An Implicit Spatiotemporal Shape Model for Human Activity Localization and Recognition

A. Oikonomopoulos
Computing Department
Imperial College London
London,UK
aoikonom@imperial.ac.uk

I. Patras
Electronic Engineering Department
Queen Mary University of London
London,UK
ioannis.patras@elec.qmul.ac.uk

M. Pantic
Computing Department
Imperial College London
EEMCS,Univ. Twente, NL
m.pantic@imperial.ac.uk

Abstract

In this paper we address the problem of localisation and recognition of human activities in unsegmented image sequences. The main contribution of the proposed method is the use of an implicit representation of the spatiotemporal shape of the activity which relies on the spatiotemporal localization of characteristic, sparse, 'visual words' and 'visual verbs'. Evidence for the spatiotemporal localization of the activity are accumulated in a probabilistic spatiotemporal voting scheme. The local nature of our voting framework allows us to recover multiple activities that take place in the same scene, as well as activities in the presence of clutter and occlusions. We construct class-specific codebooks using the descriptors in the training set, where we take the spatial co-occurrences of pairs of codewords into account. The positions of the codeword pairs with respect to the object centre, as well as the frame in the training set in which they occur are subsequently stored in order to create a spatiotemporal model of codeword co-occurrences. During the testing phase, we use Mean Shift Mode estimation in order to spatially segment the subject that performs the activities in every frame, and the Radon transform in order to extract the most probable hypotheses concerning the temporal segmentation of the activities within the continuous stream.

1. Introduction

Due to its practical importance for a wide range of vision-related applications like video retrieval, surveillance, and Human-Computer Interaction, vision-based analysis of human motion is nowadays one of the most active fields of computer vision. Given a video sequence, humans are usually able to deduce quickly and easily information about its content. However, when it comes to computers, robust activity detection still remains a very challenging task. Different conditions that might be prevalent during the conduction of an activity, like a moving camera, dynamic background, occlusions, abrupt illumination changes and multiple subjects in the scene, pose a significant difficulty in the development of a robust motion analysis framework. This is evident from the abundance of different motion analysis approaches that have been developed [20] [19].

Local spatiotemporal feature-descriptor representations have been extensively used for activity recognition, due to their robustness against illumination, clutter, and viewpoint changes. Detection of keypoints, in particular, has been a very popular choice, due to their sparsity and detection simplicity. Typical examples include keypoints detected using sets of separable linear filters, including gaussian kernels [11], Difference of Gaussians (DoG) [14] and 1D Gabor filters [7], [17]. Wong and Cipolla [27] use global information in terms of dynamic textures in order to detect their salient points and minimize noise. Their resulting representation is then used in a non-negative matrix factorization scheme [12] in order to obtain a subspace representation of the activities. Features based on shape representations have also been extensively investigated, the most notable being the shape contexts of Belongie et al. [21], the Motion History Images (MHIs) of Bobick and Davis [5] and the space-time shapes of Blank et al. [4]. Features based on video volume tensors are implemented in [10], reporting interesting results. Interesting results have also been reported using features based on the human visual cortex, like the C-features of Jhuang et al. [9].

Despite their success in action classification, the main disadvantage of these methods is that they work on temporally segmented sequences. That is, they do not provide any means of localizing the actions in a continuous video stream. On the other hand, the problem of segmentation and classification has been extensively studied in static images, especially in the case of object detection/classification. Voting frameworks, in particular, have been very popular, due to their local nature and robustness

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

against clutter and occlusions. Typical examples are the boundary fragment model of Opelt et al. [1] and the implicit shape model of Leibe et al. [13], where the positions of the object descriptors are stored, during training, with respect to the object center. During testing, each descriptor that is matched against the codebook casts probabilistic votes to where the object center lies. In this way an estimate of the position of the object center is obtained. A similar method is presented by Marszalek and Schmid [15], where features belonging to the foreground (i.e. the object of interest) are positively weighted compared to the ones belonging to the background. Internal structure of objects has also been extensively studied, the most notable being the self similarity descriptor of Shechtman and Irani [23]. Their concept is also extended in order to deal with human activities. Their method requires a minimal training set in order to work, consisting of a single action instance.

