INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 11, NO. 2, JUNE 2006, 82-96

A Reconfigurable Multi-Camera Active-Vision System for Object Recognition via Shape-from-Shading

Ardevan BAKHTARI and Beno BENHABIB

ABSTRACT – **The active surveillance of manoeuvring targets with multiple dynamic cameras requires an effective planning methodology to dynamically select and position the groups of cameras for optimal performance. This paper presents such a methodology for the real-time reconfiguration of a multi-camera active-vision system for the recognition of dynamic objects in the presence of multiple static and/or mobile obstacles. An agent-based *coordination strategy* determines how many and, specifically, which cameras should be used at each data-acquisition instant in order to optimize the performance of the surveillance system. A *positioning strategy* determines the optimal location of each chosen camera at all data-acquisition instants.**

The proposed sensing-system reconfiguration methodology has been implemented on an experimental prototype set-up for automated object recognition via *Shape-from-Shading* (SFS). In contrast to previously proposed algorithms, the recognition method presented in this paper does not require images to be taken at constant viewing angles. Furthermore, our algorithm fuses data from several object images acquired from varying viewpoints and with different lighting conditions. Our simulations and experiments have shown that multi-camera active sensing and data fusion tangibly increase the accuracy and robustness of a recognition process.

Index Terms: *Active-vision, surveillance, sensor fusion, recognition, shape-from-shading.*

1. INTRODUCTION

In dynamic multi-object environments, an active-sensing system may provide autonomous surveillance of an Object-of-Interest (OoI) as it moves through the workspace. The environment may be cluttered with static and/or mobile objects that are not of interest (i.e., obstacles), which may prevent the viewing of the target for some periods of time. Surveillance is defined here as the data-acquisition and analysis process for the recognition and/or parameter estimation of targets (i.e., features) on OoIs.

Manuscript received November 11, 2006; revised May 31, 2006.

Computer Integrated Manufacturing Laboratory, Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Rd., Toronto, Ontario, Canada, M5S 3G8, bakhtar@mie.utoronto.ca and beno@mie.utoronto.ca.

This work has been supported by the Natural Sciences and Engineering Research Council of Canada.

On-line planning can be used to dynamically reconfigure an active sensing-system based on the current and estimated future states of the environment to improve the performance of the surveillance system [1]. Sensing-system planning requires the dynamic selection of (*a*) the number of sensors to be utilized for data acquisition, and (*b*) their optimal position and orientation (pose).

1.1 Sensor Planning in Static Environments

Traditionally, sensor planning has been utilized for determining the configuration of a set of sensors in static surveillance environments. Sensor planning in a static environment has been categorized as either "generate-and-test" or "synthesis" [1]. In generate-and-test methods, sensor placement plans are determined based on the task constraints by searching through discretized sensor configurations. An example of such a sensor planner was presented in [2], where a single robot moves a sensor to observe features on a stationary OoI. A virtual sphere, created around the OoI, represents all the possible poses (positions and orientations) for the sensor. The sphere is discretized and poses that are unoccluded and fit within the workspace of the robot are selected. Similarly, in [3], the sensing planner tries to find the minimum number of viewpoints that would allow observation of all the features on an OoI. In order to accomplish this objective, not only the virtual sphere around the OoI, but also the surface of the OoI itself is discretized.

Synthesis methods determine sensor configurations by using the analytical relationship between task requirements and the sensor parameters. The requirement for an analytical formulation of the sensing task makes the system highly application specific. For example, in [4], the sensor planner synthesizes a region of viewpoints by first imposing a 3-D bound on the position of the camera by each of the task constraints. The intersections of these bounds are considered to be regions of acceptable viewpoints. Similarly, in [5], the proposed system automatically determines the viewing direction that allows the entire OoI to become visible while minimizing distortion in the image. The system works by taking points along the outer edge of the OoI and creating uncertainties in sensor observations

by data fusion and optimal sensor placement: an example is given in [6], where optimal 2-D sensor placements are determined for a number of similar sensors. The system in [7] uses an off-line, generate-and-test method with an on-line synthesis method to optimally place dissimilar sensors (range, intensity, and stereo cameras) for OoI inspection.

1.2 Sensor Planning in Dynamic Environments

1.2.1 Single-Object Environments

Recently, there has been greater interest in sensor planning in dynamic environments, e.g., [8-10]. Most current systems address this problem by utilizing methods developed for sensor planning in static environments. For example, the system proposed in [9] optimizes sensor configurations off-line by discretizing time and treating each time instant as a static case to be able to utilize the sensor-planning method presented in [2]. This off-line approach requires the motion of the OoI to be known *a priori* with enough accuracy to make the sensor planning process successful. The system presented in [10] uses an off-line heuristics method to determine sensor motions in 2-D based on an *a priori* known OoI trajectory and an on-line controller to readjust sensor motions to account for deviations in actual OoI trajectory from the expected.

In contrast to the abovementioned, the system proposed in [11] does not require *a priori* knowledge of the OoI's trajectory to accomplish the sensor-planning task. The system discretizes the workspace into a number of sectors and, once the OoI enters a sector, the sensors assigned to the sector provide synchronous information about the OoI. The system in [12] utilizes multiple sensors through an agent-based sensing method, where each mobile sensor's path is independently determined through a triangulation method to avoid obstacles. The system in [13] also uses autonomous agents; however, unlike [13], the agents negotiate to achieve the necessary level of coordination for accomplishing the given sensing task, while maximizing the amount of the target that can be observed at any given time. The system in [14] combines sensor-placement constraints and the shape and current pose of the OoI via a Bayesian network for task-specific sensor planning.

1.2.2 Multi-Object Environments

Multi-object surveillance environments have been classified into two categories: (1) single target and (2) multi-target. In a single-target (but, multi-object) environment the system must perform sensor planning not only based on the trajectory of the OoI (target) but also other objects that are *not of interest* but may act as occlusions. Examples of systems capable of sensor planning in multi-object environments were presented in [15] and [16]. Both require the OoI and the surrounding environments to be modelled as 3D polyhedrons so that constraints such as occlusions can be determined at each discretized time instants. Numerical optimization methods are used to determine sensor locations and optical settings that eliminate occlusions at each time instant. The methods presented in [17] and [23] use pre-determined constraints, such as occlusions, field of view, and travel limits, to dynamically plan the motion of the single (robot-mounted) camera. In a multi-target (multi-object) environment, sensor planning aims to maximize the number of targets that are observed while minimizing the uncertainty associated with the observations.

It should be noted that the proposed systems described above mostly rely on off-line modeling of the environment and assume knowledge of the trajectory of the target. In contrast, the sensor-planning method proposed in this paper is capable of on-line sensing-system reconfiguration without *a priori* knowledge of the OoI trajectory or a complete model of the surroundings.

1.3 Agent-Based Sensor Planning

Recently, a number of agent-based approaches have been proposed to the problem of real-time sensing-system reconfiguration in order to decrease complexity and increase robustness and scalability. For example, in [20], the proposed system uses a collection of sensor agents to track multiple moving targets. An agent is considered to be a Pan-Tilt-Zoom (PTZ) camera plus a dedicated computer for camera control and image processing. The agents scan the workspace for a target and once one is detected, they share the OoI information. Each agent independently determines whether it should contribute to the surveillance of this target or search for a new target.

In [21] and [22], multiple mobile sensors, modeled as separate agents, are used to detect and recognize targets. This system, in contrast to the one presented in [20], utilizes purely cooperative agents[1]. The system performance is significantly improved; however, complexity of the required conflict management strategy is also significantly increased.

