Adaptive Fingerprinting in Multi-Sensor Fusion for Accurate Indoor Tracking

Alberto Belmonte-Hernández, Gustavo Hernández-Peñaaloza, Federico Álvarez, Giuseppe Conti

Abstract—Indoor Localization and Tracking have become an attractive research topic because of the wide range of potential applications. These applications are highly demanding in terms of estimation accuracy and rise a challenge due to the complexity of the scenarios modeled. Approaches for these topics are mainly based on either deterministic or probabilistic methods such as Kalman or Particles Filter. These techniques are improved by fusing information from different sources such as wireless or optical sensors. In this paper, a novel Multi-sensor Fusion using Adaptive Fingerprint (MUFAF) Algorithm is presented and compared with several multi-sensor indoor localization and tracking methods. MUFAF is mainly divided in four phases: first, a Target Position Estimation (TPE) process is performed by every sensor; second, a Target Tracking Process (TTP) stage; third, a Multi-Sensor fusion (MMF) combines the sensor information and finally, an Adaptive Fingerprint Update (AFU) is applied. For TPE, a complete environment characterization in combination with a Kernel Density Estimation (KDE) technique are employed to obtain object position. A Modified Kalman Filter (MKF) is applied to TPE output in order to smooth target routes and avoid outliers effect. Moreover, two fusion methods are described in this work: Track-To-Track Fusion (TTTF) and Kalman Sensor Group Fusion (KSGF). Finally, AFU will endow the algorithm with responsiveness to environment changes by using Kriging interpolation to update the scenario fingerprint. MUFAF is implemented and compared in a testbed showing that it provides a significant improvement in estimation accuracy and long-term adaptivity to condition changes.

Index Terms—Wireless, Receive Signal Strength Indicator (RSSI), Indoor Tracking, Kernel Density Estimation, Kalman filter, Multi-Sensor Fusion.

I. INTRODUCTION

Indoor positioning and tracking have attracted an extensive research effort because of their usefulness for a broad range of applications such as audience pattern generation [1], costumer analysis in retail [2], surveillance [3], business related activities [4], healthcare behavioral monitoring [5] among others [6], [7].

The aim is mainly to extract location and routes from indoor targets to provide applications which make use of both estimated values. One approach is the use of cameras [3], [8], but cost, occlusions, poor/inadequate lighting conditions or privacy issues limit their applicability in several scenarios. In this paper we will focus on a very common approach which is the use of a sensors deployment to detect and track the objects, however the precision in the location and tracking is key to the applications using such data.

Recent research works face this problem by implementing wireless sensors in combination with processing algorithms that combine data gathered from multiple nodes [2], [6], [7]. Within the methods to estimate indoor position, most popular are Time-of-Arrival (ToA) [9], Time Difference of Arrival (TDoA) [10], Angle-of-Arrival (AoA) [11] and Received Signal Strength Indicator (RSSI) measurements [12], being the techniques more often employed ToA and RSSI. On the one hand, ToA techniques use timestamps lags from a sent/received packet. These methods allow high precision in the final estimation with relative low background processing. However, hardware cost limitations in addition to strict synchronization requirements increase the system complexity making this approach hard to implement. On the other hand, RSSI values can be measured by a large number of wireless devices but RSSI modeling is generally tedious due to the harsh propagation conditions and processing techniques to achieve a good accuracy [18].

Because of the high potentiality of indoor tracking applications, low cost and the wide range of devices that can be used, the technical approach presented in this work relies on RSSI-based estimation.

RSSI position estimation works have been presented in the literature for indoor tracking [13], [14], [15], [16], employing theoretical propagation models to estimate the distance and/or in combination with deterministic methods for tracking [14]. However, these propagation models neglect the obstacles, which yields to a high error rate in estimations.

To correct and improve such problem with obstacles, a very extended approach is based on the employment of Fingerprinting technique which characterizes the scenario by splitting into cells and performing spatial sampling [13], [14] with the aim of modeling the signal propagation in the grid. This technique can be applied for both deterministic and probabilistic methods. Deterministic methods depend heavily on the resolution of the fingerprint cells: the shorter the number of cells is, the lower the estimation accuracy is. Probabilistic methods consider a few sensors introducing a higher mathematical treatment to fuse the information gathered [15], [16] and in this way result get improved.

Additionally, recent works incorporate an adaptive update stage to improve the algorithmic accuracy [17], [19] which is a growing research topic as it allows to adjust fingerprint-based technology to changes in environment conditions.

Therefore, following this line to improve the accuracy and motivated by the heavy estimation requirements in terms of precision for indoor tracking, and considering the large availability of commercial devices (e.g. smartphones) with several sensors (wireless communication standards) in this work a Multi-sensor Fusion based on Adaptive Fingerprinting (MUFAF) Algorithm is presented to perform the object track-
ing (in this paper we used IEEE 802.15.1, IEEE 802.15.4 and IEEE 802.11) based on RSS measurements.

To achieve the presented aims, that in the paper are compared to other approaches, MUFAF works as follows. MUFAF starts by performing a statistical position estimation called Target Position Estimation (TPE) in each of the available sensors (measuring in the different wireless interfaces). A complete environment characterization in combination with a Kernel Density Estimation (KDE) technique are employed to obtain object position. and a Modified Kalman Filter (MKF) is applied to TPE output in order to smooth target routes and avoid outliers effect.

Once data is captured, a Target Tracking Procedure (TTP) is done to obtain the object routes from individual sensors. Subsequently, the fusion of the sensors information is processed by a Multi-Sensor fusion (MMF) process which combines the sensor data using Track-To-Track Fusion (T2TF) and/or Kalman Sensor Group Fusion (KSGF), which are described and compared in the paper.

Last step is to provide an Adaptive Fingerprint Update (AFU) to cope with the environment changes. AFU uses a novel adaptive fingerprint update technique to adjust the algorithm according to scenario variations based on Kriging interpolation.

The remainder of this paper is organized as follows: In next section II, the problem statement, notation and assumptions are addressed. III, Target Position Estimation (TPE) is detailed. Section IV describes the Target Tracking Procedure (TTP). In section V, the comparison of fusion pattern architectures is defined. In section VI adaptive fingerprinting technique using Kriging is presented. Moreover, in section VII the implementation details and results of the algorithms are illustrated. Finally, conclusions and future work are drawn in section VIII.

II. PROBLEM STATEMENT

Complex indoor environments have been considered, where harsh signal propagation conditions present a challenge for appropriate characterization. For this aiming, it is assumed an indoor scenario with a set \( \mathbf{v} = \{v_i, \lambda \mid i = 1, \ldots, I; \lambda = 1, \ldots, \Lambda \} \) of Access Points (AP) located at Cartesian coordinates \( \mathbf{p}_i = (x_i, y_i) \in \mathbb{R}^2 \). Furthermore, let \( \Lambda \) be the number of sensors that each node is equipped with. In this work, four sensors from three wireless technologies are considered: IEEE802.11.g/Wi-Fi [53], IEEE802.15.1/Bluetooth v4.0 [51], IEEE802.15.4/XBee and IEEE802.15.4/CC2420 [52]. Network nodes are able to receive the Obtained Signal Strength Indicator, denoted by \( \text{RSS} \), from the monitored target. Consequently, \( \text{RSS}_{i,\lambda} \) describes the RSS measurements gathered by node \( i \) using technology \( \lambda \).

