



**HAL**  
open science

## Anchor Selection Algorithm for Mobile Indoor Positioning using WSN with UWB Radio

Antoine Courtay, Mickaël Le Gentil, Olivier Berder, Pascal Scalart, Sébastien  
Fontaine, Arnaud Carer

► **To cite this version:**

Antoine Courtay, Mickaël Le Gentil, Olivier Berder, Pascal Scalart, Sébastien Fontaine, et al.. Anchor Selection Algorithm for Mobile Indoor Positioning using WSN with UWB Radio. 2019 IEEE Sensors Applications Symposium (SAS), Mar 2019, Sophia Antipolis, France. 10.1109/SAS.2019.8706113 . hal-02302424

**HAL Id: hal-02302424**

**<https://hal.science/hal-02302424>**

Submitted on 1 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Anchor Selection Algorithm for Mobile Indoor Positioning using WSN with UWB Radio

Antoine Courta, Mickaël Le Gentil, Olivier Berder, Pascal Scalart, Sébastien Fontaine, Arnaud Carer  
Univ. Rennes - IRISA  
Email: firstname.lastname@irisa.fr

**Abstract**—Positioning a person or an object has become essential in many applications. It already exists solutions for outdoor positioning such as satellite based techniques (i.e. GPS) but indoor positioning still remains a great challenge for applications like sports monitoring, contextual visits of museum, Building Information Modeling (BIM) or automated drone missions. Classical approaches using radio communication such as *WiFi*, *Bluetooth*, *ZigBee* only give an accuracy of approximately 2.5 meters when the mobile is static, of course worse when moving. Recently some new radio communication chipsets have emerged based on Ultra Wide Band (UWB) communications. UWB allows accurate Time Of Flight (TOF) measurements, and thus distances estimations between nodes equipped with. A positioning algorithm named Best Anchor Selection for Trilateration (BAST) based on position prediction and noise estimation is proposed. Then a wearable, light and low power Wireless Sensor Network (WSN) prototype (named *Zyggie*) including an UWB chipset has been developed for algorithms comparison. Finally, an experimental testbed using real cases experiments shows that BAST can give from 1.26 up to 4.17 times better accuracy than low complexity state of the art algorithms when the mobile/person is in movement (e.g. tennis player).

## I. INTRODUCTION

Knowing the position of a device in its environment becomes an important issue for applications providing services based on the position. Many solutions have been proposed going from satellite based techniques (i.e. GPS), video tracking or radio-electrical waves combining. However, these techniques reach their limit when very high accuracy is required, especially in indoor environment. There are several approaches to estimate the location of a mobile in an indoor environment. Some of them use optical signals such as infrared, laser or video tracking approaches (like the *Vicon* system [1]) and many others use radio signal. As applications we target need low cost, portable and low power systems which can be deployed in minutes, radio signal positioning system is the best candidate.

Last years many radio standards have been explored such as *WiFi* [2], [3], *Bluetooth* [4], *ZigBee* [5], [6] concerning indoor positioning. They are all based on the Received Signal Strength Indicator (RSSI) of a signal between two devices. Performances are often disappointing due to the lack of accuracy (approximately 2.5 m) and gets even worse if the mobile is in Non Line Of Sight (NLOS), in movement or if there is multipath (walls, objects). A new radio approach known as Ultra Wide Band (UWB) has recently emerged which has the ability to do accurate Time Of Flight (TOF)

measurements.

Since 2014, *Microsoft* has launched an *Indoor Localization Competition* co-located with the *IPSN* Conference. When looking at the results of the competition [7] we can notice that more than 50% of the 3D competitors use UWB technology including the two first competitors. The best UWB competitor in 2018 has an average error of 27 cm over many fixed points measurements following a trajectory in a building. However, most of the presented prototypes are not enough miniaturized yet to be wearable or use different platforms for the mobile and anchors. Moreover, some of them use wired together anchors for network synchronization.

Unlike academic and industrial solutions mentioned above, our wireless sensor node *Zyggie* [8] is ultra compact and very light which really makes it a good candidate for all applications where weight and size are critical. Both anchors and mobile have the same architecture and the network synchronization is done wirelessly via a 2.4 GHz radio frontend. Respecting low complexity and low memory footprint constraints imposed by such wireless nodes, the main contribution of this paper is the development of a positioning algorithm named Best Anchor Selection for Trilateration (BAST) that outperforms low complexity state of the art algorithms under low power and real time computation constraints. BAST also allows to display output trajectory on portable devices (smartphone, tablet) in real time. An Android application has been developed for this purpose. A last contribution is that all experimental results were obtained using simulations based on real cases experiments. A serpentine walking trajectory and a real tennis playing game were acquired several times to evaluate algorithms performance.

