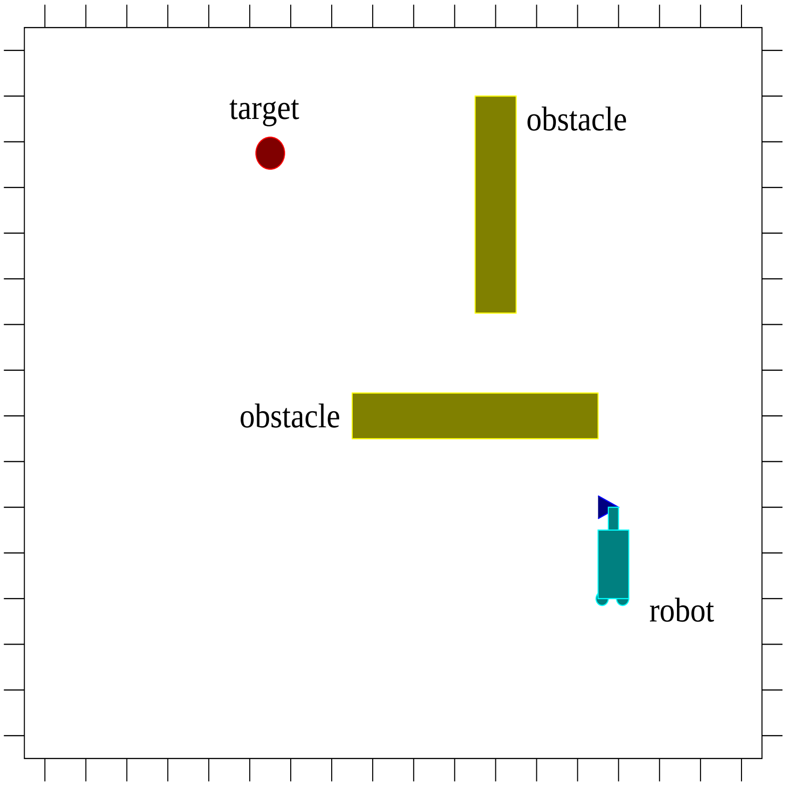
What is object search?



- Object search: search for the target within a given $3D$ environment by a robot equipped with a camera and a method of calculating crude depth
- Components
  - sensor planning
  - hardware manipulation
  - object recognition

Why Object Search?

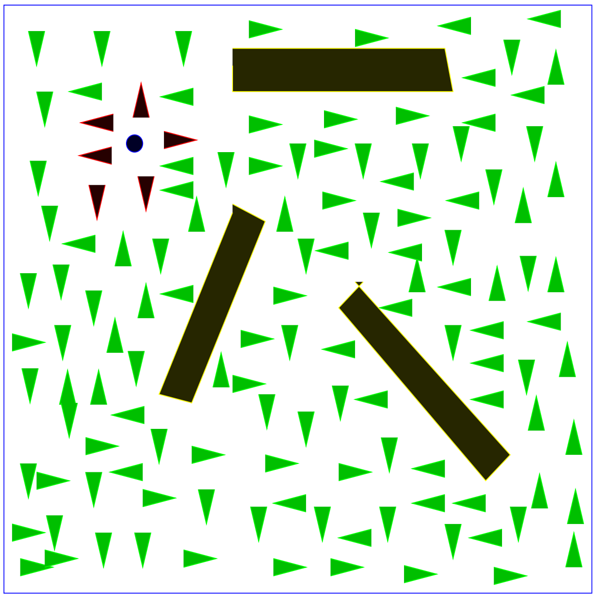
- Practical value
  - hazard environment
  - landmark in navigation
- Active vision
  - history
    - early research, engineering nature
    - Marr, fundamental principle
    - Active Vision (Bajcsy etc.): behavior emphasized
  - benefit (Bajcsy, Ballard, etc.)
  - price payed for the benefit, efficiency concern (Tsotsos)
  - Object search: Planning versus Brute Force
    - Planning: more time needed to compute the “best actions”
    - Brute force: more images needed to get the target
    - balancing point: when and how much planning will be benefit

Previous work: very little!

- Garvey, 1976: indirect search strategy
- Wixson and Ballard, 1992: indirect search analysis
- Connell, 1989: search and collecting soda cans
- Rimey and Brown, 1992: task oriented system TEA

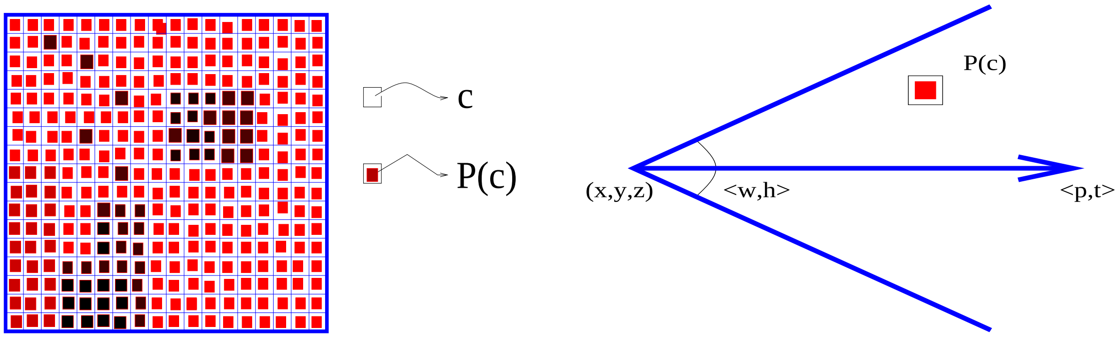
Object search is not trivial

- Object search is not simple
  - occlusion might occur
  - no perfect recognizer
  - huge number of candidate; little useful



