Active 3D Object Localization Using a Humanoid Robot

Alexander Andreopoulos, Stephan Hasler, Heiko Wersing, Herbert Janßen, John Tsotsos, Edgar Körner

2011

Active 3D Object Localization Using A Humanoid Robot

Alexander Andreopoulos, Stephan Hasler, Heiko Wersing, Herbert Janssen, John K. Tsotsos, and Edgar Körner

***Abstract*—We study the problem of actively searching for an object in a 3D environment under the constraint of a maximum search time, using a visually guided humanoid robot with twenty-six degrees of freedom. The inherent intractability of the problem is discussed and a greedy strategy for selecting the best next viewpoint is employed. We describe a target probability updating scheme approximating the optimal solution to the problem, providing an efficient solution to the selection of the best next viewpoint. We employ a hierarchical recognition architecture, inspired by human vision, that uses contextual cues for attending to the view-tuned units at the proper intrinsic scales and for active control of the robotic platform sensor's coordinate frame, also giving us control of the extrinsic image scale and achieving the proper sequence of pathognomonic views of the scene. The recognition model makes no particular assumptions on shape properties like texture and is trained by showing the object by hand to the robot. Our results demonstrate the feasibility of using state of the art vision-based systems for efficient and reliable object localization in an indoor 3D environment.**

***Index Terms*—Computer Vision, Active Vision, Visual Search, Recognition, Honda's Humanoid Robot**

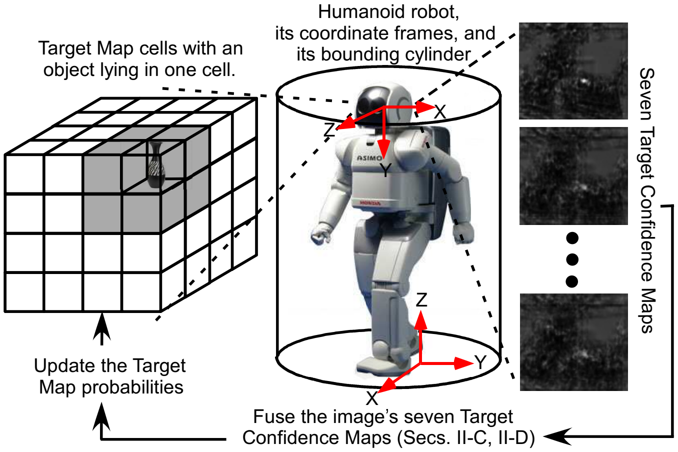

Humanoid robot,
its coordinate frames, and
its bounding cylinder
Target Map cells with an
object lying in one cell.
Seven Target Confidence Maps
X
Z
Y
ASIMO
Z
Y
X
Update the Target
Map probabilities
Fuse the image's seven Target
Confidence Maps (Secs. II-C, II-D)

Fig. 1. Acquiring an image and using the Target Confidence Maps to update a $4 \times 4 \times 4$ cell Target Map. Grey cells in the Target Map denote the Marked Candidate Cells (Sec.II-C) that are induced by the obstacle (a vase).

I. Introduction

VISION is the process of discovering from images what is present in the world and where it is [1]. Within the context of this paper, we distinguish four levels of tasks in the vision problem, which we label as follows [2]:

- *Detection*: is a particular item present in the stimulus?
- *Localization*: detection plus accurate location of item.
- *Recognition*: localization of the items present in the stimulus plus their accurate description through their association with linguistic labels.
- *Understanding*: recognition plus role of stimulus in the context of the scene.

It is generally accepted that passive approaches to the vision problem have a number of shortcomings. As a means of addressing these problems, Bajcsy introduced in 1985 the concept of *active perception* or *active vision* as "a problem of intelligent control strategies applied to the data acquisition process" [3]. Active control of a vision-based sensor offers a number of benefits [4], [5]. It allows us to: (i) Bring into the sensor's field of view regions that are hidden due to occlusion and self-occlusion. (ii) Foveate and compensate for spatial non-uniformity of the sensor. (iii) Increase spatial resolution through sensor zoom and observer motion that brings the region of interest in the depth of field of the camera. (iv) Disambiguate degenerate views due to finite camera resolution, lighting changes and induced motion [6]. (v) Deal with incomplete information and complete a task.

An active vision system's benefits must outweigh the associated execution costs [4]. Dealing with the associated costs of an active vision system is a fundamental problem in robot vision and the human visual system (HVS) [7]. In the HVS this emerges as the attention problem [8], a phenomenon subsuming the active vision problem that has recently started to emerge as an important issue in computer and robot vision. The associated costs in an active vision system include:(i) Deciding the actions to perform and their execution order. (ii) The time to execute the commands and bring the actuators to their desired state. (iii) Adapt the system to the new viewpoint, find the correspondences between the old and new viewpoint and deal with sensor noise ambiguities [4].

In [9], [10] Ye, Andreopoulos and Tsotsos discuss the problem of sensor planning for object search. They prove the intractability of finding a finite sequence of sensor states that maximizes the probability of localizing an object in a 3D search region under a search cost constraint. Given a sequence of candidate states, the expected probability of localizing the object with each state, and the cost of each state, the authors select as the next state the one that maximizes the ratio of the

Alexander Andreopoulos, Stephan Hasler, Heiko Wersing, Herbert Janssen and Edgar Körner are affiliated with the HONDA Research Institute Europe GmbH, Carl-Legien-Str.30, 63073 Offenbach/Main, Germany

Alexander Andreopoulos and John K. Tsotsos are affiliated with York University, Dept. of Computer Science & Engineering and the Centre for Vision Research, Toronto, Ontario, M3J 1P3, Canada

Alexander Andreopoulos is also affiliated with the CoR-Lab, Research Institute for Cognition and Robotics, Bielefeld University, 33615 Bielefeld, Germany. Emails: alekos@cse.yorku.ca, {stephan.hasler, heiko.wersing, herbert.janssen}@honda-ri.de, tsotsos@cse.yorku.ca, edgar.koerner@honda-ri.de.

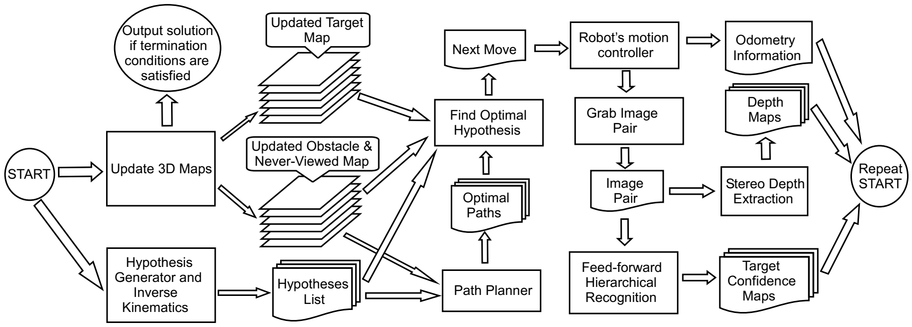


Fig. 2. A component-wise [11] break-down of the active 3D object localization architecture, outlining each executed loop iteration and defining the execution order of each component in our architecture. In Secs.II-A,II-C,II-D of the text, we describe the motivation and the implementation of the 'Update Maps' component and its outputs. In Sec.II-B of the text, we describe the components 'Hypothesis Generator and Inverse Kinematics', 'Path Planner', and 'Find Optimal Hypothesis', as well as their outputs. In Sec.II-E of the text, we describe the 'Feed-forward Hierarchical Recognition' component.

benefit to the cost, in a one-step look ahead approach.

Some of the earliest work on active object localization, includes Garvey's [12] work on searching for intermediate objects that participate in spatial relationships with the target object, in order to speed up the localization. Similarly, Wixson and Ballard [13] present an active object localization algorithm that uses intermediate objects to maximize the system's efficiency and accuracy. Such intermediate objects are usually easy to recognize at low resolutions and they are, thus, located quickly. Maver and Bajcsy [14] propose a next-view-planning algorithm to deal with occlusions and search for a target in hidden regions. Rimey and Brown's [15] TEA-1 vision system can search within a static image for a particular object and can also actively control a camera if the object is not within its field of view. Giefing *et al.* [16] propose an active vision system, that incorporates camera gaze shifts for exploring scenes. Ekvall *et al.* [17] integrate a SLAM approach with an object recognition algorithm based on receptive-field co-occurrence histograms. Other algorithms combine image saliency mechanisms with bag-of-features approaches [18], [19]. Saidi *et al.* [20] present an implementation, on a humanoid robot, of an active object localization system that uses SIFT features [21] and is based on the next-view-planner described in [9].

A number of papers have dealt with the similar problems of multi-view detection and recognition. Some of the earliest work on view planning for object recognition includes the work by Wilkes and Tsotsos [22]. The authors suggest using various behaviours for detecting objects in the presence of ambiguities such as view degeneracies [6], occlusion and limited depth information. Callari *et al.* [23], [24] define contextual knowledge as the join of a discrete set of prior hypotheses about the relative likelihood of various model parameters, given a set of object views with the likelihood of each object hypothesis as the agent explores the scene. Laporte and Arbel [25] also present a Bayesian approach to the viewpoint selection problem. Dickinson *et al.* [26] combine a Bayesian based attention mechanism, with aspect graph based object recognition and viewpoint control. Schiele and Crowley [27] use a measure called *transinformation* for building a robust recognition system. Similarly Borotschnig *et al.* [28] use an information theoretic based quantity (entropy) to decide the next view of an object that the camera should take to obtain more robust recognition in the presence of ambiguous viewpoints. Foissotte *et al.* [29] propose a next-view-planner for 3D object modelling and comment on its potential applications in multi-view recognition. Roy *et al.* [30], [31] present an active object recognition algorithm for objects that might not fit in the camera's field of view. A number of techniques for solving problems within the mobile robotics field, involve choosing a sequence of actions that reduce the amount of uncertainty under noise-free observations and noisy observations of the environment (*e.g*, MDPs and POMDPs [32], [33]). The use of POMDPs for the scene exploration and SLAM problems has gained popularity amongst the robotics community. POMDPs have been applied successfully on problems that use non-vision based sensors, and a significant research effort is currently under-way on related problems utilizing MDPs/POMDPs with mixtures of vision and non-vision based sensors [32], [34]. In the next section we describe our active object localization algorithm.

II. A Humanoid Robot that Searches

We address the problem of actively searching for an object in a 3D environment using a research version of Honda's humanoid robot (HR) (see Fig.1 and [35]), a visually guided humanoid robot with twenty-six degrees of freedom (DOF). We describe an object probability updating scheme providing a solution to the best next viewpoint selection problem. We employ a hierarchical recognition architecture inspired by human vision [36] that uses contextual scene structure cues for attending to the architecture's view-tuned units at the proper intrinsic scales and for active control of the robotic platform's position, also giving us control of the extrinsic image scale and achieving the proper sequence of pathognomonic views of the scene. Cues used include hue, stereo depth information, expected viewpoint dependent occlusions, object scale and target uniqueness within the scene context — uniqueness within each acquired image and across all acquired images.

In Fig.2 we show the system's organizational structure. Our system maintains a *target map* (Sec.II-A), encoding the probability that each position in the search space contains the centre of the object we are searching for. Our system also maintains an *obstacle map* (Sec.II-A), which encodes the structure of the explored scene. The robot we use [35] (referred to in this paper as Honda's humanoid robot, or, HR) executes a finite sequence of greedily selected movements, positioning itself to the next-best viewpoint that maximizes the probability of localizing the target object position (based on the target map probabilities), taking into consideration potential occlusions from each viewpoint (using the obstacle map information), while also minimizing the cost of moving to the new viewpoint (see Sec.II-B). As we briefly discuss in Sec.II-A, this approach helps us deal with the intractability of the object localization problem under a cost constraint. The outputs of a feedforward hierarchical recognition architecture (Sec.II-E) are transformed into single-view generative probabilities (see Sec.II-C and Sec.II-D) that encode desirable criteria of target uniqueness within each individual image, but also across multiple images. These probabilities are incorporated in a Bayesian framework that is used to update the target map probabilities. A strategy for minimizing the effects of dead-reckoning errors on the system's reliability is also described in Sec.II-B.

A. Basic Definitions

We define the active object localization problem as the problem of finding a finite sequence of viewpoints that maximize the probability of localizing the target object, subject to a cost constraint [9]. This section formalizes the problem.

Assumption 1. *We assume that exactly one instance of the target object exists in the scene.*

Our system depends on three coordinate frames, the **Heel Coordinate Frame**, the **World Coordinate Frame**, and the **Eye Coordinate Frame**. The origin of the heel coordinate frame is defined as the projection on the floor plane of the point centred between HR's two heels. Its Z-axis is parallel to the floor's normal and points upwards. Its X-axis points in HR's forward direction (see Fig.1). The world coordinate frame is the inertial frame and corresponds to the initial heel coordinate frame. Finally, the eye coordinate frame is the left camera's coordinate frame (see Fig.1).

The **search space** consists of a 3D region $[X_l, X_h] \times [Y_l, Y_h] \times [Z_l, Z_h]$ whose coordinates are expressed with respect to the world coordinate frame. The **target map** is a discretization of the 3D search space into non-overlapping 3D cells. Each cell is assigned the probability that it is the cell in the scene containing the target object's centroid (see Fig.2). The **obstacle map** is a discretization of the 3D search space into binary valued 3D cells. Each cell indicates whether it contains solid structure (see Fig.2). Finally, a **never-viewed map** discretizes the 3D search space into binary valued cells, denoting the cells that have been sensed at least once. The updating of these maps is discussed in Sec.II-C. In this paper, the discretization of the target map, never-viewed map and obstacle map is the same, and consists of cells with equal volumes ($5cm \times 5cm \times 5cm$), whose centres are uniformly sampled at $5cm$ intervals along each axis (see Fig.1). We use a set of positive integers, $C = \{1, 2, ..., |C|\}$, to index each cell in the target map, obstacle map and never-viewed map, where $|C|$ denotes the cardinality of set C. All cells of the obstacle map are initialized as containing no obstacle. A cell of the never-viewed map is initialized as 'not-viewed' if and only if the corresponding target map cell has a non-zero prior probability. Since we assume that a single target object exists in the scene, the target map cells sum to one.

