Where to Look Next in 3D Object Search

Yiming Ye   John K. Tsotsos
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 1A4

Abstract

The task of sensor planning for object search is formulated and a mechanism for "where to look next" for this task is presented. The searcher is assumed to be a mobile platform equipped with an active camera and a method of calculating depth, like stereo or a laser range finder. The formulation casts sensor planning as an optimization problem: the goal is to maximize the probability of detecting the target object with minimal cost. The search space is thus characterized by the probability distribution of the presence of the target. The control of the sensing parameters depends on the current state of the search space and the detecting ability of the recognition algorithm. In order to represent the environment and to efficiently determine the sensing parameters over time, a concept called the sensed sphere is proposed and its construction, using a laser range finder, is derived. The result of each sensing operation is used to update the status of the search space.

1 Introduction

Object search is the task of finding a given $3D$ object in a given $3D$ environment. It is clear that exhaustive, brute-force blind search will suffice for its solution; however, our goal is the design of efficient strategies for search because exhaustive search is computationally and mechanically prohibitive for non-trivial situations. Generally speaking, this task contains three parts. The first is how to select the sensing parameters so as to bring the target into the field of view of the sensor. This is the sensor planning problem for object search, which is the main concern of this paper. The second is how to manipulate the hardware so that the sensing operators can reach the state specified by the planner. The third is how to search 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 wants to interact intelligently and effectively with its environment, there is little research within the computer vision community ([9], [12], [3], [6], [8]). Connell [2] constructed a robot that roams an area searching for and collecting soda cans. The planning is very simple since the robot just follows the walls of the room and the sensor only searches the area immediately in front of the robot. This may not be very efficient since the likely presence of the target is not considered when the robot is roaming. Rimey and Brown [8] used a composite Bayes net and utility decision rule to plan the sensor action in their task-oriented system TEA. The sensor is directed to the center of mass of the expected area for a certain object based on the belief value of the net. Probability of presence is used in their system, but the detection ability of the sensor is not considered and the purpose of the sensor planning is mainly verification instead of searching. The indirect search mechanism proposed by Garvey [3] is to first direct the sensor to search for an "intermediate" object that commonly participates in a spatial relationship with the target and then direct the sensor to examine the restricted region specified by this relationship. Wixson [12] presented a mathematical model of search efficiency and predicted that indirect search can improve efficiency in many situations. The problems with indirect search are that the spatial relationships between target and "intermediate" objects may not always exist and the detection of the "intermediate" object may not always be easier than the detection of the target. It is interesting to note that the operational research community has done a lot of research on optimal search [5]. Their purpose is to determine how to allocate effort to search for a target, such as a lost submarine in the ocean or an oil field within a certain region. Although the results are elegant and beautiful in a mathematical sense, they cannot be directly applied here because the searcher model is too abstract and general and there is no sensor planning involved in their approach.

This paper proposes a practical mechanism for the task of sensor planning for object search. This task is formulated as an optimization problem: select an ordered sequence of camera configurations that maximize the probability of finding the target with minimum cost. This optimization task is further simplified as a decision problem: decide only which is the very next action to execute, considering the effect and cost of the candidate action.

2 Problem Formulation

In this section, we first explain the searcher model, the environment, and some basic concepts used in our discussion, and then formulate the object search task and give a simplified version of this task.

The searcher model is based on the ARK robot, 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 (the starting point of the camera viewing axis), (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. (x_c, y_c, z_c) can be adjusted by moving the mobile platform. (p, t) can be adjusted by the motors on the robotic head. w, h can be adjusted by the zoom lens of the camera. We assume in this paper that the camera's image plane is always coincident with its focal plane.

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 $i \neq j$. In the rest of the paper, we assume the search region is an office-like environment, and we tessellate the space into little cubes of the same size. Usually the size of the cube is determined by the size of the environment and the size of the target [13].

