Cognitive Processing (2021) 22:77–91
https://doi.org/10.1007/s10339-020-00986-4

RESEARCH ARTICLE

Online recognition of unsegmented actions with hierarchical SOM architecture

Zahra Gharaee[1,2]

Received: 30 April 2019 / Accepted: 4 July 2020 / Published online: 22 July 2020

Abstract

Automatic recognition of an online series of unsegmented actions requires a method for segmentation that determines when an action starts and when it ends. In this paper, a novel approach for recognizing unsegmented actions in online test experiments is proposed. The method uses self-organizing neural networks to build a three-layer cognitive architecture. The unique features of an action sequence are represented as a series of elicited key activations by the first-layer self-organizing map. An average length of a key activation vector is calculated for all action sequences in a training set and adjusted in learning trials to generate input patterns to the second-layer self-organizing map. The pattern vectors are clustered in the second layer, and the clusters are then labeled by an action identity in the third layer neural network. The experiment results show that although the performance drops slightly in online experiments compared to the offline tests, the ability of the proposed architecture to deal with the unsegmented action sequences as well as the online performance makes the system more plausible and practical in real-case scenarios.

Keywords Action recognition and segmentation · Self-organizing neural networks · Cognitive architecture · Online performance · Hierarchical models

Introduction

Action recognition plays an important role in interaction between any two agents. Human–robot interactions require that the robot can recognize what kind of action the human is performing. There are several other applications for action recognition systems including human–computer interaction, video retrieval, sign language recognition, medical health care, video analysis (sports video analysis), game industry and video surveillance. In earlier works (Gharaee et al. 2016, 2017b, a, c; Gharaee 2018a), Gharaee et al. developed a system for human action recognition using hierarchical architecture based on self-organizing neural networks. Recently this architecture is developed by using growing grid networks to improve its perfomance in recognizing human actions (Gharaee 2018b, 2020).

As a background to the method proposed in this article, there is a presentation of some psychological approaches to action categorization and to event segmentation. There is also a description of a few earlier computational attempts to solve the action segmentation problem. The rest of the paper is organized as follows: The proposed architecture is described in "Methods" section, the experiments on action recognition are presented in detail in "Results" section, a discussion about the method proposed of this paper in comparison with some other techniques is presented in "Discussion" section, and finally "Conclusion" section concludes the paper.

Psychological approaches of event perception and segmentation

Michotte (1963) conducted a series of early studies concerning interactions between two objects, and he argued that a causal interaction between two objects is perceived when we see the motion of the objects as a single event. Another

Handling editor: Stefan Kopp (Bielefeld University).

Reviewers: two researchers who prefer to remain anonymous.

✉ Zahra Gharaee
zahra.gharaee@gmail.com

[1] Computer Vision Laboratory (CVL), Linköping University, Linköping, Sweden

[2] Department of Philosophy and Cognitive Science, Lund University, Helgonavägen 3, 221 00 Lund, Sweden

early contributor was Gibson (1966) and Gibson (1979). He identified three kinds of events in visual perception: (1) changes in the layout of the surfaces, (2) changes in the color or texture of the surfaces and (3) the coming into or out of existence of surfaces. He argued that the presence of an invariant structure persisting throughout the change is the main factor in creating an event.

A third approach to event perception, based on biological motion, originates from the studies by Johansson (1973). He developed a method called the patch-light technique in which reflective patches are placed on the body of a subject that performs different actions. The subject is filmed in high-contrast light condition, and the film is shown to observers. The observers could only see the movements of the patch-light points, but they could nevertheless recognize, within tenths of a second that the moving light points come from a human performing an action such as walking or crawling. Gärdenfors (2007), Gärdenfors and Warglien (2012) generalize Johansson's approach, proposing that human cognition represents an action by the pattern of forces generating it.

A common feature of these three approaches to event perception is that the dynamic features of the activity are critical for perceiving and categorizing events. At the same time, they indicate that there is a higher-order stability in events that persist through these changes.

The problem of human event segmentation concerns how our perceptual system can partition the stream of experience into meaningful parts. The event segmentation theory proposed by Radvansky and Zacks (2014) is a new approach to how human cognition segments events. The event models presumed by the theory represent features of the current activity relevant to current goals and the models integrate information across the sensory modalities with information that may be more conceptual in nature.

The event segmentation theory entails that the representation of events involves biasing the pathway from sensory input to prediction. The theory says that the working models are disconnected from the sensory input and they store a static snapshot of the current event (preservation). This helps the event models overcome ambiguities and missing information. This process continues by comparing the predictions about the near future of an ongoing event with what actually happens, that is, monitoring the prediction error. If the prediction error suddenly increases, the event model will be updated by opening the inputs of the event models so that a new event is started. By opening the gates to a new operating model, perceptual information interacts with stored knowledge representations building a new event representation and when it is constructed, the prediction error is decreased and the gate closes.

Zacks et al. (2009) present three empirical experiments that have tested their event segmentation theory. The experiments are performed to investigate the ways in which the body movements of an actor predict when an observer will perceive event boundaries. In these experiments, participants segmented the movies of daily activities performed by a single actor using a set of objects on a tabletop. The results show that movement variables were significant predictors of the segmentation. The observers were more sensitive to the movements of the individual body parts and the distance between them than to the relative speed and acceleration of the body parts with respect to each other.

The psychological approach in event segmentation process introduces by Zacks et al. (2009) is based on the possibility to predict the forthcoming movements. In other words, the system needs to predict the possible future movement of the actor based on what has been observed so far. As long as the prediction fits with the incoming stream of movements, it is maintained. Otherwise, the system predicts that a new action has begun. As an example, take the action of scratching the head. The observer tracks the movements of the actor from when the arm is lifted. If the hand approaches the head and touches it, then the observer categorizes it as head scratching and when the hand moves back and leaves the head it is considered the end of the action.

Computational models of action recognition

Human action recognition methods are largely dependent on the input modalities. There are three different types of input modalities that represent the actions performed: the RGB (color images), depth maps and skeleton information. The space-time volumes, spatiotemporal features and trajectories have been utilized for action recognition through the color images in the earlier methods proposed by Schuldt et al. (2004), Dollar et al. (2005), Sun et al. (2009).

The color-based methods are sensitive to color and illumination variations, and thus, they have limitations in recognition robustness. With advent of RGB-D sensors, the action recognition methods were developed based on depth maps, which are insensitive to illumination changes and color variations, and provide us with rich 3D structural information of the scene. In the holistic approaches, the global features such as silhouettes and space-time information are extracted from depth maps like the methods proposed in Oreifej and Liu (2013), Li et al. (2010), Liu et al. (2017). Other approaches extract the local features as a set of interest points from depth sequence (spatiotemporal features) and compute a feature descriptor for each interest point like the methods proposed in Laptev (2005), Wang et al. (2012b), Wang et al. (2012a).

The cost-effective depth sensors are then coupled with the real-time 3D skeleton estimation algorithm introduced by Shotton et al. (2011). By extraction of the spatiotemporal features from the 3D skeleton information such as the relative geometric velocity between body parts, relative joint positions and joint angles in Yao et al. (2017), the position differences of the skeleton joints in Yang and Tian (2012) or the pose information together with differential quantities (speed and acceleration) in Zanfir et al. (2013) the body skeleton information in space and time is first described. Then, the descriptors are coupled with principle component analysis (PCA) or another classifier to categorize the actions.

Such methods for action recognition utilize the pre-segmented and labeled datasets of actions, while online recognition of actions is crucial in real-time experiments with unsegmented sequences of actions. Next there is a description of other attempts in the literature to design computational models for online action recognition.

A main approach for online action recognition is based on the sliding window. Jalal et al. (2017) present a method, which segments human depth silhouettes using temporal human motion information and obtains skeleton joints through spatiotemporal human body information. Then, it trains the hidden Markov model with the code vectors of the multi-fused features to recognize the segmented actions. Vieira et al. (2012) proposed a visual representation of 3D action recognition by space-time occupancy patterns. The method focuses on classifying the extracted feature vectors (interest points) from depth sensors by support vector machine.

In Ellis et al. (2013), the skeleton data of specific events are converted to a feature vector of clustered pairwise joint distances between the current frame, ten previous frames and thirty previous frames. The feature vectors are sent to classifiers that categorize actions based on canonical body poses. The method proposed in Lv and Nevatia (2006) uses a dynamic programming algorithm to segment and recognize actions simultaneously. Their method decomposes the high-dimensional 3D joint representation into a set of feature spaces where each feature corresponds to the motion of a joint or related multiple joints. A weak classifier based on the hidden Markov model is formed for each feature, and these classifiers are combined by the multi-class AdaBoost algorithm.

