Recognizing Human Action at a Distance in Video by Key Poses

Snehasis Mukherjee, Sujoy Kumar Biswas, Member, IEEE, and Dipti Prasad Mukherjee, Senior Member, IEEE

Abstract—In this paper we propose a graph theoretic technique for recognizing human actions at a distance in a video by modeling the visual senses associated with poses. The proposed methodology follows a bag-of-word approach that starts with a large vocabulary of poses (visual words) and derives a refined and compact codebook of key poses using centrality measure of graph connectivity. We introduce a ‘meaningful’ threshold on centrality measure that selects key poses for each action type. Our contribution includes a novel pose descriptor based on Histogram of Oriented Optical Flow (HOOF) evaluated in a hierarchical fashion on a video frame. This pose descriptor combines both pose information and motion pattern of the human performer into a multidimensional feature vector. We evaluate our methodology on four standard activity-recognition datasets demonstrating the superiority of our method over the state-of-the-art.

I. INTRODUCTION

Recognizing action of a distant human performer in video is useful in many applications ranging from wide-area video surveillance to evaluating game statistics by autonomous vision system. The challenge primarily lies in extracting distinct visual patterns from the gestures of human performer given the fact that tiny blob like appearance of the distant human figure (∼15% of the height of the frames) does not leave much room for modeling the limbs separately. The only reliable cue in such case is the pose specific information and we bank on the motion pattern of the poses to derive pose descriptors. Our emphasis on poses is based on the premise that human actions are composed of repetitive motion patterns and a sparse set of key poses often suffice to characterize an action. The proposed methodology follows the bag-of-word approach [1], [2]. Suppose we have $N$ documents containing words from a vocabulary of size $M$ and an $M \times N$ co-occurrence table $T$ is formed, where $T(w_i, d_j)$ stores the number of occurrences of a word $w_i$ in document $d_j$. In our approach, “words” refer to a vocabulary of human poses obtained from pose descriptor (content of each frame in a video is represented by a vector called pose descriptor) of the human figure and a “document” corresponds to the entire video sequence of a particular action type (viz. running, walking, jumping, etc.). We build a particular action descriptor by constructing a histogram of pose “words” occurring all through the video. Before describing the motivation, we first present the related works.

A. Related works

The initiatives in the field of human action recognition in image or video usually have two broad classifications - either they focus on low and mid-level feature collection (i.e., template-based approaches) or they model the high level interaction among the features (i.e., model-based approaches). Template-based approaches train suitable classifiers, or are compared to a set of event templates for recognition [3], [1]. Fathi et. al. [4] have proposed a shapelet feature based learning framework where mid level shape features are constructed from low level gradient features using AdaBoost algorithm. In a typical model-based approach, Mori et. al. have proposed a learned geometric model to represent human body parts in an image, where the action is recognized by matching the static postures in the image with the target action [5], [6]. Similarly, Cheung et. al. have used the silhouette of the body parts to represent the shape of the performer [7]. Recently, the bag-of-words model is being used to recognize actions in videos [3], [1].

Shah et. al. have used a vocabulary of local spatio-temporal volumes (called cuboids) and a vocabulary of spin-images (to capture the shape deformation of the actor by considering actions as 3D objects) [3]. Niebles et. al. also use some space-time interest points on the video as features (visual words) [1]. The algorithm of Niebles et. al., automatically learns the probability distributions of the visual words using graphical models like probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA). Messing et. al. [8] have proposed a generative mixture model for video sequences using velocity history of tracked key points, but they specifically focus on the high resolution videos. As opposed to “collection of words” representing each frame [1], Mori et. al. [2] have represented each frame as a single “word” achieving state-of-the-art accuracy in low resolution videos. There are some good efforts of combining global and local features for action recognition [9], [10], [11], [12]. Han et. al. have introduced a multiple kernel based classifier to automatically select and weight both low level and high level features for action recognition where body part detectors are used as “words” in their bag-of-words model [9]. Wang et. al. have combined global and local features and applied Hidden Conditional Random Field (HCRF) [10], [12] model for action recognition. Later Wang et. al. have learned the parameters in HCRF in a max-margin framework [11], [12] and called the new model as Max-Margin Hidden Conditional Random Field (MMHCRF). Liu et. al. have proposed a methodology that does not deal with the temporal behavior of the words and
the occurrence of words have been given equal importance throughout the video [13].

Bag-of-word based action recognition tasks either seek right kind of features or model the abstraction behind the video words. There are initiatives which study pose specific features [15], [14]. Duygulu et. al. have represented the human body by a collection of oriented rectangles and the orientations of these rectangles form a signature for each action [15]. In [14], Fengjun et. al. have introduced the concept of key poses. They search for a series of actions that best match the input sequence from existing action models. Each action is modeled as a series of synthetic 2D human poses rendered from a wide range of viewpoints. But modeling visual senses (for example, running, walking, jumping, etc.) associated with poses in videos is largely an unexplored research area. The proposed methodology deals with the visual senses associated with the poses, motivation of which is discussed next.

B. Motivation & Overview

Human poses often exhibit a strong visual sense (for example, running, walking, jumping, etc.) unambiguously. Describing the action type by merely looking at poses in video, is a challenging task given the tremendous variations present in the visual poses either in the form of external variation (viz. noise, especially in low resolution videos) or variation inherent to the human poses. Variation in poses is the primary source of visual sense ambiguity and a single pose may give out confusing interpretations about the related action type. For example, top row in Figure 1 shows some ambiguous poses as labeled by the proposed approach and by looking at them one cannot tell for certain the corresponding actions, whereas the bottom row illustrates the key poses (as labeled by the proposed approach) which unambiguously specify the related actions.

Our contribution in this paper is three-fold. First, we propose a multi-layered pose descriptor obtained in a hierarchical fashion for multiple spatial resolutions, and such pose descriptor captures both motion and pose information. The methodology we follow is to derive local HOOF from a weighted optical flow field at three spatial resolutions and then concatenating the local histograms into a single multidimensional pose descriptor (hierarchical HOOF). The foreground figure of the human performer appears as a tiny blob and the only reliable cue offered in such low-resolution videos is given by the motion pattern of the poses. Second, we seek to model visual senses exhibited by the human poses; this is similar to modeling word senses [16] – a single word implies different meaning in different contexts. Centrality theory is a standard tool to find out popular senses associated with a text word. For each visual sense (i.e. action type) we rank the poses in order of “importance” using centrality measure of graph connectivity [16]. Therefore, to sum up our first and second contributions, the proposed approach is a combination of a hierarchical HOOF and a graph theoretic key pose identification technique for distant action recognition which is in accordance with the theme of the paper. Our third contribution includes setting an appropriate threshold on the ranking order of poses in order to eliminate out ambiguous poses from key pose vocabulary. In order to achieve the unsupervised selection of threshold on pose ranking we introduce the statistical theory of ‘meaningfulness’ [17], and establish its foundation in the present context in the following Section.

