Sensor Planning in 3D Object Search

Yiming Ye and **John K. Tsotsos** *

Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 1A4
yiming@vis.toronto.edu tsotsos@vis.toronto.edu

1 Introduction

The research described in this paper addresses the task of efficiently searching for a given 3D object in a given 3D environment. The "searcher" is a mobile robotic platform equipped with a camera and, if the environment configuration is not known, a method of calculating depth, like a stereo or laser range finder. In general, this task consists of three subtasks. The first subtask is the selection of the sensing parameters so as to bring potential targets into the field of view of the sensor. This is the sensor planning problem for object search, and it is the main concern of this document. The second subtask is the manipulation of the hardware so that the sensing operators can reach the state specified by the planner. The third subtask involves searching for the target within the image. This is the object recognition and localization problem, which attracts a lot of attention within the computer vision community.

Although sensor planning for object search is very important if a robot is to interact intelligently and effectively with its environment, it is interesting to note that there is little research on this subject within the computer vision community. Garvey [3] proposes the idea of indirect search for the target. Wixson et al. [8] present a mathematical model of search efficiency and analyze the efficiency of indirect search and conclude that indirect search can improve efficiency in many situations. Connell et al. [2] construct a robot that roams an area searching for and collecting soda cans. Although these works involve the task of object search, none of them give an explicit algorithm to control the state parameters of the camera by taking both the search agent's knowledge about the target distribution and the ability of the recognition algorithm into the consideration.

This paper, for the first time, formalizes the sensor planning task for object search, analyzes the task at the complexity level and proposes a practical sensor planning strategy. It decomposes the huge space of possible sensing actions into a finite set of actions that must be considered, thus greatly simplifying the sensor planning task. Finally, experimental results are presented.

2 Problem Formulation

It is important to examine different aspects of object search individually, to study their relationship and to integrate them into a whole search system.

The searcher model is based on the ARK robot [5], which is a mobile platform equipped with a special sensor: the Laser Eye [4]. The Laser Eye is mounted on a robotic head with pan and tilt capabilities. It consists of a camera with controllable focal length (zoom), a laser range-finder and a mirror. The mirror is used to ensure collinearity of effective optical axes of the camera lens and the range finder. The state of the searcher is uniquely determined by 7 parameters $(x_c, y_c, z_c, p, t, w, h)$. Where (x_c, y_c, z_c) is the position of the camera center, (p, t) is the direction of the camera viewing axis (p is the amount of pan $0 \leq p < 2\pi$, t is the amount of tilt $0 \leq t < \pi$). w, h are the width and height of the solid viewing angle of the camera.

* The second author is a Fellow of the Canadian Institute for Advanced Research. The authors would like to thank the reviewers for their valuable suggestions. They are also grateful to Dr. Dave Wilkes, Dr. Piotr Jasiobedzki, Victor Lee, and Eric R. Harley for their comments.

An operation $\mathbf{f} = \mathbf{f}(x_c, y_c, z_c, p, t, w, h, a)$ is an action of the searcher within the region Ω. Where a is the recognition algorithm used to detect the target. An operation $\mathbf{f}$ entails: take a **perspective** projection image according to the camera configuration of $\mathbf{f}$ and then search the image using the recognition algorithm a.

The search region Ω can be in any form and it is assumed that we know the boundary of Ω exactly but we do not know its internal configuration. In practice, we tessellate the region Ω into a series of elements c_i, $\Omega = \bigcup_{i=1}^{n} c_i$ and $c_i \bigcap c_j = 0$ for $i \neq j$. c_o is the region outside Ω.

The target distribution can be specified by a probability distribution function $\mathbf{p}$. $\mathbf{p}(c_i, t)$ gives the probability that the center of the target is within cube c_i at time t. Note, we use $\mathbf{p}(c_o, t)$ to represent the probability that the target is outside the search region at time t.

The detection function on Ω is a function $\mathbf{b}$, such that $\mathbf{b}(c_i, \mathbf{f})$ gives the conditional probability of detecting the target given that the center of the target is located within c_i and the operation is $\mathbf{f}$. It is obvious that the probability of detecting the target by applying action $\mathbf{f}$ is given by $P(\mathbf{f}) = \sum_{i=1}^{n} \mathbf{p}(c_i, t_{\mathbf{f}})\mathbf{b}(c_i, \mathbf{f})$, where $t_{\mathbf{f}}$ is the time just before $\mathbf{f}$ is applied.