Among the neural network-based methods for online action recognition, there are convolutional neural network-based systems and recurrent neural network-based systems for action recognition. A multi-region two-stream R-CNN model for detecting actions in the videos is proposed by Peng and Schmid (2016), by which the motion region network generates proposals complementary to those of an appearance region proposal network. They claim that stacking optical flow over several frames significantly improves frame-level action detection. A model of segment-based 3D convolutional network is used for action localization in long videos (see Shou et al. 2016), which identifies candidate segments in a long video that may contain actions. A classification network learns action classification model to initialize the localization model, which fine-tunes the learned classification network to localize an action instance. The UntrimmedNet model proposed by Wang et al. (2017) is composed of two main components implemented with feed-forward networks: the classification module and the selection module. They learn the action model from the video input and reason about the temporal duration of the action instances.

Among the recurrent neural network-based system for action recognition is the method proposed by Singh et al. (2016), a tracking algorithm is used to locate a bounding box around the performer in the video frames, which makes a frame of reference for appearance and motion and two additional streams are trained on motion and appearance. The pixel trajectories of a frame are utilized for the motion streams and a multi-stream CNN is followed by a bidirectional long short-term memory (LSTM) layer to model long-term temporal dynamics within and between the actions. The proposed model by Dave et al. (2017) proposes an action detection model for video processing, which utilizes a series of recurrent neural networks that sequentially make top-down prediction of the future and later correct the predictions with bottom-up observations. The proposed approach by Ma et al. (2016) argues that when training the recurrent neural network and specifically a long short--term memory (LSTM) model, the detection score of the correct activity category or the detection score between the correct and incorrect categories should be monotonically non-decreasing as the model observers more of the activity. Therefore, their model suggests the design of ranking losses to penalize the model on violation of such monotonicities, which are used together with classification loss in training of LSTM models. Finally, the model of Li et al. (2016) proposes a joint classification regression recurrent neural network for online human action recognition from 3D skeleton data. The model applies the deep long short-term memory (LSTM) subnetwork to capture the complex long-range temporal dynamics and avoid the sliding window.

Among other neural networks for action recognition are ones proposed by Parisi et al. (2015), Parisi et al. (2017). The method in Parisi et al. (2015) proposes a neurobiologically motivated approach for noise-tolerant action recognition in real time. Their system first extracts pose, and motion features of the action obtained from depth maps video sequences and later classifies the actions based on the pose motion trajectories. A two-pathway hierarchy of growing when required (GWR) networks process pose motion features in parallel and integrate action cues to provide movement dynamics in the joint feature space. Then, the GWR implementation is extended with two labeling functions to classify the action samples into the action categories. In another study, Parisi et al. (2017) proposed deep neural network self-organization for life-long action recognition. The system utilizes a set of hierarchical recurrent networks for unsupervised learning of action representations with increasingly spatiotemporal receptive fields instead of handcrafted 3D features. The growth and the adaptation of the recurrent networks are driven by their ability to reconstruct temporally ordered input sequences, and this makes the lifelong learning possible for the system. The visual representation obtained from unsupervised learning is associated with the action labels to satisfy the classification purposes.

This article presents instead a biologically inspired cognitive architecture that categorizes an ongoing action in an online mode. This means that the system receives information about an ongoing event such as body postures or object trajectories and continuously analyzes the incoming data in order to categorize the action performed. To this end, the system needs to be capable of making an automatic segmentation together with categorization while different actions are sequentially performed. This is in contrast to the methods proposed by Wang et al. (2015), Parisi et al. (2015), Parisi et al. (2017), Liu et al. (2017), Hou et al. (2016), Ijjina and Mohan (2016), which rely on pre-segmented datasets of actions.

On the contrary to the methods presented by Ellis et al. (2013), Vieira et al. (2012), Lv and Nevatia (2006), the approach proposed in this article does not utilize a memory to preserve any previous frames since a trained SOM can connect consecutive features and as a result determines whether the coming frames belong to a particular action or not.

In contrast to the deep neural network-based approaches for online action recognition (Weinzaepfel et al. 2015; Peng and Schmid 2016; Shou et al. 2016; Singh et al. 2016; Ma et al. 2016; Dave et al. 2017), which utilize 2D RGB images sensitive to illumination variations, color and texture changes, the method presented in this article uses skeleton data robust to scale and illumination changes and provides us with rich 3D structural information.

Next, it comes with a description of how the biologically inspired cognitive architecture proposed in this article performs online recognition of actions. A more thorough comparison of the approach proposed in this article with other related methods in the literature is available in "Discussion" section.

Proposed approach for online action recognition

One can view a particular event as being composed of a number of key components so that when the components are presented to the system in the right order, it can correctly categorize the event. As an example, consider the event of drinking a cup of coffee. In this case, the key components could be ordered as follows: the hand reaches the cup, lifts it up, brings it to the mouth and then puts it down. Based on situational factors including where the cup is located (on the table, on the floor, etc), the properties of the cup (size, weight, shape, etc) and who the actor is (gender, age, physical condition, etc), the details of the ordered key components of the event will vary. The difference between the instances lies in how these components are combined to complete the event. The categorization of the action can be a function of kinematic factors such as position, speed, acceleration and the rate of performing a particular event.

It seems that to solve the problem, the system needs to learn the occurrence of forthcoming key components. For instance, take the earlier example of head scratching. The system tracks the movements of the actor that starts with lifting the arm. At this stage, more than one possible forthcoming action can be predicted, e.g., head scratch, high arm wave, look at watch, forward punch, etc. Since there is more than one possible categorization, the system requires more information (key components) of the action performed to make a final decision. When the forthcoming movements fit more with the key components of initially possible actions, for example, touching the head then the observer can more confidently categorize the action.

Using the key frames is also proposed for semantic segmentation in Li et al. (2017) in order to reduce the computational burden for video streams and improve the real-time performance. In their approach, the convolutional neural network is utilized with spatial stream represented by images and temporal stream represented by image differences as their inputs.

In this article, a cognitive architecture based on hierarchical self-organizing maps (SOM) consisting of two-layer SOMs together with a one-layer supervised neural network is used. The system contains a preprocessing unit consisting of ego-centered coordinate transformation, scaling and attention mechanism. The first-layer SOM is used to extract the features of each sequence of an action through observation of the preprocessed input data from a Kinect camera. The features are presented as the activation of neurons in the first-layer SOM. The key activations representing the actions are segmented by using a sliding window of fixed size and transferred to the second-layer SOM in order to cluster the second map into action categories. Finally, the third layer labels the categories that are formed in the second SOM and outputs the action names.

Here, the SOM is used for both feature extraction and pattern classification. Using the three layers of neural network in the hierarchical action recognition architecture introduces an online semi-supervised learning model (Ding et al. 2017), which resembles the human learning process in which the training samples are often obtained successively. In this way, the observations arrive in sequence and the corresponding labels are presented very sporadically.

The main contributions of this article are listed as following:

(1) This article proposes a novel approach for online recognition of unsegmented action sequences inspired by humans's event perception and segmentation.
(2) The proposed approach is developed in a hierarchical cognitive architecture for action recognition. Different layers of the architecture are inspired by the biological organisms such as the preprocessing layer and the SOM layers.
(3) Although the performance of the system drops slightly in online experiments, the system remains highly accurate in performing the task online, which is more plausible and practical in real-case scenarios.

Methods

The multilayer architecture shown in Fig. 1 is composed of several processing layers. The following section describes the implementations of the main layers such as preprocessing layer, SOM layers and one layer supervised neural network. More explanations regarding each layer are available in the earlier works (Gharaee et al. 2017a, c).

Basic hierarchical SOM architecture

*Preprocessing:*The input data of an action performer are transformed into an ego-centered coordinate system located in the joint stomach. The 3D information of the joints right hip, left hip and stomach is used to build the ego-centered coordinate system, and all skeleton joints 3D information are transformed into this new coordinate system in order to compensate for having different viewing angles in relation to the Kinect camera. A detailed description of the ego-centered coordinate transformation is presented in Gharaee et al. (2017a) and Gharaee (2020).

To compensate for the different distances to the Kinect camera, a scaling function is also applied to the input data. By transforming the skeleton postures into a standard size, the representations of the actions performed by the actor remain invariant of its different distances to the Kinect camera and as a result are set to a standard size.

