A Data Fusion System for Object Recognition based on Transferable Belief Models and Kalman Filters

Gavin Powell & Dave Marshall
Cardiff University
Cardiff
CF24 3XF
UK
G.R.Powell,dave@cs.cf.ac.uk

Richard Milliken & Keith Markham
BAe Systems/Matra Dynamics
Filton
Bristol
UK
richard.milliken,
keith.markham@bae.co.uk

Abstract - *We examine the use of fusing data from multiple data sources for use within object recognition systems. We then continue, to illustrate the system that we have created for our own object recognition needs. The data fusion model that we use is embedded within an object recognition system that analyses simulated FLIR and LADAR data to recognise and track aircraft. The data fusion is based upon the Transferable Belief Model (TBM) and Kalman filters. The system is novel due to the simulation of the sensors and the use of multiple Kalman filters and TBM's.*

Keywords: TBM, fusion system, Kalman filter, ATR.

1 Introduction

The use of data fusion within object recognition systems is becoming more evident, as it becomes apparent that better true positive rates can be achieved through the use of disparate data sources. This can be within a variety of scenarios such as robot navigation, analysis of humans, and military applications. Many of these are based upon the same idea that some object is to be observed and recognised, and then some analysis is done upon that object. Single sensor systems are applicable in certain scenarios, but many require a multi sensor, multi modality approach. Through the use of more than one sensor, and possibly modality, the object under scrutiny can be viewed in a variety of ways. Each of these collecting a different piece of information about that object. The combination of these observations allows for a more complete understanding of the object which is being viewed. It also provides a more robust definition by lessening the effect of random sensor error, or even failure.

Our application is an air to air missile based simulation system, where we use both LADAR and FLIR sensor images for the recognition of our objects. The sensors themselves are simulated within the software [1, 2] allowing for us to present any scenario to the object recognition system, for testing purposes. The system is a continual feedback model, where each fused result is fed back into the system to assist with the recognition phase in the next time cycle. This paper furthers on previous work [3] by providing a more in depth review and analysis of the fusion algorithm, and results, showing its ability to produce effective classifications of objects in a variety of challenging scenarios.

2 Paper Overview

In Section 3 we take a look at some of the different technologies available for the fusion of data from multiple sources. In Section 4 we review the techniques used for recognising objects within our system. Section 5 shows the fusion strategy that we have employed and the algorithms that go with this strategy. The results of the fusion algorithms are shown in Section 6. They show how the system reacts with a variety of data inputs. Finally we conclude our results in Section 7

3 Data Fusion Technologies

To combine data from various sensors we 'fuse' the information in such a way that the information obtained from the output is greater than the sum of its parts. There are a variety of techniques for the fusion of data, as well as different levels. The levels that we fuse at are low, intermediate and high.

In the modern world, we have the ability to analyse a scene with an ever increasing range of sensing modalities. We need to be able to bring the information together so that we get 'the best' out of each sensor, allowing for the robustness of any output to be as close to optimal as possible.

Fusion is a way in which we can combine sources of data, which are more often than not, uncertain in some way. A range of mathematical procedures have arisen over the years to help us deal with those uncertainties. Probability theory is the earliest of these approaches, but more recently others have come into play. These include fuzzy sets [4], possibility theory [5], the theory of fuzzy measures [6, 7] and evidence theory [8, 9, 10]. Each can have its strengths and weaknesses, and be more suited to particular scenarios. One of the requirements for our system was that it would be flexible enough to deal with the nature of our data. This meaning that at times matches from our data could be non existent due to sensor, or algorithm, failure. Such is the nature of our data. We also liked the idea of lack of knowledge, apriori, to be perfectly acceptable within our system. These necessities turned us toward the evidence theory approach, and then to the TBM [11]. The flexibility of this approach allows for us to model ***closed*** and ***open*** worlds, so we can entertain the possibility of objects outside of our world being the target as well as a scenario where we

have little or no information and so are not in a position to make a decision. Total ignorance can be modelled, showing a distinct lack of knowledge about our scene at any point. If we do have prior knowledge then this can be utilised. If the information that we have is only based on singleton hypothesis then the TBM approach becomes the Bayes' approach. This provides a very flexible system, which encapsulates all of our requirements. This is a novel approach that has not seen use in this domain before. There are a variety of works utilising the Dempster Shafer Theory (DST) approach, but these have shortcomings in a scenario such as ours. The superset can explode with the curse of dimensionality within DST and when combining beliefs they will become normalised, which can lead to a false sense of security when making your decisions. We use the TBM to overcome the shortcomings of the DST and create a more robust form for fusing the data.

4 The Object Recognition System

There are not three distinct levels of interpretation for all systems but this can be taken as the routine of a typical one [12].

