Improving Joint Tracking and Classification with the Transferable Belief Model and Terrain Information

Matthew Roberts
School of Computer Science & Informatics
Cardiff University
Cardiff, UK
M.S.Roberts@cs.cardiff.ac.uk

David Marshall
School of Computer Science & Informatics
Cardiff University
Cardiff, UK
Dave.Marshall@cs.cardiff.ac.uk

Gavin Powell
EADS Innovation Works
Newport, UK
Gavin.Powell@eads.com

Abstract – *This paper presents an extension to an existing Joint Tracking and Classification approach for Wireless Sensor Networks. Terrain information and how it restricts target movement is used, along with the estimated target speed, to classify a target using the Transferable Belief Model. When using terrain information, a method is used to take into account the uncertainty of the target position. In addition, a mechanism is added to the Transferable Belief Model to prevent the assignment of belief to outcomes that past behaviour have shown is not possible.*

The improvements are analysed with the use of three simulated scenarios — the results of which show consistently good classification performance. In each scenario, targets have non-constant velocity trajectories.

Keywords: Classification, sensor management, transferable belief model, wireless sensor network.

1 Introduction

This paper presents a new approach for Joint Tracking and Classification (JTAC) with a Wireless Sensor Network (WSN) which we call 'tbmTerrain'. We have improved upon previous work [8] with the use of terrain information, more realistic prior probabilities, and a method for only assigning belief to targets that are feasible. Three scenarios, which are more realistic than the scenario of previous work, are used for a performance evaluation — although due to space limitations, only one scenario is discussed in detail. Targets are modelled on real vehicles and have more complex, non-constant velocity trajectories.

Novel improvements presented in this paper are the use of terrain information and how it restricts target movement, the use of an elliptical area of a terrain map to account for position uncertainty and the subsequent weighted combination of conditional plausibilities related to the terrain coverage within the ellipse, and the addition of a mechanism to prevent the assignment of belief to outcomes that past observations have shown are infeasible. Other improvements to further increase classification performance include the use of more realistic prior probabilities and the smoothing of the estimated target's speed.

An existing approach to JTAC with terrain information, by Powell and Marshall [4], also uses a particle filter and the Transferable Belief Model (TBM). Powell and Marshall do not consider the uncertainty of the target position when using terrain information — this can result in the incorrect assignment of belief when there is enough uncertainty to move the mean of the probability distribution from the true target position over a terrain boundary. However, Powell and Marshall perform multi-target tracking which we do not consider. Approaches that treat road networks as graphs (e.g. Ristic et al. [7]) can allow for targets that travel both on-road and off-road; our approach goes further than this by considering how different types of off-road terrain affect the target movement by differing amounts (e.g. water vs. grass). A notable tracking-only approach by Fosbury et al. [2] uses 'trafficability' terrain information to deflect the target motion towards the direction which provides the least resistance to target motion.

Background information is discussed in Section 2. The problem formulation is presented in Section 3. Section 4 explains the improvements made upon previous work. The results are presented in Section 5. Finally, the conclusions and future work are discussed in Sections 6 and 7 respectively.

2 Background

We utilise the TBM, Intelligence Preparation of the Battlefield (IPB) [13], and a sensor management algorithm by Williams et al. [14] — which we briefly describe in this section along with an overview of our

previous work [8] which we extend in this paper. Due to space limitations only a very brief overview of the aforementioned topics is provided, for more details it is recommended that their respective literature is consulted.

2.1 Transferable Belief Model

The TBM [12], an extension of Dempster-Shafer Theory [1, 9], is used to model subjective beliefs. The TBM does not require an underlying probability model, and can model ignorance, uncertainty, and conflict. Two levels exist within the TBM — the *credal* and the *pignistic* levels. In the credal level, beliefs from agents are assigned to sets of events, beliefs can be updated, and beliefs from multiple agents or times can be fused. In the pignistic level, the *pignistic transform* is used to create probability functions. Both open and closed worlds are supported by the TBM — we use a closed world.

The frame of discernment, Ω, contains all of the expected possible outcomes $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$. One of the outcomes is the true state, $\bar{\omega}$. In a closed world we assume that all of the possible outcomes are in Ω, and the empty set, $\emptyset$, which is used to model conflict is not given any support. An agent quantifies the amount of support it gives to subset of Ω using a *basic belief mass (bbm)*:

$$m : 2^{\Omega} \to [0, 1] \text{ with } \sum_{A \subseteq \Omega} m(A) = 1. \tag{1}$$

The *degree of belief* of A, $bel(A)$, quantifies our belief that $\bar{\omega} \subseteq A$, and is calculated as follows:

$$bel : 2^{\Omega} \to [0, 1] \text{ with}$$
$$bel(A) = \sum_{\emptyset \neq B \subseteq A} m(B) \qquad \forall A \subseteq \Omega, A \neq \emptyset. \tag{2}$$

The *degree of plausibility* of A, $pl(A)$, is calculated as follows:

$$pl : 2^{\Omega} \to [0, 1] \text{ with}$$
$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = bel(\Omega) - bel(\bar{A}), \tag{3}$$

and quantifies how much belief could be transferred to support A. A set of *bbms* is called a *basic belief assignment (bba)*.

