Real-time Activity Recognition by Discerning Qualitative Relationships Between Randomly Chosen Visual Features

Ardhendu Behera
http://www.comp.leeds.ac.uk/behera/
Anthony G Cohn
http://www.comp.leeds.ac.uk/agc/
David C Hogg
http://www.comp.leeds.ac.uk/dch/

School of Computing
University of Leeds
Leeds, LS2 9JT, UK
Email: {A.Behera, A.G.Cohn, D.C.Hogg}
@leeds.ac.uk

Abstract

In this paper, we present a novel method to explore semantically meaningful visual information and identify the discriminative spatiotemporal relationships between them for real-time activity recognition. Our approach infers human activities using continuous egocentric (first-person-view) videos of object manipulations in an industrial setup. In order to achieve this goal, we propose a *random forest that unifies randomization, discriminative relationships mining and a Markov temporal structure*. Discriminative relationships mining helps us to model relations that distinguish different activities, while randomization allows us to handle the large feature space and prevents over-fitting. The Markov temporal structure provides temporally consistent decisions during testing. The proposed random forest uses a discriminative Markov decision tree, where every nonterminal node is a discriminative classifier and the Markov structure is applied at leaf nodes. The proposed approach outperforms the state-of-the-art methods on a new challenging video dataset of assembling a pump system.

1 Introduction

Automatic recognition of human *activities* (or *events*) from video is important to many potential applications of computer vision. A number of approaches have been proposed in the past to address the problem of generic activity recognition [1, 37]. Many activities can be recognized using cues such as space-time interest points [19, 20], joint shape and motion descriptors [5, 6, 12, 22], feature-level relationships [15, 18, 30, 39], object-hand interactions [4, 13, 16] and feature tracklets [25, 26]. All these approaches recognize activities by using some similarity measure [9], often based on motion and appearance throughout the interval in which it is performed. Most of these studies are based on computing local space-time gradients or space-time volume or other intensity features.These approaches are designed to classify activities after fully observing the entire sequence assuming each video contains a complete execution of a single activity. However, such features alone are often not sufficient for modeling complex activities since the same activity can produce noticeably different

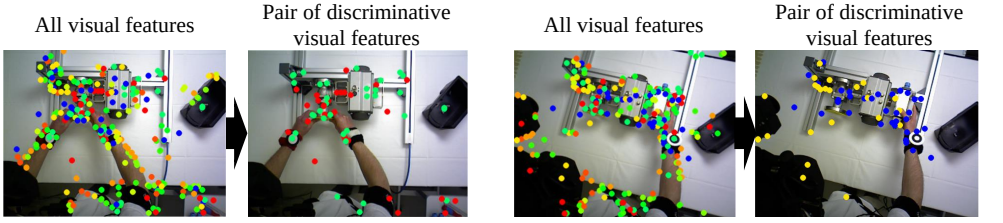


Figure 1: Proposed method recognizes (a) "Attach bearing" and (b) "Attach electric positioner" activity via learning pairs of discriminative visual features through their spatiotemporal relationships. Colored dots represent visual features (SIFT [23]).

movement patterns. For this reason, there is a growing interest in modeling spatiotemporal relationships between visual features [15, 18, 25, 30]. In this framework, we further investigate these relationships to recognize activities from a continuous live video (egocentric view) of a person performing manipulative tasks in an industrial setup. In such environments, the purpose of activity recognition is to assist users by providing instantaneous instructions from an automatic system that maintains an understanding of the on-going activities. We approach this complex problem as the composition of relatively simple spatiotemporal relationships that carry discriminative spatiotemporal statistics between visual features using a sliding window. As illustrated in Fig.1, the main idea is to learn pairs discriminative visual features based on their spatiotemporal relationships for distinguishing various activities. In order to recognize activities in real-time, we propose the use of *randomization* that considers a random subset of relational features at a time and Markov temporal structure that provides temporally smoothed output.

We propose a *random forest with a discriminative Markov decision tree* algorithm. The algorithm discovers pairs of visual features whose spatiotemporal relationships are highly discriminative and temporally consistent for activity recognition. Our algorithm is different from conventional decision trees [7, 8, 17] and uses a linear SVM as a classifier at each nonterminal node and effectively explores temporal dependency at terminal nodes of the trees. We explicitly model the spatial relationships of *left*, *right*, *top*, *bottom*, *very-near*, *near*, *far* and *very-far* as well as temporal relationships of *during*, *before* and *after* between a pair of visual features, which are selected randomly at the nonterminal nodes of a given Markov decision tree. Our hypothesis is that the proposed relationships are particularly suitable for detecting complex non-periodic manipulative tasks and can easily be applied to the existing visual descriptors such as SIFT [23], STIP [19], CUBOID [10] and SURF [3].

