Pattern Recognition Letters 30 (2009) 1049–1058

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Descriptive temporal template features for visual motion recognition

Hongying Meng [a,*], Nick Pears [b]

[a] Department of Computing and Informatics, University of Lincoln, Lincoln, LN6 7TS, UK
[b] Department of Computer Science, University of York, UK

ARTICLE INFO

Article history:
Available online 20 March 2009

Keywords:
Gesture recognition
Event recognition
Embedded vision
Motion analysis
Machine learning

ABSTRACT

In this paper, a human action recognition system is proposed. The system is based on new, descriptive 'temporal template' features in order to achieve high-speed recognition in real-time, embedded applications. The limitations of the well-known 'Motion History Image' (MHI) temporal template are addressed and a new 'Motion History Histogram' (MHH) feature is proposed to capture more motion information in the video. MHH not only provides rich motion information, but also remains computationally inexpensive. To further improve classification performance, we combine both MHI and MHH into a low dimensional feature vector which is processed by a support vector machine (SVM). Experimental results show that our new representation can achieve a significant improvement in the performance of human action recognition over existing comparable methods, which use 2D temporal template based representations.

1. Introduction

In this paper, we focus on the use of video for monitoring human events, with a view to real-time, embedded implementations in security systems, human–computer interaction and intelligent environments. Such events may be prescribed gestures, or simple actions, such as walking in a particular direction across the scene. In particular, our methods are designed to be appropriate for deployment in a real-time, embedded context. In this sense, we aim for compact, yet descriptive image-based (2D) motion representations, which are simple to compute. Also we aim for low complexity classification algorithms, so that both feature extraction and classification may be implemented on our flexible stand-alone video processing architecture, which is based upon field-programmable gate arrays (FPGAs). A detailed description of this hardware can be found in our previous publications (Meng et al., 2007a, 2008).

Generally, event detection in video has become an important computer vision application, particularly in the context of activity classification (Aggarwal and Cai, 1999). For human activity or behaviour recognition, many efforts have been concentrated on using state-space methods (Farmer et al., 1991) to understand human motion sequences (Bobick and Davis, 2001; Campbell and Bobick, 1995; Gao et al., 2004; Nascimento et al., 2005; Davis and Tyagi, 2006). However, these methods usually need intrinsic nonlinear models and do not have a closed-form solution. Previous work on motion descriptors uses positions and velocities of human body parts (Green and Guan, 2004), but such information is often difficult to extract automatically during unrestricted human activities.

Aggarwal and Cai (1999) present an excellent overview of human motion analysis. Of the appearance based methods, template matching has gained increasing interest (Schuldt et al., 2004; Weinland et al., 2005; Ke et al., 2005; Wong and Cipolla, 2005; Dollár et al., 2005; Niebles et al., 2006; Ogata et al., 2006; Dalal et al., 2006; Blank et al., 2005; Oikonomopoulos et al., 2006; Meng et al., 2006a,b; Wong and Cipolla, 2006; Yeo et al., 2006). An example of a temporal template motion representation, the 'Motion History Image' (MHI) (Bobick and Davis, 2001), is given in Fig. 1, where (a) is one frame from the original video clip and (b) is the MHI of this action. It is clear that MHI is an image, with the same size as the original frame, but which retains motion information associated with the action. To compute an MHI, we start by computing a foreground motion segmentation, which may be done by simple frame differencing or more complex background modelling, such as Gaussian Mixture Models (Stauffer and Grimson, 2000). The MHI is then generated as the weighted sum of past foreground segmentations and the weights decay back through time. Therefore, an MHI image contains the past foreground segmentations within itself, where most recent foreground motion is brighter than past ones. (This is mathematically defined in Section 3.)

We have elected to further develop this 'temporal template' approach. However, given that the extracted motion information from temporal templates is appearance based, we have the limitations of (i) viewpoint dependence and (ii) loss of information in the projection from 3D to 2D. Both of these problems are mitigated if the motion is predominantly parallel to the image plane. In the case of gestures, these can be designed to be in an appropriate direction relative to the camera. In the context of less deliberative,

* Corresponding author. Tel.: +44 1522 88 6974.
E-mail address: yorksman2005@gmail.com (H. Meng).

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.03.003

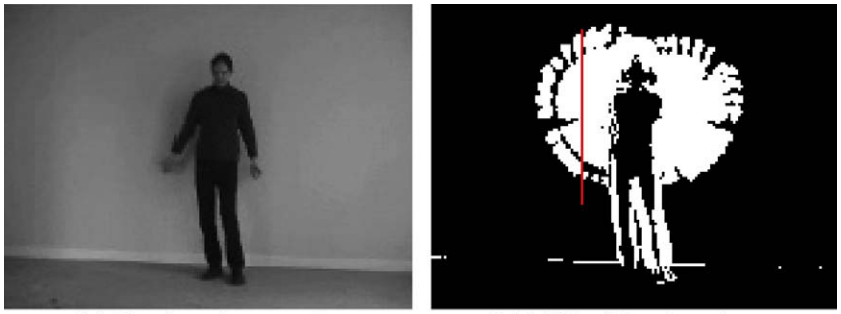

Fig. 1. Example of an MHI. (a) is one frame from the original hand-waving action video clip and (b) is the MHI of this action. The vertical red line in (b) has the pixels from (60,11) to (60,80). (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

natural human motion, the problem can only be solved by implementing multiple viewpoints and selecting the most appropriate data stream. This was a necessary compromise to allow us to implement a real-time embedded solution to our pattern recognition problem. Given that we have these restrictions, the problem that we wished to solve was how to extract more information in order to improve the state-of-the-art in *temporal template* based motion recognition approaches, while retaining a compact feature vector, that is easily deployed on embedded hardware.

In the following, we describe a human action recognition system based on a linear Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000). We address the limitations of the well-known Motion History Image (MHI) (Bobick and Davis, 2001), which is the pioneering work in temporal templates, and propose a new feature, which we call the Motion History Histogram (MHH). This representation expresses more motion information than the MHI, but also remains inexpensive to compute. We show that the MHH and its derivatives outperform the MHI in terms of action recognition on a large public database. Finally, we extract a compact feature vector from the MHH and then combine it with a 'histogram of MHI' feature to give a better classification performance than either feature type alone.

The rest of this paper is organized as follows: In Section 2, we give an overview of related work. In Section 3, we briefly review the MHI and its limitation. In Section 4, we give a detailed description of our new MHH temporal template features, designed to overcome the MHI limitation, and their compact derivatives. In Section 5, we discuss dimension reduction and feature combination, applied to MHI and MHH. In Section 6, experimental results are evaluated and, finally, we present conclusions. The work presented here is an extension of our previous work (Meng et al., 2006b, 2007b).

