Hierarchical Transfer Learning for Online Recognition of Compound Actions

Victoria Bloom [a,*], Vasileios Argyriou [a], Dimitrios Makris [a]
[a] Digital Imaging Research Centre, Kingston University, United Kingdom

Abstract

Recognising human actions in real-time can provide users with a natural user interface (NUI) enabling a range of innovative and immersive applications. A NUI application should not restrict users' movements; it should allow users to transition between actions in quick succession, which we term as compound actions. However, the majority of action recognition researchers have focused on individual actions, so their approaches are limited to recognising single actions or multiple actions that are temporally separated.

This paper proposes a novel online action recognition method for fast detection of compound actions. A key contribution is our hierarchical body model that can be automatically configured to detect actions based on the low level body parts that are the most discriminative for a particular action. Another key contribution is a transfer learning strategy to allow the tasks of action segmentation and whole body modelling to be performed on a related but simpler dataset, combined with automatic hierarchical body model adaption on a more complex target dataset.

Experimental results on a challenging and realistic dataset show an improvement in action recognition performance of 16% due to the introduction of our hierarchical transfer learning. The proposed algorithm is fast with an average latency of just 2 frames (66ms) and outperforms state of the art action recognition algorithms that are capable of fast online action recognition.

*Corresponding author.
E-mail addresses: Victoria.Bloom@kingston.ac.uk (V. Bloom), Vasileios.Argyriou@kingston.ac.uk (V. Argyriou), D.Makris@kingston.ac.uk (D. Makris).

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 and entertainment. In healthcare, action recognition enables touch-free browsing of medical images in operating rooms, physical therapy at home and in clinics and for patient monitoring. In education, action recognition can increase the engagement of users by providing realistic and immersive training simulations. 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.

Historically, action recognition research has focused on increasing accuracy on datasets in highly controlled environments. These datasets normally contained a single person that was instructed to perform a single action clearly (see Figure 1). Recognition was performed offline after viewing a complete sequence and algorithms were evaluated by the number of correctly classified sequences. A recent survey [1] showed perfect or near perfect action recognition accuracy on simple datasets with a small number of actions.

Figure 1 Simple boxing sequence with a single person performing a punch (KTH) [3]

The traditional offline approach led to simplification of the problem, overinflated accuracy and lack of applicability to real world situations. Recent research toward more realistic action recognition has changed to online action recognition where different actions are detected in real-time whilst they are being observed. However, the focus has been on recognising actions which are temporally well separated and easy to segment. In contrast, this work considers multiple actions performed in quick succession, which are critical for robust

action detection in natural user interface (NUI) applications. When multiple actions are performed in quick succession movements from different actions may temporally overlap resulting in complex poses, which we term as compound actions. For example, in a full body fighting game a player may throw punches in quick succession, one arm may still be finishing the previous punch whilst the other arm is performing the next punch or a player may leave one arm in the defend position and punch with the other arm (as shown in Figure 2). Detecting multiple actions in quick succession is a more complex problem than recognising actions which are temporally well separated.

Figure 2 Complex fighting sequences between multiple players, performing multiple actions in quick succession so that the movements temporally overlap (G3Di) [4]. Each row represents a different sequence with visual examples taken every 3 frames.

Existing work on recognising more complex actions has to date only been researched in an offline context. To evaluate the performance of action recognition algorithms on more realistic actions several datasets have been extracted from TV and film (YouTube Action Dataset [5], Hollywood Human Actions Dataset [6], UCF sports action dataset [7]). In these datasets the actions are performed in real-world scenarios with diverse and cluttered backgrounds as well as significant changes in viewpoint. The individual actions are realistic but the major limitation of these datasets is that they have been segmented into sequences containing a single action suitable for offline action recognition. The diversity and complexity of real-world datasets makes accurate labelling difficult and time consuming. To overcome this problem Ma et al. [8] employed transfer learning to transfer knowledge from a simpler domain (e.g. KTH [3]) to a more complex target domain (e.g. YouTube Action Dataset) but their approach was limited to offline action recognition. An area that has not been considered before is the potential for transfer learning to improve online action recognition.

Several NUI datasets with multiple actions in each sequence have been captured (MSRC-12 [9], G3D [10], G3Di [4]) and action points [11] provided, as temporal anchors to enable evaluation of online action recognition algorithms. Good performance has been achieved on the datasets where the actions were recorded under controlled circumstances (MSRC-12, G3D) but performance dramatically decreased when the same algorithm [4] was applied to a real-world scenario of a full body fighting game (G3Di). All three datasets contain multiple actions but the difference is that the MSRC-12 and G3D datasets contain actions that are temporally well separated whereas the G3Di dataset, contains transitions between actions and even multiple actions at the same time. Temporal merging of a user's actions results in compound actions comprising of movements from different actions, which have not been adequately addressed by existing approaches.

In this work we propose a novel hierarchical transfer learning algorithm for online action recognition of compound actions. Specifically, transfer learning is employed to allow the tasks of action segmentation and modelling to be performed on a related but simpler dataset, combined with model adaptation to improve performance on a more complex dataset. Furthermore, we represent actions hierarchically to provide the flexibility to recognise poses that are not in

the source dataset by introducing independence between limbs. Evaluation on a realistic and challenging public action dataset confirms the effectiveness of our approach.

2. Literature Review

A key requirement of many real-world applications is the ability to recognise actions online. However, recent surveys [12], [13] show that the majority of existing action recognition algorithms are offline and rely on observing a pre-segmented action sequence before classification of a single action. A common adaptation of existing approaches is to use a sliding window and classify the current frame based on the recent temporal history. This enables continuous recognition of multiple actions in real world scenarios such as monitoring elderly patients at home [14]. However, there is an additional requirement in NUI applications to detect actions with low latency so the system can provide an appropriate response to the user with no apparent delay. For example, increasing the volume on a Smart TV by raising a hand should be detected with low latency to provide natural interaction.

Existing work has demonstrated that action points [11], temporal anchors within the course of the action are important for evaluating the latency of the detection. An action point is a single pose that can be clearly and easily identified as a representative of an action. Several, sliding window approaches for online action recognition have been validated using action points [9], [15], [16]. Fothergill et al. [9] used fixed size sliding windows on the streaming data and performed the classification by a Random Forest. Similarily, Bloom et al. [15] used a fixed size sliding window and perform the classification by AdaBoost. However, the fixed size of the sliding window in both approaches is a source of classification error due to execution rate variations. To address this Zhao et al. [16] optimise the size of the segment during their feature extraction using a DTW variant for subsequence matching. However, as these methods were tested with temporally separated actions their ability to robustly detect compound actions is unclear. Especially as AdaBoost which achieved good performance on relatively simple actions [17] but when applied to more complex actions performance dramatically decreased [4].

Manual labelling of action points is possible in complex datasets as they represent the most significant part of the action, however subsequently automatically selecting a sequence of training examples around the point leads to inconsistencies. Firstly, as some actions have long duration such as defending (see Figure 2), later samples of the current action will be incorrectly selected as negative samples. Secondly, samples from another action class may be incorrectly selected due to the close proximity of neighbouring actions (see Figure 2). The first problem has been overcome by action segments [4] which incorporate the duration of the most significant part of the action. The second problem has not yet been adequately addressed but could be alleviated by reducing the need for labelling.

Transfer learning [18] has been beneficial to many machine learning research areas, including classification, regression and clustering problems to reduce the need to collect and label training data. However, transfer learning applied to action recognition is a relatively new topic with limited research in the computer vision community. Transfer learning has been used for cross-view action recognition [19], [20] to recognise human actions from different views. In both cases the methods were tested offline on a multi-view dataset (IXMAS) [21], which comprised of simple actions with simple backgrounds so it has limited applicability to real world scenarios.

