

Position Estimation from UWB Pseudorange and Angle-of-Arrival: A Comparison of Non-linear Regression and Kalman Filtering

Kavitha Muthukrishnan¹ and Mike Hazas²

¹ Pervasive Systems Group, Department of Computer Science
University of Twente, The Netherlands.

² Computing Department, Lancaster University, UK.

Abstract. This paper presents two algorithms, non-linear regression and Kalman filtering, that fuse heterogeneous data (pseudorange and angle-of-arrival) from an ultra-wideband positioning system. The performance of both the algorithms is evaluated using real data from two deployments, for both static and dynamic scenarios. We also consider the effectiveness of the proposed algorithms for systems with reduced infrastructure (lower deployment density), and for lower-complexity sensing platforms which are only capable of providing either pseudorange or angle-of-arrival.

1 Introduction

While GPS is a common solution for outdoor localisation, indoor localisation still remains an open research problem. Location systems based on conventional radio technology have relatively coarse-grained performance indoors because their signals typically cannot be resolved accurately enough to produce quantities such as a time-of-arrival or angle-of-arrival. Instead, received signal strength indication (RSSI) or other metrics are used to estimate location via a technique called *fingerprinting*, but positioning errors of several metres often result because of indoor multipath fading. By contrast, ultra-wideband (UWB) systems can employ pulses of extremely short duration to achieve much finer signal resolution. This resolution also aids in the identification of erroneous measurements due to multipath [3].

UWB positioning systems can measure time-of-arrival, time-difference-of-arrival (TDOA) or angle-of-arrival (AOA), or some combination of them. A time-of-arrival can be converted into a range estimate, but measuring time-of-arrival directly is problematic, because the synchronisation signal (typically conventional radio) and the UWB positioning pulse both travel at near the speed of light.¹ Pseudorange (based on TDOA measurements) is more attractive for some deployments, since there is no need for precise synchronisation between the transmitting and receiving entities. A network of re-

¹ If a wired tether were used to connect the receiver and transmitter, direct range measurement would be possible after calibration of the timing offset due to the tether. Direct range measurement using round-trip time-of-flight measurements are possible for devices capable of both transmitting and receiving UWB positioning pulses.

ceivers can be precisely synchronised using stable clocks which are periodically corrected via a wired reference timing signal [14]. Producing AOA estimates requires a receiver equipped with an antenna array.

One UWB positioning system [2] estimates a tag's location using pseudorange data, with a stated accuracy of about 30 cm. Ubisense is a commercial UWB-based location system which performs measurement of both pseudorange and AOA (azimuth and elevation). The advantage of using AOAs as well as pseudoranges is that location can be determined with fewer sensors, compared to systems that use just pseudoranging. The reported accuracy (using their proprietary algorithms) is 15 cm.

This paper examines the results of two positioning algorithms operating on pseudorange and AOA data. The four primary contributions of this paper are as follows: (I) a brief characterisation of the raw observations reported by two different Ubisense deployments (Sect. 3); (II) formulation of two algorithms (regression using a non-linear model, and Kalman filtering; Sect. 4) which fuse the heterogeneous observations (pseudorange, azimuth, elevation) of a UWB system; (III) a characterisation of the static (Sect. 5) and dynamic (Sect. 6) tracking performance of the two algorithms; and (IV) an evaluation for reduced sensor densities and for pseudorange-only and AOA-only data, showing that the algorithms can support minimal deployments and less sophisticated UWB positioning sensors.

2 Related work

Three approaches are generally used to calculate location using range, pseudorange, or AOA estimates. The first approach uses a simple geometric model to calculate intersection of circles (lateration), hyperbolas (hyperbolic localisation) or lines (angulation), depending on whether range, pseudorange, or AOA data is used. However, such simple algorithms typically do not take measurement error into account, and cannot make optimal use of redundant data (such as that gathered from a large number of receivers) which overspecify the solution.

By contrast, a second approach is to use optimisation algorithms which are specifically designed to find a solution which minimises the total error between the collected data and the location estimate (i.e. the *residual error*). These algorithms traverse the solution space and compute expected measurements for each estimate of the solution. The *gradient* method employs derivatives to observe the rate at which an area of the solution space converges towards an optimum. Some examples in this category are the method of steepest descent, Newton's method, and the Levenberg-Marquardt method [8]. Such methods require *model equations* which are used to express the measured values (such as pseudorange, or AOA) in terms of the position being solved. The Bat system [15], and other ultrasonic positioning systems [1] apply such model-based optimisation algorithms to compute location solutions from ranging data.

Scott et al. utilise a non-linear system of equations to compute location based on pseudoranges (in this case TDOAs of acoustic signals) [11]. Fontana comments on the performance differences between the steepest decent search and Davidon-Fletcher-Powell algorithm for estimating a tag's location using UWB pseudorange data [2]. It is

worth noting that these error minimisation algorithms do not make use of information obtained from prior location readings.

Algorithms that utilise a solution state (either current state, or current and past states) can be grouped under the third approach of *state-estimation algorithms*. State-estimation algorithms are used extensively in robot localisation. They operate by iteratively combining the previous estimate of the state (e.g. a position and orientation) with the observed measurements (range, AOA, etc.). Many state estimation algorithms exist [4], of which Kalman filtering [16] is most commonly used. The HiBall tracking system [17] employs a technique called single-constraint-at-a-time (SCAAT) tracking to model movement, and handles one observation at a time rather than obtaining multiple simultaneous measurements. The Constellation system [5] tracks a mobile unit consisting of a 3D inertial sensor and a number of ultrasonic sensors which report ranges that are fed into a SCAAT algorithm. The idea is to correct the positional drift in inertial tracking by incorporating ultrasonic range measurements in an extended Kalman filter. Smith et al. present a tracking algorithm using extended Kalman filtering of ultrasonic range data gathered from “Cricket” devices [12]. Smith et al. employ a combination of least squares minimisation, Kalman filtering and outlier rejection to predict the state of the Cricket device. While their approach is similar to our proposed Kalman filtering, we fuse heterogeneous data (pseudoranges and angles-of-arrival), and operate on UWB rather than ultrasound measurements.

