Hierarchical Union-of-Subspaces Model for Human Activity Summarization

Tong Wu †, Prudhvi Gurram ‡, Raghuveer M. Rao ‡, Waheed U. Bajwa †
† Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854
‡ U.S. Army Research Laboratory, Adelphi, MD 20783

Abstract

A hierarchical union-of-subspaces model is proposed for performing semi-supervised human activity summarization in large streams of video data. The union of low-dimensional subspaces model is used to learn meaningful action attributes from a collection of high-dimensional video sequences of human activities. An approach called hierarchical sparse subspace clustering (HSSC) is developed to learn this model from the data in an unsupervised manner by capturing the variations or movements of each action in different subspaces, which allow the human actions to be represented as sequences of transitions from one subspace to another. These transition sequences can be used for human action recognition. The action attributes can also be represented at multiple resolutions using the subspaces at different levels of the hierarchical structure. By visualizing and labeling these action attributes, the hierarchical model can be used to semantically summarize long video sequences of human actions at different scales. The effectiveness of the proposed model is demonstrated through experiments on three real-world human action datasets for action recognition and semantic summarization of the actions using different resolutions of the action attributes.

1. Introduction

The need for semantic summarization of large streams of video data has increased due the tremendous growth in the amount of user-generated video data [10] as well as the growth in number of surveillance cameras recording and producing large amounts of video data [8]. Video summarization is particularly important for applications involving bandwidth constrained environments like autonomous systems. All these systems should be able to perceive and recognize the actions or activities in the video sequences, summarize them semantically and transmit only the summaries for further evaluation and planning. Hence, high-level event or activity recognition from large streams of video data has attracted a lot of attention recently due to many practical commercial, law enforcement, and military applications [1, 9, 18, 20]. Human activity recognition approaches can be broadly divided into two types – single layered and hierarchical [1]. In single-layered approaches, sequential or space-time algorithms such as Hidden Markov Models (HMM) [11] are used for action recognition, no matter how complex the action is. On the other hand, if a complex human activity is considered to be a hierarchical model [9], it can be represented as a structure with different levels that show parts of the activity at varying resolutions. The bottom-most level of the hierarchy consists of the lowest level description (highest resolution) of an action, i.e., movement of the human body (e.g., right arm moves up, left arm moves up, torso bending, legs moving apart) and can be called as an action attribute [13]. At the next higher level, a sequence of these attributes forms a human action. As we climb this hierarchical structure, the human actions and their interactions with other human actions and objects form activities, while a sequence of these activities forms an event. An important advantage of the hierarchical model is that such structures go hand in hand with semantic or syntactic approaches. Each attribute, action, activity and event can be given a semantic label in such a case. In bandwidth constrained environments, there exist large amounts of high-resolution video data as in the case of autonomous systems, it is highly desirable to represent a long video sequence in a semantic form and transmit the semantic summary instead of the video itself [16]. Hierarchical models provide us with the flexibility to transmit such semantic information at different resolutions based on the needs of the end application.

In this paper, we focus on the bottom two layers of the complex human activity hierarchical model, i.e., human actions and their representation using attributes. One possible way of obtaining this representation is to manually define the action attributes and assign training data for each of these attributes [13]. Another way is to manually annotate videos by labeling movements (action attributes) [19]. Then, any human action in a test sequence can be described using such user-defined attributes. However, a set of user-defined action attributes may not completely describe all the human actions in the data. Also, manual assignment

of training data for each of the action attributes is time consuming, if possible, for large datasets. To overcome this problem, some research has been done to learn data-driven action attributes by clustering low-level features based on their co-occurrence in training videos [5, 14, 15]. However, video data are not usually well distributed around the cluster means and hence, the cluster statistics may not be sufficient to accurately represent the attributes.

Here, we propose a hierarchical union-of-subspaces (UoS) model to learn human action attributes from the data at different resolutions in an unsupervised manner. Inspired by eigenfaces [23] and the fact that video data are not uniformly distributed in the ambient space [4], we conjecture that the action attributes represented by subspaces can encode more variations within an attribute compared to the representations obtained using cluster statistics as done in previous methods [5,14,15]. We use the silhouette structure of the human (after background suppression and thresholding) as the feature in our approach. Each action attribute is represented by a subspace built from the silhouette features. Each frame of any human action (a sequence of silhouette frames) is assigned to a subspace based on what attribute or movement is taking place in the frame. Thus any human action can be represented as a sequence of transitions from one subspace to itself or to another subspace. Moreover, the hierarchical structure provides multi-resolution attributes of human actions. Different human actions can share one or more higher resolution action attributes. For example, if an action attribute represents a human standing in upright position, it can be shared by many human actions such as walking, bending, and waving.

One of the applications of learning the subspaces based on hierarchical UoS model is semantic summarization of long video sequences using labeled action attributes. Since we use silhouette features in this work, the subspaces corresponding to each action attribute can be visualized using the first few dimensions of the corresponding orthonormal bases and each attribute can be assigned a semantic label. Thus, semantic vocabulary is built for a dataset. A long video sequence can be semantically interpreted and summarized using the semantic labels of the subspaces to which the frames are assigned and the transitions between the subspaces. Due to the multi-resolution nature of the hierarchical structure, this semantic description of the human action can be performed at different resolutions. If training labels are available for human actions, another major application of this representation is human action recognition. Classifiers can be trained for each of the actions based on the subspace transition sequence and can be used to recognize human actions in a test video.

We use Sparse Subspace Clustering (SSC) [4], a state-of-the-art subspace clustering method, as the basic subspaces learning algorithm and build our hierarchical UoS learning algorithm on top of it. Note that in the remainder of the paper, we use the words "subspace" and "action attribute" interchangeably for convenience. We use bold lower case letters and bold upper case letters to represent vectors and matrices, respectively. Given a vector $\mathbf{v}$, its i-th element is denoted by $\mathbf{v}(i)$. The (i,j)-th element of a matrix $\mathbf{A}$ is denoted by $a(i,j)$.

