Coordinated dispatching of proximity sensors for the surveillance of manoeuvring targets

Michael D. Naish [a,*], Elizabeth A. Croft [b], Beno Benhabib [c]

[a] *Department of Mechanical and Materials Engineering, University of Western Ontario, London, Ontario, Canada N6A 5B9*

[b] *Department of Mechanical Engineering, University of British Columbia 2324 Main Mall, Vancouver, British Columbia, Canada V6T 1Z4*

[c] *Department of Mechanical and Industrial Engineering, University of Toronto 5 King's College Road, Toronto, Ontario, Canada M5S 3G8*

Abstract

The surveillance of a manoeuvring target with multiple sensors in a coordinated manner requires a method for selecting and positioning groups of sensors in real time. Herein, the principles of dispatching, as used for the effective operation of service vehicles, are considered. The object trajectory is first discretized into a number of demand instants (data acquisition times), to which groups of sensors are assigned, respectively. Heuristic rules are used to determine the composition of each sensor group by evaluating the potential contribution of each sensor. In the case of dynamic sensors, the position of each sensor with respect to the target is also specified. Our proposed approach aims to improve the quality of surveillance data in three ways: (1) The assigned sensors are manoeuvred into "optimal" sensing positions, (2) the uncertainty of the measured data is mitigated through sensor fusion, and (3) the poses of the unassigned sensors are adjusted to ensure that the surveillance system can react to future object manoeuvres. If *a priori* target trajectory information is available, the system performance may be further improved by optimizing the initial pose of each sensor off-line. The advantages of dispatching dynamic sensors over similar static-sensor systems are demonstrated through comprehensive simulations.

Key words: sensor planning, dynamic sensors, sensor fusion, demand instants, heuristic dispatching, multisensor surveillance, sensor system configuration

[*] Corresponding author.
Email addresses: naish@eng.uwo.ca (Michael D. Naish), ecroft@mech.ubc.ca (Elizabeth A. Croft), beno@mie.utoronto.ca (Beno Benhabib).

Article published in Journal of Robotics and Computer Integrated Manufacturing 19 (2003)

1 Introduction

Currently, objects to be manipulated by industrial robots must be presented at specific (fixed) locations and times to ensure success. Variations in object presentation, unless accounted for a priori, may result in the failure of the robot to grasp the object. Thus, an important research problem in the field of autonomous robotic systems has been *on-line robot-motion planning for the interception of manoeuvring objects*.

Recent research efforts have resulted in numerous methodologies for target interception that are capable of responding to real-time variations in object location (e.g., [1,2]). In order to determine a suitable interception point, however, these methods have relied on real-time estimates of the object's motion as perceived by static sensors (often, only one). Unfortunately, sensor measurements tend to be inherently noisy and prone to obstructions. Thus, this paper's objective is to contribute to the state-of-the-art in moving-object surveillance by utilizing a reconfigurable surveillance system that may comprise multiple static and dynamic sensors. The approach taken is similar to techniques developed for the coordination and dispatching of service vehicles.

Just as the system response for a taxi company can be improved by effective dispatching, the quality of the sensor data acquired by a set of sensors can be improved through appropriate selection and positioning; however, sensors introduce an additional consideration. Dispatching multiple sensors (as opposed to only one) to observe the target at a particular location provides an opportunity to significantly improve the quality and robustness of the data. Specifically, sensor fusion may be used to combine information from these multiple, coordinated sensors into a single, more accurate, representation [3].

1.1 Vehicle Dispatching

Strategies and techniques used for scheduling and dispatching service vehicles have been investigated by a number of researchers [4–10]. Typically, each vehicle is assessed for assignment to a demand based on a set of evaluation criteria, such as nearest-vehicle, longest-idle-vehicle, and least-utilized-vehicle.

Various techniques have been proposed to balance the goals of efficiency of operation and an equitable distribution of workload. These include fuzzy logic [5], neural networks [6], and genetic programming [7]. Powell [9], and Ran and Boyce [10], compare a number of other optimization approaches to the problem of dynamic vehicle allocation (DVA): (1) deterministic networks, (2) stochastic networks, (3) Markov decision processes, and (4) stochastic programming.

Among the above mentioned techniques, the rolling-horizon concept developed by Psaraftis [8] to solve a class of dynamic vehicle-routing problems is of particular interest to our work. In this algorithm, developed for cargo-ship assignments, during any given iteration, only known cargoes that must be picked up within the time period of the rolling horizon (from the current instant to a certain period of time into the future) are considered. Permanent assignments are made to eligible ships only for those cargoes available at the beginning of the rolling horizon (since more immediate events are known with greater certainty); all other cargoes are assigned tentatively to eligible ships. The rolling horizon is then shifted to the next time step.

1.2 Sensor Dispatching

Systems that utilize sensors for tasks such as inspection, identification, tracking, and surveillance often employ approaches that are specific to the requirements of a particular application. The system proposed in [11], for example, considers the placement of sensors to optimize both feature visibility and measurement reliability. An off-line "generate-and-test" method is used to determine the best sensor position for measuring each object feature with a single sensor. These simple solutions are then combined on-line to accomplish more complex tasks such as the placement of a pair of cameras for stereo inspection and sequential inspection of a set of features. Zhang [12] addressed the problem of optimally placing multiple identical sensors to be used for sensor fusion in a 2-D workspace, by minimizing the magnitude of the sensor measurement covariances.

The system proposed in [13] proposes to handle moving objects by discretizing time and computing new viewing configurations for each time interval while attempting to minimize changes in sensor position from one time interval to the next. For a robot task known a priori, Abrams *et al.* [14], seek viewpoints that satisfy task constraints over the entire task interval; if none exist, the interval is divided until satisfactory viewpoints are found. In the case that multiple sensors are used, each is allowed to be active for a different time interval.

Matsuyama *et al.* [15] use the concept of a storyboard [1] to plan the motion of multiple cameras. Camera layouts (in 2-D) are determined through optimization; camera actions and temporal camera switching and coordination are determined through heuristics. Once a plan has been developed off-line, an on-line camera control system is used to adjust the camera switching time

[1] A *storyboard* is a set of characteristic snapshots that define a desired image sequence. It is often used to plan character actions and camera positions for animation and film.

to account for spatial and temporal deviations from the storyboard, ensuring smooth dynamic scenes.

Horling *et al.* [16] take an agent-based approach to the task of coordinating a number of different fixed sensors. The workspace is divided into a number of sectors, each with a managing agent. The managing agents attempt to recruit sensors (each with a corresponding agent) to scan the workspace for targets and, once found, provide synchronized measurements that may be fused to accurately estimate the real-time object location.

2 Problem Formulation

This paper considers the sensor dispatching problem for the surveillance of a manoeuvring target using proximity sensors. Before addressing our proposed solution methodology to this problem in Section 3, surveillance, data acquisition, and dispatching are briefly discussed in the following subsections, 2.1–2.3, respectively.

2.1 Surveillance

Surveillance of a manoeuvring object refers to obtaining estimates of particular object parameters at predetermined times or positions along the object trajectory. Herein, the specific parameter of interest is the Cartesian position of a target. The times at which this information is desired are referred to as demand instants, t_j. (Often, the interval between each demand instant is a constant value, Δ.) The position of the target for a particular demand instant is referred to as a demand point, D_j, Fig. 1.

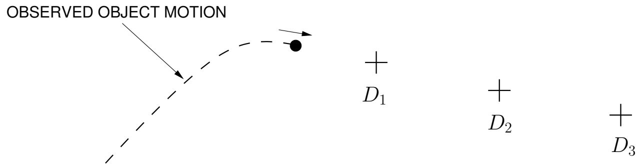


Fig. 1. Future demand points corresponding to demand instants t_1, t_2, and t_3, as predicted from observed past object motion at current time.

In this paper, it is assumed that the object trajectory covers a large part of the surveillance system's workspace—namely, the surveillance task spans relatively long distances. Thus, continuous estimation of the object parameters (i.e., tracking) is not considered in this paper. Additionally, the sensors that comprise the surveillance system may be static or dynamic; the speed and manoeuvrability of dynamic sensors are considered to be inferior to the object. As a result, it would be necessary to distribute the sensors throughout the workspace to ensure a degree of data consistency over the entire object trajectory.

2.2 Data Acquisition

Herein, a three-dimensional proximity measurement is defined as the range, bearing, and elevation to the object from a known sensor pose (that is, a known sensor position and orientation). For the sensor shown in Fig. 2, the range, r, is the linear distance between the object and the sensor frame. The bearing, θ, is the angular difference between the orientation of the sensor axis with respect to the x-z plane, α_s, and the object position. The elevation, ϕ, is the angular difference between the orientation of the sensor axis with respect to the x-y plane, β_s, and the object position. The Cartesian position of the object relative to a world coordinate frame is, thus, determined through:

$$
\begin{aligned}
x_o &= x_s + r\cos(\theta + \alpha_s)\cos(\phi + \beta_s), && \text{(1a)}\\
y_o &= y_s + r\sin(\theta + \alpha_s)\cos(\phi + \beta_s), && \text{(1b)}\\
z_o &= z_s + r\sin(\phi + \beta_s). && \text{(1c)}
\end{aligned}
$$

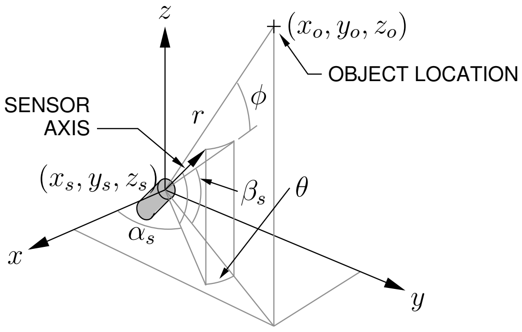


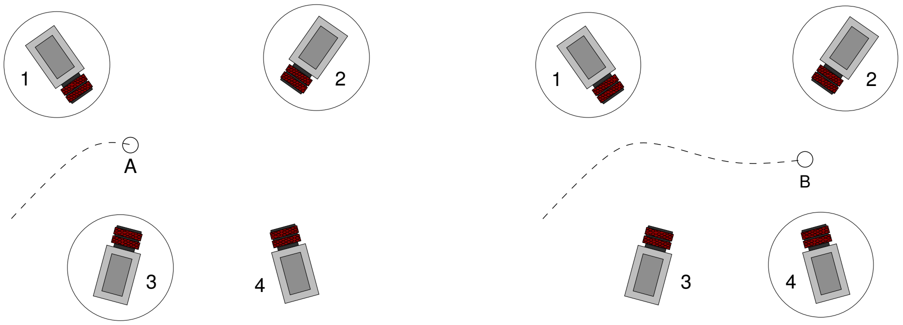
Fig. 2. Measurement variables for a proximity sensor.

