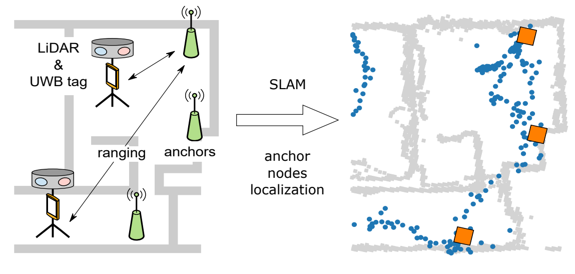
Static LiDAR Assisted UWB Anchor Nodes Localization

Marcin Kolakowski, *Student Member, IEEE*, Vitomir Djaja-Josko, *Student Member, IEEE* and Jerzy Kolakowski *Member, IEEE*

***Abstract*— The paper presents an indoor mapping and UWB (ultra-wideband) anchor localization method. The method, unlike most of the solutions described it the literature uses a static LiDAR (Light Detection and Ranging) mounted on a tripod rather than a robotic platform. It can be used at an any place, where employing a robot would be difficult (e.g. private homes), but since it requires manual LiDAR placement it would be most efficient in spaces of moderate areas. The proposed concept consists in mapping the environment of system installation while performing ranging measurements with deployed UWB anchor nodes. The SLAM (Simultaneous Location and Mapping) algorithm used for map integration and device localization relies only on LiDAR results. The matching is performed in two steps by finding an initial match based on corresponding landmarks extracted from the scans (intersections of the detected wall lines) and refining the results using an Iterative Closest Point algorithm. The anchors are localized based on the ranging results and SLAM-derived device locations using a Least-Squares based optimizer. The experiments have shown that the algorithm allows to construct a comprehensive map of the environment and localize the anchors with a root mean square error of 0.34 m, which is at similar level to analogous methods described in other works. The impact of anchor localization error on the systems performance was not significant. In both static and dynamic scenarios the difference in median errors obtained using reference and mapped anchors' locations was about 0.05 m.**



***Index Terms*— positioning, SLAM, UWB**

I. Introduction

DEPLOYING an indoor localization system at an unknown place might be a demanding task. It requires assessing the necessary number of anchors nodes, choosing their placement, installing them and accurately determining their coordinates in a chosen coordinate system. Typically, all of the steps including system infrastructure mapping are performed manually, which makes system installation a time-consuming task.

Manual mapping of the system infrastructure usually consists in determining relative anchor locations with respect to neighboring walls and then transforming them to a common coordinate system. This procedure does not pose much of a problem, when there are actual plans of the deployment area available. Unfortunately, such situation, due to frequent renovations of indoor spaces making the plans outdated, is not common and prior mapping of the environment is required. It can be performed either manually using a laser distance meter or automatically using SLAM-based (simultaneous localization and mapping) systems. As manual mapping requires making a large amount of measurements and the quality of the obtained maps rarely reflects the effort taken, employing SLAM-based solutions would be a better choice as they typically allow to achieve higher quality maps with lower effort.

A. Simultaneous localization and mapping methods

Typically, SLAM methods are applied in systems using mobile robotic platforms [1] equipped with 2D/3D LiDAR (Light Detection and Ranging) scanners [1] or vision cameras [2]. The robots are usually equipped with additional sensors allowing for their relative localization such as odometry wheels [1] or IMUs (Inertial Measurement Units) [3]. The indoor applications of SLAM solutions using sensors mounted on tripods [3] [2] are much less common.

The main reason behind this is that employing robotic platforms allows to automate the measurements and collect scans in a large number of points, which would rarely be collected manually due to time-constraints. Large number of scans and additional data from IMUs or odometry enable use of efficient SLAM methods and easier creation of high quality maps. In case of a LiDAR mounted on a tripod, the lack of additional sensors narrows the choice of applicable methods, which may make scan matching harder. However, using stationary LiDARs has some distinct advantages.

First of all, a LiDAR mounted on a regulated tripod allows to take scans at different heights, which is usually not possible

The research has been partially funded by the National Centre for Research and Development, Poland under Grant AAL2/2/INCARE/2018.

The authors are with Warsaw University of Technology, Institute of Radioelectronics and Multimedia Technology, Warsaw, Poland (e-mail: m.kolakowski@ire.pw.edu.pl)

in case of robotic platforms. That makes an opportunity to capture only the desired objects and avoid potentially problematic pieces of furniture such as mirrors. For example, in case of person localization, the scans taken at chest height would exclude low furniture such as sofas occupying places where the person can be present. Taking the scan at the same place and at different heights might also allow for extraction of room boundaries or particular furniture pieces, which might be used for different purposes (e.g presence detection). An another advantage of using stationary LiDARs is their lower overall cost. Moreover, a setup consisting of a sensor and a tripod is easier to transport, which is convenient in case of deploying localization systems at remote sites.

The use of a stationary scanner might also be justified for mapping smaller areas e.g. private flats, where the numbers of required scans would be moderate and acceptable to collect manually. Additionally, using the robots at such places might be hard or impossible due to large amount of objects cluttering the floor (carpets etc.) or the presence of pets. The authors experiences gained during deploying localization systems at several elderly people apartments confirm that [4].

As noted above, most of the developed SLAM algorithms utilize, along the registered scans, odometry and IMU data returned by the robots. The most notable examples include EKF-based (Extended Kalman Filter) SLAM [5], Rao-Blackwellized [6] particle filters and graph-based SLAM algorithms [1]. In case of stationary LiDARs, the matching can be performed solely based on landmarks extracted from the scans. Therefore, the scans should be taken with care to ensure that the consecutive scans capture enough landmarks to make scan fitting possible. Otherwise, matching far away scans might be highly ineffective [2].