More significantly transfer learning has been used cross-dataset [8], [36] to harness lab datasets to facilitate real-world action recognition. The aim is to generalise action models built from a source dataset to a target dataset, to alleviate the problem of labelling complex sequences. The source dataset typically has a clean background and each video clip may involve only one type of action and a single person, which describes most lab collected datasets. In contrast, in the target dataset the background may be cluttered and there may be multiple people and multiple actions which may overlap temporally. Cross-dataset learning aims to adapt the existing classifier from a source dataset to a new target dataset, while requiring only a small or even no labelled samples in the target dataset. Ma et al. [8] built a model within a multi-task framework so the actions of one domain are associated with its own features. The general Schatten *p*-norm was applied to mine the shared components between the lab data and the real world data. The main advantage of their approach is the ability to share knowledge between the

two datasets even if they have different action categories. However, the method was tested offline with sequences containing just a single action. Cao et al. [22] combine model adaption and action detection into a Maximum a Posterior (MAP) estimation framework for action detection. The advantage of this approach over the previous method is that it can perform spatial-temporal detection of the action within a sequence. However, as a search for the optimal 3D sub-volume is performed across all frames in the target sequence this approach is also offline.

The approaches described so far are limited to single actions or multiple actions that are temporally separated. However, in NUI applications the user may wish to perform multiple actions in quick. This temporal merging of different actions results in complex poses comprising of movements from multiple actions. Hierarchical models have been successfully applied to pose estimation [23]–[27] to recover novel poses not present in the training dataset. Hierarchical models have also been applied to improve action recognition performance [28]. Following the popular bag-of-words approach several efforts constructed a hierarchical representation of local feature descriptors but as the temporal order is ignored they are not suited to many real-world problems. To overcome this Song et al. [29] propose hierarchical sequence summarisation to capture discriminative information at various temporal resolutions. However, as the testing was performed at the sequence level this approach is limited to offline action recognition.

2.1 Contributions

We propose a novel hierarchical transfer learning algorithm for online detection of compound actions for robust action recognition in natural user interface (NUI) applications. Specifically, transfer learning is employed to allow the tasks of action segmentation and modelling to be performed on a related but simpler dataset, combined with model adaptation to improve performance on a complex NUI dataset. We represent actions using a hierarchical human body model to allow independence between low-level body parts. Our novelty is to automatically weight each low-level body part based on their discriminative ability to detect specific actions. We propose hierarchical peak poses for low latency detection which provide the flexibility to recognise poses that are not in the source dataset. Hierarchical template matching is performed with Dynamic

Time Warping (DTW) to ensure execution rate invariance and we use a sliding window approach for online recognition. Evaluation on a public dataset with complex, realistic actions demonstrates that our approach outperforms existing methods in terms of accuracy and latency.

3. Methodology

The proposed method for online action recognition consists of two phases: an offline training phase and an online testing phase as illustrated in Figure 3.

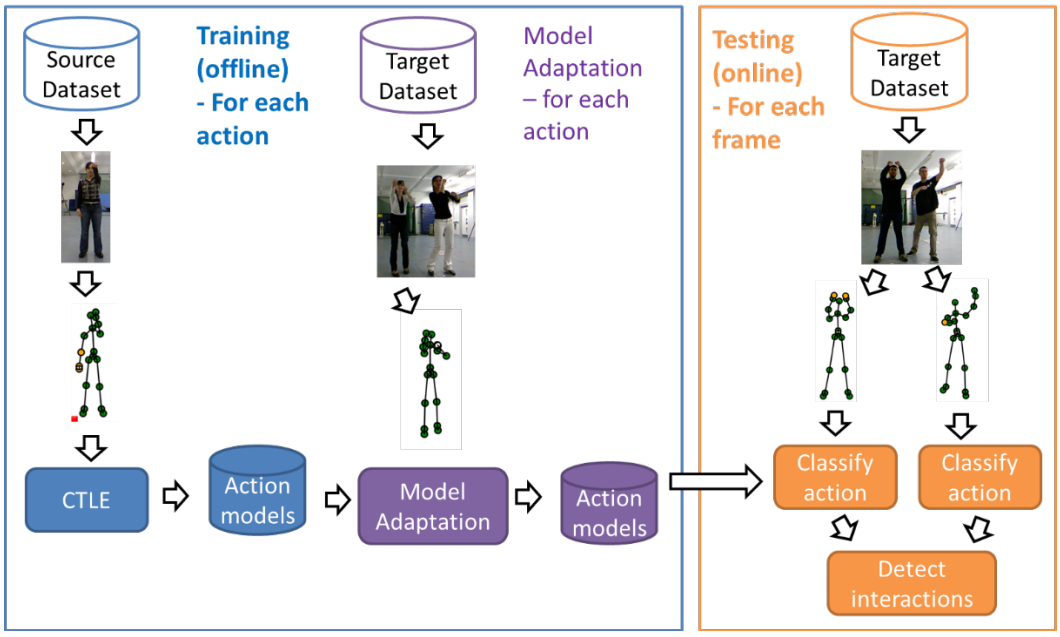

Figure 3 Methodology overview

We propose a novel hierarchical transfer learning algorithm for online detection of compound actions for fast and robust action recognition in natural user interface (NUI) applications. Our method is based on skeleton data, specifically joint angles which are viewpoint and anthropometric invariant and can be generated in real-time with a pose estimation method [30]. A key contribution is our hierarchical body model that can be automatically configured to detect actions based on the low level body parts that are the most discriminative for a particular action. Another key contribution is a transfer learning strategy to allow the tasks of action segmentation and whole body modelling to be performed on a related but simpler source dataset, combined with automatic hierarchical body model adaption on a more complex target dataset (as shown in Figure 3).

3.1 Training (source dataset)

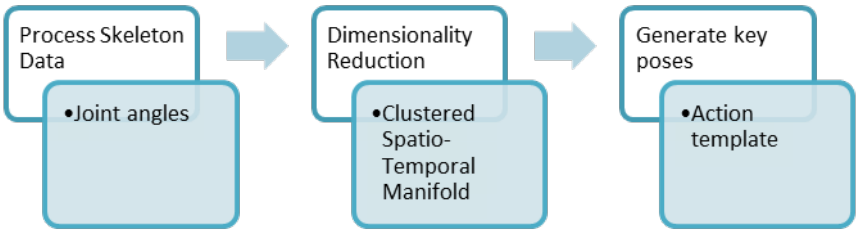

Figure 4 Training overview which is performed on the source dataset for each action

The training phase is based on our existing approach for online action detection [17] that achieved high accuracy and low latency for multiple actions that were separated temporally (see Figure 4). Our contribution is to adapt these action templates to detect compound actions by representing and detecting actions hierarchically. The two key stages in training, as published in our previous work [17] are dimensionality reduction and key pose generation. Dimensionality reduction of the skeleton data produces spatio-temporal manifolds which removes individual style whilst maintaining the temporal ordering of the poses. Clustering the manifolds and projecting the cluster centres back to the high dimensional space creates key poses. An individual key pose represents a generic pose from an action at a specific point in time and the sequence of these key poses represent the entire action (as illustrated in Figure 5). A major benefit of the clustering is that the number of key poses is significantly less than the original number of training poses which dramatically reduces the computation time and enables our approach to scale efficiently to much larger datasets.

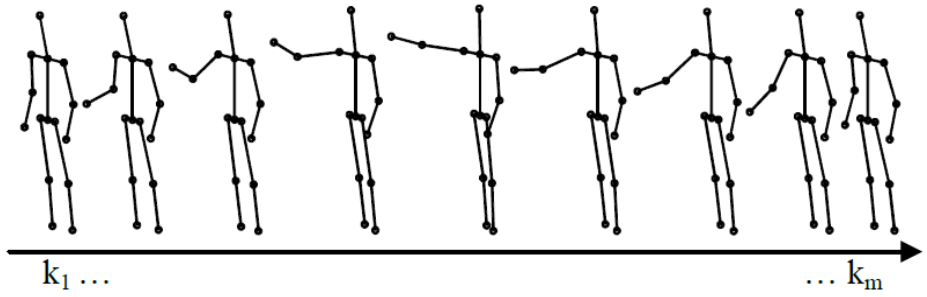

Figure 5 Right punch action template, consisting of key poses k_1 to k_m where m is the number of clusters [17]

The two stages are explained in detail below:

3.1.1. Dimensionality reduction

Stylistic variations are removed by learning a clustered spatio-temporal manifold (CSTM) for each action [17]. Given a set of training poses from the source dataset $X = \{x_i\}_{(i=1\ldots n)}$, $x_i \in \mathbb{R}^D$, distributed in a high dimensional space, Temporal Laplacian Eignemaps (TLE) [31] discovers their low dimensional representation $X' = \{x'_i\}_{(i=1\ldots n)}$, $x'_i \in \mathbb{R}^d$ where $d \ll D$ by combining two neighbourhood graphs. Temporal neighbours are the closest points in the sequential order and spatial neighbours are the geometrically similar neighbours. These neighbour relations are used in the construction of two graphs where any two vertices are connected when a neighbour relationship exists between these points. 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 (see Figure 6).

