Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Dynamic Trees for Sensor Fusion

Kittipat Kampa and K. Clint Slatton
Department of Electrical and Computer Engineering
University of Florida
Gainesville, FL
kittipat@gmail.com

J. Tory Cobb
Naval Surface Warfare Center
Panama City Division
Panama City, FL

***Abstract*—The dynamic tree (DT) graphical model is a popular analytical framework for image segmentation and object classification tasks. A DT is a useful model in this context because its hierarchical property encodes information in multiple scales and its flexible structure fits complex region boundaries better than rigid quadtree structures such as tree-structured Bayesian networks. This paper proposes a novel framework for data fusion by using a DT model to fuse measurements from multiple sensing platforms into a non-redundant representation. The structural flexibility of the DT will be used to combine common information across different sensor measurements of simulated objects of interest. The appropriate structure of the DT and its parameters for the data fusion application are presented and discussed along with fusion results from a simulated sonar survey mission.**

***Keywords*— belief propagation, dynamic tree, sonar, tree-structured Bayesian network**

I. INTRODUCTION

In many sensing applications multiple measurements of the same region or object in question are presented to a human operator to combine or fuse into a single observation with no redundant information. In underwater surveys, the path trajectories of diverse sensing systems may overlap the same region many times, producing a heterogeneous set of measurements from the same object. The answer most commonly sought regarding a particular sensed object is an accurate geographic position linked to the set of corresponding measurements gathered from the sensing systems. In this paper we propose to use a probabilistic graphical model called a dynamic tree (DT) as the analytical framework for combining redundant sonar image contact information gathered from multiple sensing platforms.

DT graphical models have been successfully implemented in image segmentation applications [1], [2], [3] since the analytical framework provides a flexible hierarchical structure which is desired in many image processing applications, e.g. texture segmentation across complex region boundaries. Here we propose a novel idea to use DTs in a hierarchical sensor fusion task. In addition to its hierarchical structure, DT configurations are based on a probabilistic framework. This framework uses probability distributions to model the relationships among sensor measurements and thus explicitly provides an estimate of uncertainty in the DT maximum *a posteriori* (MAP) solution. In other words, the solution of the DT is in terms of a distribution rather than a single value. The DT solution can be contrasted with an unsupervised clustering routine such as *k*-means clustering where the solution does not have an explicit measure of uncertainty.

Fig. 1 depicts the proposed framework for DT sensor fusion. The sensing scenario is illustrated at the top of the figure. In the cartoon, each DT in the forest represents a single object of interest. In the bottom of the figure, each DT is shown to be made up of a root node at the top level of the tree which represents true state of the object in question. Uncertainty from the sensing platform that collected the measurements on the object is represented in the intermediate level of nodes and raw sensor measurements are represented by the leaf nodes bottom level of the tree.

The rest of the paper presents a solution to the sensor fusion problem of combining redundant sensor information using the analytical framework of DTs. An overview of the properties of Bayesian networks and DTs is presented in the next section. A description of the DT sensor fusion architecture is then introduced. Procedures for optimizing the DT and thus

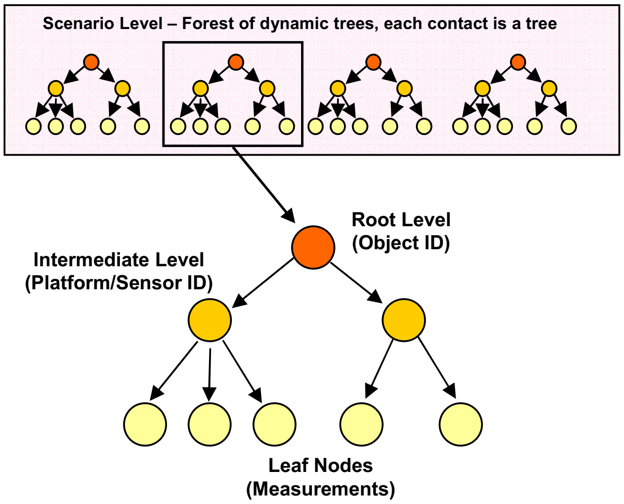


Figure 1. Dynamic tree sensor fusion architecture. In a sensing scenario data is organized by a forest of dynamic trees. Each tree represents the information known about an object sensed in the environment. Within each dynamic tree, sensor measurements (leaf nodes) are linked to the true state of the object (root nodes) through an intermediate set of nodes that define the uncertainty interjected by the particular sensor.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

finding the sensor fusion solution are then described. The remaining sections explain the incorporation of target and sonar image texture features into the DT framework for sensor measurement discrimination and include results from a simulated sensing scenario.

