

Multi-Robot Target Verification With Reachability Constraints

D. K. Grady*, M. Moll*, C. Hegde[†], A. C. Sankaranarayanan[†], R. G. Baraniuk[†], and L. E. Kavraki*

*Dept. of Computer Science, Rice University, Houston, TX 77005, USA. Email: {dkg1,mmoll,kavraki}@rice.edu

[†]Dept. of Elec. & Computer Engineering, Rice University, Houston, TX 77005, USA. Email: {ch3,saswin,richb}@rice.edu

Abstract — In search and rescue applications it is important that mobile ground robots can verify whether a potential target/victim is indeed a target of interest. This paper describes a novel approach to multi-robot target verification of multiple static objects. Suppose a team of multiple mobile ground robots are operating in a partially known environment with knowledge of possible target locations and obstacles. The ground robots’ goal is to (a) *collectively* classify the targets (or build models of them) by *identifying good viewpoints* to sense the targets, while (b) *coordinating their actions* to execute the mission and always be safe by avoiding obstacles and each other. As opposed to a traditional next-best-view (NBV) algorithm that generates a single good view, we characterize the informativeness of all potential views. We propose a measure for the informativeness of a view that exploits the geometric structure of the pose manifold. This information is encoded in a cost map that guides a multi-robot motion planning algorithm towards views that are both reachable and informative. Finally, we account for differential constraints in the robots’ motion that prevent unrealistic scenarios such as the robots stopping or turning instantaneously. A range of simulations indicates that our approach outperforms current approaches and demonstrates the advantages of predictive sensing and accounting for reachability constraints.

Keywords: *multirobot systems, image manifolds, motion planning*

I. INTRODUCTION

In many robotics applications, including search and rescue, mobile robots often use sophisticated algorithms to extract information from sensor measurements. To maximize the utility of available resources robots need to decide *when* and *where* sensor measurements should be taken and *how* they can navigate to positions that offer the most informative measurements.

We propose a new distributed sensing strategy that exploits the *intrinsic structure* of an ensemble of camera measurements to *predict* which future measurements may be useful (and worth navigating the robots to obtain) and which ones are redundant (and not worth obtaining); this is similar to the *next best view problem* (NBV) [1]. We propose a measure of informativeness of viewpoints based on the *predictability* of images obtained at the viewpoints given a set of known images. Our proposed solution enables the robots to navigate to viewpoints that are both *informative* and *reachable*.

Fig. 1 illustrates the importance of predictive sensing and reachability for the NBV problem. Traditionally, the cost to reach the NBV has been ignored, or has been arbitrarily set to the distance from the current position. For car-like robots

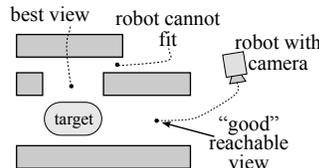


Fig. 1. Predicting informativeness of sensor measurements and reasoning about reachability are both essential for efficient target classification. While the best view of a target may be unreachable, there may exist multiple informative views that are easily reachable.

under differential constraints, this is problematic. The robots cannot simply use a reactive, control-based approach to follow a gradient based on the cost map. Furthermore, the NBV may be unreachable, or positions near it may not be informative at all (due to, e.g., occlusion). On the other hand, there may exist many reachable positions that are only marginally less informative than the best view. Since computing the next best *reachable* view is generally undecidable, we use an approximate solution and an online replanning framework to select informative views and reachable paths.

The contributions of this paper are as follows. We propose a novel formulation for multi-robot multi-target verification for car-like robots. We employ image manifolds to represent the images sensed by the robots which enables predictions of informativeness. These predictions allow us to avoid *uninformative* views, which leads to substantial savings in power usage, bandwidth, and computation. We use the concept of cost maps to encode informativeness (and lack thereof) of viewpoints. These cost maps are used by the motion planning algorithm along with constraints on reachability. By exploiting reachability robots are more likely to find short informative paths rather than potentially long paths to the “best” view point (and avoid unreachable “best” views altogether).

II. RELATED WORK

The work presented in this paper brings together ideas from diverse topics in robotics, including NBV selection, motion planning, and multi-robot coordination.

