EUROGRAPHICS 2018 / D. Gutierrez and A. Sheffer
(Guest Editors)

Volume 37 (2018), Number 2

Self-similarity Analysis for Motion Capture Cleaning

A. Aristidou[†1], D. Cohen-Or[2], J. K. Hodgins[3], and A. Shamir[1]

[1]The Interdisciplinary Center, Herzliya, Israel, [2]Tel-Aviv University, Israel, [3]Carnegie Mellon University, PA, USA

Abstract

Motion capture sequences may contain erroneous data, especially when the motion is complex or performers are interacting closely and occlusions are frequent. Common practice is to have specialists visually detect the abnormalities and fix them manually. In this paper, we present a method to automatically analyze and fix motion capture sequences by using self-similarity analysis. The premise of this work is that human motion data has a high-degree of self-similarity. Therefore, given enough motion data, erroneous motions are distinct when compared to other motions. We utilize motion-words *that consist of short sequences of transformations of groups of joints around a given motion frame. We search for the K-nearest neighbors (KNN) set of each word using dynamic time warping and use it to detect and fix erroneous motions automatically. We demonstrate the effectiveness of our method in various examples, and evaluate by comparing to alternative methods and to manual cleaning.*

CCS Concepts

•***Computing methodologies*** → ***Motion capture; Motion processing;***

1. Introduction

Motion capture has proven to be an effective technology for capturing and digitizing human (and other) dynamic movements. This technology advanced the ability to define and express complex animations, enlarging the repertoire of human activities and actions that could be used in animation. For accurate capture, the markers placed on the performer must be visible and identifiable throughout the capture. This poses a challenge when performers interact closely as occlusions lead to missing and erroneous data. Still, there is a great advantage in capturing multiple performers simultaneously as their actions and poses are more natural, especially in activities such as dancing and sports. In this work, we introduce a method to automatically analyze and fix motion capture sequences using self-similarity analysis [BCM05].

In traditional motion capture systems missing markers can create unnatural motions and outliers in the reconstructed motion, while data generated from RGB-depth streams may have motion anomalies when limbs are not directly visible to the cameras (see examples in the supplemental video). Most existing techniques aim to improve the acquisition process itself, using hybrid systems that combine different motion capture technologies (e.g., IMUs, RGB-Depths) [TZK*11, SSK*13], using statistical methods [LCP*14, PHLW15], or filtering the positions/rotations of the occluded joints [LMPF10, AL13]. However, these methods are mainly effective for single performers, short-duration occlusions, and they only deal with some level of noise but not outliers. Thus, the common practice is to have motion capture specialists visually detect the abnormalities and fix them manually. This process is time consuming and error prone, preventing the large-scale availability of motion data repositories. For example, we experiment that almost half of the salsa dances stored in the CMU motion capture library [CMU17] have at least one joint with a motion anomaly. This statistic emphasizes the difficulty of detecting erroneous motions manually. The need for automatic methods will only increase as easier marker-less motion capture systems are become available [MSS*17].

Our method allows automatic detection of errors, suggests replacement by plausible similar motions, as well as permitting other applications such as reconstructing full motion from sparse motion data. The method is based on the premise that human motion data has a high-degree of self-similarity. Therefore, given enough motion data in the correct context, erroneous motions are distinct when compared against other motions. Because our method does not operate on marker data but at the joint level, it can treat anomalies caused by any source including: markers and limbs that are (self) occluded, markers that slip or detach during capturing, markers that are mislabeled or attached incorrectly.

We combine all joint rotation values into a matrix which we call the *motion-texture*, where each column represents a frame consisting of the rotation of the joints and each row represents the time-sequence of one joint. *Motion-words* are sub-windows in the motion-texture (see rectangles in Figure 1, top). To define motion-words, we combine sub-groups (that are not necessarily disjoint) of joints based on the body connectivity and the fact that connected joints tend to move in coordination. We find closest matches for

