Location Sensor Fusion Mechanisms – A Comparison

Jörg Roth

Univ. of Applied Sciences Nuremberg
Kesslerplatz 12, 90489 Nuremberg, Germany
Joerg.Roth@FH-Nuernberg.de

1 Location Sensor Fusion Mechanisms

In order to detect the user's context or location, an application has to process position information from different sources (e.g. based on GPS sensors or cell IDs). A research field that traditionally deals with different position information is robotics. Based on e.g., odometers and ultrasonic distance measurement, a mobile robot computes the most probable location. Established approaches to compute a position from these sensors are Kalman and Particle Filters. Even though these approaches are widely used to process position sensor information, they often cannot be applied to mobile user scenarios due to significant differences: many positioning systems only have a limited coverage and availability and have a non-Gaussian measurement error distribution. Often, we receive a set of incomplete pieces of position knowledge such as:

- I currently receive WLAN cell ABC;
- checking my IP address, I know, I'm *not* at home;
- one minute ago, I received GPS position xy, and I have a speed of 5 km/h as pedestrian.

In the following we discuss important approaches to deal with such position information. All approaches have certain similarities:

- As positions based on measurements have a limited precision, statements about positions (e.g. to exactly reside at position xy) cannot be mapped to true or false. Thus, all approaches model position knowledge with the help of *probability densities* that map every position to the probability to reside at this position.
- All approaches deal with *multiple measurements at the same time* (the typical sensor fusion case) modelled by density operations. Multiple measurements are statistically represented by a multiplication of densities.
- All approaches deal with *movements*, i.e. how two points in time and the respective densities are related. Movements are statistically represented by a convolution of densities.

The approaches are very different, concerning how they represent probability densities and how they model multiple measurements and movements. The respective choices have significant influences on the usability in mobile user scenarios. In this paper, we discuss *Kalman Filters*, *Particle Filters* and *Geometric Approaches* considering the respective properties.

1.1 Kalman Filters

The Kalman Filter [5] is considered as one of the most important mathematical formalisms that deal with positioning. Detailed descriptions can be found in [3, 10]. The Kalman Filter assumes a state vector x_k (where k is a reference to the time) with arbitrary dimensions. For positioning tasks, the state vector contains typical spatial state information, i.e. the position, but also orientation, speed and acceleration. The state is unknown, but Gaussian distributed measurements z_k reflect information about the state. Two states x_{k-1}, x_k at two points in time are linearly related, expressed by a matrix A, which models movements. Further, the relation between state and measurement is expressed by a matrix H, which models the sensor characteristics.

A resulting probability density for x_k is Gaussian distributed expressed by a mean $\hat{x}_k$ (the most probably state) and an error covariance matrix P_k. A computation step contains a *Time Update Phase*, equations (1) and a *Measurement Update Phase*, equations (2).

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \qquad P_k^- = AP_{k-1}A^T + Q \tag{1}$$

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \qquad \hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \qquad P_k = (I - K_k H)P_k^- \tag{2}$$

After processing these equations for time k-1, we get more precise values for $\hat{x}_k$ and P_k. In these equations, we further have the terms Bu_k and Q that can be used to refine the time update model. For the statistical assumptions made for Kalman Filters, all relations can be expressed by few matrix multiplications and one matrix inversion. These computations can be performed efficiently even on small computers or embedded systems.

1.2 Particles Filters

Particle Filters [1, 2, 4] use a set of particles; each presents a specific potential state. A particle i contains a state vector p_i and a weight w_i which reflects the probability density for this state. In principle, a particle can have multiple dimensions, but in contrast to Kalman Filters, too many dimensions dilute to result. Typical state vectors only have three dimensions (e.g. 2D position and orientation).

The so-called *Motion Model* moves all particles according to their relative movement density. This indirectly computes a convolution. The *Perceptual Model* assigns new weights according to multiple measurements at a point in time. This indirectly computes multiplying densities. The computation steps are illustrated in fig. 1.

