Linear Latent Low Dimensional Space for Online Early Action Recognition and Prediction

Victoria Bloom, Vasileios Argyriou, Dimitrios Makris

PII: S0031-3203(17)30264-9
DOI: 10.1016/j.patcog.2017.07.003
Reference: PR 6206

To appear in: *Pattern Recognition*

Received date: 1 February 2017
Revised date: 8 June 2017
Accepted date: 2 July 2017

Please cite this article as: Victoria Bloom, Vasileios Argyriou, Dimitrios Makris, Linear Latent Low Dimensional Space for Online Early Action Recognition and Prediction, *Pattern Recognition* (2017), doi: 10.1016/j.patcog.2017.07.003

Linear Latent Low Dimensional Space for Online Early Action Recognition and Prediction

Victoria Bloom[1], Vasileios Argyriou[2*], Dimitrios Makris[2]

[1] *Coventry University, United Kingdom,* [2] *Digital Imaging Research Centre, Kingston University, United Kingdom*

Abstract

Recognition and prediction of human actions is one of the important tasks in various computer vision applications including video surveillance, human computer interaction and home entertainment that require online and real time approaches. In this work we propose a novel approach that utilizes continuous streams of joint motion data for recognizing and predicting actions in linear latent spaces operating online and in real time. Our approach is based on supervised learning and dimensionality reduction techniques that allow the representation of high dimensional nonlinear actions to linear latent low dimensional spaces. Our methodology has been evaluated using well-known datasets and performance metrics specifically designed for online and real time action recognition and prediction. We demonstrate the performance of the proposed approach in a comparative study showing high accuracy and low latency.

Keywords: Action recognition, action prediction, dimensionality reduction.

1. Introduction

The research field of human action recognition has rapidly expanded in recent years with many innovative applications in a range of sectors including healthcare, education, robotics and entertainment [1] . In healthcare, action recognition enables touch-free browsing of medical images in operating rooms, physical therapy at home and in clinics and patient monitoring. In education, action recognition can increase the engagement of users by providing realistic

and immersive training simulations. In robotics, action recognition facilitates natural interaction between humans and robots. In entertainment, action recognition enables touch-free interaction with smart TVs and games consoles for more intuitive and natural interaction. A key requirement of these interactive applications is the ability to robustly detect actions in real-time so the system can provide an appropriate response to the user with no apparent delay.

Action recognition can be categorised into four distinctive approaches: offline, online, early and prediction, as illustrated in Figure 1. Until recently, the vast majority of action recognition research focused on offline methods using pre-segmented action sequences containing a single action and information from all the frames to classify the action [2, 3, 4, 5, 6, 7, 8]. The action was recognised after its completion and the computation time was unrestricted. Similarly, early action recognition is typically performed on pre-segmented sequences but using as few observations as possible from the start of the sequence [9, 10, 11, 12, 13, 14]. These simplifications resulted in over-inflated accuracy and action recognition algorithms unsuitable for real-world applications.

In contrast, recent research has pursued the more complex challenge of online action recognition that processes a continuous stream of actions in real-time [15, 16], however the accepted latency of recognition can vary depending on the application. For example, a sign language recognition system may delay recognition until a sequence of words has been parsed [17], and therefore can benefit from increased accuracy by delaying the recognition. However, other systems require low latency and in may be benefited by early detecting the action even before its completion. Our research in this paper focuses on gaming applications where low latency is essential for a smooth user experience.

Action prediction is the most recent development in human action recognition and involves forecasting future occurrences based on recent observations. Prediction on a continuous stream with temporal localisation of the action peak before it occurs is a very challenging scenario. Action peak is defined as the segment in time when the goal of the action is being satisfied. Action prediction is a very difficult problem for machines but is naturally performed by humans

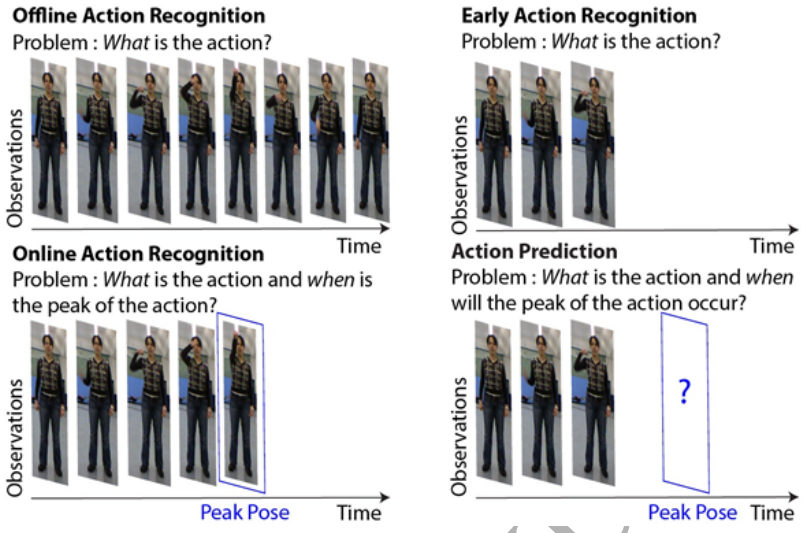

Figure 1: Observations required for offline, early, online action recognition and predictio.

to coordinate their actions in time and space to accomplish their goals. Experimental results in human-human interaction in a table tennis game showed that action prediction improves performance [18, 19, 20]. Action prediction can enhance many applications with a human-machine interface in a range of domains including home entertainment, healthcare, sports, and robotics. For example, a personal robotic assistant for the elderly can enable independent living by assisting with a range of cognitive and physical tasks to improve their quality of life. Natural social interaction between the robot and patient is important for acceptance of the robot in the patients home and can also provide vital social contact for the patient [21].

This paper aims to deal with the tasks of early action recognition and action prediction in continuous streams by introducing a novel linear latent low dimensional space based on Clustered Spatio-Temporal Manifolds (CSTM). Such challenging tasks are tackled because of two main characteristics of our methodology, the execution-rate invariance and stylistic invariance, that are achieved because of the unique combination of the Temporal Laplacian Eigenmaps (TLE) [2, 4] and k-means that allow the definition of the CSTM linear latent space.

Another important aspect is the novel action templates that provide the ability to follow the progression of an action over time in a linear manner. Furthermore, these action templates are combined with the proposed Peak Key Poses enabling online action recognition with high accuracy and low latency, which are particularly useful in applications such as gaming and HCI, where timely and precise action detection is required.

2. Literature Review

Feature selection for offline action recognition is an extremely well researched and established topic with a vast number of publications so this review begins by focusing on approaches with low computational latency that may be adapted for online action recognition. Then, a review of the more recent research into early, online action recognition and prediction approaches is provided. A more detailed analysis is provided in [1] by B. Liang and L. Zheng.

2.1. Feature Selection

Dimensionality reduction techniques have been used in conjunction with machine learning algorithms to reduce the number of considered features to improve computation time, reduce memory requirements and even improve accuracy. There are many different dimensionality reduction techniques that can be divided into feature selection and feature transformation.

Feature selection methods choose a subset of important features whereas feature transformation methods form new features, that are fewer in number than the original and are divided into filters, wrappers and embedded methods [22].Filter methods select subsets of variables by ranking individual variables with scoring functions such as correlation coefficient or mutual information criterion. Their benefits are their simplicity and computational efficiency, but may lack in performance. Wrapper methods use the prediction performance of a given classifier to assess the relative usefulness of subsets of features. In pose-based action recognition genetic algorithms have been used to determine the optimum set of skeleton joints which improved recognition rates [23].

Embedded methods incorporate variable selection in the process of training and can be more efficient than wrapper methods. Decision trees [24] and Random Forests [25] contain a built-in mechanism to perform variable selection that can estimate the importance of each feature during the classification process. Random Forests were employed by Negin et al. [26] as a discriminative feature selection tool to improve the action recognition performance of a Support Vector Machine (SVM) with a small fraction of the original pose-based features. Negin et al. [26] used features extracted from the entire sequence which has high observational latency.

2.1.1. Feature Transformation

The aim of feature transformation is to map the original high dimensional feature space to a much lower dimension, resulting in fewer features that are a combination of the original features. The advantage of feature transformation is that it handles the situation in which multiple features collectively provide good discrimination even if they provide relatively poor discrimination individually.

Schwarz et al. [27] use Laplacian Eigenmaps (LE) to suppress individual style. LE considers the spatial relationships between poses, but ignores the temporal relationships which are critical for recognising similar actions. This limitation has been overcome by spatio-temporal action manifolds [2, 3, 4, 5]. Lewandowski et al. [2, 4] proposed Temporal Laplacian Eigenmaps (TLE) that extend LE by preserving the temporal structure and suppressing the stylistic variations of the data in the low dimensional space. Gong and Medioni [3] proposed a directed traversing path on a spatial manifold to incorporate the temporal dimension. They proposed Dynamic Manifold Warping for temporal alignment followed by spatial similarity of sequences on their manifold. Vemulapalli et al. [5] proposed a new representation of skeleton data as a Lie Group which is a 6D curved manifold. Human actions were modelled as curves on this manifold. DTW was used for execution rate invariance and additionally Fourier Temporal Pyramids to handle noise. The final classification was performed with linear SVM and achieved state-of-the-art results for offline action recognition.

The spatial-temporal manifolds [2, 3, 4, 5] are invariant to personal style and execution rate invariant but as the whole sequence is used for classification the action recognition has high observational latency and requires the action to be pre-segmented.

Deep learning approaches have also been proposed for action recognition [6, 7, 8]. In these architectures, early layers learn features from unlabelled video data, in contrast to selecting hand-crafted features, while later layers may perform feature transformation and finally action recognition. The benefit of deep learning is that the features can be automatically selected without the use of prior knowledge and they have achieved comparable or even better accuracy than engineered features for offline action recognition. Nevertheless, deep learning approaches require large amounts of training data, which may not always be available.

2.2. Early action recognition

Early action recognition aims to determine the action class based on as few observations as possible, even when only part of the action has been seen. Existing early activity recognition approaches extend popular activity recognition methods such as bag-of-words (BoW) [9, 10], sequential state models [11, 12] and maximum margin methods [28, 13, 14].

