Precise Self-Calibration of Ultrasound Based Indoor Localization Systems

Armin Runge, Marcel Baunach and Reiner Kolla
Departement of Computer Engineering, University of Würzburg, Am Hubland, 97074 Würzburg, Germany
Email: {runge,baunach,kolla}@informatik.uni-wuerzburg.de

***Abstract*—Several ultrasound based localization systems consist of environmental anchor nodes, and mobile nodes, which estimate their own position by using a static infrastructure. For the location process, every anchor has to know its position. In most approaches, the location of all anchors has to be determined a priori manually. This procedure is time consuming and fault-prone. In this paper, we present Distribute & Erase and Explorer, two self-calibration methods for ultrasound based localization systems. The first uses a set of three pre-calibrated anchors to explore the whole localization system whereas the second refines the anchors' positions progressively. Both of our approaches require no additional hardware besides ultrasound receivers or transmitters, and radio transceivers, which have to be already available for the WSN based localization system.**

I. Introduction

Ultrasound (US) based localization systems are an important research area within the WSN domain. A set of small, energy-efficient, and cheap sensor nodes makes it possible to localize and track mobile objects. This is the reason, why various such systems yet exist, like Active Bat [1], AHLoS [2], Cricket [3], Buzz [4], or SNoW Bat [5]. These systems are using an infrastructure of static sensor nodes with known positions (hence called anchors), to locate mobile nodes, which are attached to the objects under surveillance. For the localization of a mobile node, the distance to several anchors with known positions must be determined. Our research is based on the localization system SNoW Bat (see Fig. 1), developed by Baunach et al. It is optimized for fast and precise localizations and is based on small and energy-efficent SNoW^{5}sensor nodes [6]. Each SNoW^{5}sensor node has a radio transceiver for communication and coordination. For distance measurement each mobile node is equipped with an US transmitter, whereas each anchor contains an US receiver. A single localization process is initiated solely from the mobile node by sending a specific radio packet followed by a certain US signal. The radio signal propagates with approximately the speed of light ($\approx 300000\,\mathrm{km/s}$) whereas the US signal propagates much slower with velocity $v_{US} \approx 331.45\,\mathrm{m/s}\sqrt{1+\frac{\theta}{273}}$ [1]. This velocity highly depends on the current ambient temperature θ, for example at $20\,^{\circ}\mathrm{C}$ the velocity is $v_{US} \approx 343\,\mathrm{m/s}$ and at $35\,^{\circ}\mathrm{C}$ it is $v_{US} \approx 352\,\mathrm{m/s}$. To calculate the current speed of US signals, each node at the SNoW Bat system measures the temperature. Every anchor that has received both signals can calculate the Time Difference of Arrival (TDoA) between the radio and the US signal, according to the thunderstorm principle [9]. The radio signal combines initiation of the localization process as well as time synchronization between the nodes, which is necessary for TDoA. After estimating the distances, the anchors reply a so called Distance Vector (DV), which contains the estimated distance and the anchors' (estimated) position. Afterwards the mobile node can calculate its position estimation with the received DVs using a localization algorithm like trilateration or multilateration [10].

[1]The given formula is a simplified version of the sound velocity formula for gases, assuming dry air. The derivation can be found in [7], [8].

II. Related work

A common problem of anchor based localization systems is the determination of the anchors' positions. A manual calibration is time consuming, since each anchor has to be measured individually, and fault-prone, because of inaccurate measurement methods and human error. Our objective was to find a self-calibration method, which achieves an accuracy of millimeters and requires no further hardware. Several self-calibration approaches are based on distance measurement between anchor nodes, such as the method introduced by Capkun et al. [11]. Local coordinate systems are generated for each anchor, which afterwards are joined together to a global coordinate system. There are also several calibration approaches based on multidimensional scaling (MDS), as the approach of Kadam et al. [12], but they are also based on the inter-node distances. In our real world localization system, the distances are measured by TDoA between ultrasound and radio signals emitted by a mobile node and received by an anchor. Therefore, the distance measurement between two anchors is not feasible without adaption of our hardware.

A GPS-based method was provided by Sichitiu and Ramadurai [13]. Instead of equipping each node with an expensive GPS-receiver, they used just one mobile node with GPS. The positions of the static anchors were determined using the distances between the anchors and the mobile node. However, GPS-based methods are not suitable for indoor localization, because of the unreliable reception of GPS signals inside buildings. A localization accuracy of about $1-2\,\mathrm{m}$ is achievable herein, which is also insufficient for us.

Moses et al. [14] reached a higher accuracy by using sensors, which measure both the direction-of-arrival (DOA) and the time-of-arrival (TOA). But the sensor nodes used in

Fig. 1. Lab view of the localization system SNoW Bat . The static anchors are attached to the ceiling of our testbed frame. The mobile node is not visible.

our localization system are not able to measure the DOA without expensive hardware modification.

Menegatti et al. investigated the capability of WSNs and autonomous mobile robots (AMRs) for calibration [15]. They introduced a self-calibration method with an AMR that determines its position through odometers attached to the driving wheels. Since this method presupposes a special mobile vehicle it is also prone to slippage of the tires.

