Computers & Graphics (2018)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Classification of motor errors to provide real-time feedback for sports coaching in virtual reality — A case study in squats and Tai Chi pushes

Felix Hülsmann[a,b,*], Jan Philip Göpfert[c], Barbara Hammer[c], Stefan Kopp[a,1], Mario Botsch[b]

[a] *Computer Graphics and Geometry Processing, Bielefeld University, Universitätsstraße 25, 33615 Bielefeld, Germany*
[b] *Social Cognitive Systems, Bielefeld University, Universitätsstraße 25, 33615 Bielefeld, Germany*
[c] *Machine Learning, Bielefeld University, Universitätsstraße 25, 33615 Bielefeld, Germany*

ARTICLE INFO

Article history:
Received August 24, 2018

sports coaching in virtual reality, motor learning environments, motor performance quality, human motion analysis, auto-generated augmented feedback

ABSTRACT

For successful fitness coaching in virtual reality, movements of a trainee must be analyzed in order to provide feedback. To date, most coaching systems only provide coarse information on movement quality. We propose a novel pipeline to detect a trainee's errors during exercise that is designed to automatically generate feedback for the trainee. Our pipeline consists of an online temporal warp of a trainee's motion, followed by Random-Forest-based feature selection. The selected features are used for the classification performed by Support Vector Machines. Our feedback to the trainee can consist of predefined verbal information as well as automatically generated visual augmentations. For the latter, we exploit information on feature importance to generate real-time feedback in terms of augmented color highlights on the trainee's avatar. We show our pipeline's superiority over two popular approaches from human activity recognition applied to our problem, k-Nearest Neighbor, combined with Dynamic Time Warping (KNN-DTW), as well as a recent combination of Convolutional Neural Networks with a Long Short-term Memory Network. We compare classification quality, time needed for classification, as well as the classifiers' ability to automatically generate augmented feedback. In an exemplary application, we demonstrate that our pipeline is suitable to deliver verbal as well as automatically generated augmented feedback inside a CAVE-based sports training environment in virtual reality.

1. Introduction

Coaching environments for motor learning are becoming a more and more popular research topic in virtual reality (VR) [1, 2, 3, 4]. They offer possibilities that are not imaginable in classical coaching, such as augmented feedback strategies or multisensory stimuli. Further, VR motion capture systems are able to provide objective kinematic data of a trainee in real-time. In their review, Miles et al. especially highlight the flexibility of VR environments and the possibility to provide extra information as a reason to use them in sports training [5]. See [6] for a general discussion of VR techniques in sports. Obviously, feedback on the trainee's performance is crucial for the success of coaching systems. A coaching system has to assess the quality of the exercise — in the following called *motor action* — performed by a trainee, and communicate this information in terms of feedback. Often, algorithms developed in the context of sports coaching either focus on the assessment of the performed motion, or on the generation of feedback. In this article, we propose an integrated pipeline that performs the detection of typical motor errors and provides results that are directly

*Corresponding author

Fig. 1. In our real-time VR coaching environment, a trainee performs exercises while being observed by a virtual coach. Our algorithm provides the virtual coach with the information necessary to apply his feedback strategies in an online manner.

interpretable in terms of automatically generated augmented visual feedback. Further, results can be linked to already existing verbal feedback strategies. In order to develop such an integrated solution, specific requirements, additional to a high classification quality, hold for the assessment of the trainee's performance:

R1 Connectable to existing feedback strategies: A coaching system should spot the occurrence of typical errors in the trainee's performance that can be linked to feedback strategies that have already been established by coaches in the real-world.

R2 Real-time: Whether feedback at early stages of the movement should be provided must be determined by the applied coaching strategies. However, to provide the coaching system a maximal range of applicability, components that assess the motor performance should deliver their results as soon as possible. If, for instance, the starting posture of a motor action is already problematic, the system should be able to intervene, to prevent the trainee from performing potentially problematic movement patterns, or even from hurting herself. For an analysis of real-world coaching and timing for the squat, we refer to [7, 8, 9].

R3 Interpretability: The classification process should be transparent and interpretable. It ideally provides information on the classified errors that can be used to generate augmented feedback in the virtual environment. Furthermore, an interpretable classifier gives experts the ability to verify whether the classifier works in a plausible way.

R4 Conservative size of data sets: Recording high quality training data and recruiting experts to perform data annotation is time consuming and expensive. Thus, the system should be able to deal with limited data sets to ensure practical usefulness of the coaching system.

R5 Minimal manual work: Manual work is expensive and reduces the usefulness of developed approaches in real-world applications. The classifier should require as few as possible, manually coded, expert knowledge.

We argue that research in the area of VR that focuses on these aspects, thus keeps in mind the ideal integration of the kinematic movement analysis in a VR coaching system, would advance the field of sports and rehabilitation coaching in virtual environments. To this end, our contributions are as follows:

- We propose a new, interpretable, and real-time pipeline towards the classification of error patterns in motor performances. It uses a reference-based Dynamic Time Warping of movement prefixes as a basis for a feature selection using Random Forest. The selected features are in a final step classified by Support Vector Machines (SVM).
- We demonstrate that our pipeline can automatically generate real-time augmented feedback based on a trainee's motion. Further, we show an exemplary application of our pipeline in a CAVE-based VR coaching environment (see Figure 1), including verbal as well as augmented visual feedback.

We use two data sets for evaluation. They consist of body-weight squats and Tai Chi push movements. These are full-body motor actions that are used in the context of rehabilitation as well as for sports training. When executed by novice trainees, various error patterns can be observed. Based on these data sets, we show the ability of our pipeline to beat the popular classifier KNN-DTW that has been found to be difficult to beat for typical time series classification tasks as shown in [10, 11]. Further, we compare our pipeline to a recent neural-network-based approach to human activity recognition [12]. Our pipeline does not only provide better classification results, but is also better suited to generate augmented visual feedback. We use skeleton data as input to provide classification results, as well as augmented visual feedback in real-time. Due to using skeleton data, our pipeline can be applied in combination with various motion capture systems, as they typically output kinematic features for the tracked subject's joints.

2. Related Work

To assess the quality of human motor performances, two main approaches have been applied. The first approach (Section 2.1) is to engineer a highly specialized method, e.g., for the evaluation of feedback strategies for a very specific type of motor action. Often, a model for specific performance patterns is manually designed drawing from expert knowledge. The second direction (Section 2.2) consists in using more general, data-based approaches that have already been used in the context of motor learning and motion assessment. In Section 2.3, we focus on more general approaches from machine learning that have not been typically used in the field.

2.1. Specific, Manually Designed Approaches

Houmanfar et al. use a manually designed scoring function to represent patients' performance changes in a rehabilitation setting [13]. Even though this approach provides compelling results in the field of application, no detailed information on occurred error patterns is gained, which would be necessary

for the application of complex coaching strategies. Other approaches make use of rule-based systems to detect the occurrence of certain error patterns. In the context of yoga training, Rector et al. define optimal yoga poses [14]. De Kok et al. went one step further by manually defining error patterns [2] that focus on the whole trajectory. One major advantage of the approaches by Rector et al. or de Kok et al. is their real-time capability (R2). Specific feedback strategies, linked to typical error patterns, can be applied immediately (R1) and the rules can be directly interpreted by experts (R3). Nearly no training data is needed (R4). Further, the results are deterministic. If the rules are correct and exhaustive, and the motion capture system works properly, an incorrect classification is unlikely to occur. However, the rules are designed manually which violates (R5). It is mostly not trivial — even when interviewing sports coaches — to obtain exact information about which features are significant or where to draw the border between a correct or an incorrect movement. And even if it is possible, the design of rules requires enormous manual effort. For each motor action and for each type of error, a detailed investigation on how to describe the motor action and the error has to be performed. For complex error patterns, this quickly becomes infeasible. Thus, we focus on approaches that automatically learn most of their information from data.

2.2. Data-based Approaches for Performance Assessment

Taylor et al. classify error patterns in rehabilitation exercises using a combination of rule-based segmentation and Adaptive Boosting on a set of manually defined features [15]. In a within-subject cross validation, the authors obtain highly convincing results. However, classification performance decreases significantly when generalizing to new subjects. Furthermore, the design of feature sets requires additional manual work. Yurtman and Barshan proposed an extension of Dynamic Time Warping (DTW) that is able to detect multiple occurrences of multiple exercise types in trajectories as well as to classify error patterns [16]. Classification is performed by comparing the just performed motion to pre-recorded templates and then selecting the best matching one similar to 1-nearest-neighbor Dynamic Time Warping (1NN-DTW). Combinations of multiple error patterns cannot be considered as long as they are not included as individually pre-recorded templates. Further, the authors did not test for inter-subject performance. Another prototype-based approach was described by Parisi et al. who propose a recursive neural network for the assessment of sports motion [17]. As indicator for motion quality, the system compares the performed motion to the desired continuation of an exercise. Single-subject evaluation leads to very high accuracies, whereas tests with multiple subjects lead to a high number of false positives. O'Reilly et al. use a neural network classifier to differentiate between correct and incorrect performances of squats and to classify error patterns [18]. A leave-one-out cross validation resulted in an accuracy of 80 % to distinguish between correct and incorrect, but only in an accuracy of 57 % for the classification of error patterns. Similar experiments were conducted by Giggins et al. [19, 20]. Kianifar et al. present an approach towards distinguishing between good, moderate, and bad performances of squat movements [21]. They use a feature vector based on manually designed features, such as skewness and range, whose dimensionality is reduced using Sparse Principal Component Analysis (SPCA). Decision Trees are used for classification. The presented approach is only able to distinguish between three coarse classes of quality and cannot spot single error patterns. In addition, manual effort is needed for feature preparation. Furthermore, SPCA is an unsupervised algorithm, which searches for a set of sparse principal components that cover as much as possible of the variance inside the data [22]. This is problematic as most of the variance could be induced due to individual differences rather than performance errors. This might be especially risky for sports movements that can differ considerably between subjects.

