Discriminative Space-time Voting for Joint Recognition and Localization of Actions.

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
i.patras@eecs.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 activity detection 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. 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 experimental results for a number of publicly available datasets, that demonstrate the efficiency of the proposed method in localizing and classifying human activities.

1. INTRODUCTION

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 and localization have inspired a number of methods in the area of activity recognition. Typical examples include space-time interest points [3], space-time cuboids [4][5], spatiotemporal salient points [6], SIFT features [7] and features that are based on the human visual system [8]. Interest points have been successfully used in approaches that employ visual codebooks for human activity representation [9] [10]. A visual codebook is usually created by clustering extracted feature descriptors in the training set. Each of the resulting centers is considered to be a codeword and the set of codewords forms the codebook. In this way, the information depicted in images and videos can be summarized as a histogram of visual words. Despite their success, an obvious disadvantage of such methods is that, by using histograms, the information concerning the spatiotemporal arrangement of the descriptors is lost. To deal with this issue, a number of methods encode the spatiotemporal relationships between the extracted codewords. For instance, Leibe et al. [11] propose an implicit shape model for object detection, where the relative position of each word with respect to the object center is maintained. In [12], a similar voting scheme is implemented for activity recognition and localization. Sivic et al. [13] propose the use of doublet codewords, while Boiman and Irani [14] propose a matching method based on feature ensembles for irregular scene detection. Finally, areas in images(videos) that share similar geometric properties and similar spatio(temporal) layouts are matched in [15], using a self similarity descriptor and the algorithm of [14].

In this paper, we extend the work of Leibe et al. [11] 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 [6]. We model the feature ensembles using a modified star graph model that is similar to the one proposed in [14], but compensates for scale changes using the scales of the features within each ensemble. During training, we create codebooks of characteristic ensembles for each class, and store the spatiotemporal positions at which each codeword is activated in the training set. This is performed with respect to a set of reference points, and with respect to the start/end of the action instance. During testing, each activated codeword ensemble 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. By doing so, we create a set of class-specific voting spaces. We use a Mean Shift algorithm [16] at each voting space in order to extract the most probable hypotheses concerning the spatiotemporal extend of the activities. Each hypothesis corresponds to a spatiotemporal volume, and is

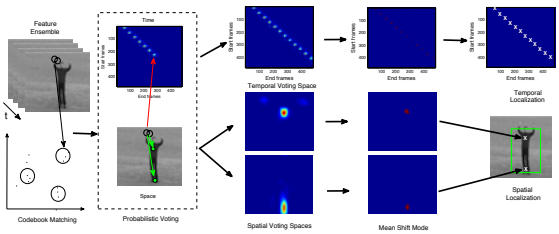


Figure 1: Overview of the spatiotemporal voting process. Activated codewords cast spatial and temporal votes with respect to the center and spatial lower bound of the subject and the start/end frame of the action instance. Mean shift is subsequently used in order to extract hypotheses concerning the position of a reference point in each frame and the temporal boundaries of an action instance.

verified by performing action category classification with a Relevance Vector Machine (RVM) [17]. An overview of the proposed spatiotemporal voting process is depicted in Fig. 1.

The remainder of this paper is organized as follows. In section 2 we present our approach. That is, the creation of our spatiotemporal models for each class and the way they are used to perform detection. Section 3 includes our experimental results, and finally, section 4 concludes the paper.

2. SPATIOTEMPORAL VOTING

We propose to use a probabilistic voting framework in order to spatiotemporally localize human activities. This framework, described in section 2.4, 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 2.1, while section 2.2 describes how these features are combined into ensembles and how ensembles are compared to each other. Each codebook is created using a feature selection process based on boosting, which selects a set of discriminative ensembles for each class, and is associated with a class-specific spatiotemporal localization model, which encodes the locations and scales at which each codeword is activated in the training set. This process is described in section 2.3. 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 2.6.

