Computer Vision and Image Understanding 113 (2009) 1064–1075

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

Human motion recognition using support vector machines

Dongwei Cao, Osama T. Masoud, Daniel Boley, Nikolaos Papanikolopoulos *

Department of Computer Science and Engineering, University of Minnesota, 4-192 EE/CS Building, 200 Union Street SE, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 25 November 2007
Accepted 12 June 2009
Available online 21 June 2009

Keywords:
Human motion recognition
Recursive filtering
Support vector machine

ABSTRACT

We propose a motion recognition strategy that represents each videoclip by a set of filtered images, each of which corresponds to a frame. Using a filtered-image classifier based on support vector machines, we classify a videoclip by applying majority voting over the predicted labels of its filtered images and, for online classification, we identify the most likely type of action at any moment by applying majority voting over the predicted labels of the filtered images within a sliding window. We also define a classification confidence and the associated threshold in both cases, which enable us to identify the existence of an unknown type of motion and, together with the proposed recognition strategy, make it possible to build a real-time motion recognition system that cannot only make classifications in real-time, but also learn new types of motions and recognize them in the future. The proposed strategy is demonstrated on real datasets.

1. Introduction

The purpose of human motion recognition is to assign a specific label to a motion. Recognition can be offline or online based on the requirements of the specific application. In offline recognition, an entire videoclip is available and the goal is to identify the type of motion recorded in the videoclip. In online recognition, the goal is to identify the motion type with only a portion of a videoclip, while it is in progress. Furthermore, motivated by the need to detect suspicious behavior in security monitoring, it is desirable for a motion recognition system to detect unknown human behaviors, which are the motion types not available when the system was built. In this paper, we propose a strategy that is applicable to both offline and online recognition, and is capable of not only recognizing known motion types but also detecting unknown motion types.

Generally, there are two tightly related steps in building a motion recognition system, i.e., extracting motion features and training a classifier using these features. The majority of relevant work in motion recognition focuses on motion feature selection, including extracting features from 2D tracking data [1–7] or 3D tracking information [8,9], or extracting motion information directly from images [10–14]. In particular, Ref. [14] proposed a new feature extracting algorithm called motion history histogram, which is an improvement over the motion history image [12] in terms of encoding the time span of movement, and provided an FPGA implementation. Given a set of features that is believed to be able to characterize the motion of interest, most recognition algorithms are based on either template matching [12,13] or state-space matching that usually uses Hidden Markov Models (HMM) [11]. Other recognition algorithms employ neural networks [2]. The performance of these recognition algorithms, especially the ones based on template matching, is highly dependent on the quality of the extracted motion features, which in general should reflect the nonlinear nature of human motions.

The basic idea behind the motion recognition strategy proposed in this paper is called *frame grouping*, where we first classify every frame of a videoclip separately, then use majority voting over the resulting labels to determine the motion type of the videoclip. Each frame is represented by a *filtered image*, which is constructed using *recursive filtering* (RF) proposed by Masoud and Papanikolopoulos [15] and encodes both the spatial layout of the scene in the current frame and the temporal relationship between the current frame and previous frames, i.e., the temporal continuity of a videoclip. The filtered image method is similar to the motion history image (MHI) and the motion history histogram (MHH). However, both MHI and MHH need some kind of video segmentation such that the video clip contains roughly one cycle of the motion, while the filtered image approach [15] requires no video segmentation and the motion information is encoded by a set of images. Given a set of labeled filtered images, the filtered-image classifier is built using a support vector machine (SVM) [16]. The reason for choosing the support vector machine is that, through an implicit mapping based on a Mercer kernel [16], some nonlinear features are extracted automatically and used for classification, and it is expected that these nonlinear features encode sufficient information to discriminate different motion types. In other words, the difficult task of feature selection in motion recognition is done implicitly. A practical reason for choosing the support vector machine to

* Corresponding author. Fax: +1 612 625 0572.
E-mail addresses: dcao@cs.umn.edu (D. Cao), masoud@cs.umn.edu (O.T. Masoud), boley@cs.umn.edu (D. Boley), npapas@cs.umn.edu (N. Papanikolopoulos).

1077-3142/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2009.06.002

classify filtered images is that it has been shown to be very effective on image classification tasks like handwritten digits recognition [17], which is similar in many ways to the filtered-image classification problem encountered here.

There have been some works on using SVM to perform human motion recognition [18,19], which differ from the current paper mainly on the motion representation strategy. The strategy in [18] is based on local spatial–temporal features and uses k-means clustering algorithm to extract a set of primitive events, which are fed into an SVM. In [19], a 3D spatial–temporal description of the motion is required as an input to an SVM. Compared with [18,19], the representation strategy in the current paper is simpler and easier to apply, while still achieving good performance.

Compared with [15], the motion recognition strategy proposed here has the following advantages: (i) for the same dataset under the same experimental setting, the strategy proposed here gives improved performance in terms of accuracy, training time, and classification time and (ii) for a videoclip with unknown motion type, which did not appear in the training dataset, the strategy in [15] will classify it into the "closest" known type among those appeared in the training dataset. This is undesirable for a recognition system since there are many ways a person or object can move and it is very difficult, or sometimes impossible, to have a training dataset that contains all possible types of motion. In other words, it was expected that some new type of motion might appear in the future, which are the motion types not represented by the training data when the system was built. The proposed strategy can identify the existence of a new motion type.

It is assumed in this paper that the camera axis is perpendicular to the direction of motion, which may be violated in real-world examples. There are undergoing works in our lab that try to relax it. In particular, Ref. [20] proposed a method to reconstructed a videoclip satisfying the assumption from videoclips that are recorded by several cameras whose optical axes are not perpendicular to the motion direction. The motion strategy proposed here can be applied to the re-constructed videoclips.

The rest of this paper is organized as follows: Section 2 briefly introduces the support vector machine, with emphasis on the classification confidence that can be used to identify unknown motion types. In Section 3, we describe the proposed strategy (RF-SVM) for offline and online recognition. Section 4 presents a series of experiments on offline recognition and online recognition, including discriminating known motion types and detecting unknown motion types. Section 5 concludes the paper with future research topics.

2. Support vector machine

In this section, we first give a brief introduction to the support vector machine (SVM) for binary classification, then describe how to assign a confidence to a classification. Since there are usually more than two types of motion, we also describe how to perform multi-class classification by combining several support vector machines.

2.1. Support vector machine

In a two-class classification problem, we are given a training dataset $\mathscr{D}_k$ of size n_k

$$\mathscr{D}_k = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^N,\ y_i \in \{1, -1\}\}, \tag{1}$$

where N is the dimension of $\mathbf{x}_i, y_i$ is the label of $\mathbf{x}_i$, and $i = 1, 2, \ldots, n_k$. The support vector machine (SVM) $f_k(\mathbf{x}_0)$ is defined as [16]

$$f_k(\mathbf{x}_0) = \operatorname{sign}(d_k(\mathbf{x}_0)) = \begin{cases} 1 : d_k(\mathbf{x}_0) \geqslant 0, \\ -1 : d_k(\mathbf{x}_0) < 0, \end{cases} \tag{2}$$

where $\mathbf{x}_0 \in \mathbb{R}^N$ is a test datum. Here, the term d_k is called *functional margin* and is defined as

$$d_k(\mathbf{x}_0) = \langle \mathbf{w}, \phi(\mathbf{x}_0) \rangle_{\mathscr{S}} + \theta, \tag{3}$$

where $\phi : \mathbb{R}^N \to \mathscr{S}$ is a (usually nonlinear) mapping from $\mathbb{R}^N$ to a Hilbert space $\mathscr{S}$ equipped with dot product $\langle \cdot, \cdot \rangle_{\mathscr{S}}$, and θ is the bias term. The optimal $\mathbf{w}$ and θ are obtained by solving the following optimization problem [16]:

$$\text{Minimize} : g(\mathbf{w}, \theta) = \frac{1}{2} \|\mathbf{w}\|_{\mathscr{S}}^2 + C \sum_{i=1}^{n_k} \xi_i \tag{4a}$$

$$\text{Subject to} : y_i \left(\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle_{\mathscr{S}} + \theta \right) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0, \quad i = 1, 2, \ldots, n_k, \tag{4b}$$

where $\| \cdot \|_{\mathscr{S}}$ is the norm induced by the inner product $\langle \cdot, \cdot \rangle_{\mathscr{S}}$, and $C > 0$ is the regularization parameter and needs to be specified a priori. The value of C controls the trade-off between the complexity of a classifier (through the term $\frac{1}{2}\|\mathbf{w}\|_{\mathscr{S}}^2$) and the accuracy of a classifier on the training dataset (through the term $\sum_{i=1}^{n_k} \xi_i$). Usually, the mapping ϕ is specified implicitly by choosing a Mercer kernel $K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$, which, for any $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^N$, has the following property:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathscr{S}}. \tag{5}$$

With the help of Lagrange multipliers, the optimal weight vector $\mathbf{w}$ has the following expansion:

$$\mathbf{w} = \sum_{i=1}^{n_k} \alpha_i y_i \phi(\mathbf{x}_i), \tag{6}$$

where the expansion coefficients $\alpha_1, \ldots, \alpha_n$ are the solution of the Wolfe dual of the optimization problem (4) [16]. There have been many efforts to develop efficient algorithms to solve the dual problem, for example, in [21–27], and an SVM can be trained quickly even if there are hundreds of thousands training data. Using Eq. (5), the functional margin $d_k(\mathbf{x})$ for a test datum $\mathbf{x}_0$ can be computed as follows:

$$d_k(\mathbf{x}_0) = \sum_{i=1}^{n_k} \alpha_i y_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_0) \rangle_{\mathscr{S}} + \theta = \sum_{i=1}^{n_k} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_0) + \theta. \tag{7}$$

The training data whose have non-zero α are called support vectors. With reference to Eq. (7), the functional margin can be computed using support vectors only. In other words, the support vectors can be seen as the most discriminative data, and functional margin is a weighted sum of these discriminative data. Putting in the context of this paper, we will have "support filtered images" and their corresponding coefficient α is a measure of their importance.

