Action Chart: A Representation for Efficient Recognition of Complex Activities

Hyung Jin Chang[1]
hj.chang@imperial.ac.uk
Jiyun Kim[2]
jiyun07@snu.ac.kr
Jungchan Cho[2]
cjc83@snu.ac.kr
Songhwai Oh[2]
songhwai@snu.ac.kr
Kwang Moo Yi[2]
kmyi@snu.ac.kr
Jin Young Choi[2]
jychoi@snu.ac.kr

[1] Department of Electrical and Electronic Engineering, Imperial College London, London, United Kingdom

[2] Department of Electrical and Computer Engineering, ASRI Seoul National University, Seoul, Korea

Abstract

In this paper we propose an efficient method for the recognition of long and complex action streams. First, we design a new motion feature flow descriptor by composing low-level local features. Then a new data embedding method is developed in order to represent the motion flow as an one-dimensional sequence, whilst preserving useful motion information for recognition. Finally attentional motion spots (AMSs) are defined to automatically detect meaningful motion changes from the embedded one-dimensional sequence. An unsupervised learning strategy based on expectation maximization and a weighted Gaussian mixture model is then applied to the AMSs for each action class, resulting in an action representation which we refer to as *Action Chart*. The *Action Chart* is then used efficiently for recognizing each action class. Through comparison with the state-of-the-art methods, experimental results show that the *Action Chart* gives promising recognition performance with low computational load and can be used for abstracting long video sequences.

1 Introduction

Action recognition has been widely studied for decades and there are many successful approaches to recognize relatively simple actions [13]. Recently, more realistic and complex activity recognition tasks have been dealt with, such as internet videos [25], surveillance videos [6], human interactions [13, 22], group activities [13] and temporally composed action sequences [19]. However, the current status of the research on complex activities is in its initial phase and far from the recognition ability of human. In our work, we are interested in recognizing temporally very long, complex and diverse action streams. Examples

of such action streams are pop dances, pantomimes, monodramas, and cooking. Reliable and efficient recognition methods of this kind of complex action streams can be useful for various tasks, such as complex activity categorization, long video abstraction and similar activity-based video retrieval in YouTube.

To deal with the long and complex action streams, we need an efficient abstraction procedure from low level feature extraction to high level classification. The existing research related with this purpose includes motion feature extraction, motion segmentation, and action classification. As for the motion feature extraction, low-level local feature descriptors, such as HoG/HoF [16] and cuboids [5] can represent local motion changes, but they can not imply temporal ordering and arrangement of features in the action sequence [19]. Global spatio-temporal templates such as spatio-temporal shapes [1] and motion history [30] have been proposed to contain such temporal ordering of motions and represent human body pose changes along a temporal sequence. Fathi *et al*. [7] proposed mid-level motion features which are built from low-level optical flow information by a learning method. Sun *et al*. [24] proposed a local descriptor and holistic feature fusion method. These methods contain more motion information than local features, but they are not appropriate for very long action sequences since they require extensive memory and high computational cost. As for motion segmentation, zero-velocity based segmentation methods [21, 29, 30] have been proposed using the zero-velocity or zero-crossing points of the motion feature stream. These methods would be reluctant to be applied to long sequences because even small noises result in many false segments. As for action recognition, the dynamic time warping (DTW) algorithm and its variations [9, 12, 13] are successfully used for recognizing action classes. However the DTW based algorithms take polynomial time and memory complexity for finding the optimal nonlinear match between two feature flows. In addition, probabilistic state transition models, such as Hidden semi-Markov Models (HSMMs) [18] and Conditional Random Fields (CRFs) [20], have been used for modeling temporal structure, and [19, 25] modeled the temporal structure using latent SVM. However, it is hard to determine the number of atomic actions and the action states ahead in activity streams. Also too much computation is required to optimize each temporal model, so they are not appropriate for long action sequences.

In this paper, we propose a pipelined motion-information embedding structure from a high dimensional local feature flow to a low dimensional attentional motion spot flow in order to recognize long and complex action streams efficiently. Each step of the proposed method is focused on extracting distinctive motion information and filtering out noise. In order to reduce high dimensional action video sequences (>640x480x6000 frames) into simple representations while retaining the necessary information, we propose a new composite motion feature generation method by combining various conventional low-level local features. The composite features characterize local, holistic, and sequential motion changes with small memories. The 21-dimensional composite feature sequences are embedded into one-dimensional feature sequences with preserving motion characteristics by proposing a hierarchical embedding method.