The rest of the paper is organized as follows. Section II summarizes the state of the art algorithms applied to a positioning process knowing distances with reference points. Section III introduces our new algorithm (BAST) for indoor positioning of a mobile. In Section IV, the developed prototype node, the experimental setup with two scenarios (walking in a room and playing tennis) are explained. In Section V, the results of all the positioning algorithms are discussed. Finally, conclusions are drawn and future works are commented in Section VI.

## II. RELATED WORK ON POSITIONING ALGORITHMS

A positioning algorithm aims at providing 2D/3D coordinates by combining some distances of a (mobile) node with respect to some fixed nodes (which positions are known

in advance) named here anchors. In the rest of the paper, resolution will be presented in 2D. For a 3D resolution, just add in the system (1) the  $z$  contribution. It is possible by knowing the respective distances of a mobile node to 3 anchors to solve the following system. The more anchors there are the more equations in the system there will be. Here is the system with 3 anchors

$$\begin{cases} d_{A1}^2 = (x_M - x_{A1})^2 + (y_M - y_{A1})^2 + n_{A1} \\ d_{A2}^2 = (x_M - x_{A2})^2 + (y_M - y_{A2})^2 + n_{A2} \\ d_{A3}^2 = (x_M - x_{A3})^2 + (y_M - y_{A3})^2 + n_{A3} \end{cases}, \quad (1)$$

where the  $d_{Ai}$  are the distances toward the anchor  $i$ , the  $x_{Ai}$  and  $y_{Ai}$  are the coordinates of the anchor  $i$ , the  $n_{Ai}$  are the noises on the measured distances and  $x_M, y_M$  the mobile' coordinates. All the following algorithms have the same starting point (i.e. the distances) to compute the mobile coordinates.

#### A. Weighted Centroid Localization

The Weighted Centroid Localization (WCL) algorithm explained in [9] consists in finding the solution point  $\mathbf{m}$  in

$$\mathbf{m} = \begin{pmatrix} x_M \\ y_M \end{pmatrix} = \sum_{i=1}^N w_{Ai} \cdot \mathbf{a}_{Ai} = \sum_{i=1}^N \frac{\frac{1}{d_{Ai}} \cdot \begin{pmatrix} x_{Ai} \\ y_{Ai} \end{pmatrix}}{\sum_{j=1}^N \frac{1}{d_{Aj}}}, \quad (2)$$

where  $\mathbf{m}$  is the researched point (mobile's coordinates),  $\mathbf{a}_{Ai}$  the coordinates vectors of the  $N$  anchors and  $w_{Ai}$  the weighting coefficients of each anchor. In our case, these coefficients are the inverse of the considered distance  $d_{Ai}$  divided by the sum of the inverses of  $d_{Ai}$ . Using WCL, it is also possible to affect a different weight to each anchor to make its influence higher in the solution.

#### B. Gauss-Newton

The Gauss-Newton method [10] is an iterative algorithm which aims at minimizing the positioning error. Mobile's coordinates  $\mathbf{m}$  are computed iteratively by solving

$$\mathbf{m}_l = \mathbf{m}_{l-1} - \mathbf{H}_{l-1}^{-1} \cdot \mathbf{g}_{l-1} = \begin{pmatrix} x_M \\ y_M \end{pmatrix}, \quad (3)$$

where  $l$  is the iteration step,  $\mathbf{H}$  the Hessian matrix and  $\mathbf{g}$  the gradient vector. Gauss-Newton algorithm has the advantage to quickly converge. On the other side, the Hessian matrix uses a  $3/2$  power on the distances which leads to more computation.