C. Theory of Meaningfulness — Eliminating Ambiguous Poses

The main idea of meaningfulness is derived from the Gestalt hypothesis [17]. According to the Gestalt theory, “grouping” is the main concept for our visual perception [31]. Points in 2D or 3D space form any random point pattern(s). However, when this same point pattern(s) exhibits some common visual characteristics (e.g., alignment, parallelism, convexity, good continuation, connectivity, etc.), then this point pattern(s) is grouped to form a new, larger visual object called a Gestalt [31]. A set of partial Gestalts, due to certain matching visual
characteristics, can be grouped to form a Gestalt or meaningful shape.

Agnes et. al. [17] have shown some computational techniques to decide whether a given partial Gestalt is meaningful or not (partial Gestalt may represent an unambiguous pose descriptor in this context). For this, they have used a general perception law, called Helmholtz principle. Suppose there are \( n \) objects, \( k \) of them having similar visual characteristics with respect to some \( a \) priori knowledge (e.g., same color, same alignment or significantly low value of pose ambiguity measure in this context). Then the question is that, are these characteristics happening by chance, or is there any significant cause to group them to form a partial Gestalt? To answer this question, we first assume that the characteristics are uniformly distributed over all the \( n \) objects and the observed characteristics are some random realization of the uniform distribution. The Helmholtz principle states that if the expectation of the observed characteristics of \( k \) objects (henceforth we call a particular visual characteristics depicted by some objects as an event) is very small, then the grouping of these objects is a Gestalt (or meaningful). We calculate the expected number of occurrences of the event, which is called the number of false alarms (NFA) of that event [17]. The NFA of an event gives a measurement of meaningfulness of the event. The smaller the NFA is, the more meaningful the event is. (Good things come in small packages!) If the NFA is less than a certain number \( \epsilon \), the event is called an \( \epsilon \)-meaningful event; otherwise it is a random event.

In this paper, we introduce a meaningful cut-off value on the ambiguity measure (the graph centrality measure, discussed in the previous subsection) of the poses in the initial codebook, to choose some unambiguous poses to construct a compact codebook. Here we vary the cut-off over the range of the ambiguity measure and each time we calculate the expectation of the event that the cut-off to be meaningful. We will show in the result section that the theory of meaningfulness gives better result compared to the procedure of selecting \( q \)-best poses for each action type [18]. Grouping all such key poses together, we build our compact codebook.

Section II presents the detailed procedure to form the visual codebook. Section III describes the procedure of constructing compact codebook from the initial codebook by ranking the pose descriptors according to their ambiguity. Section IV states the procedure for finding key poses using the concept of meaningfulness. In Section V we illustrate various action descriptor models to train the histograms (frequencies of key poses) corresponding to all action types using the compact codebook. Section VI presents the results followed by conclusions in Section VII.

II. FORMATION OF INITIAL CODEBOOK

As discussed in the introduction, our first task is to derive a multidimensional vector (called the pose descriptor) corresponding to each frame of the video. The pose descriptors, upon data condensation result into a moderately compact representation and we call it an initial codebook of visual poses. First we discuss the methodology for deriving the descriptors.

A. Deriving the pose descriptor

The optical flow [19] based motion descriptors find applications in both classification and action synthesis [2]. Since video datasets used here often contain camera shake and jitters, (and since we can not guarantee a pixel-to-pixel correspondence in successive frames) the related instability keeps disturbing optical flow pattern time and again. We advocate the use of pose information in action recognition which remains unaffected by mild camera shake or jitters and does not rely on successive frames. Also, pose specific information provides valuable cue in studying tiny blob like human figure, where the limbs are not distinctly visible. The gradient field outlines the human pose pattern, against the backdrop of more or less uniform background (e.g. a field or playground, typical in wide area surveillance); the magnitude of the image gradient is higher along the human silhouette and almost zero on the uniform background. We weight the optical flow field with the gradient strength at each point. So our pose descriptor combines the benefit of motion information from optical flow and pose information from image gradient.

Suppose we have a video sequence of some action type \( A \) having \( M \) frames. The frame \( I_i, i \in 1, 2, ..., M \) is a grey image matrix defined as a function such that for any pixel \( (x, y) \), \( I(x, y) \in \Theta \), where \( (x, y) \in Z^2 \) and \( \Theta \subset Z^+ \) determines the range of the intensity values. Corresponding to each pair of consecutive frames \( I_{i-1} \) and \( I_i, i \in 2, ..., M \), we compute the optical flow field \( F \) [19]. We derive the image gradient \( \vec{B} \) for frame \( I_i \) and we produce a weighted flow field vector \( \vec{V} \) using,

\[
\vec{V} = \vec{B} \cdot F,
\]

where the symbol ‘*’ represents the point wise multiplication of the two matrices.

The effect of this weighted optical flow field is best understood if one treats the gradient field as a band pass filter. The optical flow vectors, sometimes appearing in large magnitude on the background away from the human figure (originated due to signal noise or unstable motion), get “blurred” and eventually smoothed out by the gradient field. This is because the gradient field takes high value where the edge is prominent, preferably along the edge boundary of the human figures, but it is very low in magnitude on the background space. This way we capture the motion information of the poses, and hence the proposed descriptor is basically a motion-pose descriptor preserving the motion pattern of the human pose.

Figure 2 shows how proposed pose descriptor differs from the HOOF [20] (derived from Figure 2(a)) and the Histogram of Oriented Gradient (HOG) [21] (derived from Figure 2(b)). Due to instability in the computation of optical flow, Figure 2(a) shows noisy flow vectors even on the background; the gradient field is fairly effective in delineating the silhouette of the human figure, but misses the motion information. In contrast, the proposed pose descriptor in Figure 2(c) shows weighted optical flow vectors accumulated around the key moving parts – in this case legs – of a walking man (from soccer dataset). The gradient field attenuates the noisy optical flow and scales the latter around the key moving parts to build up a succinct representation of the pose movement.
We distribute the field vectors of $\vec{V}$ in an $L$-bin histogram. Here each bin denotes a particular octant in the angular radian space. We take the value of $L$ as 8, because orientation field is quantized enough when resolved in eight directions in digital grid, i.e., in every 45 degrees. Taking $\vec{V} = (V_x, V_y)$, histogram $H = \{h(1), h(2), ..., h(L)\}$ construction takes place by quantizing $\theta(x, y) = \arctan (\frac{V_y}{V_x})$ and adding up $m(x, y) = \sqrt{V_x^2 + V_y^2}$ to the bin indicated by the quantized $\theta$. In mathematical notation,

$$h(i) = \sum_{x,y} \begin{cases} m(x, y) & \text{when } \theta \text{ in } i^{th} \text{octant} \\ 0 & \text{otherwise} \end{cases}.$$  

We consider a three-layer image pyramid (Figure 3) to generate pose descriptor. While the first layer is the entire image matrix, the second layer in the image pyramid splits the image into 4 equal blocks and each block produces one 8-bin histogram leading to 32-dimensional histogram vector. Similarly, the third layer has 16 blocks and hence produces 128-dimensional histogram vector. All the vectors are $L1$-normalized separately for each layer and concatenated together resulting in a 168-dimensional pose descriptor. We can consider this as a hierarchical HOOF that build our pose descriptor. We pre-process the optic flow vector by weighting it with gradient strength prior to the construction of hierarchical pose descriptor.

