Early Recognition of 3D Human Actions

SHENG LI, KANG LI, and YUN FU, Northeastern University

Action recognition is an important research problem of human motion analysis (HMA). In recent years, 3D observation-based action recognition has been receiving increasing interest in the multimedia and computer vision communities, due to the recent advent of cost-effective sensors, such as depth camera Kinect. This work takes this one step further, focusing on early recognition of ongoing 3D human actions, which is beneficial for a large variety of time-critical applications, e.g., gesture-based human machine interaction, somatosensory games, and so forth. Our goal is to infer the class label information of 3D human actions with partial observation of temporally incomplete action executions. By considering 3D action data as multivariate time series (m.t.s.) synchronized to a shared common clock (frames), we propose a stochastic process called dynamic marked point process (DMP) to model the 3D action as temporal dynamic patterns, where both timing and strength information are captured. To achieve even more early and better accuracy of recognition, we also explore the temporal dependency patterns between feature dimensions. A probabilistic suffix tree is constructed to represent sequential patterns among features in terms of the variable-order Markov model (VMM). Our approach and several baselines are evaluated on five 3D human action datasets. Extensive results show that our approach achieves superior performance for early recognition of 3D human actions.

CCS Concepts: • **Information systems → Multimedia information systems**; *Data stream mining*; • **Computing methodologies → Motion capture**;

Additional Key Words and Phrases: Marked point process, 3D action recognition, motion analysis

ACM Reference format:
Sheng Li, Kang Li, and Yun Fu. 2018. Early Recognition of 3D Human Actions. *ACM Trans. Multimedia Comput. Commun. Appl.* 14, 1s, Article 20 (March 2018), 21 pages.
https://doi.org/10.1145/3131344

1 INTRODUCTION

Human motion analysis (HMA) is a highly interdisciplinary research area that attracts great interest from computer vision, machine learning, image processing, and multimedia research communities, due to the potential applications ranging from health (assistive clinical studies), human-computer interaction, information technology (content-based video retrieval), security (intelligent surveillance), and entertainment (special effects in film production and somatosensory games) to

S. Li and K. Li contributed equally to this work.
This research is supported in part by the NSF IIS award 1651902, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Award W911NF-17-1-0367.
Authors' addresses: S. Li, Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, 02115; email: shengli@ece.neu.edu; K. Li, Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, 02115; email: kongkong115@gmail.com; Y. Fu, Department of Electrical and Computer Engineering and the College of Computer and Information Science, Northeastern University, Boston, MA, 02115; email: yunfu@ece.neu.edu.

https://doi.org/10.1145/3131344

all aspects of our daily lives (Fu 2015; Li et al. 2015, 2017). Particularly action recognition, as an important research thread of HMA, builds the basis for all of the above-mentioned applications.

Though action recognition has been well studied in the 2D scenario, 3D action recognition is quite a new topic for the fields of computer vision and multimedia. Spatial and temporal structures of 3D actions bring new challenges as well as new opportunities for the research community. Some 3D action recognition methods have been proposed in recent years (Xia and Aggarwal 2013; Luo et al. 2013; Koppula and Saxena 2013; Liu and Shao 2013; Lin et al. 2016; Mahasseni and Todorovic 2016; Liu et al. 2016), and most of them are extensions from 2D action recognition, by either customizing the features to a depth camera or adjusting 2D action recognition algorithms so that it can handle new features generated by the depth sensor. For instance, Xia and Aggarwal (2013) extended the classic 2D action feature to the 3D counterpart, Luo et al. (2013) developed a dictionary learning algorithm by incorporating group sparsity and geometry constraints to represent 3D joint features, Zhang and Tian (2015) designed a depth descriptor based on the histogram of 3D facets for action recognition, and Kong et al. (2015) presented hierarchical 3D kernel descriptors for action recognition using depth sequences. In addition, a comprehensive review on 3D skeleton-based action classification was provided in Lo Presti and La Cascia (2016). Existing works usually assume that the *full* observation of 3D actions is available in the recognition phase.

This work takes it a step further, focusing on early recognition[1] of ongoing 3D actions (i.e., only *partial* observation available). It will be beneficial for a large variety of time-critical scenarios. For example, in human-computer interaction, people's intension can be predicted by early recognition of human actions captured by sensors or depth cameras, which may greatly reduce the system response time and provide a more natural experience of communication. In many real-time somatosensory games, early recognition of human actions can reduce the sense of delay and create richer, more enjoyable gaming experiences.

In this article, we investigate two types of 3D action tracking techniques as our observation. One is the human motion capture technique where the articulated human 3D structure can be tracked accurately through a set of markers on the human body. Kinesiologists use motion capture data as an effective way to produce skeletal animations. The second technique depends on a user-friendly sensor, the depth camera. Due to the recent advent of cost-effective sensors such as Kinect, depth-camera-based human action research has become a hot topic in the area. Depth cameras provide several advantages over typical visible light cameras. First, 3D structural information can be easily captured, which helps simplify the intraclass motion variation. Second, depth information provides useful cues for background subtraction and occlusion detection. Third, depth data are generally not affected by the lighting variations.

In particular, we propose a novel approach to early classify human actions from 3D observation by modeling two types of temporal patterns: (1) **temporal dynamics** and (2) **temporal dependency**. By considering 3D action data as multivariate time series (m.t.s.) observation synchronized to a shared common clock (frames), as shown in Figure 1, we introduce a stochastic process model, called dynamic marked point process (DMP), and a variational order Markov model, called prediction of partial match (PPM), to characterize these two key aspects of the prediction problem. In our previous work (Li et al. 2014), we explored the temporal dynamics and sequential cues in multivariate time series and designed a multilevel-discretized marked point process model to address the early classification problem. In this article, we still make use of the temporal dynamic information, but focus on the early recognition of 3D human actions.

[1]In our discussion, we use "early recognition," "early classification," or "prediction" interchangeably to refer to the same learning task: "identifying the class label of 3D human actions with partial observation of temporally incomplete executions."

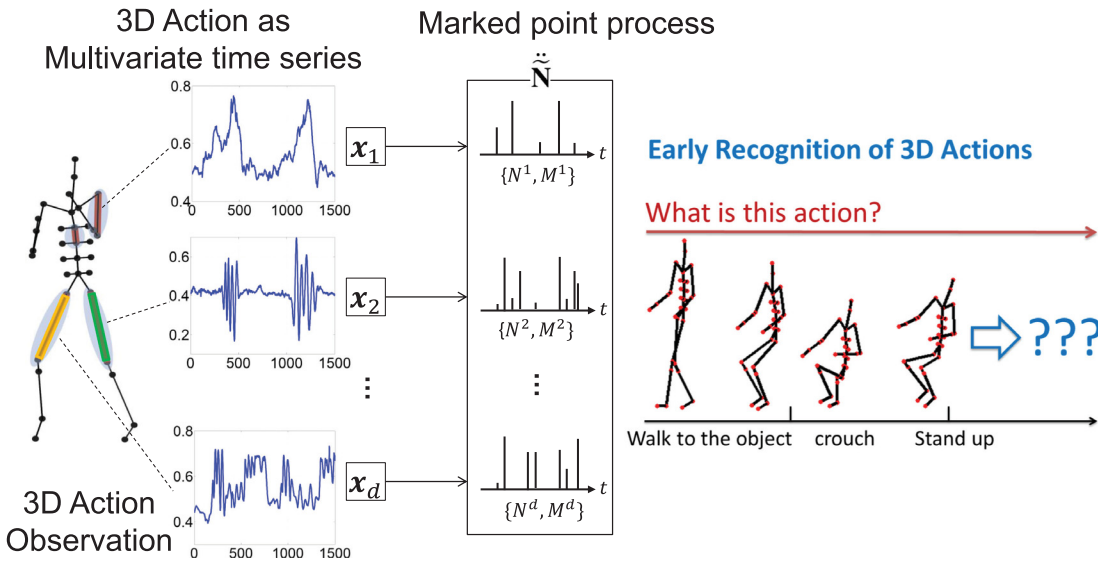


Fig. 1. Framework of early recognition of 3D human actions. The human actions are considered as multivariate time series observations, which can be interpreted as an instantiation of a multivariate marked point process.

To summarize, the main contributions of this work are summarized as follows:

First, by utilizing the m.t.s. representation for 3D action data, we propose a new dynamic marked point process model to capture the dynamic nature of human actions in 3D, which is a time-aware model that makes early recognition becomes possible.

Second, we introduce a prediction by partial matching algorithm, which captures the underlying variable-order Markov dependencies among multiple feature variables (human joints) from the 3D observation. It explores the temporal dependencies among human joints while actions are performed. The causal relationships between action segments provide important cues for early recognition.

Third, we address the problem of early recognition of human actions in 3D scenarios, where two types of 3D action observation, motion capture data and depth camera data, are evaluated.

The rest of this article is organized as follows. We briefly review some related work in Section 2. We describe the preliminaries in Section 3 and the proposed methodology in Section 4. We then show extensive experimental results in Section 5. Finally, the conclusion is drawn in Section 6.

2 RELATED WORK

In general, our work is closely related to the following topics: 3D action recognition, action prediction, early classification of time series, and point process models in computer vision.