The autocalibration algorithm [16] of Duff and Muller requires only a user who is walking around and measures distances to the anchors. As localization systems are always slightly over specified, their algorithm can calculate the anchors' positions, using a multidimensional nonlinear least squares fitting algorithm. The original algorithm requires a controller circuit equipped with a radio transmitter, anchor nodes equipped with US transmitters, and a mobile node equipped with both US and radio receiver. There is also a version that does not use radio [17]. The achieved accuracy amounted to $20\ \mathrm{cm}$. Duff's and Muller's autocalibration algorithm resembles our Distribute & Erase algorithm (section IV), but Distribute & Erase calibrates decentralized and contains several methods to increase the achievable accuracy.

In contrast to the approaches mentioned above, we require a precise self-calibration method for ultrasound based sensor networks, which just relies on small, cheap, and energy-efficient sensor nodes. Our first approach, the Explorer algorithm, is a straightforward approach, which allows three dimensional calibration within short time. But as Explorer propagates position and distance errors, which are unavoidable in a real localization system, the maximum size of the localization system to be calibrated should be limited. This is the reason, why we developed the Distribute & Erase algorithm, where occurring errors are not propagated, and which still allows precise calibrations.

Our algorithms are not limited to ultrasound based localization systems, only a reliable distance estimation between the anchor nodes and a mobile node is required. Even the calibration of RSSI based localization systems and the combination with an algorithm for automatic calibration of the propagation model, as presented in [18], is imaginable.

The remainder of this paper is organized as follows. The next section introduces our Explorer algorithm. In Section IV we explain Distribute & Erase, including the requirements and some improvements. A discussion of our experimental results follows in Section V. In Section VI we compare our both algorithms with regard to some important properties for self-calibration algorithms. Finally, future work in Section VII and a conclusion in Section VIII closes this paper.

III. Explorer

To apply the Explorer algorithm at least three pre-calibrated anchor nodes are required. In order to simplify, we assume just one single mobile node m. The pre-calibrated anchors have to be placed in such a way that the mobile node can measure a distance to all of them. Based on this requirements, the mobile node can locate itself and send its estimated position back to the involved anchor nodes. After the self-localization, the mobile node moves onto its next position on its pre-defined path, at which also at least three calibrated anchors must be able to measure their distance towards the mobile node. Each non-calibrated sanchor, which has collected at least three non collinear position estimations of m, can localize itself. Such newly calibrated anchors can then be used for further localization of the mobile node. So, Explorer works as follows:

```
while(localization system not entirely explored) {
  mobile node moves onto next position; //on path
  distance measurement;                 //using TDoA method
  self-localization of m;               //see Section IV-C
  broadcast estimated position of m;
  foreach(anchor a) {
    if(more than 3 DVs stored) {
      self-localization of a;    //see Section IV-C
    }
  }
}
```

While repeating the algorithm above, the mobile node explores the environment, hence more and more anchors get calibrated. Calibration with the Explorer algorithm takes only few steps per anchor (for our tests in Section V, 5 steps per anchor were sufficient), as an anchor is calibrated if it has stored at least three DVs.

But if Explorer calibrates an anchor with a high position error, this faulty position estimation is the basis for subsequent localizations of m, which in turn will be used for localizations of further anchor nodes. Therefore, Explorer propagates position errors, and as distance errors are unavoidable in real world localization systems, the arising errors even increase. To keep the errors as small as possible, Explorer discards distance and position estimations with a large error probability.

Detail information about the distance filtering and position estimation algorithm (also for D&E) are given in Section IV-C and IV-E. However, the appearance of position errors can not be excluded. This is the reason, why the maximum size of the localization system to be calibrated should be limited according to the desired accuracy.

Fig. 2a shows the position errors in space after a calibration using the Explorer algorithm. The pre-calibrated anchor nodes are highlighted. The serpentine path of the mobile node went from corner $(0, 0)$ to the opposite corner at $(30000, 20000)$, starting in depth direction. According to the error propagation described above, the position error increases with distance from the starting point.

A proper choice of the pre-defined path may reduce the average position error. I.e. if the mobile node starts in the middle of the anchor plane, the average distance between the pre-calibrated anchors and the other anchors is smaller than that one from Fig. 2a. But the disadvantage of an increasing distance error corresponding to the distance to the pre-calibrated anchors still remains. Furthermore, this algorithm requires a pre-defined path, which must be followed by the mobile node, as well as a certain number of pre-calibrated anchors. That is the reason why we next introduce a stable self-calibration algorithm, which is able to calibrate an unlimited number of anchors, is easy to deploy, and always is able to achieve a high accuracy in the order of millimeters.

IV. Distribute & Erase

This section introduces our self-calibration algorithm Distribute & Erase, which can replace the manual calibration process even for large localization systems. For the rest of this paper, $p_{i,real}$ denotes the real position of an anchor a_i with $i \in \{1, \ldots, n\}$, $p_{i,est}$ denotes the estimated position of anchor a_i, and A_i is the grid cell (see IV-A) of anchor a_i. In order to simplify, we assume just one single mobile node m with real position $p_{m,real}$, and estimated position $p_{m,est}$ respectively.

