Efficient Transmission of Uncertain Location Information

Christoph Behle*, Christian Hoene*, Andreas Krebs†, Mark T. Schmidt*
University of Tübingen, Sand 13, 72076 Tübingen, Germany
* christoph.behle|hoene|mark-thomas.schmidt@uni-tuebingen.de; † mail@krebs-net.de;

***Abstract*— Estimates of locations are never certain, especially in indoors environments, and it is useful not only to determine an estimate of measurement variables but also to know its uncertainty. In addition, location information is gathered at different places, devices, and sensors. This leads to the problem of transmitting uncertain location estimates efficiently across a network. We present multiple algorithms that transmit uncertain location information efficiently. Those algorithms reduce the amount of data, which is needed to convey the measurement results and their uncertainty, significantly.**

I. Introduction

GNSS receivers, which are able to determine their global positions, have a good but limited precision. Similar, indoor locating systems (IBS) provide position estimates of a rather large uncertainty as accuracies and precisions are low. This is due to the fact that IBS use sensors such as radio-frequency ranging and signal strength detectors, IMUs, and cameras, which are subjected to measurement noise and measurement errors. Thus, also the resulting location estimates are subjected to imprecision.

Luckily, many motion-triggered applications, location-based services (LBS), and sensor fusion algorithms do not require high precision and can cope well with the fact of uncertainty location measurements if they are aware of the degree of uncertainty.

Uncertainty can be presented in different forms. Typically, the output of filters such as Kalman or Particle filters are used to express the measurement estimate and its precision. Other forms include representations using Dempster-Shafer theory. Mathematically, all those formats can be expressed as a series of probability distributions $P_i(A)$ with i counting the observations and A being a vector. This vector contains coordinates that describe the position of an object and the time at which the object was located on the given position.

Frequently, the results of measurements are not available at the place where they are needed. Because of that, they need to be transmitted from a source (e.g. a filtered sensor or sensor fusion algorithm) to the sink (e.g. an app, a LBS, a cooperative sensor fusion algorithm, or a cooperative location determination).

Of course, if the communication link between source and sink is direct, even the results of Particle filters consisting of thousands of particles can be transmitted without loss in precision and without nearly any transmission delay. However, most of the time, this is not the case: As location tracking is mainly intended for mobile devices, communication is done wireless and thus, the communication link is subject of limited transmission bandwidth and transmission errors. Last not least, communication requires energy which is not that plenty on mobile devices or sensor nodes. Thus, the question arises on how to transmit the probability distributions P_i from source to sink effectively.

More precisely, encoding and decoding algorithms are needed that transmit a series of probability distributions while requiring

- a low bit and frame rate,
- a low algorithmic transmission delay, and
- moderate computational power
- without losing too much of information.

Not all coding algorithms can be superior in all of those parameters thus a set of algorithms is required that can be selected according to application specific requirements.

In this paper we propose a protocol and a set of algorithms for efficient exchange of uncertain geolocation information. We continue with describing related work in Section II. Then, we provide a problem statement and sketch algorithms on how to solve this problem (Section IV). Finally, we give an overview on scientific problems that need to be addressed before uncertain location information can be transmitted efficiently.

II. Related Work

Only a few publications address the problem of efficiently transmitting location information.

Harri et al. [1] describe a data format for transmitting geolocalization that modifies the World Geodetic System 84 (WGS84) [2] to use less overhead. To achieve this they encode the coordinates as well as the timestamps in only 16 bit instead of the 32 bit that would be default. That way they claim to reduce the overhead up to 71%.

Worral and Nebot [3] introduce a technique to automatically extract a compressed roadmap out of GPS data, so that the resulting map can be used on low power and low memory devices. To compress the data they use a clustering and linking method for the GPS data.

In a standardization draft, Hoene et al. presented a format how to encode uncertain geolocation information [4]. The authors assume that uncertain location data is represented by the output of Kalman-type, Gaussian-Sum, and Particle filters.

They used an XML presentation to transmit the filter outputs. No particular effort was spent on compressing this data.

III. Problem Statement

As described already in the introduction, we assume that location information is measured time sequentially and described with probability distributions called P_i. The function P_i expresses the probability distribution of the location of an object. As the location is time depended and time is also a measurement variable, it is reasonable to include both coordinates C and time t into the parameter A of the function P_i. We define A as $A = \begin{pmatrix} C \\ t \end{pmatrix}$.

