Human Gesture Recognition using Sparse B-spline Polynomial Representations

A. Oikonomopoulos [a] M. Pantic [a] I. Patras [b]

[a] *Computing Department, Imperial College London*
[b] *Electronic Engineering Department, Queen Mary University of London*

Abstract

The extraction and quantization of local image and video descriptors for the subsequent creation of visual codebooks is a technique that has proved extremely effective for image and video retrieval applications. In this paper we build on this concept and extract a new set of visual descriptors that are derived from spatiotemporal salient points detected on given image sequences and provide local space-time description of the visual activity. The proposed descriptors are based on the geometrical properties of three-dimensional piecewise polynomials, namely B-splines, that are fitted on the spatiotemporal locations of the salient points that are engulfed within a given spatiotemporal neighborhood. Our descriptors are inherently translation invariant, while the use of the scales of the salient points for the definition of the neighborhood dimensions ensures space-time scaling invariance. Subsequently, a clustering algorithm is used in order to cluster our descriptors across the whole dataset and create a codebook of visual verbs, where each verb corresponds to a cluster center. We use the resulting codebook in a 'bag of verbs' approach in order to recover the pose and short-term motion of subjects at a short set of successive frames, and we use Dynamic Time Warping (DTW) in order to align the sequences in our dataset and structure in time the recovered poses. We define a kernel based on the similarity measure provided by the DTW in order to classify our examples in a Relevance Vector Machine classification scheme. We present results from two different databases of human actions that verify the effectiveness of our method.

1 Introduction

Vision-based analysis of human motion is nowadays one of the most active fields of computer vision, due to its practical importance for a wide range of vision-related applications, like video retrieval, surveillance, vision-based interfaces and Human-Computer Interaction. From any given video sequence humans are usually able to deduce information about its content quickly and easily. When it comes to computers, however, robust action recognition still remains a very challenging task, evident from the abundance of different motion analysis approaches that have been developed [13].

Typically, activity recognition systems can be divided into two main categories. The first concerns tracking of body parts and the subsequent use of the resulting trajectories for recognition [17], [5]. These approaches, however, are highly dependent on the type of tracking system that is used and its target application. In addition, due to the deformable nature and articulated structure of the human body, these methods suffer from problems like accurate initialization, occlusion and high dimensionality.

A second category of systems uses sets of spatiotemporal feature descriptors in order to represent human body motion. The concept of spatiotemporal feature extraction for activity recognition stems from the domain of object recognition, where static features have been successfully used for the detection of various objects from images [8], [1]. In [7], a Harris corner detector is extended in the temporal domain, 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. In [3], human actions are treated as three-dimensional shapes in the space-time volume. The method utilizes properties of the solution to the Poisson equation to extract space-time features of the moving human body, such as local space-time saliency, action dynamics, shape structure and orientation. In [14] a local self-similarity descriptor is extracted in order to match areas in images or videos that share similar geometric properties.

Finally, in [6] a set of spatiotemporal features inspired from the human visual cortex, called C features, are extracted for the recognition of human and animal motions. The method works in an hierarchical way and the obtained features are invariant to scale changes in space and time.

Recently a number of works used visual codebooks in order to detect and recognize objects and/or humans. The visual codebook creation is performed by grouping the extracted feature descriptors in the training set using, for instance, a clustering algorithm [10]. The resulting centers are then considered to be codewords and the whole set of codewords forms a 'codebook'. In a 'bag of words' approach each instance is represented as a histogram of codewords, and recognition is performed by histogram comparison. In [2] a set of SIFT-like features are hierarchically used in order to form 'hyperfeatures' for the purpose of object recognition, while in [4] static and dynamic features based respectively on gradients and optical flow are extracted in order to detect humans in image sequences. In order to further enhance the performance of these models, several researchers have gone one step forward and encoded the spatial relationships that exist between the features. In [9], extracted features cast votes towards the center of the object from which they are extracted. In this way the system implicitly encodes the spatial relationships between the extracted features. In [15] a similar enhancement takes place by considering pairs of visual words which co-occur within local spatial neighborhoods, denoted as 'doublets'. In [11] constellations of static and dynamic bags of features are modeled in order to recognize human activities. Finally in [16], SIFT descriptors are extracted from spatial video patches and their spatial layout is encoded for the purpose of video or image retrieval.