- Need to integrate
  - recognizer's ability
  - sensor's state and it's effects, occlusion
  - initial knowledge should be used to guide the search process.
- Vision is only a component of a whole system.

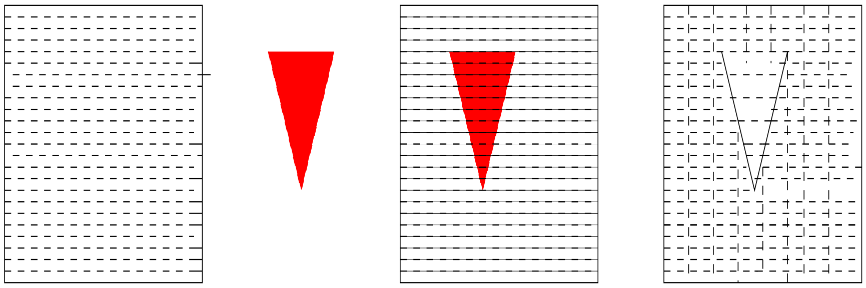
Problem Formulation



- Environment
  - Search region Ω, $\Omega = \cup_{i=1}^{n} c_i$
  - Probability distribution $\mathbf{p}(c_i, \tau)$
- Action
  - 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

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

- The Belief Update



  - Bayes formula is used to update distribution
  - α_i: target in cube c_i
  - α_o: target outside the region
  - β: after applying $\mathbf{f}$, the the recognizer detects the target

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

  - Updating rule:

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

- Effort Allocation
  - Effort allocation
    - $\mathbf{O}_{\mathbf{\Omega}} = \{\mathbf{f}_1, \ldots, \mathbf{f}_m\}$: the set of all possible operations
    - Effort allocation $\mathbf{F} \subseteq \mathbf{O}_{\mathbf{\Omega}}$,
  - For $\mathbf{F} = \{\mathbf{f}_{i_1}, \ldots, \mathbf{f}_{i_k}\}$,
    - the probability of detecting the target by this allocation is:

$$\begin{aligned} P[\mathbf{F}] &= P(\mathbf{f}_{i_1}) + [1 - P(\mathbf{f}_{i_1})]P(\mathbf{f}_{i_2}) \\ &+ \ldots \\ &+ \{\prod_{j=1}^{k-1}[1 - P(\mathbf{f}_{i_j})]\}P(\mathbf{f}_{i_k}) \end{aligned}$$

    - the total cost is:

$$C[\mathbf{F}] = \sum_{1 \leq j \leq k} \mathbf{t}(\mathbf{f}_{i_j})$$

- **The Task of Object Search**: Finding an allocation $\mathbf{F} \subseteq \mathbf{O}_{\mathbf{\Omega}}$, which satisfies $C[\mathbf{F}] \leq K$ and maximizes $P[\mathbf{F}]$.

Theoretical Study

- Theoretical study is important
  - $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k\}$
  -
$$\begin{aligned} P[\mathbf{F}] &= P(\mathbf{f}_1) + [1 - P(\mathbf{f}_1)]P(\mathbf{f}_2) \\ &\quad + \ldots + [1 - P(\mathbf{f}_1)] \ldots [1 - P(\mathbf{f}_{k-1})]P(\mathbf{f}_k) \end{aligned}$$
  - $x_1 + [1 - x_1]x_2 + \ldots + [1 - x_1] \ldots [1 - x_{k-1}]x_k$
  - $\Sigma_i x_i - \Sigma_{ij} x_{ij} + \ldots + (-1)^{k-1} x_1 x_2 \ldots x_k$
  - So, $P[\mathbf{F}]$ symmetric! ..., but ...
  - $\mathbf{F}^* = \{\mathbf{f}_2, \mathbf{f}_1, \ldots, \mathbf{f}_k\}$
  - $P(\mathbf{F}^*) = P(\mathbf{F})$
  - BUT, $P_F(\mathbf{f}_\mathbf{1}) \neq P_{F^*}(\mathbf{f}_\mathbf{1})$, probability update
  - So, $P[\mathbf{F}]$ NOT symmetric!
  - Are you sure?!

- **What We Have?**
  - 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 We Want to Know**: the behavior of this system? especially
  - $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k\}$
  - General expression for
    * probability distribution $\mathbf{p}^{[k]}(c)$
    * detection probability for an action $\mathbf{P}(\mathbf{f}_k)$
    * detection probability for an effort allocation
  - Based on initial available data
    * $\mathbf{p}^{[0]}(c_i)$
    * $\mathbf{b}(c_i, \mathbf{f}_j)$
  - How difficult the task is.

