Sensor Planning for Object Search

Yiming Ye and John K. Tsotsos

Department of Computer Science
University of Toronto
Toronto Ontario, Canada M5S 1A4

- Comments to:
  - yiming@vis.toronto.edu
  - tsotsos@vis.toronto.edu

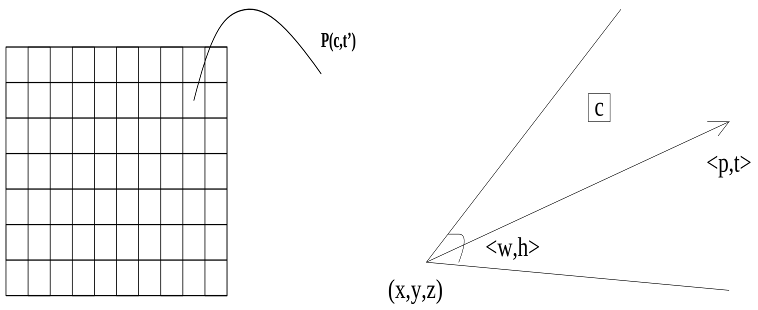
Introduction

what is object search?

- Object search: search for a 3D target within a given $3D$ environment by a robot.
- Contains
  - sensor planning
  - hardware manipulation
  - object recognition

Problem formulation

- Search region Ω, $\Omega = \cup_{i=1}^{n} c_i$
- Probability distribution $\mathbf{p}(c_i, t^{'})$
- Operation $\mathbf{f} = (x_c, y_c, z_c, p, t, w, h, a)$
- Cost function $\mathbf{t}(\mathbf{f})$
- Detection function $\mathbf{b}(c_i, \mathbf{f})$ ——— $\mathbf{p}(Detect \mid c_i, \mathbf{f})$.



- The probability of detecting the target by applying $\mathbf{f}$ is

$$P(\mathbf{f}) = \sum_{i=1}^{n} \mathbf{p}(c_i, t^{'})\mathbf{b}(c_i, \mathbf{f})$$

The Environment

- α_i: the target is in cube c_i
- α_o: the target is outside the region
- β: after applying a recognition action, the recognizer successfully detect the target
- Bayes formula is used to update distribution

$$P(\alpha_i \mid \neg\beta) = \frac{P(\alpha_i)P(\neg\beta \mid \alpha_i)}{P(\alpha_o)P(\neg\beta \mid \alpha_o) + \Sigma_{j=1}^{n} P(\alpha_j)P(\neg\beta \mid \alpha_j)}$$

- Updating rule:

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

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

- Initial available data
  - $\mathbf{p}^{[0]}(c_i)$
  - $\mathbf{b}(c_i, \mathbf{f}_j)$

- Effort allocation
  - $\mathbf{O_\Omega} = \{\mathbf{f'_1}, \ldots, \mathbf{f'_m}\}$: the set of all possible operations
  - Effort allocation $\mathbf{F} \subseteq O_\Omega$,
- For $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_k\}$,
  - the probability of detecting the target by this allocation is:

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

  - the total cost is:

$$C[F] = \sum_{1 \leq i \leq k} \mathbf{t}(\mathbf{f}_i)$$

- Object Search: Finding an allocation $\mathbf{F} \subseteq \mathbf{O_\Omega}$, which satisfies $C(F) \leq K$ and maximizes $P[F]$.

Property

- The general expression for the probability distribution $\mathbf{p}^{[k]}(c)$ after we have executed an action sequence $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_k \mid \mathbf{f}_i \in \mathbf{O}_\Omega\}$

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

  where

$$g(c, \mathbf{f}) = \begin{cases} 1 - \mathbf{b}(c, \mathbf{f}) & \text{if } c \in \Omega(\mathbf{f}) \\ 1 & \text{otherwise} \end{cases}$$

  - $\Omega(\mathbf{f})$ is the influence range of action $\mathbf{f}$. When the target is inside this range, the probability of detecting the target is not 0.

$$\Omega(\mathbf{f}) = \{c \mid \mathbf{b}(c, \mathbf{f}) \neq 0\}$$

- The general expression for the probability of detecting the target by an effort allocation $\mathbf{F} = \{\mathbf{f_1}, \ldots, \mathbf{f_q}\}$.

