2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Dynamic Sensor Planning with Stereo for Model Identification on a Mobile Platform

Jeremy Ma and Joel W. Burdick
California Institute of Technology, Pasadena, CA
jerma@caltech.edu, jwb@robotics.caltech.edu

Abstract— **This paper presents an approach to sensor planning for simultaneous pose estimation and model identification of a moving object using a stereo camera sensor mounted on a mobile base. For a given database of object models, we consider the problem of identifying an object known to belong to the database and where to move next should the object not be easily identifiable from the initial viewpoint. No constraints on the motion of the object nor the robot itself are assumed, which is an improvement on previous methods. Sensor planning is based on the selection of the control action that optimizes a cost metric based on information gain. Experimental results from the implementation of the method on a two-wheeled nonholonomic robot are presented to illustrate and validate the method.**

I. INTRODUCTION

This paper considers how a mobile robot equipped with a stereo camera system should best plan its motions so as to simultaneously identify and track moving objects in its environment. Our approach is an extension of our previous work ([1]) in which the motion of the object-to-be-identified was constrained to lie within the field of view of the stereo camera, and the motion of the robot was confined to a circular ring about the object. In this paper neither the object nor the mobile agent are constrained in their motion. In addition, appropriate approximations and simplifications have been introduced to enable real-time implementation on a modest computing platform.

This paper is organized as follows: Section II discusses related work and how our approach differs. Section III summarizes our approach, highlighting the information gain metric and associated computations. Section IV discusses the implementation details including extensions needed for real-time capabilities and how unconstrained motion of the object and mobile agent can be achieved. Section V presents our experimental results.

II. RELATED WORK

The problem of sensor planning for object identification has been investigated extensively over the past several years, more commonly referred to as an *active recognition* problem [2] [3] [4]. The approach is to use vision imagery (stereo in our case) to identify an object, determine the next sensor viewpoint should the object not be identifiable from the current view, and provide visual feedback for navigation during the identification task. The use of information theoretic concepts has been investigated as a possible solution.

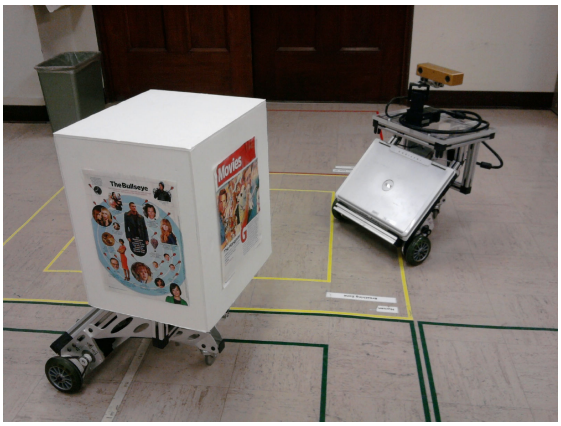

Fig. 1. The setup of our experiment is shown above where the mobile agent is equipped with a stereo-camera head mounted to a pan-tilt unit. The object to be identified is also mobile with the mobile component also a two-wheeled nonholonomic robot controlled by an external operator.

In [5], the authors use entropy as the optimization metric, and compute a discretized *entropy map* on the surface of a viewing sphere. At run time, sensor planning for object recognition is achieved by selection of the most informative view based on the precomputed entropy maps.

Paletta *et al.* [6][7] also use an information theoretic basis for their active recognition strategy that seeks to minimize expected entropy loss. Their chosen features are appearance-based parametric eigenspaces and the recognition process uses a sequential Bayesian approach. Their work does not include a procedure to estimate and track model pose, and is thus limited to identifying static objects.

Denzler and Brown [8] use mutual-information as their optimize criterion, and present a sequential decision-making process, using Monte Carlo sampling to approximate analytical solutions. State estimation is limited to model ID and no object pose estimation is considered. More recently, Eidenberger *et. al.* [9] presented a sequential Bayesian method for active object recognition based on a cost metric defined as the upper bound of the differential entropy. While their use of Gaussian mixture model approximations to prior and posterior distributions of the state variable allows for fast parametric updates, 6D pose estimation and object tracking capabilities are not explored.

Our work differs from these prior works by integrating 6D object pose estimation as a necessary part of the sensor-

planning problem. In addition, our experimental results show simultaneous sensor planning, pose estimation, and model identification for both moving object and robot.

III. APPROACH

Consider an autonomous robot equipped with a stereo camera pair mounted on a pan-tilt base and some means of on-board localization (*i.e.* odometry, SLAM, scan-matching, etc.). We assume that SIFT features ([10]) can be extracted from images as well as their corresponding 3D point locations from stereo. The agent is provided an existing database of L object models, $\{M_0, M_1, \cdots, M_{L-1}\}$, generated via a training phase as similarly described in [11], [12], and [13]. Each object is observed from various viewpoints on an equitorial ring surrounding the object. For each viewpoint, a set of SIFT features and their 3D locations are recorded and defined in an object-centered reference frame. SIFT features are chosen [14] due to their robustness in the areas of feature correspondence, re-detectability, and lighting effects. At the end of the training phase, the i^{th} model is defined by a set of N_i total database features, $\mathbf{B}_i = [\mathbf{b}_0^i \ \mathbf{b}_1^i \ \ldots \ \mathbf{b}_{N_i}^i]$.

Suppose that one database object is presented to the robot which is tasked with identifying the (possibly moving) object. The problem of where the robot should move if it is unable to correctly identify the presented model from the initial viewpoint is now considered.

