Fast Human Activity Recognition based on Structure and Motion

Jinhui Hu and Nikolaos V. Boulgouris*

Department of Electronic Engineering, King's College, London, Strand, London WC2R 2LS, UK

Department of Electronic and Computer Engineering, Brunel University, Uxbridge, Middlesex UB8 3PH, UK

Abstract

In this paper, we present a method for the recognition of human activities. The proposed approach is based on the construction of a set of templates for each activity as well as on the measurement of the motion in each activity. Templates are designed so that they capture the structural and motion information that is most discriminative among activities. The direct motion measurements capture the amount of translational motion in each activity. The two features are fused at the recognition stage. Recognition is achieved in two steps by calculating the similarity between the templates and the motion features of the test and reference activities. The proposed methodology yields excellent results when applied on the INRIA database.

Keywords: Activity, recognition, surveillance

[*] Corresponding author. Tel.: +44 1895 267629; fax: +44 1895 269782. E-mail address: nikolaos.boulgouris@brunel.ac.uk (N.V. Boulgouris).

1. Introduction

Although the earliest research in studying human movement was published in the 1850s [1], the automatic recognition of human activities [2], [3], [4], has emerged only recently as an important research area. The current research trend largely originated from a strong contemporary need for the development of applications, such as, automatic monitoring, surveillance, and intelligent human-computer interfaces. Human activity recognition is a very challenging task due to the great variability with which different people may perform the same activity.

Various approaches on activity representation and recognition have been presented during the past few years. One of the most important activity recognition techniques appeared in [5]. In that work, a motion template was introduced in order to describe a set of activities. Specifically, a binary motion-energy image (MEI) and a motion-history image (MHI) were introduced, which, when taken together, can be used as a two component version of a temporal template. Since its introduction, this approach has been widely used for the interpretation of human movement in image sequences.

The above approach was further improved in [6] in which temporal templates were extended to 3D in order to achieve viewpoint independence. The 2D silhouettes were extended to three dimensions (3D) using a visual hull [7]. Motion History Volumes (MHV) were introduced to represent human actions, which allow different camera configurations.

A popular group of approaches applied to human activity recognition use Hidden Markov Models (HMMs) [8], [9], [10], [11]. In [9], motion and shape features were represented using optical flow and eigen-shape vectors, and HMMs were applied for recognition. An object trajectory-based activity recognition method using HMMs was introduced in [10], whereas in [11], several feature extraction algorithms based on PCA, ICA, and LDA, were applied and then followed by HMM modeling for recognition.

In [12], a method was proposed for human activity recognition based on an average template with a multiple-feature vector. The features that were used include the width feature as well as spatio-temporal features. Using the extracted features, Dynamic Time Warping (DTW) was used in combination with the average template to perform recognition.

In [13], activities were modeled based on their underlying dynamics and described as a cascade of dynamical systems. Further, methods were derived for the incorporation of view- and rate-invariance into the proposed models in order to enable similar activities to be directly clustered together regardless of view point or execution speed.

In [14], an example-based activity recognition was introduced by using an activity representation scheme according to which each activity was modeled as a series of synthetic poses. Recognition was achieved by matching the input silhouettes with the key poses using an enhanced Pyramid Match Kernel algorithm.

In [15], each activity was represented by descriptors using Temporal Laplacian Eigenmaps. Subsequently, all view-dependent manifolds were automatically combined in order to find a representation in the 3D space that is independent from style and viewpoint. Dynamic time warping was applied for recognition.

In [16], an activity representation method was proposed which describes the video sequence using a set of spatiotemporal features called video-words. This was obtained by quantizing extracted 3D interest points. Then, the optimal number of video-words clusters (VWCs) was determined by grouping the redundant video-words. Classification was achieved by using a correlogram.

The method we propose in this paper uses both shape-based and motion-based features, as the combination of these two types of features can improve the efficiency of the recognition process. Our approach is based on activity templates, which capture the information in the body postures assumed during each activity, as well as of the observed motion within each activity. After activity templates are constructed and the motion is calculated, recognition is achieved by means of comparison with the corresponding features that are stored in a database of reference activities.

Recognition takes place in two stages. Initially, a number of best matches to the given test activity are calculated and, subsequently, the original selection is refined by using a selection process that is tailored to discriminating among the best matches of the first recognition stage. Experimental results show that this approach is clearly more efficient than the direct recognition of a test activity among a diverse set of activities.

In summary, the contributions of the present paper are:

- A novel method for template construction based on centered silhouettes. We found that this construction is preferable to the conventional construction based on un-processed silhouettes.

- The representation of activities in terms of a spatiotemporal profile and a motion profile.

