Multiple Hypotheses for Object Class Disambiguation from Multiple Observations

Susana Brandão
Carnegie Mellon University

Joao Costeira
Technical University of Lisbon

Multiple Hypothesis for Object Class Disambiguation from Multiple Observations

Susana Brandão
Universidade de Lisboa
Portugal
sbrandao@ece.cmu.edu

Manuela Veloso
Carnegie Mellon University
USA
mmv@cmu.edu

João P. Costeira
Universidade de Lisboa
Portugal
jpc@isr.ist.utl.pt

Abstract

The current paper addresses the problem of object identification from multiple 3D partial views, collected from different view angles, with the objective of disambiguating between similar objects. We assume a mobile robot equipped with a depth sensor that autonomously grasps an object from different positions, with no previous known pattern. The challenge is to efficiently combine the set of observations into a single classification. We approach the problem with a multiple-hypothesis filter that allows to combine information from a sequence of observations given the robot movement. We further innovate by off-line learning neighborhoods between possible hypothesis based on the similarity of observations. Such neighborhoods translate directly the ambiguity between objects, and allow to transfer the knowledge of one object to the other. In this paper we introduce our algorithm, Multiple Hypothesis for Object Class Disambiguation from Multiple Observations, and evaluate its accuracy and efficiency.

1. Introduction

We envision mobile robots capable of autonomously recognize objects in their environment. We assume that such mobile robots are equipped with a depth camera, e.g., the Kinect sensor. Such a camera provides 3D partial views of an object, namely the visible surface of the object, as illustrated in Figure 1. Our goal is to provide an algorithm to be used by mobile robots to identify an object among similar ones by gathering contiguous partial observations.

We assume that neither the number of observations nor the view angles are a-priori known. We thus propose a probabilistic approach to handle the arbitrary sequence of observations. Formally, given a library of know objects, O, we propose to estimate the object class, $\hat{o}$, from n observations $Z_{1:n} = \{\bar{z}_1, ..., \bar{z}_n\}$, $\bar{z}_i \in \mathbb{R}^L$, of the same object as seen from a sequence of n view angles, $V_{1:n} = \{\bar{v}_1, ..., \bar{v}_n\}$, $\bar{v}_i \in \mathbb{V}$, as the object $o \in O$ maximizing the a-posteriori probability $p(o|Z_{1:n}, V_{1:n})$.

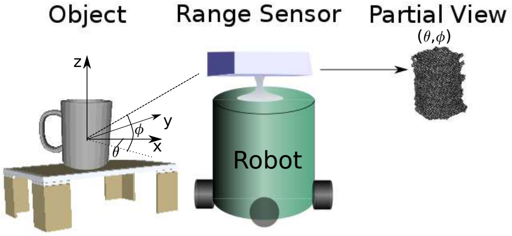


Figure 1. A mobile robot capturing a partial view of a mug from the view angle (θ, ϕ).

However, the robot does not know the sequence of view angles. While it has access at time instant n to changes in the view angle, $\bar{\Delta}_n$ through odometry, in general the initial view angle $\bar{v}_{init}$ is not known. We thus estimate the a-posteriori probability by marginalizing with respect to the initial view angle:

$$\hat{o} = \arg\max_o \sum_{v_{init} \in \mathbb{V}} p(o, \bar{v}_{init}|\bar{\Delta}_{1:n-1}, Z_{1:n}). \qquad (1)$$

Under loose assumptions we can simplify the a-posteriori probability in eq.1 by using appearance models, $p(\bar{z}|o, \bar{v})$, as building blocks. The appearance models map each partial view defined by an object o and view angle $\bar{v}$ to possible observations $\bar{z}$. By off-line learning these models, the robot can compute $\hat{o}$ during execution with little cost.

Nevertheless, we would still need to perform a dense search over all the possible partial views of all the objects. As there might be possibly infinite partial views, we resort to sampling to define hypothetical initial robot orientations. To propagate this initial hypothesis, we propose a formulation based on the Sequential Importance Resampling filter, also known as a particle filter, in a Markovian setting, [1]. These filters estimate the a-posteriori by defining a set of hypothesis, called particles. Using the sampling of the search space we can approximate the a-posteriori probability in eq. 1 at each time instant as:

$$p(o, \bar{v}_{1:n}|\Delta_{1:n-1}, Z_{1:n}) \approx \sum_{i=1}^{N_p} w_n^i \delta\left(s - s_n^i\right)) \qquad (2)$$

where each weight, w_n^i, is associated with a particle $s_n^i = (o^i, \bar{v}_n^i)$, here represented by the Dirac delta, δ, distribution defined over $s \in \mathcal{S}$, the space of all possible objects and view angles pairs. Furthermore, the weights correspond to the ratio between the probability of $p(o, \bar{v}_{1:n}|Z_{1:n}, \Delta_{1:n-1})$ evaluated at the particles center, and the density from which they were sampled, $q\left(s|Z_{1:n}, \Delta_{1:n-1}\right)$:

$$w_n^i \propto \frac{p\left(s_n^i|Z_{1:n}\Delta_{1:n-1}\right)}{q\left(s_n^i|Z_{1:n}\Delta_{1:n-1}\right)}. \qquad (3)$$