Recent work by Renaudin and Kasser demonstrates the fusion of pseudoranges and AOAs from a Ubisense system together with inertial sensing data using an extended Kalman filter [9]. Curiously, their modelling equations are quite different to ours presented below (Eqns. 1–4) which we have verified through the analysis in this paper. Moreover, the goal of Renaudin and Kasser is the *augmentation* of inertial tracking with UWB sensing. There is no literature which explores in detail the fusion of heterogeneous data (pseudorange and AOA) through regression or Kalman filtering for UWB position sensing.

3 Deployments

In all our experiments, we used hardware and software procured from Ubisense. In brief, Ubisense hardware is comprised of two entities: a tag² which emits UWB pulses when triggered by the system; and receivers (“Ubisensors”) which are typically fixed at upper extremes of the measurement volume. The receivers are networked via CAT5 cabling and it can be assumed that they are tightly synchronised (once the cable timing offsets are estimated during calibration). A workstation PC is connected to the same network, and runs the Ubisense Location Engine (LE), which can be used to configure and calibrate the system, and which produces location estimates from receiver measurements, using proprietary algorithms.

The calibration of the receivers’ position and orientation are crucial in achieving accurate location estimates. The coordinates of a receiver’s position can be estimated

² Ubisense offer both a “slim” and a “compact” tag. We chose to use the compact tag for the data collection. It is advertised that the tag emits pulses in an omnidirectional fashion, and it performed as such during our initial, informal experiments.

with sufficient accuracy via manual methods, such as measuring the distance (using a tape measure or laser rangefinder) from the receiver centre to several known points in the environment. By comparison, accurate estimation of receiver orientation³ (yaw and pitch) is more difficult without special equipment, and additionally there can be small misalignments between the plastic casing of the Ubisensor, and the plane of the UWB receiver array inside. Thus, calibration of receiver pitch and yaw is normally undertaken using a series of measurements from a tag; the measurement logging and orientation estimation process is automated by the Ubisense Location Engine. To calibrate the receiver pitch and yaw for the systems at our sites, we used two different modes of the automated process, as described below.

3.1 Twente

Six receivers were deployed covering an area of approximately 15×9 m (Fig. 1(a)). Four of the receivers were deployed in an area which was relatively empty except for some desks along the corners of the room. The remaining two receivers were placed in the rooms along the other side of the corridor, in typical office spaces containing desks, metal shelving, and other furniture. We arbitrarily chose twenty-one test points across the measurement volume. Measurements were taken with tags placed at two different heights (75 cm and 151 cm above the floor). The positions of the sensors and the test points were surveyed using tape measures.

Automatic calibration. At Twente, we used the “automatic calibration” mode of the Ubisense LE. In automatic mode, a tag is placed at a position within the measurement volume. The height of the tag must be known and provided to the LE by the user, but the system then takes measurements to estimate the horizontal position of the tag, and the pitch and yaw of those receivers which can reliably detect the tag’s pulses. The tag is then moved to another position and this process is repeated until the pitch and yaw of all receivers is estimated.

3.2 Lancaster

At another site, a smaller measurement area of 2.75×2 m was covered with five receivers. Readings were taken with a stationary compact tag placed at four heights (about 0, 7.5, 75, and 125 cm above the floor), at sixteen points across the measurement area (Fig. 1(b)). The positions of the sensors and the test points were surveyed using a Leica Total Station.⁴

Dual calibration. At Lancaster, the LE “dual calibration” mode was used to estimate receiver pitch and yaw. In this mode, a tag is placed at a known position in the environment, and its 3D coordinates are provided to the LE. Measurements of the tag’s pulses are then gathered from two receivers, and the receivers’ pitch and yaw are estimated for one or both of the receivers (selectable by the user). The dual calibration has

³ Ubisense use the aeronautical terms *yaw*, *pitch* and *roll* to describe receiver orientation [13].

⁴ A Total Station is a professional surveying device which accurately measures the range, azimuth, and elevation between itself and a reflector, and which can thus be used to estimate the 3D coordinates of the surveyed point.

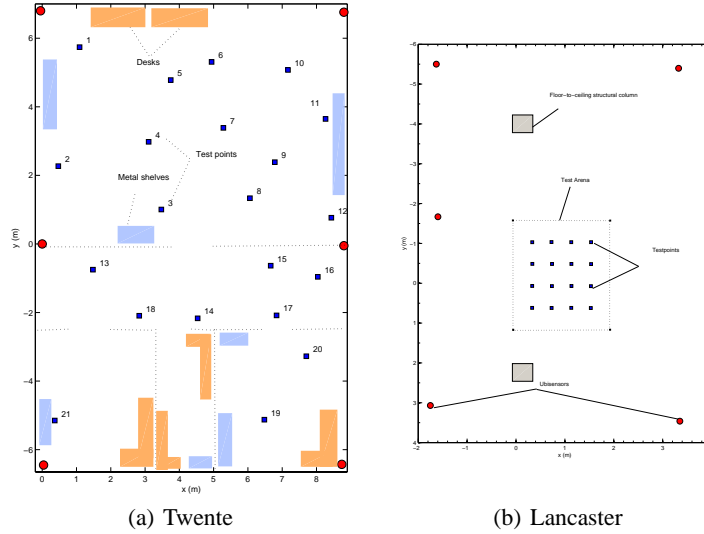


Fig. 1. Plan views of the deployment areas

the potential to more accurately estimate pitch and yaw, since the estimation process does not depend upon the accuracy of the LE's result for the tag's horizontal position (as it does in the automatic calibration mode). UbiSense recommend that when performing calibration, the tag be placed as near as possible to the boresight of the receiver(s) being calibrated. At Lancaster, we ran the dual calibration for all five receivers at each of twelve known points (whose coordinates had been surveyed using the Total Station) in or near the test arena. The twelve calibration points (which are different from the test points used in our evaluation) were each selected for their favourable line-of-sight to at least four of the five receiver units. To remove poor pitch and yaw estimates (typically due to poor line-of-sight or environmental reflections), we took the median of the twelve pitch and yaw values for each receiver as our final calibration.