In this paper, an agent-based approach is used for sensor selection and positioning in a single-target multi-object environment. In contrast to the abovementioned systems, however, external virtual

[1] Cooperative agents are those that work together to improve system performance rather than their own performance.

agents are used for conflict detection and management. The use of virtual agents ensures desirable global behaviour of the multi-agent system and simplifies the conflict-management strategy. The virtual agents also have the added advantage of reducing the amount of communications needed between the agents. Our system also differs from the systems described above in that the unused sensors are not utilized for target detection but are rather positioned in anticipation of future service requirements.

1.4 Active-Sensing for Object Recognition via Shape-from-Shading

The implementation example considered in this paper is the use of active sensing in object recognition via *Shape-from-Shading* (SFS). SFS refers to a process of recovering surface orientation from local variations in perceived brightness [28]. The main limitation to early SFS algorithms, in 3D recognition, was a possible failure in providing sufficiently accurate surface information. Some recent algorithms have succeeded in reducing noise in shape recovery, but unfortunately, they also significantly reduce surface details by applying a variety of smoothing techniques [29].

A second challenge in successful object recognition via SFS has been the complexity in constructing a database of 3D object models and comparison of surface-orientation information derived from acquired 2D images with these models. A simple and effective method for overcoming this problem has been the employment of appearance-based representation techniques, which rely on 2D *Characteristic Views* (CVs) of the object for 3D representation (e.g., [37]-[39]). The objective is to achieve an acceptable *recognition confidence* by grouping several views to yield the minimum number of CVs.

There has also been some interest in using data fusion for increased accuracy in shape recovery from photometric stereo images and, in turn, increased recognition performance (e.g., [40]-[41]). However, the majority of such algorithms require images to be acquired at constant viewing directions with only variations in lighting conditions. Although a valid technique for recognition in static environments, this restriction would severely limit the implementation of such algorithms in the surveillance of dynamic environments.

In a dynamic environment, objects (OoI and obstacles) move continuously and viewing conditions cannot be held constant. Therefore, in order to utilize multi-camera surveillance, a data-fusion technique, where information recovered from images taken at different viewing directions and lighting conditions are required. Further performance improvements can be achieved by utilization of active cameras to acquire images at preferred viewing angles.

2. SENSING-SYSTEM RECONFIGURATION METHODOLOGY

In the context of sensing-system reconfiguration, sensor dispatching attempts to maximize the effectiveness of the surveillance-system, which is used to provide estimates of OoI parameters at predetermined times along its trajectory. These predetermined times are referred to as demand instants, t_i. It is assumed that the pose of the OoI at a particular demand instant, is predicted from observations of the OoI motion rather than known *a priori*. In general, the estimation of the OoI pose at a demand instant changes (and its corresponding uncertainty diminishes) as the prediction accuracy improves over time.

If the sensing-system comprises multiple sensors (cameras, in our case), a subset of these may be sufficient to satisfy the sensing requirements of a demand instant. Namely, a data-fusion process does not need to combine information from all the sensors. Instead, a subset of sensors, herein referred to as the fusion subset, may be selected to survey the OoI at a particular demand instant, allowing other sensors to be configured in anticipation of future use. In this context, in our previous work, we addressed this *dispatching problem* using heuristics and a blackboard approach [19]. In this paper, a novel agent-based approach is applied to the problem at hand. The agent-base approach increases scalability since new cameras can be added by only creating the associated sensor agent. Furthermore, detected faulty sensors can be removed by simply removing the associated sensor agent.

2.1 Quality of the Sensing Data

A *visibility measure* is employed in the selection of cameras for their inclusion in a fusion subset and assessment of their desired poses. This metric measures the quality of the sensory data that would be collected given the environmental conditions, such as viewing angle and lighting direction. The visibility measure for the j^{th} sensor servicing the i^{th} demand instant is defined herein as

$$v_{ij} = \begin{cases} f(\theta, \gamma) & \text{if demand point is unoccluded} \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where θ is the viewing angle of the camera and γ is the lighting angle, as shown in Figure 1.

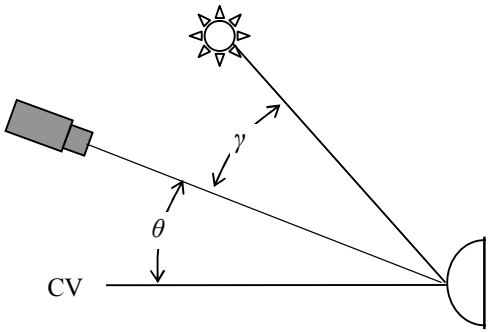


Figure 1 .Viewing and lighting angles – θ and γ, respectively.

The function $f(\theta,\gamma)$ is determined through sensor modeling. An example function used in this work, derived via two-factorial experiments, is given below, Figure 2:

$$f(\theta,\gamma) = 0.0396 - 0.0024x + 0.0000323x^2 + 0.00011y - 0.0000029y^2 + 0.000005xy\,. \quad (2)$$

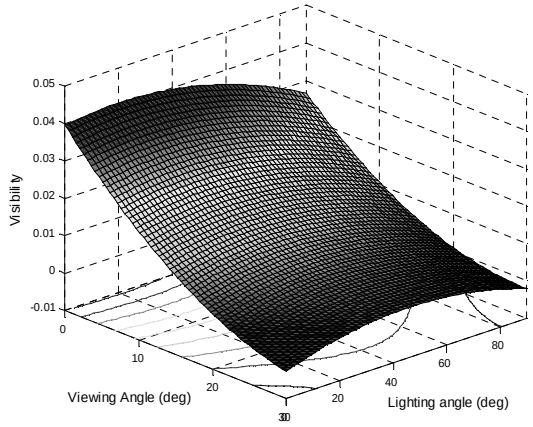


Figure 2. An example visibility function.

The visibility measure over the span of a *rolling horizon* is defined as:

$$v_j = \sum_{i=1}^{m} a_i v_{ij}\,, \quad (3)$$

where m is the number of demand instants in the rolling horizon and a_i is the weight of the i^{th} demand instant. The weight factor is constant for all sensors and represents the uncertainty in the predictions of the future object poses.

2.2 Coordination Strategy

Dispatching can be accomplished using two complementary strategies: A coordination strategy to determine how many and, specifically, which sensors should be used at each demand instant in order to optimize the performance of the surveillance system over the span of the rolling horizon; and, a positioning strategy to determine the optimal pose of each sensor at each demand instant.

The proposed agent-based system consists of multiple *sensor* agents, a *referee* agent, and a *judge* agent. Each sensor agent tries to maximize its own performance over the span of the rolling horizon. Although not directly controlled by a centralized controller, the sensor agents must abide the external rules of the environment monitored and enforced by two virtual agents. The rules are set to ensure that the collective behaviour of the sensor agents exhibits the desired system behaviour.

2.2.1 Sensor Agents

The sensor agent is responsible for choosing the demand instants that the associated sensor will service and determining its optimal poses in order to maximize the sensor's performance metric (i.e., visibility) over the span of the rolling horizon. If a demand instant is not serviced, the sensor would have zero visibility for that demand instant, however, it would allow more time for the sensor to manoeuvre for the next demand instant.

Each sensor agent searches through all possible combinations using a depth-first approach (e.g., <1,1,1> is a combination referring to servicing all demand instants in a 3-demand-instant horizon). The total search space for a sensor agent is 2^m, where m is the number of demand-instants in the rolling horizon. However, certain combinations will by definition always have lower visibility than others and, therefore, might not have to be searched. For example, if combination <1,1,1> is achievable (not occluded), then, combination <1,1,0> will not have any advantage for the sensor and will always have lower visibility; thus, it does not have to be searched.

