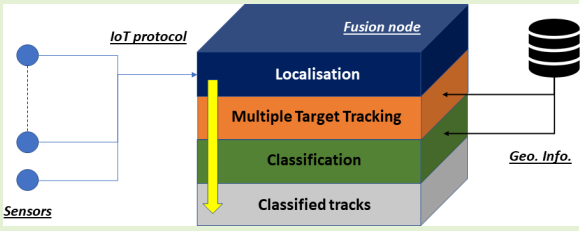
Wireless Sensor Network for Tactical Situation Assessment

Benjamin Pannetier, Jean Dezert, Julien Moras, and Raphaël Levy

***Abstract*— This paper describes a complete solution of a new dropped wireless sensor network (called "abandoned" since it is never picked up) dedicated to intelligence operation. The sensor network, named SEXTANT for Smart sEnsor X for Tactical situation AssessmeNT and presented in this paper is the achievement of several years of research and development in the fields of data fusion and smart sensors at the French Aerospace Lab. The main contribution of this paper is the presentation of experimental results obtained with our Joint Tracking and Classification (JTC) algorithm. The originality of this algorithm relies on the use contextual information in the target tracking processing and data fusion for target classification processing.**



***Index Terms*— wireless sensor network, mbed protocol, data fusion, multiple target tracking, classification.**

I. Introduction

THIS paper presents a new Wireless Sensor Network (WSN) system called SEXTANT (Smart sEnsor [X] for Tactical situAtion assessmeNT) for real-time tactical assessment in military applications. The proposed SEXTANT WSN is aimed at culminating in the next years into an operational network of a large number of smart heterogeneous sensors with onboard sensing, processing and wireless communication capabilities. In fact, SEXTANT is currently a modular laboratory platform which can integrate new sensor components (with dedicated signal processing tasks) and can evaluate their impact in the fusion process. This paper presents this achievement of the SEXTANT study and development and is focused on experimental results demonstrating our Joint Tracking and Classification (JTC) algorithm. The WSN technical requirements are described in [1].

The current state of our demonstrator allows studying the automatic data processing for fusion detections and creating tracks on moving targets. In the future, it will enable the evaluation of schemes for data collection and fusion process, as well as the demonstration of the necessity of taking into account the contextual information in the fusion process (*cf.* [2], [3]), typically the geographical information such as traffic lanes, road intersections or intervisibility areas. [1], [4], [5]. Previous works on this topic use particle filter to generate particles on the road in order to improve the ground target state estimation. This is the case for tracking targets with a UAV in [6] or for instance in [7] where the authors developed a modified particle filter (PF) to track a single vehicle by a WSN. However, in order to address long-running operational surveillance missions with a WSN, algorithms must combine low computational cost and good target tracking performances. To this aim, our JTC algorithm exploits geographical information.

In this work, we propose an adapted version of the algorithm described in [8] for our WSN tracking SEXTANT demonstrator. In line with what was proposed by Ulmke and Koch [9], we leverage the geographical information within both the Multiple Target Tracking (MTT) algorithm and the classification algorithm, for improved performances. Indeed, the target classification method is based on the knowledge of target evolution capability relatively to the type of terrain. Recent paper [10], [11] uses a constrained GMPHD for the MTT, but if this approach is efficient with a single sensor it is not adapted for multiple sensors. Also, we adapt the Multiple Hypotheses Tracker algorithm described in [8] to track several ground and aerial targets. In [12], Blasch presents a system based on coupled target tracking and identification with intent modeling. The tracking performances are improved thanks to a classifier that works in parallel of the tracker. In the same line of thought, we presented in [13] a fuzzy inference-based target classifier that uses kinematic data jointly with the contextual traffic information to identify ground and aerial targets at once. Although our previous work was sufficient as a proof of concept, high performance classification could not be achieved due to the lack of precision in inferring the geographical context on which a target is located.

The paper is organized as follows: in Section II the WSN is briefly introduced. Section III presents the improvement brought to the proposed tracker and classifier compared to previous versions. Section V describes the algorithm for MTT.

Benjamin Pannetier is with CSGroup, 22 Av. Galilée, 92350 Le Plessis-Robinson, France. Jean Dezert, Julien Moras and Raphael Levy are with ONERA, the French Aerospace Lab, Palaiseau Cedex, France. Corresponding authors emails: benjamin.pannetier@csgroup.eu, and jean.dezert@onera.fr.

Experimental results obtained on real data are given in Section VI. Finally, Section VII concludes with some perspectives for future work.

II. Sensor network description

The sensor network is composed of heterogeneous sensor nodes. Each node is composed of sensors, a processor and a battery for power supply. Five types of sensor nodes have been used in this work. Three of them are Wake-Up Nodes (WUN), which perform a permanent monitoring in order to detect the presence of a target and estimate its rough position. The two other types of node are principal nodes (PRIN), which are in sleeping mode most of the time and are woken up by a WUN node. The PRINs are dedicated to precise detection and classification of the targets under surveillance. Several sensor networks exploiting data fusion to track and classify targets exist, such as the one proposed in [14] and recently the SAPIENT[1] system which uses heterogeneous sensors [15]. Our approach is however quite different as we must limit the data fusion processing time in order to save energy of the batteries of sensor and computing nodes.

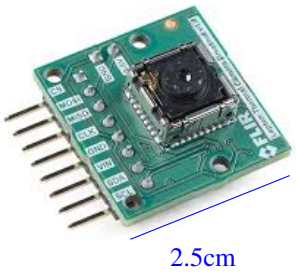
A. Wake-up node (WUN)

A Wake-Up Node (WUN) performs a permanent monitoring in order to detect the presence of a target and estimates its rough position. As the WUNs are continuously on, their power consumption must be low to increase their working life duration. Nevertheless, their refreshing rate must be high enough, while they should provide trustful data that require minimal computation cost. An additional requirement is to be able to include news sensors in the WSN at a relatively low cost. All these constraints led us to implement these nodes based on ARM microcontroller. We chose a set of boards produced by NPX (Freedom Board) running the MBED-OS developed by ARM. The communication between the nodes is realized using 6LowPan[2]. This communication stack allows a deployment of a low power meshed network.

1) Passive InfraRed Node: This type of node is equipped with a common Passive InfraRed (PIR) sensor (see Figure 1a). In order to get more information, we used a ZMotion sensor, which is a double PIR sensor with interleaving beams that enables the detection of the target together with its motion direction (to the left or to the right). The PIR node uses a FRDM-KW41Z board[3] equipped with a System-on-Ship (SoC) that embeds an ARM Cortex-M4 and a 802.15.4 modem. This node is also equipped with environments sensor (barometer, thermometer, luxmeter, hydrometer) and with localization sensor (GPS, 3 axes magnetometer, 3 axes accelerometer).

2) Microbolometer Node: This node uses a microbolometer camera to improve detection of people and vehicles. The sensor used is FLIR Lepton at 9Hz (see Figure 1b). Its associated WUN is shown in Figure 1a by replacing PIR sensor by lepton camera. The lepton camera provides a 60×80 pixel image encoded over 14 bits. The detection process consists in background subtraction and a clustering based on connecting component labelling that provides a set of bounding boxes [16]. The output is the direction of the detection in the sensor frame of reference and a class based on the ratio between the width and height of the box.

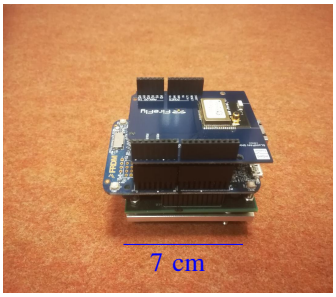
(a) ZMotion Wake-Up node.



(b) Lepton camera.

Fig. 1: Wake up node (WUN).

3) Seismometer Node: The seismometer node developed at ONERA is presented in detail in [17]. It includes a high resolution MEMS accelerometer, a microcontroller measurement and processing unit and a radio module. A high precision MEMS accelerometer has been specifically designed for that purpose. It consists in a quartz vibrating beam accelerometer [18] that enables to reach a resolution under 4.10^{-4} m.s^{-2} in a 100Hz bandwidth, allowing sensing seismic waves emitted by footsteps and vehicles within a 30m detection range. The goal of the microcontroller unit is, at first, to sense seismic raw data, then to process this high rate data and send a signal to the network only in case of detection. This signal provides a first identification data that enables the distinction between footsteps and vehicles. This node has been tested for various ground types (asphalt road, grass, etc.) and various target types as well (pedestrians, heavy and light vehicles, etc). Its main advantages are its low data rate, low power consumption, ability to work during night and through foliage, with a range detection of 30 m.



(a) Seismometer sensor node.

(b) Video principal Node.

Fig. 2: Principal node (PRIN).

[1]SAPIENT stands for Sensing for Asset Protection with Integrated Electronic Networked Technology.