There exist a number of different non-contact sensors which may be used to measure the proximity of an object. These generally utilize laser-triangulation, phase- or amplitude-modulation based electro-optical transducers, ultrasonic transducers, or stereo vision [17, 18]. In a multi-sensor surveillance system, prior to fusion, data from all such sensors must be preprocessed to perform data alignment. Data alignment transforms incoming sensor data into a common frame of reference through coordinate transformations and unit conversions.

Specific data-fusion methodologies suitable for parameter estimation include the least squares estimator (LS) and its variations: weighted least squares (WLS), Bayesian weighted least squares (BWLS), maximum likelihood estimate (MLE), and minimum squared error (MSE) [19]. These methods estimate the value that best fits a set of redundant measurements using an established optimization technique. The geometric fusion approach [20] characterizes the uncertainty of a sensor measurement by an ellipsoidal volume representing the sensor covariance matrix. Geometric fusion produces results that are similar to WLS with fewer computational resources. The Kalman Filter (KF), and Extended Kalman Filter (EKF) for non-linear systems, is another popular technique for the fusion of multisensor data [21, 22]. Kalman filtering is an iterative technique that may be formulated entirely as scalar equations, making it computationally efficient and thus suitable for real-time applications.

2.3 Dispatching

Within the context of optimal dispatching, sensor fusion does not need to combine data from all of the sensors in the system. Instead, a subset of sensors ($k \leqslant n$, where k is the subset size and n is the total number of sensors) is selected, or assigned, to survey the object at a particular demand instant. Herein, this group of sensors is referred to as a *fusion subset*, Fig. 3. At the time of data acquisition, only sensors that comprise the fusion subset become operational. The k measurements are then fused into a single representation.



(a) Demand point: A. Fusion subset: 1-2-3.

(b) Demand point: B. Fusion subset: 1-2-4.

Fig. 3. Coordination strategy for two demand points using 4 sensors. Sensors comprising each fusion subset are circled.

The general dispatching problem addressed in this paper is, thus, stated as: *Given a set of sensors, determine the best fusion subset (of sensors) for the next immediate demand instant, while considering a rolling horizon of possible future demand points*. The implications of solving this dispatching problem effectively are more important for dynamic (mobile) sensors than static ones.

For the former, the dispatching problem also involves determining the optimal placement of the (fusion subset) sensors in the workspace for the surveying of the object at the next immediate demand instant, as well as the distribution of the remaining sensors for potential future use.

In the above context, dispatching may then be accomplished using two complementary strategies. First, a "coordination strategy" specifies which sensors will be used for surveillance at each and every demand instant in the rolling horizon. The goal of the dispatcher is to select, for each demand instant, the most appropriate sensors for inclusion in a fusion subset. This selection should be based on a logical search through the sensor set, using an objective function, to evaluate the fitness of each sensor. Next, a "positioning strategy" specifies the pose of each of the sensors. This latter strategy positions each sensor whether it has been assigned (by the coordination strategy) to the first demand instant at hand, or to any one of the future demand instants.

3 Proposed Dispatching Strategy

This section outlines an approach by which sensors can be evaluated, selected, and positioned for the surveillance of a manoeuvring object. The approach is first formulated as a generic optimization that attempts to simultaneously determine suitable coordination and positioning strategies, thereby implementing an overall dispatching strategy for the system. This concurrent-solution method serves as a precursor to the heuristic-based method, which is better suited to real-time applications and adopted in this paper.

3.1 Objective Function

The coordination strategy requires that the dispatcher select the most appropriate sensors for inclusion in a fusion subset. This selection must be based on some estimate of the data quality that each sensor can provide for the demand point at hand. One such estimate of sensing quality is a measure of visibility, v_{s}, that would be inversely proportional to the sensor's measurement uncertainty and vary as its pose changes relative to the measurand (demand point location). For a multi-sensor fusion subset, the individual visibilities of all the sensors considered for inclusion in the fusion subset must be combined into a single metric, v_{f}, that would represent the benefit of this subset with respect to other possible subsets:

$$v_{\mathrm{f}} = f\Big((v_{\mathrm{s}})_i, \quad i = 1, \ldots, k\Big). \tag{2}$$

The value of the overall visibility metric given in Eq. (2) would vary as we choose different (combinatoric) sets of k sensors (from the n available) and vary their locations with respect to the (predicted) location of the demand point. An effective search methodology is necessary to determine the best set of sensors and their locations in order to maximize v_{f}.

3.2 Optimal Search Methodology

Having an objective function by which different configurations of sensors may be assessed allows for the determination of optimal solutions to the dispatching problem. One such limited-scope methodology would be only to consider a finite segment of the object trajectory during the search for best fusion subsets. This segment, referred to as the rolling horizon, would include only a limited number, m, of demand instants.

A two-level nested search would have to be carried out to simultaneously determine all the “best” fusion subsets and the locations of the sensors within each subset when acquiring information regarding their own respective object demand location (on the rolling horizon):

1. Coordination strategy: A combinatorial search technique is used to select m combinations of k sensors from the set of all sensors, n; one fusion subset for each demand point.
2. Positioning strategy: The best achievable pose for each sensor with respect to its demand point is determined using a constrained non-linear search technique. The best achievable pose is one that maximizes the visibility of the object at the demand instant, constrained by the dynamic limitations of the sensor (i.e., since one sensor may belong to multiple fusion subsets, the best achievable pose for a specific demand point may be inferior to the globally-optimal pose that could have been obtained with sufficiently fast sensors).

In the above optimization algorithm, for each demand point, Eq. (2) could be used to evaluate the fusion subset and the sensor poses.

While the above approach to dispatching would yield locally-optimal solutions, the required extensive combinatorial and non-linear searches, even when m is small, would be computationally demanding, potentially rendering the dispatching method unsuitable for on-line implementation, even with faster computers. Thus, in this paper, a heuristic approach to sensor dispatching that follows the general principle of the above locally-optimal solution is adopted.

3.3 A Heuristic Approach to Dispatching

There exist two primary differences between the above optimal method and the heuristic dispatching solution proposed below to the multisensor surveillance problem. First, the heuristic approach does not attempt to optimize each fusion subset, but rather considers each sensor individually; secondly, instead of considering all of the demand instants concurrently, each is considered sequentially (from the beginning to the end of the rolling horizon—$t_1 \ldots t_m$). An "assignment and positioning" search considers only the "first" demand instant and assigns to it the best k sensors, while a "preassignment and prepositioning" search method considers the remaining demand instants on the rolling horizon and makes pseudo-assignment of corresponding fusion subsets, which become permanent only after the "first" demand instant has been serviced.

Assignment and Positioning

Given the predicted location of the first demand point on the rolling horizon, D_j, the assignment and positioning method selects an optimal subset of sensors (of size k) from the set of all sensors, n, to service this demand instant (i.e., coordination strategy). The search method also specifies a desired pose for each sensor at the time of data acquisition (i.e., positioning strategy), Fig. 4. Assignment for t_j (i.e., selection of sensors and their best locations) occurs only once during each search interval (the time between the previous and current demand instants). (It is triggered by an object entering the workspace or the completion of the previous search interval.) Namely, the selection of the sensor set cannot be altered until the first demand instant on the rolling horizon has been serviced; however, the pose of each sensor may be altered in real-time. Pose adjustment would be handled by a replanning method described later in this paper.

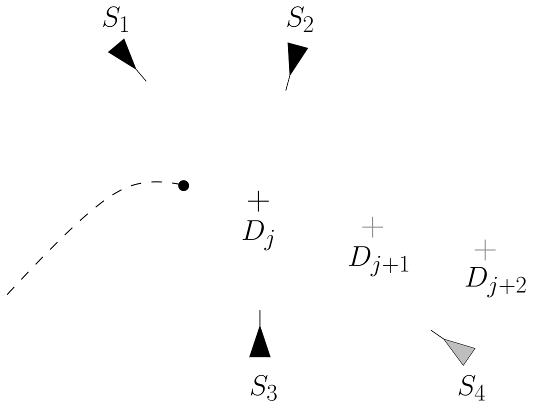


Fig. 4. Assignment and (future) positioning of S_1, S_2, and S_3 to D_j. S_4 is unassigned.