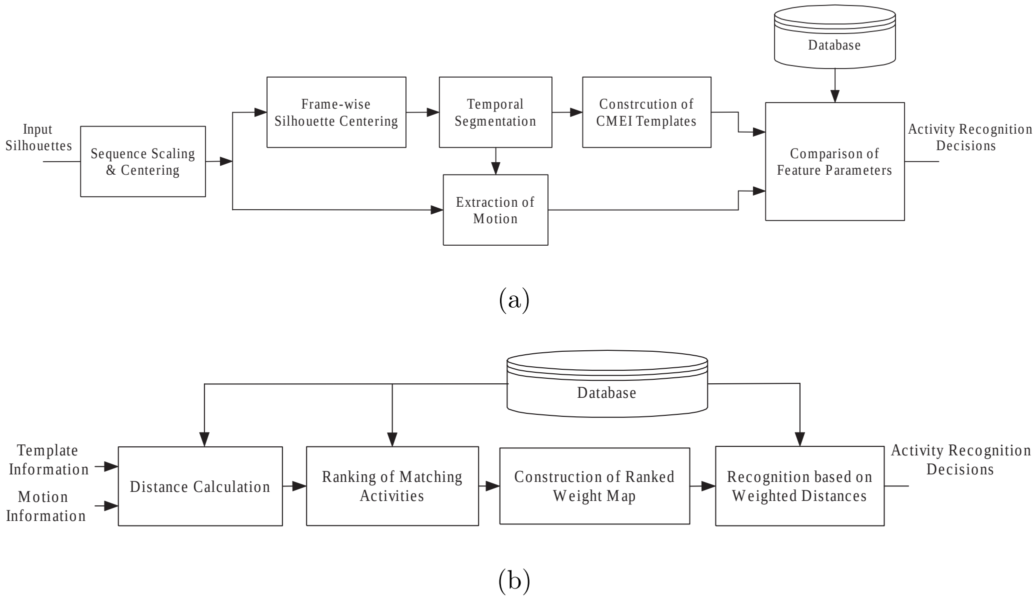
Database
Input Silhouettes
Sequence Scaling & Centering
Frame-wise Silhouette Centering
Temporal Segmentation
Constrcution of CMEI Templates
Extraction of Motion
Comparison of Feature Parameters
Activity Recognition Decisions
(a)
Database
Template Information
Motion Information
Distance Calculation
Ranking of Matching Activities
Construction of Ranked Weight Map
Recognition based on Weighted Distances
Activity Recognition Decisions
(b)

Figure 1: (a) General block diagram, (b) Detailed block diagram of the recognition process based on the motion and template information.

- A two-stage method for activity recognition based on discriminative weighting that is tailored to the bast matching activities of a given test activity.

The structure of the paper is as follows: in Section 2, the proposed feature extraction methodology is described. In Section 3, two-phase activity recognition using discriminative weighting is presented. The proposed method is experimentally assessed for activity recognition in Section 4 and, finally, conclusions are drawn in Section 5.

2. Feature Extraction For Recognition

2.1. Overview

The proposed activity recognition system is outlined in Fig 1(a). The system operates under the assumption that the input to the system is sequences of binary silhouettes that depict the side-view of the person conducting the activity. In practice, however, there are cases in which the input sequences may not depict the side-view of the person. In the experimental results section, we investigate how this possible deviation from the assumed conditions affects the recognition performance of our system. Another assumption we are making is that activity segmentation from online video streams is performed using one of the existing approaches that are available in the literature. Therefore, in this work we do not propose a new method for separating between consecutive activities in online video streams. Such an approach was presented in [17] in which temporal segmentation is based on the definition of motion boundaries, which is achieved through the computation of global motion energy.

After an initial scaling and centering stage, each activity sequence is temporally segmented into a number of parts, which define the stages in which the activity is performed. Considering the process of evolution of each activity, we came to the conclusion that four stages suit the recognition best. The first and the last stages normally are the starting and ending poses and on many occasions (i.e., when the starting and ending pose is "standing") they do not carry much discriminative information. The middle stages reflect the evolution of the activity. Having three stages in total, i.e., one middle stage only, would be insufficient. This means that at least four stages are needed for discriminative representation and feature extraction. On the other hand, the maximum number of stages could potentially be five, as an even greater number of segments (e.g., six) could not capture further distinct poses in an activity. Therefore, the choice in our case was between having four and five stages. We found that using four stages is preferable both in terms of computational efficiency and performance, although the performance difference between using four and five stages is marginal.

Based on this temporal segmentation, motion and shape-based features are extracted from the input silhouette sequence. Specifically, for each of the four parts in a sequence, a template is constructed and a motion vector is calculated in order to quantitatively detect and represent translational motion. The four motion vectors are subsequently combined with the activity templates at the decision stage in order to achieve efficient recognition. Decisions are made by calculating the distance between the features extracted from a test activity and the features extracted from activities in the reference database. This process is outlined in Fig 1(b).

2.2. Preprocessing

In general, in a video sequence showing the performance of a given activity, the person performing the activity may be standing in an arbitrary position and have an arbitrary body pose. For this reason, prior to the calculation of the template, we scale and center the silhouettes. The scaling factor is obtained by calculating the ratio of the size of the foreground object in a standard frame over the object's size in the first frame of each of the database sequences. This means that for each activity sequence there is a specific scale factor according to which all frames in this sequence are scaled.

Symbol	Notation
i	Frame index
(x, y)	Pixel co-ordinates
F	Total number of frames
s	Activity stage index
a	Activity index
N	Total number of activities
$\mathbf{T}_a$	Spatiotemporal profile for activity a
$\mathbf{t}_{as}$	sth stage template for activity a
$\mathbf{M}_a$	Motion profile for activity a
$\mathbf{m}_{as}$	sth stage motion profile for activity a
$\mathbf{R}_k$	kth ranked spatiotemporal profile
$\mathbf{r}_{ks}$	sth stage template for ranked activity
$\mathbf{w}_s$	Weight map for stage s

Table 1: Notation

Centering of the foreground object, *i.e.*, of the person conducting the activity, is applied after all silhouettes are scaled. Two kinds of centering methods were tested: in the first method, horizontal displacements were cancelled so that the foreground object is placed in the middle of the frame. The same displacement vector was used for all frames in a sequence. In the second method, silhouettes were centered on a frame by frame basis. The averaged frames corresponding to these two different approaches are shown in Fig 2. As seen, unlike the sequence-wise centering, the frame-wise centering affects the vertical displacements during the activity.