2.1 Features

The proposed framework can be utilized with any kind of local descriptors. Here, we use optical flow and spatial gradient descriptors, extracted around automatically detected spatiotemporal salient points. We use the algorithm of [6] in order to extract the set of spatiotemporal salient points, which we denote with $\boldsymbol{S} = \{(\boldsymbol{c_i}, \boldsymbol{s_i})\}$. Here, $\boldsymbol{c_i}$ is the spatiotemporal position of the i^{th} point and $\boldsymbol{s_i}$ is its spatiotemporal scale. For 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. 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. Using their horizontal and vertical components, we convert these vectors into angles and bin them into histograms using a bin size of 10 degrees.

2.2 Feature ensemble similarity

Let $e_d = (c_d, \{v_d^i, l_d^i\}_{i=1...\mathcal{M}})$ be an ensemble 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. Similar to [14], we model the joint probability $P(e_d, e_q)$ between the database ensemble e_d and the query ensemble e_q as:

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

The likelihood in Eq. 1 can be factored as:

$$P(c_d, v_d^1, ..., l_d^1, ..., c_q, v_q^1, ..., l_q^1, ...) = \alpha \prod_i \max_j (P(l_q^j | l_d^i, c_d, c_q) P(v_q^j | v_d^i)) P(v_d^i | l_d^i). \quad (2)$$

The first term in eq. 2 expresses the similarity in the ensemble topology, and the second term expresses the similarity in their descriptor values. We model the first term as:

$$P(l_q^j | l_d^i, c_q, c_d) = z_1^{-1} \exp(-((l_q^j - c_q)S_q^j - (l_d^i - c_d)S_d^i)^T \cdot \mathcal{S}^{-1}((l_q^j - c_q)S_q^j - (l_d^i - c_d)S_d^i)), \quad (3)$$

where z_1 is a normalization term, $\mathcal{S}$ is a fixed covariance matrix controlling the allowable deviations in the relative feature locations and 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. By normalizing the distance between the individual features and the ensemble center with the spatiotemporal scales of the features, we achieve invariance to scaling variations. We model the second term in the maximum in eq. 2, using an exponential distribution:

$$P(v_q^j | v_d^i) \propto z_2^{-1} \exp\left(-z_3^{-1} D(v_q^j, v_d^i)\right), \quad (4)$$

where z_2, z_3 are normalization terms, and $D(.,.)$ is the χ^2 distance. Finally, similar to [14], we model the last term of eq. 2 using examples from the database:

$$P(v_d | l_d) = \begin{cases} 1 & (v_d, l_d) \in DB \\ 0 & otherwise \end{cases}, \quad (5)$$

where v_d, l_d are an arbitrary descriptor and location.

2.3 Feature Selection and Codebook Creation

We use Gentleboost [18] 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. To do so, we randomly sample $\mathcal{L}$ (e.g. 5000) ensembles from the positive examples. Using Eq. 1, we match these ensembles to the remaining ones in the positive set and the ones in the negative set and 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

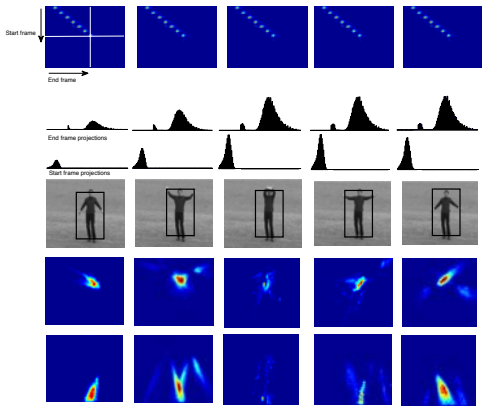


Figure 2: Spatiotemporal voting result. First row: temporal voting space. Second, third row: Start/end frame projections along lines passing from a local maximum. Evidence is accumulated as time progresses, resulting in more votes at the most probable positions. Fifth, sixth row: Spatial voting spaces. Fourth row: Fitted bounding boxes.

vectors, Gentleboost selects a set of characteristic ensembles for the positive class. By performing this process we end up with a set of characteristic ensembles for each class. 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. 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. In addition, we also store 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. During testing, $\{\boldsymbol{\theta_d}\}, \boldsymbol{S_d}$ 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.

