Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Process Model, Constraints, and the Coordinated Search Strategy

Frédéric Bourgault, Tomonari Furukawa and Hugh F. Durrant-Whyte
ARC Centre of Excellence in Autonomous Systems (CAS)
The University of Sydney, Sydney, NSW 2006, Australia
Email: {f.bourgault, t.furukawa, hugh}@cas.edu.au

Abstract— **This paper deals with the problem of coordinating a team of mobile sensor platforms searching for a single mobile non-evading target. It follows the general Bayesian active sensor network approach introduced in [2] where each decision maker plans locally based on an equivalent representation of the target state probability density function (PDF). This paper focuses on the prediction stage of the decentralized Bayesian filter. It looks at how different types of realistic external constraints may affect the target motion and how they may be taken into account in the process model. Two general classes of constraints are identified: soft and hard. A few constraint examples from each class are given to illustrate their impact on the evolution of the target state PDF. Multiple constraints of various types can be combined to increase the accuracy of the predicted PDF estimate, thus affecting the individual trajectories of the search platforms. The effectiveness of the framework is demonstrated for a team of airborne search vehicles looking for a drifting target lost in a storm at sea.**

I. Introduction

When rescue authorities receive a distress signal time becomes critical. When lost at sea, survival expectancy decreases rapidly and the primary goal of a rescue mission is to search for and find the castaways as diligently and efficiently as possible. The search, based on some coarse estimate of the target location, must often be performed in low visibility conditions with strong winds and high seas causing the location estimate to grow even more uncertain as time goes by. How this location estimate evolves and exactly how uncertain it becomes in time is the first and foremost important question that needs answering before a large team of heterogeneous platforms, e.g. high flying long range aircrafts, helicopters and ships equipped with multiple sensors could be deployed for the search.

In [2] a general Bayesian active sensor network approach to the target detection problem as described in [8] (Chapter 9) was presented. It expanded the single vehicle framework proposed in [3] to an arbitrary number of sensing platforms by integrating a fully decentralized Bayesian data fusion technique with a decentralized coordinated control scheme. In this architecture, each sensor node communicates observations on the network and builds an equivalent representation of the target state probability density function (PDF). Planning is done locally by the decision makers based on the latest PDF update and the resulting control actions are consistent and coordinated without further communication about the plans. A similar control coordination framework based on a decentralized information filter can be found in [7]. A high degree of scalability, modularity and real-time adaptability are the advantages of the decentralized approach. At any time, new rescue vehicles can join the search effort, or momentarily quit for refuelling, and the system should seemly and robustly adapt to the change.

This paper deals with the target motion prediction aspect of the general Bayesian filter. In particular it focuses on state dependent process models and on how to integrate external motion constraints of different types into the prediction stage. Two general classes of constraints are identified: strong and weak. A few constraint examples from each class are given to illustrate their impact on the evolution of the target sate PDF. Multiple constraints of various types can be combined to increase the accuracy of the predicted PDF estimate, thus affecting the individual trajectories of the search platforms.

The breakdown of the paper is as follows. Firstly, the decentralized Bayesian filtering algorithm that accurately maintains and updates the target state PDF is described in the next section. Then Section III describes two general classes of target motion constraints, i.e. soft and hard, and how they can be integrated into the prediction stage in the form of state dependent transition probability models. Then in Section IV, the effectiveness of the approach is demonstrated for multiple airborne vehicles searching for a drifting target subject to combined constraints of both soft and hard types. The coordinated solutions for different numbers of vehicles are compared illustrating the scalability of the framework. Finally, conclusions and ongoing research directions are highlighted in the last section.

II. Decentralised Bayesian Filtering

This section reviews the mathematical formulation of the decentralized Bayesian data fusion algorithm used to predict and update the target state PDF. The Bayesian approach is particularly suitable for combining in a rational manner heterogeneous non-gaussian sensor observations with other sources of quantitative and qualitative information [1], [9].

In Bayesian analysis any unknown quantity of interest is considered a random variable. The state of knowledge about such a random variable is entirely expressed in the form of a PDF. New information in the form of a probabilistic measurement or observation is combined with the previous PDF using the Bayes theorem in order to update the state of

0-7803-8232-3/04/$17.00 ©2004 IEEE

knowledge. This newly updated PDF forms the quantitative basis on which all inferences, or control decisions are made.

