Symmetry **2015**, *7*, 1670-1698; doi:10.3390/sym7041670

OPEN ACCESS

symmetry

ISSN 2073-8994

www.mdpi.com/journal/symmetry

Article

Application of Assistive Computer Vision Methods to Oyama Karate Techniques Recognition

Tomasz Hachaj [1,*], Marek R. Ogiela [2] and Katarzyna Koptyra [2]

[1] Institute of Computer Science and Computer Methods, Pedagogical University of Krakow, 2 Podchorazych Ave, Krakow 30-084, Poland

[2] Cryptography and Cognitive Informatics Research Group, AGH University of Science and Technology, 30 Mickiewicza Ave, Krakow 30-059, Poland; E-Mail: mogiela@agh.edu.pl (M.R.O.); kkoptyra@agh.edu.pl (K.K.)

* Author to whom correspondence should be addressed; E-Mail: tomekhachaj@o2.pl; Tel.: +48-126-626-322; Fax: +48-126-626-166.

Academic Editor: Sergei Odintsov

Received: 18 June 2015 / Accepted: 31 August 2015 / Published: 24 September 2015

Abstract: In this paper we propose a novel algorithm that enables online actions segmentation and classification. The algorithm enables segmentation from an incoming motion capture (MoCap) data stream, sport (or karate) movement sequences that are later processed by classification algorithm. The segmentation is based on Gesture Description Language classifier that is trained with an unsupervised learning algorithm. The classification is performed by continuous density forward-only hidden Markov models (HMM) classifier. Our methodology was evaluated on a unique dataset consisting of MoCap recordings of six Oyama karate martial artists including multiple champion of Kumite Knockdown Oyama karate. The dataset consists of 10 classes of actions and included dynamic actions of stands, kicks and blocking techniques. Total number of samples was 1236. We have examined several HMM classifiers with various number of hidden states and also Gaussian mixture model (GMM) classifier to empirically find the best setup of the proposed method in our dataset. We have used leave-one-out cross validation. The recognition rate of our methodology differs between karate techniques and is in the range of 81% ± 15% even to 100%. Our method is not limited for this class of actions but can be easily adapted to any other MoCap-based actions. The description of our approach and its evaluation are the main contributions of this paper. The results presented in this paper are effects of pioneering research on online karate action classification.

Keywords: assistive computer vision; sport actions recognition; actions segmentation; Gestures Description Language; hidden Markov models; unsupervised learning; Oyama karate

1. Introduction

Actions recognition methods enable recognition of short human movement recordings. In contrary to gesture recognition they are focused on full body analysis rather than hands or palms. Depending on application we can detect both everyday events (like for example shelf opening, sitting, or falling) and more specialized sport actions. Online action recognition methods are even more complicated because they have to deal with real-time (or nearly real time) streams of data that should be analyzed under time constraints. In those cases incoming data is often partitioned (segmented) into smaller sequences which might contain events that are targets of further analysis. In following sections we will discuss state-of-the-art methods that are used for action recognition. We will also present computer approaches that are used for the recognition and quality evaluation of karate techniques. We will also present relatively new methods of visual data acquisition with multimedia depth sensors.

1.1. Approaches to Actions Recognition

Action recognition tasks are solved with many popular and efficient classifiers. The choice of classification method is often determined by feature selection which frequently utilizes derivatives of body joint positions. In this state-of-the-art review we will concentrate manly on body joint-based features. In paper [1] authors consider the problem of joint segmentation and classification of various common-live actions in the framework of conditional random field (CRF) models. Model training is conducted by means of an efficient likelihood maximization algorithm, and inference is based on the familiar Viterbi algorithm. In paper [2] authors propose method to recognize human actions using 3D skeleton joints recovered from 3D depth data of RGBD cameras. Authors design an action feature descriptor for action recognition based on differences of skeleton joints, *i.e.*, EigenJoints which combine action information including static posture, motion property, and overall dynamics. Accumulated Motion Energy (AME) is then proposed to perform informative frame selection, which is able to remove noisy frames and reduce computational cost. Non-parametric Naive-Bayes-Nearest-Neighbor (NBNN) is used to classify multiple actions. In study [3] a Histogram-of-Oriented-Velocity-Vectors (HOVV) descriptor for skeleton data is introduced. It is a scale-invariant, speed-invariant, and length-invariant descriptor for human actions. The skeleton sequence using 2D spatial histogram capturing the distribution of the orientations of velocity vectors of the joint in a spherical coordinate system. To classify actions represented by HOVV descriptor k-nearest neighbor classifier, Support Vector Machines classifier, and Extreme Learning Machines are used. In paper [4] authors develop a graph-based method to align two dynamic skeleton sequences. Authors quantize the skeleton pose space in order to decrease redundancy in the temporal domain. They used k-means clustering for extracting the key poses. Given a sequence of keys poses they sample the signal and assign each skeleton at a time instant to its closest cluster center, *i.e.*, its closest key pose, and obtain an abstract representation for each skeleton sequence. Work [5] introduces a method for real-time gesture recognition from a noisy skeleton stream. Each pose is

described using an angular representation of the skeleton joints. Those descriptors serve to identify key poses through a Support Vector Machine multi-class classifier with a tailored pose kernel. The gesture is labeled on-the-fly from the key pose sequence with a decision forest, which naturally performs the gesture time control/warping and avoids the requirement for an initial or neutral pose. In [6] actions are modeled as a set of weighted dynamical systems associated to different model variables. Authors use time-delay embedding on the time series resulting in the evolution of model variables along time to reconstruct phase portraits of appropriate dimensions. These phase portraits characterize the underlying dynamical systems. A distance to compare trajectories is proposed within the reconstructed phase portraits. These distances are used to train SVM models for action recognition. In paper [7] authors propose a hierarchical model for action recognition. To handle confusing motions, a motion-based grouping method is proposed, which can assign each video a group label, and then for each group, a pre-trained classifier is used for frame-labeling. The final action label is obtained by fusing the classification to its frames, with the effect of each frame being adaptively adjusted based on its local properties. To achieve online real-time performance and suppressing noise, bag-of-words is used to represent the classification features. In [8] the action is represented in a 15-dimensional space using a covariance descriptor of shape and motion features—spatiotemporal coordinates and optical flow of pixels belonging to extracted silhouettes. In order to enable online action classification, authors used incremental covariance update and the on demand nearest neighbor classification. In [9] a method for action recognition is introduced where skeletal data are initially processed in order to obtain robust and invariant pose representations and then vectors of dissimilarities to a set of prototype actions are computed. The task of recognition is performed in the dissimilarity space using sparse representation. In paper [10] authors propose a gesture recognition method that integrates rough set theory with the longest common subsequence method to classify free-air gestures, for natural human-computer interaction. In this paper, gestures are encoded in orientation segments which facilitate their analysis and reduce the processing time. To improve the accuracy of gesture recognition on ambiguous gestures, rough set decision tables conditioned on the longest common subsequences is generated; the decision tables store discriminative information on ambiguous gestures. In paper [11] authors propose a classifier called the gesture description language (GDL). The very heart of our approach is an automated reasoning module. It performs forward chaining reasoning (like a classic expert system) with its inference engine every time a new portion of data arrives from the feature extraction library. All rules of the knowledge base are organized in GDL scripts having the form of text files that are parsed with a context-free grammar. Also dynamic time warping (DTW) is a well-known technique to find an optimal alignment between two given (time-dependent) sequences under certain restrictions [12]. Intuitively, the sequences are warped in a nonlinear fashion to match each other. The usage of this method has been reported in multiple papers. In [13] DTW computes a dissimilarity of movement measure by time-warping the sequences on a per sample basis by using the distance between the current reference and test sequences. The method [14] is based on histograms of action poses, extracted from MoCap data that are computed according to Hausdorff distance. The histograms are then compared with the Bhattacharyya distance and warped by a dynamic time warping process to achieve their optimal alignment. Paper [15] presents a system for ensuring home-based rehabilitation using a DTW algorithm and fuzzy logic. Paper [16] introduces a basic frame for a rehabilitation motion practice system which detects 3D motion trajectory and proposes a 3D motion matching algorithm. The similarity of trajectories is measured based on the signature using an

alignment method such as dynamic time warping, continuous dynamic time warping, or longest common sub-sequence (LCSS) method. Because actions might be interpreted as multidimensional signals also some methods that are commonly used for acoustic signals classification might be also applied for action recognition [17,18]. Beside body joint-based features actions can be represented in different manners. For example in work [19] authors use Hidden Markov Model (HMM) for analyzing human spatial behavior (proxemics) motivated by metrics used in the social sciences. They use two different feature representations—physical and psychophysical—to train HMM to recognize spatiotemporal patterns.

1.2. Recognition of Actions in Continuous Streams

The recognition of actions in continuous streams has been a target of extensive research. In work [20] authors provide a discriminative framework for online simultaneous segmentation and classification of visual actions, which deals effectively with unknown sequences that may interrupt the known sequential patterns. To this end, they employ Hough transform to vote in a 3D space for the begin point, the end point and the label of the segmented part of the input stream. An SVM is used to model each class and to suggest putative labeled segments on the timeline. To identify the most plausible segments among the putative ones authors apply a dynamic programming algorithm, which maximizes an objective function for label assignment in linear time. Most researches on human activity recognition do not take into account the temporal localization of actions. In paper [21], a new method is designed to model both actions and their temporal domains. This method is based on a new Hough method. Experiments are performed to select skeleton features adapted to this method and relevant to capture human actions. In paper [22] authors propose a method that is based on a discriminative temporal extension of the spatial bag-of-words model that has been very popular in object recognition. The classification is performed robustly within a multi-class SVM framework whereas the inference over the segments is done efficiently with dynamic programming.

1.3. Application of Deep Learning in Action Recognition

In paper [23] authors develop a 3D convolutional neural networks model for action recognition. This model extracts features from both the spatial and the temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. The developed model generates multiple channels of information from the input frames, and the final feature representation combines information from all channels. To further boost the performance, authors propose regularizing the outputs with high-level features and combining the predictions of a variety of different models. In [24] authors propose tiled convolution neural networks (Tiled CNNs), which use a regular "tiled" pattern of tied weights that does not require that adjacent hidden units share identical weights, but instead requires only that hidden units k steps away from each other to have tied weights. By pooling over neighboring units, this architecture is able to learn complex invariances (such as scale and rotational invariance) beyond translational invariance. Further, it also has much of CNNs' advantage of having a relatively small number of learned parameters (such as ease of learning and greater scalability). Paper [25] shows how to use "complementary priors" to eliminate the explaining-away effects that make inference difficult in densely connected belief nets that have many hidden layers. Using complementary priors, authors derive a fast, greedy algorithm that can learn deep, directed belief networks one layer at a

time, provided the top two layers form an undirected associative memory. The fast, greedy algorithm is used to initialize a slower learning procedure that fine-tunes the weights using a contrastive version of the wake-sleep algorithm. After fine-tuning, a network with three hidden layers forms a generative model of the joint distribution of handwritten digit images and their labels.