One of the basic scan matching methods is Iterative Closest Point (ICP) algorithm [7], which consists in finding a transformation (translation and rotation) through minimizing the minimum distances between subsets of points of the matched scans. The method yields good results, but only if the matched scans are close to each other. When the scans are taken in distant points, they typically require prior matching. In most cases it is done based on corresponding landmarks extracted from both scans.

The landmarks used for scan matching are typically easy-to-detect objects like walls, corners or characteristic points (e.g. columns). The method presented in [8] extracts line segments from the scans, finds parallel segments and estimates rotation between the scans. Then it performs fitting for randomly selected segments and chooses the best match. A different method merging scans based on detected line segments is presented in [9].

In [10], the corners in matched maps are detected and used to construct corresponding polygons, which are basis for matching. The method allows to match two maps accurately. However, its use for singular scans might not be successful, since the number of the detected corners will usually be low. The landmarks can also be more complex objects. In [11] a method, where scan features are approximated and represented as implicit functions is proposed.

The methods where the matching is done without landmarks extraction are less common. Examples of such algorithms are presented in [12] and [13], where the scans are matched by analyzing the shift between their Hough transforms.

B. Infrastructure localization

The effort associated with deploying UWB systems can be reduced even more by employing algorithms allowing for automatic positioning of the infrastructure. In literature, there are several examples of methods, which combine SLAM with UWB-based ranging in order to map the environment and determine the unknown locations of the anchor nodes [14]–[18]. In those methods the anchors are positioned based on ranging performed between a tag attached to a robot and the anchors. The measurement locations are obtained with the attached SLAM systems and sensors.

Most of the described solutions utilize camera-based SLAM [14]–[17]. In [14] a monocular camera is mounted alongside an IMU and an UWB sensor on a drone, which flies a path covering the system deployment area. The UWB sensor performs concurrent ranging with the system infrastructure and when results are collected in a sufficient number of locations (100 in the described implementation), the anchors are localized using a Least-Squares based algorithm. An analogous set of sensors is used in [15], where the drone flies around the anchors and positions them based based on visual data (the anchors are labeled with fiducial markers).

A similar solution is presented in [16], where the method fuses data from a camera, an IMU and UWB-ranging results to localize an anchor added to the system infrastructure. It allows to easily expand the system, but since it requires information on the location of the already deployed anchors it can not be employed in the considered problem.

An interesting concept is presented in [17], where a drone equipped with a camera is used to deploy anchors in an unknown environment. The camera and drone sensors are used to get initial guess of the anchors positions. The estimates are then refined with a hybrid algorithm fusing visual data with UWB-ranging. The proposed concept was verified with simulations.

The solutions with sensors other than cameras are less common. In [18] a method combining UWB-ranging with results from a LiDAR is presented. In the proposed algorithm the anchors perform peer-to-peer ranging and their relative placement is determined using an Extended Kalman Filter. The resulting constellation is then located with respect to the created map based on results of ranging between the infrastructure and a tag attached to the robot. Adding peer-to-peer ranging between the anchors allows for efficient localization of the anchors. However, since it is not a common functionality of the UWB systems, the method would be hard to implement in the already existing solutions.

The following paper presents a method adopting a similar approach. The proposed method consists in mapping the environment, where the system is installed using a stationary LiDAR mounted on a tripod while performing ranging measurements with already deployed anchor nodes using an attached tag. The registered scans are matched and the sensor is localized based on landmarks being an intersections of lines covering the detected wall segments. The locations of the anchors are then determined based on UWB-ranging results using a Least-Squares based algorithm. The presented procedure results in a map of the system deployment area and coordinates of the system anchors.

The proposed method might be used in almost all indoor deployment scenarios. However, its advantages will be most visible, while deploying systems in spaces of moderate areas (private homes, smaller offices). Those areas are often cramped with furniture and using a robot for mapping might be not possible. Additionally the number of required scans would not be high, so manual sensor placement would not pose much of a problem. Moreover, as the method does not require any special functionality from the system (e.g. anchor-to-anchor ranging), it could be applied in virtually every ranging-based indoor positioning system.

II. Method concept

The proposed infrastructure localization method assumes using a device consisting of a laser scanner and an UWB tag mounted on a regulated tripod (Fig. 1). The scanner is used to map the area of the system deployment and position the device with respect to the created plan. The tag is used for concurrent ranging between the tag and anchors comprising system infrastructure.

An exemplary scenario of the proposed method's use is presented in Fig. 2. During the procedure, the device is placed in several points spread across the area of the system installation. In order to properly map the area and position the device, those points should satisfy the following conditions:

- from each of the points at least one of the other measurement points should be directly visible (it assures that the scans will have common parts)
- the distance between the points should be moderate (in [2], some 3-meter-apart scans required manual fitting).

The gathered scans are integrated using a SLAM algorithm to construct a whole map of the area and get the location of the measurement points. The obtained locations and tag-anchor ranging results are then used to localize the system infrastructure.

In general, in the proposed concept, it would be possible to determine both x,y coordinates and the elevation of the anchor. However, as the measurements would be typically taken with the tag mounted at similar heights, the resulting elevation might be very inaccurate or not unambiguous. Therefore, the method would benefit from manual anchor elevation measurement and using the results as additional input data.

