Context-Aware Search using Cooperative Agents in a Smart Environment

Dan Xie Roderic A. Grupen Allen Hanson
Department of Computer Science, University of Massachusetts Amherst
Amherst, MA 01003, USA
{dxie,grupen,hanson}@cs.umass.edu

Abstract

In this paper we present the design of a decentralized vision-based object search system that can be used for elder care in a smart environment. In our approach, each autonomous search agent maintains separate estimates of the probability density function (PDF) of the object location and makes independent decisions about its search process. Asynchronous cooperative search is achieved by transmitting perceptual information among the agents. Our work also investigates how context such as the detection history and density of activity by people influence the estimation of the prior PDF of the target and the use of this information to improve the search efficiency. Our experimental results demonstrate that the proposed cooperative search strategy is efficient and the methods we use to incorporate contextual information into the target's posterior PDF can improve the efficiency further.

1. Introduction

Health care for the elderly poses a major challenge as the baby boomer generation ages. Part of the solution is to develop technology using sensor networks and service robotics to increase the length of time that an elder can remain at home. In addition to monitoring for illnesses and potentially life-threatening situations, an equally important challenge in in-home elderly care is providing assistance in their day-to-day life. Since moderate immobility and memory impairment are common as people age, a major problem for the elderly is locating and retrieving frequently used "common" objects such as keys, cellphone, books, etc. Therefore, it is important to develop effective and efficient approaches for automated in-home object search.

Heuristic strategies in target search. There has been considerable recent interest in addressing the problem of "target search". Bourgault et al. [3] proposed a Bayesian approach to the problem of target search by a single autonomous sensor platform. Ye et al. [10] formulated target search as an optimization problem where the goal is to maximize the probability of detecting the target within a given time constraint. Since planning such search activity is NP-complete, some heuristic strategies were proposed that often lead to practical solutions. Wixon et al. [9] use the idea of indirect search, in which one first finds an object that typically has a spatial relationship to the target, and then restricts the search in the spatial area defined by that relationship. Sujan [8] proposes an iterative planning approach driven by an evaluation function based on Shannon's information theory. The camera parameter space is explored and each configuration is evaluated according to the evaluation function. The concept of a visibility map is introduced in [6, 7] to constrain the sensor parameter space according to the detection characteristics of the recognition algorithm. These techniques reduce the dimension of the sensor parameter space.

Cooperative search strategies. In search operations, a team of intelligent agents can provide a robust solution with greater efficiency than can be achieved by single agents, even with comparatively superior mobility and sensors. The key is to develop a cooperative decentralized control strategy that allows each agent to determine its actions independently while optimizing the team's performance. A synchronized coordinated search strategy was developed in a Bayesian framework in [2]. DeLima et al. [4] proposed a rule-based search method with which multiple unmanned aerial vehicles can cooperatively search an area for mobile target detection.

The approach proposed in this paper is related to both the aspects of cooperative strategy and heuristic solution. We use multiple Pan/Tilt/Zoom (PTZ) camera nodes as the search agent to explore a specified area in a living space and focus on developing an efficient cooperative search strategy. In the design of our smart environment, the camera nodes are supposed to perform multiple tasks such as object search, people tracking, and are controlled by a resource management unit. In this framework, the search process of an agent can be interrupted by other tasks with higher priority. This problem, along with the possible node and network failures, recommends a decentralized search framework. This paper presents an asynchronous cooperative search strategy in which autonomous search agents share perceptual information, but maintain separate target PDF estimates and make independent decisions about their search strategy.

People interact with objects in the course of many tasks associated with daily living. A novel idea in this paper is leveraging user activity to improve the cost efficiency in search tasks. User activity density can be analyzed from the vision-based people tracker, and can be used to infer the region where object use may happen. Activity density is valuable information for increasing the accuracy and efficiency of object search. To the best of our knowledge, no work has been done that uses human activity information to help reduce the search space of the robot agent. In this paper we also investigate how to aggregate previous search results to estimate the prior PDF of the target object.

The remainder of this paper is structured as follows. Section 2 presents the local search using a single agent. Section 3 presents the cooperative search strategy. In Section 4 we discuss the factors that influence the prior PDF of the target and how to use it to expedite the search process. The experimental results are presented in Section 5. Conclusions and future work are given in Section 6.

2. Local Search with Single Agent

To achieve decentralized search, each search agent should be able to independently perform a complete exploration of the environment, acquire observations and decide if the target is detected in a certain region.

