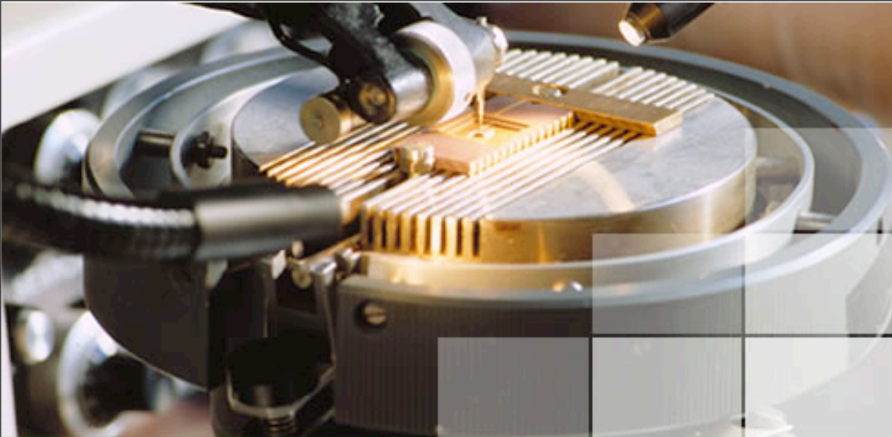


POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Use a Single Camera for Simultaneous Localization And Mapping with Mobile Objects Tracking in Dynamic Environments

Davide Migliore, Daniele Marzorati, Roberto Rigamonti, Matteo Matteucci, Domenico G. Sorrenti

Workshop on Safe navigation in open and dynamic environments. Application to autonomous vehicles

Davide Migliore
migliore@elet.polimi.it
Department of Electronics and Information,
Politecnico di Milano, Italy

What is SLAM?

Teorico e pratico

gliore - migliore@elet.polimi.it
POLITECNICO DI MILANO
Dipartimento di Elettronica e Informazione

What is SLAM?

Teorico e pratico

- The question:

"Is it possible for a mobile robot to be placed in an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map?"

H. Durrant-Whyte "Simultaneous Localization and Mapping" - RAS Magazine 2006

percezioni e azioni

What is SLAM?

Teorico e pratico

- The question:

"Is it possible for a mobile robot to be placed in an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map?"

H. Durrant-Whyte "Simultaneous Localization and Mapping" - RAS Magazine 2006

$$p(\mathbf{x}_k, \mathbf{M} | \mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_k, \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k).$$

$$\underbrace{p(\mathbf{x}_k, \mathbf{M} | \mathbf{Z}_k, \mathbf{U}_k)}_{\text{Posterior at time } k} \propto \underbrace{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{M})}_{\text{Update}} \underbrace{\int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \underbrace{p(\mathbf{x}_{k-1}, \mathbf{M} | \mathbf{Z}_{k-1}, \mathbf{U}_{k-1})}_{\text{Posterior at time } k-1} d\mathbf{x}_{k-1}}_{\text{Prediction}}$$

What is SLAM?

Teorico e pratico

- The question:

"Is it possible for a mobile robot to be placed in an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map?"

H. Durrant-Whyte "Simultaneous Localization and Mapping" - RAS Magazine 2006

$$p(\mathbf{x}_k, \mathbf{M} | \mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_k, \mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k).$$

$$\underbrace{p(\mathbf{x}_k, \mathbf{M} | \mathbf{Z}_k, \mathbf{U}_k)}_{\text{Posterior at time } k} \propto \underbrace{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{M})}_{\text{Update}} \underbrace{\int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \underbrace{p(\mathbf{x}_{k-1}, \mathbf{M} | \mathbf{Z}_{k-1}, \mathbf{U}_{k-1})}_{\text{Posterior at time } k-1} d\mathbf{x}_{k-1}}_{\text{Prediction}}$$

- One of the most important results obtained by the Robotics community
  - Implemented in a number of different domains
  - Using both parametric (EKF, UKF, EIF...) and non-parametric (FastSLAM...) approaches
  - Solved using different kind of sensors (Laser, Sonar, Cameras...)
  - Considering 3DoF or 6DoF

SLAM using Single

Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible

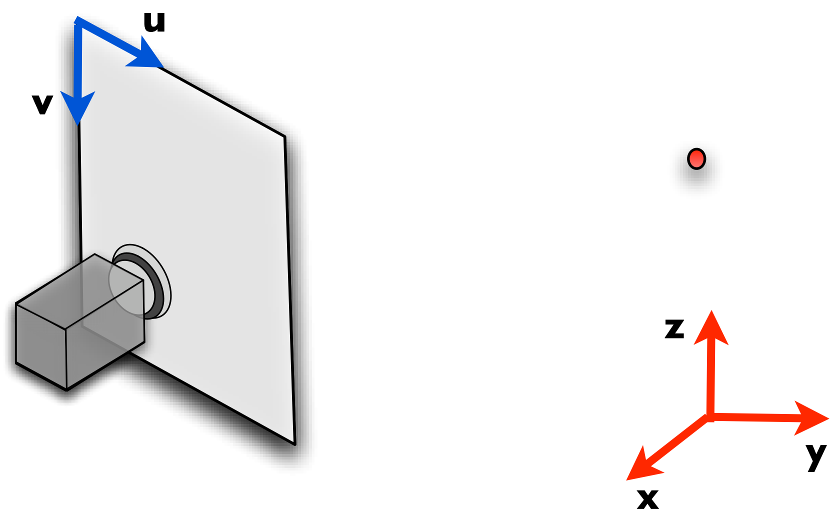
Spostamento ignoto

SLAM using Single

Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible

Spostamento ignoto

- Why is this challenging?

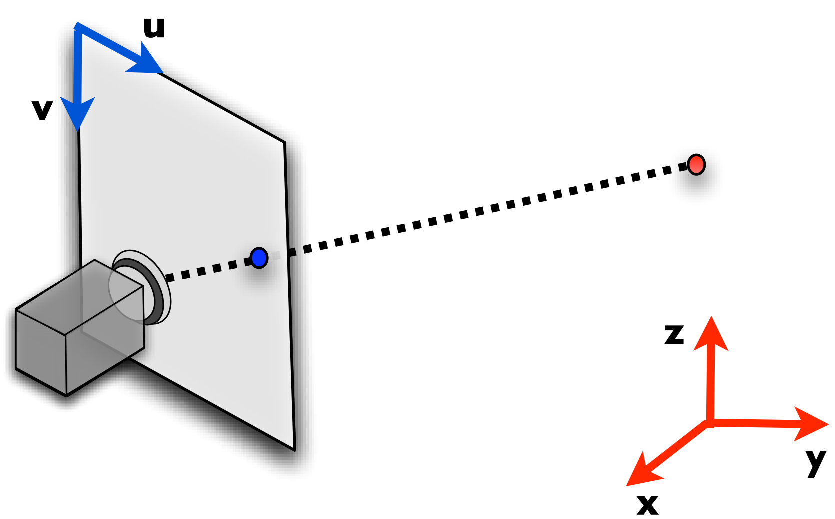


SLAM using Single

Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible

Spostamento ignoto

- Why is this challenging?

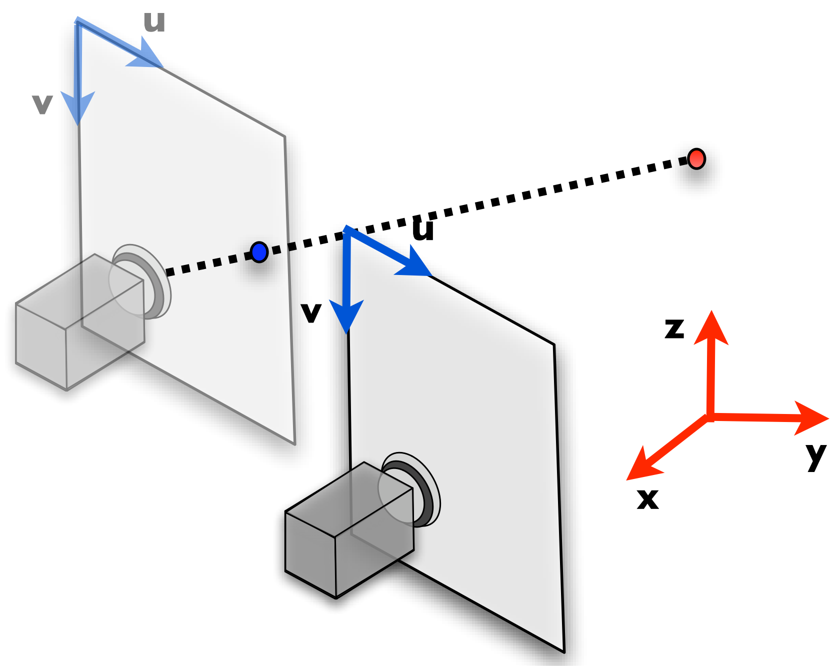


SLAM using Single

Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible

Spostamento ignoto

- Why is this challenging?

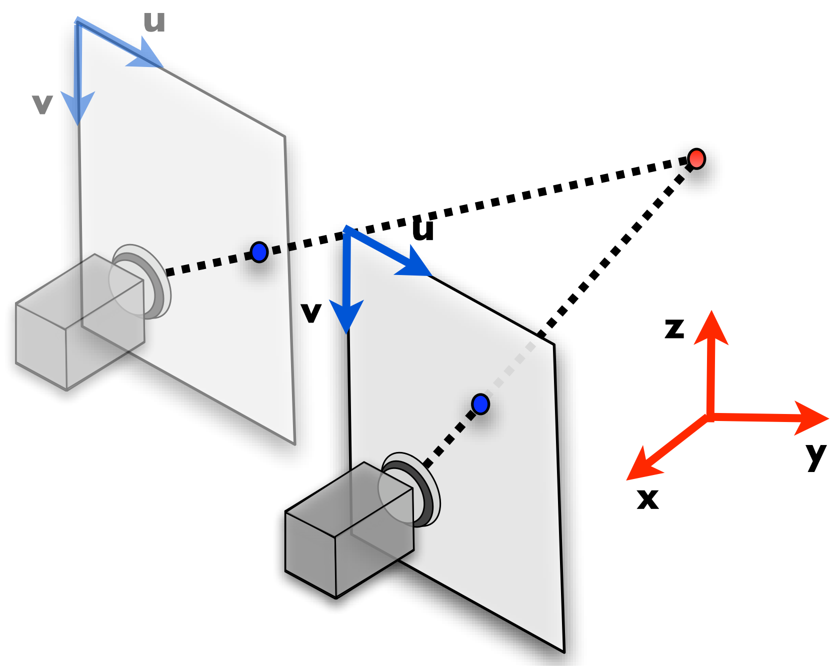


SLAM using Single

Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible

Spostamento ignoto

- Why is this challenging?

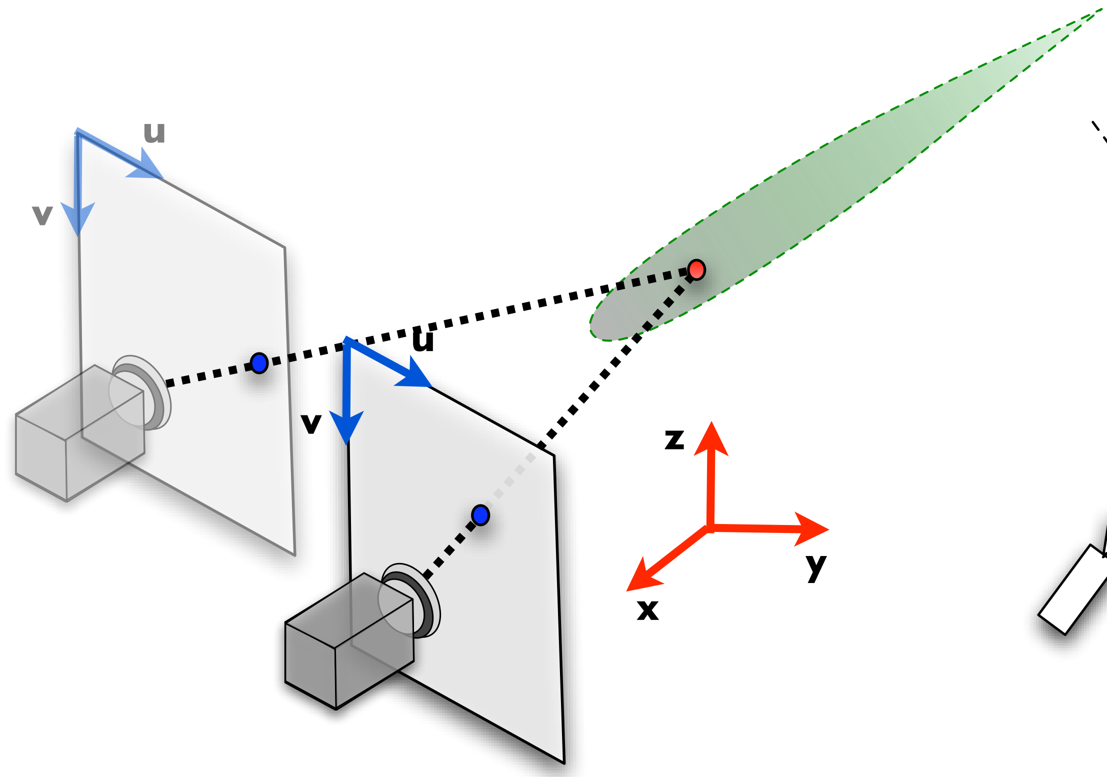


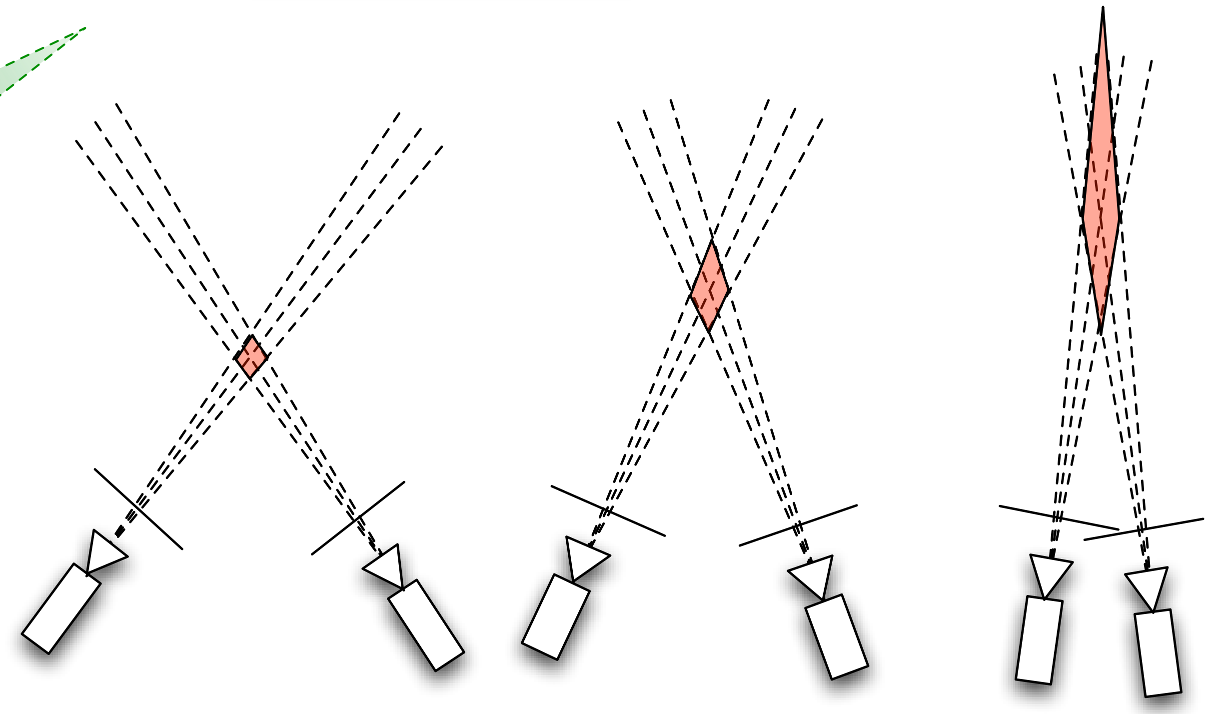
SLAM using Single

Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible

Spostamento ignoto

- Why is this challenging?



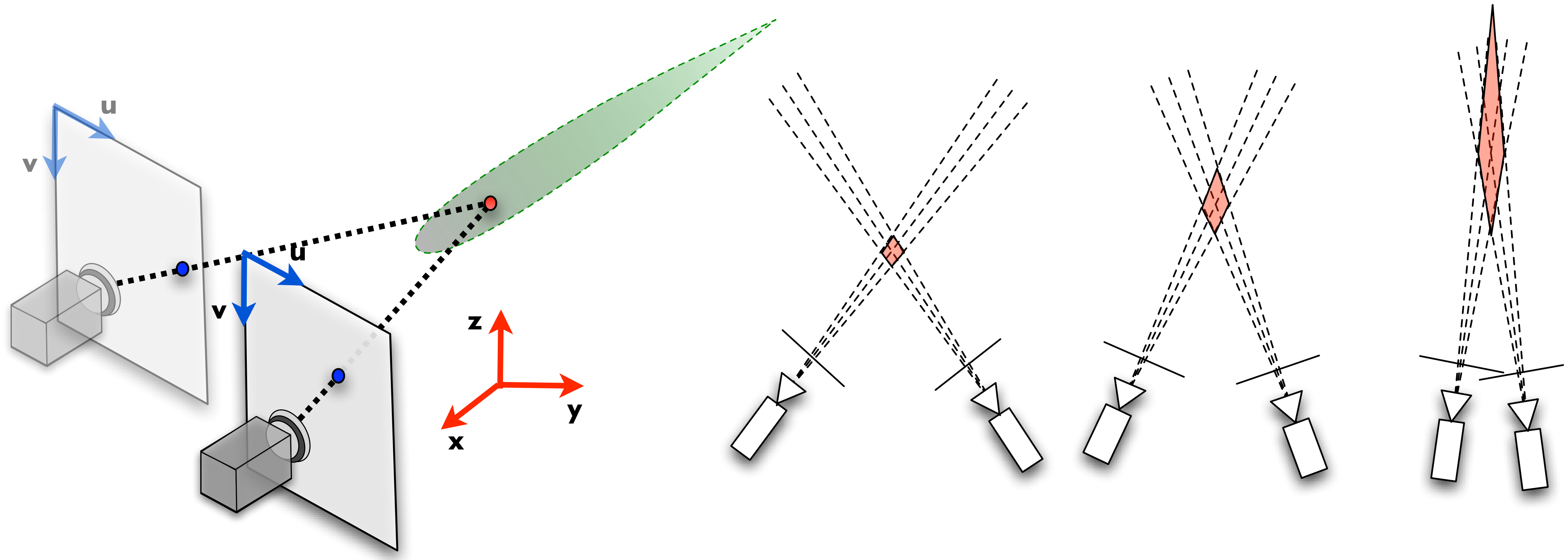


SLAM using Single

Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible

Spostamento ignoto

- Why is this challenging?