At the lowest levels the fusion of data is performed on the image to detect features and segment the image. The process can look for areas of similar data and then group it to give regions of interest. This reduces the amount of data being passed to later phases, which are computationally heavy. Fusing at the intermediate stage takes place on feature attributes and values. Here we may be trying to discover the surface shape, material type or pose of the object. At the highest level we try to classify parts of the scene or interpret features previously extracted. This is where we 'label' the scene.

Within our system we fuse data at varying levels to find the pose and object type. This can then be used to refine the lower levels in the following frame. We fuse data obtained from our object recognition stage. When recognising our objects we determine the object type and its pose. The 2D FLIR matching algorithms use template matching to acquire the 3D pose and type of the object. A fitness measure is also output indicating the quality of the match. Similarly, for the 3D LADAR match an overall fitness measure is generated from the matching process. Even though it is the actual type and pose that we fuse together we need the fitness measures to indicate which sensor is performing well at that point, or if any. The higher the fitness measure the more we believe the match from that sensor.

The matches are fused for $frame_n$ to provide a more robust result. This is then used as part of a feedback loop. This feedback adds constraints to the matching algorithms, to minimise search space, and guides the fusion and tracking process for future frames.

Figure 1 shows the system as a whole and the flow of data between its parts. It is colour coded in the following manner. Blue shows the creation of modelled scene data, which describes the objects and the world they live in. This is not necessarily a static description as the objects can move about the scene. The camera is the view taken by the missile itself as it evaluates the scene, so this can also move within the scene. Red is the transferral of 'real' data to simulated sensor data, as data from the scene is passed to our sensor models. This is where we alter the highly accurate data obtained from the rendering, so that it resembles the data that the sensors would output in the real world. Green is the FLIR algorithms that extract the object type and orientation from the 2D FLIR data through template matching and shape descriptors. Yellow is the LADAR algorithms that build the EGI and then model match this to obtain the object type and orientation. Brown are the fusion algorithms, where we use the TBM to fuse the object types that the FLIR and LADAR sensor matching algorithms provide us with, similarly the Kalman filter fuses the pose. Purple shows the feedback loops that provide information about the scene in some way to influence decisions in the matching algorithms or to update themselves with their 'running total' of type and pose to this time instant.

5 The Data Fusion System

We have spent a great deal of time considering the methodology to utilise for the fusing of object type match data within our system. This plays a key role within the system and has received careful consideration due to the specific requirements of our system. Our system has to be able to use data sets which may have complete loss over certain frames for a particular sensor. We may have prior information available that we would like to use if it is available. We may also have an open world where our database does not contain all possible matches. The method we have chosen, the TBM, is capable of dealing with all these scenarios, and supersedes previous incarnations of DST. It also is shown to be an improvement on the classical Bayesian approach. This approach has not been used in such a scenario before and so we have presented novel work, which is also open source.

The Kalman filters have been used as components in the system in a novel manner. They play a dual role in fusing the pose estimates from the matching algorithms, and also as a device for the tracking of the objects pose over time. This works in parallel with the TBM fusion of the object type, then together they form the feedback loop to add control to the pose and type matching algorithms in future frames.

5.1.1 TBM Fusion

The TBM presents a very capable method of fusing data. It is able to combine information from two distinct sources and allows for a great deal of flexibility in its approach to doing so. Its appeal for use in a sensor fusion target recognition scenario is its ability to cope well with the type of measures that we can extract from our scene. It handles lack of knowledge very well, and has the useful ability to be able to incorporate prior knowledge if and when it becomes available. In our scenario we often find ourselves in the position that data from one or both sensors is ambiguous. It also returns sensible results when