In a Markovian setting, we can update the hypothesis probability iteratively by taking into account the probability in the previous time step, a prediction of a new observation based on changes in the robot position and the new observation itself. A general formulation for a particle filter in object recognition would be:

Generate M random initial conditions :

Hypothesize M pairs of possible objects and initial orientations, $s_1^i = (o_i, \bar{v}_i)_1, i = 1, ..., M$;

For each time step, j, until Convergence :

1. Estimate a new observation, $\bar{z}_j$;
2. Propagate particles, $s_j^i = s_{j-1}^i + (0, \bar{\Delta}_{j-1})$;
3. Update the probability for each hypothesis;
4. Bootstrap by replacing low by high probability hypothesis;
5. Estimate the object identity;
6. Check convergence.

The inclusion of the object class in the state vector differentiates our problem from more common uses of particle filters, such as, tracking and localization. In particular, the object class separates the state space so that not all the partial views are reachable by a given particle. For example, if a particle is associated with object o' and view angle $\bar{v}'$, the above algorithm can update the view angle according to the robot movement, but not the object class. As hypothesis can disappear in the bootstrapping step, if at some point there is no hypothesis associated with a given object, the object is no longer considered in subsequent iterations of the algorithm.

To ensure that the whole search space is reachable at each stage of the algorithm, we take advantage that our objects are actually similar to one another. We thus contribute a multiple view object identification algorithm that, while leveraging on a Sequential Importance Resampling framework, uses an off-line learned similarity between objects and view angles. The similarity is used to find high probability hypothesis during the bootstrap and is based on observations only, i.e., independent of objects and view angles.

Our proposed bootstrap method is illustrated in 2 with an example with two very similar objects: a cup with no handle from a mug. In the first step, Figure 2(a), we map the current hypothesis into the observation space. In the second step, Figure 2(b), we search for similar observations. Finally, Figure 2(c), we inverse mapping to find all view angles that can be associated with those observations.

In the current paper we empirically show in different datasets of similar objects that the proposed approach prevents misclassifications and reduces the number of particles needed to cover the complete set of objects.

2. Related Work

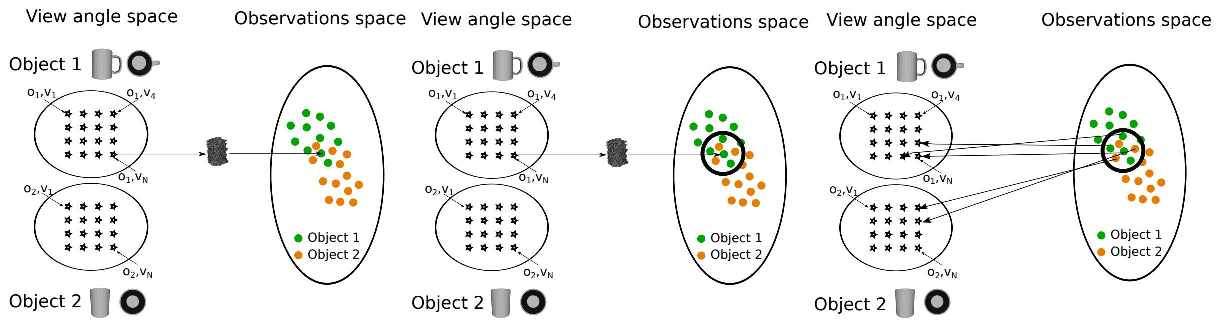
There are several approaches for merging information from multiple consecutive observations. We here highlight those related to ours either by using the same input data or by using a sampling approach and a baysian setting.

The information from consecutive 3D partial views can be used to construct complete 3D models, e.g., with the KinectFusion algorithm, [8]. However, constructing a model does not solve the classification problem. Even with an enlarged partial view, the robot would still need to represent and classify the object, e.g., using [4]. However, it would have to see the full object before attempting to recognize it. Our algorithm can provides at each moment an estimative of the object class.

Multiple-hypothesis approaches have also been extensively used for object tracking in 2D color videos, e.g., in [11], or localization of real robots actuating on the environment [5]. However, in both applications, hypothesis do not include the object class and the localization or tracking algorithms assume that the class is provided by an independent algorithm.

Notwithstanding, some tracking algorithms, such as [10, 6], have been extended to include object classification. However, in neither the examples the similarity between partial views of multiple objects is used.

