2-2016

Human activity prediction by mapping grouplets to recurrent self-organizing map

Qianru SUN
Singapore Management University, qianrusun@smu.edu.sg

Hong LIU

Mengyuan LIU

Tianwei ZHANG

Citation

SUN, Qianru; LIU, Hong; LIU, Mengyuan; and ZHANG, Tianwei. Human activity prediction by mapping grouplets to recurrent self-organizing map. (2016). *Neurocomputing*. 177, 427-440.
Available at: https://ink.library.smu.edu.sg/sis_research/4453

Neurocomputing 177 (2016) 427–440

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Human activity prediction by mapping grouplets to recurrent Self-Organizing Map

Qianru Sun [a], Hong Liu [a,*], Mengyuan Liu [a], Tianwei Zhang [b]

[a] Key Laboratory of Machine Perception, Shenzhen Graduate School, Peking University, Beijing 100871, China
[b] Nakamura Laboratory, Department of Mechano-Informatics, School of Information Science and Technology, The University of Tokyo, Tokyo 113-8685, Japan

ARTICLE INFO

Article history:
Received 7 February 2015
Received in revised form
18 November 2015
Accepted 23 November 2015
Communicated by Marco Cristani
Available online 3 December 2015

Keywords:
Human activity prediction
Spatio-temporal interest points
Mean shift
Recurrent Self-Organizing Map

ABSTRACT

Human activity prediction is defined as inferring the high-level activity category with the observation of only a few action units. It is very meaningful for time-critical applications such as emergency surveillance. For efficient prediction, we represent the ongoing human activity by using body part movements and taking full advantage of inherent sequentiality, then find the best matching activity template by a proper aligning measurement.

In streaming videos, dense spatio-temporal interest points (STIPs) are first extracted as low-level descriptors for their high detection efficiency. Then, sparse grouplets, i.e., clustered point groups, are located to represent body part movements, for which we propose a scale-adaptive mean shift method that can determine grouplet number and scale for each frame adaptively. To learn the sequentiality, located grouplets are successively mapped to Recurrent Self-Organizing Map (RSOM), which has been pre-trained to preserve the temporal topology of grouplet sequences. During this mapping, a growing RSOM trajectory, which represents the ongoing activity, is obtained. For the special structure of RSOM trajectory, a combination of dynamic time warping (DTW) distance and edit distance, called DTW-E distance, is designed for similarity measurement. Four activity datasets with different characteristics such as complex scenes and inter-class ambiguities serve for performance evaluation. Experimental results confirm that our method is very efficient for predicting human activity and yields better performance than state-of-the-art works.

1. Introduction

Recent years have witnessed the fast progress in human activity recognition [1–7]. Great advances in this field enable the computer vision system to recognize complex human activities in many applications. One important and interesting application is to predict human activities or imminent events from early video observations. Related systems range from user-friendly service machine to emergency monitor. In a smart room, if people's activity can be predicted by a user-friendly machine, the response modules will provide corresponding services or help automatically. In terms of the surveillance system, if it can recognize ongoing criminal behaviors and raise an alarm in time, it will be more helpful than just identifying the events after objects are destroyed or people are killed.

Although human activity prediction is important and interesting, it is still an open area. Firstly, most of the previous methods for traditional action recognition focus on the study of single human actions such as walking, running, and waving hands [8–10]. In our case, it is not necessary to predict short-duration actions, but the long-duration activity composed by multiple action stages [16], e.g., getting close to a person and giving a punch [11], making coffee with “take cup” – “pour coffee” – “pour milk” – “pour sugar” – “spoon sugar” – “stir coffee” [12], etc. The goal of prediction is to infer the high level activity category by observing only a few action units. Secondly, previous works use fully observed activities to train after-event classifiers and usually construct global descriptors [9,10,13–16]. This makes their models unsuitable for representing the activity in different observation stages. More efficient represent models for the unfinished activity should be developed.

Ryoo's work [17] is one of the few works explicitly focused on modeling unfinished activity (also called ongoing activity). It utilized spatio-temporal interest points (STIPs) as basic descriptors, and proposed two extensions of Bag-of-Visual-Words (BoVW) paradigm, i.e., Dynamic BoVW and Integral BoVW, to encode ongoing activity in a dynamic way. However, both models

[*] Corresponding author.
E-mail address: hongliu@pku.edu.cn (H. Liu).

http://dx.doi.org/10.1016/j.neucom.2015.11.061
0925-2312/© 2015 Elsevier B.V. All rights reserved.

obtained modest performances since BoVW ignores the discriminative contexts of local features. To solve this problem, we proposed the Recurrent Self-Organizing Map trajectory (RSOM trajectory) in our previous paper [18]. STIPs are successively mapped to the RSOM network to find the best matching units (*bmus*). Each *bmu* is a network unit whose weight vector is most similar to the input. Then, *bmus* are used to compose a growing RSOM trajectory for activity representation. This model achieved better performance than BoVW by encoding the temporal contexts of local features (STIPs). However, dense STIPs are mapped to RSOM without any feature selection, and very long RSOM trajectories take high computational costs when matching templates.

Compared with [18], this paper aims to obtain more compact and efficient RSOM trajectory. Instead of dense STIPs, sparse STIP grouplets are used for RSOM mapping. The key insight is that an activity can be divided into a sequence of body movements, and each salient movement can generate a grouplet of STIPs around or near a specific body part, e.g., eating a banana consists of picking up the banana (STIPs on the hands and arms), peeling (STIPs on the hands), putting it into mouth (STIPs on the hands, arms, and mouth), and biting it (STIPs on the mouth). We propose a scale-adaptive mean shift algorithm to locate grouplets and design a one-by-one locating method to determine the grouplet number on each frame. Located grouplets are then mapped to RSOM to generate RSOM trajectory. Before presenting our method in detail, we first introduce some relevant works of activity prediction.

1.1. Related works

Ryoo's work [17] defined that the goal of activity prediction is similar to early recognition, i.e., recognizing activities from early observed data. Most of the related works improve recognition technologies for activity prediction by leveraging the contexts in activity sequences [17,22–25,27,26,29,30].

Lv and Nevatia proposed a graph model called Action Net by considering the contexts between key poses and viewpoints [22]. Their method depends on accurate silhouette detection, which is a challenging problem in uncontrolled environments. Instead of silhouette, Ryoo [17] used local STIPs as basic descriptions. He proposed two variants of BoVW model to represent unfinished human activity, then employed standard classifiers for recognition. Cao et al. [23] extended Ryoo's work to recognize partially observed videos, and formulated the prediction in a probabilistic framework. In this work, the action sequence has to be *manually divided into temporal segments*. Raptis and Sigal [24] modeled the human action as a sparse sequence of discriminative “key frames”, depicting key poses. Here, “key frames” are encoded using a spatially localizable representation in which the components are *learned from weak annotations*. Hamid et al. [25] investigated modeling activity sequences in terms of their constituent subsequences, named event *n-grams*. Their model depends on carefully identified objects and manually annotated event-vocabulary. Compared with [23–25], our model has the ability of detecting salient body movements without any manual annotation.

Kitani et al. [26] formulated activity prediction as a decision-making process and attempted to predict people's walking path in certain environments based on the most common paths extracted from training data. Their method focuses on moving objects under distant views, while we aim to study the close-range movements whose appearances are very different. Very recently, Zhang and Parker [27] introduced a bio-inspired activity prediction approach by encoding the temporal dependencies of 3D skeleton trajectories. They use 3D motion capture systems e.g., OpenNI [28], to obtain skeletal data (with labeled joints) for representation. Their method is based on anatomical planes which can not handle videos. Hoai and Torre [29] addressed early event detection in videos using active training. They simulated frame-by-frame data as training series, then extended Structural Output SVM to accommodate the series nature. It was successfully used for inferring different kinds of time series, such as facial expressions and human activities. Nevertheless, in our work, human activities are represented by body part movement sequences, which are not pure time series. They contain not only the time-varying information of frames but also the spatial structure on each single frame. For properly measuring them, we introduce a novel distance in Section 3.3.

Most recently, Li et al. [30] proposed a generalized prediction framework. The long-duration complex activity is segmented into semantic units in terms of atomic actions, then activity prediction is converted to sequence classification. This method is useful for activities with deep hierarchical structure and repetitive structure, but not for the activities with shallow structure. Comparatively speaking, our model relies on general movement appearances, which are not sensitive to activity structure types.

In other AI fields, predicting agent behaviors or events has also been studied extensively. Neumany and Likhachev [31] proposed a generalization algorithm that allows the robot to infer new solutions with approximate preferences on missing information. Neill et al. [32] developed a new Bayesian method to monitor the evolution of the disease, which can predict the location of disease caused by emerging disease outbreaks. By exploiting the collective information of entire batches of spam e-mails, Haider et al. [33] introduced an effective method to predict jointly generated spam e-mails. Exploiting financial time-series, such as time-index [34], they predicted future financial data. These techniques, however, are not suitable for the prediction of human activity, since they can only infer the microvalue of next time stamp rather than recognize a macroevent or an activity class.

1.2. An overview of the proposed method

The main idea of this paper is to represent ongoing human activity by a sequence of body part movements considering their inherent sequentiality, then find the best matching template by a proper aligning measurement. Specifically, “body part movement” refers to STIP grouplet, and “inherent sequentiality” is encoded in RSOM trajectory. An example of RSOM trajectory representation is illustrated in Fig. 1. At the recognition stage, a combination of dynamic time warping (DTW) distance and edit distance, called DTW-E distance, executes the “aligning measurement”.

Grouplet: Poselet models have been successfully used in human action detection [35] and human parsing [36]. A similar parts model showed good performances in recognizing human actions [37], indicating that part templates can provide a rich description of human actions. The underline motivation of these works is that *salient appearances/movements can generate discriminative feature clusters near body parts*. In this paper, we search for feature clusters in the location space of STIPs, and call the results STIP grouplets. Compared with the annotation guided search in [35] and the manual multi-scale regions in [36], our search method and scale adaption method are fully automatic.

To search for STIP grouplets, we encounter two problems. First, different frames contain different grouplets (i.e., clusters) both in number and appearance. Traditional clustering algorithms, such as k-means, have to be assigned a reasonable cluster number k in prior, which is very difficult for online input frames. In this paper, we use a flexible clustering algorithm – mean shift [19]. Based on mean shift, we design a one-by-one search method to adaptively determine the cluster number for each input frame, see Section 2.3.

