Hypothesis Testing Framework for Active Object Detection

Nikolay Atanasov\textsuperscript{1}  Bharath Sankaran\textsuperscript{2}  Jerome Le Ny\textsuperscript{3}
Thomas Koletschka\textsuperscript{1}  George Pappas\textsuperscript{1}  Kostas Daniilidis\textsuperscript{1}

\textsuperscript{1}GRASP Lab, University of Pennsylvania, USA
\textsuperscript{2}CLMC Lab, University of Southern California, USA
\textsuperscript{3}Department of Electrical Engineering
Ecole Polytechnique de Montreal, Canada

IEEE International Conference on Robotics and Automation
Karlsruhe, Baden-Württemberg, Germany, May 2013
Motivation: from signal to symbol

How do we get from:

the traditional metric representation

to semantic information necessary for interactive tasks:
Central problem: object detection and pose estimation in a cluttered scene

The traditional approach in machine perception is single-view

Challenges:

Ambiguity in pose and identity

Lighting ambiguity and occlusion
Active perception

- A robot **cannot live with google-like results**, i.e. when the best hit is not on top of the list

- Can **actively move** to reveal more information about the identity and pose of an object

- Instead of placing the burden of perfect results on the static detector:
  - Adjust sensor settings
  - Choose appropriate intrinsics
  - Use sensor mobility

image: Masaki Onishi’s website
http://onishi-lab.jp/opus/index-e.html

"We do not just see, we look."
Ruzena Bajcsy (Proc. IEEE, 1988)
Sensor management:
- Solid theoretical foundations (Hero, Castañoń, Krause)
- Viewpoint selection based on information theoretic objectives
- Commonly static or stateless sensors (PZT cameras, sensor networks)
- Simple observation models; little attention devoted to real sensors

Machine perception:
- Very advanced approaches for real sensors
- Low-cost RGB-D cameras and open-source pointcloud libraries (e.g. PCL)
- Predominantly static single-view analysis or heuristic viewpoint selection
Work in active perception

Current approaches use greedy viewpoint selection and make restrictive assumptions about the sensors:

▶ Eidenberger and Scharinger ’10:
  ▶ gaussian mixture representation of an object's pose and class
  ▶ general observation model
  ▶ myopic planning to reduce the differential entropy in the pose and class distributions

▶ Velez, Roy et al. ’11:
  ▶ binary hypothesis for candidate detection: "present" vs "not present"
  ▶ greedy information gain maximization
  ▶ pose estimation is not included in the optimization

▶ Karasev, Soatto et al. ’12:
  ▶ simplified but theoretically motivated representation of an object's pose and class
  ▶ explicit form of the sensing process
  ▶ maximize conditional entropy of the next measurement
Our contribution

We introduce a static object detector (the VP-Tree), which:

- extends the vocabulary tree of Nister and Stewenius to 3D partial-view matching
- provides a coarse pose estimate in addition to class inference

We use mobility to improve upon the static detection results by:

- formulating hypotheses about the class and pose of the object
- using a general observation model to accommodate a real depth sensor
- planning a sequence of informative views non-myopically
- minimizing the probability of an incorrect decision directly
- allowing a trade-off between probability of an error and time spent deciding
Problem formulation

- **Mobile depth sensor with pose:** $x_1 \in SE(3)$
- **Assumptions on the sensor:**
  - has accurate estimate if its own pose
  - provides accurate position estimates for stationary objects
- **Database of object models** $\mathcal{D}$
- **Objects of interest** $\mathcal{I} = \{C_1, \ldots, C_L\} \subseteq \mathcal{D}$

**Objective:** Detect if any objects of interest are present in a cluttered scene and estimate their pose

**Optimality:** Trade-off between cost of movement and expected loss of an incorrect detection

- **Note:** The detection is both against known objects from $\mathcal{D}$ and unknown clutter and background
Problem formulation

- Perform segmentation and data association
- Formulate hypotheses about the class and orientation of the unknown object:
  - Discretize orientation space \( \text{sparsely} \):
    \[
    \theta = \{r_1, \ldots, r_N\} \subset SO(3)
    \]