In the searching problem, the unknown variable of interest is the target state vector $\mathbf{x}_k^t \in \mathbb{R}^{n_x}$ which in general describes the target location but could also include its attitude, velocity, and other properties. The purpose of the analysis is to find an estimate for $p(\mathbf{x}_k^t|\mathbf{z}_{1:k})$, the PDF of $\mathbf{x}_k^t$ given the sequence $\mathbf{z}_{1:k} = \{\mathbf{z}_j^i : i = 1, ..., N_s, j = 1, ..., k\}$ of all the observations made from the N_s sensors on board the search vehicles, $\mathbf{z}_j^i$ being the observation from the i^{th} sensor at time step j. The analysis starts by determining a prior PDF $p(\mathbf{x}_0^t|\mathbf{z}_0) \equiv p(\mathbf{x}_0^t)$ for the target state at time 0, given all available prior information including past experience and domain knowledge. If nothing is known other than initial bounds on the target state vector, then a least informative uniform PDF is used as the prior. Once the prior distribution has been established, the PDF at time step k, $p(\mathbf{x}_k^t|\mathbf{z}_{1:k})$, can be constructed recursively using two equations alternatively: prediction and update.

A. Prediction

A prediction stage is necessary in Bayesian analysis when the PDF of the state to be evaluated is evolving with time i.e. the target is in motion or the uncertainty about its location is increasing. Suppose the system is at time step $k-1$ and the latest PDF update, $p(\mathbf{x}_{k-1}^t|\mathbf{z}_{1:k-1})$, is available. Then the predicted target state PDF at time step k is obtained by the following Chapman-Kolmogorov equation

$$p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t)p(\mathbf{x}_{k-1}^t|\mathbf{z}_{1:k-1})d\mathbf{x}_{k-1}^t \quad (1)$$

where $p(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t)$ is a probabilistic Markov motion model. Also referred to as the process model, it describes the probability of transition of the target states. Deriving the process model from the equations of motion of the target and the probability distribution on their inputs is out of the scope of this paper. Instead, Section III will show how various types of motion constraints might affect the process model and the evolution of the target PDF.

B. Update

At time step k a new set of observations $\mathbf{z}_k = \{\mathbf{z}_k^1, ..., \mathbf{z}_k^{N_s}\}$ becomes available and the update is performed using the Bayes rule where all the observations are assumed to be independent. In other words, the update is performed by multiplying the prior PDF (posterior from the prediction stage) by all the individual conditional observation likelihoods $p(\mathbf{z}_k^i|\mathbf{x}_k^t)$ as in the following

$$p(\mathbf{x}_k^t|\mathbf{z}_{1:k}) = K\, p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1}) \prod_{i=1}^{N_s} p(\mathbf{z}_k^i|\mathbf{x}_k^t) \quad (2)$$

where the normalization coefficient K is given by

$$K = 1/\int \left[p(\mathbf{x}_k^t|\mathbf{z}_{1:k-1}) \prod_{i=1}^{N_s} p(\mathbf{z}_k^i|\mathbf{x}_k^t)\right] d\mathbf{x}_k^t \quad (3)$$

C. Active Bayesian Sensor Network

In an information gathering task such as searching, if each sensor is connected to a processing unit called a node, then it is possible through communication and fusion of the information to reconstruct at each node the global information state of the world, e.g. the target state PDF. Figure 1 depicts how the update and prediction equations are integrated in the general Bayesian sensor node of a fully connected network. Mounting the sensor node unto an actuated mobile Platform and coupling it to its own Controller makes it an active sensor. Based on the latest belief about the world $p(\mathbf{x}_{k-1}^t|\mathbf{z}_{k-1}^t)$ and the sensor state $\mathbf{x}_{k-1}^{s_i}$, the Controller sends a command $\mathbf{u}_{k-1}^{s_i}$ to the Platform to place the sensor in a desired position $\mathbf{x}_{k_{des}}^{s_i}$ with respect to the world to take the next observation.

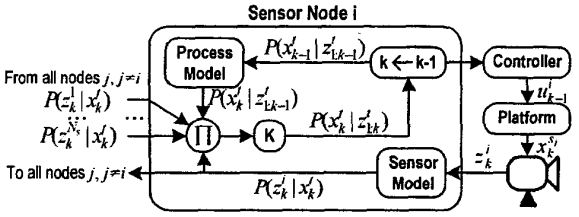

Fig. 1. Active Bayesian sensor node algorithm for a fully connected network.