A. Motion & Measurement Models

Let the state of the i^{th} model, M_i, at timestep k be defined as $\mathbf{X}_{k,i} = [\mathbf{x}_r^i \ \mathbf{x}_o^i] \in \mathbf{R}^6$, which represents the 6-DOF Euler parameters that define the object translational pose ($\mathbf{x}_r = [x\, y\, z]$) and orientation ($\mathbf{x}_o = [\alpha\ \beta\ \gamma]$) with respect to the camera-centered reference frame. Let $\mathbf{u}_{k-1} = [u_s\ u_\omega]_{k-1}$ be the control action (speed and angular rate control for a planar robot) executed by the robot at timestep $k-1$. Also, let $\mathbf{D}_k = [\mathbf{d}_0\ \mathbf{d}_1 \ \ldots \ \mathbf{d}_{n_k}]$ be defined as the set of n_k 3D-SIFT features measured at timestep k. We assume that the features have a one-to-one correspondence with a subset of database features for model M_i represented by the correspondence vector, $\mathbf{J}_i \in \mathbf{Z}^+$. The variable $\mathbf{J}_i(j) = l$ indicates that the j^{th} current measurement ($\mathbf{d}_j$) corresponds to the l^{th} database feature of the i^{th} model ($\mathbf{b}_{\mathbf{J}(j)}^i = \mathbf{b}_l^i$). While there exist many different methods of determining $\mathbf{J}_i$, our experiments use a comparison of the normed difference of the 128-element SIFT descriptor in the framework of Lowe's Best-Bin-First search algorithm [15].

Consider the diagram shown in Fig. 2 which shows the key reference frames and the spatial transformations between reference frames at one timestep and the next [1]. The transformation between the object frame and camera frame follows the recursive prediction equation:

$$G_{C_k,O_k} = G_{R_k,C_k}^{-1} G_{R_{k-1},R_k}^{-1} G_{R_{k-1},C_{k-1}} G_{C_{k-1},O_{k-1}} G_{O_{k-1},O_k} \quad (1)$$

The transform G_{O_{k-1},O_k} is derived from the object's motion model which we assume to be a random walk perturbed by noise at the velocity level. Similarly the transform G_{R_{k-1},R_k} is governed by the robot's motion model, chosen here as a basic wheel odometry model which incorporates the control action from the previous timestep, $\mathbf{u}_{k-1}$. The transforms G_{R_k,C_k} and $G_{R_{k-1},C_{k-1}}$ define the location and orientation of the camera relative to the robot at subsequent timesteps, which will be different because of the varying motions of the camera commanded by the pan-tilt base. Provided appropriate pan-tilt states are known at each timestep, these transforms can be easily determined. Lastly, the transformation $G_{C_{k-1},O_{k-1}}$ is obtained from the previous timestep's estimate of the i^{th} object's location, $\mathbf{X}_{k-1,i}$. This leads to a motion model for the state of the i^{th} object found by extracting the object Euler parameters from G_{C_k,O_k}:

$$\mathbf{X}_{k,i} = \mathbf{F}(\mathbf{X}_{k-1,i}, \mathbf{u}_{k-1}) + \eta \quad (2)$$

where the nonlinear function $\mathbf{F}$ incorporates the various frame transforms (the exact expression has been omitted for brevity) and η is Gaussian white noise.

The system's measurement model is derived in a similar manner. Since the data measurements $\mathbf{D}_k$ are received in the camera reference frame and the i^{th} object database features ($\mathbf{B}_i$) are described in an object-centered reference frame, the matched database features can be transformed into the camera reference frame via application of G_{C_k,O_k}:

$$\mathbf{D}_k = \begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{n_k} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_r^i + \mathbf{R}(\mathbf{x}_o^i)\mathbf{b}_{\mathbf{J}(0)}^i \\ \mathbf{x}_r^i + \mathbf{R}(\mathbf{x}_o^i)\mathbf{b}_{\mathbf{J}(1)}^i \\ \vdots \\ \mathbf{x}_r^i + \mathbf{R}(\mathbf{x}_o)^i\mathbf{b}_{\mathbf{J}(n_k)}^i \end{bmatrix}_k + \xi$$
$$\triangleq \mathbf{H}(\mathbf{X}_{k,i}, \mathbf{J}_i) + \xi \quad (3)$$

where $\xi \in \mathbb{R}^{3n_k \times 1}$ is Gaussian white measurement noise with covariance given by $\Sigma_m \in \mathbb{R}^{3n_k \times 3n_k}$ and $\mathbf{R}(\mathbf{x}_o^i)$ is the rotation matrix associated with G_{C_k,O_k}, explicitly stated as a function

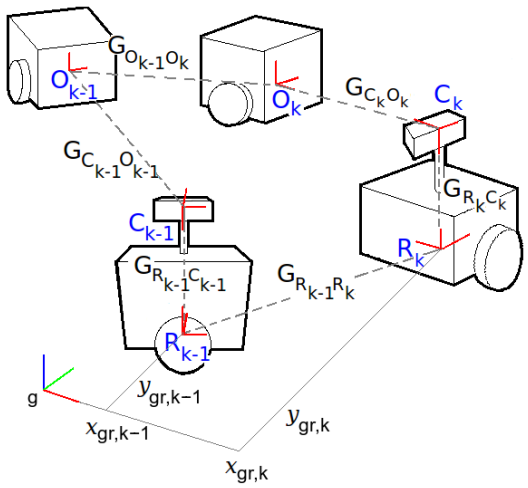

Fig. 2. There are three reference frames considered: the robot reference frame (R), the camera reference frame (C), and the object reference frame (O). For a given robot motion between timesteps $k-1$ and k, the various SE(3) frame transforms can be used to bring measurements from one frame into another to establish the motion and measurement models for the system.