II. Dynamic Tree Graphical Model

In a Bayesian network (BN) graphical model, the factorization of the joint probability of a set of random variables (or nodes) is defined by the connections between nodes. In Fig. 2 below a set of random variables $\{A,B,C,D\}$ is connected as shown. A directed link from parent node to child nodes defines the probability state of the child conditioned on the parent. For example, the link labeled with a 1 in Fig. 2 denotes the conditional probability $P(C|A)$. Root nodes or nodes which do not have a parent are defined through an unconditional prior probability. Using this notation the joint probability of the graph in Fig. 2 can be factored as

$$P(A,B,C,D) = P(A)P(B|A)P(C|A)P(D|C). \qquad (1)$$

Inferring the probability mass function of a discrete random variable or node in the model can be done by simply evaluating the evidential nodes, or nodes in which the values are known, and marginalizing or summing over all the unknown nodes. For example, to infer the probability of the states of random variable C when the values at nodes $B=b$ and $D=d$ in Fig. 2 requires a summation over the probability mass function of variable A in the numerator and A and C in the denominator of (2). This expression (i.e. the posterior) can be written according to Bayes' rule as

$$P(C \mid B=b, D=d) = \frac{\sum_{a \in A} P(a)P(B=b \mid a)P(C \mid a)P(D=d \mid C)}{\sum_{a \in A} P(a)P(B=b \mid a)\sum_{c \in C} P(c \mid a)P(D=d \mid c)} \qquad (2)$$

The computational complexity of marginalization increases as the number of nodes and links increases in the BN. In most inference cases, the probability mass function at each node in the BN needs to be updated after the introduction of new evidence into the network. For tree-structured networks, i.e. child nodes have only one parent, the marginalization or belief update at each node is carried out using a fast sum-product algorithm known as belief updating or Pearl's message passing [4]. The belief updating algorithm greatly reduces computation time and provides a standard programming framework to accomplish the marginalization over the entire network.

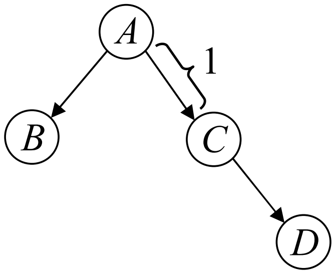


Figure 2. Bayesian network graphical model. Nodes A, B, C, and D are random variables, and links between the nodes define conditional probability relationships.

A dynamic tree (DT) graphical model [1] is a tree-structured BN whose joint probability $P(\boldsymbol{Z},\boldsymbol{X})$ includes a random variable $\boldsymbol{Z}$ that defines the linkage between nodes $\boldsymbol{X}$ in the tree. The structure or nodal linkage of the DT is defined by $\boldsymbol{Z}$ in the form of a matrix whose element in the i^{th} row and j^{th} column is written as Z_{ij}. Each element in $\boldsymbol{Z}$ is the indicator of the connection between the child node X_{i} and the parent node X_{j} in which case $Z_{\text{ij}} = 1$ if nodes X_{i} and X_{j} are connected, otherwise $Z_{\text{ij}} = 0$. Each element in the set of N nodes $\boldsymbol{X} = \{X_1, X_2, \ldots, X_{\text{N}}\}$ is a discrete random variable with a probability mass function defining the probability of nodal states. Further, the nodes in $\boldsymbol{X}$ are members of either of the 2 disjoint subsets $\boldsymbol{X}_{\text{h}}$ or $\boldsymbol{X}_{\text{e}}$ ($\boldsymbol{X} = \{\boldsymbol{X}_{\text{h}} \cup \boldsymbol{X}_{\text{e}}\}$) which are hidden nodes and evidential/observable nodes respectively. The conditional probability table (CPT), denoted by θ_{ij}, represents the state transition probabilities, or $P(X_{\text{i}}|X_{\text{j}})$, between any two connected nodes X_{i} and X_{j}. To complete the DT model, root nodes are assigned an appropriate prior probability.