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 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. Usually this distribution is assumed to be known at the beginning of the search process and it is determined by our knowledge of the world. If we know nothing about the distribution, then we can assume a uniform distribution at the beginning. 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 $\mathbf{f}$. For any operation, if the projection of the center of the cube c_i is outside the image, we assume $\mathbf{b}(c_i, \mathbf{f}) = 0$; if the cube is occluded or it is too far from the camera or too near to the camera, we also have $\mathbf{b}(c_i, \mathbf{f}) = 0$. In general [13], $\mathbf{b}(c_i, \mathbf{f})$ is determined by various factors, such as intensity, occlusion, and orientation etc. It is obvious that the probability of detecting the target by applying action $\mathbf{f}$ is given by

$$P(\mathbf{f}) = \sum_{i=1}^{n} \mathbf{p}(c_i, t_\mathbf{f}) \mathbf{b}(c_i, \mathbf{f}) \tag{1}$$

where $t_\mathbf{f}$ is the time just before $\mathbf{f}$ is applied. Let Ψ be the set of all the cubes that are within the field of view of $\mathbf{f}$ and that are not occluded, then we have

$$P(\mathbf{f}) = \sum_{c \in \Psi} \mathbf{p}(c, t_\mathbf{f}) \mathbf{b}(c, \mathbf{f}) \tag{2}$$

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. Here we use Bayes' formula. Let α_i be the event that the center of the target is in cube c_i, α_o be the event that the center of the target is outside the search region, let β be the event that after applying a recognition action, the recognizer successfully detects the target. Then $P(\neg\beta \mid \alpha_i) = 1 - \mathbf{b}(c_i, \mathbf{f})$ and $P(\alpha_i \mid \neg\beta) = \mathbf{p}(c_i, t_{\mathbf{f}+})$. Where $t_{\mathbf{f}+}$ is the time after $\mathbf{f}$ is applied. Since the above events $\alpha_1, \ldots, \alpha_n, \alpha_o$ are mutually complementary and exclusive, we get the following updating rule

$$\mathbf{p}(c_i, t_{\mathbf{f}+}) \leftarrow \frac{\mathbf{p}(c_i, t_\mathbf{f})(1 - \mathbf{b}(c_i, \mathbf{f}))}{\mathbf{p}(c_o, t_\mathbf{f}) + \sum_{j=1}^{n} \mathbf{p}(c_j, t_\mathbf{f})(1 - \mathbf{b}(c_j, \mathbf{f}))} \tag{3}$$

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)update the environment and register the space; (4)run the recognition algorithm. We assume that: (1) and (2) are same for all the actions; (3) is a constant; (4) is known for any recognition algorithm. Then, $\mathbf{t_o}(\mathbf{f})$ is only influenced by (4).

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 $\mathbf{f}_i \in \mathbf{O_\Omega}$. It is clear that the probability of detecting the target by this allocation is:

$$P[\mathbf{F}] = P(\mathbf{f}_1) + \ldots + \{\prod_{i=1}^{k-1} [1 - P(\mathbf{f}_i)]\} P(\mathbf{f}_k) \tag{4}$$

The total cost for applying this allocation is (following [10]):

$$T[\mathbf{F}] = \sum_{i=1}^{k} \mathbf{t_o}(\mathbf{f}_i) \tag{5}$$

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 $P[\mathbf{F}]$.

Since this task is NP-Complete [14], we consider a simpler problem: decide only which is the very next action to execute. Suppose we have already executed q ($q \geq 1$) actions $\mathbf{F}_q = \{\mathbf{f}_1, \ldots, \mathbf{f}_q\}$. We now want to find the next action to execute, with the hope that our strategy of finding the next action may finally lead to an **approximate** optimal solution of the object search task. For any next action $\mathbf{f}$, its contribution to the probability of detecting the target is $\Delta_P(\mathbf{f}) = \{\prod_{j=1}^{q} [1 - P(\mathbf{f}_j)]\} P(\mathbf{f})$. The additional cost is $\Delta_T(\mathbf{f}) = \mathbf{t_o}(\mathbf{f})$. Since $\{ \prod_{j=1}^{q}[1 - P(\mathbf{f}_j)]\}$ is fixed, the next action should be selected that maximizes the term

$$E(\mathbf{f}) = \frac{P(\mathbf{f})}{\Delta_T(\mathbf{f})} \tag{6}$$

Note, the above strategy may sometimes led to an optimal solution (see [14] for detail).