There is more than one valid way to implement the Bayesian filtering algorithm. For example, it is possible to represent the the target PDF using parametric functions and to perform the prediction and update stage by updating the parameters of the function. If the target PDF as well as the process model are both Gaussian, then the most effective parametric filter is the well known Kalman filter. For the searching problem however, the process model, and especially the target PDF can be highly non-Gaussian and the complete description of the density function must be maintained. In this paper the prediction and update equations will be evaluated numerically using a grid based discrete approximation of the process model, the observation likelihood and the target PDF.

III. Process Model and Constraints

It can be shown that if the process model is invariant over the target states, the prediction equation (1) boils down to a convolution operation. Practically, this convolution is performed numerically by a discrete approximation of the two PDF's on a grid, followed in sequence by the multiplication of their Fast Fourier Transforms (FFTs), and by an inverse FFT of the product to retrieve the result.

In general, the target motion model is not invariant throughout the state space. This is the case when the target motion is affected by external constraints. Then, for any given prior state $\mathbf{x}_{k-1}^t$, the probability of transition, $p(\mathbf{x}_k^t|\mathbf{x}_{k-1}^t)$, must be described by a distinct probability distribution over the subsequent states $\mathbf{x}_k^t$ and take into account all of the motion constraints. In practice, these state transition PDF's are evaluated and tabulated in an array containing $(n_{dx})^{n_{state}} \times (n_{dx})^{n_{state}}$ elements, where n_{state} is the number of target sates, n_{dx} corresponds to the number of discrete elements per state, and $(n_{dx})^{n_{state}}$ is the number of elements in the grid

representing the target PDF. Notice that when the resolution and/or the number of states increase, the size of the array can become very large. Nevertheless, with careful selection of the parameters, the technique can still be applied in a meaningful way to a variety of problems.

In this section, two classes of motion constraints are identified. The first group consists of all the strong motion restrictions, e.g. motion restricted on a curve or a plane, which also includes all kinds of un-traversable obstacles that will be referred to as hard constraints. The second type regroups all the other motion impeding constraints not included in the first, and will be referred to as the soft constraints. A physical problem could contain a mixture of a variety of soft and hard constraints.

A. Hard Constraints

Hard constraints consist of two types of obstacles that are not traversable by the target. In the first type, for every given prior state, all the probabilities corresponding to state transitions through an obstacle are set to zero and the distribution is re-normalized to a volume of one. Examples of these sort of constraints include solid objects too high for the target to cross over or un-traversable terrain topology such as cliffs or deep ravines. Hard constraints of the second type display what we call a 'netting' or accumulating effect. For these constraints, all the probabilities corresponding to transitions through the obstacle are again set to zero, but the corresponding probability mass is integrated and redistributed at the obstacle frontier facing the prior state location. Examples of this type of hard constraints could include any kind of nets and webs, or any other sort of obstacles that can be traversed by the medium carrying the target, but not by the target itself.

The following subsections illustrate examples of both types of hard constraints. In both cases, the initial target PDF is a symmetric Gaussian distribution centered at (0,0) with a standard deviation in x and y of 400 meters. The nominal unconstrained motion model is also a symmetric Gaussian distribution centered around the given prior state with a standard deviation in each direction corresponding to 0.5556m/s, i.e. a displacement standard deviation of 400m given a prediction step corresponding to 12 minutes of simulation. The scale on all the figures are in meters.

1) Type I: Un-traversable obstacle: Figure 2 illustrates the effect, through time, of the presence of an un-traversable obstacle (e.g. cliff line) on the target state PDF subjected to a diffusion process (e.g. walking randomly). The obstacle becomes clearly apparent as the PDF starts flowing around it (Fig. 2d). Multiple hard constraints can be combined to reflect for example a topographic map, a road network, or as depicted in Fig. 3, an indoor environment.