In this paper we extract a new set of visual descriptors that are derived from the spatiotemporal salient points of [12]. At each salient point location we define a spatiotemporal neighborhood with dimensions proportional to the detected space-time scale of the point. We use the locations of the salient points that are engulfed within this neighborhood in order to approximate a three dimensional piecewise polynomial, namely a B-spline. Our descriptors are subsequently derived from the geometrical properties of each polynomial as these are captured in their partial derivatives of different orders. At the next step, the whole set of descriptors is accumulated into a number of histograms, depending on the number of parameters that describe the spline and the maximum degree of its derivatives. Since our descriptors correspond to geometrical properties of the spline, they are translation invariant. Furthermore, the use of the automatically detected space-time scales of the salient points for the definition of the neighborhood ensures invariance in space and time. Similar to other approaches, where a codebook of visual words is created from appearance descriptors, we create a codebook of visual verbs by clustering our motion descriptors across the whole dataset. We use the resulting codebook in a 'bag of verbs' approach in order to recover the pose and instantaneous motion of subjects at a short set of successive frames and we use a Dynamic Time Warping scheme (DTW) in order to structure in time the recovered poses. We use the similarity measure between the examples, provided by the DTW, in order to define a kernel for a classifier based on Relevance Vector Machines (RVM). We present results in two different databases of human actions that verify the effectiveness of our method. Finally, we perform experiments in order to verify the generality of our descriptors, that is, their ability to encode and discriminate between unseen classes.

One of the main contributions of the method presented in this paper is the sparsity of the extracted descriptors, since they are extracted at spatiotemporal regions that are detected at sparse locations within the image sequence. This is contrary to the work of Blank et al [3], where a whole image sequence is represented as a space-time shape. Furthermore, the use of DTW adds structure to the recovered short-term motions of the subjects, as opposed to [3], [6], where features are matched based on the maximum similarity accross a whole video sequence. Our results are comparable [3], [6] or show improvement [11] with state of the art methodologies for the same sequences.

The remainder of the paper is organized as follows: in section 2 we describe our feature extraction process, including our B-spline approximation and the creation of our codebook. In section 3 we present our recognition method, that includes the DTW and RVM steps. In section 4 we present our experimental results and finally, in section 5 our final conclusions are drawn.

2 Representation

In this section we introduce the visual descriptors that we use in order to represent an image sequence. We will initially give some basics on B-splines and we will subsequently describe their usage in extracting local spatiotemporal image sequence descriptors. Finally, we will briefly explain the process that we followed in order to create a codebook from these descriptors.

2.1 B-spline Surfaces

Let us define an $M \times N$ grid of control points $\{P_{ij}\}$, $i = 1 \dots M$ and $j = 1 \dots N$. Let us also define a knot vector of h knots in the u direction , $U = \{u_1, u_2, \dots, u_h\}$ and a knot vector of k knots in the v direction, $V = \{v_1, v_2, \dots, v_k\}$. A B-spline surface of degrees p,q in the u and v directions respectively is given by:

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

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

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

The set of control points is referred to as the control net, while the range of the knots is usually $[0, 1]$. For this work we assume 3^{rd} degree polynomials, that is, $p = q = 3$.

