Spatiotemporal Localization and Categorization of Human Actions in Unsegmented Image Sequences

Antonios Oikonomopoulos, *Member, IEEE*, Ioannis Patras, *Member, IEEE*, and Maja Pantic, *Senior Member, IEEE*

***Abstract*—In this paper we address the problem of localization 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 ensembles of feature descriptors. Evidence for the spatiotemporal localization of the activity is accumulated in a probabilistic spatiotemporal voting scheme. The local nature of the proposed voting framework allows us to deal with multiple activities taking place in the same scene, as well as with activities in the presence of clutter and occlusion. We use boosting in order to select characteristic ensembles per class. This leads to a set of class specific codebooks where each codeword is an ensemble of features. During training, we store the spatial positions of the codeword ensembles with respect to a set of reference points, as well as their temporal positions with respect to the start and end of the action instance. During testing, each activated codeword ensemble casts votes concerning the spatiotemporal position and extend of the action, using the information that was stored during training. Mean Shift mode estimation in the voting space provides the most probable hypotheses concerning the localization of the subjects at each frame, as well as the extend of the activities depicted in the image sequences. We present classification and localization results for a number of publicly available datasets, and for a number of sequences where there is a significant amount of clutter and occlusion.**

***Index Terms*—Action detection, space-time voting.**

I. Introduction

THE goal of this work is to develop a method able to spatiotemporally localize instances of activities depicted in an image sequence and assign them to an action category. The problem, termed as activity detection, has been a long lasting subject of research in the field of computer vision, due to its importance in applications such as video retrieval, surveillance, and Human-Computer Interaction. Robust activity detection using computer vision remains a very challenging task, due to different conditions that might be prevalent during the conduction of an activity, such as a moving camera, dynamic background, occlusions and clutter. For an overview of the different approaches we refer the reader to [1], [2].

The success of interest points in object detection, their sparsity, and robustness against illumination and clutter [3] have inspired a number of methods in the area of motion analysis and activity recognition. A typical example are the space-time interest points of Laptev and Lindeberg [4], which are an extension of the Harris corner detector in time. Han *et al.* [5] extract features based on Histograms of Gradients (HoG) and Histograms of Flow (HoF) around space-time interest points for recognition of actions in movies. Dollar *et al.* [6] use 1-D Gabor filters in order to capture intensity variations in the temporal domain. In [7], this approach is refined by using Gabor filters in both spatial and temporal dimensions. Oikonomopoulos *et al.* [8] detect spatiotemporal salient points in image sequences by extending in time the spatial salient points of Kadir and Brady [9]. Lowe introduces the Scale Invariant Feature Transform (SIFT) in [10], which has been used in a variety of applications, including object (e.g., [11]) and scene classification (e.g., [12], [13]). Partly inspired by SIFT, the Speeded Up Robust Features (SURF) [14] utilize second order Gaussian filters and the Hessian matrix in order to detect interest points. Jhuang *et al.* [15] use a hierarchy of Gabor filters in order to construct their C-features. Their method is extended by Schindler and Van Gool [16], by combining both shape and optical flow responses. Finally, Ali and Shah [17] use kinematic features extracted around optical flow vectors in order to represent human activities.

Visual codebooks have been extensively used for detecting objects, humans and activities. SIFT descriptors are used in a bag-of-words framework by Li and Fei-Fei [12] for the combined problem of event, scene, and object classification. Laptev *et al.* [18] extract HoG and HoF descriptors around detected space-time interest points, and use k-means in order to construct a codebook. Similar is the work presented in [19], where SIFT features are also used. Using the space-time interest points of [4], Niebles *et al.* [20] represent each class as a distribution of visual words from the codebook and learn a pLSA model [21] on each of the representations. Similar to [15], Ning *et al.* [22] use the responses of 3-D Gabor filter banks in order to build their descriptors. A bag of words model is subsequently used in order to localize instances of human activities in videos using sliding temporal windows of varying duration. Finally,

Manuscript received January 15, 2010; revised June 01, 2010, July 30, 2010; accepted August 06, 2010. Date of publication September 16, 2010; date of current version March 18, 2011. This work has been supported in part by the European Community's 7th Framework Programme (FP7/20072013) under the Grant Agreement No. 231287 (SSPNet). The work of M. Pantic was supported in part by the European Research Council under the ERC Starting Grant Agreement No. ERC-2007-StG-203143 (MAHNOB). The work of I. Patras was supported in part by the EPSRC project EP/G033935/1. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jenq-Neng Hwang.

A. Oikonomopoulos is with the Department of Computing, Imperial College London, London SW7 2AZ, U.K. (e-mail: aoikonom@doc.ic.ac.uk).

I. Patras is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K. (e-mail: i.patras@elec.qmul.ac.uk).

M. Pantic is with the Department of Computing, Imperial College London, London, U.K., and also with the Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands (e-mail: m.pantic@imperial.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2076821

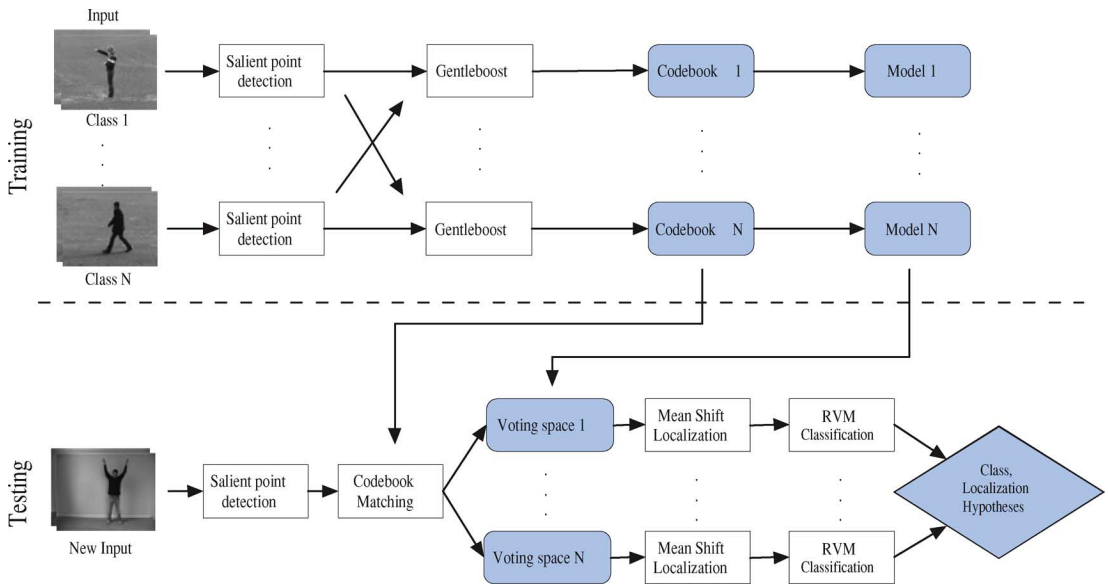


Fig. 1. Overview of the proposed approach.

Oikonomopoulos *et al.* [23] create their codebook by extracting sets of descriptors on B-Spline surfaces, fitted around detected spatiotemporal salient points.

Despite their success in object [11], [24] and scene [12] classification, 'bag of words' models are not particularly suit for localization, since, by using histograms, the information concerning the spatiotemporal arrangement of the descriptors is lost. Recently, a number of different methods have been proposed in order to deal with this issue. Leibe *et al.* [25] propose an implicit shape model for object detection, consisting of a codebook of visual words in which the relative position of each word with respect to the object center is maintained. A similar method using edge fragments is proposed by Opelt *et al.* [26]. In [27], a similar voting scheme is implemented for activity recognition and localization. The latter, however, is restricted only to the spatial localization of the subjects at each frame. Sivic *et al.* [28] propose the use of doublet codewords, while Boiman and Irani [29] propose a matching method based on feature ensembles in order to detect irregular scenes. A similar method, using constellations of static and dynamic feature collections is presented in [30]. Areas in images(videos) that share similar geometric properties and similar spatio(temporal) layouts are matched in [31], using a self similarity descriptor and the algorithm of [29]. A similar method is presented in [32], where a Self Similarity Matrix (SSM) is created for human activity recognition. Finally, Gilbert *et al.* [33] use data mining techniques in order to recover similar feature clusters from the training database, and detect activities in the presence of camera motion, occlusion and background clutter.

In this paper, we extend the work of Leibe *et al.* [25] by proposing a voting scheme in the space-time domain that allows both the temporal and spatial localization of activities. Our method uses an implicit representation of the spatiotemporal shape of an activity that relies on the spatiotemporal localization of ensembles of spatiotemporal features. The latter are localized around spatiotemporal salient points that are detected using the method described in [8]. We compare feature ensembles using a modified star graph model that is similar to the one proposed in [29], but compensates for scale changes using the scales of the features within each ensemble. We use boosting in order to create codebooks of characteristic ensembles for each class. Subsequently, we match the selected codewords with the training sequences of the respective class, and store the spatiotemporal positions at which each codeword is activated. This is performed with respect to a set of reference points, (e.g., the center of the torso and the lower bound of the subject) and with respect to the start/end of the action instance. In this way, we create class-specific spatiotemporal models, that encode the spatiotemporal positions at which each codeword is activated in the training set. During testing, each activated codeword casts probabilistic votes to the location in time where the activity starts and ends, as well as towards the location of the utilized reference points in space. In this way a set of class-specific voting spaces is created. We use Mean Shift [34] at each voting space in order to extract the most probable hypotheses concerning the spatiotemporal extend of the activities. Each hypothesis is subsequently verified by performing action category classification with a Relevance Vector Machine (RVM) [35]. A flowchart of the proposed method is depicted in Fig. 1, while an overview of the proposed spatiotemporal voting process is depicted in Fig. 2.

Compared to our previous work on human activity localization and recognition [36], the proposed framework utilizes feature ensembles which can be seen as a generalization of the codeword pairs that were used in [36]. Moreover, temporal votes are cast jointly for the start and end frames of the action instance, making hypotheses extraction a trivial task (i.e., using mean shift mode estimation). By contrast, in [36], temporal votes were cast for each phase of the action, and a Radon transform was utilized in order to extract each hypothesis. Finally, by verifying each hypothesis against all class-specific models, and by utilizing an RVM classification scheme, we managed to improve the classification accuracy of the proposed method compared to [36] for the same datasets.