Clustering is then performed on the embedded space to reduce computation time by removing redundant poses. *k*-means [32] is applied to cluster the n low dimensional points X' into m clusters $C = (c_j)_{(j=1\ldots m)}$, $c_k \in \mathbb{R}^d$, where $m \ll n$.

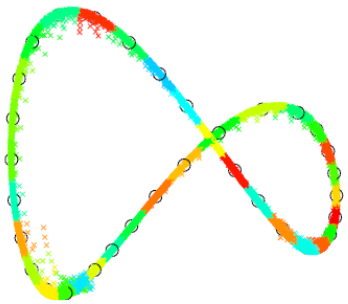

Figure 6 Clustered Spatio-Temporal manifold with the low dimensional points plotted (x_i), coloured based on the cluster to which they belong and the cluster centers (c_j) as black circles [17].

3.1.2. Key pose generation

Key poses remove redundant information to improve classification accuracy and reduce the computational latency of action detection [14], [17]. To generate key poses we follow the method proposed in [31] that uses the training set $M = \{x_i, x'_i\}_{(i=1\ldots n)}$ to learn a Radial Basis Function Network (RBFN) that

represents the mapping between the embedded and the high dimensional space [31]. Then using the RBFN mappings the cluster centers are projected into the high dimensional space to generate new poses that are a direct representation of the average poses. The implicit temporal order in the low dimensional space can be extracted from the training data to order the corresponding key poses $K = \{k_j\}_{(j=1\ldots m)}$ to create action templates (K) for each action as illustrated in Figure 5. Action templates are the high dimensional representations of the clustered spatio-temporal models and inherit their advantages, including style invariance and compactness.

3.2. Model Adaptation (target dataset)

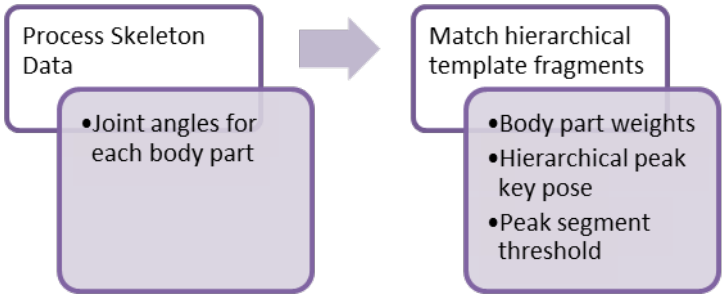

Figure 7 Model Adaptation overview which is performed on the target dataset for each action

To detect compound actions such as those performed in NUI applications we propose a hierarchical template matching algorithm (see Figure 7). Representing actions using a hierarchical model of human body allows independence between the low-level body parts $B = (b_l)_{(l=1\ldots L)}$ (as illustrated in Figure 8). Each low-level body part is represented by joint angles. Our contribution is to automatically weigh each low-level body part based on their discriminative ability to detect specific actions. Weighting the individual low-level body parts, creates flexible body part configurations at different levels of a normal body hierarchy e.g. whole body, upper body or right arm and atypical combinations such as right arm and left leg.

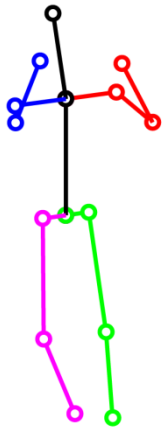

Figure 8 Low level body parts: the skeleton is divided into low level body parts, right arm (red), left arm (blue), right leg (green), left leg (pink) and torso (black).

The action peak is a fundamental concept of the proposed approach which we define as the segment in time when the goal of the action is being satisfied. For example, in a boxing game the aim of the punch is to hit the opponent which is being fulfilled when the arm is maximally extended as shown in Figure 9. The peak poses in the training data of the target dataset are manually labelled with an action label, there must be at least one frame labelled as the peak pose for each action instance. If the action peak has duration, as in the case of the defend action there will be multiple sequential labelled frames.

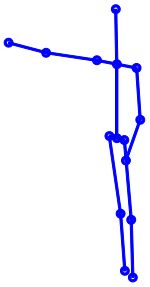

Figure 9 Action peak for right punch action

There are three main steps to adapt the action templates learnt from the source dataset for hierarchical template matching: learning the most discriminative body part combinations, detecting the most representative hierarchical peak key pose and optimising the peak segment threshold.

All three steps use exemplar matching between the peak poses in the target dataset training poses and the action templates to find the optimum matching parameters. To incorporate the temporal history of the action and increase the robustness of the matching process sequences of poses are matched rather than

single poses. To extract a fragment F from a sequence of poses $S = (s_1, s_2, \ldots s_G)$ Eq. 1 is used:

$$F(S, i) = (s_{i-s}, s_{i-s+1}, \ldots s_i) \tag{1}$$

where, i is the pose index, s is the number of poses in the fragment and G is the number of poses in sequence S and the conditions $i > s$ and $i \leq G$ are satisfied.

DTW [33] is a well-known algorithm for determining the similarity of time-series data that allows "elastic" transformation to gain execution rate invariance. The similarity of two series of poses, the query sequence $Q = (q_1, q_2, \ldots q_U)$ and the reference sequence $R = (r_1, r_2, \ldots r_V)$ can be computed using the standard DTW distance metric using Eq. 2.

$$DTW(Q, R) = \min\{c_p(Q, R), p \in P^{U \times V}\} \tag{2}$$

Where c_p is the global cost function associated with a warping path $p = (p_1, \ldots, p_H)$ and c is the local cost function, which is the Euclidean distance between two poses, which will be small if the poses are similar to each other:

$$c_p(Q, R) = \sum_{h=1}^{H} c(q_{uh}, r_{vh}) \tag{3}$$

In our previous approach [17] the DTW distance was computed for the whole body. To increase flexibility we propose a hierarchical DTW distance measurement ($HDTW$):

$$HDTW(Q, R, W) = \sum_{l=1}^{L} DTW(Q_l, R_l) W_l \tag{4}$$

For two series of poses, the query sequence Q and the reference sequence R, the similarity of low level body parts l is computed independently using the standard DTW distance metric. A weighted combination $W = (w_l)_{(l=1\ldots L)}, w_l \in (0,1)$ of the low level body part distances provides a discriminative distance metric for compound actions.

3.2.1. Body Part Combinations

The most discriminative body part combinations for each action are discovered by maximising the ratio of intra-class matches between the labelled peak poses in the target dataset training data and the action templates. This procedure is repeated

for all body part combinations, so for computational efficiency we selected binary weights, $w_l \in (0,1)$ for each of the low level body parts which results in 2^L permutations. For each permutation ε, the intra-class ratio ρ is computed by the number of intra-class matches μ over the number of total training instances in the target dataset n^y. The intra-class matches are counted for each action by exemplar matching between the peak poses from the target dataset training data and the key poses from all the action templates. For each action a, if the closest matching action template is the same action this is counted as an intra-class match. The maximum intra-class ratio represents the most discriminative body part combination for each action, as illustrated in Figure 10 and summarised in Algorithm 1.

