Joint Tracking And Classification of Nonlinear Trajectories of Multiple Objects Using the Transferable Belief Model and Multi-Sensor Fusion Framework

Gavin Powell
Geometric Computing and Computer Vision Group
School of Computer Science
Cardiff University
Cardiff
UK.
g.r.powell@cs.cf.ac.uk

David Marshall
Geometric Computing and Computer Vision Group
School of Computer Science
Cardiff University
Cardiff
UK
dave.marshall@cs.cf.ac.uk

Abstract - In this paper we present our findings of investigating non-linear multi-target tracking techniques when jointly used with object classification. The Transferable Belief Model (TBM) is utilised in the multi-target evaluation, data association and target classification stages. A particle filter is used to track each of the targets and uses a motion model that is relevant to the classification given to that target. The targets are classified based upon their motion throughout the scene and their land based position. We show how this system can deal with prior knowledge and lack of knowledge. Situations, with data of this type, regularly occur in real world scenarios and we think it is very important that any system must be able to cope well to such situations. Bayesian and regular DST methods have shortcomings when dealing with such scenarios. We show that the TBM approach can be generally more computational tractable and more robust.

Keywords: Tracking, TBM, particle filter, classification.

1 Introduction

Fusing data from multiple sensors for the tracking and classification of objects has had increasing attention as techniques and technology have evolved. To improve on the robustness of the classification and track, systems try to use information from the sensors as well as output from other modules within the system. This can involve the use of feedback loops [1] between each of the stages to give a more coherent tracking and classification system, or a completely integrated system to jointly track and classify [2]. When multiple targets are involved, then data association also needs to be accounted for within the system [3,4,5,6]. A simple feedback mechanism is shown in Figure 1. Due to a lack of data being made available a synthetic scenario has been devised to evaluate the algorithms that we have created. It is a land based target tracking and classification system that utilises multiple sensor measurements. Each sensor tracks one, and only one target at a time, reporting on the position of that target. The sensors are not able to communicate with each other and so cannot determine if they are viewing the same target, thus multiple sensors can be reporting upon the same target. This means that the problem is threefold and our system needs to be able to associate sensor measurements to targets, evaluating the number of targets present, track those targets and classify the targets.

The association of sensors to targets uses a TBM approach which evaluates the possible sensor to target configurations and decides on the correct one through conflict revealed through the TBM [7], not only will this approach discover the correct association but will also identify the number of targets to track. The tracking of the targets takes place within a Particle Filter (PF), which is linked to the TBM classification stage and uses the classification of each target to decide on the correct motion model to use. We believe that present motion models used within tracking are poor [8] and need to be improved greatly; this will ultimately produce a complex non-linear mechanism. Due to this being the future direction of our
work we are using the PF for tracking from the outset. The classification stage uses the particle filter outputs, which are target motion parameters, as well as information from a terrain map. The terrain map will identify the type of terrain that the target is supposedly placed upon. The paper has taken inspiration from previous work we have done [9,10] as well as work by various others on Joint Tracking and Classification (JTC) [1,2,8,11]. We add to this approach through the introduction of PF’s to allow for more complex non-linear tracks to be considered. In section 2 we look at the technologies used within the system. In section 3 a more detailed explanation of the system is given. In section 4 results for test scenarios are shown and discussed, and finally in section 5 conclusions are drawn as to the performance of the system.

2 Multiple Target JTC Technologies

Multiple target tracking problems have classically used extremely computationally expensive methods such as joint probabilistic data association or multiple hypothesis trackers amongst others [3,4,5,6]. We have implemented and adapted a different approach to these purely probabilistic methods using quantified beliefs based on belief functions [2]. This allows for a more tractable algorithm for finding the number of targets and the sensor to target relationship.

Tracking of targets tends to use iterative predict and update to construct a Probability Density Function (PDF) for the target parameters. The method used is dependent on the target to be tracked, or rather the parameters used in tracking that target. If tracking is based on motion then the method used will depend upon if the motion model is linear or not. Many papers are present on tracking linear models using the optimal Kalman Filter (KF) approaches [12, 13], but many targets have a non linear motion and other approaches are needed. Some of the more popular approaches to this are the Extended Kalman Filter (EKF)[14,15] and the PF [16,17,18] which approximate the posterior PDF. The PF uses a set of ‘particles’ of differing weight to achieve this approximation.

