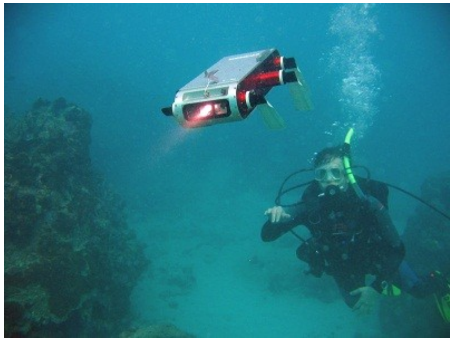
Filtrage particulaire dans la vision robotique :
Application à la commande d'un mini-drone

Céline Teulière

École Thématique Filtrage Particulaire
9 février 2012

Mobile robotics

CyCab (Inria)



Aqua2 (Independent Robotics)

Quadri-rotor (Novadem)

HRP-4 (Kawada)

A task is defined in complex and changing environments

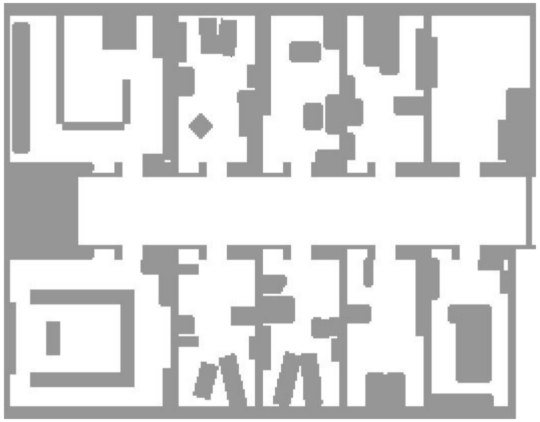
Perception from sensors (camera, IMU, GPS, Lidar...)

Action: motion

Mobile robot localization

Objective: Estimating a robot's pose relative to a map of its environment

- Position tracking
- Global localization problem
- Kidnapped robot problem
- Multi-robot localization problem



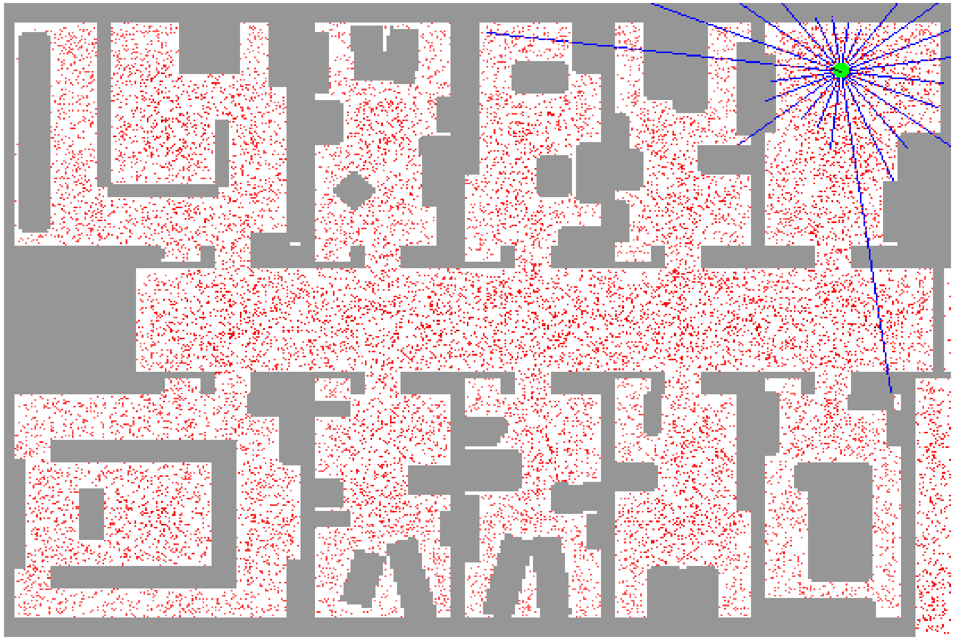
Particle filters provide interesting solutions to these problems

Localization with particle filtering: example

State: 2D position

No initial guess -> prior: uniform sampling in the free space

Measurements: proxymeters



Fox – Dellaert 99

Problem formulation

Estimation of the state $\mathbf{x}_k$

given the observations $\mathbf{z}_{1:k} = (\mathbf{z}_1, .., \mathbf{z}_k)$ up to time step k

Deterministic framework:

- Optimization of ad-hoc objective function $\hat{\mathbf{x}}_k = arg \min_{\mathbf{x}} E(\mathbf{x}_k; \hat{\mathbf{x}}_{k-1}, \mathbf{z}_k)$
- Or iterative minimization of function $E(\mathbf{x}_k; \mathbf{z}_k)$ initialized at $\hat{\mathbf{x}}_{k-1}$

Probabilistic framework:

- Sequential MAP estimate $\hat{\mathbf{x}}_k = arg \max_{\mathbf{x}} p(\mathbf{x}_k \mid \hat{\mathbf{x}}_{k-1}, \mathbf{z}_k)$
- Estimation of the probability density function (pdf) $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$

  output: $\hat{\mathbf{x}}_k = arg \max_{\mathbf{x}} p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ or $\mathbb{E}[\mathbf{x}_k \mid \mathbf{z}_{1:k}]$

Bayesian recursive filter

- Bayes rule for a Markov process:

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})}$$

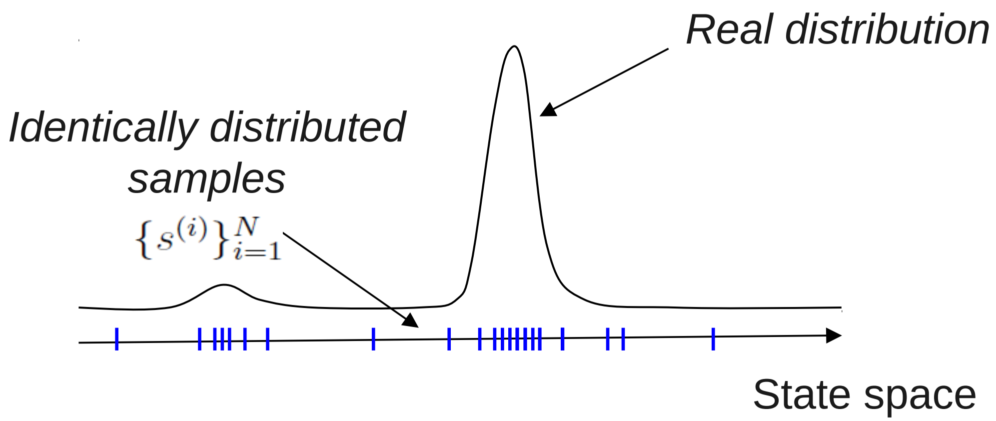
$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \propto p(\mathbf{z}_k \mid \mathbf{x}_k) \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}$$

- Bayesian recursive filter:

$$p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) \stackrel{prediction}{\longrightarrow} p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) \stackrel{correction}{\longrightarrow} p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$$

- Different possible approximations:
  - Linear, Gaussian -> Kalman ;   linearization-> EKF
  - Numerical approximation (discretization of the state space)
  - Particle approximation (sampling) -> particle filters

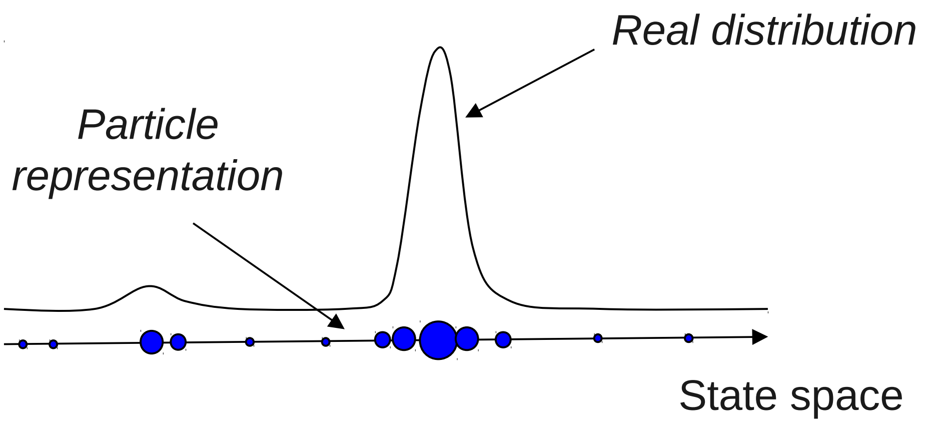
Particle approximation



$$p^N(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{s^{(i)}}(\mathbf{x})$$

$$\frac{1}{N}\sum_{i=1}^{N}\phi(s^{(i)}) \xrightarrow[N\to\infty]{} E_p[\phi(\mathbf{x})] = \int \phi(\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

Particle approximation

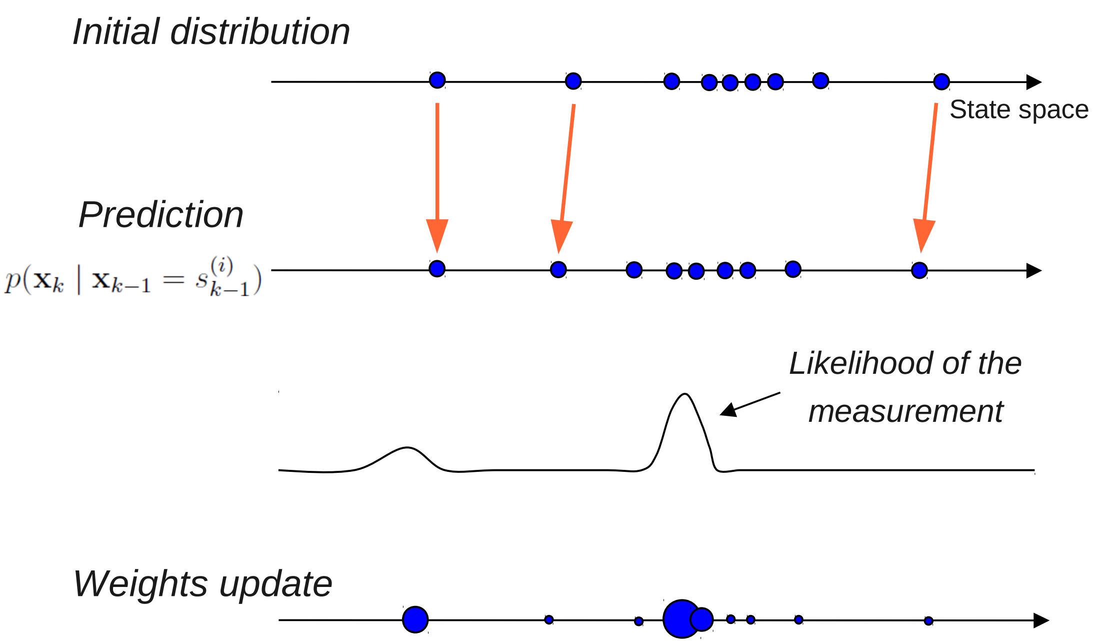


$$p^N(\mathbf{x}) = \sum_{i=1}^{N} \pi^{(i)} \delta_{s^{(i)}}(\mathbf{x})$$

$$\textstyle\sum_{i=1}^{N} \pi^{(i)} = 1$$

$$E_p[\phi(\mathbf{x})] = \int \phi(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \approx \int \phi(\mathbf{x}) p^N(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{N} \pi^{(i)} \phi(s^{(i)})$$