Definition 1. (Scene Sample Function) *A scene sample function $\mu_{v_n}(\vec{x})$, denotes the sensor output that was acquired under a parameter v_n representing the sensor state at step $n \in \mathbb{N}$ (*e.g., *v_n could represent the extrinsic camera parameters, field of view etc.), where $\vec{x}$ is an index into the scene sample function μ_{v_n}. We use λ_n to denote the sensor output acquired at step n, without specifying v_n. For example, in the case of greyscale images, $\vec{x} = (i, j)$ can denote a pixel index and $\mu_{v_n}(\vec{x}) = \lambda_n(\vec{x})$ is the intensity of pixel $\vec{x}$, assuming the camera's parameters were set to v_n when the image λ_n was acquired. Thus, event $\{\mu_{v_n}\}$ is equivalent to the occurrence of two events: the event where the sensor state is set to v_n and the event where the sensor output is function λ_n. Given some μ_{v_n}, we refer to λ_n as the* **image** *of μ_{v_n}.*

Notice that if we condition on v_n, then the conditioned event $\{\mu_{v_n}\}|\{v_n\}$ is equivalent to event $\{\lambda_n\}|\{v_n\}$. In this paper, a sensor state v_n specifies the eye coordinate frame and the heel coordinate frame with respect to the world coordinate frame, while μ_{v_n} denotes the sensor state v_n and the image λ_n that is acquired by HR's left camera (the eye coordinate frame), under state v_n. We define a probability space $\Upsilon = (X_1, \Sigma_1, p_1)$ [37] for any sensor state $v \in X_1$, where X_1 is a set of sensor parameter states, Σ_1 is a σ-algebra of X_1, and p_1 is a probability measure on X_1 whose support includes all states that are achievable by our sensor in the current scene. Similarly, for each v, we define a probability space $\Upsilon(v) = (X_v, \Sigma_v, p_v)$ with $p_v(\lambda)$ denoting the conditional probability of occurrence of an image $\lambda \in X_v$, if the image were acquired under sensor state v. The underlying probability measure, models the sensed scene uncertainty (image noise, varying illumination conditions, dead reckoning errors, etc.) and is largely unknown and difficult to model in practice.

Given a sequence $v_1, ..., v_n$ of sensor states, the total **sequence cost** $T(n)$ associated with executing the sequence is given by $T(n) \triangleq T(n-1) + \mathbf{t_o}(v_{n-1}, v_n)$ where $\mathbf{t_o}(v_{n-1}, v_n)$ denotes the sum of the costs of planning the next state v_n from state v_{n-1} and of reaching sensor state v_n from sensor state v_{n-1}. $T(1)$ is the cost of reaching state v_1 from the initial robot state. In this paper, the cost $\mathbf{t_o}(v_{n-1}, v_n)$ is proportional to the sum of the time the robot takes to plan the next move v_n and of the time the robot state takes to reach state v_n from initial state v_{n-1} (*e.g.*, the time to execute one iteration of the loop in Fig.2). We define one variant of the *constrained active object localization* (CAOL) problem as follows:

Definition 2. (Constrained Active Object Localization: Variant 1) *Find a sequence $v_1, ..., v_n$ of sensor states and the cells $i \in C$ satisfying $p(c_i^t|\lambda_n, v_n, ..., \lambda_1, v_1) \geq \theta$ and $T(n) \leq T'$ for some $\lambda_1, ..., \lambda_n$, where T' is a search cost bound, θ is a probability threshold, and c_i^t denotes the event that the centroid of target t is in cell i.*

By Def.1, $p(c_i^t|\lambda_n, v_n, ..., \lambda_1, v_1) = p(c_i^t|\mu_{v_n}, ..., \mu_{v_1})$. Solutions to the above problem can compensate for our limited knowledge on $\Upsilon(v)$ $\forall v$, and satisfy the need to minimize actuator and sensor movements, by searching for a finite sequence $v_n, ..., v_1$ that minimizes the total search time and that best samples the unknown probability spaces. In [10], it is shown that a number of variants of the problem are NP-Hard. The rest of Sec.II presents an efficient algorithm that approximates the optimal solution to the CAOL problem.

B. Hypotheses Generation and Evaluation

We now describe our next-view-planning algorithm that allows us to select the next sensor state v_n, given that we have executed actions $v_1, ..., v_{n-1}$. In Sec.II-C and Sec.II-D we discuss how to update the target, obstacle and never-viewed maps for each new sensor state v_n HR finds itself in (each loop iteration in Fig.2). We use a hypothesize-and-test approach to the next-view-planning problem that parallels the greedy and near optimal strategy for solving the Knapsack problem [9], [38]. The approach is a one-step look-ahead algorithm which also parallels the optimality of the ideal searcher [39]. For each sensor state v corresponding to one of the candidate hypotheses, we assign a score that is based on: (i) The likelihood of detecting the object from the sensor state v, given the expected occlusions and expected intrinsic scale of the projected object if it were centred in each of the target cells viewed by setting v. (ii) The expected cost of reaching state v from the current state. We proceed by defining the candidate hypotheses/sensor states over which we optimize the next-view-planner, when selecting the next sensor state.

Intuitively, our set of candidate hypotheses consists of the cross product of (i) a set of poses for the heel coordinate frame with (ii) a set of poses for the eye coordinate frame expressed with respect to the heel coordinate frame. This cross product corresponds to the set of poses from which HR can explore the scene. In more detail, the **movement list** (ML) is a finite set of coordinates that lie in $[X_l, X_h] \times [Y_l, Y_h] \times [0, 2\pi)$. The movement list corresponds to all the possible positions and orientations that we wish HR to consider for its heel coordinate frame at each iteration of the algorithm's loop (Fig.2). In our online implementation of the algorithm, the movement list is generated by uniformly sampling each dimension of $[X_l, X_h] \times [Y_l, Y_h] \times [0, 2\pi)$. The **gaze list** (GL) consists of a finite set of 3D coordinates expressed with respect to the heel coordinate frame. All the points in the gaze list must be capable of being projected on the image centre, using an HR whole-body-motion command which does not involve changing the heel coordinate frame — *e.g.*, it changes head pant/tilt, body and leg posture, but not the feet position. The **candidate hypotheses list** (CL) consists of the cross products of the movement list with the gaze list — *i.e.*, $CL = ML \times GL$. For each position of HR's heel in ML, the gaze list GL corresponds to a set of regions around the robot that can be explored ("looked at") without changing the heel position. Each $w \in CL$ is mapped to a sensor state $\texttt{map}(w)$ that has the same heel coordinate frame as w, and has an eye coordinate frame such that the Gaze List point of w projects on the frame's image centre (*i.e.*, if multiple sensor states satisfy w, then $\texttt{map}(w)$ selects one such state deterministically).

We need to define a measure of "clearance" between HR and scene obstacles, that will allow us to detect potential collisions during the path planning phase. To this extent, the **HR bounding cylinder** is defined as the 3D region encompassed by the smallest volume cylinder whose medial axis intersects the heel coordinate frame's origin, is parallel to that frame's Z-axis and completely encompasses HR (see Fig.1). We use a path planner based on Dijkstra's algorithm to determine whether there exists a path from HR's heel coordinate frame origin corresponding to the current sensor state v_{n-1} to a candidate sensor state v, and to find the shortest path to follow in moving from v_{n-1} to v. Let

$$ML' = proj_{[X_l, X_h] \times [Y_l, Y_h]}(ML) \quad (1)$$

denote the projection of the movement list on its first two dimensions. Then, the nodes of the graph used by the path planner correspond to ML'. Each pair of nodes $n_1, n_2 \in ML'$, $n_1 \neq n_2$, are connected by an edge if: (i) the two nodes fall in neighbouring cells on the Voronoi diagram of ML' and (ii) the total number of cells marked as obstacles plus the total number of cells marked as never-viewed in the corresponding maps, that also lie in HR's bounding cylinder as it traverses from n_1 to n_2, do not exceed a threshold θ'. The edge weight is the distance between the two nodes. In Sec.II-C we will need to update the target map cells which lie inside the target object's volume and are, thus, occluded from all viewpoints. We continue by defining certain data-structures for achieving this goal, which are also used by the next-view-planner.

The **target bounding cylinder** at 3D position x, consists of the 3D region encompassed by the smallest volume cylinder that would completely engulf the target object, should the target object be centred at x and be positioned 'upright', on its pre-designated base side. The cylinder's medial axis is set parallel to the world coordinate frame's z-axis.

Assume we have executed actions $v_1, ..., v_n$. We say that cell i is a **visible cell** under state v, if cell i lies in the sensor's estimated field of view under state v, cell i is not occluded from viewpoint v by any obstacle in the obstacle map built using the depth maps of $\mu_{v_1}, ..., \mu_{v_n}$, and the intrinsic scale of the projection on the image plane of target object t if it were centred in cell i, lies in the permissible range of intrinsic scales of our feedforward hierarchy (Sec.II-E). A set $V(v; v_n, ..., v_1)$ of cell indices denotes the visible cells. We estimate the best matching intrinsic scale of a target centred in cell i, by the size on the image plane of the projection of the target bounding cylinder centred in cell i. In contrast to [5], the visibility range for state v depends on the recognition scale, and not on the camera's depth of field.

Let γ_i denote a neighbourhood of constant radius centred at cell i. Let $\epsilon_n(\gamma_i, v) \in \{0, 1\}$ take a value of 1 iff there exists a cell j in neighbourhood γ_i such that $j \in V(v; v_{n-1}, ..., v_1)$, and there exists an unobstructed path from the current position/sensor state v_{n-1} to a position corresponding to state v, as calculated by the path planner. Recall that the path planner only outputs paths for which, at any point along the path, HR's bounding cylinder does not intersect too many obstacles and never-viewed cells. The intractability results in [9], [10] motivate a solution to the next-view-planning for the CAOL problem, based on the greedy approximation to the Knapsack problem. Thus, the next sensor state v_n is given by

$$v_n = \operatorname*{argmax}_{v \in CL'} \frac{\sum_{i \in C} p(c_i^t | \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) \epsilon_n(\gamma_i, v)}{\mathbf{t_o}(v_{n-1}, v)} \quad (2)$$

where $CL' = \texttt{map}(CL)$, the range of $\texttt{map}(\cdot)$ using CL as its domain. We continue the search until we have reached the maximum search cost T' specified in the CAOL problem.

An important problem is that of determining good termination conditions. One solution is to constrain the thresholding by θ in the CAOL problem, to cells that have been viewed at least once. As it is shown in [10], fusing multiple views (*i.e.*, fusing their obstacle maps, target maps and never-viewed maps with the corresponding maps we have built so far) requires accurate dead-reckoning for the resultant target probability maps, obstacle maps and never-viewed maps to be accurate. Dead-reckoning errors lead to an increased bias in the target localization and destroy any guarantees of a decreasing target map entropy, making the probability thresholding of the CAOL problem sensitive to errors and inappropriate. However, it is an inevitable fact that we need to continuously update the target probability map, obstacle map and never-viewed map as a means of guiding the where-to-look-next functionality of our localization algorithm. We, thus, take the middle road and use the updated maps to guide the where-to-look-next behaviour of the recognition algorithm. We search until the total cost of our search exceeds T', at which point, the algorithm outputs as the target location $\hat{i}_t$ the cell with the maximum single-view generative probability across all n acquired images:

$$\hat{i}_t \triangleq \operatorname*{argmax}_{i \in C'} \max_{j \in \{1,...,n\} \texttt{ such that } i \in M(v_j)} p(\lambda_j | c_i^t, v_j). \quad (3)$$

where $C' = M(v_1) \cup ... \cup M(v_n)$, and for any $j \in \{1, ..., n\}$ the function $M(v_j)$ is a superset of the cells in $V(v_j; v_j)$ that contain an obstacle according to μ_{v_j}'s depth map. Notice that $V(v_j; v_j)$ uses data only from a single view v_j.

The function $M(v_n)$ helps us deal with dead-reckoning errors, and denotes the **marked candidate cells** of iteration n. We say that cell i is a marked candidate cell at iteration n if its centre lies inside a target bounding cylinder that is centred at some cell $j \in V(v_n; v_n)$ which according to the depth map of μ_{v_n} contains an obstacle. As we will see in Sec.II-C, function $M(\cdot)$ is also used to update the target map probabilities of cells which lie inside the object volume and are, thus, occluded from all viewpoints.

Because $M(v_n)$ uses the depth map of only a single view v_n, we avoid many of the previously discussed problems caused by dead-reckoning errors. Fig.1 shows the marked candidate cells induced by an obstacle (the vase in the figure), assuming that the target bounding cylinder of the object we are searching for spans the grey cells and is centred in the cell with the obstacle. The estimation of the generative probabilities in Eq.(3), and their role in updating our maps, is discussed in Secs.II-C to II-E. The advantages of Eq.(3) in object localization are significant, since limiting the dead-reckoning errors using standard vision-based SLAM approaches is non-trivial. We could further refine the detection accuracy, at the expense of the localization accuracy however [10], by performing a new search around the cells closest to a hypothesized target position, to validate that those cells do contain the target.

C. Updating the Target, Obstacle and Never-Viewed Maps

As previously mentioned, our localization algorithm relies on a feedforward hierarchical recognition architecture, that is inspired by human vision [36]. We postpone the discussion on the training and construction of this architecture until Sec. II-E and proceed in this section by discussing the use of the outputs of this feedforward hierarchy (the *target confidence maps*) to update the target, obstacle and never-viewed maps.