Using the *Continuous Transferable Belief Model (cTBM)* [6, 11] it is possible to assign belief to masses conditional on prior probabilities:

$$m(A|x) = \prod_{\omega_i \in A} pl(x|\omega_i) \prod_{\omega_i \in \bar{A}} \left[1 - pl(x|\omega_i)\right], \tag{4}$$

where $m(A|x)$ is the mass assigned to A given x, $\bar{A} = 2^{\Omega} \backslash A$. The plausibility of x for a given class (e.g. ω_i) can be calculated for a 'bell shaped' [11] *pignistic density function*, $Betf$ using:

$$pl(x) = (x - \bar{x})\, Betf(x) + \int_x^{\infty} \left(1 - \frac{\mathrm{d}\bar{a}}{\mathrm{d}a}\right) Betf(a)\ \mathrm{d}a, \tag{5}$$

where $\bar{x}$ is defined as $Betf(x) = Betf(\bar{x})$ and $\bar{x} \leq v \leq x$; v is the mode of $Betf$.

Beliefs from two agents can be fused using various rules of combination. Dempster's rule of combination is a normalised version of conjunctive combination where the mass assigned to $\emptyset$ is always zero. This rule of combination can only be applied to closed worlds. Conflict is redistributed via normalisation — it does not take into account which *bbms* caused the conflict.

A more sophisticated method for fusing *bbas* is PCR5 [10]. It is a more correct way of redistributing conflict than Dempster's rule as it only redistributes mass to outcomes involved in conflict.

The pignistic transform can then be calculated as follows [6]:

$$BetP\{\omega_i|x\} = \sum_{A:\omega_i \in A} \frac{1}{|A|} \frac{m(A|x)}{[1 - m(\emptyset|x)]}. \tag{6}$$

$BetP$ can then be used to make a decision as to what the target is.

2.2 Intelligence Preparation of the Battlefield

Our method for incorporating terrain information into the JTAC is inspired by IPB [13]. IPB is a methodology for analysing threats in a geographic area. As part of the IPB process, terrain is analysed and labelled according to how it restricts unit movement. The intelligence used in terrain analysis includes, but is not limited to, information about vegetation, obstacles, terrain surface type, slope, and weather.

The IPB process divides terrain into three classes to indicate how restrictive movement will be — they are called 'Go', 'Slow Go', and 'No Go'. 'Go' terrain does not restrict unit movement. Terrain that is classed as 'Slow Go' provides some restriction to unit movement. 'No Go' terrain is considered to be severely restricted.

2.3 Sensor Management

We utilise a sensor management approach by Williams et al. [14] that selects multiple sensors for each time step. The planning algorithm is *non-myopic*, i.e. it plans sensor usage for more than just the a single time step. Sensor usage is determined for a rolling horizon length of N time steps. This approach results in a lower accrued communications cost compared to selecting a single sensor at each step that either minimises the expected distance to the target or provides the greatest reduction in the uncertainty of the estimate of the target's kinematic state.

At each time step, a plan is created for the N time steps. One or more sensors are selected for each step that either maximise the reduction in uncertainty whilst not exceeding a communications cost constraint for the entire N time step plan, or minimise the communications cost for plan whilst meeting a minimum constraint on the entropy of the probability distribution function of the target state; our work uses the former approach. A dynamic programming approach is used to balance this trade-off between the the quality of information obtained and the cost of obtaining it.

A leader node is used for planning and target tracking; the leader node may change at at each time step, before observations are taken. The leader node receives measurements from all active sensors, including itself, and combines them within the particle filter — this is discussed further in Section 3.

2.4 Joint Tracking and Classification with a WSN

Previous work combined a JTAC approach by Powell et al. [5] with the sensor management approach by Williams et al. [14].An overview of the JTAC approach can be found in Figure 1. At each time step the planning algorithm by Williams et al. is used to select the sensors, a change in leader node occurs (if required), and then sensors send observations (range measurements) to the leader node. The leader node combines the range measurements and uses them to update the particle filter. The particles are then used in the classification process. A *bba* is created using the speed of each particle. The *bbas* for each particle are fused using the closed world conjunctive combination, the resultant *bba* is then fused with the *bba* for previous time steps, and $BetP$ is then calculated from the *bbas*. The particle weights are then conditioned as follows:

$$\bar{w}^i_{k+1} = \begin{cases} \mathcal{N}\left(\mathbf{S}\left(\boldsymbol{x}^i_k\right); \mu, \gamma\sigma\right) & P\left(\omega_j\right) > \beta, \\ 1 & P\left(\omega_j\right) \leq \beta \end{cases}. \quad (7)$$

$\mathbf{S}\left(\boldsymbol{x}^i_k\right)$ extracts the speed from particle $\boldsymbol{x}^i_k$, μ and σ are the mean and standard deviation of ω_j, γ is used to reduce the impact of $\mathcal{N}$, and ω_j is the class with the highest classification probability. This adjusts the particle weights to reflect the current belief of the TBM once it shows that the target has been classified as class ω_j with a sufficient probability β.