Finally, an attention mechanism is applied to the input data in order to extract the parts of the body that are most active. The attention mechanism used in this architecture is inspired by human behavior, paying attention to the most salient parts of a scene, which in this case is the part of the body that moves the most during a particular action. To extract the active joints while performing an action, the joints velocity is utilized and the attention mechanism selects the four most moving joints. As a result, by reducing the dimensionality of the input data in this way, processing power and time is saved. Detailed descriptions of the preprocessing modules used in this architecture are available in Gharaee (2020).

*SOM-Layers:*A SOM consists of an $I \times J$ grid of neurons with a fixed number of neurons and a fixed topology. Each neuron n_{ij} is associated with a weight vector $w_{ij} \in R^n$ having

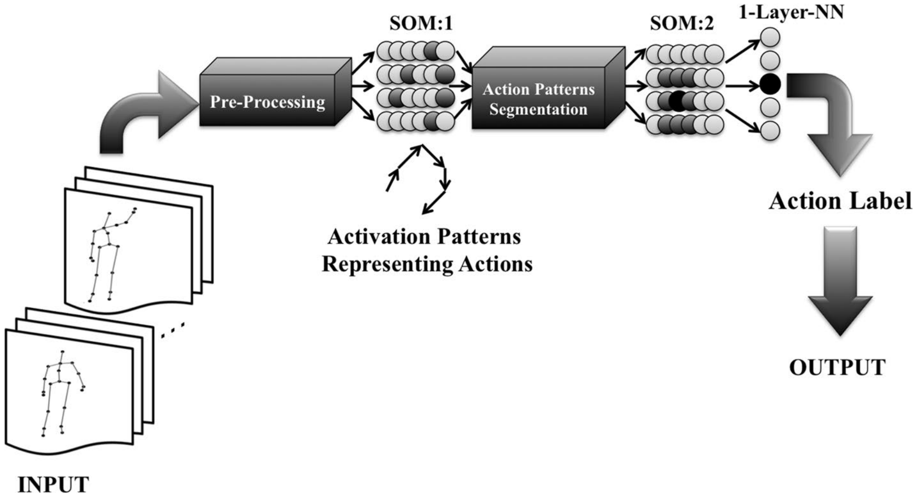
SOM:1
1-Layer-NN
SOM:2
Pre-Processing
Action Patterns Segmentation
Activation Patterns Representing Actions
Action Label
OUTPUT
INPUT

Fig. 1 Three-layer action recognition architecture. The first and second layers consist of a SOM, and the third layer is one-layer supervised neural network. The darker activation in the first SOM represents the activity trace during an action, which is also shown as patterns made of arrows. The darker activation in the second SOM shows an example of a clustered region belonging to an action with stronger activation effect, and in the center of the region is the most activated neuron of the whole map, shown in black. The third layer (1-Layer-NN) is composed of the same number of neurons as the number of actions, and the darker activation shows the action that the system recognized

the same dimension K as the input vector $x(t)$. Each element of the weight vector is represented by three dimensions, i, j and k, where $0 \leq i < I$, $0 \leq j < J$, $i,j \in N$ represent the corresponding row and column of a neuron in the grid and $0 \leq k < K$ is equal to the input dimension. For a squared SOM with equal number of rows and columns, the total number of neurons is $N \times N$. All elements of the weight vectors are initialized by real numbers randomly selected from a uniform distribution between 0 and 1.

At time t, each neuron n_{ij} receives the input vector $x(t) \in R^n$. The net input $s_{ij}(t)$ at time t is calculated using the Euclidean metric:

$$s_{ij}(t) = ||x(t) - w_{ij}(t)|| \tag{1}$$

The activity $y_{ij}(t)$ at time t is calculated by using the exponential function for each neuron of the grid:

$$y_{ij}(t) = e^{\frac{-s_{ij}(t)}{\sigma}} \tag{2}$$

The parameter σ is the exponential factor set to 10^6. The role of the exponential function is to normalize and increase the contrast between highly activated and less activated areas.

The neuron c with the strongest activation or the winner is selected because it represents the most similarity to the input vector. The weight vectors of all neurons w_{ij} are adapted by using a Gaussian function centered at a winner neuron, c:

$$c = \text{argmax}_{ij} y_{ij}(t). \tag{3}$$

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(t) G_{ijc}(t)[x(t) - w_{ij}(t)]. \tag{4}$$

The term $0 \leq \alpha(t) \leq 1$ shows the adaptation strength in which $\alpha(t) \rightarrow 0$ when $t \rightarrow \infty$. The neighborhood function $G_{ijc}(t) = e^{-\frac{||r_c - r_{ij}||}{2\sigma^2(t)}}$ is a Gaussian function decreasing with time, and $r_c \in R^2$ and $r_{ij} \in R^2$ are location vectors of neurons c and n_{ij}, respectively. All weight vectors $w_{ij}(t)$ are normalized after each adaptation. Thus, the winner neuron receives the strongest adaptation and the adaptation strength decreases by increasing distance from the winner. As a result, the further neurons are from the winner the more weakly their weights are updated.

Output-Layer The output layer of the architecture is one-layer supervised neural network, which receives the activity traces of the second-layer SOM as the input vector with length L. The length L is determined by the total number of neurons of the second-layer SOM. The output layer consists of a vector of N number of neurons and a fixed topology. The number N is determined by the number of action categories. As an example in the first experiment of this article, the number of neurons of the output layer is set to 10, which is the number of actions categories.

Each neuron n_i is associated with a weight vector $w_i \in R^n$, and each element of the weight vector is represented by two dimensions $0 \leq i < N$, $0 \leq l < L$, where all the elements of the weight vector are initialized by real numbers randomly selected from a uniform distribution between 0 and 1, after which the weight vector is normalized, i.e., turned into unit vectors.

At time t, each neuron n_i receives an input vector $a(t) \in R^n$. The activity y_i in the neuron n_i is calculated using the standard cosine metric:

$$y_i = \frac{a(t) \cdot w_i(t)}{||a(t)||||w_i||} \tag{5}$$

During the learning phase, the weights w_i are adapted by

$$w_i(t+1) = w_i(t) + \beta a(t)[y_i - d_i] \tag{6}$$

The parameter β is the adaptation strength, and d_i is the desired activity for the neuron n_i.

Action pattern segmentation

This section describes the technique utilized in hierarchical SOM architecture to implement the system for online real-time experiments with unsegmented input data of action samples. The module implemented to apply this technique receives the output vector patterns of the first-layer SOM and creates the input signal to the second-layer SOM. To this end, it extracts the activations of the first SOM, which are elicited as a result of receiving the key postures of an ongoing action sample as the input of the system. In this way, the output pattern vectors of the first-layer SOM corresponding to the input action samples are segmented. Thus, the segmentation occurs automatically one step further into the system where the action feature vectors are created and segmented instead of input posture frames.

Each action sequence is represented by the the consecutive posture frames, while each posture frame is composed of the 3D skeleton joints positions. The consecutive posture frames are applied to the system as the input.

The kinematics of actions are determined by the spatial trajectory of human skeleton components (like the joints) during the time interval the action performs. Temporal features are specified by the length and the order. The length is represented by the time interval during which the action performance is completed, and the temporal order is represented by the sequential orders of the movements. As an example, the action *high arm wave* is performed by the left arm of the actor as represented in Fig. 2 part a. The left arm is composed of the joints left shoulder, left elbow, left wrist and left hand. The 3D temporal characteristics of these joints are presented in Fig. 2 parts b, c and d. As shown there, the hand and the wrist have almost similar spatial trajectories throughout time but on different scales. (The smaller one is for the wrist and the larger one is for the hand). The elbow

