Structured Streaming Skeleton – A New Feature for Online Human Gesture Recognition

XIN ZHAO and XUE LI, University of Queensland
CHAOYI PANG, Zhejiang University and Hebei Academy of Sciences
QUAN Z. SHENG, University of Adelaide
SEN WANG, University of Queensland
MAO YE, University of Electronic Science and Technology of China

Online human gesture recognition has a wide range of applications in computer vision, especially in human-computer interaction applications. The recent introduction of cost-effective depth cameras brings a new trend of research on body-movement gesture recognition. However, there are two major challenges: (i) how to continuously detect gestures from unsegmented streams, and (ii) how to differentiate different styles of the same gesture from other types of gestures. In this article, we solve these two problems with a new effective and efficient feature extraction method—Structured Streaming Skeleton (SSS)—which uses a dynamic matching approach to construct a feature vector for each frame. Our comprehensive experiments on MSRC-12 Kinect Gesture, Huawei/3DLife-2013, and MSR-Action3D datasets have demonstrated superior performances than the state-of-the-art approaches. We also demonstrate model selection based on the proposed SSS feature, where the classifier of squared loss regression with $l_{2,1}$ norm regularization is a recommended classifier for best performance.

Categories and Subject Descriptors: I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Motion*; I.5.5 [**Pattern Recognition**]: Implementation—*Interactive systems*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Gesture recognition, feature extraction, depth camera

ACM Reference Format:
Xin Zhao, Xue Li, Chaoyi Pang, Quan Z. Sheng, Sen Wang, and Mao Ye. 2014. Structured Streaming Skeleton – A new feature for online human gesture recognition. ACM Trans. Multimedia Comput. Commun. Appl. 11, 1s, Article 22 (September 2014), 18 pages.
DOI: http://dx.doi.org/10.1145/2648583

1. INTRODUCTION

Human body gesture recognition has many valuable applications in computer vision, such as human-computer interaction, electronic entertainment, video surveillance, patient monitoring, nursing homes, smart homes, etc. Early work by Johansson [1975] suggests that movement of the human skeleton is sufficient for distinguishing different human gestures. The recent introduction of the cost-effective depth camera and the related motion capturing technique [Shotton et al. 2011] enable estimation of 3D joint positions of the human skeleton, which can further generate body motion data. This phenomenon has brought on a new trend of research on body-movement gesture

This research is partially supported by the Australian Research Council (Grant No. DP130104614) and Natural Science Foundation of China (Grant No. 61232006).

Corresponding author's addresses: X. Zhao and X. Li, Room 626, Building 78, UQ, St. Lucia, QLD 4072, Australia; email: {x.zhao, xueli}@uq.edu.au.

DOI: http://dx.doi.org/10.1145/2648583

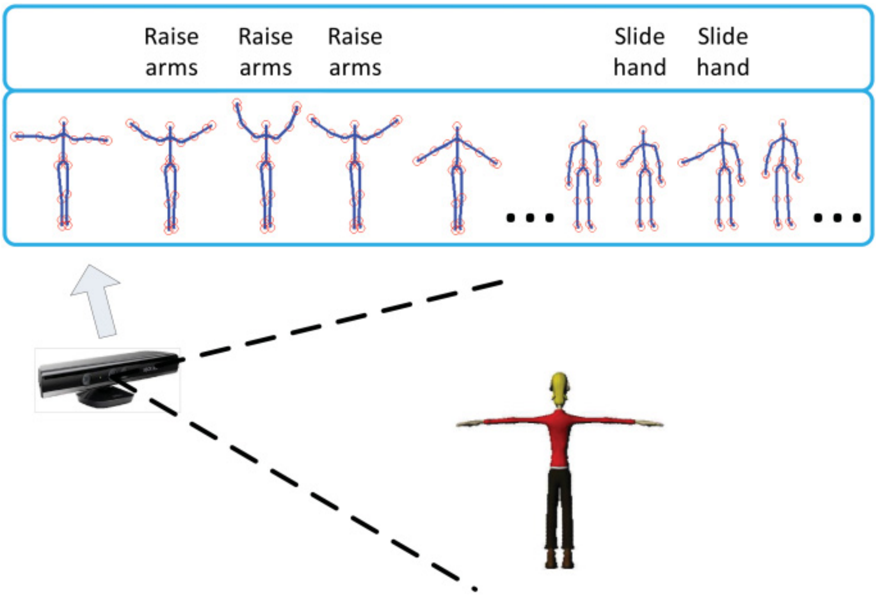


Fig. 1. Scenario of online human gesture recognition from motion datastream. A depth camera is used to capture the human skeleton datastream. The gesture labels are assigned automatically to each frame by recognition. Frames without a gesture label are considered as not belonging to any predefined gesture.

recognition [Ellis et al. 2013; Fothergill et al. 2012; Lin et al. 2012; Wang et al. 2012a; Hussein et al. 2013].

Similarities exist among three concepts: *gesture*, *action*, and *activity*. The boundaries between them are not very clear. In this article, we give our definitions based on Aggarwal and Ryoo [2011].

—*Gestures* are elementary movements of human body parts and are the atomic components used to describe meaningful motions of the human body. One example of a gesture could be “stretching arms” or “raising legs.”
—*Actions* are single-person activities that may be composed of multiple gestures organized temporally, such as “walking” and “waving.”
—*Activities* refer to interactions where two or more persons with/without objects are involved.

As atomic components, gestures are less complex than actions. Research on gesture recognition lays the foundation for action recognition. In this article, we do not consider activity recognition, because objects that could be involved in activities may not be represented by the human skeleton. A scenario of online human gesture recognition from a motion datastream is illustrated in Figure 1.

Gesture recognition needs to assign labels to gesture instances. Different gesture instances should be assigned with different labels, while the same gesture instances should be assigned with the same label. However, variations may occur to a gesture that has different appearances or different styles. We call this situation *intraclass variations*. Recently, Veeraraghavan et al. [2006] considered three sources of intraclass variations which may affect the performance of gesture recognition, namely, *viewpoint*, *anthropometry*, and *execution rate*. *Viewpoint variation* describes the relationship between the human body and the viewpoint of a camera. *Anthropometry variation* is related to the differences between human body sizes and is about human physical attributes and does not change with human movements. *Execution rate variation* is related to temporal variability caused by the speed of human movement or by different camera frame-rates. Besides these three variations, we also advocate that the *personal style of gestures* is the fourth notable intraclass variation which should also be considered, since different people may perform the same gesture differently.

There are two major challenges in dealing with intraclass variations. The first challenge is how to continuously recognize gestures from unsegmented streams. Currently, most methods choose to segment gesture instances from streaming data before the recognition of gestures [Fothergill et al. 2012; Gong et al. 2012; Wang et al. 2012a; Hussein et al. 2013]. Unfortunately, those methods suffer from the decision on the size of the segment when dealing with streaming data. By using either fixed-size or dynamic-size segments, the segmentation process itself on the streaming data introduces a new avenue of errors due to execution rate variation.

The second challenge is the ability to differentiate intraclass variations from interclass variations, because we need to decide whether those differences of gestures are within the same class or are between different classes. Misclassification errors may occur if we do not ignore the differences of intraclass variations or not discern the differences between classes. Online gesture recognition from motion datastream can be regarded as a problem of subsequence matching with multidimensional time series, where each dimension represents a specific human-body-part movement. Müller et al. [2009] and Sakurai et al. [2007] proposed approaches for segmenting motion datastream and recognizing gestures by comparing the stream data with some prelearned motion templates. Templates in their approaches are at a gesture level. A *template* is a generic gesture instance used to match the datastream for a class of gestures. These gesture-level motion template approaches have a major weakness in dealing with intraclass variations. Since the same gesture may have different instances because of intraclass variation, the problem of their approaches is twofold: first, intraclass variations cannot be differentiated if one gesture class is represented by only one single motion template. Second, if every variation is to be represented by a different motion template, there must be a large number of motion templates for a single gesture class. In practice, this is inefficient for dealing with real-time datastreams.