Overall, the data-based approaches employed in the context of sports and rehabilitation applications have three weaknesses in terms of their classification performance. First, it is often not analyzed how well the trained classifiers generalize to new subjects. Some of the addressed approaches require the system be re-trained for each user. This procedure can rarely be applied to real world coaching applications as subjects are often physically not able to provide all the required training data. Second, the motor actions and error patterns are often rather simple. Some systems only distinguish between, e.g., "good" or "bad" for a motor action that only involves a very small number of joints. Especially algorithms that use comparisons with prototypes will perform worse on more subtle errors or more complex movements when performing multi-subject evaluation as shown in [17, 15]. Here, different styles and differences between subjects might predominate differences induced by movement patterns underlying the motor errors. This holds especially as many types of complex sports movements can be executed correctly yet with different individual styles [23]. Furthermore, an analysis that only relies on an overall deviation from a prerecorded desired performance, including task-irrelevant deviations, is non-optimal when aiming at improving the trainee's performance [24, 25]. One reason is that some muscle groups are often less requested, making the associated body parts less relevant for the successful execution of a movement.

2.3. General Approaches for Human Activity Recognition

Indeed, the classification of errors in motor performances is a special case of time series classification. In this area, ground-breaking work was performed by Wilson and Bobick, who used hidden Markov models (HMM) for the recognition of gestures [26]. Other methods are based on decision trees [27], SVMs [28], or Multi-Layer Perceptrons (MLP) [29]. DTW is usually applied to temporally align two recorded trajectories. As a pseudo-metric combined with a subsequent classification, DTW has a highly positive impact on motion classification [30, 31, 10]. Xi et al. provide an extensive review comparing a large set of available classification methods, such as HMMs, MLPs, and decision trees on time series data [10]. They show that no tested classifier is able to beat a combination of DTW and 1-Nearest-Neighbor (1NN-DTW). 1NN-DTW compares the query trajectory to each available training trajectory

using DTW as distance measure. Then the most similar training trajectory is used to predict the label of the query trajectory. The superiority of this approach in comparison with other classifiers, such as Random Forests, SVM, Bayes Networks, et cetera, is supported by work from Bagnall and Lines [11]. Likewise, Yurtman and Barshan achieved good classification results using a method similar to 1NN-DTW, which, however, was limited to simple movement patterns and was not evaluated with respect to generalization to new subjects [16].

Recently neural networks have been frequently used in the related field of skeleton-based human activity recognition. They typically reach a high classification performance, especially for large training data sets. Recurrent Neural Networks (RNNs) allow for an online recognition of motor actions. For instance Li et al. propose a tree-like hierarchy of RNNs to distinguish between actions learned on thousands of sequences. Other approaches such as [33] focus on Long Short-Term Memory (LSTM) networks with trust gates to model temporal properties of the data. Another approach works on a combination of video and skeleton data [34]. Here, data is preprocessed by convolutional layers to generate higher level features. The classification is then performed by an LSTM network and a combination of classification and regression layer. Liu et al. propose context-aware attention LSTM networks to allow the network to focus on informative joints for a specific motor action. This is achieved via combining Spatio-Temporal LSTM layers with a dedicated global context memory. A recent approach by Núñez et al. also performs a temporal and spatial preprocessing of the input to improve the classification performance [12]. A convolutional neural network (CNN) preprocesses data on the spatial as well as on the temporal domain to generate higher-level features that contain relevant information for the classification task. This information is then passed to a LSTM network to account for a larger temporal context.

Two approaches seem most suitable in the context of error classification of sports movements. KNN-DTW as well as the combination of CNNs with LSTMs (from now on called CNN-LSTM). The combination of nearest-neighbor classifiers with DTW is popular and difficult to beat in classic sequence classification [11, 10]. Furthermore, related approaches have already been successfully used for the assessment of human motor performances [16]. CNN-LSTM has recently been proposed in the field of human activity recognition [12]. Although the approach has not been demonstrated to work for subtle patterns such as errors in motor performances, the preprocessing step based on CNNs seems promising as it can be expected to learn the relevant features for specific error patterns. Furthermore, the well established field of CNNs provides methods to estimate the saliency of specific features of the input of the classifier [36], which would increase the interpretability (R3) of the approach. Both approaches can be linked to existing feedback strategies, as they are — given a sufficient classification quality — able to classify typical error patterns (R1). Further, they are fully data-driven and thus require only few manual work (R5). CNN-LSTM can be expected to work in real-time (R2). Further, this approach is described as being able to work even for small data sets (R4). In our evaluation, we show that our pipeline outperforms KNN-DTW as well as CNN-LSTM.

(a) Marker placement. (b) Skeleton representation.

Fig. 2. Marker setup and reconstructed skeleton representation.

3. Domain and Data Set

To build data sets for training and testing, we first identify error patterns for the squat as well as for the Tai Chi push via consulting coaches (squat: 14 coaches, on median 9 years of experience. Tai Chi push: 1 coach, 14 years of experience), literature (e.g., [37], [38]), as well as videos from coaching sessions (for the squat only, partly from corpus described in [8], partly recorded in our own lab).

In a second step, motion data for both motor actions was recorded using an OptiTrack motion capture system (10 Prime 13W cameras). Passive markers were mostly attached to a customized motion capture suit; markers at the arms and the hands were directly attached to the subjects' skin (see Figure 2a). The usage of a marker-based system, which is a well evaluated standard procedure in biomechanical analysis, allows us to obtain highly precise motion capture data, that also covers fine-grained errors and variations in motor performances. The marker suit is designed in a way that allows a reliable positioning of markers, even if subjects sweat or are breathing heavily due to exhaustion. No other hardware that suffers from these issues was attached to the subjects. The motion capture system outputs

Table 1. Analyzed error patterns in the execution of a squat (cf. data from [8]). The numbers denote the quantity of incorrect and correct executions of the squat in our data, with respect to the corresponding pattern.

Performance Error Pattern	#Erroneous	#Correct
arched neck	33	29
feet distance not sufficient	45	33
hips do not initiate movement	23	51
hollow back	34	42
incorrect weight distribution	51	16
knees tremble sideways	23	33
legs extended at end	42	38
not symmetric	17	46
too deep	51	34
wrong dynamics	61	27

(a) Incorrect weight distribution.

(b) Right foot is incorrectly rotated.

(c) Arms return at chest height.

(d) Knees are in front of the toes.

(e) The left leg is moved forward.

(f) Arms and legs are asynchronous.

Fig. 3. The images depict examples and symptoms for error patterns of the Tai Chi push mapped onto a virtual character. These images can only provide a rough overview of how the errors could look like. Specific occurrences can deviate and require information on the rest of the movement. The patterns "non-uniform movement" and "asynchronous" are not visualized as it is difficult to depict aspects of these error patterns in one single image.

kinematic features for $k = 19$ joints (see Figure 2b) per frame at 120 Hz. Each frame consists of k joint rotations as well as k joint positions. Joint rotations are represented as quaternions $\mathbf{q}_1, \ldots, \mathbf{q}_k$. Each quaternion denotes the rotation of a joint with respect to its parent. The root rotation $\mathbf{q}_1$ describes the rotation of the root with respect to its rotation at the beginning of the movement. As root joint we use the hips. The joint positions are represented by vectors $\mathbf{t}_1, \ldots, \mathbf{t}_k \in \mathbb{R}^3$. Each $\mathbf{t}$ denotes the translation relative to the position of the root joint at the beginning of the movement. Further, we use joint angles as Euler angles, calculated from the quaternion representation, which correspond to flexion/extension, abduction/adduction and twist of the corresponding joint.

The squat data set consists of $N = 96$ squat movements coming from 50 subjects. The Tai Chi data set consists of $N = 120$ recordings coming from 24 subjects. All recordings were annotated by an expert for the presence of any of the error patterns. The expert had to add an intensity rating for each error as well as confidence ratings for each decision. These ratings were combined into a score in the interval $[0, 1]$ by averaging. Only ratings with a score above 0.5 were used for the experiment. We selected the error patterns that appeared with a frequency of at least 15 positive and negative examples for training. The resulting patterns and their frequency in the training data are listed in Table 1 and Table 2. Figure 3 and Figure 4 provide a visual overview of the errors from typical recordings mapped on a virtual character.