At each combination searched, the sensor agent determines the best achievable poses to service the selected demand instants through the positioning strategy outlined in Section 2.3. Using the optimum poses and the OoI's predicted locations, the sensor agent determines the expected achievable visibility for each combination. The sensor agent, then, evaluates the combinations searched to determine acceptable solutions. Acceptable solutions are constrained by the following *internal* rules:

1. A demand instant cannot be serviced if it is occluded; and

2. Combination [0, 0, 0, … , 0], representing a sensor not being assigned to any demand instant, is

only considered if all other combinations are occluded.[2]

Next, the sensor agent ranks all acceptable combinations in the descending order of combined visibilities. The r^{th} ranked acceptable solution for the j^{th} sensor is denoted herein as $\boldsymbol{S}_{jr}$. The sensor agent sends the first ranked acceptable solution, $\boldsymbol{S}_{j1}$, to the referee agent.

2.2.2 Referee Agent

The referee agent monitors the intentions of the sensor agents and ensures that no *external* rules are violated. External rules would depend on the surveillance task at hand and are, thus, user-specified. For example, in this work, the following external rule is defined, in order to ensure that the sensors are well distributed among the demand instants of the rolling horizon:

- *At least one sensor must be assigned to each demand instant.*

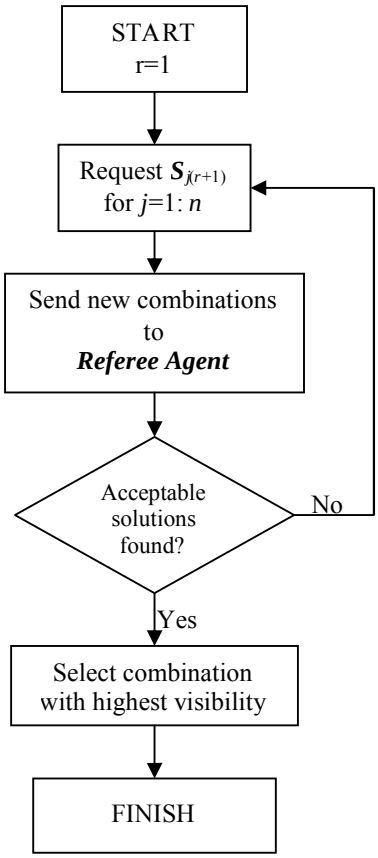
If the referee agent detects a violation of the external rules it initiates the judge agent in order to resolve the conflict.

2.2.3 Judge Agent

Upon initiation, the judge agent sends a command to each sensor agent requesting the sensor agents' 2nd-ranked acceptable solutions, $\boldsymbol{S}_{j2}$. Along with these alternate solutions, each sensor agent also sends the corresponding expected visibilities. The judge agent uses a depth-first approach[3] to search through all possible permutations of 1st and 2nd ranked solutions for combinations that would resolve the conflict (an example combination of 1st and 2nd ranked solutions of 4 sensor agents is $[S_{11}\ S_{21}\ S_{32}\ S_{41}]$). The judge agent, then, selects the combination with the highest visibility and informs the sensor agents of its decision. If no acceptable combination is found, the judge agent increases the depth of the search space by requesting the sensors agents' 3rd-ranked solutions, as shown in Figure 3. This process is repeated until an acceptable combination is found or the allowable search time has elapsed. In the event that no acceptable combination is found within the allowable search time, the 1st-ranked solutions (initial sensor agents' intentions) are used. This ensures that the system is not in a virtual deadlock if no solution exists that would satisfy the external rules.

[2] The [0, 0, … , 0] combination allows the sensor to simply follow the target so that it may be used in the future.

[3] It should be noted that the depth-first search approach does not guarantee an optimal solution (as can be done through an exhaustive search of the entire search). The objective is to search for the best acceptable solution within the allowable search time.



Note: n is the total number of sensor agents

Figure 3. Flowchart for judge agent's search for an acceptable solution.

2.3 Positioning Strategy

In a single-target multi-object environment, the positioning strategy is performed not only based on the trajectory of the OoI (i.e., the target) but also based on other objects that are not of interest but may act as occlusions. The first step in determining the best achievable pose is to determine the occluded regions in the workspace. In order to accomplish this, the pose of each object (OoI or obstacle) is predicted for the demand instant. Next, each object is modeled as a single geometric primitive (e.g., a sphere or cylinder), rather than as a collection of 3D polyhedra, in order to decrease computational complexity. Occluded regions of a sensor's workspace are determined by modeling the OoI as a light source and calculating geometric shadow volumes [25], cast by the obstacles, using the pose and size of each object in the workspace, as shown in Figure 4. The algorithm subsequently determines the region of the workspace that the sensor can travel to before the

target reaches the demand instant, referred to herein as *feasible region*. This region is defined by the sensors' dynamic motion capabilities such as maximum velocity, v_{max}, acceleration, a, as well as time to next demand instant, dt. For a sensor with one degree-of-freedom (dof) in translation (along the x axis) the feasibility region, $x_{feasible}$, is defined as

$$x_l \le x_{feasible} \le x_r . \tag{4}$$

In (4), x_r is the right limit defined by

$$x_r = v_o(dta_1) + \frac{1}{2}a(dta_1)^2 + v_{\max}(dtc_1) + v_o(dts_1) + \frac{1}{2}a(dts_1)^2, \tag{5}$$

where v_o is the current sensor velocity, dta_1, dts_1, and dtc_1 are the time the sensor travels while accelerating, decelerating, and at constant velocity, respectively, in order to get to the right travel limit, each defined by

$$dta_1 = \begin{cases} \frac{v_{\max} - v_o}{a} & if \quad \left(\frac{2v_{\max} - v_o}{a}\right) < dt \\ \frac{1}{2}(dt - \frac{v_o}{a}) & else \end{cases}, \tag{6}$$

$$dts_1 = \begin{cases} \frac{v_{\max}}{a} & if \quad \left(\frac{2v_{\max} - v_o}{a}\right) < dt \\ \frac{1}{2}(dt + \frac{v_o}{a}) & else \end{cases}, \text{ and} \tag{7}$$

$$dtc_1 = dt - (dta_1 + dts_1), \tag{8}$$

In (4) x_l is the left limit defined by

$$x_l = v_o(dta_2) - \frac{1}{2}a(dta_2)^2 - v_{\max}(dtc_2)^2, + v_o(dts_2) - \frac{1}{2}a(dts_2) \tag{9}$$

where dta_2, dts_2, and dtc_2 are the time the sensor travels while accelerating, decelerating, and at constant velocity, respectively, in order to get to the left travel limit, each defined by

$$dta_2 = \begin{cases} \frac{v_{\max} + v_o}{a} & if \quad \left(\frac{2v_{\max} + v_o}{a}\right) < dt \\ \frac{1}{2}(dt + \frac{v_o}{a}) & else \end{cases}, \tag{10}$$

$$dts_2 = \begin{cases} \frac{v_{\max}}{a} & if \quad \left(\frac{2v_{\max} + v_o}{a}\right) < dt \\ \frac{1}{2}(dt - \frac{v_o}{a}) & else \end{cases}, \text{ and} \tag{11}$$

$$dtc_2 = dt - (dta_2 + dts_2). \tag{12}$$

It should be noted that for sake of simplicity the limits of the workspace have not been included in the equations above.

Lastly, the algorithm determines an optimal sensor pose that would yield maximum visibility, which is both feasible and unoccluded (i.e., acceptable regions). This is done by discretizing the acceptable region into a pre-specified number of positions. An optimal pose is selected by evaluating the visibility metric at each discrete position.

The coordination and positioning of each sensor is repeated continuously as new information regarding the environment becomes available. This ensures that new and more accurate target pose predictions are utilized. Furthermore, as time approaches the demand instant (i.e., $dt \to 0$), the size of the acceptable region diminishes and, therefore, it would be more densely discretized resulting in more accurate sensor pose determination.