Next-best-view selection: Computing the NBV is a prominent area of research (also called “active vision”) that is primarily concerned with the online 3D reconstruction of target models [2]–[4] or entire scene environments [5], [6]. Usually, the problem setup considers one sensor, one target, and motion. In [7], a sonar-based approach that uses a cost map to encode the utility of unexplored areas is presented, but reachability is not considered. An algorithm to find points with large visibility polygons at frontiers of explored areas of unknown environments is presented in [5]. While reachability

A preliminary version of this work was presented as a poster at the IROS 2011 Workshop on Progress and Open Problems in Motion Planning.

is considered, the view point selection and path planning are treated as separate problems. A similar approach is presented in [4] for model reconstruction; there, a sampling-based planner is used to compute collision-free paths to the best viewpoints, but no differential constraints are considered. The motivation of using cost maps to characterize informativeness is similar to a Bayesian formulation [8]. However, our notion of informativeness is geometrically meaningful as we evaluate it in the context of an image manifold, while Bayesian approaches are statistically motivated and do not necessarily capture the geometric constraints underlying the problem. In [9], a statistically motivated informativeness measure is combined with geometric reasoning. By computing the image articulation manifold, we are directly geometrically motivated. Because of this, we do not estimate the correlation of images, but compute it using optical flow.

Kinodynamic motion planning: For systems with second-order dynamics, such as the car-like robots considered in this work, it is hard to find optimal solutions for motion planning problems [10], [11], but many practical approaches have been proposed. Most relax guarantees on optimality and provide weaker notions of completeness. Sampling-based algorithms [11] provide probabilistic completeness guarantees and have been successfully used for decentralized replanning for dynamic systems [12]. Such algorithms have also been combined with cost maps to find paths that minimize some cost function over paths [13].

Multi-robot coordination: There have been many approaches to let robots safely operate in the same space. They range from reactive, control-theoretic approaches to AI-based negotiation/trading-based schemes (see [14] for a detailed overview). Here, we focus on coordination with strong safety guarantees for kinodynamic robots in environments with obstacles [12]. This scheme is flexible enough to let robots coordinate on higher level tasks such as mapping unknown environments or, as in this paper, navigating to informative viewpoints in a distributed fashion. Unlike [15], we do not attempt to address the global scheduling problem, but instead let local decisions tend to drive the system to success. Although this can lead to live-lock situations, they have not been observed in simulation, noting that [12] provided results for up to 32 robots in the same relative size environments as in this paper.

III. OVERVIEW OF OUR APPROACH

The primary focus of this paper is the interaction between sensing and planning. Each robot executes a sense/plan/act loop, where it obtains a sensor measurement, plans for the next cycle, and executes the plan computed in the previous cycle. A robot may choose to *not* obtain a sensor measurement in a particular cycle if its location is predicted to be uninformative. At the heart of this sense/plan/act cycle is a geometric NBV algorithm to propose novel views to sense. In contrast to conventional NBV algorithms, we characterize the informativeness of viewpoints using a cost map to accommodate reachability constraints (candidate NBV points might be unreachable). We therefore

need to use a planning algorithm to produce dynamically feasible motions. We assume the robots have a map of the environment and are able to self-localize on this map.

Our framework has two main parts: an *offline* model building effort, where we build manifold models for each of the targets that we are interested in identifying (see Sec. IV), and an *online* real-time processing step wherein the sense/plan/act loop is executed (see Sec. V).

Offline: The goal of our offline computations is to build a framework wherein informativeness of views is characterized. As we will see later, the NBV problem can be mapped to an elegant maximization of our notion of the informativeness of viewpoints (see Sec. IV). We characterize the informativeness of a viewpoint by the size of the neighborhood on the image manifold that is predicted by the image obtained at that viewpoint.

Online: The online processing part executes a sense/plan/act loop. We focus on two processing stages: (a) an image analysis step where-in we use advances in object detection and pose estimation to locate target(s) of interest in the sensed image and suggest potential NBVs using cost maps, and (b) a path planning process that coordinates the robots to optimize some desired objective with reachability and inertial constraints. The image analysis process is non-myopic and does not simply pick the NBV according to some metric. Instead, a cost map is created to guide the planning process towards informative views. The cost map is defined by an informativeness score for all locations in the workspace and is updated at each iteration cycle by processing the aggregate set of measurements acquired by the mobile robots.