3.3 Characterisation of raw measurements

Fig. 2 shows the distributions of the receivers' raw measurements for a static tag placed at the test points at each site (sixty-four locations at Lancaster, and forty-two locations at Twente). The accuracy of the Lancaster measurements is significantly better than that of the Twente measurements (Table 1). The Twente deployment covers a wider area, and not all receivers have line-of-sight to all the test points. In its measurement and location estimation process, the LE identifies receivers whose measurements were rejected for the purposes of computing a location; the rejected measurements typically correspond to receivers which had poor line-of-sight to the tag (and may therefore represent multipath readings). We plot the raw accuracy with these rejected measurements excluded. In total, the discarded measurements represent 25% of the Twente dataset.

Even with the LE-rejected measurements excluded, the data from the Twente deployment is significantly worse in accuracy than the Lancaster data. In general, there

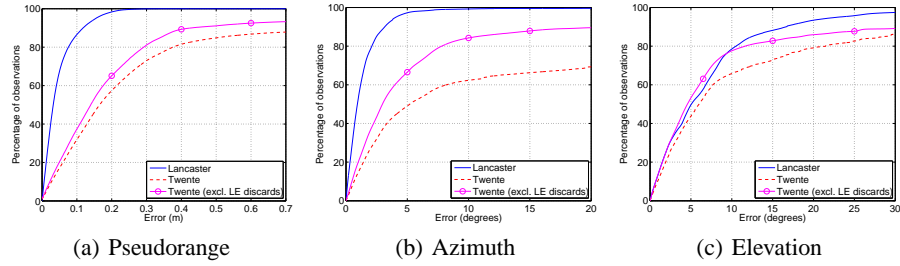


Fig. 2. Raw measurement error distributions

	Pseudorange	Azimuth	Elevation
Lancaster	6 cm	1.5°	9°
Twente	37 cm	22°	17°
Twente (excl. LE discards)	25 cm	6.8°	8°

Table 1. Seventy-fifth percentile accuracy of the raw measurements of two Ubisense deployments

are three sources of raw measurement error in any location system: (a) sensor inaccuracy; (b) calibration inaccuracy (sensor position, pitch and yaw), and (c) inaccuracies induced by environmental effects (attenuation and/or reflection of the UWB pulse). The Lancaster deployment was more carefully calibrated—a Total Station survey as opposed to hand measurements; and dual calibration of all receivers at twelve surveyed points, as opposed to automatic calibration at several arbitrary points. Moreover, the Lancaster deployment was specifically designed to accurately monitor the measurement volume; all five receivers have favourable line-of-sight to the majority of the sixty-four test points. By contrast, the Twente deployment has a much lower receiver density, covering a measurement volume containing office furniture and walls. For these reasons, we would argue that the inaccuracy of the Twente raw measurements is due to a poorer calibration (visible in the significant error offsets for some of the Twente receivers), and less favourable line-of-sight and multipath conditions caused by the environment.

In summary, the two sites represent very different kinds of deployment. The Lancaster deployment has been carefully calibrated using specialised equipment and system-specific knowledge, and has been designed to monitor a small volume using a high sensor density. The Twente installation covers a much larger volume, the environment was left unmodified (leading to a higher degree of unfavourable pulse propagation for some tag locations), and sensor position and orientation have been calibrated based on fewer measurements taken with non-specialised, less accurate equipment.

4 Overview of Algorithms

4.1 Non-linear regression

We formulate a non-linear regression algorithm [8, section 15.5] which utilises modelling equations. The regression is an iterative process which finds an estimate of the

tag's location. The estimate can be seen as a "best fit" for the pseudoranges and/or angles-of-arrival and the surveyed receiver locations and orientations, since it minimises the sum of squares of the residual errors.

Pseudoranges. For a receiver i , the receiver's pseudorange estimate (based on the measured relative arrival time of the pulse emitted by the tag) \tilde{d}_i , the receiver's 3D location (x_i, y_i, z_i) , and the tag's location (u, v, w) can be related as follows:

$$\tilde{d}_i = \sqrt{(u - x_i)^2 + (v - y_i)^2 + (w - z_i)^2} - d_c, \quad (1)$$

where d_c is the distance offset common to all receiver pseudorange measurements, and arises from the tag's unknown clock offset from the system. It is assumed that the receivers are very tightly synchronised, or their reported pseudoranges have been otherwise appropriately adjusted (using for example deployment-specific calibration information) for any time offset which may exist between receiver units.

Angles-of-arrival. The receiver-reported azimuth ϕ_i of a tag is related to the tag's position (u_s, v_s, w_s) in the receiver's frame of reference.

$$\text{Quads. I \& IV: } \phi_i = \arctan\left(\frac{v_s}{u_s}\right) \quad \text{Quads. II \& III: } \phi_i = \left(\pi + \arctan\left(\frac{v_s}{u_s}\right)\right) \quad (2)$$

Similarly, the elevation θ_i of a tag measured by a receiver is defined as

$$\theta_i = \arctan\left(\frac{w_s}{\sqrt{u_s^2 + v_s^2}}\right) \quad (3)$$

Note that a four-quadrant definition of the elevation θ_i is unnecessary, since the denominator of the arctangent operand is always positive.

To compute the tag location in the receiver's frame of reference, first the coordinates of the tag $(u_{\text{rel}}, v_{\text{rel}}, w_{\text{rel}})$ relative to the receiver are computed by subtracting the receiver coordinates from the coordinates of the tag in the global frame of reference, i.e. $(u - x_i, v - y_i, w - z_i)$. This effectively translates the origin of the global coordinate system to the location of the receiver. These tag-to-receiver relative coordinates must be transformed to the receiver's frame of reference. This is first a rotation about the Z axis by an amount corresponding to the receiver's yaw φ_i , followed by a rotation about the Y axis by an amount corresponding to the receiver's pitch ϑ_i . The rotation matrix R_{GS} can thus be defined as:

$$R_{\text{GS}} = \begin{bmatrix} \cos(\vartheta_i)\cos(-\varphi_i) & -\cos(\vartheta_i)\sin(-\varphi_i) & \sin(\vartheta_i) \\ \sin(-\varphi_i) & \cos(-\varphi_i) & 0 \\ -\sin(\vartheta_i)\cos(-\varphi_i) & \sin(\vartheta_i)\sin(-\varphi_i) & \cos(\vartheta_i) \end{bmatrix} \quad (4)$$

Note that Ubisense yaw is defined as positive anticlockwise (looking at the XY plane from the Z+ direction), and Ubisense pitch is defined as negative anticlockwise (looking at the XZ plane from the Y+ direction). The position of the tag (u_s, v_s, w_s) in sensor's frame of reference is then computed by multiplying the tag's coordinates relative to the receiver $(u_{\text{rel}}, v_{\text{rel}}, w_{\text{rel}})$ by R_{GS} .

