Sensor Planning for Object Search

Yiming Ye and John K. Tsotsos

Dept. of Comp. Sci.
Univ. of Toronto

November, 1995

Sensor Planning for Object Search

- Introduction
- Problem formulation
- - The detection function
  - The environment updating
  - The sensed sphere
- Where to look next
- Where to move next
- Experiment
- Conclusion

Introduction

---

what is object search?

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

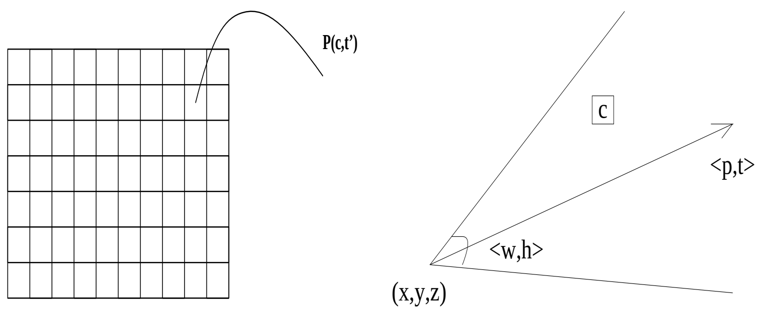
Introduction

Previous work

- Tsotsos: active vs. passive: efficiency concern
  - Planning versus Brute Force
    * Planning: more time need to compute "best" action
    * Bruce Force: more images needed to get the target
  - balancing point: when and how much planning will be benefit
- Wixson: indirect search
- Connell: looking at a certain height
- Rimey and Brown: task oriented system
- operation research: optimal search

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

- Bayes' formula is used to update distribution
- 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:

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

- Updating rule:

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

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

Problem Formulation

- 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:

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

  - the total cost is:

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

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

Property and Complexity

- Theoritical study is important
  - A search region $\Omega = \cup_{i=1}^{n} c_i$
  - a finite set of actions to be select $\mathbf{O_\Omega} = \{\mathbf{f_1}^*, \mathbf{f_2}^*, \ldots, \mathbf{f_m}^*\}$
  - 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
  - What is the behaviour of this system?
- Genaral expression for
  - probability distribution
  - detection probability for an action
  - detection probability for an effort allocation
- Based on initial available data
  - $\mathbf{p}^{[0]}(c_i)$
  - $b(c_i, \mathbf{f}_j)$

Property and Complexity

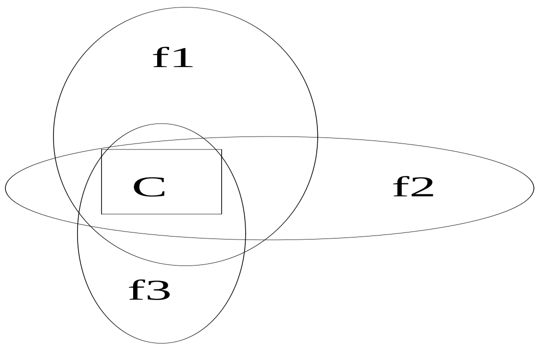
- Probability distribution:
  For allocation $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_q \mid \mathbf{f}_i \in \mathbf{O}_{\boldsymbol{\Omega}}\}$, we have

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

  where

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

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



Property and Complexity

- Detection Probability for an action
  For allocation $\mathbf{F} = \{\mathbf{f}_1 \ldots \mathbf{f}_q\}$,
-

$$\begin{aligned} P(\mathbf{f}_k) &= \sum_{j=1}^{n} \mathbf{p}^{[k-1]}(c_j)\mathbf{b}(c_j, \mathbf{f_k}) \\ &= \sum_{c \in \Omega(\mathbf{f}_k)} \mathbf{p}^{[k-1]}(c)\mathbf{b}(c, \mathbf{f_k}) \\ &= \sum_{c \in \Omega(\mathbf{f}_k) \bigcap \Omega} \mathbf{p}^{[k-1]}(c)\mathbf{b}(c, \mathbf{f_k}) \end{aligned}$$

-

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

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

Property and Complexity

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

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

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

- The formula is quite symmetric

Property and Complexity

- For an allocation $\mathbf{F} = \{\mathbf{f_1}, \ldots, \mathbf{f_q}\}$, $P(\mathbf{F})$ is not influenced by the order of the applied actions.
- For an allocation $\mathbf{F} = \{\mathbf{f_1}, \ldots, \mathbf{f_q}\}$, 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:

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

Property and 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 $P[\mathbf{F}] \geq M$

Property and Complexity

It is an NP problem

procedure NondeterministicObjectSearch(Ω,$\mathbf{O}_{\boldsymbol{\Omega}}$,$K$, M,F)

  integer F(m)

  real K,M

  for $i \leftarrow 1$ **to** m **do**

    $F(i) \leftarrow$ **choice**$(0, 1, 2, \ldots, m)$

  end for

  if $T[\mathbf{F}] > K$ or $P[\mathbf{F}] < M$ **then**

    failure

  else

    success

  endif

end NondeterministicObjectSearch

Property and Complexity

It belongs to NP-Complete class

- INSTANCE
  - A search region $\Omega = \cup_{i=1}^{n} c_i$
  - a finite set $\mathbf{O}_{\boldsymbol{\Omega}} = \{\mathbf{f_1}^*, \mathbf{f_2}^*, \ldots, \mathbf{f_m}^*\}$
  - an expected probability function $P[\mathbf{F}^{'}]$ for allocation $\mathbf{F}^{'} = \Sigma_{i=1}^{k} P^{[0]}(\mathbf{f_i})$, where $\mathbf{F}^{'} = \{\mathbf{f}_1, \ldots, \mathbf{f}_k \mid \mathbf{f}_i \in \mathbf{O}_{\boldsymbol{\Omega}}\}$
  - a cost function $T[\mathbf{F}^{'}] = \Sigma_{i=1}^{k} \mathbf{t}(\mathbf{f_i})$
  - A cost constraint K
  - an expected probability goal M
  - (Note: here we do not have updating rule and the order of actions does not matter)
- QUESTION
  - Is there an effort allocation $\mathbf{F}$ such that $T[\mathbf{F}] \leq K$ and $P[\mathbf{F}] \geq M$

Greedy Strategy: One Step Look Ahead

- NP-Complete
- Number of actions is big
- 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:

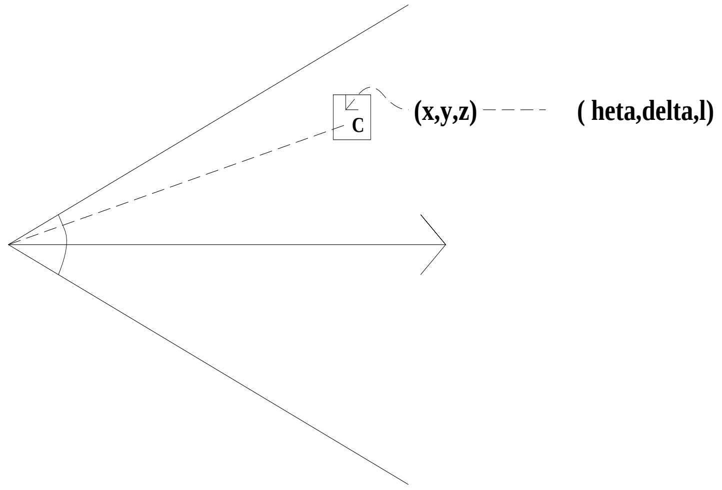
$$E(\mathbf{f}) = \frac{P(\mathbf{f})}{\mathbf{t}(\mathbf{f})}$$

Greedy Strategy: One Step Look Ahead

- Suppose the available operations $\mathbf{O_\Omega} = \{\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_m}\}$ satisfy:
  (1) $\mathbf{t}(\mathbf{f_i}) = \mathbf{t}(\mathbf{f_j})$, $1 \leq i, j \leq m$.
  (2) $\Omega(\mathbf{f}_j) \cap \Omega(\mathbf{f}_i) = \phi$, $1 \leq i, j \leq m, i \neq j$.
  Then, our the greedy strategy will generate the optimal answer.
- For our hardware
  - Where to look next
  - Where to move next

The detection function $\mathbf{b}(c, \mathbf{f})$

- Use $\mathbf{b_0}((x, y, z), \mathbf{f}) = \mathbf{b}((\theta, \delta, l), < w, h >)$ to approximate $\mathbf{b}(c, \mathbf{f})$.



- experiment

$$\frac{N_{success}}{N_{total}}$$

The detection function

It is not necessary to store $\mathbf{b}'((\theta, \delta, l), < w, h >)$ for every angle size. We can just store $\mathbf{b}'((\theta, \delta, l), < w_0, h_0 >)$ for a selected angle size $< w_0, h_0 >$. The value of other angle size $< w, h >$ can be transformed from that of the given angle size.