The goal of this work is to present a framework able to spatiotemporally localize instances of activities taking place in a scene and assign them to an action category. Our method follows a voting approach, and in this way is similar to the work of Leibe et al. [13] on object segmentation and recognition. The central novelty of our method, however, is the use of a temporal voting framework for segmenting the activities in time, which in our point of view, is the most challenging task in activity detection. We use a combination of optical flow and appearance descriptor vectors as the base for our computations, computed around spatiotemporal salient points. The latter are detected using the algorithm described in [18]. Subsequently, we create class-specific codebooks, where we take into account pairs of spatially co-occurring codewords, similar to the doublets of Sivic et al. [25]. Similar to the works of Leibe et al. [13] and, Mikolajczyk and Uemura [16], we store the positions of the codeword pairs with respect to the object center. Furthermore, we store the frame at which they occur, and create a spatiotemporal model of codeword co-occurrences for each class. We use Mean Shift Mode estimation for spatially segmenting the subjects performing the actions at each frame. In addition, we develop a novel algorithm for segmenting the activities in time, based on the Radon transform [26], with which we aim to recover patterns of temporal votes in the temporal voting space. Contrary to Gilbert et al. [8], where the temporal center of the action is estimated, our method estimates the start/end frames of the activities. Our method then accepts the most probable hypotheses as the result of the segmentation, which we use in order to classify the examples in our dataset. We demonstrate the effectiveness of our method by presenting classification and segmentation results in the challenging KTH dataset, as well as in sequences containing clutter and spatiotemporal occlusions.

The remainder of this paper is organized as follows. In section 2 we present our spatiotemporal voting framework, including the creation of the spatiotemporal co-occurrences model. In section 3 we present our segmentation procedure, giving additional emphasis on the temporal segmentation, as the central novelty of the proposed method. In section 4 we present our experimental results, including segmentation and classification in the presence of clutter and occlusion. Finally, in section 5 we draw some conclusions.

2. Spatiotemporal Voting Framework

Inspired by [13], we create a spatiotemporal model of codeword co-occurrences for each class in our dataset. Each model consists of a class-specific codebook of optical flow and appearance descriptors and a spatiotemporal probability distribution, which specifies where in space and time pairs of codebook entries (codewords) appear in the the dataset, with respect to the subject center, spatial lower bound, and beginning of the action. During testing, we extract descriptor vectors from the test set and match them to each codebook. The activated codewords cast probabilistic votes to possible centers and spatial lower bounds of the subjects for each frame, as well as for the frame at which they are located with respect to the beginning of the action.

2.1. Training set

To create a training set, we select a subset of action instances in our dataset and register these instances in space and time. More specifically, we manually localize the upper and lower bounds of the subjects at each frame of each selected instance and resize the instances so that the subjects in them have the same size. Moreover, we normalize the selected instances so that they have the same duration. The latter is performed linearly, by assuming that the execution speed of each activity is constant. To use them in our learning process, we manually localize and store the subject centers and lower bounds in the registered training set, where each center is defined as the middle of the torso.

2.2. Features

We use a combination of optical flow and spatial gradient descriptors as the base for our computations. These descriptors are extracted around spatiotemporal salient points, detected using the method of Oikonomopoulos et al. [18]. Our motivation is to detect spatiotemporal interest points and spatio(temporal) descriptors at areas with significant variation in motion information, such as motion discontinuities, rather than spatiotemporal intensity discontinuities, such as space-time intensity corners (see [11]). In order to eliminate general camera motion, including camera translation, small rotations, and scale changes (resulting from camera zoom), we apply a local median filter to the extracted optical flow field calculated in the input image sequences.