4. Classification

Our classification pipeline is trained on the data described in Section 3. It learns a classifier for each error pattern, considering each training trajectory as one data point with the label *pattern occurs* or *pattern does not occur*. In the final application, the pipeline receives a stream of frames of skeleton data from a motion capture system and outputs a label w.r.t. each error pattern. As we use skeleton data, our pipeline is highly flexible. The architecture is not restricted to specific input data, but can also be used with various motion capture algorithms, such as marker-based, but also marker-less ones, for instance [39, 40, 41, 42].

In order to develop a preferably simple classifier that satisfies all our requirements, we rely on Support Vector Machines (SVMs). They are one of the most successful machine learning algorithms in general [43]. Additionally, they are fast and especially linear kernel SVMs are easy to interpret. For classification, the SVM only has to determine on which side of a hyperplane an input query lies. More technical and analytical information concerning SVMs can be found in [44, p. 325].

In the context of motion trajectories, SVMs cannot be directly applied as they require input vectors of a fixed size. In order to represent all data on a canonical time line of fixed size, we exploit the general similarity between the trajectories that all represent the same motor action. We use DTW to warp all training and input trajectories into the timing of a fixed reference trajectory T_r. A detailed theoretical and analytical investigation of this algorithm can be found in [45, p. 69]. For each frame t of T_r, the corresponding frame in the to-be-warped trajectory is extracted. Next, for these frames, we extract all joint angles in Euler angle representation as well as the joint positions. The resulting feature vector thus has size $6|T_r|k$, where $|T_r|$ is the number of frames of the reference trajectory and k the number of joints. We have $k = 19$ and $|T_r| = 902$ for the squat movement and $|T_r| = 782$ for the Tai Chi push.

The feature vector of size $6|T_r|k$ comprises many irrelevant features. For instance, we intuitively do not consider the rotation of the wrist to be related to having a straight back. The SVM classifier might suffer from this high number of irrelevant features as shown by Weston et al. [46] and Chen and Lin [47]. According to their results, we assume a robust feature selection method to be able to help improving classifier performance. A good introduction into the area of feature selection methods can be found in [48]. In the past, Random Forests (RF) have often demonstrated to lead to good feature selection results [49, 47, 50, 51]. We use Random Forests as they tend to lead to especially good results for small sample sizes and a large number of features. Random Forests are based on Decision Trees, which learn a hierarchical set of rules to distinguish between classes. Thereby, they implicitly weight the importance of each feature. See [52] for more analytical information on Random Forests. An in-depth analysis of the theoretical background and the statistical properties of Random Forests can be found in [53]. Random Forests could be directly applied as

(a) Hips do not initiate the movement.

(b) Incorrect weight distribution.

(c) Hollow back.

(d) Too deep.

(e) Arched neck.

(f) Wrong dynamics and feet distance not sufficient.

(g) Wrong dynamics and legs extended at end.

(h) Not symmetric and feet distance not sufficient.

Fig. 4. The images depict examples and symptoms for error patterns of the squat mapped on a virtual character. These images can only provide a rough overview of how the errors could look like. Specific occurrences can deviate and require information on the rest of the movement. The pattern "knees tremble sideways" is not visualized as it is difficult to depict aspects of this error pattern in one single image.

classifiers, however classification using Random Forests leads to high computational cost, as all trees in the forest must be considered. Thus, we use a feature selection based on Random Forests as preprocessing for the SVM-based classification during training. We train one Random Forest for each error pattern on the feature vectors extracted after DTW. To train the trees, we use the Gini impurity as criterion to optimize the decision rules [54]. As break condition for growing, we require all leaves to contain only a single class or less than two samples. We observed a number of 200 trees to lead to good results. For each error pattern, the Random Forest assigns an importance value to each feature via averaging the relative importance of the feature in each decision tree. Following an idea of Bi et al. [55], we add 10 random features to each frame before performing the feature weighting. The average of their importance values is used as threshold to discard irrelevant features. For the squat, this leads to 570 features on average per error pattern (from originally over 100,000 features). For the Tai Chi push, we end up with about 500 features. We use the implementation of Random Forests that is provided by scikit-learn [56] in version 0.17.1. For each error pattern, we train one two-class SVM with linear kernel on the selected features, which are standardized via scaling to unit variance and removing the mean. The implementation of the SVM is provided by scikit-learn. Formally, the classification is finally performed via evaluating the sign of:

$$\mathbf{w}^T \mathrm{fs}(\mathrm{warp}(\mathbf{T_x})) + b, \tag{1}$$

where $\mathbf{w}$ is the weight vector which specifies the orientation of the decision surface of the SVM and b is the bias which specifies the location of the decision surface. These parameters are trained by the SVM in the final step. warp denotes the warping of input trajectory $\mathbf{T_x}$ into the timing of the reference trajectory $\mathbf{T_r}$ and fs denotes the selection of the relevant features and the scaling required for the SVM classifier.

Due to the classic DTW, this classifier only starts the classification as soon as an exercise has been finished. In order to obtain a real-time classification, we provide two extensions to this procedure: First, we use Weight-Optimized Open-End DTW (OE-DTW), as proposed in [57], to make the temporal alignment work online. As a second extension, we train multiple classifiers on prefixes of our training data, to be able to select the best matching classifier for each point in time. In more detail, the training works as follows. First, all training trajectories are warped into the timing of the reference trajectory. Then, for each error pattern, we train the above classifiers on prefixes of the training trajectories in 5 % steps. The online classification looks as follows. A trainee has performed a part of the exercise, the input prefix. We warp this input prefix into the timing of the reference trajectory using OE-DTW. OE-DTW returns, additional to the alignment, the percentage c of the reference that corresponds to the input prefix. If $c \in [5\%, 10\%)$, we select the first of our classifiers, if $c \in [10\%, 15\%)$, we select the second, and so on. We apply the classifier on the part of the warped input that matches the prefix of the reference we used for training. See Figure 5 for a visualization of the classification procedure.

5. Visual Augmented Feedback

We provide feedback in terms of a visual augmentation of the trainee's avatar. Body parts that are related to a just performed error are highlighted in red. The manual selection of the important body parts as well as the point in time when they typically

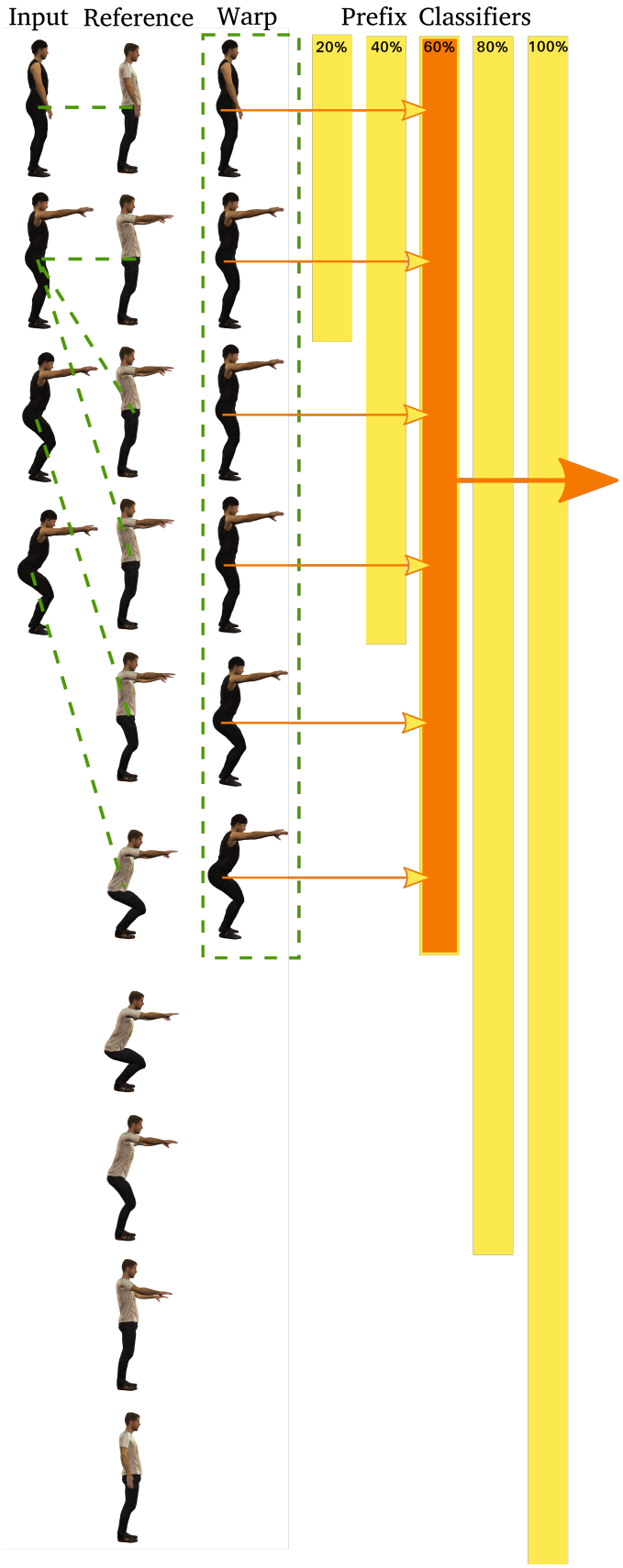


Fig. 5. Online classification of error patterns: The trainee (left column) has nearly reached the deepest point of the squat. The input trajectory is brought into correspondence (green dashed lines) with the reference using OE-DTW. Then the input is warped to the timing of the reference. The new warped trajectory (green box) corresponds to the first 60 percent of the reference trajectory. Thus, the classifier that is responsible for the first 0 % to 60 % of the reference is selected (orange) and performs the classification.

contribute to an error is a time-consuming task. Consequently, we aim at extracting a visual highlight mask that provides temporal as well as spatial information using feature importance from our classification pipeline.