The general heuristic approach to the assignment and positioning of sensors for a demand instant can be summarized as follows:

(1) Predict[2] the object's pose, D_j, with respect to the world coordinate frame, at the demand instant, t_j.
(2) For every sensor, $S_i,\ i = 1 \ldots n$:
    (a) Determine its best achievable pose with respect to D_j, and
    (b) Assess the corresponding (single sensor) visibility of D_j, v_{s}, from the best achievable pose.
(3) Rank all S_i according to their achievable visibility, $(v_{\mathrm{s}})_i$, from highest to lowest.
(4) Assign the top k ranked sensors to t_j. (The desired pose of each assigned sensor is the best achievable pose determined in Step 2a.)

Preassignment and Prepositioning

Once an assignment has been made for the first demand instant of the rolling horizon, the proposed preassignment and prepositioning search method selects sensors for pseudo-assignment to subsequent demand instants. The objective here is to position the unassigned sensors in anticipation of future service requirements. All sensors are considered for pseudo-assignment; however, only sensors that have not been assigned by the assignment and positioning module (sub-system) may be pseudo-assigned to a future demand instant, Fig. 5. Note that, previously assigned sensors are evaluated from their (assigned) desired poses and may, in effect, be assigned to multiple future demand instants, though their future position will not be determined until the next search interval. This approach aims to service each demand with the sensors that can provide the highest quality data, rather than by those that are least utilized. Additional demand instants are considered until either all sensors have been pseudo-assigned, the maximum number of future demand instants, m (equal to the rolling horizon size), has been reached, or the search interval has expired.

The general preassignment and prepositioning procedure can be summarized as follows:

(1) Let $p = 1$
(2) Predict the object's pose, D_{j+p}, with respect to the world coordinate frame, at the demand instant, t_{j+p}.
(3) For each sensor, $S_i,\ i = 1 \ldots n$:
    (a) Determine its best achievable pose with respect to D_{j+p}, and
    (b) Assess the (single sensor) visibility of D_{j+p}, v_{s}, from the best achievable pose.
(4) Rank S_i according to best visibility, $(v_{\mathrm{s}})_i$, from highest to lowest.

[2] Herein, it is assumed that, the prediction of the demand points is performed by an external prediction subsystem.

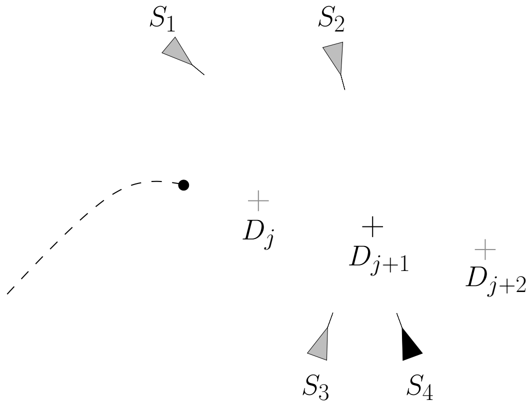


Fig. 5. Preassignment and prepositioning of S_4 to D_{j+1} and potential future poses of the remaining sensors.

(5) For the top k ranked sensors, $S_i,\ i = 1 \ldots k$, determine whether any has been assigned to an earlier demand point. Those that have not been are assigned to t_{j+p}.

(6) If $p < m - 1$ (i.e., the end of the rolling horizon has not been reached) and sensors remain unassigned, let $p = p + 1$ and return to Step 2; else, if $p = m - 1$ (i.e., the last point on the rolling horizon), assign any unassigned sensors to the final point on the rolling horizon, t_{j+m-1}.

The above approach is nearly identical to that used for assignment and positioning. The searches, however, are separated here to emphasize that different criteria may be used for assignment versus preassignment, depending on the requirements of the application and the capabilities of the system. For example, a faster algorithm may be used for preassignment, allowing more elaborate measures to be taken for the evaluation of sensors during the assignment search.

In conclusion to the above discussion of assignment and positioning and preassignment and prepositioning, one must note that the success of sensor dispatching would be dependent on the initial pose of each sensor within the workspace. This is especially true for sensors with limited dynamic capabilities with respect to the object. In general, the slower the sensors are, the more widely distributed through the workspace the sensors should be. Thus, if any part of the object trajectory is known a priori, an optimal initial configuration of the sensors can be determined. One such technique for determination of the initial surveillance-system configuration will be discussed in Section 5.

4 An Implementation Methodology

The heuristic approach to dispatching discussed in Section 3.3 is implemented in this paper using an on-line, modular surveillance-system reconfiguration-planning architecture. In this Section, after introducing a visibility measure specific to proximity sensors, the remaining subsections provide relevant details about the proposed modular architecture.

4.1 Visibility Measure for Proximity Sensing

As discussed in Section 3.1, a visibility metric can be used to evaluate the quality of information that a sensor, or a group of sensors, can provide about a demand point. The visibility measure for a single proximity sensor is considered in this paper to be:

$$v_{\mathrm{s}} = \begin{cases} \dfrac{1}{\|R\|} & \text{if the demand point is unoccluded,} \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $\|R\|$ is the Euclidean norm of the covariance matrix associated with the sensor measurement. For the proximity sensor in Fig. 2, the variance in r, θ, and ϕ may be expressed as:

$$\sigma_r^2 = \begin{cases} a + b_1 \ |r - r^*|^2 & \text{if } r < r^*, \ r \in \mathbb{R}^+, \\ a + b_2 \ |r - r^*| & \text{otherwise,} \end{cases} \tag{4}$$

$$\sigma_\theta^2 = \begin{cases} c + d \ \theta^2 & \text{if } |\theta| < \theta_{\max}, \ \theta \in \mathbb{R}, \\ \infty & \text{otherwise,} \end{cases} \tag{5}$$

$$\sigma_\phi^2 = \begin{cases} e + f \ \phi^2 & \text{if } |\phi| < \phi_{\max}, \ \phi \in \mathbb{R}, \\ \infty & \text{otherwise,} \end{cases} \tag{6}$$

where $a, b_1, b_2, c, d, e,$ and f are characteristic constants. r^* is the range between the sensor and the object at which the variance is minimal; here, the variance is equal to the constant error a. If the range is small, the variance increases proportional to b_1; if the range is large, the variance increases proportional to b_2. Similarly, for the variance in bearing and elevation, c and e are the constant measurement errors, while d and f represent the increase in variance incurred by deviations of the object position from the sensor axis. $\theta_{\max}$ and $\phi_{\max}$ limit the field of view of the sensor. Assuming that σ_r^2, σ_θ^2, and σ_ϕ^2 are uncorrelated, the covariance matrix R may be expressed in Cartesian coordinates as follows:

$$R = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 \end{bmatrix}, \tag{7}$$

where, letting $q = |r - r^*|$,

$$\sigma_x^2 = \sigma_r^2 \cos^2(\theta)\cos^2(\phi) - q^2\sigma_\theta^2 \sin^2(\theta)\cos^2(\phi) - q^2\sigma_\phi^2 \cos^2(\theta)\sin^2(\phi), \tag{7a}$$

$$\sigma_y^2 = \sigma_r^2 \sin^2(\theta)\cos^2(\phi) + q^2\sigma_\theta^2 \cos^2(\theta)\cos^2(\phi) - q^2\sigma_\phi^2 \sin^2(\theta)\sin^2(\phi), \tag{7b}$$

$$\sigma_z^2 = \sigma_r^2 \sin(\phi) + q^2\sigma_\phi^2 \cos^2(\phi), \tag{7c}$$

$$\sigma_{xy} = \sigma_{yx} = \left[\tfrac{1}{2}\sin(2\theta)\right]\left[\left[\sigma_r^2 - q^2\sigma_\theta^2\right]\cos^2(\phi) + q^2\sigma_\phi^2 \sin^2(\phi)\right], \tag{7d}$$

$$\sigma_{xz} = \sigma_{zx} = \cos(\theta)\left[\tfrac{1}{2}\sin\Big(2\phi\Big)\right]\left[\sigma_r^2 - q^2\sigma_\phi^2\right], \tag{7e}$$

$$\sigma_{yz} = \sigma_{zy} = \sin(\theta)\left[\tfrac{1}{2}\sin\Big(2\phi\Big)\right]\left[\sigma_r^2 - q^2\sigma_\phi^2\right]. \tag{7f}$$

The visibility measure for a fusion subset comprising k sensors, whose measurements are combined using sensor fusion, is defined as:

$$v_{\mathrm{f}} = \frac{1}{\|P\|}, \tag{8}$$

where P represents the fused covariance matrix,

$$P = \left[\textstyle\sum_{i=1}^{k} {R'}_i^{-1}\right]^{-1}, \tag{8a}$$

and

$$R'_i = \begin{cases} R_i & \text{if demand point is unoccluded,} \\ \emptyset & \text{otherwise.} \end{cases} \tag{8b}$$

4.2 Surveillance-System Reconfiguration-Planning Architecture

The architectural approach adopted herein is one of modularity, Fig. 6. A "blackboard" provides a controlled method for the exchange of data between the various modules, as outlined in Table 1. Information exchange is controlled by a data-management module. The rolling horizon is defined by a prediction module ,which uses observations of the object motion (provided by an observation module) to estimate the demand-point locations. Assignment and positioning and preassignment and prepositioning are carried out by their respective modules. The desired pose of each assigned sensor is reassessed by a replanning module, as the estimates of the demand-point locations become more accurate. Finally, a sensor-motion-control module commands the sensors into their assigned poses.