[1]The notation used here is that the transform $G_{AB} \in SE(3)$ transforms a 3D point defined in frame B into the reference frame of A.

of the i^{th} object's orientation parameters. Note the use of the correspondence variable $\mathbf{J}_i$ in the measurement model, which assumes the correspondences are known or at least given. This issue is discussed more below.

B. Next-best View

The information gain metric, $I_{\mathbf{u}_k}(\cdot)$, used to optimize with respect to a future control action, $\mathbf{u}_k$, is chosen to be the difference between the current model entropy and the expected model entropy based on a hypothetical future action $\mathbf{u}_k$ and the data associated with that action, $\mathbf{D}_{k+1}$:

$$I_{\mathbf{u}_k} = H(M|\mathbf{D}_{1:k}, \mathbf{u}_{1:k-1}) - E_{\mathbf{D}_{k+1}}\left[H(M|\mathbf{D}_{1:k+1}, \mathbf{u}_{1:k})\right] \quad (4)$$

where $E_{\mathbf{D}_{k+1}}[\cdot]$ denotes expectation with respect to $\mathbf{D}_{k+1}$ and the entropy function $H(M|\mathbf{D}_{1:k}, \mathbf{u}_{1:k-1})$ is defined as:

$$H(M|\mathbf{D}_{1:k}, \mathbf{u}_{1:k-1}) = \quad (5)$$
$$-\sum_{i=0} P(M_i|\mathbf{D}_{1:k}, \mathbf{u}_{1:k-1}) \log P(M_i|\mathbf{D}_{1:k}, \mathbf{u}_{1:k-1})$$

where the subscript notation $1:k$ indicates the set of measurements, actions, etc. from timestep t_1 up to and including timestep t_k. Note that the entropy $H(M|\mathbf{D}_{1:k+1}, \mathbf{u}_{1:k})$ is similarly defined. $P(M_i|\mathbf{D}_{1:k}, \mathbf{u}_{1:k-1})$ represents the discrete probability of the i^{th} model, given the set of data measurements $\mathbf{D}$ from t_1 up to t_k and the set of control actions from t_1 up to t_{k-1}.

The optimal control action, which is the next-best sensing action, is that action which optimizes the information gain:

$$\mathbf{u}^* = \underset{\mathbf{u}_k}{argmax}\ I_{\mathbf{u}_k}\ . \quad (6)$$

The remainder of this section expands Eq. (4) into meaningful and computable expressions.

To determine $I_{\mathbf{u}_k}$, the current and expected model entropies must be found. Using Bayes' Rule, the model probability in Eq. (5) can be expressed as a function of the data likelihood:

$$P(M_i|\mathbf{D}_{1:k}, \mathbf{u}_{1:k-1}) = \quad (7)$$
$$\frac{\overbrace{p(\mathbf{D}_k|\mathbf{D}_{1:k-1}, \mathbf{u}_{1:k-1}, M_i)}^{data\ likelihood} \cdot P(M_i|\mathbf{D}_{1:k-1}, \mathbf{u}_{1:k-1})}{\sum_{j=0} p(\mathbf{D}_k|\mathbf{D}_{1:k-1}, \mathbf{u}_{1:k-1}, M_j) P(M_j|\mathbf{D}_{1:k-1}, \mathbf{u}_{1:k-1})}\ .$$

Note that the data likelihood $p(\mathbf{D}_k|\mathbf{D}_{1:k-1}, \mathbf{u}_{1:k-1}, M_i)$ can be further expressed by marginalizing over the state of the i^{th} model, $\mathbf{X}_{k,i}$:

$$p(\mathbf{D}_k|\mathbf{D}_{1:k-1}, \mathbf{u}_{1:k-1}, M_i) = \quad (8)$$
$$\int p(\mathbf{D}_k|\mathbf{D}_{1:k-1}, \mathbf{X}_{k,i}, \mathbf{u}_{1:k-1}, M_i) p(\mathbf{X}_{k,i}|\mathbf{D}_{1:k-1}, \mathbf{u}_{1:k-1}, M_i)\, d\mathbf{X}_{k,i}$$

where an approximate solution can be found if appropriate assumptions are made. By making a Markov assumption that past and future data are independent if the current state is known, the first term of the integral reduces to a normal distribution via the measurement model of Eq. (3); and by assuming a locally linear approximation to the motion model of Eq. (2), the second term of the integral also reduces to a normal distribution via an Extended Kalman Filter (EKF):

$$p(\mathbf{D}_k|\mathbf{D}_{1:k-1}, \mathbf{u}_{1:k-1}, M_i)$$
$$= \int N(\mathbf{H}(\mathbf{X}_{k,i}, \mathbf{J}_i), \Sigma_m) \cdot N(\mathbf{X}_{k,i}|\bar{\mu}_{k,i}, \bar{\Sigma}_{k,i}) d\mathbf{X}_{k,i}\,,$$

where $(\bar{\mu}_{k,i}, \bar{\Sigma}_{k,i})$ are the predicted state and covariance from the EKF. By making a similar linear approximation to the measurement model, i.e. $\mathbf{H}(\mathbf{X}_{k,i}, \mathbf{J}_i) \approx \mathbf{C}_{k,i} \cdot \mathbf{X}_{k,i}$, where $\mathbf{C}_{k,i} = \frac{\partial \mathbf{H}}{\partial \mathbf{X}_{k,i}}$, then the integral associated with the data likelihood can be resolved and reduces to:

$$\begin{aligned} p(\mathbf{D}_k|\mathbf{D}_{1:k-1}, \mathbf{u}_{1:k-1}, M_i) &= N(\mathbf{D}_k|\mu_{D_k,i}, \Sigma_{D_k,i}), \quad (9)\\ \mu_{D_k,i} &= \mathbf{C}_{k,i}\bar{\mu}_{k,i}\,,\\ \Sigma_{D_k,i} &= \mathbf{C}_{k,i}\bar{\Sigma}_{k,i}\mathbf{C}_{k,i}^T + \Sigma_m\,. \end{aligned}$$