1.4. Application of Depth Sensors in Actions Recognition

Analysis of human behavior through visual information processing and classification has been a highly active research topic in the computer vision community. This was previously achieved via images from a conventional camera, however recently depth sensors have made a new type of data available [26]. Modern off-the-shelf multimedia depth sensors quickly gain popularity on commercial market. Among most popular products we can mention, the Microsoft Kinect sensor and its successor Kinect 2. They have been applied not only for gaming but to various important branches of technique and science. Among those we can mention natural user interfaces and robotics [27], education [28], medicine [29], zootechnics [30], intelligent home technology [31,32] nonverbal behavior analysis [33]. As information and communication technology continues to evolve, body sensory technologies provide learning designers new approaches to facilitating learning in an innovative way [34]. For example paper [35] presents concept of a novel Virtual Reality Educational System that utilizes action recognition technology in the role of natural user interface.

Latest research suggests that the Microsoft Kinect can validly assess kinematic strategies of postural control and can be applied as motion capture systems with limited accuracy [36]. However we must remember that data captured by this device is noisy, and sometimes even lost or shifted, especially around the edges of the depth [37].

1.5. Recognition and Quality Evaluation of Karate Techniques

Karate techniques analysis is a very challenging task due to the high speed of skilled martial artists' movements. Due to many years of intensive training professional martial artists gain speed, agility, and flexibility that is unattainable for a typical untrained person. Those factors are additional challenge to multimedia motion capture hardware and processing software that is often optimized to motion range that is available for a typical person. In order to capture karate techniques, one often uses high-end motion capture hardware or prepare dedicated hardware installations [38]. The expensive motion capture systems are capable to capture precision movement data that can be even used for example to estimate skill level. For example, in [39] authors use Spatiotemporal Morphable Models that are based on linearly combining the movement trajectories of prototypical motion patterns in space-time. Linear combinations of movement patterns are defined on the basis of spatiotemporal correspondences that are computed by dynamic programming. The authors evaluated the approach on movements of seven actors performing the karate kata "Heian Shodan".

It is also possible to use standard camera video to perform action recognition of karate techniques. In paper [40] authors propose a recognition system based on the use of local spatiotemporal features from Histographic methods, as those extracted by Histograms of Oriented Gradient (HOG) and Histograms of Optical Flow (HOF), while the reduction of the problem's dimensionality is done by applying Principal Components Analysis (PCA). For actions recognition HMM are used. The system is tested upon a database comprising sequences of shotokan karate movements (katas).

More lately, consumer level hardware has been applied for karate data acquisition and processing. Work [41] aims at automatically recognizing sequences of complex karate movements and giving a measure of the quality of the movements performed. In this work authors use skeleton features generated by low cost hardware. The proposed system is constituted by four different modules: skeleton representation, pose classification, temporal alignment, and scoring. The proposed system is tested on a set of different punch, kick, and defense karate moves. The vocabulary of key poses is obtained through a k-means clustering of the moves considered, and each cluster centroid becomes a vocabulary key pose. The classification is done with a multi-class SVM, which recognizes key poses with a one-versus-all approach. Temporal Alignment for action's scoring is done with DTW algorithm. The optimum is obtained minimizing the sum of cumulative distances between the aligned sequences. Authors in [42] proposed a calibration procedure that enables integration of skeleton data from set of tracking devices into one skeleton. The test set for our methodology was recordings of seven various Okinawa Shorin-ryu Karate techniques performed by a black belt instructor. The classification method was GDL.

We can notice that very few state-of-the-art papers deal with the problem of computer analysis and recognition of advanced karate techniques. In our opinion it might be caused by two main factors. The first is that it is difficult to gather sufficient amount of data with enough good quality for further processing, because martial artists have to be enough qualified to performed techniques correctly which requires years of practicing. The second is, that until recent years there was not much need for that type of evaluation because it was hardly possible to apply the results in practice due to the high costs of MoCap hardware and data processing software.

1.6. The Motivation for This Paper

The state-of-the art report we have presented proves that gestures and action recognition techniques are nowadays very popular and a challenging subject of research. In the last few years real-time online action recognition became the aim of research that can be quickly deployed by an industry. It is possible because nowadays personal computers have enough computational power to process large amounts of incoming visual data in real time. Also relatively cheap MoCap hardware with open programming libraries allows introducing movement analysis and recognition to everyday life.

In this paper we propose a novel algorithm that enables online action segmentation and classification. The algorithm enables segmentation of incoming MoCap data stream movements sequences that are later processed by classification algorithm. The segmentation is based on Gesture Description Language classifier that is trained with an unsupervised learning algorithm. The classification is performed by continuous density forward-only hidden HMM classifier which is stable and reliable approach for classification of presegmented recordings. Our methodology was evaluated on a unique dataset consisting of MoCap recordings of six Oyama karate martial artists including multiple champion of Kumite Knockdown Oyama karate. The dataset consists of 10 classes of actions and included dynamic actions of stands, kicks, and blocking techniques. Total number of samples was 1236. The complex karate movement classification is a challenging task due to the speed of body movements. From the other hand movement patterns are highly repetitive because they are practiced for many years by skilled martial artists. Those two facts make karate techniques classification tasks reliable tests of classifiers and recognition systems potentials. Also, nowadays there is a growing interest in thew commercial market for

solutions that are capable of using martial arts techniques recognition in computer entertainment and coaching systems. We have examined several HMM classifiers with various number of hidden states and also Gaussian mixture model (GMM) classifier to empirically find the best setup of proposed methods on our dataset. We have used leave-one-out cross validation. Our method is not limited for this class of actions but can be easily adapted to any other MoCap-based actions. The description of our approach and its evaluation are main contributions of this paper. The results presented in this paper are effects of pioneering research on online karate action classification.

2. Materials and Methods

In this section we will present the methodology we developed and used to create the online actions segmentation and classification algorithm.

2.1. Pose Representation

The Kinect 2 SDK system we used is capable of three-dimensional tracking the so-called user skeleton composed of 25 body joints. However the joints-based representation is dependent on the relative position of the user to the sensor and body proportion. Papers [4,41,43] propose to use features based on angles between vectors derived from selected body joints. The angle-based features are invariant to relative position to camera, body proportion, and are normalized to range $[0,\pi)$. For purposes of this research we have selected the slightly modified set of all features for previously mentioned papers. We have not used joints situated in the middle part of the body (*spine middle* joint of Kinect 2 SDK) because this part of body is often covered by limbs. In Figure 1 we have presented directions of vectors that were used to define our features set. The blue arches indicate the angles we considered in our features set. It is the smaller of two possible angles on the plane defined by red segments. Additionally we have consulted the selection of our features with a professional black-belt Oyama karate trainer in order to determine if it is sufficient for karate techniques descriptions. Because of MoCap hardware limitations it is impossible to correctly track positions and orientations of hands, fingers, and feet however we can observe wrists and ankles.

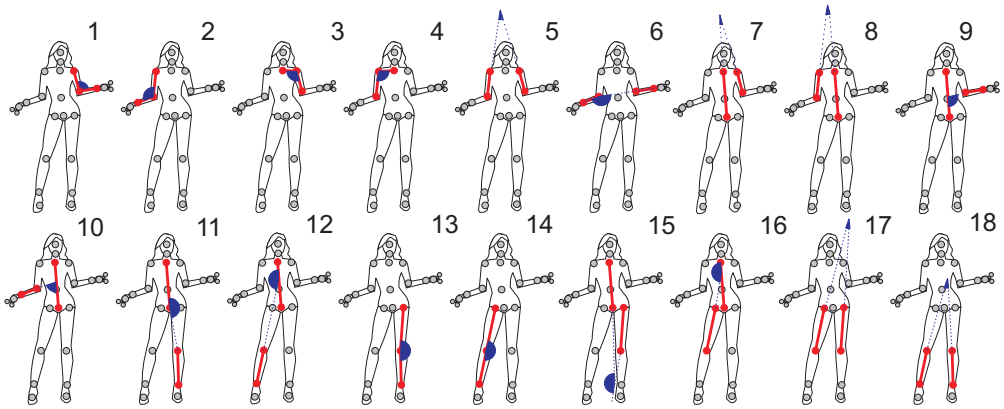

Figure 1. This figure presents directions of vectors that were used to define our features set. The blue arches indicate the angles we considered in our features set.

In order to calculate angles between two vectors we can use following procedure [44]. Let us assume that vectors $\overline{X}', \overline{Y}'$ are normalized vectors $\overline{X}, \overline{Y}$:

$$\overline{X}' = \frac{\overline{X}}{\left\|\overline{X}\right\|} \text{ and } \overline{Y}' = \frac{\overline{Y}}{\left\|\overline{Y}\right\|} \tag{1}$$

Angles are finally recalculated to degrees. The graphical illustration of Equation (2) is in Figure 2.

$$\begin{aligned} &\text{If } \theta \in \left[0, \frac{\pi}{2}\right] \text{ then } \sin\frac{\theta}{2} = \frac{\left\|\overline{X}' - \overline{Y}'\right\|}{2}; \\ &\text{If } \theta \in \left(\frac{\pi}{2}, \pi\right] \text{ then } \sin\frac{\pi - \theta}{2} = \frac{\left\|-\overline{X}' - \overline{Y}'\right\|}{2}. \end{aligned} \tag{2}$$

And eventually:

$$\angle\overline{X}, \overline{Y} = \theta = \begin{cases} 2 \cdot \arcsin\left(\frac{\left\|\overline{X}' - \overline{Y}'\right\|}{2}\right) & \text{if} \quad \overline{X}' \circ \overline{Y}' \geq 0 \\ \pi - 2 \cdot \arcsin\left(\frac{\left\|-\overline{X}' - \overline{Y}'\right\|}{2}\right) & \text{if} \quad \overline{X}' \circ \overline{Y}' < 0 \end{cases} \tag{3}$$

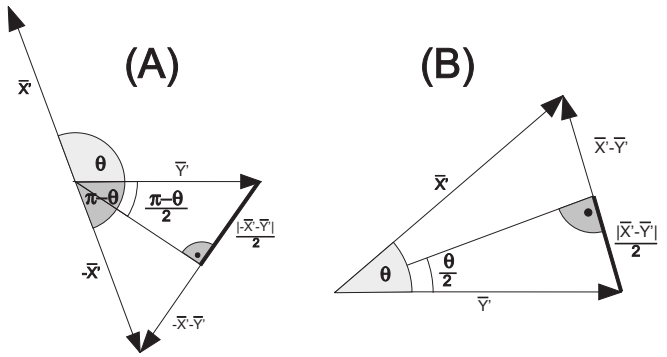

Figure 2. This figure presents a possible way of calculation angle between two normalized vectors. In case (**A**) angle $\theta \in \left(\frac{\pi}{2}, \pi\right]$; in case (**B**) $\theta \in \left[0, \frac{\pi}{2}\right]$.