2.2 Spatiotemporal Descriptors

In order to approximate a B-spline polynomial we need to initially define its control net, that is, P_{ij}. Formally, for each salient point location we want to approximate a polynomial having as control net the points within a small neighborhood O around the point in question. For a good approximation, however, ordering of the control points in terms of their spatiotemporal location is an important factor in order to avoid loops. In order to make this more clear, let us consider a set of points $L = \{l_i\}$ sampled uniformly from a circular curve. In order for a polynomial to approximate the circular curve from the L points, these points should be given in sequence, that is, $L = \{l_1, l_2, \dots, l_n\}$. If this is not the case, then the polynomial will attempt to cross the points in a different order, creating unwanted loops. Furthermore, it is clear that any points enclosed by the circle will also degrade the approximation and should not be accounted for. In order to overcome these problems, we perform two preprocessing steps on the set S: In the first step, we eliminate points that are enclosed within the closed surface defined by the boundary. In our implementation, a point lies in the boundary if it lacks any neighbors within a circular slice shaped neighborhood of radius r, minimum angle a and having the point as origin. For our implementation we selected a radius of 10 pixels and an angle of 70 degrees. In the second step, we order the selected boundary points. We do this by randomly selecting a point on the boundary as a seed and by applying an iterative recursive procedure that matches the seed point with its nearest neighbor in terms of Euclidean distance. This process repeats itself having as seed the nearest neighbor selected until there are no nearest neighbors left, that is, either an edge has been reached or all points have been accessed. One could argue that the procedure described above would select points in the convex hull of the motion, creating problems in the case of non-stationary background or if there are more than one subjects performing activities in the same scene. This however, is not true, as the whole procedure is performed locally. In effect, the amount of locality is determined by the radius r.

Let us denote by $S' = \{(\vec{s}'_i, \vec{c}'_i, y'_{D,i})\}$ the set of spatiotemporal salient points located on the motion boundary, obtained from the procedure of the previous section. For each salient point position within S' we define a spatiotemporal neighborhood N of dimensions proportional to $\vec{s}'_i$. Let us denote by O' the set of points in N. Then, for each N, we approximate a B-spline polynomial as in eq. 1. The grid of control points P_{ij} in eq. 1 corresponds to the set O', that is, each P_{ij} is a point in space-time. We should note that the grid is not and does not need to be uniform, that is, the pairwise distances of the control points can be different. The knot vectors U and V are a parameterization of the B-spline, and essentially encode the way the B-spline surface changes with respect to its control points. More specifically, the knot vector U encodes the way the x coordinates change with respect to y, while the knot vector V encodes the way both x and y change with respect to time t. Using this process, any given image sequence is represented as a collection of B-spline surfaces, denoted as $\{F_i(u,v)\}$. The number of surfaces per sequence depends on the number of points in S', since we fit one surface per salient point position. An example of a spline fitted to a set of points is presented in Fig. 1. Each member of the set $\{F_i(u,v)\}$ is essentially a piecewise polynomial in a three dimensional space. This means that we can fully describe its characteristics by means of its partial derivatives with respect to its parameters u, v. That is, for a grid of knots of dimensions $k \times h$ we calculate the following matrix R_i of dimensions $(pq-1) \times (hk)$:

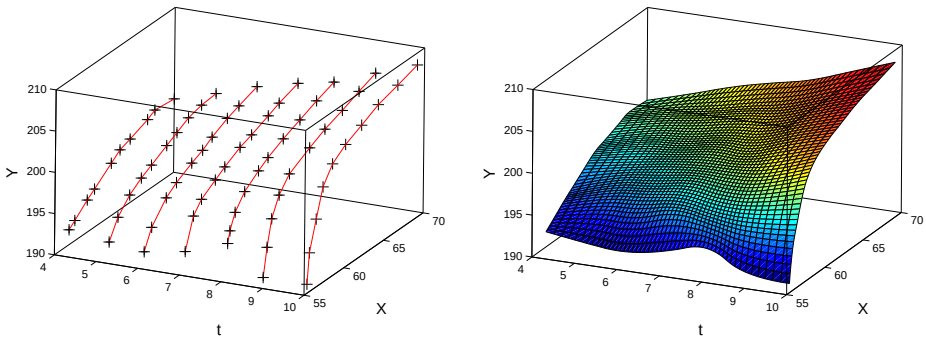

Figure 1: (a) A set of points within a spatiotemporal neighborhood N and (b) their B-spline approximation