The planning process biases the growth of a search tree containing feasible paths towards areas that have a high informativeness. This accomplishes the goal of finding informative viewpoints that are also reachable. Whenever robots are within communication range, robots not only exchange any information about the targets that they have acquired so far but also communicate their plans for the next cycle. This process is explained in [12]. The plans from each neighboring robot are transformed into another cost map. The robots then use a composite cost map, formed by combining their own cost map, as well as cost maps for the neighbors' plans, to formulate a plan that balances finding an informative path with staying out of the way of neighboring robots.

IV. OFFLINE MODEL BUILDING

When the context is clear, we will interchangeably use the terms “sensor” and “camera”; “sensing” and “imaging”; and “measurements” and “images”.

Image manifolds: Suppose we consider images as elements of some high-dimensional space. Under certain conditions it has been shown [16], [17] that, if we acquire images of a moving object with K -degrees of freedom of motion, then the ensemble of N -pixel images obtained can be modeled as lying on a K -dimensional *image articulation manifold* $\mathcal{I} =$

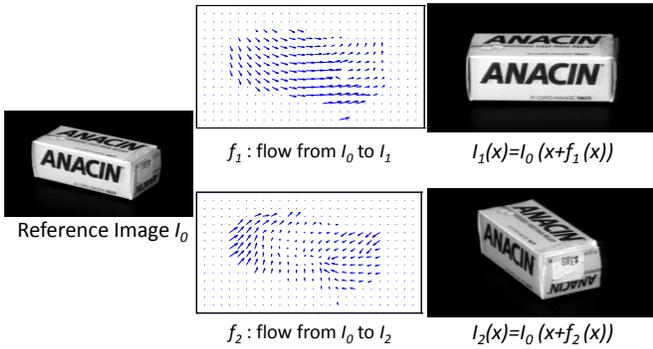


Fig. 2. Optical flow between a pair of images is defined as the pixel displacement field that maps one image onto another. Shown above is a reference image and optical flow to other images belonging to an image manifold. Note that I_0 and I_2 are flow predictable from each other. In contrast, while I_1 is flow predictable from I_0 the reverse is not true due to occlusion.

$\{I_\theta : \theta \in \Theta\}$ (IAM) embedded in \mathbb{R}^N , where Θ is the K -dimensional articulation parameter space and $I_\theta : \Theta \rightarrow \mathbb{R}^N$. Manifold models for images have been extensively studied in the context of semi-supervised classification, information fusion, and data visualization, among other applications.

Our goal is to build a notion of informativeness of images belonging to the manifolds. We achieve this by constructing operators on the manifold that link image pairs; we refer to such operators as *transport operators*.

Transport operators: A transport operator $f : \mathbb{R}^2 \mapsto \mathbb{R}^2$ is a map that enables us to “traverse” the manifold from one image I_0 to another image I_1 as follows:

$$I_1 = f(I_0) \implies I_1(\mathbf{x}) = I_0(\mathbf{x} + f(\mathbf{x})).$$

We consider the space of all such transport operators denoted as \mathcal{F} . Note that given I_0 , this space serves as a *predictive model* for images on an IAM.

We observe that the *optical flow* between a pair of images is a natural instance of transport on an IAM (see Fig. 2); consequently, optical flow can be used for *predicting* images belonging to an IAM. Optical flow computation between a pair of images is a well-studied problem in computer vision [18]–[20]. Given two images I_0 and I_1 , the optical flow between them (if it exists) is defined as the tuple $(\mathbf{v}_x, \mathbf{v}_y)$, where $\mathbf{v}_x \in \mathbb{R}^N$ and $\mathbf{v}_y \in \mathbb{R}^N$, such that

$$I(x, y) = I_0(x + \mathbf{v}_x(x, y), y + \mathbf{v}_y(x, y)). \quad (1)$$

Given two images I_0 and I_1 , we say that I_1 is *flow predictable* from I_0 if there exists an optical flow-based transport operator such that $I_1 = f \circ I_0$. Note that this notion of flow predictability is not commutative due to self-occlusions (see Fig. 2). When I_1 is flow predictable from I_0 , then there is no need to sense both images; indeed, given I_0 and the transport operator f , we can obtain I_1 using (1).

