



A method to enforce map constraints in a particle filter's position estimate

Citation

Piche, R., & Koivisto, M. (2014). A method to enforce map constraints in a particle filter's position estimate. In 2014 11th Workshop on Positioning, Navigation and Communication (WPNC), 12-13 March 2014, Dresden, Germany IEEE. <https://doi.org/10.1109/WPNC.2014.6843284>

Year

2014

Version

Peer reviewed version (post-print)

Link to publication

[TUTCRIS Portal \(http://www.tut.fi/tutcris\)](http://www.tut.fi/tutcris)

Published in

2014 11th Workshop on Positioning, Navigation and Communication (WPNC), 12-13 March 2014, Dresden, Germany

DOI

[10.1109/WPNC.2014.6843284](https://doi.org/10.1109/WPNC.2014.6843284)

Copyright

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Take down policy

If you believe that this document breaches copyright, please contact cris.tau@tuni.fi, and we will remove access to the work immediately and investigate your claim.

A Method to Enforce Map Constraints in a Particle Filter’s Position Estimate

Robert Piché & Mike Koivisto
Department of Automation Science and Engineering
Tampere University of Technology, Finland

Abstract—When particle filters are used to estimate indoor position with floor plan constraints, it can happen that, even when all the particles lie in the corridor, the particles’ mean is not in the corridor. Such a position estimate is perceived by the user as a mistake in the algorithm. Projecting the particles’ mean to the nearest corridor location is an obvious ad-hoc solution, but it is not optimal and the trajectory may be discontinuous in time. Another solution is to use a maximum a-posteriori estimate for the particle cloud where the particles in an inaccessible region are eliminated. However, this optimal solution might also have discontinuous trajectory and so it is not ideal for the real time positioning.

In this work, the following principled approach is taken. Given a particle cloud representation of a posterior distribution for position, the position estimate is defined as the solution of a least squares problem with linear inequality constraints. This problem can be solved efficiently and reliably using standard numerical optimization algorithms and codes. Results are presented for simulated data and real-world data.

I. INTRODUCTION

In indoor positioning, particle filters are often used to combine floor plan data with WiFi received signal strength (RSS) and other data, see e.g.[2]. The walls’ non-crossing constraint is easily modelled in a particle filter and taking these constraints into account generally improves position estimate considerably.

However, it can happen that, even when all the particles representing the position’s posterior distribution lie in allowed regions, the particles’ *mean* is in an inaccessible region. This typically happens when the particle cloud is L-shaped, such as when a user is turning a corner in a corridor. Consequently, even when the particle filter is working correctly, the user sees the position estimate “cutting the corner” on the floor plan, and typically perceives this wall-crossing as a defect in the positioning estimation.

An obvious ad-hoc solution to this defect is to project the particles’ mean to the nearest corridor location, and report this location as the position estimate. However, the mean of the posterior distribution is the location that is optimal in the sense of mean square error, and this optimality is lost when using an ad-hoc projection. Also, the ad-hoc projection might jump discontinuously from one wall to another as the particle cloud flows forward with time, and such jumps might also be perceived by the user as a defect.

Another solution is to compute the maximum a-posteriori estimate from the particles [1]. While this estimate would remain in the feasible region, it might still have a discontinuous trajectory, because a mode can jump even for a continuously evolving continuous density function.

In this work, the following approach is taken to computing a position estimate from the cloud of weighted particles that represents the posterior distribution. The approach is inspired by the variational characterisation of the mean that was alluded to in the previous paragraph, that is, the fact that the (weighted) mean of the particle cloud is the location from which the weighted sum of squares of distances to the particles is minimised. The situation can be visualised with the help of a well-known physical analogy for least squares

estimation: the weighted mean is the static equilibrium of a system of linear springs whose spring stiffnesses are determined by the particle weights (Fig. 1).

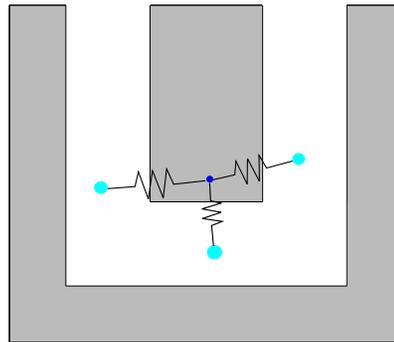


Figure 1. Physical analogy showing the mean as the static equilibrium of a system of linear springs; this is mathematically equivalent to the minimizer of the sum of squared errors.