Ryoo [9] proposed two extensions to the bag-of-words paradigm for early activity recognition: Integral bag-of-words (Integral BoW) and Dynamic bag-of-words (Dynamic BoW). The integral histogram models spatial changes in the visual words but the temporal relations are ignored. Dynamic BoW overcomes this limitation by splitting an activity into subsequences and using a sequential matching algorithm. Dynamic BoW outperforms Integral BoW which highlights the importance of temporal modelling for early recognition. Both approaches determined accuracy on sequences that were manually pre-segmented to contain a single action where results were calculated after observing ratios from 0.1 to 1.0, where 0.5 represents half the action and 1.0 the full action. Dynamic BoW achieves reasonable accuracy when half the activity has been observed. However,

the accuracy of both approaches is significantly reduced in the early part of the activity. Similarly, Cao et al. [10] use the bag-of-visual-words technique on video segments to incorporate local spatio-temporal features. Each video is uniformly divided into equal length segments and a mixture of segments of varied length and temporal shifts is used to improve execution rate invariance. However, this approach is limited to the number of scales and shifts that can be computed. Recently, Escalante et al [15] proposed a Naive Bayes classifier that accumulates evidence provided by bag-of-features from the beginning of a gesture/action, to achieve early recognition, even on continuous streaming data.

Sequential state models [11, 12] are effective at early recognition as they intrinsically preserve temporal order. Davis and Tyagi [11] proposed a Hidden Markov Model (HMM) for rapid and reliable early action recognition on manually pre-segmented sequences. Li and Fu [12] propose ARMA-HMM, an integrated autoregressive moving-average model (ARMA) with a HMM for early activity recognition on pre-segmented sequences. ARMA-HMM predicts future poses to enrich the partially observed activity sequences and improve early recognition. However, the reliance on manual pre-segmentation which has to be performed offline, negates the benefit of the early detection of these approaches.

Lan et al. [13] developed a max-margin framework for early action recognition that achieves state-of-the-art results when half the action has been observed in a manually pre-segmented sequence but the accuracy is significantly reduced in the early part of the activity. Kong et al. [14] extend the max-margin approach to multiple temporal scales and achieve state-of-the-art results when the full action has been observed which is equivalent to the classic offline action recognition problem but accuracy is lower than Lan et al. [13] when observing half of the action.

Hoai and De la Torre [28] proposed max-margin early event detectors (MMED) for early detection of a range of human activities i.e. facial expressions, gestures and actions. They extended Structured Output SVM to accommodate sequential data. Their learning formulation is a constrained quadratic optimisation problem to ensure monotonicity of the detection of partial activities. To evaluate

their approach Hoai and De la Torre [28] concatenated manually pre-segmented sequences to form longer sequences containing multiple actions to temporally detect the action as soon as possible which is an improvement over the previous scenarios in this section of single action evaluation. However, they considered each action individually by placing the action of interest at the end of the sequence and lowering the false positive rate until it reached 0% to ensure their algorithm did not detect the action of interest before it started. Therefore, MMED could not perform in a real-world scenarios of detecting multiple actions in a continuous stream. To address these issues, Huang et al [16] extended the previous work by proposing the Sequential Max-Margin Event Detectors (SSMED) which are based on multi-class classification and were evaluated on the newly-introduced CMU-MAD dataset to confirm their applicability on a continuous stream

The majority of existing approaches [9, 10, 11, 12, 13, 14] for early activity recognition focus on classifying the action as soon as possible using pre-segmented sequences. These approaches achieve reasonable accuracy after observing half the action but manual pre-segmentation simplifies the task of early detection which inflates accuracy and limits the applicability of these approaches to real-world scenarios. Compared to the other few methods that may work on continuous streams [15, 16], the linear latent space of our methodology explicitly models the overall temporal structure of each action and deals with issues such as stylistic and execution rate variation.

2.3. Online Action Recognition

Most of the existing online action recognition algorithms do not have low observational latency which is needed to ensure that the developed algorithms are suitable for real-time applications. There are two distinct approaches to address observational latency: the first is automatic action segmentation of the sequence followed by classification of the individual actions and the second is to perform continuous classification.

Automatic action segmentation is a natural progression to enable existing

offline recognition approaches to be used online. De la Torre et al. [29] use a clustering algorithm to cut sequences into action instances. However, their segmentation algorithm is processed offline so subsequent action recognition would also be offline. To overcome this limitation Gong et al. [30] fused the segmentation with matching. However, as the segmentation is based on capturing transitions between actions, the recognition can only occur after the action is complete incurring high observational latency, because of the potential difference between peak time and completion time.

An alternative approach for online action recognition with very low latency is to reduce template matching to single pose matching. Ellis et al. [11] automatically reduce the number of key poses to a single canonical pose for each action. The disadvantage of such an approach is that no temporal history of an action is used, and as a consequence matching of just a single pose may lead to false detections especially when different actions contain similar poses.

Eickeler et al. [31] proposed two methods based on HMM for continuous recognition of gestures: smoothing and filtering. The former approach achieved high accuracy but with high observational latency (12 seconds) which may be acceptable in some applications e.g. sign language recognition but not suitable for human-computer interaction. The latter approach reduced the time delay of recognition but only if the gestures were temporally isolated which limits its suitability for gaming scenarios. Natarajan and Nevatia [32] proposed a hierarchical HMM with variable size sliding temporal window to achieve high accuracy at low observational latency (average 3.2 frames) and real-time computation (28.6fps) for online action recognition. Although, this method allows continuous action recognition the method requires prior knowledge of the structure of the actions, like the limbs involved.

To precisely measure latency Nowozin and Shotton [33] introduced action points, a temporal anchor for action instances within a sequence. For example, an action point for a punch could be defined as the moment at when the arm is maximally extended. They also proposed two recognition models that can detect action points in real time. Their first approach, Firing Hidden Markov

Model [33] is a variation of HMM with an explicit firing state which detects action points when the probability of the action exceeds a threshold. In their experiments they compared offline smoothing with online filtering.

Nowozin and Shotton second approach, online Random Forests [25] was adapted for continuous action recognition using a sliding window approach. Experiments showed that Random Forest was simpler, faster and more reliable than the HMM approach [33, 34]. Similarly, Bloom et al. [35] used a sliding window and performed the classification by AdaBoost. However, the fixed size of the sliding window in these approaches is a source of error due to execution rate variations. To address this Zhao et al. [29] optimised the size of the segment during their pre-processing using a DTW variant for subsequence matching. However, as the average length of their templates is 35 frames observational latency is high. Sharaf et al. [36] achieved state-of-the-art results for online action recognition with a feature selection approach combined with a SVM. Sharaf et al. used features at multi-scales to improve execution rate invariance but their approach is computationally limited to a couple of levels which limits the execution rate invariance.

Recently, Gees et al [37] introduced the TVseries dataset and a relevant evaluation framework for the evaluation of online action detection and early action recognition. However, their evaluation protocol is not appropriate for applications where time-precise detection of the action peak is required, e.g. in gaming and HCI applications.

2.4. Action prediction

Action prediction is a recent development in human action recognition, which has received relatively little attention and is also the most difficult task as it involves forecasting future occurrences based on recent observations.

Sequential state models [12, 38] are able to predict future poses as they intrinsically preserve temporal order. Li and Fu [12] proposed ARMA-HMM, which predicts future poses to enrich the partially observed activity sequences. The focus of their work was to improve early recognition so the accuracy of the

predicted poses was not evaluated. Also, Galata et al. [38] proposed variable-length Markov models (VLMM) to encode high-order temporal dependencies for animation of human activities. They synthesised hypothetical activity sequences using the VLMM as a stochastic generator to create realistic animations with statistically accurate variations. However, the aim of their work was to generate synthetic poses rather than predict actual future poses.

Vondrick et al. [39] demonstrated the difficulty of predicting actions by demonstrating that human subjects also fail to accurately predict actions in 30% of the cases when given a single frame one second before the action starts. To handle this ambiguity they develop a deep network architecture to produce multiple predictions and use large amounts of unlabelled video data to capture common sense knowledge about the world. Although they are still far from human performance on this task they are able to achieve reasonable accuracy for such a complex task. However, further analysis of their training frames shows that the start of an action is also an ambiguous concept as some examples do contain pose information that reveal the intended action and others contain contextual information that may be used to determine the action.

There is relatively little research into action prediction and the approaches vary widely in their goals, ranging from improving early action recognition, through generating synthetic sequences to predicting the action class before the action starts. The last is the most interesting and challenging especially in scenarios where there is no contextual information. In the best of our knowledge, our work is the first ever that deals with the problem of action prediction in continuous streams.

3. Methodology

The core of our proposed methodology is the Clustered Spatio-Temporal Manifolds, which are compact style invariant models of the dynamics of human actions. They enable action classification in a continuous stream for early action detection in addition to the ability to follow the progress of the action so that

the peak can be detected with low latency or even predicted. Three inference algorithms are also proposed to enable early action recognition, online action recognition and action prediction.

The spatio-temporal manifolds are created by feature transformation to reduce style variance whilst still maintaining the temporal dynamics of the action. The main contribution of this paper is the combination of k-means and TLE, that extracts style-invariant key-poses ordered along the TLE manifold, so to define a linear latent low dimensional space for each action.

Action templates defined along the by linear latent spaces are effectively matched using DTW for execution rate invariance. Our second contribution is to reduce the high observational latency of template matching by employing a sliding window approach to match template fragments with low latency. Peak key poses are the third contribution to enable explicit location of action peak for low latency action recognition and even action prediction.

Latency is dependent on two separate factors which have been identified as observational latency and computational latency [40]. Observational latency is the time it takes the system to observe enough frames to make a decision, whereas computational latency is the actual time to perform the computation on a frame. Ellis et al. [40] measured observational latency from a rest state which is not possible feasible with in multiple action scenarios as the subjects may not return to the rest state between actions. Therefore, in this paper observation latency is defined as the time after the peak of the action at which the action is detected which at any rate is a more suitable measurement for evaluating latency for natural user interface (NUI) applications.

The proposed methodology consist of the same training phase (section 3.1) which generates the action templates and an inference phase that depends on the specific task: early action recognition (section 3.2.1), online action recognition (section 3.2.2) and prediction (section 3.2.3).