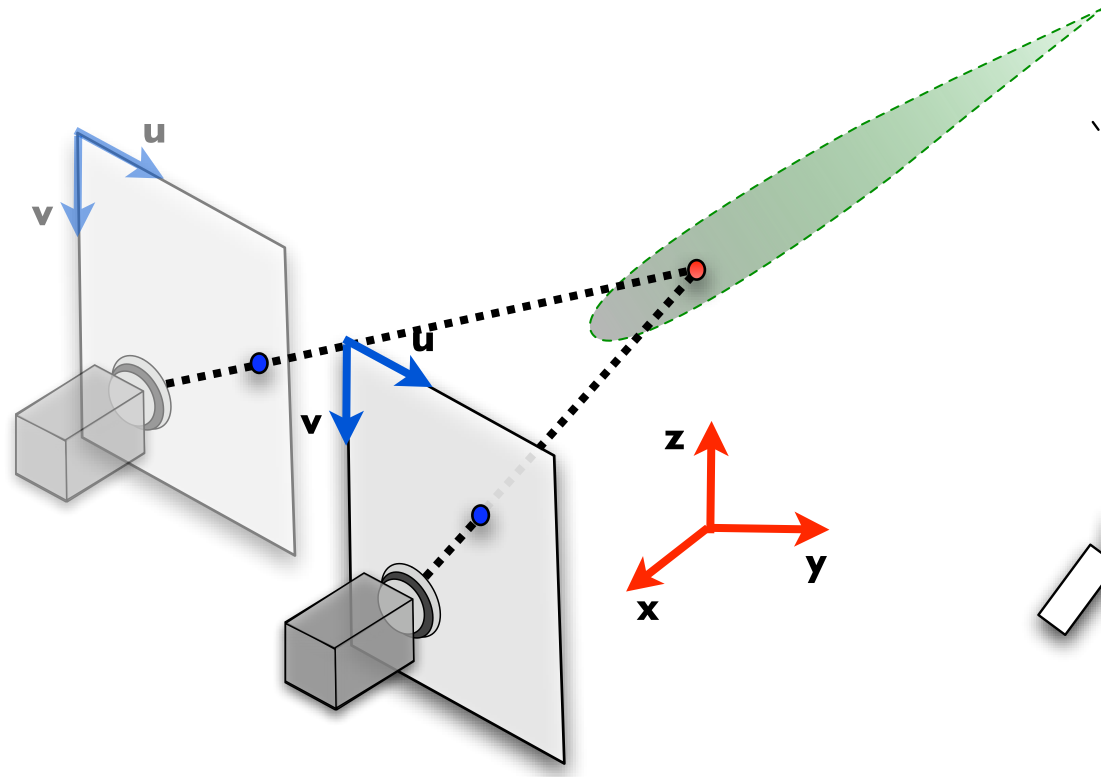
- Solutions:
  - S. Soatto et al “Structure from motion casually integrated over time” - IEEE PAMI 2002
  - A. Davison “Real-time Simultaneous Localization And Mapping with a Single Camera” - ICCV 2003, PAMI 2007
  - D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti “*On the Use of Inverse Scaling in Monocular SLAM*” - ICRA 2009 (Friday - 11:10 - room 401)

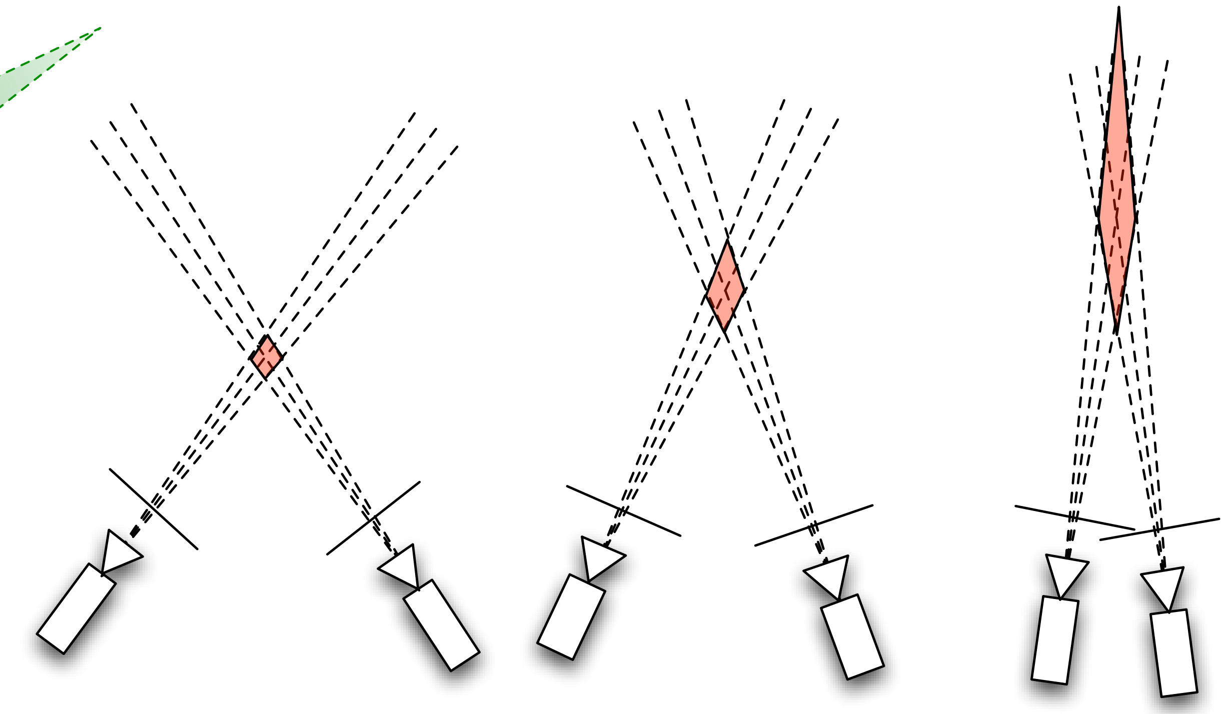
SLAM using Single

Rumore --> incertezza
Non modellizzabile con Gauss
EKF - unfeasible

Spostamento ignoto

- Why is this challenging?



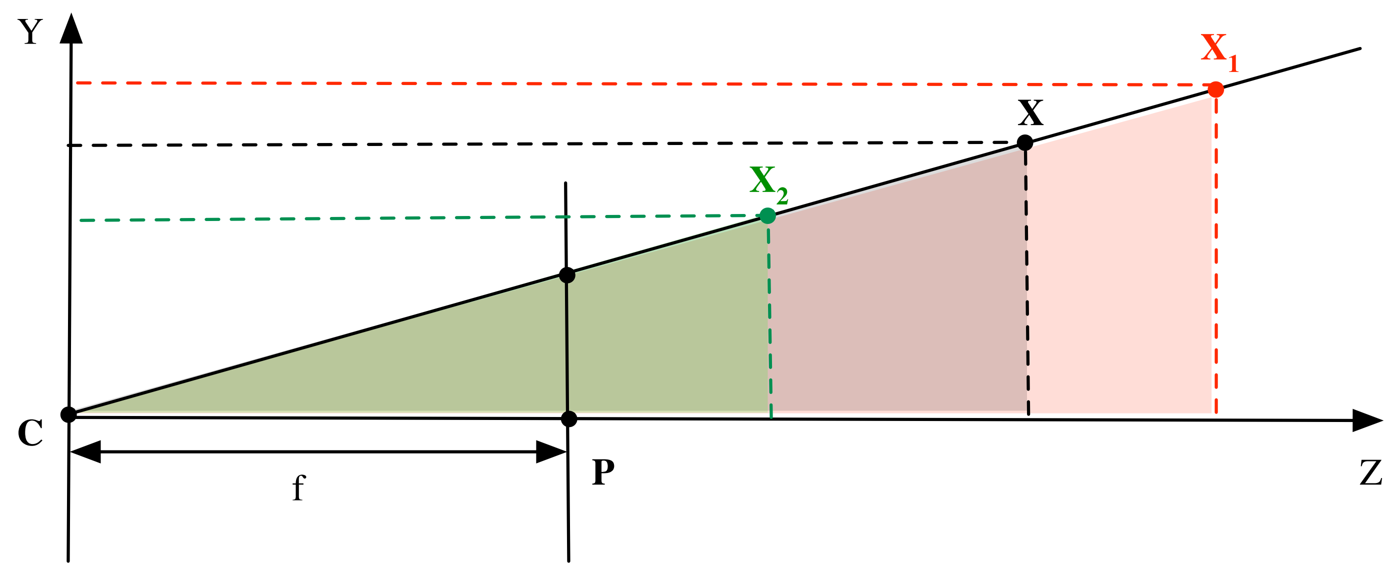


- Solutions:
  - S. Soatto et al "Structure from motion casually integrated over time" - IEEE PAMI 2002
  - A. Davison "Real-time Simultaneous Localization And Mapping with a Single Camera" - ICCV 2003, PAMI 2007
  - D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti "*On the Use of Inverse Scaling in Monocular SLAM*" - ICRA 2009 (Friday - 11:10 - room 401)

Inverse Scaling Parametrization

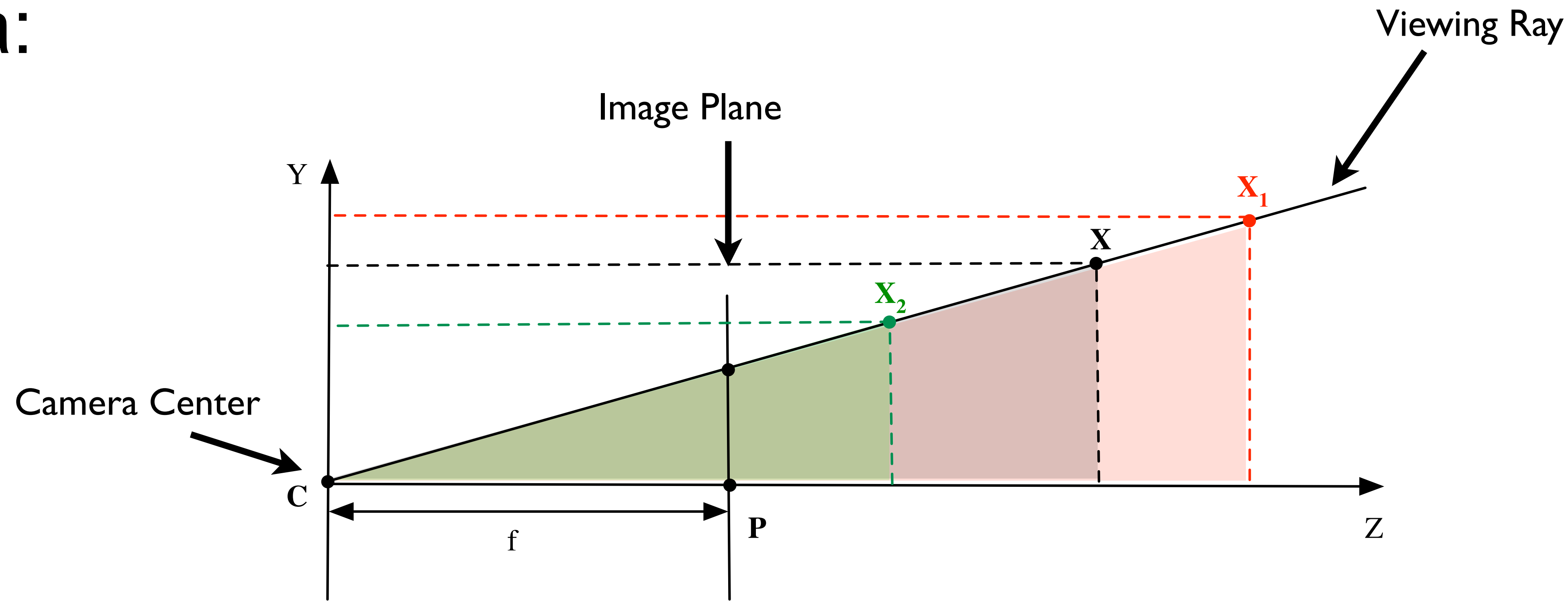
Inverse Scaling Parametrization

- Idea:



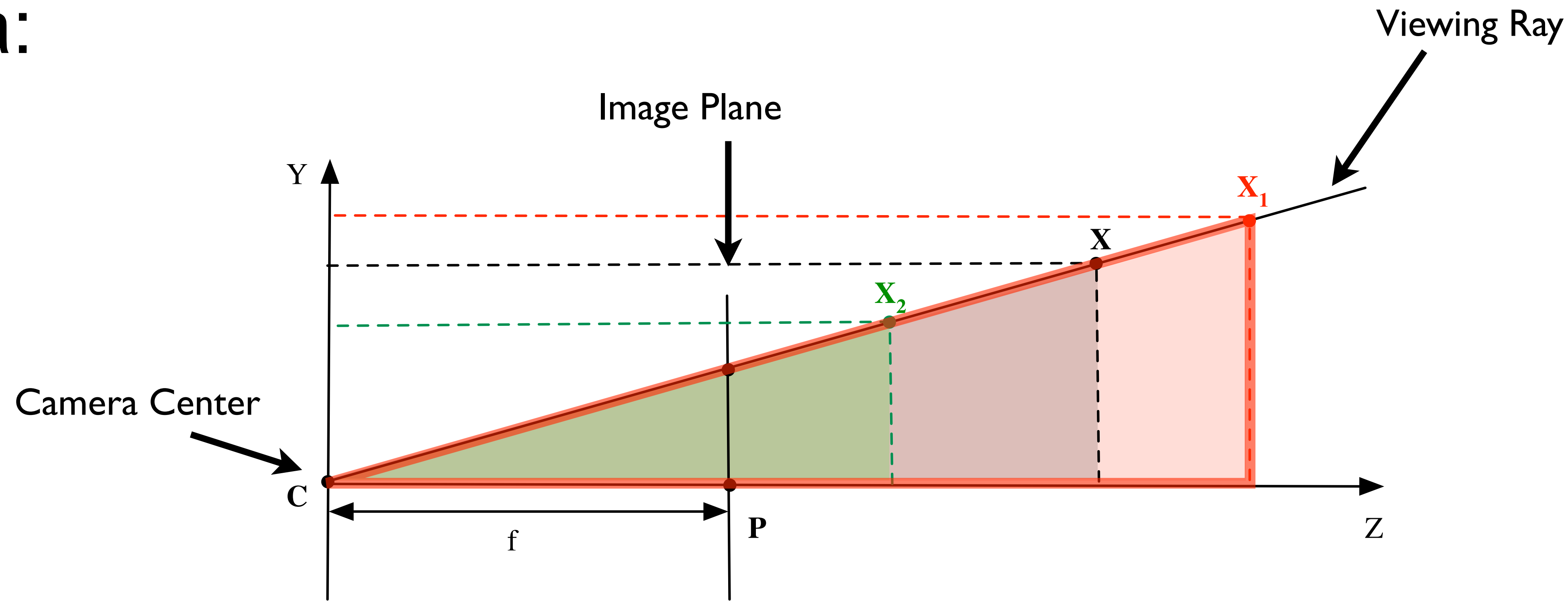
Inverse Scaling Parametrization

- Idea:



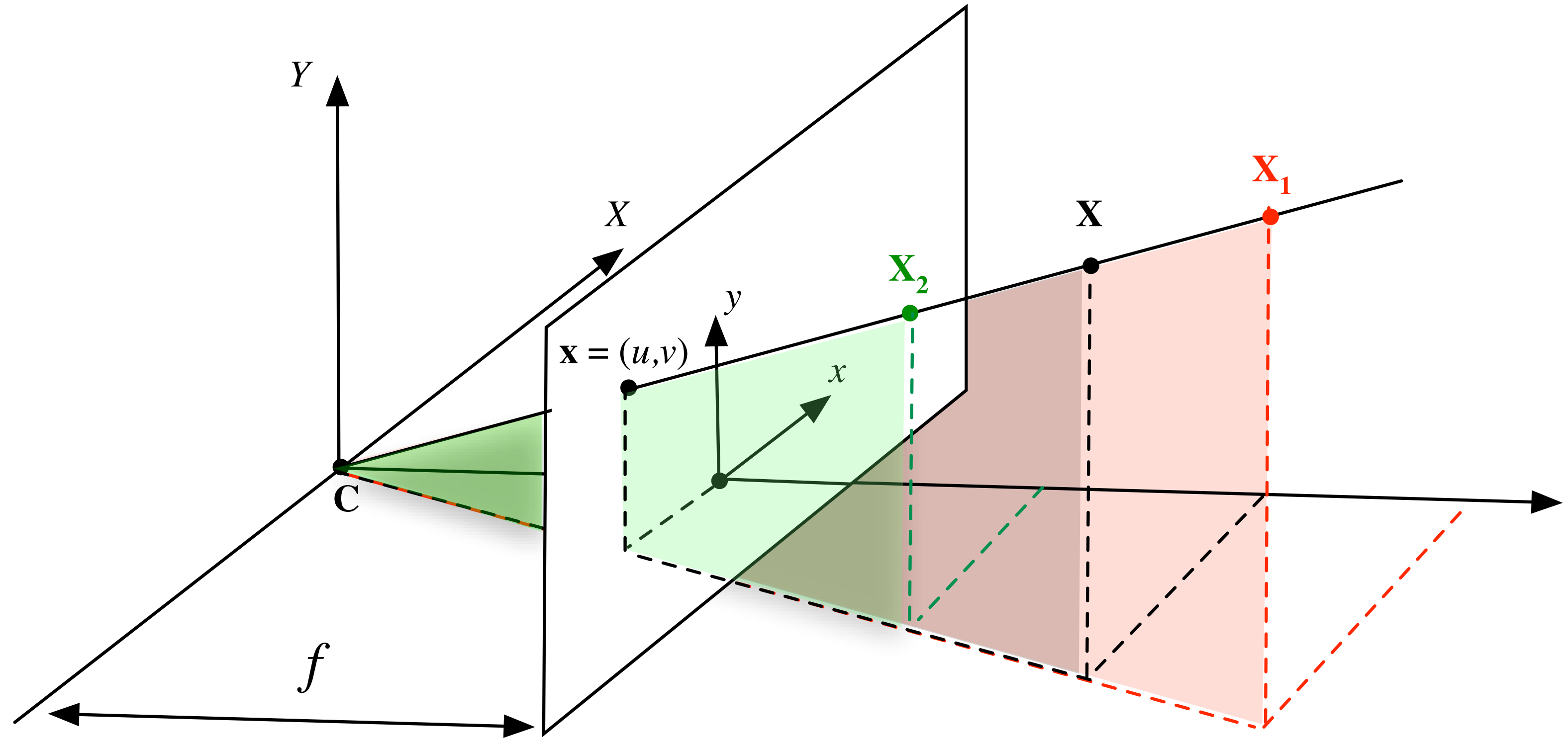
Inverse Scaling Parametrization

- Idea:



Inverse Scaling Parametrization

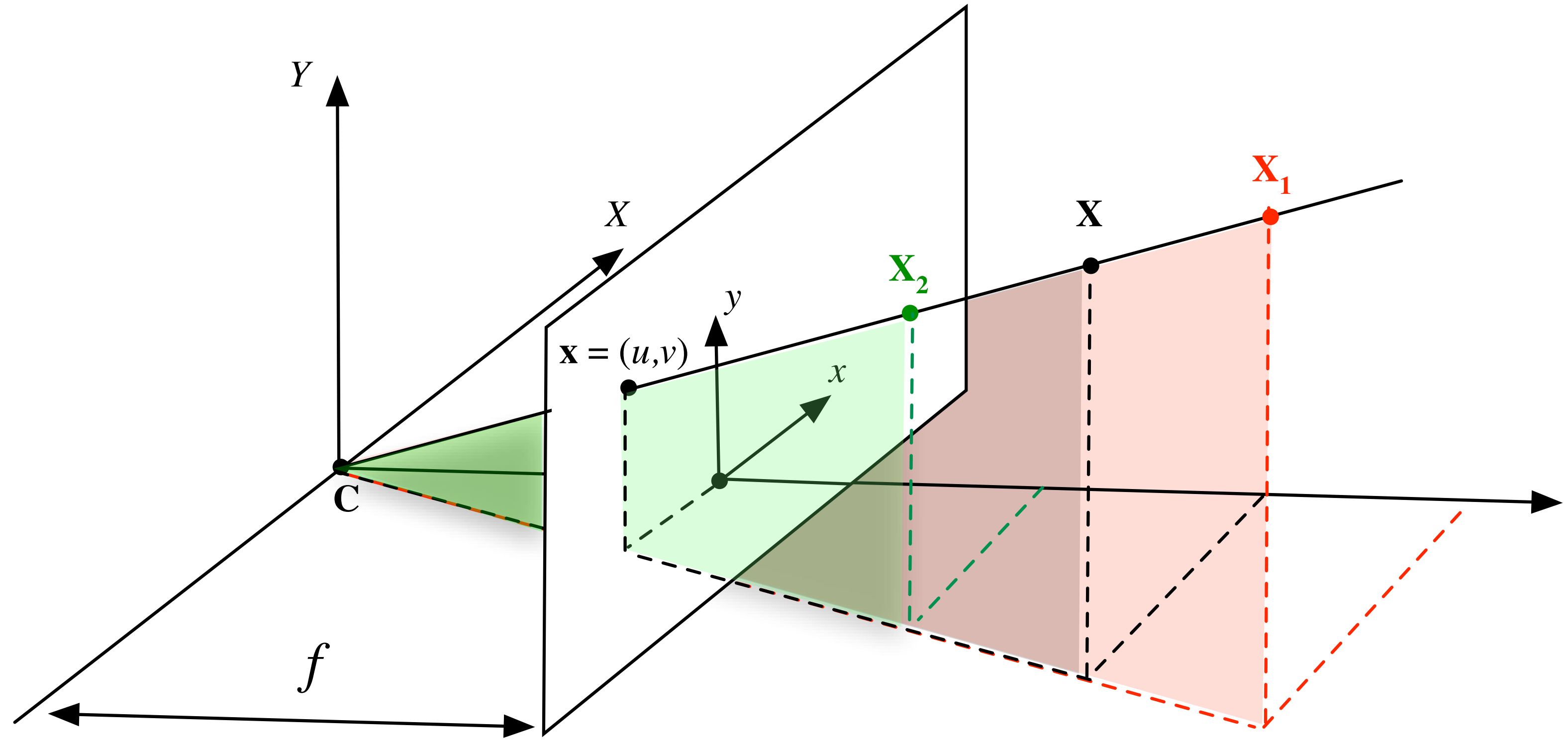
- Idea:



$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \alpha_1 \mathbf{X_1} = \begin{pmatrix} \alpha_1 X' \\ \alpha_1 Y' \\ \alpha_1 Z' \\ 1 \end{pmatrix} = \alpha_2 \mathbf{X_2} = \begin{pmatrix} \alpha_2 X'' \\ \alpha_2 Y'' \\ \alpha_2 Z'' \\ 1 \end{pmatrix}$$

Inverse Scaling Parametrization

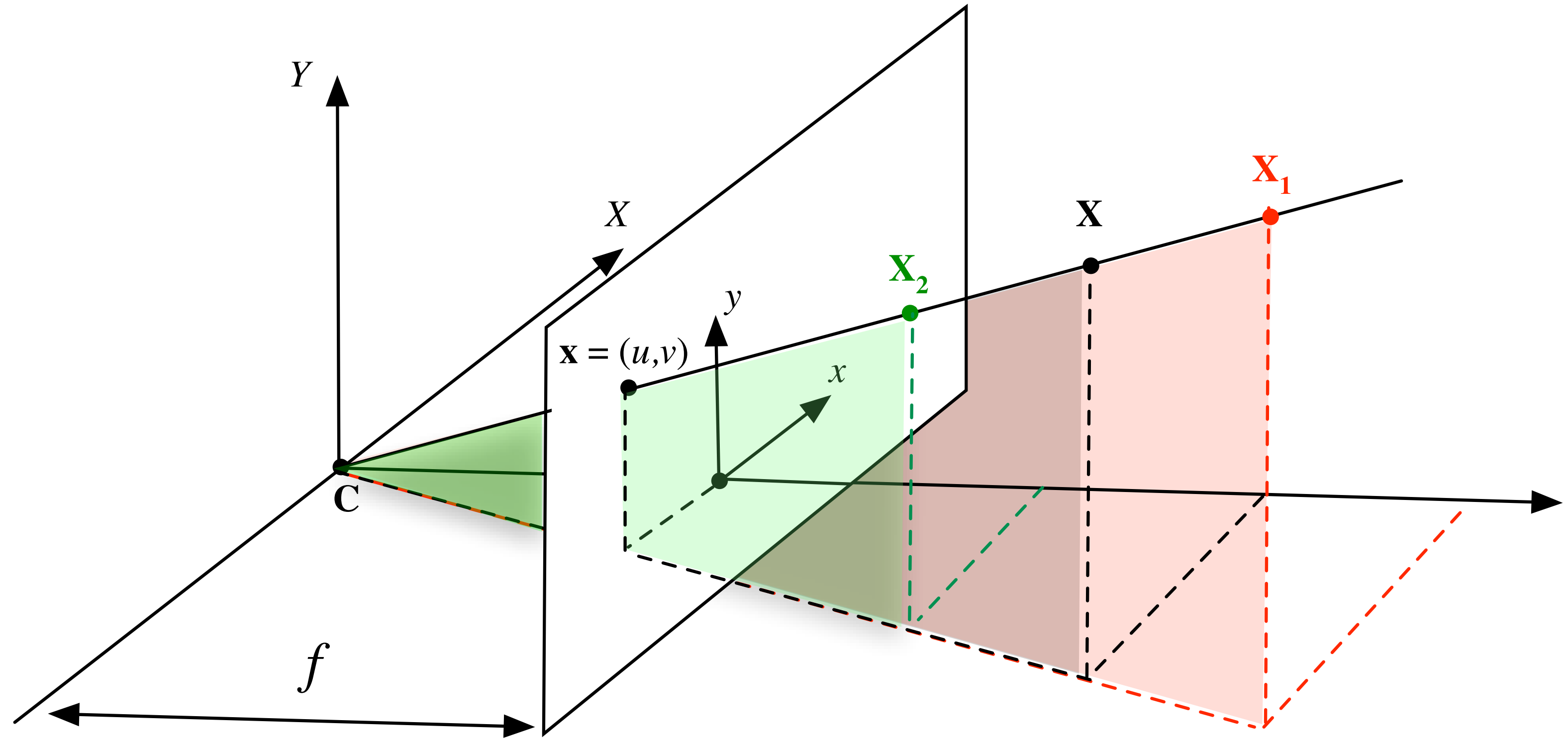
- Idea:



$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \alpha_1 \mathbf{X_1} = \begin{pmatrix} \alpha_1 X' \\ \alpha_1 Y' \\ \alpha_1 Z' \\ 1 \end{pmatrix} = \alpha_2 \mathbf{X_2} = \begin{pmatrix} \alpha_2 X'' \\ \alpha_2 Y'' \\ \alpha_2 Z'' \\ 1 \end{pmatrix} \rightarrow \mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \equiv \begin{pmatrix} X' \\ Y' \\ Z' \\ 1/\alpha_1 \end{pmatrix} \equiv \begin{pmatrix} X'' \\ Y'' \\ Z'' \\ 1/\alpha_2 \end{pmatrix}.$$

Inverse Scaling Parametrization

- Idea:



$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \alpha_1 \mathbf{X_1} = \begin{pmatrix} \alpha_1 X' \\ \alpha_1 Y' \\ \alpha_1 Z' \\ 1 \end{pmatrix} = \alpha_2 \mathbf{X_2} = \begin{pmatrix} \alpha_2 X'' \\ \alpha_2 Y'' \\ \alpha_2 Z'' \\ 1 \end{pmatrix} \rightarrow \mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \equiv \begin{pmatrix} X' \\ Y' \\ Z' \\ 1/\alpha_1 \end{pmatrix} \equiv \begin{pmatrix} X'' \\ Y'' \\ Z'' \\ 1/\alpha_2 \end{pmatrix}.$$

Undelayed Initialization

$$\mathbf{X} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \equiv \begin{pmatrix} u \\ v \\ f \\ 1/\alpha \end{pmatrix} \equiv \begin{pmatrix} u \\ v \\ f \\ \omega \end{pmatrix},$$

- D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti et al. "*Monocular SLAM with Inverse Scaling Parametrization*" - BMVC 2008

MonoSLAM with Inverse Scaling

MonoSLAM with Inverse Scaling

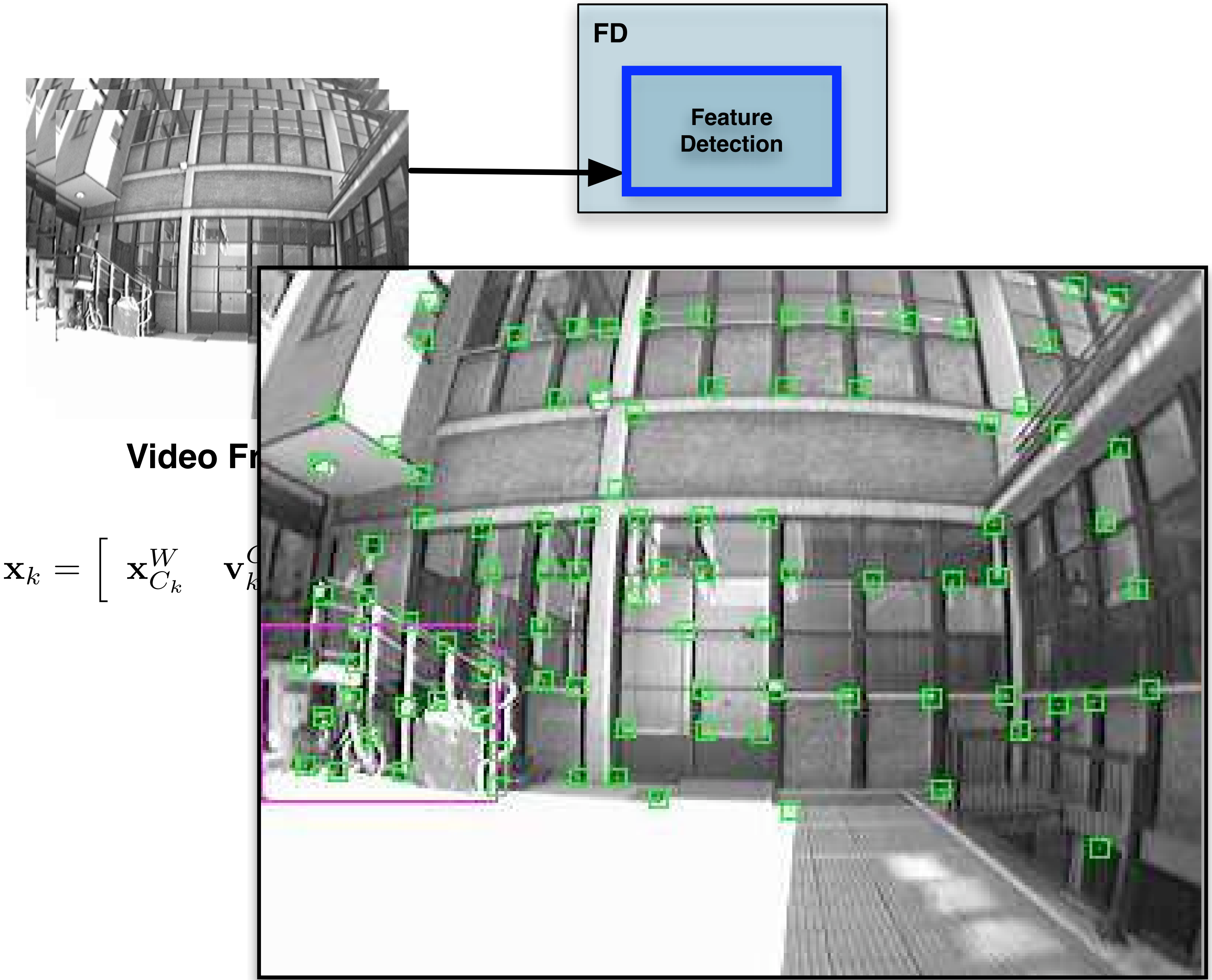
- Extended Kalman Filter

Video Frame

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{C_k}^W & \mathbf{v}_k^{C_k} & \mathbf{x}_{F_1\,k}^W & \dots & \mathbf{x}_{F_n\,k}^W & \dots & \mathbf{x}_{F_N\,k}^W \end{bmatrix}^T$$

