



HAL
open science

Vision-based navigation

Simon Lacroix

► **To cite this version:**

Simon Lacroix. Vision-based navigation. Doctoral. Spring School on Location-Based Services, École Nationale d'Aviation Civile, Toulouse (France), France. 2015. cel-01276880

HAL Id: cel-01276880

<https://hal.science/cel-01276880>

Submitted on 20 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vision-based navigation

Simon Lacroix
Robotics and InteractionS (RIS)
LAAS/CNRS, Toulouse

<http://homepages.laas.fr/simon/>



Where do I come from?

Robotics at LAAS/CNRS, Toulouse, France

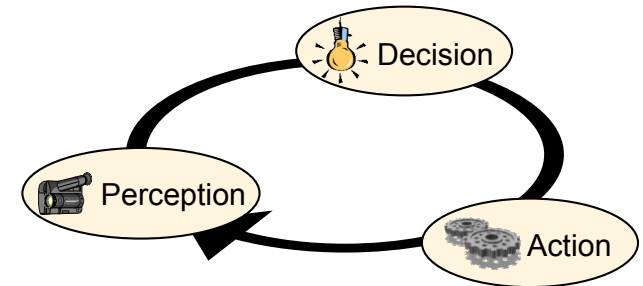
- Research topics

- Perception, planning and decision-making, control
- Plus: control architecture, interactions, ambient intelligence systems, learning

- Research domains

- Cognitive and interactive Robotics
- Aerial and Terrestrial Field Robotics
- Human and anthropomorphic motion
- Bio-informatics, Molecular motion

- Considered applications: Planetary exploration, Service and personal robotics, virtual worlds and animation, biochemistry, embedded systems, transport, driver assistance, defense, civil safety



A keyword: **autonomy**

3 research groups :

12 full time researchers

10 university researchers

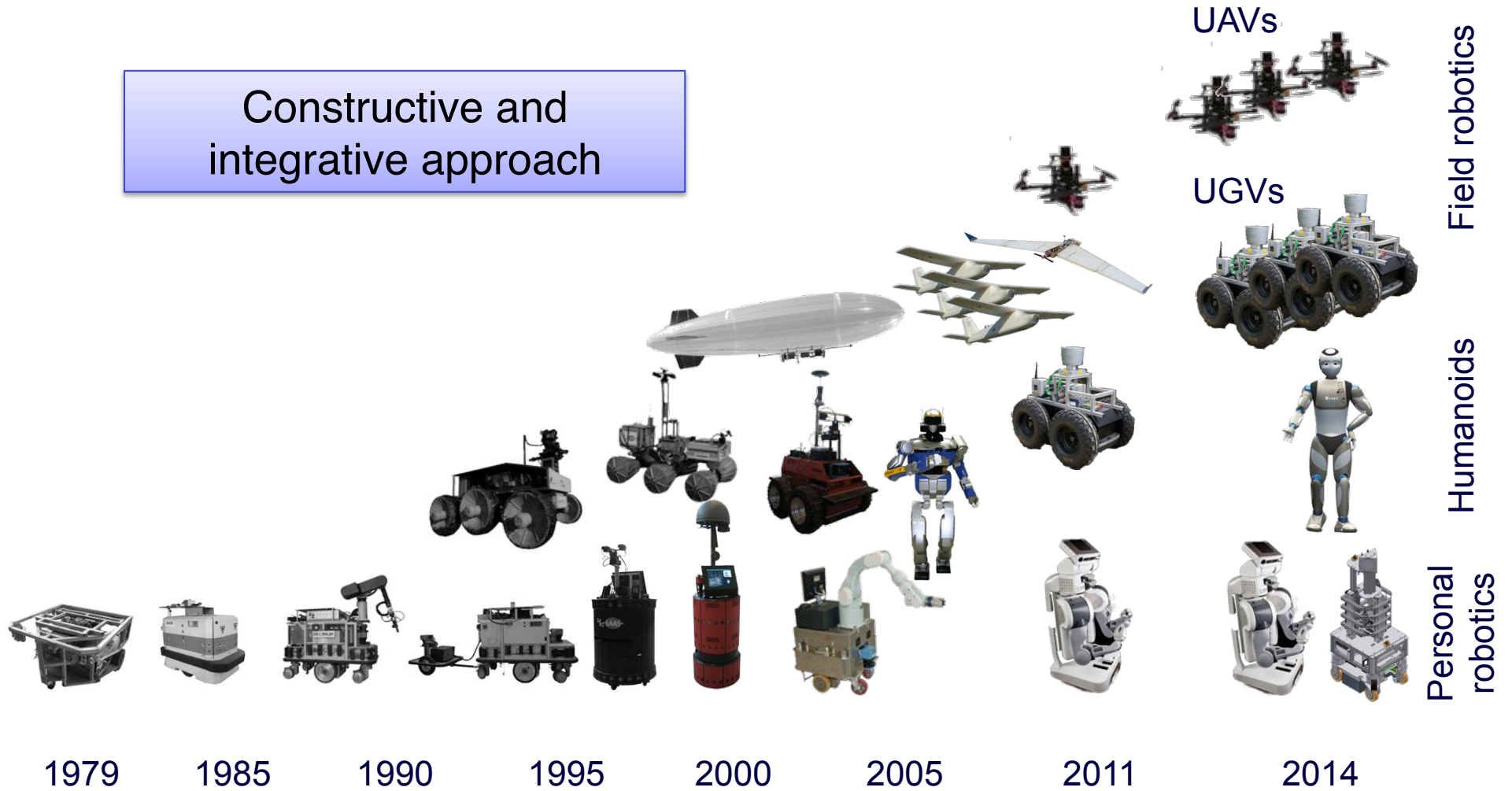
4 visitors

50 PhD students

10 post-docs

Robots @ LAAS

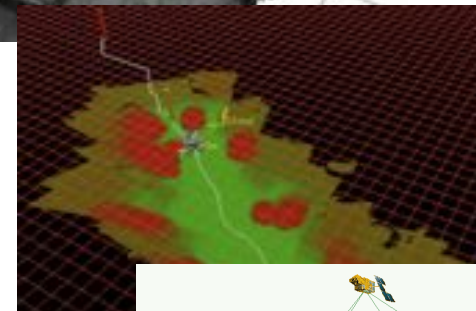
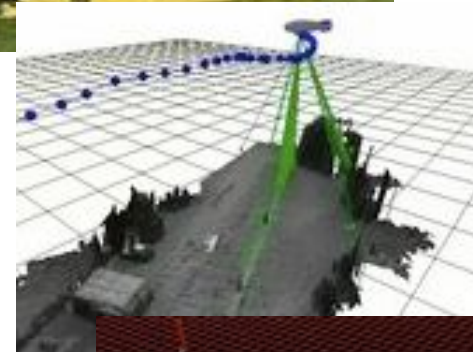
Constructive and integrative approach



Open source software tools: www.openrobots.org

What am I working on? Field robotics

- Environment perception and modeling
- Localization and SLAM
- Autonomous rover navigation
- Multi-Robot cooperation

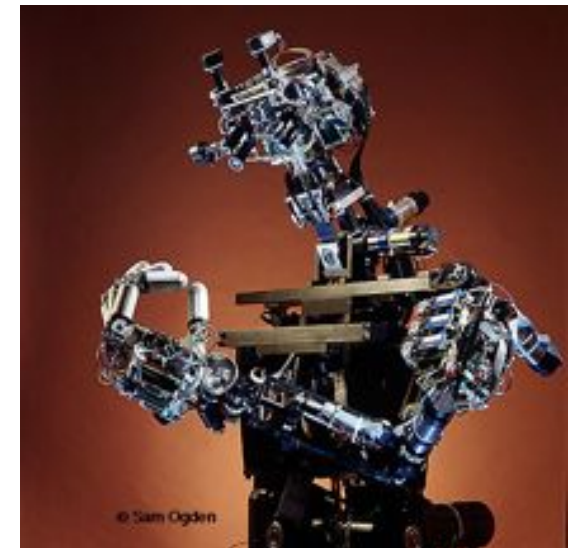
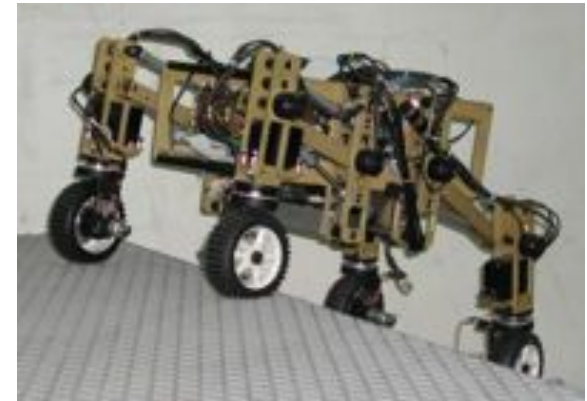
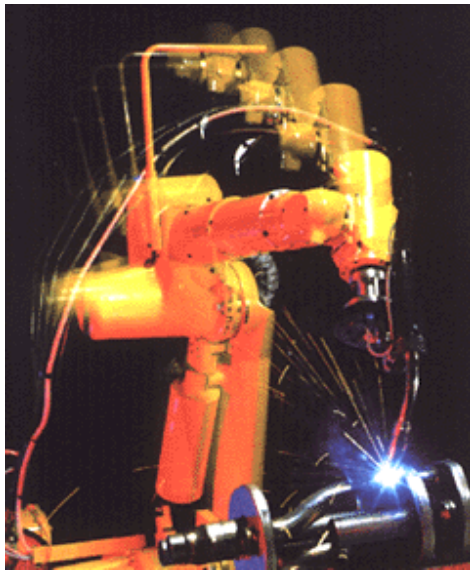


Robotics

“From automatic control to autonomous control”

In laboratories

Industrial



Robotics

“From automatic control to autonomous control”



Robots everywhere

Robotics

“From automatic control to autonomous control”

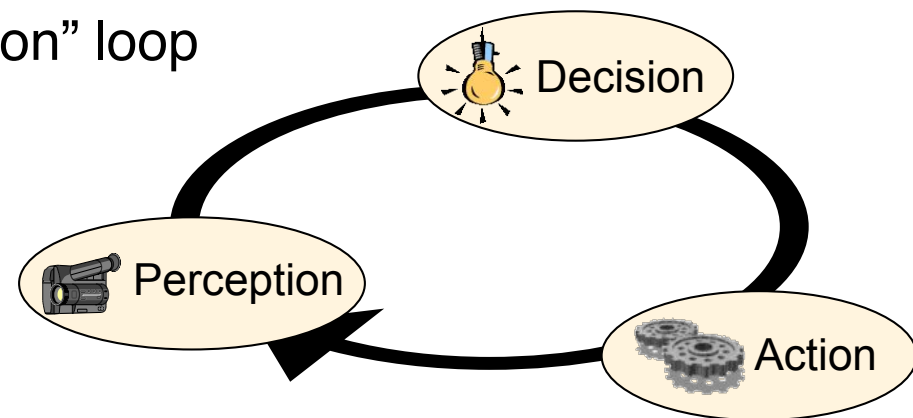
- Automatic control :
 - Well defined task (“*regulate variable*”, “*follow trajectory*”...)
 - “Direct” link between (simple) perception and action

- 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
- ...

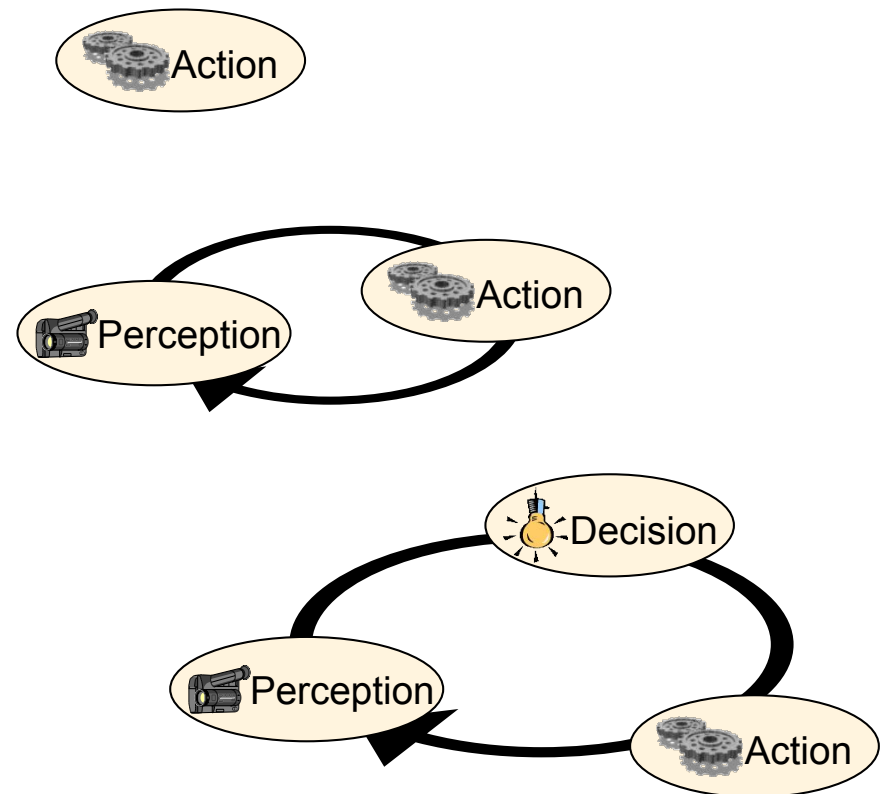


Robotics

“From automatic control to autonomous control”

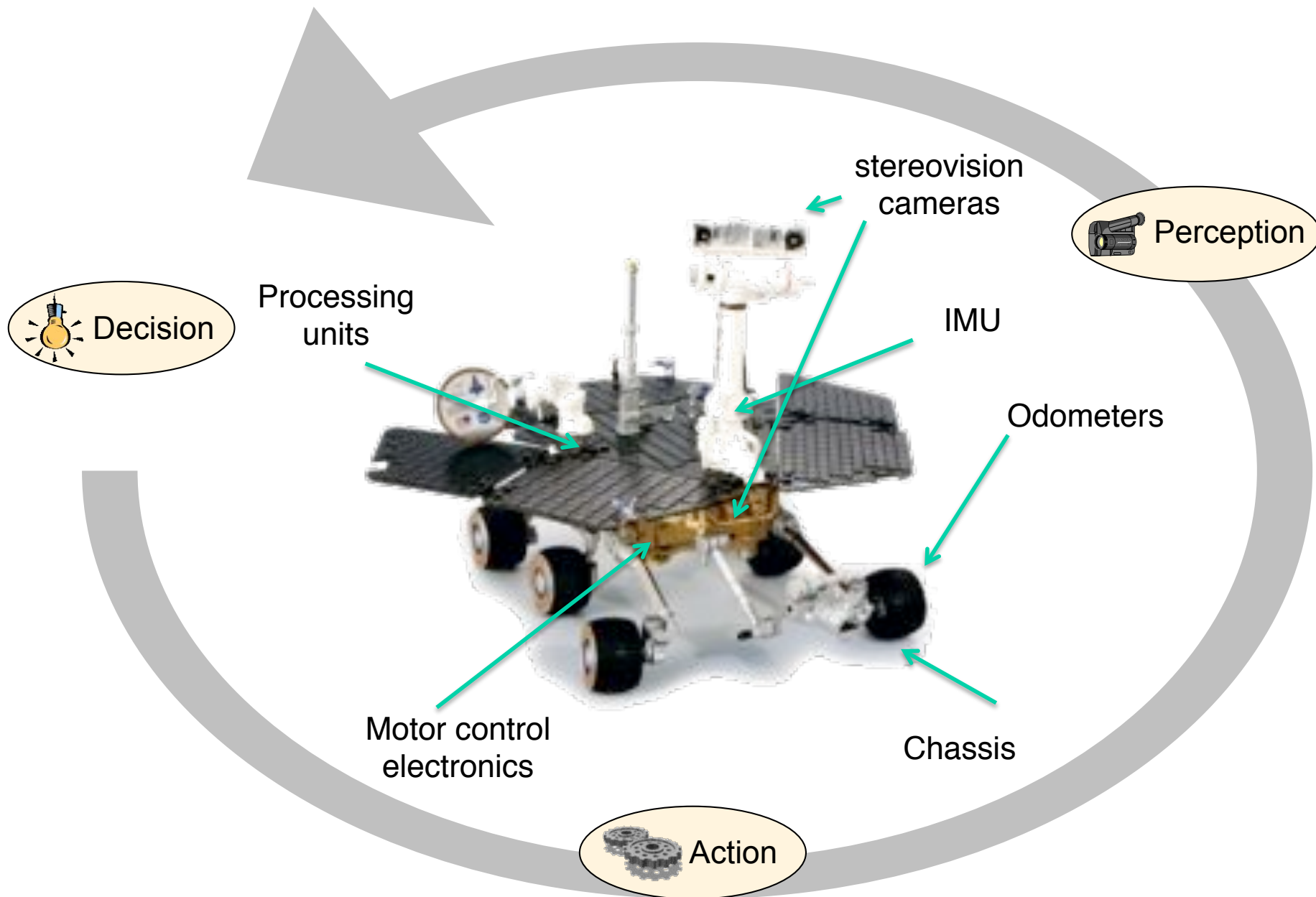
E.g. for a drone:

- Regulate heading / speed / altitude
- Follow a list ordered waypoints
- Follow a geometric trajectory
- Follow a road
- Follow a target
- Survey an area while avoiding threats and obstacles



“Decision”: notion of deliberation, planning, prediction and evaluation of the outcomes of an action

Anatomy of a robot



Anatomy of a robot

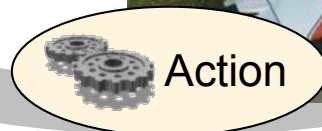
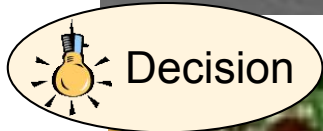


Illustration: autonomous navigation

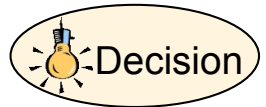
nav•i•ga•tion |,navi'gā sh ən|

noun

1 the process or activity of accurately ascertaining one's position and planning and following a route.



detect obstacles, traversable areas, localize the robot



avoid obstacles, find trajectories, itineraries

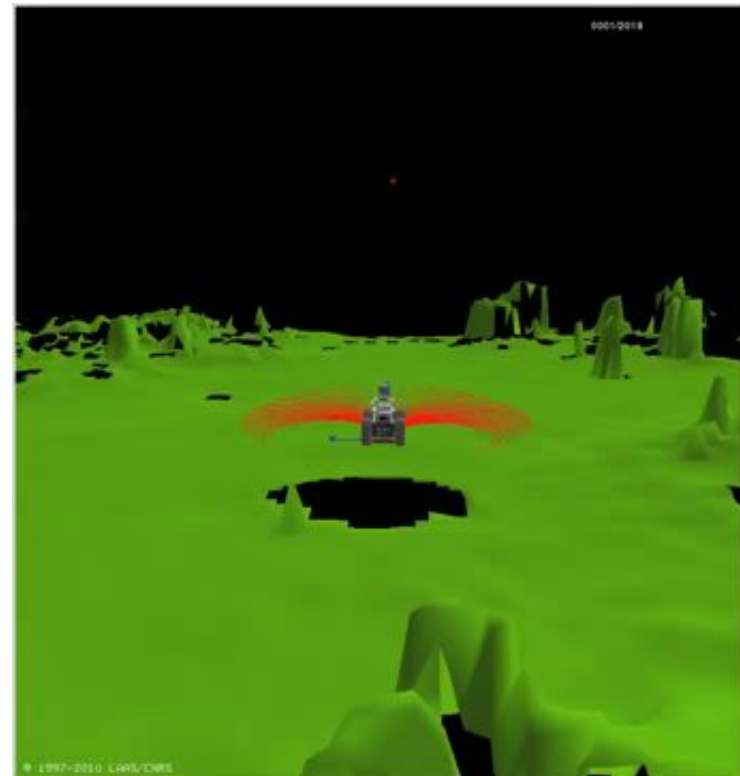
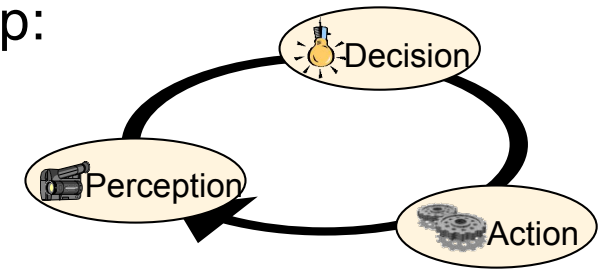


ensure the execution of the planned motions

An elementary decision: AGV obstacle avoidance

Simple instance of a perception / decision / action loop:

- Gathering data on the environment
- Structuring the data into a *model*
- *Planning* the trajectory to find the “optimal” one
- Executing the trajectory



Perception in robotics

Perception :

« Acquisition and representation of information on the environment and the robot itself »

Proprioceptive and exteroceptive sensors:

pro•pri•o•cep•tive |ˌprɒprɪə'septɪv|

adjective Physiology

relating to stimuli that are produced and perceived within an organism, esp. those connected with the position and movement of the body. Compare with **EXTEROCEPTIVE** and **INTEROCEPTIVE** .

ex•ter•o•cep•tive |ˌɛkstərɔ'septɪv|

adjective Physiology

relating to stimuli that are external to an organism. Compare with **INTEROCEPTIVE** .

Why is localization so important?



Why is localization so important *in robotics?*

Localization is required to:

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

But... what 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” |
| 4. <i>Integrity</i> of the solution? | |
| 5. <i>Disponibility</i> of the solution? | |

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

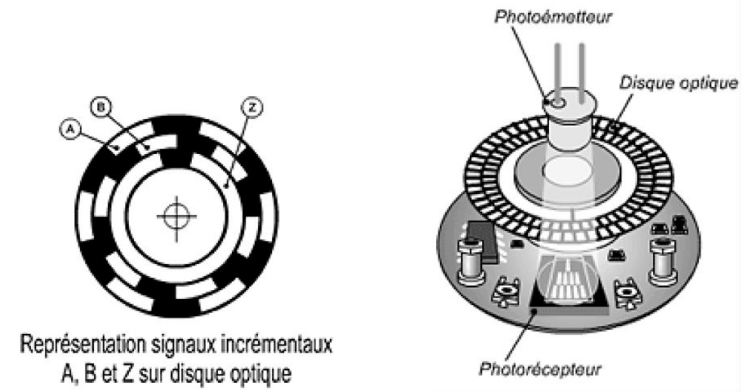
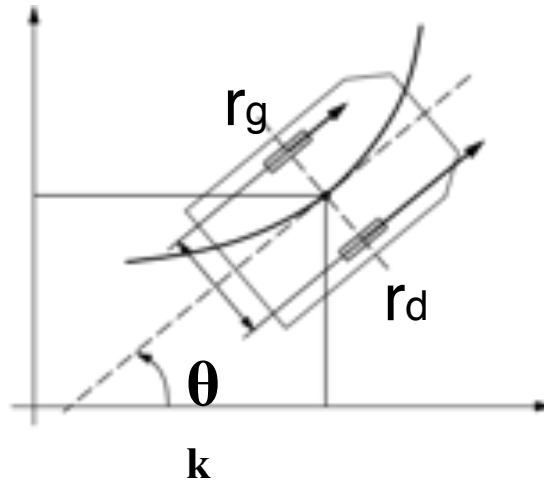
- Ensure the achievement of the missions, most often defined in localization terms (“goto [goal]”, “explore / monitor [area]”, ...)

Outline

- Autonomous robots
- On the importance of localization
- **Localization using dead-reckoning approaches**

Odometry

- Odometry: estimation of (x, y, θ) by integration of elementary motions

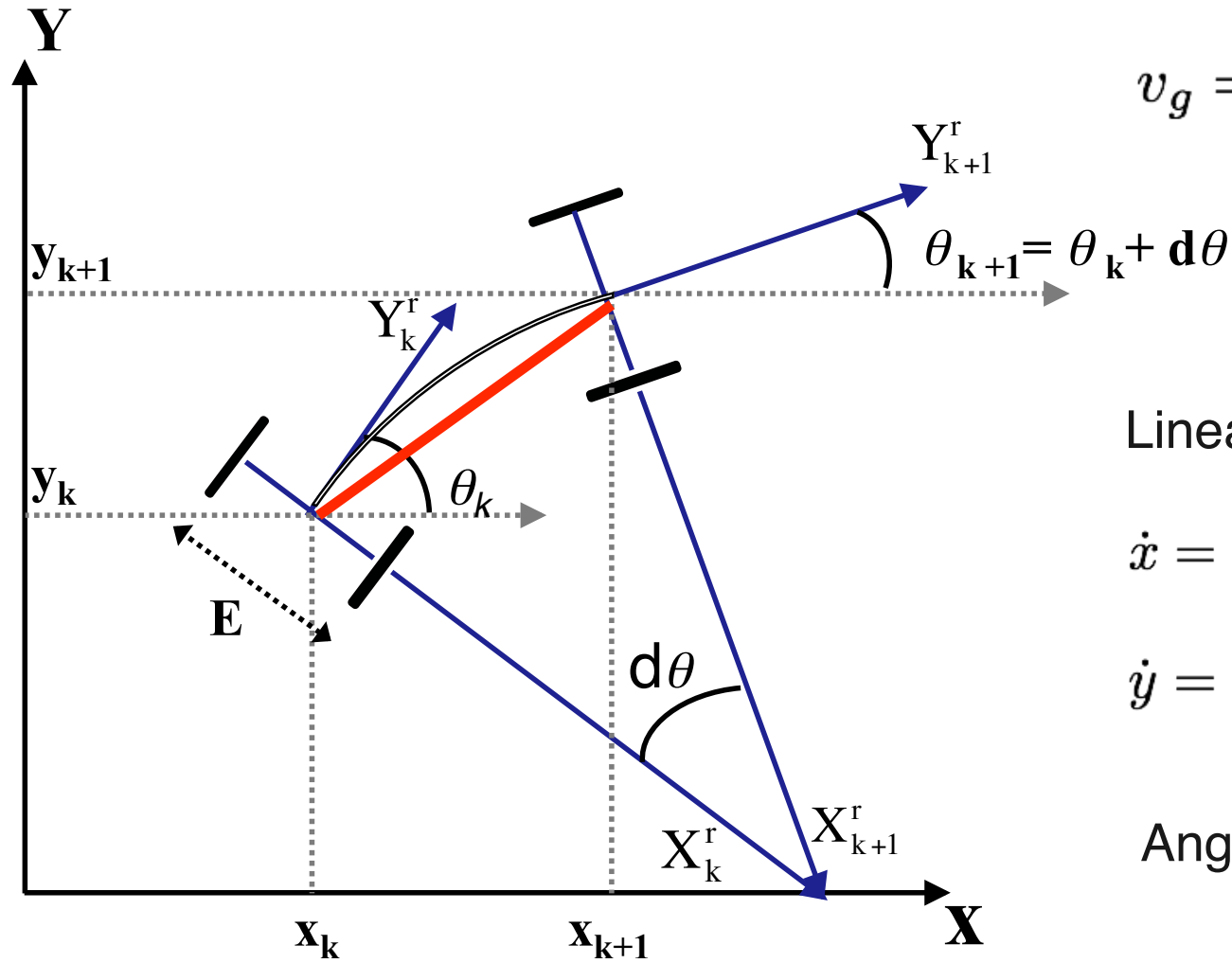


Wheel rotation encoders

Odometry

Linear wheel speeds:

$$v_g = r_g \frac{dq_g}{dt} \quad v_d = r_d \frac{dq_d}{dt}$$



Linear speed:

$$\dot{x} = \frac{1}{2}(v_g + v_d)\cos\left(\theta_k + \frac{d\theta}{2}\right)$$

$$\dot{y} = \frac{1}{2}(v_g + v_d)\sin\left(\theta_k + \frac{d\theta}{2}\right)$$

Angular speed:

$$\dot{\theta} = \frac{v_g - v_d}{E}$$

Odometry

- Example: linear motion



- Measured distance: $d_1 = \hat{d}_1 + \tilde{d}_1$

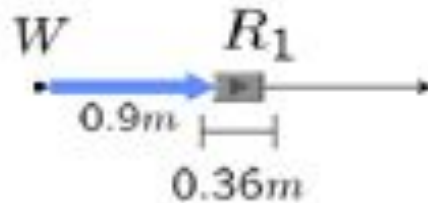
- Error model:
$$E(\tilde{d}_1) = \mu_1$$
$$\text{Var}(\tilde{d}_1) = \sigma_1^2$$