The index i counts the sequential measurement observations. Each function P_i becomes available at $T_P(i)$ and is valid until $T_P(i+1)$, when P_{i+1} can be used.

We want to transmit the functions P_i from source to sink. At the sink, we receive probability distributions called $\tilde{P}_j$, which become available at time $T_{\tilde{P}}(j)$. The transmission process is lossy, so that the received probability distributions do not match P_i perfectly.

As a measure of similarity we consider the Canonical correlation between two segments piece-wise defined, time variable probability distributions $P(A)$ and $\tilde{P}(A)$ defined as

$$P(A) = P(\begin{pmatrix} C \\ t \end{pmatrix}) = P_i(A) : T_P(i) \leq t < T_P(i+1)$$

and

$$\tilde{P}(A) = \tilde{P}(\begin{pmatrix} C \\ t \end{pmatrix}) = \tilde{P}_i(A) : T_{\tilde{P}}(i) \leq t < T_{\tilde{P}}(i+1)$$

Then, the Canonical correlation can be calculated with the two vectors of independent variables X and Y where $X = \begin{pmatrix} A \\ P(A) \end{pmatrix}$ and $Y = \begin{pmatrix} A \\ \tilde{P}(A) \end{pmatrix}$. The Canonical correlation p is the result of the maximum of $cor(a'X, b'Y)$ with a and b being two vectors that need to be determined, too.

In this work, the problem to solve is to transmit P_i so that $\tilde{P}_j$ shows a maximal Canonical correlation p to P_i under the additional constrains of limited algorithmic transmission delay, limited transmission resource and of constrains computational resources.

IV. Ideas

In this section, we sketch our ideas on how algorithms should look like in order to solve the above mentioned problem.

A. Transmitting Filter States

Location tracking is based on physical measurements, which estimate time of flight, signal strength, angle of arrives, motions, objects in images, and many other forms of sensor input. All these sensor measurements are subjected to measurement noise. Because of that, filters are used to estimate the real value of the measurement.

Many filter types have been developed. In location tracking typically only a few are applied. These include different types of Kalman-Filters, filters that work with one or multiple Gaussian Normal distributions, and Particle Filters.

Our first idea is to define probability distributions P_i by the output given by filters. In order to transmit P_i, we just sent the filter state from source to sink. More precisely, the filter state and its interpretation depend on the kind of filter used.

For example, a Kalman Filter uses a system model to estimate the probability of changes. This data is combined with a model of measurement data and control input, if any, to estimate the true value of the parameters under study. It allows linear relations between filter variables and assumes Gaussian noise distributions. Kalman filters show a robust behavior in many applications and they and the related Extended or Unscented Kalman filters are frequently used.

The result of a Kalman filter is a posteriori state vector (for example a location) and a posteriori estimated covariance. The state is given by a vector $\hat{x}_{k|k}$ and the covariance by $P_{k|k}$. Both estimates are given for the time index i. If the vector has a dimension of d (e.g., four for x,y,z, and t), then the covariance matrix has a size of $d*d$. If we to transmit P_i, then we suggest that all these three variables shall be transmitted to indicate a position estimate, its distribution, and the time of measurement. More precisely, P_i is given by a Gaussian distribution have of the transmitted mean and covariance.

Particle Filter, also called sequential Monte Carlo methods (SMC), have the advantage that arbitrary distributions can be approximated. As such, they approximate Bayesian models, which consist of probability distribution functions, which define the degree of "believe" to which a particular value is true. Particle filters approximate probability distribution function with a number of particles. More particles are placed at positions that are more likely. Each particle has Dirac shape.

The a posteriori state of a particle filter is approximated by M particles called $x_i^{(M)}$, which are weighted with $w_i^{(m)}$. The function P_i is defined by all particles, their weights and again the time index i. At the receiver P_i is reconstructed by the particles that define a piece-wise probability defined distribution function.

Gaussian Sum Particle Filters have similar properties as Kalman and Particle filters. Their probability distributions are described by the sum of Normal distributions [5]. As such, it can be seen as the combination of two previously mentioned filtering approaches.

The a posteriori state of a Gaussian Sum Filter is approximated by M Gaussian distributions called $\hat{x}_k^{(M)}$, which are weighted with $w_k^{(m)}$ and have distributions described by covariance matrices $P_{k|k}^{(M)}$. Again, all those parameters shall be transmitted.

B. Coordinate System and Datum

Our second idea is to change the coordinate system and datum in order to reduce the number of bit that need to be transmitted.

