Knowledge Difference and its Influence on a Search Agent

Yiming Ye and **John K. Tsotsos**
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 1A4
yiming@vis.toronto.edu and tsotsos@vis.toronto.edu
In First International Conference on Autonomous Agents (AA97), pp 67-74, February 5-8, 1997, M arina del Rey, California. *

Abstract

This paper studies the influence of knowledge difference on the performance of a search agent — – a controllable camera that can pan, tilt and zoom. The task of the agent is to search for a 3D target within a 3D environment. The goal is to maximize the probability of detecting the target within a given period of time. The search agent does this by autonomously controlling its state parameters to bring the target into the field of view of the camera and to make the image of the target with quality such that it can be detected by the available recognition algorithms. Before the search process, the agent has some knowledge of the position of the target. This knowledge is used to guide the sensor planning process. Typically, this knowledge is different more or less from the real situation. This "*knowledge difference*" (the discrepancy between the agent's knowledge and reality) can influence the performance of the agent. In this paper, we study how to formulate quantitatively the knowledge difference and how the performance of the agent is influenced by this difference. We also propose a method to integrate knowledge from different sources such that the *knowledge difference* can be reduced.

Introduction

An agent is a computational system that inhabits dynamic, unpredictable environments. It interprets sensor data that reflect events in the environment, and it executes motor commands that produce effects in the environment. One important property of an agent is its awareness —– it has knowledge about itself and the world. This knowledge can be used to guide its actions when exhibiting goal-directed behaviors. But sometimes the agent's knowledge may not correctly and completely represent the real world situation. This incorrectness and incompleteness of knowledge can influence the effectiveness of the agent's actions.

This paper studies the influence of *knowledge difference* on the performance of an autonomous object search agent. In general, the agent might not have complete knowledge of the state of the environment, and might not have a perfect process model of the effects of its actions on this state. Under these assumptions, the agent must use its limited knowledge to generate a plan of actions so as to fulfill the given task —– this means that the agent must be able to plan under uncertainty. Once the fallibility of the agent's predictive mechanisms is recognized, it becomes obvious that it is impossible in general to guarantee that a particular plan will achieve a given goal. So, the agent is expected to be able to determine a probability that the goal will be achieved if the plan is executed. Although the agent's initial knowledge about the environment is incomplete or invalid before the agent's planning and action execution process, this knowledge is typically used by the agent to guide its planning strategy. Thus, the following important issues arise. The first is how to represent the difference between the agent's initial knowledge about the environment and the inherent property of the environment. The second is how the knowledge difference influences the performance of the agent's planning system. The third is how to integrate knowledge from different sources so as to obtain a favorable initial knowledge. In this paper, we try to answer these questions in the context of $3D$ object search by a camera agent.

Object search is the task of searching for a given 3D object in a given 3D environment by a camera agent that can pan, tilt and zoom. It is clear that exhaustive, brute-force blind search will suffice for its solution; however, the goal of the agent is to design efficient strategies for search, because exhaustive search is computationally and mechanically prohibitive for nontrivial situations. The sensor planning task for the agent refers to the task of selecting the sensing parameters (the camera's viewing direction and viewing angle size) so as to bring the target into the field of view of the sensor and to make the target in the image easily detectable by the given recognition algorithm.

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 (Garvey 1976) proposes the idea of indirect search for the target. Wixson (Wixson 1994) presents a mathematical model of search efficiency and analyzes the efficiency of indirect search, concluding that indirect search can improve efficiency in many situations. Connell (Connell 1989) constructed 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 considering both the search agent's knowledge about the target distribution and the ability of the recognition algorithm.

In this paper, we first briefly describe the formulation of the agent's sensor planning task and the agent's sensing strategy based on this formulation. Then we study an important problem that has not been addressed before — the *knowledge difference* and its influence on the performance of the agent's sensor planning strategy.