We note the immunity of proposed pose descriptor in tackling temporal variation of human actions. A human performer may execute the same task at different speeds. Variation in speed is manifested in the local histogram pattern (the three-layer pyramidal structure) capturing the finer details that make two actions distinguishable. However, the global histogram pattern at the bottom-layer (Figure 3) of the pyramid, captures the overall motion pattern. When the same kind of activity with different speed variations exist, global histogram pattern aids the classifier to generalize the underlying pattern from speed variations of same action. Capturing motion and pose pattern at various scales is absent in HOG/HOOF; we argue that in addition to (1) this multiscale representation plays a crucial role in difference of performance (Table I), especially for videos where human performer is substantially distant/small (soccer, tower and hockey datasets).

Once we have pose descriptors we seek to quantize them for a compact representation of the descriptor set. Next section outlines the details of the visual codebook generation process.

### B. Generating initial codebook

Since human action has repetitive nature, an efficient pose descriptor derived above retains redundancy. Vector quantization by K-means [1] or K-medoids [2] are some of the clustering techniques seeking a compact representation of the descriptor sets. K-means clustering suffers from the initialization issue. Instead of clustering, we solicit the idea of data condensation because in data condensation one may afford to select multiple prototypes from the same cluster, whereas in case of clustering one seeks to identify the true number of clusters (and also their true partitioning). The data condensation leaves us with an initial codebook $S$ of visual poses where some of the redundancies of the pose space is eliminated. The formation of the initial codebook follows Maxdiff kd-tree based data condensation technique [22].

Let $C_j = \{p_1, p_2, \ldots, p_T\}$ be the shortest $j$-th sequence that gets repeated in an entire video of action type $A_n$ ($n \in \{1, 2, \ldots, \alpha\}$, $\alpha$ is the number of action types), where $p_i$, $i \in \{1, 2, \ldots, T\}$, is pose descriptor and $T$ is the length of the action cycle. For a video of action type $A_n$, $C_i$ and $C_j$ ($i \neq j$) together contain redundant information as many of the poses that occur in $C_i$ also occur in $C_j$. This follows from cyclic nature of the human action. Our intention here is not to get true partitioning of the pose space but to obtain a large vocabulary $S$ where at least some of the redundancies of the pose space is eliminated. An optimum (local) lower bound on the codebook size of $S$ can be estimated by Akaike Information Criterion (AIC), or Bayesian Information Criterion (BIC) [23] or one can directly employ X-means algorithm [24]. The Maxdiff kd-tree based data condensation technique alleviates the curse of dimensionality by mining the multi-dimensional pose descriptors into a kd-tree data structure. The kd-tree (a binary search tree) is formed with root containing all pose descriptors of an action. At each node the pivot dimension and pivot element are estimated making the splitting rule at that node. In case of Maxdiff kd-tree the pivot dimension at each node is computed by finding out in which dimension of the pose descriptors the separation (between two numbers) is maximum. The leaf nodes of the kd-tree denote pose clusters or the visual pose words; one can choose (depending on computational expense) multiple samples from each leaf node to construct the vocabulary $S$ for action type $A_n$, where $S = \{p_i \in \Re^d | i = 1, 2, \ldots, k\}$, $k$ being the cardinality of $S$. Here $d$ denotes the dimensionality of the pose descriptor. The algorithm to construct the kd-tree is explained in details in [22].

In our experiment we choose the mean of each leaf node as our pose word and learn the codebook $S$ of poses. The pose descriptor in a query video is mapped to a pose word in $S$ by descending down the kd-tree (by the tree traversal algorithm) and hitting the leaf node (Figure 3). If one selects multiple poses from the same leaf node in the construction of $S$, one can break the tie by computing the nearest neighbor (i.e., having lowest Euclidian distance) of the pose descriptor. Next we outline the scheme for ranking the poses of $S$. 

Fig. 2. (a) optical flow field, (b) gradient field and (c) weighted optical flow of one frame taken from the action “walk right” of soccer dataset. (c) shows weighted optical flow vectors nicely outline the key moving parts of the foreground body pose.
III. RANKING OF POSES USING CENTRALITY MEASURE OF GRAPH CONNECTIVITY

The initial codebook \(S\) contains poses which are often ambiguous and confusion remains in deciding what sense does a particular pose in \(S\) signify. Subsection III-A outlines the motivation behind the choice of our model; Subsection III-B demonstrates the construction of pose graphs and Subsection III-C explains how we rank poses depending on the measure of their ambiguities.

A. Motivation behind the measure of ambiguity

Poses in Figures 1(a) and 1(b) are confusing in interpreting running or walking actions. To eliminate confusions like this we identify key poses. For each action type \(A_n\) we construct a pose graph which captures how poses in \(S\) occur over the entire action length. We can define an action type \(A_n = \{C_1, C_2, \ldots, C_H\}\) as a sequence of action cycles, \(H\) being the number of action cycles in a particular video of an action type. We have already mentioned that each action cycle \(C_i\) has pose sequence \(p_1, p_2, \ldots, p_T\). When we see a pose \(p_j\) that is strongly suggestive of some particular action, we can expect that most of the action cycles \(C_i\) contain that pose; for example, Figure 1(g) illustrates one such key pose which is difficult to miss in any action cycle of a jumping video. Note that such key pose \(p_j\) not necessarily has maximum number of occurrences in action video \(A_n\). Rather such pose must occur at uniform interval, neither only in the beginning nor only at the end and nor intermittently in between. A pose \(p_j\) which has a strong sense associated (indicating that it represents action \(A_n\)) must have the highest cardinality of the following set \(\lambda_{p_j|A_n}\) given by

\[
\lambda_{p_j|A_n} = \{C_i \mid C_i \subset A_n \text{ and } p_j \in C_i\}. \tag{3}
\]

Poses which occur all of a sudden or are not prominent in each action cycle can be considered as deviations from the regular poses. Such irregular pose patterns are ambiguous and they need to be pruned out from \(S\). They happen primarily because of tremendous variability of human poses and to a lesser extent, because of associated noise (viz. shadows in tower dataset, transmission noise in soccer dataset, etc.). Next we construct a graph called pose graph for each action type.

