rocchi, M.



Manente, A.



Mucciardi, A.

Paccagnella, G.

Padova, 3 Aprile 2008

Obiettivi e step di progetto

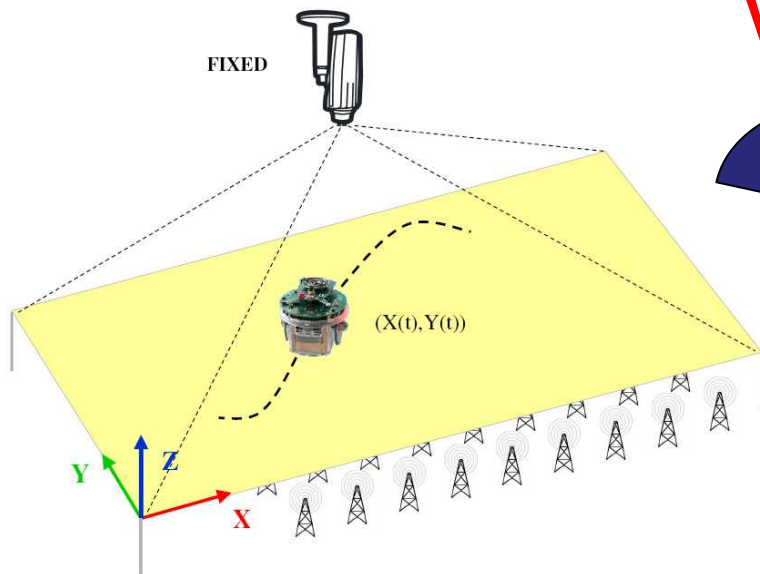
OBIETTIVO PRINCIPALE:

- **Messa a punto di un sistema multicamera per il visual tracking**

STEP DI PROGETTO:

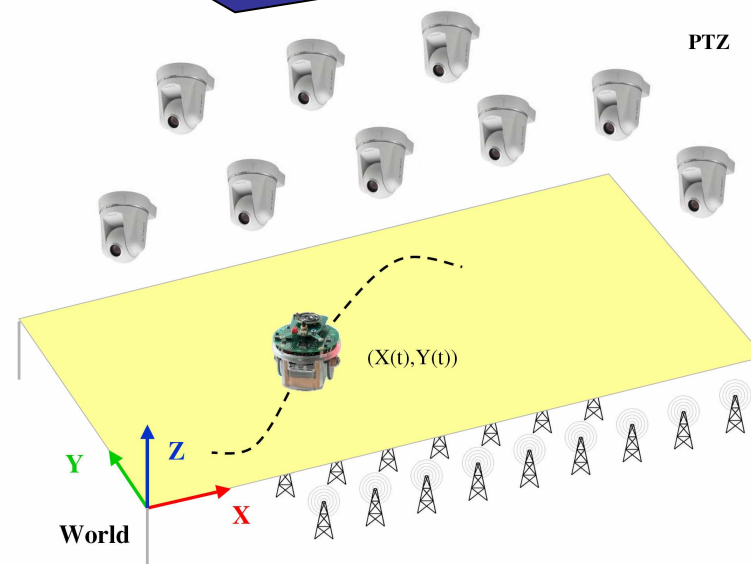
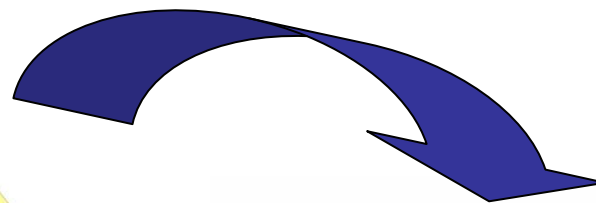
- **Design del testbed virtuale per le simulazioni**
- **Analisi e implementazione degli algoritmi di tracking**
- **Analisi e implementazione degli algoritmi di “sensor fusion”**
- **Simulazioni**
- **Risultati e sviluppi futuri**

VIRTUALIZZAZIONE



SISTEMA REALE

Una telecamera fissa



SISTEMA VIRTUALE

N telecamere pan-tilt-zoom (PTZ)

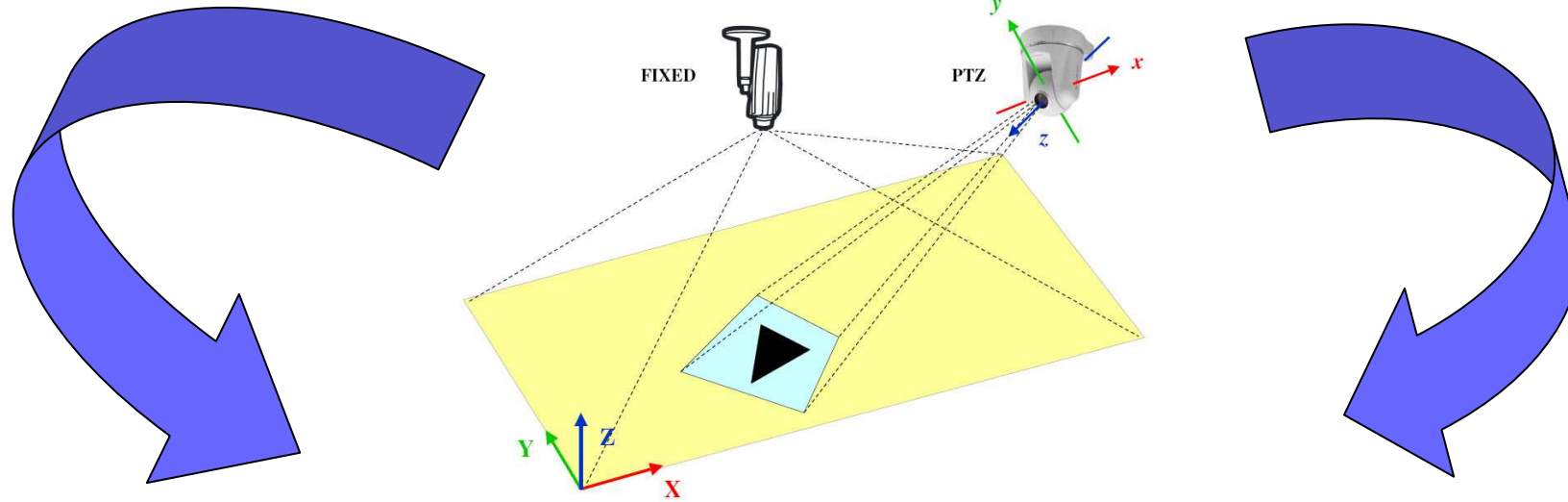


Immagine dalla
telecamera fissa

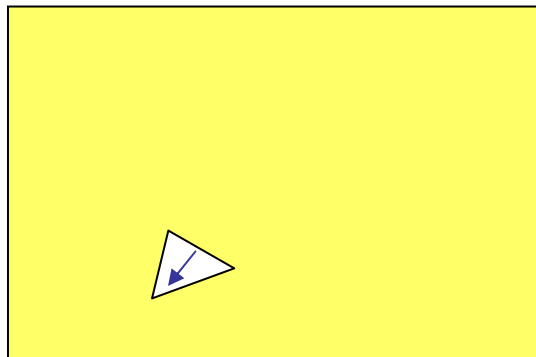
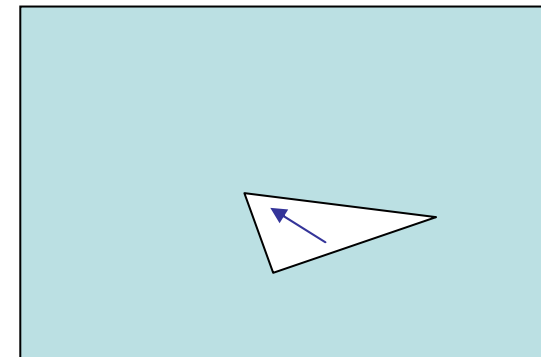
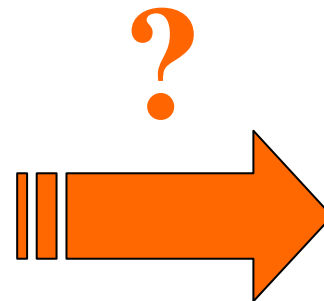


Immagine dalla
telecamera PTZ



RETTIFICAZIONE ORTOGONALE

[Liebowitz e Zisserman (1998)]

Sintesi di un'immagine da una vista fronto-parallela a un piano
a partire da un'immagine prospettica dello stesso piano

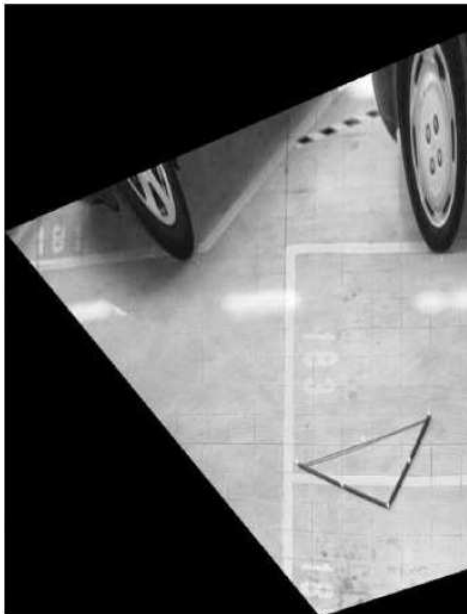


IMMAGINE DA UNA VISTA
FRONTO-PARALLELA

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



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

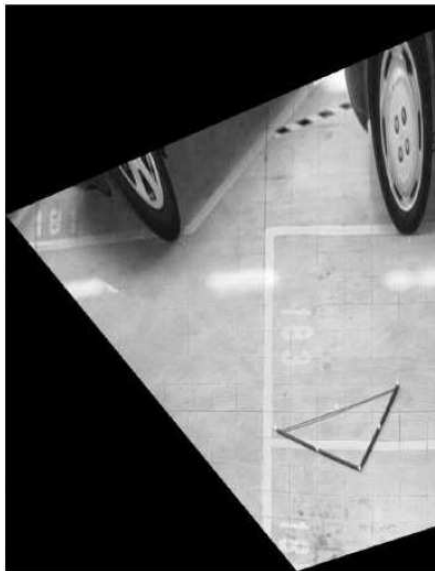


IMMAGINE DA UNA VISTA
PROSPETTICA

Rettificazione ortogonale inversa

RETTIFICAZIONE ORTOGONALE INVERSA

Sintesi di un'immagine da una vista prospettica di un piano a partire da un'immagine fronto-parallela dello stesso piano



?



