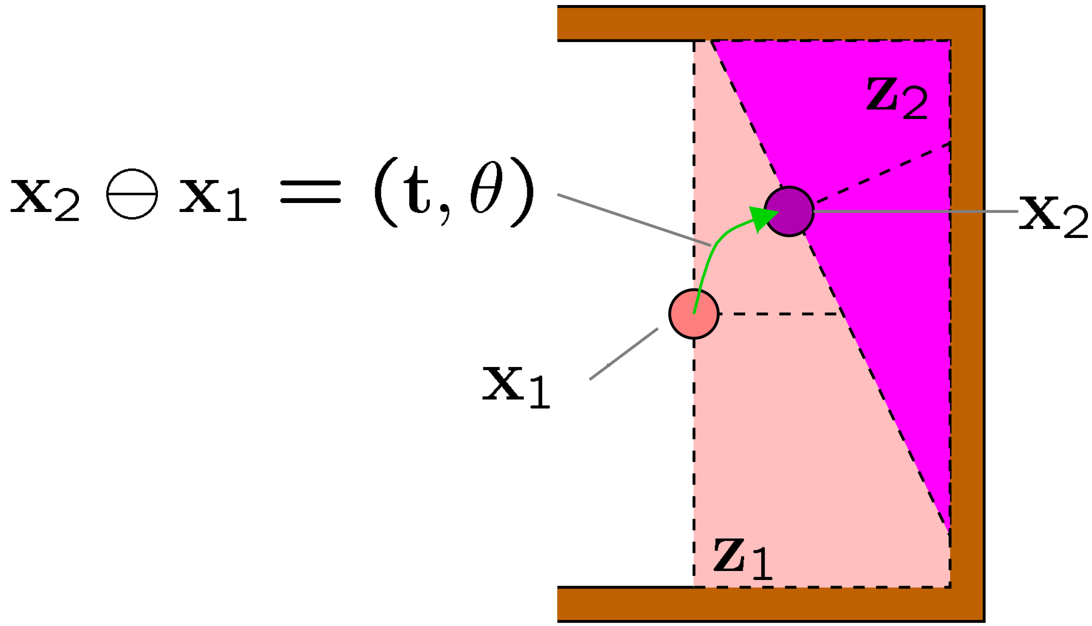
Scan matching in a probabilistic framework

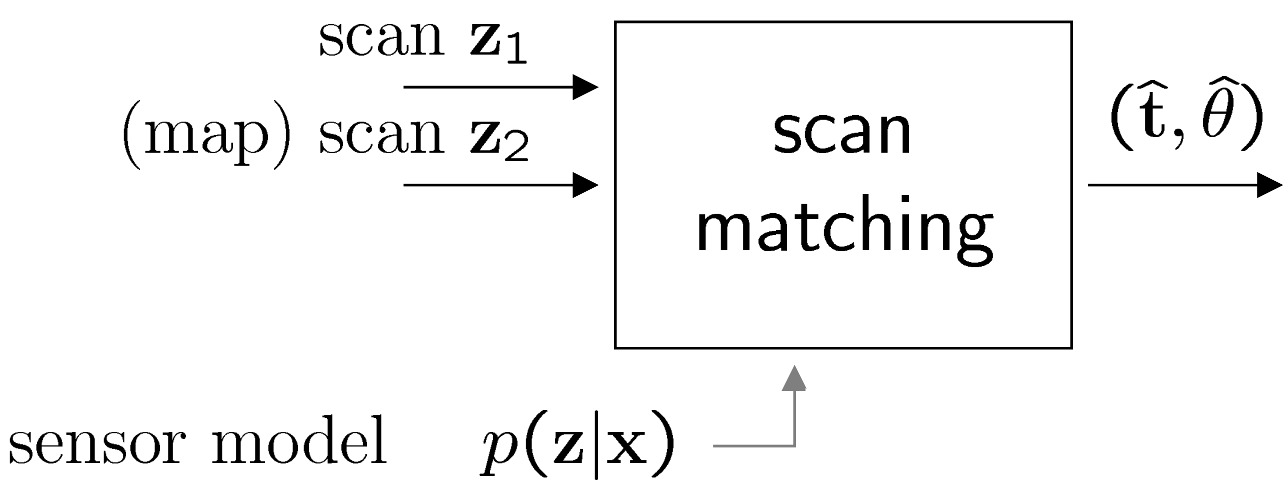
Andrea Censi

a master student in control engineering at
Universitá degli Studi di Roma *La Sapienza*
www.dis.uniroma1.it/∼acensi
andrea.censi@dis.uniroma1.it

What is scan matching

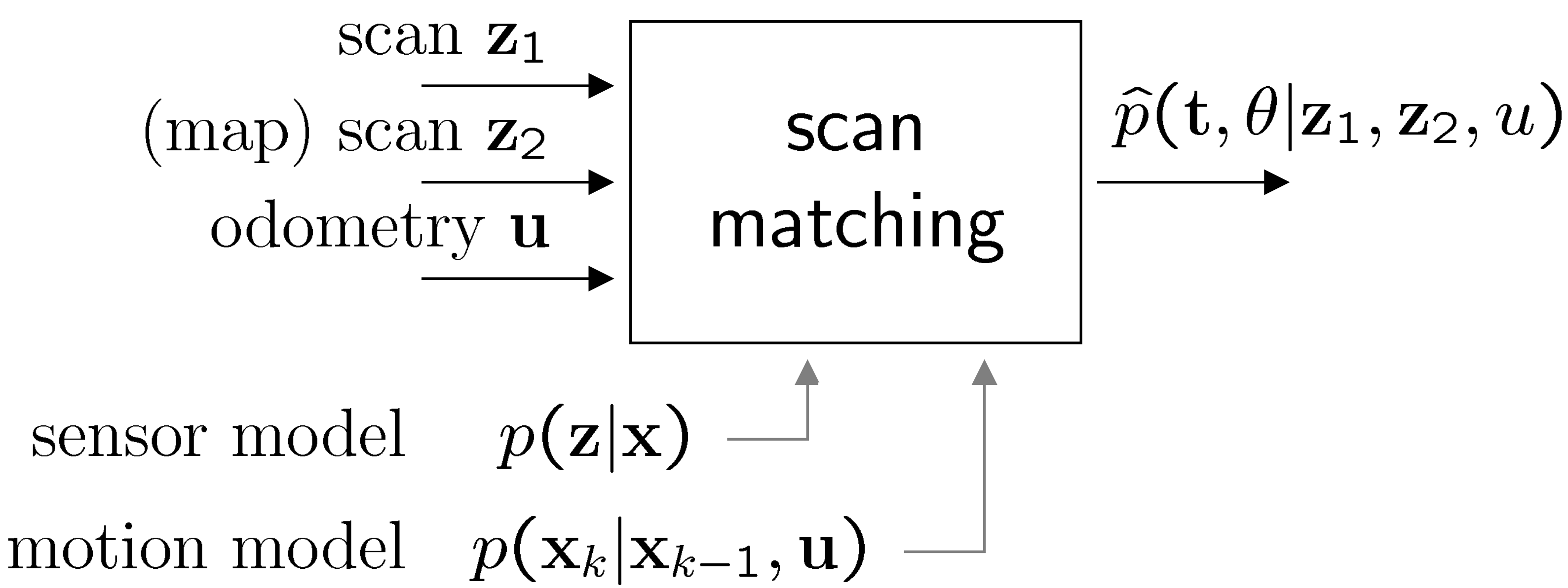


What is scan matching



Geometric interpretation: Find a rotation φ and a translation $\mathbf{t}$ which maximize the overlapping of two sets of 2D-data.

What is scan matching



Geometric interpretation: Find a rotation φ and a translation $\mathbf{t}$ which maximize the overlapping of two sets of 2D-data.

Probabilistic interpretation:
Find an approximation to the probability distribution

$$p(\mathbf{t}, \varphi | \mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{z}_t, \mathbf{z}_{t-1})$$

$\mathbf{x}$: robot pose, $\mathbf{z}$: sensor reading, $\mathbf{u}$: odometry.

Contribution of the paper

"GPM" is a new algorithm that

— uses, soundly, an arbitrary evolution model; no random sampling required *Gaussian assumption: [Minguez&al.'05]; Random sampling:* MCL*, [Silver&al.'04]*

— characterizes the uncertainty analytically, also in underconstrained situations *Sample error function around the estimate: [Bengtsson&al.'03]; analytic, elegant but bounded estimate of covariance: [Pfister&al.'02]*

— not iterative: result does not depend on first guess

Weak points of GPM:

— The environment must have some regularity to estimate surfaces' orientation.

— It is more precise than ICP, IDC, but not than last generation ICP-like methods. *[Minguez&al.'06]*

GPM overview

It is a dual of Monte Carlo Localization:

— In MCL, particles are drawn from the evolution model and weighted by the observation model.

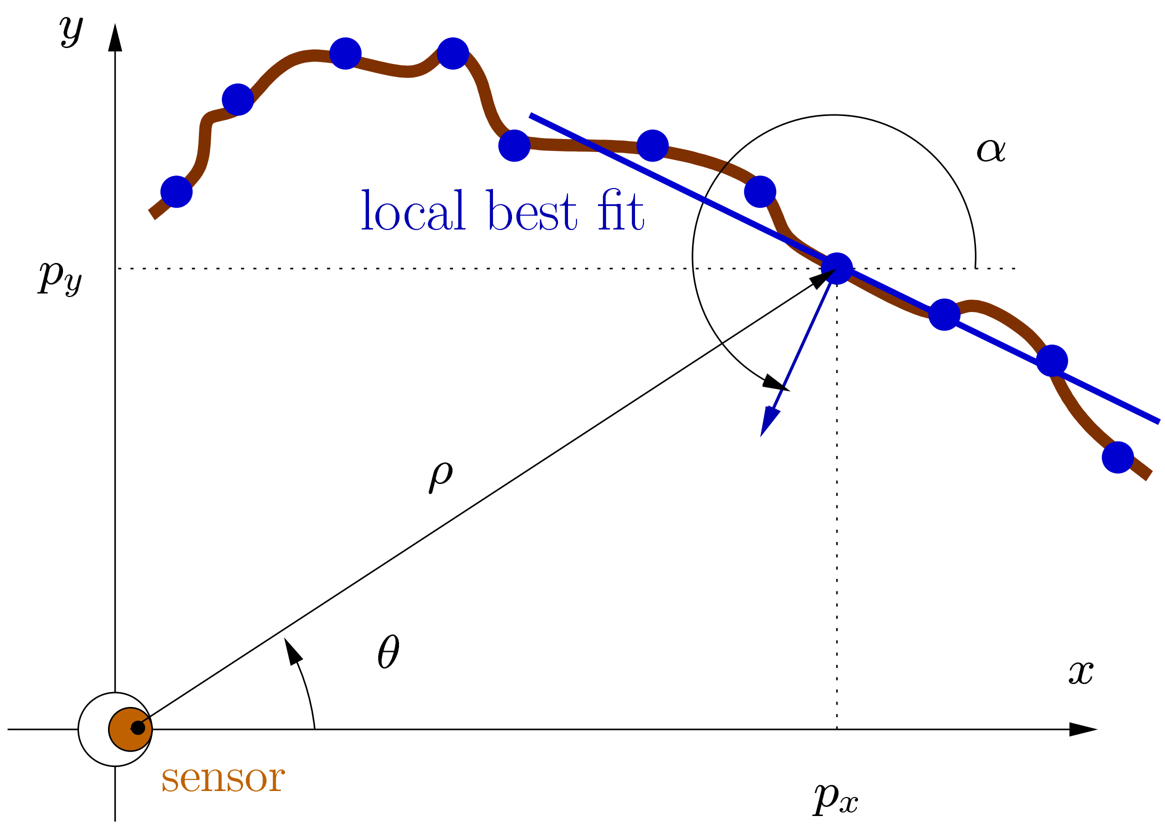
— In GPM, particles are generated (deterministically) from the observation model and weighted by the evolution model.

Summary of the algorithm:

1. Extract orientation information from the sensor data.
2. Generate a cloud of particles from the observations.
3. Weight each particle according to the evolution model.
4. Turn the particles into "constraints" to characterize uncertainty.

Extracting the orientation

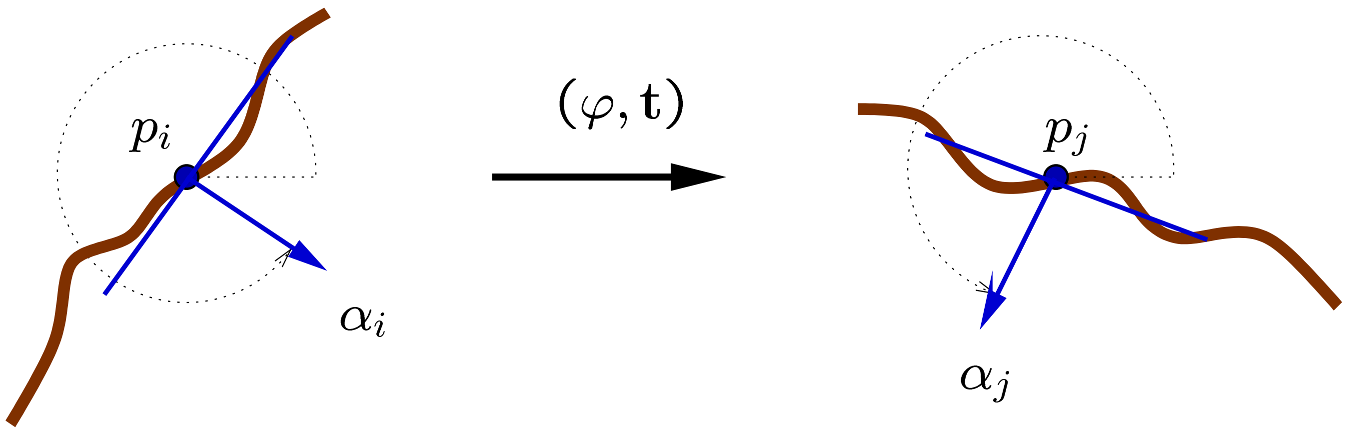
The input data are two sets of “oriented” points $\{(p_i, \alpha_i,)\}$, where p_i is the cartesian point and α_i is the direction of the normal to the surface.