Classification of objects using statistical approaches often takes either Bayesian [19] or Dempster-Shafer based methods [20], of which the TBM is an extension. Dempster Shafer allows for incomplete and uncertain evidence to be used [5,20] and allows its beliefs to be easily updated when new and contrary evidence is made available. The Bayesian approach can be greatly affected by its choice of priors, when trying to model incomplete or uncertain evidence, often to its detriment. The Bayesian approach, though, does avoid scaling issues that have been shown to be apparent within Dempster Shafer techniques [19]. Advances have been made to ensure that Dempster Shafer can be computationally efficient and its problems of scalability can be greatly reduced [21]. It is our belief that the TBM that we employ provides a greater degree of flexibility through its ability to cope with both complete and incomplete knowledge, and the benefits of this added flexibility outweigh the additional resources needed with the set based approach.

3 The JTC System

The use of the TBM and PF’s within our system has been influenced by previous work on JTC systems [2, 8,12,11]. We have previously used the TBM for the fusion of data and classification [9,10]. From these experiences we found that the TBM provided a flexible platform for use in fusing data and classifying targets. Previous work in using the TBM for JTC centred around the use of a Kalman Filter [2] for linear tracking. The motion of objects generally is non-linear and the use of a Kalman Filter would be inappropriate. Partnering the TBM with a non-linear tracker, such as the PF, allows for non-linear motion models to be utilised.

3.1 System Data

The sensors that are used within this scenario are able to detect a target and report on its position, with its position being constricted to a grid of cells overlayed on the map. The sensors do not report on differing targets over time and will only track one individual target. These sensors are unable to communicate with each other and so it is never known, a prior, the number of distinct targets that all of the sensors are looking at or if sensors are looking at the same target. This problem becomes more apparent when it is considered that the sensors all have an an error associated with their positional measurement of the target. As well as the information that is provided by the sensor a terrain map is available which will provide a terrain type at the position that the sensor is reporting a target as can be seen in Figure 2. This information is used to track and classify the targets, with the inclusion of signifying a false track, or a target of unknown type.

Figure 2. Sensor Target Detection on Terrain Map

3.2 Data Association in the TBM

Previous work by Smets [7] and Schubert [22] on data association using the TBM was applied to a simplistic scenario where targets could only be present at one of two positions. The number of targets present was found by minimising the conflict given for each possible sensor to
target relationship, where conflict is a measure of ‘correctness’ for each hypothesis of sensor to target relationships. Once the number of targets was known their positions were found by combining the evidence from the sensors. We have extended this approach to be able to manage target positions at any discrete position on the map. The positions on the map are defined as cells with a resolution that is dependant on the required accuracy of the system. Sensors are also prone to error when reporting the position of their associated target and so we have incorporated this possibility into the TBM data association. The TBM will evaluate which sensors are looking at which targets, with the inclusion of multiple sensors looking at the same target but with some error in their measurement.

3.2.1 The TBM

The TBM represents quantified beliefs based on belief functions. It works at a dual level where beliefs are entertained within the credal level and decisions, based upon these beliefs, are made in the pignistic level. The credal level receives information and disperses it to the relevant sets and allows for information to be combined, and the pignistic level performs a pignistic transform from which decisions can be made. It extends from the basic Dempster Shafer Theory[23] in several ways. It allows open and closed world assumptions to be made so that something outside of the current knowledge set can be considered, or rather accounted for, through the use of unnormalised belief functions. This is important within our classification models as we always want to be able to account for the possibility of the target being outside of our dataset of current possible matches. The TBM does not require the user to provide an underlying probability model, but it has the flexibility to incorporate one if available, again this shows the ability of the TBM approach to work with incomplete and uncertain data. Decisions can also be made based on the beliefs, rather than just presenting upper and lower limits. For a full review of the TBM see the work by Smets [24,25,26,27].

3.2.1.1 TBM Overview

The basic belief assignment, $m$, where $m(A)$ is the proportion to which all available and relevant evidence supports $A$ and only $A$ and does not imply support of any sub sets of $A$. The sum of the basic belief assignments must equal unity such

$$\sum_{A \subseteq H} m(A) = 1$$

(1)

where $H$ is the set of all possible values of some variable we are examining. The combination of two basic belief assignments $m_1$ and $m_2$ over $H$, where the basic belief assignments are provided by two distinct pieces of evidence, $m_{02}$ is defined by

$$m_{02}(A) = \sum_{B \cap C = A} m_1(B)m_2(C), \forall A \subseteq H$$

(2)

The pignistic transform that is used to make decisions is defined as:

$$\text{BetP}^e(A) = \sum_{A \subseteq A \subseteq H} m^e(A)\sum_{C \subseteq A \subseteq H} [1 - m^e(C)], \forall h \in H$$

where $|A|$ is the number of elements $H$ in $A$.