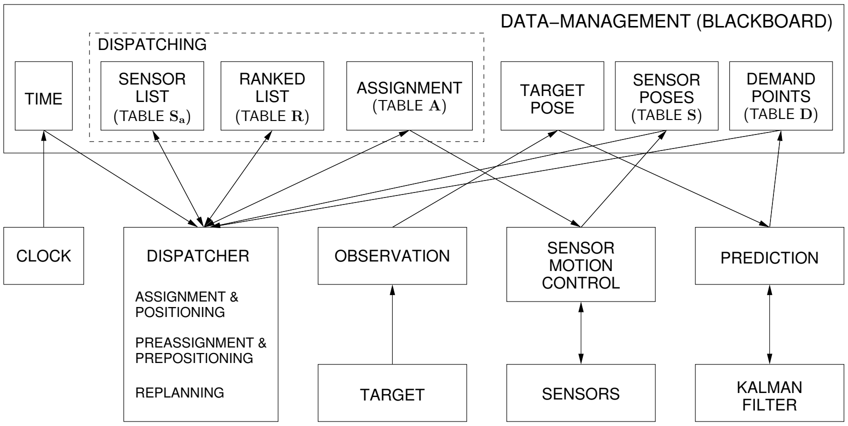

DATA–MANAGEMENT (BLACKBOARD)
DISPATCHING
TIME
SENSOR LIST (TABLE $\mathbf{S_a}$)
RANKED LIST (TABLE $\mathbf{R}$)
ASSIGNMENT (TABLE $\mathbf{A}$)
TARGET POSE
SENSOR POSES (TABLE $\mathbf{S}$)
DEMAND POINTS (TABLE $\mathbf{D}$)
CLOCK
DISPATCHER
ASSIGNMENT & POSITIONING
PREASSIGNMENT & PREPOSITIONING
REPLANNING
OBSERVATION
SENSOR MOTION CONTROL
PREDICTION
TARGET
SENSORS
KALMAN FILTER

Fig. 6. Data flow between software modules through a blackboard.

Table 1
Data exchange.

Module	Purpose	Table Interaction*				
		$\mathbf{D}$	$\mathbf{S}$	$\mathbf{S_a}$	$\mathbf{R}$	$\mathbf{A}$
Data-Management	Controls read and write access to blackboard.	R,W	R,W	R,W	R,W	R,W
Prediction	Estimates demand point locations based on target observations.	W				
Assignment and Positioning	Assigns a fusion subset (with desired poses) to the first demand instant on the rolling horizon.	R	R	R,W	R,W	W
Preassignment and Prepositioning	Pseudo-assigns sensors (with desired poses) to subsequent demand instants on the rolling horizon.	R		R,W	R,W	R,W
Replanning	Adjusts desired sensor poses as demand-point estimates improve.	R	R			R,W
Sensor-Motion-Control	Commands sensors into desired poses and reports real-time sensor pose.		W			R

Key: R indicates read access; W, indicates write access.
*See Fig. 6 for Table labels.

Data-Management Module

The surveillance system data is managed using a blackboard architecture. This common memory space ensures that multiple modules can access the current state of common system variables. The Data-Management Module provides controlled access to the blackboard, ensuring data integrity. By restricting which modules can write to different areas of the blackboard, the data cannot be inadvertently corrupted. A number of different tables, or data areas, are maintained by the Data-Management Module:

D Updated by the Prediction Module, this table maintains information about the demand instants in the rolling horizon and the predicted location of corresponding demand points.

S Maintains the current pose of each sensor, updated in real-time by the Sensor-Motion-Control Module.

$\mathbf{S_a}$ Provides a constant, "time-stamped" reference of sensor poses for assignment purposes, unaffected by adjustments made by the Sensor-Motion-Control Module. It contains both the current location of unassigned sensors and the expected (desired) pose for assigned sensors at future times.

R Used to evaluate sensors for assignment to a demand point, this table maintains a number of measures for evaluating the fitness of each sensor with respect to the demand point.

A Reflects the assignment and desired pose of each sensor.

Prediction Module

The prediction module predicts the object's future locations in the workspace (estimation of the locations of future demand instants). Namely, it determines the rolling horizon for each search interval (the time between two consecutive demand instants) and provides frequent real-time predictions of the object location for each demand instant during the search interval. These predictions are based on observations of the target pose provided by the observation module; observations may come from the very sensors that are being assigned and positioned, or from a separate set of sensors. An overhead camera is often utilized for wide-field, low-resolution estimation of object location.

Herein, a Kalman filter [23] is proposed for prediction of the object's motion. The KF is particularly suited to our application (multisensor surveillance) since each estimate has an associated error covariance matrix that provides an estimate of the uncertainty in the prediction.

Assignment and Positioning Module

The algorithm used for assignment of sensors to a demand instant (t_j) is outlined in Algorithm 1 given in Appendix A. An iterative-ranking approach is used to select sensors for assignment to the demand instant using "increasingly-complex" ranking procedures. This ensures that the best possible assignment is made, even if the system does not have the time to fully assess the capabilities and suitability of each sensor.

The sensors are first ranked using a priori information, based solely on the demand point. Next, they are ranked by computing their normalized distances (actual distance / maximum velocity of sensor) to the demand point (i.e., closer sensors are ranked higher than those further away). Finally, each sensor is ranked according to the visibility of the demand point from its sensor's best achievable pose, Section 3.3. Once all sensors have been considered, or the search time has run out, the top k ranked sensors are assigned to the demand instant at hand. The algorithm outputs the assigned subset and a desired pose for each assigned sensor.

Preassignment and Prepositioning Module

The algorithm used to select sensors for preassignment and prepositioning relative to future demand points is outlined in Algorithm 2, given in Appendix A. The algorithm is very similar to that used for assignment and prepositioning. The primary differences are that the sensors selected for assignment may have already been assigned to another (earlier) demand instant, and that the algorithm iterates through multiple demand instants ($t_{j+1} \dots t_{j+m-1}$) attempting to (pseudo) assign all sensors as discussed in Section 3.3.

Replanning Module

Replanning is initiated upon the assignment of sensors to a demand point and runs continuously during the search interval. Algorithm 3 in Appendix A outlines how the desired poses of assigned sensors may be adjusted as the estimates of the demand points improve.

The desired sensor poses are adjusted only if an improvement in object visibility can be guaranteed. This requires that the demand point at the time of assignment, D_j, lie outside of the confidence zone for the most recent estimate of the demand point location, D'_j. The confidence zone is defined by constructing an uncertainty ellipse [24] around the demand point location using the estimated error covariance matrix (from the prediction module) scaled by a user-defined variable,[3] ρ. If D_j lies within the confidence zone for D'_j, no replanning is done since adjustment of the sensor poses provides no guaranteed benefit [25]. Otherwise, D_j is replaced with D'_j and the desired poses of the corresponding sensors are determined anew, Fig. 7.

[3] Values of $\rho > 1$ increase the size of the confidence zone, decreasing the likelihood of replanning; values of $\rho < 1$ reduce the confidence zone, increasing the chance of replanning.

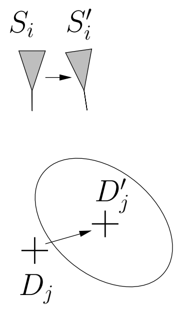


Fig. 7. Replanning for D_j. D'_j is the updated prediction of the demand point.

Sensor-Motion-Control Module

The sensor-motion-control module serves to manoeuvre each sensor into its desired pose, as read from the assignment table, **A**. This information is used in combination with knowledge of the sensor dynamics and the current pose of each sensor to plan a motion trajectory. The aim of the sensor-motion-control module is to ensure that the sensor poses are as close as possible to the desired poses at all times. Any change made to the desired pose of a sensor, either through positioning, prepositioning, or replanning, is immediately reflected in the motion plan for that sensor. The trajectory for each sensor is defined such that the error between the actual pose and the assigned pose (the best achievable pose with respect to the demand point) is minimized. For proximity sensors, this involves aligning the sensor axis with the demand point and minimizing the distance between the sensor and the demand. Thus, if the dynamic limitations of the sensor prevent the sensor-motion-control module from achieving the desired pose of the sensor, the quality of the data provided by the sensor will be as good as possible under the circumstances.

5 Initial Surveillance-System Configuration

As mentioned earlier, the initial pose of each sensor could impact surveillance quality. This is especially true in the case of slow-speed or fixed sensors. It is, therefore, desirable to search for an optimal initial pose for each sensor based on the requirements of the task at hand. This section proposes one such technique for the determination of the initial configuration for the surveillance system.

5.1 Problem Formulation and a Solution Methodology

The initial surveillance-system configuration specifies the initial poses of the set of sensors that will be available during surveillance. Determined off-line, it serves to initially distribute and position the sensors within the workspace in an optimal manner based on a priori information available for the target motion over the entire workspace. The objective in determining this initial configuration could be to maximize a visibility criterion over the entire object motion.

The overall visibility measure of the object over its "entire motion", v_c, may be stated as,

$$\max v_\mathrm{c} = w_1 \min_{i=1}^{m} (v_\mathrm{b})_i + w_2 (v_\mathrm{b})_1. \tag{9}$$

The objective function value to be maximized, v_c, in Eq. (9) above, consists of two parts: (1) chooses the minimum of all the individual best-achievable visibilities ($v_\mathrm{b} \equiv \max v_\mathrm{f}$) at the $i = 1$ to m demand instants as the representative (best achievable) overall visibility of the moving object by the surveillance system, if it were to have the initial configuration under consideration; (2) places additional emphasis on the visibility of the object at the first demand instant. w_1 and w_2 are user-chosen weighting factors which serve to balance between these two objectives, $w_1 + w_2 = 1.0$.

Eq. (9) is utilized as follows: First, a guess of the object trajectory [4] is discretized and used to define a set of demand instants. Then, for a set of initial sensor poses chosen by a search technique [5] , a comprehensive sensor dispatching simulation is conducted to survey the object at all of the demand instants. For each demand instant, sensors are dispatched and positioned to affect the best-achievable visibility as described in Section 4. The best-achievable visibility, v_b, is then determined using Eq. (2), i.e., ($\max v_\mathrm{f}$), with the assigned sensors in these "best" poses. The overall object visibility for the entire motion is subsequently determined using Eq. (9). Having determined v_c, the search technique is invoked again to vary the initial sensor poses. This process is repeated until the "best" initial configuration is found within a desired convergence value, i.e., ($\max v_\mathrm{c}$).