MonoSLAM with Inverse Scaling

- Extended Kalman Filter

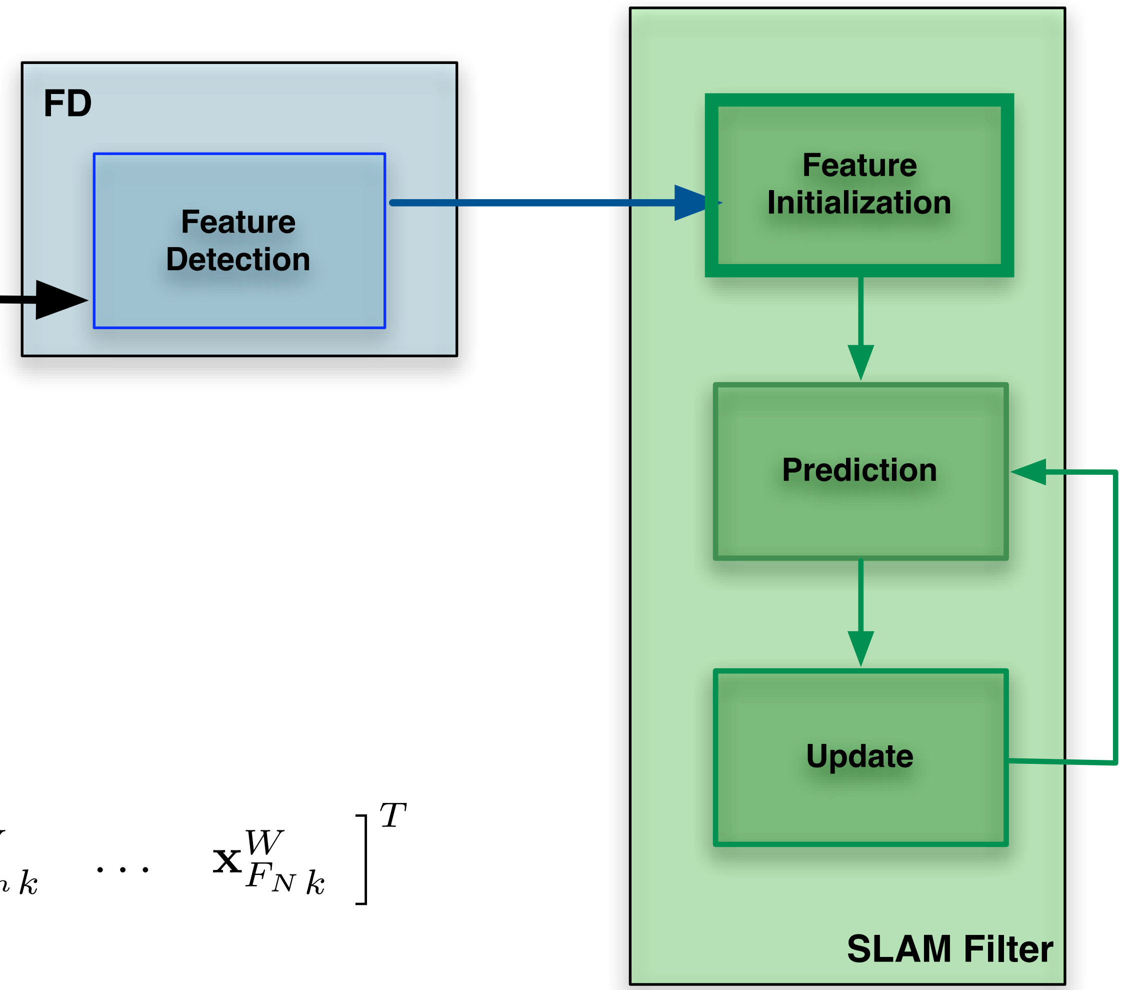


$\mathbf{x}_k = \Big[\ \mathbf{x}_{C_k}^W \quad \mathbf{v}_k^C$

MonoSLAM with Inverse Scaling

- Extended Kalman Filter

Video Frame

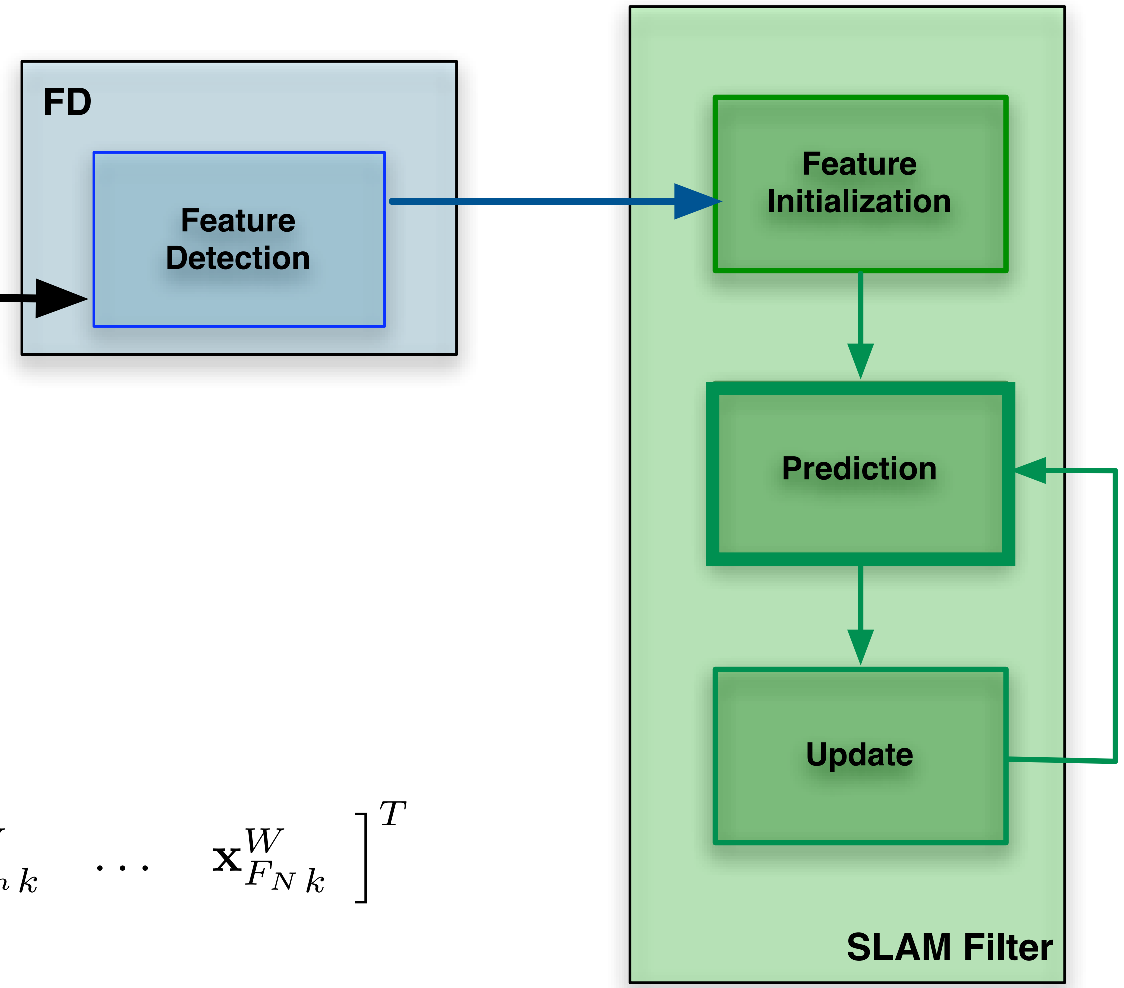


$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}^W_{C_k} & \mathbf{v}^{C_k}_k & \mathbf{x}^W_{F_1\,k} & \ldots & \mathbf{x}^W_{F_n\,k} & \ldots & \mathbf{x}^W_{F_N\,k} \end{bmatrix}^T$$

MonoSLAM with Inverse Scaling

- Extended Kalman Filter

Video Frame



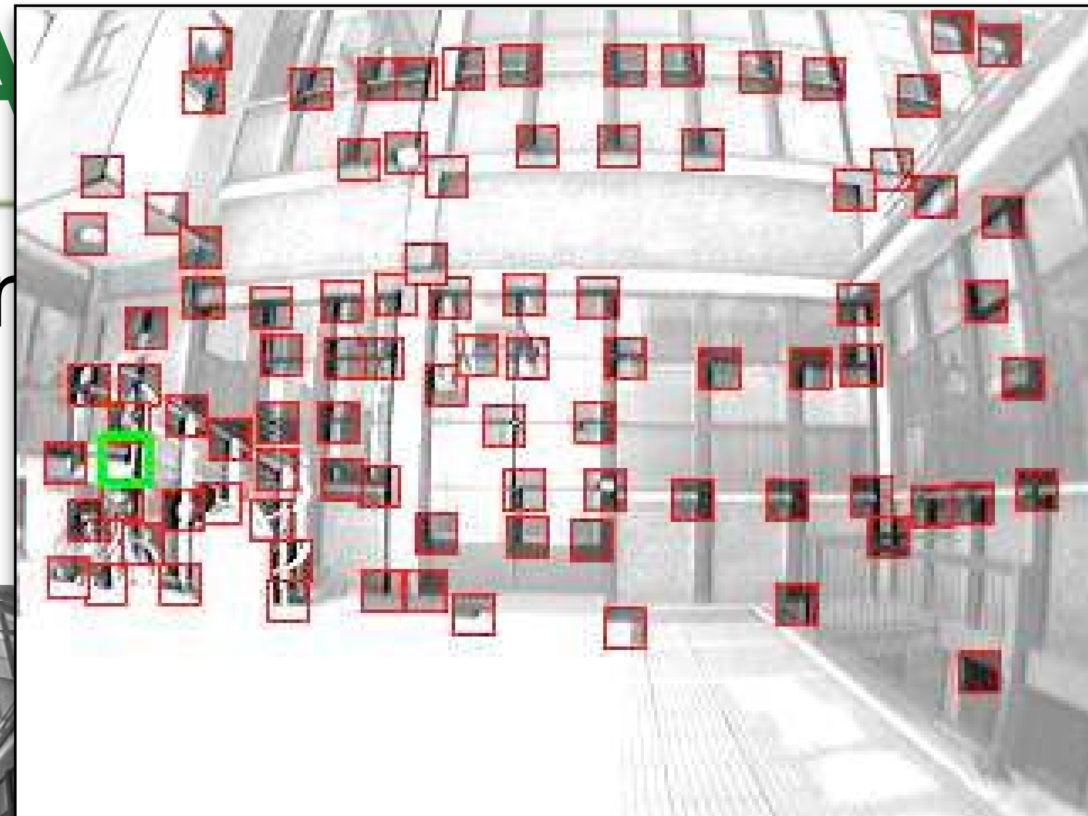
$$\mathbf{x}_k = \left[\begin{array}{ccccccc} \mathbf{x}^W_{C_k} & \mathbf{v}^{C_k}_k & \mathbf{x}^W_{F_1\,k} & \dots & \mathbf{x}^W_{F_n\,k} & \dots & \mathbf{x}^W_{F_N\,k} \end{array} \right]^T$$

$$\hat{\mathbf{x}}_k = \left[\begin{array}{c} \mathbf{x}^W_{C_{k-1}} \oplus \mathbf{x}^{C_{k-1}}_{C_k} \\ \mathbf{v}^{C_k}_k \\ \mathbf{x}^W_{F_1\,k-1} \\ \vdots \\ \mathbf{x}^W_{F_N\,k-1} \end{array} \right], \quad \begin{array}{rcl} \mathbf{v}^{C_k}_k & = & \mathbf{v}^{C_{k-1}}_{k-1} + \mathbf{a} \cdot \Delta t, \\ \mathbf{x}^{C_{k-1}}_{C_k} & = & \mathbf{v}^{C_{k-1}}_{k-1} \cdot \Delta t. \end{array}$$

$$\hat{\mathbf{P}}_k = \mathbf{J}_1 \mathbf{P}_{k-1} \mathbf{J}_1^T + \mathbf{J}_2 \mathbf{Q} \mathbf{J}_2^T$$

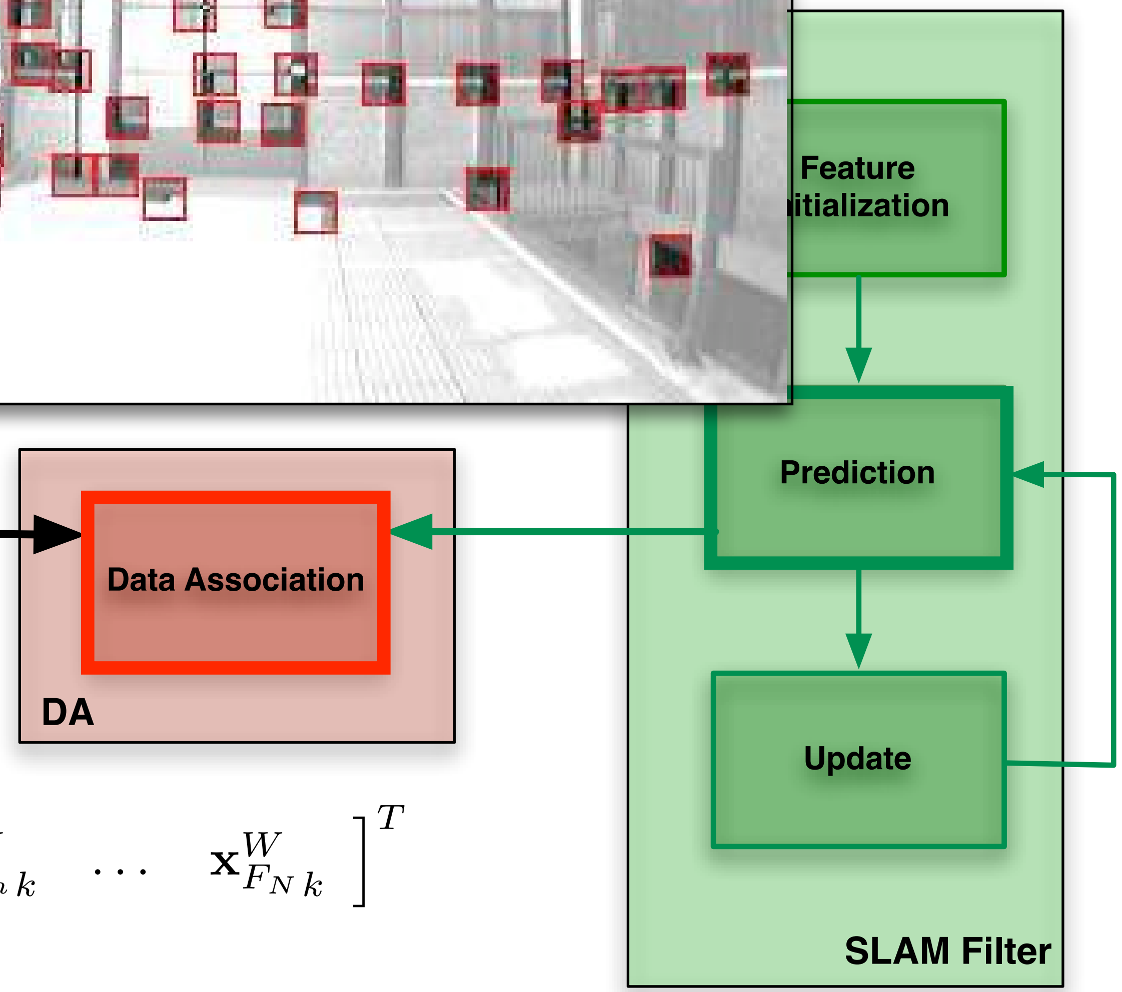
$$\mathbf{J}_1 = \left[\begin{array}{cccc} \mathbf{J}_x & \mathbf{J}_v & \dots & \mathbf{J}_{F_n} \end{array} \right], \; \mathbf{J}_2 = \left[\begin{array}{c} \mathbf{J}_{a_k} \end{array} \right]$$

MonoSLA … aling



- Extended Kal

Video Frame



$$\mathbf{x}_k = \left[\begin{array}{ccccccc} \mathbf{x}^W_{C_k} & \mathbf{v}^{C_k}_k & \mathbf{x}^W_{F_1\,k} & \dots & \mathbf{x}^W_{F_n\,k} & \dots & \mathbf{x}^W_{F_N\,k} \end{array}\right]^T$$

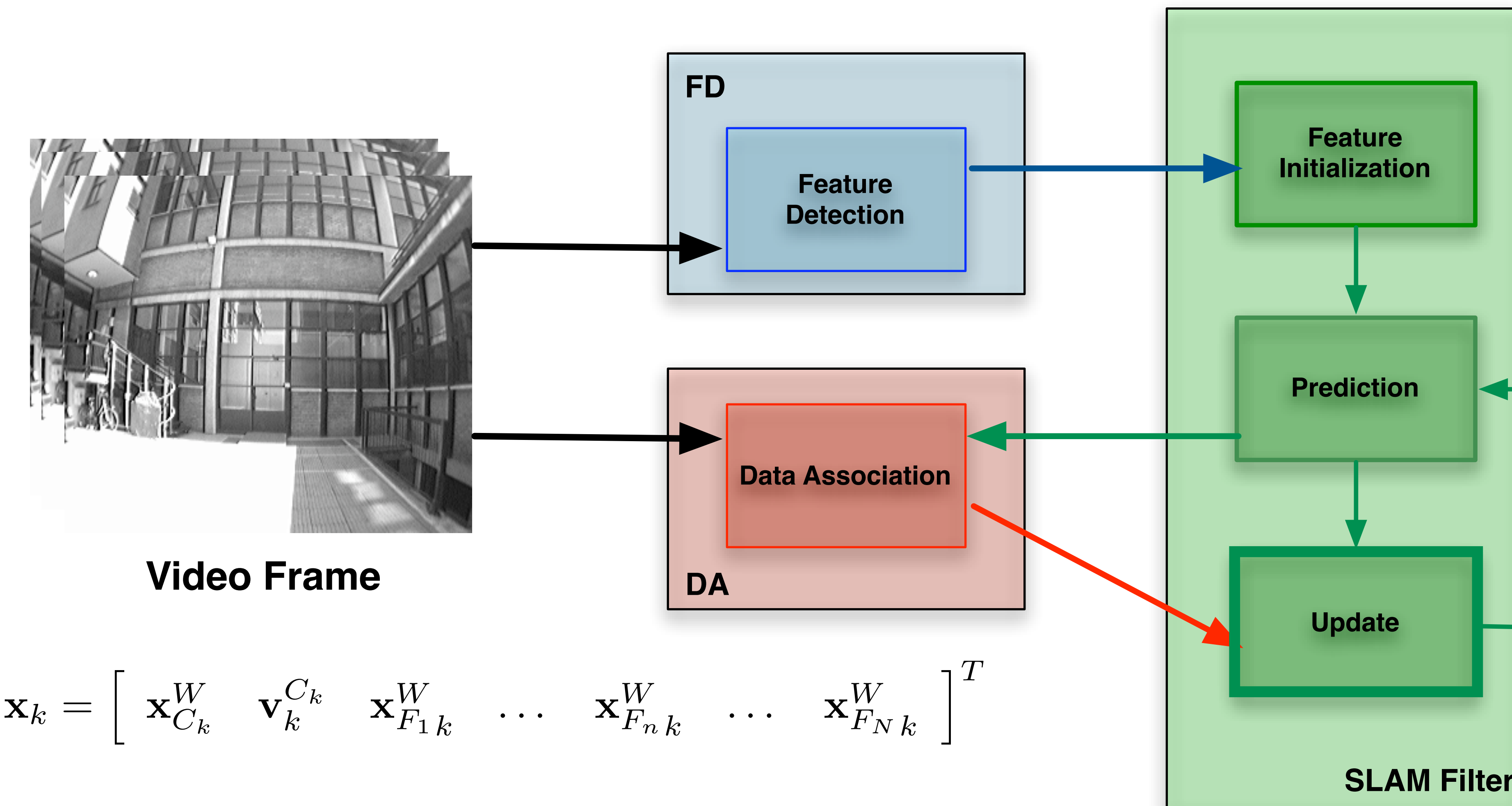
$$\hat{\mathbf{x}}_k = \left[\begin{array}{c} \mathbf{x}^W_{C_{k-1}} \oplus \mathbf{x}^{C_{k-1}}_{C_k} \\ \mathbf{v}^{C_k}_k \\ \mathbf{x}^W_{F_1\,k-1} \\ \vdots \\ \mathbf{x}^W_{F_N\,k-1} \end{array}\right], \quad \begin{array}{rcl} \mathbf{v}^{C_k}_k & = & \mathbf{v}^{C_{k-1}}_{k-1} + \mathbf{a}\cdot\Delta t, \\ \mathbf{x}^{C_{k-1}}_{C_k} & = & \mathbf{v}^{C_{k-1}}_{k-1}\cdot\Delta t. \end{array}$$

$$\hat{\mathbf{P}}_k = \mathbf{J}_1\mathbf{P}_{k-1}\mathbf{J}_1^T + \mathbf{J}_2\mathbf{Q}\mathbf{J}_2^T$$

$$\mathbf{J}_1 = \left[\begin{array}{ccccc} \mathbf{J}_x & \mathbf{J}_v & \dots & \mathbf{J}_{F_n} \end{array}\right], \; \mathbf{J}_2 = \left[\begin{array}{c} \mathbf{J}_{a_k} \end{array}\right]$$

