Experiments on Robust Indoor Localization of Mobile Devices Using Interval Arithmetic

Stefania Monica, Federico Bergenti
Dipartimento di Scienze Matematiche, Fisiche e Informatiche
Università degli Studi di Parma
43124 Parma, Italy
{stefania.monica,federico.bergenti}@unipr.it

***Abstract*—The lack of techniques and tools to estimate the position of mobile devices with high accuracy and robustness is one of the major causes that limit the provision of advanced location-based services in indoor environments. An algorithm to enable mobile devices to estimate their positions in known indoor environments is discussed in this paper under the assumption that fixed network nodes are available at known locations. The discussed algorithm is designed to allow devices to estimate their positions by actively measuring the distances from visible fixed nodes. In order to reduce the errors introduced by the arrangement of the fixed nodes in the environment, the discussed algorithm transforms the localization problem into an optimization problem, which is then solved using interval arithmetic. Experimental results on the use of the discussed algorithm with distance estimates obtained using ultra-wide band are presented to assess the performance of the algorithm and to compare it with a reference alternative. Presented experimental results confirm that the discussed algorithm provides an increased level of robustness with respect to the considered reference alternative with no loss of accuracy.**

***Index Terms*—localization as optimization, indoor localization, ultra-wide band, software agents**

I. Introduction

The possibility of robustly associating an accurate position with a mobile device in an indoor environment has recently become a major feature needed to increase the level of personalization of services and applications offered to users (e.g., [1]–[3]). Robust and accurate indoor localization can be used, for example, to personalize and increase the interactivity of visits to art exhibitions by means of educational games (e.g., [4]), it can be used to support the automation of work in warehouses (e.g., [5]), and it can be used to extend the services of location-based social networks (e.g. [6]) to large indoor areas like shopping malls, train stations, and airports. Solutions to problems related to indoor localization can be found in recent literature (e.g., [7]–[9]), but the general problem of enabling a mobile device to accurately and robustly estimate its position in a known environment is still an open issue (e.g., [10], [11]) even if specific technologies like *Ultra-Wide Band* (*UWB*) (e.g., [12], [13]) are readily available.

Software agents are expected to play a relevant role in the delivery of mentioned services and applications (e.g., [14]) because agents are normally characterized as situated entities capable of exhibiting context-aware behaviors. With this respect, the position can be considered as just another context information that agents can use to bring about their goals. Therefore, the use of the position to enrich the behaviors of agents is expected to be simple and effective. This is the reason why the algorithm used to run the experiments documented in this paper was implemented as an extension of the localization add-on module (e.g. [15]) for JADE (e.g., [16]). The localization add-on module for JADE is responsible for providing timely information to agents regarding their positions using one of the available localization algorithms. Actually, the localization add-on module extends a JADE container (e.g., [16]) executed on a mobile device with the possibility of interfacing the sensors of the device to obtain the low-level information needed to apply the chosen localization algorithm. The estimated position of the device is then passed to interested agents hosted by the container so that they can use it to bring about their goals. Note that, besides available localization algorithms, additional algorithms like the one discussed in this paper can be easily integrated in the add-on module provided that they implement a specific Java interface.

The discussions in this paper are focused on the general problem of allowing a mobile device, denoted as *Target Node* (*TN*), to estimate its position inside an indoor environment with known geometry under the assumption that fixed network nodes, denoted as *Anchor Nodes* (*ANs*), are installed at known locations. One of the major assumption of this work is that mobile devices can actively measure the distances from the ANs installed in the environment using one of the available ranging technologies. An average error of 1 m in the estimation of positions is considered acceptable for targeted application scenarios, and experimental results discussed in Section V show that such an accuracy is feasible using the discussed localization algorithm when ranging information are obtained using UWB. With minor loss of generality, it is assumed that the considered indoor environment is composed of possibly overlapping rectangular cuboids, called boxes in this paper to follow a consolidated nomenclature. Under this assumption, it is possible to focus the hunt for TNs in the environment on the single boxes that compose the environment, and therefore the localization problem can be simplified by considering only box-shaped environments. Observe that the minimum bounding box containing the considered environment can be used if the environment cannot be split into boxes, but, in this case, estimated positions of TNs should be filtered accordingly.

One of the problems that make indoor localization difficult is related to the fact that the positions of ANs in the environment strongly affect the accuracy of ordinary localization algorithms (e.g., [17], [18]), like the *Two-Stage Maximum-Likelihood* (*TSML*) [19] algorithm discussed in Section V. In particular, a significant loss of the accuracy provided by such algorithms is normally observed when ANs are not positioned properly in the environment. Such a problematic characteristic of ordinary localization algorithms is critical for indoor environments because ANs are normally located near the ceilings of rooms, which is an arrangement that exacerbates the mentioned problem. The choice of installing all ANs at (roughly) the same height near the ceilings of rooms degrades the performance of ordinary localization algorithms because some of the matrices involved in computations become strongly ill-conditioned (e.g., [5], [20]). Unfortunately, the choice of the positions of ANs in indoor environments is often dictated by practical considerations intended, for example, to limit the interference caused by people and furniture, and to maximize coverage. Therefore, the positions of ANs cannot be changed just to limit the mentioned problem of ordinary localization algorithms. The remainder of this paper is devoted to study the robustness of an alternative algorithm in situations in which the accuracy of ordinary algorithm degrades sensibly. Experimental results presented in Section V show that the accuracy of the studied algorithm is acceptable even when the accuracy of the TSML algorithm, which is used as a point of reference, degrades sensibly.

This paper is organized as follows. Section II introduces adopted notation. Section III describes the adopted approach to localization. Section IV describes the discussed algorithm and briefly presents its characteristics. Section V reports the results of the experimental campaign performed to assess the robustness of the studied algorithm. Finally, Section VI concludes the paper.

