Proc. of IEEE International Conference
on Robotics and Automation, Anchorage, Alaska, 2010.

A General Information Quality Based Approach for Satisfying Sensor Constraints in Multirobot Tasks

Yu Zhang and Lynne E. Parker

Abstract— **Many architectures have been proposed to solve tightly-coupled multirobot tasks (MT) through coalitions of heterogeneous robots. However, several issues remain unaddressed. As coalitions are formed, sensor constraints among robots are also established. For example, in a leader-follower task, follower robots must keep leader robots within their sights, while in a box-pushing task, a supervisor robot needs to track the moving direction of the box and monitor the pushing path to the goal for obstacle avoidance. The question of how to keep these constraints satisfied during the entire execution, from initial configurations to completeness of the task, remains an open issue. In addition, environmental factors, both static and dynamic, can influence the maintenance of the constraints. Moreover, problems arise when the constraints are unsatisfiable given the current circumstances. For example, the sight of the leader might be blocked or there might be obstacles blocking the view of the box. This paper proposes a general method to address these issues for various applications with sensors having certain characteristics. Our approach combines the use of sensor models, environment sampling, measures of information quality, a motion model with sampling, and a constraint model. We believe that this approach offers the first generic formulation of robotic sensor constraints that can be applied to a wide variety of applications. To illustrate this method, we apply the approach to solve robot tracking and navigation tasks both in simulation and with physical robots. Experimental results illustrate the flexibility and robustness of the approach.**

I. INTRODUCTION

When the cooperation of multiple robots is required to complete a tightly-coupled task, the task is often referred to as a multirobot task (MT) [7]. Many architectures and algorithms (e.g., [12], [6], [17], [9]) have been proposed to solve these tightly-coupled tasks through coalitions. ASyMTRe [12] is one of the most recent architectures proposed which forms coalitions by reasoning about information required. Both ASyMTRe and our work are inspired by information invariants theory [5]. However, these architectures do not specifically model sensor constraints that are introduced due to the coalitions. As a result, controlling robot motions to enable the coalitions to maintain their respective constraints during task execution has not been addressed. In general, previous works provide a robot coalition for each multirobot task but do not give details on how the coalition is to fulfill the task. For example, a coalition might be proposed for a robot navigation task in which robots without a localization capability should follow a leader robot, keeping it in their sights. However, how the followers should start following from their initial positions to the goal is not answered. Likewise, for a box pushing task, these coalition approaches do not specify how the supervisor robot should track the box and monitor the pushing path at all time.

Although approaches using control theory can be used to address this issue, such approaches are application-specific, and do not provide a general solution to broad classes of problems. In other cases, it is often assumed that the robots are in proper relative configurations for certain coalitions so that sensor constraints are trivially maintained; for example, robots might be placed in a line so that each is within sight of the one immediately behind it, while the supervisor robot is assumed to be at a certain distance to the box so that it can keep track of the box and monitor the pushing path at the same time. Unfortunately, such assumptions are often not true, since robot initial configurations are usually not known with certainty. Furthermore, maintenance of sensor constraints can be influenced by static and dynamic environmental factors. For example, an obstacle or another moving robot may prevent robots from maintaining a line. Thus, the method also has to be flexible and robust enough for controlling robot motions to maintain sensor constraints to accomplish the assigned tasks. We also need to handle the extreme cases where certain constraints become unsatisfiable, such as when the sight of the leader robot or the view of the box is blocked. Our objective is to create a single, general method for all of these situations, and for a broad class of applications using sensors having certain characteristics.

In order to maneuver robots with sensor constraints through the environment, a mechanism is needed to quantify the utility of these constraints being satisfied, so that we can flexibly control robots while maintaining reasonable utilities. Inspired by information theory [13], we propose to use measures of information quality for this purpose. While the measure of uncertainty in information theory describes the information content, measures of information quality describe the reliability of the information. For example, when the leader robot goes near an obstacle, the information quality will decrease, since the sight of the leader robot can potentially be blocked; the information quality when the box is on the boundary of the field of view of the supervisor robot is less than if it is in the center, since the sight of the box can easily be lost. One of the most important advantages of this approach is that the general notion of information quality is not dependent on specific sensor types; this enables the separation of the sensor model from the application.

This work was supported in part by NSF under grant ISS-0812117.

Yu Zhang and Lynne Parker are with the Distributed Intelligence Laboratory in the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996-3450, USA, {yzhang51,parker}@eecs.utk.edu.

Another important aspect regarding information quality is that while most of the previous research has concentrated on identifying global measures using discretized methods [16], [3], [14], our approach is based on local measures in local sensor space. The reason for using local measures is based on the observation that sensor constraints must usually only be enforced locally instead of globally. For example, a follower robot only needs to know where the leader is relative to itself. The obvious tradeoff is that optimality is achieved locally instead of globally. However, given the property of localness of sensor constraints, this simplification should still yield sufficient solutions. One important advantage of computing local measures is that the approach can be completely distributively implemented since only local information is required. Another advantage is that measures of information quality can be defined continuously in space.