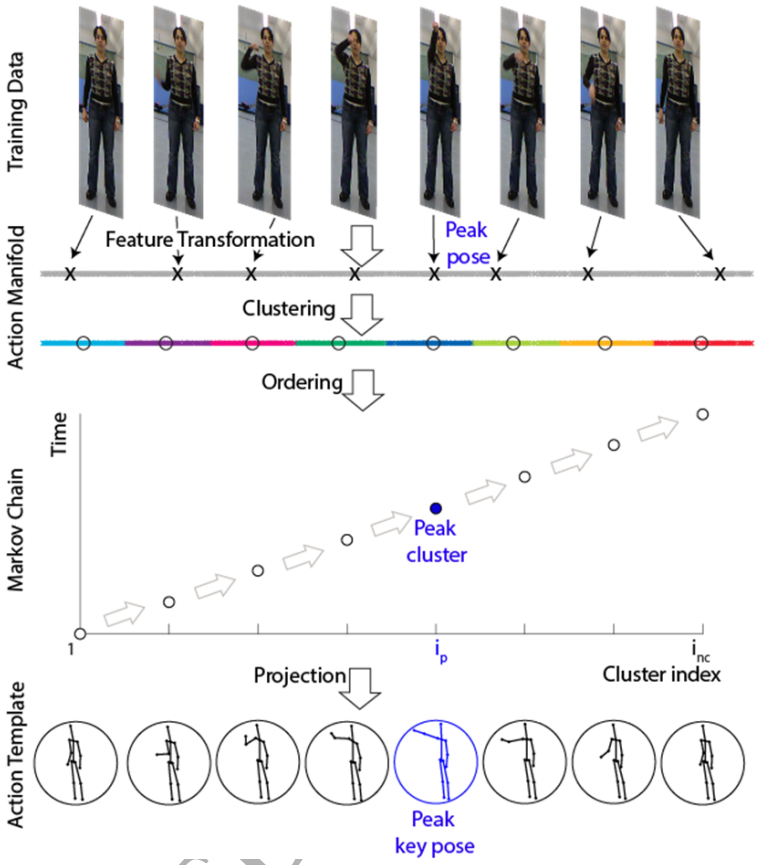

Figure 2: Action templates with four key stages: dimensionality reduction, clustering, ordering and projection.

3.1. Action Model Learning

To create the spatio-temporal action templates, there are four key stages: feature transformation, clustering, ordering and projection (as shown in Figure 2). Human actions are represented by a large number of spatio-temporal features, so the first stage is to reduce the dimensionality. Temporal dynamics are critical for action recognition and prediction so a dimensionality reduction method that preserves the temporal structure of the data in the embedded space is employed. Temporal Laplacian Eigenmaps (TLE) [2, 4] is a nonlinear feature transformation technique, that finds a new set of dimensions that are combina-

tions of the original dimensions. TLE has previously been used for offline action recognition from video sequences [2] and is suited to any time series data that contains repetitions of actions.

Pose-based features can be viewpoint and anthropometric invariant as well as generated in real-time with a pose estimation method [41]. Normalising the skeleton poses and obtaining the joint angles removes the viewpoint variations. Although the proposed algorithm is evaluated with skeleton data, the method can also be applied to other time series data. Similar to Lewandowski et al. [2] the joint angle features are defined as the quaternions of the angle between three connected joints in a single pose (e.g. right wrist, wright elbow and right shoulder) were calculated for 13 joint angles for each skeleton pose, so each high dimensional feature vector has 52 dimensions. The quaternions $f^q \in \mathbf{C}^4$ were built in the standard polar (axis-angle) form:

$$f^q = \cos(\frac{\theta}{2}) + \sin(\frac{\theta}{2})(in_x + jn_y + kn_z) \tag{1}$$

where n is the (unit length) axis of rotation, θ is the angle, and i, j and k are the imaginary basis vectors.

3.1.1. Dimensionality reduction

Temporal Laplacian Eigenmaps (TLE) algorithm [2, 4] is an unsupervised nonlinear method for dimensionality reduction for time series data. Given a set of points $\mathbf{X} = (\mathbf{x}_{i_r})_{(i_r=1\ldots n_r)}$ distributed in high dimensional space ($\mathbf{x}_{i_r} \in \mathbf{R}^D$), TLE is able to discover their low dimensional representation $\mathbf{Y} = (\mathbf{y}_{i_r})_{(i_r=1\ldots n_r)}$, ($\mathbf{y}_{i_r} \in \mathbf{R}^d$) where $d \ll D$ and n_r is the number of points in the time series, as shown in Figure 4. The key feature of the embedded manifolds is that the temporal structure of the data is preserved in the low dimensional space.

Two neighbourhood graphs are constructed during the process of dimensionality reduction, one with adjacent temporal neighbours and another with geometrically similar neighbours, as illustrated in Figure 3. The adjacent temporal neighbours are the $2n_u$ closest points in the sequential order and repetition neighbours are the points similar to x_{i_r}, extracted from repetitions of time series

fragments.

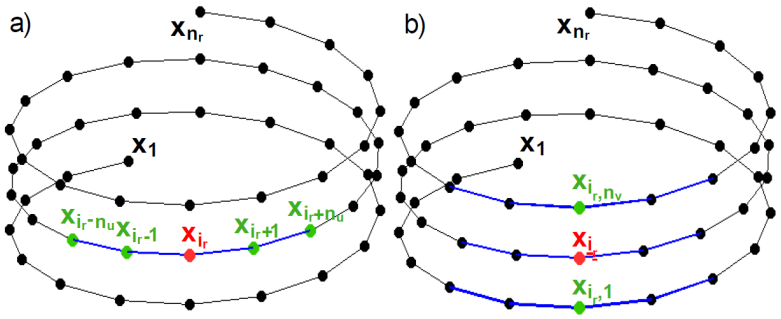

Figure 3: TLE: temporal neighbours (green dots) of a given data of a given data point, x_{i_r}, (red dots) in a) adjacent and b) repetition graphs.

Neighbourhood connections defined in the Laplacian graphs place neighbours from the high dimensional space nearby in the embedded space. Consequently, the temporal neighbours preserve the temporal structure and the spatial neighbours reduce style variability by aligning the time series in the embedded space.

3.1.2. Clustering

Clustering is then performed on the embedded manifold to remove redundant information. k-means [42] is applied to cluster the n_r low dimensional points Y into n_c clusters $\mathbf{C} = \{\mathbf{c}_{i_c}\}_{(i_c=1\ldots n_c)}$, $\mathbf{c}_{i_c} \in \mathbf{R}^d$, where $n_c \ll n_r$ as shown in Figure 4. Removing redundant information reduces the computational time of the subsequent action recognition and may also improve accuracy. Additionally, the clusters provide key points throughout an actions lifecycle that can be used to determine the current and even predict future progress. The number of clusters ($n_c = 35$) was set based on existing experiments for offline action recognition [2].

3.1.3. Ordering

The clusters discovered by k-means are unordered so the temporal relationships from the embedded manifold are exploited to order the clusters. A first-order Markov chain [43] is constructed for each action to chronologically

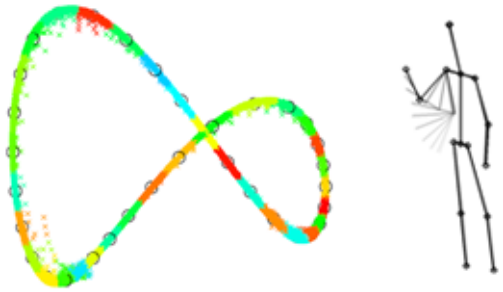

Figure 4: Clustered Spatio-Temporal manifold with the low dimensional points Y shown as points, coloured according to their cluster and the cluster centers C as black circles. Each on the clusters correspond to a key pose in the high dimensional space.

link the clusters. The Markov chain is defined by the transition matrix $\Lambda = (\lambda_{i_c,j_c})_{(i_c=1\ldots n_c, j_c=1\ldots n_c)}$ where $\lambda_{i_c j_c}$ are the cluster transition probabilities. The transition probability from cluster i_c to cluster j_c is found by counting connections between temporal neighbours on the manifold. If transitions to the same cluster are ignored, the maximum transition probability for each cluster will represent the temporal order $\mathbf{o} = (o_{i_c})_{(i_c=1\ldots n_c)}$ between the clusters as shown in Figure 5 and in Eq. 2, where $i_c \neq j_c$.

$$o_{i_c} = \arg\max_{j_c}(\lambda_{i_c j_c}) \tag{2}$$

Since the clusters are determined by k-means, their centres tend to be equally-distant along the temporal structure specified by TLE and therefore define a low dimensional linear latent space. This latent space extends the applicability of TLE from offline action recognition [2, 4] to the challenging tasks of online action recognition, early action recognition and action predictions that are tackled in this work.

3.1.4. Projection

Selecting key poses removes redundant information to improve classification accuracy and reduce the computational latency of template matching. In relevant works, key poses were estimated by identifying the most discriminative

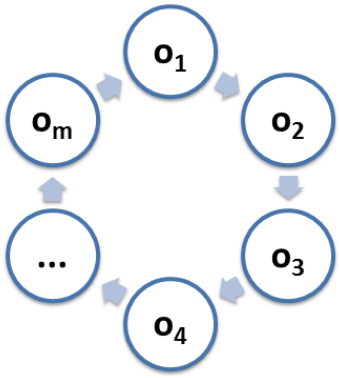

Figure 5: Cyclic clustered action manifold and highest probability transitions.

frames according to the entropy of their visual words [44] or using Adaboost [45], or by k-means clustering of training poses and selecting the closest pose to the cluster centre [46]. Ponce et al [47] also applied k-means but on aligned subsequences to identify discriminative subgestures instead. The above methods reduce stylistic variation by selecting an average pose but as the key pose represents an individual some personal style will remain. To eliminate personal style the proposed method uses the clusters from the low dimensional action manifolds and projects their centres to the high dimensional space, using the Radial Basis Function Network (RBFN) mapping to generate new poses that are not present in the training dataset.

One limitation of TLE is that it places the n_r points in a low-dimensional space but it does not learn general mapping functions that will allow new points to be projected from the low to the high dimensional space. RBFN mapping functions allow projecting new data between the low and high dimensional spaces [2]. Using $\chi = \{\mathbf{y}_{i_r}, \mathbf{x}_{i_r}\}_{((i=i_r \ldots n_r)}$ as a training set, RBFN are trained to learn the mapping between the low and the high dimensional space [2]. Then using the RBFN mappings the cluster centres $\mathbf{C}$ are projected into the high dimensional space to generate key poses $k \in \mathbf{R}^D$, that form the action templates $\mathbf{K}^a = (\mathbf{k}_{i_o})(i_o = o_1 \ldots o_{n_c})$, by using the temporal order $\mathbf{o}$ found between clusters, as illustrated in Figure 6.

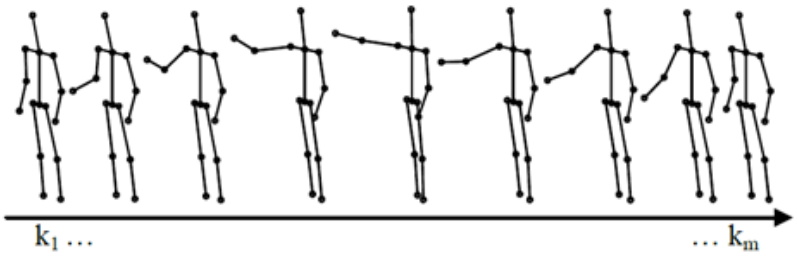

Figure 6: Cyclic clustered action manifold and highest probability transitions.