5.2 Example Configurations

For the 2-D workspace illustrated in Fig. 8 with four surveillance sensors, a number of different optimal initial surveillance-system configurations may be determined depending on the dynamic characteristics of the sensors and the expected object trajectory. Sensors are constrained to rails at the edge of

[4] Herein, a single object trajectory is assumed; a methodology suitable for multiple expected object trajectories is discussed in [26].

[5] The specific search technique utilized in our simulations was the flexible tolerance method [27,28].

the workspace, but are free to assume any position and orientation along the rail. Thus, each sensor has two degrees of freedom: rotation, α, and horizontal translation, x. The initial surveillance-system optimization problem, thus, maximizes Eq. (9) with eight parameters: the initial (planar) position and orientation for each of the four sensors. (The vertical position of each sensor is fixed by the rail to which it is constrained.)

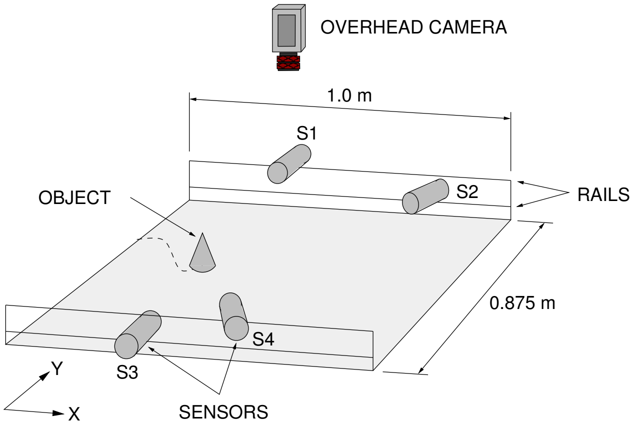
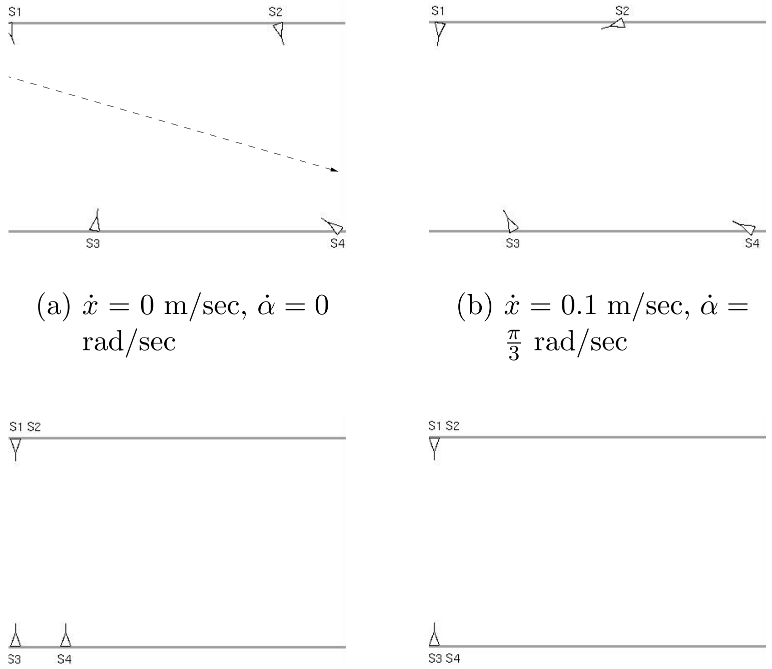


Fig. 8. Overview of example workspace configuration.

Figures 9(a)–9(d) show the outcomes of a number of different searches for which the maximum translational and rotational velocities of the sensors were varied; all of the other parameters were kept identical. In each case, the object is assumed to follow a diagonal straight line across the workspace. The weighting factors were set as: $w_1 = \frac{2}{3}$ and $w_2 = \frac{1}{3}$. v_b was evaluated using Eq. 8. Note that, Fig. 9(a) illustrates the optimal initial surveillance-system configuration for a set of (fixed) static sensors. Namely, once placed at their optimal locations, dispatching only decides on sensor assignments for the different demand instants.

6 Dispatching Simulation Examples

The examples presented in the following sections serve to illustrate different performance aspects of the proposed approach to dispatching. In particular, surveillance-system performance is evaluated under changing object trajectories and sensor dynamics. For ease of illustration, a 2-D workspace is assumed. For all examples, the object moves at approximately 0.2 m/sec. Demand points are predicted on the basis of lower-resolution images acquired from an overhead camera, Fig. 8. These observations are corrupted by Gaussian noise with σ = 0.02 m. In each simulation, fusion subsets of size 3 were used with a rolling horizon size of 3 demand instants separated by 0.6 seconds. The sensor parameters were set as follows: $a = 2.5e^{-5}$ m, $b_1 = 1.25e^{-3}$, $b_2 = 6.25e^{-5}$, r^* = 0.05 m, $c = 8e^{-5}$ rad, $d = 50$, and $\theta_\text{max} = \frac{\pi}{4}$ rad.



(a) $\dot{x} = 0$ m/sec, $\dot{\alpha} = 0$ rad/sec

(b) $\dot{x} = 0.1$ m/sec, $\dot{\alpha} = \frac{\pi}{3}$ rad/sec

(c) $\dot{x} = 0.2$ m/sec, $\dot{\alpha} = \frac{\pi}{2}$ rad/sec

(d) $\dot{x} = 2.5$ m/sec, $\dot{\alpha} = 2\pi$ rad/sec

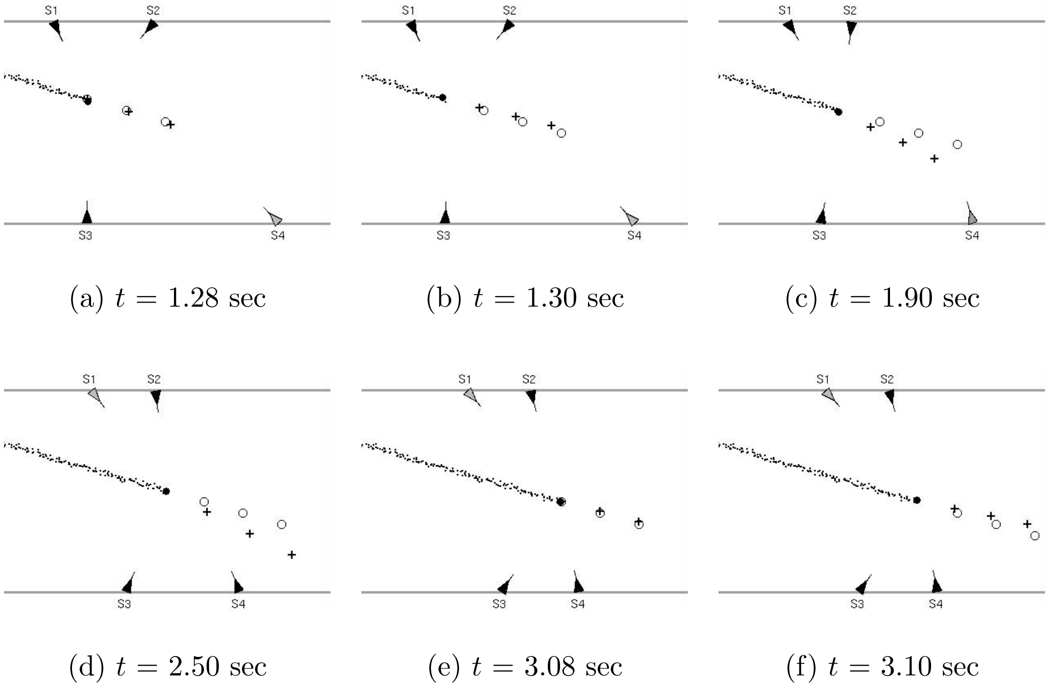
Fig. 9. Optimal initial surveillance-system configuration for sensors with varying maximum velocities.

6.1 Dispatching

The following simulation example illustrates how a surveillance system is reconfigured in real-time using the dispatching approach.

Example 1

Fig. 10 shows snapshots of a sample run for a noise-corrupted straight-line object trajectory. Here, the maximum translational velocity of each sensor is $\dot{x} = 0.1$ m/sec; the maximum rotational velocity is $\dot{\alpha} = \frac{\pi}{3}$ rad/sec. The system starts from an initial configuration optimized for these sensors and this object trajectory, Fig. 9(b). (In each figure, the predicted demand point locations are indicated by a cross (+). The actual demand point locations, if the system were capable of perfect prediction, are indicated by circles (○). Assigned sensors appear as black and preassigned sensors as grey.) The details of data acquisition and sensor assignment are outlined in Table 2.



(a) $t = 1.28$ sec (b) $t = 1.30$ sec (c) $t = 1.90$ sec

(d) $t = 2.50$ sec (e) $t = 3.08$ sec (f) $t = 3.10$ sec

Fig. 10. Straight-line trajectory.

Table 2
Assignments for straight-line trajectory.