A. Requirements of D&E

To apply our Distribute & Erase (D&E) algorithm only a few pre-conditions must be fulfilled. Not a single pre-calibrated anchor is required if relative positions are sufficient for the underlying localization system. If absolute position estimations are required, two (for 2D calibration) or three (for 3D calibration) pre-calibrated anchors at fixed positions are needed. The pre-calibration corresponds to the specification of the origin and orientation of the local coordinate system within the global coordinate system. The room size and the number of anchors to be calibrated are unlimited, only the required time for the calibration process increases with the number of anchors.

Distribute & Erase calibrates the anchors by a gradual improvement of their estimated positions. The anchors' estimated initial positions can be chosen randomly in general. However, as trilateration is used for all localizations, the positions used for a localization must not be collinear. The mobile node can take arbitrary positions for distance measurement, too, as long as each anchor measures a sufficient number of distances. In contrast to Explorer, the mobile does not need to stick on a pre-defined path.

As in Explorer, the required time for a localization step depends strongly on the time of the data aggregation. To achieve a high localization frequency, the data must be packed tightly. If the anchors are roughly arranged along a grid and each anchor knows it's cell, the Time Division Multiple Access (TDMA) communication protocol HashSlot [19] can be used. HashSlot assigns an exclusive and individual transmission slot to each anchor, depending on the anchor's cell. This ensures a minimal and predictable transfer time, and avoids mutual interference. In the case of uncalibrated anchors, the anchors' cells must be determined.

So, if HashSlot is used to control the transmission of the radio packets, every anchor just has to know its anchor cell. D&E can employ this precondition for successive calibration of all anchors. Of course, D&E can also be used with other MAC protocols, like CSMA-CA, but this reduces the calibration speed remarkably.

B. Procedure

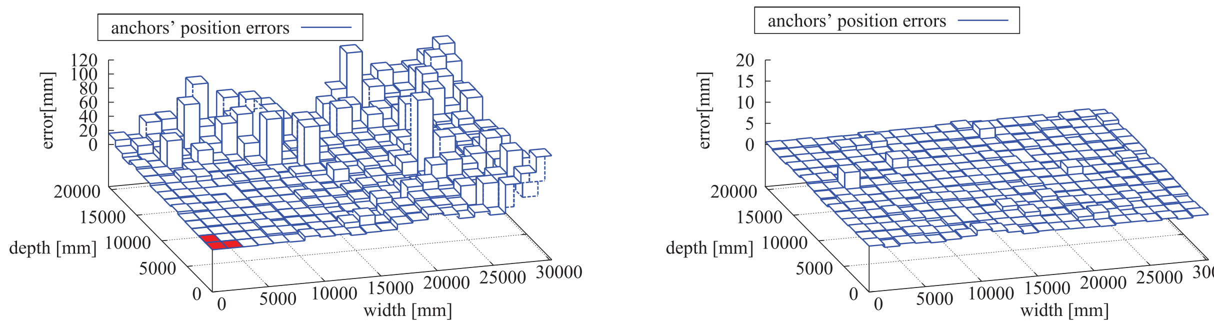
Every position in the anchor cell could be chosen as initial position $p_{i,est}$ of an anchor a_i. To simplify, each anchor's estimated position $p_{i,est}$ is initially placed in the middle of the corresponding anchor cell A_i. The mobile node takes a new position and locates itself using the received DVs from the anchors. The mobile node sends the newly estimated position back to the anchors, which in turn buffer this position for their own positioning. Algorithm executed on mobile node:

```
while(true) {
  take new position;   //change p_{m,real}
  measure distances;   //using TDoA method
  calculate p_{m,est};    //using trilateration (IV-C)
  broadcast p_{m,est};
}
```

If an anchor a_i has collected at least three non-collinear position estimations of the mobile node, it is able to update its own estimated position $p_{i,est}$, which offers a smaller position error on average. Algorithm executed on anchor node:

```
while(true) {
  //localization dependent parts
  if(new position p_{m,est} received) {
    store DV;    //p_{m,est} and measured distance to m
  }
  if(enough DVs stored) {
    update p_{i,est}; //using trilateration (IV-C)
  }
}
```

Depending on the transmitting node, a DV contains an estimated position $p_{m,est}$ or $p_{i,est}$ and the measured distance between m and the anchor a_i. One calibration step corresponds to a positional change of m. An anchor arrangement is called *totally consistent*, if the re-localization of all anchors with perfect distances only confirms the already estimated positions.



(a) Anchor plane calibrated with Explorer, the pre-calibrated anchors are marked in red

(b) Anchor plane calibrated with Distribute & Erase

Fig. 2. Position error of a calibrated anchor plane with 330 anchors

In repeating the localization of m and all anchors a_i, after a few (depending on the number of anchors) calibration steps, a *consistent*[2] anchor arrangement will be achieved (see Section IV-D).

C. Localization algorithm

For the localization process, the D&E algorithm must tolerate inaccurate anchor positions, but still has to provide precise position estimations for accurate anchors. The *adapted trilateration* fulfills these requirements and works for a node $k \in \{m, a_i, \ldots, a_n\}$ as follows:

```
generate j triplets;
generate j position estimations;
if(position estimations ∈ A_k ≥ 4) {
  p_{k,final} = coordinate−wise median of all
        estimations ∈ A_k;
}
```

The *adapted trilateration* builds j triplets from the q received DVs. If $j < \binom{q}{3}$, the triplets that span the largest triangles are selected. Throughout our tests we found that $j = \frac{1}{2}\binom{q}{3}$ results in a good trade-off between run-time and accuracy. With each of the j triplets a position is estimated using a regular trilateration. Only position estimations located in the area A_k will be considered. By discarding all estimations $\notin A_k$, obviously wrong position estimations are filtered, which leads to an improved average accuracy. The final position estimation $p_{k,final}$ is calculated using the coordinate-wise median of all considered position estimations. To achieve position estimations as accurate as possible, the *adapted trilateration* requires at least four positions $\in A_k$.

