

# Overcoming Unknown Occlusions in Eye-in-Hand Visual Search

Sina Radmard, David Meger, Elizabeth A. Croft, and James J. Little

**Abstract**— We propose a method for handling persistent visual occlusions that disrupt visual tracking for eye-in-hand systems. Our approach allows a robot to “look behind” an occluder and re-acquire its target. To allow efficient planning, we avoid exhaustive mapping of the 3D occluder into configuration space, and instead use informed samples to strike a balance between target search and information gain. A particle filter continuously estimates the target location when it is not visible. Meanwhile, we build a simple but effective map of the occluder’s extents to compute potential occlusion-clearing motions using very few calls to efficient approximations of inverse kinematics. Our mixed-initiative cost function balances the goal of directly locating the target with the goal of gaining information through mapping the occluder. Monte-Carlo optimization with efficient data-driven proposals allows us to approximate one-step solutions efficiently.

Experimental evaluation performed on a realistic simulator shows that our method can quickly obtain clear views of the target, even when occlusions are persistent and significant camera motion is required.

## I. INTRODUCTION

Robots equipped with camera sensors are increasingly common, as vision is an attractive sensor for a variety of unstructured and human aware tasks. Visual tracking, vision guided robot control (i.e., visual servoing) and visual Simultaneous Localization and Mapping (vSLAM) are just part of the diversity of robot vision methods [1, 2]. Furthermore, visual data is acquired through a wide range and combination of sensor systems, including stationary camera(s) observing the scene, mobile-robot mounted pan-tilt cameras, and eye-in-hand cameras mounted on manipulators. With the growing cadre of vision systems, there is a commensurate demand for robust robotic visual guidance in less structured environments.

We consider the problem of visual target

tracking for single-camera eye-in-hand systems. Applications include assembly line part picking, manipulation, inspection, and surveillance. Our specific focus is on recovery of visual tracking after the *loss of the target* due to an occlusion, for example, when a human co-worker intrudes between the target and the camera during an inspection or grasping process.

Previous researchers focused primarily on maintaining the target within the camera field of view (FOV) [3, 4]. Strategies to increase the camera’s tracking region were proposed to overcome tracking failures arising from boundaries due to robot joint limits or kinematic singularities [5].

Occlusions are another important source of tracking failure. Here, we classify occlusions into two types. The first involves a moving target or moving occluder where visibility is temporarily lost, but the target quickly reappears within the FOV without repositioning the camera, namely a *Temporary Occlusion*, as addressed by Tsai et al. [6]. Herein, we consider a second type, *Persistent Occlusions*, requiring search for a manipulator configuration to restore visibility of the lost target by actively “looking around” the occluder so that tracking can resume. This new configuration must avoid singularities and joint limits while ensuring the target is inside the camera FOV.

For our method to restore the view of the target, it must first obtain an understanding of the occluder’s extents. This involves: i) constructing a map that splits the world into occlusion regions, free-space, and unobserved areas, ii) continuously updating this map with new sensory data, and iii) permitting queries about “how much new information will be gained about the occlusion if the robot is moved to position  $q$ ”. Once a partial or complete map is obtained, it is possible to reason about whether looking around the known occluder boundaries may restore the view of the target.

We answer information-based planning queries

Manuscript received September 17, 2012. Financial support for this research was provided by the Natural Science and Engineering Research Council of Canada and the Canada Foundation for Innovation.

S. Radmard and E. A. Croft are in the Collaborative Advanced Robotics and Intelligent Systems Laboratory, Mechanical Engineering Dept., University of British Columbia, V6T1Z4 Canada (e-mails: sradmard@interchange.ubc.ca, ecroft@mech.ubc.ca).

D. Meger and J. J. Little are in the Laboratory for Computational Intelligence, Dept. of Computer Science, University of British Columbia, V6T1Z4 Canada (e-mails: dpmege@cs.ubc.ca, little@cs.ubc.ca).

using the concept of entropy maps, motivated by Zhang et al. [7]. Therein, a visual search strategy for eye-in-hand manipulator planning explicitly avoids uninteresting regions of the world. The authors utilized a novel extremum seeking control (ESC) over entropy maps, constructed from images, in real time, without *a priori* information. The same concept enables our motion planner to locate views that are likely to yield additional information about the occluder’s extents.