2.1 3D Action Recognition

A large number of methods have been proposed for recognizing human actions. Here we focus on methods most related to 3D actions. Readers interested in 2D action can refer to some recent survey (Aggarwal and Ryoo 2011) on this topic. Most of the existing work (Xia and Aggarwal 2013; Luo et al. 2013; Koppula and Saxena 2013; Liu and Shao 2013; Cai et al. 2015) on 3D action recognition consists of extensions from the 2D case by either customizing the features to a depth camera or by adjusting 2D action recognition algorithms so that they can handle new challenges introduced by the depth sensor. Xia and Aggarwal (2013) extended the classic 2D action feature to

a 3D counterpart, Depth STIP, which is basically a filtering method to detect interest points from RGB-D videos with noise reduction. Luo et al. (2013) presented a discriminative dictionary learning algorithm that incorporates group sparsity and geometry constraints to represent 3D joint features. Wang et al. (2012, 2014) proposed to represent 3D actions as a set of selected joints that are considered more relevant and informative to the task. And they use a framework of multiple-kernel SVM, where each kernel corresponds to an informative joint. Kong et al. (2015) proposed hierarchical 3D kernel descriptors for action recognition using depth sequences. Vemulapalli et al. (2014a) represented the 3D skeletons as curves in a Lie group that is a curved manifold. Vemulapalli et al. (2014b) designed a new skeletal representation that explicitly models the 3D geometric relationships between various body parts for action recognition. Also, for a more standard and comprehensive evaluation of this particular task, a new dataset was also provided (Hadfield and Bowden 2013) that developed a 3D kinematics descriptor for low-latency action recognition,

Another relevant topic to our work is low-latency action recognition (Hadfield and Bowden 2013). Ellis et al. (2013) designed a latency-aware learning algorithm to train a logistic regression classifier for action recognition, and explored the accuracy/latency tradeoff over a varying number of actions. Both the observational latency and computational latency are considered in this approach. Ohn-Bar and Trivedi (2013) proposed two descriptors for spatiotemporal feature extraction from color and depth images, and evaluated the model performance with partial observations. Devanne et al. (2015) extracted a compact representation of a human action by representing the 3D coordinates of the joints and their change over time as a trajectory in a suitable action space. Promising results were observed when only processing a small portion of frames of the sequence.

Most recently, deep neural networks have been designed and applied to 3D human action detection and recognition and have achieved promising performance. Du et al. (2015) designed a recurrent neural network (RNN) to model the long-term contextual information of temporal sequences. Each human skeleton was divided into five parts according to human physical structure, and then separately fed to five subnetworks. Li et al. (2016) proposed a multitask end-to-end joint classification-regression RNN to explore the action type and temporal localization information. In particular, by leveraging the merits of the deep long short-term memory (LSTM) subnetwork, this model could automatically capture the complex long-range temporal dynamics. Liu et al. (2016) further extended the RNN-based 3D action recognition to the spatiotemporal domain by introducing a tree-structure-based traversal algorithm. By incorporating an attention module, Song et al. (2017) designed an end-to-end spatiotemporal model on top of the RNN and LSTM.

Different from the above methods, our work explicitly focuses on the early recognition of 3D human actions and builds a probabilistic model to achieve this goal.

2.2 Action Prediction

Action prediction is quite a new topic in computer vision (Kong et al. 2014). Only a few existing works specifically focus on this task. The work of Ryoo (2011) first argued that the goal of activity prediction is to recognize unfinished single actions from observation of its early stage. Two extensions of the bag-of-words (BoW) paradigm, dynamic BoW and integral BoW, are proposed to handle the sequential nature of human activities. The work of Cao et al. (2013) extended Ryoo (2011) to recognize human actions from partially observed videos, where an unobserved subsequence may occur at any time by yielding a temporal gap in the video. The work of Kong et al. (2014) proposed a discriminative model to enforce the label consistency between segments. The work of Hoai and De la Torre (2012) proposed a max-margin framework for early event detection, in which video frames are simulated as sequential event streams. The work of Lan et al. (2014) presented a hierarchical representation named hierarchical movemes for future action prediction. The new representation could characterize human movements at multiple levels of granularities, ranging

from atomic movements (e.g., an open arm) to coarser movements that cover a larger temporal extent. To implement the idea of action prediction for long-duration, more complex human activities, Li et al. (2012) and Li and Fu (2014) introduce the concept of actionlets, where the sequential nature of action units is explored for the purpose of recognizing the activity class as early as possible.

The problem of prediction (or early recognition) of 3D human actions has not been extensively studied before. Our work is an attempt to develop a prediction model for 3D human actions.

2.3 Early Classification of Time Series

While there is a vast amount of literature on classification of time series (see reviews (Fu 2011; Keogh and Kasetty 2002) and recent work (Zhang et al. 2012; Ye and Keogh 2009; Wei and Keogh 2006; Xi et al. 2006; Eruhimov et al. 2007; Katagiri et al. 2012; He et al. 2013, 2015; Li et al. 2016)), early classification of ongoing time series has been ignored until quite recently (Xing et al. 2009; Ghalwash and Obradovic 2012; Xing et al. 2011; Dachraoui et al. 2015; Lin et al. 2015). The unique and nontrivial challenge here is that either features or distance metrics formulated in previous work for classification of time series might not be robust, when whole time series are not available. Additionally, early classification always makes stricter demands on time efficiency, because the algorithm will lose its merit if it unintentionally forces us to wait till the end of the time series. To the best of our knowledge, the work of Xing et al. (2009) first explicitly proposed a solution of early classification of time series to the community, though similar concepts have been raised in other two works (Bregón et al. 2006; Rodríguez et al. 2001). They developed the ECTS (Early Classification on Time Series) algorithm, which is an extension of the 1NN classification method. ECTS evaluates neighbors both in full observation and as prefixes of time series. But their algorithm is limited only to univariate time series (u.t.s.) data and assumes that all time series samples have the same length.

Following the spirit of the classic work (Ye and Keogh 2009) on discovering interpretable time series shapelets, Ghalwash and Obradovic (2012) and Xing et al. (2011) extend it to the early classification scenarios. However, all three methods are distance-based approaches, and the inherent efficiency problem is not considered for earliness. Dachraoui et al. (2015) modeled the early classification of time series as a sequential decision-making problem and achieved promising results. In our previous work (Li et al. 2014), we designed a multilevel-discretized marked point process model to address the early classification of time-series data.

2.4 Point Process Models

As a special type of stochastic process, point process has gained a lot of attention recently in the statistical learning community because of its powerful capability of modeling and analyzing rich dynamical phenomena (Jansen and Niyogi 2009; Ge and Collins 2009; Gunawardana et al. 2011; Utasi and Benedek 2011; Kim et al. 2012; Prabhakar et al. 2010). Adopting a point process representation of random events in time opens up pattern recognition to a large class of statistical models that have seen wide applications in many fields. Jansen and Niyogi (2009) applied the point process model in the context of speech recognition, especially for obstruent super-segment decoding. But a general framework for other domains was not considered. Gunawardana et al. (2011) proposed a variant of MPP model, named the Piecewise-Constant Conditional Intensity Model (PCIM), for learning temporal dependencies in event streams. Their algorithm is evaluated on two real-world applications. The first one is modeling supercomputer event logs, and the second one is forecasting future interests of web search users. Although rich temporal structure information has been encoded, they do not consider any classification possibility from that point.

Recently, Prabhakar et al. (2010) used MPP as a representation for visual events and tried to identify temporal patterns of human interactions by applying a pairwise test for Granger causality.

Table 1. Abbreviations and Symbols

Abbr.	Description
u.t.s	univariate time series
m.t.s	multivariate time series
1NN	one nearest neighbor
DTW	dynamic time warping
MPP	marked point process
MD-MPP	multilevel-discretized marked point-process
PST	probabilistic suffix tree
VMM	variable-order Markov model
HMM	hidden Markov model

Symbol	Description
X	observation of time series with full length
X', Y'	ongoing time series
$\mathbf{X}^d$	set of d-dimensional m.t.s.
$\mathbf{D}$	time series training dataset
$\mathbf{T}$	time (index set)
$\mathbf{C}$	set of class labels
$\|X\|$	length of time series
$\mathcal{F}$	classifier
$\widetilde{\mathbf{N}}$	multivariate point process
$\ddot{\widetilde{\mathbf{N}}}$	multivariate marked point process
S	number of segments by factoring timeline
Λ	trained MD-MPP model
$\mathbf{E}$	set of events
$\overline{\mathbf{D}}_\Lambda$	set of sampled discrete event streams from model Λ
$\overline{\mathbf{D}}_{Y'}$	set of sampled discrete event streams from testing Y'
$\bar{a}_i$	discrete event stream

They come from an interpretation point of view, rather than a recognition point of view. Also, Kim et al. (2012) investigated the problem of web image prediction by developing a predictive framework based on MPP. They focus on predicting future event rather than early classification of m.t.s.. Although many algorithms based on the point process model have been successfully developed to address many real-world problems, it has not been applied to predicting human actions.

3 PRELIMINARIES

3.1 Problem Definition

In the following discussion, we first provide the definition of multivariate time series and then formally define the m.t.s. classification and m.t.s. early recognition problems. Table 1 summarizes the symbols used throughout the article.