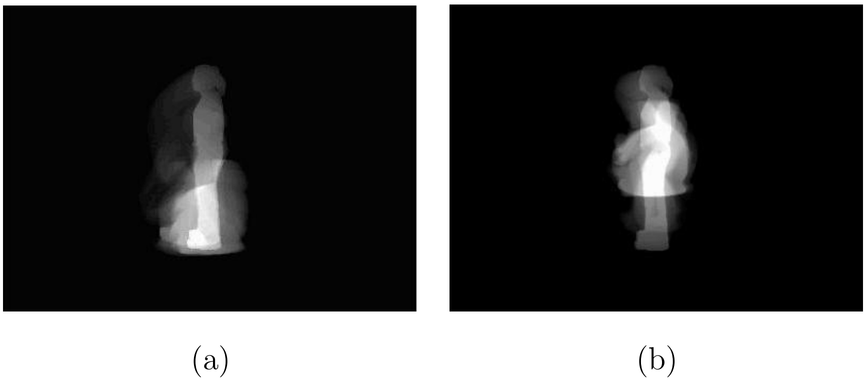

Figure 2: Different centering approaches for the calculation of average images (sitting activity). (a) Sequence-wise centering, (b) Frame-wise centering.

2.3. Temporal partitioning of activities

An activity can be performed in dissimilar ways by different persons, or even by the same person. One common difference is the speed with which activities are executed. In practice, the speed with which a person is conducting an activity may vary even during the execution of the activity itself. The great temporal variability in the way activities are performed necessitates the deployment of methods that are robust to such variations. For this reason, we partition each activity into activity stages and construct representative pose templates for each such stage. To this end, we use a simple clustering algorithm in order to effectively extract representative pose information. The steps of the clustering process are summarized below:

1. Initially, an activity sequence with F frames is divided into four continuous temporal segments; each temporal segment has roughly $F/4$ frames. Therefore, the initial temporal segment boundaries are: $f_1 = F/4, f_2 = F/2, f_3 = 3F/4, f_4 = F$.

2. An average frame $A_s, s = 1, \ldots, 4$, is calculated from each temporal segment.
3. The sequence is partitioned into new temporal segments. Specifically, new boundaries f'_s, $s = 1, 2, 3, 4$, are calculated between segments s and $s+1, s = 1, 2, 3$, based on:

$$f'_s = \arg\min_f \left[D_s(f) + D_{s+1}(f)\right] \tag{1}$$

where $D_s(f)$ and $D_{s+1}(f)$ are the Euclidean distances between the frames within each of the temporal segments and the segments corresponding average frame:

$$D_s(f) = \frac{1}{f - f_s + N} \sum_{i=f_s-N}^{f} D(I_i, A_s) \tag{2}$$

$$D_{s+1}(f) = \frac{1}{f_s + N - f + 1} \sum_{i=f}^{f_s+N} D(I_i, A_{s+1}) \tag{3}$$

4. Step 2 is repeated until convergence or until a maximum number of iterations is reached.

Using the above simple technique, a given activity is divided into four segments that correspond to four stages of the activity. A template can be constructed for each stage. This construction is described next.

2.4. Template Construction

We use two main features in our activity recognition algorithm. The first is a spatiotemporal template that is mainly aimed to capture pose information in human activities. The second feature is aimed to represent the motion that is involved in the activity.

Motion Energy Images (MEI) and Motion History Images (MHI) were proposed in [5] in order to encode, respectively, the location and the type of motion. We propose the use of a similar temporal template in our system. The similarity consists in the representation of the activity by means of four MEI-like templates. In our case, however, the construction of the MEI is based on a *centered* sequence of silhouettes. This approach makes the impact of motion even more apparent on the resulting template, which we will call *Centered MEI* (CMEI). Given an image sequence comprising frames $I_j, j = 1, 2, \ldots, F$, the binary CMEI function is defined [5] as:

$$E_{\tau i} = \bigcup_{j=0}^{\tau-1} B_{t-j}(x, y) \tag{4}$$

where τ is the duration of a movement. In our case, the value of τ is set to be the total number of frames in each stage of an activity execution. The term B_j indicates the regions of motion according to the I_j and is calculated using image-differencing:

$$B_j = C(I_{j+1}) - C(I_j) \tag{5}$$

where $C(\cdot)$ denotes the centering operation.

Based on the above calculation, the template, corresponding to the ath activity, will comprise of four *stage templates* $\mathbf{t}_{as}, s = 1, \ldots, 4$. This representation can be compactly expressed as:

$$\mathbf{T}_a = \{\mathbf{t}_{a1}, \mathbf{t}_{a2}, \mathbf{t}_{a3}, \mathbf{t}_{a4}\} \tag{6}$$

and, henceforth, it shall be referred to as *spatiotemporal profile.*

In Fig 3, the four stage templates are shown for each one of the twelve activities in the INRIA database. It can be seen that the resultant templates represent the information that *changes* throughout each activity, *i.e.*, the information that carries the most discrimination power. Due to their distinct characteristics, the four templates offer a compact activity representation of high discriminating capacity.

The above set of templates, based on the Motion Energy Image of an activity sequence, will be subsequently used for activity recognition purposes. As will be seen, despite its simplicity, this approach yields very good activity recognition performance.

2.5. Extraction of Motion Information

In our system, we take into consideration the amount of motion that takes place during the performance of an activity. As a measure of motion, in this case, we use the movement of the foreground object's center position. Unlike the template-based approach that was described previously, the method we propose for the extraction of motion is calculated based on the original sequence, without prior centering of the silhouettes, since any centering or scaling would affect the measured motion. This process is graphically illustrated in Fig 1(a).