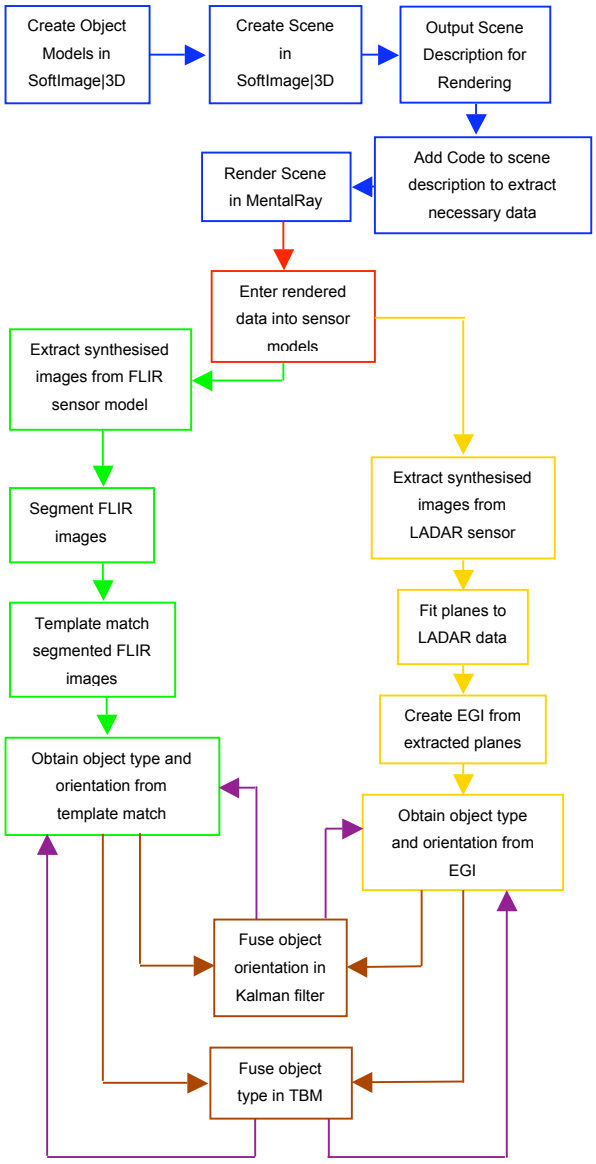

Figure 1. Data flow through the fusion system

sensors are in conflict. Some fusion methods can falsely weight their output when there is a lack of data or we are unsure if its credibility, and also produce spurious results when sensor conflicts occur, details of the TBM can be found in previous work [3].

As far as we are aware, the TBM has not been used in such a manner within the ATR field and as such this work is novel. We use the TBM within part of a feedback loop where its results are fed back to the system to allow search spaces to be pruned and a better match to be obtained. The information that we receive from our LADAR and FLIR recognition processes are used as the information that is entered into the TBM for fusing. Each sensor outputs some data from which we try to obtain a match for pose and type of our object. Accompanying the match is a level of confidence; it is these measurements that are used as our conditional plausibilities. They denote the strength of the match for the object type. Through fusing these two measures we can obtain a more robust match to the type of object.

The conditional plausibilities provided by the FLIR and LADAR sensor object type matching algorithms are denoted SF_n and SL_n respectively for frame n. Our frame of discernment, H, covers the basic types of object that we require to match against, these been a Harrier GR7, an A-10 Tankbuster and an F-15, creating the set $H=\{GR7, A10, F15\}$. Both sensor matching algorithms have knowledge of all elements of H, so that $SF_n \subseteq H$ and $SL_n \subseteq H$. These conditional plausibilities are presented to the Generalised Bayesian Transform (GBT) to obtain the mass and, belief and plausibility functions. If we have prior knowledge it is included at this point through conjunctively combining the GBT result with the prior information. The masses obtained from the GBT, $m^H[SL_n]$ and $m^H[SF_n]$, for the two sensors are combined using the conjunctive combination rule. A pignistic transform of the mass distribution, obtained from this combination, will provide us with the pignistic probability function, from which we can see the result of fusing the outputs from the two sensor algorithms. It is this probability function that we can then make our decision on. The output from the TBM at this point is just the fusion of the matches for this frame and so is denoted TBM_n. Our system is an ongoing process that receives data at each frame. To obtain the best type match possible we utilise all data received so far. This is achieved through a running TBM, for frames $1 \ldots n-1$ denoted as $TBM_{1..n-1}$, into which we fuse the output from TBM_n. The results from TBM_n are conjunctively combined with the $TBM_{1..n-1}$ to give $TBM_{1..n}$. A pignistic transform is then applied to the mass assignment of $TBM_{1..n}$ from which we are able to make decisions. These decisions are used to influence the direction that the algorithms use when trying to find a match for the data from the sensors at frame $n+1$.

5.2 Kalman Filter Fusion

The matching algorithms for the FLIR and LADAR sensors output not only object type but also pose matches. For each match that they output they also provide a measure which represents the quality of that match, from this we can find the best 5 matches from that algorithm, for that sensor at that particular time instance. This happens for both the FLIR and LADAR sensor matching algorithms. Some how we need to combine all of this data in a manner which allows us to establish the best estimate of object pose at this time instant. For this we must fuse the various observations that the algorithms have made. The Kalman Filter is ideal for tracking and fusing pose estimates.

5.2.1 Kalman Filter Tracking

In our system we implement a Kalman filter for the tracking of our object. This allows us to pass the noisy pose measurements to the filter, and for it to project forward to the next time step using physical laws of object motion. This is then used to guide the fusion process where the current pose is determined from the LADAR and FLIR sensor object matching algorithms as well as correcting the filter.