The reason that the term $t_{\mathbf{f}}$ is introduced in the calculation of $P(\mathbf{f})$ is that the probability distribution needs to be updated whenever an action fails. By using Bayes' formula, we can get the following updating rule:

$$\mathbf{p}(c_i, t_{\mathbf{f}+}) \leftarrow \frac{\mathbf{p}(c_i, t_{\mathbf{f}})(1 - \mathbf{b}(c_i, \mathbf{f}))}{\mathbf{p}(c_o, t_{\mathbf{f}}) + \sum_{j=1}^{n} \mathbf{p}(c_j, t_{\mathbf{f}})(1 - \mathbf{b}(c_j, \mathbf{f}))}$$

where $i = 1, \ldots, n, o$.

The cost $\mathbf{t_o}(\mathbf{f})$ gives the total time needed to (1)manipulate the hardware to the status specified by $\mathbf{f}$; (2)take a picture; (3)updating the environment and registering the space; (4)run the recognition algorithm.

Let $\mathbf{O}_\Omega$ be the set of all the possible operations that can be applied. The effort allocation $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_k\}$ gives the ordered set of operations applied in the search, where $\mathbf{f}_i \in \mathbf{O}_\Omega$. The probability of detecting the target by this allocation is:

$$P[\mathbf{F}] = P(\mathbf{f}_1) + [1 - P(\mathbf{f}_1)]P(\mathbf{f}_2) + \ldots + \{\prod_{i=1}^{k-1}[1 - P(\mathbf{f}_i)]\}P(\mathbf{f}_k)$$

The total cost for applying this allocation is (following [4]): $T[\mathbf{F}] = \sum_{i=1}^{k} \mathbf{t_o}(\mathbf{f}_i)$.

Suppose K is the total time that can be allowed in the search, then the task of sensor planning for object search can be defined as finding an allocation $\mathbf{F} \subset \mathbf{O}_\Omega$, which satisfies $T(\mathbf{F}) \leq K$ and maximizes $P[\mathbf{F}]$.

3 Theoretical Analysis

In this section, we analysis the sensor planning task for object search at the complexity level.

We first list some properties of the sensor planning task. Proofs are omitted (refer to [9] for detail).

Suppose $\Omega = \bigcup_{i=1}^{n} c_i$, $\mathbf{O}_\Omega = \{\mathbf{f_1^*}, \mathbf{f_2^*}, \ldots, \mathbf{f_m^*}\}$. For any operation $\mathbf{f} \in \mathbf{O}_\Omega$, we define its influence range as $\Omega(\mathbf{f}) = \{c \mid \mathbf{b}(c, \mathbf{f}) \neq 0\}$. Its complement form is: $\overline{\Omega(\mathbf{f})} = \Omega - \{c \mid \mathbf{b}(c, \mathbf{f}) \neq 0\}$.

For any effort allocation $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_q \mid \mathbf{f}_i \in \mathbf{O}_\Omega\}$. The initial probability distribution is denoted as $\mathbf{p}^{[0]}(c_1), \mathbf{p}^{[0]}(c_2), \ldots, \mathbf{p}^{[0]}(c_n), \mathbf{p}^{[0]}(c_o)$. After the application of the operation $\mathbf{f}_1$, the distribution is denoted by $\mathbf{p}^{[1]}(c_1), \mathbf{p}^{[1]}(c_2), \ldots, \mathbf{p}^{[1]}(c_n), \mathbf{p}^{[1]}(c_o)$. Generally, after the application of the operation $\mathbf{f}_i$, the distribution is denoted by $\mathbf{p}^{[i]}(c_1), \mathbf{p}^{[i]}(c_2), \ldots, \mathbf{p}^{[i]}(c_n), \mathbf{p}^{[i]}(c_o)$, where $1 \leq i \leq q$. Let $P(\mathbf{f}_i)$ represent the probability of detecting the target by applying the action $\mathbf{f}_i$ with respect to the allocation $\mathbf{F}$. Then of course we have $P(\mathbf{f}_i) = \sum_{j=1}^{n} \mathbf{p}^{[i-1]}(c_j)\mathbf{b}(c_j, \mathbf{f_i})$. Let $P^{[0]}(\mathbf{f}_i)$ represents the probability of detecting the target by applying the action $\mathbf{f}_i$ when there is no action been applied before, then we have $P^{[0]}(\mathbf{f}_i) = \sum_{j=1}^{n} \mathbf{p}^{[0]}(c_j)\mathbf{b}(c_j, \mathbf{f_i})$.