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

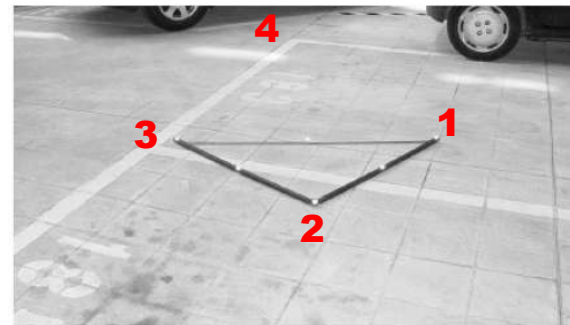
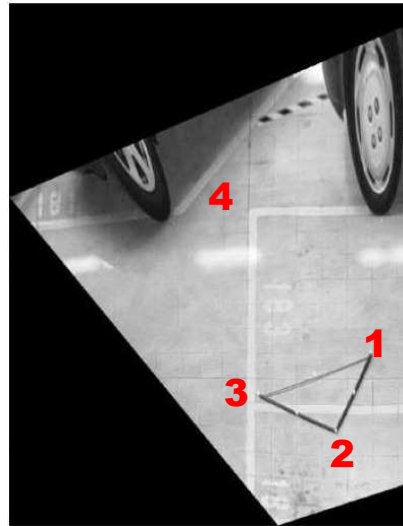
La matrice **H** codifica una trasformazione lineare non singolare del piano proiettivo in sé stesso detta **OMOGRAFIA** o **COLLINEAZIONE**

Problema: come si ricava **H** ?

Determinazione dell'omografia

H è definita a meno di un fattore di scala (ha 8 gradi di libertà)

Per definire univocamente un'omografia (a meno di un fattore di scala) è necessario individuare almeno **quattro corrispondenze di punti** (purchè almeno tre di questi non siano collineari)

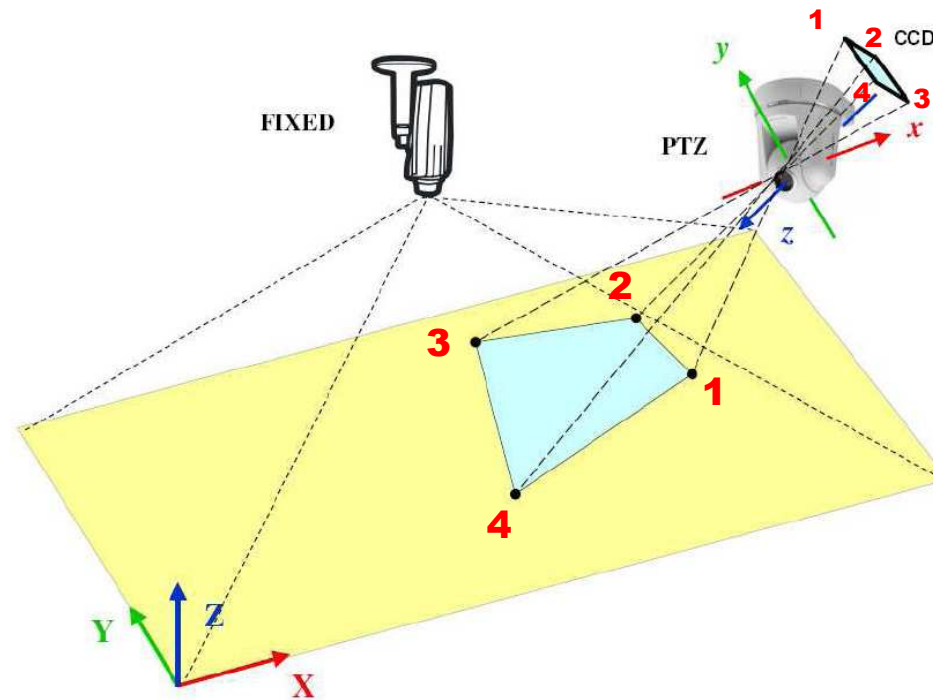


Problema: il metodo delle corrispondenze è applicabile solo se si hanno a disposizione le due immagini, nel nostro caso questo non accade

Soluzione: usiamo la geometria del sistema!

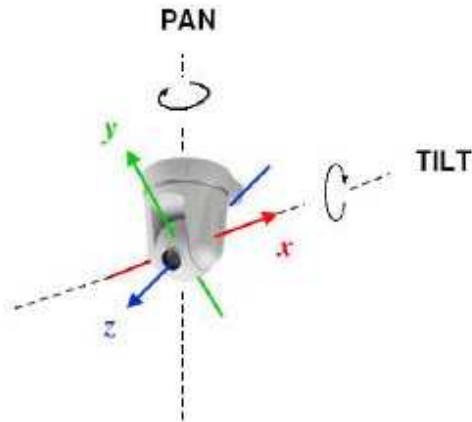
Geometria proiettiva

Come **quattro corrispondenze di punti** possiamo prendere gli **angoli del CCD e le loro proiezioni sul piano** di lavoro attraverso il centro ottico della telecamera PTZ



Per far questo è necessario avere una descrizione completa del **modello della telecamere PTZ**

Modello di telecamera PTZ



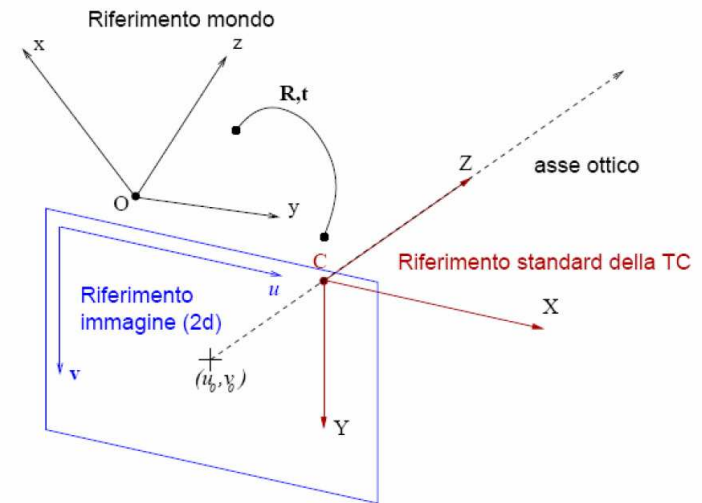
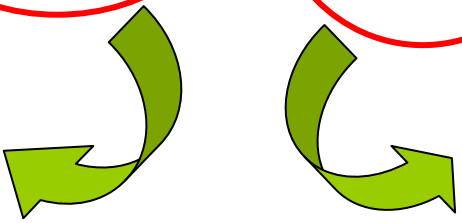
Consideriamo **noti i parametri intrinseci** (lunghezza focale, dimensione efficace del pixel) e quelli **estrinseci** (posizione telecamera nel sdr mondo)

Equazione di proiezione prospettica

$$k \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{1,1} & R_{1,2} & t_1 \\ R_{2,1} & R_{2,2} & t_2 \\ R_{3,1} & R_{3,2} & t_3 \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix}$$

parametri intrinseci

parametri estrinseci



Dal confronto tra le seguenti formule

$$k \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{1,1} & R_{1,2} & t_1 \\ R_{2,1} & R_{2,2} & t_2 \\ R_{3,1} & R_{3,2} & t_3 \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ 1 \end{bmatrix}$$

coordinate
immagine

coordinate
mondo

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

otteniamo

$$\begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} \\ H_{2,1} & H_{2,2} & H_{2,3} \\ H_{3,1} & H_{3,2} & H_{3,3} \end{bmatrix}^{-1} = \begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{1,1} & R_{1,2} & t_1 \\ R_{2,1} & R_{2,2} & t_2 \\ R_{3,1} & R_{3,2} & t_3 \end{bmatrix}$$

