ALF: An Autonomous Localization Framework for Self-Localization in Indoor Environments

Juergen Eckert*, Reinhard German* and Falko Dressler†
*Dept. of Computer Science, University of Erlangen, Germany
†Institute of Computer Science, University of Innsbruck, Austria
{juergen.eckert,german}@informatik.uni-erlangen.de, falko.dressler@uibk.ac.at

***Abstract*—A lot of algorithms and applications can benefit from position information. GPS localization has become a standard for outdoor usage. But if a higher accuracy is needed or within GPS-denied areas providing this knowledge is still an open and nontrivial topic. Especially for unknown or dynamic environments. In this paper we propose a framework which is capable of autonomously exploring unknown environments in a fully decentralized way. It provides accurate and real-time localization support for customers. The usual very time intensive manual deployment and position assignment of reference nodes is avoided. Additional we show that the algorithm can detect and handle Non Line of Sight (NLOS) issues which is a very important criteria for real world applications.**

I. Introduction

Location awareness is an important topic for mobile systems. In the field of personal computing, location-based services enabled a hitherto unknown type of interacting with an application. In other domains such as Sensor and Actor Networks (SANET), it is at least equally important [1], [2]. The demands in terms of accuracy, cost, power consumption, etc. change with different application scenarios. Typically, all localization approaches have one common characteristic: a reference is needed before a location can be determined. This can be either a map (showing obstacles) or an active (or even passive) reference grid consisting of landmarks.

GPS has become a *de facto* standard for outdoor localization. In GPS-denied areas, like urban or indoor environments, or when a higher accuracy is required, other techniques need to be used. Due to of the high variance of the system parameters and scenarios this is still a hot research topic. Besides of hardware issues related to sensing and measurement capabilities, which are not in the focus of this paper, the robust and autonomous generation of a localization reference is one of the most challenging tasks [3]. In this paper, we show how to set up such a reference system in a fully self-organizing way and how to use this reference system for accurate localization of highly mobile systems such as flying quadcopters.

Our goal is to reduce the hardware requirements to a minimum (due to cost and energy issues). Therefore, we can not use Simultaneous Localization and Mapping (SLAM) techniques, which usually use expensive laser distance sensors and/or resource-intensive image evaluation [4]. We started with developing a cheap mobile robot platforms [5] for providing a reference grid to localization customers. These robots are equipped with a short range radio for communication and with an Ultrasound (US) ranging device for neighbor and customer detection. Using a self-organizing approach, no global stages or knowledge can be assumed or will be generated during system runtime. Figure 1 depicts our Autonomous Localization Framework (ALF). Only based on inter-node distance measurements, mobile nodes are autonomously deploying themselves on the ground and start forming a robust reference grid. A customer, e.g. a quadcopter, can use this system to determine its location. A fully distributed and self-organizing algorithm creates and maintains the reference grid using an enhanced Mass-Spring-Relaxation (MSR) technique (step 1 in Figure 1). An initial localization of potential grid nodes helps to prevent the system from oscillating (step 2). The same technique is also used for customer localization (step 3) [6].

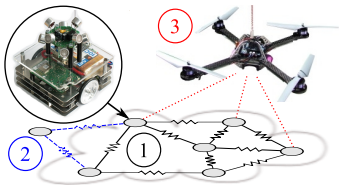

Fig. 1. Conceptual model of our Autonomous Localization Framework (ALF)

Flying systems such as quadcopters pose hard real-time requirements on the localization system. Our autonomous localization framework has been designed specially for this application. We, however, believe that the same system is well suitable for any self-localization approach that is based on autonomous mobile systems forming a reference grid. The contribution of this paper can be summarized as follows. We developed a robust and decentralized framework for enabling anchor-free localization support using low-cost robot systems (Section III). In an experimental setup, we evaluated the localization performance of our system (Section IV). We specifically show that no global state is necessary. We further show that the implementation of the framework is capable of handling hard *real-world* problems like measurement errors or Non Line of Sight (NLOS).

II. Related Work

A. Localization Systems

The Cricket system [7] is probably the best known low-cost localization system for indoor usage. It is a derivate

from the Active Bat localization system [8] and relies, like its predecessor, on US-based Time of Flight (ToF) measurements to compute positions. The reported resulting absolute position error is less than 30 cm. Although its design is fully autonomous and decentralized, a non-negligible effort is required for the reference node deployment and the bootstrapping process of the coordinate system.

Recently Chintalapudi et al. presented EZ [9], which is a WiFi-based localization approach. Basically, EZ can be seen as a follow-up to the RADAR system [10]. Both systems estimate their positions using Received Signal Strength Indicator (RSSI) measurements. Thus, the systems are at least one order of magnitude less accurate compared to those relying on US and ToF measurements. Depending on the scenario and a probability threshold P, EZ has an accuracy of 2 m to 7 m (for $P = 0.5$). The main advantage of both the EZ and the RADAR systems is that no special deployment effort is needed to bootstrap the system.

We are trying to combine the advantages of both approaches, the accuracy of US-based systems and the ease of the deployment of the WiFi systems.