The second term of the numerator in Eq. (7) can be resolved by noting that $P(M_i|\mathbf{D}_{1:k}, \mathbf{u}_{1:k}) = P(M_i|\mathbf{D}_{1:k}, \mathbf{u}_{1:k-1})$ since the anticipated control action $\mathbf{u}_k$ will not have a direct effect on the current model probability and can be safely omitted from the conditional dependence. Thus $P(M_i|\mathbf{D}_{1:k}, \mathbf{u}_{1:k})$ can be solved for recursively, provided that at the first timestep, the model probabilities are set to a priori values, i.e. $P(M_i|\mathbf{D}_0, \mathbf{u}_0) = P_{0,i}$.

Note that the calculation of the model probability (and hence the current model entropy), requires the estimation of the 6D pose of the i^{th} object, which is an integral part of computing the overall information gain.

The expected model entropy, $H(M|\mathbf{D}_{1:k+1}, \mathbf{u}_{1:k-1}, \mathbf{u}^+)$, can be computed in an analogous manner. However, due to the use of an expectation operator, the set of all valid control actions $\mathbf{u}^+$ (and the expected future data measurement associated with each action, $\mathbf{D}_{k+1}$) must be considered, which can lead to an intractable integral.

Using Monte Carlo approximations, the expected model entropy can be approximated by sampling N measurements from the future data likelihood of model M_i and comparing that measurement against all other models, as described by the following steps:

$$\begin{aligned} &E_{\mathbf{D}_{k+1}}[H^+] \quad (10)\\ &= \sum_i P(M_i|\theta, \mathbf{u}^+) \int H^+ \cdot p(\mathbf{D}_{k+1}|\theta, \mathbf{u}^+, M_i) d\mathbf{D}_{k+1}\\ &\approx \sum_i P(M_i|\theta, \mathbf{u}^+)\, \frac{1}{N} \sum_n^N H^+(\tilde{\mathbf{D}}) \end{aligned}$$

where $\theta = (\mathbf{D}_{1:k}, \mathbf{u}_{1:k})$, $H^+ \triangleq H(M|\theta, \mathbf{D}_{k+1}, \mathbf{u}^+)$ and $\tilde{\mathbf{D}} \sim p(\mathbf{D}_{k+1}|\theta, \mathbf{u}^+, M_i)$. However, it can be difficult to specify how a future set of 3D SIFT measurements should be sampled and how future feature correspondences achieved.

Note that future measurements $\tilde{\mathbf{D}}$ from model M_i sampled according to $p(\mathbf{D}_{k+1}|\theta, \mathbf{u}^+, M_i)$ are needed to determine the utility of the control action $\mathbf{u}^+$. That is, if control action $\mathbf{u}^+$ is executed, leading to measurements $\tilde{\mathbf{D}}$ (assuming M_i is the true model), the utility of action $\mathbf{u}^+$ to discriminate between all other models M_j is determined by the distribution

$p(\tilde{\mathbf{D}}|\theta,\mathbf{u}^+,M_j)$ – which is integral to the calculation H^+ in Eq. (10).

Realizing that $p(\tilde{\mathbf{D}}|\theta,\mathbf{u}^+,M_j)$ is identical to the data likelihood term of Eq. (7) with a simple index shift, the approximated results of Eq. (9) are applied to yield the following normal distribution:

$$p(\tilde{\mathbf{D}}|\theta,\mathbf{u}^+,M_j) = N(\tilde{\mathbf{D}};\mu^j_{\tilde{D}},\Sigma^j_{\tilde{D}}) \ . \tag{11}$$

Note that Eq. (11) has an inherent dependence on the correspondence vector $\mathbf{J}_j$ since $(\tilde{\mathbf{D}}-\mu^j_{\tilde{D}})$ is an array of residuals between the sampled future measurements from model M_i and the linearized expected location of that matched feature in model M_j.

If each SIFT measurement of the sampled set $\tilde{\mathbf{D}}$ is assumed mutually exclusive from all other features in the same set, then Eq. (11) can be re-written as:

$$p(\tilde{\mathbf{D}}|\theta,\mathbf{u}^+,M_j) = \prod_{m=0}^{W-1} N(\tilde{\mathbf{d}}_m|\mu^j_{\tilde{d}_m},\Sigma^j_{\tilde{d}_m}) \tag{12}$$

for $\tilde{\mathbf{D}} = [\tilde{\mathbf{d}}_0\ \tilde{\mathbf{d}}_1\ \cdots\ \tilde{\mathbf{d}}_{W-1}]$, $\mathbf{d}_m \in \mathbb{R}^3$.