The current work differs greatly from the previous examples in the sense that we use an a-priori known map between the view angle and appearance to improve our recognition, in a manner similar to what can be seen in Active Monte Carlo Recognition (AMCR) [7]. The latter introduces an algorithm for object recognition based on multiple-hypothesis, as well as the notion that when dealing with sequential class estimation there are two spaces: one associated with the object appearance and another associated with the observer dynamics. The authors also pro-



(a) Map hypothesis. (b) Find similar partial views. (c) Map back to view angles.

Figure 2. Example of the proposed bootstrap method, see the text for more details.

pose a mapping between the two, which reflects the notion of similarity between different state-vectors based on the similarity between objects. However, AMCR uses the mapping to establish a relation between two sets of particles, one that moves in the object appearance space, and the other that moves on the observers space. In the current work we re-introduce this concept of two spaces connected by an off-line mapping. However, we only require a set of particles on the observers space, as we use the mapping to infer distances from the appearance space. Furthermore, we propose more complex appearance models and similarities than those used in [7].

Finally, there is a rich literature on hypothesis testing for active object recognition, e.g., [2] and references therein. In the active context, object recognition is also formulated in a Bayesian framework, where the belief on a set of hypothesis is propagated over a sequence of actions. However, there is not a sampling approach as we here present. Instead there is an hypothesis associated with each point in the search space. Our current work is complementary to these in the sense that it provides a way to handle large search spaces.

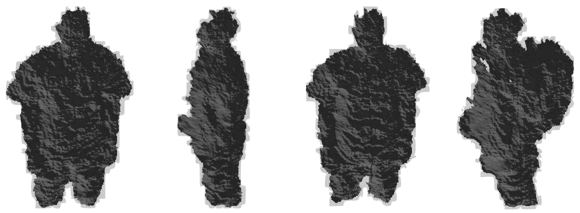
3. DATASETS

To illustrate and test our algorithm we introduce three datasets composed of very similar objects.

Throughout the paper, we illustrate the algorithm using the computer generated 3D models of the mug and cup with no handle in Figure 2. The two objects are exactly the same when seen from some view angles and are only distinguishable when the mug handle shows up in view. Thus, the two objects clearly highlight the algorithm ability of disambiguating between similar objects and the advantage of sharing knowledge between objects.

To obtain the partial views, we rendered the 3D complete models using OpenGL to obtain depth images with realistic spatial and depth resolutions as well as realistic noise [9]. We simulated the camera at 1m from the object and at view angles, $\bar{v} = [\theta, \phi]$, such that ϕ is equal to 0^o and $\theta = 12^o, 24^o, 36^o, ..., 360^o$.

We further test the performance of our algorithm in a similar setup but on a dataset collected with a Kinect sensor. The objects correspond now to human, spinning over himself with and without a bag-pack, as illustrated in Fig. 3. In each case we have a total of 24 different orientations, equally distributed around the z axis. For each orientation, we collected two sets of 25 observations. One set was used for learning the appearance models and the similarity between view angles, the other was used for the algorithm evaluation. The human was segmented in the depth images by background subtraction. This second dataset is used to identify whether the human is carrying the bag or not.



(a) No bag, front (b) No bag, left (c) Bag, front (d) Bag, left

Figure 3. Dataset of partial views collected with a Kinect sensor of a human in different orientation.

Finally, we show the potential for generalization of our algorithm with an example of intraclass object identification. Our third dataset contains partial views of the eight chairs represented in Figure 4 and retrieved from 3D Google warehouse. While they are similar to each other the chairs are not identical from any view angle. However, due to noise and sparse training dataset, it is not always possible to correctly identify an object. The partial views were obtained from a manner similar to that described for the mug and cup with no handle example. We collected three sets of partial views, one for training, one for learning similarities and the third as the testing dataset. The testing dataset contains partial views gather from 127 different view angles per chair, while the training dataset has only 13 per chair.

Figure 4. Dataset of similar chairs.