Like many recent works [4, 13, 36], we justify our framework using an egocentric paradigm for recognizing complex manipulative tasks in an industrial environment. Unlike these studies, our approach is targeted for intelligent assistive systems in order to assist naïve users while performing a task. Therefore, the system should be able to recognize activities from partial observations in real-time. There also have been previous approaches for recognizing activities using single frames [13, 28]. However, they are limited to either simple activities or require pre-trained object detectors. Similarly, there are approaches [11, 30, 34] that use spatiotemporal relationships which are adapted from Allen's temporal predicates [2]. These are generally unsuitable for incomplete observation in an egocentric paradigm.

We evaluate our method on an industrial manipulative task of assembling parts of a pump system and it outperforms state-of-the-art results. Our contributions are: (1) a framework for

recognizing live activities of a manipulative task in an industrial setup; (2) the novel combination of a random forest with randomization, discriminative relationships mining and Markov temporal structure; and (3) the use of qualitative relationships between pairs of visual features. The remaining parts of the paper are organized as follows: Sec.2 discusses related work. Sec.3 describes our spatiotemporal relationships and Sec.4 presents the proposed random forest. Experimental results are discussed in Sec.5 and the concluding remarks are given in Sec.6.

2 Related work

In the computer vision literature, several different approaches for activity recognition can be identified [1, 37]. There have been various attempts in the past to model spatiotemporal relationships as a context for action and activity recognition. Matikainen *et al*. proposed a method for activity recognition by encoding pairwise relationships between fragments of trajectories using sequencing code map (SCM) quantization [25]. Ryoo *et al*. presented a method for recognizing activities that uses spatiotemporal relations between spatiotemporal cuboids [30]. Sapienza *et al*. proposed a framework for action classification using local deformable spatial bag-of-features (LDS-BoF) in which local discriminative regions are split into a fixed grid of parts that are allowed to deform in both space and time [31]. Yao *et al*. classify human activity in still images by considering pairwise interactions between image regions [39]. Inspired by [30, 39], our framework considers spatiotemporal relationships between visual features.

In this work, the main objective is to recognize activities in real-time from the egocentric viewpoint which distinguishes it from the above-mentioned approaches. Starner and Pentland were one of the first to use an egocentric setup to recognize American sign language in real-time [35]. More recently, Behera *et al*. described a method for real-time monitoring of activities using a bag-of-relations extracted from wrist-object interactions [4]. Fathi *et al*. presented a hierarchical model of daily activities by exploring the consistent appearance changes of objects and hands [13]. Most of the above-mentioned approaches are designed to perform after-the-fact classification of activities after fully observing the activities. Furthermore, they often require object detectors for detecting wrists and objects as object-wrist interactions have been used as a cue for discriminating activities.

Random forests have gained popularity in computer vision applications such as classifiers [7, 21, 24, 39], fast means of clustering descriptors [27] and image segmentation [32]. Motivated by [17, 39], we combine randomization, discriminative training and a Markov temporal structure to obtain an effective classifier with good generalizability and temporally smoothed output for fast and efficient inference. Nevertheless, our method differs from [39] in that for each nonterminal node, we use a linear SVM on spatiotemporal relationships between randomly chosen visual words instead of using image regions. Moreover, our method uses a video stream for recognizing activities which is different from using single images.

3 Spatiotemporal Relationships

In this section, we explain the extraction of our proposed relationships by considering the spatiotemporal distribution of visual descriptors in a video sequence (*xyt i.e*. two image dimensions *xy* plus time *t*). These relationships are represented as a histogram in which each bin encodes the frequency of a particular relationship. A video sequence $v_i = \{I_1 \dots I_T\}$ con-