Now if we consider the magnitude of the distance of the object from the robot in comparison to the nominal displacements between sampled features on the object from one timestep to the next, and realize the former is much larger in scale to the latter, then the magnitude of the residuals will be on roughly the same (small) scale and can be approximated as constant. Thus treating $\Sigma^j_{\tilde{d}_m}$ and $[\tilde{\mathbf{d}}_m-\mu^j_{\tilde{d}_m}]$ as both constant matrices, *i.e.*:

$$\Sigma^j_{\tilde{d}_m} = \Sigma_o \tag{13}$$

$$\left[\tilde{\mathbf{d}}_m-\mu^j_{\tilde{d}_m}\right] = \delta_o \quad \forall\tilde{\mathbf{d}}_m \in \tilde{\mathbf{D}} \tag{14}$$

then Eq. (11) reduces to:

$$p(\tilde{\mathbf{D}}|\theta,\mathbf{u}^+,M_j) = \frac{1}{(2\pi)^{W/2}|\Sigma_o|^{1/2}} \exp\left(-\frac{W}{2}\delta_o^T\Sigma_o^{-1}\delta_o\right) \tag{15}$$

where the variable W in the density $p(\tilde{\mathbf{D}}|\theta,\mathbf{u}^+,M_j)$ of Eq. (15) is the number of SIFT features in the predicted measurement set $\tilde{\mathbf{D}}$ (*i.e.*, it is the number of expected feature correspondences between the database features of models M_j and M_i that are expected under control action $\mathbf{u}^+$). Assuming W can be found, this approximation eliminates the need to invert large covariance matrices and sample an entire measurement set of 3D SIFT features, which is a vast improvement over our previous work, and increases computational speed by an order of magnitude.

One possible approach to estimate W is as follows. Consider a true model M_i viewed from various poses during training. For each pose, $\tilde{\mathbf{X}}$, the observed features of model M_i are compared against the database of features of model M_j for a total of T times. If out of T total comparison, the l^{th} database feature of the j^{th} model matches a feature in the i^{th} model τ times, then $P^j_{\mathbf{b}_l} = \frac{\tau}{T}$. $P^j_{\mathbf{b}_l}$ represents the probability of the l^{th} feature of model M_j corresponding to any feature in the i^{th} model at relative pose $\tilde{\mathbf{X}}$. By repeating this process until the set of relative poses used during training is exhausted, a probability lookup table is generated like the one displayed in Table I. Table I shows the values used for a true model M_2 compared against model M_0.

TABLE I

PROBABILITY LOOKUP TABLE: $M_i = M_2$ AND $M_j = M_0$

True Model: M_2						M_0		
Pose $[x$	y	z	α	β	$\gamma]$	$P^0_{\mathbf{b}_0}$	$\cdots$	$P^0_{\mathbf{b}_{N_0-1}}$
0.00	0.40	1.01	0.01	0.04	1.59	0.00	…	0.00
0.01	0.40	1.01	0.01	0.09	1.61	0.05	…	0.00
0.01	0.39	1.00	0.01	0.31	1.60	0.95	…	0.00
0.01	0.40	1.00	0.02	0.44	1.59	0.35	…	0.05
			⋮			⋮	⋮	⋮
0.06	0.39	1.02	0.01	0.03	1.58	0.10	…	0.00

For L database models, $L-1$ lookup tables must be generated per model, yielding a total of $L\cdot(L-1)$ tables. The variable W in Eq. (15) can be estimated from the lookup table for assumed true model M_i and candidate model M_j by finding that relative pose which best matches the predicted state of M_i for potential control action $\mathbf{u}^+$. Once found, the set of database features of M_j is sampled in accordance with the probability values $P^j_{\mathbf{b}_l}$. The net number of positively sampled features is equal to W and will be different for different models M_j.

IV. IMPLEMENTATION

The algorithm described above was implemented using the architecture of Fig. 3. We used a multi-threaded implementation to take advantage of multicore processors.

The next-best-view calculations are divided into three functions:

- *calcCurrentEntropy*$(\cdot)$, estimates the current model entropy and updates the model probabilities.
- *calcPossibleControlActions*$(\cdot)$ generates a list of valid control actions, and can incorporate obstacles or other motion constraints into the planning process.
- *calcExpectedEntropy*$(\cdot)$ uses the set of possible actions produced by *calcPossibleControlAction*$(\cdot)$, to evaluate Eq. (15), utilizing sets of probability lookup tables.

All three functions use the output of the Extended Kalman Filter (Bayes' Filter), which is implemented as a separate block. The Extended Kalman Filter output is also needed for the evaluation of the integral in Eq. (8).

The use of the pan-tilt-unit (PTU) control loop is one of the additions to our previous work[1]. The independent movement of the camera on the PTU allows the robot to take more optimal paths toward a goal location without the added constraints of keeping the object in the field of view at all times while simultaneously satisfying the vehicle's nonholonomic wheel constraints.

The PTU control loop operates in two states: a "tracking state" and a "scanning state". The tracking state keeps the object origin in the camera field of view once it is detected. Because the model identity is not known with certainty at the