A gesture may involve multiple human body parts, so a gesture is regarded as a combination of movements of human body parts. A *movement* is regarded as an elementary motion of one part of the human body, so the granularity of a motion template should be fine-tuned to be at a human-body-part movement level in order to reduce redundant representation in the template modulation process. Once motion templates are represented at a human-body-part movement level, different gestures therefore can be represented by different combinations of motion templates, so the motion templates with fine-tuned granularity can improve the efficiency of the gesture recognition.

Based on these preceding discussions, we consider having a novel representation for extracting features of the human skeletons at a human-body-part movement level. This feature should also be able to represent inherent human motion characteristics such that the explicit prior segmentation process would be avoided. We call this new feature the *Structured Streaming Skeleton* (SSS), where the structure of streaming skeletons is represented by a combination of human-body-part movements, so an SSS can be denoted by a vector of cardinal values of attributes that are used to describe the skeleton in a frame. Each attribute in SSS is defined as a similarity distance between the current skeleton stream and a prelearned movement.

The two challenges involving the four intraclass variation problems previously mentioned can then be dealt with by using the proposed new SSS feature as follows.

—*Viewpoint and Anthropometry Variations.* Motion data is generated as normalized pairwise distances of human body joints. Pairwise joints are regarded as one part of the human body in this article. Then the distances are normalized by the human body size. Therefore, the SSS feature is viewpoint invariant and anthropometry invariant.

—*Execution Rate Variation.* The execution rate variation problem is solved by using SSS features, because at each frame, each attribute is defined as the distance between the best subsequence ending at the current frame and a movement. The best subsequence is the one which is most similar to this movement among all in the subsequences ending at the current frame. Different from prior segment approaches [Fothergill et al. 2012; Gong et al. 2012], the size of the segment can be optimized automatically during feature extraction. Therefore, the SSS feature is execution rate invariant.

—*Personal Style Variation.* To deal with this problem, we use *motion templates* at a granularity of human-body-part movements level. Each motion template is constructed by a human-body-part movement. Different from the approaches treating the motion template at a gesture level [Müller et al. 2009; Sakurai et al. 2007], we treat a template as a single-dimension human-body-part movement. Therefore, a gesture consists of multiple single-dimensional templates. One advantage is that different personal styles of gestures can be represented by different combinations of human-body-part movements. Therefore, the SSS feature can be used to achieve personal style invariance.

The rest of this article is organized as follows. Section 2 gives an overview of related work. Section 3 describes our approach in detail. Section 4 describes the experiments and evaluations. Finally, the conclusion is given in Section 5.

2. RELATED WORK

Recognition of human gestures, actions, and activities has been extensively surveyed in recent publications [Aggarwal and Ryoo 2011; Chaquet et al. 2013; Poppe 2010; Turaga et al. 2008]. Most existing approaches [Guha and Ward 2012; Li and Greenspan 2011; Wang et al. 2012b; Yang et al. 2013b; Zhang and Tao 2012] are about gesture and/or action recognition from color videos based on visual features, rather than the features of motion data that describe human-body-part movements. Because human-body-part movements can lead to better recognition of human gestures as well as actions, some researchers have developed approaches to first detect human-body-part locations then recognize human gestures and/or actions later. In most cases, their considerations are recognizing hand gestures online [Alon et al. 2009; Song et al. 2012] or recognizing pre-segmented gesture instances offline [Tran et al. 2012; Gupta et al. 2013].

In researching of online gesture and action recognition from motion datastreams, Fothergill et al. [2012] adopted a fixed-size sliding window to extract features and used random forest classifiers to achieve online gesture recognition. Unfortunately, their approaches cannot handle execution rate variation and incorrect segmentation problems properly. In addition to fixed-size sliding window techniques, some researchers work on action segmentation from streaming data then recognize the segmented action instances. Zhou et al. [2008] proposed a clustering algorithm for cutting the stream into action instances. Similarly, Gong et al. [2012] proposed an alignment algorithm for action segmentation. However, their approaches are all based on structure similarity between frames and are only suitable for segmenting cyclic actions. Since the structural similarity between frames of noncyclic gestures are not always obvious, incorrect segmentation errors may occur. Incorrect segmentation will consequently introduce errors in the classification process. Schwarz et al. [2010] generated the manifold embedding from joint positions of one frame into actions. However, the motion information is not fully considered, which may limit the approach for scaling to more complex actions.

Recently, Wang et al. [2012a] proposed learning one subset of human body joints for each action class. The subset joints are representative of one action compared to others. Additionally, they claimed that the relative positions of joints could result in more discriminative features. Hussein et al. [2013] used the covariance matrix for skeleton joint locations over time as a discriminative descriptor for gesture recognition. Multiple covariance matrices over subsequences in a hierarchical fashion are deployed to encode the relationship between joint movement and time. However, these two approaches are only applicable to the recognition of pre-segmented instances and cannot be used in online recognition of unsegmented datastreams.

Template-based methods treat gesture and action recognition as a database query problem which matches data with templates in the database. Veeraraghavan et al. [2006] learned an average sequence and related the function space of *Dynamic Time Warping* (DTW) [Berndt and Clifford 1994] to represent each class of action. Müller et al. [2009] presented a procedure where the unknown motion data is segmented and recognized by locally comparing it with available templates. The motion templates simply keep the patterns of actions in the same class, with the variations ignored. Ellis et al. [2013] explored the trade-off between action recognition accuracy and latency. They determined key frames from the motion data sequence to derive action templates. Sakurai et al. [2007] proposed an efficient approach for monitoring streams and for detecting subsequences that are similar to a given template sequence. However, in these approaches, one action class is represented by only one template, which is insufficient for dealing with intraclass variations.

With respect to feature extraction, there are three different ways to obtain features.

—*Sliding Window Feature Extraction.* A feature is derived from a fixed-size sliding window [Fothergill et al. 2012]. This feature can be used for online gesture recognition, but as we have explained in the introduction, this feature is not general enough to handle intraclass variations. We will show the superior performance of our proposed SSS feature over feature extractions that are based on a fixed-size sliding window approach.

—*Multi-Window Feature Extraction.* A feature is derived from multiple windows [Wang et al. 2012a; Hussein et al. 2013], but multiple-window features are mainly used to recognize presegmented gesture instances offline and cannot be directly used for online gesture recognition. We think the reason being that sliding multiple windows in a stream increases the number of samples and causes conflicts between samples.

—*Template-Based Feature Extraction.* A feature is derived via template matching between predefined templates and the streaming motion data. However, current template-based methods [Veeraraghavan et al. 2006; Müller et al. 2009] cannot transfer each frame into a same-sized vector for classification with machine learning methods. Our proposed SSS feature extraction is also a template-based method, but SSS can transfer each frame into a same-sized vector for classification with machine learning methods. This is the problem we solve in this article.