II. Notation

This section summarizes the notation commonly used to study real functions of several real variables, as it is used to present the localization problem and the discussed algorithm in the following sections. The set of natural numbers, including zero, is denoted as $\mathbb{N}$ and the set of positive natural numbers is denoted as $\mathbb{N}_+$. Given $n \in \mathbb{N}_+$, a multi-index $I \in \mathbb{N}^n$ is defined as an $n-$tuple of natural numbers

$$I = (i_1, i_2, \ldots, i_n) = (i_k)_{k=1}^n. \tag{1}$$

Note that it is a common notation to use an uppercase letter to denote a multi-index and to use the same letter, but lowercase and with a subscript, to denote its components. The following notation is adopted to use a multi-index $I \in \mathbb{N}^n$ as an exponent for $\mathbf{t} \in \mathbb{R}^n$ with $\mathbf{t} = (t_1, t_2, \ldots, t_n) = (t_k)_{k=1}^n$

$$\mathbf{t}^I = \prod_{k=1}^{n} t_k^{i_k}. \tag{2}$$

Note that it is a common notation to use a boldface letter to denote an $n-$tuple of real numbers, and the same letter, but lightface and with a subscript, for its components. Also note that the common understanding of $0^0 = 1$ is adopted consistently throughout this paper.

Given two multi-indices $I \in \mathbb{N}^n$ and $J \in \mathbb{N}^n$, they can be compared using the following partial order

$$I \leq J \iff i_1 \leq j_1 \wedge i_2 \leq j_2 \wedge \cdots \wedge i_n \leq j_n \tag{3}$$

and they can also be added componentwise

$$I + J = (i_1 + j_1, i_2 + j_2, \ldots, i_n + j_n). \tag{4}$$

The following abbreviation is introduced for multi-indices $H \in \mathbb{N}^n$, $I \in \mathbb{N}^n$, and $J \in \mathbb{N}^n$ to express repeated sums

$$\sum_{H \leq I \leq J} (\cdot) \quad = \quad \sum_{i_1=h_1}^{j_1} \sum_{i_2=h_2}^{j_2} \cdots \sum_{i_n=h_n}^{j_n} (\cdot). \tag{5}$$

Note that H is omitted in (5), so that only $I \leq J$ remains, if it is the null multi-index $(0)_{k=1}^n$.

The multi-index notation is particularly well-suited to study polynomial functions. A polynomial function $p : \mathbb{R}^n \mapsto \mathbb{R}$ of $n \in \mathbb{N}_+$ real variables with real coefficients can be written as

$$p(\mathbf{x}) = \sum_{I \leq L} a_I \mathbf{x}^I, \tag{6}$$

where $\{a_I\}_{I \leq L} \subset \mathbb{R}$ is the set of the coefficients of p and multi-index $L \in \mathbb{N}^n$ is the multi-degree of p.

Observe that the localization algorithm discussed in this paper works by studying the bounds of polynomial functions over boxes. Hence, it is worth recalling the following notation for intervals and boxes. A closed (real) interval from $\underline{a} \in \mathbb{R}$ to $\overline{a} \in \mathbb{R}$ is denoted as

$$[\underline{a}, \overline{a}] = \{x \in \mathbb{R} : \underline{a} \leq x \leq \overline{a}\}, \tag{7}$$

and it equals the empty set if and only if $\underline{a} > \overline{a}$. A singleton (real) interval that contains only $a \in \mathbb{R}$ is denoted as $[a] = [a, a]$. Given $n \in \mathbb{N}_+$, a (real) box $B \subset \mathbb{R}^n$ from $\underline{\mathbf{b}} = (\underline{b}_k)_{k=1}^n \in \mathbb{R}^n$ to $\overline{\mathbf{b}} = (\overline{b}_k)_{k=1}^n \in \mathbb{R}^n$ is denoted as

$$B = [\underline{\mathbf{b}}, \overline{\mathbf{b}}] = [\underline{b}_1, \overline{b}_1] \times [\underline{b}_2, \overline{b}_2] \times \cdots \times [\underline{b}_n, \overline{b}_n], \tag{8}$$

and it equals the empty set if and only if $\underline{b}_i > \overline{b}_i$ for some $1 \leq i \leq n$. Note that the notation $B_i = [\underline{b}_i, \overline{b}_i] \subset \mathbb{R}$ with $1 \leq i \leq n$ is used to refer to the closed intervals that compose the nonempty box B. Also note that the notation $B_{i \to W}$ is used to refer to the box obtained by replacing the $i-$th closed interval that composes the nonempty box B with the nonempty closed interval $W \subset \mathbb{R}$.

Interval arithmetic (e.g., [21]) uses the introduced notation to express computations whose arguments and results are closed intervals. Given two nonempty closed intervals $A = [\underline{a}, \overline{a}] \subset \mathbb{R}$ and $C = [\underline{c}, \overline{c}] \subset \mathbb{R}$, they can be added

$$A + C = [\underline{a} + \underline{c}, \overline{a} + \overline{c}], \tag{9}$$

and they can be multiplied

$$A \cdot C = [\min\{\underline{a}\,\underline{c}, \underline{a}\,\overline{c}, \overline{a}\,\underline{c}, \overline{a}\,\overline{c}\}, \max\{\underline{a}\,\underline{c}, \underline{a}\,\overline{c}, \overline{a}\,\underline{c}, \overline{a}\,\overline{c}\}]. \tag{10}$$

In both cases the result is a nonempty closed interval. Note that the product among intervals can be used to define the $m-$th power of a closed interval $A \subset \mathbb{R}$, where $m \in \mathbb{N}_+$. Also note that the convention $[0]^0 = [1]$ is adopted consistently in this paper. Given $n \in \mathbb{N}_+$, a nonempty box $B \subset \mathbb{R}^n$, and a multi-index $I \in \mathbb{N}^n$, the following notation that uses only products among intervals is normally used

$$B^I = \prod_{j=1}^{n} B_j^{i_j}. \tag{11}$$

Using the introduced notation, which is typical of interval arithmetic, a polynomial function $p : \mathbb{R}^n \mapsto \mathbb{R}$ of $n \in \mathbb{N}_+$ real variables with real coefficients can be extended to work on boxes. If the function is expressed as in (6), then for any box $B \subset \mathbb{R}^n$

$$p(B) = \sum_{I \leq L} [a_I] B^I, \tag{12}$$

where the notation for singleton intervals is used to treat the coefficients of the polynomial function as closed intervals. Observe that it is easy to prove that the result of the evaluation of $p(B)$ is a closed interval that includes the range of p over box B (e.g. [21]).