4. Partial View Representation and Similarity

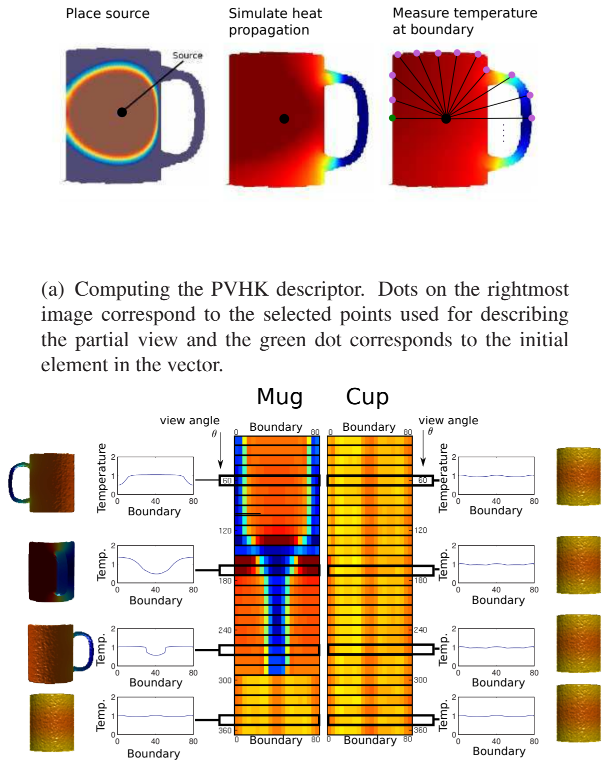
We here introduce our observation space, i.e., we introduce the partial view representation, the distance metric and similarity between partial views.

4.1. Descriptor

We represent partial views using the Partial View Heat Kernel (PVHK) representation, [3]. This representation conveys information on the distance over the surface between a point in the center of the partial view and points in the boundary. Furthermore, PVHK provides a unique descriptor to a given shape; is resilient to sensor noise; and varies smoothly with changes in the view angle. Our choice of representation was motivated by the latter property, because if the descriptor changes smoothly with changes in the view angle, we do not need to keep a dense set of partial views in the training dataset.

The descriptor itself builds upon the solution of a heat diffusion equation over the object surface and, as we illustrate in Figure 5(a), and as described in [3], it can be computed by taking three steps. First we place a heat source at the center of the object surface, at point $source$ and $t = 0$; second we simulate the heat diffusion over the surface; third, we access the temperature at some selected points c_i, $i = 1, ..., K$, in object boundary at some time $t = t_m$ which depends on the object size. The selected points in the boundary correspond to 80 points, separated by an angle of $\pi/40$ measured from the source. The descriptor is then a vector $\bar{z} \in \mathbb{R}^{80}$ where z_i is the temperature at $(i-1) \times \pi/40$, and z_1 corresponds to the leftmost point in the x-y axis that passes in the heat source as shown in Figure 5(a).

We illustrate the descriptor smoothness in Figure 5(b). The figure represents the library of partial views for two objects: a mug and cup with no handle. The 3D shapes correspond to selected partial views and the colors corresponds to the temperature at $t = t_m$. The graphic associate with the 3D shapes corresponds to the PVHK descriptor. In the center, we represent the set of descriptors, each associated with a view angle, and use color to represent temperature. We note that the descriptors can be separated in four categories. The first corresponds to shapes where the handle is on the left side. The second, associated with shapes where the handle is facing the observer. The third, to shapes where the handle is on the right side. Finally, the forth represents shapes with no handle, corresponding to the cup and some view angles of the mug.



(a) Computing the PVHK descriptor. Dots on the rightmost image correspond to the selected points used for describing the partial view and the green dot corresponds to the initial element in the vector.

(b) Mug and cup library of partial views. The 3D shapes correspond to selected partial views and their color corresponds to the temperature at $t = t_m$. In the center, we represent the set of descriptors, each associated with a view angle.

Figure 5. The Partial View Heat Kernel: how to compute it, Figure (a) and examples on two objects: a mug and a cup with no handle, Figure (b). Red regions are warmer than blue ones.

4.2. Appearance Model

We compare observations using the Modified Hausdorff distance, as it allows to compare the shape of the descriptor. We first represent the descriptor as line in 2D, i.e., the descriptor $\bar{z} \in \mathbb{R}^L$ becomes a set of points $\eta = \{[1/L, z_1], [2/L, z_2], ..., [1, z_L]\}$. Then estimate the distance between two observations using eq.4.

$$d(\bar{z}, \bar{z}') = d_{\text{H}}(\eta, \eta') = \min \left\{ \sum_{x \in \eta} \inf_{y \in \eta'} \|\bar{x} - \bar{y}\|^2, \sum_{y \in \eta'} \inf_{x \in \eta} \|\bar{x} - \bar{y}\|^2 \right\} \tag{4}$$

When available, we use sets, $Z^{o,\bar{v}} = \{z_1, z_2, ...\}$, of observations to represent a single object o and view angle $\bar{v}$, $(o,\ \bar{v})$. To compare sets, we again use the Modified Hausdorff distance:

$$d(Z, Z') = \min\left\{\sum_{\bar{x}\in Z} \inf_{\bar{y}\in Z'} d(\bar{x},\bar{y}),\ \sum_{y\in Z'} \inf_{x\in Z} d(\bar{x},\bar{y})\right\}, \tag{5}$$

where Z and Z' can have different cardinalities.