General Discussion

In its most general form, the state of an object search system can be denoted as

$$V(\tau) = (E_{conf}, S(\tau), K_{world}(\tau), T_{pos}, A)$$

Here E_{conf} is the geometric configuration of the environment. We assume E_{conf} is static, which means that the geometric configuration will not change during the search process. $S(\tau) = S(s_1(\tau), \ldots, s_m(\tau))$ is the vector of state parameters of the sensor. This state is a function of time, τ, because the sensor parameters are changing during the search process. T_{pos} is the real position of the target in the world. $K_{world}(\tau)$ is the agent's knowledge about the world at time τ. This knowledge includes two parts: one is the agent's knowledge about the geometric configuration of the environment $E^A_{conf}(\tau)$; the other is the agent's knowledge about the possible position of the target $T^A_{pos}(\tau)$. Thus, $K_{world}(\tau) = E^A_{conf}(\tau) \bigcup T^A_{pos}(\tau)$. To simplify the problem, we assume that the agent knows exactly the geometric configuration of the world; thus, $E^A_{conf}(\tau) = E_{conf}$. The agent, however, does not know the exact position of the target; thus, in general $T^A_{pos}(\tau) \neq T_{pos}$. The agent's knowledge about the possible position of the target $T^A_{pos}(\tau)$ is updated after each search action because the agent acts to acquire knowledge of the location of the target. The purpose of object search task is to make this knowledge coincide with the real situation at a certain time τ^*, that is, to make $T^A_{pos}(\tau^*) = T_{pos}$. $A = (a_1, \ldots, a_n)$ is the set of available recognition algorithms used to detect the target from the image. To simplify the discussion, we assume that there is only one recognition algorithm available, thus, $A = (a_1)$.

For the given E_{conf} and T_{pos}, there exists a group of possible sensor parameters $S_{desire} = \{S^*_1, \ldots, S^*_r\}$. If we take an image with the sensor state parameters set to S^*_i (i can be $1, \ldots, r$) and analyze the image using the existing recognition algorithm, then the target can be detected. We call any of the above possible sensor parameters $S^*_1, \ldots, S^*_r$ the "desired sensor state". The goal of object search becomes to adjust the sensor state to one of the desired states, take an image under this state and analyze the picture using the given recognition algorithm. The purpose of sensor planning is to try to realize this goal in a short time. When the target is detected, its real position T_{pos} can be calculated, at which time τ^* we can make $T_{pos} = T^A_{pos}(\tau^*)$.

Suppose that all the possible states for the sensor are $S_1, \ldots, S_N$. Then the sensor planning task is to select a state S from $S_1, \ldots, S_N$, such that $S \in S_{desire}$. This task is very difficult because normally the number of possible states N is huge, and the number of desired sensor states r is relatively small. It is almost impossible to use brute force strategy (to try each possible state one by one) to solve this problem. In order to search this huge space for a desired sensor state, the agent must use the available knowledge $T^A_{pos}(\tau)$ to guide the search process. It must also incorporate the new knowledge from each sensing operations into its knowledge body $K_{world}(\tau)$.

This paper studies the influence of the initial *knowledge difference* $\| T^A_{pos}(\tau_0) - T_{pos} \|$ on the performance of the sensor planning strategy, where τ_0 is the time before the search process.

Sensor Planning Strategy

Formulation

We need to formulate the agent's sensor planning task in a way that incorporates the available knowledge of the agent and the detection ability of the recognition algorithm.

The search region Ω can be in any form, such as a room with many tables, etc. In practice, Ω is tessellated 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, it is assumed that the search region is an office-like environment and it is tessellated into little cubes of the same size.

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. Here (x_c, y_c, z_c) is the position of the camera center (the origin 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; and a is the recognition algorithm used to detect the target.

The agent's knowledge about the possible target position can be specified by a probability distribution function $\mathbf{p}$, so that $\mathbf{p}(c_i, \tau_{\mathbf{f}})$ gives the agent's knowledge about the probability that the center of the target is within cube c_i before an action $\mathbf{f}$ (where $\tau_{\mathbf{f}}$ is the time just before $\mathbf{f}$ is applied). Note, we use $\mathbf{p}(c_o, \tau_{\mathbf{f}})$ to represent the probability that the target is outside the search region at time $\tau_{\mathbf{f}}$.

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 $\mathbf{b}(c_i, \mathbf{f}) = 0$. It is obvious that the probability of detecting the target by applying action $\mathbf{f}$ is given by

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

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

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

The reason that the term $\tau_{\mathbf{f}}$ 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, and 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 $P(\neg\beta \mid \alpha_i) = 1 - \mathbf{b}(c_i, \mathbf{f})$. It is obvious that the updated probability distribution value after an action $\mathbf{f}$ failed should be $P(\alpha_i \mid \neg\beta)$, thus we have $\mathbf{p}(c_i, \tau_{\mathbf{f}+}) = P(\alpha_i \mid \neg\beta)$. Where $\tau_{\mathbf{f}+}$ is the time after $\mathbf{f}$ is applied. Since the above events $\alpha_1, \ldots, \alpha_n, \alpha_o$ are mutually complementary and exclusive, from Bayes formula we get

$$P(\alpha_i \mid \neg\beta) = \frac{P(\alpha_i) P(\neg\beta \mid \alpha_i)}{\sum_{j=1}^{n,o} P(\alpha_j) P(\neg\beta \mid \alpha_j)}$$

So, the probability updating rule is

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

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

The cost $\mathbf{t_o}(\mathbf{f})$ gives the total time needed to perform the operation $\mathbf{f}$.

