Image and Vision Computing 31 (2013) 460–472

Contents lists available at SciVerse ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

# Vector field analysis for multi-object behavior modeling[☆], [☆☆]

Nandita M. Nayak, Yingying Zhu, Amit K. Roy-Chowdhury [*]

*University of California, Riverside, Riverside, CA, United States*

ARTICLE INFO

*Keywords:*
Optical flow
Helmholtz decomposition
Complex activity recognition

ABSTRACT

This paper proposes an end-to-end system to recognize multi-person behaviors in video, unifying different tasks like segmentation, modeling and recognition within a single optical flow based motion analysis framework. We show how optical flow can be used for analyzing activities of individual actors, as opposed to dense crowds, which is what the existing literature has concentrated on mostly. The algorithm consists of two steps — identification of motion patterns and modeling of motion patterns. Activities are analyzed using the underlying motion patterns which are formed by the optical flow field over a period of time. Streaklines are used to capture these motion patterns via integration of the flow field. To recognize the regions of interest, we utilize the Helmholtz decomposition to compute the divergence potential. The extrema or critical points of this potential indicates regions of high activity in the video, which are then represented as motion patterns by clustering the streaklines. We then present a method to compare two videos by measuring the similarity between their motion patterns using a combination of shape theory and subspace analysis. Such an analysis allows us to represent, compare and recognize a wide range of activities. We perform experiments on state-of-the-art datasets and show that the proposed method is suitable for natural videos in the presence of noise, background clutter and high intra class variations. Our method has two significant advantages over recent related approaches — it provides a single framework that takes care of both low-level and high-level visual analysis tasks, and is computationally efficient.

## 1. Introduction

Natural videos usually consist of multiple motion patterns generated by objects moving at arbitrary speeds and time intervals. They could have multiple events occurring simultaneously at arbitrary viewpoints and varying scales. The analysis of such videos can be termed as complex activity recognition. Recognition of complex activities often involves dealing with features distributed in a high dimensional space due to a higher amount of intra class variations. Algorithms dealing with such sequences should be robust to background clutter, noise and changes in viewpoint and scale. Most of the traditional activity recognition algorithms, such as in Refs. [15,2,13], work with simpler datasets like in Refs. [24,2] which place assumptions on the number of objects, scale and viewpoint of the scene. However, in real world situations it is hard to encounter such videos. Therefore, there is a need for algorithms which can handle the structure and semantics of complex activities.

A scene is a collection of moving pixels. Optical flow provides a natural representation for this motion. It represents the pixel-wise motion from one frame to the next; therefore, it captures the spatial and temporal dynamics in a video. Since a complex activity involves multiple motion patterns, it is useful to separate the motion patterns before modeling them, to reduce the search space. One way of doing this would be to compute tracks. However, it is not always feasible to compute accurate tracks in real world videos. The problem of separation of motion pattern reduces to the problem of segmentation of optical flow. Although prone to the same inaccuracies as tracks, optical flow is computed for every pixel in the video. It is therefore, a more statistically reliable indicator in the presence of noise. These factors motivate us to use optical flow as the input features for our recognition algorithm.

In this work, we recognize activities by analyzing the underlying pixelwise motion using optical flow. Each region in a video where the pixels exhibit similar motion is said to constitute a motion pattern. Individual motion patterns are considered as “events” which can be identified by segmenting the flow patterns. This motion pattern could be due to one or more objects in the scene. An activity is represented as a collection of motion patterns. Optical flow is represented using streaklines which are obtained by integrating the flow over time. The activity in a video could be composed of multiple such motion patterns, which are assumed to be correlated. Therefore, the overall match score between two videos is obtained by matching the individual motion patterns. The streaklines which constitute a motion pattern can be identified using their average shape vectors and spatio-temporal variation with respect to the average shape. This variation is modeled using a collection of linear subspaces which capture their spatio-temporal

[☆] This paper has been recommended for acceptance by Matti Pietikainen.

[☆☆] This work has been partially supported by NSF grant IIS-0905671 and the DARPA VIRAT program.

[*] Corresponding author. Tel.: +1 951 827 7886.
*E-mail addresses:* nandita.nayak@email.ur.edu (N.M. Nayak), yzhu001@ucr.edu (Y. Zhu), amitrc@ee.ucr.edu (A.K. Roy-Chowdhury).

0262-8856/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.imavis.2012.08.011

variation in a low dimensional representation. These patterns can be matched by a combination of shape comparison and subspace analysis. We validate the robustness of our algorithm by experimenting on two realistic outdoor datasets. We do not place any assumptions on the number of motion patterns in the scene. The proposed method can be used across a wide range of activities with varying scales and viewpoints. Some sample frames of the data used for activity recognition are shown in Fig. 1.

### 1.1. Related work

A major thrust of research in complex activity recognition has been in the selection of features and their representations. Different representations have been used in activity recognition, most of which can broadly be classified as local or global representations [26]. Local representations like in Refs. [13,15] identify small spatio-temporal regions in the video as the regions of interest. The spatial and temporal modeling of activities is then performed in the recognition stage. Global representation like in Refs. [3,5] on the other hand, model the scene as a whole. These representations often span a larger spatio-temporal volume, so the spatial and temporal information is captured in the features themselves. Methods such as in Ref. [9] use STIP-based features for recognition of complex activities. The recognition is then performed by modeling relationships between these features in a complex graph based or histogram based framework. We hypothesize that representing motion patterns using optical flow is more intuitive than using spatio-temporal features since the spatio-temporal information is embedded in the flow. This therefore, is a global representation. Also, unlike previous global methods which use histograms of optical flow, we explicitly model the spatial and temporal evolution of flow.

Optical flow has widely been used in the past for activity recognition. It serves as an approximation of the true motion of objects projected onto the image plane [26]. Optical flow has predominantly been used in features like space–time interest points (STIP) [16] as a part of the feature descriptor. The time series of histogram of optical flow has been modeled as a non-linear dynamical system using Binet–Cauchy kernels in Ref. [5]. Optical flow histograms have also been used to analyze the motion of individual players in soccer videos [8]. Most of such approaches utilize the statistics of optical flow for recognition rather than the flow itself. This removes the spatio-temporal structure from the flow. They also assume that the flow belongs to one object in the scene. Optical flow has been extensively used in crowd motion analysis. Dense crowd motion analysis and segmentation has been performed using optical flow in Ref. [12]. Helmholtz decomposition has been used to segment different motions in crowd scenes by streakline computation in Ref. [18]. In contrast to the above trends, we show how flow-based methods can be used in the analysis of multi-object scenes with sparse motion.

Several recent approaches have dealt with recognition of complex activities. The authors in Ref. [29] deal with multi-object activity recognition but only focus on recognition of simple actions such as entering and exiting a door, performed by multiple actors in an indoor scene which is free from clutter. We on the other hand deal with cluttered scenes. Graphical models are commonly used to encode the relationship between features. A Dynamic Bayesian network has been used to model the temporal evolution in two person activities in Ref. [20]. A grid based belief propagation method was used for pose estimation in Ref. [17]. Some methods have tried to incorporate the location of feature points into the recognition algorithm. The spatial and temporal relationships between space–time interest points have been encoded as "feature graphs" in Ref. [9]. A logic based method is used to match spatio-temporal relationships between STIP points in Ref. [22]. Complex activities were represented as spatio-temporal graphs representing multi-scale video segments and their hierarchical relationships in Ref. [4]. Stochastic and context free grammars have been used to model complex activities in Ref. [21]. Co-occurring activities and their dependencies have been studied using Dependent Dirichlet Process–Hidden Markov Models (DDP–HMMs) in Ref. [14]. Our approach to complex activity modeling relies on using dense features, namely streaklines, which capture the spatio-temporal information in a video. We compare activities in one video to those in another by modeling and comparing motion patterns in the videos.

Linear and non-linear dynamical models have been used in the past to model activities. The authors in Ref. [27] model motion as a non linear dynamical model of the changes in configurations of point objects. However, they utilized hand-picked location data with no observation noise in their experiments. Binet–Cauchy kernels were used to model single person activities as a non-linear dynamical system in [5]. Auto-regressive moving average (ARMA) models have been used to model movements as a linear dynamical system. The authors in Ref. [28] utilize ARMA models for single person gait recognition using shape deformations of the silhouette. ARMA models have been used for track based activity and gait recognition in Ref. [1]. Dynamic textures have been represented using ARMA models in Ref. [6]. We utilize a combination of non-linear shape matching and subspace analysis in our approach. We model the spatio-temporal evolution in a motion pattern using linear subspaces which can be matched using subspace angles. We also use shape matching on a non-linear manifold to compute the distance between the average shapes of two motion patterns.