Our pipeline can easily be used to generate feedback, as the specification of the hyperplane of the linear SVM can be interpreted as importance values for each feature at each time step. The separating hyperplane is expressed by $\mathbf{w}^T\mathbf{x}+b=0$, where $\mathbf{x}$ is the input. The components of $\mathbf{w}$ can be interpreted as importance values assigned to each feature. Based on this information, a visual highlight mask for each error pattern is calculated offline after training. It can then be applied inside the coaching application as soon as an error is detected.

First, joint importance is determined in two steps. The first one performs denoising for each joint. If a joint is considered important at a specific time step, but the temporal neighborhood is considered not important, the importance value is set to zero. Afterwards, for each joint, its importance values are summed-up over time leading to joint weights $\omega_j(k)$. Next, we calculate the final highlight mask and, as this mask can be precomputed, we smooth it to obtain better looking highlights. We set the values for all joints to zero whose joint importance $\omega_j(k)$ is smaller than 20 % of the largest value in $\omega_j(k)$. Then, for each frame, we sum-up all joint weights to obtain frame weights $\omega_f(t)$. These provide us with information on which point in time is in general important for the error pattern of interest. The frame weights are smoothed via applying two closing masks followed by an erosion mask. The final highlight mask $h(t,k)$ for each spatial feature k and each frame t (with respect to the canonical timeline) is then calculated by

$$h(t,k) = \begin{cases} 0, & \text{if } \omega_f(t)\omega_j(k)y(t) = 0 \\ 1, & \text{otherwise} \end{cases}. \qquad (2)$$

$y(t)$ is the binary label estimated by the classifier at frame t.

6. Evaluation and Comparisons

6.1. Classification

We applied a 5-fold cross validation that aims at between-subjects testing and similar proportions of positive and negative labels in the folds as compared to the overall data set. We measure classification quality in terms of accuracy and F1 scores

Table 2. Analyzed error patterns in the execution of a Tai Chi push. The numbers denote the quantity of incorrect and correct executions of the Tai Chi push in our data, with respect to the corresponding pattern.

Performance	#Erroneous	#Correct
non-uniform movement	47	21
left leg moves forward	67	52
knee too much in front	23	65
incorrect weight distribution	17	64
backmost foot incorrectly rotated	39	65
asynchronous	35	39
arms return at chest height	16	73

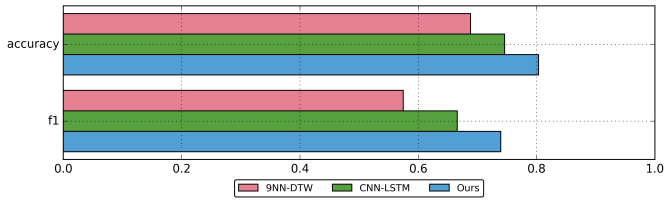


Fig. 6. Averaged scores of the classifiers on the squat data set.

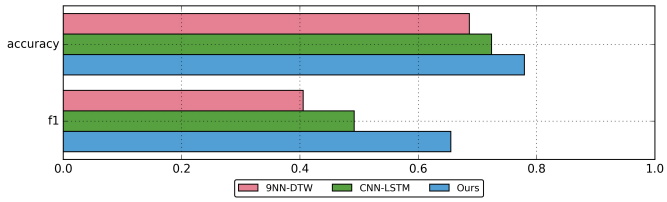


Fig. 7. Averaged scores of the classifiers on the Tai Chi push data set.

for the point in time where the classifier knows the whole input trajectory. Additionally to presenting the average classification performance per motor action and per error pattern, we check for significance on the level of error patterns. To this end, we perform a pairwise comparison of the classification success for all test trajectories by using the Wilcoxon signed-rank test with Bonferroni correction. The time measurements were conducted on a machine with Intel CPU Core i7-7700K 4.2 Ghz.

We compare our pipeline to KNN-DTW and CNN-LSTM. Further, we provide a comparison to Ad-Hoc classifiers for specific error patterns. The overall classification quality of all tested approaches is visualized in Figure 6 for the squat and Figure 7 for the Tai Chi push. For both data sets, the worst results are obtained by KNN-DTW. The best results are obtained by our own classifier. CNN-LSTM lies in between. Additionally to the summarized classification quality, we provide results for the individual error patterns. The accuracies for the single error patterns can be found in Figure 8. Concerning our pipeline, for all error patterns of the squat together, the necessary OE-DTW as well as the classification itself need 5.2 ms on average. For the Tai Chi push, we need on average 6.6 ms to perform the single OE-DTW as well as the final classification for all error patterns. The timings for the other approaches are presented in the subsequent paragraphs.

6.1.1. Comparison to Hand-crafted Ad-hoc Classifiers

Manually hand-crafted classifiers such as the ones presented in [2] are time-consuming to develop and thus violate requirement (R5), which demands few manual work. Further it is not possible to develop these hand-crafted classifiers for all types of errors. However, if they are available, they can mark a kind of ideal performance to which a data-driven classifier can be compared. For the squat, we developed hand-crafted ad-hoc classifiers similar to [2] for some of the error patterns. For the pattern "not symmetric", we defined the symmetry of a posture as the averaged quaternion distance between the rotations of the right and the mirrored rotations of the left side of the sagittal plane. To capture the trembling of the knees, we extract the lateral movement of the knees. For the other error patterns, we used manually selected joints and simple relationships between them as input. For the manually selected and preprocessed input features, we learn separating hyperplanes using a linear SVM based on the same cross validation folds as used in the experiments before. Our data-driven pipeline reaches a performance in a range similar to the results of the manually crafted classifier. However, our pipeline needs much less manual work and is not only restricted to posture-based patterns, but also takes the current point in time of an input motion into account. For the patterns "knees tremble sideways" and "not symmetric", which are not well classified by the data-driven approaches, results indicate that even manually crafted rules do not lead to better results. For the pattern "knees tremble sideways" we obtain an accuracy of 0.57 which is close to the accuracy of 0.56 obtained by our own data-driven pipeline. For the pattern "not symmetric", the manually crafted classifier obtains an accuracy of already 0.73 instead of 0.64, however the F1 score is zero.

6.1.2. Comparison to KNN-DTW

KNN-DTW is the combination of k-nearest-neighbors (KNN) as classification algorithm with Dynamic Time Warping (DTW) as distance measure. For an input query, KNN searches for the K data points that are most similar to the input. Then it returns their label, using majority vote. In order to classify a new query trajectory, KNN-DTW performs DTW with all trajectories, and then, for each error pattern of interest, returns the label of the closest trajectories that are annotated with respect to this error pattern. We use the DTW with path-length weighting as described in [57]. For KNN, we select $K = 9$, as we observed this value to lead to best results.

For the squat, 9NN-DTW leads to a classification performance of on average $accuracy = 0.69$, $f1 = 0.57$, wheras our pipeline reaches $accuracy = 0.8$, $f1 = 0.74$. Our pipeline leads to better accuracies than 9NN-DTW in eight of the ten error patterns. The differences are significant for the patterns "legs extended at end" ($p < 0.001$), "feet distance not sufficient" ($p < 0.001$), and "too deep" ($p = 0.003$). We observe a trend towards significance for the patterns "hollow back" ($p = 0.07$) and "hips do not initiate movement" ($p = 0.08$). Concerning the Tai Chi push, 9NN-DTW reaches $accuracy = 0.69$, $f1 = 0.41$ compared to $accuracy = 0.78$, $f1 = 0.65$. The accuracies of our pipeline are better in five of seven patterns. We observe significant differences between our pipeline and 9NN-DTW for the pattern "arms return at chest height" ($p = 0.03$, this is the only case, where one of the other approaches performs significantly better than our pipeline), "left leg moves forward" ($p < 0.001$), and "knee too much in front" ($p = 0.005$). We observe trends for the patterns "incorrect weight distribution" ($p = 0.08$) and "backmost foot incorrectly rotated" ($p = 0.09$). For the squat as well as for Tai Chi, 9NN-DTW needs multiple seconds to calculate all necessary DTWs for the comparison.