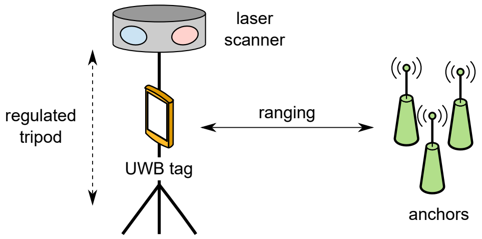


Fig. 1. Infrastructure deployment support system concept

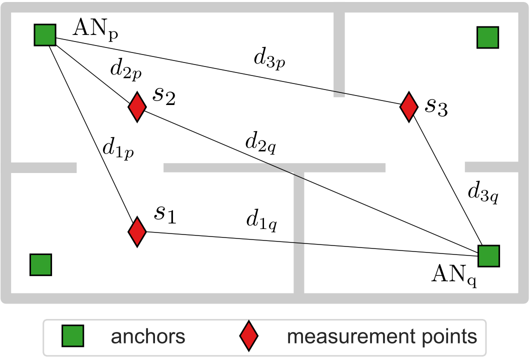


Fig. 2. An exemplary system deployment scenario ($\boldsymbol{d_{mn}}$ is a distance measured between the device in a test point $\boldsymbol{m}$ and an anchor $\boldsymbol{n}$)

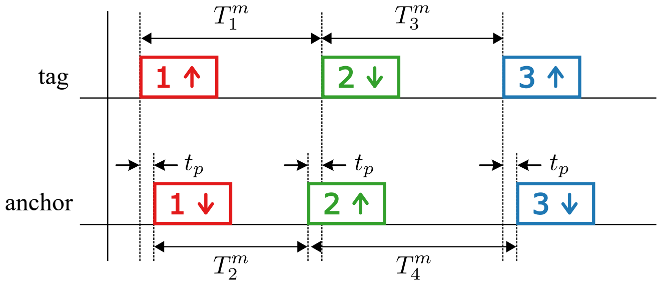


Fig. 3. An exemplary exchange of messages during SDS-TWR (Symmetric-Double-Sided Two-Way-Ranging) procedure

In the presented study, the concept was implemented in an UWB-based system [4]. The distances between the tag and the anchors were estimated using the popular SDS-TWR (Symmetrical Double Sided – Two Way Ranging) method. It relies on an exchange of messages between two devices to estimate propagation time between them and thus the distance.

An exemplary packet exchange and the measured values is presented in Fig. 3. The tag initiates ranging by sending a message addressed to a particular anchor node (ranging is performed with one anchor at a time), which registers the packet's time of arrival. After a predefined period T_2 the anchor replies with its own message. The tag, upon the reception of that message, measures its time of arrival and after T_3 time transmits the concluding packet. During the procedure both the tag and the anchor measure two time periods: between packet transmission and response reception (T_1^m and T_4^m respectively) and between packet reception and retransmission (T_2^m and T_3^m). The measured periods are affected by clock signal sources uncertainties. The measured values can be expressed as:

$$\begin{aligned} T_1^m &= T_1(1+\epsilon_{\mathrm{T}}) \qquad & T_2^m &= T_2(1+\epsilon_{\mathrm{AN}}) \\ T_3^m &= T_3(1+\epsilon_{\mathrm{T}}) \qquad & T_4^m &= T_4(1+\epsilon_{\mathrm{AN}}) \end{aligned} \tag{1}$$

where T_1, T_3, T_3, and T_4 are true period values and ϵ_{T} and ϵ_{AN} are used to describe stability of the tag's and anchor's clock signal sources. The values of ϵ_{T} and ϵ_{AN} depend on the clock signal source quality and in case of a typically used crystal oscillators might be even as high as 10^{-6} (10 ppm), which can lead to ranging biases of several meters. Based on Fig.3, the propagation time between the devices is:

$$t_p = \frac{T_1^m - T_2^m + T_4^m - T_3^m}{4} \tag{2}$$

Including the uncertainties (1) it can be expressed as:

$$\begin{aligned} t_p = & \frac{T_1 - T_2 + T_4 - T_3}{4} \\ & + \frac{(T_1 - T_3)\epsilon_{\mathrm{T}} + (T_4 - T_2)\epsilon_{\mathrm{AN}}}{4} \end{aligned} \tag{3}$$

If the following conditions are fulfilled: $T_1 \approx T_3$ and $T_2 \approx T_4$ (the periods are large in comparison to the propagation time), the measurement error caused by the tag's and anchor node's clock's uncertainties may be significantly reduced.

III. Algorithms

A. SLAM algorithm

The proposed system deployment support method consists of two steps: environment mapping and infrastructure localization. The former is performed using a scan matching algorithm, which basic steps are reported as Algorithm 1.

The input data for the scan matching algorithm are LiDAR scans $S_1 \ldots S_n$ registered in measurement points $s_1 \ldots s_n$ and a scan fitting sequence comprising pairs of consecutive scans, which will be fit to each other. The procedure starts with preprocessing of each of the collected scans. Then for each pair in the sequence, a transformation $t_{n,m}$ (translation and rotation of scan S_n with respect to scan S_m) is found. The last step is the integration of the scans into a complete map of the area.

The preprocessing consists in grouping the results by measurement angle and determining measured distance median for each group. It reduces the size of the scan and lowers the measurement noise. The resulting scan is converted to x-y coordinate system and landmarks, which will be used for initial matching are extracted as described in Algorithm 2.

