In: IEEE International Joint Conference on Neural Networks (IJCNN), pp. 2973-2978, Killarney, Ireland (2015)

Learning Human Motion Feedback with Neural Self-Organization

German I. Parisi, Florian von Stosch, Sven Magg and Stefan Wermter
Knowledge Technology Group, Department of Informatics
University of Hamburg, Germany
{parisi,8stosch,magg,wermter}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/WTM/

***Abstract*—The correct execution of well-defined movements in sport disciplines may increase the body's mechanical efficiency and reduce the risk of injury. While there exists an extensive number of learning-based approaches for the recognition of human actions, the task of computing and providing feedback for correcting inaccurate movements has received significantly less attention in the literature. We present a learning system for automatically providing feedback on a set of learned movements captured with a depth sensor. The proposed system provides visual assistance to the person performing an exercise by displaying real-time feedback to correct possible inaccurate postures and motion. The learning architecture uses recursive neural network self-organization extended for predicting the correct continuation of the training movements. We introduce three mechanisms for computing feedback on the correctness of overall movement and individual body joints. For evaluation purposes, we collected a data set with 17 athletes performing 3 powerlifting exercises. Our results show promising system performance for the detection of mistakes in movements on this data set.**

I. Introduction

Many physical activities such as dancing, yoga, and strength training are composed of a collection of well-defined movements. In this context, the correctness of postural transitions is paramount for the execution of a specific exercise since inaccuracies in posture or motion may significantly reduce the overall efficiency of the movement and increase the risk of injury [1]. For instance, in the case of strength training, correct postures improve the mechanical efficiency, thereby allowing for more weight to be lifted and therefore yielding higher effectiveness during training sessions [2]. The correct execution of a movement is crucial also for competitions, where there is a set of strict rules deciding whether a weight lifting routine is successfully executed by the athlete.

During the execution of complex movements, human proprioception may not be sufficient to be aware of postural mistakes for a timely correction. On the other hand, these mistakes may be well noticeable by an expert trainer observing the movement, thus enabling the trainee to use external feedback for correcting mistakes and avoiding deterioration [3]. However, it is not always the case that a personal trainer can be available for taking care of the execution of movements during training. Therefore, there is the motivation to provide automatic motion feedback systems able to detect mistakes during the performance of the movements and enabling the person to correct them autonomously.

In the last half decade, the use of low-cost depth sensing devices such as Microsoft Kinect has fostered the development of visual-based applications for human gait analysis and action recognition. In contrast to traditional 2D cameras, the Kinect sensor provides depth measurements used to segment human 3D motion in cluttered environments, including a set of body joints that enable us to estimate spatio-temporal properties of actions in real time [17]. The combination of Kinect-based 3D skeleton information with learning paradigms such as machine learning and neural networks has led to an extensive number of approaches for the robust detection and classification of actions (e.g. [5], [6]). In particular, these methods allow to learn a set of posture-motion properties from articulated actions to distinguish distinct action classes. However, the additional task of providing a measure on how these actions are performed and how to correct mistakes has not yet been explored likewise in the literature. Different from the classification of distinct actions, the latter task involves finding subtle differences in the execution of correctly learned actions and then triggering a mechanism to enable the person for timely correcting movement mistakes.

In this work, we describe a learning system for providing feedback on a set of learned movements captured with a Kinect device. Our system visually assists the person by showing predictions of correct postural transitions and, if mistakes are detected, it provides visual feedback with the needed corrections for an accurate execution. The learning architecture is built upon a recursive variant of neural network self-organization that is trained with a set of correct executions. During the training phase, each network adapts its topological structure to generate a spatio-temporal map of prototype movements. From this representation, we compute the overall error and individual joint errors that do not conform to the learned movement model during a subsequent execution of the movement.