2.2. Continuous-Density Hidden Markov Models Classifier

As we already mentioned hidden Markov models have found great use in actions and gestures recognition. For our need we have utilized first order continuous HMM where probability of at time $t + 1$ depends only on the state at t and each hidden state emits the continuous function with Gaussian distribution. We have evaluated five continuous Gaussian density hidden Markov model classifiers on our dataset. All of them had forward-only architecture and differed in number of hidden states from one (which was practically a Gaussian mixture model) to five. GMM is unable to model the order of movements in action however

it can be used to check the sole descriptive power of selected features. The four-state forward-only HMM architecture can be seen in Figure 3 and its transition matrix looks as follows:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \quad (4)$$

where a_{ij} are transition probabilities.

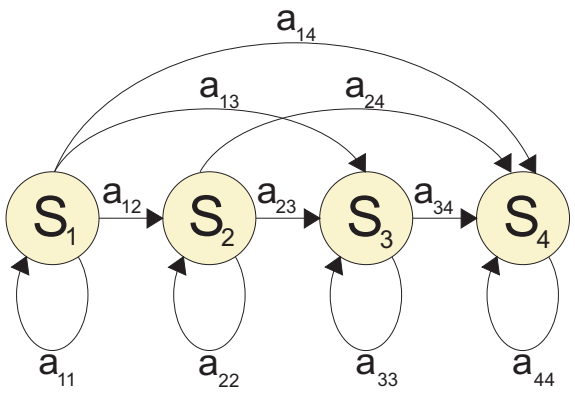

Figure 3. This figure presents a four-state forward-only HMM.

We have used the Baum–Welch algorithm to estimate the HMM parameters. In order to find the probability of observed sequences we used a forward algorithm. In order to create an HMM classifier that decides which *n*-classes of a given signal belongs we composed an HMM (the same as number of classes) and parameters of each HMM were estimated on exemplars of signals from different classes. When an unknown signal has been examined it was assigned to the class for which HMM corresponding to this class had the largest probability to produce the sequence. Due to this fact, each signal has been assigned to a single class.

The Baum–Welch algorithm is a particular case of a generalized expectation-maximization (GEM) algorithm [45]. It can compute maximum likelihood estimates and posterior mode estimates for the parameters (transition and emission probabilities) of an HMM, when given only emissions as training data.

The algorithm has two steps:

- Calculating the forward probability and the backward probability for each HMM state;
- On the basis of this, determining the frequency of the transition-emission pair values and dividing it by the probability of the entire string.

This amounts to calculating the expected count of the particular transition-emission pair. Each time a particular transition is found, the value of the quotient of the transition divided by the probability of the entire string goes up, and this value can then be made the new value of the transition.

In our case, while using normal continues HMM the implementations of this training method use the observations in the training data and the probability matrix to update the probability distributions of symbol emissions.

2.3. Unsupervised—Learned Actions Segmentation and Classification with Gestures Description Language

Gesture Description Language (GDL) is an advanced classifier that enables syntactic description and real time recognition of full-body gestures and movements [11]. Gestures are described in dedicated computer language named Gesture Description Language script (GDLs). GDL is formal context-free grammar. GDL uses an inference engine that performs forward-chaining reasoning on those rules. If a rule is satisfied, its conclusion is to memorize in the memory stack. A memory stack also holds all features set that was used so far and time stamps values that enable to calculate how much time has passed between each MoCap data acquisition. A single rule or several rules together with features might define a static body position (so called key frame) of a gesture. A gesture consists of sn ordered sequence of key frames. A key frame is typically defined as the relative position of body joints. There are many possibilities to define that position. The most useful approach is to use angles between vectors defined by body joints. That is because this type of description is nearly invariant to body rotation and proportion. Sequences of key frames are also defined in a rule. If a rule is satisfied it means that user's body is in particular position or he or she had made some gesture.

In paper [43] the unsupervised learning method for generation of GDLs has been introduced. Input data for the methods are MoCap recordings of user that periodically performs action to be recognized. There are also initial features defined in GDLs that will be used to transform input signals from cartesian frame coordinates into feature space. The unsupervised learning method uses k-means clustering of input data in feature space. The center of clusters together with standard deviation of elements from a cluster and epsilon value creates definition of R-GDL features. Those R-GDL features are used in the definition of R-GDL rules that define keyframes. This can be easily explained: if particular part of the movement pays important role in identification of gesture it should be "more visible" in feature space and create a separate cluster. The order of key frames is identify by finding most frequent n-gram. The concept of keyframes based classification is presented in Figure 4. For example let us assume that mae-geri is described by two keyframes: initial and end frame that are very similar (*keyframe 1*) and middle frame (*keyframe 2*). In the moment when in memory stack of GDL exists the sequence of keyframes *keyframe 1–keyframe 2–keyframe 1* the mae-geri action is detected.

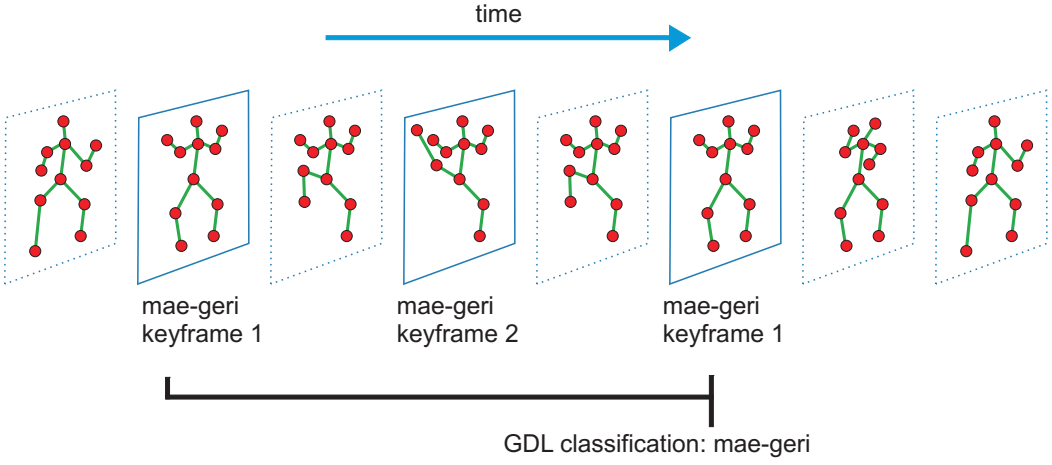

Figure 4. The concept of keyframe-based classification.

The GDL classifier trained with R-GDL method works as follows. Let us assume that we have time varying m-dimensional signal $F_{[t_i..t_j]}$ sampled in discrete time moments $t_i, t_{i+1}, \ldots, t_j$ (i < j).

$$F_{[t_i..t_j]} = \left[\overline{f_{ti}}, \ldots, \overline{f_{tJ}}\right] \tag{5}$$

And $\overline{f_{ta}} \in \mathbb{R}^m$ where $a \in [i, j]$.

A signal sample belongs to cluster $C_b(\overline{\mu_b}, \overline{\sigma_b}, \overline{\varepsilon_b})$ when:

$$\overline{f_{ta}} \subset C_b \Leftrightarrow \left|\overline{f_{ta}} - \overline{\sigma_b}\right| \lesssim \overline{\mu_b} + \overline{\varepsilon_b} \tag{6}$$

where $\overline{\mu_b}$ is a mean vector (center of cluster), $\overline{\sigma_b}$ is a standard deviation of cluster's elements and $\overline{\varepsilon_b}$ together with $\overline{\sigma_b}$ determines size of a cluster. $\lesssim$ means that all coordinates of vector in the left side have a lower value than the coordinates corresponding to them in the right side (both vectors have the same dimension). In Equation (6) the vector distance is computed using L1 norm.

GDL classifies signal $F_{[t_i..t_j]}$ to class $\mathbb{C}_c$ when:

$$GDL\left(F_{[t_i..t_j]}\right) = \mathbb{C}_c \Leftrightarrow \begin{cases} \overline{f_{tl1}} \subset C_{c1} \wedge \ldots \wedge \overline{f_{tln}} \subset C_{cn}, tl_1 < \cdots < tl_n \\ tl_n - tl_{n-1} < tc_n \ \wedge \ldots \wedge \ tl_1 - t_i < tc_1 \end{cases} \tag{7}$$

where $[C_{c1}, \ldots, C_{cn}]$ are n-element set of clusters (key frames) of an action, $[tc_1, \ldots, tc_n]$ are time constraints assigned to each key frame and $[tl_1, \ldots, tl_n]$ are moments of time when particular samples belongs to a given cluster.

In other words in GDL notation a class $\mathbb{C}_c$ is represented as the ordered set of key frames $[C_{c1}, \ldots, C_{cn}]$. A signal Equation (5) belongs to class $\mathbb{C}_c$ if and only if we can find signal samples $\left[\overline{f_{tl1}}, \ldots, \overline{f_{tln}}\right]$ that satisfies Equation (6) under time constraints $[tc_1, \ldots, tc_n]$. That means in the last tl_1 seconds ($tl_1 - t_i < tc_1$) there is a signal sample $\overline{f_{tl1}}$ that belongs to cluster C_{c1}, and in the last tl_2 seconds ($tl_2 - tl_1 < tc_2$) there is a signal sample $\overline{f_{tl2}}$ that belongs to cluster C_{c2}, *etc.* (compare with Equation (7)).

