PROCEEDINGS
SPIE—The International Society for Optical Engineering

Sensor Fusion VI

Paul S. Schenker
Chair/Editor

7–8 September 1993
Boston, Massachusetts

SPIE
P
PROCEEDINGS
SERIES

Volume 2059

PROCEEDINGS

SPIE—The International Society for Optical Engineering

Sensor Fusion VI

Paul S. Schenker
Chair/Editor

7–8 September 1993
Boston, Massachusetts

Sponsored and Published by
SPIE—The International Society for Optical Engineering

In cooperation with
Automated Imaging Association
Center for Excellence in Optical Data Processing/Carnegie Mellon University
The Advanced Technology Center for Precision Manufacturing/University of Connecticut
ISPRS—International Society for Photogrammetry and Remote Sensing

Volume 2059

P

The papers appearing in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and are published as presented and without change, in the interests of timely dissemination. Their inclusion in this publication does not necessarily constitute endorsement by the editors or by SPIE.

Please use the following format to cite material from this book:
Author(s), "Title of paper," in *Sensor Fusion VI*, Paul S. Schenker, Editor, Proc. SPIE 2059, page numbers (1993).

Library of Congress Catalog Card No. 93-85271
ISBN 0-8194-1324-0

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone 206/676-3290 (Pacific Time) • Fax 206/647-1445

Printed in the United States of America.

Conference Committee

Conference Chair

Paul S. Schenker, Jet Propulsion Laboratory

Program Committee

Terrance E. Boult, Columbia University
Ted J. Broida, Hughes Aircraft Company
Su-Shing Chen, University of North Carolina/Charlotte
David B. Cooper, Brown University
Gregory D. Hager, Yale University
Martin Herman, National Institute of Standards and Technology
Terrance L. Huntsberger, University of South Carolina
Ren C. Luo, North Carolina State University
Gerard T. McKee, University of Reading (UK)
N. Nandhakumar, University of Virginia
Bobby S. Y. Rao, University of California/Berkeley
Michael Seibert, Lincoln Laboratory/MIT
Faina Shtern, M.D., National Institutes of Health
Charles V. Stewart, Rensselaer Polytechnic Institute
Michael J. Swain, University of Chicago
Stelios C. A. Thomopoulos, The Pennsylvania State University

Session Chairs

1 Sensor Models and Validation
Bobby S. Y. Rao, University of California/Berkeley

2 Robust Methods for 3D Perception
Charles V. Stewart, Rensselaer Polytechnic Institute

3 Task-Directed Sensing
Gerard T. McKee, University of Reading (UK)

4 3D Sensing and Robotic Applications
Sukhan Lee, Jet Propulsion Laboratory

5 3D Shape Reconstruction
Terrance E. Boult, Columbia University

6 Feature Matching and Tracking
Suresh B. Marapane, University of Tennessee/Knoxville

7 Signal Analysis and Understanding
Terrance L. Huntsberger, University of South Carolina

Introduction

The Sensor Fusion conference series dates from 1988, as part of a larger set of yearly meetings held within the fall SPIE Advances in Intelligent Robotics symposium. Historically, the Sensor Fusion conference grew out of related meetings on machine vision and pattern recognition and their important engineering applications in complex automation and robotic systems. A number of workers were considering problems in which visual data was accumulated from multiple views, over multiple time observations, in different spectral bands, and from different sensor modalities such as combined visual and laser range data. There was also the issue of how to combine data not only from different observations and sensors, but also from fundamentally different information sources, one example being the integration of partial CAD models with on-line sensor data. Thus, it seemed appropriate to have a meeting that concentrated on the "fusion" issues: modeling different sensor sources, calibrating the models to one another, defining approaches and associated performance measures for data correlation across sensors, and developing efficient computational strategies for implementing the resulting "sensor fusion" strategies. These problems have good methodological content—they also have exciting engineering applications and scientific underpinnings that motivate their development. Applications of current importance include remote sensing, medical image analysis, industrial parts recognition, robotic manipulation and navigation, distributed detection and decision networks, and sensorimotor performance modeling in human-machine systems.

