A-MAL: Automatic Motion Assessment Learning from Properly Performed Motions in 3D Skeleton Videos

Tal Hakim and Ilan Shimshoni
The University of Haifa, Haifa, Israel
thakim.research@gmail.com, ishimshoni@mis.haifa.ac.il

Abstract

Assessment of motion quality has recently gained high demand in a variety of domains. The ability to automatically assess subject motion in videos that were captured by cheap devices, such as Kinect cameras, is essential for monitoring clinical rehabilitation processes, for improving motor skills and for motion learning tasks. The need to pay attention to low-level details while accurately tracking the motion stages, makes this task very challenging. In this work, we introduce A-MAL [1]*, an automatic, strong motion assessment learning algorithm that only learns from properly-performed motion videos without further annotations, powered by a deviation time-segmentation algorithm, a parameter relevance detection algorithm, a novel time-warping algorithm that is based on automatic detection of common temporal points-of-interest and a textual-feedback generation mechanism. We demonstrate our method on motions from the Fugl-Meyer Assessment (FMA) test, which is typically held by occupational therapists in order to monitor patients' recovery processes after strokes.*

1. Introduction

The capability of automatic motion assessment has recently gained high demand. One of the most significant reasons is the availability of low-cost 3D skeleton recognition devices, such as Kinect, which redefine the target audience of applications that are based on user pose and movement, including applications for motion assessment or motion learning, as well as other tasks, such as surveillance, entertainment and exercise. Addressing this problem is considered a hard task, especially when compared to the other researched tasks in the 3D skeleton-video domain, which usually give weaker attention to timings, performances and low-level measurements.

The task of automatic motion assessment is very essential for medical needs. One outstanding example is the Fugl-Meyer Assessment (FMA) [10] test, which is typically held by occupational therapists and has a numerical score. The test is performed from time to time, mainly on people who are recovering after strokes and aims to monitor their recovery processes. Except for the purpose of enabling FMA tests to be held at home, without the presence of an occupational therapist, the automation of this test has the major benefit of increased reliability, by discarding the error caused by the fact that a patient is sometimes assessed by different people at different times.

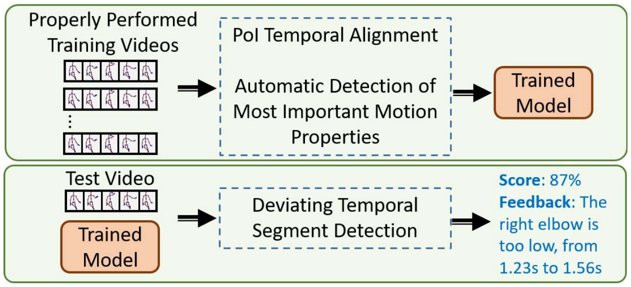

Properly Performed Training Videos
PoI Temporal Alignment
Automatic Detection of Most Important Motion Properties
Trained Model
Test Video
Trained Model
Deviating Temporal Segment Detection
Score: 87%
Feedback: The right elbow is too low, from 1.23s to 1.56s

Figure 1: The main principles of the proposed system. A model is learned solely from properly performed motions and without further annotations, by automatic extraction of motion properties, temporal alignment and learning of a frame-level statistical model. The model is used to produce score and feedback by detecting deviating temporal segments.

In this work, we aim to provide a solution for the general motion assessment task. Given a training set of 3D skeleton videos of a properly-performed motion, performed by different people, our goal is to build a statistical model that will produce feedback and a score and will deal with the challenges of the assessment task, such as identifying and paying attention to the most important low-level features and tracking the motion stages. Our resulting model produces numerical scores that accurately discriminate between the possible scores of the FMA assessment, as demonstrated in Section 5. The main principles of the proposed system are illustrated in Figure 1.

We eliminate the joint location differences related to the

[1] Code available at: http://github.com/skvp-owner/a-mal

camera location and the subject skeleton dimensions and then time-align the training videos, by automatically detecting mutual temporal points-of-interest (PoIs) and forcing them to occur at the same time in all the videos, which makes frame-level statistics more effective. The FMA motions in Figure 8, which have structured rests that can have different lengths, especially when assessing improperly performed motions, support the idea of using temporal PoIs for time-alignment, rather than existing off-the-shelf algorithms, such as Dynamic Time Warping (DTW) [2], which gives an equal weight to all the frames. In assessment mode, we time-segment the frame-level deviations, to eliminate false values and produce effective feedback, based on periods in the motion duration that clearly deviate.

The main contributions of this work are as follows. First, we present a novel step-by-step algorithm for training a model that produces scores and feedback, by only learning from a few videos of a properly-performed motion, which were captured using a relatively cheap and noisy Kinect2 device. Second, as a part of the flow, we present a novel time-warping algorithm, which is based on detection of mutual temporal PoIs, which in contrast to DTW, produces a continuous and more accurate alignment, by exploiting the rests that exist in the structure of FMA motions. Third, we present a deviation time-segmentation algorithm that deals with noisy deviations and produces effective feedback. We demonstrate the robustness of our model by showing its capability of accurately discriminating between the possible FMA performance categories using its produced numerical scores and by showing its accurate and effective produced feedback. In addition to the scientific contribution, we developed an open-source Python framework for processing skeleton videos, with a new human-readable 3D skeleton video format and a designated 3D player, which we hope will help facilitate the future research in this domain.

2. Related Work

Skeleton Tasks. While there are only a few works that try to provide solutions for the problem of motion assessment from skeleton data, the problems of motion recognition and person identification have been widely researched in recent years, especially since affordable devices such as Kinect became available. While solutions for the motion assessment problem can be adapted to perform motion recognition, solutions that were designed solely for the motion recognition problem use aggregative high level features [22, 3, 17, 11] and pay lower attention to timings, which is not satisfactory for the assessment task. However, while similarly to motion assessment, motion recognition solutions try to get rid of motion properties that are person specific, solutions for of person identification [18, 21] try to exploit them to discriminate between people.