Currently using linear regression; there are many alternatives.

Generating the particles

We create a set of hypotheses (particles) by considering all possible pairs of points (**no correspondence heuristics**).



$$
\begin{aligned}
p_j &= R_\varphi p_i + \mathbf{t} \\
\alpha_j &= \alpha_i + \varphi
\end{aligned}
$$

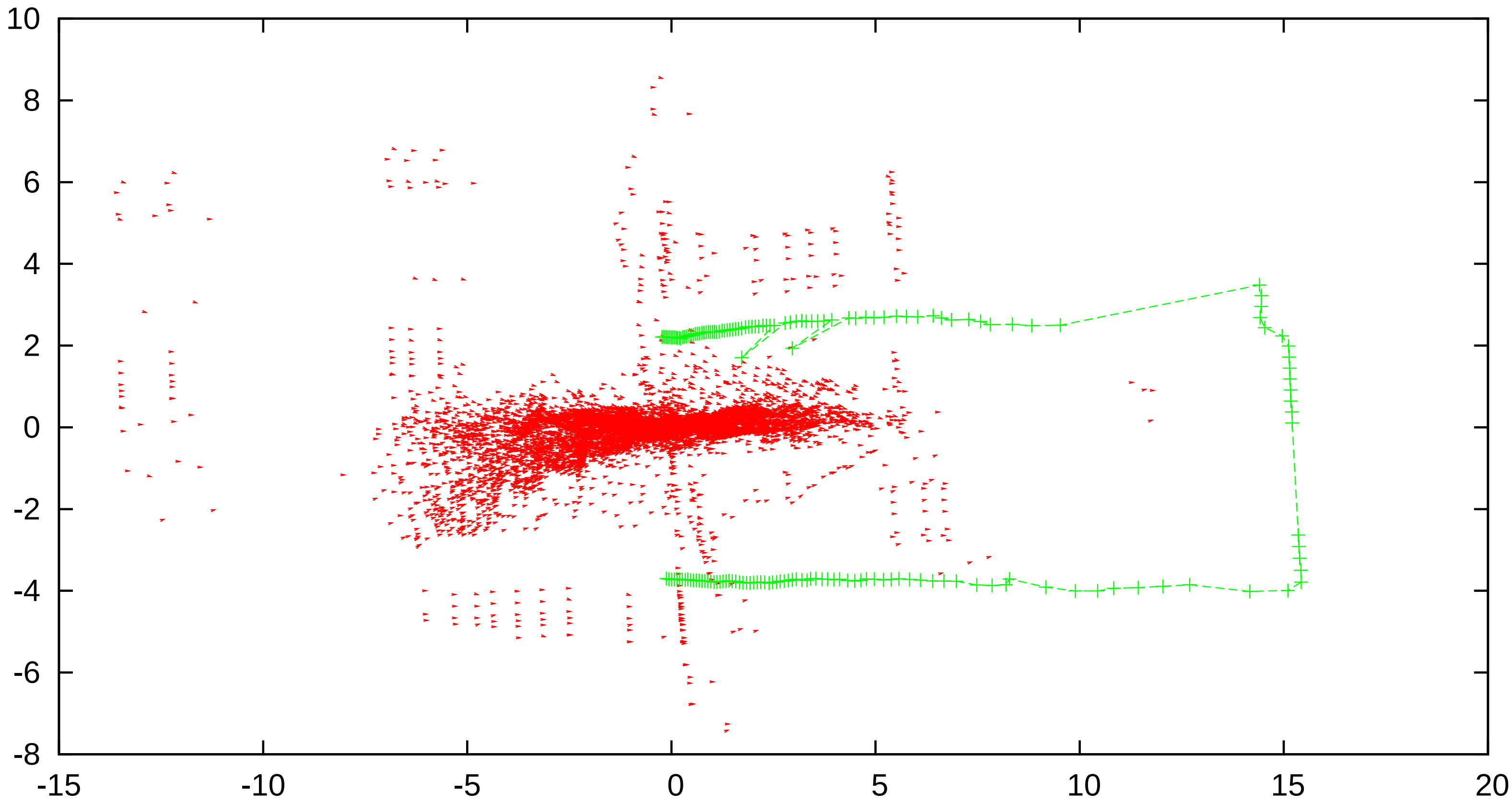
Invert to obtain

$$
\begin{aligned}
\hat{\varphi} &= \alpha_j - \alpha_i \\
\hat{\mathbf{t}} &= p_j - R_{\hat{\varphi}} p_i
\end{aligned}
$$

Each hypothesis $(\hat{\varphi}, \hat{\mathbf{t}})$ is treated as a particle (generated **deterministically**; no random sampling here).

Example (1)

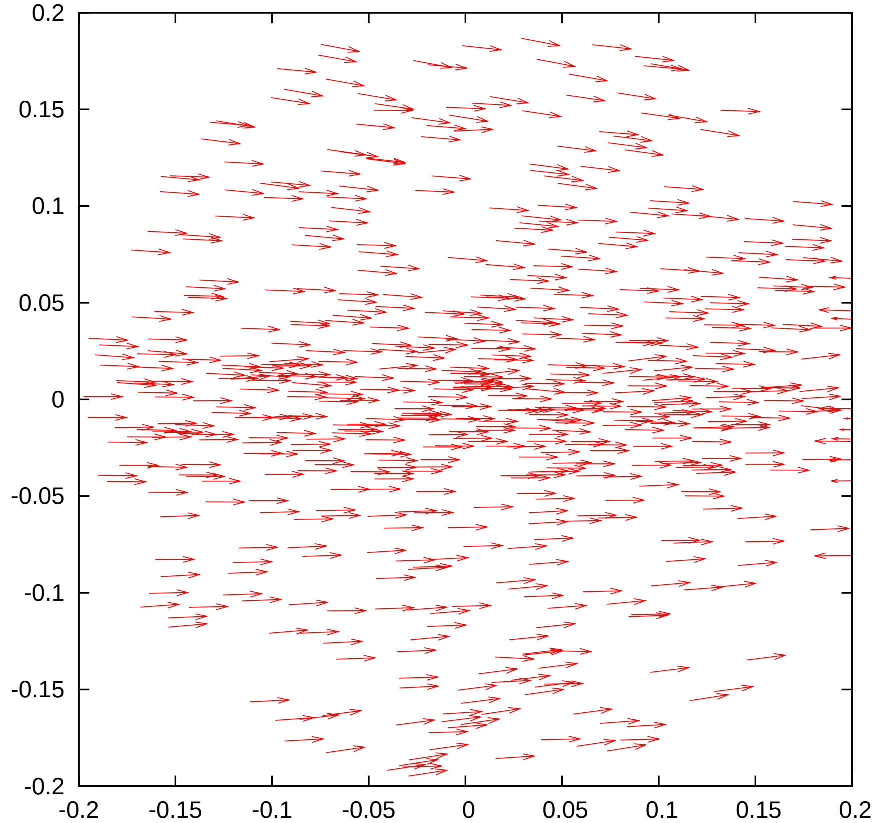
This is how the set of particles looks like:
(green is one of the sensor scans; particles are red)



We consider only the particles in a fixed ball where the evolution model is non-zero.

Example (2)

Particles with $|\varphi| \leq 20°$, $|\mathbf{t}| \leq 20cm$.

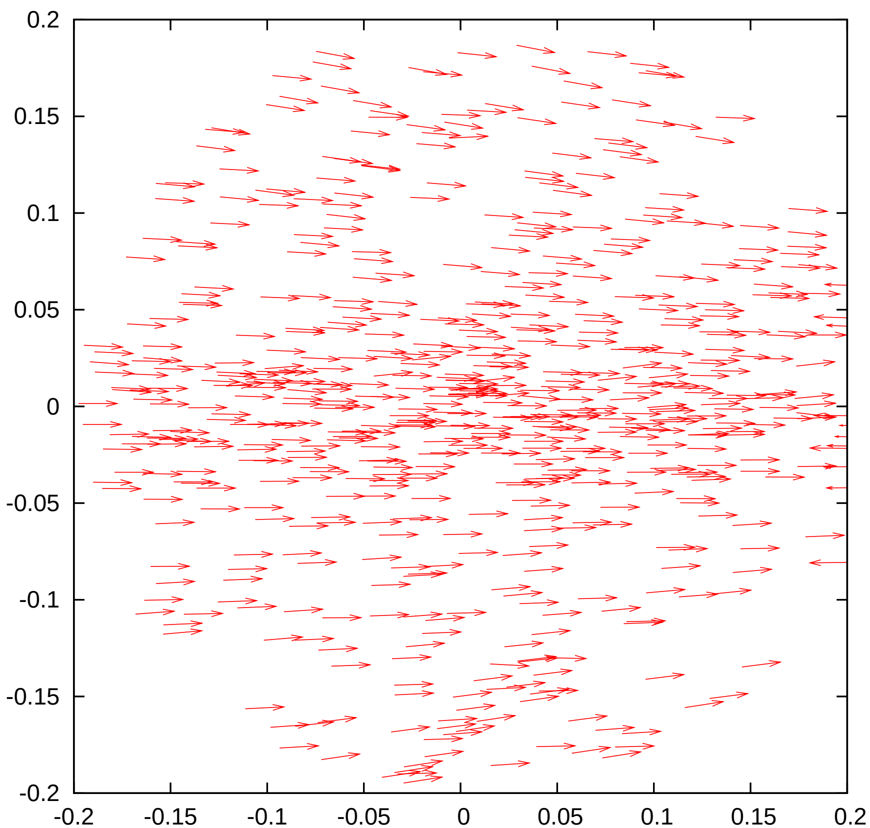


$\Rightarrow$ it is a particle approximation to $p(\varphi, \mathbf{t}|\mathbf{x}_{t-1}, \mathbf{z}_t, \mathbf{z}_{t-1})$
(little arrows represent $\hat{\varphi}$)

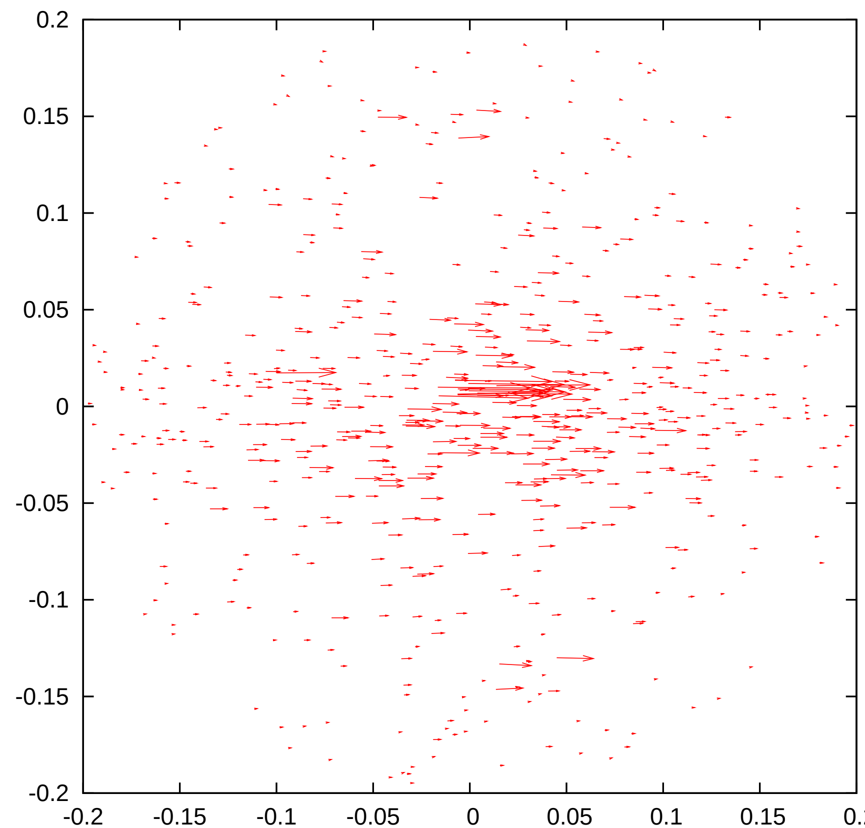
Using the evolution model

Weight by evolution model:

$$w_k = p(\varphi_k, \mathbf{t}_k | \mathbf{x}_{t-1}, \mathbf{u}_t)$$



before weighting

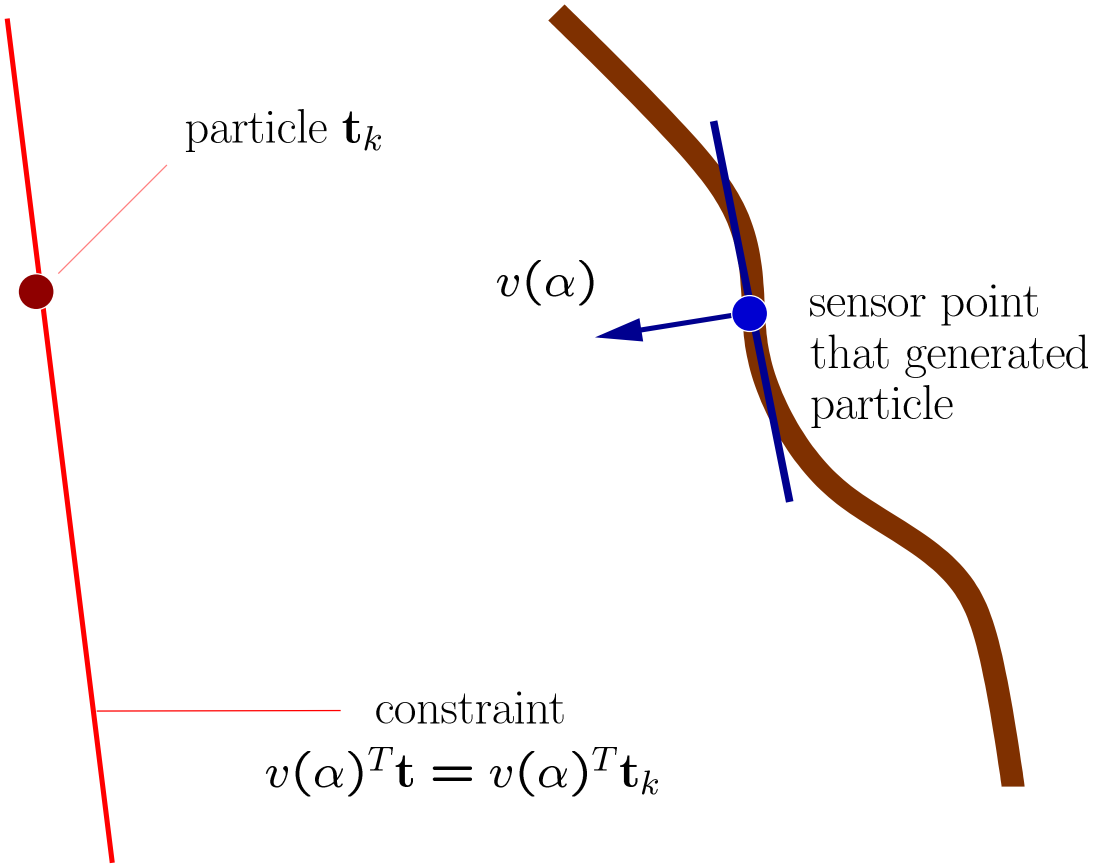


after weighting

$\Rightarrow$ now a particle approximation to $p(\varphi, \mathbf{t} | \mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{z}_t, \mathbf{z}_{t-1})$.

Least squares formulation

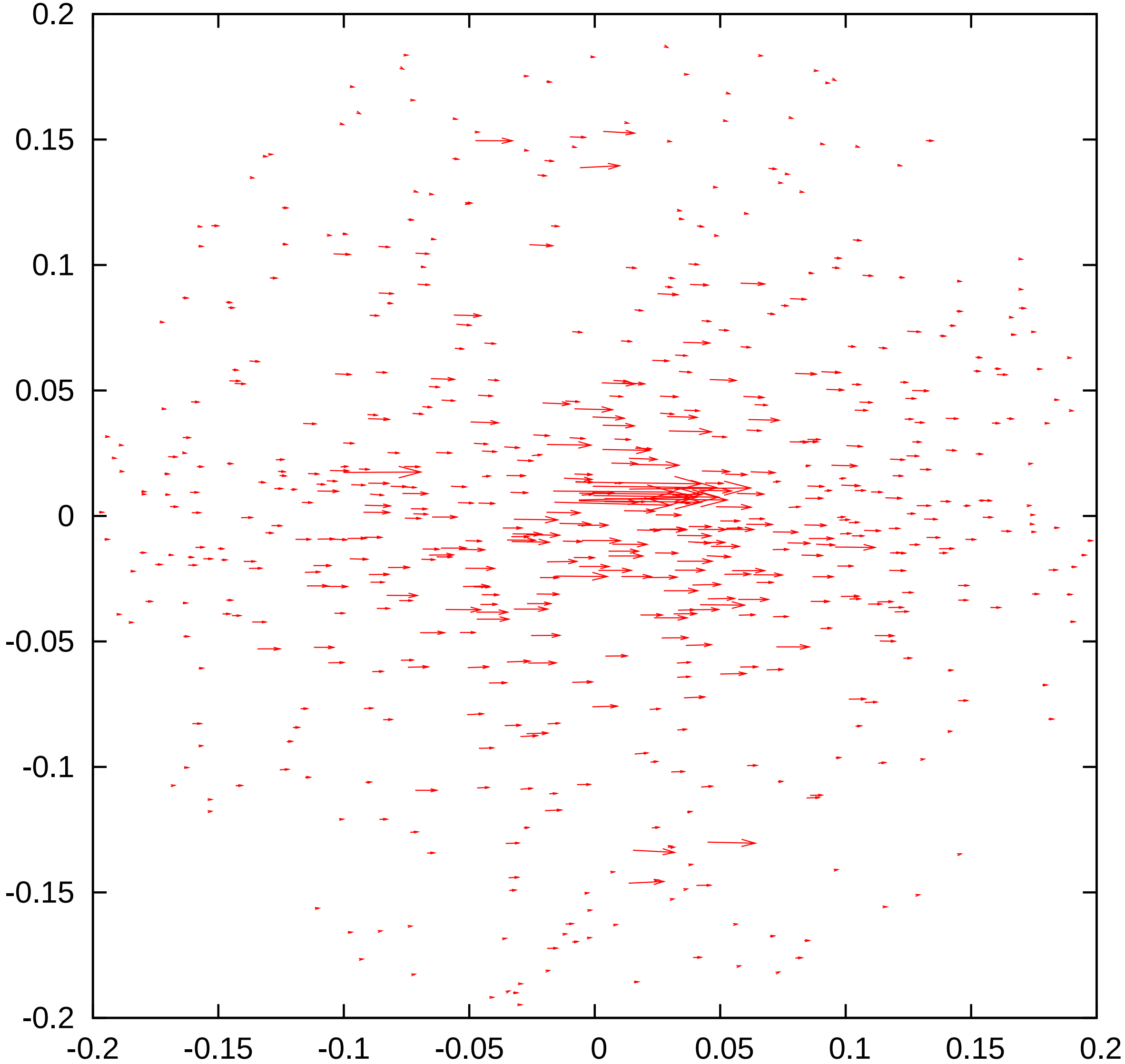
To characterize the uncertainty of the particles, we consider the information useful only along the direction of the wall.



The result is a set of constraints: a least squares problem.

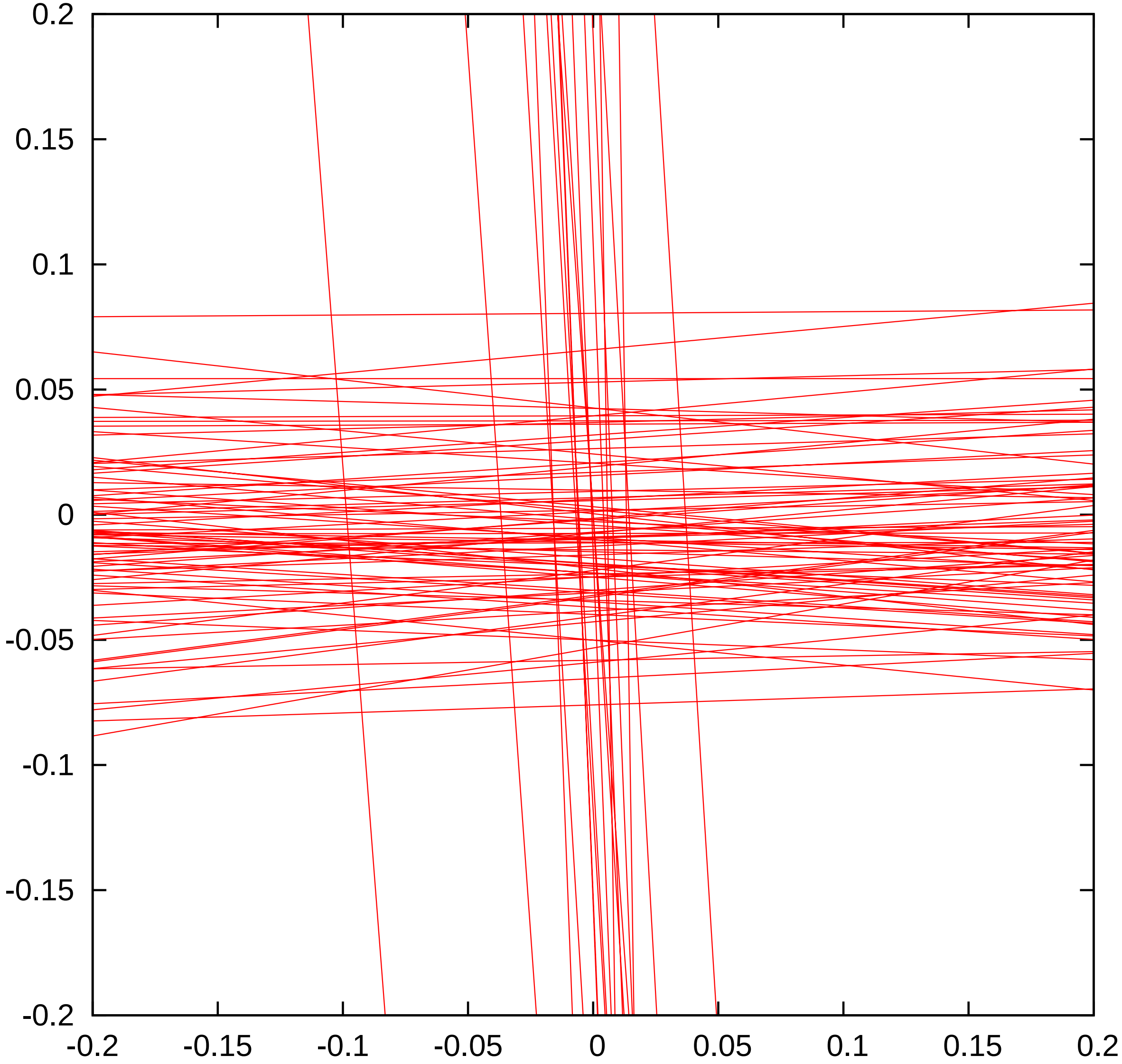
Example (3)

From particles . . .



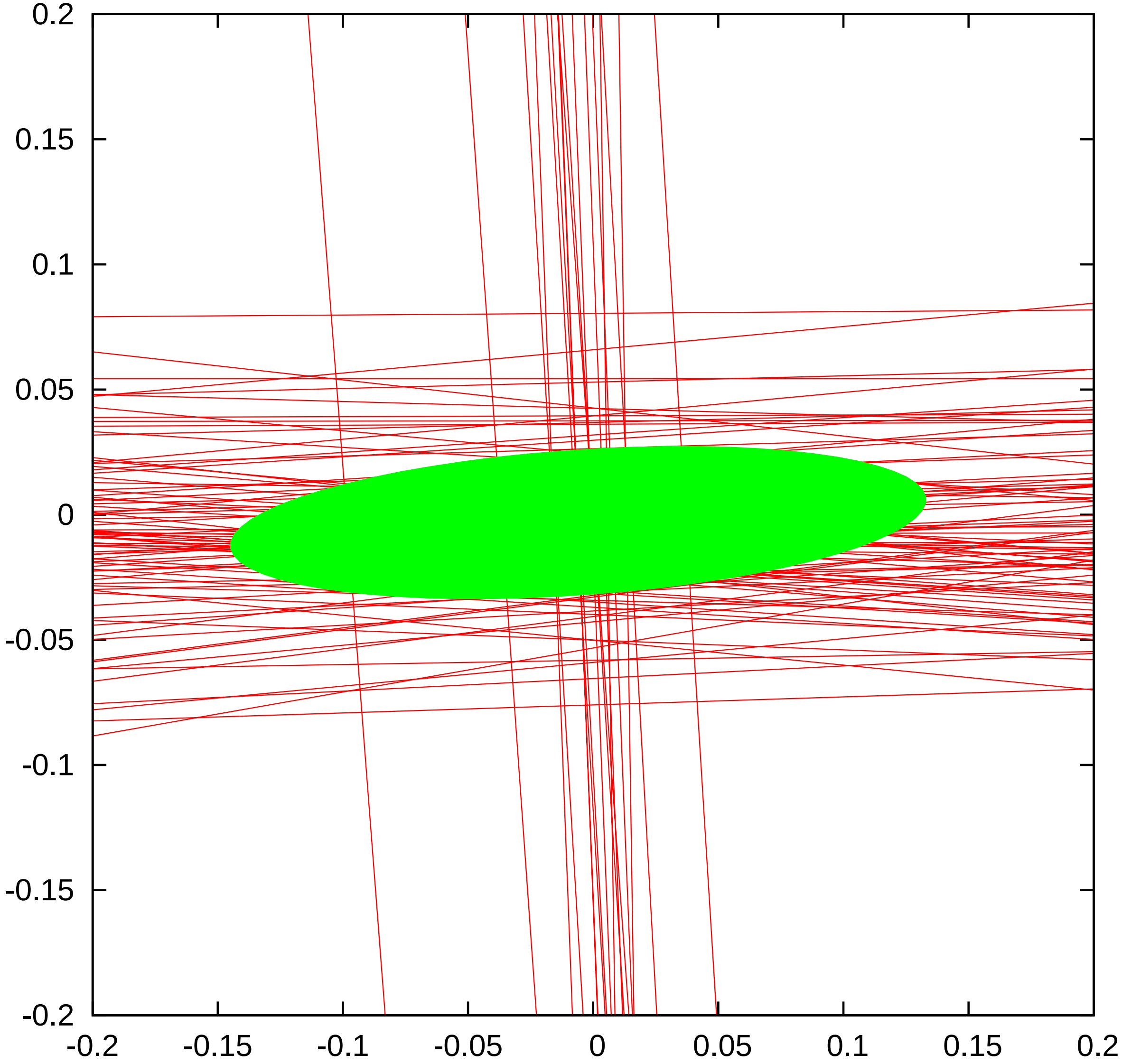
Example (3)

. . . to constraints . . .