D. Illustration and match operation

Fig. 3 shows the calibration process for a small anchor plane with 3×3 anchors and a grid-length of $\approx 1.3\,\mathrm{m}$[3]. The estimated position of each anchor is located in the middle of the corresponding cell and the real position is randomly placed anywhere within the cell. It is obvious that there is still after 800 steps a significant deviation between $p_{i,est}$ and $p_{i,real}$ (Fig. 3b). According to the plotted path in Fig. 3b it seems that the distance between two estimated positions for an anchor a_i was small in the recent steps. The reason for the small distances is an almost *consistent* anchor arrangement. To match the real and the estimated positions two anchors a_{fix_1} and a_{fix_2} with known real positions are required. But the match operation is only necessary if absolute coordinates are desired. With a_{fix_1} the translation values δ_x and δ_y can be calculated:

$$\delta_x = p_{fix_1,real}.x - p_{fix_1,est}.x$$

$$\delta_y = p_{fix_1,real}.y - p_{fix_1,est}.y$$

with $p_{i,real} = (p_{i,real}.x, p_{i,real}.y)$, and $p_{i,est}$ respectively. Three-dimensional coordinates can be translated with the translation matrix

$$A_{trans} = \begin{pmatrix} 1 & 0 & \delta_x \\ 0 & 1 & \delta_y \\ 0 & 0 & 1 \end{pmatrix}.$$

Fig. 3c shows the anchor plane translated by (δ_x, δ_y). Afterwards, the translated coordinates must be rotated around the chosen point of origin $p_{fix_1,real} = (x_0, y_0)$. The rotation angle θ is the angle between the straight connection of $p_{fix_1,real}$ and $p_{fix_2,real}$, and the straight connection of $p_{fix_1,real}$ and the translated coordinate $p_{fix_2,est}$. The rotation is realized using the transformation matrix

$$A_{rot} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & x_\theta - x_\theta \cdot \cos(\theta) + y_\theta \cdot \sin(\theta) \\ \sin(\theta) & \cos(\theta) & y_\theta - x_\theta \cdot \sin(\theta) - y_\theta \cdot \cos(\theta) \\ 0 & 0 & 1 \end{pmatrix}.$$

The affine transformation A_{at} to match all anchors is

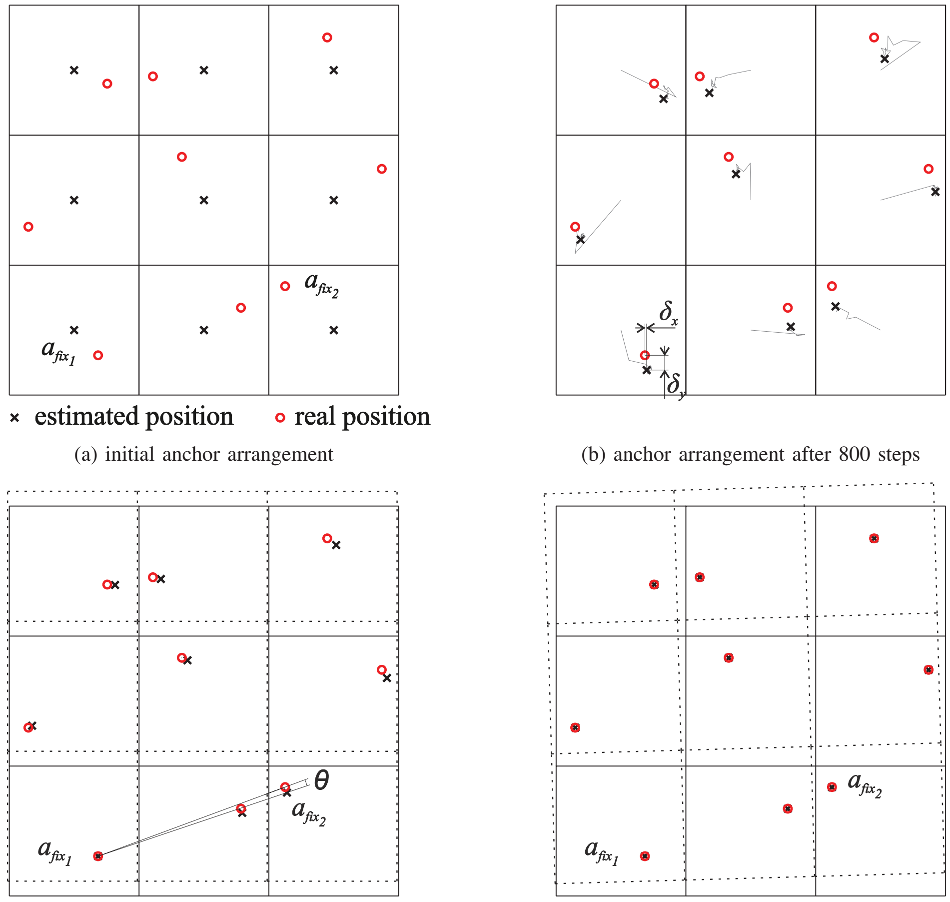
$$A_{at} = A_{rot} \times A_{trans}.$$

Fig. 3d shows the matched anchor plane, now offering an average error of just $0.7\,\mathrm{mm}$.

