IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN'16), pp. 71-76, New York, US.

Human Motion Assessment in Real Time using Recurrent Self-Organization

German I. Parisi, Sven Magg, and Stefan Wermter

***Abstract*— The correct execution of well-defined movements plays a crucial role in physical rehabilitation and sports. While there is an extensive number of well-established approaches for human action recognition, the task of assessing the quality of actions and providing feedback for correcting inaccurate movements has remained an open issue in the literature. We present a learning-based method for efficiently providing feedback on a set of training movements captured by a depth sensor. We propose a novel recursive neural network that uses growing self-organization for the efficient learning of body motion sequences. The quality of actions is then computed in terms of how much a performed movement matches the correct continuation of a learned sequence. The proposed system provides visual assistance to the person performing an exercise by displaying real-time feedback, thus enabling the user to correct inaccurate postures and motion intensity. We evaluate our approach with a data set containing 3 powerlifting exercises performed by 17 athletes. Experimental results show that our novel architecture outperforms our previous approach for the correct prediction of routines and the detection of mistakes both in a single- and multiple-subject scenario.**

I. Introduction

The analysis and assessment of human body motion has recently attracted significant interest in the healthcare community with many application areas such as physical rehabilitation, diagnosis of pathologies, and assessment of sport performance. In this context, the correctness of postural transitions is paramount during the execution of well-defined physical routines, since inaccurate movements may significantly reduce the overall efficiency of the movement and increase the risk of injury [1]. For instance, in the case of weight-lifting training, correct postures improve the mechanical efficiency of the body and allow the athlete to achieve higher effectiveness during training sessions. Similarly, in the healthcare domain, the correct execution of physical rehabilitation routines is crucial for patients to improve their health condition [2].

Human proprioception may not be sufficient to spot movement mistakes. Thus, expert trainers observing the movement can give the trainee proficient feedback for timely improving the quality of the performance and avoiding persistent inaccuracies. However, it is not the case that a personal trainer is always available to assess the quality of movements during their execution. Therefore, there is a strong motivation to develop automatic systems able to detect mistakes during the performance of well-defined routines for providing feedback in real time.

G. I. Parisi, S. Magg, and S. Wermter are with the Knowledge Technology Research Group, Department of Informatics, University of Hamburg, Germany. {parisi,magg,wermter}@informatik.uni-hamburg.de

A large number of learning-based models have been proposed that address the classification of a set of training actions (e.g., [3]). However, while the aim of action recognition is to categorize a set of distinct classes by extrapolating inter-class spatiotemporal differences, action assessment requires instead a model to capture intra-class dissimilarities that allow to express a measurement on how much an action follows its learned template. In this setting, efficient approaches to learn spatiotemporal templates for computing intra-class dissimilarities have remained an open issue. Common computational bottlenecks are the robust extraction of body features from video streams and the definition of suitable metrics aimed to compare two actions in terms of their spatiotemporal structure. The former issue has been partly addressed with the use of depth sensors that allow the efficient tracking of human motion and the estimation of a 3D skeleton model. On the other hand, effective methods for the computation of a similarity measure between two actions still represent a major challenge.

In this work, we propose a novel neural architecture that learns a set of actions from depth map videos. The quality of actions is computed in terms of how much a performed movement matches the correct continuation of a training movement. The goal of the proposed system is to provide visual assistance to the user performing an exercise by displaying real-time feedback, thus enabling the person to correct inaccurate postures and motion intensity. Our learning architecture consists of two hierarchically arranged layers with self-organizing networks that process posture and motion sequences. The first layer comprises two self-organizing networks that learn a dictionary of posture and motion features. The second layer is composed of a novel self-organizing network with recurrent connectivity that receives as input neuron activation patterns from the first layer and learns the spatiotemporal structure of a sequence.

We compared our recursive neural model with previous models of recursive self-organization on a regression task and then evaluated our feedback system on a data set with 17 athletes performing 3 powerlifting exercises. Experimental results show that our novel architecture outperforms our previous approach for the prediction of correct body motion and the detection of mistakes in both single-subject and multiple-subject scenarios.

II. Related Work

A. Human Motion Assessment

Automatic systems for the visual assessment of body motion have been previously investigated for applications

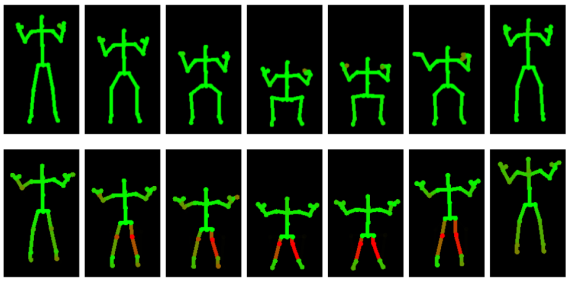

Fig. 1. Visual feedback for correct squat sequence (top), and a sequence containing *knees in* mistake (bottom, joints and limbs in red) [9].

mainly focused on physical rehabilitation and sport training.