The first step is extraction of wall segments from the scan and approximating them with straight lines by first degree polynomial fitting (the method estimates slope and intercept). For that purpose any method extracting line segments from the scan (e.g. Hough-based [19]) may be used. In the presented implementation the walls are detected in a similar fashion to [8] by analyzing consecutive scan points sorted in a clockwise order and building lines from the closely spaced points.

The detected lines are used to extract landmarks which will be used to initially match the scans. In the proposed method the landmark points are intersections of the detected lines. The resulting landmark sets include both wall intersections and virtual ones, which do not exist in real environment. Such approach may enlarge the total number of corresponding landmarks, which in case of singular scans might be limited. An example of detected lines and landmark points is presented in Fig. 4.a.

The presented scans are quite simple as the areas, where they were taken do not include a lot of furniture. In case of cluttered spaces, a number of detected lines might be substantial, which would result in a large landmark set and increase algorithms computational demands. The number of landmarks can be lowered without significant information loss by clustering closely-spaced points or imposing additional conditions such as minimum length of the detected line segments, which are used to calculate the landmark locations. In the presented study, only walls, which were longer than 0.3 m were considered (shorter lines detected in the scans resulted from cluttering objects and room corners). The obtained landmarks are then used for initial scan matching reported as Algorithm 3.

Algorithm 1 Scan matching algorithm

```
Input: scans $S_1 \ldots S_n$, scan matching sequence
Output: a map $S_{MAP}$, measurement locations $s_1 \ldots s_n$
for each scan $S_m$ do
    reduce scan size and noise
    convert to x-y coordinates
    $L_m \leftarrow$ detect landmarks                ▷ see Algorithm 2
for each pair of scans $S_m - S_n$ in sequence do
    $T_{n,m} \leftarrow$ initial matching            ▷ see Algorithm 3
    $t_{n,m} \leftarrow$ refined matching            ▷ see Algorithm 4
$S_{MAP}, s_1 \ldots s_n \leftarrow$ integrate the scans
```

Algorithm 2 Landmark extraction algorithm

```
Input: scan $S$
Output: landmark points $L$
$W \leftarrow$ wall segments in $S$ approximated with lines
$L \leftarrow [\,]$
for wall line in $W$ do
    $I \leftarrow$ intersections with the lines remaining in $W$
    add the points $I$ to $L$
$L \leftarrow$ cluster the points in $L$
```

The initial matching consists in finding three corresponding landmark points in both scans and using a Least-Squares based algorithm estimating the translation and rotation between them. The sought transform $t_{n,m}$ is defined as:

$$t_{n,m} = [t_x, t_y, \theta] \tag{4}$$

$$S_{n,t} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} S_n - [t_x, t_y] \tag{5}$$

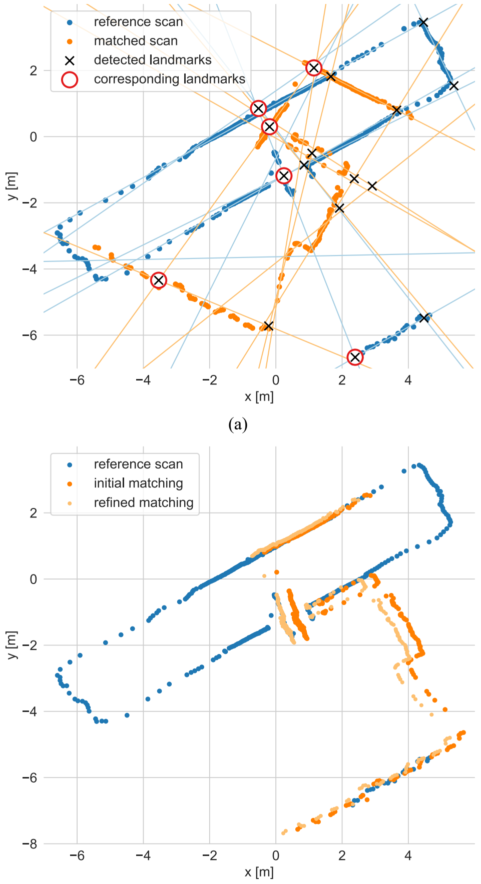
where $S_{n,t}$ is a transformed scan S_n matched to reference scan S_m, t_x and t_y are the translations of the measurement device along x,y directions and θ is the rotation between the scans (difference in initial scanner orientation in both measurement points).

The procedure starts with getting all possible 3-point combinations for both landmark sets. For each combination interpoint distances are calculated and saved alongside. Then the combinations from both sets are analyzed in search of threes of points, which may include the same landmarks. The procedure compares the inter-point distances and if the difference between them is within a specified range (in the presented implementation it was set to 0.1 m) both landmark combinations are saved as a tuple to corresponding landmarks set C_L. An example of corresponding landmarks is presented in Fig. 4.a.

The number of the corresponding landmarks used for matching could be higher as in [10], where complex polygons are considered. Such an approach might result in better accuracy but would be possible to adopt only in certain situations (e.g. scans made in the same room). In the considered case, the scans might be taken in distant points located in separate rooms and the number of corresponding landmarks would be typically lower due to different walls being captured (Fig. 4.a). However, due to the assumed tolerance, the algorithm still could erroneously return a larger set of corresponding landmarks resulting in bad matching and shadow the correct solution. To avoid such situations, the number of sought landmarks was set to three.

The resulting landmark sets are used to calculate the translation and rotation between the scans. An exemplary result of initial matching is presented in Fig. 4.b. Due to the fact that the landmarks for both scans are usually determined based on different wall segments and the algorithm allows dimension tolerance, the initial fit might not be perfect.