For instance:

$$\mu_1 = 0$$
$$\sigma_1 = 0.1 \cdot \hat{d}_1$$

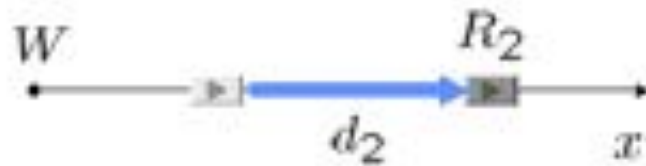
Odometry

- Example: move 0.9m



$$\begin{aligned}\hat{x}_{WR_1} &= \hat{d}_1 \\ &= 0.9m \\ \sigma_{x_{WR_1}} &= 0.09m\end{aligned}$$

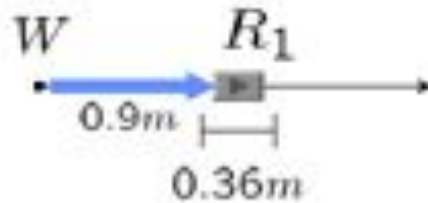
- Robot moves again 0.85m: $\hat{d}_2 = 0.85m$



$$\begin{aligned}x_{WR_2} &= \hat{d}_1 + \tilde{d}_1 + \hat{d}_2 + \tilde{d}_2 \\ E(\tilde{d}_1 + \tilde{d}_2) &= \mu_1 + \mu_2 = 0 \\ \text{Var}(\tilde{d}_1 + \tilde{d}_2) &= \sigma_1^2 + \sigma_2^2 \\ &= 0.1^2 (\tilde{d}_1^2 + \tilde{d}_2^2)\end{aligned}$$

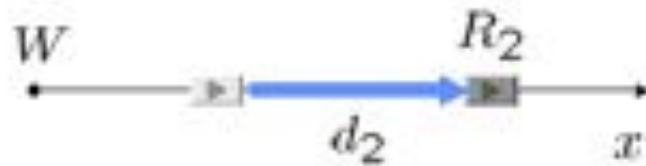
Odometry

- Example: move 0.9m



$$\begin{aligned}\hat{x}_{WR_1} &= \hat{d}_1 \\ &= 0.9m \\ \sigma_{x_{WR_1}} &= 0.09m\end{aligned}$$

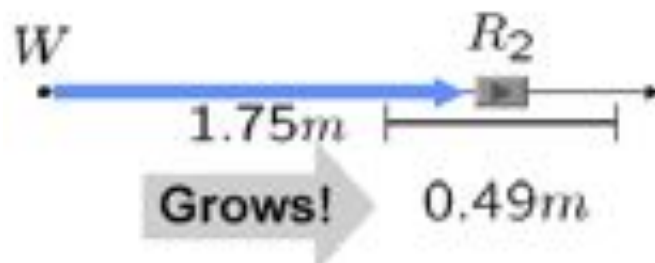
- Robot moves again 0.85m: $\hat{d}_2 = 0.85m$



$$x_{WR_2} = \tilde{d}_1 + \tilde{d}_1 + \tilde{d}_2 + \tilde{d}_2$$

$$\begin{aligned}E(\tilde{d}_1 + \tilde{d}_2) &= \mu_1 + \mu_2 = 0 \\ \text{Var}(\tilde{d}_1 + \tilde{d}_2) &= \sigma_1^2 + \sigma_2^2 \\ &= 0.1^2 (\tilde{d}_1^2 + \tilde{d}_2^2)\end{aligned}$$

- New estimation:

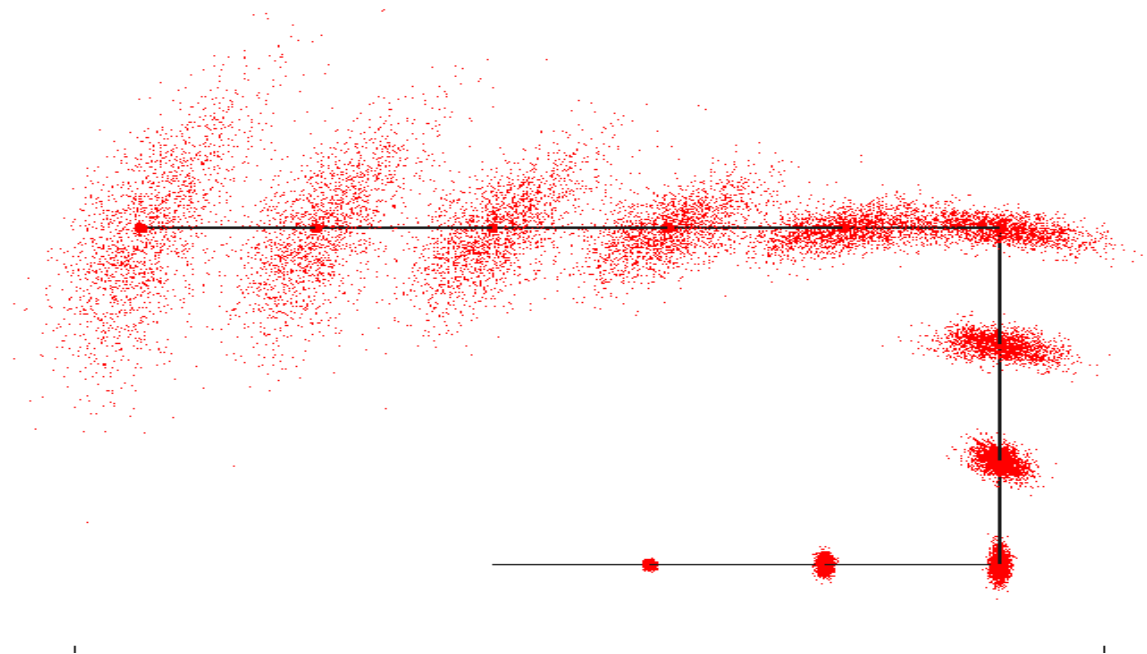


$$\begin{aligned}\hat{x}_{WR_2} &= 1.75m \\ \sigma_{x_{WR_2}} &\simeq 0.12m\end{aligned}$$

Odometry

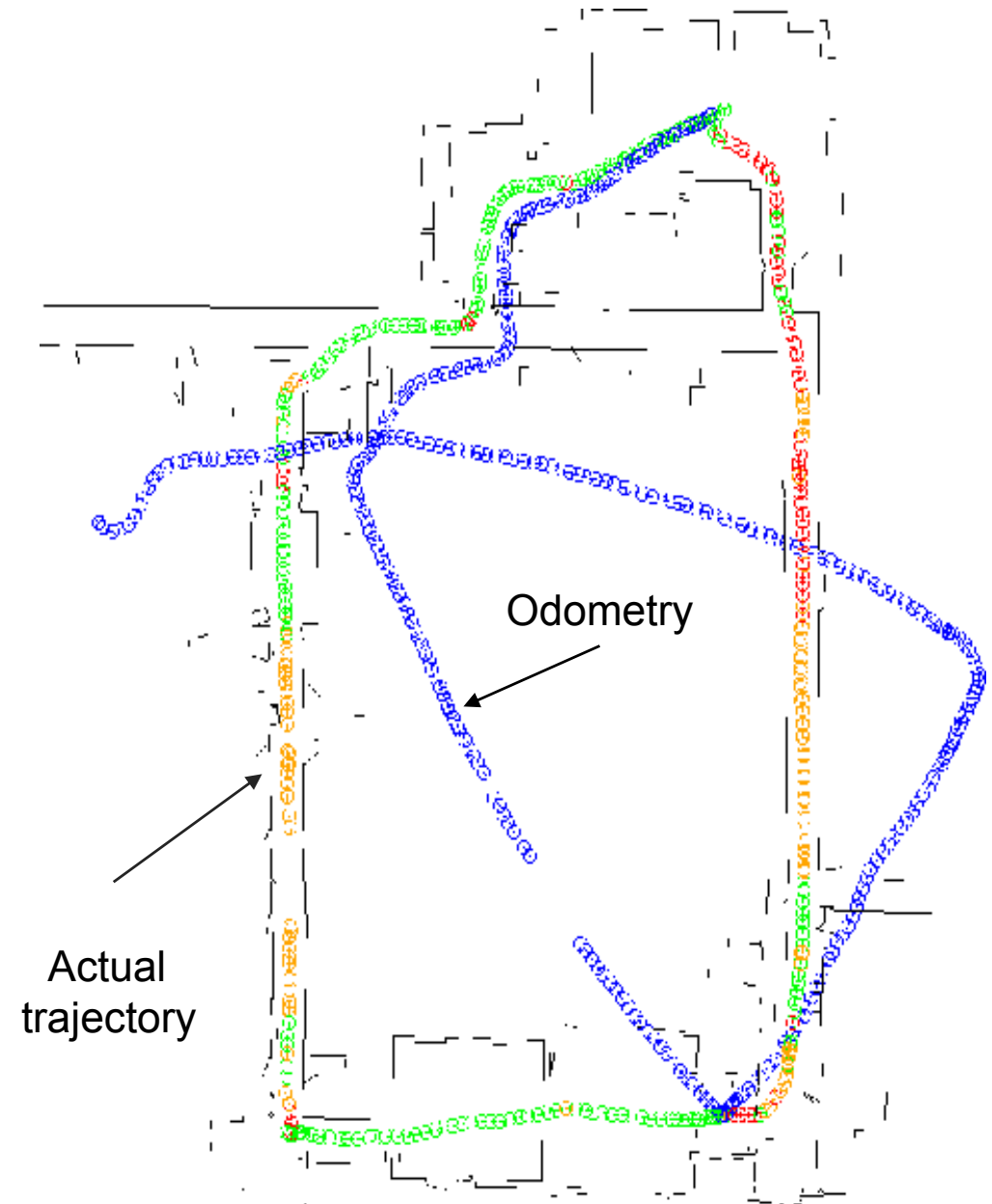
Now angular rotations come into play

Monte-Carlo simulation:



Odometry: illustration

- With an indoor robot



Localization solutions in robotics

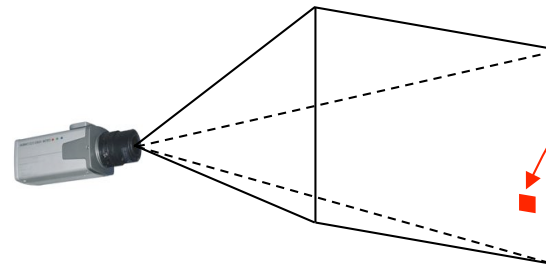
- Odometry
- Similarly, motion / accelerations sensors (inertial navigation)

Inherently drifts over time and distances, subject to slippages and skids

➔ Develop solutions relying on the robot exteroceptive sensors

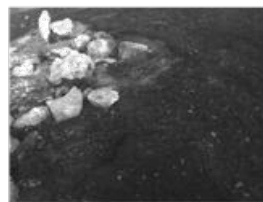
A few words on vision

- Cameras : low cost, light and power-saving
- Perceive data
 - In a volume
 - Very far
 - Very precisely

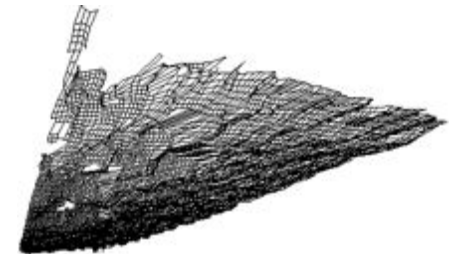


1024 x 1024 pixels
60° x 60° FOV
↓
0.06 ° pixel resolution
(1.0 cm at 10.0 m)

- Stereovision
 - 2 cameras provide depth



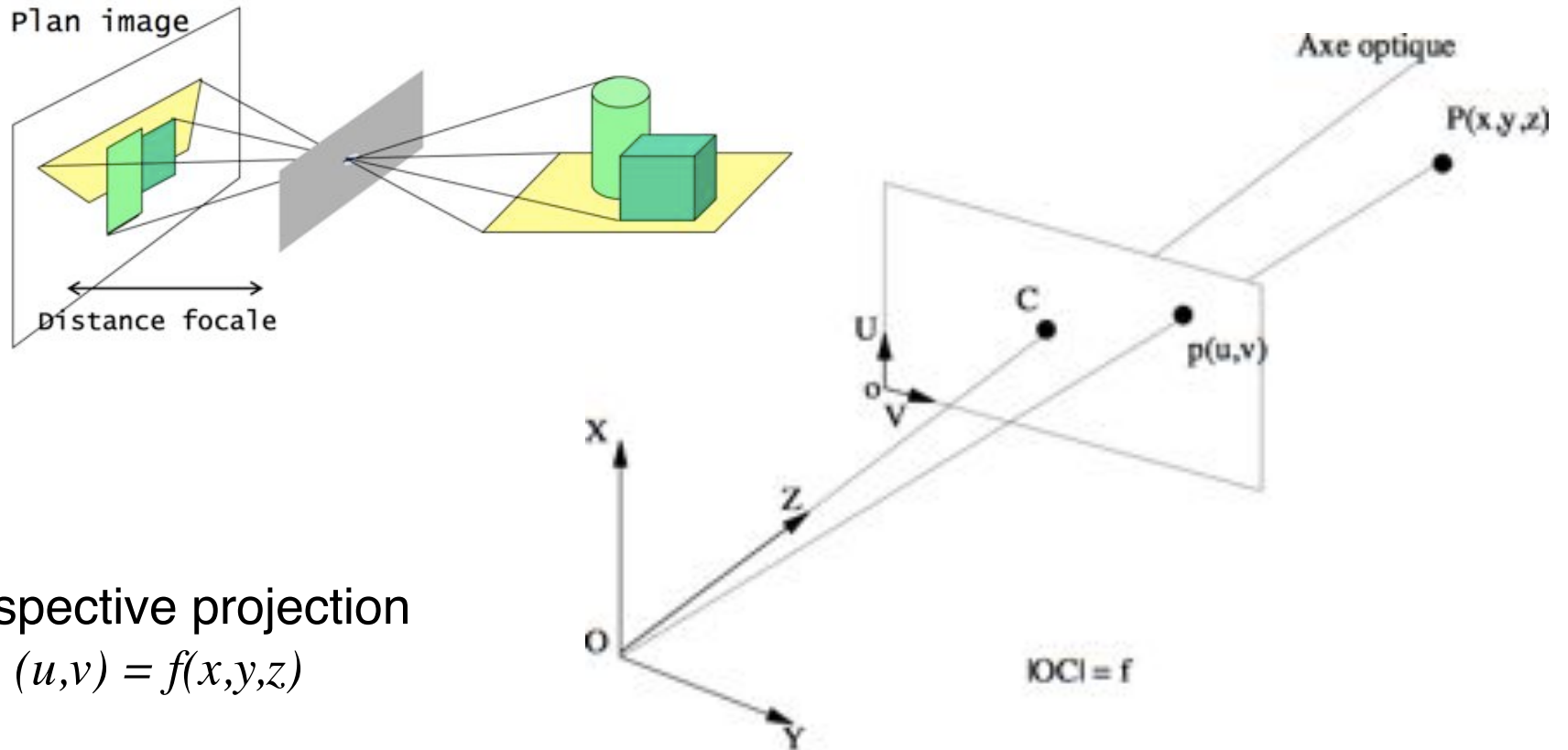
⊕



- Images carry a vast amount of information
- A vast know-how exists in the computer vision community

Camera geometric model

- Pinhole model



Perspective projection

$$(u,v) = f(x,y,z)$$

$$u = k_u f x / z + u_0$$

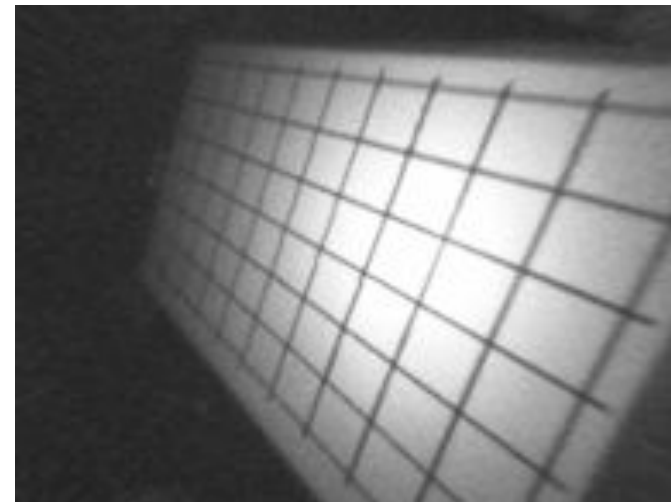
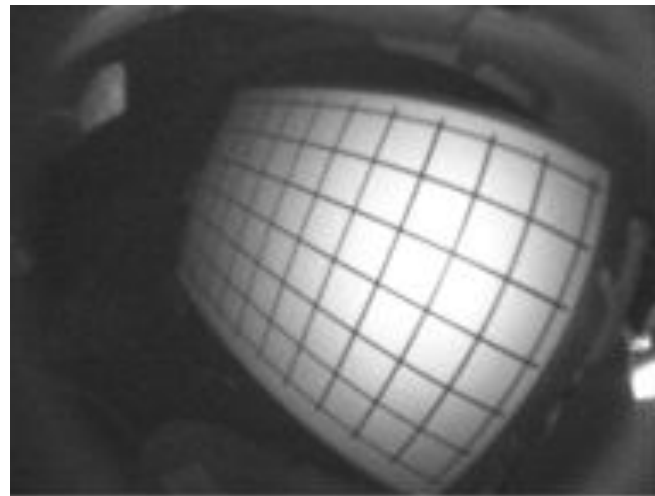
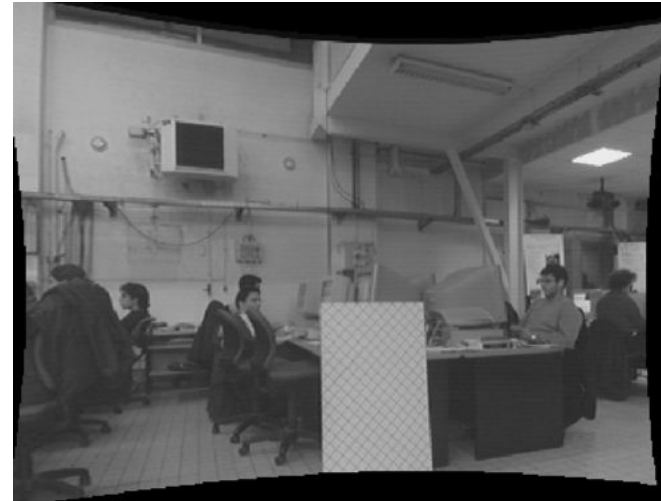
$$v = k_v f y / z + v_0$$

$$p = I_c P$$

I_c : camera intrinsic matrix

Camera geometric model

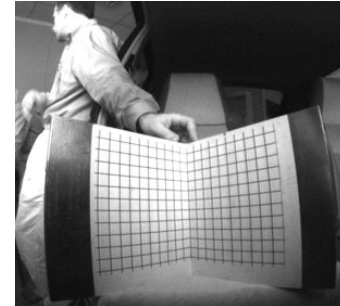
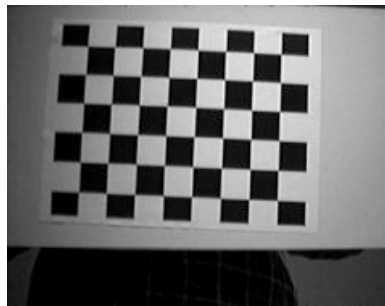
(But the real world is more complex)



Camera calibration

- Principle
 1. Acquisition of images of a known calibration pattern
 2. Extraction of the pattern features
 3. Association between extracted features and pattern features
 4. Use of a minimization technique to estimate the projection parameters

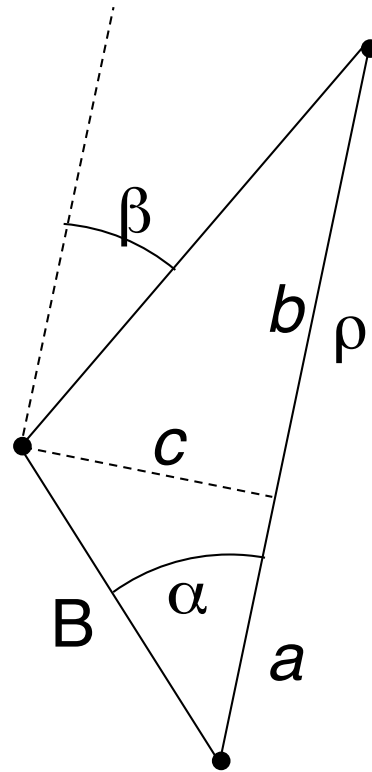
Classic
calibration
patterns



Well known techniques, available softwares on line (e.g. Matlab calibration toolbox, openCV library)

3D data from vision

- Triangulation



$$\rho = B \sin \alpha (1/\operatorname{tg} \beta + 1/\operatorname{tg} \alpha)$$

3D data from vision

- Triangulation

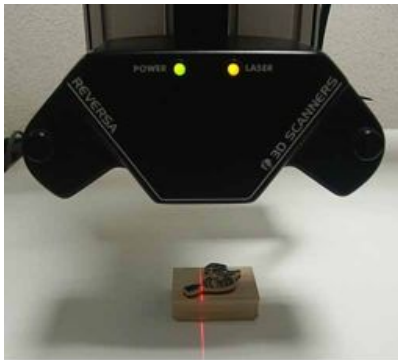


Figure 3 : Triangulation laser

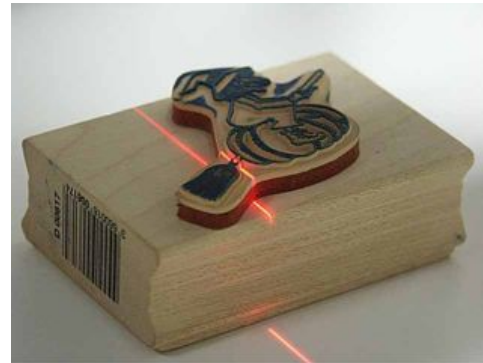
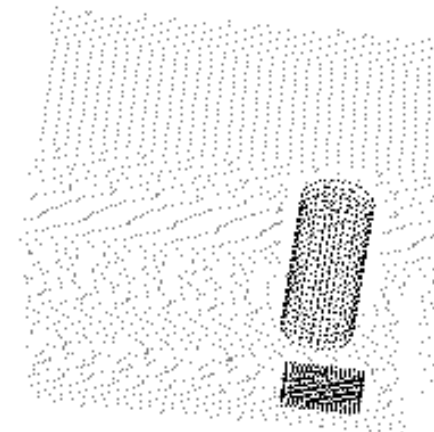
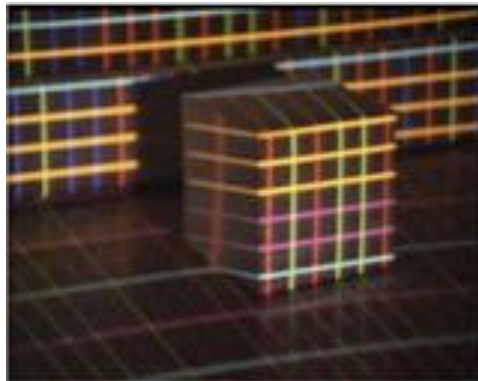


Figure 4 : Zoom sur l'objet



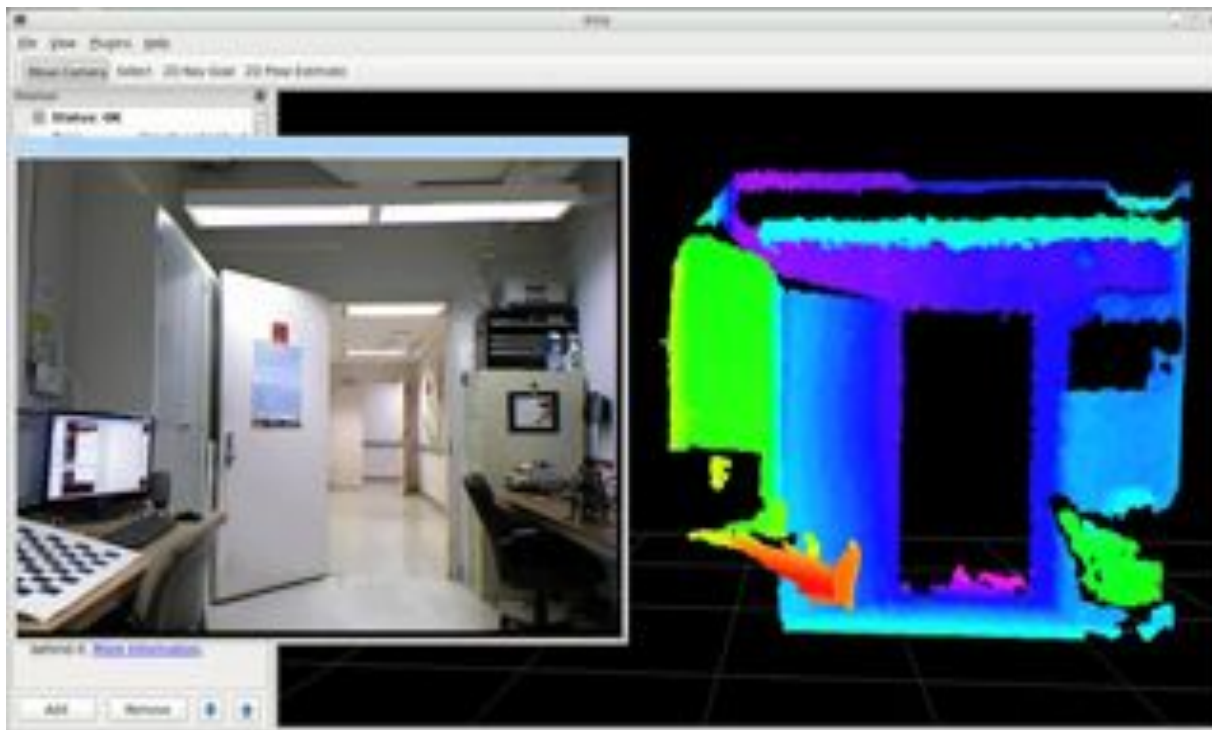
Figure 6 : Nuage de points



The precision depends on the baseline

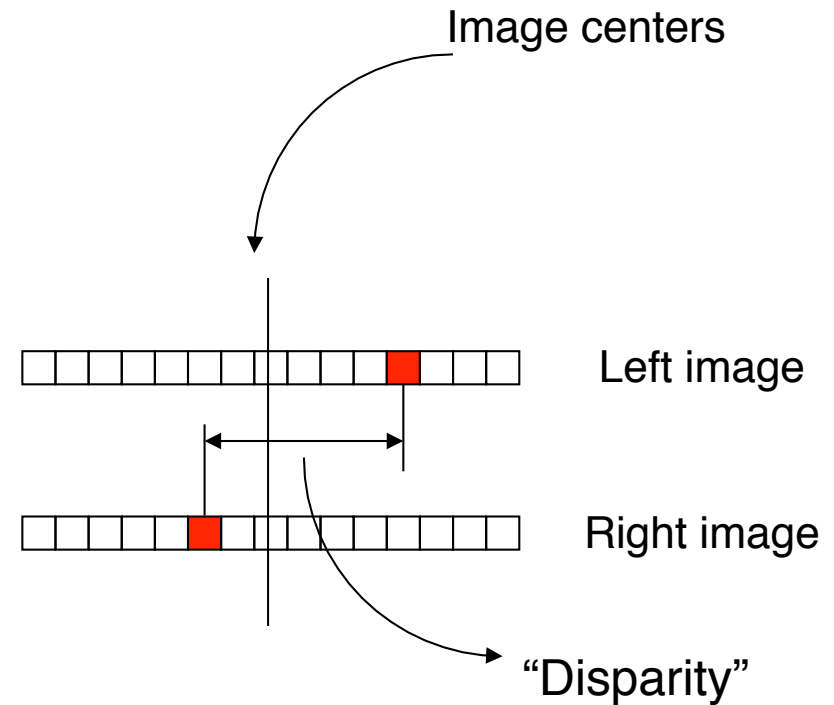
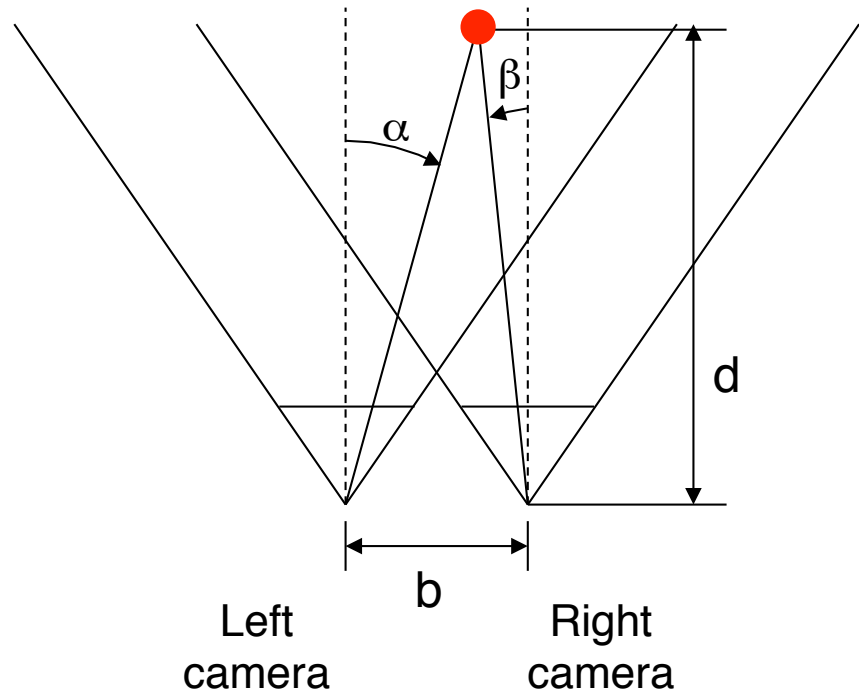
3D data from vision