The joint probability $P(\boldsymbol{Z},\boldsymbol{X})$ defines all possible nodal states and nodal linkages (hereafter called structures) of the DT. The most desirable structure of the DT is the one that maximizes $P(\boldsymbol{Z},\boldsymbol{X})$. The solution is equivalently found by maximizing the *a posteriori* (MAP) probability $P(\boldsymbol{Z},\boldsymbol{X}_{\text{h}}|\boldsymbol{X}_{\text{e}})$ [1]. In terms of the sensor fusion problem we propose in this paper, the MAP solution yields a non-redundant set of object root nodes connected to the most likely sensing platform nodes, which are connected to the most likely set of measurements that originate from the particular sensing platforms. Hence, the most important part of translating a DT framework into the sensor fusion problem is to define the sensor and object relationships in terms of the structure variable $\boldsymbol{Z}$ and the CPTs that define the internodal relationships so that the DT solutions are within the realm of possibility, and more importantly are the most likely answer given *a priori* constraints. The next section will discuss the sensor fusion problem within the framework of DTs and detail the steps necessary to optimize the structure given sensor measurements on leaf nodes of each tree.

III. Sensor Fusion Framework

In the sensor fusion DT framework, the root nodes of the DT are random variables that define the true state of the sensed objects or targets, nodes that correspond to a particular sensor make up the intermediate layer of the tree, and leaf nodes or nodes at the bottom of the tree take the value of raw sensor measurements. Using this framework, an algorithm was developed that takes the raw sensor measurements and sensor identification from an underwater sensing mission and automatically builds a DT for each unique object using a structural optimization algorithm combined with belief updating. The algorithm outputs a forest or group of DTs, each having an inferred probability distribution of the true contact location. Thus the DT sensor fusion framework yields two valuable pieces of information: 1) an object location in terms of a probability mass function and 2) a parent-child relationship that links the object to the sensing platforms and raw measurements that gathered the information about its state.

In the sensor fusion framework presented here the solution to the DT is found by maximizing an objective function $J(\boldsymbol{Z},\boldsymbol{X})$ that is the linear combination of 3 DT log joint pdfs or

$$J(\boldsymbol{Z},\boldsymbol{X}) = w_1 \ln P_1(\boldsymbol{Z},\boldsymbol{X}) + w_2 \ln P_2(\boldsymbol{Z},\boldsymbol{X}) + w_3 \ln P_3(\boldsymbol{Z},\boldsymbol{X}) \cdot \quad (3)$$

The nodes in each of the trees are random variables representing the contact location (DT 1), sonar image texture features (DT 2), and object features (DT 3) respectively. (Note: the maximization using the three DTs is analogous to assuming independence between the random variables describing location, texture, and object features.) The structure or nodal linkage in the three DTs is assumed to be the same, i.e. if the nodal linkage in one tree is changed then the two remaining trees undergo the same structural change. Fig. 3 is an illustration of the parallel DT structure.

Raw sensor information that is ultimately used to resolve the structure of the tree occupies the leaf nodes of DT 1, DT 2, and DT 3. In DT 1 the geographic position of the object is taken directly from the sensor measurement. In DT 2 three sonar image texture parameters of correlation length in the *x*-direction, correlation length in the *y*-direction, and the *K*-distribution shape parameter are extracted via the algorithm described in [5]. In DT 3 the object size and shape features are extracted by a correntropy-based technique which is detailed in [6]. The measurements in the leaf nodes assume values as soft evidence [7] which means the leaf node receives information in the form of a probability distribution rather than a fixed value or hard evidence. For DT 1, the contact location measurement of the sensor for the object in question is assumed to be a discretized multivariate Gaussian random variable with the mean centered about the measured geographic location and a co-variance determined by the navigational accuracy of the sensor. For DT 2, the texture measurements are assumed to be drawn from a discretized multivariate Gaussian distribution with a covariance commensurate with the number of samples used to estimate the texture features [8]. For DT 3, the object feature measurements are assumed to be drawn from a discretized multivariate Gaussian distribution as well.