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

  In the above expression, only the initial available data are used

- We can see:
  - $P(\mathbf{F})$ is not influenced by the order of the applied actions.
  - If we restrict the available actions such that there is no cube belonging to any two action ranges: $\Omega(\mathbf{f}_j) \cap \Omega(\mathbf{f}_i) = \phi$, if $i \neq j$, then $P[\mathbf{F}]$ can be calculated by the following:

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

Complexity

Change the Maximization Problem into Decision Problem

- INSTANCE
  - A search region $\Omega = \cup_{i=1}^{n} c_i$
  - a finite set $\mathbf{O_\Omega} = \{\mathbf{f_1}^*, \mathbf{f_2}^*, \ldots, \mathbf{f_m}^*\}$
  - an expected probability function $P[\mathbf{F}^{'}]$ for allocation $\mathbf{F}^{'}$
  - a cost function $T[\mathbf{F}^{'}]$ for allocation $\mathbf{F}^{'}$
  - an environmental updating rule
  - a cost constraint K
  - an expected probability goal M
- QUESTION
  - Is there an effort allocation $\mathbf{F}$ such that $T[\mathbf{F}] \leq K$ and $P[\mathbf{F}] \geq M$

It is an NP problem

```
procedure NondeterministicObjectSearch(Ω,O_Ω,K, M,F)
    integer F(m)
    real K,M
    for i ← 1 to m do
        F(i) ← choice(0, 1, 2, ..., m)
    end for
    if T[F] > K or P[F] < M then
        failure
    else
        success
    endif
end NondeterministicObjectSearch
```

It belongs to NP-Complete class

- By restricting $\Omega(\mathbf{f}_j) \cap \Omega(\mathbf{f}_i) = \phi$, if $i \neq j$, we get
  - INSTANCE
    - A search region $\Omega = \cup_{i=1}^{n} c_i$
    - a finite set $\mathbf{O_\Omega} = \{\mathbf{f_1}^*, \mathbf{f_2}^*, \ldots, \mathbf{f_m}^*\}$
    - an expected probability function $P[\mathbf{F}']$ for allocation $\mathbf{F}' = \Sigma_{i=1}^{k} P^{[0]}(\mathbf{f_i})$, where $\mathbf{F}' = \{\mathbf{f}_1, \ldots, \mathbf{f}_k \mid \mathbf{f}_i \in \mathbf{O_\Omega}\}$
    - a cost function $T[\mathbf{F}'] = \Sigma_{i=1}^{k} \mathbf{t}(\mathbf{f_i})$
    - A cost constraint K
    - an expected probability goal M
  - QUESTION
    - Is there an effort allocation $\mathbf{F}$ such that $T[\mathbf{F}] \leq K$ and $P[\mathbf{F}] \geq M$
- Reduction from KNAPSACK problem

Greedy Strategy: One Step Look Ahead

- NP-Complete
- Number of actions is large
- Calculate $\mathbf{P}(\mathbf{f})$ is time consuming
- One step look ahead
  Already: $\mathbf{F}_q = \{\mathbf{f}_1, \ldots, \mathbf{f}_q\}$
  Select: $\mathbf{f}$
- $\Delta_P(\mathbf{f}) = \{\ \Pi_{j=1}^{q}[1 - P(\mathbf{f}_q)]\}P(\mathbf{f})$
  $\Delta_T(\mathbf{f}) = \mathbf{t}(\mathbf{f})$.
  maximize: $\frac{\Delta_P(\mathbf{f})}{\Delta_T(\mathbf{f})}$
- Objective function:

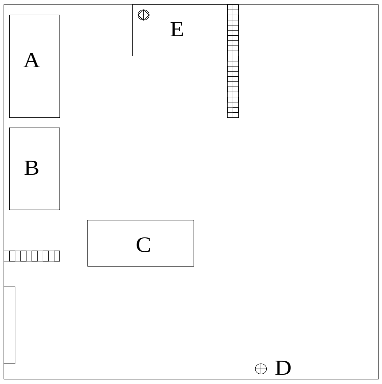
$$E(\mathbf{f}) = \frac{P(\mathbf{f})}{\mathbf{t}(\mathbf{f})}$$

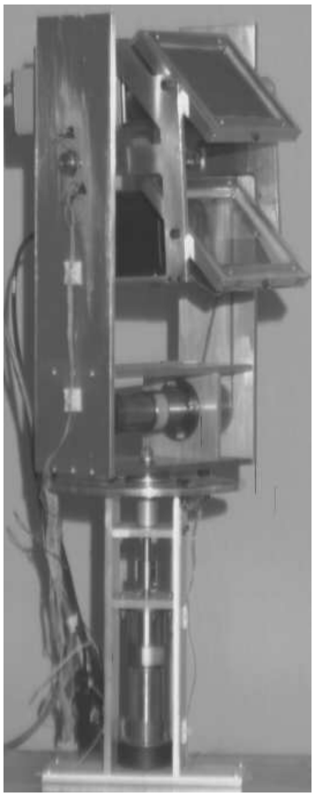
How Good is the Greedy Strategy?

- Greedy strategy may sometimes generate the optimal answer
  - Suppose the available operations $\mathbf{O_\Omega} = \{\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_m}\}$ satisfy:
    - $\mathbf{t}(\mathbf{f_i}) = \mathbf{t}(\mathbf{f_j})$, $1 \leq i, j \leq m$.
    - $\Omega(\mathbf{f}_j) \cap \Omega(\mathbf{f}_i) = \phi$, $1 \leq i, j \leq m, i \neq j$.

    Then, our greedy strategy will generate the optimal answer.
- This result is the guideline in real implementation
  - Decompose the sensing actions into a finite set of actions that must be tried. These actions satisfy $\Omega(\mathbf{f}_j) \cap \Omega(\mathbf{f}_i) = \phi$, $1 \leq i, j \leq m, i \neq j$.
  - When we only have one recognition algorithm, $\mathbf{t}(\mathbf{f_i}) = \mathbf{t}(\mathbf{f_j})$, $1 \leq i, j \leq m$ is also satisfied.
- So, we may even get an optimal answer in certain situations

