Sensor Fusion for Mobile Robot Navigation

MOSHE KAM, XIAOXUN ZHU, AND PAUL KALATA

We review techniques for sensor fusion in robot navigation, emphasizing algorithms for self-location. These find use when the sensor suite of a mobile robot comprises several different sensors, some complementary *and some* redundant*. Integrating the sensor readings, the robot seeks to accomplish tasks such as constructing a map of its environment, locating itself in that map, and recognizing objects that should be avoided or sought. Our review describes integration techniques in two categories:* low-level fusion *is used for direct integration of sensory data, resulting in parameter and state estimates;* high-level fusion *is used for indirect integration of sensory data in hierarchical architectures, through command arbitration and integration of control signals suggested by different modules.*

The review provides an arsenal of tools for addressing this (rather ill-posed) problem in machine intelligence, including Kalman filtering, rule-based techniques, behavior based algorithms, and approaches that borrow from information theory, Dempster–Shafer reasoning, fuzzy logic and neural networks. It points to several further-research needs, including: robustness of decision rules; simultaneous consideration of self-location, motion planning, motion control and vehicle dynamics; the effect of sensor placement and attention focusing on sensor fusion; and adaptation of techniques from biological sensor fusion.

I. INTRODUCTION

Robot navigation requires the guiding of a mobile robot to a desired destination or along a desired path in an environment characterized by a terrain and a set of distinct objects (such as obstacles, milestones, and landmarks). Motion planning is often designed to optimize specific performance criteria and to satisfy constraints on the robot's motion. Typical performance criteria are minimum time to arrive at a milestone and minimum control effort. Typical constraints are obstacle avoidance and a maximum robot velocity. The complexity and variety of environments where robots need to navigate, and the large number of objectives and constraints that they must satisfy, make the mobile-robot navigation problem ill posed. Moreover, physical platforms and sensor suites vary significantly from system to system, complicating further the task of generating a unified framework for sensing and control.

Manuscript received February 28, 1996; revised October 14, 1996. This work was supported by National Science Foundation Grants ECS-9057587 and ECS-9216588 and by the Electrical Power Research Institute Contract RP 8030-01.

The authors are with the Data Fusion Laboratory, Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104 USA (e-mail: kam@lorelei.ece.drexel.edu; xzhu@lorelei.ece.drexel.edu; paul_kalata@ece.drexel.edu).

Publisher Item Identifier S 0018-9219(97)00774-3.

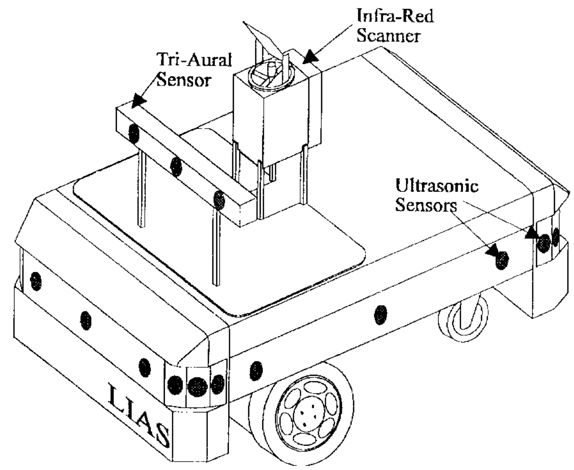

Fig. 1. The mobile robot LIAS [53], [54].

Mobile robot designers can choose from a large number of sensor types and sensing modules. These are sometimes *complementary*, sometimes *redundant*, and there exist architectures where sensors are used in both fashions. Many mobile robots carry sensors for dead reckoning (such as optical encoders and geomagnetic sensors) and for map making and self-location (such as time-of-flight ultrasonic systems and laser-based ranging systems). Some use active beacons (such as the global positioning system), or landmarks whose positions in the robot's environments are known. Other mobile robots use tactile sensors to touch obstacles in the environment and plan paths around them. In almost all designs, several sensors are operated simultaneously, and in most—sensors of different sensing principles, capabilities, and volumes-of-coverage are used in parallel. Consequently, methods of *sensor fusion* are needed to translate the different sensory inputs into reliable estimates and environment models that can be used by other navigation subsystems. Sensor fusion in this context is the process of integrating data from distinctly different sensors for detecting objects, and for estimating parameters and states needed for robot self-location, map making, path computing, motion planning, and motion execution.

A. Example One—A Multisensor Suite for a Mobile Robot

Fig. 1 shows a schematic diagram of Leuven Intelligent Autonomous System (LIAS), a mobile robot that uses several sensory modules [53], [54].

LIAS is a four-wheeled vehicle with two independently driven front wheels and two free casters at the rear. Two encoders (attached to the front wheels) and two gyroscopes provide information for position estimation. The robot is equipped with three different perception sensor systems. Fourteen standard Polaroid ultrasonic transducers are positioned around the vehicle circumference. A special "tri-aural" sensor is mounted in the front of the robot to get a better view in the drive direction, and an optical infrared range scanner, providing a complete panoramic image, is mounted on a platform overlooking the robot. An on-board transputer system executes different modular navigation tasks (which can be performed in parallel) and a host computer compiles and downloads the transputer programs.

Like many robots designed for indoor use, LIAS employs ultrasonic sensors for ranging by time-of-flight measurements. The usefulness of these sensors is limited by several factors [22]: 1) variations in the speed of propagation, 2) uncertainties in determining the time of arrival of the reflected pulse, 3) inaccuracies in the timing circuitry, 4) interaction of the incident wave with the target surface (*e.g.*, multiple reflections and "blindness" to some geometric features), and 5) wide beam-width which yields uncertain measurements within the main lobe. To compensate for these shortcomings, LIAS uses an infrared sensor that augments the ultrasonic measurements. Infrared sensors provide less-accurate range measurements when compared to the ultrasonic sensors (in LIAS the ultrasonic sensor error is specified as "less than 1 cm" while the infrared-range error is "less than 20 cm"). However, infrared sensors can provide a large number of measurements in a short time period, can easily be mounted on a scanner to provide a panoramic view, and their very narrow beam shape compares favorably with the inherent cone-shaped beam of the ultrasonic sensors. Laser and infrared range finders are therefore used to determine reliably the absence or presence of an object in the vicinity of the robot, and to identify edges (such as doorways) that are invisible to ultrasonic subsystems. The combination of readings from the two types of sensors can potentially provide a comprehensive and accurate view of the robot's environment, far superior to that achievable by each type operating alone. There is an additional tri-aural ultrasonic sensor system, consisting of three ultrasonic transducers which are placed in-line, at an interval distance of 15 cm (see Fig. 1). The center transducer serves as transmitter and receiver while the two others are receivers only. By triangulation, the tri-aural sensor is able to detect more than one object with one measurement. It provides not only the range, but also the orientation of a detected object with respect to the robot's heading direction, and some of the object features (plane, edge, corner).

B. Low Level and High Level Fusion

Sensor fusion in mobile robots is usually categorized to be *low level* or *high level*. The term *low-level fusion* is often used for direct integration of sensory data, resulting in parameter and state estimates. These estimates in turn are used by planning and motion-execution modules to generate command and control signals for the robot's motors. The term *high-level fusion* is used for indirect integration of sensory data in hierarchical architectures, through command arbitration and integration of control signals that are suggested by different modules. In the "gray" area between the two classes are architectures that synthesize command and control signals directly from sensory input—often without explicit construction of environmental models.

There is a continuous debate in the decision and estimation research community about the relevance and applicability of traditional and nontraditional approaches to detection and estimation, e.g., [9]. At issue are the availability and accuracy of plant and statistical-data models, and the robustness of decisions and estimates in the face of noise, parameter drifts, failures, unmodeled dynamics, and unmodeled statistics. Understandably, this general debate has been projected onto the area of sensor fusion, since sensory-data interpretation from several different subsystems further complicates the selection of methods, uncertainty representations, and assessments of performance. Indeed, the field of sensor fusion, and sensor fusion for robot navigation, is populated by approaches that differ in philosophy, information structure, and assumptions about the availability of reliable models.

There have been several reviews of sensor fusion, and several special issues of journals devoted to the topic. Most relevant to the area of robot navigation are the review paper by Luo and Kay [42], the report *Where am I* by Feng *et. al*. [22], the book *Integration, Coordination and Control of Multi-Sensor Robot Systems* by Durrant-Whyte [19], the December 1988 issue of the *International Journal of Robotic Research*, and the June 1996 issue of *IEEE Transactions on Industrial Electronics*. Pertinent background material is also available in the books of Bar-Shalom and Fortmann [6], Dasarathy [18], Waltz and Llinas [55], and Cox and Wilfong [16]. Good sources for recent research are proceedings of the *IEEE International Conference on Robotics and Automation, IEEE International Conference on Multisensor Fusion*, the *International Conference on Intelligent Robots and Systems*, and the SPIE conferences on Mobile Robots.

C. Organization and Approach

In the present review, we start with low-level fusion (Section II) and proceed to high-level fusion (Section III). Throughout the paper, we demonstrate the reviewed techniques with examples of physical implementations. Within the low-level fusion part (Section II), we discuss first (in Section II-A) *centralized* architectures that possess known statistics of the environment and the sensory data. These make use of the Kalman Filter and the Extended Kalman Filter (EKF). We then discuss (Section II-B) *decentralized* architectures that use statistics, and concentrate there on *fusion of estimates*. In Section II-C we discuss low level fusion when statistics are unknown, emphasizing rule-based sensor fusion and use of geometric and topological maps.

