Spatiotemporal Salient Points for Visual Recognition of Human Actions

Antonios Oikonomopoulos, Ioannis Patras, and Maja Pantic

***Abstract*—This paper addresses the problem of human-action recognition by introducing a sparse representation of image sequences as a collection of spatiotemporal events that are localized at points that are salient both in space and time. The spatiotemporal salient points are detected by measuring the variations in the information content of pixel neighborhoods not only in space but also in time. An appropriate distance metric between two collections of spatiotemporal salient points is introduced, which is based on the chamfer distance and an iterative linear time-warping technique that deals with time expansion or time-compression issues. A classification scheme that is based on relevance vector machines and on the proposed distance measure is proposed. Results on real image sequences from a small database depicting people performing 19 aerobic exercises are presented.**

***Index Terms*—Human-action recognition, relevance vector machines (RVMs), salient regions, spatiotemporal actions, spatiotemporal saliency.**

I. Introduction

Analysis and interpretation of image sequences have received a great amount of interest in computer vision for the last few years. Applications in the areas of video indexing, intelligent autonomous systems, man–machine interaction, and surveillance, to name just a few, reveal the importance of developing systems that are able to extract semantically rich representations of image sequences and to interpret them accordingly. Recognition and interpretation of human activities is a significant research area by itself since a large amount of the information in image sequences is carried in the human action.

In order to obtain a semantic description of the content of a scene, we do not need to use all the available information. Determining which part of the visual information is relevant is an open problem because it naturally depends on the semantic description that we wish to obtain. However, for a lot of applications, a good description of the scene can be obtained by considering the information around certain points of interest such as corners and edges—that is, in areas that are rich in information. For content-based image-retrieval applications, the notion of interesting points has been extensively used. According to Haralick and Shapiro [5], an interesting point is: 1) distinguishable from its neighbors and 2) its position is invariant with respect to the expected geometric transformation and to radiometric distortions. Schmid and Mohr [14] detect interesting points using a Harris corner detector and estimate gray-value differential image invariants [9], [17] at different scales. Loupias *et al.* [12] use wavelet-based salient-point detectors in order to detect global and local variations in images for content-based image-retrieval applications. Gilles introduces the notion of saliency in terms of local signal complexity or unpredictability in [4]. Detectors of interesting points are compared in [15] and [16] in terms of repeatability rate and information content.

An important issue in salient-point detection is automatic selection of the scale at which the salient points will be detected and the local features will be extracted. Lindeberg [11] integrates a scale-space approach for corner detection and search for local extremes across scales. Itti *et al.* [7] use a dyadic Gaussian pyramid approach in order to construct saliency maps from given images. The spatial distribution of each saliency map is then modeled with a dynamical neural network, in order to select locations of attention in the images. Kadir and Brady [8] extend the original Gilles algorithm and estimate the information content in circular neighborhoods at different scales in terms of the entropy. Local extremes of changes in the entropy across scales are detected and the saliency of each point at a certain scale is defined in terms of both the entropy and its rate of change at the scale in question. In addition, they develop a clustering algorithm in order to form salient regions from groups of salient points that are detected at similar locations and scales. In [6], the performance of the salient-point detector developed in [8] is examined, and an object-recognition approach using keypoints is described by Lowe [13]. The spatiotemporal arrangement of the detected salient points is then used for content-based image retrieval.

While a large amount of work has been done on image-based retrieval and object recognition, the concept of saliency has only recently begun to be used for space–time content-based video retrieval and for activity recognition. In [10], a Harris corner detector is extended in the temporal dimension, leading to a number of corner points in time, called space–time interest points. The resulting interesting points correspond roughly to points in space–time where the motion abruptly changes direction, such as stopping or starting. The resulting representations are then compared using a Mahalanobis distance metric, while k-means clustering is used to group similar events. In [3], hand gestures are recognized by using a hierarchical hand model consisting of the palm and the fingers. Color features under different scales are then extracted and a particle-filtering approach is implemented to simultaneously track and detect the different motion states occurring in the image sequence. However, the algorithm is customized solely for hand-gesture recognition. Another interesting spatiotemporal representation for human-activity recognition is presented in [1], where a temporal-template approach is implemented. The input image sequence is used to construct motion energy images (MEIs) and motion history images (MHIs), for determining where and when, respectively, motion occurs in the sequence. For recognition, a set of moment invariants is calculated for each resulting image and a Mahalanobis distance metric is applied between the sets in order to discriminate different kinds of motion.

In this paper, we detect spatiotemporal features in given image sequences by extending in the temporal direction the information-theoretic salient-feature detector developed in [8]. Our goal is to obtain a sparse representation of a human action as a set of spatiotemporal points that correspond to activity-variation peaks. In contrast to the work of Laptev and Lindeberg [10], in which a sequence is represented by the local activity endpoints (starts/stops), our representation contains the spatiotemporal points at which there are peaks in activity variation such as the edges of a moving object. Like the authors of [8], we automatically detect the scales at which the entropy achieves local maxima and form spatiotemporal salient regions by clustering spatiotemporal points with similar location and scale. Each image sequence is then represented as a set of spatiotemporal salient points, the locations of which are normalized in order to achieve invariance against the translation of the subjects performing the actions. We use the chamfer distance as an appropriate distance metric between two representations. In order to deal with different speeds in the execution

Manuscript received January 19, 2005; revised July 14, 2005. This work was supported in part by the Netherlands BSIK-MultimediaN-N2 Interaction project. The work of A. Oikonomopoulos was supported by the Greek State Scholarships Foundation (IKY). This paper was recommended by Associate Editor P. Bhatlacharya.

A. Oikonomopoulos and M. Pantic are with the Computing Department, Imperial College London, SW7 2AZ, U.K. (e-mail: A.Oikonomopoulos@gmail.com; MPantic@ieee.org).

I. Patras is with the Computer Vision and Pattern Recognition Group, Computer Science Department, The University of York, York YO10 5DD, U.K. (e-mail: I.Patras@cs.york.ac.uk).

Digital Object Identifier 10.1109/TSMCB.2005.861864

of the actions and to achieve invariance against the subjects' scaling, we propose a linear space–time-warping technique that tries to linearly warp any two examples by minimizing their chamfer distance. A simple k-nearest neighbor (kNN) classifier and one based on relevance vector machines (RVMs), introduced in [18], are used in order to test the efficiency of the representation. We test the proposed method using real image sequences, where we use aerobic exercises as our test domain. Our experimental results show fairly good discrimination between specific motion classes.

The remainder of the paper is organized as follows. In Section II, the spatiotemporal-feature detector used is described in detail. In Section III, the proposed recognition method is analyzed, including the proposed space–time-warping technique. In Section IV, we present our experimental results, and in Section V, final conclusions are drawn.

