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

A Probabilistic Framework for Stereo-Vision Based 3D Object Search with 6D Pose Estimation

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

Abstract— **This paper presents a method whereby an autonomous mobile robot can search for a 3-dimensional (3D) object using an on-board stereo camera sensor mounted on a pan-tilt head. Search efficiency is realized by the combination of a coarse-scale global search coupled with a fine-scale local search. A grid-based probability map is initially generated using the coarse search, which is based on the color histogram of the desired object. Peaks in the probability map are visited in sequence, where a local (refined) search method based on 3D SIFT features is applied to establish or reject the existence of the desired object, and to update the probability map using Bayesian recursion methods. Once found, the 6D object pose is also estimated. Obstacle avoidance during search can be naturally integrated into the method. Experimental results obtained from the use of this method on a mobile robot are presented to illustrate and validate the approach, confirming that the search strategy can be carried out with modest computation.**

Fig. 1. Our experiments used an Evolution Robotics ER-1 (a two-wheeled differential drive robot) equipped with a pan-tilt unit and a stereo camera.

I. INTRODUCTION

This paper considers the problem of 3D object search with pose estimation using a single stereo camera mounted on the pan-tilt unit (PTU) of a mobile robot. Our approach integrates several useful features in a common framework: a two-scale search strategy, a grid-based probability map that governs the search process, a recursive Bayesian map updating process, 6D pose estimation of the found object, and a simple integration of obstacle avoidance into the search planning method.

This paper is organized as follows: Section II discusses related work and how our approach differs. Section III summarizes our approach, highlighting the global and local search methods and the recursive Bayesian update equations used to maintain the probability map which governs the searching and sensing actions. Section IV discusses the details of our implementation with experimental results presented in Section V. Section VI discusses our conclusions and recommendations for future work.

II. RELATED WORK

The problem of object search has been considered as an element of sensor planning research. Ye and Tsotsos [1] developed one of the first systematic frameworks for object search that incorporated both sensor planning and object recognition. Their method was developed for a two-wheeled robot equipped with a pan-tilt-zoom camera and a laser eye. Their spherically arranged training data set encodes the probability that a given sensor movement on a sphere surrounding the object will improve detection. Their computationally expensive method can be tedious to implement given its need for the experimental construction of a detection function for all sensing parameters (pan, tilt, zoom, robot orientation) under various lighting conditions, object orientations, and background effects. Furthermore, the object recognition function was limited to a 2D technique using a blob finder based on pixel intensity.

Saidi *et al.* [2] [3] extend the work of Ye and Tsotsos to a humanoid robot where object recognition is carried out via 3D SIFT features ([4]). They present a visual attention framework that relies upon pan-tilt-zoom capabilities to generate 3D data of the sensed environment. They formulate search as the problem of optimizing sensor actions and trajectories with respect to a utility function. Their utility function incorporates target detection probability, new information gain, and motion cost. A visibility map similar to the sensed sphere of [1] filters uninformative sensing actions. While their approach is a significant improvement on the work of [1], the visibility map calculations are computationally expensive and their utility function lacks a formal Bayesian framework.

The probabilistic approach used by Chung and Burdick [5] to solve an abstract object search problem provides the Bayesian framework lacking in [3]. They develop a recursive Bayes' Filter for updating the probability of object existence in each cell of a grid map, and various different search strategies are considered in simulation with the *saccadic* search method yielding the minimum average search time – an approach that mimics the search patterns in human visual attention ([6]). Nonetheless, their method must be further developed for any specific implementation.

Petersson *et al.* in [7] considered the problem of object

search in the context of grasping and manipulation. They used a support vector machine for object recognition on a robotic platform equipped with an arm, a laser scanner, sonar, a torque sensor, and a color camera. Once recognized, the object is tracked in a *window of attention*. Though the problem of object search *with* manipulation is addressed, a crude object recognition system is used, and object pose estimation is limited to the process of aligning the current object image with a predefined reference image – an approach that works only on piecewise planar objects in positions that match the reference image and pose.

Building and improving upon the work of Petersson *et al.*, Ekvall *et al.* [8] and Lopez *et al.* [9] decomposed the object search problem into global and local search stages. Their coarse global search employed Receptive Field Coocurrence Histograms ([10]) to identify potential object locations. A mobile robot equipped with laser, sonar, and a pan-tilt-zoom camera then zooms into each hypothesized location to apply a localized object search algorithm (based on SIFT features). An a priori map built via SLAM is used to establish likely locations of known objects. Navigation is restricted to planning over a graph of pre-determined "free-space" nodes. This approach simplifies the methods of [1] and [3] and allows for simultaneous search of multiple objects. However, their approach is limited to 2-dimensional objects whose pose is crudely approximated by a single laser scan point in [8] and later moderately refined in [9] to a distance measure based on comparing the number of occupied pixels in the image against a reference image. Furthermore, much prior information is assumed given or computed offline (*e.g.* the SLAM-based map and the set of navigation nodes).