2.1. Bayesian Searching Problem

The search problem for a single agent can be represented in a Bayesian framework [3]. For a target r, the state vector of its location $\vec{x}_r \in \mathcal{X}_r$ in Cartesian space can be expressed by a probability density function (PDF) $p_r(\vec{x})$. A prior PDF $p_r(\vec{x}_0|z_0) = p_r(\vec{x}_0)$ is the representation of the prior knowledge of where the target object is before the search. Given a prior PDF and the independent observations z, the PDF of a target at time step t can be constructed recursively using Bayes' theorem. In the application of object search to living space, it is reasonable to assume that when the search process starts, the target object is stationary and not allowed to move until the search finishes. So we have $p_r(\vec{x}_t|z_{1:t-1}) = p_r(\vec{x}_{t-1}|z_{1:t-1})$. After each observation, the PDF will be updated according to the observation,

$$p_r(\vec{x}_t|z_{1:t}) = Kp_r(\vec{x}_{t-1}|z_{1:t-1}) \cdot p_r(z_t|\vec{x}_t) \tag{1}$$

Where K is the normalization factor and is given by,

$$K = 1/\int [p_r(\vec{x}_t|z_{1:t-1})p_r(z_t|\vec{x}_t)]d\vec{x}_t \tag{2}$$

The search system is designed to maximize the chances of finding the queried object in a restricted amount of time. Although using a longer time horizon can achieve a better solution, planning with a "one-step-lookahead" strategy [3, 10] that maximizes the probability of detecting the target object at time t given the observation sequence $z_1 ... z_{t-1}$ can provide reasonable performance with very low computational overhead.

2.2. Local Search Strategy

The local search task consists of three subtasks. The first subtask is the selection of the next action and the corresponding PTZ parameters so as to bring a potential search subarea into the field of view of the camera. The second is to control the hardware to realize the planned state. The third subtask involves detecting the target within the image.

We assume that the geometric configuration of the search space is $\mathcal{C}_W$, and objects can be placed only on the floor or on tables (with the same height). So the horizontal planes of the search region are tessellated into a two-layer grid $\mathcal{G}$ where the centers of the grid nodes $g_i \in \mathcal{G}$ are candidate positions to be observed.

Given $\mathcal{C}_W$, each agent c calculates a local visibility map $M_c(\vec{x}) = \{0, 1\}$ which indicates if a grid node g_i is visible to the camera or not (g_i is located outside of the limitation of the Pan/Tilt/Zoom parameters or is blocked by an occluding object). Then the local PDF map $p_{c,r}(\vec{x}_0)$ for agent r can be initialized by re-normalizing $p_r(\vec{x_0})$ in all visible areas.

$$p_{c,r}(\vec{x}_0) = Np_r(\vec{x}_0) \cdot M_c(\vec{x}) \tag{3}$$

where N is a normalization factor. The local PDF map is the core data maintained by each agent. After $p_{c,r}(\vec{x_0})$ is calculated, the agent is ready to perform search. The overall local search process is illustrated in Figure 2 (without message transmission). The agent repeats the subtasks of action planning, manipulation, and observation to explore all the visible area.

In our approach each agent uses a "one-step-lookahead" strategy to plan the next action. The action space of an agent consists of all the manipulations that bring a visible grid node to the center of the camera image. To do action planning, a value v_i is calculated for each grid node indicating the benefit of visiting this node. Given the local PDF of the target, v_i can be calculated by,

$$v_i = \sum_{g_k \in \mathcal{V}_i} p_{c,r}(x_k) \tag{4}$$

where $\mathcal{V}_i$ is the set of all grid nodes in the *observation field*, i.e., that can be observed when agent is visiting grid node g_i. As shown in Figure 1 (a), an observation field is computed by projecting a bounding box in the camera image to the Cartesian space. The bounding box is shortened by approximately 50 pixels on each side to reduce lens curvature effects. The action is selected to visit the grid node with highest value v_i.

In the manipulation step, the Pan/Tile/Zoom parameters $(\alpha, \theta, \tau) = f(\vec{x}_{g_i})$ are calculated according to the 3D position of the selected grid node and the camera. To get similar observation fields, the zoom value τ is proportional to the distance between the visited grid node and the camera.

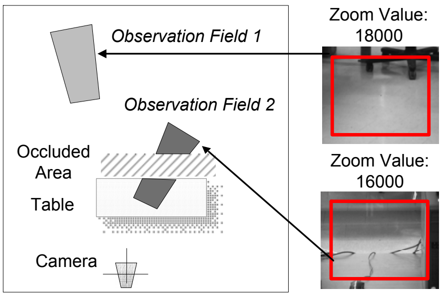


Figure 1. Observation field of camera agents.