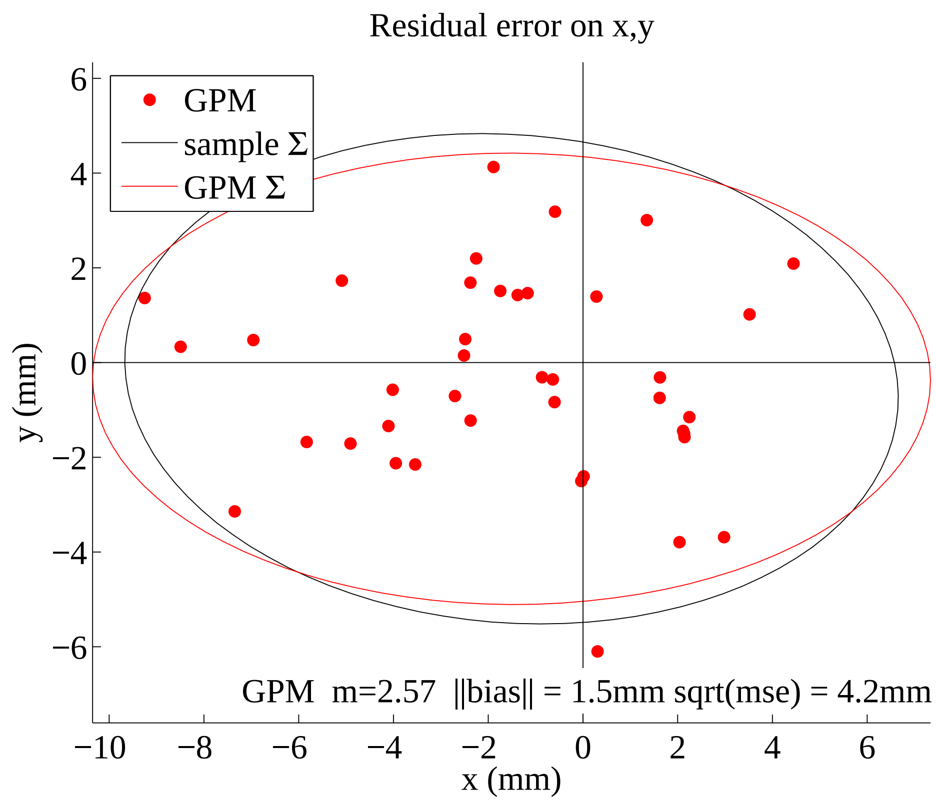
Example (3)

. . . to covariance.



The need for tuning

Experimentally, the estimated covariance is significant only up to a constant (good “shape”, bad “area”).

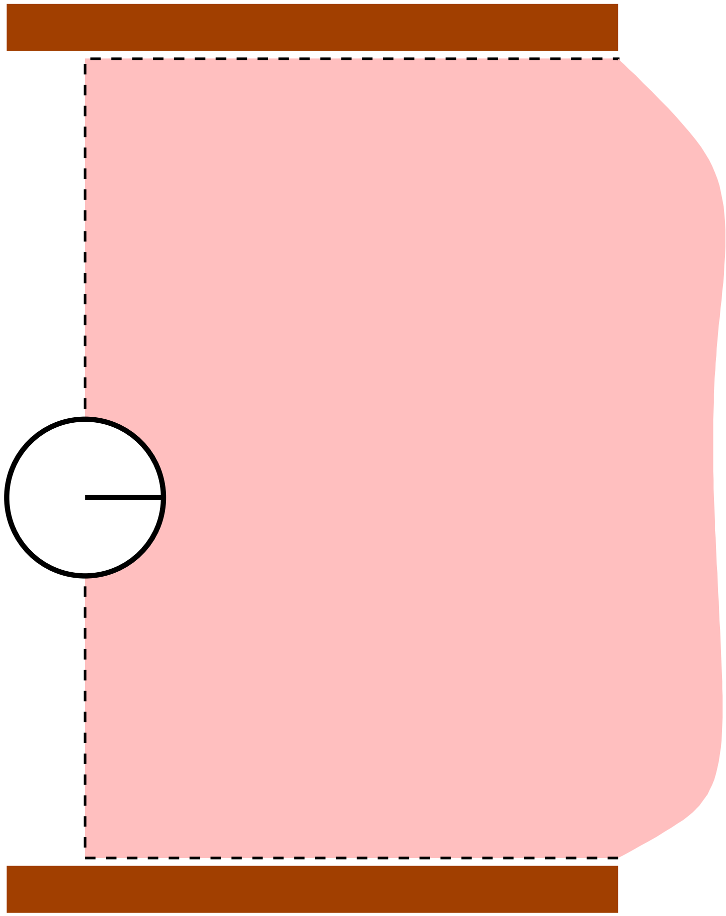


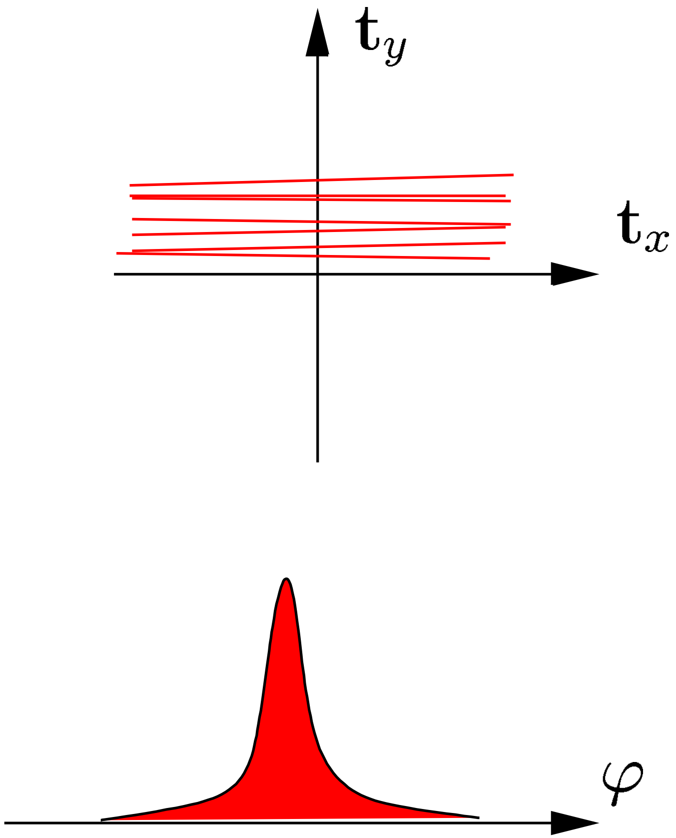
Two reasons for this:

— uncertainty should be modeled better

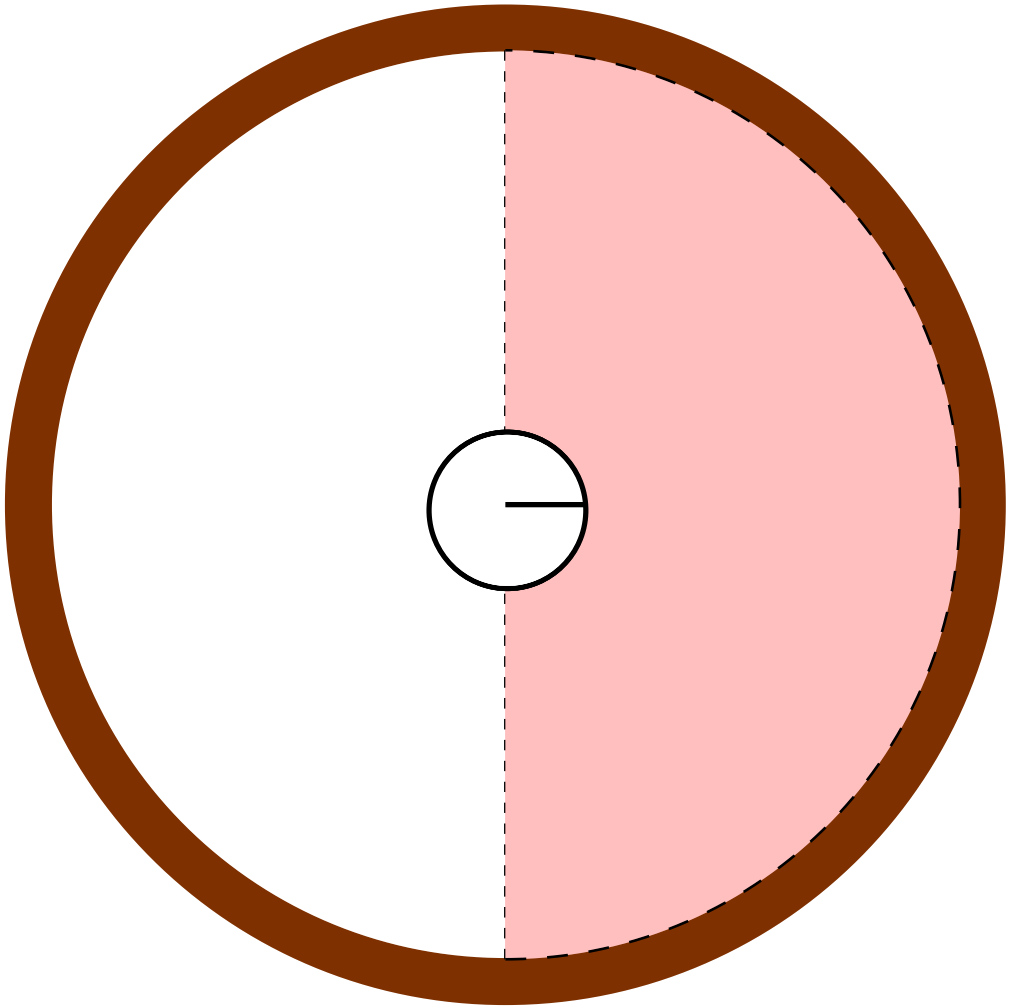
— all particles considered independent (instead, the global covariance matrix is not diagonal)