For a fixed structure $\boldsymbol{Z}$ and evidence on the leaf nodes, $P(\boldsymbol{X}|\boldsymbol{Z})$ is determined solely by the belief update or message passing algorithm [4], [7]. In the sensor fusion framework described here we assume a uniform prior on each root node. A CPT is generated between intermediate level X_I and leaf nodes X_L by assigning a multivariate Gaussian with a mean at each soft evidence value in the discretized grid. The co-variance of the CPT between the root and intermediate nodes for DT 1 is defined by the navigational uncertainty of the sensing platform. The CPT between the intermediate level and the leaf level in DT 2 and DT 3 represents the influence of the parameters used to discretize the multivariate normal soft evidence into a discretized multivariate pdf.

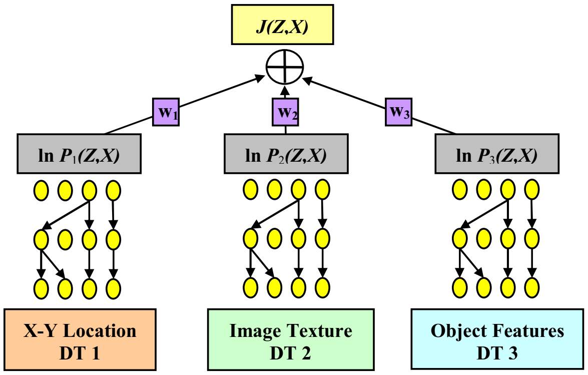


Figure 3. Parallel DT sensor fusion structure. The objective function J(Z,X) is a weighted sum of the DTs formed for the X-Y location, image texture, and object feature measurements.

Given a set of measurements, the optimal graph structure is one in which the objective function $J(\boldsymbol{Z},\boldsymbol{X})$ in (3) is maximized. A general algorithm for finding the optimal graph structure follows:

1. Input parameters: $P(\boldsymbol{Z})$ where $\boldsymbol{Z} \in \{\boldsymbol{Z}_1, \boldsymbol{Z}_2, \dots, \boldsymbol{Z}_N\}$ is a set of N possible graph structures, soft evidence on leaf nodes $\boldsymbol{X}_L$, weights w_1, w_2, w_3, and CPTs between leaf and intermediate nodes θ_{LI}.
2. For $i = 1:N$
   a. Assign structure $\boldsymbol{Z}_i$.
   b. Calculate $P(\boldsymbol{Z}_i)$.
   c. Calculate maximum a posterior value of $P_k(\boldsymbol{X}|\boldsymbol{Z}_i)$, for $k = 1,2,3$ using belief propagation.
   d. Calculate $J(\boldsymbol{Z}_i,\boldsymbol{X})$
3. Maximum value of $J(\boldsymbol{Z},\boldsymbol{X})$ over all i gives globally optimal solution.

IV. Dynamic Tree Initialization and Optimization

For this DT framework a Bell number [9] calculation yields a computationally infeasible solution for a globally optimal exhaustive search over all possible DT structures for even a small number of input nodes. For example, with only 10 measurements from a single sensor, the possible number of DT structures is 115,975. Thresholding by geographic proximity and initializing the structure via unsupervised clustering reduces the possible number of DT solutions and thus the search over the possible structure types is less time-consuming. A proximity constraint is reasonable because it is not necessary to fuse contacts that are too distant from one another assuming a reasonable accuracy in the sensor's navigation system. The following subsections describe the DT initialization and optimization algorithm that yields the local maximization of the objective function $J(\boldsymbol{Z},\boldsymbol{X})$ under these *a priori* constraints.

A. Geographic Data Clustering via LBG-VQ

The data is initially clustered by geographic proximity with the LBG-VQ algorithm [10], [11]. The algorithm yields subgroups of sensor measurements that are in close proximity to one another and aims to produce an initial DT forest structure that is near a local maximum of (3). LBG-VQ is an iterative vector quantization algorithm that meets two criteria: 1) the nearest neighbor condition, that is, each member of a cluster is assigned a label that corresponds to the closest centroid to the member and 2) a distortion criteria, i.e. the sum of the distances from all the points in a cluster to a valid centroid is less than some threshold value.