For three-dimensional calibrations a further rotation around the axis joining a_{fix_1} and a_{fix_2} is necessary. The corresponding rotation angle can be calculated with a third static anchor a_{fix_3} with known position $p_{fix_3,real}$.

[2]As already mentioned, real world distance measurements are error-prone, and this is the reason why a totally consistent anchor arrangement is hardly achieved there.

[3]The grid-length was automatically calculated depending on the height of the room, the US emission angle, and the necessary number of anchors within the US cone.



(a) initial anchor arrangement

(b) anchor arrangement after 800 steps

(c) anchor arrangement after translation

(d) anchor arrangement after rotation

Fig. 3. Anchor plane during calibration with Distribute & Erase

E. Improvements

The achievable calibration accuracy depends strongly on the error characteristic of the distance measurement. Therefore, it can be necessary to improve the accuracy by distance filtering. The error characteristic shown in Fig. 4 was measured within the real world localization system SNoW Bat . The distance error of about $\pm\lambda$ originates if the arrival time of the ultrasound signal is over- or underestimated by one single US-period. Smaller distance errors resulting e.g. from manufacturing tolerances of the US capsules, the timer resolution of the sensor nodes, and the operating system.

Instead of using just a single measured distance, the median of 100 distance measurements is used if the variance of the distance sequence is less than or equal a certain value υ. If the variance is more than υ, the complete sequence is discarded, as the probability for a large error increases with the derivation. The error probability measured in our real world localization system could be improved significantly (see Fig. 4). Note that the error characteristic was measured under stable environmental conditions.

During development, the accuracy of the calibration can be

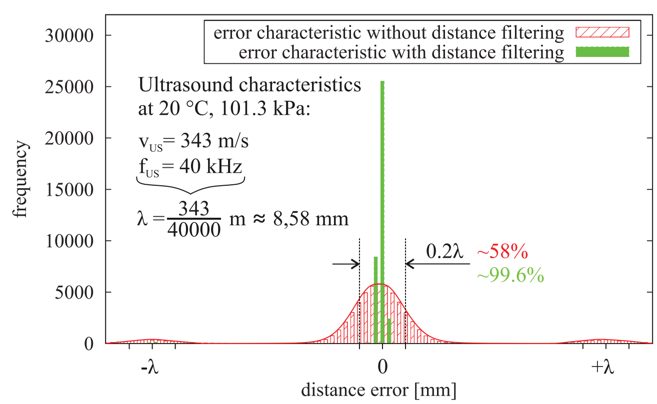


Fig. 4. Frequency density of the error characteristic of the distance measurement

determined by the average position error of all anchors

$$f_a = \frac{1}{n}\sum_{i=1}^{n} ||p_{i,est} - p_{i,real}||.$$

Because the real positions are unknown at an uncalibrated system, f_a can not be computed in real world implementations. Therefore, we searched for a metric, which allows the evaluation of the calibration accuracy and is calculable by an algorithm. For the localization of node k (mobile node or anchor) the $agreement_k$ is defined, which depends on the size of the *adapted trilateration*'s scatter plot:

$$agreement_k := \frac{1}{c} \sum_{j=1}^{c} |p_{k,final} - p_j|$$

where $p_j,\ j \in \{1, \ldots, c\}$ are the position estimations $\in A_k$ and $p_{k,final}$ is the final position estimation. If the resulting scatter plot is large, the likelihood for an imprecise position estimation is high, too. In contrast, a small scatter plot is an indication for a more precise position estimation.

The localization of a node k can also degrade accuracy, since the localization possibly is based on faulty position estimations. To shorten the calibration time, an adjustment depending on $agreement_k$ was implemented. If a new position $p_{i,est}$ is estimated for an anchor a_i, the adjustment vector $\binom{x_{t-1}-x_t}{y_{t-1}-y_t}$ from the past estimated position (x_{t-1}, y_{t-1}) to the new estimated position (x_t, y_t) is calculated. Instead of accepting the new position estimation, a weighted adjustment is performed with $\alpha \in (0, 1)$:

$$p_{i,est} = (x_{t-1}, y_{t-1})^{\mathrm{T}} + \binom{x_{t-1}-x_t}{y_{t-1}-y_t} \cdot \alpha^{agreement_i}.$$

A sufficient value of α at system start is 0.9. But as f_a decreases over time, a smaller α value should be chosen after some calibration steps. This is the reason, why α is determined by the average of the last 100 $agreement_m$ values of the mobile node m.