† a.aristidou@ieee.org

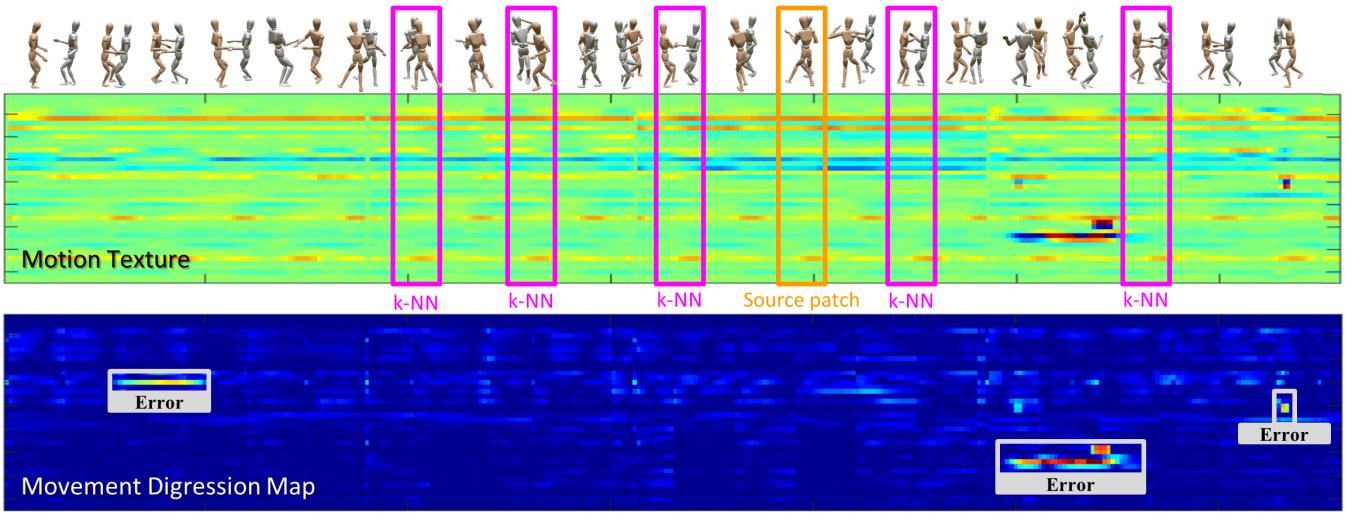

Figure 1: *When capturing multiple performers simultaneously, such as in dance scenes, occlusions and noise lead to missing and erroneous data. We define a motion-texture map (top) where each row represents the rotation angles of a joint through time, and each column is a single pose-frame. Motion-words (shown as rectangles) are short sequences of joints transformations around a motion frame. Our self-similarity analysis is based on comparing each motion-word with its K-nearest neighbors, and building the movement digression map, or MDM (bottom), that indicates unusual movements in time on specific joints. Cold colors (lower values) depict common motions in the MDM, while hot colors (indicating higher values) depict distinct motions that are often erroneous.*

each motion-word and create a mean-motion-texture by averaging the K-nearest neighbors (KNN) of the words. Subtracting the original from the mean-texture creates the *movement digression map* (MDM), where distinct motions can be seen and detected clearly (Figure 1, bottom). The MDM allows specialists to clearly identify both the frames and the specific joints with errors much faster. Moreover, this analysis suggests plausible replacements for the erroneous words automatically by using the median of the KNN.

The contribution of our method lies in three main points.

- Inspired by patch-based self-similarity techniques used in images and video, we do not examine individual motion frames or poses. Instead, we define motion-words as our basic elements for analysis. This is achieved by using joint angles instead of absolute marker positions. Joint angles are *relative* measures that allow more self-similarities to be found in the motion, regardless of the global pose and absolute position of markers.
- At the core of our self-similarity analysis is a time-scale-invariant similarity measure between two motion-words. Since similar motions can vary in duration, plus have local speed variations, we use dynamic time-warping (DTW) to compare them.
- We build an outlier-tolerant distance measure between motion-words. Our approach does not consider noisy pose parts for reconstructing the erroneous motion and allows a more fine-grained representation of the errors by only replacing the erroneous parts instead of full body poses.

We demonstrate the effectiveness of our method in detecting erroneous joints measurement using raw motion data of closely-interacting performers as well as other data. We conducted a comparison with the time required for a motion capture specialist to reliably detect errors using commercial software, or when consulted our MDM, and show that a considerable amount of time can be saved. We also evaluate it with ground truth data, manual cleaning and compare it to several alternative methods. The reader is encouraged to watch examples of the results in the accompanying video.

2. Related Work

Motion Retrieval: Several methods define a number of different distance metrics and features for matching motion sequences to similar examples from a motion database [KGP02, KPZ*04, BCvdPP08]. For instance, Kovar and Gleicher [KG04] use match webs to find numerically similar motions, Müller *et al.* [MRC05] introduce relational features for content-based retrieval, and Kapadia *et al.* [KCT*13] present an LMA-based retrieval method that integrates features which encode both the geometric and dynamic properties of motion. However, relational or geometric features are not suitable for similarity of motion at joint level [ACC15]. Chai and Hodgins [CH05] find the K-closest samples of a posture using metrics that involve pose and velocity features, and then build a graph to accelerate the search of the nearest neighbors. Similarly, Kruger *et al.* [KTWZ10] construct a kd-tree, and then create a lazy neighborhood graphs (LNG) for fast selection of motions that are temporally coherent. We use motion-words instead of single poses as our basic unit to search for similarity in the database, that additionally captures the temporal evolution of motion.

Motion capture errors: Animated motion is subject to errors from character retargeting, body penetration, wrong contact with the environment [VSHJ12, LCX16], and feet sliding [HKT10]. In this work we focus only on motion anomalies due to bad capturing (missing or erroneous marker data). Markers position can slip or be missing due to self or outside occlusions especially for closely

interacting performers. Methods that deal with incomplete or erroneous motion capture data can broadly be classified into three main categories: interpolation and filtering, statistical methods, and data-driven methods.

Interpolation techniques have been extensively used in the past to recover incomplete motion data entries [RCB98]. Several commercial products take this approach, for example, Autodesk Motion-Builder integrates linear or cubic spline interpolations to estimate the intermediate missing marker data. In general, these methods require the data to be well captured before and after the occluded section. An alternative approach is to employ filtering techniques, with carefully defined parameters, to predict the missing marker entries [SLSG01, TK05], while inferred information from neighboring markers and bone-length constraints are integrated for additional control [LMPF10, AL13]. However, the performance of this filtering methods is usually unsatisfactory when all markers are occluded for extended time periods.