Because of limited space, in this paper we only address the "where to look next " problem: how to select w, h, p, t, a of $\mathbf{f}$ so as to maximize $E(\mathbf{f})$ for a fixed camera position (For the discussion of the "where to move next" problem, please refer to [13]).

3 Detection Function

We briefly discuss the detection function in this section. For details, please refer to [13].

The standard detection function $\mathbf{b_0}((\theta, \delta, l), < a, w, h >)$ gives a measure of the detecting ability of the recognition algorithm a when there is no previous action. $< w, h >$ is the viewing angle size of the camera, (θ, δ, l) is the relative position of the center of the target to the camera, $\theta = arctan(\frac{x}{z})$, $\delta = arctan(\frac{y}{z})$ and $l = z$, (x, y, z) is the coordinate of the target center in the camera coordinate system. The value of $\mathbf{b_0}((\theta, \delta, l), < a, w, h >)$ can be obtained empirically. We can first put the target at (θ, δ, l) and then perform experiments under various conditions, such as light intensity, background situation, and the relative orientation of the target with respect to the camera center. The final value is the total number of successful recognitions divided by the total number of experiments. These values can be stored in a look up table indexed by θ, δ, l and retrieved when needed. Sometimes we may approximate these values by analytic formulas.

We only need to record the detection values of one angle size $< w_0, h_0 >$. Those of other sizes can be approximately transformed to those of size $< w_0, h_0 >$. Suppose (θ, δ, l) is the target position for angle size $< w, h >$, we want to find the value $(\theta_0, \delta_0, l_0)$ for angle size $< w_0, h_0 >$ such that $\mathbf{b_0}((\theta_0, \delta_0, l_0), < a, w_0, h_0 >) \approx \mathbf{b_0}((\theta, \delta, l), < a, w, h >)$. To guarantee this, the images taken with parameter $< \theta_0, \delta_0, l_0, w_0, h_0 >$ and $< \theta, \delta, l, w, h >$ should be almost the same. Thus, the area and position of the projected target image on the image plane should be almost the same for both images, we get

$$l_0 = l\sqrt{\frac{tan(\frac{w}{2})tan(\frac{h}{2})}{tan(\frac{w_0}{2})tan(\frac{h_0}{2})}} \tag{7}$$

$$\theta_0 = arctan[tan(\theta)\frac{tan(\frac{w_0}{2})}{tan(\frac{w}{2})}] \tag{8}$$

$$\delta_0 = arctan[tan(\delta)\frac{tan(\frac{w_0}{2})}{tan(\frac{w}{2})}] \tag{9}$$

When the configurations of two operations are very similar, they might be correlated with each other (refer to [13] for detail). Repeated actions are avoided during the search process. When independence is assumed, $\mathbf{b}(c_i, \mathbf{f})$ is calculated as follows. First, calculate the corresponding (θ, δ, l) of the center of c_i with respect to operation $\mathbf{f}$. Second, transform (θ, δ, l) into the corresponding $(\theta_0, \delta_0, l_0)$ of angle size $< w_0, h_0 >$. Third, retrieve the detection value from the look up table, or get the detection value from a formula.

4 The sensed sphere

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 environment can thus be represented by the union of these solid angles. This representation is called the **sensed sphere** [9]. We can use a laser range finder to construct the sensed sphere. First, we need to **tessellate** the surface of the unit sphere centered at the camera center into a set of surface patches. Then, we need to ping the laser at the center of each patch so as to get the radius of each solid angle.

In order to make the tessellation as uniform as possible and to make the number of mechanical operations as small as possible, we use the following method.

