On the Collaborative Object Search Team: a Formulation

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
Tel: (416) 928-2090 (416) 978-3619
FAX: 416-978-1455

Abstract

This paper gives a formulation of a collaborative object search team and studies the learning, interaction and organization within this multi-agent environment. Each team member is assumed to be a mobile platform equipped with an active camera that can take image of the environment and recognition algorithms that can analyze the resulted image and detect the target object within the image. The goal of the team is to find the target within a give time constraint. In order to do this, the agents must interact and collaborate with each other and must learn and modify the various cooperation styles based on the search results.

Introduction

Many Distributed Artificial Intelligence researchers begin to build agents that can work in a complex, dynamic multiagent domains(Tambe & Rosenbloom 1995). Such domains include virtual theater(Hayes-Roth, Brownston, & Gen 1995), realistic virtual training environments (Pimentel & Teixeira 1994)(Rao *et al.* 1993)(Tambe & Rosenbloom 1995), RoboCup robotic and virtual soccer (Kitano *et al.* 1995) and robotic collaboration by observation (Kuniyoshi *et al.* 1994), etc.

This paper focuses on our recent research effort aimed at developing theories and systems for multiagent object search — the task of searching for a $3D$ object in a $3D$ environment by a group of robots. Constructing multiagent object search systems requires facing many of the hard research challenges such as goal-driven behavior, reactivity, real-time performance, planning, learning, coordination, and spatial and temporal reasoning etc. We believe that given its *real-world nature*, the multiagent object search task reflects many of the key characteristic of other real-world, dynamic multiagent environment.

Here we study the problem of object search by a team formed by *collaborative* robotic agents. By collaborative we mean that the agents are working together collaboratively in order to realize a common goal — find the target. Multi-agent object search system is quite different from the task of object search by a single robotic agent, on which we have done an extensive research and experiments (Ye & Tsotsos 1995) (Ye & Tsotsos 1996a) (Ye 1996) (Ye & Tsotsos 1996b). The multiagent team activities are not merely a union of simultaneous, coordinated individual activities — each agent is not merely searching alone without considering other agent's action. Focusing on the multiagent aspect of the object search task brings to attention a number of challenging issues where learning is involved, such as:

- Interaction: How do agents communicate? How do agents coordinate in a team in order to find the target in the environment as early as possible? How should the patterns of interaction that characterize coordinated behavior be modeled and how do agents learn from the effectiveness of the previous patterns of interaction?
- Knowledge representation and organization: How do search agents represent their local views of the world? How is the local knowledge updated or learned as a consequence of its own action? How do search agents represent their local views of other agents? How do agents revise and learn their beliefs about other agents by exchanging information with other agents? How do agents organize the knowledge about itself, the other agents, and the world such that it can easily integrate its newly learned facts into the representation?
- Planning and Reasoning: How do agents plan their actions based on the knowledge about themselves, the knowledge about other agents, the knowledge learned from the interaction during the search process, and the knowledge learned from the effectiveness of its geometric relationships with other agents?

In this paper, we first formulate the task of multiagent object search and analyze its structure and complexity, then we study various issues in which learning is involved.

Some Basic Concepts about Object Search Agent

In this section, we describe some concepts about the multiagent object search system, such as the $3D$ region to be searched, the model of the search agent, the state parameters of a search agent, the search agent's local knowledge about the world, the operation of a search agent and its cost. We assume throughout this paper that there are totally m search agents a_1, a_2, ..., a_m available in the object search team.

The search region Ω can be of any form, and it is assumed that we know the boundary of Ω and its internal geometric configuration exactly. 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 $i \neq j$. We call each of the element c_i a cell of the environment and we assume in this paper that there are totally n cells. Usually, the search region is an office-like environment, and it is tessellated into little cubes of equal size.

The model of the search agent (Figure 1(a)(b))is assumed to be a mobile platform with a robotic head and a camera that can pan, tilt and zoom (Note: this model is based on the ARK robot and the Laser Eye (Nickerson *et al.* 1993)). The camera's image plane is assumed to be always coincident with its focal plane.

The state s_a of a search agent a is uniquely determined by 7 parameters $(x_a, y_a, z_a, w_a, h_a, p_a, t_a)$, where (x_a, y_a, z_a) is the position of the camera center, w_a, h_a are the width and height of the solid viewing angle of the camera, p_a, t_a are the the camera's viewing direction (Figure 2). The position (x_a, y_a, z_a) can be adjusted by moving the mobile platform. In the situation of a mobile platform, z_a is a fixed constant (the height of the camera), thus, only x_a and y_a are adjustable. The viewing angle size $\langle w_a, h_a \rangle$ can be adjusted by the zoom lens of the camera. Pan and tilt $\langle p_a, t_a \rangle$ can be adjusted by the motors on the robotic head.

The agent's knowledge about the possible target position is specified by a probability distribution function $\mathbf{p}$. The term $\mathbf{p}(a_i, c_j, \tau)$ gives the belief of agent a_i about the probability that the center of the target is within cell c_i at time τ. Before the team search process, each agent member should obtain its own probability distribution function about the search region. This is the agent's local knowledge about the world. This knowledge should satisfy the following constraint

$$\sum_{j=1}^{n} \mathbf{p}(a_i, c_j, \tau_{0}-) + \mathbf{p}(a_i, c_{out}, \tau_{0}-) = 1$$

where τ_0- means the time before the search process, $\mathbf{p}(a_i, c_{out}, \tau_0-)$ refers to agent a_i's belief that the target is outside the search region before the search process. If agent a_i is not able determine whether the target is more likely to be outside the search region or inside the search region, then $\mathbf{p}(a_i, c_{out}, \tau_0-) = 0.5$. If agent a_i has no knowledge about the possible target distribution within the search region, then it assumes uniform distribution

$$\mathbf{p}(a_i, c_j, \tau_{0}-) = \frac{1 - \mathbf{p}(a_i, c_{out}, \tau_{0}-)}{n}$$

for $j = 1, \ldots n$.

An operation $\mathbf{f} = \mathbf{f}(a_i, s_{a_i}, r_{a_i}^{(j)})$ is an action of the search agent a_i within the region Ω, where $r_{a_i}^{(j)}$ is the recognition algorithm used by a_i to detect the target (we assume that each agent can have several recognition algorithms that can be used to detect the target, $r_{a_i}^{(j)}$ means agent a_i's jth recognition algorithm). An operation $\mathbf{f}(a_i, s_{a_i}, r_{a_i}^{(j)})$ for a_i entails two steps: (1) take a *perspective* projection image according to s_{a_i}, and then (2) search the image using the recognition algorithm $r_{a_i}^{(j)}$.