A sample problem is now introduced to illustrate the use of the LBG-VQ clustering algorithm in the DT initialization. Referring to Fig. 4, an initial map of contacts are detected with sensors from two different sensing platforms and have locations defined by x and y. In Step 1 of the algorithm, the LBG-VQ algorithm clusters objects on the x-y input space with an output that assigns centroids (stars) to possible geographic centers of the objects. In Step 2, a DT forest is initialized with the centroids as the root nodes and the object locations as the leaf nodes. In Step 3, the leaf nodes are partitioned by centroid assignment. In Step 4, sensing platform nodes are assigned at the intermediate level depending on which sensing platform produced measurements. In Step 5, the centroids are linked to the leaf node partitions through the intermediate platform ID nodes to initialize the DT forest.

B. Dynamic Tree Optimization via Simulated Annealing

After the DT forest is initialized by the LBG-VQ clustering algorithm, the locally optimum structure is calculated using simulated annealing. Simulated annealing (SA) is a stochastic search algorithm used to find maxima in large optimization problems and draws its name from the method of allowing magnetic systems to find low-energy structures through annealing [12], [13]. In the DT sensor fusion framework, parent-child connections are changed at random and accepted if either the new structure results in a higher log posterior or randomly exceeds a threshold governed by the decreasing "temperature" of the annealing process. A proximity constraint prevents a random change to an impossible structure based on the geographic location of the candidate nodes. The stochastic nature of SA allows for "unfavorable" structures or low MAP values frequently in early iterations of the algorithm in hopes of avoiding local maxima. At later iterations, the algorithm performs like a stochastic greedy search and with high probability only accepts new structures that increase the MAP value. The structure acceptance is governed by a temperature parameter that decreases as the number of iterations increases.

Following the outline of the DT optimization algorithm described in Section III, a summary of the DT sensor fusion algorithm is presented in Fig. 5. The algorithm requires inputs of the raw sensor measurements as probability distribution functions, an initial structure, and an annealing schedule with an initial temperature.

V. Experiment and Results

To demonstrate the ability of the DT algorithm to automate a sensor fusion task, a scenario was simulated where two different sensing platforms collected a total of 28 sonar images via two different tracks or passes along an object field. A total of 7 objects were sensed varying numbers of times by the two sensing platforms. Fig. 6 depicts the sensing scenario and the groupings {1,2} and {3,4,5} were specifically included to test the ability of the DT algorithm in resolving multiple object detections that are within the navigational uncertainty of the sensing platform. The overlapping red dashed boxes in Fig. 6 represent a notional uncertainty of the object locations. This uncertainty manifests itself as an ambiguity with regards to object uniqueness in the sensor fusion task.

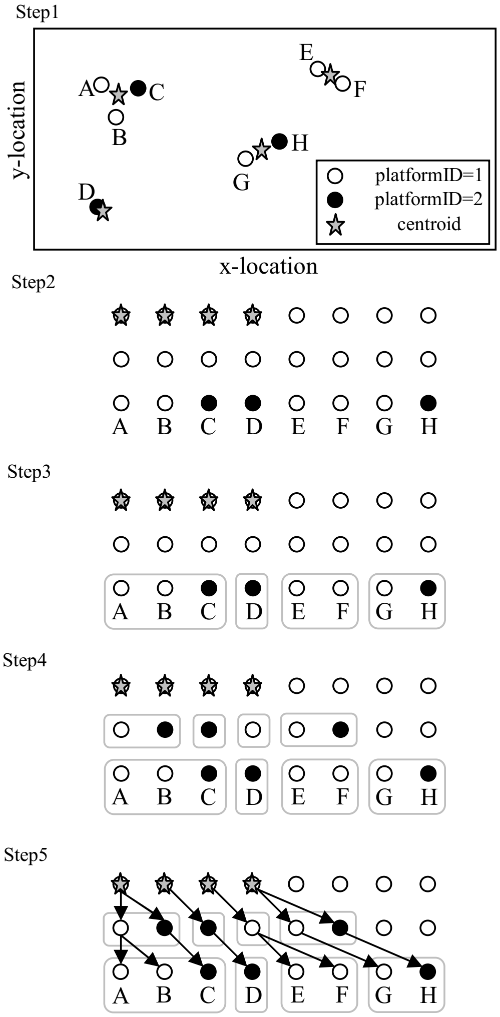


Figure 4: LBG-VQ clustering to initialize the DT forest. Contacts are first clustered by contact location then connected into an initial tree based on centroid membership and sensing platform association.