Definition 1 (Multivariate Time Series). A multivariate time series $X = \{\mathbf{x}_t : t \in \mathbf{T}\}$ is an ordered set of real-valued observations, where $\mathbf{T}$ is the index set consisting of all possible timestamps. If $\mathbf{x}_t \in \mathbb{R}^d$, where $d > 1$, for instance, $\mathbf{x}_t = \langle x_t^1, x_t^2, \ldots, x_t^d \rangle$, then X is called a d-dimensional m.t.s..

We use m.t.s. to represent 3D action observation, which essentially is a multivariate time series synchronized to a shared common clock (each frame is a sampling timestamp). We use lowercase

letters to represent scalar values and lowercase bold letters to represent vectors. We use uppercase letters to represent time series and uppercase bold letters to represent sets.

Definition 2 (Classification of m.t.s.). An m.t.s. $X = \{\mathbf{x}_t : t \in \mathbf{T}\}$ may globally carry a class label. Given $\mathbf{C}$ as a set of class labels, and a training set $\mathbf{D} = \{\langle X_i, C_i\rangle : C_i \in \mathbf{C}, i = 1, \ldots, n\}$, the task of classification of m.t.s. is to learn a classifier, which is a function $\mathcal{F} : \mathbf{X}^d \to \mathbf{C}$, where $\mathbf{X}^d$ is the set of d-dimensional m.t.s..

Definition 3 (Early Recognition of m.t.s.). Given a training set $\mathbf{D} = \{\langle X_j, C_j\rangle : C_j \in \mathbf{C}, j = 1, \ldots, n\}$ with n m.t.s. samples, $X = \{\mathbf{x}_t : t \in \mathbf{T}\}$, where $\mathbf{T}$ is the index set consisting of all possible timestamps (frame index of 3D action observation). The task of early recognition of m.t.s. is to learn a classifier, which is a function $\mathcal{F} : \mathbf{X}' \to \mathbf{C}$, where $\mathbf{X}'$ is the set of ongoing m.t.s..

We use $|X|$ to represent the *length* of time series, namely, $X = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_{|X|}}\}$. By default, X is considered as the full-length observed time series, while a corresponding *ongoing time series* of X is denoted as $X' = \{\mathbf{x}'_{t_1}, \mathbf{x}'_{t_2}, \ldots, \mathbf{x}'_{t_{|X'|}}\}$, where $\mathbf{x}'_{t_i} = \mathbf{x}_{t_i}$ for $i = 1, \ldots, |X'|$, and $t_{|X'|} < t_{|X|}$. The ratio $p = |X'|/|X|$ is called the *progress level* of X'. It's obvious that the progress level of full-length observed time series is always 1. We use X'_p to indicate an ongoing time series with progress level p.

Specifically, we can do classification along the progress of action execution and predict the class label at different progress levels of X, generating a bunch of decisions, $\{\mathcal{F}(X'_{p_1}), \mathcal{F}(X'_{p_2}), \ldots, \mathcal{F}(X'_1)\}$. In this article, we use 5% of the full action duration as an interval of generating a new prediction result, which result in 20 rounds of classification for different action progress levels. Our goal is to construct early classification function $\mathcal{F}(Y')$ by using the knowledge learned from a *temporal dynamics* model $\Pr(Y'|\Lambda)$ and a *temporal dependency* model $\Pr(Y'|\Phi)$.

3.2 Multivariate Marked Point Process

In probability theory, a *stochastic process* is a sequence of random variables indexed by a totally ordered set $\mathbf{T}$ ("time"). *Point process* is a special type of stochastic process that is frequently used as a model for a firing pattern of random events in time. Specifically, the process counts the number of events and records the time that these events occur in a given observation time interval.

Definition 4. A d-dimensional *multivariate point process* is described by $\widetilde{\mathbf{N}} = \langle N^1, N^2, \ldots, N^d\rangle$, where $N^i = \{t^i_1, t^i_2, \ldots, t^i_m\}$ is a univariate point process, and t^i_k indicates the timestamps on which a particular "event" or "property"[2] x_i has been detected. $N^i(t)$ is the total number of observed events x_i in the interval $(0, t]$, for instance, $N^i(t^i_k) = k$. Then, $N^i(t + \Delta t) - N^i(t)$ represents the number of detections in the small region Δt. Similarly, $\widetilde{\mathbf{N}}(t) = \langle N^1(t), N^2(t), \ldots, N^d(t)\rangle$.

By letting $\Delta t \to 0$, we can have the *intensity function* $\mathbf{\Lambda}(t) = \{\lambda^i(t)\}$, which indicates the expected occurrence rate of the event x^i at time t: $\lambda^i(t) = \lim_{\Delta t \to 0} N^i(t + \Delta t) - N^i(t)$ (Daley and Vere-Jones 2003). This is the key to identifying a point process.

In many real-world applications, the time landmarks of events arise not as the only object of study but as a component of a more complex model, where each landmark is associated with other random elements $M^i = \{x^i_1, x^i_2, \ldots, x^i_m\}$, called marks, containing further information about the events. Each (t^i_k, x^i_k) is a marked point, and the sequence $\{(t^i_k, x^i_k)\}$ of marked points is referred to as a *marked point processes.*

[2]In this article, the concepts "variable," "property," or "event" are interchangeably used to refer to a certain dimension of m.t.s.

Definition 5. A d-dimensional *multivariate marked point process* is described as follows:

$$\ddot{\tilde{\mathbf{N}}} = \langle \{N^1, M^1\}, \{N^2, M^2\}, \ldots, \{N^d, M^d\} \rangle, \tag{1}$$

where $\{N^i, M^i\} = \{(t_k^i, x_k^i)\}$ on $\mathbb{R}^+ \times \mathbb{R}$ is a univariate marked point process.

4 METHODOLOGY

In this section, we first introduce our DMP model by considering the temporal dynamics in 3D human actions. After that, we incorporate the temporal dependency information and present the model DMP with prediction by partial matching (DMP+PPM).

4.1 Temporal Dynamics

We aim to build a DMP model to characterize the timing and strength information of each feature (human joint).

Given a d-dimensional m.t.s., $X = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_{|X|}}\}$, where $\mathbf{x}_t = \langle x_t^1, x_t^2, \ldots, x_t^d \rangle$, and $t = t_1, t_2, \ldots, t_{|X|}$. We consider each dimension x^i as a noisy detector of certain human joints. Those detectors generate continuous values, which indicate the angle or moving speed of human joints. We call these continuous value *marks*. Then the corresponding marked point process representation of X is

$$\ddot{\tilde{\mathbf{N}}}_X = \left\langle \left\{N_X, M_X^1\right\}, \left\{N_X, M_X^2\right\}, \ldots, \left\{N_X, M_X^d\right\} \right\rangle, \tag{2}$$

where $\{N_X, M_X^i\} = \{(t_k, x_{t_k}^i)\}$ and $k = 1, \ldots, |X|$. We can see that different variables share a common clock N_X.

For a discrete feature detector only generating 0 or 1, the marked point process $\{N_X, M_X^i\}$ can be simplified to a point process $N^i = \{t_1^i, t_2^i, \ldots, t_m^i\}$. Based on the discussions in Section 3.2, a basic representation model for m.t.s. can be descried as a stationary point process:

$$\begin{aligned} \Pr(N^i) &= \prod_{k=1}^{|X|} \frac{(\lambda^i \Delta t)^{\mathbf{1}_{N^i}(t_k)}}{\mathbf{1}_{N^i}(t_k)!} e^{-\lambda^i \Delta t} \\ &= (\lambda^i \Delta t)^{m^i} e^{-\lambda^i T}. \end{aligned} \tag{3}$$

But in our case, intensity parameter λ will depend on both time and mark. For modeling time, we evenly divide the timeline into S pieces of equal-length segments. Inside each segment, the point process is assumed stationary. $\Delta\tau = \lfloor |X|/S \rfloor$ is the segment length in terms of number of frames, so the progress level at the end of the sth segment is $p = (s\Delta\tau\Delta t)/(|X|\Delta t) = s\Delta\tau/|X|$. Then, the representation model becomes

$$\Pr(N^i) = \prod_{s=1}^{S} \frac{(\lambda^i(s)\Delta t\Delta\tau)^{m_s^i}}{m_s^i!} e^{-\lambda^i(s)\Delta t\Delta\tau}. \tag{4}$$

For the modeling mark, we assume all feature dimensions have been normalized to $[0, 1]$, respectively, which results in the mark space within $[0, 1]$. We build a multilevel discretization of the mark space by splitting it into L levels. Then the point process factors into L levels of independent processes operating in each level of the mark space for a particular feature. Finally, we build a representation model as follows:

$$\begin{aligned} &\Pr(\{N_X, M_X^i\}) \\ &= \prod_{l=1}^{L} \prod_{s=1}^{S} \frac{(\lambda^i(s,l)\Delta t\Delta\tau)^{m_{s,l}^i}}{m_{s,l}^i!} e^{-\lambda^i(s,l)\Delta t\Delta\tau}, \end{aligned} \tag{5}$$

where $m^i_{s,l}$ is the number of landmarks of feature x^i in the sample's sth segment and lth level of mark space.