		Assignments				
t	Action	S_1	S_2	S_3	S_4	v_{b}
1.28	Sensors in motion	$\mathbf{t_3}$	$\mathbf{t_3}$	$\mathbf{t_3}$	t_5	—
1.30	Acquire t_3 using 1-2-3; assign 1-2-3 to t_4	$\mathbf{t_4}$	$\mathbf{t_4}$	$\mathbf{t_4}$	t_6	0.8684
1.90	Acquire t_4 using 1-2-3; assign 2-3-4 to t_5	$\mathbf{t_5}$	$\mathbf{t_5}$	$\mathbf{t_5}$	t_6	0.8689
2.50	Acquire t_5 using 2-3-4; assign 2-3-4 to t_6	t_8	$\mathbf{t_6}$	$\mathbf{t_6}$	$\mathbf{t_6}$	0.7808
3.08	Sensors in motion	t_8	$\mathbf{t_6}$	$\mathbf{t_6}$	$\mathbf{t_6}$	—
3.10	Acquire t_6 using 2-3-4; assign 2-3-4 to t_7	t_9	$\mathbf{t_7}$	$\mathbf{t_7}$	$\mathbf{t_7}$	0.8532

Key: $\mathbf{t_1}$ indicates assigned; t_1, preassigned.

6.2 Sensitivity to Dynamic Characteristics of Sensors

Example 1 (Fig. 10) demonstrated how reconfiguration performs for a surveillance system with modest dynamic capabilities. Here, the effect of changing the dynamic characteristics of the surveillance system on the performance of dispatching is investigated.

Fig. 11 presents the overall visibilities for four different surveillance systems surveying an object that is following the straight-line trajectory shown in Fig. 9(a). The performance of these systems range from very fast (at least an order of magnitude faster than the object) to a static system in which the sensors have no dynamic capability at all. (Note that, while the static sensors cannot move, dispatching is still utilized to determine the appropriate subsets of sensors (the coordination strategy) in real-time.) For each system, its own optimal initial surveillance-system configuration was determined as described in Section 5.

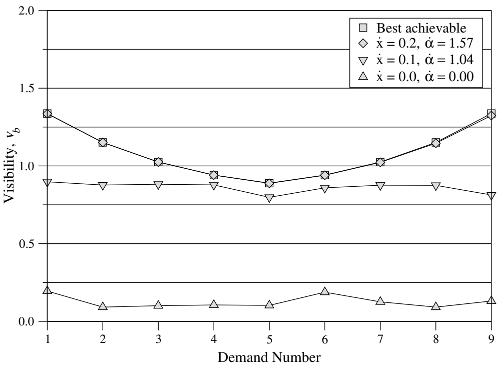


Fig. 11. Observed visibilities of a straight-line trajectory for different surveillance systems.

From Fig. 11, as expected, it can be seen that dynamic surveillance systems outperform the static surveillance system. The variation in visibility for the fast surveillance systems is an artifact of the workspace constraints placed on the sensors. By constraining the sensors to rails, they cannot maintain a constant range from the object. Thus, as the object approaches the centre of the workspace, the visibility drops due to increased range. The upper curve in Fig. 11 represents the best achievable visibility under these conditions; here, for each demand point, the assigned sensors match the x-position of the object and align their axes directly with the object ($\theta = 0$).

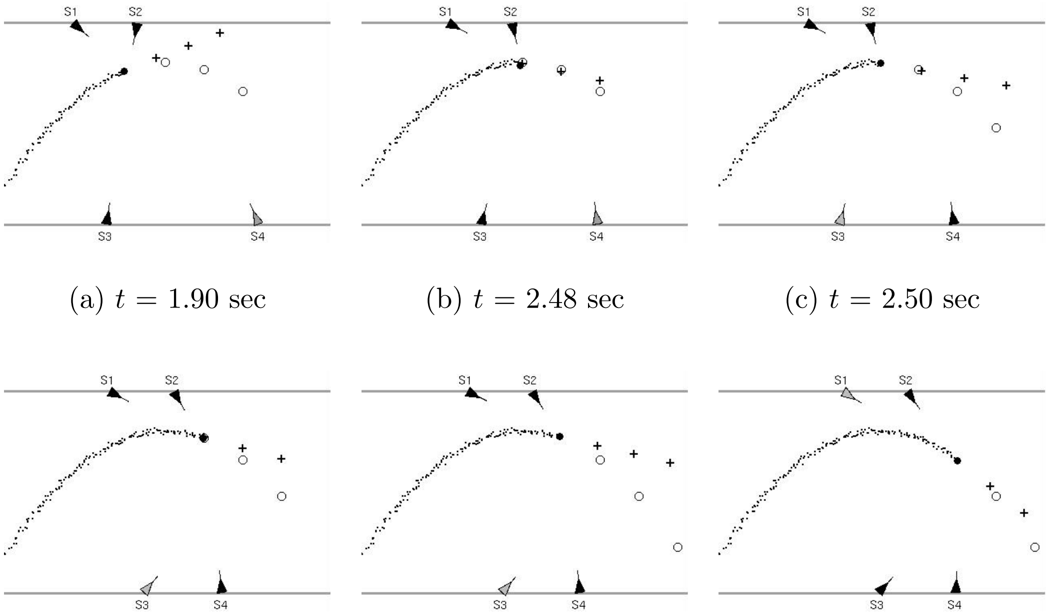
A comparison between the "best" system and the other systems indicates that increasing the speed of the surveillance system beyond an upper limit is not particularly valuable (e.g., the achieved visibilities for $\dot{x} = 2.5$ m/sec, $\dot{\alpha} = 2\pi$ rad/sec (best achievable) and $\dot{x} = 0.2$ m/sec, $\dot{\alpha} = \frac{\pi}{2}$ rad/sec are practically the same). However, it is clear that providing a surveillance system with even limited dynamic capabilities (e.g., $\dot{x} = 0.1$ m/sec and $\dot{\alpha} = \frac{\pi}{3}$ rad/sec) may significantly improve its effectiveness.

6.3 Robustness to Trajectory Variation

This section presents a simulation example that demonstrates how dispatching can be used to adapt a surveillance system to an object trajectory different from the one for which the initial configuration was optimized.

Example 2

For this example, the initial sensor poses were optimized for the expected straight-line trajectory shown in Fig.9 (i.e., the initial configuration is identical to that used for Example 1). Fig. 12 shows snapshots of a sample run for a parabolic object trajectory; the corresponding data is provided in Table 3.



(a) t = 1.90 sec (b) t = 2.48 sec (c) t = 2.50 sec

(d) t = 3.08 sec (e) t = 3.10 sec (f) t = 3.70 sec

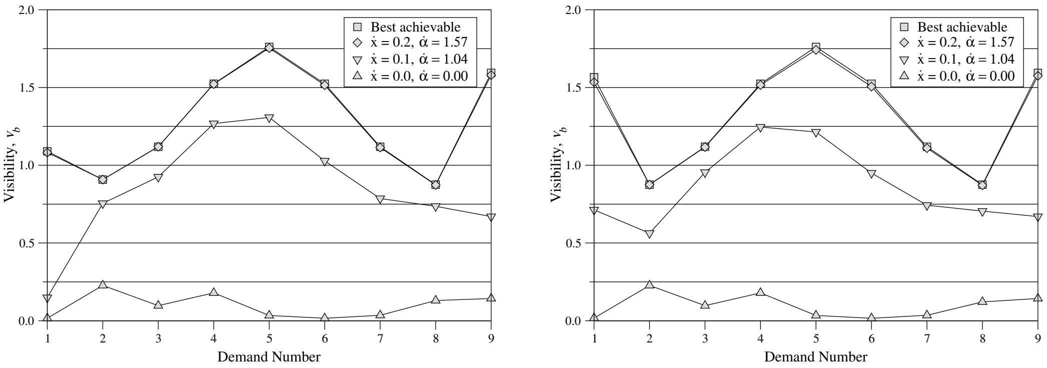
Fig. 12. Parabolic trajectory.

Table 3
Assignments for parabolic trajectory.

		Assignments				
t	Action	S_1	S_2	S_3	S_4	v_{b}
1.90	Acquire t_4 using 1-2-3; assign 1-2-3 to t_5	$\mathbf{t_5}$	$\mathbf{t_5}$	$\mathbf{t_5}$	t_6	1.2324
2.48	Sensors in motion	$\mathbf{t_5}$	$\mathbf{t_5}$	$\mathbf{t_5}$	t_6	—
2.50	Acquire t_5 using 1-2-3; assign 1-2-4 to t_6	$\mathbf{t_6}$	$\mathbf{t_6}$	t_8	$\mathbf{t_6}$	1.2686
3.08	Sensors in motion	$\mathbf{t_6}$	$\mathbf{t_6}$	t_8	$\mathbf{t_6}$	—
3.10	Acquire t_6 using 1-2-4; assign 1-2-4 to t_7	$\mathbf{t_7}$	$\mathbf{t_7}$	t_9	$\mathbf{t_7}$	1.0540
3.70	Acquire t_7 using 1-2-4; assign 2-3-4 to t_8	t_{10}	$\mathbf{t_8}$	$\mathbf{t_8}$	$\mathbf{t_8}$	0.7863

Key: $\mathbf{t_1}$ indicates assigned; t_1, preassigned.

Fig. 13(a) presents the overall visibilities of the surveillance system for different sensor selections, with dynamic capabilities ranging from static to very fast. The initial poses of the sensors for each surveillance system were determined as described in Section 5 (with the expectation of a straight-line trajectory).



(a) Parabolic

(b) Parabolic with initial reaction

Fig. 13. Observed visibilities for different surveillance systems expecting a straight-line trajectory.