6.1.3. Comparison to CNN-LSTM

The combination of Convolutional Neural Networks (CNN) and a Long Short-Term Memory (LSTM), as described by Núñez et al., is especially designed for the classification of

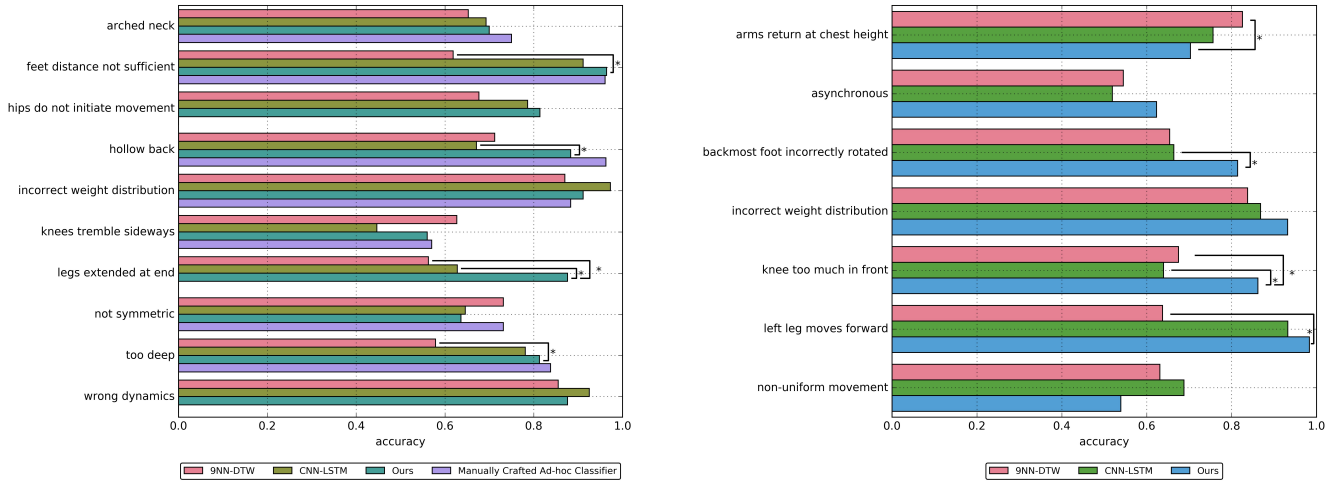


Fig. 8. Accuracies of the classifiers for the squat (left) and the Tai Chi push (right) for each error pattern. Significant differences ($p < 0.05$) are marked with a star ($*$).

human motion capture data [12]. We therefore compare to their approach and give a brief description below. Fore more analytical insights and experiments on architecture and parameters, as well as figures that visualize the architecture, we refer to the original paper [12]. Basic information on the underlying properties of CNNs and LSTMs can be found in [58].

The input movement is first processed by the CNN. The CNN learns a higher level representation of motion on the spatial as well as on the temporal domain via spatio-temporal convolution. Next, the preprocessed feature map is handled by the LSTM which covers the broader temporal context. The CNN proposed by Núñez et al. consists of six alternating Convolutional (ReLU activation; filter sizes: 20, 50, 100; kernel sizes: 3, 2, 3) and Pooling layers. The LSTM consists of 100 units. The training of the complete network, CNN-LSTM, consists of two steps. In the first step the weights of the CNN are pretrained. The CNN is not yet connected to the LSTM, but to two densly connected layers (300 units, 100 units, ReLu), followed by an output layer with sigmoid activation. Time windows are separately fed into the network together with the label of the corresponding trajectory. The network is trained for 100 epochs with a batch size of 200. Next, as suggested by Núñez et al., the dense layers are cut off and the pretrained CNN is connected to the LSTM. Complete recordings of movements that consist of a sequence of time windows are now used as input. The new network is trained for 500 epochs with a batch size of 16 using Adadelta [59]. According to [12], due to the two-stage training, higher accuracies can be achieved compared to training the final network in one step. In our implementation, we use a window size of $T = 20$ according to the experiments performed in [12] and we use a time shift of 10 that led to good results in our experiments. As input, we use joint translations as they led to better results than the combination of of translations and angles. We implemented the networks using Tensorflow [60] in version 1.6.0 and Keras [1] in version 2.1.5..

[1] `https://keras.io`

For the squat, CNN-LSTM leads to a classification performance of on average $accuracy = 0.75$, $f1 = 0.67$, wheras our pipeline reaches $accuracy = 0.8$, $f1 = 0.74$. For the squat, our pipeline leads to better accuracies than CNN-LSTM in seven of the ten error patterns. These differences are significant for the patterns "hollow back" ($p = 0.01$) and "legs extended at end" ($p < 0.001$). Concerning the Tai Chi push, CNN-LSTM reaches $accuracy = 0.72$, $f1 = 0.49$ compared to $accuracy = 0.78$, $f1 = 0.65$. The accuracies of our pipeline are better in five of seven patterns. Among them, "knee too much in front" ($p = 0.004$) and "backmost foot incorrectly rotated" ($p = 0.03$) lead to significant differences. For both motor actions, no error pattern is significantly better classified by CNN-LSTM than by our pipeline. CNN-LSTM needs approximately 8 ms for the classification of all error patterns of the squat. For the Tai Chi push, it needs around 7 ms on average.

6.1.4. Summary

In our summary, we only focus on the data-driven approaches as the rule-based ad-hoc classifiers need a high amount of manual work (R5) and as it is problematic to design them for all of the error patters. For the squat, our pipeline leads to best accuracies in six of the ten error patterns. In two cases, CNN-LSTM leads to best results, in two cases 9NN-DTW obtains best scores. Concerning the Tai Chi data set, our pipeline leads to best accuracies in five of seven patterns. One pattern is best classified by 9NN-DTW, one is best classified by CNN-LSTM. When using our own pipeline, we obtain the best averaged classification performance, followed by the CNN-LSTM, followed by KNN-DTW. All three approaches allow for the application of already existing feedback strategies linked to specific error patterns (R1). As soon as an error is detected, the corresponding feedback strategy can be triggered. The CNN-LSTM as well as our new pipeline work in real-time (R2). This is not the case for KNN-DTW, as the time needed for classification depends on the size of the training set, and is already large for one single comparison. All data-driven approaches work with small data sets (R4) and require only few manual work (R5), namely

the labeling and recording of the training data. We will focus on the evaluation of the interpretability (R3) in terms of visual augmented feedback that can be generated in the next section.

6.2. Visual Augmented Feedback

We provide a comparison of visual feedback obtained by our pipeline to visual feedback we extract from the CNN-LSTM-based approach. First, we describe how the latter can be used to generate the desired highlight masks. For neural networks, saliency maps have been established to provide information on the importance of features in the input data [36]. They are calculated via deriving the output w.r.t. the input. We use the implementation provided by keras-vis [2] in version 0.4.1. As the input data for the CNN-LSTM-based approach consists of trajectories with different lengths and different timings, we cannot pre-process a fixed visual highlight mask for each classifier, but calculate the saliences for each input. We map the saliences to highlights if the error of interest is classified for the given point in time. Preprocessing is not performed as the saliency depends on the input movement a trainee performs, and these movements are of different lengths and have different timings.

In our evaluation, we first focus on the spatial dimension, namely the joints. We examine the joint importance values exemplary for the error patterns "hollow back" and "incorrect weight distribution". For the "hollow back", a straight posture of the back is important. In our body model, the flexion of the back is specified by joint vt10. Its flexion approximates the angle between the lower part of the upper back (thoracic spine) and the upper part of the lower back (lumbar spine). The error pattern "incorrect weight distribution" occurs, if the knees and the hips move too much to the anterior. Based on [37], it is required that the knees are kept in line with the toes. Consequently, the whole lower part of the body can directly contribute to the error "incorrect weight distribution". Results for our pipeline are obtained during training and averaged over all cross validation folds. For the CNN-LSTM-based approach, we calculate the importance for all joints for each test trajectory that is correctly classified as erroneous. The resulting importance values are then averaged. For the "hollow back", Figure 9a contains the joint importances obtained by our pipeline and Figure 9b contains the results for the CNN-LSTM. The results for the pattern "incorrect weight distribution" can be found in Figure 10. For both patterns, there are joints that are similarly pointed out as important by both approaches, however, the results for the CNN-LSTM are less clear and tend to highlight joints that are not important for the given error pattern. Concerning the "incorrect weight distribution", the joints which obtained high values by our pipeline are mostly in the lower parts of the body which is in line with the theoretical information on the error patterns. These joints are mostly also selected by the CNN-LSTM, however, here, also parts of the upper body, such as sternoclavicular and shoulder are considered as important. This is problematic in terms of feedback, as the posture of the upper body w.r.t. the error pattern depends on the subject's proportions. A coach might want the subject to focus on the lower part and to automatically move the upper part in a suitable way to maintain a stable stand. Concerning the "hollow back", our pipeline selects exactly the joint that is important for the error pattern from a theoretical point of view, namely vt10. Concerning CNN-LSTM, also other less important joints, such as many joints in the lower body, obtain high values. To summarize, the joints selected by our pipeline are clearer and more suitable for visualization. Further, the joints selected by CNN-LSTM depend on the just performed movement, so selected features could vary for different inputs.