III. Localization as Optimization

The *localization as optimization* approach was proposed in previous works (e.g., [5], [20]) with the aim of reducing the loss of accuracy that characterizes ordinary algorithms for critical arrangements of the ANs, as briefly recalled in Section I. The localization as optimization approach is based on the possibility of rewriting any localization problem into a related optimization problem, which can then be solved using one of the algorithms documented in the vast literature on optimization. Observe that many of the optimization algorithms available in the literature still have problems related to ill-conditioned matrices. Therefore, specific attention must be paid to the choice of a proper optimization algorithm in order to guarantee that the reformulation of the localization problem in terms of an optimization problem would lead to high accuracy in scenarios where the arrangements of ANs is critical. This is the reason why, among the large number of optimization algorithms, the use of *Particle Swarm Optimization* (*PSO*) [22] was chosen for initial experiments. Experimental results (e.g., [5], [20]) showed that the use of PSO to implement localization as optimization proved to be effective in solving the loss of accuracy caused by critical arrangements of ANs. Unfortunately, PSO has two severe issues that limits its applicability to solve practical localization problems. First, just like all other general optimization algorithms, PSO does not guarantee that global extrema of studied functions are computed in finite time. Second, the performance of the PSO algorithm depends on a set of parameters whose optimal values can be determined only by simulating the behavior of the algorithm in the considered environment. Therefore, alternatives to the use of PSO to implement localization as optimization are under experimentation. Among them, the algorithm briefly recalled in Section IV is a specific instantiation of an algorithmic framework proposed in an upcoming paper. Interested readers can obtain from authors a preprint of the paper where the algorithmic framework is proposed and studied. In the remainder of this section, the approach of localization as optimization is recalled for the tridimensional case for the sake of clarity and to fix the adopted notation.

A set of $m \geq 4$ ANs is positioned at fixed and known locations in the considered indoor environment. The position of the $i-$th AN is denoted as $\mathbf{a}_i \in \mathbb{R}^3$ with $1 \leq i \leq m$. The true position of the TN is denoted as $\mathbf{t} \in \mathbb{R}^3$, and the true distance between the TN and the $i-$th AN is

$$r_i = ||\mathbf{t} - \mathbf{a}_i||. \tag{13}$$

The geometry of the problem suggests that the position of the TN can be found by intersecting m spheres, the $i-$th of which is centered at $\mathbf{a}_i$ and has radius r_i

$$\begin{cases} ||\mathbf{t} - \mathbf{a}_1||^2 = r_1^2 \\ ||\mathbf{t} - \mathbf{a}_2||^2 = r_2^2 \\ \qquad\qquad \vdots \\ ||\mathbf{t} - \mathbf{a}_m||^2 = r_m^2 \end{cases} \tag{14}$$

The localization problem is the problem of finding estimates of the position of the TN provided that the distances between the TN and each AN are estimated using one of the available ranging technologies. The estimated position of the TN is denoted as $\tilde{\mathbf{t}} \in \mathbb{R}^3$, while the estimated distance between the TN and the $i-$th AN is

$$\tilde{r}_i = ||\tilde{\mathbf{t}} - \mathbf{a}_i||. \tag{15}$$

Therefore, the estimated position $\tilde{\mathbf{t}}$ of the TN is expected to satisfy the following system of nonlinear equations, which is obtained by exchanging the true values with their corresponding estimated values in (14)

$$\begin{cases} ||\tilde{\mathbf{t}} - \mathbf{a}_1||^2 = \tilde{r}_1^2 \\ ||\tilde{\mathbf{t}} - \mathbf{a}_2||^2 = \tilde{r}_2^2 \\ \qquad\qquad \vdots \\ ||\tilde{\mathbf{t}} - \mathbf{a}_m||^2 = \tilde{r}_m^2 \end{cases} \tag{16}$$

Note that (16) can be used to try to compute the position estimate $\tilde{\mathbf{t}}$. Unfortunately, even if (14) always has a single solution, (16) may not have a single solution because of the errors introduced in the estimation of distances. In general, (16) may have no solutions, and only rarely it has one solution.

Note that (16) can be rewritten in matrix notation as follows

$$1_m \tilde{\mathbf{t}}^\intercal \tilde{\mathbf{t}} - 2A\tilde{\mathbf{t}} = \tilde{\mathbf{k}}, \tag{17}$$

where $1_m \in \mathbb{R}^m$ is a vector whose components are all equal to 1, $\tilde{\mathbf{k}} \in \mathbb{R}^m$ is a vector whose $i-$th element is $\tilde{k}_i = \tilde{r}_i^2 - ||\mathbf{a}_i||^2$, and A is the following $m \times 3$ anchor matrix

$$A = \begin{pmatrix} \mathbf{a}_1^\intercal \\ \mathbf{a}_2^\intercal \\ \vdots \\ \mathbf{a}_m^\intercal \end{pmatrix}. \tag{18}$$

Many localization algorithms (e.g., [18]) perform algebraic manipulations of matrices whose entries depend on the positions of the TN and of the ANs. The matrices involved in such computations can become ill-conditioned in critical situations, and this represents one of the inherent problems of such algorithms. For example, it is evident from (18) that the anchor matrix is rank deficient if all ANs are placed at the same height because the entries in its last column are all equal. As discussed in detail in Section V, the TSML algorithm is not applicable if all ANs are installed at the same height because it would require to invert a singular matrix. Similarly, if the heights of the ANs are not exactly the same, but they are sufficiently close, the TSML algorithm produces large errors because the matrix to be inverted is strongly ill-conditioned.