Data-driven methods have been popular recently due to the availability of larger databases of motion capture data. They are mainly divided into *marker-based* methods, that primarily exploit the correlations among marker trajectories to estimate the positions of missing entries [HFP*00], and *pose-based*, that typically search for similar posture patterns within a database to reconstruct missing data [HGP04, SDB*12]. In marker-based method, the main idea is to enhance information from nearby markers that share kinematic relations with the occluded markers (rigid cliques) [ZVDH03, GF16]. Park and Hodgins [PH06] fill the empty entries by performing Principal Component Analysis (PCA) to learn the spatial relationship between each marker and its neighbors. Taylor *et al.* [THR06] use Conditional Restricted Boltzmann Machine (CRMB) to impose the correlation between joint angles. A different method for recovering missing positions is to employ matrix factorization, either on block-based motion clips [PHLW15] or by combining it with temporal smoothing [BL16]. Liu *et al.* [LCP*14] treat marker motion data as a mixture of multiple low-dimensional subspaces, and assign similar moving trajectories to their corresponding subspaces using Local Subspace Affinity (LSA), whereas the local information around each trajectory creates a pairwise similarity matrix. Liu *et al.* [LM06] assign motion sequences with missing entries to pre-learnt local linear models, and then recover the occluded positions by finding the least squares solution to the available markers of that model. On the other hand, pose-based methods, such as Lou and Chai [LC10], learn a series of spatial-temporal patterns from prerecorded human motion data; then, they apply a nonlinear optimization framework to minimize the residual between the noisy input and the filtered motion. Feng *et al.* and Xiao *et al.* [FJX*15, XFJ*15] divide human pose into parts to learn multiple dictionaries that contain the spatial-temporal patterns of human motion, that are later adopted to remove the noise and outliers from noisy data using sparse coding. Holden *et al.* [HSKJ15] apply deep learning and auto-encoders to learn a human motion manifold using convolution that encodes the temporal aspect of motion and the pose subspace. Trumble *et al.* [TGM*17] learn a pose embedding from volumetric probabilistic visual hull data to accurately estimate and reconstruct 3D human poses. A bilinear spatiotemporal basis model has been proposed by Akhter *et al.* [ASK*12] that exploits spatial and temporal correlations in motion data, and has been used for filling missing entries and motion data denoising. Unlike these data-driven methods, our method does not necessarily require a pre-training session, it can operate in an unsupervised manner by analyzing the self-similarity of the input sequence, but it may also use a preprocessing stage on larger datasets.

Inspired by the recent advances in image analysis for irregularity detection [IS15], our method detects and reveals non-repetitive erroneous data. Analyzing non-local repetitions to spot anomalies on images is based on the premise that outliers typically have large variation relative to normal image-patches [BKCO16]. In a similar spirit, our method breaks motion into motion-words, and analyzes their prevalence in the data.

Motion Reconstruction: There are several data-driven methods that reconstruct full-body motion streams using sparse motion capture inputs from a small set of inertial sensors by retrieving matched motion sequences from a motion capture library [SH08, TZK*11] Xia *et al.* [XSZF16] learn a dictionary from a large number of complete-incomplete training frame pairs and then recover motions using sparse representations and the learned dictionary through an optimization model. Another option to deal with sparse motion capture data is to model the end effector trajectories and then apply a biomechanically constrained inverse kinematics model to fill in the gaps in the motion data [ACL16]

Error Detection: Most methods in the literature target the problem of reconstructing entire sequences of marker data, while ours can also detect anomalies, and replace them. So far, only a few methods aim to identify abnormal motions. For instance, Ren *et al.* [RPE*05] present a method that quantifies the naturalness of synthesized human motion. The authors decompose the motion into parts; using a large number of features, which consist of joint rotations, linear and angular velocities. They build statistical models to classify movements as natural or unnatural. Kim and Rehg [KR08] introduce epitomic analysis to generate natural motion epitomes that are used to compute a score that indicates whether a motion is natural or not. These methods evaluate the motion naturalness globally, and their naturalness score is computed as an aggregate over the entire sequence. Unlike our method, they do not operate at joint level and are not aimed at detecting the erroneous joint rotations which may occur only in short-time windows.

3. Abnormality Detection

The main goal of our work is the detection of joint rotations that have been incorrectly reconstructed because of noisy (raw) data or missing entries due to occlusions. To promote finding similarities, we convert the (possibly erroneous) raw marker positions to joint angles. Using relative joint angles, instead of absolute position of markers, allows finding more self-similarities in the motion of joints, because of the intrinsic nature of the joint rotations, which are spatially-invariant regardless of the global pose.

3.1. Self-similarity Motion Analysis

To simplify the analysis of time-dependent motion sequences, we use a *motion-texture*, T, that allows motion sequences to be treated

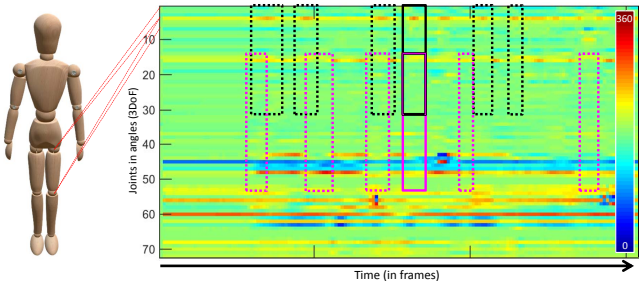

Figure 2: *Motion presented as a texture: each column represents the rotations of the joints over time. Motion-words are illustrated as a rectangle of groups of joints. Solid rectangles define a source motion-word, while dashed rectangles describe their k-nearest neighbors (KNN). Note that the KNN can have a different durations than the source.*

in a similar way to images (see Figure 2). Each column in T represents the degrees of rotations of all joints in one frame, while the horizontal axis is time. We define a *motion-word* as a narrow temporal-window of a group of joints, representing a short-time motion sequence of part of the body (see rectangles in Figure 2). In contrast to single pose feature-sets, our motion-words represent the pose's local evolution in time, allowing the use of spatio-temporal properties of the motion within the cost metric. Note that we use the term motion-texture in a different manner than Li *et al.* [LWS02]; they define a statistical model for motion and generate new instances, while we analyze the original motion as a texture of rotation angles.

