IET Computer Vision

Research Article

IET Journals
The Institution of Engineering and Technology

Histograms of sequences: a novel representation for human interaction recognition

ISSN 1751-9632
Received on 21st September 2017
Revised 14th March 2018
Accepted on 9th April 2018
doi: 10.1049/iet-cvi.2017.0471
www.ietdl.org

Aytac Cavent[1], Nazli Ikizler-Cinbis[1] ✉

[1]*Department of Computer Engineering, Hacettepe University, 06800 Ankara, Turkey*
✉ *E-mail: nazli@cs.hacettepe.edu.tr*

Abstract: This study presents a novel representation based on hierarchical histogram of local feature sequences for human interaction recognition. The authors' method basically combines the power of discriminative sequence mining and histogram representation for the effective recognition of human interactions. Our framework involves extracting visual features from the videos first, and then mining sequences of the visual features that occur consequently in space and time. After the mining step, we represent each video with a histogram pyramid of such sequences. We also propose to use soft clustering in the visual word construction step, such that more information-rich histograms can be obtained. The authors' experimental results on challenging human interaction recognition data sets indicate that the proposed algorithm performs on par with the state-of-the-art methods.

1 Introduction

Owing to its various application domains such as surveillance, human–computer interaction, sport and entertainment analysis, and more, human action and activity recognition is a constantly evolving topic of interest in computer vision research community. Most of the research up to date has focused on the analysis of singleton human actions, where only one person is performing a single action. However, real-world situations are more complicated. Complex activities are taking place in videos; an important portion of these activities are human interactions, where more than one person is involved in a particular activity.

In this paper, we aim to look at the problem of recognising human–human interactions, where the videos are taken in uncontrolled settings, and more than one actor is involved in the interaction. In this problem, there are many challenging issues to be dealt with, such as occlusions, cluttered backgrounds, scale variations, changes in light, variations in appearance of the people, moving camera, and more. These issues require many different adaptation techniques and a carefully designed framework.

In a simple and relatively good performing activity recognition system, using temporal and appearance features together is a requirement, because the actions are mostly characterised in both temporal and spatial dimensions. Many good-performing feature representations such as cuboids of histogram of oriented gradient (HOG) and histogram of optical flow (HOF) [1], space-time interest points (STIPs) [2] model the temporal structure of local actions over one to three frames at most. This short-term structure may have limited representative power in recognition of complex actions and interactions that are sustained on long time periods. A scalable temporal model that can be used to model the long duration of interactions is needed. In temporal domain, most of the existing work focus on segmentation for finding representative common subsequences over the different types of actions [3, 4]. Such approaches can cause information loss due to the presence of splits, even if the method finds best partitions among the possible ones. Therefore, we believe that a more general framework that does not propagate the error introduced in simple action recognition phase is required. This framework should handle the whole sequence more effectively for the purpose of interaction recognition.

The idea mainly explored in this paper is to effectively model the temporal co-occurrence of the local features. Co-occurrence of local features indicates important cues for accurate recognition of the interactions. Current successful representations [5] based on bag-of-words (BOW) of local features mostly ignore the relative positions; however, the relative temporal locations of these local features are important, especially if one wants to differentiate between visually similar interactions.

In this work, we propose a new feature representation that encodes the videos as a histogram of *local feature sequences*. The sequences correspond to the time-ordered list of the visual vocabulary elements where the dictionaries are constructed over the space–time interest points. Among these sequences, the frequently occurring ones are found by means of a sequence mining approach. We further apply sequence selection for better discrimination between classes. As an extension to base histogram of sequences (HoS) model, we incorporate a temporal pyramid mechanisation, inspired by the work of Choi *et al.* [6]. Here, the height of the temporal pyramid determines the temporal scales of the sequences. This new representation is likely to cover more complex spatio-temporal relationships by means of mining sequences of local interest points that frequently occur. In this way, the temporal variability of local sequences can be more accurately modelled. The proposed sequence mining and construction of this HoS representation is illustrated in Fig. 1.

We evaluate our method on the challenging human interaction data sets UT Interactions [7] and TV Human Interactions [8]. Our results indicate that the proposed representation is quite successful in discovering the discriminative patterns for the recognition of human interactions, achieving the state-of-the-art performance.

2 Related work

There are many works in the literature of computer vision that aims at recognising short-scale actions in videos. The recent surveys [9–11] include broad overviews on this topic. The first line of work is based on the local spatio-temporal features leveraged with a discriminative classifier. The second line of work uses temporal dynamic models, such as hidden Markov model, Bayesian model, and finite state models. The third line of work uses motion-based analysis, which employs motion clusters and analyses the temporal ordering of these clusters and also in the recent years, deep learning methods are proposed with the availability of the high processing power of the GPUs. The following sections provide detailed explanation of the methods.

2.2 Local interest point methods

The BOW with STIPs is employed by a number of approaches to represent the human action. This representation is combined with discriminative classifiers [2, 12, 13]. Marín-Jiménez *et al.* [14]

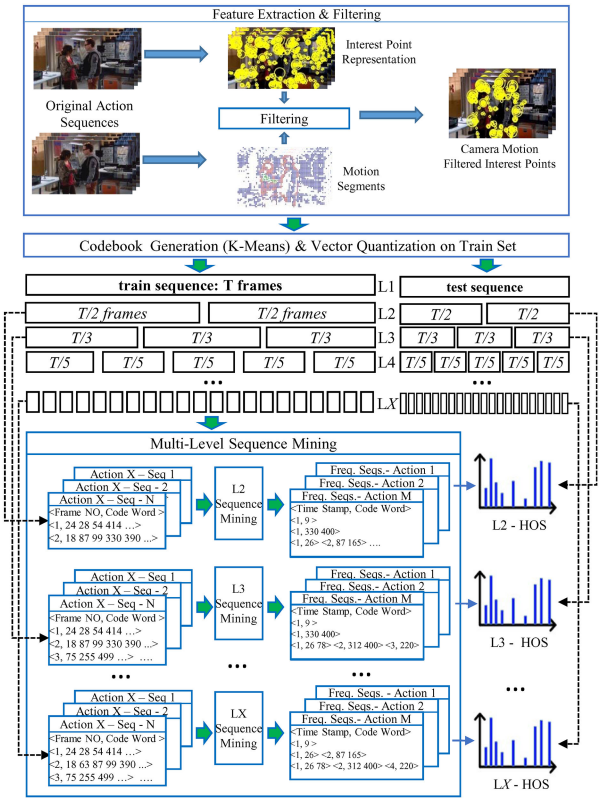

Fig. 1 *Overall framework of the proposed method*

reported the performance of the bag of STIPs on human interaction data sets. According to their results, the combination of BOW and STIP is still a competitive approach in human interaction recognition problems and the improvement of the STIP representation is an important issue for the future work. They also found out that when modelling interactions between persons, the context does not help in the recognition process and, therefore, it is needed to reject STIP densely sampled outside the person region. Despite the computational efficiency of the bag of STIPs method, the model discards spatial and temporal structure. Patron-Perez *et al.* [15] build a person-centred descriptor and use an upper body detector to find the people in every frame. The found detections are clustered to form tracks. Then, they calculate descriptors along the tracks and use support vector machine (SVM) classifier for each interaction class by using these descriptor. Hoai and Zisserman [16] also proposed a method that finds the tracks of the upper bodies of the people and for each upper body track, a track-focused descriptor is then computed based on dense trajectory descriptors [17] which encode gradient and motion cues along the trajectories. They also compute an HOG-based scene descriptor, which is the average of HOG descriptors computed on key frames. Thus, a video is represented by a human-focused descriptor and an HOG-based scene descriptor. Then, they use SVM classifier for each interaction class by using both scene and motion along the upper body.

Zhang *et al.* [18] proposed a more successful method, which captures co-occurrence information. Spatial and temporal distance between two local features are used as the core of the information. Ryoo and Aggarwal [19] introduced a new method to compare the structure of videos by using the spatio-temporal relationships between the features with the temporal and spatial predicates (before, meets, overlaps, far, near, and more). Their approach detects and localises all occurring activities. Slimani *et al.* [20] proposed a method that extracts a descriptor that captures the co-occurrence of the local interest points found in three-dimensional (3D) XYT spatio-temporal volume for each interacting person. Then the co-occurrence descriptor is used in standard discriminative classifiers.