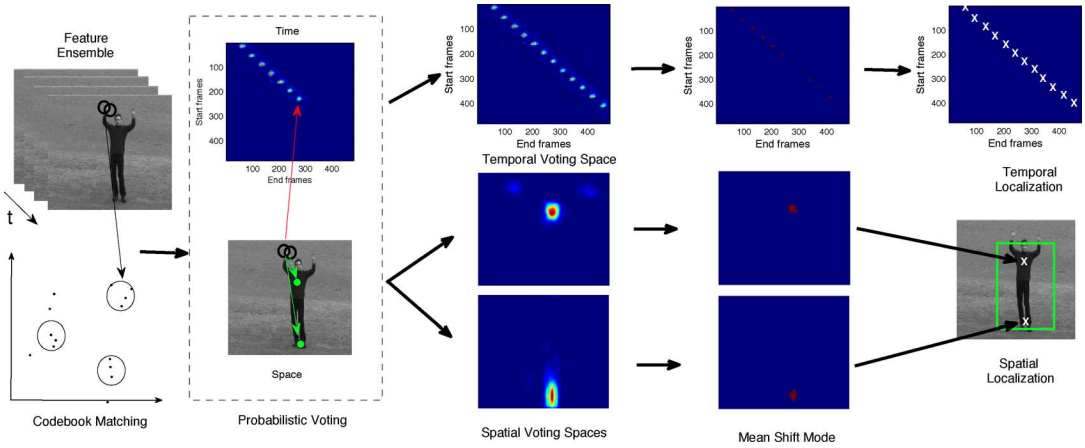


Fig. 2. Overview of the spatiotemporal voting process. Activated codewords cast spatial and temporal votes with respect to the center and spatial lower bound of the subject and the start/end frame of the action instance. Temporal votes for candidate start/end positions are cast jointly. Local maximums in the spatial and temporal voting spaces are extracted using mean shift and provide estimates for the position of a reference point in each frame of the test sequence and the temporal boundaries of an action instance respectively.

The main contributions of the proposed method are as follows:

- We propose an extension in time of the implicit shape model of Leibe *et al.* [25]. This leads to the creation of a spatiotemporal shape model, which allows us to perform localization both in space and in time.
- We propose to use feature ensembles in the proposed model, instead of single features.
- Through the use of boosting we create discriminative class-specific codebooks, where each codeword is a feature ensemble. This is in contrast to the work in [25], where no feature selection takes place. Furthermore, we propose a novel weighting scheme, in which votes from ensembles that are informative (i.e., they are characteristic of the phase of the action) are favored, while votes from ensembles that are commonly activated (i.e., they are activated in many phases of the action) are suppressed.
- Since spatiotemporal votes are accumulated from each observed ensemble in the test set, the proposed method effectively deals with occlusion, as long as a portion of the action is visible. Moreover, the use of class-specific codebooks and spatiotemporal models in a voting framework enables us to deal with the presence of dynamic background and with activities that occur simultaneously.

We demonstrate the effectiveness of our method by presenting experimental results in three different datasets, namely the KTH [37], HoHa [18] and the robustness dataset of [38]. Furthermore, we present results on synthetic and real sequences that have a significant amount of clutter and occlusion.

The remainder of this paper is organized as follows. In Section II we present our approach. That is, the creation of our spatiotemporal models for each class and the way they are used in order to perform localization and recognition. Section III includes our experimental results, and finally, Section IV concludes the paper.

II. Spatiotemporal Voting

We propose to use a probabilistic voting framework in order to spatiotemporally localize human activities. This framework, described in Section II-D, is based on class-specific codebooks of feature ensembles, where each feature is a vector of optical flow and spatial gradient descriptors. We describe the utilized feature extraction process in Section II-A, while Section II-B describes how these features are combined into ensembles and how ensembles are compared to each other. Each class-specific codebook is created using a feature selection process based on boosting, which selects a set of discriminative ensembles for each class. Each codebook is associated with a class-specific spatiotemporal localization model, which encodes the spatiotemporal locations and scales at which each codeword is activated in the training set. This process is described in Section II-C. During testing, each activated codeword casts spatiotemporal probabilistic votes, according to the information that was stored during training. Subsequently, mean shift is used in order to extract the most probable hypotheses concerning the spatiotemporal localization of an activity. Each hypothesis is then classified using Relevance Vector Machines. This process is described in Section II-F.

A. Features

The features that we use in this work consist of a combination of optical flow and spatial gradient descriptors, extracted around automatically detected spatiotemporal salient points [8]. However, the proposed framework can be utilized with any kind of local descriptors. In order to achieve robustness against camera motion, we detect the salient points on the filtered version of the optical flow field. More specifically, we locally subtract the median of the optical flow within a small spatial window. Alternatively, a global method, like an affine model, can be applied in order to compensate for the motion of the camera.

Let us denote with $N_c(\boldsymbol{s},\boldsymbol{v})$ the set of optical flow vectors that lie within a cylindrical neighborhood of scale $\boldsymbol{s}=(s,d)$, centered at location $\boldsymbol{v}=(x,y,t)$ of the motion compensated optical flow field of the input image sequence, where s,d denote the spatial and temporal scale of the neighborhood. In order to detect our salient points, we initially calculate the signal entropy $H_D(\boldsymbol{s},\boldsymbol{v})$ within the neighborhood $N_c(\boldsymbol{s},\boldsymbol{v})$:

$$H_D(\boldsymbol{s},\boldsymbol{v}) = -\int_{q\in D} p_D(q,\boldsymbol{s},\boldsymbol{v})\log_2 p_D(q,\boldsymbol{s},\boldsymbol{v})dq \qquad (1)$$

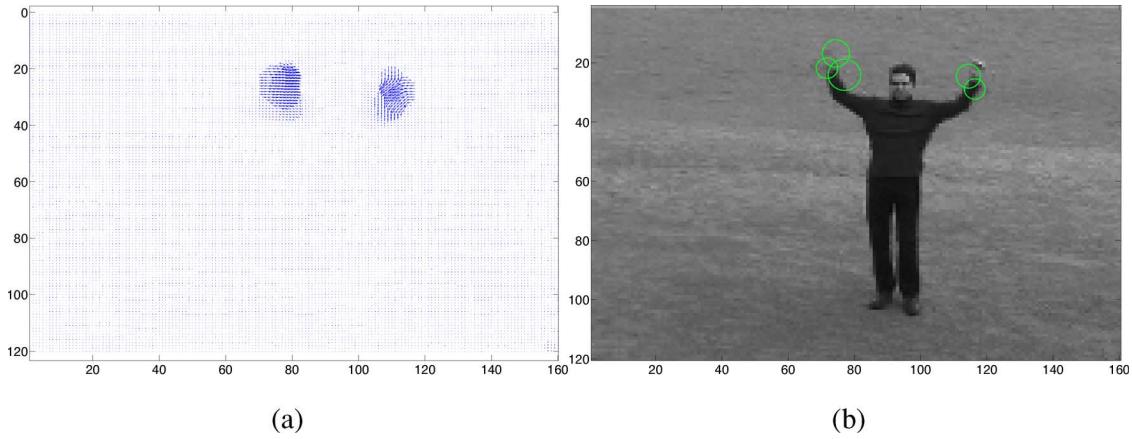


Fig. 3. (a) Estimated optical flow field for an instance of a *handwaving* action and (b) a subset of detected salient points.

where $p_D(q, \boldsymbol{s}, \boldsymbol{v})$ is the probability density of the signal histogram as a function of scale $\boldsymbol{s}$ and position $\boldsymbol{v}$. By q we denote the signal value and by D the set of all signal values. As has been mentioned before, in this work we use motion compensated optical flow vectors as signal values. We use the histogram method to approximate the probability density $p_D(q, \boldsymbol{s}, \boldsymbol{v})$. Alternatively, $p_D(q, \boldsymbol{s}, \boldsymbol{v})$ can be estimated using Parzen density estimation or any other density estimation technique. A salient point is detected at the scales for which the signal entropy is locally maximized, defined by

$$\hat{S}_p = \left\{ \boldsymbol{s} : \frac{\partial H_D(\boldsymbol{s}, \boldsymbol{v})}{\partial s} = 0 \bigwedge \frac{\partial H_D(\boldsymbol{s}, \boldsymbol{v})}{\partial d} = 0 \bigwedge \frac{\partial^2 H_D(\boldsymbol{s}, \boldsymbol{v})}{\partial s^2} < 0 \bigwedge \frac{\partial^2 H_D(\boldsymbol{s}, \boldsymbol{v})}{\partial d^2} < 0 \right\}. \quad (2)$$

A subset of the salient points detected on a frame of a *handwaving* sequence is shown in Fig. 3(b). We should note that for the detection of the depicted points, contribute a number of frames before and after the one shown in Fig. 3(a).

We use the algorithm in [39] for computing the optical flow, due to its robustness to motion discontinuities and to outliers to the optical flow equation. We use a C++ implementation of the algorithm, which is implemented in a multiscale fashion and runs at ∼2 frames per second for 120×160 pixel images on a 2 GHz Intel Centrino with 1 GB memory. There are recent real-time dense optical flow algorithms which would be more appropriate if we aimed for an optimized implementation. However, we felt that this was beyond the scope of our research. In order to form our descriptors, we take into account the optical flow and spatial gradient vectors that fall within the area of support of each salient point. This area is defined by the spatiotemporal scale $(\boldsymbol{s})$ at which each salient point is detected. Using their horizontal and vertical components, we convert these vectors into angles and bin them into histograms using a bin size of 10 degrees.

B. Feature Ensemble Similarity

We use ensembles of spatiotemporal features instead of single features in order to increase the spatiotemporal specificity of the proposed method. By doing so, sets of features that have similar spatiotemporal configuration between the training and test sets are matched. We form ensembles by sampling individual features as seeds and subsequently taking into account their $N - 1$ nearest neighbors. We discard points that have a significant degree of overlap with the seed. In our implementation, two points have a significant degree of overlap if their normalized Euclidean distance with respect to their spatiotemporal scale is smaller than a specific threshold.

