APPLICATION OF PARTICLE FILTERING IN NAVIGATION SYSTEM FOR BLIND

Szymon Ceranka, Maciej Niedźwiecki

Technical University of Gdańsk
Faculty of Electronics, Telecommunications and Computer Science
ul. Narutowicza 11/12, 80-952 Gdańsk, Poland
cerankas@post.pl, maciekn@eti.pg.gda.pl

ABSTRACT

The navigation system for the blind, equipped with the GPS receiver, digital map and dead-reckoning sensors, is described. The problem of estimation of the pedestrian position, based on information from different sources, is solved using the approach known as particle filtering. The particle clustering and convex region mapping techniques are used to guarantee that at all times the position estimates are feasible, i.e. that they comply with the constraints imposed by the digital map of the traversed area.

1. INTRODUCTION

1.1. Motivation

Walking in urban environment is so easy for most of us that we pay almost no attention to this activity, but a blind person needs much concentration to travel without the help of a guide, even on a well-known route. A person deprived of visual stimuli must base his/her spatial orientation on such methods as feeling the surface properties with feet, estimating the distance to potential obstacles from the echo of their own footsteps, recognizing subtle smells or sounds characteristic of particular places, or counting steps to the point of changing the direction of movement, if there are no other clues. A momentary distraction of attention, unexpected obstacle, unnoticed important signal or mistake while counting steps may result a loss of orientation and force a blind person to seek help from other people. Unaided walking in unknown environment, even when surrounding the place of living, usually exceeds the possibilities of the blind.

1.2. Idea of a Navigation System for the Blind

A wish to overcome the above difficulties has become the motivation to undertake work on a navigation system for the blind [2][5]. It has been assumed that the system, when required by the user, should inform about current position, describe the nearest surroundings, and give hints helping to reach desired destination. In order to complete these tasks, the system must comprise a layer determining the geographical position, a database layer describing roads, buildings and other objects in the area of system operation, a control layer and an interface layer allowing the user to give commands and receive messages from the device. This paper describes the first layer, namely the one that provides the position estimates on the basis of sensor readouts.

1.3. Properties of Navigational Sensors

The sensors used in the system are the global positioning system (GPS) receiver, the electronic compass and the pedometer. The GPS receiver facilitates accurate determination of position - in case of using differential corrections this may by accuracy of single meters. Unfortunately, obstacles occurring in urban environment, such as high buildings, trees, bridges, etc. may attenuate or reflect the satellite signals, which frequently causes significant increase of errors or even a lack of position readouts from the GPS receiver. In order to maintain spatial orientation in case of missing GPS readouts, the system also allows dead-reckoning navigation based on direction readouts from the electronic compass and speed readouts from the pedometer. Not only does the dead-reckoning navigation replace the missing GPS readouts but it also increases accuracy of the position estimation when GPS operates properly.

2. FILTERING ALGORITHM

The accuracy of the GPS readouts depends on the quality of the satellite signals received and on the current configuration of satellites used by the system; the receiver provides parameters describing the current influence of both the above factors on the accuracy. The system should estimate the position, combining measurements from all sensors described above, taking into account their accuracy.

2.1. State Space Model

Denote by $q(t)$ the state vector associated with the moving person

$$q(t) = \begin{bmatrix} q_x(t) \\ q_y(t) \end{bmatrix}$$

where $q_x(t)$ and $q_y(t)$ describe pedestrian position in the local two-dimensional cartesian coordinate system and t denotes the discrete (normalized) time.
By $z_q(t) = [z_x(t), z_y(t)]^T$ we will denote the vector of position (state) measurements obtained from the GPS device.
The true (unknown) command signal, made up of the motion speed $u_\delta(t)$ and the motion direction (angle) $u_\theta(t)$, will be denoted by $u(t)$

$$u(t) = \begin{bmatrix} u_\delta(t) \\ u_\theta(t) \end{bmatrix}$$

Finally, by $z_u(t) = [z_\delta(t), z_\theta(t)]^T$ we will denote the vector of command measurements, comprised of the pedometer readout $z_\delta(t)$ and the compass readout $z_\theta(t)$, respectively.