Savarese *et al.* [21] used the co-occurrences within local spatio-temporal regions to encode the local motion and appearance and also they used the probabilistic latent semantic analysis to learn the action classes. In our method, we can handle more complex relations in temporal domain, the length of a sequential pattern determines the temporal complexity, and the number of items within an itemset determines the spatial complexity of the sequential pattern. There is no limitation on the length of a sequential pattern. The local features, especially STIP, are highly affected by the camera movements since they have no adaptation for external movements. Khokher *et al.* [22] proposed an improved STIP detector by extracting salient interest points by taking into account long-term temporal interactions and camera motion. They encode the local features in the form of tensors in order to retain the spatio-temporal structure. To reduce the size of the tensor, they selected the features by Fisher ranking.

2.2 Models focused on temporal dynamics

For the recognition of complex activities, there are relatively fewer works. Niebles *et al.* [3] improved the temporal pyramid matching idea and propose a dynamic temporal scale selection method rather than using fixed length temporal parts. To select the best temporal representation, Niebles *et al.* [3] generate a set of random temporal scales and train classifiers for each temporal partitions, where the best temporal scales are selected according to the recognition performance. As opposed to finding best partitions, our work operates on finding the most frequently occurring feature sequences at different temporal scales. Wang *et al.* [17] proposed the use of dense trajectories to model the motion information. The dense trajectories also have some level of robustness to camera motion since the trajectories are based on optical flow of points. Zhang *et al.* [23] follow the similar method with the Wang *et al.* [17] so that they cluster trajectories using the coherent filtering and employ a multiple instance learning (C-KNN). Kong and Fu [24] proposed the max-margin action prediction machine for recognising actions in incomplete videos. They formulate the action prediction task as a structured SVM learning problem by using the dense trajectories and interest points, and incorporate composite kernels to capture non-linear classification boundaries in the prediction task. Their method requires bounding boxes or a person detector.

Gaidon *et al.* [25] represented the videos as a tree of clustered point trajectories. The authors clustered the trajectories into representative motion parts by building a cluster tree and they embedded the cluster tree into their descriptors. Burghouts and Schutte [26] modelled the spatio-temporal layout of 48 actions by using location of STIP features. They generated six different layouts based on the combination of three different coordinate systems, use of feature posterior probabilities (which are generated from learned probability density functions of the feature locations), and use of the standard histogram representation with BOW.

Li *et al.* [27] proposed a method to exploit the temporal structure by comparing the principal angles between subspaces representing the activity types. They generated Hankel matrices by use of trajectories to model the temporal information. Yu *et al.* [28] employed a Hough-transform-based voting method to recognise human actions. Since the performance of the hough voting is highly dependent on the quality and the amount of the input data, the authors suggested to use propagative Hough voting by using random projection trees (RPT). With their method, feature voting is done by using RPT rather than using votes of local features individually.

Vahdat *et al.* [29] proposed a model that represents actions as a sequence of poses. The method requires complete tracks of actors across the entire sequence and the occlusions may cause problems. Since our model uses local features, occlusions have minimal effect on the performance. Ma *et al.* [30] proposed a method that discovers a compact set of hierarchical space–time tree structures of human actions from training videos by using the hierarchical space–time segments [31]. Using an ensemble of the discovered trees, or in combination with simpler action words and pairwise structures, they build action classifiers that achieve state-of-the-art performance on two challenging data sets: High Five [8] and UCF-Sports [32]. Our method mines the local interest points instead of the high level space–time segments and we use discrete sequence

Fig. 2 *Example frames for the STIP filtering step on the TV Human Interaction data set [8]. Original frames are shown to the left of the figure and the filtered STIPs as the result of background segment identification is shown on the right. In order to identify the background, motion segments discovered by Brox and Malik [42] are used and the segment with maximum spatial deviation is taken to be the background segment. The STIPs that lie on the detected background segment is filtered out, and only foreground STIPs that lie on the foreground segments remain*

mining methods instead of tree mining and our method can capture lower level features by using the local features.

Recently, Liang *et al.* [33] proposed hierarchical feature representation structure for affective interaction recognition based on the local feature and mid-level feature descriptions. They extract local features and contours from the spatio-temporal segments of the video and represented as trajectories. Then, the local features are clustered to form a dictionary. The final video descriptor is represented as concatenation of the spatio-temporal histograms of the video segments using the calculated dictionary. Our method is different from the method of Liang *et al.* [33] in the descriptor design. We exploit the pairwise spatio-temporal relations between the local features via data mining methods instead of trajectory generation.

Key frames are also commonly used for action recognition. Liu *et al.* [34] used Adaboost to select the key frames. Raptis and Sigal [35] proposed to use a more compact key frame subset (up to 4 frames or %4 of frames). Sefidgar *et al.* [36] proposed a key component model that selects a set of key components, discriminative moments in a video sequence that are important evidence for the presence of a particular interaction. They use object trackers and find out the interaction between the objects.

2.3 Deep learning methods

With the advancement of parallel processing power (GPUs, CPU clusters), the convolutional neural networks (CNNs) started to be successfully employed for the action recognition. Tran *et al.* [37] proposed a spatio-temporal feature learning method using deep 3D CNNs. Karpathy *et al.* [38] studied the performance of CNNs in large-scale video classification, using 1 million videos with 487 categories (Sports-1M data set) obtained from YouTube videos. There are different methods [39–41] proposed to model the temporal dynamics of the actions. Simonyan and Zisserman [41] capture separate spatial and temporal recognition streams based on CNNs. Feichtenhofer *et al.* [39] proposed an improvement to [41] that is able to fuse spatial and temporal cues at several levels of granularity in feature abstraction. Li *et al.* [40] presented a multi-granular deep architecture, which is able to incorporate information at a multitude of granularity including frame, consecutive frames (motion), clip, and the entire video. They employed long short-term memory networks to incorporate long-term temporal modelling based on the granularity features. In this work, we also experiment with CNN-based features within our mining framework.

3 Proposed method

Our proposed framework involves (i) interest point extraction, (ii) filtering interest points, (iii) visual vocabulary construction, (iv) temporal pyramid construction, (v) sequence mining, and (vi) classification steps. Below, we describe each of these steps in detail.

3.1 Interest point detection and feature extraction

In computer vision community, local feature points have been proven to have good performance on matching and recognition. The attractiveness of local features comes from the fact that they are invariant to scale and occlusions, due to their locality nature. For this reason, as the low-level representation, we choose to use space–time interest points (STIPS) [2]. Extraction of STIPs involves a space–time extension of the Harris detector and is characterised by a high variation of the image values in space and non-constant motion over time. The detected points have significant variation in space–time neighbourhood. For each video point, the spatio-temporal second-moment matrix is computed using independent temporal and spatial scale values. In this work, we use the provided detector of [2] to detect the local STIPs.

3.2 Filtering STIPs

Since the STIP extraction procedure looks for high variations in both spatial and temporal domain, it is not robust to camera movements. When the camera is moving, the points that have high variation in the spatial domain are likely to have variation in the temporal domain as well. Especially, in the videos with complex backgrounds which are likely to have many corner-like structures, the number of detected feature points increases enormously.

To filter the interest points that belong to the background, we make use of a motion segmentation algorithm which analyses long-term point trajectories and applies spectral clustering [42] to find the motion segments within a video. Once the motion segments are obtained, we assume that the segment that has the maximum spatial deviation is the background segment and assume that the remaining segments are the foreground segments. By identifying the background segment in this way, we can filter out the interest points that fall into background segment. Example results of the filtering step are shown in Fig. 2. While not perfect, a great portion of the STIPs that fall onto the background region are eliminated this way.

After extracting and filtering local STIPs, local feature descriptors are extracted around each STIP. In order to capture both the shape and the motion information around the STIPs, we choose to extract the motion boundary histogram (MBH) features. The MBH descriptors [43] are derivatives of the optical flow and similar to HOF descriptor [1] and are shown to be effective descriptors for action recognition [5, 17]. We use MBH descriptors which have some level of robustness to camera motion, as the positive effect of using derivatives of the optical flow. We compute MBH descriptors at the location of STIPs. The temporal and spatial scale values of the STIPs are also used in MBH computation to form the encompassing grid of interest. We used the implementation of the STIP [1] and MBH [5] provided by the authors.

3.3 Visual vocabulary construction and sequence formation with soft clustering

After filtering the STIPs as described, we construct the visual dictionary for quantisation our high-dimensional descriptors. For this purpose, we adopt the classical vector quantisation procedure that uses k-means to cluster extracted descriptors into k words, where cluster centres become the visual words of the final visual vocabulary.