II. Spatiotemporal Salient Points

A. *Spatiotemporal Saliency*

Let us denote by $N_{\mathrm{c}}(s, \vec{v})$ the set of pixels in an image I that belongs to a circular neighborhood of radius s, which is centered at pixel $\vec{v} = (x, y)$. In [8], in order to detect salient points in static images, Kadir and Brady define a saliency metric $y_D(s, \vec{v})$ based on measuring changes in the information content of N_{c} for a set of different radii (i.e., scales). In order to detect spatiotemporal salient points at peaks of activity variation, we extend Kadir's detector by considering cylindrical spatiotemporal neighborhoods at different spatial radii s and temporal depths d. More specifically, let us denote by $N_{\mathrm{cl}}(\vec{s}, \vec{v})$ the set of pixels in a cylindrical neighborhood of scale $\vec{s} = (s, d)$ centered at the spatiotemporal point $\vec{v} = (x, y, t)$ in the given image sequence. At each point $\vec{v}$ and for each scale $\vec{s}$, we will define the spatiotemporal saliency $y_D(\vec{s}, \vec{v})$ by measuring the changes in the information content within $N_{\mathrm{cl}}(\vec{s}, \vec{v})$. Since we are interested in activity within an image sequence, we consider as input signal the convolution of the intensity information with a first-order Gaussian derivative filter. Gaussian derivative filters have been extensively used for detecting interesting points in static images. Here, we apply them in the temporal domain in order to arrive at a measure of activity. Formally, given an image sequence $I_0(x, y, t)$ and a filter G_t, the input signal that we use is defined as

$$I(x, y, t) = G_t * I_0(x, y, t). \qquad (1)$$

For each point $\vec{v} = (x, y, t)$ in the image sequence, let us calculate the Shannon entropy of the signal histogram in a spatiotemporal neighborhood around it. Let us note that we considered cylindrical spatiotemporal neighborhoods of radius s and depth d for simplicity reasons. However, more complicated shapes, such as elliptical neighborhoods at different orientations and with different axes ratios, could be considered.

The signal entropy $H_D(\vec{s}, \vec{v})$ in the spatiotemporal neighborhood $N_{\mathrm{cl}}(\vec{s}, \vec{v})$ is given by

$$H_D(\vec{s}, \vec{v}) = -\int\limits_{q \in D} p_D(\vec{s}, \vec{v}) \log_2 p_D(\vec{s}, \vec{v}) dq \qquad (2)$$

where $p_D(\vec{s}, \vec{v})$ is the probability density of the signal histogram as a function of scale $\vec{s}$ and position $\vec{v}$. By q we denote the signal value and by D the set of all signal values. In this paper, we use the values that arise from (1) as signal values. It is possible, however, to use other kinds of descriptors, such as optical flow vectors. We use the histogram method to approximate the probability density $p_D(\vec{s}, \vec{v})$. Alternatively, the probability density can be estimated using Parzen window density estimation or any other density-estimation technique.

Subsequently, we proceed with the automatic selection of the scale [8], [11]. More specifically, we consider the scales at which the entropy values achieve a local maximum as candidate salient scales. Let us define as $\hat{S}_{\mathrm{p}}$ the set of scales at which the entropy is peaked, i.e.,

$$\hat{S}_{\mathrm{p}} = \left\{ \vec{s} : \frac{\partial H_D(\vec{s}, \vec{v})}{\partial s} = 0 \bigwedge \frac{\partial H_D(\vec{s}, \vec{v})}{\partial d} = 0 \bigwedge \frac{\partial^2 H_D(\vec{s}, \vec{v})}{\partial s^2} < 0 \bigwedge \frac{\partial^2 H_D(\vec{s}, \vec{v})}{\partial d^2} < 0 \right\}. \qquad (3)$$

Then, following the approach of [8], we can define the saliency metric at the candidate scales as follows:

$$y_D(\vec{s}, \vec{v}) = H_D(\vec{s}, \vec{v}) W_D(\vec{s}, \vec{v}) \qquad \forall \vec{s} \in \hat{S}_{\mathrm{p}}. \qquad (4)$$

Equation (4) gives a measure of how salient a spatiotemporal point $\vec{v}$ is at certain candidate scales $\vec{s}$. The first term of (4) is a measure of the variation in the information content of the signal. The weighting function $W_D(\vec{s}, \vec{v})$ is a measure of how prominent the local maximum is at $\vec{s}$, and is given by

$$W_D(\vec{s}, \vec{v}) = s \int\limits_{q \in D} \left| \frac{\partial}{\partial s} p_D(\vec{s}, \vec{v}) \right| dq + d \int\limits_{q \in D} \left| \frac{\partial}{\partial d} p_D(\vec{s}, \vec{v}) \right| dq \qquad \forall (s, d) \in \hat{S}_{\mathrm{p}}. \qquad (5)$$

When a peak in the entropy for a specific scale is distinct, then the corresponding pixel probability density functions at the neighboring scales will differ substantially, giving a large value to the integrals of (5) and thus, to the corresponding weight value assigned. On the contrary, when the peak is smoother, then the integrals in (5), and therefore the corresponding weight, will have a smaller value.

In Fig. 1(b), the form of the entropy in (2) is presented, for the corresponding action whose one instance is shown in Fig. 1(a). The scale that corresponds to the distinct peak of the plot is considered a candidate salient scale, and is assigned a saliency value, according to (4) and (5).

B. *Salient Regions*

The analysis of the previous section leads to a set of candidate spatiotemporal salient points $S = \{(\vec{s}_i, \vec{v}_i, y_{D,i})\}$, where $\vec{v}_i = (x, y, t)$, $\vec{s}_i = (s_i, d_i)$, and $y_{D,i}$ are, respectively, the position vector, the scale, and the saliency value of the feature point with index i. In order to make the feature detector more robust against noise and to reduce the dimensionality of the resulting feature space, we follow a similar approach as that in [8] and develop a clustering algorithm that we apply to the detected salient points. We define in this way corresponding salient regions instead of salient points. The location of these regions should be more stable than the individual salient points, since noise is unlikely to affect all of the points within the region in the same way. The proposed clustering algorithm removes salient points with low saliency value and creates clusters that are: 1) well localized in space, time, and scale; 2) sufficiently salient; and 3) sufficiently distant from each other.

The steps of the proposed algorithm can be summarized as follows.

1) Derive a new set S_T from S by applying a global threshold T to the saliency of the points that consist S. Thresholding removes salient points with low saliency, that is

$$S_T = \{(\vec{s}_i, \vec{v}_i, y_{D,i}) : y_{D,i} > T\}. \qquad (6)$$

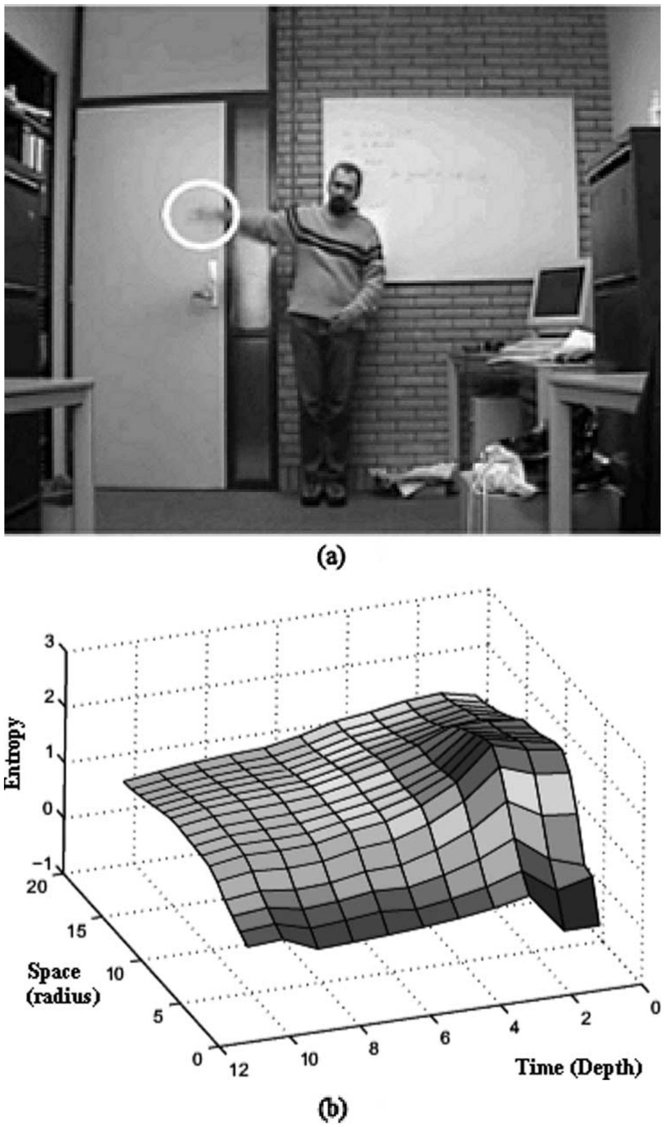

Fig. 1. (a) Single frame from a sample image sequence where the subject is raising its right hand, and (b) the corresponding entropy plot as a function of the spatial radius and temporal depth of all the applied cylindrical neighborhoods. The origin of all the applied cylindrical neighborhoods is the center of the white circle in (a).