We note that visibility reasoning, even around well-determined occluder extents, is not trivial. The manipulator operates in configuration space (C-space), and the occluder exists as a set of 3D boundaries in the robot’s workspace. On one hand, workspace planning is hampered by non-linear mappings of joint limits and singularities. On the other hand, densely mapping the entire 3D boundary into C-space with inverse kinematics (IK) is both computationally expensive and problematic for high DOF robots that can have multiple (or, if redundant, infinite) IK solutions.

Our approach involves C-space planning with infrequent mapping of 3D quantities into C-space. We are motivated by work on “Next Best View” planning [8, 9] which performs C-space planning during 3D modeling of object shapes. However, their approach of transferring the entire 3D workspace information into C-space is expensive. Herein, we show that our task-based visibility query can be answered efficiently with approximate sampling-based optimization.

The overview of our lost-target recovery algorithm (LTRA) is as follows: A target tracking module estimates the occluded target’s position based on the latest observations and a simple motion model. Simultaneously, a probabilistic occlusion map is constructed. Based on this continuously-evolving map, a planning module uses light-weight approximate IK routines only at a small number of critical points defined around known occluder boundaries and the frontiers of unknown space. These critical points provide informative seeds for our C-space sampling, which increases the efficiency both of constructing the occluder map and also reacquisition of the target.

The contributions of our algorithm are to

identify feasible visibility restoring configurations (a 3D property) without expensive C-space mapping and with minimal calls to IK. Our entropy-guided mapping term ensures that the occluder is also mapped with few sense/plan/act iterations. Our algorithm does not require *a priori* knowledge of the occluder size or shape, and the target motion can be arbitrary.

## II. PROBLEM FORMULATION AND PRELIMINARIES

This section formalizes the inputs, goals and assumptions of our LTRA. We assume the target has been previously tracked, and thus an estimate for its motion is available. Our algorithm initiates when the target is no longer visible in the FOV due to occlusion, thus, the tracking system fails. We accept that all boundaries of the occluder may not fall within the FOV, but expect that the sensor can detect at least one occluder edge as it intrudes. This generalization distinguishes our prior work [10], where all relevant occlusion boundaries were assumed visible at initialization.

Our planner must locate configurations resulting in restored target visibility. In principle, a planner with a multiple-step planning horizon could be desirable, but we consider 1-step planning in this work to maintain *real-time* operation. After planning, the robot moves to the specified configuration, more data is collected and the planner repeats if the target remains occluded.

We denote by  $X$  a point in Cartesian space,  $\mathbf{q}$  a point in C-space (representing a robot configuration),  $diag(\mathbf{Y})$  a diagonal matrix formed by a vector  $\mathbf{Y}$ , and  $\mathbf{e}_Y$  the unit vector along any vector  $\mathbf{Y}$ . The subscripts  $t$  and  $t+1$  represent the current and the next time step respectively.

## III. METHODOLOGY

We formulate the task of finding next-best-views for target recovery as optimization of a mixed-initiative cost function defined in C-space (i.e. as a function of the vector of joint angles,  $\mathbf{q}$ ):

$$\mathbf{q}_{next} = \operatorname{argmin}_{\mathbf{q}} [\Gamma_{\text{Pan}}(\mathbf{q}) + \Gamma_{\text{Tilt}}(\mathbf{q}) + \Gamma_{\text{Dis.}}(\mathbf{q}) + \Gamma_{\text{Map.}}(\mathbf{q}) + \Gamma_{\text{Vis.}}(\mathbf{q})]. \quad (1)$$

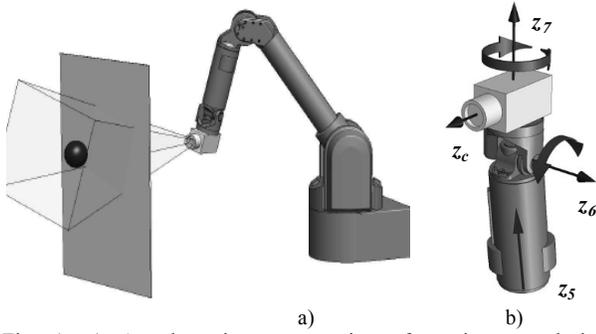


Fig. 1. a) A schematic representation of persistent occlusion. b) Mounted camera configuration on the last link. Figures generated using Barrett Technology's WAM CAD model.

The cost function terms are the pan-configuration, tilt-configuration, distance, mapping, and visibility, respectively. In the following sections we describe each term, followed by our sampling-based method for minimizing (1).