It has been shown that hard-assignment-based quantisation suffers significantly from the loss of information. When each data element is assigned to exactly one cluster, it is possible to lose the similarity information between the data point and the other clusters. Several techniques such as sparse coding [44] and locality-constraint linear coding (LLC) [45] have been proposed to deal with this situation. In our case, since we need a discrete representation to apply the subsequent sequence mining procedure, we employ soft clustering in order to compensate for such an information loss. In this formulation, rather than mapping a visual feature to a cluster centre, we map it to the closest K clusters. We select the closest clusters having a distance less than a threshold. The threshold $\tau = c \times \sigma$ is defined as a multiple of the standard deviation σ of all distances to all cluster centres. At the end, each frame is vector-quantised via this soft-clustering procedure and represented with the set of visual words that appear on that frame.

3.4 Temporal pyramid construction

When we look at the variations in the speed and the duration of actions, a single short-term temporal model may not cover the temporal structure of the actions. Some actions are sustained on the longer time period, whereas some actions are performed within a short time. The local features do not explain long-term actions well and the sequence mining procedure is not robust to small changes in the speed of execution of the actions. To solve this problem, we propose a temporal pyramid construction.

Temporal pyramid construction is done as follows: the videos are divided into fixed size subsequences at the temporal dimension. At each level of the temporal pyramid, we divide the video into the T parts where T is the corresponding Fibonacci number for that level such as 2, 3, 5, 8, 13, 21, and so on. The features in the same temporal window are assumed to occur at the same time interval and therefore assigned the same temporal label in the mining process. The depth of the temporal pyramid defines the temporal subsequence size. This process is illustrated in Fig. 1. At the higher levels of the pyramid (like *L2* in Fig. 1), the mining algorithm finds the sequential patterns that are common in the long term and at the lower levels of the pyramid (like *LX* in Fig. 1), the mining algorithm finds the common short-term sequential patterns.

3.5 Mining frequently occurring sequences

The sequential pattern mining paradigm was first introduced by Agrawal and Srikant [46] and it is a subject of data mining concerned with finding statistically relevant patterns between data examples where the values are represented in a sequential form and have associated temporal labels. The problem is to find the most co-occurring closed sequences when the data set contains time-ordered set of features. Sequential pattern mining has been successfully used in many applications that involve temporal data points. For interaction recognition, we propose to mine the frequently occurring sequential interest point patterns, and then represent this information by means of a histogram encoding.

3.5.1 Sequential pattern mining: Formally, the sequential pattern mining is defined as follows [46]: let $I = \{i_1, i_2, \ldots, i_n\}$ be the universal set of items, where i_x is an item. X is defined as an itemset, if $X \subseteq I$. A sequence $s_m = \langle X_1, X_2, \ldots, X_m \rangle$ is defined as an ordered list of itemsets. A sequence $s_1 = \langle A_1, A_2, \ldots, A_n \rangle$ is said to be contained in another sequence, $s_2 = \langle B_1, B_2, \ldots, B_m \rangle$, if there exists integers $1 \leq i_1 < i_2 < \ldots < i_n \leq m$ such that $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \ldots, A_n \subseteq B_{i_n}$, where A and B are itemsets.

Given a sequence database $\mathrm{SDB} = \{s_1, s_2, \ldots, s_k\}$, which is a set of sequences, the problem of sequential pattern mining is to find the sequential patterns that have support values greater or equal to user-specified minimum *support*. The *support* for sequential pattern candidate $\mathrm{support}_{\mathrm{SDB}}(s)$ is the fraction of total sequences that include that particular sequential pattern candidate.

The original problem of sequential pattern mining considers only the order of item occurrences but not the intervals between two items. The time interval between two items has effect on the meaning of the sequence and should be taken into account. Hirate and Yamana [47] proposed a method called generalised sequential pattern mining with time intervals (GSPM) that employs time labels for each itemsets and brings additional constraints to address time interval requirements in sequence mining. The sequences that contain time labelled itemsets are called as interval extended sequence (is) and the sequential database that is a set of interval extended sequences are called as interval extended sequence database (ISDB). Four types of additional constraints are proposed in *support* calculation. Constraint C1 is the minimum time interval required between two adjacent itemsets. Constraint C2 is the maximum time interval required between two adjacent itemsets. Constraint C3 is the minimum time interval required between head and tail of a sequence. Constraint C4 is the maximum time interval required between head and tail of a sequence. For example, consider the interval extended sequence is $= \langle (0;1)(1;1,2)(3;1,2,3) \rangle$. The time interval between head and tail of the sequence 'is' is 3 and the time interval between the second and the third itemsets is 2.

To find the frequent interval extended sequences, the algorithm GSPM [47] makes recursive projections of the interval extended sequence database with prefixes. The shortest prefix is a single item i with time label 1. Formally, let us assume $\alpha = \langle (t_{1,1}, A_1), (t_{1,2}, A_2), \ldots, (t_{1,m}, A_m) \rangle$ and $\beta = \langle (t'_{1,1}, B_1), (t'_{1,2}, B_2), \ldots, (t'_{1,m}, B_m) \rangle$ be interval extended sequences, where $t_{x,y}$ is the difference of the occurrence time of the A_x and A_y. The α is said to be contained in β if there exists integers such that $1 \leq i_1 < i_2 < \ldots < i_n \leq m$ such that $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \ldots, A_n \subseteq B_{i_n}$ and $t_{1,i} = t'_{1,i_1}, t_{1,2} = t'_{1,i_2}, \ldots, t_{1,n} = t'_{1,i_n}$. The prefix of interval extended sequence α based on the (t_β, A_β) is defined as follows:

$$\mathrm{prefix}(\alpha, A_\beta, t_{1,\beta}) = \langle (t_{1,1}, A_1), (t_{1,2}, A_2), \cdots, (t_{1,j}, A_j) \rangle \tag{1}$$

Postfix of interval extended sequence α with regard to (t_β, A_β) is defined as follows:

$$\mathrm{postfix}(\alpha, A_\beta, t_{1,\beta}) = \langle (t_{j,j}, A'_j), (t_{j,j+1}, A_{j+2}), \ldots, (t_{j,m}, A_m) \rangle \tag{2}$$

where A'_j contains the items in A_j and does not include the items in A_β. When there exists no integer j, postfix of α with regard to (t_β, A_β) becomes the following:

$$\begin{aligned} \mathrm{prefix}(\alpha, A_\beta, t_{1,\beta}) &= \varnothing \\ \mathrm{postfix}(\alpha, A_\beta, t_{1,\beta}) &= \varnothing \end{aligned} \tag{3}$$

The patterns are calculated by finding the frequent items first. For each frequent item a, where $\mathrm{support}_{\mathrm{ISDB}}(1, a) > min\_sup$, the initial interval extended sequence α is created as $< (0, a) >$ where $a \in A_i$. Then, ISDB$|\alpha$ is created. The ISDB$|\alpha$ is the projection of the ISDB with regard to α and defined as:

$$\mathrm{ISDB}|\alpha = \{\mathrm{is} | \mathrm{is} \neq \varnothing \wedge \mathrm{is} = \mathrm{postfix}(\gamma, \alpha, 0)\} \tag{4}$$

where $\gamma \in$ ISDB.

Similarly, in the next level of projection, all possible pairs $(t_{1,i}, a)$ in ISDB$|\alpha$ are generated. For each pair satisfying the minimum support constraint, β is defined as $\mathrm{prefix}(\alpha, a, t_{1,i})$ and the

projected interval extended sequence database ISDB$|\beta$ becomes a collection of postfixes of sequences in ISDB with regard to β

$$\begin{aligned}\text{ISDB}|\beta = \{\text{is}|\text{is} \neq \varnothing \wedge \text{is} = \text{postfix}(\gamma, a, t_{1,i}) \\ \wedge \text{support}_{\text{ISDB}|\alpha}(t_{1,i}, \beta) \geq \text{min\_sup} \\ \wedge \beta \text{ satisfies } C_1, C_2, C_3, \text{ and } C_4\}\end{aligned} \quad (5)$$

where $\gamma \in$ ISDB. For more details, refer to [47].