3 Problem Formulation

This paper builds upon previous work [8], and as such much of the terminology and models are similar or the same — some of which is repeated in this section along with new terminology.

The same sensor selection algorithm, by Williams et al. [14], is utilised. With the exception of resampling, the particle filter code remains unchanged. The sensor and communications costs models also remain unchanged.

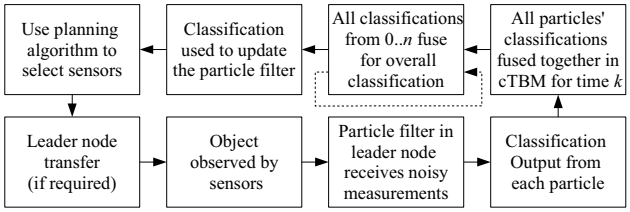


Figure 1: Data flow of previous work [8].

The non-linear target trajectories used in this paper are more realistic than that of previous work — targets slow down at turns and when travelling on more restrictive terrain. Each target class, $\omega_i \in \Omega$, is based on a real unit; where possible, target characteristics (such as maximum speed) have been obtained from authoritative literature. The target is modelled by a white noise acceleration model [3], where the target state, $\boldsymbol{x}_k$, at time k, is assumed to be $\boldsymbol{x}_k = [pos_x\ vel_x\ pos_y\ vel_y]^T$, and its dynamics are updated by

$$\boldsymbol{x}_{k+1} = \mathbf{F}\boldsymbol{x}_k + \boldsymbol{w}_k, \quad (8)$$

where

$$\mathbf{F} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

and $\boldsymbol{w}_k \sim \mathcal{N}\left(\boldsymbol{w}_k; 0, \mathbf{Q}\right)$, and T is the sampling interval. The covariance of the process noise is modelled as follows:

$$\mathbf{Q} = q \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} & 0 & 0 \\ \frac{T^2}{2} & T & 0 & 0 \\ 0 & 0 & \frac{T^3}{3} & \frac{T^2}{2} \\ 0 & 0 & \frac{T^2}{2} & T \end{bmatrix}, \quad (10)$$

where q is the constant diffusion strength.

A non-linear sensor model is used, the measurement obtained by sensors s_k at time k is calculated by

$$\boldsymbol{z}^s_k = h(\boldsymbol{x}_k, s) + \boldsymbol{v}^s_k \quad (11)$$

where $\boldsymbol{v}^s_k \sim \mathcal{N}\left(\boldsymbol{v}^s_k; 0, \mathbf{R}^s\right)$, it is independent for each s and independent of $\boldsymbol{w}_k$, $\mathbf{R}^s$ is the measurement noise covariance matrix for sensors s, and

$$h(\boldsymbol{x}_k, s) = \frac{a}{||\mathbf{L}\boldsymbol{x}_k - \boldsymbol{y}^s||_2^2 + b}. \quad (12)$$

a and b are used to model the Signal to Noise Ratio (SNR) ratio of sensors s. The measurement from each sensor has additive Gaussian noise with variance R.

A linearisation of the above equation about a nominal point, $\boldsymbol{x}^0$, is

$$\mathbf{H}^s\left(\boldsymbol{x}^0\right) = \frac{-2a}{(||\mathbf{L}\boldsymbol{x}_k - \boldsymbol{y}^s||_2^2 + b)^2}(\mathbf{L}\boldsymbol{x}^0 - \boldsymbol{y}^s)^T\mathbf{L}, \quad (13)$$

it is used in the particle filter, and also used by Williams et al. to reduce the complexity of the planning stage.

Any sensor can communicate with any other sensor. Multi-hop communication is used to minimise communication costs. The total communication cost between nodes is the sum of the direct communications costs between between the nodes along the shortest path.

As with Williams et al. [14], we use Sequential Importance Sampling (SIS) with resampling at each time step for tracking. Sensor measurements are fused within the particle filter. The probability density function (pdf) of the target's kinematic state, $\boldsymbol{x}_k$, conditional on measurements received up to and including time k, $\boldsymbol{z}_{1:k}$, is approximated by