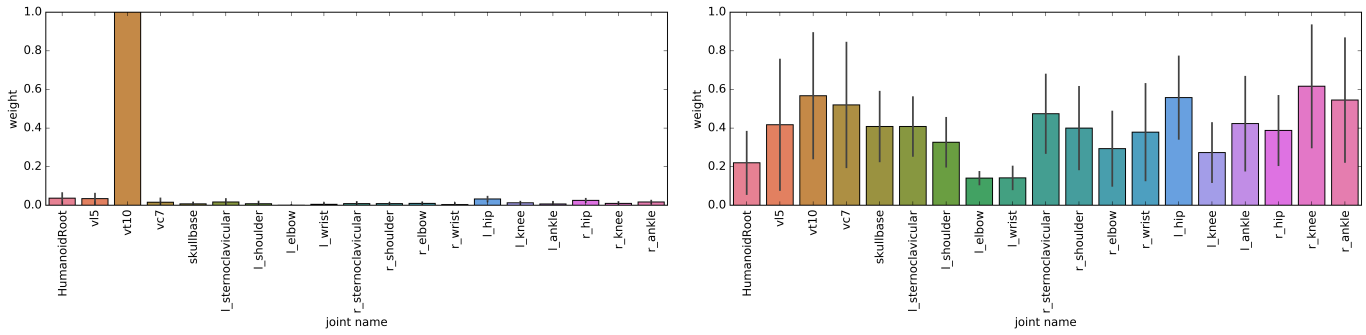
Next, we compare the overall quality of augmented feedback generated by both approaches. See Figure 11 and Figure 12 for exemplary visual highlights generated by both approaches. For the "hollow back" (cf. Figure 11a, 11b), the CNN-LSTM-based approach selects not only the back for the given example, but also joints in the lower part of the body, whereas our pipeline selects exactly the most relevant features, namely the back. Concerning the "incorrect weight distribution" (cf. Figure 11c, 11d), the features selected by CNN-LSTM (left leg) for the input are a subset of the important features, however, other relevant joints, such as the right leg as well as the hips, are not selected. In contrast, our pipeline provides a much clearer highlight of the important joints. Concerning the error pattern "too deep" (cf. Figure 12), our comparison demonstrates, that even if the joints selected by CNN-LSTM are reasonable, the timing of the feedback can be problematic. Here, the highlight for the given trajectory is shortly activated already at the beginning of the movement, thus at a point in time that does not have a direct impact on the depth. In contrast, the highlights extracted from our pipeline are visible exactly when the subject is approaching the deepest point of the movement.

In summary, the highlights generated by our pipeline are more meaningful compared to the ones generated by CNN-LSTM. Additionally, the complete highlight mask can be precomputed and, if desired, manually checked for obvious errors (e.g., an activation of highlights for error patterns such as "hollow back" that occur at a point in time that is not sufficiently related to the error itself). When relying on the CNN-LSTM, highlight masks for single performances can work sufficiently well, whereas the highlight for other movements is problematic. Consequently, our pipeline better satisfies requirement (R3), the interpretability of the classifier. See the video in the supplementary material for exemplary visualizations of automatically generated augmented feedback.

7. Exemplary Coaching Session

In the following, we describe an exemplary coaching session for the squat to demonstrate the abilities of our pipeline. The trainee is placed in a two-sided CAVE (L-shape, dimensions: 3 m × 2.3 m for each side). Each wall is operated by two projectors which run at a resolution of 2100 × 1600 pixels. For each wall, we use one NVIDIA Quadro P6000 graphics card. The images for both eyes are separated using passive filters by INFITEC. The rendering engine is self-developed and runs our scene on one single computer at around 250 fps. The whole environment has been designed for an unobtrusive and natural

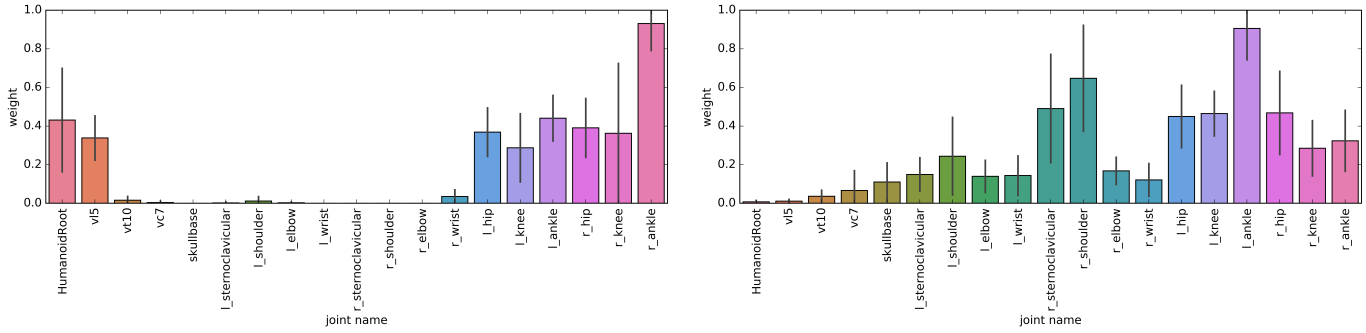
[2] https://raghakot.github.io/keras-vis/



(a) Selection by our pipeline.

(b) Selection by the CNN-LSTM-based approach.

Fig. 9. Comparison of selected joints for the error pattern "hollow back".



(a) Selection by our pipeline.

(b) Selection by the CNN-LSTM-based approach.

Fig. 10. Comparison of selected joints for the error pattern "incorrect weight distribution".

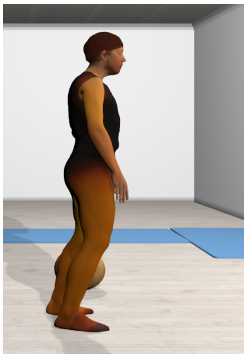
(a) "hollow back" (CNN-LSTM): Additional to the back, the legs are undesirably highlighted.

(b) 'hollow back" (ours): The crucial part (the back) is highlighted.

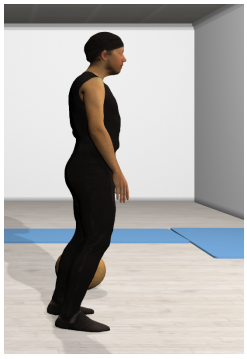
(c) "incorrect weight distribution" (CNN-LSTM): Only parts (left leg) of the relevant joints are highlighted.

(d) "incorrect weight distribution" (ours): The relevant parts in the lower body are highlighted.

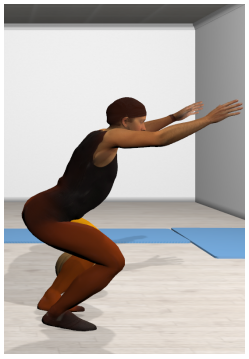
Fig. 11. Comparison of feedback generated by CNN-LSTM (a, c) and our pipeline (b,d). In (a, b) feedback for the error "hollow back" is visualized, in (c,d) the feedback for the error "incorrect weight distribution" is shown. If multiple error patterns occur at the same time, highlights are only shown for one of them.



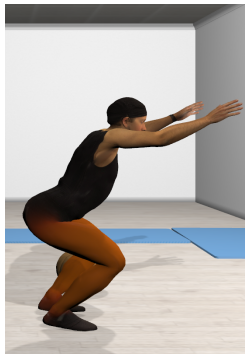
(a) CNN-LSTM: Highlights are already visualized for a short period of time at the beginning of the movement which is undesired.



(b) Ours: As the point in time is not relevant for the error pattern, no highlights are shown.



(c) CNN-LSTM: Highlights are correctly enabled (highlights on the left side of the body are slightly brighter than on the right side).



(d) Ours: The highlights are correctly enabled.

Fig. 12. Pattern "too deep": Comparison of feedback generated by CNN-LSTM and our pipeline at two different time steps. The beginning of the movement (a, b) is not relevant for the error pattern and should thus contain no highlights. The other time step (c,d) is relevant and should thus contain highlights.

