Sensor Planning and Control in a Dynamic Environment

John R. Spletzer Camillo J. Taylor
GRASP Laboratory – University of Pennsylvania
Philadelphia, PA 19104, USA
{*spletzer, cjtaylor*}*@grasp.cis.upenn.edu*

Abstract

This paper presents an approach to the problem of controlling the configuration of a team of mobile agents equipped with cameras so as to optimize the quality of the estimates derived from their measurements. The issue of optimizing the robots' configuration is particularly important in the context of teams equipped with vision sensors since most estimation schemes of interest will involve some form of triangulation.

We provide a theoretical framework for tackling the sensor planning problem and a practical computational strategy, inspired by work on particle filtering, for implementing the approach. We extend our previous work by showing how modeled system dynamics and configuration space obstacles can be handled. These ideas have been demonstrated both in simulation and on actual robotic platforms. The results indicate that the framework is able to solve fairly difficult sensor planning problems online without requiring excessive amounts of computational resources.

1. Introduction

The idea of using teams of small, inexpensive robotic agents to accomplish various tasks is one that has gained increasing currency in the field of robotics research. Figure 1 shows a picture of a Clodbuster robot which is based on a standard remote controlled motion platform and outfitted with an omnidirectional video camera – its only sensor. Using teams of these modest robots, fairly sophisticated applications such as distributed mapping, formation control and distributed manipulation have been successfully demonstrated [1, 2].

One of the more interesting aspects of these platforms is that estimates for relevant quantities in the world are formed by combining information from multiple distributed sensors. For example, the robots in the team shown in Figure 1 obtain an estimate for their relative configuration by combining the angular measurements obtained from all of the omnidirectional images and performing a simple triangulation operation.

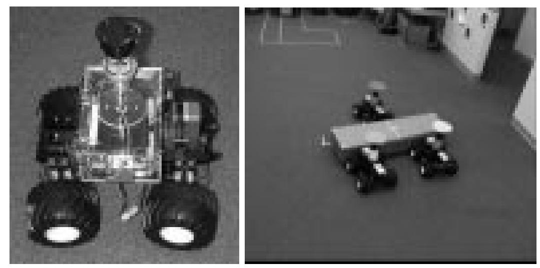

Figure 1: A single Clodbuster robot (left) and the team performing a distributed manipulation task.

Similar techniques can be used to estimate the locations of other features in the environment. In fact, one could choose to view the team as a three-eyed stereo rig where the individual eyes can actually be moved on the fly.

This capability invites the following question: given that the robot platforms are mobile, how should they be deployed in order to maximize the quality of the estimates returned by the team? This is a particularly important question in the context of robots equipped with vision sensors since most of the estimation techniques of interest in this case are based on some form of triangulation.

Similar questions arise when one considers the problem of integrating information from a sea of distributed sensors. Given that there is some cost associated with transmitting and processing data, which sensor readings should one use to form an estimate for the parameters of interest?

This paper presents a theoretical framework for discussing such questions and a practical computational approach, inspired by work on particle filtering, for tackling them. The suggested approach could be viewed as an application of the theory of games since the problem of controlling the robots' configuration is reformulated as the problem of optimizing a quality function that reflects the expected value of assuming a particular formation. Results obtained by applying this approach to practical problems are presented in Section 3. In this paper, we extend our previous work [3] in two important ways by showing how system dynamics can be handled and how obstacle avoidance can be incorporated.

It is important to note that while the approach was developed to handle the problems faced by teams of robots equipped with vision sensors, it could also be used to deploy robots equipped with other types of sensors like laser range finders or sonar systems.

1.1. Related Work

The problem of controlling sensors to optimize information gathering was considered by Bajcsy and others under the heading of Active Perception [4]. This involved fusing data from both homogeneous and heterogeneous dynamic sensors to improve various performance metrics that included ranging accuracy [5]. In this vein, our framework can be viewed as an extension of the active perception paradigm to the field of distributed mobile robots.

A significant amount of research has been directed to the problems associated with getting teams of robots to cooperate on high level tasks such as distributed manipulation, exploration and mapping [6, 7, 8]. However, far less emphasis has been placed upon optimizing the team's collective sensing capabilities. Perhaps most relevant to our approach was a methodology for distributed control proposed by Parker [9], which maximized the observability of a set of moving targets by a team of robots. In this scheme, the objective was maximization of the collective time that each target was observable by at least one robot. The accuracy of target pose estimates was not considered.