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}) \approx \sum_{i=1}^{N_p} w_k^i \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^i), \tag{14}$$

where N_p is the number of particles, w_k^i is the weight of i^{th} particle $\boldsymbol{x}_k^i$, at time k, and $\delta(.)$ is the Dirac delta function. The same distribution is calculated for the next time step by sampling from

$$q\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k^i, \boldsymbol{z}_{k+1}\right) = \mathcal{N}\left(\boldsymbol{x}_{k+1}; \hat{\boldsymbol{x}}_{k+1}^i, \mathbf{P}_{k+1}^i\right) \tag{15}$$

where

$$\hat{\boldsymbol{x}}_{k+1}^i = \mathbf{F}\boldsymbol{x}_k^i + \mathbf{K}_{k+1}^i\left[\boldsymbol{z}_{k+1} - \boldsymbol{h}(\mathbf{F}\boldsymbol{x}_k^i, s)\right] \tag{16}$$

$$\mathbf{P}_{k+1}^i = \mathbf{Q} - \mathbf{K}_{k+1}^i \mathbf{H}^s\left(\mathbf{F}\boldsymbol{x}_k^i\right)\mathbf{Q} \tag{17}$$

$$\mathbf{K}_{k+1}^i = \mathbf{Q}\left\{\mathbf{H}^s\left(\mathbf{F}\boldsymbol{x}_k^i\right)\right\}^T \cdot \cdot\left[\mathbf{H}^s\left(\mathbf{F}\boldsymbol{x}_k^i\right)\mathbf{Q}\left\{\mathbf{H}^s\left(\mathbf{F}\boldsymbol{x}_k^i\right)\right\}^T + \mathbf{R}^s\right]^{-1}. \tag{18}$$

$\boldsymbol{z}_{k+1}$ are the observations at time $k+1$. $\boldsymbol{h}(\cdot, s)$ is a vector-valued function of length equal to $|s|$. $\boldsymbol{z}_{k+1}$ is a vector of length also equal to $|s|$. Each particle is conditioned on all of the measurements received at the current time step.

$$w_{k+1}^i = c w_k^i \frac{p\left(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}^i\right) p\left(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i\right)}{q\left(\boldsymbol{x}_{k+1}^i|\boldsymbol{x}_k^i, \boldsymbol{z}_{k+1}\right)}, \tag{19}$$

where

$$p\left(\boldsymbol{z}_{k+1}|\boldsymbol{x}_{k+1}^i\right) = \mathcal{N}\left(\boldsymbol{z}_{k+1}; \boldsymbol{h}(\boldsymbol{x}_{k+1}^i, s), \mathbf{R}^s\right), \tag{20}$$

and c is a normalisation constant such that $\sum_{i=1}^{N_p} w_{k+1}^i = 1$.

By moment-matching the pdf to a Gaussian distribution, the mean and covariance are calculated as

$$\boldsymbol{\mu}_k = \sum_{i=1}^{N_p} w_k^i \boldsymbol{x}_k^i \tag{21}$$

and

$$\mathbf{P}_k = \sum_{i=1}^{N_p} w_k^i \left(\boldsymbol{x}_k^i - \boldsymbol{\mu}_k\right)\left(\boldsymbol{x}_k^i - \boldsymbol{\mu}_k\right)^T. \tag{22}$$

The sensor selection algorithm, by Williams et al. [14], is largely the same. It has been modified to only use sensors a minimum distance, $\min_d$, away from the expected position of target. This prevents the sensor measurements from being saturated when using a much higher SNR, which disrupts the target tracking.

Terrain *types* such as road and grass are grouped into terrain *classes*; the set of terrain classes, $\mathcal{T} = \{\text{Go}, \text{SlowGo}, \text{NoGo}\}$ is the same as those used in IPB. However, our approach is not limited to three terrain classes, grouping terrain types into these classes provides a sufficient level of abstraction to simplify calculations yet still provide adequate richness for the classification process.

We create terrain maps based on the IPB approach to terrain analysis. A map, represented by a bitmap image, is segmented by terrain *type* (e.g. road). Each segment of the same terrain type has the same colour. For each target class, a mapping is created from terrain type to terrain class — this allows each terrain type to affect target classes in different ways without requiring a separate map for each target class. Examples of terrain maps can be found in Section 5.

4 Improvements

The following improvements can by divided into two categories — those designed to improve classification performance, and those designed to improve feedback from classification to tracking.

4.1 Classification

The scenarios used here are more realistic than that of previous work; for this reason, a more sophisticated method of classification is required. This is achieved by using a number of improvements to the method for calculating conditional belief masses; these improvements consist of smoothing the estimated target speed, using terrain information, using a mechanism to ignore target classes are that no longer feasible, and using more realistic prior probabilities. This results in a more accurate classification.

4.1.1 Target speed

The estimated speed of the target is smoothed using:

$$s_k = \frac{\mathbf{S}\left(\boldsymbol{\mu}_k\right) + \mathbf{S}\left(\boldsymbol{\mu}_{k-1}\right) + \ldots + \mathbf{S}\left(\boldsymbol{\mu}_{k-\mathcal{W}+1}\right)}{\mathcal{W}} \tag{23}$$

where $\boldsymbol{\mu}$ is calculated using Equation 21, and $\mathcal{W}$ is the window size. A smoothed estimate is required because the estimated speed for a single time step is typically too noisy for accurate classification — especially when tracking a target with non-linear dynamics. Smoothing using Equation 23 will result in some bias, but with a sufficiently small value of T and $\mathcal{W}$ it does not negatively impact classification.