Li and Perona [2005] proposed a *Bag of Visual Words* (BoVW) model, which is used by many researchers for action recognition from color videos such as [Ryoo 2011; Wang et al. 2012b]. In this article our proposed SSS feature extraction method is different from their (BoVW) model. Our SSS feature extraction method focuses on online gesture recognition from the motion datastream. Each attribute of the SSS feature vector has specifically defined similarity to a movement and is particularly well-suited for analyzing timeseries. The BoVW model uses histograms as features for recognition of gestures. In order to count histograms, frames must be segmented first—this condition

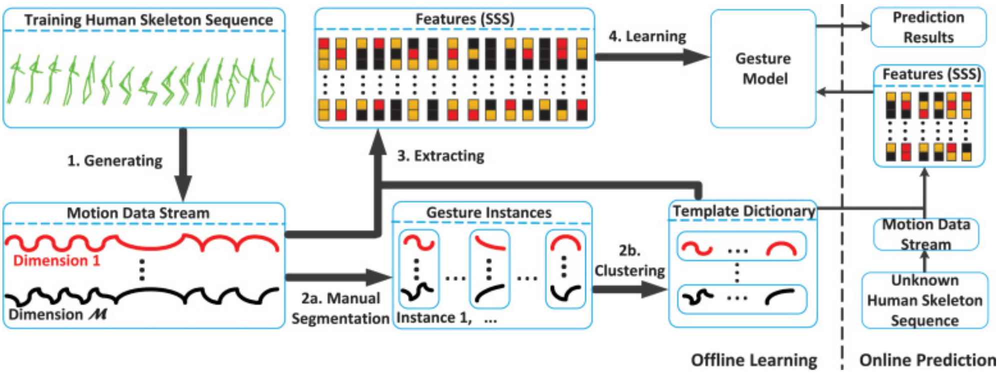


Fig. 2. Framework of our approach in both the learning stage and prediction stage.

precludes the makes BoVW model from handling online gesture recognition well. More detailed discussions on the advantages of our SSS feature extraction method are given in the next section.

3. PROPOSED APPROACH

Figure 2 shows the framework of our approach, which consists of two main stages: the *learning* and *prediction* stages. The goal of the learning stage is to construct a dictionary of templates and a gesture model. In the prediction stage, motion datastreams with unknown gestures are assigned with labels to each frame with the help of a prelearned template dictionary and the gesture model. Basically, the learning stage is offline and the prediction stage is online. We briefly describe these two stages as follows.

At the learning stage, there are four steps.

(1) *Motion Data Generation.* A training dataset captured by a depth camera consists of 3D joint positions of human skeletons. The training dataset has all gestures manually labeled on all frames. The training dataset is then scanned once. The output of this scanning is the motion datastream. The motion datastream is expressed as sequences of normalized numeric distance values of pairwise joints, which are viewpoint and anthropometry invariant. The motion datastream can be regarded as multidimensional time series. Each dimension represents a pair of specific human body joints. The dimensionality of the motion data is determined by the number of joints that motion-capture software of the depth camera can detect.

(2) *Template Dictionary Learning.* This step creates a dictionary of templates as a database of subsequences. We manually segment the training stream into gesture instances. Then we apply a clustering algorithm to group gesture instances into a dictionary of motion templates represented as a set of subsequences. Here a template is defined as a one-dimensional time series representing distance values of two joints of human body during the time of a gesture instance. For example, in Figure 3, the motion data sequence is one instance of the "slide hand" gesture. The normalized distance sequence between the joints of two hands can be a single template. As a consequence of clustering, all templates are elementary in the dictionary. We cluster each dimension of instances separately because they represent movements of different human body parts. For ordinary human gestures, different human body parts have different movements. For example, the movement patterns of two feet may not be matched with the movements of two hands, so we cluster the movements of each human body part separately, the centroids of a small number of clusters are enough to approximately represent all types of movements of this

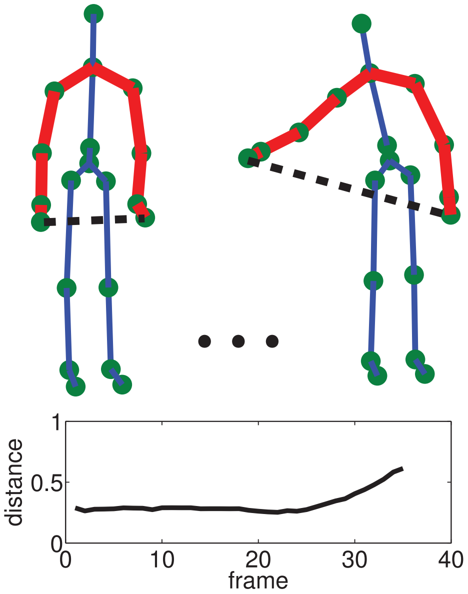


Fig. 3. Example of normalized distances. The pairwise joints are “left hand” and “right hand.” The upper part illustrates the skeleton sequence of one gesture instance. The lower part shows the normalized distances from a time series.

human body part. If we cluster movements of all human body parts together, like the BoVW model, the required number of clusters would be very large, which decreases the efficiency of online gesture recognition, so our method has advantages over the BoVM model. In our method, a gesture will be represented by a combination of a number of templates in the dictionary. This would increase the possibility using these elementary templates to compose a large number of gestures and reduce redundancy for storage, therefore improving the efficiency of online processing.

(3) *SSS Feature Extraction*. Based on the first two steps, a dictionary of templates is created. Now the training dataset will be scanned again for SSS feature extraction. The purpose of this step is to convert each frame into one SSS feature vector. Semantically, an SSS feature vector encodes the motion information in so-far scanned frames for the current frame. Here, motion information is represented as pre-learned templates. Each SSS feature vector consists of a number of attributes represented as distance values. Each value is a minimum DTW distance between all the scanned subsequences (ending at the current frame) and a template in the dictionary for the given pair of joints. It should be pointed out that a template can only be applied to the dimension it belongs to. For example, if one template is about the joints of two hands, this template could be used to match frame sequences only on the dimension about the joints of two hands. Dimensionality of an SSS feature vector is determined by the number of templates in the dictionary. We use Figure 4 to illustrate details of SSS feature extraction that also appeared in Figure 2.

As indicated by Papapetrou et al. [2011], if two sequences are similar to each other, their distances to template sequence are likely to be closer to each other. Similarly, if two sequences are not similar to each other, their distances to template sequence are likely to be farther from each other. Therefore, a distance value in our SSS feature is more meaningful and discriminative than the histogram feature in the BoVW model. SSS is specifically well-suited to the analysis of time series.

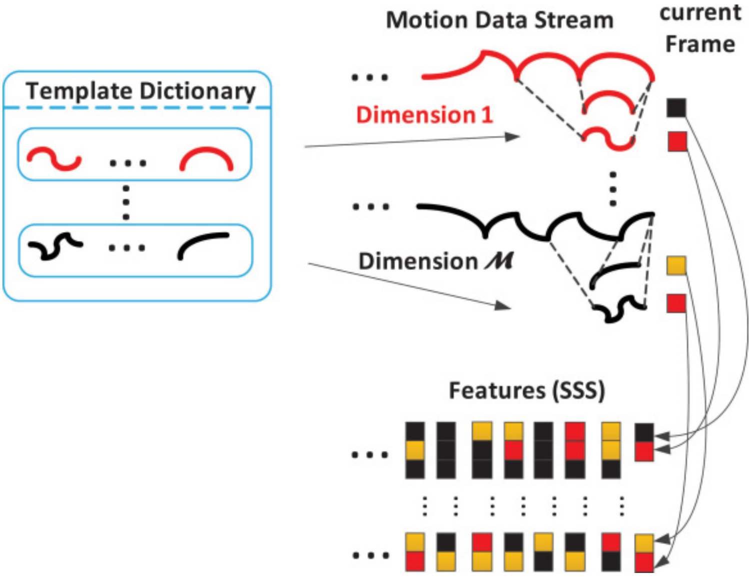


Fig. 4. Example of SSS feature extraction.

At the end of this step the labels of gestures originally assigned to the training data frames by human become SSS feature vectors assigned to each frame and ready to be used to learn the gesture model.

(4) *Model Selection for Gesture Recognition*. In this step, SSS features are fed into a classifier to learn a gesture model for recognition. In general, there are many different classifiers for different classification problems. The choice of classifier will affect the recognition performance [Bartlett et al. 2002]. Here, the questions are which classifier is suitable for SSS feature in online gesture recognition, and how to measure the performance of online gesture recognition? To answer these questions, we conduct a model selection process for the performance evaluation of the proposed SSS features.