2) Type II: Un-traversable obstacle with accumulating effect: Figure 4 shows the evolution over time of the target PDF when constrained by the same obstacle as in the previous example but with the 'netting' effect. An example of this could be a piece of wood randomly drifting on water and getting 'stuck' along a protruding obstacle such as a man made

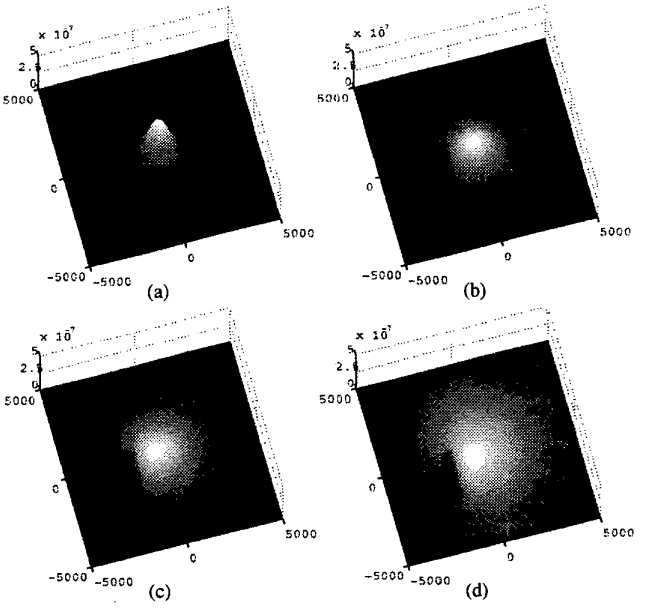

Fig. 2. Hard motion constraint of type I (random walk with a cliff line): (a) to (d) 3D views of the updated target PDF at step k = 1, 5, 10 and 30 respectively, corresponding to 12, 60, 120 and 360 minutes.

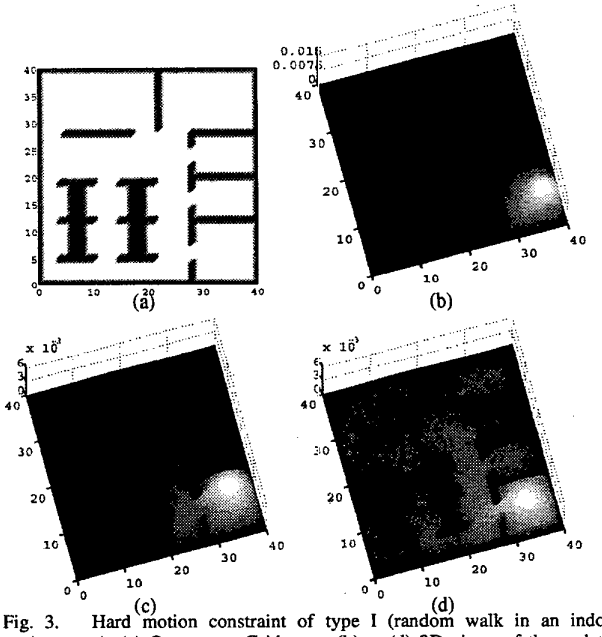

Fig. 3. Hard motion constraint of type I (random walk in an indoor environment): (a) Occupancy Grid map; (b) to (d) 3D views of the updated target PDF at step k = 1, 30 and 90 respectively, corresponding to 2, 60 and 180 seconds.

floating dock. It is quite clear from the figure that a certain amount of probability mass tends to accumulate in front of the obstacle which is quite an important thing to know when looking for a likely target location.

B. Soft Constraints

The soft constraints class includes any kind of traversable but motion impeding obstacles, such as marshlands or forested

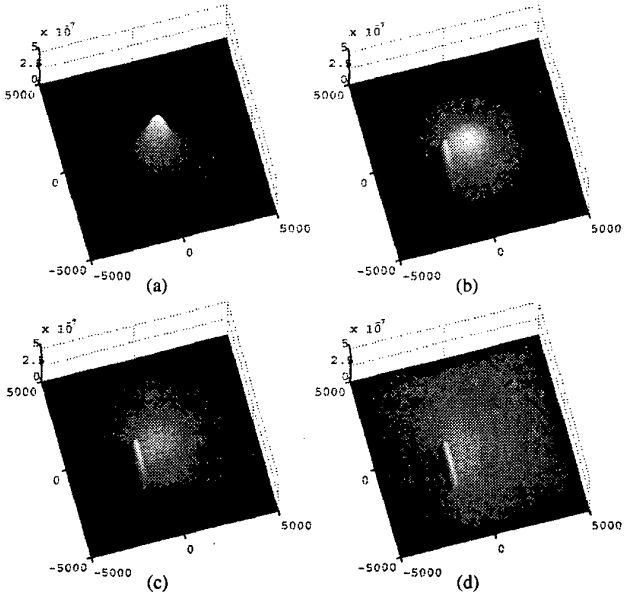

Fig. 4. Hard motion constraint of type II (obstacle with netting effect): (a) to (d) 3D views of the updated target PDF at step k = 1, 5, 10 and 30 respectively, corresponding to 12, 60, 120 and 360 minutes.