The second problem is the scale variance in real-world videos. Previous researchers proposed to adjust the bandwidth of mean shift kernel to guarantee mean shift convergence on multi-scale

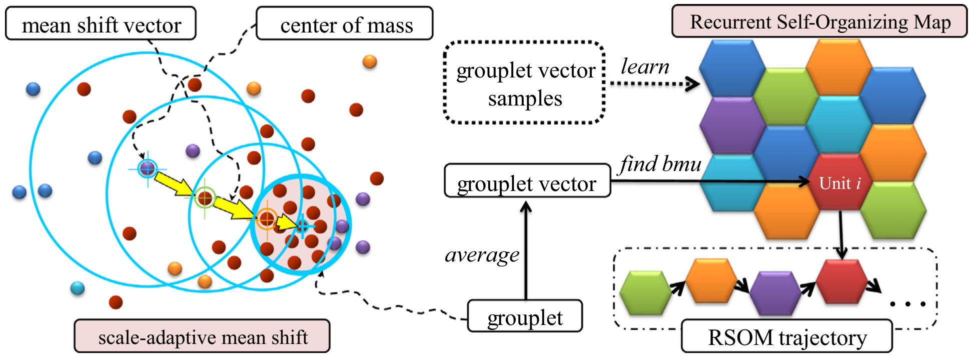


Fig. 1. An example of RSOM trajectory representation. First, an offline clustering algorithm is pre-processed, and STIPs in the same cluster are assigned the same color. Then, scale-adaptive mean shift searches for the grouplet of clustered points. Next, the points inside the grouplet, i.e., the bold blue circle, are quantized to be a grouplet vector by average pooling. After the grouplet vector finds its best matching unit (*bmu*) on a pre-trained RSOM, RSOM trajectory grows itself to include this *bmu* and represent the current observation. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

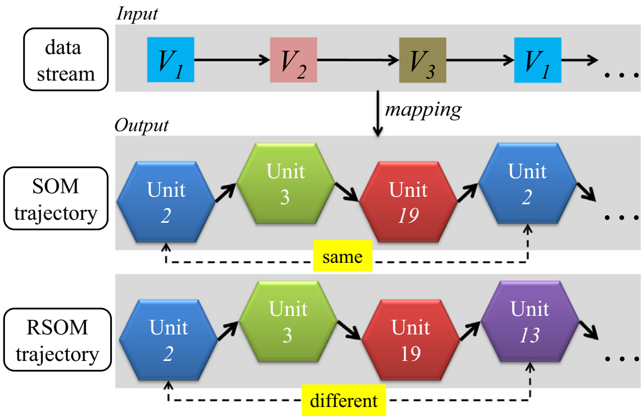


Fig. 2. The difference between SOM trajectory and RSOM trajectory.

features [38,39]. Furthermore, they demonstrated that the bandwidth of mean shift kernel can be regulated by the scale of image features and the distribution variation of feature labels inside shifting windows. Following these proposals, we use the *spatial scale of features* in combination with the *feature label variance* to adjust kernel bandwidth. The goal is to make mean shift converge to the cluster of feature points which are "dense" in spatial location and "similar" in appearance, see more details in Section 2.2.

RSOM trajectory: In Fig. 1, although grouplet vector is the averaged vector of point features, its dimension is still high. If a long sequence of grouplet vectors is directly used for representation, the computational cost of recognition will be very high. In [18], we proposed to use a time-critical model – RSOM [40] for reducing dimensions. RSOM successfully generalizes the properties of Self-Organizing Map (SOM) [41], and moreover, it involves data feedback to learn the temporal contexts of sequential data. Fig. 2 presents SOM trajectory and RSOM trajectory, where the same data with distinct input orders make a difference for RSOM but not for SOM.

In [18], we demonstrated that trajectory representation encodes body part movements in a sequential manner, and it is very successful when using completed samples for training and unfinished samples for prediction. In this paper, we inherit the use of RSOM trajectory. The innovation is that mapping object STIPs are replaced by STIP grouplets to get compact trajectories. Grouplet is a combination of different STIPs. To accommodate the diversity brought by this combination, we adopt a recursive scheme to adjust neighborhood size in RSOM learning, i.e., exploiting more neighbors to learn more diverse data [43].

DTW-E distance: Each video frame contains multiple inner-frame grouplets, where "inner-frame" means "on the same frame". On successive frames, all inner-frame grouplets are connected to be a sequence of inter-frame grouplets, where "inter-frame" means "on different frames". An example is given in Fig. 3 to show the relationship between "inter-frame" and "inner-frame". After mapping all grouplets to RSOM, resulting RSOM trajectory (inter-frame trajectory) has a special structure. Inter-frame trajectory encodes the pace variation of activities, while its component, i.e., inner-frame trajectory, encodes the spatial arrangement of active body parts at one point in time. Therefore, the measurement of RSOM trajectory should be carefully selected.

Various schemes such as HMM distance, Longest Common Sub-Sequence (LCSS), PCA+Euclidean, and dynamic time warping (DTW) distance were compared for trajectory measurement in [42]. The HMM distance needs to train a statistical model for each trajectory. LCSS is more concerned with the similarity of data shapes, and also requires exhaustive parameter setting [44]. PCA+Euclidean distance [45] needs equal-length data, which is impractical to ever-changing trajectories. DTW distance is an alignment-based measurement, which has been successfully used in series data mining [46] and sign language recognition [47]. In our case, DTW distance has the advantage of allowing some stretching flexibility to accommodate the temporal pace inconsistency. Hence, it is utilized to measure the cost of inter-frame trajectory alignment, where the inner-frame trajectories on single frames act as aligning elements.

To properly measure the distance between inner-frame trajectories, we use the edit distance which computes the minimum number of single-character edits (insertion, deletion, and substitution) required to change one word into another [48]. The motivation is that inner-frame trajectory encodes the spatial structure of body part movements, and computing the distance without ignoring the inner structure is similar to the matching process of alphabetical strings. This distance has shown high efficiency to compute weighted costs in body movement alignment [49].

After embedding edit distance into the alignment process of DTW, we get the hierarchical measurement called DTW-E distance [18]. If we regard RSOM trajectories as textual sentences, then DTW-E distance is the cost of aligning two sentences word-by-word.

The rest of this paper is organized as follows. In Section 2, we develop a scale-adaptive locator of grouplet based on mean shift and propose a one-by-one locating method to determine the grouplet number adaptively. Section 3 shows how to capture the temporal contexts of grouplets by learning RSOM. Then, we generate the RSOM trajectory based on a pre-trained RSOM and

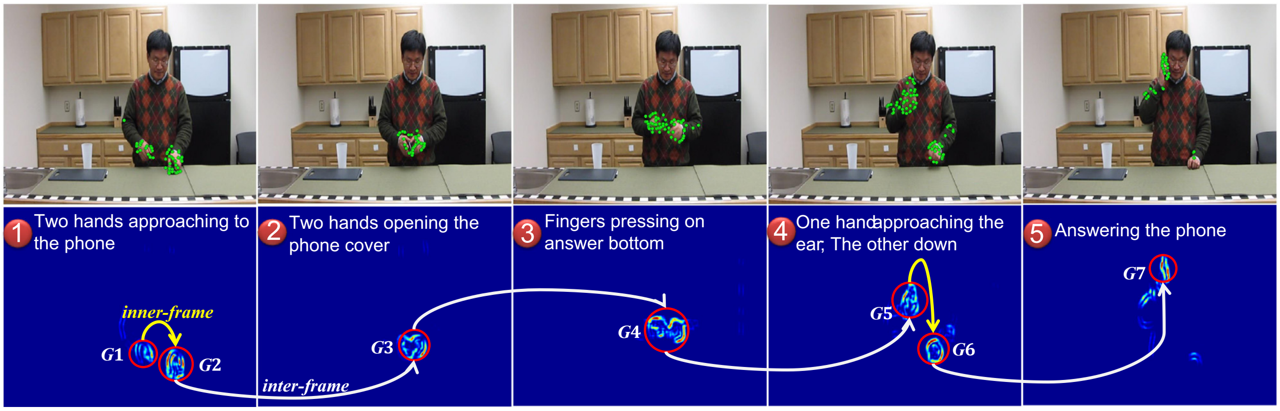

1 Two hands approaching to the phone
2 Two hands opening the phone cover
3 Fingers pressing on answer bottom
4 One handapproaching the ear, The other down
5 Answering the phone
inner-frame
inter-frame
G1
G2
G3
G4
G5
G6
G7

Fig. 3. Some salient movements of "answering a phone": picking up, opening cover, then picking to the ear. Green points are STIPs. Red circles stand for grouplets. Yellow curves connect inner-frame grouplets, while white curves connect inner-frame grouplets to form a global inter-frame sequence. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

introduce the computation of DTW-E distance. Experiments and discussions on four activity datasets are presented in Section 4, followed by the conclusions in Section 5.

2. Locating grouplets by scale-adaptive mean shift

This section introduces the notation for STIP description. Then, it presents how to adjust kernel scale to make mean shift converge to STIP grouplets. Finally, multiple STIP grouplets are located one-by-one on each frame.

2.1. Notation

In the video stream, we extract STIPs using popular feature detectors such as [9,10], then denote the point set extracted on frame t as $\mathcal{P}_t = \{\boldsymbol{p} = (\boldsymbol{x}; \boldsymbol{v}; \varphi_s, \varphi_t; l)$. Note that "point" in this paper refers to "STIP".

In the notation of $\boldsymbol{p}$, $\boldsymbol{x}$ represents the location vector on frame t, $\boldsymbol{v}$ is the high-dimension feature vector extracted from a 3D cuboid around this point, φ_s, φ_t are respectively the spatial scale and temporal scale computed by feature detector, and l is the clustering label obtained by assigning the feature vector to the closest entry in a "visual vocabulary". The "vocabulary" is learned offline by clustering algorithms such as k-means in a large set of sample feature vectors. When new points come online, their labels are rapidly computed. Mean shift can move efficiently to the grouplet based on points' locations and labels.

2.2. Scale-adaptive mean shift

The key of mean shift iteration is the computation of an offset from location $\boldsymbol{x}$ to a new location $\boldsymbol{x}'$ [19]. It is an iterative method starting with a random initial position $\boldsymbol{x}$. The kernel function K is radially symmetric and non-negative. It determines the weights of nearby points for re-estimating the mean value. The weighted density mean determined by K is computed as:

$$m(\boldsymbol{x}) = \frac{\sum_{\boldsymbol{x}_i \in \mathcal{N}} \mathrm{K}(\boldsymbol{x}_i - \boldsymbol{x}) \cdot \boldsymbol{x}}{\sum_{\boldsymbol{x}_i \in \mathcal{N}} \mathrm{K}(\boldsymbol{x}_i - \boldsymbol{x})}, \tag{1}$$

where $\mathcal{N}$ is the set of points near $\boldsymbol{x}$, and each $\boldsymbol{x}_i \in \mathcal{N}$ satisfies $\mathrm{K}(\mathbf{x}_i - \mathbf{x}) \neq 0$. Then, the objective of mean shift is to make $m(\boldsymbol{x}) \rightarrow \boldsymbol{x}$ until $m(\boldsymbol{x})$ converges.

Typically, Gaussian kernel is $K_h(\boldsymbol{x}_i - \boldsymbol{x}) = g(\| \frac{\boldsymbol{x}_i - \boldsymbol{x}}{h} \|^2)$, where K_h is called *scaled kernel*, and the kernel bandwidth h is crucial to the convergence. If kernel bandwidth is too large, the searching window would contain noise points, causing the searcher to become more easily distracted by background clutter. Besides, a kernel window that is too large may cause convergence to an area among multiple modes, rather than converging to one mode. In contrast, choosing a kernel bandwidth that is too small can "roam" around on a likelihood plateau around the mode, resulting in poor saliency localization. Hence, there is always a trade-off between the biases of the estimator. The rest of this section explains how to adjust the kernel bandwidth h in a self-adaptive manner.