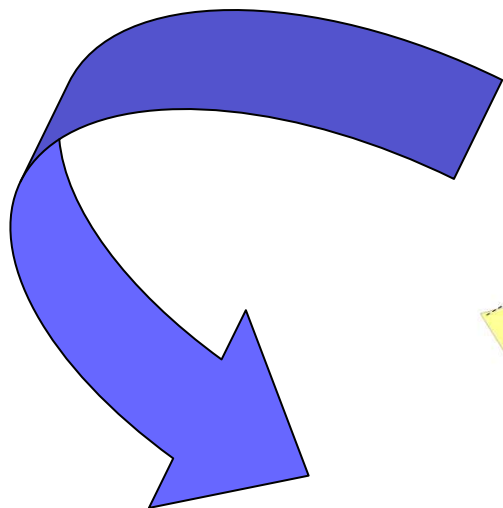


Immagine dalla
telecamera fissa

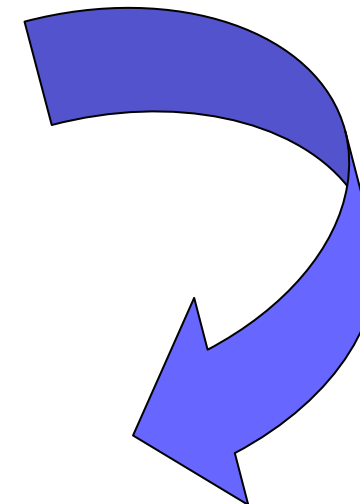
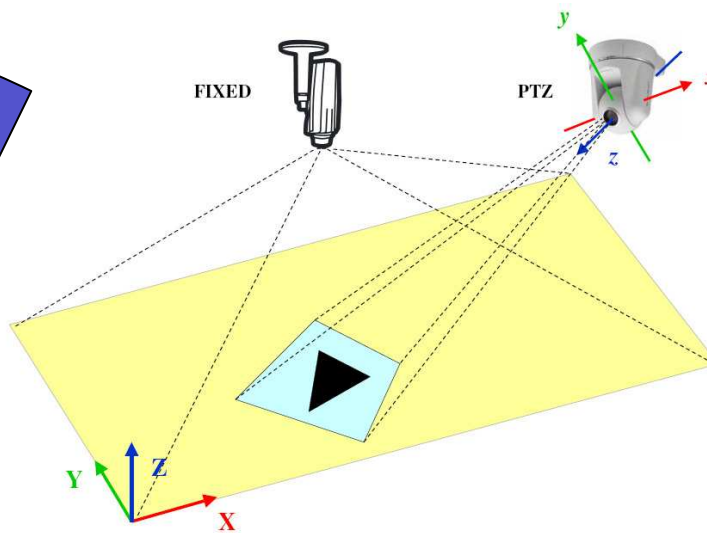
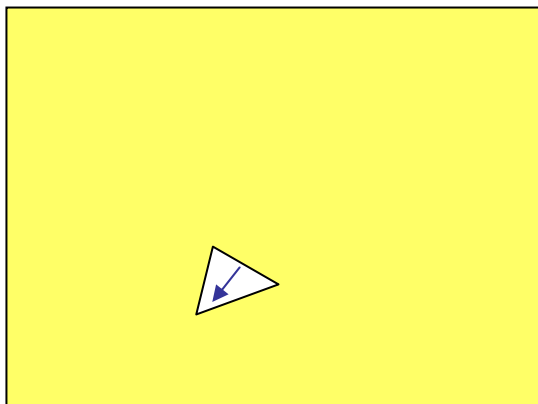
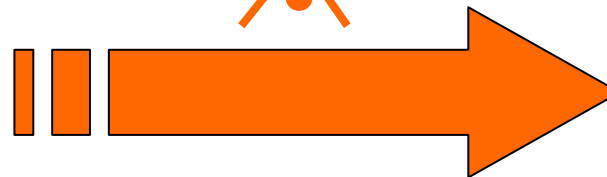
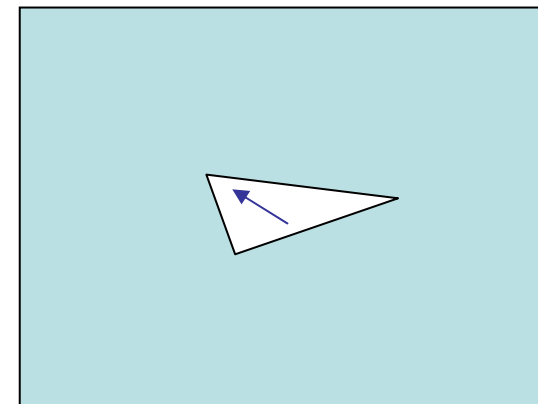


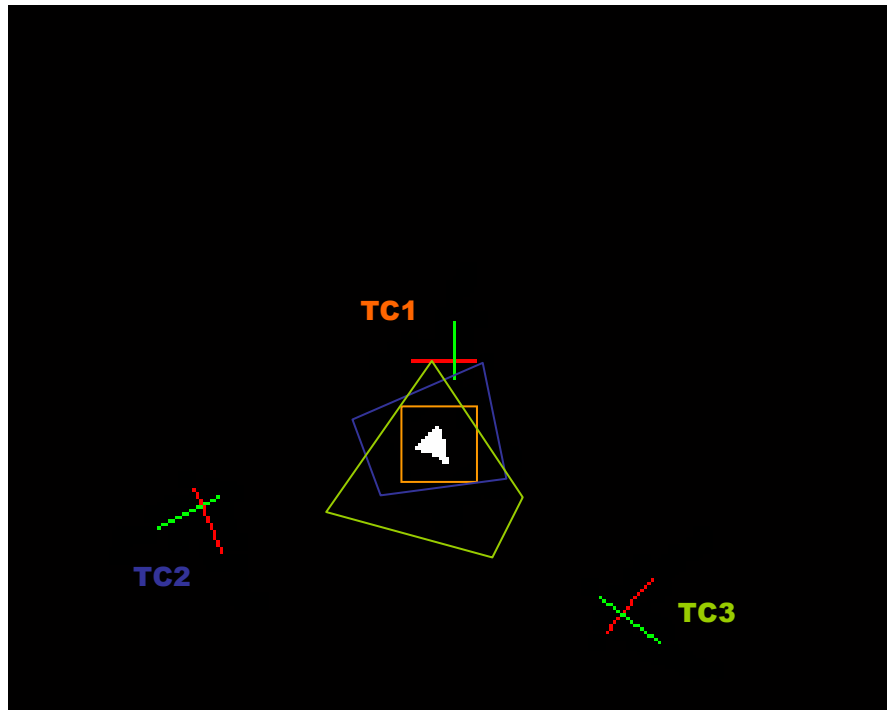
Immagine dalla
telecamera PTZ



$$\begin{bmatrix} -fk_u & 0 & u_0 \\ 0 & -fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{1,1} & R_{1,2} & t_1 \\ R_{2,1} & R_{2,2} & t_2 \\ R_{3,1} & R_{3,2} & t_3 \end{bmatrix}$$



Esempio



TC1



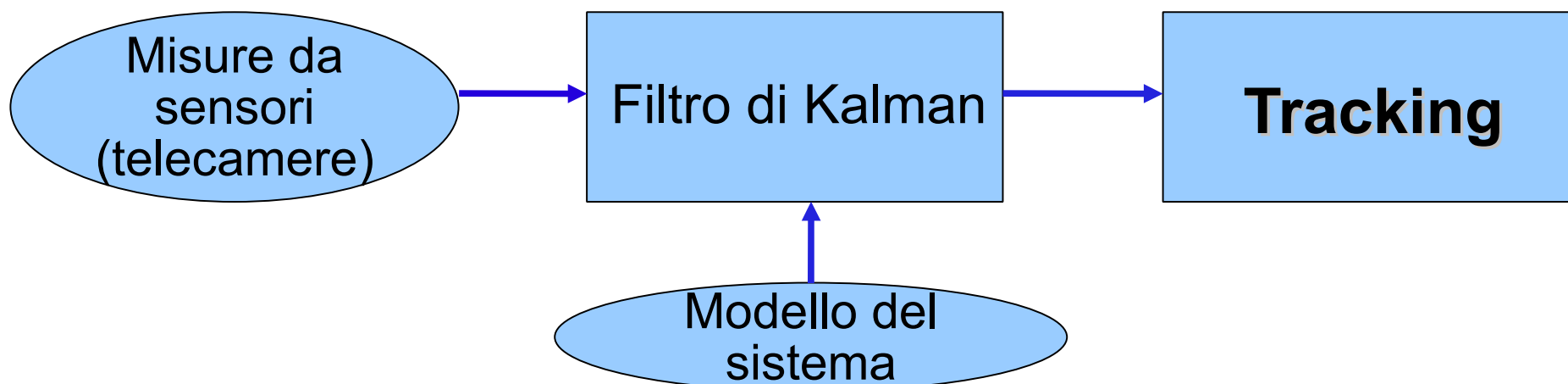
TC2



TC3

Tracking

- La soluzione più immediata per problemi di tracking e navigazione automatica è rappresentata dal filtro di Kalman, tra le cui prime applicazioni ci fu proprio il problema di stima delle traiettorie nel progetto Apollo della NASA.
- Differenza tra tracking e navigazione automatica è che il primo viene fatto dal dispositivo di misura senza conoscere le forze agenti sul veicolo, mentre il secondo viene fatto a bordo del veicolo di cui viene ricostruita la traiettoria.



Equazioni del filtro di Kalman:

Aggiornamento:

$$\hat{\mathbf{x}}(t/t) = \hat{\mathbf{x}}(t/t-1) + \mathbf{K} [\mathbf{z}(t) - \mathbf{C} \hat{\mathbf{x}}(t/t-1)], \quad t > t_0$$

Predizione:

$$\hat{\mathbf{x}}(t+1/t) = \mathbf{F} \hat{\mathbf{x}}(t/t), \quad t > t_0$$

Inizializzazione con il metodo “two steps”

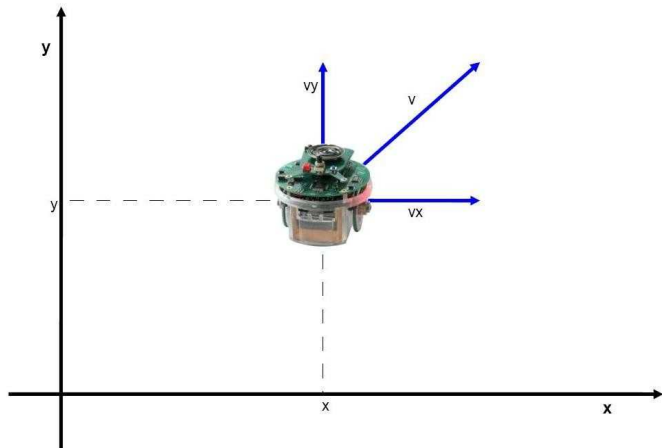
Posizione:

$$\hat{\mathbf{x}}(1/1) = \mathbf{z}(1)$$

Velocità:

$$\hat{\dot{\mathbf{x}}}(1/1) = \frac{\mathbf{z}(1) - \mathbf{z}(0)}{T}$$

Modello a velocità costante



Vettore di stato

$$\mathbf{x} = (x \dot{x} y \dot{y})$$

Modello cinematico lineare

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

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ f_x \\ 0 \\ f_y \end{bmatrix}$$

$$\mathbf{F} = e^{\mathbf{A}T}$$



Discretizzazione

Aggiornamento di stato:

$$\mathbf{x}(t+1) = \mathbf{F} \mathbf{x}(t) + \mathbf{w}(t)$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_x & 0 \\ 0 & \mathbf{F}_y \end{bmatrix}; \quad \mathbf{F}_x = \mathbf{F}_y = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

