Active Random Forests: An Application to Autonomous Unfolding of Clothes[*]

Andreas Doumanoglou[1,2], Tae-Kyun Kim[1],
Xiaowei Zhao[1], and Sotiris Malassiotis[2]

[1] Imperial College London, London, UK
[2] Center for Research and Technology Hellas (CERTH), Thessaloniki, Greece

Abstract. We present *Active Random Forests*, a novel framework to address active vision problems. State of the art focuses on best viewing parameters selection based on single view classifiers. We propose a multi-view classifier where the decision mechanism of optimally changing viewing parameters is inherent to the classification process. This has many advantages: a) the classifier exploits the entire set of captured images and does not simply aggregate probabilistically per view hypotheses; b) actions are based on learnt disambiguating features from all views and are optimally selected using the powerful voting scheme of Random Forests and c) the classifier can take into account the costs of 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, grasp point and pose estimation of garments. We show great performance improvement compared to state of the art methods.

Keywords: Active Vision, Active Random Forests, Deformable Object Recognition, Robotic Vision.

1 Introduction

Object recognition and pose estimation has been studied extensively in the literature achieving in many cases good results [15,24]. 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 [7,13,12] in active vision share the following idea: one single-view classifier is trained to recognize the type and pose of target objects, whereas a subsequent step uses the inference probabilities to plan the next actions so that conflicting hypotheses are disambiguated. Although

[*] Electronic supplementary material - Supplementary material is available in the online version of this chapter at `http://dx.doi.org/10.1007/978-3-319-10602-1_42`. Videos can also be accessed at `http://www.springerimages.com/videos/978-3-319-10601-4`

D. Fleet et al. (Eds.): ECCV 2014, Part V, LNCS 8693, pp. 644–658, 2014.

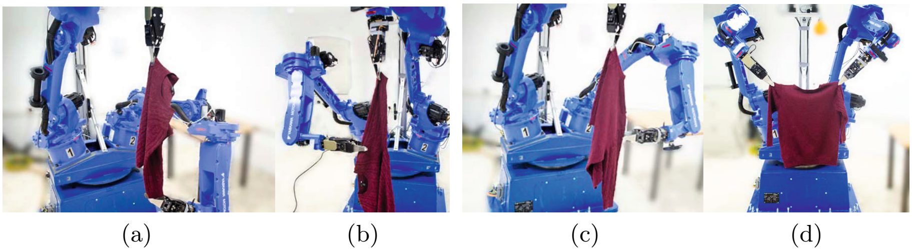


Fig. 1. Robot autonomously unfolding a shirt. a) Grasping lowest point. b) grasping 1st grasp point. c) grasping 2nd grasp point. d) final unfolded configuration

intuitive, this approach makes the combination of features from multiple views difficult whereas hypotheses from different views can only be exploited a posteriori (i.e. Bayesian formulations). In addition, their performance heavily relies on the performance of the single-view classifier. However, designing a classifier that can generalize across views is particularly challenging especially when illumination variations or deformations are considered. Another problem in active vision which hasn't been addressed by many state of the art techniques [13,12], is defining the cost associated with each action.

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 [3] 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 employ these two components in isolation [7,13,12]. Furthermore, inference is made using the entire set of captured images, 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**.

Letting the classifier decide the next disambiguating actions introduces much discriminative power to the framework, as will be shown in Section 5. We demonstrate the proposed framework in the challenging problem of recognizing and unfolding clothes autonomously using a bimanual robot, focusing on the problem of best viewpoint selection for classification, grasp point and pose estimation of garments.

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. [18] 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. [1] developed a navigation system based on entropy maps, a representation of prior knowledge about the discriminative power of each viewpoint of the objects. In a subsequent study, they presented a sequential recognition strategy using Bayesian chaining [2]. Furthermore, Callari *et al.* [4] proposed a model-based active recognition system, using Bayesian probabilities learned by a neural network and Shannon entropy to drive the system to the next best viewpoints. Also, Sipe and Casasent [19] 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 assumed independent in order to make the final inference.

