Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning

Saad Ali, *Student Member, IEEE,* Mubarak Shah, *Fellow, IEEE,*

Abstract—We propose a set of kinematic features that are derived from the optical flow, for human action recognition in videos. The set of kinematic features include divergence, vorticity, symmetric and anti-symmetric flow fields, second and third principal invariants of flow gradient and rate of strain tensor, and third principal invariant of rate of rotation tensor. Each kinematic feature, when computed from the optical flow of a sequence of images, gives rise to a spatio-temporal pattern. It is then assumed that the representative dynamics of the optical flow are captured by these spatio-temporal patterns in the form of *dominant kinematic trends* or *kinematic modes*. These kinematic modes are computed by performing Principal Component Analysis (PCA) on the spatio-temporal volumes of the kinematic features. For classification, we propose the use of multiple instance learning (MIL), in which each action video is represented by a bag of kinematic modes. Each video is then embedded into a kinematic mode-based feature space and the coordinates of the video in that space are used for classification using the nearest neighbor algorithm. The qualitative and quantitative results are reported on the benchmark data sets.

Index Terms—Action Recognition, Motion, Video Analysis, Principal Component Analysis, Kinematic Features.

1 Introduction

The recognition of human-induced actions in videos is considered an important problem in the field of computer vision. This is due, in part, to the large number of potential applications of action recognition in areas of visual surveillance, video retrieval, sports video analysis, human-computer interfaces, and smart rooms. A popular approach pursued by vision researchers for action recognition is to utilize the *motion* of the human actor, where the motion is quantified in terms of the optical flow computed from the sequence of images depicting the action ([17], [18], [28], [29]).

Since motion is an important source of information for classifying human actions, in this paper we have described a set of kinematic features that are derived from the optical flow for representing complex human actions in videos. The idea behind the introduction of kinematic features is to convert optical flow information into a more discriminative representation that can improve the motion-based action classification. The proposed kinematic features are: divergence, vorticity, symmetric and anti-symmetric optical flow fields, second and third principal invariants of flow gradient and rate of strain tensor, and third principal invariant of rate of rotation tensor. Each feature was selected to capture a different aspect of optical flow. For instance, divergence delineates the regions of optical flow that are undergoing expansion due to the movement of different limbs of the human body, while the vorticity feature emphasizes regions of optical flow that are undergoing circular motion.

Each kinematic feature, when computed from the optical flow of a sequence of images, gives rise to a spatio-temporal pattern. In order to reduce this three-dimensional information into a more manageable two-dimensional form, it is assumed that the representative dynamics of the optical flow are captured by these spatio-temporal patterns in the form of *dominant kinematic trends* or *kinematic modes*. These kinematic modes are computed by performing Principal Component Analysis (PCA) on the spatio-temporal volumes of the kinematic features. Traditionally, the PCA is performed directly on the optical flow data, i.e., on the u and v components of the optical flow which provides only a limited description of the *flow dynamics*. However, we capture the dominant dynamics of the optical flow by performing PCA directly on the kinematic features that are extracted from the optical flow, rather than on the optical flow itself.

Next, we used the identified kinematic modes of each kinematic feature for the classification of human actions. To do this, we used the multiple instance learning (MIL) approach proposed in [2]. The idea of MIL is to represent each action video as a collection or a "bag" of kinematic modes in which each kinematic mode is referred to as an *instance* representing that video. In machine learning, MIL is proposed as a variation of supervised learning for problems with incomplete knowledge about labels of

- *S. Ali is a post-doctoral research fellow at the Robotics Institute of Carnegie Mellon University, Pittsburgh, PA, 15213.E-mail: saada@cs.cmu.edu*
- *M. Shah is with the School of Computer Science, University of Central Florida, 4000 Central Florida Blvd., Orlando, FL 32816. E-mail: shah@eecs.ucf.edu*

Manuscript received October 8, 2007; revised May 15, 2008; accepted Nov 13, 2008

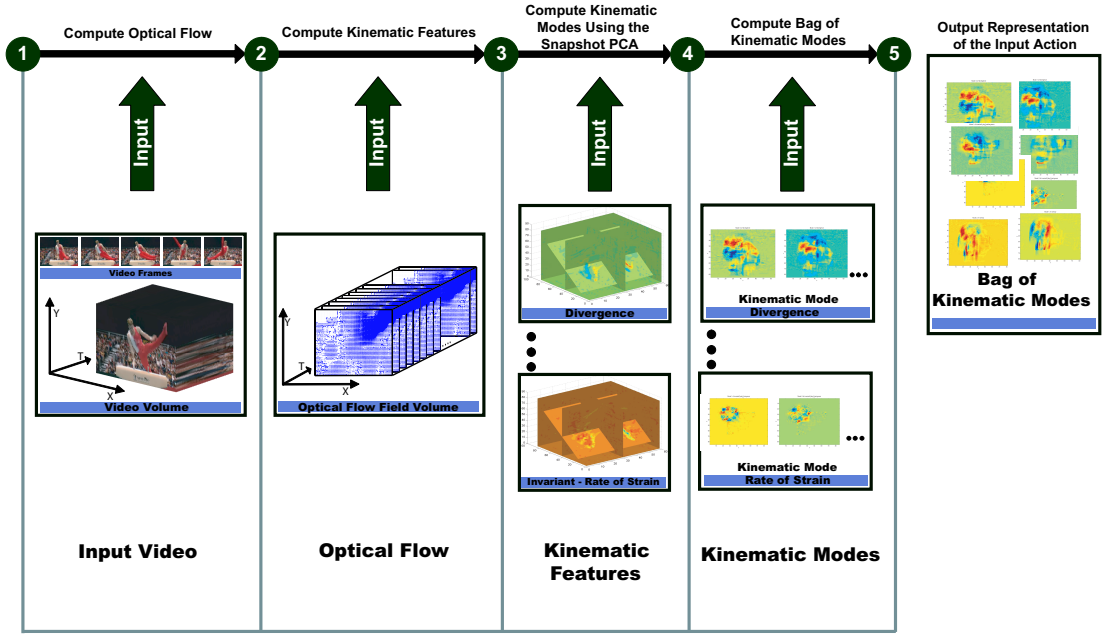

Fig. 1. Description of the process of representing a video in terms of modes of kinematic features. 1) A video containing an action is the input to the first step, which computes the optical flow between consecutive frames of the video and produces a stack of optical flow fields. 2) This stack of optical flow fields is the input to the second step, which computes the kinematic features and produces a separate, spatio-temporal volume for each feature. 3) The third step takes the volume of each kinematic feature as an input, performs the PCA, and produces the kinematic modes. 4) Finally, the video is represented as a bag of kinematic modes pooled from of all the kinematic features. This bag is then used to conduct the feature space embedding of the video.

training examples. In supervised learning, every training instance is assigned a discrete or real-valued label. In comparison, in MIL the labels are assigned only to bags of instances. In the binary case, a bag is labeled positive if at least one instance in that bag is positive, and the bag is labeled negative if all the instances in it are negative. There are no labels on the individual instances. The goal of MIL is to classify unseen bags or instances by using the labeled bags as the training data [2]. We preferred to use MIL because it provides flexibility in picking the number of kinematic modes used to represent the action. This is important because a complex action may require more kinematic modes to represent its dynamics than a simple action.

1.1 Algorithmic Overview

We provide an algorithmic overview of the proposed action recognition methodology in this section. The aim is to clarify the algorithm by breaking it down into logical blocks. A pictorial visualization of these logical blocks is provided in Figure 1.

Given a video containing an action, the steps involved in creating the corresponding bag of kinematic modes are: 1) computation of optical flow between consecutive pairs of frames of the video resulting in a three-dimensional stack of optical flow fields, where the third dimension is time; 2) computation of the above-mentioned kinematic features from this stack of optical flow fields, which results in a spatio-temporal volume for each kinematic feature; 3) computation of the dominant kinematic modes of each kinematic feature by performing PCA. The PCA is performed separately on each volume, where the temporal correlation between the slices of the volume is used to create a temporal correlation matrix or kernel matrix; 4) computation of eigen vectors of the kernel matrix. The eigen vectors correspond to the dominant modes of the optical flow in terms of the dynamics, rather than energy, and are referred to as "kinematic modes;" and 5) representation of the video as a bag of kinematic modes by pooling the kinematic modes from all the kinematic features. This bag of kinematic modes is used later to perform the feature space embedding of each video. (See Section 5 and Figure 7 for further details.) The process is repeated for all the videos in the data set.

The contributions of this work are:

- A set of kinematic features that can be used to extract different aspects of motion dynamics present in the optical flow. Collectively, these features have not been used in the context of action classification to the best of our knowledge.
- Computation of the dominant PCA basis in terms of the *dynamics* of the optical flow field instead of the optical flow field itself.
- A multiple-instance learning for the classification of action videos.

The rest of the paper is organized as follows. Section 2 covers the literature in the area of human action and activity analysis. Section 3 introduces the kinematic features. Section 4 explains the computation of kinematic modes. The MIL-based classification algorithm is presented in Section 5. The experimental results are discussed in Section 6.

2 Related Work

Human action and activity recognition is an important area of research in the field of computer vision. A comprehensive review of the research in this area has been presented in a number of survey papers ([13], [14], [15]). In this section, we have limited our discussion to some of the most influential and relevant parts of the literature.

In general, approaches for human action and activity analysis can be categorized on the basis of *representation*. Some leading representations include learned geometrical models of human body parts, space-time pattern templates, appearance or region features, shape or form features, interest-point-based representations, and motion/optical flow patterns. In next few subsections, we discuss these representations and present a brief summary of the associated work.

2.1 Appearance-based Representations

In the past, representations based on appearance features were popular. The general methodology was to learn the appearance model of the human body or hand and match it explicitly to images in a target video sequence for action and gesture recognition ([11], [12], [21]). The temporal aspects of an action or gesture were handled by training Hidden Markov Models (HMM) and their different variants. The appearance-based approach works well for gesture recognition, since the appearance of the human hand remains relatively consistent from one actor to another. However, actions involving the whole human body are difficult to handle due to changes in the clothing from one actor to another. Recent work by Jiang *et al.* ([20]) also used appearance-based representation for action recognition in images by searching for static postures using the appearance of the whole human body or parts of the body.