[2]www.researchgate.net/publication/220785778_The_6LowPAN_architecture

[3]https://os.mbed.com/platforms/FRDM-KW41Z/

B. Principal Node (PRIN)

These nodes are used to detect, localize and classify targets. Since the sensors considered here are more heavy (because of power supply to satisfy computing demand), these nodes are mostly in a sleep mode and are woken up by WUNs when targets are detected in the field of view of the sensors coupled to these nodes.

1) Optronic Node: This node uses an RGB camera to detect targets in the image (see Figure 2b). The camera is calibrated on the node and the position of the node can be estimated (based on the board localization sensors), or registered by an operator. This node is based on an Intel NUC6i5SYH (I5-6260U, 8GB RAM) that runs a Linux based OS. The deep learning object detection, named YOLO[4], is implemented using OpenCL to run on the embedded GPU, reaching 5Hz as frequency. The outputs of the algorithm are bounding boxes labelled with classes. The bounding boxes are converted to corresponding angles and sent to the fusion node. Upon request a thumbnail image of one object can be sent to the operator.

2) Radar Node: This node uses a "PSR-500 perimeter surveillance radar" from EPSI-ITNI (see Figure 3). This radar has 8W of consumption, a size fixed to 370(l) × 150(w) × 53(mm), a weight equals to 1.8kg and emits in C band frequency with FMCW waveforms. This node is based on an Intel NUC6i5SYH (I5-6260U, 8GB RAM) that runs a Linux based OS. A processing tool-chain is boarded on NUC to provide measurements of moving targets. Those measurements are transmitted to a fusion node at 3Hz with limited classification information.

Fig. 3: Radar principal Node.

III. Ground target estimation

In this section, we describe our approach to estimate ground target states by taking into account geographic layers in the tracking process.

[4]arxiv.org/abs/1506.02640

A. Definitions

1) State model: The target state at the current time t_k is defined in the local horizontal plane of the East North Up (ENU) frame by the vector:

$$\mathbf{x}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k]^T \tag{1}$$

where (x_k, y_k) and $(\dot{x}_k, \dot{y}_k)$ denote respectively the target location and velocity in the local horizontal plane at time k and T denotes the transpose operator. The elevation component is omitted because we focus here on a two-dimensional ground MTT problem on Earth surface. In fact, despite the presence of drones the selected sensors in our network are unable to provide elevation data for now. The sensor node s state, for $s = 1, \ldots, N_S$, is denoted $\mathbf{x}_k^s$:

$$\mathbf{x}_k^s = [x_k^s \ \dot{x}_k^s \ y_k^s \ \dot{y}_k^s]^T \tag{2}$$

where N_s is the number of sensor nodes.

2) Measurement model: Each principal sensor deployed on the ground provides both location measurement and classification data if possible.

$$\mathbf{z}_k^{j_s} = h^s(k, \mathbf{x}_k) + \mathbf{b}_k^s \tag{3}$$

where $j_s = 1, \ldots, m_k^s$ is the index of the j_s-th validated measurement of target by the sensor s at time t_k, m_k^s is the number of all validated measurements, h^s is the observation function of the state $\mathbf{x}_k$ from sensor s at current time t_k, and $\mathbf{b}_k^s$ is a zero-mean Gaussian measurement noise with covariance $\mathbf{R}_k^s$. The probability distribution function of the measurement (3) given the state and all previous measurements is denoted by:

$$\Lambda_k^{j_s} = p\{\mathbf{z}_k^{j_s} | \mathbf{x}_k, Z^{k-1,l}\}. \tag{4}$$

where the sub-sequence of measurements $Z^{k-1,l}$ will be defined in Section III-A.4. In the specific WSN surveillance application of this paper, we consider three types of measurements: the range $\rho_k^{j_s}$ and the azimuth $\theta_k^{j_s}$, the azimuth $\theta_k^{j_s}$ only, and the range $\rho_k^{j_s}$ only where:

$$\rho_k^{j_s} = \sqrt{(x_k - x_k^s)^2 + (y_k - y_k^s)^2} \tag{5}$$

and

$$\theta_k^{j_s} = \arctan\left(\frac{y_k - y_k^s}{x_k - x_k^s}\right) \tag{6}$$

where s corresponds to a specific sensor node among N_s sensors. For a radar sensor, the measurements are composed of range and azimuth, for seismic sensor the measurements are composed of range component only, and for optronic sensor the measurements are composed of azimuth component only.

3) Classification information: Each principal sensor is able to provide class information associated to each measurement. The set $\mathbf{C}$ of target classes is defined by:

$$\begin{aligned}\mathbf{C} \triangleq \{&\textit{Unknown}, \textit{vehicle}, \textit{animal}, \textit{PAX}, \\ &\textit{copter}, \textit{drone}, \textit{plane}, \textit{boat}\}\end{aligned} \tag{7}$$

where *PAX* stands for "person". The measurement $\mathbf{z}_k^{j_s}$ defined in (3) is associated to a class $\mathbf{c} \in \mathbf{C}$ with a likelihood vector L_C defined over $\mathbf{C}$.

4) Track definition: For notation convenience, a track $\mathcal{T}$ with unique identifier l at current time t_k is denoted by $\mathcal{T}^{k,l}$. This track is represented by its current estimated state $\hat{\mathbf{x}}_{k|k}$ with its associated covariance $\mathbf{P}_{k|k}$ and previous estimated states so that $\mathcal{T}^{k,l} \triangleq \{(\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k}), \mathcal{T}^{k-1,l}\}$. The measurements sequence $Z^{k,l} \triangleq \{\mathbf{z}_k^{j_s}, Z^{k-1,l}\}$ represents a possible set of measurements generated by the target up to time t_k and associated to the track $\mathcal{T}^{k,l}$. The sequence $Z^{k,l}$ consists in a sub-sequence $Z^{k-1,l}$ of measurements up to time t_{k-1} appended by the j_s-th measurement from sensor s available at time t_k from track $\mathcal{T}^{k,l}$.

B. Track estimation with geographic constraints

1) Prediction: To track ground targets, we use several motion models constraining the target dynamics with contextual prior information on road, buildings, forest and locations. The dynamics of the target evolving on the road are modelled by a first-order plant kinematics. The target state on the road segment r is defined by $\mathbf{x}_k^r$ where the target position (x_k^r, y_k^r) belongs to the road segment r and the corresponding heading $(\dot{x}_k^r, \dot{y}_k^r)$ in its direction. The event "the target is on road segment r" is denoted by $e_k^r = \{\mathbf{x}_k \in r\}$. Given this event e_k^r and according to a motion model $M_k^{i_r}$ for the road segment r, at time k, $i_r \in I$, where $I = \{0, \ldots, N_I\}$ is the set of motion models and the model indexed by $i_r = 0$ is the stop model. The estimation of the target state can be improved by considering the road segment r. For a constant velocity motion model, it follows:

$$\mathbf{x}_k^{i_r} = \mathbf{F}^{i_r}(\Delta_k) \cdot \mathbf{x}_{k-1}^r + \mathbf{\Gamma}(\Delta_k) \cdot \mathbf{v}_k^{i_r} \tag{8}$$

where Δ_k is the sampling time, $\mathbf{F}^{i_r}$ is the state transition matrix associated to the road segment r and adapted to a motion model $M_k^{i_r}$; $\mathbf{v}_k^{i_r}$ is a white zero-mean Gaussian random vector with covariance matrix $\mathbf{Q}_k^{i_r}$ chosen in such a way that the standard deviation σ_d along the road segment is higher than the standard deviation σ_n in the orthogonal direction associated to the road width given by the Geographic Information System (GIS). It is defined by:

$$\mathbf{v}_k^{i_r} = \mathbf{R}_{\theta_r} \cdot \begin{pmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_n^2 \end{pmatrix} \cdot \mathbf{R}_{\theta_r}^T \tag{9}$$

where $\mathbf{R}_{\theta_r}$ is the rotation matrix associated with the direction θ_r defined in the plane (O, X, Y) of the road segment r. The matrix $\mathbf{\Gamma}(\Delta_k)$ is defined in [19] and is the noise transition matrix adapted to motion model under consideration. For example, for a constant velocity motion model we have:

$$\mathbf{\Gamma}(\Delta_k) = \begin{pmatrix} \frac{\Delta_k^2}{2} & 0 \\ \Delta_k & 0 \\ 0 & \frac{\Delta_k^2}{2} \\ 0 & \Delta_k \end{pmatrix} \tag{10}$$