Because online recognition requires low latency for processing streaming data, nonlinear classifiers are not suitable for real-time processing, so we only consider linear classifiers for model selection. A linear classifier can be represented as a combination of a regression term and a regularization term. The regression term is for learning the mapping from features to labels. The regularization term is for controlling the complexity of classifier to avoid over-fitting. For the regression term, Hinge loss and Squared loss are classic ones for classification. For the regularization term, l_2 norm is a classic one for binary classification. Multiclass classification can be tackled via one-vs.-all strategy, where the regularization term is a squared Frobenius norm. $l_{2,1}$ norm is relatively new and has an advantage of feature selection, which can reduce the total number of templates to improve the efficiency of SSS feature extraction. Therefore, in this article, we considered three combinations of these elements: Hinge loss plus l_2 norm (Hinge+L2) which equals to linear support vector machine [Cortes and Vapnik 1995], Squared loss plus l_2 norm (Squared+L2) which is regularized least square classification [Rifkin et al. 2003], and Squared loss plus $l_{2,1}$ norm (Squared+L21) which can be optimized by the method in Yang et al. [2013a].

A criterion is required for evaluating the classifiers for model selection. There are two types of criteria for online gesture recognition: frame-based criterion [Fothergill et al. 2012] and instance-based criterion [Bloom et al. 2012]. In frame-based criterion, each frame is treated as a unit for calculating precision and recall. Frame-based criterion cannot reveal how the recognition actually looks like at each instance. It could be that each instance is highly fragmented in terms of the gestures recognized. Imagine every other frame is correctly recognized but the frames in between are wrong. We observed that in real-world applications, the semantics of a gesture instance can be represented by using only its first frame—the other frames are not needed. Therefore, in this article, we use more reasonable instance-based criterion, where each gesture instance is treated as an unit for calculating precision and recall. Furthermore, we use cross-validation to tune parameters.

At the prediction stage, as illustrated in Figure 2, there are three steps performed in online prediction. First, the input human skeletons captured from the depth camera are translated into motion datastream using the method described in the learning stage. Then, each frame of motion datastream is mapped into an SSS feature vector also using the method described in the learning stage. Finally, at each frame, prediction is performed by a linear regression method that assigns each feature vector with a gesture label based on the learned gesture model.

3.1. Normalized Distances of Pairwise Joints

A certain amount $\mathcal{J}$ of joint positions are estimated by the motion capturing technique from depth videos. Each joint i has three coordinates $p_i(t) = (x_i(t), y_i(t), z_i(t))$ at frame t. For each pairwise joint i and j, $1 <= i < j <= \mathcal{J}$, we calculate their normalized distances: $s_{ij} = ||p_i - p_j||_2 / path_{ij}$, where $path_{ij}$ is the path between joints i and j in the human skeleton. As shown in the upper part of Figure 3, $\mathcal{J} = 20$ joints are captured with the Microsoft Kinect system [Shotton et al. 2011], and $\mathcal{M} = \mathcal{J} \times (\mathcal{J} - 1)/2 = 190$ pairwise joints are generated. We take "left hand" and "right hand" for example. The dotted line is the Euclidean distance between them. The bold lines indicate the path of these two joints in the human skeleton. We can see that s_{ij} has no relationship with the body position, body orientation, and body size, that is, viewpoint and anthropometry invariant.

We treat the normalized distance of one pairwise joint as one dimension of motion data. Along the time axis, the distances form a one-dimensional time series, as shown in the lower part of Figure 3. Therefore, motion datastream is a multidimensional time series: $\mathbf{S}(:, t) = \{s_{ij}(t)\}$, $1 <= i < j <= \mathcal{J}$.

3.2. Template Dictionary

From training motion datastream $\mathbf{S} = [\mathbf{S}(:, 1), \ldots, \mathbf{S}(:, \mathcal{N})]$, where $\mathcal{N}$ is the number of frames in motion datastream $\mathbf{S}$, we manually segment all gesture instances and learn a template dictionary $\mathbf{D}$. One template is a one-dimension time series of one instance. For each dimension of motion data, we cluster these instances into $\mathcal{G}$ clusters. In each cluster, the gesture instance with minimum average distance to others on this dimension is chosen as one template. There are $\mathcal{G} \times \mathcal{M}$ templates in this dictionary. Therefore, template dictionary $\mathbf{D} = \{d_{ij}^{g}\}$, where i and j indicate the pairwise joints and g indicates the cluster index in this pairwise joints. A spectral clustering algorithm [Ng et al. 2002], which can be used in non-Euclidean space, is adopted for the clustering. The k-means clustering algorithm [Hartigan and Wong 1979] is not applicable to non-Euclidean space, so it is not suitable here. We also tested other clustering algorithms, such as k-medoids clustering [Kaufman and Rousseeuw 1987] and optics clustering

[Ankerst et al. 1999]. A spectral clustering algorithm is the most robust. DTW is adopted as the distance measure for a time series to eliminate execution rate variation.

3.3. New SSS Feature

For recognition, each frame in $\mathbf{S}$ is required to be represented by an SSS feature vector. We compute the distances between the best-fitting subsequences ending at this frame and template dictionary $\mathbf{D}$ as the SSS feature vector.

We use $\mathbf{S}(:, [\tilde{t} : t])$ to represent all subsequences ending at frame t. From the dictionary $\mathbf{D}$, each template d^g_{ij} is used to find one best-fitting subsequence $\mathbf{S}(:, [\hat{t}^g_{ij} : t])$ among $\mathbf{S}(:, [\tilde{t} : t])$. The distance between the template and the best-fitting subsequence on related dimension is the following minimum:

$$x^g_{ij}(t) = \left|s_{ij}\left(\left[\hat{t}^g_{ij} : t\right]\right) - d^g_{ij}\right|, \tag{1}$$

$$\hat{t}^g_{ij} = \arg\min_{\tilde{t}} \left|s_{ij}([\tilde{t} : t]) - d^g_{ij}\right|. \tag{2}$$

Here, DTW is still used as the distance measure between two sequences to eliminate execution rate variation. The stream monitoring technique [Sakurai et al. 2007] can be used to detect the optimal starting point $\hat{t}^g_{ij}$.

We further use an SSS feature vector to represent one frame. Each minimum distance x^g_{ij} according to one template d^g_{ij} is one attribute of the SSS feature vector. In this article, $\mathcal{G} \times \mathcal{M}$ templates in the dictionary $\mathbf{D}$ can generate a vector with $\mathcal{G} \times \mathcal{M}$ dimensions. We treat this vector $\mathbf{X}(:, t) = \{x^g_{ij}(t)\}$ as the SSS feature for frame t. $\mathbf{X} = [\mathbf{X}(:, 1), \ldots, \mathbf{X}(:, \mathcal{N})]$ is the SSS feature matrix for $\mathbf{S}$.