Outlier rejection. The number of observations required (pseudoranges and/or angles-of-arrival) depends upon the particular tag-to-receiver geometry. In typical Ubisense deployments, to estimate both tag location and the distance offset \hat{d}_c at minimum either two pseudoranges, one azimuth, and one elevation, *or* four pseudoranges are needed. To estimate only tag location, at least three angles-of-arrival (including at least one azimuth and one elevation) are needed. For locations estimated using more than the minimum number of observations, the standard error s of the estimate can be calculated from the residual errors e_i :

$$s = \sqrt{\frac{\sum_{i=1}^I e_i^2}{I - C}}, \quad (5)$$

where I is the number of observations being used in the non-linear regression, and C is the minimum number of observations required. C is set to 3 if only tag location is being estimated, or to 4 if the distance offset \hat{d}_c is also being estimated. The non-linear modelling process assumes the residual errors $\epsilon_{i=1\dots I}$ are normal, independent, and have equal variance and zero mean.

With heterogeneous data such as that reported by Ubisense receivers (pseudorange, azimuth, and elevation), the computed residuals have different units. Thus, before applying Eqn. 5 to estimate the standard error, each residual should be scaled by a typical magnitude of error which might be expected for each type of data. To perform scaling of residuals, we utilised divisors of 30 cm, 4° and 3° for pseudorange, azimuth, and elevation, respectively. These were chosen to work with both the Twente and Lancaster datasets, rather than being specifically tailored to either one.

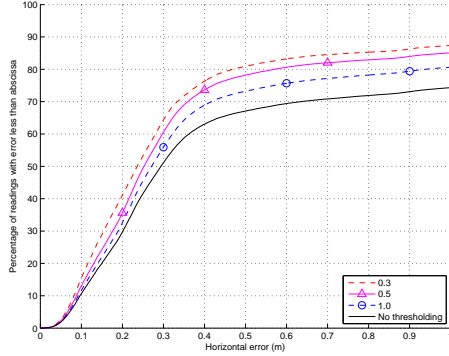
If the observations given to the regression do not corroborate one another, the result will typically have a high standard error estimate. Standardised residuals [6, ch. 4] can be used to identify the observations which agree the least with the solution. If the standard error estimate was greater than 0.30, our regression algorithm removes the observation with the highest standardised residual, and re-computes the solution. This process continues until the standard error of the estimate falls below 0.30, or the highest residual is less than 1.0.

Even after the outlying observations have been removed, the location accuracy can still be poor (Fig. 3(a)). To improve accuracy, one tactic is to simply reject location readings which have a high standard error estimate. For the regression results presented in the remainder of this paper, we have omitted estimates with a standard error higher than 0.5 (except where otherwise noted). Assuming both pseudorange and angle-of-arrival data is used, this corresponds to about 20% of the readings taken at Twente (Fig. 3(b)), and very few (less than 0.1%) of the static readings taken at Lancaster.

4.2 Extended Kalman filtering (EKF)

We formulate an Extended Kalman Filter (EKF) using a state vector \hat{x}_k with seven variables, three position variables u, v, w , user clock offset d_c and three velocity variables v_u, v_v, v_w . After any discrete time step, which is approximately 108 ms⁵ the filter has an

⁵ The tag emits a UWB pulse once every four Ubisense time slots. With our particular version of Ubisense hardware, each time slot corresponds to 27.029 ms.



Std. error threshold	75% horiz. error (cm)	Fraction accepted (%)
0.3	38.4	64.2
0.5	42.1	80.4
1.0	57.0	91.2
None	106.4	100

(a) Horizontal error distribution

(b) Trade-off of accepting regression solutions based on standard error

Fig. 3. Horizontal accuracy at Twente for different standard error thresholding levels

idea of its state and how confident it is in that state (prediction). The filter then corrects the predicted state based on the most recent measurements (pseudorange and angles) and its internal state. Note that we use an Iterative Extended Kalman Filter (IEKF) [7], which is an extension to the standard EKF, and is useful for reducing the errors that may occur due to large non-linearities in the system.

Initialisation. The filter is initialised with a posterior state estimate \hat{x}_k^- and uncertainty P_k^- . Kalman filter estimates rely heavily on these initial estimates. We set the initial state estimates based on averaging the first twenty measurements through non-linear regression (as described above), discarding any with high standard error. Since we have set the tag at the highest update rate, this corresponds to a wait of a few seconds to start the Kalman filter. We chose to use a small but non-zero value for P_k^- , meaning that there is a little uncertainty in the defined initial state.

System Model and Measurement Model. We use a *constant-velocity model*, i.e. it is assumed the tag moves at constant speed between time steps. Thus, the new state estimate \hat{x}_k will depend on the previous state estimate \hat{x}_{k-1} , constant velocity v_{xk} and a noise term w_k (as in, $\hat{x}_k = \hat{x}_{k-1} + v_{xk} dt + w_k$). In order to predict the state using the measurements, we will have to describe how the measurements are related to the state. The *measurement model* $\hat{z}_k (= H\hat{x}_k + v_k)$ describes how measurements depend on the state estimates \hat{x}_k . H is the Jacobian matrix with partial derivatives of the measurement function with respect to the state \hat{x}_k . The measurement function here represents the pseudorange (\tilde{d}_i), azimuth (ϕ_i) and elevation (θ_i) as defined earlier in Sect. 4.1.

Prediction & Correction. The predicted error covariance (P_k^-) and the state estimate (\hat{x}_k^-) for a time-step is given by:

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k \\ P_k^- &= AP_{k-1}A^T + Q. \end{aligned} \quad (6)$$

Here, A is the Jacobian matrix and Q is the process noise covariance. The process noise covariance Q for a position-velocity model includes the process covariance in position and velocity and the process covariance in the position.