To account for the environmental influence on measures of information quality, we create an approximate geometric representation of the world in the sensor field of view (FOV) by sampling the environment into particles using range sensors. We compute weights for sensor quality measurements from the sensor quality model based on these particles. The weighted sensor quality measurements are taken as the information quality measurements. To use the measurements, we sample the motion space into motion vectors using a chosen motion model and compute the information quality measurement resulting from each vector. We then choose the motion vector that leads to the best information quality measurement. For extreme cases where certain sensor constraints are unsatisfiable, we define a constraint model that manages constraint repositories. Based on the type of the constraint, which depends on the kind of information the constraint concerns, the constraint model enables indirect constraint satisfaction for alternative solutions through constraint relaxation in an application-specific manner. The model is then used in combination with other models to compute the information quality measurements for alternative solutions.

After a brief review of related work (Section II), we explain our approach in detail (Section III). Afterwards, we apply the approach to solve robot tracking and navigation tasks, presenting results from simulations and physical robot experiments (Section IV). Finally, we conclude with some discussion and plans for future work (Section V).

II. RELATED WORK

Maintenance of sensor constraints is important in many applications. The approach of [8] handles the issue in a robot deployment task by preplanning a path that ensures the satisfaction of sensor constraints. However, such a solution is application-specific. Moreover, uncertainty and dynamic factors can render the solution infeasible for practical applications. Some other research works do not assume prior knowledge of the environment, resulting in applications that usually allow more flexibility and are adaptable to different environments. In [1], a greedy strategy for target tracking is introduced that uses local sensor data for geometric reasoning. Simulation results demonstrate significant improvement in tracking performance compared to previous works. The approach of [16] uses visual information to guide robots to cooperatively complete an insertion task based on information quality. During execution of the task, one robot provides visual guidance to another robot that is to finish the insertion job. The optimal observation position is found by first constructing a geometric world map and then searching the space for the position with the best information quality. In [15], robots in a target mapping task cooperate to maximize information gain. At each step, robots take the step to maximize the utility considering the observational contributions of teammates. However, these prior works solve only specific tasks; they do not provide general solutions to related tasks. In contrast, our work is aimed at providing a general method to multirobot sensor constraint satisfaction problems, which can be applied to a variety of applications.

The approach of [10] makes use of fuzzy logic to control the motions of cooperative robot teams using visual observation. However, fuzzy logic is beneficial primarily when mathematical models are not available. The information quality based approach that we propose defines explicit models and provides flexible methods to implement the models differently for various applications, hence offering more control over behaviors of robots.

In the case of unsatisfiable sensor constraints, while most of the works mentioned do not consider this issue, the approach of [16] deals with it by defining a secondary target. No general method has been provided to solve the problem. The approach that we propose defines a constraint model that models sensor constraints explicitly and further enables indirect constraint satisfaction for alternative solutions, based on the type of the constraint. The constraint model is also used to compute information quality measurements for these alternative solutions when original constraints are difficult or impossible to be satisfied.

III. THE APPROACH

For each constraint in a given application, we need to compute information quality measurements to characterize how the constraint is satisfied in a variety of configurations. Since the measurements are affected by only two independent factors, we break the computation into two parts. First, the sensor quality measurement is used to describe how reliable the information is that is being retrieved by the sensor; second, environmental influence is modeled as a weight to the sensor quality measurement to capture how the environment affects the reliability of the information. As a consequence, we can build sensor models that do not depend on task domains as long as the same type of constraint is concerned. To handle the application-specific environmental influence, we further notice that by breaking the influence onto environment samples, some extent of generality can be achieved at the cost of assuming independent samples. After obtaining the information quality measurements, we simply need to choose the motion vector that leads to the best information quality.

One limitation of the approach is that a natural definition of the sensor quality model, described shortly afterwards, is required for each sensor that is being modeled. Sensors for which the models are difficult or unnecessary to be defined, such as bumpers, are not proper to be modeled by the approach.

We next describe the different components of our approach and roles they play. For the following discussions, we use the term *constrainer* to represent the entity that imposes sensor restrictions and *constrainee* to represent the entity that is being restricted. The two terms could refer to the same entity, where the constraint is imposed on the *constrainer* itself.

A. Sensor Model

The separation of sensor model from the application ensures the generality of this approach. Once the model is defined for a particular type of sensor, it can be applied with little or no change to other applications in which that sensor type is used, as long as the same type of constraint is concerned. In our approach, we explicitly identify two essential submodels – the sensor quality model and the sensor uncertainty model.

1) Sensor Quality Model: The sensor quality model computes a sensor quality measurement, which is a score assigned to the constrainer's configurations in the constrainee's local space, given the current configuration of the constrainer[1]. The scores are assigned according to how reliable the corresponding configurations are for constraint satisfaction without considering environmental factors. For a particular sensor, and given different configurations for the constrainer from the viewpoint of the constrainee, the model computes normalized scores for the configurations based on their reliability in terms of constraint satisfaction. Note that configuration spaces can differ from constraint to constraint, from application to application. For example, in the navigation task, we consider only the relative coordinates of the constrainer while in the box pushing task, the configurations should include both the relative coordinates of the box and distance to the box. The model is defined as a function[2], $I_s : X \rightarrow [0, 1]$, in which X is the local space of the constrainee. A value of 1 indicates the most reliable configurations.