We exploit the idea of flow predictability to define a notion of *informativeness* of an image. Given an image $I \in \mathcal{I}$, we define $B(I)$ as the largest subset of the IAM that is flow predictable given I :

$$B(I) = \{I_\theta \in \mathcal{I} \mid \exists f \text{ s.t. } I_\theta = f(I)\}.$$

We call $B(I)$ as the *flow neighborhood* of I . The cardinality of the flow neighborhood, $|B(I)|$ is the measure of the informativeness in the image I . The key idea here is that images associated with higher cardinality are more desirable to be sensed as they can predict a larger neighborhood of the IAM. We exploit this in Section V to build cost maps that reflect the relative informativeness of sensing at a particular location given previously sensed data.

Model building: The initial preprocessing phase consists of building an efficient model of the IAM \mathcal{I} for each target from sufficient training data. Given training images $\{I_1, \dots, I_M\}$ and their articulation parameter values $\{\theta_1, \dots, \theta_M\}$, we first build an atlas $\{(I_k^{\text{ref}}, B(I_k^{\text{ref}})); k = 1, \dots, M^{\text{ref}}\}$ such that $\cup_k B(I_k^{\text{ref}}) \equiv \widehat{\mathcal{I}} \supset \{I_1, \dots, I_M\}$. The images I_k^{ref} are denoted as the *reference images*. The exact number of reference images is not critical but more images will produce a more accurate IAM model at a higher (offline) computational cost. The reference images should be as close to the expected articulation parameters that will be seen in the online phase, to help with image registration. The above atlas can be built using a greedy algorithm in $O(M)$ time [21]; this algorithm relies on k -means clustering extended to the IAM. In addition to the atlas, we also compute and store the optical flows from I_k^{ref} to all images in their flow neighborhood, as well as the inverse flow from the images in the flow neighborhoods to their corresponding reference images. Note that the flow maps may suffer significantly from occlusion due to obstacles in the scene. In such cases, we store binary indicator images denoting the occluded pixels as well; we term these images *occlusion maps*.

V. ONLINE PROCESSING

The online processing framework consists of a sense/analyze/act loop. As mentioned earlier in Section III, there are two distinct processing stages: an image analysis step and a motion planning step. The main elements of these two stages are discussed next.

Preprocessing: A series of image analysis algorithms need to be performed to identify the location and the pose of the target(s) from the sensed imagery; we assume that state-of-the-art computer vision techniques for object detection provide us a bounding box around the target. Typically, such algorithms use a bag-of-features model [22] to describe target(s) in terms of their SIFT (or any other) features [23]. The end goal of these algorithms is to identify the location of the target and potentially its pose. For our applications, once a bounding box is obtained around a target, we use our precomputed manifold models to predict the NBV as follows.

Model validation: Given a set of processed images $\{I^s\}$ (images of the target separated from the background) and the precomputed models for the IAM \mathcal{I} , the next phase is to predict the NBV to guide the navigation. We achieve this using the following procedure: given a sequence of sensed images $\{I_k^s\}$ including the newly sensed images, we compute $\cup_k B(I_k^s)$, which represents the portion of the IAM that is

flow-predictable from the sensed images. We observe that for $k = 1, \dots, M^{\text{ref}}$: (1) If I_k^{ref} is flow predictable from I_k^s for some k , then $B(I_k^{\text{ref}}) \subset B(I^s)$; (2) If I_k^s is flow predictable from I_k^{ref} , then we compute the optical flow and corresponding occlusion map from I^s to I_k^{ref} . The set $B(I^s)$ includes all images in $B(I_k^{\text{ref}})$ whose occlusion map with respect to I_k^{ref} is a superset of the occlusion map of $B(I^s)$; and (3) If $\bigcup_k B(I_k^s) - \widehat{\mathcal{I}} = \emptyset$, then we have charted the IAM using the sensed images and declare the target identity as validated.

In our implementation, these steps are done in a decentralized manner with each robot processing only the images that it has acquired. Each robot builds the flow neighborhoods $B(I^s)$ over its own images. Computing the union $\bigcup_k B(I_k^s)$ is done easily by local communication of the set of image articulation parameters that each robot has captured; this avoids the need to exchange the images between the robots.