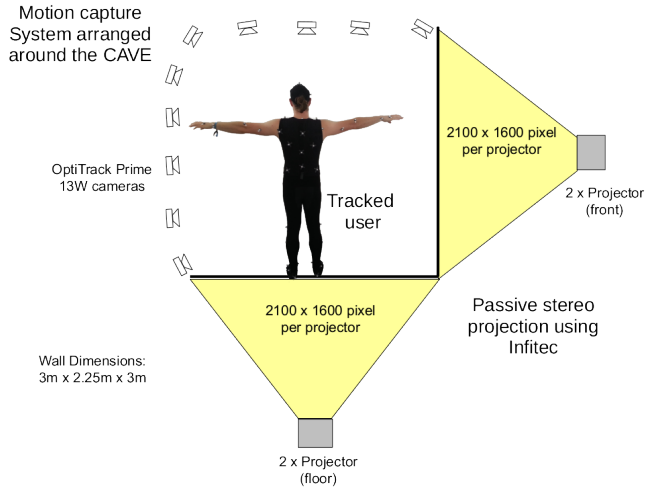


Fig. 13. Schematic overview of the VR setup.

interaction [**?**], including lightweight passive glasses and no heavy or cabled equipment attached to the trainee. We refer to [**?**] for more information on our VR environment. See Figure 13 for a schematic overview of the VR setup.

The virtual environment shows a gym which contains a virtual mirror in front of the user. The user's motion is mapped on an avatar, a 3D scan of the user, inside the virtual mirror [**?**]. The advantage of such personalized avatars has been demonstrated in [**?**]. Next to the mirror a virtual coach observes the user's motion and provides feedback (see Figure 1). The beginning and the end of each motor action are automatically detected similar to [2]. Errors in the user's performance are classified using our pipeline. Based on this information, the coach addresses the patterns "incorrect weight distribution", "feet distance not sufficient" and "hollow back". To this end, the coach uses verbal feedback as well as augmented feedback inside the virtual mirror. The augmented feedback includes a rotation of the perspective inside the virtual mirror, color highlights mapped on the virtual character (see Figure 14a), and a replay of the last squat performed by the user to observe the occurred error (see Figure 14b). The accompanying video shows the exemplary coaching session in our virtual environment. While the user performs squats, classification results are plotted as an overlay inside the video. The error patterns the coach addresses are highlighted.

8. Discussion and Conclusion

The focus of this work is on the assessment of motion performed by a trainee in a sports coaching environment in VR, using the squat and the Tai Chi push as test case. We had a special focus on the combination of error detection with the automatic generation of augmented feedback. To this end, we carved out proper requirements. We introduced a new pipeline that satisfies these requirements and consists of two main parts: The classification of motor errors and the automatic generation of augmented feedback. For evaluation, we use two motor tasks, namely the squat and the Tai Chi push. We demonstrate that our pipeline is able to beat KNN-DTW as well as a recent neural network-based approach [12] in terms of classification performance and generated augmented visual feedback. Our pipeline has been specifically designed to treat the special properties of motion data in order to classify typical errors in real-time. Consequently, known properties of the problem, such as the temporal warping or the feature selection, are covered by the architecture of the pipeline. The neural network-based approach needs to learn most of these properties from the training data which could explain the superior performance of our pipeline. The squat and Tai Chi data sets used in this publication are publicly available via the DOI: *10.4119/unibi/2930611*.

Even though general classification performance of our pipeline is high, the performance is not convincing specifically for two error patterns for the squat and one for the Tai Chi push. The pattern "arms return at chest height" is classified with a very low F1 score ($f1 = 0.1$). A possible reason could be the

(a) The hollow back is highlighted on the avatar inside the virtual mirror. The perspective of the mirror image is rotated to enable the user to observe his errors without the need to change his body's orientation.

(b) The user's last squat is replayed in the virtual mirror. The perspective of the mirror is rotated to enable the user to observe how the knees move in front of his toes which is one of the indications for the error pattern "incorrect weight distribution".

Fig. 14. Desktop rendering of feedback mechanisms that can be applied by the virtual coach.

immense imbalance between positive (16) and negative (73) examples in combination with the fairly complex error pattern. Concerning the squat, the error pattern "not symmetric" is detected with F1 scores only slightly above 0.4. This error pattern is annotated in trajectories where some joints are not symmetric between the left and the right side of the body. As this can occur in almost all joints and all phases of the movement, the feature selection cannot easily spot those features of interest that are relevant. For the other problematic pattern, "knees tremble sideways", our results look similar. This pattern describes a very subtle movement. Also, it can spread temporarily: Exactly the frames that are problematic for subject A can be correct for subject B and vice versa. Finally, the number of trembles can be different for different subjects which also makes classification harder. Focusing on such patterns that are hard to classify, is a reasonable direction of future work, as here even a hand-crafted ad-hoc classifier was unable to obtain good classification results. One possible solution could be a combination of more complex higher-level features within our pipeline. Concerning the generated augmented feedback, note that the feedback we generate can only work if the classifier itself performs well. For error patterns such as "not symmetric" or "knees tremble sideways", the classifier is unreliable, thus also the selected features have no explanatory power.

A limitation of our pipeline is that temporal properties of the movements are not covered directly. However, for motor actions where the user's timing has an influence on whether certain errors occur, temporal information could be included via adding velocity as well as information on the warping function extracted from DTW. The list of error patterns and the annotated training data for the Tai Chi movement is only based on information from a single, albeit experienced coach and on literature. Taking into account information from more experts could further improve the developed model. Another interesting focus of future work could be the application of our pipeline to further challenging motor actions, such as dancing or martial arts. As we specifically designed our pipeline with a focus on dealing with error classification in sports movements, we would assume similar results due to the general properties of the data. To enhance the overall performance of the classifier, one direction of future work could be improvements in the single components of the pipeline, for instance concerning extensions of DTW as well as an evaluation of further approaches towards feature selection such as the ones described in [62]. Concerning the augmented feedback, we even do not always need classification in order to provide feedback. In cases where just the attention of the trainee needs to be guided to the crucial parts of the movement with respect to a certain error pattern, we only need the first part of the pipeline, the temporal warping. Then, we could highlight the important joints based on Equation 2. One aspect of future work is to further investigate when to provide which amount of augmented feedback. Another direction of future research is motivated by the usability of the virtual environment. To attach motion capture markers to subjects is time-consuming. Recent approaches from pattern recognition and computer vision are able to extract the human posture from video images. Focusing on accurate marker-less and low-latency motion capture algorithms is promising to advance the field of sports coaching in virtual environments.

Acknowledgments

We would like to thank Lina Varonina, Yvonne Ritter, Irene Senna and Cornelia Frank for support in data acquisition. We would also like to thank Iwan de Kok for making available the videos from coaching sessions described in [8]. This work was supported by the Cluster of Excellence Cognitive Interaction Technology "CITEC" (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG).

References

[1] Kyan, M, Sun, G, Li, H, Zhong, L, Muneesawang, P, Dong, N, et al. An approach to ballet dance training through MS kinect and visualization in a cave virtual reality environment. ACM Transactions on Intelligent Systems and Technology (TIST) 2015;6(2):23.

[2] de Kok, I, Hough, J, Hülsmann, F, Botsch, M, Schlangen, D, Kopp, S. A multimodal system for real-time action instruction in motor skill learning. In: Proceedings of the International Conference on Multimodal Interaction. ACM; 2015, p. 355–362.

[3] Sigrist, R, Rauter, G, Marchal-Crespo, L, Riener, R, Wolf, P. Sonification and haptic feedback in addition to visual feedback enhances complex motor task learning. Experimental brain research 2015;233(3):909–925.

[4] Chan, JC, Leung, H, Tang, JK, Komura, T. A virtual reality dance training system using motion capture technology. IEEE Transactions on Learning Technologies 2011;4(2):187–195.

[5] Miles, HC, Pop, SR, Watt, SJ, Lawrence, GP, John, NW. A review of virtual environments for training in ball sports. Computers & Graphics 2012;36(6):714–726.

[6] Schack, T, Bertollo, M, Koester, D, Maycock, J, Essig, K. Technological advancements in sport psychology. Routledge; 2014, p. 953–965.

[7] Hough, J, de Kok, I, Schlangen, D, Kopp, S. Timing and grounding in motor skill coaching interaction: Consequences for the information state. In: Proceedings of the 19th SemDial Workshop on the Semantics and Pragmatics of Dialogue (goDIAL). 2015, p. 86–94.

[8] de Kok, I, Hough, J, Frank, C, Schlangen, D, Kopp, S. Dialogue structure of coaching sessions. In: Proceedings of the SemDial Workshop on the Semantics and Pragmatics of Dialogue (DialWatt). 2014, p. 167–169.

[9] de Kok, I, Hough, J, Schlangen, D, Kopp, S. Deictic gestures in coaching interactions. In: Proceedings of the Workshop on Multimodal Analyses enabling Artificial Agents in Human-Machine Interaction. ACM; 2016, p. 10–14.

[10] Xi, X, Keogh, E, Shelton, C, Wei, L, Ratanamahatana, CA. Fast time series classification using numerosity reduction. In: Proceedings of the 23rd international conference on Machine learning. ACM; 2006, p. 1033–1040.

[11] Bagnall, A, Lines, J. An experimental evaluation of nearest neighbour time series classification. arXiv preprint arXiv:14064757 2014;.

[12] Núñez, JC, Cabido, R, Pantrigo, JJ, Montemayor, AS, Vélez, JF. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. Pattern Recognition 2018;76:80–94.