It is straightforward to incorporate wall constraints into this variational characterisation of the mean: instead of an unconstrained minimum, the minimisation is done over the space of allowed positions defined by the floor plan. The result is an inequality-constrained linear least-squares minimisation problem. Thus, the estimate is optimal in the sense of minimising mean-square error. Also, because the estimate is an orthogonal projection, then if the particle cloud moves continuously in time, so does the estimate.

If, as is usually the case in practice, the feasible region can be decomposed into the union of a finite number of convex polygons, then the estimate can be computed efficiently and reliably using standard algorithms.

This work is organised as follows. In section II a standard particle filter and how it is used for positioning in a restricted 2D environment is presented. Sections III and IV present the wall-constrained mean and maximum a-posteriori position estimates for positioning. In section V the tests and results are presented and finally section VI concludes the paper.

II. PARTICLE FILTER

A particle filter is an estimation algorithm that approximates the posterior density of the state space $p(x_t|y_{1:t})$ with a set of weighted particles. Assume that the probability distributions $p(x_0)$, $p(y_t|x_t)$ and $p(x_{t+1}|x_t)$ are known and their densities are computable for each time step t . The vector x_t is called the state vector and the vector y contains the measurements.

In this work, the state is 4-dimensional vector containing 2-dimensional position and velocity vectors r_t and v_t

$$x_t = \begin{bmatrix} r_{1,t} \\ r_{2,t} \\ v_{1,t} \\ v_{2,t} \end{bmatrix}.$$

For the state and measurement model assume normally distributed state and measurement model errors

$$\begin{aligned} x_{t+1} &= f_t(x_t) + q_t, & q_t &\sim N(0, Q) \\ y_{t+1} &= h_{t+1}(x_{t+1}) + u_{t+1}, & u_{t+1} &\sim N(0, R). \end{aligned}$$

In the beginning of filtering independent and identically distributed particles $x_0^{(i)}$ for $i \in 1 \dots N$ are drawn from a given prior density $p(x_0) \sim N(m_0, D_0)$ and importance weights w_0^i are set equal such that $w_0^i = 1/N$ for all $i \in 1 \dots N$.

At a time instant $t+1$ the particles are drawn from the importance distribution which is also known as a proposal distribution [3]. In this work the state transition distribution $p(x_{t+1}|x_t)$ is used as a importance distribution and the new particles are drawn according to

$$x_{t+1}^{(i)} \sim p(x_{t+1}|x_t^{(i)}), \quad i = 1 \dots N.$$

Then, the importance weights of particles that are outside the feasible region C are set to zero. Importance weights are updated using measurements y_{t+1} , after which the unnormalised weights are scaled to sum to unity

$$\begin{aligned} w_{t+1}^i &\propto p(y_{t+1}|x_{t+1}^{(i)}), & i &= 1 \dots N \\ w_{t+1}^i &= \frac{w_{t+1}^i}{\sum_{i=1}^N w_{t+1}^i}. \end{aligned}$$

The estimate of a particle filter at a given time instant $t+1$ is a weighted mean of the particles

$$\hat{x}_{t+1} = \sum_{i=1}^N w_{t+1}^i x_{t+1}^{(i)}.$$

Every particle filter requires resampling stage to avoid all the weight concentrating to one particle. Resampling is done by sampling N new particles from the previous particle set such that the previous particle weights are used as probability densities [3]. In a standard particle filter the resampling is normally done at each time step.

III. WALL-CONSTRAINED MEAN

Consider a particle filter's posterior distribution of position in the plane at a given time instant, represented by particles located at $x^{(i)}$ and having importance weights w^i , for $i \in 1 \dots N$. Assume that the particle filter has eliminated particles that lie outside the feasible region C . Then the constrained minimum least-squares estimate is

$$\hat{x} = \arg \min_{\xi \in C} \sum_{i=1}^N w^i \|x^{(i)} - \xi\|^2. \quad (1)$$

In order to expedite the computation, let the feasible region be partitioned into disjoint convex polygons, that is, $C = \cup_{j=1}^J P_j$. Each convex polygon P_j can be characterised as the intersection of K_j half-planes:

$$\begin{aligned} P_j &= \bigcap_{k=1}^{K_j} \{x : a_{jk}x_1 + b_{jk}x_2 < c_{jk}\} \\ &= \{x : \begin{bmatrix} a_{j1} & b_{j1} \\ \vdots & \vdots \\ a_{jK_j} & b_{jK_j} \end{bmatrix} x < \begin{bmatrix} c_{j1} \\ \vdots \\ c_{jK_j} \end{bmatrix}\}, \end{aligned}$$

where the notation of inequality between vectors is to be interpreted as applying to all components.