We can simply prevent the situation that GDL classifiers detect a particular action multiple times in neighboring frames by adding a constraint that the next action of a same type cannot appear in time shorter than t_{cn}.

Let us consider online (real time) classification with GDL. In this case on the top of GDL memory stack resides features set generated from actually captured MoCap data, data one level deeper from previous capture, *etc.* When at a certain moment of time GDL returns class label $\mathbb{C}_c$ we can easily found in memory stack the whole recognized signal F with following algorithm:

Algorithm 1—Retrieving from GDL Memory Stack the Signal that was Classified to Class $\mathbb{C}_c$

//An ordered set that will contains the signal
F: = Empty
//Index of cluster—see Equation (7)
Cluster_Index: = N
//Indicates how deep in GDL memory stack lies the beginning of sequence
//(zero is a top of a stack)
Stack_Index: = 0
//While we found all clusters (keyframes)
While (Cluster_Index > 0)
Begin

//Get all conclusions that are in memory stack in depth indicated
//by Stack_Index
All_Conclusions_Array: = Conclusions(Stack_Index)
//Get features that are in memory stack in depth indicated
//by Stack_Index
Features_Array: = Features(Stack_Index)
//Append features at the begining of ordered set F
//We must remembered that signal resides on stack in reversed order
Append Features_Array **At Beginning Of** F
//If conclusions array satisfied on this level of stack contains
//keyframe with index Cluster_index
If (All_Conclusions_Array **Contains** $C_{cCluster\_Index}$)
Begin
//Select next cluster index
Cluster_Index: = Cluster_Index − 1
END
//Go deeper into the stack
Stack_Index: = Stack_Index + 1
END
//Set Stack_Index on level where last conclusion was found
Stack_Index: = Stack_Index − 1

Key frames of an action can also be indicated manually. In case of manual segmentation of key frames we can also use GDL classification Schema (5)–(7) however $\overline{\mu_b}$ is defined as a mean vector derived from key frames coordinates and $\overline{\sigma_b}$ is a standard deviation of key frames coordinates ($\overline{\varepsilon_b}$ together with $\overline{\sigma_b}$ determines size of a cluster).

It should be noted that $\overline{\sigma_b}$ can also be defined though a process where the training data are clustered using the corresponding vectors of the manually-indicated key-frames as cluster centers. Objects are assigned to the class defined by the closest cluster center by 1-NN (1-nearest neighbor) approach. That way, during training all the available training data are utilized and not only the information of key-frames. This will make this approach very similar to typical maximum-likelihood estimation procedure applied to normal mixtures however without the possibility of modifying means (cluster centers)—so further optimization of square-errors of estimates cannot be obtained.

2.4. Online Actions Segmentation and Classification—The Proposed Approach

The crucial component of a continuous action recognition system is an action segmentation method. The role of this method is action boundaries detection from incoming MoCap data stream. The GDL method we used is based on detection of so-called key frames (static poses) that have to be present in each action sequence under time constraints as in Equation (7). While we can estimate the number of required keyframes in R-GDL approach we can detect static poses with k-means clustering. However keyframes indication is not enough to perform successful segmenting of actions. It is due to the fact that

human actions movement trajectories might differ not only within a group of people but also between actions of a single user. It happens because each person has its own movement patterns and due to tracking hardware inaccuracies. To detect the key frame in MoCap stream we cannot only check the exact match with pattern but rather distance (similarity) between actual body position and key frame. We compute this similarity with L1 norm Equation (6) and we check if this distance value is below threshold value. Because of movement variety we already mentioned finding those thresholds however require statistical analysis of all training data.

In the case of online action recognition, there are no predefined boundaries that divide incoming MoCap data into samples that can be presented to classifier. Also because data is coming continuously we are not sure where a given signal begins and where ends. Signals may contain some irrelevant movements that do not belong to any classes (for example in case of karate techniques recognition we might not be interested in fact that observed person enters or leaves tatami or remains in place without doing any particular move). Due to this fact, after capturing a new MoCap frame and features selection we have to select only those time sequences that probably contain actions we are interested in. In Figure 5 we present processing pipeline of online actions segmentation and classification of the proposed approach.

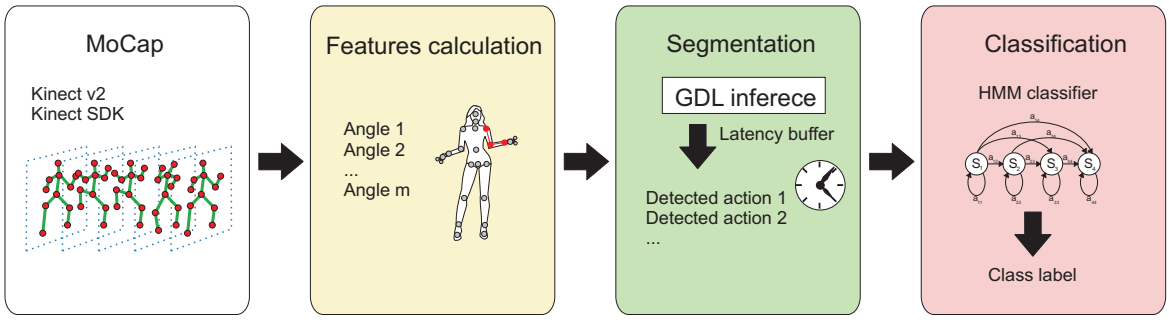

Figure 5. The processing pipeline of online actions segmentation and classification of the proposed approach.

In our implementation we have used Kinect v2 sensor and Microsoft Kinect SDK to capture data. The further experiments were performed on those offline recordings, however our implementation can run in real time (classified with frequency greater than 30 Hz) which enables it to operate on direct data stream from sensor.

In the features calculation step, the proposed solution uses 18 angle-based features that were presented in Section 2.1. Before classification starts we have to train segmentation and classification algorithm. To segment data from continuous recordings we use GDL classifier trained with R-GDL method on exemplar continuous action samples as it was described in Section 2.3. With Algorithm 1 it is possible to retrieve the signal from continuous recording and to prepare it for further classification. Each signal initially identified by GDL is later processed by HMM classifier. However it is possible that GDL classifies the same signal (or two signals that overlaps) to two different classes. In our approach we are assuming that each part of signal belongs maximally to one class and actions do not overlap in time (it is not possible that two actions happen in same time). Due to this, every time a part of signal is classified to a given class a GDL segmentation module wait some small amount of time (so called latency time) before processing signal to HMM classification module. If two or more signals are segmented in a given

latency time, the segmentation module feeds classification module with signal which is composed of all detected signals (see Algorithm 2 for detailed description).

The classification module uses forward-only HMM classifier which is trained with manually segmented action samples. The HMM classifier architecture and training algorithm are described in Section 2.2.

Algorithm 2. An Online Recordings Segmentation

```
//A set of all GDL conclusions
Conclusions: = Empty
//Indicates how deep in GDL memory stack lies the beginning of sequence
Stack_Index: = −1
//How much time passed since last segmentation
Elapsed_Time: = 0
//A minimal time latency since last segmentation—"latency buffer"
Latency: = 0.5 sec
//Initialize the GDL inference engine and stack
GDL_Interpreter: = Initialize_GDL_Interpreter
Stack: = Empty
//A class label of sequence that was recognized in this
//loop of application
Class_Label = "Not_Yet_Detected"
//While MoCap device is running
While MoCap_Is_Running
Begin
    //Generate features array from captured MoCap skeleton
    Features_Array: = GenerateFeatures(MoCap_Data)
    //Add new features array to memory stack of GDL
    Stack = Stack + Features_Array
    //Return all conclusions that can be inferred from memory stack
    New_Conclusions: = GDL_Interpreter(Features_Array)
    Class_Label = "Not_Yet_Detected"
    //If memory stack depth variable has been initialized in previous iterations
    //increase its depth by 1
    If (Stack_Index ≥ 0)
    Begin
    Stack_Index: = Stack_Index + 1
    End
    //For each conclusion inferred by GDL
    Foreach conculsion In New_Conclusions
    Begin
    //If a given conclusion indicates that an action has been detected
        If (conclusion Is Action_Conclusion)
```

```
            Begin
                //Add conclusion to set of conclusions
                Conclusions = Conclusions + conclusion
                //Find depth of the stack where is a first keyframe
                //that supports this conclusion—this value is hold
                //in Algorithm 1 in variable Stack_Index
                Index = GDL_Interpreter(conclusion)
                //Use largest value of already found stack depths
                If (Index > Stack_Index)
                Begin
                    Stack_Index = Index
                End
                //Zero the time that elapsed since last action detected
                Elapsed_Time: = 0;
            End
        End
        //Increase elapsed time by time that passed since last iteration
        //of main loop
        Elapsed_Time = Elapsed_Time + Time_Since_Last_Iteration
        //If set of conclusions is empty reset the elapsed time
        If (Conclusions Is Empty)
        Begin
            Elapsed_Time = 0;
        End
        //If time since last detected action is greater than latency
        //and Stack Index variable has been initialized
        If (Elapsed_Time > Latency And Stack_Index ≥ 0)
        Begin
            //Get the sequence from memory stack starting from Stack_Index
            //to its top
            Sequence: = GDL_Interpreter(Stack_Index, 0)
            //Classify the sequence with HMM classifier
            Class_Label: = HMM(Sequence)
            //Reset memory stack of GDL
            GDL_Interpreter.Stack = Empty
            //Reset conclusions set
            Conclusions = Empty
            Stack_Index = −1
            Elapsed_Time = 0
        End
End
```

2.5. Z-Score Calculation

After the segmentation and before classification the data is preprocessed. The preprocessing is based on Z-score calculation for each feature value:

$$\bar{z} = \frac{\bar{x} - \mu_{\bar{x}}}{\sigma_{\bar{x}}} \tag{8}$$

where $\bar{x}$ are values of single feature in data sample, $\mu_{\bar{x}}$ is a mean of elements in $\bar{x}$ and $\sigma_{\bar{x}}$ is a standard deviation of elements in $\bar{x}$.

3. Experiment and Results

In our experiment we have used our implementation of GDL classifier [46]. The HMM implementation we have used is a part of Accord. NET machine learning framework [45].