3.4. Model Selection

Three linear classifiers are independently tested, Hinge+L2, Squared+L2, and Squared+L21. All three linear classifiers can be optimized via a uniform loss function,

$$(\hat{\mathbf{W}}, \hat{\mathbf{b}}) = \mathop{\arg\min}_{\mathbf{W} \in \mathbb{R}^{(\mathcal{G}\times\mathcal{M})\times\mathcal{C}}, \mathbf{b} \in \mathbb{R}^{(\mathcal{G}\times\mathcal{M})\times 1}} ||\mathbf{W}^\top \mathbf{X} + \mathbf{b}\mathbf{e}^T_{\mathcal{N}} - \mathbf{Y}||_L + \lambda ||\mathbf{W}||_P, \tag{3}$$

where $\mathcal{C}$ is the number of gesture classes, matrix $\mathbf{Y} = [\mathbf{Y}(:, 1), \ldots, \mathbf{Y}(:, \mathcal{N})]$, $\mathbf{Y}(:, t) \in \mathbb{R}^{\mathcal{C}}$ indicates the multiple labels for frame t in stream $\mathbf{S}$. If $\mathbf{S}(:, t)$ belongs to the c^{th} gesture class, $\mathbf{Y}(c, t) = 1$, $\mathbf{Y}(e, t) = -1$, for $e \neq c$. If $\mathbf{Y}(:, t)$ does not belong to any $\mathcal{C}$ classes of gestures, $\mathbf{Y}(:, t) = -1$, matrix $\hat{\mathbf{W}}$, and vector $\hat{\mathbf{b}}$ are the gesture model. λ is the parameter for controling regularization. $\mathbf{e}_{\mathcal{N}}$ is an $\mathcal{N}$-dimensional vector of all 1s. The term $||.||_L$ is a general regression function. The regression function of the Hinge+L2 classifier is a Hinge loss function,

$$\sum_{c=1}^{\mathcal{C}} \sum_{t=1}^{\mathcal{N}} \max(0, 1 - \mathbf{Y}(c, t)(\mathbf{W}(:, c)^\top \mathbf{X}(:, t) + \mathbf{b}(c))). \tag{4}$$

The regression function of the Squared+L2 classifier and Squared+L21 classifier is a squared loss function,

$$\sum_{c=1}^{\mathcal{C}} \sum_{t=1}^{\mathcal{N}} (\mathbf{W}(:, c)^\top \mathbf{X}(:, t) + \mathbf{b}(c) - \mathbf{Y}(c, t))^2. \tag{5}$$

Table I. Notations

Symbol	Description
$\mathcal{J}$	number of joints in a human skeleton model
$\mathcal{M}$	number of joint pairs in a human skeleton model
$\mathcal{N}$	number of frames of a motion datastream
$\mathcal{G}$	number of clusters for generating template dictionary
$\mathcal{K}$	number of valid templates in the template dictionary
$\mathcal{C}$	number of predefined gestures for recognition
$\mathbf{S}$	motion datastream
$\mathbf{D}$	template dictionary
$\mathbf{X}$	SSS feature matrix

The term $||.||_P$ is a general regularization function. The regularization of the Hinge+L2 classifier and Squared+L2 classifier is a squared Frobenius norm,

$$||W||_F^2 = \sum_{d=1}^{\mathcal{G}\times\mathcal{M}} \sum_{c=1}^{\mathcal{C}} W_{i,j}^2. \tag{6}$$

The regularization of the Squared+L21 classifier is a $l_{2,1}$ norm,

$$||W||_{2,1} = \sum_{d=1}^{\mathcal{G}\times\mathcal{M}} \sqrt{\sum_{c=1}^{\mathcal{C}} W_{i,j}^2}. \tag{7}$$

With the help of $l_{2,1}$ norm regularization, many rows of $\hat{\mathbf{W}}$ are near to 0. We can prune invalid feature attributes. In this article, the weight of the m^{th} attribute in feature is measured with $||\hat{\mathbf{W}}(m,:)||_2$. We descendingly sort these weights. The first $\mathcal{K}$ attributes whose weight sum is up to 99% of the total are regarded as valid. Others are set to 0, and the related templates are noted as invalid templates. The Hinge+L2 classifier and Squared+L2 classifier cannot perform feature selection. Therefore, all templates are valid.

Hinge loss plus l_2 norm (Hinge+L2) equals linear support vector machine, where parameter λ is replaced by a slack parameter γ. It can be easily proved that $\gamma = 1/(2\lambda)$.

3.5. Prediction

In the online prediction stage, given an unknown motion datasteam, at frame t, we extract SSS feature $\mathbf{U}(:,t)$ only with the valid templates in dictionary $\mathbf{D}$. The attributes related with invalid templates are set to 0 without computation. If $max(\hat{\mathbf{W}}^\top\mathbf{U}(:,t)+\hat{\mathbf{b}}) \geq \beta$, the row index with maximum value indicates the gesture class; otherwise, this frame does not belong to any $\mathcal{C}$ classes of gestures. Here β is a parameter for leverage precision and recall.

In the online prediction stage, at each frame, the time complexity of generating motion data is $O(\mathcal{M})$, the time complexity of extracting SSS feature is $O(\mathcal{M} \times \mathcal{K} \times \mathcal{A})$, where $\mathcal{A}$ is the average length of template sequence, and the time complexity of classification is $O(\mathcal{C} \times \mathcal{K})$.

The notations used in this article are given in Table I.

4. EXPERIMENTS

We chose the MSRC-12 Kinect Gesture dataset [Fothergill et al. 2012] and the Huawei/3DLife-2013 dataset [Huawei 2013] for selecting model and evaluating our SSS feature for online gesture recognition. They are public datasets for the research of online human gesture recognition from motion datastream.

In addition, we also validated our approach in presegmented action recognition using the MSR-Action3D dataset [Li et al. 2010], which is a well-known action recognition dataset for benchmarking with relevant algorithms. However, the data has been already presegmented for evaluating action instances, so the advantages of our SSS feature on online prediction cannot be demonstrated. Moreover, we can still use this dataset for model selection and demonstrating of the advantages of our SSS feature on the lower level granularity of recognition.

4.1. Results on the MSRC-12 Kinect Gesture Dataset

The MSRC-12 Gesture dataset comprises of 594 sequences, more than 700,000 frames (approximately 6 hours and 40 minutes) collected from 30 people performing 12 classes of gestures. In total, there are 6,244 gesture instances. The ending points of all gesture instances were manually labeled. Twenty human body joints ($\mathcal{J} = 20$) are captured with the Microsoft Kinect system. The body poses are captured at a sample rate of 30Hz with an accuracy approximately two centimeters in joint positions. In this dataset, for various research methods of teaching humans on how to perform different gestures, the participants were provided with three instruction modalities and their combinations to perform gestures. The three instruction modalities are (i) text descriptions, (ii) image sequences, and (iii) video demos. There are also two combinations of the three modalities, that is, images with text and video with text. When participants are given instructions, different modalities of instructions may cause different responses.

We compare our proposed SSS feature with the sliding-window-based feature proposed in Fothergill et al. [2012]. These classifiers (Hinge+L2, Squared+L2, and Squared+L21) are used for comparison.

Following the experiment setting of Fothergill et al. [2012] , we treat the previous 34 frames and ending point as one gesture instance. Thus, the average length of templates is $\mathcal{A} = 35$ frames. A fixed window of size 20 frames is centered around each ending point. All the frames inside the window are given the same gesture label as the ending point, and other frames outside the window are regarded as negative samples. In this way, we obtain the ground truth label of each frame for evaluation. Each frame is treated as one sample for training and test.

In this article, we use instance-based criterion to measure the intra-modality generalization performance: training and testing using the same instruction modality. For each modality, there are about 10 people performing all 12 classes of gesture types. We choose the first 5 people for training, and other people for prediction. The people are ranked by their original ID given by Fothergill et al. [2012]. We observed that as the number of clusters increased, F-scores increased slowly, and the number of valid templates increased linearly. Therefore, we choose $\mathcal{G} = 20$ to balance the effectiveness as well as the efficiency. Other parameters of classifiers are tuned in the training dataset with five fold cross-validation method. For the SSS feature, the slack parameter γ in the Hinge+L2 classifier was first set as $\{10^{-5}, 10^{-4}, 10^{-3}\}$, the optimal value is stabilized at 10^{-4} by cross-validation; parameter λ in the Squared+L2 classifier was set as $\{10^3, 10^4, 10^5\}$, the optimal value is stabilized at 10^4; parameter λ in the Squared+L21 classifier was set as $\{10^1, 10^2, 10^3\}$, the optimal value is stabilized at 10^2. For the sliding window feature [Fothergill et al. 2012], the slack parameter γ in the Hinge+L2 classifier was set as $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, the optimal value is 10^{-3} or 10^{-2}; parameter λ in the Squared+L2 classifier was set as $\{10^1, 10^2, 10^3, 10^4\}$, the optimal value is is 10^2 or 10^3; parameter λ in the Squared+L21 classifier was set as $\{10^0, 10^1, 10^2, 10^3\}$, the optimal value is 10^1 or 10^2. In all experiments, parameter β was set as $\{-1, -0.9, -0.8, \ldots, 0.8, 0.9, 1\}$ for parameter tuning by cross-validation. Finally, each person has 12 F-scores for 12 gesture types. The reported F-score is an average over all the people in the prediction dataset and 12 gestures.