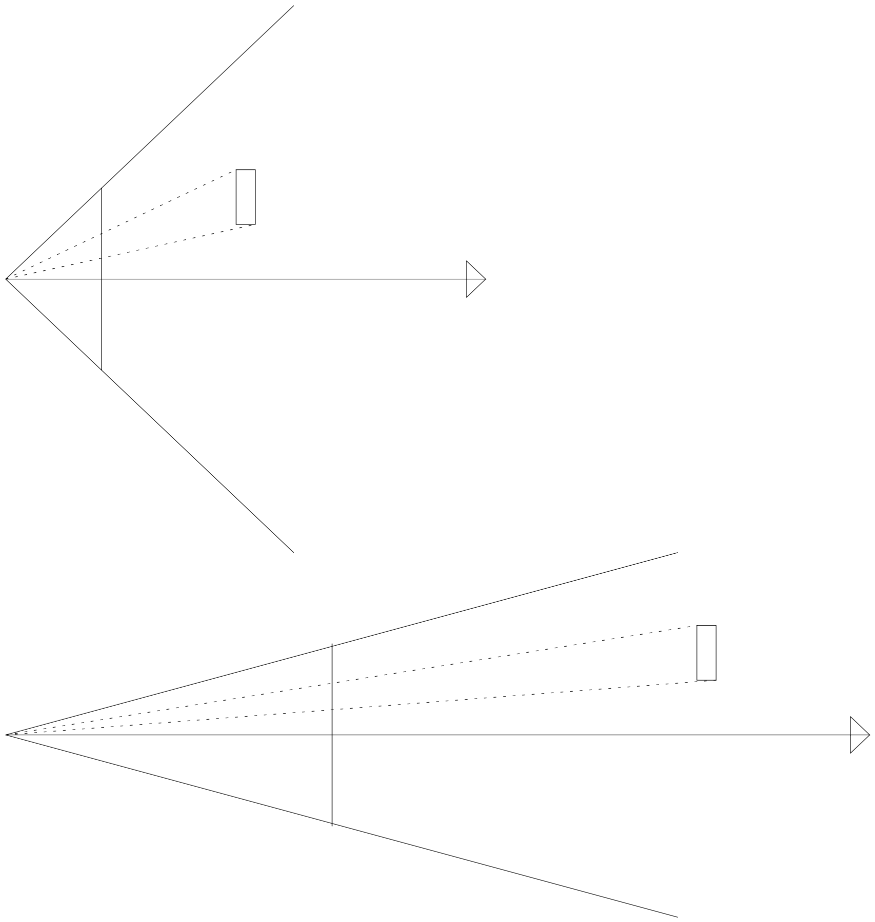


Figure 1:

The detection function

- given (θ, δ, l), find the corresponding $(\theta_0, \delta_0, l_0)$ that has the same detection value

  $$\mathbf{b}((\theta_0, \delta_0, l_0), < w_0, h_0 >) = \mathbf{b}((\theta, \delta, l), < w, h >)$$

  – scale factor: get

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

  – position factor: get

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

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

- – Transform $\mathbf{b}(\mathbf{f}, c)$ to $\mathbf{b}((\theta, \delta, l), < w, h >)$
  – retrieve $\mathbf{b}((\theta_0, \delta_0, l_0), < w_0, h_0 >)$ and get the value
  – result consistent with the misfocus property