B. Construction of pose graph

A pose graph is constructed using the poses in \(S\) as the nodes. The edge weight between each pair of poses stands for votes for importance cast by other pose nodes. The idea is that the poses which strongly characterize an action pattern should have consistently high weight between them, and ambiguous poses which occur arbitrary would get low votes from their neighborhoods [28]. The idea of “importance” of a pose is represented based on the notion of centrality - a pose is central if it is maximally connected to all other poses. The key poses are identified in the graph separately for each action repeating the same algorithmic procedure for all kinds of actions. We define a pose graph for a specific kind of actions as follows:

**Definition 1:** A pose graph for a given action type \(A_n\) is an undirected edge-labeled graph \(G = (S, E)\) where each vertex in \(G\) corresponds to a pose belonging to the initial codebook \(S\); \(E\) is the set of edges and \(\omega : E \rightarrow (0, 1]\) is the edge weight function. There is an undirected edge between the poses \(u\) and \(v\) \((u \neq v\) and \(u, v \in S\)), with edge weight \(\omega(u, v)\), where \(0 < \omega(u, v) \leq 1\). Edge weights indicate the relative contribution of two poses in representing the entire action pattern. It is assumed that no self loop exists in the pose graph.

As discussed earlier, human activity follows a sequence of pose patterns in definite order and they have a cyclic nature. For simplicity we have assumed the cycle length fixed and used a span of \(T\) frames to define an action cycle. Most of the repetitive actions in our datasets (like running, walking, jumping, etc.) complete a full cycle in and around 10 frames and we set the value of \(T\) at 10. Then the edge weight function is as follows:

\[
\omega(u, v) = \begin{cases} 
\frac{1}{\rho(u, v)} & \text{when } \rho(u, v) \neq 0 \\
0 & \text{otherwise}
\end{cases}, \tag{4}
\]

where \(\rho(u, v)\) denote how many times the pose words \(u\) and \(v\) both occur together in all the action cycles of a particular
action video. That means,
\[ \rho(u, v) = |\lambda_u|_A \cap \lambda_v|_A|. \] (5)

Construction of \( G \) requires computation of \( \rho(u, v) \) (Appendix A). For computing the set \( \lambda_u|_A \), one has to map each video frame (the pose descriptor derived from this frame) to the most appropriate element in \( S \) (using (3)). Next we define the centrality measure [25] of graph connectivity to see how important the poses are from each other in a pose graph.

C. Ranking of poses

Our next task is to evaluate the poses in \( S \) according to their “importance” in characterizing an action. An equivalent problem exists in natural language processing [16] and social network analysis [26], where importance of a node (may be a webpage or more specifically a person) in a network is identified using centrality measure of graph connectivity. Such ranking technique is in use in web search (Google’s PageRank algorithm [27]) and it has been used successfully in feature ranking for object recognition [28] and feature mining task [29]. Inspired by their success we rank poses to choose the key poses in \( S \).

There are various centrality measures of graph connectivity to accomplish the ranking task in a pose graph [16]. Basically the choice of connectivity measure influences the selection process of the highly ranked poses in \( S \). Let \( e(u) \) be a suitable measure for graph connectivity of the pose \( u \in S \). For all poses in \( S \), we induce a ranking \( \text{rank}_e \) of the poses based on \( e \) such that for any two poses \( u, v \in S \), \( \text{rank}_e(u) \leq \text{rank}_e(v) \) if \( e(u) \geq e(v) \). We adopt eccentricity as the measure \( e \) of graph connectivity [26].

Eccentricity

Definition 2: For each action, we choose the unambiguous path to evaluate the eccentricity \( e \). Eccentricity over all the chosen equidistant points in \([0,1]\). We call this low eccentricity values from \( S \) to produce the compact codebook \( \Xi \) following the definition:

\[ \Xi = \{ u \in S \mid e(u) < \delta_n \}, \delta_n \text{ being a meaningful cut-off value of } e(u) \text{ for } A_n. \]

Next we discuss the process of determining the meaningful cut-off value of \( e(u) \) for the pose \( u \) to be a key pose.

A. Selecting meaningful cut-off for eccentricity

Our next aim is to select a sufficiently low cut-off value of the eccentricity measure \( e \) value [17]. We call this low cut-off value as meaningful cut-off for \( e \) value. Any pose in \( S \) having \( e \) value lower than the meaningful cut-off value \( \eta \) is termed as meaningful pose in this paper. As noted in the introduction, our objective is to select from each action type \( A_n (n \in \{1, 2, ..., \alpha\}) \), only some poses \( u \in S \) having meaningfully low \( e \) value.

Now the problem is, how many key poses should be selected from each action for constructing the compact codebook? One procedure may be to select \( q \)-best poses (in terms of lowest eccentricity) from each action to form the compact codebook. The problem in this process is that, in reality the number of unambiguous poses is usually different for different action type. The concept of meaningfulness [17], as discussed in the introduction, gives us the opportunity to vary the number of key poses for different action type. Let us call the key poses selected using the meaningfulness concept, as meaningful poses. Next we illustrate the process of forming compact codebook \( \Xi \) from the initial codebook \( S \) following the concept of meaningfulness.

IV. FORMATION OF COMPACT CODEBOOK BY MEANINGFUL POSES

We have an initial codebook \( S \) of a number of 168-dimensional pose descriptors (visual words), each having a normalized (between 0 and 1) eccentricity value depicting a measure of centrality. The poses with ‘meaningfully’ low eccentricity values are selected from \( S \) to produce the compact codebook \( \Xi \) following the definition:

Definition 3: Given a sequence of action types \( \{A_n\}_{n=1,2,...,\alpha} \) \( \alpha \) is the number of action types and initial codebook \( S \), a set of poses \( \Xi(\subset S) \) is said to be a compact codebook if

(i) \( \Xi = \{ u \in S \mid e(u) < \delta_n \}, \delta_n \text{ being a meaningful cut-off value of } e(u) \text{ for } A_n \).

(ii) \( \forall A_n, \Xi \ni u \in S \text{ such that } u \in \Xi \).

Next we discuss the process of determining the meaningful cut-off value of \( e(u) \) for the pose \( u \) to be a key pose.