Given a training m.t.s. dataset $\mathbf{D}$ and multivariate marked point process representation $\widetilde{\ddot{\mathbf{N}}}$, the data likelihood can be computed as

$$\begin{aligned}\Pr(\widetilde{\ddot{\mathbf{N}}}|\mathbf{D}) &= \prod_{i=1}^{d} \Pr(\{N^i, M^i\}|\mathbf{D}) \\ &= \prod_{i=1}^{d}\prod_{l=1}^{L}\prod_{s=1}^{S} \frac{(\lambda^i(s,l,\mathbf{D})\Delta t\Delta\tau)^{m^i_{s,l}}}{m^i_{s,l}!} e^{-\lambda^i(s,l,\mathbf{D})\Delta t\Delta\tau},\end{aligned} \tag{6}$$

where the intensity function $\lambda^i(s,l,\mathbf{D})$ depends on the feature (human joint), the segment (action progress level), the mark-space level (observation value), and the training data. Now, we can formalize two key steps in early classification.

Step One: Learning DMP. Given n training samples, the maximum log-likelihood estimation (MLE) (Bishop 2006) is utilized to solve the problem. We follow the standard procedures in MLE and skip the details here. The estimated model parameters can be written as

$$\lambda^{i*}(s,l,\mathbf{D}) = \frac{\Sigma_{j=1}^{n} m^i_{j,s,l}}{\Sigma_{j=1}^{n}\Delta t\Delta\tau_j}, \tag{7}$$

where $m^i_{j,s,l}$ is the number of landmarks of event x^i in the jth training sample's sth time division and lth level of mark space.

Step Two: Early Recognition. Given an ongoing testing m.t.s Y' and a trained model, $\mathbf{\Lambda} = \{\lambda_{i,s,l}|L,S,\mathbf{D}\}$ (for simplicity, we use $\lambda_{i,s,l}$ to represent $\lambda^{i*}(s,l,\mathbf{D})$). First, we construct a structure of Y' by factoring it over timeline and mark space in the same way as the trained model, so that dynamics can be matched. Finally, the likelihood of Y' can be written as

$$\Pr(Y'|\mathbf{\Lambda}) \propto \prod_{i=1}^{d}\prod_{l=1}^{L}\prod_{s=1}^{\lceil p^*S\rceil} (\lambda_{i,s,l}\Delta\tau^*)^{m^i_{s,l}} e^{-\lambda_{i,s,l}\Delta\tau^*}, \tag{8}$$

where $p^* = |Y'|/(\Delta\tau^* S)$ is the progress level of Y'.

Since the length of m.t.s. can be different, given an ongoing testing m.t.s., we may not know when it will be finished. Therefore, we need to "guess" the "right" progress level of it first. Then we can apply our model appropriately. This is an important merit of our approach. Algorithm 1 shows the detail of how we compute p^*. At the beginning, we need to identify the possible range of $\Delta\tau$. After that, we estimate the minimum number of segments for the testing m.t.s. Y'. Finally, we evaluate the likelihood and provide the estimation of $\Delta\tau^*$.

Discussions: We notice that many existing techniques such as the hidden Markov model (HMM) could also be adapted to the 3D action recognition problem with partial observations. However, compared to the existing methods, the proposed DMP model has the following advantages: First, the marked point process has been shown as a good fit for characterizing the discrete multivariate time-series data such as 3D human actions, which will be further validated in the experiments. Second, different from HMM, which is a generative model and trains one model for each class, the proposed DMP model could learn a discriminative classifier that is more suitable for the recognition tasks.

ALGORITHM 1: Guess the Progress Level p^*

(1) **Find the possible range from training set**: Let $\Delta\tau_{\min} = \min\{|X_j|/S : j \in \{1, \ldots, n\}\}$, $\Delta\tau_{\max} = \max\{|X_j|/S : j \in \{1, \ldots, n\}$, Then, $\tau_{\mathbf{D}} = [\Delta\tau_{\min}, \Delta\tau_{\max}]$.
(2) **Determine the minimum number of segments**: $S' = \min\{\lceil |Y'|/\Delta\tau \rceil : \Delta\tau \in \tau_{\mathbf{D}}\}$, which ensures that different guesses of $\Delta\tau$ will be evaluated with the same number of segments, so that the likelihoods computed in step 3 will be comparable.
(3) **Evaluate the likelihood**:

$$\Delta\tau^* = \arg\max_{\Delta\tau \in \tau_{\mathbf{D}}} \prod_{i=1}^{d} \prod_{l=1}^{L} \prod_{s=1}^{S'} \frac{(\lambda_{i,s,l}\Delta\tau)^{m_{s,l}^{i}(\Delta\tau)}}{e^{\lambda_{i,s,l}\Delta\tau}}.$$

(4) **Estimate the progress level**: $p^* = |Y'|/(\Delta\tau^* S)$.

ALGORITHM 2: Temporal Dependency Model $\Pr(Y'|\Phi)$, Prediction of Partial Matching

—Assume Markov model of order k. Let the input discrete event stream be sequence x.
—For each symbol x_i that is in sequence x on place i:
  —Update probability of sequence y, $y = [x_{i-k} \ldots x_i]$.
  —Let y be a symbol in the target alphabet.
  —Update all relevant structures in DMP.
  —Perform the rest of the needed steps required in DMP.
—Output x_i uncoded and skip to the next symbol.
—Add shortened symbol to the alphabet.

The DMP model is very efficient for two reasons. First, the parameter estimation of DMP mainly depends on the maximum log-likelihood estimation, which could be solved efficiently. The time complexity is about $O(dLN)$, where d is the dimension of the sample, L is the number of levels, and N is the sample size. Second, the representation of the DMP model is very simple and compact. In this way, the high-dimensional 3D human action sequences could be represented as a very compact and low-dimensional representation, which leads to efficient testing speed in practice.

4.2 Temporal Dependency

For 3D human actions, the corresponding m.t.s. observation always has strong correlations among features (human joints). For instance, in the execution of a particular human action, a few joints will change their angles immediately after another few joints rotate to some degree according to the underlying cognitive "recipe" of that action. The identification of temporal dependencies among features allows us to utilize these causal patterns for early recognition, which improves the reasoning capability of our model. As a complement to the proposed *temporal dynamic* model, we introduce the *temporal dependency* component to DPM and propose the DMP+PPM model.

To implement the notion of *temporal dependency*, we first generate representative discrete feature sequences from continuous m.t.s. observations by sampling the feature ID according to the rate of occurrence (intensity function) of features at different time divisions (segments), which results in a discrete feature sequence. After sampling a significant number of discrete feature sequences, the temporal dependency relationship among features will be well preserved in the sampling set. Then the task of finding temporal dependencies becomes a problem of mining sequential patterns.

Specifically, let $\mathbf{E} = \{e^i : i = 1, \ldots, d \times L\}$ be the set of features.[3] And $\overline{\mathbf{D}}_{\mathbf{\Lambda}} = \{\bar{a}_1, \ldots, \bar{a}_\upsilon\}$ consists of υ times sampling according to $\mathbf{\Lambda}$. For instance, $\bar{a}_r = \{e_s^r\}_{s=1}^S, r \in \{1, \ldots, \upsilon\}$ is a sampled *feature sequence*, which means at the jth segment, we sampled one feature $e_s^r \in \mathbf{E}$. We can easily notice that $\bar{a}_i \in \mathbf{E}^*$, $|\bar{a}_i| = S$. Specific sampling probability of each feature at a particular time division (segment) can be computed according to

$$\Pr_{\text{sample}}(\text{event} = e|\text{segment} = s) = \frac{\lambda_{e,s}}{\sum_{e' \in \mathbf{E}} \lambda_{e',s}}. \tag{9}$$

Given the sampled feature sequence set, now the goal is to learn a model $\Phi = \{\phi(e|h) : h \in \mathbf{E}^*, e \in \mathbf{E}\}$, which associates a history h with the next possible feature e. We call function $\phi(e|h)$ the *next event probability function*. If we define the *history* at the jth time segment of feature sequence $\bar{a}^i$ as the subsequence $h_j(\bar{a}^i) = \{e_j^i | j \leq S\}$, then the log-likelihood of feature sequence $\bar{a}^i$, given a *temporal dependency* model Φ, can be written as

$$\Pr(\bar{a}^i|\Phi) = \sum_{j=1}^{S} \log \phi\left(e_j^i | h_{j-1}\left(\bar{a}^i\right)\right). \tag{10}$$

Given an ongoing testing m.t.s. Y' and a trained model, $\Phi = \{\phi(e|h)|\overline{\mathbf{D}}_{\mathbf{\Lambda}}\}$. We can sample feature sequence set from Y' in the same way, $\overline{\mathbf{D}}_{\mathbf{\Lambda}_{Y'}} = \{\bar{b}_1, \ldots, \bar{b}_w\}$. Then, the likelihood of Y' is

$$\Pr(Y'|\Phi) \propto \sum_{i=1}^{w} \Pr(\bar{b}_i|\Phi)). \tag{11}$$

In terms of specific implementation, we adopt the variable-order Markov model (VMM) (Begleiter et al. 2004), which is a category of algorithms for prediction of discrete sequences. It can capture both large- and small-order Markov dependencies. Therefore, it can encode richer and more flexible temporal dependencies. This can be done efficiently by the prediction by partial matching algorithm (Cleary and Witten 1984), which is an adaptive statistical data compression technique that uses a set of previous symbols in the uncompressed symbol stream to predict the next symbol in the stream. Algorithm 2 shows the details of this process. It basically shows the standard procedures of the conventional PPM algorithm, which was originally used for natural language text data compression. In our scenario, the alphabet means the set of events, and the basis algorithm is actually the proposed DMP model described in Section 4.1. We use DMP+PPM to denote our approach with a temporal dependency module.