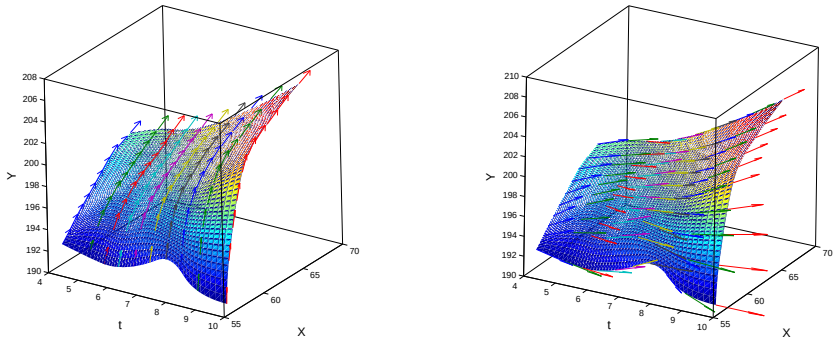

Figure 2: First derivatives with respect to (a) u and (b) v, drawn as three dimensional vectors

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

where $\partial^p/\partial u^p$ is the $p-th$ partial derivative with respect to u. From eq. 1 it is apparent that for specific values of u, v, $F_i(u, v)$ is a point in space-time, that is, a 3×1 vector. Consequently, each element of R_i is a vector of the same dimensions. In Fig. 2 an illustration of the first derivatives with respect to u and v is given. The derivatives are drawn as three dimensional vectors, superimposed on the spline from which they were extracted.

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

2.3 Codebook Creation

In order to create a codebook, applying a clustering algorithm to the whole set of descriptors is time and memory consuming. According to the authors of [9], the way a vocabulary is constructed has little impact to the final classification results. We therefore follow their approach and randomly subsample our descriptor set. Subsequently, we cluster our randomly selected features using K-means clustering. The resulting cluster centers are the codewords and the whole set of codewords constitutes the codebook. For this work we used a total number of 1000 clusters, as a compromise between representation accuracy and speed.

3 Classification

Having constructed our codebook, our goal is to be able to represent and classify any test image sequence to one of the available classes in our training set. A conventional application of a 'bag of verbs' approach would dictate that each image sequence in the dataset is represented as a histogram of visual codewords drawn from the codebook. Using the codebook in this way for our specific set of descriptors resulted in recognition rate of about 60%, using a 1-NN classifier based on the χ^2 distance between the histograms of the test and training sequences. We follow instead a different approach and use the codebook in order to recover the pose and instantaneous motion of the subjects performing the actions at a short set of successive frames. By doing this, we essentially encode each video as a collection of instantaneous motions.

As we will show in the experimental results section, even though pose recovery and subsequent classification using just a chamfer distance based nearest neighbor approach works quite well, this is not sufficient, as we would like to be able to add some structure and order in the instantaneous motions that are being recovered. A possible solution would be to use a temporal model like a Hidden Markov Model in order to encode the temporal relationships between the poses. This solution however is not practical, as the high dimensionality of the codebook would make the training of such a model cumbersome, especially in estimating the emission probabilities of the model. The use of a classification method that would be able to automatically provide these probabilities is not very practical either, as this would require manual annotation of similar poses between different examples of the same class. In order to deal with these issues, we decided to use Dynamic Time Warping (DTW) to align our sequences and subsequently apply a discriminant classifier like a Relevance Vector Machine (RVM) [18] for classification.

