Active Random Forests: An application to Autonomous Clothes Unfolding

Anonymous ECCV submission

Paper ID 1234

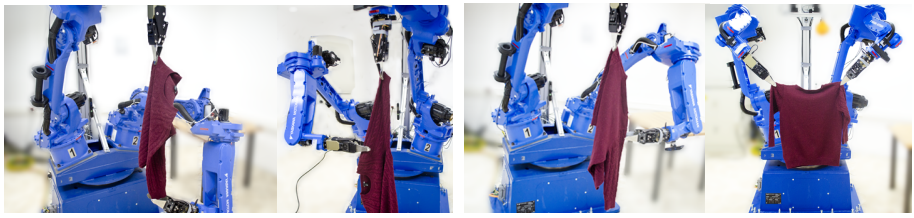
Abstract. We present *Active Random Forests*, a novel framework to address active vision problems. State of the art focuses on best viewing parameters selection like viewpoint or zooming based on single view classifiers. In contrast, we propose a multi-view classifier where the action taking process about optimally selecting viewing parameters is inherent to the classification process. This has many advantages: a) The classifier exploits the entire set of images captured at a certain time and does not simply aggregate probabilistically per view hypotheses; b) actions are made according to learnt disambiguating image features from all possible views and are optimally selected using the powerful voting scheme of Random Forests and c) the classifier can take into account the costs of its actions. The proposed framework is applied to the task of autonomously unfolding clothes by a robot, addressing the problem of best viewpoint selection in classification, pose and grasp point estimation of garments. We show great performance improvement compared to random viewpoint selection and state of the art methods.

Keywords: Active Vision, Active Random Forests

1 Introduction

Object recognition and pose estimation has been studied extensively in the literature achieving in many cases very good results [1][ ][ ]. However, single-view recognition systems are often unable to distinguish objects which depict similar appearance when observed from certain viewpoints. An autonomous system can overcome this limitation by actively collecting relevant information about the object, that is, changing viewpoint, zooming to a particular area or even interacting with the object itself. This procedure is called *active vision* and the key problem is how to optimally plan the next actions of the system (usually a robot) in order to disambiguate any conflicting evidence about the object of interest.

The majority of state of the art techniques [ ][ ][ ] in active vision share following idea: one single-view classifier is trained to recognize the type and pose of target objects, while a subsequent step uses the classification results to plan the next actions so that conflicting hypotheses are disambiguated. Although this approach is intuitive, combining features from multiple views is difficult while hypotheses from different views can be only exploited a posteriori (i.e Bayesian



(a) (b) (c) (d)

Fig. 1. Robot unfolding a shirt. a) Grasping lowest point. b) grasping 1st grasp point. c) grasping 2nd grasp point. d) final unfolding

formulations). In addition, their performance heavily relies on the performance of the single-view classifier. However, making a classifier that can generalize across views is hard, a problem which becomes even more challenging when other forms of variability are considered as well, such as illumination variations or deformations. Another problem in active vision which state of the art techniques haven't addressed yet, is the cost associated with an action. This problem appears in situations where for example moving a camera is time-consuming, and selecting arbitrary viewpoints has an impact on the efficiency of the system.

To cope with the above challenges, we propose *Active Random Forests* which can be considered as an "*active classifier*". The framework is based on classical Random Forests [2] having also the ability to control viewing parameters during on-line classification and regression. The key difference is that the classifier itself decides which actions are required in order to collect information which will disambiguate current hypotheses in an optimal way. As we will demonstrate, this combination of classification and viewpoint selection outperforms solutions which have these two components separated. Furthermore, inference is made using the entire set of images captured until a certain time, taking advantage of the various feature associations between different viewpoints. The on-line inference and action planning become extremely fast by the use of Random Forests, making the framework very suitable for real-time applications such as robotics. In summary, the main contributions of our framework are:

- **A multi-view active classifier** which combines features from multiple views and is able to make decisions about further actions in order to accomplish classification and regression tasks in an optimal way.
- **Novel decision making criteria** based on distribution divergence of training and validation sets while growing the decision trees.
- **A decision selection method** during classification and regression using the powerful voting scheme inherent to Random Forests.
- A method for taking into account the possible **costs of actions**.