2.2 Shape-based Representations

The popular shape-based representations include edges ([19]) and silhouettes of the human body ([22]). The fundamental idea behind shape-based representation is that an action consists of a series of poses that are detectable from a single frame. Each pose can be encoded using the shape features, and single frame recognition can be extended to multiple frames for robust action recognition. Silhouette-based representation was extended recently to characterize the outline of an actor's body through space and time ([3], [4]). This is done by stacking the individual silhouettes detected in each frame, giving rise to a three-dimensional volume. Yilmaz *et al.* ([3]) used the surface properties of this volume for action recognition, while Blank *et al.* ([4]) used the solution of the Poisson equation to extract space-time features of the volumes. These approaches have demonstrated robust performance on a number of actions. Note that work by [3] and [4] can also be categorized under a volume-based representation.

2.3 Volume-based Representations

The approaches based on volumetric analysis of video for action recognition include [3], [4], [5], and [6]. Ke *et al.* ([5]) extended the two-dimensional Haar features to three dimensions and learned a cascade of boosted classifiers, while Shechtman *et al.* ([6]) employed a three-dimensional correlation scheme to match the actions in a video volume. Mahmood *et al.* ([23]) also used volume representation for action recognition. Recently, *Scovanner et al.* [25] and *Liu et al.* [24] used quantized vocabularies of volumetric features that are computed using the three-dimensional SIFT (Scale Invariant Feature Transform) descriptor [26][25]. One benefit of the volume-based approach is that there is no need to build complex models of body configuration and kinematics, and recognition can be performed directly from the raw video.

2.4 Interest-Point-based Representations

Another important direction of research which has gained a lot of interest recently is the use of space-time interest points and their trajectories for action and activity analysis. Works by Laptev *et al.* ([33][34]), Oikonomopoulousm *et al.* ([35], [36]), Niebles *et al.* ([9]), Schuldt *et al.* ([37]), and Dollar *et al.* ([10]) belong to this class. The main strength of this representation is the robustness to occlusion, since there is no need to track or detect the whole human body.

2.5 Optical Flow-based Representations

Features based on motion information and optical flow, which are more relevant to our work, have been used by a number of researchers ([32], [28], [16], [17], [29], [30]). For instance, Bobick *et al.* [7] introduced motion energy image (MEI) as a way of describing the cumulative spatial distribution of motion energy in a given sequence. This description of motion is then matched against the stored models of the known actions. The MEI descriptor was later augmented by motion history image (MHI) in [8], where each pixel intensity in MHI is described as a function of motion energy. Recently, Weinland *et al.* ([27]) extended this representation to handle different viewpoints. In optical flow-based approaches ([28], [31], [17], [18], [30]), the idea is to directly use the optical flow to derive a representation which can be used for recognition. Little *et al.* ([28], [31]) used spatial distribution of the magnitude of the optical flow to derive model-free

features, while Ju *et al.* ([16]), Yacoob *et al.* ([17]), and Black *et al.* ([18]) proposed PCA-based analysis of optical flow for facial motion and expression recognition. Also Arbel *et al.* ([30]) performed PCA on optical flows that were generated by moving around a target object on a view sphere. The resulting representation was used for object recognition.

In [29], Hoey *et al.* used a Zernike polynomial basis to represent optical flow. The Zernike polynomial basis forms a complete orthogonal basis over a unit circle and can be used as a compact representation of optical flow. However, there is an inherent limitation in the precision of the computation of the Zernike polynomial because it is necessary to map the data to a circular domain. In addition, the Zernike polynomial forms an infinite basis, therefore precision is affected by the number of bases one selects to represent the data. Furthermore, the Zernike polynomial decomposes the optical flow and not the kinematics of the optical flow, which is one of the major difference between our work and that of Hoey *et al.* [29]. The features proposed in this work first try to capture the kinematics of the optical flow and then are decomposed into dominant modes using PCA. Therefore, the nature of the information captured by the work of [29] is different from ours. Additionally, in their method coefficient of Zernike polynomials were computed separately for each optical flow field of the video. These coefficient were tied together by training a HMM to handle the temporal aspect of an action. However, in our method we are computing only one set of kinematic basis per feature per video. These basis vectors directly capture the dominant temporal aspects of the action without any explicit subsequent temporal modeling using HMM.

In summary, our proposed framework is different from the optical flow-based representations in three important ways. First, we propose a new set of kinematic features for capturing the dynamics information hidden within a flow field. Second, we capture the kinematic modes of optical flow using the PCA, thereby enriching the traditional approaches of [17] and [18] which capture only the energetic modes of the optical flow. Third, we compute dominant kinematic modes present within the 3D volume of a kinematic feature by employing a temporal correlation between slices of the volume, while the works of [17] and [18] require multiple instances of the volume for the computation of optical flow modes. In addition, we perform classification by treating each action as a bag of kinematic modes in an MIL framework.

In the next section, we present details of proposed kinematic features and elaborate on their computation from the optical flow.

3 Kinematic Features

The proposed set of optical flow features is based on kinematics. The term "kinematics" emphasizes that these features are independent of forces acting on the object or mass of the object and only capture motion information. This property can be useful for recognizing action as it makes the representation independent of the physical features of the subject performing the action.

In order to compute the kinematic features, we start by computing the optical flow of a given video using a block-based correlation algorithm. Let $U(\mathbf{x}, t_i)$ represent the flow vector (u, v) at pixel location $\mathbf{x} = (x, y)$ at time t_i. It is computed by selecting a square interrogation block centered at the same pixel location in two consecutive frames F_{t_i} and $F_{t_{i+1}}$ of the sequence. Pixel values in both blocks are mean normalized and a correlation surface is obtained by performing cross correlation in the frequency domain. Peaks are located on the correlation surface and are used to compute the displacement of the pixel. The process is repeated for all possible blocks in the image. As a post-processing step, local outliers are removed by applying adaptive local median filtering and removed vectors are filled by interpolation of the neighboring flow vectors. The size of the block employed in our experiments is 16×16 pixels. The process is repeated for all frames, i.e. for $t_i, i = 1, \ldots, M$, to generate a stack of optical flows for the video. The optical flow fields computed by this algorithm for two different actions are shown in Figure 2. Note that this approach has been used to analyze fluid flows obtained by particle image velocimetry [39].

Now we describe the proposed kinematic features and the steps involved in their computation. We use the symbol $\mathcal{F} = (f^1, f^2, \ldots, f^p)$ to represent the set of kinematic features, where superscript p is the index of the kinematic feature. The symbol f^j, without any reference to the location and time, refers to the spatio-temporal volume of j-th kinematic feature. $f^j(\mathbf{x}, t_i)$ is used to refer to the value of the j-th kinematic feature at pixel location $\mathbf{x}$ at frame t_i. When referring to the kinematic features of a specific video, we use symbol f_c^j, where subscript c is the index of the video.

3.1 Divergence

Divergence of a flow field is a scalar quantity which is defined at a point $(\mathbf{x}, t_i)$ in space and time as:

$$f^1(\mathbf{x}, t_i) = \frac{\partial u(\mathbf{x}, t_i)}{\partial x} + \frac{\partial v(\mathbf{x}, t_i)}{\partial y}, \tag{1}$$

where $\frac{\partial u(\mathbf{x}, t_i)}{\partial x}$ and $\frac{\partial v(\mathbf{x}, t_i)}{\partial y}$, respectively, are the partial derivatives of u and v components of the optical flow with respect to the x and y direction at time t_i.

The physical significance of the divergence stems from the fact that it captures the amount of local expansion taking place in the fluid. This type of feature can be important for discriminating between types of motions which involve independent motion of different body parts. For instance, in "hand-waving" action, only one part of the body is involved, while in "bending" action the complete upper body plays a role. These two motions

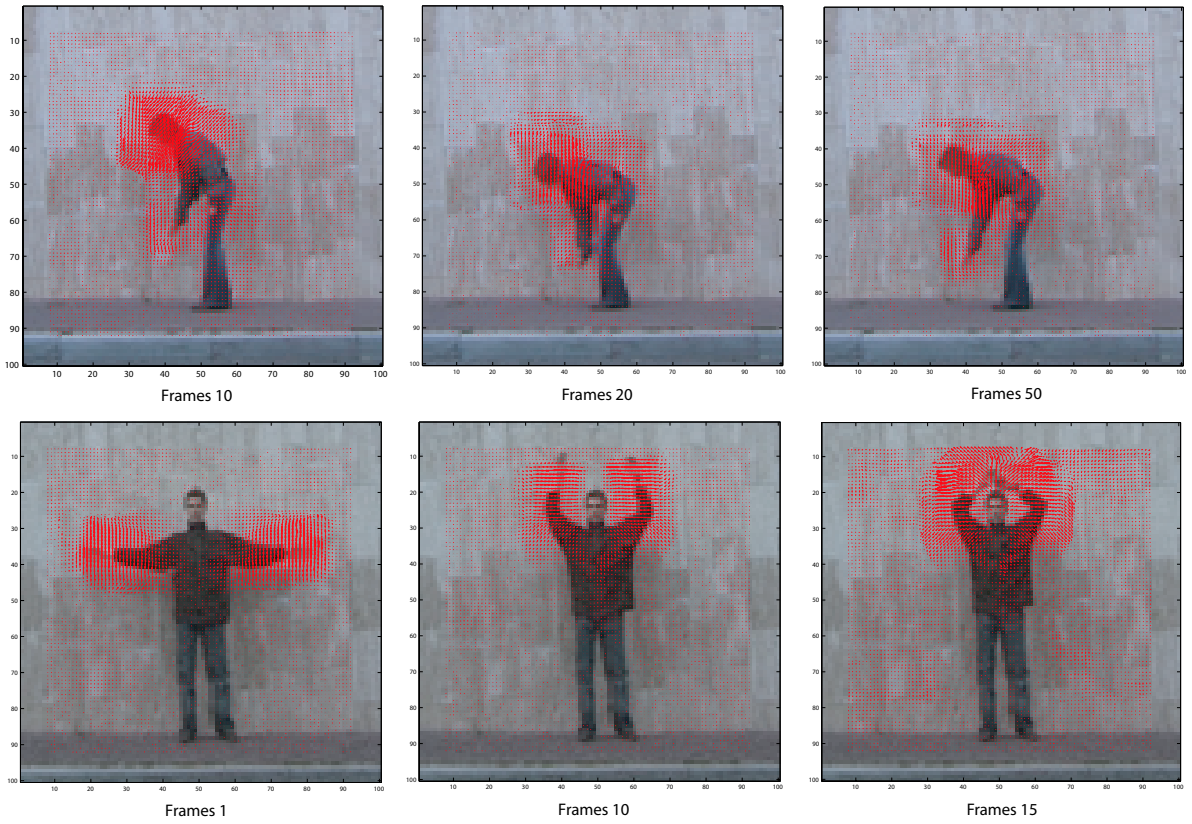

Fig. 2. Top: Optical flow of bend action at frames 10, 20, and 50. Bottom: Optical flow of hand-waving action at frames 1, 10, and 15.

will induce different types of expansion in the flow field. In addition, from fluid dynamics we know that the divergence of any incompressible fluid is zero, which implies existence of low energy in divergence fields of actions that do not involve independent motion of body parts. For instance, a "vertical jump" action, in which the whole body is expected to move up without any independent motion of arms or legs, will induce a low divergence in the corresponding optical flow. Figure 3 shows an example of the divergence field for the "bend" action at times t_{20}, t_{30} and t_{45}. It can be observed from the figure that the divergence has high values around the contours of the actor performing the action.