#### C. Trilateration

Trilateration algorithm consists here in finding a solution to the previously defined system (1). By doing some transformations, it is possible to write the system to solve such as  $\mathbf{u} = \mathbf{V} \cdot \mathbf{m} + \mathbf{e}$  with

$$\begin{aligned} \mathbf{u} &= \begin{pmatrix} (d_{A1}^2 - d_{A2}^2) - (x_{A1}^2 - x_{A2}^2) - (y_{A1}^2 - y_{A2}^2) \\ (d_{A1}^2 - d_{A3}^2) - (x_{A1}^2 - x_{A3}^2) - (y_{A1}^2 - y_{A3}^2) \end{pmatrix}, \\ \mathbf{V} &= \begin{pmatrix} 2 \cdot (x_{A2} - x_{A1}) & 2 \cdot (y_{A2} - y_{A1}) \\ 2 \cdot (x_{A3} - x_{A1}) & 2 \cdot (y_{A3} - y_{A1}) \end{pmatrix} \text{ and} \\ \mathbf{e} &= \begin{pmatrix} n_{A1} - n_{A2} \\ n_{A1} - n_{A3} \end{pmatrix}. \end{aligned} \quad (4)$$

For this example Anchor number 1 has been chosen for reference and therefore appears twice in the equations. By posing

$$\mathbf{v}_1 = \begin{pmatrix} 2 \cdot (x_{A2} - x_{A1}) \\ 2 \cdot (x_{A3} - x_{A1}) \end{pmatrix} \text{ and } \mathbf{v}_2 = \begin{pmatrix} 2 \cdot (y_{A2} - y_{A1}) \\ 2 \cdot (y_{A3} - y_{A1}) \end{pmatrix}, \quad (5)$$

the Cramer's rules give a solution that minimize the Mean Square Error (MSE) to obtain the mobile coordinates below

$$x_M = \frac{\det(\mathbf{u} \ \mathbf{v}_2)}{\det(\mathbf{V})} \text{ and } y_M = \frac{\det(\mathbf{v}_1 \ \mathbf{u})}{\det(\mathbf{V})}. \quad (6)$$

Anchors must not be aligned to obtain a solution for  $x_M$  and  $y_M$  due to a null denominator. There exists some other mathematical ways to solve the system such as Cramer-Rao Bound (CRB) [11] and Least Squares (LS) [12]. As we target low complexity computation, only Cramer method was implemented.

#### D. Multilateration

Multilateration algorithm [13] uses the previously seen developments of trilateration and applies it to all of the combinations of 3 non-aligned anchors. By this way, for a configuration with  $N$  anchors, we obtain  $C_3^N$  trilateration combinations with aligned combinations subtracted. Finally, we obtain the estimated point by doing the mean (centroid) of these points. The main drawback of this method is the fact that the more anchors your system has, the more calculations need to be done in return of a theoretically better accuracy.

There exist some other trilateration/multilateration variant based algorithms such as [14]–[17]. Even if proposed results seem to be of interest, these algorithms are either too complex in hardware or software implementation to match our constraints. Indeed we target low cost and energy efficient real time computation/displaying for mobile indoor positioning applications (in-building or sports). Authors from [14] propose to use fuzzy logic to compensate noise before multilateration processing. Their approach needs rather complex processing functions (training phases, recursive filters and coefficient updates). Other techniques are based on range free scheme as DV-Hop. They do not require complex computations but need a huge amount of anchors to achieve a relatively low accuracy. The distance between 2 anchors is the finest achievable precision [15], [16]. Authors from [15] can reach 0.85 m of accuracy using a topology of 200 anchors randomly placed in a square of  $100 \text{ m}^2$ . An other algorithm based on Bayesian filtering [17] gives interesting results in *Indoor Localization Competition* of IPSN. This statistical-based method needs Bayesian and particle filtering coupled with a maximum-likelihood estimation. Such amount of computations does not match our previously mentioned constraints.

### III. PROPOSED ALGORITHM: BEST ANCHOR SELECTION FOR TRILATERATION (BAST)

The BAST algorithm aims at selecting a triplet of anchors taking into account predicted coordinates for mobile's position