The obtained results are analyzed to find transformations, for which the scans are closely aligned and the matching could be refined using an iterative closest point (ICP) algorithm. It is done by calculating the closest point distance CPD (smallest distance between a point and a set of points) between the matched scans. The algorithm chooses five translations with the highest number of points which closest point distance is less than d_{max}. In the described implementation d_{max} was set to 0.4 m. Using lower values, the algorithm might favor transformations, for which a small number of points is aligned (e.g. erroneous fitting of one wall only). The number of translations passed to the refined matching procedure might be higher. However, the initial tests have shown that in case of all matched scan pairs taking the top five was enough to achieve the desired result.



(a)

(b)

Fig. 4. Matched scans at different stages or the algorithm: a) corresponding landmarks extraction, b) refined fitting

Algorithm 3 Initial matching procedure

```
Input: Landmarks $L_m, L_n$ and scans $S_m, S_n$
Output: initial transformations $T$
$P_m, P_n \leftarrow$ all 3-point combinations for $L_m$, $L_n$ and distances between the points
$C_L \leftarrow$ find corresponding 3-point combinations in $P_m$, $P_n$ sets (same distances criterion)
$T \leftarrow [\,]$, $scores \leftarrow [\,]$
for corresponding landmark sets $c_m$, $c_n$ in $C_L$ do
    for $i$ in $\{0, 1, 2\}$ do
        $t_{n,m} \leftarrow$ find transformation matching $c_n$ to $c_m$ shifted by $i$
        add $t_{n,m}$ to $T$
for $t$ in $T$ do
    $S_{n,t} \leftarrow$ rotate and translate $S_n$ by $t$
    $x \leftarrow$ number of $S_{n,t}$ points, which $CPD < d_{max}$
    add $x$ to $scores$
sort $T$ by scores
$T \leftarrow T[:5]$                                  ▷ return 5 best matches
```

Algorithm 4 Refined matching procedure

```
Input: scans $S_m, S_n$ and initial transformations $T$
Output: transformation $t$
$T \leftarrow [\,]$, $scores \leftarrow [\,]$
for $t$ in $T$ do
    $S_{n,t} \leftarrow$ rotate and translate $S_n$ by $t$
    $P_f \leftarrow S_{n,t}$ points, which $CPD < d_{max}$
    $t \leftarrow$ find transformation based on $P_f$ and $S_m$
    add $t$ to $T_f$
    $S_{n,t} \leftarrow$ rotate and translate $S_n$ by $t$
    $x \leftarrow$ number of $S_{n,t}$ points, which $CPD < d_{max}$
    add $x$ to $scores$
sort $T_f$ by scores
$t \leftarrow T[0]$                                   ▷ return the best match
```

In the proposed implementation, the refined matching is performed based on the iterative closest point algorithm using a Least-Squares (LS) based optimizer minimizing distance between scan points, for which CPD is lower than a set threshold d_{max} (0.4 m in the performed tests). The LS problem was solved using the Levenberg-Marquardt method.

The obtained transformations are then ranked in the same fashion as in case of initial matching and the best one is chosen as the final result. The example of the fitted scans is presented in Fig. 4.b.

The last part of the scan matching procedure is integrating the registered scans into one, complete map S_{MAP} of the system deployment area. It consists in transforming the obtained rotations and translations to obtain values relative to the first scan of the sequence. It is done by performing appropriate translations and rotations:

$$t_x^{(n,0)} = t_x^{(m,0)} + t_x^{(n,m)} \cos\theta^{(m,0)} + t_y^{(n,m)} \sin\theta^{(m,0)} \tag{6}$$

$$t_y^{(n,0)} = t_y^{(m,0)} + t_y^{(n,m)} \cos\theta^{(m,0)} - t_x^{(n,m)} \sin\theta^{(m,0)} \tag{7}$$

$$\theta^{(n,0)} = \theta^{(m,0)} + \theta^{(n,m)} \tag{8}$$

where $t_x^{n,m}$, $t_y^{n,m}$, $\theta^{n,m}$ are translation and rotation of scan n relative to scan m (matching result) and $t_x^{n,0}$, $t_y^{n,0}$, $\theta^{n,0}$ are relative to the first scan of the sequence. Finally, the locations of the measurement points n can be expressed as:

$$s_n = \left[t_x^{(n,0)}, t_y^{(n,0)}\right] \tag{9}$$

The obtained transformations are imposed on the scans and a complete map of the system deployment area is created. The locations of the measurement points are then used by infrastructure localization algorithm.

B. Infrastructure localization

The second step of the proposed system deployment support method is localization of the system infrastructure. An exemplary scenario with measured values marked is presented in Fig. 2.

In the proposed concept, the anchors are localized based on distance measurements (d_{mp}) performed by the tag placed in several points (s_m) distributed in the area covered by the system. The location of those points is derived using a SLAM algorithm described in Section III-A. In addition to the distances, the algorithm requires information on anchors' elevation. Measuring height, at which the anchor is fixed does not significantly increase the installation workload and would allow to increase anchor positioning accuracy.

In the proposed method, the anchors are localized separately using a Least-Squares based optimizer implemented using the Levenberg-Marquardt method. The optimizer minimizes the difference between the measured tag-anchor distance and the one predicted by the algorithm:

$$\min_{a_x,a_y} \sum_{i=0}^{n-1} (\|a_p - s_i\| - d_{ip})^2 \tag{10}$$

where a_p is the anchor p location and a_x, a_y are its sought x,y coordinates (a_z is measured manually at deployment and is supplied as an input to the algorithm), s_i are the locations of the measurement points and d_{ip} is the measured distance between them and the anchor. The obtained anchor locations are then saved and supplied to the localization system controller.