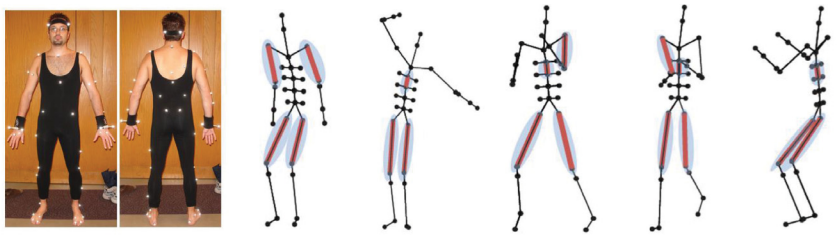
5 EXPERIMENTAL STUDIES

In this section, we present a comprehensive evaluation of our methods (DMP and DMP+PPM) on five 3D action datasets.

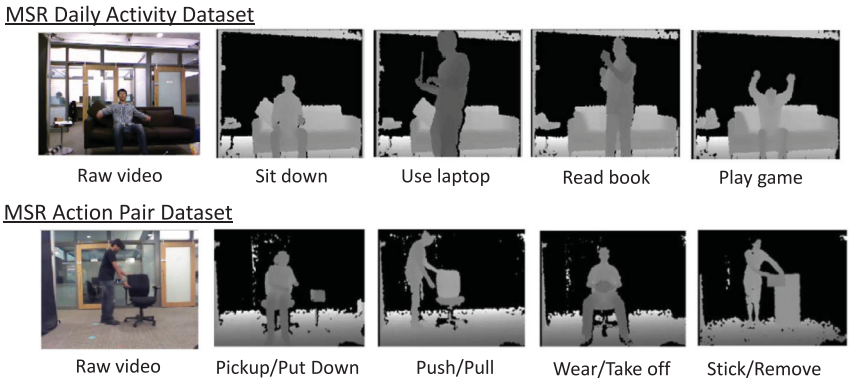
5.1 Datasets

We utilized five real-world datasets: CMU Human Motion Capture dataset (CMU), MSR Action 3D dataset, MSR 3D Action Pair dataset (Li et al. 2010), UT Kinect-Action dataset (Xia et al. 2012), and NTU RGB+D dataset (Shahroudy et al. 2016). The following details the collection and preprocessing of the five datasets. To evaluate the performance of our method on 3D human action recognition by using different types of raw features, we employ the body angle features for the CMU Motion Capture dataset and 3D body joint positions for the other datasets.

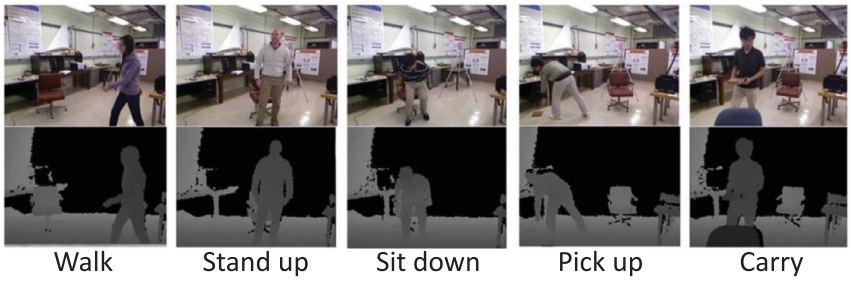
[3]With multilevel-discretized representation, the total number of features becomes $d \times L$. The DMP model can be rewritten as $\mathbf{\Lambda} = \{\lambda_{e,s} | e \in \mathbf{E}, s \in \{1, \ldots, S\}\}$ for convenience.



(a) CMU Motion Capture dataset



(b) MSR Action/Action-Pair 3D dataset



(c) UT Kinect-Action Dataset

Fig. 2. Evaluation datasets.

The **CMU Motion Capture dataset** was composed of dozens of actions performed by over 100 subjects. In our experiment, we choose the MoCa data of nine common action classes performed by a diverse number of subjects, which consists of 10 samples per class on average (total 91 samples) with an average duration of 839 frames. The nine action classes include *walk*, *run*, *jump*, *pick up*, *sitting on a motorcycle*, *cartwheel*, *boxing*, *chicken dance*, and *golf swing*. The human body is defined by a full body model of 34 bones with hierarchical structures. The action is specified by m.t.s. observations on motion angles of body bones, which describe the dynamic relationships between bones, as well as the global motion of the full body. See Figure 2(a) for the visualized body model and the definition of hierarchical structure. The original full-body degrees of freedom (DOFs) are 62. However, to reduce the computational burden, we discard some unimportant joint angles, such as fingers, thumbs, toes, and so forth, in the experiments. Finally, we select 19 body angles, which cover the DoFs of the humerus, radius, femur, tibia, and upper back. We evaluate the classification accuracy by using the "leave one out" strategy on this dataset.

The **MSR-Action 3D dataset** (Li et al. 2010) is a 3D action dataset of depth information collected from a depth camera. This dataset contains 20 actions: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, sideboxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up, and throw*. Ten subjects perform each action three times. The frame rate is 15 frames/second and resolution is 640×480. The dataset in total has 567 action samples. Some examples of the depth images are shown in Figure 2(b) in the first row. Three channels are recorded: depth maps, skeleton joint positions, and RGB video. In our experiments, we use skeleton joints as features, since our approach prefers features with strong semantic meaning so that correlations between variables are more distinctive for classification. The first five subjects are selected for training, and the other five subjects are used for testing.

The **MSR-3D Action Pair dataset** (Li et al. 2010) contains depth sequences of human-object interactions. The dataset contains six different pairs of actions: *pick up a box/put down a box, lift a box/place a box, push a chair/pull a chair, wear a hat/take off a hat, put on a backpack/take off a backpack, and stick a poster/remove a poster*, and in total has 370 videos. As in MSR Action 3D, each action is performed by 10 different subjects 3 times. Videos of five subjects are used for testing, and the other five subjects are used for training. Example frames are displayed in Figure 2(b) in the second row. As with the MSR-Action 3D dataset, we also use the skeleton joint as time-series features.

The **UT Kinect-Action dataset** (Xia et al. 2012) contains 10 different types of human actions in indoor settings: *walk, sit down, stand up, pick up, carry, throw, push, pull, wave,* and *clap hands*. A single stationary Kinect is used to capture both RGB images and depth maps. Ten subjects perform each action 2 times, resulting in a total 200 action samples in this dataset. Example frames, RGB images, and depth maps are displayed in Figure 2(c). The skeleton joints of each action video are employed as features. One hundred action samples are used for training, and the rest of the samples are used for testing.

The **NTU RGB+D dataset (NTU)** (Shahroudy et al. 2016) is currently the largest action recognition dataset with high-quality skeletons. It contains 56,880 sequences (with 4 million frames) of 60 classes, including cross-subject and cross-view settings. In this article, we only consider the cross-subject setting. Each person has 25 joints. We apply a similar normalization preprocessing step to have position and view invariance (Shahroudy et al. 2016). Since several baselines are very time consuming, it is impractical to compare the performance on the full dataset. In our experiments, we randomly select 1,200 sequences (i.e., 20 sequences per class) to create the dataset for training and testing.

5.2 Performance Comparison

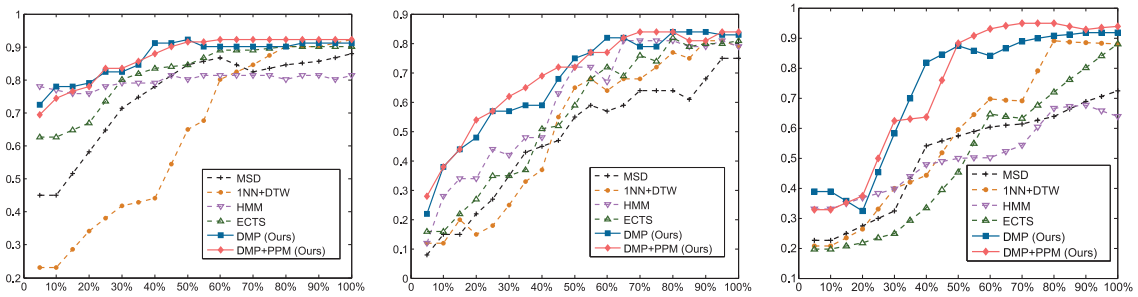
We compare our algorithms of m.t.s. early recognition (DMP and DMP+PPM) with existing alternatives that we discussed in Section 2, including 1NN with DTW (1NN+DTW) as in Keogh (2002), the ECTS algorithm as in Xing et al. (2009), multivariate shapelets detection (MSD) as in Ghalwash and Obradovic (2012), and HMM (Ghalwash et al. 2012). Table 2 summarizes the four baselines used in the experiments.