One of the most representative works in the same direction was made by Denzler *et al.* [7] 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 [20] extended this idea in tracking of multiple targets on a surveillance system. One drawback of this approach was that the accumulated evidence about the visited viewpoints did not affect the viewpoint selection strategy which was based on precomputed leant actions. An improvement over this idea was made by Laporte and Arbel [13] 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 probabilistically using Bayesian update which relies on the consistent performance of the features or the single-view classifier used (in at least some viewpoints), which is generally challenging in high dimensional feature spaces like the problem of pose estimation of deformable objects. A recent work on active vision was made by Jia et al. [12] 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 the employed similarity measure is not suitable for highly deformable objects such as garments, whereas the boosting strategy based on certain priors makes a minor improvement over [7] and [13]. Finally, there are some active vision applications to robotic systems in real scenarios [22,14,23,17] mainly based on the previously described works, showing promising results.

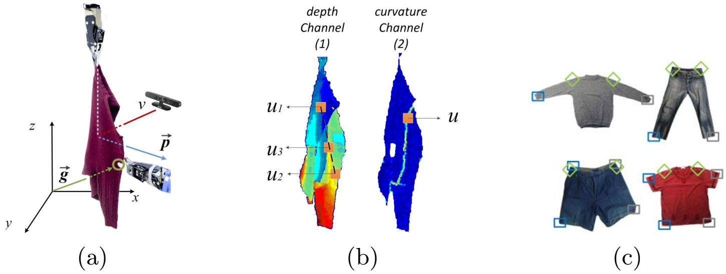


Fig. 2. Clothes Analysis. a) Grasp point g and pose vector p. b) The depth and curvature channels and the random positions used in binary pixel tests. c) Possible lowest points of clothes. Gray boxes are the symmetric points of the blue ones. Green diamonds show the desired grasping points for unfolding

Our work is based on the method proposed in [8]. In that work the 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. In that work, viewpoint selection was made sequentially by taking nearby viewpoints, which is a sub-optimal solution whilst in some cases slows down the entire process. 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 [8]. 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 garment 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 objectives in such procedure: Garment type classification, grasp points detection and pose estimation as shown in Fig. 2(a). We will describe in short these objectives, based on [8]:

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[8]. Fig 2(c) shows the 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) (diamonds). 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 Fig. 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

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 depth image (thus depicting a 3D point), $\mathbf{p}$ is a 2D vector containing the pose of the cloth defined in the XY plane as shown in Fig. 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 [8], 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 feature 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 and channel C_2 is the mean curvature of the surface[8]. Also we used the binary tests proposed in [8] containing simple pixel tests in the depth or curvature channel, which showed good results and low execution time.

While in [8] two separate forests and a POMDP were applied sequentially for classification, grasp point detection and rotation actions respectively, our new forest is able to make classification, grasp point detection and pose estimation using the same tree structure. To achieve this, we apply a hierarchical coarse to fine quality function for node splitting as in [21], so that the upper part of the trees perform classification of garments hanging from their lowest point and the lower part perform regression of grasp point or pose vectors. 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 [5] 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 (of $\mathbf{p}$ or $\mathbf{q}$), the maximum posterior probability of the samples in a node is used, with the parameter α is set to:

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

where t_c is a predefined threshold, typically set to 0.9. At a split node, the quality function in Eq. (1) is evaluated against a random set of split tests, and the one that maximizes Q is finally selected. When the maximum posterior probability $\max P(c)$ 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, in a course to fine manner.

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 depth of the trees, the current view stops being informative and the tree starts overfitting on the training samples reached the nodes. The point 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 5). The first is the *Hellinger distance*[16], a statistical measure defined over validation set S_T^j and S_D^j as:

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

when comparing the class distributions of the training set S_T^j and validation set S_D^j having C classes. $P_S(c)$ 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 squared Hellinger distance over the possible viewpoints is:

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

where

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

$\boldsymbol{\mu}_{\mathbf{q}(v)}()$ is the mean value of vectors $\mathbf{q}$ (= $\mathbf{g}(v)$ or $\mathbf{p}(v)$) in viewpoint v and A the average covariance matrix of S_T^j and S_D^j.