The theory of games has also provided inspiration for similar research in target tracking. The pursuit-evasion problem was investigated by LaValle *et al* [10]. They presented motion planning strategies that maximized the probability of keeping sight of a target as it moved through a field of obstacles. Results were limited to the case of a single pursuer/evader. Hespanha *et al* also investigated the pursuit-evasion problem, but from a multi-agent perspective [11]. They proposed a greedy approach to control a group of agents so as to maximize the probability of finding one or more evaders. In both cases, the focus was on locating and/or tracking one or more evaders. The quality of the estimates for target position was again not investigated.

In the Next Best View (NBV) problem, sensor placement is of primary concern [12, 13]. Given, for example, previous range scans of an object, an NBV system attempts to determine the next best position of the scanner for acquiring the object's complete surface geometry. As in our framework, the emphasis is optimizing sensor placement. However, NBV is intended for use in a static environment. Inherent in our approach is the ability to handle dynamic scenes which makes it more akin to a control law for distributed sensors.

2. Theoretical Approach

This section describes the theoretical framework that will be used to discuss the problem of sensor deployment. In order to ground the terminology, we will describe how various elements in the framework would relate to the scenario depicted in Figure 2. In this example, three robots are tasked with localizing a moving target.

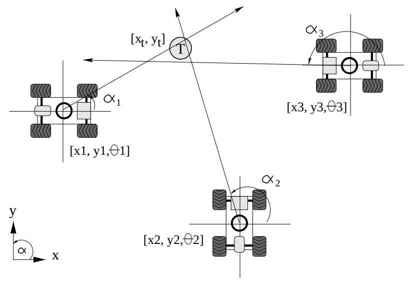

Figure 2: Target localization by a robot team.

Let $\mathcal{C}_r$ denote the configuration space of the robotic platforms. In this case, one can consider the set of vectors formed by concatenating the positions and orientations of the three platforms with respect to the base frame of reference $(x_1, y_1, \theta_1, x_2, y_2, \theta_2, x_3, y_3, \theta_3)$. Let $\rho \in \mathcal{C}_r$ denote an element of this configuration space.

Similarly let $\mathcal{C}_w$ denote the configuration space of the parameters under consideration. In Figure 2 this space is particularly simple since we need only consider the position of the moving target with respect to the base frame denoted by the vector (x_t, y_t). In general, however, this space can be much more complicated. Let $\omega \in \mathcal{C}_w$ denote an element of this configuration space.

Let $\hat{z}$ denote the measurements obtained by the robot team. For this example the vector formed by concatenating the three angles measured by the robots $(\alpha_1, \alpha_2, \alpha_3)$ serves this purpose. The hat serves to remind us that these measurements are corrupted by noise. In the sequel it will be assumed that the designer has some model for or bounds on the noise process.

Let $Est(\rho, \hat{z})$ denote a function which can be used to produce an estimate for the configuration of the world, $\hat{\omega}$, from the noisy measurements, $\hat{z}$, and the robots configuration, ρ. $Disp(\omega, \hat{\omega})$ is a function which returns a scalar value indicating the expected disparity between the estimated value $\hat{\omega}$ and the actual value ω. This value will depend upon the distribution of errors on $\hat{z}$.

$P(\omega)$ denotes a probability density function on the configuration space $\mathcal{C}_w$ which can be used to model prior information about the values of the parameters of interest. For example, one may have some information about where the target could be based on prior measurements.

Given this terminology, one can define a quality function $Q(\rho)$ as follows:

$$Q(\rho) = \int_{\mathcal{C}_w} Disp(\omega, Est(\rho, \hat{z}))P(\omega)d\omega \quad (1)$$

This function captures how the expected error in the estimate, $\hat{\omega}$, varies as the robots configuration changes.

Note that there are, of course, several alternative definitions for this quality function that are equally reasonable. One could consider the maximum expected error in the estimate or the median expected error. Different choices for the Q function may be more appropriate in certain situations.

With these notions in place, one can formulate the problem of choosing an appropriate configuration for the robots as an optimization problem as shown below.

$$\min_{\rho \in \Delta} Q(\rho) \quad (2)$$

The goal in this case is to find a choice of $\rho \in \Delta$, where $\Delta \subset \mathcal{C}_r$, which minimizes the quality function $Q(\rho)$. Limiting the optimization to a subset of $\mathcal{C}_r$, Δ, allows us to model situations where certain configurations cannot be achieved due to obstacles in the environment, sensor constraints or limitations on the range of motion of the robots.

