Action Recognition Using Local Spatio-Temporal Oriented Energy Features and Additive Kernel SVMs

Jiangfeng Yang and Zheng Ma
School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu, China
Email: 369322023@qq.com, wallsonyang@163.com

***Abstract*—Spatio-temporal oriented energy features have been proved to be an efficient feature for action recognition. It has satisfied performance on most of public databases. However, the oriented energy features were used as holistic action features for template matching in many literatures. In the paper, we proposed an action representation based on local spatio-temporal oriented energy features, and multiple feature channels are built to convert the features to descriptors. Moreover, inspired by additive kernel Support Vector Machine can offer significant improvements in accuracy on a wide variety of tasks while having the same run-time. We proposed action classifiers based on additive kernels and tested our system on KTH human action dataset for its performance evaluation. The experimental result shows our system outperforms most of recent action classification systems.**

***Index Terms*—action recognition, action representation, spatio-temporal oriented energy, additive kernels**

I. Introduction

Automatic recognition and categorization actions in video sequences is an active research topic in computer vision and machine learning, and their applications can be found in many areas, including content-based video indexing, detecting activities and behaviors in surveillance videos, organizing digital video library according to specified actions, human-computer interfaces and robotics. The challenge is how to obtain robust action recognition and categorization under variable illumination, background changes, camera motion and zooming, viewpoint changes, and partial occlusion. Moreover, the intra-class variation is often very large and ambiguity exists between actions.

Feature representation as a fundamental part of action recognition will greatly influence the performance of the recognition system. Human actions from video data inherently contain both spatial and temporal information, which requires that descriptors of actions in video sequences accurately capture and robustly encode this kind of information. Spatio-temporal oriented energy (STOE), which is derived from the filter responses of orientation selective bandpass filters, as features has been successfully applied in the fields of video tracking and action recognition. Video sequences induce very different orientation patterns in image spacetime depending on their contents. For instance, a textured, stationary object yields a much different orientation signature than if the very same object were undergoing translational motion. An efficient framework for analyzing spatio-temporal (ST) information can be realized through the use of 3D, (x, y, t), oriented energies [1].

The aforementioned energy is well-suited to form the feature representation in visual tracking and action recognition applications for three significant reasons. (1)A rich description of the target is attained due to the fact that oriented energy encompasses both target appearance and dynamics. (2)The oriented energy is robust to illumination changes. By construction, the proposed feature set provides invariance to both additive and multiplicative image intensity changes. (3)The energy can be computed at multiple scales, allowing for a multi scale analysis of the target attributes. Finer scales provide information regarding motion of individual target parts (e.g., limbs) and detailed spatial textures (e.g., facial expressions, clothing logos). In a complementary fashion, coarser scales provide information regarding the overall target velocity and its gross shape [2].

Recently, local representations based on STIPs are drawing much attention [3]-[5], among which human action recognition systems based on the bag of words (BoW) model have achieved good results in many tasks. This would be due to the fact that the BoW model has many advantages, such as being less sensitive to partial occlusions and clutter and avoiding some preliminary steps, e.g. background subtraction and target tracking in holistic methods.

In our recognition scheme, an action is considered as a conglomeration of motion energies in different ST orientations. Consider that motion at a point is captured as a combination of energies along different ST orientations at that point, when suitably decomposed. These decomposed motion energies are a low-level action representation and the basis of the action recognition

Manuscript received October 9, 2013; revised February 1, 2014.
Project number: National Nature Science Foundation of China (grant number 6127128).

doi: 10.12720/ijeee.2.2.124-129

method [6]. Moreover, additive kernel (i.e., intersection kernels) Support Vector machines (SVMs) have become popular for real-time applications as they enjoy both faster training and faster classification, with better classification accuracy and significantly less memory requirements than nonlinear kernels. Therefore, SVMs based on chi-squared, Jensen-Shannan (JS) and intersection kernel were employed as action classifiers in the paper.