2) Select the point with index i in S_T, which has the highest saliency value. Use the salient point i as a seed to initialize a salient region R_k (in the first iteration, $k = 1$). That is

$$R_k = \{i\}. \tag{7}$$

3) Add nearby points j to the region R_k as long as the intracluster variance does not exceed a threshold T_{V}. That is, as long as

$$\frac{1}{|R_k|} \sum_{j \in R_k} d_j^2 < T_{\mathrm{V}} \tag{8}$$

where R_k is the set of the points in the current region k and d_j is the Euclidean distance of the jth point from the seed point i.

4) If the overall saliency of the region R_k is lower than a saliency threshold T_{S}, that is

$$\sum_{j \in R_k} y_{D,j} \leq T_{\mathrm{S}} \tag{9}$$

discard the points in the region back to the initial set of points and continue from step 2) with the next highest salient point. Otherwise, calculate the Euclidean distance of the center of region R_k from the center of salient regions already defined in the previous steps of the algorithm—that is, from salient regions $R_{k'}$, $k' < k$.

5) If the distance is lower than the average scale of the region, discard the points in the region, put them back to the initial set of points, and continue from step 2) with the next highest salient point. Otherwise, accept the region as a new cluster and store it as the mean scale and spatial location of the points in it.

6) Form a new set S_T consisting of the remaining salient points, increase the cluster index k and continue from step 2) with the next highest salient point.

By setting the threshold T_{V} in step 2), we define clusters that have local support and are well localized in space and time. In this way, we avoid clusters with large variance in their spatiotemporal position and scale. In addition, we want to take the saliency of the points into consideration such that the overall saliency of the region is sufficient. We do this in step 3), by setting a saliency threshold T_{S}. Finally, the purpose of step 4) is to accept and create clusters that are sufficiently distant from each other. To summarize, a new cluster is accepted only if it has sufficient local support, its overall saliency value is above the saliency threshold, and it is sufficiently distant in terms of Euclidean distance from already-existing clusters.

We set the global threshold T of the first step of our clustering algorithm equal to 10% of the maximum saliency value. In order to ensure the sparseness of the resulting representation, we set the variance threshold equal to half the maximum spatial scale of our cylindrical sampling window. Furthermore, we set the saliency threshold equal to 0.1% of the global detected saliency of the scene. We have found empirically that these values were a reasonable compromise between the amount of noise that we wish to remove and the actual signal values that we want to keep. However, cross-validation methods could be used for the selection of the thresholds.

The algorithm described above requires estimation of the spatiotemporal saliency metric (4) for each point in the image sequence. This involves calculations at spatiotemporal neighborhoods at different scales, which can be computationally very expensive. For N pixels in an image sequence, $O(N(sd)(s^2d))$ number of operations are required in order to calculate the entropy, where (sd) is proportional to the number of the cylindrical neighborhoods used and (s^2d) is proportional to the average number of pixels per cylindrical neighborhood. In order to reduce the computational complexity, we also propose a two-step approach for the detection of salient points, which is an approximation of the full-search approach. In the experimental section, we will present and compare results from both approaches. In the first step of the proposed approximation approach, we select only salient points in space for every frame of the sequence. Among these detected points, there may be some that are also salient in time. We detect these in a second step, by extending the salient-feature detector in the temporal dimension. By applying this procedure, we discard image points that are not salient in space, and therefore, not likely to be salient in time. For the two-step approach, a number of operations proportional to $O(Ns(s^2) + R(s^2d)(sd))$ are needed in order to calculate the entropy. The first term of the summation is proportional to the operations needed for the detection of salient points using only spatial information. More specifically, s is proportional to the number of circular neighborhoods used and s^2 is proportional to the average number of pixels per circular neighborhood. The second term is very similar to the complexity of the full search, only in this case, R (the total number of pixels in the sequence for which the spatial entropy is maximized) is used instead of N. In general, R is one order of magnitude smaller than N, yielding a substantial reduction in the complexity of the specific approach. More specifically, the two-step approach is as follows.

1) In the first step, we detect salient regions in the spatial domain—that is, for every individual frame—without taking into account neighboring frames. The computational gain is due to the use of circular neighborhoods instead of cylindrical ones. The position vector $\vec{v}$ in (2)–(5) is two-dimensional in this case, $\vec{v} = (x, y)$, where $1 \leq x \leq N_1$ and $1 \leq y \leq N_2$. The above procedure leads to the creation of feature sets of the form $F_t = \{(x_{t,i}, y_{t,i}, s_{t,i}, y_{Dt,i}), 1 \leq t \leq K, 1 \leq i \leq L_t\}$, where t is the frame number and L_t is the total number of salient points detected in frame t.
2) In the second step, we set $\vec{v} = (x_{t,i}, y_{t,i}, t)$, $1 \leq t \leq K$, $1 \leq i \leq L_t$, and we apply (2)–(5) for cylindrical neighborhoods of scale $\vec{s} = (s, d)$. After clustering the detected spatiotemporal salient points, we derive a feature set consisting of salient regions in the space–time domain, $F = \{(x_j, y_j, t_j, \vec{s}_j, y_{Dj}), 1 \leq j \leq L\}$, where L is the number of salient regions detected.

III. Recognition of Spatiotemporal Actions

Using the feature-detection scheme described in Section II, we represent a given image sequence by a set of features, where each feature corresponds to a cylindrical salient region of the image sequence in the space–time domain. In what follows, we will define an appropriate distance metric that can be used subsequently for learning and recognition of human actions in image sequences. Indeed, a wide variety of classification schemes, ranging from kNN to support vector machines (SVMs), depends on the definition of an appropriate distance metric. We use the chamfer distance [2], as it can provide a distance measure between feature sets with unequal number of features. More specifically, for two feature sets $F = \{(x_i, y_i, t_i), 1 \leq i \leq M\}$ and $F' = \{(x'_j, y'_j, t'_j), 1 \leq j \leq M'\}$ consisting of an M and M' number of features, respectively, the chamfer distance of the set F from the set F' is defined as follows:

$$D(F, F') = \frac{1}{M} \sum_{i=1}^{M} \min_{j=1}^{M'} \sqrt{\left(x'_j - x_i\right)^2 + \left(y'_j - y_i\right)^2 + \left(t'_j - t_i\right)^2}. \tag{10}$$

In other words, the proposed distance is defined as the average over the set of the minimum Euclidean distances between the M feature points of set F and the M' feature points of set F'. Chamfer-distance transformations have been used in [2] with edge matching in order to match images of different resolutions. Here, since the number of matching points in the corresponding representations is relatively small, we loop through (10) in order to find the best matching points.

The distance metric of (10) is not symmetrical, since $D(F, F') \neq D(F', F)$. For recognition purposes, it is desirable to select a distance metric that is symmetrical. A metric that satisfies this requirement is the average of $D(F, F')$ and $D(F', F)$, that is

$$D_{\mathrm{c}}(F, F') = \frac{1}{2}\left(D(F, F') + D(F', F)\right). \tag{11}$$

Let us note that for the calculation of the distance metric we only consider the spatiotemporal position of the detected salient points.