- Autumn 2010 : Kinect



3D data from vision

- Stereovision = triangulation

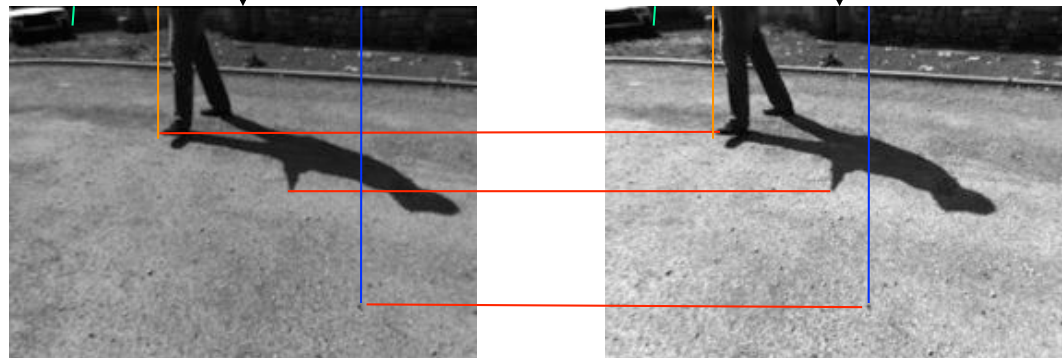


2 angles, 1 distance :
$$d = \frac{b}{\tan(\alpha) + \tan(\beta)}$$

Dense stereovision



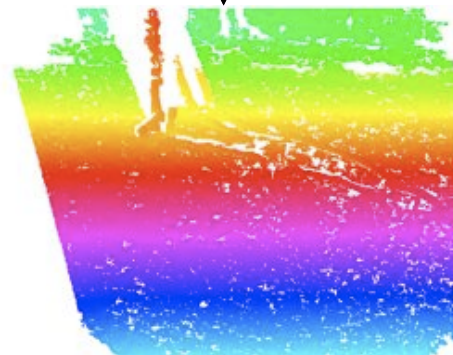
0. Image pair acquisition



Calibration data

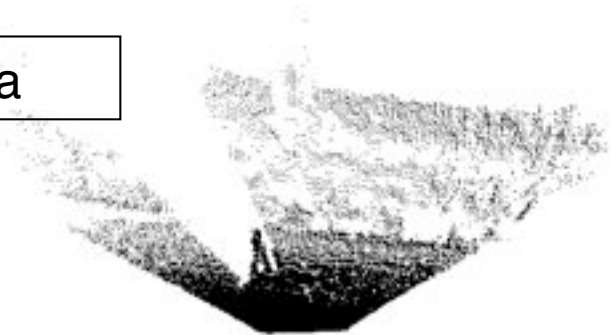
1. Image rectification

2. Pixel matching process



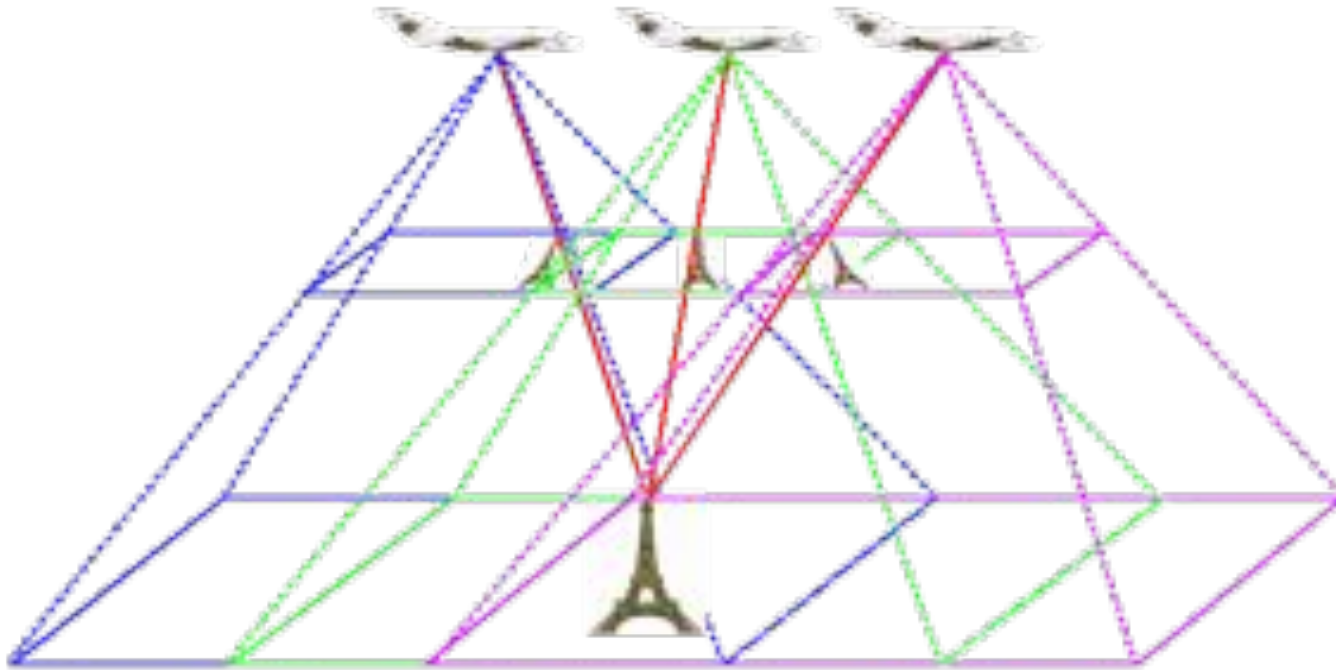
Calibration data

3. 3D reconstruction



Single camera stereovision (« Structure from Motion »)

- One moving camera



Additional difficulty: estimation of the motion parameters

(« structure and motion from motion », « SLAM »)

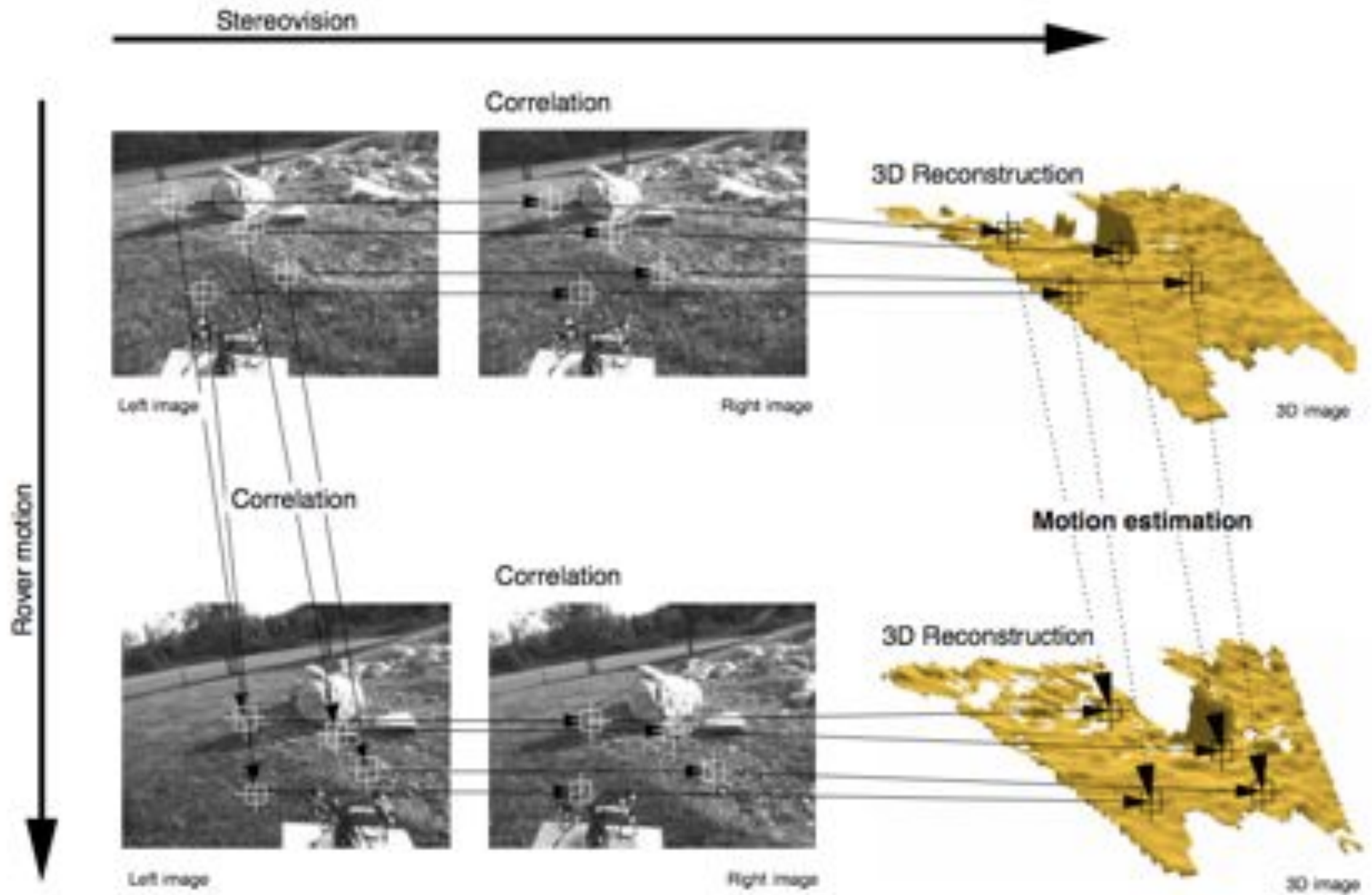
Localization solutions in robotics

- Odometry
- Similarly, motion / accelerations sensors (inertial navigation)

Inherently drifts over time and distances, subject to slippages and skids

➔ Develop solutions relying on the robot exteroceptive sensors

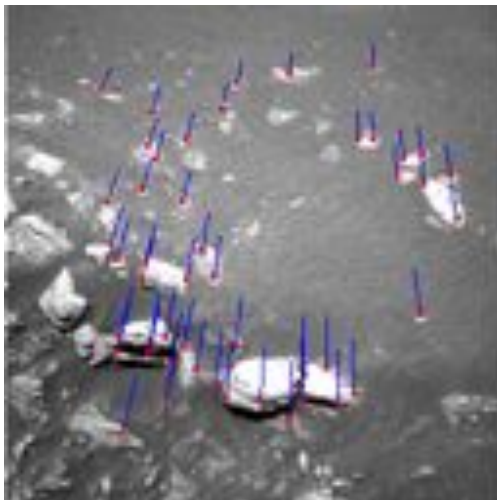
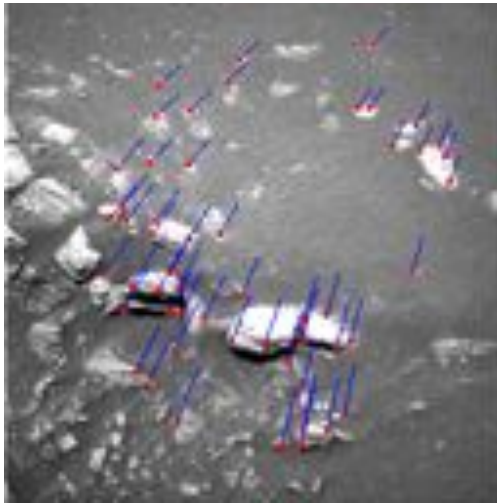
Visual odometry: principle



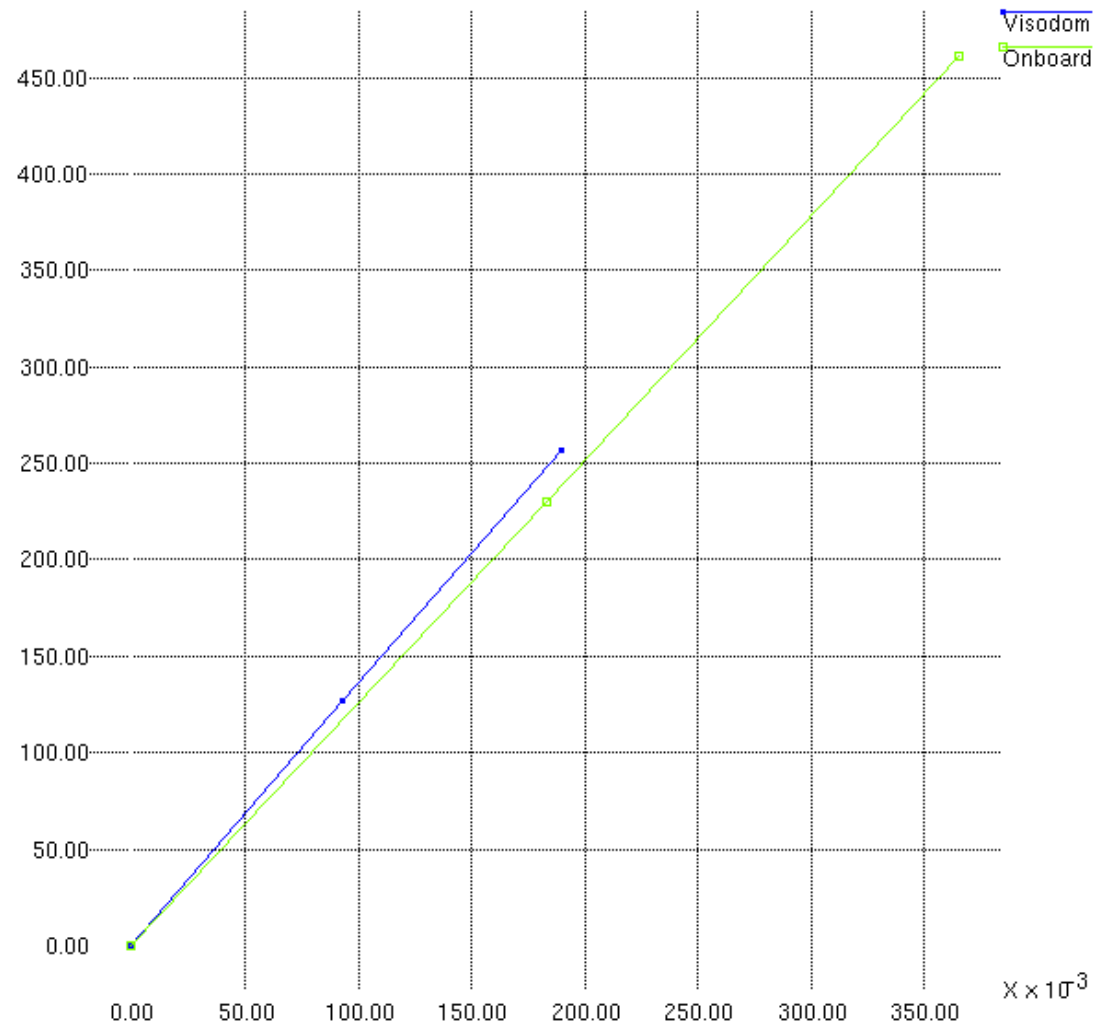
(speed X2)

Visual odometry: results

- Applied on the Mars Exploration Rovers



X Graph



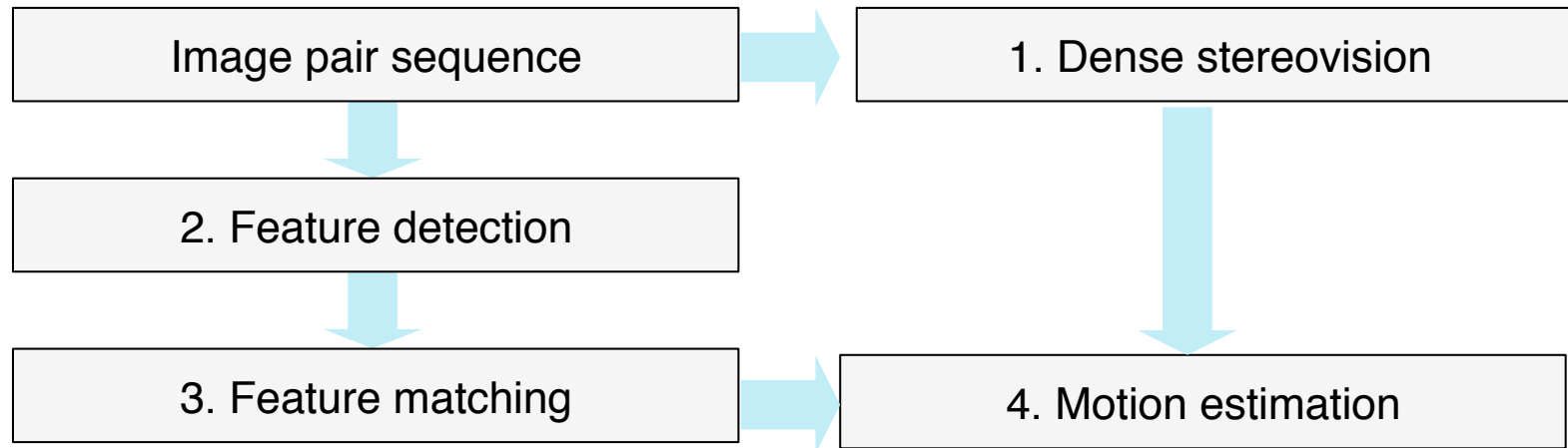
50 % slip

Visual odometry: benefits

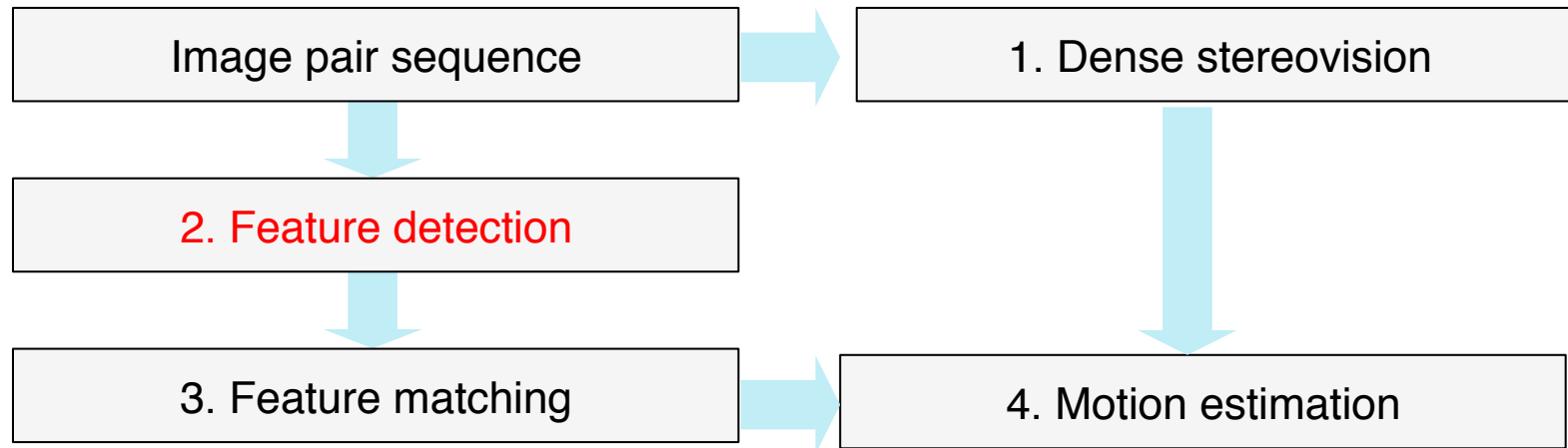
- Contrary to wheel odometry, VO is not affected by wheel slip in uneven terrain or other adverse conditions.
- More accurate trajectory estimates compared to wheel odometry (relative position error 0.1% – 2%)
- In GPS-denied environments (*e.g.* underwater, planetary or indoor), VO has utmost importance



Visual odometry: implied functions

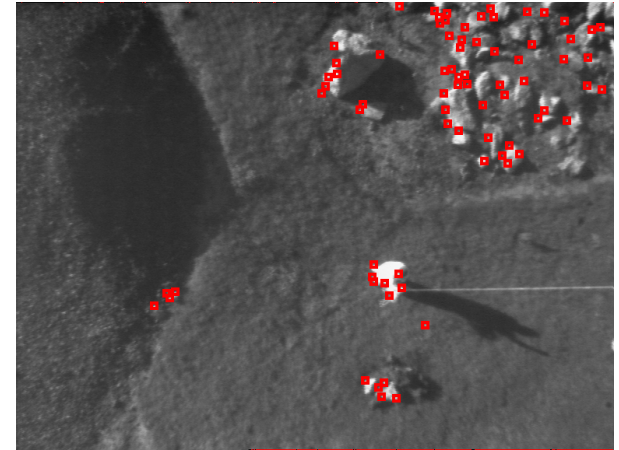


Visual odometry: implied functions



Interest points

Harris interest points: sharp peaks of the autocorrelation function



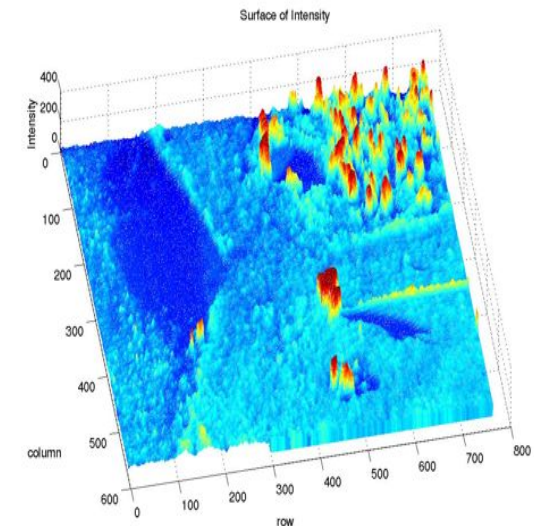
Auto-correlation matrix:

$$G(\mathbf{x}, \bar{\sigma}) \otimes \begin{pmatrix} I_u(\mathbf{x}, s\sigma)^2 & I_u(\mathbf{x}, s\sigma)I_v(\mathbf{x}, s\sigma) \\ I_u(\mathbf{x}, s\sigma)I_v(\mathbf{x}, s\sigma) & I_v(\mathbf{x}, s\sigma)^2 \end{pmatrix}$$

$$I_u(\mathbf{x}, s\sigma) = s G_u(\mathbf{x}, s\sigma) * I(\mathbf{x}),$$

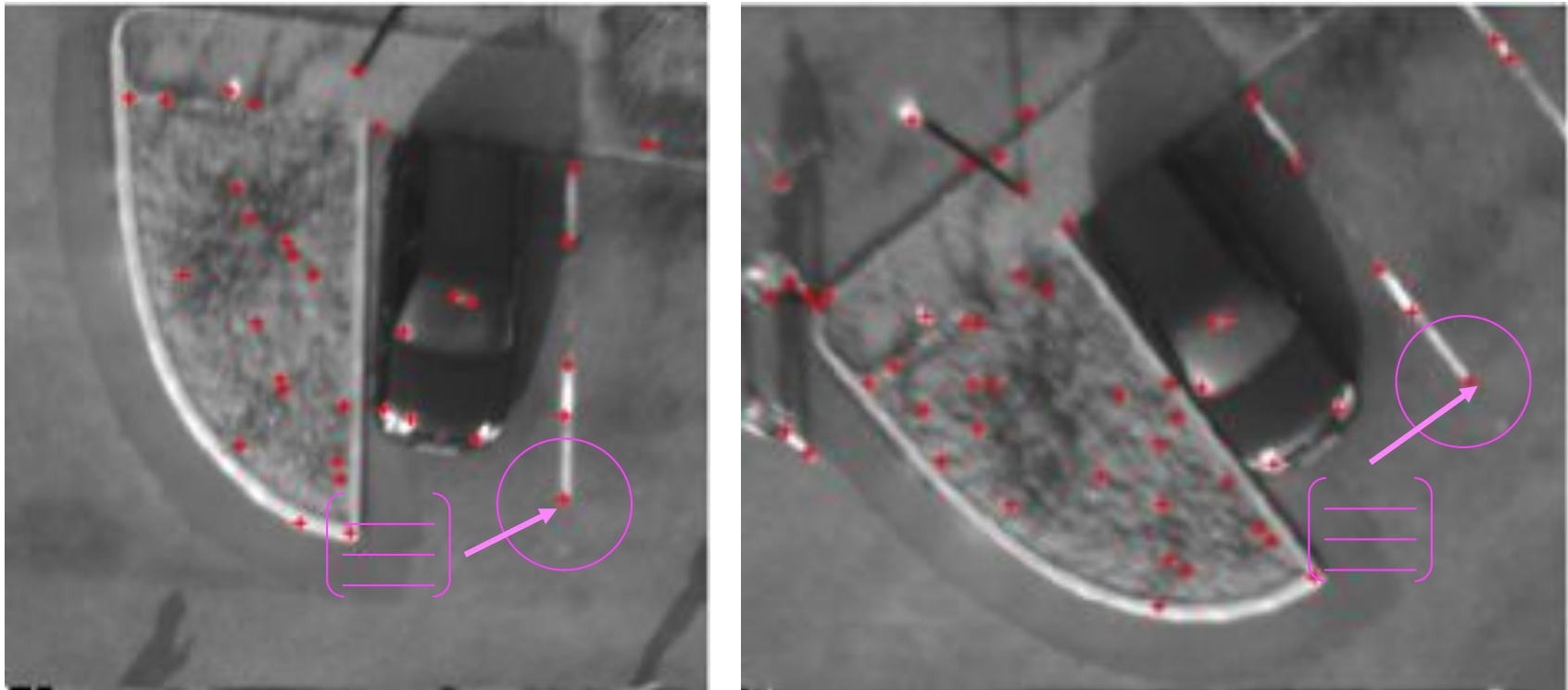
$$I_v(\mathbf{x}, s\sigma) = s G_v(\mathbf{x}, s\sigma) * I(\mathbf{x})$$

Principal curvatures defined by the two eigen values of the matrix λ_1, λ_2
(s: scale of the detection)