Chang *et al.* [4] proposed a physical rehabilitation system for young patients with motor disabilities using a Kinect sensor. The idea was to assist the users while performing a set of simple movements necessary to improve their motor proficiency during the rehabilitation period. Users were instructed by a therapist on how to perform the movements. During the autonomous execution, visual hints were shown to users to motivate the performance of the routines. Although experimental results have shown improved motivation for users using visual hints, only movements involving the arms at constant speed were considered. Furthermore, the estimation of real-time feedback in order to enable the user to spot and correct mistakes was not considered.

Similarly, Su [5] proposed the estimation of feedback for Kinect-based rehabilitation exercises by comparing performed motion with a pre-recorded execution by the same person. The comparison was carried out on sequences using dynamic time warping (DTW) and fuzzy logic with the Euclidean distance as a similarity measure. The evaluation of the exercises was based on the degree of similarity between the current sequence and a correct sequence. The system provided qualitative feedback on the similarity of body joints and execution speed, but it did not suggest the user how to correct the movement.

Paeiment *et al.* [6] proposed a method for assessing the quality of gait from sequences of people on stairs. As a measure of quality, Kinect-based body poses were compared to learned normal occurrences of a movement from a statistical model. The likelihood of a model for describing the current movement was computed frame-by-frame over a sequence of postures and motion speed. The system triggered an alarm if the current movement differed from the correct movement template. For this purpose, a proper threshold must be empirically chosen to decide the degree of tolerance with respect to the template. Although this method represents a useful application for detecting abnormal behavioral patterns, it does not provide any hints on how to correct motion mistakes.

Velloso *et al.* [7] investigated qualitative action recognition with a Kinect sensor for specifying the correct execution of movements, detecting mistakes, and providing feedback to the user. A baseline was created by asking the users to perform a routine ten times, from which individual repetitions were manually segmented. Hidden Markov Models were trained with tuples containing the joint angles and the timestamp for individual exercises. Similar to Chang *et al.* [4] and Su [5], the system was tested only on arm movements, in this case for dumbbell lifting. A strong limitation of this approach is that the correct duration and motion intensity of movements were computed by using the timestamp from body joint estimation. Therefore, although the system provides feedback to correct body posture in terms of joint angles, it does not provide any robust feedback on temporal discrepancies.

For the assessment of human motion in sports, Pirsiavash *et al.* [8] predicted scores of performed movements from annotated footage. The system compared the gradient for each body joint with a regression model from spatiotemporal pose features to scores obtained from expert judges. Feedback is provided in terms of which joints should be changed to obtain the maximum score. Different from the previously discussed approaches, this method extracts body features from RGB sequences. Thus, the estimation of body joints is not as robust as the 3D skeleton model using with a depth sensor. Experimental results showed that the system predicted scores better than non-expert humans but significantly worse than expert judges.

In our previous work, we presented a neural architecture for providing feedback on a set of learned movements captured with a Kinect sensor [9]. The system can predict a set of learned sequences and then provide visual hints to correct posture mistakes (Fig. 1). The architecture comprises a recursive self-organizing network that learns the spatio-temporal structure of input sequences and then estimates feedback as the difference between the current input and the learned template. The system has shown good results on a data set of 3 powerlifting exercises by showing the correct postures and spotting mistakes. However, it did not account for learning the motion intensity, which is crucial for exercises with variations of speed or lockouts, e.g. in weightlifting routines. Furthermore, the recursive model had a limited memory, hindering the learning of longer sequences. In this work, we will show how these two drawbacks can be addressed.

B. Recurrent Self-Organization

Different approaches have been proposed that implement recurrent self-organizing networks for processing sequences. The Temporal Kohonen Map (TKM) [12] is equipped with recurrent neurons in terms of leaky integrators. The computation of the distance of a neuron $\mathbf{w}_i$ from the input sequence $(\mathbf{x}_1, ..., \mathbf{x}_t)$ at time t with similarity measure d_W is

$$\tilde{d}_i(t) = \alpha \cdot d_W(\mathbf{w}_i, \mathbf{x}_t) + (1 - \alpha) \cdot \tilde{d}_i(t-1), \qquad (1)$$

where $\alpha \in (0; 1)$ controls the rate of signal decay, expressing the quality of the representation of the current input and the exponentially weighted past. However, in the TKM there is no explicit back-reference to previous map activity, i.e. the context is only implicitly represented by the weights.