A. Space–Time Warping

There is a large amount of variability between feature sets due to differences in the execution speed of the corresponding actions from subject to subject. Furthermore, we need to compensate for possible shifting of the representations forward or backward in time, caused by imprecise segmentation of the corresponding actions. To cope with both these issues, we have developed a linear time-warping technique with which we model variations in time using a time-scaling parameter a and a time-shifting parameter b. In addition, in order to achieve invariance against scaling of the subjects performing the actions, we introduce a scaling parameter σ in the proposed time-warping technique. Prior to warping, we transform the x and y coordinates of the detected salient regions in each sequence so that they have zero mean value. We do this in order to achieve invariance against translation. The parameters a, b, and σ are estimated with a gradient-descent iterative scheme that minimizes the chamfer distance between the sets. More specifically, let us denote by $F_w = \{(\sigma x_i, \sigma y_i, a \cdot t_i - b), 1 \leq i \leq M\}$ the feature set F with respect to feature set F'. Then, the distance between F' and F_w is given by (10) as

$$D(F_w, F') = \frac{1}{M} \sum_{i=1}^{M} \min_{j=1}^{M'} \times \sqrt{\left(x'_j - \sigma x_i\right)^2 + \left(y'_j - \sigma y_i\right)^2 + \left(t'_j - a \cdot t_i + b\right)^2}. \tag{12}$$

Similarly, the feature set F' with respect to feature set F can be represented as $F'_w = \{((1/\sigma)x'_j, (1/\sigma)y'_j, (1/a) \cdot t'_j + b), 1 \leq j \leq M'\}$, and their distance, as given by (10), as

$$D\left(F'_w, F\right) = \frac{1}{M'} \sum_{j=1}^{M'} \min_{i=1}^{M} \times \sqrt{\left(x_i - \frac{1}{\sigma}x'_j\right)^2 + \left(y_i - \frac{1}{\sigma}y'_j\right)^2 + \left(t_i - \frac{1}{a} \cdot t'_j - b\right)^2}. \tag{13}$$

The distance to be optimized follows from the substitution of (12) and (13) to (11). We follow an iterative gradient-descent approach for the adjustment of the a, b, and σ parameters. The update rules are given by

$$a^{n+1} = a^n - \lambda_1 \frac{\partial D_{\mathrm{c}}}{\partial a^n} \tag{14}$$

$$b^{n+1} = b^n - \lambda_2 \frac{\partial D_{\mathrm{c}}}{\partial b^n} \tag{15}$$

$$\sigma^{n+1} = \sigma^n - \lambda_3 \frac{\partial D_{\mathrm{c}}}{\partial \sigma^n} \tag{16}$$

where λ_1, λ_2, and λ_3 are the learning rates, and n is the iteration index. The algorithm iteratively adjusts the values of a, b, and σ towards the minimization of the chamfer distance between the two feature sets, given by (11). In order to determine the values of a, b, and σ, we need to know the values of $\partial D_{\mathrm{c}}/\partial a^n$, $\partial D_{\mathrm{c}}/\partial b^n$, and $\partial D_{\mathrm{c}}/\partial \sigma^n$ for every iteration n. Let us denote by k the index of the point in F', which is closest in terms of Euclidean distance to the point i in F, and by A_{ik} the corresponding distance. Similarly, let us denote by m the index of the point in F, which is closest to the point j in F', and by A'_{mj} the corresponding distance. Then, from (11), we get

$$\begin{aligned}\frac{\partial D_{\mathrm{c}}}{\partial a^n} &= \frac{1}{2M}\sum_{i=1}^{M}\frac{\partial}{\partial a^n}A_{ik} + \frac{1}{2M'}\sum_{j=1}^{M'}\frac{\partial}{\partial a^n}A'_{mj}\\ &= \frac{1}{2M}\sum_{i=1}^{M}\frac{\partial}{\partial a^n}\left[(x'_k-\sigma^n x_i)^2+(y'_k-\sigma^n y_i)^2\right.\\ &\qquad\left.+(t'_k-a^n t_i+b^n)^2\right]^{\frac{1}{2}}\\ &\quad+\frac{1}{2M'}\sum_{j=1}^{M'}\frac{\partial}{\partial a^n}\left[\left(x_m-\frac{1}{\sigma^n}x'_j\right)^2+\left(y_m-\frac{1}{\sigma^n}y'_j\right)^2\right.\\ &\qquad\left.+\left(t_m-\frac{1}{a^n}t'_j-b^n\right)^2\right]^{\frac{1}{2}}.\end{aligned}$$

Therefore

$$\begin{aligned}\frac{\partial D_{\mathrm{c}}}{\partial a^n} &= \frac{1}{2M}\sum_{i=1}^{M}t_i\frac{a^n t_i-t'_k-b^n}{A_{ik}}\\ &\quad+\frac{1}{2M'}\sum_{j=1}^{M'}t'_j\frac{-\frac{1}{a^n}t'_j+t_m-b^n}{(a^n)^2A'_{mj}}.\end{aligned}\tag{17}$$

Similarly, for $\partial D_{\mathrm{c}}/\partial b^n$ and $\partial D_{\mathrm{c}}/\partial\sigma^n$, we get

$$\begin{aligned}\frac{\partial D_{\mathrm{c}}}{\partial b^n} &= -\frac{1}{2M}\sum_{i=1}^{M}\frac{a^n t_i-t'_k-b^n}{A_{ik}}\\ &\quad+\frac{1}{2M'}\sum_{j=1}^{M'}\frac{\frac{1}{a^n}t'_j-t_m+b^n}{A'_{mj}}\end{aligned}\tag{18}$$

$$\begin{aligned}\frac{\partial D_{\mathrm{c}}}{\partial \sigma^n} &= \frac{1}{2M}\sum_{i=1}^{M}\frac{\sigma^n\left(x_i^2+y_i^2\right)-x_ix'_k-y_iy'_k}{A_{ik}}\\ &\quad+\frac{1}{2M'}\sum_{j=1}^{M'}\frac{-\frac{1}{\sigma^n}\left(x_j^{2'}+y_j^{2'}\right)+x'_jx_m+y'_jy_m}{(\sigma^n)^2A'_{mj}}.\end{aligned}\tag{19}$$

By using (17)–(19), we determine the values of a, b, and σ in every iteration using the update rules given in (14)–(16). The iterative procedure stops when the values of a, b, and σ do not change significantly or after a fixed number of iterations.

B. RVM Classifier

Once a distance function is defined, a large number of pattern classification methods can be used for solving the L-class classification problem of classifying a data sample (i.e., a feature set F) in one of the L classes of human actions. In this paper, we use a kNN and an RVM classification scheme, where $k = 1$. Since the application of kNN is straightforward, we will only discuss the RVM classifier.

An RVM classifier is a probabilistic sparse kernel model identical in functional form to the SVM classifier. RVMs and SVMs have been used successfully in a large range of classification problems. In their simplest form, they attempt to find a hyperplane defined as a weighted (linear) combination of a few relevance (support) vectors that separate data samples of two different classes. In RVM, a Bayesian approach is adopted for learning, where a prior is introduced over the model weights, governed by a set of hyperparameters, one for each weight. The most probable values of these hyperparameters are iteratively estimated from the data. Sparsity is achieved because the posterior distributions of many of the weights are sharply peaked around 0. Unlike the support vector classifiers, the nonzero weights of RVM are not associated with examples close to the decision boundary, but rather appear to represent prototypical examples of classes. These examples are called relevance vectors and, in our case, they can be thought of as representative executions of a human action. The main advantage of RVM is that while it is capable of a generalization performance comparable to that of an equivalent SVM, it uses substantially fewer kernel functions. Furthermore, predictions in RVM are probabilistic, in contrast to the deterministic decisions provided by SVM. In their original form, RVMs are suitable for solving two-class classification problems.