This paper is structured as follows. After giving an overview of related work on the estimation of automatic feedback in Section II, we introduce our model of body motion built upon 3D posture representations in Section III. In Section IV, we describe our learning architecture based on recursive self-organization and introduce a prediction mechanism for the correct continuation of learned movements. In Section V, we define three methods for computing feedback on overall movement correctness and individual body joints. In Section VI, we report a number of experiments along with an evaluation of our learning algorithm. For this purpose, we collected a data set with 17 athletes each performing 3 powerlifting exercises. We conclude with a discussion in Section VII.

II. Related Work

Several different approaches have been proposed by previous work to automatically provide movement feedback. For instance, Alexiadis et al. [4] and Anderson et al. [7] provided feedback on dancing and ballet movements respectively. For a specific frame, they computed a feedback score using the Euclidean distance between raw-measured joint locations of an expert and a trainee. However, since trainee postures were compared only directly with expert postures at specific points in time, it severely punished slight timing variations. This made the approach only applicable to movements that require strict timing, such as dancing to rhythm of music. Furthermore, this direct comparison of joints was only feasible for body types with similar properties to that of the expert. Rector et al. [8] provided feedback on yoga poses by comparing measured angles of body parts to valid ranges from domain-specific rules. The system required users to stand in a predefined position on a custom mat with markers. Similarly, Velloso et al. [9] provided feedback for dumbbell strength training exercises. However, approaches of this kind with hard-coded rules are unable to cope with complex movements and different body configurations.

Su [10] generated feedback for home-based rehabilitation exercises by comparing performed motion with a pre-recorded execution by the same person. The comparison was carried out through dynamic time warping (DTW) and fuzzy logic. The sequence of captured joint data was used to compare against the sequence stored in a repository. The evaluation of the exercises result was derived based on the degree of similarity between the two sequences. The system provided qualitative feedback on the similarity of joint trajectories and execution speed, while it did not suggest the patient how to correct the movement. Velloso et al. [11] presented a system which inferred a motion model from correct demonstrations captured with a depth sensor. Feedback was given in two forms: directions for limbs that were supposed to remain stationary and conformance to allowed movement ranges for moving joints. However, the presented system used an independence assumption between individual body joints, so that whole-body motion correctness may not be adequately modelled.

III. Body Motion Representation

A human body can be modelled as a spatially extended object with a set of body joints connected by limbs. We track the position of a person based on a simplified 3D model of the human skeleton. In this setting, the body is modelled as a set of N joint coordinates $\mathbf{q}_i = (x_i, y_i, z_i) \in \mathbb{R}^3, 1 \leq i \leq N$, so that at each time step t a particular body posture is represented as the collection of N joints:

$$\mathbf{p}(t) = (\mathbf{q}_i(t), ..., \mathbf{q}_N(t)). \quad (1)$$

For our experiments, we capture body motion with a Kinect v2 sensor[1] and estimate body joints using Kinect SDK 2.0 that provides a set of 25 joint coordinates at 30 frames per second. The set of joints includes the position for head, neck, wrists, elbows, shoulders, spine, hips, knees, and ankles. As shown in Fig. 1, the Kinect's skeleton model, although not strictly faithful to human anatomy, provides reliable estimations of the joints' position and orientation over time, thereby allowing to extrapolate significant properties of postural dynamics. With this setting, two human postures are perceived as similar when the set of all joint coordinates roughly coincides. This makes distance measures over the set of all joints a suitable measurement for estimating the similarity between two postures [4], [7]. In our implementation, we use the Euclidean distance. In order to yield translation-invariance, we subtract from all absolute joint coordinates the *spine_base* joint, which represents the center of the hips.

Fig. 1. Example of skeleton model with body joints and limbs for a correct squat movement.