Let $e_d = (c_d, \{v_d^i, l_d^i\}_{i=1\ldots\mathcal{M}})$ be an ensemble in the database consisting of $\mathcal{M}$ features, where c_d is the spatiotemporal center of the ensemble, and v_d^i, l_d^i are, respectively, the descriptor vector and the spatiotemporal location of the i^{th} feature. In this work we used 5 features for each ensemble, that is, $\mathcal{M} = 5$. We calculate the similarity between ensembles using a modification of the star graph model of [29]. More specifically, we model the joint probability $P(e_d, e_q)$ between the database ensemble e_d and the query ensemble e_q proportional to

$$P(e_d, e_q) \propto P\left(c_d, v_d^1, \ldots, l_d^1, \ldots, c_q, v_q^1, \ldots, l_q^1, \ldots\right). \quad (3)$$

The likelihood in (3) can be factored as

$$P\left(c_d, v_d^1, \ldots, l_d^1, \ldots, c_q, v_q^1, \ldots, l_q^1, \ldots\right) = \alpha \prod_i \max_j \left(P\left(l_q^j | l_d^i, c_d, c_q\right) P\left(v_q^j | v_d^i\right)\right) P\left(v_d^i | l_d^i\right). \quad (4)$$

The first term in the maximum in (4), that is, $P(l_q^j | l_d^i, c_d, c_q)$, expresses the similarity in the topology of the ensembles, and the second term expresses the similarity in their descriptor values. Consequently, each feature i of the ensemble e_d is matched to the feature j of the ensemble e_q with the maximum similarity in descriptor value and relative location within the ensemble. We model the first term as follows:

$$P\left(l_q^j | l_d^i, c_q, c_d\right) = z_1^{-1} \exp\Big(-\left(\left(l_q^j - c_q\right) S_q^j - \left(l_d^i - c_d\right) S_d^i\right)^T \times \mathcal{S}^{-1} \left(\left(l_q^j - c_q\right) S_q^j - \left(l_d^i - c_d\right) S_d^i\right)\Big) \quad (5)$$

where z_1 is a normalization term, and $\mathcal{S}$ is a fixed covariance matrix controlling the allowable deviations in the relative feature locations. Due to the low resolution of the utilized datasets,

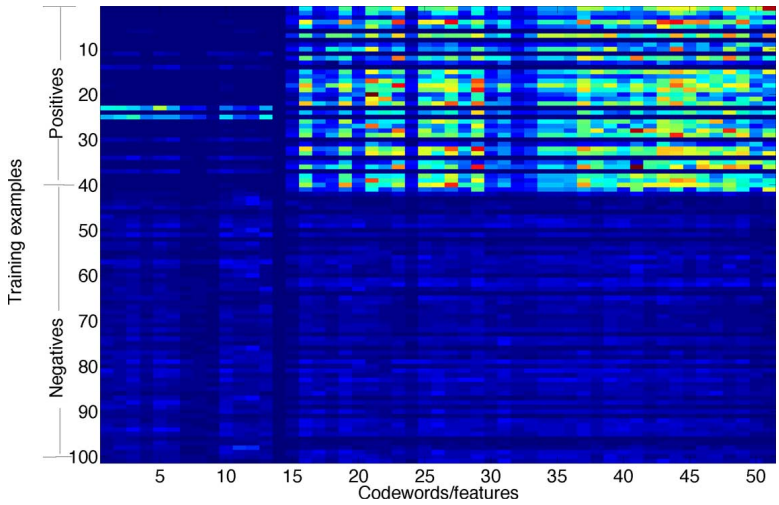


Fig. 4. Visualization of a feature selection matrix. In this example, selection is performed from 50 features, using 40 positive and 60 negative examples.

$\mathcal{S}$ was fixed in such a way so that the maximum allowable deviation to be around 5 pixels. However, for image sequences of higher resolution, larger deviations can be tolerated. Due to the use of the relative location of each feature with respect to the spatiotemporal center of the ensemble that the feature belongs to, the expression in (4) is invariant to the translational motion of the subject. Finally, S_d^i, S_q^j are diagonal matrices containing the inverse spatiotemporal scales of the points located at locations l_d^i, l_q^j respectively. That is,

$$S^i = \text{diag}((\sigma_i, \sigma_i, \tau_i)^{-1}) \tag{6}$$

where σ_i, τ_i are the spatial and temporal scales of the i^{th} feature. By normalizing the distance between the individual features and the ensemble center, we achieve invariance to scaling variations. We model the second term in the maximum in (4), that is, $P(v_j^q|v_i^d)$, as follows:

$$P\left(v_q^j|v_d^i\right) \propto z_2^{-1} \exp\left(-z_3^{-1} D\left(v_q^j, v_d^i\right)\right) \tag{7}$$

where z_2, z_3 are normalization terms, and $D(\cdot,\cdot)$ is the χ^2 distance. The latter is a popular measure for comparing histograms, and is essentially a weighted Euclidean distance. More specifically, in the χ^2 distance, the square distance between the bin entries of two histograms is weighted by the inverse sum of the bin entries. Weighting compresses the variation in the components' values, by assigning less weight to components with large values.

The last term in (4) expresses the relations within the ensemble e_d, i.e., the relation between the feature descriptor and its location. Similar to [29], we model this term using examples from the database:

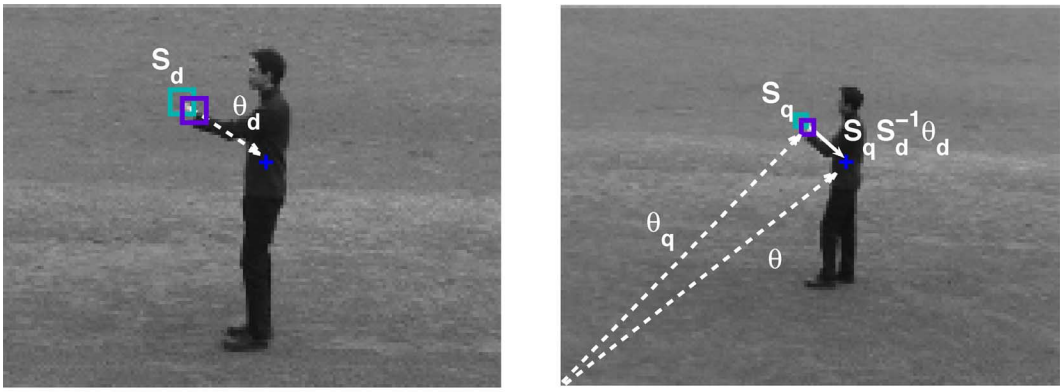
$$P(v_d|l_d) = \begin{cases} 1, & (v_d, l_d) \in DB \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where v_d, l_d are, respectively, an arbitrary descriptor and location. That is, $P(v_d|l_d)$ is equal to one if and only if the feature descriptor v_d appears in location l_d in the database.

C. Feature Selection and Codebook Creation

We use Gentleboost [40] in order to select characteristic ensembles that will form the codewords for each class-specific codebook $\boldsymbol{E}$. Our goal is to select feature ensembles that appear with high likelihood in the positive and with low likelihood in the negative examples. Let us assume that the examples of the positive class consist in total of $\mathcal{N}$ ensembles. To perform feature selection for this class, we sample at random $\mathcal{L}$ (e.g., 5000) ensembles from the initial population of $\mathcal{N}$ ensembles. Using (3), we match the sampled $\mathcal{L}$ ensembles to the remaining $\mathcal{N} - \mathcal{L}$ ensembles of the positive set and the ones in the negative set. The latter consists of the ensembles that belong to all available classes other than the positive class. By performing this procedure, we expect that ensembles characteristic of the positive set will have a high likelihood of match to ensembles in the examples belonging to that set, and a low likelihood of match to ensembles in the examples belonging to all other classes (i.e., the negative set). Since each sequence in the training set comprises of a few thousand of features, we keep the N' best matches from each one, in order to make the selection tractable. This procedure results in $N'M_p$ positive training vectors of dimension $1 \times \mathcal{L}$ and $N'M_n$ negative training vectors of the same dimension, where M_p and M_n are the total number of the positive and negative image sequences in the training set respectively. Using these training vectors, Gentleboost selects a set of characteristic ensembles for the positive class. This set is a subset of the initial set of $\mathcal{L}$ ensembles. By performing this process for each class we end up with a set of characteristic ensembles for each class. An example of the training vectors that are created is depicted in Fig. 4. As can be seen, several features, namely the first 15, are not characteristic of the class, since their likelihood of match in both positive and negative examples is low (dark areas in the figure).

We subsequently use each class-specific codebook in order to create a spatiotemporal model for each class. Each model is created by accumulating information over the spatiotemporal positions at which each codeword is activated in the training set. For each class-specific codebook, we iterate through the training sequences that belong to the same class as the codebook and activate each ensemble e_d whose likelihood of match is above a threshold. In this work we used a threshold of 0.1, which translates to a likelihood value of at least 0.8 for each of the 5 features in each ensemble in terms of their topology and descriptor similarity with the corresponding features in the codeword ensem-



(a) (b)

Fig. 5. Voting example. (a) During training, the position $\boldsymbol{\theta}_d$ and average spatiotemporal scale $\boldsymbol{S}_d$ of the activated ensemble is stored with respect to one or more reference points (e.g., the center of the subject, marked with the blue cross). (b) During testing, votes are cast using the stored $\boldsymbol{\theta}_d$ values, normalized by $\boldsymbol{S}_q\boldsymbol{S}_d^{-1}$ in order to account for scale changes. (Best viewed in color.).

bles. Subsequently, we store all the positions $\boldsymbol{\theta}_d$ at which each e_d was activated relative to a set of reference points in space and time, and a diagonal matrix $\boldsymbol{S_d}$ containing the spatiotemporal scale at which codeword ensemble e_d was activated. The scale is taken as the average of the scales of the features that constitute e_d. An illustration of this process is depicted in Fig. 5(a). During testing, the values $\{\boldsymbol{\theta_d}\}, \{\boldsymbol{S_d}\}$ are used in order to cast votes concerning the spatiotemporal extend of an activity in the test set, given that the codeword e_d is activated. This process is explained in Section II-D.

If n is the number of patches in each ensemble of the codebook, M is the number of codewords and N is the total number of patches in a test sequence, then the ensemble matching process is of complexity $O(nMN)$. For ensembles consisting of 5 patches, typical codebook size of 500 words and for a typical size of 20 K patches in a test sequence consisting of approximately 500 frames, this process took about 15 min on a 2 GHz Intel Centrino with 1 GB memory. However, the implementation of a progressive elimination algorithm similar to the one in [29] can significantly speed up the process to linear time with respect to N.

D. Probabilistic Framework

Given a codebook and a spatiotemporal localization model for each class, our goal is to estimate a set of parameters $\{\boldsymbol{\theta_s}\}, \{\boldsymbol{\theta_t}\}$ that define, respectively, the location in space-time of a human activity depicted in an unknown image sequence. We denote with $\boldsymbol{\theta_s}$, the location of a set of reference points positioned on the subject, that define its location at each frame of the image sequence. Furthermore, we denote with $\boldsymbol{\theta_t}$, the temporal extend of the activity, that is, the frame at which it starts and the frame at which it ends.