Table II. Comparison between SSS Feature and Sliding Window Feature on MSRC-12 Gesture Dataset Using Hinge+L2 Classifier

Modality\Feature	SSS feature	Sliding window feature [Fothergill et al. 2012]	improvement
Text	78.4	65.0	13.4
Images	73.5	60.8	12.7
Video	61.1	53.3	7.8
Images+Text	82.9	72.1	10.8
Video+Text	83.8	78.1	5.7
Average	75.9	65.8	10.1

Table III. Comparison between SSS Feature and Sliding Window Feature on MSRC-12 Gesture Dataset Using Squared+L2 Classifier

Modality\Feature	SSS feature	Sliding window feature [Fothergill et al. 2012]	improvement
Text	82.3	69.7	12.5
Images	78.4	62.4	16.0
Video	64.8	59.3	5.5
Images+Text	85.7	76.6	9.1
Video+Text	88.1	75.7	12.3
Average	79.9	68.7	11.1

Table IV. Comparison between SSS Feature and Sliding Window Feature on MSRC-12 Gesture Dataset Using Squared+L21 Classifier

Modality\Feature	SSS feature	Sliding window feature [Fothergill et al. 2012]	improvement
Text	78.4	67.2	11.2
Images	78.9	59.6	19.3
Video	63.8	57.0	6.7
Images+Text	82.7	72.1	10.5
Video+Text	88.8	71.7	17.0
Average	78.5	65.5	12.9

Tables II, III, and IV show the results of comparison between the SSS feature and the sliding window feature by using the Hinge+L2, Squared+L2, and Squared+L21 classifiers, respectively. The values in these tables are expressed as percentages of F-scores. From these tables, we can see that Squared+L2 classifier obtains better results (about 4% average higher F-scores) than that of the Hinge+L2 classifier. One explanation is that hinge loss yields a sparse solution [Bradley and Mangasarian 1998] which is not suitable for the SSS feature. We can also see that both the Squared+L2 and Squared+L21 classifiers obtain similar results (about 1% difference in average F-scores), but Squared+L21 classifier can reduce about 1/3 of the total templates for feature extraction, to improving the overall efficiency of online gesture recognition. Besides, using the SSS feature can obtain considerable improvements of F-scores varying between 5% to 19%. The average improvement is more than 10%. This demonstrates the superior performance of the SSS feature over the sliding window feature [Fothergill et al. 2012] in online gesture recognition.

We preform our experiments with i7 860 CPU and 4G RAM and of Matlab hybrid with parts of C code. With our approach, in the prediction stage, gesture recognition of one frame costs about less than 3ms with the Squared+L2 classifier, while the time cost of one frame is less than 2ms with the Squared+L21 classifier.

Table V. Comparison between SSS Feature and Sliding Window Feature on Huawei/3DLife-2013 Dataset

Classifier\Feature	SSS feature	Sliding window feature [Fothergill et al. 2012]	improvement
Hinge+L2	57.8	48.0	9.8
Squared+L2	59.7	45.1	14.6
Squared+L21	59.9	42.2	17.7

4.2. Results on Huawei/3DLife-2013 Dataset

The Huawei/3DLife-2013 dataset is specially designed for the ACM Multimedia Grand Challenge (2013). As suggested from the dataset developer, we use Session 2 in Dataset 1 for motion-capture-based gesture recognition [Huawei 2013].

This dataset is collected from two synchronized horizontal Kinects with the subject around 3m away from the two Kinects. Kinect 1 is placed in front of the subject and Kinect 2 is on the side of the subject. Because the estimated skeletons from Kinect 1 are reasonably accurate but the skeletons from Kinect 2 suffer from serious noises and missing data, we only use the data captured with Kinect 1 for experiments. In the dataset, there are 14 people: each of them continually performs 17 gestures. Each gesture is repeated around 5 times by each subject. These 17 gestures are classified into 3 classes: (i) simple gestures that involve mainly the upper human body (hand waving, knocking on the door, clapping, throwing, punching, push away with both hands); (ii) training exercises (jumping jacks, lunges, squats, punching and then kicking, weight lifting); and (iii) sports-related activities (golf drive, golf chip, golf putt, tennis forehand, tennis backhand, walking on the treadmill).

However, we do not actually regard "walking" as a gesture in this research. It is a cyclic action: one walking instance contains more than one cycles and can last more than 10 seconds. We cannot simply treat one walking instance as one gesture instance. In this article, we still consider "walking" as a gesture and set it to be a randomly selected 100 continuous frames from every 200 frames of walking instances.

Kinect 1 records a depth video for each person and saves it in a ".oni" file using the OpenNI software package. Each depth video has an annotation file. The annotation contains the name, start time, and end time of each manually-segmented gesture instance. We take the annotation as ground truth in the evaluation process.

In our approach, depth videos need to be converted into human skeleton sequences. We use OpenNI 2.1 and NiTE 2.2 software to generate the human skeleton sequence from each ".oni" file. Different from the MSRC-12 Kinect Gesture dataset, human body joints ($\mathcal{J} = 16$) were estimated with NiTE 2.2. Each skeleton sequence starts at the start time of the first annotated gesture instance and ends at the end time of the last annotated gesture instance. Since the skeleton software (OpenNI 2.1 and NiTE 2.2) can not be initialized for the data before frame 1,400 in person 7, these data are deleted from the experiments. Considering that the number of deleted gesture instances is less than 2% of the total number of gesture instances, we claim that the experiment results are not significantly affected by the data deletion.

We use the same experiment setting as the MSRC-12 Kinect Gesture dataset. Instance-based criterion is used to measure the effectiveness. We choose the first 5 people for training, and other people for prediction. The people are ranked by their original ID given by [Huawei 2013]. The number of clusters $\mathcal{G}$ is also fixed to 20. Other parameters of classifiers are tuned in the training dataset with five fold cross-validation method. Each person has 16 F-scores for 16 gesture types. The reported F-score is an average over all the people in the prediction dataset and 16 gestures.

Table V shows the results of comparisons between the SSS feature and sliding window feature by using Hinge+L2, Squared+L2, and Squared+L21, respectively. From this table, we can see that (1) Squared+L2 obtains better results than Hinge+L2;

Table VI. Comparison on MSR-Action3D Dataset

Method	Accuracy
Recurrent Neural Network [Martens and Sutskever 2011]	42.5
Dynamic Temporal Warping [Müller et al. 2009]	54.0
Hidden Markov Model [Lv and Nevatia 2006]	63.0
Multiple Instance Learning [Ellis et al. 2013]	65.7
SSS feature with Hinge+L2 classifier	72.5
SSS feature with Squared+L2 classifier	79.2
SSS feature with Squared+L21 classifier	81.7
Actionlet Ensemble [Wang et al. 2012a]	88.2

(2) for the SSS feature, the difference between the average F-scores of Squared+L2 and Squared+L21 is less than 1%; (3) the average improvement of the SSS feature is about 10% better than that of the Sliding Window feature. These results are consistent with those of the experiments in the MSRC-12 Kinect Gesture dataset.