Now the problem is, how many key poses should be selected from each action for constructing the compact codebook? One procedure may be to select \( q \)-best poses (in terms of lowest eccentricity) from each action to form the compact codebook. The problem in this process is that, in reality the number of unambiguous poses is usually different for different action type. The concept of meaningfulness [17], as discussed in the introduction, gives us the opportunity to vary the number of key poses for different action type. Let us call the key poses selected using the meaningfulness concept, as meaningful poses. Next we illustrate the process of forming compact codebook \( \Xi \) from the initial codebook \( S \) following the concept of meaningfulness.
becomes a Bernoulli trial. If $P_n^{(\eta)}$ is the probability of the event that the cut-off $\eta$ is meaningful, then

$$P_n^{(\eta)} = \sum_{i=t}^{D} \binom{D}{i} \nu^i (1-\nu)^{D-i}, \quad (6)$$

is the Binomial tail, where $D$ is the number of pose descriptors in $A_n$.

Then the number of false alarms (NFA) of the event (as discussed in the introduction) that the particular cut-off $\eta$ is significant for detecting meaningful poses, can be defined as

$$NFA = \tau P_n^{(\eta)}. \quad (7)$$

We calculate the NFA for each $\eta$ using (6) and (7). If the value of NFA is less than a predefined number $\epsilon$, then the corresponding cut-off value $\eta$ is $\epsilon$-meaningful.

The computation of $\eta$ and hence determining $t$ poses with eccentricities lying below $\eta$ directly influence our search for key poses. Next we estimate a suitable value for $t$ to find key poses.

**B. Estimation of parameters for meaningful cut-off**

The probability that all the poses in the action type $A_n$ have $e$ greater than $\eta$ is $\nu^D$. This is lesser or equal to the probability that at least $t$ poses have $e$ less than $\eta$, which is $P_n^{(\eta)}$. So $\nu^D \leq P_n^{(\eta)} < \frac{\epsilon}{\tau}$ (since according to (7), $NFA = \tau P_n^{(\eta)} < \epsilon$), which implies

$$D \geq \frac{\log \epsilon - \log \tau}{\log \nu} \quad (8)$$

For a given $\eta$, $\nu$ is known. The total number of pose descriptors $D$ is fixed for a given action type $A_n$. Then (8) shows the dependency between the parameters $\epsilon$ and $\tau$.

Now let us determine $t$, the minimum number of poses needed to recognize the action type $A_n$. From Hoeffding's inequality [32], for an $\epsilon$-meaningful event we can deduce the following:

$$t \geq \nu D + \sqrt{\frac{D}{2} (\log \tau - \log \epsilon)}. \quad (9)$$

The equation (9) is the sufficient condition of meaningfulness. The derivation of (9) is shown in the Appendix B. In our experiment, we take the minimum number of $t$ satisfying (9) to minimize the number of key poses and hence reduce the size of $\Xi$.

**C. Selecting maximal meaningful cut-off**

We set $\epsilon = 1$ as in [17]. This means the expected number of occurrence of the event that the cut-off $\eta = \eta'$ is meaningful for the action type $A_n$ is less than 1. Let us call this cut-off $\eta'$ as ‘1-meaningful’ cut-off value. It is clear from (6) that if $\eta'$ is a 1-meaningful cut-off value for $e$, then each cut-off value chosen from the interval $[\eta',1]$ is also 1-meaningful. Now from the 1-meaningful cut-off values, we have to select the maximal meaningful cut-off.

If $\epsilon = 1$ is a crude estimate to find the maximal meaningful cut-off for $e$, we should have some measure of meaningfulness. For this purpose, we should consider the empirical probability of a pose $u$ of action type $A_n$ to have its $e$ value to fall in the interval $[\eta',1]$. If $r_n(\eta')$ be the empirical probability of a pose of $A_n$ to have $e$ in the interval $[\eta',1]$, then

$$r_n(\eta') = \frac{D(\eta')}{D}, \quad (10)$$

where $D(\eta')$ is the number of poses in $A_n$ having $e$ greater than $\eta'$.

In general, for 1-meaningful cut-off values, $r_n(\eta') < \nu$. Using $r_n(\eta')$, we have to define a measurement of maximal meaningfulness of the cut-off value. This measurement should penalize the situation that a 1-meaningful cut-off $\zeta$ yields higher empirical probability value than $\nu$. This measurement (let us call it as $c$-value) should also help to reduce the number (not compromising with the accuracy of recognizing the action type) of meaningful poses for an action type. However, according to Definition 3, the corresponding action type $A_n$ must have at least one selected pose. Then $c$-value can be defined as,

$$c_n(\zeta) = \begin{cases} \infty & r_n(\zeta) \log \frac{r_n(\zeta)}{\nu} + (1-r_n(\zeta)) \log \frac{1-r_n(\zeta)}{1-\nu} \quad \text{when } r_n(\zeta) \geq \nu \\ r_n(\zeta) & \text{otherwise} \end{cases} \quad (11)$$

where $\zeta$ can take any value from the interval $(\eta',1)$. We take the open interval instead of the closed interval in order to avoid division by zero in (11).

For each action type $A_n$, we find the $c_n(\zeta)$ for all $\frac{1}{\tau}$ distant values of $\zeta$ from the interval $(\eta',1)$. Clearly, a more meaningful value of $\eta'$ gives lesser value of $c_n(\zeta)$. For each action type $A_n$, we find the maximal meaningful cut-off using the following definition:

**Definition 4:** A cut-off $\zeta$ is said to be maximal meaningful cut-off for the corresponding action type, if it is 1-meaningful and $\forall \ m \in (\eta',1) - \{\zeta\}, \ c_n(\zeta) \leq c_n(m)$.

The frames with $e$ value greater than $\zeta$ for the action type $A_n$ are finally chosen as meaningful poses and included in the compact codebook $\Xi$. Next we discuss the process of training the histograms corresponding to the different action types using $\Xi$.

**V. LEARNING THE ACTION DESCRIPTORS**

After a compact codebook $\Xi$ is estimated, we discuss the process of building action descriptors corresponding to action videos using $\Xi$. For each action video we build the corresponding action descriptor by finding the occurrences of the key poses in each frame $I_r$, $r = 1, 2, \ldots, M$. This happens by extracting pose descriptor $q_r$ for each frame $I_r$ and then mapping it to some key pose in $\Xi$ and thereby making a histogram count that gives number of occurrences of each key pose in the video sequence. There are four types of action descriptor models which address the mapping issues between a pose descriptor (obtained from a video frame) and a key pose, namely traditional codebook, kernel codebook, plausibility and uncertainty model based mappings [33]. Action descriptor measures how well a pose descriptor matches with a key pose in $\Xi$. We produce results using these four kinds of codebook mapping models in the results section.