To our knowledge, there is no other framework which has an action selection process inherent to the object classifier. Letting the classifier decide the next disambiguating actions gives much discriminative power to the framework, as will be shown in Section ?.

We demonstrate the proposed framework in the challenging problem of recognizing and unfolding clothes autonomously using a bimanual robot. In this problem, three objectives should be achieved: Garment categorization, pose estimation and certain grasp point detection in order to accomplish the unfolding task as shown in Figures 1 and 2(a). Furthermore, we are interested only in the best viewpoint selection as the controllable viewing parameter, although other parameters (e.g. zooming) or robot actions can be also integrated easily. In this common case of a robot grasping an object, viewpoint selection is achieved by rotating the object on the robot gripper while cameras mounted on the robot head capture images. We compare our work with the methods described in [3] and [4] using two different single-view classifiers [icra][deformable-part-model], showing the superiority of our approach.

2 Related Work

Active vision literature focuses mainly on finding efficient methods for selecting observations optimally while little attention is paid to the classifier which is kept simple. The majority of works adopted an off-line approach which consists of precomputing disambiguating features from training data. Schiele et al. [5] introduced "transinformation", the transmission of information based on statistical representations, which can be used in order to assess the ambiguity of their classifier and consequently find the next best views. Arbel et al. [6] developed a navigation system based on entropy maps, a representation of prior knowledge about the discriminative power of each viewpoint of the objects, and later, they presented a sequential recognition strategy using Bayesian chaining [7]. Furthermore, Callari *et al.* [8] proposed a model-based active recognition, using Bayesian probabilities learned by a neural network and Shannon entropy to drive the system to the next best viewpoints. Also, Sipe and Casasent [9] introduced the probabilistic feature space trajectory (FST) which can make estimation about the class and pose of objects along with the confidence of the measurements and the location of the most discriminative view. Such methods are computationally efficient both in training and testing. On the other hand, they rely mainly on their best hypotheses based on prior knowledge which can in fact have low probabilities on a test object while features from the visited viewpoints are not combined in order to make the final inference.

One of the most representative works in this direction was made by Denzler *et al.* [4] who tried to optimally plan the next viewpoints by using mutual information as the criterion of the sequential decision process. They also presented a Monte-Carlo approach for efficiently calculating this metric. Later, Sommerlade and Reid [10] extended this idea in tracking of multiple targets on a surveillance system. One drawback of this approach was that the accumulated evidence of the various viewpoints visited did not affect the viewpoint selection strategy. An improvement over this idea was made by Laporte and Arbel [3] who introduced an on-line and more efficient way of computing dissimilarity of viewpoints by using the Jeffrey Divergence weighted by the probabilistic belief of the state of

the system at each time step. This work however, combines viewpoint evidence only probabilistically using Bayesian update, and also relies on the consistent performance of the single-image classifier in estimating poses (in at least some viewpoints), which is generally very difficult, especially when dealing with deformable objects.

The work of Roy et al. [11] solves another problem in active vision, which is identifying large objects that do not fit on the camera's field of view, using Bayesian methods to handle uncertainty. The work of Zhou et al. [12] gives another perspective to active vision, making a "conditional feature sensitivity" analysis, allowing to select the most discriminative feature in an active recognition system. A recent work on active vision was made by Jia et al. [13][14] who used a similarity measure based on the Implicit Shape Model and other prior knowledge combined in a boosting algorithm in order to plan the next actions. However their similarity measure is not suitable for objects exhibiting high intra-class variations. Finally, there are some active vision applications to robotic systems in real scenarios [15] [16] [17] [18] mainly based on the previously described works, showing promising results.

Our work is based on the method proposed by Doumanoglou *et al.* [19]. In that work authors have used Random Forests for identifying garments and grasping points, while they also propose an active scheme based on POMDPs for dealing with uncertainty. Although results were promising, viewpoint selection was made only sequentially by taking nearby viewpoints, which is a sub-optimal solution while in some cases it made the whole process slow. Our work is built on the same principles, making active vision faster and more efficient by the use of Active Random Forests. In addition, we estimate the pose of the garment in order to guide the robot's gripper to grasp a desired point, which reduced grasping errors compared to the local plane fitting techniques employed in [19]. Most importantly, our framework can be easily extended to other active vision problems.