2) Estimation: As the linear motion model defined above does not consider possible maneuvers, the track may be lost by the tracker. To circumvent this problem, we use the well known Interacting Multiple Model (IMM) filter [19]. The IMM filter combines estimated states from multiple models to get a better state estimate when the target is maneuvering. The IMM is near optimal with a reasonable complexity. In Section III-B.1, a motion model $i \in I$ constrained to segment r and selected a time t_k, denoted by $M_k^{i_r}$, was defined. In a classical IMM estimator [20], the likelihood function (4) is defined for all models in I for a track $\mathcal{T}^{k,l}$, associated with the j_s-th measurement from sensor s. Thanks to Bayes rule, we can introduce in the likelihood of Equation (4), defined for each measurement $\mathbf{z}_k^{j_s}$, $(j_s = 1, \ldots, m_k^s)$, the motion model $M_k^{i_r}$ as follows:

$$p\{\mathbf{z}_k^{j_s} | Z^{k-1,l}\} = \Lambda_k^{j_s} = \sum_{i_r=0}^{N_I} \Lambda_k^{i_r,j_s} P\{M_k^{i_r} | Z^{k-1,l}\} \tag{11}$$

where $Z^{k-1,l}$ is the sequence of measurements associated with the track $\mathcal{T}^{k,l}$ and $\Lambda_k^{i_r,j_s}$ is the likelihood of the motion model i_r, (for $i_r = 0, 1, \ldots, N_I$), defined by:

$$\Lambda_k^{i_r,j_s} = p\{\mathbf{z}_k^{j_s} | M_k^{i_r} Z^{k-1,l}\} \tag{12}$$

The motion model probability $P\{M_k^{i_r} | Z^{k-1,l}\}$ is denoted by $\mu_k^{i_r}$. As in [20], by introducing "stopped" motion model, we obtain the following motion model likelihoods:

$$\begin{aligned} \Lambda_k^{i_r,j_s} =& P_D \cdot p\{\mathbf{z}_k^{j_s} | M_k^{i_r}, Z^{k-1,l}\} \cdot (1 - \delta_{j_s,0}) \\ &+ (1 - P_D) \cdot \delta_{j_s,0} \end{aligned} \tag{13}$$

where P_D is the probability of detection of the tracked target, while $\delta_{j_s,0}$ is the Kronecker index defined by $\delta_{j_s,0} = 1$ if $j_s = 0$ (no measurement) and $\delta_{j_s,0} = 0$ otherwise. The likelihood of the stopped target model (*i.e.* for $i_r = 0$) is then:

$$\Lambda_k^{0,j_s} = p\{\mathbf{z}_k^{j_s} | M_k^0, Z^{k-1,l}\} = \delta_{j_s,0} \tag{14}$$

The steps for deriving the IMM filter under road r constraint are the same than for the classical IMM filter: Step 1. Under the assumption of several possible models for road segment r as defined previously, the mixing probabilities are given for all pairs $(i_r, i'_r) = \{0, \ldots, N_I\}^2$ by:

$$\mu_{k-1|k-1}^{i_r|i'_r} = \frac{p_{i_r,i'_r} \cdot \mu_{k-1}^{i'_r}}{\bar{c}_{i_r}} \tag{15}$$

where $\bar{c}_{i_r} = \sum_{i'_r=0}^{N_I} p_{i_r,i'_r} \mu_{k-1}^{i'_r}$ is a normalizing factor. The probability of model switch, from motion model i_r to i'_r, depends on the Markov chain according to a *a priori* transition probability p_{i_r,i'_r} which does not depend on the constraint r. For the different models we consider, we use the mode transition probability matrix presented in [20] whose diagonal components for $i_r = 0, \ldots, N_I$ are:

$$p_{i_r,i_r} = \min\{u, \max\{v, 1 - \frac{\Delta_k}{\tau_{i_r}}\}\} \tag{16}$$

where τ_{i_r} is the mean sojourn time in motion model i_r, while $v = 0.5$ and $u = 0.95$ are the lower and upper limits for the no-transition (stay-in-place) probabilities, respectively. The other transition matrix elements are chosen in $[0, 1]$ such that $\sum_{i'_r=0}^{N_I} p_{i_r,i'_r} = 1$.

Step 2. The mixed estimate of the target state under the road segment r constraint is defined for $i_r = 0, \ldots, N_I$ by:

$$\bar{\mathbf{x}}^{i_r}_{k-1|k-1} = \sum_{i'_r=0}^{N_I} \mu^{i_r|i'_r}_{k-1|k-1} \hat{\mathbf{x}}^{i'_r}_{k-1|k-1} \tag{17}$$

where $\bar{\mathbf{x}}^{i_r}_{k-1|k-1}$ is the weighted target state estimate at time index $k-1$ on road segment r for the i-th motion model. The covariance of the estimation error is given by:

$$\bar{\mathbf{P}}^{i_r}_{k-1|k-1} = \sum_{i'_r=0}^{N_I} \mu^{i_r|i'_r}_{k-1|k-1} \cdot [\mathbf{P}^{i_r}_{k-1|k-1} + (\hat{\mathbf{x}}^{i'_r}_{k-1|k-1} - \bar{\mathbf{x}}^{i_r}_{k-1|k-1}) \cdot (\hat{\mathbf{x}}^{i'_r}_{k-1|k-1} - \bar{\mathbf{x}}^{i_r}_{k-1|k-1})^T] \tag{18}$$

Step 3. As aforementioned, the motion models are constrained to the associated road segment. Each constrained mixed estimate (17) is predicted and then associated to a new road, or to several new ones (in crossroad cases) which yields to the modification in the dynamics according to the new segments. The mixed estimates (17) and (18) are used as inputs for the filter matched to $M^{i_r}_k$. It uses the reports associated to the track $\mathcal{T}^{k,l}$ to calculate $\hat{\mathbf{x}}^{i_r}_{k|k}$, $\mathbf{P}^{i_r}_{k|k}$ (with Kalman filter associated to motion model $M^{i_r}_k$) and the corresponding likelihood (11). The estimates of each filter are obtained with a constrained Kalman filter (see [21] for more details).

Step 4. The model probability update is done for $i_r = 0, 1, \ldots, N_I$, and a measurement $j_s = 1, \ldots, m^s_k$ by :

$$\mu^{i_r}_k = \frac{1}{c} \cdot \Lambda^{i_r,j_s}_k \cdot \bar{c}_{i_r} \tag{19}$$

where c is a normalization coefficient and $\bar{c}_{i_r}$ is given in (15).

Step 5. The unconstrained combined state estimate (called global state estimate) is the sum of each constrained local state estimate weighted by each model probability,*i.e.* :

$$\hat{\mathbf{x}}_{k|k} = \sum_{i_r=0}^{N_I} \mu^{i_r}_k \hat{\mathbf{x}}^{i_r}_{k|k} \tag{20}$$

Its error covariance estimate is given by:

$$\mathbf{P}_{k|k} = \sum_{i_r=0}^{N_I} \mu^{i_r}_k \cdot [\mathbf{P}^{i_r}_{k|k} + (\hat{\mathbf{x}}^{i_r}_{k|k} - \hat{\mathbf{x}}_{k|k}) \cdot (\hat{\mathbf{x}}^{i_r}_{k|k} - \hat{\mathbf{x}}_{k|k})^T] \tag{21}$$

where $\mu^{i_r}_k$ is the probability of the motion model $M^{i_r}_k$, at current time t_k.

Here, we have presented briefly the principle of the IMM algorithm constrained to only one road segment r. However, a road section is often composed with several road segments. When the target is transitioning from one segment to another one, the problem is to choose the segments with the corresponding motion models that can better fit the target dynamics. The choice of a segment implies the construction of the directional process noise. That is why the IMM motions model set varies with the road network configuration. In such context the Variable Structure IMM (VS-IMM) offers a better solution for ground target tracking on road networks. Such algorithm has been referred as VS IMMC (C standing for Constrained) and was presented in details in [21].

step 6. Constrained state estimate. If the target is on the road r, we apply the following constraint:

$$\hat{\mathbf{x}}^r_{k|k} = \arg\max_{e^r_k} (\hat{\mathbf{x}}_{k|k}) \tag{22}$$

where e^r_k represents the constraint that the estimated target state $\hat{\mathbf{x}}_{k|k}$ belongs to the road r. So, the optimized estimated state $\hat{\mathbf{x}}^r_{k|k}$ must satisfy the following constraint:

$$\begin{cases} a\hat{x}_{k|k} + b\hat{y}_{k|k} + c = 0 \\ \langle [\hat{\dot{x}}_{k|k}, \hat{\dot{y}}_{k|k}]^T | \overrightarrow{n_r} \rangle = 0 \end{cases} \tag{23}$$

where (a, b, c) are the parameters of the straight line equation of road r in the East North Up (ENU) direction and $\overrightarrow{n_r}$ is the normal vector of road r. The constraint (23) can be expressed as follows:

$$\tilde{\mathbf{D}} \hat{\mathbf{x}}_{k|k} = \mathbf{L} \tag{24}$$

with

$$\tilde{\mathbf{D}} = \begin{bmatrix} a & 0 & b & 0 \\ 0 & a & 0 & b \end{bmatrix} \tag{25}$$

and

$$\mathbf{L} = \begin{bmatrix} -c \\ 0 \end{bmatrix} \tag{26}$$

Based on Lagrangian relaxation method [22] applied to (22), we obtain the state vector estimate $\hat{\mathbf{x}}^r_{k|k}$:

$$\hat{\mathbf{x}}^r_{k|k} = \hat{\mathbf{x}}_{k|k} - \mathbf{P}_{k|k}\tilde{\mathbf{D}}^T(\tilde{\mathbf{D}}\mathbf{P}_{k|k}\tilde{\mathbf{D}}^T)^{-1}(\tilde{\mathbf{D}}\hat{\mathbf{x}}_{k|k} + \mathbf{L}) \tag{27}$$

and its associated error covariance $\mathbf{P}^r_{k|k}$ constrained on road r:

$$\mathbf{P}^r_{k|k} = (\mathbf{I} - \mathbf{W}_k)\mathbf{P}_{k|k}(\mathbf{I} - \mathbf{W}_k)^T \tag{28}$$

where $\mathbf{I}$ is the identity matrix with the same dimension of $\mathbf{P}_{k|k}$ and the matrix $\mathbf{W}_k$ is defined by :

$$\mathbf{W}_k = \mathbf{P}_{k|k}\tilde{\mathbf{D}}^T(\tilde{\mathbf{D}}\mathbf{P}_{k|k}\tilde{\mathbf{D}}^T)\tilde{\mathbf{D}} \tag{29}$$

The prediction III-B.1 and estimation III-B.2 processings are sufficient to be introduced in usual MTT algorithm for tracking multiple maneuvering ground targets. However, the scope of the paper is to track and classify jointly the targets with WSN. In the next section, we present our approach to combine class information provided by sensors and class information obtained from track behavior given by estimated states (27) and its associated covariance (28).