In the high-level fusion part (Section III), we describe behavior-based architectures (Section III-A) including the subsumption architecture and various voting schemes. We then present (Section III-B) unified frameworks for sensory processing that employ neural networks.

Like all reviews of this kind, ours represents the state of the art as it is reflected in the present literature, as well as the authors' views and perhaps prejudice. We emphasize techniques that are more general, mathematically explicit, and established, over techniques that are more heuristic in nature and depend heavily on details of the platform and the environment. We provide more detailed explanations of sensor fusion with known statistics, and the majority of the algorithms that we review fall under *low-level fusion*. These preferences stem from the observations that 1) sensor fusion with known statistics is a better posed problem, analytically tractable and easier to describe than other approaches, 2) many algorithms that must operate without the benefit of statistical models are based on known-statistics methods in philosophy and architecture, and 3) a much larger body of work exists on low-level fusion compared to high-level fusion.

II. LOW-LEVEL FUSION

A. Low-Level Fusion with Known Statistics in Centralized Architectures

Sensor fusion with known statistics often relies on well-developed techniques such as maximum *a posteriori* and maximum-likelihood estimation, and adapts results from Kalman filtering, Bayesian team theory, and game theory. The sensors in the mobile robot's sensor suite are usually viewed as members of a team, namely an organization whose members possess a common single goal and a common payoff function. The goal is to determine a parameter $\theta \in \mathcal{R}^m$ from a set of observations $\boldsymbol{z} \in \mathcal{R}^n$, knowing that θ and $\boldsymbol{z}$ are related through a relation of the form $g(\boldsymbol{z}, \theta) = 0$. Here $g: \mathcal{R}^n \times \mathcal{R}^m \rightarrow \mathcal{R}^p$ is a known nonlinear function [4], [21]. The sought parameters θ can be features in the robot's environment or the state of the robot in the environment. For example, g can be the model of a physical geometric object that maps a compact region $\boldsymbol{Z} \subseteq \mathcal{R}^n$ in Euclidean n-space to a point $\theta \in \mathcal{R}^m$ in the parameter space [20]. Each function g describes a type of geometric object (e.g., a line, a surface or a polygon) and each value of θ represents a specific realization of such object. The sensor observes the value of θ in order to build a map of its environment. For plane surfaces, for example, the relation is

$$g(\boldsymbol{z}, \theta) = \theta_x x + \theta_y y + \theta_z z + 1 = 0 \quad (1)$$

with $\boldsymbol{z} = [x \quad y \quad z]^T$ and $\theta = [\theta_x \quad \theta_y \quad \theta_z]^T$. A specific plane is represented as a specific value of θ in $\mathcal{R}^3$.

1) Example Two—Feature Extraction from Stereo: In [4], Ayache and Faugeras study the use of passive stereo to collect three-dimensional (3-D) information by a mobile robot. They assume that each observation collected by the robot, $\boldsymbol{r} \in \mathcal{R}^n$, is corrupted with additive zero mean Gaussian noise $\boldsymbol{\epsilon}$, *i.e.*,

$$\boldsymbol{r} = \boldsymbol{r}_0 + \boldsymbol{\epsilon}, \quad E(\boldsymbol{\epsilon}) = \boldsymbol{0}, \quad E(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T) = \Lambda^{n \times n}. \quad (2)$$

The problem is to use a number of observations $\{\boldsymbol{r}_i\}_{i=1}^N$ to estimate the parameter vector θ such that the estimate $\hat{\theta}$ is "best" in satisfying the relation $g_i(\boldsymbol{r}_i, \theta) = 0$ in some sense. θ can be for example the 3-D coordinate of a point observed by the stereo system. Omitting the subscripts for a moment, a Taylor-series approximation of $g(\boldsymbol{r}, \theta)$ about $\boldsymbol{r}_0$ and θ^* can be calculated, where θ^* is a "good" initial estimate of θ, to obtain

$$\begin{aligned} 0 &= g(\boldsymbol{r}_0, \theta) \\ &\approx g(\boldsymbol{r}, \theta^*) + \left[\frac{\partial g}{\partial \boldsymbol{r}}(\boldsymbol{r}, \theta^*)\right](\boldsymbol{r}_0 - \boldsymbol{r}) \\ &\quad + \left[\frac{\partial g}{\partial \theta}(\boldsymbol{r}, \theta^*)\right](\theta - \theta^*). \end{aligned} \quad (3)$$

In (3), $[\partial g / \partial \boldsymbol{r}]$ is a $p \times n$ matrix and $[\partial g / \partial \theta]$ is a $p \times m$ matrix. This equation can be rewritten [4, p. 46] to approximate a linear measurement equation

$$\boldsymbol{z} = \boldsymbol{H}\theta + \boldsymbol{v} \quad (4)$$

with

$$\begin{aligned} \boldsymbol{z} &= -g(\boldsymbol{r}, \theta^*) + \frac{\partial g}{\partial \theta}(\boldsymbol{r}, \theta^*)\theta^*, \\ \boldsymbol{H} &= \frac{\partial g}{\partial \theta}(\boldsymbol{r}, \theta^*) \quad \text{and} \quad \boldsymbol{v} = -\frac{\partial g}{\partial \boldsymbol{r}}(\boldsymbol{r}, \theta^*)\boldsymbol{\epsilon}. \end{aligned}$$

Both $\boldsymbol{z}$ and $\boldsymbol{H}$ are known, since $g, \boldsymbol{r}$ and θ^* are known. Furthermore, the second-order statistics of $\boldsymbol{v}$ are known, namely

$$\begin{aligned} E(\boldsymbol{v}) &= 0 \\ \boldsymbol{R} &= E(\boldsymbol{v}\boldsymbol{v}^T) = \left[\frac{\partial g}{\partial \boldsymbol{r}}(\boldsymbol{r}, \theta^*)\right]\Lambda\left[\frac{\partial g}{\partial \boldsymbol{r}}(\boldsymbol{r}, \theta^*)\right]^T. \end{aligned} \quad (5)$$

For N observations $\{\boldsymbol{r}_i\}_{i=1}^N$, each with known g_i and covariance Λ_i, the ith measurement is

$$\boldsymbol{z}_i = \boldsymbol{H}_i\theta + \boldsymbol{v}_i \quad (6)$$

and the corresponding $\boldsymbol{R}_i = E(\boldsymbol{v}_i\boldsymbol{v}_i^T)$ can be calculated. In the terminology of the Kalman filter, (6) is a measurement equation on the process θ, which is constant with respect to $i: \theta_i = \theta_{i-1}$. It is now possible to use Kalman filtering to estimate θ. The process starts with an initial estimate $\hat{\theta}_0$ of θ and its associated covariance matrix $\boldsymbol{S}_0 = E[(\hat{\theta}_0 - \theta)(\hat{\theta}_0 - \theta)^T]$. The new estimate and the estimation covariance matrix are, in this case

$$\hat{\theta}_i = \hat{\theta}_{i-1} + \boldsymbol{K}_i(\boldsymbol{z}_i - \boldsymbol{H}_i\hat{\theta}_{i-1}) \quad (7)$$

$$\boldsymbol{K}_i = \boldsymbol{S}_{i-1}\boldsymbol{H}_i^T(\boldsymbol{R}_i + \boldsymbol{H}_i\boldsymbol{S}_{i-1}\boldsymbol{H}_i^T)^{-1} \quad (8)$$

$$\boldsymbol{S}_i = (\boldsymbol{I} - \boldsymbol{K}_i\boldsymbol{H}_i)\boldsymbol{S}_{i-1}. \quad (9)$$

When all the measurements have been processed, the parameter θ is known by its *a posteriori* estimate $\hat{\theta}_N$ and the corresponding covariance matrix $\boldsymbol{S}_N = E[(\hat{\theta}_N - \theta)(\hat{\theta}_N - \theta)^T]$. The estimate $\hat{\theta}_N$ is "best" in the sense of minimizing

$$(\theta - \hat{\theta}_0)^T \boldsymbol{S_0}^{-1}(\theta - \hat{\theta}_0) + \sum_{i=1}^{N}(\boldsymbol{z}_i - \boldsymbol{H}_i\theta)^T \boldsymbol{R}_i^{-1}(\boldsymbol{z}_i - \boldsymbol{H}_i\theta). \tag{10}$$

This performance criterion demonstrates how the filter explicitly weights noise in the measurements. If the ith measurement is noisy, its corresponding weight $\boldsymbol{R}_i^{-1}$ is "small."