2. Related work

Bobick and Davis (2001) pioneered the idea of temporal templates (Moeslund et al., 2006). They used Motion Energy Images (MEI) and MHI to recognize many types of aerobics exercise. Bradski and Davis (2002) proposed the Motion Gradient Orientation (MGO) to explicitly encode changes in an image introduced by motion events.

Davis (2001) also presented a useful hierarchical extension for computing a local motion field from the original MHI representation. The MHI was transformed into an image pyramid, permitting efficient fixed-size gradient masks to be convolved at all levels of the pyramid, thus extracting motion information at a wide range of speeds. The hierarchical MHI approach remains a computationally inexpensive algorithm to represent, characterize, and recognize human motion in video.

Schuldt et al. (2004) proposed a method for recognizing complex motion patterns based on local space–time features in video and they integrated such representations with SVM classification schemes for recognition.

The work of Efros et al. (2003) focuses on the case of low resolution video of human behaviours, targeting what they refer to as the 30 pixel man. In this setting, they propose a spatio-temporal descriptor based on optical flow measurements, and apply it to recognize actions in ballet, tennis and football datasets.

Weinland et al. (2005) introduced Motion History Volumes (MHV) as a free-viewpoint representation for human actions in the case of multiple calibrated, and background-subtracted, video. They presented algorithms for computing, aligning and comparing MHVs of different actions performed by different people from a variety of viewpoints.

Ke et al. (2005) studied the use of volumetric features as an alternative to the local descriptor approaches for event detection in video sequences. They generalized the notion of 2D box features to 3D spatio-temporal volumetric features. They constructed a real-time event detector for each action of interest by learning a cascade of filters based on volumetric features that efficiently scanned video sequences in space and time.

Ogata et al. (2006) proposed Modified Motion History Images (MMHI) and used an eigenspace technique to realize high-speed recognition of six human motions.

Wong and Cipolla (2005) proposed a new method to recognize primitive movements based on MGO extraction and, later, used it for continuous gesture recognition (Wong and Cipolla, 2006).

Recently, Dalal et al. (2006) proposed Histogram of Oriented Gradient (HOG) appearance descriptors for image sequences and developed a detector for standing and moving people in video.

Dollár et al. (2005) proposed a similar method where they use a new spatio-temporal interest point detector to obtain a global measurement instead of the local features in Efros et al. (2003). Niebles et al. (2006) also use spatial-time interest points to extract spatial–temporal words as their features. Yeo et al. (2006) estimate motion vectors from optical flow and calculate frame-to-frame motion similarity to analyse human action in video.

Blank et al. (2005) regarded human actions as three-dimensional shapes induced by silhouettes in space–time volume. They adopted an approach for analyzing 2D shapes and generalized it to deal with volumetric space–time action shapes.

Oikonomopoulos et al. (2006) introduced a sparse representation of image sequences as a collection of spatio-temporal events that were localized at points that were salient both in space and time for human action recognition.

We note that, in some of these methods, the motion features employed are relatively complex (Efros et al., 2003; Schuldt et al., 2004; Weinland et al., 2005; Niebles et al., 2006; Dalal et al.,

2006; Dollár et al., 2005; Blank et al., 2005; Oikonomopoulos et al., 2006; Ke et al., 2005; Yeo et al., 2006), which implies significant computational cost when building the features. Some of them require tracking or other prohibitive computational cost processes (Bobick and Davis, 2001; Bradski and Davis, 2002; **?**; Wong and Cipolla, 2005, 2006; Ogata et al., 2006; Blank et al., 2005), which, for the time being, makes them not suitable for real-time embedded vision applications.

3. The Motion History Image

Several representations have been proposed to compact the whole motion sequence of a video clip into one image to represent the motion. Such images are often termed 'temporal templates'. The most popular 'temporal template' motion feature is the Motion History Image (MHI) (Bobick and Davis, 2001). We now review this in outline and then we discuss its main limitation, that we aim to overcome, by design of a new feature type.

An MHI is the weighted sum of past images and the weights decay back through time. Therefore, an MHI image contains the past images within itself, in which the most recent image is brighter than past ones. Normally, an MHI $H_\tau(u,v,k)$ at time k and location (u,v) is defined by the following Eq. 1:

$$H_\tau(u,v,k) = \begin{cases} \tau, & \text{if } D(u,v,k)=1, \\ \max\{0, H_\tau(u,v,k-1)-1\}, & \text{otherwise}, \end{cases} \quad (1)$$

where the motion mask $D(u,v,k)$ is a binary image representing the foreground motion, obtained from subtraction of frames, and τ is the maximum duration a motion is stored. In general, τ is chosen as the constant 255, allowing the MHI to be easily represented as a gray scale image with one byte depth. Thus an MHI pixel can have a range of values, whereas an MEI (motion energy image) is its binary version, that can easily be computed by thresholding $H_\tau > 0$.

Fig. 1 shows an example of the MHI motion features. Although we do not see the original video clip of the action, we can still determine that the subject moved both arms above their head. An MHI can be generated frame by frame and only three frames (previous frame, current frame and MHI) need to be stored at any one time. The final output associated with each action is an image, which has the same size as the original input frames.

3.1. Limitations of the MHI

In order to have a detailed look at the MHI, we have selected the pixels on the red line in the MHI of Fig. 1b. If some action happened at frame k on pixel (u,v), then $D(u,v,k)=1$, otherwise $D(u,v,k)=0$. The locations of these pixels are $(60,11),(60,12),\ldots,(60,80)$. For a pixel (u,v), the motion mask $D(u,v,:)$ of this pixel is the binary sequence:

$$D(u,v,:) = (b_1, b_2, \ldots, b_N), \quad b_i \in \{0,1\}, \quad (2)$$

where $N+1$ is the total number of frames (as we need two images at the start of the sequence to generate b_1).

All of the motion masks on the red line are shown in Fig. 2. Each row is $D(u,v,:)$ for one fixed pixel (u,v) and a white block represents '1' and black block represents '0' in the sequences. The green line is the motion mark $D(60,50,:)$ and it has the following sequence (3):

$$00000000011010000000000000000000000001010000. \quad (3)$$

From the definition of MHI in Eq. 1 it can be observed that, for each pixel (u,v), MHI actually retains the time since the last action occurred. That is, only the last '1' in the sequence (3) is retained in the MHI at pixel (60,50). It is clear that previous '1's in the sequence, when some action occurred, are not represented. It is also clear that almost all the pixels have more than one '1' in their sequence, implying that much motion information is lost when an MHI is generated.