Subsequent to the salient point detection, we extract optical flow and appearance descriptor vectors that lie within the area of support of the salient points, defined by the spatiotemporal scale at which the points are detected. We use the algorithm in [3] for computing the optical flow, and spatial gradient vectors as appearance descriptors. Using their horizontal and vertical components, we convert these vectors into angles and we bin them into histograms using a bin size of 10 degrees. This results into two vectors of size 1×37 for every salient point position, one for each of the optical flow and appearance part of the representation. By concatenating these vectors we end up with a single descriptor vector of size 1×74.

2.3. Spatiotemporal co-occurrences model

In order to create our models we follow a multi-step approach. First, the extracted descriptors are clustered in order to create a set of codebooks I_C, one for each action category. The χ^2 distance is used as a metric between the descriptors. In the second step, we iterate through the training set and match the descriptors with the codebook entries. As in [13], we activate each codeword i whose distance is smaller than the standard deviation of the cluster J_i that the codeword represents. Subsequently, we recover pairs of activated codewords that co-occur in the dataset. To do so, we follow a similar approach as the one of Sivic et al. [25], by pairing each activated codeword with one of its 5 nearest neighbors. From these pairs we discard the ones that significantly overlap in terms of their spatial scale. For each remaining codeword pair we store the positions of each pair member with respect to the center of the subject as well as with respect to the subject's lowest bound (i.e. the subject's feet). Let us here denote with m and m' the indexes of the pair of descriptors that activate the codeword pair. Then, the positions of the descriptor pair members with respect to the subject center are stored as $(r_m, \phi_m), (r'_m, \phi'_m)$, where r_m, r'_m are the distances of each pair member from the subject center and ϕ_m, ϕ'_m are the angles between the vector defined by each pair member and the subject center, with the horizontal axis. Similarly, the positions of the pair members with respect to the subject's lower spatial bound are stored as $(d_m, \theta_m), (d'_m, \theta'_m)$. In addition to these parameters, we also store the angle $\alpha_{mm'}$ between the vector defined by the pair members and the horizontal axis, as well as the scales σ_m, σ'_m at which the descriptors that constitute the pair were extracted. An illustration of these parameters for one of the members of a pair is shown in Fig. 1. We follow a similar process for the temporal part of our model, by storing the frame number f_m at which each pair is occurring in the training set with respect to the beginning of the action, as well as the temporal scales t_m, t'_m of the members of each pair. The latter are automatically detected by the spatiotemporal salient point detection algorithm of [18].

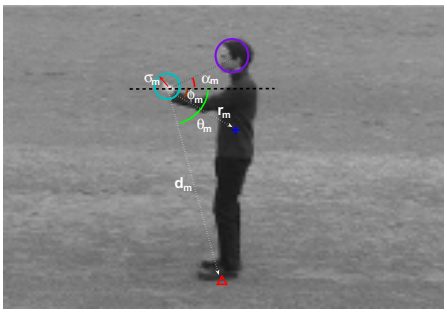

Figure 1. An overview of the stored spatial parameters per each codeword pair during the learning phase. For illustration purposes, only the position parameters for one of the constituent codewords are depicted.

During testing, we extract from the test sequences a set of optical flow and appearance descriptor vectors, localized at sparse locations detected by the algorithm described in [18]. In order to compensate for camera motion, we perform both salient point detection and descriptor extraction on filtered versions of the optical flow field of the action. Subsequently, we match the descriptors to the codebook to recover the activated codewords and we activate, as in the learning phase, doublet codeword pairs. We perform this procedure once for each of the action classes in the dataset. Let us denote by q and q' the indexes of the descriptor pairs in the test set and i and i' respectively the indexes of the codewords that they activate. Since we want to take into account only pairs with similar spatial configuration as the ones stored during learning, we only keep pairs whose relative angle $\alpha_{qq'}$ is similar to stored values $\alpha_{mm'}$, where m and m' belong to the clusters J_i and J'_i respectively. Using the information stored during the learning phase, we cast spatial and temporal votes for possible subject centers/lower spatial bounds and frame numbers respectively. This procedure is equivalent to a Generalized Hough Transform [2]. Each member of the activated codeword pair casts votes independently. To compensate for spatiotemporal scale variations, we normalize the votes using the scales σ_q of the activated codewords in the test set and the σ_m values that were stored during learning. A center hypothesis for each codeword is given by the following equation:

$$\begin{bmatrix} x_h \\ y_h \end{bmatrix} = \begin{bmatrix} x_q \\ y_q \end{bmatrix} - r_m \frac{\sigma_q}{\sigma_m} \begin{bmatrix} \sin(\phi_m) \\ \cos(\phi_m) \end{bmatrix}, \quad (1)$$

where h refers to the hypothesis, (x_q, y_q) is the location of the test codeword and (x_h, y_h) are the coordinates of the center hypothesis. By normalizing with the scale ratio σ_q/σ_m, we achieve invariance to scale changes.

A smaller/larger scale indicates that the subject is of a smaller/larger size than the one during training, and therefore, the votes should be adjusted so that the center hypothesis is closer/further from the codeword than the stored center. For the subject's lower spatial bound hypothesis the same equation holds, by just substituting r_m, ϕ_m with d_m, θ_m respectively.

For the temporal case, each codeword casts votes for the temporal phase of the action according to the following equation:

$$f_h = \frac{t_q}{t_m} f_m, \tag{2}$$

where $f_h, f_m \in \{1 \dots L\}$ are the hypothesis and model frame numbers respectively and t_q, t_m are the temporal scales of the test codewords and the model respectively. By using the temporal scales for normalization, we achieve invariance in temporal scale changes, since a smaller/larger temporal scale in the test set indicates a compression/expansion in time with respect to the training set.

2.4. Probabilistic Formulation

We apply a probabilistic framework for our spatial and temporal votes, inspired by [13] [28]. In this section, we will describe the framework for the temporal case. Similar equations are used in the spatial case by substituting time with space variables.

Let us denote by e_q an observed spatiotemporal patch. The probability that this patch is located on frame f_h with respect to the beginning of the activity is given by:

$$p(f_h) = \sum_q p(f_h|e_q)p(e_q), \tag{3}$$

where we assume that $p(e_q)$ is a uniform distribution, i.e. $p(e_q) = 1/K$, where K is the total number of observed patches. Each e_q is matched to a set of codewords J_i. By marginalizing $p(f_h|e_q)$ on J_i we get:

$$p(f_h|e_q) = \sum_{J_i} p(f_h|J_i, e_q)p(J_i|e_q). \tag{4}$$

Then, we model the term $p(J_i|e_q)$ as:

$$p(J_i|e_q) \propto \exp\left(\frac{-D(J_i, e_q)}{s_i^2}\right), \tag{5}$$

where $D(.)$ is the χ^2 distance and s_i^2 is the variance of the cluster J_i for codeword entry i. The term $p(f_h|J_i, e_q)$ is independent of e_q, and expresses the probabilistic vote on f_h given an activated codebook entry i. Let us denote by f_m the temporal vote of the instance m that belongs to the cluster J_i, and that is stored during training. Then, $p(f_h|J_i)$ is defined as:

$$p(f_h|J_i) = \sum_m p(f_h|f_m, J_i)p(f_m|J_i). \tag{6}$$

The first term of the summation in eq. 6 is independent of J_i, since a vote is casted from each individual member of J_i. Furthermore, our votes are limited at possible frame instances, and therefore $p(f_h|f_m) = \delta(f_h - f_m)$. Finally, we assume a uniform distribution for $p(f_m|J_i)$, that is, $p(f_m|J_i) = 1/M$, where M is the number of instances associated with J_i.

3. Spatiotemporal Segmentation

We apply Mean Shift Mode Estimation [6] to localize the most probable center and lower spatial bound of the subject per frame. Subsequently, we use these hypotheses and anthropometric models in order to localize a bouncing box on the subject, as shown in Fig. 2.