Our meetings began with engineers and computer scientists, and grew to encompass psychologists and neural scientists. Several computational paradigms have emerged, consistent with backgrounds and interests of conference participants. The earliest papers had a decidedly decision-theoretic flavor, with classical probability and statistics used as modeling tools. Subsequent developments expanded to include fuzzy and evidential (Dempster-Shafer) reasoning, and a wide variety of both data and model-driven fusion strategies were presented. The two dominant themes of the conference became 3D object perception and task-driven sensing. The latter area benefited significantly from participation of artificial intelligence researchers. In general, 3D perception has been a touchstone for most conference participants: 3D shape-and-motion estimation, and the associated recognition of associated objects and events, has deep technical roots and applications in most scientific communities.

Earlier years of the Sensor Fusion conference emphasized a particular "technical theme" area. For example, in the 1991 conference we chose a "control paradigms and data structures" theme, with the intent of comparing and contrasting the computational models used by engineers, computer scientists, psychologists, and others in formulating sensory data fusion problems. As a whole, this approach has been useful, helping to generate applications interest, focus topical sessions, and stimulate interdisciplinary interactions. Perhaps we will revisit the "theme" setting in future meetings, as emerging new interest areas justify. At present, the yearly conference appears to draw a focused, balanced, and strong set of contributions of its own accord. We will let you, the reader, judge.

(continued)

In conclusion, I thank the program committee members for their dedicated work in assembling the conference, and I express my regard to the authors for their thoughtful contributions and spirited presentations in both oral and poster formats.

Paul S. Schenker

Contents

SESSION 1

Sensor Models and Validation

Chair

Bobby S. Y. Rao
University of California/Berkeley

Sensor Models and a Framework for Sensor Management

Alex Gaskell*, Penelope Probert†

Robotics Research Group, Oxford University,
19 Parks Road, Oxford, OX1 3PJ. England
Email: [apg,pjp]@uk.ac.oxford.robots

Abstract

We describe the use of Bayesian belief networks and decision theoretic principles for sensor management in multi-sensor systems. This framework provides a way of representing sensory data and choosing actions under uncertainty. The work considers how to distribute functionality between sensors and the controller. Use is made of logical sensors based on complementary physical sensors to provide information at the task level of abstraction represented within the network. We are applying these methods in the area of low level planning in mobile robotics.

A key feature of the work is the development of quantified models to represent diverse sensors, in particular the sonar array and infra-red triangulation sensors we use on our AGV. We need to develop a model which can handle these very different sensors but provides a common interface to the sensor management process. We do this by quantifying the uncertainty through probabilistic models of the sensors, taking into account their physical characteristics and interaction with the expected environment. Modelling the sensor characteristics to an appropriate level of detail has the advantage of giving more accurate and robust mapping between the physical and logical sensor, as well as a better understanding of environmental dependency and its limitations. We will describe a model of a sonar array, which explicitly takes into account features such as beam-width and ranging errors, and its integration into the sensor management process.

1 Introduction

The aim of sensor management or planning is to make best use of the available sensors and in the limited time available so as to gain information about the environment and reduce uncertainty. The choice of where to look and with which sensor involves the tradeoff between accuracy and time, and is inherently task directed. In order to select the most appropriate sensor we need to quantify the characteristics of each sensor by modelling its operation and the way in which it senses features in the environment. Such a system is described in [1] that incorporates task direction within a decision theoretic model to produce a strategy or plan for gathering sensor information. We describe the development of a framework for carrying out the sensor management process that uses decision theory in a meta-level context for choosing which actions to take. We use probabilistic belief networks as a means for representing and updating uncertainty about the operating environment and the way in which features are sensed. Central to the framework is careful modelling of the sensors and consideration for the type and representation of information they should gather.

1.1 Meta-level reasoning perspective

The process of sensor management is developed from the viewpoint of meta-level reasoning, and in particular the control of deliberation under time pressure and limited resources [2]. We first describe how an agent performs meta-reasoning and its decision theoretic formulation and then show how it can be used for sensor management.

*Support for apg was provided by British Nuclear Fuels Ltd. under a SERC CASE award.
†Support for both authors was provided in part by ACME.