The filter computes the posterior state estimate by taking the prior state estimate and combining with the Kalman gain K_k times the difference between the actual measurement (pseudorange and angles) and a measurement prediction ($\hat{z}_k = H\hat{x}_k + v_k$), called the *innovation* or residual r . If the innovation is zero, then the predicted state estimate exactly reflects the real measurement. But if there is a difference between the predicted and the observed measurement, then the prior state estimate needs to be updated. In eq. 7, K_k determines to what extent the innovation should be used in the posterior state estimation. Based on the measurement noise R and the prior error covariance P_k^- , the gain can favour the innovations or the measurements more. The measurement noise R for pseudorange and angles (azimuth and elevation) is set to be 10 cm, 7° and 10° respectively.

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ \hat{x}_k^{i+1} &= \hat{x}_k^- + K_k (z_k - H(\hat{x}_k^- - \hat{x}_k^i)) \\ P_k &= (I - K_k H) P_{k-1}^- \end{aligned} \quad (7)$$

In Eqn. 7 the Jacobian matrix H is evaluated at the most recent intermediate state estimate \hat{x}_k^i (difference between the IEKF and EKF). After a number of iterations or when the intermediate state estimate does not differ with more than a certain threshold from \hat{x}_k^{i-1} , the filter sets the posterior state estimate and estimates its posterior uncertainty. It is important to note that the IEKF computes the uncertainty in the state only after it finds the most accurate intermediate state estimate. Though the computations involved in IEKF are larger than the standard EKF, the state estimates will be better because of re-evaluation of the measurement function and the Jacobian.

Validation gating. It is imperative to employ some form of outlier rejection (also known as validation gating in Kalman literature) as part of the filter, since noisy measurements can cause the filter arrive at a bad state estimate. In most cases, innovation or residual r is used to identify outliers. An approach to eliminate outliers is based on $r^2 S^{-1} > \gamma$, where S^{-1} is a scalar based on the state and γ is an empirically-chosen parameter [12]. Alternatively, the distribution of innovations can be used to detect innovations that are unlikely to occur. Recent work by Renaudin looks at the possibility of eliminating outlying measurements based on human body orientation and comparing the predicted state with the current measurements. Another interesting approach is to check the condition for *optimality* of the filter. It is reported [10] that through the usage of statistical methods it is possible to check if the innovation sequence is *white* (a sufficient and necessary condition for testing optimality). If the filter shows sub-optimality, R and Q can be adjusted in order to make the filter optimal, thereby making the Kalman filter *adaptive*. We however chose to use the simple strategy of using the innovation distribution (based on setting thresholds determined empirically) to identify outlying measurements.

5 Static Positioning

For the static positioning experiments at Lancaster, a tag was placed at the sixteen test points (Fig. 1(b)), at four different heights. At each of these sixty-four locations, receiver readings were gathered for about one thousand tag pulses and about 1000 readings were gathered. At Twente, a tag was placed at the twenty-one test points (Fig. 1(a)) at two different heights, for a total of forty-two locations, with about 1500 readings at each. For our experiments at both sites we set the tags to run at the highest update rate, emitting a UWB pulse once every four Ubisense time slots, which for our Ubisense hardware is equivalent to about ten times per second. All receiver observations were fed into our algorithms, even those rejected by the Ubisense LE (see Fig. 2).

5.1 Results from heterogeneous observations

Fig. 4 shows the accuracy of the regression and Kalman filtering algorithms when they are provided with the heterogeneous measurements (pseudorange and angles-of-arrival) produced by the receivers. Note that the vertical accuracy at Lancaster is more than thirty centimetres worse than the horizontal accuracy. This might be explained in two ways. First, the geometry of the Ubisense deployment (receivers are approximately coplanar) and the types of measurements they take (pseudorange, azimuth, and elevation) mean that there is more information about the tag's horizontal position solution (heavily contributed to by pseudorange and azimuth), and less about the vertical position of the tag (primarily affected by the elevation readings). Second, referring to Fig. 2, one can see that the pseudorange and azimuth accuracy tend to be better than the elevation accuracy. At Twente, the opposite is true: the vertical accuracy tends to be superior to the horizontal. However, about 30% of the Twente raw azimuth readings are off by more than 20° , whereas the elevation accuracy is comparable to that at Lancaster (Fig. 2(c)).

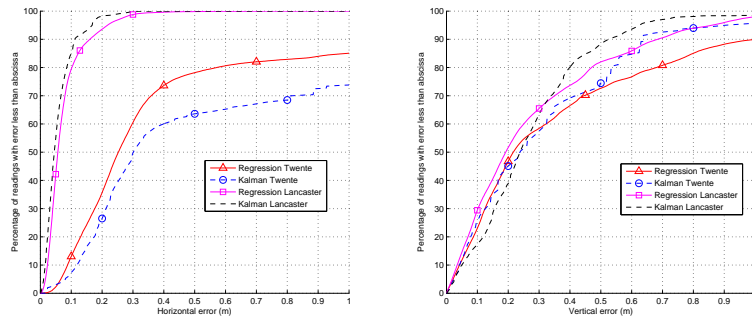


Fig. 4. Positioning static tags using heterogeneous measurements (pseudorange and AOA)

On the Lancaster data, the Kalman filter performs slightly better than the regression. This is because the Kalman filter utilises its predicted estimate for the position

Algorithm	Site	75% confidence level (cm)		90% confidence level (cm)		Estimates accepted (%)
		Horizontal	Vertical	Horizontal	Vertical	
Regression	Twente	42.07	54.74	155.59	101.04	80.42
Kalman	Twente	107.82	51.86	276.95	63.72	N/A
Regression	Lancaster	8.83	42.13	16.02	68.46	99.98
Kalman	Lancaster	7.51	37.35	11.14	52.61	N/A

Table 2. Performance summary of static positioning using heterogeneous data

of the tag, in addition to new receiver observations, to produce the new tag estimate. By contrast, the regression algorithm utilises only the receiver observations from the current time slot to compute a solution. Thus the Kalman filter can be slightly more accurate, especially for static situations where the location estimate is stable for consecutive observations. Note that for the Twente data, the Kalman filter has much worse horizontal error than the regression. However, this is because the regression rejects results with a high standard error estimate. Rejecting no readings (Fig. 3), the regression’s 75% horizontal accuracy of 106 cm is comparable to the Kalman filter’s 108 cm.