Osservazioni:

$$\mathbf{z}(t) = \mathbf{C} \mathbf{x}(t) + \mathbf{v}(t)$$

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



Maneuvering target

$$\mathbf{x}(t+1) = \mathbf{F} \mathbf{x}(t) + \mathbf{G} \mathbf{u}(t) + \mathbf{v}(t)$$

A. Ingresso modellato
come processo casuale

1. Rumore
bianco

2. Rumore di
Markov
autocorrelato

1. La manovra si verifica
tra due istanti di
campionamento, con
caratteristica istantanea.

B. Ingresso stimato in
tempo reale

1. Input
Estimation

2. Variable
dimension filter

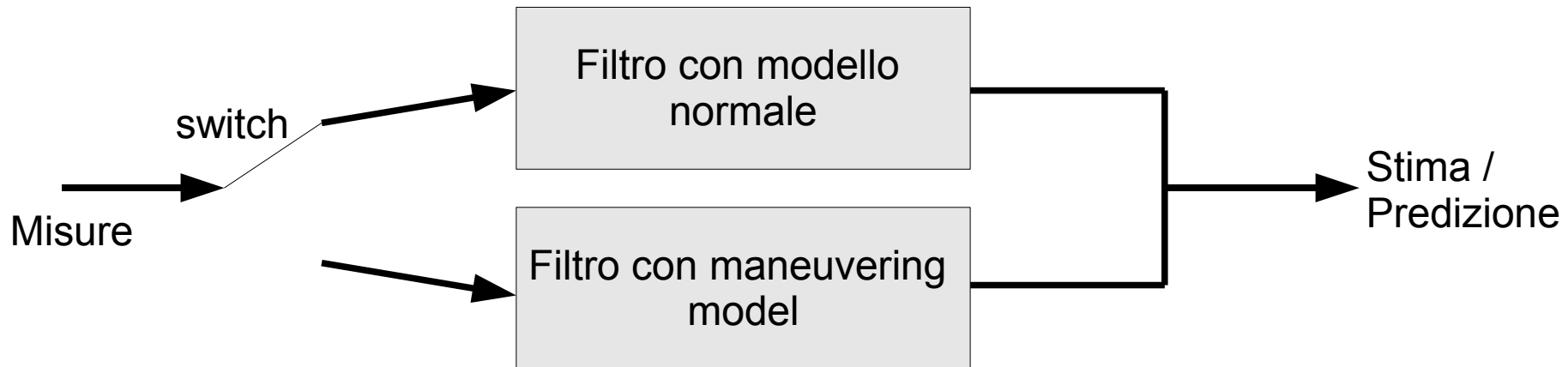
2. Misure frequenti
durante la manovra,
tali cioè da mantenere
l'inseguimento durante
questa.



Variable Dimension Filter

Concetto fondante: la manovra viene considerata come parte inerente alla dinamica del target e pertanto viene compresa nello stato.

Si utilizza quindi un nuovo modello, alternato a quello normale, che viene attivato solo quando è presente una manovra sul veicolo.



Maneuvering model = **modello ad accelerazione costante**

$$\mathbf{x} = (x \dot{x} y \dot{y} \ddot{x} \ddot{y})$$



Variable dimension filter

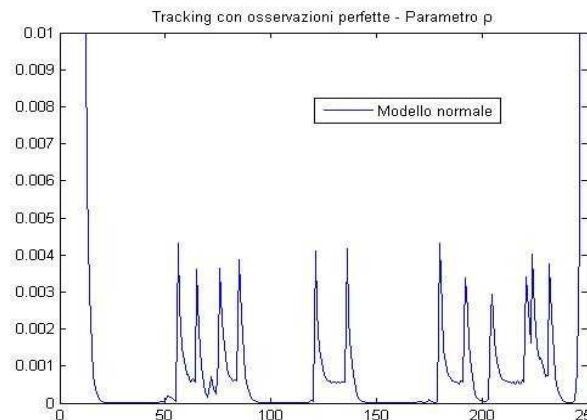
Per decidere quando è attiva una manovra sul veicolo e quindi quando si vuole passare a utilizzare il maneuvering model si utilizza un parametro a memoria finita:

$$\rho(t) = \alpha \rho(t-1) + v'(t) S^{-1} v(t);$$

con $0 < \alpha < 1$ arbitrario, $v(t)$ è l'innovazione ($y(t) - C x(t|t-1)$) e S la sua varianza.

Da α dipende la memoria del parametro; infatti $s = 1 / (1-\alpha)$ può essere considerato come la memoria effettiva su cui viene testata la presenza di rumore.

Quando il parametro $\rho(t)$ supera una determinata soglia si deduce che è attiva una manovra e dall'istante successivo si utilizza il maneuvering model.





Inizializzazione Maneuvering Model

→ Inizializzazione dell'**accelerazione** all'istante t-s:

$$\hat{x}_{4+i}^m(t-s/t-s) = \frac{2}{T^2} [y_i(t-s) - \hat{y}_i(t-s/t-s-1)], \quad i=1,2$$

→ Le componenti di stima della **posizione** (dalle misure corrispondenti):

$$\hat{x}_{2i-1}^m(t-s/t-s) = y_i(t-s), \quad i=1,2$$

→ Le componenti di **velocità** vengono corrette dalla stima dell'accelerazione:

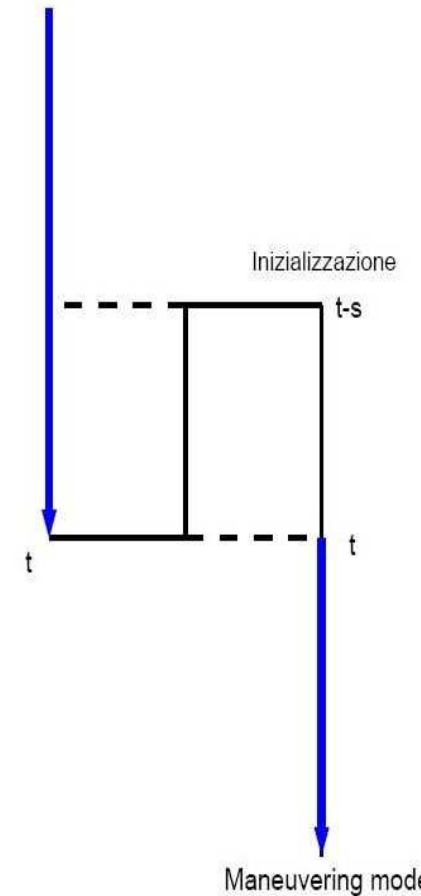
$$\hat{x}_{2i}^m(t-s/t-s) = \hat{x}_{2i}(t-s/t-s-1) + T \hat{x}_{4+i}^m(t-s/t-s), \quad i=1,2$$

E' necessario poi stabilire un criterio che indichi quando finisce la manovra per tornare ad utilizzare il modello normale. Per questo si considera che quando la manovra termina l'accelerazione torna nulla, quindi si considera il parametro:

$$\sigma_a(t) = \hat{a}'(t/t) P_a^m(t/t)^{-1} \hat{a}(t/t)$$

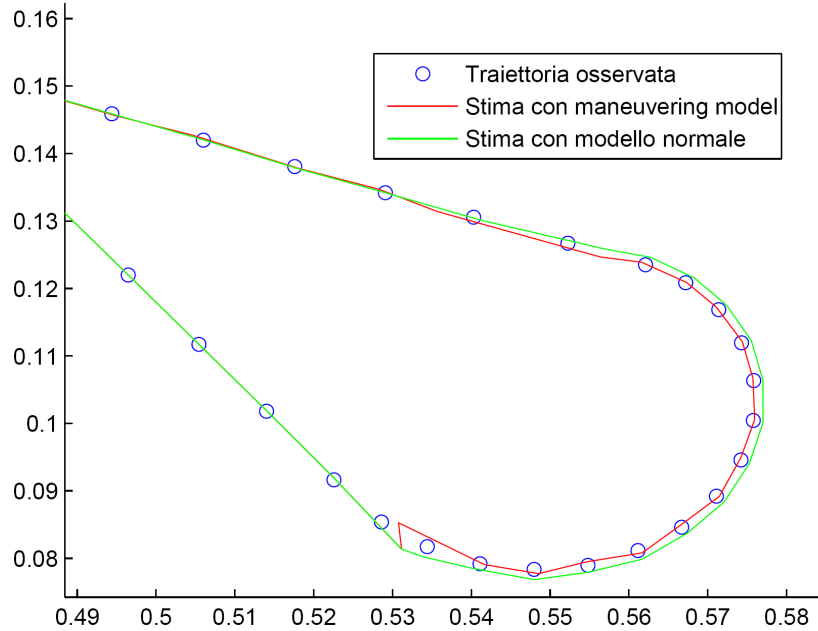
e quando è troppo piccolo (accelerazione trascurabile) si torna al modello normale.

