

---

# Visual-inertial hybridization with SLAM

## Application to mobile robotics

Simon Lacroix  
LAAS/CNRS, Toulouse



# Mobile robotics

---



Robots everywhere

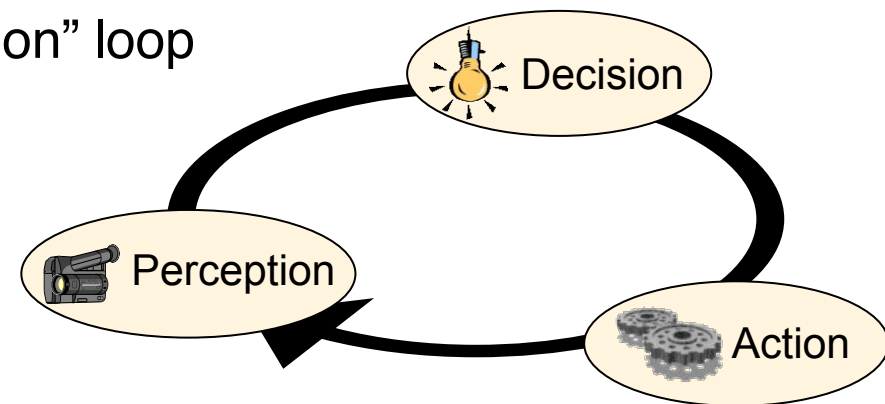
## “From automatic control to autonomous control”

- Automatic control :
  - Well defined task (“*regulate variable*”, “*follow trajectory*”...)
  - “Direct” link between (simple) perception and action
- Autonomous control :
  - More general task (“*reach position*”, “*monitor area* ”...)
  - Calls for *decisional processes*

⇒ “perception / Decision / Action” loop

Plus :

- Processes *integration*
- Learning
- Interaction with humans
- Interactions with other robots
- ...



# Outline

---

- On the importance of localization in robotics
- RT-SLAM
- Outlook

# On the importance of localization

---

Localization is required to:

- Ensure the spatial consistency of the environment models
- Ensure the achievement of the missions, most often defined in localization terms (“goto [goal]”, “explore / monitor [area]”, ...)
- Ensure the lowest level (locomotion) controls / diagnosis
- Ensure the proper execution of paths / trajectories

# Environment models spatial consistency

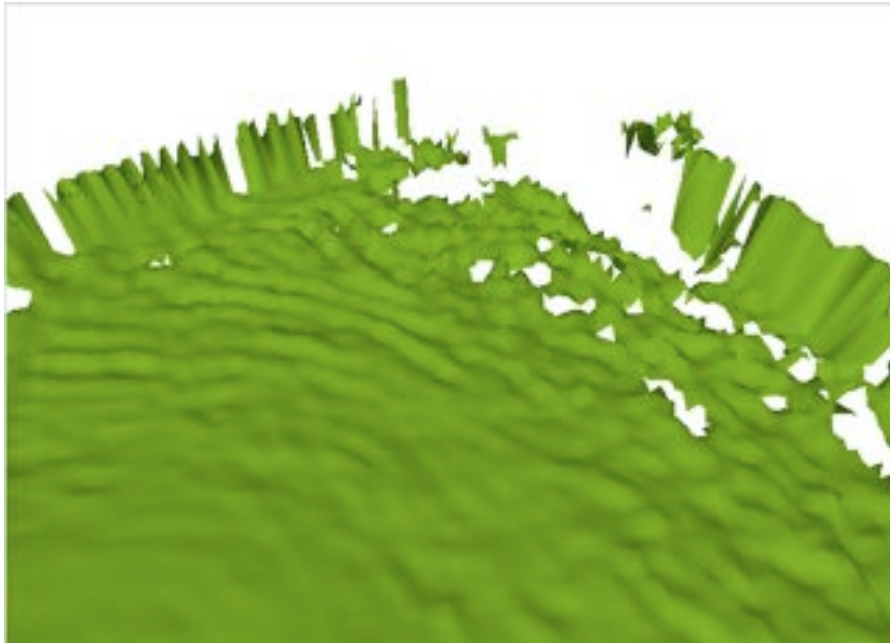
---



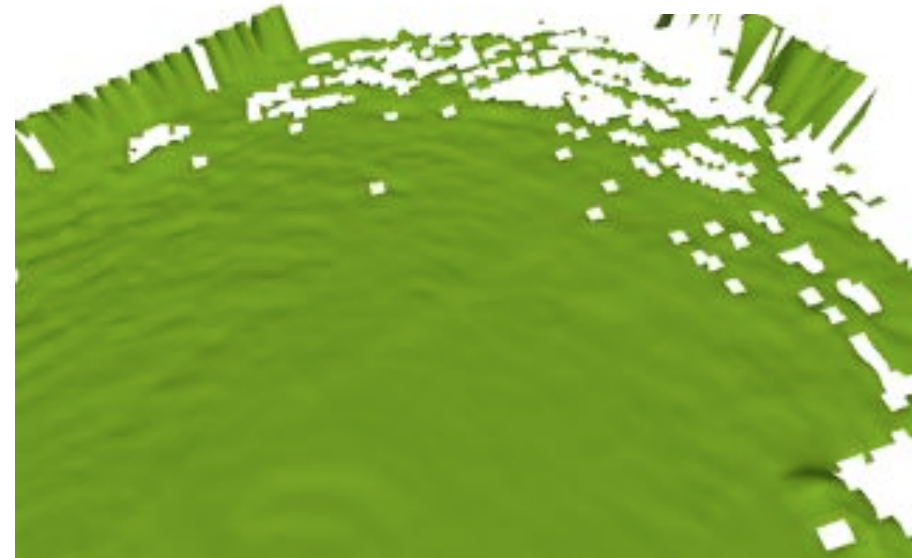
# Environment models spatial consistency

---

- Velodyne lidar provides chunks of 64 points @ 3.5 kHz:  
1° error on pitch yields a 17cm elevation error @ 10m



2m/s, GPS RTK @ 20Hz  
+ Xsens AHRS @ 50Hz  
+ FOG gyro @ 50Hz



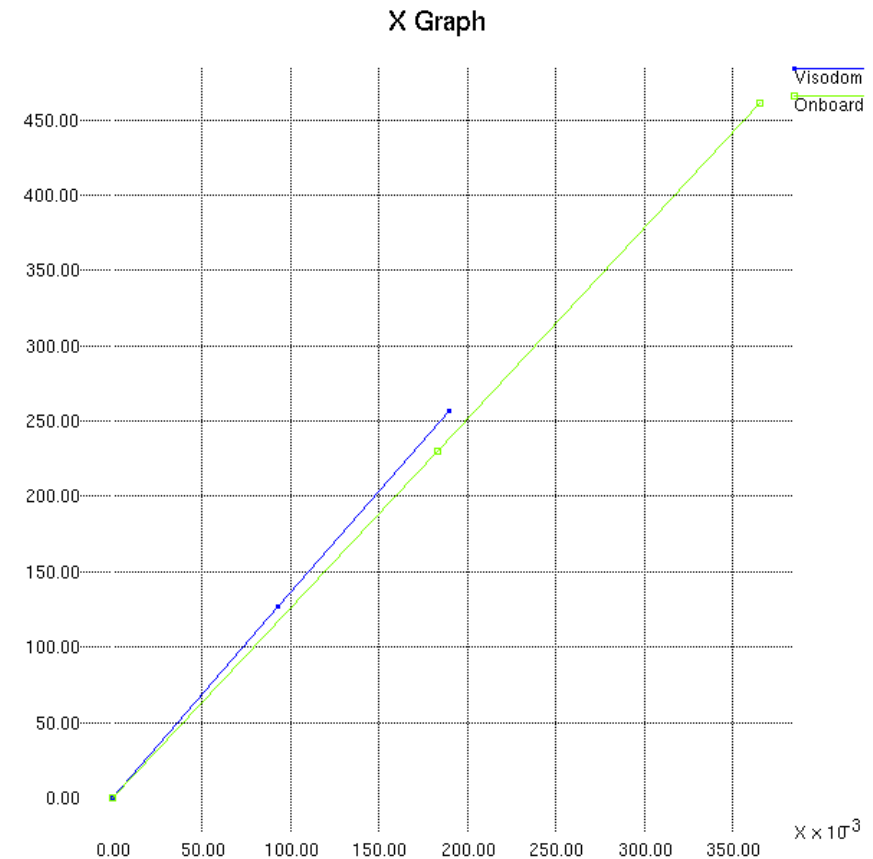
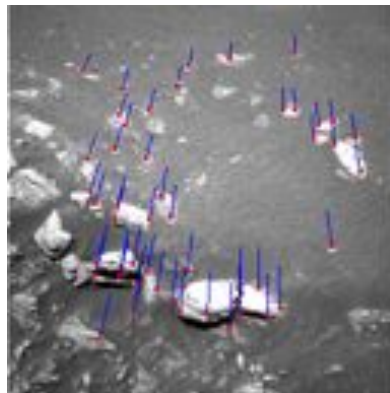
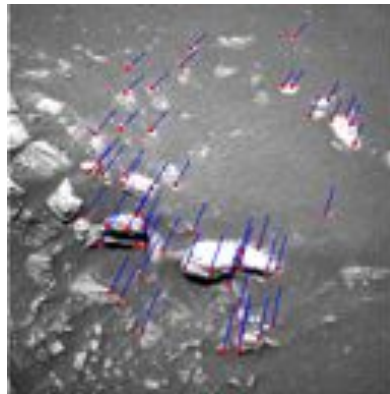
2m/s, RT-SLAM @ 100Hz

Required localization : *cm* accuracy, local (or global) frame, frequency defined by the range data acquisition rate (often high)

# Proper execution of trajectories

Trajectories are defined by geometric primitives : a precise localization estimate is required

50% slip  
→ visual  
odometry  
(Spirit in  
2004)



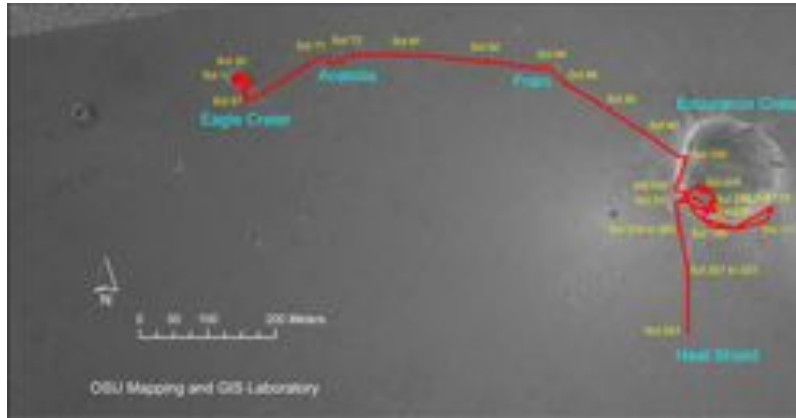
Required localization : *cm* accuracy, local frame, frequency function of speed



