



SLAM e Visione Robotica

Corso di Robotica Industriale

SLAM: Navigazione umana

Feature



Feature



Feature



SLAM: definizione

- **SLAM**: Simultaneous Mapping and Localization
- **CML**: Concurrent Mapping and Localization

E' possibile per un veicolo che si trova in un ambiente sconosciuto, **costruire una mappa** dell'ambiente e contemporaneamente usare quest'ultima per **localizzarsi**?

SLAM: Applicazioni

Esplorazioni planetarie



Mars Sojourner

Esplorazioni di ambienti sottomarini

SLAM: problema di osservazione

Formalizzazione come problema di osservabilità

Linearizzazione

Filtro di Kalman

Ellissoidi di confidenza

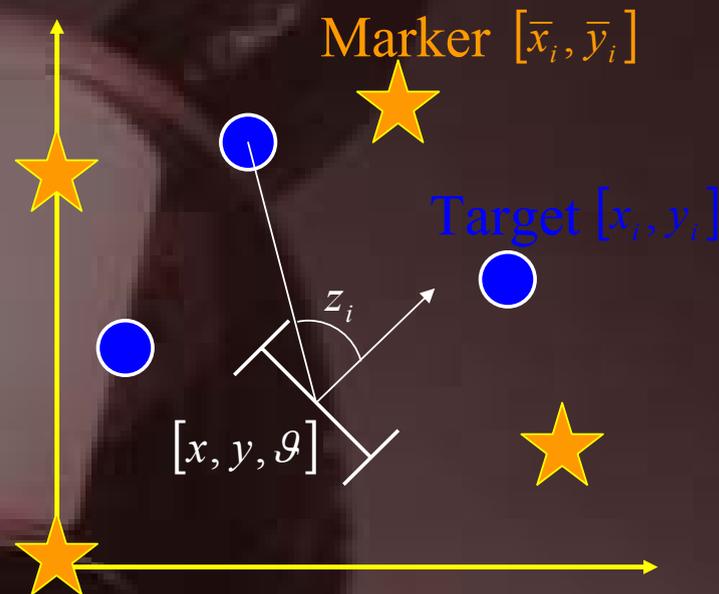
SLAM: modello

Vettore di stato

$$x = \left[\underbrace{x \quad y \quad \vartheta}_{\text{Veicolo}} \quad \underbrace{x_1 \quad y_1}_{T_1} \quad \dots \quad \underbrace{x_N \quad y_N}_{T_N} \right]^T$$

Evoluzione

$$\dot{x} = \begin{bmatrix} \cos(\vartheta) \\ \sin(\vartheta) \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u1 + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} u2$$



Uscite

$$z_i = h_i(x) = \pi - \vartheta + \arctan(y - y_i, x - x_i) \quad i = 1..N$$

$$z_j = h_j(x) = \pi - \vartheta + \arctan(y - \bar{y}_j, x - \bar{x}_j) \quad j = N..N + M$$

SLAM: modello linearizzato

Matrice di osservazione

$$A = 0 \quad \longrightarrow \quad \left[C^T \quad A^T C^T \quad \dots \quad A^{n-1T} C^T \right]^T = C$$

$$C = \left[\begin{array}{ccc|cccccc} c_{1,1} & c_{1,2} & -1 & -c_{1,1} & c_{1,2} & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & -1 & \vdots \\ c_{N,1} & c_{N,2} & -1 & 0 & 0 & 0 & 0 & \dots & -c_{N,1} & -c_{N,2} \\ \hline c_{N+1,1} & c_{N+1,2} & -1 & & & & & & & \\ \vdots & \vdots & -1 & & & & & & & \\ c_{N+M,1} & c_{N+M,2} & -1 & & & & & & & \end{array} \right]$$

$O_{M \times N}$

SLAM: modello linearizzato

- Studiando il rango della matrice C si può concludere che il sistema risulta non completamente osservabile se:
 - Vi sono presenti dei target $N \neq 0$
 - Vi sono presenti meno di tre marker $M < 3$

SLAM: modello non lineare

L'esperienza nella navigazione dice che la localizzazione è possibile con $M=2$, qualsiasi sia il numero N di target