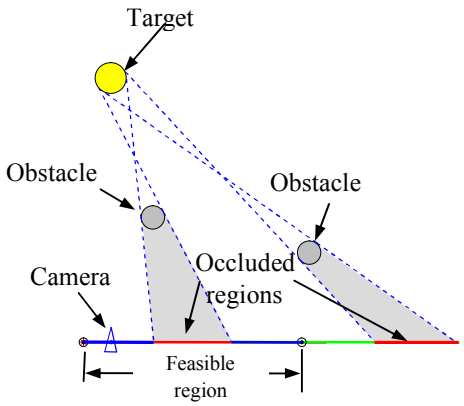


Figure 4. An example of occluded regions of a sensor's workspace.

3. SHAPE FROM SHADING

Majority of SFS algorithms (e.g., [40]-[41]) require images to be acquired at constant viewing directions with only variations in lighting conditions. Although a valid technique for recognition in static environments, this restriction would severely limit the implementation of such algorithms in the surveillance of dynamic environments. The SFS-based recognition algorithm proposed in this paper, on the other hand, uses a high-level data-fusion technique to fuse information recovered from images acquired at different viewing directions and lighting conditions from several cameras. It should be mentioned that SFS algorithm require a point light source for accurate shape recovery.

3.1 Depth Maps

There exist four major SFS techniques: minimization approaches, propagation approaches, local approaches, and linear approaches. Minimization approaches (e.g., [48]-[49]) obtain the solution (local surface orientations) by minimizing an error function via established optimization techniques such as gradient decent. Although computationally expensive, minimization approaches produce the most accurate results. Propagation approaches (e.g., [50]-[51]) propagate the shape information from a set of known surface points (e.g., singular points) to the whole image. Local approaches (e.g., [52]) derive shape based on the assumption of surface type. Both propagation and local SFS approaches, typically, require a large amount of *a priori* knowledge about the surface being reconstructed. Linear approaches (e.g., [42] and [53]) compute the solution based on the linearization of the reflectance map.[4] This approach is very computationally efficient; however the accuracy of the algorithm drops significantly with increasing non-linearity of the reflectance map. In this paper, a minimization approach initialized by an estimated surface, recovered through a linearization approach, is used. The aim is to reduce computational cost by having a good initial estimate from the linearization approach and achieving the required accuracy for object recognition through the minimization approach.

The minimization SFS technique described below is a variation of the one originally presented in [40], where a triangular element surface model and a linearized reflectance map are used to iteratively determine accurate surface normals. The algorithm starts by dividing the image into a set of non-overlapping triangular sectors. It is assumed that the image intensity within each triangular domain is homogeneous, so that a direct relationship between image intensity and surface nodal height can be established via the following image irradiance equation:

$$E(x_c, y_c, z_c) = R_e\left(n'_x(x_c, y_c), n'_y(x_c, y_c)\right), \quad (13)$$

where $R_e(n'_x, n'_y)$ is the reflectance map and $n'_x(x_c, y_c) = \partial z(x_c, y_c)/\partial x_c$ and $n'_y(x_c, y_c) = \partial z(x_c, y_c)/\partial y_c$ represent the local surface orientation in camera coordinates, Figure 5.

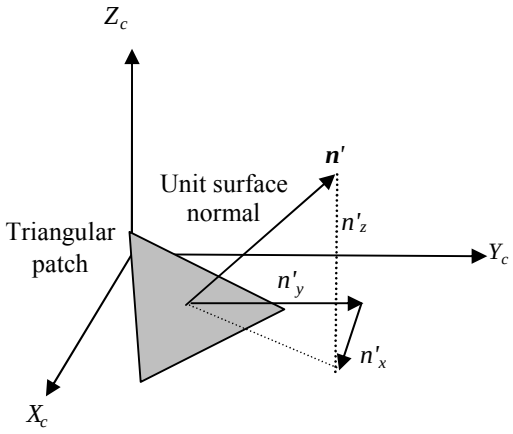


Figure 5. Triangular patch and the projections of its surface normals.

Assuming ideal Lambertian surface illumination by a single distant point light source, the reflectance map is expressed as:

$$R_e = \begin{cases} \eta \dfrac{K}{\sqrt{1+(n'_x)^2+(n'_y)^2}}, & k \geq 0, \\ 0, & k < 0 \end{cases} \quad (14)$$

where

$K = -n'_x \cos\tau_L \sin\sigma_L - n'_y \sin\tau_L \sin\sigma_L + \cos\sigma_L$; and

η is the composite albedo of the surface, and τ_L and σ_L are the tilt and slant angles of the illumination direction, respectively, Figure 6.

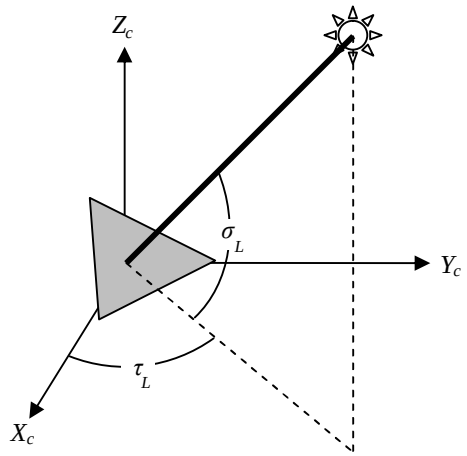


Figure 6. Illumination direction defined by its slant and tilt angles.

[4] The reflectance map determines the proportion of light reflected as a function of local surface normals.

In order to determine the surface normal, for each triangular patch, a global cost function is defined. The cost function is the sum of squared brightness errors over each triangular image domain:

$$E_u = \sum_u \left(E_u - \hat{E}_u\right)^2, \tag{15}$$

where E_u and $\hat{E}_u$ are the observed and reconstructed image intensities over the u^{th} triangular domain, respectively. The cost function is minimized using the iterative method described in [40]. The surface normals are sufficient for the recognition algorithm used in this paper. However, if needed the surface normals can also be used to recover a relative depth map, as illustrated in Figure 7, by assigning reference height to a triangular patch and determining the relative height of the centre of neighbouring patches using their surface normals [54].

In order to reduce the number of required iterations, the initial estimate of local surface normals are based on the results obtained by using the closed-form SFS method presented in [42]. This approach reduces the non-linear SFS problem into a linear one through linearization of the reflectance map. Using the linear reflectance map, the algorithm provides a non-iterative, closed-form solution using Fourier transform.

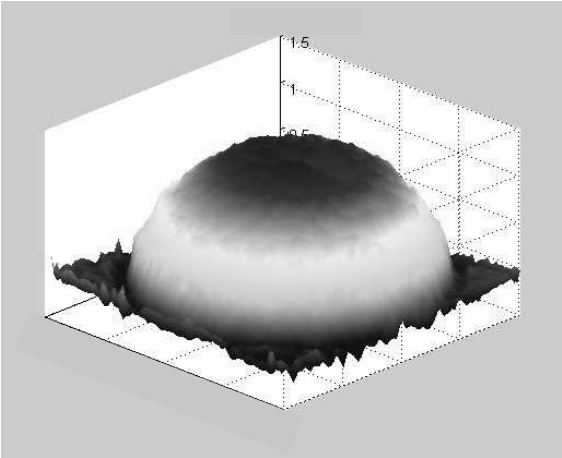


Figure 7. Sample depth map of a spherical object recovered through SFS.