# Locomotion control / diagnosis

---

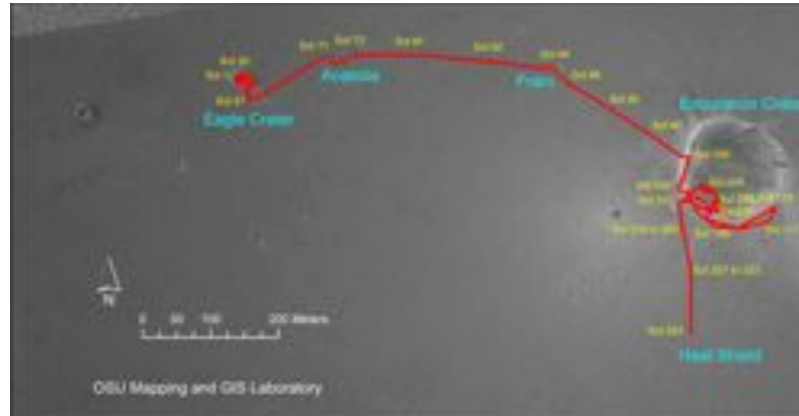
Opportunity  
traverse



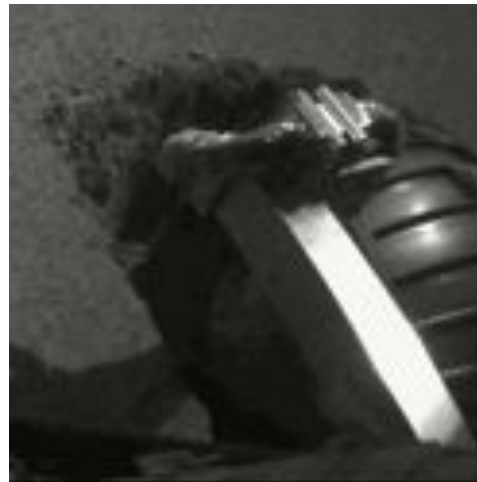
# Locomotion control / diagnosis

---

Opportunity  
traverse



April 26th, 2005, on  
the way to Victoria  
Crater

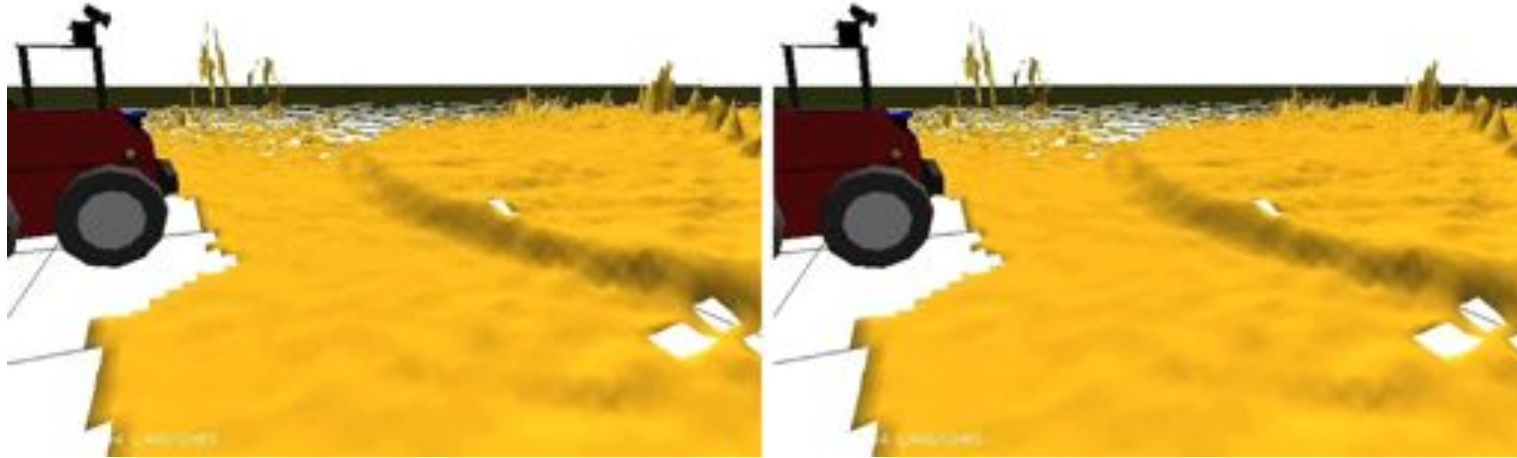


Rear left wheel

Required localization : *cm* accuracy, local frame, high frequency

# Locomotion control / diagnosis

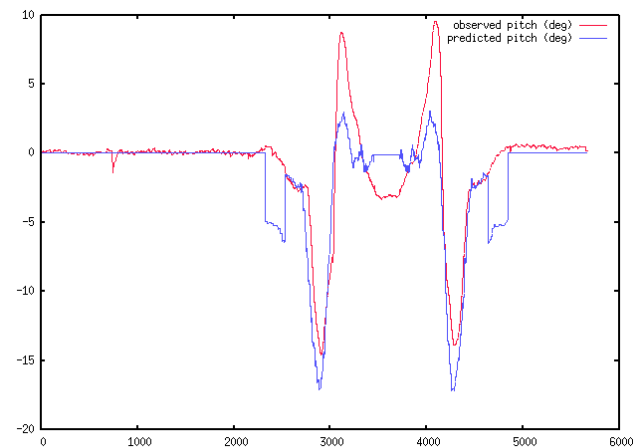
Monitoring the terrain profile



predicted

measured

Predicted vs. observed  
robot pitch angle

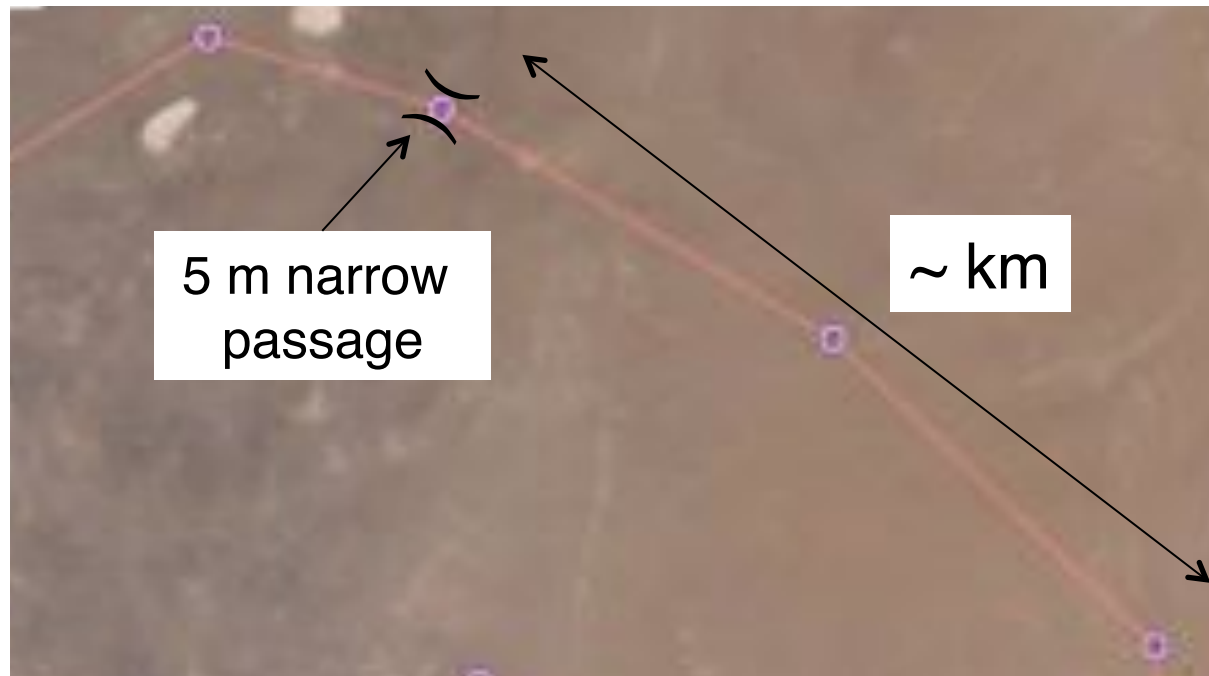


Required localization : *cm* accuracy, local frame, high frequency

# Ensuring mission achievement

---

- Odometry, VO, INS, SLAM (without loop closures) localisation solutions eventually drift
  - ⇒ No way to ensure the achievement of global mission plans



⇒ Need to rely on “absolutely localized” features : geo-referenced maps

Required localization :  $m$  accuracy, global *absolute* frame, very low frequency

# Which localization?

---

Essential questions to answer:

- |                          |                                 |
|--------------------------|---------------------------------|
| 1. With which precision? | From <i>cm</i> to <i>meters</i> |
| 2. In which frame?       | Absolute vs. local              |
| 3. At which frequency?   | From <i>kHz</i> to “sometimes”  |

*cm* accuracy,  
@ > 100 Hz,  
local frame

- Ensure the lowest level (locomotion) controls
- Ensure the proper execution of paths / trajectories
- Ensure the spatial consistency of the built models

~*m* accuracy,  
“sometimes”,  
global frame

- Ensure the achievement of the missions, most often defined in localization terms (“goto [goal]”, “explore / monitor [area]”, ...)

# Which localization?

---

Essential questions to answer:

1. With which precision? From *cm* to *meters*
2. In which frame? Absolute vs. local
3. At which frequency? From *kHz* to “sometimes”
4. Which availability?
5. Integrity?

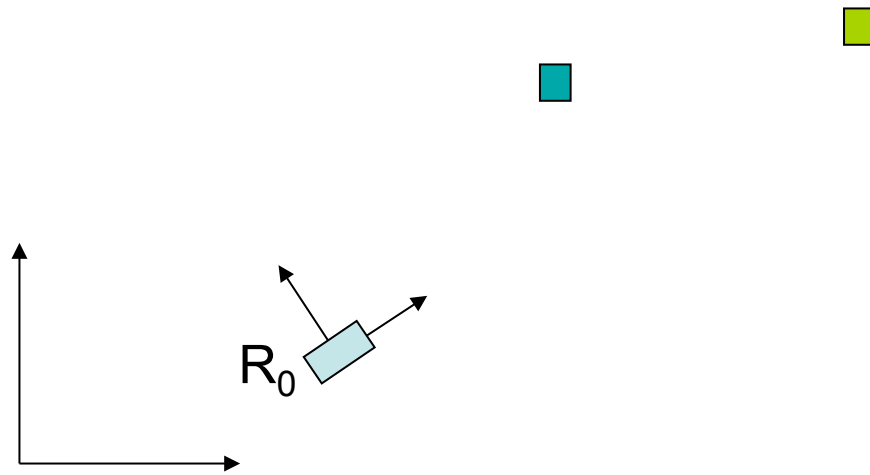
# Outline

---

- On the importance of localization
- RT-SLAM
  - SLAM in one slide
- Outlook

# Principle of SLAM

---

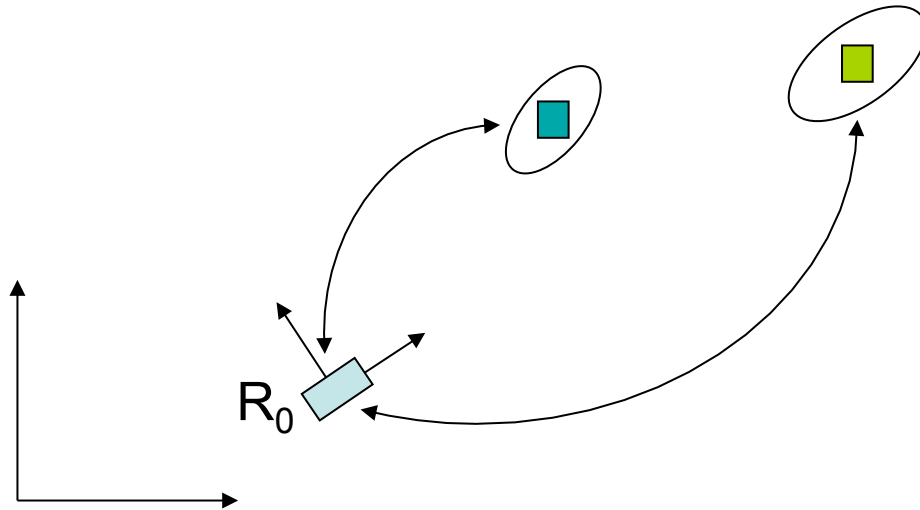


– Landmark detection



# Principle of SLAM

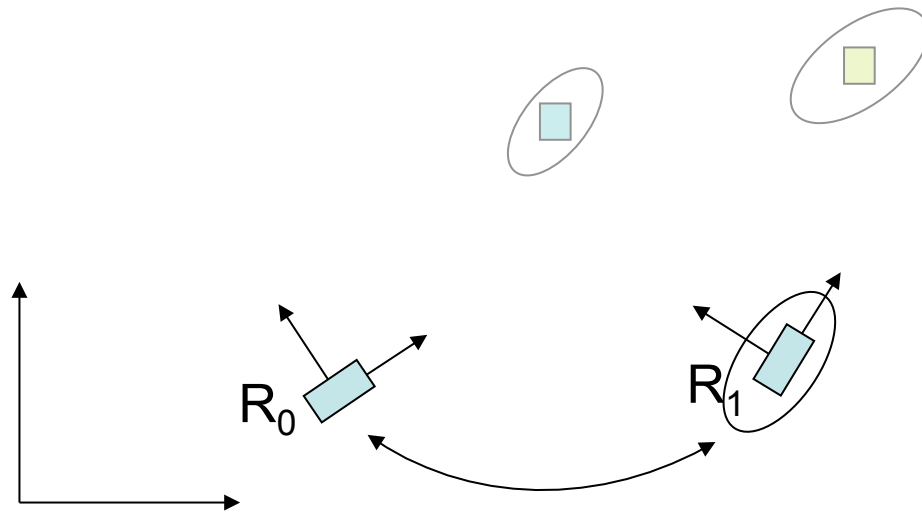
---



- Landmark detection
- Relative observations (measures)
  - Of the landmark positions

# Principle of SLAM

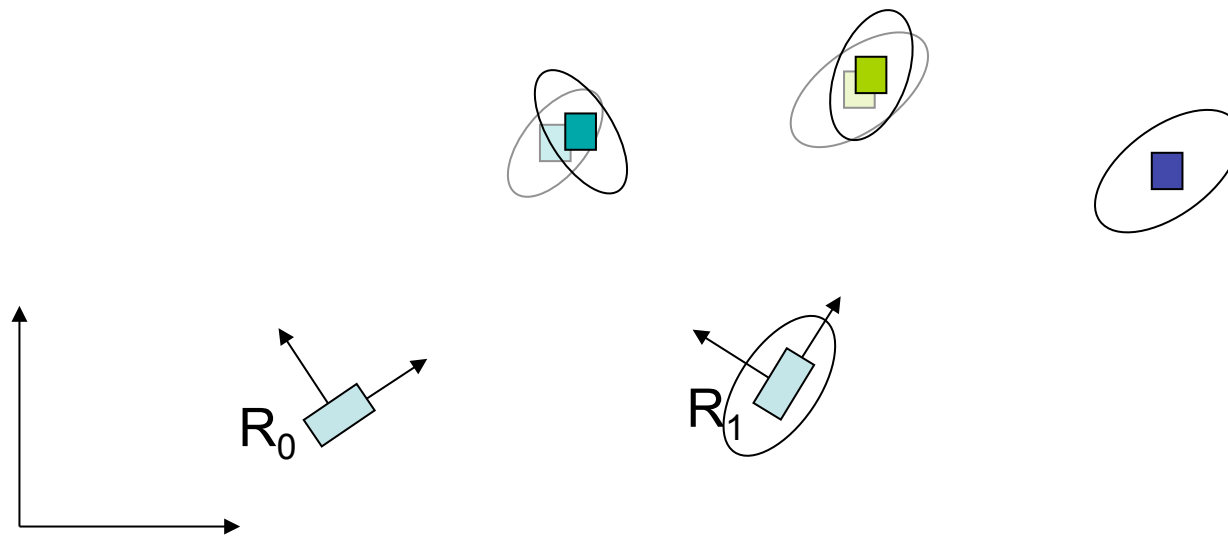
---



- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions

# Principle of SLAM

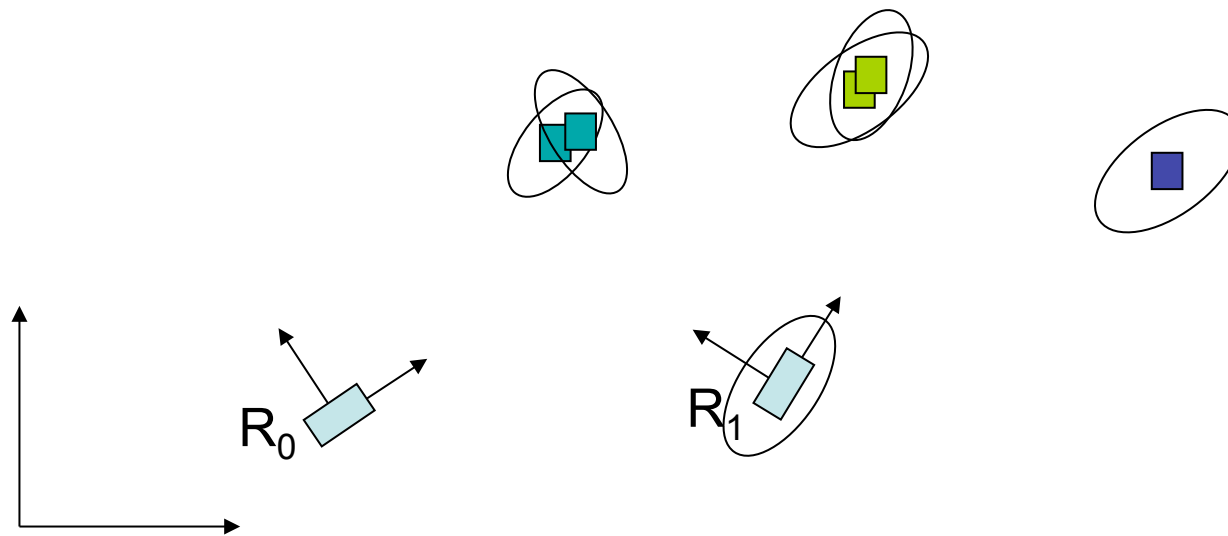
---



- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions

# Principle of SLAM

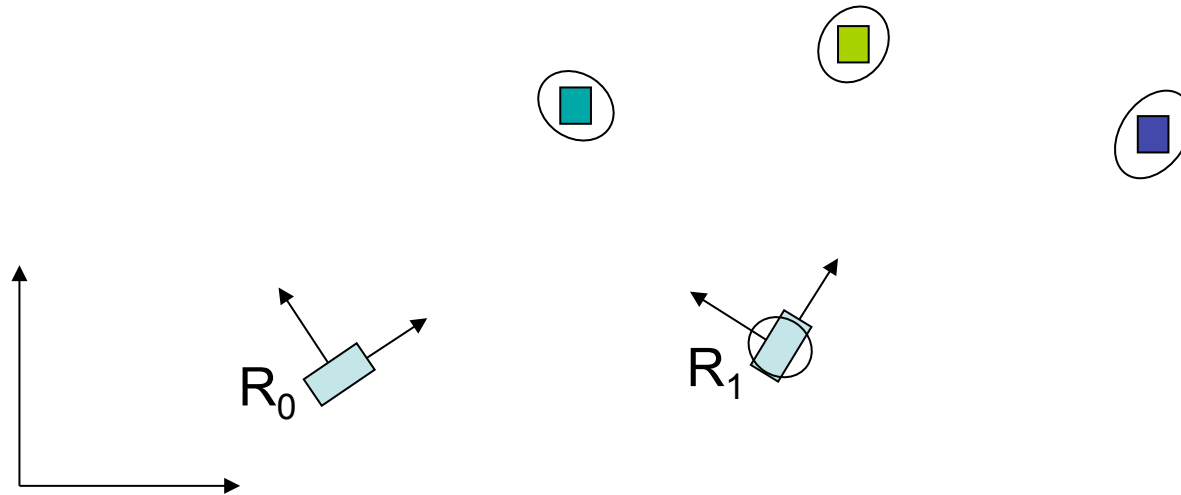
---



- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions
- Observation associations