Recursive filter steps



$$p^N(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) = \sum_{i=1}^{N} \pi_{k-1}^{(i)} \delta_{s_{k-1}^{(i)}}(\mathbf{x}_{k-1})$$

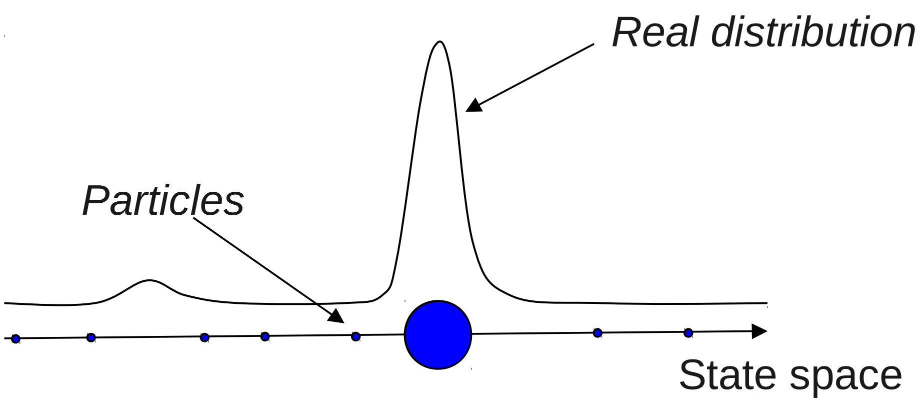
$$p^N(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) = \sum_{i=1}^{N} \pi_{k-1}^{(i)} \delta_{s_k^{(i)}}(\mathbf{x}_k)$$

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} \pi_k^{(i)} \delta_{s_k^{(i)}}(\mathbf{x}_k)$$

$$\pi_k^{(i)} = \frac{\pi_{k-1}^{(i)} p(\mathbf{z}_k \mid \mathbf{x}_k = s_k^{(i)})}{\sum_{i=1}^{N} \pi_{k-1}^{(i)} p(\mathbf{z}_k \mid \mathbf{x}_k = s_k^{(i)})}$$

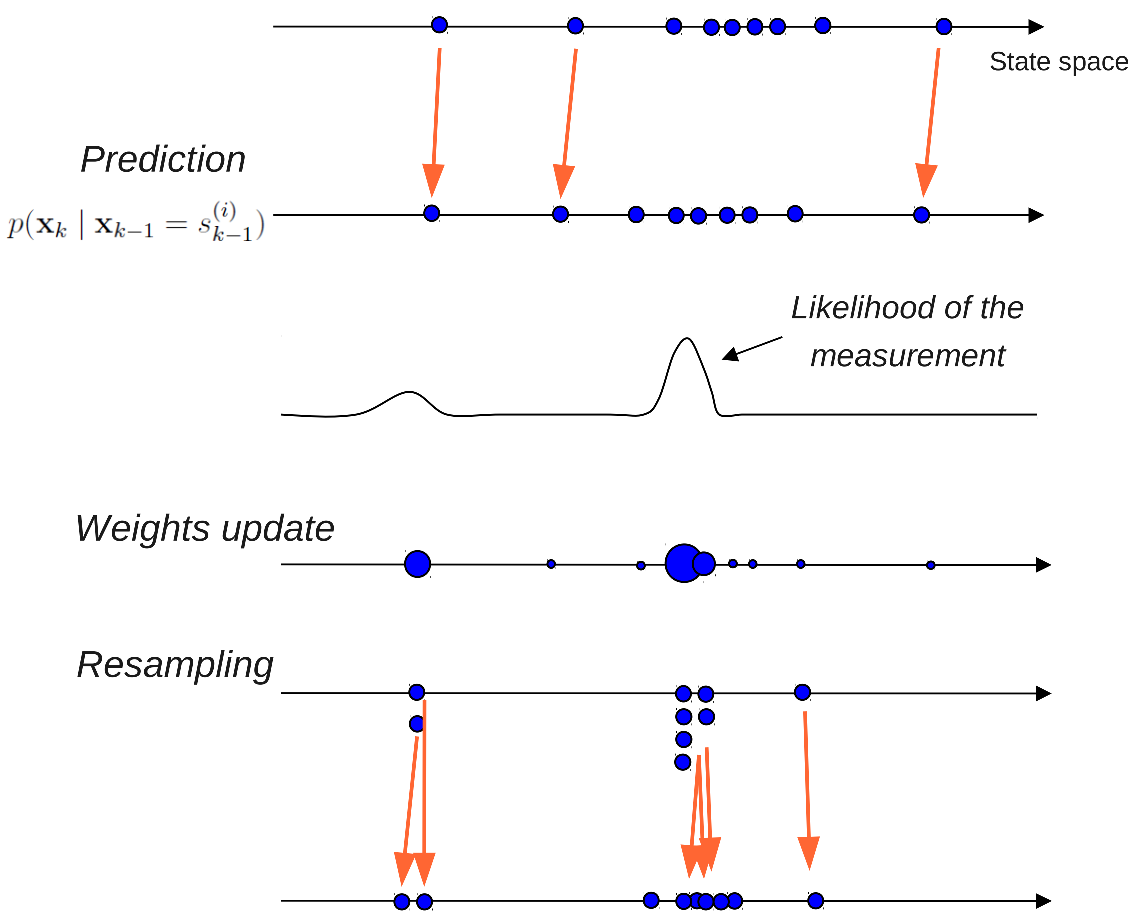
Degeneracy issue

- This first algorithm tends to a situation where a single particle has a strong weight and the other have a negligible weight.



$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (\pi^{(i)})^2}$$ measures the degeneracy of the particle set:
if all the weight are equal, the coefficient is the number of particles

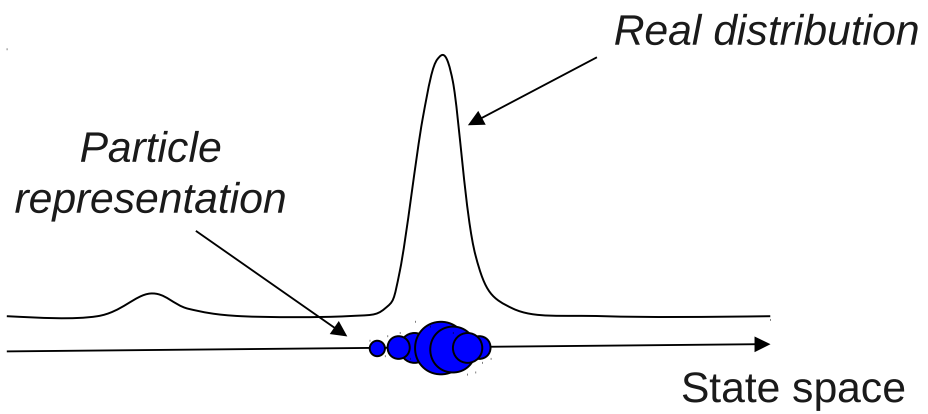
Sampling Importance Resampling (SIR) steps



$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N} \pi_k^{(i)} \delta_{s_k^{(i)}}(\mathbf{x}_k)$$

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (\pi^{(i)})^2}$$

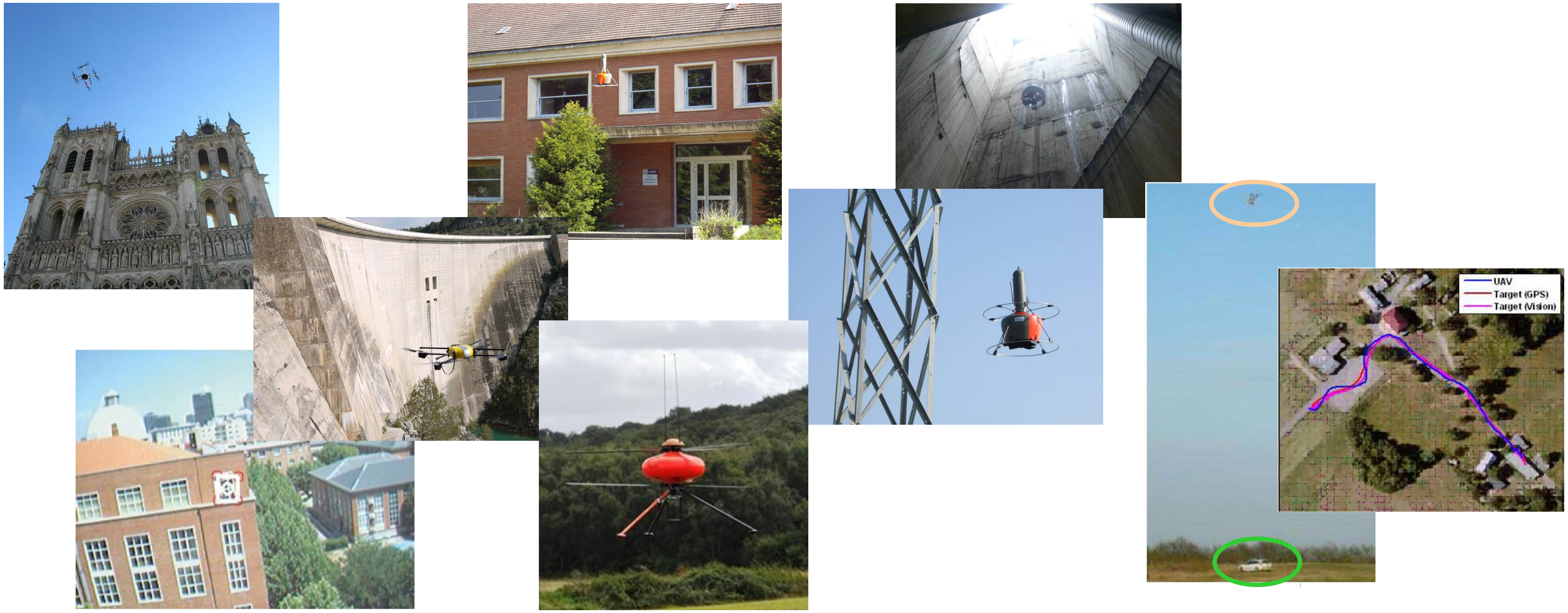
Sample impoverishment issue



- Need to have a good representation of the regions of interest without losing to much information on the space

Examples of particle filter applications: UAV tracking and localization

Unmanned aerial vehicles (UAV)