B. Mobility Support / Mobile Robots

In general, mobility support has been identified as a source for improving the localization accuracy [11], [12]. Thus, for a accurate and low-effort localization, the individual platforms need to be mobile. Furthermore, localization is one of the key requirements to coordinate mobile robot systems. In (multi-)robot localization applications, the individual units are usually equipped with quite resource and energy expensive components. Laser distance scanners and computer vision are commonly used approaches [4], [13]. Frequently, both sensing techniques are combined. On the positive side, this enables the use of SLAM techniques. However, the resource requirements are well above typical sensor network applications. Furthermore, the advanced position sensors are actually only needed during the construction of the reference grid, assuming that customer localization should be performed using less expensive technology.

Our approach is to rely on cheap mobile platforms for deploying a minimal initial sensor array. Although all of our sensors are mobile, this is not a major issue. For example, Shell et al. developed a system where mobile platforms have been used to deploy immobile human audible beacons [14]. The same technique can be used in our approach to further reduce the system costs for the sensor deployment.

C. Non Line of Sight

Even if the reference nodes could be placed in an intelligent way, measurement errors cannot be excluded. Depending on the signal propagation characteristics, different sources for erroneous measurements need to be considered. One of the most critical issues is the NLOS case. The Cricket system [7] circumvented this issue by placing the reference node on the ceiling. Therefore, emitters and receivers always have Line of Sight (LOS) and nearly no measurement errors due to NLOS are to be expected. However, this restricts the scenario to indoor applications and a manual deployment of the nodes, i.e. mounting them to the ceiling.

Due to multi-path propagation of NLOS measurement pairs, the localization errors can get very large. A common approach to determine the LOS connections is to execute multiple measurements and to statistically reduce the error. Opposed to the traditional approach of a time history of range measurements, Guvenc et al. [15] only require the amplitude and delay statistics of a channel in order to perform a NLOS detection. Another approach has been presented by Chan et al. [16]. Based on probably incorrect measurements, they identify NLOS conditions by evaluation the distribution function of all possible positions.

Both approaches compute 2D positions. In the this paper, we show that NLOS can be identified without additional statistics if 3D positions are available together with some easy to collect information.

III. Autonomous Localization Framework

In the following, we introduce our Autonomous Localization Framework (ALF), which is based on a number of components for setting up and updating the reference grid and determining customer positions using US measurements. We stepwise explain the different algorithms needed for an optimized localization behavior. Some of the described techniques are optional techniques, introduced to improve the accuracy or to obtain additional system information. The entire framework has been built in a modular way: individual components can be replaced depending on the potentials of the used hardware. It should be noted that two assumptions must be fulfilled: nodes need to be capable of measuring to direct neighbors and they need to be able to exchange information with those neighbors.

The key objectives of the localization framework are

- the initial fully self-organizing deployment of an irregular reference grid, i.e. use of autonomous mobile robots driving to appropriate places in the environment that span a coordinate system by cooperatively assigning themselves relative (anchor-free) positions,
- the accurate customer localization based on the positions of reference grid, and
- the continuous update of the localization grid if the environmental conditions change, e.g. due to new obstacles or added / removed robot systems.

A. Advanced Mass-Spring-Relaxation

ALF is based on accurate localization as a basis technique. In our previous work, we presented a distributed version of the MSR algorithm, the Advanced Mass-Spring-Relaxation (advMSR) [3]. Our localization algorithm is based on distance measurements and also, if possible, takes Angle of Arrival (AoA) measurements into account. The advMSR is inherently self-organizing, thus, assuming no global knowledge about the available nodes and topology. Basically, the localization generates tuples $(north, east)^T$ for each node, which are used by the ALF system.

The principles of the position estimation and the maintenance process of the reference grid is reported in [3]. In the following, we assume the availability of such localization techniques and discuss system oriented aspects of how areas can be explored, how the heading of a node can be determined, and we study communication aspects of the framework. A key focus is on the handling of measurement errors due to NLOS situations.

B. Area Exploration

In the initialization phase, we assume that all nodes start at a similar location and need to be uniformly distributed over a given but unknown area. In consequence, the nodes need to be able to autonomously determine their initial coordinates (relative to each other) to finally spread over the area.

In the following, first, the localization accuracy and the probability for a successful setup are determined. Basically, the more neighbors are allowed per node and the *better* they are placed the more accurate the system will get. However, a higher neighbor connectivity also means that the costs will increase significantly. Secondly, the total system costs are defined. This is obviously a trade-off between area coverage, accuracy, and cost.

Similar to the MSR-based grid maintenance algorithm, we again used a force vector approach. We define different zones around each node to optimize the node distribution process. Let r_{min} and r_{max} define two radii, which divide the area Z_i around the origin $p(N_i)$ of each node N_i into three zones (here, $\|\cdot\|_2$ is the euclidean distance norm). Then, the zones can be defined as:

$$Z_{restricted,i} := \{\vec{x} \in \Re^2 \mid 0 < \|p(N_i) - \vec{x}\|_2 < r_{min}\} \quad (1)$$

$$Z_{desired,i} := \{\vec{x} \in \Re^2 \mid r_{min} \leq \|p(N_i) - \vec{x}\|_2 \leq r_{max}\} \quad (2)$$

$$Z_{attractive,i} := \{\vec{x} \in \Re^2 \mid r_{max} < \|p(N_i) - \vec{x}\|_2 < \infty\} \quad (3)$$

As the zone names indicate, neighboring nodes are not allowed in within $Z_{restricted,i}$. Thus, r_{min} indirectly defines the maximum density of nodes and the associated maximum system costs per ground area. $Z_{desired,i}$ is the desired belt around a node N_i where neighbors should be located. Node N_i will only connect to nodes within this region. The number of neighboring nodes within this belt $|Z_{desired,i}|$ is equal to the edge degree $d(N_i)$ of node N_i from a graph theory point of view ($|Z_{desired,i}| = d(N_i)$). A threshold λ is used to control the system costs. This threshold is basically limiting the maximum connectivity degree: $3 \leq d(N_i) < \lambda$, where 3 is the minimum number of required connections for calculating a 2D position.