The TBM differs from Dempster Shafer in that it uses unnormalised combinations of beliefs and as such it doesn’t constrain $m(\emptyset) = 0$. This is a key aspect of the TBM’s usage here. If $m(\emptyset) > 0$ for $m_{02}$ then we can deduce that there is some conflict involved between the two sources of information $m_1$ and $m_2$. The combinations of sensors to target relationships that can minimise $m(\emptyset)$ will be the best combination, and best describe the most suitable sensors to targets association.

The TBM works with a frame of discernment which gives $2^n$ sets, where $n$ is the number of singleton elements, or possible target classes, or positions in this case. It is quite obvious that when using a large number of singleton elements, as is typically the case in our data association algorithms, operations on such large sets can become computationally expensive. However, we employ techniques to reduce the time needed for operations on these frames of discernment through the use of binary tree structuring which is used for all frames of discernment. This enables operations to be performed efficiently, and only sets of possibly use, as in their basic belief assignment is not zero, are evaluated. We also condition each frame of discernment with its core and order the frames of discernment based on their average set sizes. For more information on these and other techniques see the work of Wilson [21].

3.2.2 Data Association Algorithm

Each sensor $1...I$ will have some confidence level attached to it which signifies the reliability of that sensor, if the sensor is completely reliable then the confidence is 1, likewise if it is broken then the confidence is 0. We will discount the sensor based upon the confidence given to it [7]. Through discounting we will redistribute the remaining confidence to the proposition that the target can be in any of the the cells thus accounting for the unreliability in that sensors measurement. To incorporate the localisation error that is associated with the sensor measurements we also distribute a portion of the belief that the sensor has given to the target position to its immediately surrounding cells. It is this action that allows for the TBM fusion process to cluster sensor target positions that are similar and that have originated from the same target.

The algorithm will then evaluate the number of targets present through analysing the conflict between sensors for targets $i$, where $i = 1...n$, and $n$ is some limit such that $n \leq l$. This is done by taking each possible combination of sensor to target associations and combining the belief functions for those sensors that are proposed to be looking at the same target, and so are clustered. The empty set values for each cluster are added for that that combination to find the overall internal conflict that that combination
gives. If the resulting overall conflict is zero then the number of targets can be described exactly by that combination of sensor to target associations, in reality this is unlikely to happen unless there is no localisation error for the sensor measurements, or there are the same number of targets as sensors. As the empty set is normally non zero, some threshold must be used to decide on the applicability of that decision. This threshold will control the ability of the algorithm to cluster sensors that are reporting a target at similar positions.

The possible number of combinations for each \( t_i \) is a Stirling Number of the Second Kind of the form

\[
\text{stirling}(n, k) = \frac{1}{k!} \sum_{i=0}^{k-1} (-1)^i \binom{k}{i} (k-i)^n
\]  

which can be calculated recursively quite simply. To calculate the actual combinations in an efficient manner can be done using:

1. We find all possible combinations of sensors for target \( k, k = 1 \cdots i \) that obeys
   a) \( |s^j_i| - (i-k) \leq |s^j_s| \geq 1 \) (there must be enough sensors remaining to place at least 1 in each of the remaining clusters)
   b) \( s^j_i \text{ min} > s^{k-1}_j \text{ min} \) (the lowest sensor number of \( s^j_i \) must be larger than the lowest sensor number of \( s^{k-1}_j \), the previous cluster. This prevents replications)

at each of those combinations we remove \( s_j \) from \( s_i \) to give us our remaining ‘pool’ of sensors to cluster and then goto 1 and increment \( k \)

<table>
<thead>
<tr>
<th>Algorithm 1. Sensor Combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>If we have the set of sensors ( s_j = {s_1, s_2, \ldots, s_j} ) all looking at ( t_i ) targets we want to find all the combinations of putting the sensors ( s ) into ( i ) unique cluster combinations. For example, by unique we mean that for ( j = 6 ) sensors ( {{s_1, s_2, s_3, s_4, s_5, s_6}} = {{s_1, s_2, s_3}, {s_4, s_5, s_6}} = {t_1, t_2 \ldots t_3} )</td>
</tr>
</tbody>
</table>

\[ \text{where each combination is related to a target, so the above is clustering for 3 targets.} \]

3.2.3 Target Position Decision Making

If we decide that there is more than one sensor reporting on the position of the same target, we use the previously described TBM to evaluate the position of that target. Through combination using Equation 2 we fuse the positional measurement of each sensor that reports on that individual target. Each sensor will have a belief function associated with that describes where it thinks the target is positioned, it is these belief functions that we combine. A pignistic transform, given by Equation 3, is then performed on this combined belief function. This will give us the probabilities of the possible target positions from which a decision can easily be made.