Lemma 1. For allocation $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_q \mid \mathbf{f}_i \in \mathbf{O}_\Omega\}$, we have

$$\mathbf{p}^{[k]}(c) = \frac{\mathbf{p}^{[0]}(c)t_1(c)t_2(c)\ldots t_k(c)}{(1 - P(\mathbf{f}_1))(1 - P(\mathbf{f}_2))\ldots,(1 - P(\mathbf{f}_k))}$$

where

$$t_i(c) = \begin{cases} 1 - \mathbf{b}(c, \mathbf{f}_i) & \text{if } c \in \Omega(\mathbf{f}_i) \\ 1 & \text{otherwise} \end{cases}, i = 1, \ldots, k$$

Lemma 2 For the given allocation $\mathbf{F} = \{\mathbf{f_1}, \ldots, \mathbf{f_q}\}$, we have

$$\begin{aligned} P(\mathbf{F}) = \sum_{i_1=1}^{q} P^{[0]}(\mathbf{f}_i) &+ (-1)^{2+1} \sum_{1 \le i_1 < i_2 \le q} \Big[\sum_{c \in \Omega(\mathbf{f}_{i_1}) \bigcap \Omega(\mathbf{f}_{i_2})} \mathbf{p}^{[0]}(c)\mathbf{b}(c, \mathbf{f}_{i_1})\mathbf{b}(c, \mathbf{f}_{i_2}) \Big] \\ &+ \ldots \\ &+ (-1)^{r+1} \sum_{1 \le i_1 < i_2 < \ldots < i_r \le q} \Big[\sum_{c \in \Omega(\mathbf{f}_{i_1}) \bigcap \Omega(\mathbf{f}_{i_2}) \bigcap \ldots \bigcap \Omega(\mathbf{f}_{i_r})} \mathbf{p}^{[0]}(c)\mathbf{b}(c, \mathbf{f}_{i_1})\mathbf{b}(c, \mathbf{f}_{i_2}) \ldots \mathbf{b}(c, \mathbf{f}_{i_r}) \Big] \\ &+ \ldots \\ &+ (-1)^{q+1} \Big[\sum_{c \in \Omega(\mathbf{f}_1) \bigcap \Omega(\mathbf{f}_2) \bigcap \ldots \bigcap \Omega(\mathbf{f}_q)} \mathbf{p}^{[0]}(c)\mathbf{b}(c, \mathbf{f}_1) \ldots \mathbf{b}(c, \mathbf{f}_q) \Big] \end{aligned}$$

Lemma 3 For a given allocation $\mathbf{F} = \{\mathbf{f_1}, \ldots, \mathbf{f_q}\}$, $P[\mathbf{F}]$ is not influenced by the order of the applied actions. That is to say: $P[\mathbf{F}]$ is symmetric about actions $\mathbf{f}_1, \ldots, \mathbf{f}_q$.

Lemma 4 For an allocation $\mathbf{F} = \{\mathbf{f_1}, \ldots, \mathbf{f_q}\}$, if we restrict the available actions such that there is no cube belongs to any two action ranges: $\Omega(\mathbf{f}_j) \bigcap \Omega(\mathbf{f}_i) = \phi$, if $i \neq j$, then $P[\mathbf{F}]$ can be calculated by the following:

$$P(\mathbf{F}) = \sum_{i=1}^{q} P^{[0]}(\mathbf{f}_i)$$

Now, we prove that the sensor planning task for object search is *NP-complete*. First, the maximization problem is changed into the equivalent decision problem.

