Particle Filter Applications

Sensor Fusion

Fredrik Gustafsson
fredrik.gustafsson@liu.se

Gustaf Hendeby
gustaf.hendeby@liu.se

Linköping University

Purpose

Illustrate the particle filter with some practical navigation applications.

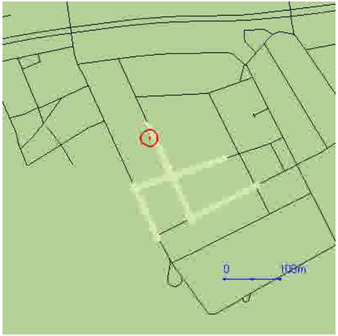
- All practical navigation systems rely on GPS (or more generally GNSS).
- GPS easy to jam and spoof, which is a security threat.
- GPS does not work when signals are blocked (behind big buildings, indoor, underground, underwater, etc).
- Need for support and backup system.
- The particle filter (PF) ideal tool in many applications!

Car Navigation by Odometry

Goal: GPS backup, support or replacement in situations when GPS is not available (the Manhattan problem, tunnels).

Basic idea:

- Prior given (from GSM cell data), and random points on the street network are sampled.
- Wheel speed sensors provide speed and yaw rate,
- Odometric model moves all particles
- Street map provides constraints
- Particle filter takes care of everything



Odometric Model

- The simplest form of odometric model is

$$\begin{aligned}
\psi_{k+1} &= \psi_k + T_s \dot{\psi}_k, \\
X_{k+1} &= X_k + T_s v_k \cos(\psi_k), \\
Y_{k+1} &= Y_k + T_s v_k \sin(\psi_k),
\end{aligned}$$

- Wheel speeds ω_3, ω_4 on rear axle provide virtual sensors of speed and yaw rate.

$$\begin{aligned}
v_x^m &= \frac{\omega_3 r_3 + \omega_4 r_4}{2} \\
\dot{\psi}^m &= \frac{\omega_3 r_3 - \omega_4 r_4}{B}.
\end{aligned}$$

  These are seen as input to the odometric model.

- The state is thus $x_k = (X_k, Y_k, \psi)^T$.

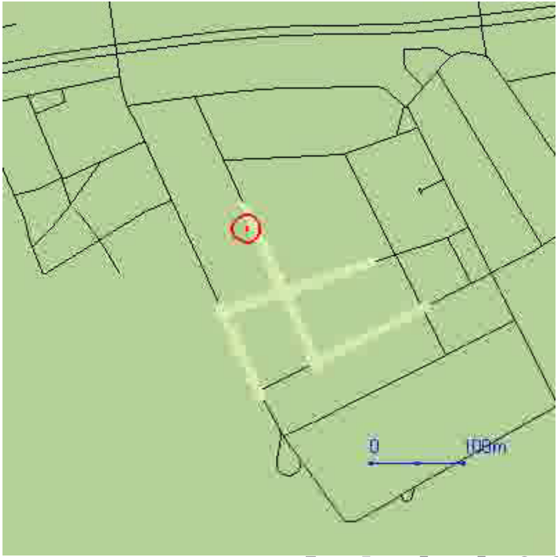
Road Constraints

There are different ways how to use road constraints in the PF.

- *Virtual measurement.* This is the most natural solution, but also the most ineffective. A lot of particles will diverge from the road in the time update, get a low probability and then not resampled. If off-road driving is an option, this is a good alternative though.
- *Road-constrained time update.* Here the state noise is chosen to make the vehicle move to a feasible point. This is much more effective, but requires some non-standard sampling. This is the solution demonstrated here.
- *Manifold model.* Here the state of the vehicle is just position p_k along a road segment i, and a discrete state for which road segment the vehicle is located at. This is the most effective solution, but hardest to implement.