Section 1.2 shows that we need to locate clustered points which are not only "dense" in spatial location but also "similar" in feature appearance. Accordingly, kernel density estimation function $F = \frac{1}{|\mathcal{N}|} \sum_{i=1}^{|\mathcal{N}|} \mathrm{K}_h(\boldsymbol{p}_i - \boldsymbol{p})$ should meet following rules:

1. if $\| \boldsymbol{x}_i - \boldsymbol{x} \| < \| \boldsymbol{x}_j - \boldsymbol{x} \|$, then $\mathrm{F}(\boldsymbol{p}_i) > \mathrm{F}(\boldsymbol{p}_j)$,
2. if $\| \boldsymbol{v}_i - \boldsymbol{v} \| < \| \boldsymbol{v}_j - \boldsymbol{v} \|$, then $\mathrm{F}(\boldsymbol{p}_i) > \mathrm{F}(\boldsymbol{p}_j)$,

where $\boldsymbol{x}$ and $\boldsymbol{v}$ are the mean of location space and feature space, respectively. Rules above indicate that points close to the mean in location space or feature space are the direction of shifting.

Based on the rules above, we obtain Eqs. (2) and (3) that adjust the kernel bandwidth in two steps: (1) Eq. (2) makes the bandwidth keep pace with the spatial scale change of center points, and leads to a group of "dense" points; (2) Eq. (3) satisfies the rule of "similar" by estimating the variation of point labels

$$\begin{cases} h_s = h_{prev} \cdot \dfrac{\varphi_{hs}}{\varphi_{prev}} & (2) \\ h = h_s \cdot e^{1 - \frac{\sigma_{hs}^2}{\sigma_{prev}^2}} & (3), \end{cases}$$

where the notations are as visualized in Fig. 4; h_{prev}, h_s and h are the bandwidths of previous window $\mathcal{N}_{prev}$, intermediate window $\mathcal{N}_{hs}$ and final window $\mathcal{N}_h$; φ_{hs}, φ_{prev} represent the spatial scales of center points in $\mathcal{N}_{hs}$ and $\mathcal{N}_{prev}$.

In Eq. (3), $\sigma_{hs}^2, \sigma_{prev}^2$ denote the feature variances inside $\mathcal{N}_{hs}$ and $\mathcal{N}_{prev}$. The following formula takes σ_{hs} as an example:

$$\sigma_{hs}^2 = \frac{1}{|\mathcal{N}_{hs}|^2} \sum_{\boldsymbol{p}_i \in \mathcal{N}_{hs}} \sum_{\boldsymbol{p}_j \in \mathcal{N}_{hs}} (l_i - l_j)^2, \tag{4}$$

where l_i and l_j are the feature labels of $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$. Then, ${\sigma_{prev}}^2$ corresponds to $\mathcal{N}_{prev}$ in Eq. (4). Fig. 4 presents an example of adjusted bandwidth compared with a fixed one. In Fig. 4(b), it is notable that if window $\mathcal{N}_{hs}$ gathers a bigger proportion of heterogeneous points than $\mathcal{N}_{prev}$, the bandwidth h_s will shrink to h to exclude heterogeneous points. Consequently, window $\mathcal{N}_{hs}$ is shifted to a smaller region $\mathcal{N}_h$, which contains a bigger proportion of same-labeled points.

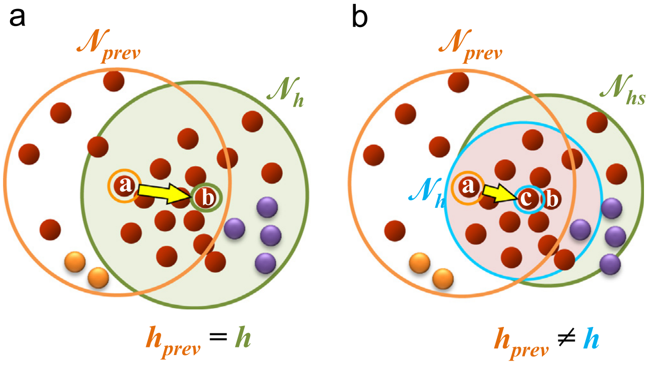


Fig. 4. The iteration in (a) uses a fixed h. Yellow arrow is a mean shift vector. Shifting windows in (b) are adjusted by our scale-adaptive method. The adjustment of $\mathcal{N}_{prev}$ to $\mathcal{N}_{hs}$ is based on the mean shift vector and center point's spatial scale φ_s, after which adjustment of $\mathcal{N}_{hs}$ to $\mathcal{N}_h$ is controlled by label variances inside $\mathcal{N}_{prev}$ and $\mathcal{N}_{hs}$. In the second adjustment, the center of mass slightly shifts from b to c. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Due to the non-convexity of many tasks, mean shift iterations might get trapped in poor local optima. Hence, we repeat 10 times in every search with randomly initialized center points. Since points are sparsely distributed, a small number of iterations are sufficient to converge in each run. Finally, the lumped grouplet that has the most converging votes is selected as the optima.

In summary, scale-adaptive mean shift searches for STIP grouplets based on both the location and appearance of STIPs. Each grouplet is a set of converged STIPs, and it is compact and stable. Moreover, the proposed scale adjustment guarantees the efficiency for handling video scale changes.

2.3. Multiple grouplets

On each frame, there are usually multiple movements caused by different body parts. Hence, we need to locate multiple grouplets to capture sufficient activity information. However, determining how many grouplets should be located on each frame beforehand is very difficult. To solve this problem, we propose to locate grouplets one-by-one, until enough information is observed.

The one-by-one locating method is performed as follows: The first grouplet is the optimal convergence mode of mean shift, and it is denoted as $\mathcal{G}_1$. Then, the second grouplet is searched for in the point space excluding the points of $\mathcal{G}_1$, i.e., $\mathcal{G}_2 \subseteq \mathcal{P}_t - \mathcal{G}_1$. Accordingly, the k-th grouplet is searched for in the point space excluding $\{\mathcal{G}_1, \ldots, \mathcal{G}_{k-1}\}$, and $\mathcal{G}_k \subseteq \mathcal{P}_t - \bigcup_{i=1}^{k-1} \mathcal{G}_i$.

Intuitively, the frame having more points should have more grouplets. Under this assumption, we associate the point quantity with the grouplet number by defining an information retaining rate ρ. It indicates that, during the one-by-one locating, if current grouplets retain a certain percentage of total points on frame t, then locating ends. Corresponding critical value of k can be computed by:

$$\left\{ \left|\bigcup_{i=1}^{k-1} \mathcal{G}_i\right| < |\mathcal{P}_t| \cdot \rho \right\} \wedge \left\{ \left|\bigcup_{j=1}^{k} \mathcal{G}_j\right| \geq |\mathcal{P}_t| \cdot \rho \right\} \qquad (5)$$

Compared with search for multiple grouplets jointly, this one-by-one method has a benefit that it is not necessary to set a unified cluster number. It allows different frames to locate different numbers of grouplets adaptively. During the locating, it prefers the optimum grouplet with the densest points in each iteration, thereby leaving sparse noisy points behind.

Note that a grouplet $\mathcal{G}_i$ contains a group of "dense" and "similar" points. Then, we compute a quantization vector $\bar{\boldsymbol{v}}_i$, named grouplet vector, to describe $\mathcal{G}_i$:

$$\bar{\boldsymbol{v}}_i = \frac{\sum_{\boldsymbol{p}_j \in \mathcal{G}_i} \boldsymbol{v}_j}{|\mathcal{G}_i|} \qquad (6)$$

After this quantization, $\bar{\boldsymbol{v}}_i$ captures the broad and intrinsic intensity variation in $\mathcal{G}_i$, thereby reducing the effect of existing heterogeneous points, which is related to the concept of isomorphism explained in [51]. Assuming that k grouplets are located on frame t, the grouplet vectors of frame t are denoted as $\{\bar{\boldsymbol{v}}(t)_i\}_{i=1}^{k}$.

In Fig. 5, multiple grouplets of "answer phone" and "drink water" are plotted to give an intuitive view. Evidently, different activity categories have significant differences in both grouplet appearance and grouplet number (e.g., between (a) and (e)). It is worth noting that there is slight diversity when an activity is performed twice by one person (e.g., between (e) and (f)), but this diversity is greatly enlarged when the activity is performed by different actors (e.g., between (e) and (g)). Therefore, further pattern quantization is necessary to alleviate these intra-class diversities. On the other hand, using feature quantization results in information loss. To avoid losing temporal contexts, which are highly informative for describing long-duration activities, we utilize a time-critical quantization model – RSOM.

3. Mapping grouplets to recurrent self-organizing map

This section introduces the basic algorithm of RSOM, and proposes to use a recursive adaption of neighborhood size to improve original RSOM. Then, we describe how to map newly located grouplets to the best matching units (*bmus*) on the map, and generate a growing RSOM trajectory for representation. Finally, we introduce the computation method of DTW-E distance.

3.1. RSOM learning

RSOM constitutes a direct temporal extension of SOM, aiming to map data from an input space $\boldsymbol{v}_I$ onto a lower dimensional space $\boldsymbol{v}_L$. SOM network is a two-dimensional rectangular or hexagonal grid of units. In this way, topological relationships in $\boldsymbol{v}_I$ are preserved, and SOM units are approximated to the probability density function of $\boldsymbol{v}_I$. Each unit i in SOM is associated with a weight vector $\boldsymbol{w}_i \in \Re^n$ with the same dimension as an input feature vector.[1] It is noteworthy that the initialization of unit weight uses random grouplet vectors from the training set. The learning process that leads to self-organization on the map can be summarized as following:

(i) When $\boldsymbol{v}(t)$ is an input vector, its best matching unit (*bmu*) on the map can be found by computing the minimum distance as:

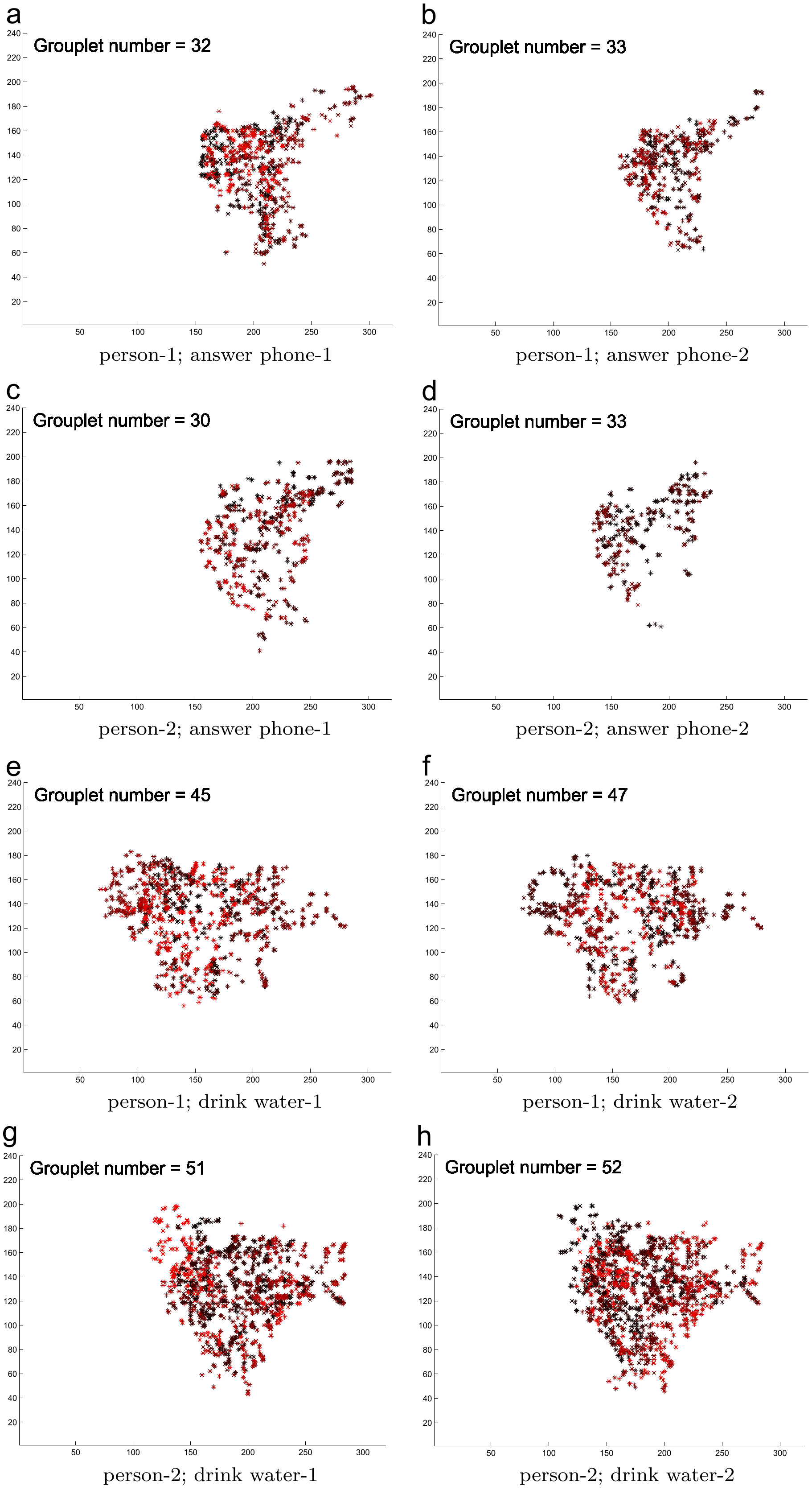
$$bmu = \arg\min_{i \in \boldsymbol{v}_L} \{\| \boldsymbol{v}(t) - \boldsymbol{w}_i(t) \|\} \qquad (7)$$

(ii) The winner *bmu* and its neighbors on the map have their weights $\boldsymbol{w}_i(t)$ updated towards $\boldsymbol{v}(t)$ as:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \alpha(t) \cdot N_{bmu,i} \cdot [\boldsymbol{v}(t) - \boldsymbol{w}_i(t)] \qquad (8)$$

where $\|\cdot\|$ is the Euclidean norm, $\alpha(t) = \alpha_i \cdot (\alpha_f/\alpha_i)^{T(i)/T_{max}} \in [0,1]$ is the learning rate, and α_i and α_f denote the initial rate and the final rate. $T(i) = 1, 2, \ldots, T_{max}$, and T_{max} is the iteration number. $N_{bmu,i}$ is the called neighborhood function and it is defined over

[1] Note that feature vector is a grouplet vector $\bar{\boldsymbol{v}}(t)_j$ where t is the frame index and j is the grouplet index on frame t. In following texts, $\bar{\boldsymbol{v}}(t)_j$ is denoted as $\boldsymbol{v}(t)$ for simplification. The index t in $\boldsymbol{v}(t)$ means the t-th grouplet, instead of frame t.



person-1; answer phone-1

person-1; answer phone-2

person-2; answer phone-1

person-2; answer phone-2

person-1; drink water-1

person-1; drink water-2

person-2; drink water-1

person-2; drink water-2

Fig. 5. The global view of overlaid STIP grouplets. Before–after relationships of grouplets can be figured out from gradient coloring from black to red. Deeper red denotes later points on the time axis. Point samples are extracted from Rochester Activities dataset [20]. (a) person-1; answer phone-1. (b) person-1; answer phone-2. (c) person-2; answer phone-1. (d) person-2; answer phone-2. (e) person-1; drink water-1. (f) person-1; drink water-2. (g) person-2; drink water-1. (h) person-2; drink water-2. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

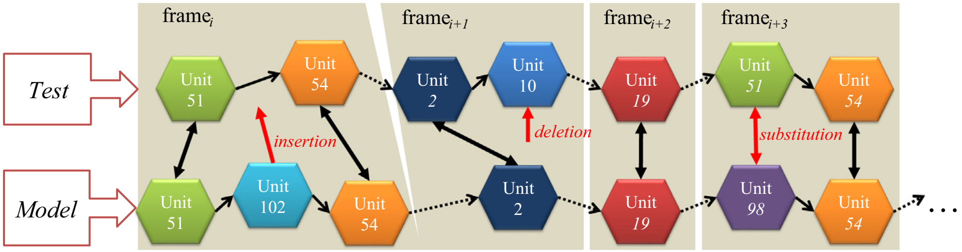


Fig. 6. Insertion, deletion and substitution for computing edit distance. Black double sided arrow indicates a successful match, and red arrow is an edit operation. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

the map units. Typically, $N_{bmu,i} = \exp\{-\|r_{bmu} - r_i\|^2/2\theta^2\}$, where $r_{bmu} \in \Re^2$ and $r_i \in \Re^2$ are the location vectors of *bmu* and *i* respectively on the map, and θ is the fixed Gaussian kernel width.

Since SOM is not originally designed to accommodate time series, its temporal extension RSOM is adopted here to learn the temporal context in the sequence of grouplet vectors. Time-critical characteristic of RSOM lies in that it utilizes both feature vectors before $\boldsymbol{v}(t)$ and $\boldsymbol{v}(t)$ itself to search for the *bmu* of $\boldsymbol{v}(t)$. This is done by associating the following recursive equation with unit i to compute a difference vector $\mathbf{u}_i(t)$:

$$\boldsymbol{u}_i(t) = \lambda \cdot [\boldsymbol{v}(t) - \boldsymbol{w}_i(t)] + (1-\lambda) \cdot \boldsymbol{u}_i(t-1) \tag{9}$$

where $0 < \lambda < 1$ is a factor determining the influence of earlier difference vectors on the current $\boldsymbol{v}(t)$. When λ is close to 0, the system of Eq. (9) involves a heavy backward memory, whereas, when λ is near 1, Eq. (9) describes a weak memory. Based on Eq. (9), searching for *bmu* in RSOM is formulated as:

$$bmu = \arg\min_{i \in \boldsymbol{v}_L} \{\|\boldsymbol{u}_i(t)\|\} \tag{10}$$

Similar to SOM, the next step is to adjust the weights of *bmu* and *bmu*'s neighbors. The key parameter is the *bmu*'s neighborhood size which represents the spatial range of units sharing similar feature appearance with *bmu*. A constant parameter θ is used in the original RSOM [40,18]. The proposed grouplet is a set of STIPs, and thus, arbitrary searching by mean shift with unknown body location or image scale must result in larger inter-ambiguity than using original STIPs [18]. To relieve this problem and boost the learning efficiency, we adopt a recursive scheme for adaptively adjusting the neighborhood size as:

$$\theta(t) = \mu \cdot \theta(t-1) + (1-\mu) \cdot \theta_{max} \cdot \mathrm{G}\left(\frac{A_{bmu}}{A_{max}}\right) \tag{11}$$

where μ is an influence factor, and $\mathrm{G}(x) = \frac{x}{1+x}$ is a monotonically increasing function with the range of $[0, 1]$. Here, θ_{max} is the maximum neighborhood size, i.e., the map size. Parameter A_{bmu} computes the local neighbor error using the average distance between *bmu* and its neighbors, and A_{max} is a normalized parameter equivalent to the maximum neighbor distance. This adjustment implies that if the neighborhood variation of sample's *bmu* is big, the neighborhood size will get larger to include more units to learn this sample, which can increase the neighbor cardinality and reduce local errors. Otherwise, the local neighborhood is relatively stable, and the reduction of neighborhood size can cut down the unnecessary update.

Then, we have $\mathrm{N}_{bmu,i}(t) = \exp\{-\|r_{bmu} - r_i\|^2/2\theta(t)^2\}$, and update the weight of unit i as:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \alpha(t) \cdot N_{bmu,i}(t) \cdot \boldsymbol{u}_i(t) \tag{12}$$

3.2. Recurrent Self-Organizing Map trajectory

An $M \times M$ map is obtained after T_{max} iterations. For simplification, one-dimensional b of the original coordinate $bmu \in \Re^2$, i.e., $b = bmu(2) \times M + bmu(1) \in [1, M^2]$, is used as the location index on the 2D map. Given the grouplet vectors on a new frame t, we can search for their *bmu* s by Eqs. (9) and (10) one-by-one. Then, the *bmu*s on frame t compose the current inner-frame trajectory $\boldsymbol{b}$ in a spatial occurrence order as Eq. (13). Inner-frame trajectories are concatenated to be an inter-frame trajectory $\boldsymbol{Trj}$ as Eq. (14)

$$\begin{cases} \boldsymbol{b}_t = [b_1, b_2, \ldots b_{k(t)}] & (13) \\ \boldsymbol{Trj} = [\boldsymbol{b}_1; \boldsymbol{b}_2; \ldots \boldsymbol{b}_t] & (14) \end{cases}$$

where t is the current frame number, and $k(t)$ is the number of grouplets on frame t. The inter-frame trajectory is also called RSOM trajectory and acts as the final representation of ongoing activity.

3.3. DTW-E distance

As introduced in Section 1.2, inter-frame trajectory has the hierarchical structure that encodes the temporal pace variation of movement sequences, while inner-frame trajectory encodes the spatial structure of body parts. Accordingly, we propose a novel measurement – DTW-E distance, where DTW distance computes the "inter-frame" warping cost and edit distance measures the "inner-frame" structural dissimilarity.

In detail, DTW finds the optimal alignment between two inter-frame trajectories if one of them is "warped" non-linearly by stretching or shrinking itself along time axis. The goal is to find the warping path for minimizing the warping cost, and in turn, the warping cost can be used as alignment distance. In this alignment process, inner-frame trajectories act as alignment units. Matching inner-frame trajectories without ignoring their inner structures is similar to matching alphabetical strings. Edit distance is thus adopted for this task. Fig. 6 shows the specific edit operations including the insertion and deletion of a single character and the substitution of a single character with another one. All operations obey the inner structures of input strings.