The approach presented in this paper improves upon the work of [1] by using a 3D object detector and also simplifies the method of [3] by replacing the computationally expensive 3D visibility map and rating function with a global and local search technique that updates the probability map. While [8] and [9] also make use of a global/local search decomposition, we add accurate 6D pose estimation of the detected object. Furthermore, our method incorporates the Bayesian framework developed by [5] for a 2D world and extends it to a real robotic system in a 3D world by coupling the 2D mobility of our robot with a pan-tilt unit to achieve object detection at various elevations. Our experimental results demonstrate robust object search with 6D pose estimation coupled with path planning and obstacle avoidance can be achieved with a single stereo camera sensor, an improvement on the vast array of sensors used in [1], [7], [9], and [10].

III. APPROACH

We assume that a (possibly nonholonomic) mobile robot is equipped with a stereo camera mounted on top of a pan-tilt unit (*e.g.*, see Fig. 1) and possesses a localization scheme (*e.g.* vision-based SLAM, on-board odometry, indoor GPS, etc.). While the position of the object to be found is not known to the robot, we assume that the object's position is stationary throughout the search process. The object is assumed to be learned during a *training phase* in which various viewpoints of the object are considered in stereo. Registered features are recorded in the object reference frame along with a reference image of the object taken at each viewpoint. The resultant set of recorded features and images constitute a feature database specific to the object being searched. We allow for multiple objects to be trained and stored, contributing to a dictionary of known objects in the robot's memory. This facilitates the setup for object search by allowing the user to simply specify which object in the dictionary to search for.

A. Probability Map

Following the Bayesian framework of [5], we assume that the workspace can be approximated by a grid-based map divided into cells whose coordinates are known in a global coordinate frame. Let $c_{i,j}$ represent the $(i,j)^{th}$ cell of the discretized search space and let $y^k_{i,j} \in \{0,1\}$ be a stochastic binary variable indicating a positive or negative detection of the object in cell $c_{i,j}$ at timestep k. Let $\mathbf{Y}_{1:k}$ denote the set of binary measurements from timestep 1 up to and including timestep k: $\mathbf{Y}_{1:k} = [y^1\, y^2 \cdots y^k]$. Let $h^k_{i,j} \in \{0,1\}$ define a hypothesis of object existence in cell $c_{i,j}$ at timestep k, such that $h^k_{i,j} = 1$ is the hypothesis that the object exists in $c_{i,j}$ at time k and $h^k_{i,j} = 0$ the hypothesis that it does not.

Suppose that a known object is placed somewhere in the workspace, but its location is unknown to the searching robot. The probability (or belief) that the object resides in $c_{i,j}$ at time k given the binary measurements, $\mathbf{Y}_{1:k}$, is $P(h^k_{i,j} = 1|\mathbf{Y}_{1:k})$. Applying Bayes' Rule, the cell probability can be expanded as:

$$P(h^k_{i,j} = 1|\mathbf{Y}_{1:k}) = \frac{P(y^k_{u,v}|h^k_{i,j} = 1, \mathbf{Y}_{1:k-1})P(h^k_{i,j} = 1|\mathbf{Y}_{1:k-1})}{P(y^k_{u,v}|\mathbf{Y}_{1:k-1})} \tag{1}$$

where $y^k_{u,v}$ indicates the measurement at time k was of the $(u,v)^{th}$ cell which is not necessarily the same as $c_{i,j}$.

The first term of the numerator defines the sensor detection model. Many different forms of detection models exist and vary depending on the type of sensor and object recognition algorithm used. The detection model of [5] is appropriate for object detection on a grid-based map and is thus used in our approach when updating the probability map after a global or local search:

$$P(y^k_{u,v}|h^k_{i,j} = 1, \mathbf{Y}_{1:k-1}) = \tag{2}$$
$$\begin{cases} P(y^k_{u,v} = 0|h^k_{i,j} = 1, \mathbf{Y}_{1:k-1}) = \beta & (u,v = i,j) \\ P(y^k_{u,v} = 1|h^k_{i,j} = 1, \mathbf{Y}_{1:k-1}) = 1-\beta & (u,v = i,j) \\ P(y^k_{u,v} = 0|h^k_{i,j} = 1, \mathbf{Y}_{1:k-1}) = 1-\alpha & (u,v \neq i,j) \\ P(y^k_{u,v} = 1|h^k_{i,j} = 1, \mathbf{Y}_{1:k-1}) = \alpha & (u,v \neq i,j) \end{cases}$$

where α and β represent the detection error probabilities for false alarms and missed detections, respectively, and are dependent on the sensor quality and the recognition modality (*e.g.* SIFT, support vector machine, color histogram, etc.).