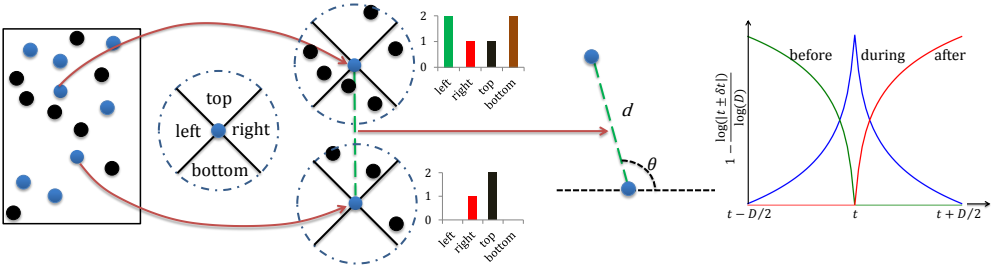
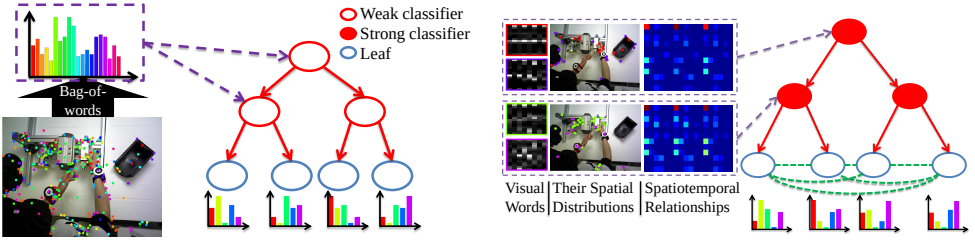


Figure 2: (a) Feature points with assigned codeword α^k (blue dots) and $\alpha^{k'}$ (black dots). (b) *Local relationships* centering each 'blue dot' (reference codeword) are created by considering surrounding 'black dots'. (c) Histogram counting qualitative *local relationships*. (d) *Global relationships* encode the relationships between a pair of visual features ('blue dot') by considering distance d and orientation θ *w.r.t.* x-axis (image plane). (e) Temporal relationships of *before*, *during* and *after* over a sliding window of duration D centered at t.

sists of T images. Every image $I_{t=1\ldots T}$ is processed to extract a set of visual features using one of the available approaches such as SIFT [23], STIP [19], SURF [3] and CUBOID [10]. Each feature $f_t = (f_t^{desc}, f_t^{loc})$ in image I_t is represented by a feature descriptor f_t^{desc} and its xy position f_t^{loc} in the image plane. A codebook of size K is generated using only the descriptor f_t^{desc} part of the features via K-means clustering. Once the codebook is generated, the descriptor part f_t^{desc} of each feature f_t is assigned the nearest codeword α^k (hard assignment) using the standard Euclidean distance *i.e.* $f_t = (\alpha_t^k, f_t^{loc})$, where $k = 1\ldots K$.

Spatiotemporal Relationships into a Histogram. The position f_t^{loc} and the assigned nearest visual word α_t^k information of feature f_t are used for the extraction of spatiotemporal relationships. For a given image I_t, a feature set $F = \{f_{t-\sigma_t:t+\sigma_t}\}$ containing all feature points over a temporal spread of σ_t is extracted. In this setting, we use $\sigma_t = 0.2$ seconds *i.e.* all frames within 0.2 seconds before and after the current frame I_t are considered. A pair of visual words α^k, $\alpha^{k'} \in$ *codebook* is randomly selected at the internal nodes of our proposed random forest (Sec.4.1). Then the respective subset of features $F_k \subset F$ and $F_{k'} \subset F$ assigned to the corresponding visual word α^k, $\alpha^{k'}$ are chosen. This is illustrated in Fig.2a, where 'blue dots' represent features from the subset F_k and 'black dots' from the respective subset $F_{k'}$. For each element in F_k, we extract the proposed *local relationships* by considering its location in the image plane. These relationships consider the elements in $F_{k'}$ which are located within a circle of radius r (experimentally we set this $1/5^{th}$ of image height) for a given element in F_k. The *local relationships* consist of *left*, *right*, *top* and *bottom* qualitative relations as depicted in Fig.2b.

The final relationships histogram describes the *global relationships* between the *local relationships* that are computed for every element in F_k. Assume there are $N = |F_k|$ element in F_k. Consequently, there are N *local relationships* structures (Fig.2c) that capture the relationships between the elements in F_k and F_k'. The *global relationships* are extracted by considering pairs of distinct elements in F_k *i.e.* pair $(f_i, f_j) \in F_k, i \prec j$ and $i, j = 1\ldots N$. For a given pair (f_i, f_j), we compute the Euclidean distance d and angle θ *w.r.t.* the image x-axis by using their respective location information, and is shown in Fig.2d. The distance $\log(d)$ is divided equally into four bins representing their respective qualitative relationships of *very-near*, *near*, *far* and *very-far*. Similarly, the direction information θ $(0-\pi)$ associated with the pair of elements f_i and f_j, is quantized into four equal orientation bins.



(a) Conventional random decision tree (b) Discriminative Markov decision tree

Figure 3: (a) Conventional decision trees. The histogram below the leaf nodes represents the posterior probability distribution $P(a|l^\tau)$. (b) The proposed decision trees sample a pair of visual words and the splitting criterion is based on the relationships between the sampled words. The posterior probability at a time step t over a video sequence is computed as $P(a_t|l_1^\tau \ldots l_T^\tau)$. Green dotted lines illustrate the temporal dependencies between leaves.