Again, the higher the connectivity $d(N_i)$ gets, the more accurate localization results can be obtained, but at increasing system costs. This problem has already been investigated in the literature. A typical value reported in the literature is in the range of 8 to 13 connections per node [17]. We experimented with those values in the scope of the ALF framework. In most experiments, already a connectivity degree of 6 led to astonishingly good results. NB: λ should not get below the error correction threshold of 4 (see Subsection III-E).

TABLE I
PLACEMENT DECISION CONSTRAINTS

Guard	Action	Use Nodes (Z_c)
$\|Z_{restricted,i}\| > 0$	drive	$Z_i \setminus Z_{desired,i}$
$3 \leq \|Z_{desired,i}\| < \lambda$	place found	-
$\lambda \leq \|Z_{desired,i}\|$	drive using $\frac{1}{2}\hat{d}_{i,j}$	$Z_{desired,i}$
$\|Z_{attractive,i}\| > 0$	drive	$Z_{attractive,i}$
else	random drive	-

Furthermore, the radii r_{min} and r_{max} must be chosen according to the optimal detection range of the used distance measurement hardware. The closer both radii get, the more regular distributions of the nodes will be achieved. However, the more difficult it becomes for the algorithm to find a valid solution. Of course, r_{max} should also not be set close to the maximum of the detection range in which the distance measurement quality decreases in a non-linear way.

During the positioning phase of each node N_i, distances $\hat{d}_{i,j}$ to all detectable neighbors are measured and sorted into the three introduced sets. The resulting actions to be taken depend on four guards that are evaluated serially:

1) if neighbors are in the restricted zone, the node has to find a more appropriate position;
2) if a sufficient number of nodes are in the desired zone ($3 \leq |Z_{desired,i}| < \lambda$), the node position can be fixed;
3) if too many nodes are in the desired zone, the node will move away from those nodes;
4) if the node has to move and if there are nodes in the attractive zone, the node will move towards those.

In any other case, the node will start searching randomly for better locations. All the actions are summarized in Table I.

In the following, we briefly introduce the force vector approach for repositioning nodes to form the reference grid. All the calculations are performed locally at a node, only taking the measurements to neighboring systems into account.

Node N_i is connected to each node from the selected set with a spring of equilibrium length of $l_0 = \frac{r_{min}+r_{max}}{2}$. Based on the chosen set Z_c ($|Z_c| = j$), the force vector $\vec{F}_i$ for the movement is computed according the basic MSR as depicted in Equation 4, where $k_{i,j}$ is the spring constant (usually set to 1), $\vec{e}_{i,j}$ characterizes the unit vector from node N_i to node $N_j \in Z_c$, and $\hat{d}_{i,j}$ represents the corresponding measured distance.

$$\vec{F}_i = \sum_{N_j \in Z_c} \vec{F}_{i,j} = \sum_{N_j \in Z_c} -\vec{e}_{i,j} k_{i,j} (\hat{d}_{i,j} - l_0) \quad (4)$$

The unit vector $\vec{e}_{i,j}$ can be obtained using two different approaches. All nodes in Z_c know there positions, thus, node N_i can localize itself according to these references and finally apply simple geometrical calculations. The problem is that parallel operation and bootstrapping become extremely difficult. Another approach is to measure angle information. Most distance ranging devices are based on multiple sensors; our system uses four independent US sensors [5]. Thus, the AoA

can be estimated. Our system provides an accuracy of at least $\pm 45°$, which is sufficient for the computations.

The final driving direction and distance are proportional to the force vector $\vec{F}_i$. Small measurement uncertainties and the mobility of nodes helps to prevent oscillations in the system to achieve fast convergence. NB: this scheme works well both in 2D as well as in 3D.

C. Heading

The previously mentioned AoA estimation can also help to estimate the heading of the platform, which is a necessary measure in various scenarios. In theory, the heading Ψ_i of a node N_i can be computed using one arbitrary neighbor N_j. According to the positions $p(N_i)$ and $p(N_j)$, the absolute heading $\Psi_{i,j}^{north} = atan2(p(N_j) - p(N_i))$ in relation to the north pole (which is the commonly assumed as $0°$)) can be determined. Subtracting the measured AoA $\hat{\Psi}_{i,j}$ results in the true heading $\Psi_{i,j} = \Psi_{i,j}^{north} - \hat{\Psi}_{i,j}$.

The described calculation becomes error prone in real scenarios due to biased measurements and positioning errors. Simple averaging over all j headings $\Psi_{i,j}$ is not possible because of two reasons. First, there is a non-linearity due to the overflow of the co-domain ($\Psi_{i,j} \in (-\pi, +\pi]$), which might lead to wrong average values. Secondly, $p(N_i)$, $p(N_j)$ or $\hat{\Psi}_{i,j}$ might be error prone and heavily affect the results.