Since we assume that the object is stationary, the second term of the numerator can be simplified: $P(h^k_{i,j} = 1|\mathbf{Y}_{1:k-1}) = P(h^{k-1}_{i,j} = 1|\mathbf{Y}_{1:k-1})$, which equals the cell probability value at

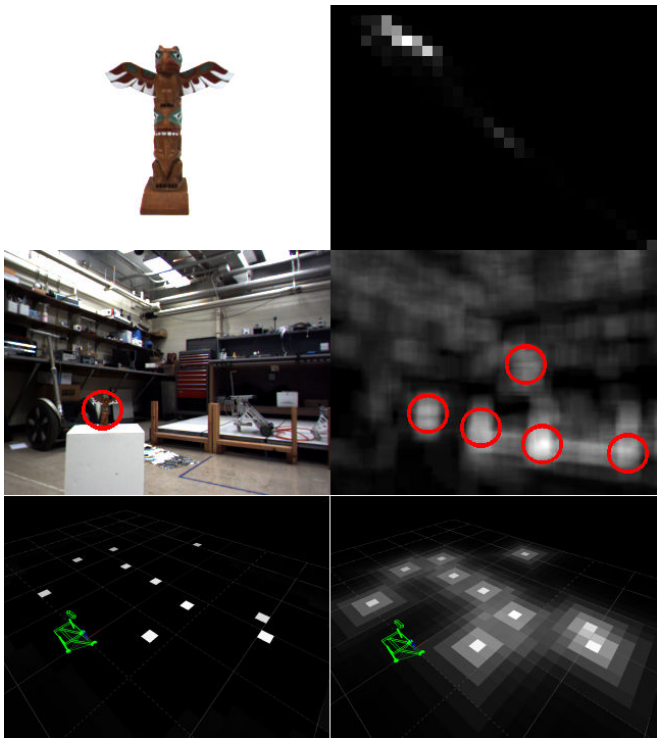

Fig. 2. The coarse global search uses a color histogram of the object to identify likely regions in the current image that resemble the object. Shown in the top left is a sample object. The top right shows the corresponding color histogram in RG (red-green) space. The middle left shows one image from a stereo pair (with the true object location circled in red) while the middle right shows the back projection of histogram comparison to subwindows of the current image. Probable object locations in the image are shown in red circles. The re-projected stereo point estimates of the cluster centers back onto the grid-based probability map are shown in the bottom left. The bottom right shows the effect of the variable β parameter to increase the likelihood of object existence in neighboring cells.

the previous time step, thus enabling the recursive nature of the probability map update. Note that at initial run time when $k=0$, $P(h^k_{i,j}=1|\mathbf{Y}_{1:k-1})=P_0$, an initial prior distribution on the probability map (see Section IV).

Lastly, the denominator of (1) is obtained by marginalizing over the object cell location:

$$P(y^k_{u,v}|\mathbf{Y}_{1:k-1})=\sum_{m,n}P(y^k_{u,v}|h^k_{m,n}=1,\mathbf{Y}_{1:k-1})P(h^k_{m,n}=1|\mathbf{Y}_{1:k-1}) \tag{3}$$

With the various terms of (1) expressed, a Bayesian recursion scheme is thus resolved for each cell $c_{i,j}$ of the probability map, M_p. In effect, the recursive update of the probability map redistributes probability mass from the explored cells where no object is found to the unexplored cells.

B. Global Search Method

As noted by [8], the search process can be divided into a global search based on a coarse detection methodology which operates over longer ranges, and local detailed search that operates well at close range. We implement a common global search method that utilizes color histograms.

During a training phase (as mentioned earlier), color images of the object (see top left image of Fig. 2) taken from various viewpoints are used to construct a cumulative model histogram using the red and green image channels, (see top right image of Fig. 2)[1]. The model histogram is stored in a dictionary which can be queried during search. At the start of a search, the robot scans its environment using a prescribed set of pan-tilt angles that cover a viewing hemisphere. For each view, the image is back projected against the model using a fixed scanning window size (see middle two images of Fig. 2). The resultant image is normalized to yield a distribution over the image pixels of the conditional probability that a given pixel is a member of the model histogram. Next, the N_{hist} highest probability pixel clusters are selected (the red circles in the middle right image of Fig. 2). Using sparse stereo the 3D locations to the peak probability values, $P'_n\ \forall n \in \{1,\cdots,N_{hist}\}$, are estimated and re-projected into cells of the grid-based probability map, M_p (see bottom left image of Fig. 2). Each peak probability location, P'_n, is then treated as an individual measurement of the cell, $c_{u,v}$ to which it is projected; *i.e.* we set $P(y^k_{u,v}=1|h^k_{u,v}=1,\mathbf{Y}_{1:k-1})=P'_n$ and update M_p according to (1). This process is repeated for all discrete pan angles during the global search stage.

Because the global search is a coarse detection model, we use a variable β parameter, initialized to $1-P'_n$ and increased exponentially for neighboring cells of $c_{u,v}$. This has the effect of incorporating high likelihood of object existence in a neighborhood of cells, as opposed to a single cell if β were kept constant (see bottom right image of Fig. 2).