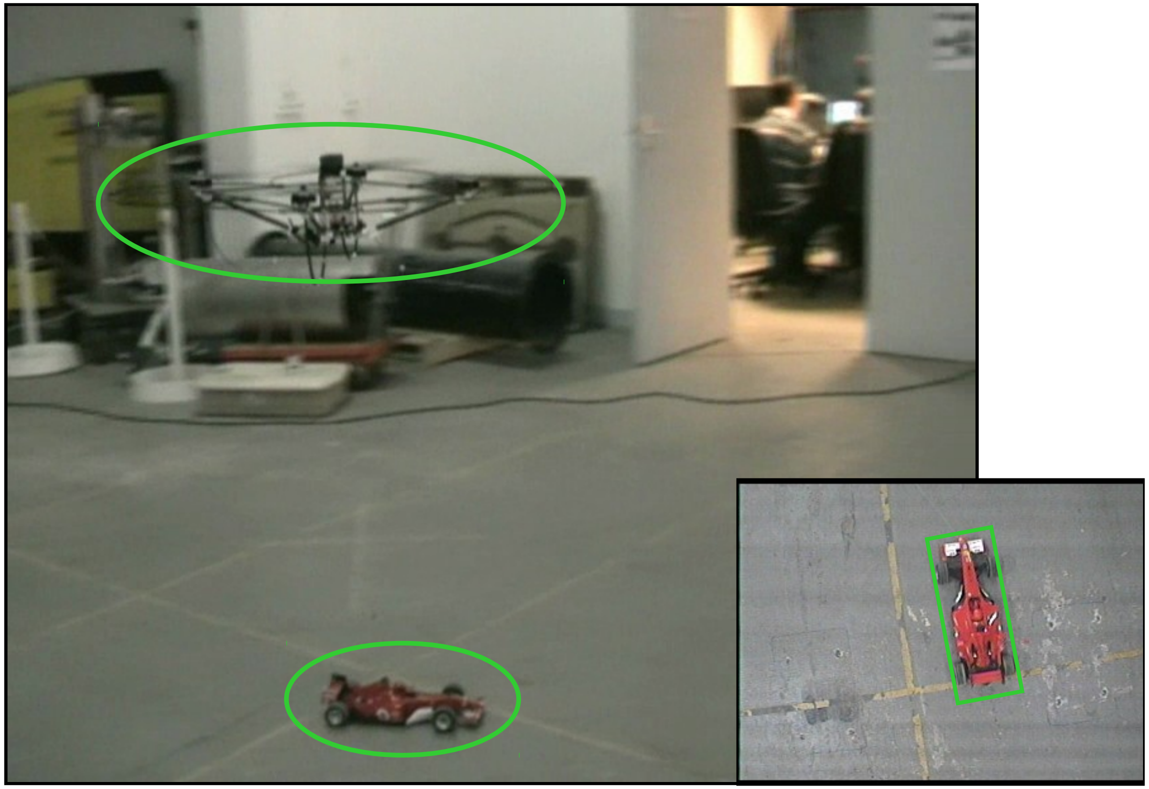


Applications?

- Surveillance, inspection: Fire, combat zone, bridges, plants, search and rescue,...

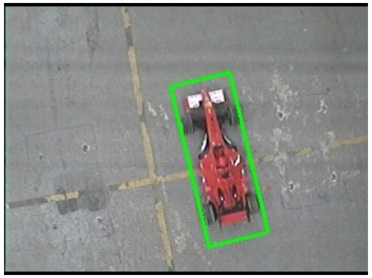
- Different prior knowledge about the environment

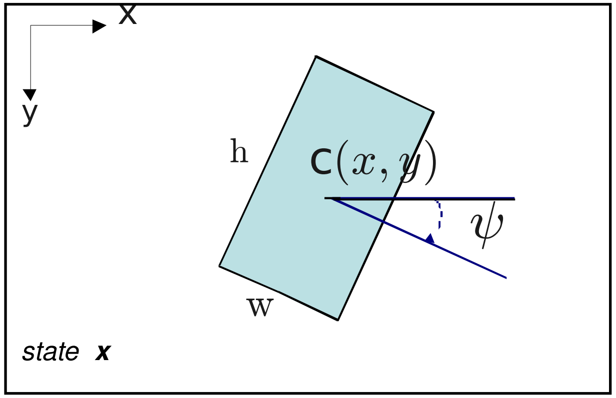
Target tracking



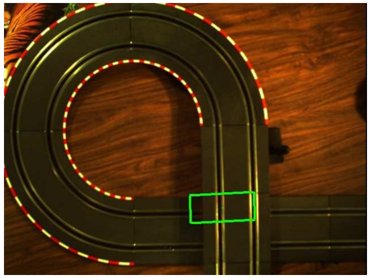
Vision based tracking

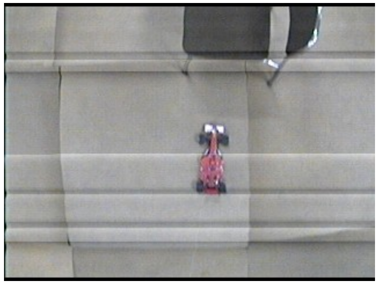
- Extracting relevant information about the position of the object in the camera frame
- Parameters to estimate :
  - Position, orientation, scale





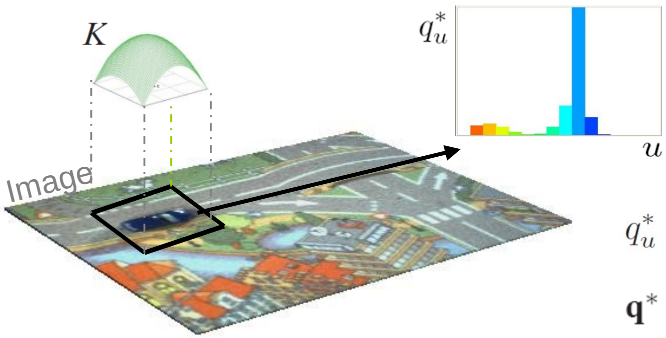
- Constraints:
  - Unknown environment
  - Occlusions
  - Poor image quality
  - Fast motion (large inter-frame displacement)
  - Real-time



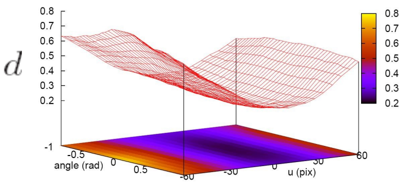


Choice of object representation:

- Non parametric representation, by color histograms



$$q_u^* = \sum_{i=1}^{n} K(\boldsymbol{l}_i - \mathbf{c})\delta_u(b(\boldsymbol{l}_i))$$
$$\mathbf{q}^* = \{q_u^*\}_{u=1\ldots m}$$



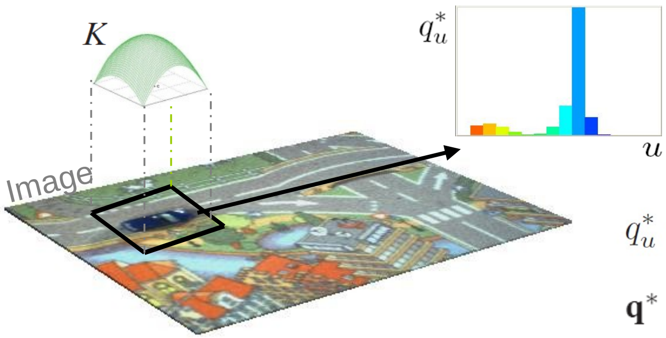
Single kernel configuration

Pros: Good robustness to appearance changes

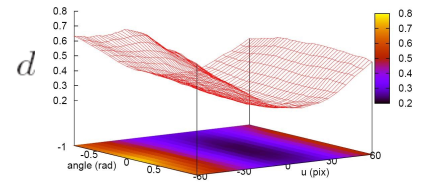
Cons: Lack of spatial information

Choice of object representation:

- Non parametric representation, by color histograms



$$q_u^* = \sum_{i=1}^{n} K(\boldsymbol{l}_i - \mathbf{c})\delta_u(b(\boldsymbol{l}_i))$$
$$\mathbf{q}^* = \{q_u^*\}_{u=1\ldots m}$$

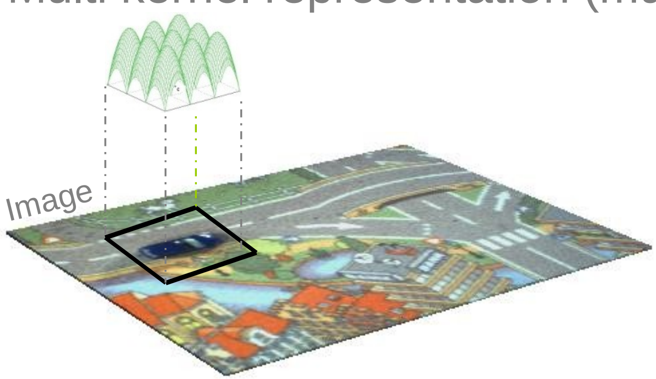


Single kernel configuration

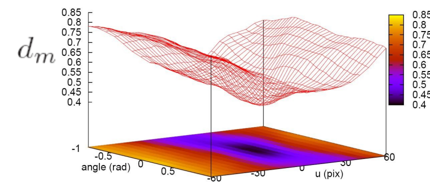
Pros: Good robustness to appearance changes

Cons: Lack of spatial information

- Multi kernel representation (multi histogram)



$$\{\mathbf{q}_j(\mathbf{x}_k)\}_{j=1..9}$$
$$d_m(\mathbf{x}_k) = \frac{1}{9}\sum_{j=1}^{9} d_j(\mathbf{x}_k)$$
$$d_j(\mathbf{x}_k) = d(\mathbf{q}_j^*, \mathbf{q}_j(\mathbf{x}_k))$$



Multi-kernel representation

Tracking scheme

Find the state which minimises the distance criterium $d_m(\mathbf{x}_k) = \frac{1}{9}\sum_{j=1}^{9} d_j(\mathbf{x}_k)$

Main tracking approaches:

Deterministic search

Iterative minimisation of the cost function by differentiation of the similarity criterium

Mean shift, gradient descent, Gauss-Newton, Levenberg-Marquardt…

[Comaniciu00][Hager04][Megret06][Fan07]

Bayesian methods

Estimation of the probability density function from the prior one

Particle filtering…

[IsardBlake98][Pérez04]

Tracking scheme

Find the state which minimises the distance criterium $d_m(\mathbf{x}_k) = \frac{1}{9}\sum_{j=1}^{9} d_j(\mathbf{x}_k)$

Main tracking approaches:

Deterministic search

Iterative minimisation of the cost function by differentiation of the similarity criterium

Mean shift, gradient descent, Gauss-Newton, Levenberg-Marquardt…

[Comaniciu00][Hager04][Megret06][Fan07]

Bayesian methods

Estimation of the probability density function from the prior one

Particle filtering…

[IsardBlake98][Pérez04]

\+ Good accuracy

- Single hypothesis
- Non robust to large occlusions

Tracking scheme

Find the state which minimises the distance criterium $d_m(\mathbf{x}_k) = \frac{1}{9}\sum_{j=1}^{9} d_j(\mathbf{x}_k)$

Main tracking approaches:

Deterministic search

Iterative minimisation of the cost function by differentiation of the similarity criterium

Mean shift, gradient descent, Gauss-Newton, Levenberg-Marquardt…

[Comaniciu00][Hager04][Megret06][Fan07]