We establish the probability that the set of observations Z corresponds to (o,v), $p(Z|o,\bar{v})$, by computing the distance between Z and $Z^{o,\bar{v}}$. We define the probabilities based on distances using an exponential distribution $p(Z|o,\bar{v}) = \exp\left(-d_{\mathrm{H}}(Z, Z^{o,\bar{v}})/\lambda^{o,\bar{v}}\right)/\lambda^{s}$. In this context $\lambda^{o,\bar{v}}$ represents the average inner distance between a descriptor of a partial view associated with object o and view angle $\bar{v}$, and the set of descriptors associated with the same partial view:

$$\lambda^{o,\bar{v}} = \sum_{z'\in Z^{o,\bar{v}}} d_{\mathrm{H}}(\{z'\}, Z^{o,\bar{v}}\backslash\{z'\})/|Z^{o,\bar{v}}| \tag{6}$$

4.3. Similarity

We define similarity, μ, between two pairs $(o,\bar{v})$ and $(o',\bar{v}')$ respectively, based on the probability that we would identify a set of descriptors from the former as being from as the latter:

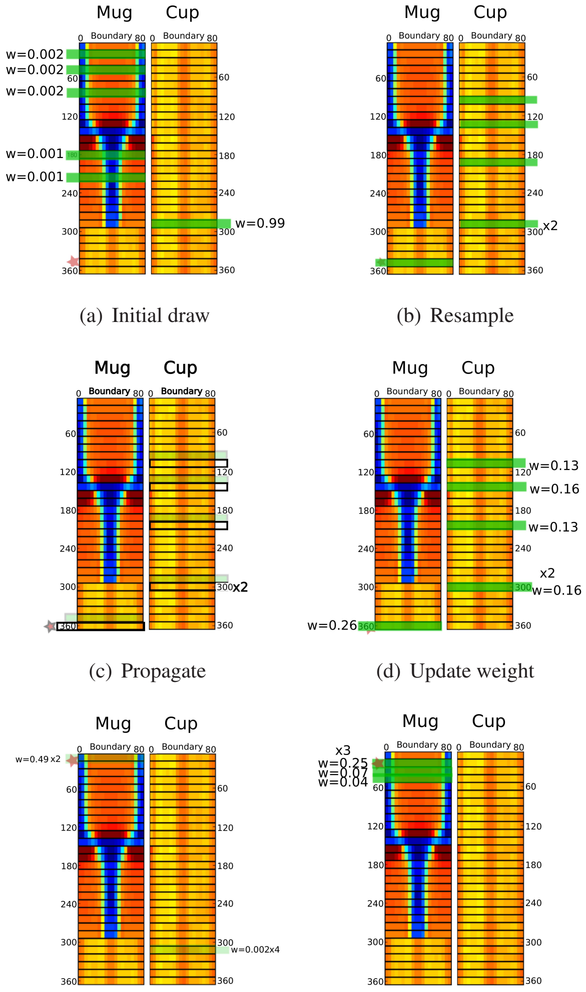
$$\mu((o,\bar{v}),(o',\bar{v}')) = p((o,\bar{v})|(o',\bar{v}')) = p(Z^{o,\bar{v}}|Z^{o',\bar{v}'}) \tag{7}$$

5. Sequential Importance Resample for Object Identification

To illustrate our implementation of the filter, presented in Algorithm 1, we start with an example of the particle filter disambiguating between a mug and a cup with no handle. We then address each of the main stages of the filter.

In our example, we start with the robot facing the mug in the view angle where it looks like the cup and collects the first observation, which is represented in Figure 6(a) with a star. In the first step, the robot draws 6 random particles. Then given the first observation, we estimate the probability of each particle, which is represented by the weights w in in Figure 6(a). While most particles are associated with the mug, they have a reduced probability and correspondingly a small weight, w. But the particle associated with the cup explains the observation. So, we collect a new set in the vicinity of the high weight particle.

Figure 6(b) represents the new set of particles and we note that all the new particles are now associated with a descriptor identical to high weight particle, albeit they are associated with both objects.



(a) Initial draw

(b) Resample

(c) Propagate

(d) Update weight

(e) Disambiguate

(f) Converge

Figure 6. Disambiguating between a mug and a cup with no handle. The current observation is marked by the start. The highlighted descriptors correspond to the set of particles and their associated weight at each time step. See the text for further details.

The robot then moves and the particles are propagated accordingly, as illustrated in Figure 6(c), where we highlight the guesses for the new observation. The weights are then updated by comparing the guess with the observation retrieved, as illustrated in Figure 6(d).

In subsequent iterations, the particles coalesce around two main guesses, Figure 6(e), but when the handle becomes visible only one survives, Figure 6(f).