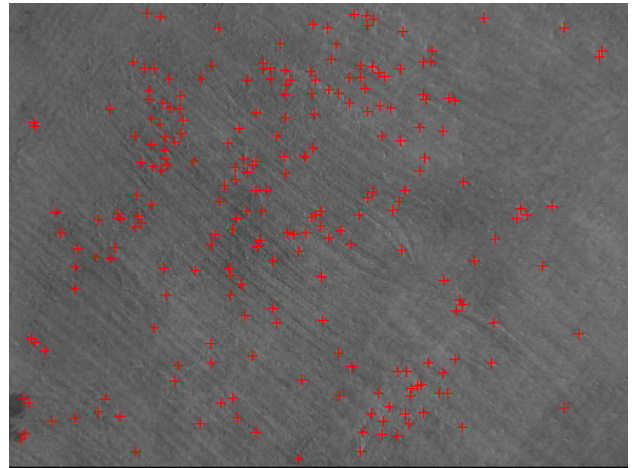
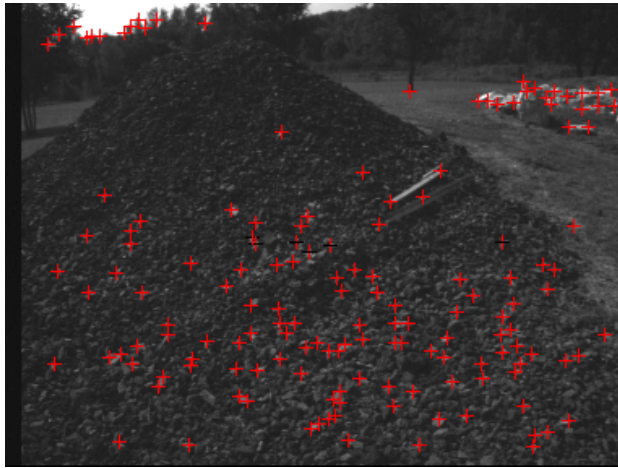
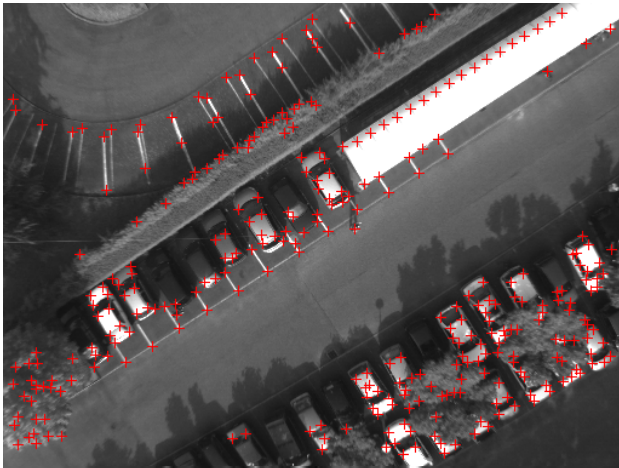
Interest points

- Local gray value invariants [Harris 88][Schmid 97] : local description vector invariant to rotation



Find matches with the similarity measure of *descriptors*

Harris interest points

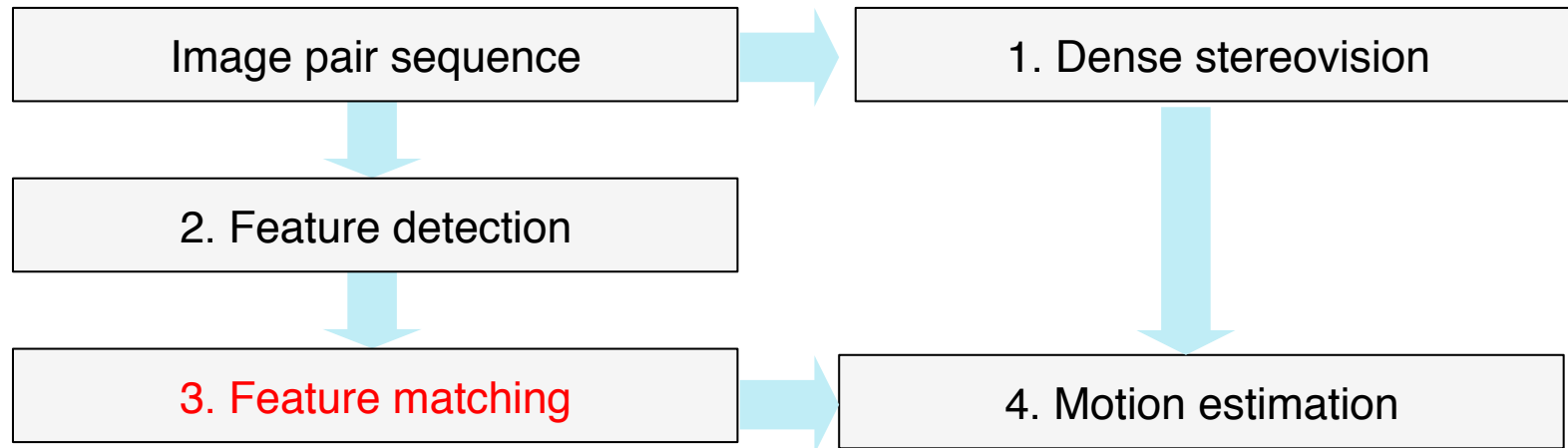


Interest points

- Numerous interest points definition available
 - Scale invariant interest points (“SIFT”, [Lowe99])
 - Speeded-up robust features (“SURF”, [Bay 2006])
 - Binary Robust Independent Elementary Features (“BRIEF”, [Calonder 2010])
 - ...

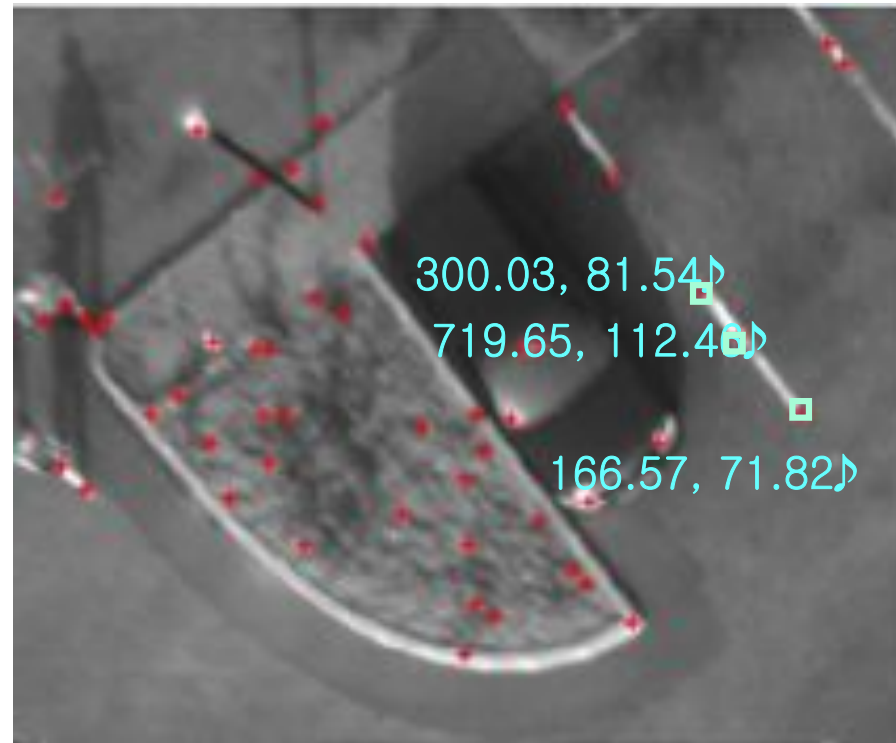
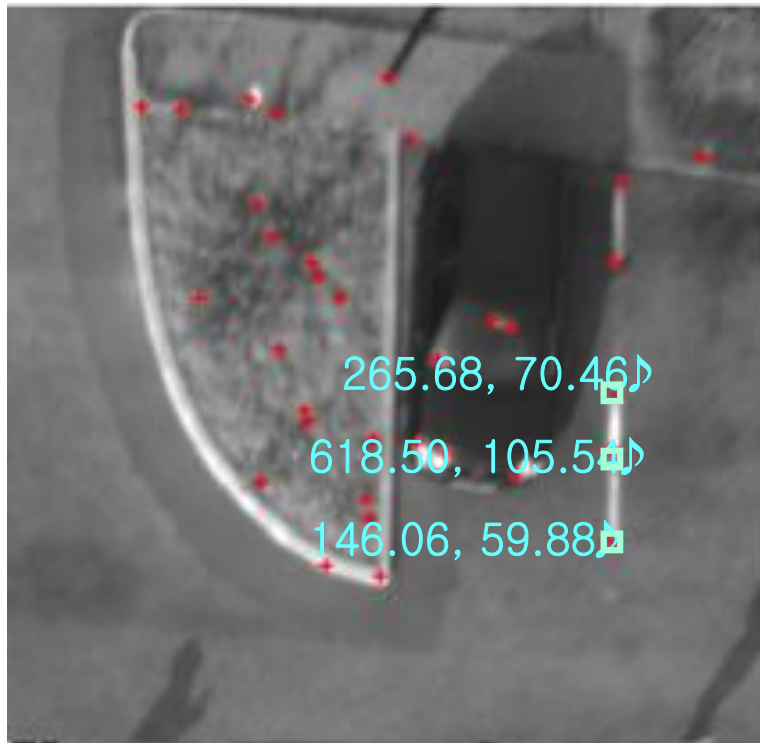
➔ The choice depends on the considered problem context

Visual odometry: implied functions



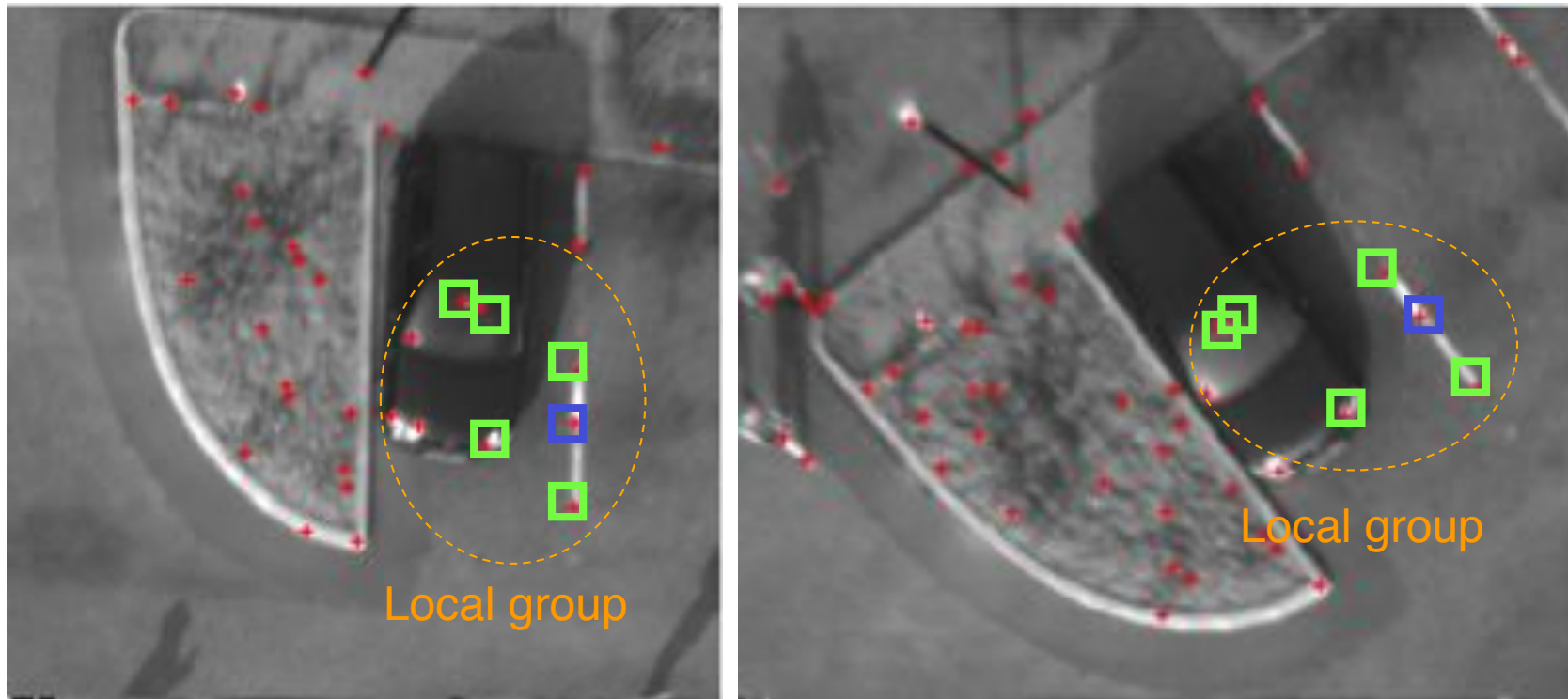
Interest points matching : principle

- If a generated match is correct,
 - Similar principal curvatures
 - *Applied to matching candidate selection*



Interest points matching : principle

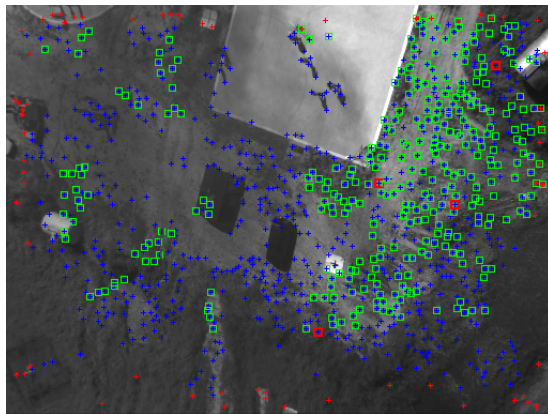
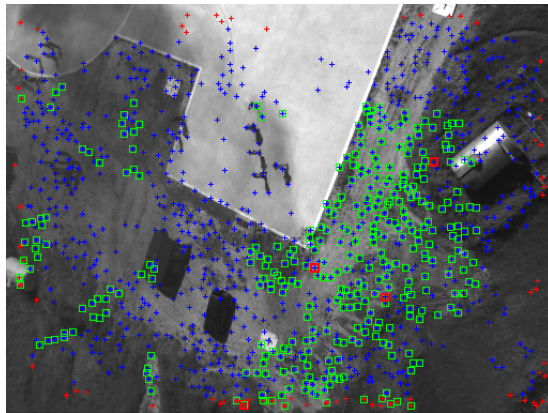
- If a generated match is correct,
 - In the vicinity of the match, neighbor matches must exist
 - *Local group matching : consideration a local region of image*



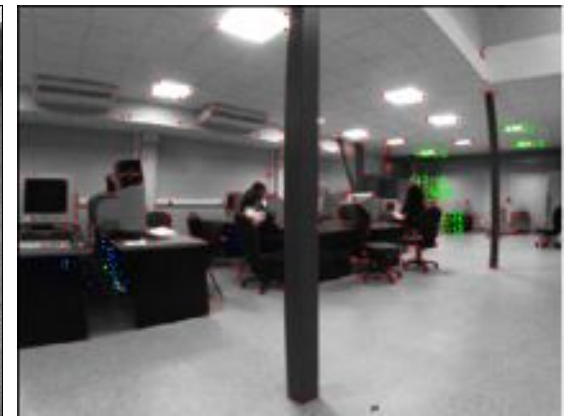
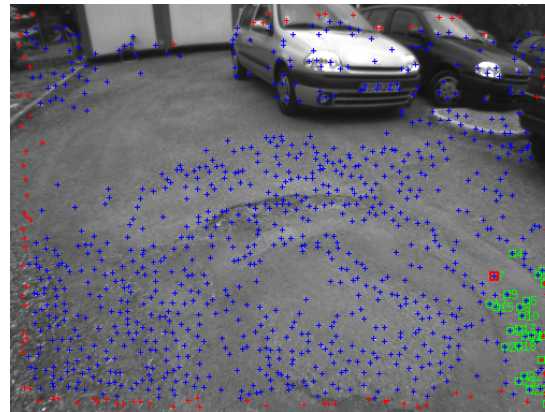
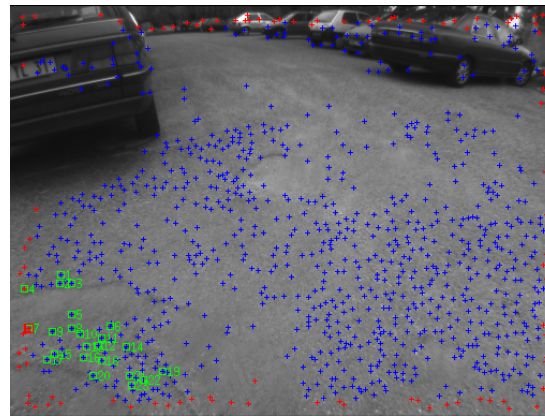
□: pivot □: members

Some matching results♪

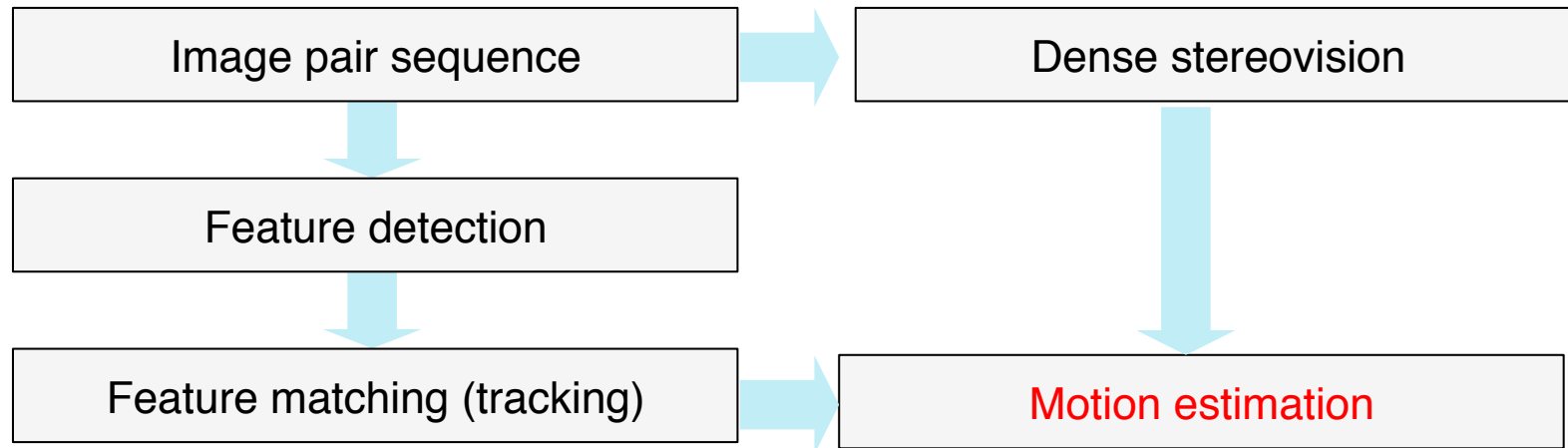
Consecutive frames



Small overlap

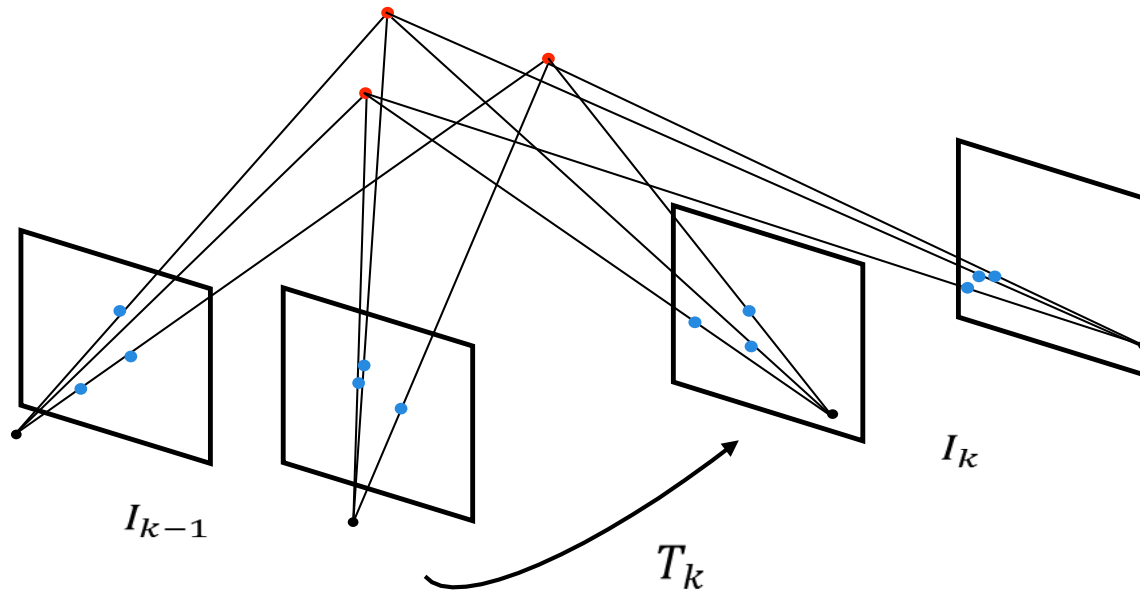


Visual odometry: implied functions



Motion estimation: problem statement

- Given two set of matched 3D points



- Find the 3D transformation T_k that minimizes the point distances

$$T_k = \begin{bmatrix} R_{k,k-1} & t_{k,k-1} \\ 0 & 1 \end{bmatrix} = \arg \min_{T_k} \sum_i \|\tilde{X}_k^i - T_k \tilde{X}_{k-1}^i\|$$

Motion estimation: least square minimization

- Given two set of matched 3D points $\{p_i\}, \{p'_i\}$
- The coordinates of matching points are linked by:

$$p'_i = R.p_i + T + N_i$$

Where N_i denotes some noise

- Least square estimation: find the 3D transform (R,T) that minimizes:

$$\Sigma^2 = \sum_{i=1}^N \|p'_i - (R.p_i + T)\|^2$$

Motion estimation: least square minimization

- Assuming a zero-mean noise on the points coordinates, the barycenter of the transformed first set and the second set should be equal:

$$p' = p''$$

where
$$p = \frac{1}{N} \cdot \sum_{i=1}^N p_i \quad ; \quad p' = \frac{1}{N} \cdot \sum_{i=1}^N p'_i \quad ; \quad p'' = \frac{1}{N} \cdot \sum_{i=1}^N \hat{R} \cdot p_i + \hat{T}$$

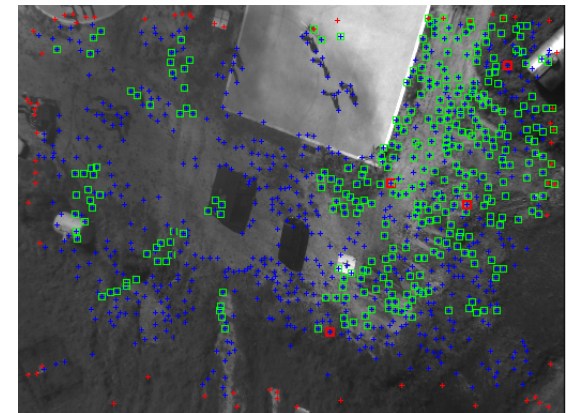
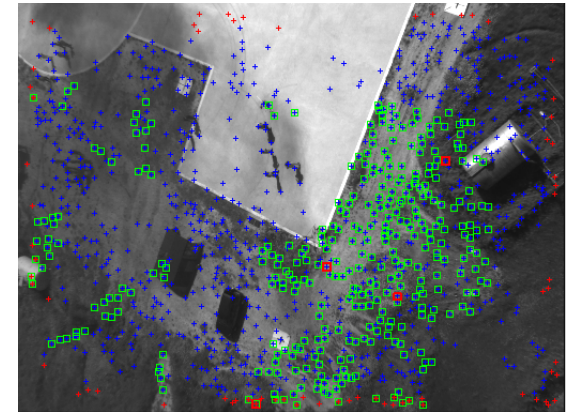
- Changing the coordinates: $q_i = p_i - p \quad ; \quad q'_i = p'_i - p'$

then we have
$$\Sigma^2 = \sum_{i=1}^N \|q'_i - R \cdot q_i\|^2$$

- Find R that minimizes Σ^2 (least square minimization)
- Estimate T as
$$\hat{T} = p' - \hat{R} \cdot p$$

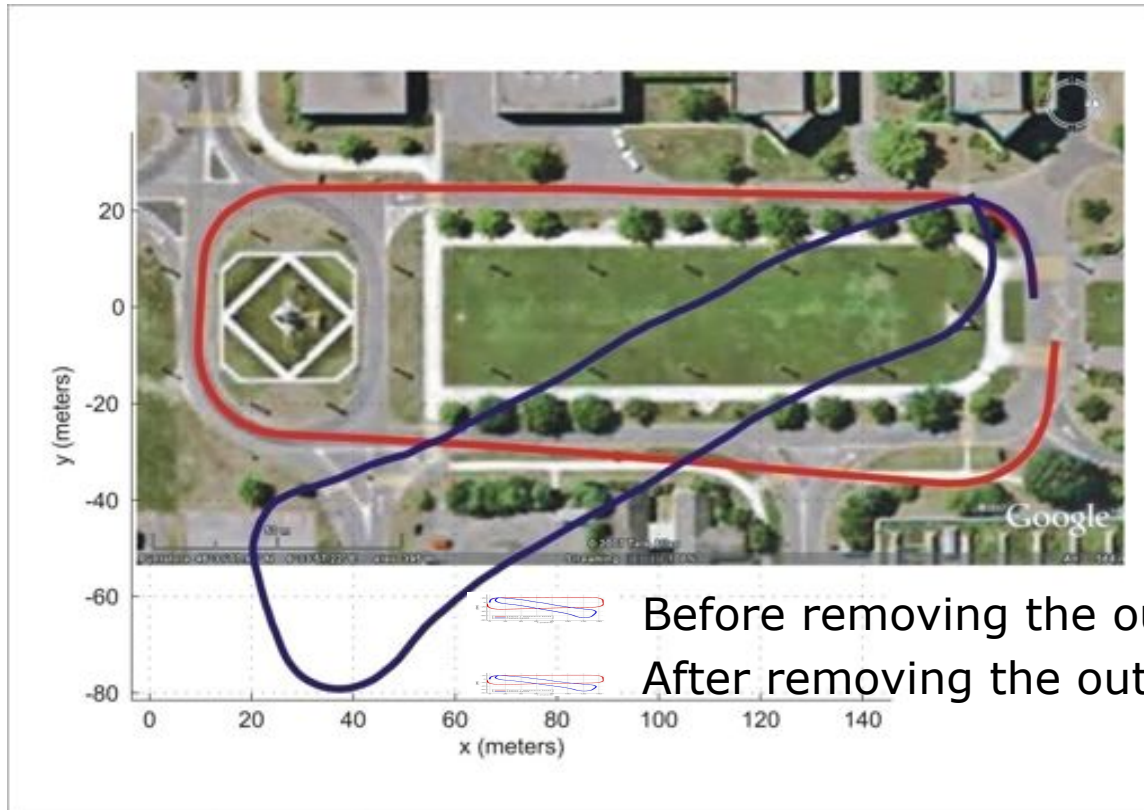
Robust estimation

- Matched points are usually contaminated by wrong data associations (= “outliers”)
- Possible causes of outliers are
 - image noise
 - occlusions
 - blur,
 - changes in view point and illumination for which the feature detector or descriptor does not account for
- For the camera motion to be estimated accurately, outliers *must* be removed
- This is the task of *robust estimation*



- “RANSAC” (“Random Sample Consensus”) [Fishler & Bolles, 1981] has been established as the standard method for motion estimation in the presence of outliers

Results / Robust estimation



- › Error at the loop closure: 6.5 m
- › Error in orientation: 5 deg
- › Trajectory length: 400 m

Before removing the outliers
After removing the outliers

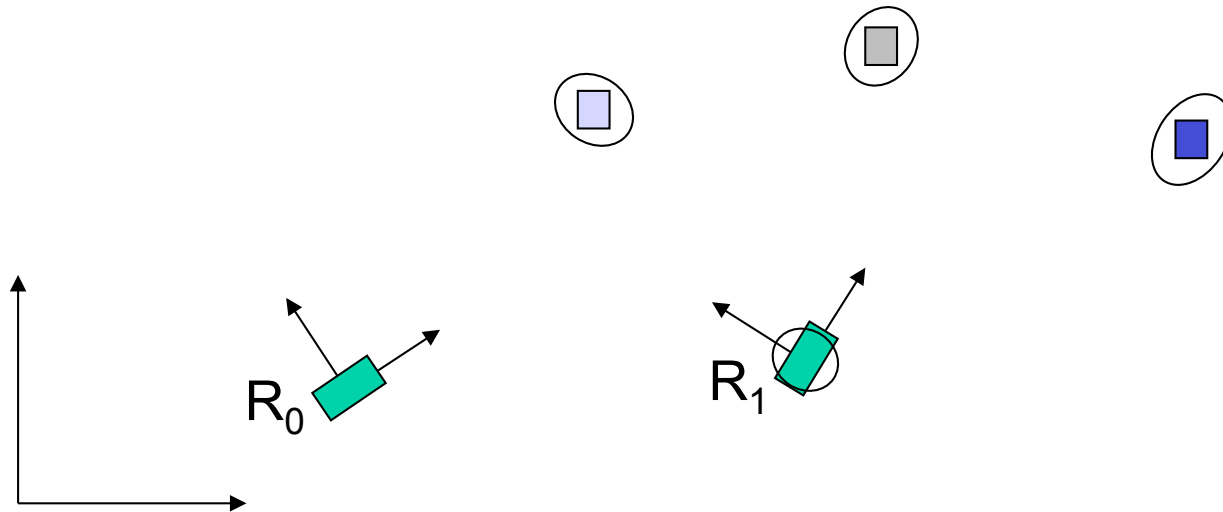
Localization solutions in robotics

- Odometry
- Similarly, motion / accelerations sensors (inertial navigation)
 - Inherently drifts over time and distances, subject to slippages and skids
 - ➔ Develop solutions relying on the robot exteroceptive sensors
- Visual odometry: akin to dead reckoning
 - Inherently drifts over time and distances
 - ➔ Develop solutions relying on the robot exteroceptive sensors that memorizes stable environment features (SLAM)

