Towards Robust Methods for Indoor Localization using Interval Data

Nacim Ramdani*, Demetrios Zeinalipour-Yazti†, Michalis Karamousadakis♯, Andreas Panayides†

*Univ. Orléans, INSA CVL, PRISME EA 4229, F45072, Orléans, France,
†Department of Computer Science, University of Cyprus, 1678 Nicosia, Cyprus,
♯SingularLogic, Achaias 3 & Troizinias, Nea Kifisia 14564, Greece.
nacim.ramdani@univ-orleans.fr; dzeina@cs.ucy.ac.cy; mkaramousadakis@singularlogic.eu; panayides@cs.ucy.ac.cy

Abstract—**Indoor localization has gained an increase in interest recently because of the wide range of services it may provide by using data from the Internet of Things. Notwithstanding the large variety of techniques available, indoor localization methods usually show insufficient accuracy and robustness performance because of the noisy nature of the raw data used. In this paper, we investigate ways to work explicitly with range of data, i.e., interval data, instead of point data in the localization algorithms, thus providing a set-theoretic method that needs no probabilistic assumption. We will review state-of-the-art infrastructure-based localization methods that work with interval data. Then, we will show how to extend the existing infrastructure-less localization techniques to allow explicit computation with interval data. The preliminary evaluation of our new method shows that it provides smoother and more consistent localization estimates that state-of-the-art methods.**

Index Terms—**Indoor, Location, Interval Analysis, Modeling.**

I. Introduction

The omni-present availability of sensor-rich smartphones along with the fact that people spend 80-90% of their time in indoor environments has recently boosted an interest around the so called *Internet-based Indoor Navigation (IIN)* [1]. These comprise of indoor models, such as floor-maps and points-of-interest, along with Internet of Things (IoT)-based raw data, such as wireless, light and magnetic signals, used to localize and track mobile users and targets.

There is a large variety of localisation methods that exhibit diverse quality performance levels regarding precision, accuracy, cost, reliability, scalability, energy efficiency and robustness [2], [3]. One reason behind low performance usually observed in localization accuracy or robustness is the noisy nature of the IoT raw data used. For instance, the WiFi received signal strength (RSS), which is most commonly used by indoor localisation techniques, is in fact susceptible to multipath effects and interference, hence shows high variability over time. These variations may naturally introduce errors and jolts in reconstructed locations. To smoothen the location estimates and improve consistency, state-of-the-art localisation techniques work either with averaged signal data, or rely on more advanced probabilistic or Bayesian approaches [4], [5]. Other approaches use as well hybrid approaches combining RSSi-fingerprinting with inertial tracking systems as in [6] where the WiFi-based and the IMU-based location estimates, along with the associated uncertainties are provided as inputs to a data fusion module that implements the hybridization scheme by means of a particle filter. In practice however, the true probability distribution to use as Likelihood or a priori in the Bayesian methods are often unknown hence need be approximated using Gauss or uniform distributions. It is therefore appealing to consider an alternative description of the errors and disturbances acting on the measurements.

In this paper we will carry out a preliminary investigation on alternative methods to deal with the uncertainty in the measured signals by working directly with interval data, i.e. data ranges or data sets computed from the raw data with no assumption of the probability distribution within the interval. These interval data are feasible domains for measurements that take into account all the errors or disturbances acting on the data. This is known as the *Unknown But Bounded Error* (UBBE) framework: the errors are assumed to revolve in a *bounded set* with *known* bounds, but no other assumption is made on the probability distribution inside the set. Such description of the errors gathers both systematic and probabilistic errors, provided that the support domain of the probability distribution is bounded (see e.g. [7], and the references therein).

Let us discuss a simple example first. Let us assume that, in a 2D planar case, one performs multilateration and uses RSS data to compute the distance of a mobile target to beacons with known position by exploiting a path-loss model. Then, instead of using an average value for RSS one uses the range of RSS values obtained over a short time window. The uncertainty in the RSS data will lead to uncertainty in the distance measurement, which means that to be consistent with one distance measurement, the mobile target no longer lays on a circle, but can be on a ring. Using several beacons, one ends up with the mobile target located in several rings, hence in the intersection of them. With such range data, one no longer aims to compute a single point position for the mobile target, but rather computes a set of feasible positions for the mobile target. The size and the shape of the feasible

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 823887.