On top of this model, we must define a measurement on the quality of the performed movements and the required feedback to correct mistakes. For this purpose, we compute at each time step t an overall error $e(t)$ that estimates the deviation from correct postures, and individual joint errors $e_i(t)$ for providing feedback on specific joint corrections during the execution. Since we are considering postures in the spatio-temporal domain, we must take into account the fact that subtle variations in speed may not affect the correctness of the movement. For instance, a barbell squat in powerlifting might be carried out at different speeds and duration while still being considered a correct squat [2]. Similarly, people vary in shape and size, leading to different body compositions and thereby postural variations during the execution of similar movements [3]. Therefore, for both timing and posture, the system should be tolerant to variances during the execution of specific movements and under those circumstances for which this tolerance is desirable. In this regard, we take into account a learning mechanism that enables the generalization of correct movements for yielding tolerance to variations when detecting relevant deviations from a learned postural pattern.

IV. Learning Architecture

As a learning technique we use a recursive extension of the Self-Organizing Map (SOM) [16], which is composed of a set of neurons arranged on a 2-dimensional network lattice. The SOM performs a non-linear approximation of the distribution of the input through a statistical learning algorithm, so that the input space of activation patterns is mapped onto the space of prototype neurons by a process of competition among the weight vectors associated to the neurons. Each prototype neuron is defined by two coordinate vectors: a lattice position and a value in the input data space, which is adjusted through training. During this learning process, the topology of the input space is preserved. While the traditional SOM is particularly suitable for processing data in the spatial domain, it is possible

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

to extend such a method to allow the processing of sequential patterns.

A. Recursive Self-Organization

Recursive alternatives of the SOM have been proposed that extend the feed-forward learning mechanism with context neurons that refer to past activations (e.g. [12], [13]), so that sequences are encoded by a recursive self-superposition of the trained neurons and the current input. The MergeSOM [14] has shown to reach a stable fixpoint during the learning phase and exhibits greater computational efficiency with respect to similar variants [15]. MergeSOM learning is carried out in two steps iterated over each training sample $\mathbf{x}$. In the first step, an inner activation for each neuron i is computed based on the difference of its weight vector $\mathbf{w}^i$ to the input $\mathbf{x}$ and of its context vector $\mathbf{c}^i$ to the recursive activation $\mathbf{y}$. Its recursive activation function is defined as:

$$d_i = (1-\alpha) \cdot \|\mathbf{x} - \mathbf{w}^i\|^2 + \alpha \cdot \|\mathbf{y} - \mathbf{c}^i\|^2, \quad (2)$$

where α is a fixed coefficient that balances the contribution of the descriptors, and $\mathbf{y}$ is the context descriptor defined as a linear combination of the previous time step's activated neuron u defined as:

$$\mathbf{y} = (1-\beta) \cdot \mathbf{w}^u + \beta \cdot \mathbf{c}^u, \quad (3)$$

where β is the merging coefficient and u is the neuron with the smallest Euclidean distance $u = \text{argmin}_i d_i$, henceforth referred to as the best-matching unit (BMU).

In the second step, the training is carried out by moving the weight and context vector towards the current input $\mathbf{x}$ and the context $\mathbf{y}$ according to:

$$\Delta\mathbf{w}^i = \epsilon \cdot h(u,i) \cdot (\mathbf{x} - \mathbf{w}^i), \quad (4)$$

$$\Delta\mathbf{c}^i = \epsilon \cdot h(u,i) \cdot (\mathbf{y} - \mathbf{c}^i), \quad (5)$$

with ϵ being a decreasing learning rate and $h(u,i)$ being a Gaussian neighbourhood function over the distance of two neurons u and i:

$$h(u,i) = exp\left(-\frac{\|\mathbf{r}^u - \mathbf{r}^i\|^2}{2\sigma^2}\right), \quad (6)$$

where $\mathbf{r}^u$ and $\mathbf{r}^i$ are the respective positions of the neurons in the network lattice and σ is a decaying function governing the neighbourhood radius.