The cost $\mathbf{t}(\mathbf{f})$ for an action $\mathbf{f} = \mathbf{f}(a_i, s_{a_i}, r_{a_i}^{(j)})$ gives the total time needed for a_i to execute the action $\mathbf{f}(a_i, s_{a_i}, r_{a_i}^{(j)})$. It includes (1) manipulate the hardware to the status s_{a_i} specified by $\mathbf{f}$; (2) take a picture using the camera on a_i; (3) update a_i's inner representation of the world; and (4) run the recognition algorithm $r_{a_i}^{(j)}$ specified by $\mathbf{f}$. We assume that: (2) and (3) are same for all the actions; (4) is known for any recognition algorithm and is constant for a given recognition algorithm.

Awareness and Knowledge Adaptation

The awareness means that the agent has knowledge about itself and the world. The knowledge adaptation means that the agent perceives the environment through actions and incorporates the perception results into its own internal representation.

For any agent a_i, it has the inner knowledge about its ability to detect the target by applying a given action $\mathbf{f}$. The detection ability is represented by the detection function $\mathbf{b}(a_i, c_j, \mathbf{f})$, which gives the conditional probability of detecting the target by agent a_i given that the center of the target is located within c_j, and the operation is $\mathbf{f}$. This function is further approximated as the conditional probability of detecting the target by a_i when the target center is located at the center of c_j. For any operation, if the projection of the center of the cell c_i is outside the image, we assume $\mathbf{b}(a_i, c_j, \mathbf{f}) = 0$; if the cell is occluded or too far from the camera or too near to the camera, we also have $\mathbf{b}(a_i, c_j, \mathbf{f}) = 0$. In general (Ye 1996), $\mathbf{b}(a_i, c_j, \mathbf{f})$ is determined by various factors, such as intensity, occlusion, and orientation, etc. The value of the detection function $\mathbf{b}(a_i, c_j, \mathbf{f})$ for $\mathbf{f}$ can be obtained by transforming from a pre-recorded standard detection function for a_i and c_j.

It is obvious that the probability of detecting the target by applying action $\mathbf{f}$ for agent a_i is given by

$$P(\mathbf{f}) = \sum_{j=1}^{n} \mathbf{p}(a_i, c_j, \tau_{\mathbf{f}}) \mathbf{b}(a_i, c_j, \mathbf{f}),$$

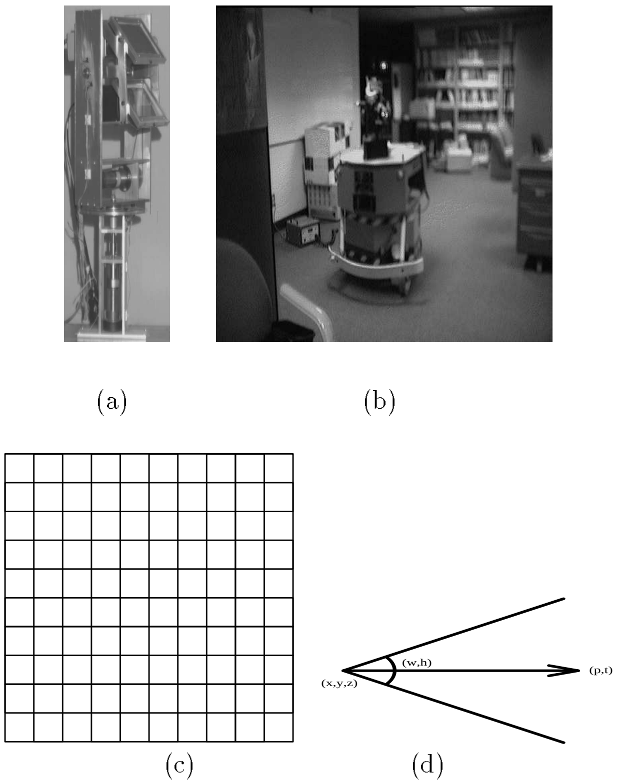

Figure 1: (a) The Laser Eye (the active camera model for the search agent). At the top is the Optech laser range finder; at the bottom is the zoom and focus controlled lens. The two mirrors are used to ensure collinearity of effective optical axes of the camera lens and the range finder. The pan-tilt unit is operated by two DC servocontrolled motors from Micromo Electronics, equipped with gearboxes and optical encoders. (b) The ARK mobile platform (the mobile platform for the search agent). (c) $2D$ illustration of the tessellation of the search region Ω. (d) $2D$ illustration of the state s_a of a search agent a.

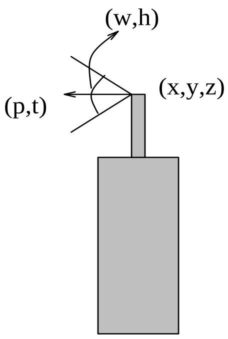

Figure 2: A 2D illustration of the state of the hardware. Where (x, y, z) is the position of the center of the camera, (w, h) is the viewing angle size of the camera, and (p, t) is the viewing direction of the camera.

where $\tau_{\mathbf{f}}$ is the time just before $\mathbf{f}$ is applied. It is obvious that if a cell c_j is outside the field of view of the camera determined by $\mathbf{f}$ or is occluded with respect to the geometry specified by $\mathbf{f}$, then $\mathbf{b}(a_i, c_j, \mathbf{f}) = 0$. We use

$$\Omega(\mathbf{f}) = \{c \mid c \in \Omega \wedge \mathbf{b}(a_i, c, \mathbf{f}(a_i, s_{a_i}, r_{a_i}^{(j)})) \neq 0\}$$

to represent those cells whose detection function values are not zero with respect to a_i and $\mathbf{f}$. $\Omega(\mathbf{f})$ is called the *influence range* for $\mathbf{f}$. Of course,

$$P(\mathbf{f}) = \sum_{c \in \Omega(\mathbf{f})} \mathbf{p}(a_i, c, \tau_{\mathbf{f}}) \mathbf{b}(a_i, c, \mathbf{f}) \qquad (1)$$