RISE MSCA ENDORSE project www.endorse-project.eu.

domain then gives quantitative information about the precision of the reconstructed position. Besides, the computed set can be regarded as an uncertainty set, i.e., a 100% confidence set for the mobile position.

The above example clearly demonstrates that the interval nature of the measurement data requires the use of dedicated set-theoretic methods as will be introduced later on. Set-theoretic methods combines complete and exhaustive-search solving algorithms with validated numerical implementations, that eventually provide *guaranteed* results in the sense that the existence or absence of a solution can be proven, and if it exists no solution is lost.

In the sequel, we will discuss the potential of such an approach for indoor location and more specifically IIN.

The core technology for IIN services is undoubtedly the type of hardware enabling the localization process. The taxonomy proposed in [8] focuses on whether new specialized hardware is needed to offer the location primitive, as opposed to either exploiting existing infrastructure (e.g., Wi-Fi network for the purpose of wireless connectivity) or no infrastructure whatsoever (e.g., inertial sensor systems). Infrastructure-free techniques exploit location-dependent measurements from existing wireless communication infrastructures, such as Wi-Fi access points and cellular base stations, without intervening on these systems. To the contrary, infrastructure-based technique explicitly exploit known position of beacons, anchors or features existing in the environment.

Infrastructure-based localization methods computing explicitly with interval signal data have already been developed in the literature. These techniques work in the UBBE framework and the feasible domains for the location estimates are obtained using guaranteed solving methods based on interval analysis [9] and solving techniques for constraint satisfaction problems [10]. Interval GPS pseudodistance were used to develop robust and intelligent vehicle navigation [11]. Interval data from range and goniometric measurements of given landmarks were used for underwater robot localisation [12]. Interval analysis and satisfiability checkers were used in [13] for tracking the position of several targets moving indoor using a network of error-prone binary sensors. All these methods are able to detect and identify erroneous measurements in real time simultaneously with the computation of the positioning domain using bounded-error measurements. They all rely on an explicit model mapping the spatial position of the target, the known position of the landmark and the measured data. For instance, [11] performs multilateration, while [12] uses also multitriangulation. These methods can easily exploit dead reckoning via data fusion as in [12]. Besides, they can naturally track multihypothesis [14].

To the best of our knowledge there is no RSS fingerprints-based localisation data that exploits *interval* data.

In this paper, we will first review state-of-the-art infrastructure-based localisation techniques that use interval data in Sect. II, along with the the required solving algorithms in Sect. III. Then introduce the preliminary blocks of our main contribution which is an infrastructure-free localisation technique that works explicitly with interval data. The technique is introduced in Sect. V, the required theoretical material is introduced in Sect. VI, and the experimental evaluation given in Sect. VII.

II. Infrastructure-based localisation techniques

In this section we are interested in localisation techniques based on signal metrics that can be mapped to the actual mobile target position using some parametric model. This holds for instance for techniques using Angle of Arrival, Time of Flight, Time Difference of Arrival, or RSS with path-loss propagation models [5].

A. System modelling

Without of generality, let us consider as a case study, the signal metrics Time of Flight (ToF) (also known as Time of Arrival), and Time Difference of Arrival (TDoA), in a planar indoor area $\mathcal{I}$. ToF exploits the signal propagation time to calculate the distance between the transmitter, radio beacon with known position, and the receiver. TDoA exploits the difference in signals propagation times from different transmitters measured at the receiver.

Let us consider N beacons at position $\vec{b}_i = (x_i, y_i)$, and denote $\vec{m} = (x, y)$ the unknown position of the mobile target. The ToF measurements $t(i)$ from transmitters i is converted into physical distance values $d(i) = c \cdot t(i)$, where c is the speed of light. The receiver $\vec{m}$ is now located on the circle given by

$$d(\vec{m}, i) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{1}$$

The TDoA measurements $\delta t(i, j)$ from transmitters i and j are converted into physical distance values $\delta d(i, j) = c \cdot \delta t(i, j)$. The receiver $\vec{m}$ is now located on the hyperbola given by

$$\delta d(\vec{m}, i, j) = d(\vec{m}, i) - d(\vec{m}, j) \tag{2}$$

Several sources of uncertainty in measuring the timings $t(i)$ and $\delta t(i, j)$ will eventually lead to uncertainty in the physical distance value. We assume an upper bound on the error is known, thus cast the estimation in the UBBE. Using the notations introduced in Sect. III, the measurement data obtained are now interval data, denoted $[d_i]$ for the distance to a beacon as in eq. (1), and $[\delta d_{i,j}]$ for the distance difference between two beacons as in eq. (2).

