

Optimization-Based Robust Localization of JADE Agents in Indoor Environments

Stefania Monica and Federico Bergenti

Dipartimento di Scienze Matematiche, Fisiche e Informatiche
Università degli Studi di Parma
Parco Area delle Scienze 53/A, 43124 Parma, Italy
{stefania.monica,federico.bergenti}@unipr.it

Abstract. This paper describes and evaluates a localization algorithm that was originally designed to overcome known issues of classic geometric localization algorithms, and that is now implemented in the localization add-on module of JADE. The algorithm is designed to support self-localization of agents running on smart devices in known indoor environments, and its current implementation acquires needed ranging information from ordinary WiFi infrastructures, with no need of dedicated infrastructures. First, the agent estimates the distances of the smart device where it is running from WiFi access points by using received signal strength during ordinary network discovery. Then, the agent uses computed distance estimates to generate estimates of its position by solving an appropriate optimization problem using particle swarm optimization. The robustness of the algorithm is discussed in the last part of the paper by comparing the performance of the algorithm against the performance of a classic algorithm in a representative scenario. Presented experimental results emphasize that the described algorithm is more robust than the classic alternative because it does not suffer from well-known numerical instability problems of geometric localization algorithms.

1 Introduction and Motivation

AAL (Ambient Assisted Living) systems aim at improving the quality of life of the elderly by increasing their autonomy, assisting them in their daily activities, and by enabling them to feel secure, protected and supported (see, e.g., [29]). The space in which AAL systems typically operate is an indoor environment where intelligent agents integrate a wide range of sensors and actuators spread in the environment to deliver AAL services to users. It is therefore far from surprising that the possibility of timely providing agents with the position of users within the environment is one of the key assets needed to implement AAL systems. Indoor localization is so closely related to AAL that the first edition of the *EvALL (Evaluating AAL Systems through Competitive Benchmarking)* [1] international competition was entirely dedicated to benchmark available indoor localization technologies in the scope of AAL. The first EvAAL competition was held in an open space environment composed by a kitchen, a dining room,

a bedroom and a bathroom, and the competition requested to implement the *personal activity management scenario*, as described in the AAL road map [29]. The main challenge that this scenario puts forward to AAL systems regards their abilities to track the position of users, to identify if they are standing or sitting, and to sense if they are performing actions, which can possibly involve the use of an appliance interfaced with the system. By making reference to the personal activity management scenario, where users are localized during their home activities, a resolution finer than 30 cm looks useless: this is about the size of a foot, is less than the body diameter and is less than the extent to which a stretched arm or leg can go. Similarly, a time resolution of half a second seems adequate in the considered scenario because no high-speed activity is expected from users. Therefore, as suggested for the EvALL competition, indoor localization can be considered sufficiently accurate to support AAL if it can provide estimates of the positions of users with an accuracy of 30 cm every 0.5 s.

Unfortunately, even if *smart devices* [26] offer enormous possibilities, they still have inadequate localization capabilities in indoor environments, which are considered essential to deliver effective *location-aware services* in indoor scenarios. If accuracy of few meters is satisfactory, outdoor localization has been already solved by ubiquitous technologies like *GPS* (*Global Positioning System*), which allows estimating the position of a smart device cheaply and effectively [6]. On the contrary, indoor localization is still an open problem and various research efforts are documented in the literature to try to solve it. For example, the use of *UWB* (*Ultra-Wide Band*) technology seems very promising with this respect because it uses short high-frequency pulses that guarantee accuracy and robustness [9, 20]. However, common smart devices are not yet equipped with UWB transceivers, and, most importantly, UWB technology is still experimented mostly to support localization, which means that the use of UWB technology for indoor localization still requires a dedicated infrastructure [16]. The cost of a dedicated infrastructure is normally considered too high for widespread adoption of UWB technology for AAL, even if its performance in localization is better than available alternatives (see, e.g., [12, 13, 15, 17, 18]).

In order to overcome the limitations that the need for a dedicated infrastructure imposes, we looked for valid alternatives to the use of UWB technology and, in this paper, we present empirical evidence that ordinary WiFi networks, which are considered ubiquitous today, can be used to support accurate and robust indoor localization. In detail, this paper discusses recent developments of the localization capabilities offered by *JADE* (*Java Agent and DEvelopment framework*) [2], one the most widely adopted agent platform, which has been successfully used for more than fifteen years in relevant academic and industrial projects. Among the various features of JADE, this paper analyses those of one of the localization algorithms implemented in the localization add-on module for JADE [4, 12, 14]. This module, which is not yet part of the core distribution of JADE, can be loaded into a JADE agent container running on an Android smart device [3] to offer self-localization capabilities in known indoor environments to agents running in the container. Interested agents are fed with estimates of their

positions in the known environment at a constant frequency, which ultimately depends on the technical characteristics of the smart device and on its power management policies. Experimental results discussed in the last part of this paper show that the accuracy of position estimates is coherent with the requirements of AAL systems, and they also show that the discussed algorithm is more robust than classic alternatives based on geometric reasoning.

The localization add-on module for JADE assumes that known WiFi *APs* (*Access Points*) are available in the considered indoor environment at known positions. Under this assumption, an agent running on a smart device where the module is active can request to be informed of its position in the environment. In order to serve such a request, the module estimates the distances between the smart device and single APs using received signal strength from responding APs during ordinary network discovery, even if the smart device is not connected to a WiFi network. Such distance estimates are then properly processed to estimate the position of the smart device, which is immediately made available to the agent. If the agent requested constant updates, the module repeats this process at the highest frequency compatible with the characteristics of the smart device, which is often more than one update per second. The processing of distance estimates is performed by a *localization algorithm* among the ones available in the module, but the module is open to accommodate other algorithms (see, e.g., [8, 24]) provided that they implement the requested interface.