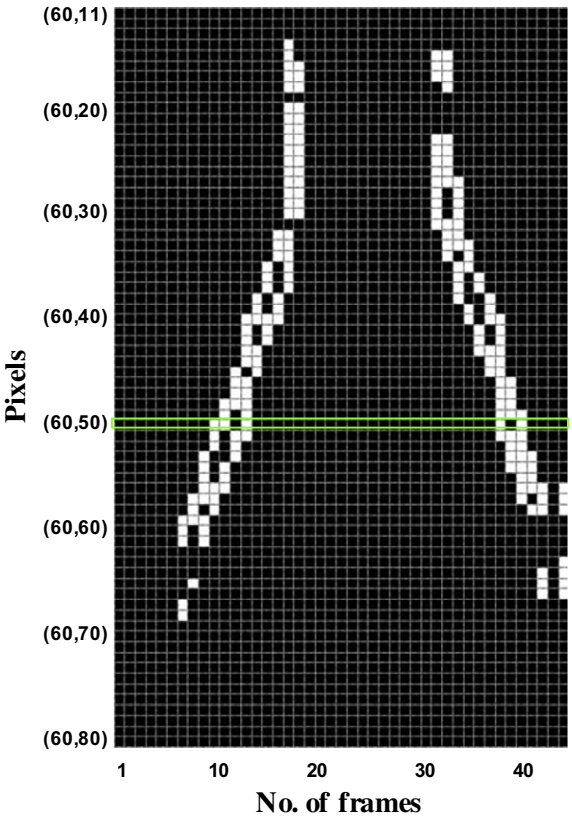

Fig. 2. $D(:,:,:)$ on the red line of Fig. 1b is shown. Each row is $D(u,v,:)$ for one fixed pixel (u,v). A white block represents '1' and a black block '0'. The horizontal green line is the 'binary frame difference history' or 'motion mask' of pixel (60,50) through time, ie, $D(60,50,:)$. (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

4. Motion History Histograms (MHH)

The above limitation of the MHI has motivated us to design a new representation (the MHH) in which all of the information in the sequence is used and, yet, it remains compact and efficient to use.

We define the patterns P_i in the $D(u,v,:)$ sequences, based on the number of connected '1's, as shown in Fig. 3:

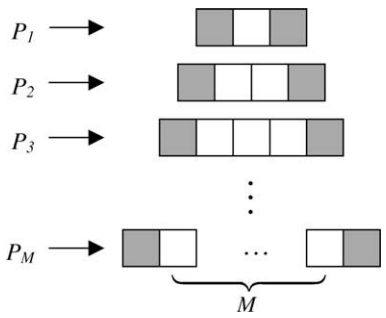

Fig. 3. The patterns P_i in $D(u,v,:)$. White and black blocks represent '1' and '0', respectively.

$$\begin{aligned} P_1 &= 010, \\ P_2 &= 0110, \\ P_3 &= 01110, \\ &\vdots \\ P_M &= 0\underbrace{1\cdots1}_{M}0. \end{aligned} \tag{4}$$

We denote a subsequence $C_{I,k}$ by Eq. (5), where I and k are the indices of starting and ending frames, and denote the set of all subsequences of $D(u,v,:)$ as $\Omega\{D(u,v,:)\}$. Then, for each pixel (u,v), we can count the number of occurrences of each specific pattern P_i in the sequence $D(u,v,:)$, as shown in Eq. (6), where $\mathbf{1}$ is the indicator function

$$C_{I,k} = b_I, b_{I+1}, \ldots, b_k, \quad 1 \leqslant I < k \leqslant N, \tag{5}$$

$$\mathrm{MHH}(u,v,i) = \sum_{(I,k)} \mathbf{1}_{\{C_{I,k}=P_i | C_{I,k} \in \Omega\{D(u,v,:)\}\}}, \quad 1 \leqslant I < k \leqslant N, 1 \leqslant i \leqslant M. \tag{6}$$

From each pattern P_i, we can build a gray scale image and we call this its 'Histogram', since the bin value records the number of this pattern type. With all the patterns $P_i,\ i = 1,\ldots,M$ together, we collectively call them the 'Motion History Histograms' (MHH) representation.

The computation of MHH is inexpensive and can be implemented by the following procedure. $D(u,v,k)$ is the binary sequences on pixel (u,v) that is computed by thresholding the differences between frame k and frame $k-1$. $I(u,v)$ is a frame index that stands for the number of the starting frame of a new pattern on pixel (u,v). At the beginning, $I(u,v) = 1$ for all (u,v). That means a new pattern starts from frame 1 for every pixel. $I(u,v)$ will be updated to $I(u,v) = k$ while $\{D(u,v,I(u,v)),\ldots,D(u,v,k)\}$ builds one of the patterns P_i $(1 \leqslant i \leqslant M)$ and, in this case, $\mathrm{MHH}(u,v,i)$ increases by 1.

Algorithm (MHH).

Input: Video clip $f(u,v,k), u = 1,\ldots,U, v = 1,\ldots,V$, frame $k = 0,1,\ldots,N$
Initialisation: Pattern M, $\mathrm{MHH}(1:U,1:V,1:M) = 0$, $I(1:U,1:V) = 1$
For $k = 1$ to N (For 1)
  Compute: $D(:,:,k)$
  For $u = 1$ to U (For 2)
    For $v = 1$ to V (For 3)
      If subsequence $\{D(u,v,I(u,v)),\ldots,D(u,v,k)\} = P_i$
        Update: $\mathrm{MHH}(u,v,i) = \mathrm{MHH}(u,v,i) + 1$
      End
      Update: $I(u,v)$
    End (For 3)
  End (For 2)
End (For 1)
Output: $\mathrm{MHH}(1:U,1:V,1:M)$

For a pattern P_i, $\mathrm{MHH}(:,:,i)$ can be displayed as an image. In Fig. 4, four patterns $P_1,\ P_2,\ P_3$ and P_4 are shown, which were generated from the hand-waving action in Fig. 1. By comparing the MHH in Fig. 4 with the MHI in Fig. 1, it is interesting to find that the MHH decomposes the MHI into different parts based on patterns. Unlike the hierarchical MHI described by Davis (2001), where only small size MHIs were obtained, MHH records the rich spatial information of an action.

The choice of the number M depends on the video clips. In general, the bigger the M is, the better the motion information will be. However, the values within the MHH rapidly approach zero as M increases. In our experiment, no more than half of the training data had the sixth pattern P_6 and so we chose $M = 5$. Furthermore, we note that a large M will increase the storage requirement for our hardware based system.

We can define the binary version of an MHH as MHH_b, as shown in Eq. (7), and the motivation for this is given in the following section

$$\mathrm{MHH}_b(u,v,i) = \begin{cases} 1, & \text{if } \mathrm{MHH}(u,v,i) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

5. Dimension reduction and feature combination

As previously stated, our work is targeted at embedded implementations using FPGA platforms. This requires us to use compact feature vectors, which retain discriminating information across the various classes of interest in a particular application. Also, the process of compacting must be simple, in order to generate a small footprint (gate count) solution on our embedded devices. The following subsections describe how we employ histograms of both MHI and MHH representations, to compact their representations, and then combine them via simple feature vector concatenation.