- \+ Good accuracy

- \- Single hypothesis
- \- Non robust to large occlusions

Bayesian methods

Estimation of the probability density function from the prior one

Particle filtering…

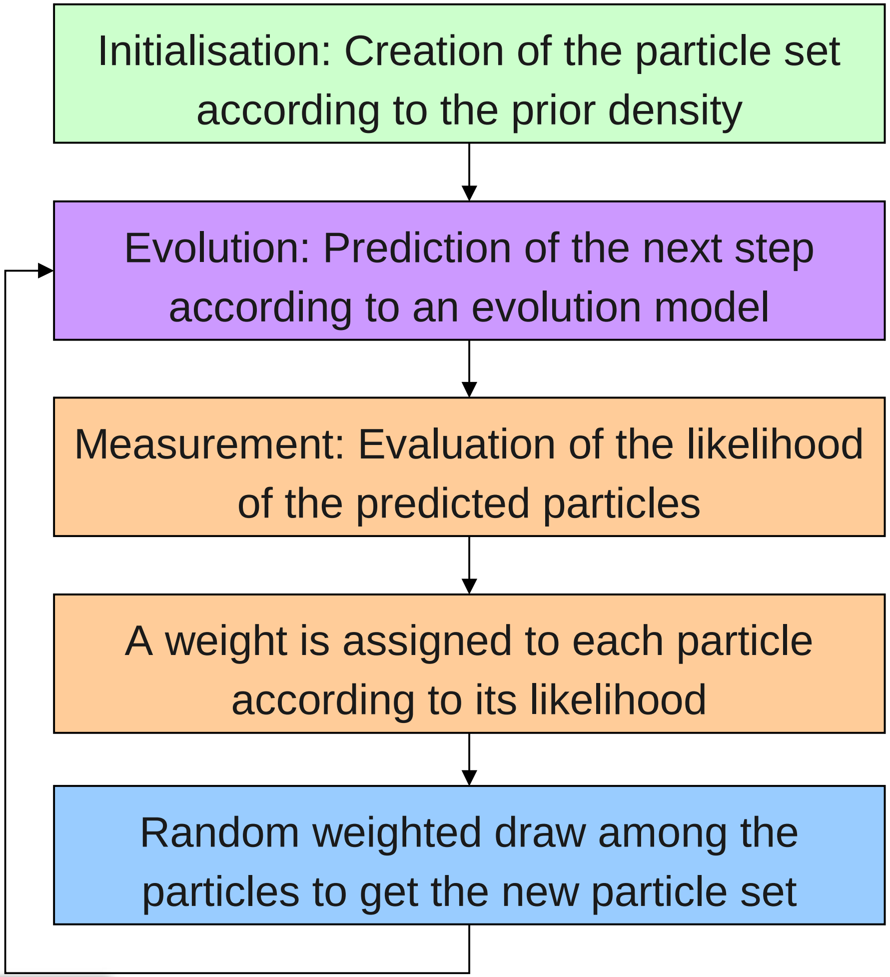
[IsardBlake98][Pérez04]

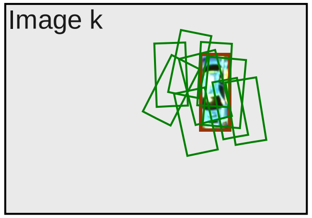
- \+ Robustness to occlusions
- \+ The whole pdf is estimated

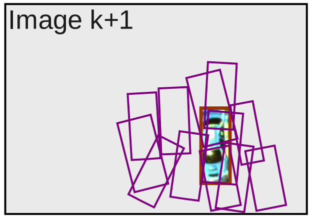
- \- Balance to find between accuracy and computation time

Color-based CONDENSATION scheme

- Reference histogram computed in the first frame

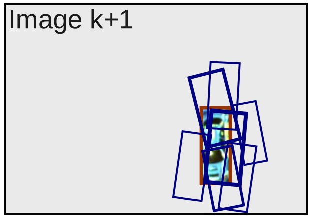






$$\{\mathbf{q}_j(\mathbf{x}_k)\}_{j=1..9}$$

$$w^{(i)} \propto p(\mathbf{z}_k \mid \mathbf{x}_k) \propto exp(-\lambda {d_m}^2(\mathbf{x}_k))$$



Color-based CONDENSATION example:

- Circuit sequence
  - **Evolution of the particles with constant velocity model**
  - **Output: weigthed mean (expectation estimate)**

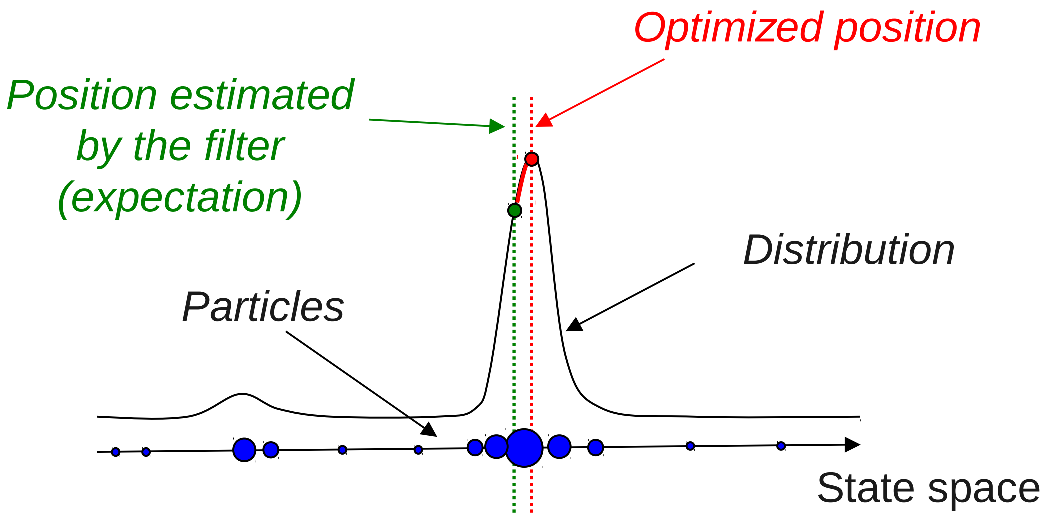
Particles

Estimate

Using deterministic search within particle filter:

How to use fewer particles while keeping a good accuracy?

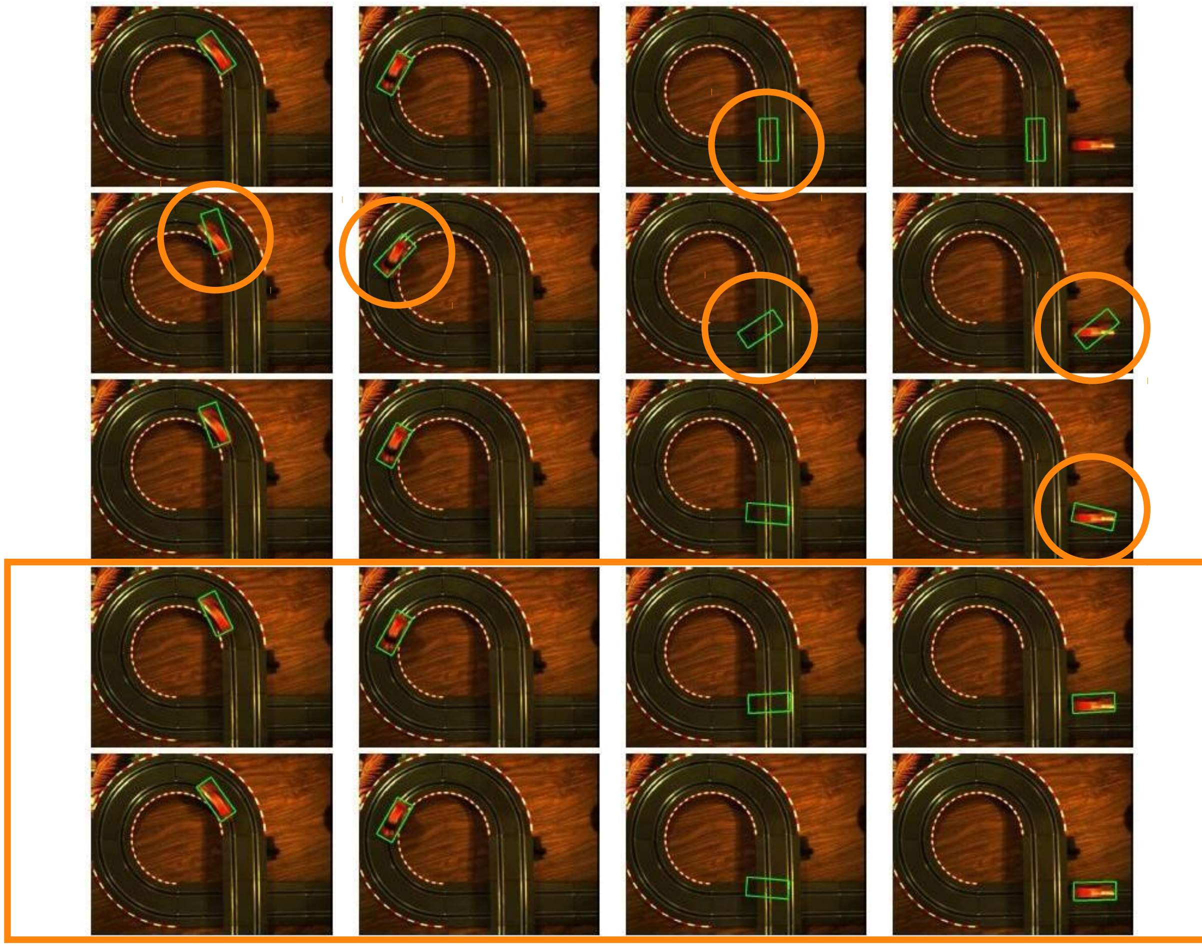
- Deterministic search from the position obtained by particle filtering



- ➡ Larger displacements
- ➡ Robustness to occlusions
- ➡ Accuracy
- ➡ Reduction of the particle number
- ➡ Reduction of the number of iterations in the minimization

Comparative results:

Sequence with full occlusion and blur



➡ **Deterministic search**

➡ **Particle filter with a single kernel (500p)**

➡ **Particle filter with multi-kernel (75p)**

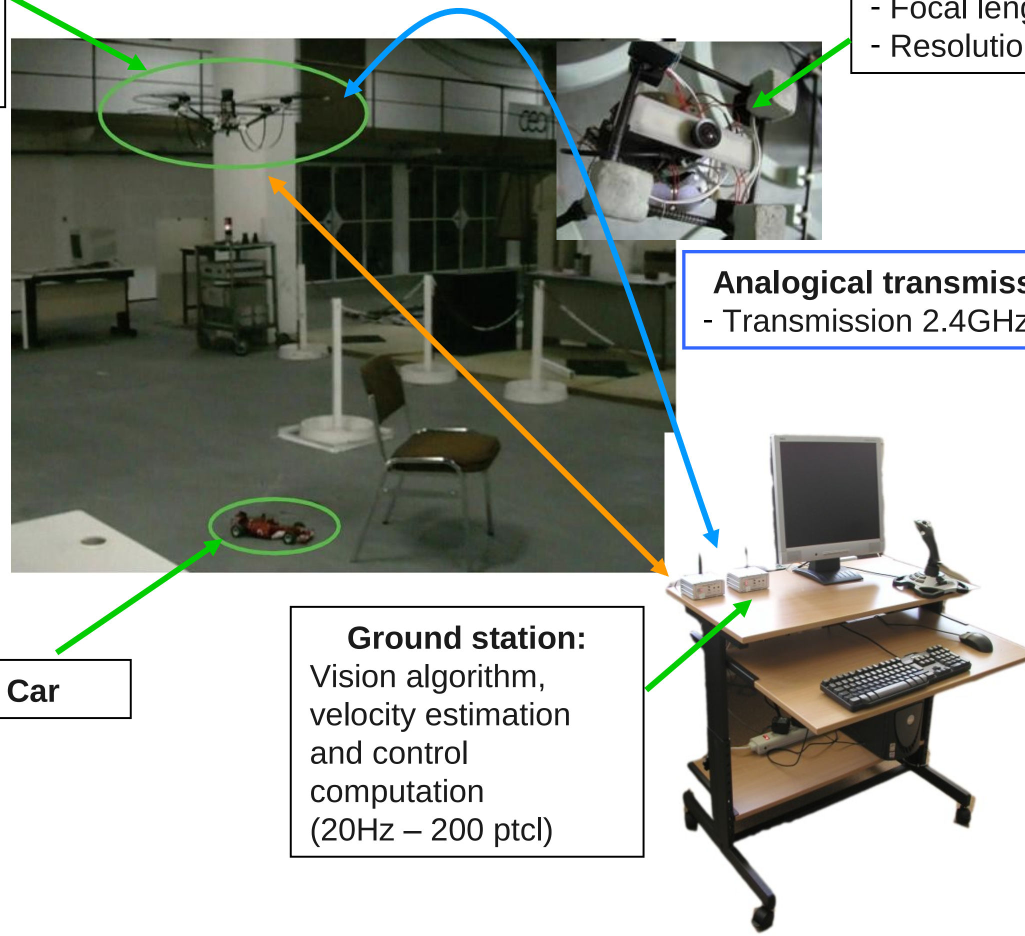
➡ **Particle filter with multi-kernel (500p)**
~4Hz / 5000 pix

➡ **Combined approach (75p)**
~10Hz / 5000 pix

Application to UAV target tracking:

UAV:
- Embedded attitude control (IMU data) 166Hz

Camera :
- Focal length f = 3.6mm
- Resolution 320x240



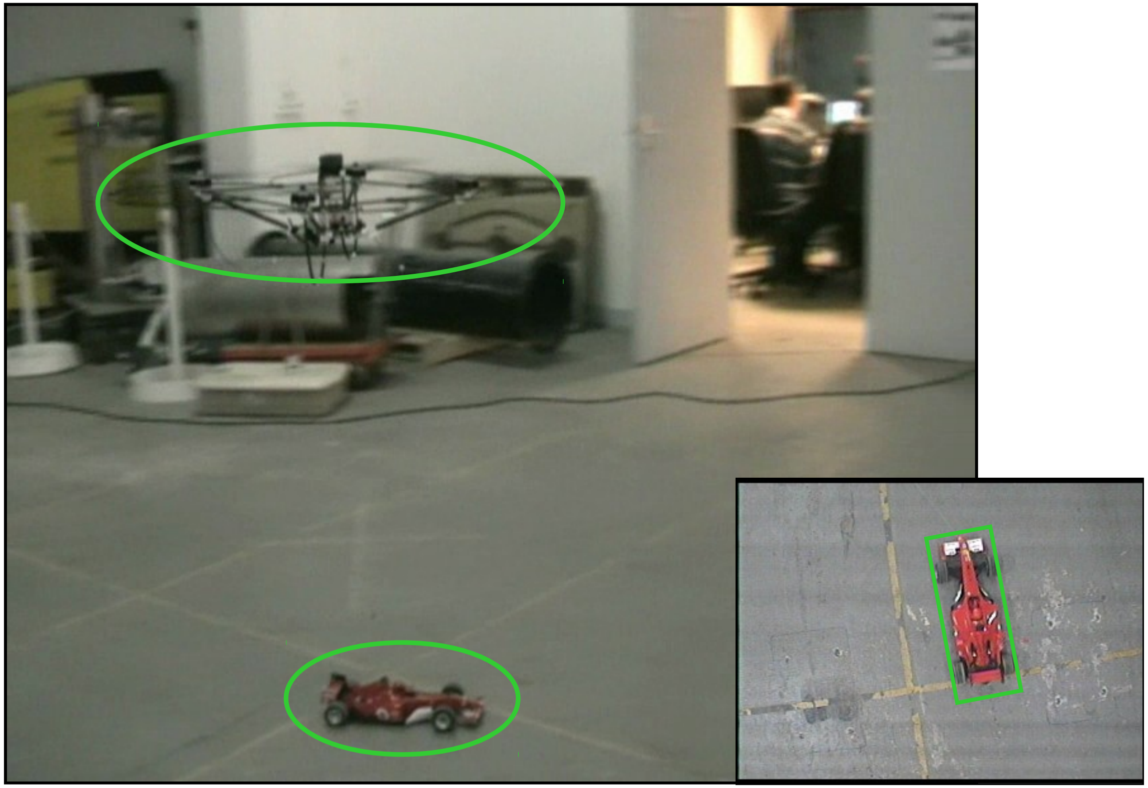
Numerical transmission:
- Transmission of the desired attitude to the UAV
- Transmission of the inertial data to the ground station

Analogical transmission:
- Transmission 2.4GHz

Car

Ground station:
Vision algorithm, velocity estimation and control computation (20Hz – 200 ptcl)

Video

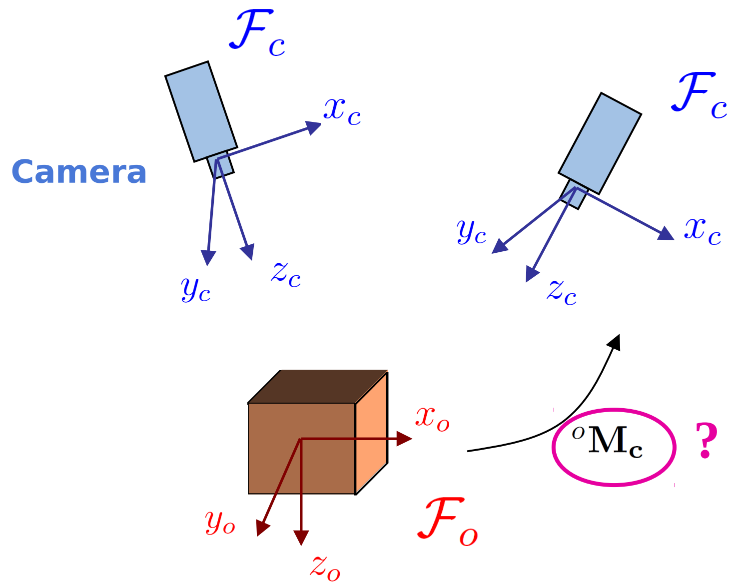


[Teuliere-IROS'11]

3D localization

Problem formulation

The hidden state is the full rigid-body transformation between the camera frame and the world frame



$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \in SE(3)$$

- Different strategies according to the prior knowledge available and the measurement function

Non Euclidean state space SE(3)

SE(3) is not a vector space but a Lie group

- Notions of distance, mean, or gaussian distribution need to be defined
  They are defined thanks to the structure of Lie Group of SE(3) which provides an exponential map between the Lie algebra se(3) and the Lie group SE(3)

$$\begin{aligned} se(3) &\mapsto SE(3) \\ \boldsymbol{\xi} &\mapsto \mathbf{M} = exp(\boldsymbol{\xi}) \end{aligned}$$

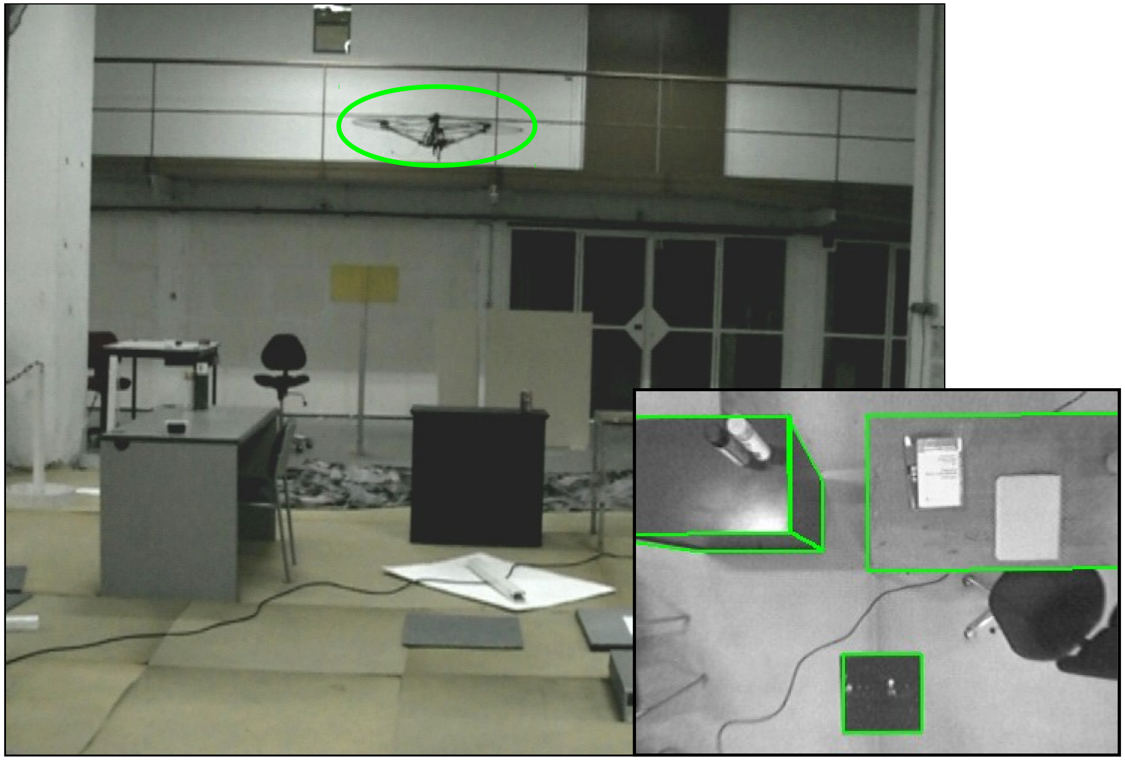
$$se(3) = \left\{ \boldsymbol{\xi} = \left[\begin{array}{cc} [\boldsymbol{\omega}]_\times & \boldsymbol{v} \\ 0 & 0 \end{array} \right] \mid [\boldsymbol{\omega}]_\times \in so(3), \boldsymbol{v} \in \mathbb{R}^3 \right\} \subset \mathbb{R}^{4\times 4}$$

- A pose matrix from SE(3) is then parametrized by a vector: $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$

$$\boldsymbol{v}, \ \boldsymbol{\omega} \in \mathbb{R}^3$$

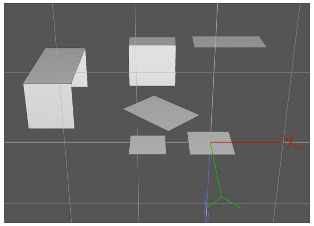
$$exp(\mathbf{v}) = exp\left(\left[\begin{array}{cc} [\boldsymbol{\omega}]_\times & \boldsymbol{v} \\ 0 & 0 \end{array} \right] \right)$$