3.1 Dynamic Time Warping

Dynamic Time Warping (DTW) is a well established technique for aligning any two sequences. The sequences are "warped" non-linearly in time in order to determine a measure of their similarity independent of certain non-linear variations in the time dimension. In order to use DTW for our problem, we consider as a sequence the series of the recovered instantaneous motions of each example, each being represented as a histogram of codewords. Since we are dealing with histograms, a suitable distance metric to use would be the χ^2 distance. Using this distance, we align our test sequences with every sequence in our training set. This procedure results in a similarity measure between the testing and training sequences, which is subsequently used in an RVM classification step.

3.2 Relevance Vector Machine

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

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

where $\{w_n\}$ are the model weights and $K(.,.)$ is a Kernel function. For our work, we use the similarity measure provided by the DTW of the previous section in order to define a kernel for the RVM. More specifically, we apply the logistic sigmoid function to the DTW similarity measure in order to obtain a distance measure instead. Subsequently, we use a Gaussian RBF to define the kernel, that is,

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

where D is the logistic sigmoid function of the DTW similarity measure and η is the width of the kernel. In the two class problem, a sample F is classified to the class $l \in [0, 1]$ that maximizes the conditional probability $p(l|F)$. For L different classes, L different classifiers are trained and a given example F is classified to the class for which the conditional distribution $p_i(l|F)$,$1 \leq i \leq L$ is maximized:

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

Class	R/P (NN)	R/P (DTW)	R/P (RVM)	Confusion Matrix								
bend	1.0/1.0	0.88/1.0	1.0/0.9	**9**	0	0	0	0	0	0	1	0
jack	1.0/0.9	1.0/1.0	1.0/1.0	0	**9**	0	0	0	0	0	0	0
jump	0.67/0.67	0.56/1.0	0.78/0.88	0	0	**7**	0	0	1	0	0	0
pjump	0.89/1.0	1.0/1.0	1.0/1.0	0	0	0	**9**	0	0	0	0	0
run	1.0/0.71	1.0/0.56	1.0/0.9	0	0	1	0	**10**	0	0	0	0
side	0.89/1.0	0.89/1.0	0.78/1.0	0	0	0	0	0	**7**	0	0	0
walk	0.5/0.71	1.0/0.83	1.0/0.83	0	0	1	0	0	1	**10**	0	0
wave1	1.0/1.0	0.78/1.0	0.78/1.0	0	0	0	0	0	0	0	**7**	0
wave2	1.0/1.0	0.78/1.0	1.0/0.9	0	0	0	0	0	0	0	1	**9**
Total	0.88/0.88	0.89/0.93	0.93/0.93									

Table 1: Recall and Precision rates for the kNN and RVM classifiers on the Weizmann dataset

Class	R/P (SP-RVM)	R/P (NN)	R/P (DTW)	R/P (RVM)
1	1.0/1.0	0.9/1.0	1.0/1.0	1.0/1.0
2	1.0/0.63	1.0/0.91	1.0/0.83	1.0/0.83
3	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
4	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
5	0.7/0.78	1.0/0.56	0.4/0.67	1.0/0.91
6	0.6/0.5	0.3/0.3	0.6/0.35	0.8/0.8
7	0.7/1.0	0.9/1.0	1.0/0.91	0.9/0.9
8	1.0/1.0	1.0/1.0	0.9/1.0	0.9/0.9
9	1.0/1.0	1.0/0.67	1.0/0.71	0.9/0.9
10	0.7/0.78	0.1/0.5	0.4/0.67	0.8/0.89
11	1.0/1.0	1.0/1.0	1.0/1.0	0.9/1.0
12	1.0/1.0	1.0/0.91	1.0/1.0	1.0/1.0
13	0.7/1.0	0.9/1.0	0.8/1.0	0.8/1.0
14	1.0/1.0	1.0/0.91	1.0/1.0	1.0/0.91
15	1.0/1.0	0.5/1.0	0.7/1.0	1.0/1.0
Total	0.89/0.91	0.84/0.85	0.85/0.88	0.93/0.94

Table 2: Recall and Precision rates for the kNN and RVM classifiers on the aerobics dataset

4 Experimental Results

In order to evaluate the proposed method we use two different datasets. The first is the one used in [3], containing 9 different actions such as walking, running and jumping. The second dataset [1] is one created by our group, containing a set of 15 different aerobic exercises, performed twice by five different subjects.