5.1. MHI Histograms

MHIs can be rendered as gray scale images, where a value of a pixel in the MHI records time information, namely when some motion most recently occurred at this particular pixel location. Thus the histogram of MHI captures information representing the speed of the motion across the image. Since an MHI is an image with values between 0 and 255, an MHI histogram is a vector of length 256. An example of an MHI histogram is given at the bottom-left of Fig. 5. For fast motions, many pixels in the MHI will have high values, whereas, for slow motions, only a few pixels in the MHI will have high values, thus these different cases will have quite different MHI histograms.

5.2. Motion Geometric Distribution (MGD)

The size of the MHH_b representation is rather large and we seek a more compact representation, which captures the geometric distribution of the motion across the image. Thus we sum each row of MHH_b (for a given pattern, P_i) to give a vector of size V rows. We obtain another vector by summing columns to give a vector of size U rows. Thus using all M levels in the binarised MHH hierarchy, we obtain a 'Motion Geometric Distribution' (MGD) vector of size $M \times (U+V)$, which is relatively compact, when compared to the size of the original MHH and MHI features. The MGD vector can thus be represented by the following Eq. (8):

$$\mathrm{MGD} = \left\{ \sum_u \mathrm{MHH}_b(u,v,i), \sum_v \mathrm{MHH}_b(u,v,i) \right\}, \quad i = 1,2,\ldots,M. \tag{8}$$

In our work, we prefer to compute the MGD by using the MHH_b feature instead of the MHH feature directly. From our experiments, it has been found that the values within the MHH decrease significantly for the large patterns. The values for P_4 and P_5, for example, are much smaller than those of P_1, P_2 and P_3. Thus, if we use the MHH directly to compute the MGD, a normalisation process is necessary in order to treat all the patterns equally. However, this normalisation process is not an easy task for our hardware implementation because of limited memory and the requirement to implement a floating-point processing ability. In contrast, computation of the MGD from the MHH_b feature does not need a normalisation process and yet we are able to retain a good performance.

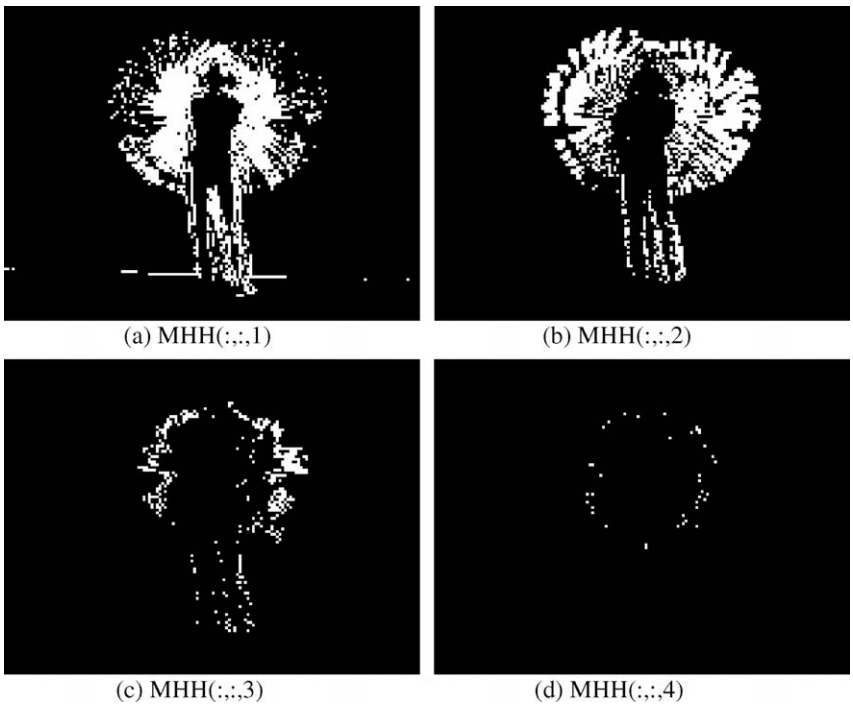

Fig. 4. MHH example. Four patterns P_1, P_2, P_3 and P_4 were selected. This results were generated from the hand-waving action in Fig. 1. Each pattern P_i, MHH$(:,:,i)$ has the same size as the original frame.

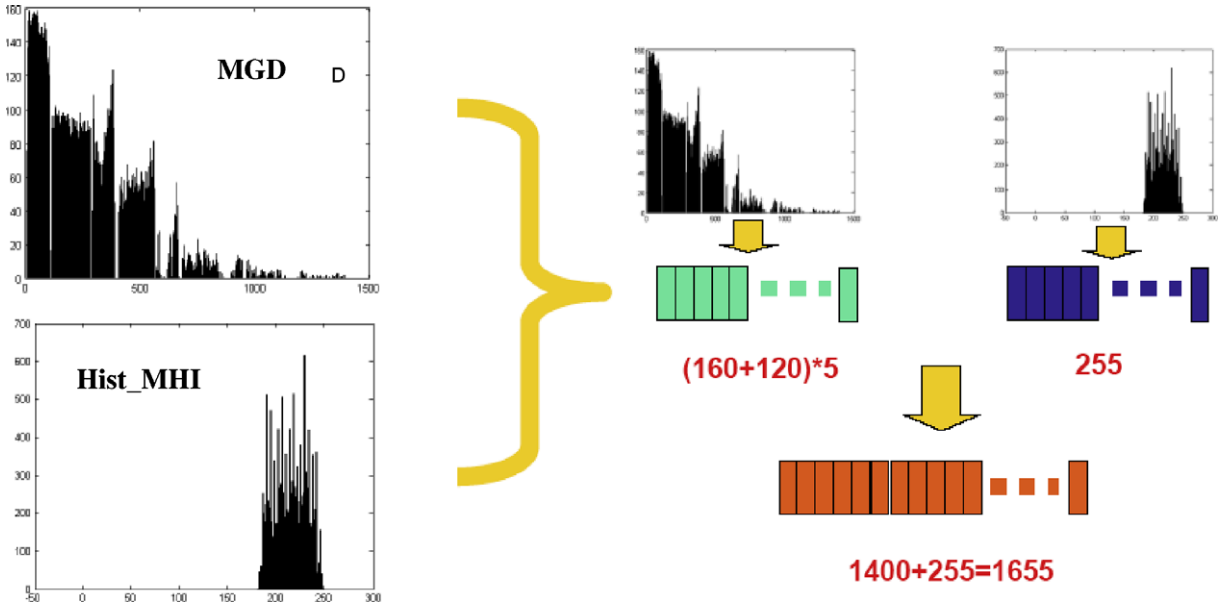

Fig. 5. Combination between MGD of the MHH and histogram of the MHI from a same video example. The frame has the size of 160×120. MGD of MHH and histogram of MHI have the size of $(160 + 120) \times 5 = 1400$ and 255, respectively.

