Image and Vision Computing 27 (2009) 1814–1825

Contents lists available at ScienceDirect

Image and Vision Computing

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

Sparse B-spline polynomial descriptors for human activity recognition

Antonios Oikonomopoulos [a,*], Maja Pantic [a,b], Ioannis Patras [c]

[a] Department of Computing, Imperial College London, UK
[b] Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, The Netherlands
[c] Department of Electronic Engineering, Queen Mary University of London, UK

ARTICLE INFO

Article history:
Received 19 May 2008
Received in revised form 10 April 2009
Accepted 16 May 2009

Keywords:
Optical flow
Median filter
Spatiotemporal salient points
B-splines
Gentleboost
Relevance Vector Machine

ABSTRACT

The extraction and quantization of local image and video descriptors for the subsequent creation of visual codebooks is a technique that has proved very effective for image and video retrieval applications. In this paper we build on this concept and propose a new set of visual descriptors that provide a local space-time description of the visual activity. The proposed descriptors are extracted at spatiotemporal salient points detected on the estimated optical flow field for a given image sequence and are based on geometrical properties of three-dimensional piecewise polynomials, namely B-splines. The latter are fitted on the spatiotemporal locations of salient points that fall within a given spatiotemporal neighborhood. Our descriptors are invariant in translation and scaling in space-time. The latter is ensured by coupling the neighborhood dimensions to the scale at which the corresponding spatiotemporal salient points are detected. In addition, in order to provide robustness against camera motion (e.g. global translation due to camera panning) we subtract the motion component that is estimated by applying local median filters on the optical flow field. The descriptors that are extracted across the whole dataset are clustered in order to create a codebook of 'visual verbs', where each verb corresponds to a cluster center. We use the resulting codebook in a 'bag of verbs' approach in order to represent the motion of the subjects within small temporal windows. Finally, we use a boosting algorithm in order to select the most discriminative temporal windows of each class and Relevance Vector Machines (RVM) for classification. The presented results using three different databases of human actions verify the effectiveness of our method.

1. Introduction

Due to its practical importance for a wide range of vision-related applications like video retrieval, surveillance, vision-based interfaces, and human–computer interaction, vision-based analysis of human motion is nowadays one of the most active fields of computer vision. One of the main goals in this field is to efficiently represent an activity captured by a camera and to accurately classify it, that is, assign it into one or more known action categories.

Given a video sequence, humans are usually able to deduce quickly and easily information about its content. In particular, they can discriminate relatively easily between a wide range of activities, even if they have observed each of the activities only a few times. By contrast, the development of computational methods for robust activity recognition still remains a very challenging task. Moving camera conditions, dynamic background, occlusions, abrupt illumination changes and multiple subjects in the scene, introduce significant difficulties in the development of a robust motion analysis framework. This is evident from the abundance of different motion analysis approaches that have been developed [1–5].

The aim of this work is to obtain a good representation of the human activity depicted in an image sequence and classify it into one or more activity categories. For robustness against occlusion and clutter, we opt for a sparse representation that is based on visual descriptors extracted around a set of spatiotemporal interesting points. An important issue that we address is handling general motions caused by a moving camera. We do so by detecting the space-time interesting points on the estimated optical flow of the image sequence. In order to filter out vectors that correspond solely to camera motion, we locally subtract the median of the optical flow vectors prior to the interesting point detection. In this way, the detected points correspond to areas of independent motion in the scene. Inspired by the success of 'bag of word' models, which rely on a codebook constructed by clustering static spatial descriptors, we build a 'bag of verbs' model by clustering local space-time descriptors. We use a boosting algorithm in order to select the most discriminant sets of codewords for each class in the training set. Finally, we use Relevance Vector Machines (RVM) for classification. The kernel of the RVM is defined according to the proximity between the test examples and the selected features of each class

[*] Corresponding author. Address: Department of Computing, Imperial College London, 180 Queensgate SW7 2AZ, London, UK. Tel.: +44 7950585131.
E-mail addresses: aoikonom@imperial.ac.uk (A. Oikonomopoulos), m.pantic@imperial.ac.uk (M. Pantic), ioannis.patras@elec.qmul.ac.uk (I. Patras).

0262-8856/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2009.05.010

in the training set. In order to demonstrate the efficiency of our approach, we present experimental results on three different datasets of human activities. The latter range from simple aerobics exercises to common everyday actions, like walking and running.

1.1. Related work

Activity recognition systems can be divided into two main categories. Within the first category fall methods that use tracking in order to represent the actions. Several methods have been proposed, including tracking of articulated models (e.g. [6–8]), tracking of landmark points (e.g. [9–12]), or methods that attempt to track specific shapes (e.g. [13,14]), like silhouettes and hand shapes. The result is subsequently used for recognition, either by taking into account the resulting trajectories of the landmark points or by taking into account the temporal transitions of the tracked models and shapes.

The difficulty in acquiring reliable trajectories for recognition has lead several researchers to assume that they are known a-priori (e.g. [11,12]). This difficulty originates from the fact that articulated objects (e.g. the human body) can undergo various changes in appearance and geometry due to rotations, deformations, rapid non-linear motions, and partial occlusions. Furthermore, the high dimensionality of the problem, and appearance changes that lead to the so-called drifting problem [15], make tracking of body parts cumbersome and in most cases unreliable.

Within the second category fall methods that rely on local spatiotemporal feature-descriptor representations. Their success in object detection and localization, their sparsity, and robustness against illumination, clutter, and viewpoint changes [16] have inspired a number of methods in the area of motion analysis and activity recognition. Detection of keypoints, in particular, has been very popular, due to their detection simplicity. A typical example are the space-time interest points [17,18], which correspond roughly to points in space-time where the direction of motion changes abruptly. A similar approach is used by Dollar et al. [19], where an activity is summarized using sets of space-time cuboids. Entropy based spatiotemporal salient point representations are used in [20], as a temporal extension of the salient point detector proposed in [21]. The method takes into account the information content of pixels within a spatiotemporal neighborhood and detects areas where there is a significant amount of motion. One of the most common type of descriptors stems from the Scale Invariant Feature Transform (SIFT). Introduced in [22], it has been used widely in a variety of applications, including object classification (e.g. [23,22]) and scene classification (e.g. [24,25]). The underlying concept in SIFT is the use of a cascade of Gaussian filters of variable width. Keypoints are subsequently detected as the extrema of the Difference of Gaussian filters (DoG) across different scales. Shape contexts [26] constitute an alternative local representation, in which a log polar histogram of the object's edges is used in order to capture local shape. Its robustness against scale changes and its ability to capture local spatial structure have made it very appealing for applications related to human detection (e.g. [27]). A similar and very effective approach for capturing local structure in space and time are the histograms of oriented gradients (HoG), extensively used for activity recognition (e.g. [28–31]). Biologically inspired representations, such as the C features, have been proposed in [32,33]. The method works in an hierarchical way and the obtained features are invariant to scale changes in space and time. Finally, Wong and Cipolla [34] use global information in terms of dynamic textures in order to minimize noise and detect their interesting points.

Illumination variability, smooth motions and, moving camera conditions, have lead several researchers to implement their methods in domains other than the intensity values at the image pixels. Optical flow, in particular, has been a popular choice. Ke et al. [35] use optical flow fields in their volumetric feature detector in order to represent and recognize actions. The authors claim that their method is robust to camera motion, however they do not explicitly handle it, making the method sensitive to less smooth motions of the camera. Shape flows [36,37] has been another method for dealing with camera motion. In this method, motion flow lines acquired by tracking [37] or using MPEG motion vectors [36] are used in order to represent the activities. Matching is done directly using the optical flow lines. However, the matching problem is NP-hard, and while relaxation methods can reduce the computational complexity, it still remains high. Fathi and Mori [38] use mid-level features consisting of optical flow and spatial gradient vectors and use two rounds of boosting in order to train their classifier. Ahmad and Lee [39] use a combination of shape flow and image moments in order to build their descriptors. However, their method relies on silhouettes that are extracted by background subtraction. Shechtman and Irani [40] propose an algorithm for correlating spatiotemporal event templates with videos without explicitly computing the optical flow. Their work, in conjunction with the temporal templates of Bobick and Davis [41] is used in [42] in order to construct a descriptor of shape and flow for detecting activities in the presence of clutter (e.g. crowds).

Exemplar-based methods, like the ones mentioned above, often require a large amount of training examples. Furthermore, in order to classify an unknown instance of an activity, the number of comparisons that have to be performed is equal to the number of the exemplars in the training set. This makes classification a time consuming process. To remedy this, a number of recent works use visual codebooks in order to detect and recognize objects and/or humans. The visual codebook creation is performed by clustering the extracted feature descriptors in the training set [43]. Each of the resulting centers is considered to be a codeword and the set of codewords forms a 'codebook'. In a 'bag of words' approach, each instance (for example an image) is represented as a histogram of codewords. Recognition is then performed by means of histogram comparison.