Skeleton Representations. Han et al. [14] divide the skeleton representations in their review into four main types of representations: Displacement-based representations, which can be pairwise or temporal [31, 35, 5, 36, 39, 19, 32, 15], orientation-based representations, which can be pairwise or temporal [12, 28, 34, 4, 40], representations based on raw-joint positions, which may only apply geometric fixes and dimensionality reduction methods [26, 16, 33, 13, 41, 7] and multi-modal representations, which combine any of the three types of representations [37, 20, 38]. In our work, we combine displacement-based and orientation-based features for alignment of videos in time, while combining all the three types of features for computing scores and feedback.

Motion Assessment. The problem of motion assessment from skeletal data has been researched over the last years. Some systems and algorithms were proposed for both numerical assessment and feedback generation. Paiement et al. [23] demonstrated assessment of gait-on-stairs motions, proposing an online assessment algorithm that is based on Hidden Markov Models (HMM) for dynamics learning and assessment and on an independent statistical model for pose learning and assessment, after reducing dimensionality and filtering noise using Diffusion Maps [6]. They used normal occurrences of a motion type in order to train a designated statistical model. Su [27] demonstrated assessment of shoulder rehabilitation exercises, describing a personal assistant system that used DTW to match video frame indices to each-other and a fuzzy logic approach to produce a score for each joint. Parisi et al. [25] demonstrated assessment of power lifting motions, introducing the MGWR recurrent neural network, an architecture that is trained to predict the next frame, and presented an algorithm for real time assessment and feedback generation. Eichler et al. [8, 9] demonstrated high performance in classification of two FMA motions into score categories, combining multiple depth cameras and proposing an algorithm for calibration and improvement of skeleton location predictions. Palma et al. [24] conducted experiments, trying to find out the effectiveness of the DTW and HMM solutions for detecting deviations from normality, during the performance of physical therapy activities. None of the existing works used deep learning, mainly because they all had to deal with very small datasets.

Temporal Alignment. The DTW algorithm [27], finds the optimal monotonic increasing index matching. Its main limitation is the fact that it only produces discrete matching between pairs of indices. HMMs [23], on the other hand, use sequential training samples to learn the state transition and observation yielding probabilities. Using Viterbi's algorithm [30], HMMs predict the most likely sequence of

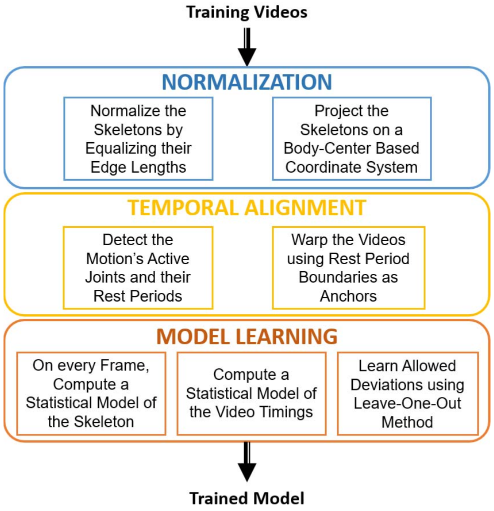

Training Videos
NORMALIZATION
Normalize the Skeletons by Equalizing their Edge Lengths
Project the Skeletons on a Body-Center Based Coordinate System
TEMPORAL ALIGNMENT
Detect the Motion's Active Joints and their Rest Periods
Warp the Videos using Rest Period Boundaries as Anchors
MODEL LEARNING
On every Frame, Compute a Statistical Model of the Skeleton
Compute a Statistical Model of the Video Timings
Learn Allowed Deviations using Leave-One-Out Method
Trained Model

Figure 2: The main model learning flow in high level, which can be seen as 3-incremental steps.

states that would yield a sequence of observations, under the Markovian assumption. Its main disadvantage is the fact that it requires the training data to be annotated with the motion stages. The Correlation Optimized Warping (COW) [29] algorithm finds the optimal time-scaling coefficients of uniformly-divided sequence segments. Its main limitation is the fact that its scales divisions that are based on timings, rather than on detected events.

3. Training Method

The method we suggest here, is based on taking a set of videos of different people properly performing the same motion and extracting frame-level features. As illustrated in Figure 2, we normalize all the skeletons and project them on a uniform coordinate system. We then temporally align the videos, such that each motion stage will occur at the same time in all the videos. Having these conditions satisfied, we extract frame-level features and learn a model. In Section 4, we describe the analysis process that is applied on test videos.

Input Format and Mathematical Notation. We assume that the input to our algorithm is a list of n skeleton videos, where a skeleton is defined by a set of joints J and the edges connecting them $e = (j_1, j_2) \in E, j_1, j_2 \in J$. The location (x, y, z) of each joint $j \in J$ is known at each frame $f \in F$ in each video. Let us denote F_i as the list of frames in video i, and let us denote L_{jfi} as the location of joint j in frame $f \in F_i$.

When we use the notation $\widehat{x}$, we refer to the normalized version of vector $\overrightarrow{x}$.

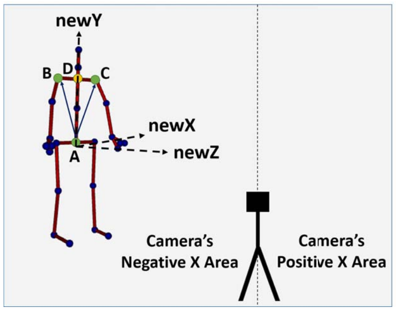


Figure 3: Joints A, B, C, D are Spine-Base, Shoulder-Left, Shoulder-Right and the average of B and C, respectively.