[13] Houmanfar, R, Karg, M, Kulić, D. Movement analysis of rehabilitation exercises: Distance metrics for measuring patient progress. IEEE Systems Journal 2016;10(3):1014–1025.

[14] Rector, K, Bennett, CL, Kientz, JA. Eyes-free yoga: an exergame using depth cameras for blind & low vision exercise. In: Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility. 2013, p. 12–19.

[15] Taylor, PE, Almeida, GJ, Kanade, T, Hodgins, JK. Classifying human motion quality for knee osteoarthritis using accelerometers. In: Annual International Conference of the IEEE Engineering in Medicine and Biology. 2010, p. 339–343.

[16] Yurtman, A, Barshan, B. Automated evaluation of physical therapy exercises using multi-template dynamic time warping on wearable sensor signals. Computer methods and programs in biomedicine 2014;117(2):189–207.

[17] Parisi, GI, Magg, S, Wermter, S. Human motion assessment in real time using recurrent self-organization. In: Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium on. IEEE; 2016, p. 71–76.

[18] O'Reilly, M, Whelan, D, Chanialidis, C, Friel, N, Delahunt, E, Ward, T, et al. Evaluating squat performance with a single inertial measurement unit. In: International Conference on Wearable and Implantable Body Sensor Networks (BSN). IEEE; 2015, p. 1–6.

[19] Giggins, O, Kelly, D, Caulfield, B. Evaluating rehabilitation exercise performance using a single inertial measurement unit. In: Proceedings of the International Conference on Pervasive Computing Technologies for Healthcare. 2013, p. 49–56.

[20] Giggins, OM, Sweeney, KT, Caulfield, B. Rehabilitation exercise assessment using inertial sensors: a cross-sectional analytical study. Journal of Neuroengineering and Rehabilitation 2014;11(1):1.

[21] Kianifar, R, Lee, A, Raina, S, Kulić, D. Classification of squat quality with inertial measurement units in the single leg squat mobility test. In: Engineering in Medicine and Biology Society (EMBC), Annual International Conference of the. IEEE; 2016, p. 6273–6276.

[22] Zou, H, Hastie, T, Tibshirani, R. Sparse principal component analysis. Journal of computational and graphical statistics 2006;15(2):265–286.

[23] Hossner, EJ, Schiebl, F, Göhner, U. A functional approach to movement analysis and error identification in sports and physical education. Frontiers in Psychology 2015;6:1339.

[24] Sigrist, R, Rauter, G, Riener, R, Wolf, P. Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review. Psychonomic bulletin & review 2013;20(1):21–53.

[25] Liu, D, Todorov, E. Evidence for the flexible sensorimotor strategies predicted by optimal feedback control. The Journal of Neuroscience 2007;27(35):9354–9368.

[26] Wilson, AD, Bobick, AF. Parametric hidden markov models for gesture recognition. IEEE transactions on pattern analysis and machine intelligence 1999;21(9):884–900.

[27] Rodríguez, JJ, Alonso, CJ. Interval and dynamic time warping-based decision trees. In: Proceedings of the ACM symposium on Applied computing. 2004, p. 548–552.

[28] Wu, Y, Chang, EY. Distance-function design and fusion for sequence data. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management. 2004, p. 324–333.

[29] Nanopoulos, A, Alcock, R, Manolopoulos, Y. Feature-based classification of time-series data. International Journal of Computer Research 2001;10(3):49–61.

[30] Adistambha, K, Ritz, CH, Burnett, IS. Motion classification using dynamic time warping. In: Multimedia Signal Processing, IEEE Workshop on. 2008, p. 622–627.

[31] Petitjean, F, Forestier, G, Webb, GI, Nicholson, AE, Chen, Y, Keogh, E. Dynamic time warping averaging of time series allows faster and more accurate classification. In: International Conference on Data Mining. IEEE; 2014, p. 470–479.

[32] Li, W, Wen, L, Chang, MC, Lim, SN, Lyu, S. Adaptive rnn tree for large-scale human action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017, p. 1444–1452.

[33] Liu, J, Shahroudy, A, Xu, D, Chichung, AK, Wang, G. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. IEEE Transactions on Pattern Analysis and Machine Intelligence 2017;.

[34] Liu, J, Li, Y, Song, S, Xing, J, Lan, C, Zeng, W. Multi-modality multi-task recurrent neural network for online action detection. IEEE Transactions on Circuits and Systems for Video Technology 2018;.

[35] Liu, J, Wang, G, Hu, P, Duan, LY, Kot, AC. Global context-aware attention lstm networks for 3d action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017, p. 1647–1656.

[36] Simonyan, K, Vedaldi, A, Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:13126034 2013;.

[37] Clark, MA, Lucett, S, Sutton, BG. NASM essentials of personal fitness training. Lippincott Williams & Wilkins; 2008.

[38] Chan, S, Luk, T, Hong, Y. Kinematic and electromyographic analysis of the push movement in tai chi. British Journal of Sports Medicine 2003;37(4):339–344.

[39] Wei, SE, Ramakrishna, V, Kanade, T, Sheikh, Y. Convolutional pose machines. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, p. 4724–4732.

[40] Cao, Z, Simon, T, Wei, SE, Sheikh, Y. Realtime multi-person 2d pose estimation using part affinity fields. In: CVPR. 2017,.

[41] Mehta, D, Rhodin, H, Casas, D, Fua, P, Sotnychenko, O, Xu, W, et al. Monocular 3d human pose estimation in the wild using improved cnn supervision. In: International Conference on 3D Vision (3DV). IEEE; 2017, p. 506–516.

[42] Mehta, D, Sridhar, S, Sotnychenko, O, Rhodin, H, Shafiei, M, Seidel, HP, et al. Vnect: Real-time 3d human pose estimation with a single rgb camera. ACM Transactions on Graphics (TOG) 2017;36(4):44.

[43] Fernández-Delgado, M, Cernadas, E, Barro, S, Amorim, D. Do we need hundreds of classifiers to solve real world classification problems. Journal of Machine Learning Research 2014;15(1):3133–3181.

[44] Bishop, CM. Pattern Recognition and Machine Learning. Springer New York, Inc.; 2006. ISBN 0-387-31073-8.

[45] Müller, M. Information retrieval for music and motion. Springer; 2007.

[46] Weston, J, Mukherjee, S, Chapelle, O, Pontil, M, Poggio, T, Vapnik, V. Feature selection for SVMs. Advances in Neural Information Processing Systems (NIPS) 2000;:668–674.

[47] Chen, YW, Lin, CJ. Combining svms with various feature selection strategies. In: Guyon, I, Nikravesh, M, Gunn, S, Zadeh, LA, editors. Feature Extraction: Foundations and Applications; vol. 207; chap. 12. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-540-35488-8; 2006, p. 315–324.

[48] Guyon, I, Elisseeff, A. An introduction to variable and feature selection. Journal of machine learning research 2003;3(Mar):1157–1182.
[49] Genuer, R, Poggi, JM, Tuleau-Malot, C. Variable selection using random forests. Pattern Recognition Letters 2010;31(14):2225–2236.
[50] Svetnik, V, Liaw, A, Tong, C, Wang, T. Application of breiman's random forest to modeling structure-activity relationships of pharmaceutical molecules. In: International Workshop on Multiple Classifier Systems. Springer; 2004, p. 334–343.
[51] Gregorutti, B, Michel, B, Saint-Pierre, P. Correlation and variable importance in random forests. Statistics and Computing 2017;27(3):659–678.
[52] Breiman, L. Random forests. Machine learning 2001;45(1):5–32.
[53] Biau, G. Analysis of a random forests model. Journal of Machine Learning Research 2012;13(Apr):1063–1095.
[54] Breiman, L, Friedman, J, Stone, CJ, Olshen, RA. Classification and regression trees. CRC press; 1984.
[55] Bi, J, Bennett, K, Embrechts, M, Breneman, C, Song, M. Dimensionality reduction via sparse support vector machines. Journal of Machine Learning Research 2003;3(Mar):1229–1243.
[56] Pedregosa, F, Varoquaux, G, Gramfort, A, Michel, V, Thirion, B, Grisel, O, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 2011;12:2825–2830.
[57] Hülsmann, F, Richter, A, Kopp, S, Botsch, M. Accurate online alignment of human motor performances. In: Proceedings of ACM Motion in Games. ACM; 2017, p. pp. 7:1–7:6.
[58] Goodfellow, I, Bengio, Y, Courville, A, Bengio, Y. Deep learning; vol. 1. MIT press Cambridge; 2016.
[59] Zeiler, MD. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:12125701 2012;.
[60] Abadi, M, Agarwal, A, Barham, P, Brevdo, E, Chen, Z, Citro, C, et al. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. Software available from tensorflow.org.
[61] Anonymous, . Anonymous title. In: Anonymous Conference. 2015,.
[62] Li, J, Cheng, K, Wang, S, Morstatter, F, Trevino, RP, Tang, J, et al. Feature selection: A data perspective. ACM Computing Surveys (CSUR) 2017;50(6):94.