Object detection in the image is achieved using mean shift with color features [1]. The targets can be commonly used objects in the home such as keys, cellphones, cups and books. If a potential object is detected in the image, the camera will bring it to the center and zoom in further to achieve a more accurate detection result. Therefore the time cost of the observation step is variable, and an asynchronous mechanism is necessary when we are designing the cooperative search strategy. The observation result is integrated to the local PDF using equation 1.

The search process terminates if the agent detects the target or completes the exploration of all its action space.

3. Cooperative Search Strategy

In this section we focus on how agents achieve cooperative search through communication. Considering that local search processes can be interrupted and can take different amount of time per observation, a decentralized and asynchronous cooperative search strategy is necessary for our application.

Comparing to methods in which each agent maintains identical PDFs [2], in the proposed search strategy, cooperative agents maintain separate search knowledge concerning the probable observability of the target and make independent decisions about their search process. Based on the local search process described in the previous section, cooperative search can be achieved by transmitting messages. There are two kinds of messages to be considered.

(1) *Pre-Observation Inhibition Message (POIM)*. Since the action sequence of the PTZ camera agent is discontinuous, the problem of collision (overlapping observation) avoidance is non-trivial. As illustrated in Figure 2, POIM is

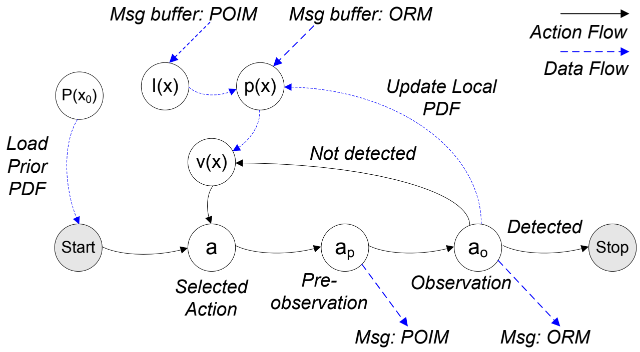


Figure 2. The cooperative search strategy for a single agent.

broadcast by an agent after it selects its next action, but before the control and observation steps are executed. POIM contains an inhibition map $I(\vec{x})$ that masks all the grid nodes to be 0 in the observation field of the selected next action. The agent receiving a POIM will combine it with its current local PDF to avoid overlapping observation. As illustrated in Figure 3, in time $t+2$, agent 2 receives the POIM from agent 1. The area that agent 1 is planning to observe is ablated in the local PDF of agent 2, which in turn plans its next action to cover the peak on the right side.

(2) *Observation Result Message (ORM)*. ORM is broadcast after an action is taken. It contains the observation result $p_r(z_t|x_t)$ produced by agent r at time t. The agent receiving this message uses it to update its current local PDF using Bayes' rule as shown in equation 1. The observation result from another agent is considered to be equivalent to the result obtained by the agent itself. As illustrated in Figure 3, the local PDF of target 1 at time $t+10$ was modified by the previous observation taken by itself, the observation results sent by agent 2, and the POIM message sent by agent 2 at time $t+5$.

Some techniques have to be applied to maintain the correctness of message transmission. First, message buffers are used to store the received POIM and ORM. Second, sometimes ORM_t^r arrives later than $POIM_t^r$ due to network latency. The agent has to detect this inconsistency by comparing the timestamp of the messages and discard ORM_t^r. In addition, the agent will discard the POIM if the corresponding ORM hasn't come for a time threshold T_{POIM}. Therefore if the agent sending the POIM fails or is interrupted by another task, the area masked by POIM will eventually be observed by other agents.

The termination criterion of the cooperative search depends on the requirement of the confidence of target detection. In our system as soon as two agents find the target, all the agents will stop. The 3D location of the target is then triangulated and reported.