In order to acquire a probability distribution over $\{\boldsymbol{\theta_s}\}$ and $\{\boldsymbol{\theta_t}\}$, we propose the use of a spatiotemporal voting scheme, which is an extension in time of the implicit shape model proposed by Leibe *et al.* [25]. In the proposed model, an activated codeword in the test set casts probabilistic votes for possible values of $\boldsymbol{\theta_s}, \boldsymbol{\theta_t}$, according to information stored during training. We use ensembles of spatiotemporal features as codewords, modeled using the star-graph model of [29]. In the following, and without loss of generality, we drop the subscripts on the $\boldsymbol{\theta_s}, \boldsymbol{\theta_t}$ parameters, and describe the utilized probabilistic framework for the generalized parameter $\boldsymbol{\theta}$. The probability of $\boldsymbol{\theta}$ can be formulated as

$$P(\boldsymbol{\theta}) = \sum_{q=1}^{Q} P(\boldsymbol{\theta}|e_q)P(e_q) \tag{9}$$

where $\{e_q\}$ is the set of observed ensembles and $P(e_q)$ is the prior probability of observing e_q. In the absence of prior knowledge, we model this probability as a uniform distribution, i.e., $P(e_q) = 1/Q$, where Q is the number of observed ensembles. Each observed ensemble e_q is matched against each codeword e_d from the codebook $\boldsymbol{E}$, which was created according to the procedure of Section II-C. By marginalizing $P(\boldsymbol{\theta}|e_q)$ on $e_d \in \boldsymbol{E}$ we get:

$$P(\boldsymbol{\theta}|e_q) = \sum_{e_d \in \boldsymbol{E}} P(\boldsymbol{\theta}|e_d, e_q)P(e_d|e_q). \tag{10}$$

The term $P(e_d|e_q)$ expresses the likelihood of match between codeword e_d and the observed ensemble e_q, and is calculated according to the process of Section II-B. After matching e_q to e_d, we consider $P(\boldsymbol{\theta}|e_d, e_q)$ as being independent of e_q. $P(\boldsymbol{\theta}|e_d)$ expresses the probabilistic vote on location $\boldsymbol{\theta}$ given that the activated codebook entry is e_d. Let us denote with $\{\boldsymbol{\theta}_d\}$ the set of the votes associated with the activated codebook entry e_d. These votes express the spatiotemporal positions at which e_d was observed in the training set, relatively to the subject/action reference system, and are learned, during training, according to the process of Section II-C. $P(\boldsymbol{\theta}|e_d)$ can be modeled as

$$P(\boldsymbol{\theta}|e_d) = w_d \sum_{\boldsymbol{\theta}_d} P(\boldsymbol{\theta}|\boldsymbol{\theta}_d, e_d)P(\boldsymbol{\theta}_d|e_d) \tag{11}$$

where w_d is a weight learned during training, which expresses how important the ensemble e_d is, in accurately localizing the action in space and time. The way w_d is calculated is described

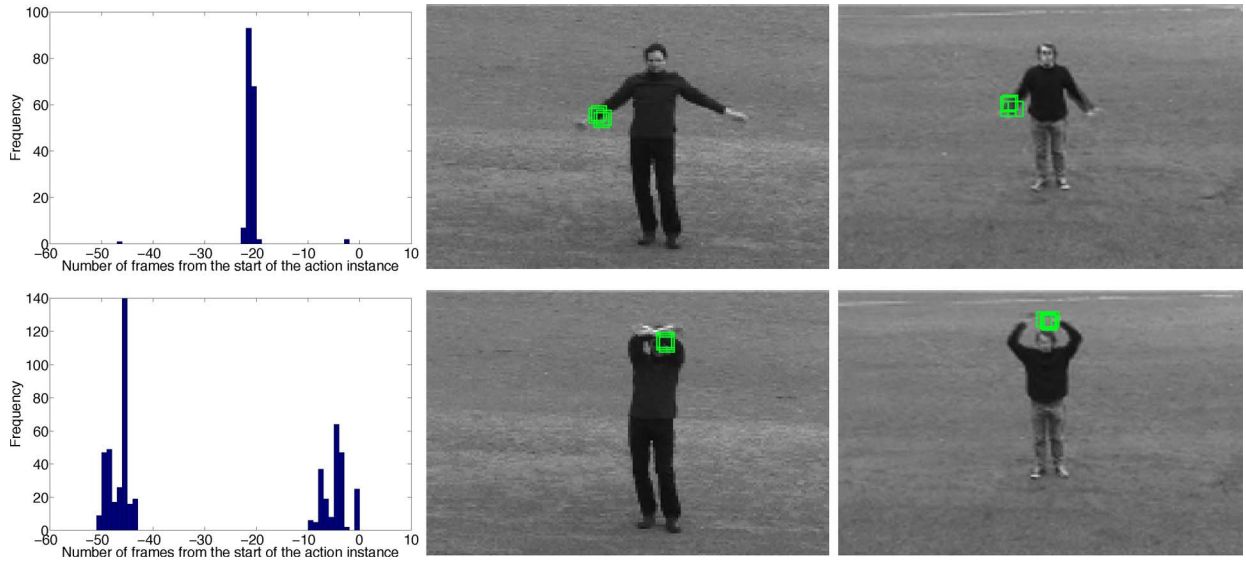


Fig. 6. Ensemble weighting example for the temporal case. Ensembles that are activated at a specific phase of the action receive a large weight (top row). Conversely, ensembles that are activated at more than one instances receive a smaller weight (bottom row).

in Section II-E. The first term of the summation in (11) is independent of e_d, since votes are cast using the $\boldsymbol{\theta}_d$ values. Votes are cast according to the following equation:

$$\boldsymbol{\theta} = \boldsymbol{\theta}_q + \boldsymbol{S}_q \boldsymbol{S}_d^{-1} \boldsymbol{\theta}_d \tag{12}$$

where $\boldsymbol{S}_q, \boldsymbol{S}_d$ are diagonal matrices containing the scale of the e_q, e_d ensembles respectively and $\boldsymbol{\theta}_q$ denotes the location of the observed ensemble e_q. The concept of (12) for the spatial case is depicted in Fig. 5(b), where the position of the center is given by the vector addition of $\boldsymbol{\theta_q}$, the position of the observed ensemble e_q, and $\boldsymbol{S}_q \boldsymbol{S}_d^{-1} \boldsymbol{\theta}_d$. The latter is the scale-normalized position at which the codeword ensemble e_d (with which e_q is matched) was observed in the training set with respect to the center of the subject. By normalizing with $\boldsymbol{S}_q \boldsymbol{S}_d^{-1}$ we achieve invariance to scale differences between the observed and the activated ensemble codeword. $\boldsymbol{S}_d, \boldsymbol{S}_q$ are calculated as the average spatiotemporal scales of the features that consist the ensembles. Since we only use the stored $\boldsymbol{\theta}_d$ and $\boldsymbol{S}_d$ values for casting our votes, we can model $P(\boldsymbol{\theta}|\boldsymbol{\theta}_d)$ as

$$P(\boldsymbol{\theta}|\boldsymbol{\theta_d}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_q - \boldsymbol{S}_d \boldsymbol{S}_q^{-1} \boldsymbol{\theta}_d) \tag{13}$$

where $\delta(.)$ is the Dirac delta function. Finally, we model $P(\boldsymbol{\theta}_d|e_d)$ using a uniform distribution, that is, $P(\boldsymbol{\theta}_d|e_d) = 1/V$, where V is the number of $\boldsymbol{\theta}_d$ values associated with e_d. Alternatively, this probability can be modeled using a density estimation method. That is, a larger probability can be assigned to the θ_d values that were more commonly observed.

The probabilistic framework that is described in this section applies for both spatial and temporal votes. For the spatial case, $\boldsymbol{S}_q, \boldsymbol{S}_d$ contain the spatial scales of the test and database ensembles respectively, while $\boldsymbol{\theta}_q$ denotes the spatial location of the observed ensemble in absolute coordinates. Therefore, $\boldsymbol{\theta}$ encodes the displacement from the center and lower bound of the subject. Similarly for the temporal case, $\boldsymbol{S}_q, \boldsymbol{S}_d$ contain temporal scales, while $\boldsymbol{\theta}_q$ denotes the temporal location of the observed ensemble with respect to either the start or the end of the image sequence. Therefore, $\boldsymbol{\theta}$ can encode two scalar temporal offsets, one to the start and one to the end of the action. We should note, finally, that in the proposed framework spatial voting is performed first, followed by voting in time. While in spatial voting we take into account the votes from all of the activated codewords, in temporal voting we take into account the votes of activated codewords that additionally contributed to the most probable spatial center. This process is described in more detail in Section II-F.

The use of class-specific codebooks/spatiotemporal localization models enables us to deal with the presence of dynamic background and multiple activities in the test set. The purpose of such models is to search for activities of a specific class in an unknown image sequence. Ideally, observed ensembles localized around activities of different class, or around any other kind of motion in the background will not match well with the codewords in the codebook, and therefore their votes according to the corresponding model will be assigned a very small probability. This is evident from (10). Finally, the use of a voting framework for localization increases the robustness of proposed method to partial occlusion. Since votes are cast from each observed ensemble in the test set, a good estimate can be acquired, as long as a good portion of the activity is still visible.

E. Localization Accuracy

In this section we will describe a methodology to learn w_d, that is, the weight that is used in (11) and expresses the importance of ensemble e_d in accurately localizing an activity in space and time. More specifically, we would like to favor votes from ensembles that are characteristic of the location at which they appear within the action instance and suppress votes from ensembles that are activated at many locations in the action instance. Let us denote by $P_d(l)$ the probability that the ensemble e_d was activated at location l. This distribution is learned during training. Then, the votes of each ensemble e_d are weighted as follows:

$$w_d = e^{-\int P_d(l) \log P_d(l) dl} \tag{14}$$

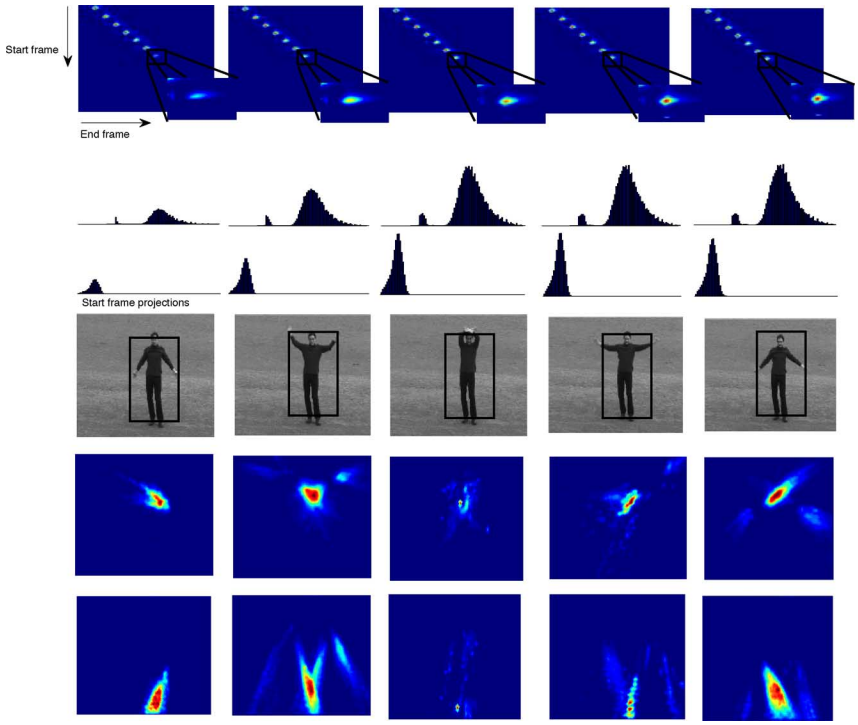


Fig. 7. Illustration of the spatiotemporal voting scheme. First row: evolution of the temporal voting space. Second, third row: Start/end frame projections along lines passing from a local maximum. Evidence is accumulated as time progresses, resulting in more votes at the most probable positions. Fifth, sixth row: Spatial voting spaces, showing the most probable positions of the center and lower bound of the subject. Fourth row: Fitted bounding boxes resulting from the maximum responses in the spatial voting spaces.