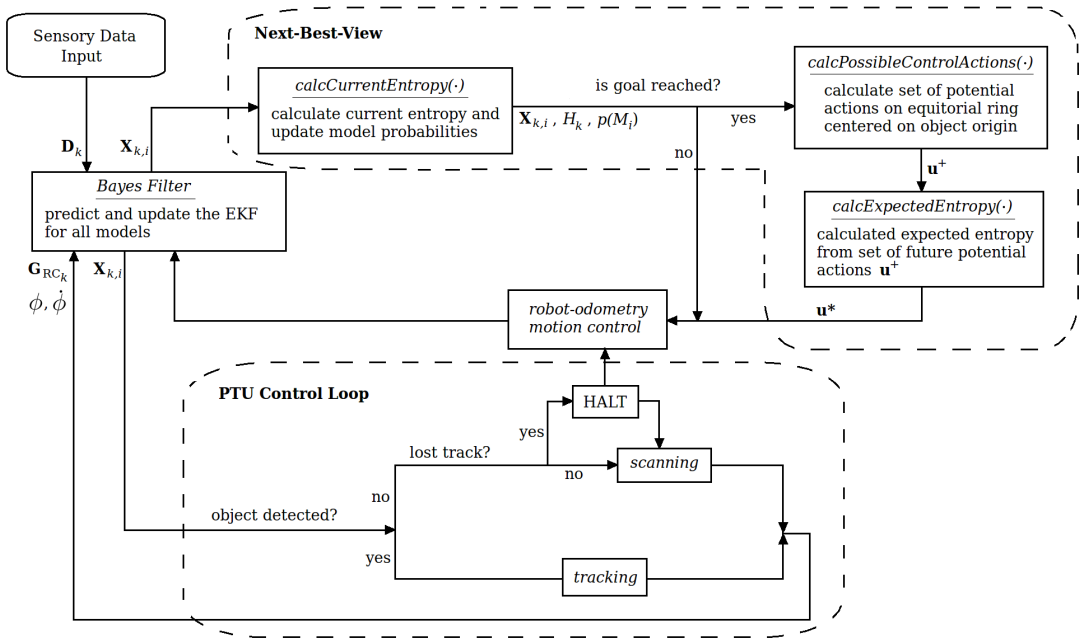

Fig. 3. Block diagram of the algorithm implementation uses in the experiments. Note that the information gain is calculated only when the system state reaches the goal determined by the optimal control action $\bar{\mathbf{u}}^*$. The pan-tilt-unit (PTU) control block is implemented as a separate thread.

time of initial detection, the location of object's origin is a vague concept: an Extended Kalman Filter is applied to each of the plausible models in the database, so that there are up to L different origins. Hence, the average origin of all models (weighted by model probability) is chosen as the origin for the purposes of tracking. Denoting the weighted origin coordinates as $\mathbf{P}_{obj} = \begin{bmatrix} x_{obj} & y_{obj} & z_{obj} \end{bmatrix}^T$. the projection of the origin on the image plane, $\mathbf{p}_{obj} \in \mathbb{R}^2$, is defined as:

$$\mathbf{p}_{obj} = \begin{bmatrix} x_{img} \\ y_{img} \end{bmatrix} = \begin{bmatrix} \frac{f \cdot x_{obj}}{z_{obj}} + c_x \\ \frac{f \cdot y_{obj}}{z_{obj}} + c_y \end{bmatrix} \tag{16}$$

where f is the focal length of the camera and (c_x, c_y) the image center. A simple proportional feedback law was used to control the pan angle slew-rate:

$$u_{pan} = -K \cdot \delta_{pan} = -K(c_x - x_{img}) = -K\frac{f \cdot x_{obj}}{z_{obj}} \tag{17}$$

A similar method can be used for PTU tilt, though tilt control was not necessary in the experiments described below.

The "scanning" PTU control state moves the camera head in a scanning pattern, and is typically deployed during the initial startup of the algorithm when the robot is searching for any known object, or when the robot loses an existing model track. In the latter case, the robot stops (HALT in Fig. 3) until the track is re-established. Depending on the re-initialized object state, a new iteration of the next-best-view calculation is carried out, or the last control action interrupted by the loss of the track is resumed.

V. EXPERIMENTAL RESULTS

Our method was implemented on an Evolution Robotics, Inc.^{TM}, ER-1 robot equipped with a PointGrey^{TM} Bumble-Bee2 color stereo-camera (image resolution of 320×240 pixels) mounted on a Directed Perception PTU-D46-17 pan-tilt unit (accuracy of 0.0514°in pan and tilt). All computations were carried out on a laptop computer (Intel^R Core2 Duo^R 2.40GHz processor) running Linux. The algorithm was written in C/C++ using the Intel OpenCV library.

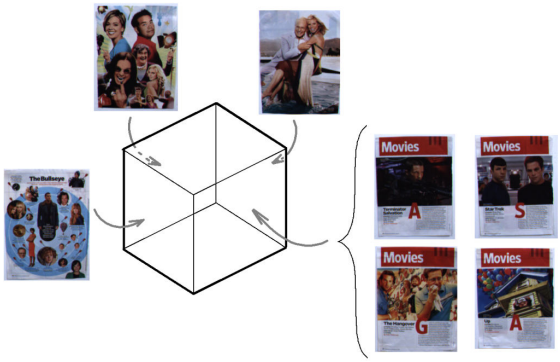

Fig. 4. The experimental data base consists of four box-like models. Each model had 3 identical faces, while the fourth face was unique to each model. Magazine pages were used as the model faces.

The object database consisted of 4 boxes, whose four sides were each covered with a distinct pattern (a magazine cutout). For each experimental run, a box was attached to another ER-1 robot in order to enable object motion. The on-board wheel odometry of the second robot was used as a reference in analyzing the pose estimates from the experiment. Fig. 4 shows the four different models used. All 4 models shared 3 identical faces, with only the last face being different for each model. This choice increased the difficulty of the model identification process, and also forced the robot to carry out sensor-planning actions. Moreover, the