We tested the proposed system on KTH human action dataset for performance evaluation. To study cuboid size and codebook size impacting upon recognition rate of action classifiers, the proposed scheme was tested under four sizes of cuboid and codebook. In all cases, the template size of ST oriented filters is set to 3×3×3. What is more, to assess recognition accuracy and training time consumption of classifiers based upon additive kernels, RBF and linear kernels were utilized in the paper.

II. The Proposed Approach

Our action categorization system is illustrated in Fig. 1 The details of the proposed system are explained as follows.

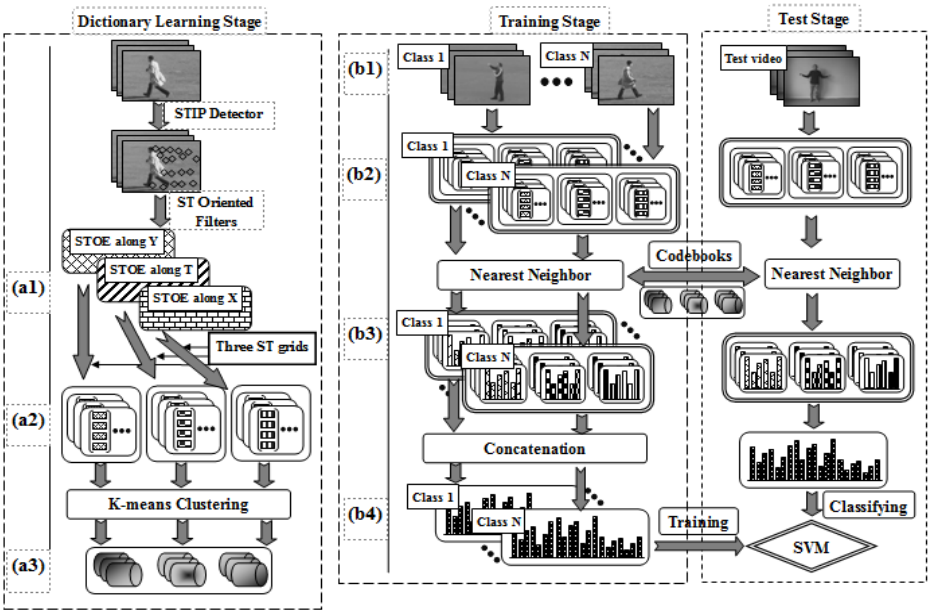

Figure 1. Framework of the proposed action classification system. In dictionary learning stage, (a1) Action motion information is decomposed into local STOE features along X, Y and T axis. (a2) STOE features are transformed into descriptors. (a3) A feature channel consists of an energy orientation and a ST grid. A feature channel produces its own codebook. In training stage, (b1) Input training action videos. (b2) Action motion information is converted into local STOE features. (b3) Each action class is modeled by nine local histograms. (b4) Each action class is represented as a final histogram, which is built by concatenating its nine sub-histograms.

A. Gamma Normalization and Frame Smoothing

Before extracting STIPs from action video, Gamma normalization should be implemented to enhance the local dynamic range of the image in dark or shadowed regions, while compress it in bright regions and at highlights. The basic principle is that the intensity of the light reflected from an object is the product of the incoming illumination (which is piecewise smooth for the most part) and the local surface reflectance (which carries detailed object-level appearance information). In the paper, Gamma normalization is realized using Matlab command "*J*= imadjust(*I*)", which maps the intensity values in grayscale image *I* to new values in *J*. This increases the contrast of the output image *J*. Moreover, to reduce the noise effect, Gaussian kernel with $\sigma = 0.5$ as smoothing filters is used to smooth each frame in action video.