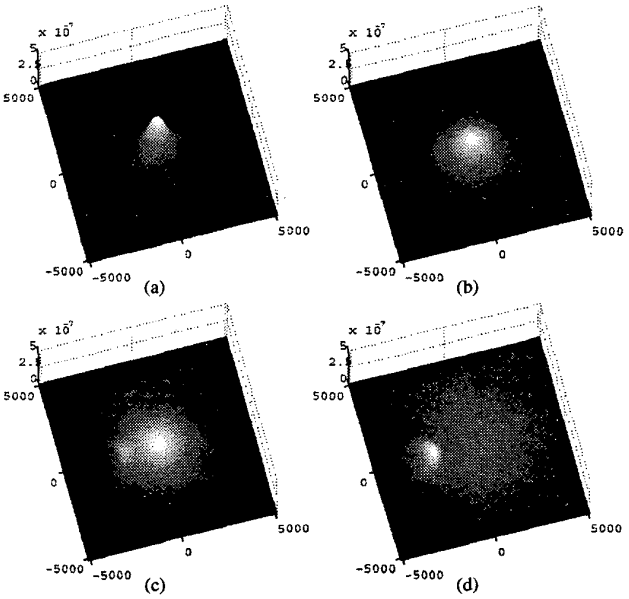

Fig. 5. Traversable obstacle (random walk in the presence of a marshland): (a) to (d) 3D views of the updated target PDF at step k = 1, 5, 10 and 30 respectively, corresponding to 12, 60, 120 and 360 minutes.

terrain, but also the more general constraints that can be described by force vector fields working either for and/or against the target motion. Wind, current and terrain topology are examples of force field type of constraints.

1) Example 1: Traversable obstacle: This example illustrates the effect of a motion impeding obstacle such as a swamp or a marshland. On Figure 5 (a) to (d), one can see the outline of a circular marshland appearing as probability mass accumulates in the region where the diffusion process is slowed down.

2) Example 2: Force fields: An examples of motion being affected by a force field is a drifting target such as a liferaft subjected to current and wind fields. Figure 6 shows a current field transportation and dissipative attributes on the target state PDF. Examples in other applications could also include electrical, magnetic or gravitational fields.

3) Example 3: Terrain topology: This example is much related to example 1, but instead of having one distinguishable soft obstacle, the whole area displays a varying degree of traversability that is a function of the terrain gradient. Such a kind of process model could be effectively used when searching for hikers lost in a mountainous terrain. Figure 7 shows the effect on the diffusion process of the mountainous terrain illustrated in Fig. 7a. In this example the nominal motion model for flat terrain is Gaussian and symmetric. Then the variance in all direction is reduced proportionally to the slope of the terrain gradient to reflect the slower walk progression in steeper terrain. Also, symmetry is broken by shortening one axis in function of the cosine of the slope angle, and then it is aligned with the slope and projected onto a 2D surface. This creates Gaussian motion ellipses with the axis aligned with the isocline of the terrain.

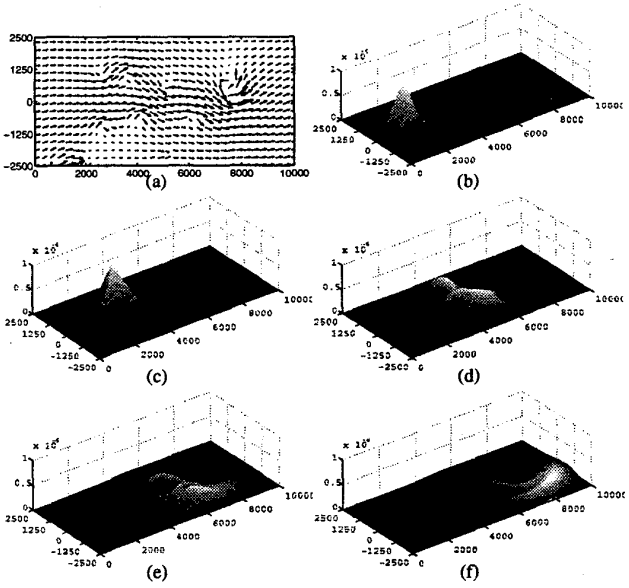

Fig. 6. Force field (drifting with the current): (a) map of the current strength and orientation, (b) to (f) 3D views of the updated target PDF at step k = 1, 5, 10, 15 and 20 respectively, corresponding to 12, 60, 120, 180 and 240 minutes.