For any agent a_i, its beliefs on the possible target positions is represented by $\mathbf{p}(a_i, c, \tau)$. But, this belief keeps changing over time as the agent perceiving the world. If an action by agent a_i finds the target, then a_i will report the result and the task is accomplished. If an action by agent a_i failed to detect the target, then a_i needs to incorporate this result into its *local* knowledge representation related to the target distribution. This is the process of *learning from action*. The agent uses Bayes' formula to update its *local* knowledge base. Let α_j be the event that the center of the target is in cell c_j, and α_{out} 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 $\mathbf{f}$ by agent a_i, the recognizer successfully detects the target. Then $P(\neg\beta \mid \alpha_j)$ gives the probability of not detecting the target by action $\mathbf{f}$ of agent a_i given that the target center is within cube c_j. Since $\mathbf{b}(a_i, c_j, \mathbf{f})$ gives the probability of detecting the target by action $\mathbf{f}$ given that the target center is within cube c_j. We have

$$P(\neg\beta \mid \alpha_j) = 1 - \mathbf{b}(a_i, c_j, \mathbf{f})$$

It is obvious that $P(\alpha_j \mid \neg\beta)$ gives the probability that the target center is within the cube c_j given that $\mathbf{f}$ failed to detect the target. If we represent the updated probability for cube c_j as $\mathbf{p}(a_i, c_j, \tau_{\mathbf{f}+})$, where $\tau_{\mathbf{f}+}$ is the time after $\mathbf{f}$ is applied. Then we should have

$$\mathbf{p}(a_i, c_j, \tau_{\mathbf{f}+}) = P(\alpha_j \mid \neg\beta)$$

Since the events $\alpha_1, \ldots, \alpha_n, \alpha_{out}$ are mutually complementary and exclusive, from Bayes' formula, we get

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

for $i = 1, \ldots, n, out$

Thus, by replacing $P(\alpha_i \mid \neg\beta)$ with $\mathbf{p}(a_i, c_j, \tau_{\mathbf{f}+})$, $P(\alpha_i)$ with $\mathbf{p}(a_i, c_j, \tau_{\mathbf{f}})$, and $P(\neg\beta \mid \alpha_i)$ with $1 - \mathbf{b}(a_i, c_j, \mathbf{f})$, we can obtain the following probability updating rule

$$\mathbf{p}(a_i, c_j, \tau_{\mathbf{f}+}) \leftarrow \frac{\mathbf{p}(a_i, c_j, \tau_{\mathbf{f}})(1 - \mathbf{b}(a_i, c_j, \mathbf{f}))}{\sum_{k=1}^{n,out} \mathbf{p}(a_i, c_k, \tau_{\mathbf{f}})(1 - \mathbf{b}(a_i, c_k, \mathbf{f}))}$$

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

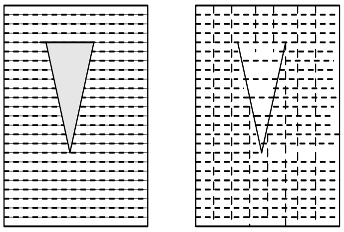

Figure 3: A 2D illustration of the belief updating process. When an agent applied an action in the environment but failed to detect the target, it's belief on the target distribution within the influence range of the action is decreased and its belief on the target distribution outside the influence range of the action is increased. This phenomena is modeled by Bayes Law.

The Agent's Sensing Ability

In this section we discuss how to obtain the detection function value for each agent and explain the concept of *effective volume* for an action of a given agent.

The value of the detection function $\mathbf{b}(a_i, c_j, \mathbf{f})$ for $\mathbf{f}$ can be obtained by transforming from a pre-recorded standard detection function for a_i and c_j. The standard detection function $\mathbf{b_0}((\theta, \delta, l), \langle r, w, h\rangle)$ gives a measure of the detecting ability of the recognition algorithm a when there is no previous action been applied. Where $\langle w, h\rangle$ 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 camera coordinate system. The value of $\mathbf{b_0}((\theta, \delta, l), \langle r, w, h\rangle)$ can be obtained by experiments. 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 $\langle w_0, h_0\rangle$. Those of other sizes can be approximately transformed to those of size $\langle w_0, h_0\rangle$. Suppose (θ, δ, l) is the target position for angle size $\langle w, h\rangle$, we want to find the value $(\theta_0, \delta_0, l_0)$ for angle size $\langle w_0, h_0\rangle$ such that

$$\mathbf{b_0}((\theta_0, \delta_0, l_0), \langle r, w_0, h_0\rangle) \approx \mathbf{b_0}((\theta, \delta, l), \langle r, w, h\rangle)$$

To guarantee this, the images taken with parameter $\langle \theta_0, \delta_0, l_0, w_0, h_0\rangle$ and $\langle \theta, \delta, l, w, h\rangle$ should be almost same. Thus, the area and position of the projected target image on the image plane should be almost same for both images, we get

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

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

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

When the configuration of two operations are very similar, they might correlated with each other (refer to (Ye 1996) for detail). Repeated actions are avoided during the search process. When independence is assumed, $\mathbf{b}(a_i, c_j, \mathbf{f})$ is calculated as following. First, calculate the corresponding (θ, δ, l) of the center of c_j with respect to operation $\mathbf{f}$ of agent a_i. Second, transform (θ, δ, l) into the corresponding $(\theta_0, \delta_0, l_0)$ of angle size $\langle w_0, h_0\rangle$. Third, retrieve the detection value from the look up table, or get the detection value from a formula.

Now we explain the concept of effective volume for a given action. 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 totally brought into the field of view of the camera and the features be detected within certain precision, can the recognition algorithm be expected to function correctly. So, for an agent's action with a given recognition algorithm and a fixed viewing angle, the probability of successfully recognize the target is high only when the target is within a certain distance range. We call this range the *effective range* and the viewing volume within this range the *effective volume*. For a given action, only those cubes that are within the *effective volume* can be examined with high detection probability. Figure 4 gives a $2D$ illustration of the effective volume for a given action.

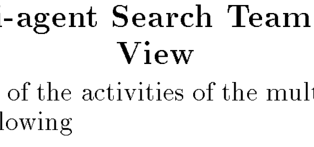

Figure 4: $2D$ illustration of the effective volume of a given action (the shaded area).

The Multi-agent Search Team: a Global View

A global view of the activities of the multiagent search team is as following

- Before the search process, each agent a_i proposes an initial target distribution $(\mathbf{p}(a_i, c, \tau_0-)$ for all $c \in \Omega)$ according to its own perception of the world (Figure 5).
- The team forms the common initial target distribution $(\mathbf{p}(c, \tau_0)$ for all $c \in \Omega)$ by combining the initial distribution of all the agents together (Figure 6).

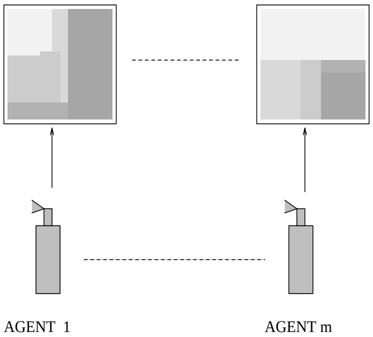

Figure 5: Each agent has a initial target distribution.

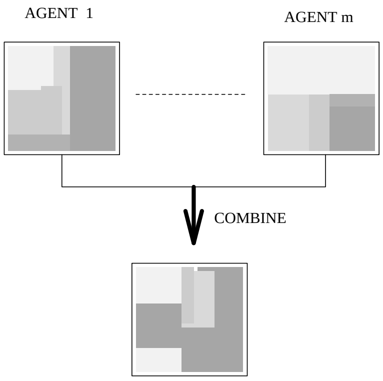

Figure 6: Common initial target distribution formation.