3.1. Video Normalization

The video normalization step is essential for making the input videos comparable, by removing unneeded, distracting information from the data.

Skeleton Dimension Normalization. We eliminate the dimensional body differences between the people in the training set, by forcing equal skeleton edge lengths in all the frames, in all the videos. For each performing person i, we compute the average length $\|e_i\|$ of each edge $e = (p, q) \in E$, over all the video frames, to get a reliable measure of its length. We then for each edge $e = (p, q) \in E$, compute its average length $\|e\|$ over all the performing people, to get the designated skeleton edge lengths. We then scale the edges in all the frames in all the videos, so their lengths will be equal to the designated skeleton. We use the Breadth-First Search (BFS) strategy to scan and alter the skeleton joints hierarchically. We start from an arbitrary joint p and alter the joint locations according to their BFS order that starts from p. We set the new location of every neighboring joint q, by adding the vector $\overrightarrow{pq}$ to the new location of p, after scaling the vector to have the same magnitude as in the designated skeleton.

Alignment to Body-Plane Coordinate System. We project all the joint 3D locations on a coordinate system that is based on a body-center plane that we compute at each frame. We use the locations of the Spine-Base, Shoulder-Left and Shoulder-Right joints, which we mark as A, B, C, respectively, as visualized in Figure 3. Let us define the point $D = \frac{1}{2}(B + C)$, as the average of the two shoulders B, C. The vector $\hat{Y}$ will be the normalized vector $\overrightarrow{AD}$. The vector $\hat{Z}$ will be the normalized vector $\overrightarrow{AC} \times \overrightarrow{AB}$. The vector $\hat{X}$, will be the normalized vector $\hat{Y} \times \hat{Z}$. We project the joints on the system by multiplying them by a matrix that consists of $\hat{X}, \hat{Y}$ and $\hat{Z}$ as row vectors. We finally subtract the Spine-Base joint location from all the joints, to translate the skeleton to the origin.

3.2. Video Temporal Alignment

We detect mutual temporal PoIs in the training videos and force them to occur at the same time in all the videos, in order to have more meaningful information at each frame. We do that by temporally scaling the sequences between them, such that they will have an equal length in all the videos. Our scaling is done using linear interpolations of joint locations.

Active Joint Detection. We want to discriminate between the active and inactive joints of the motion in order to detect the rest sequences of the active joints and use their temporal boundaries as temporal PoIs. Let us introduce $\overline{L}_{ji}$, as the mean of the joint locations L_{jfi} of joint j in video i, over all frames $f \in F_i$. We use it to compute the variance $\widehat{\sigma}^2(L_{ji})$:

$$\widehat{\sigma}^2(L_{ji}) = \frac{1}{\|F_i\| - 1} \sum_{f=1}^{\|F_i\|} \left\| L_{jfi} - \overline{L}_{ji} \right\|^2. \tag{1}$$

We use the mean of the joint variances as a threshold between the inactive and active joints of the video. We eventually consider a joint as active in the motion, if at least $p \in [0, 1]$ of the training videos agreed on it as such, where p is a parameter. We used $p = 0.8$ in our experiments.

Rest Sequences Detection. We aim to detect frame sequences in which the velocities of the active joints are close to 0. We start by filtering and smoothing the noisy videos output by the device using a median filter and a temporal Gaussian pyramid. We then compute the joint velocities, given as the discrete temporal derivative of the video, which is the subtraction of the location vectors of each joint, for every pair of consecutive frames. We then apply a median filter on the velocities. We then compute v^η as a function of the velocity v, which flips the velocity values, where $\eta < -1$ is a parameter. We use the mean as a threshold between rest and non-rest frames. We use $\eta = -1.5$ in our experiments. Figure 4 visualizes this process. We finally consider a frame as a rest frame, if at least $p \in [0, 1]$ of the active joints consider it as such. In our experiments, we use $p = 0.3$. In order to turn the rest indices into rest sequences, we iterate on the indices. If the difference between the current index and the previous index is more than $\rho \in [0, 1]$ of the video length, then we decide that the current index starts a new sequence. We use $\rho = 0.075$ in our experiments. We repeat this step over all the training videos, minding that different amounts of rest sequences can be detected for each video. We use the median over all the training videos in order to define the designated number of rests in the motion. For videos that reported more sequences, we only keep those that best match in time to the other videos. For videos that reported less sequences, we repeat the detection process, each time multiplying η by 0.9, until enough rest sequences are detected.

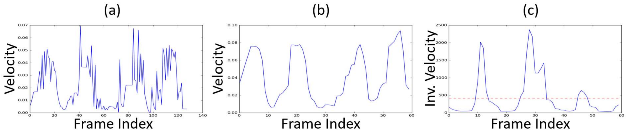


Figure 4: Detection of three motion-structural rest sequences using the velocities of the right hand. From left to right, the raw velocity magnitudes (a) are time-filtered (b) and finally flipped and thresholded (c).

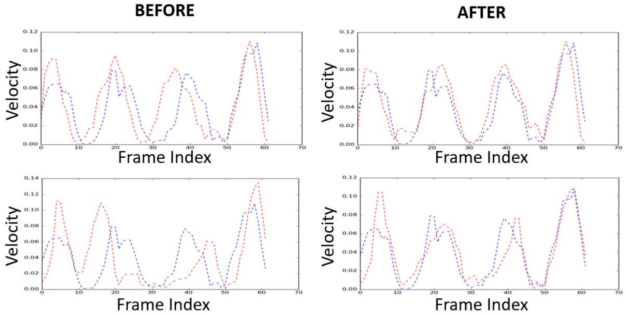


Figure 5: The temporal PoI based warping algorithm in two running examples. The blue and red lines are the hand velocity magnitudes in the reference and aligned videos, respectively (best viewed in color).