IV. Application

Ultimately, the goal of the ongoing research effort is to demonstrate the autonomous search framework on a team of unmanned air vehicles (UAV's) (Fig. 8a). A stepping stone towards this goal is to investigate the problem using simulations. The high fidelity simulator (Fig. 8b) developed at the Australian Center for Field Robotics (ACFR) has an UAV hardware server complete with various sensor models,

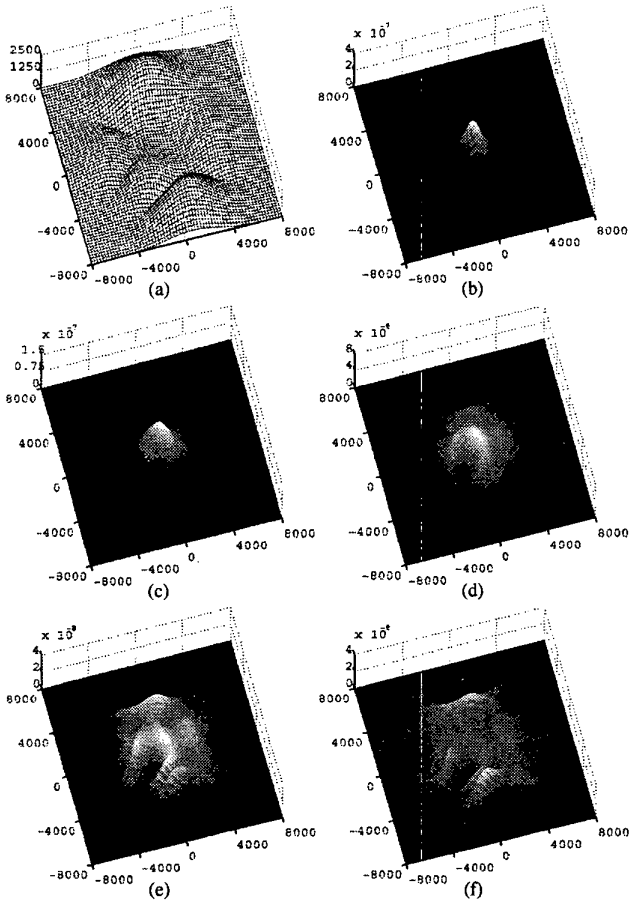

Fig. 7. Terrain topology (random walk in mountainous terrain): (a) 3D map of the terrain, (b) to (f) 3D views of the updated target PDF at step k = 1, 5, 15, 30 and 60 respectively, corresponding to 15, 75, 225, 450 and 900 minutes.

and wireless communication protocols, on which the flight software can be tested before being implemented on board the platforms almost without any modifications [4]. The rest of this section presents the results for the coordinated Bayesian search framework implemented for a team of airborne search vehicles looking for a single lost target, a liferaft, drifting in a storm at sea and in the presence of a hard obstacle such as an island. More about the implementation details of the framework and the search problem can be found in [2].

Figure 9 illustrates the results for a single vehicle greedy search, i.e 1-step lookahead control solution. Figure 9a illustrates the constraints composed of a swirly wind field and an island. Clearly seen in the sequence from Fig. 9b to c is the distortion of the initial Gaussian PDF by the wind and the accumulating effect caused by the island. Figure 10 illustrates the searching results for the same initial situation but for a team of three vehicles. On figure 10d it can be seen that for this particular case, increasing the number of searching vehicles produces better results for the same equivalent combined searching time. This happens at no extra

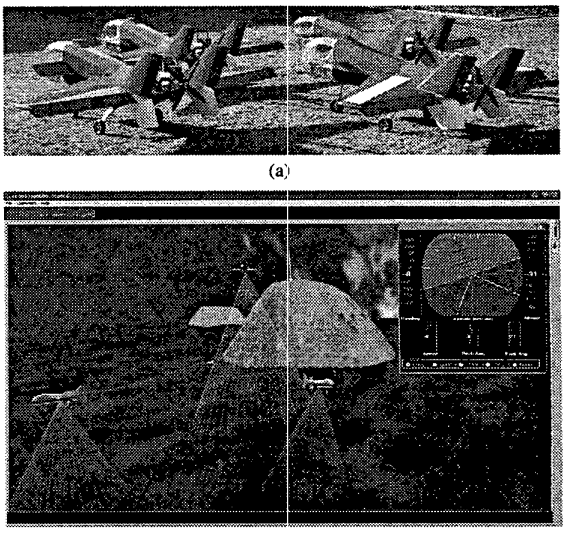

Fig. 8. (a) The fleet of Brumby Mark-III uav's been developed at ACFR. These flight vehicles have a payload capacity of up to 13.5 kg and operational speed of 50 to 100 knots. (b) Display of the high fidelity multi-UAV simulator.

computational cost for the individual decision makers and without any communication about their intentions.