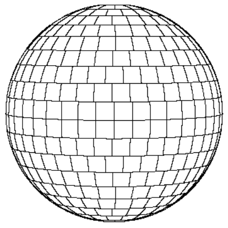


Fig. 1

First, we tessellate the range $[0, \pi]$ of tilt uniformly by a factor $2m$, m is integer. This tessellation in general depends on the complexity of the environment. Thus the tilt ranges are $[0, \alpha), \ldots, [(i-1)\alpha, i\alpha), \ldots, [\pi - \alpha, \pi)$, where $\alpha = \frac{\pi}{2m}$. Then, for each tilt range except $[0, \alpha)$ and $[\pi - \alpha, \pi)$, we tessellate the range $[0, 2\pi)$ of pan. In the tessellation, the amount of change of pan Δ_i for tilt range $[(i-1)\alpha, i\alpha)$ is

$$\Delta_i = \begin{cases} 2arcsin(\frac{sin\frac{\alpha}{2}}{sin(i\alpha)}) & \text{if } i\alpha \leq \frac{\pi}{2} \\ 2arcsin(\frac{sin\frac{\alpha}{2}}{sin[(i-1)\alpha]}) & \text{if } (i-1)\alpha \geq \frac{\pi}{2} \end{cases}$$

So, for tilt range $[(i-1)\alpha, i\alpha)$, the pan ranges are $[0, \Delta_i)$, $[\Delta_i, 2\Delta_i)$, ..., $[n_i\Delta_i, 2\pi)$. The length of each pan range is Δ_i except the last one $[n_i\Delta_i, 2\pi)$. Fig. 1 shows a side view of a tessellation with $m = 10$. The sensed sphere constructed with the tessellation scheme described above can be concisely represented as the union of all solid angles

$$\bigcup_{[t_{b_i}, t_{e_i})=[(i-1)\alpha, i\alpha), i=1,\ldots,2m} \left\{ \bigcup_{[p_{b_{i,j}}, p_{e_{i,j}})=[0,\Delta_i),[\Delta_i,2\Delta_i),\ldots,[n_i\Delta_i,2\pi)} A_{ij}(t_{b_i}, t_{e_i}; p_{b_{i,j}}, p_{e_{i,j}}; r_{ij}) \right\}$$

Where r_{ij} is the length of the radius along the direction $tilt = \frac{t_{b_i}+t_{e_i}}{2}$, $pan = \frac{p_{b_{i,j}}+p_{e_{i,j}}}{2}$. $[t_{b_i}, t_{e_i})$, $[p_{b_{i,j}}, p_{e_{i,j}})$ and r_{ij} give the range of the A_{ij}, $t_{b_i} \leq tilt < t_{e_i}, p_{b_{i,j}} \leq pan < p_{e_{i,j}}$. Note, the sensed sphere representation is similar to a radially organized occupancy grid representation [13].

5 Where to look next

We need to select w, h, p, t, a for the next action. First, we select w, h, p, t for each given recognition algorithm.

The ability of the recognition algorithm and the value of the detection function are influenced by the image size of the target. Only when the target can be brought wholly into the field of view of the camera and the features can be detected with certain precision, can the recognition algorithm be expected to function correctly. So, for an operation with a given recognition algorithm and a fixed viewing angle, the probability of successfully recognizing the target is high only when the target is within a certain distance range. We call this range the **effective range**. Our purpose here is to select those angles whose effective ranges will cover the whole distance of the 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 size for the camera is $w_0 \times h_0$ and its effective range is $[N_0, F_0]$. We want to select other necessary viewing angle sizes $w_1 \times h_1, \ldots, w_{n_0} \times h_{n_0}$ and their corresponding effective ranges $[N_1, F_1], \ldots, [N_{n_0}, F_{n_0}]$, such that $[N_0, F_0] \bigcup \ldots \bigcup [N_{n_0}, F_{n_0}] \supseteq [N_0, D]$ and $[N_i, F_i) \bigcap [N_j, F_j) = \emptyset$ if $i \neq j$. To guarantee this, we have $F_{i-1} = N_i, i = 1, \ldots, n_0$ and the area of the image of the target patch for angle size $w_{i-1} \times h_{i-1}$ at N_{i-1} and F_{i-1} should equal to the area of the image of the target patch for angle size $w_i \times h_i$ at N_i and F_i respectively. According to section 3, we can get