Online cost map computation: When $B_{\text{diff}} = \bigcup_k B(I_k^s) - \widehat{\mathcal{I}} \neq \emptyset$, then there are neighborhoods in the IAM that we need to chart and corresponding neighborhoods in the workspace that should be explored. To facilitate this, we first define the vector $\mathbf{s} \in \mathbb{R}^M$ such that $\mathbf{s}(k) = 1$ if $I_k \in B_{\text{diff}}$ and $\mathbf{s}(k) = 0$ otherwise, where $\mathbf{s}(k)$ is the k -th component of \mathbf{s} . Then, we define a real-valued function $C(x, y)$ on the *workspace*:

$$C(x, y) = \sum_{k=1}^M \mathbf{s}(k) c(|B(I_k) - B_{\text{diff}}|) \text{Ind}((x, y) \in W(\theta_k)), \quad (2)$$

where $\text{Ind}(\cdot)$ is the indicator function and $W(\theta_k)$ is the set of physical (workspace) coordinates that correspond to the articulation parameter θ_k . The quantity $c(|B(I_k) - B_{\text{diff}}|)$ is a payoff term associated with acquiring the image I_k that depends on the cardinality of the additional portion of the IAM \mathcal{I} that we are able to chart by sensing the image I_k . In this paper, we use $c(k) = \frac{k}{M}$, where M is the number of images in our training ensemble. We have freedom in our choice of $c(\cdot)$ and any monotonic increasing function can be used.

The above formulation is invariant to the order in which the images are acquired. Indeed, it does not matter if the images were captured by more than one camera. Thus, our approach can be used in the mobile *multi-camera* setup. Further, the framework can be extended to the case of *multiple targets*. In this setting, the IAM dimension increases linearly with the number of targets, but all other aspects remain the same.

Online implicit coordination through cost maps: The cost map derived from the measurements is used to guide the navigation of multiple robots. Each robot executes a replanning loop, in which each robot takes a sensor measurement iff the cost map indicates it is informative. Since the focus of this paper is on the interaction between sensing and planning, and not on communication, we will assume that updates to the cost map can be propagated relatively quickly through the robots' communication network. The robots thus have a consistent view of the world. (Communication delays, dynamic network topologies, transmission errors, and uncertainties will be addressed in future work.)

First, each robot analyzes its immediate vicinity to determine

if targets can be seen or if it still needs to navigate to such an area. It does so by computing the percentage of grid cells in the cost map in small neighborhood around the current position that contain informative viewpoints (i.e., $B(I^s)$ is above some threshold value). If this percentage is above a certain prescribed value p (e.g., $p = 20\%$ in our experiments), then the robot will use the new cost map as-is. On the other hand, if this threshold is not met, then the robot considers itself as far away from useful views and switches to using a slightly different cost map. In this case, the robot identifies informative areas in the sensing cost map and uses a brushfire algorithm to compute the workspace distance to these areas. This brushfire-based cost map is then used to guide the robot closer to informative areas. Finally, when a robot receives paths from other, neighboring robots, it will increase the costs around those paths since there is no need for two robots to sense the same areas. The computation of the effective cost map from the sensing cost map is extremely simple and can be performed just prior to motion planning in each replanning cycle. The end result is that each robot finds paths that balance the informativeness of a viewpoint with the cost to reach it. Simultaneously, the cost map assists with the coordination among the robots so that they do not compete to get to the same viewpoint (since the cost to cross paths from nearby robots causes a robot to navigate to other viewpoints). Note also that there is no mapping from robots to targets. This is a huge advantage when multiple targets can be seen from a single viewpoint, as we will demonstrate in Section VI.

Multi-robot motion planning: Due to the complex dynamics of the robots, they cannot simply follow a potential based on the cost map but rather need to plan to obtain controls that drive the robots through low-cost cells in the cost map. Each ground robot has second-order, car-like dynamics:

$$\begin{aligned} \dot{x} &= v \cos \phi \cos \theta, & \dot{y} &= v \cos \phi \sin \theta, \\ \dot{\theta} &= v \sin \phi, & \dot{v} &= u_0, & \dot{\phi} &= u_1, \end{aligned}$$

where (x, y, θ) is the pose of a robot, v is its velocity, ϕ is its steering angle, and the control vector $u = (u_0, u_1)$ controls the acceleration and turning rate. The velocities and controls are bounded, so the robots cannot stop or change direction instantaneously. This makes navigation in environments with obstacles very challenging; the robots cannot rely on purely reactive behaviors and need to plan ahead.