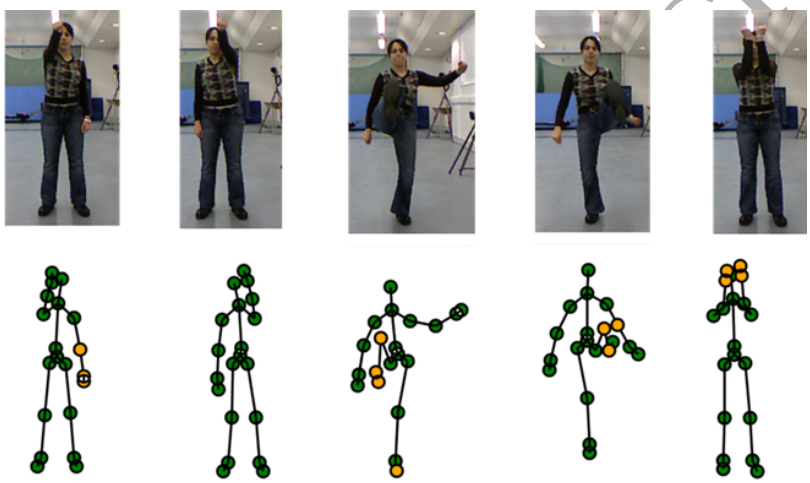

Figure 7: Peak poses for different actions, from left to right: right punch, left punch, right kick, left kick and defend.

3.1.5. Peak key pose selection

The peak of an action is a key concept, which is defined as the moment when the goal of the action is satisfied. For example, in a boxing game the aim of punching is to hit the opponent which is fulfilled when the arm is maximally extended. The poses in the dataset that fulfil the action goal are manually labelled as peak poses with one peak pose labelled for each action instance. Examples of peak poses for different actions are illustrated in Figure 7.

Key poses have been used with template matching for offline action classification [46] but the novel contribution is to select the key pose that represents

the peak of the action for online classification. Peak key poses are a novel concept, which are related to but are not the same as action points [33] or canonical poses [40]. Peak key poses also represent a single pose, but in contrast to existing approaches, they are selected from the key poses rather than the training poses, so they are invariant to individual style.

To select the peak key poses, the peak poses from the training data (shown in Figure 7) are matched against the key pose templates (shown in in Figure 6). To increase robustness, fragments of poses are matched rather than single poses which enables actions with similar poses to be correctly matched based on the temporal pose history before the action peak. To extract a fragment f^G from a sequence of poses $\mathbf{S} = (\mathbf{s}_{i_s})_{(i_s=1\ldots n_s)}$, $(\mathbf{s}_{i_s} \in \mathbf{R}^D)$, Eq. 3 is used, where n_f is the required number of poses in the fragment, i_f is the index of the last pose, n_s is the number of poses in the sequence and $i_f \leq n_s$ and $i_f - n_f \geq 0$.

$$f^G(\mathbf{S}, i_f) = (\mathbf{s}_{j_f})_{(j_f=i_f-n_f, i_f-n_f+1,\ldots,i_f)} \tag{3}$$

Assuming the peak poses in the training data have been manually selected for each action and their indices stored: $\eta = (\eta_{i_\eta})_{(i_\eta=1\ldots n_\eta)}$, the peak key poses are selected as follows: for each action a and for each peak pose index η_{i_η}, the matching key pose index i_m is found by minimising the DTW distance between the peak pose fragment from the training poses $\mathbf{X}$ and the key pose fragments from the action templates $\mathbf{K}^a$, as in Eq. 4.

$$i_m(\eta_{i_\eta}) = \arg \min_{i_k \in 1\ldots n_c} f^D(f^G(\mathbf{X}, \eta_{i_\eta}), f^G(\mathbf{K}^a, i_k)) \tag{4}$$

To find the peak key pose index i_p for the action a, ζ is initialised ($\zeta = 0_{1,n_c}$) and each time a matching key pose index i_m is found ζ_{i_m} is incremented. The peak key pose index i_p for the action is the key pose index, with the maximum number of matches ($i_p(a) = (\arg\max \zeta)$.

3.2. Action Recognition

Three action recognition methods are introduced below, online, early action recognition and action prediction, and all have a common base of online

template matching with DTW for execution rate invariance [48]. Existing approaches for offline action recognition use the entire action template which inherently has high latency [46]. To enable online recognition a sliding window approach matches recent test poses with action template fragments, as illustrated in Figure 8.

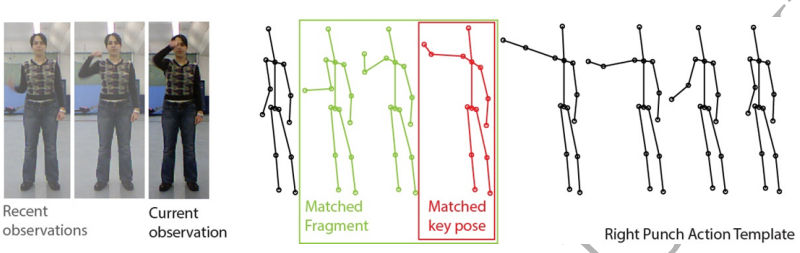

Figure 8: Template fragment matching: observed test poses and matched action template.

3.2.1. Early Action Recognition

Early action recognition aims to determine the action class, based on as few observations as possible, even when only part of the action has been seen. In most of existing work [9, 10, 11, 12, 13, 14] the sequences are pre-segmented to contain a single activity and evaluation is performed at different observation ratios, from 0.1 to 1. So an observation ratio of 0.5 represents the first half of the action and an observation of 1 is the conventional offline action recognition approach. Since the test sequences in this work are not pre-segmented, as they consider the real-time application of action recognition, the proposed method assigns an action label for each frame in a continuous stream using a sliding window. The sliding window contains the recent and current observations from the test stream to ensure no future information is incorporated into the method.

The proposed method for early action recognition is online template matching where the current test pose fragment is matched against sliding windows on each of the different action templates to obtain key pose fragments. The action class of the most similar key pose fragment is used as the action classification label for the current frame. DTW allows "elastic" transformation so actions in the test stream performed at different speeds to the action templates

can be matched. Formally, early action recognition for each sequence of test poses $\mathbf{Z} = (\mathbf{z}_{i_t})_{(i_t=1...n_t)}$, $z_{i_t} \in R^D$ is performed as follows: to find the action classification label a' for the current pose $\mathbf{z}_{i_t}$, the normalized DTW distance between the test pose fragment and test poses from all the action templates are minimised according to:

$$a^*(i_t) = \arg \min_{a \in 1...A} (\min_{i_k \in n_f...n_c} f^D(f^G(\mathbf{Z}, i_t), f^G(\mathbf{K}^a, i_k))) \quad (5)$$

The minimum normalised DTW distances for each frame of a sample sequence in the G3D dataset [49] against each action template are shown in Figure 9. The lowest distance over all the actions represents the matched action class as illustrated in Figure 9.

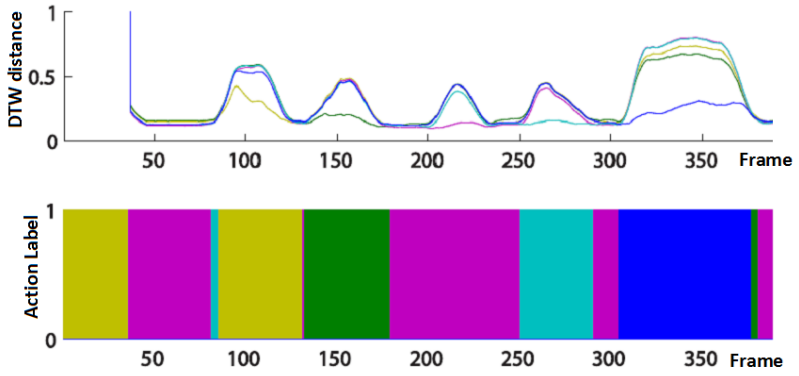

Figure 9: (Top) Normalised DTW distance for each frame (Bottom) Action classification label for each frame. At this stage all frames are classified as an action, even the neutral frames. To overcome this limitation action points are detected at the next stage to only classify the peak frame of each action.

3.2.2. Online Action Recognition

To enable continuous action recognition to be suitable for real-world applications a single point needs to be identified for each action, rather than classifying individual frames. For this reason action points [33] were introduced which are action labels with temporal anchors. Action points are used in this section to

detect the peak of the action and each action point is represented by an action label a and a timestamp t_d.

Combining online template matching with peak key poses enables online action recognition with high accuracy and very low latency [50]. To explicitly locate the moment where an action reaches its peak, poses are followed as they progress through the early stages of the action and the peak is detected by comparing the matched poses with the peak key pose.

For each test pose stream $\mathbf{Z} = (\mathbf{z}_{i_t})_{(i_t=1...n_t)}$ online action recognition consist of three main steps: the first step is to find the action classification label a^* for the current test pose z_{i_t} using the online template matching described in section 3.2.1. The second step is to determine the progress of the current action by locating the key pose on the action template that is the closest match to the current test pose. To find the matching key pose index i_m for the current test pose index i_t, the normalised DTW distance for the test pose fragment against test poses from all the action templates are minimised according to Eq. 6.

$$i_m(i_t, a^*) = \arg \min_{i_k \in n_f ... n_c} f^D(f^G(\mathbf{Z}, i_t), f^G(\mathbf{K}^a, i_k)) \tag{6}$$

The third step is to determine if the action has reached its peak. The peak key pose can be conceptually projected onto the clustered action manifold to illustrate that the peak pose is detected when the matched key pose index i_m is the same as (or slightly greater) than the peak key pose index i_p (as shown in Figure 11) and is formally defined in Eq. 7.

$$\varphi(i_m, i_p, n_k) = \begin{cases} 1 \; if \; 0 \leq i_m - i_p \leq n_k \\ 0 \; otherwise \end{cases} \tag{7}$$

where i_m is the matched key pose index for the current test pose $\mathbf{z}_{i_t}$, i_p is the index of the peak key pose and n_k is the maximum number of poses after i_p allowed to detect a peak pose. This can also be illustrated in graph format as shown in Figure 10 where the key pose index i_k, is plotted for each frame and where this cluster index line crosses the peak key pose line (dotted horizontal line) for the corresponding action an action point is detected (o).