Algorithm 1 Learn the most discriminative weights for each action

Input: Given a set of training poses from the target dataset $Y = \{y_i\}_{(i=1\ldots|Y|)}$, with manually selected peak poses from Y represented by their indices $I^a = \{i_p^a\}_{(p=1\ldots|I^a|)}$, where $i_p \in 1 \ldots |Y|$ and the superscript denotes a set of action templates $K^a = \{k_j\}_{(j=1\ldots m)}$, where m is the number of clusters.

1. For each action, $a = 1{:}\,A$
   1.1. For each permutation, $\varepsilon = 1{:}\,2^L$
      1.1.1. Initialise $\mu = 0$
      1.1.2. For each peak pose, $p = 1{:}\,|I^a|$
         1.1.2.1. Extract the peak pose fragment, $F^{\mathrm{Y}} = (Y, i_p^a)$ using Eq. 1
         1.1.2.2. $a^* = \min_{a' \in A} HDTW(F^{\mathrm{Y}}, K^{a'}, W_\varepsilon)$ using Eq. 4
         1.1.2.3. If $a^* = a$
            1.1.2.3.1. Intra-class match so increment μ
      1.1.3. Compute intra-class ratio, $\rho_\varepsilon^a = \frac{\mu}{|I^a|}$
   1.2. Select the most discriminative weights, $W^a = \arg\max_\varepsilon \rho_\varepsilon^a$
   1.3. Output the weights for this action, W^a

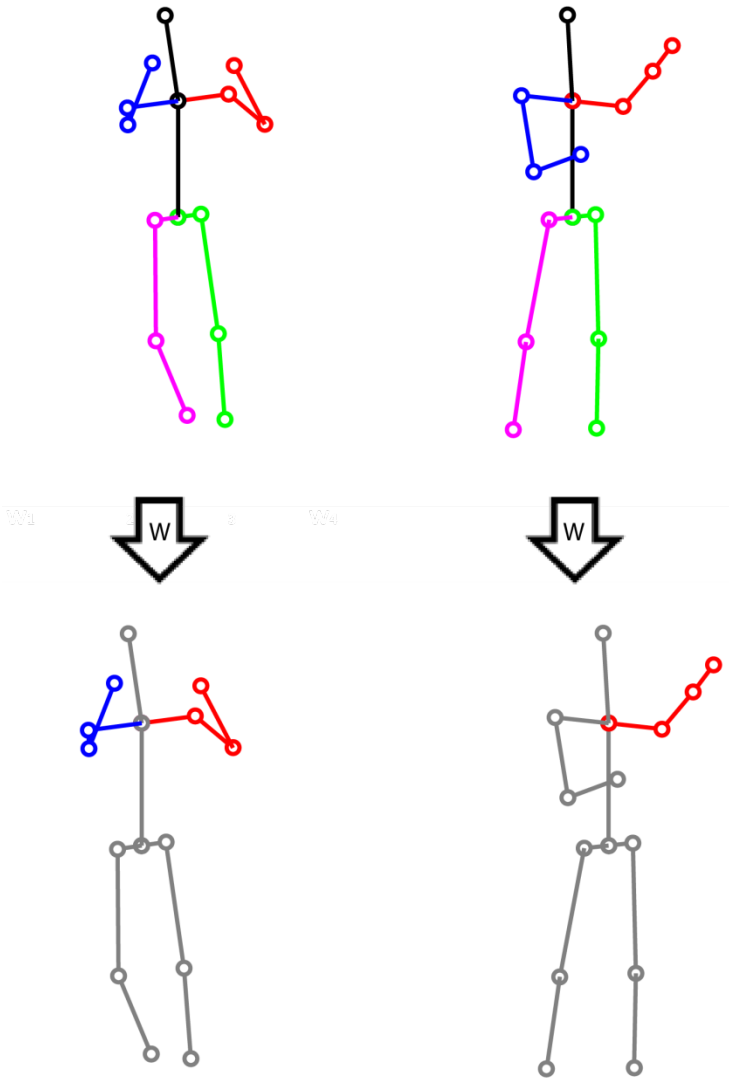

Figure 10 Body Part Combinations: The weights (W) are optimised for each action based on their ability to discriminate complex actions in the target dataset. The bottom skeletons show potential body parts configurations for the defend (left) and right punch (right) actions.

3.2.2. Hierarchical Peak Key Pose

In our previous work on simple actions, peak key poses were proposed as the generic representation of peak poses in the training data and were automatically selected from the key poses by exemplar matching with the whole body [17]. To increase robustness on compound actions we propose hierarchical peak key poses. Hierarchical peak key poses are also automatically selected from the key poses but the exemplar matching is performed using the most discriminative body parts rather than the whole body. The hierarchical peak key poses are selected as follows: for each action and for each peak pose in the target dataset training data,

the best matching key pose is found (as shown in Figure 11). A hierarchical peak key pose can be represented by its index j^a in the action template. The best matching index j^* is found by minimising the distance between the peak pose fragments F^{Y} and the key pose fragments F^{K} using the most discriminative body part combination for each action. The hierarchical peak key pose for the action is the key pose that has the maximum number of matches, as summarised in algorithm 2.

Figure 11 Hierarchical template matching: peak pose (left), best matched key pose (right)

Algorithm 2 Learn the hierarchical peak key pose

Input: Given a set of training poses from the target dataset $Y = \{y_n\}_{(n=1\ldots|Y|)}$ with manually selected peak poses from Y represented by their indices $I^a = \{i_p^a\}_{(p=1\ldots|I^a|)}$, where $i_p \in 1\ldots|Y|$ and the superscript denotes a set of action templates $K^a = \{k_j\}_{(j=1\ldots m)}$ with weights W^a:

1. For each action, $a = 1{:}\,A$
   1.1. Initialise $J = \{0\}_{(1\ldots m)}$
   1.2. For each peak pose, $p = 1{:}\,|I^a|$
      1.2.1. Extract the peak pose fragment $F^{\mathrm{Y}} = (Y, i_p^a)$ using Eq. 1
      1.2.2. Find the best matching hierarchical key pose index,
      $j^* = \arg\min_{j\in 1\ldots m} \sum_{l=1}^{L} HDTW\left(F_l^{\mathrm{Y}}, F_l^{\mathrm{K}}, W_l^a\right)$, where $F_l^{\mathrm{K}} = (K_l^a, j)$
      1.2.3. Increment J_{j^*}
   1.3. Output the hierarchical key pose index $j^a = \arg\max \; J$

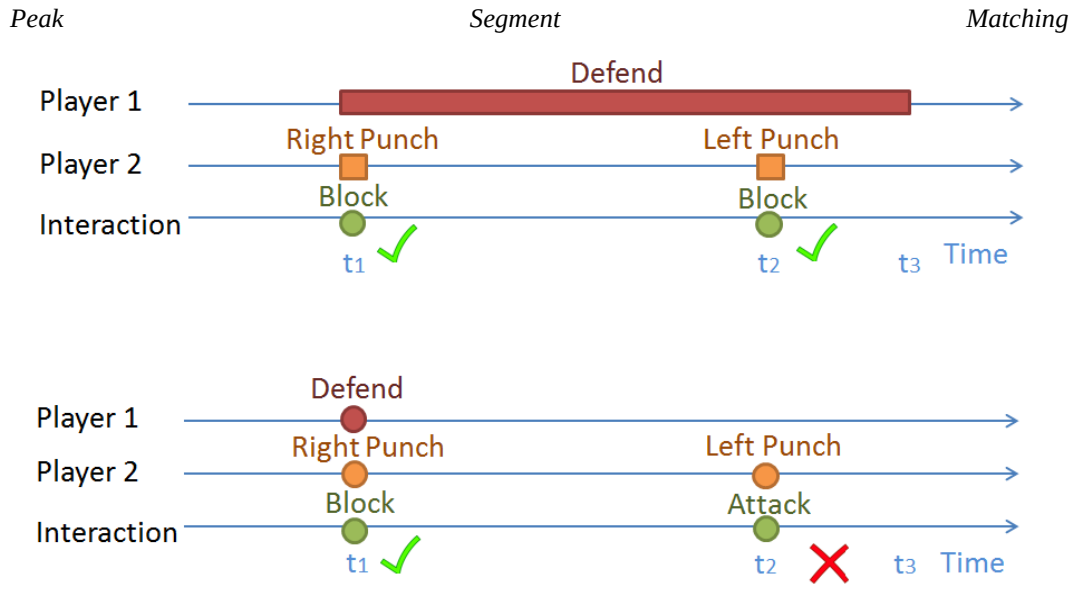

Figure 12 (Top) Interaction detection based on action segments which correctly detects actions with long duration. (Bottom) Interaction detection based on action points which only works if both actions occur at the same time and incorrectly detects interactions if an action has a long duration.