The main problems addressed in this work are three: \( a \) the accurate localization estimation of the target by means of a combination of available RSSI measurements sensed by the nodes; \( b \) tracking of the detected object in an indoor environment and \( c \) the improvement in trajectories accuracy by means of multi-sensor fusion of the different technologies considered. To tackle these challenges, several considerations are taken:

- RSS is assumed to be an independent and identically distributed (i.i.d.) Random Variable.
- Every node \( v_i, \lambda \) is able to gather a set of RSS measurements from the aforementioned technologies \( \lambda \). Notice that it is possible to obtain \( \text{RSS}_{i,\lambda} \rightarrow \emptyset \).
- Additionally, it is assumed that \( \text{RSS}_{i,\lambda} \) is a vector containing synchronized measurements from all nodes \( I \) for a particular technology \( \lambda \).
- A window-time, denoted by \( k \), is considered. Data gathered in this period will be mathematically treated as synchronized. For experiments performed in this paper, \( k = 1 \) sec. There are several factors that can affect this window-time. Firstly, the frequency sampling of the de-

---

TABLE I

MUFAF NOTATION

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda )</td>
<td>a wireless sensor technology.</td>
</tr>
<tr>
<td>( \mathbf{v} )</td>
<td>a set of nodes in the network.</td>
</tr>
<tr>
<td>( \mathbf{p}_{x,y} )</td>
<td>Estimated position for a particular technology.</td>
</tr>
<tr>
<td>( \mathbf{v}_{x,y} )</td>
<td>general definition of a node.</td>
</tr>
<tr>
<td>( \text{RSS}_{x,y} )</td>
<td>a node using a particular technology.</td>
</tr>
<tr>
<td>( c = {p_1, \ldots, L} )</td>
<td>a set of fingerprint cells.</td>
</tr>
<tr>
<td>( K(\cdot) )</td>
<td>a window time for fingerprinting in a cell.</td>
</tr>
<tr>
<td>( \mathbf{d}(p_1, p_2) )</td>
<td>Kernel Function and bandwidth.</td>
</tr>
<tr>
<td>( F(p_1, \lambda, \text{RSS}) )</td>
<td>Euclidean distance from ( p_1 ) to ( p_2 ).</td>
</tr>
<tr>
<td>( \text{AFU} )</td>
<td>Fingerprin Matrix.</td>
</tr>
<tr>
<td>( \text{TPP} )</td>
<td>Target tracking Process (TTP).</td>
</tr>
<tr>
<td>( k )</td>
<td>Time step for MUFAF iteration.</td>
</tr>
<tr>
<td>( \mathbf{x} )</td>
<td>system state for Kalman Filter.</td>
</tr>
<tr>
<td>( \mathbf{P} )</td>
<td>Kalman Transition Matrix.</td>
</tr>
<tr>
<td>( \mathbf{u} )</td>
<td>System input (Random Walk Model RWM).</td>
</tr>
<tr>
<td>( \mathbf{B} )</td>
<td>Kalman Control Matrix (RWM).</td>
</tr>
<tr>
<td>( \mathbf{G} )</td>
<td>Kalman Gain Matrix.</td>
</tr>
<tr>
<td>( \mathbf{H} )</td>
<td>Measurements Prediction.</td>
</tr>
<tr>
<td>( \mathbf{Q} )</td>
<td>Error covariance using RWM.</td>
</tr>
<tr>
<td>( \mathbf{R} )</td>
<td>Error covariance matrix from Observations.</td>
</tr>
<tr>
<td>( \mathbf{F} )</td>
<td>Noise Process.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>TPP Input: estimated position in TPE.</td>
</tr>
<tr>
<td>( \mathbf{P} )</td>
<td>Observation.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Cartesian coordinates in axis ( x ) and ( y ).</td>
</tr>
<tr>
<td>( \mathbf{P} )</td>
<td>Covariance Matrix from Kalman Process.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Multi-Sensor Fusion:</td>
</tr>
<tr>
<td>( \mathbf{P} )</td>
<td>A route from a target tracked.</td>
</tr>
<tr>
<td>( \mathbf{P} )</td>
<td>Number of steps that compose a route.</td>
</tr>
<tr>
<td>( \mathbf{P} )</td>
<td>Order of ( \mathbf{P} ) Matrix.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Covariance Matrix from Kalman process in a time ( k ) for a technology ( \lambda ).</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>position estimated after fusion stage.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Additionally, ( x_m, y_m ) are the spatial coordinates employed as input to the alpha-beta filter.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>control parameters of Alpha-Beta filter.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Adaptive Fingerprint Update (AFU).</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Set of distances in empirical Semivariogram.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Empirical Semivariogram.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Theoretical Semivariogram.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Fitting parameters in ( \mathbf{r}(\cdot) ).</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Width and Depth of a room.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>weights in Kriging estimation.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Semivariogram Matrix for prediction.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Lagrange Multiplier.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>vector containing ( v ) and ( e ).</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Similarity measure to estimated location.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Number of neighbors chosen for AFU.</td>
</tr>
<tr>
<td>( \mathbf{P}_{x,y} )</td>
<td>Decision Threshold for AFU.</td>
</tr>
</tbody>
</table>
vices considered in this work is high which allows to have a representative data-set for position estimation. It implies that the window-time can be reduced. However, outliers can decrease the estimation quality as a consequence of overestimation. In concrete, IEEE802.15.4Xbee is able to forward up to 30 packets per second (pps); IEEE802.11g-Wi-Fi frequency sampling is above 60 pps and IEEE802.15.1 Bluetooth is up to 20 pps. Secondly, the velocity of persons moving in the scene is a key parameter with a wide range of values. When a person is running in the scene, a shorter window-time can be appropriated for the position estimation. Nonetheless, the aforementioned problem can arise. Moreover, some approaches quantify an average of a normal person walking to be around 130 cms /second [29], [30]. In tests performed, 1 Second time window provides an optimal trade-off.

In order to maintain general conventions on the techniques presented, an overview of the notation employed in this work is detailed in table 1.