3.2 Vorticity

Vorticity is the measure of local spin around the axis perpendicular to the plane of the flow field. Vorticity is a more general concept than rotation/curl, although both are inter-related. It can also be defined as circulation per unit area. It is computed at a point $(\mathbf{x}, t_i)$ as follows:

$$f^2(\mathbf{x}, t_i) = \frac{\partial v(\mathbf{x}, t_i)}{\partial x} - \frac{\partial u(\mathbf{x}, t_i)}{\partial y}. \quad (2)$$

Curl ω is related to vorticity through the relation $\omega = \frac{f^2}{2}$.

Vorticity can also be used as a measure of the rigidity in the flow. Therefore, it is useful for distinguishing between actions that involve articulated motion and ones that do not. In addition, it is useful for highlighting dynamics in the flow field resulting from local circular motion of the human body or part of the body. The "bend" action is a good example of this type of motion, where circular motion of the body is around the perpendicular axis passing through the torso. This phenomenon is observable in Figure 3, where higher values are correlated with the central region of the body, as compared to the divergence, where higher values are around the contour of the body. Note that for irrotational flows $f^2(\mathbf{x}, t_i) = 0$.

3.3 Symmetric and Asymmetric Fields

Symmetric and asymmetric fields capture the dynamics that emphasize the symmetry or asymmetry of a human action around a diagonal axis. This can be an important feature for action classification as some actions represent symmetrical motion of body parts, while others do not. An interesting example is the action of raising the right hand compared to raising the left hand. If the diagonal axis is drawn from the top-left to the bottom-right of the image, then symmetric and asymmetric kinematics can help us in distinguishing between these two actions. Note that divergence and vorticity will not be able to capture this type of the symmetric and asymmetric dynamics hidden in the input flow field.

In our formulation, symmetric and asymmetric kinematics of u and v components of the flow field are

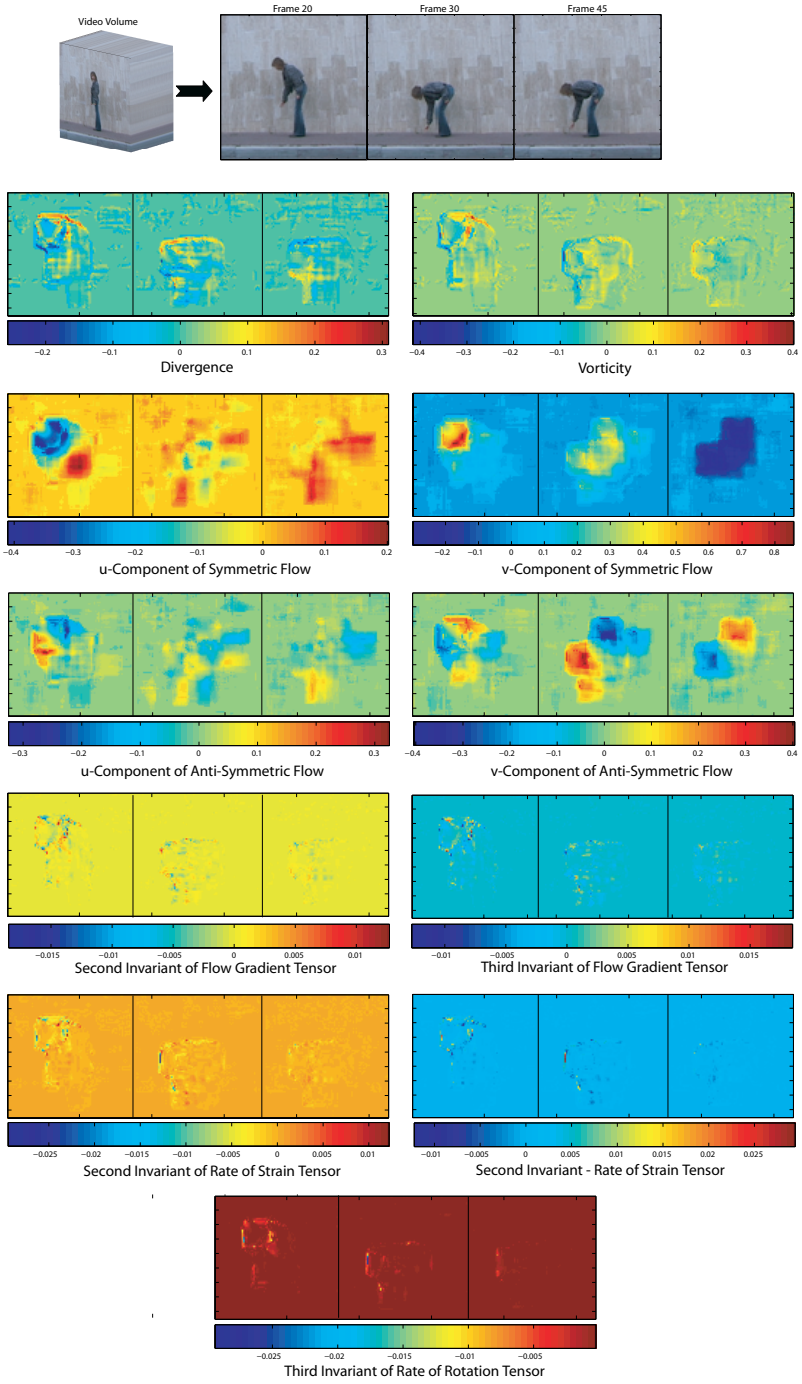

Fig. 3. Top Row: The video volume of the "bend" action and three frames from the volume. For each frame, we show the visualization of computed kinematic features in the following rows.

computed separately, resulting in four kinematic measurements in total. The symmetric kinematic features are computed as follows:

$$f^3(t_i) = u(t_i) + u(t_i)^*, \tag{3}$$

$$f^4(t_i) = v(t_i) + v(t_i)^*, \tag{4}$$

where $u(t_i)$ and $v(t_i)$ represent the u and v components of the optical flow at time t_i. Here we have removed the dependence on spatial location $\mathbf{x}$ to show that these features can easily be computed by using the matrix transpose operation. The symbol '$*$' denotes the transpose operation. The asymmetric kinematic features are computed as follows:

$$f^5(t_i) = u(t_i) - u(t_i)^*, \tag{5}$$

$$f^6(t_i) = v(t_i) - v(t_i)^*. \tag{6}$$

Figure 3 shows an example of these features computed for the "bend" action. It should be noted that if the kinematics are symmetric or asymmetric, the absolute values around the diagonal axis will be high; otherwise

the absolute values will be high on one side and low on the other. In Figure 3, features f^3 and f^4 have the high values around the diagonal, thus emphasizing the symmetry, while features f^5 and f^6 have high values on one side and low values on the other. Note that this feature imposes a restriction on the structure of the flow field, i.e., it has to be a square matrix. However, we can easily overcome this problem by resizing the images before computing the optical flow. Another solution is to change the reference axis to vertical or horizontal.

3.4 Gradient Tensor Features

Small-scale structures present in a flow field are referred to as "eddies", a term taken from the literature on turbulence phenomena in fluid flows. These structures are characterized by large velocity gradients ([38]). In a flow field representing human actions, these small-scale structures arise due to small-scale motion of different limbs. In this section, we present a few kinematic features that take these local structures into account, and are based on optical flow gradients which represent a better measure of the local structure than the optical flow itself. For this purpose, we start by computing the optical flow gradient tensor as follows:

$$\nabla U(\mathbf{x}, t_i) = \begin{pmatrix} \frac{\partial u(\mathbf{x},t_i)}{\partial x} & \frac{\partial u(\mathbf{x},t_i)}{\partial y} \\ \frac{\partial v(\mathbf{x},t_i)}{\partial x} & \frac{\partial v(\mathbf{x},t_i)}{\partial y} \end{pmatrix}. \tag{7}$$

The gradient tensor can be considered as a 2×2 matrix at each spatial location $\mathbf{x}$.

The kinematic features derived from the gradient tensor are based on the concept of tensor invariants. Tensor invariants, under full transformation group, are scalar quantities which are the combination of the tensor elements, and remain unchanged no matter which coordinate system they are referenced in. In simple terms, this invariant quantity can be obtained by summing, or contracting, all the indices of a tensor in pairs, so that no free indices remain. For example, a vector is a tensor of rank 1 whose projections on different axes change with the rotation of the coordinate frame; however its length is invariant since it remains unchanged regardless of the rotation of the coordinate system. An invariant property is particularly useful for human actions, since many interesting aspects of a flow field can be described in terms of features that are coordinate invariant.

The three principal invariants of the gradient tensor can be written as ([38]):

$$\begin{aligned} P(\mathbf{x}, t_i) &= -\text{trace}(\nabla U(\mathbf{x}, t_i)), \\ Q(\mathbf{x}, t_i) &= \frac{1}{2}(P^2 - \text{trace}(\nabla U(\mathbf{x}, t_i)^2)), \\ R(\mathbf{x}, t_i) &= -\det(\nabla U(\mathbf{x}, t_i)). \end{aligned}$$

The first invariant P is the trace of the gradient tensor, which is equal to the divergence. Therefore, we utilize only the second and third invariants as they are providing us with new information, i.e., $f^7 = Q$ and $f^8 = R$. Figure 3 provides an illustration of these features for the "bend" action.

3.5 Rate of Strain and Spin Tensor Features

The rate of strain tensor S and rate of spin tensors O can be obtained by decomposing the flow gradient tensor as follows:

$$\begin{aligned} S(\mathbf{x}, t_i) &= \frac{1}{2}(\nabla U(\mathbf{x}, t_i) + \nabla U(\mathbf{x}, t_i)^*), \\ O(\mathbf{x}, t_i) &= \frac{1}{2}(\nabla U(\mathbf{x}, t_i) - \nabla U(\mathbf{x}, t_i)^*), \end{aligned}$$

where $*$ is the matrix transpose operation. These two tensors are often used as a measure of the deformability which occurs due to the presence of the gradients in the flow field. Another way to look at it is that they represent deviations from the rigid body motion. The kinematic features from these tensors are also encoded in terms of their principal invariants, similar to the flow gradient tensor.