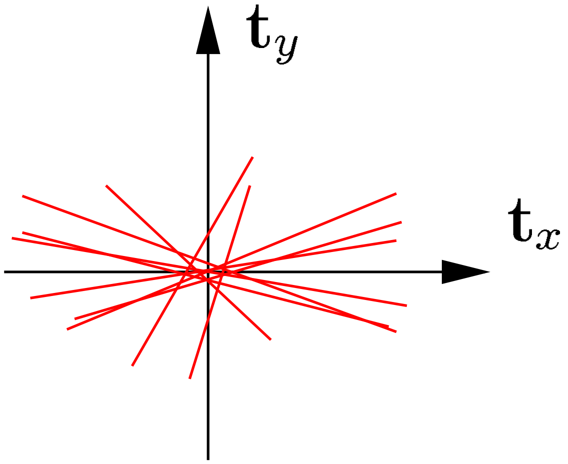
Unconstrained situations

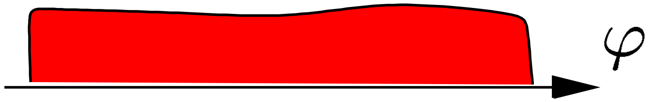




Unconstrained situations

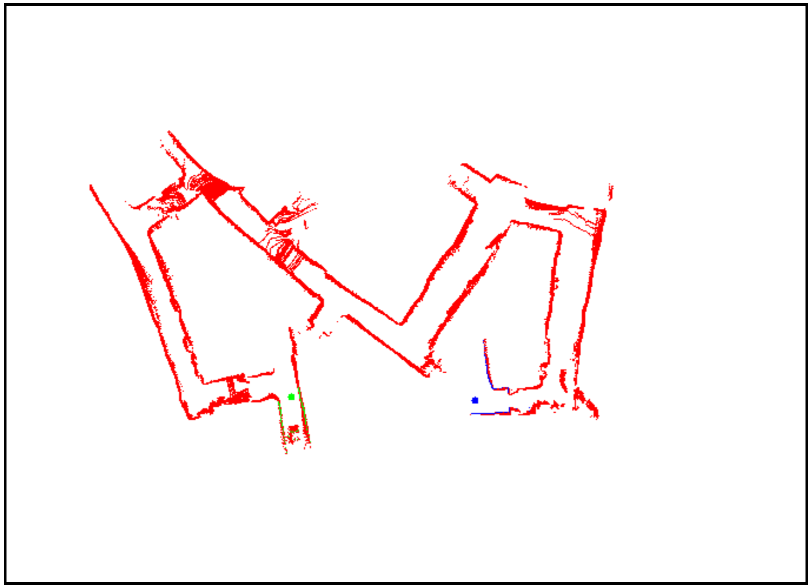




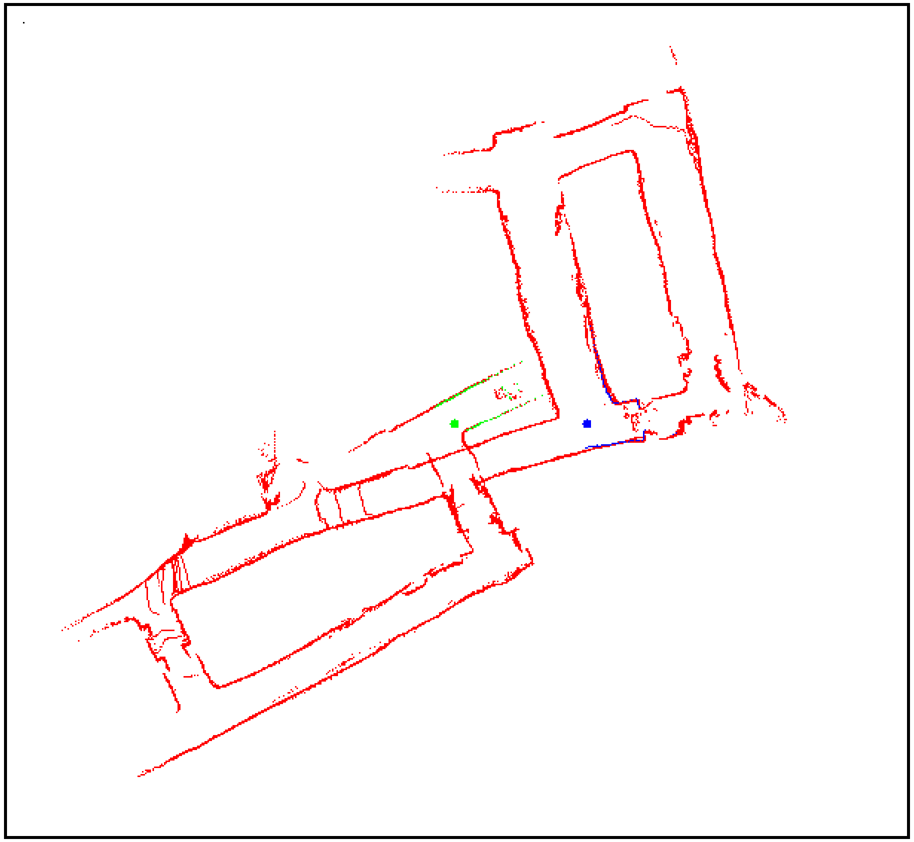


Mine example

A robot in a mine - thanks to Dirk Haehnel and the CMU group for the data files.

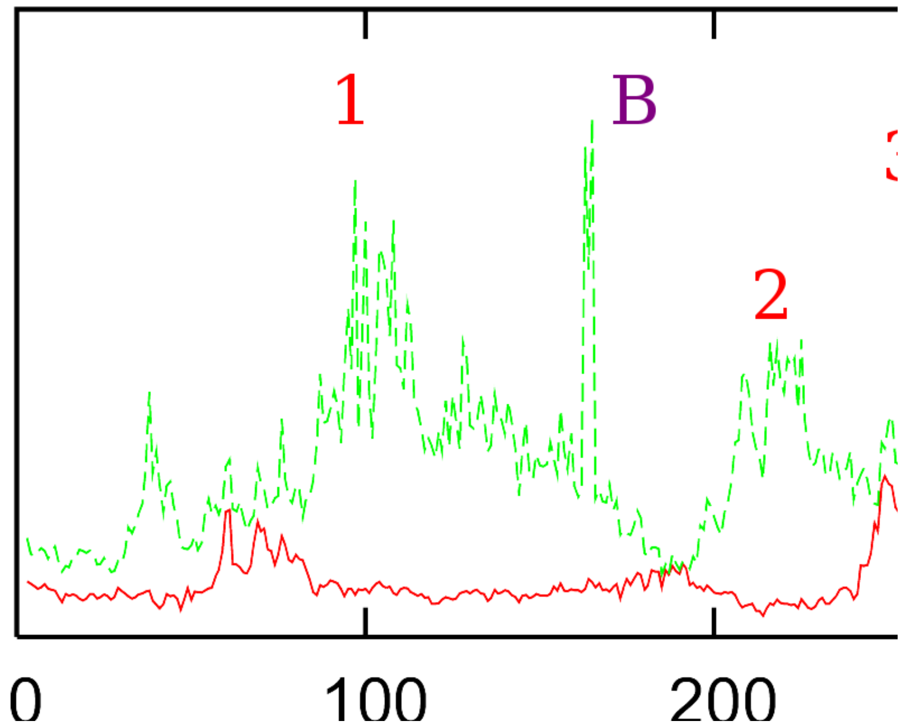


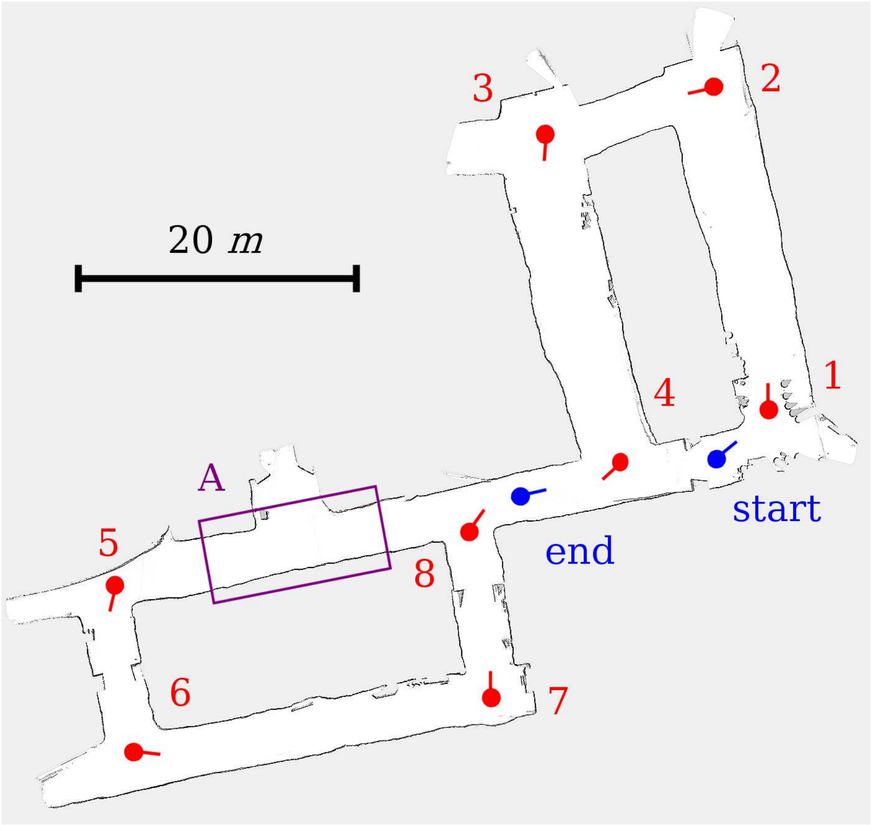
Sensor data



GPM *result*

Eigenvalues of estimated covariance





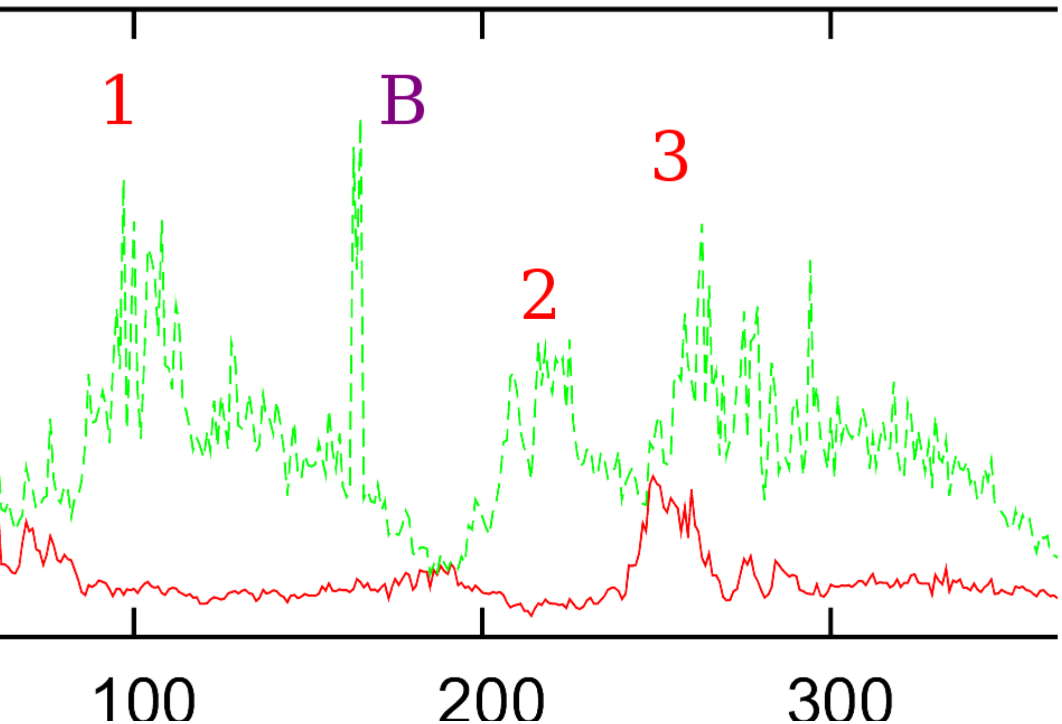
Green and red are the eigenvalues of the estimated covariance matrix.

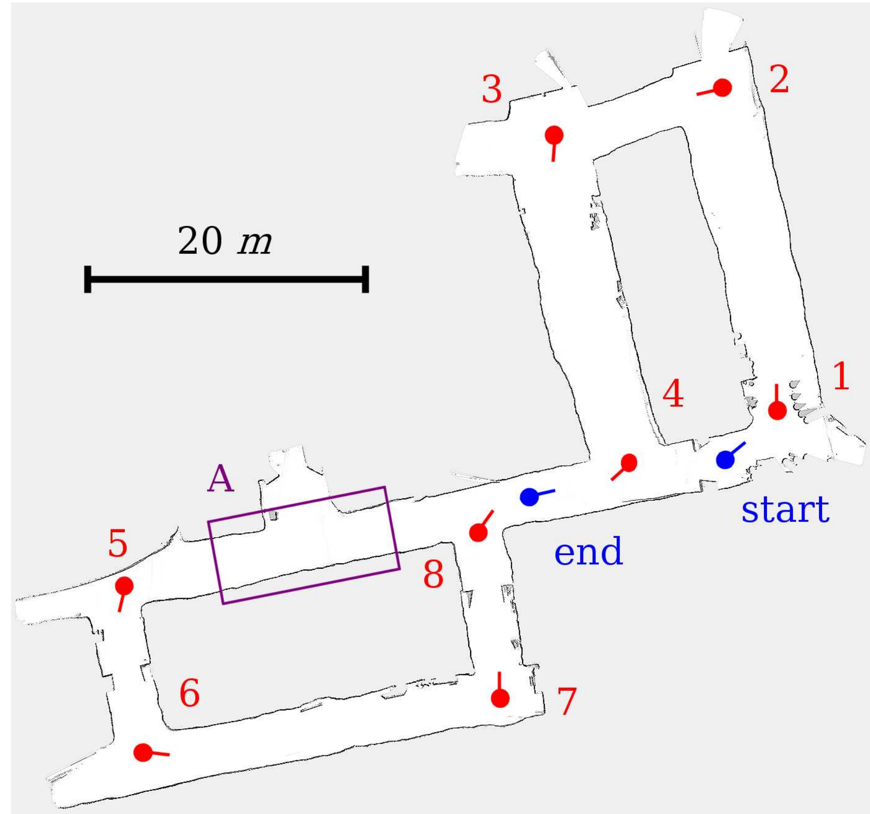
— One eigenvalue is always small (left and right walls).

— The other is big at beginning of corridors (1,2,. . .) then decreases.

— Occlusions are not fatal and detected.

Eigenvalues of estimated covariance





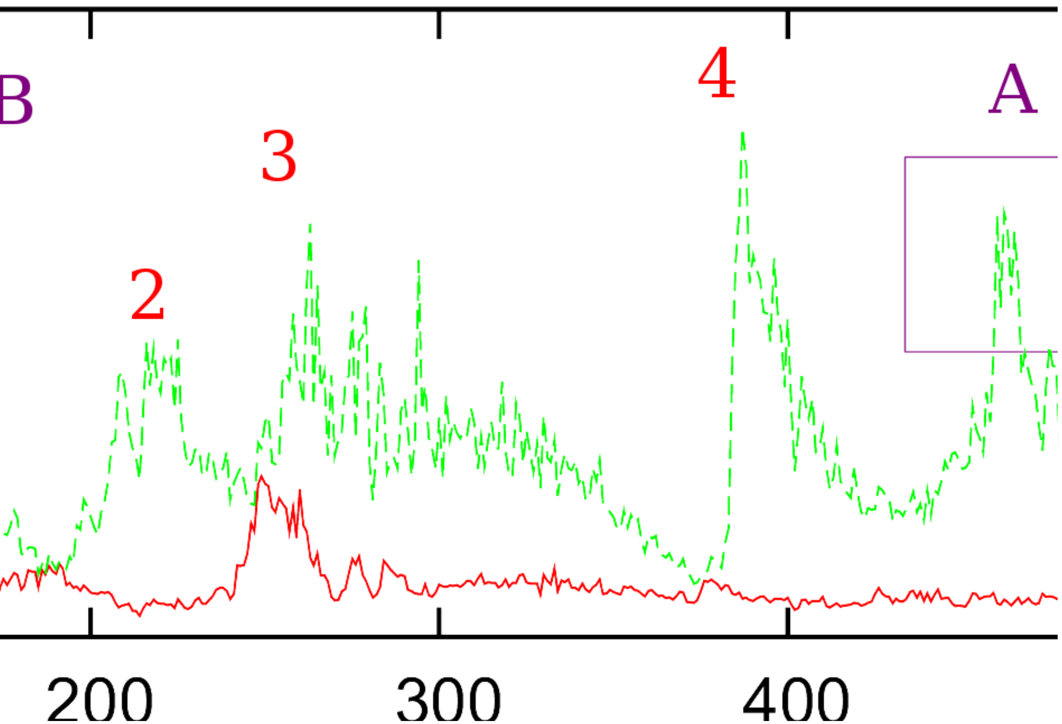
Green and red are the eigenvalues of the estimated covariance matrix.