Using sub-groups of joints allows the full-body motion to be separated into sub-motions and increases the chance of finding similarities. We define groups of joints that adhere to the human figure topology, and hierarchy to capture joints that move in a coordinated way, and they describe meaningful sub-motions. The joint groups do not form a partition or segmentation of the body and may overlap. In this work, we used the top and bottom, left and right half of the body as well as the whole body as groups to define motion-words, but other groups or combinations of sub-groups could be used as well. All the motion-words form a vocabulary of the entire input motion sequence.

To analyze the motion self-similarity, we measure the words' prevalence in a given context, by considering word-to-word similarity. To build a prevalence measure, we define the *mean-word* as the mean of the KNN-set of a given word. Words that are similar enough to the mean-word are treated as normal. In contrast, words that contain many joint rotations that disagree with their corresponding mean-word are considered distinct. To be robust to outliers, we disregard (zero out) the largest distances (in our case the 3 largest out of 24 joints) in the computation when we compare the motion-word to its mean-word. Note that the rotation values for these joints are discarded only for measuring the distance between motion-words, but are used when producing the mean-motion-words.

Distinct motion words could either contain errors or be *unique* but correct. The differentiation between erroneous and unique motions is challenging because they have very similar characteristics. In both cases, a number of joints differ from the corresponding mean word. Our method deals with this challenge in two ways: first by defining appropriate context, and second by examining joints sub-groups words. By enlarging the corpus of motion data, there is a better chance of finding repetition even for unique motions. Therefore, we define different *contexts* depending on the complexity of the motion. For simple motions (e.g., a walking sequence), the context could be a single motion capture sequence, while for more complex motions, it could include other related motion capture sequences (for example, all salsa dances of a given couple). Second, using smaller groups of joints increases the chance of matching more easily. However, because of human kinematic properties, there are spatial and temporal correlations between joints. Together, searching for word-to-word similarity in a larger context using smaller joint-groups, it is more likely to distinguish unique motions from erroneous (see the discussion regarding context determination in Section 6).

3.2. Duration Invariant Motion-Words Distance

Each motion-word contains a sequence of consecutive frames describing human (sub-)poses. The distance between two motion-words is based on the distance between two motion frames, which reflects the differences in the poses they describe. Various cost functions between body configurations can provide a distance measure to match human poses. For example, the weighted sum of the Euclidean distances between joints [KGP02], or a weighted sum of the difference in rotation between joints [LCR*02]). An extensive discussion and evaluation on cost metrics for matching motion segments can be found in Wang and Bodenheimer [WB03]. To measure the similarity between poses, we first discard the translation and align the facing direction of the root joint†, and then use the formulation presented in Lee *et al.* [LCR*02]:

$$dist_{ij}^2 = \sum_{k=1}^{m} \| \log\left(q_{jk}^{-1} q_{ik}\right) \|^2, \tag{1}$$

where m is the number of joints in the motion word, and q_{ik}, $q_{jk} \in \mathbb{S}^3$ are the complex forms of the quaternion for the kth joint in the i and j frames, respectively. The log-norm term $\| \log\left(q_{jk}^{-1} q_{ik}\right) \|^2$ represents the geodesic norm in quaternion space, which yields the distance from q_{ik} to q_{jk} on $\mathbb{S}^3$.

The distance between two motion-words could now be defined as the sum of distances between their poses in each frames. However, even for a single behavior, such as walking, human motion can vary in duration and speed. Thus, we cannot simply compare fixed duration words. Instead, we use DTW to synchronize two source and target motion-words that may have different durations or variations in speed. DTW compares the skeletal (sub-)pose for each frame of the source word against the (sub-)pose of all frames of the target word using Eq. 1, and finds the optimal matching-sequence that

† By aligning the facing direction of the pose we can compare similar poses regardless of their global position and orientation. Nevertheless, anomalies on the root joint can still be detected based on the rotation of the remaining 2DoF.

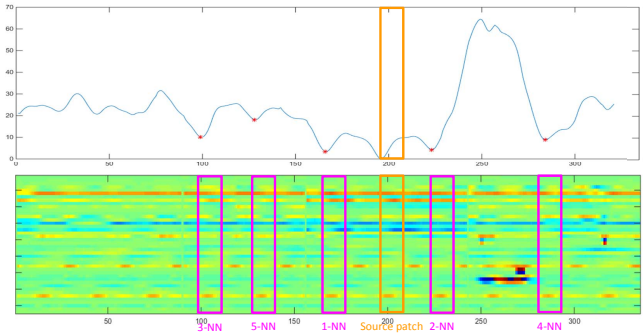

Figure 3: *The selection of the K-nearest neighbor words. The bottom image presents the motion as texture. The top image shows the distance between source word (orange rectangle) and all target words over time. The KNN words (magenta rectangles) are chosen as the K smallest local minimas of the distance function (red stars).*

minimizes the distances between matching pairs (see Appendix for more details). The distance between the two motion-words is defined as the average distance of the matched frames in the optimal matching sequence found by DTW. The time-warped target words are then resized to the source words size by resampling using a polyphase filter [Rot83]. In this manner, our method appreciates the dynamics of motion that are implicitly encoded in the motion-words and the DTW comparison method.

3.3. Defining the KNN

To find the K-nearest neighbors of a given source word, we define a function over all frames whose value is the distance between the source word and the time-warped target word centered at the frame (see Figure 3). The closest K target words are very often overlapping, and may be too close to the source word, not reflecting the real prevalence of the source word in the corpus. Therefore, to choose the KNN, we select the K local minima of the distance function, and discard any local minima that is temporally too close to the source word ($N/2$ frames, where N is the word size). This optimization assures we select KNN that originate in different regions of the motions data.