In order to perform temporal segmentation, we select the codewords that contributed to the most probable subject center and only take their temporal votes into account. An example of a temporal voting space is shown in the top part of Fig. 2, for three repetitions of a handwaving activity from the KTH dataset. In the case that the speed of the execution of an action does not change in its duration, the temporal votes will lay on a line whose tangent equals the ration of the speed of the execution of the action in the test sequence with respect to the speed of the the execution of the action in the registered training set. In order to detect these patterns and extract temporal segmentation hypotheses, we use the Radon transform [26], whose basic property is the transformation of two dimensional images containing structures similar to lines, into domains of possible line parameters.

In order to use the Radon transform we use a popular line expression of the form $\rho = xcos(\theta) + ysin(\theta)$, where θ is the angle and ρ the smallest distance to the origin of the coordinate system. Given this expression, the Radon transform of an image $g(x, y)$ is given by the following equation:

$$\breve{g}(\rho, \theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y)\delta(\rho - x\cos\theta - y\sin\theta)dx\,dy, \tag{7}$$

where $\delta(.)$ is the Dirac delta function.

We employ a temporal sliding window of varying dimensions in order to temporally segment the sequences. Our motivation lies in the notion that, for the case of a temporal window approximately coinciding with the ground truth segmentation, the peak strength of its Radon transformation will be maximum, or equivalently, the sum of temporal votes lying across the line with parameters defined by the strongest peak of the transform will be maximized. In order to illustrate this process, we present in Figure 3 a set of Radon transform domains, corresponding to a smaller, exact and a larger temporal window than the ground truth, along with the corresponding peak values.