SLAM

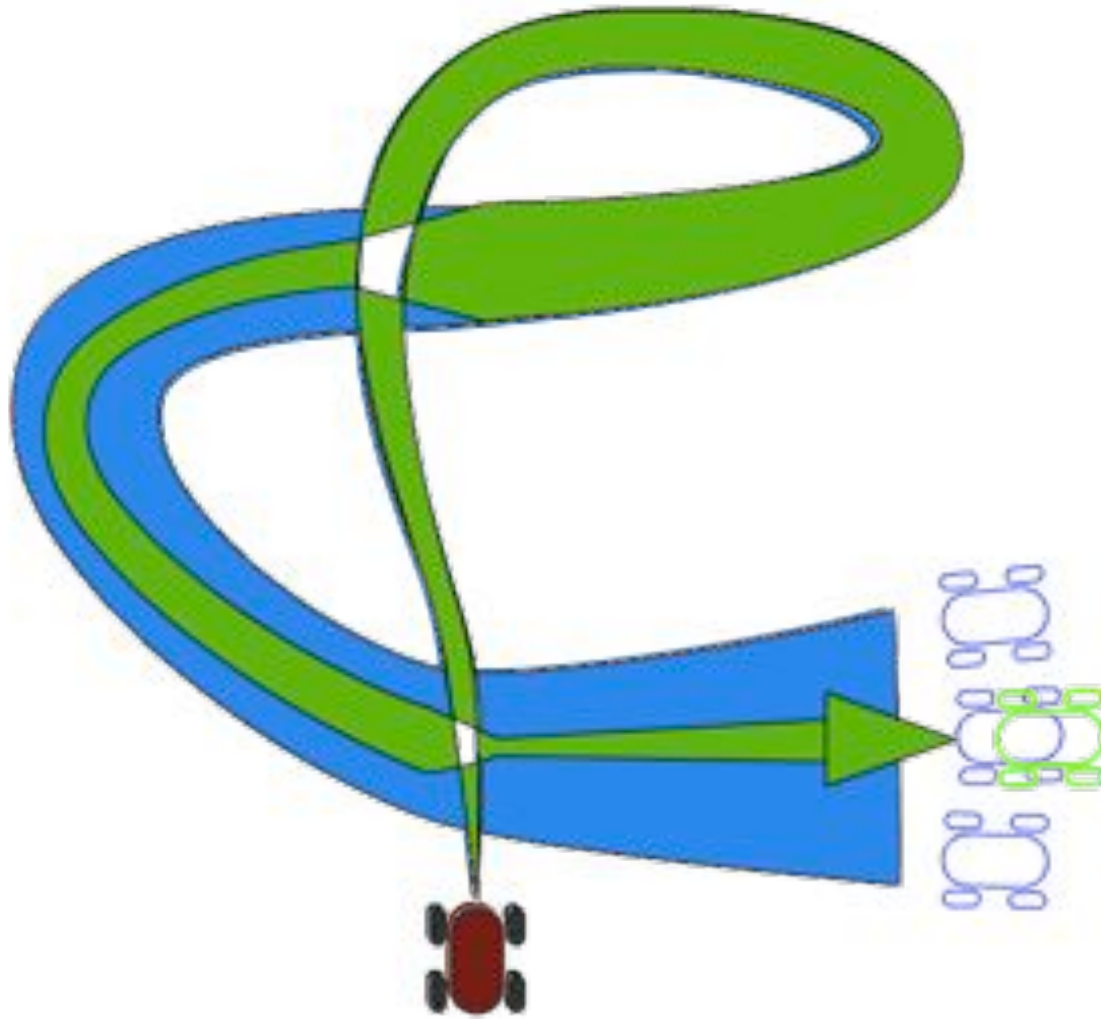
Simultaneous Localisation And Mapping

Principle of SLAM



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

Principle of SLAM



(parenthesis: 2D Lidars)



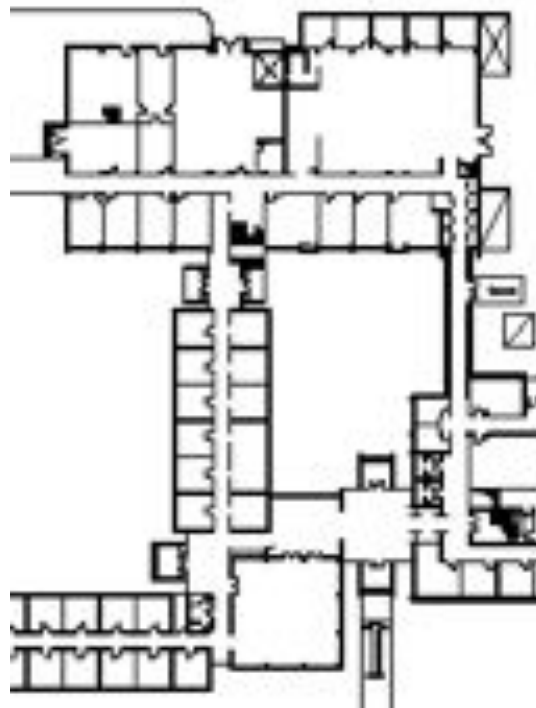
- Telemetry (« measuring distances »)

Laser («Light Amplification by Stimulated Emission of Radiation »)

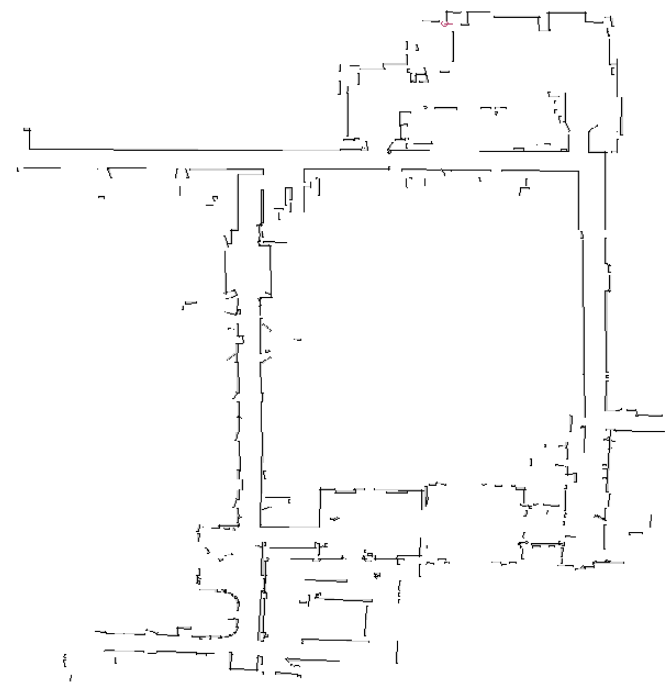
Time of flight of phase shift measures

→ LIDAR (« Light Detection And Ranging »)

In the plane:



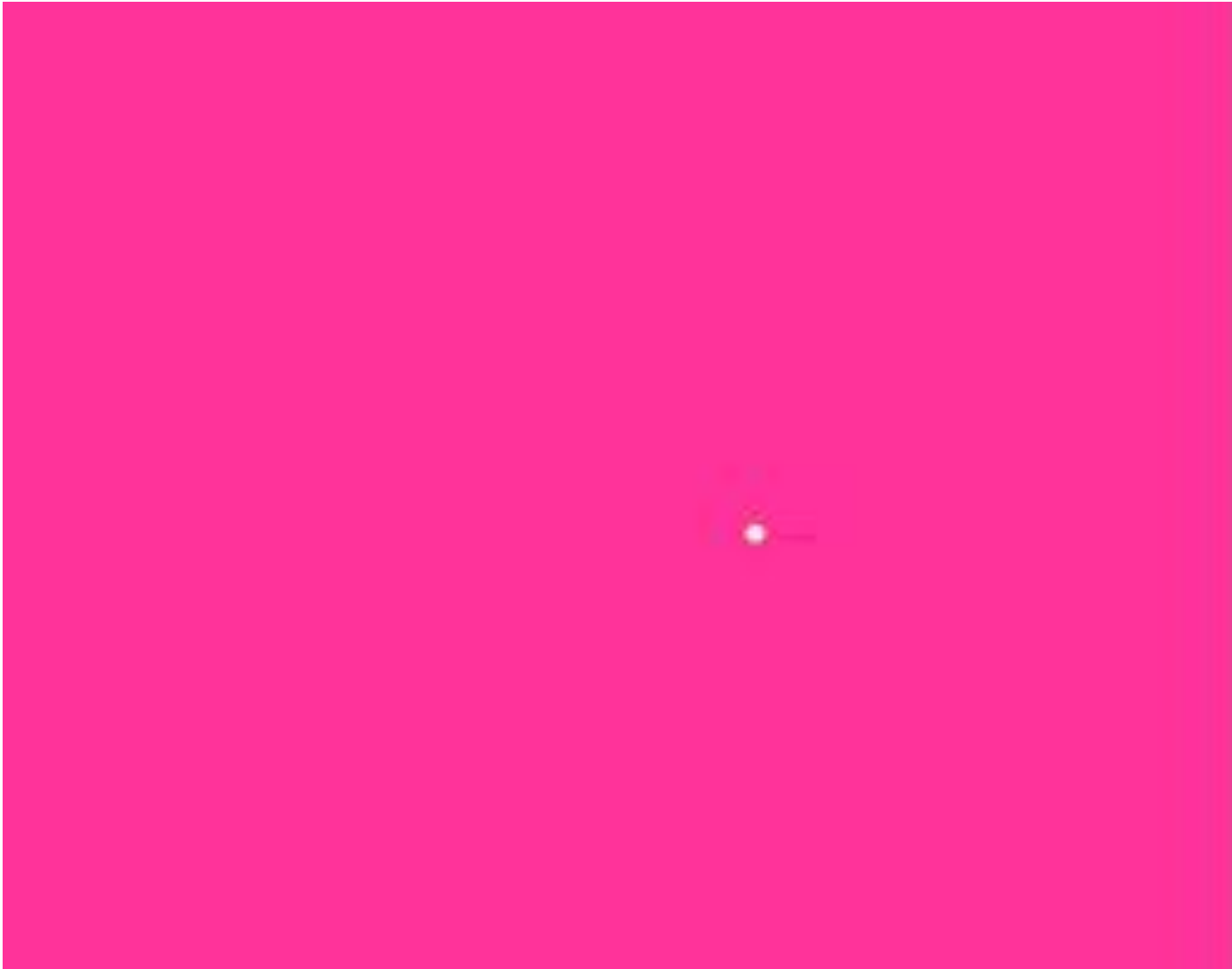
Building map



Perceived map

SLAM : illustration

Occupancy grid: on a flat plane, with odometry and “2D” LRF (S. Thrun)



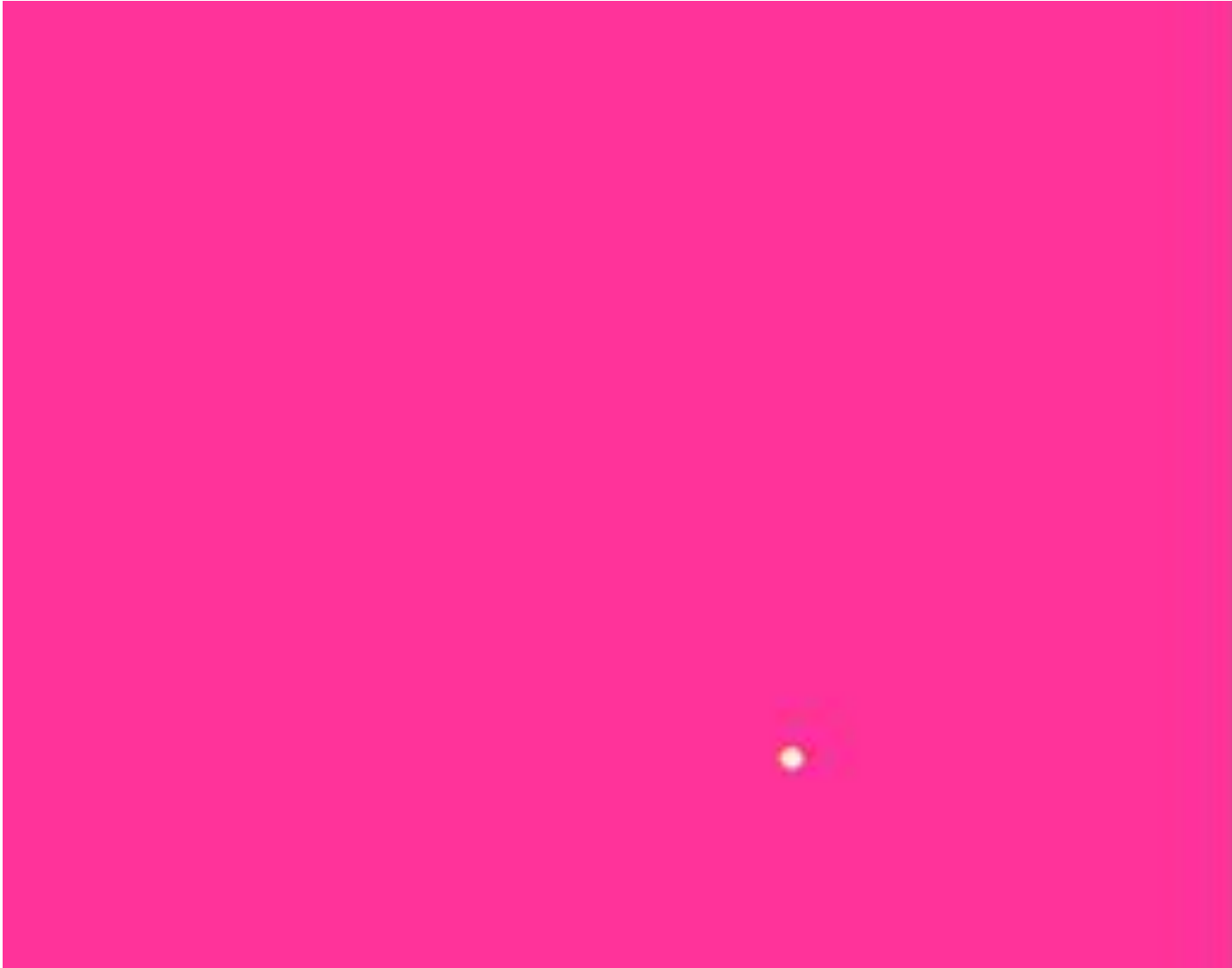
SLAM : illustration

Occupancy grid: on a flat plane, with odometry and “2D” LRF (S. Thrun)



SLAM : illustration

Occupancy grid: on a flat plane, with odometry, a “2D” LRF and SLAM (S. Thrun)



SLAM : illustration

Occupancy grid: on a flat plane, with odometry, a “2D” LRF and SLAM (S. Thrun)



SLAM must reads

There are tons of papers on SLAM...

At least, read those:

Simultaneous Localisation and Mapping (SLAM):
Part I The Essential Algorithms

Hugh Durrant-Whyte, *Fellow, IEEE*, and Tim Bailey

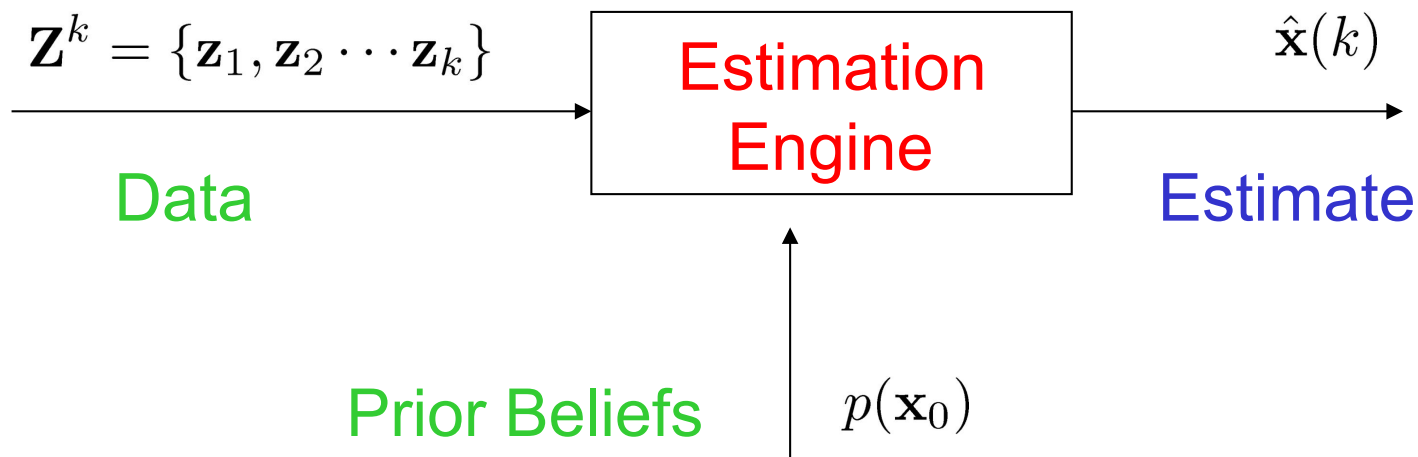
Simultaneous Localisation and Mapping (SLAM):
Part II State of the Art

Tim Bailey and Hugh Durrant-Whyte

(in *Robotics and Automation Magazine*,
Vol 13, Num 2/3, 2006)

Basics on estimation

“Estimation is the process by which we infer the value of a quantity of interest, \mathbf{x} , by processing data that is in some way dependent on \mathbf{x} .”



Basics on estimation

Maximum A Posteriori estimation:

Sensor model $p(z|x)$: “Given x , the probability of the sensor measurement being within 1m is....”

$$\underbrace{p(\mathbf{x}|\mathbf{z})}_{\text{posterior}} = \frac{p(\mathbf{z}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{z})}$$
$$\propto \underbrace{p(\mathbf{z}|\mathbf{x})}_{\text{Likelihood}} \times \underbrace{p(\mathbf{x})}_{\text{prior}}$$

Given an observation \mathbf{z} , a likelihood function $p(\mathbf{z}|\mathbf{x})$ and a prior distribution on \mathbf{x} , $p(\mathbf{x})$, the **maximum a posteriori estimator - MAP** finds the value of \mathbf{x} which maximises the posterior distribution $p(\mathbf{x}|\mathbf{z})$

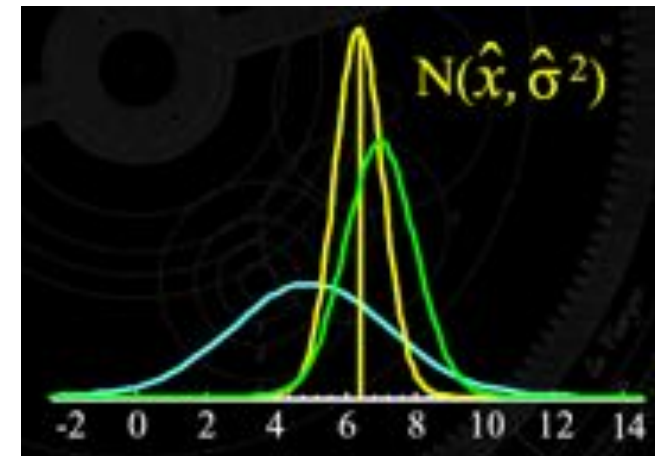
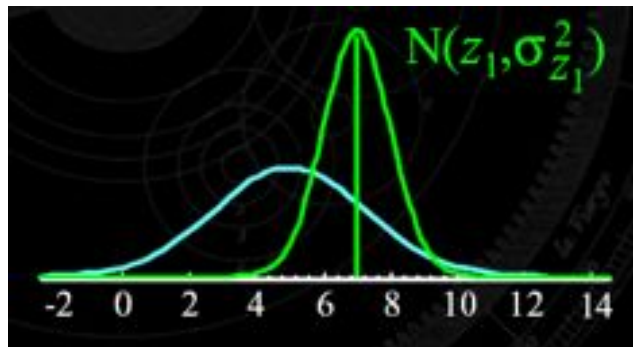
$$\hat{\mathbf{x}}_{map} = \arg \max_{\mathbf{x}} p(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \quad (1)$$

Basics on estimation

Example of MAP estimation

With normal prior and likelihood

$$z_2, \sigma_{z_2}^2$$
$$\hat{x}_2 = \dots?$$
$$\hat{\sigma}_2^2 = \dots?$$



$$\hat{x}_2 = \hat{x}_1 + K_2 (z_2 - \hat{x}_1)$$
$$K_2 = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_{z_2}^2}$$

$$\frac{1}{\sigma_2^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_{z_2}^2}$$

Basics on estimation

Recursive Bayesian estimation

$\mathbf{Z}^k = \{\mathbf{z}_1, \mathbf{z}_2 \cdots \mathbf{z}_k\}$ Sequence of data (measurements)

We want conditional mean (mmse) of x given \mathbf{Z}^k

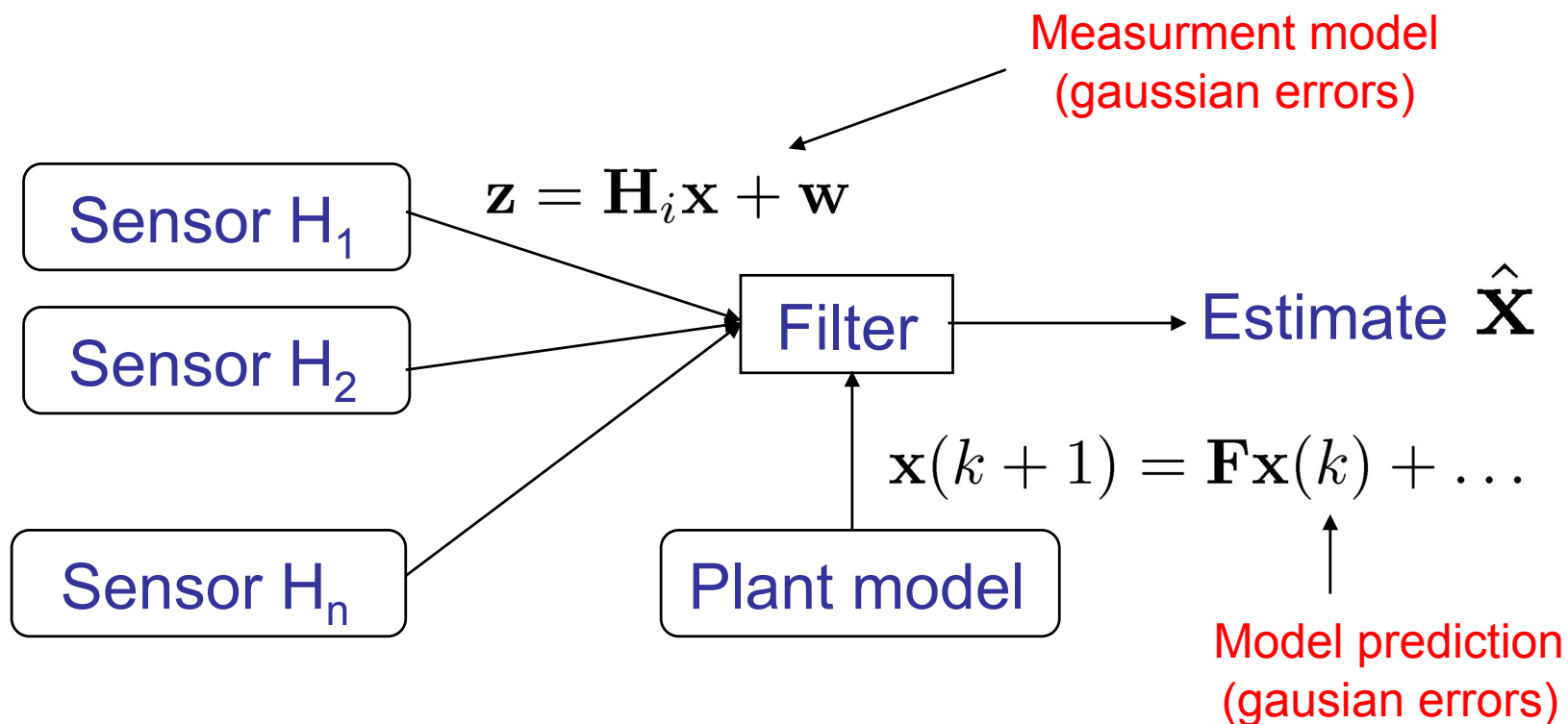
Can we iteratively calculate this – *i.e.* every time
a new measurement comes in, update our estimate?