Each robot generates a series of candidate paths using a sampling-based planner [11]. Such planners have been demonstrated to be effective in finding valid paths for constrained, dynamic systems [24] and can be guided in their search for good paths by cost maps. The robots communicate with their neighbors not only to exchange sensor updates, but also to exchange proposed plans. We use a motion coordination framework for car-like robots with the second-order dynamics described above that enables them to safely operate in the same space [12]. This framework has many highly desirable properties. A key guarantee is that no robot will collide with any other robot or any static obstacle in the environment. This guarantee holds over an infinite time horizon. During a planning

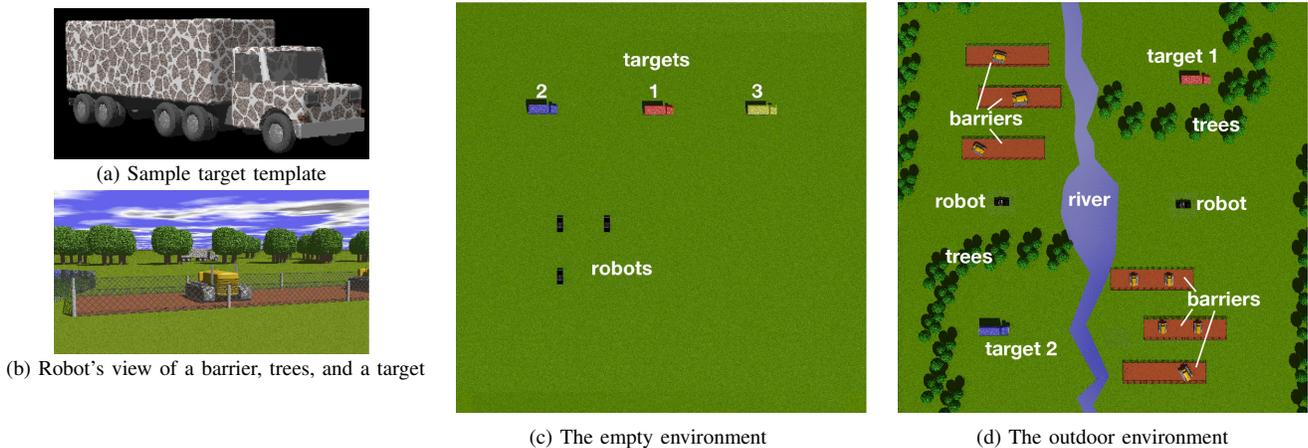


Fig. 3. Simulation setup. (a) Example image from the target (model) manifold. (b) Example image captured by a robot. (c) Top view of the empty scene. (d) Top view of the outdoor scene. The different colors for the targets denote that the targets are distinct; they do not need to be the same truck.

cycle, a robot computes one or more valid plans and sends the most informative path to its neighbors, who check whether this path is compatible with their own paths. If the path is compatible with all the neighbors' paths, the robot will execute it. However, if the robot could not find *any* valid plan, then it will fall back on a contingency maneuver (such as braking).

VI. SIMULATIONS

We have constructed a series of environments of increasing difficulty to test our approach. They are shown in Fig. 3. In the empty environment (Fig. 3(c)), we used 3 robots to verify k distinct targets ($k = 1, 2, 3$). In the more realistic environments (Fig. 3(d)), there are two targets and two robots. In the simplest version of this environment, there is only a river separating the left half of the scene from the right half. The river is not an occluding obstacle, but only a navigational obstacle. An additional level of complexity is introduced by adding trees around the targets, so that the robots cannot get close to the targets, but still have (partial) views of the target. In this scenario (“+trees” below) the robot cannot individually sense the targets completely, but together they can. In the most complex version (referred to as “+barriers” below), we have added some more navigational obstacles in the form of a few fenced-in barriers that introduce additional occlusion. Fig. 4 shows the cost map $C(x, y)$ (see (2)) for the “+trees” scenario after the two robots have taken a few measurements.