When we execute the sequential pattern mining algorithm GSPM [47], we observe that there are many redundant sequential patterns that are included in other sequential patterns that have same support value. As an example, the sequential patterns: $sp_1 = \langle(0;1,2,3)\rangle$ and $sp_2 = \langle(0;1,3)\rangle$ have support value of %75 and sp_2 is contained in sp_1, so there is no additional information gained by use of sp_2. Elimination of these sequences are done by use of *closed sequence mining* methods. A closed sequential pattern is a frequent sequential pattern that is not included in another sequential pattern having exactly the same support. For our case, we adopt the sequence mining algorithm of Hirate and Yamana [47] with the bi-directional extension [48] (BIDE). BIDE checks if a sequential pattern is closed or not, with no need to maintain the set of historical closed patterns.

To give more insight on the problem, an example set of sequences is shown in Table 1. In this example, there are four input sequences with three, four, two, and two itemsets, respectively. Each itemset consists of one or more elements and labelled with a tag to account for the time of occurrence. Table 2 shows some sequential patterns extracted by applying sequence mining to the set of example sequences provided in Table 1.

Table 1 Example input sequences for the sequence mining algorithm

ID	Sequences
Seq1	$\langle(0;1)(1;1,2,3)(2;1,3)\rangle$
Seq2	$\langle(0;1)(1;1,2)(2;1,2,3)(3;1,2,3)\rangle$
Seq3	$\langle(0;1\ 2)(1;1,2)\rangle$
Seq4	$\langle(0;2)(1;1,2,3)\rangle$

Table 2 Extracted sequential patterns from example input of Table 1 via applying sequence mining.

ID	Sequential patterns	Support, %
P2	$\langle(0;1,2)\rangle$	100
P5	$\langle(0;2)(1;1)\rangle$	100
P1	$\langle(0;1,2,3)\rangle$	75
P6	$\langle(0;1)(1;1,2)\rangle$	75
P3	$\langle(0;2)(1;1,2)\rangle$	75
P4	$\langle(0;2)(1;1,3)\rangle$	75
P7	$\langle(0;1,2)(1;1)\rangle$	75

In our case, the aim is to identify the frequently appearing sequential patterns for human interactions in videos. To this end, we need to run the mining algorithm for each class. In this action recognition domain, the ISDB corresponds to set of class-specific video sequences V_i such that ISDB $= \{V_1, V_2, \ldots, V_m\}$. A video sequence is an ordered set of frames and correspond to the interval extended sequence such that $V = \langle(t_{1,1}, F_1), (t_{1,2}, F_2), \cdots, (t_{1,n}, F_n)\rangle$, where F_i is a video frame and $t_{1,i}$ is the frame number. A frame F_i corresponds to an itemset A, where each detected visual word correspond to items of the itemsets.

Fair representation of each class in the final model can be guaranteed by the equal number of participants from each class. For finding equal number of candidate sequences from each class, we use the min-support threshold. In our experiments, the min-support values are automatically selected so that ≈ 1000 sequential patterns are mined for each class.

Fig. 3 shows extracted sequential patterns pictorially. Each pattern may have different temporal length. There may be sequential patterns at length 1 and they represent co-occurring features in the same frame. Also, same pattern may exist in different temporal locations in a video (e.g. $p2$ in Fig. 3).

3.6 Histogram computation

The histograms are constructed using the occurrence of the sequential patterns within the video sequences. The occurrence of a pattern in a sequence is defined in Section 3.5.1. The pseudo-code of the algorithm for computing these occurrences is given in Algorithm 1 (see Fig. 4). The *pttrn* in Algorithm 1 corresponds to a sequential pattern. The size of histogram is same as the number of extracted patterns. The complexity of this algorithm is proportional to the number of the input video sequences, length of the input video sequences, and the length of the patterns $O(k \times m \times n)$, where k is the number of the sequences, m the length of the sequence, and n the length of the pattern.

3.6.1 Sequential pattern selection: For the application of sequence or frequent itemset mining algorithms, there is a well-known trade-off: if the minimum support value min_sup is kept large, co-occurring sets of items can be very limited and may not cover all the interesting subpatterns. On the contrary, if the min_sup is kept low, the number of found sequential patterns can be prohibitively large and it becomes infeasible to process them all.

In order to have a more compact and discriminative set of sequences, we employ a selection procedure. The main idea of this selection is to discard the sequential patterns which are not discriminative enough. With this purpose, we first generate a maximum set of sequential patterns by running the mining algorithm for each class and collecting all patterns together. Then, for each class, we train a linear SVM on the calculated histograms using the generated patterns. The sequential patterns that acquire large weights in these linear SVMs are selected as the final set of

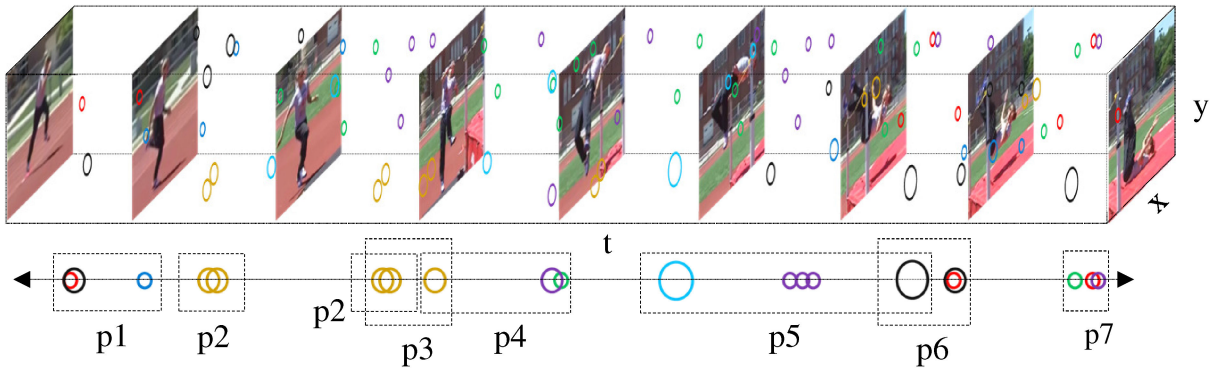

Fig. 3 *Pictorial representation of the discovered sequential patterns (when the temporal pyramid is not used). The circles in the figure represent the local interest points and the colour identifies different type of interest points in terms of visual words. The size of the circles represents the spatio-temporal scale of the features. Each pattern is identified with a number (pX) and the features in a discovered sequential pattern are grouped within a box. The length of the box determines the length of the sequential pattern. There may be overlapping features within a frame set (e.g. p2, p3, and p4)*

Input: $seq \neq \emptyset$, $pttrn \neq \emptyset$, $pttrn\_idx \geq 0$, $seq\_idx \geq 0$, $num\_occur \geq 0$
Output: $num\_occur \geq 0$ {number of occurrences}
1: **for** i in $seq[seq\_idx] : seq[last\_idx]$ **do** {loop on itemsets}
2: **if** $t(i) = t(pttrn[pttrn\_idx])$ **then**
3: $pttrn\_item \leftarrow firstItem(pttrn[pttrn\_idx])$
4: **for** j in i **do** {loop on items in itemset i}
5: **if** $pttrn\_item = j$ **then**
6: **if** $pttrn\_item = lastItem(pttrn)$ **then**
7: $num\_occur \leftarrow num\_occur + 1$
8: **else if** $pttrn\_item = lastItem(pttrn[pttrn\_idx])$ **then**
9: $num\_occur \leftarrow num\_occur + countPatternOccurrence(seq, pttrn, pttrn\_idx + 1, seq\_idx + 1, num\_occur)$.
10: **else**
11: $pttrn\_item \leftarrow nextItem(pttrn[pttrn\_idx])$
12: **end if**
13: **end if**
14: **end for**
15: **end if**
16: **end for**
17: **return** num_occur

Fig. 4 *Algorithm 1: countPatternOccurrence*

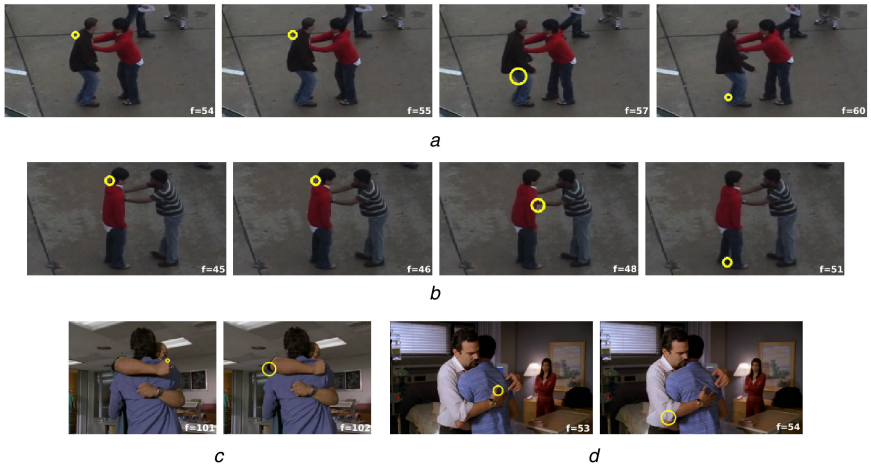

Fig. 5 *Example patterns that are discovered using our approach from the UT Interactions data set (a and b) and TV Human Interactions data set (c and d). The local feature points are displayed as yellow circles and the size of the circle represents the spatial scale of that local feature*
(a) Pushing-1, ***(b)*** Pushing-2, ***(c)*** Hugging-1, ***(d)*** Hugging-2

sequential patterns. Fig. 5 shows some example sequences found after using the sequence mining and sequence selection steps.