3 Problem Overview

We will describe our framework of Active Random Forests in the context of our target application: autonomously unfolding clothes using a dual-arm robot. This problem consists of picking a cloth from a table in a random configuration, recognizing it and bringing it into a predefined unfolded configuration. In order to unfold a garment, the robot has to grasp the article from two certain grasp points sequentially (e.g. the shoulders of a shirt) and hang it freely to naturally unfold by gravity, imitating the actions of a human (Fig. 1). There are three underlying problems in such procedure: Garment type classification, grasp points detection and pose estimation as shown in Figure 2(a). We will describe in short these objectives, based on [19]:

For classification, 4 basic garment types are considered: shirts, trousers, shorts and T-shirts. In order to reduce the configuration space of a garment picked up randomly, the robot first grasps its lowest point. Fig 2(c) shows the

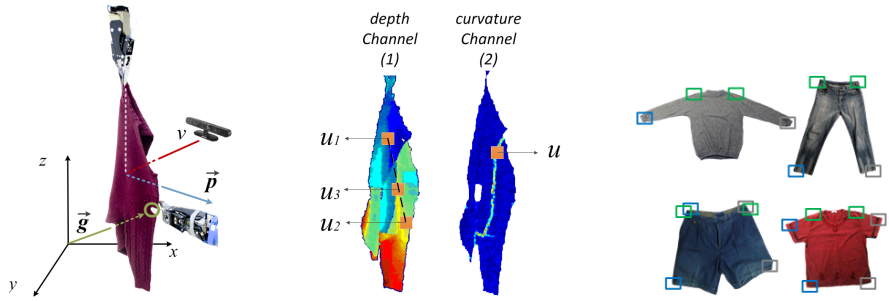


Fig. 2. Clothes Analysis. a) Grasp point and pose vectors. b) The depth and curvature channels and the random positions used in binary pixel tests. c) Possible lowest points of clothes. Gray boxes are the symmetric points of the blue ones. Green boxes show the desired grasping points for unfolding

possible lowest points which are 2 for shorts and T-shirts, and one for shirts and trousers. Therefore, the classes considered are 6, corresponding to the possible lowest points.

The grasp points used for unfolding are manually defined, shown in Fig. 2(c). The robot should sequentially find and pick these points so that a garment can be unfolded.

While pose cannot be clearly defined on deformable objects, in our problem we define it as the direction from which a desired point on the garment should be grasped by the robot arm, depicted in Figure 2(a).

In the next section we will describe how these objectives can be addressed using our Active Random Forests framework for efficient viewpoint selection.

4 Active Random Forests

As our framework is based on classical Random Forests, we will first describe the configuration of the decision trees and the training samples, then propose our decision making criteria in order to learn disambiguating actions and finally we will analyse the inference process and the real-time viewpoint selection achieved by an Active Random Forest.

4.1 Training

One training sample of Active Random Forests should consist of all the images that can be obtained from a certain training object using the possible actions and controllable viewing parameters available in the system. In our problem, only viewpoint selection is considered and therefore training samples can be represented as a tuple $(\mathbf{I}(v), c, \mathbf{g}(v), \mathbf{p}(v)), v \in \mathbf{V}$ where $\mathbf{I}$ is a vector containing the depth image of the garment, c is the class, $\mathbf{g}$ is a 2D vector containing the

position of the desired grasp point in the image, $\mathbf{p}$ is a 2D vector containing the pose of the cloth defined in the XY plane as shown in Figure 2(a) and $\mathbf{V}$ is the set of all possible viewpoints v of the garment. Viewpoints are considered around the Z axis (which coincides with the holding gripper) covering the whole 360^o degrees. We discretized the infinite viewpoint space into V equal angle bins. Vector $\mathbf{g}(v)$ is not defined if the point is not visible from viewpoint v.