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:

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

The total cost for applying this allocation is:

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

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 $P[\mathbf{F}]$.

Since this task is NP-Complete (Ye & Tsotsos 1996a), we consider a simpler problem: decide only which is the very next action to execute. Our objective then is to select as the next action the one that maximizes the term

$$E(\mathbf{f}) = \frac{P(\mathbf{f})}{\mathbf{t_o}(\mathbf{f})}$$

We have proved that in some situations, the one step look ahead strategy may lead to an optimal answer.

Selecting Camera Parameters

The agent needs to select the camera's viewing angle size and viewing direction for the next action $\mathbf{f}$ such that $E(\mathbf{f})$ is maximized. Normally, the space of available candidate actions is huge, and it is impossible to take this huge space of candidate actions into consideration. According to the image formation process and geometric relations, we have developed a method that can tessellate this huge space of candidate actions into a small number of actions that must be tried.

A brief description of the sensor planning strategy is as follows (please refer to (Ye 1996b) and (Ye & Tsotsos 1995) for detail). For 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 $\langle w_i, h_i \rangle$ (where $1 \le i \le n_0$) and the corresponding effective ranges $[N_i, F_i]$ (where $1 \le i \le n_0$) are:

$$\begin{aligned}
w_i &= 2arctan[(\tfrac{N_0}{F_0})^i tan(\tfrac{w_0}{2})] \\
h_i &= 2arctan[(\tfrac{N_0}{F_0})^i tan(\tfrac{h_0}{2})] \\
N_i &= F_0(\tfrac{F_0}{N_0})^{i-1} \\
F_i &= F_0(\tfrac{F_0}{N_0})^{i} \\
n_0 &= \lfloor \frac{ln(\frac{D}{F_0})}{ln(\frac{F_0}{N_0})} - 1 \rfloor
\end{aligned}$$

For each angle size derived above, there are an 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 with minimum overlap (Ye 1996b).

Only the actions with the viewing angle sizes and the corresponding directions obtained by the above method are taken as the candidate actions. So, the huge space of possible sensing actions is decomposed into a finite set of actions that must be tried. Finally, $E(\mathbf{f})$ can be used to select among them for the best viewing angle size and direction. After the selected action is applied, if the target is not detected, the probability distribution will be updated and a new action will be selected again.

The Influence of the Knowledge Difference

In order to use the agent's initial knowledge to guide the search process, we formulate the sensor planning problem as a task to maximize the probability of detecting the target within a given time limit. But the agent's real goal is to adjust the sensor state to one of the desired states S_{desire}. Since different initial knowledge corresponds to different sensor sequence, the effect of sensor planning strategy is influenced by the agent's initial knowledge $T^A_{pos}(\tau_0)$. We have performed many experiments to test the influence of $T^A_{pos}(\tau_0)$ on the effects of the sensor planning strategy. The task is to search for a baseball within our Lab (Figure 1(b)(c)) using a special sensor: the Laser Eye, which is a robotic head with a camera that can pan, tilt and zoom (Jasiobedzki et al. 1993) (Figure 1(a)). Experimental results are listed in Table $1(a)(b)$ as the average number of actions needed to bring the ball into the effective viewing volume of the camera by using our planning strategy. For example, the data in the second column of Table $1(a)$ refers to the situation when the target is on table A. Then the number of actions needed to bring the ball into the effective viewing volume of the camera is 1 when we give the table surface A a high probability distribution. The number of actions needed to bring the ball into the effective volume of the camera is 7 when we give the table surface C a high probability distribution (note, the target is on A). Similar explanations can be applied to other columns of Table $1(a)$ and Table $1(b)$. From Table $1(a)(b)$ we can see that the performance of the agent's planning strategy is influenced by the accuracy of the initial knowledge. The more accurate the initial knowledge, the more efficient the planning algorithm.