If (17) has exactly one solution, such a solution can be computed by using the equations of the system to state a proper minimization problem. Actually, the solution $\tilde{\mathbf{t}}$ of (17), which is the estimated position of the TN, can be found by solving the following minimization problem

$$\tilde{\mathbf{t}} = \underset{\mathbf{x} \in D}{\operatorname{argmin}} f(\mathbf{x}), \tag{19}$$

where $D \subseteq \mathbb{R}^3$ is the region of space where the TN is supposed to be located, and $f : D \mapsto \mathbb{R}$ is the function to be minimized, normally called localization cost function

$$f(\mathbf{x}) = ||1_m \mathbf{x}^\intercal \mathbf{x} - 2A\mathbf{x} - \tilde{\mathbf{k}}||^2. \tag{20}$$

Even if (17) does not have exactly one solution, the reformulation of (17) in terms of a minimization problem can still provide an estimate of the position of the TN. Note that function f is a polynomial function of three variables whose multi-degree is $L = (4, 4, 4)$. Also note that since it is assumed that the indoor environment can be split into disjoint boxes, D can be considered as a box without loss of generality.

IV. The POST$_I$ Algorithm

The *Polynomial Optimization using Subdivision Trees* (*POST*) algorithmic framework is introduced in an upcoming paper as an effective means to support the localization as optimization approach described in previous section. The POST framework is completed with a suitable method to compute lower and upper bounds of polynomial functions to obtain an algorithm to solve the minimization problems derived from localization problems. In this paper, the instantiation of the POST framework that uses interval arithmetic to compute lower and upper bounds of polynomial functions over boxes, normally called POST$_I$ algorithm, is described. The experimental results presented in Section V were obtained using an implementation of the POST$_I$ algorithm integrated within the localization add-on module for JADE recalled in Section I.

The POST$_I$ algorithm uses classic results from the literature on interval arithmetic (e.g., [21]) to compute lower and upper bounds of a given polynomial function over a given box. Then, the POST$_I$ algorithm uses such bounds to drive an ordinary subdivision method to solve the minimization problem at hand. In detail, the POST$_I$ algorithm is based on the possibility of rewriting a localization problem into a related optimization problem that aims at finding a global minimum of the localization cost function f defined in (20). Given that one of the adopted assumptions in this paper is that the considered environment can be split into possibly overlapping boxes, as discussed in Section I, the considered minimization problems can be restricted to boxes. In summary, the POST$_I$ algorithm estimates the position of the TN by finding a global minimum of a polynomial function f of $n = 3$ variables and multi-degree $L = (4, 4, 4)$ over a given box, which corresponds to the considered indoor environment.

Note that the accuracy of the POST$_I$ algorithm does not depend on the positions of the TN or of the ANs, and therefore it can be used when the accuracy of ordinary algorithms, like the TSML algorithm, degrades sensibly. In addition, note that experimental results discussed in Section V show that the POST$_I$ algorithm ensures an accuracy of localization comparable to the accuracy of the TSML algorithm when the arrangement of ANs is not critical. The comparison with the TSML algorithm is particularly relevant because the TSML algorithm is an accepted reference to assess the performance of localization algorithms. Actually, the TSML algorithm is particularly relevant in the context of localization because it is proved [23] that it can attain the Cramér-Rao lower bound [24] for the position estimator.

The literature on nonlinear programming proposes several algorithms to minimize a polynomial function. When the minimum is searched in a box, as in the scenarios considered in this paper, most of the minimization algorithms make use of the properties of the Bernstein polynomial basis (e.g., [25] for a comprehensive review of the subject and a historical retrospective). Actually, well-known properties of the Bernstein polynomial basis can be used to compute lower and upper bounds of a polynomial function over a given box, and such bounds can be used to drive the subdivision of the box using a branch-and-bound approach in search for a global minimum (e.g., [26]). In particular, for each subdivision, the so called Bernstein coefficients are computed on the two resulting sub-boxes, and such coefficients are used to obtain the needed lower and upper bounds of the polynomial function for the two sub-boxes. Unfortunately, such an approach is problematic in terms of computational cost when the polynomial function to minimize is obtained from a localization problem because of the number $n = 3$ of variables and because of multi-degree $L = (4, 4, 4)$ of the polynomial function. Even if effective algorithms for the computation of Bernstein coefficients are available (e.g., [27]–[31]), a total of 125 Bernstein coefficients are needed for each considered sub-box in the worst case, and preliminary tests showed that needed computation cost is too high for the intended applications of indoor localization briefly mentioned in Section I.

Classic results from the literature on interval arithmetic can be used to compute lower and upper bounds for a given polynomial function over a given box instead of using the bounds provided by Bernstein coefficients. The bounds obtained from the application of interval arithmetic are typically less strict

than the bounds obtained using Bernstein coefficients, but the needed computation cost for $n = 3$ variables and multi-degree $L = (4, 4, 4)$ is reduced and it is compatible with the intended applications of indoor localization. In particular, the following proposition can be used to compute lower and upper bounds of a polynomial function over a box. The proposition is shown without proof because it is a classic result of the literature on interval arithmetic (e.g., [21]).

Proposition 1: Given a polynomial function $p : \mathbb{R}^n \mapsto \mathbb{R}$ of $n \in \mathbb{N}_+$ variables with multi-degree $L \in \mathbb{N}^n$

$$p(\mathbf{x}) = \sum_{I \leq L} a_I \mathbf{x}^I, \tag{21}$$

if $B = [\underline{\mathbf{b}}, \overline{\mathbf{b}}] \subset \mathbb{R}^n$ is a box from $\underline{\mathbf{b}} \in \mathbb{R}^n$ to $\overline{\mathbf{b}} \in \mathbb{R}^n$, then

$$\underline{p} \leq \min_{\mathbf{x} \in B} p(\mathbf{x}) \qquad \max_{\mathbf{x} \in B} p(\mathbf{x}) \leq \overline{p}, \tag{22}$$

where $p(B) = [\underline{p}, \overline{p}]$ is computed using interval arithmetic.

V. Experimental results

In order to test the applicability of the POST$_I$ algorithm for indoor localization, an experimental campaign was performed in an indoor environment, as discussed in this section. Results were obtained using a commercial Android smartphone equipped with the necessary hardware and software to communicate with UWB tags and to derive distance estimates from the time of flight of UWB signals. The smartphone used for experiments was a SpoonPhone, produced by BeSpoon (www.bespoon.com), which is, to the best of our knowledge, the only commercial smartphone equipped with a dedicated programming interface to easily estimate the distance between the smartphone and a paired UWB tag. In the considered experiments, four UWB tags were used as active ANs and distance acquisition was performed by the SpoonPhone, which was used as TN. The collected distance estimates were then processed according to the POST$_I$ algorithm to obtain estimates of the position of the TN. In order to compare the performance of the POST$_I$ algorithm with that of a known and appreciated algorithm, the distance estimates passed to the implementation of the POST$_I$ algorithm were also passed to the implementation of the TSML algorithm that is readily available in the localization add-on module for JADE.