The proposed state space model of pedestrian motion has a very simple form

$$\begin{aligned} q(t) &= q(t-1) + f(u(t)) \\ z_q(t) &= q(t) + e_q(t) \\ z_u(t) &= u(t) + e_u(t) \end{aligned} \qquad (1)$$

where

$$f(u(t)) = \begin{bmatrix} u_\delta(t)\sin(u_\theta(t)) \\ u_\delta(t)\cos(u_\theta(t)) \end{bmatrix} \qquad (2)$$

and the quantities $e_q(t)$ and $e_u(t)$ denote the corresponding measurement errors. In the sequel we will assume that the state and input measurement errors are white, mutually uncorelated Gaussian processes with zero means and covariance matrices equal to P_q and P_u, respectively

$$e_q(t) \sim \mathcal{N}(0, P_q)$$

$$e_u(t) \sim \mathcal{N}(0, P_u)$$

$$P_q = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \quad P_u = \begin{bmatrix} \sigma_\delta^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}$$

We will also assume that the initial state $q(0)$ is a Gaussian variable

$$q(0) \sim \mathcal{N}(q_0, P_0)$$

with the covariance matrix P_0 assigned in accordance with the initial position uncertainty.

Since the navigation system has usually no clues how its user is going to move (stop, accelerate and/or change the direction of movement), the simple, purely reactive form of the adopted model (the motion inertia was neglected) seems to be the most reasonable unprejudiced choice.

Since both the input (command) and output (position) signals are measured with errors, the state estimation, based on (1), belongs to the class of the so-called errors-in-variables problems.

2.2. Unconstrained Particle Filter

Particle filtering is a simulation-based sequential Monte Carlo approach which can be applied to state estimation whenever the analyzed system has complex (e.g. nonlinear) dynamics [1][3]. The basic idea of particle filtering is to represent the posterior probability density function of the unknown state vector by a set of random samples (called particles) with associated weights, and to express state estimates in terms of these samples and weights.

In case of the considered navigation system, the recursive particle filtering algorithm takes on the following form:

1. Generate the initial set of N particles (state samples) and assign them equal weights:

$$q^i(0) \sim \mathcal{N}(q_0, P_0), \quad w_i(0) = \frac{1}{N} \qquad (3)$$
$$i = 1, \ldots, N$$

2. Determine the new positions of the particles using the speed and direction measurements and randomly generated realizations of the measurement errors:

$$q^i(t) = q^i(t-1) + f(z_u(t) - e_u^i(t)) \qquad (4)$$
$$e_u^i(t) \sim \mathcal{N}(0, P_u), \quad i = 1, \ldots, N$$

3. Update the weights, taking into account consistency of the particle positions with the measured position of the moving object:

$$\begin{aligned} w^i(t) &= w^i(t-1)p(z_q(t)|q^i(t)) \qquad (5) \\ &= w^i(t-1)p_{e_q}(z_q(t) - q^i(t)) \end{aligned}$$

First, compute the unnormalized weights

$$\widetilde{w}^i(t) =$$

$$= w^i(t-1)\exp\left\{-\frac{1}{2}\left(z_q(t) - q^i(t)\right)^T P_q^{-1}\left(z_q(t) - q^i(t)\right)\right\}$$

$$i = 1, \ldots, N$$

Then, normalize weights so that their sum equals to one:

$$w^i(t) = \frac{\widetilde{w}^i(t)}{\sum_{j=1}^N \widetilde{w}^j(t)}$$

When the GPS readout is not available (due to the weak or incomplete satellite signal), the weights are not updated.

4. Determine the state estimate $\widehat{q}(t)$:

$$\widehat{q}(t) = \sum_{i=1}^N w^i(t)q^i(t) \qquad (6)$$

5. If the effective number of particles is less than a threshold,

$$N_{eff} = \frac{1}{\sum_i (w^i(t))^2} < N_{th} = \frac{2}{3}N \qquad (7)$$

then resample the particles by taking N samples with replacement from the set $\{q^i(t)\}$, where probability to take sample i is $w^i(t)$, and assign them equal weights: $w^i(t) = \frac{1}{N}$ for $i = 1, \ldots, N$.