\( H_0 \): the object does not belong to \( \mathcal{I} \),

\( H_i \): the object is of class \( cl(H_i) := C_i \in \mathcal{I} \) with orientation \( or(H_i) := r_i \in \Theta \) for \( i = 1, \ldots, LN \),

\[
\begin{align*}
    i_c &= \text{mod}(i - 1, L) + 1, \\
    i_r &= (i - i_c)/L + 1
\end{align*}
\]
Problem formulation

- Restrict the motion to a set of sensor poses $\mathcal{X}(\rho)$ lying on a sphere $V(\rho)$ centered at the object:
  - Sensor points at the centroid of the object
  - For any $x^1, x^2 \in \mathcal{X}(\rho)$ there is a control input: $x^1 \rightarrow x^2$

- Introduce costs:
  
  $\Lambda_{ij} =$ cost for deciding $H_i$, when $H_j$ is correct

  $c(x^1, x^2) =$ cost of moving from $x^1$ to $x^2$ and taking another observation
Problem Formulation

- Sensor \( x_1 \in SE(3) \), objects of interest \( \mathcal{I} \), database \( \mathcal{D} \)
- Hypotheses \( \{H_i\}_{i=0}^{M-1} \), planning viewpoints \( \mathcal{X}(\rho) \), costs \( \{\Lambda_{ij}\} \), \( c(\cdot, \cdot) \)

Active Object Detection Problem

Given an object with unknown class \( C \) and orientation \( R \in SO(3) \), find a stopping time \( \tau \), a sequence of viewpoints \( x_2, \ldots, x_\tau \in \mathcal{X}(\rho) \), and a decision rule \( \delta \in \{0, \ldots, M - 1\} \), which minimize:

\[
E \left\{ \sum_{t=1}^{\tau} c(x_{t-1}, x_t) + \Lambda_{\delta, j^*} \right\},
\]

where \( j^*(C, R) \) is the true hypothesis.
Training Object Database

Axe  Bigbox  Broom  Brush  Flowerspray  Gastank  Handlebottle

Heavyranch  Pan  Pipe  Shovel  Spadefork  Spraybottle  Watercan

Wreckbar  Apples  Bathroomkit  Bottles  Cups  Glasses  Vases
- **Combined** detection and pose estimation for 3D objects via partial-view matching
- Bag-of-words model: a sparse vector of occurrence counts of words
- A word: local surface features associated with a pose and a class

**Training:**
- Discretize the viewsphere $V(\rho)$ into a set of viewpoints
- Extract templates from each model in the database
- Select keypoints uniformly and extract local surface descriptors (FPFH)
- Define words from the feature sets via $k$-means clustering
- **Combined** detection and pose estimation for 3D objects via partial-view matching
- Bag-of-words model: a sparse vector of occurrence counts of words
- A word: local surface features associated with a pose and a class

**Training:**
- Discretize the viewsphere $V(\rho)$ into a set of viewpoints
- Extract templates from each model in the database
- Select keypoints uniformly and extract local surface descriptors (FPFH)
- Define words from the feature sets via $k$-means clustering
**VP-Tree**

- **Combined** detection and pose estimation for 3D objects via partial-view matching
- Bag-of-words model: a sparse vector of occurrence counts of words
- A word: local surface features associated with a pose and a class

**Training:**
- Discretize the viewsphere $V(\rho)$ into a set of viewpoints
- Extract templates from each model in the database
- Select keypoints uniformly and extract local surface descriptors (FPFH)
- Define words from the feature sets via $k$-means clustering
**Combined** detection and pose estimation for 3D objects via partial-view matching

- Bag-of-words model: a sparse vector of occurrence counts of words
- A word: local surface features associated with a pose and a class

**Training:**

- Discretize the viewsphere $V(\rho)$ into a set of viewpoints
- Extract templates from each model in the database
- Select keypoints uniformly and extract local surface descriptors (FPFH)
- Define words from the feature sets via $k$-means clustering
VP-Tree

- **Combined** detection and pose estimation for 3D objects via partial-view matching
- Bag-of-words model: a sparse vector of occurrence counts of words
- A word: local surface features associated with a pose and a class

Training:
- Discretize the viewsphere $V(\rho)$ into a set of viewpoints
- Extract templates from each model in the database
- Select keypoints uniformly and extract local surface descriptors (FPFH)
- Define words from the feature sets via $k$-means clustering
Combined detection and pose estimation for 3D objects via partial-view matching