Side-look sonar images were generated by inserting physics-based object simulations [14] into textured backgrounds generated from a correlated K-distribution [5]. Simulated images were accurately generated for range and sensing aspect angle for the two platforms in the scenario. As an example, the series of images in Fig. 7 depicts the simulations of Object 1 in Fig. 6. (Note: The white dots in the images are fixed in x-y geographic coordinates as a reference to the different sensing aspect angles.)

```
1. Initialize parameters
     a. Accepted structure changes $A = 0$
     b. Proposed changes $P = 0$
     c. Max number of structure changes $A_{\text{max}}$
     d. Max number of total changes $P_{\text{max}}$
     e. Initial temperature $t = t_0$
     f. Minimum temperature $t_{\text{min}}$
     g. Temperature decay constant $K, (0<K<1)$
2. Initialize parallel DT structure via LBG-VQ procedure in Section IV-A
3. Calculate $J(\mathbf{Z},\mathbf{X})$ using belief propagation
4. Set $E_a = J(\mathbf{Z},\mathbf{X})$
5. Randomly change a node's parent connection in the DT under proximity constraints
6. Recalculate $J(\mathbf{Z},\mathbf{X})$ using belief updating
7. Set $E_b = J(\mathbf{Z},\mathbf{X})$
8. Set $P_c = exp((E_b - E_a)/t)$
9. IF $E_b > E_a$, accept node connection change; $E_a = E_b$, $A = A + 1$; $P = P + 1$
10. ELSEIF $P_c$>Uniform RV (0,1), accept node connection change; $E_a = E_b$, $A = A + 1$; $P = P + 1$
11. ELSE $P = P + 1$
12. ENDIF
13. IF $P > P_{\text{max}}$ OR $A > A_{\text{max}}$; $t = Kt$; $A = 0; P = 0$;
14. ENDIF
15. IF $t<t_{\text{min}}$ AND $A =0$; END Routine.
16. ELSE Return to Step 5.
17. ENDIF.
```

Figure 5: Summary of the DT optimization algorithm.

DT 1, DT 2, and DT 3 of Fig. 3 were initialized with the respective contact location, the image texture measurement, and object feature discretized probability density function (pdf). The multivariate discrete probability mass function obtained from the measurements is produced by discretizing the continuous multivariate normal pdf to a multidimensional matrix so that we can use a discrete inference framework. The contact location pdf was assumed drawn from a multivariate normal distribution with a mean equal to the x-y location and a diagonal covariance matrix $\sigma^2 I$, where $\sigma = \sigma_x = \sigma_y$. The sonar image texture features of x correlation length, l_x, and y correlation length, l_y, were estimated from the sonar images using the procedure described in [5]. To form a multivariate pdf using the features, the feature set was assumed drawn from a multivariate normal distribution with a mean equal to the estimated values of the features and a diagonal covariance matrix with a common standard deviation of σ_t. Object features were extracted from the images using the technique described in [6] and a multivariate normal distribution was created using the same procedure used for the sonar image texture pdf.

The DT algorithm was initialized and tested against the simulated data fusion scenario for varying levels of uncertainty in the sensing platforms navigation sensors and in the feature estimates. For each experiment, the parallel DT structures were initialized using the LBG-VQ algorithm. $P(\mathbf{Z})$ was assigned a uniform distribution, i.e. all linkages are equally likely thus simplifying the calculation of $J(\mathbf{Z},\mathbf{X})$. The weights of $J(\mathbf{Z},\mathbf{X})$ were empirically chosen to be $w_1 = 0.6$, $w_2 = 0.3$, and $w_1 = 0.1$ based on results from a previously simulated scenario as a training example. The variances of the multivariate normal distributions were varied to simulate different levels of navigational error and uncertainty in feature estimates.