Note that the framework is general enough to be applied to a wide range of sensor planning problems. The specifics of the task would be reflected in the definitions of $\mathcal{C}_r$, $\mathcal{C}_w$, $\hat{z}$, Est and $Disp$. Specific instances of this framework will be discussed in Section 3.

3. Computational Approach

For most interesting systems the optimization problem given in equation 2 is difficult to solve analytically. It is however, possible to approximate this process computationally. To do this we draw inspiration from prior work on particle filtering [14].

In particle filtering, probability distributions such as $P(\omega)$ are approximated by sets of tuples (ω_j, π_j), where ω_j is a single sample from $\mathcal{C}_w$ and π_j a weight that reflects the likelihood of ω_j representing the state ω. By making use of this approximation, we can replace the integral of equation 1 with a weighted summation.

$$Q(\rho) \approx \sum_j Disp(\omega_j, Est(\rho, \hat{z}))\pi_j \quad (3)$$

Recall that the proposed technique is intended for use in online applications where the robot team has an evolving estimate for the state of the system being observed and the objective is to determine how the robots should move in order to improve the quality of this estimate at the next time instant. In this context, the maximum velocities of the robots serve to limit the configurations that need to be considered and the current configuration of the team serves as a natural starting point for the optimization procedure.

One simple but effective approach to optimizing the robot configuration is to first approximate the gradient of the quality function, $\frac{\partial}{\partial \rho} Q(\rho)$, by sampling its value in the vicinity of the current robot configuration. The controller then moves the robot configuration in the direction indicated by this gradient. Alternatively one could employ standard optimization techniques, like the simplex method [15] to choose the best achievable robot configuration in the vicinity for the next time instant.

Note that it is possible to incorporate knowledge of the dynamics of the system into this framework by projecting the set of particles used to represent the distribution $P(\omega)$ through the dynamic model in the usual manner as described by Isard and Blake [14]. One can then use this particle distribution to approximate the quality function $Q(\rho)$ (see Eqn. 3), and consequently to control the motion of the robot team.

Our previous work demonstrated how teams of robots could use the framework to optimally track the position and orientation of multiple, unpredictable targets [3]. Here we show examples of how it can be extended to include modeled system dynamics and workspace obstacles.

3.1. Incorporating the Dynamical Model

Integrating target dynamics into sensor planning often provides significant improvements in tracking performance. Dynamical models can be obtained using an approximation of target dynamics, or through "learned" models as demonstrated in [14]. For our simulations, we employed the former approach.

Consider the case of n observers on the ground tracking a ball traveling through the air with some unknown initial velocity V_t. We model these observers as robots equipped with omnidirectional cameras. In this case, $\mathcal{C}_r$ represents the concatenation of the robot positions which are constrained to operations in the x-y plane, $\mathcal{C}_w \subset R^3$ represents the space of target positions. The measurement vector $\hat{z}$ denotes the n azimuth and elevation angle pairs to the target measured by members of the robot team. We assume $\hat{z}$ to be corrupted with random bounded noise generated from our sensor model. $Est(\rho, \hat{z})$ returns an estimate for the target position, $\hat{\omega}$, which minimizes the squared disparity with the measurements, $\hat{z}$, and $Disp(\omega, \hat{\omega})$ simply returns the Euclidean distance between the estimated target position and the actual value.

We approximated the dynamical model for the ball by assuming constant acceleration under gravity, and estimated its velocity from position measurements over time. Actual ball dynamics in the simulation were slightly more realistic, and also approximated drag effects using a Newtonian model.

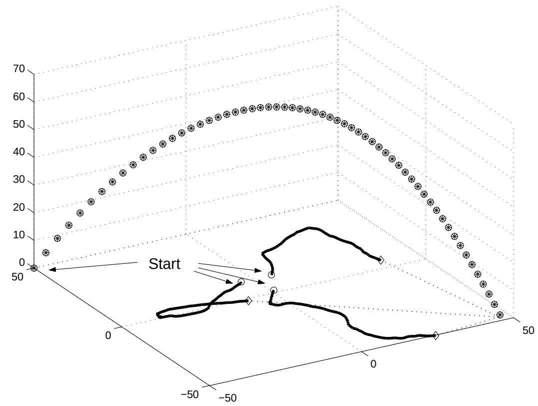

Figure 3: Ground observer trajectories optimally tracking an aerial target.