V. Summary and Ongoing Work

This paper addressed the problem of integrating realistic motion constraints in the prediction stage of the active Bayesian sensor network approach to the search problem. Two main classes of constraints were identified and a few examples were given for each one. The constraints were demonstrated to greatly affect the accuracy of the target state PDF estimate, in turn affecting the search trajectories. Work is in progress to use the constraint description from a human user input. The Bayesian framework was demonstrated to adaptively find efficient coordinated search plans in a completely decentralized way. A major appeal of the approach is that nodal computation costs are kept constant regardless of their number thus offering a high potential for scalability.

Because of the nature of the search problem, it is quite important to accurately keep track of the very non-Gaussian target state PDF. However, any grid based approach such as the one presented is intrinsically subject to the "curse of dimensionality", and as soon as one needs to increase the search area, the resolution of the grid, or the number of dimensions in the state-space, computational costs tend to get out of hand. As part of the ongoing research effort, techniques such as Monte Carlo methods, or particle filters [5], as well as the so called kernel methods for density estimation are being investigated to overcome the computational limitations.

Another limitation of the technique as presented comes from the assumption that every sensor node transmits and receives every single observation without a miss via broadcasting. Beyond the obvious bandwidth limitations, such assumptions are not quite practical in real life since communication systems are plagued by delays and intermittent transmissions. To overcome

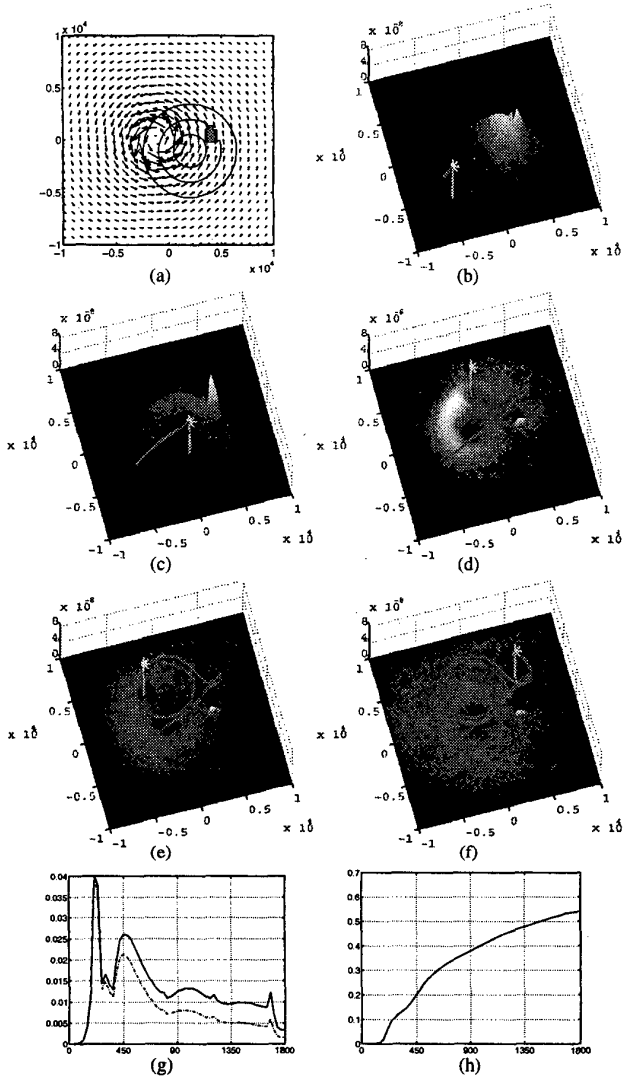

Fig. 9. One vehicle greedy search for a target drifting in a storm with a hard constraint: (a) map of the hard constraint (gray island), mean current velocities (strength and orientation arrows), and 1, 2 and 3 sigma contours of the prior PDF, (b) to (f) 3D views of the updated target PDF at step $k = 1$, 5, 15, 30 and 60 respectively, corresponding to 30, 150, 450, 900, and 1800 sec, (g) Conditional (solid line) and 'discounted' (dash dotted line) probability of detecting the target on time step k denoted $p(\overline{\mathbf{D}}_k|\mathbf{z}_{1:k-1} = \overline{\mathbf{D}}_{1:k-1})$ and $p(D_k|\overline{\mathbf{D}}_{1:k-1})p(|vi\overline{D}1 : k-1)$ respectively, and (h) corresponding cumulative probability of detection P_k.