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 created a codebook and trained the respective classifiers using all available data except for those belonging to the same class and performed by the same subject as the test exercise. We present three different sets of classification results. In the first set, each frame of a test sequence is matched with the closest frame of a training sequence in terms of their χ^2 distance and an overall distance measure is calculated as the sum of the minimum calculated frame distances. The test example is then classified to the class of the training example with the smallest overall distance (Chamfer distance). In the second set, each test example is classified to the class of the training example with the highest similarity, as this is calculated by the DTW procedure. Finally, we present results using an RVM classifier according to eq. 6. In Table 1 we present our classification results for the Weizmann dataset, in the form of recall and precision rates. Similar classification results for the aerobics dataset are given in Table 2. In the same Table, we also show classification results on this dataset based on the algorithm of [12] (denoted as SP-RVM), in which only the location of the spatiotemporal points was considered . As we can see, there is considerable improvement, which demonstrates the descriptive power of the proposed B-spline based representation.

As we can see from Tables 1 and 2, there is a slight increase in classification performance in the Weizmann and aerobics datasets using DTW, while there is a considerable increase of almost 5% by additionally

[1] This dataset is available from the author's website

using RVM. Although the increase is small, the use of DTW adds structure and consistency to the representation. In general, introduction of structure is important and expected to show benefits in datasets with larger number of classes. Using DTW, frames that are far apart from each other in terms of time cannot be matched. In the case of a classification method with no temporal structure, these kind of restrictions do not exist, and a frame in the beginning of a sequence can be matched with any frame of another sequence, as long as their χ^2 distance is small.

The average recall rate for the Weizmann dataset is about 93%. From the confusion matrix of the Table 1, we notice that there are reasonable confusions between similar classes like *jump*, *run*, *walk* and *side*, as well as *wave1* and *wave2*. Concerning the results on the aerobics dataset, we notice from Table 2 that there is low performance on classes $5, 6$ and 10 for the NN and DTW classifications, which considerably increases using the RVM classifier. The reason for this is that these classes are very similar and concern motions like squatting with an upright torso or bending while the subject is facing the camera. In order to discriminate between these motions, depth information is important, and since our features stem from salient point representations, they have difficulty recovering it.

Compared to the work of [3] and [6], our classification results are almost 4% lower. The use of DTW from our system, however, introduces structure to the recovered short-term motions and classification is performed based on this structure. On the contrary, in [3], [6] features are matched based on maximum similarity across whole image sequences. In addition, our system uses a sparse representation as opposed to [3], where a whole image sequence is represented as a space-time shape. Sparse, local representations, are shown to be significantly better in dealing with clutter and occlusions for object detection and recognition in comparison to global representations. Similar observations are expected to hold in the problem of action recognition. A sparse and structured representation is used in [11], where a recognition rate of 72.8% is reported on the Weizmann dataset, by far inferior to the 93% achieved by our method.