B. Prediction

The MergeSOM has been shown to learn a model with the representation capacity of a finite automaton [14]. We can exploit this mechanism for predicting the ideal sequence of a movement by the means of backtracking prototype neurons over a trained network. The best-matching predecessor of a neuron u can be computed by comparing its expected context descriptor to other neurons' merge vectors, i.e. vectors obtained as the linear combination of weight and context descriptors of previously activated neurons $merge(u) = \mathbf{y}$ (Eq. 3). The predecessor distance is then given by comparing a node's expected context to the merge vector:

Fig. 2. Movement prediction – Visual hints for future steps of a network trained for "Finger to nose" routine. Progressively fading violet lines represent correct execution order.

$$d^p(u,v) = \|merge(v) - \mathbf{c}^u\|, \quad (7)$$

so that the best-matching predecessor of u is given by:

$$\mathfrak{p}(u) = argmin_{v \in V}(d^p(u,v)). \quad (8)$$

For prediction purposes, however, we are not interested in computing the predecessor of a neuron, but rather in the computation of successor chains of neurons referring to future time steps. The best-matching successor of a neuron u can then be defined similarly to the predecessor (Eq. 11):

$$\mathfrak{s}(u) = argmin_{v \in V}(d^s(u,v)), \quad (9)$$

with

$$d^s(u,v) = \|merge(u) - \mathbf{c}^v\|. \quad (10)$$

With this setting, prediction for multiple time steps can then be accomplished by the recursive application of the successor function, such that a successor chain with n successors of u can be defined as:

$$\mathfrak{C}_n(u) = (u, \mathfrak{s}(u), \mathfrak{s}(\mathfrak{s}(u)), ..., \mathfrak{s}^n(u)). \quad (11)$$

We show the result of this prediction mechanism in Fig. 2. For this example, we trained a network with the correct execution of the "Finger to nose" routine, which consists of keeping your arm bent at the elbow and then touching your nose with the tip of your finger. When the person starts performing the routine after this training phase, we can see progressively fading violet lines representing the next 30 time steps, thereby providing visual assistance on how to successfully carry out the movement through spatio-temporal hints. The value 30 was empirically determined to provide a substantial reference to future steps while limiting visual clutter.

V. Extracting Feedback From Learned Models

We now define three feedback mechanisms built upon the trained networks.

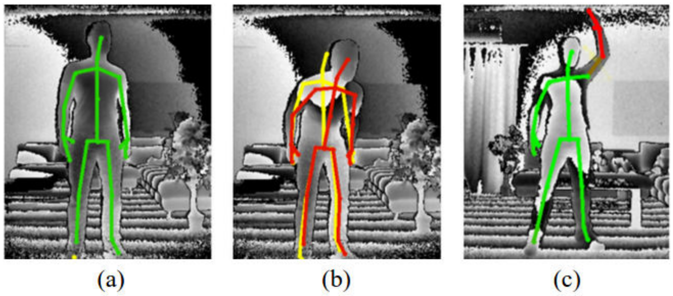

Fig. 3. Example on visual feedback for a standing posture – (a) overall error (green indicates correct posture) (b) overall error with hint (in yellow); (c) individual joint error (in red).

A. Feedback from Quantization Errors

As pointed out by Kohonen [16], a straightforward way of generating an error measure from a trained SOM-like network is to estimate the quantization error given by the difference between the input $\mathbf{x}$ and its best-matching neuron defined as:

$$\mathfrak{b}(\mathbf{x}) = \text{argmin}_j \|\mathbf{x} - \mathbf{w}_j\|. \tag{12}$$

With this mechanism in mind, we define two methods: one for computing an overall error $e(t)$ for the whole-body posture, and the other for computing individual errors $e_i(t)$ associated to specific body joints (Fig. 3). For this purpose, we exploit a distinctive characteristic of the MergeSOM with respect to other recursive alternatives that both the weight and context descriptors lie within the same data space. Thus, we can compute the overall error from the current input $\mathbf{x}$ as the recursive activation (Eq. 2) over its best-matching neuron u:

$$e(t) = (1-\alpha)\cdot \|\mathbf{x} - \mathbf{w}^u\|^2 + \alpha \cdot \|\mathbf{y} - \mathbf{c}^u\|^2. \tag{13}$$

Similarly, we can compute individual errors performing the activation of Eq. 13, but this time considering each dimension of $\mathbf{x}, \mathbf{w}, \mathbf{y}$, and $\mathbf{c}$ individually and subsequently combining the dimensionality-wise activation as follows:

$$e_i(t) = e_{\rho(i,x)} + e_{\rho(i,y)} + e_{\rho(i,z)}, \tag{14}$$

with $\rho(i,x)$ being the dimension corresponding to joint i and its x, y, z components.