Visual codebooks have been extensively used for detecting objects, humans and activities. Aiming at object recognition, in [23], SIFT-like descriptors are extracted hierarchically and a visual codebook is created for each level of the hierarchy. Then, the histogram of the descriptors at each level of the hierarchy is classified using Support Vector Machines (SVM). SIFT descriptors in a bag-of-words framework are also used in [24] for the combined problem of event, scene, and object classification, with application to sports images. In [44], a voting scheme similar to the one by Leibe et al. [45] is implemented for localization and recognition of activities. An interesting work is presented in [46], where oriented rectangles are fitted on human silhouettes and matched against a visual codebook. However, the use of silhouettes assumes knowledge of the background and is sensitive to noise and camera motion. Furthermore, the system in [46] ignores dynamic information, and a human activity is considered as a sequence of static poses.

The major weakness of 'bag of words' approaches is that, by histogramming the descriptors, any information about their relative position is lost. In an attempt to remedy this, several researchers have proposed approaches that attempt to encode the spatial relationships between the features. One such approach is the relative encoding of the feature positions by considering a reference point, i.e. the center of the object on which the feature is extracted. Notable works which employ this concept for modeling static objects are those by Marszalek and Schmid [47] and by Leibe et al. [45]. A similar method is used in [48], where the features consist of fragments belonging to object edges, while the position of each fragment is stored relatively to the object's center. Alternatives to this concept of structure include the 'doublets' of Sivic et al. [49]

and the local self similarity descriptor of Shechtman and Irani [29]. In the former, pairs of visual words co-occurring within local spatial neighborhoods are identified. In the latter, areas in images/videos that share similar geometric properties and similar space/time layout are matched. An interesting approach using the principle of 'search by example' is presented in [25]. The method uses the principles of Google by initially detecting and indexing spatial SIFT descriptors in the training set. When presented with a query image or a small image patch, the system returns a number of matches in the training set that have similar descriptor values as well as similar spatial layout. Finally, constellations of bags of features consisting of both static and dynamic descriptors are used in [50,51] in order to recognize human activities. The use of a constellation model assists in recovering the spatiotemporal structure of the descriptors in the examples.

1.2. Overview of the approach

In this paper, we propose a set of novel visual descriptors that are derived from the spatiotemporal salient points described in [20]. In order to deal with motion induced by a moving camera, we first estimate the optical flow using the algorithm proposed in [52]. In order to compensate for camera motion, we locally subtract the median of the optical flow vectors, estimated within a local window. The local nature of the filtering process that we apply helps us reduce the influence of motion components that are due to global translational motion and vectors that originate from more general camera motion, like rotation and scaling. An example is given in Fig. 2, in which the camera zoom is largely suppressed, while the remaining flow vectors that correspond to the activity that takes place in the scene, that is, the upwards motion of the subject's hands, are pronounced. The salient points that we extract, correspond therefore to areas where independent motion occurs, like ongoing activities in the scene. Centered at each salient point, we define a spatiotemporal neighborhood whose dimensions are proportional to the detected space-time scale of the point. Then, a three-dimensional piecewise polynomial, namely a B-spline, is fitted at the locations of the salient points that fall within this neighborhood. Our descriptors are subsequently derived as the partial derivatives of the resulting polynomial. At the next step, the set of descriptors extracted from each spline is accumulated into a number of histograms. This number depends on the maximum degree of the partial derivatives. Since our descriptors correspond to geometric properties of the spline, they are translation invariant. Furthermore, the use of the automatically detected space-time scales of the salient points for the definition of the neighborhood ensures invariance to space and time scaling. Similar to other approaches (e.g. [46,50]), where a codebook of visual words is created from a set of appearance descriptors, we create a codebook of visual verbs by clustering our motion descriptors across the whole dataset. Here, we use the term 'verb' instead of a 'word' for our codebook entries, since each entry corresponds to a combined shape and motion descriptor rather than just a shape descriptor. Each video in our dataset is then represented as a histogram of visual verbs. We use boosting in order to select the most informative sets of verbs for each class. Finally, we use a kernel based classifier, namely the Relevance Vector Machine (RVM) [53], in order to classify test examples into one of the classes present in the training dataset. We evaluate the proposed method using three different databases of human actions. These include the widely used Weizmann [54] and KTH [18] datasets as well as our own aerobics dataset [10]. Finally, we present experiments aimed at evaluating the generality of our descriptors, that is, their ability to encode and discriminate between unseen actions, coming from an entirely different dataset than that on which the method is trained. A list of the successive steps of our algorithm is given in Table 1.

One of the main contributions of the method proposed is the sparsity of the extracted descriptors – they are extracted at spatiotemporal regions that are detected at sparse locations within the image sequence. This is contrary to the work of, e.g. Blank et al. [54], where a whole image sequence is represented as a space-time shape. In addition, our representation allows us to automatically detect the local spatiotemporal scale of the individual events taking place in the scene, as opposed to, e.g. [34,38], where the sequences are normalized with respect to their duration. The extracted descriptors are robust to camera motion due to the use of filtered optical flow as the basis of all computations. This is in contrast to alternative methods like [54,33,50,51,19,28,46] where a stationary camera is assumed. Finally, by selecting the most discriminant features, we obtain a form of a prototypical example for each class, as opposed to e.g. [54,33], where the whole feature set is used. Our results are comparable to the ones presented in [54,33,46,19,51] and show an improvement with respect to those reported in [50,28,35,18] for the same test sequences.

The remainder of the paper is organized as follows. In Section 2 we describe our feature extraction process. This includes the optical flow computation, the detection of the salient points, the subsequent B-spline fitting and the creation of the visual codebook. In Section 3 we present our classification method, including the feature selection procedure that we applied for selecting the most discriminant time windows of each class. In Section 4 we present the experimental results and in Section 5 we draw some conclusions.

2. Representation

In this section we introduce the visual descriptors that we use in order to represent an image sequence. We will initially provide some basics on B-splines and we will subsequently describe how are they used in extracting local, spatiotemporal, image-sequence descriptors. Finally, we will briefly explain the process that we carry out in order to create a codebook from these descriptors.

2.1. B-spline surfaces

Let us define an $M \times N$ grid of control points $\{P_{ij}\}, i = 1 \ldots M$ and $j = 1 \ldots N$. Let us also define a knot vector of h knots in the u direction, $U = \{u_1, u_2, \ldots, u_h\}$ and a knot vector of k knots in the v direc-

Table 1
Successive steps of the proposed approach.

(1)	Compute the optical flow according to the algorithm of [52] (Fig. 2b), and compensate for camera motion using local median filters (Fig. 2d)
(2)	Detect spatiotemporal salient points on the resulting flow field using the algorithm of [20] (Fig. 2c)
(3)	Place each salient point at the center of a space-time cube with dimensions proportional to the space-time scale of the point (Fig. 4a)
(4)	Fit a B-spline polynomial on the salient points that fall within the space-time cube (Fig. 4b)
(5)	Compute the partial derivatives of the resulting polynomial (Fig. 5)
(6)	Bin the computed partial derivatives into a histogram and form a descriptor vector for each B-spline polynomial
(7)	Create a codebook of K verbs by clustering the resulting descriptor vectors across the whole dataset
(8)	Perform feature selection using the Gentleboost algorithm [55] in order to select the most informative sets of descriptors for each class
(9)	Perform classification using the Relevance Vector Machine [53]

tion, $V = \{v_1, v_2, \dots, v_k\}$. Then, a B-spline surface of degrees p and q, in the u and v directions respectively, is given by:

$$F(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} Q_{i,p}(u) Q_{j,q}(v) P_{ij}, \tag{1}$$

where $Q_{i,p}(u)$ and $Q_{j,q}(v)$ are B-spline basis functions of degree p and q, respectively, defined as:

$$\begin{aligned} Q_{i,0}(u) &= \begin{cases} 1 & \text{if } u_i < u < u_{i+1} \text{ and } u_i < u_{i+1} \\ 0 & \text{otherwise} \end{cases}, \\ Q_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} Q_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} Q_{i+1,p-1}(u). \end{aligned} \tag{2}$$

The grid of control points is referred to as the control net, while the range of the knots is usually equal to $[0 \dots 1]$. Essentially the number of knots determine how coarse is the approximation. That is, the larger the number of knots, the larger the number of points on which the spline is evaluated.