Warping the Videos to Match Temporal PoIs. We use the rest sequences start and end frame indices as mutual PoIs and force them to occur at the same time in all the videos. We choose one of the videos to be a reference, such that all the other videos will be temporally aligned to it. The video that is closest to the centroid of all the videos in terms of PoI indices is selected. Then, for each video we temporally scale the sequences between every pair of adjacent PoIs, such that their length will be equal to those from the reference video. Figure 5 illustrates the effectiveness of this step.

3.3. Model Learning

Based on the fact that the skeletons in all the videos are normalized and fixed to the body-center plane and that the videos are aligned in time, the features we extract on every frame are the 3D locations of the joints, their 3D velocities, their pairwise distances and the angles between their edges. Additionally, we use the original lengths of the motion and the sequences from Section 3.2 as time-related parameters. Let us introduce o_i the observation of the parameter o in video i. We only use scalars as observations, which means that vectors, such as locations and velocities, are separated into their three components. Let us introduce $\overline{o}$ as the mean of all the observations o_i and $\overline{o^i}$ as the mean of all the observations excluding o_i. We use the leave-one-out method

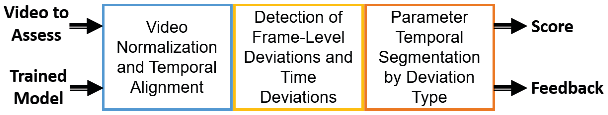

Video to Assess
Trained Model
Video Normalization and Temporal Alignment
Detection of Frame-Level Deviations and Time Deviations
Parameter Temporal Segmentation by Deviation Type
Score
Feedback

Figure 6: The assessment method in high level.

to learn the legitimate deviation of each parameter o. We do that by computing all the differences $|o_i - \overline{o^i}|$ and then computing their mean M_o and standard deviation S_o, which we will use for normalizing deviations when we assess input videos. Our final trained model contains the triplets $(\overline{o}, M_o, S_o)$ for every parameter. In addition, it contains the designated skeleton dimensions from Section 3.1, the video lengths, the detected active joints, the designated indices of the PoIs and the original lengths of the sequences from Section 3.2.

4. Assessment Method

As shown in Figure 6, given a trained model and an input video to assess, we apply the normalization stage from Section 3.1, using the designated skeleton dimensions from the trained model. We then time-warp the input video using the algorithm from Section 3.2, using the detected active joints and rest sequences from the trained model. We then compute the parameter deviations. For each parameter o, we denote d_o as the absolute difference between the input video value and the mean of observations $\overline{o}$, which we have computed in Section 3.3. We then represent the difference in standard deviations, using the distance mean M_o and standard deviation S_o, which we have in the model too:

$$D_o = \frac{d_o - M_o}{S_o + \varepsilon}. \tag{2}$$

We use $\varepsilon = 0.0005$ in our experiments, to avoid division by zero. We threshold the deviations in order to discriminate between accepted and unaccepted ones, using the threshold parameter γ, which we have set to 2.5 standard deviations in our experiments:

$$D_o = \begin{cases} D_o & D_o > \gamma \\ 0 & otherwise \end{cases}. \tag{3}$$

4.1. Parameter Deviation Time Segmentation

We aim to aggregate the frame-level deviations into temporal segments that differ by deviation type, as illustrated in Figure 7, which is essential for producing meaningful feedback and eliminate local deviation noise. As mentioned in Section 3.3, all the deviations are scalars. Therefore, the aggregated deviation types we support are: 1. No Deviation. 2. Positive Deviation. 3. Negative deviation. 4. Unstable deviation.

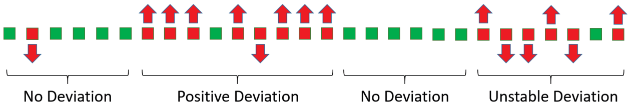


Figure 7: The segmentation aggregates the frame-level deviations (squares) of each scalar parameter into temporal segments, to produce higher-level, more meaningful feedback to the user. The algorithm aims to desirably temporally-segment the deviations, balancing between sequence classification significance and length (best viewed in color).

Sequence Classification. We build a classifier that will classify input sequences into one of the four deviation categories and output classification confidences, which will be used for finding the optimal time segmentation for each parameter. For that reason, let us introduce the rate measure R_T, which is the rate of the sequence indices that have deviation of type T. We then introduce the scattering measure S_T, which aims to tell how well deviation type indices are scattered in the sequence. It computes the index variance of deviation type indices, which we mark as $\widehat{\sigma}^2(seq_T)$ and compares it to the index variance of all the sequence indices, which we mark as $\widehat{\sigma}^2(seq)$. A small difference between the variances will indicate a good scattering.

$$S_T = \left(\frac{\widehat{\sigma}^2(seq_T) - \widehat{\sigma}^2(seq)}{\widehat{\sigma}^2(seq)}\right)^2. \tag{4}$$

We compute the classification score for each elementary deviation type T, which can be positive, negative or none, where λ is a parameter that we set to 0.25 in our experiments:

$$Score_T = \lambda R_T S_T. \tag{5}$$

We then compute the rate measure of the unstable deviation type $R_\sim = \min(1, \rho \cdot \min(R_+, R_-))$, where ρ is a parameter that we set to 2 in our experiments and $R_+, R-$ are the rate measures of the positive and negative deviation types, respectively. The scattering measure $S_\sim$ is simply the average of the scattering measures of the positive and negative deviation types. The final score of the unstable deviation class is then computed as $Score_\sim = R_\sim S_\sim$. We finally add another dummy deviation type that is used to catch all the cases that do not have a clear classification, with score $Score_* = \min(1 - S_+, 1 - S_-, 1 - S_0)$. We apply the Softmax function on the vector of scores and output the class with the highest score. If the dummy class has the highest score, then we output zero score, to prevent the sequence from becoming a segment.