In summary, to build a support vector machine, one only needs to specify the kernel K and the penalizing coefficient C a priori. A commonly used technique to select the optimal kernel K and regularization coefficient C is cross validation [28], which is also used in this paper.

2.2. Classification confidence

In order identify unknown motion types, which do not appear in the training dataset, we use a rejection-based strategy. For each classification, we assign a confidence and reject this classification if its confidence is less than a prespecified threshold. A datum whose classification is rejected is believed to come from some unknown motion types. A traditional choice for such confidence is the posterior probability of the classification. However, the support vector machine has been criticized by its inability to provide such

a posterior probability. There are some efforts to compensate for this as shown, for example, in [29]. Nevertheless, we argue that, no matter how the classification confidence is defined, the performance of this rejection-based strategy (in terms of identifying unknown motion types) depends heavily on how the confidence threshold is chosen. In other words, the availability of the posterior probability of a classification is not the critical problem here.

For these reasons, we use the following strategy to identify unknown motion types. We define the classification confidence $CF(\mathbf{x}_0)$ for a test datum $\mathbf{x}_0$ in terms of its functional margin $d_k(\mathbf{x}_0)$ and let the classification threshold T_{CF} be 1, i.e.,

$$CF(\mathbf{x}_0) = |d_y(\mathbf{x}_0)|, \tag{8a}$$

$$T_{CF} = 1, \tag{8b}$$

where $|\cdot|$ denotes absolute value.

To illustrate the meaning of these definitions, we recall the optimization problem in Eq. (4). As we mentioned before, the term $\sum_{i=1}^{n_k}\xi_i$ in Eq. (4a) measures the accuracy of a classifier on the training dataset $\mathscr{D}_k$. For $i = 1, 2, \ldots, n_k$, the term ξ_i corresponds to the ith training data $(\mathbf{x}_i, y_i)$ and can be written explicitly as

$$\xi_i = \max(0, 1 - y_i d_k(\mathbf{x}_i)), \tag{9}$$

which means that $\xi_i > 0$ if and only if $y_i d_k(\mathbf{x}_i) < 1$. In other words, for a training datum $(\mathbf{x}_i, y_i)$ with $y_i d_k(\mathbf{x}_i) < 1$, the classification based on $\text{sign}(d_k(\mathbf{x}_i))$ will be considered as an error and, by minimizing the objective function g in Eq. (4a), we penalize this error by $C\xi_i$.

For a test datum $\mathbf{x}_0$, let y_0 denote the predicted label given by the SVM classifier f_k, i.e.,

$$y_0 = f_k(\mathbf{x}_0) = \text{sign}(d_k(\mathbf{x}_0)). \tag{10}$$

According to Eq. (9), such a prediction will be considered as *error* if

$$y_0 d_k(\mathbf{x}_0) < 1, \tag{11}$$

which is equivalent to the following condition:

$$|d_k(\mathbf{x}_0)| < 1, \tag{12}$$

since $y_0 \in \{-1, 1\}$ and $y_0 d_k(\mathbf{x}_0) > 0$ according to Eq. (10). Thus, using the classification confidence and threshold defined in Eq. (8), we will reject a classification if it is considered to be an *error*.

When using a Gaussian kernel, motion samples with low classification confidence include not only those samples lying "in between" the known motion types, but also most samples which are far away from the known motion types in any direction in $\mathbb{R}^N$. Fig. 1 demonstrates this effect using the following Gaussian kernel and penalizing coefficient C:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sigma\|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad \text{with } \sigma = 0.2 \text{ and } C = 0.125. \tag{13}$$

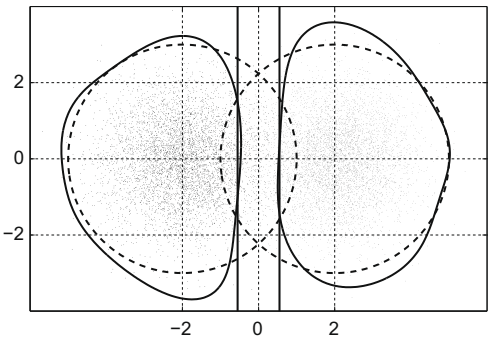


Fig. 1. Illustration of the rejection criterion defined in Eq. (8) with a Gaussian kernel. Any point outside the thick-solid curve will be rejected, i.e., treated as representing unknown types of motions.

In Fig. 1, the dashed circles enclose the regions within three standard deviations of the center of each class, the two vertical thin-solid lines enclose the "fuzzy" area in which there is at least a 10% chance of finding points from either class, and the thick-solid curves enclose the points whose classification confidence CF exceeds the threshold T_{CF}. Thus the thick-solid curves exclude both outliers (points far away in any direction) and fuzzy points (points between the two classes which are hard to classify), both of which are assumed to represent the unknown type of motion.

Thus, using the classification confidence and confidence threshold defined in Eq. (8), we assume that the unknown motion types will lie either "between" or "far away from" the known motion types. Furthermore, since the model parameter σ in the Gaussian kernel is tuned to achieve good classification performance, this strategy of detecting unknown motion types also assumes that a good value of the model parameter for classification will also be good for detecting unknown motion types. More specifically, the value of σ in the Gaussian kernel specifies the scale under which two objects should be compared and our assumption means that a good scale for classification is also a good scale for novelty detection. The unknown motion types could also be detected using one-class SVM with Gaussian kernel [30], or k-means clustering with a bound on the maximum acceptable distance from the clusters' centers. However, we would need to tune an additional set of model parameters, which is the σ of the Gaussian kernel if one-class SVM is used, and the maximum acceptable distance if k-means is used. One may also argue that the optimal parameter value for classification is not necessarily optimal for identifying outliers, which means we still need to tune a separate value for σ for novelty detection. However, as mentioned above, the value of σ in the Gaussian kernel specifies an optimal scale under which two objects should be compared, which should not vary between classification and novelty detection. Another advantage of the proposed strategy over the one-class-SVM approach and the k-means clustering approach, is that, like most novelty detection algorithms, it is hard for them to detect unknown motion types that lie between known motion types (cf. Fig. 1. The points between two closed curves cannot be identified by these two approaches).

To see that the proposed strategy can identify outliers detectable by algorithms like the one-class-SVM, we have been following an upper bound on the confidence $d(\mathbf{x}_0)$ when the Gaussian kernel is used

$$CF(\mathbf{x}_0) = |d_y(\mathbf{x}_0)|, \tag{14a}$$

$$= \left|\sum_{i=1}^{n_k} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_0) + \theta\right|, \tag{14b}$$

$$\leqslant \sum_{i=1}^{n_k} |\alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_0)| + |\theta|, \tag{14c}$$

$$\leqslant n_k C K(\mathbf{x}_i, \mathbf{x}_0) + |\theta|, \tag{14d}$$

$$< 1. \tag{14e}$$

The last inequality follows from the fact that $|\theta| < 1$ and, when $\mathbf{x}_0$ is sufficiently far from all training data, the Gaussian kernel value $K(\mathbf{x}_i, \mathbf{x}_0)$ approaches zero. If $|\theta| \geqslant 1$, according to Eq. (8), all data that are sufficiently far from all training data will be classified into the same class. Thus, using the confidence CF and the threshold T_{CF} will guarantee to identify the outliers, which are far away from all training data.

2.3. Multi-class classification

In motion recognition, there are typically more than two candidate motion types and, to deal with the resulting multi-class classification problem, we use a traditional strategy called *one-versus-the-rest* [31]. Assuming that there are L different candidate motion

types, the idea of *one-versus-the-rest* is to train L support vector machines, each of which discriminates one motion type from all the others. For example, the kth support vector machine $f_k(\mathbf{x})$ discriminates motions of type k, which is treated as class 1, from all the other types of motions, which are treated together as a single class -1. Using L support vector machines, the membership y_0 of a test datum $\mathbf{x}_0$ can be obtained using the following equation:

$$y_0 = \underset{k\in\{1,2,\cdots,L\}}{\operatorname{argmax}} d_k(\mathbf{x}_0), \tag{15}$$

where $d_k(\mathbf{x}_0)$ is the functional margin given by the kth support vector machine (7). Given the label y_0 predicted by Eq. (15), we define the classification confidence $CF(\mathbf{x}_0)$ corresponding to this classification as

$$CF(\mathbf{x}_0) = d_{y_0}(\mathbf{x}_0), \tag{16}$$

that is, the functional margin given by the support vector machine corresponding to the winning class y_0.