6. Set $t = t + 1$ and go to step 2.

2.3. Constrained Particle Filter

A data base, describing the pavement regions in the area of system operation, is a very important part of the navigation system. If we assume that a person may move only through the pavement regions, then we may use the map as an additional source of information that puts constraints on possible locations of the moving object.

The particle filter allows one to enforce such constraints in a fairly straightforward way. In order to do this one should check whether the new particle positions, determined in the second step of the particle filter algorithm, fulfil the constraints mentioned above. Each particle $q^i(t)$ that falls outside the admissible region must 'die'. It should be replaced with a particle $q^j(t)$, drawn from the set of approved particles. The probability of drawing a particle $q^j(t)$ should be proportional to its weight $w^j(t)$. Half of the weight $w^j(t)$ of the drawn particle is transferred to the replaced particle.

After introducing this modification the probability density distribution, represented by the collection of particles, is truncated to the pavement regions, as shown in Fig. 1.

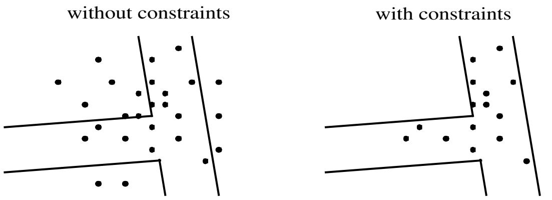

Figure 1: Constraining the admissible position of the moving object.

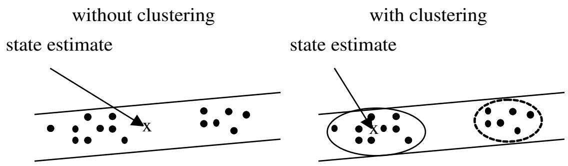

Figure 2: Influence of particle clustering on the quality of the state estimate.

2.4. Particle Clustering

During the operation of the particle filter the particles often form clusters, more or less distinct. This may happen each time the estimated probability density function becomes multimodal. In a case like this the formula (3) may yield estimates that fall outside the existing clusters (see Fig. 2), i.e. at locations where the approximated pdf is close to zero. In order to eliminate this negative effect, the method of determining the final state estimate was modified. Instead of performing step 4 of the algorithm described in subsection 2.2, the set of particles is divided into clusters. The weighed average (6) is calculated only for the particles that belong to the cluster with the maximum total (accumulated) weight.
The implemented clustering algorithm is based on the assumption that every two particles lying no more than the threshold distance d apart, belong to the same cluster. In order to limit the computational load of the clustering operation, the set of particles $\{q^i\}$ is first sorted according to the coordinate q_x^i, so that after sorting it holds $i < j \Rightarrow q_x^i \leq q_x^j$.
As a result of applying the clustering procedure, presented below, each particle q^i is assigned a cluster index $c(i)$.

1. Initialize the initial cluster indexes $\widetilde{c}(i) = 0$ and the base cluster indexes $\underline{c}(i) = i$ for $i = 1, \ldots, N$. Initialize the number of initial clusters $\widetilde{N}_c = 0$ and the number of clusters $N_c = 0$.
2. For each particle index i from 1 to N do the following:
   (a) If the initial cluster index $\widetilde{c}(i)$ assigned to the particle q^i is still zero (initial value), create new initial cluster: increase by one the number of initial clusters $\widetilde{N}_c$ and set $\widetilde{c}(i) = \widetilde{N}_c$.
   (b) Clear the list of cluster indexes L. Add the initial cluster index $\widetilde{c}(i)$ of the current particle to the list L.
   (c) Initialize j to $i + 1$.
   (d) While $j \leq N$ and $q_x^j \leq q_x^i + d$, do the following:
      i. If the distance from q^j to q^i is not greater than d, do the following:
         A. If the initial cluster index $\widetilde{c}(j)$ is zero, set $\widetilde{c}(j) = \widetilde{c}(i)$.
         B. If the initial cluster index $\widetilde{c}(j)$ is different than current index $\widetilde{c}(i)$, and the index list L doesn't yet contain the index $\widetilde{c}(j)$, then add the index $\widetilde{c}(j)$ to the list L.
      ii. Increase j by one.
   (e) Initialize the lowest base cluster index $\underline{c}_{min} = \underline{c}(\widetilde{c}(i))$.
   (f) For each initial cluster index k on the list L, if $\underline{c}(k) < \underline{c}_{min}$, set $\underline{c}_{min} = \underline{c}(k)$.
   (g) For each initial cluster index k on the list L, if $\underline{c}(k) \neq \underline{c}_{min}$, do the following:
      i. Set $\underline{c}(k) = \underline{c}_{min}$.
      ii. For each l from 1 to $\widetilde{N}_c$, if $\underline{c}(l) = k$, set $\underline{c}(l) = \underline{c}_{min}$.
3. For each initial cluster index i from 1 to $\widetilde{N}_c$, if $\underline{c}(i) = i$, do the following:
   (a) Increase the number of clusters N_c by one.
   (b) Set $g(i) = N_c$.
4. For each initial cluster index i from 1 to $\widetilde{N}_c$, if $\underline{c}(i) \neq i$, set $g(i) = g(\underline{c}(i))$.
5. For each particle index i from 1 to N set $c(i) = g(\widetilde{c}(i))$.