Algorithm 1. DTW-E distance.

```
Require: RSOM trajectories $\boldsymbol{Trj}_1 = [\boldsymbol{a}_1; \boldsymbol{a}_2; \ldots; \boldsymbol{a}_n]$,
    $\boldsymbol{Trj}_2 = [\boldsymbol{b}_1; \boldsymbol{b}_2; \ldots; \boldsymbol{b}_m]$, where $n, m$ are frame numbers and $\boldsymbol{a}_i$,
    $\boldsymbol{b}_i$ are inner-frame trajectories.
Ensure DTW-E distance $dist$ between $\boldsymbol{Trj}_1$ and $\boldsymbol{Trj}_2$
for all $i=1$ to $n$ initialize $dist(i,1) \leftarrow infinity$ end for
for all $j=1$ to $m$ initialize $dist(1,j) \leftarrow infinity$ end for
for $i=2$ to $n$ do
  for $j=2$ to $m$ do
    if $length(\boldsymbol{a}_i) = 0$ then
      $E(i,j) \leftarrow length(\boldsymbol{b}_j)$
    end if
    if $length(\boldsymbol{b}_j) = 0$ then
      $E(i,j) \leftarrow length(\boldsymbol{a}_i)$
    end if
    if $\boldsymbol{b}_j(length(\boldsymbol{b}_j)-1) = \boldsymbol{a}_i(length(\boldsymbol{a}_i)-1)$ then
      $cost_E \leftarrow 0$
```

```
        else
            $cost_E \leftarrow 1$
        end if
        $E(i,j) \leftarrow \min\{E(i-1,j)+1, E(i,j-1)+1, E(i-1,j-1)+cost_E\}$
        $dist(i,j) \leftarrow E(i,j)+\min\{dist(i-1,j), dist(i,j-1), dist(i-1,j-1)\}$
    end for
end for
return $dist$
```

The computation method of DTW-E distance is given in Algorithm 1, where $length(\cdot)$ computes the length of series, i.e., the number of grouplet vectors. To boost time efficiency, we use FastDTW [52,53] as an approximate DTW to provide the optimal or near-optimal alignment with $O(n)$ time complexity, instead of $O(n^2)$ required by DTW. In experiments, Algorithm 1 is embedded to FastDTW framework with $SearchRadius=20$.

4. Experiments and discussions

RSOM trajectory can be applied to represent a variety of human activities. Its advantage, however, is more obvious when the activity has multiple stages since the temporal contexts between activity stages can improve the discriminative ability of RSOM trajectory. This paper tests three long-duration activity datasets, namely Rochester Activities [20], UT-Interaction dataset [11], and DARPA Y1 [21]. The components of our method, grouplets, RSOM trajectory and DTW-E distance, are tested on these datasets. Comparisons with related works [17,23,24,54] are implemented on common datasets: UT-Interaction and DARPA Y1. Since activity prediction is a special application of action recognition, we additionally test our method on the newest challenging dataset called Breakfast Actions [12], and compare staged prediction accuracies with Hidden Markov Models (HMMs) [12].

4.1. Datasets and settings

Rochester activities: It contains 10 classes of daily activities: answering a phone, chopping a banana, dialing a phone, drinking water, eating a banana, eating snacks, looking up a phone number in a book, peeling a banana, eating food with silverware, and writing on a white board. Videos are performed by 5 actors of different body sizes, genders and behavioral habits, and each actor's subset consists of three-time repetitions (30 videos). Main complexities are the inter-class activity ambiguities due to many common sub-motions, e.g., eating a banana is similar to eating snacks, and turning pages in a telephone book seems to have the same hand motions with peeling a banana.

UT-interaction: This dataset has been widely tested by related methods [17,23,24,54]. It contains 6 interaction classes: “hug”, “kick”, “point”, “punch”, “push” and “shake hands”. Except that “point” is a single-body action, other activities are respectively performed by 10 pairs of actors. Experiments are implemented on the segmented version of this dataset, for which no-motion frames at the very beginning have been cut off. Following previous works, 120 videos are divided into two groups based on the filmed locations: scene-1 includes 60 videos taken on a parking lot with slightly different zoom rates and camera jitter; the other 60 videos, named scene-2, are taken on a lawn in a windy day with cluttered backgrounds: tree moves, passerby and more camera jitters.

DARPA Y1: It is a subset of videos from the Year-1 corpus of the DARPA Mind's Eye program [21]. In DARPA Y1, each video contains one of the following 7 activities: “fall”, “haul”, “hit”, “jump”, “kick”, “push” and “turn”. Following [23], we collect 20 videos for each activity class. This dataset shows much more complexity than UT-Interaction in that (1) actor size in the same activity class varies significantly in different videos; (2) activities are recorded from different camera views; (3) activity pace varies from one video to another; (4) the overhead time for an activity varies in different videos; and (5) backgrounds are complex due to non-uniform illuminations.

Breakfast actions: This newly proposed dataset is to-date one of the largest fully annotated datasets. It has 10 activities of breakfast preparation such as making coffee, orange juice, chocolate milk, tea, bowl of cereals, fried eggs, pancakes, fruit salad, sandwich, and scrambled eggs. Videos are performed by 52 different individuals in 18 different kitchen environments. The performance on this dataset demonstrates the potential of our method for handling videos recorded “in the wild”.

Low-level descriptor: Following related works [17,18,23], we extract STIPs by Dollár's interest point detector [10]. Note that other STIP detectors and descriptors are also available, and more candidates, such as [55–57], can be found in Wang's survey article [58].

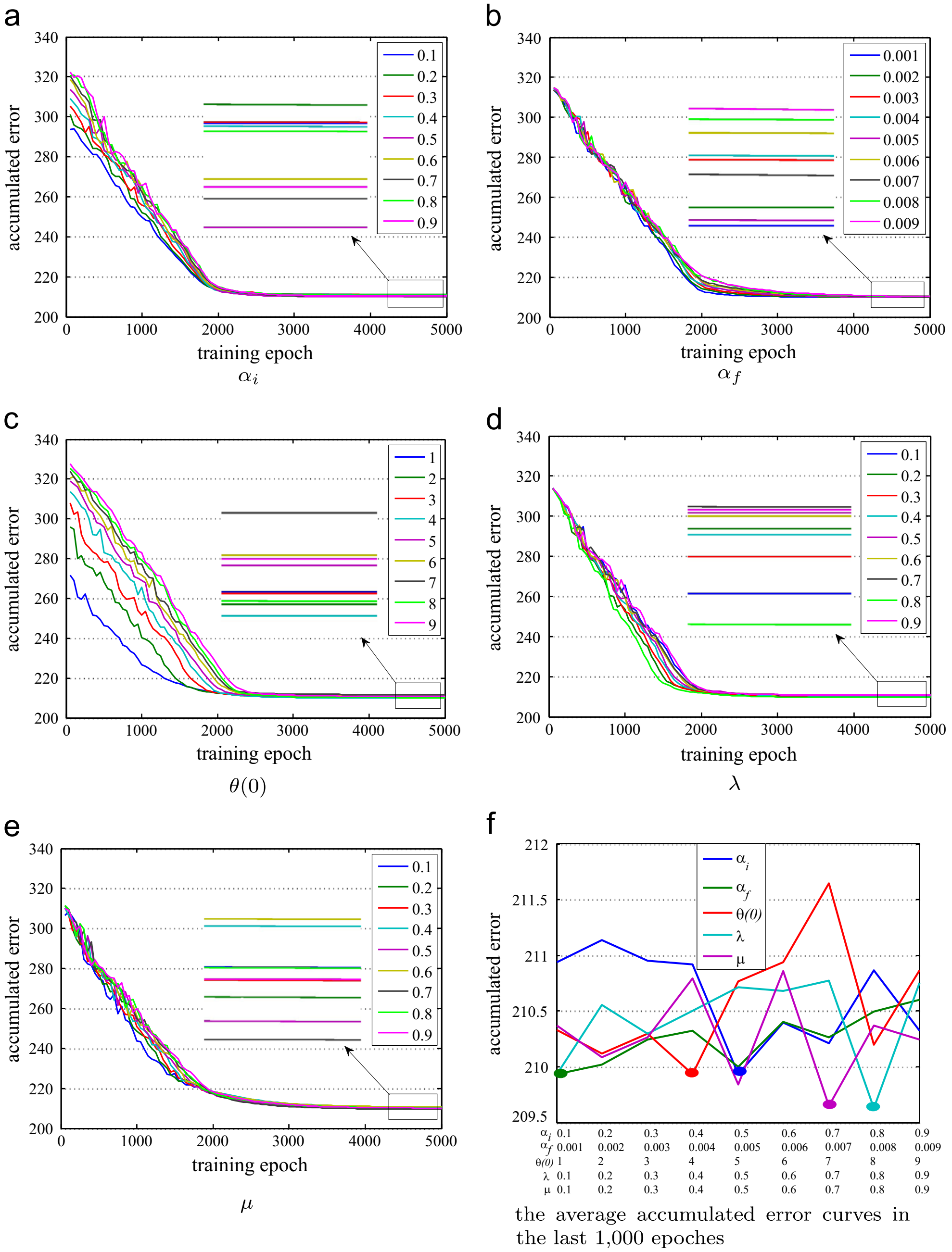
Clustering setting: As introduced in Section 2.1, a visual vocabulary is generated in prior by offline clustering, for which 100,000 features are randomly sampled from training videos. Hierarchical clustering [50] is used to ensure that similar clusters share close labels. “Gap” statistic [59] determines the cluster number as $C=450$ on Rochester Activities, $C=800$ on UT-Interaction and DARPA Y1, and $C=4000$ on Breakfast Actions.

RSOM parameter selection: We use a cyclic parameter selection method. Each time we optimize one parameter and set others as default. Firstly, we randomly select 1000 videos from 4 datasets to generate about 65,000 grouplet vectors as training samples. Involved parameters are set in reasonable ranges based on their physical meanings; initial learning rate $0.1 \le \alpha_i \le 0.9$, final learning rate $0.001 \le \alpha_f \le 0.009$, initial Gaussian kernel width $1 \le \theta(0) \le 9$, and influence factors $0.1 \le \lambda \le 0.9$, $0.1 \le \mu \le 0.9$. The first candidates in these ranges are set as default values in the first test cycle, i.e., $\alpha_i = 0.1$, $\alpha_f = 0.001, \theta(0) = 1, \lambda = 0.1$ and $\mu = 0.1$. Then, we (i) train a 10×10 RSOM for 5000 epoches with one parameter varying and others set as default, (ii) update this varying parameter to the candidate which brings the minimum average accumulated error in the last 1000 training epoches, (iii) update other parameters one-by-one in the same way, (iv) execute the next test cycle starting from the first parameter until all parameters keep unchanged.

We use the average accumulated error to measure RSOM quality. Average accumulated error curves in the last test cycle are presented in Fig. 7(a–e). Fig. 7(f) summarizes the average error curves in the last 1000 training epoches for each parameter. It is observed that the variation of amplitudes is within 209.5–212, indicating that RSOM quality is not very sensitive to these parameters. The values ($\alpha_i = 0.5$, $\alpha_f = 0.001$, $\theta(0) = 4$, $\lambda = 0.8$ and $\mu = 0.7$) associated with the minimum errors (the oval points in Fig. 7(f)) are finally selected.

Prediction protocol: Different from traditional activity recognition, the goal of prediction is to use an activity video as short as possible to make an accurate classification of its category. Following [17,23], we evaluate the prediction performance at 10 video observation ratios [10%, 20%, …, 100%], where “video observation ratio” is equivalent to “activity stage”. For example, prediction performance at observation ratio 50% describes the classification accuracy given a testing video that only has the first half of an activity.

On Rochester Activities, 120 videos taken by 4 actors are used for training, and 30 videos of the last person are used for testing in



α_i

α_f

$\theta(0)$

λ

μ

the average accumulated error curves in the last 1,000 epoches

Fig. 7. Average accumulated errors caused by different parameters. (a)–(e) present the accumulated error changes per training epoch with regards to 5 learning parameters. (f) presents the average accumulated errors in the last 1000 training epoches. In (f), colorful oval points indicate the minimum values on these curves. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 1
Recognition rate (%) and training time(s) obtained in different settings.