In order to calculate the amount and the direction of motion, we consider the sequence of silhouette center coordinates (x_{ai}, y_{ai}), $i = 1, 2, \ldots, F$, for the ath activity, $a = 1, 2, \ldots, N$. Initially, the average center coordinate $(\bar{x}_a, \bar{y}_a)$ is calculated from this sequence. Therefore, for the ath activity, a sequence of difference vectors is initially formed:

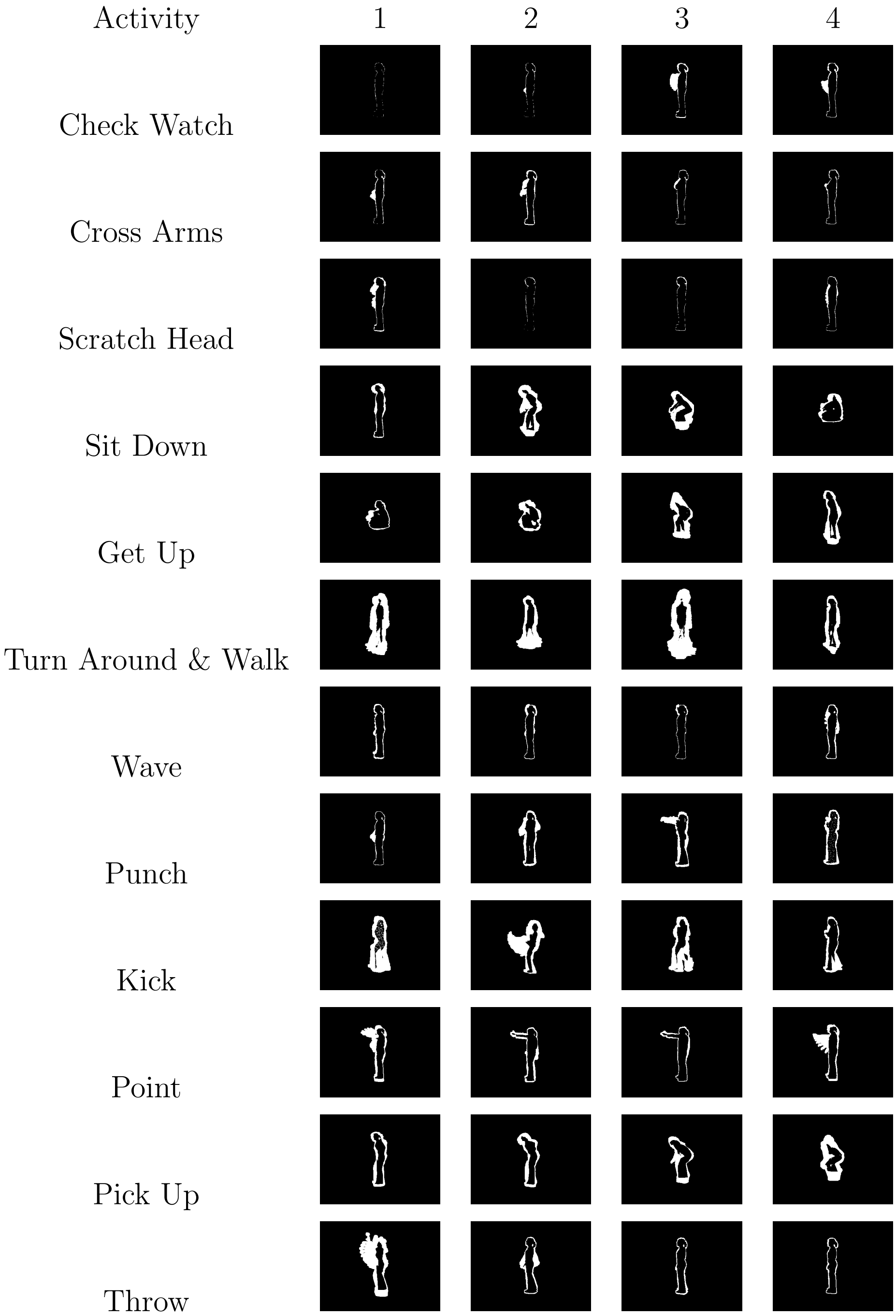

Figure 3: CMEI templates for each of the activities in the INRIA database.

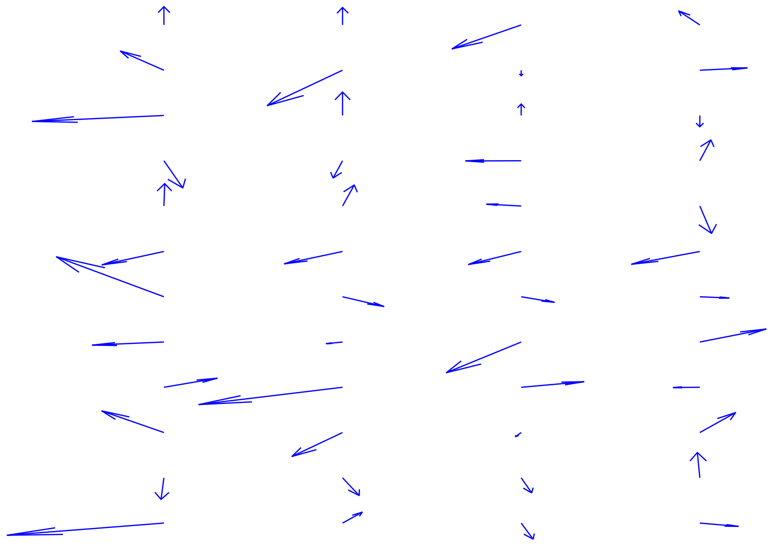

Figure 4: Graphical representation of motion profiles for each of the activities in the INRIA database. Each row of vectors represent a motion profile. The motion profile for the first activity is on the top row.

$$\mathbf{Z}_a(i) = \begin{bmatrix} x_{ai} - \bar{x}_a \\ y_{ai} - \bar{y}_a \end{bmatrix} \tag{7}$$

In the sequel, the motion for the ath activity is measured separately for the four stages in each activity:

$$\mathbf{m}_{as} \triangleq \frac{1}{F_{as}} \sum_{i \in S_a} \mathbf{Z}_{as}(i), \qquad s = 1, \ldots, 4 \tag{8}$$

where F_{as} is the number of frames in activity a and S_a is the set of frame indices in stage s. As seen, the above motion measurement essentially represents the translational motion of the center of the silhouettes with respect to the average center of the foreground object for each stage of a particular activity. Actually, $\mathbf{m}_{as}$ corresponds to the silhouette center motion between the first and the last frame of each stage. The contribution of such a feature to a system's recognition efficiency may be small in cases where the person performing the activity is standing or in case the person is engaging in an activity with very limited motion. However, in cases where the person who is conducting the activity is moving, this feature has a very considerable contribution to recognition accuracy.