Each split node of Random Decision Trees stores an array of the already seen viewpoints $\mathbf{V}'$ which also passes to its children. Starting at the root node, the only seen viewpoint is the current one ($\mathbf{V}' = \{V_0\}$). Following [19], at each node a random set of splitting tests is generated with each test containing a random seen viewpoint $v \in \mathbf{V}'$ taken from uniform distribution over $\mathbf{V}'$, a channel $C_i = \{C_1, C_2\}$, a tuple of random positions $\mathbf{M}(\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3})$ on the image (Fig. 2(b)) and a binary test $f(v, C_i, \mathbf{M}) > t$ using threshold t, selected from a pool of possible binary tests. Channel C_1 is the raw depth data of the garment as captured from a depth sensor filtered by a bilateral filter and channel C_2 is the mean curvature of the surface filtered by an average filter[19]. Also we used the binary tests proposed in [19] containing simple pixel tests in the depth or curvature channel, which showed good results and low execution time. For clarity reasons, they are shown below:

- $f(v, C_1, \mathbf{M} = \{\mathbf{u_1}, \mathbf{u_2}\}) = d_{\mathbf{u_1}}(v) - d_{\mathbf{u_2}}(v)$
- $f(v, C_1, \mathbf{M} = \{\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}\}) = (d_{\mathbf{u_1}}(v) - d_{\mathbf{u_3}}(v)) - (d_{\mathbf{u_3}}(v) - d_{\mathbf{u_2}}(v))$.
- $f(v, C_2, \mathbf{M} = \{\mathbf{u_1}\}) = |c_{\mathbf{u_1}}(v)|$

where d_u is the depth at location u, as shown in Figure 2(b).

In contrast with [19], our forest is able to make classification, grasp point detection and pose estimation using the same trees. To achieve this, we apply a hierarchical quality function for node splitting, so that the upper part of the trees perform classification and the lower part perform regression. The overall quality function has the following form:

$$Q = \alpha Q_c + (1 - \alpha) Q_r \tag{1}$$

where Q_c is a quality function for classification, Q_r a quality function for regression and α an adapting parameter. We adopt the traditional information gain using Shannon Entropy for Q_c and the corresponding information gain for continuous Gaussian distributions as defined in [21] for Q_r. Specifically, letting S be the set of training samples reaching a split node, and f be a random binary function applied to S, the latter will be split into two subsets, S_l and S_r, according to a random threshold t. Then, Q_c is the sum of the entropies of the 2 children nodes while the quality function for regression Q_r is defined as:

$$Q_r = -\sum_{i}^{\{l,r\}} \frac{|S_i|}{|S|} \sum_{v=1}^{V} \ln |\Lambda_{\mathbf{q}(v)}(S_i)| \tag{2}$$

where $\Lambda_{\mathbf{q}(v)}$ is the covariance matrix of the vectors $\mathbf{q}(v)$, with $\mathbf{q}(v) = \mathbf{g}(v)$ or $\mathbf{p}(v)$ chosen randomly. For switching between classification and regression,

the maximum posterior probability of the samples in a node is used with the parameter α being equal to:

$$\alpha = \begin{cases} 1, & \text{if } \max P(c) \leq t_c \\ 0, & \text{if } \max P(c) > t_c \end{cases} \quad (3)$$

where t_c is a predefined threshold, typically set to 0.9. At a split node, the quality function (1) is evaluated against a random set of split tests, and the one that maximizes Q is finally selected. When the maximum posterior probability of a class in a node is below t_c, the tree performs classification, otherwise performs regression of grasp point location or pose, selected randomly.

4.2 Incorporating Actions

When object recognition is not feasible by single view observations, some actions should be taken to change the current viewing conditions. Furthermore, such actions are also needed when searching for a particular region of the object which is not visible in the current view. In contrary, actions may have an execution cost which should be taken into account in the selection process. Therefore, the criteria for making a decision about an action should be the informativeness of the current observations, the belief about the visibility of the region of interest in the current observations and the execution cost of a potential action.

The analysis in section 4.1 was made taking into account the set of already seen viewpoints of the object $\mathbf{V}'$, which at the root node contains only the current view V_0. The split nodes keep splitting the training set for a few times using this view, until, in some cases in certain depths, the current view stops being informative and the tree starts overfitting on the training samples reached the nodes. The moment at which such behaviour appears is crucial and requires a further action to be taken (or another viewpoint to be seen in our problem) so that more disambiguating information can be collected. We achieve this by using a validation set in parallel with the training set and measure the divergence of the posterior distributions among these two sets in a node.

Specifically, we split the initial training set S into 2 equal-sized random subsets, with S_T being the actual training set and S_D the validation set. For finding the best split candidates at a node only the training set is considered. However, the validation set is also split using the best binary test found and is passed to the left or right child accordingly. Thus, at node j, the sample sets that arrive are the training set S_T^j and the validation set S_D^j.