B. Robust localisation using interval data

Consider the case of multilateration with n beacons located at known positions.

In the UBBE, and using the notations and definitions of Sec. III, the set of feasible positions $\vec{m} = (x,\, y)$ of a mobile target consistent with index i ToF-based measurement (1) is given by

$$\mathcal{S}_{\mathtt{ToF}}(i) = \{(x,\, y) \,|\, d(\vec{m}, i) \in [d_i]\}. \tag{3}$$

The solution set using all the measurement available is then given by

$$\mathcal{S}_{\mathtt{ToF}} = \bigcap_{i \in \{1,..,n\}} \mathcal{S}_{\mathtt{ToF}}(i). \tag{4}$$

The validity of solution set (4) depends solely on (i) the correctness of the modelling assumptions, (ii) the validity of the measurement data, and (iii) the correctness of the bound considered for the measurement error. If one of the latter assumptions are violated, the solution set (4) may be empty. If one has confidence in the modelling, then the issue lays with the measurement data and the error bounds. Such data are known as *outliers*: either the sensors are faulty, are subject to large bias or disturbance noise of magnitude larger the error bound. Robustness to outliers can be achieved by relaxing the intersection (4) to let aside up to q outliers, hence up to q sets (3), as follows

$$\mathcal{S}_{\mathtt{ToF}} = \bigcap_{i \in \{1,..,n\}}^{q} \mathcal{S}_{\mathtt{ToF}}(i). \tag{5}$$

Similarly, when TDoA-based measurement data are used, the feasible sets are as follows. The solution set consistent with one datum is defined as

$$\mathcal{S}_{\mathtt{TDoA}}(k) = \{(x,\, y) \,|\, \delta d(\vec{m}, i, j) \in [\delta d_{i,j}]\}, \tag{6}$$

where $k \in \mathcal{K} = \{k = i{+}nj, \,|\, i \in \{1,..,n\},\, j \in \{i{+}1,..,n\}\}$. The feasible domain for the mobile location is then given by the set

$$\mathcal{S}_{\mathtt{TDoA}} = \bigcap_{k \in \mathcal{K}}^{q} \mathcal{S}_{\mathtt{TDoA}}(k). \tag{7}$$

Here again, using the relaxed intersection, we will be able to deal with outliers.

Solution sets such as (5) and (7) can be characterized in a reliable and guaranteed way using Constraint Satisfaction Problems (CSP) solving methods based on interval arithmetics, interval analysis, constraint propagation and branching [15], [10].

III. Interval Analysis

Interval analysis was initially developed to account for the quantification errors introduced by the rational representation of real numbers with computers, then was extended to validated numerics [9], [15].

A scalar (real) interval $[a] = [\underline{a},\, \overline{a}]$ is a closed and connected subset of $\mathbb{R}$, where $\underline{a}$ represents the lower bound and $\overline{a}$ represents the upper bound. The width of $[a]$ is defined by $wid([a]) = \overline{a} - \underline{a}$. An interval vector (interval box, or hyperrectangle) $[x]$ is a subset of $\mathbb{R}^n$ which is the Cartesian product of scalar intervals $[x] = [x_1] \times [x_2] \times \cdots \times [x_n]$, where the ith component is the projection of $[x]$ onto the ith axis. The maximal width of $[x]$ is $\overline{wid}([x]) = \max(wid([x_1], \ldots, wid([x_n]))$. The set of the real interval vectors of dimension n is denoted $\mathrm{I}(\mathbb{R}^n)$. Real arithmetic operations are extended to intervals. Consider an operator $\circ \in \{+, -, \times, \div\}$ and $[a]$ and $[b]$ two intervals. Then: $[a] \circ [b] = [\{u \circ v \,|\, u \in [a],\, v \in [b]\}]$.