By fitting B-spline surfaces on sparsely detected spatiotemporal salient points (Section 2.3), we want to approximate the smooth motion of subjects performing certain types of activities. It is well known that polynomials of high degree tend to fit well around the control points, and theoretically, increase the accuracy of the representation. However, precise fitting of the polynomials to the control points, whose localization may also be affected by noise in the background, would make the representation too specific for a particular activity example, or in other words, lead to overfitting. Furthermore, higher order polynomials become increasingly imprecise further away from the control points. Since the latter are the sparsely detected salient points, it is evident that polynomials of high order would decrease the robustness of the representation against noise. An example is depicted in Fig. 1, where polynomials of 3rd and 8th degrees are fitted on the same set of control points. From the figure it is evident that as the order of the polynomials increases, the representation becomes less smooth and increasingly imprecise in areas between the control points. As a good tradeoff between descriptiveness and robustness we use in this work 3rd degree polynomials, that is, $p = q = 3$.

2.2. Optical flow

Our analysis relies on the motion field that is estimated using an optical flow algorithm. Our goal is to detect spatiotemporal interest points and subsequently extract spatio(temporal) descriptors at areas with significant variation in motion information, such as motion discontinuities, rather than at areas with significant spatiotemporal intensity variation, such as space-time intensity corners (see [17]). The latter approach generates too many points in the case of camera motion in a moving, textured background. By contrast, as long as the camera motion is smooth the spatiotemporal salient point detection at motion discontinuities should be invariant to it. We estimate optical flow using the algorithm of [52], due to its robustness to motion discontinuities and to outliers to the optical flow equation.

The presence of general camera motion, like camera translation, small rotations, and scale changes (resulting from camera zoom) makes the application of a motion compensation technique an essential step prior to feature extraction. In this way, the extracted features will describe solely the independent motion taking place in the scene, like human activities. In the proposed method we use local median filtering in order to compensate for the local motion component. In a similar way, a global affine motion model can be estimated, and then the corresponding component be compensated for. For both, the goal is to provide representations that are invariant (mainly) to camera motion.

The advantages of global versus local methods for obtaining representations that are invariant to certain transforms (in our case the camera motion) are a subject of ongoing debate in the field of Computer Vision. For example, in order to compensate for changes in the illumination, both local (e.g. local filtering, color-invariant features) and global models (e.g. gamma correction) have been proposed. A clear disadvantage of global parametric models is their sensitivity to outliers (in our case, independently moving objects, including the human subject). On the other hand, the disadvantage of local methods is that they result to representations that may be less descriptive (i.e. 'too invariant'). For example, after local intensity normalization gray and white areas cannot be distinguished. The motion compensation method that we use in this work falls within the area of local methods, and is very similar to the filtering that is applied in order to compensate for illumination changes, for example, for extracting Quotient Images [56]. The latter are shown to be robust to local intensity variation due to, for example, cast shadows.

Examples of motion compensation are depicted in Fig. 2 (local) and Fig. 6 (local and global). It can be seen that most vectors that are due to camera motion are suppressed, and the ones corresponding to independent motion in the scene (i.e. the human activities) are pronounced.

2.3. Spatiotemporal descriptors

After compensating for optical flow vectors that are due to camera motion, as explained in Section 2.2, we use the algorithm proposed in [20] in order to extract a set of spatiotemporal salient points $S = \{(\vec{c}_i, \vec{s}_i)\}$. Here, $\vec{c}_i = (x, y, t)$ is the spatiotemporal position of the point with index i. The vector $\vec{s}_i$ is the spatiotemporal scale at which the point was detected and has a spatial and temporal dimension. This scale is automatically detected by the algorithm in [20], as the scale at which the entropy of the signal within the local spatiotemporal neighborhood defined by it is locally maxi-

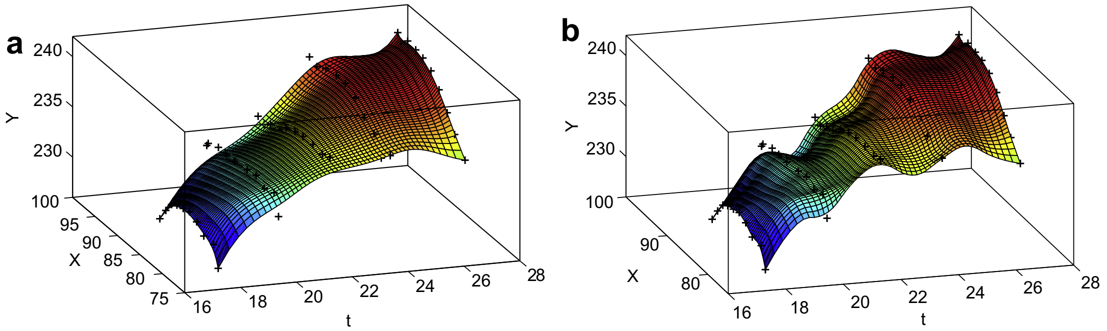


Fig. 1. B-spline representations of different orders, fitted around a set of control points (black crosses). B-splines of 3rd degree (a) fit smoothly around the control points, as opposed to 8-degree splines (b).

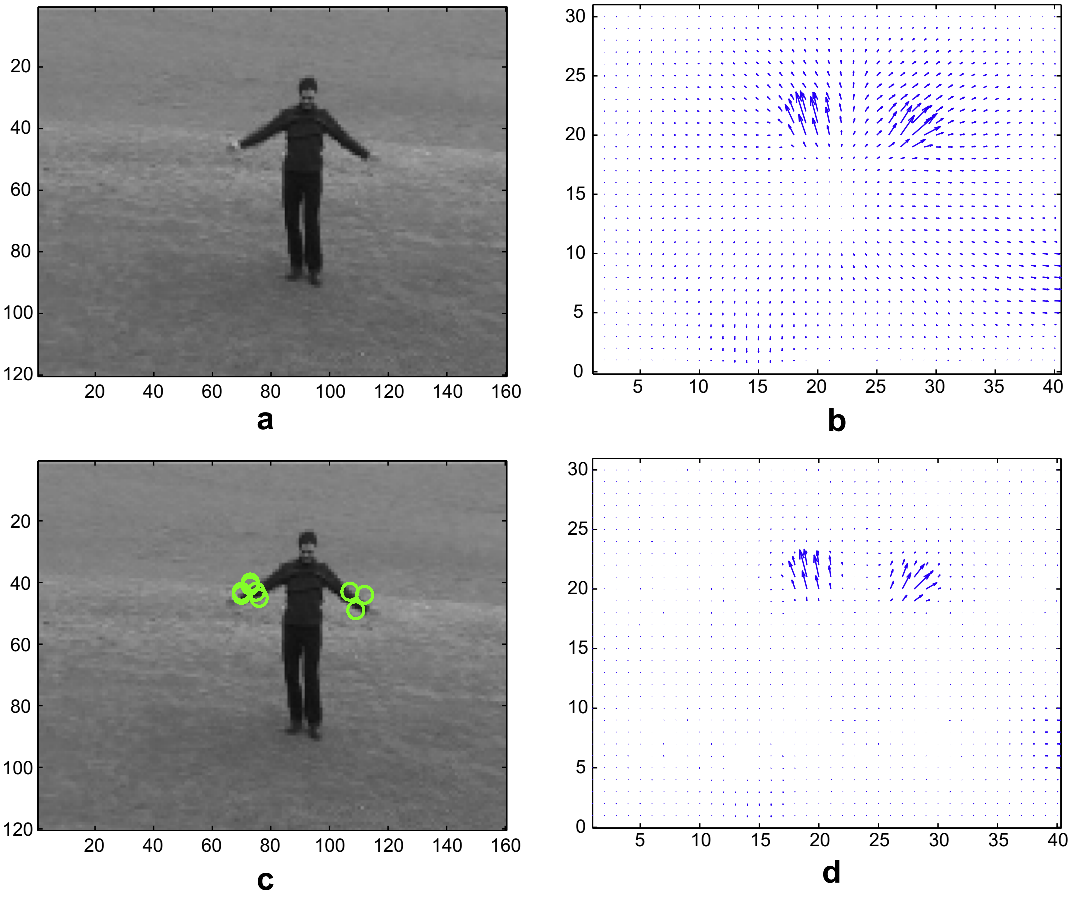


Fig. 2. (a) A single frame from a *hand-waving* sequence in which camera zoom is occurring and the corresponding optical flow field, before (b) and after (d) the application of the local median filter. Removal of flow vectors that are due to the camera zoom is evident. (c) Some of the salient points detected using the optical flow field in (d).

mized. A subset of the salient points detected on a frame of a *hand-waving* sequence is shown in Fig. 2c. We should note that for the detection of the points shown in Fig. 2c, contribute a number of frames before and after the shown frame (temporal scale).