#### A. Configuration Terms: $\Gamma_{Pan}(\mathbf{q}) + \Gamma_{Tilt}(\mathbf{q})$

In [10], capitalizing on the typical arm plus spherical wrist manipulator configuration, we used a decoupled search, with sensor placement (using the arm) in the visible region followed by a pan-tilt search (with the wrist) to reacquire the target. To facilitate the pan-tilt search, the sensor placement strategy maximizes the searchable region by ensuring that the pan and tilt joints are centered within their range of motion when the camera is pointing towards the estimated target location,  $\hat{\mathbf{X}}_{t+1}^T$ . This estimated location is provided by a particle filter based algorithm [11]. For the example camera-mount configuration (Fig. 1) on a  $n$ -DOF robot, the optimal placement is such that the distal joint  $Z$ -axis ( $\mathbf{Z}_n$ ) is perpendicular to the line connecting the camera center,  $\mathbf{X}_t^c$ , to  $\hat{\mathbf{X}}_{t+1}^T$ . This intuition is captured with two cost function terms. The first is related to the pan axis:

$$\Gamma_{Pan}(\mathbf{q}) = \left| \mathbf{Z}_n(\mathbf{q}) \cdot \mathbf{e}_{(\hat{\mathbf{x}}_{t+1}^T - \mathbf{x}_t^c(\mathbf{q}))} \right| \quad (2)$$

The second ensures that the tilt joint,  $q_{n-1}$ , is centered by aligning the  $q_{n-2}$  and  $q_n$   $Z$ -axes. This is achieved by minimizing the cross product norm:

$$\Gamma_{Tilt}(\mathbf{q}) = \|\mathbf{Z}_{n-2}(\mathbf{q}) \times \mathbf{Z}_n(\mathbf{q})\|. \quad (3)$$

One can extract the joint axes of rotation ( $Z$ -axes) in (2) and (3) from the Jacobian matrix at  $\mathbf{q}$ .

By centering joint  $q_{n-1}$ , joint  $q_{n-2}$  can redundantly compensate for  $q_n$  motion so that  $q_n$  (pan joint) can also be centered easily. Each of  $\Gamma_{Pan}$  and  $\Gamma_{Tilt}$  take values in the range [0 1], with smaller values indicating more favorable configurations. We enforce the same [0 1] range on each component of our mixed-initiative planning function, so that scaling and weighting of terms are intuitive.

#### B. Distance Term: $\Gamma_{Dis}(\mathbf{q})$

As gross motion has time and safety costs, paths for reacquiring the target should minimize manipulator travel distance. Here, we consider the Euclidean distance in  $C$ -space from the current configuration,  $\mathbf{q}_t$ , to a possible next step,  $\mathbf{q}_{t+1}$  and minimize:

$$\Gamma_{Dis}(\mathbf{q}) = \sqrt{\frac{\sum_{i=1}^n \left( \frac{q_{i,t+1} - q_{i,t}}{q_{imax} - q_{imin}} \right)^2}{n}}, \quad i = 1, \dots, n, \quad (4)$$

where  $q_{imin}$  and  $q_{imax}$  are the joint limits of the  $i^{\text{th}}$  joint and  $n$  is the number of joints. The Euclidean distance is normalized to range over [0 1].

#### C. Mapping Term: $\Gamma_{Map}(\mathbf{q})$

Our occlusion mapping process includes: i) the probabilistic model used to represent the system's partial knowledge of the occluder extents from a sequence of images; and ii) the information-theoretic measure used to capture representation completeness and fidelity and to compare the value of making each new observation. Here, we assume the occluder is a convex hull of known and potential edges. Our map construction process is illustrated in Fig. 2.

In the first camera image where the target is occluded, only a sub-set of the edges of the occluder are detected. We label these as *known edges* (e.g., edge 1 in Fig. 2). The boundaries of the image limit our spatial perception. We name the limits of known occlusion regions caused by image extents as *potential edges* (e.g., edges 5, 6, and 7 in Fig. 2). Until mapping is complete there are *unknown edges* (e.g., edge 4 in Fig. 2). Our system assumes such edges exist because the occluder is a connected, bounded object, but has no knowledge of their location until they are directly observed in the sensor.