The exponent in (14) is the Shannon entropy of the distribution of the votes that the ensemble e_d casts. Ensembles that are only activated at specific parts of the action will have a distribution with low entropy, since their votes will be concentrated in a few values, resulting in a large weight. An example is given in Fig. 6 for a *handwaving* action. More specifically, the ensemble in Fig. 6(a) is activated almost exclusively around the middle of the action, and describes the upward motion of the right hand of the subject. By contrast, the ensemble depicted in Fig. 6(b) is activated both at the start and at the end of the action, as shown in the histogram of votes at the top of the figure, and describes the motion were the hands of the subject are joined around the head. In the latter case, the votes of the corresponding codeword will receive a lower weight that the codeword in the former case. Let us note that since both codewords were selected by the process of Section II-C they are both considered informative for the class. However, since the first codeword provides more clear evidence concerning the phase of the action, it will receive a larger weight during the voting process.

F. Activity Detection

The goal of activity detection is to spatiotemporally localize and classify an activity depicted in an unsegmented image sequence. Using the probabilistic framework of Section II-D, the proposed algorithm initially casts spatial votes according to the information stored in the training stage. Since the class of the human activity is unknown, this procedure is performed for each class-specific codebook/spatiotemporal localization model. We use Mean Shift Mode [34] in order to localize the most probable centers and lower bounds of the subjects at each frame of the image sequence. Given the sparsity of each voting space, the computational cost of the application of the mean shift algorithm is negligible. In addition, we apply a Kalman filter [41] using as observations the raw estimates of these points as they are given by mean shift mode. Kalman filtering has the effect of smoothing the estimates of the points from frame to frame, and increases robustness against outliers in the mean shift mode estimation. Using the estimates of these two points, we are able to fit a bounding box around the subject, as depicted in Fig. 7. To reduce the influence of clutter, we cast temporal votes by only taking into account the ensembles that contributed to the most probable center in the spatial voting space. Finally, using Mean Shift Mode estimation on the resulting temporal voting spaces, the most probable hypotheses concerning the temporal extend of the activity are extracted. An example is depicted in the top row of Fig. 7, where the y axis indicates the frame at which the instance starts and the x axis the frame at which it ends. Since the votes for the start/end frames are cast jointly, most votes are concentrated above the main diagonal, reflecting the fact that the start frame position must temporally precede the end frame position. To illustrate the evolution of the temporal votes as time progresses, we also depict, in the same figure, 1-D projections of the temporal voting space along horizontal and vertical lines that pass through one of the local maximums. As shown in the figure, as time progresses, more evidence is accumulated concerning the most probable position in time where the action instance starts and ends.

Depending on the voting space from which each hypothesis is extracted, a class label can be assigned directly to it. We perform instead a hypothesis verification stage. Let us denote with e_{tm} the maximum response of the m spatial voting space at frame t, as this is given by mean shift mode, where m denotes the class.

That is, each e_{tm} expresses the belief of the voting algorithm that the center of the subject is at a specific location at frame t for model m. Other points (i.e., the lower bound of the subject), or a combination of them can also be used for this purpose. Furthermore, let us denote an extracted hypothesis with F_{ij}, where i, j are the indexes of the frames at which, according to the hypothesis, the activity starts and ends respectively. Our hypothesis verification step relies on the calculation of the following measure:

$$R_{ijm} = \frac{1}{(j-i)} \sum_{t=i}^{j} e_{tm}. \tag{15}$$

That is, each R_{ijm} is the average sum of the mean shift output of the m spatial voting space, between frames i, j. Using R_{ijm}, we define a thin plate spline kernel for an RVM classification scheme:

$$\mathcal{K}_{ijm} = R_{ijm} \log R_{ijm}. \tag{16}$$

We train L different classifiers, in an one against all fashion. Each classifier outputs a conditional probability of class membership given the hypothesis, $P_m(l|F_{ij}), 1 \leq m \leq L$. Subsequently, each hypothesis F_{ij} is assigned to the class for which this conditional probability is maximized. That is,

$$\text{Class}(F_{ij}) = \arg\max_m (P_m(l|F_{ij})). \tag{17}$$

Note that we assign a label to each hypothesis that is extracted and not to the whole video. This is more sensible, since a video might contain activities of more than one class. Finally, let us note that since each hypothesis F_{ij} is extracted from a class-specific voting space it could be written as F_{ijm}, where m indicates the class of the voting space from which the hypothesis is extracted. However, since at this stage the class is not fixed and assigned by the RVM, we avoid such a notation.

III. Experimental Results

We use three different datasets in order to provide experimental evaluation of the proposed algorithm. Namely, the KTH [37], the Hollywood Human actions (HoHA) [18] and the robustness dataset of [38]. Furthermore, we present results on synthetic and real sequences in which there is a significant amount of clutter and occlusion. The KTH dataset contains 6 different actions; *boxing*, *hand-clapping*, *hand-waving*, *jogging*, *running*, and *walking*, performed by 25 subjects several times under different conditions. These include scale changes, indoors/outdoors recordings, and varying clothes. The main challenges in this dataset include small camera motion, noise in the otherwise uniform background, shadows, and large variability in the conduction of the activities by the subjects.

Containing video samples from 32 movies, the HoHA dataset is one of the most challenging ones in the area of activity recognition. Each sample is labeled according to one or more of 8 action classes: *AnswerPhone*, *GetOutOfCar*, *HandShake*, *HugPerson*, *Kiss*, *SitDown*, *SitUp*, *StandUp*. The main challenge of this dataset is the huge variability of the actions depicted, due to different view-points, cluttered and dynamic background and significant camera motion.

The sequences in the robustness dataset of [38] have non-uniform, static backgrounds, and include walking activities under varying conditions. These include different viewpoints and 11 'deformation' sequences, like walking with a dog. We use this dataset only for testing, while training is performed using the walking actions of the KTH dataset.

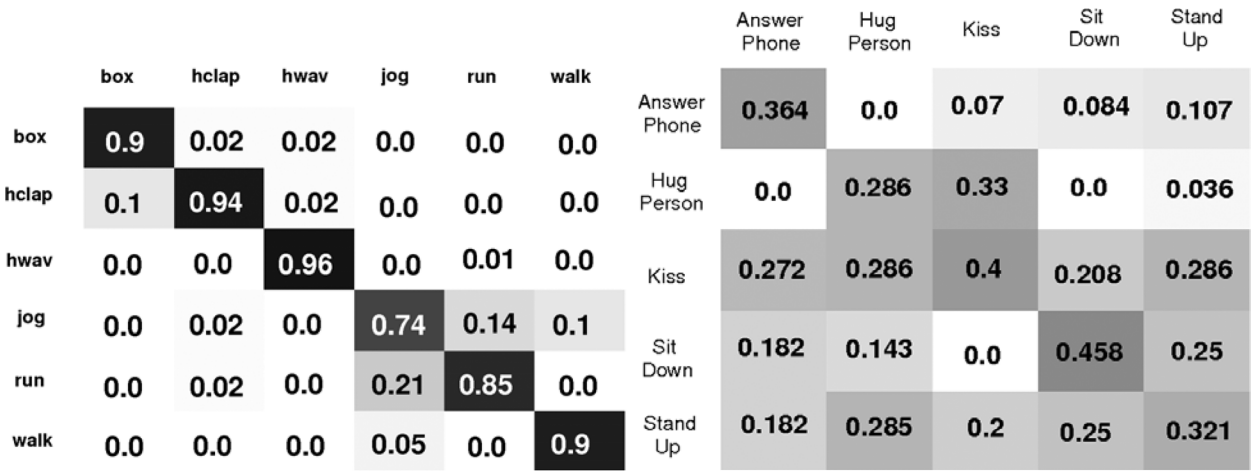
To test the performance of the proposed algorithm in the presence of occlusion, we selected 10 sequences per class from the KTH dataset, i.e., 10% of the data, and placed an artificial occluding bar of varying width in areas that are important for the recognition of that action, like, e.g., on the moving legs of subjects, in classes like *walking*. Finally, we use synthetic and real sequences in order to test the robustness of the proposed method against dynamic background, as well as to its ability to localize multiple activities in the same scene.

A. Training Set

We consider a single repetition of an activity as an action instance, like e.g., a single hand-clap or a single walking cycle. To create a training set, we manually select a subset of action instances for each class and we register them in space and time, by spatially resizing the selected instances so that the subjects in them have the same size. Moreover, we linearly stretch the selected instances so that the depicted actions in each class have the same duration. Finally, 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. We used 20 instances per class in order to create each class specific model for the KTH dataset. These consist of 5 subjects, each performing an activity 4 times. All instances that were performed by a specific subject were extracted by a single image sequence, which was not included in the test set. That leaves about 95 sequences per class for testing, i.e., 95% of the data. For the HoHa dataset, we used 10 sequences per class in order to create each model, due to the smaller number of videos compared to the KTH dataset.

B. Classification

We use activity instances pre-segmented in time in order to evaluate the classification accuracy of the proposed algorithm and compare it to the state of the art. We use the process of Section II-F in order to perform classification, where each hypothesis corresponds to a pre-segmented example. That is, we calculate, for each example, its similarity to each of the trained models according to (15) and use this similarity in order to define a kernel for the RVM, according to (16). Classification is performed in a leave-one-subject-out manner, using (17). That is, in order to classify an activity instance performed by a specific subject, we trained the RVM classifiers using all available instances apart from the ones performed by the subject in question. In Fig. 8(a), the confusion matrix for the KTH dataset is depicted. As can be seen from the figure, the largest degree of confusion is between the classes *jogging* and *running*. As noticed by Schuldt *et al.* [37], these confusions are in fact reasonable, since what appears to some people as running may appear to others as jogging and vice versa. The average recall rate achieved by the RVM classifier for the KTH dataset is 88%. By contrast, using just the measure of (15) and a 1-NN classifier, the average recall rate was about 75.2%. The largest improvement was noted on the *running* class, with an increase from 53% to 85% in the recall rate.