3. Motion recognition

In this section, we describe the proposed motion recognition strategy (RF-SVM) in two steps. We first describe how to construct the filtered image using recursive filtering (RF) and, given a set of training videoclips, how to construct the filtered-image classifier using the support vector machine (SVM). Then, we describe how to perform offline classification and online classification using RF-SVM.

3.1. Recursive filtering

In this paper, we use the *recursive filtering* proposed by [15] to encode motion information. The idea of recursive filtering is to encode both the spatial layout of the scene in the current frame and the temporal relationship between the current frame and previous frames. According to [15], for the frame I_t at time t, the filtered image F_t at time t is defined as

$$F_t = |I_t - M_t|, \tag{17a}$$
$$M_t = (1-\beta)M_{t-1} + \beta I_{t-1}, \tag{17b}$$
$$M_0 = I_0 = \text{Background}, \tag{17c}$$

where $t = 1, 2, \ldots$, and $|\cdot|$ denotes absolute value. The coefficient β is a pre-specified constant. If $\beta = 0$, the filtered image F_t will be the foreground and, if $\beta = 1$, F_t will be equivalent to image differencing. In the current study, a $\beta = 0.5$ was used, which was suggested in [15].

Fig. 2 shows the representative frames for eight types of actions and the corresponding filtered images. It can be seen from Fig. 2 that the temporal relationship among consecutive frames is encoded in the filtered images and, from the "tail" of the filtered images, we can easily tell the direction of motion, the recent trajectories of the parts of a person such as legs, and even the relative speed of different parts of the person's body. In other words, each filtered image encodes the motion information from a very recent past to now.

The filtered image F_t given by (17) is further thresholded to remove the noise and down-sampled to a lower resolution having width w and height h. Thus, each filtered image is represented by a matrix of size $w \times h$. Concatenating columns of this matrix, we obtain the representation as a vector $\mathbf{x}$ of dimension $w \times h$.

3.2. Filtered-image classifier

Without losing generality, we assume that there are L candidate motion types and there are m training videoclips. We denote the ith training videoclip as $\mathscr{X}_i$ and denote the label of $\mathscr{X}_i$ as y_i, where $y_i \in \{1, 2, \ldots, L\}$. We also assume that there are m_i frames in $\mathscr{X}_i$. As described in Section 3.1, each frame corresponds to one filtered image and is represented by a vector of length $w \times h$. Specifically, the jth frame ($j = 1, 2, \ldots, m_i$) of the ith videoclip $\mathscr{X}_i$ ($i = 1, 2, \ldots, m$) is represented by a vector $\mathbf{x}_{ij} \in \mathbb{R}^{w\times h}$. Using the idea of *frame grouping*, the ith videoclip $\mathscr{X}_i$ is represented by m_i points in $\mathbb{R}^{w\times h}$, all of which have the same label as $\mathscr{X}_i$. Thus, the training dataset $\mathscr{D}$ corresponding to the m training videoclips can be written as

$$\mathscr{D} = \bigcup_{i=1}^{m}\bigcup_{j=1}^{m_i}\{(\mathbf{x}_{ij}, y_{ij}) | \mathbf{x}_{ij} \in \mathbb{R}^N, y_{ij} = y_i\}, \tag{18}$$

where y_{ij} is the label of $\mathbf{x}_{ij}$. Using $\mathscr{D}$, we can build a filtered-image classifier $\mathscr{C}$ using support vector machines. As shown in Section 2, there are L support vector machines in classifier $\mathscr{C}$. In the rest of this section, we will describe how to make offline and online classifications using this classifier, together with how to assign a classification confidence to these classifications.

3.3. Offline classification

Using the filtered-image classifier $\mathscr{C}$ built using the training dataset $\mathscr{D}$, the offline classification is performed as follows: let $\mathscr{X}_0$ be a test videoclip having m_0 frames. After constructing a filtered image for each frame according to Eq. (17), the videoclip $\mathscr{X}_0$ is represented by

$$\mathscr{D}_0 = \{\mathbf{x}_{0j} | \mathbf{x}_{0j} \in \mathbb{R}^{w\times h},\ j = 1, 2, \ldots, m_0\}. \tag{19}$$

For $j = 1, 2, \ldots, m_0$, let y_{0j} be the label of $\mathbf{x}_{0j}$ predicted by the classifier $\mathscr{C}$ (cf. Eq. (15)). Then, the label y_0 of the videoclip $\mathscr{X}_0$ is determined using *majority voting* as follows:

$$y_0 = \underset{k\in\{1,2,\ldots,L\}}{\operatorname{argmax}} \ \mathbf{card}(\{j | y_{0j} = k,\ j = 1, 2, \ldots, m_0\}), \tag{20}$$

where $\mathbf{card}(\cdot)$ means the cardinality of a set. This procedure is detailed in Algorithm 1.

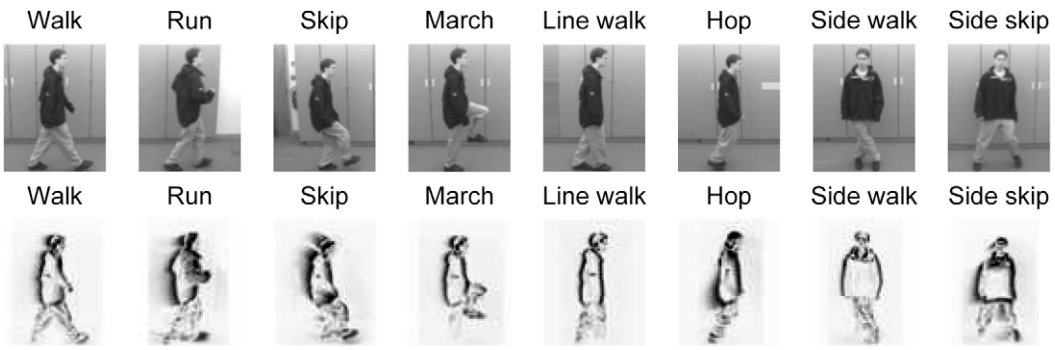


Fig. 2. Examples of frames (top row) and the corresponding filtered images (bottom row) of eight types of actions. The coefficient β in Eq. (17) is set to 0.5.

In order to identify the videoclip that represents an unknown type of motion, we need a confidence $CF(y_0)$ for the classification y_0 and the corresponding threshold T_{CF}. For $j = 1, 2, \ldots, m_0$, let CF_{0j} be the classification confidence corresponding to y_{0j} given by Eq. (16). With reference to Section 2.2, the confidence $CF(y_0)$ and the threshold T_{CF} are defined as

$$CF(y_0) = \frac{1}{m_0}\left(\sum_{y_{0j}=y_0} |CF_{0j}| - \sum_{y_{0j}\neq y_0} |CF_{0j}|\right), \tag{21a}$$

$$T_{CF} = 1, \tag{21b}$$

where the summation index j runs over $1, 2, \ldots, m_0$ (cf. Eq. (19)). This definition can be seen as an average confidence penalized by the "misclassified" filtered images. We will reject a classification y_0 if $CF(y_0) < T_{CF}$ and claim that the motion recorded in the videoclip $\mathscr{X}_0$ is unknown to the classifier.

Algorithm 1. Offline recognition

Require: A set of training videoclips $\mathscr{D}_{\text{videoclip}}$; A test videoclip $\mathscr{X}_0$ with m_0 frames.

1: Apply recursive filtering to each frame of each videoclip in $\mathscr{D}_{\text{videoclip}}$ and represent the resulting filtered image as a point in $\mathbb{R}^{w\times h}$, resulting in a training dataset $\mathscr{D}$ as defined in Eq. (18)
2: Build a filtered-image classifier $\mathscr{C}$ using support vector machines, with $\mathscr{D}$ as training dataset
3: Apply recursive filtering to each frame of the test videoclip $\mathscr{X}_0$ and represent the resulting filtered image as a point in $\mathbb{R}^{w\times h}$, resulting in a set $\mathscr{D}_0$ of size m_0 as defined in Eq. (19)
4: Classify all points in $\mathscr{D}_0$ using $\mathscr{C}$ (cf. Eq. (15))
5: The label of videoclip $\mathscr{X}_0$ is obtained by applying majority voting over the labels of points in $\mathscr{D}_0$ (cf. Eq. (20))

3.4. Online classification

There are two desired properties for an online motion recognition system:

- It should be able to recognize the most probable type of motion at each moment based only on the video presented so far.
- When the type of action changes, it should be able to detect such a change in a timely manner.

Here, we adopt the strategy of a sliding window. Denoting the width of the sliding window as b, the most probable type of motion at time t is determined using frames within the sliding window, which are frames from time $t - b + 1$ to time t. After constructing a filtered image for each frame within the sliding window using recursive filtering, the b frames within the sliding window are represented by the set $\mathscr{D}_t^b$ defined as

$$\mathscr{D}_t^b = \{\mathbf{x}_j | \mathbf{x}_j \in \mathbb{R}^{w\times h},\ j = t-b+1,\ t-b+2, \ldots, t-1, t\}. \tag{22}$$

For $j = t - b + 1, t - b + 2, \ldots, t$, let y_j be the label of $\mathbf{x}_j$ predicted by the classifier $\mathscr{C}$ (cf. Eq. (15)). Similar to offline recognition in Section 3.3, the most probable type of motion y^t at time t is determined using *majority voting*, i.e.,

$$y^t = \underset{k\in\{1,2,\ldots,L\}}{\text{argmax}}\ \mathbf{card}(\{j | y_j = k,\ j = t-b+1,\ t-b+2, \ldots, t\}), \tag{23}$$

where $\mathbf{card}(\cdot)$ means the cardinality of a set. This procedure is detailed in Algorithm 2.