- Bag-of-words model: a sparse vector of occurrence counts of words
- A word: local surface features associated with a pose and a class

Training:
- Discretize the viewsphere $V(\rho)$ into a set of viewpoints
- Extract templates from each model in the database
- Select keypoints uniformly and extract local surface descriptors (FPFH)
- Define words from the feature sets via $k$-means clustering
**VP-Tree**

- **Combined** detection and pose estimation for 3D objects via partial-view matching
- Bag-of-words model: a sparse vector of occurrence counts of words
- A word: local surface features associated with a pose and a class

**Training:**
- Discretize the viewsphere $V(\rho)$ into a set of viewpoints
- Extract templates from each model in the database
- Select keypoints uniformly and extract local surface descriptors (FPFH)
- Define words from the feature sets via $k$-means clustering
Testing:
- Select keypoints and extract features from the query
- Quantize the features into words
- Choose the closest document from the tree using NN search

**Note:** The VP-Tree works with instances (not classes) but the planning framework is independent and can be combined with a different static detector.
Observation model

- Sensor measurements are affected by occlusions and noise
- Maintain probabilities over the hypotheses
- Bayesian setting requires statistics for the operation of the sensor and the VP-Tree
- **Observation** \( z \): VP-Tree output (class and pose of closest match) instead of a pointcloud
- Given \( x \in \mathcal{X}(\rho), H_i \), compute the observation model:

\[
h^x_i(z) := \mathbb{P}(Z = z \mid x, H_i)
\]
Bayesian hypotheses update

- **Dynamic programming** formulation to solve the Active Object Detection Problem
- State at time $t$:
  \[ x_t \in \mathcal{X}(\rho) \]
  \[ p_i(t) = \mathbb{P}(H_i \mid x_{1:t}, z_{1:t}) \quad i = 0, \ldots, M - 1 \]
- Use observation model to incorporate a new measurement $z_{t+1}$:
  \[ p_i(t) = \mathbb{P}(H_i \mid x_{1:t}, z_{1:t}) \]
Bayesian hypotheses update

- **Dynamic programming** formulation to solve the Active Object Detection Problem
- State at time $t$: 
  \[ x_t \in \mathcal{X}(\rho) \]
  \[ p_i(t) = \mathbb{P}(H_i \mid x_{1:t}, z_{1:t}) \quad i = 0, \ldots, M - 1 \]
- Use observation model to incorporate a new measurement $z_{t+1}$:

\[ p_i(t) = \mathbb{P}(H_i \mid x_{1:t}, z_{1:t}) \]
Bayesian hypotheses update

- Dynamic programming formulation to solve the Active Object Detection Problem
- State at time $t$: 
  \[
  x_t \in \mathcal{X}(\rho) \quad \text{and} \quad p_i(t) = \mathbb{P}(H_i \mid x_{1:t}, z_{1:t}) \quad i = 0, \ldots, M - 1
  \]
- Use observation model to incorporate a new measurement $z_{t+1}$:
Bayesian hypotheses update

- **Dynamic programming** formulation to solve the Active Object Detection Problem

- State at time $t$:
  \[
  x_t \in \mathcal{X}(\rho)
  \]
  \[
  p_i(t) = \mathbb{P}(H_i \mid x_{1:t}, z_{1:t}) \quad i = 0, \ldots, M - 1
  \]

- Use observation model to incorporate a new measurement $z_{t+1}$:

\begin{align*}
  h_i^{x_{t+1}}(z_{t+1}) &= \mathbb{P}(Z_{t+1} = z_{t+1} \mid x_{t+1}, H_i) \\
p_i(t+1) &= \mathbb{P}(H_i \mid x_{1:t+1}, z_{1:t+1})
\end{align*}
Active hypothesis testing via DP

- Terminal cost of the DP for fixed $\tau$

$$J_\tau(x_\tau, p(\tau)) = \min_{\delta \in \{0, \ldots, M-1\}} \mathbb{E} \Lambda_{\delta, j^*} = \min_{\delta \in \{0, \ldots, M-1\}} \sum_{j=0}^{M} \Lambda_{\delta, j} p_j(\tau)$$