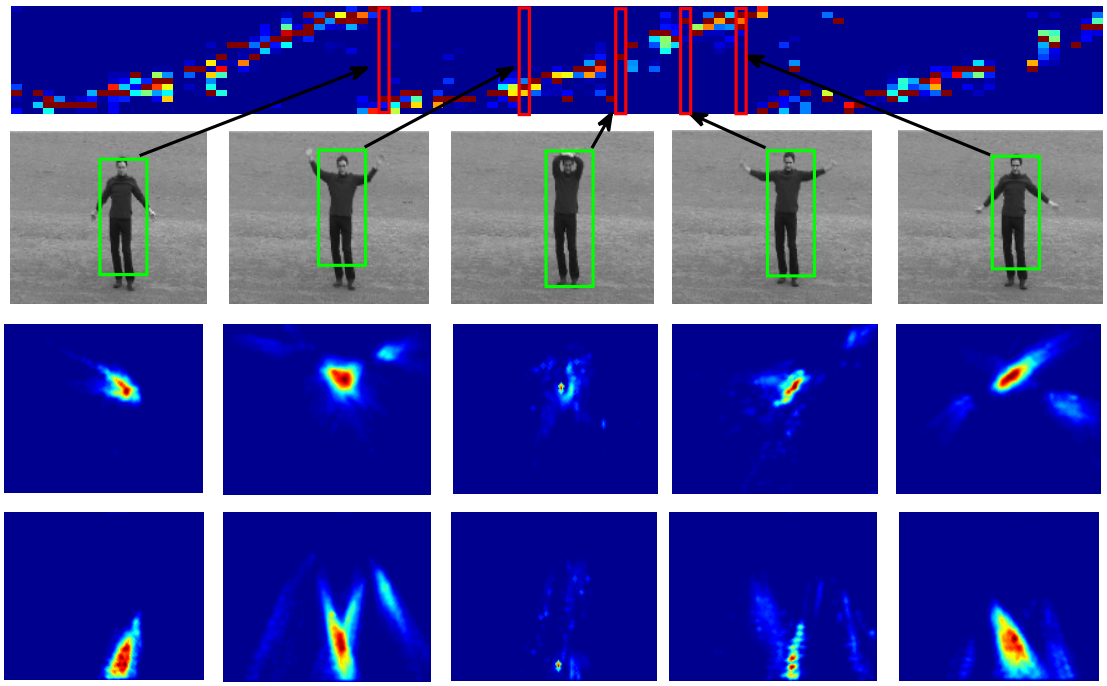

Figure 2. Illustration of the spatiotemporal voting scheme. Top row: Temporal voting space. Activated codewords in each frame of the unsegmented sequence (horizontal axis) vote for their location with respect to the beginning of the action (vertical axis). Note that frames at the beginning vote for both the start and the end of the activity and vice versa. Third, fourth row: Spatial voting spaces. Each codeword votes for its location with respect to the center and lower spatial bound of the subject. Second row: Fitted bounding box resulting from the maximum responses in the spatial voting spaces.

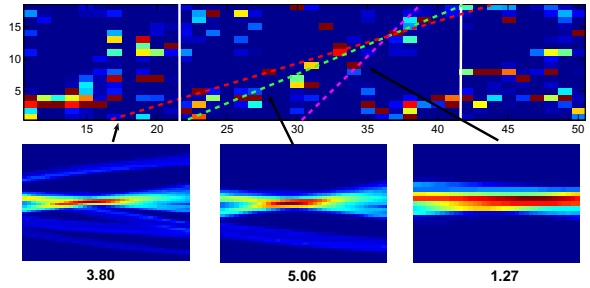

Figure 3. Top row: temporal vote pattern recovered by our algorithm (green line), marking frames 23,40 as start/end frames of the activity instance respectively. Bottom row: Radon transform domains corresponding, from left to right, to larger, exact and smaller temporal windows than the ground truth segmentation. The values noted below denote the peak value of the transform.

4. Experimental Results

4.1. Classification

For our classification experiments, we use the KTH dataset, which depicts activities like *boxing, handclapping, handwaving, jogging, running* and *walking*. We use pre-segmented activity instances in order to evaluate the classification accuracy of our algorithm, by manually segmenting the action instances in the KTH examples. Since we learn a different spatiotemporal model for each activity, we get six different responses for each test example, each corresponding to one of the learned models. To classify an example, we seek for the response that results to the strongest subject center estimates for this example, as these are given by the Mean Shift Mode estimation process. Since this estimation is performed per frame, we can obtain a score by summing up the responses for all frames. Let us denote by e_{ijk} the maximum response of the mode estimation per frame k, where i, j are, respectively, the example and class/codebook indexes. Then, each example is classified according to the following equation:

$$Class(i) = \arg\max_j \left(\sum_k e_{ijk} \right). \quad (8)$$

The average classification rate achieved by this process is 85%. From the confusion matrix in Fig. 4, we can see that *running* is frequently confused with *jogging*. While the differences between these two actions are small, as also noted by Schuldt et al [22], we believe that the main reason for

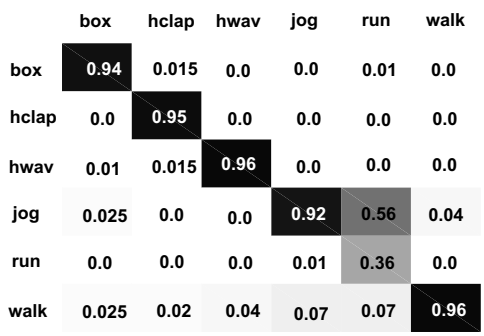

Figure 4. Confusion Matrix for the KTH dataset.

Table 1. Activity instance recovery percentages per class

Class	box	hclap	hwav	jog	run	walk
% retrieved	0.9	0.89	0.88	0.91	0.38	0.91

this confusion in our case is the temporal registration that is being performed during training, since it eliminates the basic difference between these two actions, which is speed. Finally, it is interesting to note that there are minimal confusions in the case of the *walking* class, showing that speed is not as important in order to distinguish this class from either *running* or *jogging*. This is mainly because the posture of the legs while walking is very different from that typical for running and jogging (straight rather bended).