---

**Algorithm 1** BAST Algorithm
 

---

**Input:**  $\mathbf{d}_A, \mathbf{a}_A, \mathbf{m}_{k-1}, \mathbf{m}_{k-2}$ 
**Output:**  $\mathbf{m}_k$ 


---

1: $d_A \leftarrow \text{Filter}(d_A)$ 2: $\hat{\mathbf{m}} \leftarrow \text{Predictor}(\mathbf{m}_{k-1}, \mathbf{m}_{k-2})$ 3: <b>for</b> $i \leftarrow 1, NbAnchors$ <b>do</b> 4: $A_1 \leftarrow \{d_{A_i}; \mathbf{a}_{A_i}\}$ 5: $(A_2, A_3) \leftarrow \text{NearestNonAlignedAnchors}(\mathbf{d}_A, \mathbf{a}_A)$ 6: $\mathbf{m}_i \leftarrow \text{Trilateration}(A_1, A_2, A_3)$ 7: $\delta_{MA_i} \leftarrow \text{EuclidianDistance}(\mathbf{m}_i, \hat{\mathbf{m}})$ 8: <b>end for</b> 9: $\mathbf{m}_{min} \leftarrow \text{GetMobile}(\min(\delta_{MA}))$ 10: $(A_1, A_2, A_3) \leftarrow \text{GetTriplet}(\min(\delta_{MA}))$ 11: $\mathbf{n} \leftarrow \text{Noise}(A_1, A_2, A_3, \mathbf{m}_{min})$ 12: $\mathbf{d}_{A_{Triplet}} \leftarrow \text{Distances}(A_1, A_2, A_3) - \mathbf{n}$ 13: $(A_1, A_2, A_3) \leftarrow \text{UpdateDistances}(\mathbf{d}_{A_{Triplet}})$ 14: $\mathbf{m}_k \leftarrow \text{Trilateration}(A_1, A_2, A_3)$	▷ Filter mistaken distances ▷ Estimate mobile's coordinates  ▷ Create reference anchor object ▷ Select the 2 nearest non-aligned anchors ▷ Compute trilateration ▷ Compute <i>ED</i>  ▷ Get mobile's coordinates from minimum <i>ED</i> vector ▷ Get triplet of anchors from minimum <i>ED</i> vector ▷ Compute noise distances vector from triplet ▷ Subtract noise to measured distances vector ▷ Update triplet of anchors ▷ Compute final trilateration
--	--

---

computation. This computation takes noise into account to improve measured distances. BAST steps are explained hereafter respectively to Algorithm 1.

At first, a filtering process aims at replacing mistaken values in  $d_A$  vector due to impossible or disturbed TOF measurements made by the UWB frontend. Secondly, mobile's coordinates are estimated using a one step linear forward predictor (assuming a mobile's constant speed) as defined in

$$\hat{\mathbf{m}} = 2 \times \mathbf{m}_{k-1} - \mathbf{m}_{k-2}, \quad (7)$$

where  $\hat{\mathbf{m}}$  are the estimated coordinates and  $\mathbf{m}_{k-1}$  and  $\mathbf{m}_{k-2}$  the two last computed coordinates from BAST.

Then, the aim of the *for* loop is to compute the Euclidian Distance (ED) between  $\mathbf{m}_i$  and  $\hat{\mathbf{m}}$  for  $NbAnchors$  triplets. Triplets are built by selecting a first anchor and then the 2 nearest non-aligned ones. Trilaterations are done on these triplets to get coordinates vector  $\mathbf{m}_i$ . In next step, the triplet corresponding to the minimal ED of  $\delta_{MA}$  is selected for noise computation  $\mathbf{n}$  as defined in

$$\mathbf{n} = \begin{pmatrix} d_{A1} - \sqrt{(x_{\mathbf{m}_{min}} - x_{A1})^2 + (y_{\mathbf{m}_{min}} - y_{A1})^2} \\ d_{A2} - \sqrt{(x_{\mathbf{m}_{min}} - x_{A2})^2 + (y_{\mathbf{m}_{min}} - y_{A2})^2} \\ d_{A3} - \sqrt{(x_{\mathbf{m}_{min}} - x_{A3})^2 + (y_{\mathbf{m}_{min}} - y_{A3})^2} \end{pmatrix}, \quad (8)$$

where  $\mathbf{m}_{min}$  corresponds to mobile's coordinates with minimal ED. The noise is finally subtracted to the triplet distances vector and a last trilateration allows to obtain the mobile's coordinates  $\mathbf{m}_k$ .

#### IV. EXPERIMENTAL SETUP

The experimental setup consists here in validating our algorithm and the state of the art ones by doing some measurements on our developed WSN platform in real environments. In general WSN and specific Wireless Body Area Network (WBAN), the unknown location of a node is determined in two steps. The first one is measurement step where one or more position-related metrics are measured to allow the computation

of relative distances between each pair of nodes. The second step called position estimation step (Section II) aggregates these distance measurements as input of a position estimator to produce the target/mobile node's position.