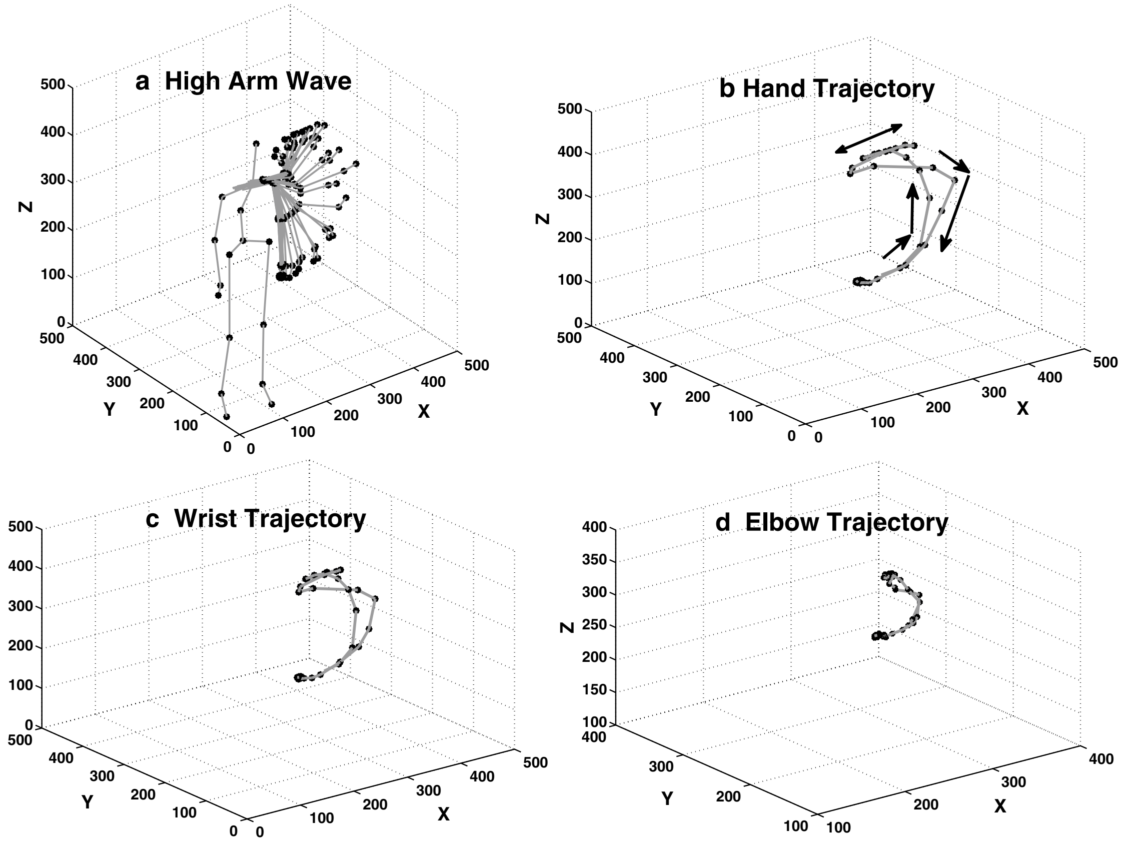

Fig. 2 3D spatial trajectories of the body parts involved the most in performing the action *high arm wave*. The performed action with the left arm is shown in part a as consecutive body postures. The spatial trajectory of the hand joint with directions of the movements is shown in part b. The spatial trajectories of the wrist and elbow joints are shown in parts c and d respectively

has a much smaller movement during acting compared to the hand and the wrist. The shoulder movement is even more limited and is thus not presented in Fig. 2.

Both the kinematic and dynamic characteristics of the action are crucial for perceiving it, and they introduce the spatiotemporal features of the action. There are actions distinguished from one another only by one of these characteristics. For example, the actions *lift up* and *put down* have similar posture frames of the arm movements representing their kinematics but with completely reversed temporal order. Therefore, the temporal order of the posture frames is the discriminating factor for these two actions.

On the other hand, if an action is seen as a number of key components, for example, in the action *horizontal arm wave* these key components can be lift the arm up, move the arm to the left/right direction, move the arm back to the reverse direction (right/left) and put the arm down. Based on the speed of performing the action, each component can contain a number of posture frames that are similar.

The spatiotemporal trajectory of an action extracted from consecutive 3D body postures of that action is received by the first-layer SOM, and they activate specific areas of the map representing the input space. Pattern vectors are formed by connecting these ordered activations of the performing action. Let us assume that there is a distinct elicited activation for each posture of an action sequence. Then, as a result there will be a series of elicited activations for the whole action sequence. So the key components of the action can relate to the key postures as well as the key elicited activations in the SOM.

Thus, action sequences are segmented by extracting and segmenting the key activations in the first-layer SOM. In the left top part of Fig. 3, all consecutive postures of the action *hand catch* are shown and in the right top part, those postures with key activations in the first-layer SOM (key postures) are depicted. The right bottom part of Fig. 3 shows the key activations of the SOM (action pattern) corresponding to the same action sample.

The same action can have a completely different visual appearance. Variations can occur because of performing speed, clothing, etc. Based on how an action is performed and the nature of the actions, the length of elicited key activations in the first-layer SOM will be different. If an action is performed too slowly, the number of similar consecutive

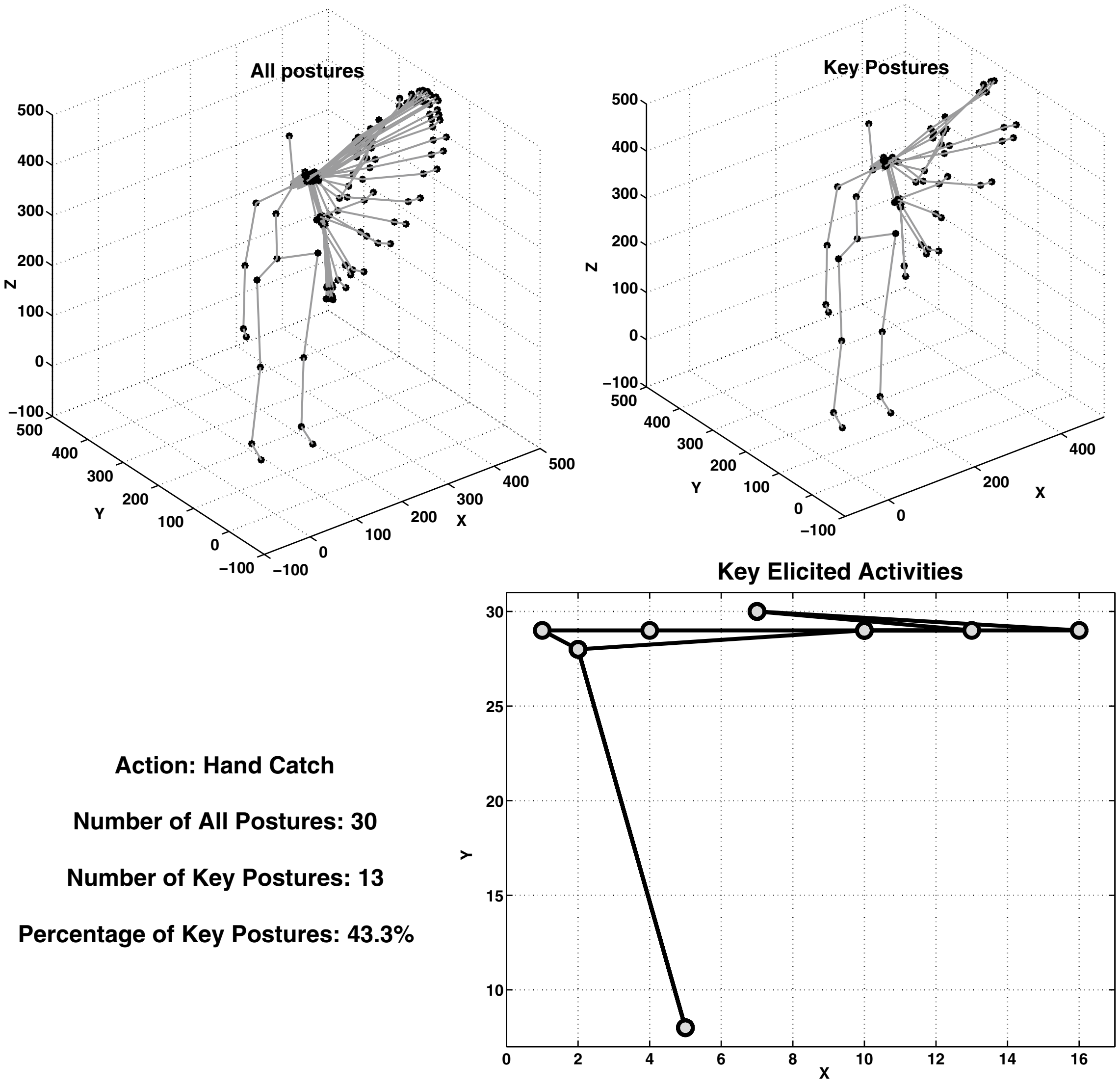

Fig. 3 Top left of the figure shows all consecutive posture frames while performing the action *hand catch*. The top right part of the figure shows only those postures with unique activation in the first-layer SOM extracted by the action patterns segmentation unit, and the bottom right part of the figure shows the action pattern as a result of key activations elicited in the first-layer SOM. In the bottom right part of the figure, the activated neurons are represented by circles and some of them are activated more than once in non-consecutive iterations

postures will increase, but each posture of an action sample is considered as a key posture if and only if it elicits a unique activity in the SOM.