2. Background: Sparse Subspace Clustering

We start with a brief review of the Sparse Subspace Clustering (SSC) algorithm described in [4]. Suppose we are given a collection of N signals in $\mathbb{R}^m$, denoted by $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{R}^{m \times N}$, and assume these N samples are drawn from a union of L subspaces $\{\mathcal{S}_\ell\}_{\ell=1}^L$ of dimension $\{d_\ell\}_{\ell=1}^L$ in $\mathbb{R}^m$, where every signal belongs to one of the subspaces in $\{\mathcal{S}_\ell\}_{\ell=1}^L$. Therefore, we can write $\mathbf{Y} = \mathbf{Y}_1 \cup \mathbf{Y}_2 \cup \cdots \cup \mathbf{Y}_L$ where each $\mathbf{Y}_\ell \in \mathbb{R}^{m \times N_\ell}$ is a submatrix of $\mathbf{Y}$ containing all the samples that belong to subspace $\mathcal{S}_\ell$ with $N_\ell > d_\ell$, and we have $\sum_{\ell=1}^L N_\ell = N$. Based on the intuition that each sample $\mathbf{y}_i$ can be expressed as a sparse linear combination of the data points from the same subspace to which $\mathbf{y}_i$ belongs, one can represent $\mathbf{y}_i$ as follows:

$$\widehat{\mathbf{a}}_i = \arg\min_{\mathbf{a}_i} \|\mathbf{a}_i\|_1 \quad \text{s.t.} \quad \mathbf{y}_i = \mathbf{Y}\mathbf{a}_i,\ \mathbf{a}_i(i) = 0, \quad (1)$$

where $\mathbf{a}_i = [\mathbf{a}_i(1), \mathbf{a}_i(2), \ldots, \mathbf{a}_i(N)]^T \in \mathbb{R}^N$ is the coefficient vector and $\|\mathbf{a}_i\|_1 = \sum_{j=1}^N |\mathbf{a}_i(j)|$. Considering all the data points in a matrix form, SSC learns a sparse coefficient matrix $\widehat{\mathbf{A}} = [\widehat{\mathbf{a}}_1, \widehat{\mathbf{a}}_2, \ldots, \widehat{\mathbf{a}}_N] \in \mathbb{R}^{N \times N}$ by minimizing the following objective function:

$$\widehat{\mathbf{A}} = \arg\min_{\mathbf{A}} \|\mathbf{A}\|_1 \quad \text{s.t.} \quad \mathbf{Y} = \mathbf{Y}\mathbf{A},\ \operatorname{diag}(\mathbf{A}) = \mathbf{0},$$

where $\operatorname{diag}(\mathbf{A})$ is the diagonal vector of matrix $\mathbf{A}$ and $\mathbf{0}$ denotes the zero vector. Using the resulting coefficient matrix $\widehat{\mathbf{A}}$, the segmentation of data points into respective subspaces $\mathbf{Y}_1, \ldots, \mathbf{Y}_L$ can be done by applying spectral clustering [17] on the similarity matrix $\mathbf{W} = |\widehat{\mathbf{A}}| + |\widehat{\mathbf{A}}|^T$, where $|\cdot|$ denotes the element-wise absolute value operation. In the case when data are corrupted by noise, SSC solves the following convex optimization problem for $\widehat{\mathbf{A}}$:

$$\widehat{\mathbf{A}} = \arg\min_{\mathbf{A}} \|\mathbf{A}\|_1 + \lambda \|\mathbf{Y} - \mathbf{Y}\mathbf{A}\|_F^2 \quad \text{s.t.} \quad \operatorname{diag}(\mathbf{A}) = \mathbf{0}.$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm and $\lambda > 0$ is a regularization parameter. The authors in [4] proposed an efficient solution for calculating the sparse coefficients $\widehat{\mathbf{A}}$ for this problem using Alternating Direction Method of Multipliers (ADMM) [2]. Note that there also exist some other UoS learning approaches, such as robust subspace clustering (RSSC) [22] and robust subspace clustering via thresholding (TSC) [7]. In this paper, we propose our hierarchical

UoS learning algorithm based on SSC due to its superior performance, although our approach is extendable to other algorithms.

3. Hierarchical Sparse Subspace Clustering

In this section, we introduce our Hierarchical Sparse Subspace Clustering (HSSC) algorithm for learning multiple levels of UoS using a collection of high-dimensional data. We use $\mathbf{Y}_{p,\ell} \in \mathbb{R}^{m \times N_{p,\ell}}$ to denote the set of signals that are assigned to the ℓ-th subspace at the p-th level of the hierarchical structure, where $N_{p,\ell}$ is the number of signals in $\mathbf{Y}_{p,\ell}$. Let L_p denote the number of subspaces at the p-th level, then we have $\sum_{\ell=1}^{L_p} N_{p,\ell} = N$ and $\mathbf{Y} = \bigcup_{\ell=1}^{L_p} \mathbf{Y}_{p,\ell}$ for all p's. The subspace underlying $\mathbf{Y}_{p,\ell}$ is denoted by $\mathcal{S}_{p,\ell}$ and its orthonormal basis is denoted by $\mathbf{D}_{p,\ell} \in \mathbb{R}^{m \times d_{p,\ell}}$, where $d_{p,\ell}$ denotes the dimension of the subspace $\mathcal{S}_{p,\ell}$.