2.3.1. Preprocessing

In this section we will describe the preprocessing steps that are followed prior to the B-spline fitting on the detected salient points. In order to fit a B-spline polynomial we first need to define its control net, that is, P_{ij}. Formally, for each salient point location we want to fit a polynomial having as control net the points within a small neighborhood around the point in question. For a good fit, however, ordering of the control points in terms of their spatiotemporal location is an important factor in order to avoid loops. In order to make this more clear, let us consider a set of points $L = \{L_i\}$ sampled randomly from an arbitrary curve, as shown in Fig. 3a. Ideally, a polynomial having the set L as its control net would approximate the curve with the one depicted as a dotted line in the same figure. However, in order for this to happen, the points in L should be given in the correct order, that is, $L = \{L_1, L_2, \dots, L_n\}$, as shown in Fig. 3a. If this is not the case, then the polynomial will attempt to cross the points in a different order, creating unwanted loops. Furthermore, it is clear that any points enclosed by the curve, like the one marked as a triangle in the same figure will also degrade the approximation and should not be considered. In order to overcome these problems, we perform two preprocessing steps on the set S of the detected salient points, both performed frame-wise.

In the first step, we eliminate points that are enclosed within the closed surface defined by the motion boundary. In our implementation, a point lies on the motion boundary if it lacks any neighbors within a circular slice shaped neighborhood of radius r, minimum angle a, and having the point as origin. This process is demonstrated in Fig. 3b, where the point in the centre of the circle is selected as being located on the boundary.

In the second step, we order the selected boundary points. We do this by randomly selecting a point on the boundary as a seed and by applying an iterative recursive procedure that matches the seed point with its nearest neighbor in terms of Euclidean distance. This process is repeated using as new seed the selected nearest neighbor, until there are no nearest neighbors left, that is, either an edge has been reached or all points have been selected.

Let us note that the described procedure above is local in nature. The primary role of r is the selection of the points that are on the motion boundary. By properly setting up the radius r, the points in the boundary of a moving object will be selected even if there are more than one subjects performing activities in the same scene, as long as they are at a distance of at least r pixels from each other. Due to the use of salient point representations (i.e. as the control points for the spline approximations), the presence of noise will minimally affect the boundary selection procedure. Due to the local entropy measurements for the salient point detection, noise will not greatly affect the conveyed information that leads to their detection. While noise may lead to the detection of spurious salient points, their saliency measure will be low compared to the points that belong to the actual motion boundary

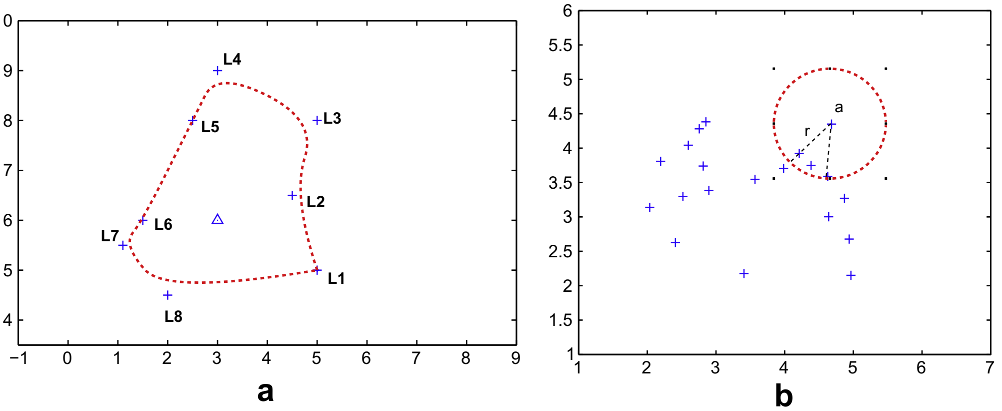


Fig. 3. (a) Points in sequence belonging to an arbitrary curve and (b) the boundary selection.

and therefore, will be discarded by the algorithm described in [20]. In this work we have empirically selected a radius of 10 pixels and an angle of 70 degrees.

2.3.2. Spline approximation

Let us denote with $S' = \{(\vec{c}'_i, \vec{s}'_i)\}$ the set of spatiotemporal salient points located on the motion boundary, that are obtained by the procedure described in the previous section. For each salient point $(\vec{c}'_i, \vec{s}'_i)$ we define a spatiotemporal neighborhood centered at c'_i with dimensions proportional to the scale vector $\vec{s}'_i$. Let us denote with O the set of points engulfed by this neighborhood (see Fig. 4a). Then, for each O, we fit a B-spline polynomial as in Eq. (1). The grid of control points P_{ij} in Eq. (1) corresponds to the set O, that is, each P_{ij} is a point in space-time. We should note that the grid is not and does not need to be uniform, that is, the pairwise distances of the control points may differ. The knot vectors U and V are a parameterization of the fitted B-spline, and essentially encode the way in which the B-spline surface changes with respect to its control points. More specifically, the knot vector U encodes the way the x coordinates change with respect to y, while the knot vector V encodes the way both x and y change with respect to t.

Using this process, any given image sequence is represented as a collection of B-spline surfaces, denoted by $\{F_i(u,v)\}$. Recall that we fit one surface per salient point position and therefore, the number of surfaces per sequence, that is, F_is, is equal to the number of points in S'. An example of a spline fitted to a set of points is presented in Fig. 4. Each member of the set $\{F_i(u,v)\}$ is essentially a piecewise polynomial in a three dimensional space. This means that we can fully describe its characteristics by means of its partial derivatives with respect to its parameters u, v. That is, for a grid of knots of dimensions $k \times h$ we calculate the following matrix R_i of dimensions $(pq-1) \times (hk)$:

$$R_i = \begin{bmatrix} \frac{\partial F_i(u_1,v_1)}{\partial u} & \cdots & \frac{\partial F_i(u_h,v_k)}{\partial u} \\ \vdots & \ddots & \vdots \\ \frac{\partial^{(p-1)(q-1)} F_i(u_1,v_1)}{\partial u^{p-1} \partial v^{q-1}} & \cdots & \frac{\partial^{(p-1)(q-1)} F_i(u_h,v_k)}{\partial u^{p-1} \partial v^{q-1}} \end{bmatrix} \tag{3}$$

where $\partial^p / \partial u^p$ is the partial derivative of order p with respect to u. Note (see Eq. (1)) that $F_i(u,v)$ is the value of the spline at u, v, that is, $F_i(u,v)$ is a 3×1 vector. Consequently, each element of the matrix in Eq. (3) is a vector of the same dimensions, and more specifically a vector that specifies the direction of the corresponding derivative. In Fig. 5 an illustration of the first derivatives with respect to u and v is given. The derivatives are drawn as three-dimensional vectors, superimposed on the spline from which they were extracted.

Our goal is to represent each F_i with a single descriptor vector. For this reason, we bin each row of R_i into a single histogram of partial derivatives and we concatenate the resulting $(pq-1)$ histograms into a single descriptor vector. This vector constitutes the descriptor of F_i and consequently the descriptor of a specific region in space and time of the image sequence. By repeating this process for each F_i, we end up with a set of descriptors for the whole sequence.

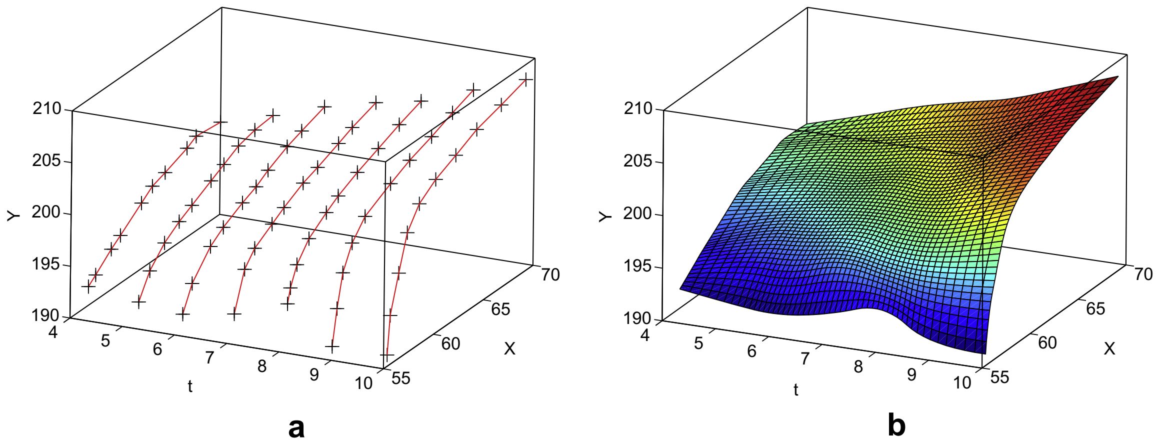


Fig. 4. (a) The set of points O that are engulfed within a spatiotemporal neighborhood. The straight line connections between the points are for illustration purposes, to depict the ones belonging to the same frame. (b) The resulting B-spline approximation described in Section 2.1.