B. STIP Detector

In the paper, the Dollar detector [4] is chosen to be STIP detector, since it generally produces a high number of STIPs, which is important for action representation built upon BoW model. In the following, we provide a brief review of the Dollar detector, which respectively employs 2D Gaussian filter in the spatial direction and 1D Gabor filter in the temporal direction. The two separate filters can produce high response at points with significant ST intensity variations. The response function R of the input image sequence $I(x, y, t)$ has the form:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \quad (1)$$

where $g(x, y; \sigma)$ is the 2D Gaussian smoothing function, applied only along the spatial dimensions, h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied temporally, which are defined as $-\cos(2\pi t\omega)\times\exp(-t^2/\tau^2)$ and $-\sin(2\pi t\omega)\times\exp(-t^2/\tau^2)$, respectively. The parameters ω and τ correspond to the spatial and temporal scales of the detector, respectively.

It is well accepted that stable and reliable STIPs extracted from action videos can offer more helpful information to improve recognition accuracy. In order to

extract such STIPs from noisy videos, the response function should be calculated at multiple spatial and temporal scales. Specifically, the response function $R(\omega,\tau)$ is defined as:

$$R(\omega,\tau)=\sum_{m=1}^{M}R(\omega_m,\tau_m),\omega_m\succ 0,\tau_m\succ 0 \quad (2)$$

where $R(\omega_m,\tau_m)$ denotes the response value of STIP detector at parameter values (ω_m,τ_m).

C. Local STOE Features

Events in a video sequence will generate diverse structures in the ST domain. For instance, a textured, stationary object produces a much different signature in image space-time than if the same object were moving. One method of capturing the ST characteristics of a video sequence is through the use of oriented energy. The energy is derived using the filter responds of oriented selective bandpass filters when they are convolved with the ST cuboid produced by a video stream.

In the paper, local STOE features are obtained by filtering using a set of Gaussian derivative filters and their corresponding Hilbert transform filters, pointwise squaring and summation over each 3D cuboid that is associated with a detected STIP. We use the approach in [7] to obtain local STOE feature. Specifically, the first step consists of filtering a small ST cuboid $\mathbf{C}(\mathbf{x}_i)$ centering on a STIP $p(\mathbf{x}_i)$, by the directionally selective filters G_2 and H_2 at orientation vector $\boldsymbol{\theta}_j=(\alpha_j,\beta_j,\gamma_j),\quad j\in\{1,\cdots,J\},(\alpha_j,\beta_j,\gamma_j)$ denotes direction cosines, J is the number of orientations. Next, the filters are taken in quadrature to eliminate phase sensitivity in the output of each filter. Local STOE energy $E_{\boldsymbol{\theta}_j}(\mathbf{x}_i)$ in cuboid $\mathbf{C}(\mathbf{x}_i)$ at orientation $\boldsymbol{\theta}_j$ can be computed according to

$$E_{\boldsymbol{\theta}_j}(\mathbf{x}_i)=(G_{2,\boldsymbol{\theta}_j}*\mathbf{C}(\mathbf{x}_i))^2+(H_{2,\boldsymbol{\theta}_j}*\mathbf{C}(\mathbf{x}_i))^2 \quad (3)$$

where $\mathbf{x}_i=(x_i,y_i,t_i),i\in\{1,\cdots,n\}$ denotes STIP coordinates of n STIPs; $*$ is convolution; G_2 denotes 3D steerable, separable filters based on Gaussian second derivative; H_2 their corresponding Hilbert transform. The method of constructing ST filters G_2 and H_2 in [7] was adopted in the paper.

In our system, motion information were decomposed into local STOE along X, Y and T axis using ST oriented filters, respectively.

D. The Descriptor

Once local STOE features were obtained, in this section, our goal is to convert the features into their descriptors. First of all, we define a support region as a cuboid containing STOE feature around a STIP, and the support region is divided into several cells by ST grids with multiple temporal scales for computing descriptor, in order to delineate the temporal variations of a local motion pattern and embed ST structure information.