INSTANCE: A search region $\Omega = \bigcup_{i=1}^{n} c_i$, a finite set $\mathbf{O}_\Omega = \{\mathbf{f_1}^*, \mathbf{f_2}^*, \ldots, \mathbf{f_m}^*\}$ (the set of all possible operations that can be applied), an expected probability function $P[\mathbf{F}']$ as described in (2) and a cost function $T[\mathbf{F}']$ as described in (3) for an effort allocation $\mathbf{F}'$. Where $\mathbf{F}' = \{\mathbf{f}_1, \ldots, \mathbf{f}_k \mid \mathbf{f}_i \in \mathbf{O}_\Omega\}$ is the ordered set of operations applied in the search. An environmental updating rule as described in (1), a cost constraint K and an expected probability goal M.

QUESTION: Is there an effort allocation $\mathbf{F}$ such that $T[\mathbf{F}] \le K$ and $P[\mathbf{F}] \ge M$

It is obvious that the above problem belongs to NP. In the following, we use proof by restriction to prove that the object search task belongs to the NP-complete class.

Theorem 1 The sensor planning for object search task described in section 4.1 is NP-Complete.

Proof:

We prove the theorem by restriction.

Actually we already known that $KNAPSACK$ problem is NP-Complete. We will restrict the object search task into $KNAPSACK$ by allowing instances in which any two operations in $\mathbf{O}_\Omega = \{\mathbf{f_1}^*, \mathbf{f_2}^*, \ldots, \mathbf{f_m}^*\}$ do not have common action region. This is to say that $\Omega(\mathbf{f_i}) \bigcap \Omega(\mathbf{f_j}) = \phi$ for any $i \neq j$.

After this restriction, by combining **Lemma 3** and **Lemma 4**, the task becomes

INSTANCE: A search region $\Omega = \bigcup_{i=1}^{n} c_i$, a finite set $\mathbf{O}_\Omega = \{\mathbf{f_1}^*, \mathbf{f_2}^*, \ldots, \mathbf{f_m}^*\}$ (the set of all possible operations that can be applied), an expected probability function $P[\mathbf{F}'] = \sum_{i=1}^{k} P^{[0]}(\mathbf{f_i})$ (from Lemma 4) and a cost function $T[\mathbf{F}'] = \sum_{i=1}^{k} \mathbf{t}(\mathbf{f_i})$ for an effort allocation $\mathbf{F}'$. Where $\mathbf{F}' = \{\mathbf{f}_1, \ldots, \mathbf{f}_k \mid \mathbf{f}_i \in \mathbf{O}_\Omega\}$ is the set of operations applied in the search. (Note, here we do not have updating rule and the order of actions does not matter from **Lemma 3**). A cost constraint K and an expected probability goal M.

QUESTION: Is there an effort allocation $\mathbf{F}$ such that $T[\mathbf{F}] \leq K$ and $P[\mathbf{F}] \geq M$

Obviously the above problem is equivalent to the *KNAPSACK* problem. Thus, the task of sensor planning for object search is NP-Complete. $\square$.

Since the sensor planning problem is NP-complete and usually the number of the available candidate actions are big, it is necessary to relax the original problem. Instead of looking for an algorithm that always generate an optimal solution, we will simply use heuristics that will generate a feasible solution for the original problem. Since the calculation of the detection probability for a given operation is time consuming, it is not practical to use complex strategies. Here we use greedy strategy. The greedy method suggests that one can device an algorithm which works in stages, considering one input at a time. At each stage, based on some optimization measure, the next candidate is selected and is included into the partial solution so far. Suppose we have already executed k ($k \geq 1$) actions $\mathbf{F}_k = \{\mathbf{f}_1, \ldots, \mathbf{f}_k\}$. We now want to find the next action $\mathbf{f}_{k+1}$ to execute, with the hope that our strategy of finding the next action may lead to an **approximate** solution for the object search task. Suppose $P[\mathbf{F}_q]$ is the expected probability of detecting the target by allocation $\mathbf{F}_q$. $T[\mathbf{F}_q]$ is the cost of allocation $\mathbf{F}_q$.