4.2. Temporal Segmentation

To evaluate the efficiency of our algorithm in temporal segmentation, we apply the process described in section 3 to the unsegmented sequences of the KTH dataset. Since we apply this process for each learned model, each resulting hypothesis corresponds to a specific class, and due to the use of a temporal sliding window, it has a specific extent in time. We assign a score to each of these hypotheses, equal to the cumulative sum of the votes that lie along the path of the hypothesis. Based on this score, we select the most probable hypotheses and we record, in Table 1 the percentage of the correctly recovered instances per class, where we consider a hypothesis as correct if it comes from the same model as the ground truth and if the segmentation is within 5 frames from the actual ground truth segmentation.

Finally, we demonstrate the robustness of the proposed method to temporal occlusion. This property is a direct consequence of the use of the Radon transform for temporal segmentation. Since the temporal patterns that the algorithm is trying to recover correspond to straight diagonal lines (constant speed assumption), they will still be detected even if a part of them is missing, or equivalently, if part of the duration of the action is obscured. In order to demonstrate this property, we apply the temporal segmentation algorithm in a test sequence which has been cropped so that it includes only half of the activity. The resulting temporal voting space is shown at the bottom part of Fig. 5, where the missing frames have been padded with zeros. Due to the use of the Radon transform, the activity is still recovered, albeit with an error Δt in the end-frame estimate.

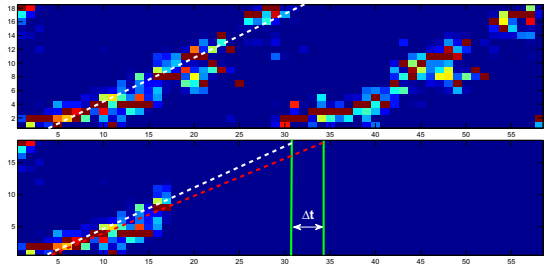

Figure 5. Handling of temporal occlusions. Actual segmentation is shown by the white dotted line. Segmentation recovered by the algorithm in the event of an obscured sequence is shown by the red dotted line. For ease of comparison, the missing frames have been padded with zero values in the bottom part of the figure.

4.3. Spatial Segmentation

In order to demonstrate the effectiveness of our method in spatially segmenting an activity in the presence of clutter and occlusion, we use a number of synthetic as well as a number of real image sequences. In Fig. 6 we show the voting spaces for a synthetic sequence of two different activities taking place simultaneously. As can be seen from the figure, the proposed method is able to distinguish both actions, since the majority of the votes in the voting spaces corresponding to the activities are concentrated on the true positions of the subjects. We show similar results in Fig. 7, using the *beach* sequence of Shechtman and Irani et al. [24], in which there are two different *walking* activities in the opposite directions, while there is a limited amount of occlusion (umbrella) and dynamic background (sea). From the resulting voting spaces, it is obvious that the proposed framework is still able to localize both subjects depicted in the sequence.

5. Conclusions

In this paper we presented a method for detecting instances of human activities in unsegmented sequences, where the term detection refers to the spatiotemporal segmentation and simultaneous classification into a set of action categories. Our spatiotemporal voting framework makes the proposed method robust to the presence of clutter and occlusion. We have presented a novel method for temporal segmentation based on the Radon transform. Moreover, we have demonstrated the robustness of our method in temporal occlusion, derived from fundamental properties

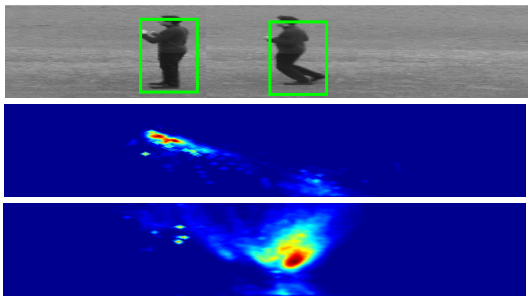

Figure 6. From top to bottom: Synthetic image sequence containing the *boxing* and *jogging* actions, spatial voting space for *boxing*, spatial voting space for *jogging*.