Key idea: “one mans posterior is another’s prior”

Basics on estimation

Kalman filter

Recursive process: estimate the system state from uncertain observations (measures) and an uncertain system model



Actual underlying state: \mathbf{x}

Basics on estimation

Kalman filter: things to know

- A recursive process
- Asynchronous
- Prediction / update structure
- Prediction increases covariances
- Update decreases covariances
- Essential importance of correlations

Mapping and Localisation

- With perfect localisation:

Given the robot position \mathbf{x}_v , and a sensor model $p(\mathbf{Z}^k | \mathbf{M}, \mathbf{x}_v)$

compute the map \mathbf{M} : $p(\mathbf{M} | \mathbf{x}_v, \mathbf{Z}^k)$

→ Mapping problem « solved » (proper management of the sensors uncertainties)

- With a perfect known map

Given a feature map \mathbf{M} , and a sensor model $p(\mathbf{Z}^k | \mathbf{M}, \mathbf{x}_v)$

compute the robot position \mathbf{x}_v : $p(\mathbf{x}_v | \mathbf{M}, \mathbf{Z}^k)$

→ Localisation problem « solved » (proper matching of sensor data and the map)

Mapping and Localisation

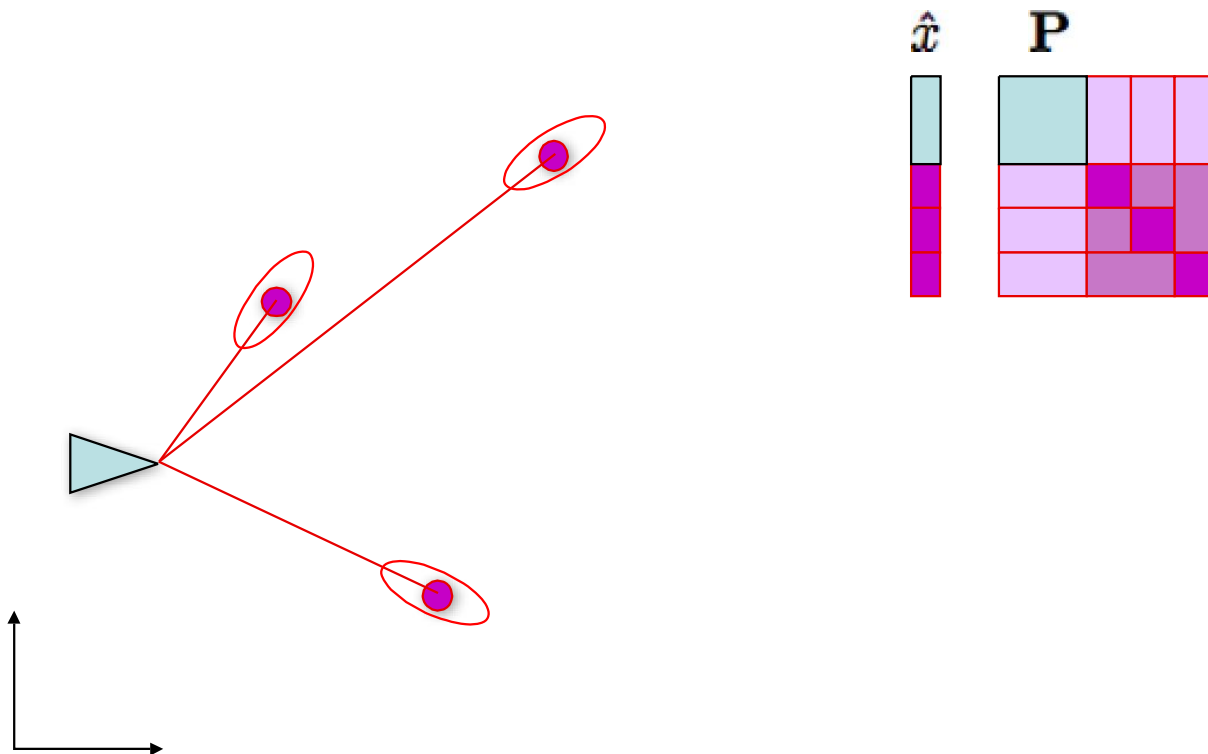
- Without perfect localisation, nor perfect map ?
⇒ Simultaneous localisation and mapping (SLAM)

Given the robot controls u_k and the sensor readings z_k , compute the map M and the robot position $x_v : p(M, x_v | u_k, z_k)$

SLAM: outline

- Simultaneous localisation and mapping
 - Problem presentation
 - Basics on estimation
 - EKF SLAM

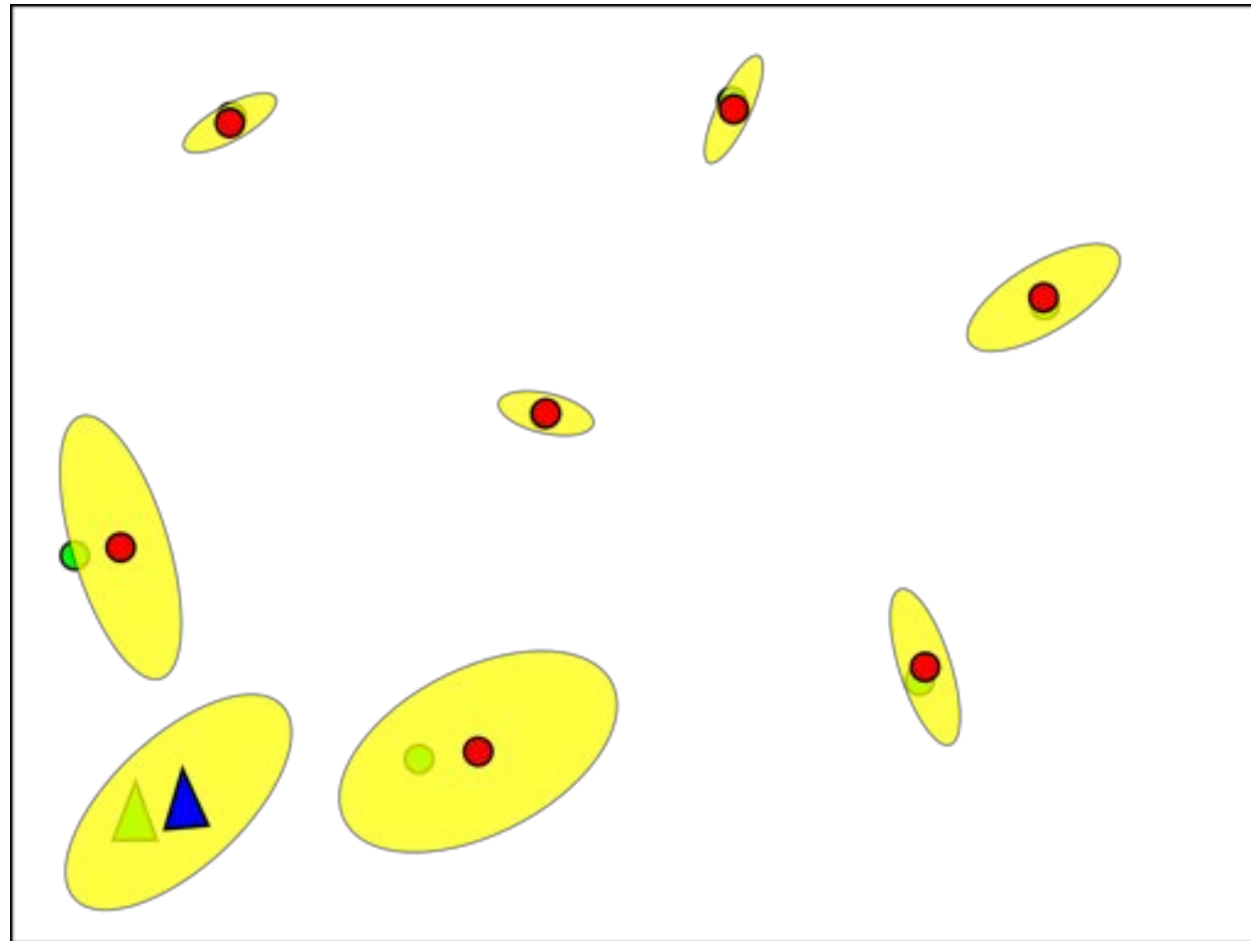
SLAM: EKF-based solution



SLAM: EKF-based solution

Importance of the correlations

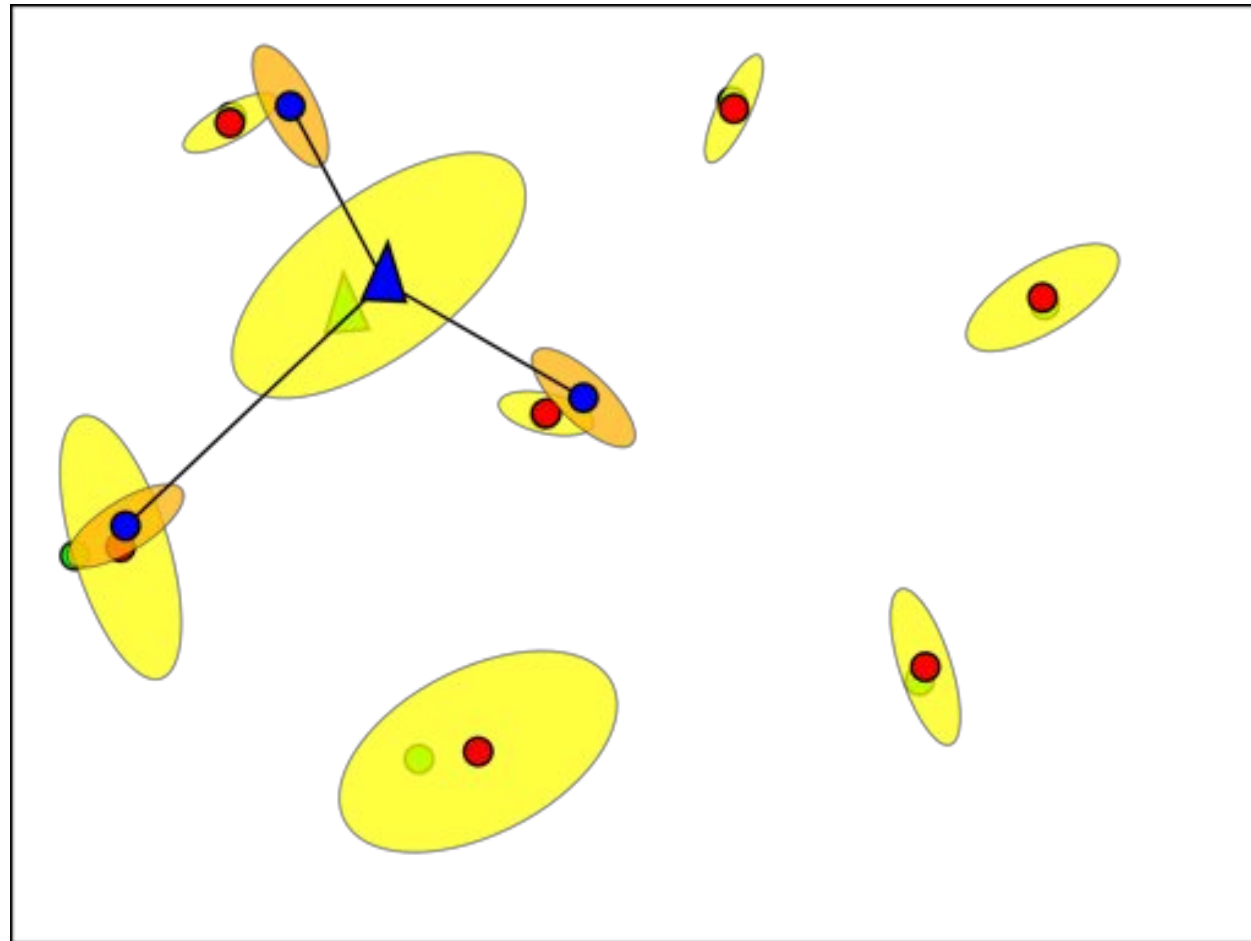
$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{mm} \end{bmatrix}$$



SLAM: EKF-based solution

Importance of the correlations

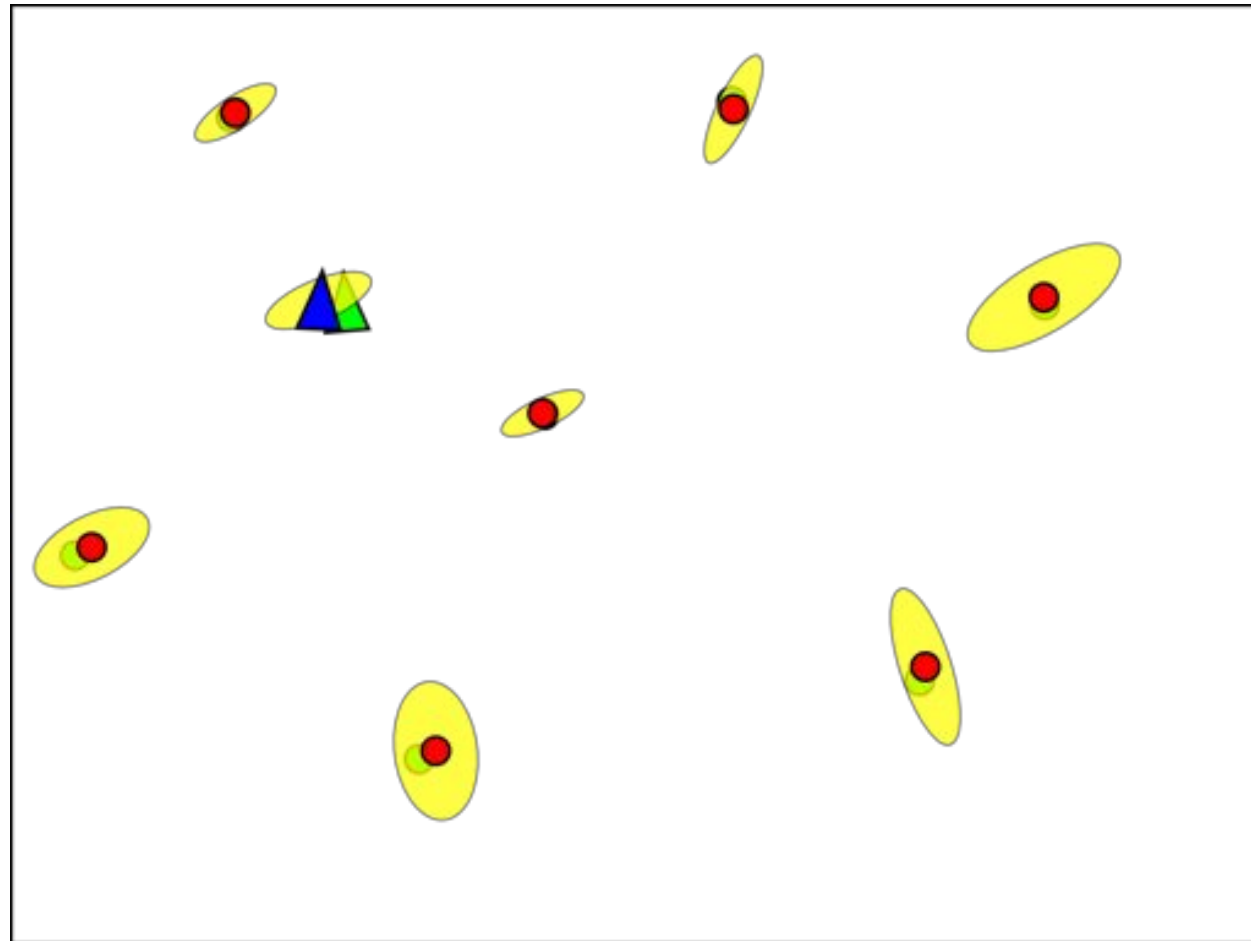
$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{vv} & \mathbf{P}_{vm} \\ \mathbf{P}_{vm}^T & \mathbf{P}_{mm} \end{bmatrix}$$



SLAM: EKF-based solution

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

Classic implementation
(e.g. vehicle position tracking)

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

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

Control input

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

noise

Prediction (static features) :

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:

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

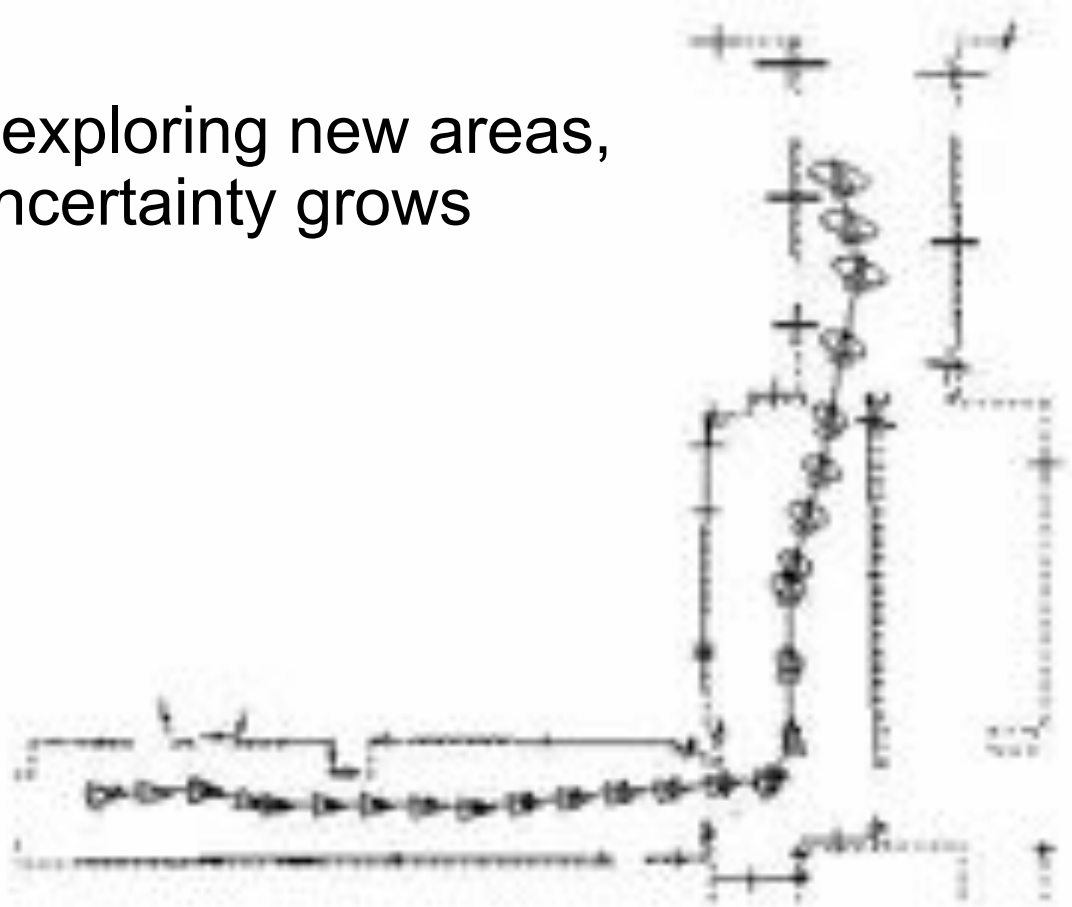
Motion estimation

noise

$$\begin{bmatrix} \mathbf{x}_v(k+1) \\ \mathbf{x}_{f,1}(k) \\ \vdots \\ \mathbf{x}_{f,n}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_v(k) \oplus \mathbf{u}(k) \\ \mathbf{x}_{f,1}(k) \\ \vdots \\ \mathbf{x}_{f,n}(k) \end{bmatrix}$$

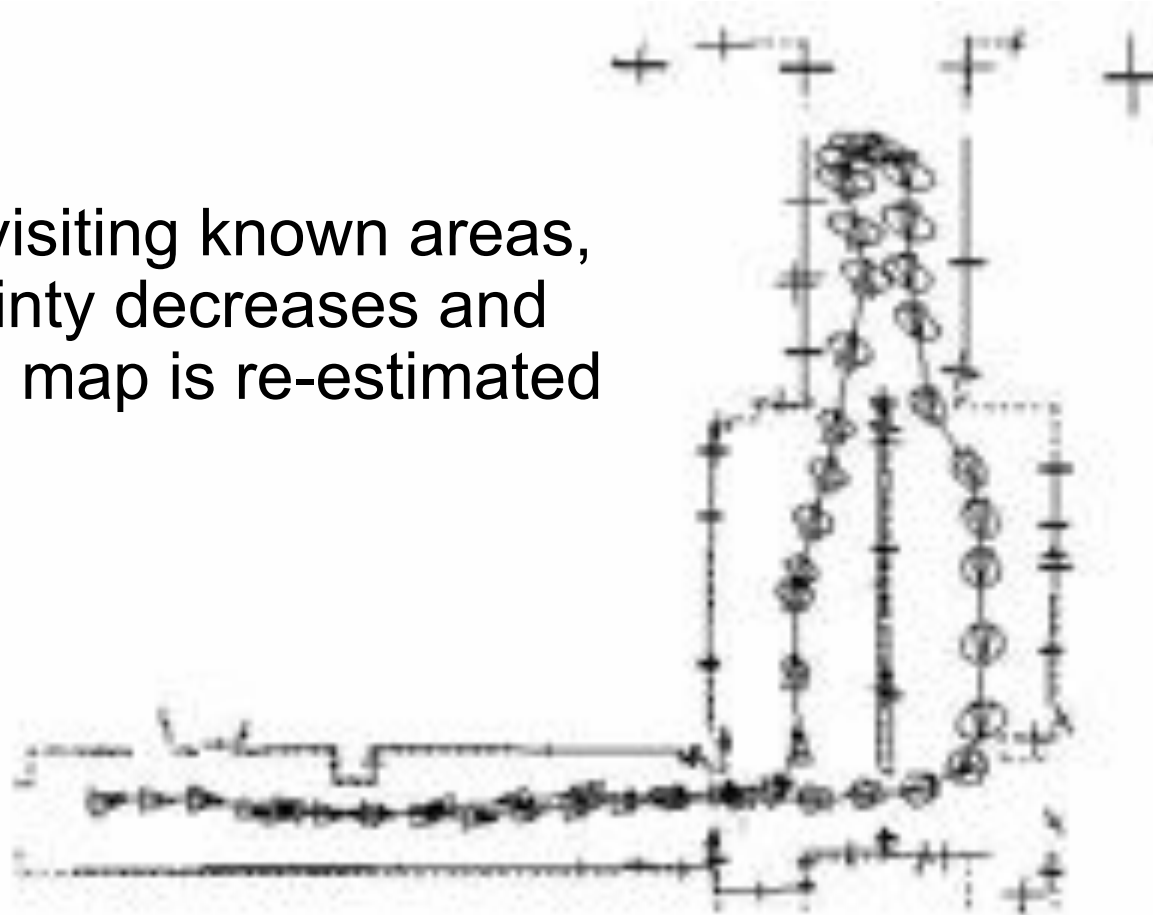
Kalman filter for SLAM

When exploring new areas,
uncertainty grows



Kalman filter for SLAM

When revisiting known areas,
uncertainty decreases and
the whole map is re-estimated



SLAM: outline

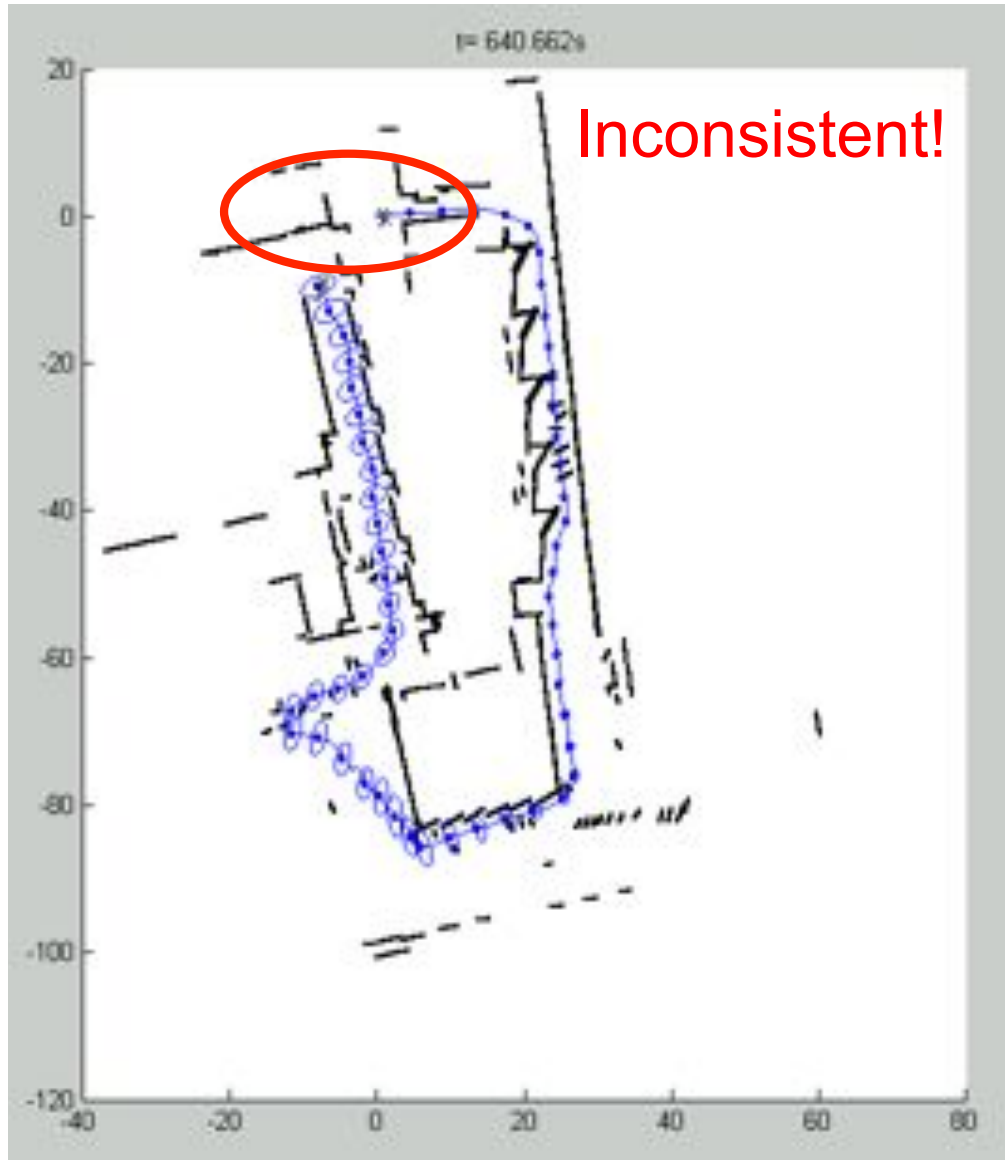
- Simultaneous localisation and mapping
 - Problem presentation
 - Basics on estimation
 - EKF SLAM
 - Main issues

EKF-SLAM Issues

1. Algorithmic complexity: $O(n^2)$

EKF-SLAM Issues

2. Non-linearities yield inconsistency



Inconsistent!

Standard data association cannot close a 250m loop

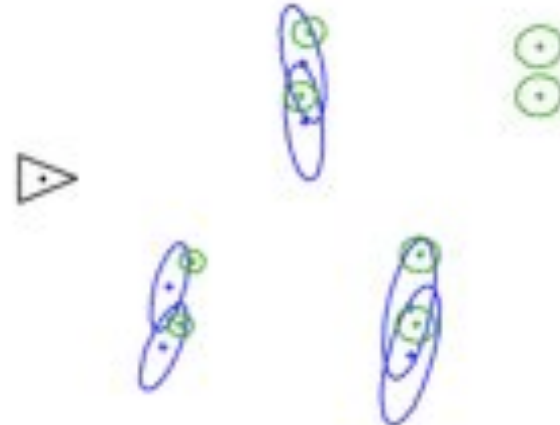
Computational complexity was not a problem here!

EKF-SLAM Issues

3. Data associations

Within the estimation framework:

Measurements
Predicted features



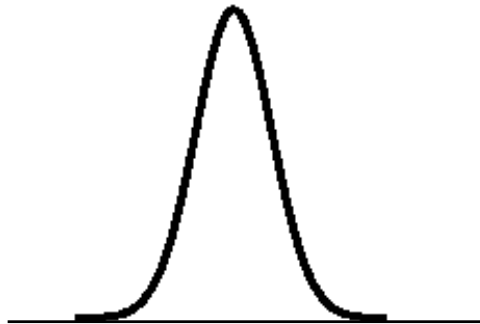
Within the estimation framework ?

Measurements
Predicted features

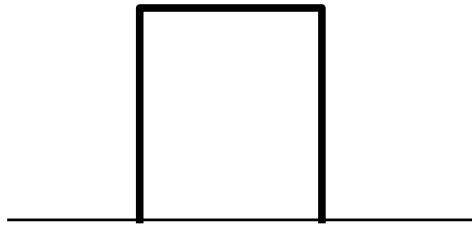


SLAM : other estimation approaches

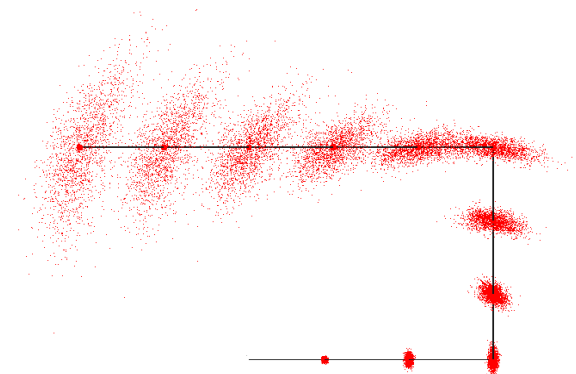
Stochastic approaches :



Kalman / information



Ensemblist



Particles

Global minimization approaches (e.g. bundle adjustment, scan matching)

SLAM: outline

- Simultaneous localisation and mapping
 - Problem presentation
 - Basics on estimation
 - EKF SLAM
 - Main issues
- Vision-based SLAM

Beyond estimation

Functions required by any SLAM implementation :

- Landmark detection
- Relative observations (measures)
 - Of the landmark positions
 - Of the robot motions
- Observation associations
- Refinement of the landmark and robot positions
- A perception process
- A perception process
- Control, signal processing...
- A perception process
- An estimation process

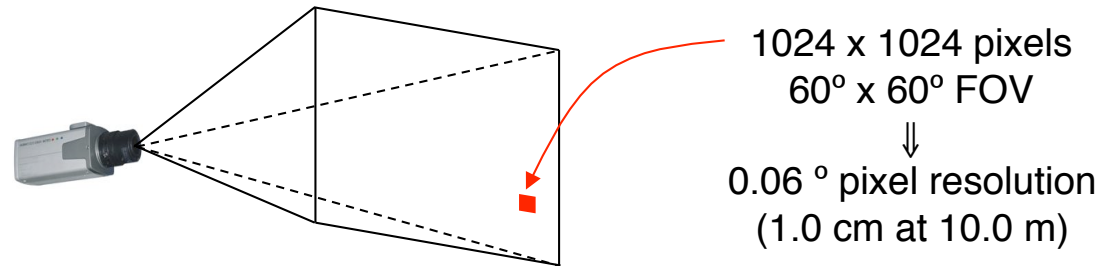
Use vision !