Consider $g : \mathbb{R}^n \to \mathbb{R}^m$; the range of this function over an interval vector $[a]$ is given by: $g([a]) = \{g(a) \,|\, a \in [a]\}$. An *i*nterval extension $[g]$ of function g can be obtained by replacing each occurrence of a real variable by the corresponding interval and each standard function by its interval counterpart. The resulting function is called the natural interval extension. We can now state the following theorem:

Theorem 1 (The Fondamental Theorem of Interval Analysis. [9]): Any interval extension $[g]$ of function g that is inclusion isotonic, i.e., satisfies the property $[a] \subseteq [b] \Rightarrow [g]([a]) \subseteq [g]([b])$, is an inclusion function for g, i.e., $\forall [a] \in \mathbb{IR}^n$, $g([a]) \subseteq [g]([a])$.

The performance of the inclusion function depend on the formal expression of g.

IV. Mutlilateration with interval data

Let us now illustrate infrastructure-based robust localisation of a mobile target via multilateration with interval data. Let us consider the case of multilateration with $n = 4$ beacons located at positions $(0,\, 0)$, $(10,\, 0)$, $(10,\, 10)$, and $(0,\, 10)$ respectively.

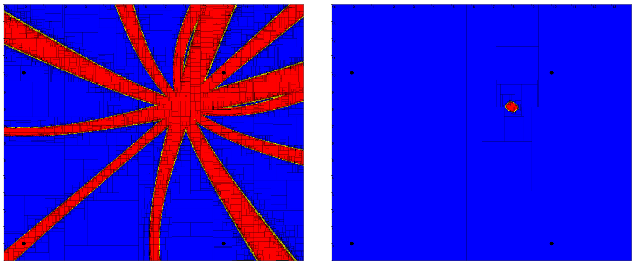
First let us consider TDoA-based multilateration. Fig. 1 shows on the left, the $n(n-1)/2 = 6$ hyperbola (6) obtained when computing Eq. (2) between the four beacons while considering bounded error on signal metrics. On the right, one can see the solution set (7) obtained (with $q = 0$). The black thick dots represent the beacons with known position. The red areas (boxes) correspond to the inner approximation of the solution set : the mobile target is certainly located within the set. The blue boxes correspond to non-solution areas : it is proven that the mobile target is not there. Besides, note that same algorithm is used to depict the solution set (7) on On Fig. 1-(b), and each hyperbola (6) on Fig. 1-(a).

In presence of outliers, e.g. induced by faulty time synchronization within one beacon, the 6 hyperbola are no longer consistent, i.e., the set (7) becomes empty. To provide robustness w.r.t such a case, one uses the $q-$relaxed intersection as in (7) to compute the solution set consistent with all but q data. Because TDoA computes time differences, the presence of synchronization errors in one beacon will impact the three TDoA where the beacon is involved. Fig. 2 shows a TDoA-based multilateration with $q = 3$. Localisation boils down to computing the intersection of 3 hyperbola out of 6. Fig. 2 shows that the solution set is composed of 4 disconnected subsets. Notice that the true position is included in the one of the subsets. The other subsets are there because it is still possible to find non-empty intersection between other combinations of 3 hyperbola. The spurious subsets may be eliminated by using extra beacons or exploiting other data modalities.

The presence of outliers does not always lead to disconnected subsets. For instance, Fig. 3 shows a ToF-based multilateration. On Fig. 3-(a), one can see the $n = 4$ rings (3) obtained when computing Eq. (1) with each beacon, while considering bounded error on signal metrics and one data outlier. On Fig. 3-(b), one can see the solution sets (5) obtained using the $q-$relaxed intersection with $q = 1$.

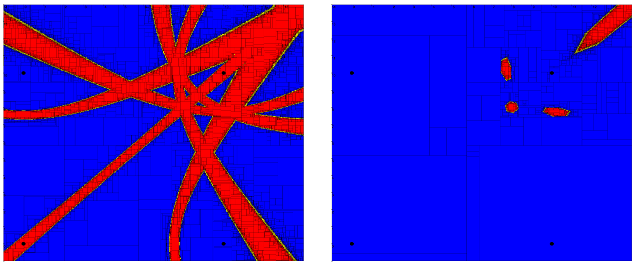
The computations are made using state-of-the-art solver pyIBex[1]. This is a set of python modules for solving nonlinear

[1] benensta.github.io/pyIbex



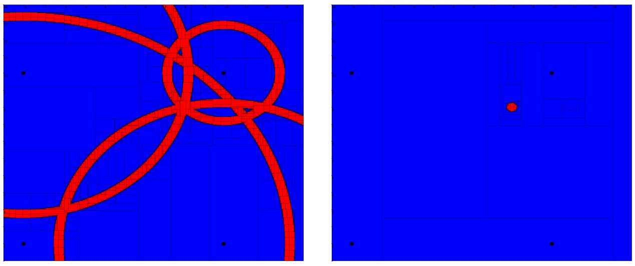
(a) The measurement data. (b) Solution set (7), $q = 0$.