Since our sensor noise model is assumed bounded, $P(\omega)$ was initially approximated from a randomly generated set of exemplars that were constrained to lie within the intersection of the sensors' error cones and all of the samples were given equal weight. The distribution was then propagated using standard particle filtering techniques. In our simulations, robot motions were constrained by the maximum robot velocity $V_r << V_t$. This served to define the limits of the set over which the optimization occurs, Δ. Results from a sample Matlab simulation for three robots are provided below. For this trial, 100 exemplars were used to approximate $P(\omega)$, and the sensor model was assumed to be bounded Gaussian noise of $\pm 5°$ with $\sigma = 1°$.

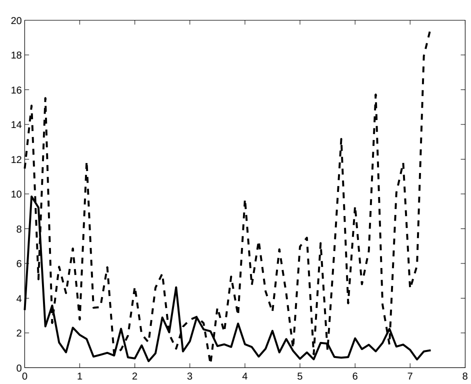

Figure 4: Measurement errors from stationary (dashed line) and moving (solid line) robot observers. Reductions in the latter case are significant across the entire target trajectory.

Figure 3 shows a representative simulation run of three robots tracking a single target. Robot trajectories are inefficient from a "distance-traveled" viewpoint, as they attempt to optimize position estimates over the target's entire flight rather than its endpoint. Figure 4 shows the error in measured target position for the same target trajectory from both stationary (dashed line) and moving (solid line) observers. When viewed in this light, the benefits of the otherwise curious robot trajectories become readily apparent. Reductions in measurement errors by a factor of 4-5 over the stationary case clearly demonstrate the effectiveness of the integrated optimization/dynamical modeling approach.

3.2. Tracking targets in a cluttered workspace

In the simulation results we have presented thus far, constraints to C_r were limited solely to pursuer dynamics and a mandatory target standoff distance. This is adequate for operations in an uncluttered workspace, but does not handle the more generic case where obstacles are present. To address the resulting additional constraints on C_r (and C_w), we assumed that the robots were able to obtain accurate pose information for obstacles in their immediate vicinity. This was consistent with our approach of generating locally optimal trajectories, and did not require *a priori* information of obstacle locations or a global map of the environment.

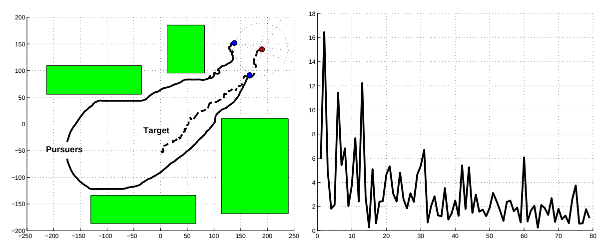

Figure 5: Tracking a point target in a cluttered environment. Significant reductions to target position error were still realizable even in the presence of obstacles.

Next, we applied standard motion planning techniques for collision avoidance in this local neighborhood. Obstacles exhibited a repulsive vector field in their immediate vicinity with an area defined as a function of robot speed, and a magnitude proportional to the separation distance to the obstacle. Once locally optimal trajectories were generated for tracking the target, desired velocity vectors were combined with the obstacle-induced repulsive force vectors to obtain a compromise trajectory for the pursuer robots. A representative simulation trial can be found in Figure 5.

While the presence of obstacles in this example constrained the robots' motion, the control law automatically adjusted their trajectories in order to compensate for these limitations and provide the best possible state estimates.

3.3. Experiments with the Clodbusters

The proposed framework has been implemented on our team of Clodbuster robots which use omnidirectional vision as their sole sensing modality. In these experiments, a pair of robot pursuers was tasked with tracking a third robot which played the role of a moving target. Two sets of trials were conducted to demonstrate operations in both cluttered and uncluttered environments. A picture of the robot team used for these trials can be seen in Figure 6.