- Probability distribution A:

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

where

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

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

- Probability distribution B:

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

- Detection Probability by $\mathbf{f}_k$

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

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

- Detection Probability for action $\mathbf{f}_k$ of $\mathbf{F}$

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

- From

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

where

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

- Separate Ω into various cases such that the above can be used

- Detection Probability for action $\mathbf{f}_k$ of $\mathbf{F}$

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

- From

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

- Find the product of $\mathbf{p}^{[k-1]}(c)$, then use

$$\sum_{c\in\Omega} \mathbf{p}^{[0]}(c)\mathbf{b}(c,\mathbf{f}_{i_1})\ldots\mathbf{b}(c,\mathbf{f}_{i_j})\mathbf{b}(c,\mathbf{f}_k)$$

$$=$$

$$\sum_{c\in\Omega(\mathbf{f}_{i_1})\cap\ldots\cap\Omega(\mathbf{f}_{i_j})} \mathbf{p}^{[0]}(c)[\mathbf{b}(c,\mathbf{f}_{i_1})\ldots\mathbf{b}(c,\mathbf{f}_{i_j})\mathbf{b}(c,\mathbf{f}_k)$$

-

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

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

- For the **effort allocation** $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_q\}$, we have

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

- The Formula is Symmetric —– Application order does not matter
- Intuitively it's true

$$\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \ldots, \mathbf{f}_r, \ldots, \mathbf{f}_q\}$$

$$\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_r, \ldots, \ldots, \mathbf{f}_q\}$$

- **Theorem:** The task of **Sensor Planning for Object Search is NP-Complete** —– by reducing to Knapsack problem.

Strategy

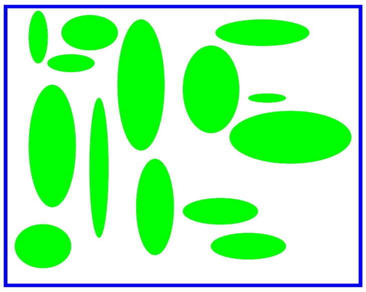
- One step look ahead "greedy strategy" objective function:

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

- Expected time of detecting the target for $\mathbf{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_k\}$

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

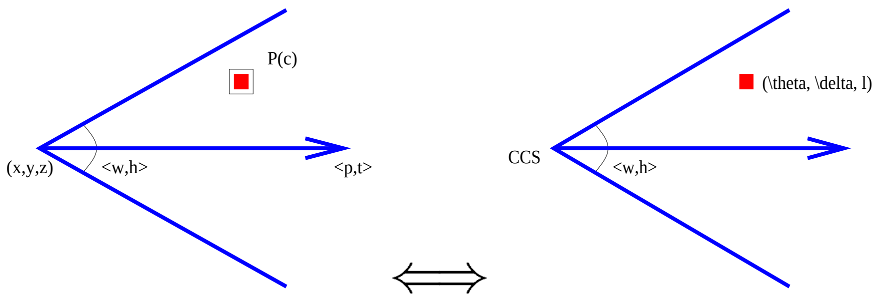
- How good is the greedy strategy?



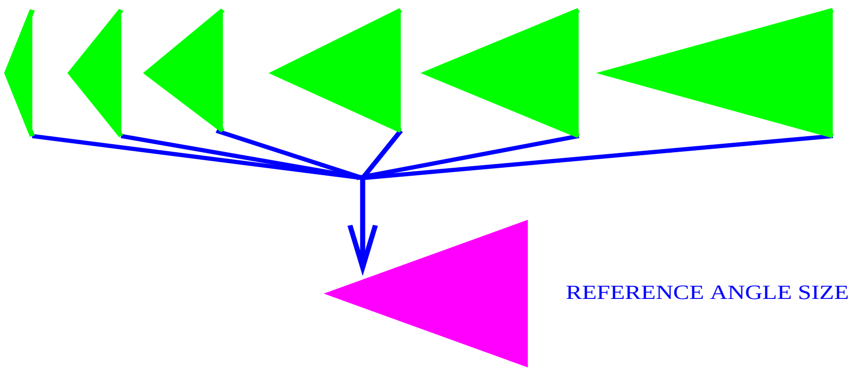
- **Where to Look Next** and **Where to Move Next**