With this representation of the feasible region, the estimate (1) can be written as

$$\hat{x} = \arg \min_{j \in 1:J} \min_{\xi^j \in P_j} \sum_{i=1}^N w^i \|x^{(i)} - \xi^j\|^2. \quad (2)$$

Each minimisation over P_j is a linear least-squares minimisation problem with linear inequality constraints, for which efficient and reliable algorithms exist, see e.g. [5, §5.2]. Implementations are available in numerical libraries, for example `lsqlin` in the MATLAB Optimization Toolbox. The computation is illustrated in Figure 2.

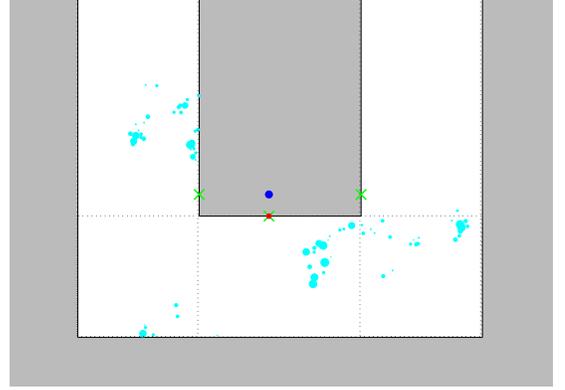


Figure 2. Computation of constrained position. The posterior is represented by particles (cyan dots) with weights (\propto area of dot). The feasible region is the union of three rectangles. The three minimizers of weighted sum of square distances over each rectangle are shown by green crosses. The position estimate (red circle) is the one that has the smallest weighted sum.

IV. MAXIMUM A-POSTERIORI ESTIMATE

To calculate a maximum a-posteriori (MAP) estimate for the particle filter the Viterbi algorithm can be used [1]. The Viterbi algorithm is a technique for the estimation of discrete state-space hidden Markov models. In a continuous state-space Markov model the discretisation of the state-space is generated automatically using any particle filtering method [1]. Consider a particle filter's posterior distribution of position in the plane represented by particles located at $x^{(i)}$ and having importance weights w^i , for $i \in 1 \dots N$. Again assume that the particle filter has eliminated particles that lie outside the feasible region. Because all the particles in the inaccessible region are eliminated the MAP estimate stays in the feasible region.

In the beginning of the filtering a posteriori densities $a_0^{(i)}$ for each particle are computed from a given prior distribution

$$a_0^{(i)} = \log(p(x_0^{(i)})), \quad i = 1 \dots N.$$

At every time instant $t \geq 0$ current set of particles are stored as a random grid denoted $\Omega_t^N = \{x_t^{(i)}\}$ for $i = 1 \dots N$ before resampling. At time $t+1 > 0$ assume the random grids Ω_t^N and Ω_{t+1}^N as well as $\{a_t^{(i)}\}$ for $i = 1 \dots N$ are available, then for $i = 1 \dots N$

$$\begin{aligned} a_{t+1}^{(i)} &= \log(p(y_{t+1}|x_{t+1}^{(i)})) + \max_{k \in \{1 \dots N\}} [a_t^{(k)} + \log(p(x_{t+1}^{(i)}|x_t^{(k)}))] \\ \psi_{t+1}^{(i)} &= \arg \max_{k \in \{1 \dots N\}} [a_t^{(k)} + \log(p(x_{t+1}^{(i)}|x_t^{(k)}))] \end{aligned}$$

The MAP estimate $\hat{x}_{t+1}^{\text{MAP}}$ is computed recursively at every time step $t + 1 > 0$

$$i_{t+1} = \arg \max_{i \in \{1 \dots N\}} a_{t+1}^{(i)}$$

$$\hat{x}_{t+1}^{\text{MAP}} = x_{t+1}^{(i_{t+1})}.$$

The Viterbi algorithm also includes a method for a MAP estimate backtracking for finding the best path from the current time instant to the beginning. For $k = t - 1, t - 2, \dots, 1$

$$i_k = \psi_{k+1}^{(i_{k+1})}$$

$$\hat{x}_k^{\text{MAP}} = x_k^{(i_k)}$$

This algorithm has a computational complexity $O(N^2t)$ and memory requirements of order $O(Nt)$ if the whole path $x_{1:t}^{\text{MAP}}$ is needed. In the case of estimation when only the latest MAP estimate is needed the memory requirements are only of order $O(N)$ and the computational complexity of order $O(N^2)$. In this case the past history of the simulated paths can be discarded and the storage requirements do not increase over time. [1]

V. TESTS AND RESULTS

We tested methods using simulated data and real-world data. The following linear state model was used in the particle filter in both test cases

$$x_{t+1} = Ax_t + q_t, \quad q_t \sim N(0, \sigma_q^2 I)$$

where

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where T is the time step size. In the tests we used $N = 400$ particles. In the beginning of the filtering a prior information $x_0 \sim N(m_0, D_0)$ was used, where

$$D_0 = \begin{bmatrix} \sigma_r^2 & 0 & 0 & 0 \\ 0 & \sigma_r^2 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix}$$

such that $\sigma_r^2 > \sigma_v^2$.