3.3 Particle Filter Tracking

The tracking of the targets takes place through the use of a particle filter. This allows for non-linear system modelling, as we expect a non linear system for our targets. Each target track will have its own filter assigned to it. The particle filter tracks the motion of the target from the information received about target position from Section 3.2. The particle filter will track the objects motion by estimating a multi-modal non Gaussian probability density function. The probability density function is approximated through the use of a number of samples (particles). The samples represent a state hypothesis and have a weight associated with each of them that signifies the quality of that state. The weighted sum of these particles will give the state estimate. The particle filter is an iterative process that involves prediction and update stages. In the prediction stage each particle is updated according to a motion model, and some noise is added as the variable of interest will indeed have some noise. Each target type has particular characteristics and as such has a different motion model. The PF uses the correct motion model for the object as determined by the classification phase. Each particle is then updated based on the latest measurement and its weight recalculated. Particles are then resampled to try and ensure that the fittest particles are kept, as they are the best state estimate, while unfit particles can be discarded.

3.4 Particle Filter Setup

Our particle filter will have a state at time \( t=k \)

\[ x^k = [\text{posx}, \text{posy}, \text{velx}, \text{vely}, \text{accelx}, \text{accely}] \]

which is approximated by our set of \( N \) samples, or particles, which is given as \( S^i_j = [x^i_j, w^i_j] \) where \( j = 1 \ldots N \) and process noise is given as \( w^k \). The motion models for classes of object differ due to their medium and have constraints based upon their class and the terrain that they are encountering. We assume that the noise within the system for the particle filter is Gaussian. The particle filter steps are [28].
3.5 TBM Target Classification

The classification of the target uses a TBM framework and takes information from the target tracker and from the terrain map. We have allowed for 3 different target types within our synthetic example, and also the possibility of a false detection. The target types are boat, tank and car. These 3 types were chosen due to their different target motions and to show the possible use of a terrain map in helping differentiate between targets, through its use as an extra feature in the classification stage. The TBM will create a probability distribution over the set of possible classes for each target. The TBM will fuse the available information so that a robust decision can be made on the classification of that target, and this classification can then be passed to the particle filter for use in its prediction stage as each class will have a motion model and constraints associated with it. Each target will have a basic belief assignment associated to it for each of the features $f_i = \{\text{T terrain}, \text{Velocity, Acceleration, TrackLength/T}\}$ that we are using to classify it, then it is each of these basic belief assignments that we combine to obtain the final decision probability. The features used are able to apply a likelihood to the class of target associated with that feature. They are self explanatory but Track Length/T is the number of detections for that target over time since track started for that target. This is used to help identify false tracks on the assumption that a false track will regularly lose information or not be detected.

3.5.1 TBM Classifier Setup

The targets to be classified have the following classes available to them:

<table>
<thead>
<tr>
<th>Class</th>
<th>Abbreviation</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boat</td>
<td>B</td>
<td>Water</td>
</tr>
<tr>
<td>Tank</td>
<td>T</td>
<td>Land</td>
</tr>
<tr>
<td>Car</td>
<td>C</td>
<td>Land</td>
</tr>
<tr>
<td>False Track</td>
<td>F</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

Table 1. Possible Target Classes

Classification on each of the features are described in the following tables:

<table>
<thead>
<tr>
<th>Terrain</th>
<th>Set</th>
<th>$m$</th>
<th>$m(\Theta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forest</td>
<td>{F}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>Road</td>
<td>{T,C,F}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>Offroad</td>
<td>{T,F}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>Water</td>
<td>{B,F}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 2. Likelihoods for Classification based on Terrain

<table>
<thead>
<tr>
<th>Vel</th>
<th>Set</th>
<th>$m$</th>
<th>$m(\Theta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&lt;10$km/h</td>
<td>{B,T,C,F}</td>
<td>1.0</td>
<td>1</td>
</tr>
<tr>
<td>$10&gt;20$km/h</td>
<td>{T,C}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>$20&gt;30$km/h</td>
<td>{C}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>$30&gt;60$km/h</td>
<td>{C,F}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>$&gt;60$km/h</td>
<td>{F}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 3. Likelihoods for Classification based on Velocity