Let us assume that two particles are ”neighbours” if they are no more than the threshold distance apart. A set of neighbouring particles will then be a set where every two particles can be connected with a chain of particles being one another's neighbours.
The clustering algorithm is based on the concepts of initial clusters and base clusters. An initial cluster is a set of neighbouring particles, but it need not contain all such particles. A base cluster is a set of initial clusters that, after the clustering is performed, will contain all the neighbouring particles.
The concept of initial clusters is realized by assigning individual particles to initial clusters, i.e. by setting an initial cluster index for every particle.
The concept of base clusters is realized by assigning individual initial clusters to base clusters, i.e. by setting a base cluster index for every initial cluster. A base cluster index is actually the lowest of indexes of initial clusters, which have already been assigned to a particular base cluster.
Initial cluster indexes are assigned permanently (within the scope of one execution of the algorithm). On the other hand, base cluster indexes can be changed during one execution of the algorithm.
For every particle processed, all successive particles on the x axis are checked for being its neighbours. If a found neighbouring particle has not been assigned to any initial cluster, then it is assigned to the current initial cluster. If a found neighbouring particle has already been assigned to an initial cluster, then that initial cluster is merged with the current initial cluster; this is done after checking all the successive particles, by assigning equal base cluster indexes to every initial cluster found to be neighbouring with the current initial cluster.
Finally, the initial cluster indexes and the base cluster indexes are resolved to assign a uniform and consistent cluster index to every particle.

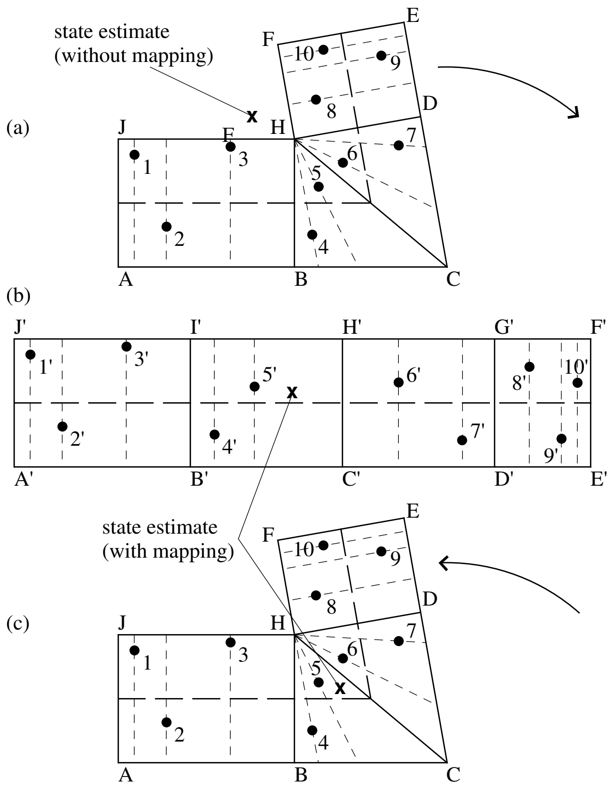

Figure 3: Mapping of a track section into a rectangle: (a) the original track section, (b) the track section transformed into a rectangle (c) the rectangle transformed back into the track section.