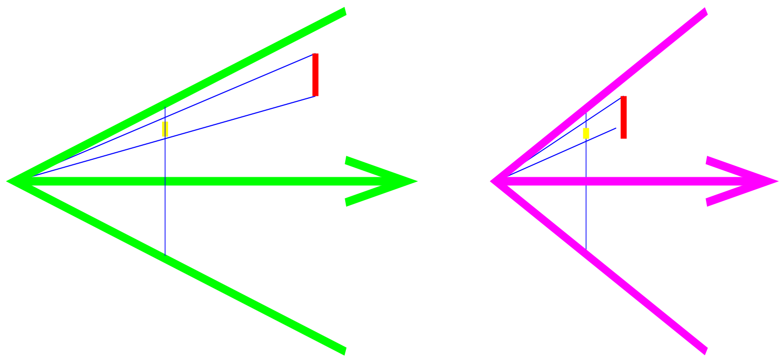
Detection function

- $P(\mathbf{f}) = \Sigma_{i=1}^{n} \mathbf{p}(c_i, \tau)\mathbf{b}(c_i, \mathbf{f}) \rightsquigarrow \mathbf{b}(c, \mathbf{f})$
- $\mathbf{b}(c, \mathbf{f}) \iff \mathbf{b}(\langle w, h\rangle, \langle \theta, \delta, l\rangle) \rightsquigarrow$ in CCS



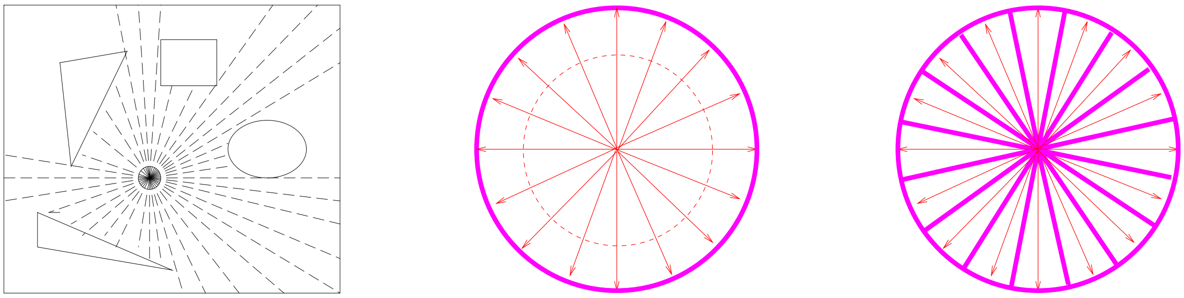
- Only need the values for one angle size, $\mathbf{b}(\langle w, h\rangle, \langle \theta, \delta, l\rangle) \approx \mathbf{b}(\langle w_0, h_0\rangle, \langle \theta_0, \delta_0, l_0\rangle)$



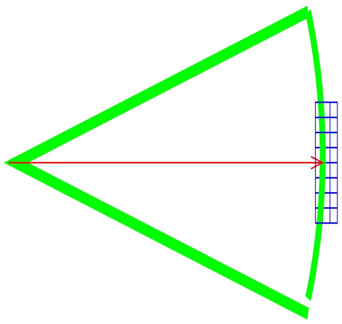


Occlusion Info. - Sensed Sphere

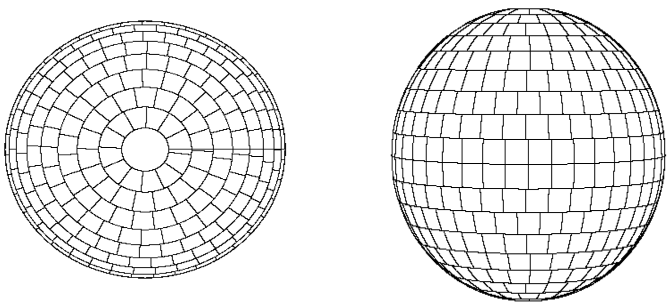
- Continues - Impossible $\rightsquigarrow$ Discrete