Esempio: 2 Marker, 0 Target

$$\dot{x} = \begin{bmatrix} \cos(\mathcal{G}) \\ \sin(\mathcal{G}) \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2$$

$$z_1 = \pi - \mathcal{G} + \arctan(y, x)$$

$$z_1 = \pi - \mathcal{G} + \arctan(y, x - d)$$



Sistema linearizzato

$$A=0$$

$$B = \begin{bmatrix} \cos(\mathcal{G}) & 0 \\ \sin(\mathcal{G}) & 0 \\ 0 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} -\frac{y}{x^2 + y^2} & \frac{x}{x^2 + y^2} & -1 \\ -\frac{y}{(x-d)^2 + y^2} & \frac{x}{(x-d)^2 + y^2} & -1 \end{bmatrix}$$

$$\ker \begin{bmatrix} C^T & A^T C^T & \dots & A^{n-1T} C^T \end{bmatrix}^T \neq 0$$

Spazio non oss. di dim. 1

Sistema non lineare

- La codistribuzione di osservabilità ha rango pieno. Quest'ultima è ottenuta differenziando le funzioni che compongono lo spazio di osservabilità; tali funzioni sono tutte le h_i e tutte le derivate di **Lie** del tipo $L_{x_1} \dots L_{x_k} h_i$ con $x_j = f, g_1, \dots, g_m$

Sistema non lineare

$$L_{g_1} h_1 = \frac{\partial h_1}{\partial X} g_1 = \frac{x \sin(\vartheta) - y \cos(\vartheta)}{x^2 + y^2}$$

$$L_{g_2} h_1 = \frac{\partial h_1}{\partial X} g_2 = -1$$

$$L_{g_1} h_2 = \frac{\partial h_2}{\partial X} g_1 = \frac{(x-d) \sin(\vartheta) - y \cos(\vartheta)}{(x-d)^2 + y^2}$$

$$L_{g_2} h_2 = \frac{\partial h_2}{\partial X} g_2 = -1$$

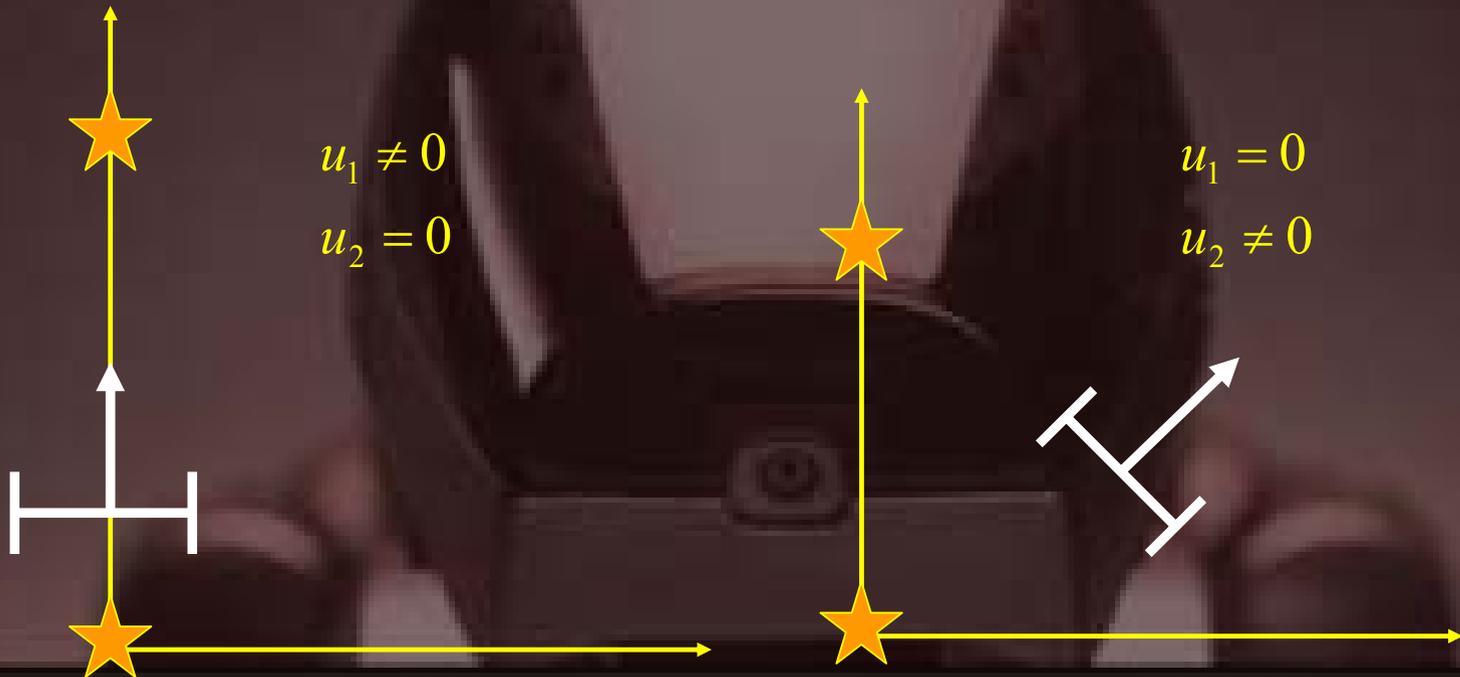
$$L_{g_2} L_{g_1} h_1 = \frac{\partial L_{g_1} h_1}{\partial X} g_2 = \frac{x \cos(\vartheta) + y \sin(\vartheta)}{x^2 + y^2}$$