Theorem 1. (Bayesian Target Map Updating) *Assume* $p(\lambda_n | c_i^t, v_n, \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) = p(\lambda_n | c_i^t, v_n)$, $p(c_i^t | \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) = p(c_i^t | v_n, \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1)$. *Then,* $p(c_i^t | \lambda_n, v_n, ..., \lambda_1, v_1) =$

$$\frac{p(c_i^t | \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) p(\lambda_n | c_i^t, v_n)}{\sum_j p(c_j^t | \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) p(\lambda_n | c_j^t, v_n)}. \quad (4)$$

Proof: See the Appendix. Since we condition on v_n, then $p(\mu_{v_n} | c_i^t, v_n) = p(\lambda_n | c_i^t, v_n)$ (see Def.1). The theorem's second assumption implies that positioning the sensor without acquiring an image does not provide any information. ■

Eq.(4) links the discriminative problem of calculating $p(c_i^t | \lambda_n, v_n, ..., \lambda_1, v_1)$, to the generative problem of modelling $p(\lambda_n | c_i^t, v_n)$. Thm. 1 assumes that λ_n is conditionally independent of previous sensor readings/states, given the cell i where the target is centred and given state v_n. By Assumption 1, exactly one instance of the target exists in the scene, which implies that events c_i^t, v_n are sufficient to determine which regions of λ_n (if any) may correspond to the projection of the target object on the image plane and which regions must correspond to the background, making the assumptions in Thm.1 realistic simplifications to our problem. Due to the difficulty in modelling an image λ_n with an arbitrary background, we are implicitly assuming that $p(\lambda_n | c_i^t, v_n)$ denotes a generative modelling of the recognition algorithm's resultant binary segmentation into the foreground (target position) and the background, based on a single view. Similarly $p(c_i^t | \lambda_n, v_n, ..., \lambda_1, v_1)$ denotes the corresponding probability of event c_i^t, based on the Bayesian fusion of multiple-views μ_{v_n},...,μ_{v_1}. The greater the uncertainty in space $\Upsilon(v_n)$, the weaker this assumption of conditional independence becomes, due to increased sources of errors (*e.g.*, dead-reckoning errors) in the mapping of an object centred in cell i to μ_{v_n}. The above-described generative probabilities make it possible to update the target map probabilities using Thm.1. Notice that previously described object localization methodologies that apply a binary object detector on each input image (*e.g.*, [5], [9]) are not suited for use with Thm.1, due to their inability to distinguish the foreground from the background in an image.

For each new iteration n, the obstacle map is updated by marking as occupied any cell that is found by our depth extraction algorithm to contain solid structure. The never-viewed map is updated at iteration n by marking as 'viewed' every cell index in $V(v_n; v_n) \cup M(v_n)$ and leaving unchanged all the other cells. Ideally, under good depth estimation and limited occlusions, each cell in the target object's volume, including the target centroid, ends up as a marked candidate cell from at least one viewpoint. For each cell $i \in$

$V(v_n;v_n)\backslash M(v_n)$ that had not been viewed at least once before iteration n and no structure is found in it at iteration n, we set $p(c_i^t|\lambda_n,v_n,...,\lambda_1,v_1)=0$ by the assignment $p(\lambda_n|c_i^t,v_n)\leftarrow 0$, implying that we consider it impossible to sense λ_n from viewpoint v_n if the target is centred in cell i. For each marked cell that had been viewed at least once before iteration n, and was assigned $p(c_i^t|\lambda_{n-1},v_{n-1},...,\lambda_1,v_1)=0$ due to it containing no structure (potentially due to stereo depth extraction errors or dead-reckoning errors), we treat that cell as if iteration n was the first time that cell became visible, by appropriately adjusting the probability value from the target map that is to be updated. We continue by defining $p(\lambda_n|c_i^t,v_n)$ for $i\in M(v_n)$ and for $i\notin V(v_n;v_n)\cup M(v_n)$ (see Eq.(4)), through the use of **target confidence maps**.

Given a scene sample function μ_v that was acquired under sensor state v, and assuming we are searching for object t, a target confidence map is the output of our single-view recognition algorithm on this input image. Such a target confidence map can be thought of as a multiscale topographic map, that assumes values in the range $[0,1]$, with higher values denoting an increased likelihood that the target object t projects with a given scale on the corresponding image region. Their construction is over-viewed in Sec. II-E. Fig. 1 overviews the process of using the confidence maps of an image, to produce the corresponding generative probabilities that are used to update the 3D target map under Thm.1. Fig.3(b) and the supplementary documentation provide further examples of the target confidence maps produced for various images. The value of the target confidence map $\mathbf{CM}(\cdot;\mu_v,v,s,t)$ of μ_v, for target t at intrinsic scale s ($1\leq s\leq N$), sensor state v and at map position $\vec{q}=(i,j)$, is called the **firing rate**, and is given by $\mathbf{CM}(\vec{q};\mu_v,v,s,t)$. We use $\overrightarrow{\mathbf{CM}}(\vec{q};\mu_v,v,t)$ to denote the N-dimensional vector of the target confidence map values at position $\vec{q}=(i,j)$ and across all N scales.

In Sec. II-E we describe how the target confidence maps are built, based on the hierarchical recognition architecture of [36]. The target confidence maps are constructed over seven scales in our implementation ($N=7$ scales). The range of $\mathbf{CM}(\vec{q};\mu_v,v,s,t)$ lies in [0,1], with a higher value implying a greater confidence that target t projects on pixel $\vec{q}$, with a scale s. The resolution of $\mathbf{CM}(\cdot;\mu_v,v,s,t)$ is the same for all scales s, and does not have to be identical to the resolution of μ_v (see Sec.II-E). This is formalized by Thm. 2, which is over-viewed below and in the Appendix. Each function $f(\cdot)=\mathbf{CM}(\ \cdot\ ;\mu_v,v,s,t)$ and $g(\cdot)=\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)$ is a sample from underlying probability measures $\Upsilon(v,s,t)=(X_{v,s,t},\Sigma_{v,s,t},p_{v,s,t})$ and $\Upsilon(v,t)=(X_{v,t},\Sigma_{v,t},p_{v,t})$ respectively, where $p_{v,s,t}(f(\cdot))$ and $p_{v,t}(g(\cdot))$ denote the probabilities of sampling $f(\cdot)$ given v,s,t, and sampling $g(\cdot)$ given v,t respectively. We use $\mathbf{CM}(v,s,t)$ and $\overrightarrow{\mathbf{CM}}(v,t)$ to denote the corresponding random variables. We let $p(\mu_v|c_i^t,v)\approx p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)|c_i^t)\triangleq p(\overrightarrow{\mathbf{CM}}(v,t)=\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)|c_i^t)$, which allows us to deal with the difficulty of modelling $\Upsilon(v)$, by modelling $\Upsilon(v,t)$ instead. Similarly, $p(\mathbf{CM}(\cdot;\mu_v,v,s,t)|c_i^t)$ is the conditional probability of $\mathbf{CM}(v,s,t)=\mathbf{CM}(\cdot;\mu_v,v,s,t)$. If $p(c_i^t)$ denotes the prior non-zero probability that the target's centroid is in cell i, then by Bayes' theorem, $p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)|c_i^t)=$

$$\frac{p_{v,t}(\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t))}{p(c_i^t)}p(c_i^t|\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)). \tag{5}$$

Our goal is to appropriately model the generative probabilities of the feedforward hierarchies in order to calculate the probabilities in Eq.(5). We study some of the properties of an ideal target confidence map and use these properties to motivate the construction of a generative model for our localization algorithm that shares similar properties. In general, the higher the firing rate at a confidence map pixel $\vec{q}$, the more likely the target projects on that position. This observation is used in Thm. 2 in the Appendix. Intuitively, Thm. 2 formalizes the ideal behaviour of the confidence maps, whereupon, the greater the belief that the target object projects on a particular image position (based on the firing rate at the corresponding confidence map position), the less likely it is that we would witness at least that intense a firing rate if we were to pick an arbitrary pixel of the target confidence map. This simple model motivates the algorithm for updating the probabilities in Eq.(5) and for localizing target object positions by attending to a particular scale and position in the target confidence maps.

From Thm.1 and by the approximation $p(\lambda_n|c_i^t,v_n)\approx p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_{v_n},v_n,t)|c_i^t)$ (recall that by Def.1 $p(\lambda_n|c_i^t,v_n)=p(\mu_{v_n}|c_i^t,v_n)$) we have $p(c_i^t|\lambda_n,v_n,...,\lambda_1,v_1)=$

$$\frac{p(c_i^t|\lambda_{n-1},v_{n-1},...,\lambda_1,v_1)p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_{v_n},v_n,t)|c_i^t)}{\sum_j p(c_j^t|\lambda_{n-1},v_{n-1},...,\lambda_1,v_1)p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_{v_n},v_n,t)|c_j^t)} \tag{6}$$

In Sec.II-D we discuss how we deal with the high dimensionality of vector $\overrightarrow{\mathbf{CM}}(\vec{q};\lambda_n,v_n,t)\ \forall\vec{q}$. We use an approximation to $p(\lambda_n|c_i^t,v_n)$ that models target uniqueness within each individual scene viewpoint and across multiple scene viewpoints.

D. Interpolating the Probability

The high dimensionality of a vector $\overrightarrow{\mathbf{CM}}(\vec{q};\mu_v,v,t)$ makes it preferable to do the generative modelling in Eq.(6), by applying a dimensionality reduction technique on $\overrightarrow{\mathbf{CM}}(\vec{q};\mu_v,v,t)$ and keeping only the most relevant map information at each step. We achieve this by attending only to the most relevant intrinsic scales of $\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)$, and by building an interpolation model for approximating $p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)|c_i^t)$. This section is devoted to this purpose. We begin by detailing the abstract data types needed to define the "knots" of the interpolation model for $p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)|c_i^t)$.

We use $proj_1(i,v,t)$ to denote the confidence map intrinsic scale that best matches the scale of the expected projection of object t on the image plane under sensor state v, assuming the target's centroid coincides with cell i's centroid. In practice, we estimate $proj_1(i,v,t)$ as earlier when estimating the visible cells, namely, by calculating the size of the projection on the image plane of the target bounding cylinder centred in cell i. We constrain our search on the target confidence maps with an intrinsic scale of $proj_1(i,v,t)$: $p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)|c_i^t)\approx$

$$p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t). \tag{7}$$

Thus, the right-hand-side of Eq.(5) is approximated by

$$p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)) \times \frac{p_{v,proj_1(i,v,t),t}(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))}{p(c_i^t)}. \quad (8)$$

Given a target confidence map for sensor state v, $proj_2(i,v)$ denotes the pixel on the target confidence map on which the centre of target map cell i projects.

In accordance with the monotonic behaviour of the ideal confidence maps (Thm. 2), we use the cumulative distribution of $\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)$, based on the histogram of the pixel firing rates induced by visible cells that lie in our search space, to ensure the monotonicity property of Thm. 2 is preserved for the arbitrary probability distributions that can occur in practice and to provide an image specific measure of uniqueness. Thus, $\forall i_1,i_2,\ 0\leq i_1<i_2\leq 1$ we have:

$$\begin{aligned}&p([i_1,1]\in\mathbf{CM}(\,\cdot\,;v,proj_1(i,v,t),t))\approx\\&p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)\geq i_1)\geq\\&p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)\geq i_2)\approx\\&p([i_2,1]\in\mathbf{CM}(\,\cdot\,;v,proj_1(i,v,t),t))\end{aligned} \quad (9)$$

where the second and third probabilities are calculated using the corresponding histogram of $\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)$, as described above. The first and fourth probabilities are defined in Thm.2 and denote the expected fraction of firing rates in a random confidence map, that lie in $[i_1,1]$, $[i_2,1]$.

The firing rate corresponding to an object whose centroid coincides with the centre of cell i, is given by $\mathbf{CM}(proj_2(i,v);\mu_v,v,proj_1(i,v,t),t)$. By Thm. 2, if we have a "good" recognition algorithm, the more likely the target object is centred in cell i — based on an increased firing rate i_1 = $\mathbf{CM}(proj_2(i,v);\mu_v,v,proj_1(i,v,t),t)$ for example —, the smaller the value of $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)\geq i_1)$ is, signifying the rarity and importance of the image region. The rarity of this firing rate, within the context of the firing rates present in the current confidence map, is used in the prior placed in the numerator of Eq.(8), which leads to:

$$\begin{aligned}&p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)|c_i^t)\approx\\&p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))\frac{p(\beta_i)}{p(c_i^t)}.\end{aligned} \quad (10)$$

where $p(\beta_i) \triangleq p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t) \geq \mathbf{CM}(proj_2(i,v);\mu_v,v,proj_1(i,v,t),t))\approx$ $p_{v,proj_1(i,v,t),t}(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))$. Based on Thm. 2, the smaller the prior $p(\beta_i)$, the more likely the target is centred in pixel $proj_2(i,v)$. Notice that in contrast to $p_{v,proj_1(i,v,t),t}(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))$, $p(\beta_i)$ provides a localized measure of uniqueness, around $proj_2(i,v)$ and within the context of a single-view μ_v, as a means of compensating for our poor knowledge of probability space $\Upsilon(v,proj_1(i,v,t),t)$ and our consequent inability to calculate $p_{v,proj_1(i,v,t),t}(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))$.

We proceed by using an interpolation scheme to model the probability $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))$. We begin by stating some definitions and some of the properties that the probability must satisfy. While $p(\beta_i)$ provides an image specific measure of the uniqueness of each marked cell i, we use $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))$ in conjunction with $p(\beta_i)$ to model a global measure of the target likelihood across the multiple images acquired during search. Eqs.(12)-(15) in the Appendix specify the parameter values of the interpolation model we will use to achieve this. Before Eqs.(12)-(15) are presented however, we need to motivate the construction of the interpolation model and define the model parameters.

Recall the definition of μ_v as a scene sample function that was acquired under sensor state v (see Def.1). Assume $\beta_{topN}(\mu_v,v,t,i)\in[0,1]$ is the ratio, with respect to the total area of μ_v's image, of the area on μ_v's image taken up by the projection of the bounding cylinder of target t, assuming target t is centred in cell i in the scene. Then $topN(\mu_v,v,t,i)$ is defined as the smallest value in interval [0,1] that satisfies $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t) > topN(\mu_v,v,t,i)) \leq \beta_{topN}(\mu_v,v,t,i)$. Similarly, $top(\mu_v,v,t,i)$ is defined as the smallest value in interval [0,1] that satisfies $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t) > top(\mu_v,v,t,i)) = 0$. Finally, $bottom(\mu_v,v,t,i)$ is defined as the largest value in interval [0,1] that satisfies $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)> bottom(\mu_v,v,t,i)) = 1$. Typically, as is the case in our feedforward hierarchy, $bottom(\mu_v,v,t,i)=0$.