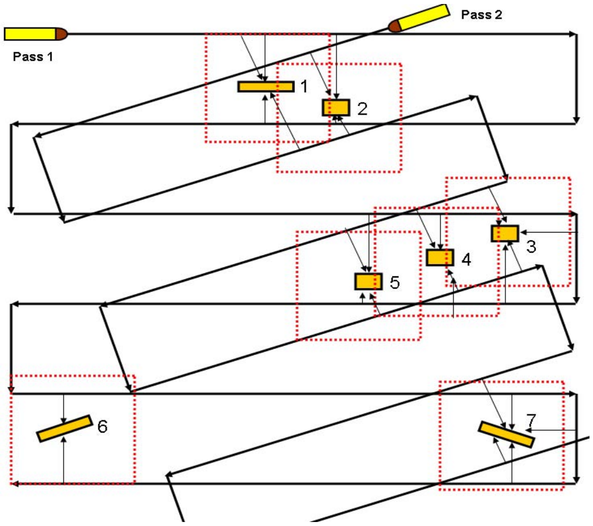
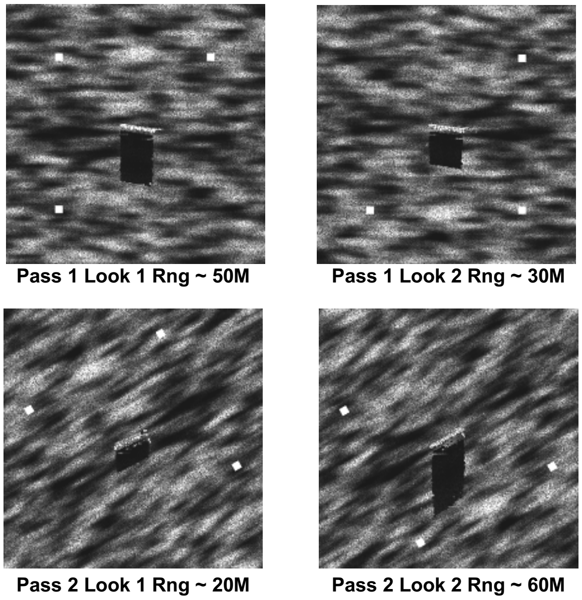


Figure 6. Simulated data fusion scenario. In the scenario, two sensing platforms make a pass through a field of seven targets at different trajectories. Red boxes around the targets represent the notional uncertainty in the objects true location, hence the need to use additional features to discern between sensed objects.

Table I organizes the results of the experiment by standard deviation in the navigation and feature estimate values. The tabulated results have two metrics. The first metric is the number of correct object calls; or rather do all of the objects in the scenario have at least one node assigned in the top level of the DT. For the first metric, the notation (+3) means at least three of the object nodes have a duplicate in the top level of the tree. The second metric is whether all the leaf nodes of the DT link to the correct root node, e.g. 25/28 means 3 leaf nodes were linked incorrectly to the ground truth root node. As expected, the best performance was obtained in the lowest noise case, $\boldsymbol{\sigma}_{\mathbf{N}} = (0.25, 0.5, 25)$ and $\boldsymbol{\sigma}_{\mathbf{F}} = (0.25, 0.5, 25)$, where $\boldsymbol{\sigma}_{\mathbf{N}}$ and $\boldsymbol{\sigma}_{\mathbf{F}}$ are the standard deviation of navigation uncertainty and feature uncertainty respectively. All the objects root nodes have been assigned correctly with three duplicates and all the leaf nodes are all linked to the correct root nodes. In the case of high noise, $\boldsymbol{\sigma}_{\mathbf{N}} = (5, 10, 200)$ and $\boldsymbol{\sigma}_{\mathbf{F}} = (5, 10, 200)$, two objects are merged in the top level of the tree and 8 leaf nodes are linked to the incorrect root node.

The computational cost of the algorithm depends on the number of leaf nodes within a subgroup and the resolution of the discretized CPTs and discretized features at the leaf nodes. In practice, the problem of the number of nodes in a subgroup largely dominates the latter factors. Table II below shows the computation time required for DT structure optimization as presented in this paper as a function of the number of nodes



Pass 1 Look 1 Rng ~ 50M

Pass 1 Look 2 Rng ~ 30M

Pass 2 Look 1 Rng ~ 20M

Pass 2 Look 2 Rng ~ 60M

Figure 7. Image examples for Object 1 in the data fusion scenario. Object 1 is a cylindrical target sensed at a broadside and off-center angle from the two sensing passes. The white dots in the images are used as reference points for image alignment.

in a single tree. The times were computed using the scenario data above on a desktop PC running MATLAB®. Although Table II is not a comprehensive benchmark for DT optimization algorithms, it does illustrate the explosion of computational complexity of finding locally optimal DT structures with more than just a few nodes.