Illustration

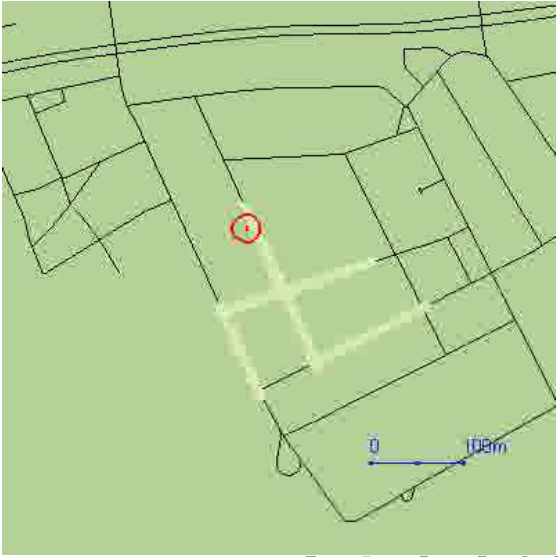
- Light green dots: particles.
- Red circle: GPS as ground truth (but be aware that it is often 20m off).
- Blue marker: estimated position activated when there is only one particle cloud left (unambiguous solution)
- Many possible solutions initially, so several clouds. One cloud after each other disappears until there is only one left.
- Implementation by Peter Hall.



http://youtu.be/aL0Tbt9Aw6k

Illustration

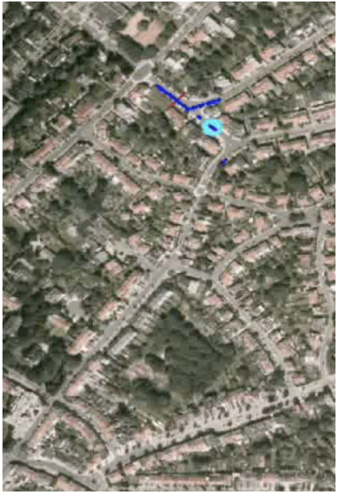
- Light green dots: particles.
- Red circle: GPS as ground truth (but be aware that it is often 20m off).
- Blue marker: estimated position activated when there is only one particle cloud left (unambiguous solution)
- Many possible solutions initially, so several clouds. One cloud after each other disappears until there is only one left.
- Implementation by Peter Hall.



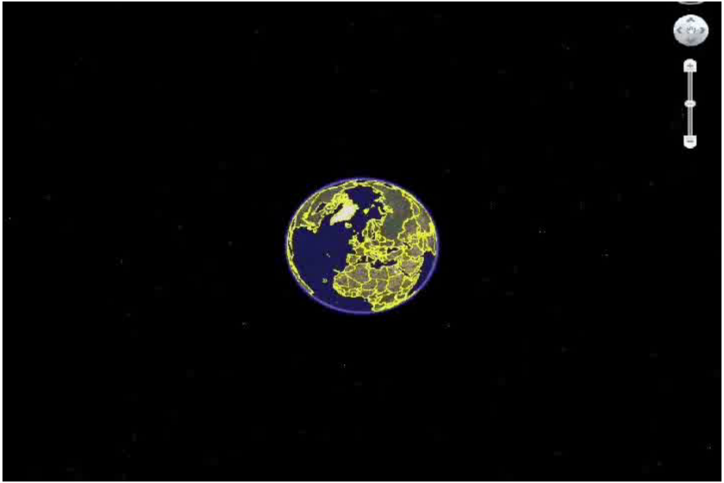
http://youtu.be/aLOTbt9Aw6k

Car Navigation by Radio Fingerprinting

- WiMAX network (Brussels here) provides signal strength measurements in the car.
- RSS map collected in a first step from cars with GPS.
- These geotagged RSS measurements provide a fingerprint $h(x_k)$.
- Simple constant velocity model used in a PF.
- Street map (optional) provides constraints.