In this paper we concentrate on an algorithm that has been recently proposed [23] to improve the robustness of classic geometric algorithms, and that is currently the best option available to the users of the module. The discussed algorithm was originally designed to use UWB signaling, but it has been recently retargeted to use WiFi signaling [4, 12, 14] for the reasons discussed above. First, the algorithm turns the localization problem at hand into a specific optimization problem. Then, it solves the optimization problem using *PSO* (*Particle Swarm Optimization*), or any other viable alternative. Solutions of the optimization problem are solutions of the localization problem, and they represent estimates of the position of the smart device. Even if this approach has already been discussed in other works (see, e.g., [5, 21, 22]), the major contribution of this paper is to study the robustness of the proposed optimization-based localization algorithm by comparing its performance against that of a classic algorithm. The chosen classic algorithm is the *TSML* (*Two-Stage Maximum-Likelihood*) algorithm [10], which is based on geometric reasoning. Empirical results presented in the last part of the paper show that the optimization-based algorithm has localization accuracy comparable to that of the TSML algorithm in ordinary situations, while it is much better in situations that are critical for localization algorithms based on geometric reasoning.

This paper is organized as follows. Section 2 explains the details of the range acquisition phase. Section 3 shows experimental results obtained in a representative indoor scenario, and it compares the performance of the optimization-based algorithm with that of the TSML algorithm. Finally, Section 4 concludes the paper and outlines possible future developments.

2 Agent-Based Localization

We have recently implemented an add-on module for JADE to provide agents with localization capabilities [4, 12, 14]. A discussion of the architecture of this module, which is briefly summarized in [12], is not needed to detail the expected localization performance, which is the major topic of this paper. Therefore, rather than focusing on the description of the internals of the module, in this paper we concentrate on the description of one of the available localization algorithms together with an experimental evaluation of its robustness.

2.1 Notation and Reference Scenarios

We consider scenarios where M WiFi APs are available and, in the following, their coordinates are denoted as

$$\underline{s}_i = (x_i, y_i, z_i)^T \quad i \in \{1, \dots, M\}. \quad (1)$$

We assume that the indoor environment is static and known to the agent, and, in particular, we assume that the coordinates \underline{s}_i of APs are known to the agent. Note that at least $M = 4$ APs are necessary to perform localization in a three-dimensional environment, unless the considered environment has special geometric characteristics.

The *ranging capabilities* that the smart device is requested to offer in order to support the discussed algorithm concern the possibility of measuring estimates of the distances between the smart device where the agent is running, which is denoted as *TN* (*Target Node*) in the rest of the paper, and the M APs used for localization. Such distance estimates are obtained by analyzing the average received power of WiFi signals traveling between the TN and responding APs during routinary network discovery. According to the Friis transmission equation [9], the average received power $\bar{P}(r)$ can be expressed as a function of the distance r between the transmitter and the receiver. The explicit formulation of the Friis transmission equation is given by

$$\bar{P}(r) = P_0 - 10\beta \log_{10} \frac{r}{r_0} \quad (2)$$

where P_0 is the known power at reference distance r_0 , and β accounts for the characteristics of the transmission. By inverting (2), the value of r as a function of $\bar{P}(r)$ can be expressed as

$$r = r_0 \cdot 10^{-\frac{\bar{P}(r) - P_0}{10\beta}}. \quad (3)$$

Hence, in order to derive an estimate of the distance r between the TN and a generic AP, it is sufficient to measure the average received power of the signal traveling between them and to apply (3). Note that each range estimate can be associated with the corresponding AP and, therefore, with its coordinates. Actually, communications between the TN and an AP during network discovery

includes the *BSSID* (*Basic Service Set IDentification*) of the latter, which can be used to identify the responding AP.

We denote the true position of the TN as $\underline{u} = (x, y, z)^T$, which is supposed to be unknown and it is the quantity to be estimated. Using this notation, the true distance between the TN and the i -th AP can be computed as

$$r_i = \|\underline{u} - \underline{s}_i\| \quad i \in \{1, \dots, M\}. \quad (4)$$

The knowledge of true distances $\{r_i\}_{i=1}^M$, together with the knowledge of the coordinates $\{\underline{s}_i\}_{i=1}^M$ of the APs, would easily determine the position of the TN because the coordinates of the TN could be found by simply intersecting the spheres centered in $\{\underline{s}_i\}_{i=1}^M$ with radii $\{r_i\}_{i=1}^M$. Mathematically, this translates into the solution of the following system of M quadratic equations

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = r_1^2 \\ \dots \\ (x - x_M)^2 + (y - y_M)^2 + (z - z_M)^2 = r_M^2. \end{cases} \quad (5)$$

Unfortunately, since true distances $\{r_i\}_{i=1}^M$ between the TN and each AP are unknown, localization can only be performed using the following system of quadratic equations

$$\begin{cases} (\hat{x} - x_1)^2 + (\hat{y} - y_1)^2 + (\hat{z} - z_1)^2 = \hat{r}_1^2 \\ \dots \\ (\hat{x} - x_M)^2 + (\hat{y} - y_M)^2 + (\hat{z} - z_M)^2 = \hat{r}_M^2 \end{cases} \quad (6)$$

which has been obtained from (5) by replacing the values of true distances $\{r_i\}_{i=1}^M$ with their estimates, denoted as $\{\hat{r}_i\}_{i=1}^M$. Due to errors on range estimates, the M spheres corresponding to the equations in (6) often do not intersect in a single point and, for this reason, a proper localization algorithm needs to be considered to find the position estimate of the TN, denoted as