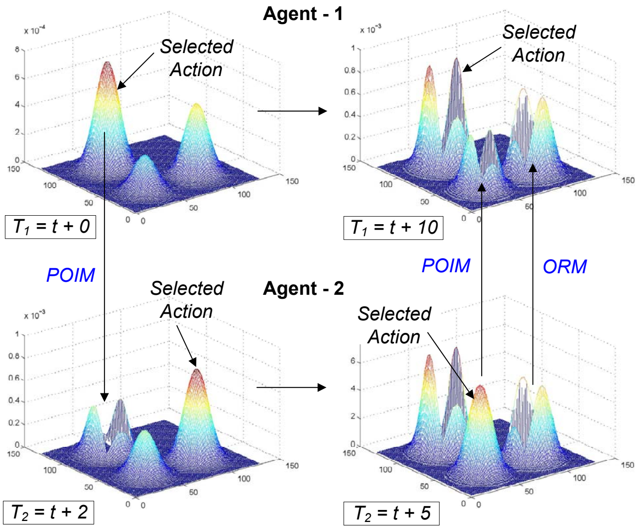


Figure 3. Local PDF update for search agents.

4. Factors That Influence the Prior PDF

Efficiency is crucial for each object search system. A brute force search approach may suffice for the solution, but will be computationally prohibitive for non-trivial situations. In the application of living space search, the probabilistic distribution of objects can be influenced by many factors. In this section we investigate the influence of two salient factors, (i) the *detection history* that represents the number of times that an object has been detected in a specific location, and (ii) the *activity density* of the people in the environment.

4.1. Detection History

The aggregation of previous search and detection results provides important cues to guide the search behavior. We can use data accumulation over an extended period to produce a prior probabilistic distribution reflecting the target's preferred location in the environment. The influence of each detection result depends on the triangulation quality of target detection.

After the target is detected, its location will be triangulated by a camera pair. The influence of this detection on the prior PDF is determined by the Cartesian observation error ellipsoid, which can be estimated by the triangulation Jacobian J. If D is the baseline between two cameras and θ_1 and θ_2 are the respective headings to the target, the uncertainty Jacobian is given as follows,

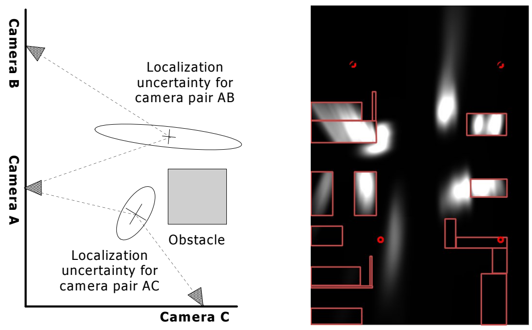
$$J = \frac{D}{\sin^2(\gamma_R - \gamma_L)} \begin{bmatrix} \sin\gamma_R \cos\gamma_R & -\sin\gamma_L \cos\gamma_L \\ \sin^2(\gamma_R) & -\sin^2(\gamma_L) \end{bmatrix}$$

The eigenvalues and eigenvectors of JJ^T define the principle directions of error amplification in stereo triangulation. This ellipsoid in Cartesian space (as shown in Figure 4 (a)) can be interpreted as the error covariance in stereo localization and consequently, the spatially anisotropic uncertainty of the stereo imaging geometry.

We use a kernel density estimation to model the detection history. The probability of observing target r at a grid location $\vec{x}$ is given by

$$p_d(\vec{x}|r) = N_d \sum_{k=1}^{D_i} K_d(\vec{x} - \vec{x}_k) \tag{5}$$

Where $\vec{x}_k$ are the locations where the target is detected, D_i is the total number of previous detections and $K_d(\cdot)$ is a suitable kernel function (here, a Gaussian). The Gaussian kernel K_d is scaled and rotated using the eigenvalues and eigenvectors of JJ^T. N_d is the normalization factor. Figure 4 (b) shows an example of the accumulated detection history where brighter areas have higher probability of containing the object.



(a) (b)

Figure 4. Detection history. (a) Localization uncertainty for camera pairs . (b) Accumulated detection history.

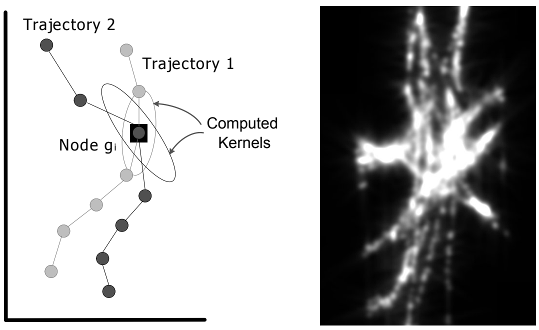
4.2. Activity density from people tracking