The indoor environment used to perform reported experiments is a 4 m wide, 10 m long, and 3 m tall corridor. All obstacles were removed from the corridor during experiments to guarantee that all ANs were in line of sight with the TN, and to reduce errors caused by multipath interference. Four different scenarios were considered, which correspond to different configurations of the ANs. For each scenario, twelve different positions of the TN were considered. Such positions are shown as red dots in Fig. 1, which also shows a schematization of the considered environment. Three different heights for the TN were considered, i.e., 3 m (near the ceiling), 1.5 m, and 0 m (near the floor). The four positions of the TN near the ceiling of the corridor are associated in Fig. 1 with

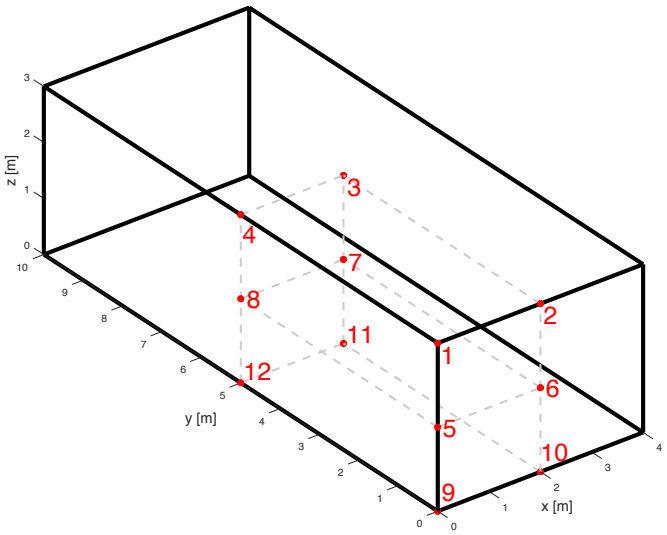

Figure 1. A schematization of the considered indoor environment, i.e., a 4 m wide, 10 m long, and 3 m tall empty corridor, is shown together with the twelve positions of the TN used for experiments (numbered in red).

numbers from 1 to 4, and their coordinates are expressed in meters in the coordinate system shown in the figure as

$$\mathbf{t}_1 = (0, 0, 3) \qquad \mathbf{t}_2 = (2, 0, 3)$$
$$\mathbf{t}_3 = (2, 5, 3) \qquad \mathbf{t}_4 = (0, 5, 3).$$

The four positions of the TN whose heights are all equal to 1.5 m are associated with numbers from 5 to 8, and their coordinates are expressed in meters in the considered coordinate system shown in Fig. 1 as

$$\mathbf{t}_5 = (0, 0, 1.5) \qquad \mathbf{t}_6 = (2, 0, 1.5)$$
$$\mathbf{t}_7 = (2, 5, 1.5) \qquad \mathbf{t}_8 = (0, 5, 1.5).$$

The four positions of the TN near the floor are associated with numbers from 9 to 12, and their coordinates are expressed in meters in the coordinate system shown in the figure as

$$\mathbf{t}_9 = (0, 0, 0) \qquad \mathbf{t}_{10} = (2, 0, 0)$$
$$\mathbf{t}_{11} = (2, 5, 0) \qquad \mathbf{t}_{12} = (0, 5, 0).$$

The localization accuracy of the two considered algorithms is measured in terms of the following position error

$$\mathbf{e} = \mathbf{t} - \tilde{\mathbf{t}}, \tag{23}$$

where $\mathbf{t}$ is the true position of the TN and $\tilde{\mathbf{t}}$ is the estimated position of the TN. Since four different scenarios corresponding to four different arrangements of the ANs were studied, and since experimental results were derived in correspondence of twelve positions of the TN for each scenario, a total of 48 experimental configurations were considered. For each considered experimental configuration, $r = 100$ estimates of the distance between the TN and each AN were acquired, and such estimates were used to derive r position estimates using the POST$_I$ algorithm. Then, r position estimates were also derived by processing the same distance estimates with

Table I
EXPERIMENTAL RESULTS IN SCENARIO 1

Position	e_P [m]	σ_P [m]	e_T [m]	σ_T [m]
$\mathbf{t}_1$	0.352	0.236	N/A	N/A
$\mathbf{t}_2$	0.284	0.103	N/A	N/A
$\mathbf{t}_3$	0.249	0.411	N/A	N/A
$\mathbf{t}_4$	0.251	0.258	N/A	N/A
$\mathbf{t}_5$	0.316	0.093	N/A	N/A
$\mathbf{t}_6$	0.316	0.070	N/A	N/A
$\mathbf{t}_7$	0.730	0.247	N/A	N/A
$\mathbf{t}_8$	0.608	0.185	N/A	N/A
$\mathbf{t}_9$	0.255	0.088	N/A	N/A
$\mathbf{t}_{10}$	0.392	0.259	N/A	N/A
$\mathbf{t}_{11}$	0.312	0.220	N/A	N/A
$\mathbf{t}_{12}$	0.506	0.151	N/A	N/A

Table II
EXPERIMENTAL RESULTS IN SCENARIO 2

Position	e_P [m]	σ_P [m]	e_T [m]	σ_T [m]
$\mathbf{t}_1$	0.400	0.489	0.430	0.198
$\mathbf{t}_2$	0.403	0.231	0.997	2.137
$\mathbf{t}_3$	0.628	0.580	2.190	2.572
$\mathbf{t}_4$	0.305	0.158	0.801	1.135
$\mathbf{t}_5$	0.350	0.174	1.421	2.468
$\mathbf{t}_6$	0.348	0.192	0.740	0.663
$\mathbf{t}_7$	0.793	0.229	0.754	0.818
$\mathbf{t}_8$	0.585	0.197	0.856	0.634
$\mathbf{t}_9$	0.246	0.092	2.387	2.903
$\mathbf{t}_{10}$	0.341	0.188	1.359	1.746
$\mathbf{t}_{11}$	0.435	0.189	1.679	1.575
$\mathbf{t}_{12}$	0.440	0.099	1.804	2.313

the TSML algorithm. Finally, the average and the standard deviation of the Euclidean norm of $\mathbf{e}$ over the r acquisitions obtained using the POST$_I$ algorithm, denoted as e_P and σ_P, were computed together with the corresponding values, denoted as e_T and σ_T, obtained using the TSML algorithm.