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:

$$JS(S_T^j \| S_D^j) = \frac{1}{C} \sum_{c=1}^{C} P_{S_T^j}(c) \log \frac{P_{S_T^j}(c)}{P_m(c)} + P_{S_D^j}(c) \log \frac{P_{S_D^j}(c)}{P_m(c)} \tag{7}$$

where P_m is the average class 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 (7) with multi-variate Gaussian distributions and compute the average over viewpoints $\mathbf{V}$, which results in:

$$JS(S_T^j \| S_D^j; \mathbf{q}) = \frac{1}{2V} \sum_{v \in \mathbf{V}} \Big(\mathbf{u}^T \left(\Lambda_{\mathbf{q}(v)}(S_T^j)^{-1} + \Lambda_{\mathbf{q}(v)}(S_D^j)^{-1} \right) \mathbf{u} + tr \left(\Lambda_{\mathbf{q}(v)}(S_T^j)^{-1} \Lambda_{\mathbf{q}(v)}(S_D^j) + \Lambda_{\mathbf{q}(v)}(S_D^j)^{-1} \Lambda_{\mathbf{q}(v)}(S_T^j) - 2\mathbf{I} \right) \Big) \tag{8}$$

where $\mathbf{u}$ is defined in Eq. (6). More details about (8) can be found in [16].

When the divergence of the training and validation set Δ ($= JS$ or HL) is above a threshold t_Δ, the node becomes an *action-selection 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 an action-selection 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 [12,19,7,13,4]:

- 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 an action-selection 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 an action-selection node in this case is depicted in Fig. 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 garment, 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 about the category or the pose of the object, 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.

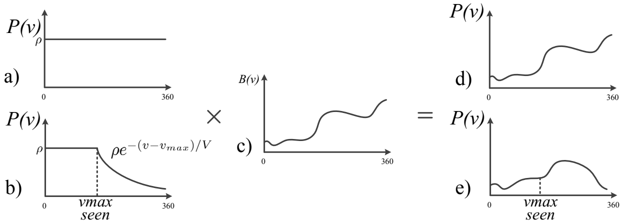


Fig. 3. Viewpoint distribution for random test selection. a) Uniform distribution, b) weighted distribution, c) Visibility map, d) Final distribution using (a), e) final distribution using (b).

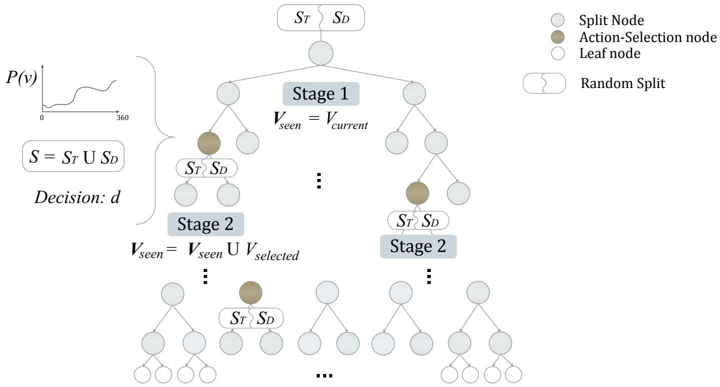


Fig. 4. Active Random Forests Training procedure

To achieve this, a visibility map B is constructed as:

$$B(v) = \frac{\sum_{s \in S^j} b(s,v)}{\sum_{v' \in \mathbf{V}} \sum_{s \in S^j} b(s,v')}, \; b(s,v) = \begin{cases} 1, & \text{if } \mathbf{g}_s(v) \text{ exists} \\ 0, & \text{if } \mathbf{g}_s(v) \text{ is not defined} \end{cases} \quad (9)$$

An example is shown in Fig. 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).