Sensed sphere

- Used to represent the environment around the camera

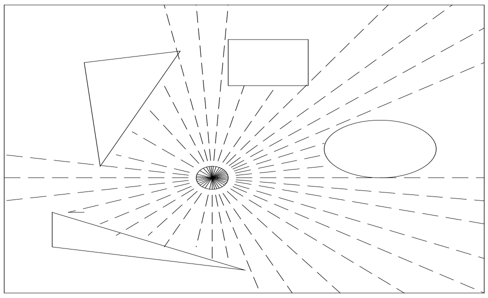
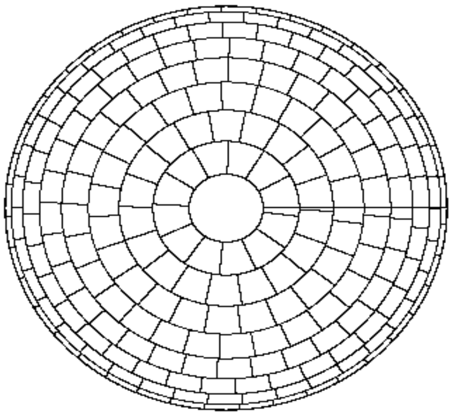
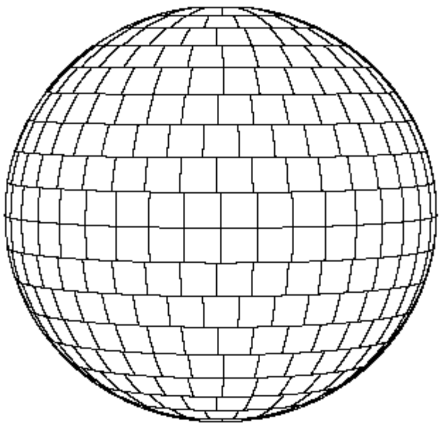


Figure 2:

- Tessellate the unit sphere in order to get approximately uniform patches and to minimize the mechanical operations when constructing the sensed sphere using laser



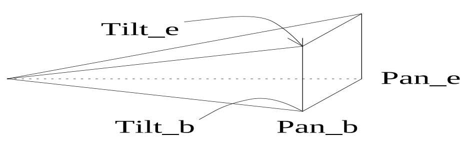


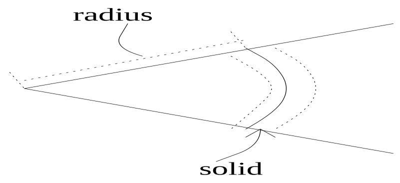
Sensed sphere

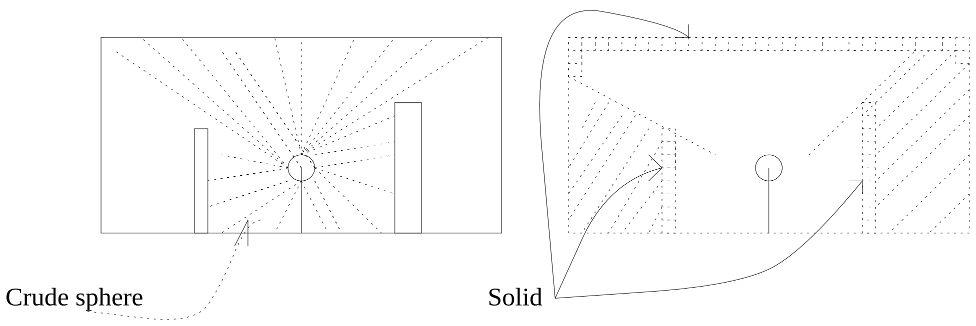
- The surrounding environment can be represented as the union of the solid angles corresponding to the patches
- $SensedSphere =$

$$\bigcup_i SolidAngle_i(pan_{b_i}, pan_{e_i}; tilt_{b_i}, tilt_{e_i}; radius_i; prob_i)$$

- Use sensed sphere to register the space: solid cube



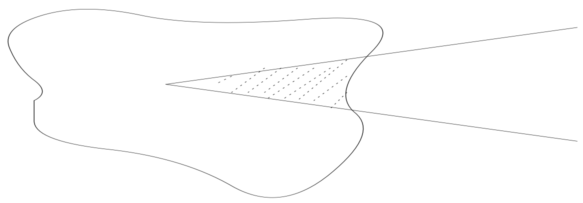




Where to look next: the task

Select best p, t, w, h for next action

- Task description
  - $\Psi = V \cap C$: those part of Ω which can be "seen" by the current camera setting p, t, w, h



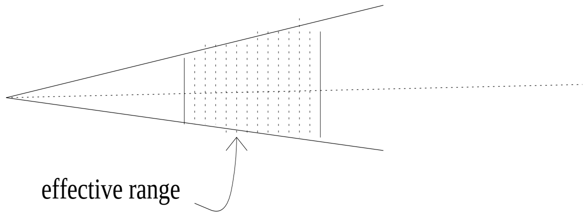
  - Our task: to select an operation $f_{p,t,w,h}$ that maximizes

$$E = \sum_{c_i \in \Psi} \mathbf{p}(c_i)\mathbf{b}(c_i, f_{p,t,w,h})$$

  - brute force: all possible $< w, h, p, t >$
    ours: only those necessary
- Task divided into
  - select angle size $< w, h >$
  - select $< p, t >$ for each candidate size.