The one-dimensional embedded feature sequence is utilized to catch distinctive motions. The distinctive motion instances are referred to as attentional motion spots (AMSs), which are automatically determined in our scheme. The AMSs appear in similar feature space-time locations for the same activity classes. We model the distribution of AMSs as a weighted Gaussian mixture model using expectation maximization (EM) in embedded feature space-temporal domain. The sketch of this model looks like a music chart, thus we name our representation as *Action Chart*, which is used for action class recognition. In order to test the

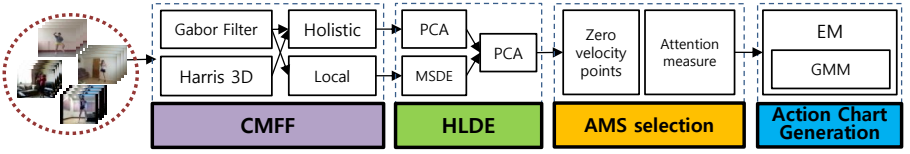


Figure 1: Overall scheme of building *Action Chart*

validity of our method, we have built a new dataset composed of various dancing sequences, which are difficult to be discriminated by the existing methods. Experimental results show that our method has good recognition performance with low computational complexity.

2 Proposed Method

The proposed method is composed of four steps: (1) composite motion feature flow generation using low-level local features, (2) hierarchical embedding of the feature flow into 1-dimensional (1-D) feature sequence, (3) AMS selection in the 1-D sequence, and (4) activity modeling and recognition using the AMSs. Figure 2 shows an overall scheme of the proposed method.

Composite Motion Feature Flow: As for the first step, the composite motion features are newly defined in this subsection by manipulating the low-level local feature information. The composite motion features extracted in each frame form a temporally sequential flow through the whole video frames, referred to as composite motion feature flow (CMFF).

As for the low-level local features to build CMFF, we use both the Gabor filtering detector [5] and the Harris-3D feature point detector with HoG/HoF descriptor [15]. The two detectors behave differently having their own strong points, and we use the two features together to take advantage of them both. As shown in [28], the Gabor filter detector [5] finds more features than Harris-3D [15], and filter responses for each feature point are available. Harris-3D detector with HoG/HoF descriptor has been shown to give good recognition performance for atomic actions.

The low-level local feature points are detected in a stack of images denoted by $I = \{I(x,y,t)|t = 1,...,N\}$. Feature point sets detected by the Gabor filter detector [5] are represented by $\{P(1),...,P(N)\}$ and each $P(t)$ contains not only feature locations p_x and p_y but also filter response values r (i.e. $P(t) = \{p_x^i(t), p_y^i(t), r^i(t)|0 \leq i \leq n_p(t)\}$ where $n_p(t)$ is the number of features detected at time t). In addition, the well-known bag-of-words approach is applied to the other local features which are detected by Harris-3D and described by HoG/HoF descriptor [15]. The descriptor codebook is generated by k-means clustering, and we set k as 1000 in the experiments. $h(t)$ is the normalized histogram of codebook memberships obtained by applying the codebook to the 100 frames centered at time t.

The CMFF ($\mathcal{M} = \{\mathcal{M}(t)|t = 1,...,N\}$) is composed of holistic ($\mathcal{M}_H$) and local ($\mathcal{M}_L$) motion features. The holistic motion feature $\mathcal{M}_H$ is composed of five measurements; motion intensity (m_I), motion extent (m_E), motion speed (m_S), motion change (m_C) and motion diversity (m_D). The motion intensity, extent, and speed represent quantitative motion property, and the change and diversity reflect qualitative motion property. At each t frame, the five measurements are obtained independently as follows:

- **Motion intensity** $m_I(t)$: The Gabor filter detector [5] finds pixels whose intensities have been changed by motion, so the number of detected feature points is proportional to motion intensity. We measure the motion intensity as $m_I(t) = n_p(t)$.

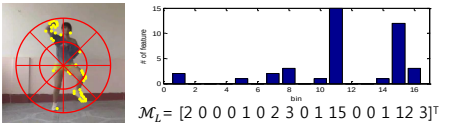


Figure 2: Local motion feature $\mathcal{M}_L$.