The overflow of the domain can be compensated by moving from angles to a vector-based representation:

$$\vec{\psi}_{i,j} = (\sin(\Psi_{i,j}), \cos(\Psi_{i,j}))^T \tag{5}$$

However, the second issue is more challenging. Summing up all heading vectors already allows to conclude to a fairly good estimation. This can further be improved by adding a weight indicating the confidence to each vector. For our distance measurement hardware, we see that the shorter the measured distance is, the more accurate the measured AoA gets [5]. Thus, the weighting function $\kappa(\cdot)$ returns larger values for nodes in close proximity.

The overall heading can be computed described in Equation 6. The angle $\Psi_i = atan2(\vec{\psi}_i)$ of the resulting vector $\vec{\psi}_i$ represents the heading of node N_i. The length of the vector $\|\vec{\psi}_i\|_2$ allows to conclude the confidence of this information. It is normalized to $\kappa(\cdot)$.

$$\vec{\psi}_i = \frac{1}{j} \sum_{N_j} \kappa(N_j) \cdot \vec{\psi}_{i,j} \tag{6}$$

D. Communication

For the ALF system, no specific communication topology is required. A sufficient criterion is that each node N_i must be able to communicate with all of its j neighboring nodes $N_j \in \hat{N} := \{N_n \in V \mid \|p(N_i) - p(N_n)\|_2 \leq r_{max}\}$ (where V represents the set of nodes). Multicast communication to all neighbors would be the optimal solution for this application, because most of the transfered information needs to be made available to all neighbors simultaneously. In wireless networks, broadcast is an appropriate alternative.

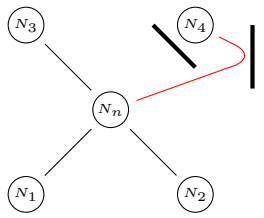

Fig. 2. NLOS scenario (top view)

In order to join an existing network, a node N_i registers to all of its neighbors N_j after finding an adequate position. This registration needs to be reliable. Thus, reliable unicast is needed for this process. If new or updated information (e.g., position $p(N_i)$, neighbor N_j, measurement $\hat{d}_{i,j}$) becomes available at node N_i, it pushes this data to all of its neighbors. These in turn evaluate the data and perform the necessary actions. This data will be transmitted more frequently. Therefore, no reliable communication channel is required.

As a result, the amount of transmitted data packets is reduced to a minimum and it also provides mechanisms for autonomously entering energy saving standby states after system convergence has been reached. Also, no network-wide flooding is needed.

E. Measurement Errors

Usually, it is assumed that a measurement path is symmetric, i.e. $\hat{d}_{i,j} = \hat{d}_{j,i}$. However, this only holds in approx. $90\,\%$ of our measurements and, depending on the scenario, may be even worse. A bad alignment between emitter and receiver of the US distance measurements can falsify the result depending on the direction, i.e. not a direct but a reflected signal triggers the measurement. Those incorrect measurements cannot be distinguished from correct measurements as the jitter is equal (the worst case scenario is static without mobility).

We try to correct such errors using a rather simple error model: Both directions need to be measured to identify asymmetric measurements. As we use ToF, the shortest flight time must be the most correct measurement ($\tilde{d}_{i,j}$ represents the measurement from node N_i to node N_j):

$$\hat{d}_{i,j} = \hat{d}_{j,i} = \min(\tilde{d}_{i,j}, \tilde{d}_{j,i}) \tag{7}$$

If $\epsilon_{i,j} = |\tilde{d}_{i,j} - \tilde{d}_{j,i}| > \theta$, where θ represents the hardware specific error range, then the node that measured the larger distance can also not rely on its AoA estimation because it received the reflected signal. Overall, this solution gave us almost $100\,\%$ reliability for the LOS cases.

Detecting NLOS links is even more challenging. Consider the example depicted in Figure 2. This network consists of four initially perfectly distributed nodes N_1, N_2, N_3, and N_4 at positions $(0,0)$, $(0,1)$, $(1,0)$, and $(1,1)$. The newly arriving node N_n starts the localization procedure to subsequently join the network. Node N_n has a NLOS link to node N_4, all other links are LOS. None of the participating nodes can distinguish between LOS and NLOS.

The distances will be determined as $\hat{d}_{n,i} = 2^{-\frac{1}{2}}; i \in [1,3];\ \hat{d}_{n,4} = 1$. Thus, using basic trilateration techniques [6], the node will be located at four equally weighted positions: $(0.5, 0.5)$, $(0.25, 0.25)$, $(0.25, 0.5)$, and $(0.5, 0.25)$. From this 2D view it is neither possible to determine the correct position nor to identify the outliers.

However, from a 3D point of view and by adding additional knowledge about the height of a system (in our case, all the robots are driving on the floor), the correct position can be obtained by looking at the Z-coordinates: 0.0, NaN, 0.43, and 0.43. Obviously, the first coordinate $(0.5, 0.5, 0.0)$ must be the correct result. Measurements that result in wrong Z-coordinates are considered NLOS links and need to be blacklisted in the localization framework. Without any loss of generality, this approach can be applied to almost any localization scheme. Obviously, additional information is necessary to identify NLOS cases. For best results, additional hardware components are needed to minimize such failures. For obtaining height information, for example simple pressure sensors could be used providing a relative accuracy of a few centimeters.