Simplifications: A ground-based mobile camera is parameterized by 3 degrees of freedom (its 2D location (x, y) and its 1D orientation ϕ) and thus characterized by a 3D articulation space Θ . However, we assume that the target can be centered in the image whenever it is within the field of view, thereby reducing the articulation space by one dimension. Second, if we ignore perspective effects, then translation of the camera towards the target results in a simple scaling of the target. By scaling images to a preset canonical size we reduce the problem by an additional dimension. Thus, we have reduced a 3D articulation space to a simple 1D articulation space, parameterized by a circle around the target $\Theta = \mathbb{S}^1$. Given $\phi \in \Theta$, the image I_ϕ

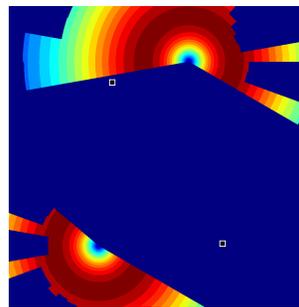


Fig. 4. Sensor cost map for the “+trees” scenario in figure 3(d). The robots' current positions are indicated by black squares. Blue corresponds to uninformative, while red corresponds to highly informative. Note that areas from which views are explainable by previous measurements as well as areas from which the targets are occluded by trees are both marked as uninformative.

corresponds to a camera located at $(r \cos \phi, r \sin \phi)$ viewing the target. We leverage this simplification in the computation of the cost map (2) of each robot at every time step as follows: we can redefine $\text{Ind}((x, y) \in W(\phi_0)) = 1$ provided $\text{atan} \frac{y-y_t}{x-x_t} = \phi_0$ and zero otherwise, where (x_t, y_t) is the location of the target. In addition to this, we add a penalty term to account for loss in resolution due to scaling of the form $e^{-\kappa((x-x_t)^2+(y-y_t)^2)}$.

Comparison with other approaches: We analyze the performance of our algorithm using two metrics: (a) time necessary to verify all targets, and (b) number of sensing measurements taken. We use the following stopping criterion: when 75% of the IAM is charted, we declare success in target verification. Our approach is compared to two simpler approaches. In the “naïve” approach the robots simply need to get a view of the target from every angle. In other words, the robots do not compute $B(I^s)$ to predict images from nearby viewpoints. In the other approach (called NBV below) the robots are repeatedly trying to navigate to the k next best views. For the NBV approach we compute the k next best views from the sensing cost map and let those be targets for k robots (without assigning targets to robots). Here, the effective cost map is defined by the sum of Euclidean distances to the k views. In all cases the planning algorithm that is guided by the cost map remains the same. The comparison with the naïve approach shows the advantage of predictive sensing, while comparing with the NBV approach will highlight the importance of exploiting reachability.

Fig. 5 summarizes the results from all simulations. In the

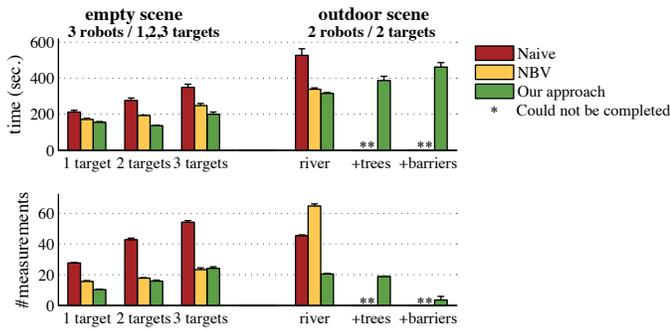


Fig. 5. **Simulation results.** (top) The mean time to verify the identity of all targets using the three different approaches for the two different scenes. For each scene there are three variations of increasing complexity, as indicated along the X-axis. For each variant we computed the mean over 10 runs. The error bar corresponds to one standard error of the mean. (bottom) A similar plot for the mean total number of measurements taken.

outdoor scene, the naïve and the NBV algorithms fail to complete the validation tasks; in these cases occlusion and navigational obstacles causes the robots to get stuck trying to get to unreachable positions. As one would expect, the NBV approach is competitive in terms of completion time with our approach in the empty scene, but it cannot handle more complex scenes. From the measurements plot in Fig. 5 we see that our approach requires fewer measurements in almost all cases. Also, the number of measurements decreases with the complexity of the outdoor scene variations. This is due to occlusion: there are simply fewer unobstructed informative views available along a path. In contrast, our approach is still able to find all the necessary viewpoints to verify the targets.

VII. CONCLUSIONS

We have presented a novel approach to multi-robot target verification of multiple static targets. This problem was cast as a distributed next-best-view problem with differential motion constraints on the robots. By reasoning about the image manifold we can predict which measurements are expected to be useful. Through simulations we demonstrated substantial savings in both the time required and the number of measurements needed when compared with alternative approaches. Through the use of cost maps the approach achieves both tight integration between sensing and planning as well as implicit coordination among the robots. The coordination is used to avoid collisions and to distribute the robots over multiple areas of interest. In future work we plan to extend the current work to experimentation on hardware platforms, opportunistic model building, pursuit-evasion games, and stealthy sensing.