Fig. 1. TDoA-based Multilateration with no outlier.



(a) The measurement data. (b) q-relaxed solution set.

Fig. 2. TDoA-based Multilateration with outliers. $q-$relaxed solution set (7) for $q = 3$.



(a) The measurement data. (b) $q = 1$

Fig. 3. ToF-based Multilateration with one outlier. The $q-$relaxed solution set (5).

problems using Interval Arithmetic tools. Initially based on the core part of Ibex-lib[2], it aims at combining the flexibility of python with the speed of C++ algorithms. The figures were built via the visualisation system VIBes[3].

V. Infrastructure-free localisation techniques

In this section we investigate ways to exploit the interval measurements data in Wi-Fi Radiomap-based indoor localization techniques such as the ones implemented in Anyplace software [1].

A. Anyplace software

Anyplace uses Wi-Fi Radiomap-based indoor localization, which stores radio signals from Wi-Fi APs in a database at a high density. The localization subsystem of Anyplace utilizes the following routine:

- In an offline phase, a logging application records the so called *Wi-Fi fingerprints*, which comprise of *Received Signal Strength (RSS)* indicators of Wi-Fi Access Points (APs) at certain locations (x, y) pin-pointed on a building floor map (e.g., every few meters).
- Subsequently, in a second offline phase, the Wi-Fi fingerprints are joint into a $N \times M$ matrix, coined the *Wi-Fi RadioMap*, where N is the number of unique (x, y) fingerprints and M the total number of APs.
- Finally, in the online phase, a user can compare its currently observed RSS fingerprint against the RadioMap in order to find the best match, using known algorithms such as K-nearest neighbour (KNN) or weighted KNN (WKNN) [4].

A similar methodology can be applied to other types of signals, for instance, magnetic fingerprints [16]. Both are considered infrastructure-free approaches, as Wi-Fi APs are ubiquitously available in urban and indoor spaces [17].

B. System model: Interval fingerprints

We assume a planar indoor area $\mathcal{I}$ containing a finite set of locations that are partially covered by a set of Wi-Fi APs $\{ap_1, ap_2, \cdots, ap_M\}$. Each ap_i has a unique ID (i.e., MAC address) that is publicly broadcast and passively received by anyone moving in the coverage of ap_i. The signal intensity v_i, $i = 1, ..., M$, at which the ID of ap_i is received at location $\vec{p_l} = (x_l, y_l)$, $l = 1, ..., N$ is termed the *Received Signal Strength (RSS)* of ap_i at $\vec{p_l}$, where -110 indicates when an ap_i is out of reach. The set of RSS values measured and the ap-IDs read at a location l is termed *fingerprint* $\vec{v_l} = (v_1, ..., v_M)$ of location l. During the online phase, the currently observed RSS fingerprint, denoted v_o, is used to provide an estimation of the actual position and the sought location $\vec{p_o}$. We further assume an indoor positioning server s that has constructed beforehand a *RadioMap* (RM), which is a database of offline *fingerprint* $\vec{v_l}$ measured at various locations $\vec{p_l} \in \mathcal{I}$.

Contrariwise to standard approaches, we further assume that at each unique location l, $l = 1, ..., N$, the range of variation of the signal intensity is captured, e.g. by sampling data during short time windows. Using the notations introduced in Sect. III, the RM now contains *interval fingerprint* $[\vec{v_l}]$ measured at location l. Fig. 4 shows for instance one interval vector $[\vec{v_l}]$ available in the RM (there are $M = 206$ AP). Note that the RSS interval range includes -110 which emphasises well the very noisy nature of RSS measurements. The actual coordinates of location l may also be subject to bounded uncertainty, i.e. $\vec{p_l} \in [\vec{p_l}] = ([x_l], [y_l])$.

The thus obtained *interval*-RM is stored in a database, where each entry T_l has the form

$$T_l = ([\vec{p_l}]; [\vec{v_l}]) \tag{8}$$

Finally, the observed RSS fingerprint during online phase is taken an interval vector $[\vec{v_o}]$.