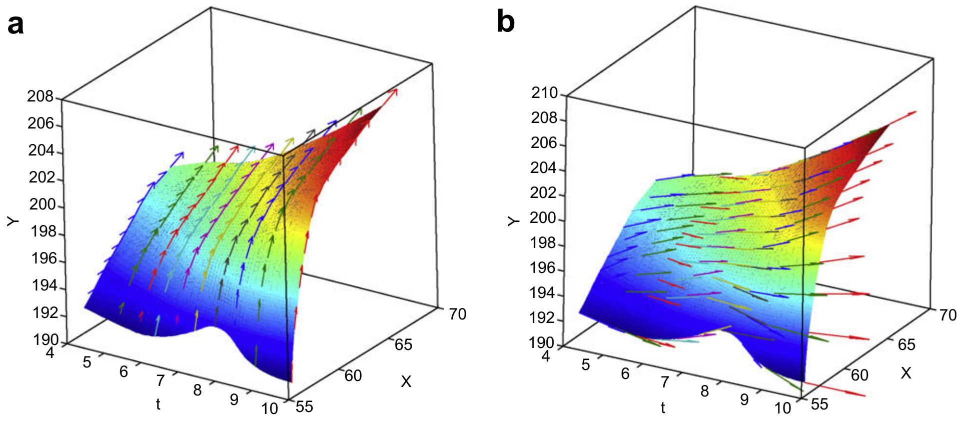


Fig. 5. First derivatives with respect to (a) u and (b) v, drawn as three-dimensional vectors.

2.4. Codebook creation

Applying a clustering algorithm to the whole set of descriptors, in order to create a codebook, is usually very time and memory consuming. As suggested in [47], the way a vocabulary is constructed has little or no impact to the final classification results. In accordance to this finding, we randomly subsample our descriptor set. Subsequently, we cluster our randomly selected features using K-means clustering. The resulting cluster centers are treated as codewords, and the set of codewords constitutes the utilized codebook. In this work we used 1000 clusters, as a compromise between representation accuracy and speed.

3. Classification

Having constructed a codebook, the goal is to represent and classify a test image sequence into one of the available classes in the training set. A conventional application of a ‘bag of verbs’ approach would dictate that each image sequence in the dataset is represented as a histogram of visual codewords drawn from the codebook. Using the codebook in this way for our specific set of descriptors resulted in recognition rates of about 60% or less, using a 1-NN classifier based on the χ^2 distance between the histograms of the test and training sequences. The χ^2 distance was selected as it is more suitable for comparing histograms than the Euclidean distance. The low recognition rate that was obtained using this approach clearly indicates that using a single histogram of codewords to describe a whole sequence is not suitable. The most plausible reason for this is that a large number of descriptors in the codebook is common to many (if not all) classes. We adopt, therefore, a different approach and use the codebook in order to recover similar temporal slices between the sequences. The descriptors that belong to these slices have a specific extent in time, depending on the temporal scales at which they were extracted (see Section 2.3.2).

Even though KNN based classification using a nearest neighbor approach based on the χ^2 distance between the temporal slices works quite well (see Table 2), it has an important drawback. A large number of frames in the dataset are likely to be common to many classes and therefore uninformative of the class. For example, for a database depicting aerobic exercises, frames that correspond to the neutral position of the human body, that is, standing upright and facing the camera with the hands resting along the body, are such common yet uninformative frames. It is apparent that these frames will be matched for all classes that contain them and they cannot be considered characteristic for a specific activity. Furthermore, while certain codewords might not be informative about the class at a certain frame, they might be informative at another. It is evident, therefore, that a selection step preceding the classification would be highly beneficial in our case.

Table 2
Recall and precision rates acquired on the three datasets for the three classification experiments.

Database	1-NN	Gentle-NN	Gentle-RVM
Aerobics	0.84/0.85	0.91/0.94	0.95/0.96
Weizmann*	0.88/0.88	0.9/0.93	0.92/0.92
Weizmann	0.78/0.82	0.83/0.84	0.84/0.84
KTH	0.67/0.68	0.78/0.78	0.81/0.82

In this work we use the GentleBoost algorithm [55] in order to select useful features for classification, due to its performance in terms of convergence speed and classification accuracy [57]. Subsequently, we use the selected features in a Relevance Vector Machine (RVM) classification scheme.

3.1. Feature selection

In feature selection by GentleBoost, at each stage a weak classifier is trained on a weighted version of the dataset. Here, each weak classifier operates on a different dimension/feature of the feature vector. At each stage the algorithm picks the weak classifier that, given the current sample weights w, separates the examples of different classes best. Then, the classification error is estimated and the samples are reweighted so that misclassified samples weigh more. This procedure is repeated until the classification error does not change significantly between iterations. The performance measure used by GentleBoost to learn the weak classifiers and evaluate their performance is the classification rate, that is, the percentage of correctly classified examples, regardless of their class.

The usual input to boosting algorithms is a set of positive and negative examples where the dimension of each example is equal to the dimension of the feature vector. The setup for selecting the most discriminative temporal slices for each class in our case is somewhat different, since a feature is the histogram of the visual verbs for a single slice, represented by a $1 \times P$ vector, where P is the number of codewords in the codebook. We create initially an $P \times M_i$ array A, where M_i is the total number of slices in all the examples of class i. Each entry $A_{p,m_i}, p = 1 \dots P,\ m_i = 1 \dots M_i$ is the percentage of codeword with index p in the m_i slice that belongs to class i. This forms the set of positive examples. Each row of A, therefore, expresses the percentage of the codeword that corresponds to that row across the whole set of positive temporal slices. Subsequently we select M_i slices from all other classes, which we consider to be the set of negative classes. Each slice is selected as the one with the minimum χ^2 distance from the corresponding M_i slice of the positive set (belonging to array A). The selected temporal slices constitute a $P \times M_i$ array B representing the set of negative examples. Each entry of B is defined in a similar way as the ones in A. Each row of B expresses the percentage of the codeword that corresponds to that row across the selected set of negative temporal slices. By concatenating arrays A and B we arrive to a $2P \times M_i$ array C. This is the input to the boosting algorithm.

By performing this procedure we expect that a slice in the positive part of C (i.e. array A) that is common to all classes, will not be selected, since it will have a very similar representation with the slice in the negative part (i.e. array B) with which it is being compared. On the other hand, a slice in the positive part of C that is unique to the class will have quite a different representation from the slice in the negative part with which it is compared and is therefore selected as being informative of the class. By performing this procedure for all classes, we end up with a set of selected temporal slices per class, which we will subsequently use for classification.

3.2. Relevance Vector Machine

A Relevance Vector Machine (RVM) classifier is a probabilistic sparse kernel classifier that is identical in functional form to the Support Vector Machine (SVM) classifier. Given a dataset of N input-target pairs $\{(F_n, l_n), 1 \leqslant n \leqslant N\}$, an RVM learns functional mappings of the form:

$$y(F) = \sum_{n=1}^{N} w_n K(F, F_n) + w_0, \tag{4}$$

where $\{w_n\}$ are the model weights and $K(.,.)$ is a kernel function. Gaussian or Radial Basis Functions (RBF) have been extensively used as kernels in RVM and can be viewed as a measure of similarity between F and F_n. For our work, we use the distance of each test example to the selected features of each class, in order to define a kernel for the RVM. More specifically, we use a Gaussian RBF to define the kernel, that is,

$$K(F, F_n) = e^{-\frac{D(F,F_n)^2}{2\eta}}, \tag{5}$$

where η is the width of the kernel and D is the average of the minimum distances between the temporal slices of the test sequence and the informative temporal slices of each class as selected by Gentleboost. In the two class problem, a sample F is classified to the class $l \in [0, 1]$ that maximizes the conditional probability $p(l|F)$. For L different classes, L different classifiers are trained and a given example F is classified to the class for which the conditional distribution $p_i(l|F), 1 \leqslant i \leqslant L$ is maximized:

$$Class(F) = \arg\max_i (p_i(l|F)). \tag{6}$$

4. Experiments

4.1. Datasets

For our experiments we use three different datasets of human activities. The first one is the KTH dataset [18], containing 6 different actions: *boxing*, *hand-clapping*, *hand-waving*, *jogging*, *running*, and *walking*. Each action is performed by 25 subjects several times under different conditions, including scale changes, indoors/outdoors recordings, and varying clothes. The second is the Weizmann dataset, used in [54], and contains nine different actions such as *walking*, *running*, and *jumping*, performed once by nine different subjects. We also used a dataset that we created[1] [10], containing 15 different aerobics exercises performed twice by five different subjects.

4.2. Camera motion

In order to demonstrate the effectiveness of the local median filter in compensating for general camera motion, we simulate the latter in videos from the aerobics dataset. In contrast to the KTH datasrt, the aerobics dataset contains sequences with textured, non-planar background. In order to simulate camera motion, we apply a rectangular cropping window around the subjects in the dataset. Subsequently, we apply rapid, random displacements of the cropping window. For comparison, we also apply a global affine model for motion compensation. We use an iterative weighted least squares algorithm for estimating the parameters of the affine model, where the weights are updated at each iteration using the robust m-estimator of Geman-McClure [58]. In Fig. 6 the results of both motion compensation techniques are depicted.