Target Pos.	A	B	C
Plan$(^{correct}_{knowledge})$	(A)1	(B)1	(C)1
Plan$(^{incorrect}_{knowledge})$	(C)7	(A)8.5	(B) 11.5

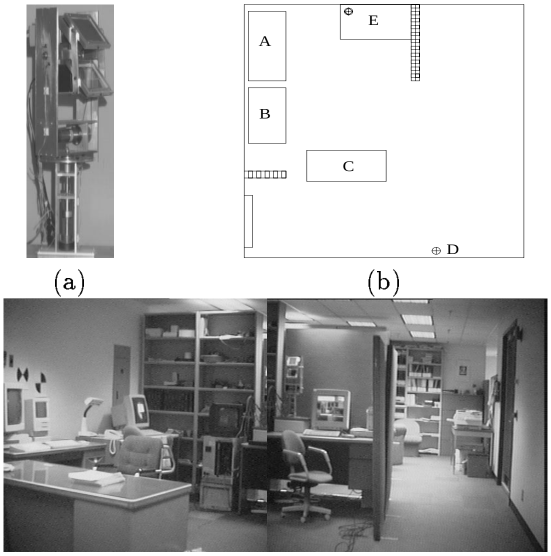
Table 1(a)

Target Pos.	A or B	A or C	A, B or C
Plan$(^{correct}_{knowledge})$	(AB)1.5	(AC)1.5	(ABC)2.5
Plan$(^{incorrect}_{knowledge})$	(C)6	(B)9	(C)4.3

Table 1(b)

The Representation of the Knowledge Difference

From above, we know that the agent's initial knowledge is very important for effective sensor planning. Suppose the center of the target is within the cube c^*, and let the coordinate of the center of c^* be (x^*, y^*, z^*). Then, the probability distribution with respect to the



(a) (b)

(c)

Figure 1: The hardware of the search agent and the search environment

(a) The Laser Eye; (b) Top view of the search environment. A, B, C, E are table surfaces. The Laser Eye is on E; (c) Composite image of the region from position D of (b).

real world situation T_{pos} is: $\mathbf{p}^*(c) = 0$ for all $c \neq c^*$ and $\mathbf{p}^*(c^*) = 1$. Of course, the search agent's *knowledge difference* $|| \ T^A_{pos}(\tau_0) - T_{pos} \ ||$ (the discrepancy between the agent's knowledge and reality) is caused by the difference $\Delta(c_i)$ associated with each cube c_i. So, it should be a function of all the $\Delta(c_i)$.

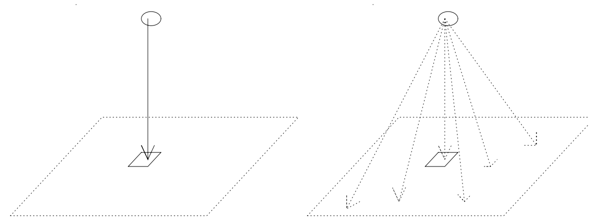
$$||T^A_{pos}(\tau_0) - T_{pos}|| = \Xi(\Delta(c_1), \Delta(c_2), \ldots, \Delta(c_n), \Delta(c_o))$$

where Ξ is an appropriate function.

Since the function Ξ is very difficult to quantify, we simply use $\sum$ to replace it, in the belief that the total knowledge difference is the sum of the discrepancy between the agent's knowledge and reality with respect to each cube. Thus, the difference can be further represented as

$$||T^A_{pos}(\tau_0) - T_{pos}|| \approx \sum_{i=1}^{n,o} \Delta(c_i)$$

For each cube, two factors are important with respect to the agent's knowledge representation and the real situation: the probability assigned by the agent to this cube and the distance of this cube to cube c^* which contains the target center. So, $\Delta(c_i)$ can be represented as a



(a) (b)

Figure 2: An intuitive explanation about $||T^A_{pos}(t_0) - T_{pos}||$.

(a) The perfect knowledge $\mathbf{p}^*$. (b) The initial knowledge according to agent's perception of the world. Some probability masses are mistakenly scattered into other cubes.

function of two factors: a probability difference $|p(c_i) - p^*(c_i)|$ and a position difference $||(x_i, y_i, z_i) - (x^*, y^*, z^*)|| = \sqrt{(x_i - x^*)^2 + (y_i - y^*)^2 + (z_i - z^*)^2}$, where (x_i, y_i, z_i) is the center of cube c_i. Thus, we write

$$\Delta(c_i) = \Theta(| \ \mathbf{p}(c_i) - \mathbf{p}^*(c_i) \ |, ||(x_i, y_i, z_i) - (x^*, y^*, z^*)||)$$

The function Θ is also very difficult to specify. For simplicity, we use the product relation to replace Θ, in the belief that the difference at each cube is proportional to both the probability difference and the position difference. Thus,

$$\begin{aligned} ||T^A_{pos}(\tau_0) - T_{pos}|| \quad \approx \quad & \sum_{i=1}^{n} \Big\{ |\mathbf{p}(c_i) - \mathbf{p}^*(c_i)| \\ & \times ||(x_i, y_i, z_i) - (x^*, y^*, z^*)|| \Big\} \end{aligned}$$

An intuitive explanation of $||T^A_{pos}(\tau_0) - T_{pos}||$ is shown in Figure 2. For the reality T_{pos}, the probability mass is totally concentrated in the cube c^* that contains the target center. But, for the agent's knowledge $T^A_{pos}(\tau_0)$, some of the probability mass is scattered into other cubes. This is what causes the knowledge difference $||T^A_{pos}(\tau_0) - T_{pos}||$. From a certain point of view, the purpose of object search is to bring the mass, which is mistakenly scattered into other cubes, back into the cube that contains the center of the target.