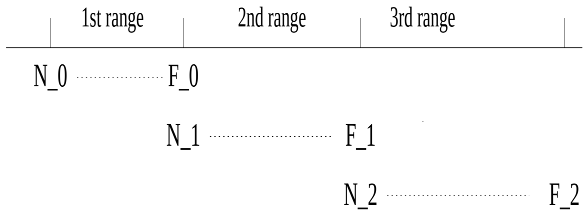
Where to look next: select angle size

1. Effective range $[N, F]$ of an viewing angle size



2. Angle size
   $< w_0, h_0 >, < w_1, h_1 >, \ldots, < w_{n_0}, h_{n_0} >$ should be selected to guarantee

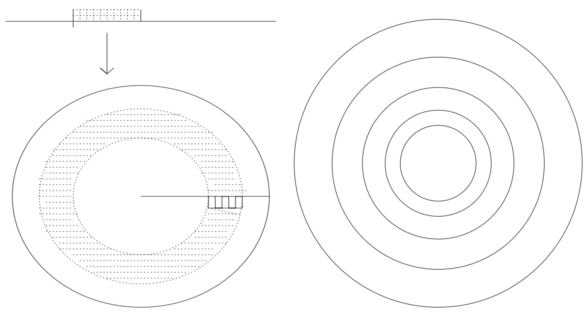
   - their effective ranges $[N_1, F_1], \ldots, [N_{n_0}, F_{n_0}]$ will cover the whole distance D of the search region
   - no overlap of their effective ranges



3. We have
   - $$w_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{w_0}{2})]$$
     $$h_i = 2arctan[(\frac{N_0}{F_0})^i tan(\frac{h_0}{2})]$$
   - $$N_i = F_0(\frac{F_0}{N_0})^{i-1}$$
     $$F_i = F_0(\frac{F_0}{N_0})^i$$
   - Number of angles:
     $$n_0 = \lfloor \frac{ln(\frac{D}{F_0})}{ln(\frac{F_0}{N_0})} - 1 \rfloor$$
4. Usually a few angle sizes are enough for search task

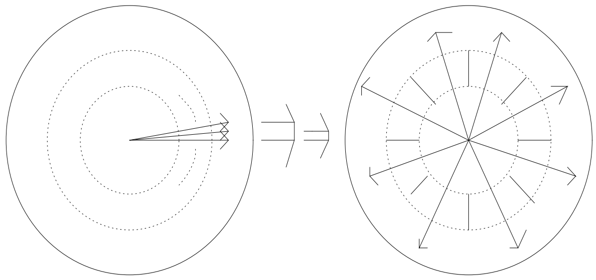
Where to look next: select viewing direction

- Divide the sensed sphere into layers according to the effective ranges of different viewing angles



- $$\bigcup_{<w,h>=<w_0,h_0>,\ldots,<w_{n_0},h_{n_0}>} \bigcup_i SolidAngle_i(pan_{b_i}, pan_{e_i}; tilt_{b_i}, tilt_{e_i}; radius_i; N_{<w,h>}, F_{<w,h>}; prob_i^{<w,h>})$$

- Many directions can be considered, but we only consider those directions with as little overlap as possible