- **Motion extent** $m_E(t)$: Motion extent measures how widely current motion is occurring in the image space, and is measured by the spatial distribution of features: a norm of standard deviation of feature locations as $m_E(t) = (\frac{1}{n_p(t)}\{\sum_{i=1}^{n_p(t)}(p_x^i(t)-\mu_x(t))^2 + \sum_{i=1}^{n_p(t)}(p_y^i(t)-\mu_y(t))^2\})^{\frac{1}{2}}$, where $\mu_x(t) = \frac{1}{n_p(t)}\sum_{i=1}^{n_p(t)} p_x^i(t)$ and $\mu_y(t) = \frac{1}{n_p(t)}\sum_{i=1}^{n_p(t)} p_y^i(t)$.

- **Motion speed** $m_S(t)$: The Gabor filter detector [5] gives strong responses to motions with a similar period of the filter. On the other hand, much faster or slower motion induce small responses. We set the filter period as 15 frames/sec, which means waving hands twice in a second (in 30 FPS video) will give the strongest response. However, general human motions are not faster than this, so we assume that small responses are only caused by slower motions than the period. So the motion speed will be proportional to filter response. We measure the motion speed of current frame by sum of filter response values as $m_S(t) = \sum_{i=1}^{n_p(t)} r^i(t)$.

- **Motion change** $m_C(t)$: Motion change measures how much current short-time motion (centered at frame t) is changed comparing to the previous short-time motion (centered at frame $t-1$). The codebook histogram, $h(t)$, describes a short-time (100 frames) motion as a vector [15, 28]. So we measure the motion changes by chi-square distance between $h(t-1)$ and $h(t)$ as $m_C(t) = \chi^2(h(t-1),h(t)) = \sum_{i=1}^{k}\frac{(h^i(t-1)-h^i(t))^2}{h^i(t-1)+h^i(t)}$.

- **Motion diversity** $m_D(t)$: Having many non-zero bins in the histogram, $h(t)$, imply that many codebook words were used when obtaining $h(t)$, thus the motion at time t is composed of diverse local motions. So the non-zero codebook diversity is measured by the entropy of $h(t)$ as $m_D(t) = -\sum_{i=1}^{k} h^i(t)\log h^i(t)$.

Each measurement is a one-dimensional data sequence. By concatenating the five measurements $\mathcal{M}_H(t) = [m_I, m_E, m_S, m_C, m_D]$, the holistic motion feature $\mathcal{M}_H$ becomes a five-dimensional data sequence ($\mathcal{M}_H \in R^{5\times N}$).

The local motion feature ($\mathcal{M}_L(t) \in R^{16}$) represents the relative location distributions of local feature points, $p_{(x,y)}$, using a concentric 16-bin histogram method as shown in Figure 2. We place the center of the concentric circular bin at an estimated center of human body, and set a radius of the circle is the half of human height. The center position and the height can be estimated using foreground information or the human detection algorithm of Wang and Suter [29]. In this paper, we estimate the values using the local features' locational information of the previous 100 frames. A mean value of the feature locations is estimated as the center, and a mean of 100 maximum distances from the center is estimated as the radius. The estimated values are recalculated for each frame.

Finally, we compose the CMFF by concatenating the holistic and local motion features; $\mathcal{M}(t) = [\mathcal{M}_H(t), \mathcal{M}_L(t)] = [m_I(t), m_E(t), m_S(t), m_C(t), m_D(t), m_L^{[1,\ldots,16]}(t)]^T$ ($\mathcal{M}(t) \in R^{21}$). The measured raw data flow is too peaky because of noise, so we smooth the data flow using local polynomial regression fitting [3] with a low degree of smoothing (span=0.03).

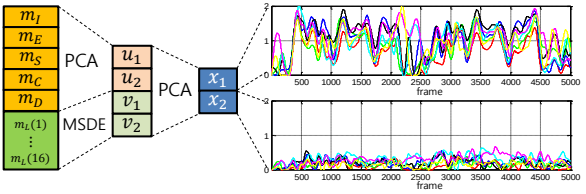


Figure 3: Concept of the proposed HLDE and the embedded data sequences of *"Gee"* by SNSD. Different color implies different people.

Hierarchical Low Dimensional Embedding: To avoid the curse of dimensionality in analyzing motion streams, it is necessary for the CMFF $\mathcal{M}$ to be embedded to a lower dimensional space. However $\mathcal{M}$ consists of two different information groups; $\mathcal{M}_H$ and $\mathcal{M}_L$. Each dimension of $\mathcal{M}_H$ implies independent and distinctive motion information, whilst the 16 dimensions of $\mathcal{M}_L$ represent only one motion information as a combination. In other words, the importance of each dimension is different.