(a) (b)

Fig. 8. Confusion matrices for the (a) KTH and (b) HoHA datasets.

In Fig. 8(b), we present the confusion matrix for the HoHa dataset. Due to the small number of representative examples, we discard classes *GetOutOfCar*, *HandShake*, *SitUp*. Furthermore, due to the presence of several examples in which the lower bound of the subjects is not visible, we only used the subject centers as reference points for this dataset. It can be observed that there are several confusions between classes that are not very similar. The largest confusion, however, is between the classes *HugPerson* and *Kiss*, since both involve two persons coming progressively closer to each other.

We use a cross-dataset approach in order to acquire classification results on the robustness dataset of [38]. That is, we consider the latter only for testing, using the models that we created on the KTH dataset. Our algorithm was able to correctly classify 9 out of the 11 sequences of the deformed set and 6 out of the 10 sequences of the multi-view set, with all confusions being between the *walking* and *jogging* classes. While Blank *et al.* [38] report 100% recognition rate on this dataset, their training is based on the Weizmann dataset of human actions [38], which does not include the *jogging* class. By removing the *jogging* class from our classification process, our classification rate on this dataset also reaches 100%.

We present, in Table I, comparative classification results between the proposed method and several methods proposed in the literature. As can be seen from Table I, the classification results that we obtained outperform the ones in, e.g., [42], [37]. Furthermore, we achieve similar results as the ones reported in [43], [44], [17]. Compared to these works, we also provide the means for localization of the actions in space and time. Furthermore, we do not assume a stationary camera as these works do. Instead, by using filtered optical flow we minimize the effect of camera motion in the extracted features. Furthermore, we do not perform any preprocessing prior to feature detection, contrary to Fathi and Mori [45], who use stabilized sequences of cropped frames centered on the human figure. Similarly, Wong and Cipolla [46] temporally normalize their sequences to have similar length. Instead, we handle temporal variations by automatically detecting temporal scale in the spatiotemporal salient point detection step and by using this scale throughout our proposed algorithm. Finally, we do not perform any background subtraction before detecting our features, as opposed to [15], [44], who use a Gaussian Mixture Model (GMM) in order to identify foreground pixels. In the proposed method, we achieve a similar effect by detecting the spatiotemporal salient points at areas in which there is significant amount of motion, as described in [8].

TABLE I
COMPARISONS OF THE PROPOSED METHOD TO VARIOUS METHODS PROPOSED ELSEWHERE FOR THE KTH DATASET

Methods	Accuracy (%)
Our method	88.0
Ke et al. [42]	62.97
Schuldt et al. [37]	71.83
Ahmad and Lee [44]	88.3
Dollar et al. [6]	81.17
Wong and Cipolla [46]	86.7
Niebles et al. [20]	81.5
Fathi and Mori [45]	90.5
Jhuang et al. [15]	91.7
Rapantzikos et al. [43]	88.3
Ali and Shah [17]	87.7

C. Localization

1) Spatial Localization: In this section we evaluate the accuracy of the proposed algorithm in localizing a subject at each frame of an image sequence. Here, we assume, that the activity class that the subject is performing is given. Following the process of Section II-F, the proposed algorithm is able to provide an estimate of the subject center and lower bound for each frame of a sequence. To account for the smooth motion of the subjects, we apply a Kalman filter to the estimates of the subject location. The results achieved for each class of the KTH dataset are depicted in Fig. 9. Using just the raw estimates, our

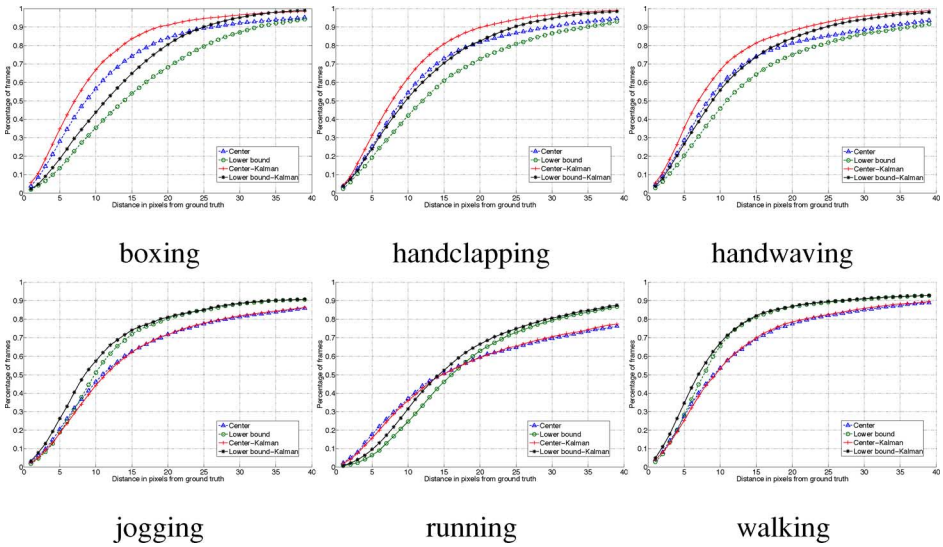


Fig. 9. Spatial localization results achieved for the subject center and lower bound, for each class of the KTH dataset. The increase in performance when applying a Kalman filter is more prominent for the *boxing*, *handclapping* and *handwaving* classes. x-axis: distance from ground truth annotation in pixels. y-axis: percentage of frames in the database at which the localization estimate's distance from the ground truth was less or equal to the values in the x-axis.

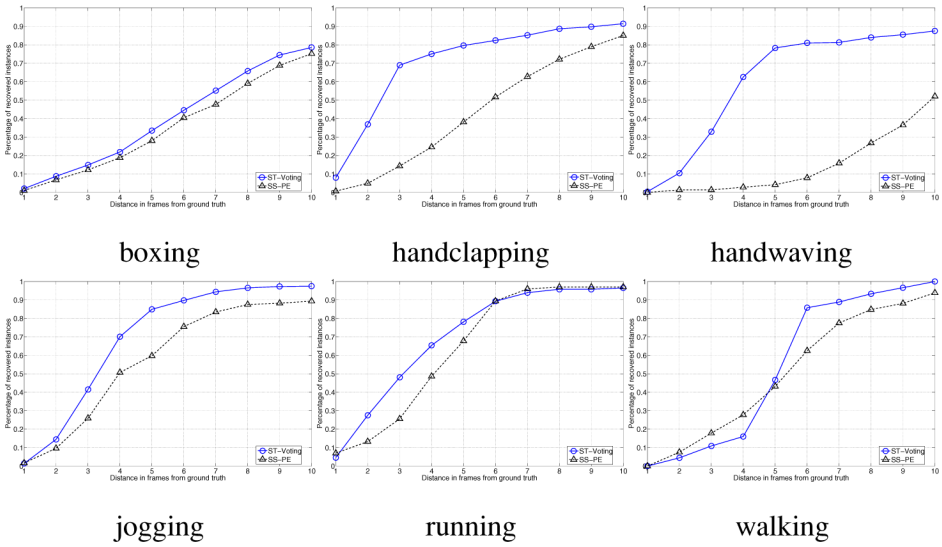


Fig. 10. Comparative temporal localization results for the 6 classes of the KTH dataset, between the proposed algorithm (ST-Voting) and the Self-Similarity with Progressive Elimination (SS-PE) algorithm of [31]. x-axis: distance from ground truth annotation in frames. y-axis: percentage of recovered instances.

algorithm is able to localize the center of the subject in 70% of all frames in the dataset on average, with the estimate's distance from the ground truth annotation being smaller or equal to 15 pixels. Given that the width of the subjects is on average 20 pixels, our estimate, in most cases, falls within its range. The worst performing class is *running*, which, for the same distance from the ground truth yields around 55% accuracy in the localization of the subject center. By applying a Kalman filter on the raw estimates, we achieve an increase in performance of about 10% for *boxing*, *handclapping* and *handwaving*, while there was a smaller increase in the performance for *jogging*, *running* and *walking*.

2) Temporal Localization: In this section we evaluate the accuracy of the proposed algorithm in localizing in time several instances of a known activity that occur in an image sequence. For this experiment, we apply the process of Section II-F, and compare each extracted hypothesis with the ground truth annotation. The latter was performed in such a way so that each annotated instance includes a single repetition of the activity, e.g., a single punch in *boxing*. Each extracted hypothesis specifies the frames in the image sequence at which the action instance starts and ends. The error of each hypothesis was calculated as the difference in frames between the ground truth annotation and the start/end frames specified by the hypothesis. In this way, we were able to construct Fig. 10, which plots the percentage of the recovered hypotheses as a function of this frame difference.

We compare these results with the ones acquired by [31]. More specifically, we compute self-similarity descriptors for all sequences in KTH and apply their progressive elimination algorithm to match a query to each sequence. Matching was performed using 5 query sequences per class from our training set and averaging the results. This gives us an estimate of the spatiotemporal extend of each recovered instance. This is similar to the hypothesis extraction process of our method, and is the

reason why we chose to perform comparison with the method of [31]. The localization accuracy achieved is depicted in Fig. 10. As can be seen from the figure, the results achieved are similar to the ones achieved by the algorithm of [31] for *boxing* and slightly better for *jogging* and *running*. For *handwaving* and *handclapping*, 70% of the extracted hypotheses are localized within 3 frames from the ground truth on average, in comparison to 15% achieved by [31].

D. Joint Localization and Recognition

In this section, we present experimental evaluation for localizing and classifying human activities that occur in an unsegmented image sequence, where both the localization and the class of the activities that occur in the sequence are unknown. Given an image sequence, each class-specific model created during training, results in a different voting space for this sequence. Using mean shift mode, a set of hypotheses is extracted from each voting space, and classified to a specific action category. Each hypothesis corresponds to an interval in time in which the activity takes place, and is assigned a weight, equal to the response in the voting space at the point at which the hypothesis was extracted. A low weight on a hypothesis means that the proposed algorithm does not have a strong belief on its validity. Therefore, by setting up a threshold on the weights, we can control which of the hypotheses are considered as being valid by the algorithm. By varying this threshold, we construct the ROC curves depicted in Fig. 11. Note that all curves are well above the main diagonal, meaning that regardless of the threshold value, the number of true positives is always larger than the number of false positives. Furthermore, the incompleteness of the depicted ROC curves reveals that a number of ground truth annotations are not detected by the algorithm. The reason is that, while the mean shift mode process is able to successfully extract the corresponding hypotheses, these are subsequently misclassified by the RVM. Therefore, the recall rate never reaches 100%. Such misclassifications occur either due to the inherent similarity between certain actions (e.g., between *running* and *jogging*) or due to the low values of the corresponding region of the voting space from which these hypotheses were extracted. Such low values can result from insufficient evidence (i.e., number of detected salient points) at certain parts of the image sequence (e.g., when the person is far, due to camera zoom).