2.4 Probabilistic framework

Our goal is to acquire a probability distribution over 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. In order to do so, we propose the use of a spatiotemporal voting scheme, which is an extension in time of the model proposed in [11]. 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 feature ensembles as codewords, modeled using the star-graph model of [14]. In the following, and without loss of generality, we drop the subscripts $\boldsymbol{\theta_s}, \boldsymbol{\theta_t}$, and describe our probabilistic framework for the generalized parameter $\boldsymbol{\theta}$. The probability of $\boldsymbol{\theta}$ can be formulated as:

$$P(\boldsymbol{\theta}) = \sum_{q=1}^{Q} P(\boldsymbol{\theta}|e_q)P(e_q), \tag{6}$$

where $\{e_q\}$ is the set of observed ensembles and $P(e_q)$ is the prior probability of observing e_q. We model the latter as a uniform distribution. Each observed ensemble e_q is matched against each codeword e_d from the codebook $\boldsymbol{E}$. By marginalizing $P(\boldsymbol{\theta}|e_q)$ on $e_d \in \boldsymbol{E}$ we get:

$$P(\boldsymbol{\theta}|e_q) = \sum_{e_d \in \boldsymbol{E}} P(\boldsymbol{\theta}|e_d, e_q)P(e_d|e_q). \tag{7}$$

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 2.2. 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. $P(\boldsymbol{\theta}|e_d)$ can be modeled as:

$$P(\boldsymbol{\theta}|e_d) = w_d \sum_{\boldsymbol{\theta}_d} P(\boldsymbol{\theta}|\boldsymbol{\theta}_d, e_d)P(\boldsymbol{\theta}_d|e_d), \tag{8}$$

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 in section 2.5. The first term of Eq. 8 is independent of e_d, since votes are cast using the $\boldsymbol{\theta}_d$ values. Votes are cast according to the following equation:

$$\boldsymbol{\theta} = \boldsymbol{\theta}_q + \boldsymbol{S}_q \boldsymbol{S}_d^{-1} \boldsymbol{\theta}_d, \tag{9}$$

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 eq. 9 for the spatial case is depicted in Fig. 3(b). By normalizing with $\boldsymbol{S}_q \boldsymbol{S}_d^{-1}$ we achieve invariance to scale differences between the observed and the activated ensemble codeword. 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:

$$P(\boldsymbol{\theta}|\boldsymbol{\theta_d}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_q - \boldsymbol{S}_d \boldsymbol{S}_q^{-1} \boldsymbol{\theta}_d), \tag{10}$$

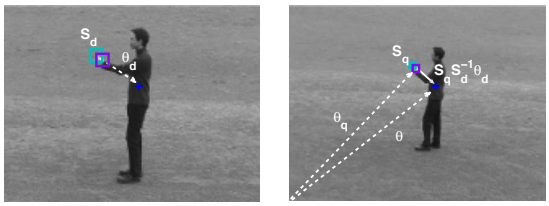
where $\delta(.)$ is the Dirac delta function. Finally, we model $P(\boldsymbol{\theta}_d|e_d)$ using a uniform distribution, that is, $P(\boldsymbol{\theta}_d|e_d) = 1/V$, where V is the number of $\boldsymbol{\theta}_d$ values associated with e_d.

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.

2.5 Localization accuracy

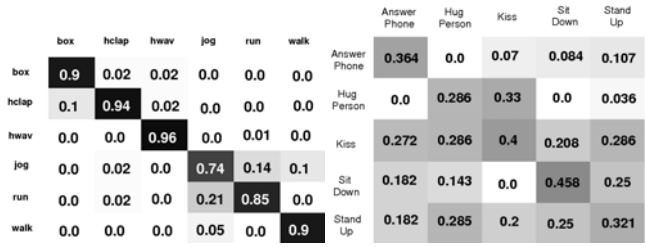
In this section we will describe a methodology to learn w_d, that is, the weight used in eq. 8 and expresses the importance of ensemble e_d in accurately localizing an activity in space and time. More specifically, we would like to favor votes from ensembles that are informative (i.e. characteristic of the location at which they appear) and suppress votes from ensembles that are commonly activated. 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:

$$w_d = e^{-\int P_d(l) \log P_d(l) dl}, \tag{11}$$



(a) (b)

Figure 3: Voting example. (a) During training, the position θ_d and average spatiotemporal scale S_d of the activated ensemble is stored with respect to one or more reference points. (b) During testing, votes are cast using the stored θ_d values, normalized by $S_q S_d^{-1}$ in order to account for scale changes.