In order to detect the appearance of an unknown type of action, we need a measure of confidence $CF(y^t)$ for the classification and the corresponding threshold T_{CF}. Such confidence measure can be defined in terms of the classification confidence levels obtained for the individual frames within the sliding window. For $j = t - b + 1, t - b + 2, \ldots, t$, let CF_j be the classification confidence corresponding to y_j given by Eq. (16). With reference to Section 2.2, the confidence $CF(y^t)$ and the threshold T_{CF} are defined as

$$CF(y^t) = \frac{1}{b}\left(\sum_{y_j=y^t} |CF_j| - \sum_{y_j\neq y^t} |CF_j|\right), \tag{24a}$$

$$T_{CF} = 1, \tag{24b}$$

where the summation index j runs over $t - b + 1, t - b + 2, \ldots, t$ (cf. Eq. (22)). We will reject the classification at time t if its confidence $CF(y^t) < T_{CF}$. If, in a video stream, many consecutive classifications are rejected, we can conclude that an unknown type of motion has happened, provided that the noise level in the video stream is not dominant.

The width b of the sliding window controls the trade-off between the sensitivity of the recognition system to the transition between different types of motions, and the robustness of the recognition system to noisy frames. The effects of different values of b will be studied in Section 4.

Algorithm 2. Online recognition

Require: A set of training videoclips $\mathscr{D}_{\text{videoclip}}$; The width b of the sliding window; A sequence of frames $I_t (t = 1, 2, \ldots)$

1: Apply recursive filtering to each frame of each videoclip in $\mathscr{D}_{\text{videoclip}}$ and represent the resulting filtered image as a point in $\mathbb{R}^{w\times h}$, resulting in a training dataset $\mathscr{D}$ as defined in Eq. (18)
2: Build a filtered-image classifier $\mathscr{C}$ using support vector machines, with $\mathscr{D}$ as training dataset
3: For $t = 1, 2, \ldots, b-1$, apply recursive filtering to frame I_t and represent the resulting filtered image as a point $\mathbf{x}_t \in \mathbb{R}^{w\times h}$
4: **for all** $t = b, b+1, \ldots$ **do**
5: Apply recursive filtering to the frame at time t and represent the resulting filtered image as a point $\mathbf{x}_t \in \mathbb{R}^{w\times h}$. Let $\mathscr{D}_t^b$ be the set of filtered images corresponding to the sliding window from $t - b + 1$ to t
6: Classify all points in $\mathscr{D}_b^t$ using the classifier $\mathscr{C}$, and the most probable type of motion y^t at time t is obtained by applying majority voting over the resulting labels
7: **end for**

4. Experimental results

In this section, we demonstrate the effectiveness of the proposed motion recognition strategy (RF-SVM) over real datasets. This section consists of three parts. First, Section 4.1 summarizes the dataset used in the experiments. In Section 4.2, we show the experimental results on offline recognition, and compare the proposed recognition strategy with the one proposed in [15]. Finally, Section 4.3 shows the results on online recognition. The capability of detecting unknown types of motions is also shown for both offline recognition (Section 4.2) and online recognition (Section 4.3).

4.1. Datasets and experimental setup

The data were collected by letting each of 29 individuals perform eight types of actions once, namely walk (W), run (R), skip (S), march (M), line walk (LW), hop (H), side walk (SW) and side

skip (SS). This resulted in 232 videoclips. We denote this set of videoclips as $\mathscr{D}_{\text{all}}$. Each videoclip has various numbers of frames and there are a total of 8741 frames in $\mathscr{D}_{\text{all}}$. Each frame is represented by a filtered image that has width $w = 25$ and height $h = 31$, i.e., its representation $\mathbf{x}_t$ given in Section 3.2 is a vector of dimension $w \times h = 775$. It should be noted that some of these eight motion types may sound not familiar, such as line walk, which is similar to walk except that, when touching the ground, the front foot's heel always touch the back foot's toe and both feet lie on the same line. One reason for using these motion types is to demonstrate the effectiveness of the proposed algorithm in discriminating different types of human motion, some of which may be similar to each other.

In order to build the filtered-image classifier $\mathscr{C}$, we need to train several support vector machines. According to Section 2.1, we need to specify the appropriate kernel K and penalizing coefficient C a priori for each support vector machine. Ideally, we should choose the optimal K and C for each SVM separately by cross validation. For simplicity, after some initial experiments, we use the following kernel K in all support vector machines throughout this paper:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{0.1 \times 775}\right). \tag{25}$$

For the penalizing coefficient C, we choose two values, which are used for the multi-class classifier in Section 4.2.1 and the two-class classifiers in Sections 4.2.2 and 4.3, respectively, i.e.,

$$C = 1.5, \quad \text{Multi-class classifier in Section 4.2.1}, \tag{26a}$$
$$C = 10, \quad \text{Two-class classifiers in Sections 4.2.2 and 4.3}. \tag{26b}$$

A more careful choice of K and C would definitely be beneficial.

4.2. Offline recognition

We show the experimental results on offline recognition in this section.

4.2.1. Recognition of known type of motion

In [15], after representing each frame using recursive filtering, a template for each type of motion is extracted using principal component analysis, and a new videoclip is classified using the nearest neighbor method based on the Hausdorff distance. We call this strategy "RF-NN", where RF stands for recursive filtering and NN stands for nearest neighbor. In this section, we will compare RF-NN with the proposed RF-SVM.

As mentioned before, there are two components in RF-SVM, one is filtered images classification using SVM and the other is majority voting. We will show in this section that majority voting is critical for the performance of the RF-SVM. It would also be interesting to show the importance of the SVM on the performance of the RF-SVM. This can be done by showing the performance of a strategy, which is denoted as "NN-MajorityVoting", that applies majority voting over labels of filtered images given by a nearest neighbor classifier using Euclidean distance, which is abbreviated as NN-Euclidean here. Although the relative performance between SVM and NN-Euclidean varies from one problem to another, SVM is expected to perform better than NN-Euclidean over image classification problems, which include the problem of filtered-image classification. The reason is that an image is usually represented by a high-dimensional vector in the Euclidean space, and SVM is more suitable than NN-Euclidean for a high dimensional problem as SVM's performance does not depend on the dimensionality of the data [32]. Experimentally, the superior performance of SVM over NN-Euclidean has been observed for the problem of recognizing hand-written digits [33]. Thus, we will not show the experimental results of "NN-MajorityVoting" in this paper.

Table 1
Offline recognition: comparison between RF-NN and RF-SVM.

	Proposed RF-SVM	RF-NN in [15]
Training time (s)	304	447
Classification time (s)	1.4	13.6
ER_{action} (%)	6.0	7.8

The comparison between RF-NN and RF-SVM is performed using a modified cross validation, which is "Leave One Person Out Cross Validation" (LOOCV-Person). In each fold of the LOOCV-Person, all videoclips performed by one person are kept as test videoclips and the classifier is built using the other videoclips. In other words, there are 8 (=1 × 8) test videoclips and 224 (=28 × 8) training videoclips in each cross validation.

The comparison is shown in Table 1, where we can see that the RF-SVM proposed in this paper outperforms the RF-NN proposed in [15] in terms of training time, classification time, and classification accuracy. More specifically, the training time for RF-NN is the time to compute principal directions, averaged over 29 cross validations. The training time for the RF-SVM is the total time to train eight support vector machines (each discriminates one motion type from all the others), averaged over 29 cross validations. For both RF-NN and RF-SVM, the classification time is the average time to classify one videoclip. This average is taken over 232 test videoc-

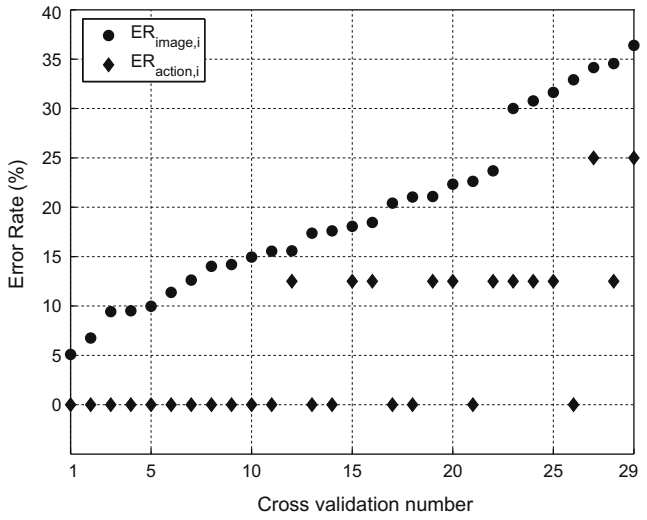


Fig. 3. Offline recognition: comparison of image error rate and action error rate over 29-fold LOOCV-Person. The average of action error rate over 29 cross validations is 6.0% and it is significantly smaller than the average of image error rate, which is 19.7%.

Table 2
Offline recognition: typical example of a confusion matrix for one cross validation. There is 1 (out of 8) action being misclassified and the action error rate is 12.5%. The image error rate is 21.1%. An empty entry means zero.