3.4. The Movement Digression Map

An essential component of our motion abnormality detection method is the generation of the *movement digression map* (MDM). The movement digression is the same size as the motion-texture, and highlights the frames where joint rotation values are distinct in the context of the overall motion. Each entry in the MDM has a divergence value in the range [0,1]. Lower values indicate that the joint rotation at the given frame has a good match in the corpus, while higher values correspond to joints with distinct joint rotation values. In contrast to previous methods that look at the whole pose, our approach highlights not only the timeframes where abnormality appears, but also the *specific* joints that are distinct. An example is illustrated in Figure 1, where the distinct parts are visualized in hot colors.

To build the MDM, we first create the *mean-motion-word* for each source word W_i by averaging the values of the KNN words that were found in the motion corpus $\bar{W}_i = \frac{1}{K}\sum_{j=1}^{K} W_i^j$. We build the *mean-motion-texture*, $\bar{T}$, by averaging all the mean-motion-words that cover every frame. Distinct joint rotation values in the input motion are generally more rare, and therefore, their joint rotation values will differ from the values in the mean-motion-texture. We define the movement digression map MDM using the following procedure. First we find the absolute difference between the motion-texture T and the mean-motion-texture: $M = |T - \bar{T}|$. Second, we normalize each row of the resulting map M by dividing it by the mean of that row (each row defines the changes in the rotation of one joint). Lastly, the movement digression map MDM, is defined using the kernel function $\mathrm{MDM}(i,j) = 1 - e^{-\mathrm{M}(i,j)^2}$.

4. Motion Reconstruction

The MDM can be used as a simple means for motion capture specialists to detect errors and clean them manually (see Section 6.1). However, our method can also automatically replace an erroneous part of the motion with a corresponding plausible set of joint rotation values that have similarities with other parts of the motion. In contrast to previous methods, we only replace the erroneous joint values and do not smooth the full motion, preserving the correct parts and the original style of the motion.

Replacing erroneous joint rotations: For automatic replacement, any element in the movement digression map whose value is higher than a threshold (we use 0.5) is marked as erroneous. Next, the values of the erroneous *joints* in erroneous frames are replaced with corresponding values taken from the median of the KNN set of the word around this frame. We use the median and not the mean KNN as the mean is not an actual motion. When there are contact-point constraints of the character's end-effector positions (as defined by the user), instead of the median, we select the word with the smallest Euclidean distance between the affected end effector(s) and the contact point(s) from the candidate KNN set.

There are numerous methods that allow blending of similar motion examples [KG03, IAF07]. Because we replace only a small set of joints that correspond to the erroneous values, blending is achieved by applying a quadratic Savitzky-Golay filter that allows smooth transitions between the original motion and the amended part. In the constrained case, to ensure that the end-effectors reach the appropriate contact points, we integrate an inverse kinematic solver (FBBIK) [Roo17] that manipulates the skeleton so that the affected end effector moves to the desired position.

Motion reconstruction from sparse data: Our self-similarity analysis can also be used to reconstruct motion from sparse representations. In this scenario, the input is a motion sequence containing only a sparse number of joints, and the goal is to reconstruct the full motion. We search for similarities between the input sequence words to the corresponding subset of joints in a motion that contains full body joints, and select the KNN words that contain the full joint set. The very first word in the output is assigned as the nearest neighbor with the smallest distance to the sparse data. Then, for temporal coherence and smooth transitions, we iteratively

select the motion-word whose first frame has the smallest distance to the last frame of the previous selected word from the candidate KNNs. Lastly, the transitions are smoothed using a quadratic Savitzky-Golay filter.

For both scenarios, replacing erroneous motions or filling in sparse data, we employ a post-processing step that involves motion refinement for foot-skating and/or foot-plant violations [LMT07].

5. Applications

In this section, we present several examples for applications of our self-similarity analysis. Methods that can automatically detect and repair errors in motion data will play a useful role in the motion capture pipeline, as detecting errors can be a time-consuming and complicated process. To locate potential anomalies, all joints for the entire duration of motion must be examined. Our method not only flags the frames that contain errors, but also indicates the joints where the error appears and provides feasible replacements. Thus, the MDM can be considered as a fast and effective auxiliary tool to detect, and later repair errors, or fill in sparse or missing data.

Implementation: For our experiments, we use the $m = 24$ most informative joints with their relative joint angles. All motion capture data were sampled at 120 frames per second; by taking into account that human motion is locally linear and smooth (see e.g., [FF05]), data were reduced for computational efficiency to 24 frames per second, without sacrificing useful information. Selecting the appropriate length for the source and target motion-words is important since short words do not capture the temporal consistency of motion (e.g. fail to detect abrupt changes in acceleration), while longer words have smaller correlation with others, resulting in an increase of false positive detections. We tested different sizes and chose the one that provided the highest accuracy in detection. We use 15 frames for source words, but skip every 5 frames to reduce computation time (hence, two consecutive source words overlap in 10 frames), and 20 frames for target words, without skipping. We search for the $K = 5$ nearest neighbors of every source word. Here too, we tested several values and found that 5 performed best. To be robust to outliers, we chose in our distance metric of eq. 1 to disregard the three joints with the largest distance for large motion-words, or one joint with largest distance for smaller motion-words (sub-groups of joints).