[2] www.ibex-lib.org

[3] enstabretagnerobotics.github.io/VIBES

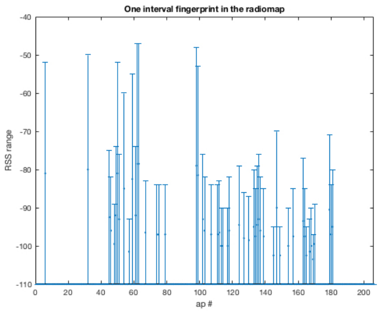


Fig. 4. Example of one interval fingerprint extracted from the radiomap.

C. *WKNN with interval data*

Since both the radiomap data and the signal measured by the mobile unit are interval data, we need to extend the WKNN approach to measure dissimilarities between interval vectors.

This idea is not new. A KNN classification method using interval data is proposed in [18], where the method mainly relies on identifying possible and necessary neighbours using the *partial* orders induced by some distance metrics computed with intervals. By construction, the method yields ambiguous decisions. To the contrary, other authors addressed the issue using *total* orders, for clustering interval data in [19], [20] and also in the framework of fuzzy sets in [21]. In these works, the distance metric used for comparing two interval vectors was the Hausdorff distance, a notion to be defined in Sect. VI, associated with the Chebyshev metric as it seems explicit formulas were readily available for online computation. However, when other metrics were used, the distance used was not the most appropriate to interval data. The reason it seems, is that the authors of these works did not find explicit formulas for computing the Hausdorff distance associated with the other metrics.

As it will be shown in the sequel, Jahn [22] gives explicit formulas that allow online computation of the Hausdorff distance h_p associated with the Minkowski norms (11). Classical WKNN can then be used to estimate the location of the mobile unit using the k nearest neighbours

$$[\hat{\vec{p}_o}] = \frac{\sum_{i=1}^{k} [\vec{p}_i]/d_i}{\sum_{i=1}^{k} 1/d_i},$$

where the distance are given by

$$d_i = h_p([\vec{v}_l], [\vec{v}_o]).$$

VI. Measuring distance between interval data

Computing the similarity between interval boxes requires specific metrics. A mere extension of standard point metrics to sets provides only a partial order. To obtain a total order on $\mathrm{I}(\mathbb{R}^n)$, we need specific metrics, as shown below.

A. *Hausdorff distance*

A metric on a set $\mathbb{X}$ induces a natural metric on the set of subsets of $\mathbb{X}$ known as the Hausdorff distance.

Definition 1 (Hausdorff Distance): Let $\mathbb{X}_1$ and $\mathbb{X}_2$ be two subsets of $\mathbb{X}$. The directed Hausdorff distance associated to the metric d is defined by

$$h_d^{\rightarrow}(\mathbb{X}_1, \mathbb{X}_2) = \sup_{x \in \mathbb{X}_1} \inf_{y \in \mathbb{X}_2} d(x, y).$$

The Hausdorff distance associated to the metric d is then

$$h_d(\mathbb{X}_1, \mathbb{X}_2) = \max\left(h_d^{\rightarrow}(\mathbb{X}_1, \mathbb{X}_2), h_d^{\rightarrow}(\mathbb{X}_2, \mathbb{X}_1)\right). \quad (9)$$

In the sequel, we consider the particular class of sets of $\mathbb{R}^n$ described on each variable x_i by an interval range, as described in Sect. III.

B. *Computing Hausdorff distance with interval vectors*

When working with interval vectors and targeting on-line implementation, explicit formulas should be derived for computing the Hausdorff distance (9). These formulas have been established by Jahn [22] for Hausdorff distances h_d associated with metrics d for which the following property holds

$$\text{for } x, y, u, v \in \mathbb{R}^n,$$
$$\forall i \in \{1, \dots, n\},\ |x_i - y_i| \le |u_i - v_i| \implies d(x, y) \le d(u, v). \quad (10)$$

This property holds also for the Minkowski metric d_p

$$d_p(x, y) = \left(\sum_i |x_i - y_i|^p\right)^{1/p}, \quad (11)$$

as well as its weighted version. Finally, we can rely on explicit formulas for the Manhattan ($p = 1$), the Euclidian ($p = 2$) and the Chebyshev ($p \to \infty$) metrics, although for the latter, the computation of the Hausdorff distance is straightforward. In the sequel, we will denote by h_p the Hausdorff distance associated with metric d_p, in particular h_1 associated with the Manhattan ($p = 1$), h_2 with the Euclidian ($p = 2$), and h_∞ with the Chebyshev ($p \to \infty$) metrics.