We use the second and third principal invariants of S and only the third invariant of O. The is because the first invariant of S is equal to the first invariant of ∇U, and therefore equivalent to the divergence. Similarly, the second invariant of O is the magnitude of the vorticity squared, which we have already taken into account. Using subscript s and o for invariants of these tensors, we have $f^9 = Q_s$, $f^{10} = R_s$ and $f^{11} = R_o$. An example output of these features for the "bend" action is shown in Figure 3.

4 Kinematic Modes

As mentioned in the introduction, the PCA of optical flow is limited in its description because it only captures the energy-containing structures of the flow field. However, to get a complete description of an action, it is important to capture the evolution of the dynamics of the flow field in space and time, and not just the evolution of the energy. This is critical because information about an action often lies at the location of moving limbs and their boundaries. Such locations may not be the most energetic parts of the optical flow.

To provide an intuitive insight into this idea, we show the first three dominant modes of the optical flow of the "two hands wave" action in Figure 4 (top row). These modes are obtained by performing PCA directly on the two-components ((u, v)) of optical flow. It can be observed that as the number of modes increases, smaller energy-containing scales are exhibited. However, these modes do not provide information about the dynamics (e.g., rotation, expansion, symmetry, etc.) of the underlying optical flow. The second row in Figure 4 shows the first three modes of divergence. It is clear that these modes are revealing characteristics of the optical flow not visible in the energy containing structures of the first row. In addition, divergence with fewer modes is able to uncover a much more complex and possibly discriminating pattern.

Therefore, in order to capture the dynamics information of the optical flow, we propose computing a set of orthogonal basis in terms of its dynamics, instead

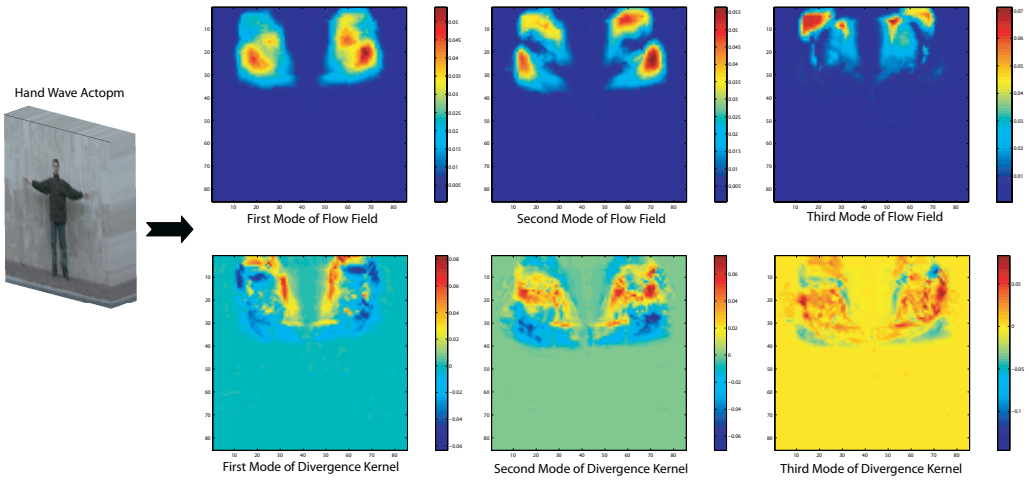

Fig. 4. Top Row: First three modes of the optical flow field for the "two hands wave" action. Bottom Row: First three modes of the divergence feature for the same action. It can be observed that the information extracted by these two sets of modes is very different. The energy-containing regions of the optical flow in the top row do not reveal the finer characteristics of the optical flow visible in the bottom row.

of its energy content. For this purpose, we perform PCA separately on spatio-temporal volumes of each kinematic feature described previously. Here we employ a computationally efficient way of obtaining the orthogonal basis using the snapshot PCA technique ([1]). The snapshot PCA uses a temporal autocorrelation matrix to compute time-independent eigenvectors by utilizing the observation that data vectors and eigenvectors span the same linear space; therefore, eigenvectors can be written as a linear combination of data vectors. We would like to emphasize that the way we apply PCA on the 3D data is different from the traditional approach. The traditional approach involves vectorizing the 3D information and constructing a co-variance matrix by using multiple instances of the vectorized data. The multiple instances of the vectorized data come from the optical flow fields, which are computed from *multiple videos* of the same action. However, in our case we want to extract kinematic modes present within an optical flow field of a *single video*, and we achieve this by using all frames of the video to compute a temporal autocorrelation matrix of optical flows.

4.1 Principal Component Analysis

Principal component analysis is a well-known technique for determining an optimal basis for the reconstruction of data. Let $\mathbf{U}(\mathbf{x}, t_i)$, $i = 1, ..., M$, represents a vectorized sequence of experimental observations. The observation at each t_i is referred to as "snapshot" of the physical process that it is measuring. In the context of the current work, $u(\mathbf{x}, t_i)$ represents the optical flow computed at frame t_i of the given video. Without any loss of generality, the time average of the observation defined by

$$\bar{U}(\mathbf{x}) = \langle \mathbf{U}(\mathbf{x}, t_i) \rangle = \frac{1}{M} \sum_{i=1}^{M} \mathbf{U}(\mathbf{x}, t_i), \tag{8}$$

is assumed to be zero. Here, the symbol $\langle . \rangle$ represents the averaging operation. The PCA then extracts the *time independent* orthonormal basis $\phi_k(\mathbf{x})$ and *time-dependent* expansion coefficients $a_k(t_i)$, such that the reconstruction

$$\mathbf{U}(\mathbf{x}, t_i) = \sum_{k=1}^{M} a_k(t_i) . \phi_k(\mathbf{x}), \quad i = 1, ..., M, \tag{9}$$

is optimal in the sense that the average least squares truncated error

$$\epsilon_m = \langle \| \mathbf{U}(\mathbf{x}, t_i) - \sum_{k=1}^{m} a_k(t_i) . \phi_k(\mathbf{x}) \| \rangle \tag{10}$$

is minimum for any given number, $m \leq M$, of the basis vectors over all possible sets of orthogonal basis. Here $\| \ . \ \|$ is the L^2-norm, where "$.$" denotes the standard Euclidian inner product. The vectors $\phi_k(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_m(\mathbf{x}))$ are called the empirical eigenfunctions or dominant PCA modes of the data.

A computationally efficient implementation of PCA, when the resolution of the spatial grid, N, is higher than the number of observations M, is a snapshot PCA ([1]). It is based on the observation that the data vectors, $\mathbf{U}(\mathbf{x}, t_i)$, and the eigenvectors, ϕ_k, span the same linear space. This implies that the eigenvectors can be represented as a linear combination of the data vectors

$$\phi_k = \sum_{i=1}^{M} v_i^k \mathbf{U}(\mathbf{x}, t_i), \quad k = 1, \ldots, M. \tag{11}$$

The coefficients v_i^k can be obtained from the solution of

$$C\mathbf{v} = \lambda \mathbf{v}, \tag{12}$$

where $\mathbf{v} = (v_1^k, \ldots, v_M^k)$ is the k-th eigenvector of the above equation, and C is a symmetric $M \times M$ matrix defined by $C_{ij} = \frac{1}{M}(\mathbf{U}(\mathbf{x}, t_i) . \mathbf{U}(\mathbf{x}, t_j))$. Here, again "$(.)$" is the standard vector inner product, i.e., $(\mathbf{U}(\mathbf{x}, t_i) . \mathbf{U}(\mathbf{x}, t_j)) = u(x_1, t_i)u(x_1, t_j) + \ldots +$

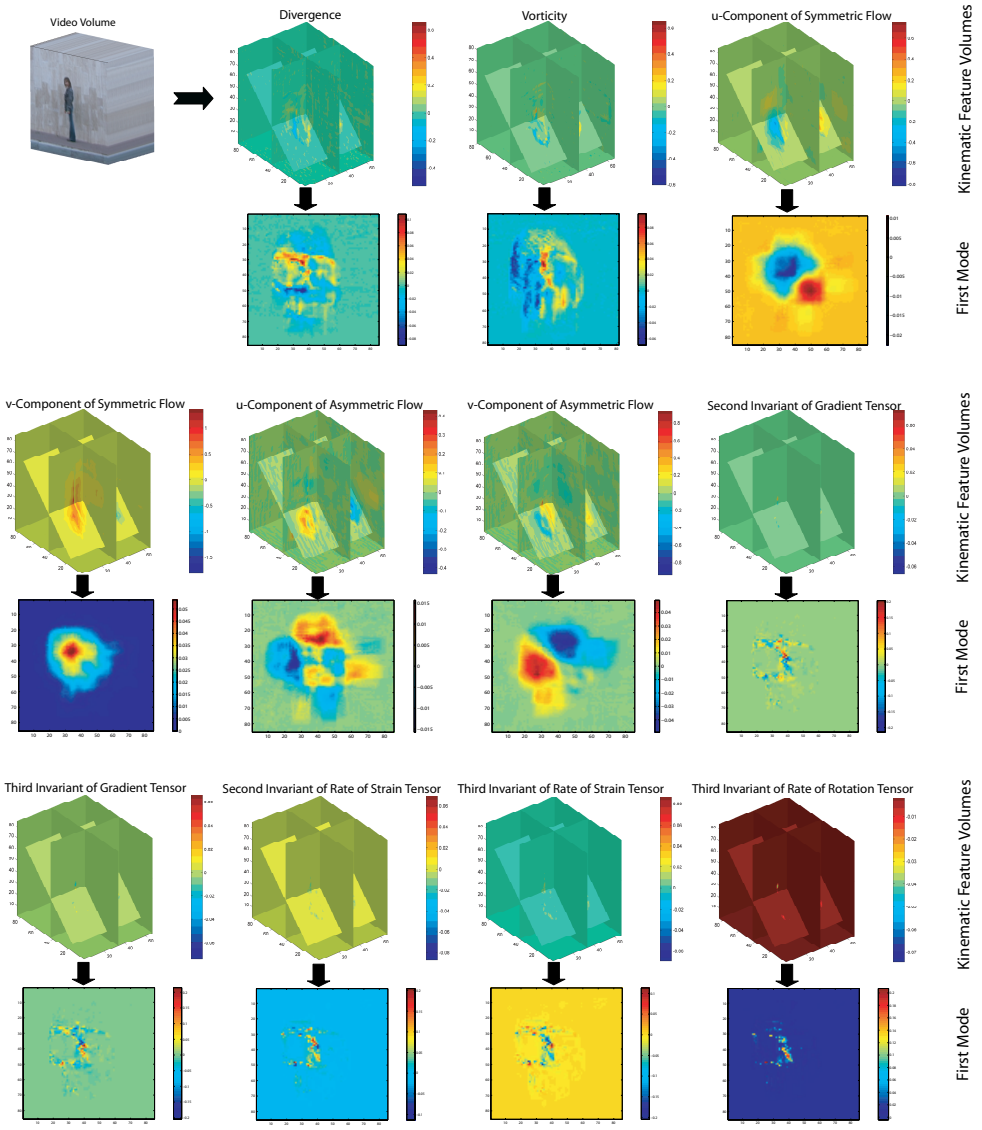

Fig. 5. The dominant kinematic modes of kinematic features for the "bend" action. The volumes are obtained by computing the kinematic features from the optical flow of the video. Three different slices are shown to emphasize the internal structure of each volume. The dominant kinematic modes are obtained by performing PCA on these volumes, as described in Section 4.