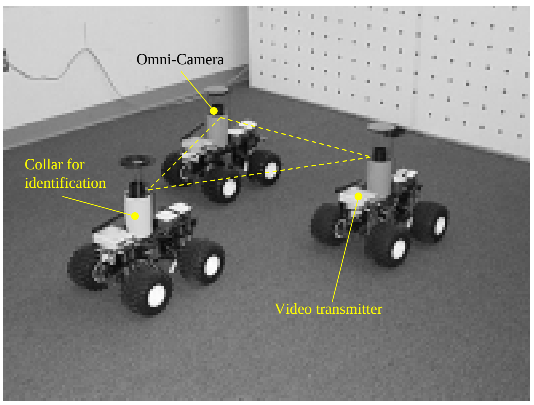

Figure 6: Clodbuster team used for experiments.

Each of the robots was fitted with a colored cylindrical color which yielded a $360°$ symmetrical target about its optical axis. A color extractor operating in YUV space was used to isolate these targets. The pursuers used these measurements to localize each other and to estimate the target's position. The complete localization process ran at a rate of 15Hz.

Figure 7: Trajectory for two pursuer robots tracking a moving target robot in an obstacle-free environment.

In the cluttered workspace trials, it was also necessary for the pursuer robots to estimate the position of obstacles. This was accomplished by generating a rangemap from the omnidirectional image to features in the environment as outlined in our previous work [16]. Target and pursuer robots were then discriminated from obstacles using their relative pose as determined during the localization phase.

For the sake of experimental expediency, the sensor model assumed that the angular measurements obtained by the robots were corrupted with additive errors drawn from a normal distribution with a variance of $\sigma = 0.5°$. This was based upon several thousand measurements from numerous representative static team poses. In truth, the statically measured values were typically lower ($\sigma = 0.1$-$0.3°$). However, we expect dynamic levels to be higher and increased σ accordingly. Experimental implementation followed closely with that used in the corresponding simulation experiment. Derivative estimation techniques were used to approximate the gradient of the Q function for optimizing the pursuers' headings. The maximum robot speed and a prescribed standoff distance served to define Δ for a given time-step. For the cluttered workspace trials, obstacles exhibited repulsive forces when the separation was less than 1 meter. Using 100 particles to approximate the probability $P(\omega)$ over the target configuration space, we were able to compute locally optimal robot configurations at a rate of 15Hz.

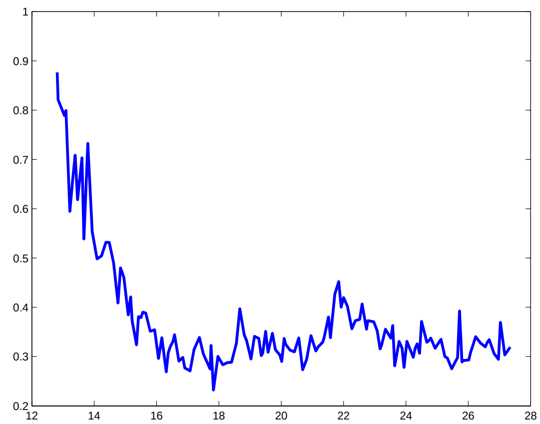

Figure 8: Estimated RMS position error (cm) vs. time for the single target case.

A representative trial from our obstacle-free experiments is shown in Figures 7 and 8. The former shows a series of images from an overhead view of the scene, while the latter shows the corresponding position error estimates. Both the trajectory and the dramatic drop in the error estimate correlate well with the corresponding simulation results presented previously [3].

Figures 9 and 10 show the corresponding trial for a cluttered workspace. The effect on the motion of the right pursuer robot was significant. In contrast to the obstacle-free case, its motion was constrained to a much narrower region. However, the control scheme automatically adjusted the path of the left pursuer to compensate for this limitation. As a result, the estimated target tracking error still fell dramatically.

Figure 9: Trajectory for two pursuer robots tracking a moving target robot in a cluttered workspace. The left pursuer adapts its trajectory to the right pursuer's mobility constraints.