We begin by applying SSC on $\mathbf{Y}$ at the first level ($p = 1$), which divides $\mathbf{Y}$ into two subspaces with clusters $\mathbf{Y} = \mathbf{Y}_{1,1} \cup \mathbf{Y}_{1,2}$ such that $\mathbf{Y}_{1,1} \cap \mathbf{Y}_{1,2} = \emptyset$. At the second level, we again perform SSC on $\mathbf{Y}_{1,1}$ and $\mathbf{Y}_{1,2}$ separately and divide each of them into 2 clusters, yielding 4 clusters $\mathbf{Y} = \bigcup_{\ell=1}^{4} \mathbf{Y}_{2,\ell}$ with $\mathbf{Y}_{1,\ell} = \mathbf{Y}_{2,2\ell-1} \cup \mathbf{Y}_{2,2\ell}$ ($\ell = 1, 2$). Using the signals in $\mathbf{Y}_{2,\ell}$ ($\ell = 1, \ldots, 4$), we estimate the four subspaces $\mathcal{S}_{2,\ell}$'s underlying $\mathbf{Y}_{2,\ell}$'s by identifying their orthonormal bases $\mathbf{D}_{2,\ell}$'s. To be specific, we obtain eigendecomposition of the covariance matrix $\mathbf{C}_{2,\ell} = \mathbf{Y}_{2,\ell}\mathbf{Y}_{2,\ell}^T$ as $\mathbf{C}_{2,\ell} = \mathbf{U}_{2,\ell}\mathbf{\Sigma}_{2,\ell}\mathbf{U}_{2,\ell}^T$, where $\mathbf{\Sigma}_{2,\ell} = \mathrm{diag}(\lambda_1, \ldots, \lambda_{N_{2,\ell}})$ is a diagonal matrix ($\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{N_{2,\ell}}$) and $\mathbf{U}_{2,\ell} = [\mathbf{u}_1, \ldots, \mathbf{u}_{N_{2,\ell}}]$. Then the dimension of the subspace $\mathcal{S}_{2,\ell}$, denoted by $d_{2,\ell}$, is estimated based on the energy threshold, i.e., $d_{2,\ell} = \arg\min_d \frac{\sum_{j=1}^{d} \lambda_j}{\sum_{j=1}^{N_{2,\ell}} \lambda_j} \geq \alpha$, where α is a predefined threshold and is set close to 1 for better representation. The orthonormal basis of $\mathcal{S}_{2,\ell}$ can then be written as $\mathbf{D}_{2,\ell} = [\mathbf{u}_1, \ldots, \mathbf{u}_{d_{2,\ell}}]$. After this step, we end up with 4 subspaces with clusters $\{\mathbf{Y}_{2,\ell}\}_{\ell=1}^4$ and their associated orthonormal bases $\{\mathbf{D}_{2,\ell}\}_{\ell=1}^4$.

For every $p \geq 2$, we decide whether or not to further divide each single cluster or subspace at the p-th level into two clusters or subspaces at the $(p+1)$-th level based on the following principle. We use a binary variable $B_{p,\ell}$ to indicate whether the cluster $\mathbf{Y}_{p,\ell}$ is further divisible at the next level or not. If it is, we set $B_{p,\ell} = 1$, otherwise $B_{p,\ell} = 0$. We initialize $B_{2,\ell} = 1$ for all ℓ's ($\ell = 1, \ldots, 4$). Consider the cluster $\mathbf{Y}_{p,\ell}$ at the p-th level and assume that M clusters already exist at the $(p+1)$-th level derived from $\{\mathbf{Y}_{p,1}, \mathbf{Y}_{p,2}, \ldots, \mathbf{Y}_{p,\ell-1}\}$. If $B_{p,\ell} = 0$, the $(M+1)$-th cluster at the $(p+1)$-th level will be the same as $\mathbf{Y}_{p,\ell}$; thus, we simply set $\mathbf{Y}_{p+1,M+1} = \mathbf{Y}_{p,\ell}$ and $B_{p+1,M+1} = 0$. If $B_{p,\ell} = 1$ (in which case $\mathbf{Y}_{p,\ell}$ corresponds to the green nodes in Fig. 1), we first split $\mathbf{Y}_{p,\ell}$ into two sub-clusters $\mathbf{Y}_{p,\ell} = \mathbf{Z}_1 \cup \mathbf{Z}_2$ using SSC and find the best subspaces $\mathcal{S}_{\mathbf{Z}_k}$ ($k = 1, 2$) that fit $\mathbf{Z}_k$'s respectively using the aforementioned strategy, while their dimensions and orthonormal bases are denoted by $d_{\mathbf{Z}_k}$'s and $\mathbf{D}_{\mathbf{Z}_k}$'s, respectively. Then for every signal $\mathbf{y}_i$ in $\mathbf{Z}_k$ ($k = 1, 2$), we compute the relative reconstruction error of $\mathbf{y}_i$ using the parent subspace basis $\mathbf{D}_{p,\ell}$ and the child subspace basis $\mathbf{D}_{\mathbf{Z}_k}$, which are defined as $e_i = \frac{\|\mathbf{y}_i - \mathbf{D}_{p,\ell}\mathbf{D}_{p,\ell}^T\mathbf{y}_i\|_2^2}{\|\mathbf{y}_i\|_2^2}$ and $\widehat{e}_i = \frac{\|\mathbf{y}_i - \mathbf{D}_{\mathbf{Z}_k}\mathbf{D}_{\mathbf{Z}_k}^T\mathbf{y}_i\|_2^2}{\|\mathbf{y}_i\|_2^2}$, respectively. The means of the relative reconstruction errors of all the signals in $\mathbf{Z}_k$ using $\mathbf{D}_{p,\ell}$ and $\mathbf{D}_{\mathbf{Z}_k}$ are denoted by E_k and $\widehat{E}_k$, respectively. Finally, we divide $\mathbf{Y}_{p,\ell}$ into $\mathbf{Z}_1 \cup \mathbf{Z}_2$ if (i) the relative reconstruction errors of the signals using the child subspace are less than the reconstruction errors of the signals using the parent subspace by a certain threshold, i.e., $(E_k - \widehat{E}_k)/E_k \geq \beta$ for either $k = 1$ or 2, and (ii) the dimensions of the two child subspaces meet a minimum requirement, that is, $\min(d_{\mathbf{Z}_1}, d_{\mathbf{Z}_2}) \geq d_{\min}$. In here, β and $d_{\min}$ are user-defined parameters and are set to avoid redundant subspaces. When either β or $d_{\min}$ decreases, we will have more subspaces. Assuming the two conditions are satisfied, the cluster $\mathbf{Y}_{p,\ell}$ is then divided by setting $\mathbf{Y}_{p+1,M+1} = \mathbf{Z}_1$ ($B_{p+1,M+1} = 1$) and $\mathbf{Y}_{p+1,M+2} = \mathbf{Z}_2$ ($B_{p+1,M+2} = 1$). The bases of the subspaces at the $(p+1)$-th level are set by $\mathbf{D}_{p+1,M+1} = \mathbf{D}_{\mathbf{Z}_1}$ and $\mathbf{D}_{p+1,M+2} = \mathbf{D}_{\mathbf{Z}_2}$. If the above conditions are not satisfied, we set $\mathbf{Y}_{p+1,M+1} = \mathbf{Y}_{p,\ell}$, $B_{p,\ell} = 0$ and $B_{p+1,M+1} = 0$ to indicate $\mathbf{Y}_{p,\ell}$, i.e., $\mathbf{Y}_{p+1,M+1}$, is a leaf cluster and this cluster will not be divided any further (which corresponds to the yellow nodes in Fig. 1). This process is repeated until we reach a predefined maximum level in the hierarchy denoted by P. The hierarchical SSC algorithm for any level $p \geq 2$ is described in Algorithm 1.