Different from traditional classification tasks, for early recognition, we focus on the predictive power of each method. An early classifier should use an observation ratio as small as possible to make an accurate prediction. To evaluate this, we do classification along the progress of time series, and predict the class label at different progress levels (observation ratio) of time series. Specifically, we use 5% of full m.t.s. duration as an interval of generating a new prediction result.

Model Construction. For the CMU Motion Capture Data, we construct a DMP model by splitting mark space into 10 levels ($L = 10$) and dividing the timeline into 20 pieces of equal-length segments ($S = 20$). To construct a DMP+PPM model, we train an order-3-bounded PPM ($O = 3$)

Table 2. Summary of the Four Baselines Used for Quantitative Comparison with Our Algorithm

Methods	Rationale	Description
One-nearest-neighbor DTW (1NN+DTW)	The state-of-the-art time-series classification algorithm	The dynamic time warping (DTW)-based distance measurements between test and training time series are computed for use in the 1NN classifier. For the m.t.s., the overall distance is measured as the average of the u.t.s. distances for all components.
Early Classification on Time Series (ECTS)	An extension of 1NN classifier to achieve early classification	The MPL (Minimum Prediction Length) for a cluster of similar time series are computed first. At the testing phase, the learned MPLs are used to select the nearest neighbor from only "qualified" candidates in terms of MPL. For the m.t.s., the overall distance is measured as the average of the u.t.s. distances for all components.
Multivariate Shapelets Detection (MSD)	An extension of time-series shapelets to achieve early classification	Multivariate shapelets are extracted using a sliding-window-based strategy. These shapelets are then pruned according to the weighted information gain.
Hidden Markov Model (HMM)	An effective statistical model for temporal pattern recognition	The HMM is selected as a representative of generative-model-based methods. A model is trained for each class. Decisions are based on likelihood ranking.



(a) CMU Motion Capture (b) MSR Action3D (c) UT Kinect-Action Dataset

Fig. 3. Performance comparisons on three datasets (see text for detailed discussions). In each figure, the vertical axis is the classification accuracy averaged over all classes, and the horizontal axis is the observation ratio, which can be viewed as the normalized timeline ($(0, T] \rightarrow (0, 1]$).

first, then do 100 times sampling ($w = 100$) of feature sequences for each m.t.s. at the testing phase. For the MSR Action 3D and 3D Action Pair datasets, we set $L = 12$, $S = 20$, $O = 3$, and $w = 100$. For the UT Kinect-Action dataset, we set $L = 8$, $S = 20$, $O = 2$, and $w = 100$.

Results. Figure 3 summarizes the quantitative comparison between our methods and four baselines. These graphs help us make the following observations:

(1) Our algorithms significantly outperform all the compared methods in most cases and achieve high prediction accuracy over different levels of observation ratios. In terms of

Table 3. Performance Comparisons on MSR 3D Action Pair Dataset (Percentage as Observation Ratios)

Methods	20%	40%	60%	80%	100%
MSD	0.17	0.35	0.44	0.56	0.62
1NN+DTW	0.21	0.38	0.48	0.62	0.74
HMM	0.31	0.36	0.59	0.72	0.76
ECTS	0.27	0.44	0.62	0.65	0.77
DMP (Ours)	**0.38**	0.49	0.67	**0.72**	**0.82**
DMP+PPM (Ours)	0.34	**0.52**	**0.67**	0.71	0.79

Bold font denotes the highest prediction accuracy in each setting.

Table 4. Performance Comparisons on NTU RGB+D Dataset (Percentage as Observation Ratios)

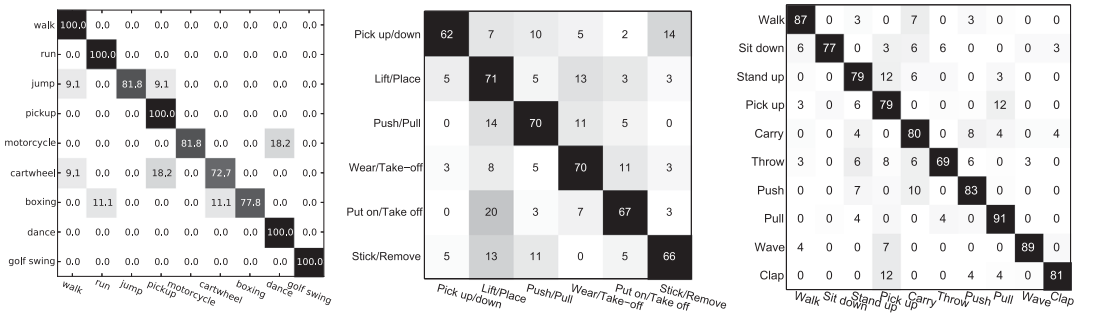
Methods	20%	40%	60%	80%	100%
MSD	0.18	0.25	0.32	0.39	0.46
1NN+DTW	0.21	0.29	0.42	0.52	0.58
HMM	0.20	0.26	0.43	0.56	0.61
ECTS	0.25	0.30	0.45	0.53	0.65
DMP (Ours)	0.29	0.37	0.50	**0.63**	0.70
DMP+PPM (Ours)	**0.31**	**0.38**	**0.54**	0.60	**0.72**

Bold font denotes the highest prediction accuracy in each setting.

full-length classification (at observation ratio 100%), 1NN-DTW is the most comparable one to ours, which demonstrates its robustness as the state-of-the-art method for time-series classification. At early stages of observation (<30%), MSD and ECTS can outperform 1NN-DTW to accomplish better early classification due to their designs on utilizing early cues. As a latent state model, HMM is relatively less dependent on full-length observation. Table 3 and Table 4 show detailed comparisons of six methods on the MSR 3D Action Pair dataset and the NTU RGB+D dataset. DMP and DMP+PPM achieve better results than other baselines.

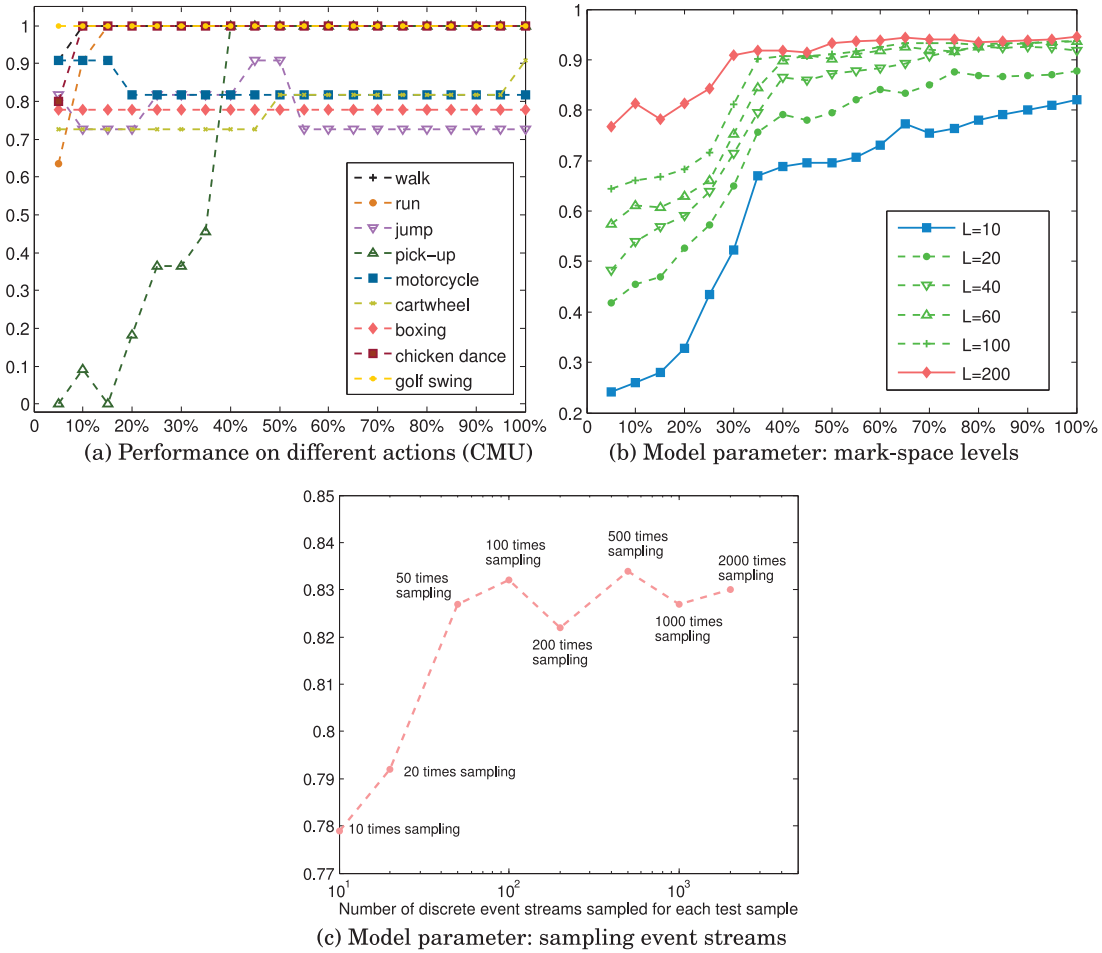
(2) Each type of 3D action data has different predictability, which means the discriminative segments of m.t.s. may appear at different stages of time series. As illustrated in Figure 3, we achieved near-optimal classification accuracy at the observation ratio of 40% in the CMU motion capture data, and 60% in the MSR 3D action data. Figure 4(a) shows the corresponding detailed results in a confusion matrix in CMU motion capture data when the observation ration is 40%. Figures 4(b) and 4(c) show the confusion matrices on the MSR 3D Action Pair dataset and the UT Kinect-Action dataset, when the observation ratios are 60% and 50%, respectively. We can observe that in these cases, our method clearly achieves good performance with certain earliness. In addition, better earliness and recognition accuracy are observed on the Motion Capture dataset, and the reason might be that this dataset is relatively cleaner than depth camera data, where each feature dimension exactly corresponds to a particular joint in the human body model shown in Figure 2.