A. Main contributions

The key contribution of this work is the proposal of a complete system for indoor localization and tracking that overcomes the difficult conditions of the scenarios in terms of propagation interferences. Moreover, MUFAF is scalable and modular. Experiments performed have shown that the algorithms described are able to track several objects simultaneously. Furthermore, MUFAF outperforms existing techniques therefore it can be considered as an adaptive system fusing multiple data sources. The main technical contributions of this paper are:

- Several multi-sensor architecture patterns for tracking systems are compared. Some of the most precise methods for indoor tracking are described and assessed in this work. As a result, MUFAF Algorithm for indoor localization using multi-sensor fusion from various technologies.
- An Adaptive Fingerprint Update (AFU) procedure is proposed to ensure that MUFAF will change according to the varying scenario conditions. This algorithm is based on Kriging interpolation that is a statistical technique appropriated for the position estimation. Nonetheless, KDE employs the Fingerprinting data to estimate the target location.

Algorithm 1: MUFAF: Fingerprint Stage

| Inputs: v → set of Access Points // e → set of fingerprint cells. // RSS$_{i,\lambda}$: RSSI measured by Access Point i by technology $\lambda$. // $\tau$ is the predefined fingerprinting period and t is a timer. |
| Outputs: F → Fingerprint Matrix. |
| 1: procedure FINGERPRINT TRAINING |
| 2: for all nodes in v do |
| 3: for each cell l in e do |
| 4: $p_l = (x_l, y_l)$ |
| 5: for all technologies $\lambda$ do |
| 6: while $t - \tau > 0$ do |
| 7: $F(p_l, \lambda, RSS) \leftarrow RSS_{i,\lambda}$ |
| 8: end while |
| 9: end for |
| 10: end for |
| 11: end procedure |

position estimation applied to sensor RSSI measurements. Large variance of measurements gathered is one of the most critical problems associated to RSSI estimation, specially for indoor environments, even in case of static objects [20]. An example of this variance is shown in figure 1 where a large amount of RSSI measurements are taken from a static object equipped with several sensors. A common approach to deal with this issue is by applying Fingerprinting Technique [21], [22], [23]. This technique allows to obtain a proper scenario calibration by splitting it into a set of cells $e = \{p_l| l = 1, \ldots, L\}$ with geometrical center at $p_l = (x_l, y_l) \in \mathbb{R}^2$ respectively. The larger the number of cells $L$ is, the higher the spatial resolution is. Afterward, measures are taken from each node $v_l$ with the target object located at the center of every cell $p_l$ for a predefined time interval $\tau$. The longer the calibration time is, the better the cell characterization is. The entire Fingerprint process is detailed in algorithm 1.

As a result of this procedure, a distribution of the RSSI is captured in every cell for every technology $\lambda$ by every node $v_l$. For this purpose, the multi-dimensional $F$ matrix has been defined to contain the fingerprinting distributions. Therefore, $F(p_l, \lambda, RSS)$ represents the set of fingerprint measurements gathered at cells position $p_l \in \mathbb{R}^2$ by sensor technology $\lambda$ by nodes $v_l \in v$. Moreover, $F$ matrix is the input of the TPE process described in next subsection. As an example, in figure 2 are shown the fingerprint cells location for the experiments room presented in this work (left) and some examples of RSSI fingerprinting distributions (right).

B. Target Position Estimation (TPE)

There exist several works in the literature for position estimation based on Fingerprint statistics [39], [41], [42], [43]. Most of these works apply deterministic techniques
to characterize the RSSI distribution for each cell. In [39], the K-Nearest Neighbors (KNN) technique is employed to triangulate the target position adopting euclidean distance as metric. In [41], a comparison of various metrics is carried out in an indoor building giving as result that Mahalanobis distance outperforms Manhattan and Euclidean-Distance results. Furthermore, in [42] a weighted K-Nearest Neighbor approach is presented. Moreover, in [43] it is proposed a complete overview of several deterministic methods that include all the aforementioned metrics. However, taking into consideration a large set of measures collected for every cell in Fingerprint stage, a deterministic approach can neglect the wide distribution of measurements per cell, therefore a probabilistic approach can be considered for the estimation.

Kernel Density Estimation (KDE) is a non-parametric technique employed to obtain the Probability Density Function (PDF from now on) of a random variable with independent and identically distributed samples [24], [25]. KDE estimates the likelihood that Fingerprintsing distributions match the measures in the region of a Kernel function. Therefore, the distance is fitted with Kernel functions, denoted by $K(\cdot)$ and the PDF in our problem can be calculated by the following function:

$$p(\text{RSS}_i, \lambda | p_j) = \frac{1}{Nh_N} \sum_{n=1}^{N} K \left( \frac{\text{RSS}_i, \lambda - F(p_i, \lambda, \text{RSS}_i^0)}{h} \right)$$

(1)

where $N$ is the set of samples in every fingerprinting cell center $p_j$ by the node $v_i$. Furthermore, $h$ is a smoothing parameter also known as kernel bandwidth. $\text{RSS}_i, \lambda$ is the node measurement. In addition, $F(p_i, \lambda, \text{RSS}_i^0)$ is the set of RSSI values gathered by node $v_i$ at fingerprint cell $p_j$ with the corresponding technology $\lambda$.

The goal pursued is to estimate a non-parametric function that fits the RSSI values distribution better than known (parametric) distributions. Due to the nature of the data modeled, a modified version of KDE known as Nadaraya-Watson Kernel regression is used [26]. This method is applied because of its appropriateness when there is no prior knowledge of the relationship between the variables under study since these estimators are only based on either smoothing or regression functions.

There are other non-parametric options that could have been employed such as the Priestley-Chao and Gasser-Miller smoother or the K-th Nearest-Neighbor (K-NN) weights [40]. However, Priestly-Chao and Gasser-Miller strict boundary bias problems make Nadaraya-Watson method the most suitable estimator for RSSI-based applications. In fact, it has been demonstrated that Nadaraya-Watson variance is up to a 50% lesser than the other methods [40]. However, K-NN smoothing can attain similar results to the ones obtained by using kernel estimation when an appropriate bandwidth $h$ parameter is adjusted.

According to this method, the joint probability function can be estimated as:

$$p(p, \text{RSS}_i, \lambda) = \frac{1}{L} \sum_{l=1}^{L} \frac{1}{(h_N)^2(h_N)} K \left( \frac{d(p, p_l)}{h_N} \right)$$

(2)
where \( d(p, p_l) \) represents the distance from every fingerprinting cell center to the estimated point and \( h_x, h_N \) are the bandwidth parameters respectively. Finally, the expectation \( \mathbb{E}[F(p_l, \lambda, \text{RSS})] \) yields to a vector containing the RSS mean for every node.