 0-8194-1324-0/93/$6.00

In any real system the amount of deliberation that can be carried out is bounded by the need to come to a decision and execute some action. Deliberative computation only has an effect on the internal state of the agent and has the aim of gaining knowledge so as to improve the quality of the external decisions that are made. For this reason the value of a computation is derived from its expected effects, consisting of :

- changes in the external environment due to the passage of time
- the possible revision of the agent's intended actions in the real world

The role of meta-reasoning is to direct the deliberation by selecting between the computations that are available so as to provide information that has most value in choosing what action to take. When an agent performs meta-reasoning it is reasoning about entities, internal to the system under consideration. Object level reasoning is concerned with objects and actions in the external world. The meta-level, on the other hand, uses a model of the operation of its object level in order to predict the outcome of a possible sequence of object level steps.

By breaking down the deliberative computation into a number of atomic steps and treating them as actions the decision theoretic mechanism of selecting the action with highest expected utility is used to direct action according to the current situation. At any given time the agent has a default external action or final decision α that is the one that currently has the highest expected utility. In addition, there is a set of deliberative computational actions $\{S_j\}$ that affect only the internal state of the agent and might cause the agent to revise its default action. Hence, the agent has to choose one from the set $\{\alpha, S_1, \cdots, S_n\}$, or in other words, either stop and act or deliberate further. A utility function is defined for the external actions in each world state. From this function, the expected value of each of the internal actions is given by the *expected net increase in utility* that would result if the computation were carried out and a different external action selected.

In the context of sensor management we equate the computational actions S_i to the possible sensor actions. The set of sensor actions is determined by the physical sensors and processing algorithms used, as well as the set of features that can be focussed upon. This leads to the use of abstract logical sensors as the sensor actions. It can be seen that the utility function is crucial in defining the behaviour of the system because it is used as the basis for choosing both the computational and external actions. It is split into 2 components so that we can consider the quality of an action separately from considerations about time pressure. The intrinsic utility is defined for each external action over the states of the world and is used to calculate the net increase in utility of a computational action. The time cost is defined for each computational action and is a function of the action's duration. The final *net value* of a computational action is given by subtracting the time cost from the net gain in intrinsic utility.

1.2 Use of probabilistic belief networks

Probabilistic belief networks [3] are a way of representing and reasoning with uncertainty about discrete variables and the dependencies that exist between them. A graphical notation is used, where the nodes are the variables and have a probability distribution, or *belief*, associated with them. Nodes are connected by directed links that signify causality and make explicit the dependencies between variables. The dependencies between variables are quantified by means of conditional probability distributions, whereby each node is conditional on the values of its parent nodes. A computational mechanism based upon the Bayesian framework of probability theory is used to propagate uncertainty throughout the network and update beliefs given evidence about any variable.

We use a probabilistic network to represent features in the environment about which we wish to gather information. The network makes explicit the relationships between these entities in the world, as well as the relationships between the entities and sensory observations. When an observation is made it is put into the network and the beliefs of each node updated. This network represents the state at one particular point in time. It is extended in the style of [4] to allow for reasoning over time by duplicating the nodes for each different instance of time and using probabilistic dependencies between the successive instances to model their change over time. This is shown in figure 1. Inference in the network can now be used to predict the future value of a variable given all prior evidence and a model of how the world entities change over time. The predicted beliefs in the values of world features are used to calculate the expected utilities of the sensory actions described in the previous section.

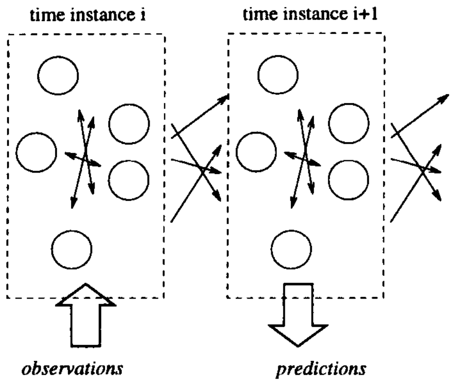

Figure 1: The temporal belief network

2 The application - Sensing for obstacle avoidance

We apply the framework to the task of obstacle avoidance for mobile robots in structured environments. When an unexpected obstacle is encountered a decision has to be made about what avoidance manoeuvre to take before collision occurs, the choices being to either sidestep to the left or right of the obstacle, or backoff and replan an alternative route to the goal position. In order to determine whether the pathways around the obstacle give sufficient clearance a number of sensors are employed.