3.6.2 Formation of HoS representation: After selection step, we have a set of discriminative patterns for each class and the videos are represented as an HoS where the attributes are the final set of selected sequential patterns and the values are the normalised number of occurrences of those sequential patterns. More specifically, let $P = \{p_1, p_2, \ldots, p_k\}$ be a set of sequential patterns which includes all patterns selected for each class, α a sequence, and k is the number of sequential patterns. The histogram representation is:

$$H_\alpha = [f(\alpha, p_1, 0, 0, 0), f(\alpha, p_2, 0, 0, 0), \ldots, f(\alpha, p_k, 0, 0, 0)] \quad (6)$$

where f function counts the number of pattern p_i occurrences in α as defined in Algorithm 1.

When the temporal pyramid structure is used, we concatenate the histograms from each class at each temporal pyramid level into a single histogram and same classification method is applied. If we assume L as the number of levels in the temporal pyramid, there will be $2^L - 1$ temporal segments. The video sequence a is temporally split into 2^{i-1} parts at level i such that $\alpha^i = \langle (1, A_1^i), (2, A_2^i), \ldots, (2^{i-1}, A_{2^{i-1}}^i) \rangle$ where $0 < i \leq L$ and A is the itemset within that split. Also, the itemsets at level i will be concatenation of lower level itemsets: $A_j^i = \left[A_{2*j-1}^{i+1} A_{2*j}^{i+1}\right]$ where $0 < j \leq 2^{L-1}$. Then, the histogram at level i is:

$$H_\alpha^i = \left[f(\alpha^i, p_1^i, 0, 0, 0), f(\alpha^i, p_2^i 0, 0, 0), \ldots, f(\alpha^i, p_k^i 0, 0, 0)\right] \quad (7)$$

After the histograms at each level are concatenated, the final histogram pyramid for sequence α becomes concatenation of the histograms at all levels:

$$\mathrm{HoS}_\alpha = \left[H_\alpha^1 H_\alpha^2 \cdots H_\alpha^L\right] \quad (8)$$

This becomes the representation for each sequence and video sequences are classified using this compact representation.

3.7 Classification

For this purpose, we use SVM classifiers with radial basis function (RBF) kernel in a one-versus-all manner. We find the best cost and gamma values for the RBF kernel by making cross-validation on the train data set.

4 Experiments

4.1 Data sets

We have tested our method on two benchmark data sets of human interactions. These are UT Interactions data set [7] and TV Human Interactions data set [8]. Both of these data sets are collected from sources of real-world videos. These data sets are particularly suitable for evaluating the recognition of more complex activities that involve more than one person or more than a singleton action.

TV Human Interactions data set: This data set contains a total of 300 videos collected from TV programmes [8]. There are 200 videos containing 4 different classes of daily interactions, namely hugging, high five, kissing, and hand shaking, and 100 videos that do not belong to any class, hence labelled as negative. While using this data set, we follow the training and testing split

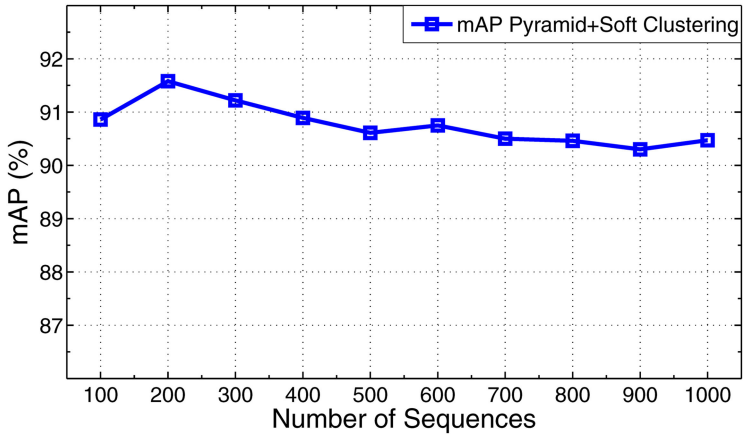

Fig. 6 *Effect of number of candidate sequential patterns for each class at every level of the pyramid for TV Human Interactions data set*

provided by Patron *et al*. [8]. The data set is split into two sets, each set containing 25 video clips for each interaction and 50 video clips for negative samples. In this data set, the human body bounding box annotations are provided, but our method does not require any of such annotations, so we do not use any annotations, other than the class labels in training.

UT Interactions: The UT Interactions data set [7] contains videos of continuous executions of six classes of human–human interactions: shake-hands, point, hug, push, kick, and punch. For the classification task, the data set provides 120 video segments divided into 2 sets. Set#1 and set#2 are both composed of ten video sequences taken on a parking lot, where each sequence contains at least one execution per action class. The videos of set#1 are taken with slightly different zoom rate, and their backgrounds are mostly static with little camera jitter. The videos in set#2 are taken on a lawn in a windy day. Background is moving slightly and the videos contain more camera jitters. For performance comparison, 10-fold cross-validation method is used as suggested by Ryoo and Aggarwal [7] for each set. Again, we do not make use of any annotations.

4.2 Implementation details

After the frequent sequence extraction, since there can be duplicate sequences across the different classes, the duplicate sequences are removed first. The total number of unique sequential patterns for both data sets becomes <3000. The length of the longest sequential pattern at the highest level is set as 6 for the TV Human Interactions data set and as 16 for UT Interactions data set. The length of the shortest sequential pattern at the lowest level is 1 for both of the data sets. We observe that the found patterns in TV Human Interactions data set are shorter than those of UT Interactions data set.

We set the distance threshold for the soft clustering as the standard deviation of the all distances to the all cluster centres. The sequential pattern mining parameters are set as follows: $C1$(minimum interval between two itemsets) is 0 to cover co-occurrence information as much as possible, $C2$ (maximum interval between two itemsets) is 5 which is found by experimental analysis on TV Human Interactions data set, $C3$ (minimum whole interval) is 0, and $C4$ is ∞ to cover all sequential patterns at all lengths.

The UT Interactions data set has small number of videos and the 10-fold cross-validation is used for the final performance on the test set. We have performed a 3-fold cross-validation on the train set to select the number of level in the temporal pyramid and the number of patterns. Since the train set is small, %100 accuracy is obtained for most of the runs. For this reason, we omit parameter selection in this data set and use the same set of parameters with TV Human interactions data set.

In addition to working with local STIP features, we also experiment with CNN features on the TV Human Interactions data set as an input to the HoS method. For this purpose, we use the pre-trained VGG-16-based spatial network (split#1 of the UCF-101 data set) of Feichtenhofer *et al.* [39] and we fine-tune this network on TV Human Interactions data set. We extract FC6 and Relu5-5 outputs as the input features to sequence mining and evaluate the results. When working with CNN features, we apply the same pipeline except that there is no interest point filtering applied.

4.3 Experiments on TV human interactions data set

4.3.1 Performance of hierarchical HoS: There are two important parameters in forming the HoS descriptor: (i) the number of class-specific mined patterns for each pyramid level, (ii) the number of selected class-specific patterns for each level of the pyramid after the sequence selection step. We do cross-validation on the training data sets to select these two parameters. We extract same number of patterns for each level of pyramid in the experiments. Fig. 6 shows the performance (mAP) of number of the generated patterns for every level of the pyramid on the training data sets as described in Section 3.5.1 before the sequence selection step. Our intent was to first find the best number of initial candidate sequential patterns and then apply sequential pattern selection. The best performance is found when the number of generated patterns is 200 which means that the initial descriptor length is $200 \times 5 = 1000$ (where 5 is the number of classes) for each level.