Modello normale

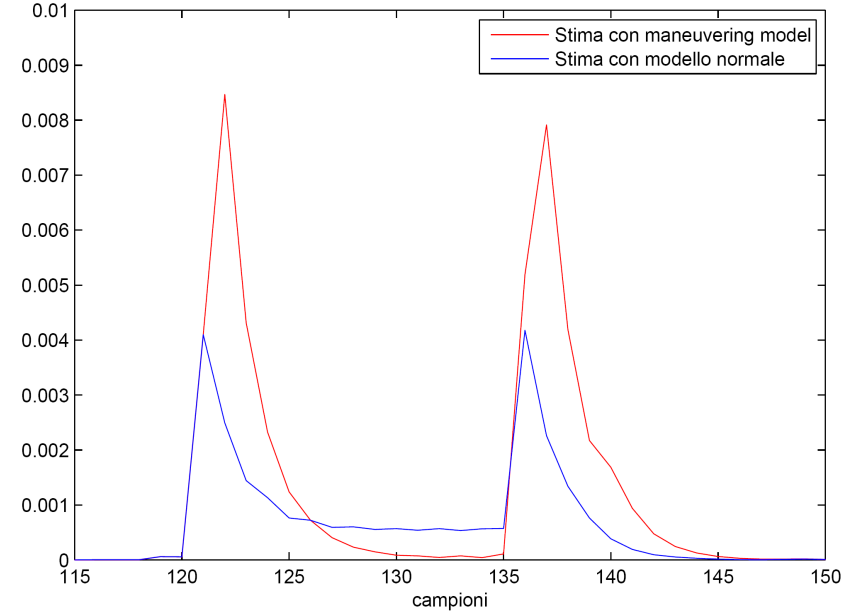


Maneuvering target - Risultati

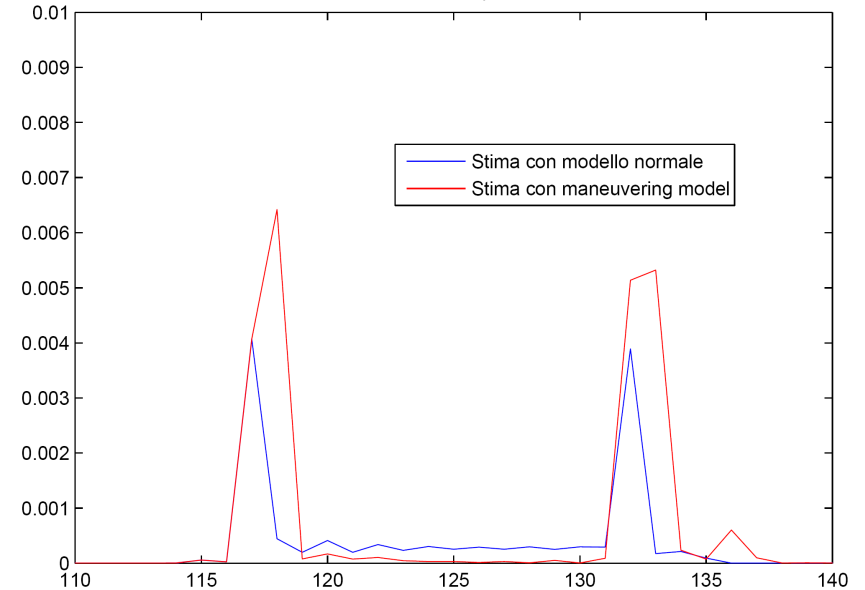
Tracking con osservazioni perfette - Confronto modello normale con maneuvering model



Tracking con osservazioni perfette - Confronto ρ modello normale e maneuvering model

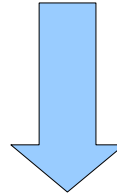


Tracking con osservazioni perfette - Confronto ρ modello normale e maneuvering model



Il motivo per cui conviene utilizzare i due modelli passando dall'uno all'altro e non, ad esempio, solo il maneuvering model, è che in assenza di manovre un modello di ordine maggiore aumenta l'errore di stima sia per la posizione che per la velocità.

Si combinano 2 o più rilevazioni provenienti da diversi sensori.



Migliora la stima

Vantaggi:

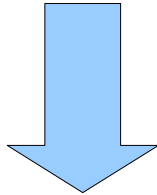
- Si riduce l'incertezza delle singole misurazioni
- Aumenta la reiezione al rumore
- Possono essere tollerati malfunzionamenti dei singoli sensori
- Può essere estesa la copertura della rete
- Soluzione a problemi di occlusione parziale



Tracking con 2 telecamere

1. **Fusione al passo di predizione:** lo scambio di informazione tra due tc (A e B) avviene al passo di predizione. Per fare ciò si utilizza un modello del tipo:

$$\mathbf{x}(t+1) = \begin{bmatrix} \mathbf{x}^A(t+1) \\ \mathbf{x}^B(t+1) \end{bmatrix} = \begin{bmatrix} (\mathbf{I} - \Lambda) \mathbf{F}^A & \Lambda \mathbf{F}^B \\ \Lambda \mathbf{F}^A & (\mathbf{I} - \Lambda) \mathbf{F}^B \end{bmatrix} \begin{bmatrix} \mathbf{x}^A(t) \\ \mathbf{x}^B(t) \end{bmatrix} + \begin{bmatrix} \mathbf{w}^A(t) \\ \mathbf{w}^B(t) \end{bmatrix}$$
$$\mathbf{z}(t) = \begin{bmatrix} \mathbf{z}^A(t) \\ \mathbf{z}^B(t) \end{bmatrix} = \begin{bmatrix} \mathbf{H}^A & 0 \\ 0 & \mathbf{H}^B \end{bmatrix} \begin{bmatrix} \mathbf{x}^A(t) \\ \mathbf{x}^B(t) \end{bmatrix} + \begin{bmatrix} \mathbf{v}^A(t) \\ \mathbf{v}^B(t) \end{bmatrix}$$



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

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

2 filtri di kalman
decentralizzati
con ingressi

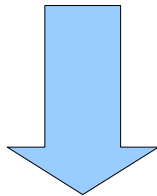


Tracking con 2 telecamere

2. **Fusione al passo di stima** : si utilizza la matrice dei pesi Λ nell'equazione delle osservazioni:

$$\mathbf{x}(t+1) = \begin{bmatrix} \mathbf{x}^A(t+1) \\ \mathbf{x}^B(t+1) \end{bmatrix} = \begin{bmatrix} \mathbf{F}^A & 0 \\ 0 & \mathbf{F}^B \end{bmatrix} \begin{bmatrix} \mathbf{x}^A(t) \\ \mathbf{x}^B(t) \end{bmatrix} + \begin{bmatrix} \mathbf{w}^A(t) \\ \mathbf{w}^B(t) \end{bmatrix}$$

$$\mathbf{z}(t) = \begin{bmatrix} \mathbf{z}^A(t) \\ \mathbf{z}^B(t) \end{bmatrix} = \begin{bmatrix} (\mathbf{I} - \Lambda) \mathbf{H}^A & \Lambda \mathbf{H}^B \\ \Lambda \mathbf{H}^A & (\mathbf{I} - \Lambda) \mathbf{H}^B \end{bmatrix} \begin{bmatrix} \mathbf{x}^A(t) \\ \mathbf{x}^B(t) \end{bmatrix} + \begin{bmatrix} \mathbf{v}^A(t) \\ \mathbf{v}^B(t) \end{bmatrix}$$



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

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



Scelta dei pesi

Λ = peso, serve a bilanciare i contributi delle diverse telecamere. Esistono numerosi criteri di assegnazione dei pesi. Consideriamo due casi, ovvero:

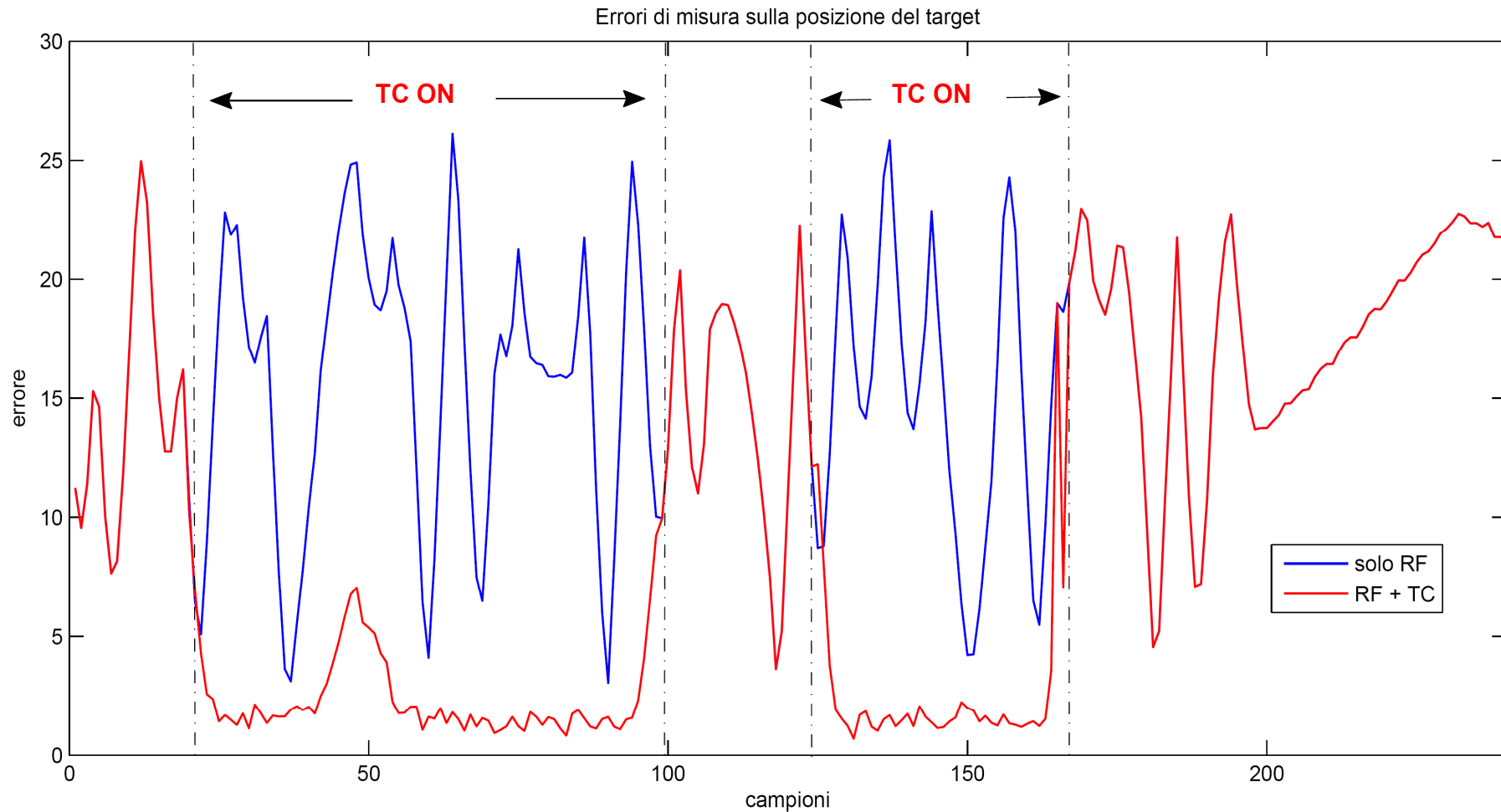
a) **pesi uniformi**, ovvero si assegna lo stesso peso a tutte le misure:

$$\lambda^i = \frac{1}{n}$$

b) **pesi dipendenti dalla distanza** del veicolo dal baricentro delle telecamere:

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

Si ottiene in questo caso un sistema tempo-variante in quanto i pesi variano ad ogni istante.