$u(x_N, t_i)u(x_N, t_j)$. In this way, the eigenvectors of the $N \times N$ matrix, R, can be found by computing the eigenvectors of a $M \times M$ matrix C, due to the relation in Equation 11, which is computationally more efficient when $N \gg M$.

The eigenvectors $\mathbf{v}$ are then used to compute the dominant temporal modes present in the optical flow, using the relation in Equation 11. Note that the computed eigenvectors are time independent; however, we can reconstruct the optical flow at any instant of time by computing the time-dependent temporal expansion coefficient as $a_k(t_i) = (\mathbf{U}(\mathbf{x}, t_i).\phi_k(\mathbf{x})), \quad i = 1, \ldots, M.$

4.2 Computation of Kinematic Modes

Our goal is to compute the kinematic modes representative of the dynamics of the flow field. The kinematic modes are not always the most energetic part of the flow, however they are essential ingredients of the spatio-temporal patterns representing the human action.

Therefore, to obtain kinematic modes of the optical flow, we compute *the orthogonal basis of the kinematic features of the optical flow field*. Theoretically, this can be done by treating the kinematic features $f^1(\mathbf{x}, t_i), f^2(\mathbf{x}, t_i), \ldots, f^{11}(\mathbf{x}, t_i)$, as kinematic kernels for the application of PCA. That is, we represent the kernel matrix C used in the eigenvalue problem (Equation 12) as:

$$C_k(t_i, t_j) = \frac{1}{M}(f^k(t_i).f^k(t_j)), \tag{13}$$

where k is the index of the kinematic feature being used. Here symbol $f^k(t_i)$, without spatial dependence, refers

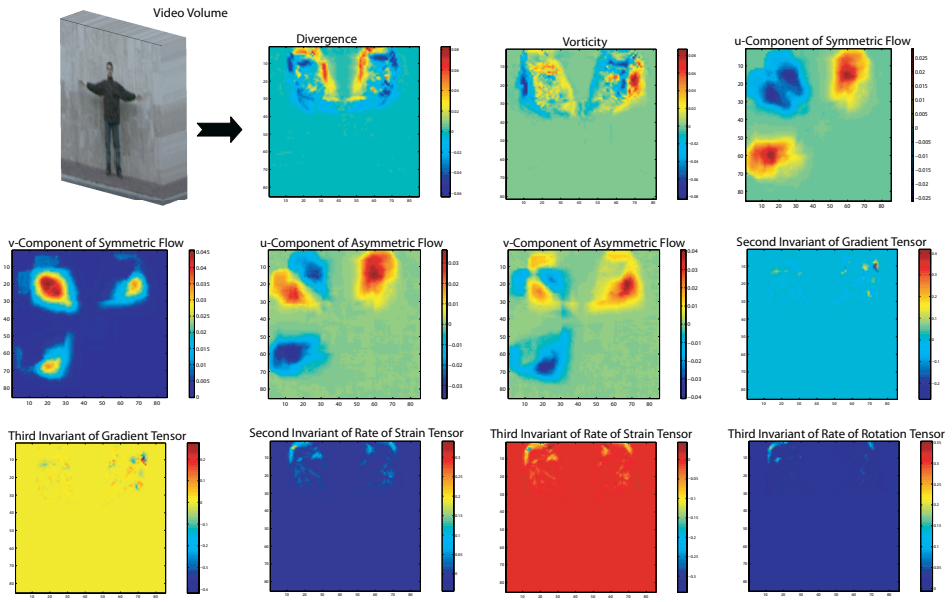

Fig. 6. The dominant kinematic modes of kinematic features for the "two hands wave" action.

to the vectorized version of $f^k(\mathbf{x}, t_i)$. From this kernel matrix we derive the time-independent eigen vectors, or kinematic modes, and time-dependent expansion coefficient. Note that we are able to do this because the newly constructed kernel matrix, C_k, satisfies the total positivity constraint. Two examples of kinematic modes computed in this way for the "bend" and the "hand-wave" action are shown in Figures 5 and 6, respectively.

5 Multiple Instance-based Learning

The action classifier is learned using multiple instance learning (MIL), in which each example is represented by several feature vectors called "instances". For the problem at hand, this translates into representing each action by several kinematic modes, where each kinematic mode is treated as an instance representing the action. In other words, we represent each video as a bag (collection) of kinematic modes, where the label of the action is associated only with the bag. The goal is then to learn the kinematic mode-based representation of each action so that we can predict whether a new bag represents that action or not. This is achieved by embedding bags into a *kinematic mode or instance based feature space*, and using the coordinates of the bag in that space for classification. The fundamental idea behind the embedding procedure is that each kinematic mode in a training set (positive as well as negative) can be viewed as an attribute or a feature for representing a bag. This point will become more clear through the following mathematical formulation of the idea based on the nomenclature of [2].

Let $B_i^+ = \{s_{i1}^{+f^1}, s_{i2}^{+f^1}, s_{i2}^{+f^2}, \ldots, s_{ij}^{+f^k}, \ldots\}$ (See Figure 7) denotes the ith positive bag, and $s_{ij}^{+f^k}$ represents the jth kinematic mode in B_i. The superscript f^k is used to identify the type of the kinematic feature generating the kinematic mode, and k ranges from $k = (1, \ldots, 11)$. The number of total kinematic modes in B_i^+ is represented by n_i^+, which is equal to the total number of modes (or eigenvectors) retained across all kinematic features for the video corresponding to B_i^+. Similarly, $B_i^- = \{s_{i1}^{-f^k}, s_{i2}^{-f^k}, \ldots\}$ represents the ith negative bag with n_i^- kinematic modes in it. The number of positive and negative bags in the training set is represented by l^+ and l^-, respectively.

As mentioned earlier, each kinematic mode in the training set, regardless whether it belongs to a positive or a negative bag, can be viewed as an attribute or feature for generating the instance-based representation of the bag. Therefore, we line up all the kinematic modes from all the bags into a set $\mathcal{C}$ (See Figure 7) and re-index the kinematic modes as $s_e^{f^k}$, where $e = \{1, \ldots, (\sum_{i=1}^{l^+} n_i^+ + \sum_{i=1}^{l^-} n_i^-)\}$. In other words, $\mathcal{C} = \{s_1^{f^1}, s_2^{f^1}, s_3^{f^2}, \ldots, s_j^{f^k}, \ldots, s_{e-2}^{f^{10}}, s_{e-1}^{f^{11}}, s_e^{f^{11}}\}$, and contains the kinematic modes of all kinematic features of all the videos in the training set.

For instance, if we keep two kinematic modes per kinematic feature per video, we will have $n_1^+ = 22$ kinematic modes for each video as there are 11 kinematic features in total. Then $s_1^{f^1} \in \mathcal{C}$ will be the first kinematic mode of kinematic feature f^1 of video B_1^+, while $s_{23}^{f^1} \in \mathcal{C}$ (which is the 23rd element of $\mathcal{C}$) will be the first kinematic mode of kinematic feature f^1 of video B_2^+. Following this notation, the kinematic modes of negative bags will span the range $(2 \times 11 \times l^+) + 1$ to e. This is also illustrated in Figure 7.

Since each member of $s_e^{f^k} \in \mathcal{C}$ is a feature, the conditional probability of a feature belonging to the bag