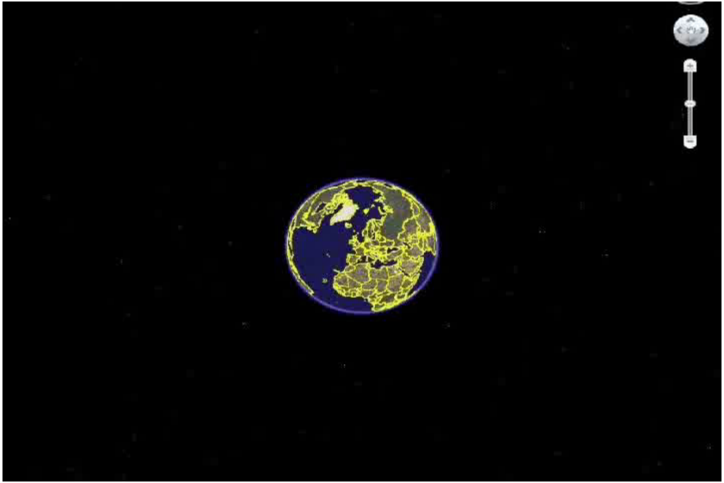
Car Navigation by Radio Fingerprinting



http://youtu.be/-6y2N5FD_3A

First half of video shows PF without street constraint, second half with road constraint (with superior performance). Implementation by Mussa Bhsara.

Car Navigation by Radio Fingerprinting

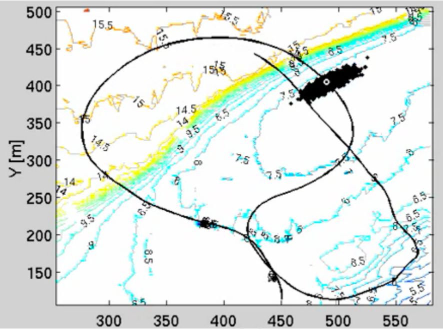


http://youtu.be/-6y2N5FD_3A

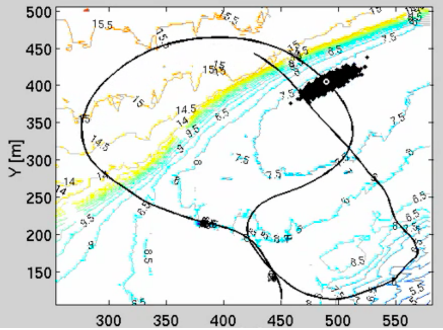
First half of video shows PF without street constraint, second half with road constraint (with superior performance). Implementation by Mussa Bhsara.

Underwater Navigation

- Underwater vessel measures its own depth and distance to bottom.
- The sum of these is considered as a depth measurement y_k.
- Sea chart provides depth $h(X_k, Y_k)$ as a function of position.
- An inertial navigation system provides the motion model with speed and gyro signals as inputs.
- PF integrates all this into a navigation system.



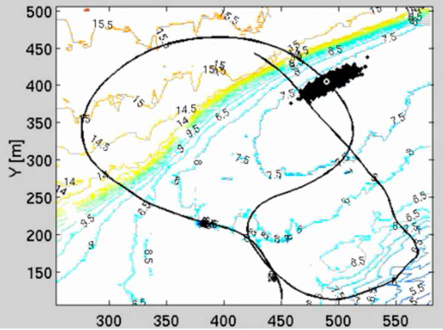
Underwater Navigation



http://youtu.be/JxUjVEn87yE

Video shows how a uniform prior quickly converges to a unimodal particle cloud. Note how the cloud changes form when passing the ridge. Tobias Karlsson produced the map, implementation and tests.

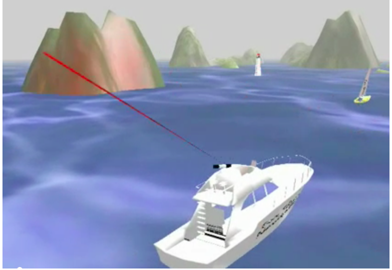
Underwater Navigation



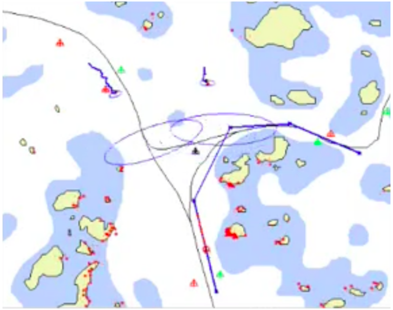
Video shows how a uniform prior quickly converges to a unimodal particle cloud. Note how the cloud changes form when passing the ridge. Tobias Karlsson produced the map, implementation and tests.