All experiments were run on a six-core PC with Intel i7-6850K at 3.6GHz, 32GB RAM using MATLAB R2014b under Windows 10 operating system. We used motion data taken from the Carnegie Mellon University motion capture database [CMU17], from Seoul National University movement research lab [WLO*14], as well as optical motion data acquired using an 8-cameras PhaseSpace Impulse X2 motion capture system [Pha17]. For the optical acquisition, the subjects usually wore 38 markers (active LEDs), and their 3D positions were located by the surrounding cameras. We also collected skeleton motion data using two RGB-depth sensors (Microsoft Kinects) [iPi17]. The RGB-depth data was post-processed to smooth motion and remove jitters. Raw marker data, as well as RGB-depth motion data, were transformed into Biovision hierarchy format (.bvh) that includes absolute root position and orientation of the relative joint angles.

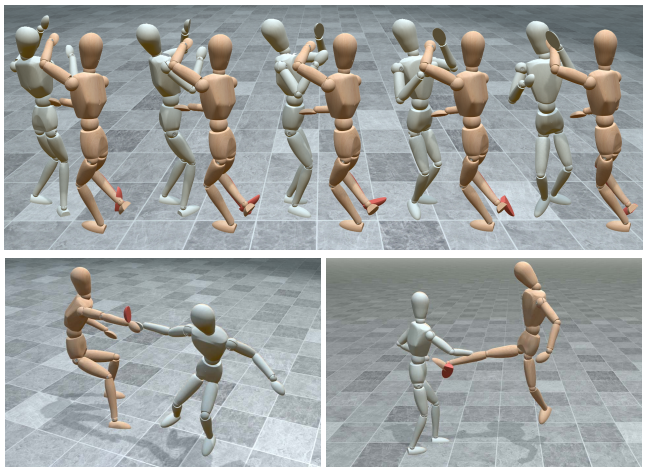

Figure 4: *Error detection and correction on close interacting performers. Top: a salsa dance motion taken from the CMU motion capture database, where the performer is closely interacting with his/her dance-partner (subjects 60 & 61). Bottom: performers pushing each other and a kung-fu fight. Such data contains a number of joint anomalies; the erroneous parts are highlighted in red, while the automatic plausible motion that was used to replace the joint rotation are colored in brown.*

Performing self-similarity analysis on a 20 second motion (with approximately 95 motion-words) takes approximately 140 seconds. However, as we were not using a search structure, the execution time grows exponentially when using larger contexts. For instance, on a database with total length of approximately 600 seconds, our method required around three hours to perform the similarity analysis, and detect the erroneous areas. One way to improve scalability is to use a search structure and dimensionality reduction techniques, such as Principal Component Analysis; however, the use of DTW as the distance measure between words makes it difficult to combine it with such methods. Thus, we reduce the computation time by dividing the error detection process into two stages. In the first stage, we apply the word-to-word self-similarity analysis on all the joints, but search only locally (e.g., using only the motion capture sequence itself). Such local similarity analysis may have a number of false positive detections. In a second stage, we use sub-groups of joints and compare against a larger-scale context but only on areas around the distinct words found in the first step. Experiments show that the two-step framework obtains almost as high precision and recall rates, while substantially reducing the required computation time.

Error Detection and Replacement. Examples for error detection and replacement using our method on motions generated using optical markers data are shown in Figure 4, and on data generated from RGB-depth sensors in Figure 5. For all examples, the character's skeleton was divided into four overlapping groups. The context for the word-to-word similarity of the marker-based clips included a medium-scale database of 15 salsa and contemporary dances, respectively, while the length of each motion clip in the database is in the range of 10-15 seconds. For the RGB-depth mo-

Figure 5: *Error detection and correction on motion generated from RGB-depth data (left). The original skeleton as returned by the iPi software is shown in the middle, and the right image presents the repaired motion using our method. The context data in this case came from optical motion capture acquisition.*

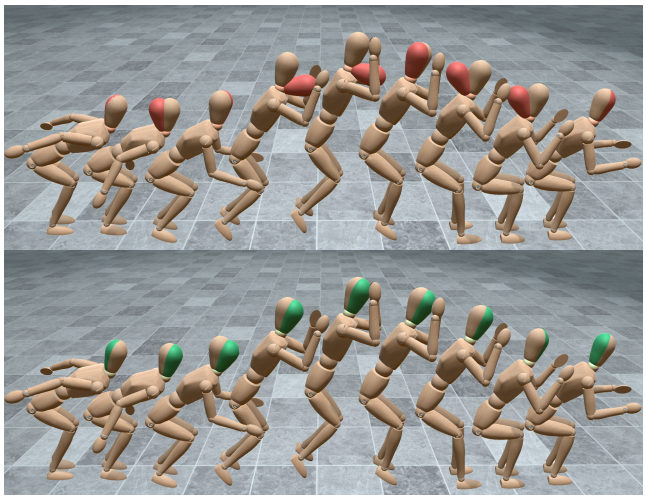

Figure 6: *In this example, motion data of different type of motion are used as the context to detect and replace errors. The top image shows a gymnastics example, where the head rotations have been artificially modified (shown in red). Using contemporary dance motions, our method successfully detects and corrects the error (shown in green at the bottom).*

tion clips, the context used was also a medium-scale database of 15 similar locomotion clips taken from the CMU database.

Using Different Contexts. In general, there is a higher detection rate when subgroups and lager contexts are used. However, when data availability is limited, we can still detect and reconstruct motions using other types of motions. The example described in Figure 5 uses data from optical motion capture as the context for correcting a RGBD capture. In another example, finding a large and clean dataset for gymnastic motions comprising of four different pirouettes is difficult. As the length of these motions is short (approximately six seconds), there were not enough similar movements to build the mean-motion-word. We created a larger-scale context by using contemporary dance sequences in addition to the gymnastics sequences. Even though these motions are of a different type, our method can successfully detect the erroneous joint rotations (see Figure 6); that was possible because our method found similarities in the context of sub-words.