People interact with objects in the course of many tasks associated with daily living. The aggregation of commonly observed tracking trajectories representing daily activities of the human subject provides a wealth of associated information about the possible locations of the objects related to these activities. To model activity density, a robust person tracking module using the PTZ camera network has been developed and used in our system. The same camera nodes used in object search are also used as tracking agents. Each PTZ camera performs tracking on its captured frames using color and edge features. The PTZ camera node periodically sends the location information of the tracked person to the central PC. Using corresponding blobs from each camera, the system can triangulate the location of people so that global 3D tracking is achieved. [1]

Given the observed trajectories, activity density can be modeled using anisotropic kernels [5], and an object's PDF can be estimated by normalizing the activity density map using the normalization factor N_a,

$$p_a(\vec{x}|r) = N_a \sum_{k=1}^{T_i} K_a(\vec{x} - \vec{x}_k) \tag{6}$$

where T_i is the total number of tracking points observed. For each tracking point p, we first compute the resultant $\vec{V}(p)$ of the motion vectors passing through that point in the same trajectory. The Gaussian kernel K_a is scaled and rotated using the magnitude $|\vec{V}|$ and the direction θ of the resultant vector respectively. Note that different from the method in [5] where only one anisotropic kernel is computed for each location using the average of all the motion vectors passing through that grid node, in our approach a grid node may have more than one kernel computed. The kernel for each tracked point is calculated using only the motion vectors passing through this node and on the same trajectory. By this means small scale movements are preserved. Figure 5 (a) shows two computed anisotropic kernels overlapping in one location. Figure 5 (b) shows the target's PDF estimated with activity density after 137 trajectories were recorded.



(a) (b)

Figure 5. Activity density. (a) Anisotropic kernels. (b) Accumulated activity density by people tracking.

4.3. Fusion of two measurements

The measures of detection history and activity density defined above can be used separately to estimate the prior PDF of the target. We can also compose them together to get an overall estimate of the prior PDF,

$$p(\vec{x}|r) = N p_h(\vec{x}|r) \cdot p_a(\vec{x}|r) \tag{7}$$

[1]The tracking subsystem is also tolerant to failure or task interruption of the camera agent by adopting multiple "Fault Containment Units (FCU)" [5] to achieve redundancy in 3D tracking.

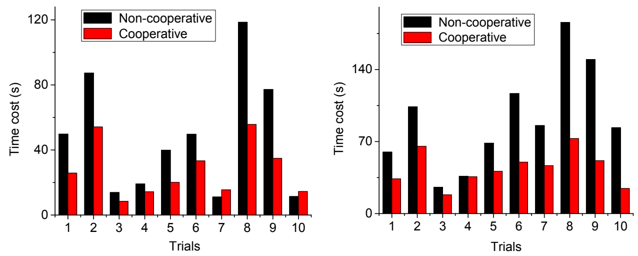
where N is a normalization factor that ensures that the sum of probability from all nodes is 1.

5. Experimental Results

Our experimental smart space consists of four fixed Sony EVI-D100 PTZ cameras that are used for both object search and people tracking. Each camera is attached with a small local computer containing an Intel 2.5GHz dual-core processor to form an agent. The local computer is used to process the captured images and communicate to the central PC and other agents over an 802.11g wireless network. In our experiment, we use a paper card with solid green color as the target object. We tessellate the horizontal plane into grid nodes with a resolution of 2×2 cm. Two horizontal levels are considered, the floor plane (0 feet height) and the surface of all the tables (2.4 feet height).

5.1. Cooperative search performance

To evaluate the benefit of using the cooperative search strategy, we first compare the efficiency of the proposed cooperative search strategy to the method without the cooperation mechanism, in which all agents perform independent search without information exchange. In this experiment no context information is used, i.e., the prior PDF of the target is a uniform distribution. We measured the "time to detect" cost of the search in 10 independent tests. In each test the target object was placed randomly in the environment. Figure 6 (a) shows the time cost when the first agent detects the target and (b) shows the time cost when second agent detects the target in the same test. In almost every trial, the proposed cooperative search strategy is significantly more efficient than the approach without cooperation.



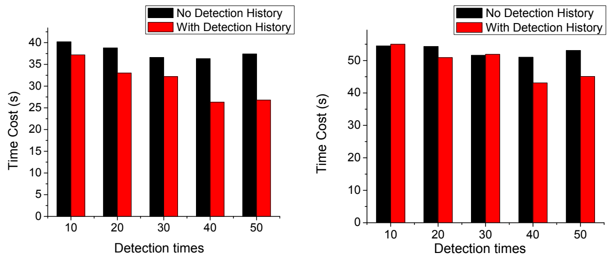
(a) (b)

Figure 6. Cooperative search performance. (a) First detection. (b) Second detection.