IV. Track classification

We apply the classification method we proposed in [13] to each confirmed track delivered by the WSN multiple target tracker. To this end, we use two pieces of information as inputs to the classifier: the estimated class of the track according to the geographical information on the one hand and the estimated class based on sensor measurements on the other hand. We frame the class estimation and combination problem into the Transferable Belief Model (TBM) [23]. The following subsections will succinctly describe the different mathematical concepts used in the classifier, before going deeper into its architecture.

A. Basics of belief functions

The belief function theory, also known as Dempster-Shafer Theory (DST) has been introduced in 1976 by Shafer in his Mathematical Theory of Evidence book [24]. In DST, a discrete and finite frame of discernment (FoD) is defined as $\Omega = \{\omega_i, i = 1, \ldots, c\}$ consisting of c exclusive and exhaustive hypotheses for the problem at hand. In our case each ω_i represents a specific target class. A mass function can be defined over the power-set of Ω, denoted by 2^Ω, which is the set of all subsets of Ω. For example, if the frame of discernment $\Omega = \{\omega_1, \omega_2, \omega_3\}$, then its power-set is $2^\Omega = \{\emptyset, \omega_1, \omega_2, \omega_1 \cup \omega_2, \omega_3, \omega_1 \cup \omega_3, \omega_2 \cup \omega_3, \Omega\}$, where $\emptyset$ denotes the empty set and $\cup$ the disjunction operator. A mass function, also named Basic Belief Assignment (BBA), is mathematically defined as a mapping $m : 2^\Omega \rightarrow [0; 1]$ which satisfies $m(\emptyset) = 0$ and

$$\sum_{A \in 2^\Omega} m(A) = 1 \tag{30}$$

All the elements $A \in 2^\Omega$ such that $m(A) > 0$ are called focal elements of m. The mass $m(\omega_i)$ represents the support degree that the target is of class ω_i only. In pattern classification problems, if A is a set of classes (e.g. $A = \omega_i \cup \omega_j$), $m(A)$ can be used to characterize the imprecision degree (partial ignorance) among the classes ω_i and ω_j. $m(\Omega)$ denotes the total ignorance degree. $m(\Omega) = 1$ is called the vacuous mass function and characterizes the total ignorance about the class of the target. It is a neutral element in the fusion process.

The plausibility of A is defined for all $A \subseteq \Omega$ by

$$Pl(A) = \sum_{B \in 2^\Omega |\ A \cap B \neq \emptyset} m(B). \tag{31}$$

The quantity $Pl(A)$ can be interpreted as an upper bound of an unknown probability, $P(A)$.

In the DST framework, Dempster's rule of combination (D-S rule) is adopted to combine BBAs provided by independent and reliable sources. For example in the case of a multi-classifier system, each classifier can be considered as a source of evidence, which output can be represented by a mass function.

Considering two distinct sources of evidence, whose BBAs are respectively defined by m_1 and m_2, the D-S combination of m_1 and m_2 is denoted $m_1 \oplus m_2 = m_{1\oplus 2}$, and defined by $m(\emptyset) = 0$, and $\forall A \in 2^\Omega, A \neq \emptyset$ as:

$$m_{1\oplus 2}(A) = \frac{1}{1 - K} \sum_{B,C \in 2^\Omega, B \cap C = A} m_1(B) \cdot m_2(C) \tag{32}$$

where $K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C) < 1$ is the total conjunctive conflicting mass. If $K = 1$ (corresponding to the total conflicting case) D-S rule cannot be applied.

The decision process, i.e the track classification, is done as within TBM framework proposed by Smets and Kennes [23]. The TBM is composed by two levels: a credal level, used to represent and maintain beliefs quantified by belief functions, and a pignistic level where those beliefs are used to make decision.

The pignistic transformation defined by Smets in [25] approximates a BBA by a subjective probability measure thanks to a uniform redistribution of the mass of each partial ignorance (if any) on the singletons involved in it. More precisely, the pignistic probability derived from a BBA m is a mapping $BetP\{m\} : \Omega \rightarrow [0, 1]$ such that:

$$BetP\{m\}(\omega_k) = \frac{1}{1 - m(\emptyset)} \sum_{A \subseteq \Omega, \omega_k \in A} \frac{m(A)}{|A|} \tag{33}$$

The decision is then made by selecting the element ω_k having the highest pignistic probability and greater than a given threshold $\gamma \in [0, 1]$:

$$\omega_0 = \underset{\omega_k \in \Omega, BetP\{m\}(\omega_k) > \gamma}{\arg\max} BetP\{m\}(\omega_k) \tag{34}$$

In our application and tests we have set $\gamma = 0.7$.

If there is no class ω_k such that $BetP\{m\}(\omega_k) > \gamma$, then the target is assigned to the closure class $\mathbf{c} = \{Unknown\}$.

B. Classifier description

The classifier we designed operates following three steps:

- Step 1. A fuzzy inference system takes as input the kinematic data (speed and position) of the track, and outputs a vector of possibilities;
- Step 2. a function maps the possibility vector and probability of sensor class from the inference system onto basic belief assignments;
- Step 3. a fusion function then combines the new mass functions for the tracks with previous ones, and derives the associated pignistic probabilities.

1) Fuzzy Inference system: As described in [13], the first step is to define the membership functions regarding the terrain and the velocity for each hypothesis described in (7). Because we do not have enough data for learning the membership functions, we build them , using expert knowledge. We have defined a set of types for the terrain under surveillance:

$$\begin{aligned} \mathrm{T} \triangleq \{&\text{road network}, \text{forest}, \text{building}, \text{grass land}, \\ &\text{hydrographical area}, \text{unknown}\ \} \end{aligned} \tag{35}$$

Then, for each type of terrain of T, we construct a *"trafficability"* capacity for each target class of (7), according to the minimum (and maximum) speed for each target class on each terrain type. This trafficability capacity is then obtained with multi-fuzzy logic inference system, where each inference is conditioned by a terrain attribute. In this way, when the tracker updates the confirmed track $\mathcal{T}^{k,l}$, we extract from (27) and (28) its estimated position, velocity and angular velocity and uncertainties. Furthermore, based on terrain type estimation, we can use the fuzzy inference system to obtain the possibility distribution over Ω for the target under tracking.

At this step of the algorithm, the inference system provides for any confirmed track $\mathcal{T}^{k,l}$ with kinematic parameters K, a likelihood vector $L_{K,\mathrm{T}}$ defined over the set of classes considering the terrain T where the target is located.

2) From Likelihood to Belief functions: Based on what we described in [13], the output of the fuzzy inference system is a likelihood vector $L_{K,T} = [L\mathbf{c}_1, L\mathbf{c}_2, \ldots, L\mathbf{c}_8]$ where $L\mathbf{c}$ is the likelihood that the observed track, with kinematic parameter K and terrain parameter T, belongs to class $\mathbf{c} \in \mathbf{C}$. On the other hand, we recall that the likelihood that the observed track, with sensor information is noted L_C and described in Section III-A.3.

These likelihood vectors are then transformed into a plausibility functions following the approach proposed in [26] as follows:

$$Pl_\Omega(A;x) = \sup_{\omega \in A} pl(\omega;x) = \frac{\sup_{\omega \in A} L(\omega;x)}{\sup_{\omega \in 2^\Omega} L(\omega;x)}, \forall A \subseteq \Omega \tag{36}$$

where $L(.;x)$ is the likelihood function and pl is the contour function such that $pl(\omega) = Pl(\{\omega\})$. From the plausibility function, we then derive the mass function of A using the Möbius inverse formula:

$$m_\Omega(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|+1} Pl(\bar{B}), \quad \forall A \subseteq \Omega \tag{37}$$

where $\bar{B}$ is the complement of B relatively to Ω.