For target t and intrinsic scale s, we define $eer(s,t)$ as the **equal error rate** of the corresponding confidence map's firing rates. The equal error rate $eer(s,t)$ is the firing rate threshold when it is equally likely for a confidence map firing rate above or below that threshold to represent a false-positive or a false-negative with regards to target t projecting at a scale s. The equal error rates are estimated during the training process overviewed in Sec.II-E. Thus, ideally, if $\vec{q} = proj_2(i,v)$ and $s = proj_1(i,v,t)$, $p(\mathbf{CM}(\vec{q};v,s,t) > eer(s,t)|c_i^t) = p(\mathbf{CM}(\vec{q};v,s,t) < eer(s,t)|\neg c_i^t)$, which implies that for all scales s', $eer(s',t)$ provides a firing rate threshold which is equally likely to represent the presence and the absence of the target. We use the equal error rates to normalize the firing rates across scales and make them comparable with each other.

By Assumption 1 and presuming an ideal confidence map, if the target is present in μ_v, then part of the target must project somewhere on the image with a firing rate of at least $topN(\mu_v,v,t,i)$, effectively meaning that any image region with a firing rate less than $topN(\mu_v,v,t,i)$ does not correspond to a target projection. In conjunction with the single image specific measure of uniqueness $p(\beta_i)$, we use the equal error rates to normalize the firing rates across scales and make them comparable to each other, thus, adding a global measure of uniqueness across all captured images. Each of $topN(\mu_v,v,t,i)$, $top(\mu_v,v,t,i)$ and $bottom(\mu_v,v,t,i)$ is mapped to probabilities $p_{topN}(\mu_v,v,t,i)$, $p_{top}(\mu_v,v,t,i)$ and $p_{bottom}(\mu_v,v,t,i)$ respectively, using linear functions (defined in the Appendix) which take into account the effects of the equal error rates. As we will see, these normalized probabilities specify the "knots" of the interpolation model at three possible values of $p(\beta_i)$.

We now have the means of presenting the approximation to $p(\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)|c_i^t)$ used in Eq.(6). As hinted by Eq.(10), we can estimate this probability by modelling $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))$ as a non-increasing function of $p(\beta_i)$ that also depends on p_{topN}, p_{bottom}, p_{top}, $p(c_i^t)$, β_{topN} and satisfies the following constraints:

(i) $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)) \leq min(1, \frac{p(c_i^t)}{p(\beta_i)})$.

(ii) If $p(\beta_i) = \beta_{topN}(\mu_v,v,t,i) > p(c_i^t)$, then the probability $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t)$ is set equal to $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))\frac{p(\beta_i)}{p(c_i^t)} = p_{topN}(\mu_v,v,t,i)$.

(iii) If $p(\beta_i) = 1 > p(c_i^t)$, then the probability $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t)$ is set equal to $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))\frac{p(\beta_i)}{p(c_i^t)} = p_{bottom}(\mu_v,v,t,i)$.

(iv) If $p(\beta_i) = p(c_i^t)$, then the probability $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t)$ is set equal to $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))\frac{p(\beta_i)}{p(c_i^t)} = p_{top}(\mu_v,v,t,i)$.

(v) If $p(\beta_i) < p(c_i^t)$, then the probability $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t)$ is constant for all $p(\beta_i) < p(c_i^t)$ and is set equal to $p_{top}(\mu_v,v,t,i)$.

Constraints $(i) - (v)$ guarantee that if $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t) \geq 0.5$, the firing rate of $top(\mu_v,v,t,i)$ exceeds $eer(proj_1(i,v,t),t)$ (providing a global measure of uniqueness) and $p(\beta_i)$ is sufficiently small (guaranteeing local target uniqueness within μ_v). The higher the value of $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t)$, the more likely it is that the target projects on the image plane and comparisons of the generative probabilities across different μ_v become meaningful. Furthermore, the definition of p_{topN} guarantees that if $p_{topN} = 0.5$, the number of cells that can be assigned $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t) \geq 0.5$ is constrained by the expected projection size of the target object on the image plane. Notice that for $p(\beta_i) = 0$, Eq.(10) has a value of zero. However, the value $p(\beta_i) = 0$ signifies a rare event for which we would like to assign a high probability to Eq.(10), which is why we treat the case $p(\beta_i) < p(c_i^t)$ separately (remember that $p(c_i^t) \neq 0$ for any updated cell i). Notice that in our online test runs, we assign a uniform distribution to each cell in our search space, which implies that $\forall i, p(c_i^t) << 1$, meaning that in practice case (v) plays a role for very few cells i and $\beta_{topN}(\mu_v,v,t,i) > p(c_i^t)$.

As long as we model $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))$ as a non-increasing function of $p(\beta_i)$ that also satisfies constraints $(ii) - (v)$ from above, it will also satisfy constraint (i), making $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))$ a valid probability. We model $p(c_i^t|\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t))$ as a piecewise differentiable and non-increasing function of $p(\beta_i)$ composed of piecewise components of the form $\frac{\alpha_j}{p(\beta_i)} + \gamma_j$ for each interval j, which reduces $p(\mu_v|c_i^t,v)$ to a function of $p(\beta_i), p_{top}(\mu_v,v,t,i), p_{topN}(\mu_v,v,t,i), p_{bottom}(\mu_v,v,t,i)$, $p(c_i^t)$ and $\beta_{topN}(\mu_v,v,t,i)$, that is piecewise linear in terms of $p(\beta_i)$. An LU-decomposition provides the solution for $p(\beta_i) \in [p(c_i^t), \beta_{topN}(\mu_v,v,t,i)]$ (specified by assigning values to the parameters α_1,γ_1) and for $p(\beta_i) \in [\beta_{topN}(\mu_v,v,t,i),1]$ (specified by assigning values to the parameters α_2,γ_2). Analytic expressions for these parameters are in the Appendix. The approximation of $p(\mu_v|c_i^t,v)$ by modelling $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t)$, provides a compromise between modelling an image specific measure of uniqueness — recall that $p(\mathbf{CM}(\cdot;\mu_v,v,proj_1(i,v,t),t)|c_i^t)$ is a function of $p(\beta_i)$ — and a global measure of uniqueness across all images — we take into consideration an absolute measure of uniqueness across all images, via constraints $(i) - (v)$. In conjunction with Eqs.(6) and (10), this completes the discussion on updating of the target map probabilities when cell $i \in V'(v_n) \triangleq V(v_n;v_n) \cup M(v_n)$. If $i \notin V'(v_n)$, we use Assumption 1 to set an equal generative probability $\forall i \notin V'(v_n)$, by letting $p(\lambda_n|c_i^t,v_n) \leftarrow \min_{j\in V'(v_n)}\{1-p(\lambda_n|c_j^t,v_n)\}$.

Notice that by our construction of the generative probabilities, $p(\lambda_n|c_{i_t}^t,v_n) \approx 1$ when $i_t \in V'(v_n)$, where i_t is the cell where the target object is centred. One might argue that our formulation is incorrect since typically $\sum_\lambda p(\lambda|c_i^t,v) > 1$, for arbitrary i. However, as we see from Eq.(4), for all $1 \leq i \leq |C|$ the discriminative probability $p(c_i^t|\lambda_n,v_n,...,\lambda_1,v_1)$ is independent of scale factors applied on the generative probabilities $p(\lambda_n|c_i^t,v_n)$, implying that the ratios of the generative probabilities is what characterizes Eq.(4), and not the individual magnitudes of the generative probabilities.

E. Building the Target Confidence Maps

To calculate the target confidence maps $\overrightarrow{\mathbf{CM}}(\cdot;\mu_v,v,t)$ of target t for a given RGB input image that was acquired under sensor state v (*e.g.*, scene sample function μ_v), we apply a multi-scale convolutional template matching. This is not done on the original RGB images but on the output of the hierarchical feed-forward architecture described in detail in [36] and over-viewed in this section. This architecture is based on weight-sharing and a succession of feature detection and pooling stages (see Fig. 3(a)) and is meant to simulate some of the shape processing mechanisms of the ventral visual pathway. As it is shown in [40] and Fig.3(b), this recognition model can be trained interactively in an online fashion for up to 50 arbitrary objects by manual demonstration, using unconstrained in-hand rotation. Unlike other methods, like SIFT-based recognition, it imposes no constraints on strong planar textures or canonical views of the objects.

The first feature-matching layer S1 is composed of four orientation sensitive Gabor filters. We use a threshold function to apply a Winner-Take-Most mechanism between features located at the same position in each map. The subsequent C1 layer, sub-samples the S1 features by pooling down to a quarter of the original resolution in both directions, using a Gaussian receptive field and a sigmoidal nonlinearity. The fifty features in the intermediate layer S2 are obtained by sparse coding and are sensitive to local combinations of the features from the C1 layer. Layer C2 again performs spatial pooling and reduces the resolution by half in each direction. The fifty shape maps in C2 are extended by three color maps, generated by down-sampling the RGB channels of the input image.

The templates are trained by acquiring one-thousand views of each one of the target objects. Fig.4 shows examples of views of the objects used in our results. The objects are held in front of a cluttered background and frames are grabbed using HR's stereo camera system. The region containing the object is determined based on a depth criterion and is scaled to a fixed output resolution, as described in [40], [41]. Besides the object views, a large set of clutter views are collected and used as negative training examples.

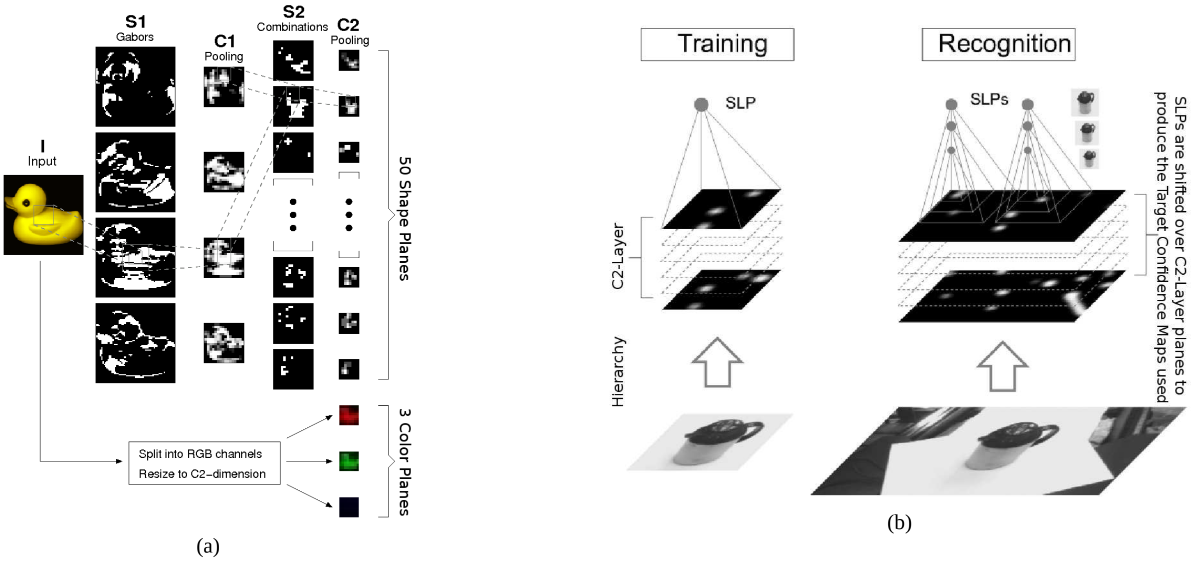


Fig. 3. (a)Feed-forward architecture of [36]. The architecture consists of a shape path and a color path. The shape path consists of several layers. The S1-layer convolves the gray-scale image with gabor filters of 4 different orientations and computes a Winner-Take-Most nonlinearity. The C1-layer pools the magnitude of the results to a lower resolution. The S2-layer responds to local combinations within the C1-layer and the C2-layer performs an additional pooling. The color path splits the input image into its RGB channels and down-samples them to match the dimension of the planes in C2. (b)Training and use of SLPs. In the training process object views were presented in different scales to generate scale-sensitive SLPs on top of C2. In the recognition process the receptive fields of these SLPs were shifted over the C2-layer of the whole scene to get a position and size-sensitive response, in the form of target confidence maps over seven intrinsic scales. The size of the target object's projection on the image plane, specifies the intrinsic scale that returns the best response.

The template responds strongly to views of the current object and responds weakly to views of other objects or clutter. We train a single layer perceptron (SLP) for each combination of object and scale. We use an SLP with a sigmoidal non-linearity to restrict the output range to $[0, 1]$. For a given scale, first the images in the database are down-sampled and afterwards, their C2 activations are calculated. Then, an SLP is trained for each object, using its own views as positive examples and all other views as negative examples.

The training is done for seven different intrinsic scales ($1 \leq s \leq 7$), covering object sizes between 64×64 and 160×160 pixels from input images of 800×600 pixels. During the search for a certain object, the corresponding seven scale-sensitive template SLPs are used to convolve the C2 activation of the current input image, as shown in Fig. 3(b). The output corresponding to each scale's SLPs defines the target confidence map $\mathbf{CM}(\cdot; \mu_v, v, s, t)$. Examples of target confidence maps over multiple scales are available in the supplementary material documentation.