C. Local Search Method with 6D Pose Estimation

A more refined local search is used when the robot lies within a fixed distance of a peak in the search probability map. Our local sensing method uses SIFT feature matching, optimized by using a kd-tree with Lowe's Best-Bin-First Search schema as described in [11]. However, as opposed to the methods in [8] and [9], which also use SIFT features for object identification, our local search procedure can also estimate the 6D pose of the identified object.

During the training phase described above, object model SIFT features are recorded from multiple viewpoints around the object, with the 3D location (in an object-centered reference frame) of the point SIFT feature estimated from sparse stereo. Let $\mathbf{B}=[\mathbf{b}_0\ \mathbf{b}_1\ \cdots\ \mathbf{b}_W]$ denote the set of SIFT stereo features, where $\mathbf{b}_i=[b_x\ b_y\ b_z\ \tilde{\mathbf{d}}]$ and $\tilde{\mathbf{d}}\in\mathbb{R}^{128}$ is the 128-dimensional SIFT feature descriptor.

During a local search, the measurement at time t_k consists of n_k 3D SIFT features: $\mathbf{D}_k=[\mathbf{d}_0\ \mathbf{d}_1\ \cdots\ \mathbf{d}_{n_k-1}]$ where $\mathbf{d}_i=[d_x\ d_y\ d_z\ \tilde{\mathbf{d}}]$. Lowe's Best-Bin-First Search matching scheme is then used to identify a correspondence $\mathbf{J}\in\mathbb{Z}^+$ between the measurements $\mathbf{D}_k$ and database features $\mathbf{B}_k$ such that $\mathbf{J}(i)=j$ indicates a match between $\mathbf{d}_i$ and $\mathbf{b}_j$.

While the object is assumed to be stationary, its position and orientation relative to the robot is not given a priori. This requires an estimator that accurately estimates the position and orientation of the object for any possible initial pose. On

[1]While color histograms using red, green, and blue channels had been considered, similarly adequate results were found from using only the red and green channels with slight improvements in computational speed.

this front, we use an Extended Kalman Filter (EKF) with a state vector $\mathbf{X}_k = [\mathbf{x}_r\ \mathbf{x}_o] \in \mathbb{R}^6$ defined as the 6D object pose as seen in the camera reference frame ($\mathbf{x}_r = [x\ y\ z]$ denotes the translational pose and $\mathbf{x}_o = [\theta\ \rho\ \phi]$ the object *z*-*y*-*x* Euler angles).

The object motion model was chosen as:

$$\mathbf{X}_k = \mathbf{X}_{k-1} + \eta \tag{4}$$

which describes a static object with some zero-mean Gaussian system noise, η – a necessary term to allow for the object state to move and converge on an accurate pose for any given object orientation and position. Since in our implementation the robot is paused during the local search, the robot dynamics are not integrated into the motion model. Should object pose tracking be needed while the robot moves and/or pans/tilts the camera while running the EKF, the motion model can be easily augmented to include the robot dynamics.

Assuming that the set of unmatched measurements in $\mathbf{D}_k$ have been removed, leaving m matches, the EKF measurement model can be defined as:

$$\begin{bmatrix} \mathbf{d}_0 \\ \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_{m-1} \end{bmatrix}_k = \begin{bmatrix} \mathbf{x}_r + \mathbf{R}(\mathbf{x}_o)\mathbf{b}_{\mathbf{J}(0)} \\ \mathbf{x}_r + \mathbf{R}(\mathbf{x}_o)\mathbf{b}_{\mathbf{J}(1)} \\ \vdots \\ \mathbf{x}_r + \mathbf{R}(\mathbf{x}_o)\mathbf{b}_{\mathbf{J}(m-1)} \end{bmatrix}_k + \xi$$
$$\triangleq \mathbf{H}(\mathbf{X}_k, \mathbf{J}) + \xi \tag{5}$$

where ξ is white Gaussian measurement noise associated with the stereo measurement process[2], and $\mathbf{R} \in SE(3)$ is the rotation matrix formed from the object's Euler angle estimates, $\mathbf{x}_o$. For increased computational efficiency, the burdensome SIFT calculations are limited to a region of interest (ROI) surrounding the object once the object track has been initialized (object initialization is based on the minimum number of observed SIFT matches and the covariance of object pose). Furthermore, to handle the problem of potential mismatches in $\mathbf{J}$, if the object is initialized a geometric feasibility check is applied to each found feature correspondence—*i.e.* each matched feature is compared to an expected location in the object reference frame using the initialized pose and features that fall outside of a distance threshold are rejected.

It is important to note that the physical space associated with a cell may not be entirely visible in a single camera view. To address the issue of *where to look* in the cell, we take an approach similar to [1] and define a search sequence of 9 predefined pan-tilt configurations that cover a viewing hemisphere. If an object is detected, the PTU is adjusted to localize the detected object in the image center of the image frame. If the object is not detected in any section, the cell probability is updated using (1), and the next peak location is determined. Because the local search is generally effective at close ranges, we use a constant α and β parameter which has the effect of suppressing the searched cell when the object is not detected. Implicit in this approach is the underlying assumption that if the object lies in a given grid cell, it can be detected within the viewing hemisphere. This assumption requires the navigation system to deliver the robot to a position facing the center of the grid cell and in close enough range for the local search function to be effective.