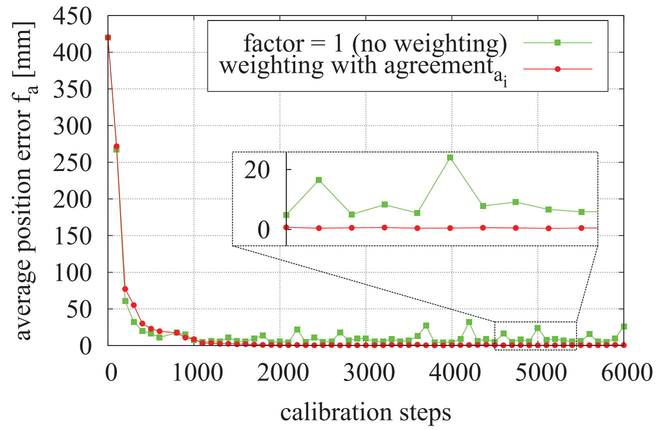
V. Experimental results

Fig. 5a shows the progress of f_a for a calibration with Distribute & Erase of 11×11 anchors with and without a weighted adjustment. The advantage of the weighted adjustment becomes even more obvious for a higher amount of calibration steps, where unique distance errors recurrently increase the average position error without any weighting.

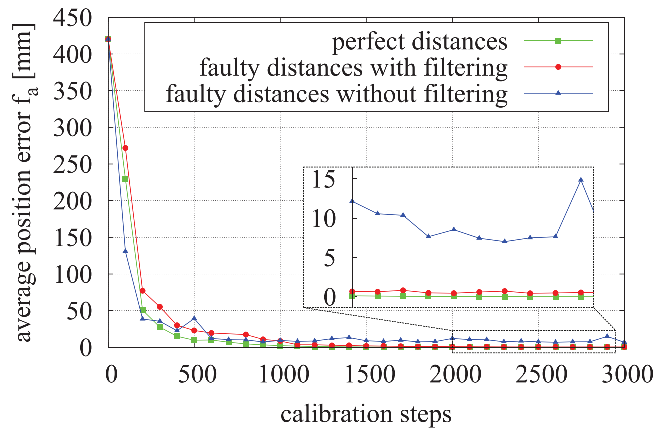
As mentioned in Section IV-E, the achievable calibration accuracy depends strongly on the accuracy of the distance measurements. Fig. 5b shows the average error during calibration of 11×11 anchors. With perfect distances, 1200 calibration steps are required to achieve an average error $f_a < 1\,\mathrm{mm}$, and 3000 steps for $f_a \approx 0.0\,\mathrm{mm}$. Even with faulty distances an accuracy $< 1\,\mathrm{mm}$ can be achieved after 1800 steps using distance filtering. Without distance filtering, large distance errors can increase the average error f_a, e.g. the average error is still $> 5\,\mathrm{mm}$ even after 3000 steps.

The time to achieve a calibration of a certain accuracy depends also on the number of anchors to be calibrated. Fig. 5c shows the required steps to calibrate various numbers of anchors using exact distance measurements.

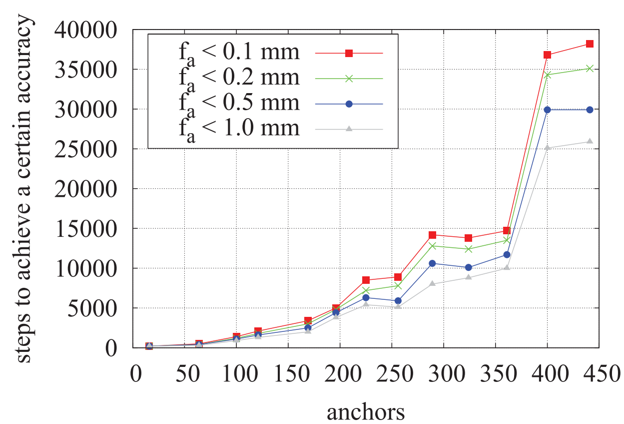
The most time consuming parts of the calibration are the movement of m and the data aggregation. Assuming a localization frequency of $5\,\mathrm{Hz}$, 100 distance measurements



(a) Progress of f_a with and without weighting



(b) Progress of f_a for different distance errors



(c) Necessary steps to achieve a certain accuracy for a given number of anchors

Fig. 5. Simulated results of the calibration with Distribute & Erase

per step for distance filtering, a duration of 10 seconds per movement of m, and 1800 calibration steps (Fig. 5b), the calibration of an anchor plane with 11×11 anchors takes:

$$t_{D\&E} = 1800 \times (10\,\mathrm{s} + \frac{100}{5\,\mathrm{Hz}}) = 15\,\mathrm{h} \tag{1}$$

This time can be reduced if several mobile nodes are used, measured distances are stored (see Section VII), or if a reduced

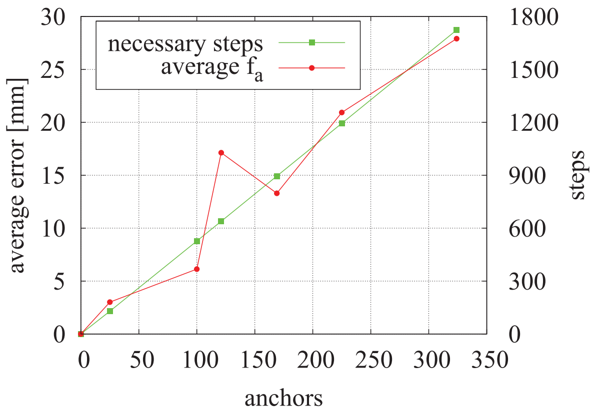


Fig. 6. Necessary steps and achievable f_a using Explorer for a given number of anchors

accuracy is acceptable. As D&E is a progressive calibration method, the last calibration steps could also be done during the operation of the localization system.