For the robot tracking and navigation tasks, the sensor quality model is defined for laser and camera sensors in a 2D Euclidean space,

$$I_s(x_{(l,\,\theta)}) = a * \frac{l_{max} - l}{l_{max}} + (1.0 - a) * \frac{\theta_{max} - |\theta|}{\theta_{max}} \quad (1)$$

in which the $(l,\, \theta)$ pair is the relative position of the constrainer in a polar coordinate system and l_{max}, θ_{max} are the sensor distance and angle ranges, respectively. Here, a is the weighting factor. Informally, this model prefers locations that are nearer to the sensor, and those that are nearer to the midline of sight of the sensor.

2) Sensor Uncertainty Model: The sensor uncertainty model captures the noise characteristics for a sensor, defined as a density function of sensor readings, $U_s : (X \mid X) \rightarrow \Re$.

For the robot tracking and navigation tasks, we define the model as linear normal distributions for laser and camera,

$$U_s(x_{(l',\,\theta')} \mid x_{(l,\,\theta)}) \sim N((l,\, \theta),\, M\Sigma_s M^T) \quad (2)$$

in which M is the scaling matrix and Σ_s is the covariance matrix for laser or camera,

$$M = \begin{bmatrix} |l - l'| & 0 \\ 0 & |\theta - \theta'| \end{bmatrix} \quad (3)$$

Although the sensor quality and sensor uncertainty models can be dependent on each other, for simplification, we assume independence between them. In most cases, the sensor uncertainty model is intrinsic to a particular sensor while the sensor quality model is primarily dependent on properties of the current task. When the two models are dependent, such as the robot tracking task, we have found that our approach still performs well. In ongoing work, we are exploring the use of other, perhaps more accurate, sensor uncertainty models.

B. Environment & Uncertainty Sampling

We introduce a sampling method to incorporate environmental influence on measures of information quality. First we represent different objects in the environment by considering them as composed of samples. Then, we compute the influence of these samples on the measures of information quality as dictated by the application. Generality is achieved by varying the influence of samples for different applications. One of the advantages of using a sampling method is that geometric reasoning is implicitly taken into account by an approximate geometric representation created by the samples. Another advantage is that sensor uncertainty can be easily incorporated by sampling again on these environment samples based on the uncertainty model of the range sensor using the Metropolis-Hastings algorithm [4], which can sample from any density function using a candidate generation function.

For sampling the environment, we apply a k-means clustering algorithm to range sensor readings, such that readings corresponding to different objects fall into different clusters. Based on the environment's complexity with respect to the robots, we then choose a granularity (i.e., density of particles) for sample creation. The created samples naturally form a geometric representation of the environment. For sensor uncertainty, we sample again on these samples based on the uncertainty model of the range sensor.

For each range sensor reading, environment sampling yields n samples, $S : \{s_1,\, s_2,\, ...,\, s_n\}$. For each environment sample s_i, after uncertainty sampling, we have a new set of samples, $S_i : \{s_i^1,\, s_i^2,\, ...,\, s_i^M\}$, for constant M. Figure 1(a) shows a scenario with M = 3.

C. Measures of Information Quality

Since samples from the constrainer itself usually have no impact on measures of information quality, we need

[1] In the following discussions, we concentrate on the model for a single constrainer. The model can be easily extended for multiple constrainers.

[2] In the following discussions, the subscript of 's' refers to a particular sensor.

to separately consider these samples. For each environment sample s_i, we compute the likelihood of the sample being the constrainer using the sensor uncertainty model for detecting the constrainer. The likelihood is computed as $\eta_i = U_s(x_{s_i} \,|\, x_{cr})/Z$, where x_{cr} represents the constrainer's configuration in the constrainee's local space and Z is a normalization constant. For the robot tracking and navigation tasks, we simply assume that samples within a short distance (e.g., $0.2m$) from the detected configuration of the constrainer are samples from the constrainer. Thus, we have $Z = U_s(x_{cr} \pm c \,|\, x_{cr})$, where c is a constant vector. Elsewhere, the probability is determined by a normal distribution based on the sensor uncertainty model.

The influence of the environment is accounted for in the form of weights for sensor quality measurements. The weights are computed in an application-specific manner to ensure that its complement captures the probability of risk imposed by the environment for constraint satisfaction given a configuration for the constrainer in the constrainee's local space. We maintain a probability measure for the weights so that the information quality measurements are also restricted to $[0, 1]$, thus describing the probabilities of reliability of the information. The uniformity separates the information quality measurements from the implementation details of the sensor model and environmental influence.