##### A. Zyggye node architecture

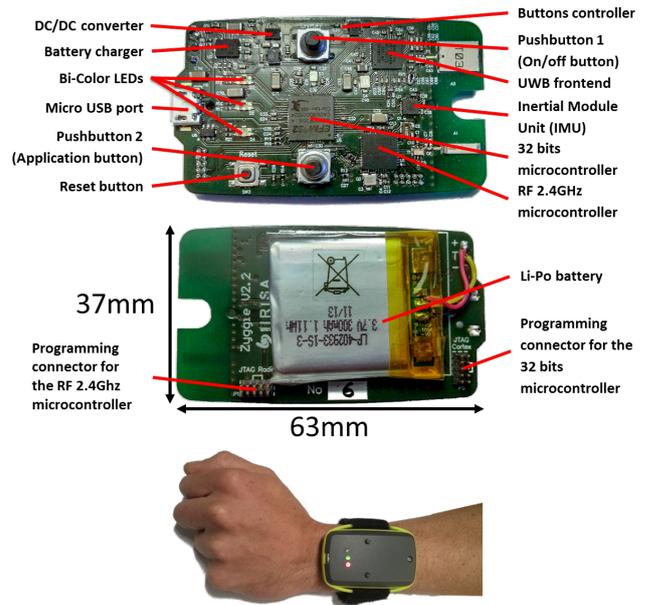


Fig. 1. Zyggye WSN Hardware illustration.

The Zyggye node [8] is a Wireless Sensor Node with UWB radio capabilities that was developed in our lab. Its architecture is represented in Fig. 1. Zyggye is composed of:

- An *Silicon Labs EFM32 ARM CortexM4* micro-controller which is the master of the node and also controls the UWB radio chip,

- An *ATMEGA2564RFR2* micro-controller which drives the 2.4 GHz radio for data exchanges and network synchronization,
- A *DecaWave DW1000 UWB* radio chip [18] for TOF measurements,
- An *InvenSense* Inertial Measurement Unit (IMU) for acceleration, angular speed and magnetic field captures.

Unlike solutions presented in [7], our node is ultra compact (37 mm by 63 mm) and only weights 32 g (including battery) which makes it a real good candidate for all applications where weight and size are critical. Some experiments with UWB radio enabled at full speed were done; it leads to a battery life time of around 5 hours which is enough for sportsman or in building applications. For both following contexts anchors are *Zyggie* nodes and the person to localize was wearing a *Zyggie* node on his wrist. Another *Zyggie* node is connected to a computer to synchronize the network (i.e. anchors and mobile nodes) via the 2.4 GHz radio and also to collect distances measurements via the UWB radio of the mobile node respect to the anchors. A custom TDMA MAC protocol allocates time slots to each anchor doing their TOF measurements. Data are then processed and plotted on the computer/tablet using the previous presented algorithms. In order to evaluate algorithms' performance, two contexts were defined: a quiet walking context and a fast moving context (i.e. tennis player).

### B. Indoor Serpentine Walking Context

For this context, data were acquired in a big room of nearly 243 m<sup>2</sup> (26.4 m length by 9.2 m width). The ideal trajectory was marked on the floor and one step per second (around 3 km/h) was done following this trajectory. To study the impact of the trajectory and the used anchors for the positioning algorithms, vertical and horizontal serpentine trajectories were acquired ten times. In order to test our BAST algorithm a six anchors configuration is used as represented in Fig. 2.

### C. Indoor Tennis-man Playing Context

For this context, data were acquired on a tennis court during real tennis playing conditions. All acquisitions were done in parallel with a high precision vision positioning system called *Vicon* [1]. Data issued from the *Vicon* system were considered as the reference to compute the performance of all algorithms. A four anchors configuration is used and the anchors locations were at the four corners of the court.

## V. RESULTS

Tables I and II present the performance results of all previously defined algorithms in terms of average error, error variance, variation of average error and variance and computation time. BAST serpentine trajectory is plotted in Fig. 2.