(a) (b)

Figure 4: Confusion Matrix for (a) KTH and (b) HoHA.

where the exponent is the Shannon entropy of l. In this way, ensembles that are informative will have a distribution with low entropy, since their votes will be concentrated in a few values, resulting in a large weight.

2.6 Activity detection

Using the probabilistic framework of section 2.4, the proposed algorithm initially casts spatial votes for each class-specific codebook-model pair, according to the information stored in the training stage. We use Mean Shift to localize the most probable centers/lower bounds of the subjects at each frame, and apply a Kalman filter [19] in order to smooth the point estimates from frame to frame. Using these estimates, we fit a bounding box around the subject, as depicted in Fig. 2. 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 on the resulting temporal voting spaces, the most probable hypotheses concerning the temporal extend of the activity are extracted.

We perform a hypothesis verification stage in order to assign a label to each extracted hypothesis. 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. Furthermore, let us denote an extracted hypothesis with F_{ij}, where i, j are the frame indexes at which the activity starts and ends respectively. Our hypothesis verification step relies on the calculation of the following measure:

$$R_{ijm} = \frac{1}{(t_j - t_i)} \sum_{t=t_i}^{t_j} e_{tm}. \quad (12)$$

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

$$\mathcal{K}_{ijm} = R_{ijm} \log R_{ijm}. \quad (13)$$

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

$$Class(F_{ij}) = \arg\max_m (P_m(l|F_{ij})). \quad (14)$$

3. EXPERIMENTAL RESULTS

3.1 Training set

We consider a single repetition of an activity as an action instance, e.g. a single hand-clap. 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.

3.2 Classification

We use activity instances pre-segmented in time to evaluate the classification accuracy of our algorithm and compare it to the state of the art. We use the process of section 2.6 to perform classification, where each hypothesis corresponds to a pre-segmented example. In Fig. 4(a), the confusion matrix for the KTH dataset is depicted. The largest degree of confusion is between *jogging* and *running*. The average recall rate achieved by the RVM classifier for the KTH dataset is 88%. By contrast, using just the measure of eq. 12 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.

In Fig. 4(b), we present the confusion matrix for the HoHa dataset. Due to the small number of representative examples, we discard classes *GetOutOfCar*, *HandShake*, *SitUp*. It can be observed that there are several confusions between classes that are not very similar. The largest confusion, however, is between *HugPerson* and *Kiss*, since both involve two persons coming progressively closer to each other.

3.3 Localization

3.3.1 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 2.6, 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. 5. The worst performing class in these experiments 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 the *boxing*, *handclapping* and *handwaving* classes, while there was a smaller increase in the performance of the *jogging*, *running* and *walking* classes.

3.3.2 Temporal localization

In this section we evaluate the ability of the proposed algorithm in localizing in time instances of a known activity that occur in an image sequence. For this experiment, we apply the process of section 2.6, and compare each extracted hypothesis with the ground truth annotation. Each extracted hypothesis specifies the frames at which the action instance starts and ends. The error of each hypothesis was calculated as the difference in frames between the ground truth and the start/end frames specified by the hypothesis. In this way, we construct Fig. 6, which plots the percentage of the recovered hypotheses as a function of this frame difference. We compare these results with the ones acquired by the algorithm of [15]. By implementing their algorithm, we compute self-similarity descriptors for all sequences in the KTH dataset and apply their progressive elimination algorithm to match a query to each sequence. Matching was performed using 5 query sequences per class and averaging the acquired results. This gives us an estimate of the spatiotemporal extend of each recovered instance. The localization accuracy achieved is depicted in Fig. 6. As can be seen from the figure, the results achieved are similar to the ones achieved by the algorithm of [15] for *boxing* and slightly better for *jogging* and *running*. For *handwaving* and *handclapping*, 70% of the hypotheses extracted by the proposed algorithm are localized within 3 frames from the ground truth on average, in comparison to 15% achieved by [15].