$$\hat{\underline{u}} = (\hat{x}, \hat{y}, \hat{z})^T. \quad (7)$$

In order to derive a proper localization algorithm, let us first observe that system (6) can be re-written in matrix notation as

$$\underline{\mathbf{1}} \hat{\underline{u}}^T \hat{\underline{u}} + \underline{\underline{A}} \hat{\underline{u}} = \hat{\underline{k}} \quad (8)$$

where $\underline{\mathbf{1}}$ is vector with M elements equal to 1, $\hat{\underline{k}}$ is a vector whose i -th element is $\hat{r}_i^2 - (x_i^2 + y_i^2 + z_i^2)$, and $\underline{\underline{A}}$ is the following $M \times 3$ matrix

$$\underline{\underline{A}} = -2 \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \\ x_M & y_M & z_M \end{pmatrix}. \quad (9)$$

Algorithms to solve (8), based on least square techniques, on Taylor series expansion, or on Maximum-Likelihood methods, can be found in the relevant literature on localization (see, e.g.,[27]).

2.2 The Optimization-Based Localization Algorithm

The literature proposes various algorithms to solve (8) (see, e.g., [27]), and all such algorithms suffer from numerical instability in correspondence of peculiar configurations of APs in space, e.g., if APs are aligned [23]. In order to derive a more robust algorithm, we proposed in [23] to reformulate (8) as an optimization problem, and we described an approach based on *PSO* (*Particle Swarm Optimization*) to solve it, as outlined in the rest of this section.

Observe that (8) can be written as a minimization problem, according to

$$\hat{\underline{u}} = \arg \min_{\underline{u}} F(\underline{u}) \quad (10)$$

where $F(\underline{u})$ represents the *fitness function* defined as

$$F(\underline{u}) = \|\hat{k} - (\underline{1}\underline{u}^T\underline{u} + \underline{A}\underline{u})\|. \quad (11)$$

In order to solve the minimization problem (10), we proposed to use the PSO algorithm, which was introduced in [11]. According to such an algorithm, the set of potential solutions of a minimization problem can be considered as a swarm of particles whose positions and velocities are iteratively updated according to proper rules. Such rules are inspired by biological phenomena like the movements of birds in swarms. In the context of optimization problems, such rules are meant to move all the particles toward the position corresponding to the optimal solution of the considered minimization problem.

In detail, the algorithm that we adopted to solve the minimization problem (10) works as follows. First, the positions of particles are randomly initialized in the *search space*, which, in our context, corresponds to the physical environment where the TN and the APs are situated. The initial positions are denoted as $\underline{x}^{(i)}(0)$, where $i \in \{1, \dots, S\}$ is the index of the generic particle and S is the number of particles. Analogously, the velocity of the i -th particle is initialized with the value $\underline{v}^{(i)}(0)$. After the initialization phase, positions and velocities of all the particles are updated at each iteration $t \in \mathbb{N}$ to simulate interactions among individuals [25]. More precisely, at the t -th iteration, the velocity of the i -th particle whose position is $\underline{x}^{(i)}(t)$ is updated as follows [7, 28]

$$\begin{aligned} \underline{v}^{(i)}(t+1) = & \omega(t)\underline{v}^{(i)}(t) + c_1 R_1(t)(\underline{y}^{(i)}(t) - \underline{x}^{(i)}(t)) \\ & + c_2 R_2(t)(\underline{y}(t) - \underline{x}^{(i)}(t)) \quad i \in \{1, \dots, S\} \end{aligned} \quad (12)$$

where

- $\underline{y}(t)$ is the best position globally reached so far;
- $\underline{y}^{(i)}(t)$ is the best position reached so far;
- $\omega(t)$ is the so called *inertial factor*;
- c_1 is a positive real parameter called *cognition* parameter;
- c_2 is a positive real parameter called *social* parameter; and
- $R_1(t)$ and $R_2(t)$ are independent uniform random variables in $(0, 1)$.

From (12) it can be easily observed that the velocity of a particle at iteration $t + 1$ is obtained as the sum of three addends. The first addend is related to the velocity of the particle at the previous iteration t , which is weighed according to the inertial factor $\omega(t)$. The second addend is meant to move each particle toward the best position it reached so far. Note that such a best position is the one which corresponds to the lowest value of the fitness function and, therefore, $\underline{y}^{(i)}(t)$ can be expressed as

$$\underline{y}^{(i)}(t) = \arg \min_{\underline{z} \in X^{(i)}} F(\underline{z}) \quad X^{(i)} = \{\underline{x}^{(i)}(0), \dots, \underline{x}^{(i)}(t)\}. \quad (13)$$

Finally, the third addend aims at moving each particle towards the *global best position*, namely the position which corresponds to the smallest value of the fitness function among all those reached by any particle in the swarm [25]. Hence, $\underline{y}(t)$ is expressed as

$$\underline{y}(t) = \arg \min_{\underline{z} \in Y} F(\underline{z}) \quad Y = \{\underline{y}^{(1)}(t), \dots, \underline{y}^{(S)}(t)\}. \quad (14)$$

Typically, the inertial factor $\omega(t)$ is chosen as a decreasing function of t , in order to guarantee low dependence of the solution on the initial population, and to reduce the exploration ability of the swarm as the number of iterations increases, making the method more similar to a local search in the last iterations [28].

The velocities computed with (12) are used to update the positions of particles at each iteration according to the following rule

$$\underline{x}^{(i)}(t + 1) = \underline{x}^{(i)}(t) + \underline{v}^{(i)}(t) \quad i \in \{1, \dots, S\}. \quad (15)$$

From (15) it can be observed that the position of the i -th particle at iteration $t + 1$ is simply obtained by adding $\underline{v}^{(i)}(t)$ to its previous position.