The filter works by using a feedback loop, in this loop the filter estimates the system state and then uses the measurements as a feedback. The Kalman filter equations are of two types being time update and measurement update. The time update is the forward projection of the state estimate, apriori. The measurement update is through the use of the feedback obtained form the measurements; this in essence updates the estimate to produce a better one, aposteriori. The equations for the time update are:-

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (1)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2)$$

Where $\hat{x}_k^-$ is the prior state estimate at step *k* and $\hat{x}_k$ being the posterior state estimate at step *k*. The $n \times n$ matrix *A* relates the state at *k-1* to the current state *k*. The $n \times l$ matrix *B* relates the optional control input *u* to the state *x*. The matrix *P* is the error covariance estimate, a priori being denoted by a super minus. Finally the matrix *Q* being the process noise covariance. From Equation 1 and Equation 2 it can be seen how the current state estimate and covariance error states are projected from step *k-1* to *k* to give the a prior estimates. The equations for the measurement update are:-

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (3)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (4)$$

$$P_k = (1 - K_k H)P_k^- \quad (5)$$

Where the $n \times m$ matrix *K* is the Kalman gain, this is the measure that minimises the posterior error covariance P_k. What the Kalman gain does is to weight the filter toward the incoming measurement or the posterior estimate. This weighting being based on the measurement error covariance, *R*, and the prior estimate error covariance, P_k^-. The value z_k being the incoming measurement at step *k*, and finally the $m \times n$ matrix *H* relates the state to the measurement z_k. From Equation 3 to Equation 5 we can see that the prior measurements are being projected forward to a posterior states.

5.2.2 Filter Design

To design our filter we looked at the information that the fusion algorithms were able to output. The Kalman filter that we use for fusing the pose estimates outputs a single pose estimate for that time instance, also the tracking filter will produce a projected posterior pose estimate. The pose estimate produced from the fusion filter is used as the measurement going into the tracking filter, and is what the system thinks the next pose will be. The start state of our object, in terms of pose, for the fusion filter is the projected forward state obtained from the tracking filter and form a part of the feedback loop, allowing the tracking filter to guide the fusion process. The quality of the match, obtained from the tracker and its covariance is modelled with the matrix *R*. The smaller the values that are entered here, the more the tracker will believe that the incoming measurement, z_k, is correct. The measurement, *z*, is a vector with eight components, those being *(*$quaternion_P${theta, x, y, z}, *$quaternion_{AV}${theta, x, y, z}).* These components are in fact two quaternions, with $quaternian_P$ describing the pose and $quaternian_{AV}$ describing the angular velocity. The quaternion describing angular velocity enables the filter to more reliably project ahead to the next time step and to make a prior state estimate. To calculate this we store the pose estimate for previous time instances and find the difference:-

$$AngularVelocity_k = Pose_{k-1} - Pose_{k-2} \quad (6)$$

This can be made more accurate by using more pose measurements, but we found that this worked well enough and managed to react quickly to changes. We also experimented with angular acceleration but found that performance of the filter became less stable even though its reaction time was improved.

The process noise covariance matrix, *Q*, is assumed to be constant. The measurement, *z*, is directly related to the state estimate, *x*, so the matrix, *H*, that relates the measurement to the state is just an identity matrix. The apriori error covariance matrix *P* and the inter frame state relationship matrix *A* are also identity matrices due to the direct relation between the measurement and the state estimate. The initial state estimates are all unit quaternions, this is an acceptable assumption, and doesn't have very much effect on the filter as it settles down after a couple of frames. To fully test the tracking filter it was implemented into a mouse driven tracker using OpenGL to model an aircraft and give a visual feedback. Here the user can control the ground truth pose using the mouse. Noise is added to the ground truth to become the measurement, $quaternion_P$ of the vector *z*, which is given to the filter. In this way we are able to test and tune the filter in terms of the how to project forward using equations such as Equation 6.

5.2.3 Fusion Algorithm

To fuse the matches that have been obtained from both sensors we use a Kalman filter. The matching algorithm gives us the top five pose estimates from each sensor. Along with an estimate of the pose a measure is provided to signify the quality of that match. This is used to weight each match through the adjustment of the measurement error covariance matrix. At each frame, or set of matches, a new filter is created. The initial state estimate entered into the filter is taken as the a posteriori forward projection, $\hat{x}_k$, from the tracking filter. The angular velocity derived from the tracking filter is not used as all of the matches have occurred at the same time instance, so the state is just a four component vector $quaternion_P\{theta_P, x_P, y_P, z_P\}$, which is a quaternion describing the objects pose. Then each match is individually passed to the filter and updated.