True label	Predicted label							
	W	R	S	M	LW	H	SW	SS
W			1					
R		1						
S			1					
M				1				
LW					1			
H						1		
SW							1	
SS								1

Table 3
Offline recognition: experimental results on detecting unknown types of motion. A recognition system is built to recognize two types of motions and tested on all three types of motions. The unknown motion type is written in bold font. See text for explanation.

Truth	Predictions		
	Walk (W)	Run (R)	**Rejected**
Case A: hop is unknown			
Walk (W)	19	0	10
Run (R)	0	20	9
Hop (H)	2	2	25
	Walk (W)	**Rejected**	Hop (H)
Case B: run is unknown			
Walk (W)	24	5	0
Run (R)	8	18	3
Hop (H)	0	0	29
	Rejected	Run (R)	Hop (H)
Case C: walk is unknown			
Walk (W)	26	3	0
Run (R)	1	22	6
Hop (H)	0	6	23
	Walk (W)	Run (R)	Hop (H)
Case D: baseline			
Walk (W)	28	0	1
Run (R)	0	28	1
Hop (H)	0	0	29

lips, since there are 29 cross validations and each cross validation has eight test videoclips. The action error rate ER_{action} quantifies the accuracy of RF-SVM and RF-NN, and it is defined as

$$ER_{\text{action}} = \frac{1}{29}\sum_{i=1}^{29} ER_{\text{action},i} = \frac{1}{29}\sum_{i=1}^{29} \frac{NV_{\text{misclassified},i}}{NV_{\text{test},i}}, \tag{27}$$

where $ER_{\text{action},i}$ is the fraction of misclassified test video clips in the ith cross validation, $NV_{\text{misclassified},i}$ is the number of misclassified test videoclips in the ith cross validation, and $NV_{\text{test},i}$ is the number of test videoclips in the ith cross validation.

For the proposed RF-SVM, Fig. 3 compares the action error rate $ER_{\text{action},i}$ versus the image error rate $ER_{\text{image},i}$ over 29 folds cross validations. The image error rate $ER_{\text{image},i}$ is the fraction of misclassified filtered images in the ith cross validation, and it is defined as

$$ER_{\text{image},i} = \frac{NF_{\text{misclassified},i}}{NF_{\text{test},i}}, \tag{28}$$

where $NF_{\text{misclassified},i}$ is the number of misclassified filtered images of all test video clips in the ith cross validation, and $NF_{\text{test},i}$ is the number of filtered images of all test video clips in the ith cross validation. The image error rate is calculated based on the labels of filtered images and is given by step 4 in Algorithm 1. It can be seen from Fig. 3 that $ER_{\text{action},i} < ER_{\text{image},i}$ in all cross validations. This means that *majority voting* is important for the effectiveness of RF-SVM and it makes RF-SVM robust to ambiguous frames in a videoclip. In the ideal case, we can correctly classify an action as long as more than half of its filtered images are classified correctly. This observation is especially important when there are similar gestures or gaits between different types of actions. For example, walking and marching have similarities after both feet touch the ground and the rear leg begins to retract. For completeness, the confusion matrix for a typical cross validation is shown in Table 2.

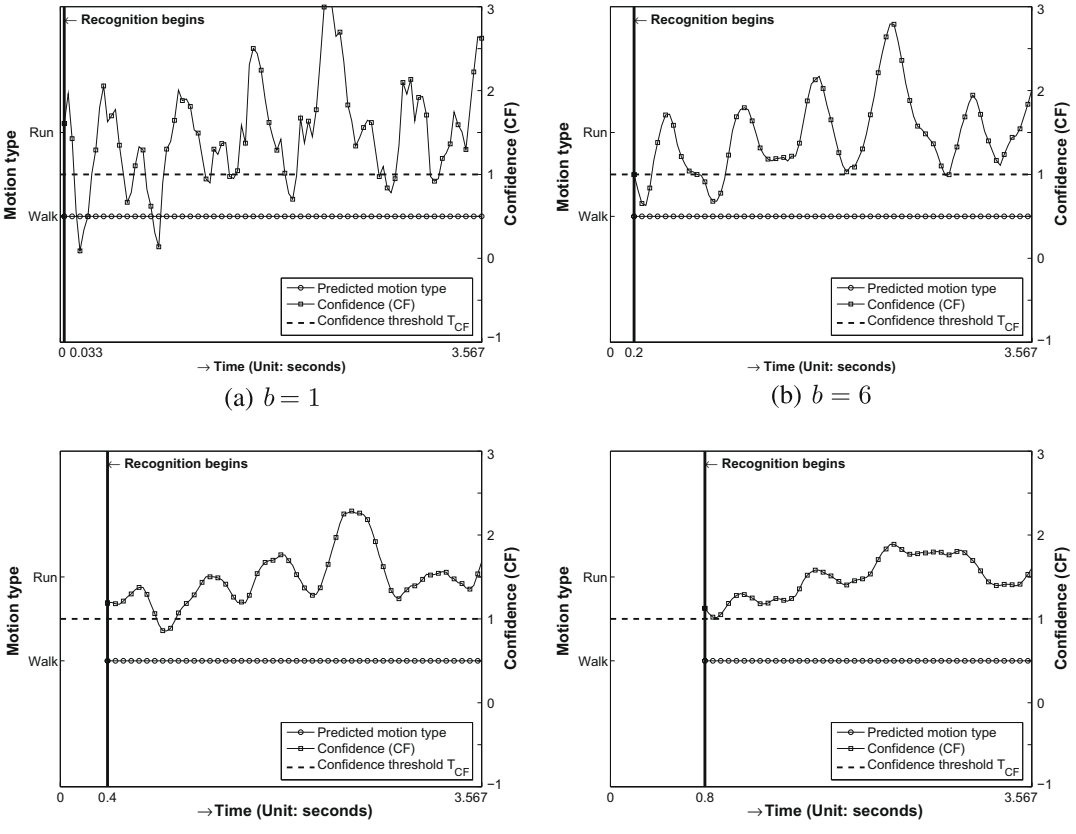


Fig. 4. Online classification for a videoclip of *pure walking* by the system that can *recognize walking and running* using four different values for the width b of the sliding window.

4.2.2. Recognition of unknown types of motion

Here, we show how the proposed motion recognition strategy, together with the proposed classification confidence, can identify the motion types that are not present in the training dataset. For this purpose, we choose three types of motions, which are walk (W), run (R), and hop (H). We built a recognition system that can recognize two types of motions, and tested the system using a test dataset that contains videoclips of all three types. We expect that the videoclips of two known types, which are present in the training dataset, will be recognized correctly with confidence larger than the threshold (cf. Eq. (21)). At the same time, we expect that the videoclips whose classifications are rejected will be the videoclips corresponding to the unknown motion type. The experiment proceeds in a way similiar to cross validation by repeating Algorithm 3 three times and, in each time, one motion type is assumed to be unknown and the other two motion types are assumed to be known. For clarity, we assume that walk and run are known and hop is unknown in Algorithm 3.

Algorithm 3. Experimental setup for offline recognition

1: Denote **A** be the confusion matrix defined as

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \qquad (29)$$

where the meaning of each entry is illustrated as follows:

Truth	Predictions		
	Walk (W)	Run (R)	**Rejected**
Walk (W)	a_{11}	a_{12}	a_{13}
Run (R)	a_{21}	a_{22}	a_{23}
Hop (H)	a_{31}	a_{32}	a_{33}

For example, entry a_{21} is the number of videoclips of run that are predicted as walk

2: Initialize **A** by setting $a_{ij} = 0 \ \forall \ i,j = 1,2,3$

3: **for all** $i = 1,2,\ldots,29$

4: Let the set of testing videoclips $\mathscr{D}_{\text{test}}$ consist of 3 videoclips (walk, run, march) performed by the ith person

5: Train a motion classifier $\mathscr{C}$ that can discriminate two known motion types, i.e., walk and run, using 56 training videoclips, which correspond to the walk and run videoclips performed by the other 28 persons