We used a leave-one-subject-out approach in order to evaluate our method. This means that for any test example, the created codebook contains information about the class of this example, although from different same-class examples. We would like to determine, if our features are general enough to handle completely unknown classes, that is, given a codebook of verbs how well is this codebook able to discriminate classes that did not contribute at all to its creation. Our motivation for this experiment lies in the fact that our system is able to consistently recover short-term motion in small spatiotemporal regions. Therefore, given an unknown class that shares a number of similar such regions with several known classes, there should be some limited ability for good discrimination. We performed two different experiments. In the first experiment we created a codebook from 14 classes of the aerobics dataset, completely excluding class 3, which was kept out for testing. The result was 8 out of 10 instances of the test class correctly classified. In the second experiment, we created a codebook from the whole aerobics dataset and tested it on the Weizmann dataset. The classes between these two datasets are completely different, except for the class *jack* of the Weizmann dataset which is similar to class 1 of the aerobics dataset and classes *wave1* and *wave2* of the Weizmann dataset which look like classes 2 and 7 of the aerobics dataset. The average recall rate for this experiment was 67.5%, with the worst performing classes being *jump*, *run* and *walk*. This result is reasonable, as these classes do not seem to share common poses with the ones in the aerobics dataset. These results indicate that it might be possible to use the proposed descriptors for representing new classes of actions. We intend to investigate on the issue of the size of the action database and perform the same experiments with features that are currently the state of the art in the field, like the features of [3], [6] and [11].

5 Conclusions

In this paper we presented a feature based method for human activity recognition. The features that we extract stem from automatically detected salient points and contain static information concerning the moving body parts of the subjects as well as dynamic information concerning the activities. We used the extracted features in order to recover the pose and the short-term motion of the subject in a 'bag of verbs' approach. Our results show that our representation is able to recover the kind of motion performed in a variety of different cases. Furthermore, our preliminary experiments show that our system is able to generalize well and handle unknown classes, which do not contribute to the creation of the utilized codebook at all.

Our future directions include additional experiments in order to determine the robustness of the proposed method in more challenging scenarios, like in the presence of dynamic background or moving camera. Furthermore, we intend to implement different, more efficient methods for codebook creation.

Acknowledgments

This work is in part financially supported by the MAHNOB project funded by the European Research Council under the ERC Starting Grant agreement No. ERC-2007-StG-203143.

References

[1] A. Pinz A. Opelt and A. Zisserman. A boundary-fragment-model for object detection. In *Proceedings of the European Conference on Computer Vision*, 2006.

[2] A. Agarwal and B. Triggs. Hyperfeatures multilevel local coding for visual recognition. *European Conference on Computer Vision*, 1:30–43, 2006.

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Proc. IEEE Int. Conf. Computer Vision*, 2:1395 – 1402, 2005.

[4] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *European Conference on Computer Vision*, 2:428 – 442, 2006.

[5] T.X. Han, H. Ning, and T.S. Huang. Efficient Nonparametric Belief Propagation with Application to Articulated Body Tracking. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 1:214– 221, 2006.

[6] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A Biologically Inspired System for Action Recognition. *Proc. IEEE Int. Conf. Computer Vision*, 2007.

[7] I. Laptev and T. Lindeberg. Space-time Interest Points. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 432 – 439, 2003.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110, 2004.

[9] M. Marszalek and C. Schmid. Spatial weighting for bag-of-features. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2118– 2125, 2006.

[10] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, pages 985–992, 2006.

[11] J.C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[12] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal Salient Points for Visual Recognition of Human Actions. *IEEE Trans. Systems, Man and Cybernetics Part B*, 36(3):710 – 719, 2005.

[13] R. Poppe. Vision-based human motion analysis: An overview. *Comp. Vision, and Image Understanding*, 108(1-2):4–18, 2007.

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

[15] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering Objects and their Location in Images. *Proc. IEEE Int. Conf. Computer Vision*, 1:370 – 377, 2005.

[16] J. Sivic and A. Zisserman. Video Google: Efficient visual search of videos. In *LNCS*, volume 4170, pages 127–144, 2006.

[17] B. Stenger, A. Thayananthan, P.H.S. Torr, and R. Cipolla. Model-Based Hand Tracking Using a Hierarchical Bayesian Filter. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(5):1372–1384, 2006.

[18] M.E. Tipping. The Relevance Vector Machine. *Advances in Neural Information Processing Systems*, pages 652 – 658, 1999.