- Each agent a_i uses the common initial distribution $\mathbf{p}(c, \tau_0)$ as its own initial distribution $\mathbf{p}(a_i, c, \tau_0)$ used during the search process (Figure 7). That is, performing the substitution

$$\forall c, \mathbf{p}(a_i, c, \tau_0) \leftarrow \mathbf{p}(c, \tau_0)$$

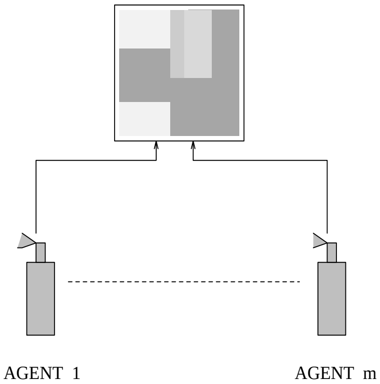

Figure 7: Initial distribution used before the search process for each agent.

- The selection and execution of actions for different agents goes in parallel (Figure 8). But for each agent, its action selection and execution must be in sequence. This means that for a single agent, it can select and execute the next action only after the current action is finished.

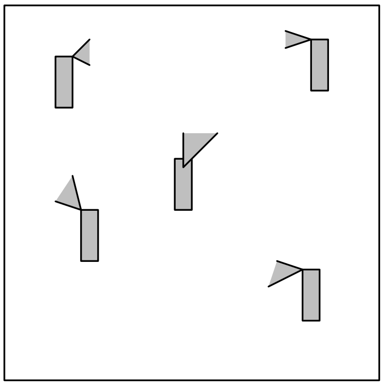

Figure 8: Agents search for the target together at the same time.

  - Each agent selects its own action based on its own knowledge and the knowledge *learned* from interaction and communication with other agents.
  - If the selected action succeeds, the corresponding agent announces the success result immediately and the team search process terminates.
  - If the selected action fails, the corresponding agent will broadcast the failure news to the rest of the agents, together with the action's state parameters and effective volume. Then the corresponding agent updates its own knowledge base about

the target distribution and starts selecting its next action.

- The team search process will terminate when the available time is used up.

Let K be the total time that is available for search. Let Ω_{a_i} be the set of all the actions which can be selected by agent a_i $(1 \leq a_i \leq m)$. Suppose

$$\mathbf{F}_{a_i} = \Big\{\mathbf{f}(a_i, s_{a_i}^{(1)}, r_{a_i}^{(j_1)}), \mathbf{f}(a_i, s_{a_i}^{(2)}, r_{a_i}^{(j_2)}), \ldots, \mathbf{f}(a_i, s_{a_i}^{(N_{a_i})}, r_{a_i}^{(j_{N_{a_i}})})\Big\}$$

is the set of actions selected by agent a_i $(1 \leq a_i \leq m)$ from Ω_{a_i} during the search process. Then we call $\mathbf{F}_{a_i}$ the *effort allocation* for agent a_i. The *effort allocation for the search team* is the union of the effort allocations of all the agents of the team

$$\mathbf{F} = \mathbf{F}_{a_1} \bigcup \mathbf{F}_{a_2} \bigcup \cdots \bigcup \mathbf{F}_{a_m}$$

Let $P[\mathbf{F}]$ be the probability of detecting the target by the team effort allocation $\mathbf{F}$, K is the total time that can be used by the team. Then the task of object search by a multiagent team can be defined as finding a team effort allocation $\mathbf{F}$ such that $P[\mathbf{F}]$ is maximized within the time constraint K.

The Initial Common Target Distribution

Before the search process, each agent has some knowledge about the whereabouts of the target. This knowledge can come from different sources. Because the object search team is collaborative, the team should form a common initial target distribution by considering each agent's knowledge. One way to get the common initial distribution is to use the Dempster-Shafer theory. Here we simply generate the initial common target distribution $\mathbf{p}(c, \tau_0)$ (for $c \in \Omega$) by forming a weighted sum of the probability distributions of all the agents. This means that for all $c \in \Omega$, perform the following operation:

$$\mathbf{p}(c, \tau_0) \leftarrow \sum_{k=1}^{m} w_k \mathbf{p}(a_k, c, \tau_0 -)$$

Where $w_1, \ldots, w_m$ are weights that satisfy

$$\sum_{k=1}^{m} w_k = 1 \quad and \quad w_k \geq 0,\ 1 \leq k \leq m$$

It is obvious that the newly generated target distribution satisfies

$$\begin{aligned}
\sum_{i=1}^{n,out} \mathbf{p}(c_i, \tau_0) &= \sum_{i=1}^{n,out} \Big(\sum_{k=1}^{m} w_k \mathbf{p}(a_k, c_i, \tau_0 -)\Big) \\
&= \sum_{k=1}^{m} \Big(w_k \sum_{i=1}^{n,out} \mathbf{p}(a_k, c_i, \tau_0 -)\Big) \\
&= \sum_{k=1}^{m} w_k \\
&= 1
\end{aligned}$$

In order to make this strategy work effectively, it is important to obtain a set of values $w_1, \ldots, w_m$ such that they can reflect the relative credibility of the target distributions from different agents. If the multiagent team has not been working together before, then

$$w_1 = \ldots = w_m = \frac{1}{m}.$$

This is because that the team does not know the relative reliability of the agents.

In general, more reasonable value of w_k can be obtained by *training and learning.*

Each set of training is obtained by running the team search process for m cases, each case uses a different agent's initial probability distribution as the initial global probability distribution (Note: there is no weighted sum operation).

Suppose the time needed to detect the target for a case when agent k's initial probability distribution is used is minimum, then we say that agent k wins this set of training. The result illustrates that agent k's knowledge sources are more reliable that other agent's knowledge sources for this training.

So, each set of training can select an agent whose initial knowledge is more reliable. Suppose we have performed Q set of training, then the value of w_k for agent k will be the number of trainings won by agent k divided by Q.

Detection Probability for an Effort Allocation

We derive a general expression to calculate $P[\mathbf{F}]$ in this section in a global view. By global view we mean that the probability of detecting the target by applied actions and the probability distribution update after each action is applied are tracked in a global consistent fashion. In a global consistent fashion means that if an action can detect the target, then the success news is broadcast at the moment this action is selected. If an action cannot find the target, then the environment is updated at the moment this action is selected.

In order to get a general expression, let us first considering a special case by assuming that the actions in $\mathbf{F}$ are applied in sequence (Note: in real application, the actions of different agents can be executed at the same time).