Optimal Time Segmentation. The trivial, yet useless optimal segmentation will be treating each frame as a perfectly classified segment. We exclude such segmentations, by rewarding longer sequences, by multiplying their classification scores by $(1 + \ln(|seq|))^\xi$, where $|seq|$ is the sequence length and $\xi \in \mathbb{R}$ is a parameter. We use $\xi = 0.2$ in our experiments. Let us define $L = \{0, 1, ..., |seq| - 1\}$ as the set of all frame indices in the sequence seq. Let us denote $\mathbb{P}(L)$ as the power set of L, such that $|\mathbb{P}(L)| = 2^{|seq|}$ and such that every $l \in \mathbb{P}(L)$ is a sorted array. We want to find the subset of sorted indices $l \in \mathbb{P}(L)$ that gives the highest average classification score when used as boundaries, which denote $|l| + 1$ segments. We therefore want to find

$$\begin{aligned} \arg\max_{l\in\mathbb{P}(L)}\{ & \frac{1}{|l|+1}[cs(seq[0:l[0]])+ \\ & \sum_{i=1}^{|l|-1} cs(seq[l[i-1]:l[i]]) + cs(seq[l[|l|-1]:|seq|]) \end{aligned}, \tag{6}$$

where the function $cs(sequence)$ returns the sequence classification score, multiplied by the sequence length reward and where $seq[i : j]$ is the subsequence of seq from index i to index j, non inclusive. In order to find the optimal segmentation, we implement a dynamic programming algorithm, with the array $Scores$ that will store for each index in the sequence the score of the best segmentation up to that index, such that for each $i \in \{1, 2, |seq| - 1\}$:

$$\begin{aligned} Scores[i] = \max_{0\le j<i}\{ & \frac{1}{Seg_j + 1}(Scores[j] \cdot Seg_j \\ & + cs(seq[j+1 : i+1]))\} \end{aligned}, \tag{7}$$

where Seg_j denotes the number of segments in the optimal segmentation up to index j.

4.2. Numerical Assessment Score

We aggregate the parameter deviations into a final quality score. We reduce noise by utilizing the classified segments from Section 4.1, discarding all the deviations that were not a part of a segment classified as deviating. We divide our parameters into three sets: A, N and T, which are the set of active-joint related parameters, the set of non-active joint related parameters and the set of time-related parameters, respectively, as described in Section 3.3. For each set $S \in \{A, N, T\}$, we define the subset $S_d \subseteq S$ to be the set of all the deviating parameters from S. We reduce more noise by multiplying all the active-joint related deviations by α^d, where $\alpha \in (0, 1)$ is a parameter and d is the average number of deviating segments. We do it as we expect longer, yet fewer deviating segments on improper performances. We use $\alpha = 0.75$ in our experiments. The score of the parameter set S will be:

$$Score(S) = 1 - \min(1, \frac{1}{|S|}\sum_{s\in S_d} s), \tag{8}$$

We then compute the final score using the weight parameters $\alpha_A, \alpha_N, \alpha_T \in \mathbb{R}$, such that $\alpha_A + \alpha_N + \alpha_T = 1$, which determine the weights of active joint related parameters, non-active joint related parameters and time parameters, respectively. The final score we output is:

$$Score = \alpha_A Score(A) + \alpha_N Score(N) + \alpha_T Score(T). \tag{9}$$

In Section 5, we demonstrate how the produced score accurately discriminates between motions of different performance qualities. We use $\alpha_A = 0.73$, $\alpha_T = 0.25$ and $\alpha_N = 0.02$ in our experiments. We use a low value for α_N in order to almost cancel its contribution to the score, while keeping the non-active joint deviations usable for textual feedback production.

4.3. Feedback Generation

We utilize the output of the deviation segmentation algorithm described in Section 4.1 and collect all the segments that were not classified as proper ones, where each segment represents the behavior of a single parameter over a range of time. Since all our parameters are scalars, the deviation types can only be too high, too low or unstable. An example for such a segment can be a too high horizontal velocity of the right elbow, between two points in time. In addition to the joint-related deviations, we also use all the time parameters. We compute the loss each segment has caused to the final score, according to the process described in Section 4.2 and sort them by importance, to output the most crucial and effective feedback items. We stop generating feedback items when the loss of the current item is less than a half of the loss of the previous item, or when we have already output five items. Figure 10 demonstrates feedbacks that were generated by the algorithm.

5. Experimental Results

FMA Motion Dataset. We tested our method on a dataset of three FMA motion types that are illustrated in Figure 8. We acquired it in a hospital, under guidance of an occupational therapist, who also labeled the motions with FMA scores. These motions have 3 possible scores: 0 if the motion cannot be performed, 1 if it is improperly performed, or 2 if the motion is properly performed. The properly-performed motions, which are the only motions we use for training, were performed by the medical staff, men and women ranging between 30-60 years old. The other motions, were performed by the Occupational Therapy department patients, men and women ranging between 45-75 years old, who were recovering after strokes. The numbers of training and testing video clips are detailed in Table 1 and Table 2, respectively.