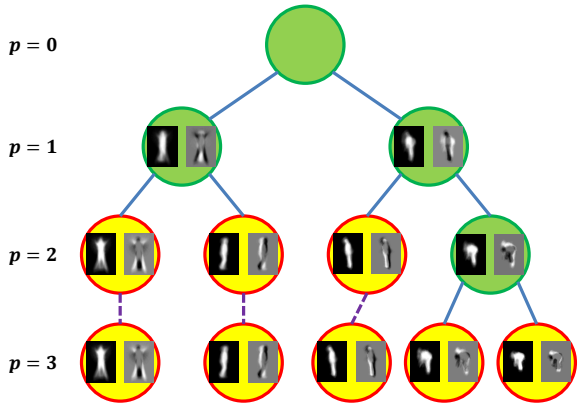


Figure 1. An example of using HSSC to learn a hierarchical UoS model. Each circle represents one cluster/subspace, and the first two vectors of each $\mathbf{D}_{p,\ell}$ are plotted in each circle. The green nodes represent the clusters that are further divided in the next level of the hierarchy. Leaf nodes are represented as yellow, these clusters cannot be further divided and are the final attributes obtained at the bottom most level of the hierarchical model.

Fig. 1 also shows an example of applying HSSC to deter-

mine the action attributes for three actions: bend, jumping jack and jump in the Weizmann dataset [6]. Here, the maximum number of levels in the model P is set to 3 because we don't expect to have more than $2^3 = 8$ subspaces at the bottom level for only three actions. The hierarchical UoS model is initialized with the silhouette features of all the actions from multiple subjects at the top ($p = 0$). Each silhouette frame (obtained after background suppression and thresholding) in the video sequence is a data sample $\mathbf{y}_i$ in both SSC and HSSC. Inside each node, we visualize the first two basis vectors of the subspaces obtained at each level. We can see that at the first level ($p = 1$), the attributes corresponding to two actions, jumping jack and jump are represented by one subspace and attributes corresponding to the bend action are represented by another subspace. At the second level ($p = 2$), the action attributes corresponding to jumping jack and jump are separated into two different subspaces. While the action attributes of the bend action, which is a more complex action with wider range of movement, are divided into two subspaces, representing the more upright part of the bend action as one higher resolution attribute and the lower part of the bend action as another higher resolution attribute. The lower part of the bend action is further divided into two more attributes at the next level ($p = 3$) while the other attributes are left as they were. Thus, we can see that as p increases, the variations within each action can be identified, extracted and represented using more number of higher resolution action attributes.

The proposed HSSC algorithm has some obvious advantages over flat SSC for learning human action attributes. First, in the case of flat SSC, one has to define the number of subspaces into which the data are to be clustered [4]. This requirement puts a constraint of prior knowledge about the data in that we need to know the number of underlying human action attributes present in the data. Such an approach moves away from data-driven learning. On the other hand, HSSC algorithm only requires the knowledge of a maximum level P, and it can stop before it reaches the P-th level if no clusters can be further divided. HSSC algorithm is designed in such a way that all the variations within each action can be identified automatically to determine the final number of action attributes. Second, HSSC provides us with multiple resolutions of action attributes, which are extremely useful for semantic labeling and understanding of human actions as explained in Section 1. The multi-resolution attributes can be used for semantic summarization of long video sequences at different resolutions starting from giving just an overview of the action to detailed explanation of movements occurring in the video. Flat SSC can provide us with only one set of action attributes at one single resolution, which depends on the number of subspaces that is the input to the algorithm. The empirical results presented in Section 4 illustrate these benefits of HSSC with examples.

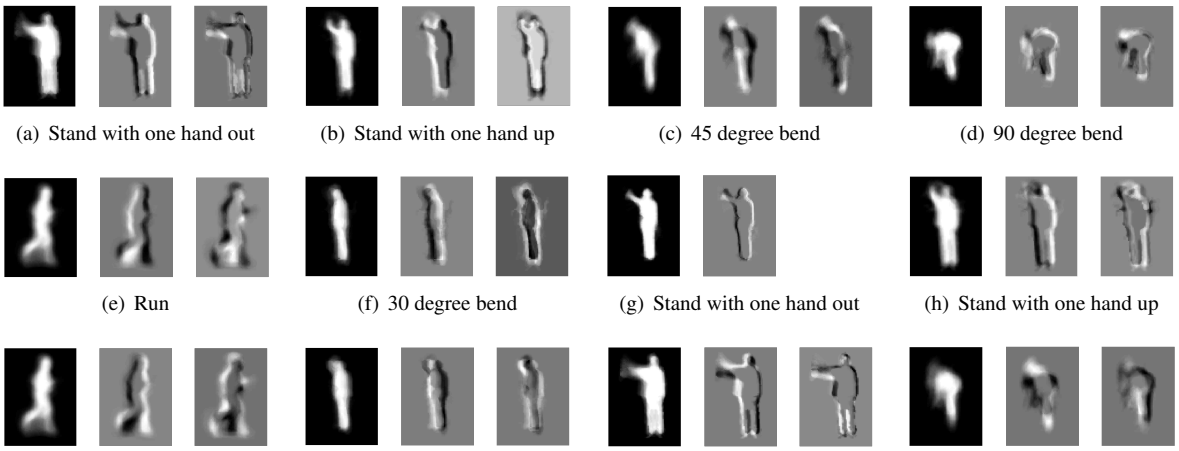
Algorithm 1: Hierarchical Sparse Subspace Clustering (the p-th level)

Input: A set of clusters $\{\mathbf{Y}_{p,\ell}\}_{\ell=1}^{L_p}$, their underlying subspace bases $\{\mathbf{D}_{p,\ell}\}_{\ell=1}^{L_p}$ and $\{B_{p,\ell}\}_{\ell=1}^{L_p}$, and parameters α, β and $d_{\min}$.