Supposing a support region is divided into $\eta=\eta_1\times\eta_1\times\eta_2$ cells, where $\eta_1\times\eta_1$ is spatial grid arrangement, is temporal segment number. For each cell, the mean absolute difference (MAD, m_1), the second (variance, m_2), third (skewness, m_3) central moments are computed. The mean absolute difference (MAD) for cell c with size s=s1×s2×s3 is defined as

$$\mathrm{MAD}=\frac{1}{s}\sum_{u_1=1}^{s_1}\sum_{u_2=1}^{s_2}\sum_{u_3=1}^{s_3}|e(u_1,u_2,u_3)-mean(c)| \quad (4)$$

$$m_k=\frac{1}{s}\sum_{u_1=1}^{s_1}\sum_{u_2=1}^{s_2}\sum_{u_3=1}^{s_3}(e(u_1,u_2,u_3)-mean(c))^k,k=1,2,3, \quad (5)$$

$$mean(c)=\frac{1}{s}\sum_{u_1=1}^{s_1}\sum_{u_2=1}^{s_2}\sum_{u_3=1}^{s_3}(e(u_1,u_2,u_3)) \quad (6)$$

where $e(u_1,u_2,u_3)$ denotes energy value at position (u_1,u_2,u_3). For a support region divided into η cells, each cell produces a column vector $[m_1,m_2,m_3]^T$. Hence, a matrix $\mathbf{M}\in R^{3\times\eta}$ is obtained to describe STOE feature inside the support region. Next, normalized matrix $\overline{\mathbf{M}}\in R^{3\times\eta}$ is constructed by dividing each row of $\mathbf{M}$ by ℓ_1 norm of the row. All row vectors of $\overline{\mathbf{M}}$ are concatenated on top of each other to form a single vector $\mathbf{m}\in R^d,d=3\times\eta$. The vector $\mathbf{m}$ is the descriptor of STOE feature inside the support region.

In the proposed approach, three ST grids were established to partition a support region into 2×2×1, 2×2×2 and 2×2×3 cells, respectively. The combination of the three ST grids with three ST orientations results in 9 possible feature channels. Each feature channel decomposes motion information within a cuboid centering a STIP into local STOE feature, and partitions the support region into several cells. As each cell produces a 3-D vector, the dimensionalities of descriptor corresponding to the above ST grids are 12, 24 and 36, respectively. Fig. 2 illustrates that three ST grids transform local STOE features into descriptors.

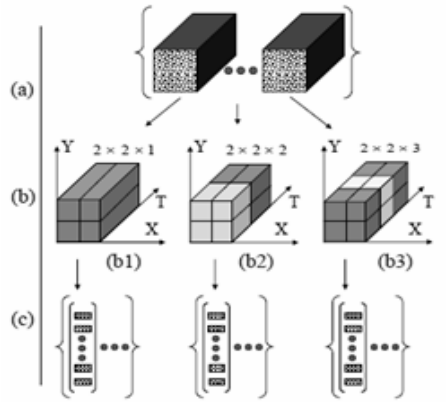

Figure 2. A set of local STOE features is converted into three sets of descriptors using three ST grids. (a) A set of support regions. (b) ST grids with three temporal scales. (b1) ST grid with size 2×2×1. (b2) ST grid with size 2×2×2. (b3) ST grid with size 2×2×3. (c) Three sets of descriptors are built, using ST grids partitioning support regions.

E. Codebooks

The descriptors produced by a feature channel bring about a codebook by implementing *K*-means clustering algorithm and Euclidean distance as metric. Therefore, nine feature channels result in nine codebooks. The descriptors then are encoded by corresponding codebook to obtain local histogram. All local histograms are concatenated to form a final histogram for training and test the proposed system. Hence, an action video sequence is represented as a final histogram vector. In our system, all local histograms have same length.