An action-selection node can now select the next best viewpoint v_{best} randomly evaluating binary tests from 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 action-selection 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. 4). That is: $\mathbf{V}' = \mathbf{V}'_{parent} \cup v_{best}$. This stage follows the same hierarchical quality

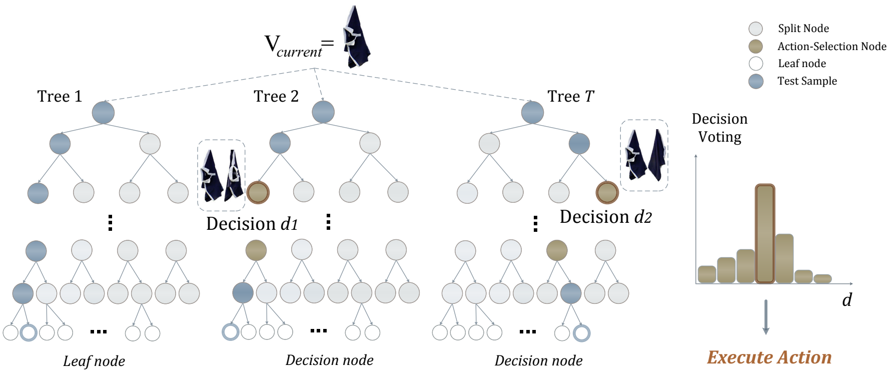


Fig. 5. Active Random Forests Inference procedure

function in Eq. (1) and the tree continues growing until another action-selection 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 [9], 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 of a garment, which is grasped and hanging from its lowest point, is captured and starts traversing the trees. Although in some trees the current view can reach a leaf node, in other trees it reaches an action-selection node where other viewing parameters are needed or another viewpoint is required (Fig. 5). Then, the action-selection nodes vote for the next best action that should be taken for collecting more information, in a similar way that leaf nodes vote for the best class of an object. Next, the most voted action is executed and another image is captured. The trees that voted for the selected action can be now traversed further by using the newly acquired image, and some of them may reach a leaf node. However, if there are not enough leaf nodes, being below a threshold N_L, this process continues iteratively until N_L leafs are reached. In each iteration, the most voted action is executed. 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 be traversed as deep as possible. 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 Fig. 5 illustrates the framework. We should mention that it is not required that all the trees should reach a leaf node, as some may have ended in an action-selection node. Parameter N_L is discussed in the experimental results, in Section 5.

Algorithm 1. ARF Inference

```
Input: A trained ARF, the current arbitrary viewpoint $V_{current}$
Output: garment class c, grasp point location $\mathbf{g}$ and pose $\mathbf{p}$
function INFERENCE($ARF$)
    $V_{seen} = \{V_{current}\}$                    ▷ Initialize 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
                $Leafs \leftarrow Leafs \cup node$
                $ARF \leftarrow ARF \setminus T$
            else if $node = action\_selection\ node$ 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 $Leafs$
```