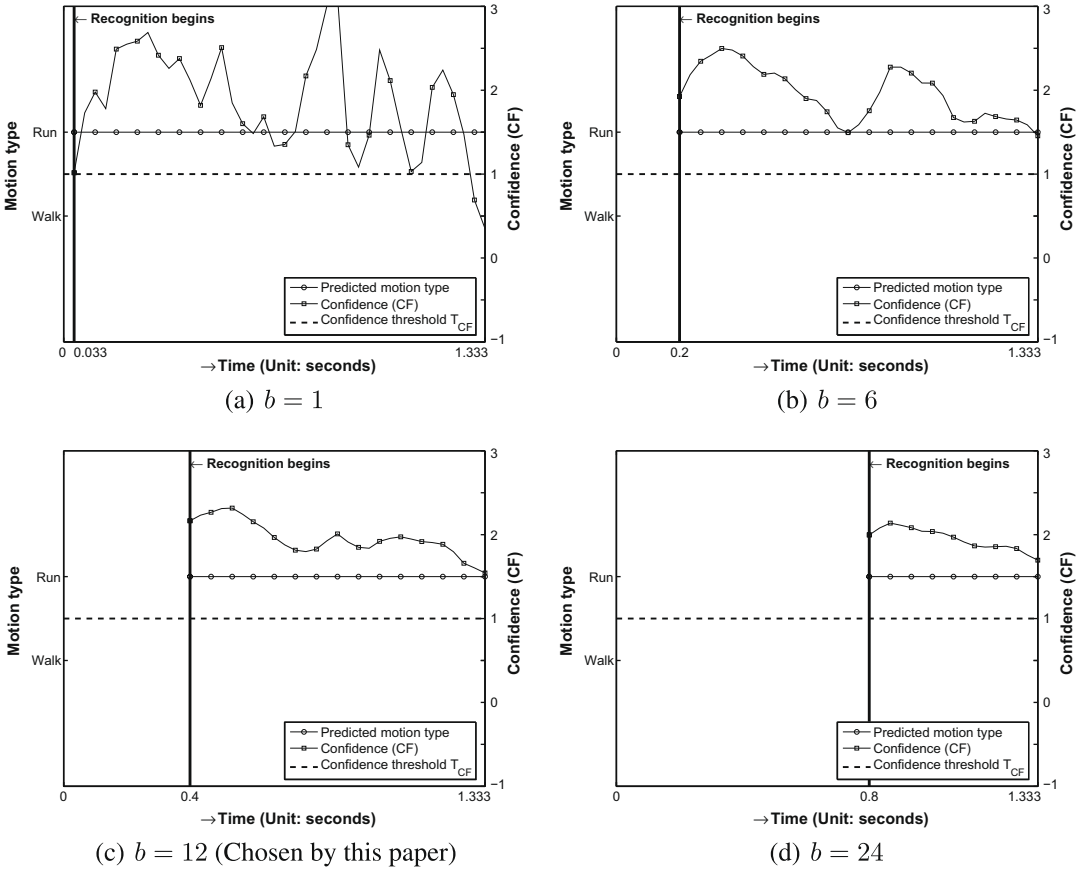
6: Classify $\mathscr{D}_{\text{test}}$ using $\mathscr{C}$, and let the resulting confusion matrix be $\mathbf{A}'$

7: Update **A** by setting $\mathbf{A} = \mathbf{A} + \mathbf{A}'$

8 **end for**

9 Return the confusion matrix **A**

Table 3 shows the experimental results in terms of four confusion matrices, three of which (case A–C) are given by the above procedure with one type of motion being unknown and the other one (Case D) is obtained in a similar way but assuming all three types of motion are known, i.e., the motion classifier is trained using data from all three motion types. Thus, the confusion matrix in case D can be used as a basis to check the quality of confusion matrices in case A–C. In case A–C, the row corresponding to the unknown type of motion is written in bold font and the videoclips of the unknown type should appear in the column **Rejected**, since their classification confidence should be less than the threshold. The close similarity between the confusion matrices in case A–C and the confusion matrix in case D shows that the proposed strat-



(a) $b = 1$ (b) $b = 6$

(c) $b = 12$ (Chosen by this paper) (d) $b = 24$

Fig. 5. Online classification for a videoclip of *pure running* by the system that can *recognize walking and running* using four different values for the width b of the sliding window.

egy is quite effective in identifying unknown types of motion. We will demonstrate this fact further in the next section for online recognition.

4.3. Online recognition

In this section, we demonstrate the performance of the proposed strategy RF-SVM on online motion recognition from two aspects. First, we show that the proposed strategy can reliably recognize known types of motions and quickly detect transitions between two different types of motions. The effects of the width b of the sliding window are also examined. Second, we show that the proposed strategy can identify the existence of unknown types of motions effectively.

To carry out our experiments, we build a system that can recognize walk and run. The set of training videoclips consists of walking and running sequences performed by 28 persons (from the 1st person to the 28th person) and it is a subset of the dataset used in offline recognition. There are 56 videoclips. Each videoclip has various numbers of frames and there are a total of 1841 frames. Each frame corresponds to one filtered image, which is represented by a matrix of size 25×31, i.e., a vector of dimension 775. Since there are only walking and running sequences in the training dataset, this system can *only* recognize walk and run, and any other motion type, like march, is unknown to this system. An unknown type of motion will be identified by a low classification confidence (cf. Eq. (24)). The set of test videoclips consists of the videoclips performed by the 29th person and another person, which is denoted as person 30.

The experimental results are summarized in Figs. 4–10. Unless noted otherwise, the following is true in all these figures: (i) The video is recorded at a rate of 30 frames per second; (ii) With reference to the left Y-axis, the line with circles shows the classifications as a function of time. This line is drawn based on the predictions at all instances, i.e., all frames, which include both odd-numbered frames and even-numbered frames. However, for clarity purposes, only the even-numbered frames are indicated with circles in the figure; and (iii) With reference to the right Y-axis, the line with squares shows the confidence for the classification as a function of time, and the horizontal dashed thick-solid line shows the confidence threshold defined in Eq. (24). The line with squares is drawn based on the confidences at all instances, i.e., all frames, which include both odd-numbered frames and even-numbered frames. However, for clarity purposes, only the even-numbered frames are indicated with squares in the figure.

4.3.1. Recognition of a known type of motion

We first show the online recognition results for videoclips containing only one type of action. The videoclips used here are the walking and running sequences performed by the 30th person. Figs. 4 and 5 show the classification performance for walking and running with four different values for the width b of the sliding window, which are 1, 6, 12, and 24. Since all videos were recorded at a rate of 30 frames per second, the recognition begins 0.033, 0.2, 0.4, and 0.8 s after the video begins, respectively.

With reference to Fig. 4, we can see that the sliding window has the effect of making the recognition robust to noisy frames, i.e., filtered images. Fig. 4(a) shows the results for $b = 1$, which is essen-

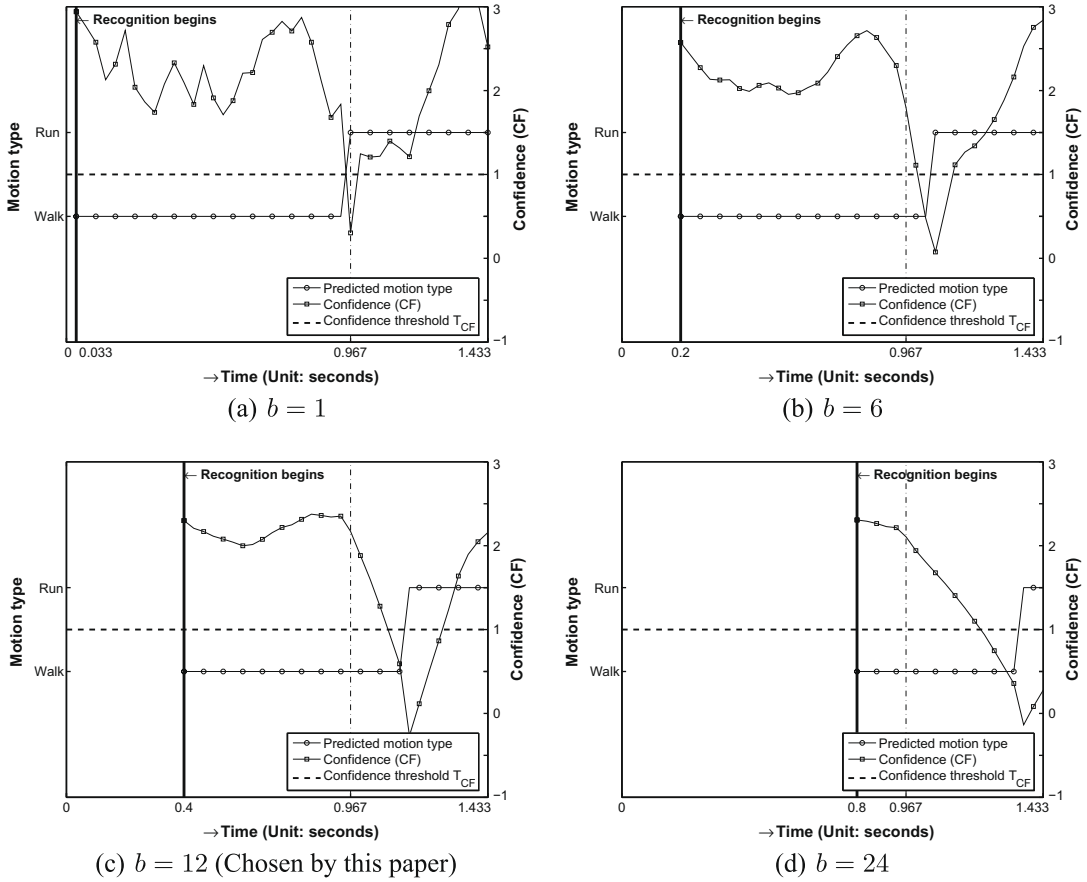


Fig. 6. Online classification for a videoclip having *an artificial transition from walking to running* by the system that can *recognize walking and running* using four different values for the width b of the sliding window. The vertical dash–dot line indicates the time of transition.

tially the label and confidence for each individual frame. Although all frames are correctly classified as "walk", as indicated by the variation of the confidence over time, we can see that some frames are pretty noisy. As b grows larger, the confidence changes more slowly over time. Similar observation can be made by looking at Fig. 5, which shows the classification results on a videoclip of running. Thus, a large sliding window is preferred in terms of the robustness of the recognition system to noisy frames. However, a large sliding window has its own disadvantages when we want to detect the transition between different types of motions, as shown in Figs. 6 and 7.