3.2 Single-Image Object Recognition

The result of the SFS algorithm is a set of local surface normals, which are used to determine the local slant and tilt angles. These angles are computed relative to a *mean* normal direction calculated over the entire image (reducing the sensitivity of the recognition process to rotations about the optical axis). The mean surface normal, $\hat{\boldsymbol{n}}$, is defined by:

$$\hat{\boldsymbol{n}} = \frac{1}{(Rw)(Cl)} \sum_{I=1}^{Rw} \sum_{J=1}^{Cl} \boldsymbol{n}'(I,J), \tag{16}$$

where Rw and *Cl* are the number of rows and columns in the image, respectively, and $\boldsymbol{n}'(I,J)$ is the local unit normal vector. The local slant, σ, and tilt, τ, angles are, then, expressed as

$$\sigma(I,J) = \cos^{-1}\left(n'_z(I,J)\right) - \cos^{-1}\left(\hat{n}_z\right), \tag{17}$$

and

$$\tau(I,J) = \tan^{-1}\left(\frac{n'_y(I,J)}{n'_x(I,J)}\right) - \tan^{-1}\left(\frac{\hat{n}_y}{\hat{n}_x}\right), \tag{18}$$

where $\hat{n}_x$, $\hat{n}_y$, and $\hat{n}_z$ are the components of the mean surface normal, $\hat{\boldsymbol{n}}$, along the X_c, Y_c, and Z_c axis respectively.

The overall image is represented by a 2D histogram of local slant and tilt angles, as shown in Figure 8. This allows the utilization of a standard histogram-recognition scheme [43]. It should be noted that although the scheme ignores the spatial arrangement of an image, it provides a recognition method that is invariant to minor deviations of object position within the image. In this work, the distance between two histograms is measured using the Root-Mean-Squares (RMS) value of their difference. This method is chosen for its low computational expense over more sophisticated comparison methods such as Bhattacharyya distance and matrix norm based on Singular Value Decomposition (SVD) [46].

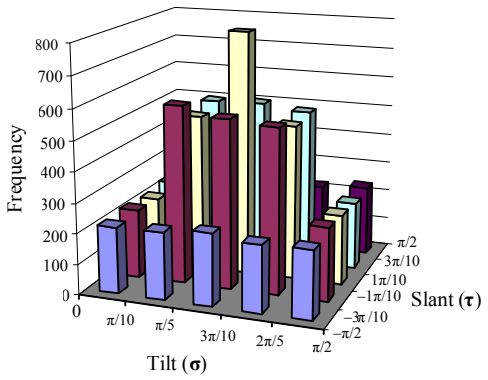


Figure 8. Example 2D histogram obtained via SFS.

Every time an image is acquired, it is processed to extract a needle-map that is used for calculating a 2D histogram. The histogram is, then, compared to

histograms of objects within the database[5]. This result is in terms of an *RMS* value for each object within the database. We, then, construct a similarity vector of 1/*RMS*,

$$\delta = \left[\frac{1}{RMS(\hat{\Pi} - \Pi_1)}, \frac{1}{RMS(\hat{\Pi} - \Pi_2)}, \ldots \right], \qquad (19)$$

where $\hat{\Pi}$ is the image histogram and Π_1 is the histogram of the first object in the database, Π_2 the second object, and so on. Figure 9 shows the recognition process.

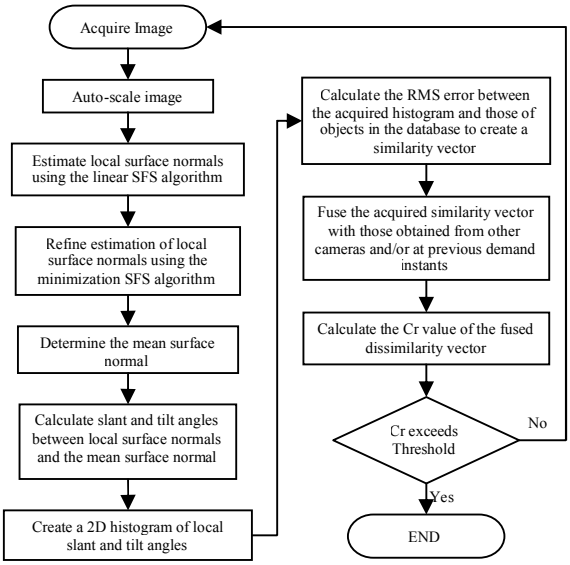


Figure 9. Algorithmic flowchart of the recognition process.

3.3 Characteristic Views

In order to reduce the negative impact of variations in viewing direction on object recognition, we utilize multiple CVs to describe each object in the database. Each CV represents the object from a different viewing direction. Since a series of CVs are required to describe an object, the recognition method is essentially an appearance-based technique. In this work, we use three CVs to describe each object in the database, as illustrate d in Figure 10.

3.4 Multi-Image Fusion

The resulting similarity vectors from multiple cameras that have participated in the surveillance of the object-of-interest (OoI) are fused in order to reduce uncertainty and increase robustness of the recognition algorithm. The fusion process is a weighted average one based on the visibility metric,

$$\Delta_c = \frac{\sum_j \sum_i \delta_{cij} v_{ij}}{\sum_j \sum_i v_{ij}}, \qquad (20)$$

where Δ_c is the c^{th} element of the multi-camera similarity vector $\boldsymbol{\Delta}$, δ_{cij} is the c^{th} element of the similarity vector of the j^{th} camera at the i^{th} demand instant, and v_{ij} is its visibility metric. As can be noted, all similarity vectors acquired from each camera, starting at the first demand instant to the current demand instant, are fused. The OoI is identified as the one with the maximum value in the current multi-camera similarity vector.

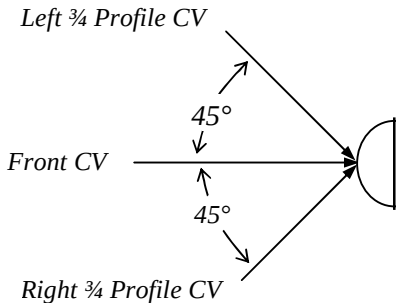


Figure 10. Characteristic views (top view).

3.5 Confidence of Recognition

The proposed system continues imaging the target at every demand instant until it has been identified with a preset confidence level. As before, the confidence level of recognition, *Cr*, is defined via the multi-camera similarity vector,

$$Cr = \frac{\Delta_{\max_1} - \Delta_{\max_2}}{\Delta_{\max_1}}, \qquad (21)$$

where $\Delta_{\max_1}$ and $\Delta_{\max_2}$ are the first and second maxima of the elements of the similarity vector $\boldsymbol{\Delta}$, respectively.

[5] The database is created by acquiring images of the objects at varying lighting conditions. The photometric SFS algorithm fuses the images in order acquire a more accurate surface representation than would be achievable from any one image.

4. EXPERIMENTS

4.1 Experimental Set-up

An experimental prototype set-up is devised to examine the proposed sensing-system reconfiguration methodology for object recognition via SFS. This system uses four mobile cameras to identify a single OoI, manoeuvring through the workspace on a planar trajectory, as shown in Figure 11. The environment is cluttered with dynamic obstacles that act as occlusions. A stationary overhead camera obtains gross estimates of the motions of all objects within the workspace. Two fixed light sources are positioned on each side the workspace. Based on the information obtained from the overhead camera and the known position of the light sources, the dispatching algorithm selects and positions cameras for optimal target imaging.

The linear SFS algorithm used in this paper provides an estimate of the lighting direction in camera coordinates. Through calibration, estimates from each camera at different demand instants can be used to determine the light position, as shown in Figure 12. However, this estimate is only available after two demand instants.[6] Furthermore, errors in estimation of the position of the OoI, camera parameters obtained via calibration, as well as the estimation of lighting direction will all affect the accuracy in determining the position of the light source. Therefore, in this work, the position of the light source is assumed to be known, as would be the case in many surveillance systems. This is consistent with the generic active-vision problem where lighting conditions as well as camera poses are controlled by the surveillance system.