Figure 3 and Table 2 show how well the above formula can represent the real situation. In order to make the visualization easier, we use $2D$ instead of $3D$ to illustrate the result. The $2D$ region is assumed to be a rectangle with size 145×145. The center of the target is at $(75, 75)$. Each pixel represents a square in

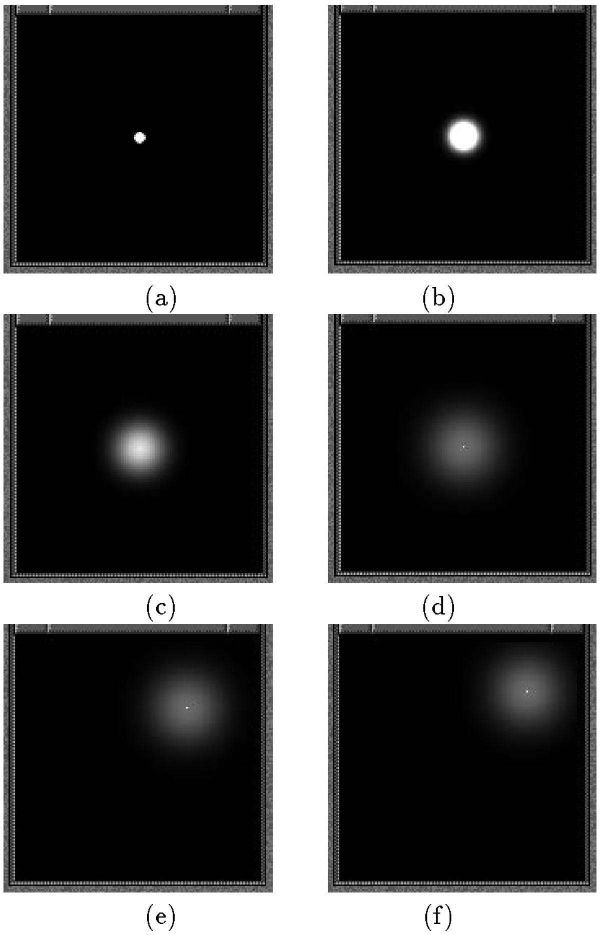


Figure 3: The initial probability from the agent's knowledge. The real position of the target is at the center of the region.

(a) $\sigma_x = \sigma_y = 1$, $\mu_x = \mu_y = 75$;
(b) $\sigma_x = \sigma_y = 5$, $\mu_x = \mu_y = 75$;
(c) $\sigma_x = \sigma_y = 10$, $\mu_x = \mu_y = 75$;
(d) $\sigma_x = \sigma_y = 15$, $\mu_x = \mu_y = 75$;
(f) $\sigma_x = \sigma_y = 15$, $\mu_x = \mu_y = 105$;
(g) $\sigma_x = \sigma_y = 15$, $\mu_x = \mu_y = 115$.

the region. The brightness of each pixel represents the probability assigned to the corresponding square. The higher the value of the probability, the brighter the pixel intensity.

The distribution $T^A_{pos}(\tau_0)$ from the agent's knowledge is assumed to be in the form of a 2-variate normal distribution

$$\mathbf{p}(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}[\frac{(x-\mu_x)^2}{\sigma_x^2}+\frac{(y-\mu_y)^2}{\sigma_y^2}]} \qquad (2)$$

Figure 3 illustrates how $||T^A_{pos}(\tau_0) - T_{pos}||$ changes when $\sigma_x, \sigma_y, \mu_x, \mu_y$ change. Figure 3(a)(b)(c) shows the situation that the center (the mean vector) of the distribution is the same as the target center, while the value of σ changes. It is easy to see that a bigger σ value corresponds to a fuzzier knowledge. This phenomena is clearly represented by the calculated knowledge difference (Table 2), as the bigger the σ_x and σ_y, the bigger the value of $||T^A_{pos}(\tau_0) - T_{pos}||$. Figure 3(d)(e)(f) shows the situation that the value of variance σ_x, σ_y is fixed but the center of the distribution (the mean vector) is changing. Of course, the nearer the center of the distribution to the real target center, the better our knowledge. The phenomenon is also correctly represented by the calculated knowledge difference (Table 2), since the smaller the distance, the smaller the value of $\nabla_{knowledge} = ||T^A_{pos}(\tau_0) - T_{pos}||$.