Hence, we have at this point for each track, considering its velocity and position, the support degree that it belongs to any subset of classes. The output of this process are the mass functions $m_{C,k}$ and $m_{K,k}$ defined over Ω obtained respectively from the likelihoods L_C and $L_{K,T}$.

3) Fusion and Decision Making: At each update of a track, *i.e.* at time k, a new estimate of its velocity and position is provided by the tracker and a new mass function $m_{K,k}$ is derived as previously defined.

The global mass function at time k, denoted by $m_{\mathscr{C},k}$ is obtained by fusing the new mass function $m_{C,k}$ with the previous global mass function at the previous time t_{k-1} $m_{\mathscr{C},k-1}$ by Dempster's rule of combination (Eq. (32)), that is:

$$m_{\mathscr{C},k} = m_{K,k} \oplus m_{C,k} \oplus m_{\mathscr{C},k-1} \tag{38}$$

Decision on the class label to assign to the track is then made following Eq. (34).

The fusion operation defined in (32) plays an important role in our classification and decision schemes. We make the assumption that over time, the target will have a discriminant behavior (excluding incompatible kinematic or position situations) leading to discard some target classes. Yet, because the estimation of the terrain type over which the target is located, is itself based on an estimated position of the target, it can thus be imprecise or even wrong. Consequently, it could yield discarding the correct class among $\mathbf{C}$ defined in (7). In our previous work [13], the GIS data was Boolean, meaning that the derivation of $m_{K,k}$ was based in a single attribute leading to quite high missclassification rate. In this work, in order to improve the contextual discounting in the mass function and by that the classification rate, we change our approach compared to [13]. Rather, we consider several attributes for the terrain from which we derive several BBAs obtained through the inference system for each attribute. The final BBA is then obtained by combining all discounted BBAs for each terrain attribute in order to get $m_{K,k}$.

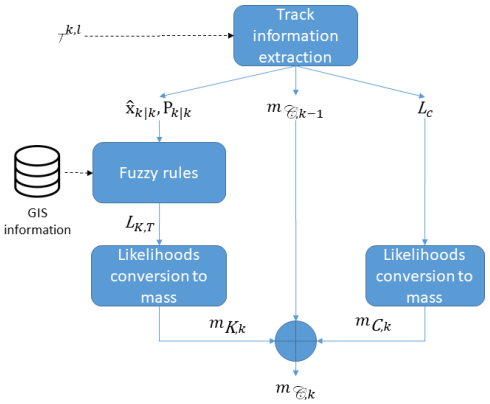


Fig. 4: Representation of the sensor and kinematic based classifier. Conversions refers to Section IV-B.2 to transform likelihoods or probabilities into mass functions.

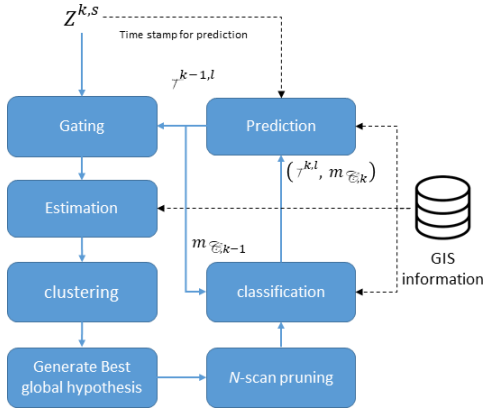


Fig. 5: TO-MHT algorithm flowchart with geographical information.

V. Data Fusion and Association

In sections III and IV, we have respectively presented our approach to track and classify a single target. However we work with multiple sensors and multiple targets that implies track-to-measurement association problems. Algorithm must be adapted to track targets with several sensors. The proposed method is based on the (TO-MHT) framework [27], which takes advantage of the track tree structure to manage and maintain hypotheses sets. Each association between track and measurement generates hypotheses represented by a new branch in the tree of the possible joint association. Figure 5 shows the basic elements of a typical TO-MHT system. The prediction and estimation processings refer respectively to sections III-B.1 and III-B.2. The classification process in figure 5 refers to part IV. Hypotheses are reformed from tracks at each scan and the tracks that survive pruning step are predicted to the next scan where the process continues. An overview of the core components of the TO-MHT framework given in Sections V-A and V-B provides some efficient strategies for limiting the number of hypotheses.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2021.3129181, IEEE Sensors Journal

A. Centralized fusion

According to the communication network structure described in Section II, all measurements $\mathbf{z}_k^{j_s}$ provided by the N_s sensors, are broadcastover the network to reach a centralized fusion node. This node processes sequentially the set of measurements by sensor packet $Z^s(k)$, named scan, defined for $(j_s = 1, \ldots, m_k^s)$ and $(s = 1, \ldots, N_s)$ by:

$$Z^s(k) = \{\mathbf{z}_k^{j_s}\} \tag{39}$$

where m_k^s is the number of measurements at time t_k from the sensor s and N_s the number of sensors.

B. Multiple target tracking hypotheses

TO-MHT uses a tree structure to store all possible track-to-measurement association hypotheses. Evaluation of alternative track formation hypotheses requires a probabilistic expression that includes all aspects of data association problem. These aspects include the prior probability of target presence, the false alarm density of each sensor, the detection sequences and dynamic consistency of the observation contained in the track. It is convenient to use the Log-Likelihood Ratio (LLR) to evaluate each hypothesis (*i.e* a track-to-measurement association). The LLR is also termed "track score". For a given track $\mathcal{T}^{k,l}$, it is convenient to use the LLR as a score of a track $\mathcal{T}^{k,l}$ because it can be expressed at current time k in the following recursive form [28]:

$$L_{k,l} = L_{k-1,n} + \Delta L_{k,l} \tag{40}$$

with

$$\Delta L_{k,l} = \log\left(\frac{\Lambda_k^{j_s}}{\lambda_{fa}}\right) \tag{41}$$

and

$$L(0) = \log\left(\frac{\lambda_{fa}}{\lambda_{fa} + \lambda_{nt}}\right) \tag{42}$$

where λ_{fa} and λ_{nt} are respectively the false alarm rate and the new target rate per unit of surveillance volume. $\Lambda_k^{j_s}$ is the global likelihood function associated to a measurement j_s for $(j_s = \{1, \ldots, m_k^s\})$ given in (4).

The first step consists to apply prediction process (presented in the subsection III-B.1) for all tracks. The second step is the gating (*i.e.* the selection) of measurements followed by the track formation. When the new set $Z^s(k)$ of measurements from sensor s is received, a standard gating procedure [28] is applied in order to determine the valid measurement reports for track pairings. The existing tracks are predicted and estimated with IMM at first (presented in subsection III-B.2), and then extrapolated confirmed tracks are formed. When the track is not updated with reports, the stop-motion model is activated. The process of clustering is used to put altogether the tracks that share common measurements. The clustering limits the number of hypotheses to generate, and therefore it can drastically reduce the complexity of tracking system. The result of the clustering is a list of tracks that are interacting. The next step is to create hypotheses of compatible tracks. For each cluster, multiple compatible hypotheses are formed to represent the different compatible tracks scenarios. Each hypothesis is evaluated according to the track score function (41) associated to the different tracks. Then, a technique is necessary to find the set of hypotheses that represents the most likely tracks collection. The unlikely hypotheses and associated tracks are deleted by a pruning method (if the hypothesis probability is less than a chosen threshold P_{Hypo}), and only the N_{Hypo} best hypotheses are kept in the system. For each track, a *a posteriori* probability is computed, and a classical *N-Scan* pruning approach [28] is used to delete the most unlikely tracks. With this approach the most likely tracks are selected to reduce the number of tracks. However, the *N-Scan* technique combined with the constraint implies that other tracks hypotheses (*i.e.* constrained on other road segments) are arbitrary deleted. To avoid this problem, we modify the *N-Scan* pruning approach in order to select the N_k best tracks on each N_k road sections. Finally, the class of all updated tracks is done according the process presented in the section IV. It is worth noting that the class is never used for gating or track hypotheses evaluation steps.

VI. Experimental results

In this paper, we illustrate the concept of our "abandoned" WSN to achieve tracking and classification in a wide surveillance area, but for several constraints[5] we cannot provide for now precise performance evaluations based on many real experimental acquisitions.

A. Scenario description

In order to investigate the feasibility of the proposed algorithms and the SEXTANT system for the ground surveillance application, we have performed live recordings on site with real ground sensors and test vehicles and pedestrians equipped with GPS transponders. The scenario contains both live recordings from the GPS and a simulated part.

The live recordings were carried out at Palaiseau on July 9th, 2018. The ground sensor network was deployed according to figure 6. Table I represents the sensor types and how they were deployed. This table gives also the standard deviations (in range σ_ρ and azimuth σ_θ) of each detection and the minimal and maximal distance of detection (d_{min} and d_{max} respectively). The sensors were radars, seismics, optronic and PIR sensors. The ground sensor network was composed of a number of sensors, each associated with a node with transmission capability as described in section II. For the geographic layer we have used shapefiles with metric precision. The only grass, forest and road layers were used (there is no water on site). The PIR nodes were only used to wake up principal sensors.