Hardware: The experimental system uses four cameras to recognize the target as it manoeuvres through the workspace on planar trajectories. All cameras have one-dof rotational capability (pan), while two of the cameras can also translate linearly, Table 1. The environment (500×500 mm) is cluttered with other objects that are marked as *occlusions*. A single static overhead camera, with a wide-angle lens, is used to survey the entire workspace for target tracking: namely, the overhead camera is used to obtain gross estimates of the motions of all subjects within the workspace. If, in practice, this may not be possible, other tracking methods that utilize multiple static cameras may be used (e.g., [55]).

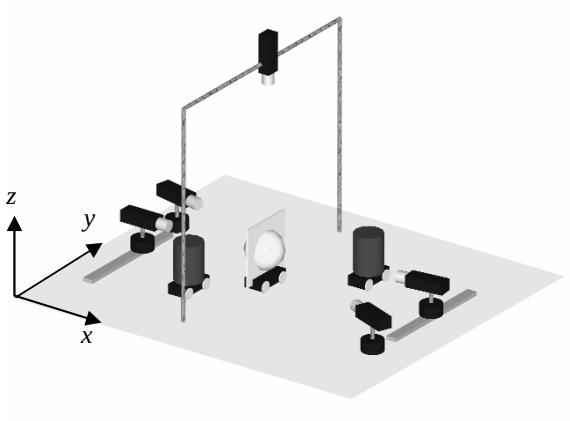


Figure 11. Experimental system layout.

Table 1. Sensing System Hardware specifications.

Hardware	Characteristic
Linear Stages	*Range: 300 mm* *Positional Accuracy: 18 μm* *Max velocity: 1.5 m/sec*
Rotary Stages	*Positional Accuracy: 12 arc sec* *Max velocity: 15 rev/sec*
Horizontal CMOS Cameras	*Resolution: 640x480 pixels*
Overhead CCD Camera	*Resolution: 640x480 pixels*

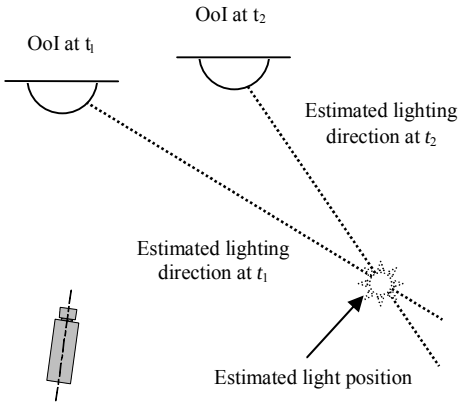


Figure 12. Calculating light position from estimates of lighting directions obtained at different demand instants via SFS.

Software: The surveillance system's software consists of a collection of primary agents (*Sensor Agents*, *Referee Agent*, and *Judge Agent*) and

[6] For applications where illumination is assumed to be a distant light source (e.g., the sun), the lighting direction does not change from one demand instant to the next and estimation of light source position will not be necessary. Therefore, a single demand instant will be sufficient to estimate lighting direction for subsequent demand instants.

other agents that provide supporting functions (i.e., *Tracking and Prediction Agent* and *Data-Fusion Agent*), as shown in Figure 13.

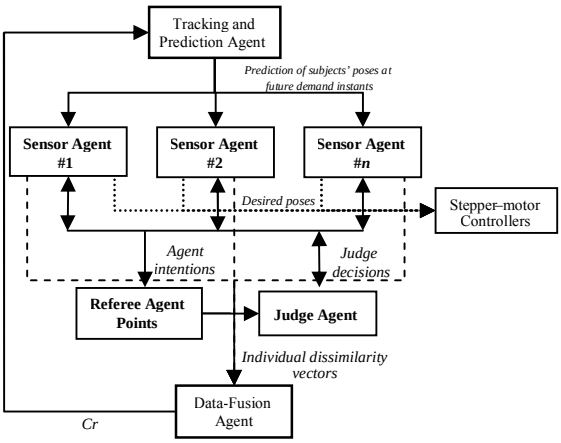


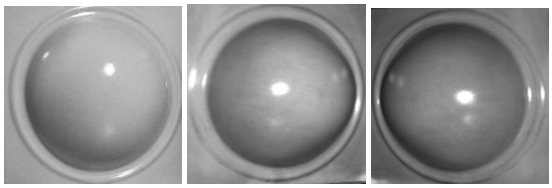
Figure 13. Software architecture of the active surveillance system.

Experimental Results

The experiment presented herein discusses the recognition of the object shown in Figure 14, where the database consists of six arbitrarily chosen objects from the Columbia Object Image Library [47], as shown in Figure 15. The system's mobility is restricted by 40 mm/s translational camera velocities. The OoI and obstacles manoeuvre through the workspace at 20 mm/s and 23 mm/s, respectively, on trajectories shown in Figure 16. In this experiment, the demand instants are set to 5 seconds apart to allow the system to perform shape recovery, histogram matching, data-fusion, and recognition. This results in 6 demand instants during the OoI's trajectory.

At each demand instant, assigned cameras image the target, recover surface information, and perform object recognition. The result from each camera is, then, fused with the results obtained from other cameras in the current and previous demand instants. The surveillance system is expected to continue servicing demand instants until the OoI is recognized with a predefined confidence. However, in this experiment the confidence of recognition threshold was purposely not defined so that the object would be imaged for the entire duration of its trajectory. The confidence of recognition associated with each system, after every demand instant, is plotted in Figure 18 and the achievable visibility is shown in Table 1. Some pre-processed sample images acquired by the system during the experiment are shown in Figure 17.

As previously mentioned, the data-fusion of information gathered by multiple cameras observing the target from varying viewing directions has not been previously implemented for object recognition via SFS. Therefore, in this experiment, the confidence of recognition of a single camera (Camera 4) without data-fusion is also plotted on Figure 18, in order to highlight the performance increase achieved through data-fusion.



(a) (b) (c)

Figure 14. Sample *database* images for OoI (a) front and (b) left ¾ profile, and (c) right ¾ profile.

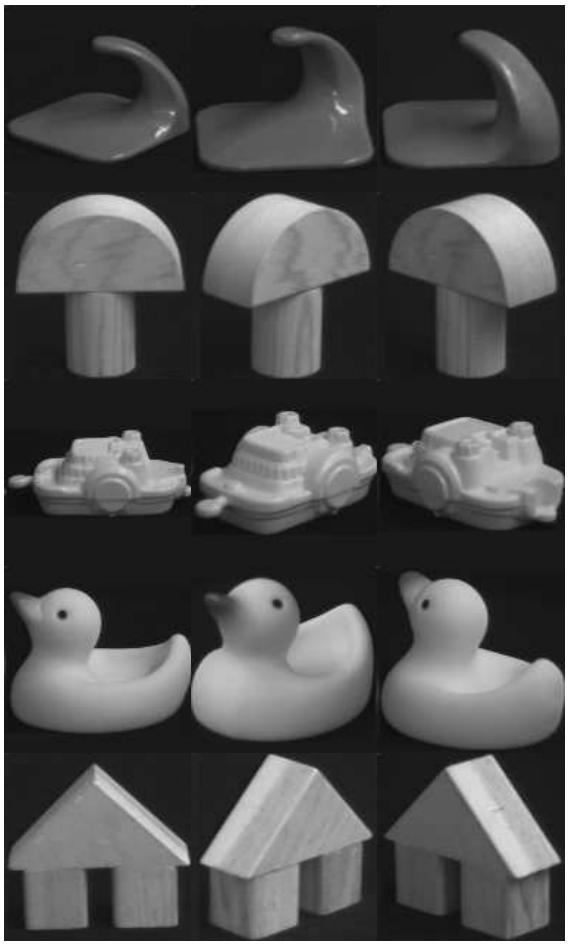


Figure 15. Sample images from the Columbia Object Image Library used for front, left ¾ side, right ¾ side view databases.