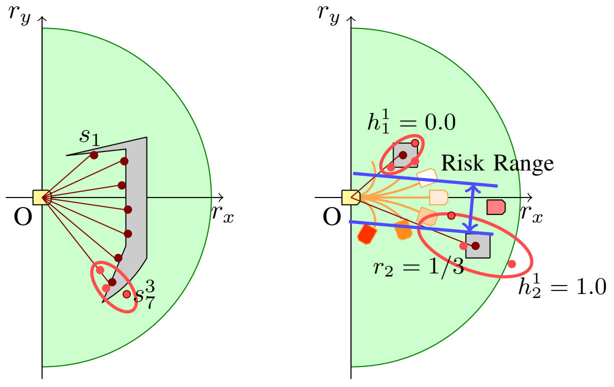
One important note is that we assume that samples exert influence on sensor quality measurements independently. The assumption is generally true unless geometric structures have to be specifically modeled (e.g., finding triangle shaped objects in the environment). For each $s_i^j \in S_i$, we compute $h_i^j = H_s(x_{s_i^j} \,|\, x_{cr})$, where $H_s : (X \,|\, X) \rightarrow [0, 1]$ is a function that captures the probability of risk considering uncertainty samples. The probability of risk considering an environment sample s_i is then computed as $r_i = C_{app}(h_i^1, h_i^2, \ldots h_i^M)$, where C_{app} is dependent on the application. A scenario is shown in Figure 1(b) where probability of risk is computed as the ratio of the uncertainty samples falling in risk range. Obstacle avoidance would be implicitly accomplished by computing environment influence in similar manners, since samples of the obstacle would increase the probability of risk towards the obstacle's direction.

Since samples from the constrainer are generally assumed to have no influence, the probability of risk for each sample s_i should be weighted by $1.0 - \eta_i$, where η_i is the probability of the sample being the constrainer. The weight for the sensor quality measurement is simply the joint probability of no risk considering all environment samples. As we assume independent samples, the final weight for the sensor quality measurement is $w = \prod_i (1.0 - r_i * (1.0 - \eta_i))$.

The final information quality measurement is simply computed as $\imath = \imath_s * w$, where $\imath_s$ is the sensor quality measurement.

D. Motion Model & Motion Sampling

The motion model is used to predict the resulting configuration given the current configuration and motion vector, $F_m : (P, V) \rightarrow P$. We use the common differential drive motion model, which has the form $r = v/\omega$, where r is the radius of movement.



(a)

(b)

Fig. 1. (a) Environment sampling using laser sensor with uncertainty sampling. Dark red particles are environment samples and bright red particles at the very bottom are uncertainty samples for one of the environment samples. (b) The probability of risk considering each environment sample is determined by the proportion of uncertainty samples falling in the risk range. The information quality measurements are shown graphically for several sampled motion vectors. The higher the resulting information quality, the whiter the resulting configuration is drawn. Notice that though the sensor quality model would prefer the motion vector facing the target, the combined information quality model would choose to curve a little to reduce the blocking risk from the nearest uncertainty sample.

Since measures of information quality can vary from application to application, methods for computing optimal solutions are generally impossible. To overcome the difficulty, we sample the motion space into motion vectors, $V : \{v_1, v_2, \ldots v_D\}$, and choose the one with the best resulting information quality measurement. Figure 1(b) explains the sampling and selection process in a simple scenario.

E. Constraint Model

The constraint model enables indirect constraint satisfaction for alternative solutions based on the type of the constraint. The model is then combined with other models to compute the information quality measurements for these solutions in case of unsatisfiable constraints. For example, in the robot navigation task, for successful maneuvering through a narrow hallway, constraints must be relaxed such that some follower robots switch from direct tracking of the leader to indirect tracking through other follower robots. For the box pushing task, the constraint for tracking the box should be relaxed by tracking the pusher robots to infer the box pushing direction when view of the box is blocked.

A graph representation is used for the constraint model in which nodes represent entities while arrows represent constraints among entities. The entities with incoming arrows are constrainers while entities with outgoing arrows are constrainees. For example, in the robot navigation task, robots with incoming arrows are leader robots and robots with outgoing arrows are follower robots. In the box pushing task, an arrow goes from the supervisor robot to the box.

In order to compute the information quality measurements for alternative solutions, we correlate it with the path of indirect satisfaction. Let's consider a robot navigation task with four robots. R_i is initially configured to follow R_j. However, due to occlusion, R_i is allowed to track R_j indirectly through R_{b1} and R_{b2}, indicated as follows: $R_i \rightarrow R_{b1} \rightarrow R_{b2} \leftarrow R_j$. Note that the sensors along the constraint path can be different as long as different sensor models are used. The information quality measurement for this alternative solution is computed as a function of local constraints along the path: $\imath_{R_i \rightarrow R_j} = L_{app}(\imath_{R_i \rightarrow R_{b1}}, \imath_{R_{b1} \rightarrow R_{b2}}, \imath_{R_{b2} \leftarrow R_j})$ in an application-specific way. The information quality measurements for local constraints are sent from robots in correlation with the constraints. In such a way, the constraint model remains transparent to various applications having constraints of the same type. Thus, generality is achieved. For the robot navigation task, we simply define L_{app} to be a function that multiplies all terms. For the box pushing task, the alternative solution is to track the pusher robots instead, otherwise the computation remains the same as tracking the box.

F. The Algorithm

The general algorithm for our information quality (IQ) based approach is given in Algorithm 1. The algorithm starts with environment sampling. Then, for every motion vector, the algorithm computes the resulting information quality measurement using the procedures described previously. For alternative solutions, we incorporate the path information provided by the constraint model and compute the final information quality measurement. If the measurement is less than a threshold, the algorithm searches for an alternative solution and repeats the calculation. The threshold can be dynamically adjusted to increase motion stability. The algorithm either returns a motion vector or failure when no alternative solution is found.