<table>
<thead>
<tr>
<th>Accel</th>
<th>Set</th>
<th>$m$</th>
<th>$m(\Theta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(High)$&gt;0.4$m/s</td>
<td>{C,F}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>(Medium)$0.15&gt;0.4$m/s</td>
<td>{T,C}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>(Low)$&lt;0.15$m/s</td>
<td>{B,T,C}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 4. Likelihoods for Classification based on Acceleration

<table>
<thead>
<tr>
<th>Track Length/T</th>
<th>Set</th>
<th>$m$</th>
<th>$m(\Theta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Long)$&gt;50%$</td>
<td>{B,T,C}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>(Short)$&lt;50%$</td>
<td>{F}</td>
<td>0.9</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 5. Likelihoods for Classification based on Track length ratio

The values used for the classification boundaries and basic belief masses for the classifiers were intuitively chosen for this example scenario and allow for a small amount of error, 10%, in the system.

The classification of the targets will be decided by:

\[ m_{\text{combined}} = m \oplus m_1 \oplus m_2 \oplus m_3 \]

Combining all the features for target $t_i$, such that $m_{\text{combined}} = BetP(m_{\text{combined}})$

Then the class of target $t_i$ will be the singleton with the maximum $BetP$ or the set with the best $BetP$.

4 Results

The results shown here show examples of how the algorithms within our system operate when presented with test data, we go on to discuss the results and why they have occurred.

4.1 Data Association

The data association will calculate the number of targets, the sensor to target relationship and finally the target positions based on the fused sensor information. The results this are shown below.
4.2 Target Classification

Some examples of classification are given here to describe, at a more manageable level, how the classification algorithm works and performs.
4.3 Target Tracking and Classification

To demonstrate the joint tracking and classification ability of the algorithms described previously, a simple single target track and classification example is given. Figure 4 shows the terrain map, and overlayed on this is the positional and velocity reading from one sensor of one target. The corresponding particle filter best estimate is also shown with its associated particle cloud and an indication of the standard deviation of the particles is given. Figure 5 shows the path of the detected object along with the tracked path. This tracking takes place in conjunction with the TBM classifier. The results of the classifier for this track sequence is given in Figure 6. We can see that initially the classifier believes that the object is of the set \{Tank, Car\}. Other similarly ignorant sets also have initial values, but these soon resort to 0 as more information is received. The object is moving with relatively smooth acceleration and velocity, shown by the spacing of the positions on the track. The object in general follows the path of the road but in a few instances it wanders from the road, this could be due to sensor error, but it is indeed the information that we are presented with. This has a marked effect on the output of the classifier. The instances where the target is offroad can be seen as steps in the graph, so at around time frame 13, 25 and 37.

Because a car cannot leave the road this leads us to believe that the object is in fact a ‘Tank’. It is possibly a little unnerving that these very slight excursions from the road result in such a marked change, and suggest that more suitable values could be used for the uncertainty given in the classifiers features, shown in Table 2 to Table 5, this should allow the classifier to keep open other possible options. Another factor that would aid in encompassing the possibility of sensor error for target position would be to evaluate the terrain type of surrounding cells and allow for some of the likelihood measure to be assigned to these.

5 Conclusion

We have presented a novel method for associating multiple targets to measurements and allowed for these targets to be tracked and classified with information been shared between the stages of the process to enable a more robust outcome to be achieved. We have added to previous work [2] by looking at the inclusion of PF’s with the TBM for tracking and classifying, we think that this is valuable as tracking of objects in reality is highly non-linear and needs to be addressed through improved models of target motions. We feel that integration of a system will become complete when all the stages are put into a TBM framework. At present we are able to efficiently and effectively track and classify targets within this scenario and have the flexibility to adapt the system for other scenarios. Future work will be the integration of the TBM and PF so that the JTC is more coherent, and the addition of more realistic and complex motion models. This will then be applied to real world scenarios to give a better assessment of performance in comparison to current JTC technologies.
6 Acknowledgements

Thanks go to the Ministry of Defence, Data Information
Fusion Defence Technology Centre for funding project
4.10 Statistical Fusion of Battlefield Data for which this
paper is a part of, and also to Philippe Smets for
comments and suggestions regarding the TBM.