Effect of reduced receivers. Decreasing the required infrastructure can reduce installation and calibration cost. To show how well our algorithms support reduced infrastructure, we compute the tag locations based on the observations from subsets of receivers. As expected, the accuracy of both algorithms decreases as infrastructure density decreases. However, the Kalman filter degrades much more gracefully; note in particular the difference in the vertical accuracy results for Kalman and regression. Moreover, the regression rejects more readings as deployment density decreases, whereas the Kalman filter consistently supplies reliable estimates regardless of density.

	No. of receivers	75% confidence level (cm)		90% confidence level (cm)		Estimates accepted (%)
		Horizontal	Vertical	Horizontal	Vertical	
Non-linear regression	All	8.83	42.13	16.02	68.46	99.98
	4	11.38	47.59	18.25	75.05	99.66
	3	15.02	55.82	22.85	86.29	95.07
	2	24.24	64.76	36.29	108.11	52.78
Kalman filtering	All	7.51	37.35	11.14	52.61	Not applicable
	4	9.10	37.38	13.69	53.02	
	3	11.38	37.60	17.07	51.93	
	2	14.45	35.74	17.26	53.87	

Table 3. Reducing deployment density (heterogeneous data, Lancaster static measurements)

5.2 Results from homogeneous observations

Both the algorithms we have presented are capable of computing location estimates using only pseudoranges or angles-of-arrival. It is interesting to consider their performance on homogenous data in order to judge our algorithms’ application for UWB

position-sensing platforms having fewer capabilities than Ubisense. For example, receivers not equipped with array processing would not be capable of measuring AOA. Or, nodes in a distributed, wireless sensor network might only be able to measure angle-of-arrival, as the nanosecond-level node synchronisation required for pseudorange is prohibitively difficult without a wired connection.

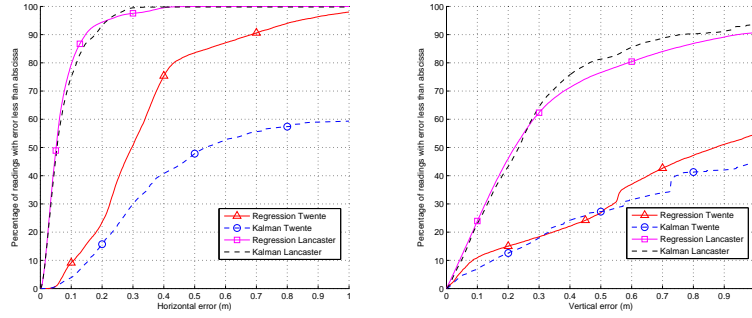


Fig. 5. Positioning static tags using pseudoranges only

Data type	Algorithm	Site	75% conf. level (cm)		90% conf. level (cm)		Estimates accepted (%)
			Horizontal	Vertical	Horizontal	Vertical	
Pseudo-ranges	Regression	Twente	39.80	141.28	68.30	233.93	18.48
	Kalman	Twente	330.62	491.51	1983.08	1229.96	N/A
	Regression	Lancaster	8.85	46.48	14.66	94.44	84.14
	Kalman	Lancaster	10.03	39.17	17.75	75.80	N/A
AOAs	Regression	Twente	190.55	64.10	435.11	103.02	63.93
	Kalman	Twente	452.10	59.62	678.16	119.24	N/A
	Regression	Lancaster	19.56	38.95	30.55	70.40	98.1
	Kalman	Lancaster	17.18	37.15	26.89	54.92	N/A

Table 4. Performance summary of static positioning using homogeneous data

Pseudoranges only. Fig. 5 shows the accuracy of the two algorithms operating on pseudorange data only, at both sites. As in the heterogeneous case, the regression performs significantly better than the Kalman filter on the Twente data; however note the ratio of readings rejected by the regression—over four-fifths for the Twente dataset (Table 4). Even for the Lancaster dataset, the regression rejects over 15% of readings. On the Lancaster data, the Kalman filter’s accuracy is comparable, and it provides constant updates (no readings are rejected). Note that in general the vertical accuracy is inferior compared to the heterogeneous case. This is because the vertical solution is less constrained when only pseudoranges are used (no elevation data is present). The 3D accuracy of the Twente results are especially poor, since the accuracy of the underlying pseudoranges is much lower than in the Lancaster deployment (37 cm compared to 6 cm, at the 75% confidence level).

Angles-of-arrival only. Using only angles-of-arrival, the two algorithms perform comparably (Fig. 6 and Table 4). The accuracy for the Lancaster deployment is quite favourable (horizontal and vertical 75% confidence better than 20 and 40 cm, respectively). This is because for the majority of the Lancaster readings, a reliable azimuth and elevation were reported by all five receivers. The Twente AOA-only results are quite the opposite of the Twente pseudorange-only results. While the Twente vertical accuracy is about as good as the heterogeneous case (Fig. 4), around half of the readings exhibited a horizontal accuracy worse than one metre. Inspecting the per-receiver distributions reveals that three out of the six Twente receivers returned very poor azimuth estimates about 25% of the time. (As mentioned above, this is most likely due to environmental effects and an inaccurate calibration.) Thus, without the pseudorange contribution to the solution, the seventy-fifth percentile horizontal accuracy falls from several tens of centimetres in the heterogeneous case, to several metres in the AOA-only case.

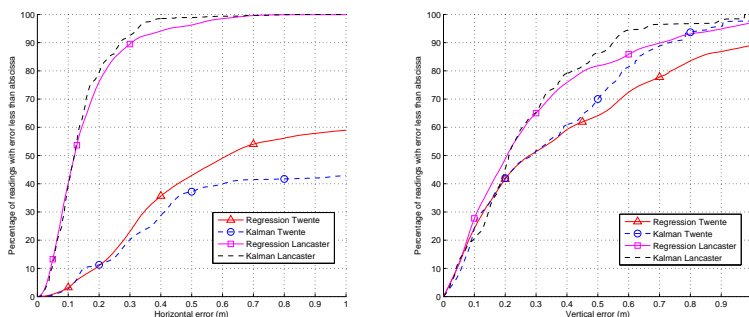


Fig. 6. Positioning static tags using angles-of-arrival (azimuth and elevation)