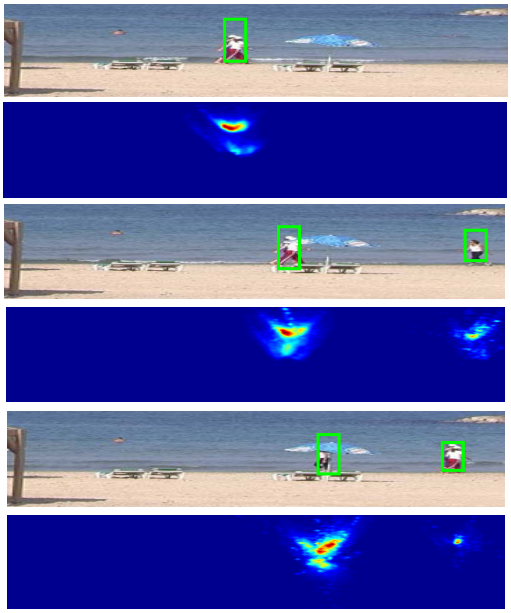

Figure 7. Instances of the beach sequence, along with center estimates and resulting localization.

of the Radon transform.

Acknowledgments

This work has been supported in part by the EC's 7th Framework Programme [FP7 / 2007-2013] under grant agreement no. 231287 (SSPNet) and in part by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB).

References

[1] A. P. A. Opelt and A. Zisserman. A boundary-fragment-model for object detection. In *Proceedings of the European Conference on Computer Vision*, 2006. 2

[2] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):109–122, 1981. 3

[3] M. Black and P. Anandan. A framework for the robust estimation of optical flow. In *ICCV*, pages 231–236, 1993. 3

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *Proc. IEEE Int. Conf. Computer Vision*, volume 2, pages 1395 – 1402, 2005. 1

[5] A. Bobick and J. Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(3):257 – 267, 2001. 1

[6] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(8):790 – 799, 1995. 4

[7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. *VS-PETS*, pages 65– 72, 2005. 1

[8] A. Gilbert, J. Illingworth, and R. Bowden. Scale invariant action recognition using compound feature mined from dense spatio-temporal corners. In *European Conference on Computer Vision*, volume 530, pages 222–233, 2008. 2

[9] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A Biologically Inspired System for Action Recognition. In *Proc. IEEE Int. Conf. Computer Vision*, 2007. 1

[10] T. Kim, S. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 20087. 1

[11] I. Laptev and T. Lindeberg. Space-time Interest Points. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 432 – 439, 2003. 1, 2

[12] D. Lee and H. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401(6755):788–791, 1999. 1

[13] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. *ECCV'04 Workshop on Statistical Learning in Computer Vision*, pages 17–32, 2004. 2, 3, 4

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91 – 110, 2004. 1

[15] 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. 2

[16] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[17] J. 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. 1

[18] 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. 2, 3

[19] M. Pantic, A. Pentland, A. Nijholt, and T. Huang. Human computing and machine understanding of human behavior: A survey. *Lecture Notes in Artificial Intelligence*, 4451:47–71, 2007. 1

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

[21] J. M. S. Belongie and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(4):509 – 522, 2002. 1

[22] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, volume 3, pages 32– 36, 2004. 5

[23] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 2

[24] E. Shechtman and M. Irani. Space-time behavior based correlation or how to tell if two underlying motion fields are similar without computing them? *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29(11):2045–2056, 2007. 6

[25] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering Objects and their Location in Images. In *Proc. IEEE Int. Conf. Computer Vision*, volume 1, pages 370 – 377, 2005. 2, 3

[26] P. Toft. *The Radon Transform - Theory and Implementation*. PhD Thesis, 1996. http://pto.linux.dk/PhD. 2, 4

[27] S. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *Proc. IEEE Int. Conf. Computer Vision*, pages 1–8, 2007. 1

[28] X. Yu, Y. Li, C. Fermuller, and D. Doermann. Object detection using a shape codebook. In *British Machine Vision Conference*, 2007. 4