To train the SLPs, we use 800 views per object. The remaining 200 views are used to evaluate the offline recognition performance. When classifying a test image by determining the maximal activated SLP, we observe that for the smallest scale, views of different objects are confused in about 25% of the cases, whereas, for the largest scale, the error rate was 19%. However, in this work the task is not object recognition (*i.e.*, competition of different object hypotheses), but localization of a pre-specific target object. Therefore, it is more important to determine how well the SLPs corresponding to the target, separate its views from all other input (views from other objects but mainly clutter views). This is addressed by means of an ROC analysis which shows the relation between the false-positive and false-negative rate of detection as a function of a threshold parameter. The equal error rate (EER) denotes the threshold value where both false-positive and false-negative rates are equal. Fig. 4(c) shows that some objects have a very low probability of being confused (*e.g.*, objects 4, 7, 9) while for other objects, this separation is worse (especially for objects 13, 14, 16). These results are reflected in the evaluation of the object search performance in Secs.III,IV. Despite these differences, the chosen representation and processing is not constrained to certain types of objects since HR can learn a representation of the target object directly before the search.

III. Experimental Setup

A. Test Protocol

We evaluate the active localization algorithm by its reliability and speed in localizing the target objects. To evaluate our method systematically and in a reproducible way, we record a number of data sets, and use these in offline simulated test runs. We also perform real-time tests verifying the online performance. We test our algorithm by searching for twenty different targets (Fig.4) under five different test scenarios. Note that the search space in Scenarios 3,4,5 is significantly larger than that of Scenarios 1,2 (see Fig.5). In each scenario, all target objects are positioned in the scene 'upright', on their pre-specified bases. The cost function in Scenarios 1,2,3,5 is defined as $\mathbf{t_o}(v_{n-1}, v_n) = c_1 + c_2 d_n$ and depends on the optimal path distance d_n chosen by Dijkstra's algorithm, where constant c_2 denotes the inverse of HR's walking speed and c_1 is the expected processing time for all other components in each iteration of the algorithm's loop (Fig.2). The target, obstacle and never-viewed maps are discretized using $5cm \times 5cm \times 5cm$ cubes, as in [5]. Scenarios

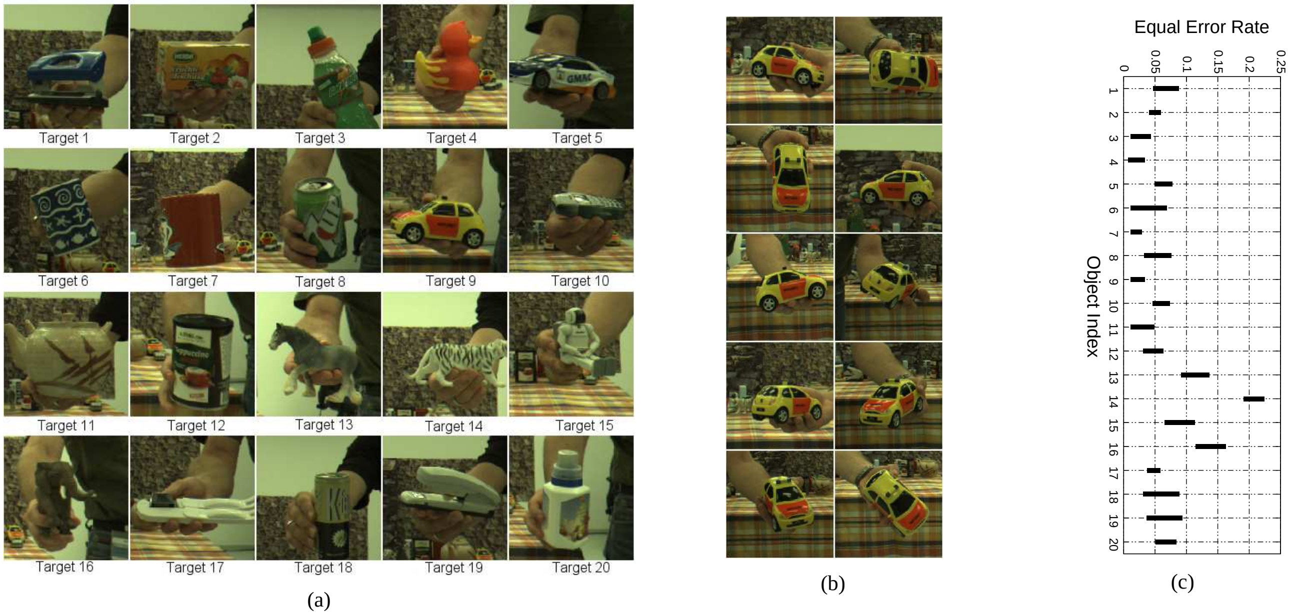


Fig. 4. (a) Random samples from the training set used to learn each of the twenty objects. **Target 1**: Hole Puncher. **Target 2**: Fruit Tea Box. **Target 3**: Gatorade Bottle. **Target 4**: Tiger Duck. **Target 5**: Opel Race Car. **Target 6**: Textured Cup. **Target 7**: Red "cafe" mug. **Target 8**: 7up Can. **Target 9**: "Notarzt" Ford. **Target 10**: Space Nokia. **Target 11**: Tea Pot. **Target 12**: Capuccino Box. **Target 13**: Horse. **Target 14**: White Tiger. **Target 15**: Asimo Sitting. **Target 16**: Elephant. **Target 17**: Garlic Press. **Target 18**: Koffee Dose Can. **Target 19**: Stapler. **Target 20**: Compo Fertilizer. (b) The recognition model is trained by manual presentation of each of the twenty objects, with 3D rotations covering the expected robot viewing variation. (c) Equal error rates for each of the twenty objects. The bar lengths denote the variation of the equal error rates across the seven scales used.

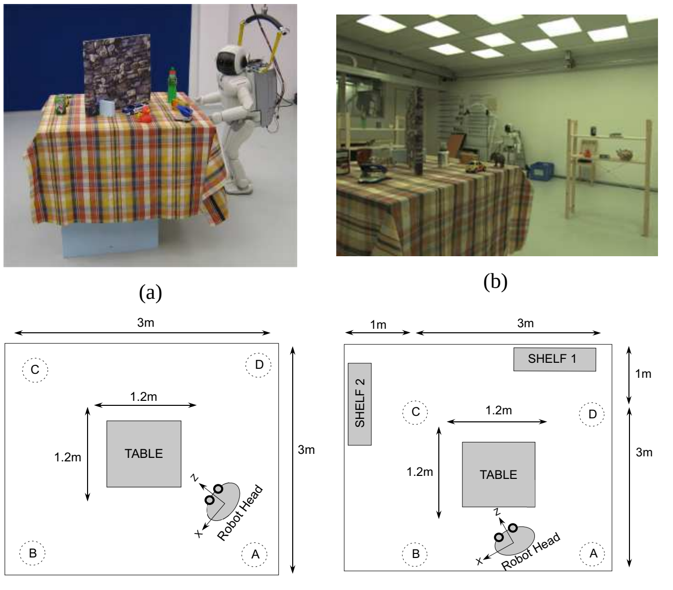
1 and 2 investigate the performance of the algorithm as the number of possible viewpoints from which the search region is sensed, increases. The comparisons between Scenarios 3 and 4 quantify the benefits of using the greedy next-view-planner (Scenario 3), as compared to simply moving at each step to the scene position where the probability of detecting the target is maximized (Scenario 4), by having $\mathbf{t_o}(\cdot,\cdot)$ always return a constant movement cost value in Scenario 4. The comparison between Scenarios 3 and 5 quantifies the benefits of using our greedy where-to-look-next algorithm (Scenario 3) as opposed to randomly searching for the target, by assigning a random score to each of the candidate hypotheses $v \in CL'$ in Eq.(2) for which there exists a path from the current sensor state to state v (Scenario 5). The comparison between Scenarios 4 and 5 quantifies the role that the target maps play in choosing the best next view for detecting the target, when we ignore the sequence cost function $\mathbf{t_o}(\cdot,\cdot)$. In Section III-B we describe each scenario's dataset in detail. Notice that since Scenarios 3,4,5 differ only in the next-view-planner used, the same offline dataset is used in these three scenarios. For each of the five scenarios, we execute 80 test runs. In these 80 test runs, we search for each of the 20 objects by starting the search from the four positions shown in Fig.5.

To evaluate the search performance in each scenario, we use the ground truth position of all the objects in the scenario, with respect to the world coordinate frame. For each object in each scenario, we measure its position (its centroid) in the world coordinate frame using a measuring tape, as a means of evaluating target localization reliability. However, using these measurements alone by themselves, to determine whether HR has localized the target, is insufficient. This is because of potentially small errors in making these measurements, dead-reckoning errors in the estimates of the heel-positions in the samples of our dataset, small stereo depth estimation errors, as well as irregular object surfaces. We, thus, define two metrics based on which the results in Sec. IV are built:

The **image score** of a particular scene sample function μ_{v_i} is defined as $\max_{j \in M(v_i)} p(\lambda_i | c_j^t, v_i)$, the maximum generative probability of all the marked candidate cells of step i. The **maximal target image** im_{max} of a given test run, is the image with the highest image score amongst a set **S** of images. We define **S** as the largest subset of the set of images captured during the test run, that satisfies the following constraint: $\forall \lambda_i \in \mathbf{S}$, $\exists j_m \in M(v_i)$ such that the centroid of cell j_m projects on the target object in image λ_i, $j_m = \operatorname{argmax}_{j \in M(v_i)} p(\lambda_i | c_j^t, v_i)$, and the estimated ground truth of the target's centroid in the world coordinate frame (estimated using a measuring tape, as previously described) is within distance ϵ of cell j_m's centroid.

For each image in a test run, there corresponds an image score and the 3D world coordinate of the associated cell. If the 3D cell of an image score falls within $20cm$ of the expected target position, by projecting the 3D coordinate of the cell back in the image plane we visually determine whether the image score was due to detection of the object, independently of dead-reckoning errors. As we discuss in more detail in Sec.IV, by finding for $\epsilon = 20cm$ the maximal target images of numerous test runs, and by investigating how their respective image scores rank compared to other image scores, we obtain a good evaluation metric for the algorithm.

We use the Small Vision System by SRI International for the stereo depth extraction [42]. Our system was developed using a set of tools created by Honda for building large scale distributed intelligent systems [11]. These include component



(a) (b) (c) (d)

Fig. 5. (a)Scenario 1 and 2 setup, with targets 1-10 situated on the table. (b)Scenario 3,4,5 setup with all 20 targets present in the scene, as viewed by HR's head camera. (c)Bird's eye view of the setup of Scenarios 1 and 2 within a $3m \times 3m$ maximum walk and search region. (d)Bird's eye view of the setup of Scenario 3,4,5 in the $4m \times 4m$ walk and search region. Points A,B,C,D represent the four different starting positions of HR.

models BBCM and BBDM (Brain Bytes Component/Data Model respectively), design and monitoring systems, and the middleware RTBOS (Real Time Brain Operating System) for executing the component models on a variety of computer platforms. The diagram in Fig.2 outlines a component-wise breakdown of our system. All component models are coded in C. We employ the walking algorithm and whole-body motion control system that was developed for use with HR [43].

We use a hypothesize-and-test next-view-planner and as such, it is easily parallelizable. We take full advantage of this to make our system real-time and suitable for live demonstrations. To speed up the algorithm, the hypothesis evaluation for the next-view-planning is applied on a coarser scale of the target, obstacle and never-viewed maps, by reducing the resolution of each dimension of the maps by half. The neighbourhoods γ_i in Eq.(2), correspond to the dimensions of the cells used in these coarser-scale maps. Furthermore, the hypotheses are evaluated in parallel on eight threads running concurrently on a server with two Quad-Core CPUs.

B. Test Data

We now describe the creation of the offline datasets. In all scenarios, HR starts the search from four different initial heel positions A, B, C, D, as shown in Fig.5(c),(d). We have also implemented an online version of the system, which works in real-time ($c_1 \approx 3\mathbf{s}$, $c_2 \approx 3\mathbf{s}/\mathbf{m}$ in the cost function) and thus, does not rely on the view-sampling data of the offline version of the loop. The online system is currently being used for real-time demonstrations of this work, in which HR points at the object once it is localized. A demonstration of online search is available in the supplementary material section of the journal. Online and offline search differ in that the offline dataset is created by acquiring one sample image for each element in a set $CL'' \subseteq CL$ (see Sec.II-B), and thus for the offline testing, the optimization in Eq.(2) takes place over the corresponding subset $\mathtt{map}(CL'') \subseteq CL'$, while the path planner still optimizes its paths over ML' (see Eq.(1)).

Scenarios 1 and 2 (Fig.5(a),(c)) take place inside a $3m \times 3m \times 1.5m$ search region and involve placing the targets on a table with a $1m \times 1m$ surface area at $0.84m$ height, and having HR search for each of the twenty targets in a $1.2m \times 1.2m \times 1.2m$ region encompassing the table. In Scenarios 1,2, the target map's prior is uniformly distributed inside the $1.2m \times 1.2m \times 1.2m$ region and is assigned a zero prior probability everywhere else in the search region (see Fig.5(c)), effectively instructing the algorithm to ignore the zero probability regions. Notice that since these zero prior probability regions do not contain any solid structure, our algorithm assigns them a zero target map probability (see Sec.II-C), even when their prior is not set to zero. So by setting certain regions to a zero prior, we are effectively investigating the algorithm performance when searching regions containing mostly solid structure and occlusions, where a recognition algorithm is needed to determine if the target object is present. We place 10 objects at a time on the table. When we are searching for one of targets 1-10, targets 1-10 are positioned on the table and when we are searching for one of targets 11-20, targets 11-20 are positioned on the table. A separating wall is always placed bisecting the table's surface to limit the number of viewpoints from which each target is visible.

In Scenario 1, we create the offline dataset HR uses, by having HR sample the search space by facing the table while simultaneously walking sideways with $0.5m$ step intervals around the periphery of a $2m$ by $2m$ square path centred at the table's centre (Fig.5(a)). At each step, HR acquires six images and the corresponding heel coordinate and eye coordinate frames of HR, that uniformly sample the search region, for a total of 102 pairs of stereo images (*i.e.*, $|CL| = 102$). Each one of these image pairs represents a candidate hypothesis which is evaluated when determining where to move next. This allows us to perform rigorous and exhaustive testing of the algorithm's performance, that is difficult to perform using an online version of the loop. Note that the order in which we acquire the images is irrelevant, and what is important is to have accurate information on the heel coordinate frame and the eye frame coordinates under which each image is acquired. In Scenario 2, we enlarge the set of images, by having HR walk around a $2m$ by $2m$ square path and a $3m$ by $3m$ square path centred at the table's centre, while maintaining the same $1.2m \times 1.2m \times 1.2m$ uniformly distributed search space region, and using $0.5m$ steps with HR always facing the table (Fig.5(a)). This enlarges the set of images/candidate hypotheses to 252, gives greater variability in the scales with which each object is sampled, and increases the number of candidate hypotheses, while maintaining the Scenario 1 prior.