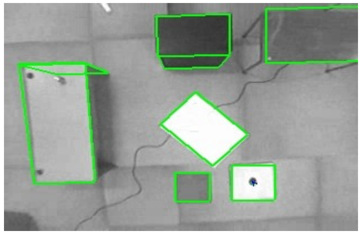
Localization with prior knowledge of the 3D model



3D model-based localization

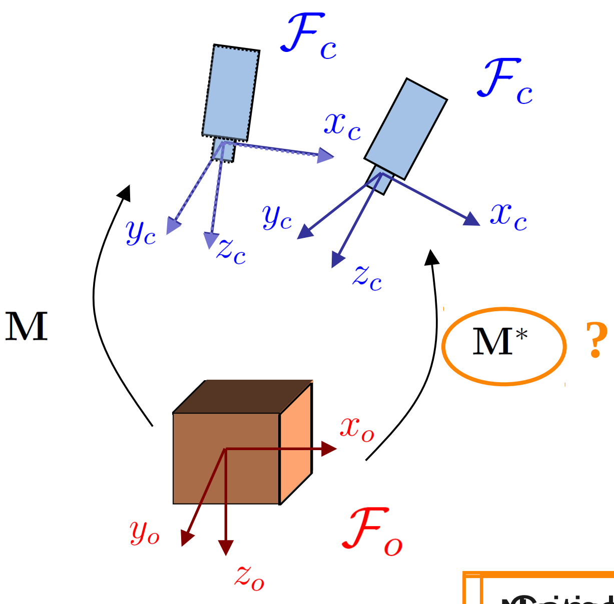
- Given:
  - A 3D model
  - A calibrated camera





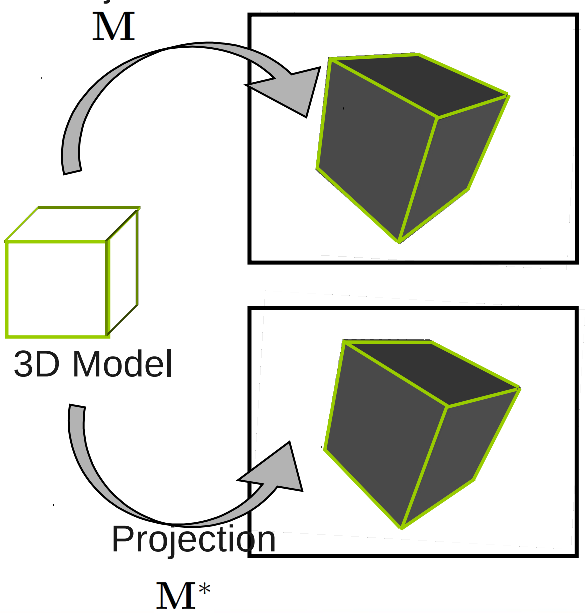
- Idea: the real 3D pose is the one providing a good alignement between the current frame and the projection of the model

Real world



New pose M* ?

Image plane



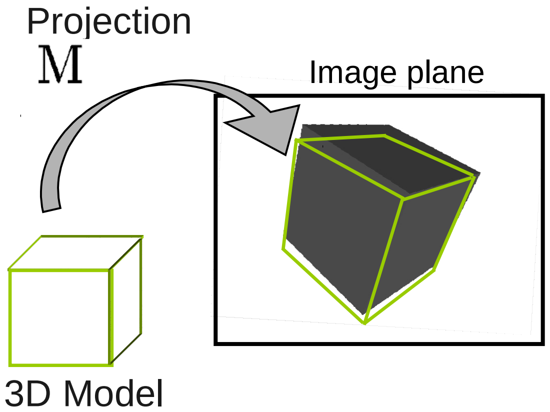
- Measurement?

Deterministic / Probabilistic approaches

Optimization based approaches

- Minimization of the reprojection error expressed as a cost function or an error to regulate to zero

$$\widehat{{}^c\mathbf{M}_w} = arg \min_{{}^c\mathbf{M}_w} \sum_{i=1}^{n} d^2(\mathbf{x}_i, pr({}^c\mathbf{M}_w, {}^w\mathbf{X}_i))$$

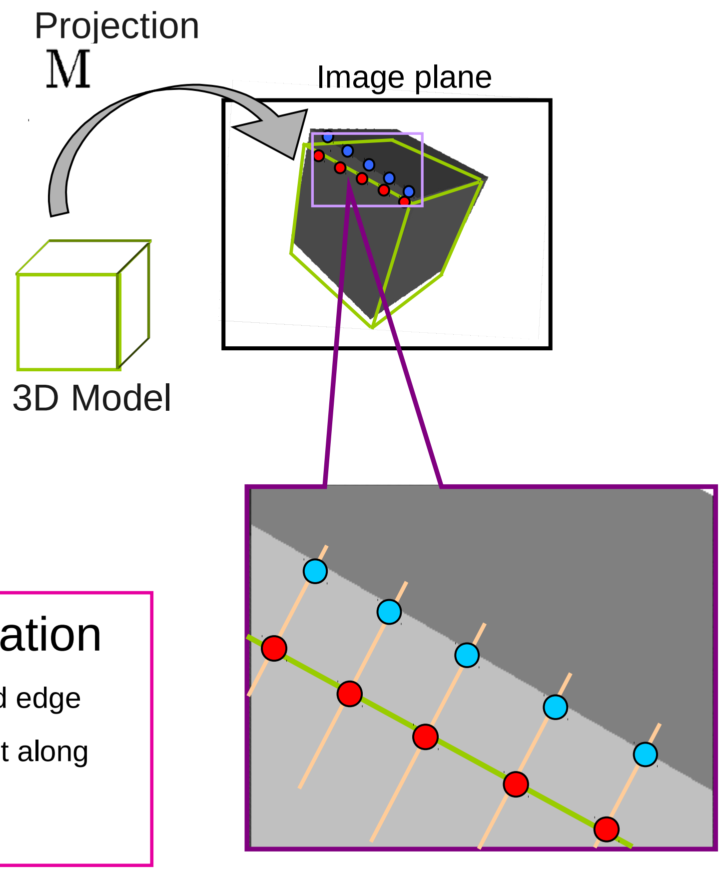


Deterministic / Probabilistic approaches

Optimization based approaches

→ Minimization of the reprojection error expressed as a cost function or an error to regulate to zero

$$\widehat{{}^c\mathbf{M}_w} = arg \min_{{}^c\mathbf{M}_w} \sum_{i=1}^{n} d^2(\mathbf{x}_i, pr({}^c\mathbf{M}_w, {}^w\mathbf{X}_i))$$



M → Projection of the model → Error vector computation

Error vector computation

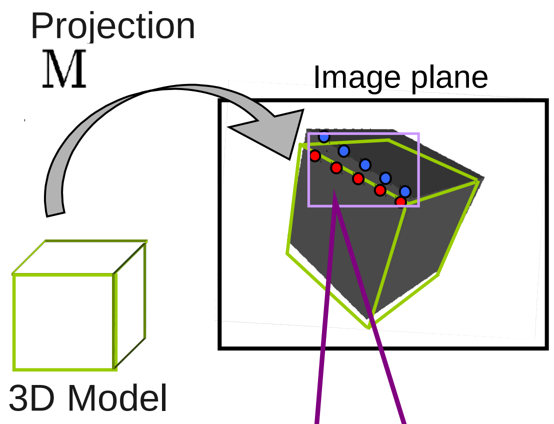
- Sample points on each projected edge
- Search for the strongest gradient along the normal

Deterministic / Probabilistic approaches

Optimization based approaches

- Minimization of the reprojection error expressed as a cost function or an error to regulate to zero

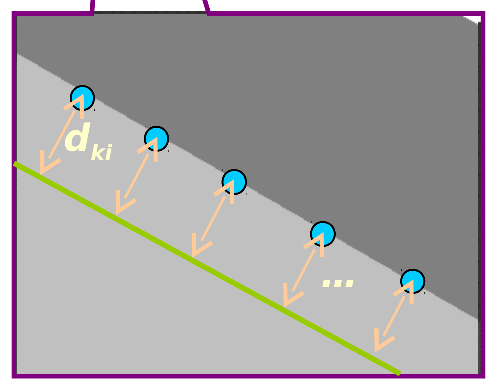
$$\widehat{{}^{c}\mathbf{M}_w} = arg \min_{{}^{c}\mathbf{M}_w} \sum_{i=1}^{n} d^2(\mathbf{x}_i, pr({}^{c}\mathbf{M}_w, {}^{w}\mathbf{X}_i))$$



M → **Projection of the model** → **Error vector computation**

- Sample points on each projected edge
- Search for the strongest gradient along the normal

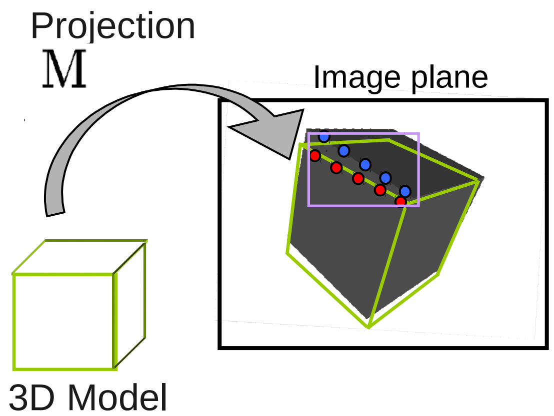
→ e

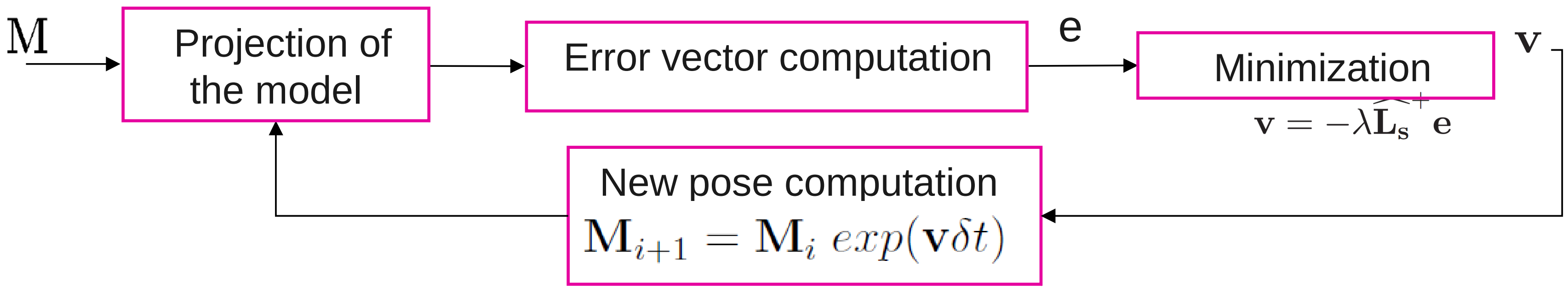


Deterministic / Probabilistic approaches

Optimization based approaches

➡ Minimization of the reprojection error expressed as a cost function or an error to regulate to zero
[Drummond02], [Comport06]





Deterministic approach

Example:
Métivier conference
room
[Comport-ISMAR03]