$$w_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{w_0}{2})] \tag{10}$$

$$h_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{h_0}{2})] \tag{11}$$

$$N_i = F_0(\frac{F_0}{N_0})^{i-1} \tag{12}$$

$$F_i = F_0(\frac{F_0}{N_0})^i \tag{13}$$

Where $1 \leq i \leq n_0$. Since $N_i \leq D$, we get $i \leq \frac{ln(\frac{D}{F_0})}{ln(\frac{F_0}{N_0})} - 1$. So, $n_0 = \lfloor \frac{ln(\frac{D}{F_0})}{ln(\frac{F_0}{N_0})} - 1 \rfloor$.

The sensed sphere can be further divided into several layers according to the effective viewing angle sizes derived above. This layered sensed sphere LSS can be represented as

$$LSS = \bigcup_{<w_k, h_k>, k=0,\ldots,n_0} LSS_{<w_k,h_k>} \tag{14}$$

Where $LSS_{<w_k,h_k>}$ is the layer corresponding to angle size $< w_k, h_k >$, $LSS_{<w_k,h_k>} = \bigcup_{i,j} LA_{ij}^{<w_k,h_k>}$. The layered solid angle slice $LA_{ij}^{<w_k,h_k>}$ is the intersection of the solid angle A_{ij} of the sensed sphere and the current layer $LSS_{<w_k,h_k>}$. There is a probability $p_{ij}^{<w_k,h_k>}$ associated with $LA_{ij}^{<w_k,h_k>}$, which gives the sum of the probabilities of all the cubes that belong to $LA_{ij}^{<w_k,h_k>}$ (Note: the distance of each cube to the camera center should be less than r_{ij} of A_{ij}).

The following is used to select the viewing direction for a given angle size $< w_k, h_k >$. First, tessellate the sphere using the method in section 4 with angle size equal to $min\{w_k, h_k\}$. Each resulting patch corresponds to a viewing direction. Second, calculate the sum of all $p_{ij}^{<w_k,h_k>}$ for those $LA_{ij}^{<w_k,h_k>}$ that belong to a given patch. This is the probability for this patch. Third, the direction $< p_k, t_k >$ whose corresponding patch has the maximum probability is the best direction for size $< w_k, h_k >$.

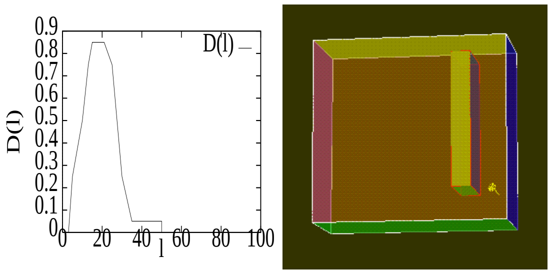
For each recognition algorithm, we can find $n_0 + 1$ candidates $< w_k, h_k, p_k, t_k >$ $(0 \leq k \leq n_0)$. Then, use $P(\mathbf{f})$ to select among them to get the best candidate for this algorithm. Lastly, because different algorithms have different costs, we use $E(\mathbf{f})$ to select among the best candidates for all the recognition algorithms so as to find the next action to be applied. The environment needs to be updated if the target is not found after the selected action is applied.