As distance errors are unavoidable, and can cause position errors, a high fault-tolerance is of particular importance for self-calibration methods. Distribute & Erase purges such position errors as the position error of an anchor is distributed to its neighboring nodes and will be erased using a match operation. Fig. 2b shows the anchors' position errors for a larger anchor plane (15×22 anchors) in comparison to the Explorer algorithm (Fig. 2a). Please note also the different z-ranges of both plots.

Because the calibration algorithm Explorer does not work progressively, the comparison to D&E is of limited value. Even single distance errors can corrupt the whole calibration in Explorer. Therefore, the variance of a series of average position errors f_a is large. Fig. 6 shows the achievable error f_a and the necessary steps for the calibration of a various number of anchors using Explorer. All values were determined by the average of three calibrations.

Unlike D&E, the necessary steps increase linearly with the number of anchors by using the Explorer algorithm. Moreover, the achievable average position error of all anchors increases with the number of anchors to be calibrated. The reason for the rapid increase of f_a is the error propagation property, which increases with the distance to the pre-calibrated anchors. Using the conditions from equation 1, but assuming just 640 steps, the calibration of an anchor plane with 11×11 anchors takes

$$t_{Exp} = 640 \times (10\,\mathrm{s} + \frac{100}{5\,\mathrm{Hz}}) \approx 5.3\,\mathrm{h}$$

with an average position error $f_a \approx 17\,\mathrm{mm}$. As shown in Fig. 5b, D&E achieves an average error $f_a < 50\,\mathrm{mm}$ after 400 steps and $f_a < 20\,\mathrm{mm}$ after 600 steps.

VI. Comparison

In the following we compare some important properties of self-calibration algorithms for our Explorer and D&E algorithms.

Achievable position error: As already shown in Section V, using distance filtering, D&E achieves an average error $f_a < 1\,\mathrm{mm}$ after a sufficient number of calibration steps (cf. Fig. 5c). Explorer achieves just for small anchor planes a small average error f_a (cf. Fig. 6), because the position error of the anchors increases with the distance to the pre-calibrated ones.

Fault tolerance: Explorer propagates position errors, which leads to bad calibrations or even an impossible calibration for widespread localization systems. D&E purges such errors, as occurring position errors can be compensated with further localizations. For real world WSNs the fault tolerance is of particular importance, since errors can occur, which have never occurred before.

Limitations: As shown in Section III, using Explorer limits the size of the localization system to be calibrated. To enhance the size of the calibratable anchor plane, a separation into several sub-planes would be possible, but each sub-plane needs at least three pre-calibrated anchors. Furthermore, the mobile node has to follow a pre-calculated path. For D&E there are no such restrictions. D&E calibrates an unlimited number of anchor nodes and the mobile node can take arbitrary positions for distance measurement, as long as about the same number of distances are measured to all anchors.

Parameters: Before Explorer can be used, a path for the mobile node and several other parameters, like the velocity of the mobile node on its path, must be considered. Most of these parameters depend on the infrastructure of the localization system and are hard to determine automatically. In D&E, there are no such parameters to be taken into account, since the positions of the mobile node can be chosen arbitrarily. However, both algorithms rely on parameters, which depend on the error characteristic of distance measurement (e.g. for the distance filtering). But these parameters can be easily figured out automatically.

Progressivity: D&E calibrates by enhancing roughly estimated positions. Therefore, the last calibration steps can also be gained during operation of the localization system. In contrast, Explorer calibrates not progressively. An estimated position is never changed and hence the localization system can not be used until the calibration process is finished.

Decentralized calibration: Due to the low computational effort and the low memory requirements, D&E as well as Explorer can be implemented on low performance microcontrollers (typically used in WSNs). The accuracy is independent of centralized or decentralized operation mode. Therefore, even if the distance filtering is CPU or space-intensive, both algorithms will operate in SNoW Bat .

Speed of calibration: As shown before in this section, Explorer needs fewer steps to calibrate the same anchor plane as D&E. But we are currently working on several improvements of D&E, which will reduce the number of necessary steps (see Section VII).

VII. Future work

As shown in Section III, the achievable accuracy with Explorer depends on the chosen path of the mobile node. That is why we examine, how an optimal path can be selected, depending on the localization system's infrastructure and the error characteristic.

To reduce the required time for a calibration with D&E, we try to minimize the number of necessary steps. A measured distance $d_{i,t}$ between an anchor a_i and the mobile node m at time t depends only on both positions $p_{i,real}$ and $p_{m,real}$ at time t. So, $d_{i,t}$ is completely independent from estimated positions. Therefore, a distance estimation d_{i,t_1} at time t_1 remains valid, even if the estimated positions $p_{i,est}$ or $p_{m,est}$ are updated at time $t_2 > t_1$.