When we execute a next action $\mathbf{f}$, the effort allocation becomes $\mathbf{F}_{q+1} = \{\mathbf{f}_1, \ldots, \mathbf{f}_q, \mathbf{f}\}$. The expected probability of detecting the target is $P[\mathbf{F}_{q+1}] = P[\mathbf{F}_q] + \Delta_P(\mathbf{f})$, where $\Delta_P(\mathbf{f}) = \{\prod_{j=1}^{q}[1 - P(\mathbf{f_j})]\} \times P(\mathbf{f})$. The total cost becomes $T[\mathbf{F}_{q+1}] = T[\mathbf{F}_q] + \Delta_T(\mathbf{f})$, where $\Delta_T(\mathbf{f}) = \mathbf{t}(\mathbf{f})$. Our strategy for selecting the next action is: the next action $\mathbf{f_{q+1}}$ should be selected that maximizes the term $\frac{\Delta_P(\mathbf{f})}{\Delta_T(\mathbf{f})}$. Since the term $\{\prod_{j=1}^{q}[1 - \sum_{i=1}^{n} \mathbf{p}(c_i, t_{\mathbf{f_j}})\mathbf{b}(c_i, \mathbf{f_j})]\}$ is fixed no matter what the next action $\mathbf{f}$ is selected, our strategy becomes: the next action $\mathbf{f_{q+1}}$ should be selected that maximizes the term

$$\mathbf{E}(\mathbf{f}) = \frac{\sum_{i=1}^{n} \mathbf{p}(c_i, t_{\mathbf{f}})\mathbf{b}(c_i, \mathbf{f})}{\mathbf{t}(\mathbf{f})}$$

In some situations, the simple greedy strategy may even generate the optimal answers as stated in Lemma 5.

Lemma 5: Suppose the available operations $\mathbf{O}_{\Omega} = \{\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_m}\}$ satisfy:

(1) $\mathbf{t}(\mathbf{f_i}) = \mathbf{t}(\mathbf{f_j})$, $1 \leq i, j \leq m$.

(2) $\Omega(\mathbf{f}_j) \bigcap \Omega(\mathbf{f}_i) = \phi$, $1 \leq i, j \leq m, i \neq j$.

Then, the above greedy strategy generates the optimal answer.

4 Detection Function

The reference detection function $\mathbf{b}((\theta_0, \delta_0, l_0), < a, w_0, h_0 >)$ gives the probability of detecting the target when the target is at $(\theta_0, \delta_0, l_0)$ relative to the camera and the camera angle size is $< w_0, h_0 >$. Normally, we only need to record the detection function values of $< w_0, h_0 >$. The detection function values of other angle sizes can be found by the following transformation (for detail, see [9]):

$$l_0 = l\sqrt{\frac{tan(\frac{w}{2})tan(\frac{h}{2})}{tan(\frac{w_0}{2})tan(\frac{h_0}{2})}}; \quad \theta_0 = arctan[tan(\theta)\frac{tan(\frac{w_0}{2})}{tan(\frac{w}{2})}]; \quad \delta_0 = arctan[tan(\delta)\frac{tan(\frac{w_0}{2})}{tan(\frac{w}{2})}]$$

In general, when we need to find $\mathbf{b}(c, \mathbf{f})$, we can first calculate the relevant $\mathbf{b}((\theta, \delta, l), < a, w, h >)$, then transform it into the known $\mathbf{b}((\theta_0, \delta_0, l_0), < a, w_0, h_0 >)$ to obtain the detection function value.

5 Where to Look Next

For a given robot position and a given recognition algorithm, there are many possible viewing angle sizes. However, the whole search region can be examined with high probability of detection using only a small number of them. For a given angle size, the probability of successfully recognizing the target is high only when the target is within a certain range of distance. This range is called the effective range for the given angle size. Our purpose here is to select those angles whose effective ranges will cover the entire depth D of the search region and at the same time there will be no overlap of their effective ranges. Suppose that the biggest viewing angle for the camera is $w_0 \times h_0$ and its effective range is $[N_0, F_0]$. Then the necessary angle sizes and the corresponding effective ranges are:

$$w_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{w_0}{2})]; \quad h_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{h_0}{2})] \quad N_i = F_0(\frac{F_0}{N_0})^{i-1} \quad F_i = F_0(\frac{F_0}{N_0})^i$$

Where $1 \leq i \leq \lfloor \frac{ln(\frac{D}{F_0})}{ln(\frac{F_0}{N_0})} - 1 \rfloor$.