[2]We omit the Kalman Filter prediction-update equations for brevity. See Maybeck [12] for a detailed derivation of the EKF.

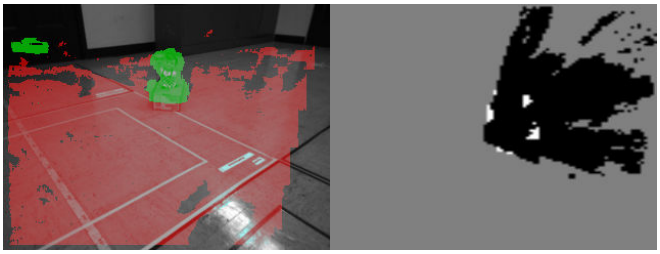

Fig. 3. Robot navigation uses a costmap based on stereo imagery. *Left:* an image taken from an experiment, with space categorized as *free* (red) or *obstacle* (green). *Right:* the navigation costmap generated from the stereo images. White cells correspond to *obstacle*, black cells correspond to *free* and gray cells are *unknown*.

D. Robot Navigation and the Costmap

Navigation is an important part of object search. Previous work [8] [9] relied upon a graph of predetermined free-space nodes to determine navigable paths. This approach limits the number of allowable robot positions and is not guaranteed to cover the search space.

To navigate, the robot maintains a grid-based costmap, M_c (different from M_p), which is searched via the A^* algorithm to find the best feasible path to a selected search location. Each navigation map cell is classified as *free*, *unknown*, or *obstacle* by comparing points in the cell with the estimated ground plane. Points that lie within a threshold distance of the ground plane are labeled as *free*, while others are labeled as *obstacle*. All other cells are considered *unknown* until labeled as either *free* or *obstacle* (see Fig. 3). We take a conservative approach and give priority to *obstacle* if two types of points are projected into the same cell. Furthermore, when applying the A^* search algorithm, obstacles are grown on the map to account for the size of the robot.

To limit computational complexity of the obstacle detection process, the image is downsampled to 320×240 pixels while the robot navigates between search locations. To improve ground plane and obstacle detection, the stereo camera is tilted downwards.

IV. IMPLEMENTATION

It is important to note that the Bayesian recursion update, as mentioned in Section III-A, allows for various initial distributions of P_0. If there is prior information that an object is likely to exist in one region (or room) as opposed to another, P_0 can be weighted to have more mass in a concentrated area (similar to the various initial cases considered by [5]). The Bayesian update of (1) appropriately adjusts the probability map to balance incoming measurements against prior information, such that at each planning action, the peak

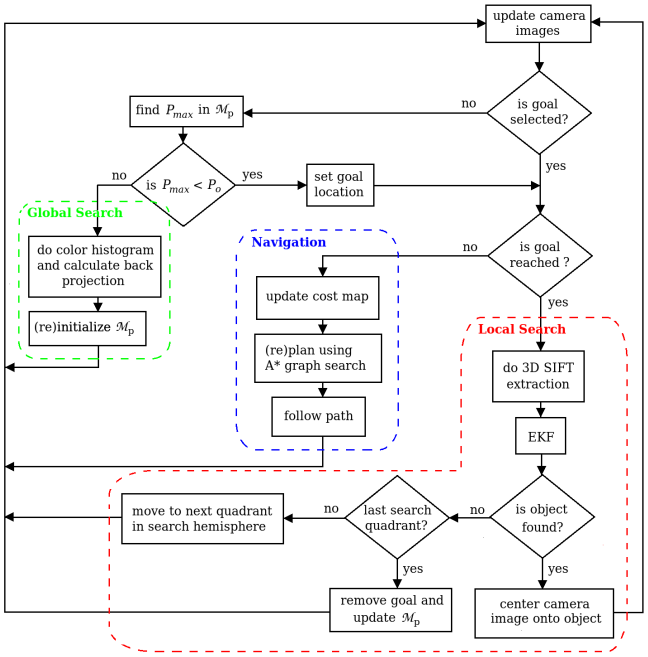

Fig. 4. Block diagram of the process data flow. The dashed boundaries surround the main components of our approach.

probability location can always be selected to yield the most probable object location.

While this framework naturally extends to search in multiple rooms, our particular implementation considers a uniform P_0 in a large room. Fig. 4 summarizes the process data flow of our specific implementation. The dashed boundaries isolate the main algorithm components discussed in Section III. To balance when to apply a global versus a local search, we invoke the global search if the maximum probability in M_p is less than a threshold, P_o. Such an event can occur: 1) at startup, when M_p is set to uniform P_0, and 2) when all cells have been visited and the peak probability of any one cell is not significantly large. This latter case occurs during a missed detection of the object over the entire global search space – thus, invoking an entirely new search process. For the majority of the search process, the local search procedure is the active mode. If a putative object location (a local peak in the prior map probability) is found to be empty, the probability value in that cell is reduced (via (1)) and a new goal is selected and planned to via the navigation function.