Other models use less restricted recurrence. For instance, in the RecSOM [13], the distance of a neuron from the input

sequence at time t is computed as

$$d_i(t) = \alpha \cdot d_W(\mathbf{w}_i, \mathbf{x}_t) + \beta \cdot \|\mathbf{c}_i - R_{t-1}\|, \quad (2)$$

$$R_{t-1} = (exp(-\tilde{d}_1(t-1)), ..., exp(-\tilde{d}_N(t-1))), \quad (3)$$

where $\mathbf{c}_i$ are the context descriptors of each neuron, R_{t-1} is the context vector of the previous time step, N is the number of neurons in the map, and $\|\cdot\|$ denotes the Euclidean distance. This preserves the information available within the activation at the last timestep. However, this is computationally expensive due to the high-dimensional contexts attached to each neuron. A more compact model was introduced by the SOM-SD [14], where an additional context vector is used for each neuron, but only the last winner index is stored as information of the previous map state such that

$$\tilde{d}_i(t) = \alpha \cdot d_W(\mathbf{w}_i, \mathbf{x}_t) + \beta \cdot d_G(\mathbf{I}_{t-1} - \mathbf{c}_i), \quad (4)$$

where $\mathbf{I}_{t-1}$ denotes the index of the winner neuron at $t-1$ and d_G is a the grid distance measure. However, this recurrent activation cannot be used for arbitrary lattice shapes since it relies on fixed grid distances to update the winning neuron and its neighbours.

Different approaches with context learning have been proposed that use compact reference representation for arbitrary lattice topologies. The MergeSOM [15] combines a compact back-reference with a weighted contribution of the current input and the past. Each neuron is equipped with a weight vector $\mathbf{w}_i$ and a temporal context $\mathbf{c}_i$, the latter representing the activation of the entire map in the previous timestep. The recursive activation function of a sequence is given by the linear combination

$$\tilde{d}_i(t) = \alpha \cdot d_W(\mathbf{w}_i, \mathbf{x}_t) + (1-\alpha) \cdot d_W(\mathbf{c}_i, \mathbf{C}_i), \quad (5)$$

$$\mathbf{C}_i = \beta \cdot \mathbf{w}_{I(t-1)} + (1-\beta) \cdot \mathbf{c}_{I(t-1)}, \quad (6)$$

where $\alpha, \beta \in (0, 1)$ are fixed parameters, $\mathbf{C}_i$ is a global context vector, and $I(t-1)$ denotes the index of the winner neuron at $t-1$. This model converges to an efficient fractal encoding of sequences with high temporal quantization accuracy. Furthermore, context learning can be applied not only to lattices with arbitrary topology, but also to incremental approaches that vary the number of neurons over time. For instance, a Growing Neural Gas (GNG) model equipped with context learning (MergeGNG, [17]) uses the activation function defined by Eq. 5 and 6 to compute winner neurons and creates new neurons with a temporal context. A general approach for updating the weight and context vectors is using the competitive Hebbian learning rule [10]. In the next section, we describe how to implement context learning with a special type of growing self-organizing network.

III. Proposed Method

Our architecture consists of two hierarchically arranged layers with self-organizing networks processing posture and motion sequences (Fig. 2). The first layer is composed of two Growing When Required (GWR) networks, G^P and G^M, that learn a dictionary of posture and motion feature vectors respectively. The second layer comprises a recursive GWR, G^I, that learns neuron activation patterns from G^P and G^M.

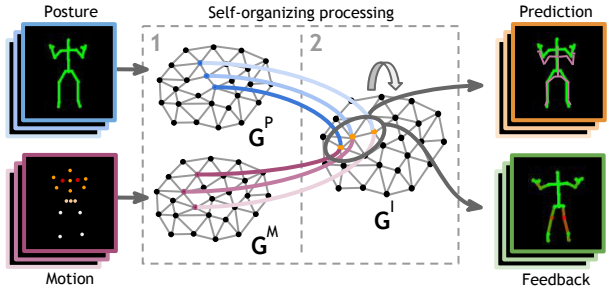

Fig. 2. Multilayer learning architecture with incremental self-organizing networks. In Layer 1, two GWR networks learn posture and motion features respectively. In Layer 2, a recursive GWR learns spatiotemporal dynamics of body motion. This mechanism allows to predict the ideal continuation of a learned sequence and compute feedback as the difference between its expected behavior and its current execution.