The execution of the PSO algorithm is terminated once a stopping criterion is met. Possible stopping criteria for the PSO algorithm are the reach of a maximum number of iterations or the reach of a satisfying value of the fitness function. Once the execution of the algorithm terminates, the solution is identified as the position of the particle in the global best position, which is the particle with the lowest value of the fitness function.

The PSO algorithm is used to solve the localization problem formulated in (10). In order to obtain experimental results shown in the rest of this paper, we set the value of the inertial factor to 0.5; the values of c_1 and c_2 are equal and they are set to 2 (so that the average values of $c_1 R_1(t)$ and of $c_2 R_2(t)$ is 1); the size of the population S is set to 40, and the stopping condition corresponds to the reach of 50 iterations. These values proved to be efficient for localization purposes when WiFi signaling is used instead of UWB signaling, for which different parameters are used [19]. Note that the use of WiFi signaling typically requires a pre-filtering of distance estimates to compensate signal fluctuations, which is not adopted in this paper to assess robustness independently of pre-filtering.

3 Experimental Results

Relevant experimental results obtained using the described optimization-based algorithm are shown in this section. Presented results are used to assess the robustness of the algorithm by comparing its performance against that of the *TSML (Two-Stage Maximum-Likelihood)* algorithm [10]. The TSML algorithm is chosen because it is widely used and it is considered as reference in the field of localization. The performance of the two algorithms is compared in four representative scenarios, in all of which the position of the TN can be expressed in meters as

$$\underline{u} = (0, 1, 1)^T. \quad (16)$$

For each of the considered scenarios, 100 estimates of the position of the TN are derived using the described optimization-based algorithm. Such estimates are denoted as

$$\hat{\underline{u}}_P^{(j)} = (\hat{x}_P^{(j)}, \hat{y}_P^{(j)}, \hat{z}_P^{(j)}) \quad j \in \{1, \dots, 100\}. \quad (17)$$

Similarly, 100 estimates of the position of the TN using the TSML algorithm are also derived, and they are denoted as

$$\hat{\underline{u}}_T^{(j)} = (\hat{x}_T^{(j)}, \hat{y}_T^{(j)}, \hat{z}_T^{(j)}) \quad j \in \{1, \dots, 100\}. \quad (18)$$

In order to compare the performance of the two localization algorithms, we use as performance metrics the distances between the true position \underline{u} of the TN and its estimates. Using the notation introduced in (18), the distance between the true position of the TN and its j -th estimate obtained by applying the TSML algorithm can be denoted as

$$d_T^{(j)} = \|\hat{\underline{u}}_T^{(j)} - \underline{u}\| \quad j \in \{1, \dots, 100\}. \quad (19)$$

The definition in (19) allows computing the average distance error relative to the TSML algorithm as

$$d_{\text{avg}}^{\text{TSML}} = \frac{1}{100} \sum_{j=1}^{100} d_T^{(j)}. \quad (20)$$

Analogous definitions relative to the optimization-based localization algorithm can be adopted. Actually, the accuracy of the optimization-based algorithm is evaluated in terms of

$$d_P^{(j)} = \|\hat{\underline{u}}_P^{(j)} - \underline{u}\| \quad j \in \{1, \dots, 100\} \quad (21)$$

which represent the distance between the true position of the TN and its estimates $\hat{\underline{u}}_P^{(j)}$. From (21), the average distance error obtained from the optimization-based algorithm can be computed as

$$d_{\text{avg}}^{\text{PSO}} = \frac{1}{100} \sum_{j=1}^{100} d_P^{(j)}. \quad (22)$$

In all the considered scenarios we perform localization relying on the distance measurements from $M = 4$ APs, but we remark that the considered localization algorithms can be applied also with a larger number of APs. In the remaining of this section, we consider different scenarios and we compare the performance of the TSML algorithm with that of the optimization-based algorithm. To do so, range estimates from each one of the 4 APs are acquired and used to estimate the position of the TN according the TSML algorithm, whenever possible. Then, the same procedure is applied to derive 100 position estimates according to the optimization-based algorithm. The performance of both algorithms is evaluated in terms of the values of the distance errors (19) and (21) and in terms of their averages (20) and (22).

3.1 First Scenario

We start by considering a configuration of APs whose coordinates, expressed in meters, can be written as follows

$$\begin{aligned} \underline{s}_1 &= (0, 0, 2)^T & \underline{s}_2 &= (1, 1, 2)^T \\ \underline{s}_3 &= (0, 2, 2)^T & \underline{s}_4 &= (-1, 1, 2)^T. \end{aligned} \quad (23)$$

In such a configuration, all APs are located on the same plane since they are placed at the same height. Such a configuration is not strange, since in many indoor scenarios the APs are positioned at the same height, typically close to the ceiling. Notably, the proposed arrangement of APs makes the TSML localization algorithm inapplicable. Actually, the TSML algorithm adopts a geometric approach and the matrices involved in the evaluation of the position of the TN become singular if all the APs used for localization are situated on the same plane. Hence, it is not possible to use the TSML to obtain position estimates of the TN with the configuration of AP in (23). At the opposite, the applicability of the optimization-based algorithm is not influenced by the geometric arrangement of APs. Hence, the optimization-based algorithm can be successfully applied with the configuration of APs in (23), leading to accurate position estimates, as shown in Fig. 1. Actually, Fig. 1 shows the values of distance errors $d_P^{(j)}$ (red dots) and the value of the average distance error $d_{\text{avg}}^{\text{PSO}}$ (yellow line), which is approximately equal to 22 cm. From Fig. 1 it can be concluded that the optimization-based algorithm guarantees accurate position estimates, even in this scenario where algorithms based on geometric reasoning, e.g., the classic and widely adopted TSML algorithm, cannot be applied.