5.2. Search performance with Context information

This section presents the improvement of search efficiency by using context information. The detection history and activity density are applied to build target's prior PDF respectively and are compared to the search results where no context information is used to refine the PDF. In these experiments, the cooperative search strategy is always used.

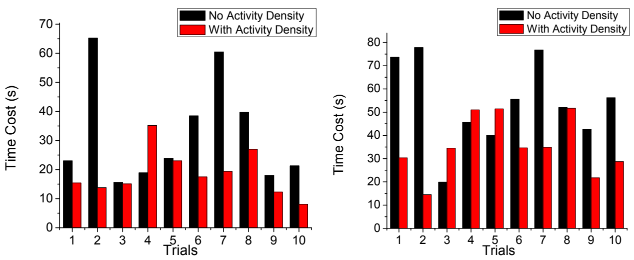
Figure 7 illustrates the influence of the detection history on the time required for a 1st and 2nd detection in the search. The PDF was formed incrementally. There are 5 repetitions of 10 trials each, each repetition uses detection history accumulated from the previous repetition. The average time cost of those 10 search trials is calculated and shown in the figure. Here black columns represent the result using a uniform PDF, and red columns corresponds to the result using detection history information. The figure shows that the 1st and 2nd detection time for search is reduced as more detection history is accumulated. After 50 trials, the time cost for searching for a target is about 70% of the time required using the uniform distribution.



(a) (b)

Figure 7. Cooperative search performance with detection history. (a) First detection. (b) Second detection.

The influence of the activity density is given in Figure 8. In this experiment, the system monitored the movement of the people in the environment and accumulated activity density over time. After 137 trajectories were generated and used to establish the prior PDF, 10 search trials were executed. In these trials people walk around in the room and place the object randomly. Figure 8 shows the comparison between the search results with and without the prior PDF conditioned using activity density information. It can be seen that the search efficiency was improved by an average of 42% when the activity density is used.



(a) (b)

Figure 8. Cooperative search performance with activity density. (a) First detection. (b) Second detection.

6. Conclusion and Future Works

This paper presents a complete object search system that can be used for elder care in smart home environment. A decentralized and asynchronous cooperative search strategy is developed so that the system is tolerant to the failure and interruption of the search agent. Context information that influences the search performance is investigated. Our experimental results demonstrate that the proposed cooperative search strategy is efficient and the methods we use to incorporate context information into the target's probabilistic distribution can improve the efficiency further.

While our study demonstrates that the system is efficient and useful, there is still room for improvement. (1) We plan to use more sophisticated algorithm to recognize fine-grained activities such as Walking, Standing, and Reading a book. Object use can be inferred from activity recognition, and can be used to build more accurate model for object location, (2) some mobile agents can be incorporated in the current search framework so that the blind area of the fixed PTZ cameras can be eliminated, (3) and we are also very interested in investigating how the current search system can cooperate with human beings who may participate in the search as well.

References

[1] Intel open source computer vision library.

[2] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte. Coordinated decentralized search for a lost target in a bayesian world. *IROS*, 2006.

[3] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte. Optimal search for a lost target in a bayesian world. *Field and Service Robotics*, 2006.

[4] P. DeLima and D. Pack. Toward developing an optimal cooperative search algorithm for multiple unmanned aerial vehicles. *International Symposium on Collaborative Technologies and Systems*, 2008.

[5] D. Karuppiah, R. Grupen, A. Hanson, and E. Riseman. Smart resource reconfiguration by exploiting dynamics in perceptual tasks. *Proceedings of International Conference on Robotics and Systems*, 2005.

[6] F. Saidi, O. Stasse, and K. Yokoi. A visual attention framework for a visual search by a humanoid robot. *International Conference on Humanoid Robots*, 2006.

[7] F. Saidi, O. Stasse, K. Yokoi, and F. Kanehiro. Online object search with a humanoid robot. *IROS*, 2007.

[8] V. A. Sujan and S. Dubowsky. Efficient information-based visual robotic mapping in unstructured environments. *International Journal of Robotics Research*, 2005.

[9] L. E. Wixson. Gaze selection for visual search. *Ph.D. dissertation, Department of Computer Science, University of Rochester*, 1994.

[10] Y. Ye and J. K. Tsotsos. Sensor planning in 3d object search. *CVIU*, 1996.