3.1. The Dataset

The dataset was recorded on standard PC equipped with Intel Core i5-20 CPU 3.00 GHz, 8 GB of RAM, an AMD Radeon HD 6570 graphic card with Windows 8 Home Premium 64 Bit. We used Kinect for Windows v2 as a data capturing device. Skeleton tracking was performed with Kinect SDK 2.0 library. Data acquisition frequency was 30 Hz. The dataset consisted of MoCap recordings of six volunteers including multiple champion of Kumite Knockdown Oyama karate. We recorded four types of defense techniques (gedan-barai, jodan-uke, soto-uke and uchi-uke) three types of kicks (hiza-geri, mae-geri and yoko-geri) and three stands (kiba-dachi, kokutsu-dachi and zenkutsu-dachi). The stands were preceded by fudo-dachi and were also evaluated as actions (not as static body positions). Kicks were done with the right foot and blocks were done with the right hand. In Figure 6 we present important stages of karate techniques we have evaluated. During recording sessions, each participant periodically performed a single karate technique (for example gedan-barai 20 times) until all 10 techniques has been recorded. After that, those recordings have been segmented to samples that contained only single repetition of technique. In Table 1 we present quantities of MoCap movement samples we gather from six volunteers. Total number of samples was 1236.

The user data gathered by Kinect and tracked by Microsoft Kinect SDK library contains tracking noises. According to research [47] in a more controlled body posture (e.g., standing and exercising arms), the accuracy of the joint estimation is comparable to high-end motion capture. However, in general postures, the variability of the current implementation of pose estimation is about 10 cm which is a large value compared to the size not only of a particular limb but the whole user's body. The measurements could be used to assess general trends in movement, but for quantitative estimation an improved skeletonization with an anthropometric model is needed.

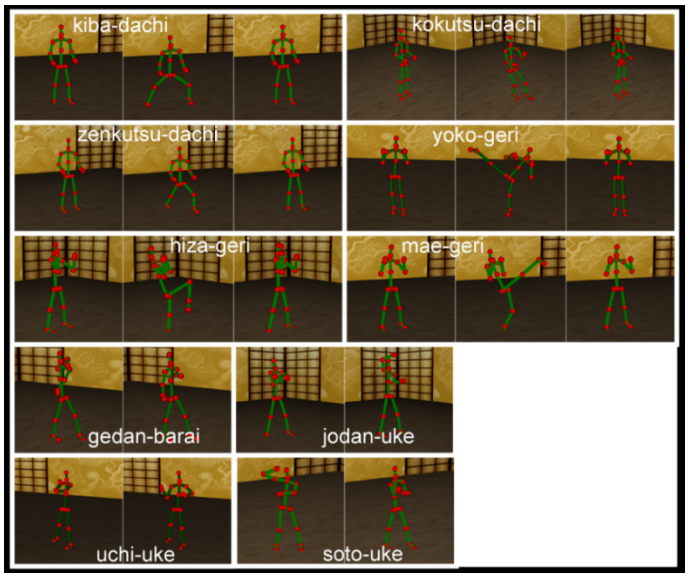

Figure 6. This figure presents three-dimensional visualization of MoCap data from our dataset. Pictures present important phases of karate actions.

3.2. Classification of Segmented Recordings with Hidden Markov Model Classifier

In the first experiment we evaluated five continuous Gaussian density hidden Markov model classifiers on our dataset. All of them had forward-only architecture and differ in number of hidden states from one (which was practically Gaussian mixture model) to five. We have used segmented dataset that was introduced in Section 3.1 and features set described in Section 2.1. We have performed leave-one-out cross validation where movements of five persons have been used as a training dataset and recordings from the remaining one was used for validation. In Table 2 and Figures 7 and 8 we have presented averaged classification results of leave-one-out cross validation of hidden Markov models on segmented karate dataset plus/minus standard deviation. The detailed results are presented in Tables A1–A5 in the Appendix.

3.3. Classification of Unsegmented Recordings with Gestures Description Language and Hidden Markov Model Classifier

In the second experiment, we evaluated our methodology that was described in Section 2.4 on unsegmented recordings of actions. Similarly as in Section 3.2 we have used leave-one-out cross validation however this time we used original (not segmented) recordings of actions from dataset described in Section 3.1. Obviously while we combined GDL with HMM classifiers the movement recordings from the same five people were used to training both approaches.

At first, we have trained GDL classifier with R-GDL approach setting clusters number of k-means clustering to 2. GDL and R-GDL make it possible to evaluate data with more keyframes however as can be seen in Figure 6 two clusters seems to be enough to correctly describe each considered action. In the case of all stands and kicks, each movement starts and ends with an identical (or very similar) body position (which is a first keyframe) while the middle part of the movement is a second keyframe (compare with Figure 4). Each blocking activity is characterized by initial and final positions which is also two keyframes. Of course we can use more than two keyframes however it might cause the loss of generality and the clustering will be more sensitive to outliners in dataset.

Table 1. This table presents quantities of MoCap movement samples we have gather from six volunteers.

Volunteer ID	Gedan-Barai	Hiza-Geri	Jodan-Uke	Kiba-Dachi	Kokutsu-Dachi	Mae-Geri	Soto-Uke	Uchi-Uke	Yoko-Geri	Zenkutsu-Dachi
1	20	20	20	20	20	20	20	20	20	20
2	20	19	19	20	20	20	20	20	20	20
3	21	21	21	21	21	21	21	21	21	20
4	21	20	21	21	21	21	21	21	21	22
5	21	21	21	21	21	21	21	21	21	21
6	21	20	21	21	21	21	21	21	21	21

Table 2. This table presents averaged classification results of leave-one-out cross validation of five hidden Markov models on segmented karate dataset plus/minus standard deviation.

Hidden states	Gedan-Barai	Hiza-Geri	Jodan-Uke	Kiba-Dachi	Kokutsu-Dachi	Mae-Geri	Soto-Uke	Uchi-Uke	Yoko-Geri	Zenkutsu-Dachi
1 state (GMM)	77% ± 31%	86% ± 12%	98% ± 4%	100% ± 0%	100% ± 0%	96% ± 5%	90% ± 14%	71% ± 19%	99% ± 2%	99% ± 2%
2 states	90% ± 13%	87% ± 11%	93% ± 11%	100% ± 0%	100% ± 0%	97% ± 5%	81% ± 26%	86% ± 10%	99% ± 2%	99% ± 2%
3 states	91% ± 11%	92% ± 9%	95% ± 7%	100% ± 0%	100% ± 0%	96% ± 5%	86% ± 21%	98% ± 5%	99% ± 2%	99% ± 2%
4 states	96% ± 4%	93% ± 7%	94% ± 12%	100% ± 0%	100% ± 0%	97% ± 5%	96% ± 7%	98% ± 4%	98% ± 2%	99% ± 2%
5 states	94% ± 7%	92% ± 9%	96% ± 8%	100% ± 0%	100% ± 0%	97% ± 5%	90% ± 17%	94% ± 8%	99% ± 2%	99% ± 2%

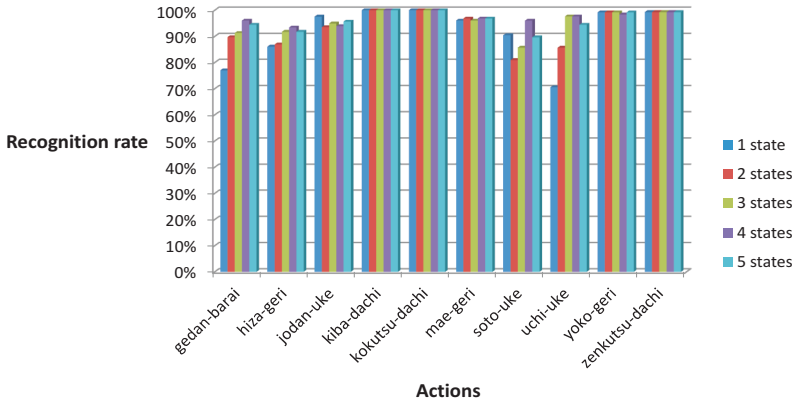

Figure 7. This figure presents averaged classification results of leave-one-out cross validation of five hidden Markov models on segmented karate dataset basing on data from Table 2.

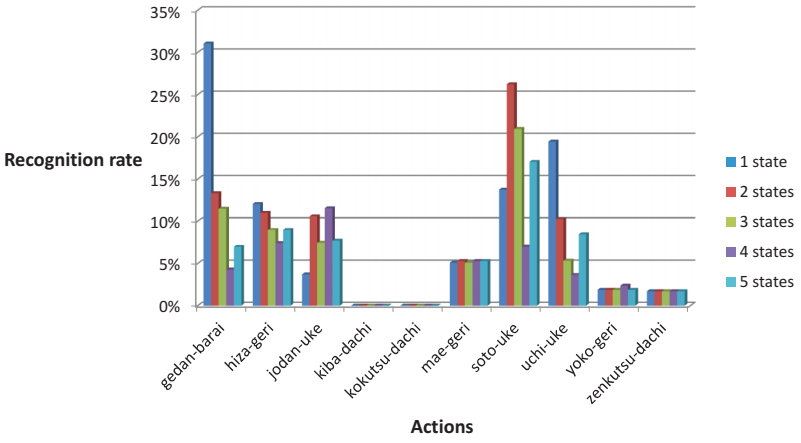

Figure 8. This figure presents standard deviation of classification results of leave-one-out cross validation of five hidden Markov models on segmented karate dataset basing on data from Table 2.

As we expected, key frames of movements we detected to correspond to those presented in Figure 6. We have constructed the final classification pipeline as follows (see Figure 5): the continuous MoCap data is segmented into samples as it was described in Section 2.4. After the evaluation presented in the previous section, we found out that four-state HMM classifier has highest recognition rate and the smallest standard deviation of results. Due to this, we have used this classifier in the classification phase.

Additionally we have evaluated the proposed auto-segmentation GDL-based schema trained with R-GDL approach by comparing it with GDL-schema with manually indicated key frames that we described in the end of the Section 2.3. The purpose of this experiment is to expose the real efficiency of the auto-segmentation scheme, enabling direct comparison with simple manual segmentation. In Table 3 and Figure 9 we present averaged classification results of leave-one-out cross validation of proposed classification approach with manually-trained GDL classifier.

In Table 4 and Figure 10 we present averaged classification results of leave-one-out cross validation of proposed classification approach. In Algorithm 2 we have set a latency parameter to 0.5 sec which is about half of the length of the shortest action in our dataset.

Table 3. This table presents averaged classification results of leave-one-out cross validation of proposed classification approach on unsegmented karate dataset plus/minus standard deviation. This approach uses manually-trained GDL classifier.