V. EXPERIMENTAL RESULTS

Our approach has been implemented on an Evolution Robotics^{TM} ER-1 mobile robot equipped with a Point Grey Research^{TM} BumbleBee2 stereo color camera downsampled to a resolution of 320x240. The stereo camera is mounted on a Directed-Perception^{TM} PTU-D46-17 pan-tilt unit. In the experiments summarized below, robot pose is estimated via wheel odometry. All computations were performed on a laptop computer (Intel Pentium(R) M 1.86GHz processor) running Linux. The algorithm was written in C/C++ using the Intel OpenCV library.

For each object in our database, a series of trials were designed to test: the ability of the global search procedure to generate a valid prior probability map, M_p; the ability of the local search method to identify and estimate object pose at various object heights; and lastly the ability of the robot to navigate on the generated costmap for proper obstacle avoidance. As such, trials were conducted by placing each object at three different heights and at random locations in the workspace. Additionally, the same set of trials were repeated, this time with obstacles placed in the direct planned path of the robot from the previous runs.

The top row of Fig. 5 shows the various objects used to generate our database. The order of the shown objects indicate the increasing order of complexity (left to right) as measured in terms of successful runs with the presented approach. For brevity, representative results from a select number of trials highlighting the strengths and weaknesses of the described method are presented.

The bottom three rows of Fig. 5 show the results of trials from three objects spanning the range of complexity from the existing objects in our database. The far left column presents a monocular image from one of the search sections, with the object manually highlighted in red for convenience[3]. The second column of images shows a relevant portion of the probability map calculated during the initial global search (cell resolution is 20×20cm). Multiple probable object locations are typically hypothesized from the coarse search, requiring subsequent local search attempts. Superimposed on the probability map is: the global coordinate frame origin (green), the initial planned path (yellow), and the goal location (red). Since path planning is executed repeatedly during navigation mode (see Fig. 4), the initial planned path is subject to change should an obstacle appear later in that path. The third column displays the costmap at the end of the trial (cell resolution of 10×10cm). Superimposed on that map (magenta) is the history of robot location as estimated by wheel odometry. In each of the trials, the robot ends at the true object location. The fourth column shows the results of the local search recognition and 6D pose estimation algorithm.

Note also the initial planned path in the bottom row trial ended in a region not containing one of the objects. Nonetheless, when the robot does not find the object during its local search attempt, it replans to the next probable location (towards the upper right) which does contain the object, which is then accurately detected.

From the series of experiments carried out, it became clear the common features among the successful objects that allowed the algorithm to succeed: objects of fairly large size with uniform color enable the color histogram approach of the global search method to quickly identify the most probable location as the correct one in the probability map;

[3]Note the shown image does not necessarily correspond to a forward facing camera. It is a snapshot taken during the initial scan of the environment of the global search step. Fig. 6 shows a panoramic view of the laboratory and serves as a good reference for the reader on the relative location of the object to the robot.