- Solid cube

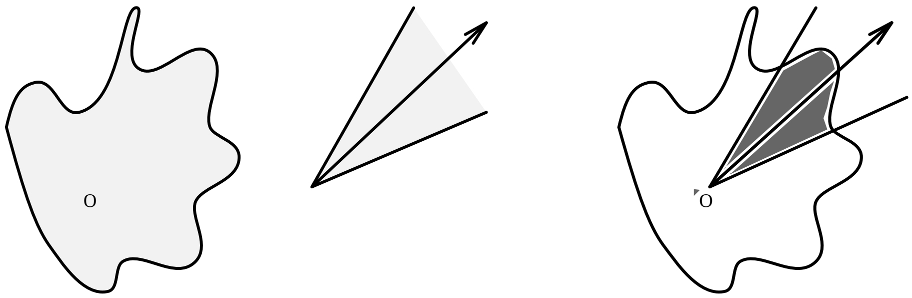


- Our tessellation of the unit sphere



Where to Look Next

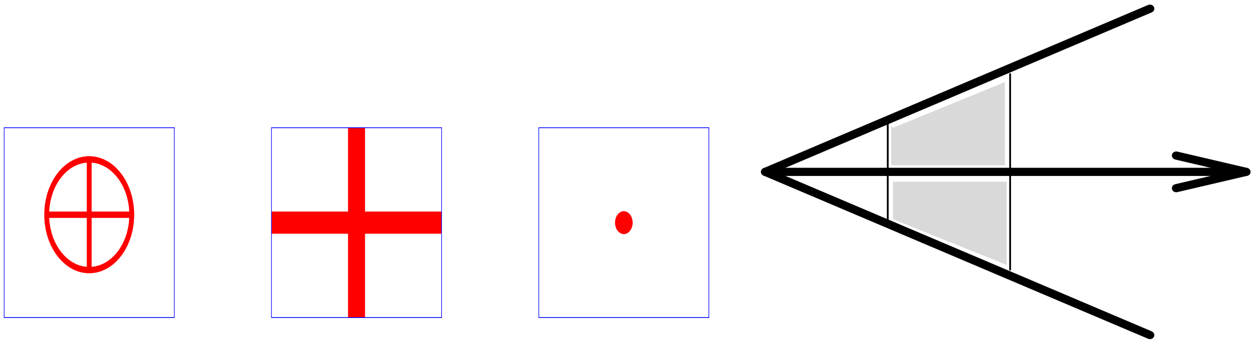
- Task description
  - $\Psi = ViewVolume \cap SensedSphere$



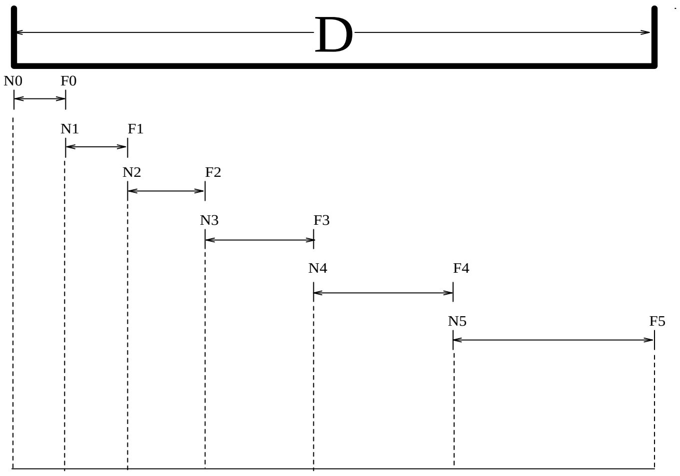
  - Maximizes $E = \Sigma_{c_i \in \Psi}\, \mathbf{p}(c_i)\mathbf{b}(c_i, \mathbf{f}_{p,t,w,h})$
- Brute Force?
  - if
    * $N_{\langle w,h \rangle} = 1500$; $N_{pan} = 360$; $N_{tilt} = 360$
    * $P(\mathbf{f})$ for 1 second
  - then: need 54000 hours, or 6 years and 2 months!
- Our Approach
  - Decompose the huge space into those that must be considered.
  - Task divided into
    * select angle size $< w, h >$
    * select $< p, t >$ for each candidate size.

Selecting Angle Sizes

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

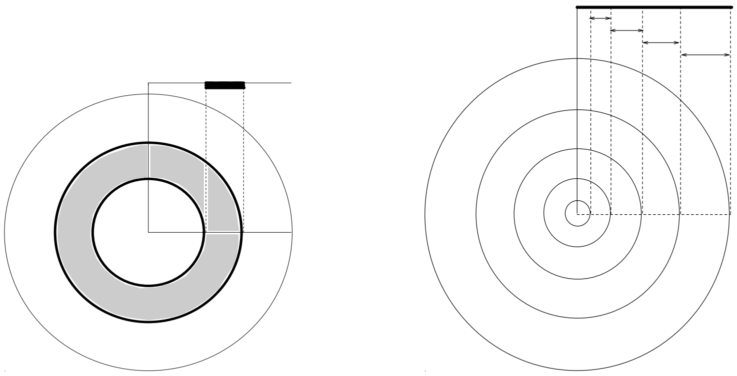


- Angle size $\langle w_0, h_0 \rangle, \langle w_1, h_1 \rangle, \ldots, \langle w_{n_0}, h_{n_0} \rangle$ should be selected to guarantee
  – their effective ranges $[N_1, F_1], \ldots, [N_{n_0}, F_{n_0}]$ will cover the whole distance D of the search region
  – no overlap of their effective ranges