MonoSLAM with Inverse Scaling

- Extended Kalman Filter



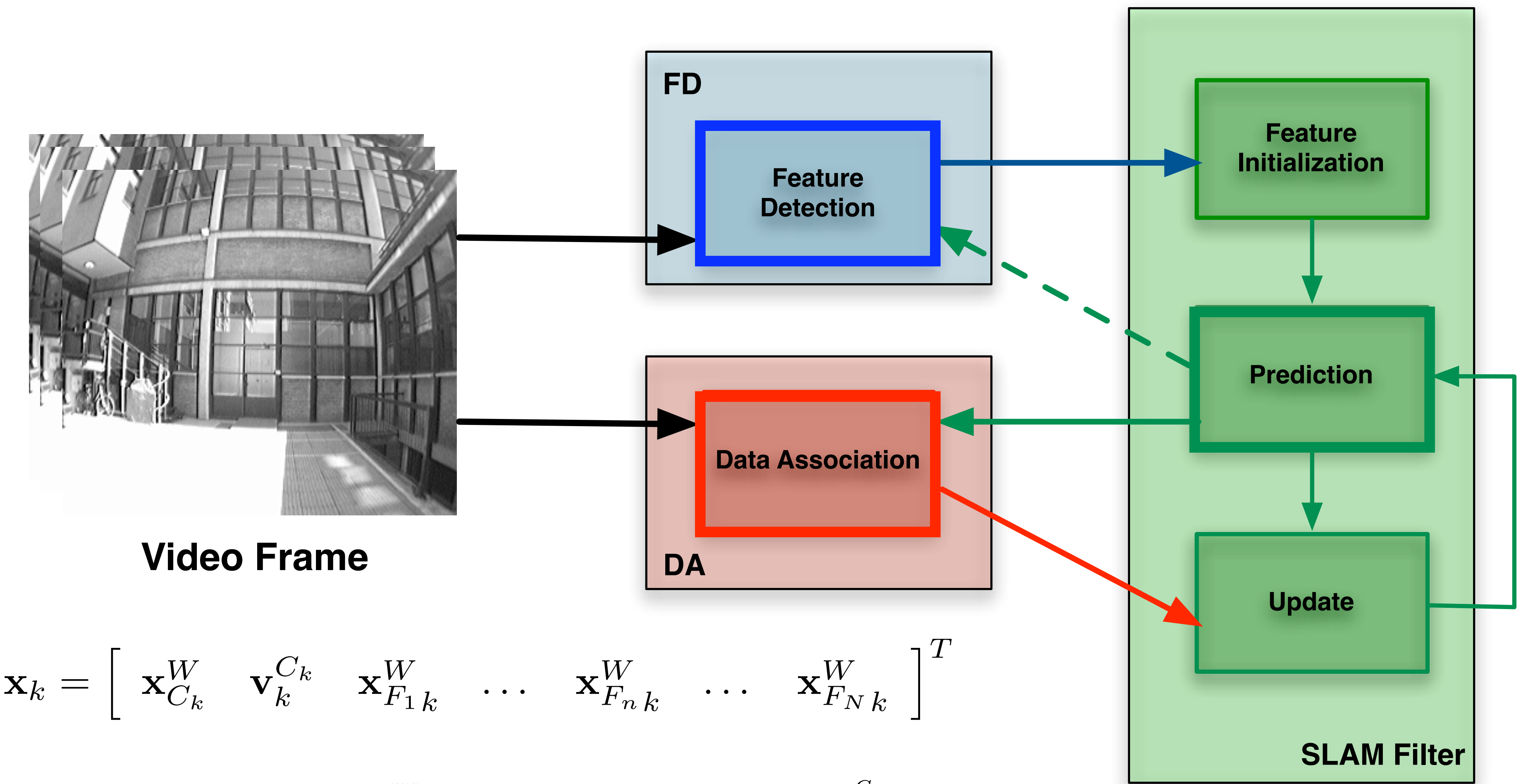
$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}^W_{C_k} & \mathbf{v}^{C_k}_k & \mathbf{x}^W_{F_1\,k} & \ldots & \mathbf{x}^W_{F_n\,k} & \ldots & \mathbf{x}^W_{F_N\,k} \end{bmatrix}^T$$

$$\mathbf{h}^{C_k}_n = \mathbf{M}\left(\mathbf{R}^{C_k}_W\left(\begin{bmatrix} x^W_{F_n} \\ y^W_{F_n} \\ z^W_{F_n} \end{bmatrix} - \omega^W_{F_n}\mathbf{r}^W_{C_k}\right)\right) \quad \mathbf{h}_{k,n} = \begin{bmatrix} \frac{h^{C_k}_{x,n}}{h^{C_k}_z} \\ \frac{h^{C_k}_{y,n}}{h^{C_k}_z} \end{bmatrix}.$$

$$\mathbf{S} = \mathbf{H}_k\hat{\mathbf{P}}_k\mathbf{H}^T_k + \mathbf{W}_k\mathbf{R}_k\mathbf{W}^T_k$$
$$\mathbf{K} = \hat{\mathbf{P}}_k\mathbf{H}^T_k\mathbf{S}^{-1}$$
$$\mathbf{P}_k = \hat{\mathbf{P}}_k - \mathbf{K}\mathbf{S}\mathbf{K}^T$$
$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{K}\left(\mathbf{z}_k - \mathbf{h}_k\right)$$

MonoSLAM with Inverse Scaling

- Extended Kalman Filter



$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}^W_{C_k} & \mathbf{v}^{C_k}_k & \mathbf{x}^W_{F_1\,k} & \dots & \mathbf{x}^W_{F_n\,k} & \dots & \mathbf{x}^W_{F_N\,k} \end{bmatrix}^T$$

$$\mathbf{h}^{C_k}_n = \mathbf{M}\left(\mathbf{R}^{C_k}_W\left(\begin{bmatrix} x^W_{F_n} \\ y^W_{F_n} \\ z^W_{F_n} \end{bmatrix} - \omega^W_{F_n}\mathbf{r}^W_{C_k}\right)\right) \quad \mathbf{h}_{k,n} = \begin{bmatrix} \frac{h^{C_k}_{x,n}}{h^{C_k}_z} \\ \frac{h^{C_k}_{y,n}}{h^{C_k}_z} \end{bmatrix}.$$

$$\mathbf{S} = \mathbf{H}_k\hat{\mathbf{P}}_k\mathbf{H}_k^T + \mathbf{W}_k\mathbf{R}_k\mathbf{W}_k^T$$
$$\mathbf{K} = \hat{\mathbf{P}}_k\mathbf{H}_k^T\mathbf{S}^{-1}$$
$$\mathbf{P}_k = \hat{\mathbf{P}}_k - \mathbf{K}\mathbf{S}\mathbf{K}^T$$
$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{K}\left(\mathbf{z}_k - \mathbf{h}_k\right)$$

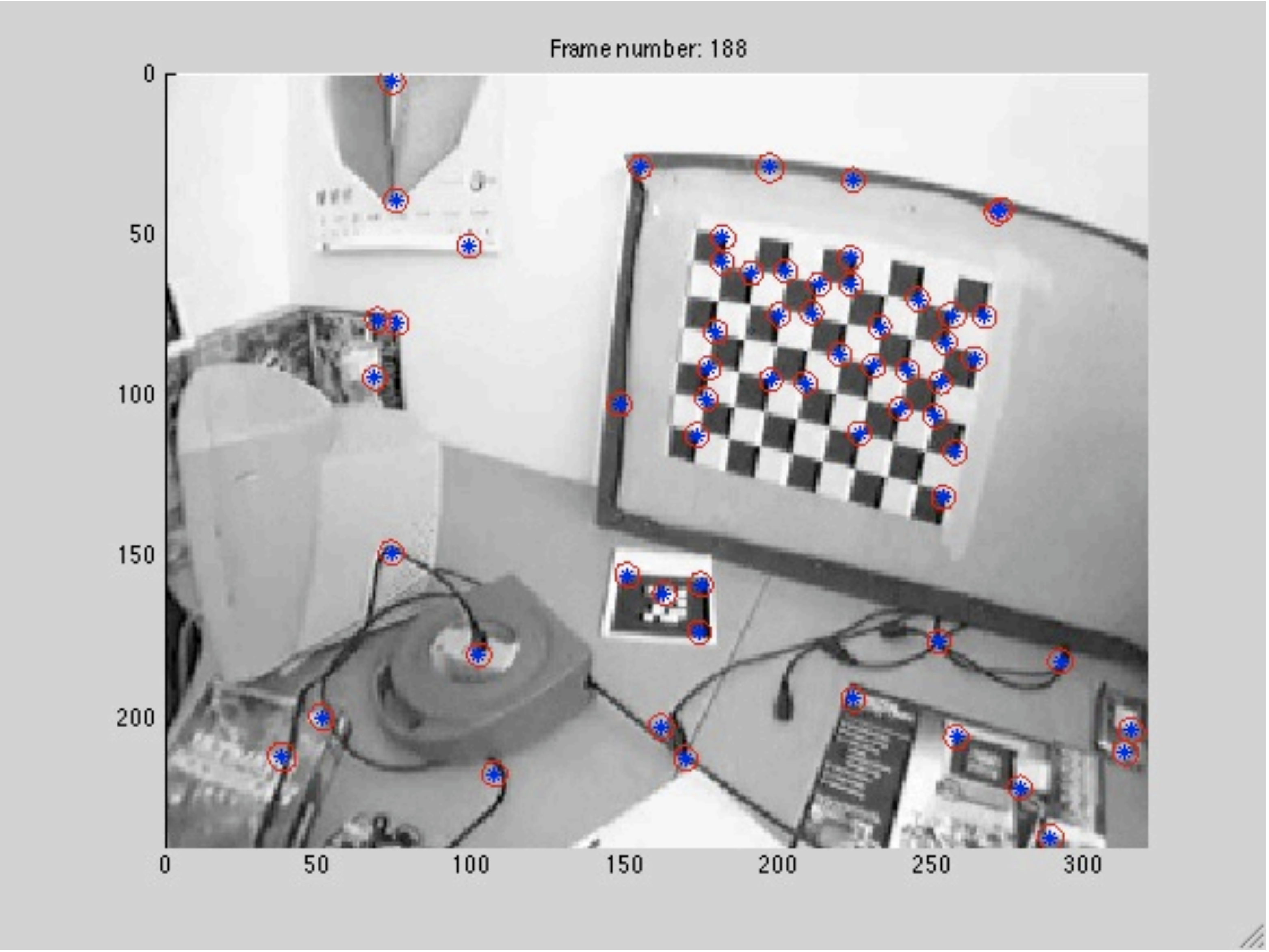
Experimental Results

Experimental Results

- Real Dataset

- D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti "*On the Use of Inverse Scaling in Monocular SLAM*" - ICRA 2009 (Friday - 11:10 - room 401)

SLAM in Dynamic Environments

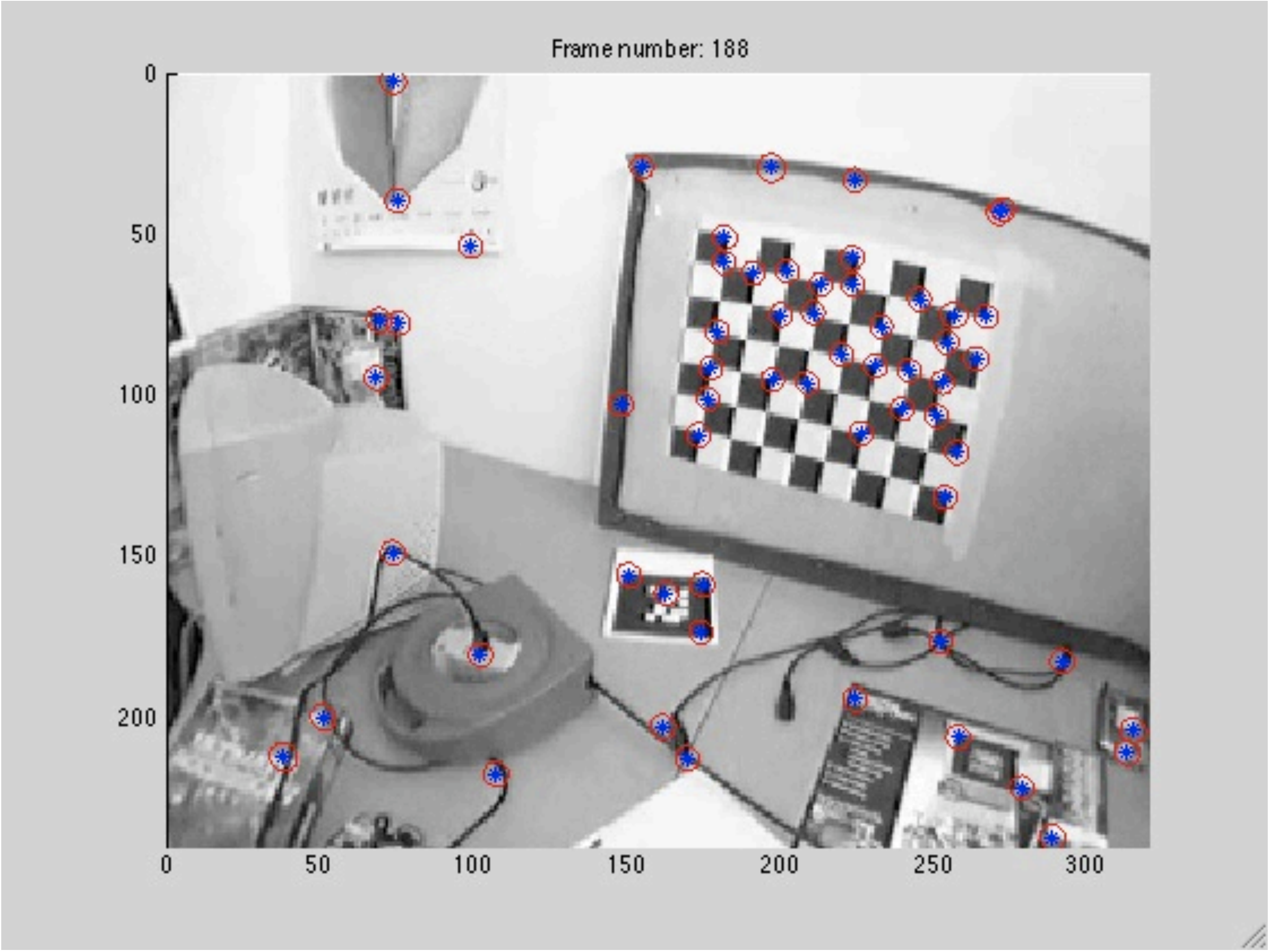


SLAM in Dynamic Environments

- Open issue:
  - Consistency of the estimates in scene containing moving objects

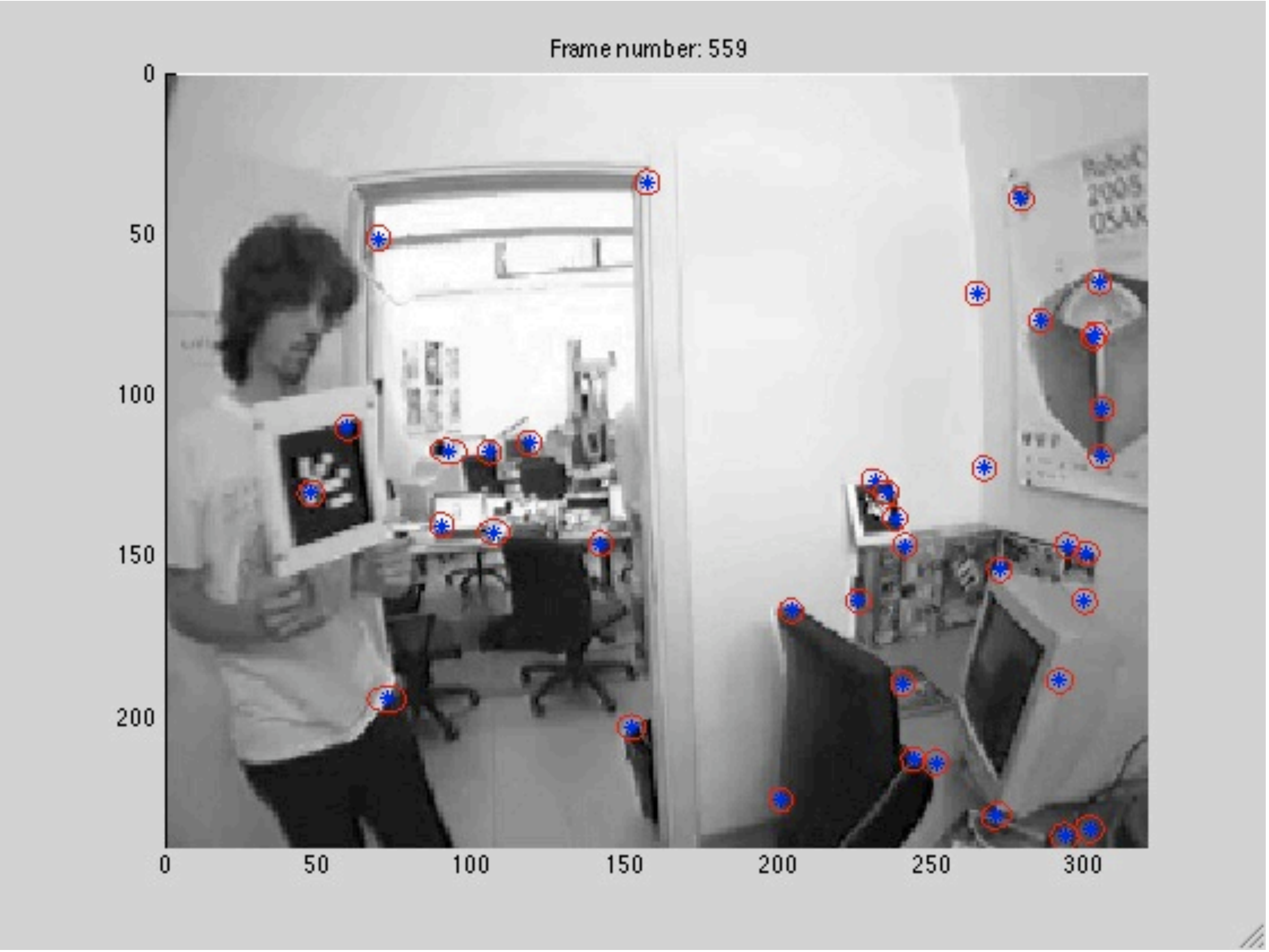
SLAM in Dynamic Environments

- Open issue:
  - Consistency of the estimates in scene containing moving objects