5.3. Combining MHH and MHI histogram features

If different features are able to capture different discriminating properties of a video clip, then a combination of such features is likely to improve classification performance over either feature alone. Indeed, this is what we find in the following section, which describes our evaluation. Based on the simplicity requirement of our embedded systems, our two feature vectors are combined in the simplest way by concatenating these two feature vectors into a higher dimensional vector. Fig. 5 shows an example of a combination between the MGD of the MHH and the histogram of the MHI from the same video.

6. Evaluation of our system

6.1. Experimental setup

For the evaluation of our system, we use a challenging human action recognition database, recorded by Schuldt et al. (2004), which is both large and publically available. It contains six types

of human actions (walking, jogging, running, boxing, hand-waving and hand-clapping) performed several times by 25 subjects in four different scenarios: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3) and indoors (s4).

This database contains 2391 sequences. All sequences were taken over homogeneous backgrounds with a static camera with 25 Hz frame rate. The sequences were down-sampled to the spatial resolution of 160×120 pixels and have a time length of 4 s in average. To the best of our knowledge, this is the largest video database with sequences of human actions taken over different scenarios. All sequences were divided with respect to the subjects into a training set (8 persons), a validation set (8 persons) and a test set (9 persons).

In our experiment, the classifiers were trained on a training set while classification results were obtained on the test set. In all our experiments, the same parameters were used. The threshold in frame differencing was chosen as 25 and τ was chosen as 255 for MHI construction. The most suitable choice of the number of patterns M for MHH computation depends on the video clips and is a tradeoff between the compactness of the representation and the expressiveness of the representation. Building a frequency histogram of the patterns extracted from the training clips indicates that no more than half of the training data had the sixth pattern. Thus the number of patterns was chosen to be $M = 5$.

The size of the MHI is $160 \times 120 = 19200$, which is the same width as that of the frames in the videos. In our experiment, the SVM is implemented using the SVM^{light} software (Joachims, 1999). In SVM training, choosing a good parameter C value is not so straightforward and can significantly affect classification accuracy (Hastie et al., 2004) but in order to keep our system simple, the default value of C in SVM^{light} is used in all the experiments.

Fig. 6 shows examples in each type of human action in this dataset. In order to compare our results with those in (Ke et al., 2005; Schuldt et al., 2004), we use the exact same training set and testing set in our experiments. The only difference is that we did not use the validation dataset in training. Our experiments are carried out on all four different scenarios. In the same manner as Ke et al. (2005), each sequence is treated individually during the training and classification process. In all the following experiments, the parameters were are the same.

6.2. Performance on single features

We have tested the performance of the fundamental motion features MHI, MMHI and MGO in our system. Fig. 7 shows these three motion features extracted from the action examples shown in Fig. 6. In order to keep our system simple for hardware implementation, we use the simplest method to transform the motion features (MHI, MMHI and MGO) into a plain vector based on the pixel scan order (row by row) to feed the SVM classifier.

Firstly, we tested the system performance on the four different subsets of the whole dataset. The results can be seen in Fig. 8. The correctly classified percentage on these data subsets indicates how many percent of the action clips in the testing set were correctly recognized by the system. It is clear that the MHI feature gave the best classification performance in all the four subset while the MGO feature gave poor results for all four data subsets. We also can see that subset s2 (outdoors with scale variation) is the most difficult subset in the whole dataset.

From the experiments, it can be seen that this type of system can get reasonable results. The MHI based system looks better than the MMHI system in the experiments. The disadvantage for MMHI is that it can only work well in the case of an uncluttered and static background. If there is background motion or noise, this will be recorded in the feature vector and will reduce the performance of the classification.

For the whole dataset, the classification confusion matrix is a good measure for the overall performance in this multi-class classification problem. Table 1 shows the classification confusion matrix based on the method proposed in Ke et al. (2005). Table 2 shows the confusion matrix obtained by our system based on MHI. The confusion matrices show the motion label (vertical) versus the classification results (horizontal). Each cell (i, j) in the table shows the percentage of class i action being recognized as class j. Thus the main diagonal of the matrices show the percentage of correctly recognized actions, while the remaining cells show the percentages of misclassification. The trace of the matrix shows the overall classification rate. In Table 1, the trace is 377.8 and since there are six classes, then the overall mean classification rate is $377.8/6 = 63\%$.

In comparison with Ke's method, we use a simple MHI feature rather than large volumetric features in which the dimension of a feature vector might be a billion, yet the performance of our

Fig. 6. Six types of human action in the database: (a) walking (b) jogging (c) running (d) boxing (e) hand-clapping (f) hand-waving.

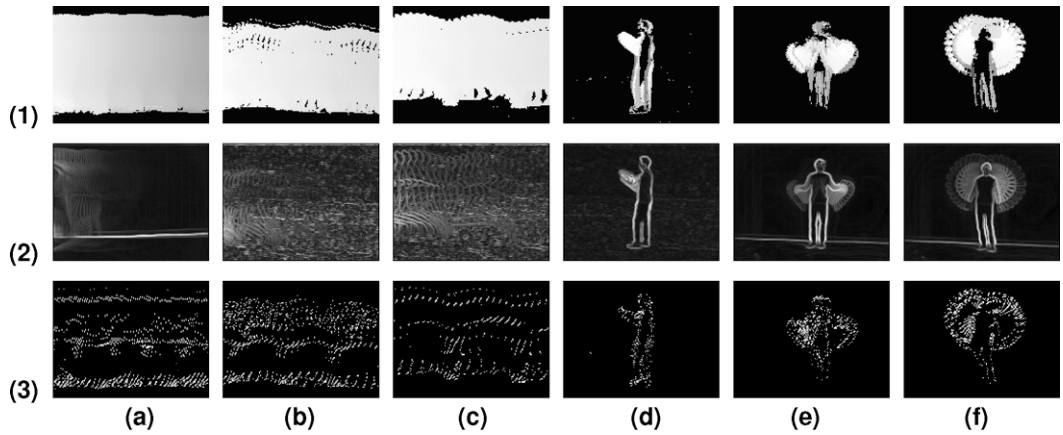

Fig. 7. The (1) MHI, (2) MMHI and (3) MGO for the six actions in the dataset: (a) walking (b) jogging (c) running (d) boxing (e) hand-clapping (f) hand-waving.