this problem, work in progress also involves developing a channel filter [6] to allow the Bayesian network to be tree connected and hence drastically reduce the communication loads that are incurred in a fully connected network, as well as allowing intermittent burst communications.

Beyond the demonstration of the approach on a team of UAV's, the ultimate objective of this research is to eventually have multiple platforms participating in actual search and rescue (SAR) missions with real-time cooperative planning and fully integrated human-robot interactions. The technique has

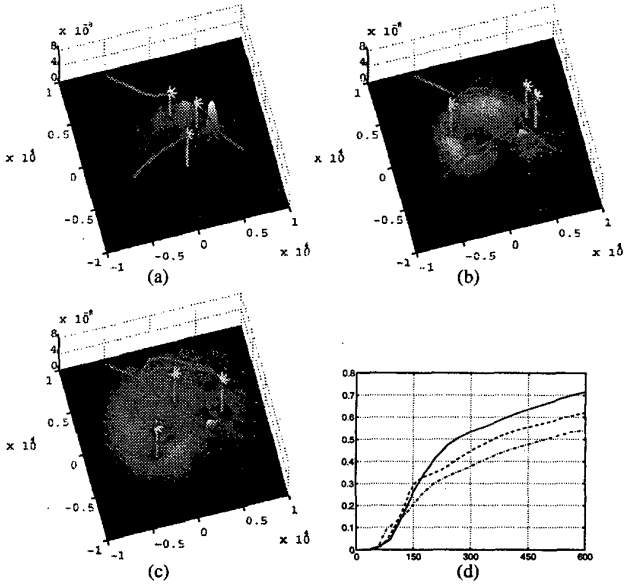

Fig. 10. Three vehicles greedy search for a target drifting in a storm with a hard constraint: (a) to (c) 3D views of the updated target PDF at step $k = 5$, 10, and 20 respectively, corresponding to 150, 300, and 600 sec, and (d) the corresponding P_k for the 3 vehicles search (solid line), the 2 vehicles 900s search (dashed line), and the 1 vehicle 1800s search (dash dotted line) compressed on 600s.

the potential to greatly improve upon current SAR protocols, which in turn might be critical in saving human lives.

ACKNOWLEDGMENTS

This work is partly supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government. The authors also wish to thank Ali Göktoğan from ACFR, the developer of the RMUS simulator, for his assistance with the simulation implementation.

REFERENCES

[1] J.O. Berger. *Statistical decision theory and Bayesian analysis*. Springer series in statistics. Springer-Verlag, New York, 2nd edition, 1985.

[2] F. Bourgault, T. Furukawa, and H.F. Durrant-Whyte. Coordinated decentralized search for a lost target in a bayesian world. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'03)*, October 2003.

[3] F. Bourgault, T. Furukawa, and H.F. Durrant-Whyte. Optimal search for a lost target in a bayesian world. In *Int. Conf. on Field and Service Robotics (FSR'03)*, July 2003.

[4] A.H. Goktogan, E. Nettleton, M. Ridley, and S. Sukkarieh. Real time multi-uav simulator. In *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[5] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F*, 140(2):107–113, April 1993.

[6] S. Grime and H.F. Durrant-Whyte. Communication in decentralized systems. *IFAC Control Engineering Practice*, 2(5):849–863, 1994.

[7] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte. Information-theoretic coordinated control of multiple sensor platforms. In *IEEE Int. Conf. on Robotics and Automation (ICRA'03)*, 2003.

[8] J.S. Przemieniecki. *Mathematical Methods in Defense Analyses*. AIAA Education Series. American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 2nd edition, 1994.

[9] L.D. Stone, C.A. Barlow, and T.L Corwin. *Bayesian Multiple Target Tracking*. Mathematics in Science and Engineering. Artech House, Boston, 1999.