2) Kalman Filtering for Data Fusion: Example Two is an application of the Kalman filter to the integration of sensory information in robot navigation. The Kalman Filter and the EKF [31] and [7, ch. 5] are the most popular tools proposed in the literature for sensor fusion in mobile robot navigation. In the *Kalman filter* formulation, the observations $\boldsymbol{z}(k) \in \mathcal{R}^n$ are described (or approximated) by the linear model

$$\boldsymbol{z}(k+1) = \boldsymbol{H}(k+1)\boldsymbol{x}(k+1) + \boldsymbol{v}(k+1) \tag{11}$$

where $\boldsymbol{x} \in \mathcal{R}^m$ is a state vector, $\boldsymbol{H} \in \mathcal{R}^{n\times m}$ is an observation model, and $\boldsymbol{v} \in \mathcal{R}^n$ is the observation noise. The state vector satisfies a linear discrete-time state transition equation

$$\boldsymbol{x}(k+1) = \boldsymbol{F}(k)\boldsymbol{x}(k) + \boldsymbol{G}(k)\boldsymbol{u}(k) + \boldsymbol{w}(k) \tag{12}$$

where $\boldsymbol{F} \in \mathcal{R}^{m\times m}$ is the system model, $\boldsymbol{G} \in \mathcal{R}^{m\times q}$ is the control model, $\boldsymbol{u} \in \mathcal{R}^q$ is a known input (control or sensor motion), and $\boldsymbol{w} \in \mathcal{R}^m$ is the input noise.

As usual we assume independent, zero mean, white noise processes

$$\begin{aligned} E[\boldsymbol{w}(k)] &= E[\boldsymbol{v}(k)] = 0, \quad E[\boldsymbol{w}(k)\boldsymbol{w}^T(j)] = \boldsymbol{Q}(k)\delta_{k,j} \\ E[\boldsymbol{v}(k)\boldsymbol{v}^T(j)] &= \boldsymbol{R}(k)\delta_{k,j}, \quad E[\boldsymbol{w}(k)\boldsymbol{v}^T(j)] = 0 \end{aligned} \tag{13}$$

where $\delta_{k,j}$ is the Kronecker delta function $(\delta_{k,j} = 0, k \neq j; \delta_{k,j} = 1, k = j)$.

The optimal mean square error estimate of $\boldsymbol{x}(k)$ given $\boldsymbol{z}(1), \cdots, \boldsymbol{z}(j)$ $(k \geq j)$ is

$$\hat{\boldsymbol{x}}(k|j) = E[\boldsymbol{x}(k)|\boldsymbol{z}(1), \cdots, \boldsymbol{z}(j)] \tag{14}$$

and the conditional covariance matrix of $\hat{\boldsymbol{x}}(k|j)$ is

$$\begin{aligned} \boldsymbol{P}(k|j) = E\{[\boldsymbol{x}(k) - \hat{\boldsymbol{x}}(k|j)][\boldsymbol{x}(k) \\ - \hat{\boldsymbol{x}}(k|j)]^T|\boldsymbol{z}(1), \cdots, \boldsymbol{z}(j)\}. \end{aligned} \tag{15}$$

The Kalman filter algorithm provides recursively an estimate $\hat{\boldsymbol{x}}(k+1|k+1)$ in terms of the previous estimate $\hat{\boldsymbol{x}}(k|k)$ and the most recent observation, $\boldsymbol{z}(k+1)$. It involves a cycle of *prediction*, *observation*, and *updating* (there is often an additional *data validation* step prior to updating). For the prediction, validation, and updating equations, see [31] and [7].

The measurement model for the EKF is

$$\boldsymbol{z}(k+1) = h[k+1, \boldsymbol{x}(k+1)] + \boldsymbol{v}(k+1) \tag{16}$$

and the dynamics are assumed to be

$$\boldsymbol{x}(k+1) = f[k, \boldsymbol{x}(k), \boldsymbol{u}(k)] + \boldsymbol{w}(k). \tag{17}$$

The vector-valued function h and f are, in general, time varying. The EKF framework is developed through a series expansion of the nonlinear dynamics and of the measurement equation. For the prediction, validation, and updating equations see [7, sec. 10.3].

The literature has many examples for the use of Kalman filters in robot navigation [4], [5], [14], [17], [27], [34], [37], [40], [44]. Kriegman *et al*. [37] studied stereo vision and navigation in buildings; Crowley [17] used Kalman filtering to aggregate readings from ultrasonic sensors; and Cox [14] fused odometry and range finder readings for an autonomous robot in an office environment. Recently, Hong and Wang [27] have used a Kalman filtering approach to integrate sensory data which are both noisy (due to stochastic uncertainty) and fuzzy (due to inaccuracy of the data measuring process). In the process they have developed a compression technique that arrests the growth in "fuzziness" under extended fuzzy-arithmetic operations.

The most common use of Kalman filters in sensor fusion for navigation is to construct and maintain a model of the mobile-robot environment, and to monitor the position of the moving robot in that environment. The filters estimate simultaneously the parameters of the features or landmarks needed in the environment model, and the state of the moving robot. Two important applications are in *building visual maps* and in navigating *automated guided vehicles*.

a) Building visual maps: Ayache and Faugeras [4] have used the Kalman filter formulation [(7)–(9)] to build a 3-D description of the environment of a mobile robot, using passive vision. They have studied several important problems using this approach. The first is *stereo reconstruction*, starting with the determination of a 3-D position vector of a point-object, from its (matched) projections on the image planes of a stereo camera. The description of the position estimate includes an assessment of the estimate uncertainty. This procedure was then extended to lines and planes, and applied to recognition and localization of 3-D objects. The result is a 3-D description of the robot's environment in terms of its geometry, along with the uncertainty of the characterizing primitives (points, lines, planes). The description is attached to a local coordinate frame. The second problem studied in this framework was *registrating stereo pairs*—allowing the estimation of robot displacement from two 3-D descriptions of the robot environment (obtained before and after the displacement has occurred). By matching primitives that are present in both descriptions, an estimate of the displacement can be calculated. This estimate can be used further to improve the description of the environment, to build estimates of the 3-D transformations between frames, and to provide a measure of the uncertainty of these transformations.

Finally the information was used to update, in each local frame, the description of the geometry and uncertainty of the primitives corresponding to parts of physical objects visible in another frame. The algorithms were implemented and tested on images of indoor environments captured by stereo cameras [4, pp. 56–61], [5, pp. 814–819].

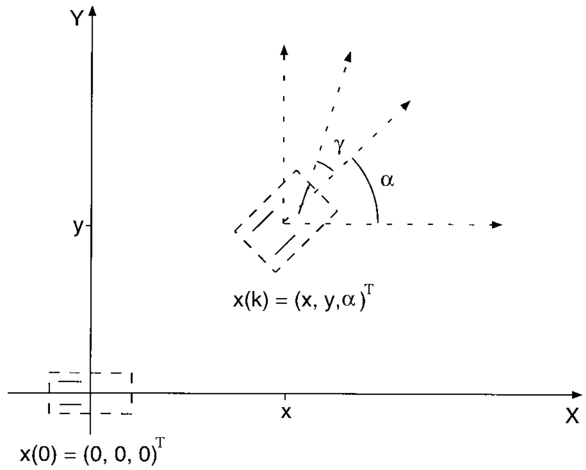

Fig. 2. AGV system geometry [11].

b) Position estimation for an autonomous guided vehicle: In [11], Borthwick and Durrant–Whyte study the dynamic self-location of an autonomous guided vehicle (AGV). AGVs are one of the most important applications of mobile robots, with a variety of anticipated uses in industrial environments. Since the floor plan and the location of obstacles in an industrial plant are often constant for considerable periods of time, AGVs can take advantage of known *a priori* maps of their environments. Moreover, industrial-plant environments usually offer naturally occurring features that can replace the artificial beacons and special landmarks often placed as features in the robot's environment. In the process of self-location, the map is used to identify these features within the environment and regard them as beacons for navigation purposes. The advantage of using existing features often comes at the expense of increased computation time during the *stop, look, localize* navigation cycles.

The AGV studied in [11] fuses information from an *a priori* map (where known features are held) and from information supplied by an infrared range scanner. The map is segmented into cells, and a list of features (points, lines, edges, corners) is associated with each cell. The system geometry is depicted in Fig. 2.

The AGV operates within a two-dimensional (2-D) Cartesian environment. Its state $\boldsymbol{x} = [x\, y\, \alpha]^T$ is referenced with respect to an origin set in the global coordinate system $[X\, Y]^T$. $\alpha$is the bearing of the vehicle. The control input to the AGV, $\boldsymbol{u}(k)$, consists of two components—the AGV's velocity and the steering-wheel angle $\boldsymbol{u} = [V\, \gamma]$. The system's model is

$$\dot{x}(t) = V \cos[\alpha(t) + \gamma] \tag{18}$$

$$\dot{y}(t) = V \sin[\alpha(t) + \gamma] \tag{19}$$

$$\dot{\alpha}(t) = \frac{V}{B} \sin\gamma \tag{20}$$

where B is the baseline of the vehicle. The observation model is

$$\boldsymbol{z}(t) = \boldsymbol{h}(\theta, \boldsymbol{x}) + \boldsymbol{w} = \begin{bmatrix} R \\ \phi \end{bmatrix} \tag{21}$$

where θ is a parameter (feature type), R is the range to the feature, ϕ is the bearing to the feature with respect to the vehicle heading, and $\boldsymbol{w}$ is the motion noise (assumed Gaussian).