Any location is relative to a frame of reference. The frame of reference defines the position, orientation and other properties of a coordinate system, in which an object is located. A number of geodetic reference frames have been defined such as WGS84, ETRS89, or ITRF2005. Typically, they define the reference point and the orientation of the coordinate system.

In robotics, reference frames are used, too. They are referencing to a zero point, have an orientation, and may be scaled, mirrored, rotated. For example, a so called transformation matrix can be applied to the location vector to transform coordinate systems.

Commonly in navigation, besides Cartesian also Polar coordinate systems are used. In addition, a polar coordinate system has the benefit that – for example – circular bands can be described easily if the rotating angles have a high uncertainty or are not defined.

To describe the location of an object, the reference frame has to be named or defined, and the type of coordinate system must be given. If this reference frame is transmitted only once, then the compression efficiency can be reduced as smaller number and a more compact representation of data can be used.

C. Demand driven precision

The third idea is to transmit data only at the precision that the source is requiring.

In the following we assume the following situation: we assume that there are two peers. One peer, in the following called receiver, wants to obtain uncertain geolocation data from the other peer, in the following called transmitter. The receiver might already have uncertain location data and be hence not interested in the full information known to the transmitter. The information is hold in form of Kalman filters. The receiver might be interested either only in locally defined parts of the data or only up to a certain precision. Our transmission format is designed to minimize the data transmitted. It can also reduce the data transmission if the peers want to exchange data as equal peers or if more than two peers participate.

In order to reduce the amount of data to transmit, the receiver informs the sender about the exactness he needs to approximate its locations and also a probability distribution of his guessed location. The exactness is transmitted as a quantity between 0 and 1, where 0 implies the receiver needs the exact data the sender has and 1 will be let the sender just send one Kalman filter that is an approximation of the data the receiver has with no promise on the quality. If Δ is between 0 and 1 the sender will transmit the data such that the receiver can for each position compute the probability of his location up to an error of Δ.

Without the need to transmit exact data the data stream can be significantly reduced. There are some observations that led to this compression:

- Any Kalman filter with a weight less than Δ/k does not need to be transmitted (assuming there are k filters in total).
- Any Kalman filter with a low weight and high distance from the probability area of the receiver can be ignored.
- If there are two Kalman filter with similar position and large distance from the probability area of the receiver these two Kalman filters can be approximated by a single Kalman filter.

Also the algorithm of the sender needs to be efficient in the sense that the sender might have only a limited amount of resources to compute the information he want to transmit.

D. Compression

The fourth idea is to explore two compression methods to transmit uncertain geolocation data. In the following, we describe two compression algorithms can be applied to Particle filters or Kalman filters:

Method 1: We send the complete list using standard compression techniques. This method is used if either the full list is requested or the list is too short to expect a gain using the Method 2 describe below. We choose the format in such a way that we can expect good compression if the positions and covariance matrices are similar. We can expect such similarities if the particles stem from a common source like a sensor.

Method 2: We send the data in a progressive based on a tree structure. This leads to the same benefits as the first method. Additionally, this allows an interactive protocol. The partial submitted data gives the receiver a preliminary view over the information available. This allows the receiver to request only parts of the data by asking only for parts of the tree and to stop the protocol if the data transmitted has the desired precision. The protocol falls back to the first method if the number of elements of a compartment falls below a threshold.

In the following, we first describe the format for Particle filters and explain later how the additional data for Kalman filters, i.e. the covariance matrix is transmitted.

1) Full list transmission: If we want to transmit the data in form of the list, we assume that the receiver and sender have an agreement on the following information: the bounding box of the particles transmitted in form of two vectors and the precision to which the position data should be transmitted. This information can be already known to both sides if this version is used within a larger protocol else the receiver transmits his wishes as part of the request. The transmitter sends only the particles within the bounding box up to the wished precision.

The information is sent by first transmitting the list of particles $x_i^{(M)}$, and then the list of weights $w_i^{(M)}$. The list of n-dimensional vectors is send as n lists of scalars, one list representing one dimension, instead of sending a list of n-tuples. To exploit correlation between the data we send scalar values $a_1, \ldots, a_l$ of k bits each by sending first the l first bits of $a_1, \ldots, a_l$, then the second bits of $a_1, \ldots, a_l$ and so on. If the data is correlated we can expect to have series of similar data which gives good compression rates using standard compression techniques like zip.