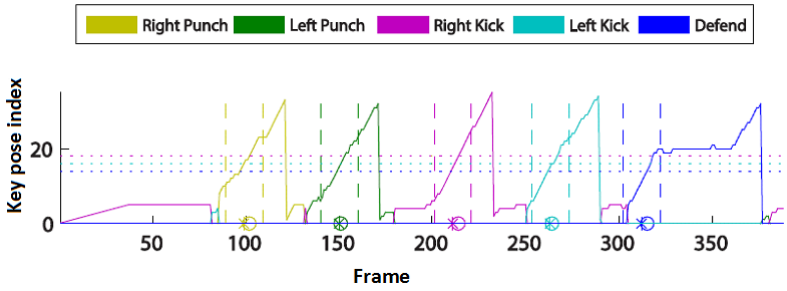

Figure 10: Clustered Action Manifold cluster indices for each frame with ground truth action points (*) and detected action points (o).

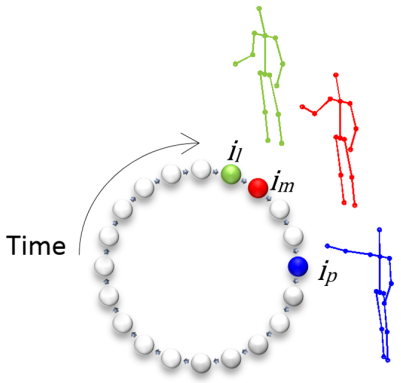

Figure 11: Right Punch Clustered Action Manifold with peak key pose index i_p with matched key pose index i_m and last matched key pose index i_l.

3.2.3. Action Prediction

There are relatively few approaches to action prediction and the approaches vary widely in their goals, ranging from improving early action recognition [12], through generating synthetic sequences [38] to predicting the action class before the action starts [39]. In this subsection a novel approach to action prediction is proposed where action peaks are predicted in a continuous stream before the peak has been observed. Action points are used in this section to represent the action peak and each prediction is represented by an action label a, a timestamp for the predicted action peak t_p to determine the timeliness of the prediction

and a timestamp at the time the prediction was made t_d to measure how far in advance the predictions can be accurately made.

For each test pose stream $\mathbf{Z} = (\mathbf{z}_{i_t})_{(i_t=1\ldots n_t)}$, online prediction consists of four main steps: the first step is to find the action classification label a^* for the current test pose $\mathbf{z}_{i_t}$ using the online template matching, using Eq. 5 described in section 3.2.1. The second step is to determine the progress of the current action by locating the key pose index i_m on the action template that is the closest match to the current test pose, using Eq. 6, described in section 3.2.2. The third step is to store the n_m most recent sequential pose matches of the current action class a' to maintain the history of the action progress $\theta = (i_m(\theta_t, a^*))_{(\theta_t=i_t-n_m\ldots i_t)}$.

The fourth step is to perform the action prediction using the recent action history and regression. Although the dynamics of human actions are nonlinear in the high dimensional space, our embedded clustered spatio-temporal representation establishes a linear latent space. This is demonstrated in Figure 10, which shows time along the horizontal axis and the key pose index along the vertical axis. Therefore, linear regression is proposed to quickly predict the action peak. For the current test pose z_{i_t}, when n_m sequential key pose matches of the same action class a' have been observed, their key pose indices θ, are fitted to a straight line by least-squares regression and the equation of the line is derived by Eq. 8.

$$(\alpha^{'}(\alpha^*, i_t), \beta^{'}(\alpha^*, i_t)) = \arg\min_{\alpha,\beta} \sum_{\theta_t=i_t-n_m}^{i_t} (i_m(\theta_t, \alpha^*) - \alpha - \beta t_i)^2 \qquad (8)$$

where $\alpha^{'}$ is the y-intercept of the least squares line and $\beta^{'}$ is the gradient.

The least squares line is extended to predict future poses using the derived equation. The peak key pose line is a horizontal line with a y-intercept of the peak key pose index i_p for the corresponding action. The point where the extended least squares line intersects the peak pose horizontal line is the estimated time t_p of the peak with time of detection $t_d = i_t$ (see Figure 12). Extreme cases are excluded by setting thresholds on the minimum and maximum

gradient of the slope. The gradient of the line represents the execution speed of the current test subject and is independent on the speed of subjects observed in the training set. Fast subjects will match key poses in the action template faster than slower subjects resulting in a steeper slope. A key benefit of the proposed temporal prediction is that it is invariant to execution speed as it utilises the gradient of the slope which is formed based on the speed of the current subject.

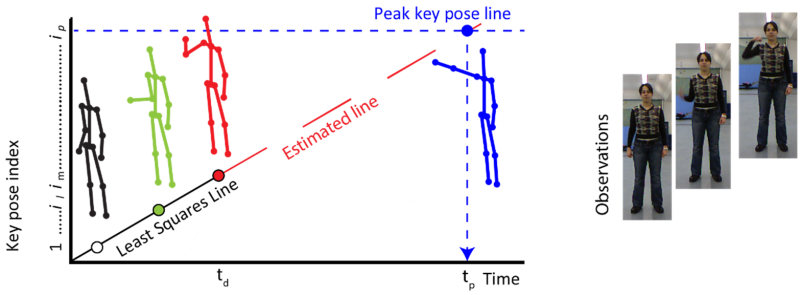

Figure 12: Linear regression at time t_d to predict the time t_p at which the partially observed action will reach its peak.

4. Experiments

4.1. Datasets

The performance of the proposed algorithms are evaluated using publicly available datasets designed specifically for real time action recognition: G3D [49] and MSRC-12 [34]. Both datasets provide sequences of skeleton data captured using the Kinect pose estimation pipeline at 30fps.

The MSRC-12 dataset comprises of 30 people performing 12 gestures. These gestures are categorised into two categories: iconic and metaphoric gestures. The iconic gestures directly correspond to real world actions and represent first person shooter (FPS) gaming actions. There are six FPS gaming actions: crouch, shoot, throw, night goggles, change weapon and kick. The dataset was obtained using different instruction modalities and the modality that produced the most accurate results was video + text.

Table 1: The total number of training and testing instances for gaming action datasets.

Dataset	Actions	Subjects	Repetitions	Cross Validation	Training Action Instances	Testing Action Instances
G3D	5	10	3	10	1350	150
MSRC-12	6	10	10	10	5400	600

The G3D dataset contains 10 subjects performing 20 gaming actions grouped into seven categories. The subjects are diverse in terms of gender, clothing and hair styles and in contrast to other action recognition datasets a G3D sequence contains different actions in the same sequence as shown in Figure 7. The fighting category was selected as it has substantial variations in execution rate as well as personal style. The fighting category contains five gaming actions: right punch, left punch, right kick, left kick and defend.

Action point annotations of the peak poses are available for the MSRC-12 dataset and G3D dataset to precisely measure the latency of action recognition methods as well as the accuracy. Comparative studies are conducted separately for performance in the specific tasks of online action recognition, early action recognition and action prediction.

A leave-person(s) out cross validation protocol was used where a set of people is removed to obtain the minimum test set that contains instances of all actions. For the MSRC-12 dataset this may be more than one actor as not every actor performs all the actions for the video + text modality. For the G3D dataset this is simply one actor as all actors perform all the actions. The remaining large set is used for the training. This process is repeated 10 times with different subsets of people to obtain the general performance. The total number of training and testing instances for each dataset used in the following experiments is shown in Table 1.

4.2. Online Action Recognition

4.2.1. Performance Metrics

For a fair comparison with existing approaches the same latency aware metric was used as initially proposed by [34] and later adopted by [36]. For a specified amount of latency (Δ) the action point F_1 score [33] determines whether a detection made at time t_{d_a} for action a is correct in relation to a ground truth action point at time t_{g_a} by using the following formula:

$$\Phi_a(t_{d_a}, t_{g_a}, \Delta) = \begin{cases} 1 \; if \; |t_{g_a} - t_{d_a}| \leq \Delta \\ 0 \; otherwise \end{cases} \tag{9}$$

For a specified amount of latency (Δ) precision p_r and recall r_e are measured for each action a and combined to calculate a single F_1-score.

$$F_1(a, \Delta) = 2\frac{p_{r_a}(\Delta) r_{e_a}(\Delta)}{p_{r_a}(\Delta) + r_{e_a}(\Delta)} \tag{10}$$

As online action recognition algorithms need to detect multiple actions, the mean F_1 score over all actions is used, defined as:

$$F_1(A, \Delta) = \frac{1}{|A|} \sum_{a \in A} F_1(a, \Delta) \tag{11}$$

The detected action points are compared to the ground truth action points using the action point metric to obtain a mean action point F_1-score at a fixed latency Δ, where $\Delta = 333ms$ the same as the studies by [36] and [34].

4.2.2. Comparative Study

Clustered Spatio-Temporal Manifolds that are proposed in this paper are evaluated against five algorithms: Random Forest [34], Dynamic Feature Selection [35], SVM-RFE [36] and our own implementations of Random Forests and AdaBoost that provide baselines for further experiments. For all the experiments the number of positive training samples selected around the action point was ± 8 and all other samples were used as negative training samples. The optimal positive sample size was found by varying this parameter between ± 1 and ± 20 on the training set.

- **Random Forests**: the 3 parameters that affect the performance of the Random Forest are the number of trees in the forest, the depth of each tree and the number of selected features at each node. Exhaustive searching of every combination of these 3 parameters is computationally prohibitive so in order to find the optimal forest configuration, 27 forests were trained with a combination of (10, 50 and 200) n_T trees, of depth (4, 6 and 8) with (10, 100 and 297) features selected at each node. Parameter selection was performed using cross validation on the training set. The best values of 200 trees, of depth 8 and 10 features at each node were found.

- **AdaBoost**: A comparison of Random Forests and AdaBoost in a different field [30] showed that AdaBoost can provide higher classification accuracy at the cost of less efficient computation. The standard version of AdaBoost is sensitive to noise in the dataset so Gentle AdaBoost [51] was selected as it gives less weight to outlier data points. As AdaBoost is also based on Decision Trees it has similar parameters: the number of weak classifiers which is the number of trees and the depth of the trees. Similarly, exhaustive searching is computationally prohibitive so in order to find the optimal configuration, 16 models were trained with a combination of (10, 50, 100 and 200) trees of depth (1, 3, 5 and 8). Parameter selection was performed using cross validation on the training set. The best values of 100 trees and depth 5 were found.

- **Clustered Spatio-Temporal Manifolds**: To learn manifolds for each action the algorithm requires manual segmentation of the start and end of the action and all frames are used for training. It is important to note that this segmentation is only required in the training phase and is not performed in the testing phase. The annotated action points are additionally used to learn the peak key poses. The parameters for the proposed approach are the target dimensionality d, the number of clusters n_c in the manifold, the fragment size n_f and the number of clusters n_k that can be skipped at the peak. The target dimensionality ($d = 3$), was

Table 2: Action Point F1-scores at $\Delta = 333ms$, the average and standard deviations over ten leave-persons-out runs are shown. The results shown in italics were published by the method authors, all other results were generated by our own implementations.