6 Experiment

We assume only one recognition algorithm is available for all the experiments. The first simulation experiment (results shown in Fig. 2 and Fig. 3) is used to test the general scheme of our algorithm. The biggest angle for the sensor is $\frac{\pi}{4} \times \frac{\pi}{4}$. The detection function is $\mathbf{b_0}((\theta, \delta, l), < a, \frac{\pi}{4}, \frac{\pi}{4} >) = D(l)(1 - \frac{1}{6}\frac{\theta}{\frac{\pi}{4}})(1 - \frac{1}{6}\frac{\delta}{\frac{\pi}{4}})$, where $D(l)$ is shown in Fig. 2(a). The search region is shown in Fig. 2(b). Only two angle sizes are needed to examine the search region with respect to the first robot position. They are $\frac{\pi}{4} \times \frac{\pi}{4}$ and 0.376×0.376. Their effective ranges are $[11, 27]$ and $[27, 66]$ respectively. We assume the outside probability is 0.5 and the distribution within the room is uniform at the beginning. Fig. 3(b) shows that the actions are selected by our algorithm to only examine the unoccluded region.

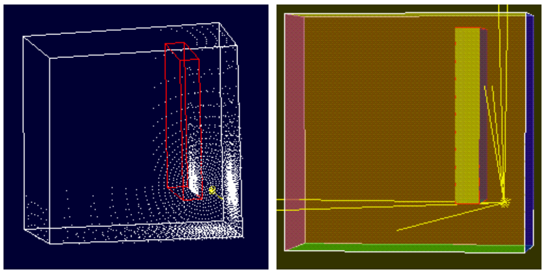
The environment, the robot position and the sensor model for the second simulation (results shown in Fig. 4) are same as those of the first except that there is no obstacle in the room. The target distribution satisfies a 3-variate normal distribution $N(\mu, \mathbf{\Sigma})$, where the mean vector $\mu = (25, 25, 15)^T$, the covariance matrix $\mathbf{\Sigma} = diag(\sigma^2, \sigma^2, (\frac{\sigma}{2})^2)$. We can notice from Fig. 4 that the number of actions needed to reach the detection limit for the planning strategy is much smaller than that for non-planning strategy. This illustrates that the planning strategy is more efficient.

The real experiment is performed in our lab using the Laser Eye. The task is to search for a white baseball within the region shown in Fig. 5 (a)(b). Four criteria are used by the recognition algorithm. They are intensity $I_0 = 119$, blob size $B_{min} = 250$ pixels, $B_{max} = 1375$ pixels, and roundness percentage $R_0 = 0.91$. The algorithm first generates a bi-



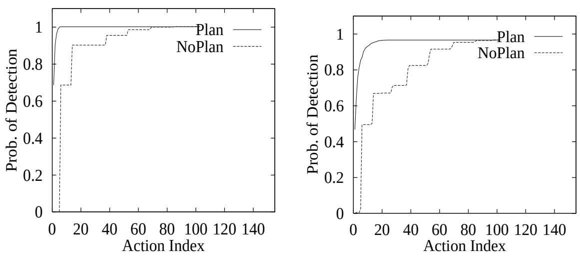
(a) (b)

Fig. 2: Simulation 1. (a) The $D(l)$; (b) The room ($length \times width \times height = 50 \times 50 \times 25$), the obstacle ($5 \times 37 \times 19$) and the robot (the white blob, $height = 15$);



(a) (b)

Fig. 3: Simulation 1. (a) The sensed sphere at the first position. White points represent the intersections of the laser and the environment. (b) Top 7 actions selected. The short and long lines correspond to the viewing axis of actions with angle size $\frac{\pi}{4} \times \frac{\pi}{4}$, and 0.316×0.316 respectively.



(a) (b)

Fig. 4: Simulation 2. The detection probabilities $P[\mathbf{F}]$ of the planning strategy (Plan) and those of the non-planning strategy (NoPlan). (a)$\sigma = 1$; (b)$\sigma = 5$. The non-planning strategy means first execute every action corresponding to the first layer one by one, then execute every action corresponding to the second layer one by one.

nary image by thresholding the image using I_0. Then it performs morphological and region growing operations. Only the white blob whose size is between B_{min} and B_{max} and whose roundness is bigger than R_0 can be taken as the target. Two angle sizes are needed to search the region. They are $41^o \times 39^o$ and $20.7^o \times 19.7^o$. Their effective ranges are $[1.47m, 3.01m]$ and $[3.01m, 6.16m]$ respectively. Experimental results are listed in Table 1, which gives the average number of actions needed to find the ball for planning and non-planning strategies. Where (A) means to give the region corresponding to the table surface A high probability at the beginning. The same for (B), (C), (ABC). From the second and third row of Table 1, we can see that the planning strategy is much more efficient when our knowledge about the distribution is correct at the beginning. From the fourth row of Table 1 we can see that the performance of the planning strategy is not very satisfactory when we have misleading information at the beginning. One of the above test results is shown in Fig. 5 (d)(e)(f)(g)(h).