For each angle size derived above, there are infinite number of viewing directions that can be considered. We have designed an algorithm that can generate only directions such that their union can cover the whole viewing sphere without overlap. Only the actions with the viewing angle sizes and the corresponding directions obtained by above method are taken as the candidate actions. So, we have decomposed the huge space of possible sensing actions into a finite set of actions that must be tried. Finally, we can use $E(\mathbf{f})$ to select among them for the best viewing angle size and direction.

After the selected action is applied, if the target is not detected, we need to update the probability distribution and select a new action. When the benefit of the new action is too low, the robot is required to change its position.

6 Where to Move Next

The goal of the "where to move next" task is to select the next robot position such that, at the new position, the camera can examine those parts of the region that have high probability of presence of the target but which are occluded if viewed from the previous positions. In order to perform this task, first we need to decide which candidate positions the robot **can reach**, then, we need to know which one is the **most beneficial** among these reachable positions. In order to simplify the navigation problem (our primary task is object search, not navigation), we limit the movement of the robot from the current position to the next position to be on **straight lines**.

We have developed an algorithm that can decide which positions the robot can reach from the current position. To select the next robot position among these reachable candidate positions, we need to know the approximate unoccluded region that can be seen by the camera with respect to each reachable candidate position. The space around the center of the camera can be divided into a set of solid angles. Each solid angle is associated with a radius which is the length of an emitting line along the direction of the central axis of the solid angle from the origin. The unoccluded region around the camera center can thus be represented by the union of these solid angles. This representation is called a sensed sphere. The expected sensed sphere SS_e refers to the sensed sphere that can be constructed at the reachable candidate positions. SS_e can be **calculated** by using the boundary of the search region and the solidity property of the cubes in the region (see Figure 1 for a $2D$ illustration). Each candidate position is associated with an expected sensed sphere. For each SS_e, we can calculate the probability of presence of the target by summing up the probabilities of all the cubes that are within SS_e: $\sum_{c_i \in SS_e} \mathbf{p}(c_i)$. We then select the position whose corresponding expected sensed sphere has the maximum probability of presence of the target as the next robot position.

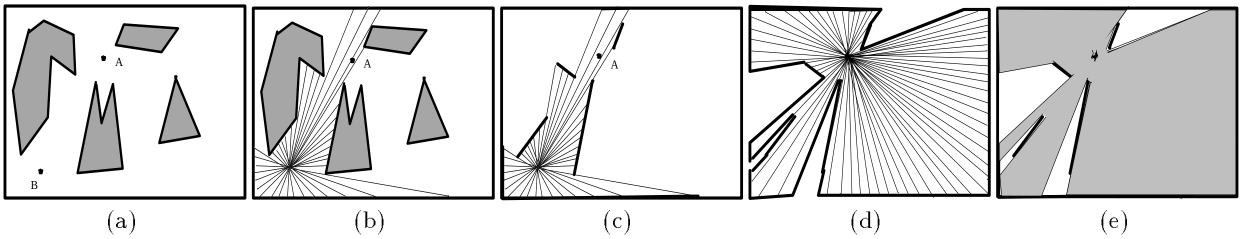

Fig. 1.: $2D$ illustration of the calculation of the expected sensed sphere. (a) The environment, **B** is the robot's current position, **A** is the candidate position. The shaded area are occupied by solid obstacles. (b) The sensed sphere constructed by laser at the first position **A**. (c) The solid cubes generated based on results of (b). (d) The expected sensed sphere calculated by just taking the boundary of the searching region into consideration. (e) The expected sensed sphere at the candidate position **A**.

After the robot moves to the next position, it will begin the "where to look next" process again to search for the target.

At this point, we must consider the inaccuracy of the robot movement. We have proposed a strategy to incorporate the inaccuracy of the robot movement into the robot's knowledge about the environment and the target distribution (see [9] for detail).