Figs. 6 and 7 show the ability of the proposed strategy in detecting transitions between different known types of motions. These test videoclips are obtained by manually concatenating a walking videoclip and a running videoclip of the 29th person. The advantage of these "artificial" transitions over the "real" transitions, which are performed by a person, is that the exact time when the transition occurs is known, and it corresponds to the position where two videoclips are concatenated. With reference to Figs. 6 and 7, we can see that the transition between walking and running can be identified and the delay, which is the time elapsed from when a transition happens until such a transition is identified by the system, depends on the size of the sliding window. A large sliding window will give a long delay. For example, when $b = 24$, the delay is more than 0.5 s according to Figs. 6(d) and 7(d). Thus, in terms of prompt identification of motion change, we would prefer a small sliding window. However, as we mentioned in the last paragraph, a recognition system with a small sliding window has the disadvantage of being non-robust to noisy frames. This non-robustness is exemplified in Figs. 7(a) ($b = 1$) and 7(b) ($b = 6$), where we can see the degraded performance in the beginning of the video.

Thus, as a trade-off, we use a sliding window of width $b = 12$ in the rest of this paper. As shown by Figs. 4(c), 5(c), 6(c), and 7(c), with $b = 12$, all known motion types are correctly classified and, except around the transition point, the classification confidence is above the threshold at almost every instance, which demonstrate the effectiveness of the proposed strategy for online motion recognition.

4.3.2. Recognition of unknown types of motion

Here, we show the ability of the proposed strategy in detecting unknown types of actions. Following the discussion in Section 4.3.1, a sliding window of width $b = 12$ is used in this section, which means that the recognition begins 0.4 s after the video begins. The unknown type of action in this experiment is marching.

Fig. 8 shows the classification results for a marching sequence performed by the 30th person. It can be seen that the classification confidence $CF < T_{CF}$ at almost every instance. This means that an action of unknown type has been detected and, in this case, this is marching.

Fig. 9 shows the classification results for the videoclip having a real transition from walking to marching, which was performed by the 30th person. It should be pointed out that, in real life, a transition between different types of actions happens gradually and the exact time when such a transition occurs is not available. Thus, a transition period is manually labeled here, that is, the time when the subject stops walking (thin vertical dash-dot line) and the time when it is obvious that the subject is marching (thin vertical dash line) are both labeled. It can be seen from Fig. 9 that, from the

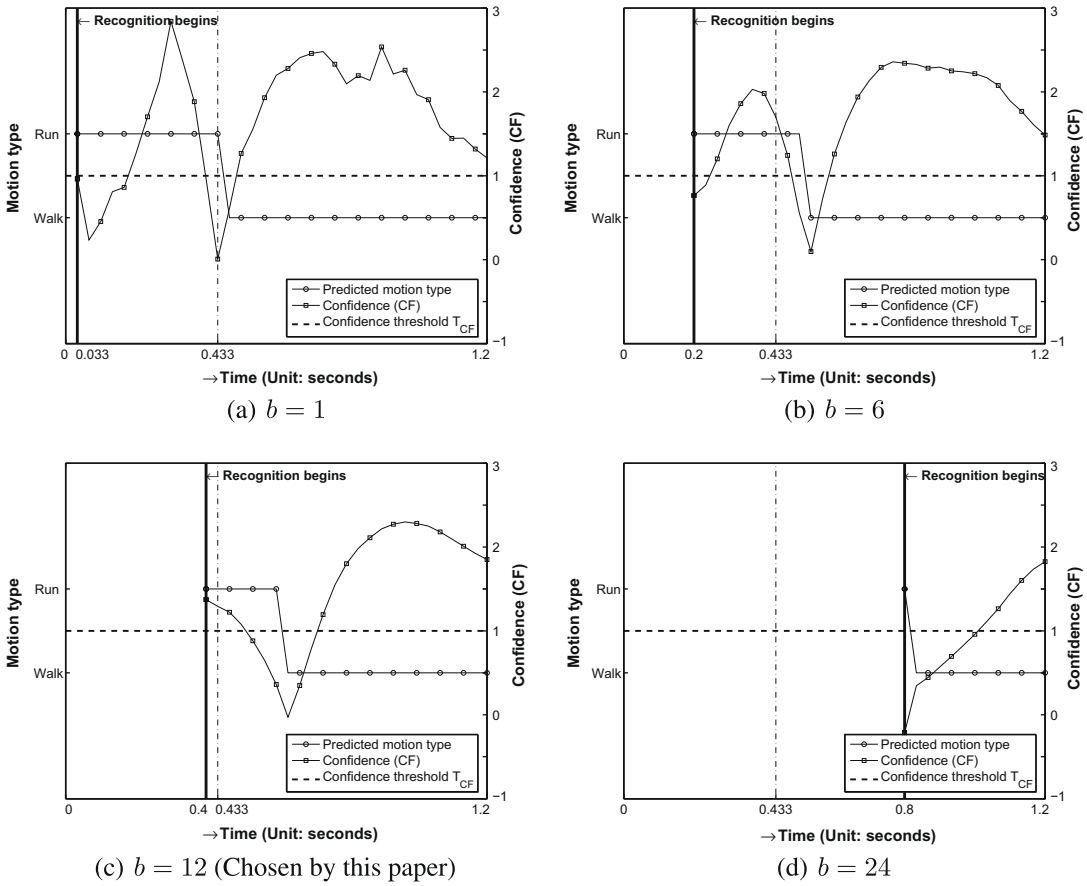


Fig. 7. Online classification for a videoclip having *an artificial transition from running to walking* by the system that can *recognize walking and running* using four different values for the width b of the sliding window. The vertical dash–dot line indicates the time of transition.

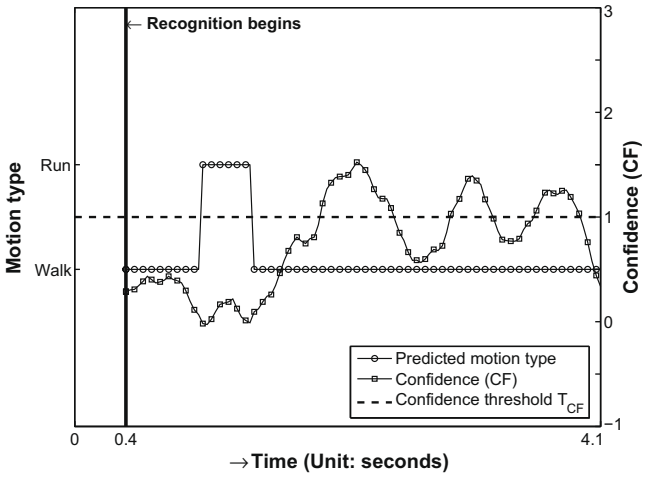


Fig. 8. Online classification for a videoclip of *pure marching* by the system that can *recognize walking and running* only. Notice that the low confidence of the classification makes the classification at almost all instances to be rejected.

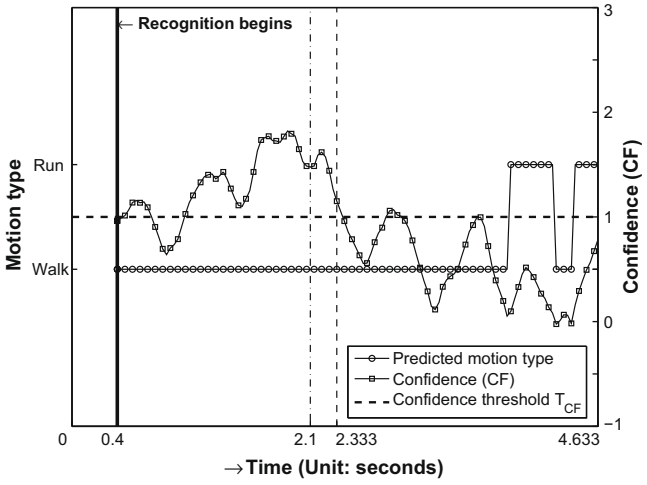


Fig. 9. Online classification for a videoclip having *a real transition from walking to marching* by the system that can *recognize walking and running* only. The time when the subject stops walking is indicated by a **vertical dash–dot line**, and the time when it is obvious that the subject is marching is indicated by a **vertical dash line**.

beginning of walking to the finish of walking, all the classifications are correct and most of the corresponding confidences *CF* are larger than the threshold T_{CF}. During the transition period, that is, from the finish of walking to the beginning of marching, the confidence drops significantly. After the beginning of marching, almost all classifications' confidences are smaller than the threshold T_{CF}, based on which we can tell the existence of an unknown type of motion.

As a comparison, we build another classifier that is capable of recognizing walking and marching. The training dataset consists of walking and marching sequences performed by the same 28 persons. Fig. 10 shows the classification results. We can see that, except during the transition period, both the walking part and the marching part are classified correctly with confidence *CF* being larger than the threshold T_{CF} at most instances.

Combining Figs. 8–10, we can see that the proposed online recognition strategy is capable of not only recognizing *known* motion types but also detecting the existence of *unknown* motion types.