Motion Structures and Objectives. As illustrated in Figure 8, in motion number 1, the patient should raise their

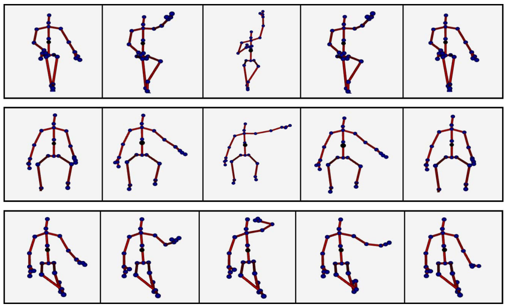


Figure 8: Representative frames from proper performances of the three FMA motion types our model has been trained and tested on. The first motion contains three motion rests, while the second and third motions contain one rest each.

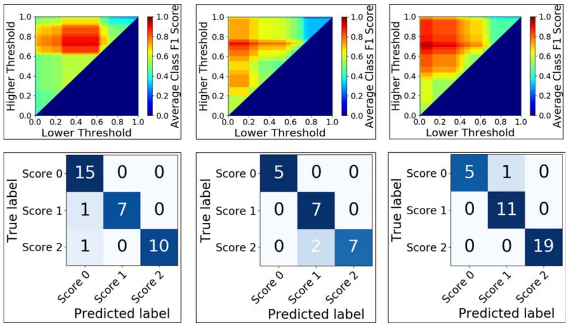


Figure 9: From left to right, the average classification F1 scores at each pair of thresholds, of our three tested FMA motion types and their corresponding confusion matrices with the best pair of thresholds (best viewed in color).

Motion	#Clips	#People	#People with Two Clips
Motion 1	11	7	4
Motion 2	9	5	4
Motion 3	19	10	9

Table 1: The numbers of training videos and performing people per motion type. The training videos only contain properly-performed motions.

Motion	#Class 0 clips	#Class 1 clips
Motion 1	15	8*
Motion 2	5	7
Motion 3	6	11

Table 2: The number of testing videos of classes 0, 1 per motion. (*) As we only had a single genuine video of class 1 for Motion 1, we added seven artificial videos, by applying minor distortions on proper videos.

hand in front of their body with a fully extended elbow, such that their shoulder is in 90°, then raise their hand until it is straight above their head, keeping their elbow extended, then lower their hand back to the front of the body and finally, lower and relax their hand. In motion number 2, the patient should raise their hand aside of the body with a fully extended elbow, such that their shoulder is in 90°, then lower and relax their hand. In motion number 3, the patient should raise their hand and touch their head, by raising their elbow aside of the body and bending it, with shoulder in 90°, then lower and relax their hand.

Evaluation and Ablation Study. We aim to show that even though the model has only been trained on properly-performed motions, the numerical scores it produces can accurately discriminate between videos from the three FMA score groups. When we assess videos from classes 0 and 1, we will use a model that has been trained on all the proper videos of the motion. When we assess videos from class 2, which are the proper videos, we will use a dedicated trained model for each tested video, such that the videos of the tested person will not be included in the training set. As illustrated in Figure 9, for each of the three motions, we look for the pair of thresholds that will most accurately divide the scores into the three classification groups and show the confusion matrix they produce. Table 3 shows the results of the ablation study we have conducted. Since our test data contains videos of real patients, who usually perform the motions slowly, we also show the classification accuracies with $\alpha_T = 0$, to demonstrate that our model can accurately discriminate between the categories even when timings are ignored. In addition, we turn on and off the joint-grouping feature, denoted by **JG**, the time-warping feature, denoted by **TW** and the deviation-segmentation feature, denoted by **DS**, to demonstrate the importance of each. The joint grouping feature is responsible for giving joints that were detected as active a higher weight in the score. The time-warping feature is responsible for video alignment during training and testing. The deviation-segmentation feature is responsible for exclusion of deviating parameters that do not belong to a deviating segment. The ablation study demonstrates the essentiality of each component and the superiority of our temporal PoI based time warping algorithm over DTW. The COW time-warping results were excluded, as they underperformed both DTW and our warping algorithm.

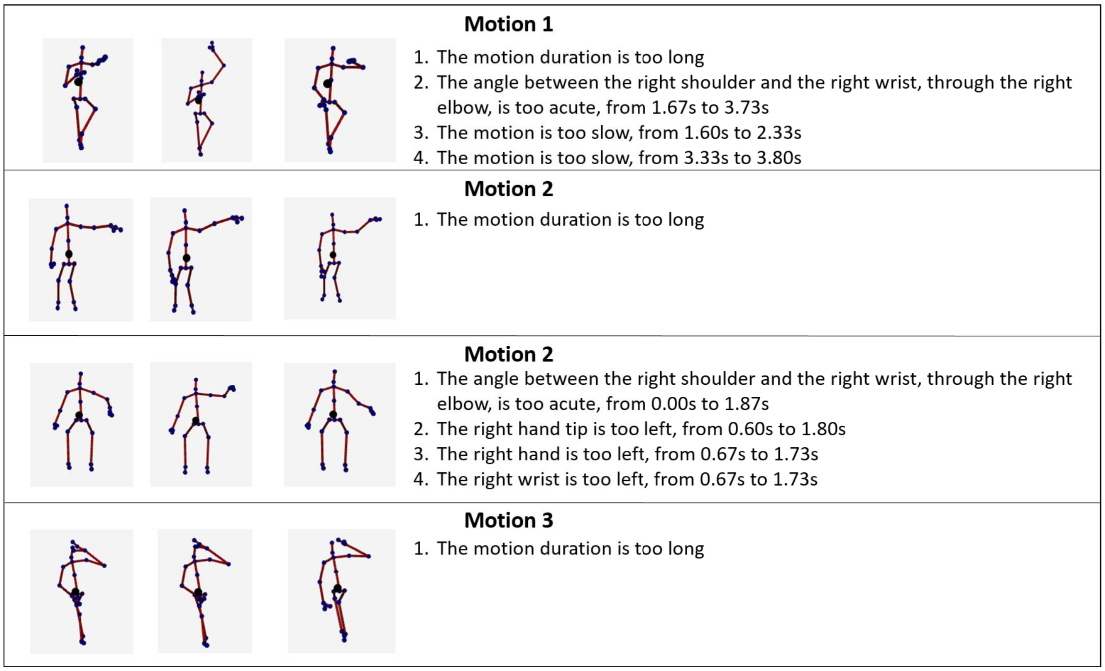
Produced Feedback Examples. We used the feedback generation method detailed in Section 4.3, to generate textual feedback for the most crucial deviations of each of the tested videos. As explained in the method in Section 4.3, the number of feedback items can differ between videos and depends on their relative significance. Figure 10 shows representative frames from improperly performed motions and the feedback that has been generated for them.