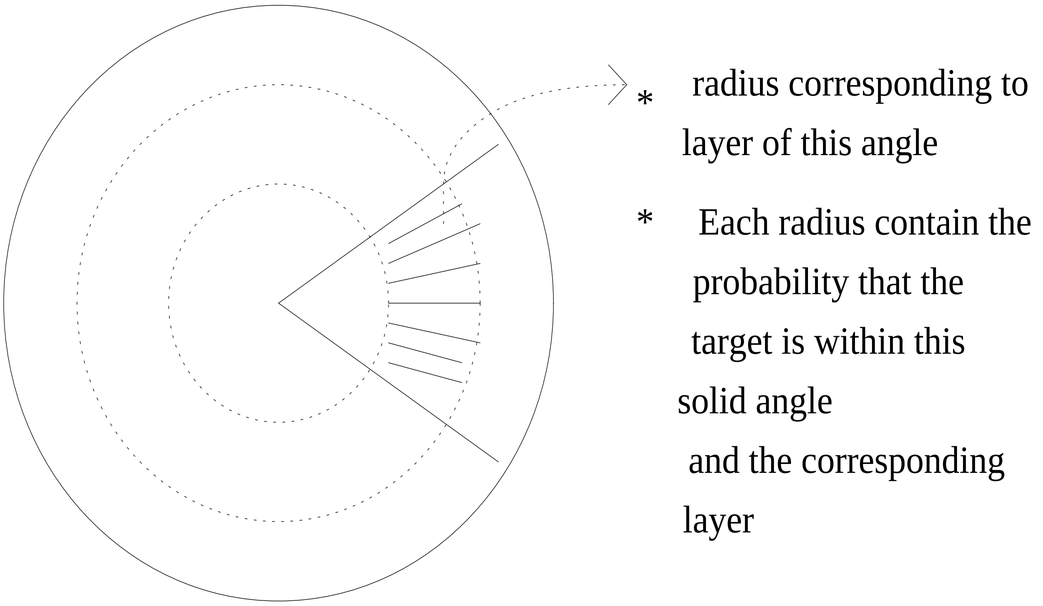


Where to look next: select viewing direction

- Approximate the detection probability for this direction by

$$\sum_{radius_i \in \mathbf{W}} prob_i^{<w,h>}$$

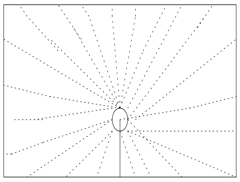
$\mathbf{W}$ is the constrained region by view volume



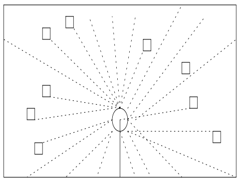
- The direction with the maximum probability will be selected as the best direction for this angle size $< w, h >$

Where to move next: find (x,y)

- Determine which position the robot can reach based on solid cubes
- Calculate the probability for each candidate position
  - "construct" the expected crude sphere based on the known search region boundary



  - Use the solid cube to constrain this sphere



  - Calculate the probability that the object is inside each expected sphere
- Select the position with maximum probability as the next position and drive the robot.

Simulation

- Sensor model: the biggest angle is $\frac{\pi}{4}$, the detection function is

$$b(c, f_{\pi/4}) = b(r, \varphi, \theta) = D(r)(1 - \frac{1}{6}\frac{\varphi}{\frac{\pi}{4}})(1 - \frac{1}{6}\frac{\theta}{\frac{\pi}{4}})$$

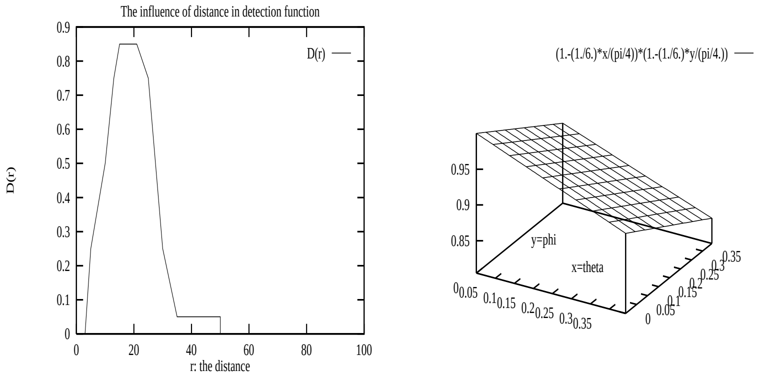
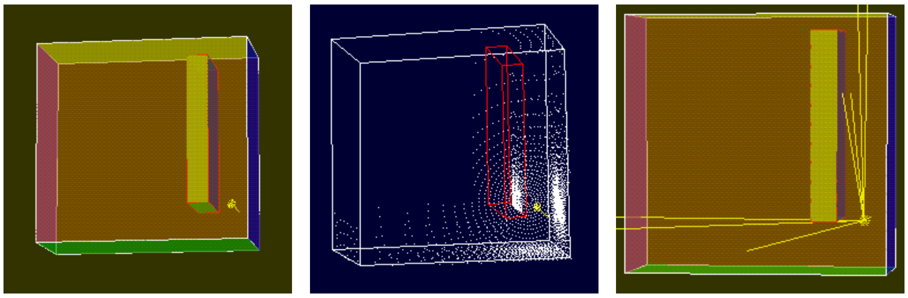