In order to determine the presence of overfitting, the training set is compared against the validation set at each split node. For measuring the divergence of two sets, we have experimented with two alternative metrics which were tested and compared in the experimental results (Section ?). The first is the *Hellinger distance*[22], a statistical measure defined over validation set S_T^j and S_D^j as:

$$HL(S_T^j \| S_D^j) = \frac{1}{\sqrt{2}} \sqrt{\sum_{c=1}^{6} \left(\sqrt{P_{S_T^j}(c)} - \sqrt{P_{S_D^j}(c)} \right)^2} \quad (4)$$

when comparing the class distributions of the training set S_T^j and validation set S_D^j. $P_S(c)$ is the class probability distribution of the set S. The Hellinger distance satisfies the property $0 \leq HL \leq 1$ and it takes its lowest value 0 when training and validation set distributions are identical and its maximum value 1 when one distribution is 0 when the other is positive. Similarly, assuming that grasp point and vectors at node j are normally distributed variables, the averaged Hellinger distance over the possible viewpoints is:

$$HL^2(S_T^j \| S_D^j; \mathbf{q}) = \frac{1}{V} \sum_{v \in \mathbf{V}} 1 - \frac{\left(|\Lambda_{\mathbf{q}(v)}(S_T^j)||\Lambda_{\mathbf{q}(v)}(S_D^j)| \right)^{\frac{1}{4}}}{|A|^{\frac{1}{2}}} \exp\{-\frac{1}{8}\mathbf{u}^T A^{-1}\mathbf{u}\} \quad (5)$$

where

$$\mathbf{u} = \boldsymbol{\mu}_{\mathbf{q}(v)}(S_T^j) - \boldsymbol{\mu}_{\mathbf{q}(v)}(S_D^j) \quad (6)$$

$$A = \frac{1}{2}\left(\Lambda_{\mathbf{q}(v)}(S_T^j) + \Lambda_{\mathbf{q}(v)}(S_D^j) \right) \quad (7)$$

and $\boldsymbol{\mu}_{\mathbf{q}(v)}()$ is the mean value over vectors $\mathbf{q}$ ($= \mathbf{g}(v)$ or $\mathbf{p}(v)$) in viewpoint v.

The other metric is the so called *Jensen–Shannon divergence* which measures the information divergence of two probability distributions and is actually a symmetric version of the *Kullback–Leibler* divergence. Measuring the class distribution divergence of training and validation sets, Jensen–Shannon divergence is defined as:

$$JS(S_T^j \| S_D^j) = \frac{1}{6} \sum_{c=1}^{6} P_{S_T^j}(c) \log \frac{P_{S_T^j}(c)}{P_m(c)} + P_{S_D^j}(c) \log \frac{P_{S_D^j}(c)}{P_m(c)} \quad (8)$$

where P_m is the average distribution of S_T and S_D. Again, JS satisfies the property $0 \leq JS \leq 1$, where 0 indicates identical distributions while 1 indicates maximum divergence. For measuring the information divergence of our continuous variables over two sets, we substitute (8) with multi-variate Gaussian distributions and compute the average over viewpoints $\mathbf{V}$, which results in:

$$JS(S_T^j \| S_D^j; \mathbf{q}) = \frac{1}{2V} \sum_{v \in \mathbf{V}} \left(\mathbf{u}^T \left(\Lambda_{\mathbf{q}(v)}(S_T^j)^{-1} + \Lambda_{\mathbf{q}(v)}(S_D^j)^{-1} \right) \mathbf{u} + tr\left(\Lambda_{\mathbf{q}(v)}(S_T^j)^{-1} \Lambda_{\mathbf{q}(v)}(S_D^j) + \Lambda_{\mathbf{q}(v)}(S_D^j)^{-1} \Lambda_{\mathbf{q}(v)}(S_T^j) - 2\mathbf{I} \right) \right) \quad (9)$$

where $\mathbf{u}$ is the same used in (6). More details about (9) can be found in [22].