Upon receipt of an observation (Δt time units after the last update), an EKF algorithm produces a prediction of the AGV's state $\boldsymbol{x}(k+1|k)$, and the error covariance $\boldsymbol{P}(k+1|k)$ associated with this prediction. (The detailed expressions for state transition and variance calculation are given in [11, sec. V].) The prediction step is followed by a *validation* step, during which the predicted position $\boldsymbol{x}(k+1|k)$ is used in conjunction with the observation to estimate the map cell which is of interest. The observation has the format $(type(\theta), R, \phi)$, where *type* belongs to the set $\{point, line, corner, edge\}$. The algorithm searches for features held in the map which can validate the observation, and selects one that has minimum Mahalanobis distance. The filter is then updated with the observation according to the standard EKF equations, using a different observation function $\boldsymbol{h}$ for each *type* of target observed.

A similar approach is described by Lapin [39] who studied an AGV with a tricycle wheel system. This robot uses an incremental optical encoder (on each rear wheel), an absolute encoder (on a steering shaft), and a camera-based landmark tracking system.

c) Applicability: Kalman filtering offers a powerful method for low-level fusion in mobile robots, provided that the filter's modeling assumptions can be satisfied, and that the uncertainty models that it requires are available and reliable. The data integration process executed by the Kalman filter is modular and can accommodate a large variety of sensory measurements as long as the error covariance matrices are available. There exist a well-developed body of literature on performance, stability and consistency of the filter, as well as tests for data validation and performance consistency. These tests provide means to reject bad data on-line, and to check that state-estimate errors satisfy certain conditions on their statistics in spite of the use of approximations [7, sec. 5.4, p. 388]. There also exist different techniques for error compensation in linearized filters such as the use of pseudonoise covariance and compensation for bias in errors. As we explain in Section II-C, there are applications where there is significant difficulty in modeling the sensor readings under a unified statistical model, and alternative techniques, mostly rule-based, have to be employed.

B. Low-Level Fusion with Known Statistics in Decentralized Architectures

So far we have assumed that a single central processor executes all the data fusion necessary for the robot. With the growing complexity of robotic platforms, centralized data fusion may become less attractive. Decentralization may be required due to increasing computational and communication loads, and due to reliability concerns. To save on storage and communication, robot designers may choose to perform local estimation of parameters from available data, followed by global fusion of the estimates. Alternatively they may use decentralized architectures that distribute the

Kalman filter equations among a number of sensing nodes, ensuring graceful degradation in the face of node failings.

1) Decentralization: To increase reliability and parallelization of computations, several architectures distribute filtering to interconnected local estimators. Rao *et al.* [48] suggested a fully decentralized multisensor architecture.[1] They use a fully connected decentralized architecture comprised of communicating nodes. The nodes are initialized with state estimates and state-covariances. Each node collects observations from its environment, validates them, and uses the validated observations to update its state via standard Kalman filtering. Each node then communicates its estimated state to all other nodes, and receives estimates from all other nodes. Once again the data are validated, and used by each node for a second update. The estimate available locally is now the global estimate, and is guaranteed to be the same one obtained by a fully centralized Kalman filter. The architecture is more resistant to processor failure than the centralized counterpart, and both computation and communication overheads associated with decentralization are low. While naturally more suitable for surveillance over large geographical areas with physically dispersed sensors, this architecture was tested using CCD cameras and optical sensors in an industrial-floor setting suitable for mobile robots [48, sec. 5].

2) Fusion of Estimates: Architectures such as the one studied in [48] require integration of estimates from several Kalman filters. In [6], Bar-Shalom and Fortmann study the fusion of estimates $\hat{\boldsymbol{x}}^i$ and $\hat{\boldsymbol{x}}^j$—provided respectively by estimators i and j—on the same target (both $\hat{\boldsymbol{x}}^i$ and $\hat{\boldsymbol{x}}^j$ can be estimates, or one can be an estimate and the other a prediction). In general, the state estimation error $\tilde{\boldsymbol{x}}^i(k) = \boldsymbol{x}(k) - \hat{\boldsymbol{x}}^i(k)$ and $\tilde{\boldsymbol{x}}^j(k) = \boldsymbol{x}(k) - \hat{\boldsymbol{x}}^j(k)$ are dependent due to the common process noise [e.g., $\boldsymbol{w}(k)$ in (12)]. The fact that the two measurement noise sequences are independent is not sufficient for independence of the estimation errors. However, in order to appreciate the form of the fusion rule, we assume first that $\tilde{\boldsymbol{x}}^i$ and $\tilde{\boldsymbol{x}}^j$ are indeed statistically independent, and further that they are normally distributed. The dynamics of the object tracked by the robot are described by (12) and the measurements are described by

$$\boldsymbol{z}^m(k+1) = \boldsymbol{H}^m(k+1)\boldsymbol{x}(k+1)+\boldsymbol{v}^m(k+1), \qquad m = i, j. \tag{22}$$

The optimal (minimum mean square error) combination of $\hat{\boldsymbol{x}}^i$ and $\hat{\boldsymbol{x}}^j$ (obtained by taking $\hat{\boldsymbol{x}}^i$ as "prior" mean of $\boldsymbol{x}$) is

$$\begin{aligned}\hat{\boldsymbol{x}}^{ij} &= \hat{\boldsymbol{x}}^i + \boldsymbol{P}^i(\boldsymbol{P}^i + \boldsymbol{P}^j)^{-1}(\hat{\boldsymbol{x}}^j - \hat{\boldsymbol{x}}^i)\\ &= \boldsymbol{P}^j(\boldsymbol{P}^i + \boldsymbol{P}^j)^{-1}\hat{\boldsymbol{x}}^i + \boldsymbol{P}^i(\boldsymbol{P}^i + \boldsymbol{P}^j)^{-1}\hat{\boldsymbol{x}}^j\end{aligned} \tag{23}$$

where $\boldsymbol{P}^i$ and $\boldsymbol{P}^j$ are the covariance matrices of $\boldsymbol{x}^i$ and $\boldsymbol{x}^j$, respectively. The covariance associated with the fused estimate $\hat{\boldsymbol{x}}^{ij}$ is

$$\boldsymbol{M}^{ij} = \boldsymbol{P}^i - \boldsymbol{P}^i(\boldsymbol{P}^i+\boldsymbol{P}^j)^{-1}\boldsymbol{P}^i = \boldsymbol{P}^i(\boldsymbol{P}^i+\boldsymbol{P}^j)^{-1}\boldsymbol{P}^j. \tag{24}$$

Equation (23) provides insight about the weights used to fuse the different estimates, and is often used as a (suboptimal) fusion rule for estimates even when the estimation errors are dependent. For example, in the robotic platform Blanche [14], an optical range finder and odometry are used simultaneously. A pair of position-estimates and their standard deviations $\{(x_0, \sigma_0), (x_m, \sigma_m)\}$ is calculated from the odometry readings and from a matching process.[2] Once the pairs $\{(x_0, \sigma_0), (x_m, \sigma_m)\}$ are available, a new estimate (x_c, σ_c) is calculated according to

$$x_c = x_0 + \frac{\sigma_0}{\sigma_0 + \sigma_m}(x_m - x_0) \tag{25}$$

$$\frac{1}{\sigma_c^2} = \frac{1}{\sigma_0^2} + \frac{1}{\sigma_m^2}. \tag{26}$$

As Cox [14] indicates, (25) and (26) provide an intuitively correct solution which takes into account with a larger weight the source whose uncertainty is smallest.

> For the case of the vehicle moving down a parallel corridor, very good estimates of the vehicle's orientation and position normal to the corridor can be made by the matcher, but the position of the vehicle along the corridor cannot be estimated by the matcher and its associated standard deviation is infinite. In these situations we rely on odometry until the matcher detects line features that are not parallel to the corridor.

The updated value is fed back to the odometry where it is used as the new value from which the current position is estimated.

We return now to the case of *dependent* state estimation errors, $\tilde{\boldsymbol{x}}^i(k|k)$ and $\tilde{\boldsymbol{x}}^j(k|k)$. The expression for the fused estimate and the covariance of the fused estimate are now

$$\begin{aligned}\hat{\boldsymbol{x}}^{ij}(k|k) = &\,\hat{\boldsymbol{x}}^i(k|k) + [\boldsymbol{P}^i(k|k) - \boldsymbol{P}^{ij}(k|k)]\\ &\times [\boldsymbol{P}^i(k|k) + \boldsymbol{P}^j(k|k) - \boldsymbol{P}^{ij}(k|k)\\ &- \boldsymbol{P}^{ij^T}(k|k)]^{-1}\\ &\times [\hat{\boldsymbol{x}}^j(k|k) - \hat{\boldsymbol{x}}^i(k|k)]\end{aligned} \tag{27}$$

and

$$\begin{aligned}\boldsymbol{M}^{ij}(k|k) = &\,\boldsymbol{P}^i(k|k) - [\boldsymbol{P}^i(k|k) - \boldsymbol{P}^{ij}(k|k)]\\ &\times [\boldsymbol{P}^i(k|k) + \boldsymbol{P}^j(k|k) - \boldsymbol{P}^{ij}(k|k)\\ &- \boldsymbol{P}^{ij^T}(k|k)]^{-1}\\ &\times [\boldsymbol{P}^i(k|k) - \boldsymbol{P}^{ij}(k|k)]^T.\end{aligned} \tag{28}$$

In these relations $\boldsymbol{P}^{ij}(k|k) = E\{\tilde{\boldsymbol{x}}^i(k|k)[\tilde{\boldsymbol{x}}^j(k|k)]^T\}$ (see [6, sec. 10.3] for the complete derivation).