As can be noted from Figure 18, the confidence of recognition is tangibly higher using data fusion than what would be achievable with a single camera. Furthermore, the confidence of recognition for the fused multi-camera system continuously increases, whereas in the case of a single camera, without data fusion, the confidence of recognition does not show any positive trend and may even produce false recognitions (e.g., Demand Instant 4 with *Cr*=0.3 recognized the OoI as the *wooden peg*, shown in second column of Figure 15). Thus, the experiments verified the enhanced recognition performance achieved by utilizing and fusing data from multiple cameras observing the OoI from varying viewing directions.

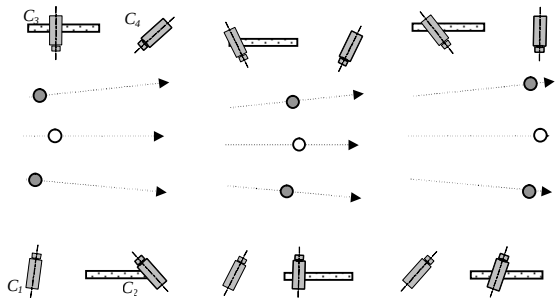


Figure 16. Camera and OoI poses at (a) 2^{nd}, (b) 4^{th}, and (c) 6^{th} demand instants.

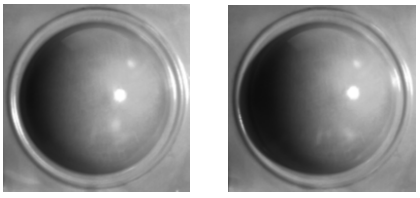


Figure 17. Sample images obtained by (a) Camera 3 at Demand-Instant 1 and (b) Camera 4 at Demand Instant 5.

5. CONCLUSIONS

A novel agent-based methodology is presented for the coordinated selection and positioning of groups of active-cameras (i.e., dispatching) for the autonomous recognition of a manoeuvring target in a multi-object dynamic environment. Furthermore, an efficient and effective object recognition method via shape-from-shading is also presented. In contrast to majority of work presented in the literature, our system fuses data from multiple cameras observing the target from varying viewpoints. It has been shown in simulations and experiments (some of which are presented herein) that multi-camera data-fusion can tangibly increase the accuracy and robustness of a recognition process.

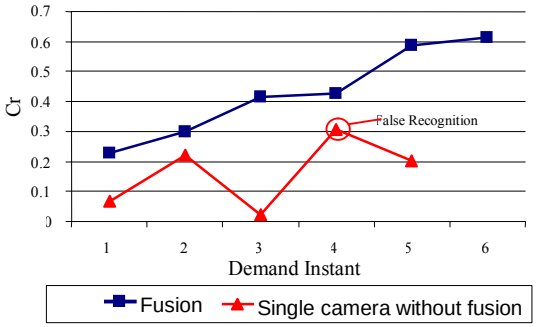


Figure 18. Confidence of recognition, *Cr*, after each demand instant.

Table 1. Visibility of each camera at every demand instant.

Demand Instant	C_1	C_2	C_3	C_4
1	0.04	0.03	0.04	0.02
2	0.03	0.03	0.04	0.03
3	*Occluded*	0.04	0.03	0.02
4	*Occluded*	0.04	0.01	0.01
5	0.04	0.03	0.03	0.01
6	0.03	0.01	0.03	*Occluded*

REFERENCES

[1] K.A. Tarabanis, P.K. Allen, and R.Y. Tsai, "A survey of sensor planning in computer vision," *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 1, pp. 86-104, Feb. 1995.

[2] S. Sakane, T. Sato, and M. Kakikura, "Model-based planning of visual sensors using a hand-eye action simulator: HEAVEN," In B. Espiau, editor, *Proc. Conf. on Advanced Robotics*, pp. 163–174, Versailles, France, Oct. 1987.

[3] G.H. Tarbox and S.N. Gottschlich, "Planning for complete sensor coverage in inspection," *Computer Vision and Image Understanding*, Vol. 61, No. 1, pp. 84-111, Jan. 1995.

[4] C.K. Cowan and P.D. Kovesik, "Automated sensor placement from vision task requirements," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 3, pp. 407-416, May 1988.

[5] D. P. Anderson, "Efficient algorithms for

automatic viewer orientation," *Trans. Comp. & Graphics*, Vol. 9, No. 4, pp. 407-413, 1985.

[6] H. Zhang, "Two-dimensional optimal sensor placement," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 25, No. 5, pp. 781-792, May 1995.

[7] E. Trucco, M. Umasuthan, A.M. Wallace, and V. Roberto, "Model-based planning of optimal sensor placements for inspection," *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 2, pp. 182-194, Apr. 1997.

[8] W. Sheng, N. Xi, M. Song, and Y. Chen, "CAD-guided sensor planning for dimensional inspection in automotive manufacturing," *IEEE-ASME Transactions on Mechatronics*, Vol. 8, No. 3, pp. 372-380, Sep. 2003.

[9] R. Niepold, S. Sakane, and Y. Shirai, "Vision sensor set-up planning for a hand-eye system using environmental models," In *Proceedings of the Society of Instrument and Control Engineers of Japan*, Vol. 7, No. 1, pp. 1037-1040, Hiroshima, Japan, July 1987.

[10] T. Matsuyama, T. Wada, and S. Tokai, "Active image capturing and dynamic scene visualization by cooperative distributed vision," In S. Nishio and F. Kishino, editors, *Advanced Multimedia Content Processing*, Vol. 11, No. 4, pp. 252-288, Springer-Verlag, Berlin, 1999.

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

[12] J. R. Spletzer, and C. J Taylor, "Dynamic sensor planning and control for optimally tracking targets," *Int. Journal of Robotic Research,* Vol. 22, No. 1, pp. 7-20, Jan. 2003.

[13] M. Kamel, and L. Hodge, "A coordination mechanism for model-based multi-sensor planning," *Proc. of the IEEE Int. Symp. on Intelligent Control*, pp. 1143-1149, Vancouver, Canada, Oct. 2002.

[14] H. Zhou, and S. Sakane, "Sensor planning for mobile robot localization using Bayesian network inference," *J. of Advanced Robotics,* Vol. 16, No. 8, pp. 751-771, 2002.

[15] R.Y. Tsai and K. Tarabanis, "Model-based planning of sensor placements and optical settings," in *Sensor Fusion II: Human and Mach. Strategies*, pp. 936-944, Philadelphia, PA, Nov. 1989.

[16] R.Y. Tsai and K. Tarabanis, "Occlusion-free sensor placement planning." In *Machine Vision for Three Dimensional Scenes*, H. Freeman, Ed., pp. 349-356, Orlando, FL: Academic, 1990.

[17] S.G. Goodridge and M.G. Kay, "Multimedia sensor fusion for intelligent camera control," in P*roc of IEEE/SICE/RSJ Multi-sensor Fusion and Integration for Intelligent Systems*, pp. 934-940, Washington, D.C, Dec. 1996.

[18] R.E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of ASME, Journal of Basic Engineering*, Vol. 82, No. 4, pp. 35-45, 1961.

[19] A. Bakhtari, M. Eskandari, M.D. Naish, and B. Benhabib "A multi-sensor Surveillance system for active-vision based object localization," Proc. of the *IEEE Int. Conf. System, Man and Cybernetics*, pp. 1013-1018, Washington, D.C., Oct. 2003.