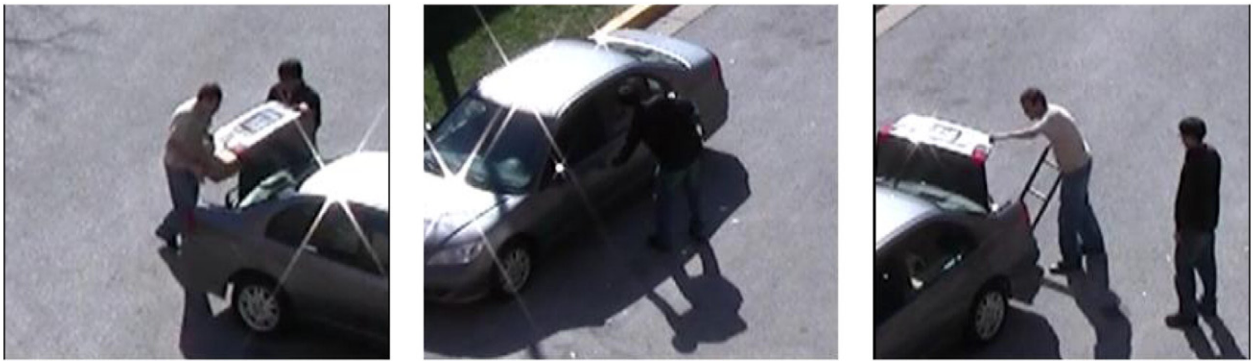

**Fig. 1.** The figure shows sample frames of the VIRAT dataset used for recognition. The first figure shows a person loading a trunk, the second figure shows a person entering a vehicle and the third figure shows a person closing a trunk. We notice other people in the scene adding to background clutter. Lighting changes and shadows add noise to the data.

### 1.2. Some definitions

These are the definitions of some of the commonly used terminology in the paper:

*Motion pattern* — a spatio-temporal region in a video in which all pixels exhibit similar motion. Each motion pattern is considered as an "event" in the video.

*Activity* — the action which is to be recognized in a video. An activity is composed of one or more events.

*Streakline* — the locus of all points in a video which have passed through a particular pixel.

*Particle* — an abstraction of a point on a streakline.

### 1.3. Main contributions

1. The main contribution of our work is to propose a unified framework for analysis of activities. We provide an end-to-end system that can perform a bottom-up analysis starting from pixel wise motion to identifying motion regions in the volume to segmentation and modeling of these regions. Some state-of-the-art methods like in Refs. [9], which deal with similar datasets, explore spatio-temporal information at a feature level. Our method on the other hand explores spatio-temporal information at a global level. This has the advantage that we can segment out different events occurring in the video and then model them in a single framework. Thus, we propose a framework based on the analysis of flow that is able to handle the entire image analysis pipeline — from the low level to the high level processing.
2. Another contribution of this work lies in the use of optical flow for multi-object behavior analysis. Unlike previous methods which utilize optical flow in the form of motion statistics [5], we model the actual dynamics of flow rather than using histograms which do not retain the spatial and temporal information. Therefore, we provide a framework for representation and comparison of complex activities using optical flow.
3. Although we have built upon the work in Ref. [18] which uses streaklines and Helmholtz decomposition for crowd segmentation, there are several differences in our work as compared to theirs in the modeling and in the application of streaklines. First, the objective of the proposed method in Ref. [18] is to segment a video into different regions exhibiting similar motion, whereas our objective is to explicitly model every motion pattern in a video for the purpose of activity recognition. In Ref. [18], the authors propose a method to perform a space segmentation of the streaklines at every frame, whereas we deal with spatio-temporal segmentation of the entire volume. We compute the distance between critical points to identify time segments of motion patterns. In Ref. [18], the Helmholtz decomposition is again used to compute a divergence factor, which is then used to identify abnormal activities. Here, we use the Helmholtz decomposition to identify the regions which are of interest to us for the purpose of modeling and recognizing activities. Therefore, we have extended the method in Ref. [18] to work not just on crowded environments but also in videos which contain sparse motion.

### 1.4. Overview of proposed approach

The overall algorithm is described in Fig. 2. The goal of our algorithm is to model the activity in a video as a combination of motion patterns. There are two components to the algorithm — identification of motion patterns and modeling and comparison of motion patterns.

The identification of motion pattern involves identifying regions in the video which correspond to useful motion and segmenting these regions into individual motion patterns. These regions of interest are termed as motion regions. We start by computing the optical flow at each time instant. Optical flow is highly susceptible to noise which can result in spurious patterns which are difficult to analyze. Therefore, we work with streaklines which are obtained by integrating optical flow over time. Motion regions are then identified as the streaklines which show a significant amount of motion. We demonstrate a framework based on the Helmholtz decomposition of a vector field to extract these regions.

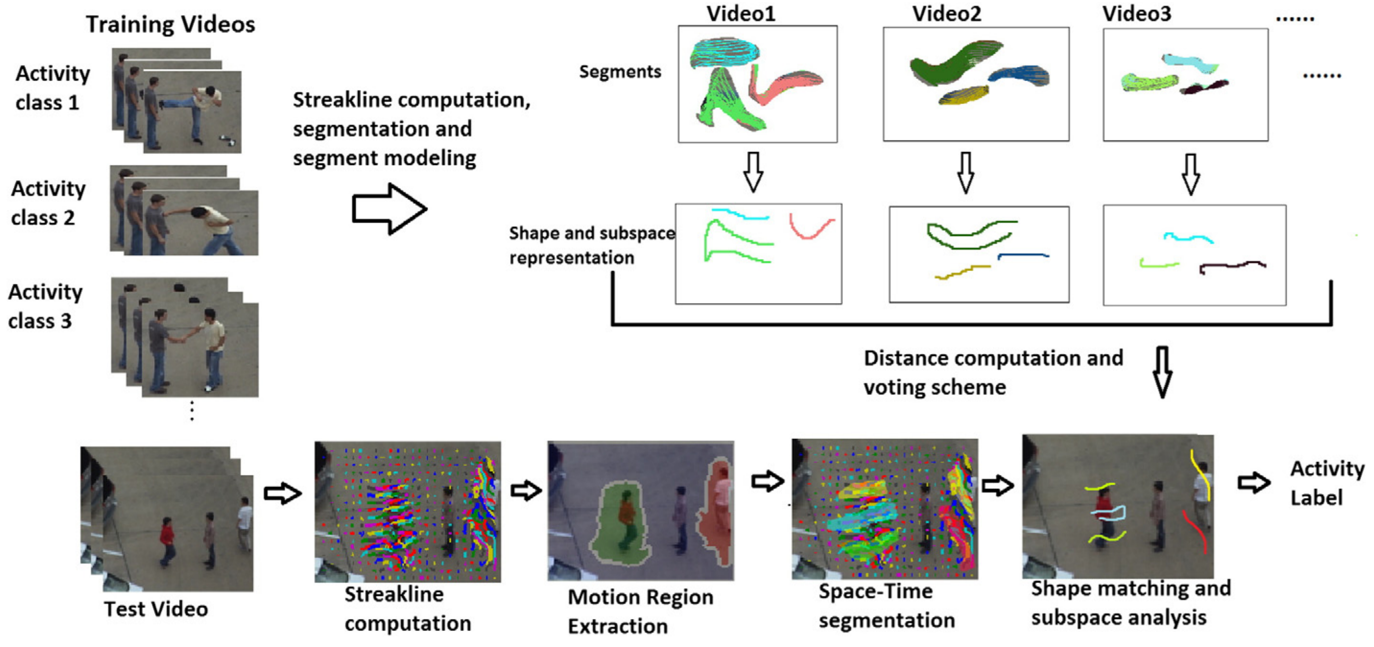

**Fig. 2.** The figure shows the overall framework of the proposed method.

Once we identify the streaklines which correspond to the motion regions in a video, motion patterns are recognized by performing a space–time clustering on these streaklines. We demonstrate a method of identifying time segments of streaklines using the Helmholtz decomposition. We further perform a space segmentation by running a clustering algorithm. After the segmentation step, each space–time segment is considered as an individual motion pattern in the video.

After identifying the motion patterns, we need to model them and define a distance measure to compare motion patterns across videos. We compute the average preshape of the streaklines and a linear subspace representation for the spatio-temporal variation about the average preshape for each group of streaklines constituting a motion pattern. Given a set of videos for training and a test video, we compute the models for all the training data. The test data is matched to each of the training data by a combination of shape matching and subspace matching algorithm. The final match score is obtained by a time warping over the time segments. The test video is classified using an N-nearest neighbor classification.

### 1.5. Organization

The organization of the paper is as follows: We start by introducing streaklines which are the input features to our algorithm in Section 2. We define streaklines and explain how they can be computed from optical flow. In Section 3, we explain the process of identifying regions of the video which contain useful motion information using streaklines. This is done using the Helmholtz decomposition. Next, in Section 4, we demonstrate the extraction of motion patterns in video. These motion patterns are obtained by a process of time and space segmentation on the motion regions. In Section 5, we explain the modeling of individual motion patterns and our framework for recognition of activities using these models. In Section 6, we demonstrate the experiments we conducted to validate our approach.

## 2. Streakline representation of motion patterns

The first step of our algorithm is to represent a video using streaklines. Streaklines are a concept derived from fluid dynamics to represent a time-varying flow field. Suppose we inject a set of particles in the flow field continuously at certain points in the field, the path traced by these particles are called streaklines.

More formally, a streakline is defined as the locations of all particles that passed through a particular point over a period of time. It can be computed by initializing a set of particles at every time instant in the field and propagating them forward with time according to the flow field at that instant. This results in a set of paths, each belonging to one point of initialization. It can be shown that the streakline representation has advantages over other representations like streamlines and pathlines in being able to capture changes in the field as well as in smoothness of the resulting representation.

Given a video with $n$ pixels per frame for a duration of $N$ frames, we compute streaklines $s_1,\ldots,s_n$ where $s_i = [X_i, Y_i]^T$, $X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,N}]^T$, $Y_i = [y_{i,1}, y_{i,2}, \ldots, y_{i,N}]^T$, $s_i \in \mathbb{R}^{2N}$ for $i = 1,2,\ldots,n$. Every point on the streakline $(x_{i,t}, y_{i,t})$ corresponds to a particle $p$ initialized at pixel $i$ at time instant $t$.