Based on the above, the motion information, corresponding to the ath activity, will comprise of the four stage motion vectors $\mathbf{m}_{as}, s = 1, 2, \ldots, 4$. This can be compactly written as:

$$\mathbf{M}_a = \{\mathbf{m}_{a1}, \mathbf{m}_{a2}, \mathbf{m}_{a3}, \mathbf{m}_{a4}\} \tag{9}$$

and, henceforth, will be referred to as *motion profile.*

The four motion vectors for each of the 12 activities in the INRIA database are shown in Fig 4. As seen, the motion profile of an activity includes a good amount of discrimination power and, by itself, it could be used as a means for recognition. Results using this type of information will be presented in the experimental evaluation section. The above motion information will be used in combination with the CMEI templates of the previous section in order to achieve accurate recognition of activities.

3. Two-phase Activity Recognition

3.1. Distance Calculation

Given a test sequence depicting an unknown activity, our objective is to recognize the activity that is being performed by comparison with a set of reference activities. Using our system, activity recognition is achieved by comparing the spatiotemporal and motion profiles of the unknown test activity to those of each of the reference activities. Recognition is achieved based on two types of extracted features, namely, the *CMEI templates in the spatiotemporal profiles* and the *activity motion profile.*

For the sake of description of our methodology, let us assume that a spatiotemporal profile $\mathbf{T}_g$ is constructed from an unknown test activity sequence. In order to recognize the index g of the unknown activity, distances are calculated between the profile obtained from the unknown test activity and the N activity profiles in a reference database. These distances, denoted T_D, are compactly expressed as:

$$T_D[a] = d(\mathbf{T}_g, \mathbf{T}_a) \triangleq \sum_{s=1}^{4} d(\mathbf{t}_{gs}, \mathbf{t}_{as}), \qquad a = 1, 2, \ldots, N \tag{10}$$

where $d(\cdot)$ denotes the Euclidean distance, and $\mathbf{T}_a$ is the profile constructed during the training session for the ath reference activity.

In a similar way, we can calculate the motion distance M_D between the motion profile $\mathbf{M}_g$, which was extracted from the test sequence, and the N reference motion profiles that correspond to the N activities in the reference database:

$$M_D[a] = d(\mathbf{M}_g, \mathbf{M}_a) \triangleq \sum_{s=1}^{4} d(\mathbf{m}_{gs}, \mathbf{m}_{as}), \qquad a = 1, 2, \ldots, N \tag{11}$$

Since it is reasonable to expect that T_D and M_D will have unequal contributions to recognition performance, the total dissimilarity between a test activity and the ath reference activity is defined as:

$$D[a] = T_D[a] + qM_D[a], \qquad a = 1, 2, \ldots, N \tag{12}$$

In the above definition, q is a parameter that is aimed to normalize the contribution of the two distances during the calculation of the total distance. The parameter q depends on the size of the foreground objects in the activity video sequences and it is automatically readjusted whenever a change is made in the scaling factor in the silhouette preprocessing stage. The value of q is practically calculated as the value that equalizes the mean values of structural distances and motion distances within the training set of activities.

In case there are several instances of each activity in the reference database, then the distance $D[a]$ in eq. (12) represents the distance between the test activity and the instance of the ath activity in the database *that yields the minimum distance.*

3.2. Discriminative Weighting

Considering that the issue of temporal variability of activities has been addressed by our system with the extraction of four characteristic spatiotemporal templates, the main remaining obstacle in recognizing an activity correctly is the existence of different activities that look similar in the reference database. The consequence of the above is that the variation between different activities may appear to be smaller than the variation between different instances of the same activity. Therefore, a given test activity may yield a fairly small distance even when compared with a different activity in the database.

One of the most popular ways to deal with problems like the above and maximize recognition efficiency is by means of subspace projection using Linear Discriminant Analysis (LDA) [18]. In such cases, the application of LDA requires the conversion of images into long vectors that are subsequently used for the calculation of eigenvectors and variance matrices. Since this calculation can be difficult, the method in [19] is normally used in order to make the problem computationally tractable. Unfortunately, the subspace that can be obtained using this method is of dimension equal to the number of classes. Since we only have a relatively small number of activities, the resultant analysis would be quite restricting and would not generally give good performance in the present scenario.

Another, much simpler, way to maximize recognition efficiency is by applying weighting that *highlights* the differences between activities during the calculation of the distances. In this way, the template distance $d(\mathbf{t}_{gs}, \mathbf{t}_{as})$ in eq. (10) can be replaced by a weighted distance defined as:

$$\tilde{d}(\mathbf{t}_{gs}, \mathbf{t}_{as}) \triangleq \sum_{x} \sum_{y} \tilde{\mathbf{w}}(x,y) |\mathbf{t}_{gs}(x,y) - \mathbf{t}_{as}(x,y)|, \qquad s = 1, \ldots, 4 \qquad (13)$$

where $\tilde{\mathbf{w}}(x,y)$ is the weighting coefficient at template position (x,y). The weighting coefficients should be greater in template areas that differ among different activities and smaller coefficients in areas of similarity. Consequently, if we attempt to design a weight map in order to optimally distinguish among different activities, the distribution of energy on the weight map will be primarily dependent on activities that are very dissimilar. On the contrary, similar activities will make smaller contributions to the weight map. Clearly, a weight map calculated as above will be inefficient for distinguishing between activities with small differences. Therefore, the problem of distinguishing between similar activities cannot be dealt with using the above straightforward weight map design.