4.1.2 Utilising Terrain Information

Terrain information is used in the classification process to take into account how the underlying terrain may hinder a target's progress. An uncertainty ellipse at N_σ standard deviations is used to calculate the underlying terrain — this provides a more robust classification than using a single point at $\mathbf{L}\mu_k$ as the terrain may vary over a small area and there will usually be some inaccuracy as to the exact location of the target. The proportions of each terrain class within the ellipse are stored in f, a $|\mathcal{T}|$ by $|\Omega|$ array, such that $\sum_{\tau\in\mathcal{T}} f[\tau,\omega_i] = 1 \ \forall \ \omega_i \in \Omega$. The notation $f[\tau,\omega_i]$ is used to denote the proportion of the ellipse coverage that is of the terrain class τ for the target class ω_i.

4.1.3 Ignoring Infeasible Target Classes

One problem that exists with using the TBM to recursively fuse *bbms* is that belief can be assigned to a target class that from earlier behaviour is obviously not feasible — Powell et al. [5] refer to this as a "lack of intelligent memory." We have modified the TBM to produce this expected behaviour by calculating which targets are feasible, and only allowing conditional plausibility to be assigned to masses that are only for feasible targets.

At each time step k, the feasible classes $\mathcal{F}_k$ are calculated as follows:

$$\mathcal{F}_k = \begin{cases} \left(\bigcup_{\substack{\forall\omega\in\Omega, \\ \mathcal{S}(\omega)\rho\geq s_k}} \omega \right) \bigcap \mathcal{F}_{k-1} & \text{if } k \geq \iota; \\ \Omega & \text{otherwise,} \end{cases} \tag{24}$$

where $\mathcal{S}(\omega)$ is the maximum speed of target class ω, and ρ is the speed tolerance; ρ is due to a small amount of bias from Equation 23 and inaccuracy when estimating $\boldsymbol{\mu}_k$. If at any time k, $\mathcal{F}_k = \emptyset$, then $\mathcal{F}_k$ is reset to $\mathcal{F}_k = \Omega$. $\mathcal{S}(\omega)$ is calculated as follows:

$$\mathcal{S}(\omega) = \begin{cases} \mathcal{S}_{\text{Go}}(\omega) & \text{if } f[\text{Go},\omega] > 0; \\ \mathcal{S}_{\text{SlowGo}}(\omega) & \text{if } f[\text{Go},\omega] = 0 \text{ and} \\ & \quad f[\text{SlowGo},\omega] > 0; \\ \mathcal{S}_{\text{NoGo}}(\omega) & \text{otherwise.} \end{cases} \tag{25}$$

$\mathcal{S}_{\text{Go}}(\omega)$, $\mathcal{S}_{\text{SlowGo}}(\omega)$, and $\mathcal{S}_{\text{NoGo}}(\omega)$ are the maximum speeds for target class ω whilst travelling over terrain classes Go, Slow Go, and No Go respectively. $\mathcal{F}_k$ is then used when calculating conditional plausibility to ignore target classes that are infeasible (See Equation 26).

4.1.4 More Realistic Prior Probabilities

A Gaussian pignistic density was used in previous work; this was sufficient for a constant velocity target trajectory but it is not suitable for more realistic target trajectories. The pignistic density function, $Betf$, used in this paper is asymmetric; there is a slow increase in $Betf(y)$ from $y = 0$ to the expected maximum speed of the target, and then a sharp decline from the expected maximum speed toward ∞. Examples of this can found in Figures 2(b)–2(d).

4.1.5 Combining Plausibilities

In order to create *bbms* from the estimated target speed and the underlying terrain, it is necessary to combine conditional plausibilities. Conditional plausibility is calculated for each terrain and target class given the estimated target speed; it is then combined for each target class resulting in a conditional plausibility for each target class. Using the combined conditional plausibility results in a *bba* that takes into account not only target speed and the terrain the target is travelling over, but also the uncertainty of the target's kinematic state.

The conditional plausibilities are combined as follows:

$$pl_{\text{tbmTerrain}}(\omega|s_k) = \begin{cases} \sum_{\tau\in\mathcal{T}} f[\tau,\omega]\, pl_\tau(\omega|s_k) & \text{if } \omega \in \mathcal{F}_k; \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

where $pl_\tau(\omega|s_k)$ is the conditional plausibility calculated for target ω using priors for the terrain class τ, and speed s_k. $pl_{\text{tbmTerrain}}$ can then be used to create conditional *bbms* for time k using Equation 4.

Any remaining mass that would usually be assigned to the empty set is added to the most uncertain (yet still feasible) set, Υ. This is better than assigning the remaining mass to Ω if some target classes are infeasible. This is a also a better method than normalising the *bba* as it may give a false indication of the true outcome. Once the *bba* has been calculated for time k, it is fused with the existing *bba* which was created using the target state estimate for times 1 to $k-1$. The resulting *bba* is used to calculate $BetP$.

4.2 Feedback to the Particle Filter

In addition to improving the classification process, feedback from classification to the particle filter has been improved. A new method for this feedback is presented here. The aim of this new method is to use information from the classification process to reduce the uncertainty of the target position.