In order to use RVMs in an L-class classification problem, we will train multiple (L) RVMs, each of which separates a class of human actions from all other classes of human actions. Given a data sample F, each of the L RVMs gives a probability that F belongs to each of the L classes. A data sample is classified to the class with the highest probability. In what follows, we will first briefly outline the use of the RVMs for a two-class classification problem and then we will formally define our classification scheme for the L-class classification problem.

Given a training dataset of N input-target pairs $\{(F_n, l_n), 1 \leq n \leq N\}$, an RVM learns the weights $w =< w_1, \ldots, w_n >$, such that the conditional probability $P(l|w, F)$ can be used for predicting the label l of a data sample F. Learning is performed using a maximum *a posteriori* estimation scheme where: 1) the conditional $P(l|w, F)$ is appropriately modeled and 2) a prior probability $p(w|a)$ ensures that the weight vector w is sparse.

More specifically, given a training dataset of N input-target pairs $\{(F_n, l_n), 1 \leq n \leq N\}$, an RVM learns functional mappings of the form

$$y(F) = \sum_{n=1}^{N} w_n K(F, F_n) + w_0 \tag{20}$$

where $\{w_n\}$ are the model weights and $K(\cdot,\cdot)$ is a kernel function, which in the case of RVM can be viewed as a basis function. Gaussian or radial basis functions have been extensively used as kernels in RVM and can be viewed as a similarity measure between F and F_n. In our case, we use as a kernel a Gaussian radial basis function defined by the distance function of (11). That is

$$K(F, F_n) = \mathrm{e}^{-\frac{D_{\mathrm{c}}(F,F_n)^2}{2\eta}} \tag{21}$$

where η is the kernel width. Using a Bayesian approach, it is assumed that the conditional probability $p(l|F)$ is Gaussian—that is, $N(l|y(F), \sigma^2)$. The mean of this distribution for a given F is modeled by $y(F)$, as defined in (20). For classification, we want to predict the posterior probability of class membership given the input F. The conditional probability $P(l_n|w, F_n)$ is given by

$$P(l_n|w, F_n) = \sigma\left\{y(F_n)\right\}^{l_n}\left[1-\sigma\left\{y(F_n)\right\}\right]^{1-l_n} \tag{22}$$

where $\sigma(y) = 1/(1+\mathrm{e}^{-y})$ is the logistic sigmoid function. Since maximum-likelihood estimation of the weights will lead to severe overfitting, a Gaussian prior is introduced over the weights

$$p(w|a) = \prod_{i=1}^{N} N\left(w_i|0, a_i^{-1}\right) \tag{23}$$

where a_i is an individual hyperparameter for every weight, leading to an α vector of N hyperparameters. In order to estimate the weights, an

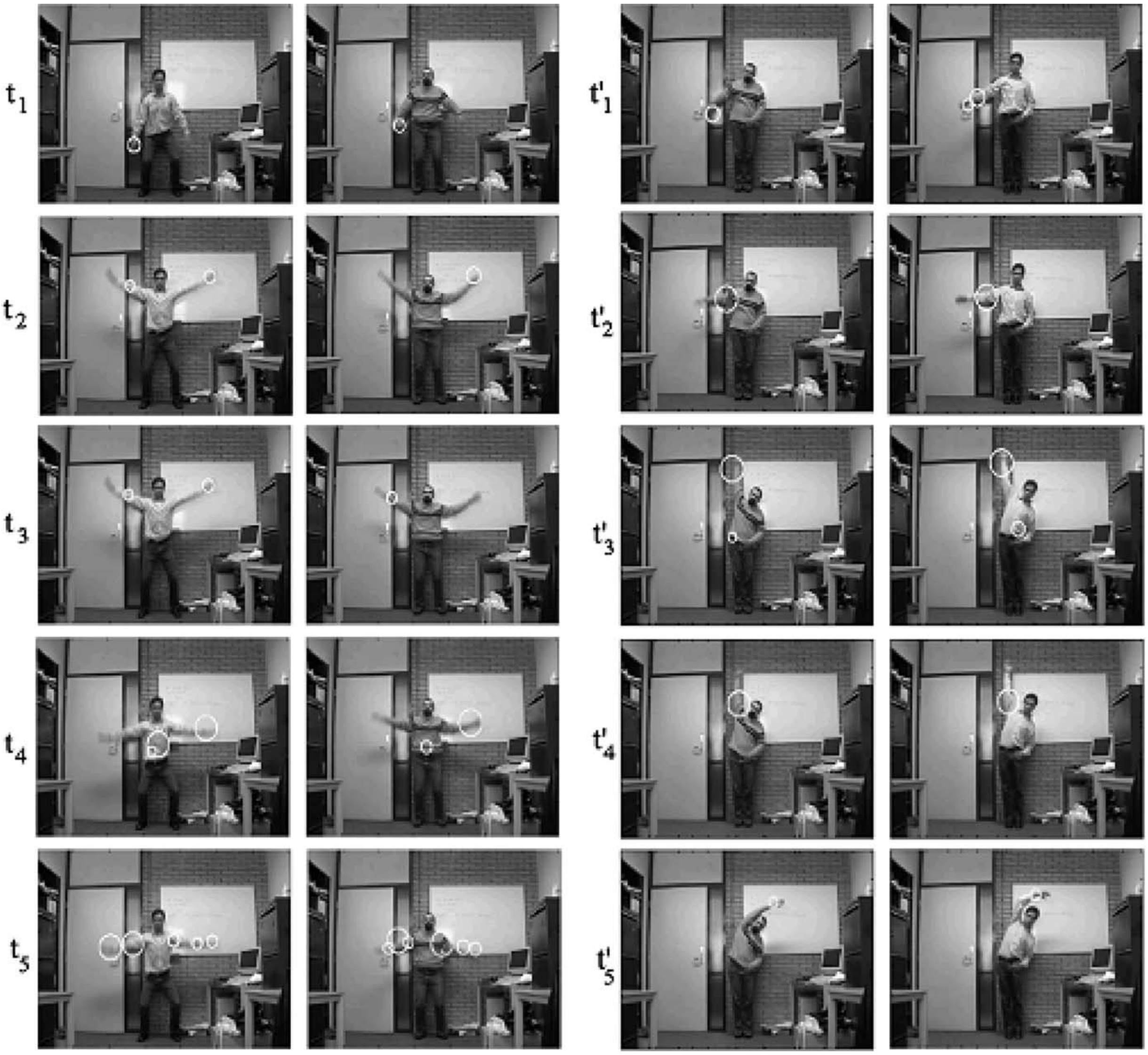

Fig. 2. Detected spatiotemporal features in four sample image sequences, corresponding to two action classes, for five time instances, t_i, t_i', $i = 1, \ldots, 5$. For each class, the detected regions are drawn for two different subjects performing the action.

iterative procedure is utilized, and a Gaussian approximation over the posterior of the weights is calculated. From that, the hyperparameters are updated and the process is repeated until the change in the hyperparameter values is minimal or when a maximum number of iterations has been reached. A detailed description of the training process of an RVM classifier can be found in [18].