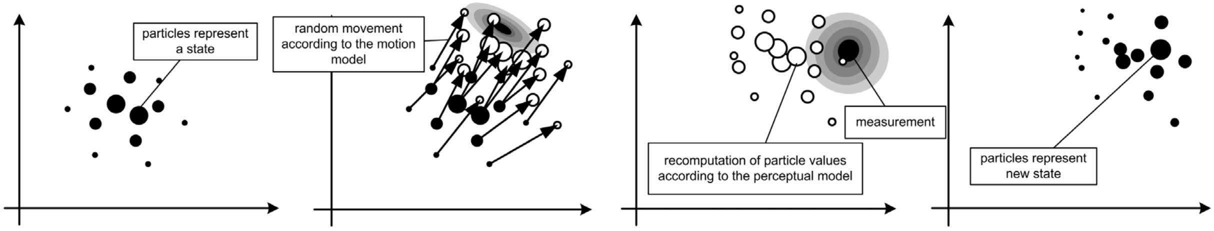


Fig. 1. Idea of Particle Filters

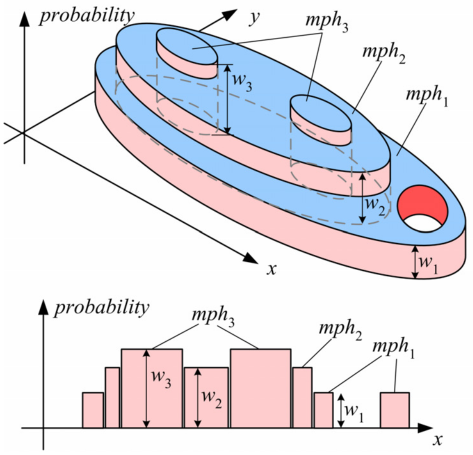
Particle Filters support a huge variety of densities. Increasing numbers of particles improve the precision, but also increase the required memory and processing time. Particle Filters also have to face the *degeneracy problem* where all but one particle have a weight near 0. Approaches such as *Sequential Importance Sampling* (*SIS*) and *Sampling Importance Resampling* (*SIR*) counteract this problem applying a resampling step to particles.

1.3 Geometric Approaches

Geometric approaches model a density and the respective model operations by geometric objects (typically polygons). The *Area model* [6, 8] could be viewed as a simple statistical representation where the area border separates two regions with a uniform distributed position probability – inside the area the integral of probabilities is 1, outside it is 0. The computations based on this idea are quite simple, using the geometric intersection, but this simplification significantly dilutes the position knowledge, especially for Gaussian distributed sensor information.

An innovative new approach is *MAP*[3] (*Multi-Area Probability-based Positioning by Predicates*) [9]. Any piece of information about the location at any point in time is mapped to a *location predicate* (e.g. "I'm not at home", "GPS measured position xy"). A predicate is in turn mapped either to a probability *density representation* (for a predicate that describes a location of a single point in time) or a *convolution operation* (for a predicate that describes movement between two points in time). When all predicates are processed, we get a set of densities – one for each considered time stamp. Depending on the application, the most probable location (the *centroid*) or a set of local maximum values can be computed.

Density operations (especially multiplication and convolution) are executed with the help of geometric operations widely known in the area of spatial modelling. For this, so-called *simple features* [7] are used for which efficient software libraries are available. From the variety of geometric objects, *MAP*[3] only requires the *multipolygon with holes* (*mph*) that represents the most common approximating two-dimensional structure. An mph contains a number of polygons representing the surface. Each of it in turn contains a number of polygons that represent the holes in the surface. Different probability levels are represented by a height value assigned to every mph. In addition, mphs of a density do not overlap. Fig. 2 shows how a density is represented in MAP[3] and the respective list of required mph operations.



density ops	mph ops							
	point in area	∪	∩	\	*surface area*	*buffer*	*centroid*	*dist.*
prob. of pos.	×							
resample density		×			×			
multiply densities			×					
convolution		×	×	×	×	×	×	
multiply negation			×	×				
centroid of density					×		×	
maxima of density		×					×	×
area with prob.>0		×						

Fig. 2. MAP[3] density representation

Whereas some operations can easily be executed on this representation, the convolution is crucial. But assuming some simplifications, this operation can be approximated using the so-called *buffer* operation [7] that computes all points that do not exceed a certain distance to an mph.

2 Requirements and Comparison

Position data in location- or context-based scenarios often are harder to process compared to data in e.g. robotics scenarios. This is a result of the specific characteristics of these data.

- As many approaches to determine the current position based on non-Gaussian distributions, especially on the *cell of origin* (*COO*), such information also has to be considered. Beyond a certain distance to an access point (especially outdoors), signal strengths do not significantly extend knowledge about the position [11], thus we often can only use the cell border to define the set of potential positions. Additional types of measurements (polygonal cells, measured direction or distance) lead to further densities as presented in fig. 3.

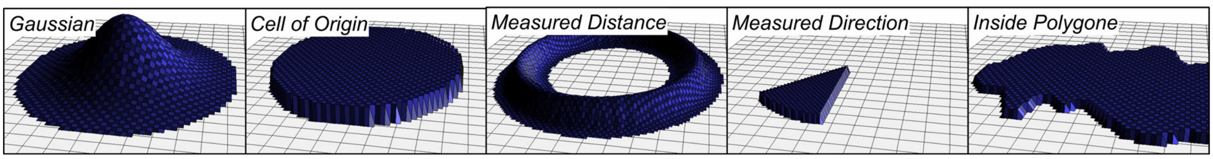


Fig. 3. Densities of different location measurements

- A reasonable approach must be able to consider multiple *alternative paths*. E.g., consider a user is at the crossroads to two streets going in nearly the same direction. We further assume that the position data is not precise enough to detect the choice. For a certain time, we thus have to consider both paths for potential positions until further information determines the choice.
- *Negated* information should be modelled. E.g., the current IP address may indicate that an end-user device does not reside at home. *Not* to reside somewhere means to reside nearly anywhere in the world and a nearly infinite space of potential positions has to be considered. In addition, negative statements about the position can be *uncertain*: this means the statement is not only affected by measurement errors but also can be wrong at all with a certain probability. As an example: not to receive a certain WLAN you may reside outside the cell, but also the access point may be switched off.
- *External spatial data* should be integrated. A huge amount of spatial information (e.g. roadmaps, information about places) may only be accessible over network. An appropriate algorithm must be able to download additional spatial data at runtime. This especially means to limit the search space for data lookup as not an entire spatial database can be downloaded.