# Principle of SLAM

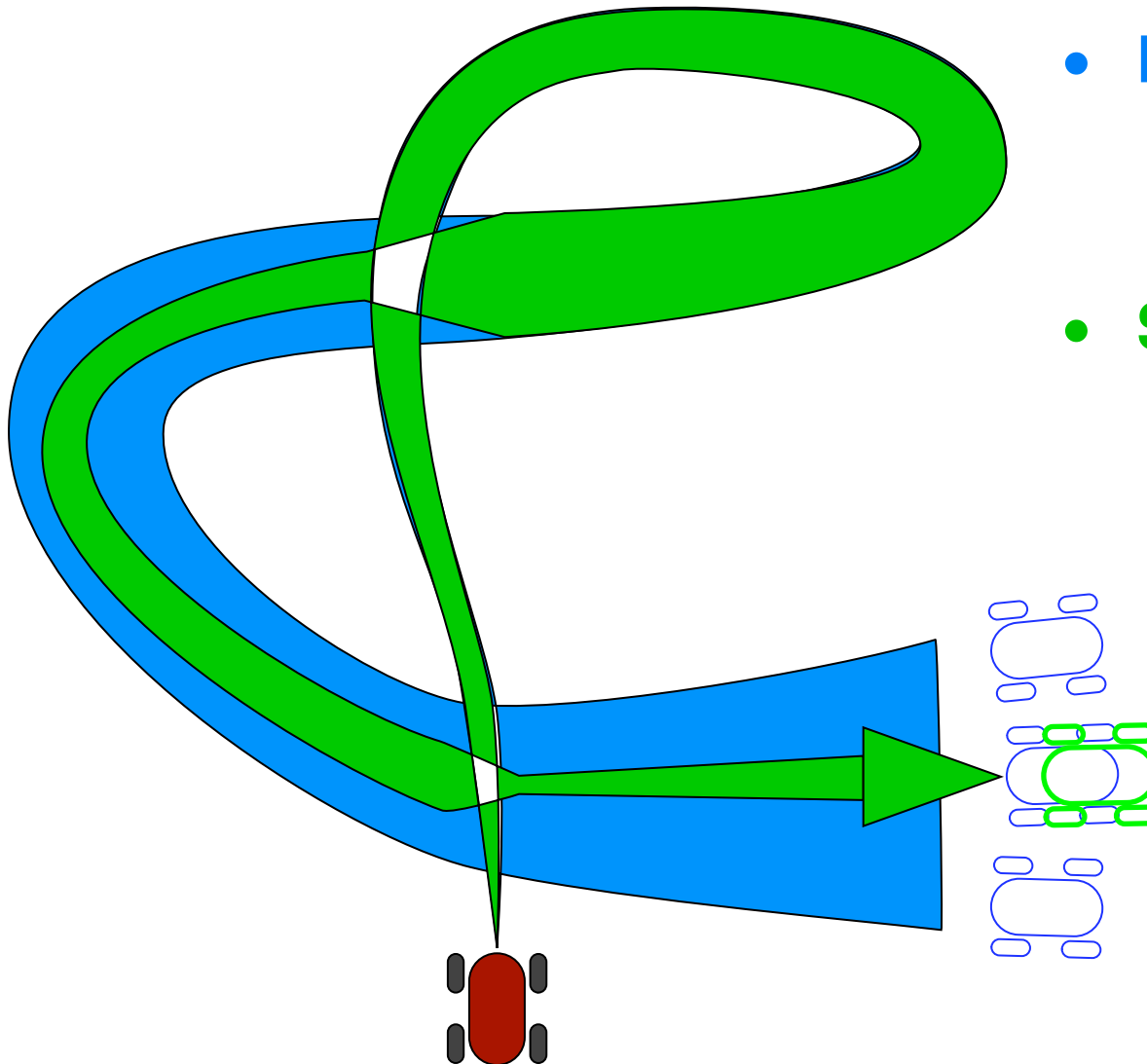
---



- Landmark detection
- Relative observations (measures)
  - Of the landmark positions
  - Of the robot motions
- Observation associations
- Refinement of the landmark and robot positions

# Principle of SLAM

---



- **Dead reckoning**

- Monotonic increase of the position uncertainty

- **SLAM**

- “memory effect” of the mapping
- Loop closures: position uncertainty decrease

# Dozens of SLAM solutions...

---

Essential questions to answer:

1. Which prediction model / sensor? Motion assumption, dynamic model, odometry, INS, visual odometry...
2. Which exteroceptive sensors? Vision (and all its variations), Lidars, Radars...
3. Which landmarks? Points / lines / planes / dense signal (no landmark)...
4. Which estimation formalism? KF++, IF, PF, Set membership, non linear optimization...
5. How to detect loop-closures?

- On the importance of localization
- RT-SLAM
  - SLAM in (more than) one slide
  - RT-SLAM

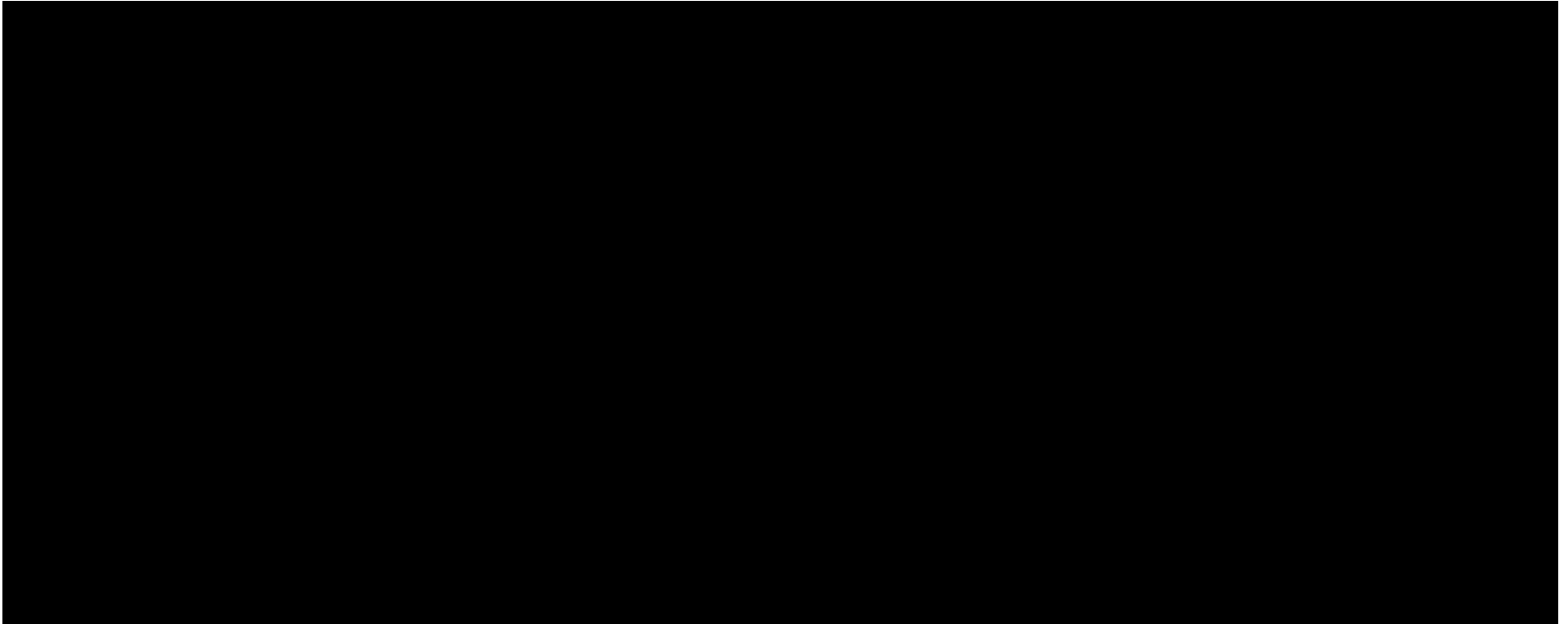
(Joan Sola, Cyril Roussillon, Aurélien Gonzalez, Jean-Marie Codol)

- Outlook



# A glimpse at RT-SLAM capacity

---



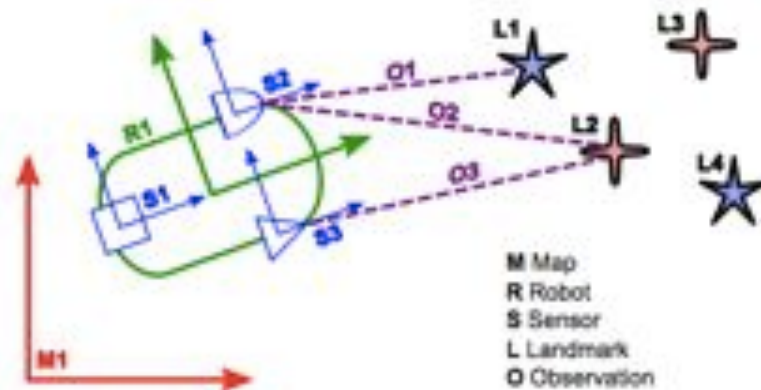
<http://rtslam.openrobots.org>

# RT-SLAM choices

---

1. Which prediction model / sensor?  
Motion assumption, dynamic model, odometry, INS, visual odometry...
2. Which exteroceptive sensors?  
Vision (and all its variations), Lidars, Radars
3. Which landmarks?  
Points / lines / planes / dense signal (no landmark)...
4. Which estimation formalism?  
KF++, IF, PF, Set membership, non linear optimization...
5. How to detect loop-closures?

# RT-SLAM generic architecture



Chaque objet est générique

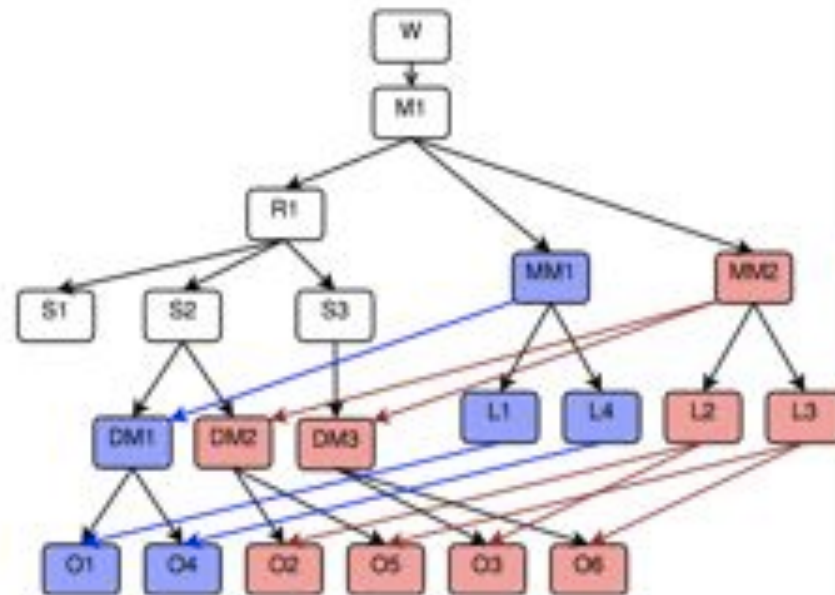
► **Carte** : moteur d'estimation

► **Robot** : modèle de prédiction

► **Capteur** : modèle, matériel

► **Amer** : modèle, descripteur

► **Observation** : modèle, variables intermédiaires



► **DataManager** : stratégie d'observation des amers

► **MapManager** : stratégie de gestion de vie des amers

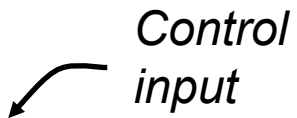
# Kalman filter for SLAM

---

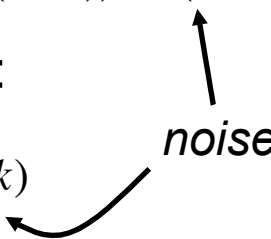
Classic implementation  
(e.g. vehicle position tracking)

System state:  $x(k)$ , with variance  $P_x$

System model:

$$x(k+1) = f(x(k), u(k+1)) + v(k+1)$$


Observation model:

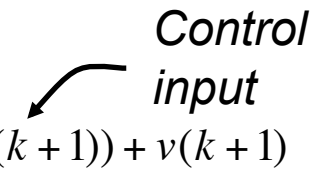
$$z(k) = h(x(k)) + w(k)$$


# Kalman filter for SLAM

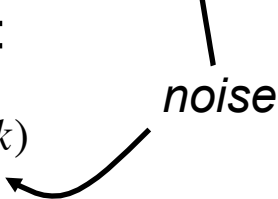
Classic implementation  
(e.g. vehicle position tracking)

System state:  $x(k)$ , with variance  $P_x$

System model:  
 $x(k+1) = f(x(k), u(k+1)) + v(k+1)$



Observation model:  
 $z(k) = h(x(k)) + w(k)$



SLAM implementation

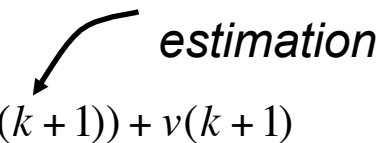
System state:

$x(k) = [x_p, m_1, \dots, m_N]$ , with  $x_p = [\phi, \theta, \psi, t_x, t_y, t_z]$


and  $m_i = [x_i, y_i, z_i]$

$P(k) = \begin{bmatrix} P_{pp}(k) & P_{pm}(k) \\ P_{pm}(k) & P_{mm}(k) \end{bmatrix}$

System model:  
 $x(k+1) = f(x(k), u(k+1)) + v(k+1)$



Observation model:  
 $z(k) = h(x(k)) + w(k)$

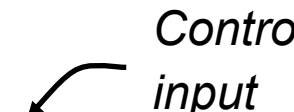


# Kalman filter for SLAM

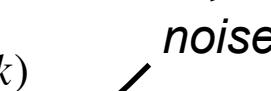
Classic implementation  
(e.g. vehicle position tracking)

System state:  $x(k)$ , with variance  $P_x$

System model:  
 $x(k+1) = f(x(k), u(k+1)) + v(k+1)$



Observation model:  
 $z(k) = h(x(k)) + w(k)$



Steps of the estimation process:

1. Prediction
2. Observation
3. Update

SLAM implementation


System state:

$$x(k) = [x_p, m_1, \dots, m_N], \text{ with } x_p = [\phi, \theta, \psi, t_x, t_y, t_z]$$

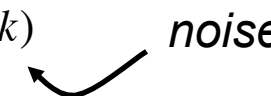
$$\text{and } m_i = [x_i, y_i, z_i]$$

$$P(k) = \begin{bmatrix} P_{pp}(k) & P_{pm}(k) \\ P_{pm}(k) & P_{mm}(k) \end{bmatrix}$$

System model:

$$x(k+1) = f(x(k), u(k+1)) + v(k+1)$$


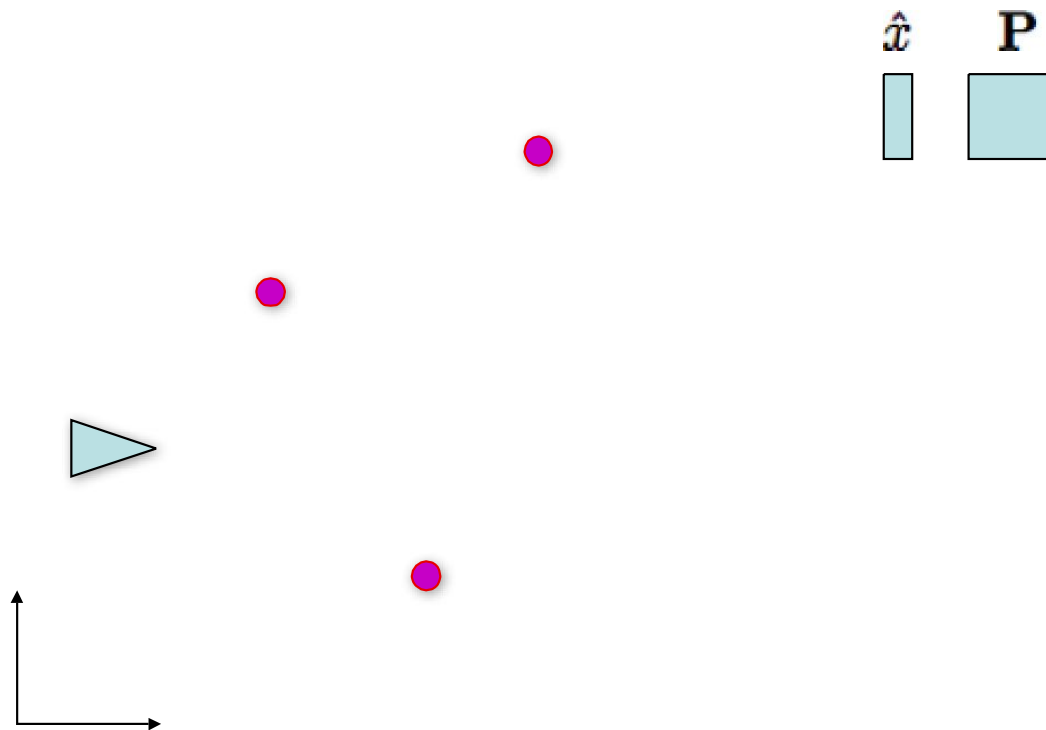
Observation model:

$$z(k) = h(x(k)) + w(k)$$


Why choosing an EKF?