The particle $p$ is initialized at the $i^{th}$ pixel of the frame at time instant $t$. For the subsequent frames, the particle is propagated from its old position $(x_{i,t}^{old}, y_{i,t}^{old})$ to its new position $(x_{i,t}^{new}, y_{i,t}^{new})$ using the particle advection.

$$\begin{aligned} x_{i,t}^{new} &= x_{i,t}^{old} + u\left(x_{i,t}^{old}, y_{i,t}^{old}\right) \\ y_{i,t}^{new} &= y_{i,t}^{old} + v\left(x_{i,t}^{old}, y_{i,t}^{old}\right) \end{aligned} \tag{1}$$

where $u(x,y)$ and $v(x,y)$ are the X and Y components of the instantaneous optical flow at position $(x,y)$.

Streaklines are ideally suited for motion analysis in video. Because they are computed over a larger interval of time as compared to optical flow, they are more robust to noise and easier to analyze than

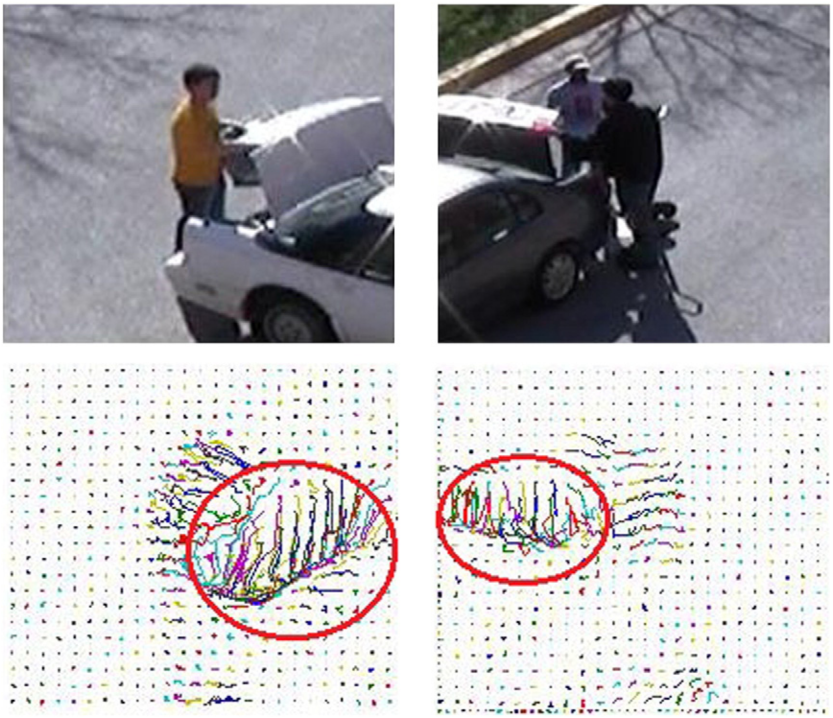

**Fig. 3.** The figure shows the streaklines for people opening a trunk in two videos. The circled region shows the similarity in the activity captured by the streaklines.

optical flow. They capture the pixel-wise spatio-temporal information in a video. Similar activities will result in similar streaklines, therefore modeling and comparison of streaklines can be used for activity classification. Fig. 3 illustrates the streaklines for similar activities being performed in different scenes. We notice that the streaklines look similar in the circled region.

## 3. Identification of motion regions

Motion in a video is often sparse. In most natural videos, motion is confined to small regions in the video. Since we compute streaklines at every pixel in each time frame, the size of the computed data is the same as the number of pixels in the video. To reduce the computational space and increase efficiency, we first need to reduce the size of the data. This can be done by identifying regions of meaningful motion in the video. We refer to such regions as “motion regions”.

There are several ways by which we could identify the motion regions in a video. For example, in Ref. [18], the authors perform segmentation on the whole volume and then eliminate small insignificant segments. However, this may not be computationally efficient, especially if the meaningful regions are small compared to the whole volume. Also, for our purpose, we do not need to identify every single streakline which represents motion. We are interested in those regions in the spatio-temporal volume which are most distinctive for the purpose of recognition. The Helmholtz decomposition has widely been used in the past to recognize distinctive points in a vector field. We utilize this concept derived from fluid dynamics to recognize motion regions.

The Helmholtz decomposition is a concept derived from physics, which states that any smooth field can be uniquely decomposed into an irrotational component and a solenoidal component. The extrema of these components are termed as critical points. In particular, the extrema of the irrotational field occur at regions of high divergence and convergence. Therefore, these would be the distinctive regions of the flow field that we are interested in modeling. Since optical flow is highly transient, we propose to use a flow field, which we call the “motion field” derived from the streaklines to compute the Helmholtz decomposition. We compute an aggregate flow by averaging the value of flow over a set of $k$ frames. This aggregate flow represents the average motion which each pixel has undergone. Next, we apply a smoothing function over this field to make it differentiable. The resultant field is known as the motion field **F**.

In this section, we will explain in detail, the computation of motion regions from the motion field using the Helmholtz decomposition.

### *3.1. Helmholtz decomposition of flow field*

The Helmholtz decomposition theorem states that any arbitrary vector field which is assumed to be differentiable can be decomposed into a curl free (irrotational) component and a divergence free (solenoidal) component [10], i.e.,

$$\mathbf{F} = \mathbf{F}_{sol} + \mathbf{F}_{irr}, \tag{2}$$

where **F** is the overall field, $\mathbf{F}_{sol}$ represents the solenoidal component and $\mathbf{F}_{irr}$ represents the irrotational component, $\mathbf{F} \in \mathbb{R}^{m \times n}$ where $m \times n$ is the video frame size.

Since $\mathbf{F}_{sol}$ is divergence free, we have $\nabla \mathbf{F}_{sol} = 0$. Similarly, since $\mathbf{F}_{irr}$ is curl free, we have $\nabla \times \mathbf{F}_{irr} = 0$. We can also define a scalar potential $\phi$ and a vector potential **A** such that

$$\mathbf{F} = -\nabla\phi + \nabla \times \mathbf{A}. \tag{3}$$

We see an illustration of the Helmholtz decomposition of a vector field in Fig. 4. We notice that the first component is purely a rotational field whereas the second component is purely divergent. Below, we will illustrate the extraction of regions of interest from the motion field using this decomposition.

#### *3.1.1. Computing the flow field components*

According to the Helmholtz decomposition, the motion field is composed of an irrotational and solenoidal component. We also mentioned that the motion field can be expressed in terms of a scalar potential ($\phi$) and a vector potential (**A**). We can obtain the irrotational and solenoidal components of the motion field from the scalar and vector potentials respectively. We will follow the technique described

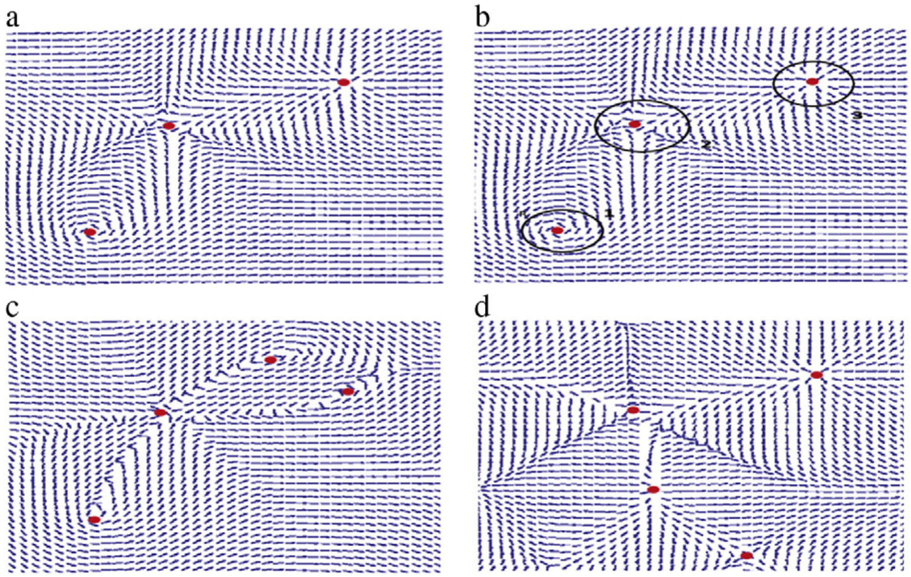

**Fig. 4.** Decomposition of a flow field: The figure shows a sample flow field and its decomposition into the irrotational and solenoidal components. The critical points are marked in red on each image. Figure a) shows the original flow field; Figure b) is the original flow field marked with regions containing critical points. We notice that the critical point in region 3 is an attracting focus and the critical points in regions 1 and 2 are repelling nodes; Figure c) represents the solenoidal component of original flow; and d) represents the irrotational component of original flow. We can see that the irrotational field has no rotational component and the solenoidal field is divergence free (purely rotational).

in Ref. [10] to solve for the scalar and vector potentials. The scalar potential can be obtained by projecting onto the curl-free component and solving the following variational problem:

$$\arg\min_{\phi} \int_{\Lambda} \|\mathbf{F} + \nabla\phi\|^2 dA, \Lambda \subset \Re^2 \quad (4)$$

where $\Lambda$ is the image domain under consideration and $A$ is the area. It can be shown that the solution to $\phi$ is obtained by solving the following Poisson equation [10]:

$$\nabla.\mathbf{F} = \nabla^2\phi \quad (5)$$

$$\mathbf{F} + \nabla\phi.\hat{n} = 0 \ \text{in}\partial\Lambda \quad (6)$$

where $\hat{n}$ is the unit outward normal to the boundary $\partial\Lambda$.