As can be seen from the figure, both methods efficiently filter out the majority of the flow vectors that are due to the camera motion. For the case of the global model, there exist a number of residual flow vectors that do not belong to the occurring activity (frames (a, d, e)). While the median filter does not seem to suffer from this problem, it occasionally tends to filter out vectors that belong to the activity. This is evident in frames (b) and (c), and is directly related to the size of the utilized filtering window. In this paper, we used a window of 25 × 25 pixels.

4.3. Classification results

We performed our classification experiments using cross validation, carried out in the leave-one-subject-out manner. That is, in order to classify a test example performed by a specific test subject, we created a codebook and trained the respective classifiers using all available data instances except of those belonging to the same class and performed by the same subject as in the test example. In order to assess the impact of each step of our methodology (the feature selection and the RVM classification), we present classification results from three different experiments. In the first experiment, each temporal slice of a test sequence is matched with the closest slice of a training sequence in terms of their χ^2 distance. The overall distance measure between the image sequences is then calculated as the average of the minimum calculated slice distances. The test example is then classified to the class of the training example for which the smallest overall distance has been calculated. In the second experiment the slices of each test example are matched against the selected slices of each class (selected by the Gentleboost in the feature selection step). Once again, this is done in terms of the χ^2 distance. The test sequence is then assigned to the class for which the smallest resulting distance has been calculated. Finally, in the third experiment we present the classification results obtained using RVM as a classifier. More specifically, we use the distance of each test example to the selected slices of each class in order to define the kernel of the RVM, according to Eq. (6). For comparison, we present two different results for the Weizmann dataset. In the first, the *skip* class is included in the database. In the second one, this class is not included, since several researchers present results that do not include this class. This class is arguably the most difficult to recognize [54,33]. The collective classification results for all three datasets and all three experiments, in terms of recall and precision rates are given in Table 2, where the reduced Weizmann dataset is denoted by *Weizmann**.

As we can see from Table 2, there is a considerable increase in classification performance on all three datasets when the feature selection step is introduced, that is, when the most discriminative temporal slices/windows per class are selected for training. This re-

[1] This dataset is available upon request. Please contact A. Oikonomopoulos for further information.

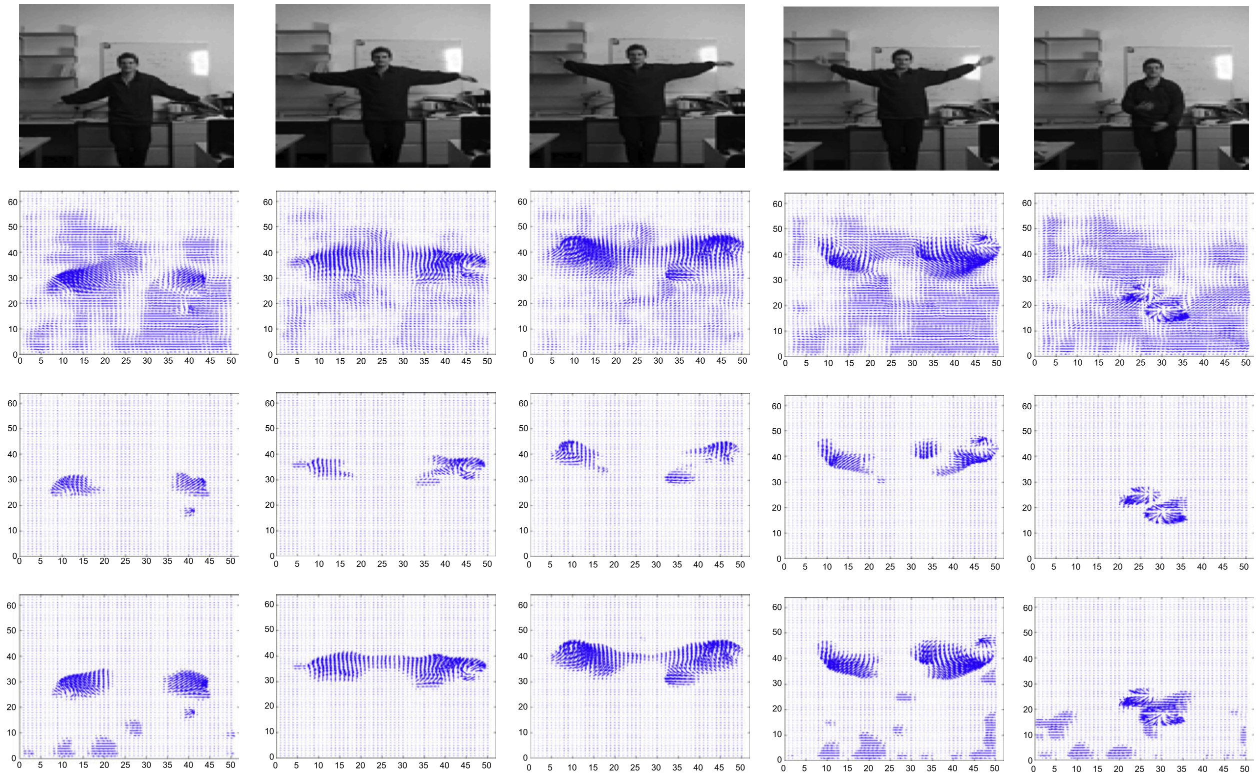
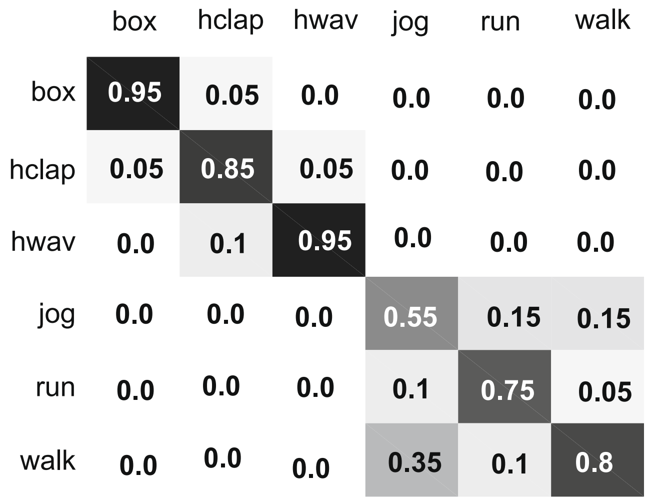


Fig. 6. Motion compensation results using local median filters and a global affine model. Top row: individual frames. 2nd row: estimated optical flow. 3rd row: remaining flow vectors after locally subtracting the median of the optical flow. Bottom row: remaining flow vectors after the application of robust global affine motion compensation.

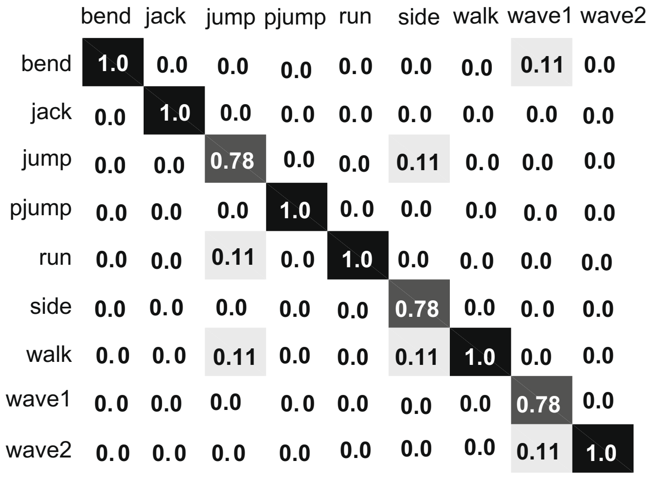
sult clearly suggests that slices which are common in a large number of classes have a negative impact on the classification performance. This justifies our choice to conduct feature selection. On the other hand, there is only a slight increase in classification performance on the Weizmann and KTH datasets by additionally using RVM for classification, while the increase for the aerobics dataset is about 4%. We attribute this to the fact that the most informative elements are already selected by our feature selection scheme. We should note, however, that the contribution of the RVM classification step is always positive, but not very significant except for the aerobics dataset.