	Random Forest [34]	Random Forest	Ada Boost	Dynamic Feature Selection [35]	SVM-RFE [36]	Clustered Spatio-Temporal Manifolds
Feature Vector	Multi-frame	Single-frame	Single-frame	Single-frame	Multi-frame	Multi-frame
G3D	-	0.894 (0.155)	0.884 (0.147)	0.910 (0.128)	0.937	**0.978 (0.026)**
MSRC-12	0.765 (0.070)	0.619 (0.148)	0.675 (0.156)	0.744 (0.270)	-	**0.773 (0.124)**

determined by applying the maximum likelihood intrinsic dimensionality estimator [50]. The number of clusters ($n_c = 35$) was set based on existing experiments for offline action recognition [2]. The number of poses in the fragment ($n_f = 10$) was set to match the size of the smoothing window S in [35]. To find the value for n_k an exhaustive search was performed within the training set to maximise the F-score. The optimum value is ($n_k = 0$) for the MSRC-12 and ($n_k = 14$) for the G3D dataset. No smoothing window was applied to the frame based distance results, and the final output from the algorithm was the detected action points for each sequence.

4.2.3. Online Recognition Results

The experimental results show that the proposed Clustered Spatio-Temporal Manifolds achieved state-of-the-art accuracy for online action recognition with low latency. The experiments demonstrate the proposed method achieves the highest accuracy, 77.3% and 97.8% on the MSRC-12 and G3D datasets respectively (see Table 2 for a comparison with existing approaches). A breakdown of the results by action shows increased performance of the proposed method over the comparative methods in every action in the G3D dataset (see Figure 14). The graphs show the methods action point F_1-score for each action in

the dataset and the average across all actions. There is also considerable improvement on actions in the MSRC-12 dataset with similar poses (e.g. change weapon and night goggles) which were difficult to discriminate without the temporal history (see Figure 15). The higher accuracy of the proposed method may be attributed to the improved execution rate invariance gained by matching template fragments with DTW instead of fixed size feature windows as used by Fothergill et al. [34] and Sharaf et al. [36]. Although both Zhao et al. [29] and Ellis et al. [40] also perform online action recognition they use the non-gaming actions in the MSRC-12 dataset so a comparison with their accuracy results is not possible.

The proposed method runs in real time (60fps) with low average observational latency of 2 frames (67ms). The observational latency of the proposed approach is very low in comparison to [29]. that have an observation latency of 830-1500ms. The significantly lower observation latency of the proposed method was achieved by using considerably less frames in the sliding window than [29] in conjunction with the explicit identification of the peak key pose.

Figure 13 is an example sequence from the G3D dataset which illustrates the low latency that is achieved by the explicit peak pose (dotted horizontal line). The ground truth action points (*) and the vertical dashed lines represent the time window ($\pm\Delta$) where the action point is deemed to be correctly detected. The detected action points (o) show that the proposed approach has a very low latency and high accuracy.

4.3. Early action recognition

Most work on early recognition has been done in the video modality on activities that were pre-segmented [9, 10, 11, 12, 28, 13, 14, 38] and therefore a direct comparison is not feasible. Instead pose-based approaches for online action recognition have been adapted for early action recognition in a continuous stream to evaluate their effectiveness at a similar task.

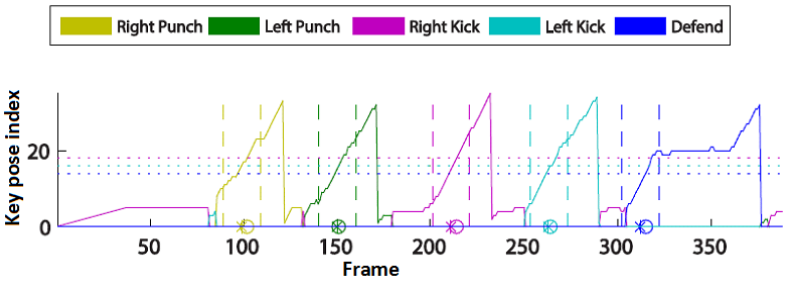

Figure 13: Clustered Action Manifold cluster indices for each frame with ground truth action points (*) and detected action points (o).

4.3.1. Performance Metrics

In the video domain, Hoai and De la Torre [28] recorded the F_1-scores as the action of interest unrolled from 0.1 to 1 and refer to this as the F_1-score curve. However, the percentage of action observed can only be calculated for sequences that have been pre-segmented to contain a single action. [16] uses the metrics of Percentage of Discarded Classes (PDC) and Percentage of early Labelling (PEL), while [37] proposes the calibrated Average Precision (cAP). However, none of these provide an insight how early action recognition results are evolved over time and prior to the peak of an action. Lan et al. [13] use the temporal distance (in frames) to report accuracy. In real world scenarios such as gaming the videos are not pre-segmented, instead action points are provided as temporal anchors and the latter frame-based metric seems the most appropriate measurement. For example, the methods performance at a temporal stage -20 describes the classification accuracy given all of the testing frames up to 20 frames before the action peak.

4.3.2. Comparative Study

The algorithms evaluated in the previous section with source code available were adapted for early action recognition: Random Forests, AdaBoost, Dynamic Feature Selection. Before the final detection step these algorithms output a frame based classification that is used for early action recognition. Similarly,

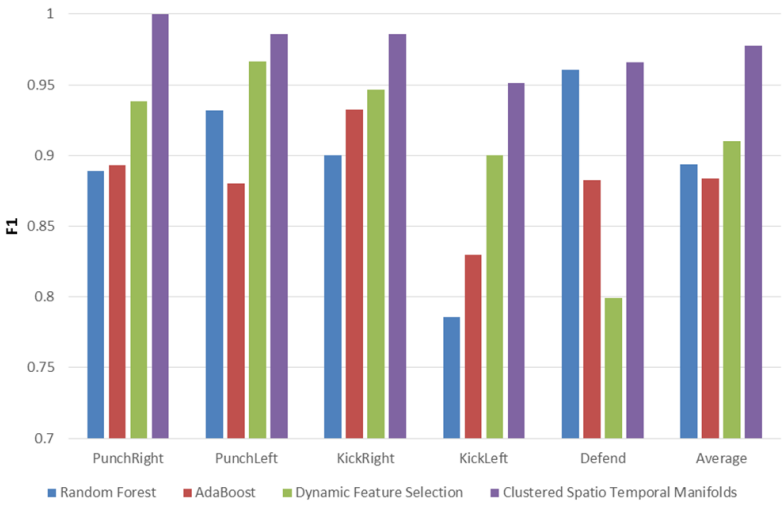

Figure 14: G3D Fighting Online Action Recognition Results by Action.

Clustered Spatio-Temporal Manifolds, the algorithm proposed in this paper, also outputs a frame-based classification before the final action detection.

4.3.3. Early Action Recognition Results

The proposed method significantly outperforms all of the comparative methods at all temporal stages across both datasets as illustrated in Figure 16 and Figure 17. The graphs show the methods frame F_1-score at different temporal stages from 20 frames before the action peak -20 to the peak of the action 0. The proposed method reaches 80% accuracy 16 and 10 frames before the action peak on the MSRC-12 and G3D datasets respectively, whereas the comparative methods achieve less than 30% accuracy at similar stages. The significant improvement in classification accuracy especially in the early stages of the action can be attributed to the proposed temporal models. The majority of failure cases were in the neutral or very early stage of the action as shown in Figures 18 and 19 where the action is ambiguous. The proposed method achieves 97.8% and 100% accuracy on the MSRC-12 and G3D dataset respectively at the action peak. The failure cases at the action peak in the MSRC-12 dataset were

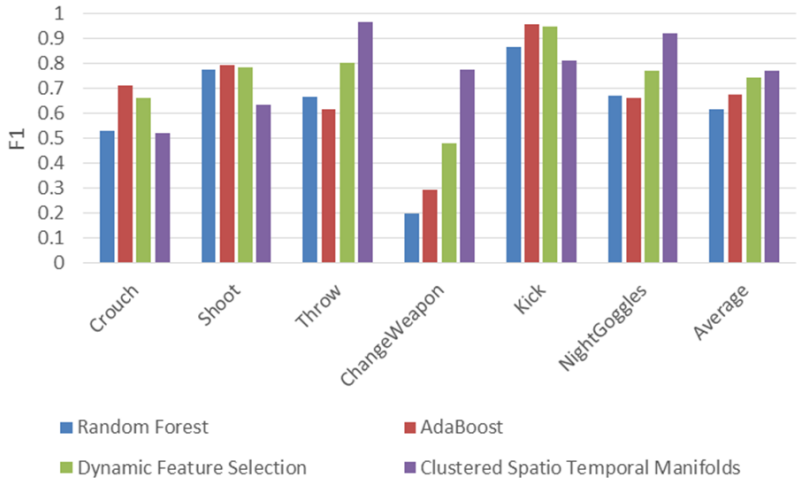

Figure 15: MSRC-12 Fighting Online Action Recognition Results by Action.

mainly due to the Change Weapon action which in some cases appears very similar to the neutral pose at the peak as illustrated in Table 4. The action peak frame based F_1 results are higher than the action point F1 scores reported in the previous section because the frame based metric used in this section is only concerned with classification and not the temporal detection of the action peak of which the latter is a more difficult task. Finally, the proposed approach obtains 76.3% on the MSRC-12 dataset 20 frames before the peak which may be attributed to the fact that the MSRC-12 actions typically have longer onset than G3D actions, especially the Change Weapon, Shoot and Throw actions.

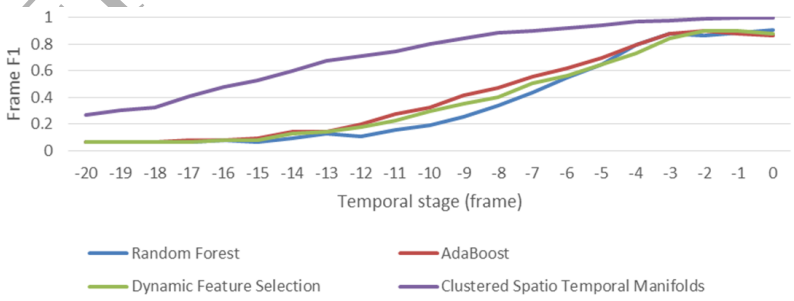

Figure 16: G3D Frame F1-scores, the average over ten leave-persons-out runs are shown.