A similar formulation can be derived for the vector potential. The solenoidal component can be solved using the following variational problem.

$$\arg\min_{\mathbf{A}} \int_{\Omega} \|\mathbf{F} - (\nabla \times \mathbf{A})\|^2 dA, \Omega \subset \Re^3, \quad (7)$$

the optimum solution of which is obtained by the following PDE formulation:

$$\nabla \times \mathbf{F} = \nabla \times \nabla \times \mathbf{A} = \nabla(\nabla.\mathbf{A}) - \nabla^2\mathbf{A} \quad (8)$$

$$\mathbf{F} - (\nabla \times \mathbf{A}) \times \hat{n} = 0 \ \text{in}\partial\Omega. \quad (9)$$

Here $\hat{n}$ is the unit outward normal to the boundary $\partial\Omega$. Since we have the curl $(\nabla\times)$ to be an operator in three dimensions, for an arbitrary **A** we need to extend the two dimensional field **F** to 3D by setting the z-component to zero.

On solving the above equations, we obtain the scalar potential $\phi$ and the vector potential **A**. The irrotational and solenoidal components of the flow field are accordingly obtained as

$$\mathbf{F}_{irr} = \nabla\phi \quad (10)$$

$$\mathbf{F}_{sol} = \nabla \times \mathbf{A}. \quad (11)$$

### 3.2. Motion regions using the Helmholtz decomposition

The irrotational component of the Helmholtz decomposition carries useful information about the sources and sinks of the motion field. These sources and sinks are a result of motion in a video, therefore they can be used to identify regions of motion in the video. The sources and sinks are also known as critical points. A point $C(x_0,y_0)$ is defined as a singular/critical point of the vector field if $C(x_0,y_0) = (0,0)^T = 0$ and $C_1(x,y) \neq 0$ for any other point $C_1$ with coordinates $x \neq x_0, y \neq y_0$ in the neighborhood of $(x_0,y_0)$.

Consider a point $\mathbf{v}(x,y)$in the irrotational field in 2D given by

$$\mathbf{v}(x,y) = \begin{bmatrix} u(x,y) \\ v(x,y) \end{bmatrix}.$$

The Jacobian matrix of the irrotational field at a point $(x,y)$ on the field denoted by $J_v$ is given by

$$J_{\mathbf{v}} = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix}$$

where $u_x$ and $v_x$ are the partial derivatives of $u$ and $v$ with respect to $x$ and $u_y$ and $v_y$ are the partial derivatives of $u$ and $v$ with respect to $y$. The determinant of the Jacobian at $(x,y)$ is denoted as $|J_{\mathbf{v}}|$. The critical points are identified by finding those points in the field where $u$ and $v$ are zero, but $|J_{\mathbf{v}}| \neq 0$. The critical points of the vector field and its components from Helmholtz decomposition are marked in Fig. 4.

As mentioned before, the critical points of the irrotational field occur in regions of high convergence and divergence in the field. Intuitively, these would be the most distinctive regions of the motion field, and therefore, we would want to model the streaklines which correspond to these regions. Therefore, we define a motion region as a set of streaklines which pass within a small distance of a critical point. Here, we set the distance as 5 pixels for a frame size of 150×200, however, this distance can be modified based on the resolution of the video. An example of the motion regions identified using critical points is shown in Fig. 5.

## 4. Segmentation of motion patterns

The motion information in a video is contained in the form of motion patterns. Each video could contain multiple motion patterns, each said to correspond to an “event”. These motion patterns vary in time durations as well as in space. Activity recognition in such videos requires modeling of the motion patterns as well as studying the spatio-temporal relationships between them. We perform activity recognition in two steps — identification of motion patterns and modeling of motion patterns.

An activity in a video can be composed of one or more motion patterns. Since we are dealing with complex, real-world scenarios, there could also be motion patterns which are introduced by background clutter or noise. To make our algorithm robust to these factors, we do not place any assumptions on the number or locations of motion patterns in the scene. Our next task therefore, is to identify motion

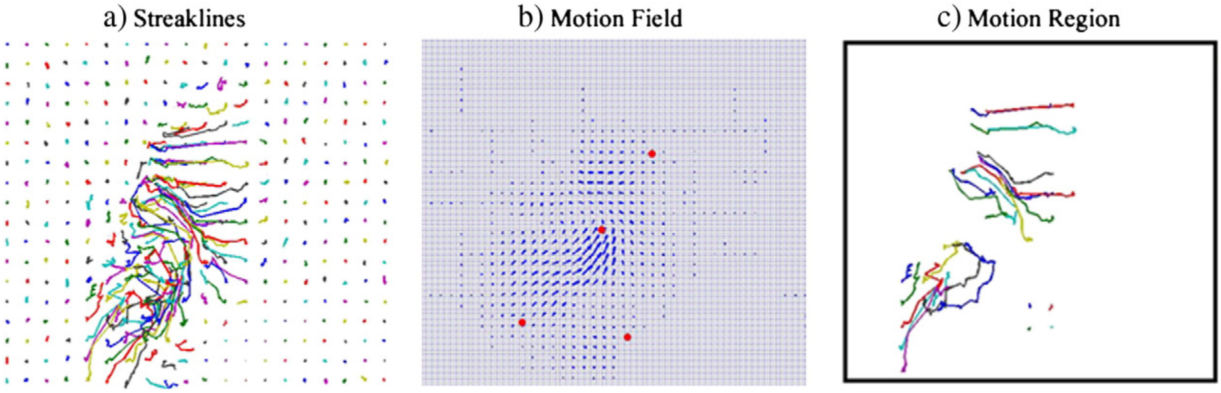

Fig. 5. The figure shows the extraction of motion regions from streaklines. Figure a) shows the streaklines of the action “open trunk”. Figure b) shows the corresponding motion field. The critical points of the motion field are marked in red. The streaklines extracted using these critical points are shown in Figure c) and constitute the motion regions of the video.

patterns. Because we represent a video as a group of streaklines, the task of identification of motion pattern is performed by a segmentation of streaklines. We segment the streaklines both in time and space domain.

### 4.1. Time segmentation of streaklines

We propose that the critical points extracted using Helmholtz decomposition can also be used for time segmentation of streaklines. This is based on the observation that whenever there is not much change in the motion pattern from one time instant to another, the location of the critical points and their characteristics do not change much. On the other hand, when a new motion pattern originates, a new critical point emerges, or when an existing motion pattern ends, a critical point disappears. Therefore, by associating the critical points from one frame to the next, we can identify the start and end points of motion patterns. Each critical point is associated with a motion region. Therefore, a motion region exists in the duration in which the corresponding critical point is observed. To associate critical points from one frame to the next, we use the following distance measure as described in Ref. [25].

Every critical point $C(x,y)=(u(x,y),v(x,y))^T$ is mapped to a circular coordinate system $(\gamma(x,y),r(x,y))$ given by

$$cos\gamma = \frac{u_x+v_y}{sqrt\left(u_x+v_y\right)^2+\left(v_x-u_y\right)^2} \tag{12}$$

$$sin\gamma = \frac{v_x-u_y}{sqrt\left(u_x+v_y\right)^2+\left(v_x-u_y\right)^2} \tag{13}$$

$$r = \frac{1}{2}+\frac{u_xv_y-v_xu_y}{u_x^2+u_y^2+v_x^2+v_y^2} \tag{14}$$

where $u_x,u_y,v_x,v_y$ are elements of the Jacobian of the critical point C denoted by $J_C$. The similarity measure between two critical points is given by the Euclidean distance between them in the $(\gamma,r)$ plane as defined in Eq. (15):

$$d_c\left(C_i,C_j\right) = \sqrt{r_1^2+r_2^2-2r_1r_2cos(\gamma_1-\gamma_2)}. \tag{15}$$

Therefore, we compute the critical points in every frame and compute the distance between critical points from one frame to the next using Eq. (15). It is seen that the critical points that arise due to the same event in adjacent frames have a very small distance and can therefore be associated. Whenever a new critical point arises, a new event is said to begin and when the critical point disappears, an event ends. The streaklines that belong to the motion region associated with the critical point in the time interval in which a critical point is observed is said to constitute the time segment. Fig. 6 shows some examples of time segmentation using our algorithm.