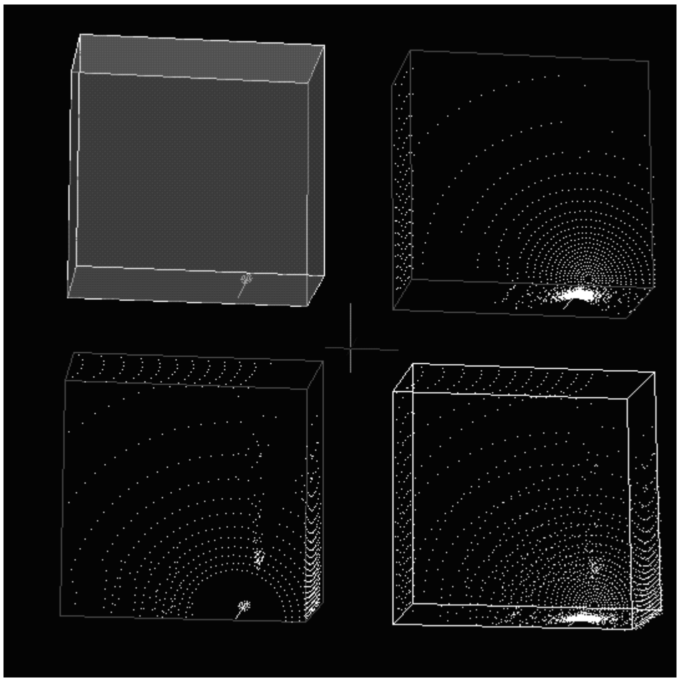
Figure 3: (a) The influence of the distance (b) The influence of the angle φ, θ

- Three angles needed in simulation
  - angle 45^o, range $[11, 27]$
  - angle 22^o, range $[27, 66]$
  - angle 9^o, range $[66, 162]$

Simulation

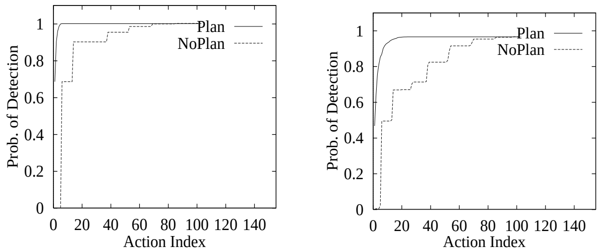
- Simulation 1: room–$50 \times 50 \times 25$; obstacle—$5 \times 37 \times 19$; robot height—15; initial position $(45, 10)$





Simulation

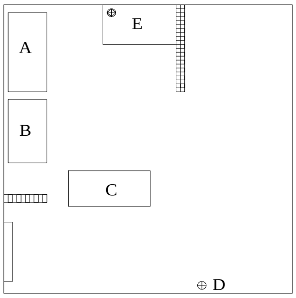
- Compare with non-planning strategy



(a) (b)

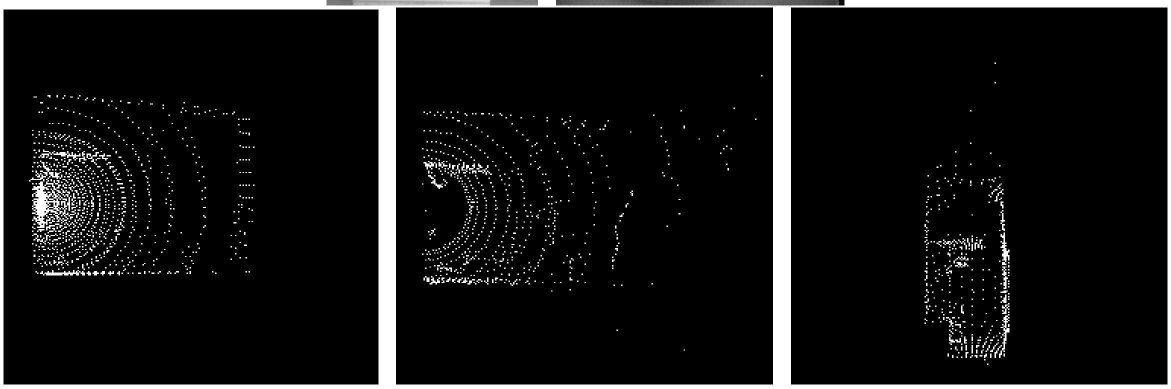
The detection probabilities $P[\mathbf{F}]$ of the planning strategy (Plan) and those of the non-planning strategy (No Plan). (a)$\sigma = 1$; (b)$\sigma = 5$. The non-planning strategy means first execute every action corresponding to the first layer one by one, then execute every action corresponding to the second layer one by one.