Traditional action descriptor model follows hard partitioning. Given a vocabulary (i.e., the compact codebook obtained
after training) of poses, the traditional bag-of-words implementation requires allocating each pose descriptor \( q_r \) of the query action video, to a key pose \( p_i \in \Xi \) and thereby building the codebook histogram \( AD_T \) (Action Descriptor following Traditional action descriptor model), where each bin \( i \) of \( AD_T \) keeps the occurrence count of key pose \( p_i, \ i = 1, 2, \ldots, k \) (\( k \) is the number of key poses).

\[
AD_T(i) = \frac{1}{M} \sum_{r=1}^{M} \left\{ 1 \quad \text{when} \quad i = \arg \min_j(Distance(p_j, q_r)) \right\}
\]

(12)

where \( Distance(p_j, q_r) \) denotes any suitable distance metric, i.e., the Euclidean distance.

But such hard allocation fails to take into account the fact that how one can derive discrete visual words (pose descriptors) with distinct identity in the continuum of video attributes. That is, a question may arise, how do we know whether a pose from the query video is a key pose (i.e., belongs to \( \Xi \)) or not? It may happen that the pose corresponding to a video frame is not a key pose and in that case \( \Xi \) has no suitable candidate for allocation. This problem is solved by modeling the codeword plausibility [33]. In plausibility model, a robust alternative to the codeword histogram is the kernel density estimation which may be a Gaussian kernel \( K_σ(x) = \frac{1}{\sqrt{2\pi}σ} \exp(-\frac{x^2}{2σ^2}) \).

A kernel weighted function provides the remedy for hard mapping of codewords to the compact codebook. In fact, it measures the weight that says how closely the codeword matches with the key pose. Hence we obtain the kernel action descriptor model \( AD_K \) (Kernel action descriptor) as follows:

\[
AD_K(i) = \frac{1}{M} \sum_{r=1}^{M} K_σ(Distance(p_j, q_r)).
\]

(13)

The action descriptor \( AD_P \) (Plausibility action descriptor) following plausibility model can be defined as,

\[
AD_P(i) = \frac{1}{M} \sum_{r=1}^{M} \left\{ K_σ(Distance(p_j, q_r)) \quad \text{when} \quad i = \arg \min_j(Distance(p_j, q_r)) \right\}
\]

(14)

The action descriptor \( AD_U \) (Uncertainty action descriptor) following uncertainty model can be defined according to [33] as,

\[
AD_U(i) = \frac{1}{M} \sum_{r=1}^{M} \frac{K_σ(Distance(p_j, q_r))}{\sum_{i=1}^{N} K_σ(Distance(ξ_i, q_r))},
\]

(15)

where \( N \) is the number of key poses in \( \Xi \) and \( ξ_i \) is the \( i \)th key pose in \( \Xi \).

The key pose plausibility gives higher weight to more relevant pose descriptor. If a pose descriptor happens to be a key pose, the relevant key pose in \( \Xi \) associates higher weight to that pose descriptor. On the other hand if the pose descriptor is ambiguous and hence does not correspond to any of the key poses in \( \Xi \), then the action descriptor gets a lower weight corresponding to that ambiguous pose descriptor. In fact, such ambiguous pose descriptor contributes insignificantly to the histogram corresponding to an action video. Once we derive histograms corresponding to all action types, the histograms act as feature descriptors for a given video sequence for classification. We train histograms for all the action types and classify the actions using support vector machines with radial basis function. Next we illustrate the results to show the efficiency of our approach.

VI. RESULTS AND DISCUSSIONS

The choice of dataset for experimentation is made keeping in mind the focus of our paper - recognizing action at a distance. Soccer [34], Tower [35], Hockey [36] datasets contain human performer far away from the camera and the height of the performer is ~15% of the height of the frames. Only exception is the KTH [37] dataset where we have evaluated our proposed methodology on medium size (~40% of the height of the frames) human figure.

Since the aim of this paper is not to track or segment human figure, but to study the motion patterns of a distant performer, the datasets we have used already have the bounding boxes with performer (approximately) centralized. So by frame we refer to a bounding box, the size of which may or may not be same in successive frames. Since such bounding boxes do not leave much room for background, different action cycles in an action sequence can not appear in different image positions. In case it does — as in few of the action videos in KTH dataset — we crop the human figure to its appropriate size. Note that such cropping destroys the pixel-to-pixel correspondence between two successive frames, but our proposed methodology is designed specifically to alleviate this problem.

A. Experiments

We use support vector machines for classification of target video with radial basis function following a “Leave-one-out” scheme. We measure the accuracy of our method (as well as the competing methods) by percentage of correct detection of actions. For each dataset we show the confusion matrix obtained using the proposed approach. Our approach is efficient in detecting human actions in comparison to state-of-the-arts as shown in Table I. Table I also shows that use of the compact codebook increases the efficiency of the proposed approach compared to the result obtained by using the initial codebook \( S \). The major time consuming step is learning the meaningful poses for each action separately. But this is done once and we reap benefit later while classifying video with a small set of just 4 or 5 selected poses per actions. The average time consumed for learning meaningful poses by each action amounts to little less than one minute. After the detection of meaningful poses in \( S \), our approach takes only a few seconds for both learning and testing in our MATLAB™ version 7.0.4 implementation in a machine with processor speed 2.37 GHz, 512MB RAM.

It is true that camera shake or jitter and camera zooming or panning may influence the weighted optical flow patterns. But in the proposed approach, the optic flow field is weighed with gradient strength to boost the flow vectors along the silhouette of the figure. More importantly, we do not resort to pixel-level study (that would invite noise into consideration), rather we study the overall motion patterns of poses. Our three-layer architecture studies both gross and finer details
of motion patterns at three different spatial resolutions. Table I shows that the proposed approach gives less than 80% accuracy for Soccer dataset, where camera shaking disturbs the proposed approach in recognition process. But even in that case the proposed approach gives better result compared to the competing approaches.

We show in Table I the results of our experiments using raw HOOF [20] (row 3) and HOG [21] (row 4) features as well as hierarchical HOOF without using (1) (row 5) and hierarchical HOG (row 6). Table I shows that hierarchical implementations of HOOF/HOG in multiple layers work better than their raw counterparts, but our proposed hierarchical HOOF computed on gradient weighted optic flow (using (1)) outperforms all others.

Table I shows the results on the whole KTH data for each of the methods. We have made a further study on the KTH dataset which is shown in Table II. KTH has 150 video segments of each of the four different scenarios: outdoor, outdoor with scale variation, outdoor with different cloths and indoor. Out of the 600 videos in KTH dataset, 150 contain rapid camera zooming and panning (scale variation). In that case, the accuracy of the proposed method degrades by 13%, but it gives better accuracy compared to the competing methods. Table II shows the results of the proposed and the competing approaches on each of the four scenarios.