7 Experimental Results

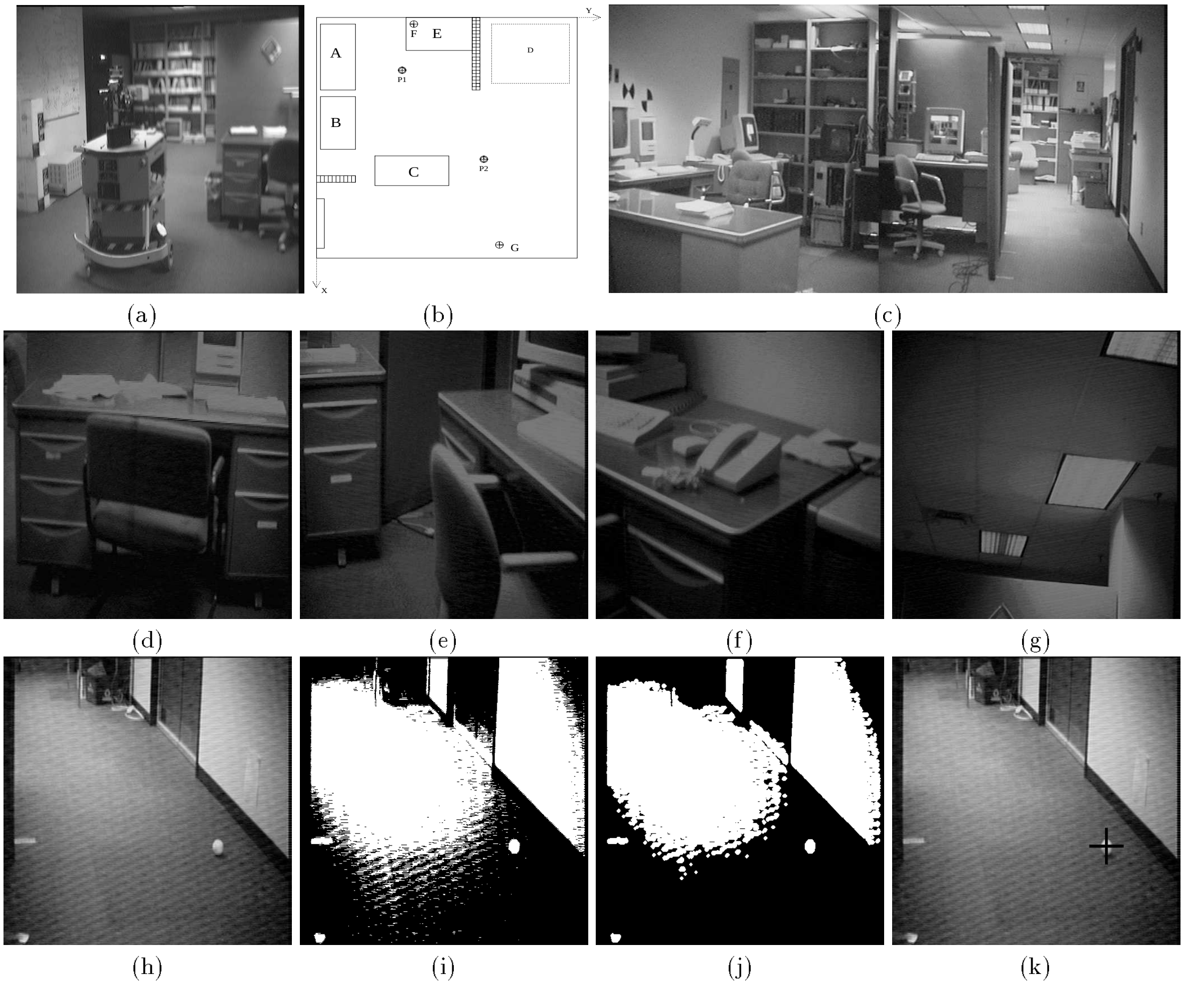

Fig. 2.: The real experiment performed in our Lab with the ARK robot.

Experiments are performed in our lab using the ARK robot (Figure 2(a)). The search region is a part of our lab. Figure 2(b) is the top view of the region, Figure 2(c) is the image of the region taken from point G. The task is to search for a white baseball within this region. Before the search process, we give the table top **B**, table top **C**, and the floor region **D** high probability of presence (Figure (b)). The real position of the baseball is within the region **D**. Initially, the robot is at P_1 of Figure (b). Figure 2(d)(e)(f)(g) is the top four images taken by the head. The first action examines table **C**, the second and third actions examine table **B**. The fourth action simply points upward and none of the specified region is examined. No action is selected to check region **D**, because **D** is occluded by the office wall. Since the probability of detecting the target by the fourth action is very low, the robot drives to the next position P_2 (Figure 2(b)). P_2 is selected by the "where to move next" algorithm. At the new position, the robot begin the search process again. The first action detects the target (Figure 2(h)). Figure 2(i)(j)(k) is the image analysis result, where the target is found (Figure 2(k)).