$M \leftarrow 0$.
for all $\ell = 1$ to L_p **do**
  if $B_{p,\ell} = 1$ **then**
    $\theta \leftarrow 0$.
    Split $\mathbf{Y}_{p,\ell}$ into $\mathbf{Z}_1$ and $\mathbf{Z}_2$ using SSC.
    $\forall k = 1, 2$, estimate $d_{\mathbf{Z}_k}$ and $\mathbf{D}_{\mathbf{Z}_k}$ of $\mathcal{S}_{\mathbf{Z}_k}$ using $\mathbf{Z}_k$.
    $\forall k = 1, 2$, compute E_k and $\widehat{E}_k$.
    If $(E_k - \widehat{E}_k)/E_k \geq \beta$, $\theta \leftarrow \theta + 1$.
    If $\theta \geq 1$ and $\min(d_{\mathbf{Z}_1}, d_{\mathbf{Z}_2}) \geq d_{\min}$
      $\forall k = 1, 2$, $\mathbf{Y}_{p+1,M+k} \leftarrow \mathbf{Z}_k$, $\mathbf{D}_{p+1,M+k} \leftarrow \mathbf{D}_{\mathbf{Z}_k}$, and $B_{p+1,M+k} \leftarrow 1$.
      $M \leftarrow M + 2$.
    Else $\mathbf{Y}_{p+1,M+1} \leftarrow \mathbf{Y}_{p,\ell}$, $\mathbf{D}_{p+1,M+1} \leftarrow \mathbf{D}_{p,\ell}$, $B_{p+1,M+1} \leftarrow 0$, and $M \leftarrow M + 1$.
  else
    $\mathbf{Y}_{p+1,M+1} \leftarrow \mathbf{Y}_{p,\ell}$, $\mathbf{D}_{p+1,M+1} \leftarrow \mathbf{D}_{p,\ell}$, $B_{p+1,M+1} \leftarrow 0$, and $M \leftarrow M + 1$.
  end if
end for
$L_{p+1} \leftarrow M$.

Output: A set of clusters $\{\mathbf{Y}_{p+1,\ell}\}_{\ell=1}^{L_{p+1}}$, subspace bases $\{\mathbf{D}_{p+1,\ell}\}_{\ell=1}^{L_{p+1}}$ and $\{B_{p+1,\ell}\}_{\ell=1}^{L_{p+1}}$.

To demonstrate the reason why HSSC outperforms flat SSC, we consider the following example. We perform HSSC with $P = 3$ to learn attributes for three actions: bend, run and one-hand wave in Weizmann dataset [6]. It has 6 subspaces as the leaf nodes, and the first three basis vectors of each leaf subspace are illustrated in Fig. 2 (a)-(f). We then apply SSC to learn 6 attributes and all those subspaces are depicted in Fig. 2 (g)-(l). It can be seen that the attributes learned using hierarchical SSC capture the variations and full range of movement within the bend action in a better way compared to SSC (as can be seen in (c), (d) (f) and (j), (l)). While the first subspace of SSC (seen in (g)) does not provide any additional information of one-hand wave action. We also present the three action video sequences for 9 subjects in Weizmann dataset as transition sequences of the learned attributes in Fig. 3. The run action is represented by a single subspace in both methods. As expected, the bend action is represented by more number of attributes in HSSC compared to SSC. In terms of subspace transitions, some frames of one-hand wave action are represented by attribute (f) in HSSC, which corresponds to

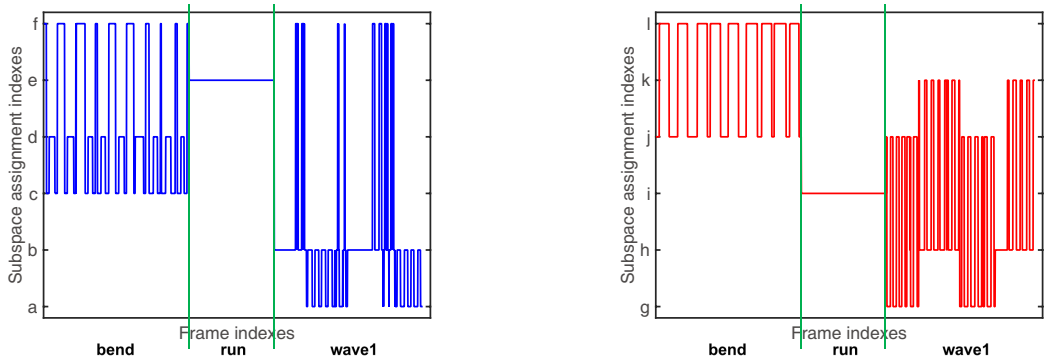


(a) Stand with one hand out (b) Stand with one hand up (c) 45 degree bend (d) 90 degree bend

(e) Run (f) 30 degree bend (g) Stand with one hand out (h) Stand with one hand up

(i) Run (j) 30 degree bend (k) Stand with one hand out (l) Bend

Figure 2. Visualization of subspaces (first three dimensions) learned using the frames of three actions: bend, run and one-hand wave. (a)-(f) represent the leaf subspaces learned by HSSC. (g)-(l) represent the subspaces which are learned using SSC.



(a) (b)

Figure 3. Subspace assignment result of the video frames from bend, run and one-hand wave actions using subspaces learned from (a) HSSC and (b) SSC.

the bend action attribute. However, in SSC, there are more frames in one-hand wave action which are represented by the attribute (j) (see Fig. 3(b)). Thus, the action recognition performance using HSSC will be better compared to SSC.

3.1. Complexity Analysis for Flat SSC and Hierarchical SSC

We first investigate the computational complexity of flat SSC. As proposed in [4], SSC mainly consists of three steps: learning the sparse coefficients $\widehat{\mathbf{A}}$ using ADMM, computing the normalized Laplacian matrix from $\widehat{\mathbf{A}}$, and K-means clustering (with L clusters) on the normalized Laplacian matrix. For the first two steps, the complexity is implementation dependent, but the worst case would be $O(N^3)$, where N is the number of samples in the training data $\mathbf{Y}$. The complexity of K-means clustering on the normalized Laplacian matrix is $O(N^2L)$. Therefore, by assuming $L \ll N$, the overall complexity of SSC is $O(N^3)$.