Finally, we performed experiments on hand-raising detection using the political debates dataset of [20]. Hand raising activities in political debates could potentially be an important cue for agreement/disagreement detection. Here, we consider a single raising and lowering of the speaker's hand as a single hand-raising activity instance. We used 10 hand raising instances in order to train the corresponding model, and tested the proposed algorithm on 20 test sequences of political debates. The latter include view-point and scene changes, camera zoom and videos where the onset and offset of the action were out of the camera's view. The localization results that we achieved are depicted in Fig. 7(b), while in Fig. 7(a) a still frame of a hand raising instance is shown. As can be seen from Fig. 7(a), the proposed algorithm was able to localize 90% of the extracted hypotheses within 10 frames from the ground truth annotation.

3.4 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 in the sequence are unknown. Given an image sequence, each model created during training, results in a different voting space for this sequence. Using mean shift, a set of hypotheses is extracted from each

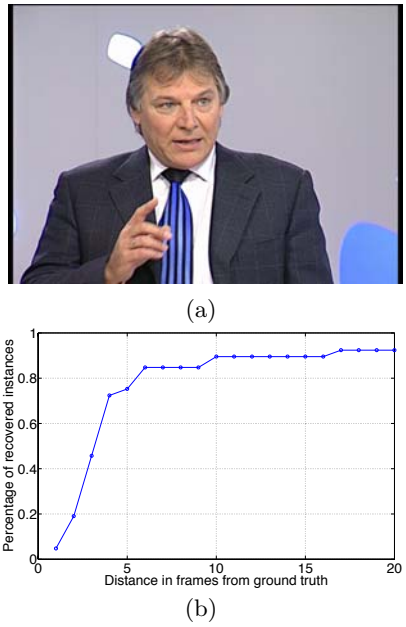


Figure 7: (a) Instance of hand-raising. (b) Achieved temporal localization result for the hand-raising instances.

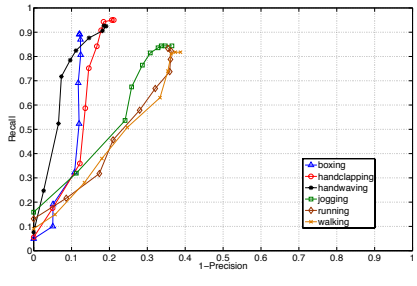


Figure 8: ROC curves corresponding to each class of the KTH dataset.

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 of the voting space at the extraction point. A low weight on a hypothesis means that the proposed algorithm does not have a strong belief on its validity. By setting up a threshold on the weights, we can control which of the hypotheses are considered as being valid. By varying this threshold, we construct the ROC curves of Fig. 8, for each class of the KTH dataset. Note that all curves are well above the main diagonal, meaning that the number of true positives is always larger than the number of false positives.

4. CONCLUSIONS

In this work we presented a framework for the localization and classification of actions. The proposed method utilizes class-specific codebooks of characteristic ensembles and class-specific spatiotemporal 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 ev-

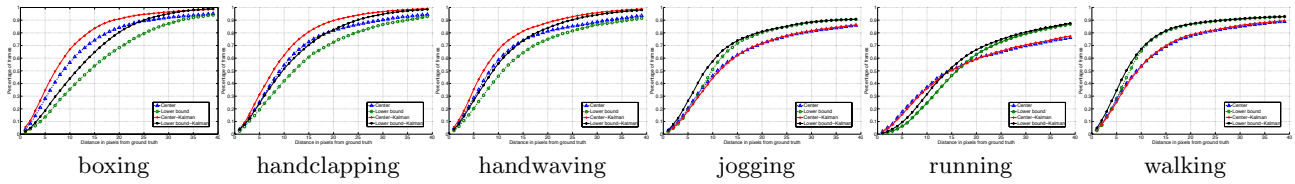


Figure 5: Spatial localization results achieved for the subject center and lower bound. Applying a Kalman filter to the raw outcomes of the mean shift mode estimator lead to an increase in the localization performance. x-axis: distance from ground truth annotation in pixels. y-axis: percentage of frames 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.