Sistema non lineare

Si può far vedere che, dO ha generalmente **rango pieno**.

$$dO = \begin{bmatrix} dL_{g_1} h_1 \\ dL_{g_1} h_2 \\ dL_{g_2} h_1 \\ dL_{g_2} h_2 \\ dL_{g_1} L_{g_2} h_1 \end{bmatrix}$$

Casi di non osservabilità



Il Filtro di Kalman (1960)

- Il filtro di Kalman affronta il problema di stimare lo stato di un sistema dinamico

$$x_{k+1} = A_k x_k + B_k u_k + w_k$$

$$z_k = H_k x_k + v_k$$

- Con w_k, v_k rispettivamente rumore di processo e sulle uscite. Si assume che siano processi aleatori bianchi, indipendenti tra loro con distribuzione di probabilità gaussiana con covarianza Q, R

Il Filtro di Kalman

$$e_k^- = x_k - \hat{x}_k^-$$

Errore di stima a priori

$$e_k = x_k - \hat{x}_k$$

Errore di stima a posteriori

$$P_k^- = E[e_k^- e_k^{-T}]$$

Matrice di covarianza dell'errore a priori

$$P_k = E[e_k e_k^T]$$

Matrice di covarianza dell'errore a posteriori

Filtro di Kalman

Trovare un'equazione che calcola una stima **a posteriori** dello stato come combinazione lineare di una stima **a priori** e di una differenza pesata tra una misura reale e una predetta dal modello delle uscite. La differenza è chiamata innovazione di misura o **residuo**. Il residuo riflette la discrepanza tra la misura reale e quella predetta.

$$\hat{x}_{k+1} = \hat{x}_{k+1}^- + K(z_k - \hat{z}_k)$$

La matrice di peso K , detta Guadagno di Kalman, è scelta in modo da minimizzare la matrice di covarianza dell'errore a posteriori

$$K_k = P_{k+1}^- H^T (H P_{k+1}^- H^T + R)^{-1}$$

Filtro di Kalman: predizione e correzione

Sistema

$$\begin{aligned}x_{k+1} &= A_k x_k + B_k u_k + w_k \\z_k &= H_k x_k + v_k\end{aligned}$$

Predizione

$$\begin{aligned}\hat{x}_{k+1}^- &= A_k \hat{x}_k + B_k u_k \\ \hat{z}_k &= H_k \hat{x}_k \\ P_{k+1}^- &= A P_k A^T + Q\end{aligned}$$

Correzione

$$\begin{aligned}K_k &= P_{k+1}^- H^T (H P_{k+1}^- H^T + R)^{-1} \\ \hat{x}_{k+1} &= \hat{x}_{k+1}^- + K (z_k - \hat{z}_k) \\ P_{k+1} &= (I - K H) P_{k+1}^-\end{aligned}$$

Ellissoidi di confidenza

- Variabile Aleatoria Gaussiana

$$f_X(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{(x - \eta_x)^2}{2\sigma_x^2}}$$

- Vettore Aleatorio Gaussiano

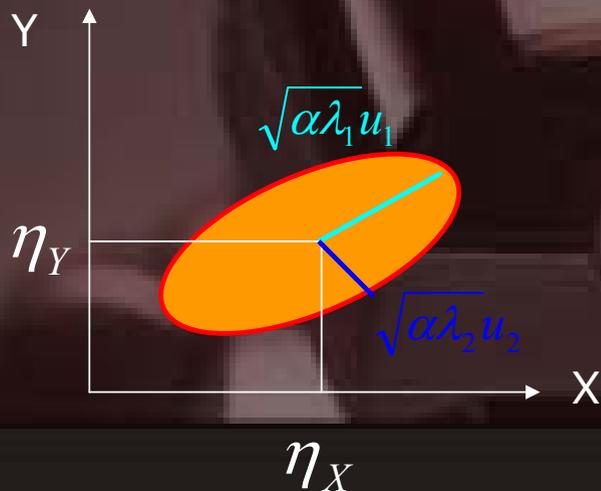
$$f_X(x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} e^{-\frac{1}{2}(x - \eta_x)^T \Sigma^{-1} (x - \eta_x)}$$

$$\Sigma = E[(x - \eta_x)(x - \eta_x)^T] \quad \text{Matrice di Covarianza}$$

Ellissoidi di confidenza

$f_X(x)$ Costante per $(x - \eta_x)^T \Sigma^{-1} (x - \eta_x) = \alpha$

$$x^T \Sigma^{-1} x = x^T U^T \Lambda U x = z^T \Lambda z \quad \text{con } z = Ux$$