Micro UAVs

- Cameras : low cost, light and power-saving

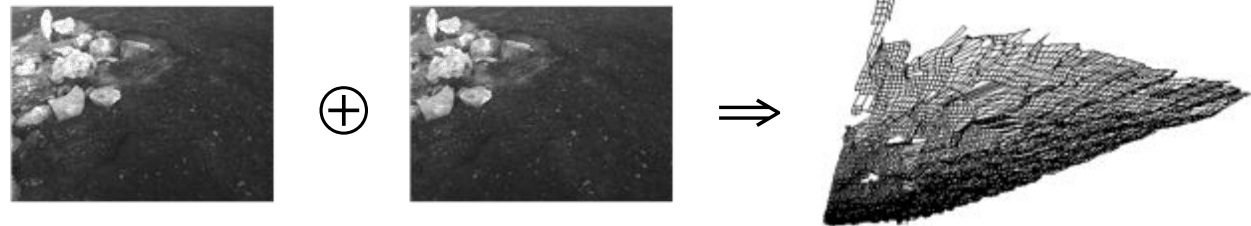


- Perceive data
 - In a volume
 - Very far
 - Very precisely



- Loss of depth is (almost) not anymore a difficulty

- Stereovision
 - 2 cameras provide depth



- Images carry a vast amount of information
- A vast know-how exists in the computer vision community

SLAM : what kind of landmarks ?

A good landmark:

- Should be *discriminant* (easy to associate)
- Should be invariant wrt. the viewpoint
- Its position (or part of it) should be observable — with an associated *error model*

Some point features have these properties:

- Harris features
- SIFT features

Stereovision SLAM

- Landmark detection → Vision : interest points
- Relative observations (measures)
 - Of the landmark positions → Stereovision
 - Of the robot motions → Visual motion estimation
- Observation associations → Interest points matching
- Refinement of the landmark and robot positions → Extended Kalman filter

Landmark observation : stereovision

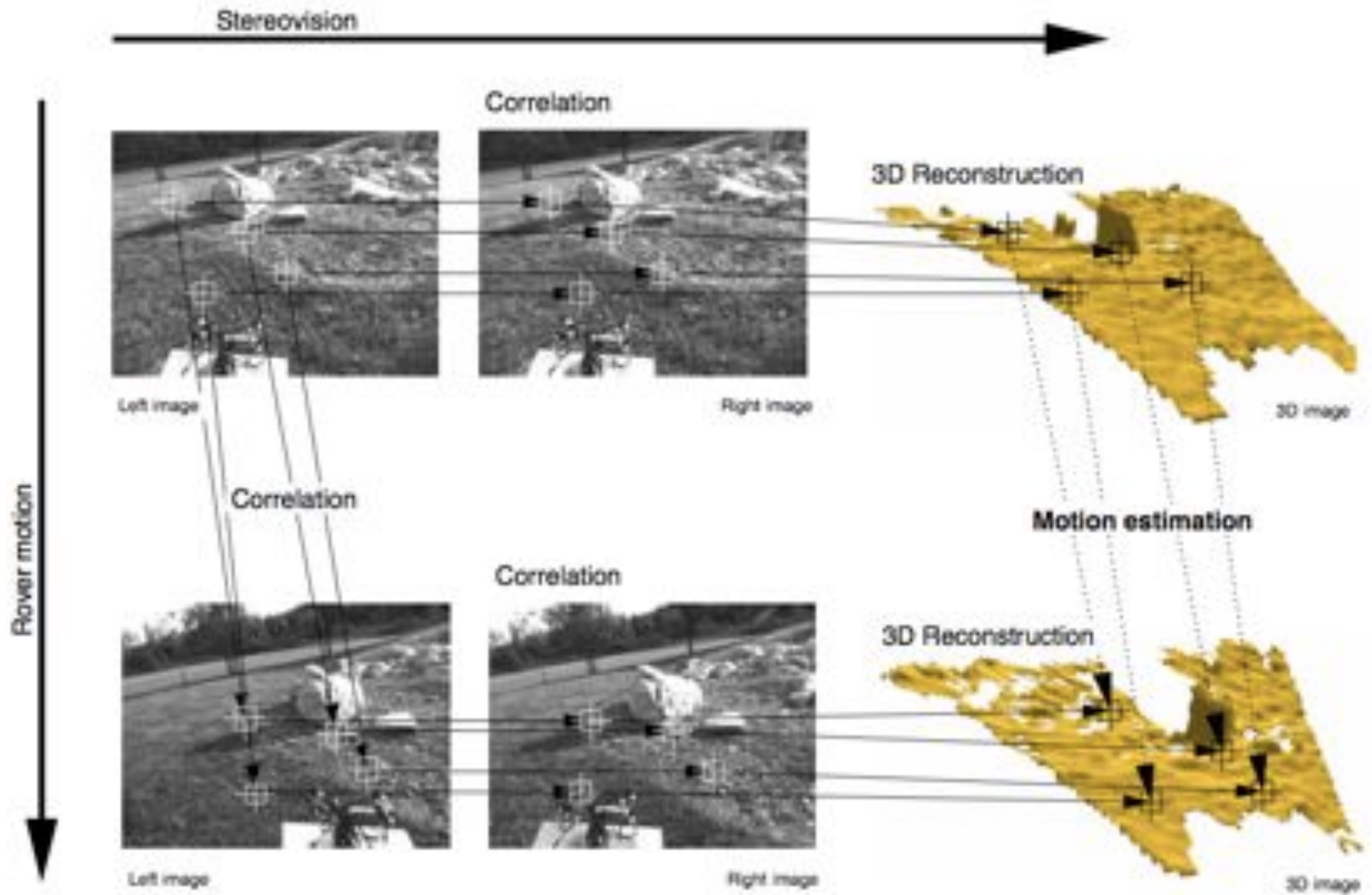
Dense
stereovision

or

IP matching
applied on stereo
frames (even
easier)



Visual odometry: principle

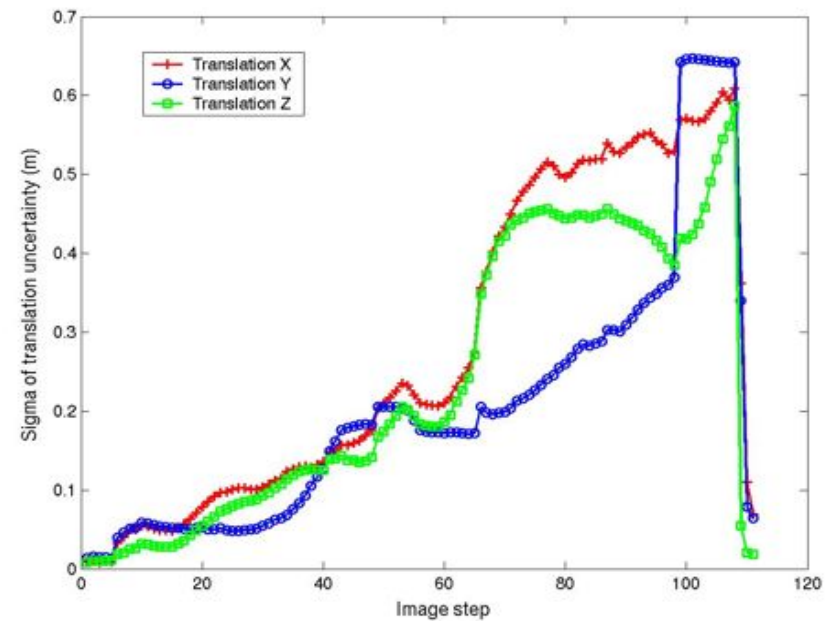
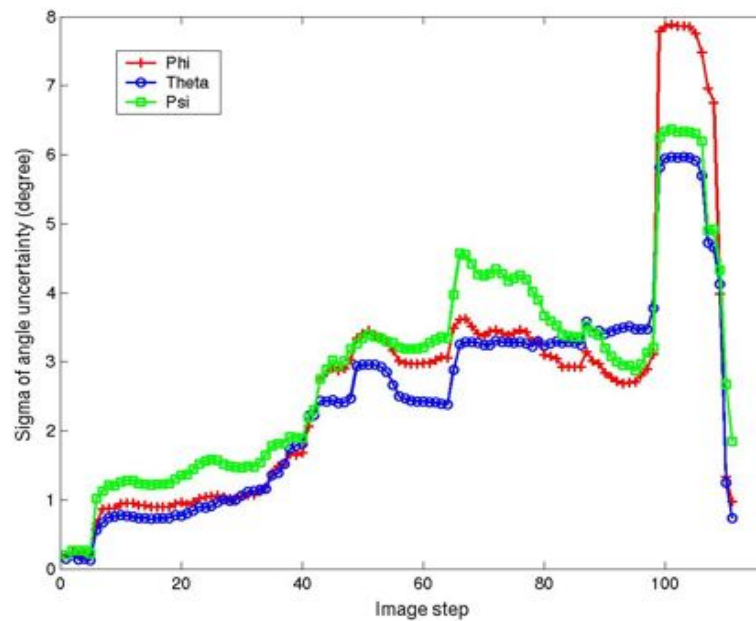
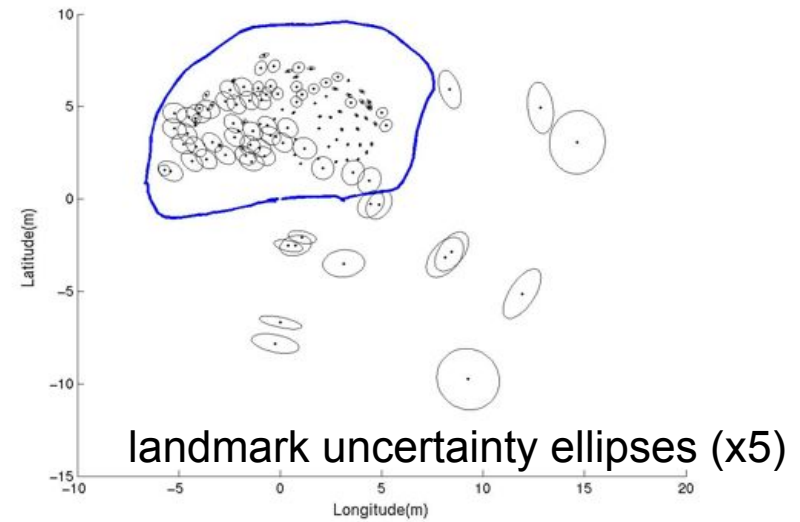


Stereovision SLAM

- Landmark detection → Vision : interest points \Leftarrow OK
- Relative observations (measures)
 - Of the landmark positions → Stereovision \Leftarrow OK
 - Of the robot motions → Visual motion estimation \Leftarrow OK
- Observation associations → Interest points matching \Leftarrow OK
- Refinement of the landmark and robot positions → Extended Kalman filter \Leftarrow OK

On a ground rover

- 110 stereo pairs processed, 60m loop



On a ground rover

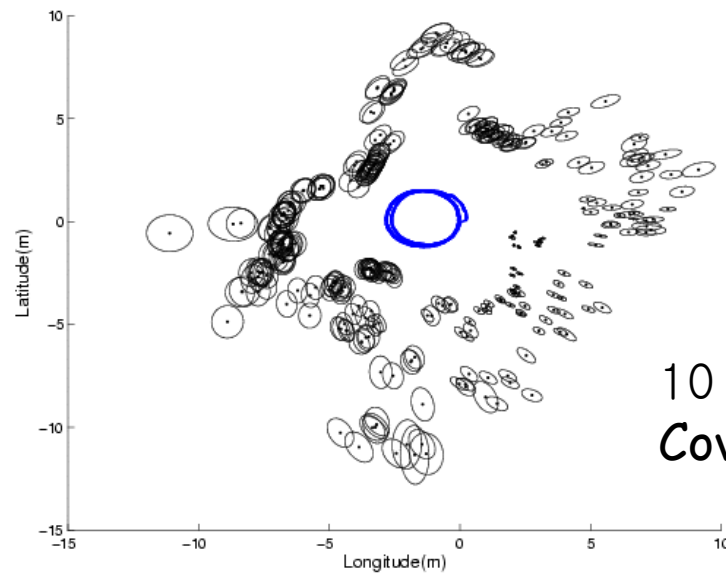
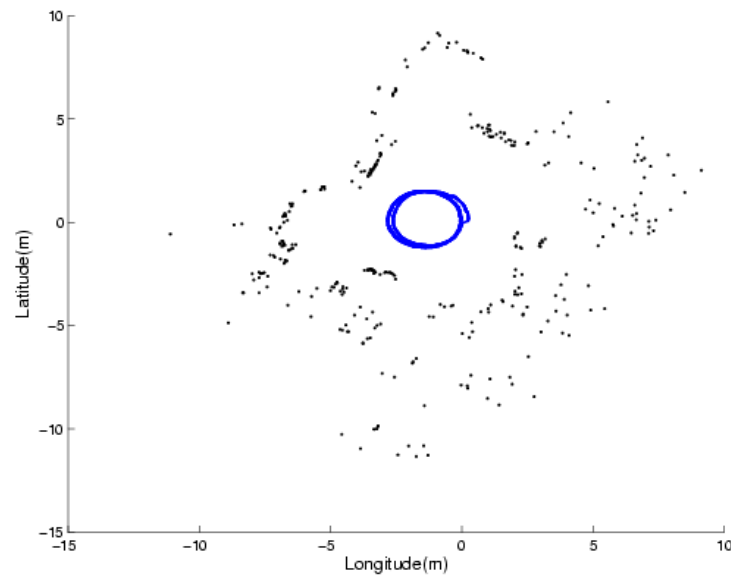
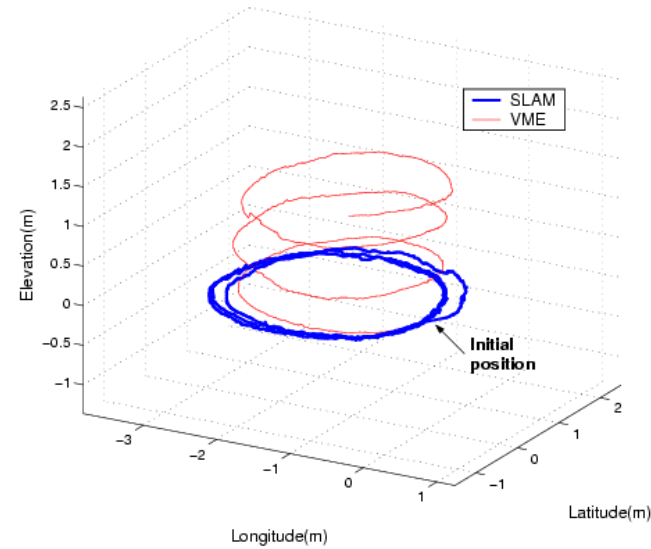
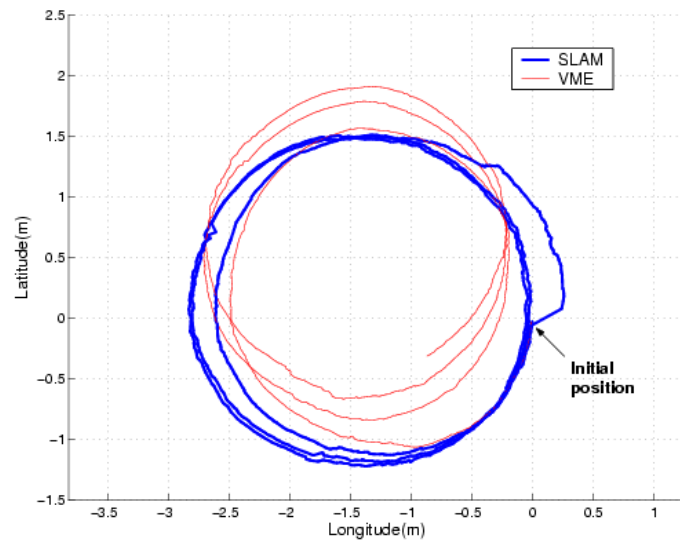


- 110 stereo pairs processed, 60m loop

	Frame 1/100 Reference	Reference Std. Dev.	VME result	VME Abs.error	SLAM result	SLAM Std. Dev.	SLAM Abs. error
Θ	0.52°	0.31°	2.75°	2.23°	0.88°	0.98°	0.36 °
Φ	0.36°	0.25°	-0.11°	0.47°	0.72°	0.74°	0.36 °
Ψ	-0.14°	0.16°	1.89°	2.03°	1.24°	1.84°	1.38°
t_x	-0.012 m	0.010 m	0.057m	0.069 m	-0.077 m	0.069 m	0.065 m
t_y	-0.243 m	0.019 m	-1.018 m	0.775 m	-0.284 m	0.064 m	0.041 m
t_z	0.019m	0.015 m	0.144m	0.125 m	0.018m	0.019 m	0.001 m

In indoor environments

- About 30 m long trajectory, 1300 stereo image pairs

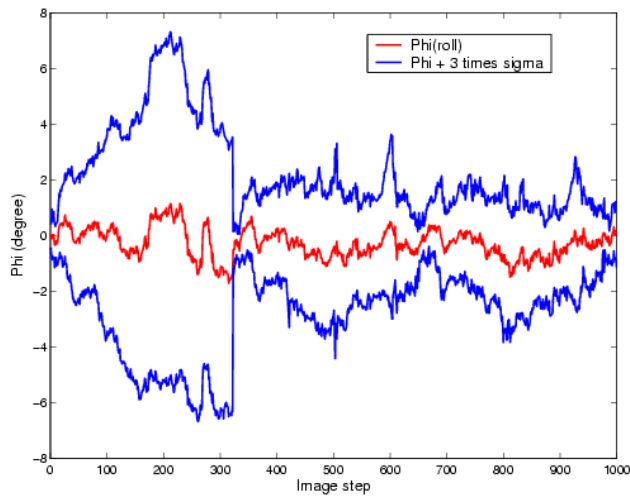
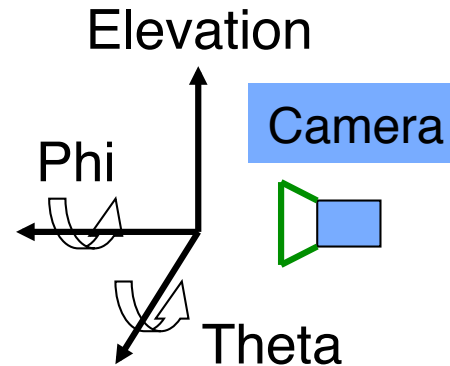


10 times
Cov. ellipse

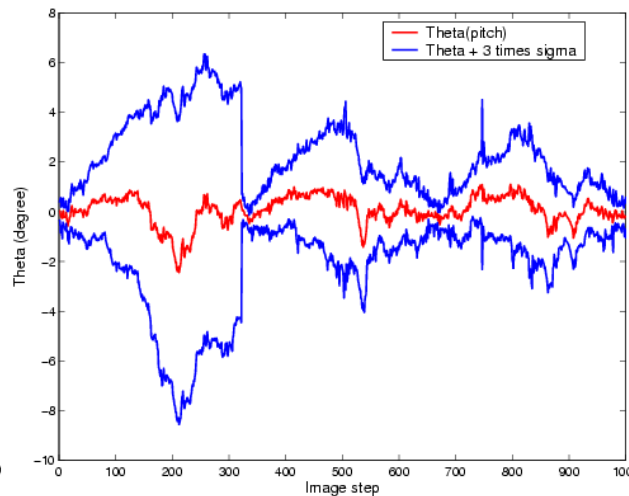
In indoor environments

- About 30 m long trajectory, 1300 stereo image pairs

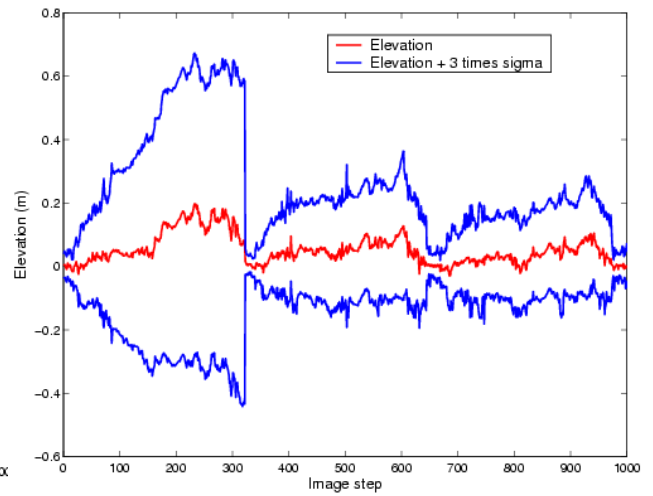
Two rotation angles (**Phi**, **Theta**) and **Elevation** must be zero



Phi



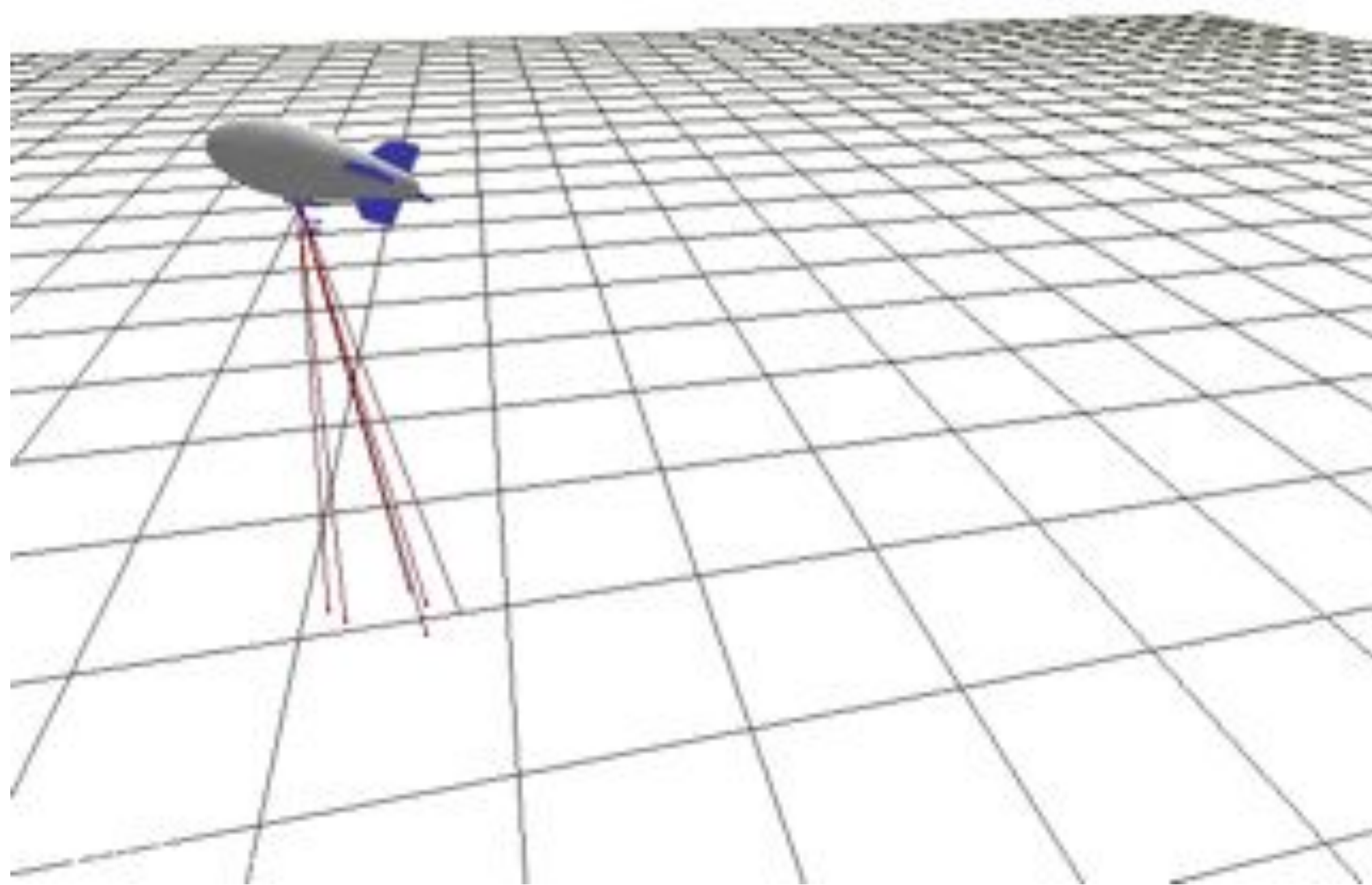
Theta



Elevation

With a blimp





SLAM: outline

- Simultaneous localisation and mapping
 - Problem presentation
 - Basics on estimation
 - EKF SLAM
 - Main issues
- Vision-based SLAM
 - SLAM with stereovision
 - SLAM with monocular imagery

SLAM with monocular vision

Micro UAVs



Smartphones, wearable devices



SLAM with monocular vision

Prediction: any information on the robot motion

Landmarks:

- interest points = 3D (x,y,z) points



Application of the « usual » SLAM steps:

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

Partial observations!
(« bearings-only »)

But why is it particularly difficult?