- The **average error** ( $\epsilon_S$ ) for the serpentine context is the mean error of all the recorded walking scenarios,
- The **average error** ( $\epsilon_T$ ) for the tennis player context is the mean error of all the recorded playing sequences,
- The **average standard deviation** ( $\sigma_S$  and  $\sigma_T$ ) is the mean of all standard deviations for both contexts,

- The **variation of the average error** ( $\Delta\epsilon_S$  and  $\Delta\epsilon_T$ ) and the **variation of the standard deviation** ( $\Delta\sigma_S$  and  $\Delta\sigma_T$ ) are the average gains of BAST compared to other algorithms,
- The **computation time** ( $T_{compS}$  and  $T_{compT}$ ) has been obtained running algorithms on *Matlab* on a desktop computer equipped with *Intel Xeon* processor and 8 GB RAM.

TABLE I  
ALGORITHMS COMPARISON FOR HORIZONTAL SERPENTINE CONTEXT

	BAST	Multilateration	Trilateration	Gauss-Newton	WCL
$\epsilon_S$ (m)	0.06	0.17	0.09	0.08	1.15
$\Delta\epsilon_S$		×2.83	×1.50	×1.33	×19.1
$\sigma_S$ (m)	0.06	0.19	0.18	0.07	0.92
$\Delta\sigma_S$		×3.17	×3	×1.17	×15.3
$T_{compS}$ (ms)	0.59	0.32	0.26	0.32	0.26

TABLE II  
ALGORITHMS COMPARISON FOR TENNIS CONTEXT

	BAST	Multilateration	Trilateration	Gauss-Newton	WCL
$\epsilon_T$ (m)	1.08	1.36	2.11	1.51	4.50
$\Delta\epsilon_T$		×1.26	×1.95	×1.40	×4.17
$\sigma_T$ (m)	0.76	0.98	1.68	1.72	0.96
$\Delta\sigma_T$		×1.29	×2.21	×2.26	×1.26
$T_{compT}$ (ms)	0.61	0.32	0.26	0.32	0.26

WCL algorithm is the worst algorithm due to the fact that all anchors try to attract the solution point towards them. Consequently, estimated mobile coordinates are not on the real trajectory.

For both contexts, trilateration works quite well. But the error can grow a lot when the mobile is not placed inside the virtual triangle formed by the 3 selected anchors.

As multilateration does the mean of all the available trilaterations, the previous mentioned phenomenon appears everywhere on the estimated trajectory which leads to a higher error for our contexts.

Gauss-Newton results are quite identical as BAST results for the serpentine context. Gauss-Newton works on the 6 anchors and as previously explained BAST chooses the best 3 anchors for trilateration. When movements are faster as in the real tennis playing game our BAST algorithm outperforms all other algorithms by giving at a minimum 1.26 and at a maximum 4.17 times better accuracy, to the penalty of a nearly doubled computation time which is still acceptable for real time executions and displaying.

By analyzing *Zyggie* IMU results, we have noticed that the

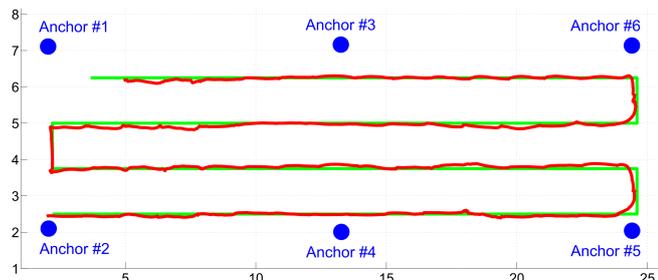


Fig. 2. BAST results: blue dots are anchors, red one is the mobile and green curve ideal trajectory. The estimated trajectory is in red.

TOF measurements were really drifting when the player hits the ball due to a high acceleration. We can conclude that for tennis player, placing a *Zyggie* node on the wrist may not be the ideal position. Some new experiments could be done with the node on the torso, on a foot or some other places.

We have noticed that for some places in the environment between two consecutive iterations, our BAST algorithm has to switch between anchors. This is due to the fact that they are all approximately at the same distance. An improvement could be to introduce a memory effect of the last 3 selected anchors (a temporal filter will be developed for this purpose) in order to decide to switch or not, as the same manner it is done in phone cellular network when your phone switches from a base station to another depending on your map position.