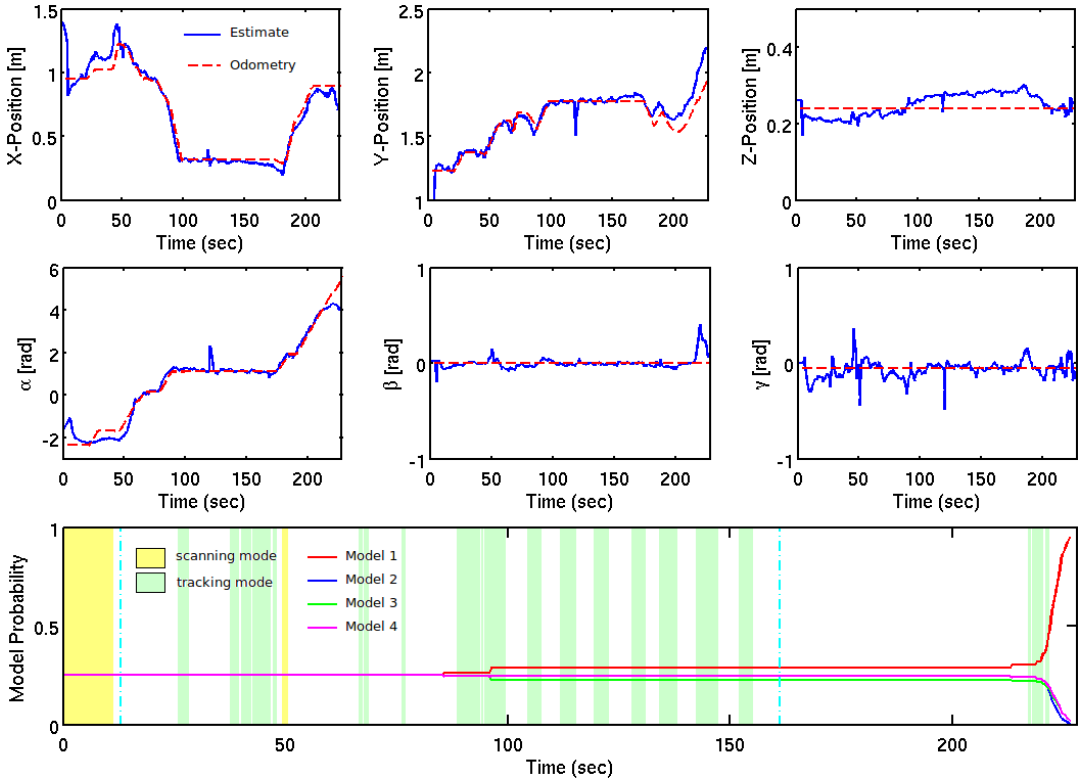

Fig. 5. The object position and orientation estimates (as defined in the global ref. frame) are respectively shown in the top and second rows. The estimated pose is shown by the solid-blue line and the object reference pose (as estimated by odometry on the moving object) is shown in the dotted-red line. The third row shows the model probabilities versus time. The yellow and green bars indicate the time periods during which the PTU control loop was actively *scanning* or actively *tracking*, respectively. The cyan dotted lines indicate moments at which a new sensor planning action was commanded and executed.

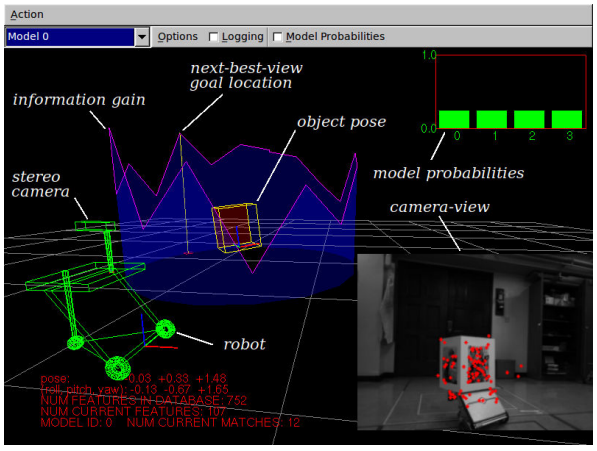

Fig. 6. An annotated screenshot of the visual interface developed for our experiment. The mobile robot is shown in green and the mobile object is in the center of the equitorial ring.

second robot was remotely controlled by a human operator in such a manner so as to prevent the discriminating model face from being observed by the robot. This strategy required repeated sensing and planning operations on the part of the robot until the model probability of one model peaked to near 100% certainty.

In the first trial, Model 1 was selected as the true model. Fig. 5 shows the estimated object poses as a function of time[2]. Plotted against the pose estimates are the mobile object's pose from on-board wheel odometry. Note some error spikes in the pose estimates which can be attributed to feature mismatches.

The bottom row of Fig. 5 plots the model probabilities against time to indicate the history of model confidence throughout the experiment. Note that initially all model probabilities stay uniform, as the discriminating face of the object has yet to be observed. As continued sensor planning actions are executed (indicated by the cyan dotted lines), the discriminating face of the object is eventually seen (at $t = 160s$) and the true model is identified. The yellow and green stripes indicate the time periods under which the PTU control loop was actively *scanning* (yellow) or actively *tracking* (green); all other time periods indicate passive tracking of the control loop.

The experiment was repeated with other models chosen as the true model, the results of which are shown in Fig. 7 (6-DOF pose estimates have been omitted for brevity). As a result of sensor planning actions, the true model is eventually identified in each experiment by bringing the discriminating face of the object into the robot's field of view.

Fig. 6 shows the visual interface developed for our system. The equitorial ring is centered on the object with the information gain plotted at various locations along the ring.

[2]While multiple pose estimates exist for all plausible models in the database, only the pose estimates for the true model are shown.