Parameters	σ_x	σ_y	μ_x	μ_y	$\nabla_{knowledge}$
(a)	1	1	75	75	1.214148
(b)	5	5	75	75	6.265085
(c)	10	10	75	75	12.532640
(d)	15	15	75	75	18.802147
(e)	15	15	105	105	19.958609
(f)	15	15	115	115	26.120419

Table 2

Obtain the Initial Distribution

The initial probability distribution $T^A_{pos}(\tau_0)$ has a huge influence on the agent's performance. A good initial distribution can lead to a quick discovery of the target, while misleading knowledge can greatly degrade the performance. It is thus important to obtain an initial distribution $T^A_{pos}(\tau_0)$ such that the knowledge difference is as small as possible.

T^A_{pos}

knowledge, such as our habits to put the target at a particular place. Since usually we have more than one knowledge source, we need to combine knowledge from all the sources to obtain the initial distribution.

Our strategy is to first use the Dempster-Shafer theory (Schubert 1994) to integrate knowledge derived from a variety of sources. Then, we transform the integrated result into a reasonable probability distribution.

As discussed before, the search space Ω is tessellated into a series of elements c_i. This tessellation actually forms the frame of discernment $\Theta = \{c_1, c_2, \ldots, c_n, c_o\}$. For each knowledge source S_i, we can obtain a belief function m_{S_i}. To combine the belief functions from two knowledge sources S_i and S_j, we use Dempster-Shafer's rule of combination (Schubert 1994)

$$\begin{aligned} m(A) &= m_{S_i} \bigoplus m_{S_j}(A) \\ &= \frac{\sum_{X \bigcap Y = A} m_{S_i}(X) \cdot m_{S_j}(Y)}{1 - \sum_{X \bigcap Y = \phi} m_{S_i}(X) \cdot m_{S_j}(Y)} \end{aligned} \quad (3)$$

where X is a focal element of m_{S_i}; Y is a focal element of m_{S_j}; $m = m_{S_i} \bigoplus m_{S_j}$ is the combined basic probability assignment; and A is a focal element of m.

After integrating the belief functions from all knowledge sources, we need to find a rule that can transform the results into a probability distribution to be used by the sensor planning algorithm. Let $A = \{c_{i_1}, \ldots, c_{i_k}\} \in 2^\Theta$ be a focal element of the combined basic probability assignment m. The mass $m(A)$ corresponds to the part of the belief that is restricted to A and that, due to lack of further information, cannot be allocated to a proper subset of A. In order to build the needed probability distribution, we distribute $m(A)$ equally among the atoms of A. Therefore, $\frac{m(A)}{k}$ is given to each c_{i_j}, $1 \leq j \leq k$. This procedure corresponds to the *Insufficient Reason Principle* (Schubert 1994): if one must build a probability distribution on n elements, given a lack of information, give a probability $\frac{1}{k}$ to each element. This procedure is repeated for each mass m. Thus, we have the following rule to calculate the probability distribution:

$$p(c_i) = \sum_{c_i \in A, A \in 2^\theta, m(A) \neq 0} \frac{m(A)}{|A|} \quad (4)$$

This is the initial distribution that will be used by the agent.

Experiments

We have performed experiments to test our strategy. The task is to search for a baseball. The knowledge comes from two sources. One is from physics. The