Another approach to particle clustering, though only briefly characterized, was proposed in [4].

2.5. Dealing with Non-Convex Track Sections

Even if we eliminate all particles lying outside the pavement regions and all particles not belonging to the cluster with the highest total weight, the state (position) estimate, calculated as the weighted average of individual particle states, may fall outside the pavement area. This may happen whenever the analyzed fragment of the pavement network, occupied by the dominant cluster, forms a non-convex polygon. To rule out such situations, a simple invertible transformation is used, that maps back and forth each non-convex track section into a convex area.

The particles are transformed from the track section into the convex area, and a weighed average of the transformed particle states is calculated. The result of averaging is then transformed back into the original state space where it forms the state estimate. This estimate is guaranteed to fulfil the constraints imposed on the state space.

The database describes the track segments as areas of uniform width r and two specified end points. A union of two adjacent track segments forms two trapeziums, as shown in Fig. 3(a).

First, from the pavement network we select two adjacent track segments that contain the heaviest subset of particles of the heaviest cluster ("the heaviest" means here "having the greatest total weight"). If the particles of the heaviest cluster are contained in one single pavement segment, no mapping is necessary, because all particles are already contained in a convex area.

Fig. 3(a) shows a track section consisting of two trapezoidal track segments $ACHJ$ and $CEFH$. The first step of the transformation is to rotate the trapezoid $CEFH$ around the point C in order to bring the section CE in line with the section AC. The second step consists of mapping the triangles BCH and CDH into the rectangles $B'C'H'I'$ and $C'D'G'H'$, respectively - see Fig. 3(b).

To describe the mapping for the triangle BCH we will introduce a local coordinate system with the origin in point B, the X axis pointing to C, and the Y axis pointing to H. The point (x, y) inside the triangle BCH is then transformed into the point (x', y') in the rectangle $B'C'H'I'$:

$$x' = \begin{cases} x\frac{r}{r-y} & \text{if } y \neq r \\ |\overline{BC}| & \text{if } y = r \end{cases}$$

$$y' = y$$

The points in the triangle CDH are transformed in an analogous way. Fig. 3(b) shows a set of points 1'-10', obtained by applying the transformation to the points 1-10 shown in Fig. 3(a).

The weighed average of the states of transformed particles is then mapped back into the original state space, using an inverse transformation easily derived from the above, to obtain the state estimate as shown in Fig. 3(c).

Note that in the inverse transformation all points of the section $G'I'$ are transformed into a single point H.

3. CONCLUSIONS

A navigation system for the blind was described and it was shown that particle filtering enables reliable state (position) estimation in the presence of constraints imposed on the state space. Particle clustering and convex mapping techniques were used to guarantee that the estimation results are at all times meaningful.

4. REFERENCES

[1] Arulampalam S., Maskell S., Gordon N., Clapp T.: *A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking*. IEEE Transactions on Signal Processing, 2002, vol. 50, no. 2, pp. 174-188, http://citeseer.nj.nec.com/arulampalam02tutorial.html.

[2] Ceranka S.: *Sensor Fusion Algorithms for Pedestrian Location*. Eighth International Conference on Methods and Models in Automation and Robotics, Szczecin, 2002, vol. 2, pp. 1343-1348.

[3] Gustafsson F., Gunnarsson F., Bergman N., Forssell U., Jansson J., Karlsson R., Nordlund P.: *Particle Filters for Positioning, Navigation and Tracking*. http://citeseer.nj.nec.com/gustafsson02particle.html.

[4] Milstein A., Snchez J. M., Williamson E. T.: *Robust Global Localization Using Clustered Particle Filtering*. http://citeseer.nj.nec.com/sanchez02robust.html.

[5] Niedzwiecki M., Zawalich L. *ARGO - a Navigation and Message System for the Blind*. Fourth International Symposium on Methods and Models in Automation and Robotics, Miedzyzdroje, 1997, pp. 687-691.