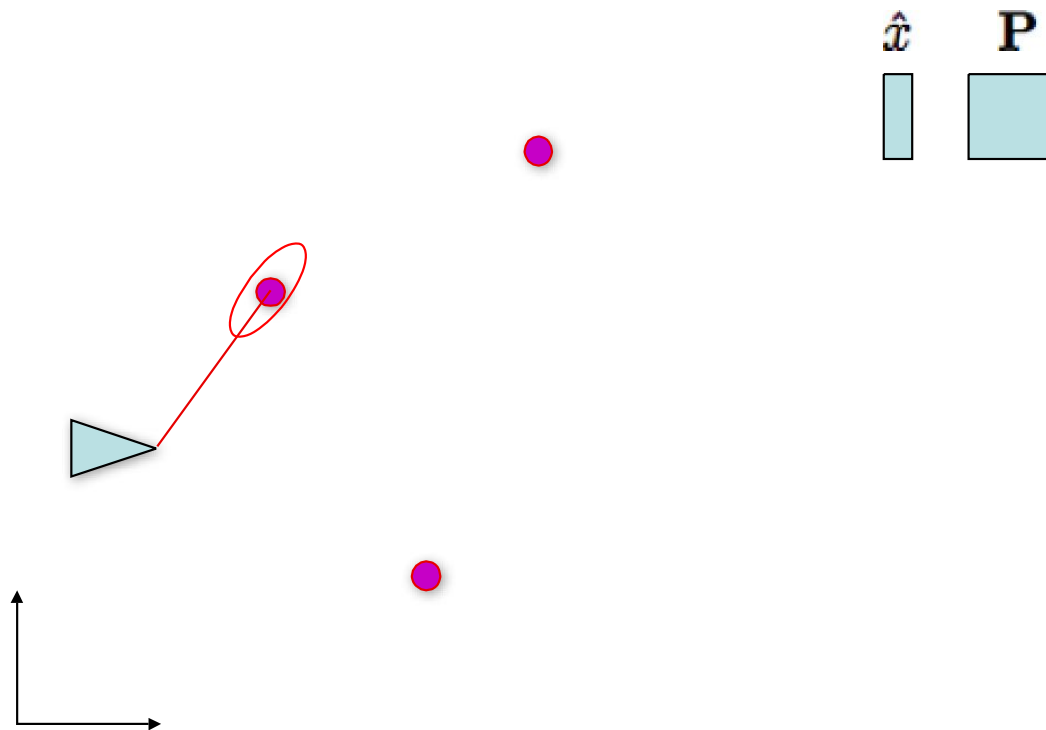
# Kalman filter for SLAM

---



# Kalman filter for SLAM

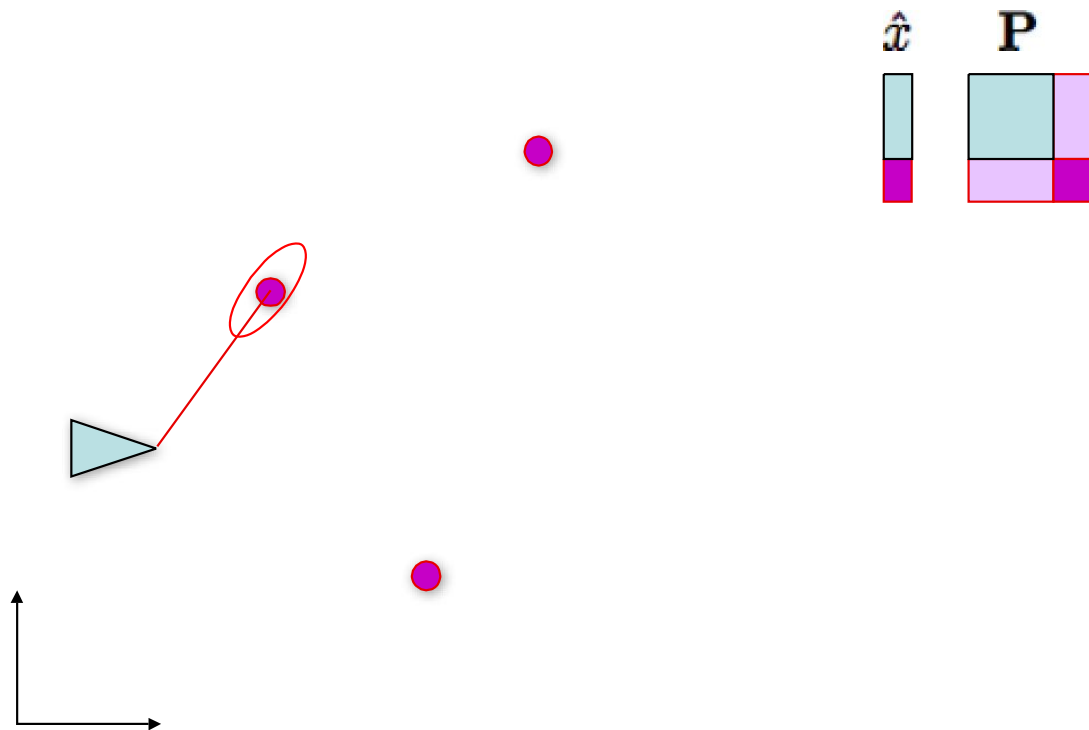
---





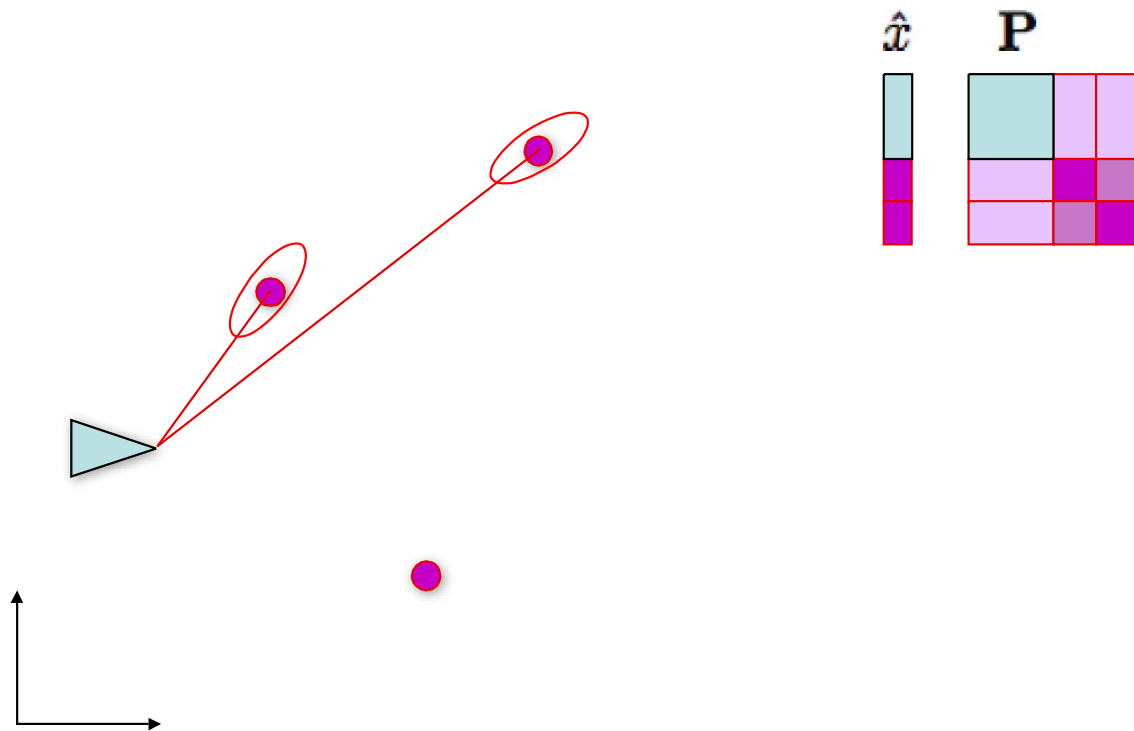
# Kalman filter for SLAM

---



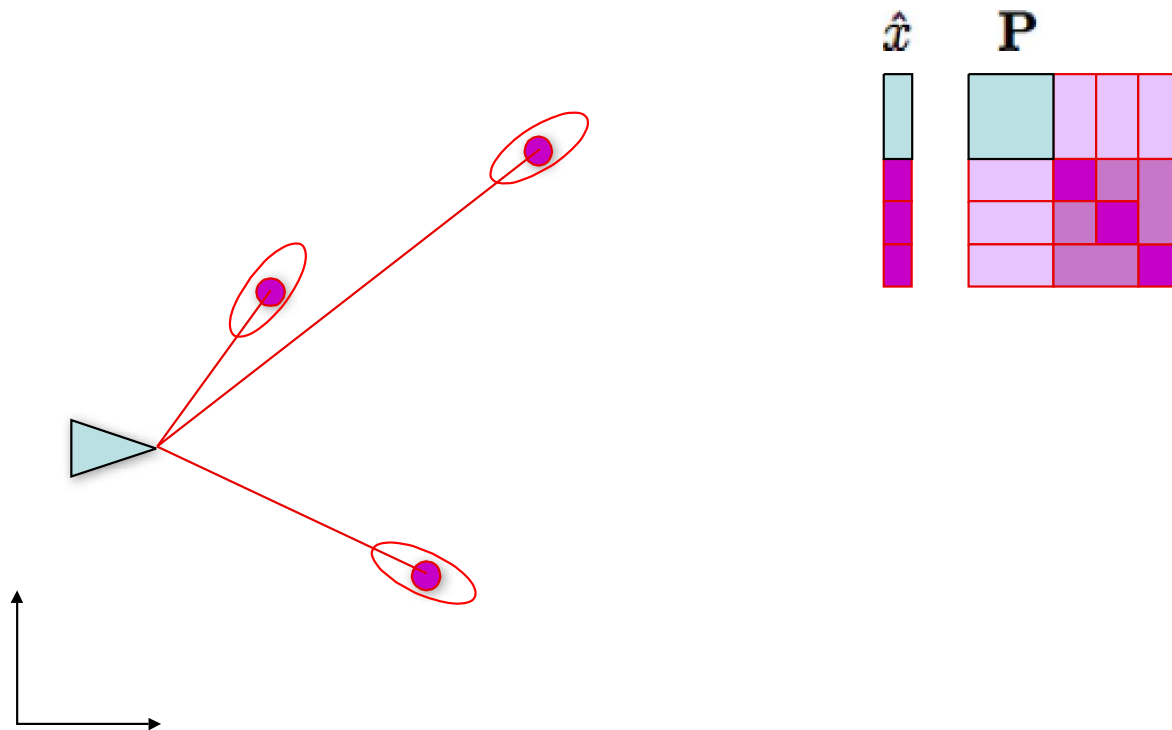
# Kalman filter for SLAM

---



# Kalman filter for SLAM

---

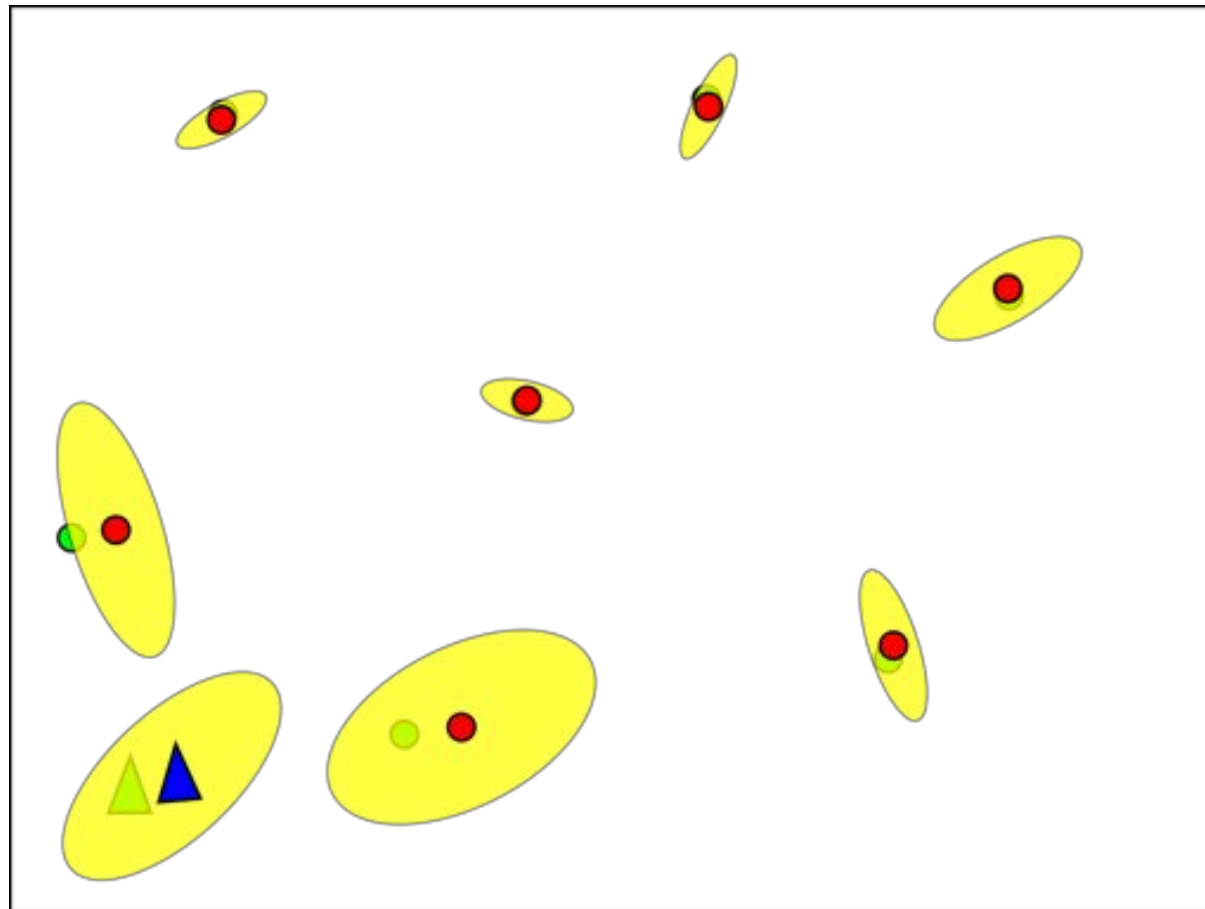


# Kalman filter for SLAM

---

Importance of the correlations

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{mm} \end{bmatrix}$$

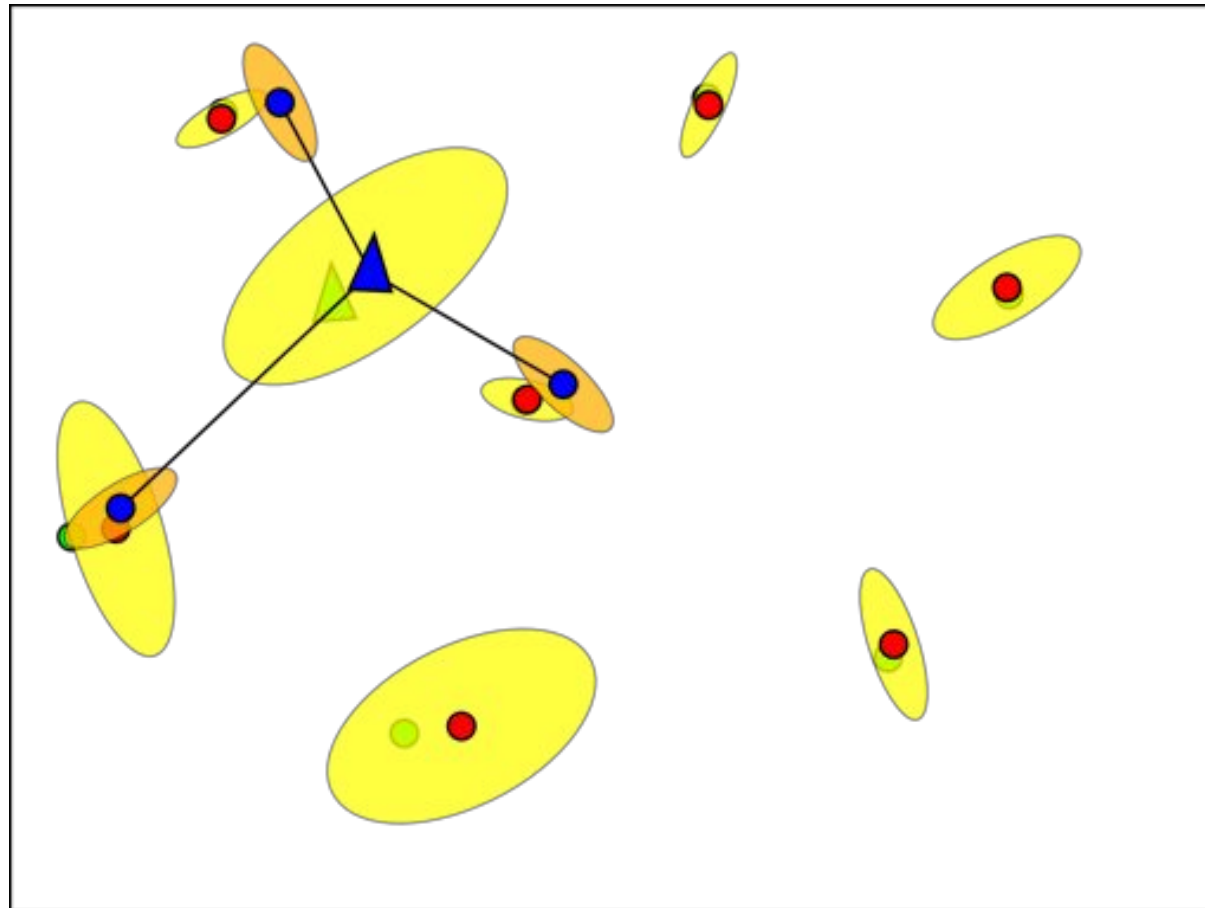


# Kalman filter for SLAM

---

Importance of the correlations

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{mm} \end{bmatrix}$$

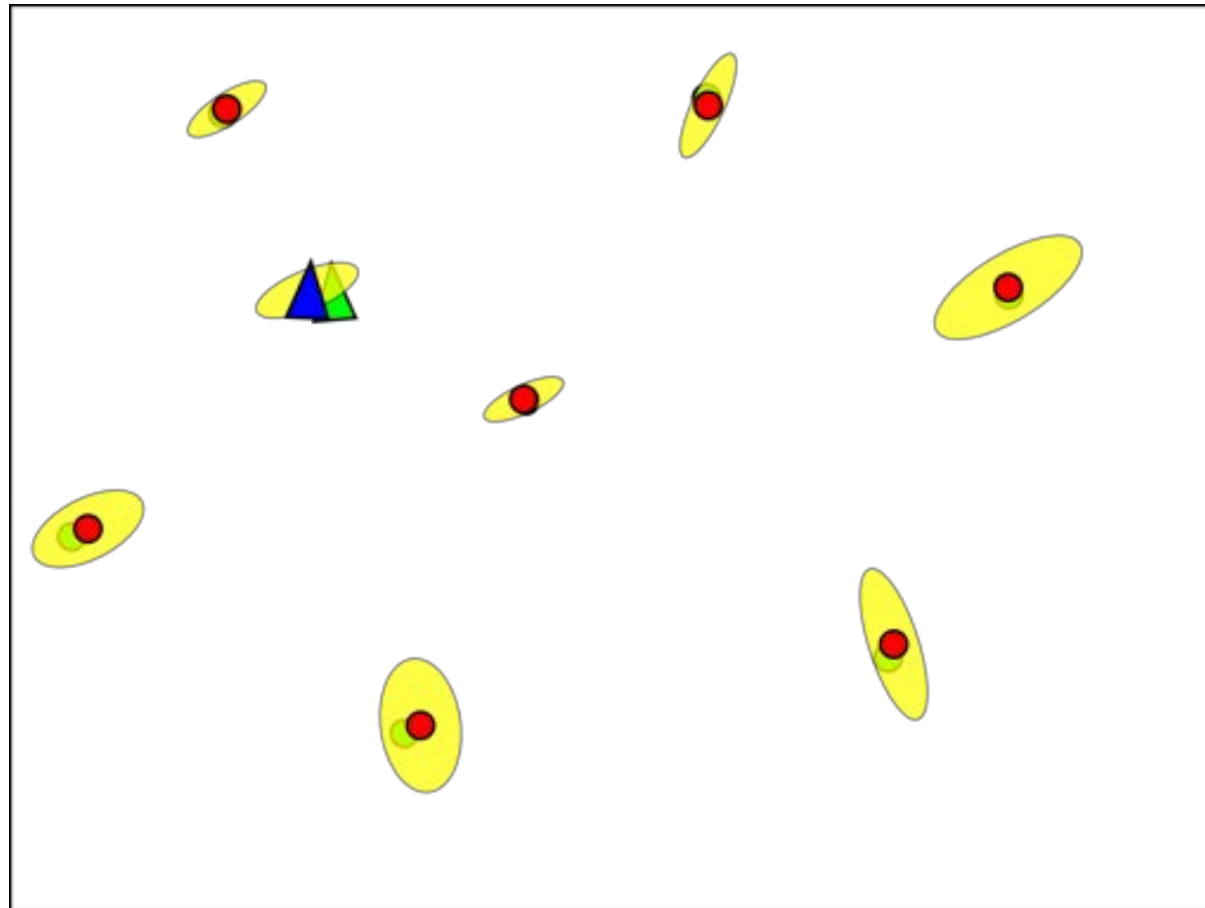


# Kalman filter for SLAM

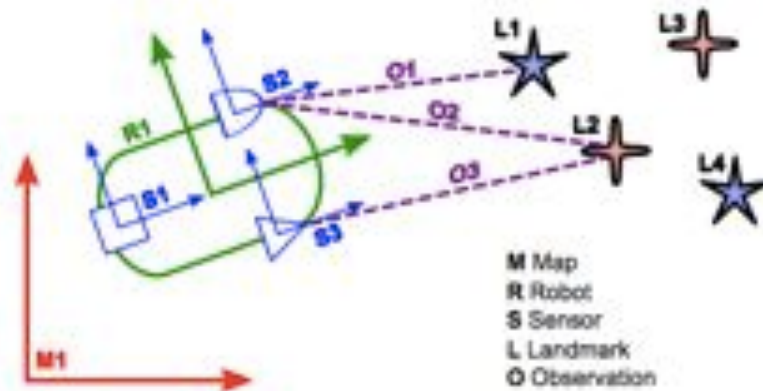