Next there is a detailed description of how the proposed approach to recognizing unsegmented action sequences is developed in hierarchical SOM architecture. Let us start with input space of the actions, which is composed of action sequences:

$$input = \left\{ s_1, s_2, s_3, \dots, s_i, \dots, s_{N_s} \right\}, \tag{7}$$

where $0 < i < N_s$ and N_s are the total number of action sequences of the dataset, which is 276 for the first experiment. Each action sequence s_i is composed of the consecutive posture frames:

$$s_i = \left\{ p_1, p_2, p_3, \dots, p_j, \dots, p_{N_p} \right\}, \tag{8}$$

where $0 < j < N_p$ and N_p are the total number of posture frames representing an action sequence and it varies for different action sequences. A posture frame p_j contains spatiotemporal features represented by 3D information of the skeleton joints:

$$p_j = \left\{ d_1, d_2, d_3, \dots, d_k, \dots, d_{N_d} \right\}, \tag{9}$$

where $0 < k < N_d$ and N_d are the full dimension of spatiotemporal features. For 3D features of the skeleton joints positions, N_d can represent the total number of identified joints in 3D. As an example if 20 joints are extracted from each posture frame, then $N_d = 20 \times 3 = 60$.

After some preprocessing, the consecutive spatiotemporal feature vectors represented by posture frames p_i are received by the first-layer SOM. The activity traces of the first-layer SOM are extracted as 2D positions of the activated neurons. The consecutive elicited activities of each action sequence are:

$$a_i = \left\{[x_1, y_1]_i, [x_2, y_2]_i, \dots, , [x_q, y_q]_i, \dots [x_{L_i}, y_{L_i}]_i\right\}, \tag{10}$$

where $0 < i < N_s$ and $[x_q, y_q]$ show the location of a neuron on the 2D neural map. The term $0 < q < L_i$ and L_i shows the full length of a pattern vector varying for different action sequences. The consecutive activity pattern vector of all action sequences is:

$$A = \left\{a_1, \quad a_2, \quad a_3, \dots, \quad a_i, \dots, a_{N_s}\right\}. \tag{11}$$

At time t, 2D locations of the elicited activation of first-layer SOM, $a_{in}(t) = [x_t, y_t]$, are received as the input of pattern vector segmentation layer. Based on how the actions are performed, there are similar consecutive elicited activation representing similar posture frames of the action sequence. Such similar consecutive activity traces are first mapped into a unique activation. Then, a constant length of segmentation T is applied to segment the action sequences, which determines when the elicited activations representing the action starts/ends. The segmented vector, which is the result of receiving real-time action sequences, is:

$$\begin{aligned} a_{out} = \{&[x_0, y_0], [x_1, y_1], \dots [x_T, y_T], \\ &[x_{T+1}, y_{T+1}], [x_{T+2}, y_{T+2}], \dots, [x_{2T}, y_{2T}], \\ &[x_{2T+1}, y_{2T+1}], [x_{2T+2}, y_{2T+2}], \dots, [x_{3T}, y_{3T}], \dots, [x_{N_tT}, y_{N_tT}]\}, \end{aligned} \tag{12}$$

where T is a constant value used for segmentation and N_t is the total number of segmented vectors. The segmentation size is calculated from the average length of the key activity traces for all action sequences of the training dataset. The same size is applied to the activity traces of all action sequences in both training and test dataset. The action pattern vectors should be segmented in a way to encompass key activations of all action sequences, so ideally it should not be too large to not contain the key activations of more than one action sequence and at the same time, it should not be too small so that it does not ignore the key activations of a single action sequence.

The size of segmentation is usually determined empirically in the experiments. By training the system for several trials, the segmentation size might be updated. As an example in the first experiment of this article, the longest and the shortest activity traces for a subset of dataset contain 59 and 12 key activations, respectively, and after some tuning the segmentation size T is set to 30.

Results

The performance of the architecture shown in Fig. 1 is evaluated in two experiments. In these experiments, two kinds of input data are used. First, the architecture is tested on a publicly available dataset called MSR-Action3D dataset (Wan 2015). The MSR-Action3D is the first public benchmark RGB-D set collected by a Kinect sensor, and it provides us with the skeleton data of the actions performed. For the second experiment, a new dataset is collected, which is composed of 3D postures of a human actor performing actions using a Kinect sensor to validate the system in online experiments.

The neural modeling framework Ikaros (Balkenius et al. 2010) has been used to implement the architecture and also to perform the experiments. The results were filmed, and demo movies were created for both experiments. The movies are accessible on the Web page [21].

Experiment 1

In the first experiment, the ability of the hierarchical SOM architecture to categorize actions is tested in an online mode by using a dataset of actions composed of sequences of 3D joints postures. This dataset contains 276 samples with ten different actions performed by ten different subjects in two to three different events. Each action sample is composed of a sequence of frames where each frame contains 20 joint positions expressed in 3D Cartesian coordinates. The actions of the first experiment are: *high arm wave*, *horizontal arm wave*, *using hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *tennis swing*.

All action samples of the MSR-Action3D dataset are segmented and labeled with the names of the corresponding action, subject and event. Thus, to run the first experiment in online real-time mode the system is provided with random selection of unsegmented data of consecutive actions as the input and is using the labeling information to validate the system performance by comparing whether the recognized action by the architecture correctly matches the real action performed.

For this experiment, the dataset was split into a training set containing 80% of the action instances randomly selected from the original dataset and a generalization test set containing the remaining 20% of the instances. Then, the neural network system was trained with randomly selected instances from the training set in two phases, the first to train the first-layer 30 × 30 neurons SOM and the second to train the second-layer 35 × 35 neurons SOM and the output-layer containing ten neurons.

In order to make the result invariant of the order of action sequences, different random selections of test samples are applied to the trained system and the average categorization results of all test samples of each action are shown in Fig. 4. This process is repeated for different random selections of training and test samples from all action sequences.

As shown in Fig. 4, the actions are correctly categorized already after a few iterations from when their input patterns are applied to the second SOM. One iteration counts when a unique posture frame is received by the first-layer SOM. Naturally, it takes some iterations for the input pattern to cover all the corresponding key activations as a result of first-layer SOM receiving key postures of the corresponding action. The correct categorization continues for several iterations, and then, it shifts to zero because of the updating process, which occurs in the activation of the neuron representing the correct action performs. During updating process, system starts building an input sequence pattern for a new action. In this experiment, the average performance of 75% correct categorization is obtained for the generalization test data when the segmentation technique is used.

A certainty measure is also used in which the online categorization of the action performed is given as output if it continues for a number of consecutive iterations, which is between five and seven iterations (having around 80% of the action input). This is done to achieve a more stable and robust recognition result. For this, it is necessary that the sliding window contains the key activations of an action sequence continuously for a number of consecutive iterations, which is a restriction for the categorization task. The 75% correct categorizations should be compared to 83% correct that was obtained (Gharaee et al. 2017b) when the dataset was segmented in advance in an offline experiment. The results show that the performance drops when the segmented action pattern vector of fixed length is used, but at the same time the system is capable of running online experiments.