A. Scenario 1

In the first scenario, the four ANs are placed at the same height near the ceiling of the considered environment. Using the coordinate system shown in Fig. 1, the positions of the ANs have the following coordinates expressed in meters

$$\mathbf{a}_1 = (0, 0, 3) \qquad \mathbf{a}_2 = (4, 0, 3)$$
$$\mathbf{a}_3 = (0, 10, 3) \qquad \mathbf{a}_4 = (4, 10, 3).$$

As discussed in Section I, the fact that all ANs share the same height is problematic for ordinary algorithms, and the TSML algorithm is actually inapplicable in this scenario. For this reason, only the accuracy of the POST$_I$ algorithm can be evaluated. Table I shows the results related to the accuracy of the POST$_I$ algorithm using the values e_P and σ_P defined above for the $r = 100$ experiments and for each one of the twelve positions of the TN shown. Observe that values of e_P vary from 0.249 m to 0.730 m. Therefore, it can be concluded that the accuracy of localization is compatible with the intended applications discussed in Section I. The values of standard deviations are also compatible with practical applications because they vary from 0.07 m to 0.411 m.

B. Scenario 2

In the second scenario, the height of two of the four ANs is reduced with respect to the previous scenario. Using the coordinate system shown in Fig. 1, the positions of the ANs have the following coordinates expressed in meters

$$\mathbf{a}_1 = (0, 0, 2.5) \qquad \mathbf{a}_2 = (4, 0, 3)$$
$$\mathbf{a}_3 = (0, 10, 2.5) \qquad \mathbf{a}_4 = (4, 10, 3).$$

Table II shows the accuracy of the algorithms in terms of e_P and σ_P (evaluated using the POST$_I$ algorithm), and of e_T and σ_T (evaluated using the TSML algorithm). All values are computed using $r = 100$ repetitions of the experiments for each one of the twelve positions of the TN shown in Fig. 1. Results show that the accuracy of the POST$_I$ algorithm is similar to the accuracy measured in the first scenario. As a matter of fact, the values of e_P vary between 0.246 m and 0.793 m, and the values of σ_P vary between 0.092 m and 0.580 m. The similarity between the results obtained in the two scenarios is not surprising because the position of the ANs did not change significantly.

Table II also shows the results related to the accuracy of the TSML algorithm, expressed in terms of e_T and σ_T. Observe that such results are affected by significant errors in correspondence of some positions of the TN. In detail, the value of the average error e_T is larger than 2 m when the TN was positioned in $\mathbf{t}_3$ and $\mathbf{t}_9$, and it is larger than 1 m when the TN was positioned in $\mathbf{t}_5$, $\mathbf{t}_{10}$, $\mathbf{t}_{11}$, and $\mathbf{t}_{12}$. Moreover, for 8 positions out of 12, the value of σ_T is larger than 1 m and it is often larger than 2 m, which emphasizes the inaccuracy of position estimates obtained with the TSML algorithm for the studied arrangement of ANs. From obtained results it can be concluded that the performance of the POST$_I$ algorithm is good in this scenario, while the TSML algorithm leads to very inaccurate position estimates.

C. Scenario 3

As observed in the previous scenario, the reduction of the height of two ANs by 0.5 m is not sufficient to obtain accurate position estimates with the TSML algorithm. For this reason, in this scenario the height of two ANs is further reduced with respect to the previous scenario and it equals 1.5 m. Hence, using the coordinate system shown in Fig. 1, the positions of the ANs have the following coordinates expressed in meters

$$\mathbf{a}_1 = (0, 0, 1.5) \qquad \mathbf{a}_2 = (4, 0, 3)$$
$$\mathbf{a}_3 = (0, 10, 1.5) \qquad \mathbf{a}_4 = (4, 10, 3).$$

Observe that the new height of the two relocated ANs may not be sufficient to guarantee visibility in practical applications because of the possible presence of obstacles. However, in the considered scenario, the corridor is empty and, therefore, visibility is guaranteed just like in previous scenarios.

Table III shows the accuracy of the algorithms in terms of e_P and σ_P (evaluated using the POST$_I$ algorithm), and of e_T and σ_T (evaluated using the TSML algorithm). All values are

Table III
EXPERIMENTAL RESULTS IN SCENARIO 3

Position	e_P [m]	σ_P [m]	e_T [m]	σ_T [m]
$\mathbf{t}_1$	0.335	0.382	0.314	0.126
$\mathbf{t}_2$	0.331	0.126	0.345	0.193
$\mathbf{t}_3$	0.533	0.388	0.479	0.708
$\mathbf{t}_4$	0.303	0.064	0.573	0.103
$\mathbf{t}_5$	0.404	0.276	0.446	0.745
$\mathbf{t}_6$	0.417	0.290	0.196	0.098
$\mathbf{t}_7$	0.557	0.474	0.896	0.706
$\mathbf{t}_8$	0.339	0.242	0.566	0.234
$\mathbf{t}_9$	0.287	0.070	0.756	0.745
$\mathbf{t}_{10}$	0.419	0.358	0.867	0.923
$\mathbf{t}_{11}$	0.335	0.161	0.650	0.456
$\mathbf{t}_{12}$	0.467	0.090	0.626	0.248