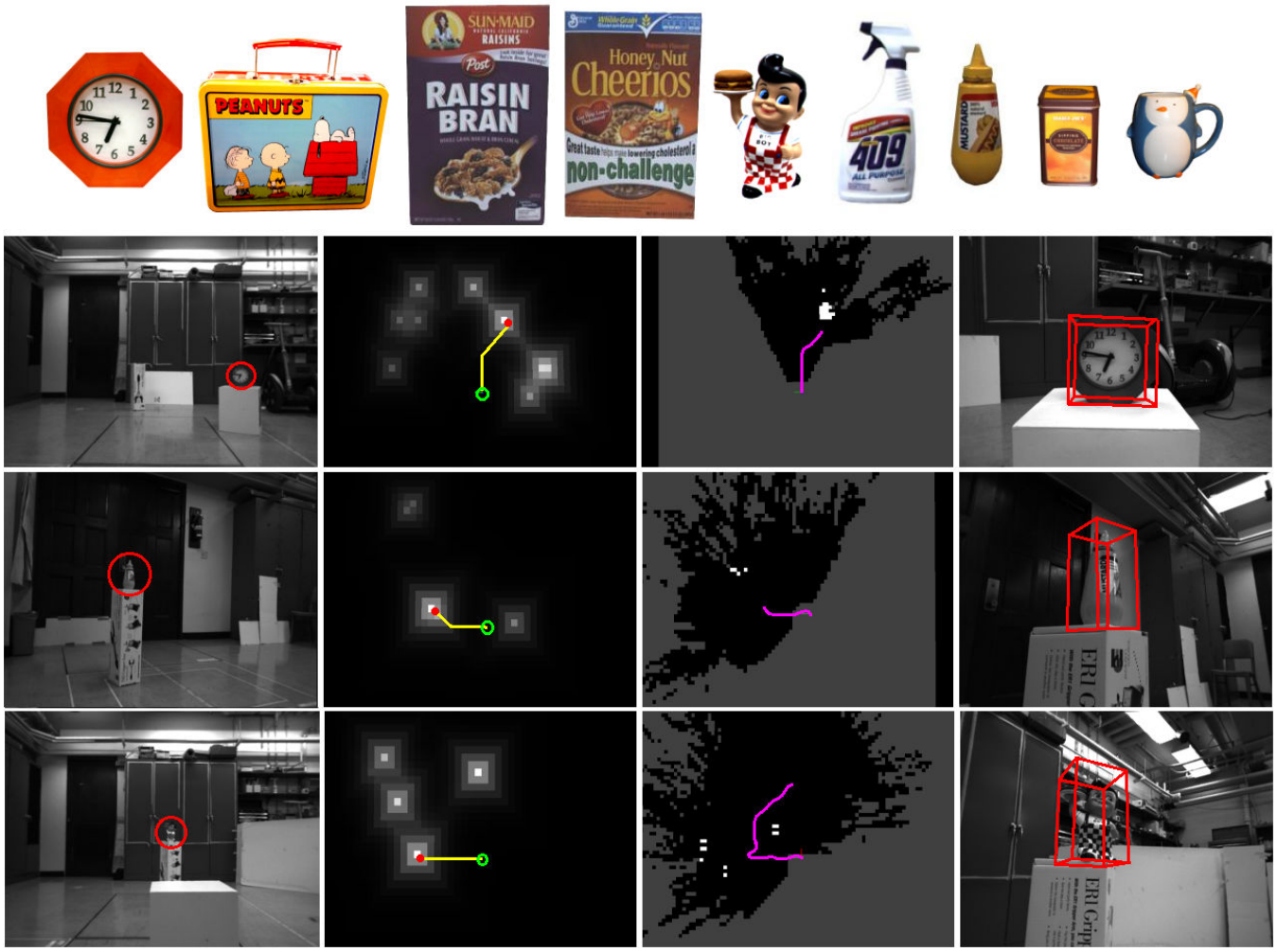

Fig. 5. The top row shows the set of objects used to populate our database of known objects (arranged in order of increasing difficulty from left to right). The remaining three rows each corresponds to a different trial. The first column shows an image snapshot taken during the global scan with the true object location manually highlighted in red. The second column shows the probability map resulting from global search. The initially planned robot path is superimposed (yellow). The third column shows the costmap at the end of the trial, with the robot pose history (magenta). The final column shows the 6D pose estimation of the found object.

objects with a good amount of texture allow for the refined local search method to accurately estimate the pose of the object.

The most difficult objects tested in our database was the penguin cup—a highly glossy and featureless object of fairly small size (top right object in Fig. 5). Not surprisingly, this particular object failed all test trials. What was interesting to note in the experiments for this object was that since the local search always failed to find the object in any cells, it allowed the recursive Bayesian update of the probability map (as defined in (1)) to run its course.

Fig. 6 shows the results of a particular failed trial with the penguin cup. The top image of the figure shows a panoramic view of the environment stitched together from the initial scan of images taken by the robot (the horizontal green line indicates the horizon and the vertical green line indicates the viewing angle associated with a forward facing robot). The bottom row of images show sequential snapshots of the probability map (left to right) taken each time after a local search had been applied and failed to find the object. The width of each grid-based map shown in the bottom row matches the combined field of view of the panoramic image above.

Note that at the beginning of the experiment, three probable locations stand out in the initial probability map (bottom row, left). The objects in the scene corresponding to the cell locations have been highlighted in the panoramic image, with the true object circled in red and distractor objects circled in orange.

Initially, the robot plans a path to investigate a cell location to the right, corresponding to the actual location of the penguin cup. However, once the object is not found in that cell (due to a lack of features), the probability is reduced and the next probable location is planned to (bottom row, second image from left), which in this case corresponds to the fire extinguisher on the wall. Upon failing to find the object there, the probability is again reduced and the next probable location visited (bottom row, third image from