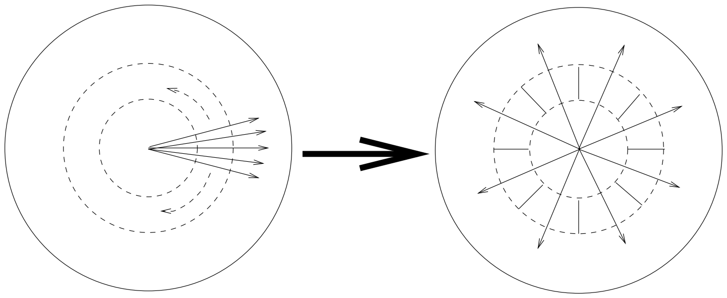


Selecting Viewing Direction

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

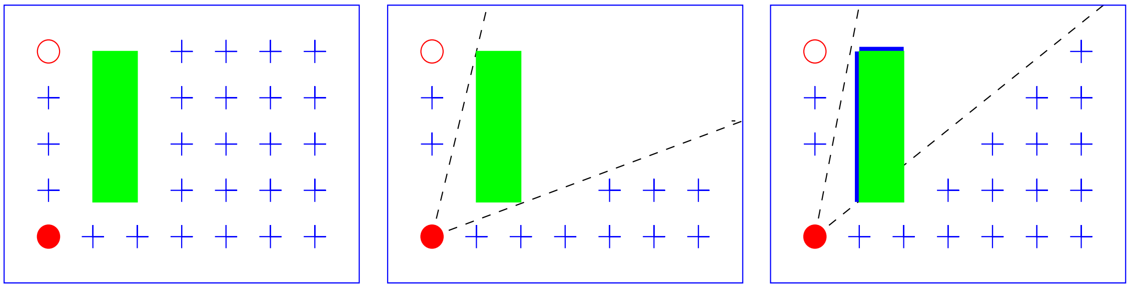


- Many directions can be considered, but we only consider those directions with as little overlap as possible

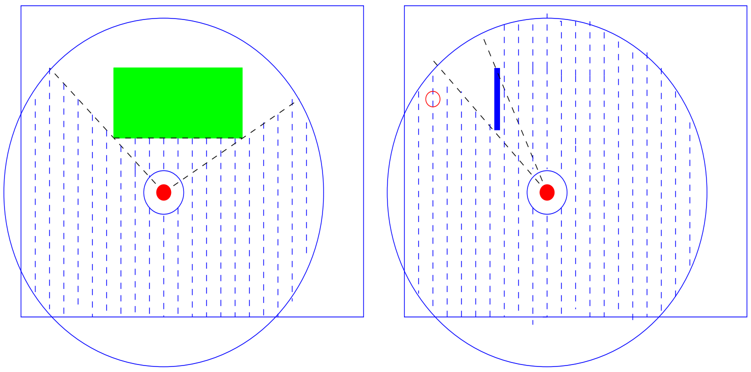


Where to move next

- Candidate position: the position that the robot can reach



- Benefit for a candidate position: the probability of the presence of the target within the region that can be examined by the position.

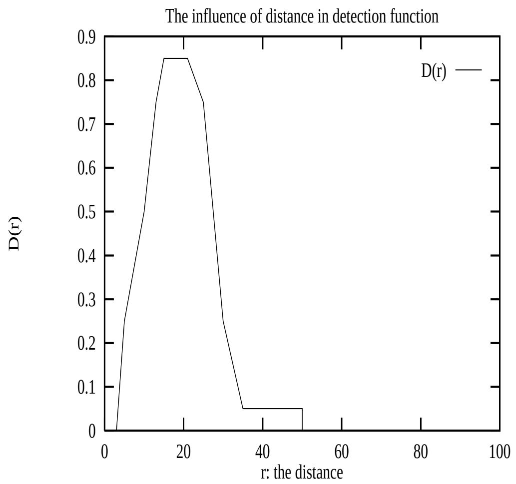


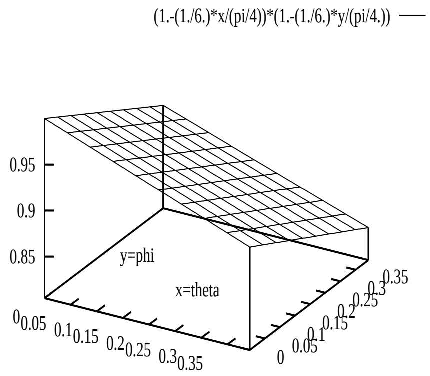
- Select the position with maximum benefit as the next position and drive the robot.

Simulation

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

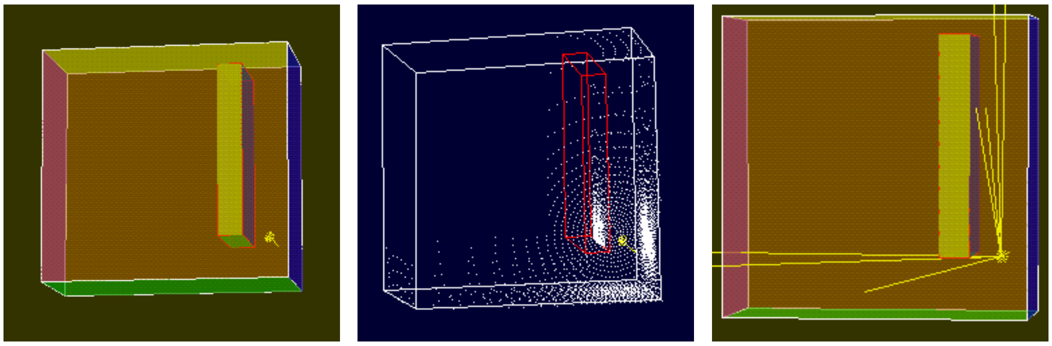
$$b(c, f_{\pi/4}) = b(l, \varphi, \theta) = D(l)(1 - \frac{1}{6}\frac{\varphi}{\frac{\pi}{4}})(1 - \frac{1}{6}\frac{\theta}{\frac{\pi}{4}})$$