The accuracy graphs in Figure 4 show the accuracy of our approach for different action descriptor models [33] over different number of key poses per action. Among the four types of action descriptor models described in Section V, the plausibility model described by (14) gives better accuracy for all the four datasets, as shown in Figure 4 and also in Table III.

Table III also shows us how the concept of meaningfulness enhanced the efficiency of the proposed approach. This gives consistent result compared to the result obtained by selecting some fixed number of poses for each action type to construct the compact codebook. All the accuracy measurements given in Table III are obtained by the plausibility model.

In [13], all occurrences of words (analogous to poses in present context) are given equal importance throughout the video without considering the temporal behavior of them. In contrast, the edge weight function (4) captures the relative contribution of each pose in representing the entire action pattern. Also, to what size the final vocabulary will reduce is still an open problem in [13], and one has to again rely on trying with different values of K in K-means clustering to obtain the final high level vocabulary (Figure 5). Also the vocabulary size is slightly larger as shown in Figure 5. Moreover, action descriptor proposed in [13] seems to be the histograms built over the learned semantic vocabulary. Since the vocabulary size is already reduced, there may not be any one-to-one correspondence between a pose descriptor and the new pose word; some of the pose descriptors could well be assigned to a pose which is no longer a part of the condensed vocabulary. Hence, we have experimented with various descriptor-to-word assignment models, but any such discussion is missing in [13]. Table I summarizes the performance of our pose descriptor when used in diffusion distance embedding [13].

In [14], Fengjun et. al. have selected some points on the silhouette of the human performer and have calculated the motion energy (sum of squared displacements of similar points) between each two successive frames and choose the frame with locally optimum motion energy. Datasets like KTH, where the size of performer is large enough, permits explicit choice of joint positions on human body. But for other datasets where the human performer is distant and small, it is difficult to get good local optima. The graphs in Figure 6 shows that attempt at explicit joint tracking do not pay much reward in noisy and low resolution datasets as used in our experiments unlike high resolution images of indoor scenes of [14].

First row of Figure 7 shows the key poses (“running” sequence of Tower dataset) selected by the proposed approach. Key poses in Figure 7(e)-(l) are extracted using [14] which are too many including some less prominent ones. Figure 8 shows
TABLE II

PERFORMANCE OF PROPOSED METHOD COMPARED TO THE STATE-OF-THE-ARTS ON KTH DATASET

<table>
<thead>
<tr>
<th>Methods</th>
<th>KTH outdoor</th>
<th>KTH varying scale</th>
<th>KTH cloth change</th>
<th>KTH indoor</th>
<th>Whole KTH data</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-LDA [2]</td>
<td>98.33</td>
<td>91.17</td>
<td>99.53</td>
<td>91.87</td>
<td>91.20</td>
</tr>
<tr>
<td>HOOF feature [20]</td>
<td>100</td>
<td>76.67</td>
<td>99.33</td>
<td>92</td>
<td>92</td>
</tr>
<tr>
<td>HOOF feature [21]</td>
<td>100</td>
<td>74.67</td>
<td>99.33</td>
<td>92</td>
<td>92</td>
</tr>
<tr>
<td>Diffusion maps embedding [13]</td>
<td>88</td>
<td>74.67</td>
<td>97.33</td>
<td>89.33</td>
<td>90.33</td>
</tr>
<tr>
<td>Proposed Method using poses in S</td>
<td>80</td>
<td>65.67</td>
<td>78.67</td>
<td>67.33</td>
<td>67.65</td>
</tr>
<tr>
<td>Proposed Method using meaningful poses in S</td>
<td>100</td>
<td>79.33</td>
<td>100</td>
<td>92</td>
<td>92.83</td>
</tr>
</tbody>
</table>

Fig. 4. Accuracy plot with number of key poses per action using different action descriptor models for (a) Soccer dataset, (b) Tower dataset, (c) Hockey dataset and (d) KTH dataset.

TABLE III

ACCURACY (%) OF PROPOSED METHOD COMPARED TO WHEN CHOOSING SOME FIXED NUMBER OF POSES PER ACTION

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Proposed method with number of selected poses per action</th>
<th>Proposed method using meaningful poses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soccer</td>
<td>50.30, 75.53, 80.46, 87.33, 77.33, 84.41, 5.05, 92.33</td>
<td></td>
</tr>
<tr>
<td>Tower</td>
<td>87.41, 84.34, 80.34, 88.33, 71.33, 85.33, 57.33, 92.33</td>
<td></td>
</tr>
<tr>
<td>Hockey</td>
<td>86.35, 85.35, 87.35, 89.33, 72.35, 85.33, 55.33, 92.33</td>
<td></td>
</tr>
<tr>
<td>KTH</td>
<td>86.35, 87.67, 84.33, 90.33, 79.33, 89.33, 50.33, 92.33</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 5. Performance of Liu et al. [13] with respect to the codebook size for (a) Soccer (our codebook size 20), (b) Tower (our codebook size 22), (c) Hockey (our codebook size 21) and (d) KTH datasets (our codebook size 20). The dotted lines represent performance using initial vocabulary and the continuous lines represent performance using semantic vocabulary [13].

Guity is high and only a handful of unambiguous poses exist in each action class. So the confusion matrix (Table IV) of the proposed approach is obtained by taking fewer number of poses (Table III) for each of the eight actions in comparison to other datasets.

2) Tower dataset [35]: The Tower dataset for human action recognition consists of 108 video sequences of nine different actions of six different people, each person showing each
(Table V) illustrates the class wise recognition rate for each action. The nine actions are, “pointing (P)”, “standing (S)”, “digging (D)”, “walking (W)”, “carrying (C)”, “running (R)”, “wave 1 (W1)”, “wave 2 (W2)”, “jumping (J)”. The Tower dataset is actually a collection of aerial action videos where the performer is filmed from tower top and he appears as a tiny blob of height 30 pixels approximately. The approximate bounding rectangles of the human performer as well as foreground filter-masks are supplied with the dataset.

Since each video clip contains a single action, the video clips are already grouped into respective action classes and we do not need any preprocessing step. The confusion matrix (Table V) illustrates the class wise recognition rate for each action twice. The nine actions are, “pointing (P)”, “standing (S)”, “digging (D)”, “walking (W)”, “carrying (C)”, “running (R)”, “wave 1 (W1)”, “wave 2 (W2)”, “jumping (J)”.