What makes the categorization task difficult is when the actions have similar components. As an example, take the actions *high arm wave*, *using hammer*, *hand catch*, *high*

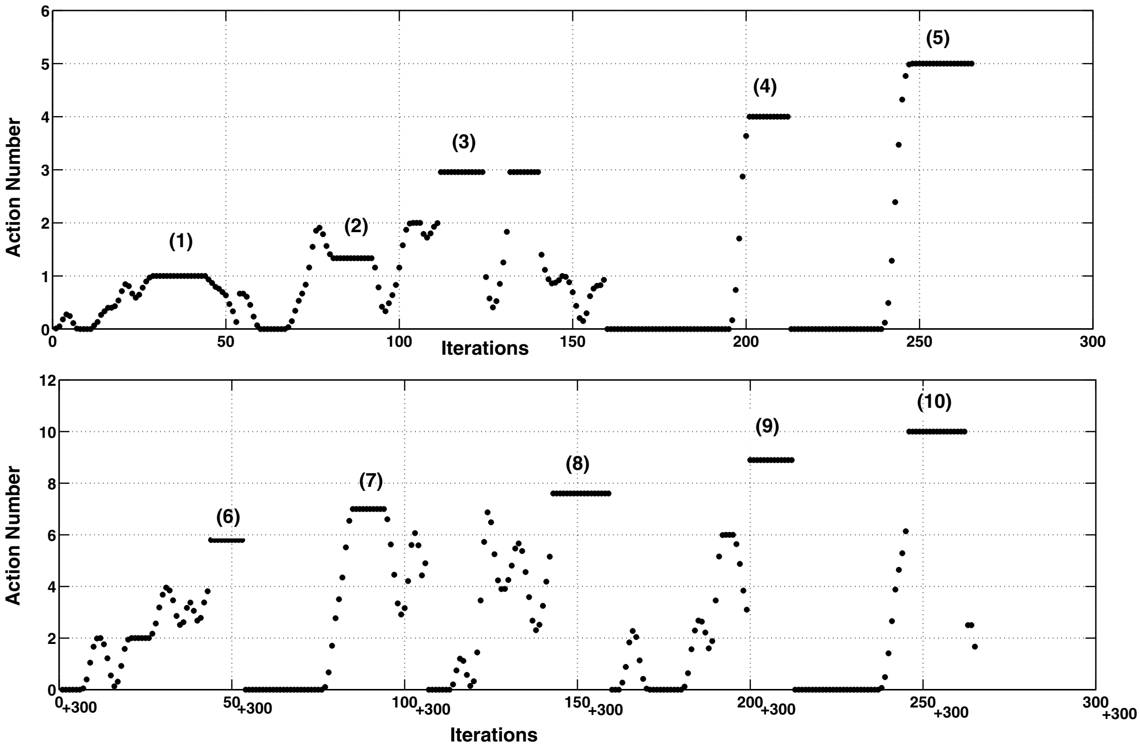

Fig. 4 Online action recognition experimental results obtained through application of segmentation on MSRAction3D dataset. The upper row shows the recognition results of the actions: *high arm wave (1)*, *horizontal arm wave (2)*, *using hammer (3)*, *hand catch (4)* and *forward punch (5)*, while the lower row depicts the recognition results of the actions: *high throw (6)*, *draw x (7)*, *draw tick (8)*, *draw circle (9)* and *tennis swing (10)*. One iteration counts when a unique posture frame is received by the first-layer SOM. The average recognition accuracy corresponding to the action performed is calculated and multiplied by the action number and plotted

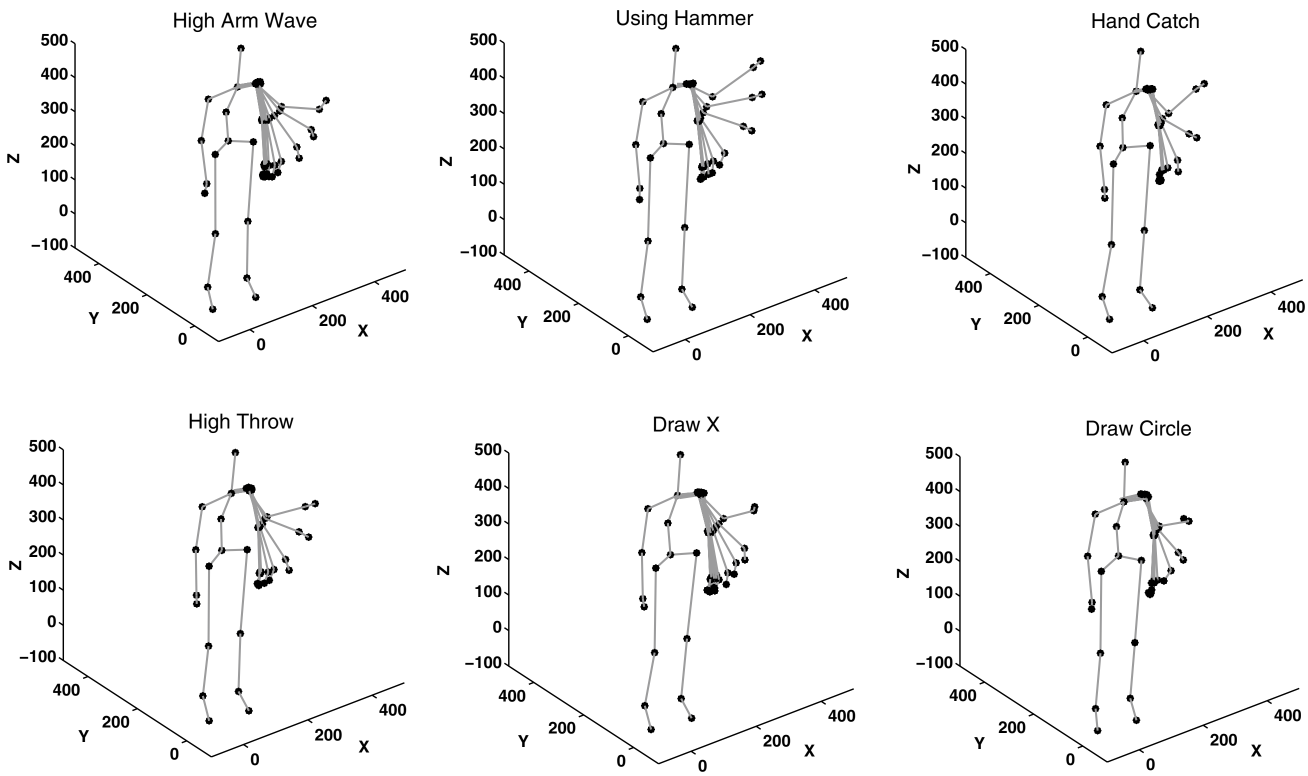

Fig. 5 First postures of the six actions performed in order to represent the similarities of these postures that belong to distinct actions

throw, *draw X* and *draw circle*. In all of these actions, the first postures begin with lifting the arm up. It can be seen as the same movement although the actions are different. Figure 5 shows a number of beginning postures of these actions, and the plotted postures are similar even though they belong to distinct actions. Therefore, the system should categorize them as different categories to distinguish them. The similarities also increase the delay of the system before it makes the right guess as to the action performed.

Experiment 2

In the second experiment, the online experiment presented in Gharaee et al. (2016) is developed by implementing the segmentation technique proposed in this study. Then, a dataset of actions composed of 3D joints postures is collected by using a Kinect sensor. The actions of the second experiment are: *high arm wave*, *forward punch*, *draw x*, *draw circle* and *tennis swing*.

In this experiment, the same architecture is used with similar preprocessing layer to the input data. The system is trained on a dataset containing 60 sequences of five different actions in which there are 12 different sequences of each action. The actions are performed by a single actor in two to three different events.

After the system has been trained on the dataset with the result of 100% correct categorization accuracy, an online real-time generalization test experiment is performed on the trained weights of the system in which the actor performed similar actions in front of a Kinect camera. The actions are selected randomly and performed in several trials. As a result of this experiment, an average accuracy of 88.74% correct categorization for the generalization test experiments is obtained.

The proposed segmentation technique makes it possible to run online experiments. It means that the system is capable of categorizing actions continuously as the actions are performed. This is different and more difficult compared to the condition when the system is tested in offline mode on pre-recorded datasets. For human–robot interaction and many other applications of action recognition, it

Table 1 Performance of online action categorization task in the real-time experiments by segmenting action patterns

	Offline test experiment with segmented actions (%)	Online test experiment with unsegmented actions (%)
Experiment.1	83	75
Experiment.2	94.29	88.74

is mandatory to be able to perform the categorization in an online mode.

In this study, the aim has been to show the capacity of the SOM architecture in online implementations. By running the experiments online, the system accuracy drops slightly, as shown in Table 1, compared to when pre-segmented data are given, but the practicality of the online mode outweighs the drop.

Discussion

In this article, an action recognition task is performed in online real-time experiments. Therefore, the segmentation problem in dealing with the datasets of unsegmented action sequences needs to be solved. The segmentation problem addressed in this article is related to detection of the start and/or end of the action performed in a time series of consecutive action sequences.

The simple way is to manually segment the untrimmed videos of action sequences, which is highly expensive. Most of the methods such as (Wang et al. 2015; Parisi et al. 2015, 2017; Liu et al. 2017; Hou et al. 2016; Ijjina and Mohan 2016) rely on pre-segmented datasets of actions, and thus, they are evaluated with respect to the benchmarks containing labeled action sequences.