- Stage costs for $t = 0, \ldots, (\tau - 1)$:

$$J_t(x_t, p(t)) = \min_{v \in \mathcal{X}(\rho)} \left\{ c(x_t, v) + \mathbb{E}_{Z_{t+1}} J_{t+1}(v, T(p(t), v, Z_{t+1})) \right\},$$

where $T(p, x, z)$ is the Bayes update operator

- Infinite-horizon DP:

$$J(x, p) = \min \left\{ \min_{\delta \in \{0, \ldots, M-1\}} \sum_{j=0}^{M-1} \Lambda_{\delta, j} p_j, \right\}$$

$$\min_{v \in \mathcal{X}(\rho)} \left\{ c(x, v) + \mathbb{E}_Z \{ J(v, T(p, v, Z)) \} \right\}$$

- Approximate policy $\hat{\mu} : (x_t, p(t)) \rightarrow x_{t+1}$ computed off-line
Performance analysis

- $\rho = 1\text{m}$, $\mathcal{X}(\rho) = 42$ viewpoints in the upper hemisphere of $V(\rho)$

- Costs:

$$\Lambda_{ij} = \begin{cases} 75, & i \neq j \\ 0, & i = j \end{cases}$$

$$c(x^1, x^2) = \gcd(x^1, x^2) + c_0,$$

where $\gcd(\cdot, \cdot)$ is the great-circle distance between $x^1, x^2 \in V(\rho)$ and $c_0 = 1$ is a fixed measurement cost.

- $\mathcal{I} = \{\text{Handlebottle}\}$

- Hypotheses:

$$H_0 = \text{The object is not a Handlebottle}$$

$$H_i = \text{The object is a Handlebottle with yaw } (i - 1)60 \text{ deg}$$

for $i = 1, \ldots, 6$
Performance analysis

- Compare four approaches for selecting viewpoints from $\mathcal{X}(\rho)$ using the same static detector (VP-Tree):
  - **Static**: take a single measurement from the starting viewpoint and decide
  - **Random**: random walk on the viewsphere, which avoids revisiting viewpoints
  - **GMI**: greedy policy maximizing mutual information (widely used in practice)
  - **AHT**: our active hypothesis testing policy

- **Metrics**:
  - **Accuracy**
  - **Measurement cost**: $\sum_{t=1}^{\tau} c_0$
  - **Movement cost**: $\sum_{t=2}^{\tau} gcd(x_{t-1}, x_t)$
  - **Decision cost**: $\Lambda_{\delta,j^*}$
Real experiments (Real AHT):

- Simulations were recreated in a real environment
- Asus Xtion RGBD camera attached to the wrist of a PR2 robot

<table>
<thead>
<tr>
<th></th>
<th>Accuracy (%)</th>
<th>Avg Number of Measurements</th>
<th>Avg Movement Cost</th>
<th>Avg Decision Cost</th>
<th>Avg Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True Positive</td>
<td>True Negative</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td>54.21</td>
<td>89.87</td>
<td>1.00</td>
<td>0.00</td>
<td>30.52</td>
</tr>
<tr>
<td>Random</td>
<td>70.41</td>
<td>92.33</td>
<td>2.76</td>
<td>2.21</td>
<td>19.85</td>
</tr>
<tr>
<td>GMI</td>
<td>78.73</td>
<td>94.64</td>
<td>2.79</td>
<td>2.25</td>
<td>14.25</td>
</tr>
<tr>
<td>AHT</td>
<td>84.39</td>
<td>93.2</td>
<td>2.35</td>
<td>1.85</td>
<td>11.71</td>
</tr>
<tr>
<td>Real AHT</td>
<td>76.25</td>
<td>98.05</td>
<td>2.53</td>
<td>2.44</td>
<td>15.48</td>
</tr>
</tbody>
</table>

More details
Accuracy of orientation estimates

- Object orientations in real scenes are not discretized
- An orientation refinement step is performed:
  - the object model at the hypothesized pose is aligned to the observed surface using ICP
- The final decision includes both a class and a continuous pose estimate
- Evaluate the accuracy of the continuous orientation estimates
- Distance metric on SO(3) to measure the error between estimate and ground truth orientations

\[ d(q_1, q_2) = \cos^{-1}(2\langle q_1, q_2 \rangle^2 - 1) \]
Accuracy of orientation estimates