We remark that, especially when an existing WiFi infrastructure is used to support localization, the configuration in which all APs are located at the same height is typical, and often it is preferred over other configurations. In the rest of this section, in order to make the TSML algorithm applicable, we change the height of one of the APs, namely the fourth, and we compare the performance of the two localization algorithms in various scenarios.

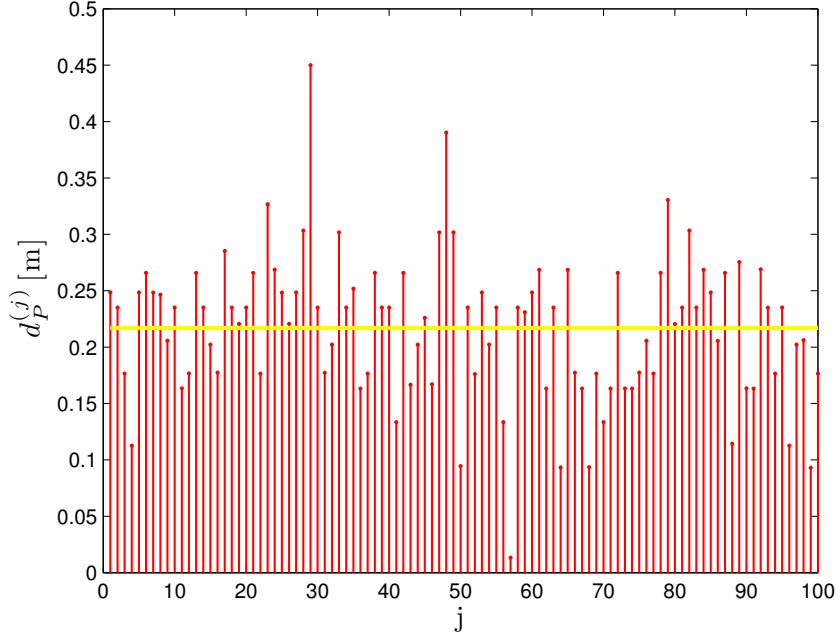


Fig. 1. The values of distance errors obtained with the configuration of APs in (23) are shown when applying the optimization-based algorithm. The value of the average distance error (yellow line) is also shown with a line across the diagram.

3.2 Second Scenario

In this second scenario, we maintain the positions of the APs whose coordinates are denoted as \underline{s}_1 , \underline{s}_2 , and \underline{s}_3 in (23) and we only move the fourth AP. The height of the fourth AP is reduced by 1 cm, so that the coordinates of APs, expressed in meters, can be written as

$$\begin{aligned} \underline{s}_1 &= (0, 0, 2)^T & \underline{s}_2 &= (1, 1, 2)^T \\ \underline{s}_3 &= (0, 2, 2)^T & \underline{s}_4 &= (-1, 1, 1.99)^T. \end{aligned} \quad (24)$$

Observe that the configuration of the APs is very similar to that considered in the first scenario, the only difference being the height of the fourth AP. The small decrease of one of the APs may seem insignificant, but it has a strong impact on the applicability of the TSML algorithm. Obviously, a decrease of 1 cm may fall into the tolerance of the measurement of the height of APs, and it should be considered frequent even when APs are supposed to be at the same height. As a matter of fact, a change of only 1 cm in the height of an AP is sufficient to guarantee that the matrices involved in the TSML algorithm are no longer singular. Hence, in this case, the TSML algorithm can be used to derive

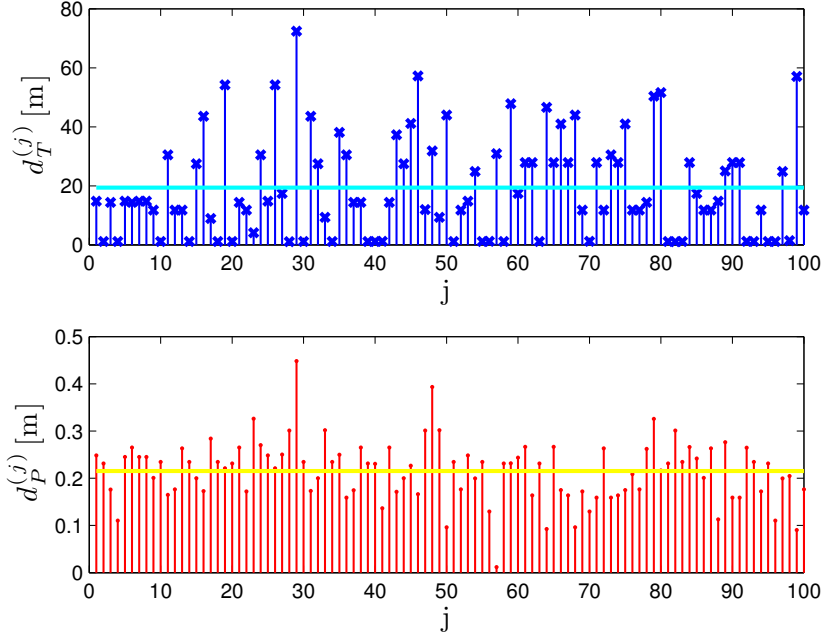


Fig. 2. The values of distance errors obtained with the configuration of APs in (24) are shown when applying: the TSML algorithm (upper diagram); the optimization-based algorithm (lower diagram). The values of average distance errors $d_{\text{avg}}^{\text{TSML}}$ (cyan line) and $d_{\text{avg}}^{\text{PSO}}$ (yellow line) are also shown.

estimates of the position of the TN. The upper diagram in Fig. 2 shows the values of distance errors $d_T^{(j)}$ (blue crosses) relative to the estimates of the position of the TN obtained according to the TSML algorithm. The same figure also shows the value of the average distance error $d_{\text{avg}}^{\text{TSML}}$ (cyan line), which is approximately equal to 20 m. It can be concluded that the position estimates obtained from the application of the TSML algorithm when APs are nearly coplanar are often far inaccurate and, hence, such estimates cannot be used in practice. This is caused by the fact that the matrices used in the TSML algorithm, despite being not singular, are ill-conditioned because APs are nearly placed on the same plane. Hence, it can be concluded that the position estimates obtained according to the widely known TSML algorithm are not reliable in the considered scenario, due to the topology of the APs.