Algorithm 1 The General Algorithm for IQ Based Approach

```
sample the environment, $S : \{s_1, s_2, ..., s_n\}$
sample the motion vector, $V : \{v_1, v_2, ... v_D\}$
while true do
  find candidate constrainer configuration $x_{cr}$
  for all $s_i$ in $S$ do
    compute the likelihood $\eta_i = U_s(x_{s_i} \,|\, x_{cr})/Z$
    sample using the sensor uncertainty model of the
    range sensor, $S_i : \{s_i^1, ... s_i^M\}$
  end for
  for all $v_k$ in $V$ do
    predict new configuration $x_{cr}^k = F_m(x_{cr}, v_k)$
    compute $\imath_s^k = I_s(x_{cr})$
    for all $s_i$ in $S$ do
      for all $s_i^j$ in $S_i$ do
        compute $(h_i^j)^k = H_s(x_{s_i^j} \,|\, x_{cr})$
      end for
      compute $r_i^k = C_{app}((h_i^1)^k, (h_i^2)^k, ...(h_i^M)^k)$
    end for
    compute the weight for the sensor quality measure
    $w^k = \prod_i (1.0 - r_i^k * (1.0 - \eta_i))$
    compute $\imath^k = \imath_s^k * w^k$
  end for
  find $\imath^* = max_{i=1}^{D}(\imath^i)$
  if indirectly satisfied then
    compute $\imath^* = L_{app}(Path_{alt})$
  end if
  if $\imath^* >=$ a threshold then
    return $v^*$ (corresponding to $\imath^*$)
  end if
  search for alternative solution using constraint model
  if no alternative solution found then
    return failure
  end if
end while
```

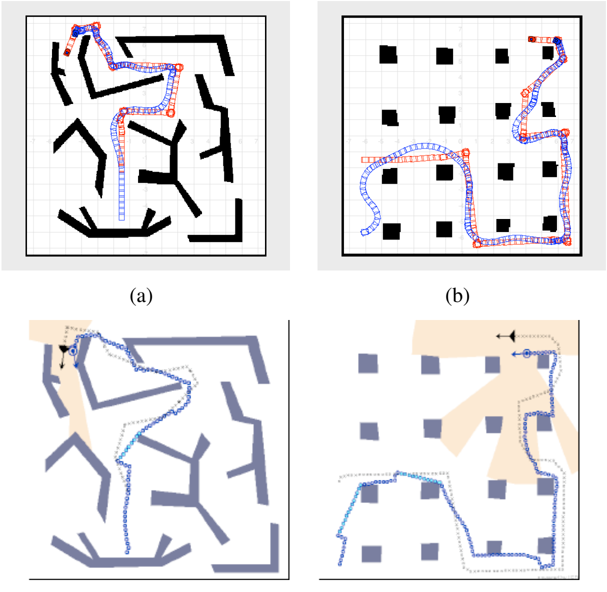
IV. EXPERIMENTS & RESULTS

We demonstrate our approach by applying it to two applications in simulation using Stage and with physical robots. In the robot tracking task, a tracking robot uses either a camera or a laser sensor to detect the target and tries to keep the target in its sight. While it is usually assumed that the maximum speed of the tracking robot is higher than the target, we set the maximum speed to be the same for the tracking robot and target as in [1]; in the robot navigation task, a teams of robots are to go to the same goal position. We often assume that only one of these robots has a localization capability (e.g., using GPS) and others can detect teammates using either a camera or laser sensor. The goal is for all the robots to reach the goal position. To enhance robustness, we implement a simple error recovery method for going to the last seen target or leader robot position when the target or leader is lost.

A. Simulations

We first compare our model for the tracking task with the specific approach of [1]. The approach of [1] uses range sensor readings for reasoning about the geometry of the environment. The information is then used to find the motion vector that minimizes the probability of losing the target. Their approach is, in some sense, optimal given only local information, and demonstrates intelligent behaviors for target tracking. Although we do not model the geometry of the environment explicitly as they do, our robots show certain behaviors similar to the ones in their work, e.g., swinging aside to decrease the future risk. Figure 2 shows the tracking trajectories for both approaches in two complex environments. In both environments, the robot starting from the bottom is the tracking robot. Compared with trajectories of simulations in [1], our simulations have smoother trajectories, which is due to the motion model we use. The tracking performance comparison is shown in Table I, in which the two approaches are clearly comparable. Statistical analysis for our IQ based approach is done using data collected over ten runs for each environment[3]. One

[3] Statistical analysis is not possible for the approach of [1] due to unavailability of the software platform on which it was run.



(a)

(b)

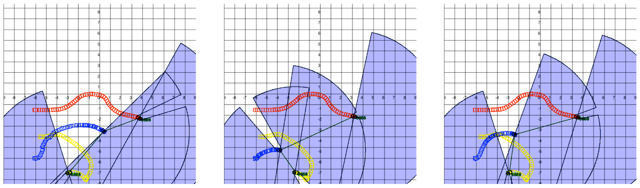
(c)

(d)

Fig. 2. (a) Tracking trajectory in Maze for our approach. (b) Tracking trajectory in City Block for our approach. (c) Tracking trajectory in Maze for the approach of [1]. (d) Tracking trajectory in City Block for the approach of [1].

advantage of our approach is that adapting our model to track multiple targets is straightforward simply by averaging the information quality measurements for multiple targets; while their approach can provide optimal solutions for each target, it is unclear how to combine the solutions for multiple targets. The most significant limitation of [1] is that it is application-specific while our approach can be extended to various applications.