Applying the Minimum Mean Square Error (MMSE) criteria over the conditional density probability obtained by Bayes theorem:

\[
z = \hat{p}_{\text{MMSE}} = \mathbb{E}(p | \text{RSS}_l, \lambda) = \int_G p(p | \text{RSS}_l, \lambda) \, dx
\]

it is obtained that:

\[
\hat{p}^{(\lambda)} = \sum_{l=1}^{L} w_l p_l
\]

where \( p_l \) are the fingerprint cells center. Superscript is used to indicate that the positions estimated were obtained by using a particular technology. The target position \( \hat{p} \) is calculated as the weighted sum of the estimations for every cell using the Kernel function. The weights \( w \) are determined by:

\[
w_l = \frac{K\left(\frac{\text{RSS}_l, \lambda - \mathbb{E}[F(p_l, \lambda, \text{RSS})]}{h_N}\right)}{\sum_{j=1}^{L} K\left(\frac{\text{RSS}_j, \lambda - \mathbb{E}[F(p_j, \lambda, \text{RSS})]}{h_N}\right)}
\]

The numerator contains the current Kernel for cell \( l \in c \) and the denominator collects the sum of all Kernel values for every cell. The Nadaraya-Watson Kernel regression is the MMSE estimator of \( p \). The TPE output is the estimated value \( z = \hat{p} \) which is assumed to be static in \( k \).

The parameter selection for the algorithm to fit the PDFs in equation (5) is carried out using the exponential kernel (6).

\[
K(x) = \frac{1}{2} e^{|x|}
\]

where \(|x|\) represents the euclidean distance. There are several kernels in the literature but the computation complexity is higher. In our work, the last section VII presented that exponential kernel improve the estimation accuracy than using others most common kernels like the Gaussian kernel reducing the algorithm time.

Furthermore, the bandwidth selection is based on [48] where a study of different method are carry out to obtain this parameter. Based on the tables presented in this work the parameter is setting equal to \( h_N = 0.8 \).

IV. TARGET TRACKING PROCESS (TTP)

Tracking stage is intended to minimize the effect of noise in \( p \) by filtering outliers giving as a result the peaks-free object route. In order to maintain notation in a simple manner, superscript used in equation (4) has been removed since the tracking system is described for a single technology. Several methods have been proposed for this purpose, however in
modeled is unknown. Predict the state of a system, even if the nature of the system is unknown. KF consists of two stages: (a) time update (prediction) and (b) measurement update (correction). The former is in charge of projecting future estimators of the current state and error covariance. The latter will predict new state estimations.

One of the most interesting KF features is the ability to predict the state of a system using information from the previous estimations. KF assumes the system to be described by a linear stochastic model, where the error associated to the system as well as the additional information incorporated are presented as normal distribution variables with zero mean and variance $\sigma^2$.

A complete illustration of Kalman Filter is drawn in figure 4. This filter performs the Best Linear Unbiased Estimator (B.L.U.E) of a system state using information from the current and past measurements.

In this paper, a Modified version of the Kalman Filter (MKF) [31] is applied. The reason to use MKF is that the model considered in this system is linear. Consequently, a linear version of KF taking into consideration velocity and acceleration parameters represents a good trade-off between complexity (i.e. Extended Kalman Filter) and performance (velocity and acceleration improve the estimation results). In addition, Extended Kalman Filter does not provide the optimal solution in case of wrong initial state inputs, which is highly probable in the scenarios considered in this work [44].

Therefore, MKF algorithm considers the speed axis $v_x$ and $v_y$ variables for the state $x$ in addition to the position vector $\hat{\mathbf{p}} \rightarrow (\hat{x}, \hat{y})$ of the tracked target. These variables in combination with the use of a more realistic movement model known as Random Walk Model [49] represent an advantage in terms of estimation accuracy.

Finally, matrix $\mathbf{H}$ relates the state with the measure $z_k$ where $z_k$ is the TPE output and $\eta \sim N(0, \sigma)$ the measurement noise respectively.

In this paper, a Modified version of the Kalman Filter (MKF) [31] is applied. The reason to use MKF is that the model considered in this system is linear. Consequently, a linear version of KF taking into consideration velocity and acceleration parameters represents a good trade-off between complexity (i.e. Extended Kalman Filter) and performance (velocity and acceleration improve the estimation results). In addition, Extended Kalman Filter does not provide the optimal solution in case of wrong initial state inputs, which is highly probable in the scenarios considered in this work [44].

Therefore, MKF algorithm considers the speed axis $v_x$ and $v_y$ variables for the state $x$ in addition to the position vector $\hat{\mathbf{p}} \rightarrow (\hat{x}, \hat{y})$ of the tracked target. These variables in combination with the use of a more realistic movement model known as Random Walk Model [49] represent an advantage in terms of estimation accuracy.

where $A$ matrix relates the $k$ state with the $k+1$ time state. Furthermore, $B$ is the control input associated to $u_k$, which is the system input. Furthermore, $P$ matrix is the a posteriori error covariance matrix from Kalman process that collects the deviation in the estimations and $Q$ is the covariance matrix of the process noise.

Moreover, Measurement update (correction) stage in figure 4 is described as follows:

(1): The matrix $G$, called Kalman Gain or blend factor, adjusts the trade-off between the estimation and the error measurement.

(2): Equation to calculate the posterior state $x_{k+1}$ as a linear combination of the a priori $x_k$ estimator and the weighted difference between current observations $z_k$ and a measurement prediction $\hat{z}_k$.

(3): Update the error covariance based on the Kalman Gain. Moreover, the observation $z \in \mathbb{R}^2$ contains the spatial coordinates of the position $x$.

$$
\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \eta_k \quad (8)
$$
Therefore, the system can be expressed as follows:

\[
A_k = \begin{bmatrix}
1 & 0 & dt & 0 \\
0 & 1 & 0 & dt \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
\end{bmatrix}
\]

(9)

where the states \(x_k\) are composed by axis positions and speed:

\[
x_k = [x, y, v_x, v_y]'
\]

(10)

Additionally, \(H\) remains as the general Kalman formulation whereas \(B_k\) will be modified as follows:

\[
H = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
\end{bmatrix}, B_k = \begin{bmatrix}
\frac{1}{2} dt^2 \\
\frac{1}{2} dt^2 \\
0 \\
0 \\
\end{bmatrix}
\]

(11)

Main modifications presenting the movement information and are included in different matrices of the process, where \(a_k \equiv \sigma_d^2\) is the acceleration parameter. This parameter is set to 0.1 based on the normal people velocity when they are walking. Moreover, \(R\) matrix contains the error variance of TPE estimation:

\[
R = \begin{bmatrix}
\sigma_x^2 & 0 \\
0 & \sigma_y^2 \\
\end{bmatrix}
\]

(12)

where \(\sigma_x, \sigma_y\) represent the error deviation in TPE estimation in \(x\) and \(y\) respectively and are settled to the TPE mean error for each technology used. The values of \(R\) were obtained by performing several independent experiments for each of the technologies considered in this work. Note that consequently, the Kalman Filter has its own \(R\) Estimation Error Covariance Matrix corresponding to IEEE802.15.1, IEEE802.15.4 and IEEE802.11g standards respectively.