5.1. Initialize particles

We start the particle filter by sampling uniformly at random N initial particles, $\mathcal{S}_0 = \{s_0^1, ..., s_0^N\}$, from the set of possible objects and view angles, $\mathcal{S}$. To each particle, we associate a weight $w_0^i = 1/N$ for all $i = 1, ..., N$.

```
Input: (i) Appearance models; $p(Z|o,\bar{v})$;
(ii) Similarity $\mu((o,\bar{v})|(o',\bar{v}'))$
Output: Object identity: $\hat{o}$
Initialization
$t \leftarrow 0$;
$\mathcal{S}_0 \leftarrow sampleUniformlyAtRandom()$;
$w_0 \leftarrow uniformWeights()$;
$notConverged \leftarrow true$;
while notConverged do
    $t \leftarrow t+1$;
    $Z_t \leftarrow getNewObservation()$;
    $\Delta_t \leftarrow getDisplacement()$;
    for $i \leftarrow 0, i < N, i++$ do
        $\mathcal{S}_t \leftarrow propagateParticles(\mathcal{S}_{t-1}, \Delta_{t-1})$;
        $\tilde{w}_t \leftarrow estimateAPriori(w_{t-1}, \mathcal{S}_t)$ ;
        $restart \leftarrow checkRestart(\tilde{w}_t)$;
        if restart then
            $\mathcal{S}_t \leftarrow sampleUniformlyAtRandom()$;
        else
            $w_t \leftarrow estimateAPosteriori(\tilde{w}_t)$ ;
            $(notConverged, \hat{o}) \leftarrow$
            $checkConvergenceIdentify(\mathcal{S}_t)$
            $\mathcal{S}_t \leftarrow bootstrap(w_t, \mu)$;
        end
    end
end
```

Algorithm 1: Particle filter for object identification.

5.2. Propagate particles

At each time step t, we propagate the particles by changing the view angle according to the robot movement in the object coordinate system $\bar{\Delta}_{t-1}$.

We thus define the function $f : \mathcal{S} \times [0, 2\pi] \times [0, \pi] \rightarrow \mathcal{S}$ that updates each particle $s^i = (o^i, \bar{v}^i)$, associated with the object o^i and the view angle v^i, given a robot movement $\bar{\Delta}$:

$$f(s^i, \bar{\Delta}) = (o^i, \bar{v}^i + \bar{\Delta}) \tag{8}$$

5.3. Estimate the a-priori

From a new set of observations, Z_t, we estimate the a-priori probability distribution by updating each weight as $\tilde{w}_t^i = w_{t-1}^i p\left(Z_t|s_t^i\right)$.

5.4. Restarting the filter

When none of the particles explains the current set of observations, i.e., all weights $\tilde{w}$ are small, we draw a new set of particles and stop the robot movement. We restart the filter until a set of particles explains the current observation, i.e., when the sum of all the weights is higher than some threshold $Th_{restart}$.

5.5. Estimate the a-posteriori

The a-posteriori is given by normalizing across all the a-priori weights, $\tilde{w}$.

$$w_t^i = \tilde{w}_t^i / \sum_{i=1}^{Np} \tilde{w}_t^i. \tag{9}$$

5.6. Bootstrap

During bootstrap, we eliminate low weight particles and replace them with particles in the neighborhood of those with high weight.

We say that a particle has a low weight by comparing it with the weight of the highest hypothesis, w_h^{max}. The weight of an hypothesis, $h^j = (o^j, \bar{v}^j)$, corresponds to summed weight of all the particles s^i equal to h^j.

Thus, given a threshold $\tau_{boot} \in [0, 1]$, we remove from $\mathcal{S}_t$ all the particles for which $w^i / w_h^{max} < \tau_{boot}$.

We then re-populate $\mathcal{S}_t$ with the partial views more similar to the set of the remaining particles, $\mathcal{S}_t^{remain}$.

We define the similarity $\mu((o, \bar{v}), \mathcal{S})$ between the pair (o,$\bar{v}$), and a set of particles, $\mathcal{S}$, as a weighted sum over the similarity between the partial views and each particle in $\mathcal{S}$:

$$\mu(o, \bar{v}, \mathcal{S}_n^{remain}) = \sum_{i=1}^{\left|\mathcal{S}_n^{remain}\right|} w^i p\left(o, \bar{v}|s^i\right). \tag{10}$$

The new particles are then sampled using Stochastic Universal Sampling assuming a probability distribution proportional to the similarity. However, only view angles that have a similarity above some threshold σ_{min} are considered.

5.7. Test convergence and identify object

The algorithm converges when all the particles agree on the object class. By imposing such a strong consensus we prevent most false positives as, due to the bootstrap step, we ensure that as long as the observations are consistent with two objects, we have particles from the two objects.