This experiment shows that an approximate geometric representation is sufficient for capturing important characteristics of the environment. A robot utilizing this information can take proper actions to maintain sensor constraints. Applying our approach to the tracking task also shows that the approach works well for applications in which the sensor quality and sensor uncertainty models are highly correlated. The power of the information quality based approach lies in the fact that measures of information quality can be defined differently so that the resulting applications demonstrate different behaviors. To show these effects, we implement a multiple target tracking task with different sensor quality models. In Figure 3(a), we consider only distance in the model, in Figure 3(b) we consider only angle, and in Figure 3(c) we consider both. Notice the different behaviors demonstrated by the tracking robot (blue). For the model that considers only distance, the tracking robot clearly moves more in order to decrease the distances to the targets; for the model that considers only angle, the tracking robot moves little while turning more to minimize angle errors to the targets; for the model that considers both, the behavior is a combined solution and is more effective in terms of multiple target tracking, since targets are less likely to be out of sight at the next moment. Another interesting application would be dynamic robot formation, in which the sensor quality model is implemented to favor certain positions relative to other robot teammates.



(a)

(b)

(c)

Fig. 3. (a) Multiple target tracking with respect to distance only. (b) Multiple target tracking with respect to angle only. (c) Multiple target tracking with respect to both distance and angle.

To show robustness of the approach, we ran the robot navigation task with different initial configurations. We generate ten random configurations for follower robots with $x \in [-16.0, -12.0]$, $y \in [2.0, 3.8]$ and angle $\in [-60°, 60°]$. The robots are to maneuver through the environment shown in Figure 4 with these configurations; all three follower robots are initially configured to follow the leader (red). An obstacle is manually added for a clearer demonstration. For all configurations, the robots reconfigure using the constraint model at different times due to occlusions and reach the goal position successfully. Figure 4 gives snapshots for one of the initial configurations. Figures 4(e) and 4(f) are enlarged configurations from Figures 4(c) and 4(d), edited to show the sensor constraint satisfaction paths. We use the color of the constrainer as the color of the current satisfied constraints such that initial sensor constraints should all be colored red. Notice the changes in constraint satisfactions as the robots navigate through the environment. This result not only shows the robustness of our approach for different initial configurations, but it also shows that the constraint model adds an autonomous reconfiguration capability to robot teams by considering both static and dynamic environmental factors.

B. Physical Experiments

For the tracking and navigation tasks in physical experiments, we use the technique presented in [11] for constrainer detection using cameras. This technique uses specially designed markers for identifying robots carrying the marker and computing the relative positions using a camera sensor. Only the sensor model needs to be changed for this purpose. For robot tracking task, to determine if prediction of the information quality measurements is helpful, we construct a baseline approach for comparison which simply uses the vector field histogram method (VFH) [2], provided in Player and Stage, to drive the robot to the target while performing fast obstacle avoidance using laser sensor readings to greedily find the largest opening that the robot can go through. We allow some tolerance for angle error to the target, since otherwise the tracking robot would often be turning and unable to keep up with the target. For a fair comparison, the error recovery method is also added into the baseline approach. During execution of the task, the

TABLE I

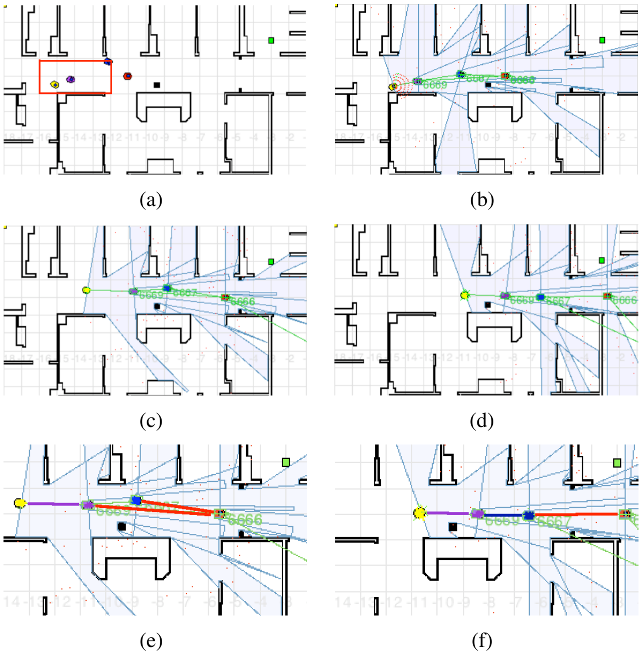
PERFORMANCE COMPARISON FOR TRACKING TASK IN SIMULATION

Env.	IQ Based Approach			Approach of [1]		
	Total No. Steps	No. Steps Visible	No. Times Lost (Steps)	Total No. Steps	No. Steps Visible	No. Times Lost (Steps)
Maze	114 ± 3.8	108 ± 4.0 (87 ± 2.1%)	1 (14 ± 2.5)	82	74 (90%)	1 (8)
City Blocks	177 ± 4.3	165 ± 5.5 (91 ± 1.5%)	1 (16 ± 2.5)	156	131 (84%)	2 (13, 12)