A sliding window of duration $D = 4$ seconds is used to capture the temporal relationship of the above-mentioned qualitative spatial relations. The center of the sliding window is positioned on the current frame I_t. The temporal relation for each pair of features is modeled using three basic relationships of *before*, *during* and *after* by considering time intervals within the window (Fig.2e). The amount of contribution is based on its position within the sliding window and is decided by the weight associated with the respective *before*, *during* and *after* curves in Fig.2e *i.e.* $w = 1 - \{log(|t \pm \delta t|)/log(D)\}$, where $t - D/2 \leq \delta t \leq t + D/2$. This implies if the image $I_{t+\delta t}$ is close to the reference image I_t then it gives more weight w to the bin *during* than the bin *after*. The total number of bins in our final spatiotemporal relationship histogram is 4 (*local relationships*) $\times$4 ($log(d)$) $\times$4 (θ) $\times$ 3 (*temporal*).

4 Random Forests for Modeling Activities

We begin with a brief review of the random forest framework proposed by Breiman [8]. A random forest is a multi-class classifier consisting of an ensemble of decision trees and each tree is created using some form of randomization. Every internal node contains a binary test that best splits the data in that node into two subsets for the respective left and right child node. The splitting is stopped when a leaf node is reached. Each leaf node of every tree provides the posterior probability of activity classes and is computed as a histogram representing the proportion of training examples belonging to that class (Fig.3a). An example is classified by sending it down every tree and accumulating the leaf distributions from all the trees. The posterior probability of activities a at leaf node l of tree τ is represented as $P(a|l^\tau)$, where a is the total number of activities classes in the training set. A test example is assigned an activity label a^* by taking the arg max of the averaged posterior probabilities *i.e.* $a^* = \arg\max_a \sum_{\tau=1}^{\mathcal{T}} P(a|l^\tau)$.

In the following sections, we present the process of obtaining $P(a|l^\tau)$ using the proposed approach. Further details about the learning procedure for the conventional random forest can be found in [7, 8, 21, 32].

4.1 Discriminative Markov Decision Trees

In order to recognize activities from video sequences, we propose a random forest consisting of *discriminative Markov decision trees* which unifies three important concepts: (1) *Ran-*

Algorithm 1: Pseudocode for growing decision trees in the proposed random forest framework.

1: **for** *tree* $\tau = 1 \rightarrow \mathcal{T}$ **do**
2: Randomly select a subset of training samples $S' \subset S$; (Sec.4.2)
3: **if** `SplitNode` (S') **then**
4: Randomly assign a binary label;(Sec.4.2)
5: Randomly sample the candidate pairs of visual words; (Sec.4.2)
6: Compute the relationships histogram **h** (Sec.3);
7: Select the best pair of visual words to split S' into two subsets S'_l and S'_r (Sec.4.2);
8: `SplitNode` (S'_l) and `SplitNode` (S'_r).
9: **else**
10: Return the posterior probability $P(a|l^\tau)$ for the current leaf.
11: **end if**
12: **end for**

Proposed Random Forest (SIFT+STIP)	**68.56%**
Proposed Random Forest (SIFT)	66.54%
Proposed Random Forest (STIP)	56.34%
Wrist-object relationships (Behera *et al.* [8])	52.09%
Conventional Random Forest (SIFT+STIP)	57.11%
Conventional Random Forest (STIP)	49.39%
Conventional Random Forest (SIFT)	53.28%
χ^2-SVM (SIFT+STIP)	63.19%
χ^2-SVM (STIP)	54.19%
χ^2-SVM (SIFT)	53.21%

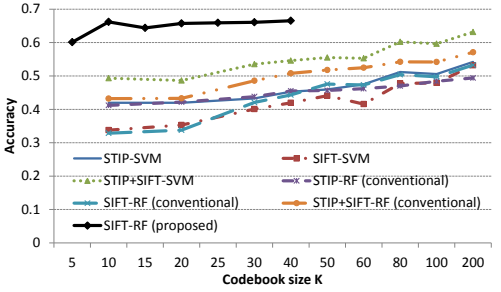
Table 1: Performance comparison for the leave-one-subject-out experiments on our new challenging dataset. In χ^2-SVM, SIFT+STIP is the concatenation of the both bag-of-words features.

domization to explore the codebook space; (2) *Discriminative training* to extract the most important spatiotemporal relationships between visual words; and (3) Exploring *temporal consistency* between *leaf nodes* for encoding sequential information (Fig.3b). In our discriminative classifier, we use feature vectors describing the spatiotemporal qualitative relationships between randomly selected pairs of visual words. The sampling space is $K \times (K+1)/2$ (including self pairing *i.e.* (α^k, α^k)) for a given codebook size of K.