When the divergence of the training and validation set Δ ($= JS$ or HL) is above a threshold t_Δ, the node becomes a *decision node* and an action should be taken in order to change the viewing parameters, which in our problem is a rotation of the robot gripper in order to change the viewpoint v. Therefore, in a decision node the whole set of possible viewpoints $\mathbf{V}$ is considered in the selection of the best random test.

There are two main directions regarding the selection criteria of a new viewpoint, from which only the first has been studied in the literature:

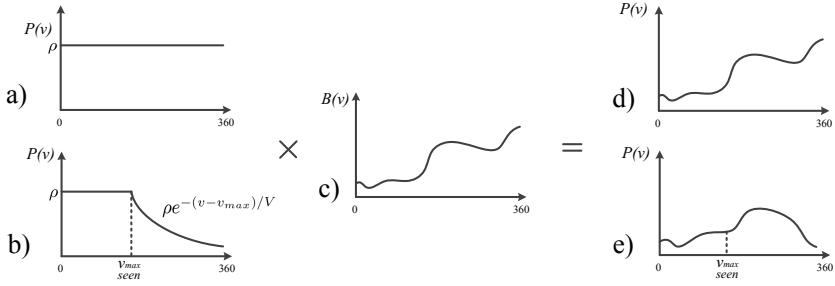


Fig. 3. Viewpoint distribution for random test selection. a) Uniform distribution, b) weighted distribution, c) Visibility map, d) Final distribution using (a), e) final distribution using (b).

- Viewpoints can be reached at the same cost, while when moving from viewpoint i to viewpoint j, no further information can be captured from the viewpoints in between.
- Moving from viewpoint i to viewpoint j has a cost relative to the distance of i and j, while when moving from i to j, images from the intermediate viewpoints can be also captured without additional cost.

Our problem belongs to the second category, however we consider also the first case for comparison with previous works. Assuming no cost for the transition between viewpoints, the distribution of $\mathbf{V}$ used for randomly selecting a new viewpoint in a decision node is uniform (Fig. 3(a)). For our problem however, it is more realistic to assume a cost relevant to the degrees of rotation of the gripper needed to see a viewpoint, while during rotation, all intermediate images can be captured. The distribution of $\mathbf{V}$ in a decision node in this case is depicted in Figure 3(b). If the furthest viewpoint seen so far is v_{max}, then all viewpoints $v = 1...v_{max}$ are also seen and have equal distribution ρ to be selected, as no action is required. The next viewpoints have an exponential distribution $\rho e^{-(v-v_{max})/V}$ for $v = (v_{max}+1)...V$. Parameter ρ can be easily found by solving $\sum_{v=1}^{V} P(v) = 1$. Using such distribution, further viewpoints are less likely to be selected by a split test. Modifying the distribution from which the viewpoints v are randomly selected and tested, is equivalent to weighting them.

One other issue when searching for a particular region of an object like a grasp point on a cloth, is that it may be invisible in the acquired images. In this case, a viewpoint is needed so that not only it disambiguates the current belief, but it also makes the particular region visible. The visibility of samples reaching a node can be measured by the vectors in $\mathbf{g}(v)$ where viewpoints with non-visible grasp points are not defined. To achieve this, a visibility map B is constructed as:

$$B(v) = \frac{\sum_{s \in S^j} b(s,v)}{\sum_{v' \in \mathbf{V}} \sum_{s \in S^j} b(s,v')}, \; b(s,v) = \begin{cases} 1, & \text{if } \mathbf{g}_s(v) \text{ exists} \\ 0, & \text{if } \mathbf{g}_s(v) \text{ is not defined} \end{cases} \tag{10}$$

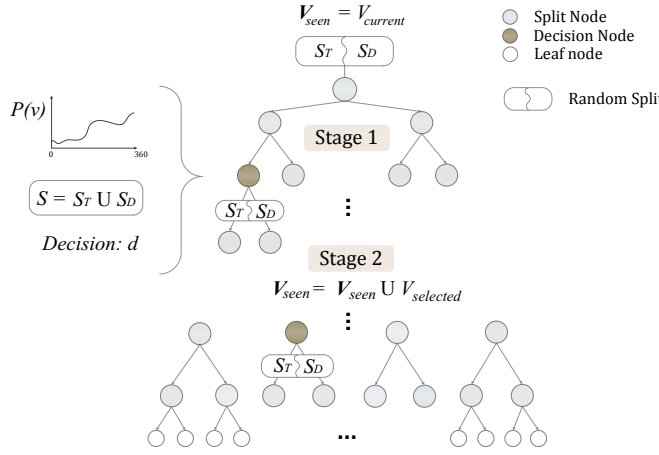


Fig. 4. Active Random Forests and decision making. Method overview.