To handle this problem, we propose a hierarchical low dimensional embedding (HLDE) method. First, we embed the five dimensional holistic feature vectors, $\mathcal{M}_H$, onto a two-dimensional space, $U = [u_1, u_2]^T$ using principal component analysis (PCA). We then simultaneously apply mean and standard deviation distance embedding (MSDE) [11] to reduce the local feature vectors $\mathcal{M}_L$ having 16 dimensions into two-dimensional vectors $V = [v_1, v_2]^T$. Then, we perform PCA again on the four-dimensional vectors, $W = [U;V] = [u_1, u_2, v_1, v_2]^T$, reducing it to two-dimensional feature vectors, $X = [x_1, x_2]^T$, as shown in Figure 3.

The proposed embedding method is aimed to reduce dimension while preserving meaningful and useful motion information. We measured how much motion information is preserved through each embedding step. We consider the amount of information retained in the component and auto-correlation among different actors in the same class. As a result, we choose only x_1 for the final action recognition (i.e. $X = x_1$). Because the x_1 includes most motion information (83.0% avg.) of actions and shows the highest auto-correlation value for the same class (32.9 times more than x_2).

Attentional Motion Spot Selection: Psychological study [31] reports that segmentation of ongoing activity into meaningful actions is essential for perception and small memory. The segmentation is strongly related to motion changes [31]. By mimicking the human perception mechanism, we propose a method to catch and focus on distinctive instances along the motion feature flow X. The motion feature flow X is a sequential data $X = \{x(t)|t = 1...N\}$ where $x(t)$ is chosen by the first principal component x_1 obtained in HLDE. We define the distinctive instances as attentional motion spot (AMS), and we use velocity (the first derivative) of X to find the AMS, which is similar to the human mechanism of using motion changes as a clue for segmentation. We define a zero-velocity points set $Z = \{z_1, z_2, ...\} = \{t|\Delta x(t) = x(t+1) - x(t) = 0\}$ and its *convexity index* $\xi(t)$ as

$$\xi(t) = \begin{cases} 1 & \Delta^2 x(t) \leq 0 \\ -1 & \Delta^2 x(t) > 0, \end{cases} \tag{1}$$

where $\Delta^2 x(t) = \Delta x(t) - \Delta x(t-1)$. The number of zero-velocity points is determined automatically. To avoid the false detection problem that other zero-velocity based methods [21, 29, 30] suffer from, we introduce an attention measure η at j^{th} zero-velocity point z_j defined as

$$\eta(z_j) = \left| \frac{x(z_j) - x(z_{j-1})}{z_j - z_{j-1}} \right| + \left| \frac{x(z_{j+1}) - x(z_j)}{z_{j+1} - z_j} \right|. \tag{2}$$

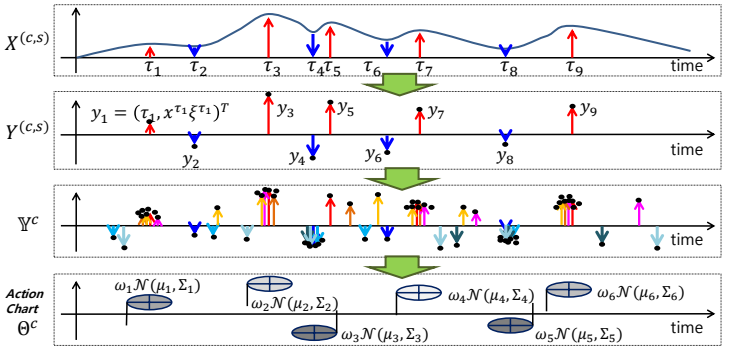


Figure 4: Illustration of AMS selection and *Action Chart* generation.