Figure 7: *Integrating contact constraints to further improve motion reconstruction. The motion containing error (red hand) is shown on the left. The middle image shows the reconstructed posture using median, and the right image shows the reconstruction incorporating hand contact constraints (green hand).*

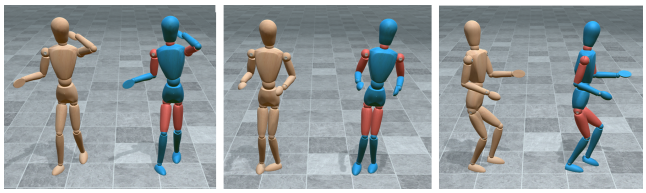

Figure 8: *Using a sparse data of nine selected joint rotations (out of the 24 available joints) shown in red, we have managed to reconstruct the full-skeleton motion of a salsa dancing from a database consisting of similar motion capture-data.*

Contact Constraints. Figure 7 illustrates an example that demonstrates the importance of contact constraints. It compares the reconstructed motion when the median of the KNN was selected to a version where the closest word to meet the contact constraints (contact between the hands of the two performers) was selected.

Sparse Motion Reconstruction. Figure 8 illustrates an example from salsa dancing (subject: 61_04) where full-skeleton motion has been reconstructed using a set of only nine joint rotation values (neck, left and right shoulders, elbows, hips and knees), by searching for similarities in a database of 29 salsa dances. The reconstructed postures are similar to the original motion, illustrating that our method selects the most appropriate motion-words for refining motions (tiny differences in the reconstruction compared to the original motion can be seen mainly in the feet).

6. Evaluation

In this section, we present several experiments evaluating our method, as well as compare to previous approaches and to manual correction. We use data of closely interacting motions (e.g., salsa dances, waltz and kung-fu fighting), as well as highly dynamic and complex movements (e.g., contemporary dance and gymnastics), and locomotion. Such data contains numerous joints with anomalies and comprise many different error types, including rapid changes in rotation that cause unrealistic joint accelerations, abnormal joint rotations, and joint rotations that violate human constraints.

We used our method to detect erroneous joint rotations in the CMU motion capture database that remained even after manually

cleaning. In complex and dynamic motions such as Salsa dance with closely interacting performers there are many errors (in subject 60 we found errors in 4 different joints, in 10 out of 15 clips, and in subject 61 we algorithmically found and manually verified errors in 3 joints, in 3 out of 15 clips). However, even in simple motions such as locomotion, it is sometimes difficult to manually find and clean all errors.

Precision & Recall: Because the number of erroneous words is very small compared to normal words, the accuracy of any method would be very high if it simply found no errors. The important measure to examine is recall, i.e. what percentage of errors are detected? We conducted a statistical analysis to compute the recall $(TP/(TP+FN))$ and precision $(TP/(TP+FP))$, where TP denotes the true positive rate of detecting joint rotation anomalies, FN the false negative rate, and FP the false positive rate. We use two different types of motion datasets: human locomotion, and a complex dynamic dance motion. The total length of motion data used for evaluation is approximately 40,000 frames, resulting in 8,000 motion-words. As ground truth we used both data that was examined by a motion capture specialist, and data where we artificially introduced some errors. About 1,000 words have been verified to include at least one joint with erroneous rotations. The detailed statistics, as well as computation time (per clip) for varying sizes of databases are reported in Table 1. Both recall and precision of our method are very high for both error types.

We compared our method to three baseline alternatives: (a) a method that checks whether the joint rotation values violate anthropometric constraints, (b) a method that detects unrealistic abrupt changes in joint rotations by checking if their derivative is above a threshold, and (c) a combination of the two methods. We searched over a large dataset of clean data to define the limits for both the joints' rotation values, and rotation change rate. Even though the precision rate of these methods is high (see Table 1), the recall rates are significantly lower than our method, indicating that only a small percentage of errors are detected. These methods perform worse than ours because they cannot detect anomalies in motion related to the temporal and spatial correlation of the joints. A joint may satisfy the rotational or derivative constraints, but if it does not respect its temporal and spatial correlations with other joints, the overall motion may look unnatural. Moreover, these methods can only detect the erroneous areas, and can not suggest a plausible replacement as our method does.

Context Determination: The tradeoff between the complexity of the input motion and the size of context used for similarity analysis impacts the error detection rate. The different contexts presented in Table 1 include only the motion capture sequence itself (local), 5-10 similar additional sequences (small-scale), and 20-30 sequences (large-scale). Each sequence is approximately 15 seconds long. The table states the time per-clip required to detect the error in the different scenarios. Simple repetitive movements, such as human locomotion, have high precision and recall rates even when using small-scale contexts. In contrast, complex motions have higher rate of false positive detections when using only local context. Using larger datasets of similar motions allows finding matches for less common motion words leading to better distinction between unique and erroneous motions (see Figure 9). In general, the use of larger contexts increases the recall rate, at the cost of higher computational time.