During the data recordings for the real part the only sensor radar 3, optronic sensors 5,7 and WUN nodes 4 and 6 were used.

B. Ground truth

The figures 7 and 8, represents views of the trajectories of the target. Only targets 1_Car, 2_PAX, 5_Car, 6_Car are true targets. The five other targets have been simulated 4_Car, 8_Car,

[5]mainly due to budget limitations.

TABLE I: sensor parameters table.

id sensor	sensor type	$\sigma_\rho(m)$	σ_θ ($^\circ$)	$d_{min}(m)$	$d_{max}(m)$
1	PIR	-	0.1	0.1	10
2	seismic	2	-	1	30
3	radar	5	0.01	10	280
4	WUN	-	-	-	250
5	optronic	-	0.001	5	150
6	WUN	-	-	-	250
7	optronic	-	0.001	5	15
8	PIR	-	0.1	0.1	10
9	WUN	-	-	-	250
10	WUN	-	-	-	250
11	radar	5	0.01	10	280
12	WUN	-	-	-	250
13	WUN	-	-	-	250
14	WUN	-	-	-	250
15	WUN	-	-	-	250
16	seismic	2	-	1	30
17	PIR	-	0.1	0.1	10
18	fusion node	-	-	-	-

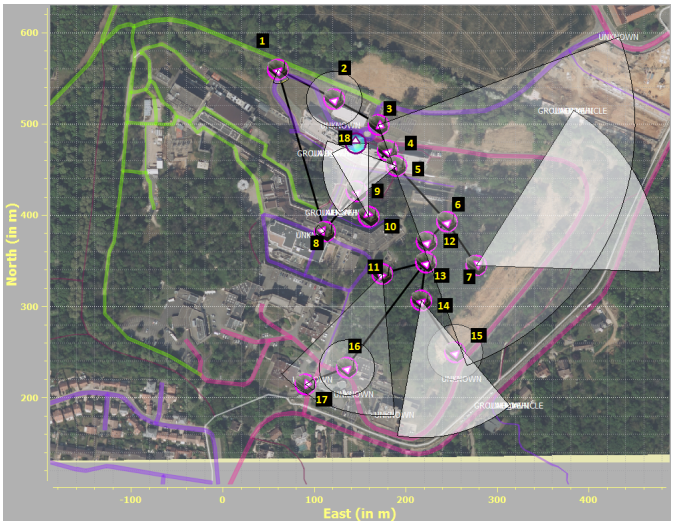


Fig. 6: Node locations on ONERA's site.

9_PAX, 10_PAX, 11_PAX for the need of demonstration. We can observe maneuvering targets on and out of the road with crossing targets problem.

C. MTT parameters

At the fusion node we have plugged a C2 (Control and Command) station to display the results of MTT algorithms. In our experiment, the setting of tracker parameters have been done as follows:

1) SB-MHT parameters: The maximum number of track hypotheses N_{Hypo} in cluster is set to 20 and the minimal probability P_{Hypo} to keep each track hypothesis is fixed to 0.01. The *N-Scan* number to prune the tree of track hypotheses is fixed to 3. The validation gate parameters are $\delta_{1D} = 2$ for the bearing only measurements and $\delta_{2D} = 4.835$ for range and azimuth measurements. For the track initialization: each report at every scan is considered as a new track and declared as "initialized". The initialized track is declared as "tentative track" if the score (40) of each track hypothesis is greater than 2. If the track score (40) is greater than 10 the "tentative track" is declared as "confirmed".

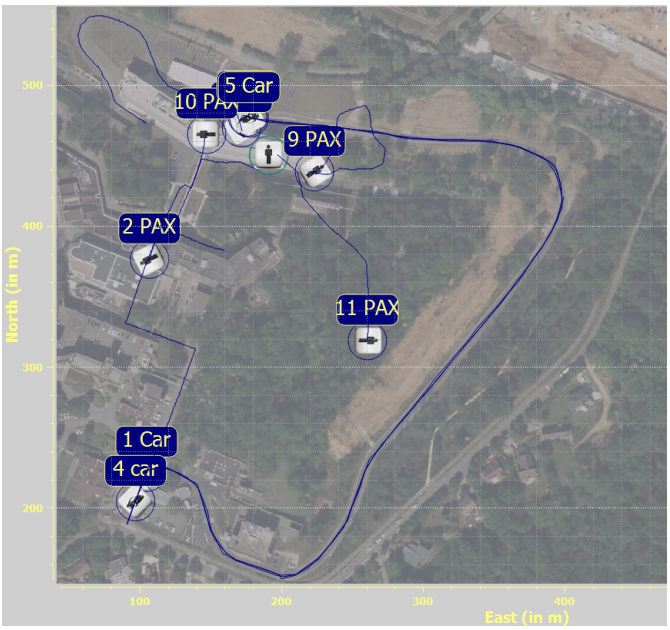


Fig. 7: Trajectories of the targets at the beginning.

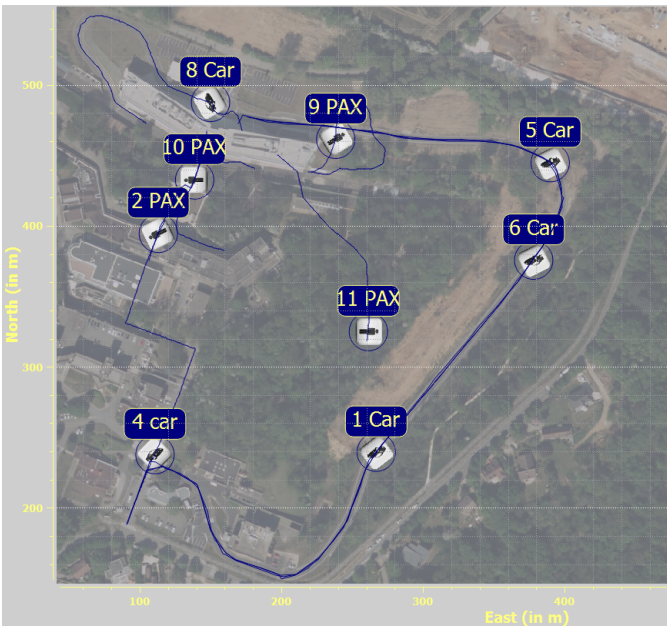


Fig. 8: Trajectories of the targets after 1 minutes.

2) IMM parameters: The motion models are constant velocity motion models. A motion model M^1 to track the targets which move with constant velocity, a motion model M^2 with a big state noise to palliate the maneuvers of the target and a stop-model M^0. The noises of the previous motion models have been set respectively as : $\sigma = 0.4m.s^{-2}$ for M^1, $\sigma = 4m.s^{-2}$ for M^2 and $\sigma = 0.1m.s^{-2}$, for M^0. In our experiment we did use the following prior probability vector:

$$\mu_0 = [0.9\ 0.1\ 0]' \tag{43}$$

with the following sojourn time (defined in [19]) : $\tau_0 = 1$, $\tau_1 = 40$ and $\tau_2 = 20$.

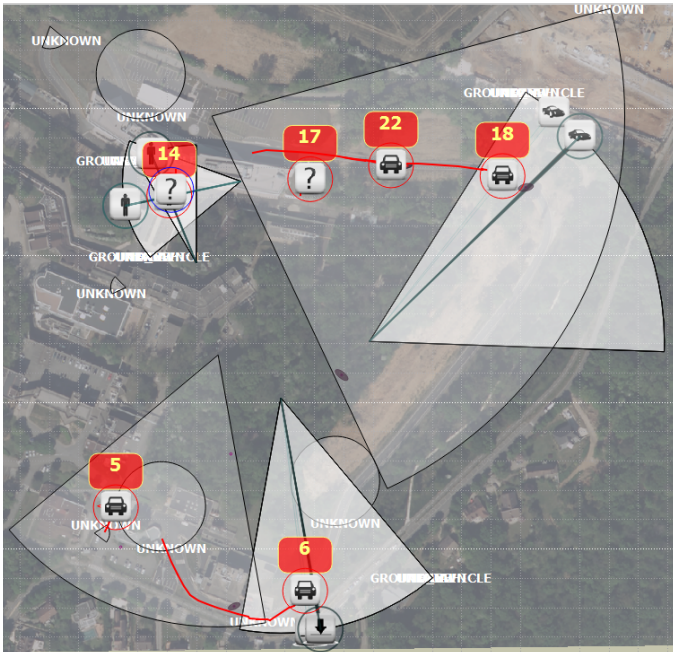


Fig. 9: Track situation at 10 h 42 m 20 s.