Table IV
EXPERIMENTAL RESULTS IN SCENARIO 4

Position	e_P [m]	σ_P [m]	e^T [m]	σ^T [m]
$\mathbf{t}_1$	0.275	0.068	0.583	0.551
$\mathbf{t}_2$	0.309	0.065	0.298	0.111
$\mathbf{t}_3$	0.326	0.333	0.313	0.336
$\mathbf{t}_4$	0.454	0.144	0.783	0.374
$\mathbf{t}_5$	0.350	0.213	0.426	0.446
$\mathbf{t}_6$	0.271	0.111	0.315	0.103
$\mathbf{t}_7$	0.218	0.328	0.385	0.509
$\mathbf{t}_8$	0.241	0.121	0.600	0.114
$\mathbf{t}_9$	0.308	0.070	0.257	0.104
$\mathbf{t}_10$	0.483	0.420	0.627	0.589
$\mathbf{t}_11$	0.346	0.309	0.878	0.194
$\mathbf{t}_12$	0.409	0.069	0.782	0.168

computed using $r = 100$ repetitions of the experiments for each one of the twelve positions of the TN shown in Fig. 1. Results show that the accuracy of the POST$_I$ algorithm is similar to the accuracy measured in the first and in the second scenarios. Actually, the values of e_P vary between 0.287 m and 0.557 m, and the values of σ_P vary between 0.064 m and 0.474 m. The results obtained in the considered scenarios are not surprising because the accuracy of the POST$_I$ algorithm is expected to be independent from the position of ANs.

Table III also shows the results related to the accuracy of the TSML algorithm, expressed in terms of e_T and σ_T. In this scenario, results obtained with the TSML algorithm are sufficiently accurate for many applications. As a matter of fact, the values of the average error e_T vary between 0.196 m and 0.896 m, and the values of the standard deviation σ_T vary between 0.098 m and 0.923 m. A comparison between the values of the average errors e_P obtained with the POST$_I$ algorithm and the values of the average errors e_T obtained with the TSML algorithm shows that the characteristics of the two algorithms are comparable in this scenario. Analogous considerations can be derived from the values of the standard deviation σ_P obtained with the POST$_I$ algorithm and the value of the standard deviation σ_T obtained with the TSML algorithm, even though the values of σ_P are actually lower than the corresponding values of σ_T for 9 out of the 12 positions of the TN.

D. Scenario 4

In this scenario, the height of the two relocated ANs is further reduced and they are placed near the floor of the corridor while the other two ANs are still at their original positions. Using the coordinate system shown in Fig. 1, the positions of the ANs are expressed in meters as

$$\mathbf{a}_1 = (0, 0, 0) \qquad \mathbf{a}_2 = (4, 0, 3)$$
$$\mathbf{a}_3 = (0, 10, 0) \qquad \mathbf{a}_4 = (4, 10, 3).$$

As in the previous scenario, the new height of the two relocated ANs may not be sufficient to guarantee visibility in practical applications because of the presence of obstacles.

Table IV shows the accuracy of the algorithms in terms of e_P and σ_P (evaluated using the POST$_I$ algorithm), and of e_T and σ_T (evaluated using the TSML algorithm). All values are computed using $r = 100$ repetitions of the experiments for each one of the twelve positions of the TN shown in Fig. 1. Results show that the accuracy of the POST$_I$ algorithm is similar to the accuracy measured in the first, second, and third scenarios. As a matter of fact, the values of e_P vary between 0.218 m and 0.483 m, and values of σ_P vary between 0.065 m and 0.420 m. Such results are in agreement with those obtained in previous scenarios, which is not surprising, since the accuracy of the POST$_I$ algorithm is expected to be independent from the position of ANs.

Table IV also shows the accuracy of the TSML algorithm using the quantities defined previously for the $r = 100$ repetitions of experiments and for the twelve positions of the TN shown in Fig. 1. In this case, the accuracy obtained with the TSML algorithm is similar to the accuracy obtained in the third scenario. Actually, the values of the average error e_T vary between 0.257 m and 0.878 m, and the values of the standard deviation σ_T vary between 0.103 m and 0.589 m, which ensures an accuracy of localization compatible with intended applications discussed in Section I. As for the previous scenarios, the results reported in Table IV show that the performance of the two algorithms are similar in this case.

VI. CONCLUSIONS

The POST$_I$ algorithm is discussed in this paper as a viable opportunity to support the localization as optimization approach (e.g., [5], [20]). The POST$_I$ algorithm is based on consolidated results from the literature on nonlinear programming, and it relies on a subdivision method to search for the position of the TN in the considered indoor environment, which is assumed to be composed of possibly overlapping rectangular cuboids. According to the current implementation of the algorithm, the TN can actively obtain estimates of the distances from ANs using UWB. An alternative implementation of the algorithm that uses WiFi signaling to obtain distance estimates was experimented, but the accurate and robust distance estimates that UWB provides ensures far better localization performance and such an implementation is not discussed in this paper.

One of the most relevant characteristics of the POST$_I$ algorithm is that it provides a level of accuracy that is independent from the position of the ANs in the environment. This is not true for other localization algorithms, like the TSML algorithm, whose performance significantly degrades when problematic arrangements of ANs are used. Experimental results shown in the last part of this paper confirm that the POST$_I$ algorithm leads to accurate position estimates in all considered scenarios, independently from the positions of ANs. Actually, the average localization errors and the corresponding standard deviations are independent from the considered position of the TN and from the adopted arrangement of ANs. At the opposite, significant localization errors are produced by the TSML algorithm when ANs are installed in critical arrangements. In detail, presented experimental results show that the POST$_I$ algorithm outperforms the TSML algorithm in terms of average localization error and corresponding standard deviation when considering scenarios where all the ANs are almost coplanar. Such a result is particularly significant because the TSML algorithm is often used as a reference for the assessment of the performance of localization algorithms because it is well known [23] that it can attain the Cramér-Rao lower bound [24] for the position estimator.

Finally, it is worth noting that the critical arrangements of the ANs that make the use of the TSML impractical are those in which all the ANs are located at (roughly) the same height. Unfortunately, this is one of the most common arrangements in indoor environments, where all the ANs are typically placed at the same height close to the ceilings of rooms to ensure a good line-of-sight coverage. According to such considerations, the use of the POST$_I$ algorithm for robust and accurate localization seems a valid opportunity for envisaged applications of indoor localization, which demands that practical considerations regarding the line-of-sight coverage of environments are seriously taken into consideration.