In general, NLOS can be detected if three measurement tuples are available together with additional height information. If one or more additional tuples are available all NLOS links can be identified as long as three correct LOS links are available.

The third important source of error is the co-called *flip ambiguity* problem, which is typical for almost any localization technique [18]. Due to small measurement errors, nodes may get wrong initial coordinates. This typically happens if the reference nodes for the localization are nearly collinear placed. In such situations, very small measurement errors significantly falsify the Z-component. Thus, it is very likely that those get always ignored. We counteract this issue exploiting the mobility of our nodes, using short movements in random directions.

F. Network Bootstrapping

An initial network V' needs to be constructed before nodes can regularly join. Three nodes ($|V'| = 3$) are sufficient to span up a plain. However, to avoid placement errors at the very beginning, a fourth node is required. All four nodes need to be fully connected and well placed to safely bootstrap the grid: Three nodes generate the coordinate system and the fourth node localizes itself according to this trio to verify that there is no NLOS in the initial network (see Section III-E). In order to speed up the initial bootstrapping, we implemented the bootstrapping in a central manner, even though the localization process is fully decentralized. A root node requests all measurements from three nodes, performs all the computations related to the initial localization step, and, afterwards, assigns the initial positions. This central approach with very small computational efforts clearly outperforms other solutions that require clearly more time for the initialization and a high number of transmissions to identify NLOS issues.

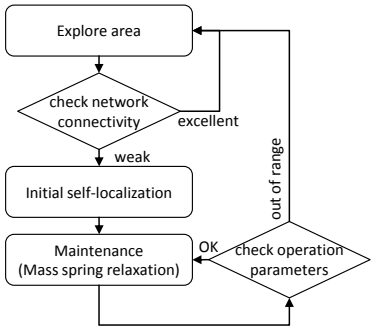

Fig. 3. System behavior of the ALF localization framework

G. Overall System Behavior

The system behavior of ALF for each node N_i is summarized in Figure 3. After starting the system, each node is searching for an existing localization grid. If the particular node is immobile, it is also feasible to sleep for a certain time and then retry to establish a connection. As soon as the node finds a sparsely covered part of the network, it starts the initial self-localization according to [6] before it can become part of the grid by registering with the neighbors in the grid.

The robot then continuously tries to improve its initial position using our advMSR technique. As soon as the estimated localization error falls below a certain threshold, the robot is ready to serve customers' localization requests. In theory, the system can stay in this state for a very long time. No additional measurements, computations, or transmissions need to be performed. However, if errors can not be solved or if the connectivity level falls below a predefined threshold, the robot can re-enter the initial phase and move to another position. No global knowledge or additional synchronization is involved to establish the grid.

IV. Evaluation

In order to evaluate the localization performance of the Autonomous Localization Framework, we performed a number of experiments, each focusing on specific characteristics of the algorithm. First, we briefly show that the used experimental setup, i.e. our localization hardware, performs the advMSR algorithm with similar accuracy as previously estimated in simulations [3]. We then carefully evaluated the behavior in NLOS situations, before finally assessing the overall system performance.

A. Advanced Mass-Spring-Relaxation Performance

The core of the ALF framework is based on the Advanced Mass-Spring-Relaxation technique. It has been carefully examined in a custom build simulator for node sizes of up to 1000 nodes [3]. The algorithm only interacts with direct neighbors and does not require any global knowledge. The advMSR self-localization performed extremely well in simulation as previously reported. In order to validate the algorithm before starting to assess the overall performance of our framework, we repeated those simulation experiments with lower node numbers

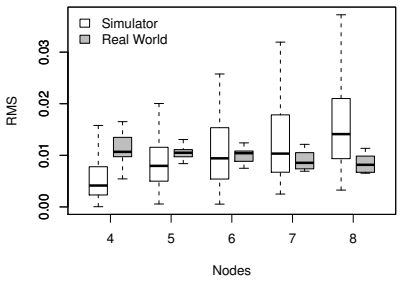

Fig. 4. Validation of simulation and experimental results

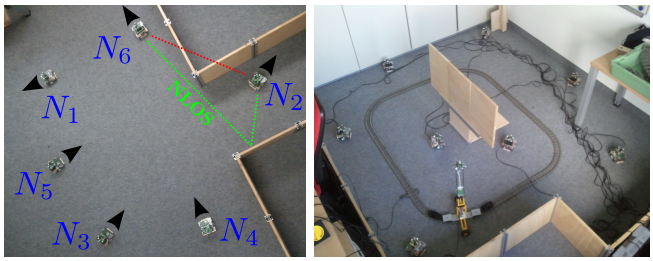

Fig. 5. NLOS experiment (left) and customer localization (right)

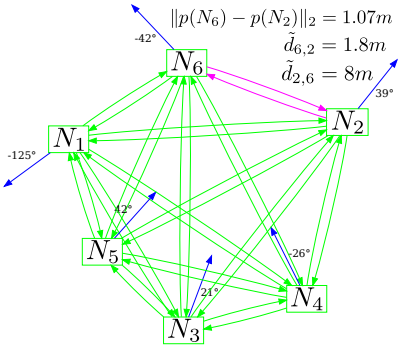

Fig. 6. Resulting network representation

and also executed similar experiments using our localization hardware. Basically, up to 8 nodes were placed randomly in the environment before starting the advMSR procedure.