## VI. CONCLUSION

An enhanced algorithm for mobile indoor positioning was presented in this paper. Indoor positioning algorithms are based on distances measurements from a mobile node to anchor nodes. These distance measurements are issued from UWB radio exchanges which allow to do TOF computations. A WSN demonstrator called *Zyggie* has been developed for this purpose. *Zyggie* has the advantage to be wearable (small form factor), light and low power. Some real experiments were done using *Zyggie* in two contexts; a walking context and a tennis playing game context. Collected data were passed through BAST and low complexity state of the art algorithms for comparisons.

In all contexts but especially when the mobile moves fast, the BAST algorithm outperforms all other algorithms by giving at a minimum 1.26 and at a maximum 4.17 times better accuracy. Perspectives can be pointed out from the current work:

- An adaptive selection of the used algorithm could be done in real time depending on how fast the mobile moves and how much energy remains in the battery. For instance, as Gauss-Newton is less complex than BAST and gives nearly the same performance when the mobile moves slowly, computation time and energy consumption could be saved.

- Depending on the application, it can be more than one mobile to localize (team sports, visitors in a museum for instance), a multi-user protocol using anchors needs also to be explored.

## REFERENCES

- [1] "Vicon system." [Online]. Available: <https://www.vicon.com/>
- [2] N. L. Dortz, F. Gain, and P. Zetterberg, "Wifi fingerprint indoor positioning system using probability distribution comparison," in *the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 2301–2304.
- [3] M. Shchekotov, "Indoor localization methods based on wi-fi lateration and signal strength data collection," in *the 17th Conference of Open Innovations Association (FRUCT)*, April 2015, pp. 186–191.
- [4] Y. Wang, Q. Yang, G. Zhang, and P. Zhang, "Indoor positioning system using euclidean distance correction algorithm with bluetooth low energy beacon," in *the International Conference on Internet of Things and Applications (IOTA)*, Jan 2016, pp. 243–247.
- [5] S. Fang, C. Wang, T. Huang, C. Yang, and Y. Chen, "An enhanced zigbee indoor positioning system with an ensemble approach," *IEEE Communications Letters*, vol. 16, no. 4, pp. 564–567, April 2012.
- [6] L. Luoh, "Zigbee-based intelligent indoor positioning system soft computing," *Soft Computing*, vol. 18, no. 3, pp. 443–456, 2014.
- [7] "Microsoft indoor localization competition - ipsn 2018." [Online]. Available: <https://www.microsoft.com/en-us/research/event/microsoft-indoor-localization-competition-ipsn-2018/>
- [8] A. Courtay, M. L. Gentil, O. Berder, A. Carer, P. Scalart, and O. Sentieys, "Zyggie: a wireless body area network platform for indoor positioning and motion tracking," in *the IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2018.
- [9] J. Blumenthal, R. Grossmann, F. Golasowski, and D. Timmermann, "Weighted centroid localization in zigbee-based sensor networks," in *IEEE International Symposium on Intelligent Signal Processing (WISP)*, Oct 2007, pp. 1–6.
- [10] J. Nocedal and S. Wright, *Numerical Optimization: 10.3 Algorithms for Nonlinear Least-Squares Problems*. Springer New York, 2006.
- [11] B. Yang and J. Scheuing, "Cramer-rao bound and optimum sensor array for source localization from time differences of arrival," in *the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, March 2005, pp. 961–964.
- [12] J. Caffery, "A new approach to the geometry of toa location," in *the 52nd IEEE-VTS Fall Vehicular Technology Conference*, vol. 4, 2000, pp. 1943–1949.
- [13] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *the 9th International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 2003, pp. 81–95.
- [14] H. Chenji and R. Stoleru, "Toward accurate mobile sensor network localization in noisy environments," *IEEE Transactions on Mobile Computing*, vol. 12, no. 6, pp. 1094–1106, June 2013.
- [15] C. Jo and C. Lee, "Multilateration method based on the variance of estimated distance in range-free localisation," *Electronics Letters*, vol. 52, no. 12, pp. 1078–1080, 2016.
- [16] A. E. Assaf, S. Zaidi, S. Affes, and N. Kandil, "Accurate nodes localization in anisotropic wireless sensor networks," in *the IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB)*, Oct 2015, pp. 1–5.
- [17] D. Dardari, P. Closas, and P. M. Djurić, "Indoor tracking: Theory, methods, and technologies," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, April 2015.
- [18] "Decawave uwb chipset." [Online]. Available: <http://www.decawave.com/products/dw1000>