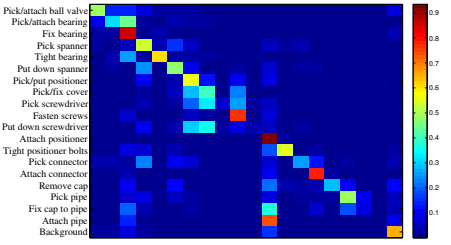
4.2 Growing Trees with Randomized Learning

An overview of generating the proposed random forest is shown in Algorithm 1. Each tree is trained separately on a random subset $S' \subset S$ of the training data S (step 2 in Algorithm 1). $S = \{I\}$ is a set consisting frames belonging to training sequences. Learning proceeds recursively, binary splitting the training data at internal nodes into the respective left and right subsets S'_l and S'_r (step 3). The binary splitting is done in the following four stages: randomly assign all frames from each activity class to a binary label (step 4); randomly sample a pair of visual words from the codebook sampling space (step 5); compute the spatiotemporal relationships histogram **h** using the sampled visual words as described in Sec.3 (step 6); and use a linear SVM to learn a binary split of the training data using the extracted **h** as feature vector. At a given internal node, assume there are $a' \subseteq a$ activity classes. We uniformly sample $|a'|$ binary variables and assign all frames of a particular activity class to a binary label. Using the extracted relationship histogram **h**, we learn a binary SVM at each internal node and send the data sample to the left child if $\mathbf{w}^T\mathbf{h} \leq 0$ otherwise to the right child, where **w** is the set of weights learned through the linear SVM. Using the information gain criteria in [7], each binary split corresponds to a pair of visual words and is evaluated on the training frames that falls in the current node. Finally, the split that maximizes the information gain is selected (step 7). The splitting process is repeated with the newly formed subsets *i.e.* S'_l and S'_r (step 8). The current node is considered as a leaf node (*i.e.* there is no further splitting) if it encounters any of the following conditions: (i) predefined maximum depth has been reached, (ii) the total number of training samples is less than a predefined threshold and (iii) the information gain is insignificant (step 3).

Implementation Details. Each tree is trained using a random subset consisting 80% of the training frames. At each nonterminal nodes, we use the default setting of $\sqrt{K \times (K+1)/2}$



(a) Comparison with varying codebook size



(b) Confusion matrix ($K = 10$)

Figure 4: (a) Performance comparison with various baselines with increasing codebook size. The performance of the proposed method using $K = 5$ (60.13%) is better than most of the baselines with $K = 200$. (b) Confusion matrix of the proposed method using SIFT ($K = 10$).

pairs of visual words in selecting the node split tests. We restrict the depth of our tree to 10 and the minimum number frames in a nonterminal node to 2% of the total training frames. In all our experiments, we generate 200 trees per forest.

4.3 Inference

The proposed inference algorithm computes the posterior marginals of all activities a_t over a frame I_t at t. Assume there are T frames in a given video sequence ($t = 1 \dots T$). For a given tree τ and frame sequence $I_1 \dots I_T$, the respective sequence of visited leaf nodes is $l_1^\tau \dots l_T^\tau$ (Fig.3b). Using this sequence of leaf nodes, our goal is to compute the posterior distribution $P(a_t|l_1^\tau \dots l_T^\tau)$ of activities over the frame I_t. The smoothed output over the whole forest is achieved by averaging the posterior probabilities from all $\mathcal{T}$ trees:

$$a_t^* = \arg\max_{a_t} \sum_{\tau=1}^{\mathcal{T}} P(a_t|l_1^\tau \dots l_T^\tau) \tag{1}$$

From now onwards, we discuss the computation of the posterior probabilities from a single tree and therefore, for clarity we will not use the tree term τ. The right side of the above equation (1) can be expressed as: $P(a_t|l_1 \dots l_T) = P(a_t|l_1 \dots l_t, l_{t+1} \dots l_T)$ *i.e.* the probability distribution is expressed by breaking at the time point t. By applying Bayes rule and conditional independence of the leaf sequence $l_1 \dots l_t$ and $l_{t+1} \dots l_T$ given activities a_t:

$$P(a_t|l_1 \dots l_T) \propto P(a_t|l_1 \dots l_t) P(l_{t+1} \dots l_T|a_t) \tag{2}$$