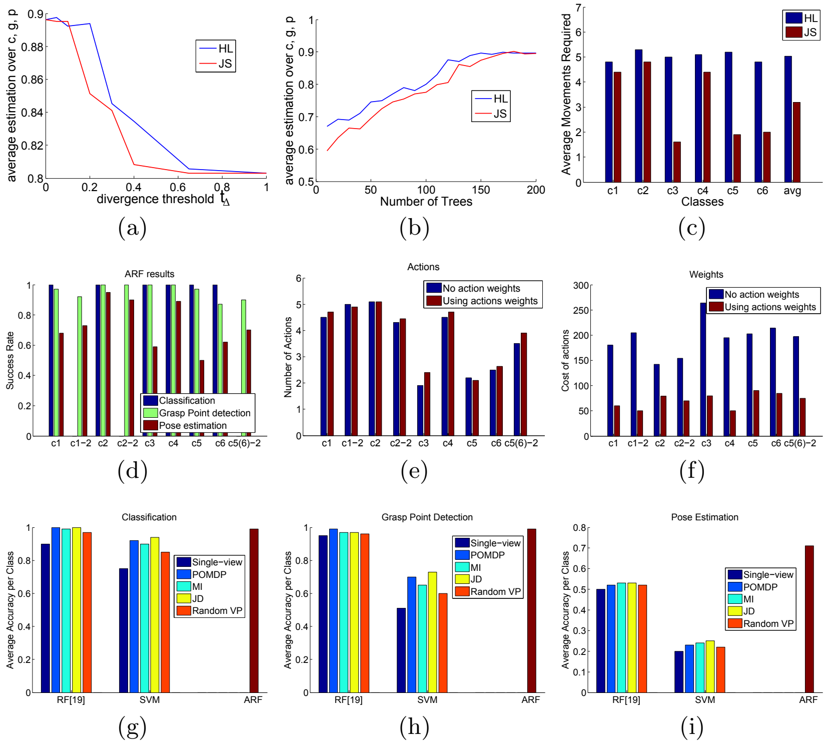
We should also note that in the experiments, this voting scheme produces a response similar to a delta function, significantly concentrated to one action. Such response is the result of the combination of many weak classifiers which vote for the most discriminating view at a time. We finally note that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

5 Experimental Results

Experimental Setup. To evaluate the ARF framework, we used our database which consists of 24 clothes, 6 of each type. Each garment was grasped by the robot gripper from each lowest point(s) 20 times to capture the random cloth configurations, collecting 40 depth images while it was rotating 360 degrees around its vertical axis. The total number of images collected is 57,600 taking into account the symmetric images as well. Another 480 unseen images for each category were used as our test samples. The training samples consist of sets of images $\mathbf{I}(v)$ containing images of a certain garment from every viewpoint v and having every arbitrary view as the first view. The steps involved in the unfolding process using the robot are: grasp the lowest point, recognize the garment and detect the 1^{st} desired grasp point and pose, grasp desired point, search for the 2^{nd} desired grasp point and pose (no classification needed), grasp final point and unfold. In the experiments bellow, classes $c_1 - c_6$ correspond to: *shirts, trousers, shorts grasped from* 1^{st} *lowest point (leg), shorts grasped from the* 2^{nd} *lowest point (waist), T-shirts grasped from the* 1^{st} *lowest point (waist), T-shirts grasped from the* 2^{nd} *lowest point (sleeve).* We train an ARF using these classes so that the robot can recognize the cloth and grasp the first desired point, based on its pose. Furthermore, we train another ARF which is used to detect the 2^{nd} desired point and pose. The second ARF does not perform classification as it is already addressed. The second ARF is trained using images from clothes hanging from their first grasp point. Thus, we define as c_i-2 the class c_i when hanging from the 1^{st} grasp point and no classification is calculated for it. Last, We have discretized the possible viewpoints into 40 equal bins of 9 degrees each, which provides enough accuracy keeping training time reasonable(few hours). We assume a correct grasp point estimation if it is at most 10cm close to ground truth, whereas 18 degrees divergence is allowed for a correct pose estimation.

Parameter Analysis. An important issue in the experiments was setting up the parameters correctly. The first parameter which needs to be defined is t_{Δ}, the threshold of the divergence of the training and validation sets of a node, above which a new decision should be made. Fig. 6(a) shows the average performance of classification, grasp point and pose estimation of an ARF containing a large number of trees (discussed below) with t_{Δ} varying from 0 to 1 for both metrics HL and JS. When t_{Δ} is 0, every node in the forest becomes an action-selection node and the forest tends to overuse the possible viewpoints available for inference increasing the total number of actions required. On the other hand, when t_{Δ} is 1, there is no action-selection node and the forest behaves as a single-view classifier. Fig. 6(a) shows that when HL is used, performance starts decreasing for $t_{\Delta} > 0.2$ while the same happens when JS is used for $t_{\Delta} > 0.1$. These are the limit values for t_{Δ}, above which the classifier tends to behave as a single-view classifier and below which it starts using redundant actions. Having t_{Δ} defined for both of our metrics, the next parameters that should be defined are the total number of trees and the minimum number of leaf nodes N_L needed by an ARF in order to make an inference. Because ARFs have a decision voting scheme along with the leaf-node aggregation, we make the following observation: Assuming that N_x leaf nodes are sufficient to make an inference and an ARF has reached $N_x - 1$ leafs, it would be desired to have another N_x trees to vote for the next decision. Therefore, N_L is set to $N_T/2$, which is half the number of trees in the forest. Fig. 6(b) shows the average accuracy of our ARF, making use of the previous observation. Both metrics reach the same level of accuracy with JS requiring more trees. However, Fig. 6(c) shows that by using JS the forest requires significantly less movements than HL to achieve the same results. Therefore, JS was used for all the subsequent experiments.