Surface Navigation using Radar Map Matching

On-board radar measures range to shore and possible its speed, sea chart provides conditional distance to shore $h(x_k)$.



http://youtu.be/zkPiWRgnDg4

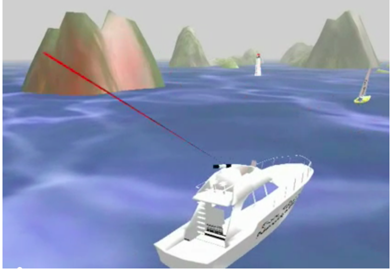


http://youtu.be/MGJQotL79Ns

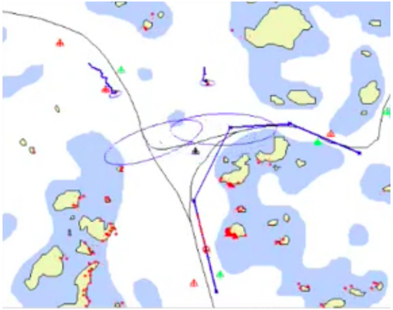
First video shows animated ship and radar measurements. Second video shows radar measurements overlaid on sea chart (given estimated position), the estimated (PF) position of the own ship, and the estimated (EKF) positions of other ships and 1 minute prediction of their position (collision warning). Implementations by Rickard Karlsson and Andreas Rönnebjerg

Surface Navigation using Radar Map Matching

On-board radar measures range to shore and possible its speed, sea chart provides conditional distance to shore $h(x_k)$.



http://youtu.be/zkPiWRgnDg4



http://youtu.be/MGJQotL79Ns

First video shows animated ship and radar measurements. Second video shows radar measurements overlaid on sea chart (given estimated position), the estimated (PF) position of the own ship, and the estimated (EKF) positions of other ships and 1 minute prediction of their position (collision warning). Implementations by Rickard Karlsson and Andreas Rönnebjerg

Sprint Orienteering

- Runners are wearing connected GPS.
- Their position is broadcasted live and can be replayed afterwards.
- Sprint orienteering takes place in cities, where buildings are sometimes blocking the GPS signals.
- The PF with a simple constant velocity model can correct outliers in GPS to provide a feasible trajectory that satisfies the rules.
- Rules in short: only allowed to run on the bright colours on the map.
- This provides the virtual measurement in the PF.

Sprint Orienteering

- Example from world championship in Venice 2014, and the winner Soren Bobach's race
- Red line: GPS position as broadcasted and recorded. Many times infeasible (crosses buildings, jumps over water)
- Green line: PF estimate.
- Blue line: ground truth, annotated by the runner himself.
- Work by Mathias Hallmén

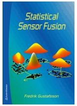
https://youtu.be/tu2HfpJne5M

Sprint Orienteering

- Example from world championship in Venice 2014, and the winner Soren Bobach's race
- Red line: GPS position as broadcasted and recorded. Many times infeasible (crosses buildings, jumps over water)
- Green line: PF estimate.
- Blue line: ground truth, annotated by the runner himself.
- Work by Mathias Hallmén

Summary

- PF can integrate convention sensor and motion models with unconventional information.
- PF handles non-unimodal distributions excellently (many clouds).
- Road constraint was a theme in the first two applications. This can be handled by the PF in different ways, but not by a KF.
- Excitation varies over time for map supported applications (car turning or vessel passing an underwater ridge).
- What we did not mention: PF is easy to implement, but tuning and practical considerations can be less straightforward.



Section 14.4