The results of the Huawei/3DLife-2013 dataset are lower than the results of the MSRC-12 Gesture dataset because there are some very similar gestures in the 16 gestures, such as “hand waving” and “throwing”, “golf drive” and “golf chip”, and some gestures are other gestures’ subsequence, such as “punching” and “punching and then kicking”.

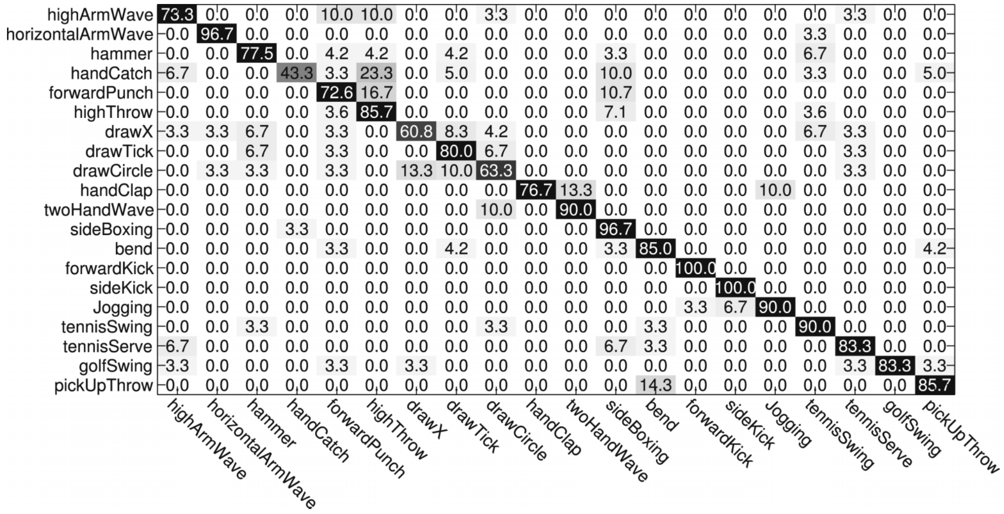
4.3. Results on MSR-Action3D Dataset

The MSR-Action3D dataset [Li et al. 2010] comprises of 557 presegmented action instances. There were 10 people performing 20 classes of gestures. Same as with the MSRC-12 Kinect Gesture dataset, human body joints ($\mathcal{J} = 20$) were captured with the Microsoft Kinect system.

Because the instances have been manually segmented, we simplify online extracting features by computing the distances between the presegmented instance and the template dictionary directly. Each instance is treated as one sample. We use Hinge+L2, Squared+L2, and Squared+L21 for experiments. The number of clusters $\mathcal{G}$ is also fixed to 20. The parameters in classifiers are optimized on the test sets. In comparing all approaches, the parameters are optimized in the same way. The fairness of the comparison is evidenced by using the same experiment setting (i.e., the method of partitioning of the training datasets and test datasets and the method of parameter tuning) on the same standard dataset.

We compare our approaches with state-of-the-art methods on the cross-subject test setting [Li et al. 2010; Wang et al. 2012a], where the samples of half the people are used as training data, and the rest are used as testing data. As Table VI shows, our approaches outperform the other time-series-based methods [Ellis et al. 2013; Lv and Nevatia 2006; Martens and Sutskever 2011; Müller et al. 2009], which treat the motion data as an undivided whole set. The only approach [Wang et al. 2012a] that outperforms ours uses a subset of joints for classification, which is similar to our approach, but it focuses on recognition at presegmented document level and cannot be used in online recognition from unsegmented streams.

The Squared+L21 classifier obtains the highest accuracy among the three tested classifiers. The confusion matrix of the SSS feature with the Squared+L21 classifier is illustrated in Figure 5. We can see that for most actions, our approach works well, while for similar actions such as “hand catch” and “high throw”, “draw X” and “draw circle”, there are some misclassifications. It can be seen that, for each action, there are about ten instances performed by five people for training, which may be insufficient to distinguish these similar gestures.



	highArmWave	horizontalArmWave	hammer	handCatch	forwardPunch	highThrow	drawX	drawTick	drawCircle	handClap	twoHandWave	sideBoxing	bend	forwardKick	sideKick	Jogging	tennisSwing	tennisServe	golfSwing	pickUpThrow
highArmWave	73.3	0.0	0.0	0.0	10.0	10.0	0.0	0.0	3.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0
horizontalArmWave	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0	0.0
hammer	0.0	0.0	77.5	0.0	4.2	4.2	0.0	4.2	0.0	0.0	0.0	3.3	0.0	0.0	0.0	0.0	6.7	0.0	0.0	0.0
handCatch	6.7	0.0	0.0	43.3	3.3	23.3	0.0	5.0	0.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0	5.0
forwardPunch	0.0	0.0	0.0	0.0	72.6	16.7	0.0	0.0	0.0	0.0	0.0	10.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
highThrow	0.0	0.0	0.0	0.0	3.6	85.7	0.0	0.0	0.0	0.0	0.0	7.1	0.0	0.0	0.0	0.0	3.6	0.0	0.0	0.0
drawX	3.3	3.3	6.7	0.0	3.3	0.0	60.8	8.3	4.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.7	3.3	0.0	0.0
drawTick	0.0	0.0	6.7	0.0	3.3	0.0	0.0	80.0	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0
drawCircle	0.0	3.3	3.3	0.0	3.3	0.0	13.3	10.0	63.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0
handClap	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	76.7	13.3	0.0	0.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0
twoHandWave	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.0	0.0	90.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
sideBoxing	0.0	0.0	0.0	3.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
bend	0.0	0.0	0.0	0.0	3.3	0.0	0.0	4.2	0.0	0.0	0.0	3.3	85.0	0.0	0.0	0.0	0.0	0.0	0.0	4.2
forwardKick	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0
sideKick	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
Jogging	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.3	6.7	90.0	0.0	0.0	0.0	0.0
tennisSwing	0.0	0.0	3.3	0.0	0.0	0.0	0.0	0.0	3.3	0.0	0.0	0.0	3.3	0.0	0.0	0.0	90.0	0.0	0.0	0.0
tennisServe	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.7	3.3	0.0	0.0	0.0	0.0	83.3	0.0	0.0
golfSwing	3.3	0.0	0.0	0.0	3.3	0.0	3.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.3	83.3	3.3
pickUpThrow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.3	0.0	0.0	0.0	0.0	0.0	0.0	85.7

Fig. 5. The confusion matrix of our proposed approach for the MSR-Action3D dataset.

5. CONCLUSIONS

Depth cameras are now widely used in applications of human-computer interaction. There is a growing need to apply depth cameras in human behavior detection, such as gesture, action, and activity recognition. The effective and efficient recognition of human gestures in a real-time fashion has a significant impact on the recognition of human actions.

In a nutshell, our contributions are as follows.

—*New SSS Feature.* We proposed a novel feature, namely, *Structured Streaming Skeleton* (SSS), for online gesture recognition from motion datastreams to deal with four types of intraclass variations (i.e., viewpoint, anthropometry, execution rate, and personal style), thereby effectively and efficiently solving the incorrect segmentation and inadequate template matching problems.

—*None Prior Segmentation for Online Recognition.* We detect the size of the segment by dynamically matching with pre-learned templates. Execution variation is eliminated and there is no avenue for errors made by prior segmentations.

—*Fine-Tuned Granularity of Motion Templates.* We create a motion template dictionary at a granularity of elementary body-part-movement level. We consider the human body as a combination of many small parts and perform body part analysis separately. One advantage is that personal styles of gestures can be represented by different combinations of human-body-part movements.

—*High Effectiveness.* Because of the discriminative nature of the SSS feature, superior performance is achieved even with simple classifiers, with an average improvement of F-scores by 10% compared with the sliding-window-based feature.