The η is used for filtering out noisy zero-velocity points by thresholding. Finally, we find an *attentional point* set $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\} = \{z_j | \eta(z_j) > \varepsilon\}$ (ε is 0.005 in the experiments) as shown in the first chart of Figure 4. n is the number of attentional points determined automatically. As shown in the second chart of Figure 4, the i^{th} AMS y_i is composed of two components; the temporal location, τ_i, and its corresponding feature value, $x(\tau_i)$, multiplied by convexity index $\xi(\tau_i)$ i.e. $y_i = [\tau_i, x(\tau_i)\xi(\tau_i)]^T$. For each action class $c \in 1...C$ and each actor $s \in 1...S^c$ ($S^c$is a number of actors in action class c), we define AMS set as $Y^{(c,s)} = [y_1^{(c,s)}, ...]^T$ and total AMS set as $\mathbf{Y}^c = [Y^{(c,1)}, \ldots, Y^{(c,S^c)}]^T$ which contains 2-dimensional random vector. Each motion stream $X^{(c,s)}$ is temporally aligned and normalized using DTW [27] in a preprocessing step. Then, each AMS set, $Y^{(c,s)}$, of each motion stream is generated independently.

Action Chart Generation and Recognition: We assume that each $\mathbf{Y}^c$ follows a weighted Gaussian mixture model (GMM) distribution in the feature space-time domain. The probability density function can then be written as

$$p(\mathbf{Y}^c|\theta^c) = \sum_{m=1}^{k^c} \omega_m^c p(\mathbf{Y}^c|\theta_m^c), \tag{3}$$

where $\omega_1^c, ..., \omega_{k^c}^c$ ($\omega_m^c \geq 0, m = 1, ..., k^c$, and $\sum_{m=1}^{k^c} \omega_m = 1$) are the *weights of each component*, each θ_m^c is a set of Gaussian parameters $\theta_m^c = \{\mu_m^c, \Sigma_m^c\}$ defining m^{th} component, and $\Theta^c \equiv \{\theta_1^c, ..., \theta_{k^c}^c, \omega_1^c, ..., \omega_{k^c}^c\}$. We define the Θ^c as *Action Chart* of action class c. In Figure 5, each component of Θ^c is represented as a shaded ellipse located at μ_m^c with size Σ_m^c. The darkness of each ellipse is proportional to corresponding weight ω_m^c.

To estimate each Θ^c, the expectation maximization (EM) algorithm is used. However, the basic EM algorithm has two major weaknesses; rough initialization can produce singularity and the user has to set the number of components. In order to avoid the singularity problem, we set all the covariance matrices of each action class to be equal (i.e. $\theta_m^c = \{\mu_m^c, \Sigma^c\}$), and we adopt the Figueiredo and Jain [8] algorithm for unsupervised parameter estimation. So, the component number of each action class (k^c) is adaptively selected as shown in Figure 5.

The class of the test action stream X^{test} is determined through maximum log-likelihood. The AMS set of X^{test} is obtained and represented as $Y^{test} = \{y_1^{test}, ..., y_{n^{test}}^{test}\}$. The class recognition is performed by matching the Y^{test} and the trained *Action Chart* Θ^c one by one,

$$\log p(Y^{test}|\Theta^c) = \log \prod_{i=1}^{n^{test}} p(y_i^{test}|\Theta^c) = \sum_{i=1}^{n^{test}} \log \sum_{m=1}^{k^c} \omega_g^c \mathcal{N}(y_i^{test}|\mu_m^c, \Sigma^c) \tag{4}$$

$$\widehat{c} = \underset{c}{\operatorname{argmax}} \{\log p(Y^{test}|\Theta^c)\}. \tag{5}$$

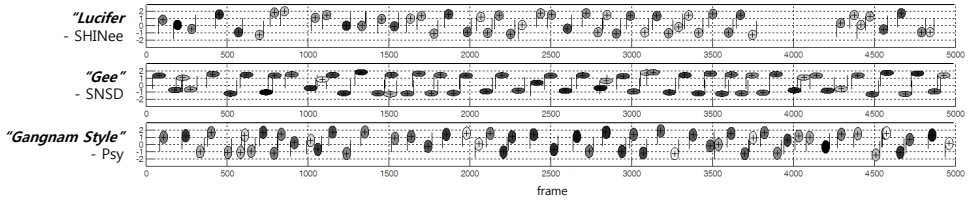


Figure 5: Generated *Action Charts* for the Pop-Dance dataset.

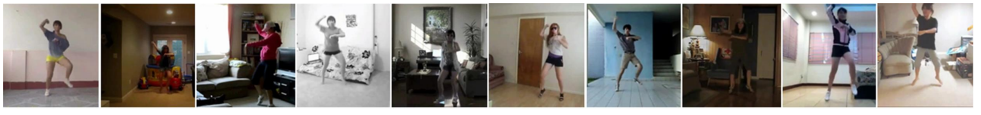


Figure 6: Sample frames of Pop-Dance dataset (Gangnam Style). Even people are dancing the same part of dance, they look different.