It should again be noted that no explicit controllers were needed for maneuvering the formation. Trajectories were implicitly generated by the Q function which captured the notion of a good configuration. Additionally, as implemented the computational complexity of this framework scales linearly with both the number of targets and the number of robots, making it well suited for distributed, multi-robot applications

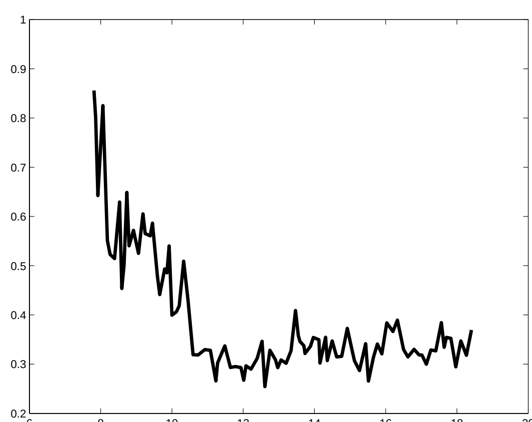

Figure 10: Estimated RMS position error (cm) vs. time for the single target case with obstacles. Results are comparable to the obstacle-free case.

4. Conclusions

This paper presents an approach to the problem of controlling the configuration of a team of mobile agents so as to optimize the quality of the estimates derived from their measurements. We provide a theoretical framework for tackling the sensor planning problem, and a practical computational strategy for implementing the approach while accounting for both model system dynamics and obstacles in the environment. The ideas have been demonstrated both in simulation and on an actual robotic platform, and the results indicate that the system is able to solve fairly difficult sensor planning problems online without requiring excessive amounts of computational resources.

Future work will investigate the issues involved in applying the framework to scenarios involving occluding obstacles and to teams of robots with heterogeneous sensing capabilities.

Acknowledgments : This material is based upon work supported by the National Science Foundation under a CAREER Grant (Grant No. 9875867) and by DARPA under the MARS program.

References

[1] R. Alur et al, "A framework and architecture for multirobot coordination," in *Proc. Seventh International Symposium on Experimental Robotics*, Honolulu, Hawaii, Dec. 2000.

[2] J. Spletzer et al, "Cooperative localization and control for multi-robot manipulation," in *International Conference on Intelligent Robots and Systems*, Maui, Hawaii, Oct 2001.

[3] J. Spletzer and C.J. Taylor, "A framework for sensor planning and control with applications to vision guided multi-robot systems," in *Computer Vision and Pattern Recognition Conference*, Kauai, Hawaii, Dec 2001.

[4] R. Bajcsy, "Active perception," in *Proceedings of the IEEE, Special Issue on Computer Vision*, August 1988, vol. 76-8, pp. 996–1005.

[5] E. Krotkov and R. Bajcsy, "Active vision for reliable ranging: Cooperating focus, stereo, and vergence," *International Journal of Computer Vision*, vol. 11-2, pp. 187–203, Oct 1993.

[6] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceeding of the IEEE Int. Conference on Robotics and Automation*, San Francisco, CA, April 2000, pp. 476–481.

[7] M. Mataric, "Issues and approaches in the design of collective autonomous agents," *Robotics and Autonoumous Systems*, vol. 16, no. 2-4, pp. 321–331, Dec 1995.

[8] D. Rus, A. Kabir, K. Kotay, and M. Soutter, "Guiding distributed manipulation with mobile sensors," in *Multi-Sensor Fusion and Integration*, 1996.

[9] L. Parker, "Cooperative robotics for multi-target observation," *Intelligent Automation and Soft Computing*, 1999.

[10] S. LaValle, H. Gonzalez-Banos, C. Becker, and J. Latombe, "Motion strategies for maintaining control of a moving target," in *Proceeding of the IEEE Int. Conference on Robotics and Automation*, Albuquerque, NM, April 1997, pp. 731–736.

[11] J. Hespanha, H. Kim, and S. Sastry, "Multiple-agent probabilistic pursuit-evasion games," in *Proceedings of the 38th Conf. on Decision and Control*, Phoenix, AZ, December 1999, pp. 2432–2437.

[12] R. Pito, "A solution to the next best view problem for automated surface acquisition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 1999, vol. 21.

[13] I. Stamos and P. Allen, "Interactive sensor planning," in *Computer Vision and Pattern Recognition Conference*, Santa Barbara, CA, June 1998, pp. 489–495.

[14] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29-1, pp. 5–28, 1998.

[15] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1993.

[16] A. Das et al, "Real-time vision based control of a nonholonomic robot," in *Proceeding of the IEEE Int. Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 1714–1719.