A. Posture-Motion Sequences

The Kinect's skeleton model (Fig. 1), although not faithful to human anatomy, provides reliable estimations of the joints' position over time. This allows us to extract significant properties of postural dynamics. For our approach, we tracked the position of a person based on a simplified 3D model of the human skeleton using a set of K joint coordinates $\mathbf{j}_i = (x_{j_i}, y_{j_i}, z_{j_i}), 1 \leq i \leq K$, so that at each timestep t the body posture is represented as the collection of K joints $\mathbf{p}(t) = (\mathbf{j}_i(t), ..., \mathbf{j}_K(t))$. We computed motion intensity from posture sequences with the inter-frame difference between consecutive joint pairs.

For our experiments, we used body motion with a Kinect v2 sensor[1] and estimated body joints using Kinect SDK 2.0 that provides a set of 25 joint coordinates at 30 frames per second. We used the joints for *head, neck, wrists, elbows, shoulders, spine, hips, knees*, and *ankles*, for a total of 13 3D-joints (39 dimensions). In order to obtain translation-invariance, we subtracted the *spine_base* joint (the center of the hips) from all absolute joint coordinates.

B. Hierarchical, Growing Self-Organizing Processing

The GWR [11] is a growing self-organizing neural network that learns prototype representations of the inputs while preserving their topological properties. The network is composed of a set of neurons and their associated weight vectors $\mathbf{w}_j$ linked by a set of edges. During the training, the network starts with two neurons and then dynamically changes its topological structure to better match the input space using competitive Hebbian learning [10].

At each iteration, the network computes the activation as the difference between the current input and its best-matching prototype neuron. Additionally, a firing counter is used to compute how much neurons have fired. This is in favour of training existing neurons over creating new ones, and then adding new neurons whenever the network activity with respect to the input is smaller than a given threshold. This mechanism allows the network to adapt to

[1]Microsoft Kinect 2.0 – microsoft.com/en-us/kinectforwindows/develop/

changing input distributions faster than other models of growing self-organization, e.g. with respect to the Growing Neural Gas [16] where new neurons are added at fixed intervals. The learning algorithm will iterate over the training set until a given stop criterion is met, e.g. a max number of training epochs.

In the first layer of our architecture (Fig. 2), the GWR networks G^P and G^M learn respectively a set of posture and motion prototype vectors used to efficiently represent the temporal structure of a sequence in the next layer. This hierarchical scheme has the advantage of using a fixed set of learned features to compose more complex patterns in the second layer, where the recursive network G^I is trained with sequences of posture-motion activation patterns from the first layer to learn the spatiotemporal structure of the input. From a dataset $\mathbf{X}$ with n samples, we compute the best-matching neuron of the input sequence with respect to the trained network with N neurons, so that a sequence of input activations from the training set is given by

$$\Omega(\mathbf{X}) = \{\mathbf{w}_{b(\mathbf{x}_1)}, \mathbf{w}_{b(\mathbf{x}_2)}, ..., \mathbf{w}_{b(\mathbf{x}_n)}\}, \tag{7}$$

with $b(\mathbf{x}_i) = \arg\min_{i \in N} \|\mathbf{x}_i - \mathbf{w}_j\|$ computing the index of the neuron (or prototype vector) that minimizes the distance to the current input. We denote the dataset of posture and motion vectors as $\mathbf{P}$ and $\mathbf{M}$ respectively. The training dataset for G^I, $\mathbf{I}$, is given by the horizontal concatenation of the set of activations over $\mathbf{P}$ and $\mathbf{M}$, i.e. $\mathbf{I} = \{\Omega(\mathbf{P}) ^\frown \Omega(\mathbf{M})\}$.

C. Recursive GWR

To learn the spatiotemporal structure of the input in G^I, we extend the traditional GWR algorithm [11] for efficient context learning [15]. We adopt the distance function

$$d_n(t) = \alpha \cdot \|\mathbf{x}_t - \mathbf{w}_n\|^2 + (1-\alpha) \cdot \|\mathbf{C}_t - \mathbf{c}_i\|^2, \tag{8}$$

$$\mathbf{C}_t = \beta \cdot \mathbf{w}_{b_{t-1}} + (1-\beta) \cdot \mathbf{c}_{b_{t-1}}, \tag{9}$$

where α and β are constant values that modulate the influence of the current input and the past. Specifically for our recursive GWR model, the update functions of the weight and context neurons become

$$\Delta\mathbf{w}_i = \epsilon_i \cdot \eta(i) \cdot (\mathbf{x}_t - \mathbf{w}_i), \tag{10}$$

$$\Delta\mathbf{c}_i = \epsilon_i \cdot \eta(i) \cdot (\mathbf{C}_t - \mathbf{c}_i), \tag{11}$$

where ϵ_i is the learning rate and $\eta(i)$ is the firing counter.