6. Performance Evaluation

We evaluate the algorithm performance with respect to both its accuracy at identifying objects, its efficiency and its possible use in different problems. We used the human dataset for accessing the improvement on accuracy and the efficiency achieved by using our bootstrap method. We then use chairs dataset to show that the strong similarity from specific view angles is not a requirement.

6.1. Accuracy

The accuracy accesses whether the algorithm reaches the correct identification at convergence t_{conv}. We consider two experiments to access the impact of the proposed bootstrap approach on accuracy. First, we compare our algorithm with an alternative one where the bootstrap introduces new particles at view angles near those of high weight particle filters. Second, we evaluate the accuracy as a function of the number of particles replaced at each iteration.

Both experiments run on the human dataset, starting in the same initial state, with the human carrying the bag facing the camera, i.e., in a ambiguous state. Furthermore, to account for the stochastic nature of the algorithm, we repeat each experiment 30 times and the results we here present are the averages over the trials.

In the first experiment, we fix the convergence criteria and the conditions for restart and resample. The accuracy comparison between algorithms is presented in Figure 7(a). The results show that we have a significant increase in accuracy when using the similarity between observations as the criteria for sampling new particles. The impact is more noticeable when the number of particles is kept small.

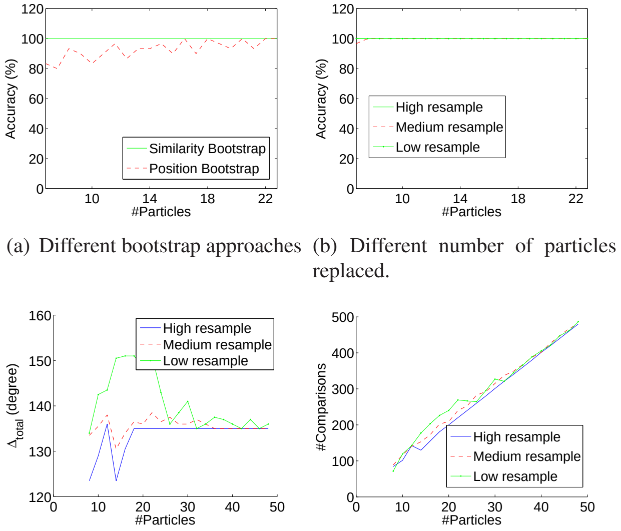
Furthermore, we note that reducing the number of particles replaced at each iteration has little to no effect in terms of recognition, as we show in Figure 7(b). The number of replaced particles is controlled by the threshold τ_{boot}, that defines the minimum ratio between a particle weight and the highest hypothesis weight so that the particle is not discarded. By increasing the necessary ratio, we are increasing the number of particles that are discarded and increasing the search of alternative partial views to explain a sequence of observations.

6.2. Efficiency

We associate efficiency to the effort required to correctly differentiate between objects. The effort can be either mechanical, evaluated in terms of the distance a robot would have to travel, and computational, evaluated in terms of the total number of comparisons between partial views. Again both were evaluated on the human dataset, using same setup as the one used to access accuracy.

The distance the robot has to travel is associated with how much of the object surface it needs to cover before identifying it. Our results, represented in Figure 7(c), show that the robot would have to cover on average 150^o of the human, i.e., it did not had to see the complete object.

The number of comparisons between partial views corresponds to the number of particles used in the experiment times the number of iterations used. Our results, represented in Figure 7(d), show that for smaller sets of particles the robot would require less comparisons using our algorithm than applying exhaustive search. There are 48 known partial views in the dataset, thus exhaustive search requires 48 comparisons. As the objects are ambiguous, we need at least two observations, i.e., 96 comparisons, to identify the object. Our results show that we can use more observations and from more view angles, and still be competitive in computational terms.



(a) Different bootstrap approaches (b) Different number of particles replaced.

(c) Change in the view angle until identification. (d) Total number of comparisons

Figure 7. Evaluating efficiency and accuracy.

6.3. Intra-class identification

Training datasets do not usually cover all the possible views of the objects. Both by acquisition, storage and evaluation constraints, we cannot expect that each view angle grasped by a robot was previously seen in training. In this case, and specially when objects are from the same class, some partial views become misclassified, as we represent in the confusion matrix in Figure 8. The figure represents the confusion matrix between the testing dataset, composed of partial views collected from 127 different view angles per chair, $\mathbb{V}_{test} = \{[45^o, 0^o], [45^o, 2.8^o], ..., [45^o, 360]\}$, and the training dataset composed of partial views from 13 view angles per chair, $\mathbb{V}_{train} = \{[45^o, 0^o], [45^o, 28.4^o], ..., [45^o, 360^o]\}$

Using Algorithm
refalg:particle with particles that could only populate the training dataset, i.e., that only covered 13 view angles of the set of chairs, we were able to recognize all the eight chairs in the view angles from the testing dataset. The results we present in Figure 9 correspond to the aggregated accuracy over all the chairs and for 10 different initial view angles. Given the initial view angle, the robot observed the whole object at intervals of 15^o degrees. At each position, the robot collected two observations and at the end of the path the robot identifies the chair. We thus cover all the possible view angles in the testing dataset, $\mathbb{V}_{test}$.