Some existing methods for online action recognition detect the action as a single point in time [9], [17] whereas others incorporate the duration of the action [14], [34]. The duration of the action is important for subsequently detecting interactions between multiple players in a sports game [4] and illustrated by Figure 12.

Peak key poses [14] were limited to detecting a single temporal point so we introduce a threshold τ to incorporate the duration of the peak. Similar to [14], [34] we introduce a threshold τ for action detection but instead of specifically learning a threshold for each action we learn a single threshold for all actions. Confining, the threshold to a single parameter reduces the time taken to adapt the model and this time will not increase even if more actions are considered, providing scalability to larger datasets.

The threshold τ and fragment size s are learnt on the training part of the target dataset by optimising the action point metric F1 [11] with our hierarchical template matching algorithm (summarised in Algorithm 3) but using the training data from the target dataset rather than the testing data.

3.3. Testing (target dataset)

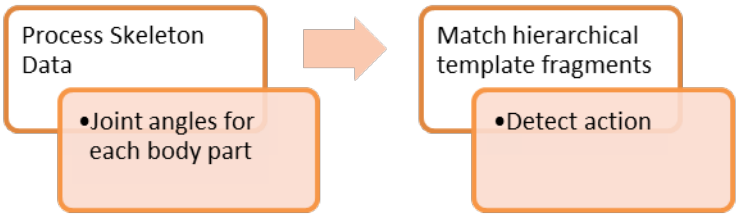

Figure 13 Testing overview which is performed on the target dataset

We propose a hierarchical template matching algorithm with a temporal sliding window for online action recognition (summarised in Algorithm 3). For each new frame the sliding window buffer is updated and compared with learnt exemplars. The minimum hierarchical DTW distance to the nearest neighbour is used to detect the action (see Figure 13).

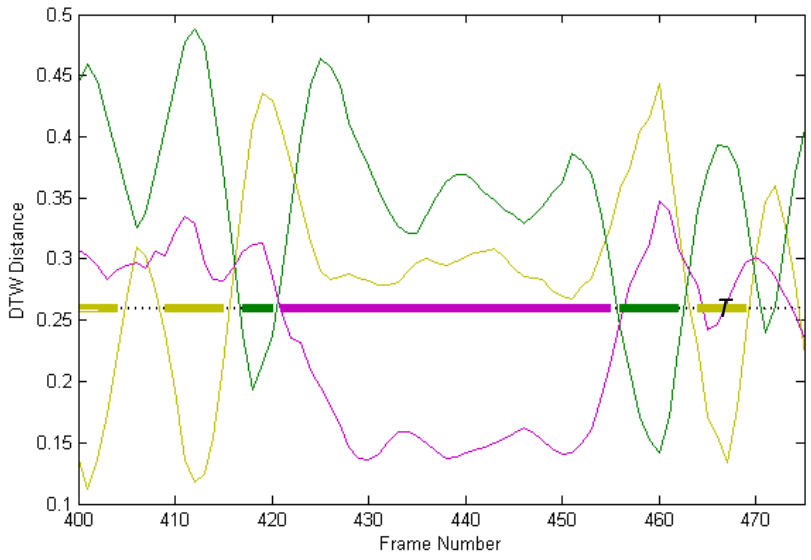

Figure 14 Normalised hierarchical DTW distances: the lowest value represents the most similar action, where this value is lower than the threshold τ it represents the detected action. The right punch is displayed in yellow, left punch displayed in green and the defend in magenta.

The hierarchical matching process is performed using DTW to ensure execution rate invariance. The normalised hierarchical DTW distances d^*, are recorded for each frame as illustrated in Figure 14. To detect actions in real-time we compare the lowest hierarchical DTW distance at each frame with a threshold τ. τ discriminates which pose fragments are most similar to the peak key pose

fragment. Therefore, whilst pose fragments are similar to the peak key pose fragment ($d^* \leq \tau$) the action is at its peak, as shown by the coloured segments on Figure 14. Before and after the peak, the pose fragments will be less similar ($d^* > \tau$) and therefore the action is not considered at its peak.

Algorithm 3 Online hierarchical template matching

Input: Given a set of testing poses from the target dataset $Z = \{z_i\}_{(i=1\ldots|Z|)}$, a set of action templates $K^a = \{k_j\}_{(j=1\ldots m)}$, with weights W^a, hierarchical peak key poses indices j^a, the fragment size s, and distance threshold τ:

1. For each testing pose, $i = 1{:}\,|Z|$
   - 1.1. Add the current test pose to the test fragment, $F^{\mathrm{Z}} = F^{\mathrm{Z}} \cup z_i$
   - 1.2. If $i \geq s$
     - 1.2.1. $F^{\mathrm{Z}} = F^{\mathrm{Z}} \setminus z_{i-s}$
   - 1.3. For each action, $a = 1{:}\,A$
     - 1.3.1. Extract the key pose fragment, $F^{\mathrm{K}} = (K^a, j^a)$ using Eq. 1
     - 1.3.2. Compute $HDTW(F^{\mathrm{Z}}, F^{\mathrm{K}}, W^a)$ using Eq. 4
   - 1.4. $d^* = \min_{a \in A} HDTW(F^{\mathrm{Z}}, F^{\mathrm{K}}, W_a)$
   - 1.5. If $d^* < \tau$
     - 1.5.1. $a^* = \arg\min_{a \in A} HDTW(F^{\mathrm{Z}}, F^{\mathrm{K}}, W^a)$
     - 1.5.2. Output "Action a^*"
   - 1.6. Else, output "No action"

One of the advantages of using clustering to identify peak poses is that the computational time is independent on the size of the training dataset, although it is linearly dependent on the number of actions. In case of many actions, a parallel implementation, i.e. one thread per action, would achieve real-time performance.

4. Experiments

In this section we present experiments to evaluate the ability of our online action recognition method to improve accuracy at low latency in complex scenarios.

4.1. Datasets

The performance of our algorithm is evaluated using publicly available datasets designed specifically for real time action recognition: G3D [10], MSRC-12 [9] and G3Di [4]. All datasets contain multiple actions in each sequence in a controlled indoor environment with a fixed camera, a typical setup for NUI applications. Both datasets provide sequences of skeleton data captured using the Kinect pose estimation pipeline at 30fps. However, G3D contains scripted actions which are temporally well separated whereas G3Di was captured using a gamesourcing approach where the users were recorded whilst playing computer games and consequently contains more complex actions which overlapping temporally. The G3Di also contains noisier skeleton data than G3D as there was interference from multiple Kinects during the recording, making it more realistic of a home scenario where there may be interference from the sunlight.

The G3D dataset contains 10 subjects performing 20 gaming actions grouped into seven categories. The fighting category was selected as it has the same actions as the G3Di boxing category although there are substantial variations in execution rate as well as personal style between these two datasets due to the different recording environments. The G3D fighting category contains five gaming actions: right punch, left punch, right kick, left kick and defend.

The MSRC-12 dataset comprises of 30 people performing 12 gestures. These gestures are categorized 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. Whereas metaphoric actions represent abstract concepts for manipulating a music player e.g. raise volume of the music. The dataset was obtained using different instruction modalities and the modality that produced the most accurate results was video + text so we will use this particular subset of the dataset.