An example is shown in Figure 3(c). When visibility is low in the collected views, $B(v)$ is multiplied with the current distribution of the set $\mathbf{V}$ calculated previously, so that preference is given to the viewpoints where the grasp point is more probable to be visible, as shown in Fig. 3(d)–(e).

A decision node can now select the next best viewpoint v_{best} randomly evaluating binary tests and selecting viewpoints taken from the calculated distribution $P(v)$. The random tests are evaluated on the whole set $S = S_T^j \cup S_D^j$. This results in finding the best viewpoint v_{best} which optimally separates the diverging samples and helps the tree disambiguate its hypotheses. The samples that arrive at each child of the decision node are again split randomly into training and validation sets and the tree enters the next stage where again only the seen viewpoints are considered, which are now increased by 1 (Fig. 5). That is: $\mathbf{V}' = \mathbf{V}'_{parent} \cup v_{best}$. This stage follows the same hierarchical quality function (1) and the tree continues growing until another decision node is encountered or a leaf node is created. The criteria of making a leaf node is setting a minimum number of samples allowed in a node. Finally, in the leaf nodes, along with the class distribution $P(c)$ we store only the first 2 modes of $\mathbf{g}(v)$ and $\mathbf{p}(v)$ per class as in [23], weighted by the class probability, for memory efficiency during inference.

4.3 Inference

In order to make an inference using an Active Random Forest, the current arbitrary view is captured and starts traversing the trees. Although in some trees the current view can reach a leaf node, in other trees it reaches a decision node where other viewing parameters are needed or another viewpoint is required. The requests from all the trees are accumulated in a voting array, and the action needed in order to take the most voted camera parameters or viewpoint

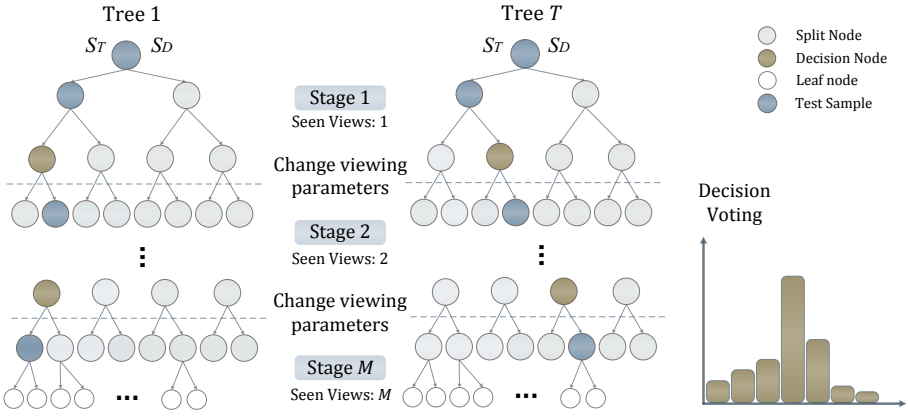


Fig. 5. Active Random Forests and decision making. Method overview.

is executed. The trees that requested these parameters can grow further, and some of them may reach a leaf node. The other trees keep their requests for the next iteration. In each iteration, the most voted camera parameters are taken. The system updates the set of images captured at the end of each iteration with the last observation so that the whole set can be used by the trees in order to grow as deep as possible. The process stops when a certain amount of leaf nodes reached and inference can be made. The final inference about the class is made by averaging the class distribution of the leaf nodes. Grasp point detection and pose estimation are made using Hough voting from the vectors $\mathbf{g}$ and $\mathbf{p}$ of the leafs in the 3D space, combining all the viewpoints seen. Algorithm 1 summarizes the inference procedure and Figure 5 illustrates the framework.

We should note that in the experiments, this voting scheme produces a response similar to a delta function, significantly concentrated to one action. Such response is the result of the combination of many weak classifiers which vote for the most discriminating view at a time. Also, the more discriminative a view is, the more leaf nodes are reached. Therefore, inference is achieved using minimum number of actions, while if the first view is discriminative enough, no further actions may be required.