IV. Experiments

The proposed concept has been experimentally tested in a hybrid BLE (Bluetooth Low Energy)-UWB localization system developed at Warsaw University of Technology [4]. The UWB part of the system is primarily TDOA (Time Difference of Arrival) based, so utilizing the method required implementing SDS-TWR ranging procedures between the tag and the anchors. Ranging results were transmitted by the anchors over WiFi to a localization server installed on a computer. The obtained measurement data were processed using a Python-based implementation of the proposed method.

The experiment was performed at the office space of the university. The area, comprised three rooms and an adjacent corridor. The system used in the study consisted of 7 anchors, which were placed on the walls and tripods. The layout of the area and localization of the anchors are presented in Fig. 5.a. The performed experiment consisted of two phases:

- environment mapping and infrastructure localization,
- assessing an impact of anchors localization errors on system's performance.

The scans and tag-anchor distance measurements were taken in 18 points distributed across the rooms. The locations of the points are presented in Fig. 5.a. In each point, one-minute long measurement was taken. The area scans were captured using a Scanse Sweep LiDAR sensor (as of 2020 retired and out of stock) working with 2 Hz rotation and 500 Hz sampling rates. The system tag allowed for ranging with a rate of 10 Hz. Given that there were 7 anchors to range with, the distance from each anchor was measured approximately 100 times. The anchor nodes' locations were determined based on the averaged values. The results of the environment mapping and infrastructure localization are presented in Fig.5.a.

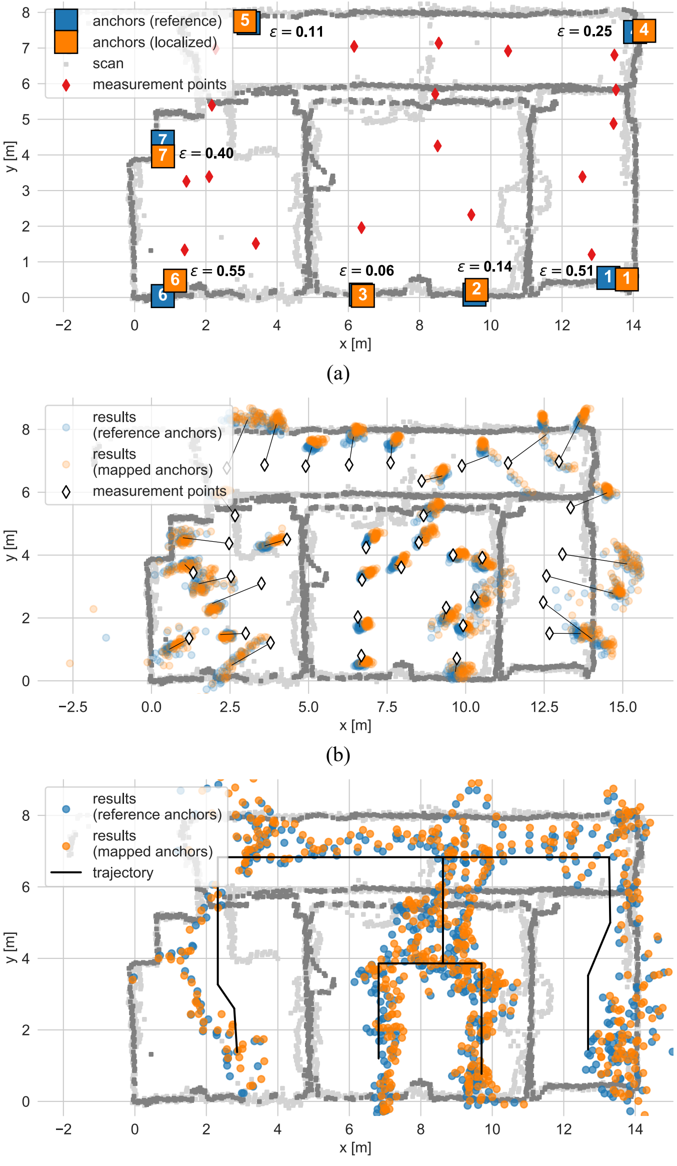
The gathered scans were matched appropriately, which resulted in a comprehensive map of the environment. Mounting the scanner on the regulated tripod allowed to capture scans without furniture, which can be used to determine room boundaries (marked with dark grey) often required by presence detection algorithms.

The calculated anchor locations were compared to reference data obtained using a laser distance meter. The localization errors of anchors are presented alongside the results in Fig. 5.a. An accuracy comparison of the proposed method and solutions described in the literature is presented in Table I.

The best localization accuracy was achieved in case of the anchors placed in the middle of the area, where the error did not exceed 0.15 cm. The worst results were obtained for anchors located in the corners of the outer rooms. The maximum localization error was about 0.55 m. It might result from an unfavorable measurement points configuration (the points were placed along a line and did not surround the localized anchors) and the fact that due to wall attenuation the anchors were unreachable from some points and the number of measured distances was lower. Also the SDS-TWR measurements performed between the tag and the anchor separated by the wall may give biased results due to introduced delays.