- $\mathcal{I} = \{\text{Watercan}\}$
- Same $\{\Lambda_{ij}\}$ and $c(\cdot, \cdot)$
- Ground-truth yaw and roll varied: $0^\circ : 7.5^\circ : 360^\circ$
- Discretize yaw in 6 bins and roll in 4 bins:

$H_0 =$ The object is not a Watercan

$H_i =$ The object is a Watercan with:

- Roll $= 90i_r^\circ$ and Yaw $= 60i_y^\circ$
- $i = 1, \ldots, 24$, $i_y = \text{mod}(i - 1, 6)$,
- $i_r = (i - i_y - 1)/6$
Pros & cons

Benefits:
- Observation model applicable to real sensors
- Planning framework is independent of the static detector
- Viewpoint selection is non-myopic
- Probability of an incorrect decision is minimized directly

Drawbacks:
- Extensive off-line training phase
- No explicit mechanism to handle occlusions
- The sequence of viewpoints is selected with respect to a single object
- Requires accurate sensor pose estimates
Conclusions

- Introduced the VP-Tree, which combines detection and pose estimation for 3D objects

- Proposed a non-myopic planning approach, which trades off the cost of movement and the expected loss of an incorrect decisions

- Verified the validity of our approach both in simulations and real-world experiments

- Our approach outperforms greedy viewpoint selection methods and provides a significant improvement over static detection

- Future work:
  - Introduce an occlusion model and use it in the planning stage
  - Obtain a non-myopic policy, which is recomputable on-line
  - Introduce sensor dynamics in the active hypothesis testing
Hypotheses Testing Framework for Active Object Detection

Nikolay Atanasov\textsuperscript{1}, Bharath Sankaran\textsuperscript{2}, George J. Pappas\textsuperscript{1}, and Kostas Daniilidis\textsuperscript{1}

\textsuperscript{1} GRASP Laboratory, University of Pennsylvania
\textsuperscript{2} CLMC Laboratory, University of Southern California
Thank you!
Active hypothesis testing as a POMDP

- Replace decision rule $\delta$ by a set of sink states $A = \{a_0, \ldots, a_{M-1}\}$
- Redefine the cost of movement for $s^1, s^2 \in \mathcal{X}(\rho) \cup A$:

$$
c'(p, s_1, s_2) = \begin{cases} 
  c(s_1, s_2), & s_1, s_2 \in \mathcal{X}(\rho) \\
  \sum_{j=0}^{M-1} p_j \Lambda_{s_2,j}, & s_1 \in \mathcal{X}(\rho), s_2 \in A \\
  0, & s_1 = s_2 \in A \\
  \infty, & \text{otherwise}
\end{cases}
$$

- Redefine the state-transition function:

$$
T'(p, s, z) = \begin{cases} 
  T(p, s, z), & s \in \mathcal{X}(\rho) \\
  p, & s \in A
\end{cases}
$$

- Bellman optimality equation for a POMDP:

$$
J(s, p) = \min_{s' \in \mathcal{X}(\rho) \cup A} \left\{ c'(p, s, s') + \mathbb{E}_Z \{ J(s', T'(p, s', Z)) \} \right\}
$$

- Approximate policy using a point-based POMDP algorithm (SARSOP): $\hat{\mu} : \mathcal{X}(\rho) \cup A \times [0, 1]^M \rightarrow \mathcal{X}(\rho) \cup A$
## Extended simulation results