The complete training algorithm is as follows:

1) Start with a set of two random neurons $A = \{\mathbf{w}_1, \mathbf{w}_2\}$ with context vectors $\mathbf{c}_1$, $\mathbf{c}_2$
2) Initialize an empty set of connections $E = \emptyset$
3) Initialize the empty global context $\mathbf{C}_1 = 0$
4) At each iteration, generate an input sample $\mathbf{x}_t$
5) Select best and second-best matching neurons (Eq. 8): $b = \arg\min_{n \in A} d_n(t)$ and $s = \arg\min_{n \in A/\{b\}} d_n(t)$
6) Create a connection $E = E \cup \{(b, s)\}$ if it does not exist and set its age to 0
7) If $(exp(-\|\mathbf{x}_t - \mathbf{w}_b\|^2) < a_T)$ and $(\eta(b) < f_T)$ then:
   Add a new neuron r ($A = A \cup \{\mathbf{w}_r\}$):
   $\mathbf{w}_r = 0.5 \cdot (\mathbf{w}_b + \mathbf{x}_t)$, $\mathbf{c}_r = 0.5 \cdot (\mathbf{C}_t + \mathbf{x}_t)$
   Update edges between neurons:
   $E = E \cup \{(r, b), (r, s)\}$, $E = E/\{(b, s)\}$
   Increase by 1 the age of all the other edges
8) Update weight and context vectors of best-matching neuron b and its neighbours:
   $\Delta\mathbf{w}_b = \epsilon_b \cdot \eta(b) \cdot (\mathbf{x}_t - \mathbf{w}_b)$, $\Delta\mathbf{w}_i = \epsilon_n \cdot \eta(i) \cdot (\mathbf{x}_t - \mathbf{w}_i)$,
   $\Delta\mathbf{c}_b = \epsilon_b \cdot \eta(b) \cdot (\mathbf{C}_t - \mathbf{c}_b)$, $\Delta\mathbf{c}_i = \epsilon_n \cdot \eta(i) \cdot (\mathbf{C}_t - \mathbf{c}_i)$
9) Update global context $\mathbf{C}_t$ for next timestep (Eq. 9)
10) Reduce the firing counters of the best-matching neuron and its neighbours i:
   $\eta(b) = \eta(b) + (\tau_b \cdot \kappa \cdot (1 - \eta(b)) - \tau_b)$,
   $\eta(i) = \eta(i) + (\tau_i \cdot \kappa \cdot (1 - \eta(i)) - \tau_b)$,
   with τ, κ constants controlling the curve behaviour.
11) Remove all edges with age larger than a_{max} and remove neurons without edges
12) If the stop criterion is not met, go to Step 4

The recursive GWR architecture avoids the drawback of our previous approach using a MSOM, where the number of neurons of the networks had to be decided a priori. Furthermore, since the GWR does not have a fixed lattice topology, it can better represent the feature space.

D. Feedback from Prediction

The underlying idea for assessing the quality of a sequence is to measure how much the current input sequence differs from a learned template. In other words, provided that the trained model G^I is able to predict a training sequence with a satisfactory degree of accuracy, it is then possible to quantitatively compute how much a novel sequence follows this expected pattern.

We define a function that computes the difference of a current input sequence, Ω_t , from its expected input, i.e. the prediction of the next element of the sequence given Ω_{t-1}:

$$\mathfrak{f}_\Omega(t) = \|\Omega_t - \mathfrak{p}(\Omega_{t-1})\|, \tag{12}$$

$$\mathfrak{p}(\Omega_{t-1}) = \mathbf{w}_p \text{ with } p = \arg\min_{j \in N} \|\mathbf{c}_j - \Omega_{t-1}\|. \tag{13}$$

Since the weight and context vectors of the prototype neurons lie in the same feature space as the input ($\mathbf{w}_i, \mathbf{c}_i \in \mathbb{R}^{|\Omega|}$), it is possible to provide joint-wise feedback computations. The recursive prediction function $\mathfrak{p}$ can be applied an arbitrary number of timesteps into the future. Therefore, after the training phase is completed, it is possible to compute $\mathfrak{f}_\Omega(t)$ in real time with linear computational complexity $\mathcal{O}(|A|)$, which depends on the number of neurons of a trained model.

