A Framework for Sensor Planning and Control with Applications to Vision Guided Multi-robot Systems.

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. The 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.

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.

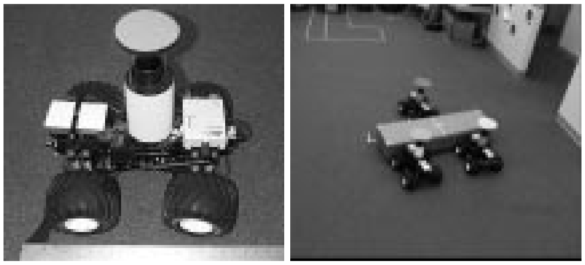

Figure 1: A single Clodbuster robot (left) and the team performing a distributed manipulation task.

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 4.

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 [3]. This involved fusing data from both homogeneous and heterogeneous dynamic sensors to improve various performance metrics that included ranging accuracy [4]. 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 [5, 6, 7]. 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 [8], 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* [9]. 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 [10]. 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 [11, 12]. 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 Figure 2. In this example, three robots are tasked with localizing a moving target.

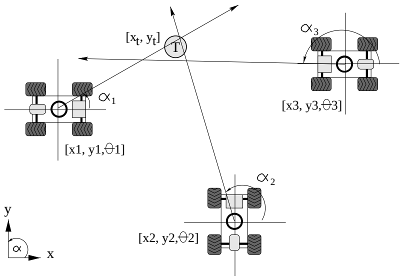

Figure 2: Target localization by a robot team.

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:

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

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.

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

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 4.

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 [13].

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.

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

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 [14] 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 observed 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 [13].

4. Experimental Results

4.1. Simulation Experiments

In order to demonstrate the utility of the proposed framework, we have applied it to three sensor planning problems in simulation: tracking a single point target, tracking multiple point targets, and tracking a box. Each of these scenarios is explained in more detail below.

4.1.1 Tracking a single point target

For the first scenario we consider n robots equipped with omnidirectional cameras, and tasked with tracking a single static or randomly moving point target as shown in Figure 2. $\mathcal{C}_r$ represents the concatenation of the robot positions, $\mathcal{C}_w$ the target position space, and $\hat{z}$ the n angles to the target measured by the 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.

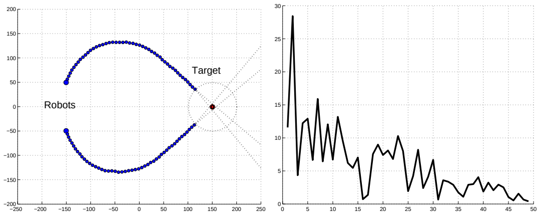

Figure 3: Generated trajectories (left)and disparity measurements for 2 robots tracking a static point target.

Since our sensor noise model is assumed bounded, $P(\omega)$ is approximated from a randomly generated set of exemplars that are constrained to lie within the intersection of the sensors' error cones and all of the samples are given equal weight.

In our simulations, robot motions are constrained by the maximum robot velocity and the robots positions are limited by mandating a minimum standoff distance to the target. These serve to define the limits of the set over which the optimization occurs, Δ. Results from Matlab simulations for two robots with both static and dynamic targets are provided below. For these trials, 100 exemplars were used to approximate $P(\omega)$, and the sensor model (for all trials) was assumed to be bounded Gaussian noise of $\pm 5^{\circ}$ with $\sigma = 1^{\circ}$.

Figure 3 shows the static target case for two robots. Trajectories for this symmetric case are predictable and consistent with simulation results, as are the dramatic drops in estimation error over time. Similar results are obtained for the case of an unpredictably moving target, as shown in Figure 4.

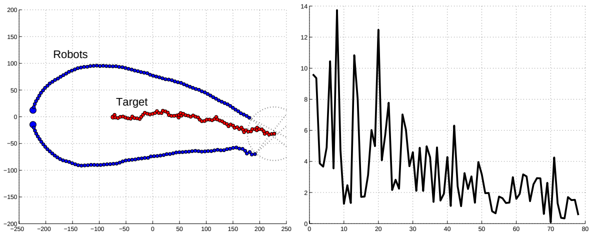

Figure 4: Generated trajectories (left) and disparity measurements for 2 robots tracking an unpredictable moving point target.