When required, feedback to the particle filter is achieved by updating the covariance matrix, $\mathbf{P}_k$, and then sampling a new set of particles from a multivariate Gaussian distribution parameterised by $\boldsymbol{\mu}_k$ and the updated covariance matrix. If there is more than one terrain class covered by the uncertainty ellipse then the parts of the covariance matrix related to position are reduced, resulting in a conditioned covariance matrix, $\bar{\mathbf{P}}_k$. Assuming that the state vector is $[pos_x\ vel_x\ pos_y\ vel_y]^T$, $\bar{\mathbf{P}}_k$ is calculated as follows:

$$\bar{\mathbf{P}}_k = \begin{bmatrix} \kappa & \kappa & \kappa & \kappa \\ \kappa & 1 & \kappa & 1 \\ \kappa & \kappa & \kappa & \kappa \\ \kappa & 1 & \kappa & 1 \end{bmatrix} .* \mathbf{P}_k, \tag{27}$$

where κ is the conditioning factor, and $.*$ is the notation used to denote element-by-element multiplication (the same notation used by Matlab). If there is

only one terrain class covered by the uncertainty ellipse then $\bar{\mathbf{P}}_k = \mathbf{P}_k$. After calculating $\bar{\mathbf{P}}_k$, resampling is performed. Each particle, $\boldsymbol{x}_k^i$, and it's respective weight, w_k^i, is sampled and calculated respectively, as follows:

$$\boldsymbol{x}_k^i \sim \mathcal{N}\left(\boldsymbol{x}_k^i; \boldsymbol{\mu}_k, \bar{\mathbf{P}}_k\right),\ w_k^i = 1/N_p, \tag{28}$$

where $\boldsymbol{\mu}_k$ is calculated using Equation 21. We call the above resampling method 'parametric resampling'.

Parametric resampling does not preserve the potentially complex and non-Gaussian nature of the particle distribution. This makes it inferior to systematic resampling, but it does provide a way to 'reshape' the distribution in a straightforward manner. A hybrid approach is used that results in a compromise between the two resampling methods: parametric resampling is used when conditioning is required, systematic resampling is used in all other cases.

As the new resampling method has the potential to remove some of the detail from the particle distribution, the particle filter has been modified to resample less often. Resampling is performed at each time step only if the estimated effective sample size is lower than a threshold, $\bar{N}_{eff} < N_{thr}$. This results in conditioning taking place less often, but it aims strikes a balance between providing feedback from classification to tracking, and preserving the complexity of the particle distribution.

5 Results

Three scenarios (A, B, and C) were used to test the improvements presented. Due to space limitations only Scenario A is discussed in detail, although results are shown for all scenarios; tbmTerrain, is compared with our previous approach and the approach of Williams et al. [14]. As Williams et al. and previous work, horizon lengths of 5, 10, and 25 are used, and 100 Monte Carlo trials were used for each combination of parameters.

A different sensor layout is used for each of the 100 Monte Carlo trials in each scenario, and the same set of 100 sensor layouts is used for each combination of parameters. A single target trajectory was used for each scenario as opposed to a different target trajectory for each trial (as per Williams et al.) — this is because random variations in the target's kinematic state could have resulted in a trajectory that travels too fast or over the wrong terrain type. The potential pitfalls of using a single target trajectory for each scenario have been mitigated by using 100 different sensor layouts. Sensor layouts were generated using a uniform distribution.

The terrains created were based on real geographic areas using Ordnance Survey map data. In Scenario A (Section 5.1), an amphibious light tank drives along a road, then over a river and grass. In Scenario B, a car drives up to a roundabout, around it, then away from it — remaining on the road surface at all times. In Scenario C, a main battle tank initially travels along a road, and then off-road. The assumed effect of the each terrain type on the performance of each target class can be found in Table 1. Ω for Scenario A contains all 6 target classes, Ω for scenarios B and C contain all of the classes *except* the bicycle.

Target Class	Road	Grass	Water	Buildings	Marsh	Trees	Steep Land
Pedestrian	G	G	N	N	S	G	S
PT76	G	G	S	N	S	S	N
T62	G	G	N	N	N	S	N
Challenger 1	G	G	N	N	N	S	N
ZiL 41041	G	S	N	N	N	S	N
Bicycle	G	S	N	N	N	S	N

Table 1: The assumed best performance of each target class for each terrain type. The letters 'G', 'S', and 'N' indicate Go, Slow Go, and No Go respectively .

The results for Scenario A are shown as the Scenario is more complex than that of Scenarios B and C. In Scenario B the target remains on a single terrain type (Road). In Scenario C, the target travels on a less complex route compared to Scenarios A and B. For classification performance, there is a larger improvement in classification performance for tbmTerrain compared to previous work for Scenarios B and C.

Figure 2 shows the $Betf$ that is calculated for each target class and terrain class, $Betf$ for our previous approach is also shown.

In all scenarios, N_{thr}, equivalent to 66% was used, with $T = 0.25$, $q = 50$, and $\min_d = 50$ m.