TABLE II

PERFORMANCE COMPARISON OF IQ BASED APPROACH AND VFH APPROACH

Initial	Information Quality (IQ) Based Approach			VFH Approach		
Configurations	Total Tracking Time	Time in Track	Track to Goal	Total Tracking Time	Time in Track	Track to Goal
Config. 1	30.1	20.2 (67%)	YES	29.7	5.3 (18%)	NO
Config. 2	30.4	19.2 (63%)	YES	26.5	9.9 (37%)	YES
Config. 3	30.0	17.9 (60%)	YES	26.6	2.4 (9%)	NO
Config. 4	26.9	13.4 (50%)	YES	18.7	4.1 (22%)	NO
Config. 5	27.5	18.8 (68%)	YES	27.2	7.1 (26%)	YES



(a) (b) (c) (d) (e) (f)

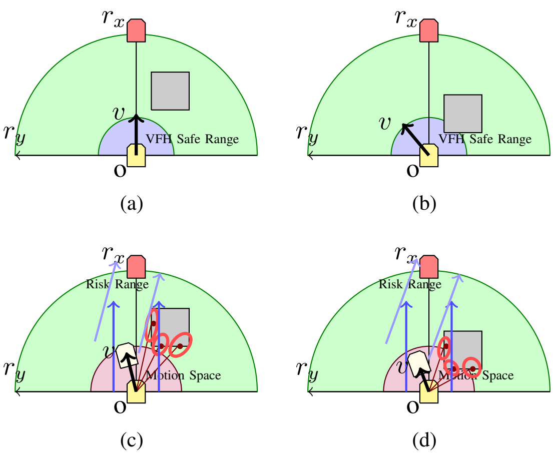
Fig. 4. (a) A typical initial configuration. The red box shows the boundaries for random configurations. All three follower robots are initially configured to follow the leader robot (red). (b) The yellow robot switches to follow the purple robot due to occlusion from the purple robot and the wall. (c) The purple robot starts switching due to occlusion from the blue robot and obstacle (black). (d) Robots follow one by one through the environment and the reconfiguration is complete. (e) Sensor constraint satisfaction paths for robot configurations in (c). (f) Sensor constraint satisfaction paths for robot configurations in (d).

leader robot moves through waypoints forming a zigzag path with obstacles in the environment. We ran the robot tracking task using both approaches in five different initial configurations and compare the results. Table II shows the statistics for the five runs[4]. While a robot using our approach successfully tracks to the goal for all five runs, a robot using the baseline approach has a success rate of only 40%. Furthermore, the average percentage of time in track for successful runs is also much higher compared to the baseline approach. Figure 5 provides a graphical explanation of how the approaches affect the behaviors of the tracking robot differently. Specifically, we explain and compare choices of motion vectors in two different scenarios for the two approaches. Clearly, when provided with exactly the same information (i.e., laser readings and marker's relative position), a robot using our approach utilizes the information more intelligently and decides the proper actions to take at each step. We believe that by reducing the computational load and incorporating motion prediction, the performance of our approach can be further improved. For the navigation task, we create two scenarios where two follower robots are to follow the leader robot. Experiments show distinct behaviors of the follower robots. Figures 6 and 7 show snapshots from the runs and compare the behaviors. The distinct behaviors show that robots using our approach can reason properly about differences in environments for sensor constraint satisfaction. Coupled with a constraint model which defines relaxation procedures appropriately, adapting robot teams to more complex environments for real applications is possible. An example application would be for robot teams to autonomously reconfigure to move through human populated environments of complex geometries with humans walking around.

V. CONCLUSIONS AND FUTURE WORK

The paper presents our approach to a general method for sensor constraint satisfaction for various applications using sensors having certain characteristics. For the sensor model, we explicitly identify two essential submodels – the sensor quality model and the sensor uncertainty model. In order to account for environmental factors, we use a sampling method for an approximate geometric representation of the environment and formalize measures of information quality.

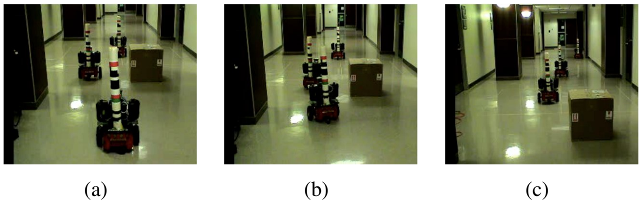
[4] Notice from the table that the running times for the baseline approach are shorter since we stopped the robots when the baseline approach failed to track and the target was unrecoverable.