F. Additive Kernel SVMs

Discriminative classifiers based on Support Vector Machines (SVMs) and variants of boosted decision trees are two of the leading techniques used in vision tasks ranging from object detection, multiclass object recognition to texture classification. Classifiers based on boosted decision trees such as [8], have faster classification speed, but are significantly slower to train. Furthermore, the complexity of training can grow exponentially with the number of classes [9]. On the other hand, given the right feature space, SVMs can be more efficient during training. Part of the appeal of SVMs is that, nonlinear decision boundaries can be learnt using the "kernel trick". However, the run-time complexity of a nonlinear SVM classifier can be significantly higher than a linear SVM. Thus, linear kernel SVMs have become popular for real-time applications as they enjoy both faster training and faster classification, with significantly less memory requirements than non-linear kernels.

Recently, additive kernel SVMs (AKSVMs) have been broadly used in many of the current most successful object detection and recognition algorithms efficient, as well as real-time applications. The class of kernels includes the intersection kernel $K_{\min}(\mathbf{x},\mathbf{y})$, chi-squared kernel $K_{\chi^2}(\mathbf{x},\mathbf{y})$, and JS kernel $K_{JS}(\mathbf{x},\mathbf{y})$. They are defined as following,

$$K_{\min}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \min\{x_i, y_i\} \quad (8)$$

$$K_{\chi^2}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} 2(x_i y_i)/(x_i + y_i) \quad (9)$$

$$K_{JS}(x,y) = \sum_{i=1}^{n} (\frac{x_i}{2}) \log_2 \frac{x_i + y_i}{x_i} + (\frac{y_i}{2}) \log_2 \frac{x_i + y_i}{y_i} \quad (10)$$

In addition of the additive kernels provided, RBF kernel $K_{RBF}(\mathbf{x},\mathbf{y})$ and linear kernel $K_{linear}(\mathbf{x},\mathbf{y})$ are employed as performance benchmark in our experiment.

$$K_{RBF}(\mathbf{x},\mathbf{y}) = \exp(-\gamma \|\mathbf{x}-\mathbf{y}\|^2), \gamma \succ 0 \quad (11)$$

$$K_{linear}(\mathbf{x},\mathbf{y}) = \mathbf{x}^T.\mathbf{y} \quad (12)$$

III. Experimental Result

A. Experimental Setting

We test our algorithm on the KTH human motion dataset. KTH human motion dataset is a video sequence dataset of human actions. Each video has only one action. The dataset contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several periods by 25 subjects in different scenarios of outdoor and indoor environment with scale change. It contains 600 short sequences.

In the preprocessing stage, contrast normalization is used to reduce the influence of illumination changes. STIPs are extracted from action video using Dollar STIP detector with multiple temporal and spatial scales. To extract stable, reliable STIPs, ω, τ are set to $\{(\omega,\tau) \mid \omega = 0.1, 0.2, 0.3; \tau = 1.2, 1.3, 1.4, 1.5\}$. STIP detector respond is obtained by summing over the individual responds. We build codebooks from all videos of each action from two subjects. We perform leave-one-out (LOO) to test the efficiency of our approach in recognition. Specifically, for each LOO run, the videos of 24 subjects are used for training a model, and the videos of the remained subject are used for testing the model, and computing a confusion matrix for evaluation. The results are reported as the average confusion matrix of 25runs.

Accuracy rate of the recognition system based on BoW model is sensitive to the size of cuboid. Too big cuboid can obtain much irrelevant information to action motion, such that the discriminative power of individual cuboid is damaged; whereas, too small size cuboid enclose insufficient motion information for categorization, and hence the information provided is highly unreliable. We test our system in the case of four cuboid sizes: 9(pixels)×9(pixels)×15(pixels), 9×9×9, 7×7×9 and 7×7×5. Moreover, computational efficiency and accuracy are discussed in the section.

B. Discussion

Table I shows classification accuracy of SVMs based on five kernels in the case of four cuboid sizes, respectively.