6 Dynamic Tracking

At Lancaster, dynamic data traces were taken in the test arena. Data was recorded (Fig. 7(a)) in traces lasting approximately seven minutes each, resulting in about 2500 pulse readings per trace. We gathered two types of dynamic data (Fig. 7(b)): “robot” data (four traces) was generated as a Lego Mindstorms robot roamed the arena (velocity 0.16 m/s); and “walking” data (three traces) was recorded as a person pulled the robot around in the arena using a tether (peak velocity of about 1 m/s). The ground truth positions of the tag in the dynamic experiments were recorded using computer vision-based localisation. Two cameras equipped with fisheye lenses were placed above the measurement area, a fiducial marker was rigidly attached to the top of the tag, and reacTIVision software was used to perform accurate localisation in real time. Since there are slight differences in the timestamps between the logged camera estimates and the ubisense estimates, we use a weighted-average method to interpolate the camera data for the corresponding ubisense timestamp. Where there were large gaps in the camera timestamps, the interpolated values may not be valid, hence interpolation was done only if the surrounding data points are close in time (shorter than 300 ms). For gaps greater

than this, the Ubisense readings are discarded, to allow accurate comparison. For most traces, this method resulted in about 10% of receiver observations being dropped.

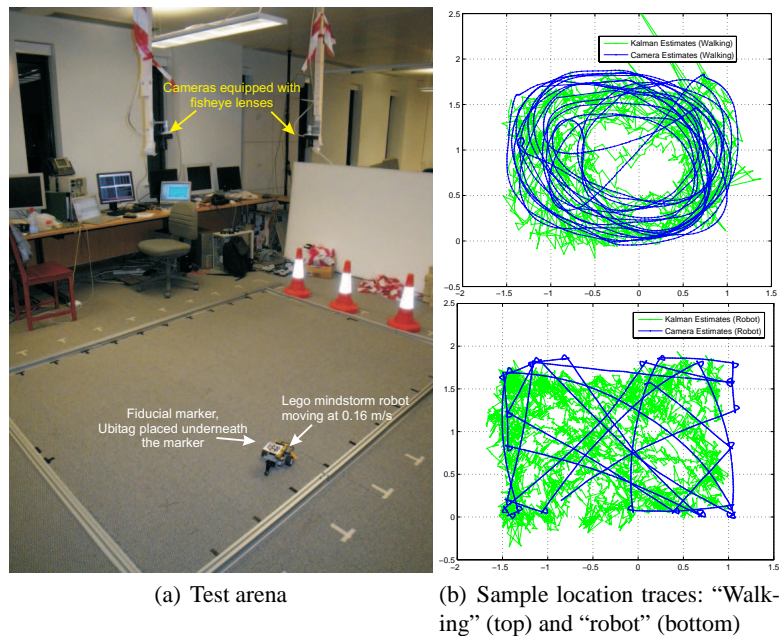


Fig. 7. Lancaster dynamic experiment

The location accuracy for the “robot” and “walking” traces is roughly equivalent for each algorithm, despite the large difference in the speed of the tag. This is likely due to the sufficient update rate of Ubisense readings (about 10 Hz) for both types of trace. Since the accuracy does not change appreciably between the two speeds, for the homogeneous dynamic case (Fig. 9 and Table 5), we consider only the three “walking” traces. The “walking” homogeneous results are comparable to the estimates for static data taken at similar height. For example, the regression operating on pseudoranges only yields a 75% horizontal dynamic accuracy of 20 cm, which is exactly the same accuracy as for the static readings taken at a height of 7.5 cm.

7 Ubisense Location Engine Estimates

Although the goal of this paper is to characterise our algorithms’ performance with different deployments and types of UWB positioning data (pseudorange and angle-of-arrival), we provide here the estimates produced by the Ubisense Location Engine for purposes of comparison. Fig. 10 shows the static tag positioning accuracy of the LE when configured to “default no filtering.” The regression and Kalman algorithm results for the same data are also shown. Like our regression algorithm, the LE produces a

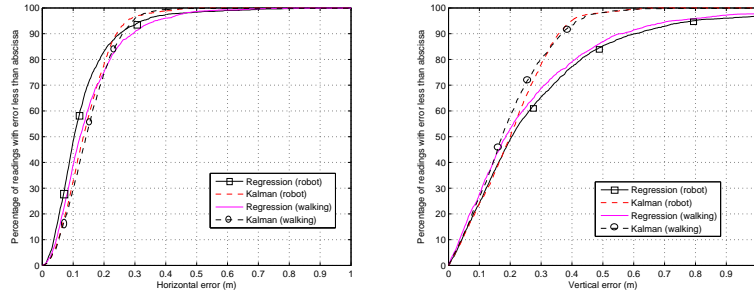


Fig. 8. Dynamic tracking of tags using heterogeneous data (Lancaster)

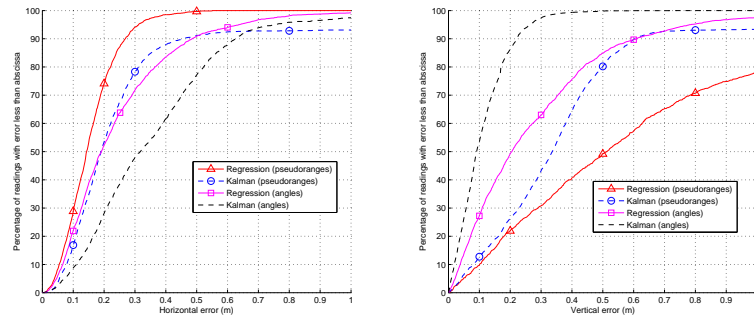


Fig. 9. "Walking" tracking accuracy using homogeneous data (Lancaster)

Data type	Algorithm	75% conf. level (cm)		90% conf. level (cm)		Estimates accepted (%)
		Horizontal	Vertical	Horizontal	Vertical	
Pseudoranges and AOs	Regression	18.31	35.31	27.46	55.43	99.66
	Kalman	21.02	26.07	31.42	36.12	N/A
Pseudoranges only	Regression	20.29	90.63	26.86	167.24	61.86
	Kalman	33.30	46.27	55.84	53.90	N/A
AOs only	Regression	32.43	39.58	48.53	61.01	94.83
	Kalman	48.69	15.87	62.87	22.16	N/A

Table 5. Dynamic tracking performance summary (Lancaster). All values shown pertain to the location results of the three "walking" data traces.

measure of standard error. But, because we do not know how the LE standard error is calculated, for this comparison we have not rejected any estimates from the LE or the regression results. With no filtering, the Ubisense LE performs quite comparably to our proposed algorithms for static tags. Likewise, the dynamic accuracy of the LE is similar to our results (Fig. 8) if “no filtering” is set.