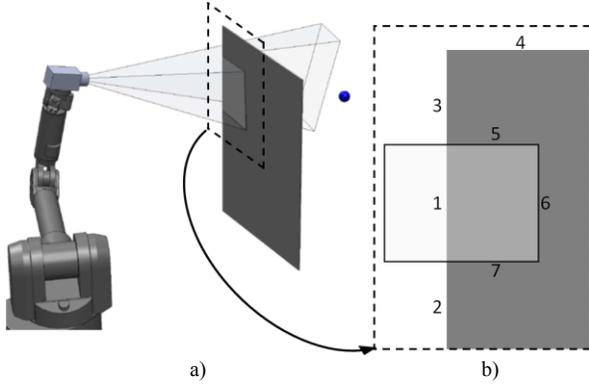


Fig. 2. a) Detecting only a single edge of the occluder. b) A view of the occlusion plane with the projected camera FOV, defining different types of occlusion edges. 1-3 represent known edges (due to convex hull assumption), 4 represents an unknown edge and 5-7 represent potential edges.

Our probabilistic mapping model is formed from the edge-set by a flood-fill operation. Regions surrounded by known and potential edges are occlusion with certainty. The occlusion/visibility of regions beyond potential edges is considered uniformly uncertain, as our only *a priori* knowledge of the occluder shape is that it is bounded and connected. Finally, areas outside known edges are considered visible with certainty.

Our mapping algorithm incrementally constructs the edge map with each new image observation in order to resolve potential and unknown edges into known edges. Each camera motion that observes a previously unseen portion of the occluder will lead to some additional map information, but we encourage actions that lead to as much new knowledge as possible. This is captured with expected Information Gain,  $IG(\mathbf{q})$ , as:

$$IG(\mathbf{q}) = H_{p_t}(m) - E(H_{p_{t+1}}(m|\mathbf{q})), \quad (5)$$

where  $H_{p_t}(m)$  refers to the entropy of the occupancy probability,  $p_t$ , at time step  $t$ , of a map,  $m$ , defined as:

$$H_{p_t}(m) = -\sum_j p_t(m_j) \log(p_t(m_j)). \quad (6)$$

Here we use an occupancy grid map where  $m_j$  denotes the grid cell with index  $j$ . Hence, the information gain is defined as the measure of all the unexplored grid cells set. This is computed by

a ray casting procedure similar to [12].

Finally we compute the maximum possible Information Gain by resolving all of the remaining edges to a known state within a single step, and normalizing each computed  $IG(\mathbf{q})$  by this value,  $IG_t$ , to obtain  $\Gamma_{\text{Map}}(\mathbf{q})$  in the range [0 1]:

$$\Gamma_{\text{Map}}(\mathbf{q}) = IG(\mathbf{q})/IG_t. \quad (7)$$

#### D. Visibility Term: $\Gamma_{\text{Vis}}(\mathbf{q})$

The final term in our cost function measures whether the target will be visible at the proposed location (*i.e.*, the line from the camera to the target does not pass through the occluder), the overall goal of our system. The visible region,  $R_{\text{vis}}$ , is computed using shadow planes. The *shadow plane* (visibility plane) separates the environment around the target into two half-spaces of visible and shadow region constructed by occluder edges. Fig. 3 illustrates the geometry used to construct shadow planes from the target's estimated position,  $\hat{\mathbf{X}}_T$ , and an occluding edge. Each edge of the occluder defines a shadow plane.

C-space planning points,  $\mathbf{q}$ , are mapped via Forward Kinematics (FK) to find the camera Cartesian locations. These locations are checked against  $R_{\text{vis}}$  to determine whether or not  $\mathbf{q}$  leads to target visibility. The cost-function term is derived from this computation as follows:

$$\Gamma_{\text{Vis}}(\mathbf{q}) = \begin{cases} 0.01 & \text{if } \mathbf{X}_c(\mathbf{q}) \in R_{\text{vis}} \\ 1 & \text{if } \mathbf{X}_c(\mathbf{q}) \notin R_{\text{vis}} \end{cases}. \quad (8)$$

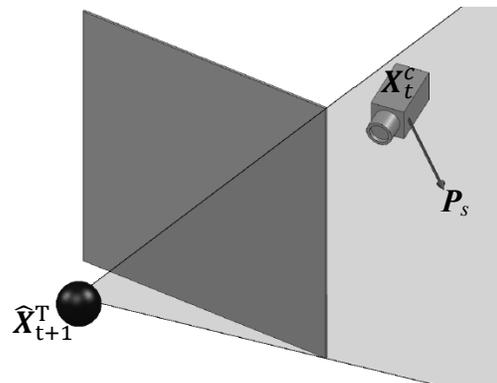


Fig. 3. A schematic representation of a shadow plane and closest point.