The G3Di dataset contains 12 people split into 6 pairs. Each pair interacted through a gaming interface showcasing six sports: boxing, volleyball, football, table tennis, sprint and hurdles. Boxing is a competitive sport and the interactions can be decomposed by an action and counter action. The boxing actions were right punch, left punch and defend and the interactions between the players are shown in Table 1. The total number of action and interaction instances used for our experiments is shown in Table 2.

Table 1 Gaming interactions for the boxing scenarios in G3Di.

Sport	*Action*	*Counter Action*	*Interaction*
Boxing	Right Punch	Defend	Block
	Left Punch	Defend	Block
	Right Punch	Other	Attack
	Left Punch	Other	Attack
	Right Punch	Right Punch	Attack
	Right Punch	Left Punch	Attack
	Left Punch	Left Punch	Attack

Table 2 The total number of action and interaction instances used from each dataset

Dataset	*Action Classes*	*Interaction Classes*	*Subjects*	*Action / Interaction Instances*	*Frames*
G3D (Boxing)	5	NA	10	150	12,870
MSRC-12 (Iconic Gestures)	6	NA	10	502	4782
G3Di (Fighting)	3	2	12	317 + 257 = 574	6784

4.2. Skeleton Data

Joint angles are viewpoint and anthropometric invariant and can be generated in real-time with a pose estimation method [30]. More specifically, the skeleton poses are first normalised and then the three angles defining each joint position are computed and represented by a 4-D quaternion. The skeleton is parameterised as a high dimensional feature vector by concatenating quaternions for all joints. For each pose 13 quaternions are calculated so each feature vector has 52-dimensions (see [14] for more details).

4.3. Comparative Study

The following is a brief introduction of the comparison algorithms in our experiments:

- AdaBoost: AdaBoost has shown high accuracy and low latency for online action recognition [5], [17]. AdaBoost was trained on the source dataset and the parameters: the number of training frames around each peak pose the sliding smoothing window size were optimised on the training part of the target dataset and the method was evaluated on to the target testing data.
- Clustered Spatio-Temporial Manifolds (CSTM): is a state-of the art approach for low latency online action recognition [17]. CSTM was trained on the source dataset and the parameters: the template size and the stream size and the peak pose detector were optimised on the training part of the target dataset and the method was evaluated on to the target testing data.
- Hierarchical Transfer Segments (HiTS): The proposed method in this paper, a version of CSTM extended for transfer learning, allowing knowledge to be transferred from simple actions in a source dataset to complex actions in a target dataset by adapting the learnt models with a hierarchical pose representation. The parameters: peak segment matching threshold (τ=0.22) and fragment size ($s = 7$) were optimised on the training part of the target dataset and the method was evaluated on to the target testing data.

For all the above experiments we performed leave one-person out cross validation on the target dataset; each cross validation fold was trained on 11 subjects and tested on the remaining subject.

4.4. Performance Metrics

Evaluating of action recognition algorithms has previously been done in isolation, focusing historically on high accuracy and more recently also on low latency. However, in reality most actions form part of an interaction where the duration of the action is important. To test our proposed algorithm in a realistic context we employ the interaction detection and evaluation framework [4] and the action point metric [11] which is the most commonly used metric for online action recognition.

4.4.1. Action Point Metric

For evaluation we use an existing latency-aware performance metric for based on temporal anchors known as action points [11]. For a specified amount of latency (Δms) the action point F1-score determines whether a detection made at time t_p for action a is correct in relation to a ground truth action point at time t_g by using the following formula:

$$\Phi(t_p, t_g, \Delta) = \begin{cases} 1 & \text{if } (\,|\, t_g - t_p \,|\, \leq \Delta) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

For a specified amount of latency (Δms) the precision and recall are measured for each action and combined to calculate a single F-score.

$$\mathrm{F1-score}(a, \Delta) = 2\frac{\mathrm{prec}_a(\Delta)\ \mathrm{rec}_a(\Delta)}{\mathrm{prec}_a(\Delta) + \mathrm{rec}_a(\Delta)} \tag{6}$$

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

$$\text{Average F1-score}(A, \Delta) = \frac{1}{|A|} \sum_{a \in A} \mathrm{F1-score}(a, \Delta) \tag{7}$$

4.4.2. Interaction Detection Framework

The Interaction Detection Framework [4] enables online interaction recognition between multiple people by detecting their individual actions independently and

combining them by a set of interaction rules to infer the interaction. This modular approach is applicable for NUI and enables interaction between people that are not in the same physical location. Actions from different people are detected independently. At each frame, these detections are combined to infer the current interaction. The interaction rules include the valid combinations of actions (as depicted in Table 1) together with timing constraints. The action (*a*) and counter action (*ca*), are checked at each frame together with a timing constraint (*f*) to detect interactions in real time using Eq. 8. The timing constraint depends on the scenario, for example all the interactions in boxing are instant (f = 0), the action and counter action co-occur.

$$\psi\,(a_s, a_e, ca_s, ca_e) = \begin{cases} 1\ if\ (a_s + f \leq ca_e)\ and\ (ca_s \leq a_e + f) \\ 0\ otherwise \end{cases} \qquad (8)$$

Where s and e represent the start and end of the action segment respectively and $s \leq e$.

4.5. Results

Our method (HiTS) outperforms existing state of the art approaches for fast online action and interaction recognition, as shown in Figure 155. Both AdaBoost and CSTM show a significant drop in accurately detecting actions on the G3Di (Fighting) dataset in comparison with previously published results [17] on the G3D (Boxing) dataset. This is significant especially as the G3Di (Fighting) actions are a subset of the G3D (Boxing) actions but confirms our hypothesis that compound actions are more difficult to detect than multiple actions that are temporally well separated.

Additionally, we highlight the recognition accuracy for each category of action and interaction for a more detailed analysis of each method, as shown in Figure 16. A significant outcome is that even though CSTM [17] can detect all of the action categories it is unable to detect any interactions which are comprised of actions with duration, specifically the block interaction. In addition to showing the limitation of this approach it also highlights a weakness of the action point metric [11] which does not incorporate the duration of the action peak. Interaction detection is improved by our baseline method Peak Segment Matching (PSM) which instead of a binary decision for matching a peak key pose introduces a threshold which can detect the duration of the peak. The key contributions of this

paper are the hierarchical body model (HSM) and a transfer learning strategy (TSM). Individually, applied to our baseline method these contributions actually decrease the action and interaction recognition but together (HiTS) they form a powerful combination that significantly increases the action and interaction recognition, as shown in Figure 12. Intuitively, our hierarchical representation is only useful if adapted to the target dataset.

In this paper we are exclusively interested in action recognition approaches that are suitable for NUI applications. Research has shown that a delay of 100ms is not perceivable by the user [35]. Therefore, in this section we have only compared our method against online action recognition methods that are capable of fulfilling this requirement. Table 3 shows that all the methods we evaluated are capable of detecting actions with a low average latency of approx. 2 frames, which is equivalent to 66ms. We did not evaluate online action recognition methods with high latency (830-1500ms [16], 2000ms [14]) as they are better suited to other applications.

Table 3 A comparison of the average action latency

Method	Average Action Latency (frames)
AdaBoost	2.12
CSTM	2.00
PSM	1.60
TSM	1.41
HSM	1.94
HiTS	2.36

Figure 17 illustrates a typical failure case caused by noisy skeleton data at the action level resulting in an incorrect interaction to be inferred. The main limitation of our approach is that we only utilise the skeleton modality which is subject to interference from sunlight.

The dependency of the proposed transfer learning methodology on the amount of training data used from the target dataset is investigated. Specifically, Figure 18 demonstrates the action and interaction recognition performance (F1) for varying number of training subjects. The proposed method may achieve similar results to other competitor methods, i.e. around 0.6 and 0.4 F1 score for action and

interaction recognition respectively (see Figure 15) with almost half the training data from the target dataset, i.e. 6 subjects.

Regarding the template size *s* in theory it is possible to use different values in the matching process. However, in practice it was not computationally feasible to test all of these combinations so in our experiments we actually used a single parameter s which was learnt on the training part of the target dataset. Figure 19 shows how this parameter affects performance. This parameter does not model the duration of the action as the graph shows that even 3 frames (100ms) can accurately detect the action peak and overall performance is fairly consistent for higher values.