In the last three scenarios (Scenarios 3, 4, 5) we enlarge the size of the search space and the number images/candidate hypotheses. The search space consists of a $4m \times 4m \times 1.5m$ region (Fig.5(b),(d)) with the same table centred inside the

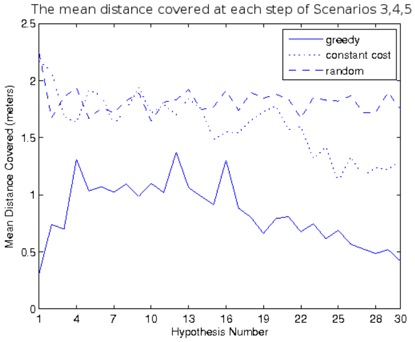


Fig. 6. The mean distance covered for each executed hypothesis of Scenarios 3,4,5 (we graph the first 30 executed hypotheses), using three different next-view-planners: The greedy algorithm, using a constant cost function and a random next-view-planner. A single tailed t-test shows that there is a statistically significant difference ($p \approx 0.02$) between the constant cost and random planner. Between the other two pairs of next-view-planners the p-value is smaller ($p < 0.001$).

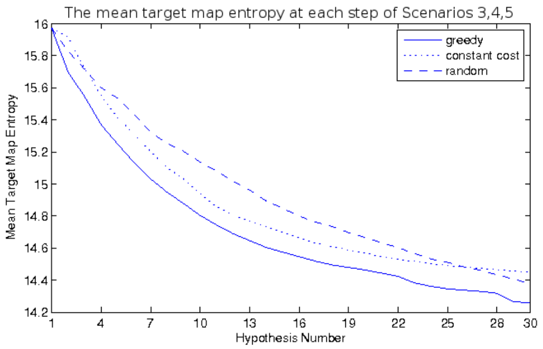


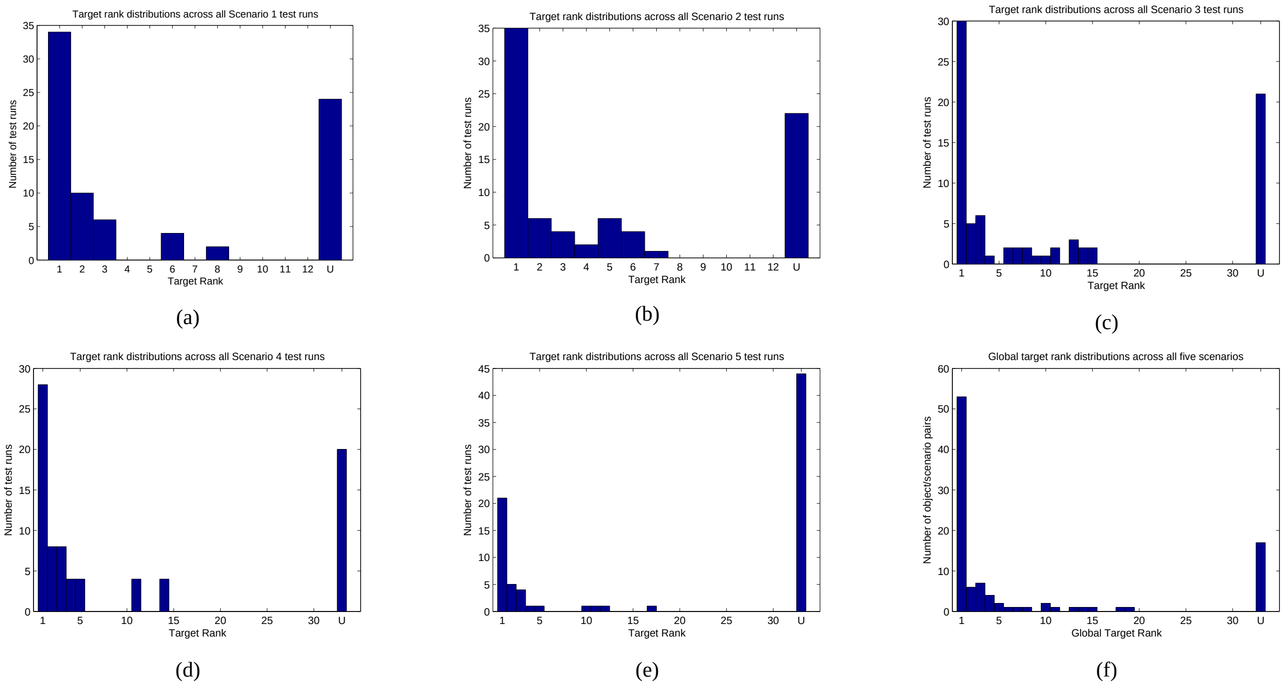
Fig. 7. The mean entropy of the target maps for all twenty objects for the first 30 executed hypotheses of Scenarios 3,4,5. Notice that the greedy algorithm consistently outperforms the entropies of the constant cost next-view-planner and the random next-view-planner. A single tailed t-test shows that there is a statistically significant difference ($p < 0.001$) between all three pairs of next view planners.

bottom $3m \times 3m$ region and two shelves positioned in the top-most and left-most part of the region, as shown in Fig.5(d). The target map prior is set to a uniform prior distribution at a $1.2m \times 1.2m \times 1.2m$ region containing the table and at the top-most $1m \times 4m \times 1.2m$ and left-most $4m \times 1m \times 1.2m$ region containing the shelves, as shown in Fig.5(d). This specifies the volume we want to search for objects. Everywhere else in the search region, the target map prior probability is set to zero as per Scenarios 1, 2. This zero prior can speed up the search by pre-specifying large empty-space regions which cannot contain the target object. Such zero-prior regions could be specified manually or determined automatically before the search starts, using vision sensors or range finders (lasers and sonars are more reliable than vision sensors are in poorly textured regions) in conjunction with standard SLAM algorithms, since empty regions obviously cannot contain the target object(s) we are searching for. Notice, however, that our system's implementation does not presuppose the existence of such zero priors, nor does it necessarily require them in order to function correctly. If we are dealing with an environment whose obstacle layout does not change significantly over time, the use of such zero prior regions is preferable, as it would result in faster search times during future online search runs, by inhibiting the costly rediscovery of large obstacles that affect the path planner. Five objects are placed on each shelf and the other ten objects are placed on the table. HR creates the offline dataset by moving around the periphery of a $2m$ by $2m$ square path and a $2.5m$ by $2.5m$ square path centred at the table's centre in a clockwise and counter-clockwise direction. Each step interval is $0.5m$ long. For each step, fifteen images are acquired, uniformly sampling the region in front of HR (pan range [-80, 80] degrees, tilt range [-15, 30] degrees), resulting in fifteen images/candidate hypotheses for each step. Since HR moves in both a clockwise and counter-clockwise direction, for each heel position thirty images are acquired, densely sampling the entire search region. This results in 1110 images/candidate hypotheses that HR can choose from for its next view (*i.e.*, $|CL| = 1110$). In Scenario 3 we use the above set of candidate hypotheses to test the full algorithm described in this paper. In Scenario 4 we assign a constant value to the cost of each movement, effectively making the cost function independent of the current position of HR. As long as the entire walk space is accessible from each position, the constant cost scenario is independent of HR's starting position. In Scenario 5, we randomly choose the next movement from the 1110 hypotheses available in our dataset, as per Sec. III-A.

HR is a bipedal robot, with good dead-reckoning precision compared to typical wheeled robots. This allows us to focus on the object localization problem, without worrying about the errors in localizing the position of HR within the map. As long as HR completes its search within a certain number of steps, we can assume that HR's dead-reckoning is fairly accurate. In order to quantify this claim, in most sequences of captured images, HR started and ended from the same heel coordinate. In none of these cases where the error was quantified, was HR's ending heel position more than about $10cm$ away from its starting heel position. In all cases HR covered a total of 8-20 meters and rotated a total of roughly 360-720 degrees, demonstrating good dead-reckoning precision. Both in the online mode and in the offline mode — during the offline dataset creation — HR lost most of its dead-reckoning precision during rotations. We, thus, minimized the number of rotations performed during the dataset creation. For each executed path, HR either walks (forwards, backwards, sideways or diagonally) or makes an on the spot rotation, and avoids high-curvature turns while walking.

IV. Results

The goal of this project is to have HR search in a room for a certain object and once the object is found, to have HR point at it. Therefore, one metric based on which we judge the quality of our localization algorithm is the number of pointing actions HR would have to execute until it points at the correct object. The *target rank* is the metric that we use for this purpose: Assume we are given a list of the image scores for all the distinct images captured in a given test run, where the concept of an image score was defined in Sec.III-A. Also assume that the image scores are sorted in descending order and based on



(a) (b) (c)

(d) (e) (f)

Fig. 8. (a)-(e)The distribution of the target ranks for the test runs of each one of Scenarios 1-5 respectively. (f) The distribution of the global ranks for all five different scenarios. Any global rank that is unknown or that is greater than thirty, corresponds to one tick in the bin labelled U. Detailed tables of the results on individual test runs from which these tables are derived, are available in the supplementary material section of the paper.

TABLE I

THE MEAN±STANDARD DEVIATION AND THE MEDIAN NUMBER OF EXECUTED HYPOTHESES (hyp) AND DISTANCE COVERED IN METERS ($dist$) UNTIL THE MAXIMAL TARGET IMAGE IS ACQUIRED, USING THE TEST RUNS WHERE THE TARGET IS ASSIGNED A RANK OF ONE ($d = 1$), USING THE TEST RUNS WHERE THE TARGET IS ASSIGNED A RANK OF AT MOST THREE ($d \leq 3$), USING THE TEST RUNS WHERE THE TARGET IS NOT ASSIGNED A RANK OF U ($d < U$) AND USING THE TEST RUNS WHERE THE TARGET IS ASSIGNED A RANK OF AT MOST U ($d \leq U$). NOTICE THAT FOR CASE $d \leq U$, IF IN A CERTAIN TEST TRIAL THERE IS NO MAXIMAL TARGET IMAGE (THUS BEING ASSIGNED A RANK OF U IN TABLES I,II), WE USE THE TOTAL NUMBER OF EXECUTED HYPOTHESES AND THE TOTAL DISTANCE COVERED IN PERFORMING THE CALCULATION. $\mathbf{t_o} = c_1 \cdot hyp + c_2 \cdot dist$, WHERE $c_1 \approx 3\mathbf{s}$, $c_2 \approx 3\mathbf{s/m}$, PROVIDES AN ESTIMATE OF THE EXPECTED RUNNING TIME OF THE ONLINE SYSTEM, UNTIL A TARGET IS LOCALIZED.

	Scenario 1		Scenario 2		Scenario 3		Scenario 4		Scenario 5	
	mean	median	mean	median	mean	median	mean	median	mean	median
$\mathbf{d = 1}$										
hyp :	6.3 ± 3.1	6	6 ± 3.6	5	12.9 ± 6.8	11	14.9 ± 9.9	17	13.8 ± 8.9	16
dist :	10.4 ± 5.6	10.8	10.4 ± 6.6	8.7	12.8 ± 7.5	12	27.8 ± 17.2	31.0	26.5 ± 17.4	26.9
$\mathbf{d \leq 3}$										
hyp :	6.5 ± 3.1	6.5	5.9 ± 3.5	4	12.9 ± 6.7	12	14.5 ± 8.8	17	13.8 ± 9.2	12.5
dist :	10.8 ± 5.7	10.9	10 ± 6.6	8.7	12.6 ± 7.3	11.9	27.4 ± 15.1	31.0	25.4 ± 17.2	22.6
$\mathbf{d < U}$										
hyp :	6.1 ± 3.2	6	5.9 ± 3.5	4.5	12.8 ± 7.1	12	12.9 ± 8.7	9	14.8 ± 9.2	14.5
dist :	10.4 ± 6	10.8	10.3 ± 6.8	8.7	12.2 ± 7.8	11.9	24.2 ± 15.1	20.7	27.3 ± 17.1	25.8
$\mathbf{d \leq U}$										
hyp :	8.1 ± 4.1	8	7.7 ± 4.2	9	17.5 ± 9.9	15	17.4 ± 11.0	20	24.3 ± 10.6	31
dist :	12.4 ± 6.2	13.6	12.9 ± 7.3	15.1	15.6 ± 9.1	14.6	31.3 ± 18.7	32.5	43.1 ± 18.6	52.2

this sorted order, HR sequentially points at the corresponding image score cells. The target rank of this test run is defined as the position in this sorted list (its “rank” in the list), of the image score corresponding to the maximal target image. Ideally, the target rank has a value of 1, indicating that the first object HR points at is the object it is searching for. If no maximal target image is found in a given test run, we assign an “unknown” rank, denoted by symbol U. The *global rank* is similar to the target rank, only that the rank is evaluated with respect to the images acquired from all four starting positions A, B, C, D of any given scenario and any given object. Thus, for every global rank value, there correspond four target ranks. Fig.6 and Fig.7 compare the average distance covered and average target map entropy respectively for Scenarios 3, 4, 5, and for each executed hypothesis. Table I quantifies how long it typically takes to localize a target in each of the different scenarios, as explained in the table caption. As explained in the caption of Fig.8, the figure’s first five sub-figures show the distribution of target ranks for each individual scenario, while Fig.8(f) is the distribution of global ranks from all five scenarios, where for notational convenience any global rank that is unknown (U) or is greater than 30 is placed under the bin labelled U. Detailed analytical results of all the test runs from which the relevant graphs and histograms are derived, are available in the supplementary material section of the journal. Examples of the walk paths chosen by HR, as well as examples of how the obstacle maps and target maps evolve over an executed test run are also available in the supplementary documentation. We also performed a number of test runs with the online version of the active search algorithm, by searching for some of the targets that the offline test runs indicate are reliably localizable (targets 1,3,4,6,7,9,11), in order to confirm that the search reliability implied by the offline tests, also generalizes to the online case. All objects were successfully localized. As previously indicated, a demonstration of one such test run is available in the supplementary documentation.