The effect of the number of the class-specific selected patterns is shown in Fig. 7. The number of the class-specific patterns is determined at the sequential pattern selection step as defined in Section 3.6.1. The best result is obtained when 150 class-specific patterns are selected from each level of the temporal pyramid.

Selecting the number of pyramid levels is another important factor of our HoS representation. For this purpose, we perform 5-fold cross-validation on training set to select the number of pyramid levels. We construct a temporal pyramid up to depth 6. Fig. 8 shows the contribution of cumulative pyramid levels. We observe that the recognition performance increases up to level 6, due to the additional information acquired from lower granularity of data in each subsequent temporal level. However, we observe that, after level 6, the recognition performance drops. This is mostly due to the large variations in the short-term activities and the noise caused by short-term motions present in the videos.

4.3.2 Performance of single-level HoS: Fig. 9 shows the effect of the number generated candidate sequential patterns (as described in Section 3.5.1 before the sequence selection step) on the performance (mAP), when the temporal pyramid is not used, i.e. single-level HoS. According to Fig. 9, the best results are obtained when the histogram size is 1500. Fig. 10 shows the effect of the number of the class-specific selected patterns on the performance after the sequence selection step on the training data set. The number of the class-specific patterns is determined at the sequential patterns selection step as defined in Section 3.6.1.

4.3.3 Comparison with existing work: We compare our method with the several methods as shown in Tables 3 and 4. In the method

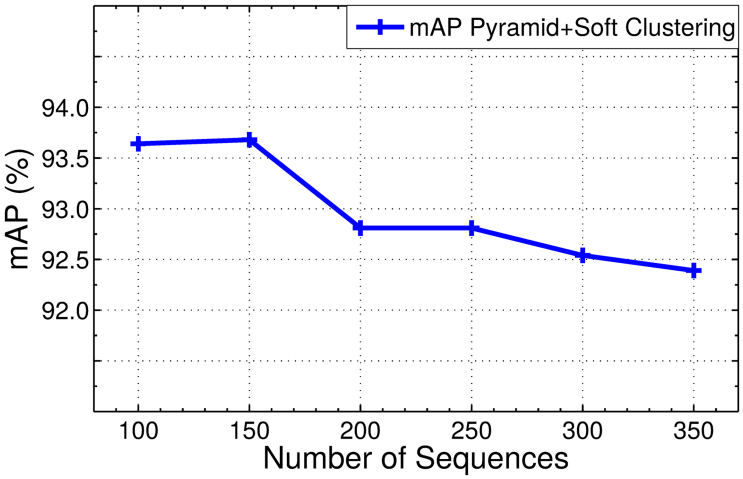

Fig. 7 *TV Human Interactions data set, effect of number of selected sequential patterns for each class at each level of the pyramid after the sequential pattern selection step*

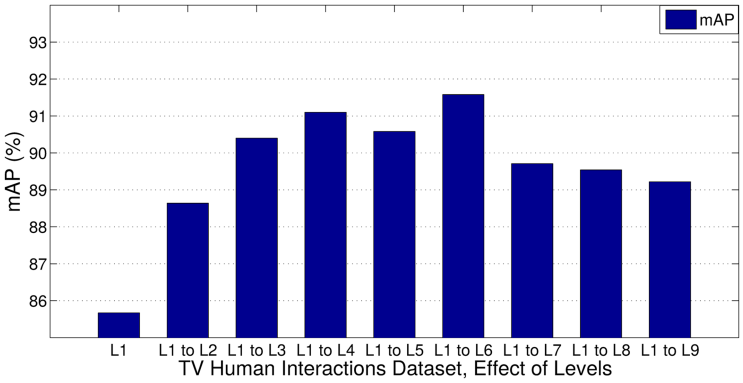

Fig. 8 *Effect of the number of pyramid levels on the recognition performance on TV Human Interactions data set. L1 to LX represent that the HoS representation has been formed using level 1 to level X*

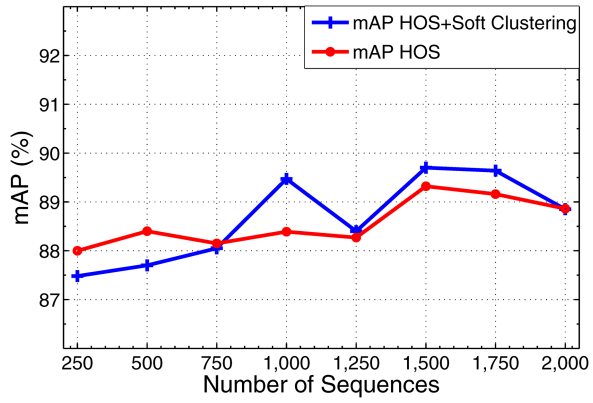

Fig. 9 *Effect of number of candidate sequential patterns when the temporal pyramid is not used on TV Human Interactions data set*

of Patron-Perez *et al.* [15], head orientation features are included in the final descriptors, to enrich their representation. Marín-Jiménez *et al.* [14] apply standard BOW + STIP method and discard the STIPs outside the person region in training phase by using person bounding boxes provided within the database. We report the best results of Patron-Perez *et al.* [15] when including the negative videos as part of the retrieval task, which have settings using manual or automatic annotations and structured learning (SL). Hoai and Zisserman [16] use upper body annotations to train the upper body detector. Similar to our experimental setting, Yu *et al.* [28] and Gaidon *et al.* [25] did not use any annotations. As can be seen, in the presence of negative videos, our method outperforms other methods that do not use any annotations. When the negative videos are included, our method is % + 1.4 better than Yu *et al.* [28]. Note that, the method of Li *et al.* [27] has the best results in this data set, but their method requires manual annotation. When the negative videos are not included (Table 4) in the evaluation, our method is % + 3.0 better than Yu *et al.* [28] and almost same as Khokher *et al.* [22]. Khokher *et al.*'s method [22] uses a short window video stabilisation step to gain robustness against the camera motion. In our framework, we do not apply such a video stabilisation and therefore, we can say that the features are affected by the camera motion. We believe that our framework could also make use of such an addition of a video stabilisation step.

The effects of the proposed soft clustering and the temporal pyramid extension are also analysed in Tables 3 and 4. The soft clustering has nearly %6 positive effect on the average recognition performance. The performance gain on mAP when using the hierarchical temporal pyramid is nearly +%13 (when the negative videos are included) and +%12 (when the negative videos are excluded) in the TV Human Interaction data set. We observe that the best performance is achieved by the proposed HoS formulation using soft clustering and hierarchical temporal pyramid with local interest point features.

We also experiment with deep learning-based features in our framework and the results are presented in Tables 3 and 4. We observe that the recognition performance is lower, compared to using local interest points (%3.7 below when the negative videos are included and %9.4 below when the negative videos are excluded). This is mostly due to the non-local nature of the deep features, since they operate over the whole frame. When working with features extracted from the whole frame, the proposed HoS method could not take advantage of finding co-occurring local patterns in a single video frame or finding multiple local patterns within multiple frames.

Fig. 11 shows the recognition performance (mAP) of the proposed method and its extensions for each class. We observe that soft clustering has greater contribution for handshake class than the others, due to the high spatial variability in the execution of this

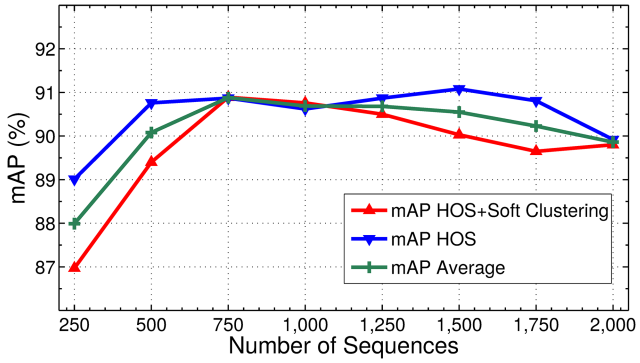

Fig. 10 *Effect of number of selected sequential patterns when the temporal pyramid is not used on TV Human Interactions data set after the sequential pattern selection step*

Table 3 Performance comparison on the TV Human Interactions data set when the negative videos are included in the retrieval task. 'DF' refers deep spatial features