(3) We have a few interesting observations that are reasonably in accordance with our domain knowledge. In Figure 5(a), we present detailed performance of our approach over nine different action classes in the CMU Motion dataset. The action "pick up" is difficult to be recognized at early stages, because it is executed by first walking to the object, then



(a) CMU Motion (at **40%** ratio) (b) MSR 3D Action-Pair (at **60%**) (c) UT Kinect-Action (at **50%**)

Fig. 4. Confusion matrices on three datasets.



(a) Performance on different actions (CMU) (b) Model parameter: mark-space levels

(c) Model parameter: sampling event streams

Fig. 5. Detailed results (a) and model parameter analysis (b and c) on CMU Motion Capture dataset.

picking it up. The component subaction "walking to object" makes it confusing with the class "walk." Another component subaction, "crouching to pick up object," makes it confusing with the class "jump." In the UT Kinect-Action action, we also observed several failure cases that involved the confusion of "pick up" and "pull." One possible reason is that the DMP model only uses L levels to characterize the complex actions, while L is a

relatively small number. As a result, some detailed information might be lost after such a compression. However, lots of details might be necessary to distinguish some actions that appear similar at the early stage.

(4) Temporal dependency among variables is important for early stages of recognition, by comparing the performance of DMP and DMP+PPM from Table 3. On the CMU Motion dataset, two algorithms achieve comparable performance. On the MSR Action 3D dataset, DMP+PPM performs better than DMP in most cases. Notably, when the observation ratio is 20% or 40% (i.e., very early recognition), DMP+PPM improves the accuracy of DMP by over 6%. It demonstrates that PPM helps incorporate the temporal dependency information for effective early recognition. On the MSR 3D Action Pair dataset, DMP+PPM achieves higher accuracy than DMP when the observation ratio is 40%, but lower accuracy when the ratio is 20%. This might be due to the complexity of actions in the 3D Action Pair dataset. In other words, a very short observation might not be sufficient for predicting the complex actions in this scenario.

(5) In addition, we also compare our algorithm with the Moving Pose (MP) (Zanfir et al. 2013), which is a fast nonparametric model for early recognition. On the MSR Action 3D dataset, MP achieves accuracies of 43% and 74% when the observation ratios are 20% and 40%, respectively, while our algorithm DMP+PPM achieves 54% and 69% in these two cases, as shown in Table 3. Compared with MP, our algorithm is more suitable for recognizing 3D human actions at the very early stage.

5.3 Model Parameters

To show the impact of model parameters on the results, we present Figure 5(b) and Figure 5(c) as illustrations of two key parameters in our approach: one is the number of mark-space levels L; the other is the sampled event stream number w. Figure 5(b) shows the trend of performance improvement with increasing number of mark-space levels, which suggests that this dataset prefers a more detailed discretization of mark space.

Figure 5(c) proved our claim that a "sufficient" number of sampling of discrete event streams will preserve most of the sequential pattern information. Also, the "sufficient" time is not necessarily a very big number. As shown in the figure, a relatively small number (100) of samplings can achieve near-optimal performance on the CMU Motion Capture dataset.

5.4 Time Efficiency

Since our goal is to identify the 3D actions quickly before we observe the full length of actions, the algorithm efficiency becomes very important. All previous work (Xing et al. 2009, 2011; Ghalwash and Obradovic 2012) consists of extensions of traditional distance-based approaches, which are computationally too demanding. However, in many cases, the practical merit of early recognition methods lies in a quick and accurate recognition. Thus, another important advantage of our algorithms is the time efficiency compared to other alternatives.

Figure 6 shows the runtime comparison (per 100 samples) at the testing phase of each algorithm. All methods are tested on a 2.4GHz four-cores workstation with 24.0GB memory. Both of our algorithms achieve better time efficiency and are more than one order of magnitude faster.

6 CONCLUSION

Action recognition is an important research study of human motion analysis. In recent years, 3D observation-based action recognition has been receiving increasing interest in the multimedia and computer vision community due to the recent advent of cost-effective sensors, such as depth

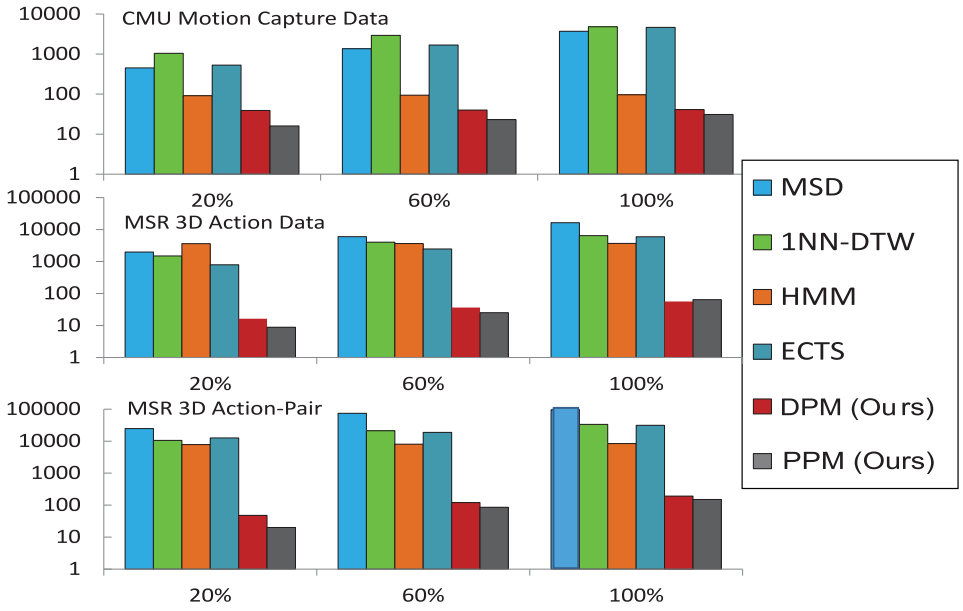


Fig. 6. Time efficiency comparison. Since the time expenses of different methods are not in the same order of magnitude, we use logarithmic scale to make them visible at the same time.

camera Kinect. This work goes one step further, focusing on early recognition of ongoing 3D actions, which is beneficial for a large variety of time-critical applications.

In this article, we propose a novel approach for early recognition of 3D action data by modeling two types of temporal patterns: *temporal dynamics* and *temporal dependency*. The major contributions include a dynamic marked point process model for representing m.t.s. and a time-dependency model prediction by partial matching to characterize the temporal dependency relationships among multiple feature dimensions. We have empirically shown that our approach is superior in the early recognition task for 3D actions in terms of prediction accuracy. Our approach does not assume that all the action samples have the same length of duration, but it relies on the segments of different progress levels to be roughly matched among samples in the same class.

In our future work, we would like to consider both the spatial and temporal information when modeling the 3D human actions with marked point process. In addition, we will extend this model to more general cases, where more complex intensity functions can be applied to capture the structure of data within different domains.

REFERENCES

J. K. Aggarwal and Michael S. Ryoo. 2011. Human activity analysis: A review. *ACM Computing Surveys (CSUR)* 43, 3 (2011), 16.

Ron Begleiter, Ran El-Yaniv, and Golan Yona. 2004. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research* 22 (2004), 385–421.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

Aníbal Bregón, M. Aránzazu Simón, Juan José Rodríguez, Carlos Alonso, Belarmino Pulido, and Isaac Moro. 2006. Early fault classification in dynamic systems using case-based reasoning. In *Current Topics in Artificial Intelligence*. Springer, 211–220.

Xingyang Cai, Wengang Zhou, and Houqiang Li. 2015. Attribute mining for scalable 3D human action recognition. In *Proceedings of the 23rd ACM International Conference on Multimedia*. ACM, 1075–1078.

Yu Cao, Daniel Barrett, Andrei Barbu, Siddharth Narayanaswamy, Haonan Yu, Aaron Michaux, Yuewei Lin, Sven Dickinson, Jeffrey Mark Siskind, and Song Wang. 2013. Recognizing human activities from partially observed videos. In *IEEE International Conference on Computer Vision and Pattern Recognition*. 2658–2665.

John G. Cleary and Ian Witten. 1984. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications* 32, 4 (1984), 396–402.

CMU Graphics Lab Motion Capture Database. Retrieved from http://mocap.cs.cmu.edu.

Asma Dachraoui, Alexis Bondu, and Antoine Cornuéjols. 2015. Early classification of time series as a non myopic sequential decision making problem. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases Part I*. 433–447.