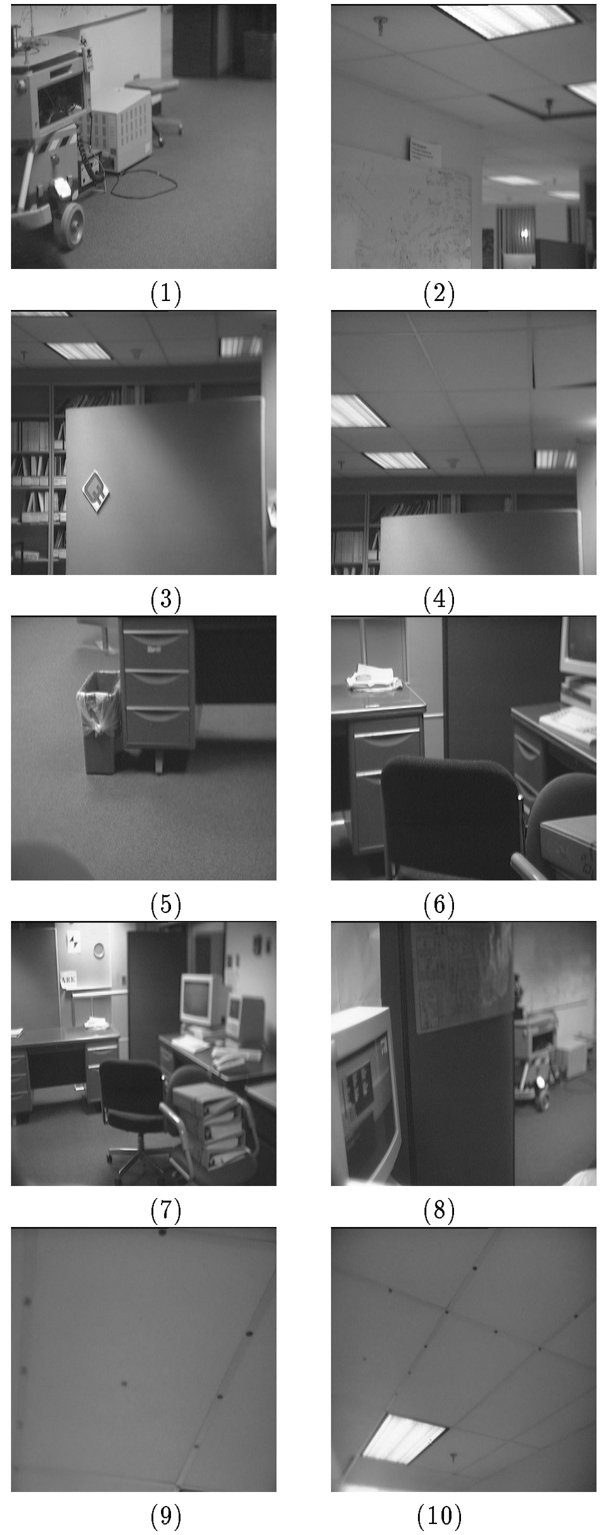


Figure 4: Uniform target distribution before the search process.

The top 10 images taken when the initial distribution is unknown. The target is not detected within these 10 images.

other is from our restrictions on the possible position of the baseball. From physics, a baseball can only appear on a horizontal surface. So, for cube c, if c is around the table surface, the chair surface, or the floor surface, then we give the belief mass $m_{S_1}(c) = 1$, otherwise, $m_{S_1}(c) = 0$. For any focal element A which is the union of several cubes $A = c_{i_1} \bigcup \ldots \bigcup c_{i_j}$, where $j > 1$, we have $m_{S_1}(A) = 0$. Finally, we normalize m_{S_1} such that the total belief mass is 1.

The second knowledge source restricts position of the baseball to heights above 1 meter. So, for cube c, if c is above 1 meter, then $m_{S_2}(c) = 1$, otherwise $m_{S_2}(c) = 0$. We also need to normalize m_{S_2} such that the total belief mass is 1. After we get the values of m_{S_1} and m_{S_2} for each cube, Dempster-Shafer's role of combination is used to integrate m_{S_1} and m_{S_2} to obtain the combined belief mass. This belief mass is transformed into a probability distribution. The resulting probability distribution is used as the agent's initial knowledge. The experimental results show that with the use of the Dempster-Shafer's rule of combination, the performance of the sensor planning system is enhanced. Figure 4 shows the top 10 images taken in a test when the initial knowledge is unknown (the Dempster-Shafer rule of combination is not used). The initial target distribution is thus initialized as a uniform distribution. The target is not detected within these 10 images. Figure 5 shows the top three images taken in another test when the Dempster-Shafer rule of combination is used. The third action detects the target.

Conclusion

This paper addresses the issue of *knowledge difference* and its influence on the behavior of a goal directed agent. The concept of *knowledge difference* is defined and its influence on the performance of a $3D$ object search agent is tested. We suggest a way to quantitatively represent the *knowledge difference* and a way to obtain the initial knowledge for a $3D$ object search agent. We believe that the concept of *knowledge difference* is very important for agent theory in general, and there are many interesting and challenging issues that need to be studied.