### E. Monte-Carlo Optimization

The previous sections have described each term in mixed-initiative cost function (1). The goal of our LTRA planner is to determine the minimum cost arm configuration at each step. Since all of the terms in (1) are non-linear - the mapping and visibility terms are discontinuous - we have chosen a gradient free minimization method based on data-driven importance sampling.

To avoid joint limits and singularities, we draw samples from the robot's C-space. To avoid exceeding joint limits, we introduce a penalty function,  $W_q$ , on C-space samples,  $\mathbf{q}$  :

$$W_q = \begin{bmatrix} w_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_n \end{bmatrix}_q, \quad (9)$$

$$[w_i]_q = \left( 1 - e^{-k \frac{(q_i - q_{imin})(q_{imax} - q_i)}{(q_{imax} - q_{imin})^2}} \right),$$

$$i = 1, \dots, n.$$

where  $k$  is a user defined parameter used to control the acceptable safe distance from the limits, and is set to 0.2 in our experiments.

Uniform sampling of a manipulator's  $n$ -DOF solution space would be prohibitively expensive. Hence, we focus our samples to promising C-space regions using a data-driven proposal distribution. This proposal distribution uses information about the current manipulator position and map representation to sample two heuristically promising types of C-space regions: those that accomplish *zooming back* and those that allow *looking around* the occluder. The overall proposal is formed from these two sub-components.

The *Type I* proposal component captures *zooming back* to gather more information by broadening the camera's FOV. We construct a multivariate Gaussian distribution,  $\pi_I$ , in C-space with mean at  $\mathbf{q}_t$ . We consider a covariance,  $\Sigma$ , that directs the distribution towards a location farther from the target than the current camera, while it orients the camera towards the predicted target location. The covariance is formed from the pseudo inverse of the Jacobian matrix  $J_t^\dagger$ .

$$\pi_I = \mathcal{N}(\mathbf{q} | \mathbf{q}_t, \Sigma), \quad (10)$$

$$\Sigma = W_{q_t} \text{diag}(\mathbf{e}_{\delta q_t}), \quad \delta \mathbf{q}_t = J_t^\dagger (\mathbf{X}_t^c - \widehat{\mathbf{X}}_{t+1}^T).$$

The *Type II* proposal component guides the robot towards *looking around* the occluder to find the target. We construct one Gaussian for each shadow plane,  $l$  ( $l=1, \dots, n_p$ ). The mean of each distribution in C-space,  $\mathbf{q}_{P_{s,l}}$ , is derived using IK at the closest point  $P_{s,l}$  to the current camera location that falls on the shadow plane, as shown in Fig. 3. The *Type II* proposal component is:

$$\pi_{II} = \sum_{l=1}^{n_p} \mathcal{N}(\mathbf{q} | \mathbf{q}_{P_{s,l}}, \Sigma_l), \quad (11)$$

$$\Sigma_l = W_{q_{P_{s,l}}} \text{diag}(\mathbf{e}_{(q_{P_{s,l}} - q_t)}).$$

IK is an expensive operation and our method utilizes very few executions of this procedure relative to methods that exhaustively map 3D into C-space. Also, rather than using a complex or tailored solver, we relied on two options for efficient approximate IK solutions. The first involves 1-step along the gradient direction defined by the inverse of the manipulator's Jacobian evaluated at the current configuration. While this only yields valid solutions in a small region of convergence the computational cost is very low [10]. The second solver, a slightly more costly gradient based iterative method with random re-starts, is capable of handling configuration changes and singularities [13].

Our overall sampling policy combines the components in (10) and (11) to form the following multivariate Gaussian mixture model as the net proposal:

$$\pi_t = \pi_I + \pi_{II}. \quad (12)$$

The cost function, (1) is evaluated at a number of samples drawn from  $\pi_t$ , and the sample with minimum cost is output from our planner as the best next arm configuration. An overview of our proposed LTRA is presented below.