For all of the simulations that used our previous approach, a condition factor of $\gamma = 3.5$ was used. Instead of creating a *bba* for each particle, creating N_p *bbas* for each time step, we have created a single *bba* at each time step using $\boldsymbol{\mu}_k$ in order to reduce computational requirements.

The classification accuracies of our new approach compared to our previous approach can be found in Figure 4. Our new approach results in consistently good classification performance which is not the case for the approach taken in previous work.

5.1 Scenario A

Scenario A consists of an amphibious light tank moving from the top right to the bottom of a 630m by 436m region (See Figure 3). The target travels over road, grass, and water. 20 sensors are used in each Monte Carlo trial, with $R = 1.75$ and $a = 1500000$. Each simulation is 493 time steps long. The tbmTerrain trials used the following parameter values: $\iota = 30$, $\mathcal{W} = 7$,

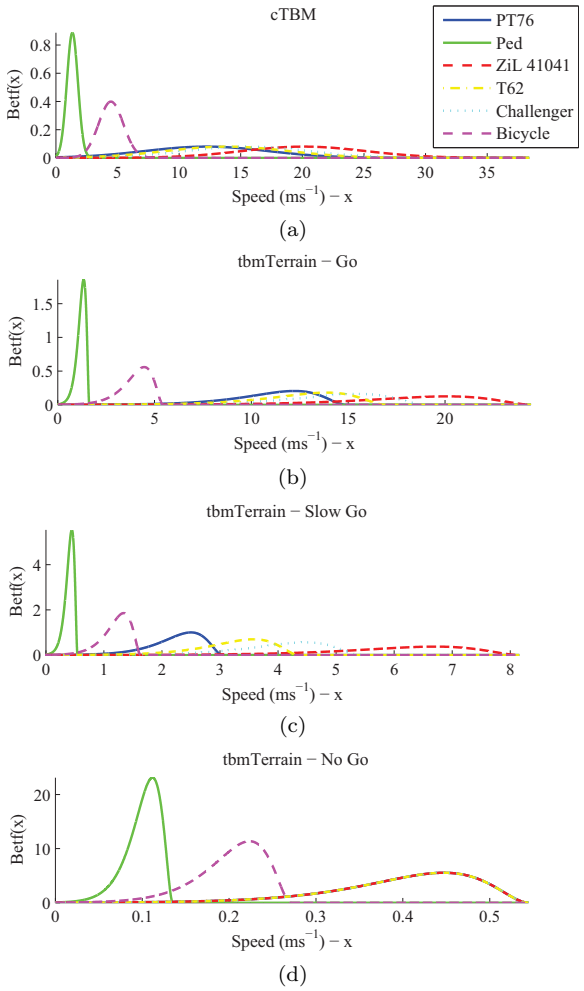


Figure 2: $Betf$ for our previous approach (a) and tbmTerrain terrain classes 'Go' (b), 'Slow Go' (c), and 'No Go' (d).

$\rho = 1.75$, and $N_\sigma = 5$ — these values were determined experimentally.

A comparison of classification performance can be found in Figure 4(a). The tracking performance is similar across all approaches for the same horizon length; this may be due to a combination of our new approach using an inferior resampling method when feedback from classification to tracking occurs, and using a fairly simplistic method for providing the feedback. As tracking performance is similar for all approaches with the same horizon length, the sensor usage is similar too, resulting in similar communications costs. This effect was seen in all three scenarios.

6 Conclusions

We have extended previous work [8] to produce a more sophisticated WSN JTAC algorithm. Most of the work presented has focused on improving classification performance — which includes the use of terrain information in the classification process. Our new approach to WSN JTAC has been evaluated along with the approach of our previous work and the tracking-only approach of Williams et al. [14]. Details of the improvements have been discussed in Section 4.

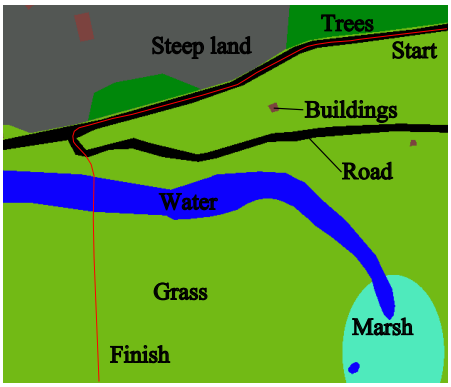
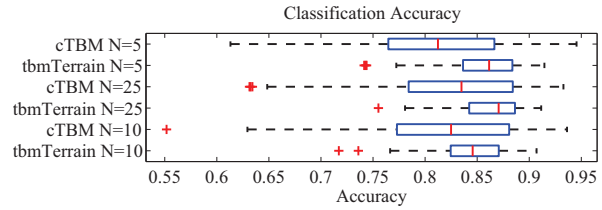
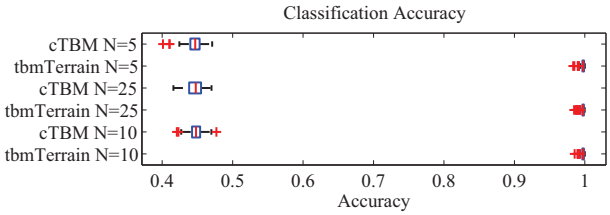


Figure 3: Target trajectory and terrain for Scenario A. The red line is the target trajectory.