On the contrary, the position estimates obtained from the optimization-based algorithm in the same scenario are sufficiently accurate, as shown in the lower diagram of Fig. 2, where the values of distance errors $d_P^{(j)}$ (red dots) are shown. The same figure also shows the value of the average distance error $d_{\text{avg}}^{\text{PSO}}$ (yellow line), which is approximately 22 cm, as in the first scenario. The values of $d_P^{(j)}$

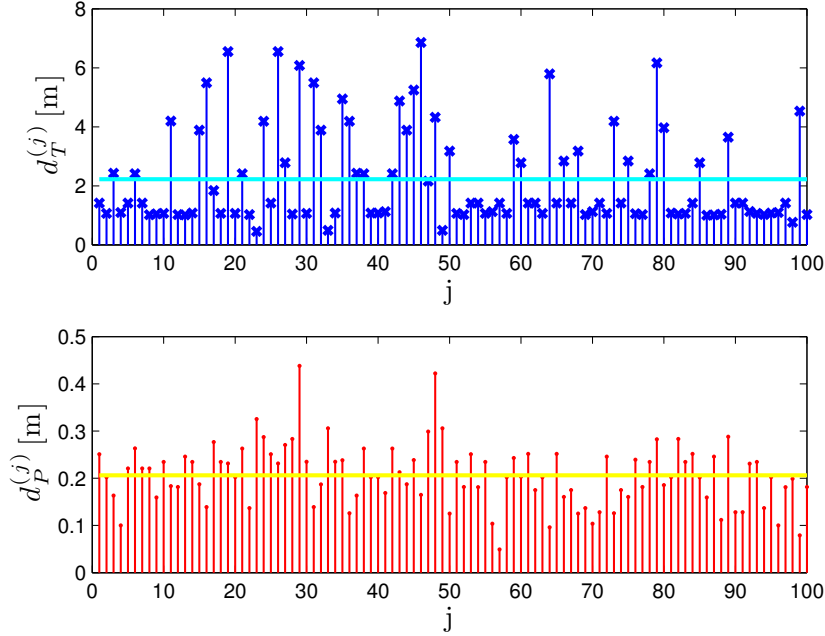


Fig. 3. The values of distance errors obtained with the configuration of APs in (24) are shown when applying: the TSML algorithm (upper diagram); the optimization-based algorithm (lower diagram). The values of average distance errors $d_{\text{avg}}^{\text{TSML}}$ (cyan line) and $d_{\text{avg}}^{\text{PSO}}$ (yellow line) are also shown.

make the optimization-based position estimates sufficiently accurate for many applications. From the results shown in Fig. 2, it can be observed that the values $d_P^{(j)}$ are far smaller than the corresponding values of $d_T^{(j)}$ and, hence, results shown in Fig. 2 confirm that the optimization-based algorithm is more robust than the TSML algorithm.

3.3 Third Scenario

Let us now change again the position of the fourth AP, keeping the remaining APs in the same positions as in previous scenarios. We consider the following coordinates for the fourth AP

$$\underline{s}_4 = (-1, 1, 1.9)^T. \quad (25)$$

According to (25), the height of the fourth AP is reduced by 10 cm with respect to the configuration in the first scenario and by 9 cm with respect to the configuration in the second scenario. Results obtained with this configuration of APs are shown in Fig. 3, where the upper diagram refers to the TSML algorithm

and the lower diagram refers to the optimization-based algorithm. The upper diagram shows the values of $d_T^{(j)}$ (blue crosses) and the value of the average distance error $d_{\text{avg}}^{\text{TSML}}$ (cyan line). A comparison between Fig. 2 and Fig. 3 shows that the values of $d_T^{(j)}$ obtained in the third scenario are smaller by one order of magnitude than those obtained in the second scenario. However, the position estimates obtained from the TSML algorithm still have to be considered as inadequate, since the average distance error is $d_{\text{avg}}^{\text{TSML}} = 2.23$ m and in some cases the values of $d_T^{(j)}$ are larger than 6 m. Once again, this is due to the topology of the APs, according to which the localization problem formulated by TSML algorithm remains ill-conditioned.

The optimization-based algorithm, instead, confirms its reliability, as shown from results in the lower diagram of Fig. 3, where the values of distance errors $d_P^{(j)}$ are shown (red dots). The same figure also shows the value of $d_{\text{avg}}^{\text{PSO}}$ (yellow line) which is approximately equal to that obtained in first and second scenarios, namely 22 cm.