Recognition accuracy. We compared the average recognition accuracy of the five kernel SVMs on KTH dataset. It can be seen that recognition accuracy of AKSVMs is higher than that of others under the same configuration. The best recognition rates of chi-squared, JS, and intersection kernel SVMs achieve 94.2%, 93.9%, and 93.6%, respectively, which outperform or approach the recent action recognition systems. Meanwhile, recognition accuracy of SVMs based on RBF and linear kernels just reach 91.5% and 91.7%, respectively. Table II shows recognition rate comparison of our method and other recent recognition systems. It can be seen that our systems achieve better performance.

The partial reason why AKSVM classifiers perform better than that based on RBF, linear kernels lies in that: ST grids, which partition support region for computing descriptor, essentially involve overlaps between them, although the ST grids are in general expected to be complementary to each other. The best performance of a

particular action category generally entails only a linear combination of the ST grids. As a result, there is no doubt that the redundant information exist in the resulting local histograms. It is concluded that the AKSVM classifiers efficiently exploit the redundant information to improve their recognition rate; RBF and linear kernel classifiers fail to efficiently do it.

TABLE I. CLASSIFICATION ACCURACY ON THE KTH DATASET AT VARIOUS CUBOID SIZES AND CODEBOOK SIZES.

Cuboid size:9×9×15 Codebook size							
	100	200	300	400	500	600.	700
kernel	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.
Chi-sq.	90.3	91.5	91.7	92.5	93.2	93.2	**93.3**
JS	91.5	91.2	92.2	92.7	93.2	93.3	**93.6**
Inters.	91.0	91.2	91.7	91.8	92.2	92.4	**92.5**
RBF	89.3	89.5	89.6	89.8	90.2	90.5	**90.8**
Linear	89.8	90.1	89.5	89.8	90.2	90.2	**90.6**

Cuboid size:9×9×9 Codebook size							
	100	200	300	400	500	600.	700
kernel	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.
Chi-sq.	91.0	92.0	92.8	93.0	**93.7**	93.3	93.2
JS	90.3	92.1	93.0	93.2	**93.9**	93.3	93.5
Inters.	92.0	91.8	92.0	92.9	**93.6**	93.3	93.2
RBF	89.7	88.9	89.6	90.3	90.3	**90.4**	90.5
Linear	90.6	90.5	90.3	91.3	91.3	**91.5**	90.7

Cuboid size:7×7×9 Codebook size							
	100	200	300	400	500	600.	700
kernel	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.
Chi-sq.	90.0	91.6	92.3	92.9	93.3	93.6	**94.2**
JS	90.0	91.6	92.3	92.5	92.5	92.3	**92.5**
Inters.	90.0	91.5	92.3	92.1	92.3	92.7	**93.2**
RBF	89.3	89.5	89.8	90.1	**90.5**	90.1	90.3
Linear	90.5	90.8	90.6	91.1	**91.7**	91.3	91.5

Cuboid size:7×7×5 Codebook size							
	100	200	300	400	500	600.	700
kernel	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.	Acc.
Chi-sq.	91.1	92.2	92.2	92.1	**92.5**	92.1	92.0
JS	91.0	92.1	92.2	92.3	**92.8**	92.5	92.3
Inters.	90.8	91.5	92.0	92.0	**92.5**	91.8	91.5
RBF	88.0	89.7	90.2	90.8	**91.5**	90.8	90.8
Linear	88.3	90.4	91.0	91.0	**91.7**	91.2	91.5

TABLE II. COMPARISON WITH PREVIOUS WORK ON THE KTH DATASET.

Method	Accuracy
[10]	91.8%
[9]	93.2%
[11]	93.9%
[12]	94.5%
[13]	94.2%
[14]	94.5%
[15]	93.9%
[16]	93.6%
Our method (chi-sq. kernel)	**94.2**%
Our method (JS kernel)	**93.9**%
Our method (Inters. Kernel)	**93.6**%

The size of cuboid is another reason impacting on recognition rate. Too larger or too small cuboid size could bring about negative effect to recognition rate as recognition rate. Too larger or too small cuboid size could bring about negative effect to recognition rate discussed above.