The term $\mathbf{f_{0:t}} = P(a_t|l_1 \dots l_t)$ provides the probability of ending up in any particular activity after visiting the first t leaf nodes and is essentially the "forward message pass". Similarly, $\mathbf{b_{t+1:T}} = P(l_{t+1} \dots l_T|a_t)$ provides the probability of visiting the remaining leaf nodes given the starting point $P(a_t|l_1 \dots l_t)$ and is known as the "backward message pass". The respective forward $\mathbf{f_{0:t}} = \mathbf{f_{0:t-1}} \mathcal{A} P(l_t|a_t)$ and backward $\mathbf{b_{t+1:T}} = \mathcal{A} P(l_{t+2}|a_{t+2}) \mathbf{b_{t+2:T}}$ probabilities are computed using the forward-backward algorithm [29]. $\mathcal{A}$ is the activities transition probability matrix and is computed using the activity labels of all frames belonging to the training sequences. The probability $P(l_t|a_t)$ of reaching a leaf node l_t given activity a_t is estimated by applying Bayes rule *i.e.* $P(l_t|a_t) = P(a_t|l_t)P(l_t)/P(a_t)$, where $P(a_t|l_t)$ is the posterior activities distributions (histogram) in the leaf node l_t of our decision tree (Fig.3b).

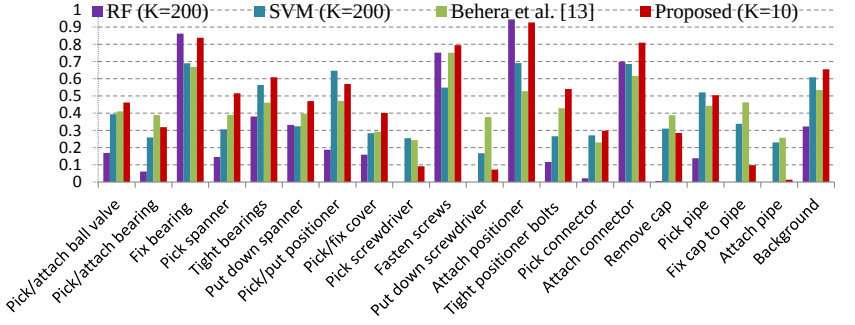


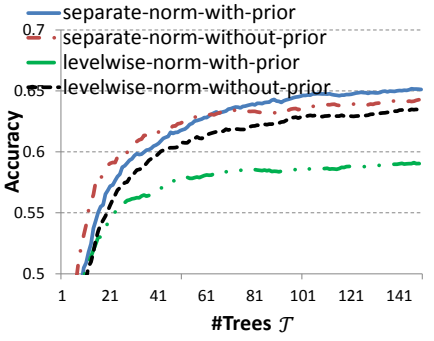
Figure 5: Comparison of the performance of live activity recognition. SIFT bag-of-words ($K = 200$) results in accuracy of 53.21% using χ^2-SVM and 53.28% using conventional random forest. The method in [4] results in 52.09%. The proposed method is 66.20% ($K = 10$) significantly better than the baselines, where the random chance is 5%.

5 Experiments

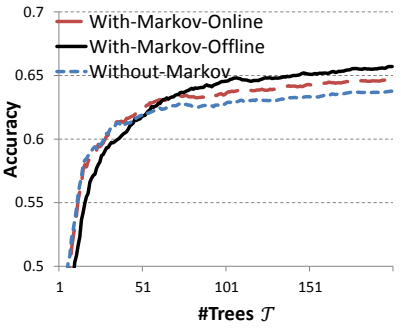
In order to validate our novel activity recognition framework, we use a challenging industrial task (Ball Valve Assembly) in which a person installs a new ball valve and assembles the components of a pump system. The task includes complex bimanual manipulations involving many tools, and shiny and textureless small objects. The task is classified with background and 19 different activities, namely: (1) Pick/attach ball valve, (2) pick/attach bearing, (3) fix bearing, (4) pick spanner, (5) tight bearing, (6) put down spanner, (7) pick/put positioner, (8) pick/fix cover, (9) pick screwdriver, (10) fasten screws, (11) put down screwdriver, (12) attach positioner, (13) tighten positioner bolts, (14) pick connector, (15) attach connector, (16) remove cap, (17) pick pipe, (18) fix cap to pipe and (19) attach pipe. Some activities appear multiple times within the task. For example, 'pick up spanner' and 'put down spanner' activities are called each time a part is attached to the pump system. The dataset consists of 30 video sequences captured from 6 participants executing the task (30 fps, $\sim 210,000$ frames)[1]. Training and testing sets are based on leave-one-subject-out as is done in [4, 13].