### Lost-Target Recovery Algorithm (LTRA)

1. Initialization: target is occluded at time  $t$ .
  - Estimate  $\hat{\mathbf{X}}_{t+1}^T$  by particle filter and obtain  $\mathbf{X}_t^c$  through FK.
  - For  $l=1, \dots, n_p$  find  $\mathbf{P}_{s,l}$  through updated map  $m$ , then obtain  $\mathbf{q}_{P_{s,l}}$  by IK.
  - Derive  $n_s$  samples from  $\pi_t$  in (12).
2. Cost function evaluation: for  $r=1, \dots, n_s$  do
  - Calculate (2) and (3) through Jacobian matrix at  $\mathbf{q}_r$ .
  - Calculate (4) and (7) at  $\mathbf{q}_r$ .
  - Calculate (8) through FK at  $\mathbf{q}_r$ .
  - Calculate  $\Gamma_r$  as the summation of the terms in (1).
3. Optimization: find  $\mathbf{q}_r$  with minimum  $\Gamma_r$  and set  $\mathbf{q}_{next} = \mathbf{q}_r$ .
  - Move to  $\mathbf{q}_{next}$  and acquire new sensory data.
  - If target is recovered, abort and resume tracking, otherwise continue.
4. Update: if the occluder is not thoroughly mapped, update the map,  $m$ , based on acquired sensory data.
  - Set  $t = t+1$  and go to 1.

## IV. EXPERIMENTS

Evaluation of a multi-faceted (non-analytically tractable) planning approach such as ours requires multiple repeated trials to test robustness to a variety of configurations. To efficiently validate the system we have implemented a detailed simulator of a Barrett WAM 7-DOF robot, a moving target, and an occluder with simple geometry. As shown in Fig. 5, each joint angle command issued by our planner results in a simulated image that is used to drive subsequent planning. Target tracking and occlusion analysis are also simulated, although for simplicity we assume error-free visual measurement is possible.

Multiple trials involving a variety of initial conditions of the arm were executed, and planning results are aggregated to compare methods. A single trial starts with the target recently lost from view due to occlusion. The planner provides a new robot pose, simulating arm motion to that pose, and then the simulation updates the target and occluder information accordingly. This is repeated until the target is re-acquired. The following presents numerical results of our method within this simulation environment.

### A. Simulation Results

We validate our approach against a number of

simple base-line strategies:

- *Pure pan-tilt* refers to a strategy that does not re-locate the arm, but simply moves the last two joints to orient the camera towards the target. This planner cannot make clearing motions, so will only restore object tracking when the target’s own motion moves it clear of the occluder.
- *Random motion* refers to a planner that randomly selects a set of joint angles within the C-space of the robot at each step.
- *Random sampler* begins with the same number of samples as the following two methods. These derived samples from a uniform distribution over C-space are weighed using our cost function.
- *LTRA-IK* refers to our complete approach including the Gaussian-mixture proposal based on shadow planes and our cost function. However, the promising closest point “seeds” are mapped from Cartesian space to C-space using a more computationally expensive gradient based IK.
- *LTRA-IJ* (Inverse Jacobian) refers to our complete approach including the Gaussian-mixture proposal based on shadow planes and our cost function. However, the promising closest points “seeds” are mapped from Cartesian space to C-space using the cheaper 1-step estimation based on the pseudo inverse of the robot’s Jacobian.

Table I lists the initial conditions used for each of the experiment-types in our evaluation. We have attempted to cover a range of reasonable joint configurations. Of note, the pose labeled “home” is a singular configuration for our simulated manipulator.

TABLE I. INITIAL CONDITIONS

Parameter	Terms	Quantity
$\mathbf{q}_{Elbow\ Down}$	$[q_i(0)], i = 1, \dots, 7$	[0 -1.57 0 1.57 -1.5 0 0]
$\mathbf{q}_{Home}$	$[q_i(0)], i = 1, \dots, 7$	[0 0 -1.3 0 0 -0.2 0]
$\mathbf{q}_{Elbow\ Up}$	$[q_i(0)], i = 1, \dots, 7$	[-1.57 0 -1.57 1.57 0.5 0.2 1]
$\mathbf{q}_{min}$	$[q_i]_{min}, i = 1, \dots, 7$	[-2.6 -2 -2.8 -0.9 -4.76 -1.55 -3]
$\mathbf{q}_{max}$	$[q_i]_{max}, i = 1, \dots, 7$	[2.6 2 2.8 3.1 1.24 1.55 3]
$\mathbf{X}_T(0)$	$[x(0)y(0)z(0)]_T$	[-1.4 5 0.2]
$\mathbf{V}_T$	$[\dot{x} \dot{y} \dot{z}]_T$	[0.1 0 0.02]
$\mathbf{X}_{Occluder}$	$[x_1 y_1 z_1;$	[-0.5 2 0.2;
	$x_2 y_2 z_2;$	-0.5 2 1.2;
	$x_3 y_3 z_3;$	0.5 2 1.2;
	$x_4 y_4 z_4]$	0.5 2 0.2]