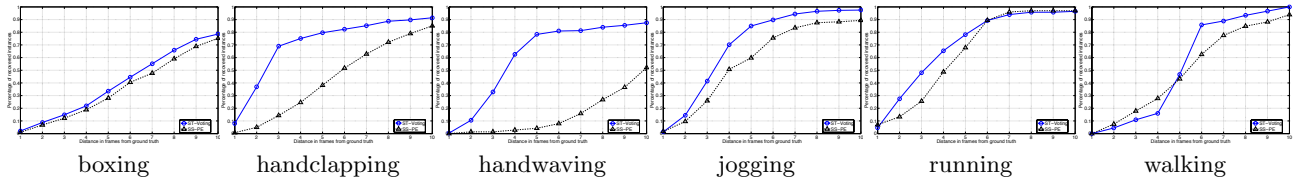


Figure 6: Comparative temporal localization results for the 6 classes of the KTH dataset, between the proposed algorithm (ST-Voting) and the Self-Similarity with Progressive Elimination (SS-PE) algorithm of [15]. x-axis: distance from ground truth annotation in frames. y-axis: percentage of recovered instances.

idence for the spatiotemporal localization of the activity in a probabilistic spatiotemporal voting scheme. We presented results on publicly available datasets that demonstrate the efficiency of the proposed algorithm in human activity detection. Finally, we demonstrated the effectiveness of the proposed method by presenting comparative classification and localization results with the state of the art.

Acknowledgments

This work has been funded in part by the European CommunitySs 7th Framework Programme [FP7/2007Ú2013] under the grant agreement no 231287 (SSPNet). The work of Maja Pantic is also funded in part by the European Research Council under the ERC Starting Grant agreement no. ERC-2007-StG-203143 (MAHNOB).

5. REFERENCES

[1] Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing Journal **28** (2010) 976–990

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

[3] Laptev, I., Lindeberg, T.: Space-time Interest Points. In: IEEE Conf. on Computer Vision and Pattern Recognition. (2003) 432 – 439

[4] Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. VS-PETS (2005) 65– 72

[5] Bregonzio, M., Gong, S., Xiang, T.: Recognising action as clouds of space-time interest points. In: IEEE Conf. on Computer Vision and Pattern Recognition. (2009) 1–8

[6] Oikonomopoulos, A., Patras, I., Pantic, M.: Spatiotemporal Salient Points for Visual Recognition of Human Actions. IEEE Trans. Systems, Man and Cybernetics Part B **36** (2006) 710 – 719

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

[8] Jhuang, H., Serre, T., Wolf, L., Poggio, T.: A Biologically Inspired System for Action Recognition. In: Proc. IEEE Int. Conf. Computer Vision. (2007) 1–8

[9] Ning, H., Hu, Y., Huang, T.: Searching human behaviors using spatial-temporal words. In: Proc. IEEE Int. Conference on Image Processing. Volume 6. (2007) 337–340

[10] Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: IEEE Conf. on Computer Vision and Pattern Recognition. (2009) 2929–2936

[11] Leibe, B., Leonardis, A., Schiele, B.: Combined object categorization and segmentation with an implicit shape model. ECCV'04 Workshop on Statistical Learning in Computer Vision (2004) 17–32

[12] Mikolajczyk, K., Uemura, H.: Action recognition with motion-appearance vocabulary forest. In: IEEE Conf. on Computer Vision and Pattern Recognition. (2008) 1–8

[13] Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering Objects and their Location in Images. In: Proc. IEEE Int. Conf. Computer Vision. Volume 1. (2005) 370 – 377

[14] Boiman, O., Irani, M.: Detecting irregularities in images and in video. In: Proc. IEEE Int. Conf. Computer Vision. Volume 1. (2005) 462–469

[15] Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: IEEE Conf. on Computer Vision and Pattern Recognition. (2007) 1–8

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

[17] Tipping, M.: The Relevance Vector Machine. Advances in Neural Information Processing Systems (1999) 652 – 658

[18] Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. Stanford University Technical Report (1993)

[19] Bar-Shalom, Y., Fortmann, T.: Tracking and Data Association. Academic Press (1988)

[20] Vinciarelli, A., Dielmann, A., Favre, S., Salamin, H.: Canal9: A database of political debates for analysis of social interactions. In: Proc. IEEE Int. Conf. Affective Computing and Intelligent Interfaces. Volume 2. (2009) 96–99