Method	Rochester		UT scene-1		UT scene-2		DARPA Y1	
	Rate	Time	Rate	Time	Rate	Time	Rate	Time
BoVW	71.3	15.3	76.7	7.0	68.2	8.4	32.8	3.2
Single-f BoVW+RSOM	88.4	403.4	84.5	270.9	75.4	201.6	39.5	99.0
Mean shift G+RSOM	89.8	428.7	79.3	212.6	79.1	123.2	61.7	186.3
Grouplets+RSOM (ours)	**98.4**	586.0	**100**	355.6	**97.7**	251.7	**71.3**	477.3
STIP+RSOM [18]	94.2	28,433.9	100	12,600.3	**100**	7056.7	62.5	2069.5

each round. There are 10 folds of cross-validations on UT-interaction scene-1, scene-2, and 20 folds of cross-validations on DARPA Y1, in accordance with [11,23]. Following [12], 5-fold cross-validation is used on Breakfast Actions. Prediction results are obtained by training k Nearest Neighbors (kNN) classifiers, referring to the source code of [60]. Note that all accuracies are the average results of 20 runs.

4.2. Component evaluation

4.2.1. Grouplet

To evaluate the superiority of our grouplet locator – scale-adaptive mean shift, we implement four settings: BoVW on whole videos (BoVW); single-frame BoVW (single-f BoVW); Grouplets located by a fixed-scale mean shift (mean shift G); Grouplets located by our scale-adaptive mean shift (ours). The temporal contexts of grouplet sequences and single-frame BoVW histograms are uniformly learned by RSOM. We also show the results using STIPs without any feature selection [18]. Table 1 presents the recognition results at 100% observation ratio. The "training time" refers to the average time for training kNN classifiers. Since parameters such as cluster number C vary in different settings, we show the results with optimal parameters.

As we can see from Table 1, our method outperforms others in most situations. BoVW ignores all feature relationships, thereby obtaining the lowest accuracies. However, it costs much less training time than RSOM because it uses simple Euclidean distance for measurement. Single-frame BoVW incorporating RSOM leads to performance improvements. However, this improvement is relatively low when recognizing human activity in noisy environments such as UT-Interaction scene-2 and DARPA Y1, since single-frame BoVW encodes all feature points including noises. In contrast, our method extracts dominant feature grouplets and discards scattered noises. Compared with fixed-scale mean shift, our method obtains 20.7% and 18.6% improvements in two scenes of UT-Interaction, validating the effectiveness of our method for scale-zooming videos.

On Rochester Activities, we obtain recognition rate that is 4.2% higher than [18]. The reason may be that we filter out scattered noises during grouplet locating. More importantly, grouplets are much sparser than STIPs. An intuitive comparison of their quantities can refer to Fig. 5 so that many points generate a very small number of grouplets. Point quantity may be even bigger if including the noisy points which have been filtered out. Using sparse grouplets generates very short RSOM trajectory, thereby greatly boosting the computation efficiency. For example, on Rochester Activities, the training time of our method is nearly 50 times less than [18].

In UT-Interaction scene-2, it is interesting that using grouplets achieves recognition rate that is 3.3% lower than using STIPs. It reveals the unavoidable disturbance in scenes with moving backgrounds. In practice, during the one-by-one grouplet locating, our method may be confronted with intensive noises in late phases since dense features on active body parts have been included in prior grouplets. Therefore, determining when to stop the locating becomes a problem, and it depends on the information retaining rate ρ defined in Eq. (5), where $0 \le \rho \le 1$. A bigger ρ indicates more grouplets are preserved, and it results in a lower accuracy when testing videos contain many noisy features. We select ρ based on a half-dataset validation on $\rho = 0.1, 0.2, \ldots, 1.0$. According to the results in Fig. 8(b), $\rho = 0.8$ is selected for Rochester Activities dataset which has less noises than others, and $\rho = 0.6, \rho = 0.5, \rho = 0.7$ are respectively used for UT-Interaction scene-1, scene-2 and DARPA Y1.

4.2.2. RSOM trajectory

To test the efficiency of RSOM trajectory, we implement the settings without RSOM and with fixed neighborhood RSOM (RSOM-θ). Table 2 presents corresponding recognition rates and training costs. It is evident that employing grouplets without BoVW or RSOM is very time-consuming, as both RSOM mapping and BoVW statistic involve dimensionality reduction. The third and fourth rows show that our neighborhood adaptive RSOM achieves 5.9%/2.2%/11.8% improvements over θ-fixed RSOM, without consuming too much time. It is worth noting that improvements are relatively high on Rochester Activities and DARPA Y1 which contain long-duration activities. The reason may be that longer activities usually contain a larger number of STIPs which compose more diverse grouplets. Neighborhood adaptive scheme can exploit more neighbors to learn such diversity more efficiently.

To test the stability of RSOM model for unseen videos, we use cross-dataset (c–d) validation where RSOM is pre-trained with random samples from other datasets. Corresponding results are given at the bottom rows of Table 2. Recognition accuracies on all datasets are satisfying. In particular, we observe that the performance of our neighborhood adaptive RSOM is more stable than θ-fixed RSOM, e.g., for DARPA Y1, and cross-dataset training results

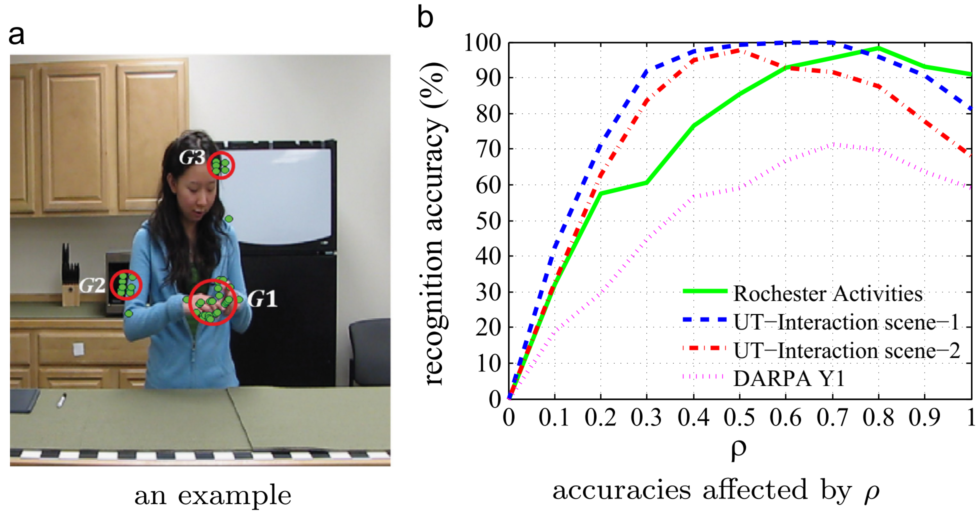


Fig. 8. The example in (a) is an "answer phone" activity. $G2$ and $G3$ seem more likely to be discarded than $G1$ when ρ is small enough. The influences of ρ on three datasets are shown in (b).

Table 2
Recognition rate (%) and training time(s) obtained with or without RSOM. "c-d" represents cross-dataset validation.

Method	Rochester		UT scene-1		UT scene-2		DARPA Y1	
	Rate	Time	Rate	Time	Rate	Time	Rate	Time
Grouplets	75.7	7825.6	83.3	3505.1	72.5	4024.5	44.3	5200.0
Grouplets + BoVW	82.0	8.5	87.6	4.5	72.1	3.3	41.5	9.2
Grouplets + RSOM-θ	92.5	599.6	100	200.2	95.5	183.3	59.5	303.8
Grouplets + RSOM (ours)	**98.4**	586.0	**100**	355.6	**97.7**	251.7	**71.3**	477.3
Grouplets + RSOM-θ (c–d)	88.3	597.0	97.6	199.9	90.3	184.8	54.0	301.6
Grouplets + RSOM (ours, c–d)	96.7	579.2	100	356.0	96.5	273.2	68.9	474.4

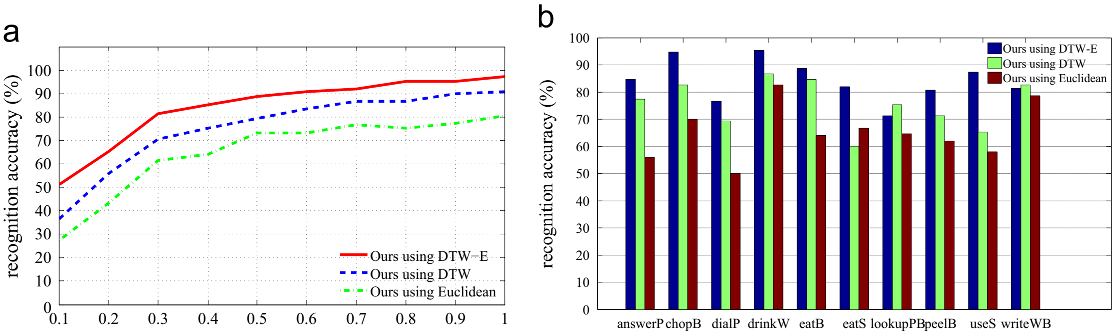


Fig. 9. Recognition rates using different RSOM trajectory measurements on Rochester Activities dataset. (a) Average rates at 10 observation ratios. (b) Average rates for 10 activities.

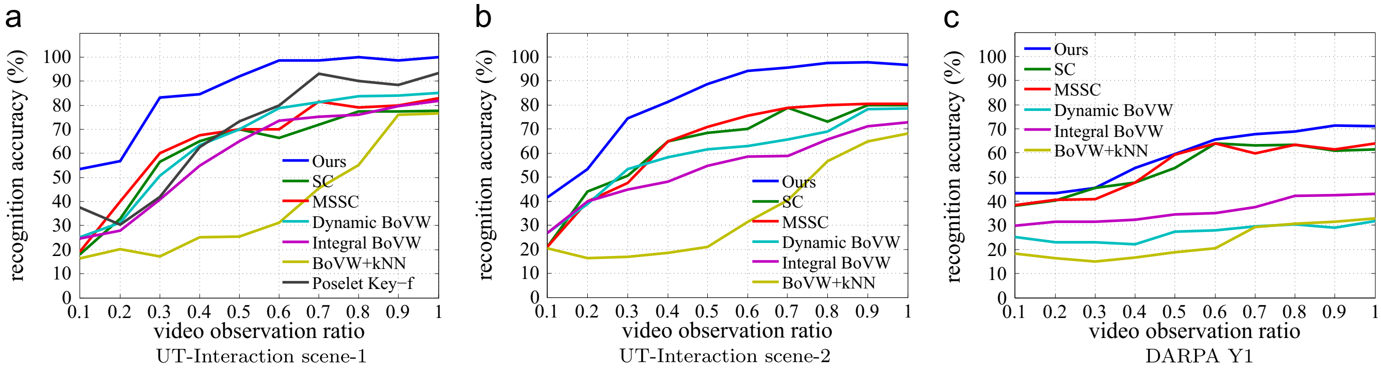


Fig. 10. Performance curves of implemented methods with respect to 10 observation ratios. (a) UT-Interaction scene-1. (b) UT-Interaction scene-2. (c) DARPA Y1.

in 5.5% accuracy reduction by "Grouplets + RSOM-θ" but only 2.4% reduction by ours.