Acknowledgments

The authors are grateful to both reviewers and to Eric Harley, Dave Wilkes, and Piotr Jasiobedzki for their valuable comments and to Victor Lee for introducing GL library. This work was supported by IBM Centre for Advanced Studies and ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates Inc., Indus-

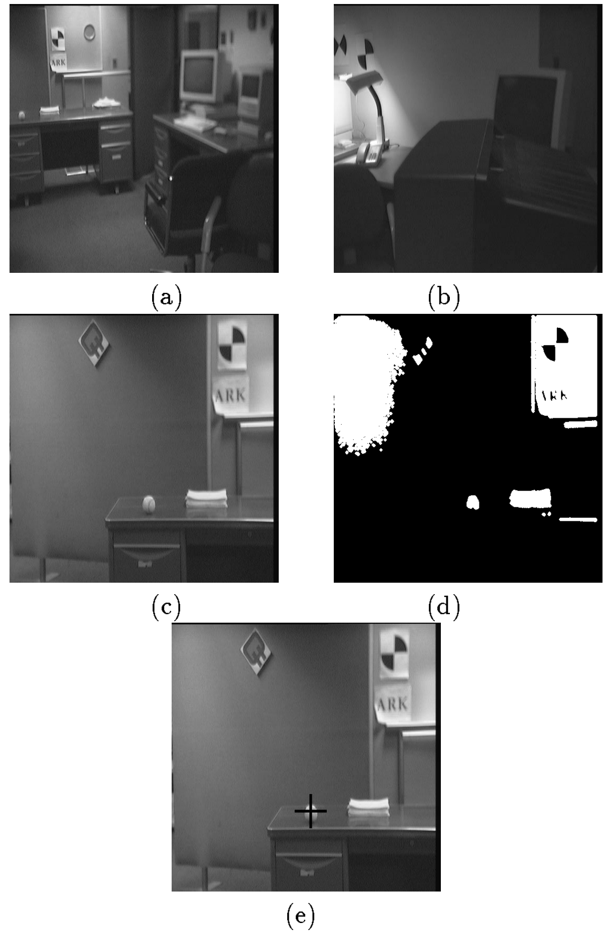


Figure 5: When Dempster-Shafer theory is used to obtain the initial distribution.

(a)(b)(c) The top three images selected by the sensor planning algorithm. The initial target probability distribution is obtained from Dempster-Shafer theory. The baseball is put on table **C**. *Although the ball appeared in the first image (a), the algorithm failed to detect it because it is outside the effective range of the action (camera viewing angle size $41^o \times 39^o$). The third action ((c), camera view angle size $20.7^o \times 19.7^o$) finds the target; (d) The image of (c) after region growing; (e) The result of the image analysis of (d), where the target is detected.*

try Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro Technologies, and Atomic Energy of Canada Limited. John Tsotsos is a Fellow of the Canadian Institute for Advanced Research.

References

Bajcsy, R. 1985. Active perception vs. passive perception. In *Third IEEE Workshop on Vision*, 55–59.

Connel. 1989. *An Artificial Creature*. Ph.D. Dissertation, AI Lab, MIT.

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

Jasiobedzki, P.; Jenkin, M.; Milios, E.; Down, B.; and Tsotsos, J. 1993. Laser eye - a new 3d sensor for active vision. In *Intelligent Robotics and Computer Vision: Sensor Fusion VI. Proceedings of SPIE. vol. 2059*, 316–321.

K., T.; Allen, P.; and Tsai, R. 1992. A survey of sensor planning in computer vision. Technical report, Comp. Sci. Dept., Columbia University.

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

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

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

Schubert, J. 1994. *Cluster-based specification techniques in Dempster-Shafer Theory for an evidential target tracks*. Ph.D. Dissertation, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, SWEDEN.

Tsotsos, J. 1990. Analyzing vision at the complexity level. *The behavioral and brain science* 13:423–469.

Wixson, L. E. 1994. *Gaze Selection for Visual Search*. Ph.D. Dissertation, Department of Computer Science, University of Rochester.

Ye, Y., and Tsotsos, J. K. 1995. Where to look next in 3d object search. In *IEEE International Symposium for Computer Vision*.

Ye, Y., and Tsotsos, J. K. 1996a. Sensor planning in 3d object search: its formulation and complexity. In *The 4th International Symposium on Artificial Intelligenceand Mathematics*.

Ye, Y., and Tsotsos, J. K. 1996b. Tracking with pan, tilt, and zoom camera. In *Preparation*.

Ye, Y. 1997. *Sensor Planning for Object Search*. Ph.D. Dissertation, Department of Computer Science, University of Toronto.

(Bajcsy 1985) (Connel 1989) (Garvey 1976) (Jasiobedzki *et al.* 1993) (K., Allen, & Tsai 1992) (Koopman 1980) (Maver & Bajcsy 1990) (Reece & Shafer 1992) (Schubert 1994) (Tsotsos 1990) (Wixson 1994) (Ye & Tsotsos 1995) (Ye & Tsotsos 1996a) (Ye 1997) (Ye & Tsotsos 1996b)