Performance and Comparisons. Fig. 6(d) shows the performance of ARF in all possible situations, with pose estimation being the most challenging objective. This figure was created without considering the weights of the actions. In the opposite case however results were very similar, thus Fig. 6(d) represents both scenarios. These two cases are compared in Fig. 6(e) which shows that weighting actions slightly increases the required viewpoints needed for inference. On the other hand, in Fig. 6(f), the required actions in the case of considering their weights have significantly lower cost than the actions in the first case, without sacrificing accuracy. The cost of an action was considered to be the degrees of rotation the gripper required in order to reach the desired viewpoint. Fig. 6(f)



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 6. Plots from experimental results showing: a) the divergence threshold t_Δ, b) Number of trees, c) average number of movements, d) ARF success rates, e) Number of movements for weighted and non-weighted actions policy f) average cost of actions of the two policies, g-h-i) Classification-Grasp Point-Pose estimation

shows the sum of the costs of all the actions needed for inference. In order to compare the ARF results, we have used two kinds of baseline methods: 1)single-view classification methods without incorporating actions; 2) active viewpoint selection methods based on a single-view method and utilizing information from entire history of selected viewpoints by updating the probability of the current state after each action. The first single-view classifier is based on Random Forests[8], modified to perform pose estimation. The second such classifier is based on multi-class and regression SVM[11,10]. The features used were the raw depth image of a garment and the HOG features[6] applied on the depth image. The first active vision technique used is based on POMDP[8], the second uses the viewpoint selection criterion proposed in [7] based on mutual information (displayed as *MI*) and the third uses Jeffrey Divergence metric as proposed in [13](displayed as *JD*). In all cases, we executed a random viewpoint selection for comparison. Finally, for a fair comparison we did not take into account the costs of actions and the visibility map (Eq. 9). Fig. 6(g) - 6(i) show the results for classification, grasp point detection and pose estimation respectively. In all cases, methods based on the SVM classifier had the worst performance. In classification

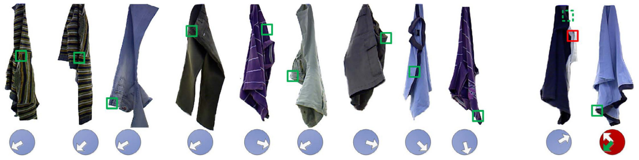


Fig. 7. Success and failure cases (the last two) of some clothes. The arrow under each cloth indicates its pose. The first error is in grasp point detection, the second in pose estimation.

and point detection, the single-view classifiers have consistent good performance and therefore the active vision approaches had a positive impact on the inference. In both cases, ARF achieves equal accuracy with the best active vision technique in each case. The power of ARFs however, is shown in Fig. 6(i), where they outperform previous works for pose estimation by almost 20%. The reason is that when dealing with such a challenging problem, the single-view inference has low accuracy producing many equally probable hypotheses. This makes classical active vision approaches perform similar to a random viewpoint selection strategy. In contrast, ARF combines features from the most discriminant views learned in training, and thus is not so affected from single-view uncertainty. Last, for achieving all the three objectives all active vision techniques were allowed to execute at most 20 actions, above which no further improvement was noticed, even when all viewpoints were seen. In contrast, as shown in Fig. 6(c), ARF shows high accuracy with an average of 3.5 moves, which is significantly lower. Fig. 7 shows some success and failure cases using some test clothes. The failures on the right are due to wrong grasp point detection and wrong pose estimation respectively. Also our supplementary video[1] shows the whole unfolding procedure using a dual arm robot, along with comparisons of ARF with the state of the art in real scenarios.