References

1. Mustafa Ozuysa, Vincent Lepetit, P.F.: Pose estimation for category specific multiview object localization. CVPR (2009)
2. Breiman, L.: Random forests. Machine Learning **45**(1) (2001) 5–32
3. Catherine Laporte, T.A.: Efficient discriminant viewpoint selection for active bayesian recognition. IJCV (2006)
4. Joachim Denzler, C.M.B.: Information theoretic sensor data selection for active object recognition and state estimation. PAMI (2002)
5. B. Schiele, J.L.C.: Transinformation for active object recognition. ICCV (1998)

Algorithm 1 ARF Inference

Input: A trained ARF, the current arbitrary viewpoint $V_{current}$
Output: garment class c, grasp point location **g** and pose **p**
function INFERENCE(ARF)
  $V_{seen} = \{V_{current}\}$ ▷ Initialze the set of seen viewpoints
  $Leafs = \emptyset$ ▷ Initialize the set of leaf nodes reached
  while *true* **do**
    Initialize $decisionVotes$ array to 0
    for all Trees T in ARF **do**
      $node \leftarrow traverse(T, V_{seen})$
      if $node = leaf$ **then**
        $Leafs \leftarrow Leafs \cup node$
        $ARF \leftarrow ARF \backslash T$
      else if $node = decisionNode$ **then**
        Increase $decisionVotes[node \rightarrow decision]$
    if Number of $Leafs > N_L$ **then**
      break
    Execute Action for Decision: $d = \text{argmax}_d(decisionVotes(d))$
    Update current view $V_{current}$
    $V_{seen} \leftarrow V_{seen} \cup V_{current}$
  return Average class c and Hough Votes $H_{\mathbf{g}(v)}$, $H_{\mathbf{p}(v)}$ from $Leafs$
end function

6. Tal Arble, F.P.F.: Viewpoint selection by navigation through entropy maps. ICCV (1999)
7. Tal Arble, F.P.F.: On the sequential accumulation of evidence. IJCV (2001)
8. Francesco G. Callari, F.P.F.: Recognizing large 3-d objects through next view planning using an uncalibrated camera. ICCV (2001)
9. Michael A. Sipe, D.C.: Feature space trajectory methods for active computer vision. PAMI (2002)
10. Eric Sommerlade, I.R.: Information-theoretic active scene exploration. CVPR (2008)
11. Sumantra Dutta Roy, Santanu Chaudhury, S.B.: Active object recognition: Looking for differences. IJCV (2001)
12. Xiang Sean Zhou, Dorin Comaniciu, A.K.: Conditional feature sensitivity: A unifying view on active recognition and feature selection. ICCV (2003)
13. Zhaoyin Jia, Yao-Jen Chang, T.C.: Active view selection for object and pose recognition. ICCV Workshops (2009)
14. Zhaoyin Jia, Yao-Jen Chang, T.C.: A general boosting-based framework for active object recognition. BMVC (2010)
15. Julia Vogel, N.d.F.: Target-directed attention: Sequential decision-making for gaze planning. ICRA (2008)
16. David Meger, Ankur Gupta, J.J.L.: Viewpoint detection models for sequential embodied object category recognition. ICRA (2010)
17. Kai Welke, Jan Issac, D.S.T.A.R.D.: Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot. ICRA (2010)
18. B. Rasolzadeh, M. Bjorkman, K.H.D.K.: An active vision system for detecting, fixating and manipulating objects in the real world. IJRR (2010)
19. Andreas Doumanoglou, Andreas Kargakos, T.K.K.S.M.: Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. ICRA 2014

20. Rosenfeld, B.: The curvature of space. In: A History of Non-Euclidean Geometry. Volume 12 of Studies in the History of Mathematics and Physical Sciences. Springer New York (1988) 280–326
21. Criminisi, A.: Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Foundations and Trends in Computer Graphics and Vision, 7(2-3):81227 (2011)
22. Pardo, L.: Statistical inference based on divergence measures. CRC Press (2005)
23. Ross Girshick, Jamie Shotton, P.K.A.C.A.F.: Efficient regression of general-activity human poses from depth images. ICCV (2011)