The vehicle used is the Oxford AGV, a small prototype of a small factory vehicle originally developed by GEC FAST division. It is operated in our basement laboratory that has several large pillars that divide the room into connected aisles. The sensors we use on the AGV consist of:

- sonar ring around the vehicle
- line-of-sight rotating infra red scanner
- triangulating infra red range scanner

Detailed models of the operation and physical characteristics have been developed for the infra-red sensors, and processing algorithms for edge detection, and free pathway detection implemented [5]. These sensors provide a complementary set of characteristics. The sonar operates well at long distances with a wide field of view, but suffers from lack of angular accuracy. On the other hand, the infra-red sensors provide accurate measurements at short and medium ranges. The field of view is restricted by the amount of time available for processing the scans or the time taken to physically rotate the sensor. In addition, the infra-red radiation is absorbed by dark or black surfaces and so dark objects are not always detected. These characteristics are used within the sensor management process to determine which to use. For example, if the amplitude of a returned infra-red signal is low then a dark obstacle could be expected and the sonar used instead, even though accuracy may not be as high.

The utility function used to defines the preference order over each external action in each state of the world. The actions available are SIDESTEP LEFT, SIDESTEP RIGHT and BACKOFF. The world state here consists of the sizes of the left and right pathways around the obstacle, each of which can take on the discrete values TOO-SMALL, JUST-PASSABLE or OK. The function is shown in figure 2.

In effect this says that the worst case - i.e attempting to go through a blocked path - has minimum utility, whereas going through a clear pathway has maximum utility of 1. The intermediate case where the path is just passable has a utility, c, that lies between these, the exact value of which is determined by the amount of risk that can be tolerated. The cost of backoff, k, is constant for all cases and can be determined dynamically by considering any alternative path information that may be available from a higher level path planner.

left path:	SMALL	SMALL	SMALL	JUST	JUST	JUST	OK	OK	OK
right path:	SMALL	JUST	OK	SMALL	JUST	OK	SMALL	JUST	OK
SIDESTEP LEFT	0	0	0	c	c	c	1	1	1
SIDESTEP RIGHT	0	c	1	0	c	1	0	c	1
BACKOFF	k	k	k	k	k	k	k	k	k

Figure 2: Utility function

3 Logical sensors within this framework

Logical sensors [6] are used in order to bridge the gap between the low level data provided by the physical sensors and the higher level features upon which the utility function is defined and decisions are made. The essential feature of the logical sensor is the encapsulation of the physical sensor and processing into one unit. In general a logical sensor reading may be derived from multiple physical readings and there may be a choice of several different algorithms to perform the processing. The model of the logical sensor corresponds to the sensor actions that are selected within the sensor management decision process and fits well into the belief network representation. Each sensor action is equivalent to a logical sensor with a *single* algorithm. If different algorithms are used on top of the same physical sensor to measure the same feature then these are treated as different actions. This allows for the different timing and error characteristics to be distinguished between when evaluating the sensor actions.

Within the belief network that represents the features of interest for the utility function, the sensor readings are instantiated as evidence and then propagated to give the updated beliefs. The evidence can take one of two forms, either specific or virtual, the difference being in how the uncertainty distribution for the reading is represented. Which of the forms is used depends on the operation of the sensor and is described in more detail later in the section. In both cases, however, an ***observation node*** is used, as shown in figure 3. The conditional probability distribution $P(obs|feature)$ gives the measure of uncertainty in a reading. Such nodes were used in [4] for a robot localisation and tracking problem. In the case of sensors that can be directed to focus on a specific instance of a feature type, or a sensor that provides evidence for more than one feature at a time, there will be one observation node for each of the features that it can view. For example, a directable infra-red scanner can be moved to measure any one of several clear pathways, each of which will have an observation node. This means that within the network each sensor can have multiple observation nodes associated with it.

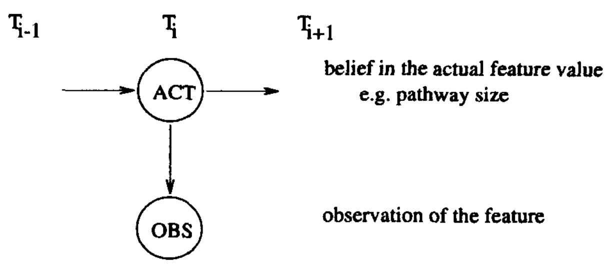

Figure 3: The actual value and observation node pair

The benefit of using logical sensor actions is that they provide a common interface to the belief network as defined by the evidence that is instantiated into the observation nodes of the network. The range of values that the evidence can take is given by the feature observed and not by the mechanism employed.