Finally, according to the Random Walk Model, the covariance matrix \(Q\):

\[
Q = \begin{bmatrix}
\frac{1}{4} dt^4 & 0 & \frac{1}{4} dt^3 & 0 \\
0 & \frac{1}{4} dt^4 & 0 & \frac{1}{4} dt^3 \\
\frac{1}{2} dt^3 & 0 & dt^2 & 0 \\
0 & \frac{1}{2} dt^3 & 0 & dt^2 \\
\end{bmatrix} \sigma_q^2
\]

(13)

where \(\sigma_q\) parameter must be fixed. For people acceleration in a normal walking behavior, it is equal to 1. This value has been chosen according to the behavior estimation proposed in [29], [30].

Finally, for the sake of simplicity, it is assumed that the outputs of the KF are denoted using the same notation than previous sections. Therefore, the outputs are \(\hat{x}_k, \hat{y}_k\) for estimated positions and \(P_k\) for error covariance.

V. MULTI-SENSOR PROCESSING ARCHITECTURES

In this paper, mainly two multi-sensor schemes are considered: (1) Track-To-Track Fusion (TTTF) and (2) Kalman Sensor Group Fusion Architecture (KSGF). Both architecture patterns are shown in figure 3. In the former, processing stages (TPE and TTP) are performed separately for each sensor to achieve high-level inferences (routes) that are subsequently fused. Nonetheless, in the latter scheme the sensor measurements are combined in the TTP stage. Both schemes are described and compared in next subsections and results are presented in section VII.

A. Track-To-Track Fusion Architecture TTTF

This architecture pattern relies on fusing the routes obtained by each technology \(\lambda\) separately. The proposed algorithm applies the Maximum Likelihood Estimator (MLE) to the probability density function of the available routes [32]. For this purpose, \(\mathcal{R}\) is defined to be the set of routes. Additionally, it is assumed that every route is composed by \(n\) number of steps \(\mathcal{R} = \{\mathbf{p}_1, \ldots, \mathbf{p}_{\mathcal{R}}\}\). Notice that the algorithm is able to fuse information even if there is not measurements from all technologies available in the time window considered. Therefore, the probability density function can be described as follows:

\[
p(\mathcal{R}^\lambda) = \prod_{k=1}^{\mathcal{R}} \prod_{\lambda=1}^{\mathcal{A}} \frac{1}{\sqrt{2\pi}^D} e^{-\frac{1}{2}(\mathbf{p}_k - \mathbf{p}_k^\lambda)'\mathbf{P}_{\lambda,k}^{-1}(\mathbf{p}_k - \mathbf{p}_k^\lambda)}
\]

(14)

where \(\mathbf{P}_{\lambda,k}\) and \(\mathbf{p}_k\) are the outputs from TTP described in section IV. Note that \(\mathbf{p}\) describes the position estimated from previous stages (TPE and TTP), whereas \(\mathbf{p}\) denotes the final output of the Fusion stage. Moreover, \(D\) is the range of \(\mathbf{P}\), which denotes the number of sensors available in a particular window-time in the network. From equation (14), it has been demonstrated [45] that MLE can be obtained by minimizing the result of \(\mathbf{p}\) and setting the gradient equal to 0. As a result, the state estimation for every location is calculated as follows:

\[
\mathbf{P}_k = \left( \sum_{\lambda=1}^{\mathcal{A}} \mathbf{P}_{\lambda,k}^{-1} \right)^{-1}
\]

(15)

Note that, both the estimation \(\mathbf{P}_k\) and \(\mathbb{E}(\mathbf{p}_k) = \mathbf{P}_k\) are unbiased. Finally, the covariance of the estimate, denoted as \(\Sigma_k\), is given by:

\[
\Sigma_k = \left( \sum_{\lambda=1}^{\mathcal{A}} \mathbf{P}_{\lambda,k}^{-1} \right)^{-1}
\]

(16)

It can be shown that this result is the Kalman Filter update equation applied to each local route in each partition. Further details can be found in [45].

B. Kalman Sensor Group Fusion Architecture KSGF

The second architecture presented is the Kalman Sensor Group Fusion (KSGF) which exploits the Kalman filter properties for multi-sensor data fusion [33]. KSGF method directly incorporates the position estimations of each sensor in a single Kalman Filter. However, the system must be properly conditioned to obtain an accurate route estimation.

General KF system equations were described in section IV. However, some modifications must be introduced in the filter to support multi-sensor functionalities. In concrete, update matrix \(H\), described in equation (11), must be adapted to incorporate the sensor inputs:
Algorithm 2 MUFAF: TPE, TTP, TTTF Stages

1: procedure TPE:
   Inputs: RSSf, λ \rightarrow synchronized RSSI measurements for a window time k. // F \rightarrow Fingerprint Matrix.
   for each \lambda do
     2: \mathbf{w}_j = \sum_{j=1}^{L} K \left( \frac{\text{RSS}f, \lambda - E[F(p_j, \lambda, \text{RSS})]}{h_N} \right)
     3: \mathbf{w}_t = \frac{\mathbf{w}_j}{\mathbf{w}_j} \quad \text{(4)}
     4: \mathbf{p}^\lambda = \sum_{t=1}^{L} \mathbf{w}_t \mathbf{p}_t \quad \text{(4)}
   end for
   end procedure

2: procedure TTP:
   Inputs: \hat{\mathbf{p}}^\lambda \rightarrow estimated positions for every technology \lambda
   for each \lambda do
     5: \hat{x}_{k+1}, \hat{\mathbf{p}}_{k+1} \rightarrow Kalman Time Update Fig 4
     6: \hat{x}_k, \hat{\mathbf{p}}_k \rightarrow Kalman Measurement Update Fig 4
     7: \hat{\mathbf{p}}^\lambda \leftarrow \hat{x}_k
     8: \hat{\mathbf{p}}_{\lambda,k} \leftarrow \hat{\mathbf{p}}_k
   end for
   end procedure

3: procedure TTTF:
   Inputs: \hat{\mathbf{p}}^\lambda \rightarrow Estimated position from TTP for every technology. // \mathbf{P}^\lambda.k Error Covariance Matrix from Kalman Process.
   for each \lambda do
     9: \hat{x}_{k+1}, \hat{\mathbf{p}}_{k+1} \rightarrow Kalman Time Update Fig 4
     10: \hat{x}_k, \hat{\mathbf{p}}_k \rightarrow Kalman Measurement Update Fig 4
     11: \hat{\mathbf{p}}^\lambda \leftarrow \hat{x}_k
     12: \hat{\mathbf{p}}_{\lambda,k} \leftarrow \hat{\mathbf{p}}_k
   end for
   end procedure

\mathbf{R} = \begin{bmatrix}
\sigma_{x_1}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \sigma_{y_1}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \sigma_{x_2}^2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \sigma_{y_2}^2 & 0 & 0 & 0 & 0 \\
. & . & . & . & . & . & . & . \\
0 & 0 & 0 & 0 & 0 & 0 & \sigma_{x_\lambda}^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{y_\lambda}^2 \\
\end{bmatrix} \quad \text{(18)}

\text{where } \sigma \text{ parameters are the same that have been used in the TTP process for each independent technology.}

Finally, the observations vector \mathbf{z} is now extended to collect the x and y coordinates estimated from each sensor:

\mathbf{z} = \begin{bmatrix}
\hat{x}_1, \hat{y}_1, \hat{x}_2, \hat{y}_2, \ldots, \hat{x}_\lambda, \hat{y}_\lambda
\end{bmatrix}^T \quad \text{(19)}

Taking into consideration the aforementioned modifications, the KF can be performed as described in TTP. As a result, the fused route will be provided by the KSGF. Both architecture patterns can be employed for multi-sensor data fusion. The computational complexity of the architectures proposed has been addressed \cite{34} and detailed in section VII. In this work, both patterns are compared in terms of execution time. KSGF computing time grows exponentially as the number of sensors increases.