TABLE II. SIMULATION RESULTS

Planning Strategy		Initial Configuration		
		Elbow Down	Home	Elbow Up
Pure Pan-tilt	Steps mean	27	27	27
	Steps SD*	0	0	0
	Dist.** mean	0.46	0.50	0.89
	Dist. SD	0	0	0
Random Motion	Steps mean	17.4	16.45	23.9
	Steps SD	11.42	14.78	16.01
	Dist. mean	71.01	65.59	99.91
	Dist. SD	51.08	67.26	72.5
Random Sampler	Steps mean	14.65	12.10	17.05
	Steps SD	10.03	7.62	8.68
	Dist. mean	54.54	43.71	65.37
	Dist. SD	41.64	34.45	35.82
LTRA-IK	Steps mean	5.25	20.65	6.45
	Steps SD	2.43	7.29	2.78
	Dist. mean	12.49	15.78	19.48
	Dist. SD	12.82	9.66	13.44
LTRA-IJ	Steps mean	12.05	10.85	20.6
	Steps SD	7.94	12.85	3.73
	Dist. mean	5.99	3.94	6.74
	Dist. SD	2.96	4.95	1.7

\*Standard Deviation

\*\*Euclidean distance in radians

The C-space search performed by our method primarily involves points nearby the initial robot position to find configurations that are useful for mapping and target re-acquisition. Therefore, performance around a joint singularity demonstrates the robustness of our approach.

Table II summarizes the results from running our simulator for twenty trials and averaging results, in each of the cases described. There are clear trends in both the number of steps required to restore tracking and the required travel distance. Fig. 4 presents a sample experimental trial with the “Elbow Down” initial configuration.

### B. Discussion

The *Pure Pan-Tilt* strategy requires the least motion of any planner, as it keeps the majority of the robot’s joints stationary, at the cost of waiting until the object reappears from behind the occluder for a large number of steps. The *Random Motion* and *Random Sampler* approaches both draw samples from the same uniform distribution, and only differ in the *Random Sampler’s* use of our cost function to select from the sample-set.

The latter prioritizes samples that lead to the target being re-acquired more quickly, validating our cost function. The *Random Sampler* and *LTRA* strategies employ the same cost function, but *LTRA* has the benefit of our adaptive proposal distribution, so the set of samples are focused around regions likely to yield useful views.

Finally, we draw the reader’s attention to the performance of the two variants of *LTRA*. There is a clear reduction in both the number of steps and the distance travelled. In many cases, our planner is able to re-acquire the target after making a small number of mapping motions, and then choosing a sample near one of the shadow planes, allowing rapid recovery of object tracking. The primary difference of these two strategies is in the strength of the IK utilized. The simpler 1-step solution (*LTRA-IJ*) expends much less computation, but can only recover solutions near the manipulator’s current position. The iterative solver (*LTRA-IK*) will find joint angles for any reachable configuration.

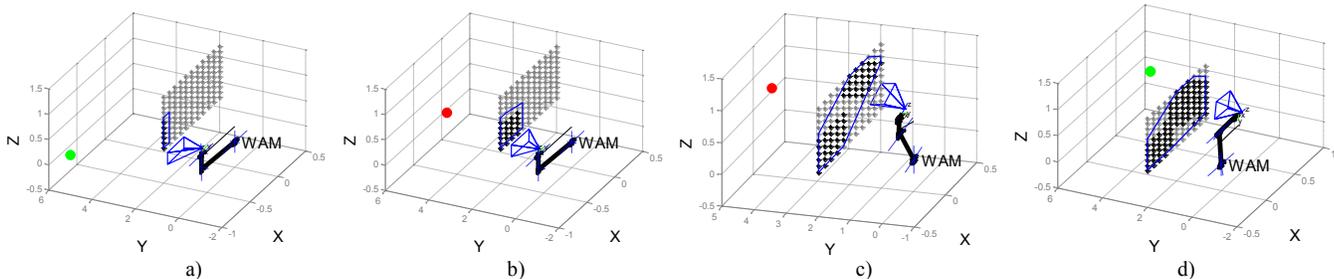


Fig. 4. Four steps of an experimental run for Elbow Down scenario given in Table 1 based on *LTRA-IJ*. a) Starting point, target is visible (green point). b) Step # 1 after occluder intervenes between robot and target (red point). Occluder is partially mapped. c) Step # 8 after occluder intervenes, occluder is almost mapped. d) Step # 12 after occluder intervenes, target is recovered (green point) and the occluder is mapped.