7 Bibliography

[1] Challa S. and Pulford G. Joint target tracking and
classification using radar and ESM sensors. IEEE
transactions on Aerospace and Electronic Systems, 37,
July 2001, pp1039-1055
[2] Smets P. and Ristic B. Kalman filter and joint
tracking and classification in the TBM framework,
Proceedings of the Seventh International Conference
on Information Fusion, Sweden, 1, 2004, pp46-53
tracking :principles and techniques. YBS Publishing,
1995
tracking. American Scientist, 80, 1992, pp128-141
[5] Blackman, S. S. Multiple-target tracking with radar
[6] Hall David L. Mathematical techniques in multi-
sensor data fusion, Norwood, MA :Artech House, 1991
[7] Ayoun A. and Smets P. Data association in multi-
target detection using the transferable belief model
International Journal of Intelligent Systems 16(10),
2001, pp1167-1182
[8] Angelova D. and Mihaylova L. Sequential Monte
Carlo algorithms for joint target tracking and
classification using kinematic radar information,
Proceedings of the Seventh International Conference on
Information Fusion, Sweden, 1, 2004, pp709-716
[9] Powell G. and Marshall D. and Markham K. and
Milliken R. Data fusion of FLIR and LADAR in
autonomous weapons systems, Proc 6th International
Conference on Information Fusion, Cairns, Australia,
2003, pp350-357
[10] Powell G. and Marshall D. The data fusion system
for object recognition, Proceedings of the Seventh
International Conference on Information Fusion,
Sweden, 1, 2004, pp54-61
for joint tracking and classification, In proc SPIE,
volume 4728, 2002, pp439-449
[12] Smets P. and Ristic B. Kalman filter and joint
tracking and classification in the TBM framework,
Proceedings of the Seventh International Conference
on Information Fusion, Sweden, 1, 2004, pp46-53
based on haar –like features and eigenfaces, 5th
IFAC Symposium on Intelligent Autonomous
Vehicles, Portugal, 2004
[14] Marins J. and Yun X. An Extended Kalman Filter for
quaternion based orientation estimation using
MARG sensors, Proceedings of the 2001 international
conference on Intelligent Robots and Systems,
kalman filtering for estimating quaternion motion,
Proceedings of 2003 American Control Conference,
IEEE Press, 2003, pp2435-2440
[16] Khan Z. and Balch T. and Dellaert F. Efficient
particle filter based tracking of multiple interacting
targets using an MRF based motion model,
Proceedings of IEEE International Conference on
Intelligent Robots and Systems, 2003 pp254-259
[17] Isard M. and MacCormick J. Bramble :A Bayesian
multiple-blob tracker, ICCV II, 2001, pp34-41
[18] Arulampalam S. and Maskell S. and Gordon N. and
Clapp T. A tutorial on particle filters for on-line
non-linear/non-gaussian bayesian tracking. IEEE
Transactions on Signal Processing, Vol 50, February
2002, pp 174-188
Bayesian algorithm for identification of tracked
targets, IEEE Transactions on Systems, Man and
Cybernetics, 23, 1993, pp1683-1698
[20] Bogler, P. L. Shafer-Dempster reasoning with
applications to multisensor target identification
systems, IEEE Transactions on Systems, Man and
Cybernetics, 17, 1987, pp968-977
Handbook of Defeasible Reasoning and Uncertainty
Management Systems, Vol 5 Pub Kluwer, Doordecht,
The Netherlands, 2000, pp421-475.
[22] Schubert J. Cluster-based Specification Techniques
in Dempster-Shafer Theory, European Conference
Symbolic and Quantitative Approaches to Reasoning
and Uncertainty (ECSQARU 95), Springer-Verlag
(LBCS), Berlin 1995, pp395-404
[23] Dempster A. P. Upper and lower probabilities
induced by a multiple valued mapping, Annals
[24] Smets P. The application of the transferable belief
model to diagnostic problems, International Journal
[25] Smets P. The transferable belief model for
quantified belief representation, In Handbook of
defeasible reasoning and uncertainty management
systems, Vol 1 1998, pp267-301
[26] Smets P. Practical use of belief functions, In
Uncertainty in AI 99, 1999, pp612-621
[27] Smets P. and Kennes R. The transferable belief
model. Artificial Intelligence, 66, 1994, pp191-234
[28] Ristic B. and Arulampalam S. and Gordon N Beyond the
Kalman Filter. Particle Filters for Tracking