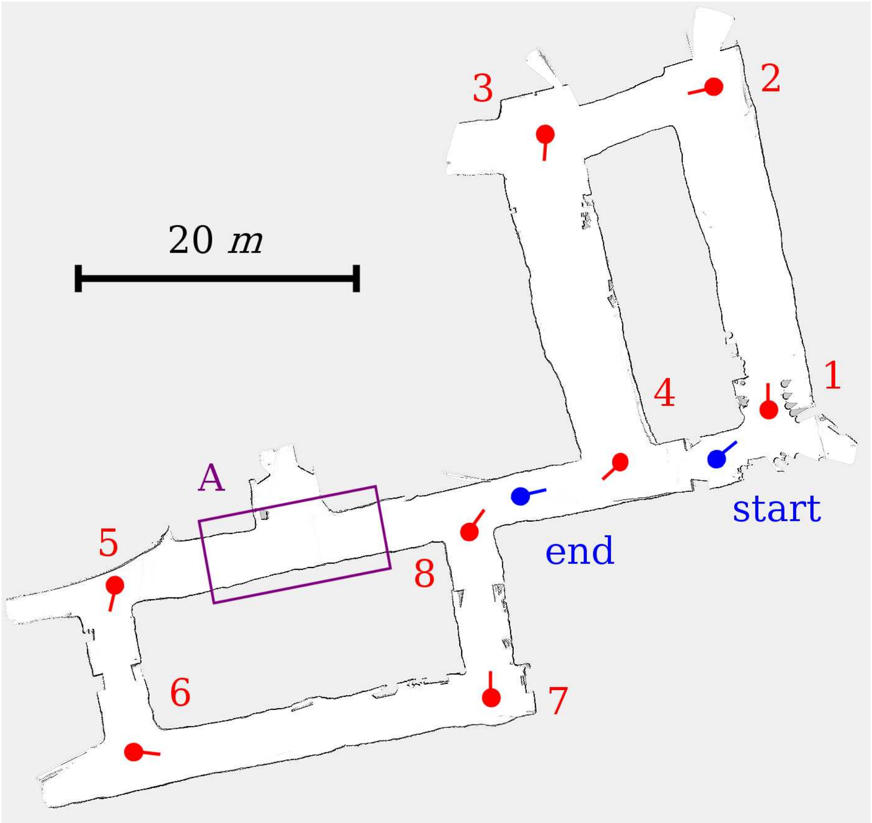
— One eigenvalue is always small (left and right walls).

— The other is big at beginning of corridors (1,2,. . .) then decreases.

— Occlusions are not fatal and detected.

Eigenvalues of estimated covariance





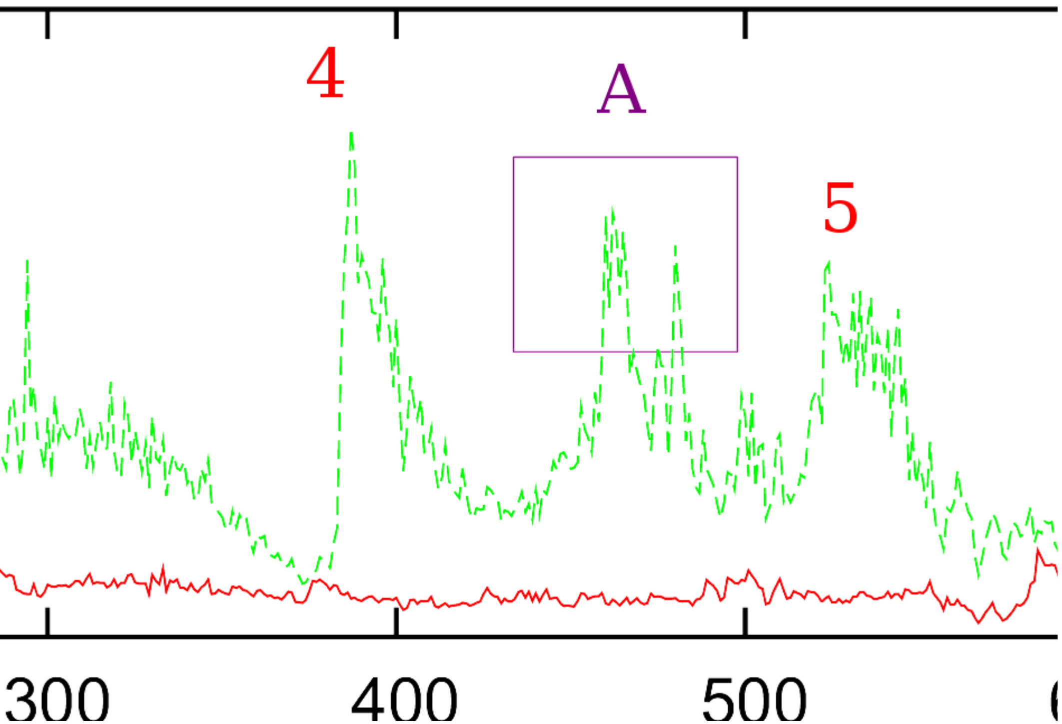
Green and red are the eigenvalues of the estimated covariance matrix.

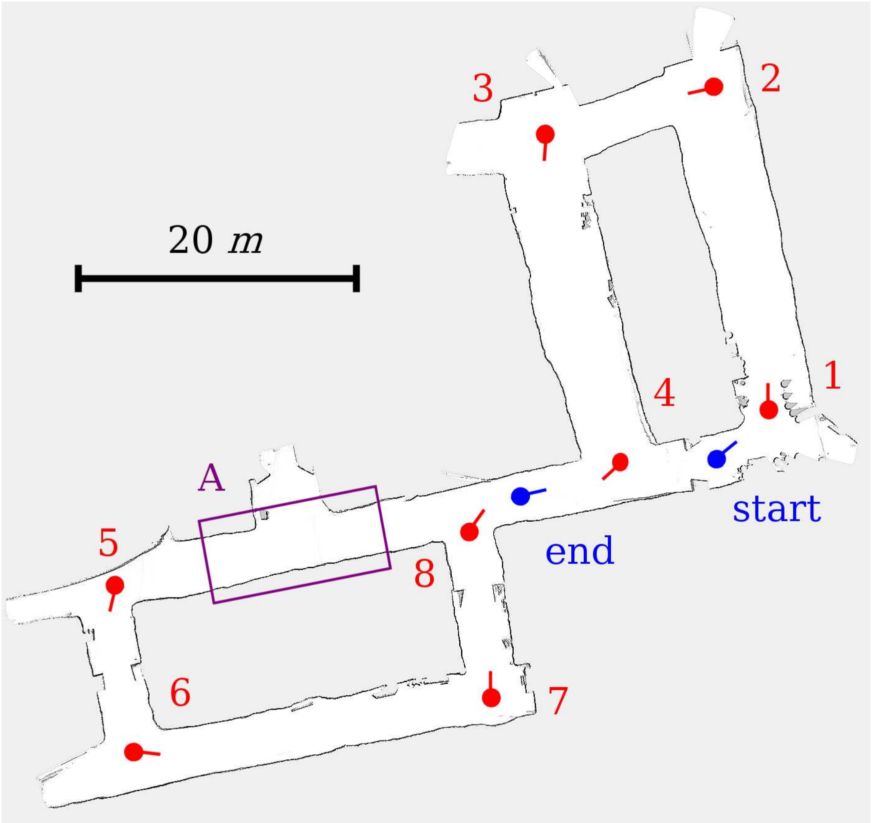
— One eigenvalue is always small (left and right walls).

— The other is big at beginning of corridors (1,2,. . .) then decreases.

— Occlusions are not fatal and detected.

Eigenvalues of estimated covariance





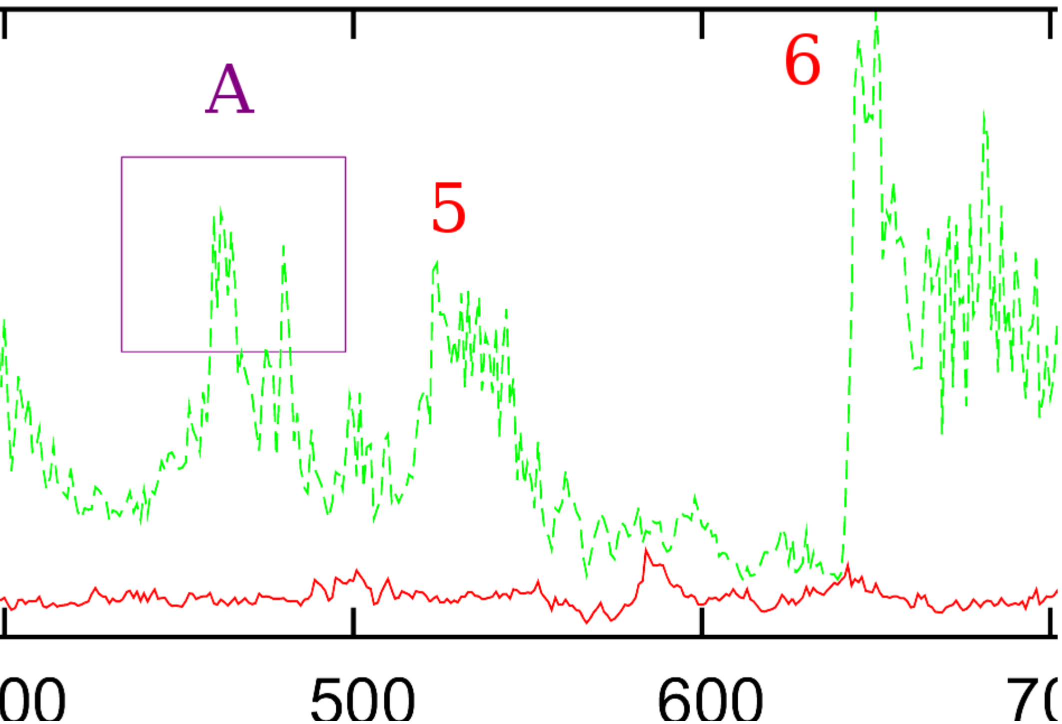
Green and red are the eigenvalues of the estimated covariance matrix.

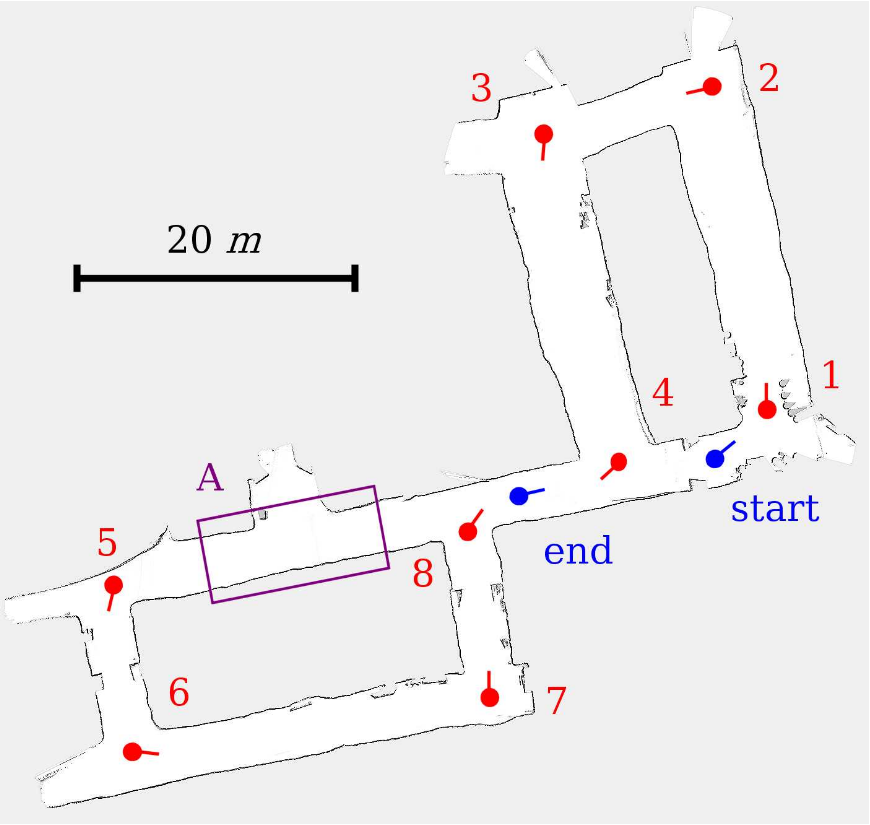
— One eigenvalue is always small (left and right walls).