3.4 Fourth Scenario

In this scenario, the fourth AP is positioned on the floor and its coordinates can be expressed as

$$\underline{s}_4 = (-1, 1, 0)^T. \quad (26)$$

Observe that the configuration of APs considered in this scenario makes the matrices involved in the TSML algorithm well-conditioned, as the APs are not on the same plane nor they are nearly coplanar. As a consequence, the obtained position estimates are accurate, as shown in the upper diagram of Fig. 4, where the values of distance errors $d_T^{(j)}$ are shown (blue crosses), and the average value of distance errors (cyan line) is shown to be $d_{\text{avg}}^{\text{TSML}} \simeq 21$ cm. Note that, according to the choice of \underline{s}_4 in (26), the distance between the TN and the fourth AP in this scenario is the same as in the first scenario, where the TSML algorithm could not be applied, which emphasizes that the applicability of the TSML algorithm only depends on the relative positions of the APs and not on the distances between the TN and the APs.

The lower diagram of Fig. 4 shows the values of distance errors $d_P^{(j)}$ obtained according to the optimization-based algorithm (red dots) and the value of $d_{\text{avg}}^{\text{PSO}}$ (yellow line), which is approximately equal to 19 cm. Such a value emphasizes that the accuracy of localization that the optimization-based algorithm ensures does not depend significantly on the position of APs. In the first scenario, where all APs were coplanar, the value of $d_{\text{avg}}^{\text{PSO}}$ was approximately equal to 22 cm, and it remained almost unchanged across all scenarios. Finally, note that from results in Fig. 4, it can be concluded that in this case the TSML algorithm and the optimization-based algorithm perform similarly in terms of the distances between the true position of the TN and its estimates.

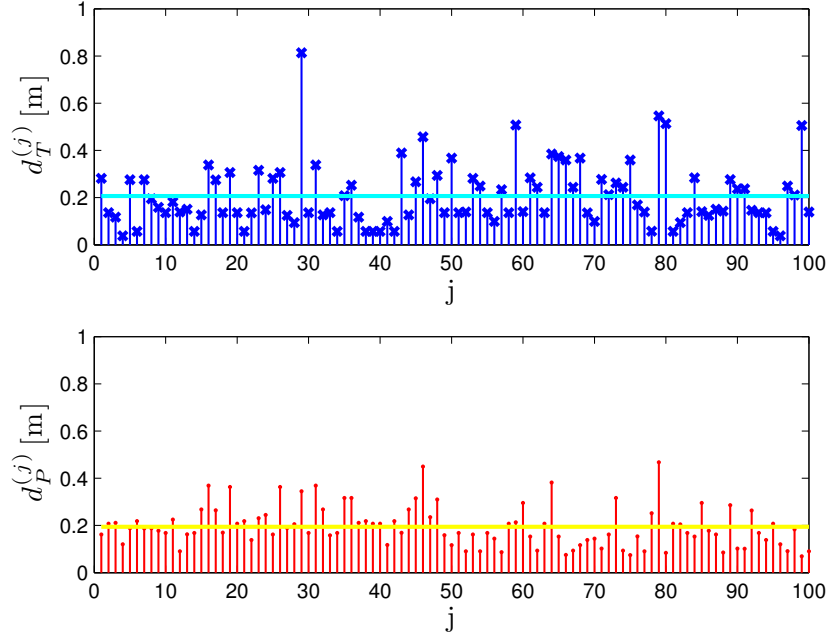


Fig. 4. The values of distance errors obtained with the configuration of APs in (24) are shown when applying: the TSML algorithm (upper diagram); the optimization-based algorithm (lower diagram). The values of average distance errors $d_{\text{avg}}^{\text{TSML}}$ (cyan line) and $d_{\text{avg}}^{\text{PSO}}$ (yellow line) are also shown.

4 Conclusions

This paper discussed an approach to self-localization of agents in known indoor environments that exhibits improved robustness over classic alternatives based on geometric reasoning. First, the agent acquires estimates of the distances from a set of known WiFi APs located in known positions by using the received signal strength during ordinary network discovery. Estimated distances are used to construct an optimization problem whose solution is an estimate of the position of the smart device in the indoor environment. Estimated position is immediately fed to interested agents running in the JADE container on the smart device. Note that the proposed technique is immediately applicable to off-the-shelf Android devices, and that no localization-specific infrastructure is needed.

The major contribution of this paper is to assess the robustness of the discussed algorithm by comparing its accuracy against that of the TSML algorithm, which is a classic alternative. From the results shown in Section 3, it can be concluded that the performance of the TSML algorithm strongly depends on the topology of APs. Actually, if the APs are coplanar or almost coplanar, then

the TSML algorithm is inapplicable or it leads to severely inaccurate position estimates, while the optimization-based algorithm is reliable also in these cases. Moreover, in cases where also the TSML algorithm performs well, the performance of the two algorithms is similar. Note that while other types of wireless technologies, such as UWB, ensure more accurate localization, WiFi has the advantage of being ubiquitously available in all realistic indoor scenarios, and that all smart devices are equipped with a dedicated transceiver.

Experimental results presented in Section 3 also show that the discussed algorithm can support effective implementation of AAL systems because it offers sufficient accuracy and speed, which are set to 30 cm and 0.5 s for scenarios like the personal activity management. Future work on this topic involves further investigation on the performance of the discussed algorithm in different indoor environments and with different configurations of APs.