In the classification phase, for the two-class problem, a sample F is classified to the class $l \in [0, 1]$, which maximizes the conditional probability $p(l|F)$. In order to use RVM classifiers for multiclass problems, one classifier is trained for each separate class. For L different classes, L different classifiers are trained and a given example F is classified to the class for which the conditional distribution $p_i(l|F)$, $1 \leq i \leq L$ is maximized, that is

$$\text{Class}(F) = \arg\max_i \left(p_i(l|F)\right). \tag{24}$$

IV. Experimental Results

Similar to Bobick and Davis [1], we use aerobic exercises as a test domain to evaluate the proposed method. We have created our own set of examples, consisting of 19 different aerobic exercises, performed by amateurs wearing everyday clothes, which have seen a video with an instructor performing the same set of exercises. Each exercise is performed twice by four different subjects, leading to a set of 152 corresponding feature sets. In the experiments that follow, we detect salient points using a full-search approach—that is, by applying cylindrical neighborhoods for every single point in the input image sequences. Further on in the same section, we will also present results for the two-step approximation procedure presented in Section II-B.

In order to illustrate the ability of the proposed method to consistently detect spatiotemporal events, we present in Fig. 2 the salient regions detected in five instances of four sample image sequences. The first two columns depict two executions of the same exercise by two different subjects while the last two columns depict the execution of another exercise by another pair of subjects. It is apparent that there is consistency in the location and scale of the detected spatiotemporal salient regions between different executions of the same exercise. The detected salient points seem to appear in areas with significant amount of activity, such as the points in space and time at which the hands move fast. Since we use as input signal the convolution of the image sequence with a first-order Gaussian derivative filter, some of the detected points are located on the edges of moving objects rather

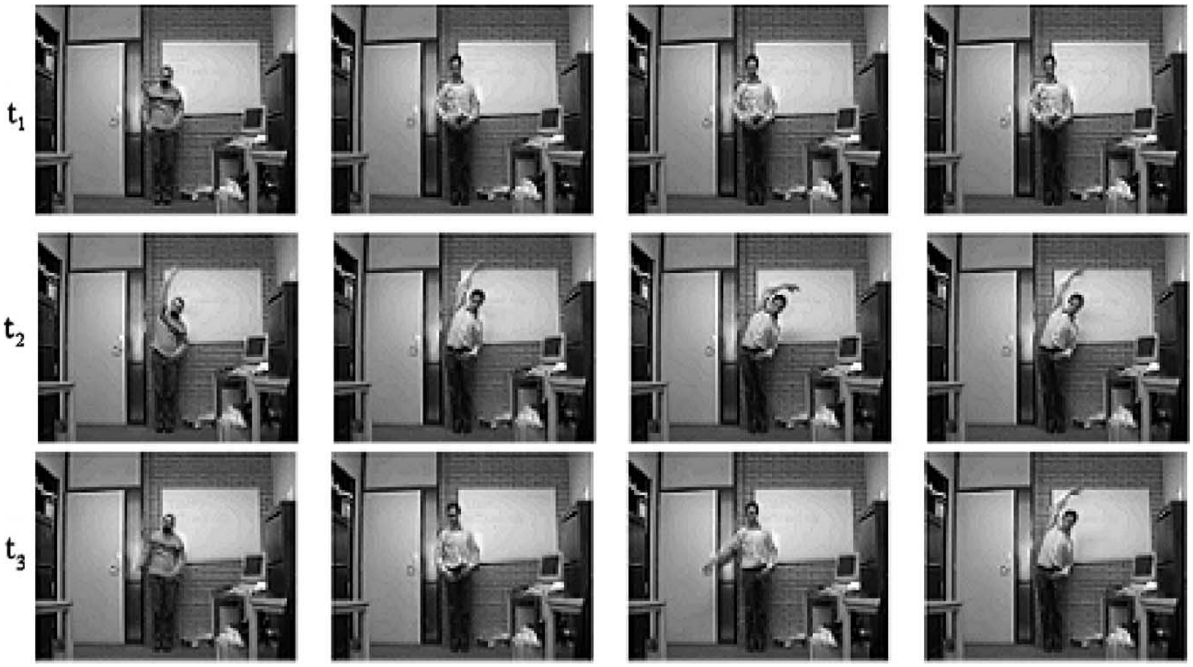

Fig. 3. Effect of time warping. (First column) Reference sequence. (Second column) Space–time-warped sequence. (Third column) Stretched sequence. (Fourth column) Original sequence.

than on the objects themselves (e.g., at instance t'_1 of the second pair of sequences). Moreover, there seems to be a correlation between the scale of the detected regions and the motion magnitude—that is, the scale of the detected regions is large when the motion is fast (instances $t_4, t_5, t'_2, t'_3, t'_4$), and smaller when the motion is slower $(t_1, t_2, t_3, t'_1, t'_5)$. This can be explained by the fact that when the motion is fast, the activity spreads over a larger spatial region than when the motion is slow. Finally, let us note that the algorithm does not guarantee that the detection of corresponding regions across the examples will occur at the same time instance. For example, at the time instances t_2 and t_3 of the first pair of image sequences of Fig. 2 (i.e., first two columns), the detection of the arms does not occur at the same, but at neighboring time instances. Note that the image sequences that are presented in Fig. 2 are time-warped pairwise.

In order to test the influence of our space–time-warping algorithm, and consequently, the robustness of our method with respect to the scale, we randomly selected one example per class from our original example set and we resized it to 1.2 and 1.5 times its initial size. We applied in each of these sequences the spatiotemporal salient-point detector of Section II and we used the resulting representations in order to warp them in space and time with the original sequences. The result for a single pair of original-resized sequences is shown in Fig. 3, where in the first column is the original and in the second column is the space–time-warped sequence. We also stretched the latter sequence in time, so that its duration matches that of the original one. The result is shown in the third column of Fig. 3. From the figure, it is clear that the space–time-warped sequence is closer to the original one, indicating that our proposed algorithm effectively warps a sequence in space and time with another, by using just the spatiotemporal salient features detected in both of them. The σ parameter for the resized sequence was calculated equal to 1.18, which is very close to the actual value of 1.2.

In order to test the efficiency of the proposed method towards recognition, we applied a simple kNN classifier and a RVM classifier to the available feature sets. We performed our experiments in the leave-one-subject-out manner. That is, in order to classify a test exercise performed by a specific test subject, we trained our classifiers using all available data except for those belonging to the same class and performed by the same subject as the test exercise.

For the kNN classifier, the label assigned to each test example was the label of the feature set belonging to the training set with the smallest resulting chamfer distance. In this way, we constructed Table I, which gives the recall and precision rates for every class. As can be seen from the table, for many classes, the recognition rate is higher than 80%, while for some classes, all the examples were correctly classified. For other classes, however, the recall and precision rates are lower (e.g., classes 7 and 13). An examination of the corresponding image sequences reveals that there is very little difference between the kind of motion performed in them, and therefore, in their resulting spatiotemporal representations, as can be seen from Fig. 4. The main difference in the sequences shown in the figure is that in one of the cases, the torso of the subject remains in the upright position throughout the execution of the exercise, while in the other, the subject bends a little in the front. Since there is only one camera placed in front of the subject, this difference cannot be depicted by the representation, and the algorithm, therefore, cannot make a clear distinction between them. The overall calculated recall rate for the kNN classifier was 74.34%.