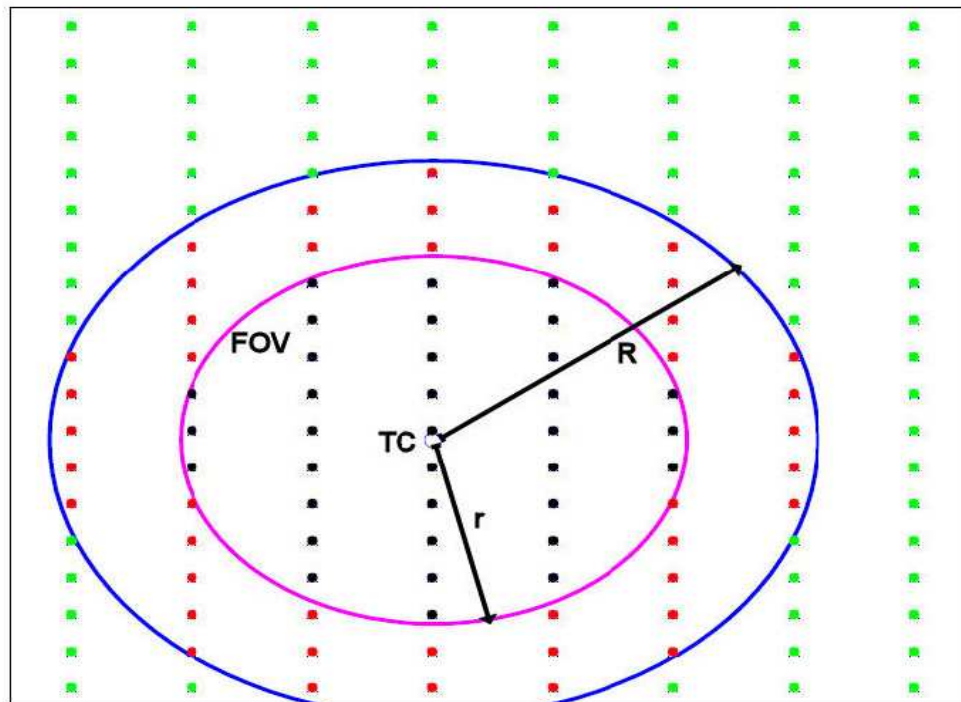


Abbiamo quindi deciso di basarci solo sull'informazione delle telecamere quando almeno una di esse è accesa.

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

$$R = r + D$$

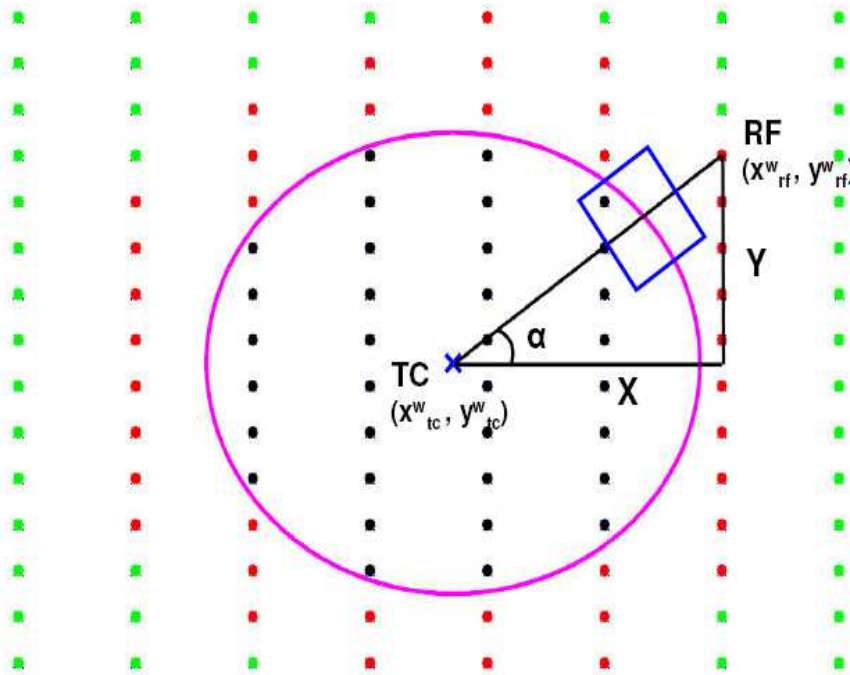
D = DIAGONALE DEL RETTANGOLO CHE RAPPRESENTA
LA ZONA COPERTA DA CIASCUN SENSORE



- 'STANDBY'
- 'ALERT'
- 'TRACKING'
- 'NOSIGNAL'

Inizializzazione delle Telecamere

$$\begin{bmatrix} x_{rf} \\ y_{rf} \end{bmatrix} = \begin{bmatrix} x^w \\ y^w \end{bmatrix} \implies \alpha = \arctan\left(\frac{Y}{X}\right) = \arctan\left(\frac{y_{rf} - y_{tc}^w}{x_{rf} - x_{tc}^w}\right)$$



- *stato 'ALERT'* :

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

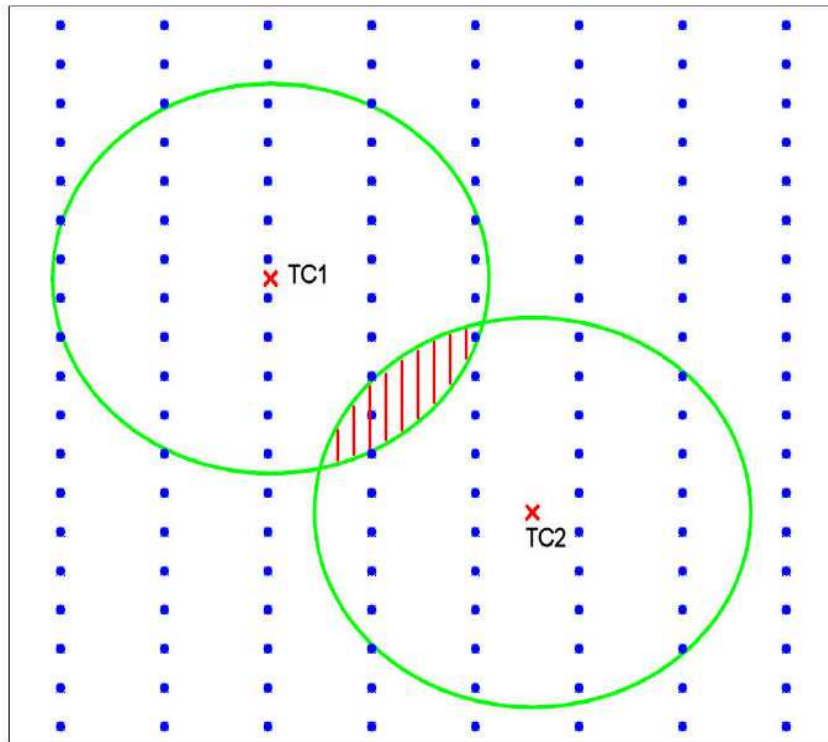
- *stato 'NOSIGNAL'* :

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

Handover

HANDOVER : operazione di passaggio dell'inseguimento del target da una telecamera ad un'altra

OVERLAPPING : sovrapposizione parziale dei campi visivi di telecamere diverse



INIZIALIZZAZIONE DELLA SECONDA TELECAMERA :