$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad U = \begin{bmatrix} u_1^1 & u_1^2 \\ u_2^1 & u_2^2 \end{bmatrix}$$

Visione Robotica: Introduzione

In un sistema di visione robotica, una o più telecamere sono connesse ad un computer che interpreta le immagini di una scena reale, estraendo informazioni utili allo svolgimento di un particolare compito.

Automazione di processi industriali

Riconoscimento di oggetti.

Controllo di manipolatori.

Navigazione robotica.

Controllo di qualità.

Sorveglianza del traffico aereo.

Riconoscimento dei volti.

Ricostruzione 3D

Comunicazioni

Codifica di immagini.

Compressione video.

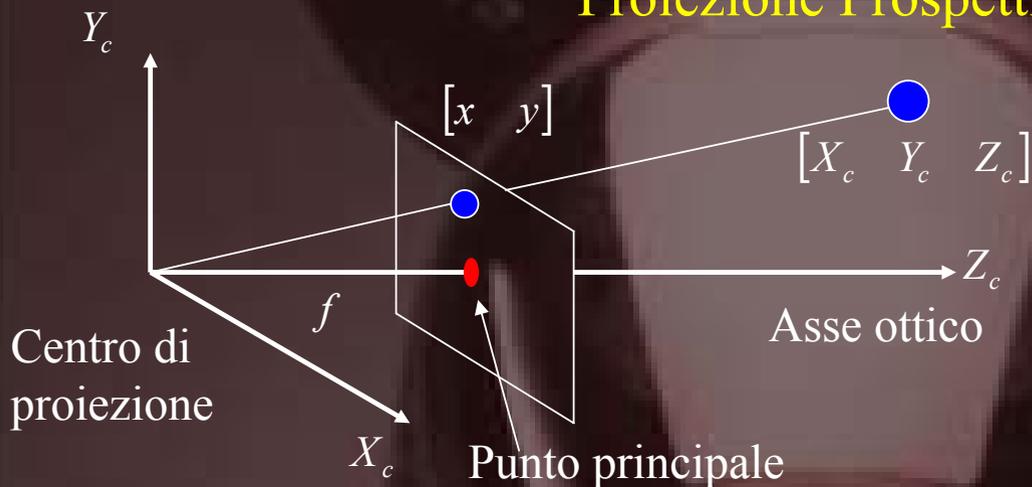
Applicazioni spaziali e militari

Inseguimento di oggetti in movimento.

Esplorazione planetaria.

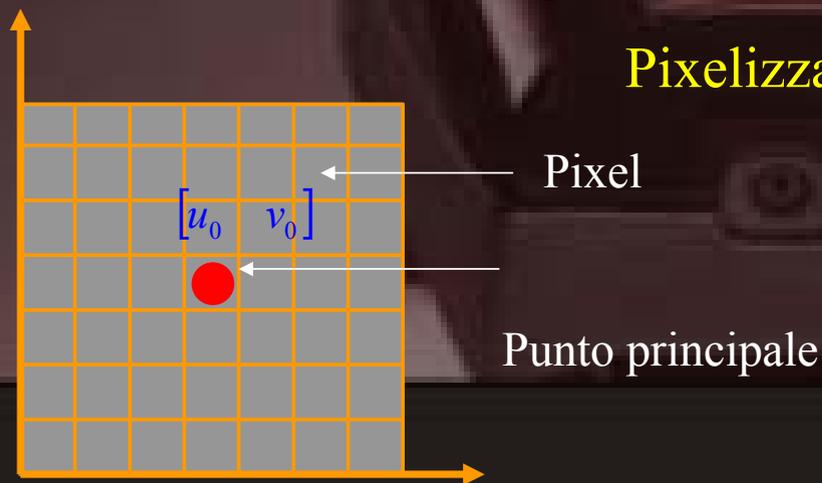
La telecamera: modello pinhole

Proiezione Prospettica



$$x = \frac{f}{Z_c} X_c$$
$$y = \frac{f}{Z_c} Y_c$$

Pixelizzazione



$$u = u_0 + k_x x$$
$$v = v_0 + k_y y$$

La telecamera: modello pinhole

$$K = \begin{bmatrix} a_x & 0 & u_0 \\ 0 & a_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Matrice di calibrazione:

Parametri intrinseci

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = K [I \mid 0] \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Coordinate omogenee

Calibrazione di una telecamera

- Ricavare la matrice di calibrazione della telecamera, inquadrando da punti di vista diversi un oggetto di geometria nota

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = K[I | 0] \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Localizzazione con Visione

1. Acquisizione immagini ← Libreria Microsoft VisSDK

2. Estrazione feature :

- Edge
- Corner

3. Tracking

4. Corrispondenze

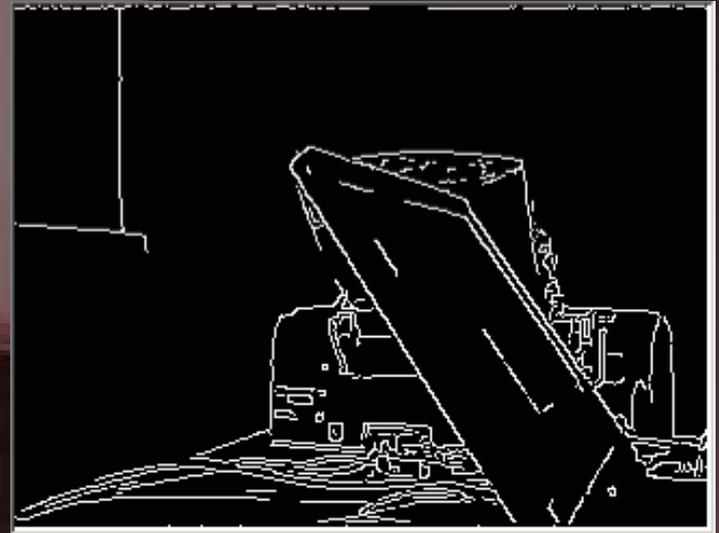
5. Ricostruzione 3D

←
←
←
Libreria Intel OpenCV

Visione: Estrazione degli edge



Immagine

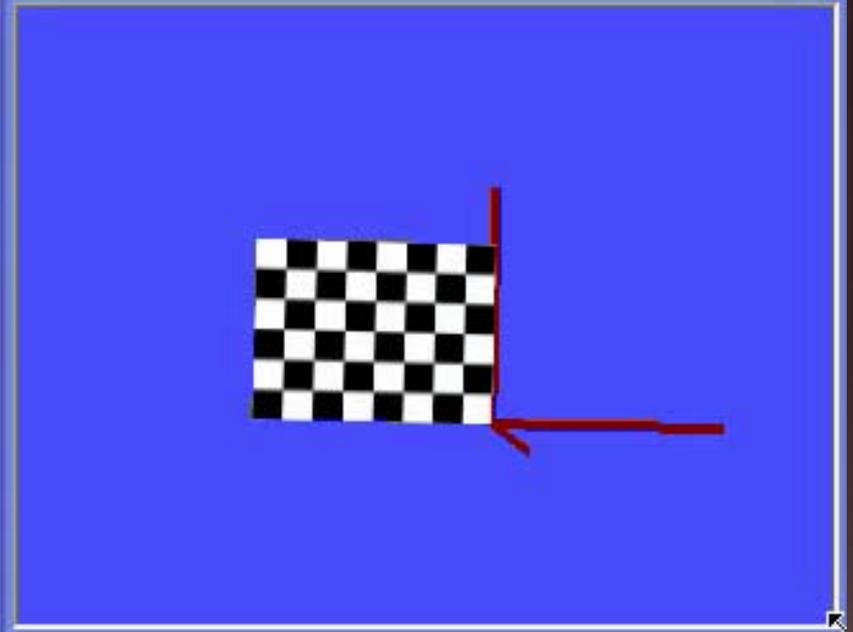
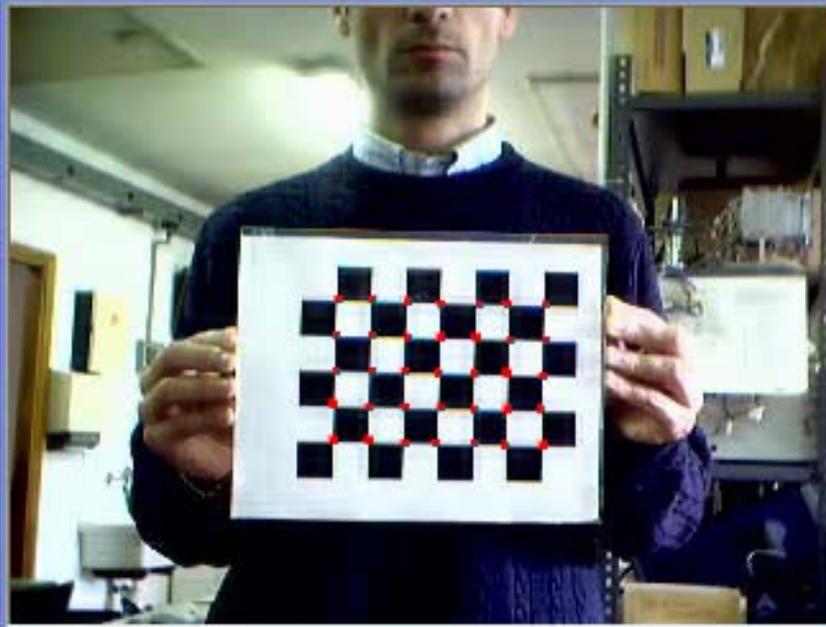