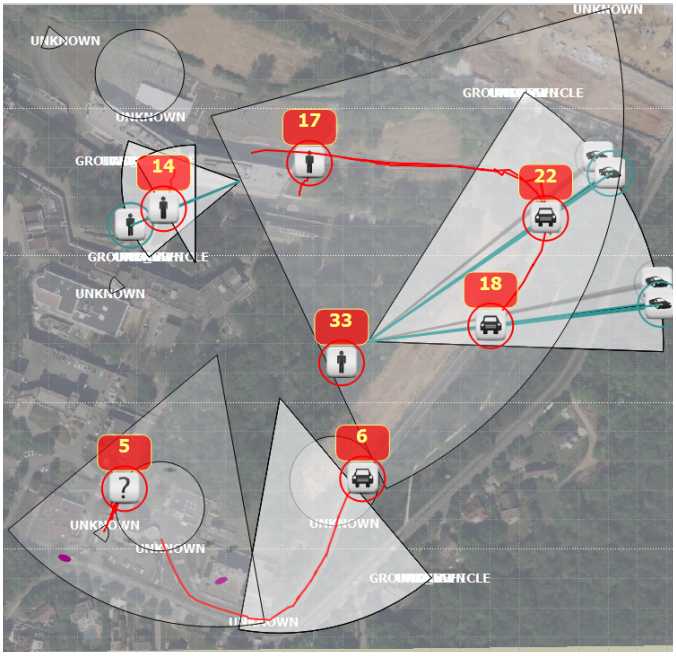


Fig. 10: Track situation at 10 h 42 m 27 s.

3) classification parameter: The classification parameter is the threshold γ used to classify a track. If the pignistic probability (33) of a target class is greater than $\gamma = 0.7$, the track is committed to the class having the max of $BetP(.)$.

D. results with environmental information

During trial, all sensor measurements were broadcasted to only one fusion node in order to work in a centralized architecture. The MTT algorithm initializes and updates tracks. We only display "confirmed" tracks (see step 2 of the MTT algorithm V-B for track confirmation or deletion). At 10 h 42 m 20 s (figure 9), ground vehicles and PAX are well tracked (the track continuity is maintained). The tracks 5,6, 17 and 22 are quickly classified as ground vehicle because the tracks are projected on road network by the algorithm. The used constraint tracking motion models carry along to keep track on the road and improve track prediction on the road. This allows to conserve the tack out sensor field of view or update the track only with bearing measurement only. This is the case for the track 6 at this time. We observe that the track 6 is maintained according to road constraint motion model despite of bearing only measurement provided by video sensor.

Target 1 crosses target 5 and 6 (figure 8). Despite the crossing the associated tracks (respectively track 6, 22 and 18) tracks are kept (figure 11). As explained previously, the constrained motion models allow to estimate precisely the track state, and to maintain the track on the road. That implies a better track prediction and improves the MTT association processing.

The figures 13 and 14 represent respectively the membership layer probability obtained and cumulated $BetP$ target classification for the target associated with track 6. If the probability that is on the road is not equal to 1 (figure 13), the estimated track velocity and acceleration is characteristic of a vehicle in

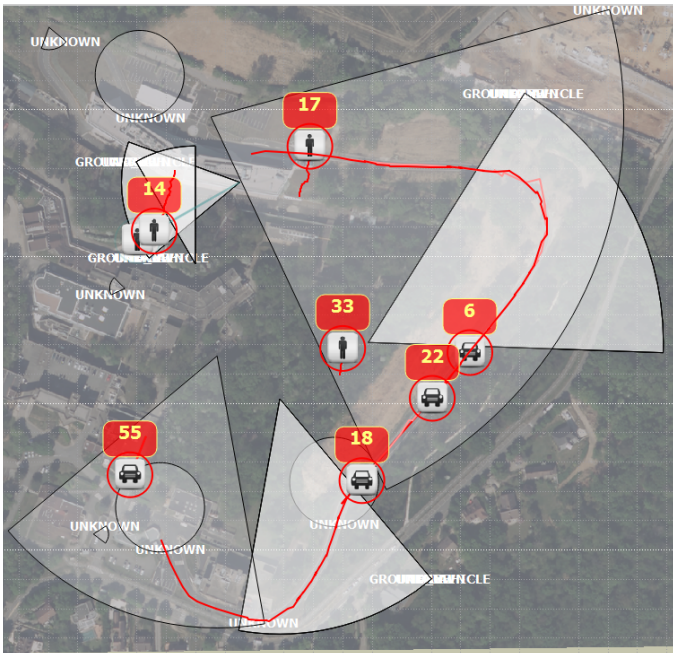


Fig. 11: Track situation at 10 h 42 m 46 s, zoom on track 4.

grass land, that is why we can observe that the classification algorithm quickly classifies the targets as "car". After the turn, the velocity of the vehicle decreases and the probability of the "car" class becomes close to the probability of "SUAV" (Small UAV) class. In fact, with this kinematic, fuzzy inferences in velocity and acceleration provide a close likelihood "Car" and "SUAV" classes. The mass conflict between those classes becomes higher. But, at this step, we can consider other fusion operator to have a better consideration of the conflict. In our future research works we will test some sophisticated fusion methods to deal more efficiently with the conflicting

information.

The track 14 associated to the PAX is initialized according triangulation of bearing measurements provided by video sensors 5 and 10.

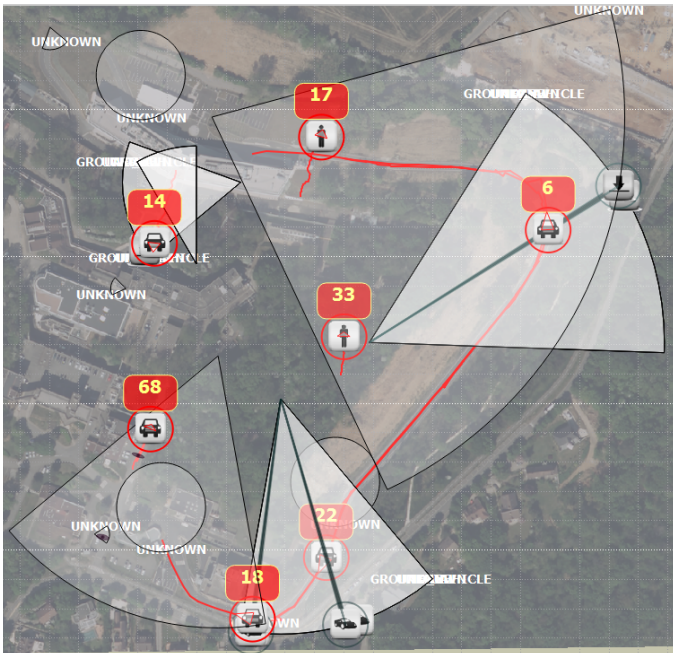


Fig. 12: Track situation at 10 h 42 m 58 s.

If we observe track 14, we can see that our classification algorithm has quickly converged to "PAX" class because of the correct classification probability of bearing optronic sensor measurement. But if we observe the cumulated $BetP$ target classification of track 17 (figure 15), we observe the conflict between three classes ("PAX", "car" and "SUAV"). The "PAX" fuzzy modelling of estimated kinematic of track 17 on grass-land geographical layer is similar with "car" (modelling of cross-country vehicle) or "SUAV".

VII. Conclusions

The aim of the research work is to develop a demonstrator platform in order to study the feasibility of a WSN joint tracking and classification technology and adapt it to improve surveillance and intelligence capabilities. In this paper we have focused on the abilities of data fusion algorithm used

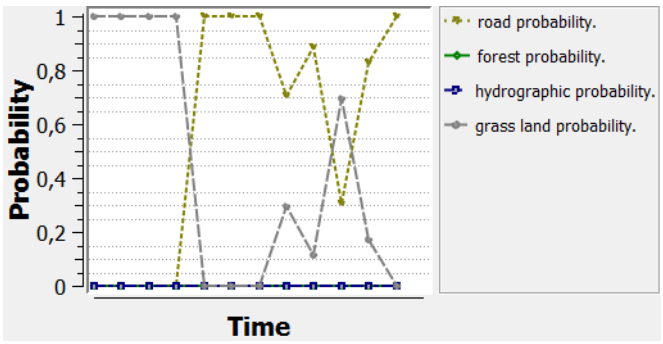


Fig. 13: Membership geographic layer probability of track 4.

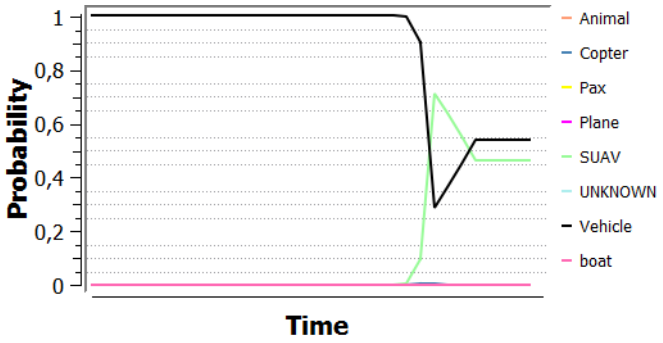


Fig. 14: Cumulated $BetP$ classification of track 4.