6. Conclusions

In this work, we presented an automatic end-to-end motion assessment and feedback generation algorithm that only learned from properly-performed motion videos, without further annotations and which overcame the data noisiness of the relatively cheap Kinect2 device. We introduced a novel continuous time-warping algorithm, based on mu-

Setting	Metric	Motion 1	Motion 2	Motion 3	Average
		$\alpha_T = 0.25$ / $\alpha_T = 0$	$\alpha_T = 0.25$ / $\alpha_T = 0$	$\alpha_T = 0.25$ / $\alpha_T = 0$	$\alpha_T = 0.25$ / $\alpha_T = 0$
TW	Mean F1	0.73 / 0.72	0.80 / 0.57	0.78 / 0.70	0.77 / 0.66
	MSE	0.32 / 0.41	0.33 / 1.00	0.17 / 0.25	0.27 / 0.55
JG+TW	Mean F1	0.76 / 0.83	0.78 / 0.58	0.80 / 0.73	0.78 / 0.71
	MSE	0.29 / 0.24	0.24 / 0.43	0.14 / 0.22	0.22 / 0.30
JG	Mean F1	0.77 / 0.82	0.86 / 0.70	0.78 / 0.76	0.80 / 0.76
	MSE	0.59 / 0.18	0.14 / 0.24	0.17 / 0.19	0.30 / 0.20
DS	Mean F1	0.83 / 0.82	0.82 / 0.78	0.85 / 0.84	0.83 / 0.81
	MSE	0.24 / 0.62	0.19 / 0.24	0.14 / 0.14	0.19 / 0.33
DS+TW	Mean F1	0.84 / 0.74	0.82 / 0.77	0.85 / 0.82	0.84 / 0.78
	MSE	0.32 / 0.68	0.19 / 0.24	0.14 / 0.17	0.22 / 0.36
DS+JG	Mean F1	0.93 / 0.90	0.87 / 0.81	0.91 / 0.87	0.90 / 0.86
	MSE	0.06 / 0.18	0.14 / 0.19	0.06 / 0.11	0.09 / 0.16
DS+JG+DTW	Mean F1	0.96 / 0.90	0.83 / 0.74	0.94 / 0.91	0.91 / 0.85
	MSE	0.03 / 0.18	0.19 / 0.24	0.06 / 0.08	0.09 / 0.17
DS+JG+TW	Mean F1	0.94 / 0.90	0.92 / 0.81	0.96 / 0.91	**0.94 / 0.87**
	MSE	0.15 / 0.18	0.10 / 0.19	0.03 / 0.08	**0.09 / 0.15**

Table 3: Ablation study. The mean F1 score and the corresponding MSE refer to the classification with the best pair of thresholds.



Motion 1

1. The motion duration is too long
2. The angle between the right shoulder and the right wrist, through the right elbow, is too acute, from 1.67s to 3.73s
3. The motion is too slow, from 1.60s to 2.33s
4. The motion is too slow, from 3.33s to 3.80s

Motion 2

1. The motion duration is too long

Motion 2

1. The angle between the right shoulder and the right wrist, through the right elbow, is too acute, from 0.00s to 1.87s
2. The right hand tip is too left, from 0.60s to 1.80s
3. The right hand is too left, from 0.67s to 1.73s
4. The right wrist is too left, from 0.67s to 1.73s

Motion 3

1. The motion duration is too long

Figure 10: The generated textual feedbacks on improperly performed motions, using the trained models, which were only trained on proper motions.

tual PoIs that were automatically detected and extracted from the training samples. Using additional techniques, such as active joint detection and parameter deviation time-segmentation, we demonstrated the robustness of our model on FMA motions, even without time-warping and when ignoring the almost obviously deviating patient timings.

In the future, we plan to extend this work and demonstrate its natural adaptation to other medical tests, such as Berg Balance Scale (BBS) [1]. Additionally, we plan to extend and improve the work by extending the variety of PoI types used as warping anchors. Another future direction may be the synthesis of training samples from our statistical model, which may serve as an input for learning algorithms that are based on the availability of large training sets.

Acknowledgements. We would like to thank the department of occupational therapy in the Galilee Medical Center for their cooperation and in particular Dorit Itah, who guided and labeled the FMA tests. We would also like to thank Nadav Eichler, Shmulik Raz and Hagit Hel-Or for their cooperation. Lastly, we would like to thank the CVPM'19 reviewers for their important insights.