A. Simulated data

In the simulations we tested performance of the MAP estimate and the constrained minimisation problem solution by using a corridor with many turns. An imagined mobile device carried at uniform speed is tracked using RSS signals from 6 nearby WiFi access points. The simulated test track is 175 time steps long and the WiFi signals are received only once every 6–9 time steps. The simulated test track is illustrated in Figure 3.

In the simulation we used following non-linear measurement model to illustrate the captured measurement from the j -th access point at time t

$$y_{j,t} = 10 \log_{10} \left(\frac{P_0}{\|r_t - s_j\|^\alpha} \right) + u_t, \quad j = 1 \dots 6$$

where s_j is the location of the j -th access point, P_0 is the power of the transmitted signal and the exponent $\alpha > 1$ is the path loss exponent. The measurement noise $u_t \sim N(0, \sigma_u^2 I)$.

The filter parameters were chosen such that the process noise $\sigma_q^2 = 0.1$ and measurement model noise $\sigma_u^2 = 2$. Initial uncertainty for the position and velocity were $\sigma_r^2 = 1$ and $\sigma_v^2 = 0.01$ respectively.

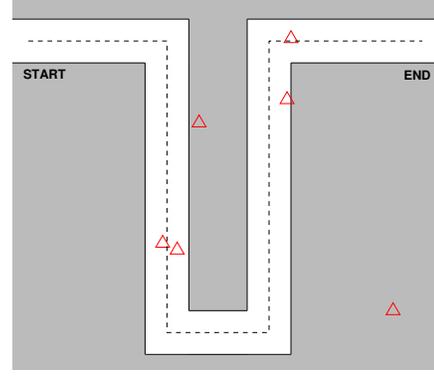


Figure 3. The simulated test track is shown by a black dash line. Grey area illustrates an inaccessible region and red triangles locations of the access points.

The MAP estimate stays in the feasible region because particles that were in the inaccessible region were eliminated. There were some time steps where the MAP estimate was closer to the true position than the unconstrained mean of the particles, but most of the time MAP estimate was jumping between the particles and that is not an ideal property in a real time positioning.

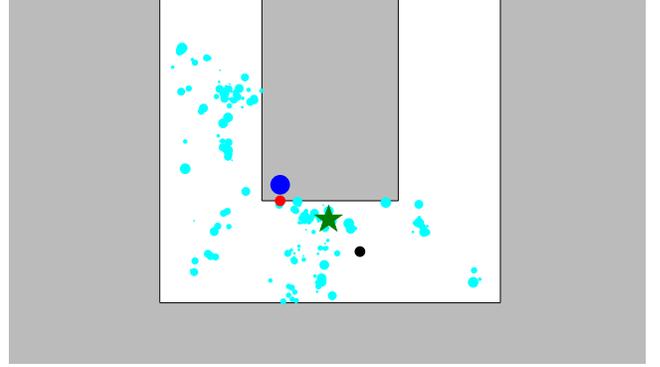


Figure 4. Simulated test track case where the unconstrained mean (blue circle) of the cyan particles is in the inaccessible region. A black circle is the true position and red circle is the constrained mean. A green star is the MAP estimate which performed slightly better at this instant.

The unconstrained mean of the particle cloud was inside the corridor most of time, but there were a few time steps in the middle of the track where the wall constraints became active. In straight parts of the corridor the unconstrained mean remained nicely in the feasible region as expected, but in some cases near turnings the particle cloud spread widely around the corner and caused the unconstrained mean drifted into the inaccessible region (Fig. 4). In those cases the constrained minimisation problem solution was nicely in the feasible region.

B. Real data

In the real data test presented methods are tested in one floor of a building in Tampere University of Technology. Data was collected using Acer Iconia tablet which was carried in hand while collecting measurements. The real data test track is shown in Figure 5. During the test track data collection the device collected WiFi measurements at 45 different time instants and each time approximately 25 WiFi access points were heard.