Care needs to be taken when designing belief networks for real applications to ensure that the computational complexity of propagating evidence does not get out of hand. This can be helped by considering the distribution of functionality between logical sensor and network. This can be seen as choosing the level of abstraction of the features detected by the sensor, from the simplest point range reading to a high level clear pathway detector, given that the top level features required for the utility function are already fixed and represented as nodes in the network. In the case of obstacle avoidance these top level features are the left and right pathways around the obstruction. There are 2 possibilities for the type of logical sensor used. For the first case clear pathway sensors are used to directly give evidence for the left/right pathway observation nodes. In the second case evidence is put in at the level of the obstacle edges in the form of range and angle. From these edges it is possible to specify a conditional distribution for the clear pathway that computes the size by propagating uncertainties.

Complexity of the network is a function of the size of the node state spaces as well as the number of nodes. Therefore the design of the sensor should consider what discrete values a reading should take, aiming to make the space of observations as small as possible without abstracting out useful information that is relevant to the decision task. [7] describes this discretisation aspect of sensor abstraction for sonar sensors used for mobile robot localisation. Distance readings are broken down into three discrete values, short, medium and long and then used to determine what type of junction it is in so that it can find its position with respect to a world map.

Experience in developing more explicit models for obstacle avoidance has shown that in order to keep down the network inference time as much of the processing should be put within the logical sensor. Evidence put into the network should ideally be at the same abstract level as used in the utility function. Transformation from one feature type to another is more suited to the standard imperative style algorithms. Furthermore, because the size of the state spaces of the nodes needs to be kept small, the readings should be chosen so as to take on the minimum number of discrete values. This has led to the choice of logical sensor actions as clear-pathway-detectors that return the values (SMALL, JUST-PASSABLE, OK), and emphasises the importance in the use of ***discrete abstract sensor actions*** within this framework.

Having defined the sensor in terms of its level of feature abstraction and the state space of its readings, the uncertainty associated with the physical sensor and algorithm used needs to be modelled and quantified in the form of probability distributions. If no modelling is carried out then the probability distributions have to be derived purely from experimental data and is likely to be too environmentally specific and not sufficiently robust to any change within the application environment. We now describe in more detail the two forms in which evidence from the sensors is used to update the beliefs within the network and the types of sensor that are suited to each.

3.1 Specific evidence sensors

This is the usual method by which evidence concerning the value of a node in the network is instantiated, in our case setting the observation node evidence to the logical sensor's reading. The basic dependencies that affect the accuracy of the reading and state of the sensor are represented by additional nodes and arcs within the network.

We have made the addition of several new nodes to the basic network model shown in figure 3. The most important of these in terms of the uncertainty of a reading is the ACCURACY node, whilst the STATUS and IN-VIEW nodes as a way of including information about the operation and control of the sensor. Around these nodes will fit parameters that are specific to the specific sensor, such as its orientation or the scan density.

- ACCURACY node modifies the uncertainty distribution for an observation, in effect providing a way of setting the variance of the observation. Its value may be given explicitly with a sensor reading (e.g. the amplitude of the returned infra-red beam), or may be dependent on the values of other variables such as the distance to the obstacle.

- STATUS node has states that represent the idea of undefined and unavailable operation of the sensor. The undefined state is used with directable sensors for when the feature to be observed is not within the field of view. The sensor may be unavailable due to fault or constraints on its use, such as not being able to fire all sonar together or being a shared resource.

- IN-VIEW node is used for directable sensors where a reading is meaningless unless it is focussing its attention on the feature. This node is connected to the status node in order to affect the belief in its undefined state.

Information about the status of a sensor and constraints on its use can be added into the network by further extending it with new nodes and dependencies. [8] uses an invalidating node in order to explicitly represent a defective sensor and model the degradation of performance over time. An alternative way of extending the model is by using the idea of separate sub-systems that may or may not use probabilistic network reasoning, but that interface by supplying evidence for the appropriate nodes. [9] have developed a system for choosing what information to gather in a visual question-answering system that uses a composite Bayes net. The domain knowledge for this task is of different sorts and so instead of a single network, a separate networks are used to reflect this structure.