4.2.3. *DTW-E distance*

We evaluate the performance of DTW-E distance on Rochester Activities dataset, and compare it with DTW distance and Euclidean distance. For computing DTW distance, all trajectory units are equally treated as warping units whether they are inner-frame or inter-frame. Euclidean distance measures different-length trajectories according to the length of the shorter one.

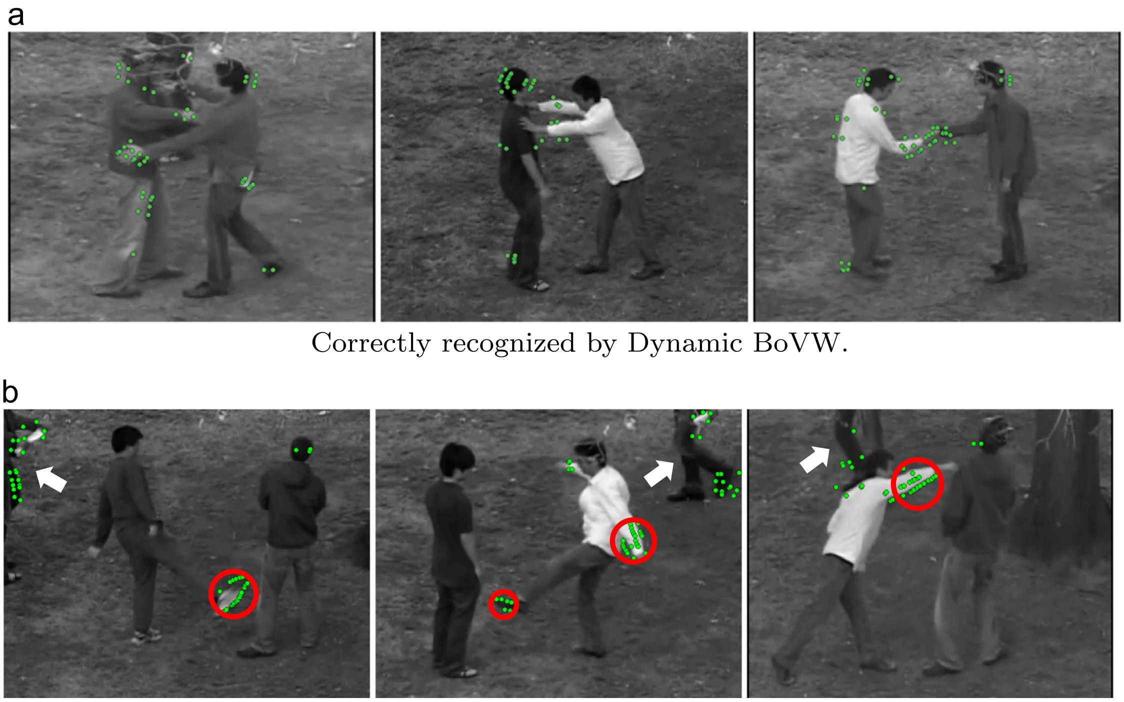
Fig. 9(a) presents the prediction accuracies at 10 observation ratios. The accuracy gaps between three curves remain approximately stable, which indicates that the improvement brought by distance selection has little relationship with observation stages. Compared with Euclidean distance, DTW achieves better performances as it greatly reduces the tempo diversity caused by different actors. The superiority of DTW-E over DTW, approximately 10%, can be attributed to the fact that DTW-E accords with the spatio-temporal structure of RSOM trajectory, i.e., it makes a difference between inter-frame temporal information and inner-frame spatial arrangement.

In Fig. 9(b), it is notable that not all activity classes are better classified using DTW-E. For "looking up a phone number in a book" and "writing on a white board", DTW distance performs better than DTW-E distance, most likely because the mismatch in edit distance is greatly enlarged when matching very short inner-frame trajectories extracted from finger motions. Taking the word editing as an example, if there is a confusion between "a" and "e", we charge a higher edit cost for revising a short word "en" to be "an" than revising a long word "waekday" to be "weekday", as we can make use of the contexts in long words to reduce edit cost.

4.3. *Comparisons with state-of-the-art methods*

In this section, we compare our method with Integral BoVW and Dynamic BoVW [17], SC and MSSC [23], Poselet Key-framing model [24], and BoVW with kNN classifiers, which can handle activity prediction. Following [17,23], UT-Interaction and DARPA Y1 are used as benchmarks. Prediction accuracies at 10 observation ratios are given in Fig. 10.

In Fig. 10(a and b), it is evident that our method outperforms others at all observation ratios. In particular, we achieve great



Correctly recognized by Dynamic BoVW.

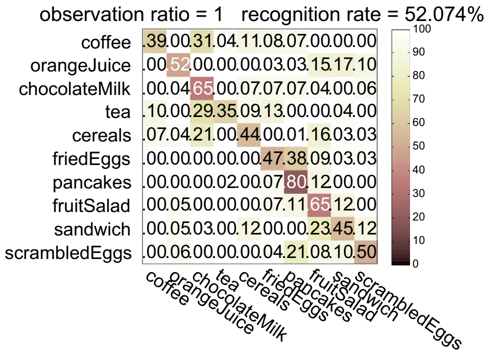
Incorrectly recognized by Dynamic BoVW, but rectified by Ours.

Fig. 11. Examples classified or misclassified by Dynamic BoVW at observation ratio $\le 50\%$. In (a), most of the feature points are extracted on the human bodies. In (b), red circles mark grouplets, and white arrows point out moving disturbances. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

improvements over the state-of-the-art BoVW model – Dynamic BoVW. For example, in UT-Interaction scene-1, we can achieve the accuracy higher than 80% after observing only the first 30% of the videos, while the Dynamic BoVW must observe more than 70% of the videos to reach the same accuracy. One reason is that when predicting 30%-observed information with 100%-observed templates, our trajectory representation commits less false than BoVW histogram whose 30% observation ratio is significantly different from the 100% template. In another respect, BoVW uses all detected points without feature selection, while our method only uses salient grouplets based on their "dense" and "similar" point appearances. Taking the snapshots in Fig. 11(b) as examples, "neatly arranged" points on actors' kicking feet and swinging arms are located as grouplets, and points on passing-by objects are relatively scattered and tend to be filtered out.

It is observed from Fig. 10(b) that Cao's SC and MSSC [23] have better performance than Dynamic BoVW when observation ratio $> 40\%$. It indicates the superiority of sparse coding methods for handling noise environments with passing-by objects. Our method achieves 53.3% accuracy at observation ratio = 20%, nearly 10% higher than SC, which demonstrates that our method has high discriminative ability at very early observation stages. On DARPA Y1 dataset (Fig. 10(c)), our performance has a stable increasing tendency after observation ratio = 60%, while both SC and MSSC fluctuate or even go down. We attribute this phenomenon to our employment of information retaining rate ρ, which enables our one-by-one locating method to select top-ranking grouplets.

Additionally, we test our method on the newest challenging dataset – Breakfast Actions [12]. The results at observation ratio = 100% are presented in the confusion table of Fig. 12. Making drinks, such as coffee, chocolate milk and tea, get mixed up with each other. Making cereals tends to be confused with making drink activities for they have many common body part movements like pouring and stirring.



observation ratio = 1 recognition rate = 52.074%

	coffee	orangeJuice	chocolateMilk	tea	cereals	friedEggs	pancakes	fruitSalad	sandwich	scrambledEggs
coffee	.39	.00	.31	.04	.11	.08	.07	.00	.00	.00
orangeJuice	.00	.52	.00	.00	.00	.03	.03	.15	.17	.10
chocolateMilk	.00	.04	.65	.00	.07	.07	.07	.04	.00	.06
tea	.10	.00	.29	.35	.09	.13	.00	.00	.04	.00
cereals	.07	.04	.21	.00	.44	.00	.01	.16	.03	.03
friedEggs	.00	.00	.00	.00	.00	.47	.38	.09	.03	.03
pancakes	.00	.00	.00	.02	.00	.07	.80	.12	.00	.00
fruitSalad	.00	.05	.00	.00	.00	.07	.11	.65	.12	.00
sandwich	.00	.05	.03	.00	.12	.00	.00	.23	.45	.12
scrambledEggs	.00	.06	.00	.00	.00	.04	.21	.08	.10	.50

Fig. 12. The confusion table of full-video recognition on Breakfast Actions.

Table 3
Prediction accuracies (%) at 4 observation stages of Breakfast Actions.

Observation ratio	25%	50%	75%	100%
HOGHOF + HMMs [12]	19.33	27.13	38.90	48.50
Ours without annotation	14.31	22.78	26.95	38.03
Ours with annotation	30.67	39.10	45.50	52.07

To compare with the generative model HMMs [12] further, we implement the prediction experiments at 4 observation ratios and present all results in Table 3. "Without annotation" implies the first part of sub-motion location is based on our automatic grouplet locating. "With annotation" indicates that experimental data have been manually annotated as movements sequences, following the Table 1 of [12]. For example, making coffee is composed of "take cup" – "pour coffee" – "pour milk" – "pour sugar" – "spoon sugar" – "stir coffee" consecutively. Note that [12] uses annotated data for training HMMs.

In Table 3, we observe that our method with annotation outperforms HMMs by a greater margin when dealing with earlier observations (e.g., 11.34% higher than HMMs using 25% of the videos) than with full observations (e.g., 3.57% higher than HMMs using 100% of the videos). The reason is that RSOM encodes the sequential contexts of body movements through thousands of training, making the RSOM trajectory highly discriminative for recognizing unfinished activities. More importantly, our method has better performances at earlier observation stages, which fits well to the concept of activity prediction.

5. Conclusions

In this paper, we propose to use the RSOM trajectory of body part movements to represent ongoing human activity. The motivation is to enable the early recognition by using a highly flexible and discriminative representation. Specifically, scale-adaptive mean shift searches for body part movements in the form of STIP grouplets. Then, the sequential contexts of grouplets are learned in RSOM through iterative training. When new videos come, STIP grouplets are located one-by-one, then mapped to RSOM to produce RSOM trajectories. For prediction, DTW distance and edit distance are combined to measure the structural dissimilarity between RSOM trajectories.

Experiments on Rochester Activities, UT-Interaction and DARPA Y1 are carefully carried out to demonstrate that our method is much more efficient than BoVW based methods. Additional tests on Breakfast Actions dataset reveal that our method outperforms generative models such as HMMs for recognizing real-world videos, especially at early observation stages.

Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC, No. 61340046), National High Technology Research and Development Program of China (863 Program, No. 2006AA04Z247), Scientific and Technical Innovation Commission of Shenzhen Municipality (Nos. JCYJ20120614152234873 and JCYJ20130331144716089), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20130001110011).

The authors wish to express special thank to Mr Andrew Shin (The University of Tokyo) for his great help on proof-reading.

References

[1] G. Guo, A. Lai, A survey on still image based human action recognition, Pattern Recognit. 47 (10) (2014) 3343–3361.

[2] D.D. Dawn, S.H. Shaikh, A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector, Vis. Comput., 2015, http://dx.doi.org/10.1007/s00371-015-1066-2.

[3] D. Oneata, J. Verbeek, C. Schmid, Action and event recognition with fisher 660 vectors on a compact feature set, in: IEEE International Conference on Computer Vision, 2013, pp. 1817–1824.

[4] H. Wang, C. Schmid, Action recognition with improved trajectories, in: IEEE International Conference on Computer Vision, 2013, pp. 3551–3558.