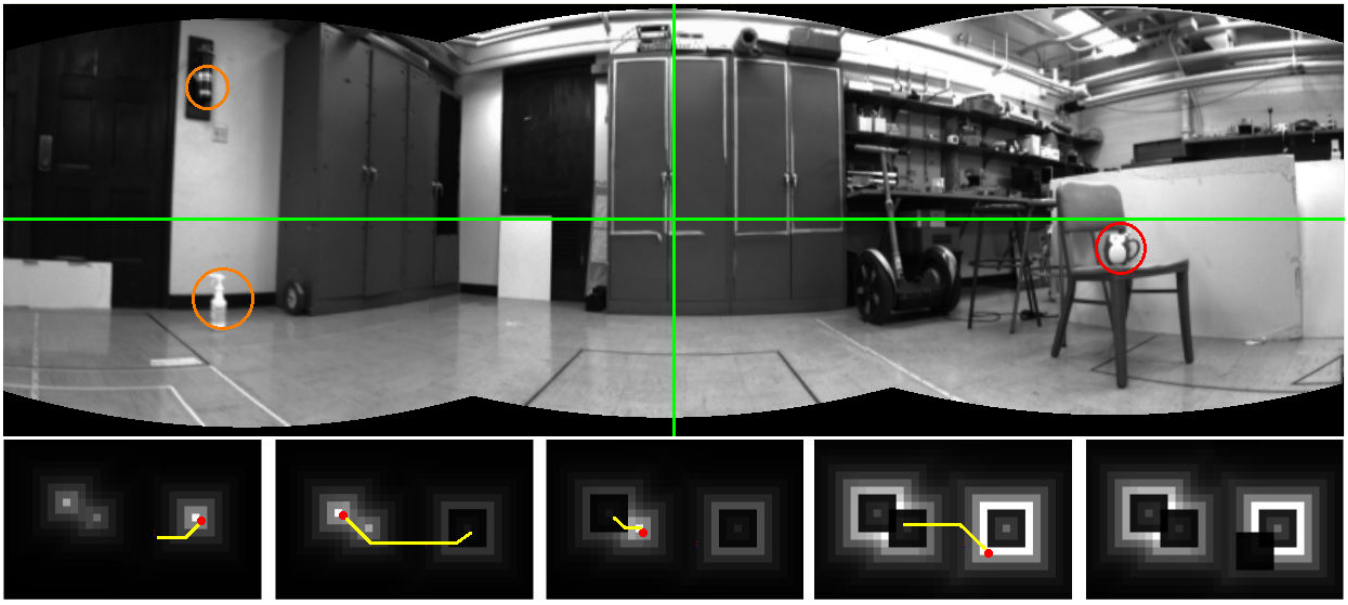

Fig. 6. Experimental results from the object search failed attempt for the penguin cup. Shown in the top view is the panoramic view of the scene and the bottom row shows the progress of the grid-based probability map as the robot investigates more probable cells.

left), which now corresponds to the spray bottle on the floor. Note that because of the reduced probability of the cells associated with the fire extinguisher, probability mass has now been increased in the cells associated with the spray bottle and also the surrounding cells of the first visited cell. Once the spray bottle is found not to be the penguin cup, the robot reduces the probability of the cell and now revisits a neighboring cell of the first visited cell (bottom row, second from right). When the object is not found there, the robot again reduces the probability of that cell and continues to the next probable location (bottom row, right image). At this point, the experiment was terminated by the operator, but the resulting data illustrates the ability of the search algorithm to use the Bayesian update to identify probable search locations.

VI. CONCLUSIONS AND FUTURE WORK

We presented an improved stereo vision search procedure that uses Bayesian updating of a grid-based probability map to drive the search process. By using a global and local search method, object search with path planning, obstacle avoidance, and 6D pose estimation has been demonstrated on three different model objects using only a single stereo sensor and modest computation. In future work, we aim to consider multiple objects and remove the constraint on object stationarity. Furthermore, we hope to improve the model histogram by considering a different color space (HSV) with harsher environments and varying lighting conditions.

REFERENCES

[1] Y. Ye and J. K. Tsotsos, "Sensor planning for 3d object search," *Comp. Vision and Image Understand.*, vol. 73, pp. 145–168, 1999.

[2] F. Saidi, O. Stasse, and K. Yokoi, "A visual attention framework for search behavior by a humanoid robot," in *Proc. IEEE RAS/RSJ Conf. on Humanoid Robots*, 2006, pp. 346–351.

[3] ——, "Active visual search by a humanoid robot," in *Proc. IEEE/RAS Int. Conf. on Advanced Robotics*, 2007, pp. 360–365.

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

[5] T. Chung and J. Burdick, "A decision making framework for control strategies in probabilistic search," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 4386–4393.

[6] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[7] L. Petersson, P. Jensfelt, D. Tell, M. Strandberg, D. Kragic, and H. Christensen, "Systems integration for real-world manipulation tasks," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 2500–2505.

[8] S. Ekvall, D. Kragic, and P. Jensfelt, "Object detection and mapping for service robot tasks," *Robotica*, vol. 25, no. 2, pp. 175–187, 2007.

[9] D. Lopez, K. Sjo, C. Paul, and P. Jensfelt, "Hybrid laser and vision based object search and localization," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2008, pp. 3881–3887.

[10] S. Ekvall and D. Kragic, "Receptive field cooccurrence histograms for object detection," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2005, pp. 84–89.

[11] J. S. Beis and D. G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," in *Proc. IEEE Conf. Comp. Vision and Pattern Recognition*, 1997, pp. 1000–1006.

[12] P. S. Maybeck, *Stochastic models, estimation, and control*. Academic Press, 1979, vol. 141.