V. Discussion

From Figs.8(a),(b) we observe few differences between Scenarios 1 and 2 (recall that both scenarios use the greedy next-view-planner). For example, the percentage of test runs with a target rank of 1-2 and a target rank of U are almost the same. We should point out that from the $3m \times 3m$ periphery of Scenario 2, most objects' projections on the image plane are too small to be recognized by the intrinsic scales of our feed-forward hierarchy, indicating that the greedy algorithm is capable of compensating by moving sufficiently close to the targets. Recognition rates and the distances covered until the target is localized, also remain similar, indicating that the greedy next-view-planner is not sensitive to an increased set of viewpoints on the same search space, and that the viewpoints of Scenario 1 suffice for good localization.

By comparing Scenarios 1, 2 with Scenario 3 (all three of which use the greedy next-view-planner), we reach some conclusions as to how a more complex scene (Scenario 3 has a significantly greater search space and a significantly larger candidate hypotheses list than the other two scenarios) affects the performance of the localization algorithm. From Fig.8 and the relevant tables in the supplementary material section of the journal, we notice only a small change in the median target rank and the average number of test runs that do not contain a maximal target image (*i.e.*, the test runs marked with an unknown target rank U). As expected, there is a slight degradation of the results' quality in Scenario 3 due to the increased search space size, but this performance decrease is not sufficient to indicate that the algorithm does not scale well. From Table I we notice an interesting phenomenon. While for Scenario 3 there is a noticeable increase in the number of executed hypotheses —compared to Scenarios 1, 2— until the target is first localized, the increase in the total distance covered until the target is first localized is not quite as large. This implies that the average distance covered for each executed hypothesis until the target is first localized is smaller in Scenario 3. This likely occurs because the volume covered by the two shelves is quite close to the table's volume, and the greedy algorithm tends to make smaller steps by switching between searching the shelf space and the table space in order to decrease the total distance covered. We would expect the optimal solution to have a constant ratio for the number of executed hypotheses to the distance covered across Scenarios 1,2,3, if the shelves were far away from the table. This shows that while the greedy next-view-planner is not guaranteed to be optimal, its performance is far better than that of a typical baseline next-view-planner. We investigate this in more detail with Scenarios 4, 5 below. This shows that the greedy next-view-planner does manage to constrain the total distance covered while maintaining an acceptable recognition performance.

From Table I and Figs.6,7,8, we can compare the performance of Scenario 3 vs. baseline Scenarios 4 (constant cost function) and 5 (randomized cost). From Fig.6 we observe that the greedy algorithm covers on average significantly smaller distances for each executed hypothesis than the other two scenarios, while localizing the targets as reliably as Scenario 4 and significantly more reliably than Scenario 5 (notice the explosion of U labelled test runs in Fig.8(e)). Furthermore, we notice in Fig.6 that the greedy next-view-planner and the constant cost planner distances start to decrease roughly after hypothesis 13. This is likely due to the greater certainty as to the location of the target —where the target is and is not located—, causing HR to cover smaller distances on average. Notice that the random next-view-planner's distances are constant and do not tend to decrease as the number of executed hypotheses increases. In Fig.7 we notice that the greedy next-view-planner results in a significantly smaller target map entropy after executing each hypothesis. A somewhat surprising result is that the greedy next-view-planner also leads to a lower target map entropy than the constant cost next-view-planner. Since the constant cost next-view-planner ignores the movement costs and simply looks at the next most probable location of the object, one would think that Scenario 4 (which uses a constant cost function) would result in covering longer distances than Scenario 3, but with a faster decreasing entropy. However, as we see in Fig.7 this is not the case. This seems to occur because the constant cost function executes on average hypotheses that cover greater distances (Fig.6). This results in a greater number of small patches of never-viewed search regions, which retain their uniform prior and which accumulate over time and lead to a greater overall entropy. Notice that the entropies in Fig.7 tend to converge to a non-zero horizontal asymptote. This is due to big regions in our search space that are never viewed by HR, specifically regions under the table. This, however, does not in any way affect the next-view-planner's decisions, as over time, an obstacle map is built around these regions and HR does not sum over those regions' probabilities when choosing where to look next (see Eq.(2)). Overall, the results of Scenarios 3, 4, 5 have justified the use of the greedy next-view-planner as an efficient approximation to the optimal next-view-planner. Just as the greedy approximation to the Knapsack problem offers an efficient and often optimal solution to the problem [38], so does the greedy next-view-planner offer an efficient solution to the problem that performs better than the baseline cases. From Table I we see that the distance covered until the target is first localized (i.e., all cases excluding $d \leq U$) does not depend on the target's recognition certainty (i.e., it does not depend on which of the three cases $d = 1$, $d \leq 3$, $d < U$ we are dealing with). If we include test runs when HR does not localize the target ($d \leq U$) we end up with greater values.

We notice in Fig.8 that the distributions are bimodal, clustered around a rank of 1 and a rank of U. We view this as an indication that the likelihood of localization due to chance is trivial in our results, because if that were the case, we would expect to see a more uniform spread in the distributions. We notice in Fig.8(f) that the proportion of U-ranked test runs is significantly lower than it is in any of the other five sub-figures.

This implies that by increasing the number of viewpoints from which a scene is examined, and without applying any sort of improvements to the single-view recognition algorithm, the presented algorithm can significantly increase the true positive and true negative rates. Notice that this improvement occurs regardless of the next-view-planner used, as it is easily verified from Fig.8 or by comparing the target ranks and global ranks in the supplementary material section of the paper. This shows that without striving for major improvements in single-view recognition, improvements to the next-view-planner can lead to significantly better results. In other words, the importance of intelligent search algorithms should not be trivialized, and the importance of avoiding degenerate viewpoints [26] should not be underestimated either. While we have shown that our greedy next-view algorithm is superior in many ways to other baseline algorithms, the next-view-planning problem is, in our opinion, far from optimally solved, as it is also argued in [10].

The work described in this paper constitutes the first active visual search algorithm ever implemented on a humanoid robot developed by Honda [35]. Compared to much of the related work described in the introduction, our work is purely vision based and does not use other types of sensors such as range finders. This follows the premise around which Honda's humanoid robot project [35] is structured, of building robotic systems that emulate human locomotion and the human visual system, both in terms of the hardware used (*e.g.,* using a visually guided humanoid robot) and the software architecture used (*e.g.,* using a hierarchical feedforward recognition system inspired by human vision, and a next-view-planner that shares a number of behavioural properties with an ideal searcher), thus, constituting one of the most advanced neuromorphic systems currently described in the literature for performing visual search. In related work, such as [17], [18], non-vision based SLAM techniques are often used for the map building and self-localization problem. Such techniques are typically superior than vision based algorithms are, especially in poorly textured environments. In the presented work, the problem of self-localization is circumvented due to HR's good dead-reckoning. However, vision-based SLAM techniques, or landmark localization techniques, will have to be applied in future work to make HR capable of searching vastly larger spaces. The presented optimization algorithm evaluates all candidate hypotheses when deciding where to move/look next. Thus, as with all exhaustive search algorithms, it does not easily fall in local minima. However, it does not scale as well as gradient-descent-like optimization or linear-programming-based approaches do. In contrast to POMDPs which use an infinite time horizon, our optimization algorithm uses a one-step look-ahead, which suffices for certain vision tasks. We did not incorporate an error model in the disparity measurements, since our use of Marked Candidate Cells around each detected scene obstacle was proven sufficient in practice to handle the effects of small depth estimation errors on the target map's updating. Furthermore, the lack of an error model speeds up our algorithm significantly, making real-time performance easier to achieve. As we discovered in practice, achieving near real-time performance is a non-trivial task, and depends on many problem parameters, such as the search space size. The next-view-planner described does not forbid more complex motions, such as squatting, from taking place. Executing such motions is a matter of having appropriate inverse kinematics libraries that can position the sensor in the desired state.

VI. Conclusions

We have shown that fast and reliable 3D object localization is feasible if we place some reasonable constraints on the problem. Such constraints include placing bounds on the size of the search space, having controlled illumination conditions, having small dead-reckoning errors, and limiting the search to objects that are well recognized by the feedforward hierarchy. We have discussed the intractability of the localization problem. We have shown that a greedy approximation to the constrained active localization problem, that is based on the greedy approximation to the Knapsack problem, can perform better in terms of localization speed than random search and search that ignores the search movement costs. Furthermore, the greedy next-view-planner does not lead to a decrease in the reliability of the localization. We briefly discussed the trade-offs of localizing vs. detecting a target object. We used these results as motivation to show that even without perfect dead-reckoning, it is possible to localize the position of an object accurately enough to perform a number of tasks. Future work may include using HR to grasp the object once it has been localized, using voice commands to provide feedback to HR and make the search more interactive, and to deal with dynamic environments changing over time. Future work can also include an extensive analysis of the effects on the results of other parameters (camera resolution, depth-of-field, and search space dimensions for example) both qualitatively and quantitatively. Performing a cascade of experiments with scenarios where the difficulty in localizing the object is progressively increased (through an increase in the degree of object occlusion, or an increase in the objects' similarity for example), could provide more insights on the system's limitations and on ways to improve its performance.

Appendix

A. Addendum to Sec.II-C

Proof of Theorem 1

Proof: Notice that $\sum_j p(c_j^t, \lambda_n | v_n, \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) = \sum_j p(\lambda_n | c_j^t, v_n) p(c_j^t | \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1)$. Thus $p(c_i^t | \lambda_n, v_n, ..., \lambda_1, v_1) = \frac{p(c_i^t | \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) p(\lambda_n | c_i^t, v_n)}{\sum_j p(c_j^t | \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) p(\lambda_n | c_j^t, v_n)}$ iff $p(c_i^t | \lambda_n, v_n, ..., \lambda_1, v_1) = \frac{p(c_i^t | \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) p(\lambda_n | c_i^t, v_n)}{\sum_j p(c_j^t, \lambda_n | v_n, \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1)}$. But this last equation holds iff $p(c_i^t | \lambda_n, v_n, ..., \lambda_1, v_1) p(\lambda_n | v_n, \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) = p(c_i^t | \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) p(\lambda_n | c_i^t, v_n)$ which in turn holds iff $p(\lambda_n | c_i^t, v_n, \lambda_{n-1}, v_{n-1}, ..., \lambda_1, v_1) = p(\lambda_n | c_i^t, v_n)$, which holds by assumption. ■

Theorem 2 Preliminaries

Assume $X_{[D,1]} \in [D, 1]$, $Y_{[0,D)} \in [0, D)$ are unknown functions that depend on $\vec{q}, \mu_v, v, s, t, D \in (0, 1)$ and represent the values of $\mathbf{CM}(\vec{q}; \mu_v, v, s, t)$ in image areas containing the target ($X_{[D,1]}$) or background ($Y_{[0,D)}$) respectively (see

Fig.3(b)). Thus, for any sample function $\mathbf{CM}(\cdot;\mu_v,v,s,t)$ that is returned by random variable $\mathbf{CM}(v,s,t)$, its pixel $\vec{q}$ satisfies

$$\mathbf{CM}(\vec{q};\mu_v,v,s,t) = \begin{cases} X_{[D,1]} & \text{if target } t \text{ is sensed by } \mu_v \text{ and projects at} \\ & \text{intrinsic scale } s \text{ and encompasses } \vec{q} \\ Y_{[0,D)} & \text{otherwise} \end{cases} \qquad (11)$$

where $D \in (0,1)$ is also an unknown function of μ_v, v, s, t and denotes a threshold that separates the confidence map values between those corresponding to the localized object and the background for sensor output μ_v. Given only v, s and t, μ_v is unknown (the image data of μ_v is a random sample from $\Upsilon(v)$), and we can thus view D as a random variable.

Theorem 2. (Ideal Monotonicity) *As it becomes more likely that a target confidence map value represents the presence of the target object, the probability of observing at least that small of a value in the confidence map, decreases:* $\forall v_0, s_0, t_0$ *and* $\forall i_1, i_2$, $0 \leq i_1, i_2 \leq 1$, *we have* $p(i_1 \geq D) \leq p(i_2 \geq D)$ *if and only if* $p([i_1,1] \in \mathbf{CM}(v_0,s_0,t_0)) \geq p([i_2,1] \in \mathbf{CM}(v_0,s_0,t_0))$ *where* $p([i',1] \in \mathbf{CM}(v_0,s_0,t_0)) \triangleq \int_{i'}^{1} \int p(\mathbf{CM}(\,\cdot\,;\mu_{v_0},v_0,s_0,t_0) = i) dp_{v_0,s_0,t_0} di$, *with the inner-most integral denoting the Lebesgue integral of* $p(\mathbf{CM}(\cdot;\mu_{v_0},v_0,s_0,t_0) = i)$ *over* $\Upsilon(v_0,s_0,t_0)$ *and* $p(\mathbf{CM}(\cdot;\mu_{v_0},v_0,s_0,t_0) = i)$ *denoting a density function of* i*, for the pixel-values/firing-rates* i *contained in* $\mathbf{CM}(\cdot;\mu_{v_0},v_0,s_0,t_0)$.