---

Importance of the correlations

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{mm} \end{bmatrix}$$



# Models



Chaque objet est générique

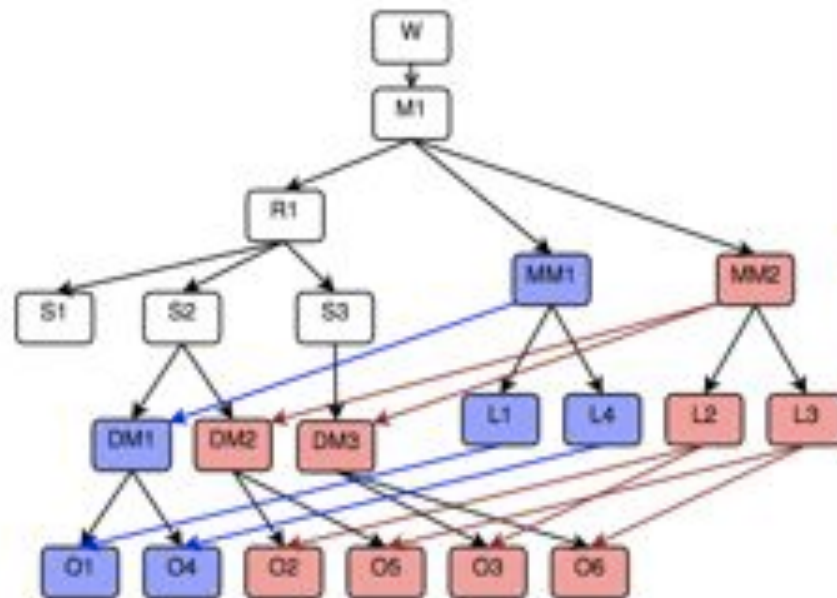
► **Carte** : moteur d'estimation

► **Robot** : modèle de prédiction

► **Capteur** : modèle, matériel

► **Amer** : modèle, descripteur

► **Observation** : modèle, variables intermédiaires



► **DataManager** : stratégie d'observation des amers

► **MapManager** : stratégie de gestion de vie des amers

## Motion models:

- From constant position to constant jerk

Scale drift issue with monocular vision

- Using INS

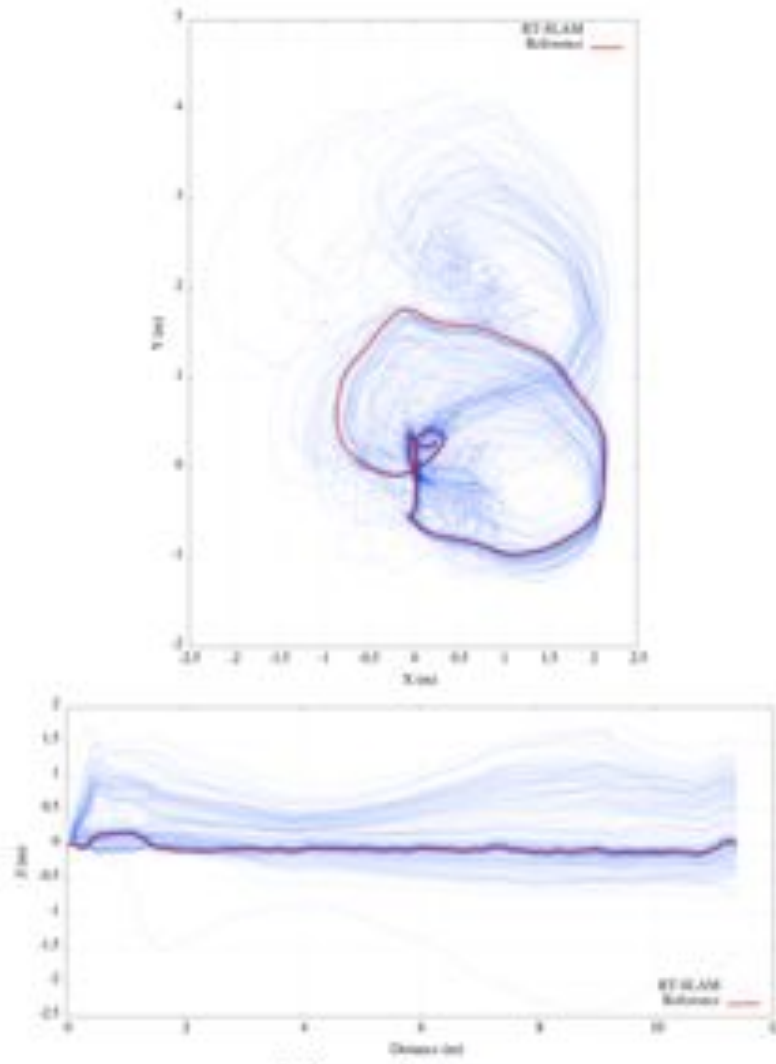
Need to estimate biases (OK for gyros, harder for accelerometers)

Need to estimate camera / INS relative position (within the filter is OK)