To compute feedback, we use the predictions estimated by $\mathfrak{p}$ as hints on how to perform a routine over 100 timesteps into the future, and then use $\mathfrak{f}_\Omega(t)$ to spot mistakes on novel sequences that do not follow the expected pattern for individual joint pairs. A mistake can then be detected when $\mathfrak{f}_\Omega(t)$ exceeds a given threshold $\mathfrak{f}^T$ over i timesteps. Visual representations of these computations can then provide useful qualitative feedback to assist the user on the correct performance of the routine and the correction of mistakes (Fig. 1). Different from our previous model, our current approach learns also motion intensity to better detect

temporal discrepancies. Therefore, it is possible to provide more accurate feedback on posture transitions and the correct execution of lockouts.

IV. Experimental Results

A. Time Series Analysis

We compared the performance of our MGWR on a time series analysis task with other two well-established models of recursive self-organization: Merge Neural Gas (MNG) [15] and Merge Growing Neural Gas (MGNG) [17]. For the analysis we used the Mackey Glass time series, a continuous and chaotic function that has been used to evaluate the temporal quantization of recursive models. It is defined by the differential equation $\frac{dx}{d\tau} = bx(\tau) + \frac{ax(\tau-d)}{1+x(\tau-d)^{10}}$ and depending on the values of the parameters, it displays a range of pseudo-periodic dynamics. For evaluation purposes, it is generally used with $a = 0.2$, $b = -0.1$, and $d = 17$. Similar to previous comparison schemes in the literature [13], all the models were evaluated by their temporal quantization error (TQE) for 30 steps in the past with 150,000 elements of the series. The TQE for the map at time t is defined as:

$$e(t) = \sum_{i=1}^{N} \left(\sum_{j:I(j)=i} \|\mathbf{x}^{j-t} - \sum_{j:I(j)=i} \mathbf{x}^{j-t}/\gamma_i\|^2/\gamma_i \right)^{1/2} /N, \quad (14)$$

where N is the number of neurons, γ_i is the number of timesteps in which neuron i becomes the winner.

For MGWR learning, we used the following training parameters: insertion threshold $a_T = 0.95$, learning rates $\epsilon_b = 0.01$, and $\epsilon_n = 0.001$, maximum age $a_{max} = 200$, firing counter parameters $\tau_b = 0.3$, $\tau_i = 0.1$, $\kappa = 1.05$, firing threshold $\eta_T = 0.1$, and context learning parameters $\alpha = 0.6$, $\beta = 0.7$ with 100 training epochs. The training parameters of MNG and MGNG were set according to previously reported experiments [15], [17].

The TQE for the recursive models MSOM, MNG, MGNG and our MGWR is reported in Fig. 3, showing how the four models behave quite similar, with the MGWR slightly outperforming the others. The average TQE over 30 timesteps was MSOM= 0.0795, MNG= 0.0749, MGNG= 0.0721, and MGWR= 0.0697. Although both the MSOM and MNG are not growing methods, the latter performs better since the topology of the MNG network is not fixed, thus yielding a smaller quantization error.

B. Feedback on Powerlifting Routines

We used the dataset of 3 powerlifting exercises performed by 17 volunteering athletes (9 male, 8 female) collected at the Kinesiology Institute of the University of Hamburg [9]:

- E1) *High bar back squat*: One repetition consists of crouching with a loaded barbell behind the back until the hips are lower than the knees and then standing up;
- E2) *Deadlift*: Lift a loaded barbell off the ground to the hips, then lower back to the ground;

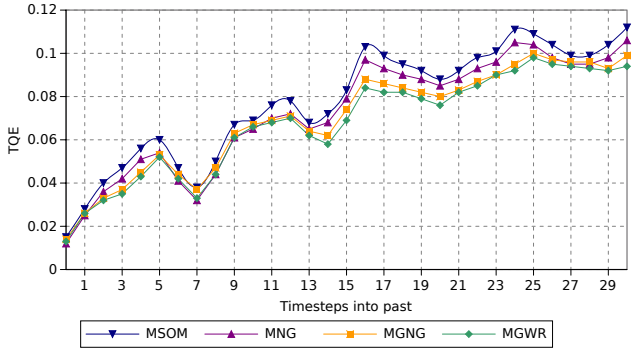

Fig. 3. Temporal quantization error over 30 timesteps into past for the Mackey-Glass time series.