Figure 4 shows that for real world experiments the constructed network representation accuracy is within the co-domain of our simulator. The evaluation is based on relative node distances between the calculated positions and the actual ones. The Root Mean Square (RMS) of the normalized error of both the simulation and the lab experiments are similar. The RMS has been identified as an accurate measure to compare self-localization solutions [19]. The figure depicts the RMS for different network sizes in form of boxplots. The thick line represents the median. The rectangular boxes contains $50\,\%$ of the measurements indicating the $25\,\%$ and $75\,\%$ quantiles. Finally, the whiskers show the $2.5\,\%$ and $97.5\,\%$ quantiles.

B. NLOS

Figure 5 (left) shows a typical NLOS situation. The edge between nodes N_6 and N_2 cannot be measured correctly due to an obstacle. Multipath propagation effects result in the following measured distances: $\tilde{d}_{6,2} = 1.8\,\mathrm{m}, \tilde{d}_{2,6} = 8\,\mathrm{m}$, whereas the true physical distance is $d_{6,2} = 1.04\,\mathrm{m}$.

The accumulated result as used internally by ALF of a selected run is depicted in Figure 6. In this figure, all the nodes, their headings, as well as the connecting links are drawn. Green links are correct from the algorithm's point of view. Mangenta links indicate blacklisted ones, which are not considered for further computations. As can be seen, ALF reliable detected and ignored only the NLOS link. The distance error was $|\|p(N_6) - p(N_2)\|_2 - d_{6,2}| = 3\,\mathrm{cm}$. Without this blacklisting scheme, no correct solution could be found.

In total, we conducted two experiments (ten repetitions for each) with the same setup. For the NLOS detection, a threshold ξ_A is needed to decide whether to accept the computed position or not. For the first experiment, we used $\xi_A = 6\theta$, where θ represents the hardware specific error range. In three out of ten repetitions, it was not possible to correctly detect the NLOS link during the first initialization. However, after some nodes autonomously disconnected and reconnected (see Figure 3 *parameters out of range*), a valid solution could be found. For the second set of experiments, we used a smaller threshold of $\xi_A = 4\theta$. In this configuration, the system detected the NLOS link with $100\,\%$ accuracy.

C. Overall System Performance

ALF has primarily been designed to provide self-localization services to mobile customers such as flying quadcopters. Unfortunately, it is not easily possible to accurately steer a quadcopter on a given trajectory to verify the localization system's accuracy. Thus, we used a toy train to simulate the quadcopter by moving the localization unit on a well defined trajectory. In our experiment, the sensing device was mounted on a stick at an altitude of $64\,\mathrm{cm}$. It is driving on a rectangularly shaped trajectory through an autonomously constructed reference grid consisting out of nine nodes. In this setup, straightforward and curve movement can be investigated. Figure 5 (right) shows a picture of the testbed. For NLOS measurements, we placed an obstacle in the middle of the set. This results in at least $30\,\%$ corrupted measurements. Similar values have been reported in the literature [20].

1) LOS: In a first experiment and as an reference, we removed the obstacle in the middle of the set. Figure 7 depicts the resulting coordinates for the client in form of a scatter plot for the top view. The train was driving 10 times around the railroad line with a speed of approx. $1\,\mathrm{m\,s^{-1}}$. The darker an area gets the more often the sensor was detected in this area. The rectangle of the railroad can clearly be identified.

As the mechanical construction on the train was slightly instable, the sensing hardware can swing a few centimeters on the top of the stick. These oscillations distort the integrated prediction filter, resulting in accepted incorrect positions. In contrast to our initial approach [6], we used a linear multistep method (the Adams-Bashforth-Method) to improve the predication in the curves. Measuring the actual position of the train on the ground is unfortunately too inaccurate,

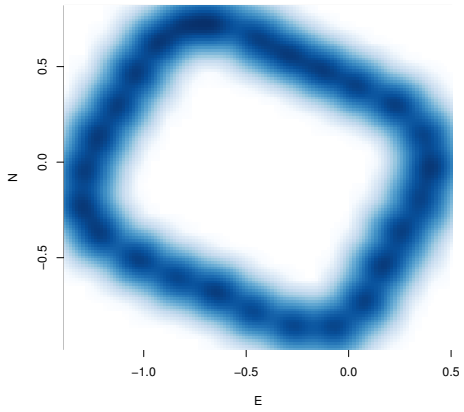

Fig. 7. Scatter plot of the localization results (LOS)

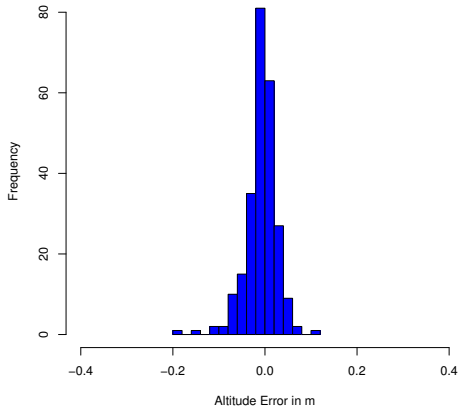

Fig. 8. Histogram of the altitude error (LOS)

because we had no odometry available with an accumulated error smaller than a few centimeter. Therefore, we evaluated the measured altitude. The altitude of the sensing device was exactly constant (64 cm; the altitude change due to the oscillations is negligible). A histogram for the altitude error is shown in Figure 8. As can be seen, the accuracy is very high. Numerical values presented in Table II indicate that 50 % of the measurements are within ± 2.2 cm of the correct altitude. The maximum measurement error was 19.1 cm.