As expected, the faster dynamic system ($\dot{x} = 0.2$ m/sec, $\dot{\alpha} = \frac{\pi}{2}$ rad/sec) performs almost perfectly, the slower dynamic system ($\dot{x} = 0.1$ m/sec, $\dot{\alpha} = \frac{\pi}{3}$ rad/sec) fairs somewhat worse, and both dynamic systems outperform the static system. The reduction in visibility for the first demand is a result of the difference between the expected and actual trajectories. The initial-configuration planning module optimally placed the sensors on the top rail close to the left edge of the workspace and the sensors on the bottom rail close to the right edge of the workspace—an appropriate strategy for an object trajectory from the upper-left to the lower-right, but not ideal for the parabolic trajectory that was observed. When the object entered from the lower-left, the sensors did not have an opportunity to react (sensing of the first demand point is almost instantaneous and determined through the initial configuration, not dispatching). As a result, the sensors were not in an optimal pose, nor was the initial fusion subset appropriate.

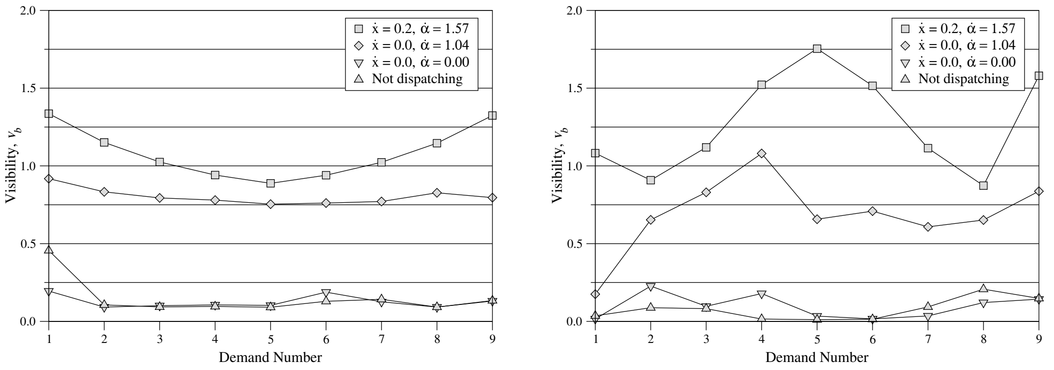
Fig. 13(b) illustrates how the performance of each system would be altered, if they were provided with prediction information prior to surveillance of the first demand point. In this case, observations of the object (using the overhead camera) began outside of the workspace. These observations allowed for the a priori initialization of the Prediction Module, which in turn provided the dispatching and positioning modules with better estimates of the first demand point and allowed them sufficient time to optimally position the sensors for best visibility. For all of the dynamic systems, the performance is improved.

Several observations may be made from this example: First, the simulations confirm that the surveillance system can still provide valuable information, even when the actual object trajectory deviates significantly from the expected object trajectory. Second, adaptation of the surveillance system to the trajectory is a function of the dynamic capabilities of the sensors. In other words, while the surveillance system's performance may degrade as the actual trajectory deviates from expectation, the degradation is more marked for slower sensors. (It is important to note that even static systems still provide surveillance information, just at a lower visibility). This would indicate that the slower the system, the more important a reasonable initial guess of the object trajectory becomes.

6.4 Dispatching vs. Non-Dispatching Systems

Fig. 14 compares a number of systems using the dispatching methodology presented in this paper with a system that does not use dispatching at all. The dispatching systems select subsets of three sensors from the four available for use in a sensor fusion process; the non-dispatching system simply fuses the measurements from all four sensors. In other words, the non-dispatching approach may be stated as: "all the sensors, all the time".

In the comparative runs considered in Fig. 14(a), each system started with an initial configuration that was optimized for a straight-line object trajectory and observed the same trajectory "with noise". In Fig. 14(b), each surveillance system was configured under the assumption that the object would follow a straight-line trajectory, while the actual observed trajectory was parabolic.



(a) Straight-line

(b) Parabolic

Fig. 14. Observed visibilities for dispatching vs. non-dispatching surveillance systems.

From these Figures, it is clear that dispatching systems outperform non-dispatching systems, provided that they have at least limited dynamic capabilities. This is best exemplified by the dynamic system having only rotational capability ($\dot{x} = 0$ m/sec, $\dot{\alpha} = \frac{\pi}{3}$ rad/sec) that still significantly outperforms the non-dispatching system. One should not conclude from this that $\dot{\alpha}$ capability is more important than $\dot{x}$; in fact, if the sensors can move as fast as the object along the rails, rotational ability will not improve the object visibility at all.

However, for static systems, the advantage of the dispatching approach is not apparent. From a performance perspective, the user of a static surveillance system may be better off to simply use all of the sensors at once. The use of dispatching for a static system may only make sense if the costs associated with processing the data from all of sensors at once compromises the real-time performance of the system (e.g., high-resolution image processing). In this case, dispatching provides an effective mechanism to select an appropriate subset for processing.

7 Conclusions

A method for maximizing the effectiveness of moving-object surveillance using multiple sensors is presented in this paper. The overall goal of the method is to position sensors in response to changing task-space demands. This is shown to be possible using a two-part dispatching strategy, comprising coordination and positioning strategies. The motion of each sensor is evaluated based on the quality of information that each can provide for specified object locations. From this, a group of sensors (for use in a sensor fusion context) may then be assigned to a particular demand point. In addition, the sensors that are not required for the most imminent surveillance task are assigned to future predicted demands and manoeuvred in a controlled manner, rather than remaining idle or moving randomly.

The surveillance-system configuration is adjusted according to a continual reevaluation of the capabilities of each sensor and the sensing requirements. This is a reactive procedure, executed on-line; therefore, no absolute condition of optimality is imposed. In fact, the ability of the proposed dispatching approach to effectively adjust the sensor poses is very dependent on the number of sensors used, the manoeuvrability of the sensors, and their initial poses. Variations in each of these parameters, in addition to the accuracy of the object motion prediction, would affect the overall performance of the system. In this context, the use of a dispatching methodology, combined with dynamic sensors and sensor fusion, has been shown to provide a considerable benefit over (fixed) static-sensor surveillance systems. The increased accuracy and robustness of a dynamic system makes it suitable for tasks, such as the robotic interception of manoeuvring objects, that require high-quality, real-time sensor information.

Appendix A

This appendix provides a detailed description of the algorithms used for assignment and positioning, preassignment and prepositioning, and replanning, introduced Section 4. Note that, t_{cur} in each algorithm refers to the current time provided by a real-time clock; and, s_{f} is a factor of safety used to ensure that there is sufficient time to carry out assignment and, once an assignment has been made, to manoeuvre the sensors into their assigned poses. m is the maximum number of demand instants that will be considered by Algorithms 2 and 3.

Algorithm 1 Assignment and Positioning

```
 Initialize Table $\mathbf{S_a}$ with the current sensor poses (from Table $\mathbf{S}$) and the current time, $t_{\text{cur}}$.
First level of refinement:
 Initialize Table $\mathbf{R}$ with a priori desired poses for each sensor based on the current demand instant, $t_j$.
 Clear the Assignment Table, $\mathbf{A}$.
 if no time remains in interval (i.e., $t_j < t_{\text{cur}} + s_{\text{f}}$) then
   Goto OUTPUT
 end if
Second level of refinement:
 for $i = 1$ to $n$ (number of sensors) do
   For row (sensor) $i$ of Table $\mathbf{R}$, calculate the normalized distance between the demand point $D_j$ and the sensor (whose pose is specified in Table $\mathbf{S_a}$). Place the result in Column $d$ of Table $\mathbf{R}$.
   Sort Table $\mathbf{R}$ by distance (Column $d$).
   if no time remains in interval (i.e., $t_j < t_{\text{cur}} + s_{\text{f}}$) then
     Goto OUTPUT
   end if
 end for
Third level of refinement:
 for $i = 1$ to $n$ (number of sensors) do
   Determine the best achievable pose for the $i$th sensor (row) from Table $\mathbf{R}$ (whose pose is specified in Table $\mathbf{S_a}$) with respect to $D_j$ (pose specified in Table $\mathbf{D}$) in the time remaining in the interval ($t_j - t_{\text{cur}} - s_{\text{f}}$). Place the result in $P_{\text{d}}$ of Table $\mathbf{R}$.
   Determine the required motion time to move the sensor from the current pose (in Table $\mathbf{S_a}$) to the desired pose ($P_{\text{d}}$ in Table $\mathbf{R}$). Store the result in Column $t_{\text{motion}}$ of Table $\mathbf{R}$.
   Calculate the visibility of $D_j$ from best achievable pose, $P_{\text{d}}$, using Eq. (3). Place the result in Column $v_{\text{s}}$ of Table $\mathbf{R}$.
   Sort Table $\mathbf{R}$ by visibility (Column $v_{\text{s}}$).
   if the current index is greater than the fusion subset size (i.e., $i \geq k$) then
     Determine the maximum motion time from the first $k$ sensors (rows) in Table $\mathbf{R}$ (i.e., max of Column $t_{\text{motion}}$).
     if there is not enough time to manoeuvre the sensors (i.e., $t_j < t_{\text{cur}} + t_{\max} + s_{\text{f}}$) then
       Goto OUTPUT
     end if
   end if
 end for
OUTPUT:
 for $i = 1$ to $k$ (fusion subset size) do
   Assign, in Table $\mathbf{A}$, the $k$th ranked sensor in Table $\mathbf{R}$ to $D_j$, along with its desired pose, $P_{\text{d}}$.
   Update Table $\mathbf{S_a}$ with the desired pose of the assigned sensor. Place the time of $t_j$ in Column $t$ of Table $\mathbf{S_a}$.
 end for
```