— The other is big at beginning of corridors (1,2,. . .) then decreases.

— Occlusions are not fatal and detected.

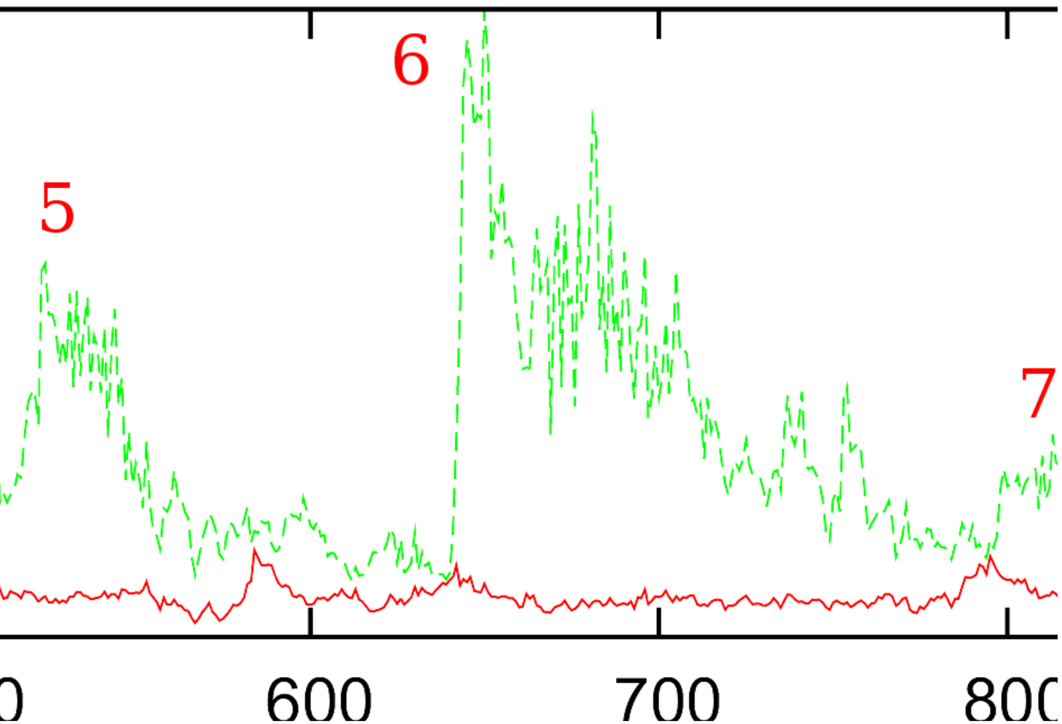
Eigenvalues of estimated covariance

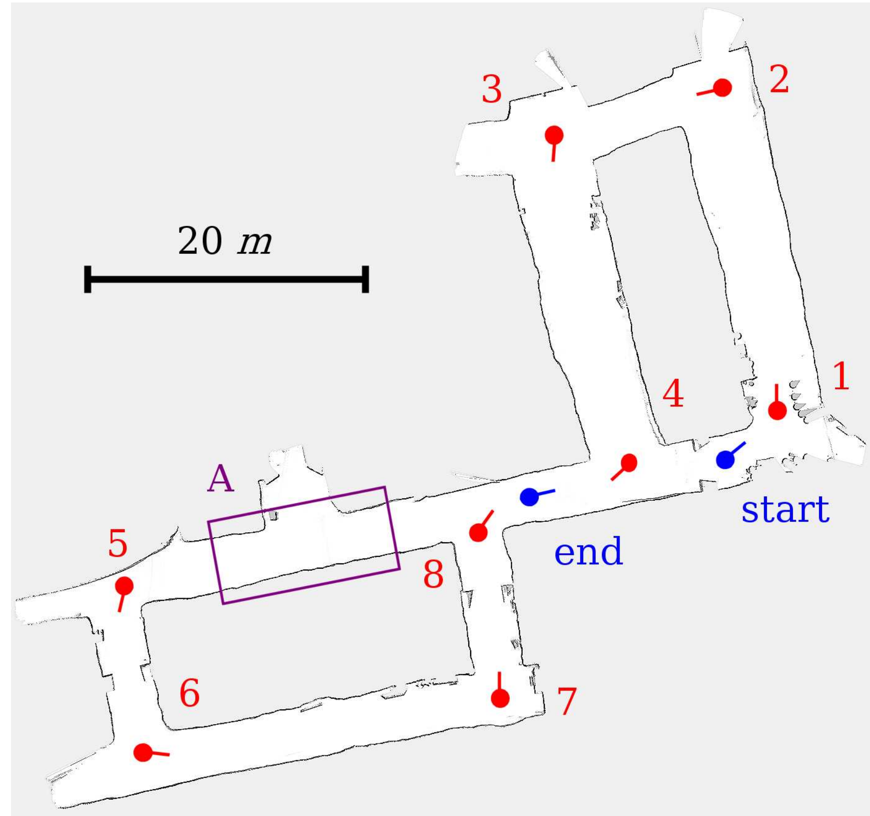




Green and red are the eigenvalues of the estimated covariance matrix.
— One eigenvalue is always small (left and right walls).
— The other is big at beginning of corridors (1,2,. . .) then decreases.
— Occlusions are not fatal and detected.

Eigenvalues of estimated covariance





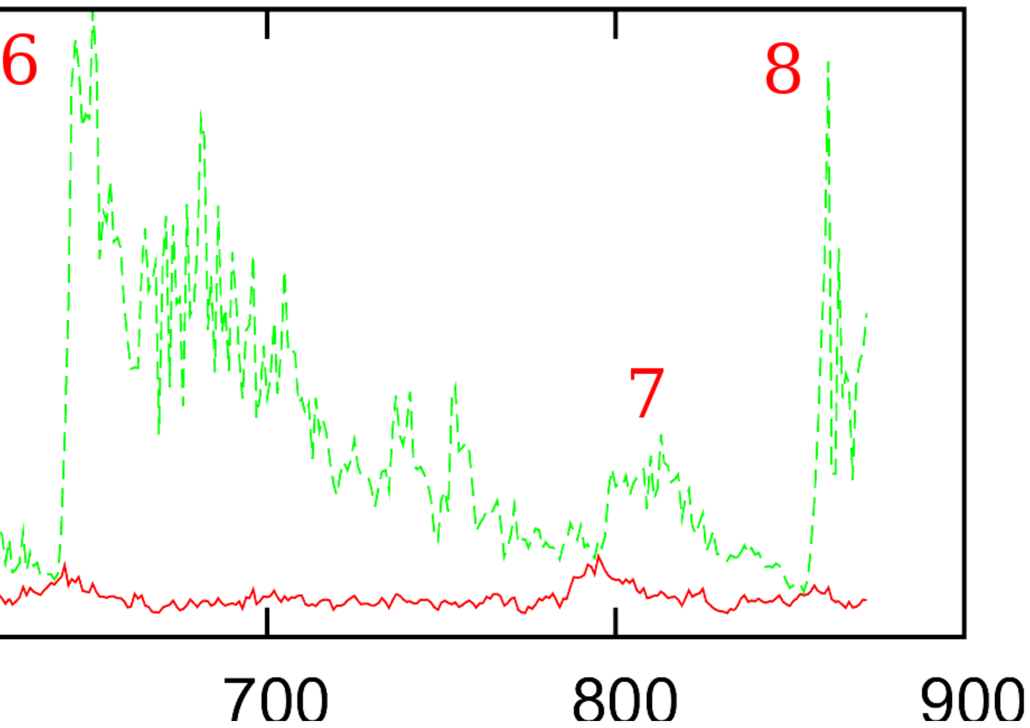
Green and red are the eigenvalues of the estimated covariance matrix.

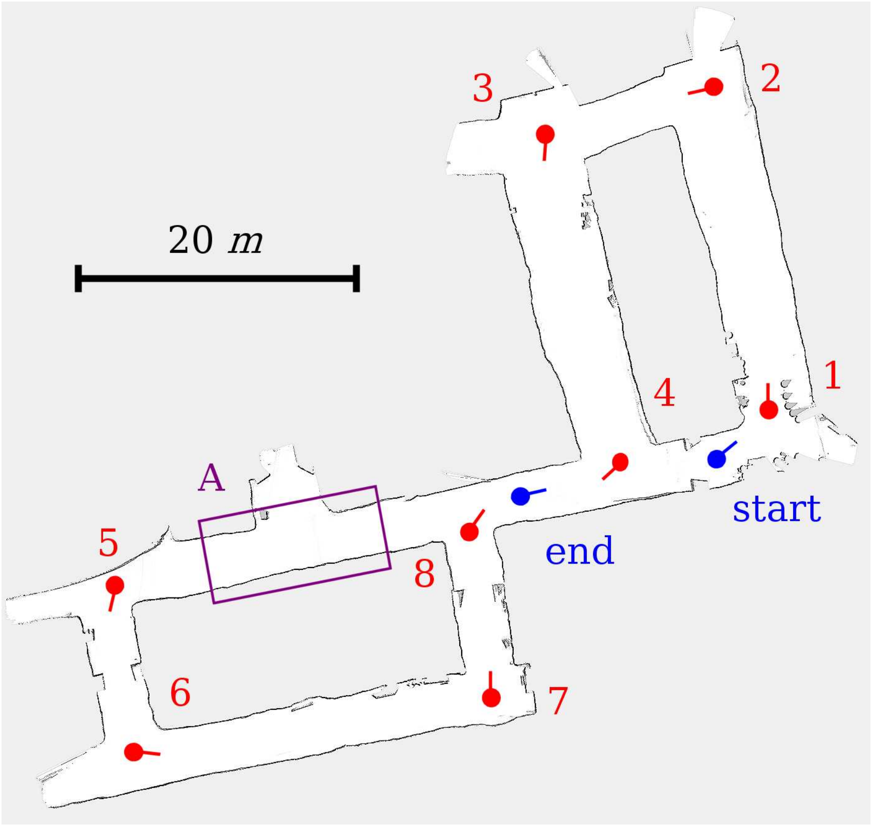
— One eigenvalue is always small (left and right walls).

— The other is big at beginning of corridors (1,2,...) then decreases.

— Occlusions are not fatal and detected.

Eigenvalues of estimated covariance





Green and red are the eigenvalues of the estimated covariance matrix.

— One eigenvalue is always small (left and right walls).

— The other is big at beginning of corridors (1,2,. . .) then decreases.

— Occlusions are not fatal and detected.

Comparison with MbICP, IDC, ICP

Cited from [Minguez&al. IEEE T-RO'06]. Real-world data; each scan is matched against itself; search space is $(0.4m, 0.4m, 90°)$.

errors (m)	MbICP	IDC	ICP	GPM
< 0.001	80.3%	74.9%	52.2%	58.8%
$(0.001, 0.005)$	18.8%	16.5%	42.0%	28.5%
$(0.005, 0.01)$	**0**	0.3%	**0**	7.1%
$(0.01, 0.05)$	**0**	0.8%	0.01%	5.3%
> 0.05	0.7%	7.3%	5.8%	**0**

— GPM does not have very large errors; error for φ is 0 if scans are equal.

— MbICP is more precise when it converges.

— Probably [Pfister&al.'02], [Biber&al.'03] would have results similar to MbICP.

Conclusion and future work

— GPM's strong points:
- uses, soundly, an arbitrary evolution model (also multimodal)
- characterizes the uncertainty analytically, also in underconstrained situations
- not iterative: result does not depend on first guess

— GPM's weak points:
- It is not usable in totally unstructured environments.
- Iterative methods are more precise *when they converge near the right solution*.

— GPM's future work:
- Exploit the multimodality of the particle distribution.
- Try some interpolation schema to compensate for the sparseness of the sensor data.