- E3) *Dumbbell lateral raise*: Start with the arms at side of the body, then raise the dumbbells sidewards while keeping the elbows higher than the wrists.

To evaluate the system for the computation of feedback, we also used a set of typical mistakes for each routine:

- E1) *Good morning* (Horizontal back angle), *Half squat*, *Knees in*;
- E2) *No lockout*, *Rounded back*;
- E3) *Low elbows*.

We evaluated our method for computing feedback with individual and multiple subjects. We divided the correct body motion data with 3-fold cross-validation into training and test sets and trained the models with data containing correct motion sequences. Each network was trained for 100 epochs. For the test phase, both the correct and incorrect movements were used with feedback threshold $\mathfrak{f}^T = 0.7$ over 100 frames.

Our expectation was that the output of the feedback function would be higher for sequences containing mistakes. We observed true positives (TP), false negatives (FN), true negatives (TN), and false positives (FP) as well as the measures true positive rate (TPR or sensitivity), true negative rate (TPR or specificity), and positive predictive value (PPV or precision). Results for single- and multiple-subject data on E1, E2, and E3 routines are displayed in Table I and II respectively, along with a comparison with the best-performing feedback function $\mathfrak{f}_b$ from our previous approach [9] that predicted the next input from a graph containing the successor distances for all neurons.

The evaluation on single subjects shows that the system successfully provides feedback on posture errors with high accuracy. A drawback of our previous model was a limited memory due to the number of neurons being fixed a priori and a fixed network topology yielding a higher quantization error. In our current approach, the MGWR networks grow dynamically to better represent the spatiotemporal structure of the sequences. This allows us to reduce the temporal quantization error over longer timesteps (Fig. 3), so that more accurate feedback can be computed and thus reduce the number of false negatives and false positives (Table I and II). Furthermore, since the networks can create new neurons according to the distribution of the input, each network can

TABLE I

SINGLE-SUBJECT EVALUATION.

		TP	FN	TN	FP	TPR	TNR	PPV
E1	$\mathfrak{f}_\mathfrak{b}$	35	10	33	0	0.77	1	1
	$\mathfrak{f}_\Omega$	35	2	41	0	0.97	1	1
E2	$\mathfrak{f}_\mathfrak{b}$	24	0	20	0	1	1	1
	$\mathfrak{f}_\Omega$	24	0	20	0	1	1	1
E3	$\mathfrak{f}_\mathfrak{b}$	63	0	26	0	1	1	1
	$\mathfrak{f}_\Omega$	63	0	26	0	1	1	1

TABLE II

MULTI-SUBJECT EVALUATION.

		TP	FN	TN	FP	TPR	TNR	PPV
E1	$\mathfrak{f}_\mathfrak{b}$	326	1	7	151	0.99	0.04	0.68
	$\mathfrak{f}_\Omega$	328	1	13	143	0.99	**0.08**	**0.70**
E2	$\mathfrak{f}_\mathfrak{b}$	127	2	0	121	0.98	0	0.51
	$\mathfrak{f}_\Omega$	139	0	0	111	**1**	0	**0.56**
E3	$\mathfrak{f}_\mathfrak{b}$	123	0	8	41	1	0.16	0.75
	$\mathfrak{f}_\Omega$	126	0	15	31	1	**0.33**	**0.80**

learn a larger number of possible executions of the same routine, thus being more suitable for training sessions with multiple subjects.

Tests with multiple-subject data shows a significantly decreased performance, mostly due to a large number of false positives. This is not necessarily a flaw linked to the learning mechanism, but rather a consequence of the fact that people have different body configurations and, therefore, different ways to perform the same routine. A solution to attenuate this issue is to set different values for the feedback threshold $\mathfrak{f}^T$. For larger values, the system would tolerate more variance in the performance. On the other hand, one must consider whether a higher degree of variance is desirable based on the application domain; for instance, rehabilitation routines may be tailored to a specific subject based on their specific body configuration and health condition.

V. CONCLUSION

We presented a learning-based method that provides visual assistance to the person performing an exercise by displaying real-time feedback, thus enabling the user to correct inaccurate body motion. The quality of actions is computed in terms of how much a performed movement matches the correct continuation of a learned sequence template. The main contribution of our work is a novel recursive neural network, the MGWR, that uses growing self-organization for the efficient learning of input sequences. With respect to our previous model [9], the current approach accounts also for learning motion intensity to better predict and assess the dynamics of actions. We evaluated our system with a data set with 3 powerlifting exercises, showing that we outperform our previous approach for the detection of mistakes, in particular for the multiple-subject scenario.