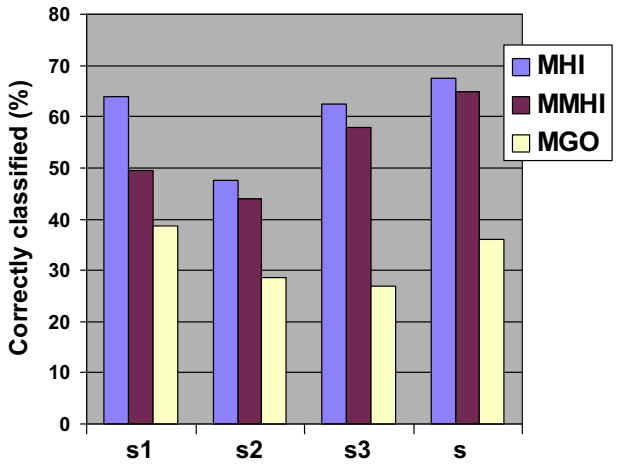

Fig. 8. Correctly classified percentage for separate data subset: s1 (outdoors), s2 (outdoors with scale variation), s3 (outdoors with different clothes) and s4 (indoors).

Table 1
Ke's confusion matrix (Ke et al., 2005), trace = 377.8, mean performance = 63%.

	Walk	Jog	Run	Box	Clap	Wave
Walk	**80.6**	11.1	8.3	0.0	0.0	0.0
Jog	30.6	**36.2**	33.3	0.0	0.0	0.0
Run	2.8	25.0	**44.4**	0.0	27.8	0.0
Box	0.0	2.8	11.1	**69.4**	11.1	5.6
Clap	0.0	0.0	5.6	36.1	**55.6**	2.8
Wave	0.0	5.6	0.0	2.8	0.0	**91.7**

Table 2
MHI's confusion matrix, trace = 381.2, mean performance = 63.5%.

	Walk	Jog	Run	Box	Clap	Wave
Walk	**53.5**	27.1	16.7	0.0	0.0	2.8
Jog	46.5	**34.7**	16.7	0.7	0.0	1.4
Run	34.7	28.5	**36.1**	0.0	0.0	0.7
Box	0.0	0.0	0.0	**88.8**	2.8	8.4
Clap	0.0	0.0	0.0	7.6	**87.5**	4.9
Wave	0.0	0.0	0.0	8.3	11.1	**80.6**

system is marginally better on this dataset. Notice from Table 2 that the actions of walking, jogging and running are not well discriminated, as they are similar actions, performed at different speeds, whereas the more distinctive actions of boxing, hand-waving and hand-clapping appear to be easier to classify.

Table 3
MHI_S's confusion matrix, trace = 377.7, mean performance = 62.95%.

	Walk	Jog	Run	Box	Clap	Wave
Walk	**56.9**	18.1	22.2	0.0	0.0	2.8
Jog	45.1	**29.9**	22.9	1.4	0.0	0.7
Run	34.7	27.8	**36.1**	0.0	0.0	1.4
Box	0.0	0.0	0.0	**89.5**	2.1	8.4
Clap	0.0	0.0	0.0	5.6	**88.9**	5.6
Wave	0.0	0.0	0.0	12.5	11.1	**76.4**

Table 4
Hist. of MHI's confusion matrix, trace = 328.6, mean performance = 54.8%.

	Walk	Jog	Run	Box	Clap	Wave
Walk	**62.5**	32.6	0.0	1.4	1.4	2.1
Jog	12.5	**58.3**	25.0	0.0	0.0	4.2
Run	0.7	18.8	**77.1**	0.0	0.0	3.5
Box	4.9	2.8	0.7	**17.5**	61.5	12.6
Clap	4.9	2.1	0.7	11.1	**75.0**	6.3
Wave	5.6	3.5	6.9	20.1	25.7	**38.2**

In a second series of experiments, we tested low dimensional features, which are generated from fundamental motion features, such as MHI. Sub-sampling is easy to implement in hardware by any factor of 2 and this can be done in both rows and columns of the motion feature. Tables 3 and 4 show the results based on down-sampling the MHI by a factor of 64 (a factor of 8 for both row and column) and the histogram of (sub-sampled) MHI, respectively. It can be seen that sub-sampling the MHI to a significantly lower resolution does not have an unduly detrimental effect on classification performance. This feature performed well in distinguishing the last three groups (box, clap, wave). On the other hand, the histogram of MHI did not perform well in terms of overall performance but has the power to distinguish the first three groups of action (walk, jog, run), which are similar shaped actions, executed at significantly different speeds. This demonstrates that these two features make different types of information more explicit to the classification process and this provides a strong motivation for combining different feature types.

Fig. 9 shows examples in each type of human action and their associated MHH motion features. For the MHH, it is hard to deal with the whole feature in our target hardware system as, with the number of patterns set to 5, the MHH has a relatively high dimension of $5 \times 160 \times 120 = 96000$. Thus, we constructed a

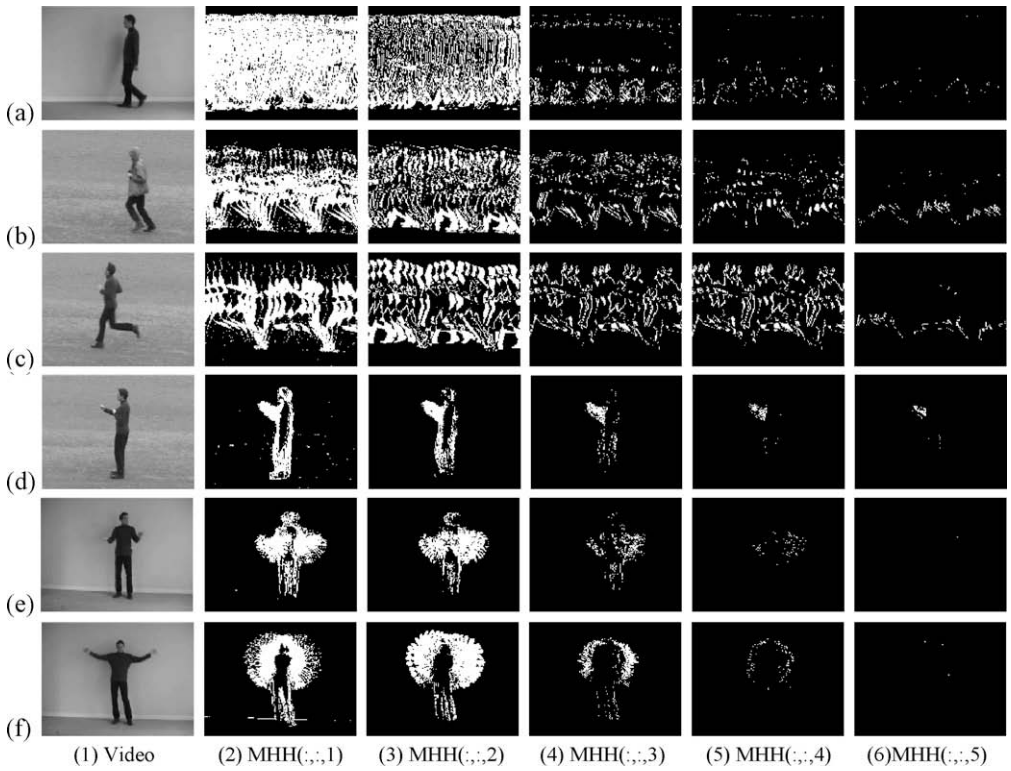

Fig. 9. The six database human actions and associated MHH features: (a) walking (b) jogging (c) running (d) boxing (e) hand-clapping (f) hand-waving.

smaller sized MHH_S by averaging the pixels in an 8×8 block, so that the size of all MHH feature vectors is reduced to $20 \times 15 \times 5 = 1500$. Our MGD feature also has a small size of $(160 + 120) \times 5 = 1400$.