4.1.2 Tracking multiple point targets

For the second scenario, we examine the more interesting problem of n robots tracking m independently moving, unpredictable point targets. This problem can be tackled in much the same manner as the previous one. $\mathcal{C}_w$ now represents the concatenation of possible target positions, and $\hat{z}$ the corresponding $n \times m$ angles measured from robots to targets. $Est(\rho, \hat{z})$ approximates the position of every target, and $Disp(\omega, \hat{\omega})$ returns the sum of disparities between estimated and true target positions. Results from a pair of simulation runs can be found in Figures 5 and 6. In these trials, 3 unpredictable targets were tracked by 5 and 4 robots, respectively. In both runs, 25 exemplars were used for each target to approximate $P(\omega)$.

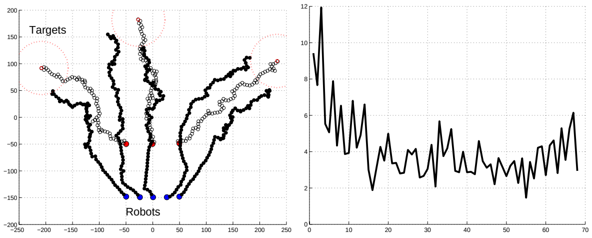

Figure 5: Generated trajectories and summed disparity measurements for 5 robots tracking 3 point targets.

Note the behavior of the robots as they move from their original positions to more advantageous vantage points. The robots automatically split off to track the targets without any need for sophisticated switching rules to decide which robots are assigned to which targets. The final configuration is simply a consequence of the definition of the Q function that the system attempts to optimize. Note also that it is not possible in these scenarios to assign two robots to every target so the robots distribute themselves automatically to come up with the best composite estimate.

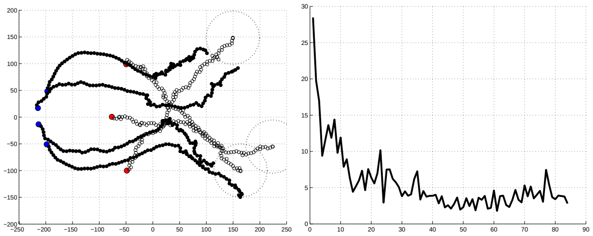

Figure 6: Generated trajectories and summed disparity measurements for 4 robots tracking 3 point targets.

4.1.3 Tracking a box

For the third case, we consider the problem of using the measurements from the robots to estimate the configuration of a box in the scene. This example demonstrates how the proposed framework can be applied to scenarios where the state estimate is not simply the concatenation of a set of point locations. Here the configuration space $\mathcal{C}_w$ is identified with $SE(2)$ and elements of this set denote the position and orientation of the box. The robots can measure the angles to all of the visible box corners, $\hat{z}$. The estimation function $Est(\rho, \hat{z})$ as always is nonlinear, and minimizes the disparity between the predicted angles to the visible corners and the actual measurements. $Disp(\omega, \hat{\omega})$ returns the sum of the distances between the recovered box corners and the actual corners. For these trials, 20 "box" exemplars were used to estimate $P(\omega)$. Sample simulation results can be found in Figures 7 and 8.

In both cases we can see that the robots not only migrate to positions more advantageous for tracking the corner features, but also for maximizing the *number* of visible features. The latter effect is a result of the Est function using only the visible corners to estimate the box pose. Inherently better estimates are obtained when more features are available.

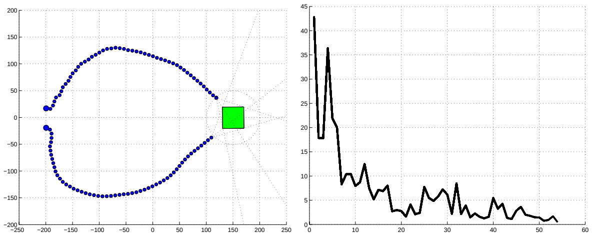

Figure 7: Generated trajectories and summed disparity measurements for 2 robots tracking a static box.