To analyze the computational complexity of hierarchical SSC, we assume that there are 2^p clusters at the p-th level ($p = 0, 1, \ldots, \log_2 L$). For each cluster at the p-th level ($0 \leq p \leq \log_2 L - 1$), we run SSC on each cluster to obtain two sub-clusters at the next level. We again use $N_{p,\ell}$ to denote number of signals that are assigned to the ℓ-th cluster at the p-th level. As discussed earlier, the computational complexity of applying SSC on these $N_{p,\ell}$ samples for two clusters will be $O(N_{p,\ell}^3)$. For the sake of exposition, we make another assumption that $N_{p,\ell} = N/2^p$ for all ℓ's. In such a case, the overall complexity at the p-th level is $O((N/2^p)^3 \times 2^p) = O(N^3/4^p)$. The sum of the complexity orders over all the levels gives us the overall complexity of hierarchical SSC as $\sum_{p=0}^{\log_2 L-1} O(N^3/4^p) = O(\frac{4}{3}N^3(1 - \frac{1}{L^2}))$. The computational complexity of HSSC is slightly more than that of flat SSC. However, the ad-

vantages of HSSC over flat SSC in learning better action attributes at different resolutions without any prior knowledge of the number of attributes significantly outweighs this slight increase in computational complexity.

3.2. Action Recognition Using Learned Subspaces

In this section, we describe the classification strategy to perform action recognition using the hierarchical UoS model learned by HSSC algorithm. We first learn the subspaces, which are the action attributes, by applying HSSC on the silhouette feature sequences of human actions in an unsupervised manner, where each silhouette frame (each data sample) is vectorized and normalized to unit ℓ_2 norm. We assume HSSC ends up with L_P leaf subspaces and the orthonormal bases of these subspaces can be represented by $\{\mathbf{D}_{P,\ell} \in \mathbb{R}^{m\times d_{P,\ell}}\}_{\ell=1}^{L_P}$.

Suppose there are V actions and R subjects in the training set. We use $\mathbf{\Phi}_{v,r} \in \mathbb{R}^{m\times s_{v,r}}$ to denote the video sequence of the r-th subject with the v-th action, where $s_{v,r}$ denotes the number of frames in the video. We assign every frame in one video sequence $\mathbf{\Phi}_{v,r}$ ($v \in \{1,\ldots,V\}, r \in \{1,\ldots,R\}$) to the "closest leaf subspace" and we use $\boldsymbol{\phi}_{v,r} \in \mathbb{R}^{s_{v,r}}$ to denote the vector which contains the resulting subspace assignment indexes. This vector represents the sequence of action attributes and the transitions involved in the human action video. All training video samples have subspace transition vectors $\boldsymbol{\phi}_{v,r}$'s. Then for a test video $\mathbf{\Psi} \in \mathbb{R}^{m\times s}$ with s denoting the number of frames in this video, we first perform subspace assignment for all the frames in $\mathbf{\Psi}$ and we use $\boldsymbol{\psi} \in \mathbb{R}^s$ to denote the resulting transition vector. Then we use a nearest neighbor classifier to perform action recognition, i.e., $\mathbf{\Psi}$ is declared to be in a particular action class v' for which the average of distances between the transition vector $\boldsymbol{\psi}$ and all the training transition vectors $\boldsymbol{\phi}_{v',r}$'s in the v'-th class is the smallest. Note that the video sequences, and hence the subspace assignment vectors, are of different lengths. In order to make the action recognition process temporal-scale invariant, we use Dynamic Time Warping (DTW) method [21] on the Grassmann manifold, described in Algorithm 2, where the element-wise distances used in here are the normalized subspace distances between the leaf subspaces. Mathematically speaking, for every pair of the subspaces $\mathcal{S}_{P,\ell}$ and $\mathcal{S}_{P,\widehat{\ell}}$, the normalized subspace distance between these two subspaces on the Grassmann manifold (in Step 3 of Algorithm 2) is defined as $d_u(\mathcal{S}_{P,\ell}, \mathcal{S}_{P,\widehat{\ell}}) = \sqrt{1 - \frac{\mathrm{tr}(\mathbf{D}_{P,\ell}^T \mathbf{D}_{P,\widehat{\ell}} \mathbf{D}_{P,\widehat{\ell}}^T \mathbf{D}_{P,\ell})}{\max(d_{P,\ell}, d_{P,\widehat{\ell}})}}$ [24], where $\mathrm{tr}(\cdot)$ denotes the trace operation.

4. Performance Evaluation

In this section, we report the experimental results obtained by applying the proposed HSSC approach on human action video datasets to learn the action attributes. Our first objective is to semantically interpret and label the learned action attributes from HSSC and to investigate the utility of the multi-resolution action attributes in semantic description of long action sequences. The secondary goal is to evaluate the effectiveness of these learned attributes in human action recognition and to compare the quantitative results to other UoS learning methods. In all the following experiments, we use the noisy variant of the optimization program (i.e., ADMM) of SSC and set $\lambda_z = \alpha_z/\mu_z$, where λ_z and μ_z are as defined in [4, (13) & (14)] and the parameter α_z

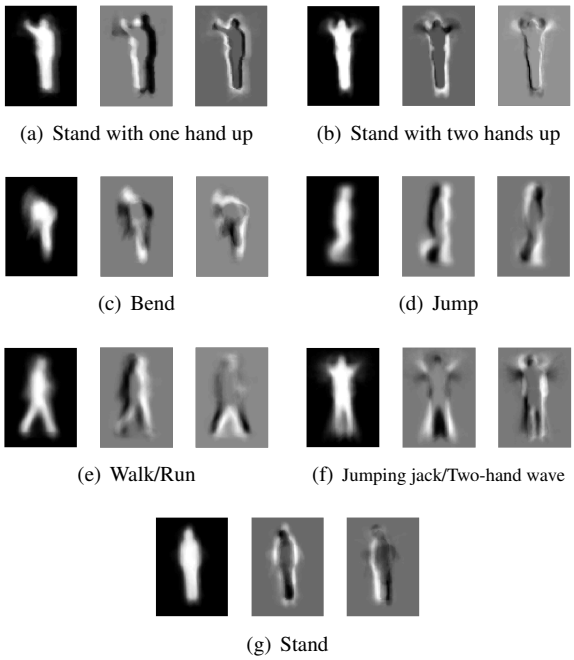
Algorithm 2: Dynamic Time Warping on the Grassmann Manifold

Input: Two subspace assignment sequences $\boldsymbol{\phi} \in \mathbb{R}^{s_1}$ and $\boldsymbol{\psi} \in \mathbb{R}^{s_2}$, leaf subspace bases $\{\mathbf{D}_{P,\ell}\}_{\ell=1}^{L_P}$ of $\mathcal{S}_{P,\ell}$'s.