Experiment



(a)

(b)



Experiment

- Recognition Algorithm
  - $I_0 = 119$
  - $B_{min} = 250$ pixels
  - $B_{max} = 1375$ pixels
  - $R_0 = 0.91$
- Effective Range
  - $< 1.47m, 3.01m >$
  - $< 3.01m, 6.16m >$
- Experimental Results

	Blob Size	Percentage
N_0	1366 pixels	0.9356
F_0	273 pixels	0.9158

	Blob Size	Percentage
N_1	1332 pixels	0.9114
F_1	271 pixels	0.9077

Experiment

Effective range

Experiment

- Knowledge influence

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

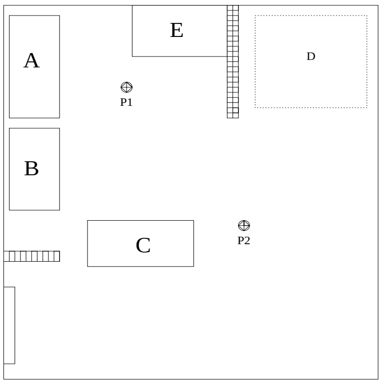
•

Experiment: Move next

- Recognition algorithm
  - $I_0 = 110$
  - $B_{min} = 170$ pixels
  - $B_{max} = 5500$ pixels
  - $R_0 = 0.91$
- .

	Blob Size	Percentage
N_0	5478 pixels	0.9237
F_0	171 pixels	0.9474

- Environment

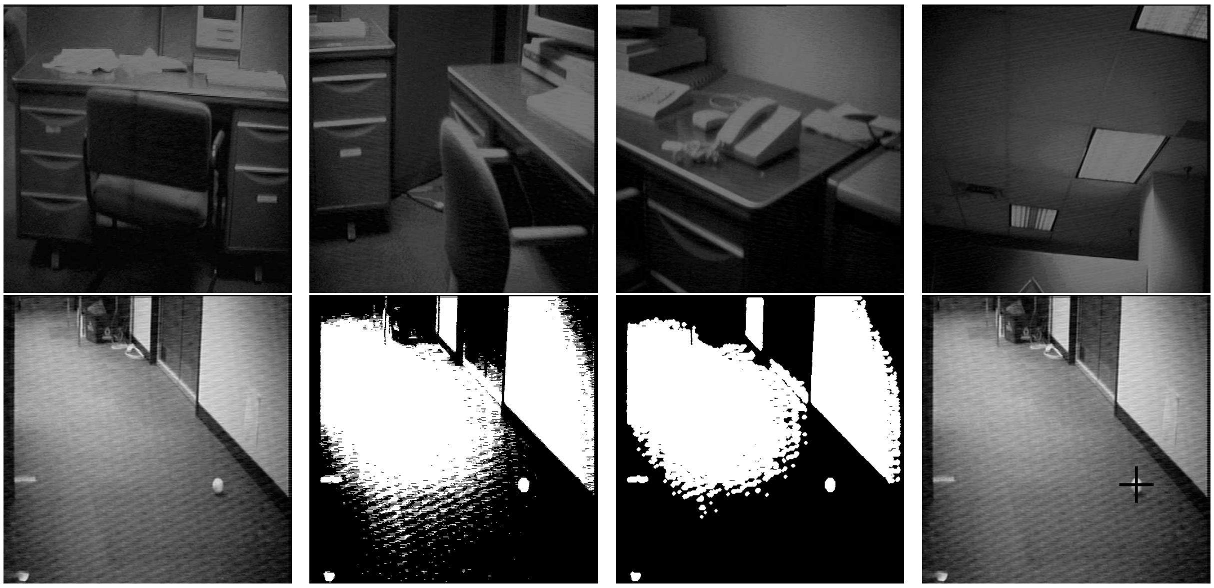


Experiment: Move next

- Give B, C, D high probability

  Four actions selected at P_1

Tilt	Pan	Probability of Detection
1.801100	0.000000	0.037403
2.315700	5.655361	0.036140
2.315700	4.948441	0.036654
1.286500	5.366405	0.006105



Conclusion: contribution

- Formulized the sensor planning task for object search
- Analyzed the properties and complexity of this task
- Proposed a search strategy and tested it with experiment