Table III and Table IV show confusion matrixes of AKSVMs for the KTH dataset under various cuboid sizes. The confusion matrixes show the largest confusion occurs between "boxing" and "hand clapping", "walking" and "jogging", and between "jogging" and "running". Several reasons should be responsible for such confusion. Firstly, the decomposed motion energy of the actions mainly distributes along certain orientation, such as significant amount of discomposed motion energy of "boxing" and "hand clapping" distributes along X axis. Hence, the information obtained about action motion direction is insufficient for distinguishing them. Secondly, the speed difference of similar action class is so small that distinguishing them is difficulty, especially between "walking" and "jogging", and between "jogging" and "running".

TABLE III. CONFUSION MATRIXES OF AKSVMS FOR THE KTH DATASET. CHI-SQUARED (TOP LEFT), JS (TOP RIGHT), AND INTERSECTION KERNELS (BOTTOM). ROWS ARE GROUND TRUTH, COLUMNS ARE MODEL RESULTS. BOXING(S1), HAND-CLAPPING(S2), HAND-WAVING(S3), JOGGING(S4), WALKING(S5), RUNNING (S6).

each local histogram with 700D and cuboid size 9×9×15. Average recognitions are 93.3%, 93.6% and 92.5% for chi-squared, JS, and intersection kernels, respectively.

	S1	S2	S3	S4	S5	S6
S1	**96,96 96**	3,3,3 3	0	1,1 1	0	0
S2	3,3 3	**96,96 96**	1,1 1	0	0	0
S3	2,2 2	2,2 2	**96,96 96**	0	0	0
S4	0	0	0	**97,96 97**	3,4 3	0
S5	0	0	0	2,4 2	**88,86 88**	10,10 10
S6	0	0	0	1,1 1	12,14 12	**87,85 87**

each local histogram with 500D and cuboid size 9×9×9. Average recognitions are 93.7%, 93.9% and 93.6% for chi-squared, JS and intersection kernel, respectively.

	S1	S2	S3	S4	S5	S6
S1	**95,96 96**	4,3 3	0,1 0	1,0 1	0	0
S2	3,3 3	**96,96 96**	1,1 1	0	0	0
S3	2,2 2	1,1 2	**97,97 96**	0	0	0
S4	0	0	0	**99,98 97**	1,2 3	0
S5	0	0	0	3,4 2	**89,87 88**	5,9 10
S6	0	0	0	1,1 1	13,13 12	**86,86 87**

TABLE IV. CONFUSION MATRIXES OF AKSVMS FOR THE KTH DATASET. CHI-SQUARED (TOP LEFT), JS (TOP RIGHT), AND INTERSECTION KERNELS (BOTTOM). ROWS ARE GROUND TRUTH, COLUMNS ARE MODEL RESULTS. BOXING(S1), HAND-CLAPPING(S2), HAND-WAVING(S3), JOGGING(S4), WALKING(S5), RUNNING (S6).

each local histogram with 700D and cuboid size 7×7×9. Average recognitions are 94.2%, 92.5% and 93.2% for chi-squared, JS, and intersection kernels, respectively.

	S1	S2	S3	S4	S5	S6
S1	**97,96 97**	2,3 2	1,1 1	0	0	0
S2	3,3 3	**96,96 96**	1,1 1	0	0	0
S3	2,1 2	1,2 2	**97,97 96**	0	0	0
S4	0	0	0	**98,97 97**	2,3 3	0
S5	0	0	0	3,4 2	**89,88 88**	8,8 10
S6	0	0	0	1,1 1	11,14 12	**88,85 87**

each local histogram with 500D and cuboid size 7×7×5. Average recognitions are 92.5%, 92.8% and 92.5% for chi-squared, JS, intersection kernels, respectively.