The partial view observation models assumed an exponential distribution with $\lambda = 0.08$. The similarity μ was learned using an independent dataset.

The results show that, by collecting information from

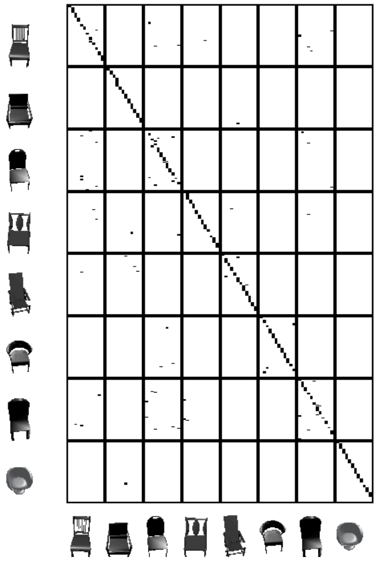


Figure 8. Confusion matrix between the testing and training dataset. See text for details.

multiple partial views and using our similarity metric, we were able to identify the objects correctly in all the cases. We were also able to do so using a sampling even sparser than the 13 view angles per object in the training dataset, as we obtained a perfect accuracy with only 7 partial views per object.

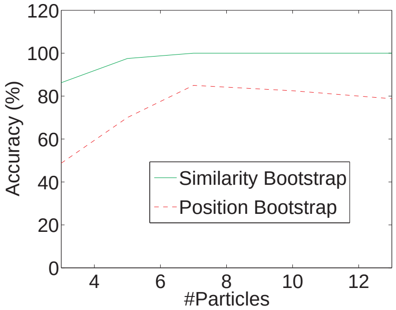


Figure 9. Aggregate accuracy as a function of the number of particles per object. See text for details.

7. Conclusions and Future Work

This paper presents a novel algorithm for the disambiguation of similar objects by collecting and combining observations from a sequence of view angles. The algorithm leverages on a similarity metric between observations to off-line learn neighborhoods between view angles. The neighborhoods are used when bootstrapping hypothesis and ensuring that they reflect the objects ambiguity.

The proposed approach has two main advantages: i) reduces the number of false positives as ambiguous observations lead to an even distribution of particles among the objects; and ii) reduces the number of particles required for estimation, as the particles can cover a much more diverse set of partial views.

The applications of the proposed algorithm are not not constrained to objects with strong similarities. Given the motivating results we here present, we intent to extent our dataset to more demanding scenarios with a larger number of objects and partial views. Larger datasets present challenging problems for example at the initialization level. While here we initialized all the hypothesis blindly and obtained a fair coverage of the search space, larger spaces might require a large number of initial hypothesis. We intent to approach this problem using the learned neighborhoods we here proposed.

Acknowledgment

Support for this research was provided by the FCT (Portuguese Foundation for Science and Technology) through the Carnegie Mellon Portugal Program under award 22129.1.5004609, by the National Science Foundation award number NSF IIS-1012733.The views and conclusions contained in this paper are those of the authors only.

References

[1] M. Arulampalam, N. G. S. Maskell, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *In IEEE Trans. on Sig. Processing*, 2002.

[2] N. Atanasov, B. Sankaran, J. L. Ny, T. Koletschka, G. J. Pappas, and K. Daniilidis. Hypothesis testing framework for active object detection. In *ICRA*, 2013.

[3] S. Brandão, J. Costeira, and M. Veloso. Partial view heat kernel descriptor. In *ICRA*, 2014.

[4] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *CVPR*, 2010.

[5] B. Coltin and M. Veloso. Multi-observation sensor resetting localization with ambiguous landmarks. In *AAAI*, 2011.

[6] J. Czyz, B. Ristic, and B. Macq. A particle filter for joint detection and tracking of color objects. *IVC*, 2007.

[7] F. V. Hundelshausen and M. Veloso. Active monte carlo recognition. In *GCAI*, 2007.

[8] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *UIST*, 2011.

[9] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 2012.

[10] K. Okada, M. Kojima, S. Tokutsu, T. Maki, Y. Mori, and M. Inaba. Multi-cue 3d object recognition in knowledge-based vision-guided humanoid robot system. In *IROS*, 2007.

[11] K. Okuma, A. Taleghani, N. D. Freitas, O. D. Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *ECCV*, 2004.