Baselines: We use two different classification techniques: SVM and random forest [8] using a histogram representing bag-of-words for each sliding window. For SVM, we use the χ^2-kernel for better accuracy as reported in [38]. We train a χ^2-SVM by generating a bag-of-words built over the sliding window on STIP [19]. Similarly, we train another χ^2-SVM on SIFT [23]. We further concatenate the STIP and SIFT and train a third χ^2-SVM for performance comparison. Similarly, we train two different random forests: one using STIP and another using SIFT. We linearly combine the output of these two forests to get the joint performance. Experimentally, we found that by combining the output of the forests performs 2.5% better than using a random forest on concatenating STIP and SIFT bag-of-words histogram. The results are shown in Table 1. In χ^2-SVM, using both STIP and SIFT perform better than individual (STIP: 54.19%, SIFT: 53.21%) and we got the similar trend using conventional random forest (STIP: 49.39%, SIFT: 53.28%). In most of the classification techniques, the performance using only SIFT is better than STIP. This could be due to the uncontrolled movements of the camera in an egocentric setup.

[1] Dataset and source code are available at www.engineering.leeds.ac.uk/computing/research/ai/BallValve/index.htm



(a) Normalization



(c) Markov effect

Figure 6: (a) Effect of various normalization with prior (distribution of codewords). (b) Performance comparison of our proposed random forests with and without temporal structure (Fig.3b). For $K = 20$, without temporal structure the accuracy is 63.76% and with the structure the accuracy is 64.74% and 65.71% online and offline, respectively.

Proposed Method: We compare our method to the baselines and the state-of-the-art work in [4] which models the wrist-object and object-object interactions using qualitative and functional relationships. The work in [4] uses a generic object detector for the key objects. The detection and tracking is done using RGB-D video. That method achieves 52.09% accuracy on our dataset consisting of 20 activity classes. The proposed method achieves 66.54% ($K = 40$) using SIFT only where random chance is 5%. This is a significant improvement over the existing approaches presented in the Table 1. One of the main reason for the better performance of our proposed method is that the state-of-the-art method relies on the quality of the object detections. Our dataset consists of manipulative tasks and often the key objects are partially occluded. Furthermore, the activities consist of metallic textureless objects which makes it difficult for the object detector. The proposed method overcomes these problems by using spatiotemporal relationships at the feature level and captures both the wrist-object appearance and motion information. Furthermore, the proposed approach achieved accuracy of 66.54% using only SIFT features. By combining both SIFT and STIP, we get a significant boost in recognition accuracy (68.56%) in comparison to the baseline evaluations. For live activity recognition we use only SIFT features since STIP is computationally more expensive.

We compare the performance of our method with the baselines with increasing codebook size K and is shown in Fig.4a. For $K = 5$, the proposed method performs better than the most of the baselines for $K = 200$. This is mainly due to the way we encode the spatiotemporal qualitative relationships between randomly chosen pairs visual words. The performance of the proposed method increases with K. However, the training time for the proposed forest increases with K as the number of unique pairs $K \times (K+1)/2$ grows. Nevertheless, the testing time is the same for all K as the splitting criterion is based on the relationships between a single pair which is obtained generating the tree (Fig.3b). The average computational time for computing the proposed relationships is around 5 milliseconds on a single core 2.8 GHz, 64-bit PC. Therefore, the proposed method can easily be applicable for the monitoring of live activities as the performance is significantly better than the baselines as well as state-of-the-art [4] for a smaller value of K. Fig.4b shows the confusion matrix of the proposed method for live activity recognition. We present the performance comparison with the baselines for live activity recognition using SIFT features in Fig.5.

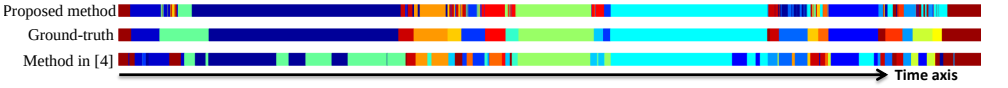


Figure 7: Task recognition result of one of the live test videos. The horizontal axis represents time. Each color represents an activity assigned to the sliding window. The middle bar represents the ground-truth, the bottom bar shows the results using method [4] (52.09%) and the top bar represents the result of our proposed method (66.54%). The output of the proposed method is smoother and matches better with the ground-truth in comparison to [4].