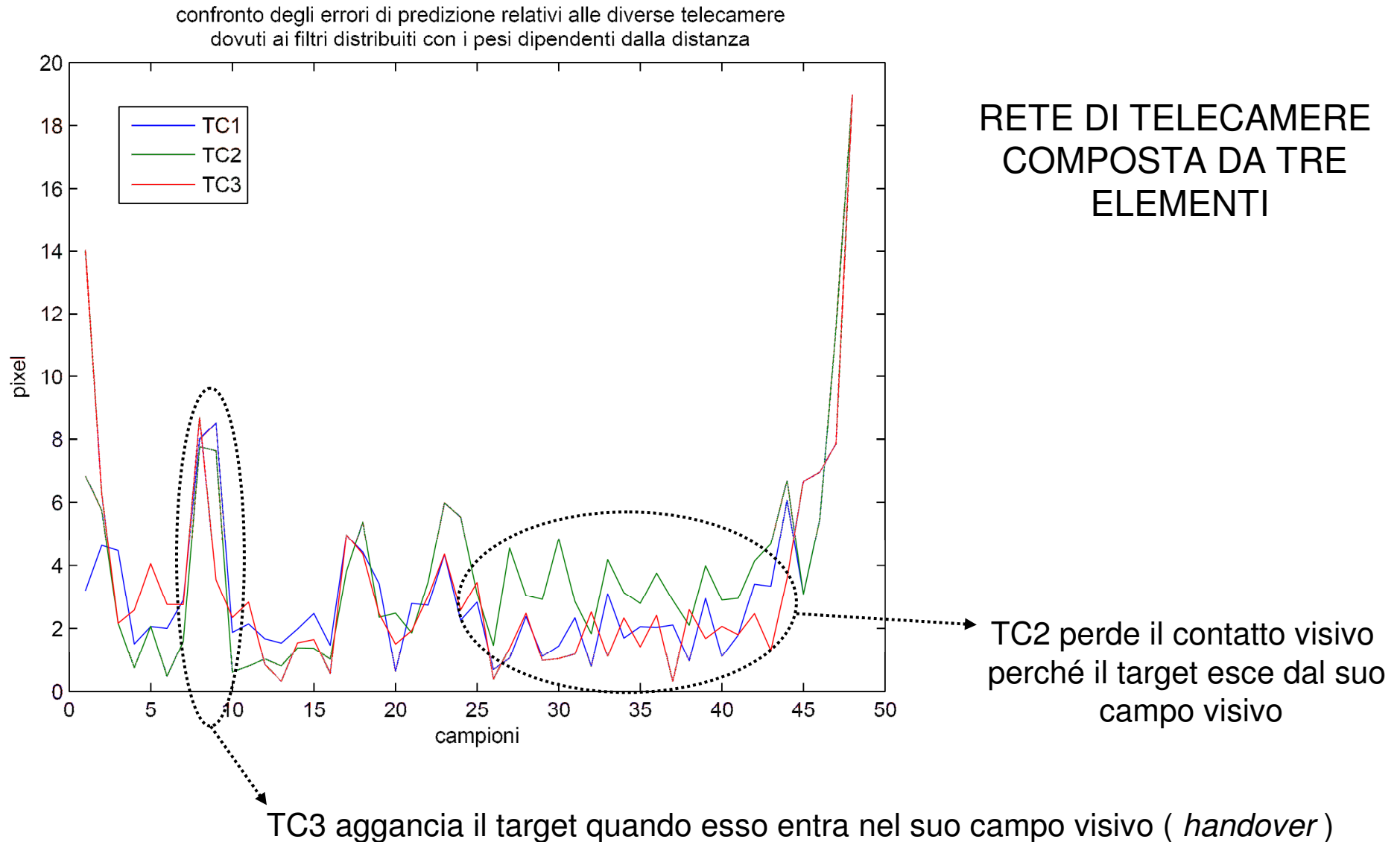
- *DALLE OSSERVAZIONI*

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

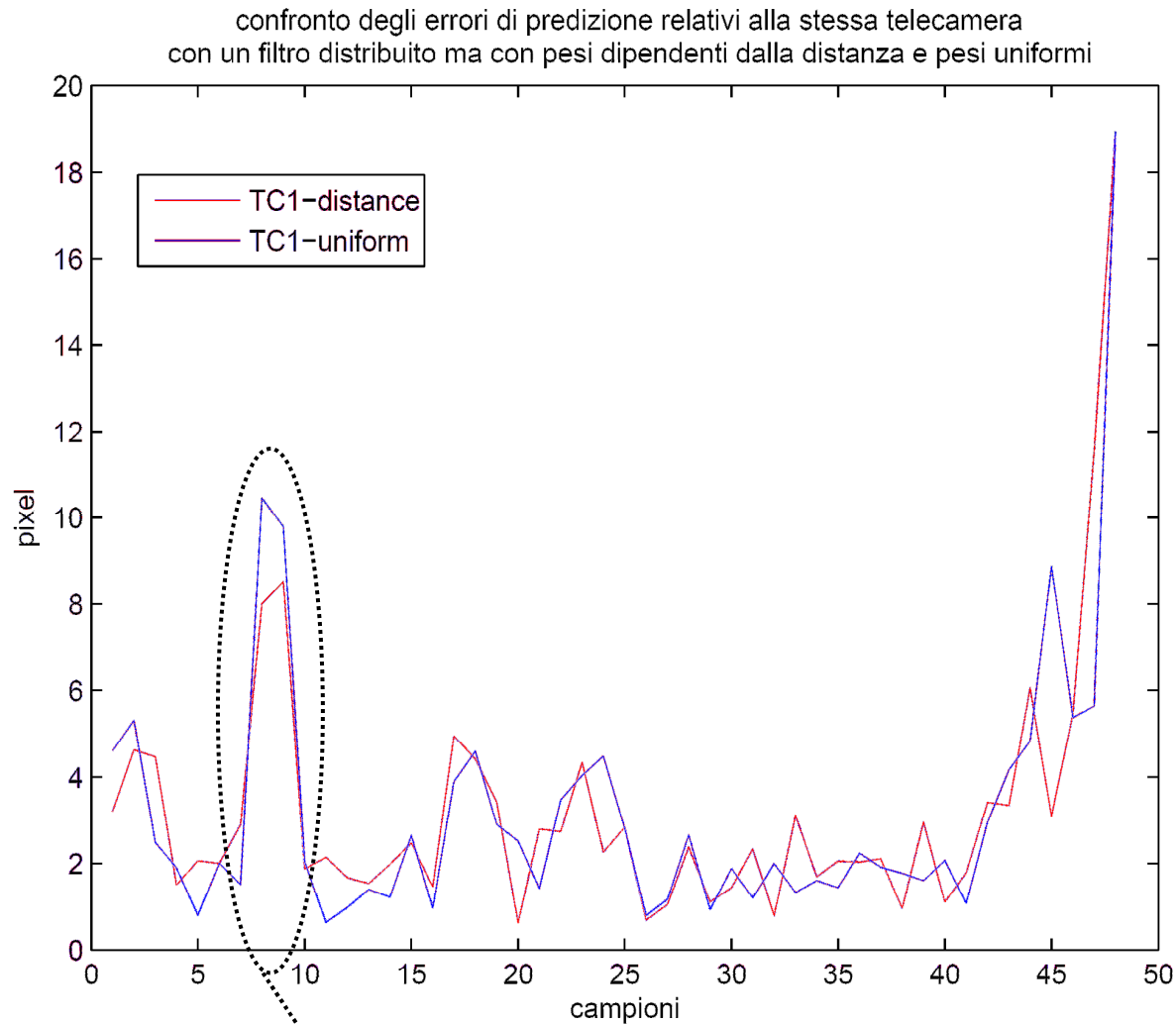
- *DALLE PREDIZIONI*

$$\begin{bmatrix} x_{im} \\ y_{im} \end{bmatrix} = \begin{bmatrix} x_{pred}^w \\ y_{pred}^w \end{bmatrix}$$

Risultati Simulazioni 1



Risultati Simulazioni 2

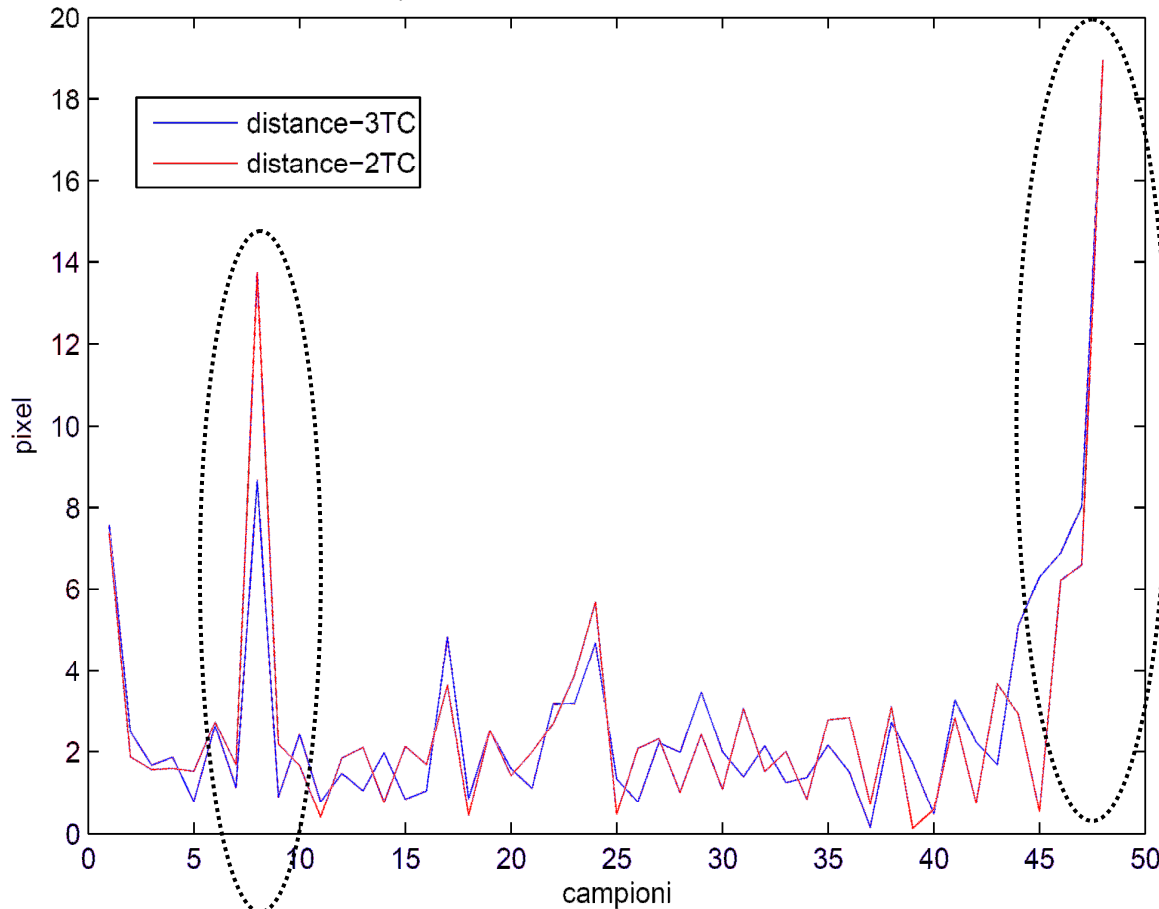


- Comportamento simile dei due algoritmi nell'intera simulazione tranne che nel momento dell'handover

TC3 aggancia il target quando esso entra nel suo campo visivo (*handover*)

Risultati Simulazioni 3

confronto dell'errore di predizione
per il filtro centralizzato con pesi dipendenti dalla distanza
quando si hanno due o tre telecamere