[20] N. Ukita and T. Matsuyama, "Real-time cooperative multi-target tracking by communicating active vision agents," *Proc. of the Int. Conf. on Information Fusion*, pp. 439-446, Queensland, Australia, 2003.

[21] D.J. Cook, P. Gmytrasiewicz, and L.B. Holder "Decision-theoretic cooperative sensor planning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 10, pp. 1013-1023, Oct. 1996.

[22] D. Cook, "Reconfiguration of multi-agent planning systems," *Proc. Artificial Intelligence Planning Systems*, pp. 225-230, Chicago, IL, 1994.

[23] E. Merchand and G.D. Hager "Dynamic sensor planning in visual servoing," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1988-1993, Leuven, Belgium, 1998.

[24] R. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, Vol. 3, No. 4, pp. 323-344, 1987.

[25] K. Thakur, F. Cheng, and K.T. Miura, "Shadow generation using discretized shadow volume in angular coordinates," *Proc. of the IEEE Computer Graphics and Applications,* Canmore, Canada, pp. 224-233, 2003.

[26] R. Brooks and S.S. Iyengar, *Multi-sensor fusion: Fundamentals and applications with software*, Englewood Cliffs, NJ: Prentice Hall, 1998.

[27] K. Marzullo, "Tolerating failures of continuous-valued sensors," *ACM Transactions on Computer Systems*, Vol. 8, No. 4, pp. 284-304, 1990.

[28] B.K.P. Horn, "Understanding image intensity," *Artificial Intelligence*, Vol. 8, pp.

201-231, 1977.

[29] B.K.P. Horn and M.J. Brooks, "The variation approach to shape from shading," *Computer Vision, Graphics, and Image Processing*, Vol. 33, No. 2, pp. 174-208, 1986.

[30] J. Oliensis and P. Dupuis, "An optimal control formulation and related numerical methods for a problem in shape reconstruction," *Annals of Application Probability*, Vol. 4, No. 2, pp. 287-346, 1994.

[31] E. Rouy and A. Tourin, "A viscosity solutions approach to shape-from-shading," *Journal of Numerical Analysis*, Vol. 29, No. 3, pp. 867-884, 1992.

[32] R. Kimmel and A. M. Bruckstein, "Tracking level-sets by level-sets: a method for solving the shape from shading problem," *Computer Vision and Image Understanding*, Vol. 62, No. 1, pp. 47-58, 1995.

[33] P. L. Worthington and E. R. Hancock, "3D surface topography from intensity images," Proc. of the *IEEE Int. Conf. on Computer Vision*, pp. 911-917, Kerkyra, Greece, 1999.

[34] N. A. Nokra, L. Lecornu, B. Zerr, B. Solaiman, and C. Sintes, "Generation of an ideal DEM by fusion shape from shading and interferometry bathymetries for seafloor remote sensing" *Conference of SPIE*, pp. 204-215, Barcelona, Spain, 2004.

[35] W. Y. Zhao and R. Chellappa "Illumination-insensitive face recognition using symmetric shape-from-shading." Proc. of the *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 286-293, Hilton Head Island, SC, 2000.

[36] A. Tankus, N. Sochen, and Y. Yeshurun, "Reconstruction of medical images by perspective shape-from-shading," *Conference on Pattern Recognition*, pp. 778-781, Cambridge, UK, 2004.

[37] M. Seibert and A.Waxman. "Adaptive 3-D object recognition from multiple views," *IEEE Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 107-124, 1992.

[38] D. Kriegman, "Computing stable poses of piecewise smooth objects," *Computer Vision, Graphics and Image Processing*, Vol. 55, No. 2, pp. 109-118, 1992.

[39] S. Petitjean, "The enumerative geometry of projective algebraic-surfaces and the complexity of aspect graphs," *Journal of Computer Vision*, Vol. 19, No. 3, pp. 261-287, 1996.

[40] K. M. Lee and C. C. Kuo, "Shape from shading with perspective projection," *CVGIP: Image Understanding*, Vol. 59, No. 2, pp. 202-212, 1994.

[41] K.M. Lee and C.C. Kuo, "Shape Reconstruction from Photometric Stereo," *Computer Vision and Pattern Recognition*, pp. 479-484, Champaign, IL, 1992.

[42] Alex P. Pentland, "Linear shape from shading," *International Journal of Computer Vision*, Vol. 4, No. 2, pp. 153-162, 1990.

[43] P. Devijver and J. Kittler, *Pattern recognition – a statistical approach*, Englewood Cliffs, New Jersey Prentice-Hall, 1982.

[44] A. Bakhtari, M. Eskandari, M. Naish, and B. Benhabib, "A multi-sensor surveillance system for active-vision based object localization," *Proc. of the IEEE Int. Conf. System, Man and Cybernetics*, pp. 1013-1018, Washington, DC, 2003.

[45] K. Thakur, F. Cheng, and K.T. Miura, "Shadow generation using discretized shadow volume in angular coordinates," *IEEE Computer Graphics and Applications Conf.*, pp. 224-233, Canmore, Canada, 2003.

[46] B. Huet, E. R. Hancock, "Shape recognition from large image libraries by inexact graph matching," *Pattern Recognition in Practice*, pp. 1259-1269, Vlieland, Netherlands, 1999.

[47] S. A. Nene, S. K. Nayar, and H. Murase. Columbia Object Image Library *(coil-100)*. *Technical Report CUCS-006-96*, Columbia Univ., 1996. Available online at: http://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php, viewed 6 March 2006.

[48] Q. Zheng and R. Chellappa. "Estimation of illuminant direction, albedo, and shape from shading," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 7, pp. 680-702, 1991.

[49] K. M. Lee and C. C. J. Kuo, "Shape from shading with a linear triangular element surface model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 8, pp. 815-822, 1993.

[50] E. Rouy and A. Tourin, "A viscosity solutions approach to shape from shading," *SIAM Journal on Numerical Analysis*, Vol. 29, No. 3, pp. 867-884, 1992.

[51] J. Oliensis, "Shape from shading as a partially well-constrained problem," *Computer Vision, Graphics, and Image Processing: Image Understanding*, Vol. 54, pp. 163-183, 1991.

[52] A. P. Pentland, "Local shading analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, pp. 170-187, 1984.

[53] P.S. Tsai and M. Shah, "Shape from shading using linear approximation," *Image and Vision Computing Journal*, Vol. 12, No. 8, pp. 487-498, 1994.

[54] O. Ikeda, "Use of four surface normal approximations and optimization of light direction for robust shape reconstruction from single images," *Proc. of the IEEE Canadian Conference on Computer and Robot Vision*, pp. 84-91, London, Canada, 2004.

[55] H. de Ruiter and B. Benhabib, "*Tracking* of rigid bodies for autonomous surveillance," Proc. of the *IEEE Int. Conf. on Mechatronics and Automation*, Niagara Falls, Canada, July, 928-933, 2005.

[56] R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, "Algorithms for cooperative multisensor surveillance," *Proc. of the IEEE,* Vol. 89, No. 10, pp. 1456-1477, 2001.

Ardevan Bakhtari received the B.A.Sc degree and PhD degree in mechanical engineering from the University of Toronto, Toronto, ON, Canada. His research interests are in the area of mechatronics, including active-vision, multi-sensor surveillance, and agent-based control.

Beno Benhabib is a Professor in the Department of Electrical and Computer and Mechanical and Industrial Engineering at the University of Toronto, Toronto, ON, Canada. His main research interests are in the area of robotics and automation, primarily for manufacturing. He is the author of the book, *Manufacturing: Design, Production, Automation and Integration* (Marcel Dekker, 2003) and more than 240 articles in international conferences and journals. In the past two decades, he has supervised the research work of over 90 postdoctoral fellows and graduate students.