Until now, a measured distances from the mobile node at position $p_{m,real}$ to the anchors a_i are used only for a single localization of m and a re-localization of the involved anchors a_i. After the localizations the distance estimation is discarded. Hence, the memory requirements and the simulated calibration times are low. Unfortunately the time for a simulated calibration depends strongly on the computational effort, which is high for the localization algorithm (e.g. trilateration) and low for positional changes or distance measurement. In real world calibrations, the necessary time for positional changes of m and distance measurements are considerably higher than the runtime of the localization algorithm. This is the reason, why we are working on an implementation, which will reuse the already measured distances.

The number of required steps for the calibration of a single anchor depends mainly on the error characteristics of the US distance measurement and several environmental factors. Therefore, we expect to get an upper bound of necessary steps, which increases linear with the number of anchors to be calibrated, like in Explorer.

To demonstrate the robustness and fault tolerance of our algorithms described above, we are currently working on a two dimensional as well as a three dimensional real world implementation. Herein, our main focus lies on realistic radio effects like unreliable links, packet loss, or faulty data reception.

VIII. Conclusion

In this paper we introduced our self-calibration algorithms Explorer and Distribute & Erase. The Explorer algorithm is suitable for fast calibration of small localization systems. In this algorithm, the mobile node explores the whole localization system using at least three pre-calibrated anchors. Disadvantages of Explorer are the limited size of the localization system to be calibrated, the need of pre-calibrated anchor nodes, and a pre-defined path to follow.

Distribute & Erase is a precise self-calibration approach, suitable even for large localization systems. Starting with roughly estimated positions $p_{i,est}$, D&E calibrates progressively by improving the position estimations. Several improvements were developed to enhance the accuracy and speed of the calibration. We showed that the achievable accuracy highly depends on the accuracy of the distance measurement system. With the error characteristics measured in our real world localization system an average position error $f_a < 1\ \mathrm{mm}$ was achievable.

Distribute & Erase can replace the hard, time-consuming, and fault-prone manual calibration. Furthermore it also operates distributed, autonomously, and without further hardware or special a priori knowledge. Therefore, D&E is suitable for the fast, cheap, and easy calibration of localization systems during deployment. As D&E calibrates progressively, it is also suitable for recalibration, especially after changes of the infrastructure.

References

[1] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The anatomy of a context-aware application," *WIRELESS NETWORKS*, vol. 8, pp. 187–197, 1999.

[2] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, 2001, pp. 166–179.

[3] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *6th ACM International Conference on Mobile Computing and Networking*, 2000.

[4] M. McCarthy, "The Buzz: Narrowband Ultrasonic Positioning for Wearable Computers," Ph.D. dissertation, University of Bristol, April 2007.

[5] M. Baunach, R. Kolla, and C. Mühlberger, "SNoW Bat: A high precise WSN based location system," Universität Würzburg, Lehrstuhl für Informatik V, Tech. Rep. 424, 2007.

[6] M. Baunach, R. Kolla, and C. Mühlberger, "SNoW5: A versatile ultra low power modular node for wireless ad hoc sensor networking," in *5. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze, FGSN, 2006*, 2006, pp. 55–59.

[7] P. Tipler and G. Mosca, *Physics for Scientists and Engineers*, ser. Physics for Scientists and Engineers. W. H. Freeman, 2007, no. Bd. 3, ch. 14.

[8] D. A. Bohn, "Environmental Effects on the Speed of Sound," *J. Audio Eng. Soc*, vol. 36, no. 4, pp. 223–231, 1988.

[9] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2007, ch. 8.

[10] M. Ilyas and I. Mahgoub, Eds., *Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems*. CRC Press, 2005, ch. 20–21.

[11] S. Capkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad hoc networks," in *Cluster Computing*, 2001, pp. 3481–3490.

[12] R. Kadam, S. Zhang, Q. Wang, and W. Sheng, "Multidimensional Scaling Based Location Calibration for Wireless Multimedia Sensor Networks," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, ser. IROS. Taipei, Taiwan: IEEE Computer Society, 2010, pp. 5654–5659.

[13] M. L. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," in *Proceedings of the First IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, 2004.

[14] R. L. Moses, O. L. Moses, D. Krishnamurthy, and R. Patterson, "A Self-Localization Method for Wireless Sensor Networks," *EURASIP Journal on Applied Signal Processing*, vol. 4, pp. 348–358, 2002.

[15] E. Menegatti, L. Lazzaretto, and A. Zanella, "Self-localization of Wireless Sensor Nodes by means of Autonomous Mobile Robots," in *Proceedings of the Tyrrhenian International Workshop on Digital Communication Wireless Communications (TIWDC 2007)*, 2007.

[16] P. Duff and H. Muller, "Autocalibration Algorithm for Ultrasonic Location Systems," in *Proceedings of the 7th IEEE International Symposium on Wearable Computers*, ser. ISWC '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 62–.

[17] T. Snowsill, "A New Method For Wireless Ultrasound Positioning," Master's thesis, University of Bristol, August 2007.

[18] P. Barsocchi, S. Lenzi, S. Chessa, and F. Furfari, "Automatic virtual calibration of range-based indoor localization systems," *Wireless Communications and Mobile Computing*, 2011.

[19] M. Baunach, R. Kolla, and C. Mühlberger, "A Method for Self-Organizing Communication in WSN Based Localization Systems: HashSlot," in *Second IEEE International Workshop on Practical Issues in Building Sensor Network Applications, SenseApp 2007*, 2007.