C. Alpha-Beta Filter for noise reduction

Fusion stage improves the quality of the estimated route in general terms as it takes the information from all sensor technologies. However, due to the high values in covariance matrices employed for Multi-Sensor fusion stage, some outliers can arise or propagate. Additionally, as the Multi-Sensor Fusion can calculate routes with partial information (even a single technology with a high error rate), the outliers can appear or introduce noise in future estimations. These outliers corrupt the normal target path which can yield to inappropriate routes modeling or wrong patterns generation among others.

In order to minimize the noise and outliers effect that can arise as result of multi-sensor stage, an additional filtering stage is applied. There exist several proposals for routes smoothing, however in this paper, the well known Alpha-Beta filter \cite{35,36} is used. The reasons to employ the Alpha-Beta filter are the lower complexity, the lesser computational cost and that system model details are not required. Furthermore, taking into consideration that the effect of outliers is much lesser in the presence of low noisy estimations, it has been shown that the alpha-beta filter estimations performance is similar to Kalman Filters \cite{47}.

Similarly to equation (9), initial states of spatial coordinates and axis speeds are considered. From Fusion stage, the notation for estimated positions is \hat{\mathbf{p}} = (\hat{x}_m, \hat{y}_m). This position is the input of the Alpha-Beta filter. Therefore, the position prediction can be obtained as follows:

\hat{x}_{k+1} = \hat{x}_k + \hat{v}_x dt \quad \hat{y}_{k+1} = \hat{y}_k + \hat{v}_y dt \quad \text{(20)}

Additionally, error measurements can be calculated from predicted states by the subtraction of the fusion algorithm \hat{x}_m, \hat{y}_m from the predicted position:
Algorithm 3 MUFAF: Adaptive Fingerprinting Update (AFU)

Inputs: \( \tilde{p} \rightarrow \) Position estimated after fusion stage. // \( F \rightarrow \) fingerprint Matrix. // \( \text{RSS}_{i,\lambda} \rightarrow \) RSSI measurements gathered for every node and technology.

1: procedure AFU:
2: for each \( \lambda \) do
3: for each point \( i, j \) in \( c \) do
4: Calculate \( \tilde{g}(d(p_i, p_j)) ; (28) \)
5: end for
6: Fit into theoretical model \( \tilde{g}(\cdot) \)
7: Create \( \Gamma \) Matrix
8: Create \( b, b_{\lambda+1} \leftarrow 1 \)
9: Solve SLE \( w \leftarrow \Gamma^{-1}b ; (33) \)
10: Obtain \( \text{RSS}_p = \sum_{i=1}^{M} q_i E[F(p_i, \lambda, \text{RSS})] ; (32) \)
11: if \( ||\text{RSS}_p - \text{RSS}_{i,\lambda}||_{\mu} \) then
12: Update: \( F(p_{\lambda}, \lambda, \text{RSS}) \leftarrow \text{RSS}_p \)
13: end if
14: end for
15: end procedure

However, an interesting approach is reached by adaptively update the fingerprinting dataset based on the quality of the information gathered. Consequently, a proposal for AFU is presented in this work. This algorithm endows the entire MUFAF framework with responsiveness to indoor condition changes.

AFU can be described as follows: the input of the system is the RSSI values from the target tracked and the outputs from TPE, TTP and Fusion stages are the estimated positions of the mentioned target. AFU aim is to verify the similarity degree of the RSSI gathered with the value estimated at this location using the Fingerprint dataset. There exist a wide range of classical estimators in the literature. However, an interesting approach is given by the use an interpolation technique called Kriging [36], [37].

This technique was initially employed for geospatial analysis, however the method has been successfully applied to several sciences. The main goal of Kriging is to estimate the field behavior at unknown locations based on the available measurements.

In general terms, Kriging is a Best Linear Unbiased Estimator (B.L.U.E) which aims to minimize the mean square error. As aforementioned, if \( z \) denotes the observation:

\[
\minimize |\hat{z}(\hat{p}) - z(\hat{p})| \tag{26}
\]

The main feature (and differential) of Kriging interpolation is that guarantees to be unbiased even if stationary of field is not known. This can be expressed as:

\[
E[\hat{z}(\hat{p}) - z(\hat{p})] = 0 \tag{27}
\]

Statement (27) represents the main advantage over traditional methods based on covariance. This condition is satisfied due to the use of a statistical tool called semivariance. Its
general estimator is given by the Experimental Variogram, denoted by $\hat{\gamma}(\cdot)$:

$$\hat{\gamma}(h) = \frac{1}{2N(h)} \sum_{i,j=1}^{N(h)} (z(p_i) - z(p_j))^2$$  \hspace{1cm} (28)

where $N(h)$ represents the number of RSSI measurements gathered at a predefined distance $h$.

The Semivariance is employed to exploit the spatial similarity without dependency of the field mean. Additionally, square difference (28) endows the estimator with robustness to field outliers.

Specifically, in MUFAF all fingerprint cells $c$ are employed to build the Semivariogram:

$$\hat{\gamma}_c(h) = \frac{1}{2||c(h)||} \sum_{c(h)} (RSS_{p,\lambda} - RSS_{p_2,\lambda})^2$$  \hspace{1cm} (29)

$\forall i, j \in c \land d(x_i, x_j) \in h$. This process is performed in the case of square/rectangular grids as the ones considered in figure 2, the lags distribution can be expressed as:

$$d(x_i, y_j) = \sqrt{\left(\frac{d_1}{2} x_i - \frac{d_2}{2} y_j\right)^2 + \left(\frac{d_1}{2} y_i - \frac{d_2}{2} y_j\right)^2}$$  \hspace{1cm} (30)

where $x_i, y_j \in c$ are the spatial coordinates of the cells center. Furthermore, let assume that $W$ and $D$ are the indoor width and depth expressed in centimeters respectively. Additionally, $L = n \times m$ is the total number of cells and $n$ and $m$ are the number of cells per side (width and depth). Finally, $d_1$, $d_2$ denote the distance from the center of a cell to the center of a neighbor cell in every axis. Consequently, it has been shown [46] that if the number of intervals within lags in $h$ satisfies to be lesser than $\sqrt{\frac{\pi \sqrt{W^2 + D^2}}{L}}$, then it will exist at least a Semivariance measurement in every $h$ interval.