3 Experimental Results

We implemented our algorithm and the method by Niebles *et al*. [19] in Matlab for simulation with Intel Core i7 3.40GHz processor and 16.0GB RAM. We used the binaries provided by [15] to extract Harris-3D feature point and the HoG/HoF feature descriptors, and Matlab code for Gabor filter based feature detector were provided by the author of [5]. The VLFeat [26] and LIBSVM [2] libraries were used for the bag-of-words codebook and SVM.

Pop-Dance Dataset: Well-known action datasets such as KTH [23], Weizmann [1] and HMDB51 [14] are relatively short and contain only one action in a video clip. Also, the number of atomic motions of the Olympic Sports dataset [19] is still small (3 to 5) and the motions are relatively simple. Therefore, they are not appropriate for evaluating long and complex action sequence recognition algorithms.

We built a new dataset which contains motion sequences of people dancing following choreographies of pop songs. The dataset is composed of video clips downloaded from YouTube. Each person in the dataset dances differently in his/her own style to the same music. Also the dance motions show large variations depending on camera view point, human scale, appearance, clothes, shadow and illumination conditions as shown in Figure 6. The dataset contains 10 dances: *"You and I"*-IU, *"Goodbye Baby"*-MissA, *"Alone"*-Sistar, *"Twinkle"*-TTS, *"Be My Baby"*-Wonder Girls, *"Lupin"*-Kara, *"Electric Shock"*-Fx, *"Lucifer"*-SHINee, *"Gee"*-SNSD, and *"Gangnam Style"*-Psy. Each dance was performed by 10 different people. In total, the dataset is composed of 100 dancing video sequences of 100 different people. The average length of video clips is 6190 frames long. To the best of our knowledge, this is the longest action video clip of one person acting in the vision community.

Validation of Proposed Method: To verify the effects of the proposed CMFF and HLDE, we measured the classification performance of the proposed method under various configurations of features and embedding methods using the Pop-Dance dataset. Through this experiment, we can validate an importance of each component of the proposed method. As shown in Figure 7, without all holistic motion features ($\mathcal{M}_H$), large degradation in performance (21%) is shown, which implies the holistic features take a significant role in representing action characteristics. Among the holistic features, motion intensity(m_I) and motion speed (m_S) are shown to be influential to the performance. The result shows the best performance when using the all features as proposed in our paper. Also, to show the effect of

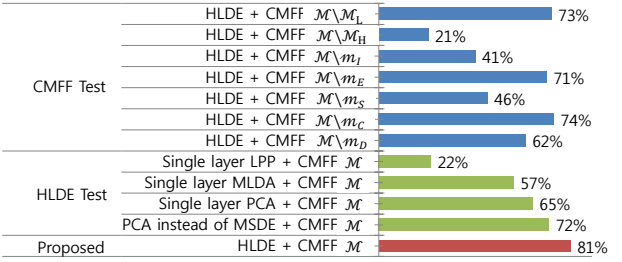
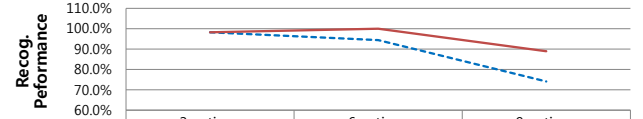


Figure 7: Performance of our method with different configurations. The 'single layer PCA+CMFF' means that PCA is applied to all 21 feature channels. The LPP and MLDA imply locality preserving projection [10] and multi-class linear discriminant analysis [17] respectively. The result shows that the proposed method outperforms all other configurations.



	3 actions	6 actions	9 actions
Niebles et al.	98.2%	94.4%	74.1%
Proposed	98.2%	100.0%	88.9%

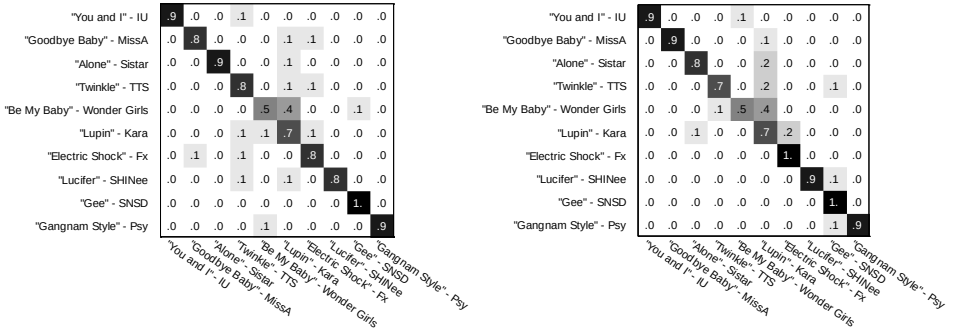
Figure 8: Recognition performance comparison using synthesized Weizmann dataset.