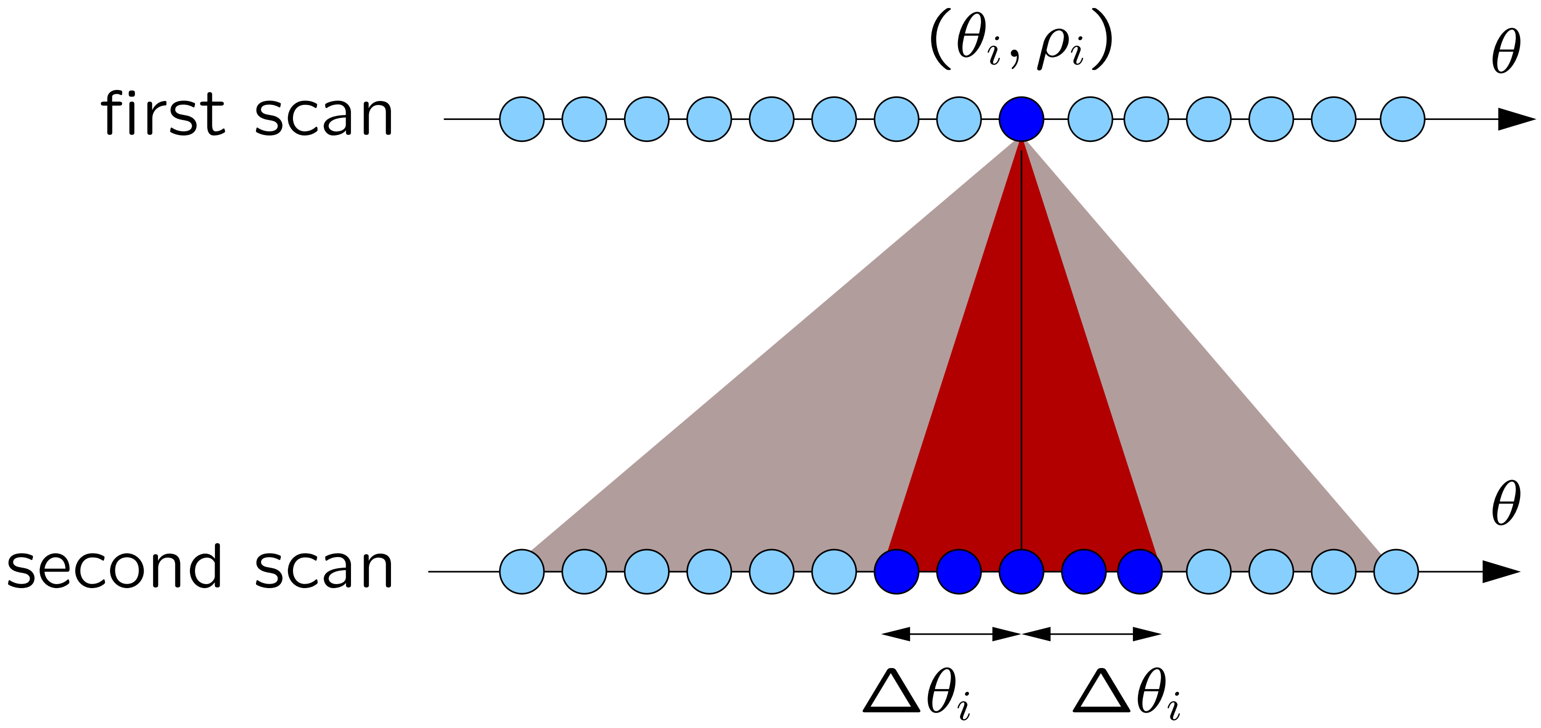
GPM performance

Square environment, $4m \times 4m$. Random sampling of poses, uniform $400mm, 20°$.

	$\lvert\text{bias}_{xy}\rvert$	$\sqrt{\text{MSE}_{xy}}$	$\lvert\text{bias}_{\varphi}\rvert$	$\sqrt{\text{MSE}_{\varphi}}$
360 rays	$0.6mm$	$11.1mm$	$< 0°$	$0.10°$
180 rays	$2.4mm$	$11.4mm$	$0.01°$	$0.13°$
90 rays	$4.5mm$	$27.4mm$	$0.09°$	$0.40°$
45 rays	$12.3mm$	$36.4mm$	$0.08°$	$0.59°$

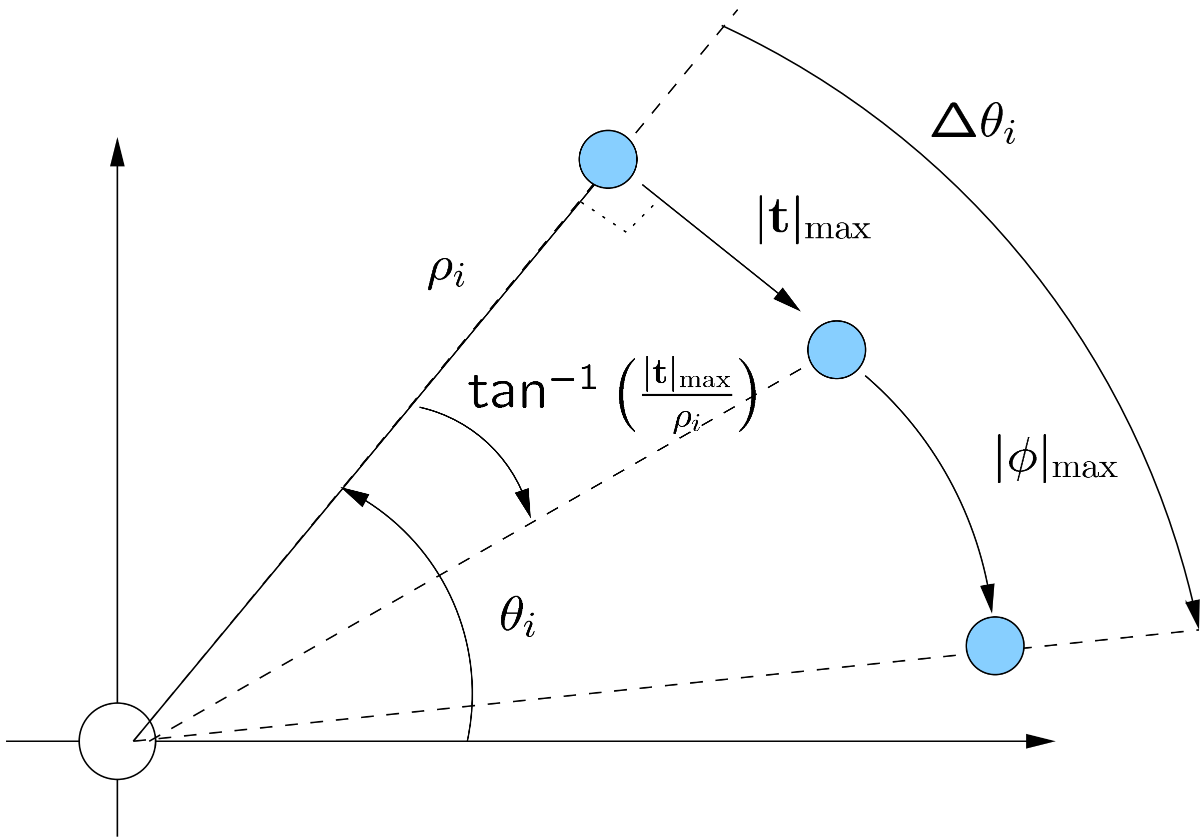
Fast correspondence search

We can make GPM faster by exploiting the radial ordering of the scans and searching for a bound for $\Delta\theta$.



Fast correspondence search

Intuitively, the maximum variation occurs when the point is (in either order) translated by $|\mathbf{t}|_{\text{max}}$ perpendicular to p_i, then rotated by $|\varphi|_{\text{max}}$.



Therefore

$$\Delta\theta_i = \tan^{-1}\left(\frac{|\varphi|_{\text{max}}}{\rho_i}\right) + |\varphi|_{\text{max}}$$

LSE formulation

We derived

$$p_j = R_\varphi p_i + \mathbf{t} \qquad \Rightarrow \qquad \hat{\mathbf{t}} = p_j - R_{\hat{\varphi}} p_i$$

To consider the information only along direction $\alpha_k = \alpha_i$ multiply both sides by the versor $(\cos\alpha_k \sin\alpha_k)$ which we abbreviate as $v(\alpha_k)$.

$$v(\alpha_k)^t \hat{\mathbf{t}} = v(\alpha_k)^t (p_j - R_{\hat{\varphi}} p_i) := y_k$$

Now the set of hypotheses is a set of constraints:

$$v(\alpha_k)^t \mathbf{t} = y_k + m/w_k \cdot \epsilon$$

where m is a tuning constant.

LSE

$$L\mathbf{t} = Y + R \cdot \epsilon$$

$$\begin{aligned} L &= \big(v(\alpha_1) \cdots v(\alpha_k) \cdots v(\alpha_K)\big)^t \\ Y &= \begin{pmatrix} y_1 & \ldots & y_k & \ldots & y_K \end{pmatrix}^t \\ R &= m \cdot \mathsf{diag}\{1/w_1, \ldots, 1/w_k \ldots 1/w_K\} \end{aligned}$$

Beware of the assumptions that will lead to a diagonal noise covariance matrix:

— each constraint is independent (instead, more than one constraint are generated by the same reading)

— the w_k do not have a probabilistic interpretation

LSE solution

The LSE solution is

$$\bar{\mathbf{t}} = (L^t R^{-1} L)^{-1} L^t R^{-1} Y$$

We must invert:
— the R covariance matrix (assumed diagonal)
— a 2×2 matrix $C = (L^t R^{-1} L)^{-1}$ (invertible if L is full rank).

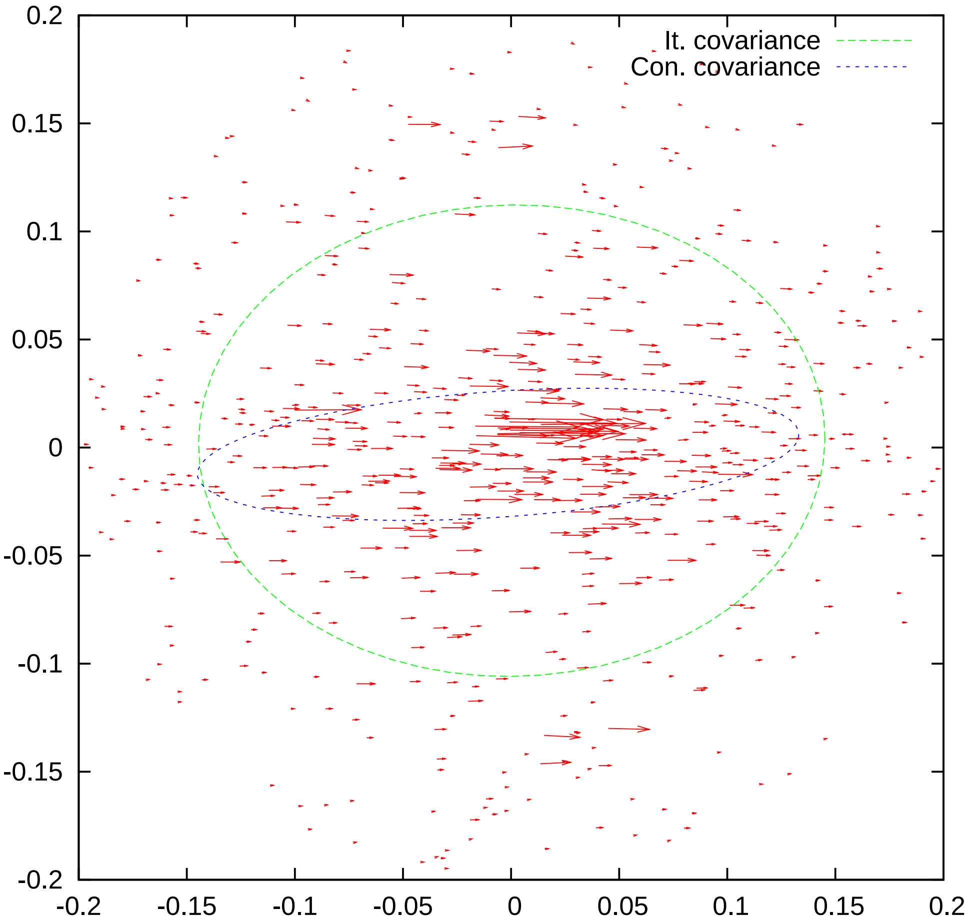
The solution is

$$\begin{aligned} C &= m(\sum_k [w_k v(\alpha_k) v(\alpha_k)^t])^{-1} \\ \bar{\mathbf{t}} &= \left(\sum_k [w_k v(\alpha_k) v(\alpha_k)^t] \right)^{-1} \sum_k [w_k y_k v(\alpha_k)] \end{aligned}$$

The choice of a tuning constant m does not bias the estimate of φ.

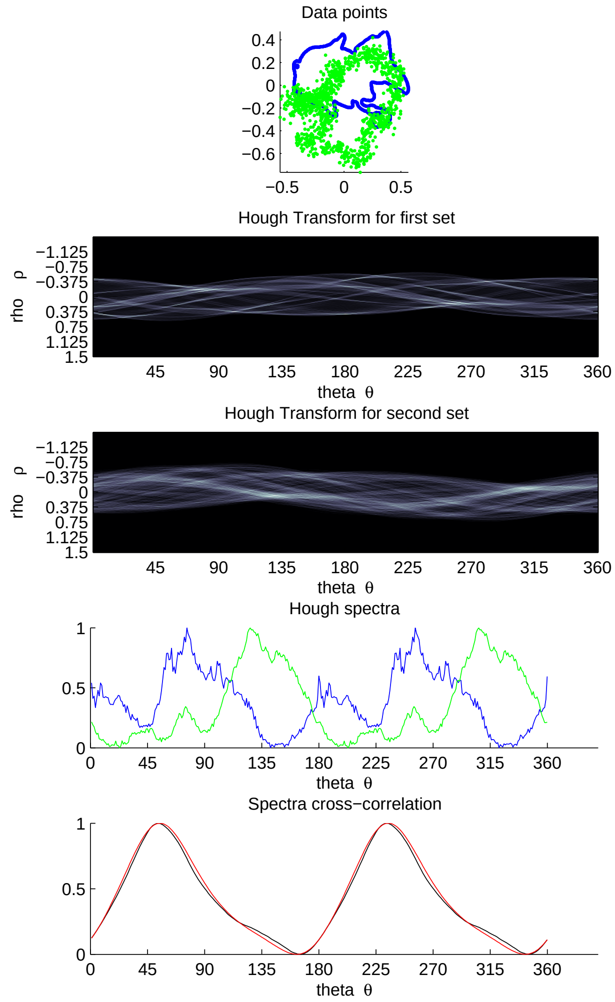
Improved covariance

The covariance represents the uncertainty better.



GPM vs HSM

HSM is a scan matcher presented by Censi, Grisetti, Iocchi at ICRA'05 (paper and source code on my website).



HSM's pros:

- HSM does global searches.
- HSM is correct and complete for exact input.
- HSM does not need orientation information.

HSM's cons:

- HSM uses a cross-correlation operator: time is quadratic in resolution.
- HSM does not characterize uncertainty.