Initialize: A matrix $\mathbf{E} \in \mathbb{R}^{(s_1+1)\times(s_2+1)}$ with $e(1,1) = 0$ and all other entries in the first row and column are ∞.

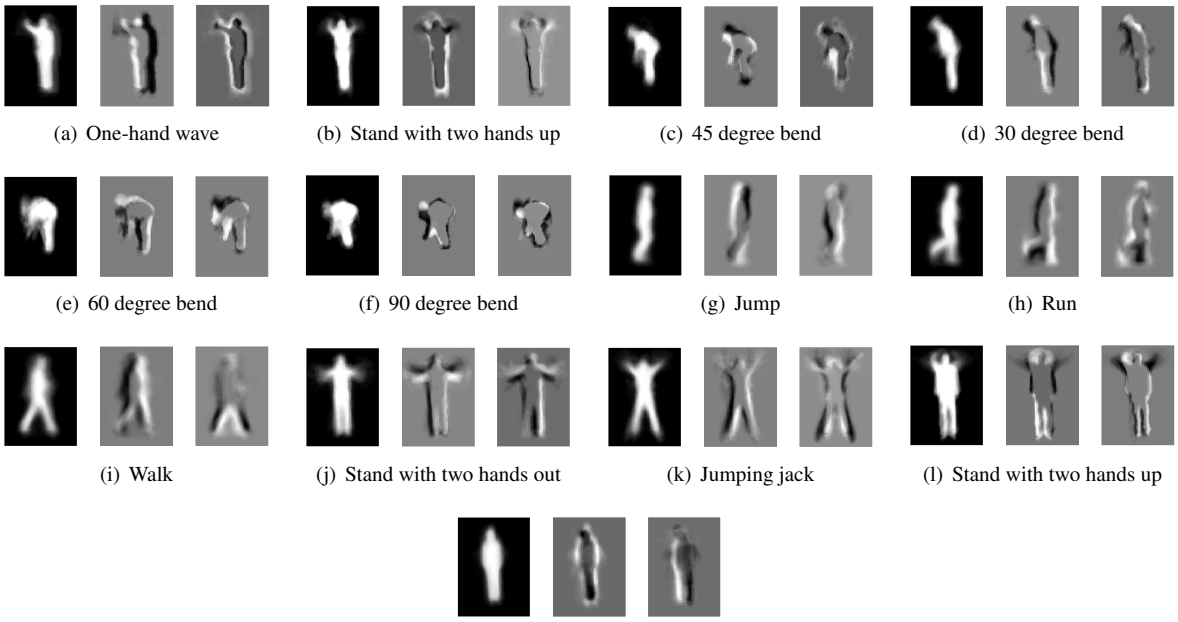
1: **for all** $i = 1$ to s_1 **do**
2: **for all** $j = 1$ to s_2 **do**
3: $c \leftarrow d_u(\mathcal{S}_{P,\boldsymbol{\phi}(i)}, \mathcal{S}_{P,\boldsymbol{\psi}(j)})$.
4: $e(i+1, j+1) \leftarrow$ $c + \min(e(i, j+1), e(i+1, j), e(i, j))$.
5: **end for**
6: **end for**

Output: The distance between $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ is $e(s_1+1, s_2+1)$.



(a) Stand with one hand up (b) Stand with two hands up (c) Bend (d) Jump (e) Walk/Run (f) Jumping jack/Two-hand wave (g) Stand

Figure 4. Visualization and interpretation of attributes at the 3rd level of HSSC for Weizmann dataset.



(a) One-hand wave (b) Stand with two hands up (c) 45 degree bend (d) 30 degree bend

(e) 60 degree bend (f) 90 degree bend (g) Jump (h) Run

(i) Walk (j) Stand with two hands out (k) Jumping jack (l) Stand with two hands up

(m) Stand

Figure 5. Visualization and interpretation of attributes at the bottom (5th) level of HSSC for Weizmann dataset.

Figure 6. Subspace transition of a long sequence using subspaces at the 5th level (top) and the 3rd level (bottom). The subspace assignment indexes in the top/bottom figure correspond to the attributes in Fig. 5 and Fig. 4, respectively.

varies in different experiments.

4.1. Semantic Labeling and Summarization

In this section, we visualize the learned attributes from HSSC at two different resolutions, give them semantic labels and use them for semantic summarization of multiple actions in a long video sequence. We apply HSSC on the Weizmann dataset with parameters $P = 5$, $\alpha = 0.9$, $\beta = 0.05$, $d_{\min} = 4$ and $\alpha_z = 20$. HSSC returns $L_P = 13$ leaf subspaces at the 5th level and 7 subspaces at the 3rd level. We show the first three dimensions of the orthonormal bases of those subspaces (attributes) here and give them interpretative (semantic) labels in Fig. 5 and Fig. 4, respectively. To demonstrate the semantic summarization of a long video sequence, we create a sequence by concatenating the bend and two-hand wave sequence of one subject and visualize the subspace transition of the frames in Fig. 6. We can interpret the actions using the attribute assignment

within the interval defined by green lines based on the corresponding labels in Fig. 5 and Fig. 4. At Level 5 (Fig. 6, top), human actions in the first half of the video sequence can be described as 30 degree bend followed by 60 degree bend, 90 degree bend, 60 degree bend again, and 30 degree bend, which can be interpreted as a full range bend action as done in Level 3 (Fig. 6, bottom). Next, at Level 5, the actions in the second half of the video sequence are Stand followed by two alternating attributes: Stand with two hands out and Stand with two hands up. The complete action can be described as a two-hand wave, which is precisely what is done at a lower resolution in Level 3. Therefore, we can say that the attributes generated at different levels of HSSC algorithm can be used for semantic summarization of video sequences at different resolutions.

4.2. Action Recognition: Evaluation on Different Datasets

In this section, we compare the performance of the proposed HSSC algorithm to flat SSC [4], RSSC [22], and TSC [7] with the number of clusters set (i) to be the same number of subspaces generated by HSSC at the bottom-most level (which is denoted by *Algorithm-*L_P) and (ii) to be the same as the number of actions (which is denoted by *Algorithm-*V). The parameter α_z for flat SSC is the same as the one for HSSC. In the case of RSSC, we set $\lambda = 1/\sqrt{d}$ as per [22], where d is the mean of the subspace dimensions returned by SSC-L_P/SSC-V. The tuning parameter q in TSC is set $q = \max(3, \lceil N/(L \times 20) \rceil)$, where L is equal to L_P and V for TSC-L_P and TSC-V, respectively.