As known, the Hausdorff distance h_∞ associated with metric d_∞ can be computed as

$$h_\infty([a], [b]) = \max(\max_i |\underline{a}_i - \underline{b}_i|, \max_i |\overline{a}_i - \overline{b}_i|)$$

We can introduce the mapping $\psi : \mathrm{I}(\mathbb{R}^n) \to \mathbb{R}^{2n}$ defined by $\pi([x]) = (\underline{x}_1, \dots, \underline{x}_n, \overline{x}_1, \dots, \overline{x}_n)$. Then for any interval vectors $[a]$, $[b]$ in $\mathrm{I}(\mathbb{R}^n)$, the equality

$$h_\infty([a], [b]) = d_\infty(\pi([a]), \pi([b])) \quad (12)$$

holds. The sought explicit formulas should then be of the form of (12). They are given in [22].

VII. Experimental evaluation of the localisation method using interval fingerprints

Le us illustrate indoor localisation using a RM database that contains interval fingerprints as explained in Sect. V-B.

The *interval*-RM contains $N = 52$ interval fingerprints obtained at positions covered by $M = 206$ unique AP. At each position, several RSS measurements were made over a short time interval. The range of RSS values obtained were used to

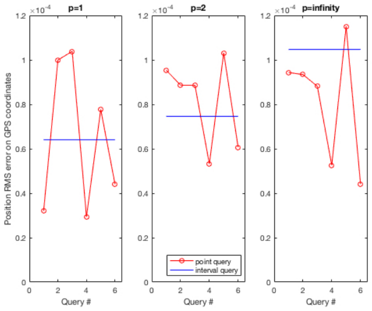


Fig. 5. RMS position error using interval data query with interval fingerprints *vs* point data query with averaged fingerprints. (a) $p = 1$, (b) $p = 2$, (c) $p \to \inf$.

compute the interval fingerprint, along with an averaged value, in order to compare the two approaches, i.e., the interval data approach proposed in this paper with the standard approach using averaged values.

Fig. 5 shows the position RMS error at a given location. The position is reconstructed using the RM. On the one hand, we use a range of RSS measured over a short time interval and search for similarities in the *interval*-RM, the RMS errors are denoted *interval query*. On the other hand, we use each of the single measurements and search similarities in the *averaged*-RM. The RMS errors are denoted *point query*. This is done for the three metrics obtained using Hausdorff distance h_p versus standard metric d_p, when $p = 1$, i.e., Manhattan metric, when $p = 2$, i.e., Euclidian metric, and when $p \to \infty$, i.e., Chebyshev metric. It is clear that the *point query* method exhibits large variations in the reconstructed position.

Therefore, the interval fingerprint method advocated in this paper may be used as a standalone technique to smoothen the location estimates and improve the consistency, whereas state-of-the-art localisation techniques need to alter the measurements by averaging or need to rely on other external modalities hence using extra hardware.

VIII. Concluding remarks

We have shown that there exist effective algorithms that make it possible to use interval data within indoor localisation techniques.

With infrastructure-based techniques, interval data may be used to provide uncertainty bounds for the reconstructed localisation, thus ensuring more reliable navigation in noisy environment without the need of probabilistic assumptions.

With infrastructure-less techniques, the new method we propose for RSS fingerprint-based localisation works directly with interval RSS data to provide smoother and more consistent localisation estimates. Future work will consider further experimental evaluation of the new method and its implementation into the Anyplace software.

References

[1] D. Zeinalipour-Yazti and C. Laoudias, “The anatomy of the anyplace indoor navigation service,” in *ACM SIGSPATIAL Special*, ser. SIGSPATIAL‘17, vol. 9, no. 2. ACM Press, 2017, book, pp. 3–10.

[2] A. Basiri, E. S. Lohan, T. Moore, A. Winstanley, P. Peltola, C. Hill, P. Amirian, and P. F. e Silva, “Indoor location based services challenges, requirements and usability of current solutions,” *Computer Science Review*, vol. 24, pp. 1 – 12, 2017.