SLAM in Dynamic Environments

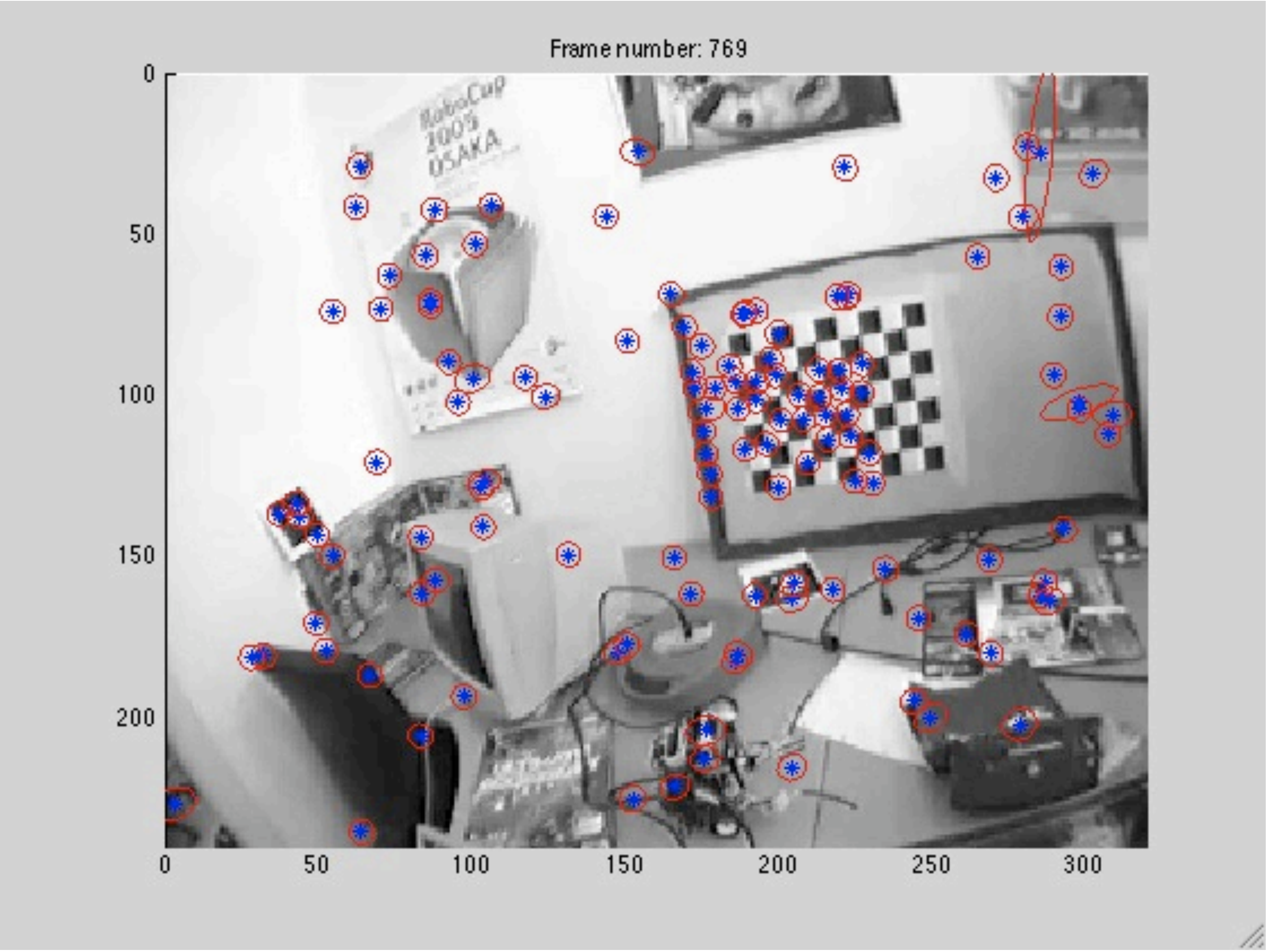
- Open issue:
  - Consistency of the estimates in scene containing moving objects



Davide Migliore - migliore@elet.polimi.it

SLAM in Dynamic Environments

- Open issue:
  - Consistency of the estimates in scene containing moving objects



SLAM in Dynamic Environments

SLAM in Dynamic Environments

- SLAM

$$\underbrace{p(\mathbf{x}_k, \mathbf{M} | \mathbf{Z}_k, \mathbf{U}_k)}_{\text{Posterior at time } k} \propto \underbrace{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{M})}_{\text{Update}} \underbrace{\int p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \underbrace{p(\mathbf{x}_{k-1}, \mathbf{M} | \mathbf{Z}_{k-1}, \mathbf{U}_{k-1})}_{\text{Posterior at time } k-1} d\mathbf{x}_{k-1}}_{\text{Prediction}}$$

SLAM in Dynamic Environments

SLAM and DATMO

$$\underbrace{p(\mathbf{x}_k, \mathbf{M}|\mathbf{Z}_k, \mathbf{U}_k)}_{\text{Posterior at time } k} \propto \underbrace{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{M})}_{\text{Update}} \underbrace{\int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)\, \underbrace{p(\mathbf{x}_{k-1}, \mathbf{M}|\mathbf{Z}_{k-1}, \mathbf{U}_{k-1})}_{\text{Posterior at time } k-1}\, d\mathbf{x}_{k-1}}_{\text{Prediction}}$$

$$\underbrace{p(\mathbf{x}_k, \mathbf{M}, \mathbf{O}_k|\mathbf{Z}_k, \mathbf{U}_k)}_{\text{Posterior at time } k} \propto$$

$$\propto \underbrace{p(\mathbf{z}_k|\mathbf{O}_k, \mathbf{x}_k)}_{\text{Update}} \underbrace{\int p(\mathbf{O}_k|\mathbf{O}_{k-1})\, \underbrace{p(\mathbf{O}_{k-1}|\mathbf{Z}_{k-1}, \mathbf{U}_{k-1})}_{\text{Posterior at time } k-1}\, d\mathbf{O}_{k-1}}_{\text{Prediction}} \longleftarrow \text{BOT}$$

$$\underbrace{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{M})}_{\text{Update}} \underbrace{\int p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k)\, \underbrace{p(\mathbf{x}_{k-1}, \mathbf{M}|\mathbf{Z}_{k-1}, \mathbf{U}_{k-1})}_{\text{Posterior at time } k-1}\, d\mathbf{x}_{k-1}}_{\text{Prediction}} \longleftarrow \text{SLAM}$$

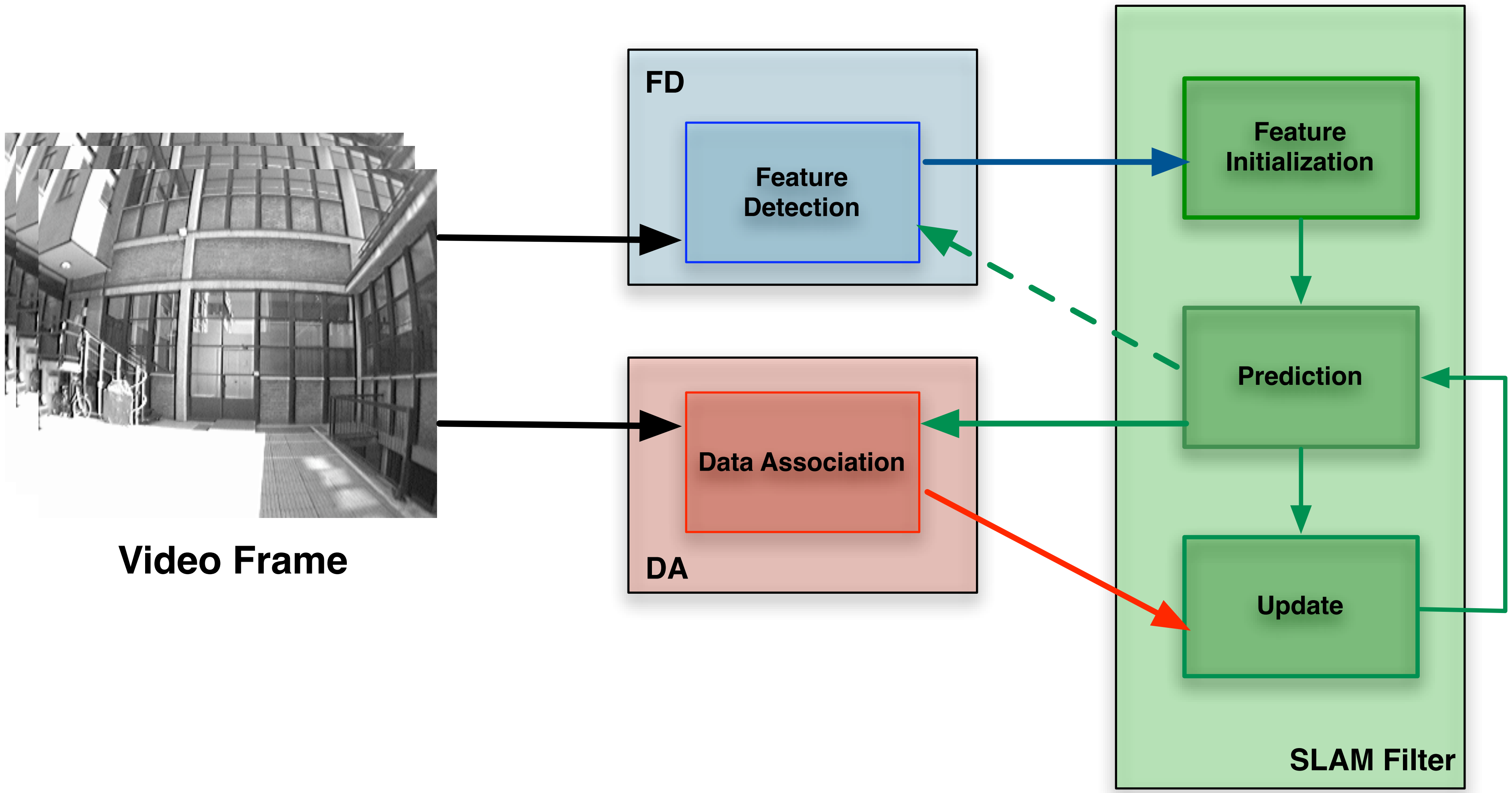
- C.-C. Wang, C. Thorpe and S. Thrun. *Online Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects: Theory and Results from a Ground Vehicle in Crowded Urban Areas*. ICRA 2003.

MonoSLAM with BOT

- Extended Kalman Filter

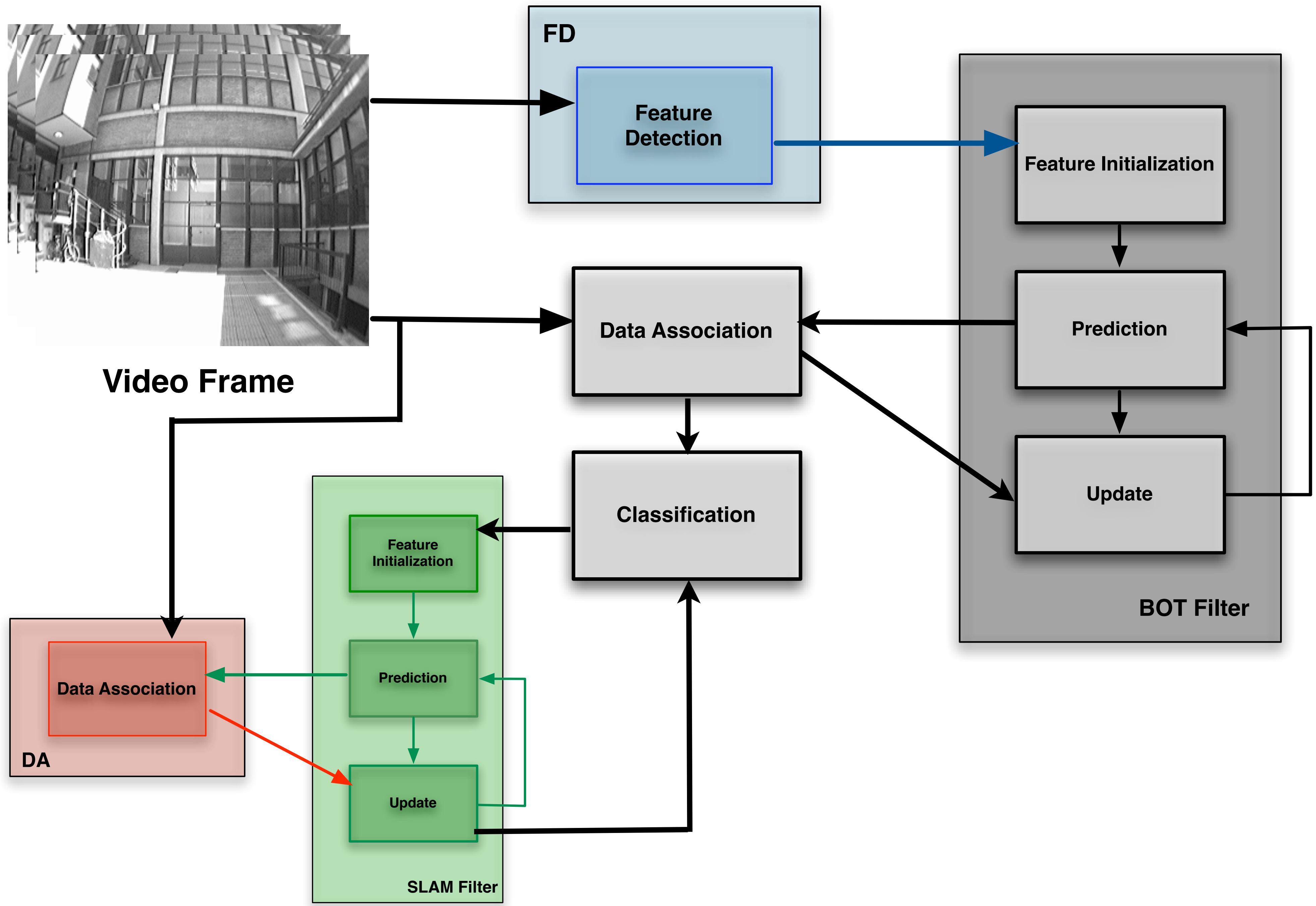
MonoSLAM with BOT

- Extended Kalman Filter

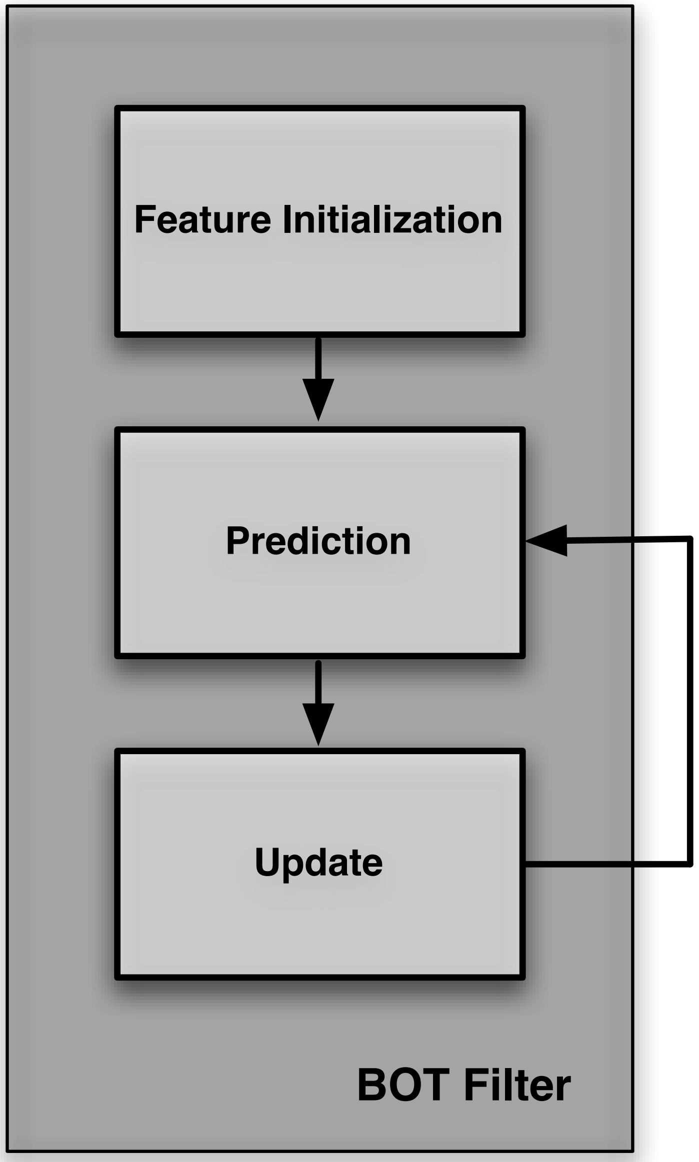


MonoSLAM with BOT

- Extended Kalman Filter



Bearing Only Tracking



Bearing Only Tracking

- Again Inverse Scaling Parametrization
  - EKF BOT (Robocentric):
    - *State*

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_{F_k}^{C_k} \\ \mathbf{v}_{F_k}^{F_k} \end{bmatrix}$$