	S1	S2	S3	S4	S5	S6
S1	**96,96 96**	3,3 3	0	1,1 1	0	0
S2	3,3 3	**95,95 95**	2,2 2	0	0	0
S3	2,2 2	1,1 2	**97,97 96**	0	0	0
S4	0	0	0	**97,95 96**	3,5 4	0
S5	0	0	0	5,5 4	**86,82 84**	9,13 11
S6	0	0	0	1,1 1	15,15 14	**84,84 85**

IV. CONCLUSION

In the paper, we presented an action recognition system based on local STOE feature and AKSVMs. Using ST oriented filters, motion information is decomposed into STOE feature. Then, multiple feature channels are used to convert STOE features to descriptors. Finally, AKSVMs are employed as action classifiers. The experimental result certifies that the proposed system achieves better recognition accuracy.

ACKNOWLEDGMENT

This work is supported by National Nature Science Foundation of China (grant number 6127128).

REFERENCES

[1] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *Journal of the Optical Society of America*, vol. 2, no. 2, pp. 284-299, 1985.
[2] K. J. Cannons, J. M. Gryn, and R. P. Wildes, "Visual tracking using a pixelwise spatiotemporal oriented energy representation," in *Proc. European Conference on Computer Vision*, 2010, pp. 511-524.
[3] I. Laptev and T. Lindeberg, "Local descriptors for spatiotemporal recognition," in *Spatial Coherence for Visual Motion Analysis*, 2006, pp. 91-103.
[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005, pp. 65-72.
[5] G. Willems, T. Tuytelaars, and L. V. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Computer Vision-ECCV*, 2008, pp. 650-663.
[6] K. G. Derpanis, M. Sizintsev, and K. Cannons, "Efficient action spotting based on a spacetime oriented structure representation," in *Proc. the IEEE Computer Vision and Pattern Recognition*, 2010, pp. 1990-1997.
[7] K. G. Derpanis and J. M. Gryn, "Three-dimensional nth derivative of Gaussian separable steerable filters," in *Proc. the International Conference on Image Processing*, 2005.
[8] A. Torralba, K. Murphy, and W. Freeman, "Sharing features: Efficient boosting procedures for multiclass object detection," in *Computer Vision and Pattern Recognition*, 2004.
[9] M. Bregonzio, S. Gong, and T. Xiang, "Recognising action as clouds of space-time interest points," in *Proc. the IEEE Computer Vision and Pattern Recognition*, 2009, pp. 1948-1955.
[10] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. the IEEE International Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1-8.
[11] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1996-2003.
[12] A. Gilbert, J. Illingworth, and R. Bowden, "Fast realistic multi-action recognition using mined dense spatio-temporal feature," in *Proc. the International Conference on Computer Vision*, 2009, pp. 925-931.
[13] W. Brendel and S. Todorovic, "Activities as time series of human postures," in *Proc. European Conference on Computer Vision*, 2010, pp. 721-734.
[14] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2046-2053.
[15] Q. Le, W. Zou, S. Yeung, and A. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3361-3368.
[16] B. Li, M. Ayazoglu, T. Mao, and O. Camps, "Activity recognition using dynamic subspace angles," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3193-3200.

Jiangfeng Yang was born in Yunnan Province, China, in 1975. He received the B.S. degree in applied physics from the University of Yunnan, Kunming, in 1997, and the Master of engineering degree in computer software and theory from the Kunming University of Science and Technology, Kunming, in 2009. He is currently pursuing the Ph.D. degree with the Department of Communication and Information Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include action recognition, object recognition and classification.

Zheng Ma was born in Chengdu Province, China, in 1956. He received the B.S. degree from UESTC, in 1982. He is a professor and Vice President of UESTC. His research interests include signal processing, information security, image processing, spectral estimation, array signal processing, and information theory.