Deterministic / Probabilistic approaches

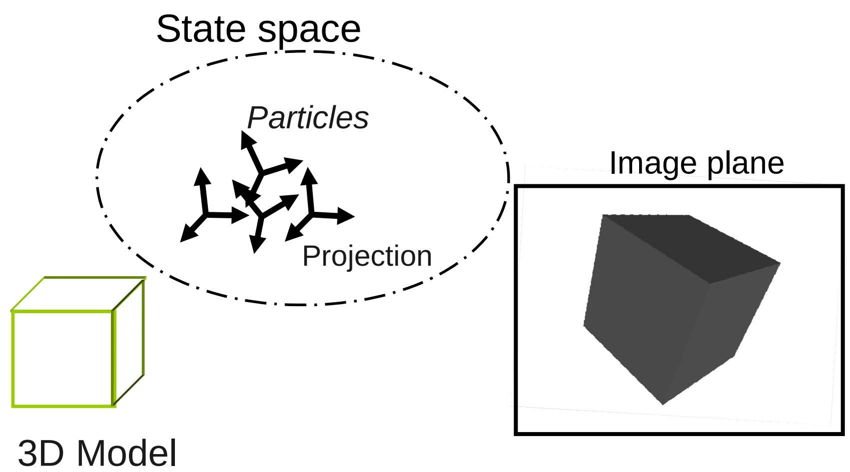
Deterministic approaches

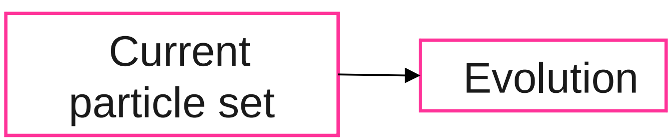
- Least square formulation
- Virtual visual servoing

Bayesian approaches

- Particle filter [PupilliCalway06]

⇒ Set of N particles





Deterministic / Probabilistic approaches

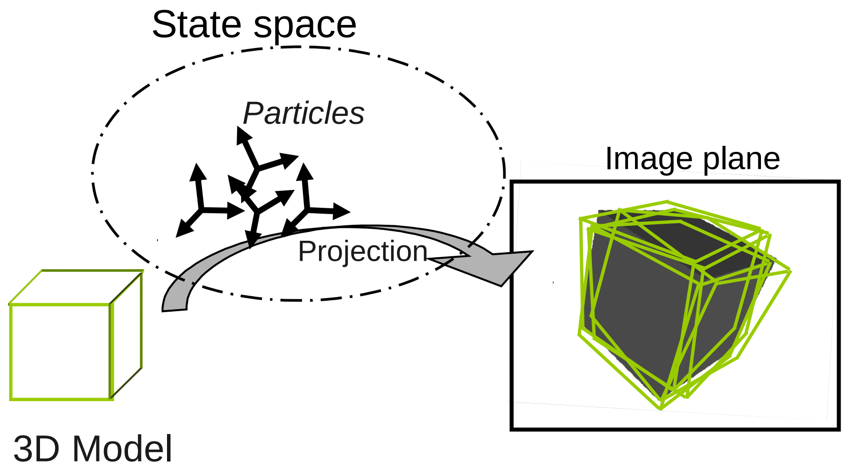
Deterministic approaches

- Least square formulation
- Virtual visual servoing

Bayesian approaches

- Particle filter [PupilliCalway06]

→ Set of N particles



Current particle set → Evolution → Likelihood computation

Deterministic / Probabilistic approaches

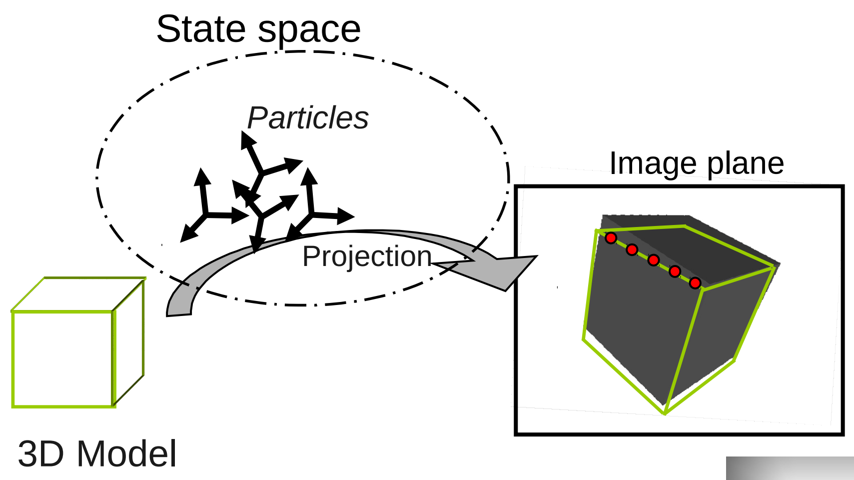
Deterministic approaches

- Least square formulation
- Virtual visual servoing

Bayesian approaches

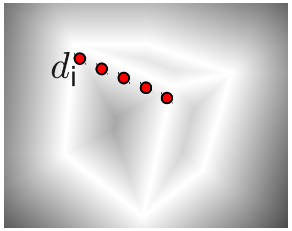
- Particle filter [PupilliCalway06]

➡ Set of N particles

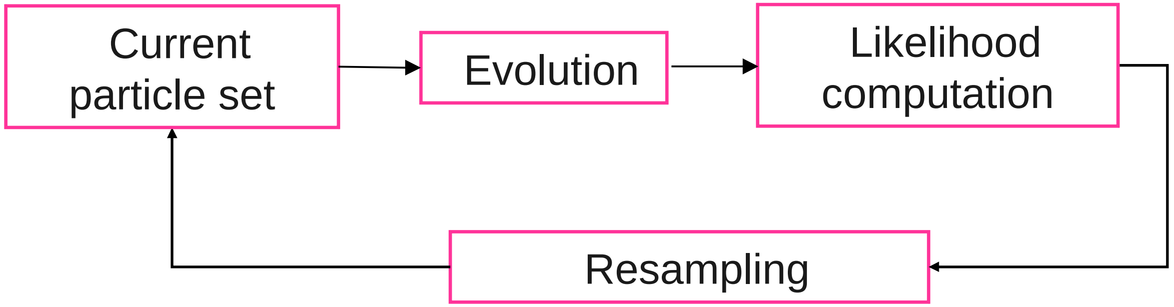


$$d(s) = \frac{1}{n}\sum_i d_i$$

$$p(\mathbf{z} \mid \mathbf{x} = s) = e^{-\lambda\left(\frac{d(s)-d_{min}}{d_{max}-d_{min}}\right)^2}$$