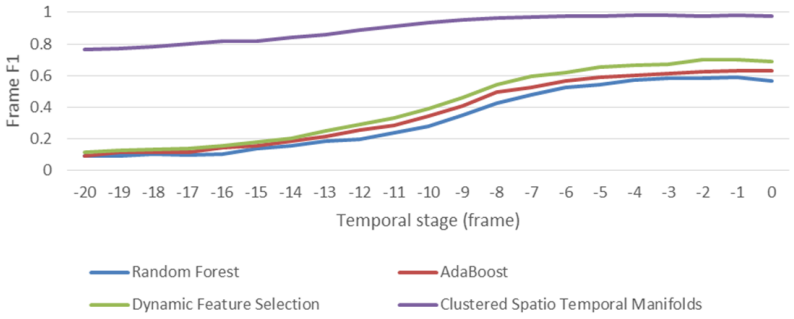

Figure 17: MSRC-12 Frame F1-scores, the average over ten leave-persons-out runs are shown.

4.4. Action Prediction

The existing work on action prediction has also been performed in the video modality and therefore a comparison is not feasible. Instead, the comparative pose-based approaches for early action recognition have been extended with the same linear regression as described in 3.2.3 to evaluate their effectiveness at action prediction.

4.4.1. Performance Metrics

Huang and Kitani [16] use average frame distance (AFD) to evaluate the accuracy of their predicted poses. AFD is a good measure of the spatial prediction but does explicitly measure the latency of the temporal prediction. In the proposed method the emphasis is on the temporal prediction of the peak pose, to the best of our knowledge there are no existing metrics for predicting the peak of the action. However, the Action Point F_1-score is a latency-aware metric for online action recognition that can be adapted to measure the accuracy of the predicted action points t_{p_a} instead of measuring the accuracy of the detected action points t_{d_a}, by modifying Eq. 9 to Eq. 12.

$$\Phi_p(t_{p_a}, t_{g_a}, \Delta) = \begin{cases} 1 \; if \; |t_{g_a} - t_{p_a}| \leq \Delta \\ 0 \; otherwise \end{cases} \tag{12}$$

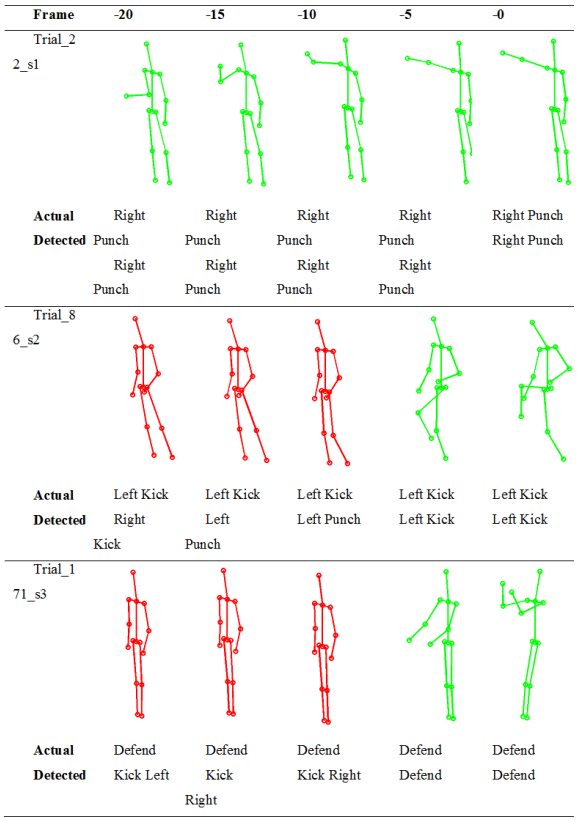

Figure 18: G3D Temporal Frame Based Results: Correct classifications are shown in green and failure cases in red. The majority of failure cases were in the neutral or very early stage of the action.

For a new test sequence, the arrival of data can be simulated and the predicted action point F_1-scores recorded. The predicted action point metric measures instances rather than frame based predictions so it will be referred to as the action point F_1-score curve.

4.4.2. Comparative Study

To extend the early recognition algorithms with linear regression, the methods need to output a certainty measure for each action at each frame. This is the case for two out of the three algorithms evaluated in the previous section: Ad-

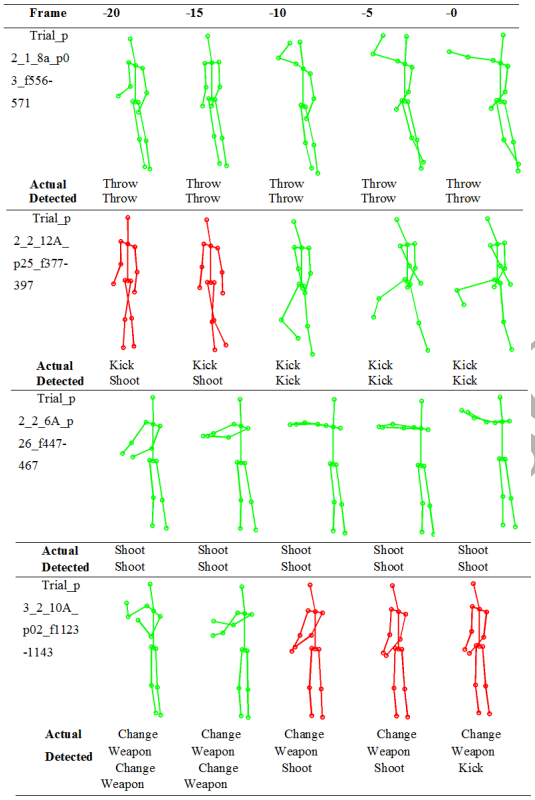

Figure 19: MSRC-12 Temporal Frame Based Results: Correct classifications are shown in green and failure cases in red. The majority of failure cases were in the neutral or very early stage of the action but there were also some cases at the peak of the action as in some cases the peak pose for Change Weapon is very similar to the neutral pose.

aBoost and Dynamic Feature Selection. Random Forests could not be adapted for prediction as the frame based result was a classification. Clustered Spatio-Temporal Manifolds, the algorithm proposed in this paper, outputs a cluster index for each frame which can be used in conjunction with the peak key pose index for prediction. The parameter required for prediction is the number of sequential frames for the linear regression. An exhaustive search was performed on the training set and the optimum result for AdaBoost and Dynamic Feature Selection was ($n_m = 2$) and for the Clustered Spatio-Temporal Manifolds the

optimum value was ($n_m = 6$).

4.4.3. Action Prediction Results

To measure how precisely the peak of the action can be predicted for all subjects the action point F_1 metric was captured as the continuous stream progressed. The proposed method significantly outperforms all of the comparative methods at all temporal stages on the G3D dataset as illustrated in Figure 20 and across the majority of temporal stages on the MSRC-12 dataset as illustrated in Figure 21. The graphs show the methods action point F_1-score at different temporal stages from 20 frames before the action peak -20 to the peak of the action 0. The proposed method works in a continuous stream, where the prediction is made as early as possible and early incorrect predictions decrease the final F_1-score. Even at the action peak prediction accuracy is less than online action recognition as the latter approach delays the detection until the peak has been observed. The proposed method reaches 38.1% and 45.6% 10 frames before the action peak. Predicting the point in time at which the peak pose will occur is a much more complex task than early detection of the action class or online action recognition, so a decrease in performance is expected. This is supported by the fact that the comparative approaches only reached a maximum of 24% at 10 frames before the action peak. The improvement in prediction of the proposed method can be attributed to the style invariant temporal model that is learnt for each action which includes explicit identification of a generic peak key pose.

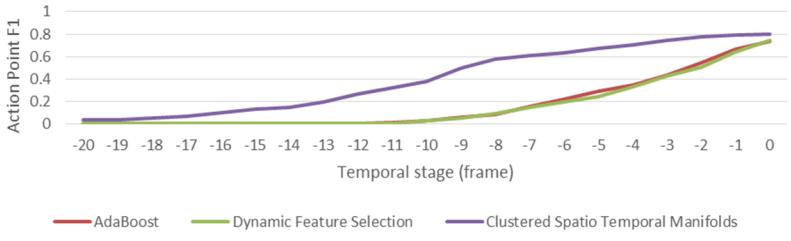

Figure 20: G3D Action Point F1-score curves, the average over ten leave-persons-out runs are shown.

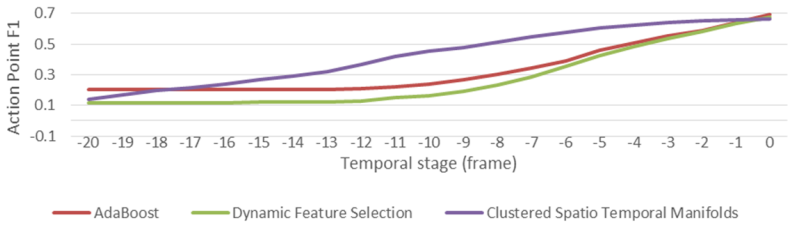

Figure 21: MSRC-12 Action Point F1-score curves, the average over ten leave-persons-out runs are shown.

A key benefit of the proposed prediction framework is that it is invariant to execution speed; the experimental results show that the regression line for a faster subject has a steeper gradient than the regression line for slower subject performing the same action and in both cases the action peak is detected correctly (see Figure 22).

5. Conclusion

The core of the proposed methods in this paper are the Clustered Spatio-Temporal Manifolds, which are compact style invariant models of the complex dynamics of human actions. They enable action classification in a continuous stream for early action detection in addition to the ability to track the progress of the action so that the peak can be detected with low latency or even predicted.

Application such as early action recognition and action prediction are feasible thanks to a linear latent space defined by the combination of TLE and k-means; TLE reduce style variance whilst still maintaining the temporal dynamics of the action, while k-means leads to equally distant cluster centres along the action temporal structure.

The action templates were effectively matched using DTW for execution rate invariance. To reduce the high observational latency of template matching a sliding window approach was used to match template fragments with low latency. The proposed approach achieved high accuracy for early action recognition and in contrast to existing approaches can operate in a continuous stream.