### 4.2. Segmentation of streaklines in space

Each video segment could be made up of more than one motion pattern. Each motion pattern could correspond to one object in a scene, or a part of an object in the scene. We therefore, perform a clustering of streaklines in space such that the streaklines in each individual cluster exhibit similar motion. Each cluster is said to belong to one motion pattern or event in the video. To perform a segmentation of the motion

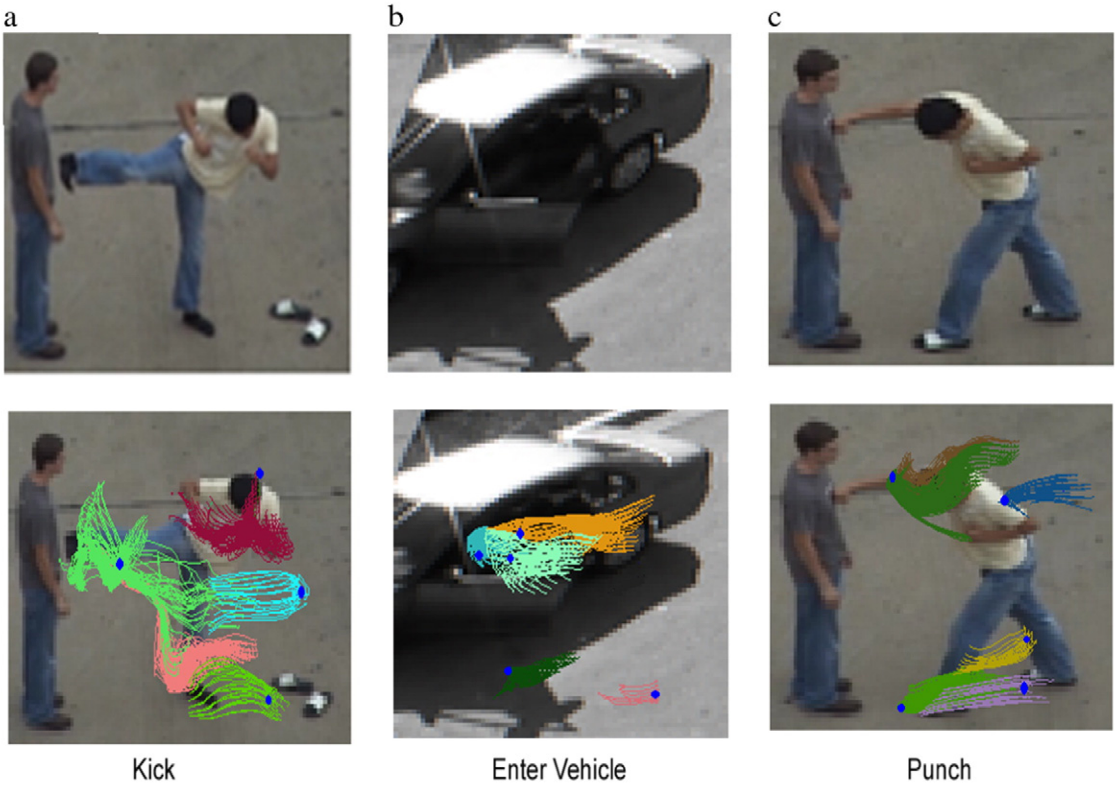

**Fig. 6.** Examples of time segmentation of streaklines using the Helmholtz decomposition. The first row shows a sample frame and the second row displays the time segmented streaklines. Each segment is marked in a different color. The critical points are marked in blue.

patterns, we will first transform the streaklines into a shape space. The shape representation of streaklines is given below.

### 4.2.1. Shape representation of streaklines

Consider a streakline $s \in \mathbb{R}^{2k}$ in a time segment of length $k$.

$$s = [x_1 \quad x_2 \quad . \quad . \quad x_k \quad y_1 \quad y_2 \quad . \quad . \quad y_k]^T.$$

Next, we remove the scaling and translation from $s$ to obtain a normalized vector $c$. This is done by subtracting the mean from $s$ and scaling to unit norm.

$$c = \frac{Ps}{\|Ps\|}, \tag{16}$$

where $P = I_{2k} - \frac{1}{k} I_{D_{2k}}$, $I_{2k}$ being the $2k \times 2k$ identity matrix and $I_{D_{2k}}$ is the $2k \times 2k$ matrix given by $I_{D_{2k}} = \begin{bmatrix} 1_{k\times k} & 0_{k\times k} \\ 0_{k\times k} & 1_{k\times k} \end{bmatrix}$, where $1_{k\times k}$ is a $k \times k$ matrix of ones. This normalized vector is independent of translation and scale and is called a preshape vector of a collection of points [7].

### 4.2.2. Extraction of motion patterns

Suppose a video is made up of $p$ underlying motion patterns $[M^1,, M^2,, \ldots,, M^p]$. Let each motion pattern $M^i$, $i\{1,,2,,..,,p\}$ contain $n^i$ pre-shape vectors $[c_1^i,, c_2^i,, s_3^i,, \ldots,, c_{n^i}{}^i]$, where $c_j^i \in \mathbb{R}^{2k\times 1}$. Let $\left[\tilde{}M^1,, \tilde{}M^2, \ldots,, \tilde{}M^p\right]$ be our estimates of the motion pattern. Estimation of the motion patterns can be performed in a clustering framework. Here, we use the average preshape of the motion pattern as the representative model for clustering. The average preshape of a motion pattern $M^i$ is given by

$$\bar{c}^i = \sum_{j=1}^{n^i} c_j^i. \tag{17}$$

The projection error between a preshape $c$ and an average preshape $\bar{c}^i$ of motion pattern $M^i$ is calculated as the square of the Euclidean norm of their distance, $\left\|c - \bar{c}^i\right\|^2$. Therefore, a preshape $c \in M^i$ if

$$||c - \bar{c}^i||^2 \leq \left\|c - \bar{c}^{i'}\right\|^2, \forall i' \neq i \tag{18}$$

where we are using the Euclidean norm as an approximation of the actual pre-shape vector norm (Procrustes distance). The above clustering problem can be solved using a standard k-means clustering framework.

Because we do not want to assume the number of motion patterns in the video, we set a threshold on the model residue $\varepsilon_{thresh}$ and compute $p$ such that

$$\sum_{i=1}^{p} \sum_{j=1}^{n^i} \left\|c_j^i - \bar{c}^i\right\|^2 \leq \varepsilon_{\text{thresh}}. \tag{19}$$

Some examples of space segmentation are shown in Fig. 7.

## 5. Activity modeling and recognition

In the previous section, we computed a set of motion patterns as well as the average preshape of each motion pattern. This average preshape provides us with the mean path traced by the object or part of the object which is involved in the event. The average preshape $\bar{c}^i$ of motion pattern $M^i$ therefore, can be used to model the motion pattern. Apart from the average preshape, the streaklines can be characterized by their spatio-temporal evolution. To make this evolution independent of its location, we model the evolution as the variation of the preshapes of a motion pattern about the average preshape. Each motion pattern $M^i$ contains $n^i$ preshapes of length $k^i$. The spatio-temporal evolution of preshapes can be modeled by examining the linear subspaces along which there is maximum variation in the data. This can be achieved by a subspace analysis of the data. The task of activity classification requires a comparison between the average preshape as well as the similarity between their subspaces. In this section, we will explain these steps in detail.

### 5.1. Comparison of average preshapes

Consider two preshape vectors $c^i$ and $c^j$ of motion patterns $M^i$ and $M^j$. To compare $c^i$ and $c^j$, we first need to ensure that they are of the same length. This is done by resampling the preshape vectors to a length l. Here, $l$ can be a constant or a function of the duration of the time segment. The distance between the resampled preshape vectors can be measured by the full Procrustes distance [7] which is the Euclidean distance betweenthe Procrustes fit of the preshapes $\bar{c}^i$ and $\bar{c}^j$. The Procrustes fit $(\beta, \theta, (a + jb)$ is chosen to minimize the distance given by

$$d_s^{(i_1, i_2)} = \left\|\bar{c}^i - \bar{c}^j \beta exp^{j\theta} - (a + jb) 1_l\right\|, \tag{20}$$

where $\beta$ is the scale, $\theta$ is the rotation and $(a + jb)$ is the translation, $1_l$ is the $l$ dimensional column vector of ones. Since the preshapes have already been normalized, the estimated scale $\beta \approx 1$ and the estimated translation $(a + jb) \approx 0$. The rotation will be obtained as $\theta = arg(c^{i\mathrm{T}} c^j)$.

### 5.2. Subspace analysis

Let the preshapes constituting a motion pattern $M^i$ be $C^i = c_j^i$, $j = 1..n^i$, where $n^i$ is the number of streaklines in $M^i$. Since the average preshape captures the average motion in $M^i$, we wish to model the spatio-temporal variation in the motion pattern $M^i$ using subspace analysis. We use the preshape vector $c^i$ to compute a linear subspace representation for $M^i$.

A linear subspace representation for $C^i$ can be computed by a principal component analysis of the covariance matrix of $C^i$ given by

$$R^i = \frac{1}{n^i} \sum_{j=1}^{n^i} \left(c_j^i - \bar{c}^i\right) \left(c_j^i - \bar{c}^i\right)^T \tag{21}$$

where $R^i$ is the covariance matrix. We choose the first $r$ eigenvectors $V_1^i, V_2^i \ldots V_r^i$ of $R^i$ as the orthogonal vectors for the low dimensional representation of $C^i$. The value of $r$ is chosen experimentally.