3) Registrating: The fusion rules that we have used assume that the estimates $\hat{\boldsymbol{x}}^i(k)$ and $\hat{\boldsymbol{x}}^j(k)$ correspond to the same target. Several hypothesis-testing techniques are available to determine whether or not this is true [6, sec. 10.2–3], [15].

4) Robustness: Data consistency and data combining were further studied in a series of papers by Mintz and his co-workers on robust sensor fusion—using statistical decision theory [45], [56], [57]. Mintz *et al.* have developed a robust test of the hypothesis that data from different sensors is consistent, and a robust procedure for combining the data which pass this preliminary test. The observation

[1] Based on previous work by Hashemipour *et al.* [26] on decentralized multisensor structures for parallel Kalman filtering.

[2] The matching process uses range finder measurements to match features in a local map to a global line segment grid (see Section II-C).

model was $z = \theta + V$ where V represents additive sensor noise and θ is the sensed parameter of interest. Robustness in the context of this work refers to "the statistical effectiveness of the decision rules when the probability distribution of the observations noise and the *a priori* position information associated with the individual sensors are uncertain" [45]. Mintz *et al*. note that most studies of data fusion (such as the ones that we have quoted so far) assume that sensor noise can either be adequately modeled by Gaussian distributions with known means and covariances, or by distributions characterized only by specified first and second moment (using procedures that are equivalent to making Gaussian-distribution assumptions). Clearly, sampling distributions that possess heavy tails (*e.g.*, departures from the Gaussian model in the form of ϵ-contamination uncertainty classes) could result in significant errors if a Gaussian model is used to develop the decision rules.

C. Low-Level Fusion with Unknown Statistics

The methods of Section II-A and II-B rely on well-defined uncertainty models, and execute variants of maximum likelihood or maximum *a posteriori* estimation. Such models are difficult to develop when the navigation depends on patterns and signatures in maps (*e.g.*, in landmark navigation). Often they are also difficult to obtain for data that are combined from sensors that use significantly different principles and processing techniques. Yet diverse sensors are deliberately used by mobile robots to provide complementary information; one of the key advantages of the multisensor suite is that some sensors provide information unavailable from others.

When uncertainty models are either unavailable or are meaningless, *ad hoc* techniques have often been used within the domain of sensory data. In almost all cases, the advantages of a sound formal basis were lost, though several attempts were made to provide alternative "unified" frameworks in these situations. It is instructive to point out that in spite of their seemingly different procedures, most of these frameworks still execute the same *prediction*, *observation*, *validation*, and *updating* cycle used by the Kalman filter.

1) Rule-Based Sensor Fusion: To avoid the difficulty in modeling the sensor readings under a unified statistical model, several robotic applications use rule-based algorithms. Often these algorithms do not require an explicit analytical model of the environment. Expert knowledge of the sensors' characteristics and prior knowledge about the environment are used in the design of feature extraction, mapmaking, and self-location strategies. The resulting rules, after some experimentation, are usually simple and robust. The obvious disadvantage is limited domain of applicability—the insights used to create rules for a specific environment cannot be easily exported to other environments. Changes in the sensor suite and in the environment may require reevaluation and recompilation of the rule set.

a) Example three—fusion of sonar and infrared sensors: Flynn [23] developed a simple set of heuristic rules for a robot that uses an ultrasonic sensor and a near-infrared proximity sensor. As we observed in example one, ultrasonic sensors have wide angles but give relatively accurate depth measurements. Infrared sensors have excellent angular resolution but poor depth measurements—they are therefore well-suited for detection of edges and large depth discontinuities such as doorways. Fusion of measurements from both ultrasonics and infrared has the potential to provide robots with good readings in both depth and angular position.

The following rules have been used by Flynn to validate the data from the sensors and to determine which sensor to rely upon when conflicting information is obtained.

1) If the sonar reading is greater than the maximum range for the near-infrared sensor, then ignore the near-infrared output.
2) If the sonar reading is at its maximum value, then the real distance is greater.
3) Whenever the infrared sensor detects a change from no-detection to detection, and the associated sonar reading is less than 10 ft, then a valid depth discontinuity has been detected.

Using these few simple rules, the original sonar boundary was redrawn to take into account features that were found by the near-infrared sensor. Significant improvements in the mapping of laboratory environments were demonstrated in [23].

2) Geometric and Topological Maps: Rule-based sensor fusion is often used in map-assisted positioning, employing geometric and topological maps [22, sec. 8.3]. A *geometric map* represents objects according to their absolute geometric relationship using databases such as a grid map, line map, or polygon map. The geometric maps created by the mobile robot are matched against global maps that include the expected features. *Topological maps* record the geometric relationship between the observed features rather than the absolute position with respect to a frame of reference. Consequently, locating position using topological maps is independent of accumulated position errors, and depends on matching an observed relationship between features to a stored relationship in a graph. In some systems, a "signature" of a region is used to locate a robot within a large area.

We have already mentioned the autonomous robot vehicle Blanche [14], which uses geometric maps for navigation. Blanche employs an optical range finder and odometry. The robot's position in the map is obtained by fusing a known map of the robot's environment, range data, and odometry data. Positioning is based on matching a local grid map to a global line segment grid. This matching phase is similar to a recurring problem in computer vision, namely association of an image of arbitrary position and orientation relative to a model. The process starts by extracting features from the readings of the robot's sensor—in Blanche's case, an optical range finder (this step is often difficult and noisy). Next, correspondence between the image and the model features is determined, using a constrained search. In Blanche's case, the constraints are on the robot position—using a bound on the error from dead-reckoning data. When the correspondence is known, determination of the congruence is straightforward [14, pp. 198–200]. The calculation of the congruence using range finder measurements includes an estimate of the position's standard deviation. Once pairs of position-estimate and standard deviation are available from odometry (x_0, σ_0) and matching (x_m, σ_m), aggregation follows the estimation fusion approach described in (25) and (26).

Topological maps for navigation were used by Taylor [52], Kortenkamp and Weymouth [35], and Courtney and Jain [13]. An interesting application of topological maps was proposed recently by Janèt *et al*. [30]. They have considered the determination of the region within a map where a robot resides as a first step toward finer determination of the robot's position within that region. Janèt *et al*. used a "signature" of the region as the characteristic to identify, and employed Kohonen's neural networks to classify the signatures. This approach can exploit existing neural-network algorithms for *character recognition* for the (coarse) location of a robot in a previously-charted environment. The "signatures" can also be used to indicate that the robot has already visited in the past a region where it is wandering now.

3) "Unified" Frameworks: There have been several suggestions of unified frameworks for sensory measurements, when statistics are unknown or unavailable. Some are generalizations and extensions of Bayesian approaches, primarily toward Shafer–Dempster reasoning [10], [24], [29] and fuzzy logic. Others extend the Kalman-filtering method [27] or realize behavior-based algorithms [8], [41], [43] (see Section III-A).

Recently Joshi and Sanderson [32] have proposed a unified framework which characterizes data by an information-theoretic measure, namely the *coding complexity* of observed data on a Universal Turing Machine. This is a model-based approach, calculating complexity on the basis of system knowledge of the environment. The environment is specified by an "environmental model library," and the "best interpretation" is the environment model (structure and parameters) that minimizes the complexity of observed data [32, p. 2670]. The approach was tested on a problem of estimating 2-D *pose from touch* data using an imposed uncertainty model in the absence of the true underlying statistics.

III. High-Level Fusion

The low-level fusion algorithms discussed so far typically provide parameters and estimates to modules whose task is planning and execution of motion. Often the sensor fusion algorithm will supply input to a map maker or a path planner. Some of the techniques that we explore in this section are different; they provide direct command and control signals to the robot's locomotion subsystems—effectively integrating sensor interpretation with the planning task. In the next section (Section III-A) commands and control signals from different modules are integrated in an hierarchical navigation architecture. In the subsequent section (Section III-B), a neural network associates sensor input with commands to motors.

A. Behavior-Based Architectures

A widely studied approach to robot navigation in dynamic environments is the family of behavior-based robot control schemes, which use *situationally reactive behaviors* and *goal-oriented deliberative behaviors* [1]–[3], [12], [41], [47], [50]. With this approach, the mobile robot control problem is decomposed into a set of behaviors, rather than a set of functional modules. The behaviors are organized in hierarchical architectures (such as the *motor schema based architecture* [3] and the *subsumption architecture* [12], or in lateral voting schemes [47], [50]. As Rosenblatt and Thorpe [50] note, "reactive components provide the basic capabilities which enable the robot to achieve low-level tasks without injury to itself or the environment, while deliberative components provide the ability to achieve higher level goals and avoid mistakes which could lead to inefficiencies or even mission error."

1) The Subsumption Architecture: In explaining the difference between traditional architectures and those that use task-achieving behaviors, Brooks [12] compares a typical "horizontal decomposition" to his proposed "vertical decomposition." The horizontal decomposition, typical in the systems that we have explored in Section II, is represented by: 1) sensing, 2) mapping sensor data into a world representation, 3) planning, 4) task execution, and 5) motor control.