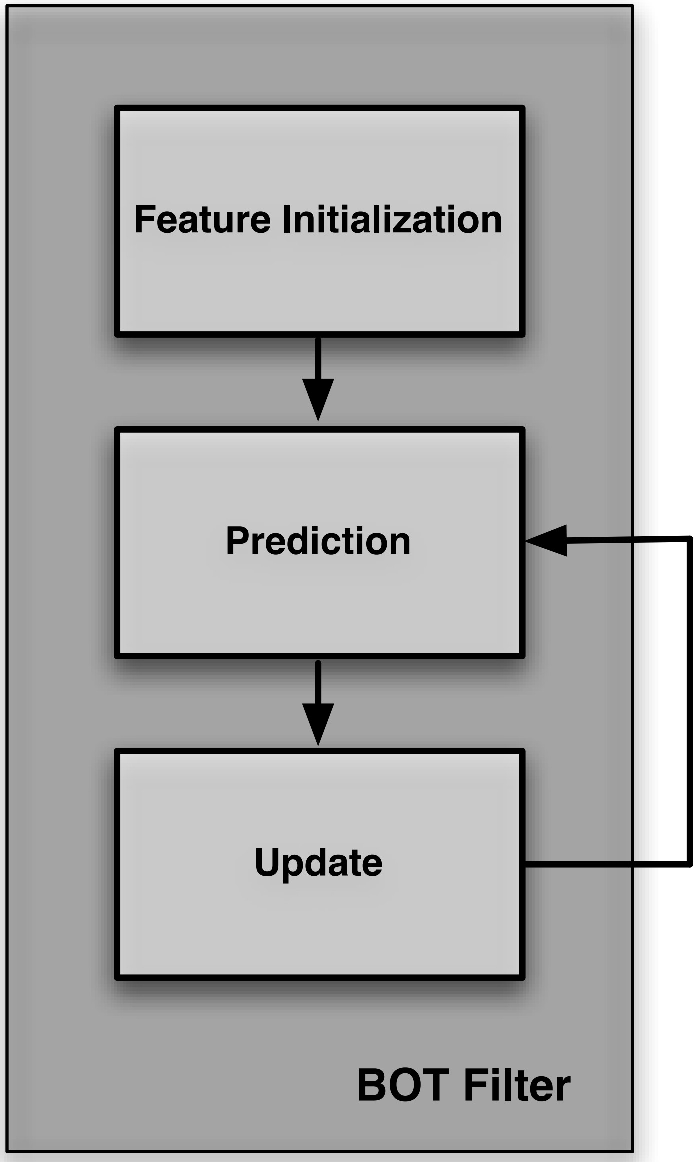
- *Motion Model*

$$\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_{F_{k+1}}^{C_{k+1}} \\ \mathbf{v}_{F_{k+1}}^{F_{k+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{C_k}^{C_{k+1}} \oplus \mathbf{x}_{F_k}^{C_k} \oplus (\mathbf{v}_{F_{k+1}}^{F_k} \Delta t) \\ \mathbf{x}_{F_k}^{F_{k+1}} \oplus \mathbf{v}_{F_k}^{F_k} \end{bmatrix}$$

- *Measurement Model*

$$\mathbf{h}_k = \begin{bmatrix} h_{k_x} \\ h_{k_y} \\ h_{k_z} \end{bmatrix} = \mathbf{M}\mathbf{x}_{F_k}^{C_k} \qquad \mathbf{M} = \begin{bmatrix} f_{c_x} & 0 & cc_x \\ 0 & f_{c_y} & cc_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{h}_k = \begin{bmatrix} h_{k_u} \\ h_{k_v} \end{bmatrix} = \begin{bmatrix} h_{k_x}/h_{k_z} \\ h_{k_y}/h_{k_z} \end{bmatrix}$$



Bearing Only Tracker Observability

Bearing Only Tracker Observability

- Linear motion model (w.r.t. the Shadow filter state)

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{RT}_{C_k}^{C_{k+1}} & \mathbf{R}_{C_k}^{C_{k+1}} dt \\ \mathbf{0} & \mathbf{R}_{C_k}^{C_{k+1}} \end{bmatrix},$$

Bearing Only Tracker Observability

- Linear motion model (w.r.t. the Shadow filter state)

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{RT}_{C_k}^{C_{k+1}} & \mathbf{R}_{C_k}^{C_{k+1}} dt \\ \mathbf{0} & \mathbf{R}_{C_k}^{C_{k+1}} \end{bmatrix},$$

- Non linear measurement model... but

$$\begin{bmatrix} h_{k_x} \\ h_{k_y} \\ h_{k_z} \end{bmatrix} = \mathbf{Ix}_{F_k}^{C_k} \qquad \mathbf{h}_k = \begin{bmatrix} h_{k_u} \\ h_{k_v} \end{bmatrix} \begin{bmatrix} h_{k_x}/h_{k_z} \\ h_{k_y}/h_{k_z} \end{bmatrix} \qquad \begin{array}{l} 0 = h_{k_x} - h_{k_u} h_{k_z}. \\ 0 = h_{k_y} - h_{k_v} h_{k_z}. \end{array}$$

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & -h_{k_u} & 0 & 0 \\ 0 & 1 & -h_{k_v} & 0 & 0 \end{bmatrix}.$$

Bearing Only Tracker Observability

- Linear motion model (w.r.t. the Shadow filter state)

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{RT}_{C_k}^{C_{k+1}} & \mathbf{R}_{C_k}^{C_{k+1}} dt \\ \mathbf{0} & \mathbf{R}_{C_k}^{C_{k+1}} \end{bmatrix},$$

- Non linear measurement model... but

$$\begin{bmatrix} h_{k_x} \\ h_{k_y} \\ h_{k_z} \end{bmatrix} = \mathbf{Ix}_{F_k}^{C_k} \qquad \mathbf{h}_k = \begin{bmatrix} h_{k_u} \\ h_{k_v} \end{bmatrix} \begin{bmatrix} h_{k_x}/h_{k_z} \\ h_{k_y}/h_{k_z} \end{bmatrix} \qquad \begin{matrix} 0 = h_{k_x} - h_{k_u} h_{k_z}. \\ 0 = h_{k_y} - h_{k_v} h_{k_z}. \end{matrix}$$

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & -h_{k_u} & 0 & 0 \\ 0 & 1 & -h_{k_v} & 0 & 0 \end{bmatrix}.$$

- How to check the observability

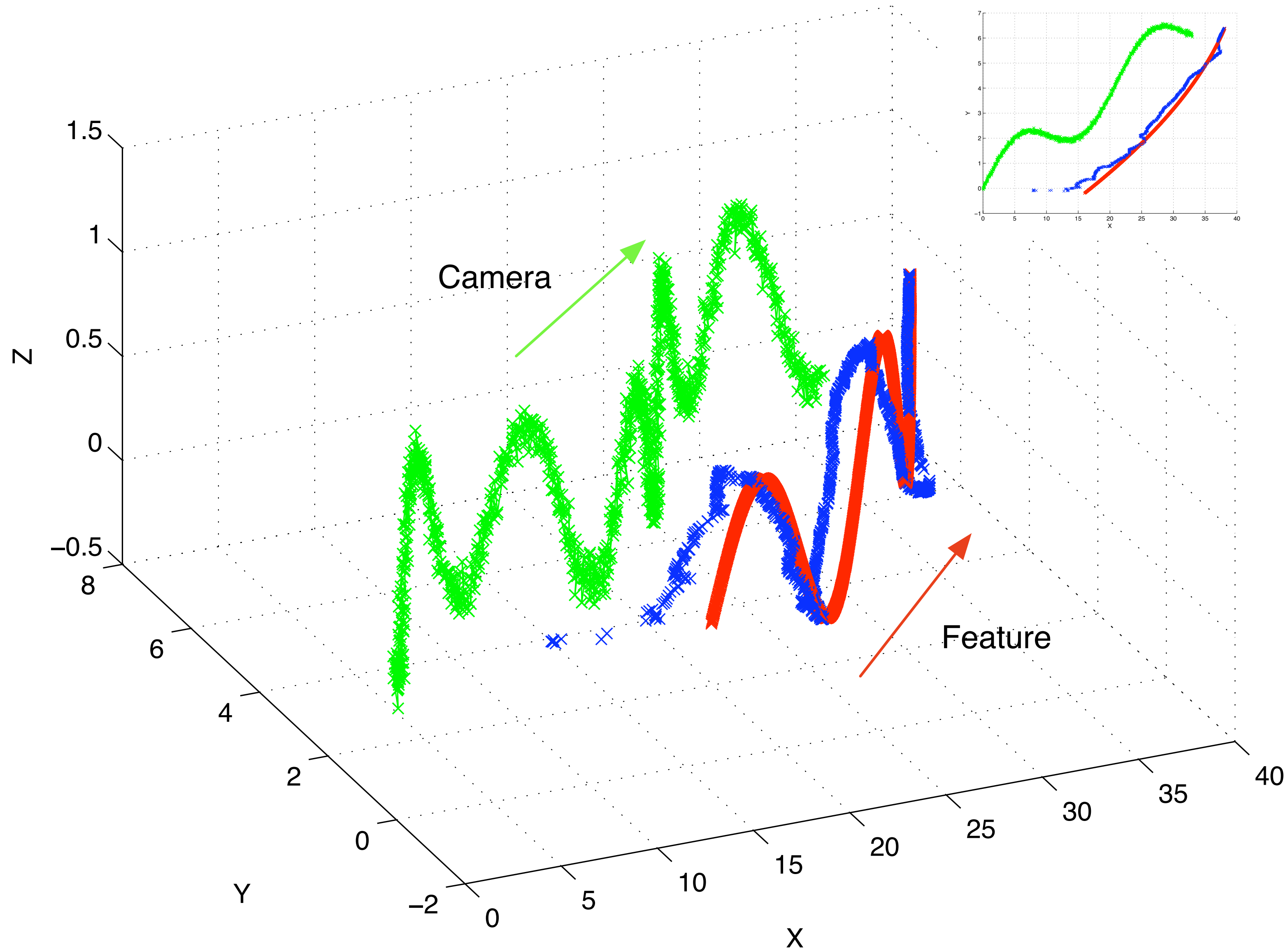
$$\begin{cases} z_0 = \mathbf{H}_0 \mathbf{X}_0 \\ z_1 = \mathbf{H}_1 \mathbf{F}_1 \mathbf{X}_0 \\ z_2 = \mathbf{H}_2 \mathbf{F}_2 \mathbf{F}_1 \mathbf{X}_0 \\ \dots \\ z_k = \mathbf{H}_k \mathbf{F}_k \dots \mathbf{F}_2 \mathbf{F}_1 \mathbf{X}_0 \end{cases} \qquad \mathbf{O}_k = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \mathbf{F}_1 \\ \mathbf{H}_2 \mathbf{F}_2 \mathbf{F}_1 \\ \vdots \\ \mathbf{H}_k \mathbf{F}_k \dots \mathbf{F}_2 \mathbf{F}_1 \end{bmatrix}$$

  - Under the assumption of random camera movements (hand shacking) and noisy measurements the observability is possible

Bearing Only Tracker Observability

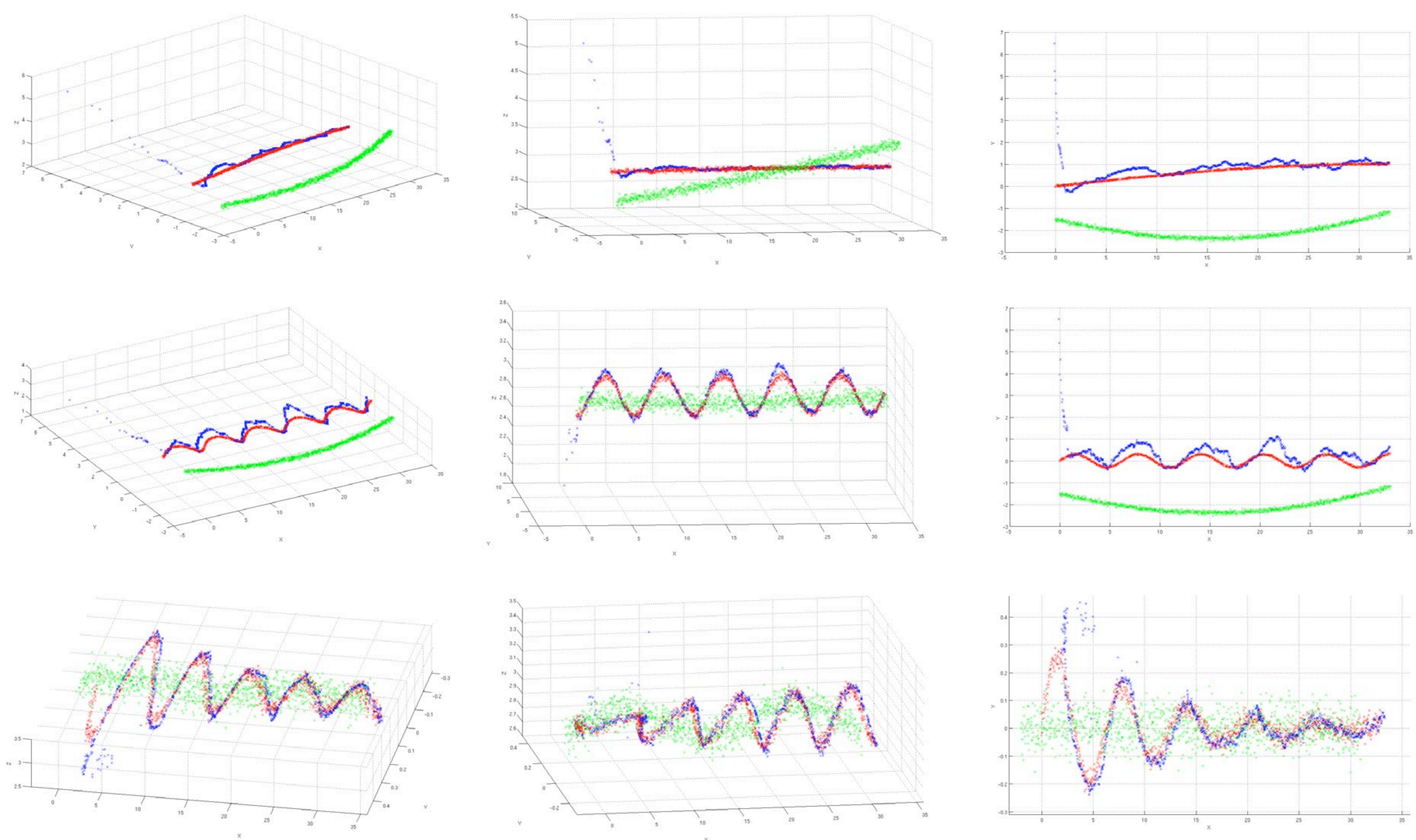
Bearing Only Tracker Observability

- Simulated Results (camera position is given)



Bearing Only Tracker Observability

- Simulated Results (camera position is given)



Davide Migliore - migliore@elet.polimi.it

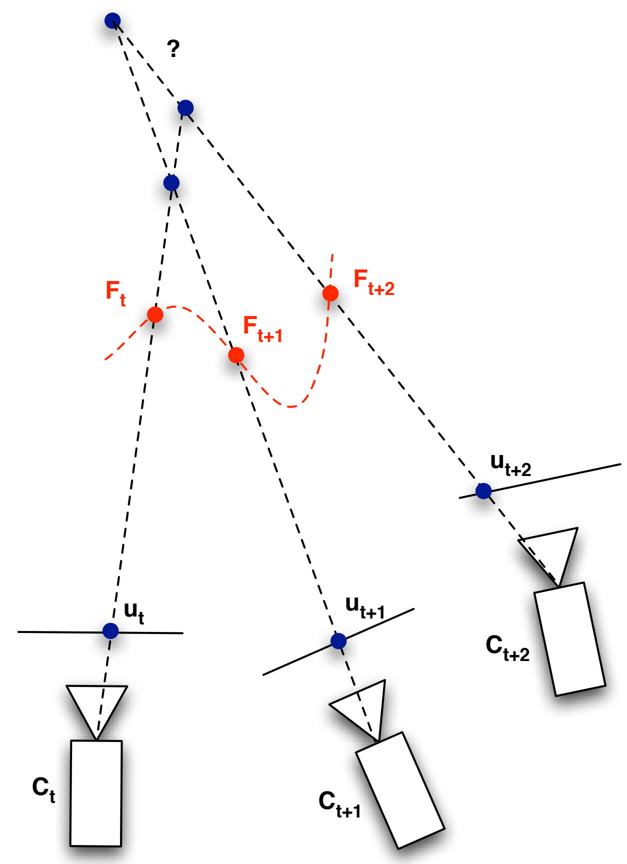
Moving Features Classification

Animare

Moving Features Classification

- We need to identify moving features
  - We need at least two viewing rays!