B. Feedback Through Prediction

Recursive prediction carried out through successor chains (Eq. 11) seems to adequately tell how a correct movement is to be continued. During its execution, we could then measure the correctness of such movement in terms of how much it deviates from its predicted pattern. We then define a function that provides feedback on the current posture $\mathbf{p}(t)$ (Eq. 1) with respect to the predicted postures from i previous time steps as follows:

$$\mathfrak{f}_{\mathfrak{b}}(t,i) = \|\mathfrak{s}^i(\mathbf{x}^{\mathfrak{b}(\mathbf{p}(t-i))}) - \mathbf{p}(t)\|. \tag{15}$$

This function, however, does not take into account alternative prediction paths and does not yield variance in timing. We then assume that given the best-matching neurons at time $t-i$ and t, we can compute to which extent the latter would have been predicted by the former, without considering the time difference i, so that $e(t)$ will yield a small value for a correct prediction, whereas a higher value for a wrong prediction. For this purpose, we infer the probability of a neuron v being predicted as a distant successor of another neuron u by computing the shortest path distance $\delta(u,v)$ from u to v in a graph $G_{\mathfrak{s}}$ containing the successor distances of all the neurons. In this case, the feedback function is defined as:

$$\mathfrak{f}_{\delta}(t,i) = \delta(\mathfrak{b}(\mathfrak{p}(t-1)), \mathfrak{b}(\mathfrak{p}(t))). \tag{16}$$

In order to project the output of the prediction-based techniques to use as overall error $e(t)$, we aggregate their output over the last λ previous time steps as follows:

$$e(t) = \sum_{i=1}^{\lambda} \mathfrak{f}(t,i)\cdot\gamma^i, \tag{17}$$

where γ is a decay factor to favour more recent previous time steps over older ones.

VI. Experimental Results

We now present our experimental results on a data-set of 3 powerlifting movements used for the training, validation, and test of the proposed system.

The data collection took place at the Kinesiology Institute of the University of Hamburg, Germany, where 17 volunteering participants (9 male, 8 female) performed 3 different powerlifting exercises:

- 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;
- 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.

For a thorough evaluation of our system, we also recorded a set of typical mistakes for each routine:

- E1)
  - M1) *Good morning*: Raising the hips without raising the chest with an excessively horizontal back angle;
  - M2) *Half squat*: Going only halfway down to the ground;
  - M3) *Knees in*: Bow the knees toward each other during the lift.
- E2)
  - M1) *No lockout*: The execution is carried out properly, but the lift is stopped before the lockout;

In: IEEE International Joint Conference on Neural Networks (IJCNN), pp. 2973-2978, Killarney, Ireland (2015)

M2) *Rounded back*: The back is heavily rounded during the lift instead of being in a straight line.

E3) M1) *Low elbows*: Lateral lifts performed with the wrists being higher than the elbows.

Correct and incorrect executions were captured with a Kinect v2 at 30 frames per second. The participants executed the routines frontal to the sensor placed at 1 meter from the ground. We processed video sequences with Kinect SDK to segment motion and extract 3D joint coordinates frame by frame. We manually segmented single repetitions for all exercises.