Technique	No Decision	Gedan-Barai	Hiza-Geri	Jodan-Uke	Kiba-Dachi	Kokutsu-Dachi	Mae-Geri	Soto-Uke	Uchi-Uke	Yoko-Geri	Zenkutsu-Dachi
gedan-barai	17% ± 13%	71% ± 23%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	3% ± 5%	9% ± 16%	0% ± 0%	0% ± 0%
hiza-geri	27% ± 10%	0% ± 0%	69% ± 7%	0% ± 0%	0% ± 0%	0% ± 0%	4% ± 6%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
jodan-uke	14% ± 9%	0% ± 0%	0% ± 0%	84% ± 10%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	2% ± 4%	0% ± 0%	0% ± 0%
kiba-dachi	5% ± 7%	0% ± 0%	0% ± 0%	0% ± 0%	95% ± 7%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
kokutsu-dachi	16% ± 13%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	77% ± 14%	0% ± 0%	0% ± 0%	7% ± 15%	0% ± 0%	0% ± 0%
mae-geri	27% ± 14%	0% ± 0%	7% ± 8%	0% ± 0%	0% ± 0%	0% ± 0%	55% ± 17%	1% ± 2%	11% ± 24%	0% ± 0%	0% ± 0%
soto-uke	14% ± 12%	1% ± 2%	1% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	68% ± 19%	16% ± 17%	0% ± 0%	0% ± 0%
uchi-uke	12% ± 15%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	86% ± 16%	0% ± 0%	0% ± 0%
yoko-geri	19% ± 14%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	1% ± 2%	79% ± 12%	0% ± 0%
zenkutsu-dachi	5% ± 10%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	95% ± 10%

Table 4. This table presents averaged classification results of leave-one-out cross validation of proposed classification approach on unsegmented karate dataset plus/minus standard deviation. This approach uses GDL classifier trained with R-GDL method.

Technique	No Decision	Gedan-Barai	Hiza-Geri	Jodan-Uke	Kiba-Dachi	Kokutsu-Dachi	MAE-Geri	Soto-Uke	Uchi-Uke	Yoko-Geri	Zenkutsu-Dachi
gedan-barai	11% ± 14%	83% ± 24%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	9% ± 20%	0% ± 0%	0% ± 0%
hiza-geri	20% ± 18%	0% ± 0%	81% ± 15%	0% ± 0%	0% ± 0%	0% ± 0%	5% ± 7%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
jodan-uke	1% ± 2%	0% ± 0%	0% ± 0%	97% ± 4%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	2% ± 4%	0% ± 0%	0% ± 0%
kiba-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
kokutsu-dachi	0% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	99% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
mae-geri	22% ± 15%	0% ± 0%	2% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	82% ± 18%	1% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%
soto-uke	6% ± 7%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	90% ± 12%	6% ± 6%	0% ± 0%	0% ± 0%
uchi-uke	1% ± 14%	0% ± 0%	2% ± 4%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	90% ± 16%	0% ± 0%	0% ± 0%
yoko-geri	1% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	99% ± 2%	0% ± 0%
zenkutsu-dachi	1% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	98% ± 2%

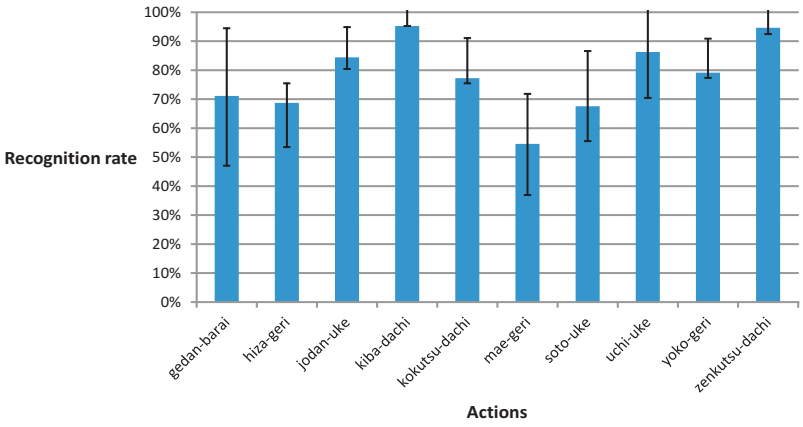

Figure 9. This figure presents averaged classification results of leave-one-out cross validation of proposed classification approach on unsegmented karate dataset. Black bars represent standard deviation. To make this plot we used data from Table 3.

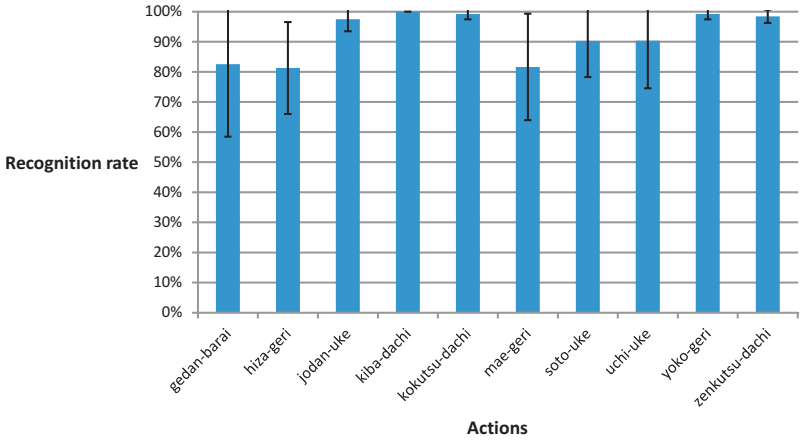

Figure 10. This figure presents averaged classification results of leave-one-out cross validation of proposed classification approach on unsegmented karate dataset. Black bars represent standard deviation. To make this plot we used data from Table 4.

4. Discussion

The results presented in Table 2 clearly demonstrate that HMM are capable to product robust actions classifiers. The proper selection of features enables even HMM with low number of hidden states to model distribution of features value across time space. The most important causes of all classification errors were inaccuracies of body part tracking. The cheap depth-based markerless MoCap system has a large problem with correctly estimatng positions of body joints while limbs are closely situated to each other or are touching the user's corpus. The other important problem is the speed of karate gestures and range of leg movements. This is especially visible in the case of kicks when feet in time less than a second change their position from the ground to a location over the head. However those MoCap problems will always happen while using consumer-quality devices and our method appears to be mostly resistant to them.

The number of tanning samples seems to be adequate to solve our problem. The validation results with leave-one-out method proved good generalization of the method without overfitting. The most difficult for

correct classification were blocking techniques. Gedan-barai, jodan-uke, soto-uke, and uchi-uke were often confused with each other. As can be seen in Table A1 the Gaussian mixture model obtained far worst results than other classifiers. For example, the recognition rate of uchi-uke was 71% ± 19% and surprisingly was very often confused with mae-geri (25% ± 20%). That means that this model is insufficient to perform correct recognition. The similar situation was in the case of gedan-barai (recognition 77% ± 31%) that was confused with mae-geri (11% ± 14%), jodan-uke (1% ± 2%), soto-uke (7% ± 11%) and uchi-uke (4% ± 6%). Those errors have been nearly eliminated after increasing number of hidden states. For two-states HMM uchi-uke was correctly recognized in 86% ± 10% (gedan-barai 90% ± 13%), for three-states 98% ± 5% (91% ± 11%), for four-states 98% ± 4% (96% ± 4%) and five-states 94% ± 8% (94% ± 7%). As can be seen after increasing number of states from four to five a recognition rate slightly dropped that might be a sign of classifiers overfitting (losing the generalization potential). Due to this fact in evaluation we did not consider a larger number of states than five. The other hard-to-recognize class was hiza-geri which movements range is within mae-geri kick. In Gaussian mixture model 86% ± 12% hiza-geri exemplars were correctly classified while 7% ± 8% was misclassified as mae-geri. After increasing the number of states, recognition rates also increased (respectively to 87% ± 11% for two states, 92% ± 9% for three states, 93% ± 7% for four and 92% ± 9% for five) however, it was constantly confused with mae-geri. The karate stands kiba-dachi and kokutsu-dachi were 100% correctly classified while zenkutsu-dachi in 99% ± 2%. Despite errors mentioned before overall classification result for four-states HMM were very good and range from 93% ± 7% for hiza-geri to 100% for kiba-dachi and kokutsu-dachi.

The second experiment was far more challenging because proposed classifier that was a composition of GDL segmentation and four-states HMM classification modules has been confronted with continuous (unsegmented) karate recordings. Our approach copes with this task quite well. This time however there is a possibility that beside wrongly assigning a class label to the data sample, the classifier fails in segmenting an action from MoCap recording (in Tables 3 and 4 we call this situation "no decision") or splitting a single action into two or more (however this situation never happened and we will not consider it in further discussion). We can easily notice that an auto-segmentation R-GDL schema outperforms the GDL classifier trained with manually segmented data. In Table 3 we can observe more "no decision" errors than in Table 4. That is because in manual segmentation schema the $\overline{\sigma_b}$ parameter might be underestimated due to the fact that this training method uses only information about key frames while R-GDL performs clustering of whole training dataset. Because both segmentation methods confuse similar classes in further discussion we will concentrate on classifier trained with R-GDL method (Table 4 and Figure 10).

Similarly to the first experiment, the most challenging actions were blocking techniques and front kicks (mae-geri and hiza-geri). When GDL fails in correct determination of an action segment it might change the probability distribution among the sample. The worst results were obtained for hiza-geri (81% ± 15%) and it was most often confused with mae-geri (5% ± 7%) or not detected at all (20% ± 18%). Also mae-geri and gedan-barai has a recognition rate close to 80% (82% ± 18% and 83% ± 24% respectively). Mae-geri was confused with hiza-geri (2% ± 2%) soto-uke (1% ± 2%) or not classified (22% ± 15%) while gedan-barai was confused with uchi-uke (9% ± 20%) or not classified 11% ± 14%. In the case of other blocks, the results were better: 97% ± 4% of jodan-uke, 90% ± 12% of soto-uke and 90% ± 16% of uchi-uke were correctly classified. Jodan-uke is also the only class for which recognition rate of unsegmented data was higher than on segmented (97% ± 4% to 94% ± 12% respectively). This phenomena can be easily explained: as we mentioned in Section 2.4 GDL-based segmenting leaves some

"safety margin" (latency buffer) in case more than one class would be detected in ongoing processing. As the HMM has been learned on noisy MoCap data the slight change of section range might cause small fluctuations in recognition rates both in plus and in minus (in case jodan-uke it is 3%).