Among the methods for online action recognition tasks such as STOP feature vectors in Vieira et al. (2012), canonical poses in Ellis et al. (2013) and feature spaces of a joint or related multiple joints in Lv and Nevatia (2006), there are certain features extracted first. Then, these features are applied to a particular classifier to be categorized. An important question concerns to what degree the extracted features represent the action sequences in other words, how much information is lost in the data compression of the features.

The feature extraction in some of these methods is performed for fixed time intervals, for example every five frames in Vieira et al. (2012) or between 10 and 30 previous frames in Ellis et al. (2013). A limitation of this approach is that it requires having access to a certain number of frames for each iteration, which necessitates a capacity to preserve previous information and may also result in a delay in achieving results.

In contrast to the methods proposed in Ellis et al. (2013), Vieira et al. (2012), by learning the sequential relation of the consecutive posture frames, the approach proposed here is independent of allocation of memory to preserve any previous frames since a trained SOM can connect consecutive features through connecting consecutive activated neurons of the lattice, and as a result determine whether coming frames belong to a particular action or not.

The system produces the action label when it perceives a number of consecutive key frames, which is less than T for the majority of the action sequences. There is no preset delay considered in the structure of the system, and the delay occurs mainly due to the fact that the certainty of the system increases when more key features are observed. Because of this, there is less delay in obtaining a categorization response from the system. This aspect accords with human action categorization. As an example, when a person lifts up his arm he might want to look at his watch, scratch the head or wave to greet a friend. Thus, one cannot recognize what he is doing until more key components of the action are received.

In the approach proposed in this article, instead a neutral pause between actions is not employed as done in Vieira et al. (2012), so the whole stream of actions is applied as the input of the system. Moreover, the allocation of both the start and the end of each action sequence could be critical and not only the start of the action, as has been proposed in Ellis et al. (2013). The hierarchical SOM system addresses the problem of allocating the beginning and ending of the actions by learning the sequential relations between the consecutive frames so that when it receives two consecutive frames, it can detect whether they belong to the same action or not.

Although the system proposed in Lv and Nevatia (2006) is claimed to be capable of automatic recognition and segmentation of 3D human actions, it is not clear how this system works in online experiments because it is tested on a collected MoCap dataset of actions. In contrast, this study presents an online experiment by using a Kinect sensor in real time and the ability of the system is tested on new data of online actions.

The neural network-based approaches for online recognition of actions proposed in Weinzaepfel et al. (2015), Peng and Schmid (2016), Shou et al. (2016), Singh et al. (2016), Ma et al. (2016), Dave et al. (2017) use the RGB images as the input modality. Although the RGB images provide input data with rich characteristics of shape, color and texture, they are 2D images sensitive to illumination variations, color and texture changes, so the performance of the task is largely dependent on the quality of the input images. In fact, it is quite expensive to produce and use high-quality images of actions, since this requires expensive cameras for data collection and the dataset produced by these cameras contains high-resolution images of large dimensionality, which necessitates more time and processing power for data analysis. Another limitation of these approaches is that most of deep learning methods rely on largely labeled training data, which add even more cost in running the experiments.

On the other hand, the skeleton data are robust to scale and illumination changes and provide us with rich 3D structural information of the scene by calculating the positions of the human joints in 3D space as a high-level feature representing the kinematics and dynamics of the actions.

Furthermore, skeleton data can be invariant to human body rotation and speed of the motions. Similar to the method proposed by this article, the skeleton information is utilized in an online action detection approach based on joint classification regression recurrent neural network (see Li et al. 2016). This method utilizes the 3D joints input similar to the method proposed by this article. The proposed approach of Li et al. (2016) is tested on an input set of actions containing ten different actions and obtains average recognition accuracy of 65%. Although different types of action input are used to test the model proposed by Li et al. (2016) and the model proposed in this article, the proposed model of this article obtains better results. As shown in the result section, my architecture obtains overall recognition accuracy of 75% and 88.74% in online test experiments on two action sets containing ten and five different actions.

The proposed architecture here is capable of recognizing actions in online experiments. The system extracts key features of each action sequence, represented as a pattern vector, and uses the learned vector as the representative of that action sequence. First-layer SOM of the architecture learns the consecutive postures of actions and extracts the action patterns while the second SOM classifies the segmented patterns into action categories. To increase the categorization certainty, it is checked whether the system sends the same action as its output during a few iterations and, if so, that action is considered as the output of the system. By using this method, the performance accuracy drops slightly compared to pre-segmented data, but the certainty of the system is improved.

Using cognitive mechanisms such as attention makes the system more biologically plausible. Since the attention mechanism is inspired by human behavior, paying attention to the most salient parts of a scene, in this case, is the part of the body that moves the most during a particular action. As a result, by reducing the dimensionality of the input data in this way, processing power and time is saved and at the same time, the performance of the action recognition system significantly increased (see also Gharaee et al. 2014).

It should be mentioned that in the approach presented in this article, the system is trained on limited dataset of labeled action sequences and it is able to generalize, i.e., the network can recognize or characterize input it has never encountered before.

The results of the experiments performed in this study show that the recognition of unsegmented actions in online test experiments is quite high. When the performance results of this paper are compared to our earlier empirical studies, whether they are online experiments (Gharaee et al. 2016, 2017c) or offline experiments (Gharaee et al. 2017b, a), there is a decrease in the acquired recognition accuracy of the system. An explanation for this may be that different sequences of actions span different time intervals, while a constant cutoff length is allocated to all of them through the sliding window.

One limitation of using 3D skeleton data is with the reduction in accuracy due to the environmental noise and the transformation of different modalities. Another limitation with the method proposed in this article is with setting the size of segmented action pattern vectors. It requires computational effort to find out the best size for action pattern segmentation for the training data. On the other hand, if the test actions performed too different from the training samples, the recognition accuracy might drops.

Conclusion

In summary, this article proposes a new method for human action recognition and segmentation using a SOM-based system. The hierarchical architecture consists of two-layer SOMs together with one-layer supervised neural network. The system is validated on different experiments. First, the system is tested on the MSR-Action3D dataset, and then, it is validated on a dataset collected by a Kinect sensor in online experiments.

In order to improve action recognition and segmentation performance, it is planned plan to design and implement a sliding window of the pattern vectors with variable size that is adapted to the actual size of the key activations of the corresponding action sequence. Another plan would be to develop a method for solving the segmentation problem that calculates the prediction error between the actual forthcoming movement of the actor and the predicted one by the system while an action is performed and use this error value to determine when the action ends and a new action begins. To this end, the associative SOM (see Hesslow 2002 and Johnsson et al. 2009) for action recognition and segmentation could be used.

Acknowledgements Open access funding provided by Lund University. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP). The author would like to thank Prof. Peter Gärdenfors and Dr. Magnus Johnsson for their comments to improve this article.

Compliance with ethical standards

Funding The author has been supported by Lund University, Cognitive Science.

Conflict of interest The author of this article, Zahra Gharaee, declares that she has no conflict of interest.

Informed consent Informed consent was obtained from all individual participants included in the study.

Ethical standards All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards.

Human and animal rights This article does not contain any studies with animals performed by the author.