After all of the matches have been added to the filter we take the state estimate as being the prior state estimate $\hat{x}_k^-$. This gives us a current pose estimate based on the matching algorithms outputs for both FLIR and LADAR sensor. This can now be fed to the tracking Kalman filter.

6 Results

Results are presented for various object recognition scenarios/sequences of frames. Results showing the performance of the fusion algorithms for both the object type and its orientation are shown.

6.1 TBM Data Fusion for object type

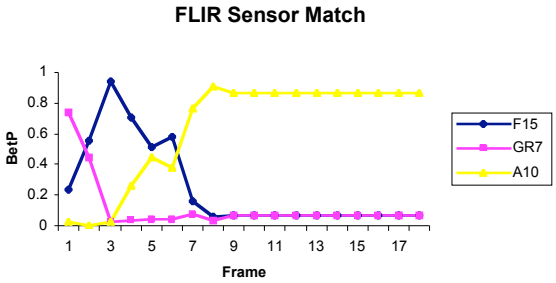

Figure 2. BetP for matches from the FLIR sensor.

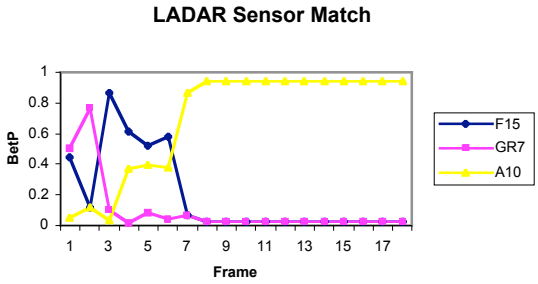

Figure 3. BetP for matches from the LADAR sensor.

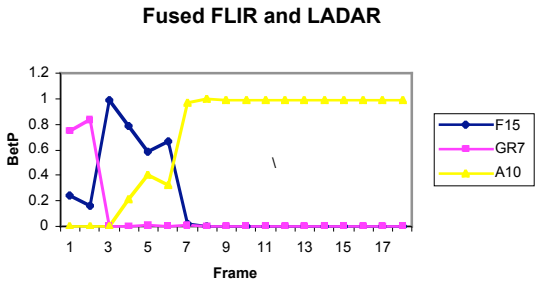

Figure 4. BetP from the fusion of FLIR and LADAR sensor matches at frame *n*.

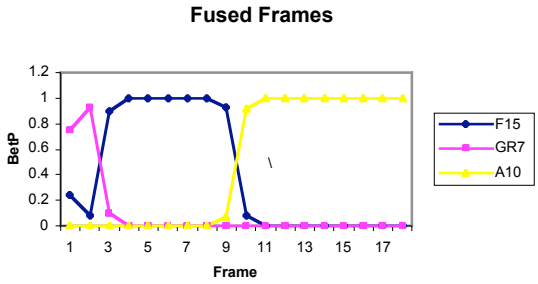

Figure 5. BetP from the fusion of the fused FLIR and LADAR sensor matches for frames *1..n*.

Figure 2 and Figure 3 show the results of fusing all the FLIR matches together and the same with the LADAR sensor. Both FLIR and LADAR sensor are in agreement to the object that they are viewing. It can be seen that in frames 1…7 both sensors are matching to an F15. This then changes favour for an A10. Figure 4 shows the result of fusing these two sensor matches outputs. Figure 5 is a rolling fused total for the output of the fusion of all the FLIR matches (Figure 2) and the fusion of all the LADAR matches (Figure 3) up until that particular frame, and so is the fusion, of all frames to $frame_n$, of the data shown in Figure 4. It can be seen that it recovers well to the change of object type that both sensors are suggesting we are tracking.

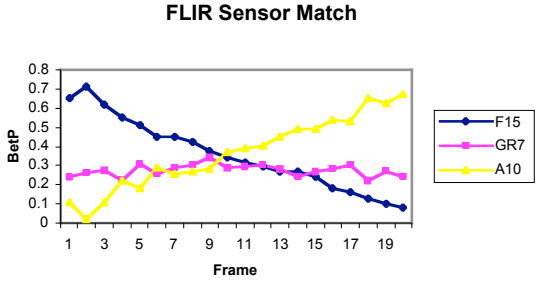

Figure 6. BetP for matches from the FLIR sensor.

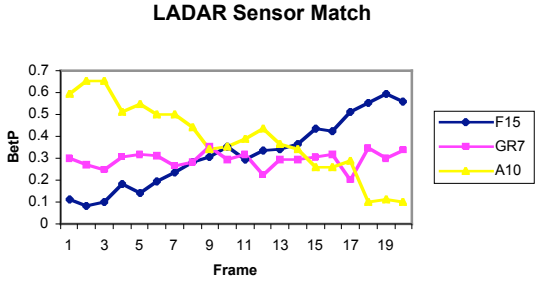

Figure 7. BetP for matches from the LADAR sensor.