Tables 5 and 6 show the results when using features MHH_S and MGD respectively. From these two tables, it is very clear that both MHH_S and MGD improve the overall performance, but they are failed to classify the 'jogging' class. The reason is that these video clips are quite similar to walking and running. It is hard to distinguish them correctly even by human observation. This motivates us to consider the combination of the MGD with a feature that is particularly sensitive to the speed of the motion, as is the case with the 'histogram of MHI'.

Table 5
MHH_S's confusion matrix, trace = 417.3, mean performance = 69.55%.

	Walk	Jog	Run	Box	Clap	Wave
Walk	**88.9**	1.4	6.3	0.7	1.4	1.4
Jog	56.9	**2.1**	38.2	0.7	2.1	0.0
Run	22.2	0.7	**75.7**	0.0	1.4	0.0
Box	0.0	0.0	0.0	**96.5**	0.7	2.8
Clap	0.0	0.0	0.0	4.2	**93.1**	2.8
Wave	0.0	0.0	0.0	22.2	16.7	**61.1**

Table 6
MGD's confusion matrix, trace = 432.6, mean performance = 72.1%.

	Walk	Jog	Run	Box	Clap	Wave
Walk	**85.4**	4.9	2.8	2.8	2.8	1.4
Jog	65.3	**9.2**	23.6	2.1	0.0	0.0
Run	18.8	8.3	**68.8**	1.4	0.0	2.8
Box	0.0	0.0	0.0	**91.6**	2.8	5.6
Clap	1.4	0.0	0.0	6.3	**92.4**	0.0
Wave	0.0	0.0	0.0	7.6	6.9	**85.4**

6.3. Performance on combined features

From the previous subsection, we found that different features had different powers in distinguishing classes of action. In order to overcome their own limitations, we combine the MGD (derived from the MHH) and the histogram of the MHI, by concatenation. Table 7 shows the confusion matrix obtained from our system when these combined features were used. From this table, we can see that the overall performance has a significant improvement over Ke's method, which is based on volumetric features. Note that a good performance, averaging over 80% in correct classifications, is achieved in distinguishing the six actions in this challenging dataset.

We compared our results with other methods on this challenging dataset and we summarize the correctly classified rates in Table 8. From this table, we can see that MHH has made a significant improvement in comparison with MHI. Furthermore, the MGD feature gives a better performance than the MHH itself. The best performance, which gives significantly better classification results, came from the combined feature, which is based on the histogram of the MHI and the MGD.

It should be mentioned here that some performance results presented in the literature (Dollár et al., 2005; Niebles et al., 2006; Yeo et al., 2006) are quoted as higher than ours, when

Table 7
MGD & Hist. of MHI's confusion matrix, trace = 481.9, mean performance = 80.3%.

	Walk	Jog	Run	Box	Clap	Wave
Walk	**66.0**	31.3	0.0	0.0	2.1	0.7
Jog	13.9	**62.5**	21.5	1.4	0.0	0.7
Run	2.1	16.7	**79.9**	0.0	0.0	1.4
Box	0.0	0.0	0.0	**88.8**	2.8	8.4
Clap	0.0	0.0	0.0	3.5	**93.1**	3.5
Wave	0.0	0.0	0.0	1.4	6.9	**91.7**

Table 8
Overall correctly classified rate (%) for all the methods on this open, challenging dataset.

Method	Rate (%)
SVM on local features (Schuldt et al., 2004)	71.7
Cascade of filters on volumetric features (Ke et al., 2005)	63
SVM on MHI (Meng et al., 2006b)	63.5
SVM_2K on MHI & MMHI (Meng et al., 2006a)	65.3
SVM on MHH_S	69.6
SVM on MGD	72.1
SVM on HWT of MHI & Hist. of MHI (Meng et al., 2007b)	70.9
SVM on MGD & Hist. of MHI	**80.3**

performing classification experiments on (parts of) this public dataset. However, we have not included these in Table 8, because they are not directly comparable to ours, either because they have omitted to use all of the data, or because they have performed an easier classification task, or both. For example, Dollár et al. (2005) achieved a correct classification rate of 81.2%, but the evaluation omitted scenarios s2 and s4. Our earlier results, shown in Fig. 8, clearly indicate that scenario s2 is the most difficult subset. This is also witnessed in the results of our best system (MGD and histogram of MHI), which give a mean performance of 80.4%, 63.5%, 82.4% and 87% for the four scenarios s1, s2, s3 and s4, respectively. When we combine data from the s1 and s3 subsets, as used by Dollár et al. (2005), and re-train our SVM classifiers on this combined data set, then we obtain a mean performance of 82.8%, which is slightly better than the result of Dollár et al. (2005) (81.2%). Niebles et al. (2006) obtained similar results with 81.5% and Yeo et al. (2006) obtained 86.0%, but they did an easier task of classifying each complete sequence (containing four repetitions of same action) into one of six classes, while our method was trained the same way as other papers (Ke et al., 2005; Meng et al., 2006a,b, 2007b), that is to detect a single instance of each action, within arbitrary sequences in the dataset. Furthermore, Yeo et al. (2006) did not use the difficult subset 2 of the dataset, as was the case in (Dollár et al., 2005).

7. Conclusion and discussion

In this paper, we have proposed new, descriptive representations for human action recognition systems, which may easily be implemented in an embedded computer vision context. The proposed method does not rely on accurate tracking as many other works do, since most tracking algorithms incur an extra computational cost for the system. Our system is based on simple 'temporal template' features in order to achieve high-speed recognition in real-time embedded applications.

In order to improve the state-of-the-art performance in temporal template based systems, we have proposed a new representation for motion information in video and this is called the Motion History Histogram (MHH). The representation extends previous work on temporal template (MHI related) representations by additionally storing frequency information as the number of times motion is detected at every pixel, further categorized into the length of each motion. In essence, maintaining the number of contiguous motion frames removes a significant limitation of MHI, which only encodes the time from the last observed motion at every pixel. It can be used either independently, or combined with information derived from the MHI, to give human action recognition systems with improved performance over existing comparable temporal template based systems.