In order to overcome this problem, we propose using a two-phase approach in which, once all distances are calculated as above, the activities are first ranked in order of increasing distance. Subsequently, the K reference activities that rank higher, *i.e.* those that exhibit the greatest similarity with the test activity, are used for the design of a weight map that is aimed to facilitate discrimination among these K activities. Apparently, we need the actual matching reference activity to always be among the K best matches in order to be able to recognize the test activity in the second phase of the classification process. However, the greater K is, the lower the efficiency of the weighted approach will be. In this work, we use $K = N/3 = 4$, as it was found that this choice represents a good compromise between recognition efficiency in the two phases of the algorithm. The impact of choice of K in the first phase of the algorithm is shown in Table 2. As seen, in the vast majority of cases, the actual matching reference activity is among the four best matches.

The weight map calculated based on the K highest ranking activities is

	Rank							
Act No.	1	2	3	**4**	5	6	7	8
1	72	83	97	100	100	100	100	100
2	83	95	100	100	100	100	100	100
3	87	100	100	100	100	100	100	100
4	98	100	100	100	100	100	100	100
5	100	100	100	100	100	100	100	100
6	100	100	100	100	100	100	100	100
7	83	97	100	100	100	100	100	100
8	37	53	62	85	98	100	100	100
9	82	87	95	100	100	100	100	100
10	35	57	78	88	100	100	100	100
11	73	87	93	100	100	100	100	100
12	58	60	63	87	92	100	100	100

Table 2: Cumulative match scores for the performance (in percent) of the first phase of the classification algorithm.

now tailored to the task of distinguishing between activities that, despite being different, they look similar to the test activity. This approach is expected to be more efficient than discrimination techniques that are based on all activities in the database.

For the calculation of the weight map, we denote the spatiotemporal profile of the kth ranked reference activity as:

$$\mathbf{R}_k = \{\mathbf{r}_{k1}, \mathbf{r}_{k2}, \mathbf{r}_{k3}, \mathbf{r}_{k4}\}, \qquad k = 1, 2, \ldots, K \tag{14}$$

In the above expression, k is index of the the ranked reference activities, *i.e.*, $\mathbf{R}_1$ is the spatiotemporal profile of the reference activity that exhibits the smallest distance with the test activity, $\mathbf{R}_2$ exhibits the second smallest such distance and so on. We calculate the weight map based on the profile coefficients that appear to contribute to the discrimination among the K ranked profiles $\mathbf{R}_{ks}, k = 1, 2, \ldots, K$, that correspond to the activities that are most similar to the test activity.

We define the total *"between"* difference $\mathbf{v}_s^B(x, y)$ in pixel position (x, y) between *different* ranked activities as:

$$\mathbf{v}_{Bs}(x, y) = \frac{1}{K^2} \sum_{k=1}^{K} \sum_{l=1}^{K} |\mathbf{r}_{ks}(x, y) - \mathbf{r}_{ls}(x, y)|, \qquad s = 1, \ldots, 4 \tag{15}$$

As seen, a separate difference matrix is calculated for each activity stage s. Considering the symmetricity of the template differences in eq. (15), the above expression can be equivalently written as:

$$\mathbf{v}_{Bs}(x, y) = \frac{1}{K^2} \sum_{k=1}^{K-1} \sum_{l=k+1}^{K} 2|\mathbf{r}_{ks}(x, y) - \mathbf{r}_{ls}(x, y)|, \qquad s = 1, \ldots, 4 \tag{16}$$

Subsequently, for the K ranked activities, we calculate a total *"within"* difference matrix using H different instances of the *same* activity:

$$\mathbf{v}_s(i, j) = \frac{1}{KH^2} \sum_{k=1}^{K} (\sum_{b=1}^{H-1} \sum_{c=b+1}^{H} 2|\mathbf{r}_{ks}^b(x, y) - \mathbf{r}_{ks}^c(x, y)|), \qquad s = 1, \ldots, 4 \tag{17}$$

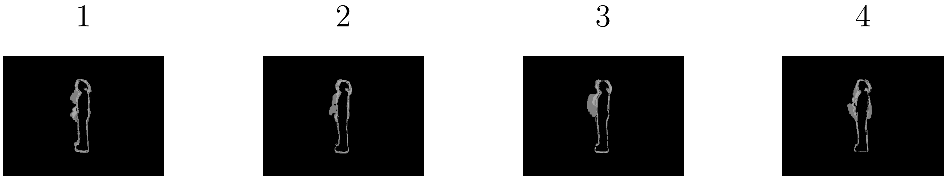

Figure 5: Weight map for a set of best matches comprising of activities: *check watch*, *cross arms*, *scratch head*, and *wave*.

In a way that is reminiscent of Linear Discriminant Analysis, when applying eq. (13), we can emphasize "between" differences and suppress "within" differences by using weighting coefficients calculated based on the ratio of eq. (16) and (17). Specifically, the elements $\mathbf{w}_s(x, y)$ of the weight map can be calculated as:

$$\mathbf{w}_s(x, y) = \frac{\mathbf{v}_{Bs}(x, y)}{L + \mathbf{v}_s(x, y)}, \qquad s = 1, \ldots, 4 \tag{18}$$

where L is a small number that is aimed to prevent the denominator of the right-hand side from becoming zero (in our experiments we used $L = 0.5$).

A weight map determined based on four activities: *check watch*, *cross arms*, *scratch head*, and *wave*, is shown in Fig 5. As can be seen, despite the fact that the differences between these activities are very subtle, recognition is facilitated by focusing the recognition process on exactly these differences. This performance would not have been possible if the weight map calculation had been based on all activities in the database.