The vertical decomposition, on the other hand, uses *levels of competence*, and admits a layered structure like the following [12]:

Level 0. Avoid contact with objects (whether the objects move or are stationary).

Level 1. Wander around aimlessly without hitting things.

Level 2. "Explore" the world by observing places in the distance that look reachable and then heading for them.

Level 3. Build a map of the environment and plan routes from one place to another.

Level 4. Notice changes in the "static" environment.

Level 5. Reason about the world in terms of identifiable objects and perform tasks related to certain objects.

Level 6. Formulate and execute plans that involve changing the state of the world in some desirable way.

Level 7. Reason about the behavior of objects in the world and modify plans accordingly.

Corresponding to this structure are layers of control that constitute a *subsumption architecture*. At the bottom is *Level 0 control system*—a complete robot-control system that achieves Level 0 competence. The next level, *Level 1 control system*, is able to examine data from Level 0 system and inject data into Level 0's internal interfaces. With this level of control over Level 0's data flow, Level 1 control system is able to achieve Level 1 competence. The same process is repeated level by level, to create a hierarchical top-down architecture.

Higher-level layers are engaged in more abstract reasoning (which is computationally slow), while low-level layers

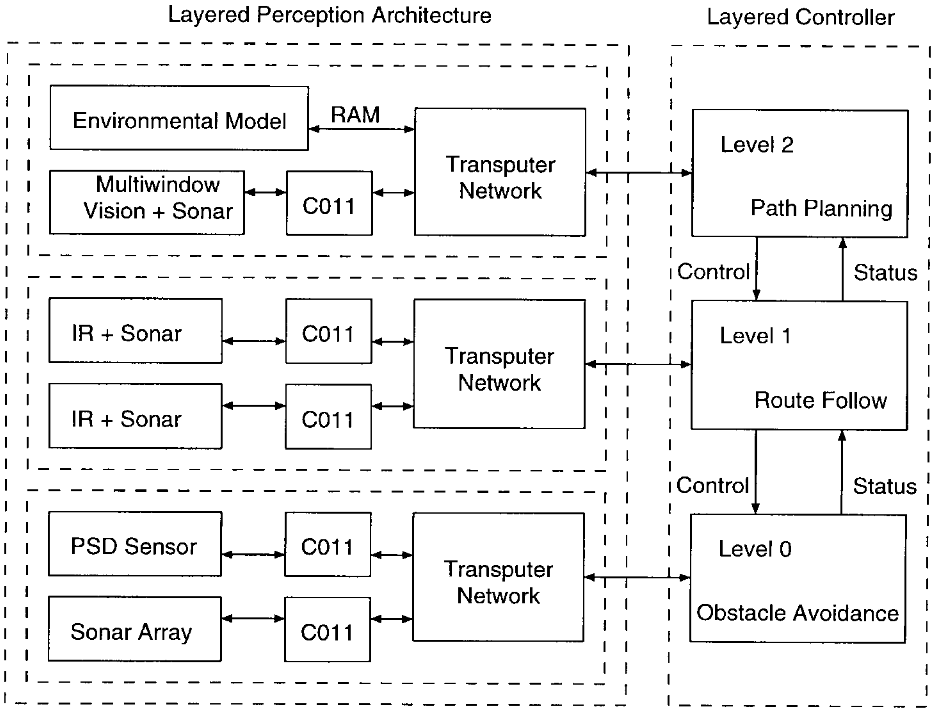

Fig. 3. A layered architecture for sensing [1].

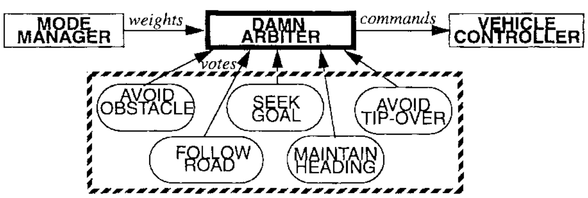

Fig. 4. Behaviors sending votes to the arbiter in the DAMN architecture [50].

engage in faster numerical calculations. In general each behavior is assigned a priority; the behavior with highest priority is in control while all others are subservient.

The subsumption architecture performs fusion of sensory data as well as fusion of estimates and decisions. Often different sensory subsystems are allocated to each level, though sensors can in principle be shared. Adams *et al.* [1] proposed a specific three-layer architecture of this kind for a physical robot (see Fig. 3). In their architecture, Level 0 achieves *obstacle avoidance*, Level 1 achieves *route following* and Level 2 achieves *path planning*. Level 0 uses a sonar array; Level 1 uses an integrated infrared/sonar sensor; and Level 2 uses a combination of vision and sonar.

Sensor fusion in subsumption architectures is performed at two levels. At the layer level, "low-level" fusion typically follows algorithms described in Section II, directly aggregating sensor readings. Fusion of sensory information between layers is achieved as the high-level layers draw information from lower-level layers to synthesize their own estimates and decisions. The translation of sensory input into layer decisions (the synthesis of reactive and deliberative behaviors) can be made in a number of ways. Several subsumption architectures use potential fields for obstacle-avoidance at level 0 [33], [49]. Others use rule-based expert systems and fuzzy logic [41], [53], [54]. Neural networks and genetic algorithms have also been tried [46].

2) Voting Schemes and Distributed Architectures: Several distributed "single layer" architectures were developed as an alternative to the top-down hierarchical architectures in behavior-based robot navigation [2], [50]. The rationale is that top-down architectures sometimes overly restrict lower-level layers, and dependence on lower layers requires continuous monitoring of the progress of desired actions with significant communications overhead [47], [50].

a) Example four—A voting scheme for off-road navigation: An example of a lateral voting architecture is the Distributed Architecture for Mobile Navigation (DAMN) (Fig. 4) suggested by Langer *et al.* [38] and by Rosenblatt and Thorpe [50]. They built and tested a behavior-based system for off-road navigation. The realized behaviors were: "avoid obstacles," "follow road," "seek goal," "maintain heading," and "avoid tip-over." Each behavior sent a vote to a *command arbiter*, "votes" being numbers between -1 and $+1$ for each vehicle action. The common arbiter then performed command fusion by weighting these decisions by a set of real numbers originating from a *mode manager* [50].

The fusion technique used by the arbiter effectively selects the command which has the most votes from the behaviors, rather than averaging the commands (which is the technique used by the *motor schema framework* [2]). Arbitration via vector addition may result in a command which is not satisfactory to *any* of the contributing behavior, and indeed the motor schema framework suffers from occasional trapping in local minima.

In example four, two levels of fusion are in use. Each module has its own sensory suite, and uses low-level fusion

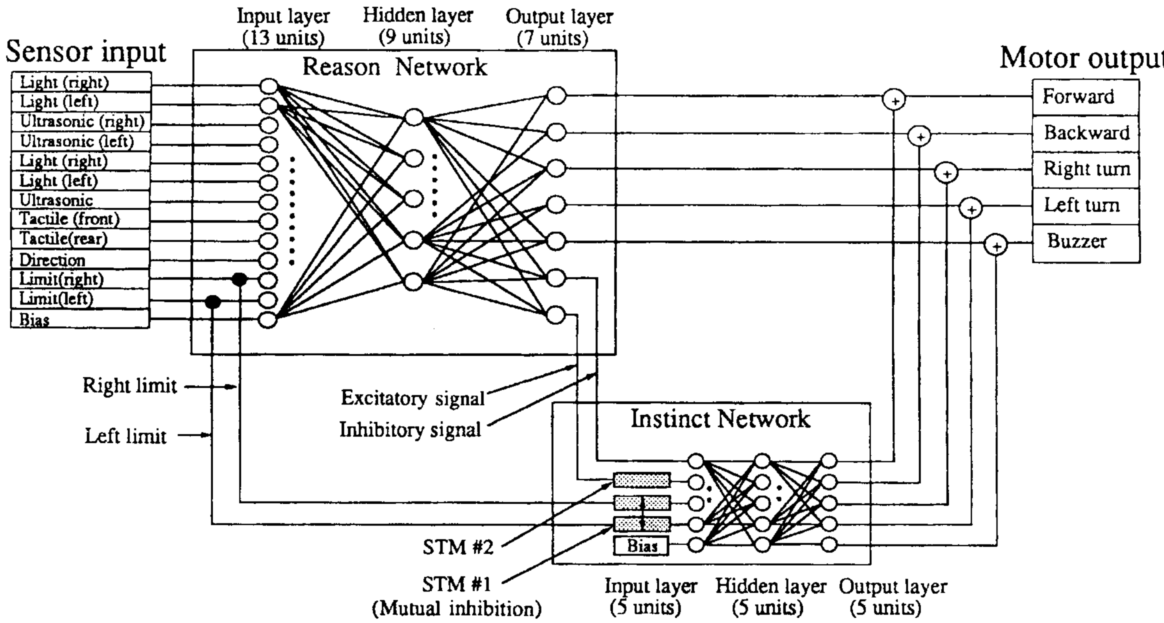

Fig. 5. A hierarchical neural network for a mobile robot [46].

algorithms on the data. High-level fusion, which is *command fusion* in this case, is done on the resulting suggested actions, not directly on the sensory measurements. Related architectures, using fuzzy logic and reinforcement learning, are described in [8].