6 Conclusion

We presented Active Random Forests, a framework for addressing active vision problems, and applied it to the task of autonomously unfolding clothes. We have focused on best viewpoint selection in classification, key point detection and pose estimation of 4 types of garments. The idea of incorporating the decision process of executing disambiguating actions inside Random Forests and combining features from multiple views outperformed classical active vision techniques, especially in the challenging problem of pose estimation of clothes. Furthermore, the required number actions is significantly reduced. This framework is also open to other actions which can be integrated like zooming to a particular region or any kind of interaction with the object. This direction is left as future work.

Acknowledgment. This work was supported by the EC under the project FP7-288553 CloPeMa. A. Doumanoglou is also supported by the scholarship, and X. Zhao and T-K Kim are in part supported by EPSRC grant (EP/J012106/1).

[1] Supplementary material can be found at: `http://clopema.iti.gr/ECCV-2014/`

References

1. Arble, T., Ferrie, F.P.: Viewpoint selection by navigation through entropy maps. In: ICCV (1999)
2. Arble, T., Ferrie, F.P.: On the sequential accumulation of evidence. IJCV (2001)
3. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
4. Callari, F.G., Ferrie, F.P.: Recognizing large 3-d objects through next view planning using an uncalibrated camera. In: ICCV (2001)
5. 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), 81–227 (2011)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)
7. Denzler, J., Brown, C.M.: Information theoretic sensor data selection for active object recognition and state estimation. PAMI (2002)
8. Doumanoglou, A., Kargakos, A., Kim, T.K., Malassiotis, S.: Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In: ICRA (2014)
9. Girshick, R., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A.: Efficient regression of general-activity human poses from depth images. In: ICCV (2011)
10. Guo, G., Fu, Y., Dyer, C.R., Huang, T.S.: Head pose estimation: Classification or regression? In: ICPR (2008)
11. Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., Tibshirani, R.: The elements of statistical learning, vol. 2. Springer, Heidelberg (2009)
12. Jia, Z., Chang, Y.-J., Chen, T.: A general boosting-based framework for active object recognition. In: BMVC (2010)
13. Laporte, C., Arbel, T.: Efficient discriminant viewpoint selection for active bayesian recognition. IJCV (2006)
14. Meger, D., Gupta, A., Little, J.J.: Viewpoint detection models for sequential embodied object category recognition. In: ICRA (2010)
15. Ozuysa, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: CVPR (2009)
16. Pardo, L.: Statistical inference based on divergence measures. CRC Press (2005)
17. Rasolzadeh, B., Bjorkman, M., Huebner, K., Kragic, D.: An active vision system for detecting, fixating and manipulating objects in the real world. IJRR (2010)
18. Schiele, B., Crowley, J.L.: Transinformation for active object recognition. In: ICCV, pp. 249–254 (1998)
19. Sipe, M.A., Casasent, D.: Feature space trajectory methods for active computer vision. PAMI (2002)
20. Sommerlade, E., Reid, I.: Information-theoretic active scene exploration. In: CVPR (2008)
21. Tang, D., Yu, T., Kim, T.K.: Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In: ICCV (2013)
22. Vogel, J., de Freitas, N.: Target-directed attention: Sequential decision-making for gaze planning. In: ICRA (2008)
23. Welke, K., Issac, J., Schiebener, D., Asfour, T., Dillmann, R.: Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot. In: ICRA (2010)
24. Zhao, X., Kim, T.K., Luo, W.: Unified face analysis by iterative multi-output random forests. In: CVPR (2014)