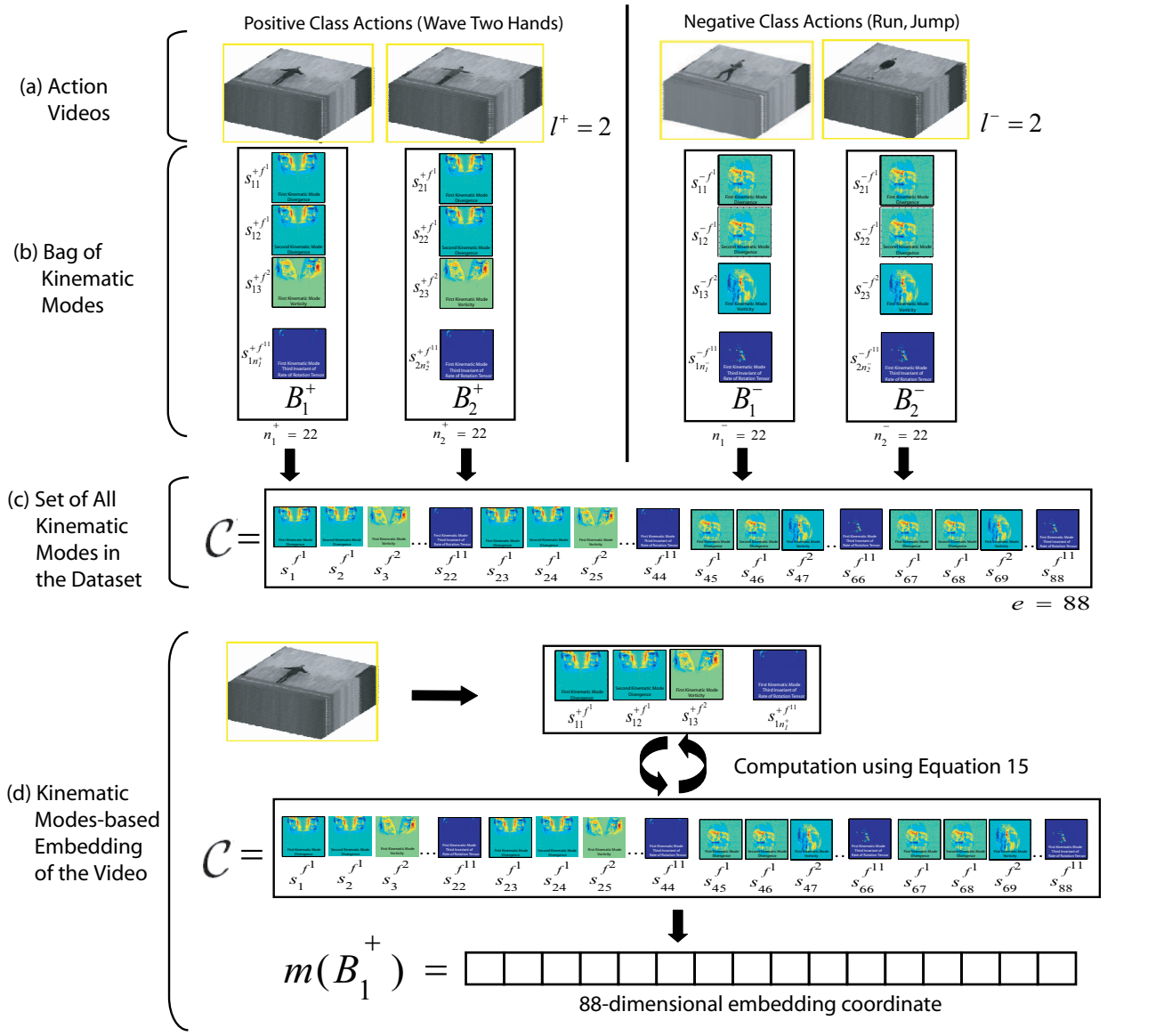

Fig. 7. (a) The data set consists of positive class videos from the "two hand wave" action and negative class videos from actions ''run" and ''jump". (b) For each video, the bag of kinematic modes is computed by performing PCA on the kinematic features of the video. Two kinematic modes per kinematic feature are retained giving rise to 22 kinematic modes for each video. (c) The set $\mathcal{C}$ is constructed by pooling all kinematic modes from all the videos in the data set. The kinematic modes are re-indexed resulting in $e = 88$ modes. (d) Next, the embedding coordinate of each video is computed. We are showing the computation process only for the first video. The values of the embedding coordinate $m(B_1^+)$ are populated by computing the similarity between the kinematic modes of video B_1^+ and kinematic modes in set $\mathcal{C}$ using Equation 15.

B is $P(s_e^{f^k} \mid B)$. Then,

$$[P(s_1^{f^1} \mid B), P(s_2^{f^1} \mid B), \ldots, P(s_j^{f^k} \mid B), \ldots, P(s_e^{f^{11}} \mid B)] \quad (14)$$

determines values of all the features for the bag B. Note that notation B without superscript $+$ or $-$ is used to represent a bag when the label of the bag does not matter. The feature space defined by the above vector is called the "instance-based" or "kinematic mode-based" feature space $\mathbb{F}_{\mathcal{C}}$, where each bag is a point in this space. Following [2], the conditional probability $P(s_e^{f^k} \mid B)$ of e-th kinematic mode $s_e^{f^k}$ independent of the bag label can be written as

$$P(s_e^{f^k} \mid B_i) \propto d(s_e^{f^k}, B_i) = \max_j exp(\frac{-\|s_{ij}^{f^j} - s_e^{f^k}\|^2}{\sigma^2}), \quad (15)$$
$$\forall f^j = f^k.$$

For example, the value of the z-th dimension of the embedding coordinates for B_i is equal to the similarity between the z-th kinematic modes in the set $\mathcal{C}$, and the closest kinematic mode of the same type from the bag B_i. This similarity is represented by the function d in the equation. The justification for the above equation comes from the observation that it is the most-likely-

cause estimator ([40]) for the cases when $\mathcal{C}$ is a single-point concept class. Note that we have imposed the additional restriction that the similarity between kinematic modes can only be computed if they are from the same kinematic feature. Next, the complete embedding coordinate $m(B_i)$ of the bag B_i in $\mathbb{F}_{\mathcal{C}}$ is written as:

$$m(B_i) = [d(s_1^{f^1}, B_i), d(s_2^{f^1}, B_i), d(s_3^{f^2}, B_i), \ldots, d(s_j^{f^k}, B_i), \ldots, d(s_e^{f^{11}}, B_i)]^T. \quad (16)$$

This way each action video is mapped to an e-dimensional vector. The process of computation of the embedding coordinate $m(B_i)$ is illustrated in Figure 7. Finally, the nearest-neighbor classifier is learned using the instance space coordinates $m(B_i)$ of bags belonging to different actions. For visualization purposes, the coordinates for different actions from the Weizmann action data set are shown in Figure 8.

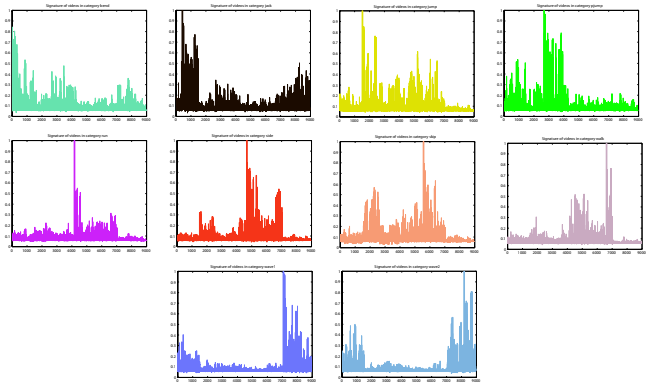

Fig. 8. Coordinate values of the 10 actions classes of the Weizmann action data set. The visual difference in the coordinate values of these different actions in the kinematic mode-based instance space is quite obvious.

6 Experiments

We present an evaluation of the proposed action recognition algorithm on two publicly available data sets: the Weizmann action data set ([4]) and the KTH action data set ([37]). Comparison is performed with other methods that have reported their results on these data sets. In addition, we compare the performance against the optical flow-based classification using the Weizmann data set. The goal of these experiments was to determine the presence or absence of the target action in the given video.

6.1 Weizmann Action Data Set

This data set contains 10 actions performed by 9 different actors. The actions are running, walking, skipping, jumping-jack,jumping forward on two legs, jumping in place on two legs, jumping sideways, waving with two hands and waving with one hand. This data set is a good starting point at which to test the feasibility of the algorithm. Originally, the data set consisted of 90 videos but we extended the number of action sequences by further dividing the videos. Each chunk of video contains roughly one cycle of an action. The length of the cycle was determined by observing one video per action and manually selecting the appropriate cycle length for that action. The selected cycle length was then used to divide the remaining videos of the action. We ignored any portion of the video in which the actor was not fully visible. This processing resulted in total of 180 videos. Next, background subtraction was used to extract the bounding boxes of moving actors. Each bounding box was resized to a pre-specified size of 100×100 pixels. The bounding boxes were stacked to generate the space-time volume of the action, which was then used to compute the optical flow. The resulting optical flow was employed to compute all the kinematic features and dominant kinematic modes. The number of kinematic modes per feature was used as a parameter in the experiments and we report results with different choices of this parameter. Once dominant kinematic modes for all training videos were computed, we embedded the videos into the kinematic mode-based feature space as described in Section 5. The value of parameter σ was fixed at 0.7 after varying it from 0.1 to 1, because 0.7 gave the best average result across different number of kinematic modes. Finally, the embedding coordinates of the training videos were used to learn a nearest neighbor classifier.

The testing was performed in a "leave-one-actor-out" setting, where the classifier was trained using all the videos except those corresponding to the actor in the test video. This process was repeated many times so each action video was treated as test video at least once. Figure 9 shows the confusion tables of different runs with different values of the "number of kinematic modes". Using just one kinematic mode per feature, we were able to achieve the mean accuracy of 80.3 percent for all ten actions. As we increased the number of kinematic modes per feature, an increase in the accuracy was observed. The best result was obtained by using four kinematic modes per feature, beyond which the change in the accuracy was insignificant. It is important to note that we are able to achieve a high level of accuracy by using just four kinematic modes per feature, which emphasizes our initial observation that computing orthogonal basis in terms of dynamics will reveal far more complex structures in fewer modes as compared to the energy-based orthogonal basis. The complex structures, in turn, translate into a more discriminating representation which helps our algorithm do a better job of distinguishing between different types of actions.

The confusion table corresponding to the best run in Figure 9 shows that the "run" action was often confused with the "skip" action. If we closely observe these two actions, the similarity between them is quite evident. In both actions, actors move across the field of view at almost the same speed and their limbs behave similarly, except in the "skip" action one of the legs which remains

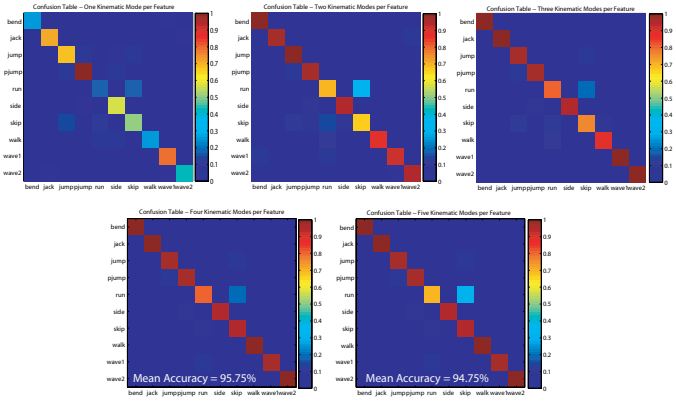

Fig. 9. Confusion tables for the Weizmann action data set. The parameter that was varied across the experiments was the number of kinematic modes per feature.

	1-Mode	2-Modes	3-Modes	4-Modes	5-Modes
Optical Flow	74.2%	76.43%	82.77%	82.25%	85.8%
Kinematic Features	80.3%	89.68%	93.18%	95.75%	94.75%

Fig. 10. Comparison of optical flow-based classification to the kinematic features based classification. Each column compares the mean accuracy on Weizmann action data set for different number of modes.

in the air throughout. This is a very minor difference, but our method was still able to deal with it adequately and only confused a few of the test examples.