Method	Annotation	mAP
Wang *et al.* [17]	not required	%53.4
Yu *et al.* [28]	not required	%56.0
HoS	not required	%38.2
HoS (SoftClust)	not required	%43.9
HoS (SoftClust + TempPyrd + DF)	not required	%53.7
HoS (SoftClust + TempPyrd)	not required	%57.4
Marín-Jiménez *et al.* [14]	automatic	%39.2
Patron-Perez *et al.* [15] (SL)	automatic	%42.4
Patron-Perez *et al.* [15] (SL)	manual	%54.8
Hoai and Zisserman [16]	manual	%56.3
Li *et al.* [27]	manual	%68.0

Table 4 Performance comparison on the TV Human Interactions data set when the negative videos are excluded. 'DF' refers deep spatial features

Method	Annotation	mAP
Gaidon *et al.* [25]	not required	%62.4
Ma *et al.* [30]	not required	%64.4
Yu *et al.* [28]	not required	%66.2
Khokher *et al.* [22]	not required	%69.1
HoS	not required	%51.4
HoS (SoftClust)	not required	%57.4
HoS (SoftClust + TempPyrd + DF)	not required	%59.8
HoS (SoftClust + TempPyrd)	not required	%69.2

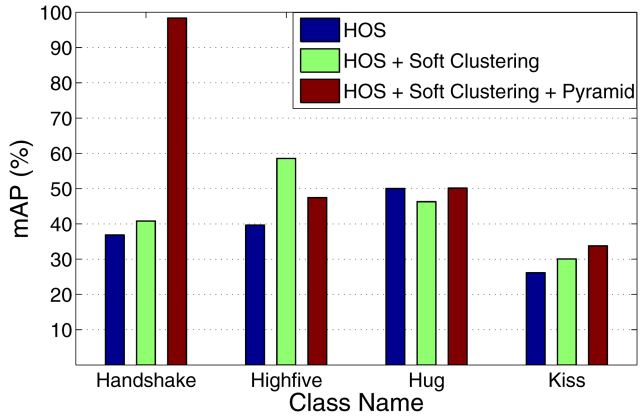

Fig. 11 *Classwise mAP performance of the proposed HoS representations, together with its extensions on TV Human Interactions data set. Best viewed in colour*

Table 5 Performance comparison on the UT Interaction data set

Method	Set #1 Acc	Set #2 Acc	AVG Acc
HoS + SoftClust + TempPyrd	%**95.0**	%**95.0**	%**95.0**
Kong and Fu [24]	%**95.0**	—	%**95.0**
Burghouts and Schutte [26]	%93.3	—	%93.3
Raptis and Sigal [35]	%93.3	—	%93.3
Zhang *et al.* [18]	%**95.0**	%90.0	%92.5
Yu *et al.* [28]	%93.3	%91.7	%92.5
Liang *et al.* [33]	—	—	%92.3
Sefidgar at al. [36]	%93.3	%90.0	%91.7
Vahdat *et al.* [29]	%93.0	%90.0	%91.5
HoS + SoftClust	%88.3	%93.3	%90.8
Marín-Jiménez *et al.* [14]	%86.0	%88.0	%87.0
HoS	%88.3	%85.0	%86.7
Ryoo [49]	%85.0	—	%85.0
Patron-Perez *et al.* [15]	%84.0	%86.0	%85.0
Zhang *et al.* [23]	%76.0	%78.0	%77.0
BOW [18]	%75.0	—	%75.0
Slimani *et al.* [20]	%40.6	%66.7	%53.6

The best scores are highlighted in bold font.

interaction. We observe that temporal pyramid has greater contribution for `Hug` class, and this is largely due to the temporal extent of the Hug interaction videos, i.e. they tend to be longer than other class instances. Kiss class as relatively lower performance and when we analyse the reason, we observe that a smaller number of local features have been generated since there is not much significant movement in execution of this interaction.

4.4 Experiments on UT interactions data set

We also test our method on another benchmark human interactions data set, UT Interactions. We compare our method with bag of words baseline and the state-of-the-art methods. The results are presented in Table 5.

According to results provided in Table 5, our method is significantly better than the standard BOW baseline and it achieves the state-of-the-art performance. In this data set, we observe that set#2 is more challenging than set#1, because set#2 has camera jitters and moving background. The contribution of the soft clustering to HoS representation is higher in set#2. Our method with soft clustering and hierarchical temporal pyramid gives the best results in set#2. In set#1, the method of Zhang *et al.* [18], Kong and Fu [24], and our method have the same performance. Note that, in these results, Marín-Jiménez *et al.* [14] and Patron-Perez *et al.* [15] did not include the class *point* in their experiments. We should also note that the method of Kong and Fu [24] requires bounding boxes or a person detector, while our method does not require such additional annotations and/or auxiliary tools.

Table 6 shows the confusion matrix when HoS representation is used with soft-clustering and hierarchical temporal pyramid. Most of the confusion is between `hug` and `handshake` classes. In the UT Interactions data set, the performance gain on accuracy of the hierarchical temporal pyramid and soft clustering is nearly +%4.

4.5 Time and space complexities

In terms of time complexity, the most complex step of our algorithm is the sequence mining step. The running time of algorithm is mostly dependent on the number of patterns in the search space. When there are many similar frequent patterns in a sequence database, the number of extracted sequential patterns may increase exponentially. In our experiments, we have searched for the min_sup that generates required number of patterns by starting min_sup from the %99 and decreasing with a linear weight until the required number of patterns are found. With this method, the total running time of the mining algorithm for UT Interactions was <25 min for each fold on a standard dual-core computer, and for

Table 6 Confusion table of UT Interactions data set for HoS representation is used with soft clustering and temporal pyramid

	Hshake	Hug	Kick	Point	Punch	Push
hshake	**19**	0	0	1	0	0
hug	2	**18**	0	0	0	0
kick	0	0	**20**	0	0	0
point	0	0	0	**20**	0	0
punch	0	0	1	0	**19**	0
push	0	0	1	0	1	**18**

The best scores are highlighted in bold font.

TV Human Interactions it takes <10 min. The space complexity of the algorithm is similar to the time complexity, the mining algorithm uses <4 Gb memory for both of the data sets.

5 Conclusion

In this work, we propose a novel representation for human interaction recognition in videos. The idea is to build histogram of local feature sequences to model the relations between local spatio-temporal feature points. To this end, we incorporate sequence mining methods from data mining research. We show that extracting temporal relations between local feature points via frequent sequence mining and representing them via HoS framework can improve the recognition performance.

By using sequence mining algorithms and sequence selection step, the experiments show that our final set of sequences not only frequent but also discriminative. According to the experiments, the hierarchical sequences and the soft clustering improves the recognition performance in all cases. With the help of the temporal pyramid for enhancing the histograms, the sequential patterns cover the whole video at multiple temporal scales and this hierarchical structure makes the HoS more robust and comprehensive.

The evaluation on two benchmark data sets shows the effectiveness of the proposed approach. Our method can be used with different local feature representations as desired and the local features which are robust to camera movements may improve the performance. While being on par with the state-of-the-art methods for human interaction recognition, our framework has several advantages, like requiring less supervision and manual annotations, less auxiliary tools, offering an easily extendible framework to work with other types of features.

Future work includes enhancing the proposed representation further with local CNN features and using this representation in conjunction with other deep learning techniques.

6 Acknowledgments

This research was supported in part by TUBITAK Career Development Award 112E149.

7 References

[1] Laptev, I., Marszaek, M., Schmid, C., *et al.*: 'Learning realistic human actions from movies'. IEEE Int. Conf. on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1–8

[2] Laptev, I.: 'On space-time interest points', *Int. J. Comput. Vision*, 2005, **64**, (2), pp. 107–123

[3] Niebles, J. C., Chen, C., Fei-Fei, L.: 'Modeling temporal structure of decomposable motion segments for activity classification'. European Conf. on Computer Vision, Crete, Greece, 2010

[4] Wang, L., Qiao, Y., Tang, X.: 'Mining motion atoms and phrases for complex action recognition'. IEEE Int. Conf. on Computer Vision, Sydney, Australia, 2013, pp. 2680–2687

[5] Wang, H., Schmid, C.: 'Action recognition with improved trajectories'. IEEE Int. Conf. on Computer Vision, Sydney, Australia, 2013, pp. 3551–3558

[6] Choi, J., Wang, Z., Lee, S., *et al.*: 'A spatio-temporal pyramid matching for video retrieval', *Comput. Vis. Image Underst.*, 2013, **117**, (6), pp. 660–669

[7] Ryoo, M.S., Aggarwal, J.K.: 'UT-Interaction dataset, ICPR'. Available at http://cvrc.ece.utexas.edu/SDHA2010/Human_Interaction.html