Normalization Strategies: We look into the effect of 4 different normalization strategies on our qualitative relationships histogram and their influence on the overall performance (Fig.6a). Experimentally, we found that the performance is slightly better using the L_2-norm separately on the *before*, *during* and *after* relationships in our histogram than a single normalization. The level-wise normalization is used in each level while generating our relationship histogram. First, the *local relationships* histogram (*left*, *right*, *top* and *bottom*) is normalized using the L_1-norm. Then, the *global relationships* histogram (*very-near*, *near*, *far* and *very-far*) is normalized again using the L_1-norm (Sec.3). Finally, we use the above-mentioned separate L_2-norm on the *before*, *during* and *after* relationships. We found that using a separate L_2-norm gives better performance than the level-wise normalization (Fig.6a).

In bag-of-words approaches [14, 20, 33], specific visual words will normally be significantly biased towards certain activities classes. Therefore, a classifier learnt on the spatiotemporal relationships between a pair of visual words will have corresponding prior preferences for those activities classes. In order to include this prior, we assign the relationships between each pair of visual words ($\alpha^k, \alpha^{k'} \in codebook$) with a weight $w_{k,k'} = h_k + h_{k'}$, where h is a histogram with K bins representing the distribution of K visual words in a sliding window (bag-of-words distribution). The performance is better by using this prior with the separate L_2-norm and is shown in Fig.6a.

The influence of adding temporal links (Fig.3b) in our discriminative decision tree in the proposed forest is shown in Fig.6b for $K = 20$. Offline and online refer to the respective evaluation using complete observation (full activity sequence) and partial observation *i.e.* from beginning to the current time step t. As expected in modeling sequential data, the performance is improved $\approx 2\%$ using these temporal links.

6 Conclusions and future work

We present a random forest with discriminative Markov decision tree algorithm to recognize activities. The proposed algorithm finds a pair of visual features whose spatiotemporal relationships are highly discriminative and uses a Markov temporal structure that provides temporally consistent decisions. The proposed method can be easily applicable for live monitoring of activities and does not require the intermediate step of object detection. The proposed framework is evaluated on a new dataset comprising industrial manipulative tasks and outperforms the result of state-of-the-art methods. Future work is to include functional relationships between visual features.

Acknowledgements: This work was partially funded by the EU FP7-ICT-248290 (ICT Cognitive Systems and Robotics) grant COGNITO (www.ict-cognito.org), FP7-ICT-287752 grant RACE (www.project-race.eu) and FP7-ICT-600623 grant STRANDS (www.strands-project.eu).

References

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):1–16, 2011.

[2] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11): 832–843, 1983.

[3] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[4] A. Behera, D. C. Hogg, and A. G. Cohn. Egocentric activity monitoring and recovery. In *ACCV*, 2012.

[5] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[6] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Trans. PAMI*, 23(3):257–267, 2001.

[7] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *ICCV*, 2007.

[8] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[9] F. Cuzzolin and M. Sapienza. Learning pullback hmm distances. *IEEE Trans. on PAMI*, 36(7):1483–1489, July 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2013.181.

[10] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[11] K. S. R. Dubba, A. G. Cohn, and D. C. Hogg. Event model learning from complex videos using ilp. In *ECAI*, pages 93–98, 2010.

[12] A. A. Efros, A. C. Berg, E. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003.

[13] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *ICCV*, 2011.

[14] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[15] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV*, pages 925–931, 2009.

[16] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, 2007.

[17] M. I. Jordan, Z. Ghahramani, and L. K. Saul. Hidden Markov decision trees. In *NIPS*, 1996.

[18] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *IEEE CVPR*, 2010.

[19] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[20] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[21] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *CVPR*, 2005.

[22] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009.

[23] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2): 91–110, 2004.

[24] R. Marée, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *CVPR*, 2005.

[25] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *ECCV*, 2010.

[26] R. Messing, C. Pal, and H. Kautz. Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, 2009.

[27] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, 2007.

[28] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

[29] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989.

[30] M. S. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, 2009.

[31] Michael Sapienza, Fabio Cuzzolin, and Philip H.S. Torr. Learning discriminative space-time action parts from weakly labelled videos. *Int. Journal of Computer Vision*, pages 1–18, 2013.

[32] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[33] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[34] M. Sridhar, A. G. Cohn, and D. C. Hogg. Unsupervised learning of event classes from video. In *AAAI*, 2010.

[35] T. Starner and A. Pentland. Real-time American sign language recognition from video using hidden Markov models. In *Proc. of Int'l Symposium on Computer Vision*, 1995.

[36] E. H. S. Taralova, F. Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *IEEE Workshop on Egocentric Vision, in conjunction with CVPR*, 2009.

[37] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Trans. Circuits Syst. Video Techn.*, 18(11):1473–1488, 2008.

[38] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *CVPR*, 2010.

[39] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011.