3.3. Recognition

Once the weight map has been determined, weighted template distances are calculated between the test activity and the reference activity templates. The weighted template distance is defined as:

$$\tilde{T}_D[a] = \tilde{d}(\mathbf{T}_g, \mathbf{T}_a) \triangleq \sum_{s=1}^{4} \tilde{d}(\mathbf{t}_{gs} - \mathbf{t}_{as}) \tag{19}$$

and the associated total weighted distance is:

$$\tilde{D}[a] = \tilde{T}_D[a] + qM_D[a], \qquad a = 1, 2, \ldots, N \tag{20}$$

where the value of the parameter q is selected according to the process described in the beginning of this section.

The system recognizes the test activity based on the minimum total weighted distance among all results:

$$G = \arg\min_{a} \tilde{D}[a] \tag{21}$$

where G is the index of the recognized activity.

4. Experimental Results

In order to evaluate the performance of our system, we tested the proposed algorithm on the INRIA Xmas Motion Acquisition Sequences (IXMAS) Database [6]. The INRIA multi-view database includes 12 daily-life activities each performed 3 times by 12 actors. Surrounded with 5 fixed cameras, each capturing 23 frames per second, the actors freely choose their position and orientation while they perform the activities. All 12 activities are performed in the same order, but with a different execution rate, depending on the actors. For the evaluation of our method, we used 72 sequences, *i.e.*, 72 different instances of each activity. Therefore, we used 864 (72×12) activity executions in total.

In our experiments, we used views "1" and "2" from the INRIA database which are different as they are captured using different cameras. For the construction of the *reference* (i.e., training) spatiotemporal profiles and the extraction of the *reference* motion profiles, we used twelve activity sequences, which were chosen randomly from these two views (six from each). Each of these reference sequences contained all 12 activities. This means that 144 (12×12) activity executions were used for training. The remaining 720 (60×12) activity executions were used as test sequences.

Initially, we applied our baseline method, using template and motion information, without applying any weighting on the spatiotemporal profiles. The first three columns of Table 3 report results based on the independent application of the motion profile, the spatiotemporal *Centered MEI profile* (CMEI), as well as their combination (CMM). As seen, the performance of these features when used independently is not always good. However, if they are combined using eq. (20), then the resulting method, termed *Centered MEI with Motion* (CMM), exhibits apparent performance improvements, especially if compared with the independent use of the motion feature.

Subsequently, we applied the two-phase process described in Section 3. The four best matches for each given test activity were calculated and a weight map was designed in order to facilitate recognition among these four

			Baseline		Weighted	
No.	Action	Motion	CMEI	CMM	wCMEI	wCMM
1	Check Watch	61.67	70.00	71.67	88.33	**91.67**
2	Cross Arms	45.00	76.67	83.33	86.67	**90.00**
3	Scratch Head	46.67	83.33	86.67	81.67	**88.33**
4	Sit Down	**100**	96.67	98.33	98.33	98.33
5	Get Up	**100**	**100**	**100**	**100**	**100**
6	Turn & Walk	**100**	98.33	**100**	**100**	**100**
7	Wave	33.33	81.67	83.33	83.33	**85.00**
8	Punch	21.67	36.67	36.67	**68.33**	**68.33**
9	Kick	31.67	81.67	81.67	85.00	**86.67**
10	Point	43.33	33.33	35.00	61.67	**63.33**
11	Pick up	76.67	68.33	73.33	80.00	**81.67**
12	Throw	31.67	56.67	58.33	71.67	**76.67**
Average		57.64	73.61	75.69	83.75	**85.83**

Table 3: Activity recognition rates by using motion profiles, CMEI templates, combined CMM profiles, and discriminate weighting.

matches. Results are reported in the last two columns of Table 3 for the weighted CMEI (wCMEI) profile, and the combined *weighted CMEI with motion*, termed wCMM. As seen, the recognition rate is very considerably improved when compared with the un-weighted CMM method. Despite its simplicity, the combination of the motion profile with the weighted spatiotemporal profile yields excellent performance. Using our current system, the test activity sequences are recognized correctly at an average recognition rate of

No.	Action	1	2	3	4	5	6	7	8	9	10	11	12
1	Check Watch	91.7	3.3	3.3	0	0	0	1.7	0	0	0	0	0
2	Cross Arm	5.0	90.0	3.3	0	0	0	1.7	0	0	0	0	0
3	Scratch Head	5.0	3.3	88.3	0	0	0	3.3	0	0	0	0	0
4	Sit Down	0	0	0	98.3	0	0	0	0	0	0	1.7	0
5	Get Up	0	0	0	0	100	0	0	0	0	0	0	0
6	Turn & Walk	0	0	0	0	0	100	0	0	0	0	0	0
7	Wave	3.3	1.7	6.7	0	0	0	85.0	0	0	1.7	0	1.7
8	Punch	6.7	0	8.3	0	0	0	5	68.3	0	10	0	1.7
9	Kick	0	1.7	0	1.7	0	0	0	3.3	86.7	1.7	1.7	3.3
10	Point	3.3	8.3	5	0	0	0	3.3	13.3	0	63.3	0	3.3
11	Pick Up	0	0	0	8.3	3.3	0	0	1.7	3.3	0	81.7	1.7
12	Throw	5	0	1.7	0	0	0	10	3.3	0	3.3	0	76.7

Table 4: Confusion Matrix of our final system on the INRIA Database.