References

[1] K. Berg, S. Wood-Dauphinee, and J. Williams. The balance scale: reliability assessment with elderly residents and patients with an acute stroke. *Scandinavian journal of rehabilitation medicine*, 27(1):27–36, 1995. 8

[2] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD Workshop*, volume 10, pages 359–370. Seattle, WA, 1994. 2

[3] S. Boubou and E. Suzuki. Classifying actions based on histogram of oriented velocity vectors. *Journal of Intelligent Information Systems*, 44(1):49–65, 2015. 2

[4] L. W. Campbell and A. F. Bobick. Recognition of human body motion using phase space constraints. In *Proceedings of the Fifth International Conference on Computer Vision, 1995.*, pages 624–630. IEEE, 1995. 2

[5] X. Chen and M. Koskela. Online RGB-D gesture recognition with extreme learning machines. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction*, pages 467–474. ACM, 2013. 2

[6] R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006. 2

[7] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1110–1118, 2015. 2

[8] N. Eichler, H. Hel-Or, I. Shimshoni, D. Itah, B. Gross, and S. Raz. 3D motion capture system for assessing patient motion during fugl-meyer stroke rehabilitation testing. *IET Computer Vision*, 12(7):963–975, 2018. 2

[9] N. Eichler, H. Hel-Or, I. Shmishoni, D. Itah, B. Gross, and S. Raz. Non-invasive motion analysis for stroke rehabilitation using off the shelf 3D sensors. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018. 2

[10] A. R. Fugl-Meyer, L. Jääskö, I. Leyman, S. Olsson, and S. Steglind. The post-stroke hemiplegic patient. 1. a method for evaluation of physical performance. *Scandinavian journal of rehabilitation medicine*, 7(1):13–31, 1975. 1

[11] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban. Histogram of oriented displacements (HOD): Describing trajectories of human joints for action recognition. In *Twenty-Third International Joint Conference on Artificial Intelligence*, pages 1351–1357, 2013. 2

[12] Y. Gu, H. Do, Y. Ou, and W. Sheng. Human gesture recognition through a kinect sensor. In *IEEE International Conference on Robotics and Biomimetics (ROBIO), 2012*, pages 1379–1384. IEEE, 2012. 2

[13] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham. 3D pose from motion for cross-view action recognition via nonlinear circulant temporal encoding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2601–2608, 2014. 2

[14] F. Han, B. Reily, W. Hoff, and H. Zhang. Space-time representation of people based on 3D skeletal data: A review. *Computer Vision and Image Understanding*, 158:85–105, 2017. 2

[15] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for RGB-D activity recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5344–5352, 2015. 2

[16] D.-A. Huang and K. M. Kitani. Action-reaction: Forecasting the dynamics of human interaction. In *European Conference on Computer Vision*, pages 489–504. Springer, 2014. 2

[17] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations. In *Twenty-Third International Joint Conference on Artificial Intelligence*, volume 13, pages 2466–2472, 2013. 2

[18] I. Kviatkovsky, I. Shimshoni, and E. Rivlin. Person identification from action styles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 84–92, 2015. 2

[19] J. Luo, W. Wang, and H. Qi. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1809–1816, 2013. 2

[20] S. Z. Masood, C. Ellis, A. Nagaraja, M. F. Tappen, J. J. LaViola, and R. Sukthankar. Measuring and reducing observational latency when recognizing actions. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 422–429. IEEE, 2011. 2

[21] B. C. Munsell, A. Temlyakov, C. Qu, and S. Wang. Person identification using full-body motion and anthropometric biometrics from kinect videos. In *European Conference on Computer Vision*, pages 91–100. Springer, 2012. 2

[22] E. Ohn-Bar and M. Trivedi. Joint angles similarities and HOG2 for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 465–470, 2013. 2

[23] A. Paiement, L. Tao, S. Hannuna, M. Camplani, D. Damen, and M. Mirmehdi. Online quality assessment of human movement from skeleton data. In *British Machine Vision Conference*, pages 153–166, 2014. 2

[24] C. Palma, A. Salazar, and F. Vargas. HMM and DTW for evaluation of therapeutical gestures using kinect. *arXiv preprint arXiv:1602.03742*, 2016. 2

[25] G. I. Parisi, S. Magg, and S. Wermter. Human motion assessment in real time using recurrent self-organization. In *25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 71–76. IEEE, 2016. 2

[26] M. Reyes, G. Dominguez, and S. Escalera. Featureweighting in dynamic timewarping for gesture recognition in depth data. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011*, pages 1182–1188. IEEE, 2011. 2

[27] C.-J. Su. Personal rehabilitation exercise assistant with kinect and dynamic time warping. *International Journal of Information and Education Technology*, 3(4):448, 2013. 2

[28] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from rgbd images. In *IEEE International Conference on Robotics and Automation (ICRA), 2012*, pages 842–849. IEEE, 2012. 2

[29] G. Tomasi, F. Van Den Berg, and C. Andersson. Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(5):231–241, 2004. 3

[30] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967. 2

[31] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012*, pages 1290–1297. IEEE, 2012. 2

[32] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu. Modeling 4d human-object interactions for event and object recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3272–3279, 2013. 2

[33] P. Wei, N. Zheng, Y. Zhao, and S.-C. Zhu. Concurrent action detection with structural prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3136–3143, 2013. 2

[34] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012*, pages 20–27. IEEE, 2012. 2

[35] X. Yang and Y. L. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012*, pages 14–19. IEEE, 2012. 2

[36] B. Yao and L. Fei-Fei. Action recognition with exemplar based 2.5 d graph matching. In *European Conference on Computer Vision*, pages 173–186. Springer, 2012. 2

[37] G. Yu, Z. Liu, and J. Yuan. Discriminative orderlet mining for real-time recognition of human-object interaction. In *Asian Conference on Computer Vision*, pages 50–65. Springer, 2014. 2

[38] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3D kinematics descriptor for low-latency action recognition and detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2752–2759, 2013. 2

[39] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng. Online human gesture recognition from motion data streams. In *Proceedings of the 21st ACM International Conference on Multimedia*, pages 23–32. ACM, 2013. 2

[40] F. Zhou, F. De la Torre, and J. K. Hodgins. Hierarchical aligned cluster analysis for temporal clustering of human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):582–596, 2013. 2

[41] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 2