Edge estratti

Visione: Tracking dei corner



Feature Tracking (Lucas Kanade)

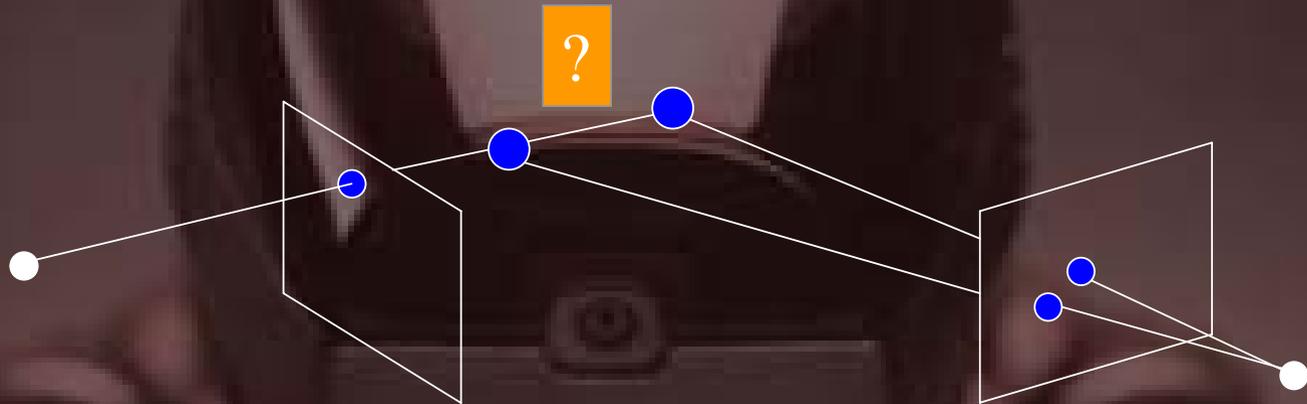


Microsoft Visual C++
Microsoft VisSDK
Intel OpenCV Library

OpenGL

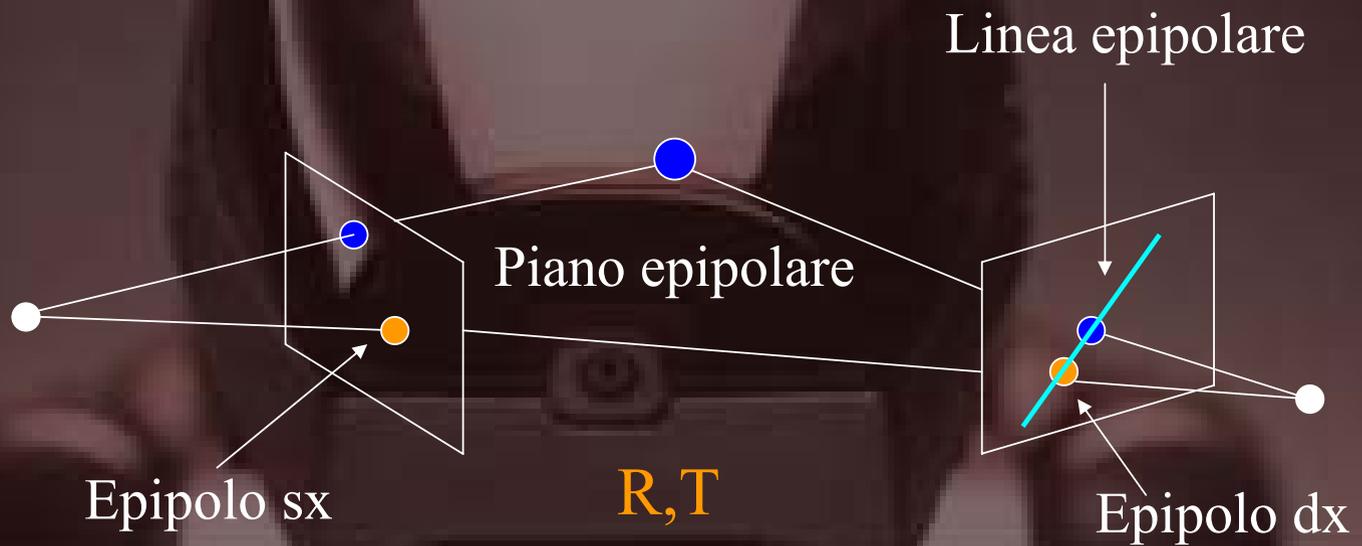
Stereovisione

Problema delle corrispondenze