E. Occlusions

We use synthetic image sequences to demonstrate the robustness of our method against occlusion, where we used vertical or horizontal bars to occlude parts of human activities, as depicted in Fig. 12. We performed our experiments using 10 sequences from each class, i.e., 10% of the data, with a variable bar width. To determine the effect of the occlusion in classification accuracy, we selected sequences that were correctly classified in the classification stage of Section III-B. Despite the occlusion, our algorithm was able to correctly classify all of the selected sequences. We present, in Fig. 13, average spatial localization results for all of the selected examples as a function of the degree of occlusion. The latter is defined as the ratio between the activity extend in space and the width of the occluding bar. Note that for actions like *handclapping*, the spatial activity extend only covers the moving hands of the subject. As can be seen from Fig. 13, our method is robust to relatively small amounts

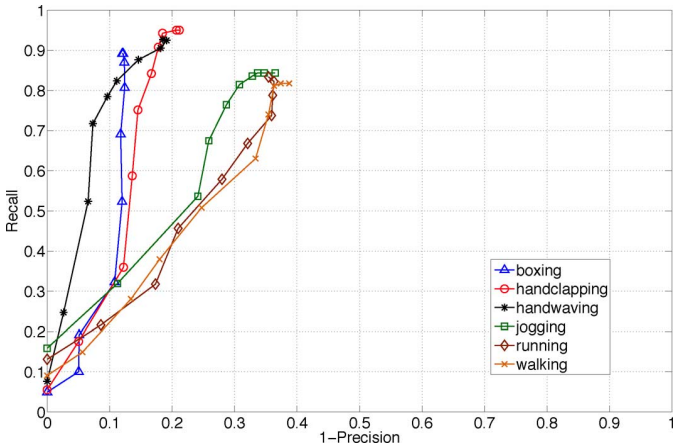


Fig. 11. Joint Localization and recognition: ROC curves corresponding to each class of the KTH dataset.

of occlusion. For 60% of occlusion, that is, the largest degree tested, there was a 20% drop in the localization accuracy of the subject center compared to no occlusion at all, with the estimate of the center being within a radius of 10 pixels from the ground truth annotation. However, our method behaves very well for smaller amounts of occlusion, with an average drop of about 10% in performance for a 35% degree of occlusion.

Finally, we performed experiments where the synthetic bar occludes the limbs of the subjects during the apex (e.g., in *handwaving*) or throughout the conduction of the activity (e.g., in *walking*). The localization accuracy achieved, compared with no occlusion at all is depicted in Fig. 13(c). As can be seen from the figure, there is only a small drop in localization performance. We conclude, therefore, that the proposed method is able to sufficiently localize a subject, as long as a good portion of the activity is not affected by the occlusion.

F. Dynamic Background/Multiple Activity Detection

We use synthetic and real sequences in order to demonstrate the robustness of the proposed algorithm against dynamic background. Our goal is to demonstrate that the proposed algorithm is not distracted by movement that is due to a varying background or irrelevant activities in the scene. To simulate such conditions, we create synthetic sequences in which more that one activities are depicted in the same frame, as shown in Fig. 14(a). Our goal is to localize each activity regardless of the presence of the other. A depiction of the spatial voting spaces derived by the application of the *boxing* and *handwaving* models for one instance of the activity is given in Fig. 14. As can be seen from the figure, each model manages to suppress the information coming from activities other than its class. For instance, the votes attained by the *boxing* model are concentrated around the subject that performs this activity. The reason for this is that ensembles that are localized around the *handwaving* subject do not match well or at all the codewords in the *boxing* codebook. In Fig. 15 we present the effect of this experiment to the achieved spatial localization, after applying a Kalman filter on the outcomes of the mean shift mode estimator. For comparison, we also plot the same estimates for the clean sequences. As can be seen from the figure, due to false codeword matches, the localization accuracy of the center of the

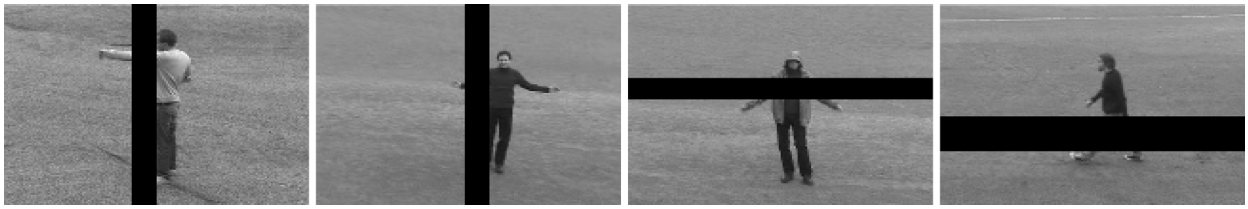
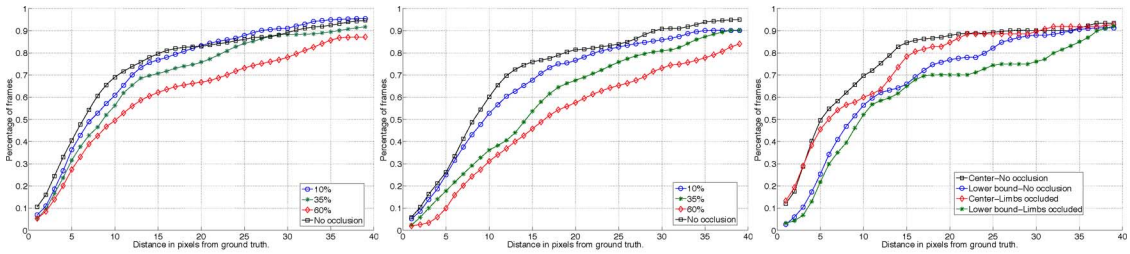


Fig. 12. Occlusion settings for the *boxing*, *handclapping*, *handwaving* and *walking* classes. The setting for the *jogging* and *running* classes is similar to that of the *walking* class.



(a) (b) (c)

Fig. 13. Average spatial localization results for the selected occluded sequences. (a) Center of the subject (b) Lower bound of the subject. (c) Average localization accuracy achieved for the center and lower bound of the subject when the tips of the limbs are occluded.

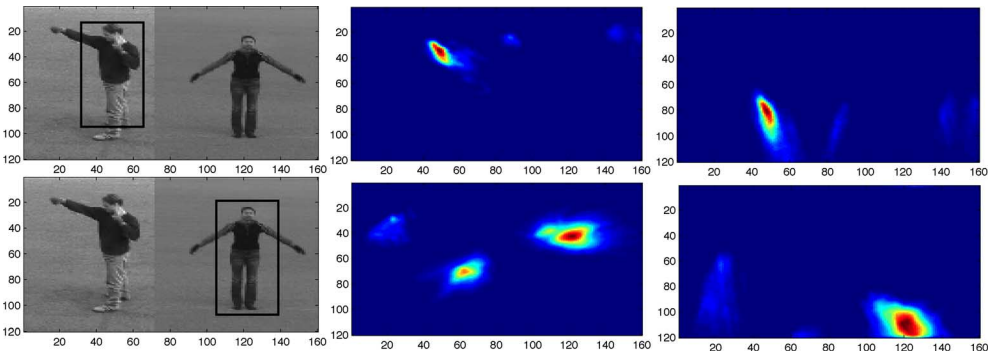


Fig. 14. Voting spaces for center and lower bound derived using the (a) *boxing* and (b) *handwaving* models, and rectangles fitted around the subjects using these voting spaces. Notice that each model favors votes belonging to the activity it was trained for.

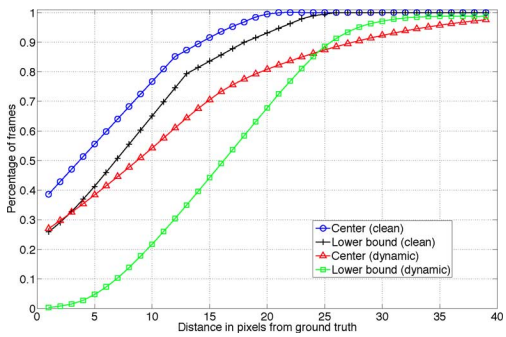


Fig. 15. Average spatial localization accuracy results achieved for the sequences depicting multiple activities. For comparison, the accuracy achieved on the clean sequences is also depicted.

subject drops about 10%, while for the subject's lower bound the effect is more severe. We depict, in Fig. 16, the temporal voting spaces created using the *boxing* and *handwaving* models. As can be seen, there are 6 peaks in the *boxing* and 2 peaks in the *handwaving* temporal voting space, corresponding to the number of instances of these activities depicted in the image sequence under consideration. Using Mean Shift mode, we extract the corresponding hypotheses, and following the process of Section II-F, the spatiotemporal volumes that correspond to those hypothesis are classified in an RVM based classification scheme. Finally, we depict, in Fig. 17, the spatial voting spaces acquired using the *handclapping* and *boxing* models for an instance of the multi-KTH dataset. As can be seen from the figure, and similar to the synthetic sequences presented earlier, each model manages to suppress information coming from activities other than its class.

IV. Conclusion

In this work, we presented a framework for the localization and classification of actions. The voting nature of the proposed method allows us to perform spatiotemporal localization and classification in sequences that have not been pre-segmented. The proposed method uses class-specific codebooks of characteristic ensembles and class-specific models that encode the spatiotemporal positions at which the codewords in the codebook are activated during training. The codebook-model pairs are utilized during testing, in order to accumulate evidence for the spatiotemporal localization of the activity in a probabilistic

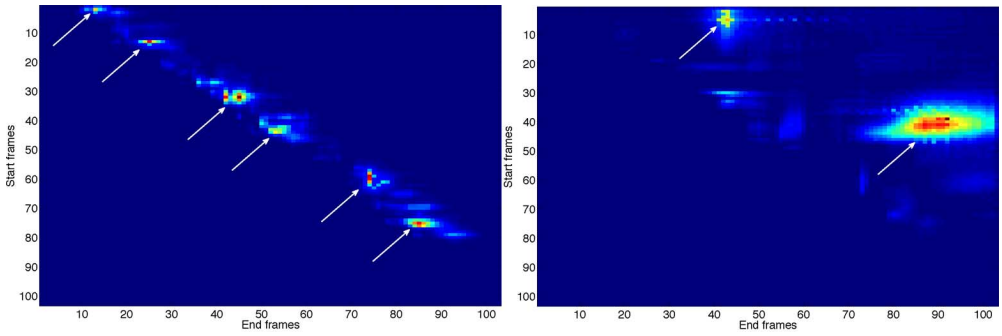


Fig. 16. Temporal voting spaces corresponding to the image sequence of Fig. 14, for (a) boxing and (b) handwaving. Using Mean Shift, 6 instances of *boxing* are extracted from (a) and 2 instances of *handwaving* in (b).