In case of yoko-geri kick recognition rate was 99% ± 2%. It is noteworthy that yoko-geri has the largest recognition rate for all types of kicks—it was because while performing this side kick the whole body can be observed by single MoCap sensor and one part of body is not covered by other ones like it is in front kicks. We can also notice that hiza-geri and mae-geri recognition failed mostly because the movements had not been segmented by GDL and presented to HMM classifier (the sign of it is high value of no decision error). That was caused by the fact that tracking software failed to correctly estimate an angle in the hip and knee. The recognition rate of karate stands were 100% for kiba-dachi, 99% ± 2% for kokutsu-dachi and 98% ± 2% for zenkutsu-dachi which is nearly identical as results from first experiment on segmented dataset.

5. Conclusions

Based on results from the previous section we can conclude that the proposed method deals very well with problem of online segmentation and classification of karate techniques. The recognition rate is between 81% ± 15% for hiza-geri and 100% ± 0% for kiba-dachi. The proposed method can be used in real-time applications and analyzes the incoming data with minimal latencies. The prototype implementation of the proposed approach we made is capable of processing incoming data with the same speed as it comes from the MoCap system (30 Hz) on consumers' quality hardware. As we showed in the discussion section, the main source of errors of our approach are tracking errors of MoCap data however even though we used off-the-shelf depth sensors, the results we obtained are very satisfying. We have validated our approach and proven its applicability on a unique karate dataset. It has to be noticed that martial arts actions are far more difficult to classify than typical common-life movements due to speed, agility, and flexibility of the body that is unattainable for typical not trained person. Our method is not limited only to sports or to particular type of data acquisition (MoCap) protocol. The segmentation based on cluster analysis with R-GDL method can be performed on any type of features set which features values that are within the normalized common range.

In our future research we would like to apply our methodology to data capture with high-end MoCap systems and to apply it to construct virtual reality karate training systems which will support practicing and teaching this sport. Virtual reality has been already introduced to various sports [48] however not to karate. After applying trajectory evaluation and pattern classification/matching methods we could evaluate not only if a particular martial artist performs a particular technique but also how well he or she does it compared to the best world-class martial artists. We would like also to apply the web-based viewing technology to this future solution similar to those presented in [49]. In our opinion, cheap contemporary MoCap hardware due its limitation is unable to produce good quality data with which it is possible to perform measurements valuable for sport. Some papers report the initial research in this field however, before dealing with real high-quality data, the problem is still open and unsolved.

Conflicts of Interest

The authors declare no conflict of interest.

Appendix

Table A1. This table presents averaged classification results of leave-one-out cross validation of 1-state hidden Markov (Gaussian mixture model) models on segmented karate dataset plus/minus standard deviation.

Technique	Gedan-Barai	Hiza-Geri	Jodan-Uke	Kiba-Dachi	Kokutsu-Dachi	Mae-Geri	Soto-Uke	Uchi-Uke	Yoko-Geri	Zenkutsu-Dachi
gedan-barai	77% ± 31%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	11% ± 14%	7% ± 11%	4% ± 6%	0% ± 0%	0% ± 0%
hiza-geri	0% ± 0%	86% ± 12%	0% ± 0%	0% ± 0%	0% ± 0%	7% ± 8%	0% ± 0%	7% ± 11%	0% ± 0%	0% ± 0%
jodan-uke	0% ± 0%	0% ± 0%	98% ± 4%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	2% ± 2%	0% ± 0%	0% ± 0%
kiba-dachi	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
kokutsu-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
mae-geri	0% ± 0%	2% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	96% ± 5%	0% ± 0%	2% ± 4%	0% ± 0%	0% ± 0%
soto-uke	0% ± 0%	0% ± 0%	2% ± 4%	0% ± 0%	0% ± 0%	1% ± 2%	90% ± 14%	7% ± 10%	0% ± 0%	0% ± 0%
uchi-uke	0% ± 0%	1% ± 2%	3% ± 4%	0% ± 0%	0% ± 0%	25% ± 20%	1% ± 2%	71% ± 19%	0% ± 0%	0% ± 0%
yoko-geri	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	99% ± 2%	0% ± 0%
zenkutsu-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	99% ± 2%

Table A2. This table presents averaged classification results of leave-one-out cross validation of 2-states hidden Markov (Gaussian mixture model) models on segmented karate dataset plus/minus standard deviation.

Technique	Gedan-Barai	Hiza-Geri	Jodan-Uke	Kiba-Dachi	Kokutsu-Dachi	Mae-Geri	Soto-Uke	Uchi-Uke	Yoko-Geri	Zenkutsu-Dachi
gedan-barai	90% ± 13%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	1% ± 2%	6% ± 10%	3% ± 5%	0% ± 0%	0% ± 0%
hiza-geri	0% ± 0%	87% ± 11%	1% ± 2%	0% ± 0%	0% ± 0%	6% ± 7%	0% ± 0%	7% ± 9%	0% ± 0%	0% ± 0%
jodan-uke	0% ± 0%	0% ± 0%	93% ± 11%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	7% ± 11%	0% ± 0%	0% ± 0%
kiba-dachi	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
kokutsu-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
mae-geri	0% ± 0%	1% ± 2%	1% ± 2%	0% ± 0%	0% ± 0%	97% ± 5%	0% ± 0%	2% ± 4%	0% ± 0%	0% ± 0%
soto-uke	0% ± 0%	0% ± 0%	2% ± 5%	0% ± 0%	0% ± 0%	2% ± 5%	81% ± 26%	14% ± 19%	0% ± 0%	0% ± 0%
uchi-uke	0% ± 0%	1% ± 2%	5% ± 7%	0% ± 0%	0% ± 0%	8% ± 11%	1% ± 2%	86% ± 10%	0% ± 0%	0% ± 0%
yoko-geri	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	99% ± 2%	0% ± 0%
zenkutsu-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	99% ± 2%

Table A3. This table presents averaged classification results of leave-one-out cross validation of 3-states hidden Markov (Gaussian mixture model) models on segmented karate dataset plus/minus standard deviation.

Technique	Gedan-Barai	Hiza-Geri	Jodan-Uke	Kiba-Dachi	Kokutsu-Dachi	Mae-Geri	Soto-Uke	Uchi-Uke	Yoko-Geri	Zenkutsu-Dachi
gedan-barai	91% ± 11%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	2% ± 2%	3% ± 5%	4% ± 7%	0% ± 0%	0% ± 0%
hiza-geri	0% ± 0%	92% ± 9%	0% ± 0%	0% ± 0%	0% ± 0%	6% ± 7%	0% ± 0%	3% ± 6%	0% ± 0%	0% ± 0%
jodan-uke	0% ± 0%	0% ± 0%	95% ± 7%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	5% ± 7%	0% ± 0%	0% ± 0%
kiba-dachi	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
kokutsu-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
mae-geri	0% ± 0%	2% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	96% ± 5%	0% ± 0%	2% ± 4%	0% ± 0%	0% ± 0%
soto-uke	0% ± 0%	0% ± 0%	5% ± 11%	0% ± 0%	0% ± 0%	2% ± 4%	86% ± 21%	8% ± 11%	0% ± 0%	0% ± 0%
uchi-uke	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	2% ± 5%	0% ± 0%	98% ± 5%	0% ± 0%	0% ± 0%
yoko-geri	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	99% ± 2%	0% ± 0%
zenkutsu-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	99% ± 2%

Table A4. This table presents averaged classification results of leave-one-out cross validation of 4-states hidden Markov (Gaussian mixture model) models on segmented karate dataset plus/minus standard deviation.

Technique	Gedan-Barai	Hiza-Geri	Jodan-Uke	Kiba-Dachi	Kokutsu-Dachi	Mae-Geri	Soto-Uke	Uchi-Uke	Yoko-Geri	Zenkutsu-Dachi
gedan-barai	96% ± 4%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	1% ± 2%	2% ± 4%	0% ± 0%	0% ± 0%
hiza-geri	0% ± 0%	93% ± 7%	0% ± 0%	0% ± 0%	0% ± 0%	7% ± 7%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
jodan-uke	0% ± 0%	0% ± 0%	94% ± 12%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	6% ± 12%	0% ± 0%	0% ± 0%
kiba-dachi	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
kokutsu-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
mae-geri	0% ± 0%	2% ± 4%	0% ± 0%	0% ± 0%	0% ± 0%	97% ± 5%	0% ± 0%	2% ± 2%	0% ± 0%	0% ± 0%
soto-uke	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	96% ± 7%	4% ± 7%	0% ± 0%	0% ± 0%
uchi-uke	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	2% ± 4%	0% ± 0%	98% ± 4%	0% ± 0%	0% ± 0%
yoko-geri	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	2% ± 2%	0% ± 0%	0% ± 0%	98% ± 2%	0% ± 0%
zenkutsu-dachi	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	99% ± 2%

Table A5. This table presents averaged classification results of leave-one-out cross validation of 5-states hidden Markov (Gaussian mixture model) models on segmented karate dataset plus/minus standard deviation.