[5] H. Jhuang, J. Gall, M. Black, C. Schmid, Towards understanding action recognition, in: IEEE International Conference on Computer Vision, 2013, pp. 3192–3199.

[6] E. Taralova, F. de la Torre, M. Hebert, Motion words for videos, in: European Conference on on Computer Vision, Lecture Notes in Computer Science, vol. 8689, 2014, pp. 725–740.

[7] N.A. Harbi, Y. Gotoh, A unified spatio-temporal human body region tracking approach to action recognition, Neurocomputing 161 (2015) 56–64.

[8] M. Blank, L. Gorelick, E. Shechtman, M. Irani, R. Basri, Actions as space–time shapes, in: IEEE International Conference on Computer Vision, vol. 2, 2005, pp. 1395–1402.

[9] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in: IEEE International Conference on Pattern Recognition, vol. 3, 2004, pp. 32–36.

[10] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (2005) 65–72.

[11] M.S. Ryoo, J.K. Aggarwal, UT-interaction dataset, ICPR Contest on Semantic Description of Human Activities (2010).

[12] H. Kuehne, A. Arslan, T. Serre, The language of actions: recovering the syntax and semantics of goal-directed human activities, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 780–787.

[13] S. Savarese, A. DelPozo, J.C. Niebles, L. Fei-Fei, Spatial-temporal correlations for unsupervised action classification, in: IEEE Workshop on Motion and Video Computing, 2008, pp. 1–8.

[14] Q. Sun, H. Liu, Action disambiguation analysis using normalized google-like distance correlogram, in: Asian Conference on Computer Vision (ACCV 2012), Lecture Notes in Computer Science, vol. 7726, 2013, pp. 425–437.

[15] Q. Sun, H. Liu, Learning spatio-temporal co-occurrence correlograms for efficient human action classification, IEEE International Conference on Image Processing, 2013, pp. 3220–3224.

[16] M. Xin, H. Zhang, H. Wang, M. Sun, D. Yuan, ARCH: adaptive recurrent-convolutional hybrid networks for long-term action recognition, Neurocomputing, 2015, http://dx.doi.org/10.1016/j.neucom.2015.09.112.

[17] M.S. Ryoo, Human activity prediction: early recognition of ongoing activities from streaming videos, IEEE International Conference on Computer Vision, 2011, pp. 1036–1043.

[18] Q. Sun, H. Liu, Inferring ongoing human activities based on recurrent self-organizing map trajectory, in: British Machine Vision Conference, 2013, pp. 11.1–11.10.

[19] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Anal. Mach. Intell. 17 (8) (1995) 790–799.

[20] R. Messing, C. Pal, H. Kautz, Activity recognition using the velocity histories of tracked keypoints, in: IEEE International Conference on Computer Vision, 2009, pp. 104–111.

[21] Video Dataset from DARPA Mind's Eye Program, ⟨www.visint.org⟩, 2011.

[22] F. Lv, R. Nevatia, Single view human action recognition using key pose matching and viterbi path searching, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[23] Y. Cao, D. Barrett, A. Barbu, Recognizing human activities from partially observed videos, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2013, pp. 2658–2665.

[24] M. Raptis, L. Sigal, Poselet key-framing: a model for human activity recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2650–2657.

[25] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, C. Isbell, A novel sequence representation for unsupervised analysis of human activities, Artif. Intell. 173 (14) (2009) 1221–1244.

[26] K. Kitani, B.D. Ziebart, J.A.D. Bagnell, M. Hebert, Activity forecasting, in: European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 7575, 2012, pp. 201–214.

[27] H. Zhang, L.E. Parker, Bio-inspired predictive orientation decomposition of skeleton trajectories for real-time human activity prediction, in: IEEE International Conference on Robotics and Automation, 2015, pp. 3053–3060.

[28] H. Zhang, L. Parker, 4-dimensional local spatio-temporal features for human activity recognition, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011, pp. 2044–2049.

[29] M. Hoai, F. De la Torre, Max-margin early event detectors, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2863–2870.

[30] K. Li, Y. Fu, Prediction of human activity by discovering temporal sequence patterns, IEEE Trans. Pattern Anal. Mach. Intell. 36 (8) (2014) 1644–1657.

[31] B. Neumany, M. Likhachev, Planning with approximate preferences and its application to disambiguating human intentions in navigation, in: IEEE International Conference on Robotics and Automation, 2013, pp. 415–422.

[32] D. Neill, A. Moore, G. Cooper, A Bayesian spatial scan statistic, Adv. Neural Inf. Process. Syst. 18 (2006) 1003–1010.

[33] P. Haider, U. Brefeld, T. Scheffer, Supervised clustering of streaming data for email batch detection, in: ACM International Conference on Machine Learning, 2007, pp. 345–352.

[34] K.-J. Kim, Financial time series forecasting using support vector machines, Neurocomputing 5 (1) (2003) 307–319.

[35] L. D. Bourdev, J. Malik, Poselets: body part detectors trained using 3d human pose annotations, in: IEEE International Conference on Computer Vision, 2009, pp. 1365–1372.

[36] Y. Wang, D. Tran, Z. Liao, Learning hierarchical poselets for human parsing, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1705–1712.

[37] G. Sharma, F. Jurie, C. Schmid, Expanded parts model for human attribute and action recognition in still images, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 652–659.

[38] B. Leibe, B. Schiele, Scale-invariant object categorization using a scale-adaptive mean-shift search, in: Lecture Notes in Computer Science, vol. 3175, 2004, pp. 145–153.

[39] Y. Ke, R. Sukthankar, M. Hebert, Efficient temporal mean shift for activity recognition in video, Annual Conference on Neural Information Processing Systems Workshop on Activity Recognition and Discovery, 2005.

[40] M. Varsta, José del R. Millán, J. Heikkonen, A recurrent self-organizing map for temporal sequence processing, Artificial Neural Networks—ICANN'97, Springer, Berlin, Heidelberg (1997), p. 421–426.

[41] T. Kohonen, M. R. Schroeder, T. S. Huang. Self-Organizing Maps. 3rd ed., Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2001.

[42] Z. Zhang, K. Huang, T. Tan, Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes, in: International Conference on Pattern Recognition, vol. 3, 2006, pp. 1135–1138.

[43] J. Feng, C. Zhang, P. Hao, Online learning with self-organizing maps for anomaly detection in crowd scenes, in: IEEE International Conference on Pattern Recognition, 2010, pp. 3599–3602.

[44] M. Vlachos, G. Kollios, D. Gunopulos, Discovering similar multidimensional trajectories, in: IEEE International Conference on Data Engineering, 2002, pp. 673–684.

[45] F.I. Bashir, A.A. Khokhar, D. Schonfeld, Segmented trajectory based indexing and retrieval of video data, in: IEEE International Conference on Image Processing, 2003, pp. II–623.

[46] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E. Keogh, Querying and mining of time series data: experimental comparison of representations and distance measures, in: Proceedings of the VLDB Endowment, no. 2, vol. 1, 2008, pp. 1542–1552.

[47] E.J. Keogh, M.J. Pazzani, Scaling up dynamic time warping for datamining applications, ACM International Conference on Knowledge Discovery and Data Mining, 2000, pp. 285–289.

[48] V.I. Levenshtein, Binary codes capable of correcting deletions, insertions and reversals, Sov. Phys. Dokl. 10 (8) (1966) 707–710.

[49] T. Glaser, L. Zelnik-Manor, Incorporating temporal context in Bag-of-Words models, in: IEEE International Conference on Computer Vision Workshops, 2011, pp. 1562–1569.

[50] W. Zhang, D. Zhao, X. Wang, Agglomerative clustering via maximum incremental path integral, Pattern Recognit. 46 (11) (2013) 3056–3065.

[51] S. Edelman, Representation and Recognition in Vision. The MIT Press: Cambridge, Massachusetts, 1999.

[52] S. Salvador, P. Chan, Toward accurate dynamic time warping in linear time and space, Intell. Data Anal. 11 (5) (2007) 561–580.

[53] Z. Lin, Z. Jiang, L.S. Davis, Recognizing actions by shapemotion prototype trees, in: IEEE International Conference on Pattern Recognition, 2009, pp. 444–451.

[54] F. Yuan, G.S. Xia, H. Sahbi, V. Prinet, Mid-level features and spatio-temporal context for activity recognition, Pattern Recognit. 45 (12) (2012) 4182–4191.

[55] I. Laptev, M. Marszalek, C. Schmid, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[56] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: ACM International Conference on Multimedia, 2007, pp. 357–360.

[57] A. Kläser, M. Marszalek, C. Schmid, A spatio-temporal descriptor based on 3d-gradients, in: British Machine Vision Conference, 2008, pp. 995–1004.

[58] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: British Machine Vision Conference, 2009, pp. 124.1–124.11.

[59] R. Tibshirani, G. Walther, T. Hatie, Estimating the number of clusters in a data set via the gap statistic, J. R. Stat. Soc.: Ser. B (Stat. Methodol.) 63 (2) (2001) 411–423.

[60] M. Bregonzio, S. Gong, T. Xiang, Recognising action as clouds of space-time interest points, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1948–1955.

Qianru Sun received the Bachelor degree of Information and Computing Science in 2010, and is working toward the Doctor degree in the School of EE&CS, Peking University (PKU), China.

Her research interests include human action recognition & anomaly detection. She has published articles in Neurocomputing, British Machine Vision Conference (BMVC), Asian Conference on Computer Vision (ACCV), IEEE International Conference on Image Processing (ICIP) and Acoustics, Speech, and Signal Processing (ICASSP).

Hong Liu received the Ph.D. degree in Mechanical Electronics and Automation in 1996, and serves as a full professor in the School of EE&CS, Peking University (PKU), China. Liu has been selected as Chinese Innovation Leading Talent supported by "National High-level Talents Special Support Plan" since 2013.

He is also the Director of Open Lab on Human Robot Interaction, PKU, his research fields include computer vision and robotics, image processing, and pattern recognition. Liu has published more than 150 papers and gained Chinese National Aero-space Award, Wu Wenjun Award on Artificial Intelligence, Excellence Teaching Award, and Candidates of Top Ten Outstanding Professors in PKU. He is an IEEE member, vice president of Chinese Association for Artificial Intelligent (CAAI), and vice chair of Intelligent Robotics Society of CAAI. He has served as keynote speakers, co-chairs, session chairs, or PC members of many important international conferences, such as IEEE/RSJ IROS, IEEE ROBIO, IEEE SMC and IIHMSP, recently also serves as reviewers for many international journals such as Pattern Recognition, IEEE Transactions on Signal Processing, and IEEE Transactions on PAMI.

Mengyuan Liu received the Bachelor degree of Intelligence Science and Technology in 2012, and is working toward the Doctor degree in the School of EE&CS, Peking University (PKU), China.

His research interests include action recognition and localization. He has published articles in Neurocomputing, IEEE International Conference on Image Processing (ICIP), Acoustics, Speech, and Signal Processing (ICASSP) and International Conference on Robotics and Biomimetics (ROBIO).

Tianwei Zhang received his Master degree of Electronic Science and Technology in 2013, and is working toward the Doctor degree in the Department of mechanoinformatics, The University of Tokyo, Japan.

His research interests include robotic motion planning, video processing and 3-D reconstruction. He has published articles in Neurocomputing, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) and International Conference on Robotics and Biomimetics (ROBIO).