Let

$$\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_q\}$$

where

$$q = N_{a_1} + \ldots + N_{a_m} = \mid \mathbf{F}_{a_1} \mid + \mid \mathbf{F}_{a_2} \mid + \ldots + \mid \mathbf{F}_{a_m} \mid$$

and

$$\tau_{\mathbf{f}_1} < \tau_{\mathbf{f}_2} < \ldots < \tau_{\mathbf{f}_q}$$

The correspondence of the action index for $\mathbf{f}_i$ with the actions executed by each agent is illustrated in the following

$$\mathbf{F} = \Big\{ \overbrace{\mathbf{f}_1, \ldots, \mathbf{f}_{N_{a_1}}}^{\mathbf{F}_{a_1}}, \ldots, \overbrace{\mathbf{f}_{N_{a_1}+\cdots+N_{a_{m-1}}+1}, \ldots, \mathbf{f}_q}^{\mathbf{F}_{a_m}} \Big\}$$

The search process goes as following: at the beginning, $\mathbf{f}_1$ is selected and applied, the probability distribution used by $\mathbf{f}_1$ is the initial common target distribution. If $\mathbf{f}_i$ $(1 \leq i \leq q)$ succeed, then the news is broadcast to all agents and the search process terminate; if $\mathbf{f}_i$ failed, then the probability distribution is updated, and this updated probability distribution will be used by $\mathbf{f}_{i+1}$. It is clear that the expected probability of detecting the target by allocation $\mathbf{F}$ is:

$$\begin{aligned} P[\mathbf{F}] &= P(\mathbf{f}_1) + [1 - P(\mathbf{f}_1)]P(\mathbf{f}_2) \\ &\quad + \ldots \\ &\quad + \{ \prod_{i=1}^{k-1} [1 - P(\mathbf{f}_i)] \} P(\mathbf{f}_q) \end{aligned} \tag{2}$$

Formula (2) is not intuitive in finding regularities about P[$\mathbf{F}$] and it is not appropriate in defining the probability of detecting the target by an effort allocation when the actions of different agents can be executed in parallel. In order to obtain a representation of the detection probability in general sense, we need to do some algebraic transformations.

For any operation $\mathbf{f} \in \mathbf{F}$, we define its influence range as $\Omega(\mathbf{f}) = \{c \mid \mathbf{b}(c, \mathbf{f}) \neq 0\}$, where $\mathbf{b}(c, \mathbf{f}) = \mathbf{b}(a, c, \mathbf{f})$, a is the agent executing action $\mathbf{f}$. In the following discussion, we will use $\mathbf{b}(c, \mathbf{f})$ to represent $\mathbf{b}(a, c, \mathbf{f})$, with a understanding that the agent executing $\mathbf{f}$ is a. Please note that $\Omega(\mathbf{f})$ is different from the effective volume of $\mathbf{f}$, $\Omega(\mathbf{f}) \neq EV(\mathbf{f})$. The initial probability distribution is denoted as $\mathbf{p}^{[0]}(c_1)$, $\mathbf{p}^{[0]}(c_2)$, ..., $\mathbf{p}^{[0]}(c_n)$, $\mathbf{p}^{[0]}(c_{out})$. Of course, we have $\mathbf{p}^{[0]}(c_i) = \mathbf{p}(c_i, \tau_0)$, for $i = 1, \ldots, n, out$. After the application of the operation $\mathbf{f}_1$, the distribution is denoted by $\mathbf{p}^{[1]}(c_1)$, $\mathbf{p}^{[1]}(c_2)$, ..., $\mathbf{p}^{[1]}(c_n)$, $\mathbf{p}^{[1]}(c_{out})$. Generally, after the application of the operation $\mathbf{f}_i$, the distribution is denoted by $\mathbf{p}^{[i]}(c_1)$, $\mathbf{p}^{[i]}(c_2)$, ..., $\mathbf{p}^{[i]}(c_n)$, $\mathbf{p}^{[i]}(c_{out})$, 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

$$P^{[0]}(\mathbf{f}_i) = \sum_{j=1}^{n} \mathbf{p}^{[0]}(c_j) \mathbf{b}(c_j, \mathbf{f_i})$$

After some calculations, we can obtain the general expression of the target distribution as following

Lemma 1. For allocation $\mathbf{F}$, we have

$$\mathbf{p}^{[k]}(c) = \frac{\mathbf{p}^{[0]}(c)[1 - \mathbf{b}(c, \mathbf{f}_1)] \ldots [1 - \mathbf{b}(c, \mathbf{f}_k)]}{(1 - P(\mathbf{f}_1))(1 - P(\mathbf{f}_2)) \ldots, (1 - P(\mathbf{f}_k))} \tag{3}$$

By using *Lemma 1*, we can obtain the general expression for the detection probability for any action in $\mathbf{F}$.

Lemma 2 For allocation $\mathbf{F}$, we have

$$\begin{aligned} P(\mathbf{f}_k) &= \frac{1}{(1 - P(\mathbf{f}_1)) \ldots (1 - P(\mathbf{f}_{k-1}))} \Big\{ P^{[0]}(\mathbf{f}_k) \\ &\quad + (-1)^1 \sum_{i_1=1}^{k-1} \Big(\sum_{c \in \Omega(\mathbf{f}_{i_1} \mathbf{f}_k)} \mathbf{p}^{[0]}(c) \mathbf{b}(c, \mathbf{f}_{i_1}) \mathbf{b}(c, \mathbf{f}_k) \Big) \\ &\quad + (-1)^2 \sum_{1 \leq i_1 < i_2 \leq k-1} \\ &\quad \Big(\sum_{c \in \Omega(\mathbf{f}_{i_1} \mathbf{f}_{i_2} \mathbf{f}_k)} \mathbf{p}^{[0]}(c) \mathbf{b}(c, \mathbf{f}_{i_1}) \mathbf{b}(c, \mathbf{f}_{i_2}) \mathbf{b}(c, \mathbf{f}_k) \Big) \\ &\quad + (-1)^3 \sum_{1 \leq i_1 < i_2 < i_3 \leq k-1} \Big(\sum_{c \in \Omega(\mathbf{f}_{i_1} \mathbf{f}_{i_2} \mathbf{f}_{i_3} \mathbf{f}_k)} \\ &\quad \mathbf{p}^{[0]}(c) \mathbf{b}(c, \mathbf{f}_{i_1}) \mathbf{b}(c, \mathbf{f}_{i_2}) \mathbf{b}(c, \mathbf{f}_{i_3}) \mathbf{b}(c, \mathbf{f}_k) \Big) \\ &\quad + \ldots \\ &\quad + (-1)^{k-1} \sum_{c \in \Omega(\mathbf{f}_1 \ldots \mathbf{f}_{k-1} \mathbf{f}_k)} \\ &\quad \mathbf{p}^{[0]}(c) \mathbf{b}(c, \mathbf{f}_1) \mathbf{b}(c, \mathbf{f}_2) \ldots \mathbf{b}(c, \mathbf{f}_{k-1}) \mathbf{b}(c, \mathbf{f}_k) \Big\} \end{aligned}$$

for $k = 2, \ldots, q$.