The average recall rate for Gentle-RVM approach applied to the KTH dataset is about 81%. From the confusion matrix in Fig. 7, we can see that confusions are commonly made between similar classes *running* and *jogging*. However, as noticed by Schuldt et al. [18], these confusions are in fact reasonable, since what appears to some people as running may appear to others as jogging and vice versa. Concerning the Weizmann* dataset (where the *skip* class is excluded), the average recall rate of Gentle-RVM approach is 92%. From the confusion matrix in Fig. 8, we can see that there are some confusions between similar classes like *jump*, *run*, *walk*, and *side*, as well as *wave1* and *wave2*. However, as we can see from Fig. 8, these confusions are rather rare. Finally, we performed similar classification experiments using a global affine model for camera motion compensation. The parameters of the model were estimated as described in Section 4.2. We achieved a 75% average recall rate for the



	box	hclap	hwav	jog	run	walk
box	0.95	0.05	0.0	0.0	0.0	0.0
hclap	0.05	0.85	0.05	0.0	0.0	0.0
hwav	0.0	0.1	0.95	0.0	0.0	0.0
jog	0.0	0.0	0.0	0.55	0.15	0.15
run	0.0	0.0	0.0	0.1	0.75	0.05
walk	0.0	0.0	0.0	0.35	0.1	0.8

Fig. 7. Confusion matrix for the KTH dataset.



	bend	jack	jump	pjump	run	side	walk	wave1	wave2
bend	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.11	0.0
jack	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jump	0.0	0.0	0.78	0.0	0.0	0.11	0.0	0.0	0.0
pjump	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
run	0.0	0.0	0.11	0.0	1.0	0.0	0.0	0.0	0.0
side	0.0	0.0	0.0	0.0	0.0	0.78	0.0	0.0	0.0
walk	0.0	0.0	0.11	0.0	0.0	0.11	1.0	0.0	0.0
wave1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.78	0.0
wave2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.11	1.0

Fig. 8. Confusion matrix for the Weizmann* dataset.

KTH dataset, using Gentleboost for feature selection and RVM for classification.

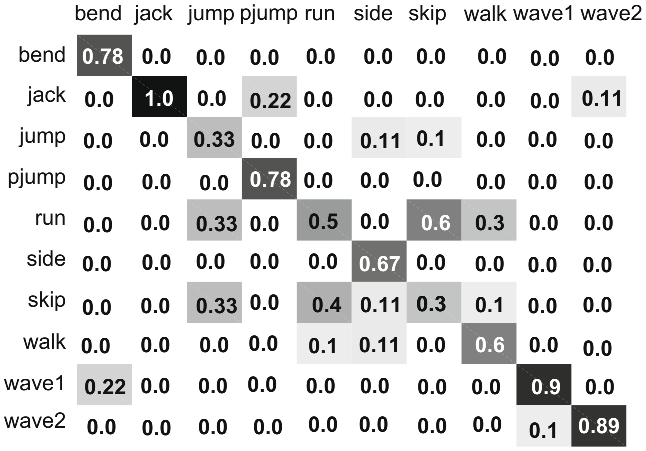
As shown in Table 3, the results that we obtained for the Gentle-RVM approach on the KTH dataset outperform the ones presented in, e.g. [35,18]. Furthermore, we achieve similar results as the ones reported in [19,51]. Contrary to our method, however, these works do not specifically address camera motion, since they assume a stationary camera. Furthermore, we do not apply any preprocessing step to the raw image sequences prior to feature detection, contrary to Fathi and Mori [38], who use stabilized sequences of cropped frames, centered on the human figure as their input. Similarly, Wong and Cipolla [34], temporally normalize their sequences to have similar length. Instead, we handle temporal variations by automatically detecting temporal scale in the spatiotemporal salient point detection step and by using this scale in order to define the neighborhoods for the B-spline approximation. Finally, we do not perform any background subtraction before detecting our features, as opposed to Jhuang et al. [33] and Ahmad and Lee [39]. The latter, use a Gaussian Mixture Model (GMM) in order to identify foreground pixels as the ones which vary over time. In the proposed method, however, we achieve a similar effect by detecting the spatiotemporal salient points at areas in which there is significant amount of motion, as described in [20].

Concerning the Weizmann dataset, our results are almost 4% lower than those reported in [54] and [33]. However, besides handling camera motion, the main advantage of our method compared to these works is the feature selection step. By contrast in [54,33] the whole set of the extracted features is used for classification purposes. In addition, our system uses a sparse representation as opposed to [54], where a whole image sequence is represented as a space-time shape. Sparse, local representations, are shown to be significantly better in dealing with clutter and occlusions for object detection and recognition in comparison to global representations (e.g. see [59]). Similar observations can be therefore expected in the case of action recognition problems. As can be seen from the results in Table 2 and Figs. 7 and 8, this assumption proved to be true. The only other work presented so far in the body of the related literature that uses a sparse and structured representation is that proposed in [50]. However, a recognition rate of 72.8% is reported on the Weizmann dataset which is by far inferior to the 92% achieved by our method.

As previously mentioned, we used cross validation in a leave-one-subject-out manner in order to evaluate our method. This means that for any test example, the codebook contains information about the class of this example. We would like to determine here, if our features are general enough to handle completely unknown classes. That is, given a codebook of visual verbs we want to examine how well can this codebook discriminate classes that did not contribute to its creation. Our motivation for this experiment lies in the fact that our system is able to consistently recover short-term motion in small spatiotemporal regions. Therefore, given that an unknown class can share a number of similar spatiotemporal regions with several known classes, there should be some ability for good discrimination. We performed two different experiments. In the first experiment we created a codebook from 14 classes of the aerobics dataset. The remaining class was used for testing. In other words, the remaining class was represented by using visual verbs defined for other classes. The result was that 8 out of 10 instances of the test class were correctly classified. In the second experiment, we created a codebook from the whole aerobics dataset and tested it for discrimination of classes from the Weizmann dataset. The classes between these two datasets are almost completely different. Exceptions are the classes *jack*, *wave1*, and *wave2* of the Weizmann dataset which are also present in the aerobics dataset. The average recall rate for this experiment was 67.7%, with the worst performing classes being *jump*, *run*, *walk*, and *skip*, as we can see from the confusion matrix of Fig. 9. However, poor results for these classes could be expected, as these classes do not seem to share common frames with classes of the aerobics dataset. Overall, these results indicate that it might be possible to use the proposed descriptors for representing new classes of actions. We intend to investigate this issue in further detail using all action databases and performing the same experiments with features that are currently the state of the art in the field, like those proposed in [54,33] and [50] (i.e. Poisson, C2, and Gradient features).

5. Conclusions

In this paper, we presented a feature-based method for human activity recognition. The features that we extract stem from automatically detected salient points and contain static information concerning the (moving) body parts of the subjects as well as dy-



	bend	jack	jump	pjump	run	side	skip	walk	wave1	wave2
bend	0.78	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jack	0.0	1.0	0.0	0.22	0.0	0.0	0.0	0.0	0.0	0.11
jump	0.0	0.0	0.33	0.0	0.0	0.11	0.1	0.0	0.0	0.0
pjump	0.0	0.0	0.0	0.78	0.0	0.0	0.0	0.0	0.0	0.0
run	0.0	0.0	0.33	0.0	0.5	0.0	0.6	0.3	0.0	0.0
side	0.0	0.0	0.0	0.0	0.0	0.67	0.0	0.0	0.0	0.0
skip	0.0	0.0	0.33	0.0	0.4	0.11	0.3	0.1	0.0	0.0
walk	0.0	0.0	0.0	0.0	0.1	0.11	0.0	0.6	0.0	0.0
wave1	0.22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.0
wave2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.89

Fig. 9. Confusion matrix for the Weizmann dataset, where the codebook used for representing the examples was created from the aerobics dataset.

Table 3
Comparisons of the proposed method to various methods proposed elsewhere for KTH dataset.

Methods	Features	Classifier	Weaknesses/benefits	Accuracy (%)
Our method	B-splines	Gentleboost + RVM	Camera motion handling, sparse representation (+)	80.8
Ke et al. [35]	Optical flow	Boosting	Robust to camera motion (+), but no specific handling (−)	62.97
Schuldt et al. [18]	ST interesting points [17]	SVM	Stationary camera (−), results on selected sequences (−)	71.83
Ahmad et al. [39]	Flow + moments	MDHMM	Background subtraction (−)	88.3
Dollar et al. [19]	Gabor filters	NN	Stationary camera (−)	81.17
Wong et al. [34]	DoG + NMF	SVM	Samples preprocessed into similar temporal length (−)	86.7
Niebles et al. [51]	Gabor filters	pLSA + SVM	Stationary camera (−)	81.5
Fathi et al. [38]	Optical flow	Adaboost	Stabilization (−)	90.5
Jhuang et al. [33]	C features	SVM	Background subtraction (−)	91.7

namic information concerning the movements/activities. Furthermore, our features are robust to camera motion, through the use of filtered optical flow for their extraction. We use the extracted features to recover similar temporal windows that essentially encode the short-term motion typical for a given activity in a 'bag of verbs' approach. Our results show that our representation is able to recover a wide variety of different motion/activity classes. Furthermore, our experiments show that our system is able to generalize well and handle unknown classes (i.e. those that did not contribute to the creation of the utilized codebook). To the best of our knowledge, this is the first approach to human activity recognition that achieves generalization to unknown classes.