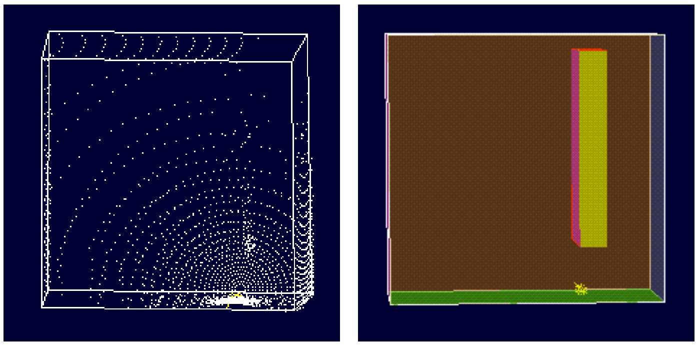


- Three angles needed in simulation
  - angle $\langle 45^o \times 45^o \rangle$, effective range $[11, 27]$
  - angle $\langle 22^o \times 22^o \rangle$, range $[27, 66]$
  - angle $\langle 9^o \times 9^o \rangle$, range $[66, 162]$

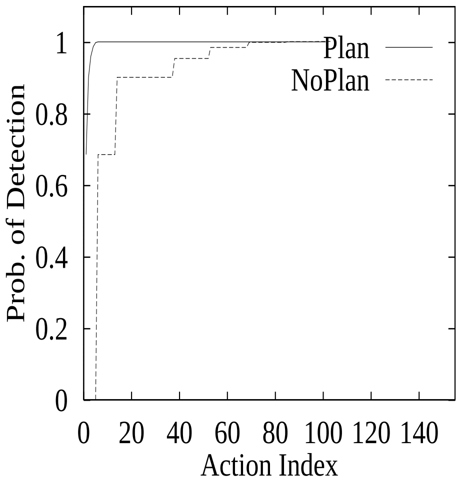
- **Simulation 1**: room–$50 \times 50 \times 25$; obstacle—$5 \times 37 \times 19$; robot height—15; initial position $(45, 10)$
- – Where to Look Next



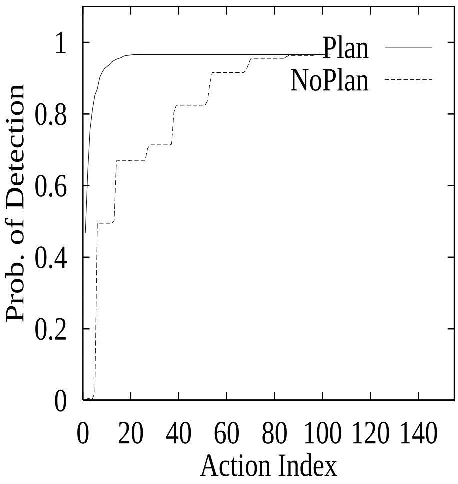
  – Where to Move Next



- **Simulation 2**
  - Comparison ($P[\mathbf{F}]$) between planning strategy and non-planning strategy.
  - Environment, robot position, sensor model same as before
  - No obstacle
  - Probability distribution - normal distribution $N(\mu, \Sigma)$, high at the center of the environment $\mu = (25, 25, 15)^T$. $\Sigma = diag(\sigma^2, \sigma^2, (\frac{\sigma}{2})^2)$ The small the value of σ, the dense the distribution



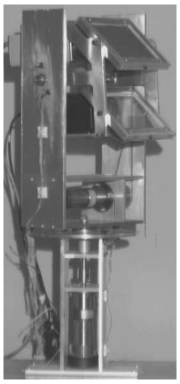
($\sigma = 1$)

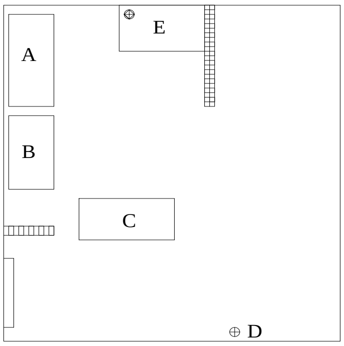


($\sigma = 5$)