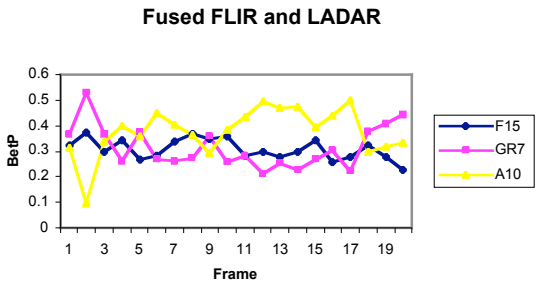

Figure 8. BetP from the fusion of FLIR and LADAR sensor matches at frame *n*.

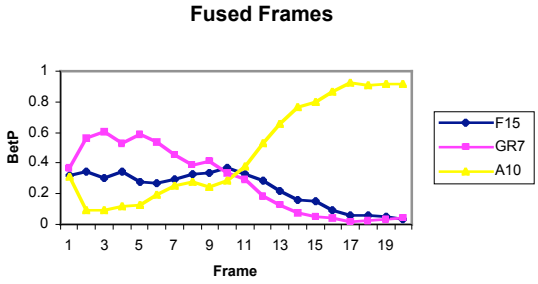

Figure 9. BetP from the fusion of the fused FLIR and LADAR sensor matches for frames *1..n*.

Figure 6 and Figure 7 show the results of the LADAR and FLIR sensor been in disagreement between the F15 and A10 and also that they both change what they believe they are matching to over the time duration. Figure 8 shows the fusion of the FLIR and LADAR sensor at each individual frame. Over initial frames the GR7 is the strongest, when the two sensors outputs have been combined, which then gives way to the A10. This trend becomes even more apparent when looking at the result of fusing all frames to $frame_n$ as shown in Figure 9. Here we can clearly see that the A10 is the most prominent over the latter frames as its dominance in the fused frames (Figure 8) from frame 10 onwards begins to take effect. It is also possible to measure the disagreement between sensors, which can then be used as an additional component for the decision process. In this case we can see that both sensors are in disagreement and a decision based purely on the fused frames output may be unwise, or may need to be scaled in some manner.

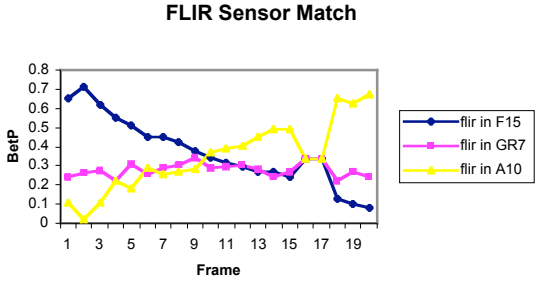

Figure 10. BetP for matches from the FLIR sensor.

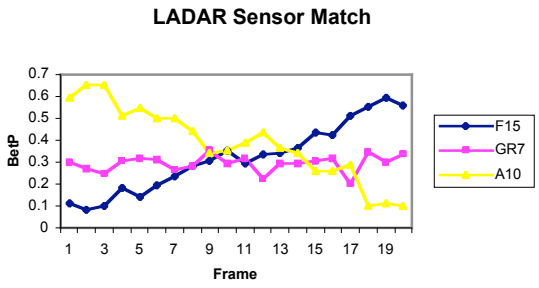

Figure 11. BetP for matches from the FLIR sensor.

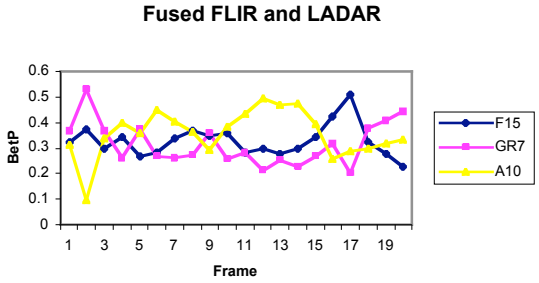

Figure 12. BetP from the fusion of FLIR and LADAR sensor matches at frame *n*.

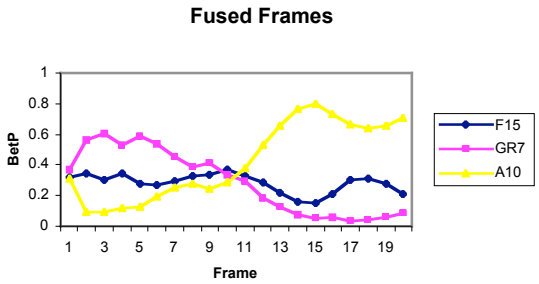

Figure 13. BetP from the fusion of the fused FLIR and LADAR sensor matches for frames *1..n*.