—*Model Selection.* We conducted experiments for model selection of classifiers with the SSS feature. Compared to the function of hinge loss regression, squared loss regression is more suitable for the SSS feature. In our experiments, both l_2 norm regularization and $l_{2,1}$ norm regularization have achieved almost the same effectiveness. However, with respect to efficiency, we recommend squared loss regression with $l_{2,1}$ norm regularization. Its effectiveness can achieve as high as 88% of F-score measured with instance-based criterion, and its efficiency can achieve as fast as 2ms per frame on a general desktop machine.

The proposed SSS feature can be further exploited. Our further research will consider (i) machine learning methods that can be incorporated with SSS feature extraction for online gesture recognition; (ii) online gesture recognition with inaccurate skeleton data to reduce gesture recognition errors that are caused by incomplete skeleton tracking.

REFERENCES

J. K. Aggarwal and M. S. Ryoo. 2011. Human activity analysis: A review. *ACM Comput. Surv.* 43, 3 (2011), 16.

Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff. 2009. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 9 (2009), 1685–1699.

Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: Ordering points to identify the clustering structure. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*. 49–60.

Peter L. Bartlett, Stéphane Boucheron, and Gábor Lugosi. 2002. Model selection and error estimation. *Machine Learn.* 48, 1–3 (2002), 85–113.

D. Berndt and J. Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the KDD Workshop*, Vol. 10. 359–370.

Victoria Bloom, Dimitrios Makris, and Vasileios Argyriou. 2012. G3D: A gaming action dataset and real time action recognition evaluation framework. In *Proceedings of the Computer Vision and Pattern Recognition Workshops (CVPRW)*. 7–12.

Paul S. Bradley and Olvi L. Mangasarian. 1998. Feature selection via concave minimization and support vector machines.. In *Proceedings of the International Conference on Machine Learning (ICML)*, Vol. 98. 82–90.

Jose M. Chaquet, Enrique Carmona, and Antonio Fernández-Caballero. 2013. A survey of video datasets for human action and activity recognition. *Comput. Vision Image Understand.* 117, 6 (2013), 633–659.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learn.* 20, 3 (1995), 273–297.

C. Ellis, S. Z. Masood, M. F. Tappen, J. J. LaViola, and R. Sukthankar. 2013. Exploring the trade-off between accuracy and observational latency in action recognition. *Int. J. Comput. Vision* 101, 3 (2013), 420–436.

Simon Fothergill, Helena M. Mentis, Pushmeet Kohli, and Sebastian Nowozin. 2012. Instructing people for training gestural interactive systems. In *Proceedings of the ACM Annual Conference on Human Factors in Computing Systems (CHI)*. 1737–1746.

Dian Gong, Gérard Medioni, Sikai Zhu, and Xuemei Zhao. 2012. Kernelized temporal cut for online temporal segmentation and recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 229–243.

Tanaya Guha and Rabab K. Ward. 2012. Learning sparse representations for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 8 (2012), 1576–1588.

Raj Gupta, Alex Yong-Sang Chia, and Deepu Rajan. 2013. Human activities recognition using depth images. In *Proceedings of the 21st ACM International Conference on Multimedia*. 283–292.

Huawei. 2013. Huawei/3DLife ACM Multimedia Grand Challenge 2013. http://mmv.eecs.qmul.ac.uk/mmgc2013/ (2013).

J. A. Hartigan and M. A Wong. 1979. A k-means clustering algorithm. *J. Royal Stat. Soc. C* 28 (1979), 100–108.

Mohamed E. Hussein, Marwan Torki, Mohammad A. Gowayyed, and Motaz El-Saban. 2013. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2466–2472.

G. Johansson. 1975. Visual motion perception. *Sci. Am.* 232, 6 (1975), 76–88.

Leonard Kaufman and Peter Rousseeuw. 1987. Clustering by means of medoids. In *Statistical Data Analysis Based on the L1-Norm and Related Methods.* Birkhäuser Basel, 405–416.

Fei-Fei Li and Pietro Perona. 2005. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2. 524–531.

Hong Li and Michael Greenspan. 2011. Model-based segmentation and recognition of dynamic gestures in continuous video streams. *Pattern Recogn.* 44, 8 (2011), 1614–1628.

W. Li, Z. Zhang, and Z. Liu. 2010. Action recognition based on a bag of 3d points. In *Proceedings of the CVPR Workshop*. 9–14.

Shih-Yao Lin, Chuen-Kai Shie, Shen-Chi Chen, and Yi-Ping Hung. 2012. Action recognition for human-marionette interaction. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 39–48.

F. Lv and R. Nevatia. 2006. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 359–372.

J. Martens and I. Sutskever. 2011. Learning recurrent neural networks with Hessian-free optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*. 1033–1040.

Meinard Müller, Andreas Baak, and Hans-Peter Seidel. 2009. Efficient and robust annotation of motion capture data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA)*. 17–26.

A. Y. Ng, M. I. Jordan, and Y. Weiss. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*. 849–856.

Panagiotis Papapetrou, Vassilis Athitsos, Michalis Potamias, George Kollios, and Dimitrios Gunopulos. 2011. Embedding-based subsequence matching in time-series databases. *ACM Trans. Datab. Syst.* 36, 3 (2011), 17.

Ronald Poppe. 2010. A survey on vision-based human action recognition. *Image Vision Comput.* 28, 6 (2010), 976–990.

Ryan Rifkin, Gene Yeo, and Tomaso Poggio. 2003. Regularized least-squares classification. *Nato Sci. Series Sub Series III Comput. Syst. Sci.* 190 (2003), 131–154.

M. S. Ryoo. 2011. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 1036–1043.

Yasushi Sakurai, Christos Faloutsos, and Masashi Yamamuro. 2007. Stream monitoring under the time warping distance. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*. 1046–1055.

L. A. Schwarz, D. Mateus, V. Castañeda, and N. Navab. 2010. Manifold learning for ToF-based human body tracking and activity recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*. 1–11.

Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. 2011. Real-time human pose recognition in parts from single depth images. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 1297–1304.

Yale Song, David Demirdjian, and Randall Davis. 2012. Continuous body and hand gesture recognition for natural human-computer interaction. *ACM Trans. Interactive Intell. Syst.* 2, 1 (2012), 5.

K. N. Tran, I. A. Kakadiaris, and S. K. Shah. 2012. Part-based motion descriptor image for human action recognition. *Pattern Recog.* 45, 7 (2012), 2562–2572.

Pavan Turaga, Rama Chellappa, Venkatramana S. Subrahmanian, and Octavian Udrea. 2008. Machine recognition of human activities: A survey. *IEEE Trans. Circuits Syst. Video Technol.* 18, 11 (2008), 1473–1488.

Ashok Veeraraghavan, Rama Chellappa, and Amit K. Roy-Chowdhury. 2006. The function space of an activity. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. 959–968.

Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. 2012a. Mining actionlet ensemble for action recognition with depth cameras. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 1290–1297.

Sen Wang, Yi Yang, Zhigang Ma, Xue Li, Chaoyi Pang, and Alexander G. Hauptmann. 2012b. Action recognition by exploring data distribution and feature correlation. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*. 1370–1377.

Yi Yang, Zhigang Ma, Alexander G. Hauptmann, and Nicu Sebe. 2013a. Feature selection for multimedia analysis by sharing information among multiple tasks. *IEEE Trans. Multimedia* 15, 3 (2013), 661–669.

Yang Yang, Imran Saleemi, and Mubarak Shah. 2013b. Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 7 (2013), 1635–1648.

Zhang Zhang and Dacheng Tao. 2012. Slow feature analysis for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 3 (2012), 436–450.

F. Zhou, F. Torre, and J. K. Hodgins. 2008. Aligned cluster analysis for temporal segmentation of human motion. In *Proceedings of the IEEE Conference on Automatic Face and Gesture Recognition (FG)*. 1–7.

Received January 2014; revised June 2014; accepted June 2014