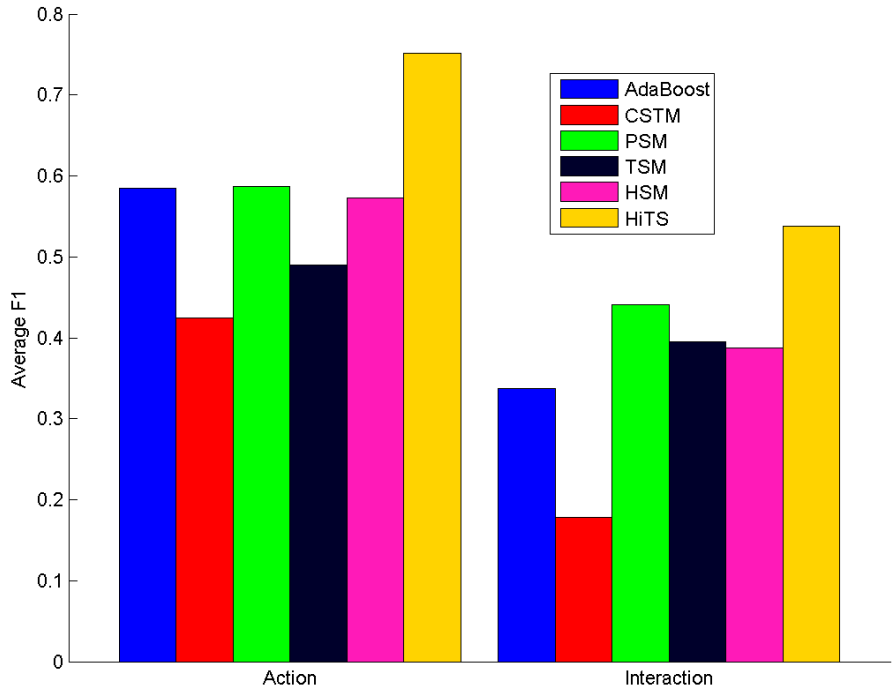

Figure 15 Performance comparison of the different approaches. Our method (HiTS) outperforms the others for both action and interaction detection.

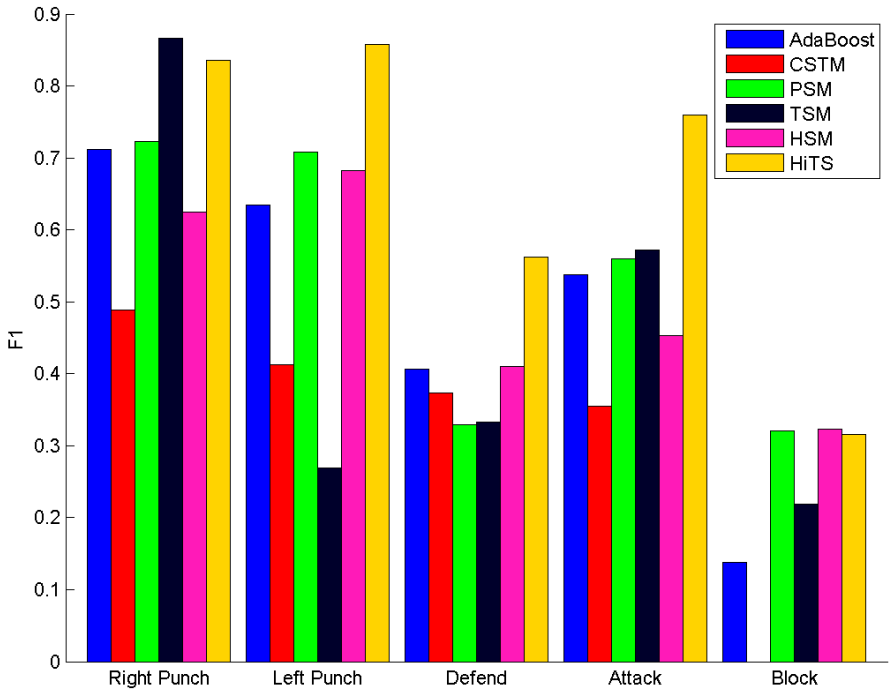

Figure 16 Action recognition results (left) and interaction recognition results (right) for each category of the G3Di (Fighting) dataset using different algorithms

Figure 17 Example of a typical failure case caused by noisy skeleton data. The colour image (right) shows that this is a block interaction but our algorithm detects an attack interaction as the defend action is not correctly detected due to incorrect skeleton data for the player on the left. This instance will be penalised twice by the action point metric, firstly a FP for the attack and secondly a FN for the block.

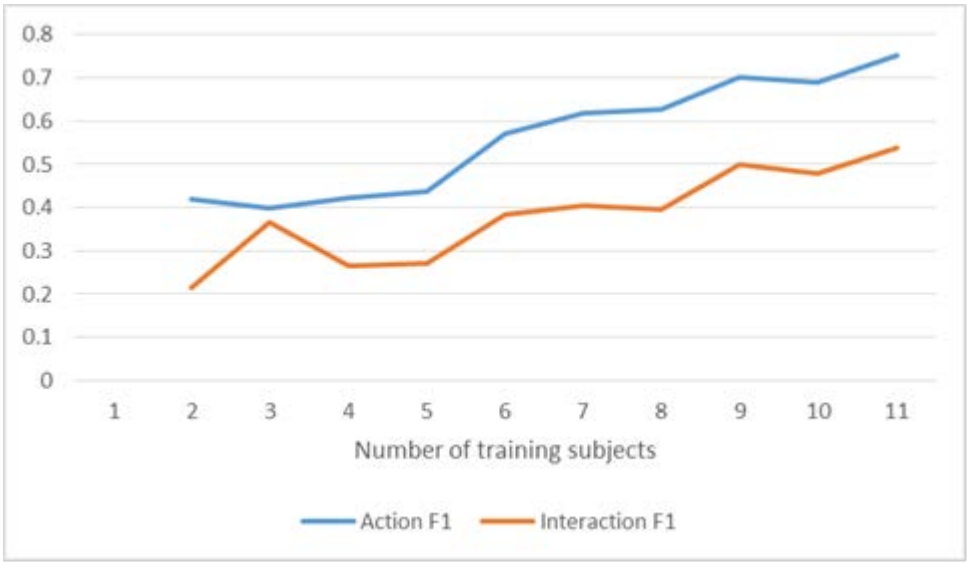

Figure 18 The relationship of the required number of training subjects and the obtained accuracy (F1 score) both for action and interaction analysis.

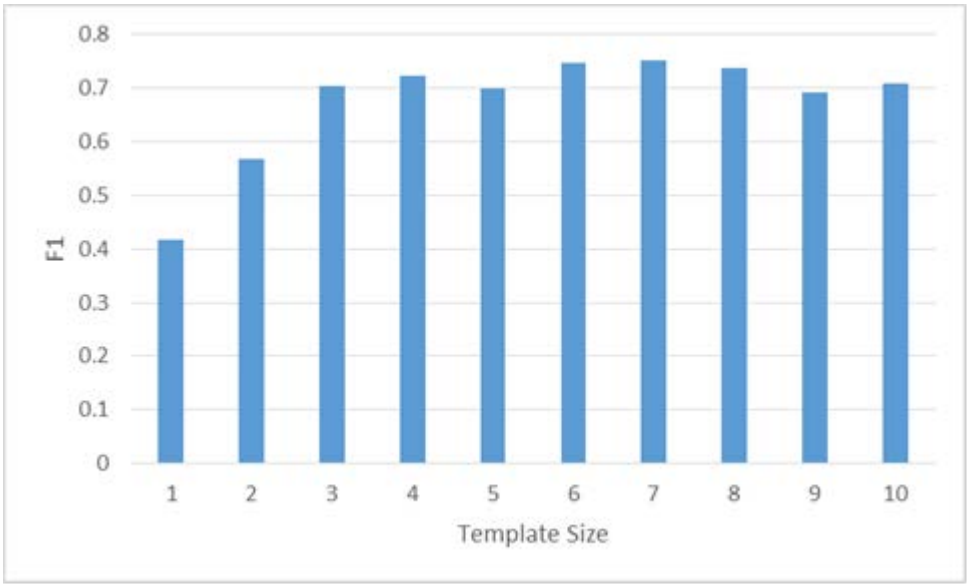

Figure 19 An example that indicates the relationship between the template size and the obtained accuracy (F1 score).