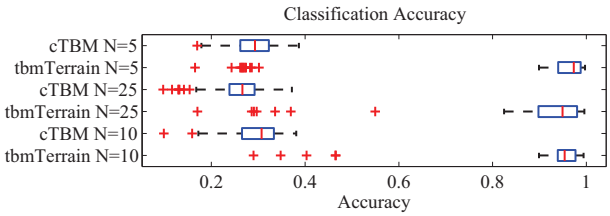
This paper has presented various novel methods to improve classification with the TBM. One of which is the use of terrain information related to how it restricts each target class's movement. Another is the use of an elliptical area of the terrain map to account for position



(a) Scenario A



(b) Scenario B



(c) Scenario C

Figure 4: Classification accuracies for Scenarios A, B, and C. Each box plot is created from the mean value of $BetP(X)$ over time for each simulation, where X is the ground truth target class. 'cTBM' denotes our previous approach.

uncertainty and the subsequent weighted combination of conditional plausibilities related to the terrain coverage within the ellipse. We have also added a memory to the TBM to prevent it from assigning plausibility (and hence belief) to outcomes that earlier estimates have shown is not feasible.

The results have shown a consistently good classification performance for our new approach; this is due to the improvements presented in Section 4. It appears that the new method of feedback from the TBM to the particle filter does not improve tracking accuracy. A consequence of the poor feedback from the TBM to the particle filter is that the target tracking for all three approaches evaluated have a similar performance (for the same horizon length), resulting in similar sensor utilisation levels and associated communications costs.

7 Future Work

There are a number of potential further improvements. Firstly, the uncertainty of the target's position could be more effectively used than a simple uncertainty ellipse; since the ellipse represents a Gaussian distribution cut-off at a set threshold.

An effective method of providing feedback from the TBM to the particle filter remains to be found, the poor performance of this step limits the overall performance of both the approach presented here and that of previous work. Once a more effective method for feedback has been found, it will be possible to see if the reduction in the uncertainty of the kinematic state estimate reduces the sensor usage and associated communications costs.

8 Acknowledgements

The authors would like to thank Peter Talbot-Jones, EADS Innovation Works for helping to create the scenarios, and ARCCA, Cardiff University for providing the CPU time to run the simulations. The first author would like to thank EADS for funding the research.

References

[1] A. P. Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society*, 30(2):205–247, 1968.

[2] A. M. Fosbury, T. Singh, J. L. Crassidis, and C. Springen. Ground target tracking using terrain information. In *Proceedings International Conference on Information Fusion 2007*, pages 1–8, 2007.

[3] X. R. Li and V. P. Jilkov. Survey of maneuvering target tracking. part I: Dynamic models. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1333–1364, Oct 2003.

[4] G. Powell and D. Marshall. Joint tracking and classification of nonlinear trajectories of multiple objects using the transferable belief model and multi-sensor fusion framework. In *Proceedings International Conference on Information Fusion 2005*, volume 2, 2005.

[5] G. Powell, D. Marshall, P. Smets, B. Ristic, and S. Maskell. Joint tracking and classification of airbourne objects using particle filters and the continuous transferable belief model. In *Proceedings International Conference on Information Fusion 2006*, Jul 2006.

[6] B. Ristic and P. Smets. Belief function theory on the continuous space with an application to model based classification. *Proceedings of Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU*, pages 4–9, 2004.

[7] B. Ristic, S. Arulampalam, and N. Gordon. Terrain-aided tracking. In *Beyond the Kalman Filter*, chapter 10. Artech House, 2004. ISBN 1-58053-631-x.

[8] M. Roberts and D. Marshall. Using classification to improve wireless sensor network management with the continuous transferable belief model. In Edward M. Carapezza, editor, *Unmanned/Unattended Sensors and Sensor Networks V*, volume 7112, page 711204. SPIE, 2008.

[9] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, NJ, 1976.

[10] F. Smarandache and J. Dezert. Information fusion based on new proportional conflict redistribution rules. *Proceedings International Conference on Information Fusion 2005*, 2:907–914, july 2005.

[11] P. Smets. Belief functions on real numbers. *International Journal of Approximate Reasoning*, 40 (3):181–223, 2005.

[12] P. Smets and R. Kennes. The Transferable Belief Model. *Artificial Intelligence*, 66(2):191–234, 1994.

[13] United States. *Intelligence Preparation of the Battlefield: Field Manual 34-130*. Headquarters, Dept. of the Army, Washington, D.C., Jul 1994.

[14] J. L. Williams, J. W. Fisher III, and A. S. Willsky. Approximate dynamic programming for communication-constrained sensor network management. *IEEE Transactions on Signal Processing*, 55(8):4300–4311, Aug 2007.