Our experimental results encourage further work in the direction of embedding our system in an assistive robot companion which could interact with the user and motivate the correct performance of physical rehabilitation routines and sports training. This is supported by a number of studies in which robots were used for motivating the users to perform a set of health-related tasks [18], [19], [20]. Furthermore, the assessment of motion plays a crucial rule not only for the detection of mistakes on training sequences, but also in the timely recognition of gait deterioration, e.g. linked to age-related cognitive declines. In this context, growing learning architectures are particularly suitable for this task, since they may adapt to the user through longer periods of time while still detecting significant changes in their motor skills.

Acknowledgements

This research was partially supported by the DAAD German Academic Exchange Service (Kz:A/13/94748) and the DFG (Deutsche Forschungsgemeinschaft) for the project Cross-modal Learning TRR-169 / A5. The authors would like to thank Florian von Stosch for the dataset collection.

REFERENCES

[1] R. Kachouie, S. Sedighadeli, R. Khosla, and M-T. Chu, M-T. Socially Assistive Robots in Elderly Care: A Mixed-Method Systematic Literature Review. Int. J. Hum. Comput. Interaction 30(5):369–393, 2014.

[2] E. Velloso, A. Bulling, G. Gellersen, W. Ugulino, and G. Fuks. Qualitative activity recognition of weight lifting exercises. Augmented Human International Conference, 116–123 (ACM), 2013.

[3] G.I. Parisi, C. Weber, and S. Wermter. Self-Organizing Neural Integration of Pose-Motion Features for Human Action Recognition. Frontiers in Neurorobotics, 9(3), 10.3389/fnbot.2015.00003, 2015.

[4] Y-J. Chang, S-F. Chen, J-D. Huang. A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. Research in Developmental Disabilities 32(6):2566–2570, 2011.

[5] C.J. Su. Personal rehabilitation exercise assistant with kinect and dynamic time warping. Intl. Journal of Information and Education Technology, 3(4):448–454, 2013.

[6] A. Paiement, L. Tao, S. Hannuna, C. Camplani, M. Damen, and M. Mirmehdi. Online quality assessment of human movement from skeleton data. In: British Machine Vision Conference (BMVC), 2014.

[7] E. Velloso, A. Bulling, and H. Gellersen. MotionMA: Motion modelling and analysis by demonstration. In: SIGCHI Conference on Human Factors in Computing Systems, 1309–1318 (ACM), 2013.

[8] H. Pirsiavash, C. Vondrick, and A. Torralba. Assessing the quality of actions. In: ECCV, 556–571, 2014.

[9] G.I. Parisi, F. v. Stosch, S. Magg, S. Wermter. Learning human motion feedback with neural self-organization. In: IJCNN, pp. 2973-2978, 2015.

[10] T. Martinetz. Competitive Hebbian learning rule forms perfectly topology preserving maps. In: ICANN, pp. 427–434, Springer, 1993.

[11] S. Marsland, J. Shapiro, and U. Nehmzow. A self-organising network that grows when required. Neural Networks 15:1041–1058, 2002.

[12] G.J. Chappell and J.G. Taylor. The temporal Kohonen map. Neural networks 6:441–445, Elsevier, 1993.

[13] T. Voegtlin. Recursive self-organizing maps. Neural Networks 15(8):979-991, 2002.

[14] M. Hagenbuchner, A. Sperduti, and A.C. Tsoi. A self-organizing map for adaptive processing of structured data. Neural Networks, 14(3):491–505, 2003.

[15] M. Strickert, B. Hammer. Merge SOM for temporal data. Neurocomputing 64:39–71, 2005.

[16] N. Fritzke. A Growing Neural Gas Network Learns Topologies. In: NIPS 7, pp. 625–632, MIT Press, 1995.

[17] A. Andreakis, N.v. Hoyningen-Huene, M. Beetz M. Incremental unsupervised time series analysis using merge growing neural gas. In: WSOM, pp. 10–18 (Springer), 2009.

[18] K. Dautenhahn. Robots as social actors: Aurora and the case of autism. In: Third Cognitive Technology Conference, 1999.

[19] C. D. Kidd and C. Breazeal. A robotic weight loss coach. In: AAAI, pp. 1985–1986, AAAI Press, 2007.

[20] M. Nalin, I. Baroni, A. Sanna, and C. Pozzi. Robotic companion for diabetic children: emotional and educational support to diabetic children, through an interactive robot. In: ACM SIGCHI, pp. 260–263, 2012.