References

1. Barsocchi, P., Chessa, S., Furfari, F., Potortì, F.: Evaluating ambient assisted living solutions: The localization competition. *IEEE Pervasive Computing* 12(4), 72–79 (2013)
2. Bellifemine, F., Caire, G., Greenwood, D.: *Developing multi-agent systems with JADE*. Wiley Series in Agent Technology, John Wiley & Sons (2007)
3. Bergenti, F., Caire, G., Gotta, D.: Agents on the move: JADE for Android devices. In: Proc. 15th Workshop “Dagli Oggetti agli Agenti” (*WOA 2014*). CEUR Workshop Proceedings, vol. 1260. RWTH Aachen University (2014)
4. Bergenti, F., Monica, S.: Location-Aware Social Gaming with AMUSE. In: *Advances in Practical Applications of Scalable Multi-agent Systems, the PAAMS Collection*. LNCS, vol. 9662, pp. 36–47. Springer (2016)
5. Bergenti, F., Monica, S.: An optimization-based algorithm for indoor localization of JADE agents. In: Proc. 18th Workshop “Dagli Oggetti agli Agenti” (*WOA 2017*). CEUR Workshop Proceedings, vol. 1867. RWTH Aachen University (2017)
6. Bulusu, N., Heidemann, J., Estrin, D.: GPS-less low cost outdoor localization for very small devices. *IEEE Personal Communications* 7(5), 28–34 (2000)
7. Eberhart, R., Kennedy, J.: A new optimizer using particles swarm theory. In: Proc. 6th Int. Symp. Micro Machine and Human Science (*MHS 1995*). pp. 39–43 (1995)
8. Farid, Z., Nordin, R., Ismail, M.: Recent advances in wireless indoor localization techniques and system. *Journal of Computer Networks and Communications* (2013)
9. Gezici, S., Poor, H.V.: Position estimation via Ultra-Wide-Band signals. *Proceedings of the IEEE* 97(2), 386–403 (2009)
10. Ho, K.C., Lu, X., Kovavisaruch, L.: Source localization using TDOA and FDOA measurements in the presence of receiver location errors: Analysis and solution. *IEEE Transactions on Signal Processing* 55(2), 684–696 (2007)
11. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE Int. Conf. Neural Networks (*ICNN 1995*). pp. 1942–1948 (1995)
12. Monica, S., Bergenti, F.: Location-aware JADE agents in indoor scenarios. In: Proc. 16th Workshop “Dagli Oggetti agli Agenti” (*WOA 2015*). CEUR Workshop Proceedings, vol. 1382, pp. 103–108. RWTH Aachen University (2015)
13. Monica, S., Bergenti, F.: A comparison of accurate indoor localization of static targets via WiFi and UWB ranging. In: *Trends in Practical Applications of Scalable*

- Multi-Agent Systems, the PAAMS Collection. Advances in Intelligent Systems and Computing, vol. 473, pp. 111–123. Springer (2016)
14. Monica, S., Bergenti, F.: Experimental evaluation of agent-based localization of smart appliances. In: Proc. 14th European Conference on Multi-Agent Systems (*EUMAS 2016*). LNAI, vol. 10207, pp. 293–304. Springer (2016)
 15. Monica, S., Bergenti, F.: An experimental evaluation of agent-based indoor localization. In: Proc. Computing Conference 2017. pp. 638–646. IEEE Press (2017)
 16. Monica, S., Bergenti, F.: Indoor localization of JADE agents without a dedicated infrastructure. In: Proc. 15th German Conference on Multiagent System Technologies (*MATES 2017*). LNAI, vol. 10413, pp. 256–271. Springer (2017)
 17. Monica, S., Ferrari, G.: Impact of the number of beacons in PSO-based auto-localization in UWB networks. In: Proc. 15th European Conf. Applications of Evolutionary Computation (*EvoApplications 2013*). LNCS, vol. 7835, pp. 42–51. Springer (2013)
 18. Monica, S., Ferrari, G.: Optimized anchors placement: An analytical approach in UWB-based TDOA localization. In: Proc. 9th Int. Wireless Communications & Mobile Computing Conference (*IWCMC 2013*). pp. 982–987. IEEE Press (2013)
 19. Monica, S., Ferrari, G.: Particle swarm optimization for auto-localization of nodes in wireless sensor networks. In: Proc. 11th Int. Conf. Adaptive and Natural Computing Algorithms (*ICANNGA 2013*). LNCS, vol. 7824, pp. 456–465. Springer (2013)
 20. Monica, S., Ferrari, G.: Accurate indoor localization with UWB wireless sensor networks. In: Proc. 23rd IEEE Int. Conf. Enabling Technologies: Infrastructure for Collaborative Enterprises (*WETICE 2014*). pp. 287–289. IEEE Press (2014)
 21. Monica, S., Ferrari, G.: Swarm intelligent approaches to auto-localization of nodes in static UWB networks. *Applied Soft Computing* 25, 426–434 (2014)
 22. Monica, S., Ferrari, G.: A swarm intelligent approach to 3D distance-based indoor localization. In: Proc. 17th European Conf. Applications of Evolutionary Computation (*EvoApplications 2015*). LNCS, vol. 9028, pp. 91–102. Springer (2015)
 23. Monica, S., Ferrari, G.: A swarm-based approach to real-time 3D indoor localization: Experimental performance analysis. *Applied Soft Computing* 43, 489–497 (2016)
 24. Pavani, T., Costa, G., Mazzotti, M., Conti, A., Dardari, D.: Experimental results on indoor localization techniques through wireless sensors network. In: Proc. IEEE Vehicular Technology Conference. pp. 663–667. IEEE Press (2006)
 25. Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization. *Swarm Intelligence Journal* 1(1), 33–57 (2007)
 26. Poslad, S.: *Ubiquitous Computing: Smart Devices, Environments and Interactions*. John Wiley & Sons (2009)
 27. Shen, G., Zetik, R., Thomä, R.S.: Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment. In: Proc. 5th Workshop Positioning, Navigation and Communication (*WPNC 2008*). pp. 71–78. IEEE Press (2008)
 28. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proc. IEEE Int. Conf. Evolutionary Computation (*ICEC 1999*). pp. 69–73. IEEE Press (1999)
 29. Van Den Broek, G., Cavallo, F., Wehrmann, C. (eds.): *AALIANCE Ambient Assisted Living Roadmap*. IOS Press (2010)