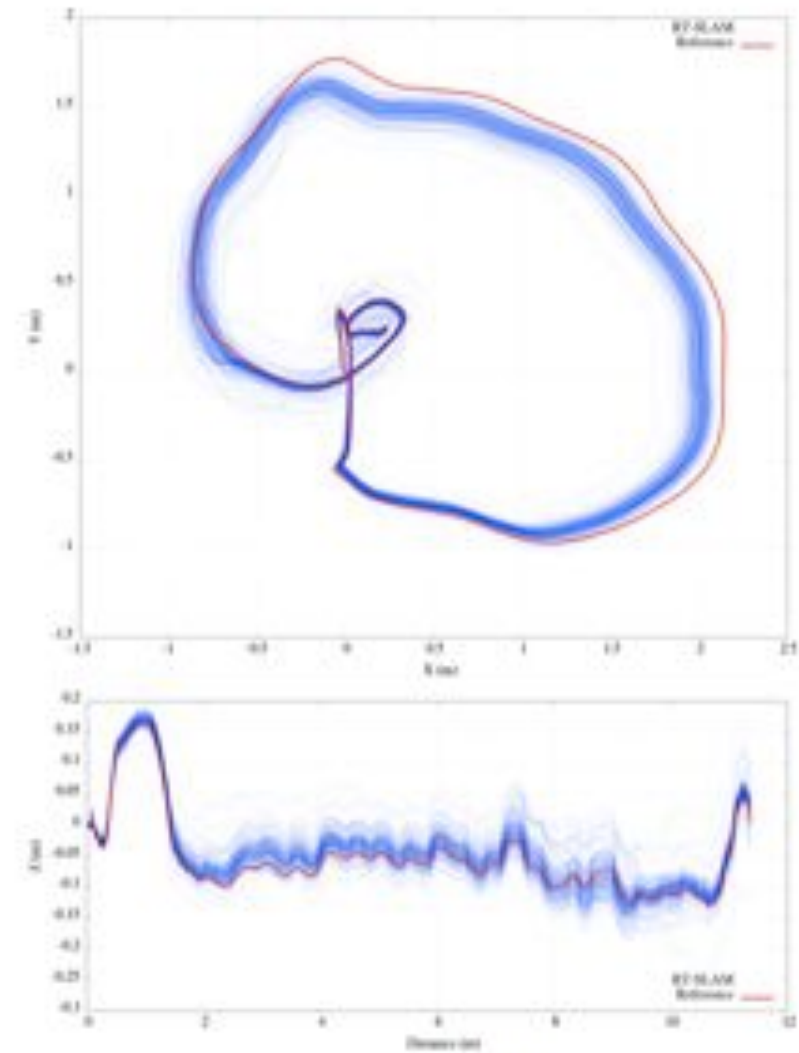
Synchronization might be an issue



# Prediction models



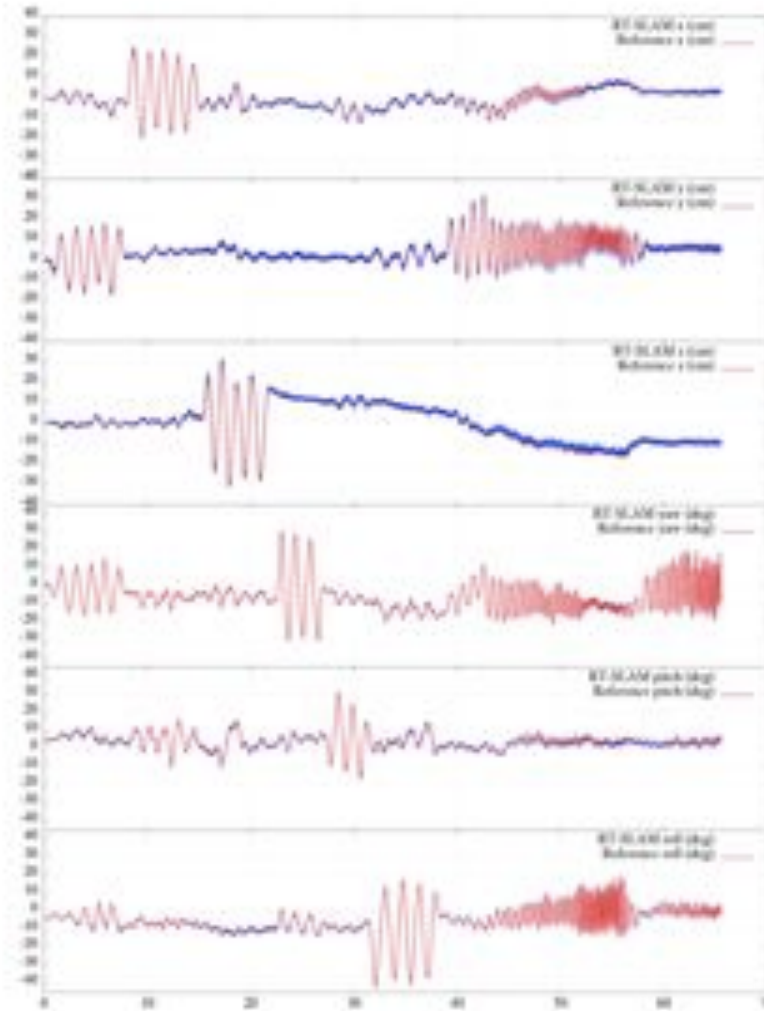
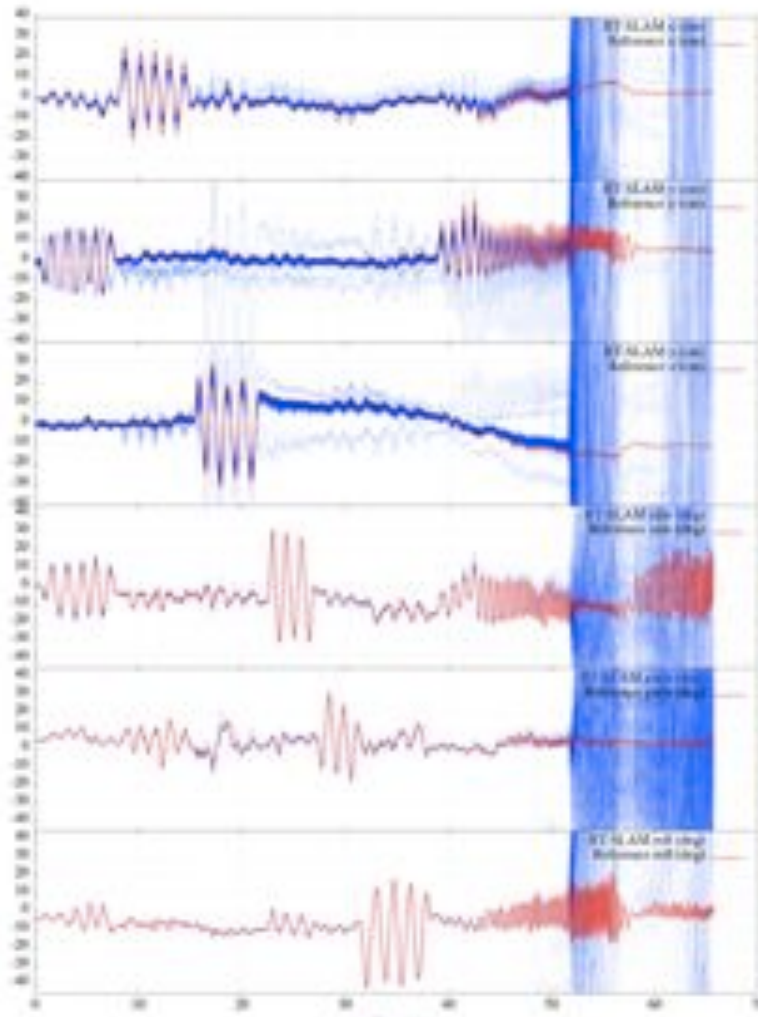
Visuel pur



Visuel-inertiel

# Prediction models

---



# Landmark models (points)

---

## Paramétrisation euclidienne

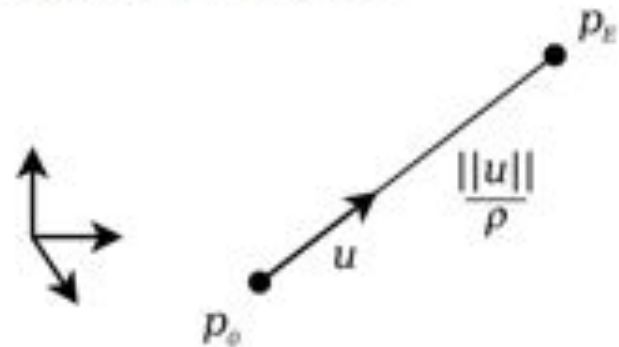
- ▶ Compacte et efficace :  $(x, y, z)$
- ▶ Impossibilité de représenter l'incertitude infinie sur la distance

## Paramétrisation AHP [Montiel 06] [Sola 10]

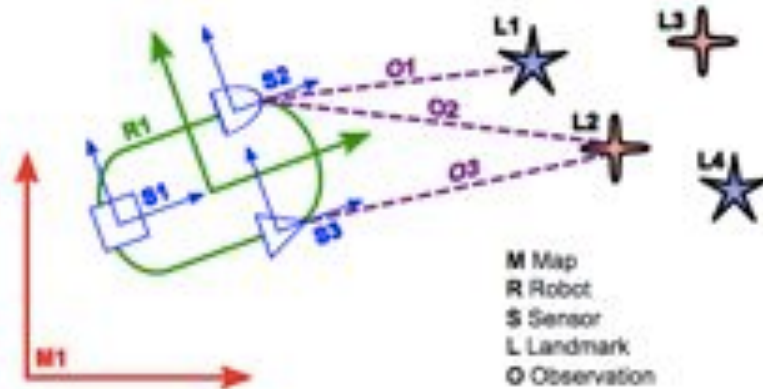
- ▶ Concentrer l'incertitude → relatif et polaire/homogène
- ▶ La représenter avec une gaussienne → distance inverse
- ▶ 7 paramètres :  $(\mathbf{p}_0, \mathbf{u}, \rho)$

## Reparamétrisation

- ▶ Selon critère de linéarité



# Managing data and landmarks



Chaque objet est générique

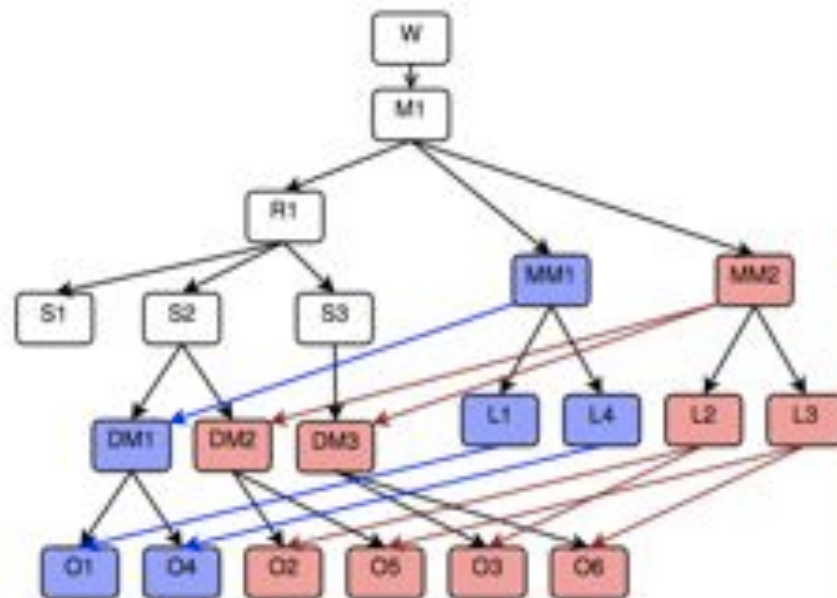
► **Carte** : moteur d'estimation

► **Robot** : modèle de prédiction

► **Capteur** : modèle, matériel

► **Amer** : modèle, descripteur

► **Observation** : modèle, variables intermédiaires



► **DataManager** : stratégie d'observation des amers

► **MapManager** : stratégie de gestion de vie des amers

# Managing data and landmarks

---

## Recherche active [Davison 07]

- ▶ Exploiter les infos a priori du filtre pour diriger la recherche
- ▶ Réduit le temps d'appariement  
Permet une gestion temps réel dur



## Élimination des éléments aberrants

- ▶ Important pour ne pas corrompre le filtre
- ▶ Contrôle de compatibilité (*gating*)
- ▶ One-point RANSAC [Civera 09]

# Managing data and landmarks

---

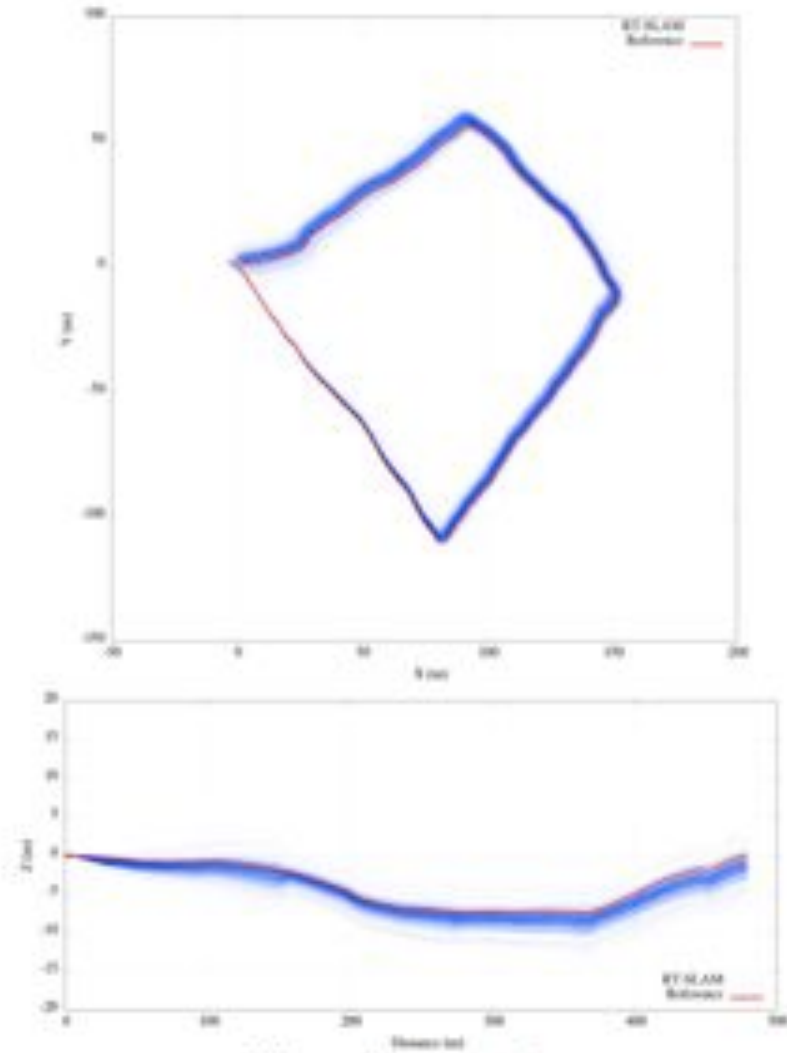
## Optimiser l'observabilité

- ▶ Création d'amers. . .
- ▶ pour assurer une bonne répartition. . .
- ▶ dans l'image.