References

Balkenius C, Morén J, Johansson B, Johnsson M (2010) Ikaros: building cognitive models for robots. Adv Eng Inform 24(1):40–48. https://doi.org/10.1016/j.aei.2009.08.003

Dave A, Russakovsky O, Ramanan D (2017) Predictive-corrective networks for action detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.1109/CVPR.2017.223

Ding S, Xi X, Liu Z, Qiao H, Zhang B (2017) A novel manifold regularized online semi-supervised learning model. Cognit Comput. https://doi.org/10.1007/s12559-017-9489-x

Dollar P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse spatio-temporal features. In: IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance, pp 65–72. https://doi.org/10.1109/VSPETS.2005.1570899

Ellis C, Masood SZ, Tappen MF, Laviola JJ Jr, Sukthankar R (2013) Exploring the trade-off between accuracy and observational latency in action recognition. Int J Comput Vision 101:420–436

Gärdenfors P (2007) Representing actions and functional properties in conceptual spaces. In: Body, language and mind, vol 1, pp 167–195. Mouton de Gruyter, Berlin

Gärdenfors P, Warglien M (2012) Using conceptual spaces to model actions and events. J Seman 29:487–519

Gharaee Z (2018a) Action in mind: a neural network approach to action recognition and segmentation. Cognitive science. Lund University, Lund

Gharaee Z (2018b) Recognizing human actions by a multi-layer growing grid architecture. ICNN 2018: International Conference on Neural Networks, Prague, Czechia, 22–23 March 2018

Gharaee Z (2020) Hierarchical growing grid networks for skeleton based action recognition. Cogn Syst Res 63:11–29. https://doi.org/10.1016/j.cogsys.2020.05.002

Gharaee Z, Fatehi A, Mirian MS, Ahmadabadi MN (2014) Attention control learning in the decision space using state estimation. Int J Syst Sci (IJSS) 47:1659–1674. https://doi.org/10.1080/00207721.2014.945982

Gharaee Z, Gärdenfors P, Johnsson M (2016) Action recognition online with hierarchical self-organizing maps. In: Proceedings of the international conference on signal image technology and internet based systems (SITIS). https://doi.org/10.1109/SITIS.2016.91

Gharaee Z, Gärdenfors P, Johnsson M (2017a) First and second order dynamics in a hierarchical som system for action recognition. Appl Soft Comput 59:574–585. https://doi.org/10.1016/j.asoc.2017.06.007

Gharaee Z, Gärdenfors P, Johnsson M (2017b) Hierarchical self-organizing maps system for action classification. In: Proceedings of the international conference on agents and artificial intelligence (ICAART). https://doi.org/10.5220/0006199305830590

Gharaee Z, Gärdenfors P, Johnsson M (2017c) Online recognition of actions involving objects. Biol Insp Cognit Archit (BICA) 22:10–19. https://doi.org/10.1016/j.bica.2017.09.007

Gibson JJ (1966) The senses considered as perceptual systems. Houghton Mifflin, Oxford

Gibson JJ (1979) The ecological approach to visual perception. Lawrence Erlbaum, Hillsdale

Hesslow G (2002) Conscious thought as simulation of behaviour and perception. Trends Cognit Sci 6:242–247. https://doi.org/10.1016/S1364-6613(02)01913-7

Hou Y, Li Z, Wang P, Li W (2016) Skeleton optical spectra based action recognition using convolutional neural networks. IEEE Trans Circuits Syst Video Technol. https://doi.org/10.1109/TCSVT.2016.2628339

Ijjina EP, Mohan CK (2016) Classification of human actions using pose-based features and stacked auto encoder. Pattern Recogn Lett 83:268–277. https://doi.org/10.1016/j.patrec.2016.03.021

Jalal A, Kim YH, Kim YJ, Kamal S, Kim D (2017) Robust human activity recognition from depth video using spatiotemporal multi-fused features. Pattern Recogn 61:295–308. https://doi.org/10.1016/j.patcog.2016.08.003

Johansson G (1973) Visual perception of biological motion and a model for its analysis. Percept Psychophys 14(2):201–211

Johnsson, M.: http://magnusjohnsson.se/

Johnsson M, Balkenius C, Hesslow G (2009) Associative self-organizing map. In: Proceedings of the international joint conference on computational intelligence (IJCCI), pp 363–370

Laptev I (2005) On space-time interest points. Int J Comput Vis 64:107–123

Li R, Gu D, Liu Q, Long Z, Hu H (2017) Semantic scene mapping with spatio-temporal deep neural network for robotic applications. Cognit Comput. https://doi.org/10.1007/s12559-017-9526-9

Li W, Zhang Z, Liu Z (2010) Action recognition based on a bag of 3d points. In: IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW), pp 9–14. https://doi.org/10.1109/CVPRW.2010.5543273

Li Y, Lan C, Xing J, Zeng W, Yuan C, Liu J (2016) Online human action detection using joint classification-regression recurrent neural networks. In: European conference on computer vision, vol 9911, pp 203–220

Liu M, Liu H, Chen C (2017) Enhanced skeleton visualization for view invariant human action recognition. Pattern Recogn 68:346–362. https://doi.org/10.1016/j.patcog.2017.02.030

Lv F, Nevatia R (2006) Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In: Proceedings of the conference on computer vision-ECCV, vol 5, pp 359–372. https://doi.org/10.1007/1174

Ma S, Sigal L, Sclaroff S (2016) Learning activity progression in lstms for activity detection and early detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.1109/CVPR.2016.214

Michotte A (1963) The perception of causality. Basic Books, New York

Oreifej O, Liu Z (2013) Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). https://doi.org/10.1109/CVPR.2013.98

Parisi GI, Tani J, Weber C, Wermter S (2017) Lifelong learning of human actions with deep neural network self-organization. Neural Netw. https://doi.org/10.1016/j.neunet.2017.09.001

Parisi GI, Weber C, Wermter S (2015) Self-organizing neural integration of pose-motion features for human action recognition. Front Neurorobot. https://doi.org/10.3389/fnbot.2015.00003

Peng X, Schmid C (2016) Multi-region two-stream R-CNN for action detection. In: European conference on computer vision, vol 9911, pp 744–759

Radvansky GA, Zacks JM (2014) Event cognition. Oxford University Press, Oxford

Schuldt C, Laptev I, Caputo B (2004) Recognition human actions: a local SVM approach. In: Proceedings of IEEE international conference on pattern recognition, vol 3, pp 32–36. https://doi.org/10.1109/ICPR.2004.1334462

Shotton J, Fitzgibbon A, Cook M, Sharp T, Finocchio M, Moore R, Kipman A, Blake A (2011) Real-time human pose recognition in parts from single depth images. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1297–1304. https://doi.org/10.1109/CVPR.2011.5995316

Shou Z, Wang D, Chang SF (2016) Temporal action localization in untrimmed videos via multi-stage CNNS. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1049–1058. https://doi.org/10.1109/CVPR.2016.119

Singh B, Marks TK, Jones M, Tuzel O, Shao M (2016) A multi-stream bi-directional recurrent neural network for fine-grained action detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.1109/CVPR.2016.216

Sun J, Wu X, Yan S, Cheong LF, Chua TS, Li J (2009) Hierarchical spatio-temporal context modeling for action recognition. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 2004–2011. https://doi.org/10.1109/CVPR.2009.5206721

Vieira AW, Nascimento ER, Oliveira GL, Liu Z, Campos MF (2012) Stop: space-time occupancy patterns for 3d action recognition from depth map sequences. In: Iberoamerican congress on pattern recognition , vol 7441, pp 252–259. https://doi.org/10.1007/978-3-642-33275-3-31

Wan YW (2015) MSR action recognition datasets and codes. http://research.microsoft.com/en-us/um/people/zliu/actionrecorsrc/. Accessed 2015

Wang J, Liu Z, Chorowski J, Chen Z, Wu Y (2012a) Robust 3d action recognition with random occupancy patterns. Springer, Computer Vision-ECCV p, pp 872–885

Wang J, Liu Z, Wu Y, Yuan J (2012b) Mining actionlet ensemble for action recognition with depth cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1290–1297

Wang L, Xiong Y, Lin D, Van Gool L (2017) Untrimmed nets for weakly supervised action recognition and detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. https://doi.org/10.1109/CVPR.2017.678

Wang P, Li W, Gao Z, Zhang J, Tang C, Ogunbona PO (2015) Action recognition from depth maps using deep convolutional neural networks. IEEE Trans Hum-Mach Syst 46:498–509. https://doi.org/10.1109/THMS.2015.2504550

Weinzaepfel P, Harchaoui Z, Schmid C (2015) Learning to track for spatio-temporal action localization. In: Proceedings of the IEEE international conference on computer vision, pp 3164–3172. https://doi.org/10.1109/ICCV.2015.362

Yang X, Tian Y (2012) Eigenjoints-based action recognition using naïve-bayes-nearest-neighbor. In: IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW), pp 14–19. https://doi.org/10.1109/CVPRW.2012.6239232

Yao H, Jiang X, Sun T, Wang S (2017) 3d human action recognition based on the spatial-temporal moving skeleton descriptor. In: IEEE international conference on multimedia and expo (ICME), pp 937–942. https://doi.org/10.1109/ICME.2017.8019498

Zacks JM, Kumar S, Abrams RA, Mehta R (2009) Using movement and intentions to understand human activity. Cognition 112:201–216. https://doi.org/10.1016/j.cognition.2009.03.007

Zanfir M, Leordeanu M, Sminchisescu C (2013) The moving pose: an efficient 3d kinematics descriptor for low-latency action recognition and detection. In: IEEE international conference on computer vision (ICCV). https://doi.org/10.1109/ICCV.2013.342

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.