85.83%, which constitutes a significant improvement on the performance of the baseline system. As will be discussed later, this performance also constitutes an improvement over other recently published methods, such as those in [14], [15], [16]. The confusion matrix reporting confusion between activities recognized by the proposed wCMM system is shown in Table 4. Table 4 shows that the system is occasionally prone to confuse the “point” and the “punch” activity, which is consistent with the results presented in Table 3. The less satisfactory performance on these two activities is due to their inherent similarity as well as the great variability with which subjects are performing the “punch” and “point” activities in the testing set that we use

No.	Action	inter	intra
1	Check Watch	88.33	93.33
2	Cross Arms	90.00	91.67
3	Scratch Head	86.67	91.67
4	Sit Down	98.33	98.33
5	Get Up	100	100
6	Turn & Walk	100	100
7	Wave	83.33	88.33
8	Punch	58.33	68.33
9	Kick	80.00	85.00
10	Point	63.33	63.33
11	Pick up	81.67	81.67
12	Throw	73.33	76.67
Average		83.61	86.53

Table 5: Evaluation of the proposed wCMM method under viewpoint variations.

for our experiments.

In order to test the performance of our system under viewpoint variation, two views with moderate differences are chosen. We report results in two forms, first we use different views for training and testing, and then we train and test using activity sequences from the same view. The results are shown in Table 5. As seen, although there is a decrease in recognition performance in the cross-view experiment, the decrease is not dramatic and demonstrates that our system can work well even when the actual view is different from the assumed one.

Finally, we compared our wCMM method with a variety of other existing techniques for activity recognition. Specifically, the other methods in our comparison are the Action Net method [14], the Action Manifolds [15], as well as the method in [16]. The recognition performance of our system in comparison to the recognition performance of other approaches is shown in Table. 6. As seen, our wCMM method outperforms the other methods in the comparison for activity recognition, which reinforces our confidence about the advantages that our approach offers.

Method	wCMM	Action Net [14]	Action Manifolds [15]		VWCs [16]
View	single	multiple	multiple	single	multiple
Recognition Rate	**85.83**	80.6	83.1	80.3	78.5

Table 6: Comparison of our proposed method in comparison to other competing methods in terms of average recognition performance.

5. Conclusion

In this paper, we presented a method for the recognition of human activities. The proposed approach was based on the construction of a set of templates for each activity as well as on the measurement of the motion in each activity. Templates were designed so that they capture the structural and motion information that is most discriminative among activities. The direct motion measurements capture the amount of translational motion in each activity. The two features are fused at the recognition stage. Recognition is achieved in two steps by calculating the similarity between the templates and the motion features of the test and reference activities. The proposed methodology yielded excellent results when applied on the INRIA database.

6. Acknowledgements

This work was supported in part by the European Commission under Contract FP7-215372 ACTIBIO

The authors would like to thank the anonymous reviewers for their careful and constructive review, which resulted in a greatly improved manuscript.

[1] E.Muybridge, The Human Figure in Motion, Dover Pulications, 1901.

[2] J. K. Aggarwal, Q. Cai, Human motion analysis: A review, Computer vision and image understanding 73 (3) (1999) 428–440.

[3] D. M. Gavrila, The visual analysis of human movement: a survey, Computer vision and image understanding 73 (1) (1999) 82–98.

[4] W. Liang, H. Weiming, T. Tan, Recent developments in human motion analysis, Pattern Recognition 36 (3) (2003) 585–601.

[5] A. F. Bobick, J. W. Davis, The recognition of human movement using temporal templates, IEEE Tran. on Pattern Analysis and Machine Intelligence 23 (3) (2001) 257–267.

[6] D. Weinland, R. Ronfard, E. Boyer, Free viewpoint action recognition using motion history volumes, Computer vision and image understanding 104 (2-3) (2006) 249–257.

[7] A. Laurentini, The visual hull concept for silhouette-based image understanding, IEEE Tran. on Pattern Analysis and Machine Intelligence 16 (1994) 150–162.

[8] M. Piccardi, O. Perez, Hidden markov models with kernel density estimation of emission probabilities and their use in activity recognition, in: IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, Minnesota, US, 2007, pp. 1–8.

[9] F. Niu, M. A. Mottaleb, Hmm-based segmentation and recognition of human activities from video sequences, in: IEEE Int. Conf. on Multimedia and Expo, Amsterdam, Netherlands, 2005, pp. 804–807.

[10] F. I. Bashir, A. A. Khokhar, D. Schonfeld, Object trajectory-based activity classification and recognition using hidden markov models, in: IEEE Tran. on Image Processing, 2007, pp. 1912 – 1919.

[11] J. J. L. Md. Zia Uddin, T. S. Kim, Independent component feature-based human activity recognition via linear discriminant analysis and hidden markov model, in: IEEE EMBS Conf., 2008, pp. 5168–5171.

[12] S. Cherla, K. Kulkarni, A.Kale, V. Ramasubramanian, Towards fast, view-invariant human action recognition, in: IEEE Conf. on Computer Vision and Pattern Recognition Workshops, 2008, pp. 1–8.

[13] P. Turaga, A. Veeraraghavan, R. Chellappa, Unsupervised view and rate invariant clustering of video sequences, Computer vision and image understanding 113 (2009) 353–371.

[14] F. Lv, R. Nevatia, Single view human action recognition using key pose matching and viterbi path searching, Computer Vision and Pattern Recognition CVPR 2007 2008 (2007) 1–8.

[15] M. Lewandowski, D. Makris, J. Nebel, View and style-independent action manifolds for human activity recognition, Computer vision 6316 (2010) 547–560.

[16] J. Liu, S. Ali, M. Shah, Recognizing human actions using multiple features, Computer Vision and Pattern Recognition CVPR 2008 2008 (2008) 1–8.

[17] D. Weinland, R. Ronfard, E. Boyer, Automatic discovery of action taxonomies from multiple views, in: IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 2006, pp. 1639 – 1645.

[18] P. H. R.O. Duda, D. Stork, Pattern Classification, John Wiley & Sons, Inc., 2001.

[19] M. Turk, A. Pentland, Face recognition using eigenfaces, in: Conf. on Computer Vision and Pattern Recognition, 1991.