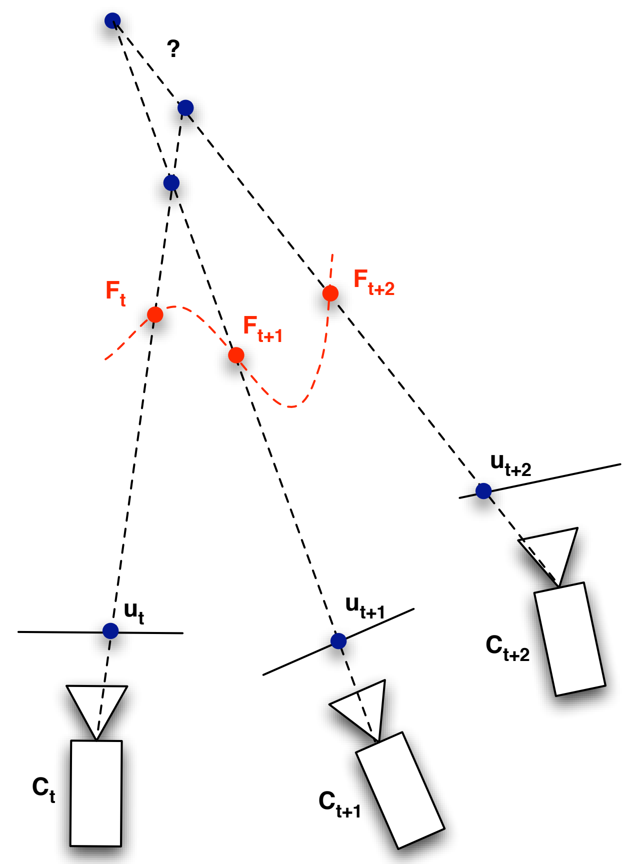
Animare



Moving Features Classification

- We need to identify moving features
  - We need at least two viewing rays!

Animare

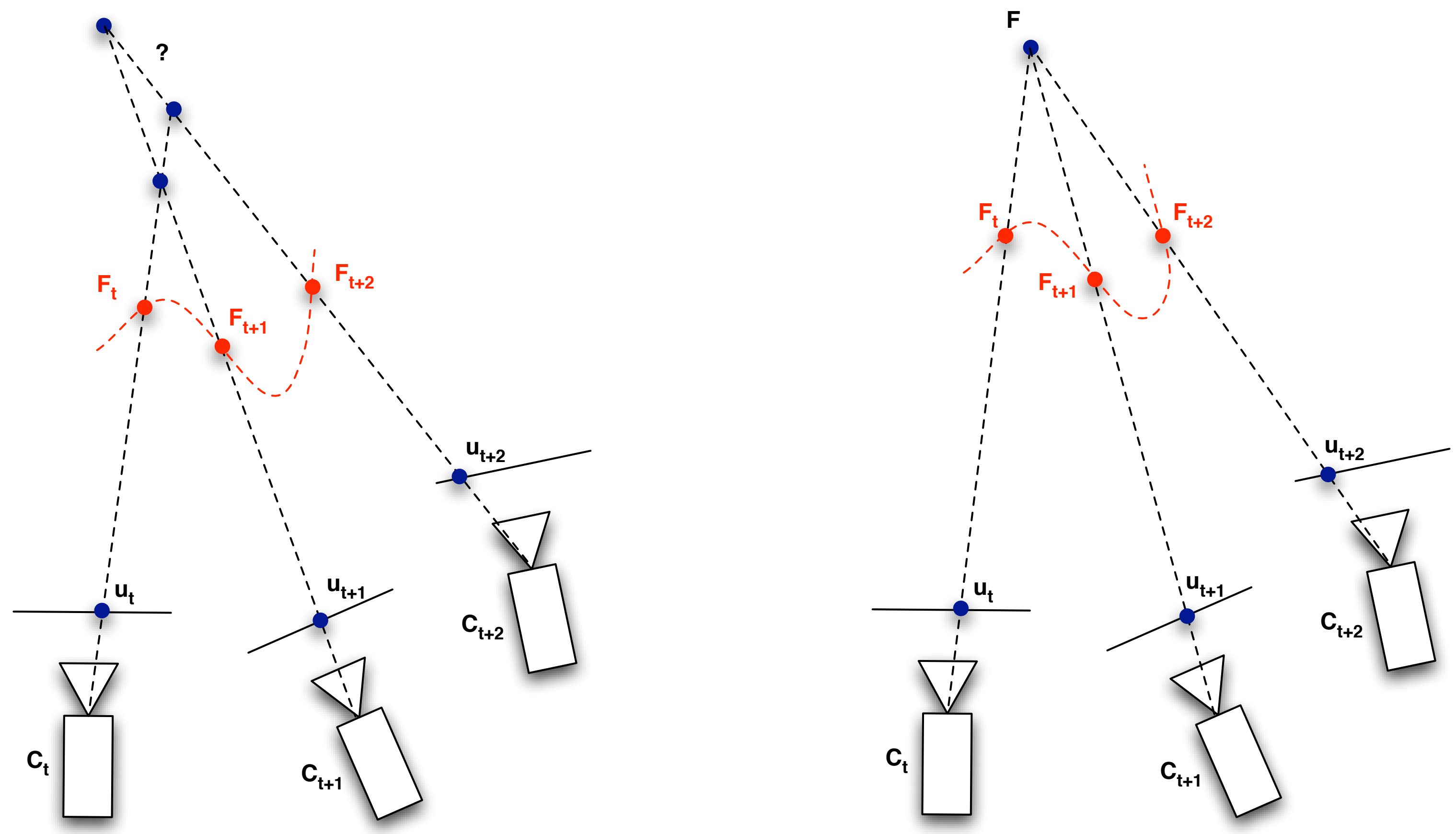


  - The idea:
    - *Save a first viewing ray and move*
    - *Save a second viewing ray (after a while) and move*
    - *If the intersections of the third viewing ray with the others are distinct, then the feature is moving*

Moving Features Classification

- We need to identify moving features
  - We need at least two viewing rays!

Animare



- The idea:
  - *Save a first viewing ray and move*
  - *Save a second viewing ray (after a while) and move*
  - *If the intersections of the third viewing ray with the others are distinct, then the feature is moving*

Uncertain Projective Geometry

point

Uncertain Projective Geometry

- Problem: we have to take into account the uncertainties

point

Uncertain Projective Geometry

- Problem: we have to take into account the uncertainties
- Solution: use the Uncertain Geometry Reasoning
  - Construction mechanism:
    - $\mathrm{O}(\cdot)$ *(for 3D lines)*
    - $\Pi(\cdot)$ *(for 3D points and 3D planes)*

$$\mathbf{\Pi}(\mathbf{X}) = \frac{\partial \mathbf{X} \wedge \partial \mathbf{Y}}{\partial \mathbf{Y}} = \begin{pmatrix} W_1 & 0 & 0 & X_1 \\ 0 & W_1 & 0 & -Y_1 \\ 0 & 0 & W_1 & -Z_1 \\ 0 & -Z_1 & Y_1 & 0 \\ Z_1 & 0 & -X_1 & 0 \\ -Y_1 & X_1 & 0 & 0 \end{pmatrix} \qquad \mathbf{O}(\mathbf{L}) = \frac{\partial \mathbf{X} \wedge \partial \mathbf{L}}{\partial \mathbf{X}} = \begin{pmatrix} 0 & L_3 & -L_2 & -L_4 \\ -L_3 & 0 & L_1 & -L_5 \\ L_2 & -L_1 & 0 & -L_6 \\ L_4 & L_5 & L_6 & 0 \end{pmatrix}.$$

- S. Heuel - *Uncertain Projective Geometry: Statistical Reasoning for Polyhedral Object Reconstruction*. Springer (2004)
- D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti et al. "*Integration of 3D Lines and Points in 6DoF Visual SLAM by Uncertain Projective Geometry*" - ECMR 2007
- D. Marzorati, M. Matteucci, D. Migliore, D. G. Sorrenti et al. "*Data Fusion by Uncertain Projective Geometry in 6DoF Visual SLAM*" - VISAPP 2008

point

Uncertain Projective Geometry

- Problem: we have to take into account the uncertainties
- Solution: use the Uncertain Geometry Reasoning
  - Construction mechanism:
    - $\mathrm{O}(\cdot)$ *(for 3D lines)*
    - $\Pi(\cdot)$ *(for 3D points and 3D planes)*

Uncertain Projective Geometry

- Problem: we have to take into account the uncertainties
- Solution: use the Uncertain Geometry Reasoning
  - Construction mechanism:
    - $\mathrm{O}(\cdot)$ *(for 3D lines)*
    - $\Pi(\cdot)$ *(for 3D points and 3D planes)*
  - A bilinear expression to compute relationship and propagate their uncertainties

$$z = f(x, y) = O(x)y = \Pi(y)x$$ *(x line, y point and f intersection)*

$$(z, \Sigma_{zz}) = \left(O(x)y, O(x)\Sigma_{yy}O^T(x) + \Pi(y)\Sigma_{xx}\Pi^T(y)\right).$$

Uncertain Projective Geometry

- Problem: we have to take into account the uncertainties
- Solution: use the Uncertain Geometry Reasoning
  - Construction mechanism:
    - $\mathrm{O}(\cdot)$ *(for 3D lines)*
    - $\Pi(\cdot)$ *(for 3D points and 3D planes)*
  - A bilinear expression to compute relationship and propagate their uncertainties

$$z = f(x, y) = O(x)y = \Pi(y)x$$ *(x line, y point and f intersection)*

$$(z, \Sigma_{zz}) = \left(O(x)y, O(x)\Sigma_{yy}O^T(x) + \Pi(y)\Sigma_{xx}\Pi^T(y)\right).$$

  - We can express the previous test as a probabilistic one to verify the intersection between point and line

$$R(x, y) \Leftrightarrow d = O(x)y = \Pi(y)x = 0 \qquad \Sigma_d = O(x)\Sigma_x O(x)^T + \Pi(y)\Sigma_y \Pi(y)^T$$

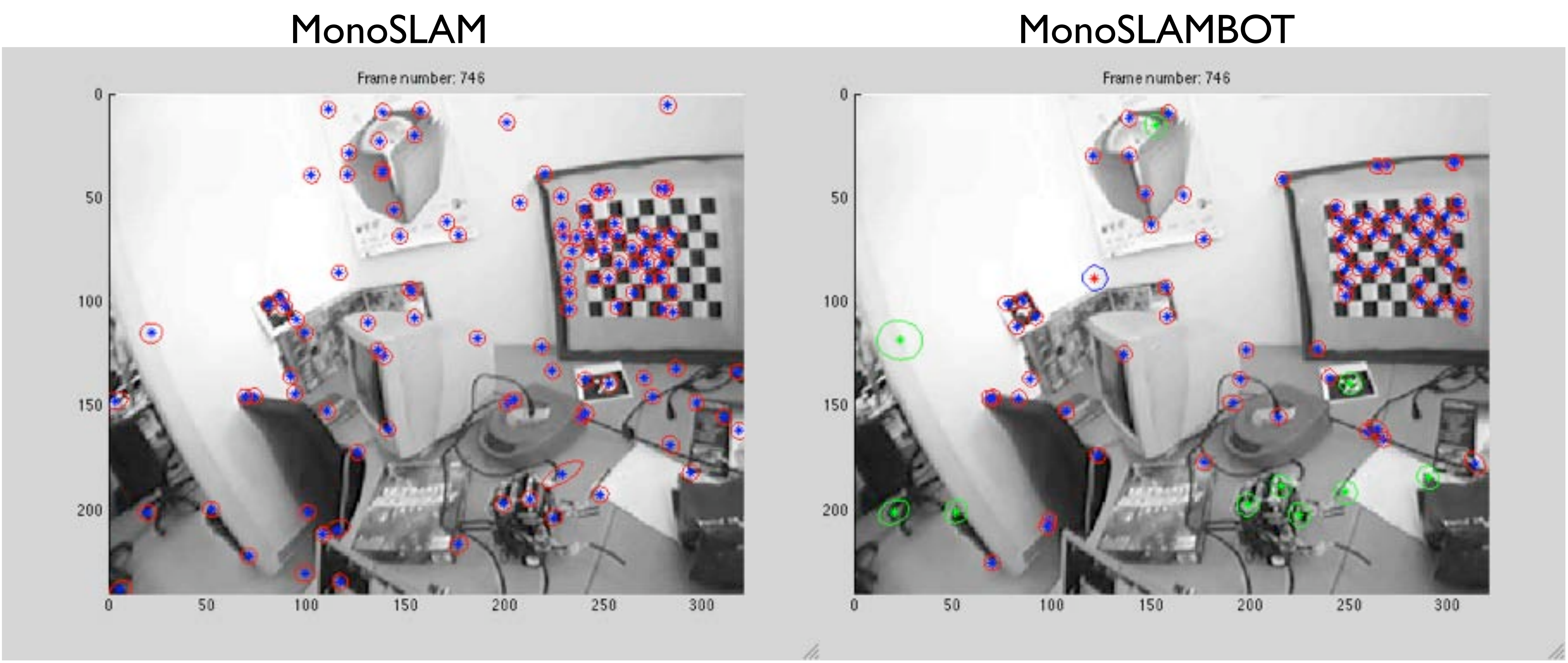
    - *Statistical Test (Chi-square test):*

$$T = d^T \Sigma_d^{-1} d$$

Experimental Results

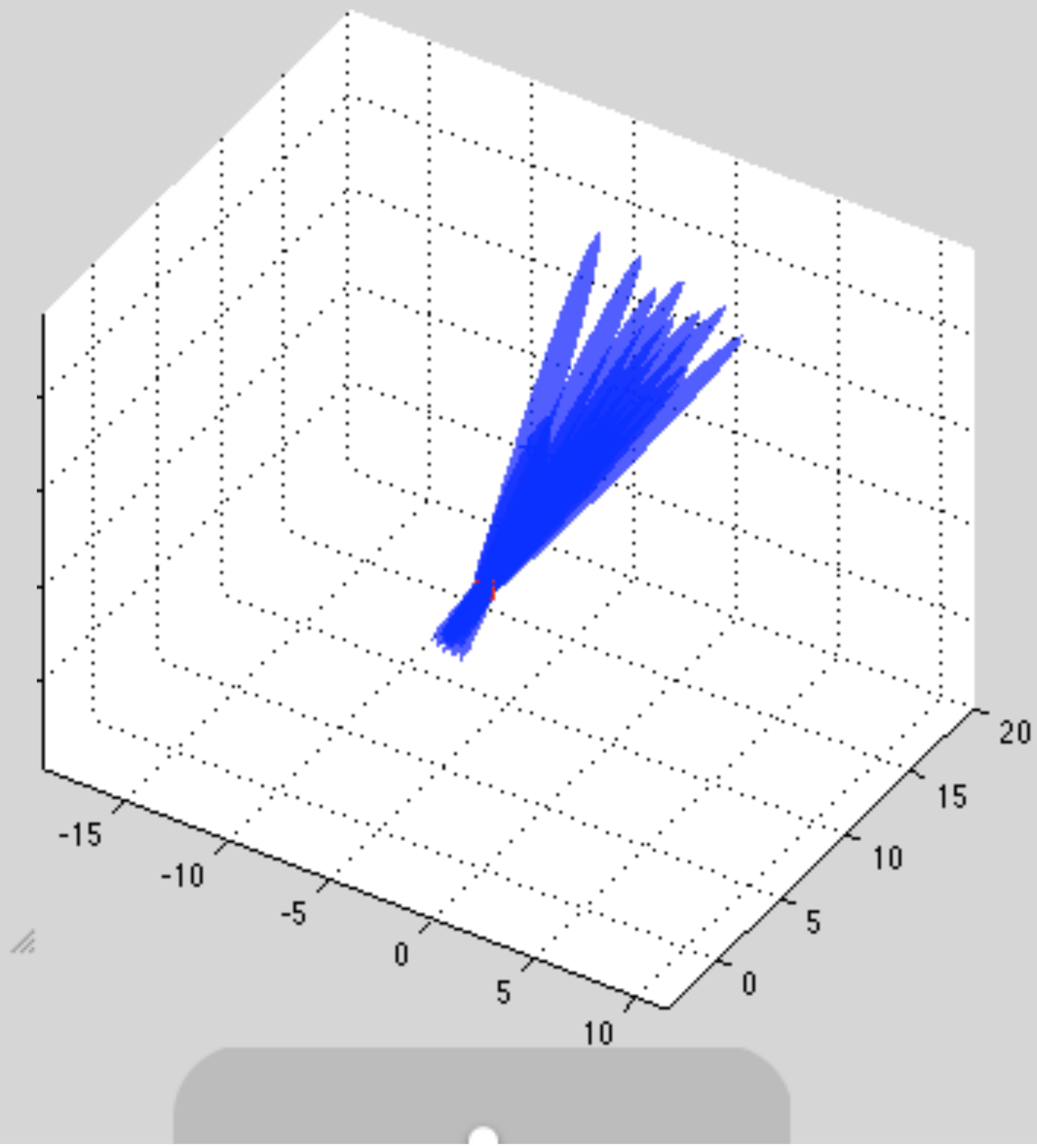
Experimental Results

- Real Dataset



Experimental Results

- Real Dataset



Conclusions

Conclusions

- Principal results achieved:
  - The Inverse Scaling Parametrization (ISP) for Bearing Only Tracking
  - Integration of SLAM and Moving Feature Tracking

Conclusions

- Principal results achieved:
  - The Inverse Scaling Parametrization (ISP) for Bearing Only Tracking
  - Integration of SLAM and Moving Feature Tracking

- Ongoing Works (airwiki.elet.polimi.it):
  - Accurate Stability Analysis of BOT Filter
  - Large Maps (CI-SLAMBOT - submitted to IROS 2009)
  - Improve Data association
  - StereoSLAM, BiCamSLAM, OmniSLAM
  - Sensor Fusion: IMU, GPS, Sick Lasers...
  - Model based objects tracking
  - Test with http://www.rawseeds.org/ dataset

Conclusions

- Principal results achieved:
  - The Inverse Scaling Parametrization (ISP) for Bearing Only Tracking
  - Integration of SLAM and Moving Feature Tracking

- Ongoing Works (airwiki.elet.polimi.it):
  - Accurate Stability Analysis of BOT Filter
  - Large Maps (CI-SLAMBOT - submitted to IROS 2009)
  - Improve Data association
  - StereoSLAM, BiCamSLAM, OmniSLAM
  - Sensor Fusion: IMU, GPS, Sick Lasers...
  - Model based objects tracking
  - Test with http://www.rawseeds.org/ dataset
  - Autonomous vehicle navigation

Thanks for your attention

- Questions?!?