Figure 9 shows individual measurements for a small part of the railroad. The upper plot depicts the altitude measures and estimations for different East-coordinates. The lower plots shows North-East-coordinates. In our experiment, the train drove counterclockwise. The original railroad as well as the fixed altitude are drawn in dashed green lines. In our plots, the big red dots depict accepted sampling points for the client localization.

TABLE II
ALTITUDE ERROR IN METER

Exp.	Min	1^{st} Qu.	Median	Mean	3^{rd} Qu.	Max
LOS	-0.191	-0.022	-0.005	-0.008	0.011	0.111
NLOS	-0.377	-0.025	-0.002	-0.009	0.020	0.348

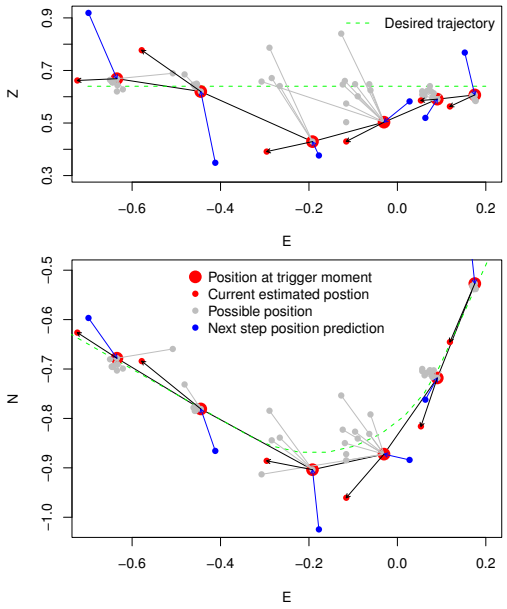

Fig. 9. Trajectory of the localization experiment

We collected all the measurement information approx. every 260 ms. Between these intervals, the train moves continuously. As soon as the latest sampling position is known to the system, the current position is estimated (represented in form of small red dots in Figure 9) based on the last sampling points and the elapsed time. This process works as follows: Before a sampling point is accepted, it is compared to the predicted position using the last sampling points (small blue dots in Figure 9). The best matching position out of all current measurements (small grey dots in Figure 9) is chosen as the new sampling point. More details on the estimation can be found in [6].

In the plot, it can easily be seen that both the prediction and the estimation process have certain inertia as they rely on the same sampling points. Therefore, as the train moves acound the corner, the outer measurements get chosen as sampling points (Figure 9, lower plot). This represents the worst case scenario in our experiment. We picked this example to show the following problem: In this situation, the altitude of the possible positions (grey) drops with the distance to the center of the railroad due to the probe reaches the border of the network as well as due to measurement errors (Figure 9, upper plot). We furthermore see that more suitable positions are available but never chosen according to the prediction model. Using clients such as typical quadcopter that provide internal sensors (e.g., gyroscopes and accelerometers), this information can be used to improve the prediction and to avoid those measurement errors.

2) NLOS: Figure 10 depicts the resulting coordinates of the NLOS scenario. We used the same conditions as described for the LOS measurements, except that we used an obstacle in the middle of the system. In contrast to the LOS case, more noise can be observed but the rectangle can still clearly be identified. However, four positions can be perceived outside

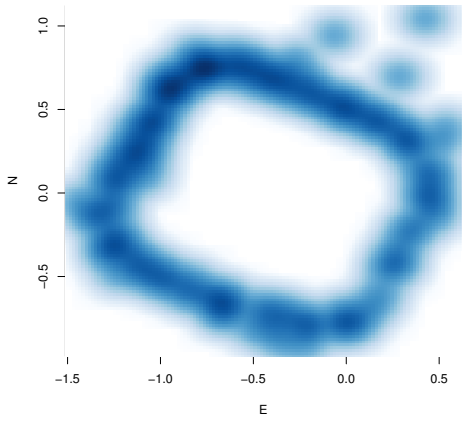

Fig. 10. Scatter plot of the localization results (NLOS)

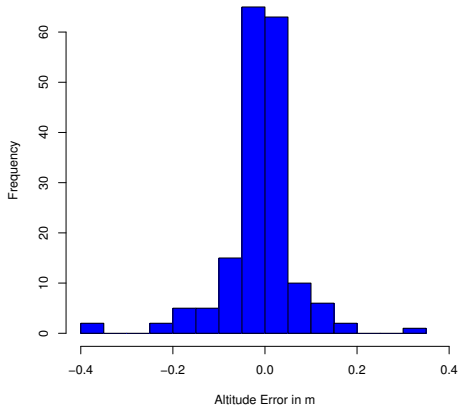

Fig. 11. Histogram of the altitude error (NLOS)

of the trajectory, which represent outliers in the localization experiment. In 2 out of 10 rounds, the position information got corrupted (actually, always in the same curve) due to significant oscillations (± 8 cm) in combination with NLOS.