The similarity between the subspace representation of motion patterns $M^i$ and $M^j$ is given as the sum of the $r$ principal angles between the corresponding subspaces [11], i.e.

$$d_\theta^{(i,j)} = \sum_{j=1}^{n^i} \arccos\left(V_m^{iT} V_m^j\right). \tag{22}$$

### 5.3. Overall distance computation

The total distance between a training and test video is computed as follows: For the training sequences, it is assumed that the motion patterns pertaining to the training activity have been identified and modeled. For the test sequence, there could be a different number of motion patterns. For every motion pattern in the training data,

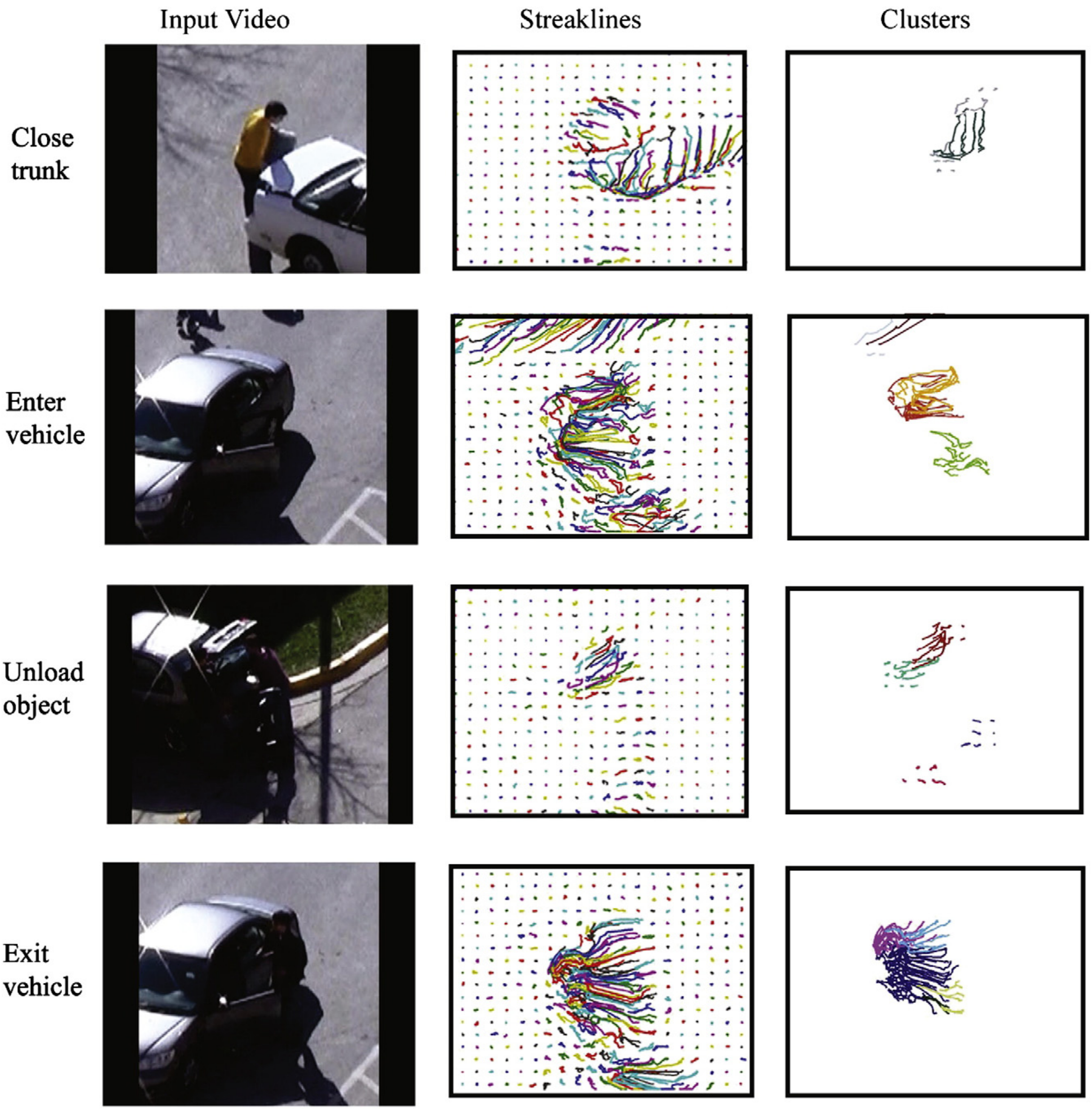

**Fig. 7.** The figure shows the streaklines and the clusters for activities in the VIRAT dataset. The clusters are marked with different colors.

we find the closest motion pattern in the test data. This distance is computed as follows:

Consider a training video with $n_r$ motion patterns and a test video with $n_T$ motion patterns. The distance between a motion pattern $M^i$ in the training video and $M^j$ in the test video is given by the weighted average

$$d(i,j) = w_1 d_s^{(i,j)} + w_2 d_\theta^{(i,j)}, \tag{23}$$

where $d_s$ and $d_\theta$ are the shape and subspace distances given in Eqs. (20) and (21). $w_1$ and $w_2$ are the weights which are set such that the overall distance lies in the range of 0–1. These weights are determined using the training data. For each motion pattern $M^i$, we choose the best match as that motion pattern in the test video which has the least distance $D^i$. The total distance between a training and a test video is given by the sum of the best match distances for all motion patterns, i.e.,

$$D = \sum_{i=1}^{n_r} D^i. \tag{24}$$

We use a $k$-nearest neighbor classifier for recognition of activities. i.e. considering the $k$ closest training clips, the activity is classified as that category to which most of the $k$ neighbors correspond.

Therefore, the steps in recognition of activities using our algorithm are as follows:

1. For each training video $v$, compute the motion patterns $M^1, M^2 \ldots M^{p_v}$. Model each motion pattern $M^i$ using the average preshape $c^i$ and $r$ eigenvectors $V_1^i, V_2^i, \ldots V_r^i$.

2. For the given test video $t$, compute the motion patterns and the model for each motion pattern. The distance between every motion pattern $M^i$ in the training video and $M^j$ in the test video is computed using Eq. (23).
3. For each motion pattern $M^i$, the least distance $D^i$ with a test motion pattern is chosen as the best match.
4. The total distance between two videos is given by the sum of distances of the best match between their motion patterns.
5. The distance between every training video and the test video is computed. The activity in the test video is classified using a $k$-nearest neighbor classifier.

## 6. Experiments

To validate our approach, we perform experiments on two publicly available complex datasets. Each of these datasets involves outdoor scenes and multiple actors interacting in the presence of noise and background clutter.

### 6.1. Dataset

The first set of experiments was conducted on the UT Interaction dataset [23]. This dataset consists of high resolution video of two actors performing actions such as handshake, kicking, hugging, pushing, punching and pointing. Each task is performed by 10 actors in outdoor environments. Each video is of a duration of approximately 3 s. Often there are people walking or performing other activities in the background, causing background clutter. We test our method on this dataset to validate the use of our method for analysis of articulated motion. We demonstrate and compare our results with three previous methods which use the same dataset.

The second set of experiments was conducted on the VIRAT dataset. The VIRAT public dataset [19] contains activities involving people–people and people–vehicle interactions. The people–vehicle activities include person opening and closing the trunk, person entering and exiting a vehicle and person loading and unloading objects from the vehicle. Often, there are other people moving in the scene causing background clutter. There is variation in the scale as well as orientation of objects in the dataset. Often, there are shadows or occlusions leading to a high amount of noise in the scene.

As mentioned before, the critical points of the irrotational field occur in regions of high convergence and divergence in the field. Intuitively, these would be the most distinctive regions of the motion field, and therefore, we would want to model the streaklines which correspond to these regions. Therefore, we define a motion region as a set of streaklines which pass within a small distance of a critical point. Motion patterns are identified by time and space segmentation. We use a simple heuristic to identify motions that belong to "other" categories. If the distance between a motion pattern in the test video and every motion pattern in the training data as computed using Eq. (23) exceeds a pre-defined minimum, the pattern is marked as belonging to some "other" category and removed from consideration for total distance computation.

### 6.2. Results on UT Interaction data

Our method performed well on the UT Interaction data. The videos are of different lengths and the activities are performed from two different viewpoints. We computed streaklines over the entire video. The motion regions were found to be concentrated around the limbs of the persons involved due to the nature of the activities. It was found that most of the activities were composed of two to three events. For example, the “pointing” action is composed of the person raising his hand and then lowering it. Similarly, “shaking hands” is composed of two people