VI. Conclusions

A dynamic tree sensor fusion architecture was presented to automate the task of combining like objects that have discriminating measurements. A description of the DT structure optimization algorithm including its initialization by LBG-VQ was explained in detail. An experimental scenario of two sensing platforms traversing an object field was simulated at two different sensing angles. Measured geographic locations and image features were extracted and used to fuse redundantly sensed objects via optimization of the DT structure or nodal linkage. Data fusion results for various levels of sensor noise were presented and the DT fusion method was shown to be effective for the scenario presented. The fusion error rate for the algorithm presented was shown to be approximately proportional to the level of noise in the feature measurements and the sensing platform navigational error. A short discussion of the computational complexity of the DT structure optimization as a function the number of nodes in the network was presented in closing.

TABLE I. Experimental Results for Data Fusion Scenario

				Navigation uncertainty (σ_N)			
			f1	0.25	0.5	1	5
			f2	0.5	1	2.5	10
Feature noise (σ_F)	f1	f2	f3	25	50	75	200
	0.25	0.5	25	7/7 (+3) 28/28	7/7 (+2) 26/28	6/7 20/28	5/7 20/28
	0.5	1	50	7/7 (+3) 27/28	7/7 (+2) 26/28	6/7 20/28	5/7 20/28
	1	2.5	75	7/7 (+4) 27/28	7/7 (+1) 24/28	6/7 20/28	5/7 20/28
	5	10	200	7/7 (+4) 25/28	7/7 (+4) 25/28	7/7 (+3) 25/28	5/7 20/28

Acknowledgment

The authors thank the Office of Naval Research for financial support of this research.

References

[1] N. Adams, C. Williams, “Dynamic Trees for Image Modelling,” *Image and Vision Computing*, vol. 21, pp. 865-877, 2003.

[2] S. Todorovic and M. C. Nechyba, "Dynamic trees for unsupervised segmentation and matching of image regions," *IEEE TPAMI,* vol. 27, pp. 1762-1777, November 2005..

[3] A. J. Storkey and C. K. L. Williams, "Image modeling with position-encoding dynamic trees," *IEEE TPAMI,* vol. 25, pp. 859-871, 2003.

[4] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan-Kauffman, 1988, pp. 239-255.

[5] J. T. Cobb and K. C. Slatton, "A Parameterized Statistical Sonar Image Texture Model," *Proc. SPIE Defense and Security Symposium,* vol. 6953, March 2008.

[6] E. Hasanbelliu, “Mine Classification Using Correntropy-Based Matched Filter,” *Systems, Man, and Cybernetics 2009*, in press.

[7] F. V. Jensen and T. D. Nielsen, *Bayesian Networks and Decision Graphs*, 2nd ed.: Springer, 2007.

[8] C. Oliver, “The Sensitivity of Texture Measures for Correlated Random Clutter,” *Inverse Problems*, vol. 5, pp. 875-901, 1989.

[9] E. W. Weisstein, "Bell Number," From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/BellNumber.html.

[10] S. Lloyd, "Least squares quantization in PCM," *Information Theory, IEEE Transactions on,* vol. 28, pp. 129-137, 1982.

[11] Y. Linde, A. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design," *Communications , IEEE Transactions on,* vol. 28, pp. 84-95, 1980.

[12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, pp. 671-680, 1983.

[13] R. Duda, P. Hart, D. Stork, Pattern Classification, 2nd ed., Wiley, 2001, pp. 351-357.

[14] G. Sammelmann, J. Christoff, J. Lathrop, “Synthetic Images of Proud Targets,” *OCEANS ’94 Proceedings*, vol. 2, pp. 266-271, Sep. 1994

TABLE II. Computational Time

Number of nodes per tree	**Computational time per tree**
2	~20 seconds
3	~20 seconds
4	~80 seconds
5	5 minutes
8	29 minutes
9	35 minutes