ACKNOWLEDGEMENTS

This work was supported by grants NSF CCF-0431150, CCF-0926127, CCF-1117939, 0713623, 0920721, 1018798; DARPA N66001-11-C-4092 and N66001-11-1-4090; ONR N00014-10-1-0989, N00014-11-1-0714, and N00014-12-10579; AFOSR FA9550-09-1-0432; ARO MURI W911NF-07-1-0185 and W911NF-09-1-0383; and the Texas Instruments Leadership University Program. Additional support has been provided by an NSF Graduate Fellowship for DKG.

REFERENCES

- [1] J. E. Banta, L. R. Wong, C. Dumont, and M. A. Abidi, "A next-best-view system for autonomous 3-D object reconstruction," *IEEE Trans. Systems, Man and Cybernetics, Part A*, vol. 30, no. 5, pp. 589–598, 2000.
- [2] R. Pito, "A solution to the next best view problem for automated surface acquisition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 10, pp. 1016–1030, 1999.
- [3] E. Dunn, J. van den Berg, and J.-M. Frahm, "Developing visual sensing strategies through next best view planning," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Oct. 2009, pp. 4001–4008.
- [4] L. Torabi and K. Gupta, "Integrated view and path planning for an autonomous six-dof eye-in-hand object modeling system," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Oct. 2010, pp. 4516–4521.
- [5] H. H. González-Baños and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *Intl. J. of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.
- [6] J. Faigl, M. Kulich, and L. Přeučil, "A sensor placement algorithm for a mobile robot inspection planning," *Journal of Intelligent & Robotic Systems*, pp. 1–25, 2010.
- [7] R. Grabowski, P. Khosla, and H. Choset, "Autonomous exploration via regions of interest," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Oct. 2003.
- [8] C. Laporte and T. Arbel, "Efficient discriminant viewpoint selection for active bayesian recognition," *International Journal of Computer Vision*, vol. 68, no. 3, pp. 267–287, 2006.
- [9] J. Velez, G. Hemann, A. Huang, I. Posner, and N. Roy, "Planning to perceive: Exploiting mobility for robust object detection," in *Proc. ICAPS*, 2011.
- [10] J. Canny, *The Complexity of Robot Motion Planning*. Cambridge, MA: MIT Press, 1987.
- [11] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.
- [12] K. E. Bekris, D. K. Grady, M. Moll, and L. E. Kavraki, "Safe distributed motion coordination for second-order systems with different planning cycles," *Intl. J. of Robotics Research*, vol. 31, no. 2, pp. 129–149, Feb. 2012.
- [13] L. Jaillet, J. Cortés, and T. Siméon, "Sampling-based path planning on configuration-space costmaps," *IEEE Trans. on Robotics*, vol. 26, no. 4, pp. 635–646, Aug. 2010.
- [14] L. E. Parker, "Path planning and motion coordination in multiple mobile robot teams," in *Encyclopedia of Complexity and System Science*, R. Meyers, Ed. Springer Verlag, 2009.
- [15] L. Matignon, L. Jeanpierre, and A. Mouaddib, "Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [16] H. Murase and S. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *Intl. J. of Computer Vision*, vol. 14, no. 1, 1995.
- [17] D. Donoho and C. Grimes, "Image manifolds which are isometric to Euclidean space," *J. Math. Imaging and Computer Vision*, vol. 23, no. 1, July 2005.
- [18] B. Horn and B. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
- [19] C. Liu, "Beyond pixels: exploring new representations and applications for motion analysis," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.
- [20] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Trans. PAMI*, 2010.
- [21] A. C. Sankaranarayanan, C. Hegde, S. Nagaraj, and R. G. Baraniuk, "Go with the flow: Optical flow-based transport operators for image manifolds," in *Allerton Conference on Communication, Control and Computing*, Allerton, IL/USA, September 2011.
- [22] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. Ieee, 2006, pp. 2169–2178.
- [23] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [24] K. E. Bekris, K. I. Tsianos, and L. E. Kavraki, "Safe and distributed kinodynamic replanning for vehicular networks," *ACM/Springer Mobile Networks and Applications*, vol. 14, no. 3, pp. 292–308, 2009.