For the sake of simplicity we have assumed in all of these scenarios that the robots can accurately measure their position and orientation with respect to a global frame of reference. Note that we could consider error in the positioning of the robots within this framework by adding extra noise terms to the measurements or by including the robots configuration as part of the state to be estimated.

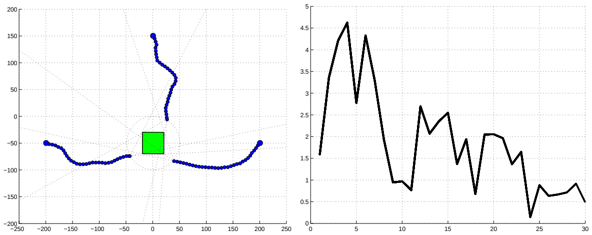

Figure 8: Generated trajectories and summed disparity measurements for 3 robots tracking a static box.

4.2. Experiments with the Clodbusters

The proposed framework has also been implemented on our team of Clodbuster robots. In this experiment, a pair of robot pursuers were tasked with tracking a third robot which played the role of a moving target. A picture of the robot team can be seen in Figure 9.

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

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.

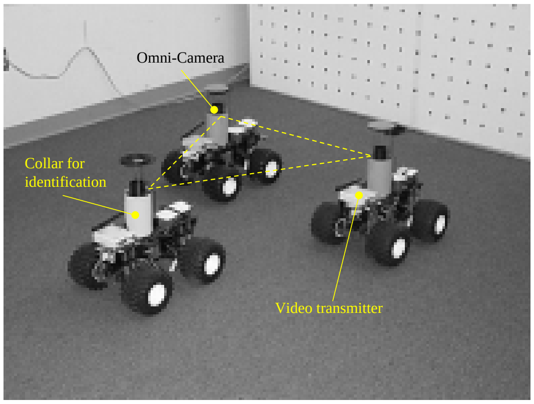

Figure 9: Clodbuster team used for experiments.

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. 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. A representative trial from our experiments is shown in Figures 10 and 11. Figure 10 shows a series of images from an overhead view of the scene. Figure 11 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 in Figure 4.

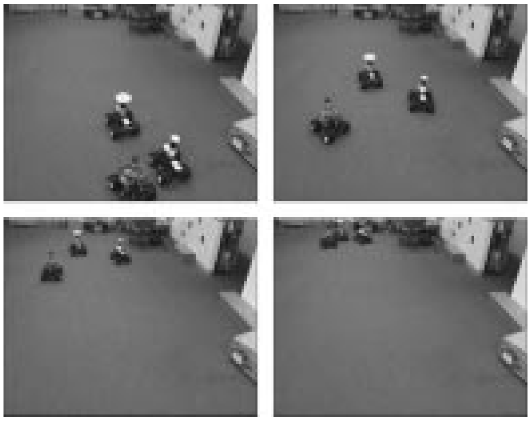

Figure 10: Trajectory for two pursuer robots tracking a moving target robot.

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

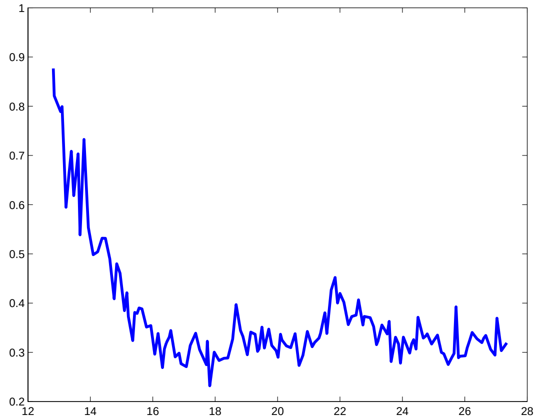

Figure 11: Estimated RMS position error (cm) vs. time for the single target case.

5. 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. 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 localizationa and control for multi-robot manipulation,” in *International Conference on Intelligent Robots and Systems*, Maui, Hawaii, Oct 2001.

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

[4] 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.

[5] 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.

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

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

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

[9] S. LaValle, H. Gonzalez-Banos, C. Becker, and J. Latombe, “Motion strategoes 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.

[10] 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.

[11] 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.

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

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

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