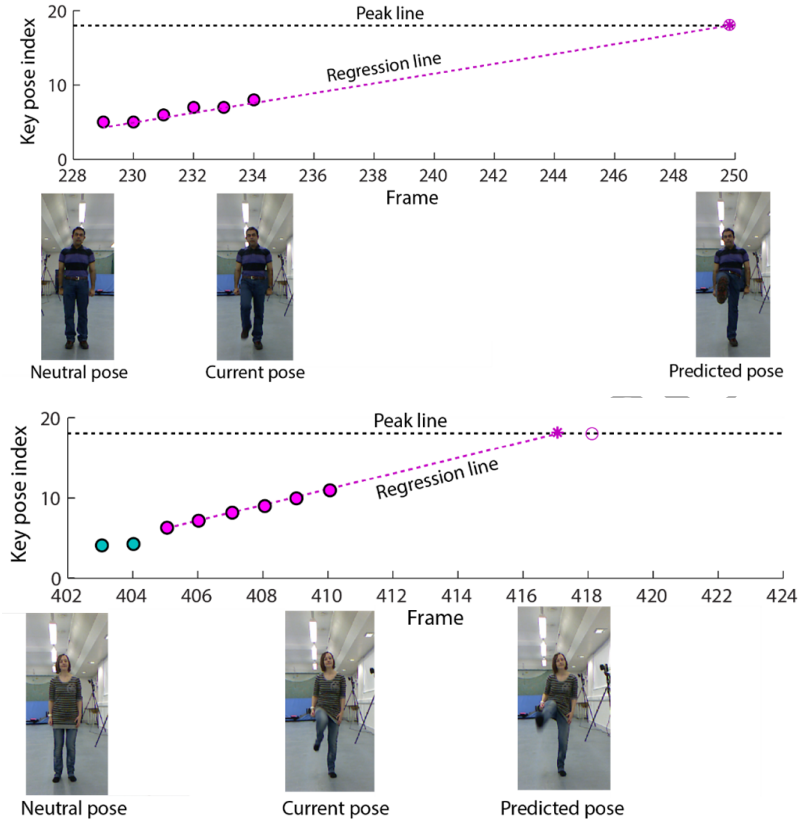

Figure 22: Two subjects performing a (right kick), at different speeds (classified right kick poses blue(), classified left kick poses pink(), ground truth peak pose pink(*), predicted peak pose).

Peak key poses were introduced to explicitly and precisely locate the moment where an action reaches its peak which enabled low latency recognition before the completion of the action. Experimental results on publicly available gaming action datasets demonstrate high accuracy with very low latency.

This paper also introduced the novel and challenging problem of predicting the action peak in a continuous stream. The proposed solution integrates the recent action progress history with regression for fast estimation of the peak. Experiments on public action recognition datasets showed that the proposed

method outperforms the comparative approaches and makes reasonable predictions even when there is a significant variation in the style and execution rate of the subject.

References

[1] B. Liang, L. Zheng, A survey on human action recognition using depth sensors, International Conference on Digital Image Computing: Techniques and Applications (DICTA), Adelaide, SA (2015) 1–8.

[2] M. Lewandowski, D. Makris, J. Nebel, Temporal extension of laplacian eigenmaps for unsupervised dimensionality reduction of time series, in International Conference on Pattern Recognition (2010) 161 164.

[3] D. Gong, G. Medioni, Dynamic manifold warping for view invariant action recognition, Int. Conf. Comput. Vis. (2011) 571578.

[4] M. Lewandowski, D. Makris, S. Velastin, J.-C. Nebel, Structural laplacian eigenmaps for modeling sets of multivariate sequences, IEEE Trans. Cybern.

[5] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3d skeletons as points in a lie group, IEEE Conf. Comput. Vis. Pattern Recognit.

[6] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, A. Baskurt, Sequential deep learning for human action recognition, in Human Behavior Understanding, Springer (2011) 2939.

[7] S. Ji, W. Xu, M. Yang, K. Yu, 3d convolutional neural networks for human action recognition, Pattern Anal. Mach. Intell. IEEE Trans. 35 (1) (2013) 221231.

[8] G. Taylor, R. Fergus, Y. LeCun, C. Bregler, Convolutional learning of spatio-temporal features, Lect. Notes Comput. Sci. (including Subser. Lect.

Notes Artif. Intell. Lect. Notes Bioinformatics) 6316 (PART 6) (2010) 140153.

[9] M. Ryoo, Human activity prediction: Early recognition of ongoing activities from streaming videos, Int. Conf. Comput. Vis. (2011) 10361043.

[10] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J. Siskind, S. Wang, Recognizing human activities from partially observed videos, CVPR.

[11] J. Davis, A. Tyagi, Minimal-latency human action recognition using reliable-inference, Image Vis. Comput. 24 (5) (2006) 455472.

[12] K. Li, Y. Fu, K. Li, Y. Fu, Arma-hmm: A new approach for early recognition of human activity, in Pattern Recognition (ICPR), 21st International Conference on (2012) 17791782.

[13] T. Lan, T. Chen, S. Savarese, A hierarchical representation for future action prediction, Comput. VisionECCV (2014) 689704.

[14] Y. Kong, D. Kit, Y. Fu, A discriminative model with multiple temporal scales for action prediction, in ECCV - European Conference on Computer Vision (2014) 596611.

[15] H. J. Escalante, E. F. Morales, L. E. Sucar, A nave bayes baseline for early gesture recognition, Pattern Recognition Letters 73 (2016) 91 – 99.

[16] D. Huang, S. Yao, Y. Wang, F. De La Torre, Sequential Max-Margin Event Detectors, Springer International Publishing, Cham, 2014, pp. 410–424.

[17] T. Starner, A. Pentland, Real-time american sign language recognition from video using hidden markov models, Comput. Vision, 1995. Proceedings., Int. Symp. (1995) 265270.

[18] N. Sebanz, H. Bekkering, G. Knoblich, Joint action: bodies and minds moving together, Trends Cogn. Sci. 10 (2) (2006) 706.

[19] S. Streuber, G. Knoblich, N. Sebanz, H. Blthoff, S. delaRosa, The effect of social context on the use of visual information, Exp. brain Res. 214 (2) (2011) 27384.

[20] K. Verfaillie, A. Daems, Representing and anticipating human actions in vision, Vis. cogn. 9 (1-2) (2002) 217232.

[21] J. Broekens, M. Heerink, H. Rosendal, Assistive social robots in elderly care: a review assistive social robots, Gerontechnology 8 (2).

[22] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 11571182.

[23] J. P.-L. P. Climent-Prez, A. Chaaraoui, F. Flrez-Revuelta, Optimal joint selection for skeletal data from rgb-d devices using a genetic algorithm, in Advances in Computational Intelligence 7630 (2013) 163174.

[24] R. F. L. Olshen, J. Breiman, C. J. Stone, Classification and regression trees, Wadsworth Int. Gr.

[25] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 532.

[26] F. Negin, F. Ozdemir, C. Akgul, K. A. Yuksel, A. Ercil, A decision forest based feature selection framework for action recognition from rgb-depth cameras, in International Conference on Image Analysis and Recognition (2013) 648657.

[27] L. Schwarz, D. Mateus, G. Mnchen, V. Castaeda, Manifold learning for tof-based human body tracking and activity recognition, in British Machine Vision Conference (2010) 80.180.11.

[28] M. Hoai, F. Torre, Max-margin early event detectors, Int. J. Comput. Vis.

[29] X. Zhao, S. Wang, X. Li, H. Zhang, Online action recognition by template matching, LNCS (2013) 269272.

[30] X. Miao, J. Heaton, A comparison of random forest and adaboost tree in ecosystem classification in east mojave desert, in Geoinformatics 18th International Conference on (2010) 16.

[31] S. Eickeler, A. Kosmala, G. Rigoll, Hidden markov model based continuous online gesture recognition, in International Conference on Pattern Recognition 2 (1998) 12061208.

[32] P. Natarajan, R. Nevatia, Online, real-time tracking and recognition of human actions, in IEEE Workshop on Motion and video Computing (2008) 18.

[33] S. Nowozin, J. Shotton, Action points: A representation for low-latency online human action recognition, Techincal Rep. (2012) 118.

[34] S. Fothergill, H. Mentis, P. Kohli, S. Nowozin, Instructing people for training gestural interactive systems, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (2012) 17371746.

[35] V. Bloom, V. Argyriou, D. Makris, Dynamic feature selection for online action recognition, in Human Behavior Understanding, Lecture Notes in Computer Science Switzerland Springer International Publishing LNCS (8212) (2013) 6476.

[36] A. Sharaf, M. Torki, M. Hussein, M. El-Saban, Real-time multi-scale action detection from 3d skeleton data, in IEEE Winter Conference on Applications of Computer Vision (WACV).

[37] R. D. Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, T. Tuytelaars, Online action detection, arXiv preprint arXiv:1604.06506.

[38] A. Galata, N. Johnson, D. Hogg, Learning variable-length markov models of behavior, Comput. Vis. Image Underst. 81 (3) (2001) 398413.

[39] C. Vondrick, H. Pirsiavash, A. Torralba, Anticipating the future by watching unlabeled video, Apr.

[40] C. Ellis, S. Masood, M. Tappen, J. Laviola, R. Sukthankar, Exploring the trade-off between accuracy and observational latency in action recognition, Int. J. Comput. Vis. 101 (3) (2013) 420436.

[41] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, IEEE 2 (3) (2011) 12971304.

[42] T. Kanungo, D. Mount, N. Netanyahu, A. Wu, C. Piatko, A local search approximation algorithm for k-means clustering, Spec. Issue 18th Annu. Symp. Comput. Geom. - SoCG2002 28 (23) (2003) 89112.

[43] A. Drake, Discrete-state markov processes, in Fundamentals of Applied Probability Theory, New York: McGraw-Hill (1967) 163203.

[44] Z. Zhao, A. Elgammal, Information theoretic key frame selection for action recognition, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2008, pp. 109.1–109.10, doi:10.5244/C.22.109.

[45] L. Liu, L. Shao, P. Rockett, Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition, Pattern Recogn. 46 (7) (2013) 1810–1818.

[46] A. Chaaraoui, J. Padilla-Lpez, P. Climent-Prez, F. Flrez-Revuelta, Evolutionary joint selection to improve human action recognition with rgb-d devices, Expert Syst. Appl. 41 (3) (2014) 786794.

[47] V. Ponce-Lpez, H. Jair-Escalante, S. Escalera, X. Bar, Gesture and action recognition by evolved dynamic subgestures, British Machine Vision Conference, BMVC.

[48] P. Senin, Dynamic time warping algorithm review, USA.

[49] V. Bloom, D. Makris, V. Argyriou, G3d: A gaming action dataset and real time action recognition evaluation framework, in Computer Vision and Pattern Recognition Workshop (CVPRW) IEEE Conference on (2012) 712.

[50] V. Bloom, D. Makris, V. Argyriou, Clustered spatio-temporal manifolds for online action recognition, 22nd International Conference on Pattern Recognition, Stockholm, Sweden.

[51] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting, Ann. Stat. 28 (1998) 2000.