In order to classify a test example using the RVMs, we constructed 19 different classifiers, one for each class, and we calculated for each test example F the conditional probability $p_i(l|F)$, $1 \leq i \leq 19$. Each example was assigned to the class for which the corresponding classifier provided the maximum conditional probability, as depicted in (24). Note that for estimating each of the $p_i(l|F)$, an RVM is trained by leaving out the example F as well as all other instances of the same exercise that were performed by the subject from F. The corresponding recall and precision rates are also given in Table I, where an improvement in their values is visible for some classes. However, there is still confusion between classes (e.g., classes 7 and 13), which the kNN classifier also mixes up. This is due to the minor differences in the corresponding representations, as can be seen from Fig. 4. In Table II, the confusion matrix generated by the RVM

TABLE I
RECALL AND PRECISION RATES FOR THE kNN AND RVM CLASSIFIERS

Approach	Class Labels	1	2	3	4	5	6	7	8	9	10
Full Search	kNN Recall/Precision	1 / 1	1 / 1	1 / 1	1 / 1	0.38 / 0.5	0.25 / 0.67	0.63 / 0.45	1 / 1	0.75 / 1	1 / 0.8
	RVM Recall/Precision	1 / 1	1 / 1	1 / 1	0.75 / 1	0.38 / 0.6	0.25 / 0.5	0.63 / 0.63	0.75 / 1	0.75 / 1	1 / 1
2-step	RVM Recall/Precision	1 / 1	1 / 1	1 / 0.89	0.75 / 0.75	0.38 / 0.5	0.5 / 0.44	0.63 / 0.71	0.75 / 0.75	0.88 / 0.78	1 / 1

Approach	Class Labels	11	12	13	14	15	16	17	18	19	Total
Full Search	kNN Recall/Precision	0.75 / 1	0.75 / 0.75	0.38 / 0.21	1 / 1	0.5 / 0.29	0 / 0	0.75 / 1	1 / 1	1 / 1	0.7434
	RVM Recall/Precision	0.88 / 1	0.5 / 0.57	0.75 / 0.35	1 / 0.89	0.88 / 0.64	0.5 / 0.5	1 / 0.89	0.88 / 0.88	0.88 / 0.78	0.7763
2-step	RVM Recall/Precision	1 / 1	0.63 / 0.63	0.25 / 0.25	0.75 / 0.86	0.75 / 0.46	0.38 / 0.5	0.75 / 0.86	1 / 1	0.63 / 0.71	0.7368

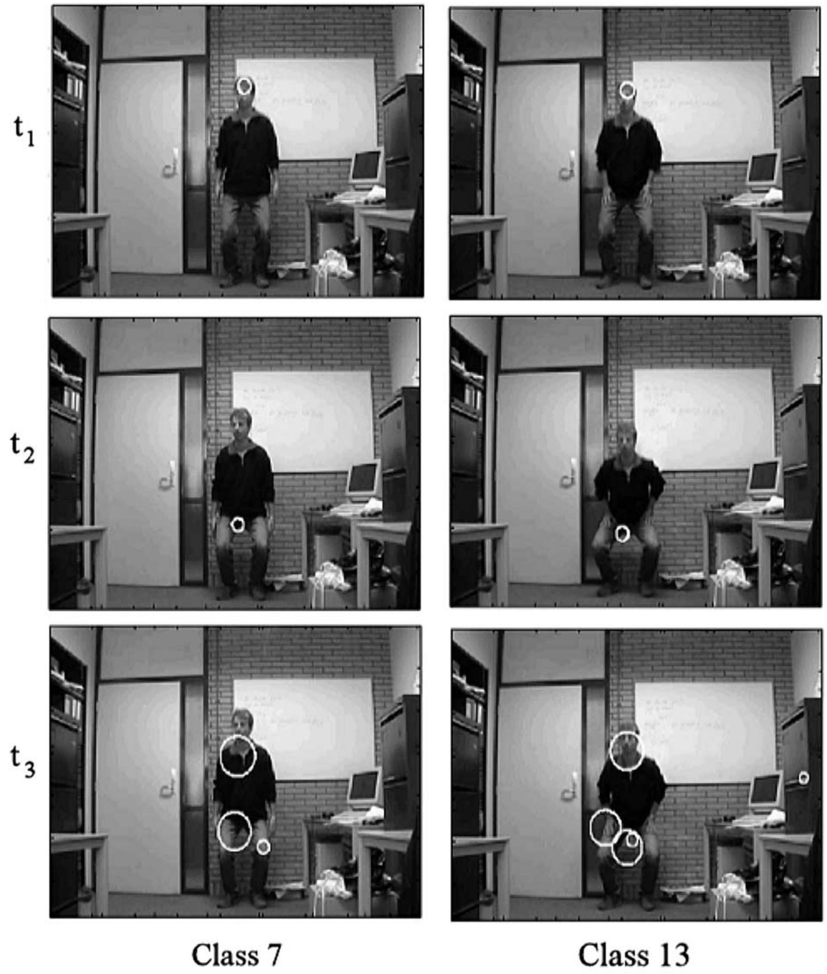

Fig. 4. Detected spatiotemporal features in two misclassified image sequences for three time instances, t_i, $i = 1, \ldots, 3$. The sequences correspond to two different action classes that are performed by the same subject.

classifier is given. It is obvious from the table that there are mutual confusions between specific classes, for instance, classes 5, 6, 7, 12, and 13. As mentioned earlier, the reason for some of these confusions lies to the inadequacy of a single camera to capture the valuable depth information needed in order to discriminate the classes in question. The global recall rate for the RVM classifier was 77.63%, which is a relatively good performance, given the small number of examples with respect to the number of classes, and the fact that the subjects were not trained.

The confusion matrix of Table II conceals the fact that, for some of the misclassified examples, the correct matching move might be very close in terms of distance to the closest move selected. We used the average ranking percentile in order to extract this kind of information and to measure the overall matching quality of our proposed algorithm. Let us denote with r^{F_n} the position of the correct match for the test example $F_n, n = 1 \ldots N_2$, in the ordered list of N_1 match values. Rank r^{F_n} ranges from $r = 1$ for a perfect match to $r = N_1$ for the worst possible match. Then, the average ranking percentile is calculated as follows:

$$\overline{r} = \left(\frac{1}{N_2} \sum_{i=1}^{N_2} \frac{N_1 - r^{F_n}}{N_1 - 1} \right) 100\%. \qquad (25)$$

Since our dataset consists of 152 test image sequences divided in 19 separate classes, it follows that $N_1 = 19$ and $N_2 = 152$. Each of the 19 match values are provided for each example by the 19 trained RVM

TABLE II
RVM CONFUSION MATRIX

Class labels	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	Total
1	**8**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
2	0	**8**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
3	0	0	**8**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
4	0	0	0	**6**	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	8
5	0	0	0	0	**3**	2	0	0	0	0	0	0	3	0	0	0	0	0	0	8
6	0	0	0	0	2	**2**	2	0	0	0	0	0	2	0	0	0	0	0	0	8
7	0	0	0	0	0	0	**5**	0	0	0	0	0	1	0	0	1	0	0	1	8
8	0	0	0	0	0	0	0	**6**	0	0	0	2	0	0	0	0	0	0	0	8
9	0	0	0	0	0	0	0	0	**6**	0	0	0	0	0	0	1	0	1	0	8
10	0	0	0	0	0	0	0	0	0	**8**	0	0	0	0	0	0	0	0	0	8
11	0	0	0	0	0	0	0	0	0	0	**7**	0	1	0	0	0	0	0	0	8
12	0	0	0	0	0	0	0	0	0	0	0	**4**	4	0	0	0	0	0	0	8
13	0	0	0	0	0	0	1	0	0	0	0	1	**6**	0	0	0	0	0	0	8
14	0	0	0	0	0	0	0	0	0	0	0	0	0	**8**	0	0	0	0	0	8
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**7**	1	0	0	0	8
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	**4**	0	0	0	8
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**8**	0	0	8
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**7**	1	8
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	**7**	8
Total	8	8	8	6	5	4	8	6	6	8	7	7	17	9	11	8	9	8	9	152

classifiers. The average ranking percentile for the RVM classifier was 97.25%. Its high value shows that, for the majority of the misclassified examples, the correct matches are located in the first positions in the ordered list of match values.