6.1.1 *Comparison with Optical Flow-based Classification*

The objective of this experiment was to show the benefit of using kinematic features over the traditional optical flow-based representation. The experiment was performed in the following manner. We selected the same 180 videos and their optical flows which were used to compute the kinematic features in the previous section. The dominant modes of the optical-flow were computed by performing the snapshot PCA on the optical flows of each video. The dominant optical flow modes were then used to embed each video into an optical flow-based feature space utilizing method described in Section 5. All of the parameter values were kept the same as those used in the construction of the kinematic mode-based feature space. Again, for testing, we used a "leave-one-actor-out" setting. A comparison of the mean-accuracy for different experimental runs with different "numbers of modes" is presented in the table in Figure 10. In each case the gain in performance using the kinematic features is between 5-10 percent. This proves that the representation based on kinematic features is much more discriminating and powerful, compared to simple optical flow-based representation.

6.1.2 *Scale*

Next, an experiment was performed to test the robustness against scale changes. The action classifier mentioned above for 100×100 pixels bounding box is used for this purpose. The number of modes per kinematic feature was fixed at four. Then, we resized each bounding box to half its original size (50×50) and recomputed the optical flow. The optical flow was computed by using the block-based correlation described in Section 3, but using a window of size 8×8 pixels. The optical flow was resized to 100×100 pixels for subsequent kinematic feature computation. Using the original classifier, testing was performed in the "leave-one-actor-out" fashion and we observed a slight degradation in performance. The mean accuracy was 91.3 percent compared to 95.75 percent at the original resolution. The main reason was the loss of detail around crucial body parts, such as hands and legs, at this resolution, which ultimately leads to degraded information in the kinematic features as well. On the other hand, when we resized the bounding to twice the size (200×200, optical flow window size to 32×32 pixels) and performed the classification, the mean accuracy (95.2 percent) remained almost the same. This further supports the above argument because increasing the size does not affect the details regarding the motion of crucial body parts.

6.2 KTH Action Data Set

In order to further verify the performance of our algorithm, we conducted experiments on the KTH data set. It consists of six different actions: hand clapping, hand waving, boxing, running, walking and jogging. This is a challenging data set for a number of reasons. First, the camera is no longer static and often zooms in and zooms out while capturing the action. Therefore, the size of the actor varies considerably within the sequence. Second, each action is performed by a number of actors in varying postures and lighting. Finally, the camera angle also varies across instances of the same action.

Since our algorithm requires person localization for learning and recognition, we first performed person segmentation within each video. Another option was to have a sliding window that moves over the image at all possible scales, but that would have been computationally very expensive. The person segmentation was done using the contour evolution method proposed in [41]. The evolution process minimizes the background likelihood of the region within the contour by using a level set based framework. For each frame, we initialized the contour along the boundary. Figure 11 shows the output of this step for the boxing action. Once the actor was localized within a frame, we put a bounding box around him and computed the optical flow within the bound box only. The computed flow field was resized to a predefined size of 70×70 pixels. The computed optical flow fields were then stacked up to generate a space-time volume of flow fields for the video. Next, the resulting space-time volume of optical flows was employed to compute the kinematic features and kinematic modes.

For the experiments, the input data set was divided into a training and a testing set using the standard

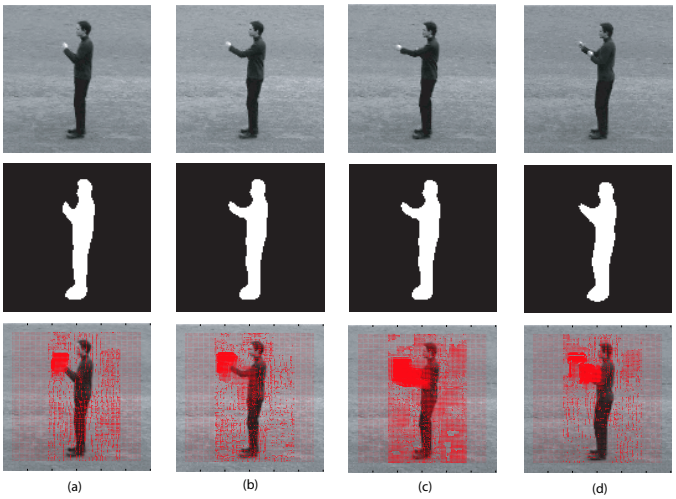

Fig. 11. The process of optical flow computation on the KTH data set. First Row: Frames from the input sequence. Second Row: Masks obtained after the segmentation. Third Row: Optical flow fields computed only for the region containing the person. (a-d) Results for frame 1, 3, 5 and 7, respectively.

	Boxing	Hand Clapping	Hand Waving	Jog	Run	Walk
Boxing	88.5%	10.2%	1.3%	0	0	0
Hand Clapping	5.35%	86.44%	8.21%	0	0	0
Hand Waving	5.4%	7.43%	84.46%	1.12%	1.59%	0
Jog	0	0	0	86.2%	9.78%	4.003%
Run	0	0	0	6.1%	91.51%	2.39%
Walk	0	0	0	7.49%	3.4%	89.11%

Fig. 12. The table shows the confusion matrix obtained by our method for the KTH data set.

splits [44]. Note that each video in this data set has multiple cycles of the same action; however, we only use one cycle for training and testing. For each experiment, the kinematic mode-based feature space was constructed using the kinematic modes belonging only to the training examples. The value of sigma was kept at 0.7 for this purpose. A nearest neighbor classifier was learned using the coordinates of each video in the embedding space. Ten kinematic modes per feature were used for this experiment.

Table in Figure 12 shows the confusion matrix obtained by our method. We are able to achieve mean accuracy of 87.7 percent on these six actions. It can be observed that the jogging action is confused most of the time with the running action and vice versa. This is understandable as these actions have a high degree of similarity in terms of the kinematics. In these videos, the actors performed the actions without any strict experimental restriction on the speed with which they ran; therefore, it was quite possible that many of the jogging sequences were performed at speeds comparable to the running action. In addition, the PCA attempts to capture only the dominant flow dynamics of each feature, and therefore does not take into account small variations in speed. The observation is further verified by the confusion between running and jogging actions, and confusion of the walking action with both the running and jogging actions. It is important to note that the performance of our algorithm depends heavily on the quality of the optical flow. In the KTH data set the computation of the optical flow was relatively difficult due to rapid illumination changes and blurring between frames.

Table 1 shows a comparison between the performance of our algorithm and that of other methods which have reported results on the KTH data set. We achieved better recognition accuracy than all methods ([9], [10], [5], [37],[43]) except [42]. However, we want to emphasize that Table 1 should be interpreted in the light of the fact that our proposed method and Kim *et al.* [42] have used additional information for training by temporally segmenting the subject performing the action. The rest of the methods rely on supervision provided by the data set only. Nevertheless, it is interesting to note that the approaches with respect to which we achieved higher accuracy are the ones that use sparse features, such as interest points, distributed volumetric features, quantized, and spatio-temporal words. The sparsity of the features allows these methods to focus only on specific regions of the spatio-temporal pattern. This may be a good strategy in scenarios where it is expected that changes in viewpoint will be significant, and the actor may undergo occlusion. But it is not a good idea to compute sparse features when one has access to complete spatio-temporal patterns, because by computing sparse features, we will discard useful pieces of global information.

We also performed an experiment to show the benefit of kinematic features over the traditional optical flow-based representation on this data set. The experiment setup was the same as described previously for the Wizemann action data set. The same partitioning of the data set was used as mentioned in [44]. Ten dominant optical flow modes were used per video to embed the video into optical flow-mode based feature space. The classification is again performed using the nearest neighborhood classifier, and we achieved a mean accuracy of 79.22 percent. Therefore, the gain of kinematic features based representation (87.7 percent mean accuracy) over optical flow representation was close to 8 percent. This shows that the kinematic features are again capturing discriminating patterns which help achieve better performance.

6.3 Feature Contributions

The purpose of this experiment was to validate the contribution of each feature towards action classification using the Weizmann action data set. In order to do that, a nearest neighbor classifier was learned using one feature at a time and then its performance was tested on the

Niebles *et al.* [9]	Dollar *et al.* [10]	Ke *et al.* [5]	Schuldt *et al.* [37]	Kim *et al.* [42]	Wong *et al.* [43]	Our Approach
81.5%	81.17%	62.96%	71.72%	95.33%	71.16%	87.7%

TABLE 1
Comparison of mean classification accuracy on the KTH data set.

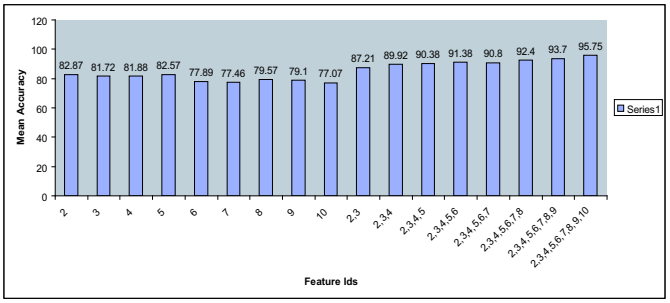

Fig. 13. The bar chart shows the mean-accuracy of the proposed algorithm for different combinations of kinematic features. The first nine bars corresponds to the results obtained using only one feature at a time. While the remaining eight bars correspond to different combination of features. The label on each bar corresponds to the kinematic feature(s) used for that experiments. They are 2=Divergence, 3=Vorticity, 4=Symmetric Flow (both u and v components), 5=Asymmetric Flow (both u and v components), 6=Second Invariant of the Gradient Tensor, 7=Third Invariant of the Gradient Tensor, 8=Second Invariant of the Rate of Strain Tensor, 9=Third Invariant of Rate of Strain Tensor, 10=Third Invariant of Rate of Rotation Tensor, respectively.

data set. For instance, the performance of divergence feature was measured by representing each video using only feature f^1, which was followed by computation of kinematic modes using the snapshot PCA. Next, kinematic modes-based embedding of each video was performed and a nearest neighbor classifier was learned. The testing was performed using a "leave-one-actor-out" cross validation. In order to derive a comparison with the performance obtained using all the features on this data set, we kept all the parameters (number of modes, value of sigma, etc.) the same as for the experiments reported in Section 6.1. Next, we began adding features one by one and observed how the overall performance changed as the number of features increased. Again, the experimental parameters are kept the same. The results of these experiments are summarized in Figure 13. The x-axis shows the feature IDs (which are described in the caption), while the y-axis corresponds to the mean accuracy for each run.