Target Pos.	A	C	A, B or C
$\mathbf{Plan}(^{correct}_{knowledge})$	(A)1	(C)1	(ABC)2.5
No Plan	8	10	9
$\mathbf{Plan}(^{misleading}_{knowledge})$	(C)7	(B) 11.5	(C)4.3

Table 1

In our strategy, the huge space of possible sensing actions is decomposed into a limited number of actions that must be tried. With respect to these limited actions, our algorithm may even generate a near optimal action sequence for the object search task in some situations. Suppose the available operations are $\mathbf{O_\Omega} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_m\}$. For any operation $\mathbf{f} \in \mathbf{O_\Omega}$, we define its influence range as $\Omega(\mathbf{f}) = \{c \mid \mathbf{b}(c, \mathbf{f}) \neq 0\}$. We have previously proved the following result ([14]): if $\mathbf{O_\Omega}$ satisfies: (A) $\mathbf{t_o}(\mathbf{f}_i) = \mathbf{t_o}(\mathbf{f}_j)$, $1 \leq i, j \leq m$; (B) $\Omega(\mathbf{f}_j) \bigcap \Omega(\mathbf{f}_i) = \phi$, $1 \leq i, j \leq m, i \neq j$. Then, the "one step look ahead" strategy generates the optimal answer. From our action selection algorithm, the condition (B) may sometimes be approximately satisfied. This is because the available actions are associated with different layers of the LSS and, for a given layer, we tessellate the sphere such that different actions have no overlap or little overlap of their viewing volumes. So, when there is only one recognition algorithms available (thus (A) is satisfied), our "one step look ahead" strategy for "where to look next" may generate a near optimal answer.

We do not have space to address the "where to move next" problem in this paper. But it is interesting to note that Wixson [12] has done $2D$ simulation experiments and that his results are actually consistent with our results. Please refer to [13] for detail.

7 Conclusion

In this paper, we formulate the sensor planning task for object search and present a practical strategy for the "where to look next" problem of this task. By introducing the concept of sensed sphere and layered sensed sphere, we are able to decompose the huge space of possible sensing actions into a finite set of actions that must be tried and to select the next action among these finite set of actions. By combining the detecting ability of a recognition algorithm and the knowledge of the probability distribution of the target, we argue that the object search problem is quite different from the exploration problem. The concept of detection function for a recognition algorithm may find its applications in other vision task, such as to serve as an evaluation matrix in the comparison of recognition algorithms.

The theory has been applied using a platform equipped with a camera and a laser range finder. Experiments so far have been successful as a proof of concept. We would like to apply the theory to other kind of tasks such as to search for a target on a cluttered table top using a stereo camera.

Acknowledgements

The second author is the CP-Unitel Fellow of the Canadian Institute for Advanced Research. The first author is grateful to Dr. Dave Wilkes and Dr. Piotr Jasiobedzki for their valuable comments, suggestions and generous help, to Victor Lee for introducing and help with the GL library.

References

[1] R. Bajcsy. Active perception vs. passive perception. In *Third IEEE Workshop on Vision*, pages 55-59, Bellaire, 1985.

[2] Connell. *An Artificial Creature.* Ph.D thesis, AI Lab, MIT, 1989.

[3] T. D. Garvey. Perceptual strategies for purposive vision. Technical Report Note 117, SRI International, 1976.

[4] P. Jasiobedzki etc.. Laser Eye - a new 3D sensor for active vision. In *Intelligent Robotics and Computer Vision: Sensor Fusion VI. Proc of SPIE. vol. 2059*, pages 316-321, Boston, 1993.