HLDE, we tested it with different dimension reduction schemes with CMFF.

Action Recognition Performance: The recognition performance of our method was compared to the other well-known methods in three ways. First, our method was tested with a set of synthesized complex actions using the Weizmann dataset [1], which is a new evaluation for complex actions proposed by [19]. Second, we used the proposed Pop-Dance dataset with the whole sequence as the query using 10-fold validation strategy. Third, we used the Pop-Dance dataset with only a part of the sequence as the query. The same codebook, generated beforehand for each dataset, was used for all methods compared.

Firstly, a synthesized set of complex action sequences is constructed by concatenating 3 simple motions from the Weizmann action database [1]: 'jump', 'wave' and 'jack'. In [19] only 6 complex action classes are generated using 3 simple motions, but we increased the number of complex action classes by allowing repetition of the 3 atomic motions. Figure 8 shows the recognition performance with respect to the number of atomic actions in the sequence compared to [19]. As the number of atomic actions increases, our method shows better recognition performance than [19].

Secondly, performance comparison results for the Pop-Dance dataset using the whole sequence as the query is shown in Table 1. We compared our method with four methods; SVM with CMFF (separate SVM classifiers were trained for each class using RBF kernel), DTW with CMFF (similar to methods used in [9, 12]), the method by Laptev *et al.* [15], and the state-of-the-art method by Niebles *et al.* [19]. We used a linear kernel for [19] and a χ^2 kernel for [15]. We obtained the best recognition performance as well as a very short computational time compared to the other methods as shown in Table 1. These results show that the proposed *Action Charts* effectively model each action sequence in an abstract manner. The efficiency of our method comes from the fact that we use only AMSs and not the whole data for evaluating the fitness. This is similar to looking at the "scores" of a song to determine which song a person is listening to, which can be done efficiently. Note that DTW achieves better recognition result than SVM. This is not surprising because the CMFF



(a) (b)

Figure 9: Confusion matrices of the proposed method for experiments on the Pop-Dance dataset using (a) the whole sequences and (b) small parts as the query

Algorithm	Performance (%)	Total Test Time (sec)
CMFF+HLDE+SVM	15	94818.0
CMFF+HLDE+DTW	69	18739.0
Laptev [15]	25	20.9
Niebles *et al.* [19]	66	31235.0
Proposed	**81**	**783.8**

Table 1: Recognition performance and time comparison with widely used methods.

itself is a temporal feature flow. A confusion matrix for the results of our method is shown in Figure 9(a).

Finally, our method can be applied not only to the case when the whole action sequence is given as a query, but also to the case when only small parts of the sequences are given. Recognizing with a partial action sequence is important for practical applications such as video retrieval. In our test setting, whole sequences were used for training and 100 random portions (with random length longer than 1000 frames, and random positions) were used for testing. We compared our method only with DTW since all other methods are not available for this kind of testing. Average performance is shown in Table 2 and Figure 9(b) is the confusion matrix for the recognition results of our method. Our method shows promising results both in recognition performance and computational time.

Unsupervised Action Abstraction: An unsupervised action abstraction is performed by concatenating frames around components with large weights of *Action Chart*. This is a reasonable way to create abstracts of videos, since the attentional parts are very similar to the human concept of distinctive points in the video. We have experimentally validated this by comparing automatically found attentional points with manually indicated distinctive parts of each dance. This evaluation method is similar to the methodologies used for psychological studies [31]. Comparison with the human annotations coincides by 74.4($\pm$7.6)%, being quite similar. This shows that our abstraction method is reasonable. The abstraction results

Algorithm	Performance (%)	Avg. Recog. Time per One Test Video (sec)
DTW	24	132.9
Proposed	**83**	**121.9**

Table 2: Recognition performance and computation time for recognizing one cropped video.

will be released as a supplementary material. The supplementary material accompanying the paper [1] provides a video with the abstraction results obtained in two dance sequences.