[3] G. Oguntala, R. Abd-Alhameed, S. Jones, J. Noras, M. Patwary, and J. Rodriguez, “Indoor location identification technologies for real-time iot-based applications: An inclusive survey,” *Computer Science Review*, vol. 30, pp. 55 – 79, 2018.

[4] B. Li, J. Salter, A. G. Dempster, and C. Rizos, “Indoor positioning techniques based on wireless LAN,” in *1st IEEE Int Conf on Wireless Broadband and Utra Wideband Communications*, 2006, pp. 13–16.

[5] F. Zafari, A. Gkelias, and K. K. Leung, “A survey of indoor localization systems and technologies,” *CoRR*, vol. abs/1709.01015, 2017.

[6] C.-L. Li, C. Laoudias, G. Larkou, Y.-K. Tsai, D. Zeinalipour-Yazti, and C. G. Panayiotou, “Demo: Indoor geolocation on multi-sensor smartphones,” in *Proceedings of the 11th International Conference on Mobile Systems, Applications and Services*, ser. Mobisys‘13, Taipei, Taiwan, June 25 - 28, 2013, conference, pp. 503–504.

[7] C. Combastel, V. Puig, T. Rassi, and T. Alamo, “Special issue: Set-membership methods applied to FDI and FTC,” *International Journal of Adaptive Control and Signal Processing*, no. 30, pp. 147–426, 2016.

[8] D. Zeinalipour-Yazti, C. Laoudias, K. Georgiou, and G. Chatzimilioudis, “Internet-based indoor navigation services,” *IEEE Internet Computing*, vol. 21, no. 4, pp. 54–63, 2017.

[9] R. E. Moore, *Interval Analysis*. Englewood Cliffs, N.J.: Prentice-Hall, 1966.

[10] G. Chabert and L. Jaulin, “Contractor programming,” *Artificial Intelligence*, vol. 173, no. 11, pp. 1079–1100, Jul 2009.

[11] V. Drevelle and P. Bonnifait, “Reliable positioning domain computation for urban navigation,” *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 3, pp. 21–29, Fall 2013.

[12] B. Desrochers and L. Jaulin, “A minimal contractor for the polar equation: Application to robot localization,” *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 83 – 92, 2016.

[13] M. H. Amri, Y. Becis, D. Aubry, N. Ramdani, and M. Fränzle, “Robust indoor location tracking of multiple inhabitants using only binary sensors,” in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2015, pp. 194–199.

[14] E. Colle and S. Galerne, “A multihypothesis set approach for mobile robot localization using heterogeneous measurements provided by the internet of things,” *Robotics and Autonomous Systems*, vol. 96, pp. 102 – 113, 2017.

[15] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis*, Springer, Ed. London: Springer, 2001.

[16] A. Nikitin, G. Chatzimilioudis, C. Laoudias, P. Karras, and D. Zeinalipour-Yazti, “Indoor localization accuracy estimation from fingerprint data,” in *Proceedings of the 18th IEEE International Conference on Mobile Data Management (MDM’17)*, 2017, pp. 196–205.

[17] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, “A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned,” in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, ser. IPSN’15. ACM, 2015, pp. 178–189.

[18] V.-L. Nguyen, S. Destercke, and M.-H. Masson, “K-Nearest Neighbour Classification for Interval-Valued Data,” in *11th International Conference on Scalable Uncertainty Management (SUM 2017)*, ser. Lecture Notes in Computer Science, no. 10564, Oct. 2017, pp. 93–106.

[19] R. M. de Souza and F. de A.T. de Carvalho, “Clustering of interval data based on city-block distances,” *Pattern Recognition Letters*, vol. 25, no. 3, pp. 353–365, 2004.

[20] M. Chavent, “A Hausdorff distance between hyper-rectangles for clustering interval data,” in *Classification, Clustering, and Data Mining Applications.*, D. B. D., F. McMorris, P. A. P., and W. Gaul, Eds. Springer, Berlin, Heidelberg, 2004.

[21] F. de A.T. de Carvalho and E. C. S. oes, “Fuzzy clustering of interval-valued data with city-block and hausdorff distances,” *Neurocomputing*, vol. 266, pp. 659 – 673, 2017.

[22] K.-U. Jahn, “Evaluation of hausdorff distances in interval mathematics,” *Computing*, vol. 45, no. 1, pp. 69–77, Mar 1990.