A histogram showing the altitude error (Figure 11) allows to assess the localization accuracy. The numerical values in Table II indicate that 50 % of the measurements have an error of less than ± 2.5 cm. The significant outliers of nearly ± 40 cm are significant but happen statistically very infrequently.

V. Conclusion

We presented a fully decentralized and stateless localization framework, which is capable of autonomously spanning up a reference grid in unknown environments. Based on this grid, customers such as quadcopters can be accurately localized in real-time. Using the MSR theory, which is based on rather simple equations, we were able to design a system that does not need any *a priori* knowledge or a global database. Therefore, it can easily be used for embedded systems with limited energy and memory resources. In summary, it can be said that our Autonomous Localization Framework is providing very accurate localization accuracy. This also holds for handling NLOS situations, which is a strong requirement for real world applications.

References

[1] L. Hu and D. Evans, "Localization for Mobile Sensor Networks," in *10th ACM International Conference on Mobile Computing and Networking (MobiCom 2004)*. Philadelphia, PA: ACM, September 2004, pp. 45–57.

[2] L. Lazos and R. Poovendran, "SeRLoc: Robust localization for wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 1, no. 1, pp. 73–100, August 2005.

[3] J. Eckert, F. Villanueva, R. German, and F. Dressler, "A Self-Organizing Localization Reference Grid," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 14, no. 3, pp. 4–6, July 2010.

[4] S. Hochdorfer and C. Schlegel, "6 DoF SLAM using a ToF Camera: The Challenge of a Continuously Growing Number of Landmarks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*. Taipei, Twaiwan: IEEE, October 2010, pp. 3981–3986.

[5] J. Eckert, K. Koeker, P. Caliebe, F. Dressler, and R. German, "Self-localization Capable Mobile Sensor Nodes," in *IEEE International Conference on Technologies for Practical Robot Applications (TePRA 2009)*. Woburn, MA: IEEE, November 2009, pp. 224–229.

[6] J. Eckert, R. German, and F. Dressler, "An Indoor Localization Framework for Four-rotor Flying Robots Using Low-power Sensor Nodes," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 2, pp. 336–344, February 2011.

[7] N. B. Priyantha, "The Cricket Indoor Location System," PhD Thesis, Massachusetts Institute of Technology, June 2005.

[8] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, "The Anatomy of a Context-Aware Application," *ACM/Springer Wireless Networks*, vol. 8, pp. 187–197, March 2002.

[9] K. Chintalapudi, A. P. Iyer, and V. N. Padmanabhan, "Indoor Localization Without the Pain," in *16th ACM International Conference on Mobile Computing and Networking (MobiCom 2010)*, Chicago, IL, September 2010, pp. 173–184.

[10] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," in *19th IEEE Conference on Computer Communications (IEEE INFOCOM 2000)*, Tel-Aviv, Israel, March 2000.

[11] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller, "Mobile-Assisted Localization in Wireless Sensor Networks," in *24th IEEE Conference on Computer Communications (INFOCOM 2005)*, Miami, FL, March 2005, pp. 172–183.

[12] M. L. Sichitiu and V. Ramadurai, "Localization of Wireless Sensor Networks with a Mobile Beacon," in *1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*. Fort Lauderdale, FL: IEEE, October 2004, pp. 174–183.

[13] T. Tasaki, S. Tokura, T. Sonoura, F. Ozaki, and N. Matsuhira, "Mobile Robot Self-Localization Based on Tracked Scale and Rotation Invariant Feature Points by Using an Omnidirectional Camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2010)*. Taipei, Taiwan: IEEE, October 2010, pp. 5202–5207.

[14] D. A. Shell and M. J. Matari, "Directional Audio Beacon Deployment: an Assistive Multi-Robot Application," in *IEEE/RSJ International Conference on Robotics and Automation (ICRA 2004)*, New Orleans, LA, May 2004, pp. 2588–2594.

[15] I. Güvenç, C.-C. Chong, F. Watanabe, and H. Inamura, "NLOS identification and weighted least-squares localization for UWB systems using multipath channel statistics," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, p. 36, January 2008.

[16] Y. T. Chan, W. Y. Tsui, H. C. So, and P. C. Ching, "Time-of-Arrival Based Localization Under NLOS Conditions," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 17–24, January 2006.

[17] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-Free Distributed Localization in Sensor Networks," MIT Laboratory for Computer Science, Tech. Rep. TR-892, April 2003.

[18] A. A. Kannan, B. Fidan, and G. Mao, "Analysis of Flip Ambiguities for Robust Sensor Network Localization," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 2057–2070, May 2010.

[19] J. Eckert, F. Villanueva, R. German, and F. Dressler, "Considerations on Quality Metrics for Self-localization Algorithms," in *5th IEEE/IFIP International Workshop on Self-Organizing Systems (IWSOS 2011)*, vol. LNCS 6557. Karlsruhe, Germany: Springer, February 2011, pp. 104–115.

[20] R. Casas, A. Marco, J. Guerrero, and J. Falco, "Robust Estimator for Non-Line-of-Sight Error Mitigation in Indoor Localization," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 1–8, 2006.