- Comportamento simile dei due algoritmi nell'intera simulazione tranne che nel momento dell'handover

la rete di telecamere perde il contatto visivo con il target e si passa alla localizzazione da sensori

handover

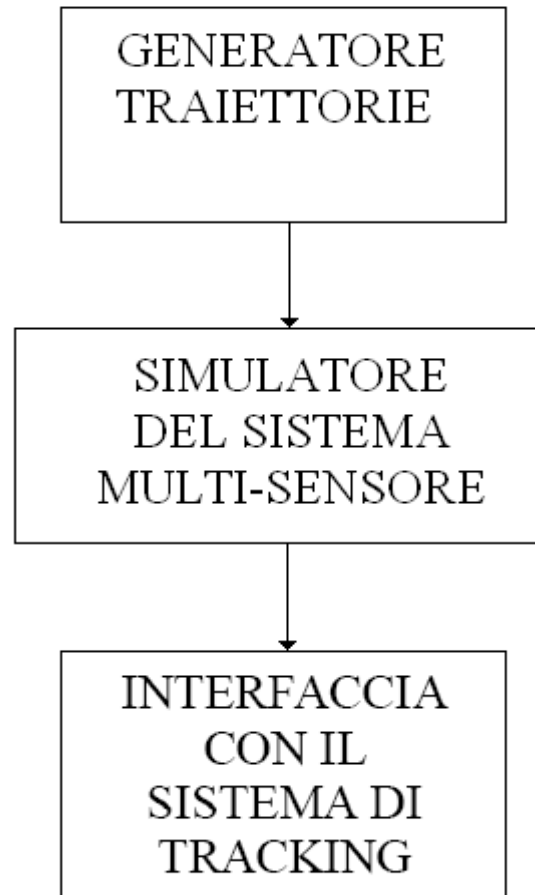


Interfaccia Grafica

Le idee che stanno alla base

- Soddisfare le esigenze dell'utente
- Come poter simulare la realtà
- Adattare il codice all'idea

Funzionalità del programma



Interfaccia del Sistema Telecamere

Gui_sceltaCamera

Per Scegliere l'ultima camera cliccare con il tasto destro il mouse

cam1
cam2
cam3
cam4

Conferma Modifica

Cancela

Aggiungi Camera

nome cam
cam4

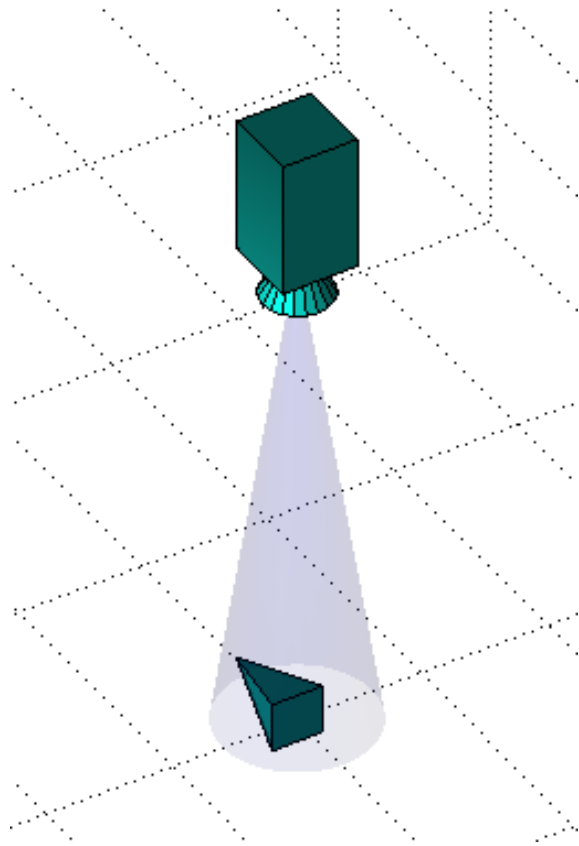
Asse x	Asse y	Copertura	Theta
7.9152	7.1414	1.25	39.6

Visualizza

Realizzazione dei componenti

- Realizzazione della struttura telecamera
- Idea di movimentazione e di stato
- Virtualizzazione dei frame di ogni telecamera
- Creazione dell'ambiente

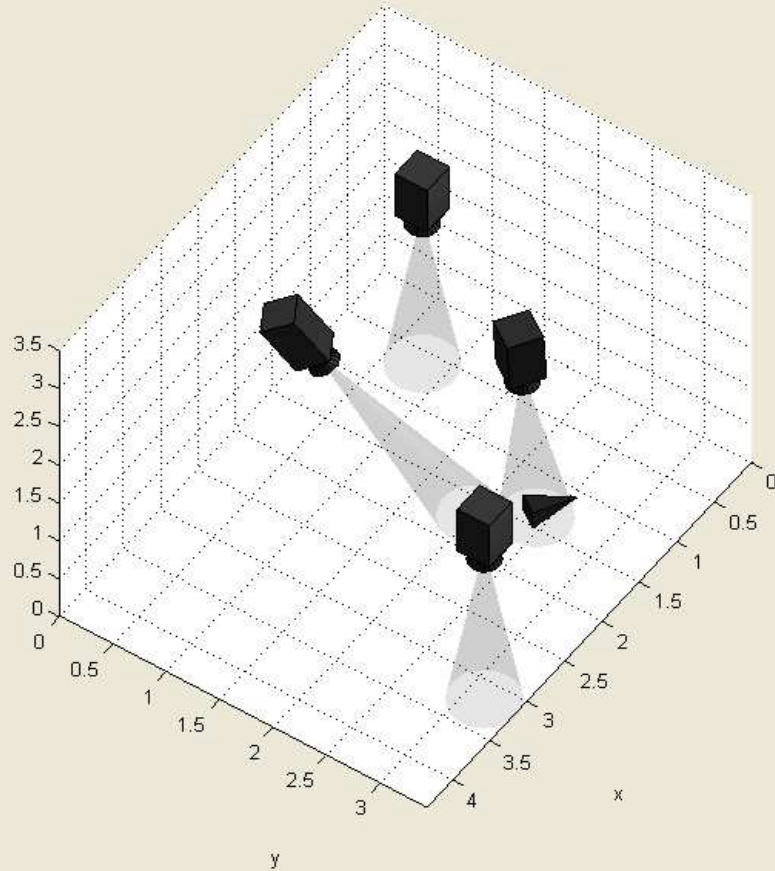
La telecamera



- Componenti
- Denotazione di Denavit-Hartenberg

Stati della telecamera

GUI_ambiente



Stand-by



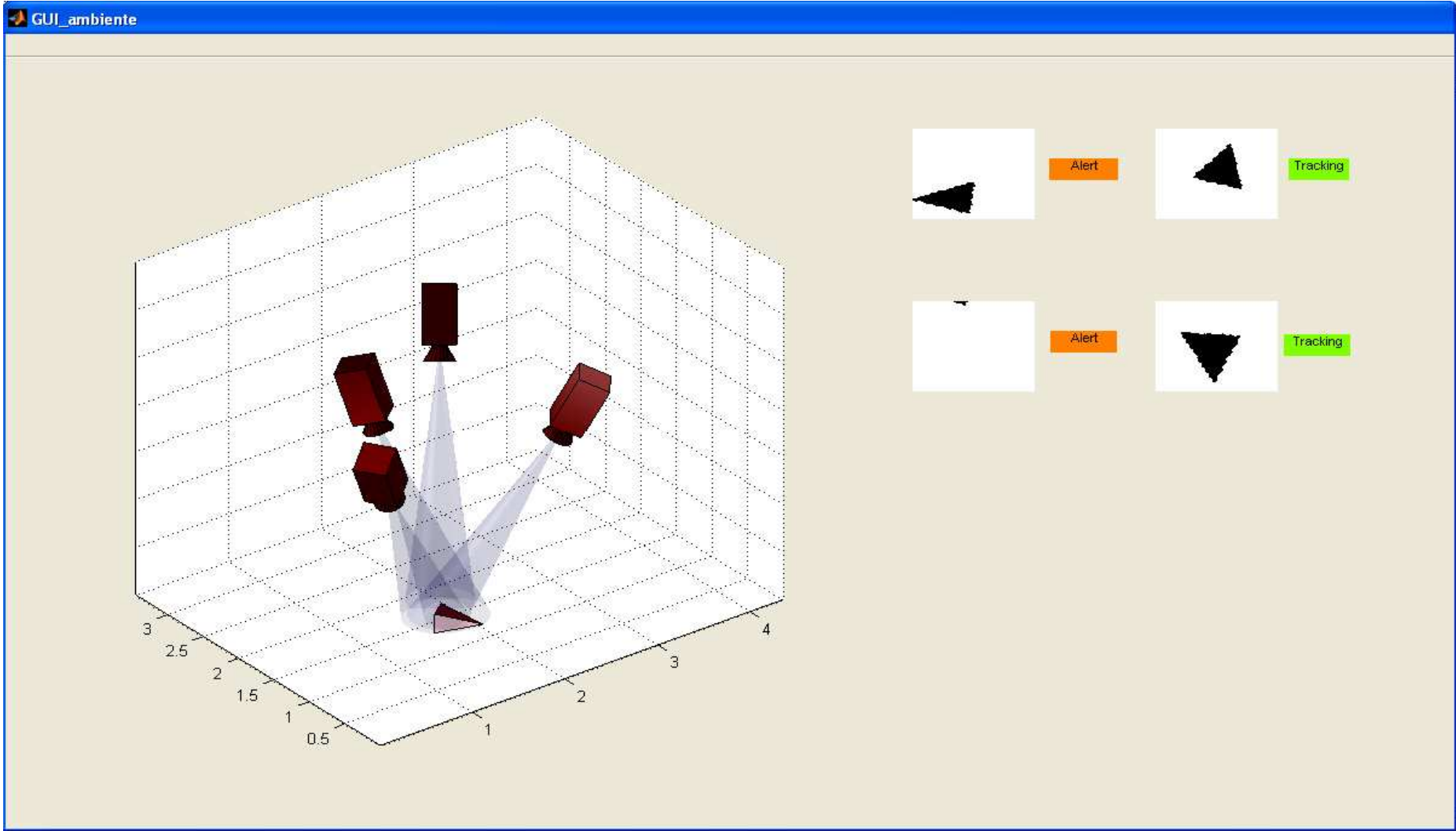
Stand-by



Alert



Tracking





Conclusioni

Sviluppi futuri e possibili applicazioni

- Testare gli algoritmi in un ambiente reale
- Multi-target Tracking
- Art Gallery problem
- Autocalibrazione telecamere sfruttando overlapping regions
- Zoom-While-Tracking