Figure 10 shows FLIR drop out occurring in frames 16 and 17. A lot of data is lost by dropping frames in this area as the BetP was heavily placed on the A10. This is shown in Figure 12 as the BetP for the A10 falls as the onus is put on the LADAR sensor, which is matching to an F15 at this point, and so the F15 rises in probability. Figure 13, which shows the rolling total, for fusing all frames to $frame_n$, should be able to cope with this drop out, even though it has happened at a significant point. A slight drop is witnessed, but due to the trend of it been an A10 in previous frames, the drop is not so that a change in opinion of the match takes place.

Table 1 shows what has happened after fusing all of the available data together for $frame_{1...5}$ for the data from Figure 12. By this point the basic belief assignments (bba's) have almost all been assigned to singletons, the bba's for the other sets are not zero, but are very small. This is because the bba's for the singletons have taken the largest values after the conditional plausibilities have been

assigned to the singletons and then propagated using the GBT for each individual frame for the FLIR and LADAR TBM's. Over time this has swayed the TBM, through combination, to give larger bba's for these values. This measure is not added to the supersets in any way as the bba is just the measure of how much that set and no sub or super sets is supported. This is not the case for the belief, where the values are propogated to the supersets of the singletons. As can be seen in Table 1 the belief for the frame of discernment is 1, denoting that we are sure that the correct match is in here somewhere. It can also be noted that the belief, plausibility and BetP are the same for all sets, this is because only the singleton bba's have noticeable values, and so the TBM resorts to a Bayesian probability.

Set	*Cond Plaus*	*Bba*	*Belief*	*Plausibility*	*BetP*
{∅}	0	0	0	0	0
{F15}	0	0.28153	0.281	0.28153	0.281
{GR7}	0	0.58709	0.587	0.58709	0.587
{F15,GR7}	0	0	0.868	0.86862	0.868
{A10}	0	0.13138	0.131	0.13138	0.131
{F15,A10}	0	0	0.412	0.41291	0.412
{GR7,A10}	0	0	0.718	0.71847	0.718
{F15,GR7,, A10}	0	0	1	1	1

Table 1. Values for Fusing the fused FLIR and LADAR sensor matches up to frame 5 for Figure 10 to Figure 14

6.2 Kalman filter fusion for object pose

Matching the orientation of an object is done using quaternions and so the data presented to the Kalman filter are the *x, y, z* components of the axis that the object is rotating about and *theta*, the angle that it rotates about that axis. The error matrix in the Kalman filter indicates to the filter how much we believe in the measurement that we are providing it with. This is obtained from the match quality returned from the matching algorithms.

Figure 14 and Figure 15 show two of the components from the orientation quaternion for the overall result from the Kalman filter fusion algorithms. At each frame the filter receives data, which is the result of fusing the FLIR and LADAR match results within a separate Kalman filter, which it fuses to its current best estimate of the orientation. This best estimate is obtained by fusing all previous frames. For clarity two of the components of the quaternion are graphed individually. Looking at Figure 14 and Figure 15 we can see that the fused FLIR and

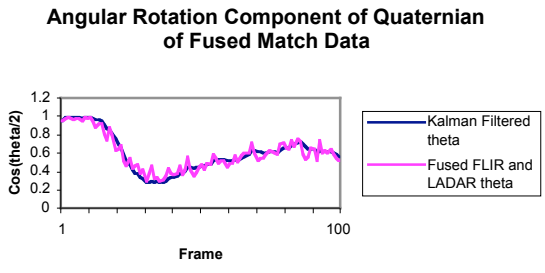

Figure 14. The *theta* (ang rotation) component of the quaternion for overall fused output of the FLIR and LADAR matches, via a Kalman filter.

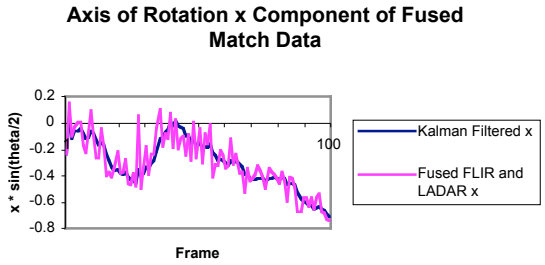

Figure 15. The *x* component of the quaternion of the overall fused output of the FLIR and LADAR matches, via a Kalman filter.