A. Training Parameters

We trained a different MergeSOM network for each routine. The MergeSOM training parameters were empirically set by choosing the temporal quantization error (TQE) as a performance measure. The TQE is a temporal generalization of the SOM quantization error [16] and measures the average quantization errors of the network over the past inputs [15]. It thus measures the specificity of neurons to sequences of data. In Fig. 4 we show the TQE with different pairs of the recursive parameters α and β (Eq. 2 and 3 respectively) for a network of 1600 neurons after 100 training epochs with linearly decaying learning rates $\epsilon_0 = 0.1$ and $\epsilon_f = 0.01$, and exponentially decreasing neighbourhood rates $\sigma_0 = \sqrt{N_n}$ and $\sigma_f = 0.0001$. The lowest TQE was found for the values $\alpha = 0.6$ and $\beta = 0.7$. For providing feedback over trained networks, we empirically set the aggregation parameter $\gamma = 0.8$ with $\lambda = 100$ (Eq. 18).

B. Evaluation

We evaluated the three feedback extraction techniques introduced in Sec. 5 on individual subjects and subsequently on multiple subjects. We divided the correct motion data with 3-fold cross-validation into training and test repetitions and trained the models on data containing correct movement sequences. For the test phase, both correct and incorrect movement sequences were used. Our expectation was that the output of feedback functions will be higher for sequences containing mistakes. For evaluation purposes, we used a binary classification test in which each sequence is labelled by the system as *positive* if a mistake was detected, and *negative* otherwise. In this setting, we empirically defined error thresholds τ for each feedback function, with $\tau(e(t)) = 0.2$, $\tau(\mathfrak{f}_{\mathfrak{b}}(t,i)) = 2.5$, and $\tau(\mathfrak{f}_{\delta}(t,i)) = 0.6$. Fig. 5 shows visual feedback for a correct squat sequence and another sequence containing *knees in* 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), true negative rate (TPR), and positive predictive value (PPV). Results for single-subject and multiple-subject data on E1, E2, and E3 routines for the three feedback functions are displayed in Table I and Table II respectively. Single-subject evaluation shows that the system successfully provides feedback on posture errors for the set of 3 training movements with high accuracy. The evaluation for multiple-subject data shows rapidly decreased performance due to many reported false positives, especially for the prediction-based techniques $\mathfrak{f}_{\mathfrak{b}}$ and $\mathfrak{f}_{\delta}$. A likely cause is the model size

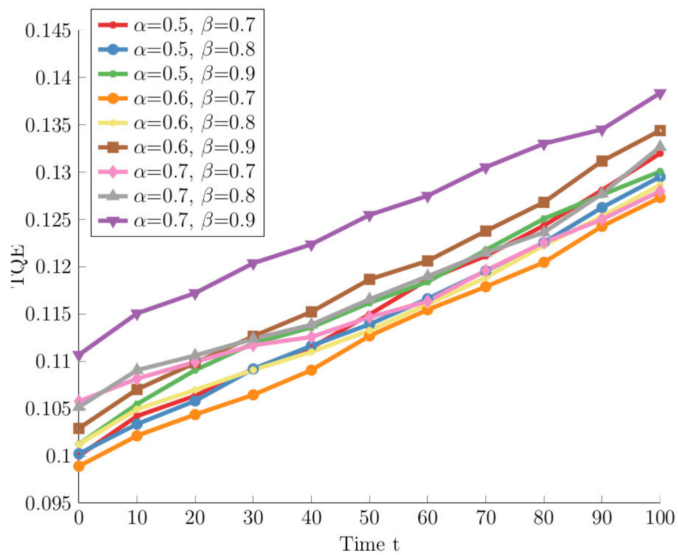

Fig. 4. Evaluation of the recursive activation parameters α and β with the temporal quantization error (QTE) over time instances. Lowest QTE found for $\alpha = 0.6$ and $\beta = 0.7$.