REFERENCES

[1] Z. Farid, R. Nordin, and M. Ismail, "Recent advances in wireless indoor localization techniques and system," *Journal of Computer Networks and Communications*, vol. 2013, 2013.

[2] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[3] C. A. Patterson, R. R. Muntz, and C. M. Pancake, "Challenges in location-aware computing," *IEEE Pervasive Computing*, vol. 2, no. 2, pp. 80–89, 2003.

[4] S. Monica and F. Bergenti, "Location-aware social gaming with AMUSE," in *Advances in Practical Applications of Scalable Multi-agent Systems, the PAAMS Collection, PAAMS 2016*, ser. Lecture Notes in Computer Science, vol. 9662. Sevilla: Springer, 2016, pp. 36–47.

[5] S. Monica and G. Ferrari, "A swarm-based approach to real-time 3D indoor localization: Experimental performance analysis," *Applied Soft Computing*, vol. 43, pp. 489–497, 2016.

[6] S. Purushotham and C.-C. J. Kuo, "Personalized group recommender systems for location- and event-based social networks," *ACM Transactions on Spatial Algorithms and Systems*, vol. 2, no. 4, pp. 16:1–16:29, 2016.

[7] G. Skoumas, D. Pfoser, A. Kyrillidis, and T. Sellis, "Location estimation using crowdsourced spatial relations," *ACM Transactions on Spatial Algorithms and Systems*, vol. 2, no. 2, pp. 5:1–5:23, 2016.

[8] S.-Y. Teng, W.-S. Ku, and K.-T. Chuang, "Toward mining stop-by behaviors in indoor space," *ACM Transactions on Spatial Algorithms and Systems*, vol. 3, no. 2, pp. 7:1–7:38, 2017.

[9] X. Zhou, T. Chen, D. Guo, X. Teng, and B. Yuan, "From one to crowd: A survey on crowdsourcing-based wireless indoor localization," *Frontiers of Computer Science*, vol. 12, no. 3, pp. 423–450, 2018.

[10] P. Pannuto, B. Kempke, L.-X. Chuo, D. Blaauw, and P. Dutta, "Harmonium: Ultra wideband pulse generation with bandstitched recovery for fast, accurate, and robust indoor localization," *ACM Transactions on Sensor Networks*, vol. 14, no. 2, pp. 11:1–11:29, 2018.

[11] X. Teng, D. Guo, Y. Guo, X. Zhou, and Z. Liu, "CloudNavi: Toward ubiquitous indoor navigation service with 3D point clouds," *ACM Transactions on Sensor Networks*, vol. 15, no. 1, pp. 1:1–1:28, 2019.

[12] S. Gezici and H. V. Poor, "Position estimation via Ultra-Wide-Band signals," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 386–403, 2009.

[13] Z. Sahinoglu, S. Gezici, and I. Güvenc, *Ultra-wideband positioning systems: Theoretical limits, ranging algorithms and protocols*. Cambridge, U.K.: Cambridge University Press, 2008.

[14] S. Monica and F. Bergenti, "Location-aware JADE agents in indoor scenarios," in *Proceedings of the* 16^{th} *Workshop "From Objects to Agents"*, ser. CEUR Workshop Proceedings, vol. 1382. RWTH Aachen, 2015, pp. 103–108.

[15] S. Monica and F. Bergenti, "Indoor localization of JADE agents without a dedicated infrastructure," in *Multiagent System Technologies*, ser. Lecture Notes in Computer Science, vol. 10413. Cham: Springer, 2017.

[16] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi, "JADE – A Java Agent DEvelopment framework," in *Multi-Agent Programming*. Springer, 2005, pp. 125–147.

[17] G. Shen, R. Zetik, and R. S. Thomä, "Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment," in *Proceedings of the Workshop on Positioning, Navigation and Communication (WPNC 2008)*. Hannover: IEEE, 2008, pp. 71–78.

[18] F. Mekelleche and H. Haffaf, "Classification and comparison of range-based localization techniques in wireless sensor networks," *Journal of Communications*, vol. 12, no. 4, 2017.

[19] K. C. Ho, L. Xiaoning, and L.-o. Kovavisaruch, "Source localization using TDOA and FDOA measurements in the presence of receiver location errors: Analysis and solution," *IEEE Transactions on Signal Processing*, vol. 55, no. 2, pp. 684–696, 2007.

[20] S. Monica and G. Ferrari, "Impact of the number of beacons in PSO-based auto-localization in UWB networks," in *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science, vol. 7835. Berlin: Springer, 2013.

[21] W. Tucker, *Validated numerics: A short introduction to rigorous computations*. Princeton: Princeton University Press, 2011.

[22] M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: A review," *Evolutionary Computation*, vol. 25, no. 1, pp. 1–54, 2017.

[23] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, 1994.

[24] J. Shao, *Mathematical Statistics*, ser. Springer Texts in Statistics. New York: Springer-Verlag, 2003.

[25] R. T. Farouki, "The Bernstein polynomial basis: A centennial retrospective," *Computer Aided Geometric Design*, vol. 29, no. 6, pp. 379–419, 2012.

[26] J. Garloff, "The Bernstein algorithm," *Interval Computations*, vol. 2, pp. 154–168, 1993.

[27] R. T. Farouki and V. T. Rajan, "Algorithms for polynomials in Bernstein form," *Computer-Aided Geometric Design*, vol. 5, no. 1, pp. 1–26, 1988.

[28] J. Garloff and A. P. Smith, "Solution of systems of polynomial equations by using Bernstein expansion," in *Symbolic Algebraic Methods and Verification Methods*. Vienna: Springer, 2001, pp. 87–97.

[29] S. Ray and P. S. Nataraj, "An efficient algorithm for range computation of polynomials using the Bernstein form," *Journal of Global Optimization*, vol. 45, pp. 403–426, 2009.

[30] A. P. Smith, "Fast construction of constant bound functions for sparse polynomials," *Journal of Global Optimization*, vol. 43, no. 2, pp. 445–458, 2009.

[31] S. Ray and P. S. Nataraj, "A matrix method for efficient computation of Bernstein coefficients," *Reliable Computing*, vol. 17, pp. 40–71, 2012.