There exist several theoretical models for the Semivariogram such as Linear, Spherical, Gaussian and Exponential [38]. Gaussian model is appropriate for processes with small short-term variations of similarity whereas Spherical or Exponential are the most suitable options for applications where considerable variations arise in short-distance. Due to the high variance in measurements distribution presented in section II, in this paper, Spherical Semivariogram theoretical model is employed:

$$\gamma(h) = ng + Rg \left(\frac{3h}{2S_i} - \frac{h^3}{2S_i^3}\right)$$  \hspace{1cm} (31)

where $ng$ is called the nugget effect. This parameter considers the uncertainty of the Semivariogram estimation when the separation between points tends to 0. $Rg$ is the Range, that is the maximum distance taken into consideration for two samples. Finally, the Sill ($S_i$) is the maximum variance (dissimilarity) between two samples. These parameters are drawn in figure 5.

As aforementioned, Kriging provides the (B.L.U.E). Therefore, the field value $\tilde{RSS}_p$ can be described as a weighted sum of the available node measurements:

$$\tilde{RSS}_p = \sum_{i=1}^{M} w_i \mathbb{E}[\gamma(p_i, \lambda, RSS)]$$  \hspace{1cm} (32)

where $w = \{w_1, \ldots, w_M\}$ are the weights which sum must fulfill $\sum w = 1$ to guarantee unbiasedness. Furthermore, the weights are obtained by solving the following System of Linear Equations (SLE):

$$\Gamma s = b$$  \hspace{1cm} (33)

where:

$$\Gamma = \begin{bmatrix} \gamma(d(p_i, p_j)), & i, j = 1, \ldots, M \\ 1, & i = M + 1, j = 1, \ldots, M \\ 0, & i = M + 1, j = M + 1 \end{bmatrix}$$

is an $M + 1 \times M + 1$ Matrix capturing the spatial similarity among the available nodes in the fingerprint cell distribution. Additionally:

$$s = [w, \mathcal{L}]$$

$s$ vector contains the weights $w$ and the $M + 1$ value is the Lagrange Multiplier $\mathcal{L}$ that ensures the unbiased nature of the estimator and

$$b = \begin{bmatrix} \gamma(d(p_1, p_1)), \gamma(d(p_1, p_2)), \ldots, \gamma(d(p_1, p_M)), 1 \end{bmatrix}$$

$b$ vector contains set of theoretical semivariances from the nodes to the estimated point $p$.

Finally, by solving the aforementioned System of Linear Equations SLE (33), the RSSI estimation is obtained. However, to determine the fingerprint update, the following decision rule has been implemented:

$$||\tilde{RSS}_p - RSS_i,\lambda|| \leq \mu$$  \hspace{1cm} (34)
where $\mu$ is an empirical threshold which value must be iteratively calculated. The process consists on: (a) perform a significant number of randomly generated routes (i.e for the experiments carried out 1000 routes were generated). (b) For these routes, the decision threshold must be initially adjusted to a low value (i.e $\mu = 10^{-1}$) which implies a low update rate as most of the measurements will be neglected in the fingerprint dataset. (c) Calculate the mean error of the routes with this threshold, (d) Iteratively increase the $\mu$ value until a large number (i.e the high boundary in experiments presented in this work was settled to 5) which means that all measurements will be accepted to update the fingerprint dataset and repeat the mean error calculation, (e) Find the $\mu$ value that minimizes the mean error function.

The general outline for Kriging implementation is described in algorithm 3. In this algorithm, the sequential steps to obtain the RSSI estimation and compare it with data gathered (RSSI from sensors) in order to make a decision on the Fingerprint update stage.

In addition to the theoretical model choice for the Semivariogram, there are some parameters that can affect the proper functionality of the Kriging interpolation. A critical parameter is the decision threshold $\mu$. The process aforementioned needs to be perform with no prior knowledge of the network behavior. However, the better the process is carried out the higher the precision and responsiveness of MUFAF algorithm will be. Moreover, the number of lags ($h$) is relevant to attain an appropriate fitting. Furthermore, the number of neighbors is important to reach the best trade-off between complexity and accuracy. In next section, experiments executed as well as the results obtained will allow a better understanding of the process described in this section as well as the improvement of the results compared with classical estimation techniques.

VII. EXPERIMENTAL RESULTS

A. Experiment Setup

In this section, the experiments performed to assess MUFAF and the results obtained in comparison with existing methods are detailed. In the simulations, the laboratory scenario drawn in figure 2 is considered. This room represents a typical indoor environment with human activity, people flow controlled and harsh propagation conditions due to multipath fading, obstacles due to furnitures and wireless interferences due to electronic devices, further details can be found in [18].

Additionally, as described in section II, four $v = v_1, \ldots, v_4$ nodes are deployed at the lab corners. Furthermore, every node is equipped with four sensors $\Lambda = 4$: IEEE802.15.1 (Bluetoot), IEEE802.11.g (WiFi), and two IEEE802.15.4 receivers (XBee and CC2420). Firstly, TPE and TPP performance are assessed by comparing MUFAF to other methods that employ fingerprint technique. Afterward, both pattern architectures are contrasted with covariance-based methods. Moreover, a complete framework experiment is carried out for multi-person tracking. Several aspects such as the number of steps per route, the number of neighbors available and the time-evolution of AFU
were assessed in the results presented.

B. Computational Cost

The entire process for MUFAF implementation can be stated as:

- Create Fingerprint dataset.
- Performing TPE.
- Performing TPP.
- Multi-Sensor Fusion.
- Alpha-Beta Filter.
- Performing AFU.

Taking into account fingerprinting stage, for every technology (A) considered, the F matrix must be created. Furthermore, Kernel Density Estimation computational complexity has been proven to be $O(NL)$ \cite{54}. Therefore, the cost of executing Fingerprinting and TPE is given by $cost_{TPE} = \lambda[O(L) + O(LN)]$.

Moreover, depending on the architecture pattern choice, the order of the aforementioned MUFAF tasks can change as well as the associated computational cost. In case of TTTT, the computational complexity of solving the separate Kalman Filter for every technology is $\lambda[O(x^2) + O(p^2)]$ where $x$ and $p$ lengths are 4 and 2 respectively. In case of KFSG, the cost can be expressed as $O((Ax)^2)$. Additionally, $\alpha - \beta$ filter perform few linear operations: $O(x)$. Finally, AFU involves the solution of a System of Linear Equations of size $M + 1$. Consequently, AFU computational complexity is $O((M+1)^2)$.