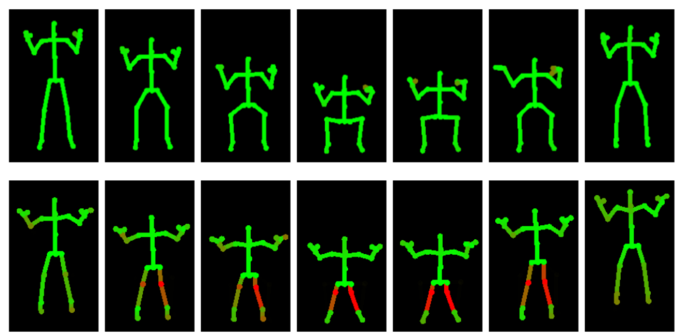

Fig. 5. Visual feedback for correct squat sequence (first line), and a sequence containing *knees in* mistake (second line, joints in red).

being adequately large for learning a movement from one subject, but not from multiple subjects. Thus, prediction of future poses on multiple-subject data fails.

VII. Discussion

In this work, we presented a learning system for estimating automatic feedback on a set of well-defined movement sequences captured with a Kinect sensor. Our learning architecture is built upon an extended variant of neural network self-organization for processing spatio-temporal patterns that allows to predict future movements and provides an assessment on the correctness of a current movement sequence.

Our evaluation on three powerlifting routines showed high system performance for single-subject data. The system was able to predict a set of learned sequences and then provide visual hints with the needed feedback to correct postural mistakes in real time (Fig. 5). As shown in Table I, feedback functions yielded varying performance depending on different movement routines. The evaluation on multiple-subject data shows a decreased performance for all the movement sequences (Table II). However, this aspect does not necessarily represent a drawback, since the main focus of our system is

In: IEEE International Joint Conference on Neural Networks (IJCNN), pp. 2973-2978, Killarney, Ireland (2015)

TABLE I. SINGLE-SUBJECT EVALUATION.

		TP	FN	TN	FP	TPR	TNR	PPV
E1	$e(t)$	41	4	32	1	0.91	0.97	0.97
	$\mathfrak{f}_{\mathfrak{b}}$	35	10	33	0	0.77	1	1
	$\mathfrak{f}_{\delta}$	29	16	26	7	0.64	0.78	0.8
E2	$e(t)$	23	1	19	1	0.95	0.95	0.95
	$\mathfrak{f}_{\mathfrak{b}}$	24	0	20	0	1	1	1
	$\mathfrak{f}_{\delta}$	24	0	20	0	1	1	1
E3	$e(t)$	63	0	26	0	1	1	1
	$\mathfrak{f}_{\mathfrak{b}}$	63	0	26	0	1	1	1
	$\mathfrak{f}_{\delta}$	55	8	26	0	0.87	1	1

TABLE II. MULTI-SUBJECT EVALUATION.

		TP	FN	TN	FP	TPR	TNR	PPV
E1	$e(t)$	326	1	40	118	0.99	0.25	0.73
	$\mathfrak{f}_{\mathfrak{b}}$	326	1	7	151	0.99	0.04	0.68
	$\mathfrak{f}_{\delta}$	320	7	0	158	0.98	0	0.66
E2	$e(t)$	129	0	43	78	1	0.35	0.62
	$\mathfrak{f}_{\mathfrak{b}}$	127	2	0	121	0.98	0	0.51
	$\mathfrak{f}_{\delta}$	129	0	0	121	1	0	0.51
E3	$e(t)$	123	0	32	17	1	0.65	0.88
	$\mathfrak{f}_{\mathfrak{b}}$	123	0	8	41	1	0.16	0.75
	$\mathfrak{f}_{\delta}$	123	0	1	48	1	0.02	0.72

tailored feedback for a specific person over person-independent generalization of movements. Nevertheless, a possible approach for addressing this limitation is the use of growing networks that dynamically adapt the model size to represent a greater number of sequences. Since the recursive mechanism of the MergeSOM does not depend on the network topology, we could adopt a recursive incremental model such as Merge Growing Neural Gas [18], for which the number of neurons and the lattice topology are not established a priori. Additionally, the fading memory used in the MergeSOM model loses predictive power on long motion sequences. For addressing this issue, we could use an extension of MergeSOM that allows to store multiple contexts for each neuron [19]. This mechanism would allow to train the networks with multiple β values and then average their feedback output yielding higher precision over a range of times scales. Furthermore, this extended model could be combined with a growing topology [20].