B. "Unified" Frameworks—The Use of Neural Networks

In Section II-C2, we discussed efforts to develop unified frameworks for sensory measurements when statistics are unknown or unavailable. Neural networks provide a means to create such unified frameworks for sensory data. The basic idea is to train neural networks to associate given sets of sensory data (input) and desired robot action (output). When the training set spans the expected input domain, the neural network is expected to interpolate between the learned measurement-action tuples and provide the correct commands to the actuators.

1) Example Five—Hierarchical Neural Network for Mobile Robot Control: Nagata *et al*. [46] devised a mobile robot controlled by a hierarchical neural network. The network, shown in Fig. 5, receives input signals from sensors, and transmits on/off commands to motors. A memoryless *reason network* translates sensory inputs into behaviors. An example of a behavior is "move forward when the infrared sensor on the head detects light." An *instinct network* which has short-term memory units (STM's), translates sensory input into series of behavior patterns that the robot should be taking over time. For example: "repeat a cycle of right and left turns until the infrared sensor on the head receives light." The outputs to the motors are controlled through the interplay between sensory input, reasoned behavior, and memory of past actions. The result is the execution of learned pattern behaviors, imprinted in the neural-network weights and thresholds by a variant of the back-error propagation weight-adjustment algorithm.

Neural networks offer a unified framework for sensor fusion, since within the networks' architecture they do not impose strict requirements on data representation and interpretation. The network is "free" to discover complex patterns of data amalgamation during the training phase, and retune them during operation. There are few restrictions on the manner by which information from different sensors is fused, and with appropriate pre-processing and scaling, data can be represented to the input layers in different formats, levels and domains. The ability of many neural network models to interpolate gives them the additional advantage of processing successfully some input combinations for which they have not been specifically trained. Indeed, in Nagata's experiments, the robot was capable of determining "on its own" (i.e., successfully interpolate), correct and desirable actions—either explicitly from the behavior patterns that it had learned, or implicitly by using these patterns to infer the correct action.

The major hindrance in using neural-network frameworks for sensor fusion stems from (theoretical and practical) limitations on their learning and interpolating capabilities. Designers of robot-control architectures require structures that guarantee realization of input-output relationships to a specified accuracy in finite, known training epochs—a goal that remains elusive with neural networks. Estimates of network interpolation errors present a significant challenge. The great care that Nagata *et al*. had to exercise in their experiments to filter training data exemplifies these difficulties. For additional information about the use of neural networks for cross-modality sensor fusion, see [28].

IV. Future Research

Multisensor navigation of a robot in an arbitrary environment is an ill-posed problem due to the complexity and variety of environments, platforms, and sensors that are involved. It is therefore safe to predict that we shall continue to see a steady stream of new systems and new architectures aimed to solve specific sensor fusion problems for robot navigation. While not totally absent from the literature, several directions seem to call for further investigation.

Interrelationship Between Navigation Tasks: Most of the systems contemplated and built so far execute a *stop, look, localize* cycle, effectively assuming that self location, map making, path planning, motion planning and motor control are distinct, almost separate, operations—interrelated through simple input-output connections in a block diagram architecture. Yet it is clear, as one example, that the degree of resolution needed from a map is directly affected by requirements for path planning, as well as by map information already available (*e.g.*, if you want to flee from a room, discovery of one wide obstacle-free corridor to a doorway displaces the need to conduct a high-resolution analysis of all possible paths out). Task interrelationships is one avenue to reduce the computational complexity of tasks in robot navigation, but it is not compatible with the modular black-box approach that most mobile-robot designers take at the present time. The same observation holds for sensor placement and dynamic attention-focusing of sensors during robot motion.

Robust Sensor Fusion and Motion Control: It is likely that progress in robust control, along with the existing body of knowledge on statistical decision theory will provide new algorithms for robust sensor fusion and motion control. The robustness of the controller will determine the resolution of the fused data. As the degrees of uncertainty and modeling errors that can be tolerated by dynamic models become better understood and better quantified, feasibility and performance of sensor fusion in robotics will benefit similarly.

Adaptive and Learning Algorithms: It is unlikely that one technique or one architecture will provide a uniformly superior solution for navigation problems. Along with fusion of sensory data we shall see more systems executing command-and-control fusion, and the fusion of outputs from several algorithms. In this light it is useful to investigate *dynamic credit assignment* to algorithms and sensors; dynamic, environment-dependent *attention focusing* for sensors and algorithms; and *adaptive selection of modules* for decision and control in time-varying environments. The current availability of sensor suites of wide, varied scope, and the existence of a rich library of navigation architectures and sensor fusion algorithms provide solid foundation for progress in these directions.

Biological Sensor Fusion: Studies in sensor fusion for computer vision and mobile robots often cite models of biological fusion, such as the ones conducted on pigeons and bats [36], [51]. However the use of these models in actual machines is often metaphorical, using the biological architecture as a general guideline. The apparent success of living creatures in executing complex modes of navigation using multisensory input, and the ever-growing body of knowledge on biological sensory and motor control systems, point to significant potential benefits from research in this direction.

References

[1] M. D. Adams, H. Hu, and P. J. Probert, "Toward a real-time architecture for obstacle avoidance and path planning in mobile robot," in *Proc. IEEE Int. Conf. on Robot. Automat.*, Mar. 1990, pp. 584–589.
[2] R. C. Arkin, "Motor schema based navigation for a mobile robot," *Proc. IEEE Int. Conf. on Robot. Automat.*, 1987, pp. 264–271.
[3] R. C. Arkin and R. R. Murphy, "Autonomous navigation in a manufacturing environment," *IEEE Trans. Robot. Automat.*, vol. 6, pp. 445–454, 1990.
[4] N. Ayache and O. D. Faugeras, "Building, registrating and fusing noisy visual maps," *Int. J. Robot. Res.*, vol. 7, no. 6, pp. 45–65, 1988.
[5] ——, "Maintaining representations of the environment of a mobile robot," *IEEE Trans. Robot. Automat.*, vol. 5, no. 6, pp. 804–819, 1989.
[6] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. San Diego: Academic, 1987.
[7] Y. Bar-Shalom and X. R. Li, *Estimation Tracking*. Boston: Artech House, 1993.
[8] H. R. Beom and H. S. Cho, "A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning," *IEEE Trans. Syst., Man Cybern.*, vol. 25, pp. 464–477, 1995.
[9] J. C. Bezdek, Special Issue "Fuzziness vs. probability–the Nth round," *IEEE Trans. Fuzzy Syst.*, vol. 2, no. 1, 1994.
[10] P. L. Bogler, "Shafer-Dempster reasoning with applications to multisensor target identification system," *IEEE Trans. Syst., Man Cybern.*, vol. 17, pp. 968–977, 1987.
[11] S. Borthwick and H. Durrant-Whyte, "Dynamic localization of autonomous guided vehicles," in *Proc. 1994 IEEE Int. Conf. on Multisensor Fusion*, 1994, Las Vegas, NV, pp. 92–97.
[12] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Automat.*, vol. 2, pp. 14–23, 1986.
[13] J. Courtney and A. Jain, "Mobile robot localization via classification of multisensor maps," *Proc. IEEE Int. Conf. on Robot. Automat.*, San Diego, CA, May 8–13, 1994, pp. 1672–1678.
[14] I. J. Cox, "Blanche—An experiment in guidance and navigation of an autonomous robot vehicle," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 193–204, 1991.
[15] I. J. Cox and M. L. Miller, "On finding ranked assignments with applications to multitarget tracking and motion correspondence," *IEEE Trans. Aerospace Electron. Syst.*, vol. 1, no. 1, pp. 486–489, 1995.
[16] I. J. Cox and G. T. Wilfong, *Autonomous Robot Vehicles*. New York: Springer Verlag, 1990.
[17] J. Crowley, "World modeling and position estimation for a mobile robot using ultrasonic ranging," *Proc. IEEE Int. Conf. on Robot. Automat.*, Scottsdale, AZ, May 14–19, 1989, pp. 674–680.
[18] B. Dasarathy, *Decision Fusion*. Los Alamitos, CA: IEEE Computer Soc., 1994.
[19] H. F. Durrant–Whyte, *Integration, Coordination, and Control of Multi-Sensor Robot Systems*. Boston: Kluwer, 1987.
[20] ——, "Sensor models and multisensor integration," *Int. J. Robot. Res.*, pp. 97–113, vol. 7, no. 6, 1988.
[21] O. D. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Boston, MA: MIT Press, 1993.
[22] L. Feng, J. Borenstein, and H. R. Everett, "Where am I—sensors and methods for autonomous mobile robot positioning," Tech. Rep. UM-MEAM-94-21, Univ. Michigan, Ann Arbor, 1994.
[23] A. M. Flynn, "Combining sonar and infrared sensors for mobile robot navigation," *Int. J. Robot. Res.*, pp. 5–14, Dec. 1988.
[24] T. Garrey, J. Lowrance, and M. Fischler, "An inference technique for integrating knowledge for disparate sources," *Proc. 7th Int. Joint Conf. on Artificial Intell.*, Vancouver, BC, 1981, pp. 319–325.
[25] R. Giralt, R. Chatila, and M. Vaisset, "An integrated navigation and motion control system for autonomous mobile robots," in *Robotics Research*, vol. 1. Cambridge, MA: MIT Press, 1983, pp. 191–214.
[26] H. R. Hashemipour, S. Roy, and A. J. Laub, "Decentralized structures for parallel Kalman filtering," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 88–93, 1988.