The root mean square error (RMSE) of anchor localization was about 0.34 m. At first sight the accuracy of the proposed method is worse than of the ones described in the literature (Table I). However, the results with the lowest reported errors, are not entirely reliable as they were obtained via simulations [17] and for localization of a single anchor [16]. The difference in accuracy between the proposed method and solutions working in a similar fashion [14], [18] is not much and might simply result from a fact, that the anchor localization was determined based on measurements collected in a smaller number of points (18 vs 100 in [14]).

The effect of the infrastructure localization errors was assessed in the second part of the experiment, which consisted in localizing static tags and a moving person. In both scenarios, localization was performed using a hybrid RSS (Received Signal Strength)-TDOA Unscented Kalman Filter based algorithm [4] based on reference and determined anchor locations. The results are presented in Fig. 5b-c. Comparisons of Empirical Cumulative Distribution Functions (ECDF) for mean localization and trajectory errors are presented in Fig. 6. The trajectory error was defined as a closest distance of a localization result to the reference movement trajectory.



(a)

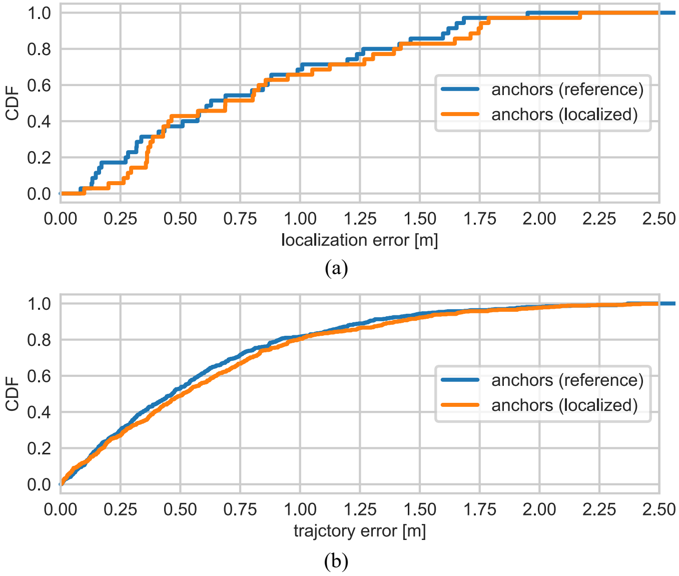
(b)

(c)

Fig. 5. The experiment results: a) anchor nodes localization, ϵ denotes a localization error of particular anchors. Room boundaries determined by performing scans at 2.5 m height are marked with dark grey. b) static tags localization, c) moving person localization

TABLE I
ANCHOR LOCALIZATION RMSE COMPARISON

Method	RMSE [m]
LiDAR, UWB-ranging (proposed method)	0.34
Camera, IMU, UWB-ranging [14]	0.25–0.48
Camera, IMU [15]	0.07–0.12
Camera, IMU, UWB-ranging [16]	0.06
Camera, IMU, UWB-ranging [17]	0.03
LiDAR, UWB-ranging [18]	0.07–0.21



(a)

(b)

Fig. 6. Empirical cumulative distribution functions: a) localization error of static tags, b) trajectory error during localization of a moving person

In both static and dynamic case, the localization accuracy based on reference and determined anchor locations is at similar levels. In case of static tag localization the distance between the results is at a level of several centimeters. The trajectories obtained in dynamic scenario are slightly shifted but their shape is preserved and in they can be used to determine the users whereabouts and path that he/she walked.

The whole procedure of gathering data for mapping and infrastructure localization took about 25 minutes (sensor placement, data gathering and copying). It is possible to lower that time by shortening measurement duration in each of the points. Given that the SDS-TWR are usually precise, the lower number of ranging samples should not negatively impact the method's performance.

V. CONCLUSIONS

The UWB anchor positioning method proposed in the paper adopts an approach, where the anchors are localized based on ranging results between them and a tag. The main difference from the solutions described in the literature is that the tag is mounted on a tripod rather than a mobile robotic platform or a drone. The method can find its application in places, where using a robot would be hard or impossible. Due to a need of manual tripod placement, it would be most efficient in case of moderate areas such as apartments or small office places.

The method utilizes a scan matching algorithm designed to fit scans based solely on range data supplied by the LiDAR. Unlike most of the described methods it does not use any additional data allowing for establishing geometric constrains between the scans. The algorithm could find its application in other different non-infrastructure localization scenarios, for example in mapping indoor spaces without use of robotic platforms.

The accuracy of the proposed anchor localization method is at a similar level to those reported in the literature. Additionally, the experiments have shown that the anchor localization errors do not significantly impact positioning accuracy. This shows that the proposed method might be considered as an alternative solution to the methods proposed in other works and can be successfully used to reduce the workload associated with deploying indoor positioning systems at unknown locations.

The proposed method shows potential for future development. The anchor localization algorithm could be improved by introducing Non-Line of Sight working conditions detection and mitigation, which could improve overall positioning accuracy. The scan matching algorithm could also be optimized and other type of landmarks could be extracted to make the fitting more accurate.

References

[1] H. Gao, X. Zhang, J. Wen, J. Yuan, and Y. Fang, “Autonomous Indoor Exploration Via Polygon Map Construction and Graph-Based SLAM Using Directional Endpoint Features,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1531–1542, Oct. 2019.

[2] Y. Chen, C. Jiang, L. Zhu, H. Kaartinen, J. Hyyppa, J. Tan, H. Hyyppa, H. Zhou, R. Chen, and L. Pei, “SLAM Based Indoor Mapping Comparison:Mobile or Terrestrial ?” in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*. Wuhan: IEEE, Mar. 2018, pp. 1–7.