The graph shows that the results obtained by individual features are reasonable, but the performance is far below the results obtained by using all of the features. It is also clear from the graph that the addition of more features helps to obtain a steady increase in overall performance. This is reflected in the increasing trend in the bars corresponding to different combinations of features. The best result of 95.75 percent is obtained by using all kinematic features together. We believe this is because each feature is providing complementary information, thus allowing our method to construct a more discriminative embedding for the videos. Furthermore, we observed that the increase in the number of kinematic features is especially helpful in distinguishing actions which are very similar to each other, such as walking, running and skipping actions. It can be concluded that these features provide a unique description of the action, which if used collectively, can achieve much better performance than the individual features.

7 Discussion & Conclusion

Although, our proposed algorithm works well for the task of action recognition, a number of weaknesses are apparent. First, the kinematic features are not view-invariant because the same action viewed from different angles will induce a different optical flow on the images. One solution is to discretize the number of views and learn a separate kinematic feature-based representation for each view. Second, occlusion will severely affect the performance of our algorithm especially in cases where a crucial body part is occluded, e.g., the hand in the case of the "wave" action. The main reason is the holistic nature of our approach. However, if occlusion only obscures the parts of the body that are not taking part in the action, our approach is expected to be robust. The quality of optical flow is also an important factor in our algorithm.

The current implementation of the algorithm is in Matlab. On a dual core Pentium processor with 1GB RAM, it takes 30-60 seconds to compute optical flow for a video containing 60 frames, where each frame is resized to 100×100 pixels. The computation of kinematic features takes 30-40 seconds per video, while the computation of kinematic modes require an average of 10 seconds per video. The embedding step is the slowest part of the algorithm due to excessive memory consumption and the iterative nature of the program. For instance, on the Weizmann action data set, it takes 5-10 minutes to compute the embedding coordinates of all 180 videos.

In summary, in this paper we have explored the utility of kinematic features derived from motion information for the task of human action recognition in videos. The kinematic features are computed from the optical flow. The features are divergence, vorticity, symmetric and anti-symmetric optical flow fields, second and third principal invariants of flow gradient and rate of strain tensor, and third principal invariant of rate of rotation tensor. Next, it was hypothesized that the dynamic information of the optical flow is represented by the kinematic features in terms of dominant kinematic trends or modes. These dominant kinematic modes are computed by performing PCA on each kinematic feature. For classification, we proposed a multiple instance learning (MIL)

model where each action video is treated as a bag or a collection of kinematic modes. Each bag is embedded into a kinematic mode-based feature space, in which the coordinates of the videos in this space are used for classification using the nearest neighbor classifier.

REFERENCES

[1] L. Sirovich, *Turbulence and the Dynamics of Coherent Structures: Part I-III*, Quart. Appl. Math, 45, pp.561-590, 1987.
[2] Y. Chen, J. Bi, and J. Z. Wang, *MILES: Multiple Instance Learning via Embedded Instance Selection*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 28, No. 12, 2006.
[3] A. Yilmaz and M. Shah, *Actions Sketch: A Novel Action Representation*, IEEE CVPR, 2005.
[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, *Actions as Space-Time Shapes*, IEEE ICCV, 2005.
[5] Y. Ke, R. Sukthankar, and M. Hebert, *Efficient Visual Event Detection using Volumetric Features*, IEEE ICCV, 2005.
[6] E. Shechtman and M. Irani, *Space-Time Behavior Based Correlation*, IEEE CVPR, 2005.
[7] A. F. Bobick and J. Davis, *An Appearance-Based Representation of Action*, IEEE CVPR, 1996.
[8] A. F. Bobick and J. Davis, *The Recognition of Human Movement using Temporal Templates*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 23, No. 3, 2001.
[9] J. C. Niebles, H. Wang, and L. Fei-Fei *Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words*, BMVC, 2006.
[10] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, *Behavior Recognition via Sparse Spatio-Temporal Features*, IEEE International Workshop on VS-PETS, 2005.
[11] T. Darrell and A. Pentland, *Classifying Hand Gestures with a View-Based Distributed Representation*, NIPS, 1993.
[12] J. Yamato, J. Ohya, and K. Ishii, *Recognizing Human Action in Time Sequential Images Using Hidden Markov Model*, IEEE CVPR, 1992.
[13] C. Cedras and M. Shah, *Motion based Recognition: A Survey*, Image and Vision Computing, 1995.
[14] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata, *Articulated and Elastic Non-Rigid Motion: A Review*, In Workshop on Motion of Non-Rigid and Articulated Objects, 1994.
[15] J. K. Aggarwal and Q. Cai, *Human Motion Analysis: A Review*, Computer Vision and Image Understanding, 1999.
[16] S.X. Ju, M.J. Black, and Y. Yacoob, *Cardboard People: A Parameterized Model of Articulated Image Motion*, 2nd International Conference on Automatic Face and Gesture Recognition, 1996.
[17] Y. Yacoob and M. Black, *Parameterized Modeling and Recognition of Activities*, Computer Vision and Image Understanding, 1999.
[18] M. J. Black and Y. Yacoob, *Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion*, IEEE ICCV, 1997.
[19] C. Carlsson and J. Sullivan, *Action Recognition by Shape Matching to Key Frames*, Workshop on Models Versus Exemplars in Compuetr Vision, 2001.
[20] H. Jiang, M. S. Drew and Z. N. Li , *Successive Convex Matching for Action Detection*, IEEE CVPR, 2006.
[21] T. Starner and A. Pentland, *Visual Recognition of American Sign Language using Hidden Markov Model*, Internation Workshop on Automatic Face and Genesture Recognition, 1995.
[22] K. M. Cheung, S. Baker and T. Kanade, *Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture*, IEEE CVPR, 2003.
[23] T. S. Mahmood, A. Vasilescu and S. Sethi, *Recognition Action Events from Multiple View Points*, IEEE Workshop on Detection and Recognition of Events in Video, ICCV, 2001.
[24] J. Liu, S. Ali, and M. Shah, *Recognizing Human Actions Using Multiple Features*, IEEE CVPR, 2008.
[25] P. Scovanner, S. Ali, and M. Shah, *A 3-Dimensional SIFT Descriptor and its Application to Action Recognition*, ACM Multimedia, 2007
[26] D. G. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, International Journal of Computer Vision (IJCV), 60(2), 2004.
[27] D. Weinland, R. Ronfard and E. Boyer, *Free View-point Action Recognition Using Motion History Volumes*, Computer Vision and Image Understanding, 2006.
[28] J. Little and J. E. Boyd, *Recognizing People by Their Gait: The Shape of Motion*, Journal of Computer Vision Research, 1998.
[29] J. Hoey and J. Little, *Representation and Recognition of Complex Human Motion*, IEEE CVPR 2000.
[30] T. Arbel, F. Ferrie and M. Mitran, *Recognizing Objects from Curvilinear Motion*, BMVC, 2000.
[31] J. Little and J. Boyd, *Describing Motion For Recognition*, SCV, 1995.
[32] A. Efros, A. Berg, G. Mori and J. Malik, *Recognizing Action at a Distance*, IEEE ICCV, 2003.
[33] I. Laptev and T. Lindeberg, *Space Time Interest Points*, IEEE ICCV, 2003.
[34] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, *Learning Realistic Human Actions from Movies*, IEEE CVPR, 2008.
[35] A. Oikonomopoulousm, I. Patras and M. Pantic, *Spatiotemporal Saliency for Human Action Recognition*, IEEE ICME, 2005.
[36] A. Oikonomopoulousm, I. Patras and M. Pantic, *Kernel based Recognition of Human Actions Using Spatiotemporal Salient Points*, IEEE ICME, 2005.
[37] C. Schuldt, I. Laptev and B. Caputo, *Recognizing Human Actions: A Local SVM Approach*, IEEE ICPR, 2004.
[38] P. A. Durbin and B. A. Pettersson Reif, *Statistical Theory and Modeling for Turbulent Flows*, *John Wiley & Sons* , 2003.
[39] http://urapiv.wordpress.com
[40] O.Maron, *Learning from Ambiguity*, Dept.of Electrical and Computer Science, Massachusetts Inst.of Technology, Cambridge, 1998.
[41] A. Yilmaz, Xin Li, and Mubarak Shah, *Contour-Based Object Tracking with Occlusion Handling in Video Acquired Using Mobile Cameras*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 26, No.11, 2004.
[42] T.K. Kim, S.F. Wong, and R. Cipolla, *Tensor Canonical Correlation Analysis for Action Classification*, IEEE CVPR, 2007.
[43] S.F. Wong and R. Cipolla, *Extracting Spatiotemporal Interest Points using Global Information PDF*, IEEE ICCV, 2007.
[44] http://www.nada.kth.se/cvap/actions/00sequences.txt

S aad Ali received the BS degree in Computer System Engineering from the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan in 2003, and Phd in Computer Science from the University of Central Florida in 2008. Currently, he is working as a post doctoral research fellow at the Robotics Institute of Carnegie Mellon University. His research interests include surveillance in crowded scenes and aerial imagery, action recognition, dynamical systems, and object recognition. He has published a number of papers in top vision conferences such as ICCV, CVPR, and ECCV. He is a student member of IEEE.

D r. Mubarak Shah, Agere Chair Professor of Computer Science, is the founding director of the Computer Visions Lab at UCF. He is a co-author of three books (Motion-Based Recognition (1997) and Video Registration (2003), and Automated Multi-Camera Surveillance: Algorithms and Practice (2008)) all by Springer. Dr. Shah is a fellow of IEEE, IAPR and SPIE. In 2006, he was awarded a Pegasus Professor award, the highest award at UCF, given to a faculty member who has made a significant impact on the university, has made an extraordinary contribution to the university community, and has demonstrated excellence in teaching, research and service. He was an IEEE Distinguished Visitor speaker for 1997-2000 and received IEEE Outstanding Engineering Educator Award in 1997. He received the Harris Corporations Engineering Achievement Award in 1999, the TOKTEN awards from UNDP in 1995, 1997, and 2000; Teaching Incentive Program award in 1995 and 2003, Research Incentive Award in 2003, Millionaires Club awards in 2005 and 2006, University Distinguished Researcher award in 2007, honorable mention for the ICCV 2005 Where Am I? Challenge Problem, and was nominated for the best paper award in ACM Multimedia Conference in 2005. He is an editor of international book series on Video Computing; editor in chief of Machine Vision and Applications journal, and an associate editor of ACM Computing Surveys journal. He was an associate editor of the IEEE Transactions on PAMI, and a guest editor of the special issue of International Journal of Computer Vision on Video Computing.