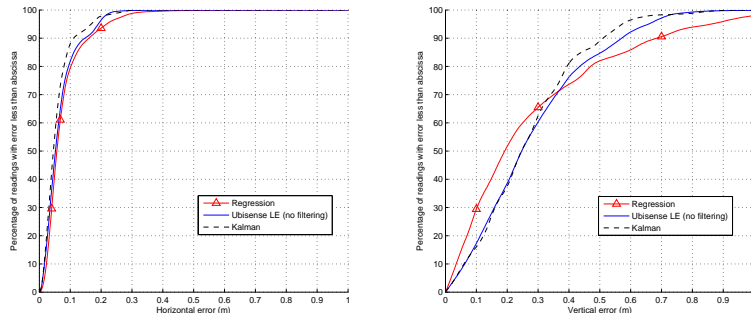


Fig. 10. Ubisense Location Engine accuracy (Lancaster, static readings)

8 Conclusion

This paper has presented a non-linear regression and a Kalman filtering algorithm designed specifically to process the heterogeneous data which UWB positioning systems are capable of producing. Both algorithms fuse the different types of raw data (pseudorange, azimuth, and elevation) effectively. For reliably accurate raw data (as produced by the Lancaster deployment), the algorithms exhibit similar performance, and we would select the Kalman filter since it provides a more consistent (if at times slightly less accurate) stream of location estimates. Supplied with reliable readings, the Kalman filter performs better than regression as deployment density decreases. For deployments with poor calibration and/or less reliable, “noisy” readings (as in the Twente data), we would select the non-linear regression algorithm for its accuracy, despite the high ratios (20–80%) of rejected readings.

We have shown that the algorithms can work well on homogeneous data (pseudoranges or AOAs), despite the reduction of the information contributing to the location solution. Under certain configurations, noisy, homogeneous data can be pathological, as in the Twente pseudorange-only vertical accuracy, or the Twente AOA-only horizontal accuracy. When working on homogeneous data with reliable accuracy, the algorithms continue to produce good location estimates, as seen in the Lancaster results.

In many of our results, the vertical location accuracy was worse than the horizontal location accuracy. As noted previously, this is very much a function of the geometry of the receivers in our deployments. One can envision receiver deployment strategies which correct the bias, such as fixing some receivers high on the wall (as in our test deployments), and some receivers to the ceiling with their boresight facing the floor. The ceiling-mounted receivers’ reported AOAs would contribute more heavily to the horizontal estimate, and the pseudoranges (in combination with pseudoranges received from wall-mounted receivers) would contribute much more to the vertical estimate.

As future work, we plan to explore *adaptive* Kalman filters. This is particularly interesting as Kalman performance is heavily influenced by the choice of the model

parameters, and making the filter adaptive can automatically tune the model parameters based on the current measurement. Usage of other probabilistic algorithms, for example particle filtering, may also be a fruitful area for investigation. Another avenue of research is to extend the presented algorithms to address *online auto-calibration* of reference nodes. Called simultaneous localisation and mapping (SLAM) in robotics, this can also be used for calibration-sensitive infrastructure to bring us a step closer in realising easily deployable yet accurate positioning systems.

Acknowledgements. We are grateful to Dr Andy Ward for providing insightful comments and suggestions on this work. We are deeply indebted to Carl Fischer for setting up the camera tracking system and collecting the dynamic data.

References

1. P. Duff and H. Muller. Autocalibration algorithm for ultrasonic location systems. In *Proceedings of the Seventh IEEE International Symposium on Wearable Computers, White Plains, NY*, October 2003.
2. R. J. Fontana, E. Richey, and J. Barney. Commercialization of an ultra wideband precision asset location system. In *Proceedings of the IEEE Conference on Ultra Wideband Systems and Technologies*, pages 369–373, Reston, Virginia, USA, Nov. 2003.
3. R. J. Fontana. Experimental results from an ultra wideband precision geolocation system. In *Ultra-Wideband, Short-Pulse Electromagnetics*, pages 215–224, May 2000.
4. D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, pages 24–33, 2003.
5. E. Foxlin, M. Harrington, and G. Pfeifer. Constellation: a wide-range wireless motion-tracking system for augmented reality and virtual set applications. In *Proceedings of SIG-GRAPH 98*, pages 371–378. Addison Wesley, 1998.
6. S. A. Glantz and B. K. Slinker. *Primer of Applied Regression and Analysis of Variance*. McGraw-Hill, New York, second edition, 2000.
7. R. Negenborn. Robot localization and Kalman filters: On finding your position in a noisy world. Master’s thesis, Utrecht University, 2003.
8. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.
9. V. Renaudin and B. M. M. Kasser. Optimal data fusion for pedestrian navigation based on UWB and MEMS. In *Proceedings of PLANS*, 2008.
10. R. Mehra. On the identification of variances and adaptive Kalman filtering. *IEEE transactions on Automatic Control*, AC-15(2):175–184, April 1970.
11. J. Scott and B. Dragovic. Audio location: Accurate low-cost location sensing. In *Proceedings of the International Conference on Pervasive Computing*, 2005.
12. A. Smith, H. Balakrishnan, M. Goraczko, and N. B. Priyantha. Tracking moving devices with the Cricket location system. In *Proceedings of the Second International Conference on Mobile Systems, Applications and Services (MobiSys)*, June 2004.
13. Ubisense Ltd. Location Engine config manual, 2007.
14. A. Ward. Calibration of a location system. Patent PCT/GB2007/001425, Nov 2007.
15. A. M. R. Ward. *Sensor driven Computing*. PhD thesis, Corpus Christi College, University of Cambridge, 1998.
16. G. Welch and G. Bishop. An introduction to the Kalman filter. Technical Report 95-041, Dept. of Computer Science, University of North Carolina, 2006.
17. G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Colucci. The HiBall Tracker: high-performance wide-area tracking for virtual and augmented environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*, 1999.