Fig. 5. a) View from eye-in-hand camera. Detected target (the yellow glove) is marked by a red circle and the detected known occluder edge is shown by a red line. b) View of experimental set-up.

Our results show that when the iterative solver is used, our method requires fewer steps, but produces paths with larger manipulator motions (i.e., each step is larger on-average). This is intuitive, since the more complete solver enables us to jump between promising regions within the workspace, while the 1-step method considers a larger number of small motions.

## V. CONCLUSION AND FUTURE WORK

This paper has described a method to regain visual tracking of a lost target for an eye-in-hand manipulator system. Our method captures many intuitive aspects of this problem, including the active construction of a map of the occluding object, and the use of this map to compute poses that allow the robot to look around the occluder. Our algorithm rarely employs expensive IK (once per shadow plane), and the remainder of our computations are completed efficiently in C-space.

As a result we are able to rapidly trade off searching for the target with acquiring further information about the occluder's extents. This rapid search approach is essential when tracking quickly moving targets. Our results have demonstrated that the disparate planning goals of our system can be effectively balanced using our multi-objective cost function.

Our future work will include advancing the occlusion map as well as the information-theoretic feedback to the planner that leads to effective mapping actions. We are investigating smoothing models for the occlusion probability within unobserved regions as well as more detailed modeling of occluder's extents. Finally, in previous work we have implemented a simple target re-acquisition strategy on a physical manipulator and performed hardware trials [10]

(Fig. 5). We are currently extending the presented simulation approach for demonstration on that platform for further validation of our approach.

## REFERENCES

- [1] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *Robotics & Automation Magazine*, IEEE, vol. 13; 13, pp. 82-90, 2006.
- [2] M. V. Danica Kragic, "Vision for Robotics," vol. 1, pp. 1, 2009.
- [3] G. Chesi, K. Hashimoto, D. Prattichizzo and A. Vicino, "Keeping features in the field of view in eye-in-hand visual servoing: a switching approach," *Robotics, IEEE Transactions on*, vol. 20; 20, pp. 908-914, 2004.
- [4] N. R. Gans, Guoqiang Hu, K. Nagarajan and W. E. Dixon, "Keeping Multiple Moving Targets in the Field of View of a Mobile Camera," *Robotics, IEEE Transactions on*, vol. 27, pp. 822-828, 2011.
- [5] B. J. Nelson and P. K. Khosla, "Strategies for increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance," *Int. J. Robotics Res.*, vol. 14, pp. 255-269, 1995.
- [6] Chi-Yi Tsai, Kai-Tai Song, X. Dutoit, H. Van Brussel and M. Nuttin, "Robust Mobile Robot Visual Tracking Control System Using Self-Tuning Kalman Filter," *Computational Intelligence in Robotics and Automation*, 2007. International Symposium on, pp. 161-166.
- [7] Yinghua Zhang, M. Rotea and N. Gans, "Sensors searching for interesting things: Extremum seeking control on entropy maps," in *Decision and Control and European Control Conference*, 2011. 50th IEEE Conference on, 2011, pp. 4985-4991.
- [8] Y. Yu and K. Gupta, "C-space Entropy: A Measure for View Planning and Exploration for General Robot-Sensor Systems in Unknown Environments," *The International Journal of Robotics Research*, vol. 23, pp. 1197-1223, December 01, 2004.
- [9] L. Torabi and K. Gupta, "An autonomous six-DOF eye-in-hand system for in situ 3D object modeling," *The International Journal of Robotics Research*, vol. 31, pp. 82-100, January 01, 2012.
- [10] S. Radmard, D. Meger, E. Croft and J. Little, "Overcoming occlusions in eye-in-hand visual search," in *American Control Conference*, 2012, pp. 4102-4107.
- [11] S. Radmard and E. A. Croft, "Approximate recursive bayesian filtering methods for robot visual search," in *Robotics and Biomimetics*, 2011. IEEE International Conference on, 2011, pp. 2067-2072.
- [12] L. Freda, G. Oriolo and F. Vecchioli, "Sensor-based exploration for general robotic systems," in *Intelligent Robots and Systems*, 2008. IEEE/RSJ International Conference on, 2008, pp. 2157-2164.
- [13] G. K. Singh and J. Claassens, "An analytical solution for the inverse kinematics of a redundant 7DoF manipulator with link offsets," in *Intelligent Robots and Systems*, 2010. IEEE/RSJ International Conference on, 2010, pp. 2976-2982.