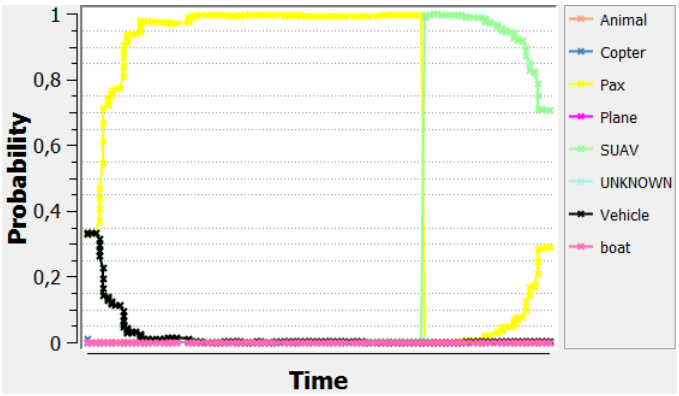


Fig. 15: Cumulated $BetP$ classification of track 17.

to detect, track and classify multiple targets thanks to a smart wireless sensor network. Based on owner developed sensors, we have proceeded to real trials in order to validate the data fusion algorithm. The proposed Joint Classification and MTT algorithm confirms existing results on real data. We have proved, with real data, the capability of SEXTANT system to estimate the target states (location and velocity) with sensor network constraints. Each motion model is submitted to ground network test to constrained it (or not) on the road. The contextual constraints improve the track estimation and track continuity. This is more efficient to maintain track when the target is maneuvering in unobservable area. After the estimation step, the MTT algorithm has proceeded to target classification taking into account the classification provided by sensor and the contextual classification estimated by the track behavior correlated to terrain layer. We have succeeded to build a behavioral context-based joint tracking-classification (JTC) system. Based on our tests, the target classification using fuzzy logic seems promising, as a virtual sensor to provide a classification indication for the tracker. Different researches could be done in future works to enhance the performances of this JTC based on the wireless sensor applications described in [29]. Combined with our results, some preliminary insights and recommendations for the use and the development of new generations of smart sensors in operational exercises have been identified, and they will orient new research in this field. As perspective works, we plan to improve this WSN surveillance system on several points: 1) improve the estimation of layer membership probability thanks to neuro-fuzzy techniques, 2)

improve the target classification thanks to a more efficient fusion rule like the proportional conflict redistribution rule, 3) improve the decision-making strategy based on belief functions, and 4) improve the tracking precision with more sophisticated maneuvering models.

REFERENCES

[1] B. Pannetier, J. Dezert, and J. Moras, "Ground target tracking and classification in an unattended wireless sensor network," *SPIE Conference, Baltimore*, May 2015.

[2] E. Blasch, E. Bossé, and D. Lambert, *High-Level Information Fusion Management and Systems Design*. Artech House, 2012.

[3] L. Snidaro, J. García, J. Llinas, and E. Blasch, *Context-Enhanced Information Fusion - Boosting Real-World Performance with Domain Knowledge*. Springer, Cham, 2016.

[4] C. Yang, E. Blasch, and M. Bakich, "Nonlinear constrained tracking of targets on road," *FUSION 2005, Conf.*, July 2005.

[5] C. Yang and E. Blasch, "Fusion of tracks with road constraints," *FUSION 2008, Conf.*, July 2008.

[6] C. C. Bidstrup and al., "Tracking multiple vehicles constrained to a road network from a uav with sparse visual measurements," *American Control Conference (ACC)*, July 2019.

[7] M. Ekman and H. Palsson, "Ground target tracking of vehicles in a wireless sensor network," *FUSION 2012, Conf., Singapore*, July 2012.

[8] B. Pannetier and J. Dezert, "Track segment association with classification information," *Sensor Data Fusion: Trends Solutions Applications (SDF),Workshop on*, 2012.

[9] M. Ulmke and W. Koch, "Road-map assisted ground moving target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, no. 42, pp. 1264–1274, 2006.

[10] J. Zheng and M. Gao, "Tracking ground targets with a road constraint using a gmphd filter," *Sensors journal*, 2018.

[11] L. S. J. Sjudin, M. Marcusson and L. Hammarstrand, "Extended object tracking using sets of trajectories with a phd filter," *FUSION 2021*, November 2021.

[12] E. Blasch, "Modeling intent for a target tracking and identification scenario," *SPIE Conference*, August 2004.

[13] B. Pannetier and al., "Data fusion for target tracking and classification with wireless sensor network," *SPIE Conference, Edinburgh*, vol. 9986, March 2016.

[14] E. P. Blasch, J. Dezert, and P. Valin, "DSmT applied to seismic and acoustic sensor fusion," *Proceedings of the 2011 IEEE National Aerospace and Electronics Conference (NAECON)*, 2011.

[15] P. A. Thomas, G. F. Marshall, D. J. Stubbins, and D. A. Faulkner, "Towards an autonomous sensor architecture for persistent area protection," *SPIE Conference, Edinburgh, United Kingdom*, October 2016.

[16] A. AbuBaker, R. Qahwaji, S. Ipson, and M. Saleh, "One scan connected component labeling technique," *2007 IEEE International Conference on Signal Processing and Communications, Dubai*, 2007.

[17] L. Raphaël, J. Moras, and B. Pannetier, "Vibrating beam mems seismometer for footstep and vehicle detection," *IEEE SENSORS*, 2016.

[18] L. Raphael and al., "Nonlinear regime operation for a high resolution vibrating beam ugs seismometer," *Analog Integrated Circuits and Signal Processing*, vol. 82, no. 3, pp. 621–626, 2015.

[19] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Artech House, 2001.

[20] T. Kirubarajan and Y. Bar-Shalom, "Tracking evasive move-stop-move targets with an mti radar using a VS–IMM estimator," *IEEE Transactions Aerospace and Electronics Systems*, vol. 39, pp. 1098–1103, July 2003.

[21] B. Pannetier, V. Nimier, and M. Rombaut, "Multiple ground target tracking with a GMTI sensor," *MFI, Conf., Heidelberg, Oct.*, oct 2006.

[22] J. Zheng, J. Min, and H. He, "Tracking ground targets with road constraint using a gaussian mixture road-labeled phd filter," *13th International Conference on Machine Learning and Computing*, Feb. 2021.

[23] P. Smets and R. Kennes, "The transferable belief model," *Artificial Intelligence*, vol. 66, no. 2, pp. 191 – 234, 1994.

[24] G. Shafer, *A Mathematical Theory of Evidence*. Princeton University Press, 1976, vol. 1.

[25] P. Smets, "Decision making in the TBM: the necessity of the pignistic transformation," *International Journal of Approximate Reasoning*, vol. 38, no. 2, pp. 133 – 147, 2005.

[26] I. Couso, L. Sanchez, and T. Denoeux, "Special issue: Harnessing the information contained in low-quality data sources likelihood-based belief function: Justification and some extensions to low-quality data," *International Journal of Approximate Reasoning*, vol. 55, pp. 1535–1547, 2014.

[27] S. Pan, Q. Bao, and Z. Chen, "An efficient to-mht algorithm for multi-target tracking in cluttered environment," *IEEE 2nd Advanced Information Technology, IAEAC*, 2017.

[28] S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. Artech House, 1999.

[29] H. M. A. Fahmy, *Concepts, Applications, Experimentation and Analysis of Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis*. Springer, 2021.

Benjamin Pannetier was born in Paris on November 30th, 1979. He got his Ph.D. in Automatic Control and Signal Processing from the University of Grenoble in 2006. From 2005 to July 2021 he has been a research engineer at the French Aerospace Lab (ONERA) working on target tracking, detection, estimation theory and data fusion for the battlefield surveillance systems for the french army. Since August 2021 he is with CSGroup Company working on defense applications.

Jean Dezert got his Ph. D. in Automatic Control and Signal Processing from Paris XI University in 1990. During 1991-1992 he did a postdoc at UConn under the supervision of Professor Bar-Shalom. Since 1993, he is Senior Research Scientist at the French Aerospace Lab. His current research interests include tracking, information fusion, belief functions and decision-making under uncertainty. Jean Dezert has been President of International Society of Information Fusion, and been involved in technical program committees and panel discussions of several international conferences. He has co-edited eight books, written 6 book chapters, and around 240 technical papers. For more details, see https://www.onera.fr/fr/staff/jean-dezert

Julien Moras was born in Vénissieux, France, in 1985. He received the Engineer's Degree from the ISAE in 2008 and Ph.D. in technology of information and systems from the UTC in 2013. Since 2013 he is a research engineer at the French Aerospace Lab (ONERA). His research interests include data fusion theory in wireless sensor networks and robotic embedded perception for autonomous navigation.

Raphael Levy was born in Paris in 1979. He got his Ph.D. in Microtechnology engineering from University of Paris-Saclay in 2005. Since 2007, he is a research engineer at the French Aerospace Lab (ONERA). His current research interests include High performance MEMS sensors design and associated microtechnologies. He is now Head of Sensors and Micro/Nano Technology Unit.