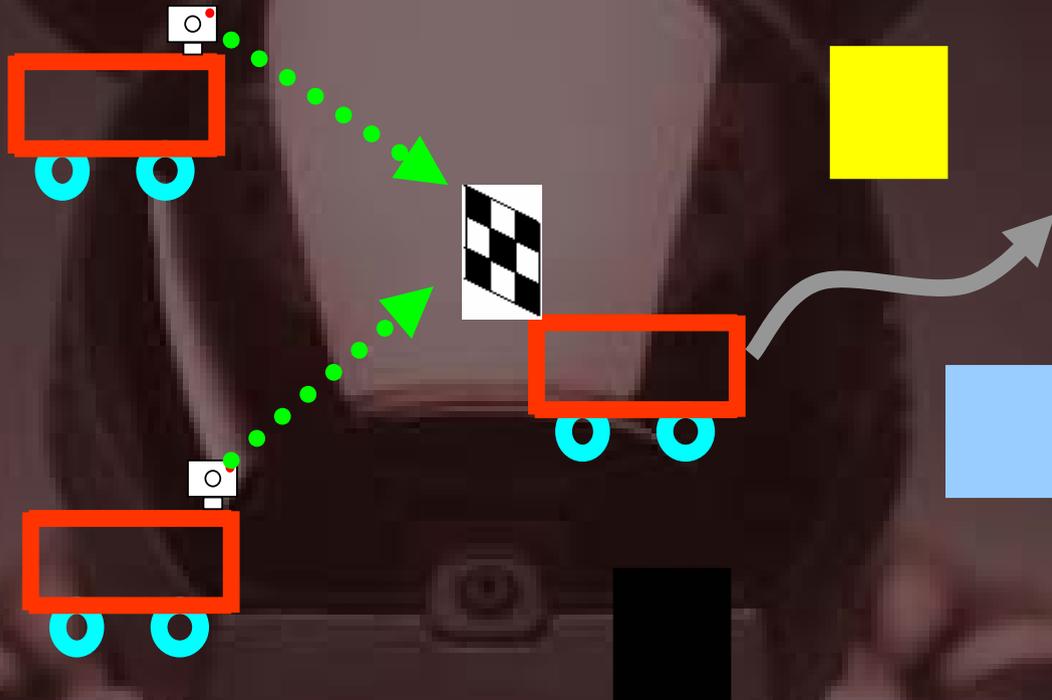


Stereovisione

Geometria Epipolare



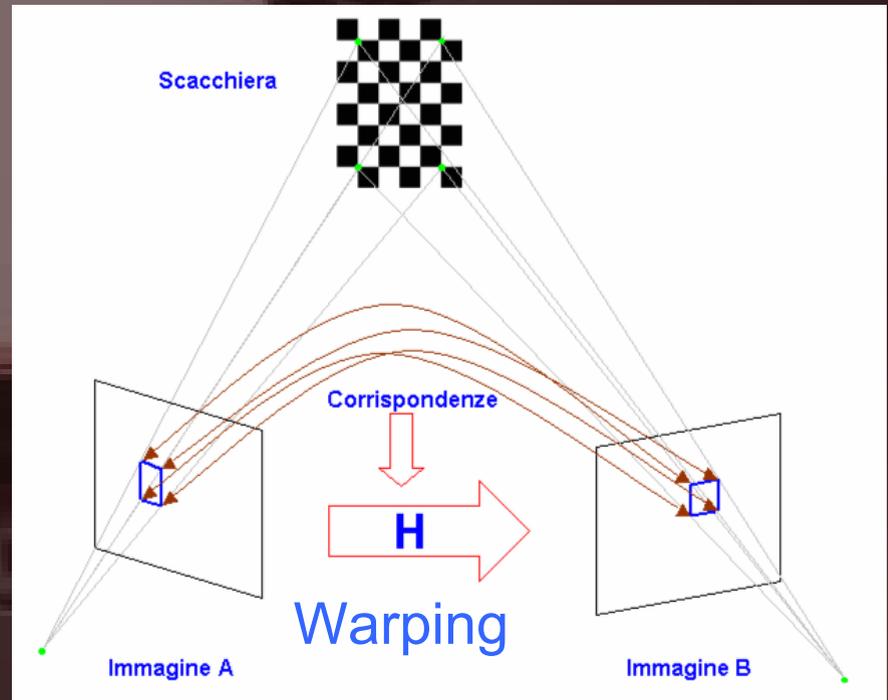
Costruzione di una mappa



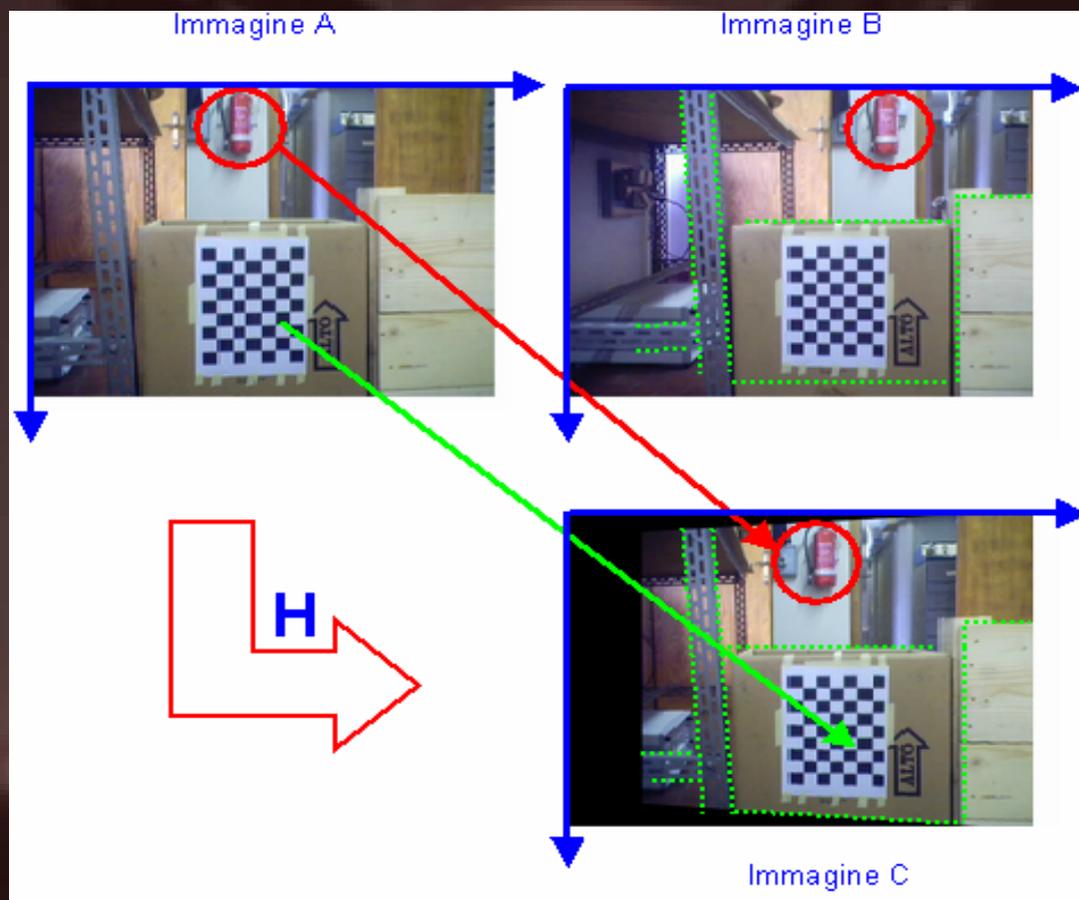
Costruzione di una mappa

Trasformazione omografica:
trasformazione tra due piani

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$$



Costruzione di una mappa



Costruzione di una mappa

Immagine B



Immagine A



H

Immagine deformata



Piano estratto



Costruzione di una mappa



Costruzione di una mappa



Costruzione di una mappa