Bearing-only SLAM

Generic SLAM

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

Stereovision SLAM

- Vision : interest points
 - Stereovision
 - Visual motion estimation
- Interest points matching
- Extended Kalman filter

Bearing-only SLAM

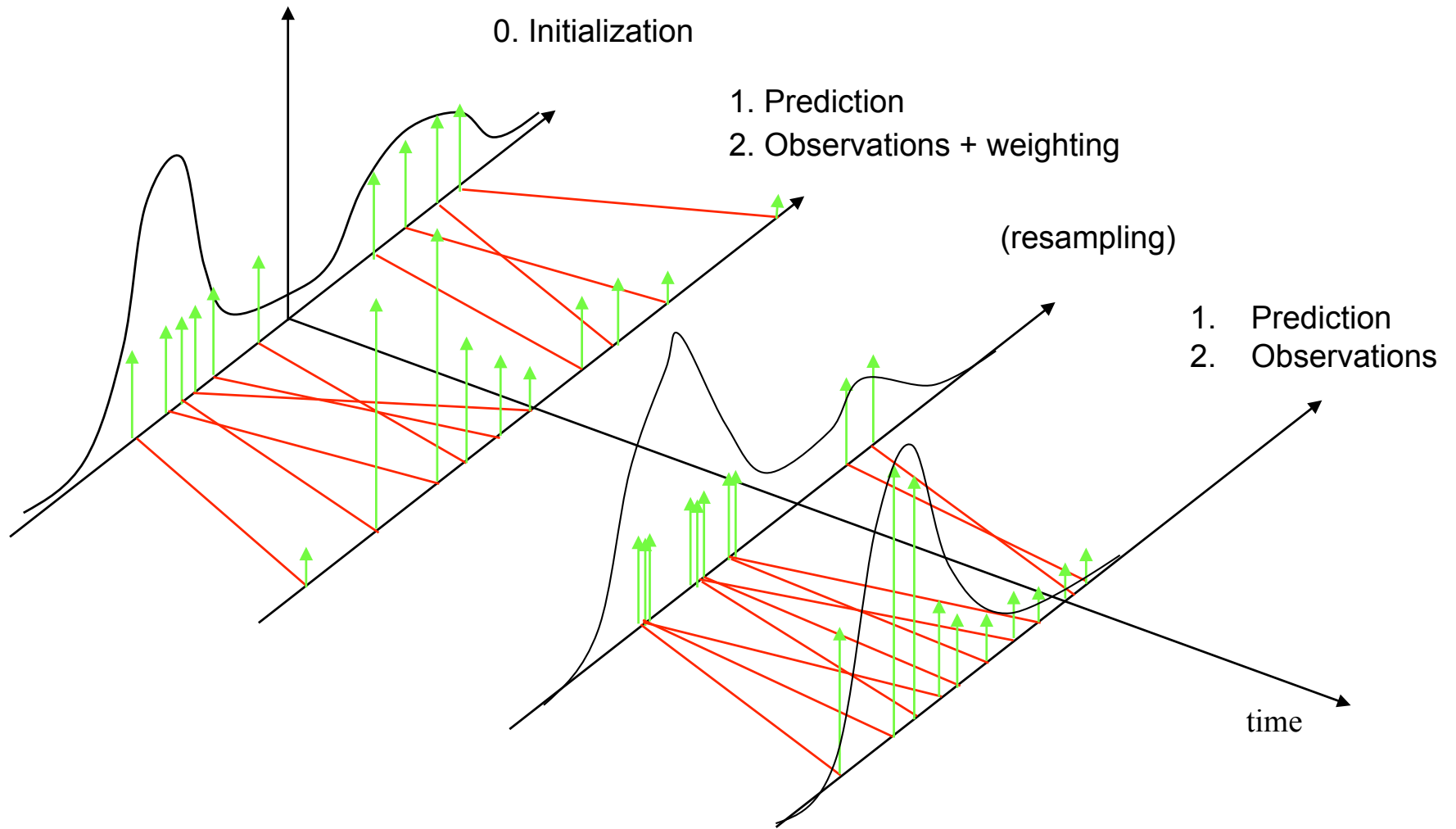
Generic SLAM

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

Monocular SLAM

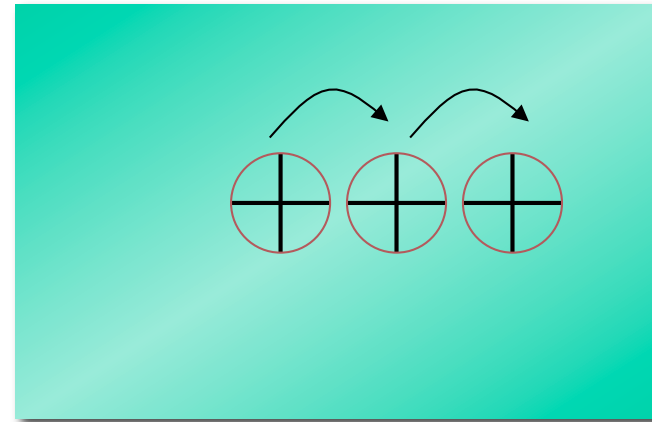
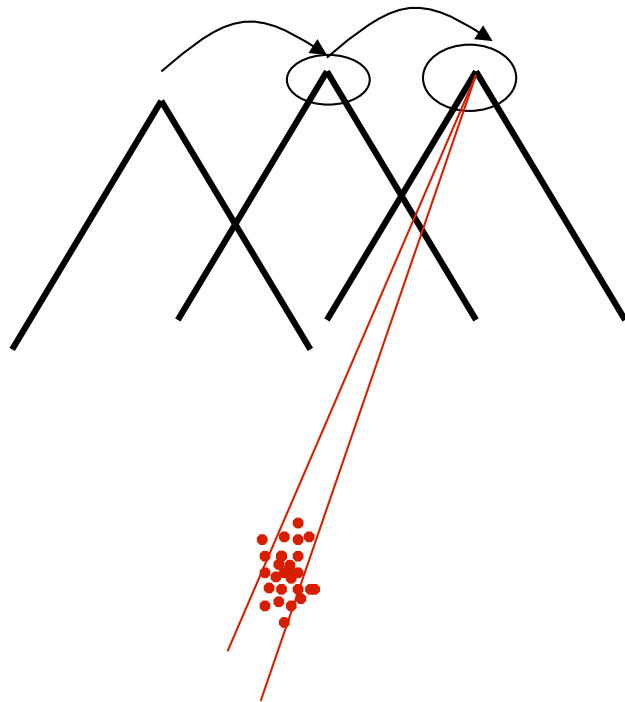
- Vision : interest points
 - « Multi-view stereovision »
 - INS, Motion model...
- Interest points matching
- Extended Kalman filter

(Particle filtering)



Bearing-only SLAM: landmark initialisation

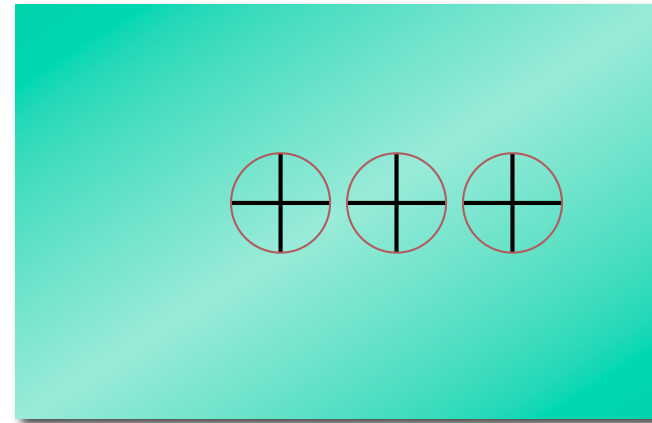
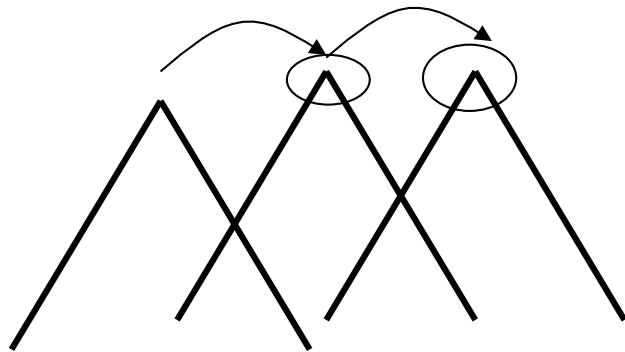
« Initialisation filter » \approx particle filter



One interest point
matched in the image

Bearing-only SLAM: landmark initialisation

« Initialisation filter » \approx particle filter



One interest point
matched in the image

Issues with this approach:

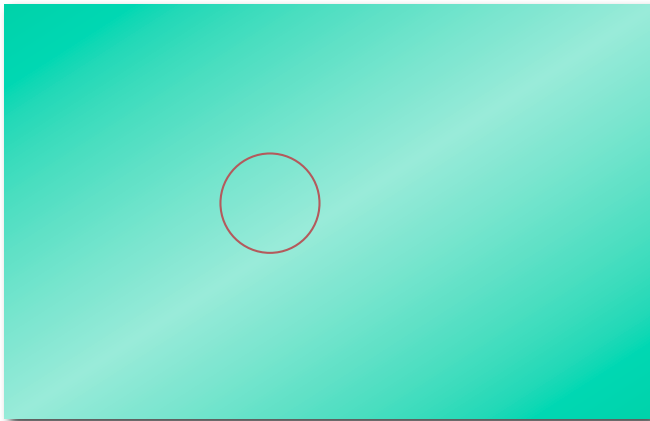
- Complexity
- Some landmarks remain non-observable
- Numerous observations are lost

Landmark initialisation

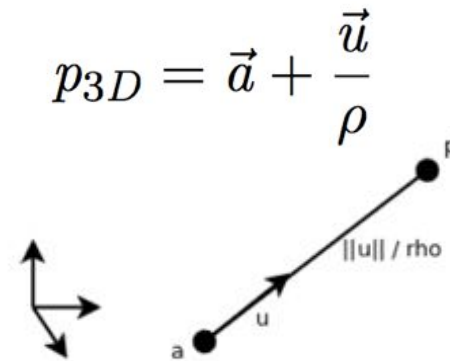
« Initialisation », 2nd approach: inverse depth parametrization

Solution established in 2006 [J. Montiel / A. Davison]:

- *Direct* initialization of a point the first time it is perceived
- Allows to consider points located at the infinite



One interest point in the image



One landmark in the MAP

- Every detected landmark is directly and immediatly observable

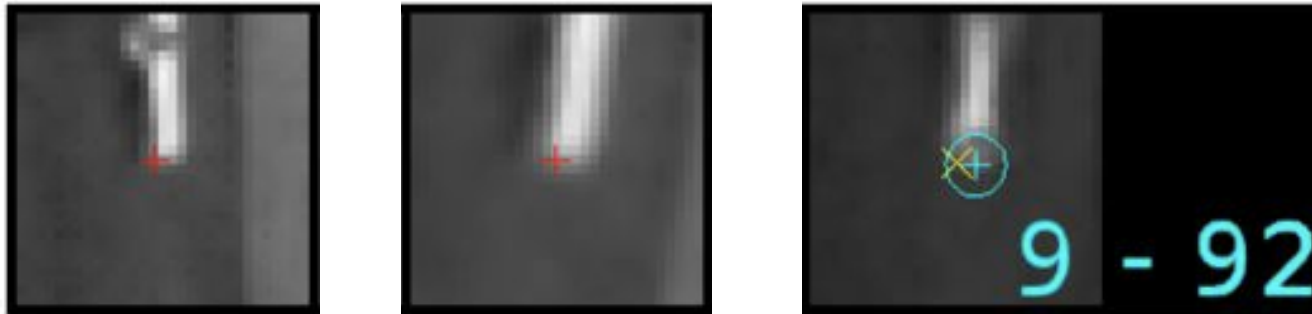
A closer look at the motion prediction

Constant velocity model

$$\text{Robot state: } \mathcal{R} = (\mathbf{p} \ \mathbf{q} \ \mathbf{v} \ \mathbf{w})^T$$

$$\text{Prediction: } \mathbf{p}^+ = \mathbf{p} + \mathbf{v}.dt \quad \mathbf{q}^+ = \mathbf{q} * \mathbf{w}.dt \quad \mathbf{v}^+ = \mathbf{v} \quad \mathbf{w}^+ = \mathbf{w}$$

Allows to focus the data association step:



1. Start with a descriptor of the landmark (e.g. appearance at first observation)
2. Apply affine transformation to predict the current appearance
3. Search the landmark inside the observation uncertainty ellipse

A closer look at the motion prediction

Constant velocity model

$$\text{Robot state: } \mathcal{R} = (\mathbf{p} \ \mathbf{q} \ \mathbf{v} \ \mathbf{w})^T$$

$$\text{Prediction: } \mathbf{p}^+ = \mathbf{p} + \mathbf{v}.dt \quad \mathbf{q}^+ = \mathbf{q} * \mathbf{w}.dt \quad \mathbf{v}^+ = \mathbf{v} \quad \mathbf{w}^+ = \mathbf{w}$$

- simple (no additional sensor required)
- not precise
 - bad linearization points
 - large search areas
 - difficulty to track very high motion dynamics
- no scale factor estimate possible

A closer look at the motion prediction

Constant acceleration model

$$\text{Robot state: } \mathcal{R} = (\mathbf{p} \ \mathbf{q} \ \mathbf{v} \ \mathbf{w} \ \mathbf{v}_a \ \mathbf{w}_a)^T$$

$$\begin{aligned} \text{Prediction: } \quad \mathbf{v}^+ &= \mathbf{v} + \mathbf{v}_a \cdot dt \\ \mathbf{w}^+ &= \mathbf{w} + \mathbf{w}_a \cdot dt \\ \mathbf{v}_a^+ &= \mathbf{v}_a \\ \mathbf{w}_a^+ &= \mathbf{w}_a \end{aligned}$$

- simple (no additional sensor required)
- not very precise
 - bad linearization points
 - large search areas
- no scale factor estimate possible

A closer look at the motion prediction

Using an inertial measurement unit

$$\text{Robot state: } \mathcal{R} = (\mathbf{p} \ \mathbf{q} \ \mathbf{v} \ \mathbf{a}_b \ \mathbf{w}_b \ \mathbf{g})^T$$

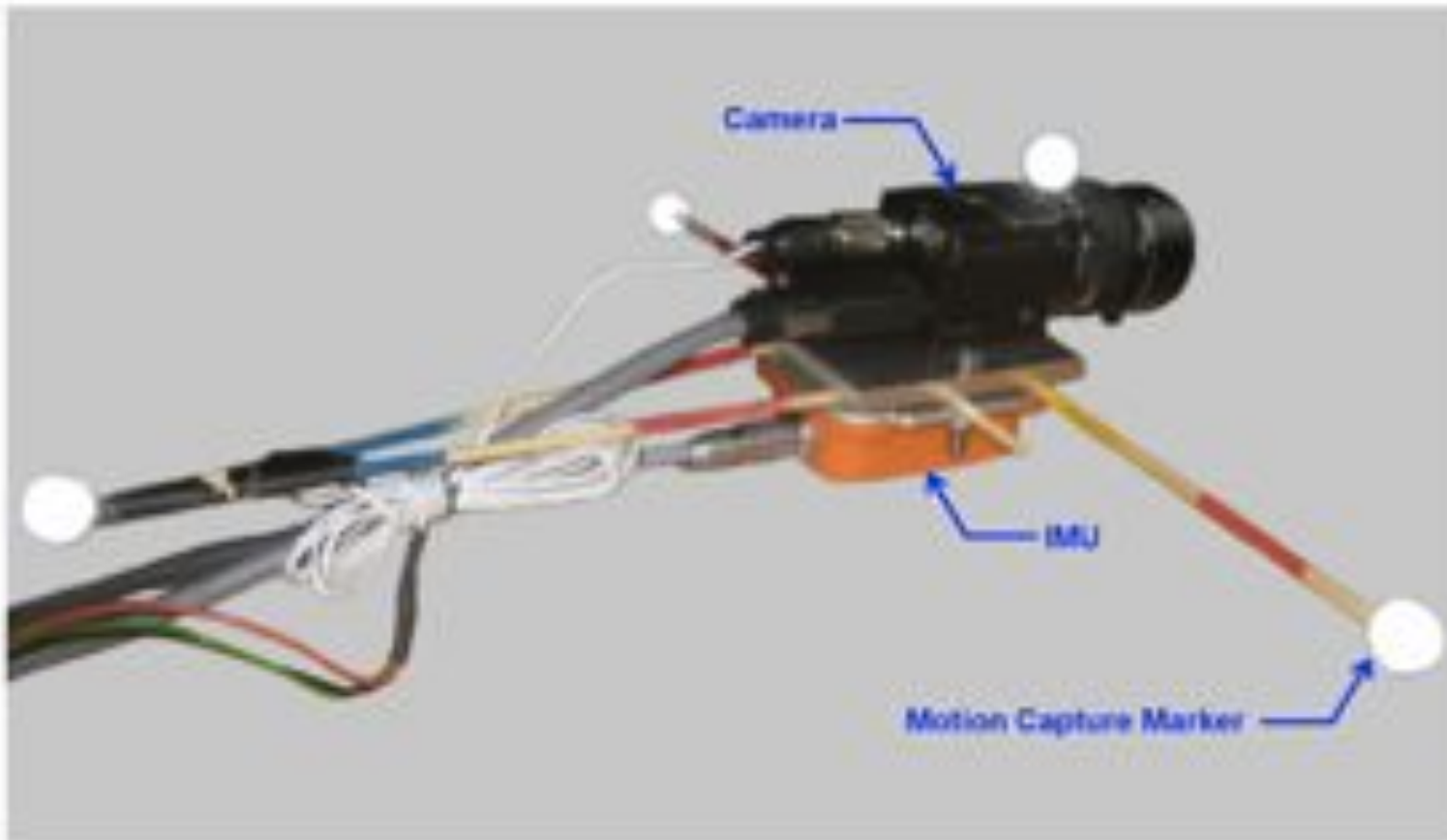
Prediction:

$$\begin{aligned}\mathbf{p}^+ &= \mathbf{p} + \mathbf{v} \cdot dt \\ \mathbf{q}^+ &= \mathbf{q} \otimes v2q((\mathbf{w}_m - \mathbf{w}_b) \cdot dt) \\ \mathbf{v}^+ &= \mathbf{v} + (q2R(\mathbf{q}) \cdot (\mathbf{a}_m - \mathbf{a}_b) + \mathbf{g}) \cdot dt \\ \mathbf{a}_b^+ &= \mathbf{a}_b \\ \mathbf{w}_b^+ &= \mathbf{w}_b \\ \mathbf{g}^+ &= \mathbf{g}\end{aligned}$$

- additional sensor required
- precise
 - good linearization points
 - small search areas
 - possibility to track very high motion dynamics
- scale factor estimate possible

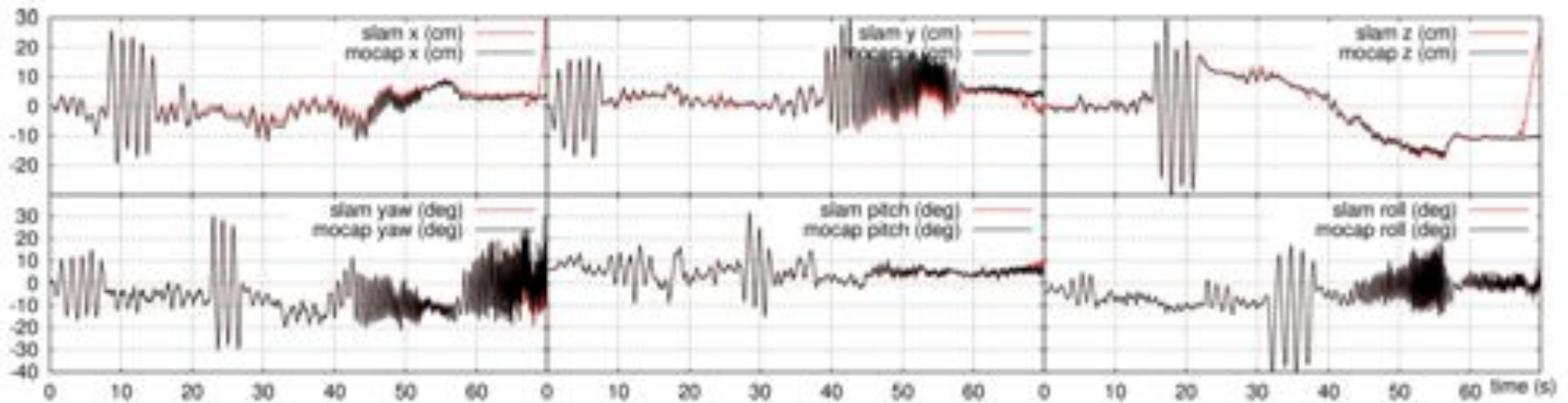
A closer look at the motion prediction

Using an inertial measurement unit: results



A closer look at the motion prediction

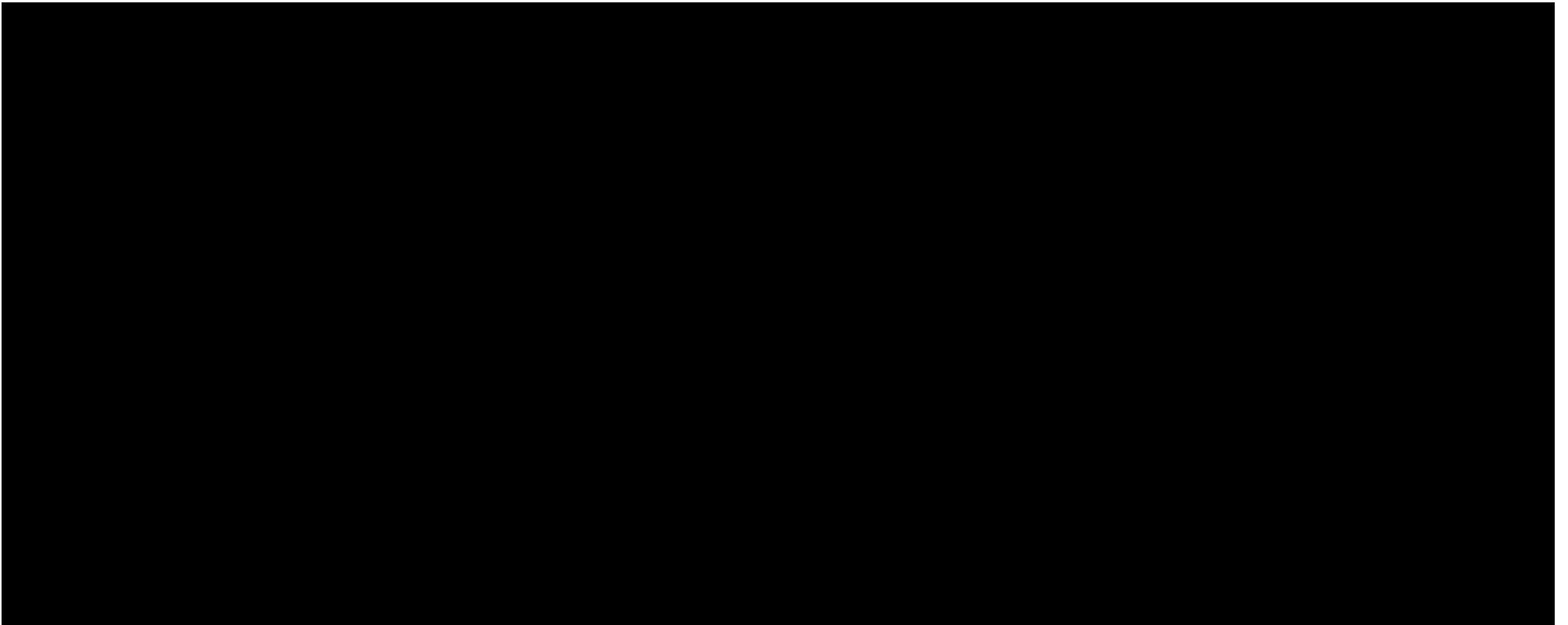
Using an inertial measurement unit: results



Estimated trajectory of the 6 position parameters

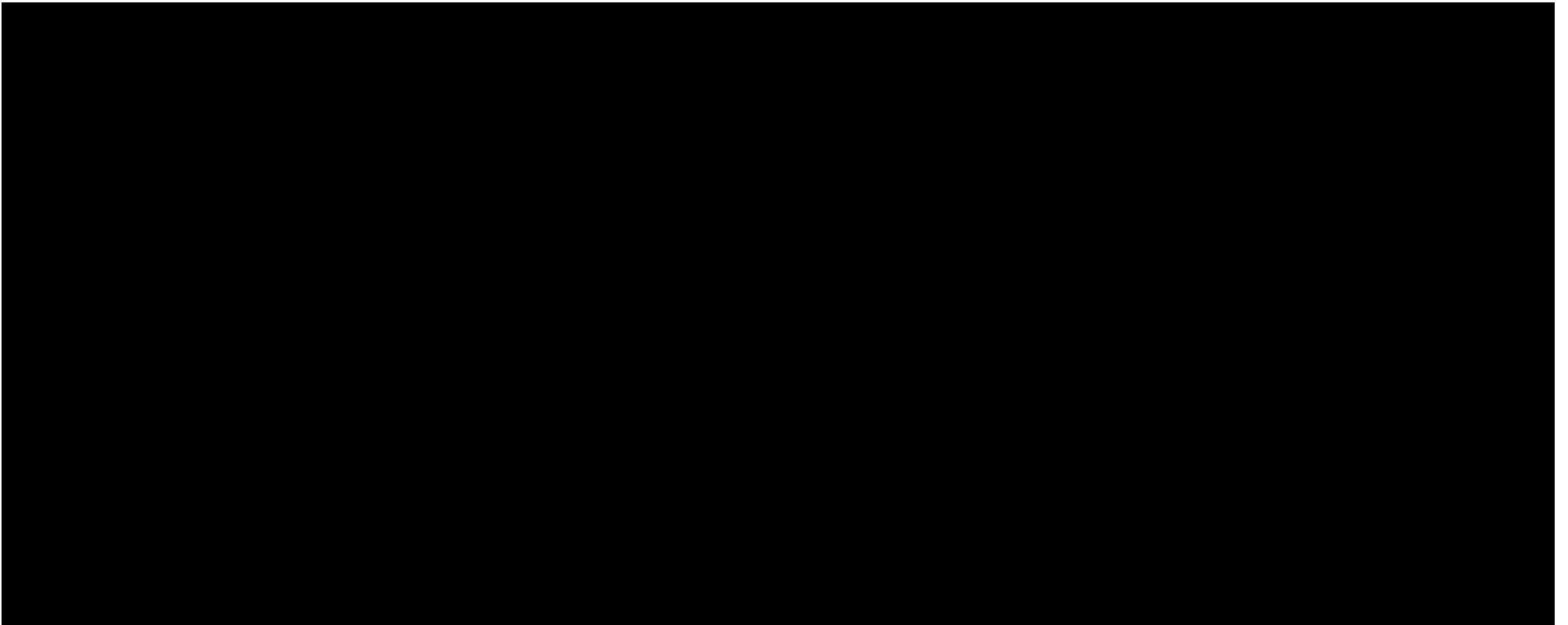
A closer look at the motion prediction

Using an inertial measurement unit: results



A closer look at the motion prediction

Using an inertial measurement unit: results



A closer look at the motion prediction

Using an inertial measurement unit: summary

- An additional proprioceptive sensor can help if :
 - high frequency
 - not too noisy
- Can also be directly used in the prediction step if :
 - complete
 - never faulty

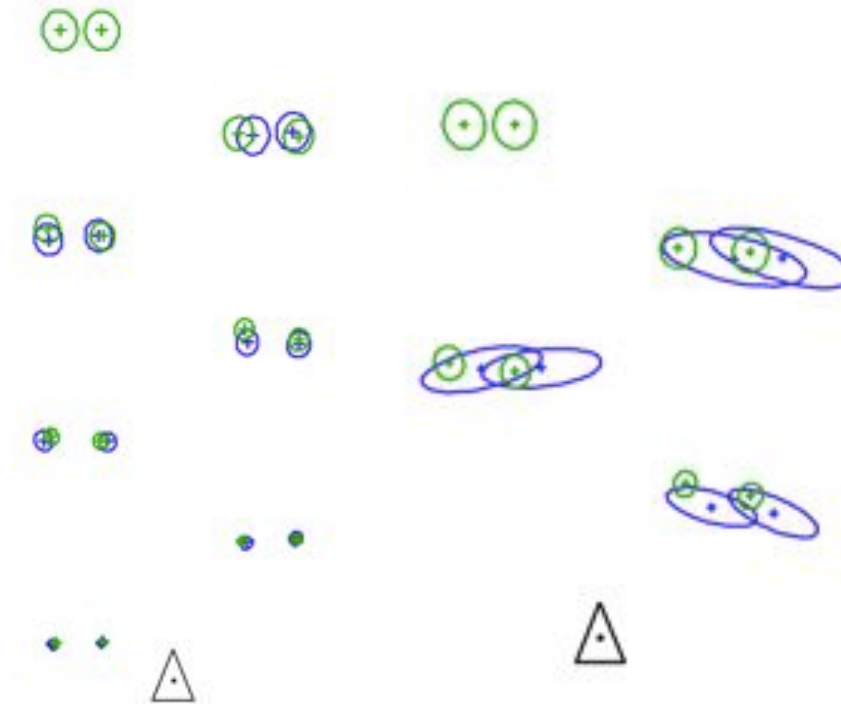
SLAM: outline

- Basics of simultaneous localisation and mapping
- SLAM with monocular vision
 - Solutions to the landmark initialisation problem
 - Importance of the motion prediction, introduction of an IMU
 - What about loop-closing?

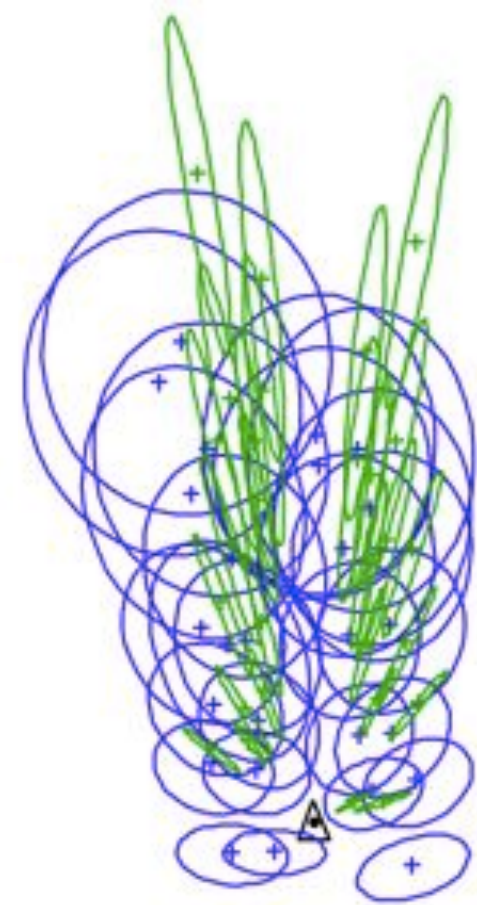
A closer look at data association

Data association relying on geometric information can become tricky...

Blue: predicted
(mapped) features
Green: observed
features



2 nearly consecutive cases



Large loop-closing
case

→ Better use landmark matching techniques

Overall summary

- Autonomous robots, on the importance of localization
- Localization using dead-reckoning approaches
- SLAM
 - Problem presentation
 - Basics on estimation
 - EKF SLAM
 - Main issues
- Vision-based SLAM
 - SLAM with stereovision
 - SLAM with monocular imagery
 - Solutions to the landmark initialisation problem
 - Illustrations
 - Importance of the motion prediction, introduction of an IMU
 - (Visual loop-closing detection)