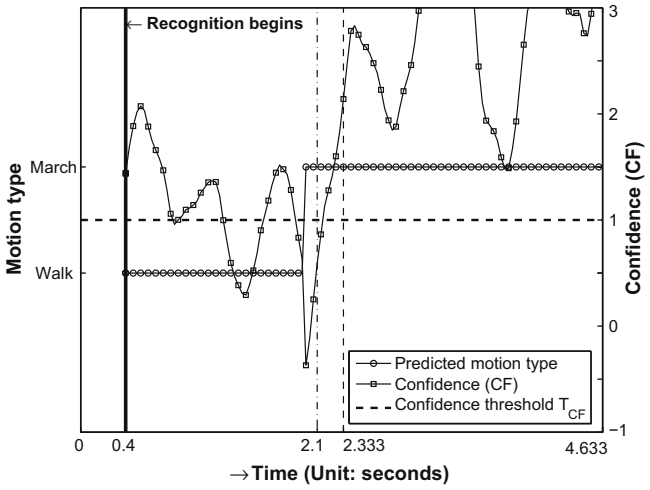


Fig. 10. Online classification for a videoclip having *a real transition from walking to marching* by the system that can *recognize walking and marching* only. The time when the subject stops walking is indicated by a **vertical dash–dot line** and the time when it is obvious that the subject is marching is indicated by a **vertical dash line**.

5. Discussion and future work

A novel human motion recognition strategy is proposed in this paper, which combines the technique of recursive filtering and frame grouping and uses the powerful classification algorithm of the support vector machine. This strategy solves the problem of motion recognition by classifying filtered images using support vector machines and applying majority voting over the resulting labels. The effectiveness of the proposed strategy is demonstrated using real datasets for both the offline recognition and the online recognition. The performance of offline recognition is shown to be superior to a strategy based on PCA and Hausdorff distance in terms of training time, classification time, and classification accuracy. In online recognition, the proposed strategy can identify actions of known types and the transitions between different types of actions. For both offline and online recognitions, actions of unknown types can also be identified reliably.

The current research can be extended in the following directions. The first direction is to build an adaptive online recognition system that can update itself after detecting actions of unknown types. The challenges here include how to pick up the data representing the unknown types of motions, how to separate more than two unknown motion types, and how to update the system efficiently.

The ability of real-time recognition is critical in many applications and, thus, the second direction is to implement a *real-time* recognition system. The rationality of this direction is demonstrated in Table 4, which shows the average time to decide the type of motion at an instance. Knowing the fact that a videoclip is recorded at a rate of 30 frames per second, which is 0.033 s per frame, the proposed strategy is fast enough for real-time classifica-

Table 4
Online recognition: time needed to determine the most probable type of motion at an instance.

No.	Steps	Time (s)
1.	Recursive filtering	0.0109
2.	Classifying filtered images	0.0182
3.	Majority voting	<0.00005
Total	0.0291	

tion. It should be noted that the time for tracking is not included in Table 4 since tracking was run on a different computer separately. In addition, it is well-known that the classification time of SVM is roughly proportional to the number of support vectors and, when the number of training data is large, the number of support vectors usually turns out to be large. Thus, when we have huge numbers of labeled videoclips, the time to classify filtered images may be longer than that presented in Table 4. Fortunately, as shown in [34], the classification speed illustrated in Table 4 can be further improved with little influence to the accuracy.

Third, it is assumed in the current study that the subject performing the motion is located in the center of the filtered image. This is accomplished through tracking and it may be violated when, for example, we try to speed up the classification process by using a less accurate tracking algorithm. When the subject is not always located in the center of the image, we need to build a translation invariant filtered-image classifier. Techniques that will be explored include using virtual support vector machines and using tangent distance based kernels [31].

Acknowledgments

The authors would like to express their gratitude to the anonymous reviewers for their thoughtful comments. This work has been supported in part by the National Science Foundation through Grants #IIS-0219863, #IIS-0208621, #IIS-0534286, #CNS-0224363, #CNS-0324864, #CNS-0420836, #IIP-0443945, #IIP-0726109, #CNS-0708344 and #CNS-0821474, the US Army Research Laboratory, the US Army Research Office under Contract No. 911NF-08-1-0463 (Proposal 55111-CI), the Minnesota Department of Transportation, and the ITS Institute at the University of Minnesota.

References

[1] N.H. Goddard, The perception of articulated motion: recognizing moving light displays, Ph.D. Thesis, University of Rochester, 1992.
[2] Y. Guo, G. Xue, S. Tsuji, Understanding human motion patterns, in: 12th International Conference on Pattern Recognition, 1994, pp. 325–329.
[3] W.H. Dittrich, Action categories and the perception of biological motion, Perception 22 (1993) 15–22.
[4] Y. Yacoob, M.J. Black, Parameterized modeling and recognition of activities, Computer Vision and Image Understanding 73 (2) (1999) 232–247.
[5] C. Bregler, Learning and recognizing human dynamics in video sequences, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 568–574.
[6] K. Rangarajan, W. Allen, M. Shah, Matching motion trajectories using scale space, Pattern Recognition 26 (4) (1993) 595–610.
[7] V. Pavlovic, J. Rehg, Impact of dynamic model learning on classification of human motion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000, pp. 788–795.
[8] L.W. Campbell, A.F. Bobick, Recognition of human body motion using phase space constraints, in: International Conference on Computer Vision, 1995, pp. 624–630.
[9] D.M. Gavrila, L.S. Davis, 3D model-based tracking of humans in action: a multiview approach, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1996, pp. 73–80.
[10] N. Krahnstöver, M. Yeasin, R. Sharma, Towards a unified framework for tracking and analysis of human motion, in: Proceedings of the IEEE Workshop on Detection and Recognition of Events in Video, 2001, pp. 47–54.
[11] J. Yamato, J. Ohya, K. Ishii, Recognizing human action in time-sequential images using a Hidden Markov Model, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1992, pp. 379–385.
[12] J.W. Davis, A.F. Bobick, The representation and recognition of human movement using temporal templates, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 928–934.
[13] R. Polana, R.C. Nelson, Low level recognition of human motion, in: Proceedings of the IEEE Workshop on Non-rigid Motion, 1994, pp. 77–82.
[14] H. Meng, N. Pears, M. Freeman, C. Bailey, Motion history histograms for human action recognition, in: Embedded Computer Vision, 2009, p. 139.
[15] O. Masoud, N. Papanikolopoulos, A method for human action recognition, Image and Vision Computing 21 (2003) 729–743.
[16] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[17] B. Schölkopf, P. Simard, A. Smola, V.N. Vapnik, Prior knowledge in support vector kernels, in: M.I. Jordan, M.J. Kearns, S.A. Solla (Eds.), Advances in Neural Information Processing Systems, vol. 10, 1998, pp. 640–646.
[18] C. Schüldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in: International Conference on Pattern Recognition, vol. III, Cambridge, UK, 2004, pp. 32–36.
[19] T. Mori, M. Shimosaka, T. Sato, SVM-based human action recognition and its remarkable motion features discovery algorithm, in: International Symposium on Experimental Robotics, ISER2004, Singapore, 2004.
[20] R. Bodor, B. Jackson, O. Masoud, N. Papanikolopoulos, Image-based reconstruction for view-independent human motion recognition, in: IEEE International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, 2003, pp. 1548–1553.
[21] E. Osuna, R. Freund, F. Girosi, An improved training algorithm for support vector machines, in: A. Island (Ed.), IEEE Neural Networks for Signal Processing VII Workshop, 1997, pp. 276–285.
[22] T. Joachims, Making large-scale support vector machine learning practical, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Boca Raton, FL, 1999, pp. 169–184.
[23] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Boca Raton, FL, 1999, pp. 185–208.
[24] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, IEEE Transactions on Neural Networks 11 (1) (2000) 124–136.
[25] D.L. Boley, D. Cao, Training support vector machine using adaptive clustering, in: M.W. Berry, U. Dayal, C. Kamath, D. Skillicorn (Eds.), Proceedings of the 4th SIAM International Conference on Data Mining, SIAM, Lake Buena Vista, FL, 2004, pp. 126–137.
[26] D. Cao, D.L. Boley, On approximate solutions to support vector machines, in: Proceedings of the 6th SIAM International Conference on Data Mining, SIAM, Bethesda, MD, 2006, pp. 534–538.
[27] R.-E. Fan, P.-H. Chen, C.-J. Lin, Working set selection using second order information for training support vector machines, Journal of Machine Learning Research 6 (2005) 1889–1918.
[28] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley–Interscience, New York, 2000.
[29] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: A.J. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (Eds.), Advances in Large Margin Classifiers, MIT Press, Boca Raton, FL, 2000, pp. 61–74.
[30] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, R. Williamson, Estimating the support of a high-dimensional distribution, Tech. Rep. MST-TR-99-87, Microsoft Research, November 1999.
[31] B. Schölkopf, A.J. Smola, Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Boca Raton, FL, 2002.
[32] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: C. Nédellec, C. Rouveirol (Eds.), Proceedings of the European Conference on Machine Learning, Springer, Chemnitz, DE, 1998, pp. 137–142.
[33] Y. LeCun, L.D. Jackel, L. Bottou, A. Brunot, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, V. Vapnik, Comparison of learning algorithms for handwritten digit recognition, in: F. Fogelman-Soulie, P. Gallinari (Eds.), ICANN, vol. 2, 1995, pp. 53–60.
[34] E.E. Osuna, F. Girosi, Reducing the run-time complexity in support vector machines, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Boca Raton, FL, 1999, pp. 271–283.