[5] B. O. Koopman. *Search and Screen: general principles with historical applications.* Pergaman Press, Elmsford, N.Y, 1980.

[6] J. Maver and R. Bajcsy. How to decide from the first view where to look next. In *Proceedings of the DARPA Image Understanding Workshop*, 1990.

[7] D. Reece and S. Shafer. Using active vision to simplify perception for robot driving. Technical Report CMU-CS-91-199, Comp. Sci., Carnegie Mellon, 1992.

[8] R.D. Rimey and C.M. Brown. Where to look next using a bayes net: incorporating geometric relations. In *Second European Conference on Computer Vision*, pages 542–550, Italy, 1992.

[9] J.K. Tsotsos. 3D Search Strategy. *Internal ARK Working Paper, Department of Computer Science, University of Toronto*, 1992.

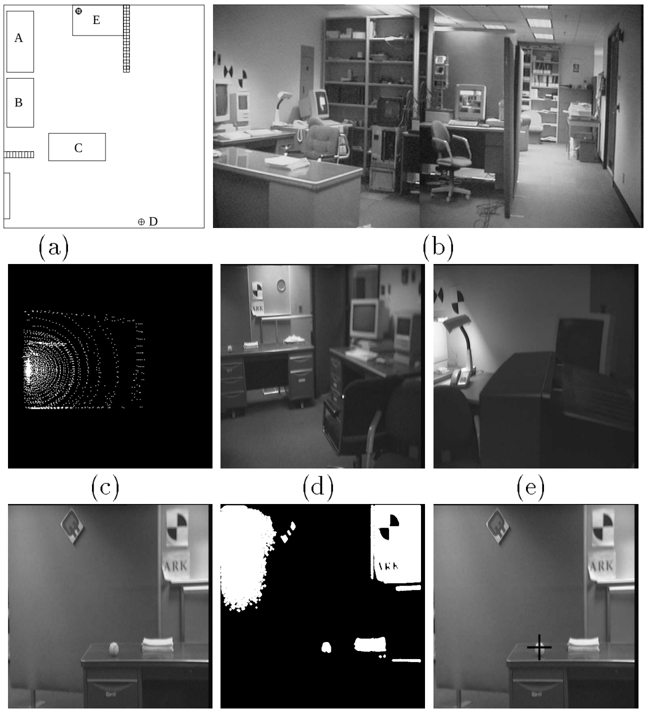
[10] J.K. Tsotsos. Active verses passive perception, which is more efficient? IJCV, 7(2), 1992.

[11] D. Wilkes and J.K. Tsotsos. Active object recognition. In *CVPR*, pages 136–141, USA, 1992.

[12] L. Wixson. *Gaze Selection for Visual Search.* Ph.D thesis, Comp. Sci. Dept., Univ. of Rochester, May 1994.

[13] Y. Ye and J. Tsotsos. Sensor Planning for Object Search. Technical Report RBCV-TR-94-47, Comp. Sci. Dept., Univ. of Toronto, 1994.

[14] Y. Ye and J. Tsotsos. Sensor Planning for Object Search: its Formulation, Property and Complexity. In *36th Annual Symposium on Foundations of Computer Science*, USA. *Submitted*, 1995.



(a) (b)
(c) (d) (e)
(f) (g) (h)

Fig. 5: A real experiment. (a) Top view of the search region, where A, B, C, E are table surfaces. The Laser Eye is on E; (b) Composite image of the region from position D of (a); (c) Sensed sphere from Laser Eye; (d), (e), (f) One of the image sequences of the real experiment by using the planning strategy, where the target is assumed on A, B, or C and we give (ABC) high probability at beginning. Although the ball appeared in the first image (d), the algorithm failed to detect it because it is outside the effective range of the action (size $41^o \times 39^o$). The third action ((f), size $20.7^o \times 19.7^o$) found the target; (g) The image of (f) after region growing etc.; (h) The result of the image analysis of (g), where the target is detected.