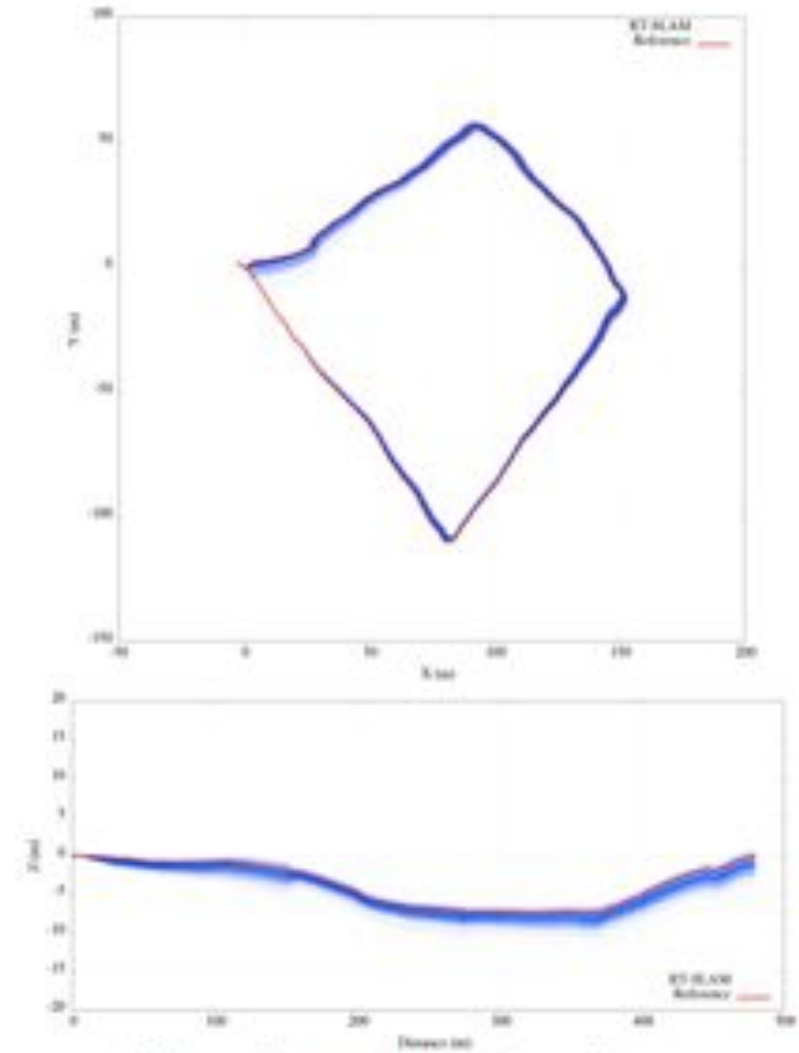
## Limiter le nombre d'amers

- ▶ Suppression d'amers. . .
  - ▶ pour limiter dynamiquement la densité. . .
  - ▶ dans la carte.
-

# Adding other sensors: odometry



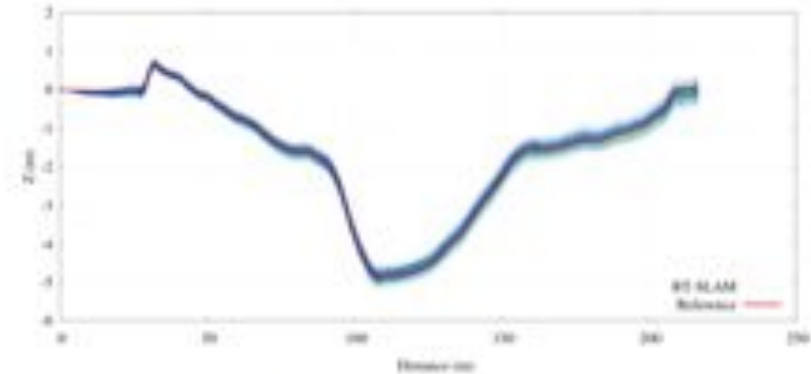
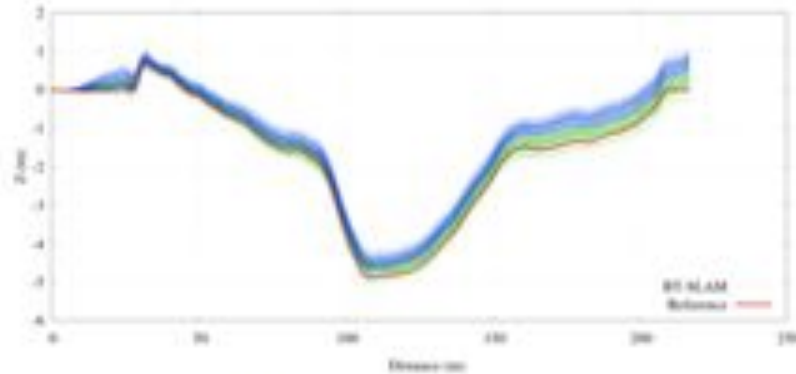
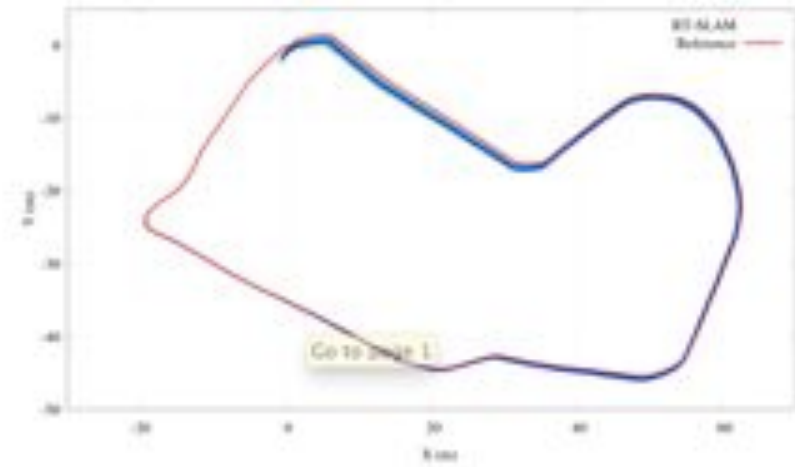
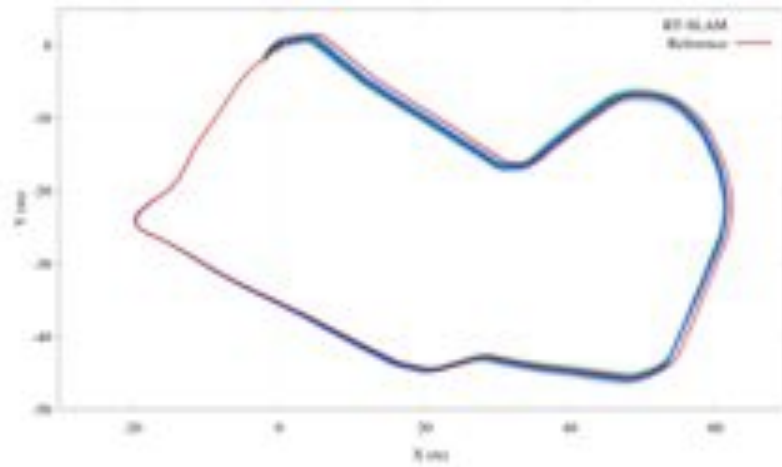
Visuel-inertiel



Visuel-inertiel-odométrie

# Adding other sensors: more cameras

---

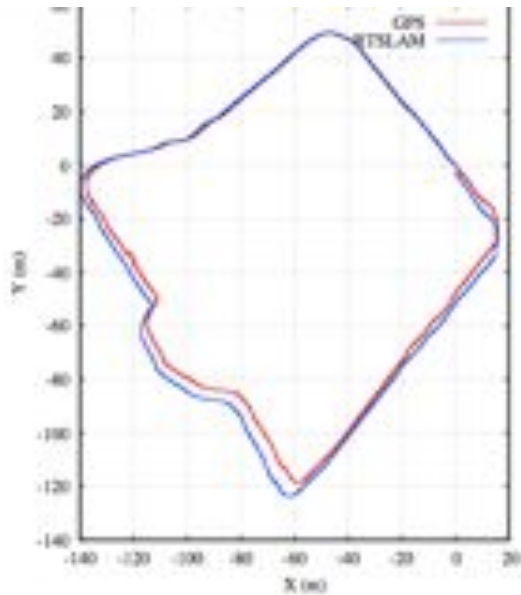


Visuel-inertiel-odométrie

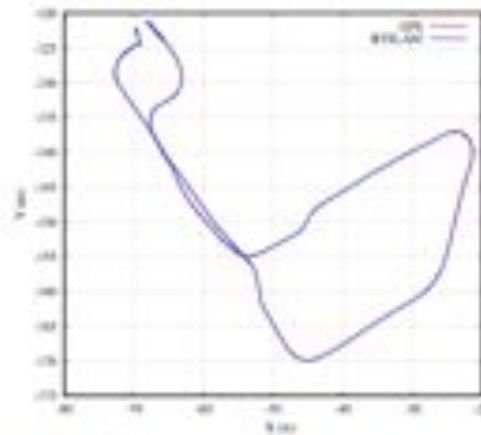
Bicam-inertiel



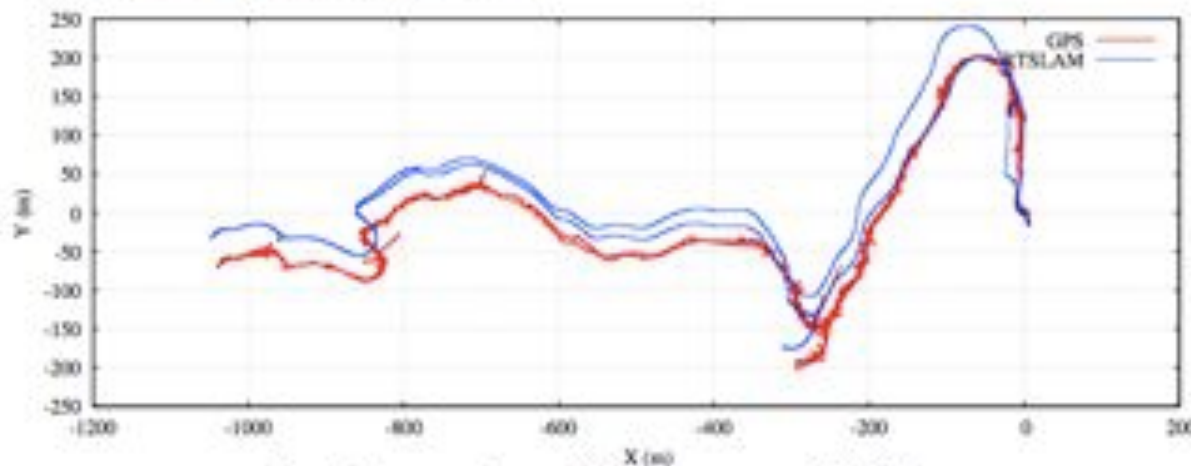
# Misc results



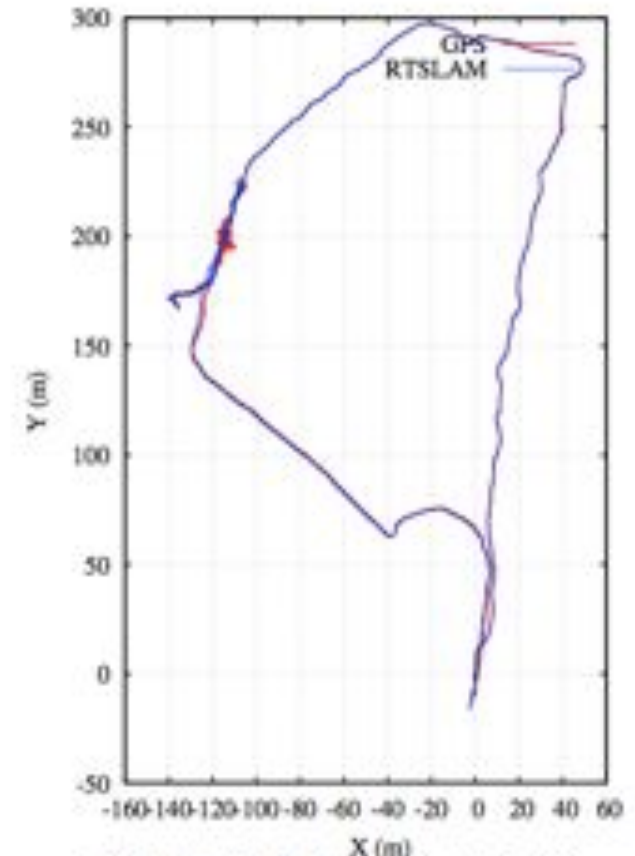
Caylus, 15k images, 450 m



LAAS, 6k images, 130 m



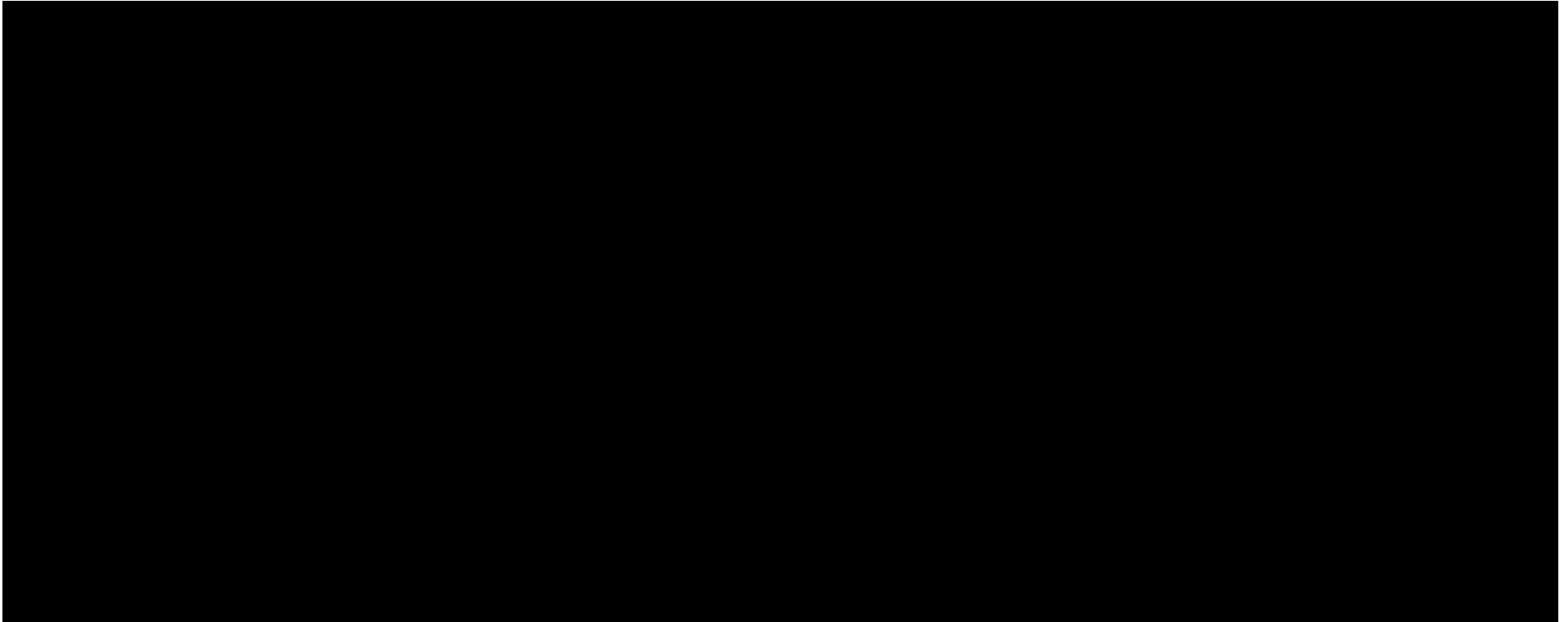
Berchtesgaden, 100k images, 3000 m



Thun, 70k images, 800 m

## Misc results

---



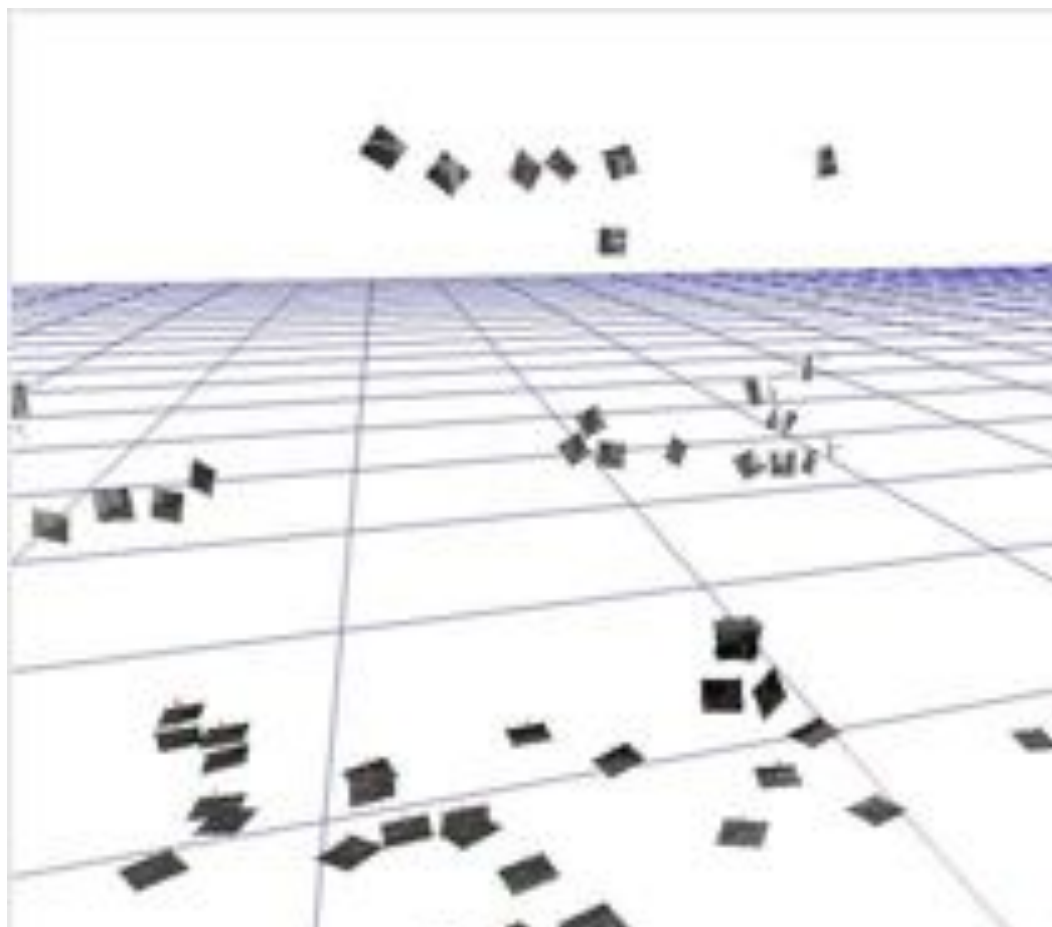
## Une implémentation de SLAM-EKF...