Algorithm 2 Preassignment and Prepositioning

```
Let $p = 1$.
while $p < m$ (size of rolling horizon) and sensors remain unassigned do
 *First level of refinement:*
  Initialize Table $\mathbf{R}$ with a priori desired poses for each sensor based on the current demand
  instant, $t_{j+p}$.
  if no time remains in interval (i.e., $t_j < t_{\mathrm{cur}} + s_{\mathrm{f}}$) then
    Goto OUTPUT
  end if
 *Second level of refinement:*
  for $i = 1$ to $n$ (number of sensors) do
    For row (sensor) $i$ of Table $\mathbf{R}$, calculate the normalized distance between the demand point
    $D_{j+p}$ and the sensor (whose pose is specified in Table $\mathbf{S_a}$). Place result in Table $\mathbf{R}$.
    Sort Table $\mathbf{R}$ by distance.
    if no time remains in interval (i.e., $t_j < t_{\mathrm{cur}} + s_{\mathrm{f}}$) then
      Goto OUTPUT
    end if
  end for
 *Third level of refinement:*
  for $i = 1$ to $n$ (number of sensors) do
    if the $i$th sensor has already been assigned to a demand instant then
      Determine the best achievable pose for the $i$th sensor (row) from Table $\mathbf{R}$ with respect
      to $D_{j+p}$ in the time available for motion ($t_{j+p} - s_{\mathrm{f}} -$ time of demand instant sensor $i$ is
      assigned to.) Place the result in $P_{\mathrm{d}}$ of Table $\mathbf{R}$.
    else
      Determine the best achievable pose for the $i$th sensor (row) from Table $\mathbf{R}$ with respect to
      $D_{j+p}$ in the time available for motion ($t_{j+p} - t_{\mathrm{cur}} - s_{\mathrm{f}}$). Place the result in $P_{\mathrm{d}}$ of Table $\mathbf{R}$.
    end if
    Calculate the visibility of $D_{j+p}$ from best achievable pose, $P_{\mathrm{d}}$, using Eq. (3). Place result in
    Table $\mathbf{R}$.
    Sort Table $\mathbf{R}$ by visibility.
    if no time remains in interval (i.e., $t_j < t_{\mathrm{cur}} + s_{\mathrm{f}}$) then
      Goto OUTPUT
    end if
  end for
 OUTPUT:
  if $p < m - 1$ then
    for $i = 1$ to $k$ (fusion subset size) do
      if Sensor in row $i$ of Table $\mathbf{R}$ has not been assigned to a prior demand instant then
        Assign, in Table $\mathbf{A}$, the $i$th ranked sensor in Table $\mathbf{R}$ to $D_{j+p}$, along with its desired
        pose, $P_{\mathrm{d}}$.
        Update Table $\mathbf{S_a}$ with the desired pose of the assigned sensor. Place the time of $t_{j+p}$ in
        Table $\mathbf{S_a}$.
      end if
    end for
  else if $p = m - 1$ (last point on rolling horizon) then
    for $i = 1$ to $n$ (number of sensors) do
      if Sensor in row $i$ of Table $\mathbf{R}$ has not been assigned to a prior demand instant then
        Assign, in Table $\mathbf{A}$, the $i$th ranked sensor in Table $\mathbf{R}$ to $D_{j+p}$, along with its desired
        pose, $P_{\mathrm{d}}$.
        Update Table $\mathbf{S_a}$ with the desired pose of the assigned sensor. Place the time of $t_{j+p}$ in
        Table $\mathbf{S_a}$.
      end if
    end for
  end if
  Let $p = p + 1$
end while
```

Algorithm 3 Replanning

```
while the demand instant $t_j < t_{\mathrm{cur}} + s_{\mathrm{f}}$ do
  Iterate through demand points:
  for $p = 0$ to $m - 1$ do
    Iterate through sensors:
    for $i = 1$ to $n$ do
      if the $i$th sensor in Table A is assigned to demand point $D_{j+p}$ then
        Get current estimate of $D_{j+p}$ from Table D.
        Compute the position errors $\delta_x$, $\delta_y$, and $\delta_z$ as the difference between position of $D_{j+p}$
        at the time of assignment (in stored in Table A) and the current position of $D_{j+p}$ (in
        Table D).
        Compare the position error with the prediction uncertainties for current estimate of $D_{j+p}$
        (scaled by $\rho$):
        if $\delta_x > \rho\sqrt{\sigma_x^2} \vee \delta_y > \rho\sqrt{\sigma_y^2} \vee \delta_z > \rho\sqrt{\sigma_z^2}$ then
          Update the location of $D_{j+p}$ for sensor $i$ in Table A.
          Update the desired pose, $P_{\mathrm{d}}$, for the $i$th sensor in Table A with respect to the updated
          $D_{j+p}$.
        end if
      end if
    end for
  end for
end while
```

References

[1] Croft EA, Fenton RG, Benhabib B. An on-line robot planning strategy for target interception. J Robotic Systems 1998;15(2):97–114.

[2] Mehrandezh M, Sela MN, Fenton RG, Benhabib B. Proportional navigation guidance for robotic interception of moving objects. J Robotic Systems 2000;17(6):321–340.

[3] Luo RC, Kay MG. Multisensor integration and fusion for intelligent machines and systems. In: Abidi MA, Gonzalez RC, editors. Data fusion in robotics and machine intelligence. San Diego: Academic Press, 1992.

[4] Viswanadham N, Narahari Y. Performance modeling of automated manufacturing systems. Englewood Cliffs NJ: Prentice Hall, 1992.

[5] Shrivastava M, Chande PK, Monga AS. Taxi dispatch: A fuzzy rule approach. IEEE Proceedings of the Conference on Intelligent Transportation Systems, 1997. p. 978–982.

[6] Potvin J-Y, Shen Y. A neural network approach to the vehicle dispatching problem. IEEE Proceedings of the International Joint Conference on Neural Networks, vol. 2, 1991. p. 1230–1235.

[7] Benyahia I, Potvin J-Y. Decision support for vehicle dispatching using genetic programming. IEEE Trans Systems Man Cybernet–Part A 1998;28(3):306–314.

[8] Psaraftis HN. Dynamic vehicle routing problems. In: Golden BL, Assad AA, editors. Vehicle routing: methods and studies. Amsterdam: North-Holland, 1988.

[9] Powell WB. A comparative review of alternative algorithms for the dynamic vehicle allocation problem. In: Golden BL, Assad AA, editors. Vehicle routing: methods and studies. Amsterdam: North-Holland, 1988.

[10] Ran B, Boyce D. Modeling dynamic transportation networks: an intelligent transportation system oriented approach, 2nd ed. Berlin:Springer, 1996.

[11] Trucco E, Umasuthan M, Wallace AM, Roberto V. Model-based planning of optimal sensor placements for inspection. IEEE Trans Robotics Automation 1997;13(2):182–194.

[12] Zhang H. Two-dimensional optimal sensor placement. IEEE Trans Systems Man Cybernet 1995;25(5):781–792.

[13] Niepold R, Sakane S, Shirai Y. Vision sensor set-up planning for a hand-eye system using environmental models. Proceedings of the Society of Instrument and Control Engineers of Japan, Hiroshima, Japan, 1987. p. 1037–1040.

[14] Abrams S, Allen PK, Tarabanis K. Computing camera viewpoints in an active robot work cell. Int J Robotics Research 1999;18(3):267–285.

[15] Matsuyama T, Wada T, Tokai S. Active image capturing and dynamic scene visualization by cooperative distributed vision. In: Nishio S, Kishino F, editors. Advanced multimedia content processing. Berlin: Springer-Verlag, 1999.

[16] Horling B, Vincent R, Mailler R, Shen J, Becker R, Rawlins K, Lesser V. SPT: distributed sensor network for real time tracking, Technical Report 00-49, University of Massachusetts, Amherst MA, 2000.

[17] Saad RE, Bonen A, Smith KC, Benhabib B. Proximity sensing for robotics. In: Webster JG, editor. The measurement, instrumentation, and sensors handbook. Boca Raton FL: CRC Press, IEEE Press, 1999.

[18] Ruocco SR. Robot Sensors and Transducers. Milton Keynes, England:Open University Press, 1987.

[19] Hall DL. Mathematical Techniques in Multisensor Data Fusion. Boston: Artech House, 1992.

[20] Nakamura Y, Xu Y. Geometrical fusion method for multisensor robotic systems. In: Luo RC, Kay MG, editors. Multisensor integration and fusion for intelligent machines and systems. Norwood NJ: Ablex Publishing Corporation, 1995.

[21] Bar-Shalom Y, Li X-R. Estimation and tracking: principles, techniques, and software. Boston: Artech House, 1993.

[22] Broida TJ. Kinematic and statistical models for data fusion using Kalman filtering. In: Abidi MA, Gonzalez RC, editors. Data fusion in robotics and machine intelligence. San Diego: Academic Press, 1992.

[23] He D, Hujic D, Mills JK, Benhabib B. Moving-object recognition using premarking and active vision. IEEE Proceedings of the International Conference on Robotics and Automation, vol. 3, 1996. p. 1980–1985.

[24] Kosaha A, Kak AC. Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties. CVGIP: Image Understanding 1992;56(3):271–329.

[25] Erdmann M. Randomization in robot tasks. Int J Robotics Research 1992;11(5):399–436.

[26] Naish MD, Croft EA, Benhabib B. Simulation-based sensing-system configuration for dynamic dispatching. IEEE Proceedings of the International Conference on Systems, Man, and Cybernetics, vol. 5, 2001. p. 2964–2969.

[27] Paviani DA, Himmelblau DM. Constrained nonlinear optimization by heuristic programming. Operations Research 1969;17:872–882.

[28] Himmelblau DM. Applied nonlinear programming, New York: McGraw-Hill, 1972.