5. Conclusion

In this work we presented a novel hierarchical transfer learning algorithm for fast online action recognition. It overcomes the limitations of existing approaches by representing the human body hierarchically and learning the most discriminative body parts to detect compound actions. A transfer learning strategy was introduced to allow the tasks of action segmentation and whole body modelling to be performed on a related but simpler dataset. Combined with

hierarchical model adaptation on a more complex dataset to introduce independence between limbs and provide the flexibility to recognise poses that are not in the source dataset. Evaluation on a public target dataset that is more challenging and realistic than the source dataset shows our hierarchical transfer learning algorithm significantly increases performance at low latency. As the target dataset was recorded whilst users were actually playing a game the actions are more natural than subjects that are given instructions or restrictions and demonstrates the viability of our algorithm for use in real-world applications.

The limitation of our approach is that we only utilise the skeleton modality which is subject to interference from sunlight. Our future work is improve the robustness of our algorithm by fusing features from the depth or colour with our hierarchical skeleton features and evaluate its effectiveness using the G3Di multi-modal dataset.

References

[1] H. Liu, R. Feris, and M. Sun, "Benchmarking Datasets for Human Activity Recognition," in *Visual Analysis of Humans*, T. B. Moeslund, A. Hilton, V. Krüger, and L. Sigal, Eds. London: Springer London, 2011, pp. 411–427.

[2] A. Barbu, D. Barrett, and W. Chen, "Seeing is worse than believing: Reading people's minds better than computer-vision methods recognize actions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014, pp. 612–627.

[3] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004, vol. 3, pp. 32–36 Vol.3.

[4] V. Bloom, V. Argyriou, and D. Makris, "G3Di : A Gaming Interaction Dataset with a Real Time Detection and Evaluation Framework," in *European Conf. on Computer Vision Workshops (ECCVW)*, 2014.

[5] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos 'in the wild,'" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1996–2003.

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

[7] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action MACH A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'*, 2008, pp. 1–8.

[8] Z. Ma, Y. Yang, F. Nie, N. Sebe, S. Yan, and A. G. Hauptmann, "Harnessing Lab Knowledge for Real-World Action Recognition," *Int. J. Comput. Vis.*, vol. 109, no. 1–2, pp. 60–73, Apr. 2014.

[9] S. Fothergill, H. Mentis, P. Kohli, and S. Nowozin, “Instructing people for training gestural interactive systems,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 1737–1746.

[10] V. Bloom, D. Makris, and V. Argyriou, “G3D: A gaming action dataset and real time action recognition evaluation framework,” in *Computer Vision and Pattern Recognition Workshop (CVPRW), 2012 IEEE Conference on*, 2012, pp. 7–12.

[11] S. Nowozin and J. Shotton, “Action Points: A Representation for Low-latency Online Human Action Recognition,” *Techincal Rep.*, pp. 1–18, 2012.

[12] J. K. Aggarwal and M. S. Ryoo, “Human activity analysis: A review,” *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16:1–16:43, Apr. 2011.

[13] J. Han, L. Shao, D. Xu, and J. Shotton, “Enhanced computer vision with microsoft kinect sensor: a review.,” *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–34, Oct. 2013.

[14] A. Chaaraoui and F. Flórez-Revuelta, “Continuous Human Action Recognition in Ambient Assisted Living Scenarios,” in *First International Workshop on Enhanced Living EnvironMENTs (ELEMENT)*, 2014, pp. 1–8.

[15] V. Bloom, V. Argyriou, and D. Makris, “Dynamic Feature Selection for Online Action Recognition,” in *Human Behavior Understanding, Lecture Notes in Computer Science*, vol. LNCS, no. 8212, Switzerland: Springer International Publishing, 2013, pp. 64–76.

[16] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng, “Online Human Gesture Recognition from Motion Data Streams,” in *ACM Multi-Media 2013*, 2013, pp. 23–32.

[17] V. Bloom, D. Makris, and V. Argyriou, “Clustered Spatio-temporal Manifolds for Online Action Recognition,” in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2014, pp. 3963–3968.

[18] S. Pan and Q. Yang, “A survey on transfer learning,” … *Data Eng. IEEE Trans.*, vol. 22, no. 10, 2010.

[19] A. Farhadi and M. Tabrizi, “Learning to recognize activities from the wrong view point,” *Comput. Vision–ECCV 2008*, pp. 154–166, 2008.

[20] J. Liu and M. Shah, “Cross-view action recognition via view knowledge transfer,” *Comput. Vis. …*, 2011.

[21] D. Weinland, R. Ronfard, and E. Boyer, “Free viewpoint action recognition using motion history volumes,” *Comput. Vis. Image Underst.*, vol. 104, no. 2–3, pp. 249–257, Nov. 2006.

[22] L. Cao, Z. Liu, and T. Huang, “Cross-dataset action detection,” *Comput. Vis. pattern …*, 2010.

[23] Y. Wang, D. Tran, and Z. Liao, “Learning hierarchical poselets for human parsing,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1705–1712.

[24] Y. Tian, C. Zitnick, and S. Narasimhan, “Exploring the spatial hierarchy of mixture models for human pose estimation,” *Comput. Vision–ECCV 2012*, 2012.

[25] L. Raskin, M. Rudzsky, and E. Rivlin, “Using hierarchical models for 3D human body-part tracking,” *Image Anal.*, pp. 11–20, 2009.

[26] J. Darby, B. Li, N. Costen, D. Fleet, and N. Lawrence, “Backing Off: Hierarchical Decomposition of Activity for 3D Novel Pose Recovery.,” *BMVC*, 2009.

[27] A. Moutzouris, J. Martinez-del-Rincon, J.-C. Nebel, and D. Makris, “Efficient tracking of human poses using a manifold hierarchy,” *Comput. Vis. Image Underst.*, Oct. 2014.

[28] L. Han, X. Wu, W. Liang, G. Hou, and Y. Jia, “Discriminative human action recognition in the learned hierarchical manifold space,” *Image Vis. Comput.*, vol. 28, no. 5, pp. 836–849, May 2010.

[29] Y. Song, L. Morency, and R. Davis, “Action recognition by hierarchical sequence summarization,” *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 3562 – 3569, 2013.

[30] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, *Real-time human pose recognition in parts from single depth images*, vol. 2, no. 3. IEEE, 2011, pp. 1297–1304.

[31] M. Lewandowski, D. Makris, and J. Nebel, “Temporal Extension of Laplacian Eigenmaps for Unsupervised Dimensionality Reduction of Time Series,” in *International Conference on Pattern Recognition*, 2010, pp. 161 – 164.

[32] T. Kanungo, D. M. Mount, N. S. Netanyahu, A. Y. Wu, and C. D. Piatko, “A Local Search Approximation Algorithm for k-Means Clustering,” *Spec. Issue 18th Annu. Symp. Comput. Geom. - SoCG2002*, vol. 28, no. 2–3, pp. 89–112, 2003.

[33] P. Senin, “Dynamic Time Warping Algorithm Review,” USA, 2008.

[34] I. Kviatkovsky, E. Rivlin, and I. Shimshoni, “Online action recognition using covariance of shape and motion,” *Comput. Vis. Image Underst.*, vol. 129, pp. 15–26, Dec. 2014.

[35] S. K. Card, G. G. Robertson, and J. D. Mackinlay, “The information visualizer: An information workspace,” in *Proc. ACM CHI*, 1991, pp. 181–188.

[36] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake, Real-Time Human Pose Recognition in Parts from a Single Depth Image, IEEE CVPR 2011