Technique	Gedan-Barai	Hiza-Geri	Jodan-Uke	Kiba-Dachi	Kokutsu-Dachi	Mae-Geri	Soto-Uke	Uchi-Uke	Yoko-Geri	Zenkutsu-Dachi
gedan-barai	94% ± 7%	0% ± 0%	2% ± 2%	0% ± 0%	0% ± 0%	1% ± 2%	1% ± 2%	2% ± 5%	0% ± 0%	0% ± 0%
hiza-geri	0% ± 0%	92% ± 9%	0% ± 0%	0% ± 0%	0% ± 0%	6% ± 7%	0% ± 0%	3% ± 6%	0% ± 0%	0% ± 0%
jodan-uke	0% ± 0%	0% ± 0%	96% ± 8%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	4% ± 8%	0% ± 0%	0% ± 0%
kiba-dachi	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
kokutsu-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	100% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%
mae-geri	0% ± 0%	2% ± 4%	0% ± 0%	0% ± 0%	0% ± 0%	97% ± 5%	0% ± 0%	2% ± 2%	0% ± 0%	0% ± 0%
soto-uke	0% ± 0%	0% ± 0%	3% ± 5%	0% ± 0%	0% ± 0%	2% ± 5%	90% ± 17%	5% ± 7%	0% ± 0%	0% ± 0%
uchi-uke	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	5% ± 9%	0% ± 0%	94% ± 8%	0% ± 0%	0% ± 0%
yoko-geri	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	0% ± 0%	99% ± 2%	0% ± 0%
zenkutsu-dachi	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	0% ± 0%	1% ± 2%	0% ± 0%	99% ± 2%

References

1. Chatzis, S.P.; Kosmopoulos, D.I.; Doliotis, P. A conditional random field-based model for joint sequence segmentation and classification. *Pattern Recognit.* **2013**, *46*, 1569–1578.
2. Yang, X.; Tian, Y. Effective 3D action recognition using EigenJoints. *J. Visual Commun. Image Represent.* **2014**, *25*, 2–11.
3. Boubou, S.; Suzuki, E. Classifying actions based on histogram of oriented velocity vectors. *J. Intell. Inf. Syst.* **2015**, *44*, 49–65.
4. Celiktutan, O.; Akgul, C.B.; Wolf, C.; Sankur, B. Graph-based analysis of physical exercise actions. In Proceedings of the 1st ACM international workshop on Multimedia indexing and information retrieval for healthcare (MIIRH '13), Barcelona, Catalunya, Spain, 21–25 October 2013; pp. 23–32.
5. Miranda, L.; Vieira, T.; Martínez, D.; Lewiner, T.; Vieira, A.W.; Campos, M.F.M. Online gesture recognition from pose kernel learning and decision forests. *Pattern Recognit. Lett.***2014**, *39*, 65–73.
6. López-Méndez, A.; Casas, J.R. Model-based recognition of human actions by trajectory matching in phase spaces. *Image Vis. Comput.* **2012**, *30*, 808–816.
7. Jiang, X.; Zhong, F.; Peng, Q.; Qin, X. Online robust action recognition based on a hierarchical model. *Visual Comput.* **2014**, *30*, 1021–1033.
8. Kviatkovsky, I.; Rivlin, E.; Shimshoni, I. Online action recognition using covariance of shape and motion. *Comput. Vis. Image Underst.* **2014**, *129*, 15–26.
9. Theodorakopoulos, I.; Kastaniotis, D.; Economou, G.; Fotopoulos, S. Pose-based human action recognition via sparse representation in dissimilarity space. *J. Visual Commun. Image Represent.* **2014**, *25*, 12–23
10. Nyirarugira, C.; Kim, T.Y. Stratified gesture recognition using the normalized longest common subsequence with rough sets. *Signal Process. Image Commun.* **2015**, *30*, 178–189.
11. Hachaj, T.; Ogiela, M.R. Rule-based approach to recognizing human body poses and gestures in real time. *Multimed. Syst.* **2014**, *20*, 81–99.
12. Meinard, M. *Information Retrieval for Music and Motion*; Springer: Heidelberg, Germany, 2007.
13. Arici, T.; Celebi, S.; Aydin, A.S.; Temiz, T.T. Robust gesture recognition using feature pre-processing and weighted dynamic time warping. *Multimed. Tools Appl.* **2014**, *72*, 3045–3062.
14. Barnachon, M.; Bouakaz, S.; Boufama, B.; Guillou, E. Ongoing human action recognition with motion capture. *Pattern Recognit.* **2014**, *47*, 238–247.
15. Su, C.-J.; Chiang, C.-Y.; Huang, J.-Y. Kinect-enabled home-based rehabilitation system using Dynamic Time Warping and fuzzy logic. *Appl. Soft Comput.* **2014**, *22*, 652–666.
16. Pham, H.-T.; Kim, J.-J.; Nguyen, T.L.; Won, Y. 3D motion matching algorithm using signature feature descriptor. *Multimed. Tools Appl.* **2015**, *74*, 1125–1136.
17. Głowacz, A.; Głowacz, W. DC machine diagnostics based on sound recognition with application of LPC and Fuzzy Logic. *Prz. Elektrotech. Electr. Rev.* **2009**, *85*, 112–115.
18. Głowacz, A.; Głowacz, W. Diagnostics of Direct Current motor with application of acoustic signals, reflection coefficients and K-Nearest Neighbor classifier. *Prz. Elektrotech. Electr. Rev.* **2012**, *88*, 231–233.
19. Mead, R.; Atrash, A.; Matarić, M.J. Automated Proxemic Feature Extraction and Behavior Recognition: Applications in Human-Robot Interaction. *Int. J. Soc. Robot.* **2013**, *5*, 367–378.

20. Kosmopoulos, D.; Papoutsakis, K.; Argyros, A. Online segmentation and classication of modeled actions performed in the context of unmodeled ones. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014.
21. Chan-Hon-Tong, A.; Achard, C.; Lucat, L. Simultaneous segmentation and classification of human actions in video streams using deeply optimized Hough transform. *Pattern Recognit.* **2014**, *47*, 3807–3818
22. Hoai, M.; Lan, Z.; de la Torre, F. Joint segmentation and classication of human actions in video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 20–25 June 2011; pp. 3265–3272.
23. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3d convolutional neural networks for human action recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231.
24. Le, Q.; Ngiam, J.; Chen, Z.; Chia, D.; Koh, P.; Ng, A. Tiled convolutional neural networks. In Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 6–11 December 2010; pp. 1279–1287.
25. Hinton, G.E.; Osindero, S.; Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554.
26. Chen, L.; Wei, H.; Ferryman, J. A survey of human motion analysis using depth imagery. *Pattern Recognit. Lett.***2013**, *34*, 1995–2006.
27. Wang, B.; Li, Z.; Ye, W.; Xie, Q. Development of human-machine interface for teleoperation of a mobile manipulator. *Int. J. Control Autom. Syst.* **2012**, *10*, 1225–1231.
28. Jagodziński, P.; Wolski, R. Assessment of Application Technology of Natural User Interfaces in the Creation of a Virtual Chemical Laboratory. *J. Sci. Educ. Technol.* **2015**, *24*, 16–28.
29. Galna, B.; Barry, G.; Jackson, D.; Mhiripiri, D.; Olivier, P.; Rochester, L. Accuracy of the Microsoft Kinect sensor for measuring movement in people with Parkinson's disease. *Gait Posture* **2014**, *39*, 1062–1068.
30. Kongsro, J. Estimation of pig weight using a Microsoft Kinect prototype imaging system. *Comput. Electron. Agric.* **2014**, *109*, 32–35.
31. Mastorakis, G.; Makris, D. Fall detection system using Kinect's infrared sensor. *J. Real Time Image Process.* **2014**, *9*, 635–646.
32. Planinc, R.; Kampel, M. Introducing the use of depth data for fall detection. *Pers. Ubiquitous Comput.* **2013**, *17*, 1063–1072.
33. Won, A.S.; Bailenson, J.N.; Stathatos, S.C.; Dai, W. Automatically Detected Nonverbal Behavior Predicts Creativity in Collaborating Dyads. *J. Nonverbal Behav.* **2014**, *38*, 389–408.
34. Xu, X.; Ke, F. From psychomotor to "motorpsycho": Learning through gestures with body sensory technologies. *Educ. Technol. Res. Dev.* **2014**, *62*, 711–741.
35. Hachaj, T.; Baraniewicz, D. Knowledge Bricks—Educational immersive reality environment. *Int. J. Inf. Manag.* **2015**, doi:10.1016/j.ijinfomgt.2015.01.006.
36. Clark, R.A.; Pua, Y.H.; Fortin, K.; Ritchie, C.; Webster, K.E.; Denehy, L.; Bryant, A.L. Validity of the Microsoft Kinect for assessment of postural control. *Gait Posture* **2012**, *36*, 372–377.
37. Song, X.; Zhong, F.; Wang, Y.; Qin, X. Estimation of Kinect depth confidence through self-training. *Visual Comput.* **2014**, *30*, 855–865.

38. Kolahi, A.; Hoviattalab, M.; Rezaeian, T.; Alizadeh, M.; Bostan, M.; Mokhtarzadeh, H. Design of a marker-based human motion tracking system. *Biomed. Signal Process. Control* **2007**, *2*, 59–67.
39. Ilg, W.; Mezger, J.; Giese, M. Estimation of Skill Levels in Sports Based on Hierarchical Spatio-Temporal Correspondences. *Pattern Recognit.* **2003**, *2781*, 523–531.
40. Stasinopoulos, S.; Maragos, P. Human action recognition using Histographic methods and hidden Markov models for visual martial arts applications. In Proceedings of the 2012 19th IEEE International Conference on Image Processing (ICIP), Orlando, FL, USA, 30 September–3 October 2012.
41. Bianco, S.; Tisato, F. Karate moves recognition from skeletal motion. In Proceedings of the SPIE 8650, Three-Dimensional Image Processing (3DIP) and Applications 2013, Burlingame, CA, USA, 12 March 2013.
42. Hachaj, T.; Ogiela, M.R.; Piekarczyk, M. Dependence of Kinect sensors number and position on gestures recognition with Gesture Description Language semantic classifier. In Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, Krakow, Poland, 8–11 September 2013; pp. 571–575.
43. Hachaj, T.; Ogiela, M.R. Full-body gestures and movements recognition: User descriptive and unsupervised learning approaches in GDL classifier. In Proceedings of the SPIE 9217, Applications of Digital Image Processing XXXVII, San Diego, CA, USA, 23 September 2014.
44. Ogiela, M.R.; Hachaj, T. *Natural User Interfaces in Medical Image Analysis: Advances in Computer Vision and Pattern Recognition*; Springer International Publishing: Cham, Switzerland, 2015.
45. Accord.NET Framework. Available online: http://accord-framework.net/ (accessed on 1 September 2015).
46. GDL Technology. Available online: http://cci.up.krakow.pl/gdl/ (accessed on 1 September 2015).
47. Obdržálek, Š.; Kurillo, G.; Ofli, F.; Bajcsy, R.; Seto, E.; Jimison, H.; Pavel, M. Accuracy and Robustness of Kinect Pose Estimation in the Context of Coaching of Elderly Population. In Proceedings of the 34th International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), San Diego, CA, USA, 28 August–1 September 2012.
48. Vignais, N.; Kulpa, R.; Brault, S.; Presse, D.; Bideau, B. Which technology to investigate visual perception in sport: Video *vs.* virtual reality. *Human Mov. Sci.* **2015**, *39*, 12–26.
49. Piorkowski, A.; Jajesnica, L.; Szostek, K. Computer Networks. *Commun. Comput. Inf. Sci.* **2009**, *39*, 218–224.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/4.0/).