**Table IV**

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>D</th>
<th>L</th>
<th>Lu</th>
<th>Rd</th>
<th>R</th>
<th>Rl</th>
<th>W</th>
<th>Wr</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lu</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>W</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Wr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

**Table V**

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>D</th>
<th>L</th>
<th>Lu</th>
<th>Rd</th>
<th>R</th>
<th>Rl</th>
<th>W</th>
<th>Wr</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lu</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>W</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Wr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

**Table VI**

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>D</th>
<th>L</th>
<th>Lu</th>
<th>Rd</th>
<th>R</th>
<th>Rl</th>
<th>W</th>
<th>Wr</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lu</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>W</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Wr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

**Table VII**

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>D</th>
<th>L</th>
<th>Lu</th>
<th>Rd</th>
<th>R</th>
<th>Rl</th>
<th>W</th>
<th>Wr</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>L</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Lu</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rd</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>R</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Rl</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>W</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Wr</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
</tbody>
</table>

**B. Discussion**

Although, our proposed algorithm produces state of the art accuracies on four standard datasets specifically suited for distant action recognition, we would like to make a thorough discussion highlighting the advantage and limitation of the proposed methodology.

First, any action recognition model should be robust to speed variations. The three-layer architecture captures both globally descriptive and locally detailed motion patterns of poses. When a performer acts faster, the rapid displacement of body parts (between two successive video frames) will result in larger optic flow vectors; but while computing histogram patterns for each of the three layers, we normalize them separately and the associated variability diminishes restoring
the overall action pattern. Also, speed variation alters the temporal length of action, but key poses — that characterize a motion pattern — will keep on occurring throughout the action sequence, and their occurrence will be reflected in the action descriptor. Hence our action descriptor is not very sensitive to speed variations, we map each frame (i.e. pose descriptor) in an action video specifically to one (hard assignment) or many (soft assignment) key poses in the refined codebook; the intention is to see the distribution of key poses all over the action sequence.

One important point in relation to the last passage is that action cycle does not directly influence the action descriptor. Action cycle only assists in the construction of pose graph, and also in identifying pose nodes that are central to characterizing an action pattern. The purpose behind keeping the action cycle fixed is to divide the action sequence roughly in some uniform intervals. The idea is to see which poses are occurring in a regular fashion, distinct enough to detect in almost at regular intervals. The ambiguous poses often occur intermittently, and may be in the beginning or at the end. Our intention is to get rid of such irregularity; key poses are supposed to happen regularly and a rough division of a video sequence in equal intervals is quite sufficient to identify them. However, when the speed variation is noticeably different resulting in altogether different key poses, of course, our method will not classify such actions as same. But in that case the actions usually become different, for example running and jogging actions may be in the beginning or at the end. Our intention is to get the distribution of key poses all over the action sequence.

Another important point is that we intend to move towards more granular level of action recognition. Hence, we have classified “run right” and “run left” as two separate events (Soccer). However, the proposed methodology can recognize both such events as same category if one just keeps relevant action descriptors in the training set as training instances (as we did in KTH & Tower datasets achieving state of the art accuracy).

VII. CONCLUSIONS

This paper studies the action recognition with meaningful poses. From an initial and large vocabulary of poses, the proposed approach prunes out ambiguous poses and builds a small but highly discriminatory codebook of meaningful poses. The proposed sparse descriptor of human poses uses histogram of oriented field vectors in a multi-resolution framework. By the notion of centrality theory of graph connectivity we extract the meaningful poses which, we argue, contain semantically important information in describing the action in context. Forming a codebook of meaningful poses, we evaluate our methodology on four standard datasets of varying complexity levels and report improved performance when compared with benchmark algorithms. Presently our algorithm works for single performer; extending it to recognize multiple actions in the same scene may be a future research direction.

APPENDIX A

Algorithm 1 for Mining Poses Using Maxdiff kd-tree [22]

Algorithm 1 Maxdiff kd-tree data structure [22] for pose mining

1. treetype kdtree {
   left child : lpointer;
   right child : rpointer;
   pivot dimension : pivotval;
   pointer to pose in S : poseindex;
}

2. getpose(x, kdtree node)
   input: pose descriptor x ∈ ℜ^d, kd-tree root;
   output: index of pose p in S that best matches x;
   if((node.lpointer == NULL) OR (node.rpointer == NULL))
       return node.poseindex;
   else {
       if[x[node.pivotdimen] ≥ node.pivotval]
           return getpose(x, node.rpointer);
       else return getpose(x, node.lpointer);
   }

APPENDIX B

Proof of Equation (9)

In our problem, D is the number of poses in action type A_n in the codebook. Then we can formulate the problem by an i.i.d. sequence of T random variables \{X_q\}_{q=1,2,3,...,T}, such that 0 < X_q ≤ 1. Let us define X_q as,

\[ X_q = \begin{cases} 
1 & \text{when } \epsilon(q) < \eta \\
0 & \text{otherwise} 
\end{cases} \]

for a given \eta. We set \[ S_D = \sum_{q=1}^{D} X_q \] (i.e., the number of poses of A_n having \epsilon value greater than \eta) and \[ \nu D = E[S_D] \]. Then for \nu D < t < D (since \nu is a probability value less than 1), putting \sigma = \frac{1}{\nu} as in [17], according to Hoeffding’s inequality,

\[ P_{n}^{(\eta)} = P(S_D ≥ t) ≤ e^{-D(\sigma \log \frac{\sigma}{\nu} + (1-\sigma) \log \frac{1-\sigma}{1-\nu})}. \]

In addition, the right hand term of this inequality satisfies,

\[ e^{-D(\sigma \log \frac{\sigma}{\nu} + (1-\sigma) \log \frac{1-\sigma}{1-\nu})} ≤ e^{-D(\sigma-\nu)^2 H(\nu)} ≤ e^{-2D(\sigma-\nu)^2}, \]

where

\[ H(\nu) = \begin{cases} 
\frac{1-\frac{\sigma}{\nu} \log \frac{\sigma}{\nu}}{2\nu(1-\nu)} & \text{when } 0 < \nu < \frac{1}{2} \\
\frac{1}{2} & \text{when } \frac{1}{2} \leq \nu < 1 
\end{cases} \]
This is Hoeffding’s inequality. We then apply this for finding the sufficient condition of $\epsilon$-meaningfulness. If $t \geq \nu D + \sqrt{\frac{\log \tau - \log \epsilon}{H(\nu)}}$, then putting $\sigma = \frac{\epsilon}{\tau}$ we get

$$D(\sigma - \nu)^2 \geq \frac{\log \tau - \log \epsilon}{H(\nu)}. \quad (20)$$

Then using (17) and (20) we get,

$$P_n(\nu) \leq e^{-D(\sigma - \nu)^2 H(\nu)} \leq e^{-\log \tau + \log \epsilon} = \frac{\epsilon}{\tau}. \quad (21)$$

This means by definition of $\eta$ it is meaningful (according to (7)).

Since for $\nu$ in $(0,1)$, $H(\nu) \geq 2$ (according to (19)) so from (21) we get the sufficient condition of meaningfulness as (9).

REFERENCES