3.2 Virtual evidence sensors

It is sometimes not possible to quantify the sources of uncertainty for a sensor in advance because there is too much dependence on environmental conditions or variables. Prior distributions for these variables could be determined experimentally for various classes of environment, for example cluttered office rooms and warehouse aisles, but this method is not always appropriate and the inaccuracies introduced would be too great. This does not mean that probability distributions cannot be calculated given run-time information about these environmental quantities, but that the prior knowledge is insufficient and not amenable to representation within the network because it would add too much complexity. In this case sensor readings are instantiated into the network in the form of *virtual evidence* [3]. Instead of giving the reading as a specific value, likelihood information for each possible value is given in the form of a likelihood ratio. This can be seen as an external judgement on the node's probability distribution. This form of evidence is used for sonar readings that provide an overall view of the environment, providing initial detection of obstacles and a coarse estimate of pathways.

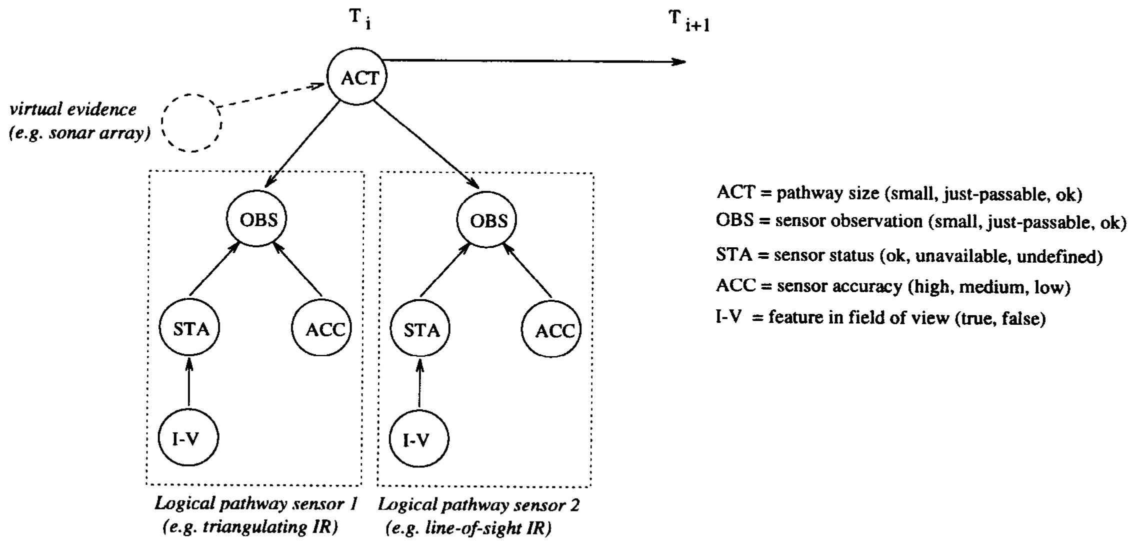

Figure 4: Final network for pathway

The final network structure for a single pathway that has 2 logical clear pathway sensors that provide specific evidence, and one logical sensor giving virtual evidence is shown in figure 4. Such a structure is used to represent the logical sensors based upon the line of sight and triangulation laser range sensors. Models for these sensors are described in [5, 10]. These identify and quantify the sources of uncertainty in a reading, and in particular the dependence of accuracy on the amplitude of the returned signal. The dominating factor in the returned amplitude is surface reflectance and not just the range of the target. A measure of the amplitude is given for each reading and so the evidence instantiated into the network consists of both the observation and the accuracy measure.

4 Example of an abstract sonar model

In this section we develop a model for a logical sensor for detecting clear pathways that is based on a simple sonar array. Sonar has the advantage of being able to give an overall view of the environment at ranges up about 4 or 5 metres, however the wide beam width means that measuring the position or size of a feature, in this case a clear pathway, will be inaccurate. The aim is to extract as much information as possible about the size of the pathways around an obstacle by considering the way in which the array senses the features and the sources of uncertainty. This information is then put into the belief network in the form of virtual evidence for each of the pathway size nodes. It is intended that this coarse information is used as an initial estimate which can then be used by the sensor management process to direct the more accurate infra-red sensors.