LADAR data (purple line) is noisy, and this is the result of the matching process not been completely accurate. This is to be expected, and the smoothing of this data is the purpose of fusing it together within the Kalman filter. The Kalman filtered data (blue line) is the result of fusing all of the available data to this point. Due to the use of the Kalman filter we also project ahead to the next frame and track the object through this medium. Figure 14 and Figure 15 represent the object making a banking turn. The fused tracking of this object works well through the whole of the banked turn particularly so in the x axis of rotation where we can see that the algorithms are having slightly more trouble gaining a consistent result due to the x-axis having the most rotation, and thus noise, in this banking manoeuvre.

7 Conclusions

We have looked at the fusion of data from disparate sources for the recognition of airborne objects. We have demonstrated our methodologies for the solving of this complex problem. We have achieved a system that can competently fuse data from LADAR and FLIR sensors to achieve a better overall match to the object we want to pursue.

We believe our choice of system to be the best fusion system for our scenario. This came about through good research, design and testing. Initial research and readings were on Bayesian analysis of data and DST. We decided that the DST provided us with a better formula for describing knowledge within our model, mainly through being able to model complete ignorance, without the need for a prior information. Initial tests of the DST, and continued research, found that there were some weak areas within the DST. Amongst these short fallings was its inability to look at open world scenarios, and to incorporate a prior information properly. We discovered, work done by Smets, that was an extension of Dempster's original work, that encapsulated such points within it. We chose to adopt and adapt some of Smets' work for our own system and incorporated it into the system to fuse the data from the FLIR and LADAR sensors. This work was new to the fusion arena for air-to-air missile target recognition applications and was published in 2003 [3].

The Kalman Filter is not a new piece of technology by any means, but its due to its tried and

trustedness that we chose to use it for our fusion and tracking. It is adaptable to track rotations through the use of quaternions, and can be used to fuse matches as well. Results of the Kalman filter system were good and even the multiple filter system ran impressively fast, making for a very good demonstration piece of software.

Overall the system has proven itself to be able to simulate data, recognise objects and fuse information successfully. Future work for the system as a whole would be aimed at improving the detection phase, so that segmentation could be more intelligent. Also the LADAR recognition phase could be altered, or possibly run on a more powerful machine to allow for an increase in accuracy while not making the run time of the program unacceptable.

8 Reference List

[1] G.Powell, D.Marshall and K.Markham, "Simulation of LADAR and FLIR data for autonomous tracking of airbourne objects", SPIE Aerosense 2000, proceedings of SPIE Vol. 4026, April 2000.

[2] G.Powell, R.R.Martin and A.D.Marshall, K.Markham, "Simulation of FLIR and LADAR data using graphics animation software", Proc. Pacific Graphics 2000, Eds. B. Barsky, Y. Shinagawa, W. Wang, pp126-134, IEEE Computer Society Press, 2000 ISBN 0 7695 0868 5

[3] G.Powell, D.Marshall, R.Milliken and K.Markham, "Data fusion of FLIR and LADAR in autonomous weapons systems". Proceedings of Sixth International Conference on Information Fusion (Fusion 2003) Cairns, Australia, July 8-11, 2003. pp 350-357, ISBN: 0-9721844-3-0

[4] L.A.Zadeh, "Fuzzy sets", Information and Control, Vol 8(3), pp338-353. 1965.

[5] L.A.Zadeh, "Fuzzy sets as a basis for a theory of possibility", Fuzzy Sets and Systems, Vol 1(1), pp3-28, 1978.

[6] M.Sugeno, "Theory of fuzzy intergrals and its applications", PhD Dissertation, Tokyo Institue of Technology, Tokyo, 1974.

[7] M.Sugeno, "Fuzzy measures and fuzzy integrals: A survey", In: Gupta, SAridis and Gaines, eds., Fuzzy Automata and Decision Processes, North Holland, Amsterdam and New York, pp89-102, 1977

[8] A.Dempster, "Upper and lower probability inferences based on a sample from a finite univariate population", Biometrika, Vol 54 (3&4), pp515-528, 1967.

[9] A.Dempster, "Upper and lower probabilities induced by a multivalued mapping", The Annals of Mathematical Statistics, Vol 38 (2) pp 325 – 339, April 1967.

[10] G.Shafer, "A mathematical theory of evidence", Princeton University Press, 1976.

[11] P.Smets, "Un modele mathematico-statistique stimulant le processus du diagnostic medical", Un published doctoral dissertation, Universite Libre de Bruxelles, 1978.

[12] J.Bevington and K.Siejko, "Ladar sensor modelling and image synthesis for ATR algorithm development", Automatic Object Recognition IV, Proc of SPIE, Vol 2756 pp 62-75, May 1996.