4 Conclusions

In this paper we proposed a novel method for recognizing long and complex action streams such as dance videos. We proposed a new motion feature flow descriptor generation method using local features and a hierarchical low-dimensional embedding method in order to represent the motion changes as one dimensional feature. Attentional motion spots can be adaptively detected based on significant temporal changes in motion flow. Feature space-temporal groups of AMSs have been modeled as weighted Gaussian mixture models, and the final representation has been termed an *Action Chart*. In order to validate the proposed method, we generated a new complex action dataset; the Pop-Dance dataset. The experimental results showed that the *Action Chart* could give a promising recognition performance with a very low computational load. Furthermore it could be used for abstracting a long video sequence aims. Our method can contribute to recognizing repetitive sequential activities (e.g. workplace safety, retail fraud detection or sweethearting, and product quality assurance) and sequentially combined action tasks (e.g. sign language and cooking menu), which are our future research.

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, oct. 2005.

[2] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[3] William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):pp. 829–836, 1979.

[4] T. Darrell and A. Pentland. Space-time gestures. In *CVPR*, pages 335 –340, jun 1993.

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, October 2005.

[6] Helmut Grabner Fabian Nater and Luc Van Gool. Temporal relations in videos for unsupervised activity analysis. In *BMVC*, 2011.

[7] Alireza Fathi and Greg Mori. Action recognition by learning mid-level motion features. In *CVPR*, june 2008.

[8] Mario A.T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(3):381–396, 2002.

[1] Also available at http://www.youtube.com/watch?v=HeJJJnehWa0

[9] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury. A “string of feature graphs" model for recognition of complex activities in natural videos. In *ICCV*, pages 2595–2602, 2011.

[10] Xiaofei He and Partha Niyogi. Locality preserving projections. In *In Advances in Neural Information Processing Systems 16*. MIT Press, 2003.

[11] Yoonho Hwang, Bohyung Han, and Hee-Kap Ahn. A fast nearest neighbor search algorithm by nonlinear embedding. In *CVPR*, June 2012.

[12] Nazli Ikizler and Pinar Duygulu. Human action recognition using distribution of oriented rectangular patches. In *Worshop on Human Motion, LNCS*, pages 271–284, 2007.

[13] J.K.Aggarwal and M.S.Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, 43(3), April 2011.

[14] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, pages 2556 –2563, nov 2011.

[15] I. Laptev, M Marszalek, C Schmid, and B Rozenfeld. Learning realistic human actions from movies. In *CVPR*, june 2008.

[16] Ivan Laptev. On space time interest points. *IJCV*, 64(2/3):107–123, 2005.

[17] Tao Li, Shenghuo Zhu, and Mitsunori Ogihara. Using discriminant analysis for multi-class classification: and experimental investigation. *Knowledge and Information Systems*, 10(4):453–472, 2006.

[18] Pradeep Natarajan and Ramakant Nevatia. Coupled hidden semi markov models for activity recognition. In *WMVC*, 2007.

[19] Juan Carlos Niebles, Chih-Wei Chen, , and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, September 2010.

[20] A. Quattoni, S. Wang, L. p Morency, M. Collins, T. Darrell, and Mit Csail. Hidden-state conditional random fields. *IEEE TPAMI*, 2007.

[21] Yong Rui and P. Anandan. Segmenting visual actions based on spatio-temporal motion patterns. In *CVPR*, 2000.

[22] M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *CVPR*, june 2006.

[23] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004.

[24] Ju Sun, Xiao Wu, Shuicheng Yan, Loong-Fah Cheong, Tat-Seng Chua, and Jintao Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, june 2009.

[25] Kevin Tang, Li Fei-Fei, and Daphne Koller. Learning latent temporal structure for complex event detection. In *CVPR*, June 2012.

[26] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[27] Ashok Veeraraghavan, Amit K. Roy-Chowdhury, and Rama Chellappa. Matching shape sequences in video with applications in human movement analysis. *IEEE TPAMI*, 27(12):1896 –1909, dec 2005.

[28] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[29] Liang Wang and David Suter. Learning and matching of dynamic shape manifolds for human action recognition. *IEEE Transaction on Image Processing*, 16(6), june 2007.

[30] Daniel Weinland. *Action Representation and Recognition*. PhD thesis, Institut National Polytechnique De Grenoble, oct 2008.

[31] Jeffrey M. Zacks and Khena M. Swallow. Event segmentation. *Current Directions in Psychological Science*, 16(2):80–84, 2007.