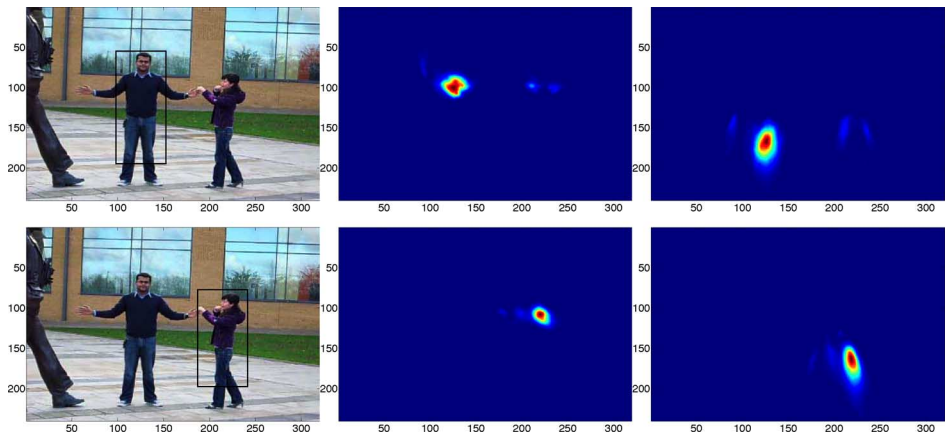


Fig. 17. Spatial localization example on the multi-KTH sequence. Voting spaces and localization result achieved for *handclapping* (top row) and *boxing* (bottom row).

spatiotemporal voting scheme. We presented results on publicly available datasets and demonstrated the robustness of the proposed method in the presence of occlusion and dynamic background. Furthermore, we showed the ability of the proposed method in localizing and classifying multiple activities that take place in the same scene. Finally, we demonstrated the effectiveness of the proposed method by presenting comparative classification and localization results with the state of the art.

REFERENCES

[1] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput. J.*, vol. 28, pp. 976–990, 2010.
[2] M. Pantic, A. Pentland, A. Nijholt, and T. Huang, "Human computing and machine understanding of human behavior: A survey," *Lecture Notes in Artificial Intelligence*, vol. 4451, pp. 47–71, 2007.
[3] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, pp. 1615–1630, 2005.
[4] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. 432–439.
[5] D. Han, L. Bo, and C. Sminchisescu, "Selection and context for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1933–1940.
[6] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *VS-PETS*, 2005, pp. 65–72.
[7] M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1948–1955.
[8] A. Oikonomopoulos, I. Patras, and M. Pantic, "Spatiotemporal salient points for visual recognition of human actions," *IEEE Trans. Syst., Man Cybern. B*, vol. 36, pp. 710–719, 2006.
[9] T. Kadir and M. Brady, "Scale saliency: A novel approach to salient feature and scale selection," in *Int. Conf. Vis. Inf. Eng.*, 2000, pp. 25–28.
[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004.
[11] A. Agarwal and B. Triggs, "Hyperfeatures multilevel local coding for visual recognition," in *Eur. Conf. Comput. Vis.*, 2006, vol. 1, pp. 30–43.
[12] l. J. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in *Proc. ICCV*, 2007, pp. 1–8.
[13] J. Sivic and A. Zisserman, "Video Google: Efficient visual search of videos," *LNCS*, vol. 4170, pp. 127–144, 2006.
[14] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Comput. Vis. Image Understand.*, vol. 110, pp. 346–359, 2008.
[15] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
[16] K. Schindler and L. V. Gool, "Action snippets: How many frames does human action require?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[17] S. Ali and M. Shah, "Human action recognition in videos using kinematic features and multiple instance learning," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 2, pp. 288–303, 2010.
[18] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[19] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2929–2936.
[20] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," in *Proc. British Machine Vis. Conf.*, Edinburgh, U.K., 2006, vol. 1, Online.
[21] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. SIGIR*, 1999, pp. 50–57.
[22] H. Ning, Y. Hu, and T. Huang, "Searching human behaviors using spatial-temporal words," in *Proc. IEEE Int. Conf. Image Process.*, 2007, vol. 6, pp. 337–340.

[23] A. Oikonomopoulos, M. Pantic, and I. Patras, "Sparse b-spline polynomial descriptors for human activity recognition," *Image Vis. Comput. J.*, vol. 27, pp. 1814–1825, 2009.
[24] N. Ikizler and P. Duygulu, "Human action recognition using distribution of oriented rectangular patches," *Lecture Notes in Computer Science*, vol. 4814, pp. 271–284, 2007.
[25] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Proc. ECCV'04 Workshop on Statistical Learning in Computer Vision*, 2004, pp. 17–32.
[26] A. Opelt, A. P. , and A. Zisserman, "A boundary-fragment-model for object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 575–588.
[27] K. Mikolajczyk and H. Uemura, "Action recognition with motion-appearance vocabulary forest," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[28] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering objects and their location in images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 370–377.
[29] O. Boiman and M. Irani, "Detecting irregularities in images and in video," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 462–469.
[30] J. Niebles and L. Fei-Fei, "A hierarchical model of shape and appearance for human action classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[31] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
[32] I. Junejo, E. Dexter, I. Laptev, and P. Prez, "Cross-view action recognition from temporal self-similarities," in *Eur. Conf. Comput. Vis.*, 2008, vol. 2, pp. 293–306.
[33] A. Gilbert, J. Illingworth, and R. Bowden, "Fast realistic multi-action recognition using mined dense spatio-temporal features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 925–931.
[34] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 790–799, 1995.
[35] M. Tipping, "The relevance vector machine," *Advances in Neural Information Processing Systems*, pp. 652–658, 1999.
[36] A. Oikonomopoulos, I. Patras, and M. Pantic, "An implicit spatiotemporal shape model for human activity localization and recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR09), Workshop on Human Communicative Behavior Analysis*, 2009, vol. 3, pp. 27–33.
[37] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, vol. 3, pp. 32–36.
[38] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, vol. 2, pp. 1395–1402.
[39] M. Black and P. Anandan, "A framework for the robust estimation of optical flow," in *Proc. ICCV*, 1993, pp. 231–236.
[40] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," Stanford Univ., Stanford, CA, 1993, Tech. Report.
[41] Y. B. Shalom and T. Fortmann, *Tracking and Data Association*. New York: Academic Press, 1988.
[42] Y. Ke, M. Ans, and R. S. Hebert, "Efficient visual event detection using volumetric features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, vol. 1, pp. 166–173.
[43] K. Rapantzikos, Y. Avrithis, and S. Kollias, "Dense saliency-based spatiotemporal feature points for action recognition,," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1–8.
[44] M. Ahmad and S. Lee, "Human action recognition using shape and clg-motion flow from multi-view image sequences," *Pattern Recognit.*, vol. 41, pp. 2237–2252, 2008.
[45] A. Fathi and G. Mori, "Action recognition by learning midlevel motion features,," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
[46] S. Wong and R. Cipolla, "Extracting spatiotemporal interest points using global information,," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

Antonios Oikonomopoulos (S'00–M'10) received the B.Sc. degree in electrical and computer engineering from the Electrical Engineering Department, Aristotle University of Thessaloniki, Greece, in 2002, the M.Sc. degree in media and knowledge engineering from the Electrical Engineering Department, Delft University of Technology, The Netherlands, in 2006, and the Ph.D. degree in computer science from the Computing Department, Imperial College London, U.K., in 2010.

Currently, he is a Research Associate at the Computing Department of Imperial College London, U.K. His research interests lie in the field of computer vision and pattern recognition, and more specifically in human activity analysis, image and video retrieval, human-machine interaction and digital image processing.

Dr. Oikonomopoulos was a recipient of the Greek State Scholarships Foundation (I.K.Y.) from 2003 to 2007.

Ioannis (Yiannis) Patras (S'97–M'02) received the B.Sc. and M.Sc. degrees in computer science from the Computer Science Department, University of Crete, Heraklion, Greece, in 1994 and in 1997, respectively, and the Ph.D. degree from the Department of Electrical Engineering, Delft University of Technology, The Netherlands, in 2001.

He has been a Postdoctorate Researcher in the area of multimedia analysis at the University of Amsterdam, and a Postdoctorate Researcher in the area of vision-based human machine interaction at TU Delft. Between 2005 and 2007 he was a Lecturer in computer vision at the Department of Computer Science, University of York, York, U.K. He is a Senior Lecturer in computer vision in the School of Electronic Engineering and Computer Science at Queen Mary University of London. He is or has been in the organizing committees of IEEE SMC 2004, Face and Gesture Recognition 2008, and ICMR2011, and was the general chair of WIAMIS 2009. He is an associate editor of the *Image and Vision Computing Journal* and the *Journal of Multimedia*. His research interests lie in the areas of computer vision and pattern recognition, with emphasis on motion analysis, and their applications in multimedia retrieval, multimodal human computer interaction, and visual communications. Currently, he is interested in the analysis of human motion, including the detection, tracking and understanding of facial and body gestures.

Maja Pantic (M'03–SM'06) is a Professor in affective and behavioral computing in the Department of Computing, Imperial College London, London, U.K, and at the Department of Computer Science, University of Twente, Enschede, The Netherlands. She is one of the world's leading experts in the research on machine understanding of human behavior including vision-based detection, tracking, and analysis of human behavioral cues like facial expressions and body gestures, and multimodal human affect/mental-state understanding. She has published more than 100 technical papers in these areas of research. In 2008, for her research on Machine Analysis of Human Naturalistic Behavior (MAHNOB), she received European Research Council Starting Grant as one of 2% best young scientists in any research field in Europe.

Prof. Pantic is also a partner in several FP7 European projects, including the currently ongoing FP7 SSPNet NoE, for which she is the scientific coordinator. She currently serves as the Editor in Chief of the *Image and Vision Computing Journal* and as an Associate Editor for the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS PART B (TSMC-B). She has also served as the General Chair for several conferences and symposia including the IEEE FG 2008 and the IEEE ACII 2009. Her website is http://www.doc.ic.ac.uk/maja/.