### True Hypothesis

<table>
<thead>
<tr>
<th></th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>H4</th>
<th>H5</th>
<th>H6</th>
<th>H0</th>
<th>Avg Number of Measurements</th>
<th>Avg Movement Cost</th>
<th>Avg Decision Cost</th>
<th>Avg Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>60.35</td>
<td>3.86</td>
<td>1.00</td>
<td>2.19</td>
<td>1.48</td>
<td>2.19</td>
<td>28.92</td>
<td>1.00</td>
<td>0.00</td>
<td>29.74</td>
<td>30.74</td>
</tr>
<tr>
<td>H2</td>
<td>5.53</td>
<td>53.90</td>
<td>2.19</td>
<td>1.00</td>
<td>1.48</td>
<td>1.95</td>
<td>33.94</td>
<td>1.00</td>
<td>0.00</td>
<td>34.57</td>
<td>35.57</td>
</tr>
<tr>
<td>H3</td>
<td>4.86</td>
<td>4.62</td>
<td>51.49</td>
<td>3.90</td>
<td>2.21</td>
<td>1.24</td>
<td>31.68</td>
<td>1.00</td>
<td>0.00</td>
<td>36.38</td>
<td>37.38</td>
</tr>
<tr>
<td>H4</td>
<td>4.34</td>
<td>4.34</td>
<td>6.01</td>
<td>49.13</td>
<td>1.95</td>
<td>1.24</td>
<td>32.98</td>
<td>1.00</td>
<td>0.00</td>
<td>38.15</td>
<td>39.15</td>
</tr>
<tr>
<td>H5</td>
<td>3.88</td>
<td>1.96</td>
<td>1.24</td>
<td>2.20</td>
<td>56.11</td>
<td>1.24</td>
<td>33.37</td>
<td>1.00</td>
<td>0.00</td>
<td>32.92</td>
<td>33.92</td>
</tr>
<tr>
<td>H6</td>
<td>5.07</td>
<td>1.24</td>
<td>2.44</td>
<td>2.44</td>
<td>1.72</td>
<td>54.29</td>
<td>32.82</td>
<td>1.00</td>
<td>0.00</td>
<td>34.28</td>
<td>35.28</td>
</tr>
<tr>
<td>H0</td>
<td>0.56</td>
<td>1.09</td>
<td>3.11</td>
<td>1.93</td>
<td>0.32</td>
<td>3.13</td>
<td>89.87</td>
<td>1.00</td>
<td>0.00</td>
<td>7.60</td>
<td>8.60</td>
</tr>
</tbody>
</table>

**Overall Average Total Cost:** 31.52

### Predicted Hypothesis (%)

<table>
<thead>
<tr>
<th></th>
<th>Static</th>
<th>Random</th>
<th>Greedy MI</th>
<th>AHT</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>73.78</td>
<td>82.63</td>
<td>87.98</td>
<td>87.98</td>
</tr>
<tr>
<td>H2</td>
<td>1.96</td>
<td>0.80</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>H3</td>
<td>1.00</td>
<td>1.09</td>
<td>1.09</td>
<td>1.09</td>
</tr>
<tr>
<td>H4</td>
<td>1.48</td>
<td>1.48</td>
<td>1.47</td>
<td>1.47</td>
</tr>
<tr>
<td>H5</td>
<td>1.48</td>
<td>1.24</td>
<td>0.49</td>
<td>0.49</td>
</tr>
<tr>
<td>H6</td>
<td>1.72</td>
<td>2.11</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>H0</td>
<td>0.07</td>
<td>2.11</td>
<td>1.49</td>
<td>1.49</td>
</tr>
</tbody>
</table>

**Overall Average Total Cost:**
- Static: 31.52
- Random: 24.82
- Greedy MI: 19.29
- AHT: 15.91

---

Atanasov*, Sankaran*, Le Ny, Koletschka, Pappas, Daniilidis

Hypothesis Testing Framework for Active Object Detection, ICRA’13
## Extended real-experiment results

<table>
<thead>
<tr>
<th>True Hypothesis</th>
<th>Avg Number of Measurements</th>
<th>Avg Movement Cost</th>
<th>Avg Decision Cost</th>
<th>Avg Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>87.5</td>
<td>2.5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>H2</td>
<td>2.5</td>
<td>80.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>H3</td>
<td>7.5</td>
<td>0.0</td>
<td>72.5</td>
<td>0.0</td>
</tr>
<tr>
<td>H4</td>
<td>0.0</td>
<td>0.0</td>
<td>70.0</td>
<td>10.0</td>
</tr>
<tr>
<td>H5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>75.0</td>
</tr>
<tr>
<td>H6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>H0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.97</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Overall Average Total Cost: **20.45**


L. Mihaylova, T. Lefebvre, H. Bruyninckx, and J. D. Schutter, “Active robotic sensing as decision making with statistical methods,” 2003.


——, “Planning to perceive: Exploiting mobility for robust object detection,” in *Int. Conf. on Automated Planning and Scheduling (ICAPS)*, Freiburg, Germany, Jun. 2011.