C. Experiment Results

The first experiment was run to assess TPE stage. It was executed for a total of 1000 different routes across the room such as the drawn in figure 2. Initially, every route $\Omega_A = \{x_1, \ldots, x_{step-90}\}$ was composed of 90 steps. A fingerprint dataset has been created with the cells distribution shown in the scenario figure 2. Measurements were carried out with the available sensors reading simultaneously for a period $T = 5$ minutes per cell. The number of samples vary according to the technology and distance to the network nodes as previously described in section II. Specifically, for the experiments performed, 28 cells were figured therefore the fingerprint process lasted $28 \times 5 = 140$ minutes.

The average error for every method is detailed in Table II. MUFAF results are significantly better than deterministic methods. In addition, MUFAF results are better than Gaussian Kernel methods except for IEEE 802.11.g WiFi sensors. This is probably due to two aspects: (a) distances considered are not long enough to appreciate IEEE 802.11.g WiFi signal
attenuation and (b) the histogram distributions are similar to Gaussian functions.

Furthermore, results obtained for the aforementioned sensor technologies are shown in figure 6. In this figure, it can be observed that IEEE802.15.1 Bluetooth technology errors in estimation are larger than the other technologies. The reasons for this result can be the low power transmission of IEEE802.15.1 Bluetooth devices and interferences. Conversely, 802.15.4 based technologies perform with a mean error under 70 cm.

Moreover, the architecture patterns described in section V were also analyzed. In figure 7, a particular experiment is provided showing the error between the real path and the estimations obtained using the proposed architectures. In the left side, the error variance is large in the initial steps, but it can be observed how this variance is reduced along time due to convergence of P matrix in KF.

Conversely to expected results, TTTF outperforms KSGF. This can be due to the fact that TTTF employs as much Kalman Filters as technologies (λ) available. Therefore, the dimension (range of matrices in KF) is lower than the single one of KSGF. Nonetheless, for a longer assessment period the accuracy of estimations is not significantly enhanced. This is shown in figure 8, where it can be appreciated that the initial TTTF error is smaller than KSGF, but after some steps in the iteration (≥ 100 steps) it is stabilized. On the other hand, KSGF error is lesser in the long term however, both patterns converge to an error around 45 cm.

Similarly, Adaptive Fingerprinting Update (AFU) was evaluated. The first experiment consisted in comparing Kriging interpolation with Covariance based methods. In the subsequent experiments, the number of steps for each route is 90. The results are shown in figure 10. In the left side of the figure, MUFAF is tested by using the inverse covariance method [50] whereas the right side of the figure illustrates MUFAF using Kriging interpolation. It can be observed that the method proposed in this paper converges faster to a minimum error value. On the one hand, blue line shows the fitting model that minimizes the minimum square error (MSE) of estimations for TTTF. On the other hand, red dotted line describes the model that minimizes MSE for KSGF. Results show that MUFAF mitigates the impact of changing conditions. Additionally, the accuracy after some hundred steps is at least between 10 − 15 cm better.

In order to evaluate the evolution of the distribution, an experiment consisting in an initial Fingerprinting containing only 50 measurements per cell and technology is performed. Firstly, in figure 9 the evolution of histograms through AFU is illustrated. In top figures, the initial samples distribution is shown while in bottom figures the evolution of distribution after 1000 samples is displayed.

However, one of the most relevant aspects for AFU is the threshold definition. An experiment has been carried out to obtain the performance of AFU applying several values to μ. The results are depicted in figure 11, where the average error for 1000 routes with fixed μ values show that the optimal value
hands a device equipped with the technologies mentioned in number of cells, the interpolation relevant are not attained. {4, 8} yields to the lower error. For large range from $M = \ldots$ the number of neighbors chosen for interpolation in 1000 routes. Fig. 12. Mean error expressed in centimeters for Kriging algorithm varying according to the number of active users. 

Additionally, the unexpected multipath, reflection and related events make the variance of fingerprinting dataset to increase. Furthermore, the employment of advance filtering techniques such as Particle Filters can help to increase the efficiency and precision of estimations performed. Finally, in general terms, technologies employed in this work can be fused to improve the accuracy of estimations.

Finally, an experiment with several persons randomly walking across the room was performed. Every person held in their hands a device equipped with the technologies mentioned in this work. The fingerprint dataset was initially composed of 50 samples per cell. The results of this test are drawn in table III. Surprisingly, for a single person, the estimation accuracy is lower than multi-person cases. This is mainly due to the faster fingerprint update when the measurements gathered is doubled. However, results for 3 and 4 persons walking simultaneously is similar. The decision threshold $\mu$ allows update the fingerprinting dataset only for high correlated estimations. Additionally, the unexpected multipath, reflection and related events make the variance of fingerprinting dataset to increase according to the number of active users.

The entire dataset containing fingerprinting measurements for every technology, as well as example routes are totally available for testing purposes and comparison with the algorithms described here. This information can be download from http://www.gatv.ssr.upm.es/~ghp/

VIII. CONCLUSIONS AND FUTURE WORK
In this paper, a complete framework for accurate indoor tracking has been described. MUAF is composed by several algorithms that allow to adaptively track an object in an indoor scenario. Furthermore, MUAF takes advantage of the multiple information sources by means of a fusion strategy. Finally, AFU guarantees the most updated information in the fingerprint data. As an evidence of results obtained, it can be concluded that the fingerprint technique in combination with probabilistic techniques such as Kernels outperform the results obtained by employing deterministic techniques.

Moreover, in the view of results obtained, it can be concluded that the Track-to-Track Fusion strategy outperforms Kalman Sensor Group Fusion for short estimation periods. However, for mid-term applications KSGF can be the most appropriate choice. Up to the best of our knowledge, Kriging interpolation has not been ever used for Adaptive Fusion Update and in this work has been demonstrated that this technique can be successfully employed. The potential application of this technique for statistical estimation such as the KDE has been proven.

As next steps in this Framework development, it is important to highlight the need of improving the process to estimate the $\mu$ parameter, or even to develop techniques that allow to endow it with adaptability to the environment conditions. Furthermore, the employment of advance filtering techniques such as Particle Filters can help to increase the efficiency and precision of estimations performed. Finally, in general terms, technologies employed in this work can be fused to improve the accuracy of estimations.

As future work, this algorithm can be used in combination with computer vision techniques to unequivocally identify people in an indoor environment. Person identification for trackers in Computer Vision is a major issue that can be faced by fusing images information with data gathered from the technologies presented in this work. Additionally, the inclusion of inertial sensor can significantly improve the estimations using absolute orientation values. Moreover, AFU can be analyzed by applying Kriging variants such as Universal Kriging that can be applied to non-stationary fields.

ACKNOWLEDGMENT
This work was supported by the European Project: ICT4Life http://ict4life.eu/ Grant no. 600090 within the H2020 Research and Innovation Programme.

REFERENCES


[34] Shau-Shun Jan, Shuo-Ju Yeh, Ya-Wen Liu, “Received Signal Strength Database Interpolation by Kriging for a Wi-Fi Indoor Positioning System,” Sensors 2015, 15(9), 21377-21393.