Daryl J. Daley and David Vere-Jones. 2003. *An Introduction to the Theory of Point Processes*. Vol. 1. Springer-Verlag.

Maxime Devanne, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. 2015. 3-D human action recognition by shape analysis of motion trajectories on Riemannian manifold. *IEEE Transactions on Cybernetics* 45, 7 (2015), 1340–1352.

Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1110–1118.

Chris Ellis, Syed Zain Masood, Marshall F. Tappen, Joseph J. LaViola, and Rahul Sukthankar. 2013. Exploring the trade-off between accuracy and observational latency in action recognition. *International Journal of Computer Vision* 101, 3 (2013), 420–436.

Victor Eruhimov, Vladimir Martyanov, and Eugene Tuv. 2007. Constructing high dimensional feature space for time series classification. In *Knowledge Discovery in Databases: PKDD*. Springer, 414–421.

Tak-chung Fu. 2011. A review on time series data mining. *Engineering Applications of Artificial Intelligence* 24, 1 (2011), 164–181.

Yun Fu. 2015. *Human Activity Recognition and Prediction*. Springer.

Weina Ge and Robert T. Collins. 2009. Marked point processes for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2913–2920.

Mohamed Ghalwash and Zoran Obradovic. 2012. Early classification of multivariate temporal observations by extraction of interpretable shapelets. *BMC Bioinformatics* 13, 1 (2012), 195.

Mohamed F. Ghalwash, Dusan Ramljak, and Zoran Obradovic. 2012. Early classification of multivariate time series using a hybrid HMM/SVM model. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine*. 1–6.

Asela Gunawardana, Christopher Meek, and Puyang Xu. 2011. A model for temporal dependencies in event streams. In *Proceedings of the Annual Conference on Neural Information Processing Systems*. 1962–1970.

Simon Hadfield and Richard Bowden. 2013. Hollywood 3d: Recognizing actions in 3d natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3398–3405.

Guoliang He, Yong Duan, Rong Peng, Xiaoyuan Jing, Tieyun Qian, and Lingling Wang. 2015. Early classification on multivariate time series. *Neurocomputing* 149 (2015), 777–787.

Guoliang He, Yong Duan, Tieyun Qian, and Xu Chen. 2013. Early prediction on imbalanced multivariate time series. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*. ACM, 1889–1892.

Minh Hoai and Fernando De la Torre. 2012. Max-margin early event detectors. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. 2863–2870.

Aren Jansen and Partha Niyogi. 2009. Point process models for event-based speech recognition. *Speech Communication* 51, 12 (2009), 1155–1168.

Hideki Katagiri, Ichiro Nishizaki, Tomohiro Hayashida, and Takanori Kadoma. 2012. Multiobjective evolutionary optimization of training and topology of recurrent neural networks for time-series prediction. *Computer Journal* 55, 3 (2012), 325–336.

Eamonn Keogh. 2002. Exact indexing of dynamic time warping. In *Proceedings of the 28th International Conference on Very Large Data Bases*. 406–417.

Eamonn Keogh and Shruti Kasetty. 2002. On the need for time series data mining benchmarks: A survey and empirical demonstration. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 102–111.

Gunhee Kim, Li Fei-Fei, and Eric P. Xing. 2012. Web image prediction using multivariate point processes. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1068–1076.

Yu Kong, Dmitry Kit, and Yun Fu. 2014. A discriminative model with multiple temporal scales for action prediction. In *Proceedings of the European Conference on Computer Vision*. Springer, 596–611.

Yu Kong, Behnam Satarboroujeni, and Yun Fu. 2015. Hierarchical 3D kernel descriptors for action recognition using depth sequences. In *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*. 1–6.

Hema Koppula and Ashutosh Saxena. 2013. Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation. In *Proceedings of the 30th International Conference on Machine Learning*. 792–800.

Tian Lan, Tsung-Chuan Chen, and Silvio Savarese. 2014. A hierarchical representation for future action prediction. In *Proceedings of the European Conference on Computer Vision, Part III*. 689–704.

Kang Li and Yun Fu. 2014. Prediction of human activity by discovering temporal sequence patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 8 (2014), 1644–1657.

Kang Li, Jie Hu, and Yun Fu. 2012. Modeling complex temporal composition of actionlets for activity prediction. In *Proceedings of the European Conference on Computer Vision*. Springer, 286–299.

Kang Li, Sheng Li, and Yun Fu. 2014. Early classification of ongoing observation. In *Proceedings of the IEEE International Conference on Data Mining*. 310–319.

Kang Li, Sheng Li, Sangmin Oh, and Yun Fu. 2017. Videography-based unconstrained video analysis. *IEEE Transactions on Image Processing* 26, 5 (2017), 2261–2273.

Sheng Li, Kang Li, and Yun Fu. 2015. Temporal subspace clustering for human motion segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. 4453–4461.

Sheng Li, Yaliang Li, and Yun Fu. 2016. Multi-view time series classification: A discriminative bilinear projection approach. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 989–998.

Wanqing Li, Zhengyou Zhang, and Zicheng Liu. 2010. Action recognition based on a bag of 3d points. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 9–14.

Yanghao Li, Cuiling Lan, Junliang Xing, Wenjun Zeng, Chunfeng Yuan, and Jiaying Liu. 2016. Online human action detection using joint classification-regression recurrent neural networks. In *Proceedings of the European Conference on Computer Vision*. 203–220.

Liang Lin, Keze Wang, Wangmeng Zuo, Meng Wang, Jiebo Luo, and Lei Zhang. 2016. A deep structured model with radius–margin bound for 3D human activity recognition. *International Journal of Computer Vision* 118, 2 (2016), 256–273.

Yu-Feng Lin, Hsuan-Hsu Chen, Vincent S. Tseng, and Jian Pei. 2015. Reliable early classification on multivariate time series with numerical and categorical attributes. In *Proceedings of the 19th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining - Part I*. 199–211.

Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. 2016. Spatio-temporal lstm with trust gates for 3d human action recognition. In *Proceedings of the European Conference on Computer Vision*. Springer, 816–833.

Li Liu and Ling Shao. 2013. Learning discriminative representations from RGB-D video data. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*. 1493–1500.

Liliana Lo Presti and Marco La Cascia. 2016. 3D skeleton-based human action classification. *Pattern Recognition* 53, C (2016), 130–147.

Jiajia Luo, Wei Wang, and Hairong Qi. 2013. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'13)*. IEEE, 1809–1816.

Behrooz Mahasseni and Sinisa Todorovic. 2016. Regularizing long short term memory with 3D human-skeleton sequences for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3054–3062.

Eshed Ohn-Bar and Mohan Trivedi. 2013. Joint angles similarities and HOG2 for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 465–470.

Karthir Prabhakar, Sangmin Oh, Ping Wang, Gregory D. Abowd, and James M. Rehg 2010. Temporal causality for the analysis of visual events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1967–1974.

Juan J. Rodríguez, Carlos J. Alonso, and Henrik Boström. 2001. Boosting interval based literals. *Intelligent Data Analysis* 5, 3 (2001), 245–262.

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*. 1036–1043.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. NTU RGB+ D: A large scale dataset for 3D human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1010–1019.

Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. 2017. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 4263–4270.

Ákos Utasi and Csaba Benedek. 2011. A 3-D marked point process model for multi-view people detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3385–3392.

Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014a. Human action recognition by representing 3D skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 588–595.

Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014b. Human action recognition by representing 3D skeletons as points in a lie group. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. 588–595.

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

Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. 2014. Learning actionlet ensemble for 3D human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 5 (2014), 914–927.

Li Wei and Eamonn Keogh. 2006. Semi-supervised time series classification. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 748–753.

Xiaopeng Xi, Eamonn Keogh, Christian Shelton, Li Wei, and Chotirat Ann Ratanamahatana 2006. Fast time series classification using numerosity reduction. In *Proceedings of the International Conference on Machine Learning*. 1033–1040.

Lu Xia and J. K. Aggarwal. 2013. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2834–2841.

Lu Xia, Chia-Chih Chen, and J. K. Aggarwal. 2012. View invariant human action recognition using histograms of 3d joints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 20–27.

Zhengzheng Xing, Jian Pei, Philip Yu, and Ke Wang. 2011. Extracting interpretable features for early classification on time series. In *Proceedings of the SIAM International Conference on Data Mining*. 247–258.

Zhengzheng Xing, Jian Pei, and Philip S. Yu. 2009. Early prediction on time series: A nearest neighbor approach. In *International Joint Conference on Artificial Intelligence*. 1297–1302.

Lexiang Ye and Eamonn Keogh. 2009. Time series shapelets: A new primitive for data mining. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 947–956.

Mihai Zanfir, Marius Leordeanu, and Cristian Sminchisescu. 2013. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 2752–2759.

Chenyang Zhang and Yingli Tian. 2015. Histogram of 3d facets: A depth descriptor for human action and hand gesture recognition. *Computer Vision and Image Understanding* 139 (2015), 29–39.

Zhang Zhang, Jun Cheng, Jun Li, Wei Bian, and Dacheng Tao. 2012. Segment-based features for time series classification. *Computer Journal* 55, 9 (2012), 1088–1102.

Received January 2017; revised June 2017; accepted August 2017