In terms of our work on combining features from MHI and MHH, we acknowledge that we have not generated a provably optimal feature combination. Rather, we have selected two of the most promising feature types that we have tested, that make explicit different aspects of the motion sequence, and we have shown that the combined performance is better than either feature performance alone. This suggests that further fruitful research would be to search for optimal combinations of temporal template type features, for example using a genetic algorithm, which is well suited to such a task.

In comparison with local SVM methods of Schuldt et al. (2004) and a cascade of filters on volumetric features of Ke et al. (2005), our feature vectors are computationally inexpensive. Even though we do not use a validation dataset for parameter tuning in SVM training, we have demonstrated a significant improvement in recognition performance over other methods tested in a comparable rigorous way on the large, public human action database (Schuldt et al., 2004).

References

Aggarwal, J.K., Cai, Q., 1999. Human motion analysis: A review. Computer Image and Vision Understanding 73 (3), 428–440.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R., 2005. Actions as space–time shapes. In: ICCV, pp. 1395–1402.

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

Bradski, G.R., Davis, J.W., 2002. Motion segmentation and pose recognition with motion history gradients. Machine Vision Appl. 13 (3), 174–184.

Campbell, L.W., Bobick, A.F., 1995. Recognition of human body motion using phase space constraints. In: ICCV, pp. 624–630.

Cristianini, N., Shawe-Taylor, J., 2000. An Introduction to Support Vector Machines (and other kernel-based learning methods). Cambridge University Press, Cambridge, UK.

Dalal, N., Triggs, B., Schmid, C., 2006. Human detection using oriented histograms of flow and appearance. In: ECCV (2), pp. 428–441.

Davis, J.W., 2001. Hierarchical motion history images for recognizing human motion. In: IEEE Workshop on Detection and Recognition of Events in Video, pp. 39–46.

Davis, J.W., Tyagi, A., 2006. Minimal-latency human action recognition using reliable-inference. Image Vision Comput. 24 (5), 455–472.

Dollár, P., Rabaud, V., Cottrell, G., Belongie, S., 2005. Behavior recognition via sparse spatio-temporal features. In: VS-PETS.

Efros, A.A., Berg, A.C., Mori, G., Malik, J., 2003. Recognizing action at a distance. In: ICCV, pp. 726–733.

Farmer, J., Casdagli, M., Eubank, S., Gibson, J., 1991. State-space reconstruction in the presence of noise. Physica D 51, 52–98.

Gao, J., Hauptmann, A.G., Bharucha, A., Wactlar, H.D., 2004. Dining activity analysis using a hidden markov model. In: ICPR (2), pp. 915–918.

Green, R.D., Guan, L., 2004. Quantifying and recognizing human movement patterns from monocular video images – Part II: Applications to biometrics. IEEE Trans. Circuits System Video Technol. 14 (2), 191–198.

Hastie, T., Rosset, S., Tibshirani, R., Zhu, J., 2004. The entire regularization path for the support vector machine. <citeseer.ist.psu.edu/hastie04entire.html>.

Joachims, T., 1999. Making large-scale SVM learning practical. In: Oikonomopoulos, Antonios, Patras, Ioannis, Pantic, Maja (Eds.), Advances in Kernel Methods – Support Vector Learning. MIT-Press, USA. <http://svmlight.joachims.org/>.

Ke, Y., Sukthankar, R., Hebert., M., 2005. Efficient visual event detection using volumetric features. In: ICCV, Beijing, China, October 15–21, 2005, pp. 166–173.

Meng, H., Pears, N., Bailey, C., 2006a. Human action classification using SVM_2K classifier on motion features. In: LNCS, vol. 4105. Istanbul, Turkey, pp. 458–465.

Meng, H., Pears, N., Bailey, C., 2006b. Recognizing human actions based on motion information and SVM. In: 2nd IET Internat. Conf. on Intelligent Environments. IET, Athens, Greece, pp. 239–245.

Meng, H., Pears, N., Bailey, C., 2007a. A human action recognition system for embedded computer vision application. In: CVPR Workshop on Embedded Computer Vision.

Meng, H., Pears, N., Bailey, C., 2007b. Motion information combination for fast human action recognition. In: 2nd Internat. Conf. on Computer Vision Theory and Applications (VISAPP07), Barcelona, Spain.

Meng, H., Freeman, M., Pears, N., Bailey, C., 2008. Real-time human action recognition on an embedded, reconfigurable video processing architecture. J. Real-Time Image Process. 3 (3), 163–176.

Moeslund, T., Hilton, A., Kruger, V., 2006. A survey of advances in vision-based human motion capture and analysis. Computer Image and Vision Understanding 103 (2–3), 90–126.

Nascimento, J.C., Figueiredo, M.A.T., Marques, J.S., 2005. Recognition of human activities using space dependent switched dynamical systems. In: IEEE Internat. Conf. on Image Processing, ICIP.

Niebles, J., Wang, H., Fei-Fei, L., 2006. Unsupervised learning of human action categories using spatial–temporal words. In: BMVC06, p. III:1249.

Ogata, T., Tan, J.K., Ishikawa, S., 2006. High-speed human motion recognition based on a motion history image and an eigenspace. IEICE Trans. Inform. Systems E89 (1), 281–289.

Oikonomopoulos, A., Patras, I., Pantic, M., 2006. Kernel-based recognition of human actions using spatiotemporal salient points. In: Proc. of CVPR Workshop 06, vol. 3, pp. 151–156. <http://pubs.doc.ic.ac.uk/Pantic-CVPR06-1/>.

Schuldt, C., Laptev, I., Caputo, B., 2004. Recognizing human actions: A local SVM approach. In: ICPR, Cambridge, UK.

Stauffer, C., Grimson, W.E.L., 2000. Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Machine Intell. 22 (8), 747–757 <http://citeseer.ist.psu.edu/stauffer00learning.html>.

Weinland, D., Ronfard, R., Boyer, E., 2005. Motion history volumes for free viewpoint action recognition. In: IEEE Internat. Workshop on modeling People and Human Interaction (PHI'05). <http://perception.inrialpes.fr/Publications/2005/WRB05>.

Wong, S.-F., Cipolla, R., 2005. Real-time adaptive hand motion recognition using a sparse bayesian classifier. In: ICCV-HCI, pp. 170–179.

Wong, S.-F., Cipolla, R., 2006. Continuous gesture recognition using a sparse bayesian classifier. In: ICPR (1), pp. 1084–1087.

Yeo, C., Ahammad, P., Ramchandran, K., Sastry, S., 2006. Compressed domain real-time action recognition. In: IEEE Internat. Workshop on Multimedia Signal Processing (MMSP) – 2006. IEEE, Washington, DC, USA.