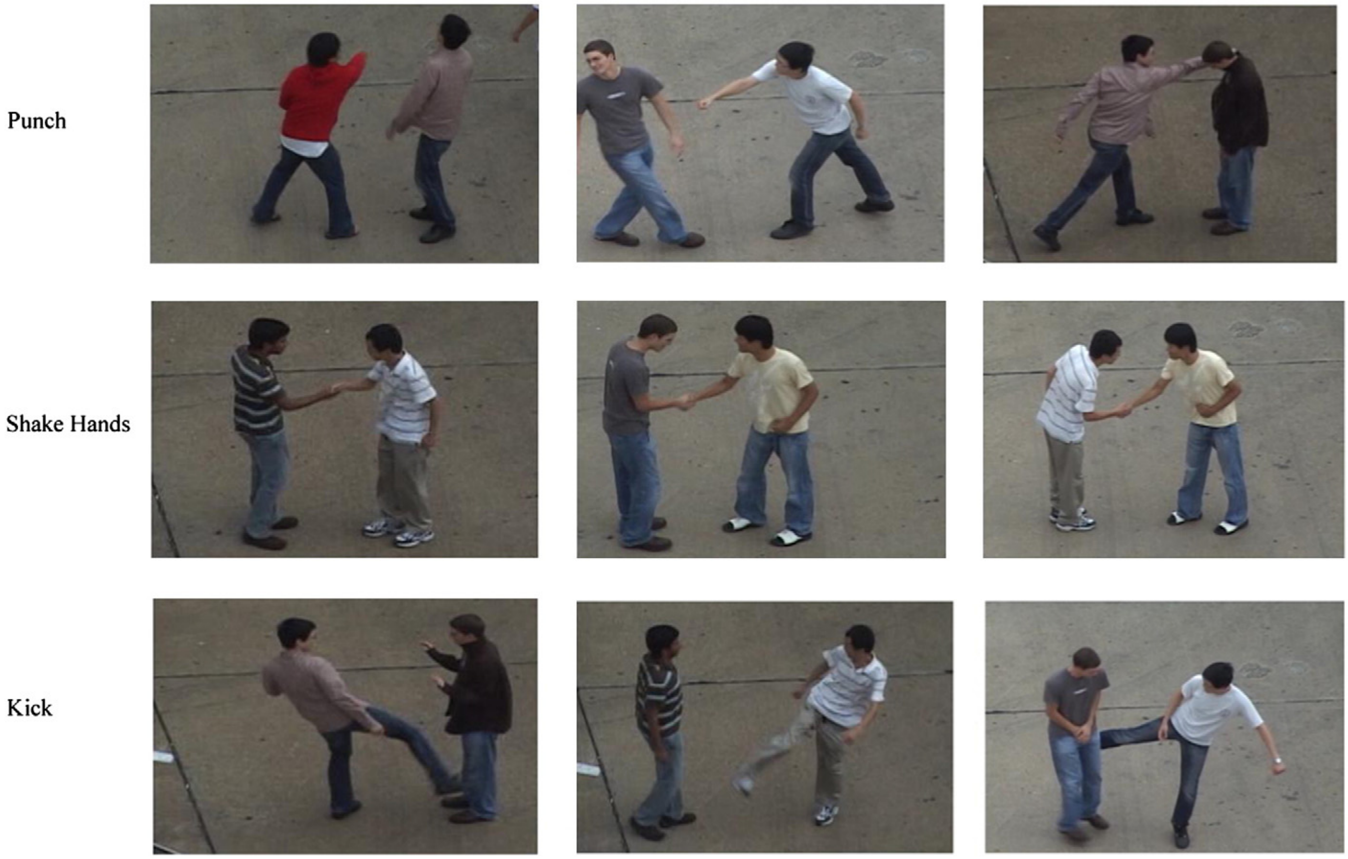

**Fig. 8.** Examples of retrieved results for the UT Interaction dataset.

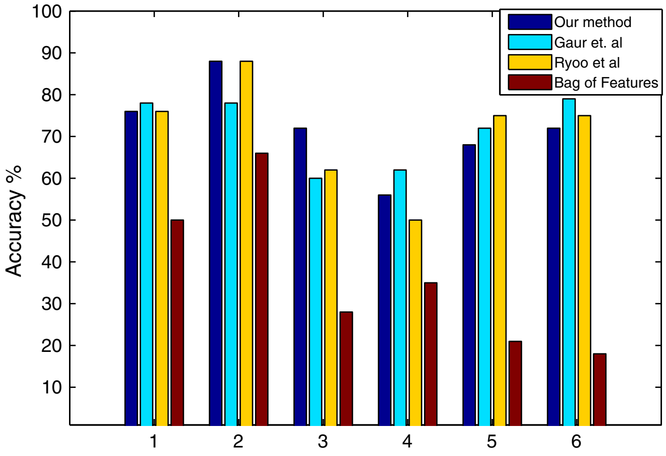

**Fig. 9.** The figure shows the accuracy of recognition using the UT Interaction data and comparison with previous methods. The activities are: — 1 — shake hands, 2 — hug, 3 — point, 4 — punch, 5 — kick, 6 — push.

approaching each other, shaking their hands and dispersing. The spatial segmentation separated out the articulated motion in the video. Some examples of retrieved results are shown in Fig. 8.

We use a leave one out strategy for activity recognition. 9 out of 10 sequences were used for training and the remaining for testing. It was found that the performance of our method on the UT Interaction dataset was similar to the other state of the art methods like in Ref. [22]. We achieved an overall recognition accuracy of 72.0%, while Ref. [22] achieves an accuracy of 70.8% and Ref. [9] achieves an accuracy of 70.6%. The method worked well on activities like hug and shake hands where the motion patterns were highly distinguishable. The performance for activities like punch and kick were slightly lower since the events were similar to each other. The comparison of our method to other previous STIP-based approaches is shown in Fig. 9. It can be seen that on an average, our method performs as well as other previous STIP-based methods and better than Bag of Features. The advantage of our method as compared to these previous methods is that the spatio-temporal relationships in a STIP-based method have to be explicitly modeled using graphs or other complex structures. Therefore, as the activities get more complex, the graph gets more complex and the computation increases exponentially. Whereas in our method, the spatio-temporal relationships are embedded in the streaklines, therefore the computational cost is linear with respect to the number of streaklines in a motion pattern. Moreover, we provided a unified bottom-up analysis framework starting from the low-level features (streaklines), segmenting them into individual regions of interest, identifying events and modeling activities as a combination of events. This is unlike other competing methods which consider the entire volume as a set of features and models them, or requires different tools to do the low level processing (which are not dealt with in detail in those papers). This has been given in more detail in Section 6.4.

### 6.3. Results on the VIRAT data

The experiments conducted on the VIRAT data test the robustness of our approach to the presence of clutter and variations in scale. The generation of normalized preshape vectors from streaklines handles

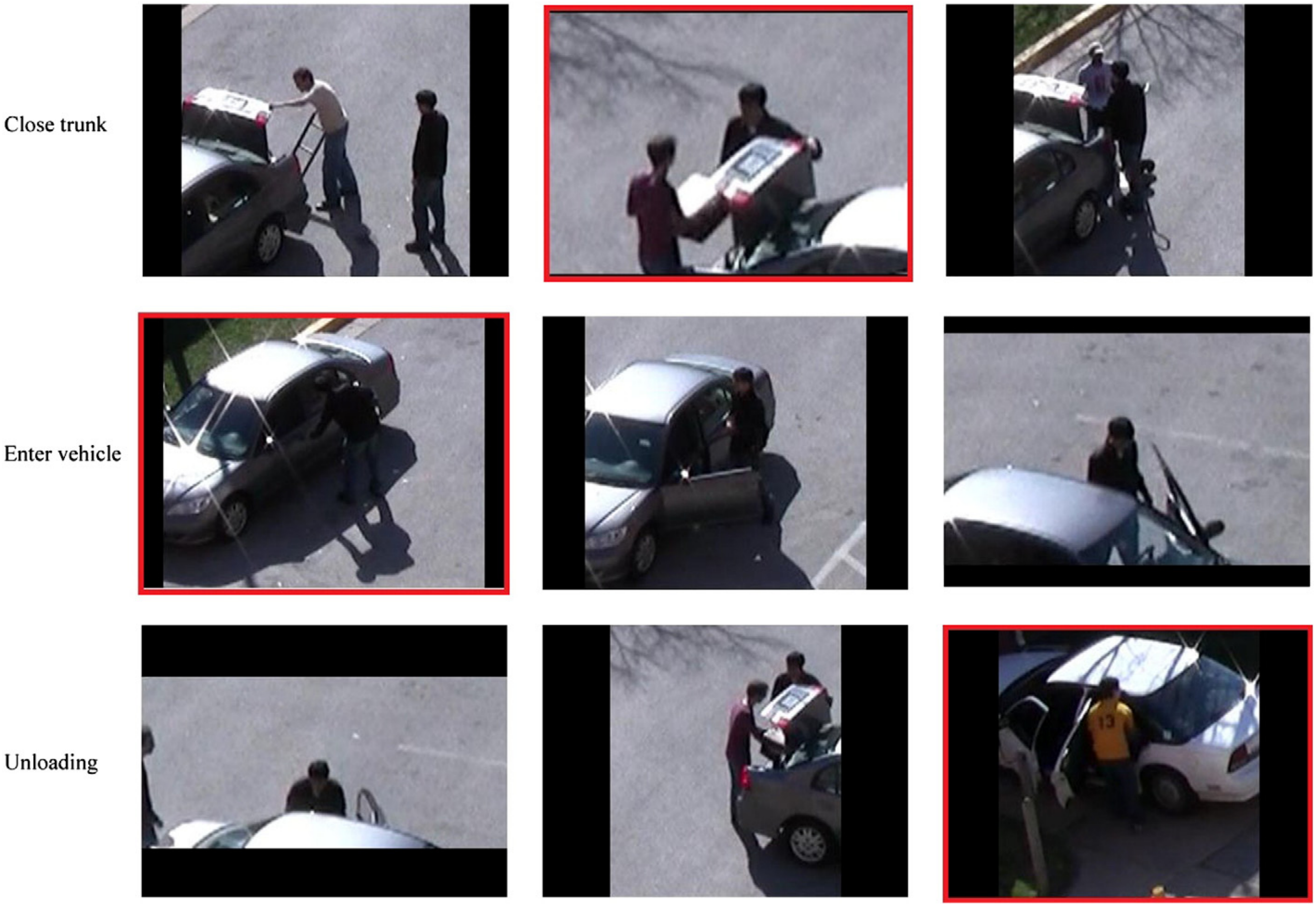

**Fig. 10.** Example of results for the VIRAT dataset showing some true positives and false negatives for actions close trunk, enter vehicle and unloading. The false negatives are marked in red.

the difference in scale. The rotation invariant shape comparison handles the changes in viewpoint to some extent. Activities like closing and opening trunk consisted of one event, while the other activities often consisted of two or three events. For example, entering a vehicle is composed of opening the door, sitting inside the vehicle and closing the door. The space segmentation separated individual objects in the scene and helped in the elimination of background clutter.