In order to verify the robustness of our method to scaling, we performed the same classification experiments as before, using resized versions of the original image sequences. More specifically, we randomly selected one example per class and we resized it to 1.2 and 1.5 times its initial size. Classification was performed by considering each resized example as the test set and the entire initial set of examples as the training set, except for those belonging to the same class and performed by the same subject as the test example. For the resized-by-1.2 set, 14 out of 19 examples were correctly classified with the kNN classifier, and 13 out of 19 with the RVM classifier, while for the resized-by-1.5 set, 13 out of 19 examples were correctly classified by both classifiers.

The clustering process of Section II-B clusters the detected salient points into salient regions by selecting the point with the highest saliency value in the set as the starting point. In order to examine the sensitivity of our method with respect to the estimates of the saliency values, we performed the same classification experiments as before, but with noisy versions of the original unclustered representations. More specifically, we added to the saliency values of the detected salient points Gaussian noise of zero mean and variance σ. The resulting representations were clustered once again using the process of Section II-B, and the same classification experiments as before were performed. Salient points whose saliency was less than 0 after the noise addition were not taken into account during the clustering process.

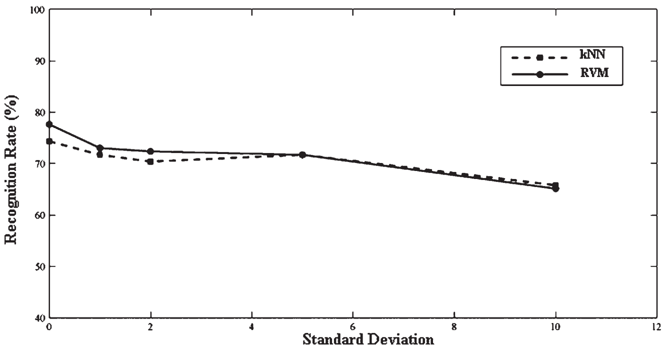

Fig. 5. Overall recognition rate, with respect to the sigma of the Gaussian noise, that was added to the saliency values of the detected salient points prior to clustering. (Dashed line) kNN. (Solid line) RVM.

The overall recognition rate that was achieved, for five levels of noise of increasing variance, is plotted in Fig. 5. From the figure, we conclude that the saliency values of the detected salient points carry important information, since the performance deteriorates as the noise increases. However, the deterioration is not very large, considering the amount of added noise.

Finally, we compared our method with the work of Bobick and Davis [1] on temporal templates. In [1], each single-view test example was matched against seven views of each example in the training set, which in turn, was performed several times by an experienced aerobics instructor. A performance of 66.67% (12 out of 18 moves) was achieved. Our training set, however, consists of single-view examples, performed several times by nonexpert subjects. Furthermore, noise and shadow effects in the sequences of our dataset create small nonzero pixel regions in areas of the corresponding MEIs and MHIs where no motion exists. The overall recognition rate that was achieved was 46.71%. Removal of most of the spurious areas with different simple morphological operations (removal of small connected components) led to deterioration in the overall performance.

We also present in Table I the RVM classification results for the two-step approximation of a full search, presented in Section II-B. As can be seen, the recall and precision rates are lower than the ones corresponding to the full-search approach, leading to an overall recognition rate of 73.68%. However, the reduction is low, and therefore, it remains a good alternative to the full-search method when faster recognition is required.

V. Conclusion and Future Research

In this paper, we extended the concept of saliency from the spatial to the spatiotemporal domain, in order to represent human motion by using a sparse set of spatiotemporal features that, loosely speaking, correspond to activity peaks. We did this by measuring changes in the information content of neighboring pixels, not only in space but also in time. We devised an appropriate distance measure between sparse representations containing different numbers of features, based on the chamfer distance. The proposed distance measure allows us to use an advanced kernel-based classification scheme, the RVMs. We devised an iterative space–time-warping technique that aligns in time the representations and achieves invariance against scaling, while translation invariance is achieved through transformation of the features' location so that they have zero mean. We have illustrated the efficiency of our representation in recognizing human actions using as a test domain aerobic exercises. We presented results on real image sequences that illustrate the consistency in the spatiotemporal localization and scale selection of the proposed method. Classification results are presented for two different types of classifiers, displaying the efficiency of the representation in discriminating actions of different motion classes. Furthermore, the classification results clearly illustrate the superiority of the proposed kernel-based classification scheme over the simple kNN classification.

In future research, we wish to increase the discriminating power by investigating the extraction of spatiotemporal features around the spatiotemporal salient points. This is a natural extension of similar methods that extract texture features around the detected points in the spatial domain. Making the method robust to rotation is also an important issue, which can be potentially achieved by introducing an additional parameter for rotation in the proposed space–time-warping technique. Furthermore, our clustering results are affected by the order of detection of new clusters, especially in the case where two or more highly salient points are in nearby locations. By examining more sophisticated clustering algorithms, it is possible to remedy this and potentially enhance the efficiency of the representation. Finally, the recognition rate can be potentially increased by using more advanced classification schemes.

Acknowledgment

The work presented in this paper was conducted at the Delft University of Technology. The data set was collected while I. Patras was with the ISIS group at the University of Amsterdam.

References

[1] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[2] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 849–865, Nov. 1988.

[3] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition*, Washington, DC, May 2002, pp. 405–410.

[4] S. Gilles, "Robust description and matching of images," Ph.D. thesis, Dept. Eng. Sci., Univ. Oxford, Oxford, U.K., 1998.

[5] R. Haralick and L. Shapiro, *Computer and Robot Vision II*. Reading, MA: Addison-Wesley, 1993.

[6] J. S. Hare and P. H. Lewis, "Salient regions for query by image content," in *Proc. Int. Conf. Image and Video Retrieval*, Dublin, Ireland, Jul. 2004, pp. 317–325.

[7] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," U.K.*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998, Surrey.

[8] T. Kadir and M. Brady, "Scale saliency: A novel approach to salient feature and scale selection," in *Proc. Int. Conf. Visual Information Engineering*, Surrey, U.K., Nov. 2000, pp. 25–28.

[9] J. J. Koenderink and A. J. van Doorn, "Representation of local geometry in the visual system," *Biol. Cybern.*, vol. 55, no. 6, pp. 367–375, Mar. 1987.

[10] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Nice, France, Oct. 2003, pp. 432–439.

[11] T. Lindeberg, "Feature detection with automatic scale selection," *Int. J. Comput. Vis.*, vol. 30, no. 2, pp. 77–116, Nov. 1998.

[12] E. Loupias, N. Sebe, S. Bres, and J.-M. Jolion, "Wavelet-based salient points for image retrieval," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, BC, Canada, Sep. 2000, vol. 2, pp. 518–521.

[13] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Computer Vision*, Kerkyra, Greece, Sep. 1999, vol. 2, pp. 1150–1157.

[14] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 530–535, May 1997.

[15] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 151–172, Jun. 2000.

[16] N. Sebe and M. S. Lew, "Comparing salient point detectors," *Pattern Recognit. Lett.*, vol. 24, no. 1–3, pp. 89–96, Jan. 2003.

[17] B. M. ter Haar Romeny, L. M. J. Florack, A. H. Salden, and M. A. Viergever, "Higher order differential structure of images," *Image Vis. Comput.*, vol. 12, no. 6, pp. 317–325, Jul./Aug. 1994.

[18] M. E. Tipping, "The relevance vector machine," in *Advances Neural Information Processing Systems*, Denver, CO, Sep. 1999, pp. 652–658.