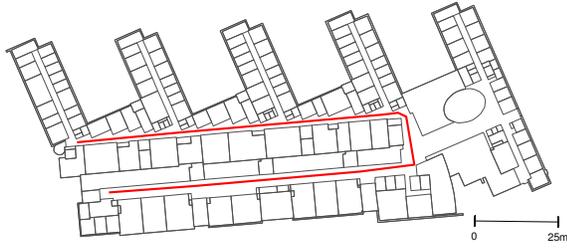


Figure 5. Real test track in a university building.

WiFi learning data was collected off-line from all over the building. These fingerprints were used to estimate path loss parameters as well as locations of the heard access points [4].

In the real data tests the unconstrained mean performed well and only occasionally it drifted into the inaccessible region. However, in those rare cases the constrained mean worked as expected and stayed in the feasible region in the corridor. In the straight parts of the corridor the constrained and unconstrained mean performed well and most of the time it was closer to the true position than the MAP estimate (Fig. 6).

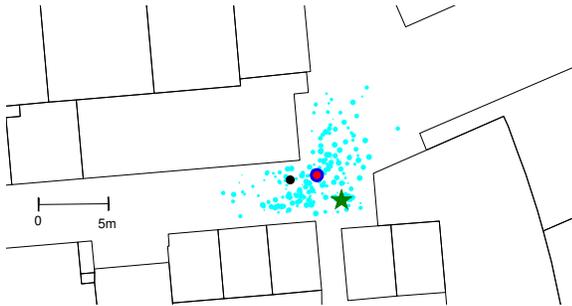


Figure 6. The mean of the particle cloud performed better than the MAP estimate most of the time.

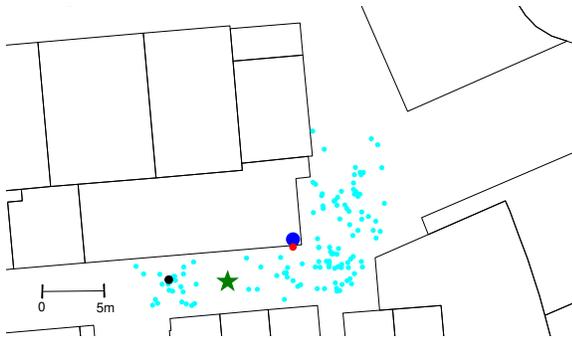


Figure 7. Last turning in the real data test where the MAP estimate (a green star) is closer to the true position than the constrained mean (a red circle) at the specific time instant.

The MAP estimate was again jumping between the particles but it stayed in the corridor because particles in the inaccessible region were eliminated. Like in the simulations, at a few time instants the MAP estimate was closer to the true position than the constrained mean as shown in Figure 7.

VI. CONCLUSION

In this work we proposed a method to enforce map constraints to a particle filter's position estimate. This method for ensuring that the

mean sequence does not cross walls is admittedly more complicated than an ad-hoc remedy, but it is a principled solution with attractive theoretical properties, and it is straightforward to compute using standard numerical algorithms.

The constrained mean of the particle cloud gave better results compared to the MAP estimate. At least in real time indoor positioning the jumpiness of the MAP estimate is not ideal despite the fact that the MAP estimate was closer to the true position at some specific time instants. Computationally the constrained mean is slightly faster than the MAP estimate and most of the time calculation of the constrained mean is not even necessary if the unconstrained mean of the particle cloud lies inside the feasible region.

VII. ACKNOWLEDGMENT

This research was funded by HERE, a Nokia business.

REFERENCES

- [1] Godsill, S.; Doucet, A.; West, M; Maximum a posteriori sequence estimation using Monte Carlo particle filters. *Annals of the Institute of Statistical Mathematics*, vol. 53, no. 1, 82–96, 2001.
- [2] Widyawan; Klepal, M.; Beauregard, S.; A backtracking particle filter for fusing building plans with PDR displacement estimates, *WPNC 2008*, pp. 207–212, 2008.
- [3] Nurminen, H.; Ristimäki, A.; Ali-Löytty, S.; Piché, R; Particle filter and smoother for indoor localisation, *IPIN 2013*
- [4] Nurminen, H.; Talvitie, J.; Ali-Löytty, S.; Müller, P.; Lohan, E-S.; Piché, R.; Renfors, M.; Statistical Path Loss Parameter Estimation and Positioning Using RSS Measurements in Indoor Wireless Networks, *IPIN 2012*
- [5] Åke Björck, *Numerical Methods for Least Squares Problems*, SIAM, 1996