Distance map



Example

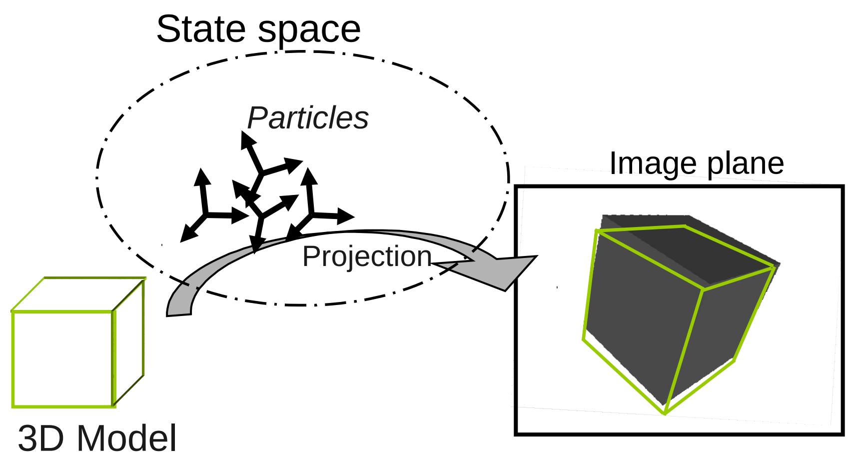
Example:
Box sequence

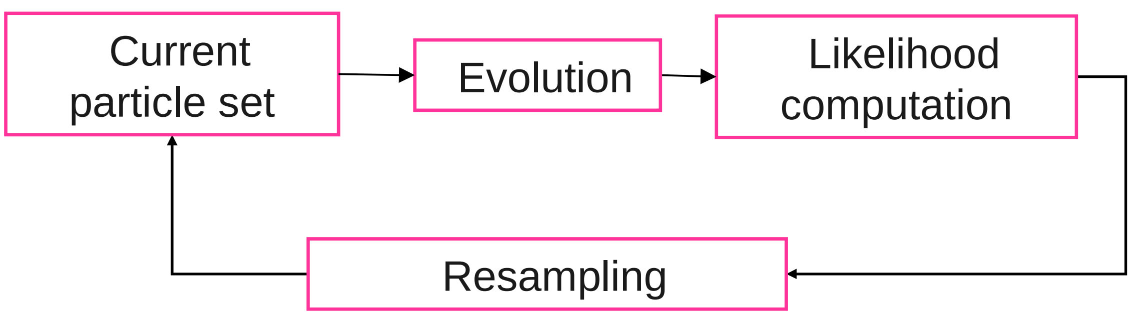
Example

Example:
Window sequence

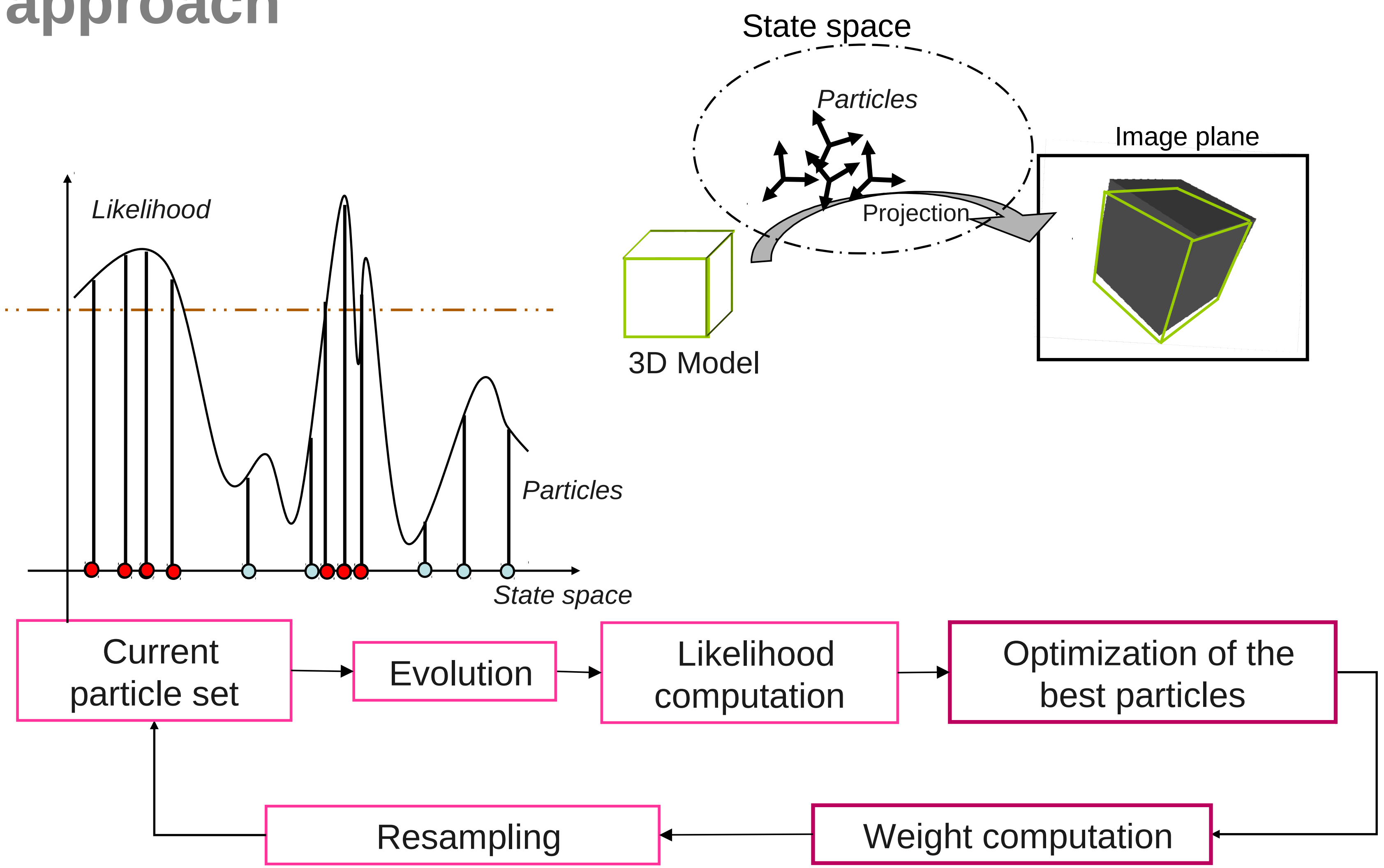
100 particles

Combination deterministic/probabilistic approach





Combination deterministic/probabilistic approach



Reweighting

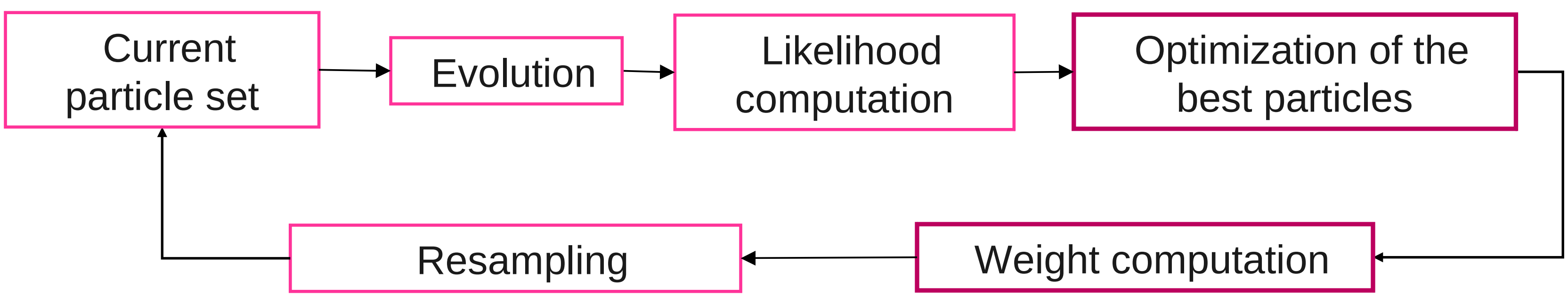
The proposal function is approximated by a mixture of Gaussian centered on the predicted and optimized particles

$$\pi_k^{(i)} \propto \frac{f_k(s_k^{(i)})}{g_k(s_k^{(i)})} p(\mathbf{z}_k \mid \mathbf{x}_k = s_k^{(i)})$$

$$\int \phi(\mathbf{x}) p^N(\mathbf{x}) d\mathbf{x} = \frac{1}{N} \sum_{i=1}^{N} \frac{p(s^{(i)})}{q(s^{(i)})} \phi(s^{(i)})$$

$$f_k(\mathbf{x}_k) = \frac{1}{N} \sum_{i}^{N} \mathcal{N}(s_k^{(i)}, \Sigma)(\mathbf{x}_k)$$

$$g_k(\mathbf{x}_k) = \frac{N}{N + N^*} \left(\frac{1}{N} \sum_{i=1}^{N} \mathcal{N}(s_k^{(i)}, \Sigma)(\mathbf{x}_k) + \frac{1}{N^*} \sum_{i=1}^{N^*} \mathcal{N}(s_k^{*(i)}, \Sigma)(\mathbf{x}_k) \right)$$



Example

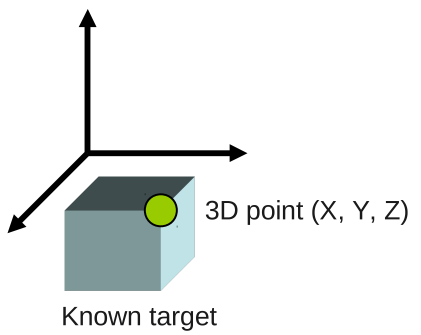
Comparative results

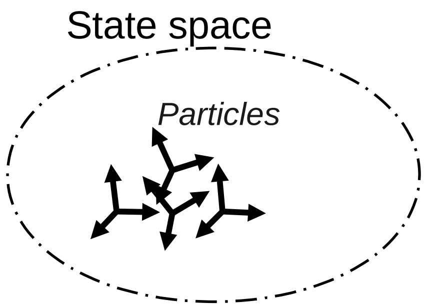
MH : Particle filter with optimized particles
25 particles ~10Hz

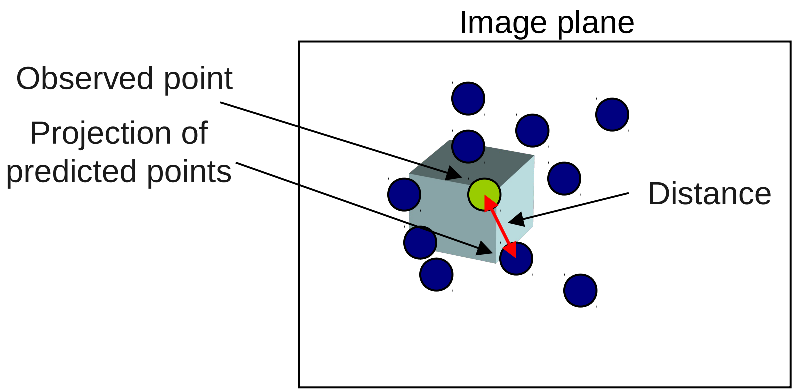
UAV (Kalman filter!)

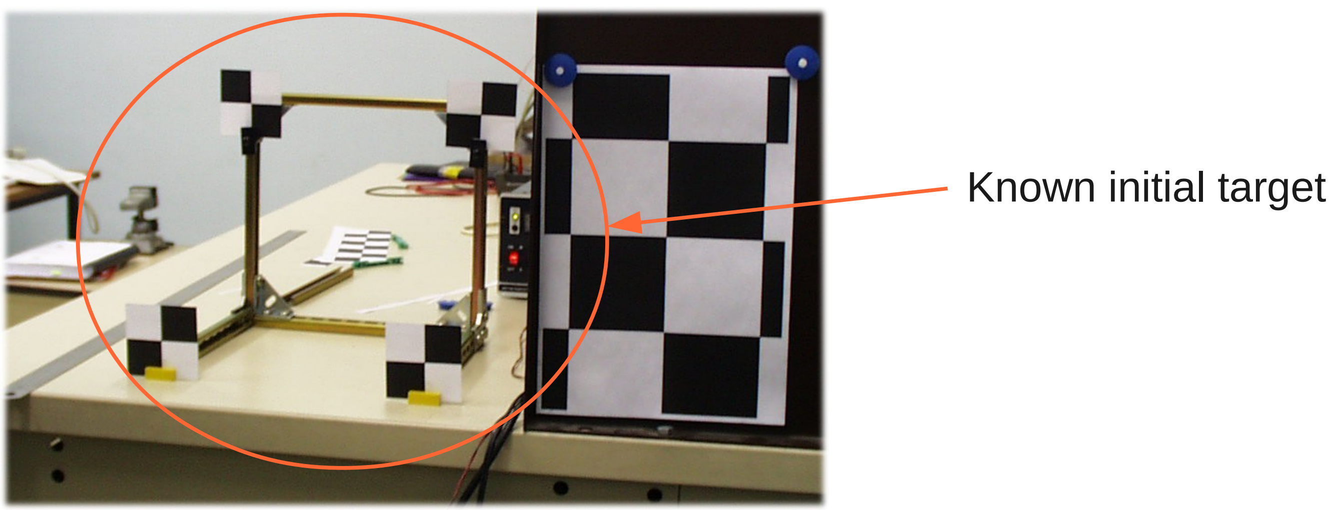
UAV navigation
Teuliere IROS'10

Second 3D localization example: no full model available









Principle

Example:
Filter initialization
(Eck, CEA-LIST)

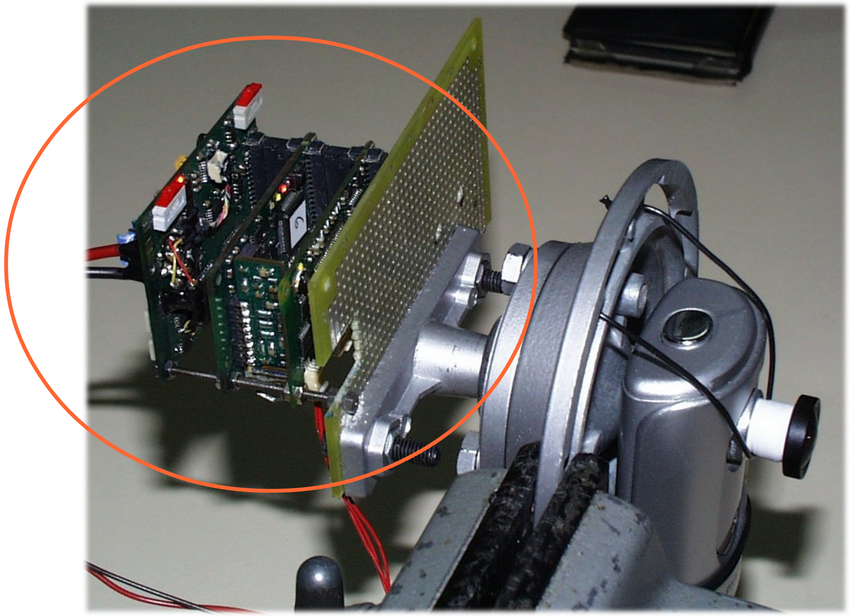
Particle filtering approach

Example:
Pupilli-Calway
BMVC05

Fusion with inertial sensor

IMU: gyrometers/accelerometers

- Gyrometers are used for the prediction (dynamic model of the UAV)
- Accelerometers are used in the likelihood
    - The obtained gravity vector should be vertical in the world frame for a pose to be correct



Results

Example:
Building exploration strategy
CEA-LIST

Conclusion

In this presentation we illustrated:

- Examples of mobile robotics issues: localization and tracking
- Deterministic and probabilistic approaches to vision based methods
- A recall on the most classic particle filter algorithms
- The importance of the choice of the measurement function
- The possibility to easily integrate other measurements
- Basic ideas to use deterministic search to guide the particle set towards regions of interest
- Real applications to UAV tasks