We use three public datasets for this purpose: the Weizmann action dataset [6], the Keck gesture dataset [12], and the UT-Tower action dataset [3]. We evaluate all the subspace/attribute learning approaches based on a leave-one-subject-out experiment. To be specific, we pick all the videos of one subject (see Section 3.2) for testing at one time, while using all other videos as training samples. The Weizmann dataset consists of $V = 10$ different actions: walk, run, jump, gallop sideways, bend, one-hand wave, two-hands wave, jump in place, jumping jack, and skip. Each action is performed by nine subjects. The original resolution of the frames is 180×144. We align all the binary silhouette sequences and crop them into 87×63 frames, thereby the dimensionality of data is $m = 5481$. The hierarchical SSC is performed with parameters described in Section 4.1 and it returns $L_P = 13$ leaf subspaces for final attributes. The Keck gesture dataset was collected using a camera with 640×480 resolution. It consists of $V = 14$ different actions, including turn left, turn right, attention left, attention right, flap, stop left, stop right, stop both, attention both, start, go back, close distance, speed up, and come near. Each of these 14 actions is performed by three people. In each video sequence, the subject repeats the same action three times. Therefore the total number of video sequences in this dataset is $14 \times 3 \times 3 = 126$. We crop all the silhouette sequences to 380×280 resolution and downsample all the video frames by a factor of 4 in each dimension for computational purposes, with the resulting sequences being of size 95×70 and hence $m = 6650$. We perform hierarchical SSC with parameters $P = 6$, $\alpha = 0.98$, $\beta = 0.02$, $d_{\min} = 3$ and $\alpha_z = 100$, in which case it returns $L_P = 18$ leaf subspaces at the bottom-most level. The UT-Tower action dataset contains a collection of 108 low resolution videos and there exist $V = 9$ different actions in this dataset, including pointing, standing, digging, walking, carrying, running, wave1, wave2, and jumping. Each action is performed twice by 6 individuals, which results in a total of 12 video sequences per action. We use the bounding boxes and foreground masks provided by the authors of [3] to extract silhouettes. All the silhouette sequences are of size 49×61 ($m = 2989$). We perform hierarchical SSC with parameters $P = 6$, $\alpha = 0.95$, $\beta = 0.04$, $d_{\min} = 4$ and $\alpha_z = 150$, obtaining $L_P = 11$ final subspaces. The recognition results of different algorithms for the three datasets are shown in Table 1. We can see that by representing the human actions using the attributes learned by HSSC, we are able to recognize the actions at a superior rate compared to other techniques.

Table 1. Recognition results (%) on different datasets

Data ↓ Method →	HSSC	SSC-L_P	SSC-V	RSSC-L_P	RSSC-V	TSC-L_P	TSC-V
Weizmann [6]	**91.11**	83.33	76.67	57.78	65.56	87.78	83.33
Keck [12]	**78.57**	57.94	67.46	34.13	37.30	53.17	53.97
UT-Tower [3]	**76.85**	75.93	73.15	60.19	63.89	65.74	62.04

5. Conclusions

An advantage of the proposed Hierarchical Sparse Subspace Clustering (HSSC) is that it does not need the number of clusters to be specified and does not require labeled training data to learn human action attributes, while avoiding generation of trivial attributes. HSSC also provides the action attributes at multiple resolutions, which can be later visualized, labeled and used for human action summarization in long video sequences. Empirical results on real video datasets show the effectiveness of our approach and its superiority over the state of the art for human action recognition as well as its utility in semantic interpretation of human activities at multiple resolutions.

Acknowledgement

This work was supported in part by the NSF under grants CCF-1218942 and CCF-1453073, by the Army Research Office under grant W911NF-14-1-0295, and by an Army Research Lab Robotics CTA subaward.

References

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

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

[3] C.-C. Chen, M. S. Ryoo, and J. K. Aggarwal. UT-Tower dataset: Aerial view activity classification challenge, 2010, `http://cvrc.ece.utexas.edu/SDHA2010/Aerial_View_Activity.html`.

[4] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2765–2781, 2013.

[5] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1–8, 2008.

[6] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2247–2253, 2007.

[7] R. Heckel and H. Bölcskei. Robust subspace clustering via thresholding. *arXiv:1307.4891*, 2013.

[8] Z. Ji, Y. Su, R. Qian, and J. Ma. Surveillance video summarization based on moving object detection and trajectory extraction. In *Proc. IEEE Int. Conf. Signal Process. Syst. (ICSPS)*, pages 250–253, 2010.

[9] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah. High-level event recognition in unconstrained videos. *Int. J. Multimed. Inf. Retr.*, 2(2):73–101, 2013.

[10] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2698–2705, 2013.

[11] E. Kim, S. Helal, and D. Cook. Human activity recognition and pattern discovery. *IEEE Pervasive Comput.*, 9(1):48–53, 2010.

[12] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 444–451, 2009.

[13] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3337–3344, 2011.

[14] J. Liu and M. Shah. Learning human actions via information maximization. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 1–8, 2008.

[15] J. Liu, Y. Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 461–468, 2009.

[16] D. Moore and I. Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *Proc. Nat. Conf. Artif. Intell.*, pages 770–776, 2002.

[17] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 849–856, 2001.

[18] W. Niu, J. Long, D. Han, and Y.-F. Wang. Human activity detection and recognition for video surveillance. In *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, pages 719–722, 2004.

[19] D. Ramanan and D. A. Forsyth. Automatic annotation of everyday movements. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pages 1547–1554, 2003.

[20] M. S. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me? In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 2730–2737, 2013.

[21] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust., Speech, Signal Process.*, 26(1):43–49, 1978.

[22] M. Soltanolkotabi, E. Elhamifar, and E. J. Candès. Robust subspace clustering. *Ann. Stat.*, 42(2):669–699, 2014.

[23] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 586–591, 1991.

[24] L. Wang, X. Wang, and J. Feng. Subspace distance analysis with application to adaptive Bayesian algorithm for face recognition. *Pattern Recognit.*, 39(3):456–464, 2006.