By using Lemma 2, we can prove the following important lemma.

Lemma 3 For allocation $\mathbf{F}$, we have

$$\begin{aligned} P(\mathbf{F}) &= \sum_{i=1}^{q} \Big(\sum_{j=1}^{n} \mathbf{p}(c_j, \tau_0) \mathbf{b}(c_j, \mathbf{f}_i) \Big) \\ &\quad + (-1)^{2+1} \sum_{1 \leq i_1 < i_2 \leq q} \\ &\quad \Big(\sum_{c \in \Omega(\mathbf{f}_{i_1} \mathbf{f}_{i_2})} \mathbf{p}(c, \tau_0) \mathbf{b}(c, \mathbf{f}_{i_1}) \mathbf{b}(c, \mathbf{f}_{i_2}) \Big) \\ &\quad \ldots \end{aligned}$$

$$+(-1)^{r+1} \sum_{1 \leq i_1 < i_2 < \ldots < i_r \leq q} \Big(\sum_{c \in \Omega(\mathbf{f}_{i_1}\mathbf{f}_{i_2}\ldots\mathbf{f}_{i_r})} \mathbf{p}(c, \tau_0)\mathbf{b}(c, \mathbf{f}_{i_1})\mathbf{b}(c, \mathbf{f}_{i_2})\ldots\mathbf{b}(c, \mathbf{f}_{i_r})\Big)$$

$$\ldots$$

$$+(-1)^{q+1} \Big(\sum_{c \in \Omega(\mathbf{f}_1\mathbf{f}_2\ldots\mathbf{f}_q)} \mathbf{p}(c, \tau_0)\mathbf{b}(c, \mathbf{f}_1)\ldots\mathbf{b}(c, \mathbf{f}_q)\Big) \quad (4)$$

Lemma 3 explicitly reveals a very important property of $P[\mathbf{F}]$: the order of actions been applied does not have any influence on the value of $P[\mathbf{F}]$. This agrees with our common knowledge. Because whether an action $\mathbf{f}_i = \mathbf{f}(a_i, s_{a_i}, r^j_{a_i})$ can detect the target is determined only by the agent a_i, the state parameters s_{a_i} of a_i, the recognition algorithm $r^j_{a_i}$ used to analyze the image taken by a_i, the real target position and the real situation (lighting, background, etc.) surrounding the target. Since these factors will not change with time, the time when the action $\mathbf{f}(a_i, s_{a_i}, r^j_{a_i})$ is applied has no effect on whether this action can detect the target. If action $\mathbf{f}_i$ detects the target in one application order in the effort allocation $\mathbf{F}$, then, it will detect the target in another application order in the effort allocation $\mathbf{F}$; if action $\mathbf{f}_i$ fails to detect the target in one application order, then, it will fail to detect the target in another application order.

Because of Formula (4)'s advantage discussed above, it is used as the general definition of the expected probability of detecting the target by an effort allocation $\mathbf{F}$ in a multiagent search team.

So, in the real team search process where actions can be applied concurrently by different agents, we will use Formula (4) to calculate the expected probability of detecting the target by the given effort allocation.

The Task of Multiagent Search and its Complexity: a Global View

The object search task for a multiagent object search team $\{a_1, \ldots, a_m\}$ can be defined as the task for each agent a_i to find a set of operations $\mathbf{F}_{a_i}$, such that the expected probability of detecting the target $P[\mathbf{F}]$ by the effort allocation $\mathbf{F}$ (formed by the union of the effort allocations of all the agents of the team

$$\mathbf{F} = \mathbf{F}_{a_1} \bigcup \mathbf{F}_{a_2} \bigcup \ldots \bigcup \mathbf{F}_{a_m}$$

is maximized with the constraint that the cost for the effort allocation for each agent a_i is less than or equal to K:

$$\mathbf{t}(\mathbf{f}_{a_1^{(1)}}) + \mathbf{t}(\mathbf{f}_{a_1^{(2)}}) + \ldots + \mathbf{t}(\mathbf{f}_{a_1^{(N_{a_1})}}) \leq K$$

$$\mathbf{t}(\mathbf{f}_{a_2^{(1)}}) + \mathbf{t}(\mathbf{f}_{a_2^{(2)}}) + \ldots + \mathbf{t}(\mathbf{f}_{a_2^{(N_{a_2})}}) \leq K$$

$$\vdots$$

$$\mathbf{t}(\mathbf{f}_{a_q^{(1)}}) + \mathbf{t}(\mathbf{f}_{a_q^{(2)}}) + \ldots + \mathbf{t}(\mathbf{f}_{a_q^{(N_{a_q})}}) \leq K$$

where $\mathbf{f}_{a_i^{(j)}}$ $(1 \leq j \leq N_{a_i})$ refers to the jth actions of agent a_i.

This is a NP-hard problem. We can use prove by restriction to prove this result. By assuming that $m = 1$, the task becomes for agent a_1, find an effort allocation such that $P[F]$ is maximized and

$$\mathbf{t}(\mathbf{f}_{a_1^{(1)}}) + \mathbf{t}(\mathbf{f}_{a_1^{(2)}}) + \ldots + \mathbf{t}(\mathbf{f}_{a_1^{(N_{a_1})}}) \leq K$$

Since this simplified problem ($m = 1$) is NP-hard (please refer to (Ye & Tsotsos 1996a) for proofs), thus the multiagent object search problem is also NP-hard. Because of this, it is impractical to design a team search strategy that can always generate an effort allocation that maximizes the probability of detecting the target. Thus, the goal becomes for each member of the team to corporate with each other so as to find the target as early as possible.

Learning, Interaction and Planning

The multiagent object search task described in this paper is an ideal task to study the relationships between learning, interaction and planning in multiagent environments.

Ideally, an agent should plan its actions based on a perfect knowledge about the current situation of the world (such as the target distribution), the ability of itself and all the other agents, and the activities of the whole team performed so far. But because of the limited computation power, limited memory, and limited communication power, an agent is not able to track everything. For example, it is not possible for an agent to update its *own* knowledge base every time when *other* agents executed an action and failed to detect the target. This is because the probability updating process is time consuming. In the multiagent search team, the agent updates the probability distribution only after its *own* action is applied.

But, it is also not efficient if an agent just performs searching task only based on its own knowledge without considering other agents actions. Because this may cause many extraneous activities. Thus, a certain among of communications is essential for the team search task. The question is how much communications are needed and how much memory the agents must maintain in order to keep a satisfactory coordination behavior for the whole team.