In summary, our system successfully computed real-time feedback on a set of learned posture sequences. The reported results show the promising contribution of our learning system for movement assessment and person-specific assistance in fields where the correct execution of accurate movement routines plays a crucial role such as sports and physical rehabilitation.

ACKNOWLEDGMENTS

The authors would like to thank Cornelius Weber for valuable feedback and Erik Strahl for technical support.

This work was partly supported by the DAAD German Academic Exchange Service (Kz:A/13/94748) - Cognitive Assistive Systems Project, and by the DFG German Research Foundation (grant #1247) - International Research Training Group CINACS (Cross-modal Interaction in Natural and Artificial Cognitive Systems)

REFERENCES

[1] B. P. Hamill. Relative safety of weightlifting and weight training. *Journal of Strength & Conditioning Research*, 8(1):53–57, 1994.

[2] S. Hancock, F. Wyatt, and J. Lon Kilgore. Variation in barbell position relative to shoulder and foot anatomical landmarks alters movement efficiency. *International Journal of Exercise Science*, 5(3):183–195, 2012.

[3] M. Rippetoe. Starting strength: Basic barbell training, 3rd ed. *The Aasgaard Company*, Wichita Falls, TX, 2011.

[4] D. S. Alexiadis, P. Kelly, P. Daras, N. E. O'Connor, T. Boubekeur, and M. Ben Moussa. Evaluating a dancer's performance using Kinect-based skeleton tracking. In Proceedings of the 19th ACM international conference on Multimedia, pp. 659–662. ACM, 2011.

[5] G. I. Parisi and S. Wermter. Hierarchical SOM-based detection of novel behavior for 3D human tracking. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 1380–1387, Dallas, TX, USA, 2013.

[6] G. I. Parisi, C. Weber and S. Wermter. Human action recognition with hierarchical growing neural gas Learning. In *International Conference on Artificial Neural Networks (ICANN)*, pp. 89–96, Hamburg, Germany, 2014.

[7] F. Anderson, T. Grossman, J. Matejka, and G. Fitzmaurice. YouMove: Enhancing movement training with an augmented reality mirror. In Proceedings of the 26th annual ACM symposium on User interface software and technology, pp. 311–320. ACM, 2013

[8] K. Rector, C. L Bennett, and J. A Kientz. Eyes-free yoga: an exergame using depth cameras for blind & low vision exercise. In Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility, pp. 12. ACM, 2013.

[9] E. Velloso, A. Bulling, H. Gellersen, W. Ugulino, and H. Fuks. Qualitative activity recognition of weight lifting exercises. In Proceedings of the 4th Augmented Human International Conference, pp. 116–123. ACM, 2013.

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

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

[12] T. Voegtlin. Recursive self-organizing maps. *Neural Networks*, 15(8):979–991, 2005.

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

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

[15] B. Hammer, A. Micheli, A. Sperduti, and M. Strickert. Recursive self-organizing network models. *Neural Networks*, 17(8):1061–1085, 2004.

[16] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[17] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[18] A. Andreakis, N. v. Hoyningen-Huene, and M. Beetz. Incremental unsupervised time series analysis using merge growing neural gas. In: Advances in Self-Organizing Maps, pp. 10–18. Springer, 2009.

[19] P. A. Estevez and R. Hernandez. Gamma SOM for temporal sequence processing. In: Advances in Self-Organizing Maps, pp. 63–71. Springer Berlin Heidelberg, 2009.

[20] P. A. Estevez and J. R. Vergara. Nonlinear time series analysis by using gamma growing neural gas. In: Advances in Self-Organizing Maps, pp. 205–214. Springer, 2013.