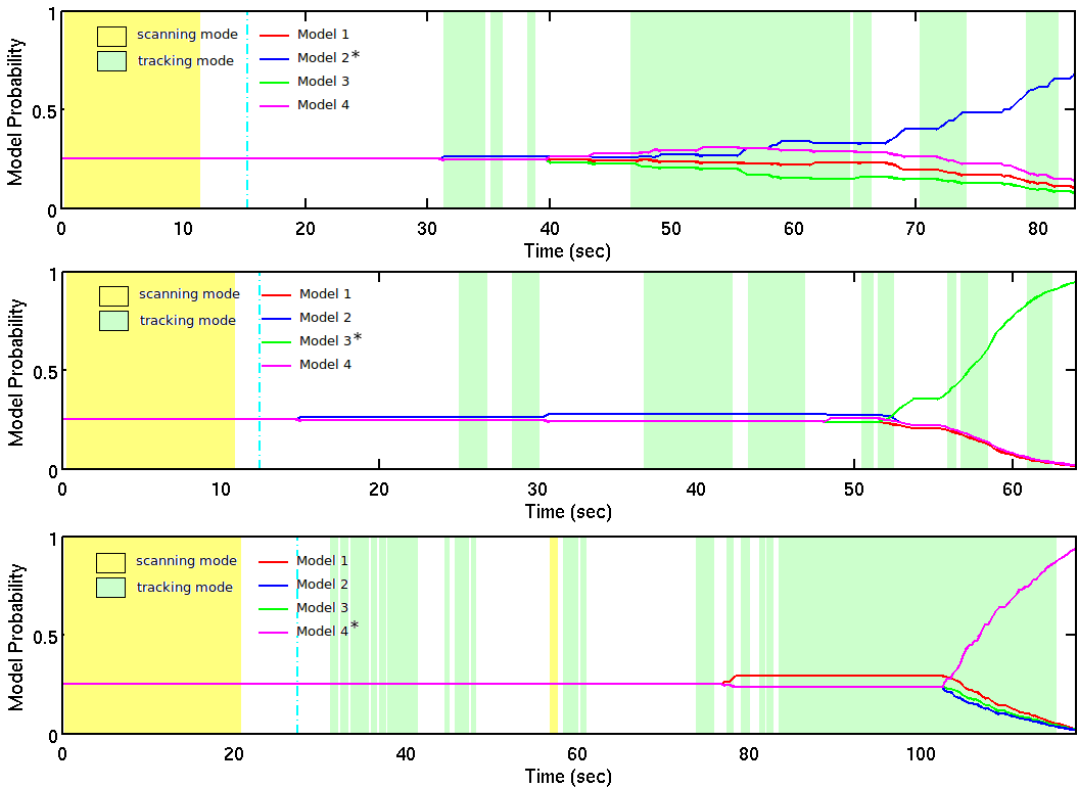

Fig. 7. Model probabilities are plotted against time for three additional test experiments, with different models chosen as the true model (as shown by the asterisk).

The peak information gain is denoted by the vertical yellow line, and is marked as the future goal location.

VI. CONCLUSIONS AND FUTURE WORK

This paper developed a method for dynamic sensor planning for model identification. We showed how 6-DOF pose estimation and tracking can be integrated into the optimization of the information gain metric. This derivation allows for the robot to naturally select the motions and sensing actions which are needed to discriminate an evading object. We also developed sensible approximations that allowed for a convenient separation of the computationally burdensome parts of the method into an off-line structure that could be stored in a simple look-up table format. Our experiments validated that our the multi-threaded control system can execute in real-time on a modest laptop computer. Future work will extend our proposed method to numerous database objects and consider the use of this technique as a part of an overall strategy for visual search of moving objects.

REFERENCES

[1] J. Ma and J. Burdick, "Sensor planning for object pose estimation and identification," in *In Proc. of the IEEE Int. Workshop on Robotic and Sensors Environments*, 2009 (to appear).

[2] F. G. Callari and F. P. Ferrie, "Autonomous recognition: Driven by ambiguity," in *In Proc. of the Conf. on Computer Vision and Pattern Recognition*, 1996, pp. 701–707.

[3] E. Rivlin, Y. Aloimonos, and A. Rosenfeld, "Purposive recognition: An active and qualitative approach," in *In Proc. of the SPIE*, 1991, pp. 225–240.

[4] J.-Y. Herve and Y. Aloimonos, "Exploratory active vision: theory," in *Proc. of the Conf. on Computer Vision and Pattern Recognition*, 1992, pp. 10–15.

[5] T. Arbel and F. P. Ferrie, "Entropy-based gaze planning," in *In Proc. of the 2nd IEEE Workshop on Perception for Mobile Agents*, 1999.

[6] L. Paletta, M. Prantl, and A. Pinz, "Learning temporal context in active object recognition using bayesian analysis," in *Proc. of the 15th Int. Conf. on Pattern Recognition*, vol. 1, 2000, pp. 695–699.

[7] L. Paletta and A. Pinz, "Active object recognition by view integration and reinforcement learning," *Robotics and Autonomous Systems*, vol. 31, pp. 71–86, 2000.

[8] J. Denzler and C. Brown, "Information theoretic sensor data selection for active object recognition and state estimation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 145–157, 2002.

[9] R. Eidenberger, T. Grundmann, W. Feiten, and R. Zoellner, "Fast parametric viewpoint estimation for active object detection," in *Proc. of the Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, 2008, pp. 309–314.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, pp. 91–110, 2004.

[11] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *Int. J. Computer Vision*, vol. 66, no. 3, pp. 231–259, 2006.

[12] J. Ponce, S. Lazebnik, F. Rothganger, and C. Schmid, "Toward true 3d object recogntion," in *In CVPR Workshop on Generic Object Recognition and Categorization, IEEE Computer Society*, 2004.

[13] H. Najafi, G. Yakup, and N. Nassir, "Fusion of 3d and appearance models for fast object detection and pose estimation," in *Asian Conf. on Computer Vision*, vol. 3852, 2006, pp. 415–426.

[14] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3d objects," *Int. J. of Computer Vision*, vol. 73, no. 3, pp. 263–284, 2007.

[15] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the Int. Conf. on Computer Vision*, vol. 2, 1999, pp. 1150–1157.