An agent's internal knowledge can be divided into two parts: local knowledge and global knowledge. The local knowledge is the agent's knowledge about itself and the influence of its own action on the world. The global knowledge is its knowledge about other agents and the effects of the actions of other agents on the world. Each agent should have a perfect local knowledge, but it is not able to have a perfect global knowledge because of various limitations. Usually the global knowledge is obtained by interaction, communication and *learning* from other agents during the search process. A search agent's planning system is influenced by both its local knowledge and the *learned* global knowledge.

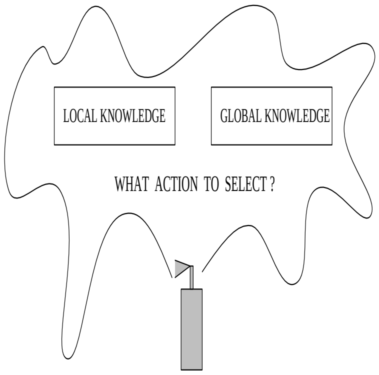

Figure 9: A search agent's planning system is influenced by both its local knowledge and the global knowledge.

Learning and Where to Look Next

The task of "where to look next" for an agent a_i can be defined as: for a fixed position (x_c, y_c, z_c), select the state parameters (w, h, p, t, r) for the next action $\mathbf{f}$ of agent a_i such that the probability of detecting the target is maximized.

The Local Knowledge If only local knowledge is considered, the task is the same as object search by a single agent, on which we have done an extensive research. We have designed a strategy to select the best action such that $P(\mathbf{f}) = \sum_{c \in \Omega(\mathbf{f})} \mathbf{p}(a_i, c, \tau_\mathbf{f}) \mathbf{b}(a_i, c, \mathbf{f})$ is maximized (Ye & Tsotsos 1995).

But in multiagent environment, the $P(\mathbf{f})$ calculated this way does not perfectly reflect the real probability of detecting the target by action $\mathbf{f}$. Because the knowledge distribution $\mathbf{p}(a_i, c, \tau_\mathbf{f})$ used in the calculation is agent a_i's local knowledge. This distribution does not reflect the influences of the actions of the other agents (because agent a_i is not able to update the probability distribution whenever other agents apply an action on the environment). So, even though a_i thought that the region within the effective volume $EV(\mathbf{f})$ of $\mathbf{f}$ has high probability, it might not be so, since other agents may have already examined this region extensively. From the global point of view, the probability within $EV(\mathbf{f})$ may not be so high as a_i thought at the moment $\tau_\mathbf{f}$.

Thus, a_i needs to integrate the influences of other agents' applied actions into its action planning process.

The Learned Global Knowledge Although each agent a_i is not able to update its target distribution whenever other agent executes an action, it is able to record a *crude* information about how the cells of the region have been checked by other agents.

As discussed before, for a given action $\mathbf{f}$, only those cubes that are within $\mathbf{f}$'s effective volume $EV(\mathbf{f})$ can be checked with high confidence. Thus, whenever an action $\mathbf{f}$ is executed, only the examination situation of those cubes that are within $EV(\mathbf{f})$ need to be recorded, the examination situation of other cubes need not to be recorded. To do this, each agent a_i maintains the Examination Situation Map $E(a, c)$. $E(a, c)$ gives the number of times that the cell c falls into the effective volume of the actions executed by agent a. Each agent a_i must maintain the Examination Situation Map for every other agent a_j $(1 \leq j \leq m, j \neq i)$ of the search team and every cell c $c \in \Omega$ of the search region Ω (Figure 10).

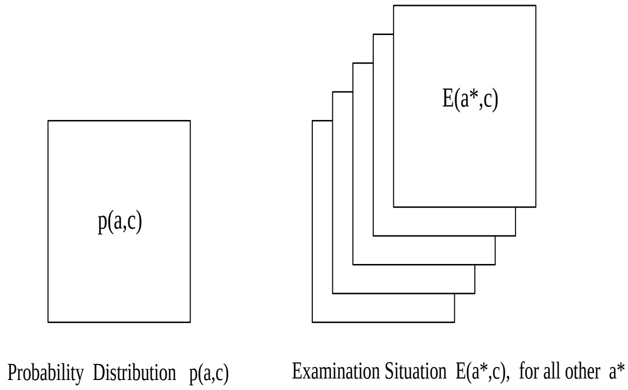

Figure 10: An agent's local and global knowledge.

For each agent, the Examination Situation Map is maintained through a process of *learning by communication*. During the search process, as soon as an agent a_i selected an action $\mathbf{f}(a_i, s_{a_i}, r^j_{a_i})$ to execute, it will broadcast all the cells that are within the effective volume $EV(\mathbf{f})$ of action $\mathbf{f}$ to all the other members of the team. Upon receiving the broadcast from a_i, each other team members a_j $(1 \leq j \leq m, j \neq i)$ will update its Examination Situation Map as following:

$$\forall c \in EV(\mathbf{f}) \text{ perform } E(a_i, c) \leftarrow E(a_i, c) + 1.$$

The Influence of the Learned Global Knowledge on Planning The next action $\mathbf{f}$ are selected by both local knowledge and external stimuli. The local knowledge is the probability of detecting the target $P(\mathbf{f})$ by action $\mathbf{f}$. The external stimuli $ES(\mathbf{f})$ is derived from the *learned global knowledge*, which is represented by the Examination Situation Map. The objective function for agent a_i is: select an action $\mathbf{f}$ that maximizes the following weighted sum

$$w_1 P(\mathbf{f}) - w_2 ES(\mathbf{f}) \tag{5}$$

where w_1, w_2 are weights, and $ES(\mathbf{f})$ are defined as following

$$ES(\mathbf{f}) = \sum_{c \in \Omega(\mathbf{f})} \sum_{j \neq i}^{m} E(a_j, c) \tag{6}$$

Learning, Coordination and Where to Move Next

The goal of "where to move next" for an agent a_i in a multiagent object search team can be defined as: select the agent position (x_c, y_c) for agent a_i such that the chances of detecting the target for agent a_i and the whole team is maximized.

The Local Knowledge and the Learned Global Knowledge If only local knowledge is considered, then the task is the same as the "where to move next" task for a single agent. The strategy of selecting the next agent position is straight forward. For each candidate position (x, y), there is a range of space that can be checked by the camera without occlusion. We call this range the sensed sphere $SS(x, y)$ for position (x, y) (please refer to (Ye & Tsotsos 1995) (Ye 1996) for more detail). The sum of the probability of all the cells within the corresponding sensed sphere is called the sensible probability for this position $S_{prob}(x, y)$. The task is to find a position (x, y) such that the sensible probability is maximized.