[8] Patron, A., Marszalek, M., Zisserman, A., *et al.*: 'High five: recognising human interactions in tv shows'. Proc. of the British Machine Vision Conf., Aberystwyth, England, 2010

[9] Aggarwal, J.K., Ryoo, M.S.: 'Human activity analysis: a review', *ACM Comput. Surv.*, 2011, **43**, (3), pp. 1–43

[10] Vrigkas, M., Nikou, C., Kakadiaris, I.: 'A review of human activity recognition methods', *Front. Robot. AI*, 2015, **2**, (28), pp. 1–28

[11] Weinland, D., Ronfard, R., Boyer, E.: 'A survey of vision-based methods for action representation, segmentation and recognition', *Comput. Vis. Image Underst.*, 2011, **115**, (2), pp. 224–241

[12] Marszałek, M., Laptev, I., Schmid, C.: 'Actions in context'. IEEE Int. Conf. on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009

[13] Wang, H., Ullah, M. M., Kläser, A., *et al.*: 'Evaluation of local spatio-temporal features for action recognition'. British Machine Vision Conf., London, England, 2009

[14] Marín-Jiménez, M., Yeguas, E., Nicolás, P.: 'Exploring stip-based models for recognizing human interactions in tv videos', *Pattern Recognit. Lett.*, 2013, **34**, (15), pp. 1819–1828

[15] Patron-Perez, A., Marszalek, M., Reid, I., *et al.*: 'Structured learning of human interactions in tv shows', *IEEE Pattern Anal. Mach. Intell.*, 2012, **34**, (12), pp. 2441–2453

[16] Hoai, M., Zisserman, A.: 'Talking heads: detecting humans and recognizing their interactions'. IEEE Conf. on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014

[17] Wang, H., Klaser, A., Schmid, C., *et al.*: 'Action recognition by dense trajectories'. IEEE Conf. on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011, pp. 3169–3176

[18] Zhang, Y., Liu, X., Chang, M., *et al.*: 'Spatio-temporal phrases for activity recognition'. 12th European Conf. on Computer Vision, Florence, Italy, 2012, pp. 702–721

[19] Ryoo, M.S., Aggarwal, J.K.: 'Spatio-temporal relationship match: video structure comparison for recognition of complex human activities'. IEEE 12th Int. Conf. on Computer Vision, Kyoto, Japan, 2009

[20] Slimani, K. N. e. H., Benezeth, Y., Souami, F.: 'Human interaction recognition based on the co-occurrence of visual words'. IEEE Conf. on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 2014, pp. 461–466

[21] Savarese, S., DelPozo, A., Niebles, J. C., *et al.*: 'Spatial-temporal correlations for unsupervised action classification'. Proc. of the IEEE Workshop on Motion and Video Computing, Copper Mountain, CO, USA, 2008

[22] Khokher, M. R., Bouzerdoum, A., Phung, S. L.: 'Human interaction recognition using low-rank matrix approximation and super descriptor tensor decomposition'. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 2017, pp. 1847–1851

[23] Zhang, B., Rota, P., Conci, N., *et al.*: 'Human interaction recognition in the wild: analyzing trajectory clustering from multiple-instance-learning perspective'. IEEE Int. Conf. on Multimedia and Expo, Turin, Italy, 2015, pp. 1–6

[24] Kong, Y., Fu, Y.: 'Max-margin action prediction machine', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, **38**, (9), pp. 1844–1858

[25] Gaidon, A., Harchaoui, Z., Schmid, C.: 'Activity representation with motion hierarchies', *Int. J. Comput. Vision*, 2014, **107**, (3), pp. 219–238

[26] Burghouts, G.J., Schutte, K.: 'Spatio-temporal layout of human actions for improved bag-of-words action detection', *Pattern Recognit. Lett.*, 2013, **34**, (15), pp. 1861–1869

[27] Li, B., Ayazoglu, M., Mao, T., *et al.*: 'Activity recognition using dynamic subspace angles'. IEEE Conf. on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 2011, pp. 3193–3200

[28] Yu, G., Yuan, J., Liu, Z.: 'Propagative hough voting for human activity recognition'. Proc. of the 12th European Conf. on Computer Vision, Florence, Italy, 2012, pp. 393–706

[29] Vahdat, A., Gao, B., Ranjbar, M., *et al.*: 'A discriminative key pose sequence model for recognizing human interactions'. IEEE Int. Conf. on Computer Vision Workshops, Barcelona, Spain, 2011, pp. 1729–1736

[30] Ma, S., Sigal, L., Sclaroff, S.: 'Space-time tree ensemble for action recognition'. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015

[31] Shugao, M., Zhang, J., Ikizler-Cinbis, N., *et al.*: 'Action recognition and localization by hierarchical space-time segments'. IEEE Int. Conf. on Computer Vision, Sydney, Australia, 2013, pp. 2744–2751

[32] Rodriguez, M. D., Ahmed, J., Shah, M.: 'Action mach: a spatio-temporal maximum average correlation height filter for action recognition'. Proc. of IEEE Int. Conf. on Computer and Pattern Recognition, Anchorage, AK, USA, 2008

[33] Liang, J., Xu, C., Feng, Z., *et al.*: 'Affective interaction recognition using spatio-temporal features and context', *Comput. Vis. Image Underst.*, 2016, **144**, pp. 55–165

[34] Liu, L., Shao, L., Rockett, P.: 'Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition', *Pattern Recognit.*, 2013, **46**, (7), pp. 1810–1818

[35] Raptis, M., Sigal, L.: 'Poselet key-framing: A model for human activity recognition'. IEEE Conf. on Computer Vision and Pattern Recognition, Oregon, PO, USA, 2013, pp. 2650–2657
[36] Sefidgar, Y. S., Vahdat, A., Se, S., *et al.*: 'Discriminative key-component models for interaction detection and recognition', *Comput. Vis. Image Underst.*, 2015, **135**, (C), pp. 16–30
[37] Tran, D., Bourdey, L. D., Fergus, R., *et al.*: 'C3d: generic features for video analysis'. IEEE Int. Conf. on Computer Vision, Santiago, Chile, 2015
[38] Karpathy, A., Toderici, G., Shetty, S., *et al.*: 'Large-scale video classification with convolutional neural networks'. IEEE Conf. on Computer Vision and Pattern Recognition, Washington, USA, 2014
[39] Feichtenhofer, C., Pinz, A., Zisserman, A.: 'Convolutional two-stream network fusion for video action recognition'. The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Zurich, Switzerland, 2016
[40] Li, Q., Qiu, Z., Yao, T., *et al.*: 'Action recognition by learning deep multi-granular spatio-temporal video representation'. Proc. of the 2016 ACM on Int. Conf. on Multimedia Retrieval, New York, NY, USA, 2016, pp. 159–166
[41] Simonyan, K., Zisserman, A.: 'Two-stream convolutional networks for action recognition in videos'. Proc. of the 27th Int. Conf. on Neural Information Processing Systems, Montreal, Canada, 2014, pp. 568–576
[42] Brox, T., Malik, J.: 'Object segmentation by long term analysis of point trajectories'. European Conf. on Computer Vision, Crete, Greece, 2010, pp. 282–295
[43] Dalal, N., Triggs, B., Schmid, C.: 'Human detection using oriented histograms of flow and appearance'. Proc. of the 9th European Conf. on Computer Vision, Graz, Austria, 2006, pp. 428–441
[44] Lee, H., Battle, A., Raina, R., *et al.*: 'Efficient sparse coding algorithms'. Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 2006, pp. 801–808
[45] Wang, J., Yang, J., Yu, K., *et al.*: 'Locality-constrained linear coding for image classification'. IEEE Int. Conf. on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 3360–3367
[46] Agrawal, R., Srikant, R.: 'Mining sequential patterns'. Proc. of the Eleventh Int. Conf. on Data Engineering, Taipei, Taiwan, 1995, pp. 3–14
[47] Hirate, Y., Yamana, H.: 'Generalized sequential pattern mining with item intervals', *J. Comput.*, 2006, **1**, (3), pp. 51–60
[48] Han, J., Wang, J., Li, C.: 'Frequent closed sequence mining without candidate maintenance', *IEEE Trans. Knowl. Data Eng.*, 2007, **19**, (8), pp. 1042–1056
[49] Ryoo, M.S.: 'Human activity prediction: early recognition of ongoing activities from streaming videos'. Proc. of the 2011 Int. Conf. on Computer Vision, Barcelona, Spain, 2011, pp. 1036–1043