We now check how existing approaches meet these requirements.

The Kalman Filter extracts the maximum information about the position, if the sensors fulfil the required statistical properties. Unfortunately, in some typical scenarios these presumptions are not valid: COO measurements do not provide a Gaussian density with a certain mean; negated information cannot be modelled at all; alternative potential paths cannot be modelled as the system always assumes a single probable position.

Particle Filters relax some of the statistical assumptions. Especially, they can follow alternative paths as some particles model one alternative and further particles a second one. However, if the number of parallel alternatives increases, the average number of particles decreases and thus the overall precision. With a sufficient (usually high) number of particles, COO measurements can in principle be modelled. The negation, however, is difficult to express if we do not limit the potential space. Finally, Kalman and Particle Filters have their problems with external data as they expect to access all spatial information locally.

Fortunately, the geometric approach, especially with MAP^3 meets the requirements above:

- COO measurements and non-Gaussian densities can easily be expressed by MAP^3 densities.
- Alternative paths can be expressed using multiple polygons each representing a path.
- MAP^3 introduced a special convolution operation that deals with negated and uncertain position statements.
- A spatial index for external databases can easily be derived from the representation using the area of non-improbable locations from the latest intermediary result.

Table 1 summarizes the characteristics.

3 Conclusion

Even though traditional approaches such as Kalman Filters and Particle Filters are widely used to deal with position sensor data, they have certain drawbacks. Especially in the area of location-based services and context-awareness they are often not suitable due to the specific characteristics of position data and the required operations.

New promising geometric approaches such as MAP^3 on the other hand meet many requirements: they are able to deal with negated and uncertain predicates, alternative paths and non-Gaussian densities that often appear in location- or context-based scenarios. In addition, spatial indices to access external spatial databases can easily be computed.

Table 1. Comparison of the different approaches

	Kalman Filter	Particle Filter	Geometric (MAP^3)
number of dimensions	unlimited	low	2D
space size	unlimited	small	unlimited
Gaussian distributions	yes	yes	yes
non-Gaussian	no	only if small size	yes
negated statements	no	no	yes
uncertain statements	no	no	yes
external data	no	no	yes
alternative paths	no	low	yes
compute centroid	easy	easy	easy
compute maxima	impossible	difficult	easy
multiplication	easy	implicitly solved	easy
convolution	easy	implicitly solved	difficult
area of potential pos.	not possible	difficult	easy
computational demands	low	low	medium

References

1. Doucet A., de Freitas N., Gordon N. (eds): Sequential Monte Carlo in Practice, Springer-Verlag, New York, 2001
2. Fox D., Thrun S., Burgard W., and Dellaert F.: Particle Filters for mobile robot localization. Sequential Monte Carlo Methods in Practice. Springer. New York, 2001
3. Grewal M. and Andrews A.: Kalman Filtering: theory and practice. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1993
4. Hightower J., Borriello G.: Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study, in Proc. of the 6th Int. Conference on Ubiquitous Computing (Ubicomp 2004), Sep. 2004, 88-106
5. Kalman R.: A new approach to linear Filtering and prediction problems. Transactions ASME Journal of Basic Engineering 82 (1960), 35-44
6. Leonhardi A., Kubach U.: An Architecture for a Distributed Universal Location Service. Proc. of the European Wireless '99 Conference, 1999, 351-355
7. Open Geospatial Consortium Inc.: OpenGIS® Implementation Specification for Geo–graphic information - Simple feature access - Part 1: Common architecture & Part 2: SQL option. John R. Herring (ed.), 2006
8. Roth J.: A Decentralized Location Service Providing Semantic Locations. Computer Science Report 323, Habilitation thesis, University of Hagen, Jan. 2005
9. Roth J.: Inferring Position Knowledge from Location Predicates, 3rd International Symposium on Location- and Context-Awareness (LoCA 2007) Sept. 20-21, 2007, Oberpfaffenhofen (Germany), to appear
10. Sorenson H. W.: Least-Squares estimation: from Gauss to Kalman. IEEE Spectrum, Vol. 7, July 1970, 63-68
11. Wang, Y., Jia, X., Lee, H. K. and Li, G.Y.: An indoor wireless positioning system based on WLAN infrastructure. 6th Int. Symp. on Satellite Navigation Technology Including Mobile Positioning & Location Services, Melbourne, Australia, 22-25 July 2003