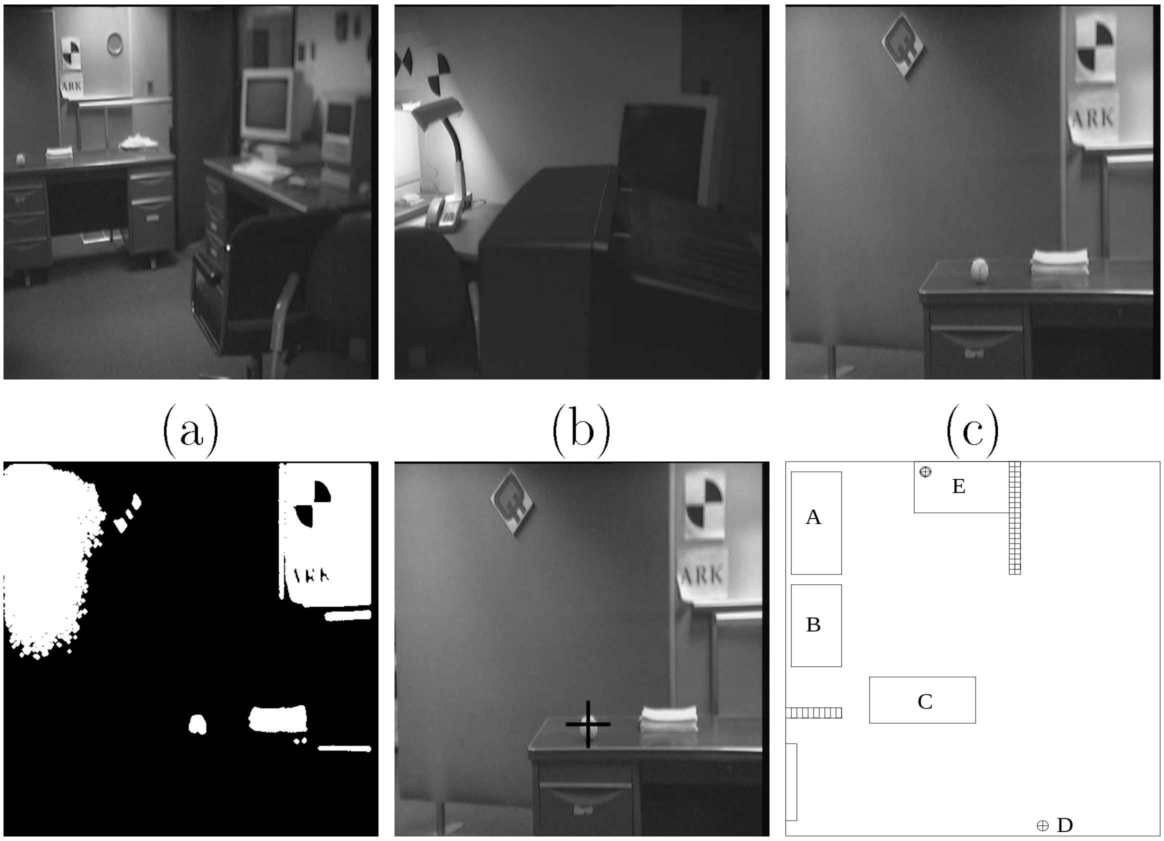
The Environment and the Search Agent





Real Experiment

(a), (b), (c): top three actions selected during a search process. The third action detects the target (baseball).
(d), (e): image processing result.
(f): The laser eye is on table E.

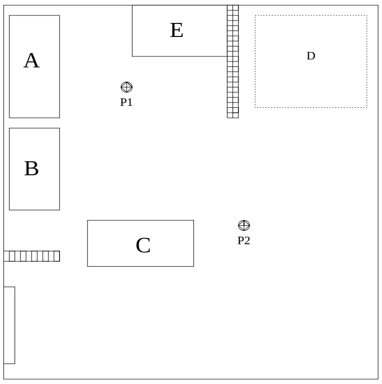


(a) (b) (c)

(d) (e) (f)

Move to Avoid Occlusion

(b), (c), (d), (e): The four actions selected by the robot when it is at P_1.
(f): The first action when the robot is at P_2.
(g), (h), (i): The image processing result.



(a)

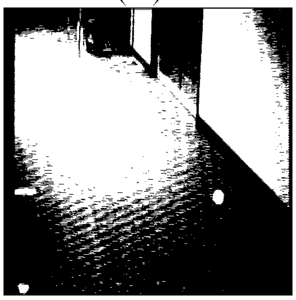
(b)

(c)

(d)

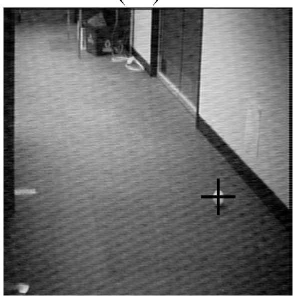
(e)

(f)



(g)

(h)



(i)

Conclusion

- Formalized the sensor planning task for object search
- Proved that this task is NP-Complete
- Decomposed the huge space of possible sensing actions into a finite set of actions to be tried.
- Introduced the concept of detection function for a recognition algorithm.
- Proposed an efficient algorithm for sensor planning and tested this algorithm with real experiments