Similar to the previous discussion, the local knowledge $S_{prob}(x, y)$ does not perfectly reflect the real sensible probability for the position (x, y). The agent should also include the *learned global knowledge* $E_s(x, y)$ into its decision making. $E_s(x, y)$ gives the measurement about how much the region within the sensed sphere of position (x, y) has been checked by other agents

$$E_S(x, y) = \sum_{c \in SS(x,y)} \sum_{j \neq i} E(a_j, c) \tag{7}$$

Coordination For the "where to move next" task of the multiagent search team, coordination among different agents should also be taken into consideration. For example, if all the agents find a position (or positions near this one) has the highest sensible probability, then when coordination is not considered, all the agents will be gathered around this position to perform actions. This is not desirable for the team as a whole because there is no agents to check other regions of the search space and many of the actions may be redundant. So, coordination is needed for agents in a multiagent team to decide their positions. From the global point of view, the agents should be distributed evenly across the search space. Thus, a coordination factor $D(x, y)$ should be considered in planning the next action. $D(x, y)$ specifies how far the new position (x, y) is to other agents of the team (Figure 11).

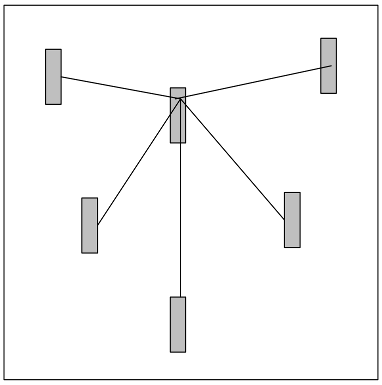

Figure 11: The relative positions for different agents are also very important for efficient search. The positions of the agents in the team should be carefully selected such that the extraneous activities are avoided.

Learning, Coordination and Planning The next position (x, y) for agent a_i should be selected by integrating the local knowledge, the learned global knowledge, and the coordination factor together. The goal is to maximize the following weighted term

$$w_1 S_{prob}(x, y) - w_2 E_s(x, y) + w_3 D(x, y) \tag{8}$$

where w_1, w_2, and w_3 are weights, and

$$D(x, y) = \sum_{j=1, j \neq i}^{m} [w_{ij} \sqrt{(x - x_j)^2 + (y - y_j)^2}] \tag{9}$$

The value of w_{ij} $(1 \leq j \leq m,\ j \neq i)$ is used to balance the importance of the distance between different agents with respect to a_i. These value must satisfy

$$\sum_{j=1, j \neq i}^{m} w_{ij} = 1$$

If there is no preference, then we set

$$w_{12} = \ldots = w_{1m} = \frac{1}{m-1}$$

A better values can be obtained by *learning from experience* for agent a_i. The method is to include each success experience into the values of w_{ij}.

Suppose during a search process, the weights for agents a_i are w_{ij} $(1 \leq j \leq m,\ i \neq j)$. When this process finally finds the target, we can get the distances l_{ij} $(1 \leq j \leq m,\ i \neq j)$ between agent a_i and other agents a_j $(1 \leq j \leq m,\ i \neq j)$. The relative information about these distance should be incorporated into the future weights by updating the current weights w_{ij}. For example, if the distance between a_i and another agent a_j is the longest among all the other agents when the target is detected, then the updated weights should give more weight to w_{ij}, such that during the next team search process there is a factor in the new weights that encourage a bigger distance between a_i and a_j. Suppose

$$l_{sum} = \sum_{i=1, i \neq j}^{m} l_{ij}$$

We suggest to use the following updating rules to incorporate the new distance information:

$$w_{ij} \leftarrow \delta w_{ij} + (1 - \delta) \frac{l_j}{l}$$

(for all $1 \leq j \leq m$, $j \neq i$). Where $0 \leq \delta \leq 1$ is a constant. It is easy to know that the sum of the updated w_{ij} is still 1.

Conclusion

In this paper, we formulate the multiagent object search task and prove that this task is NP-Complete from a global point of view. We analyze various issues that learning is involved in a multiagent object search system, including: (A) the local knowledge updating rule for an agent; (B) a method to obtain the initial common target distribution; (C) the learning of the global knowledge through interaction and communication with other agents; (D) learning and interaction so as to improve the "where to look next" task; (E) learning and coordination so as to improve the "where to move next" task.

The work reported in this paper is only in its primitive stage. In future work we will endeavor to develop a more detailed interaction and learning theory for the multiagent object search task. Given the highly cooperative nature of this task, we believe that a further study of this problem will reveal a deeper relationships between learning, interaction and organizations in multiagent environments.

Acknowledgments

The authors would like to thank both reviewers for their valuable comments. This work is supported by a research fellowship from IBM Canada Centre For Advanced Studies and by the ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates Inc., Industry Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro Technologies, and Atomic Energy of Canada Limited. John K. Tsotsos is a fellow of the Canadian Institute for Advanced Research.

References

Hayes-Roth, B.; Brownston, L.; and Gen, R. 1995. Multiagent collabration in directed improvisation. In *Proceedings of the International Conference on Multi-Agent Systems (ICMAS-95)*.

Kitano, H.; Asada, M.; Kuniyoshi, Y.; Noda, I.; and Osawa, E. 1995. The robot world cup initiative. In *Proceedings of IJCAI-95 Workshop on Entertainment and AI/Alife*.

Kuniyoshi, Y.; Rougeaux, S.; Ishii, M.; Kita, N.; Sakane, S.; and Kakikura, M. 1994. Cooperation by observation: the framework and the basic task pattern. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Nickerson, S.; Jenkin, M.; Milios, E.; Down, B.; Jasiobedzki, P.; Jepson, A.; Terzopoulos, D.; Tsotsos, J.; Wilkes, D.; Bains, N.; and Tran, K. 1993. Ark: Autonomous navigation of a mobile robot in a known environment. In *Intelligent Autonomous Systems-3*, 288–296.

Pimentel, K., and Teixeira, K. 1994. *Virtual reality: through the new looking glass*. Blue Ridge Summit: Windcrest/McGraw-Hill.

Rao, A.; Lucas, A.; Morley, D.; M., S.; and G., M. 1993. Agent-oriented architecture for air-combat simulation. Technical Report Technical Note 42, The Australian Artificial Intelligence Institute.

Tambe, M., and Rosenbloom, P. 1995. Resc: An approach for real-time, dynamic agent tracking. In *Proceedings of the International Joint Conference on Artificial Intelligence*.

Ye, Y., and Tsotsos, J. K. 1995. Where to look next in 3d object search. In *1995 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 Intelligence and Mathematics*.

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

Ye, Y. 1996. *Sensor planning in 3D object search*. Ph.D. Dissertation, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 1A4.