The future directions of our research include additional experiments in order to determine the robustness of the proposed method in more challenging scenarios, like in the presence of dynamic background. To wit, an obvious improvement of our method which would enable handling of dynamic background, is to take into account the spatiotemporal consistency of the features. This would enable not only dynamic background handling, but activity segmentation as well, in cases where more than one activity is taking place within the same scene (e.g. several activities occurring one after the other and/or two or more people in the scene doing different activities).

Acknowledgement

This work has been supported in part by the European Research Council under the ERC Starting Grant agreement No. ERC-2007-StG-203143 (MAHNOB).

References

[1] R. Poppe, Vision-based human motion analysis: an overview, Computer Vision and Image Understanding 108 (2007) 4–18.

[2] M. Pantic, A. Pentland, A. Nijholt, T. Huang, Human computing and machine understanding of human behavior: a survey, Lecture Notes in Artificial Intelligence 4451 (2007) 47–71.

[3] S. Mitra, T. Acharya, Gesture recognition: a survey, IEEE Transactions on Systems, Man and Cybernetics (SMC) Part C 37 (2007) 311–324.

[4] T.B. Moeslund, A. Hilton, V. Krueger, A survey of advances in vision-based human motion capture and analysis, Computer Vision and Image Understanding 104 (2006) 90–126.

[5] J. Aggarwal, Q. Cai, Human motion analysis: a review, Computer Vision and Image Understanding 73 (1999) 428–440.

[6] L. Sigal, S. Bhatia, S. Roth, M. Black, M. Isard, Tracking loose-limbed people, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1., 2004, pp. 421–428.

[7] D. Ramanan, D.A. Forsyth, A. Zisserman, Tracking people by learning their appearance, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 65–81.

[8] H. Moon, R. Chellappa, 3d shape-encoded particle filter for object tracking and its application to human body tracking, EURASIP Journal on Image and Video Processing (2008).

[9] M. Abdelkader, A. Roy-Chowdhury, R. Chellappa, U. Akdemir, Activity representation using 3d shapemodels, EURASIP Journal on Image and Video Processing (2008).

[10] A. Oikonomopoulos, M. Pantic, Human body gesture recognition using adapted auxiliary particle filtering, in: Advanced Video and Signal Based Surveillance, 2007, pp. 441–446.

[11] A. Yilmaz, M. Shah, Recognizing human actions in videos acquired by uncalibrated moving cameras, in: Proceedings of IEEE International Conference Computer Vision, vol. 1., 2005, pp. 150–157.

[12] Y. Sheikh, M. Sheikh, M. Shah, Exploring the space of a human action, in: Proceedings of IEEE International Conference Computer Vision, vol. 1, 2005, pp. 144– 149.

[13] B. Stenger, A. Thayananthan, P. Torr, R. Cipolla, Model-based hand tracking using a hierarchical Bayesian filter, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1372–1384.

[14] F. Guo, G. Qian, Monocular 3d tracking of articulated human motion in silhouette and pose manifolds, EURASIP Journal on Image and Video Processing (2008).

[15] I. Matthews, T. Ishikawa, S. Baker, The template update problem, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 810–815.

[16] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 1615–1630.

[17] I. Laptev, T. Lindeberg, Space-time Interest Points, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 432–439.

[18] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, vol. 3., 2004, pp. 32–36.

[19] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, VS-PETS (2005) 65–72.

[20] A. Oikonomopoulos, I. Patras, M. Pantic, Spatiotemporal salient points for visual recognition of human actions, IEEE Transactions on Systems, Man and Cybernetics Part B 36 (2005) 710–719.

[21] T. Kadir, M. Brady, Scale saliency: a novel approach to salient feature and scale selection, International Conference on Visual Information Engineering (2000) 25–28.

[22] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2004) 91–110.

[23] A. Agarwal, B. Triggs, Hyperfeatures multilevel local coding for visual recognition, in: European Conference on Computer Vision, vol. 1., 2006 pp. 30–43.

[24] l.J. Li, L. Fei-Fei, What, where and who? classifying events by scene and object recognition, in: ICCV, 2007, pp. 1–8.

[25] J. Sivic, A. Zisserman, Video google: efficient visual search of videos, LNCS 4170 (2006) 127–144.

[26] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 509–522.

[27] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1. 2005, pp. 886–893.

[28] C. Thurau, Behavior histograms for action recognition and human detection, in: ICCV Workshop on Human Behavior, 2007, pp. 299–312.

[29] E. Shechtman, M. Irani, Matching local self-similarities across images and videos, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[30] I. Laptev, P. Perez, Retrieving actions in movies. in: Proceedings, IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[31] A. Klaeser, M. Marszalek, C. Schmid, A spatiotemporal descriptor based on 3d gradients, in: British Machine Vision Conference, 2008.

[32] T. Serre, L. Wolf, T. Poggio, Object recognition with features inspired by visual cortex, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, vol. 2., 2005, pp. 994–1000.

[33] H. Jhuang, T. Serre, L. Wolf, T. Poggio, A biologically inspired system for action recognition, in: Proceedings, IEEE International Conference on Computer Vision, 2007.

[34] S. Wong, R. Cipolla, Extracting spatiotemporal interest points using global information, in: Proceedings, IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[35] Y. Ke, R. Sukthankar, M. Hebert, Efficient visual event detection using volumetric features, in: Proceedings, IEEE International Conference on Computer Vision, vol. 1, 2005, pp. 166–173.

[36] H. Jiang, R. Martin, Finding actions using shape flows, in: European Conference on Computer Vision, 2008.

[37] P. Natarajan, R. Nevatia, View and scale invariant action recognition using multiview shape-flow models, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[38] A. Fathi, G. Mori, Action recognition by learning mid-level motion features, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[39] M. Ahmad, S. Lee, Human action recognition using shape and clg-motion flow from multi-view image sequences, Pattern Recognition 41 (2008) 2237–2252.

[40] E. Shechtman, M. Irani, Space-time behavior based correlation, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1. 2005, pp. 405–412.

[41] A. Bobick, J. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 257–267.

[42] Y. Ke, R. Sukthankar, M. Hebert, Event detection in crowded videos, in: Proceedings, IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[43] F. Moosmann, B. Triggs, F. Jurie, Fast discriminative visual codebooks using randomized clustering forests, NIPS (2006) 985–992.

[44] K. Mikolajczyk, H. Uemura, Action recognition with motion-appearance vocabulary forest, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[45] B. Leibe, A. Leonardis, B. Schiele, Combined object categorization and segmentation with an implicit shape model, ECCV'04 Workshop on Statistical Learning in Computer Vision (2004) 17–32.

[46] N. Ikizler, P. Duygulu, Human action recognition using distribution of oriented rectangular patches, Lecture Notes in Computer Science 4814 (2007) 271–284.

[47] M. Marszalek, C. Schmid, Spatial weighting for bag-of-features, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, vol. 2., 2006, pp. 2118–2125.

[48] A. Opelt, A. Pinz, A. Zisserman, A boundary-fragment-model for object detection, in: Proceedings of the European Conference on Computer Vision, 2006.

[49] J. Sivic, B. Russell, A. Efros, A. Zisserman, W. Freeman, Discovering Objects and their Location in Images, in: Proceedings, IEEE International Conference on Computer Vision, vol. 1., 2005, pp. 370–377.

[50] J. Niebles, L. Fei-Fei, A hierarchical model of shape and appearance for human action classification, in: Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[51] J.C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial–temporal words, in: British Machine Vision Conference, vol. 1, 2006.
[52] M. Black, P. Anandan, A framework for the robust estimation of optical flow, in: ICCV, 1993, pp. 231–236.
[53] M. Tipping, The relevance vector machine, Advances in Neural Information Processing Systems (1999) 652–658.
[54] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, in: Proceedings, IEEE International Conference on Computer Vision, vol. 2. 2005, pp. 1395–1402.
[55] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Stanford University Technical Report, 1993.
[56] A. Shashua, T. Riklin-Raviv, The quotient image: class-based re-rendering and recognition with varying illuminations, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 129–139.
[57] A. Torralba, K. Murphy, W. Freeman, Sharing features: efficient boosting procedures for multiclass object detection. in: CVPR, 2004, pp. 762–769.
[58] M. Black, A. Rangarajan, On the unification of line processes, outlier rejection, and robust statistics, International Journal of Computer Vision 19 (1996) 57–91.
[59] R. Fergus, P. Perona, A. Zisserman, Weakly supervised scale-invariant learning of models for visual recognition, International Journal of Computer Vision 71 (2007) 273–303.