Besides relying on the compression we can also reduce the transmitted data for the following two reasons: since the most significant bits are determined by the bounding box this bits do not have to be transmitted for the points. Further, if we have sent the bits up to the requested precision we can stop the transmission.

2) Tree based approach: We use a tree approach to send the particles. Quadtrees were introduced [6] to store two-dimensional data. The extension to three dimensions is called Octree and often used in computer graphics.

The tree has $k = 2^d$ children per node, where d is the dimension of the $x_i^{(M)}$. The tree subdivides the space by splitting each dimension into two parts. This fixes a bit in the entry of $x_i^{(M)}$ for the corresponding dimension. If a node stands for a region its children stand each for one of the compartments of that region.

The tree is build up in the following way: the root contains all particles within the bounding box. While a node u contains more particles than a threshold t, split the corresponding area and the corresponding nodes as children to u. Then move the particles to their corresponding newly created child nodes of u; repeat.

For each node we will send the sum of the weights of the particles in that node. We call the sum of the weights of the particles contained in the compartment defined by a node u the weight of u. The root node has weight one. Assume that u has weight $\bar{w}$. To transmit the weights of the k children of u, we do not transmit k weights, but only the differences between the actual weight w of a child and the average weight $\bar{w}/k$. This means we transmit $\Delta w = \bar{w} - w/k$.

If we have reached a leaf node we send the particles for that compartment as a full list. In this case not w_i but Δw_i.

The partial received information gives blurred information contained in the list. This partial information can be already enough information for the receiver to decide which region might be worth of further transmission. Sending back his area of interested the transmitter needs not to send the complete tree but can send only those parts that contain more information the receiver is interested in. Further it allows to stop transmission if the receivers has gained the amount of precision he is interested in. We can also expect a reduction of the data sent if the points in $\vec{p}$ form clusters. If several points lay within the same region of the tree descending into that region specifies the leading bits for this points simultaneously. These bits do not have to be transmitted if the list of these points is transferred.

If we transmit data from Kalman filters we cannot directly average over the particles. Instead, for each compartment we additionally transmit two normal distributions by $x_{min}, w_{min}, r_{min}$ and a particle filter by $x_{max}, w_{max}, r_{max}$. The covariance matrices are given by multiplying the identity matrix with r_{min} and r_{max} respectively. The two distributions shall fulfill the following constraint: for any position the effects of the Kalman filters from the compartment are bounded by the two normal distributions. This allows estimating the effects of the particle filters in this compartment on the other compartments and the receiver to decide if he wishes these particles to be transmitted or not.

V. Conclusion & Outlook

This initial work presents ideas of transmitting and compressing uncertain, relative and transformed location estimates. It is aimed as a basis of first scientific discussions, because we believe that much more research has to be done before uncertain location information can be transmitted and compressed efficiently.

Our next steps will include collection of real, measured data set of Kalman and Particle filters used for indoor locating systems. After that, we will compress them with the methods described in this paper and we will measure the compression quality and effectiveness.

Acknowledgements

This research has been financed in parts by the Baden-Württemberg Stiftung in the BW-FIT project AmbiSense.

References

[1] J. Harri, F. Filali, and C. Bonnet, "Rethinking the overhead of geo-localization information for vehicular communications," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, Oct. 2007, pp. 2111 –2115.

[2] Wikipedia, "World geodetic system — wikipedia, the free encyclopedia," 2011, [Online; accessed 7-April-2011]. [Online]. Available: http://en.wikipedia.org/w/index.php?title=World_Geodetic_System&oldid=420991993

[3] S. Worrall and E. Nebot, "Automated process for generating digitised maps through gps data compression," in *Australasian Conference on Robotics and Automation, Brisbane, Australia*, 2007.

[4] C. Hoene, A. Krebs, C. Behle, and M. Schmidt, "Bayesian location identifier," IETF draft-hoene-geopriv-bli-00.txt, work in progress, Oct. 2010. [Online]. Available: http://www.ietf.org/id/draft-hoene-geopriv-bli-00.txt

[5] J. Kotecha and P. Djuric, "Gaussian sum particle filtering," *Signal Processing, IEEE Transactions on*, vol. 51, no. 10, pp. 2602 – 2612, Oct. 2003.

[6] R. A. Finkel and J. L. Bentley, "Quad trees: A data structure for retrieval on composite keys," *Acta Inf.*, vol. 4, pp. 1–9, 1974.