[27] L. Hong and G.-J. Wang, "Integrating multisensor noisy and fuzzy data," *Proc. 1994 Int. Conf. on Multi-Sensor Fusion and Integration for Intell. Syst.*, 1994, Las Vegas, NV, pp. 199–206.
[28] T. Huntsberger, "Biologically motivated cross-modality sensory fusion system for automatic target recognition," *Neur. Networks*, vol. 8, no. 7/8, pp. 1215–1226, 1995.
[29] T. Huntsberger and S. N. Jayamamurthy, "A framework for multi-sensor fusion in the presence of uncertainty," in *Proc. Workshop on Spatial Reasoning and Multi-sensor Fusion*, 1987, St. Charles, IL, pp. 345–350.
[30] J. A. Janét, R. Gutierrez–Osuna, T. A. Chase, M. White, and R. C. Luo, "Global self-localization for autonomous mobile robots using self-organizing Kohonen neural networks," in *Proc. Int. Conf. on Intell. Robots and Syst.*, 1995, Los Alamitos, CA, vol. 3, pp. 504–509.
[31] A. H. Jazwinsky, *Stochastic Processes and Filtering Theory*. New York: Academic, 1970.
[32] R. Joshi and A. C. Sanderson, "Multisensor fusion and unknown statistics," in *Proc. IEEE Int. Conf. on Robot. Automat.*, Piscataway, NJ, 1995, pp. 2670–2677.
[33] O. Khatib, "Real time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 96–99, 1986.
[34] L. Kleeman and R. Kuc, "Mobile robot sensor for target localization and classification," *Int. J. Robot. Res.*, vol. 14, no. 4, pp. 295–318, 1995.
[35] D. Kortenkamp and T. Weymouth, "Combining sonar and vision sensing in the construction and use of topological maps for mobile robots," *IEEE Trans. Robot. Automat.*, under review.
[36] M. L. Kreithen, "Orientational strategies in birds: a tribute to W. T. Keeton," in *Behavioral Energetics: The Art of Survival in Vertebrates*. Columbus, OH: Ohio State Univ., 1983, pp. 3–28.
[37] D. J. Kriegman, E. Triendl, and T. O. Binford, "Stereo vision and navigation in buildings for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 5, no. 6, pp. 792–803, 1989.
[38] D. Langer, J. K. Rosenblatt, and M. Hebert, "A behavior-based system for off-road navigation," *Int. J. Robot. Res.*, vol. 10, no. 6, pp. 776–782, 1994.
[39] B. D. Lapin, *Adaptive Position Estimation for an Automated Guided Vehicle*, SPIE, vol. 1831: Mobile Robots 7, 1992, pp. 82–94.
[40] J. J. Leonard, H. F. Durrant–Whyte, and I. J. Cox, "Dynamic map building for an autonomous mobile robot," *Int. J. Robot. Res.*, vol. 11, no. 4, 1992.
[41] W. Li, "Fuzzy logic based robot navigation in uncertain environments by multisensor integration," in *Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intell. Syst.*, 1994, pp. 259–265.
[42] R. C. Luo and M. G. Kay, "Multisensor integration and fusion in intelligent systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, no. 5, pp. 901–931, 1989.
[43] D. Madsen and R. Gunderson, "An autonomous rover navigation scheme using fuzzy logic," in *Proc. Joint Conf. on Informat. Sci.*, 1994, Pinehurst, NC, pp. 204–207.
[44] S. Maeyama, A. Ohya, and S. Yuta, "Non-stop outdoor navigation of a mobile robot," in *Proc. Int. Conf. on Intell. Robots and Syst.*, 1995, vol. 1, pp. 130–135.
[45] R. McKendall and M. Mintz, "Robust fusion of location informtion," in *Proc. Int. Conf. on Robot. Automat.*, Philadelphia, PA, 1988, pp. 1239–1242.
[46] S. Nagata, M. Sekiguchi, and K. Asakawa, "Mobile robot control by a structured hierarchical neural network," *IEEE Contr. Syst. Mag.*, vol. 10, no. 3, pp. 69–76, 1990.
[47] D. W. Payton, J. K. Rosenblatt, and D. M. Keirsey, "Plan guided reaction," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 6, pp. 1370–1382, 1990.
[48] B. S. Rao, H. F. Durrant–Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *Int. J. Robot. Res.*, vol. 12, no. 1, pp. 20–44, 1993.
[49] E. Rimon and D. E. Koditchek, "Exact robot navigation using artifical potential functions," *IEEE Trans. Robot. Automat.*, vol. 8, no. 5, pp. 501–518, 1992.
[50] J. K. Rosenblatt and C. E. Thorpe, "Combining multiple goals in a behavior-based architecture," in *Proc. Int. Conf. on Intell. Robots and Syst.*, vol. 1, 1995, pp. 136–141.
[51] J. A. Simmons *et al.*, "Composition of biosonar signals for target recognition by echolocating bats," *Neur. Networks*, vol. 8, no. 7/8, pp. 1239–1262, 1995.
[52] C. Taylor, "Building representations for the environment of a mobile robot from image data," in *Proc. 1991 SPIE Conf. on Mobile Robots*, Boston, MA, 1991, pp. 331–339.
[53] J. Vandorpe and H. Van Brussel, "A reflexive navigation algorithm for an autonomous mobile robot," *Proc. 1994 Int. Conf. on Multisensor Fusion and Integration for Intell. Syst.*, Las Vegas, NV, 1994, pp. 251–258.
[54] J. Vandorpe, H. Van Brussel, and H. Xu, "LiAS: A reflexive navigation Architecture for an intelligent robot system," *IEEE Trans. Indust. Electron.*, vol. 43, no. 3, pp. 432–440, June 1996.
[55] E. Waltz and J. Llinas, *Multisensor Data Fusion*. Boston, MA: Artech House, 1990, ch. 6.
[56] M. Zeytinoglu and M. Mintz, "Optimal fixed-size confidence procedures for a restricted parameter space," *Ann. Statistics*, vol. 12, no. 3, pp. 945–957, 1984.
[57] ——, "Robust fixed-size confidence procedures for a restricted parameter space," *Ann. Statistics*, vol. 16, no. 3, pp. 1241–1253, 1988.

Moshe Kam received the B.Sc. degree from Tel Aviv University, Israel, and the M.S. and Ph.D. degrees from Drexel University, Philadelphia, in 1977, 1985, and 1987, respectively.

He is presently a Professor of Electrical and Computer Engineering at Drexel University and Director of Drexel's Data Fusion Laboratory. His research interests are decision fusion, multisensor systems, robot navigation, forensic pattern recognition, and engineering education.

Dr. Kam has received a National Science Foundation Presidential Young Investigator award in 1990 and the C. Holmes MacDonald Award for the Outstanding Young Electrical Engineering Educator in 1991. Along with his students G. Cane and B. Callahan, he received the Outstanding Engineering Achievement Award of the Delaware Valley in 1996. for the autonomous tracked mobile robot ATR 6000.

Xiaoxun Zhu received the B.S. degree from the University of Science and Technology of China and the M.S. degree from Drexel University, Philadelphia.

He is presently a Ph.D. candidate at the Electrical and Computer Engineering Department of Drexel University, studying adaptive decision fusion and failure detection in multisensor systems. From 1992 to 1995 he was with Gradient Technology as a software engineer in the area of speech analysis.

Mr. Zhu is the first Hill Fellow at Drexel University, having been granted the George Hill, Jr. College of Engineering award in 1996.

Paul Kalata received the B.S. degree from Northwestern University, Evanston, IL, in 1963, and the M.S. and Ph.D. degrees from the Illinois Institute of Technology, Chicago, in 1966 and 1974, respectively.

From 1974 to 1979 he was with RCA Missile and Surface Radar, working on optimal filtering applications in radar control, tracking systems and missile guidance. Since 1979 he has been on the faculty of Drexel University in the Department of Electrical and Computer Engineering. His research interests are adaptive estimation, adaptive control, stochastic signal processing, and education in control engineering.

Sensor fusion concerns several field of research including multiple sensors like GPS, IMU, Wheel encoders. For the Outdoor mobile Robot to estimate the localization DGPS is used for improved accuracy. It is not sufficient to make use of only DGPS, as DGPS runs on Line of Sight (LoS), for better accuracy fused data from GPS, IMU and Odometry is preferred.Â In general, localization of mobile robot navigation in outdoor environment GPS, IMU and odometry data is incorporated in the design. These are entirely self-contained within the robot in the sense that they are not dependent on the transmission of signals from the robot or reception from an external source. Inertial sensors measurements are independent of robot physical parameters. We review techniques for sensor fusion in robot navigation, emphasizing algorithms for self-location. These find use when the sensor suite of a mobile robot comprises several different sensors, some complementary and some redundant. Integrating the sensor readings, the robot seeks to accomplish tasks such as constructing a map of its environment, locating itself in that map, and recognizing objects that should be avoided or sought.