Proof: Let $f_D(i)$ represent the density function of D and $f_{XY}(i) = \int p(\mathbf{CM}(\,\cdot\,;\mu_{v_0},v_0,s_0,t_0) = i) dp_{v_0,s_0,t_0} = p(i \in \mathbf{CM}(\,\cdot\,;v_0,s_0,t_0))$. Then $p(i_1 \geq D) \leq p(i_2 \geq D) \Leftrightarrow \int_0^{i_1} f_D(i)di \leq \int_0^{i_2} f_D(i)di \Leftrightarrow i_1 \leq i_2 \Leftrightarrow \int_{i_1}^{1} f_{XY}(i)di \geq \int_{i_2}^{1} f_{XY}(i)di$ which proves the theorem. ■

B. Addendum to Sec.II-D

If $topN(\mu_v,v,t,i) \geq eer(proj_1(i,v,t),t)$ then $p_{topN}(\mu_v,v,t,i) \triangleq 0.5$. If $topN(\mu_v,v,t,i) < eer(proj_1(i,v,t),t)$, we use a linear mapping of $topN(\cdot)$: $p_{topN}(\mu_v,v,t,i) \triangleq \frac{topN(\mu_v,v,t,i) - bottom(\mu_v,v,t,i)}{2(eer(proj_1(i,v,t),t) - bottom(\mu_v,v,t,i))}$. If $top(\mu_v,v,t,i) \geq eer(proj_1(i,v,t),t)$, $p_{top}(\mu_v,v,t,i) \triangleq \frac{1}{2} + \frac{top(\mu_v,v,t,i) - eer(proj_1(i,v,t),t)}{2(1 - eer(proj_1(i,v,t),t))}$. If $top(\mu_v,v,t,i) < eer(proj_1(i,v,t),t)$, $p_{top}(\mu_v,v,t,i) \triangleq \frac{top(\mu_v,v,t,i) - bottom(\mu_v,v,t,i)}{2(eer(proj_1(i,v,t),t) - bottom(\mu_v,v,t,i))}$. Finally, we map $bottom(\mu_v,v,t,i)$ to probability $p_{bottom}(\mu_v,v,t,i)) \triangleq 0$.

To calculate the parameters of the models $\frac{\alpha_j}{p(\beta_i)} + \gamma_j$ (where $j \in \{1,2\}$), we see that a simple LU-decomposition provides the solution for $p(\beta_i) \in [p(c_i^t), \beta_{topN}(\mu_v,v,t,i)]$:

$$\alpha_1 = p(c_i^t)\frac{p_{top}(\cdot)\beta_{topN}(\cdot) - p_{topN}(\cdot)p(c_i^t)}{\beta_{topN}(\cdot) - p(c_i^t)} \qquad (12)$$

$$\gamma_1 = p(c_i^t)\frac{p_{topN}(\cdot) - p_{top}(\cdot)}{\beta_{topN}(\cdot) - p(c_i^t)} \qquad (13)$$

and for $p(\beta_i) \in [\beta_{topN}(\mu_v,v,t,i), 1]$:

$$\alpha_2 = p(c_i^t)\frac{p_{topN}(\cdot) - p_{bottom}(\cdot)\beta_{topN}(\cdot)}{1 - \beta_{topN}(\cdot)} \qquad (14)$$

$$\gamma_2 = p(c_i^t)\frac{p_{bottom}(\cdot) - p_{topN}(\cdot)}{1 - \beta_{topN}(\cdot)}. \qquad (15)$$

Acknowledgment

A.A. would like to thank Bram Bolder, Mark Dunn, Michael Gienger, Stephan Kirstein and Inna Mikhailova of the Honda Research Institute Europe, and Dr. Jochen J. Steil, Bielefeld University, for their help. We also thank the anonymous reviewers for their insightful comments and suggestions.

References

[1] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman and Company, 1982.

[2] J. Tsotsos, Y. Liu, J. Martinez-Trujillo, M. Pomplun, E. Simine, and K. Zhou, "Attending to visual motion," *Comput. Vis. Image Und.*, vol. 100, no. 1-2, pp. 3–40, October 2005.

[3] R. Bajcsy, "Active perception vs. passive perception," in *IEEE Workshop on Computer Vision Representation and Control*, Bellaire, 1985.

[4] J. K. Tsotsos, "On the relative complexity of active vs. passive visual search," *Int. J. Comput. Vision*, vol. 7, no. 2, pp. 127–141, 1992.

[5] J. K. Tsotsos and K. Shubina, "Attention and visual search: Active robotic vision systems that search," in *Proc. 5th International Conference on Computer Vision Systems*, 2007.

[6] S. Dickinson, D. Wilkes, and J. Tsotsos, "A computational model of view degeneracy," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 21, no. 8, pp. 673–689, August 1999.

[7] J. K. Tsotsos, "Analyzing vision at the complexity level," *Behav. Brain Sci.*, vol. 13-3, pp. 423–445, 1990.

[8] L. Itti, G. Rees, and J. Tsotsos, Eds., *Neurobiology of Attention*. Elsevier/Academic Press, 2005.

[9] Y. Ye and J. Tsotsos, "Sensor planning for 3D object search," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 145–168, 1999.

[10] A. Andreopoulos and J. K. Tsotsos, "A theory of active object localization," in *Proc. Int. Conf. on Computer Vision*, 2009.

[11] A. Ceravola, F. Joublin, M. Dunn, J. Eggert, M. Stein, and C. Goerick, "Integrated research and development environment for real-time distributed embodied intelligent systems," in *Proc. Intelligent Robots and Systems*, 2006, pp. 1631–1637.

[12] T. Garvey, "Perceptual strategies for purposive vision," Technical Note 117, SRI Int'l., Tech. Rep., September 1976.

[13] L. E. Wixson and D. H. Ballard, "Using intermediate objects to improve the efficiency of visual search," *Int. J. Comput. Vision*, vol. 12, no. 2/3, pp. 209–230, 1994.

[14] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 15, no. 5, 1993.

[15] R. D. Rimey and C. M. Brown, "Control of selective perception using bayes nets and decision theory," *Int. J. Comput. Vision*, vol. 12, no. 2/3, pp. 173–207, 1994.

[16] G. Giefing, H. Janssen, and H. Mallot, "Saccadic object recognition with an active vision system," in *Proc. ICPR*, 1992.

[17] S. Ekvall, P. Jensfelt, and D. Kragic, "Integrating active mobile robot object recognition and SLAM in natural environments," in *Proc. Intelligent Robots and Systems*, 2006.

[18] D. Meger, P. Forssen, K. Lai, S. Helmer, S. McCann, T. Southey, M. Baumann, J. Little, and D. Lowe, "Curious George: An attentive semantic robot," in *Proc. Robot. Auton. Syst.*, 2008.

[19] P. Forssen, D. Meger, K. Lai, S. Helmer, J. Little, and D. Lowe, "Informed visual search: Combining attention and object recognition," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2008.

[20] F. Saidi, O. Stasse, K. Yokoi, and F. Kanehiro, "Online object search with a humanoid robot," in *Proc. Intelligent Robots and Systems*, 2007.

[21] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. Int. Conf. on Computer Vision*, 1999.

[22] D. Wilkes and J. Tsotsos, "Behaviours for active object recognition," in *SPIE Conference*, September 1993, pp. 225–239.

[23] F. Callari and F. Ferrie, "Active recognition: Using uncertainty to reduce ambiguity," in *Proc. ICPR*, 1996.

[24] ——, "Active recognition: Looking for differences," *Int. J. of Comput. Vision*, vol. 43, no. 3, pp. 189–204, 2001.

[25] C. Laporte and T. Arbel, "Efficient discriminant viewpoint selection for active bayesian recognition," *International Journal of Computer Vision*, vol. 68, no. 3, pp. 267–287, 2006.

[26] S. Dickinson, H. Christensen, J. Tsotsos, and G. Olofsson, "Active object recognition integrating attention and viewpoint control," *Comput. Vis. and Image Und.*, vol. 67, no. 3, pp. 239–260, 1997.

[27] B. Schiele and J. Crowley, "Transinformation for active object recognition," in *Proc. Int. Conf. on Computer Vision*, 1998.

[28] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz, "Active object recognition in parametric eigenspace," in *Proc. British Machine Vision Conference*, 1998, pp. 629–638.

[29] T. Foissotte, O. Stasse, A. Escande, and A. Kheddar, "A next-best-view algorithm for autonomous 3d object modeling by a humanoid robot," in *Proc. IEEE-RAS International Conference on Humanoid Robots*, 2008.

[30] S. D. Roy, S. Chaudhury, and S. Banerjee, "Recognizing large 3D objects through next view planning using an uncalibrated camera," in *Proc. Int. Conf. on Computer Vision*, 2001.

[31] S. D. Roy and N. Kulkarni, "Active 3D object recognition using appearance based aspect graphs," in *Proc. ICVGIP*, 2004, pp. 40–45.

[32] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[33] V. A. Sujan and S. Dubowski, "Efficient information-based visual robotic mapping in unstructured environments," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 275–293, April 2005.

[34] N. Roy, G. Gordon, and S. Thrun, "Finding approximate POMDP solutions through belief compression," *Journal of Artificial Intelligence Research*, vol. 23, pp. 1–40, 2005.

[35] "The Honda Humanoid Robot ASIMO." [Online]. Available: http://www.world.honda.com/ASIMO

[36] H. Wersing and E. Körner, "Learning optimized features for hierarchical models of invariant object recognition," *Neural Computation*, vol. 15, no. 7, pp. 1559–1588, 2003.

[37] H. Stark and J. W. Woods, *Probability and Random Processes with Applications to Signal Processing*. Prentice Hall, 2001.

[38] M. R. Garey and D. S. Johnson, *Computers and Intractability: A guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[39] J. Najemnik and W. S. Geisler, "Optimal eye movement strategies in visual search," *Nature*, vol. 434, pp. 387–391, 2005.

[40] H. Wersing, S. Kirstein, M. Goetting, H. Brandl, M. Dunn, I. Mikhailova, C. Goerick, J. J. Steil, H. Ritter, and E. Koerner, "Online learning of objects in a biologically motivated visual architecture," *International Journal of Neural Systems*, vol. 17, no. 4, pp. 219–230, 2007.

[41] C. Goerick, I. Mikhailova, H. Wersing, and S. Kirstein, "Biologically motivated visual behaviours for humanoids: Learning to interact and learning in interaction," in *Proc. IEEE/RSJ Int. Conf. on Humanoid Robots*, 2006.

[42] "Small Vision System, SRI International." [Online]. Available: http://www.ai.sri.com/software/SVS

[43] M. Gienger, H. Janssen, and C. Goerick, "Task-oriented whole body motion for humanoid robots," in *Proc. IEEE/RAS Int. Conf. on Humanoid Robots*, 2005, pp. 238–244.

Alexander Andreopoulos received an Honours B.Sc. degree (2003) in Computer Science and Mathematics, with High Distinction, from the University of Toronto. In 2005 he received his M.Sc. degree in Computer Science from York University, where he is currently a Ph.D. candidate in Computer Science. His research interests include active vision, visually guided robotics and medical imaging. He has received the DEC award for the most outstanding student in Computer Science to graduate from the University of Toronto, a SONY science scholarship, NSERC PGS-M/PGS-D scholarships and a best paper award.

Stephan Hasler received the Diploma degree in computer engineering from the Technical University of Ilmenau, Germany, in 2004. Currently he is with the Honda Research Institute Europe GmbH, Offenbach, Germany, where he pursues his Ph.D. degree in collaboration with Bielefeld University, Germany. His research interests are in object representation, feature extraction, and learning systems.

Heiko Wersing received the Diploma in Physics in 1996 from Bielefeld University, Germany. In 2000, he received his Ph.D. in science from the Faculty of Technology, Bielefeld University. In 2000, he became a member of the Future Technology Research Group of Honda R&D Europe, GmbH, Offenbach, Germany. Since 2003, he holds a position as a principal scientist in the Honda Research Institute Europe, at Offenbach. Since 2007, he is also co-speaker of the graduate school of the CoR-Lab Research Institute for Cognition and Robotics, Bielefeld University. His current research interests include recurrent neural networks, models of perceptual grouping and feature binding, principles of sparse coding, biologically motivated object recognition and online learning.

Herbert Janßen received his Diploma degree in Physics from the Westphalian Wilhelms University of Muenster, Germany in 1989. From 1989 to 1999 he was researcher at the Institute for Neuroinformatics, University of Bochum, Germany. From 1999 to 2004 he worked in the Honda R&D robot lab in Wakou, Japan. Since 2004 he works as a Principal Scientist for the Honda Research Institute Europe, Germany.

John K. Tsotsos received his Ph.D. in 1980 from the University of Toronto. He was on the faculty of Computer Science at the University of Toronto from 1980 to 1999. He then moved to York University appointed as Director of York's Centre for Vision Research (2000-2006) and is currently Distinguished Research Professor of Vision Science in the Dept. of Computer Science & Engineering. He is Adjunct Professor in both Ophthalmology and Computer Science at the University of Toronto. Dr. Tsotsos has published many scientific papers, six conference papers receiving recognition. He currently holds the NSERC Tier I Canada Research Chair in Computational Vision. He has served on the editorial boards of Image & Vision Computing, Computer Vision and Image Understanding, Computational Intelligence and Artificial Intelligence and Medicine and on many conference committees. He served as General Chair for IEEE International Conference on Computer Vision 1999.

Edgar Körner studied electrical engineering, control engineering, and biomedical cybernetics at the Ilmenau Institute of Technology, Germany. He received his Dr.-Ing. in biomedical cybernetics in 1977 and the Dr. Sci. in biocybernetics in 1984, both from Ilmenau Institute of Technology. From 1984 to 1987, he joined the Bioholonics Project of JRDC (Tokyo) as a research fellow dealing with brain-like vision systems. Back at the Ilmenau Institute of Technology, he continued research in biological vision and neurofuzzy control systems as an associate professor. In 1988, Dr. Körner was appointed full professor for biocybernetics and head of the Department of Neurocomputing and Cognitive Systems. In 1992 he moved to Japan to join Honda R&Ds Wako Research Center near Tokyo, focusing as a chief scientist on the brain-like computation research. In 1997 he started research in computational neuroscience, evolutionary technology, and cognitive robotics at Honda R&D Europe, where he served as an executive vice president and head of the Future Technology Research Divison. Since 2003, Dr. Körner serves as the president of the Honda Research Institute Europe GmbH. Since October 2007, he additionally serves as a co-director of the Research Institute for Cognition and Robotics at the University Bielefeld. His research focus is on brain-like artificial neural systems and self organization of knowledge representation for autonomous robots.