A leave one out strategy in conjunction with the N-nearest neighbor was used for classification. The results were compared to that using Ref. [9]. The results are shown in Fig. 11. It can be seen that our results are comparable to other state of the art methods here also. However, as mentioned before, our system presents an entire end-to-end pipeline for image analysis and is computationally efficient as discussed in Section 6.4. The method performed well in recognizing multi-person activities like people walking together and people approaching each other. The accuracy of recognition for loading and unloading was lower since the events are similar to those in entering and exiting vehicles. Some examples of videos retrieved are show in Fig. 10. The erroneous results are marked in red. In the first row, it is seen that the second example contains a person carrying an object, and was confused with unloading. This example failed to be retrieved. Similarly, shadows and occlusions have caused false negatives in the second and third rows.

### 6.4. Analysis of the results

As seen from Figs. 9 and 11, the performance of our method is comparable to that of other state of the art methods. However, the advantage of our method is that, unlike previous methods which try to analyze activities at the feature level, we propose a global approach to activity recognition. This facilitates a bottom-up analysis of a video, where we begin with the streaklines over the entire video, then compute individual motion patterns, and finally model and compare these motion patterns. Therefore, our method provides an end-to-end system which computes a set of features, segments out different events and defines a distance measure over them. This is unlike other methods like in Ref. [22], where segmentation is not an integral part of the method and has to be performed separately before the activity modeling and recognition can be done.

There is also the advantage of computational efficiency in the modeling and comparison using our algorithm. For a STIP-based method, for example in Ref. [9], a graph is matched for every time segment in the test video to every time segment in the training video. The time complexity for matching a graph with $V$ nodes and $E$ edges is known to be $O(V^2E)$. Since the number of edges for a completely connected graph with $V$ nodes is of the order of $V^2$, we can expect the time complexity of algorithms like Ref. [9] to increase exponentially with the number of feature points/nodes. In comparison, consider a motion pattern with $N$ streaklines. Our method computes the mean shape vector for each motion pattern. This requires $O(N)$ operations. Comparison of mean shape vectors using the Procrustes distance is a $O(1)$ operation. It can be shown that the subspace analysis to compute the first $k$ eigenvectors of N streaklines of length $p$ is $O(Nkp)$. Therefore, for a motion pattern with $N$ streaklines, the overall computational cost of modeling and comparison is proportional to $O(N)$, i.e. the complexity increases linearly with the number of streaklines in a motion pattern.

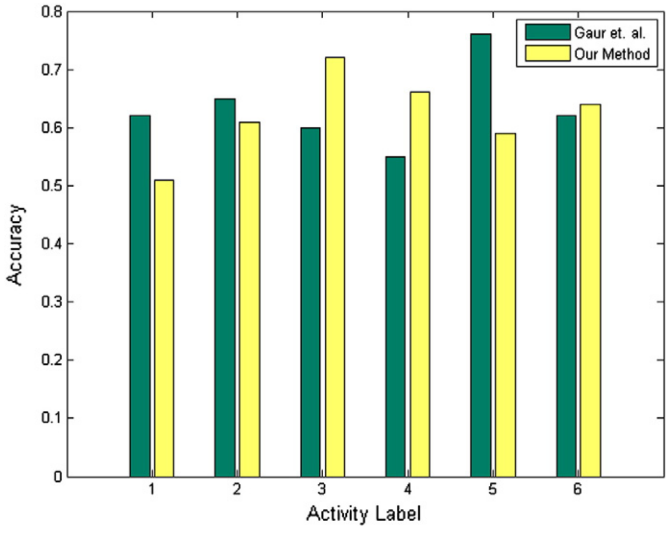

**Fig. 11.** The figure shows the recognition accuracy for the VIRAT dataset. The activities are: 1 — loading, 2 — unloading, 3 — open trunk, 4 — close trunk, 5 — enter vehicle, 6 — exit vehicle.

## 7. Conclusion

In this work, we proposed a flow-based end-to-end system for activity recognition. We modeled activities as a collection of motion patterns. We demonstrated the use of streaklines to represent and model these motion patterns. The Helmholtz decomposition was used to identify regions of useful motion which were analyzed further. The segmentation of streaklines can be used to separate motion patterns and model them individually. We also showed a method for computing the similarity between two videos using these models. Experiments were conducted on multi-object scenes with a high amount of noise and clutter. In future, we wish to extend the scheme to the analysis of activities which span a wider area and contain more complex interactions.

## References

[1] A. Bissacco, A. Chiuso, Y. Ma, S. Soatto, Recognition of human gaits, Comput. Vision Pattern Recognit. 2 (2005) II-52–II-57.

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as Space-Time Shapes, in: International Conference on Computer Vision, 2011.

[3] A. Bobick, J. Davis, The recognition of human movement using temporal templates, IEEE Trans. Pattern Anal. Mach. Intell. 23 (3) (2001) 257–267.

[4] W. Brendel, S. Todorovic, Learning Spatiotemporal Graphs of Human Activities, in: International Conference on Computer Vision, 2011.

[5] R. Chaudhary, A. Ravichandran, G. Hager, R. Vidal, Histograms of oriented optical flow and Binet–Cauchy kernels on nonlinear dynamical systems for the recognition of human actions, in: Computer Vision and Pattern Recognition, 2009.

[6] G. Doretto, A. Chiuso, Y. Wu, S. Soatto, Dynamic Textures, , 2003.

[7] I. Dryden, K. Mardia, Statistical Shape Analysis, John Wiley and Sons, 1998.

[8] A. Efros, A. Berg, G. Mori, J. Malik, Recognizing Action at a Distance, in: International Conference of Computer Vision, 2003.

[9] U. Gaur, Y. Zhu, B. Song, A. Roy-Chowdhury, String of Feature Graphs Analysis of Complex Activities, in: International Conference on Computer Vision, 2011.

[10] P. Ghosh, L. Bertelli, B. Sumengen, B. Manjunath, A nonconservative flow field for robust variational image segmentation, IEEE Trans. Image Process. 19 (2) (2010) 478–490.

[11] G.H. Golub, C.V. Loan, Matrix Computations, The Johns Hopkins University Press, 1996.

[12] M. Hu, S. Ali, M. Shah, Detecting Global Motion Patterns in Complex Videos, in: International Conference on Pattern Recognition, 2008.

[13] Y. Ke, R. Sukthankar, M. Hebert, Efficient Visual Event Detection Using Volumetric Features, in: International Conference on Computer Vision, 2005.

[14] D. Kuettel, M. Breitenstein, L.V. Gool, V. Ferrari, What's going on? Discovering spatio-temporal dependencies in dynamic scenes, in: Computer Vision and Pattern Recognition, 2010.

[15] I. Laptev, T. Lindeberg, Space–Time Interest Points, in: International Conference on Computer Vision, 2003.

[16] I. Laptev, T. Lindeberg, Local Descriptors for Spatio-Temporal Recognition, in: First International Workshop on Spatial Coherence for Visual Motion Analysis, 2004.

[17] M. Lee, R. Nevatia, Human pose tracking in monocular sequence using multilevel structured models, IEEE Trans. Pattern Anal. Mach. Intell. 31 (1) (2009) 27–38.

[18] R. Mehran, B. Moore, M. Shah, A Streakline Representation of Flow in Crowded Scenes, in: European Conference on Computer Vision, 2010.

[19] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C. Chen, J.T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, M. Desai, A large-scale benchmark dataset for event recognition in surveillance video, in: Computer Vision and Pattern Recognition, 2011.

[20] S. Park, J. Aggarwal, Recognition of Two-Person Interactions Using a Hierarchical Bayesian Network, in: ACM SIGMM International Workshop on Video Surveillance, 2003.

[21] M. Ryoo, J. Aggarwal, Recognition of composite human activities through context-free grammar based representation, in: Computer Vision and Pattern Recognition, 2006.

[22] M. Ryoo, J. Aggarwal, Spatio-Temporal Relationship Match: Video Structure Comparison for Recognition of Complex Human Activities, in: International Conference on Computer Vision, 2009.

[23] M. Ryoo, C. Chen, J. Aggarwal, A. Roy-Chowdhury, An Overview of Contest on Semantic Description of Human Activities (SDHA) 2010, in: International Conference on Pattern Recognition, 2010.

[24] C. Schuldt, I. Laptev, B. Caputo, Recognizing Human Actions: A Local SVM Approach, in: International Conference on Pattern Recognition, 2004.

[25] H. Theisel, T. Weinkauf, Vector Field Metrics Based on Distance Measures of First Order Critical Points, in: International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision, 2002.

[26] P. Turaga, R. Chellappa, V. Subrahmanian, O. Udrea, Machine recognition of human activities: a survey, IEEE Trans. Circuits Syst. Video Technol. 18 (11) (2008) 1473–1488.

[27] N. Vaswani, A. Roy-Chowdhury, R. Chellappa, Activity recognition using the dynamics of the configuration of interacting objects, in: Computer Vision and Pattern Recognition, 2003.

[28] A. Veeraraghavan, A.K. Roy-chowdhury, R. Chellappa, Matching shape sequences in video with applications in human movement analysis, IEEE Trans. Pattern Anal. Mach. Intell. 27 (12) (2005) 1896–1909.

[29] T. Wada, T. Matsuyama, Multiobject behavior recognition by event driven selective attention method, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 873–887.