8 Conclusion

An interesting phenomenon of most past computer vision research is that the purely theoretical side of computer vision is too heavily emphasized. Consequently, many theoretical results cannot be applied to real situations, because they are obtained by making unrealistic assumptions. We think it is beneficial to put more emphasis on the "task oriented vision" framework as an alternative to the classic reconstructionist paradigm. For this new framework, the vision system is just one component of the agent embedded in the world, and it should provide only the minimum perceptual functionality the agent needs to perform whatever it is doing at the moment.

Object search is a typical task for this new framework. For this task, the vision module involved is object recognition. In order to find the target, the agent has to purposefully change the sensing parameters so as to bring the target into the field of view of the camera and to make the image of the target easily recognized by the available recognition algorithms. The study of the object search task may generate insight into existing vision problems. For example, the problem of object recognition may be less difficult when the image is intelligently grabbed by a controlled camera and the image analysis is interpreted within specific contexts. Research shows that the object recognition problem can be simpler if the camera is intelligently controlled [7]. Of course, the simplicity obtained in the image analysis phase is at the cost of the complexity encountered in the camera controlling phase. Much work has been done on image analysis. It is now time to study the problems related to the control of the camera. This is one reason why the sensor planning task for object search should be studied and comprises one of the motivations of this paper.

In this paper, we study the problem of sensor planning for object search from both the aspect of theory and that of practical implementation. By introducing the concept of a detection function for the object recognition algorithm and the concept of a probability distribution for the target within the search region, we formulate the sensor planning task as an optimization problem: the goal is to maximize the probability of detecting the target within a given time constraint. By analyzing the formulation of the problem, we obtain some important properties of this task and find that this task is NP-complete in general. In order to make the problem tractable, we propose to use the greedy strategy, and point out that under some special conditions this strategy may generate an optimal answer. By considering the detection ability of operations, we are able to decompose the enormous space of possible sensing actions into a limited finite set of actions that must be tried, thus greatly reducing the complexity in the action selection process. Our theory has been applied using the ARK robot and the Laser Eye. Experiments so far have been successful as a proof of concept. A future direction is the design of a search strategy when the target is able to move. Another direction is the design of a search strategy when several robots are available.

References

1. R. Bajcsy. Active perception vs. passive perception. In *Third IEEE Workshop on Vision*, USA, 1985.
2. Connel. *An Artificial Creature*. PhD thesis, AI Lab, MIT, 1989.
3. T. D. Garvey. Perceptual strategies for purposive vision. Technical Report Technical Note 117, SRI International, 1976.
4. P. Jasiobedzki, M. Jenkin, E. Milios, B. Down, and J. Tsotsos. Laser eye - a new 3d sensor for active vision. In *Intelligent Robotics and Computer Vision: Sensor Fusion VI. Proceedings of SPIE. vol. 2059*, pages 316–321, Boston, Sept. 1993.
5. S. Nickerson, M. Jenkin, E. Milios, B. Down, P. Jasiobedzki, A. Jepson, D. Terzopoulos, J. Tsotsos, D. Wilkes, N. Bains, and K. Tran. Ark: Autonomous navigation of a mobile robot in a known environment. In *Intelligent Autonomous Systems-3*, pages 288–296, Pennsylvania, USA, February 1993.
6. J. Tsotsos. Analyzing vision at the complexity level. *The behavioral and brain science*, 13:423–469, 1990.
7. D. Wilkes and J. Tsotsos. Active object recognition. In *Proceedings of Computer Vision and Pattern Recognition*, pages 136–141, Illinois, USA, June 1992.
8. L. E. Wixson. *Gaze Selection for Visual Search*. PhD thesis, Department of Computer Science, University of Rochester, 1994.
9. Y. Ye. *Sensor planning in 3D object search*. PhD thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A4, 1996.
10. Y. Ye and J. K. Tsotsos. The detection function in object search. In *Proceedings of the fourth international conference for young computer scientist*, pages 868–873, Beijing, 1995.

This article was processed using the TEX macro package with SIRS96 style