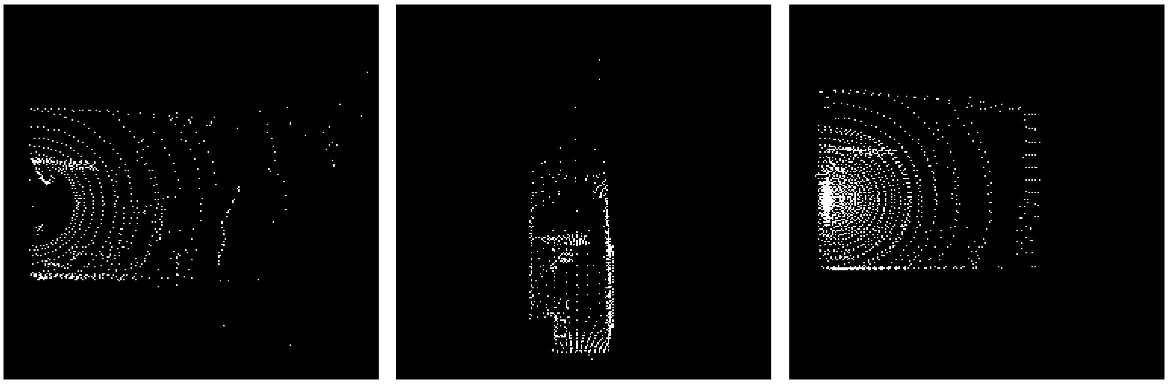
Experiment

- The sensor and the search region

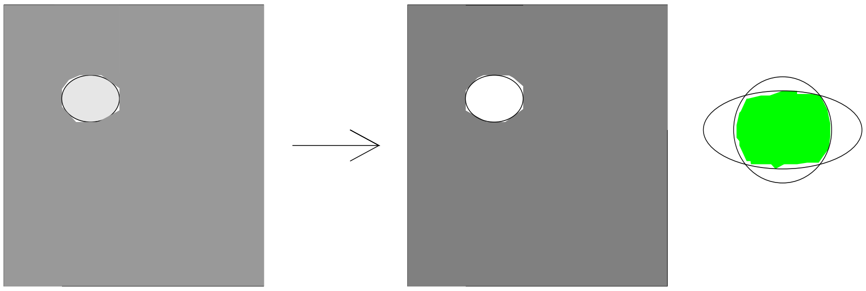




- The sensed sphere



- Task: search for a baseball
- Recognition Algorithm

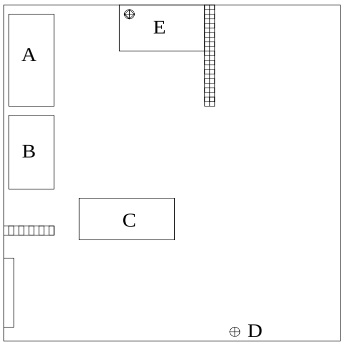


  - $I_0 = 119$
  - $B_{min} = 250$ pixels
  - $B_{max} = 1375$ pixels
  - $R_0 = 0.91$
- Effective Range
  - $< 1.47m, 3.01m >$
  - $< 3.01m, 6.16m >$
- Experimental Results

	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

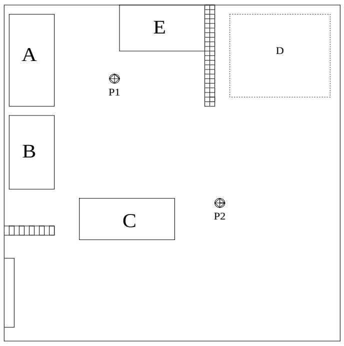
- Environment: give A, B, C high probability



- The actions selected

Experiment with ARK Robot

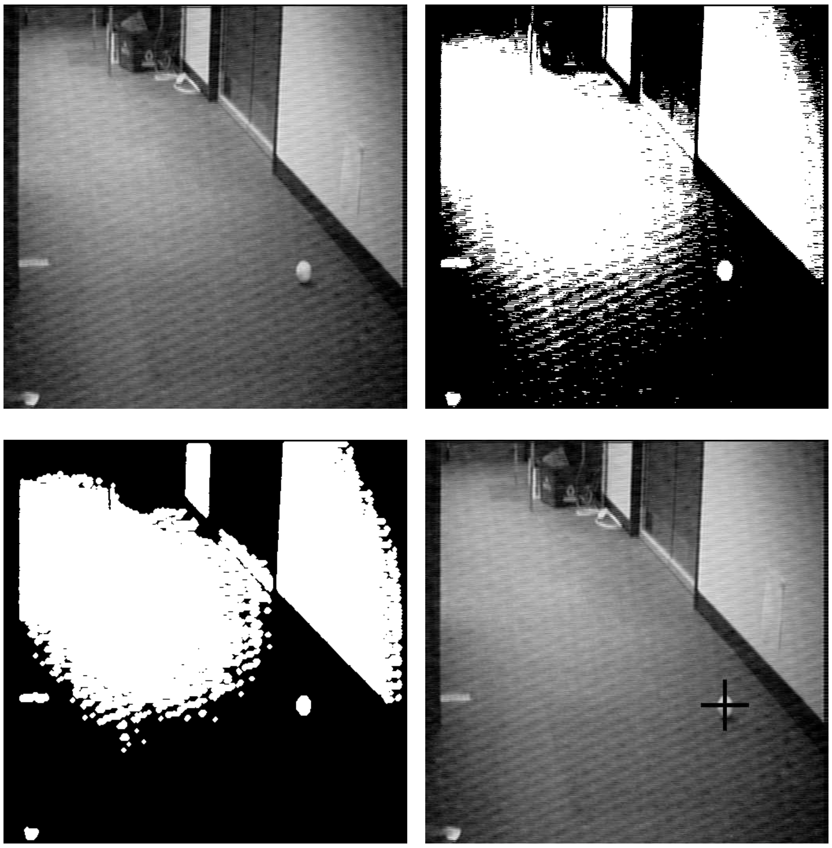
- The ARK robot and the environment

- The top 4 actions at the first position




• The action at the second position



Conclusion

- Formulated 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 experiment