(a) (b)

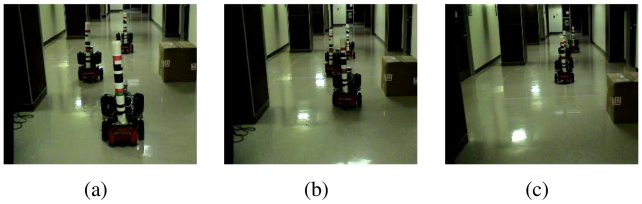
(c) (d)

Fig. 5. (a) When the obstacle is far, VFH drives tracking robot (yellow) directly to the target (red) and future tracking risk is increased. (b) When the obstacle in within the Safe Range of VFH, motion vector for avoiding obstacle dominates, potentially making target out of sight. (c) When the obstacle is far, based on environmental factors, IQ based approach chooses the motion vector resulting the maximum IQ measurement. The vector turns the tracking robot to decrease future tracking risk. (d) When the obstacle is near, IQ based approach makes the tracking robot turn more while lowering the speed to maintain certain angle error to the target. In this way, the possibility of losing the target is reduced.



(a) (b) (c)

Fig. 6. (a) An initial configuration for a navigation task where the nearer box immediately degrades the information quality for the last follower robot. (b) The last follower robot immediately switches to track the other follower robot due to the low measure of information quality for directly following the leader. (c) Robots keep in a line through the environment.



(a) (b) (c)

Fig. 7. (a) An initial configuration for a navigation task where the nearer box is moved aside so that its influence is much less. (b) The last robot keeps following the leader robot. (c) The last robot switches to track the other follower robot only when occlusion from the other follower robot reduces the information quality for following the leader.

For a specific application, we choose a motion model and sample motion space into motion vectors. The motion vector resulting in the best information quality measurement for the application is then executed. A constraint model is implemented to provide alternative solutions when certain sensor constraints are unsatisfiable. Experimental results show that our approach provides a general method for sensor constraint satisfaction by combining the use of sensor models, environment sampling, measures of information quality, a motion model with sampling, and a constraint model. As far as we know, the approach also presents the first generic formulation of robotic sensor constraints for various applications. However, limitations on the sensors and a few assumptions we make could restrict application of the approach in certain domains.

To further illustrate the generality of our approach, we plan to apply the approach to other tasks such as the box pushing task. We also plan to conduct an analysis of different sensor quality models for other interesting behaviors (e.g., dynamic robot formation). Another issue we plan to pursue is to incorporate sensor sharing and fusion for sensor constraint satisfaction, which requires an extension of our current model to enable information sharing. The approach of [12] provides inspiration for this work.

References

[1] T. Bandyopadhyay, Y. Li, M.H. Ang, and D. Hsu. A greedy strategy for tracking a locally predictable target among obstacles. In *Proc. of the IEEE Int'l. Conf. on Robotics and Automation*, pages 2342–2347, May 2006.

[2] J. Borenstein and Y. Koren. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(3):278–288, Jun 1991.

[3] F. Bourgault, A.A. Makarenko, S.B. Williams, B. Grocholsky, and H.F. Durrant-Whyte. Information based adaptive robotic exploration. In *Proc. of the IEEE/RSJ Int'l. Conf. on Intelligent Robots and System*, volume 1, pages 540–545, 2002.

[4] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

[5] B.R. Donald, J. Jennings, and D. Rus. Information invariants for distributed manipulation. In *International Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 431–459, 1997.

[6] C.H. Fua and S.S Ge. COBOS: Cooperative backoff adaptive scheme for multirobot task allocation. *IEEE Transactions on Robotics*, 21(6):1168–1178, 2005.

[7] B.P. Gerkey and M.J. Mataric. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, September 2004.

[8] A. Howard, L.E. Parker, and G. Sukhatme. Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. *International Journal of Robotics Research*, 25:431–447, 2006.

[9] N. Kalra, D. Ferguson, and A. Stentz. Hoplites: A market-based framework for planned tight coordination in multirobot teams. In *Proc. of the IEEE Int'l. Conf. on Robotics and Automation*, 2005.

[10] S.B. Marapane, M.M. Trivedi, N. Lassiter, and M.B. Holder. Motion control of cooperative robotic teams through visual observation and fuzzy logic control. In *Proc. of the IEEE Int'l. Conf. on Robotics and Automation*, volume 2, pages 1738–1743, Apr 1996.

[11] L.E. Parker, B. Kannan, Tang F., and M. Bailey. Tightly-coupled navigation assistance in heterogeneous multi-robot teams. In *Proc. of the IEEE/RSJ Int'l. Conf. on Intelligent Robots and Systems*, volume 1, pages 1016–1022, Sept.-2 Oct. 2004.

[12] L.E. Parker and F. Tang. Building multirobot coalitions through automated task solution synthesis. *Proc. of the IEEE*, 94(7):1289–1305, July 2006.

[13] C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, July and October 1948.

[14] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems*, pages 65–72, 2005.

[15] A. Stroupe and T. Balch. Value-based observation with robot teams (VBORT) using probabilistic techniques. In *Proc. of ICAR '03*, September 2003.

[16] V. Sujan and S. Dubowsky. Visually guided cooperative robot actions based on information quality. *Autonomous Robots*, 19(1):89–110, July 2005.

[17] L. Vig and J.A. Adams. Multi-robot coalition formation. *IEEE Transactions on Robotics*, 22(4):637–649, 2006.