- ▶ **générique** (capteurs, amers, stratégies)
  - ▶ **optimisée** (temps réel à 60-100 fps, VGA, niveau de gris)
  - ▶ **robuste** (répartition optimale des amers, sécurités pour erreurs d'appariement, multicapteurs)
  - ▶ **developer-friendly** (visualisation, rejeu, log, simulation)
  - ▶ **open-source** (<http://rtslam.openrobots.org>)
- 
- Did RT-Slam push the visual EKF-based slam solutions to their limits?
  - Main issue remains integrity ☹️

# Pushing the M of SLAM

---

Typical SLAM maps



+

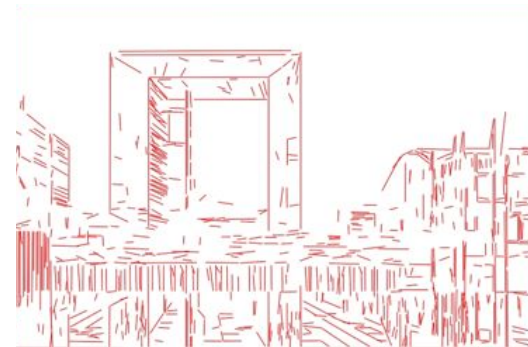
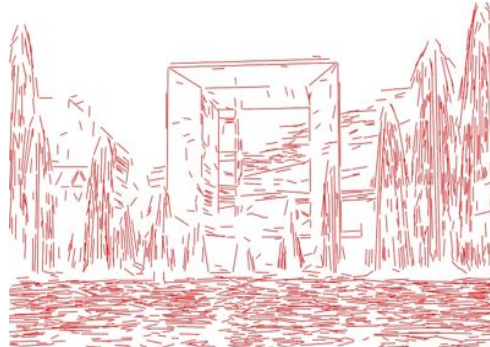
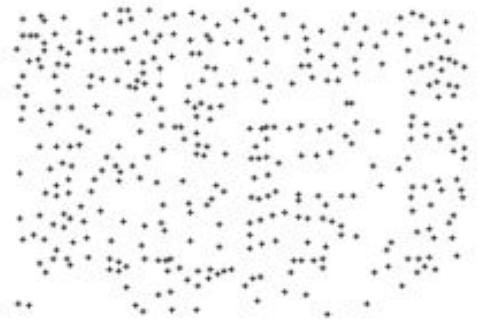


*Visual point landmark and indexed images :  
“robocentric” or “sensorcentric”*

# Pushing the M of SLAM

---

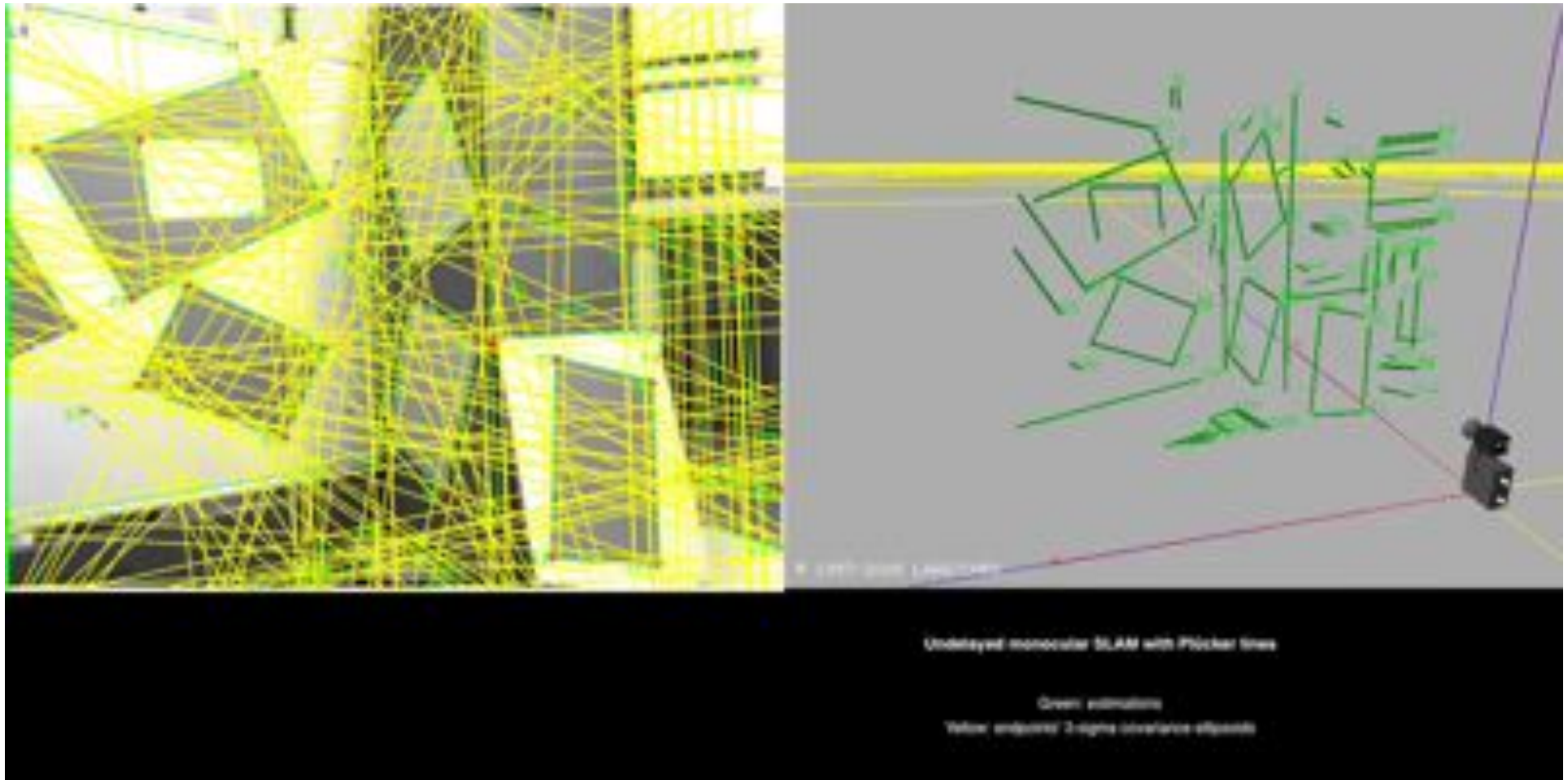
Points vs. line segments in vision



# Pushing the M of SLAM

---

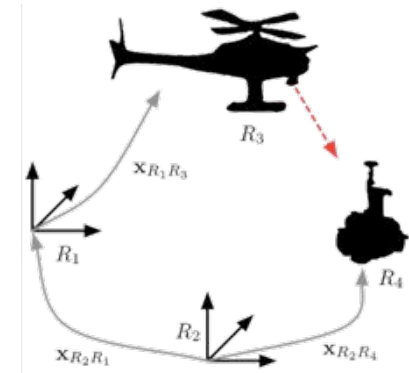
Using line segments in RT-Slam



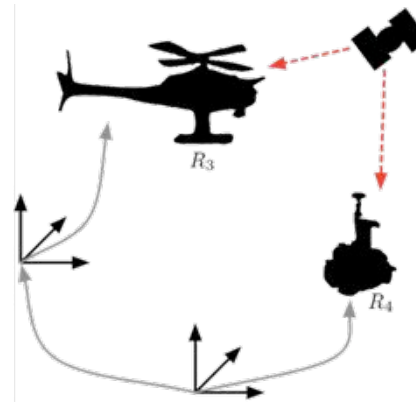
# Multi-vehicle SLAM solutions

## ➔ Various loop-closing events

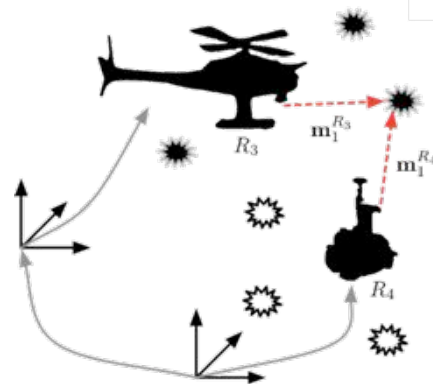
“Rendez-vous”: inter-robot pose estimation



Absolute localization  
(GPS fix / localization  
wrt. an initial map)



Inter-robot landmark  
(or map) matches



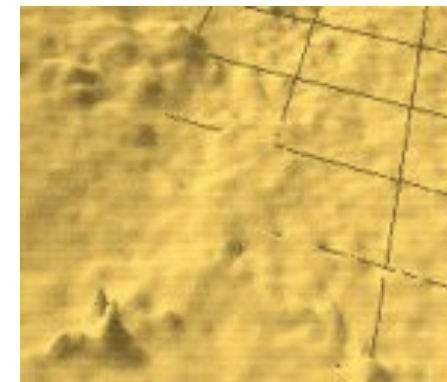
# Map-based localization

---

Robot's life is not only about SLAM...

Exploiting available data

- Geographic information systems
- Image databases





# More generally on localization

---

- A variety of needs
- A variety of solutions
  
- Need to develop an overall framework that integrates the solutions and allows their control (in a distributed multi-robot context)

Are SLAM processes at the heart  
of such a framework?

# Back to vision-based SLAM

---

- Loop closure: the magic in SLAM, but actually better processed (or focused) with additional techniques (*e.g.* image indexing)
- Impressive direct (featureless) approaches:
  - Andrew Comport @ I3S, (Nice)
  - Davide Scaramuzza @ ETHZ (Zürich)
- Vision remains fragile (important availability and integrity issues)
- Towards ultra high rate localization ( $> 1$  kHz)
  - A (necessary) shift in camera paradigms ?
  - *e.g.* Dynamic Vision Sensors (DVS)

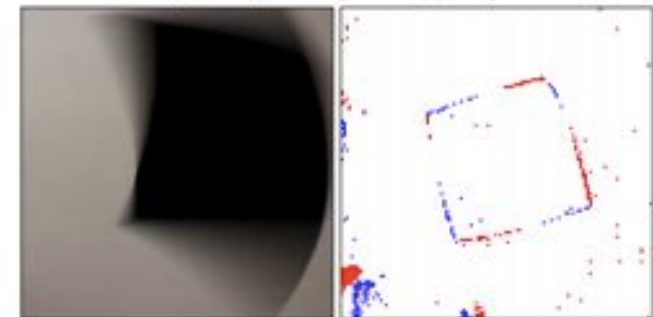
# Back to vision-based SLAM

---

- Loop closure: the magic in SLAM, better processed (or focused) with additional techniques (*e.g.* Image indexing)
- Vision remains fragile (important availability and integrity issues)
- Impressive direct (featureless) approaches:
  - Andrew Comport @ I3S, (Nice)
  - Davide Scaramuzza @ ETHZ (Zürich)
- Towards ultra high rate localization ( $> 1$  kHz)
  - A (necessary) shift in camera paradigms ?
  - *e.g.* Dynamic Vision Sensors (DVS)



(a) Our quadrotor performing a flip.



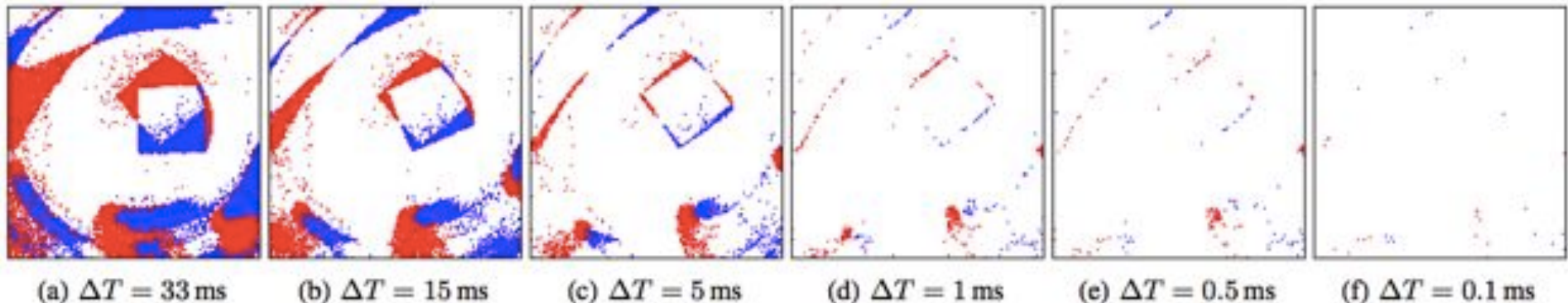
(b) Standard CMOS camera

(c) Integrated DVS events (2 ms)

# Back to vision-based SLAM

---

- Loop closure: the magic in SLAM, better processed (or focused) with additional techniques (*e.g.* Image indexing)
- Vision remains fragile (important availability and integrity issues)
- Impressive direct (featureless) approaches:
  - Andrew Comport @ I3S, (Nice)
  - Davide Scaramuzza @ ETHZ (Zürich)
- Towards ultra high rate localization ( $> 1$  kHz)
  - A (necessary) shift in camera paradigms ?
  - *e.g.* Dynamic Vision Sensors (DVS)



- On the importance of localization
- RT-SLAM
- Outlook
  - Pushing the M of SLAM
  - Consider (distributed) multi-vehicle solutions
  - Use existing maps
  - Innovative solutions in vision