[3] Y. Chen, J. Tang, C. Jiang, L. Zhu, M. Lehtomäki, H. Kaartinen, R. Kaijaluoto, Y. Wang, J. Hyyppä, H. Hyyppä, H. Zhou, L. Pei, and R. Chen, “The Accuracy Comparison of Three Simultaneous Localization and Mapping (SLAM)-Based Indoor Mapping Technologies,” *Sensors*, vol. 18, no. 10, p. 3228, Oct. 2018.

[4] J. Kolakowski, V. Djaja-Josko, M. Kolakowski, and K. Broczek, “UWB/BLE Tracking System for Elderly People Monitoring,” *Sensors*, vol. 20, no. 6, p. 1574, Mar. 2020.

[5] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, “Consistency of the EKF-SLAM Algorithm,” in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2006, pp. 3562–3568.

[6] F. Nie, W. Zhang, Z. Yao, Y. Shi, F. Li, and Q. Huang, “LCPF: A Particle Filter Lidar SLAM System With Loop Detection and Correction,” *IEEE Access*, vol. 8, pp. 20 401–20 412, 2020.

[7] G. Jiang, L. Yin, G. Liu, W. Xi, and Y. Ou, “FFT-Based Scan-Matching for SLAM Applications with Low-Cost Laser Range Finders,” *Applied Sciences*, vol. 9, no. 1, p. 41, Jan. 2019.

[8] F. Amigoni, S. Gasparini, and M. Gini, “Building Segment-Based Maps Without Pose Information,” *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1340–1359, Jul. 2006.

[9] B. Sarkar, P. K. Pal, and D. Sarkar, “Building maps of indoor environments by merging line segments extracted from registered laser range scans,” *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 603–615, Apr. 2014.

[10] Y. Sun, R. Sun, S. Yu, and Y. Peng, “A Grid Map Fusion Algorithm Based on Maximum Common Subgraph,” in *2018 13th World Congress on Intelligent Control and Automation (WCICA)*. Changsha, China: IEEE, Jul. 2018, pp. 58–63.

[11] J. Zhao, L. Zhao, S. Huang, and Y. Wang, “2D Laser SLAM With General Features Represented by Implicit Functions,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4329–4336, Jul. 2020.

[12] A. Censi, L. Iocchi, and G. Grisetti, “Scan matching in the hough domain,” in *In Proc. of Intern. Conference on Robotics and Automation (ICRA’05*, 2005.

[13] D. Graovac, S. Jurić-Kavelj, and I. Petrovic, “Mobile robot pose tracking by correlation of laser range finder scans in Hough domain,” Jul. 2010, pp. 273–278.

[14] T. H. Nguyen, T.-M. Nguyen, and L. Xie, “Tightly-coupled ultra-wideband-aided monocular visual SLAM with degenerate anchor configurations,” *Autonomous Robots*, vol. 44, no. 8, pp. 1519–1534, 2020.

[15] P. Lutz, M. J. Schuster, and F. Steidle, “Visual-inertial SLAM aided estimation of anchor poses and sensor error model parameters of UWB radio modules,” in *2019 19th International Conference on Advanced Robotics (ICAR)*. Belo Horizonte, Brazil: IEEE, 2019, pp. 739–746.

[16] X. Xu, X. Liu, B. Zhao, and B. Yang, “An Extensible Positioning System for Locating Mobile Robots in Unfamiliar Environments,” *Sensors*, vol. 19, no. 18, p. 4025, Sep. 2019.

[17] Q. Shi, X. Cui, W. Li, Y. Xia, and M. Lu, “Visual-UWB navigation system for unknown environments,” in *31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, Miami, Florida, 2018, pp. 3111-3121.

[18] Y. Song, M. Guan, W. P. Tay, C. L. Law, and C. Wen, “UWB/LiDAR Fusion For Cooperative Range-Only SLAM,” in *2019 International Conference on Robotics and Automation (ICRA)*. Montreal, QC, Canada: IEEE, May 2019, pp. 6568–6574.

[19] A. Bandera, J. Pérez-Lorenzo, J. Bandera, and F. Sandoval, “Mean shift based clustering of Hough domain for fast line segment detection,” *Pattern Recognition Letters*, vol. 27, no. 6, pp. 578–586, Apr. 2006.

Marcin Kolakowski received the B.S. and the M.S. degrees in telecommunications from Warsaw University of Technology (WUT), Warsaw, Poland in 2015 and 2016 respectively. He is currently pursuing the Ph.D. degree in informatics and telecommunications at Warsaw University of Technology. Since 2020 he is working as a Research Assistant at WUT. His main scientific interests include hybrid localization algorithms, data science and radiocommunications.

Vitomir Djaja-Josko received the BSc (’13) and M.Sc (’15) degrees in telecommunications from the Faculty of Electronics and Multimedia Technology, Warsaw University of Technology, Poland. He is pursuing the Ph.D. degree in the Institute of Radioelectronics and Multimedia Technology, where he holds a position of an assistant. His research interests include UWB signals, indoor localization and synchronization methods in positioning systems.

Jerzy Kolakowski received the M.Sc. and Ph.D. degrees in telecommunications from Warsaw University of Technology in 1988 and 2000 respectively. Since 1988 he has been with the Institute of Radioelectronics and Multimedia Technology where he holds a position of Assistant Professor. His current research interests include positioning systems, UWB technology, cellular systems