		Local		Small-Scale DB		Large-Scale DB	
		Average Time: 2min		Average Time: 20min		Average Time: 90min	
		Locomotion	Dance	Locomotion	Dance	Locomotion	Dance
SSA (ours)	**Recall:**	0.985	0.947	0.985	0.952	0.988	0.968
	Precision:	0.875	0.785	0.919	0.832	0.936	0.896
SSA with Chai & Hodgins	**Recall:**	0.934	0.846	0.937	0.867	0.937	0.873
	Precision:	0.816	0.766	0.837	0.819	0.869	0.804

	Baseline (a)		Baseline (b)		Baseline (c)	
	Locomotion	Dance	Locomotion	Dance	Locomotion	Dance
Recall:	0.529	0.365	0.075	0.049	0.587	0.395
Precision:	1.000	1.000	0.969	0.897	0.996	0.986

Table 1: *Top: the recall and precision rates (and time) of our method for detecting errors on two different types of motions and various sizes of contexts (databases used for searching KNN). We compare our scale invariant metric to using the distance measure from Chai and Hodgins [CH05]. We also show (bottom table) the statistics of three baseline methods on this data (see text for details).*

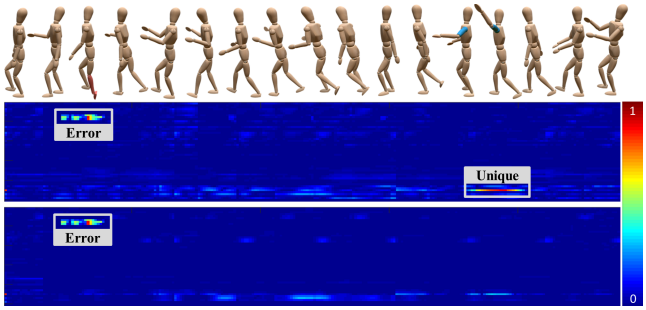

Figure 9: *The top movement digression map shows error detection on a walking example when searching locally, and the bottom map when searching on a small-scale database. Note that in the top map a unique motion was labelled erroneous, because no similar motion was included in the examined motion sequence.*

Comparing to Manual Correction Pipeline: To assess the performance of our method in repairing erroneous motion, we asked a motion capture expert to manually clean noisy raw markers data. Then, we converted both the raw and the cleaned data to joint angles. The motion generated using raw marker data was processed using our method to detect and clean the erroneous parts. Figure 10 compares the MDM created by computing the difference between the input motion and the corresponding manually cleaned motion, and our MDM. The high values, which are correlated with erroneous motion, match on the two MDMs, demonstrating the accuracy of our method in correctly detecting erroneous joints. Figure 11 also shows a quantitative comparison between the joint rotations of our automatic replacement method and the manually cleaned joint rotations; it can be observed that our method significantly improves the erroneous parts, and is very similar to the motion that has been manually corrected.

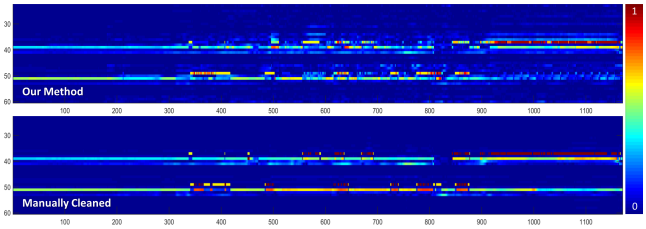

Figure 10: *The top image portrays a portion of the MDM generated by our self-similarity algorithm, and the bottom image the MDM created by computing the difference between the input motion and its corresponding manually cleaned motion; it can be observed that the high values match on the two images, demonstrating the accuracy of our method in detecting errors in joint rotations.*

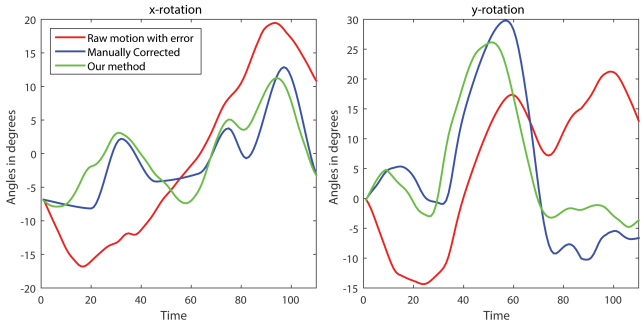

Figure 11: *Comparison of the joint x and y-rotation values between a motion that has been generated using raw motion capture data (red), a motion that has been generated using markers after being cleaned by a motion capture specialist (blue), and a motion that has been processed by our method (green). Our method successfully replaces the erroneous parts with joint rotations that match well to the manually cleaned motion.*

Distance Measure: To evaluate our distance measure, we compared performance of our method using a time-warping version of the Chai and Hodgins [CH05] distance metric to the one we use. Chai and Hodgins select the closest KNN by computing a mixture of features per pose that involves the joint's Euclidean distance and velocity. However, joints located at similar positions but with different rotations cannot be matched, returning higher error rates, especially at the end-effectors. For instance, in Figure 12, when Chai and Hodgins distance metric was used, there are more false positives (e.g. at frames 200-220 and 280-300) compared to using our distance measure. Our method's recall and precision, in this example, are 95.2% and 83.2%, respectively, while for Chai and Hodgins [CH05] measure, the rates are 86.7% and 81.9% (see Table 1).

6.1. User Study

To evaluate the impact of our movement digression-map in assisting motion capture specialists, we have conducted two studies. In the first study, five master degree students with a background in animation were selected from the multimedia department. The participants were provided with four motion streams of salsa dancing

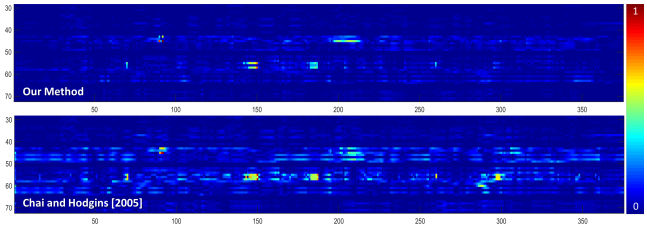

Figure 12: *The top image illustrates a sub-part of the MDM when the closest KNN candidates are selected using our time-scale invariant motion similarity metric, whereas the bottom image shows the MDM when Chai and Hodgins [CH05] method was used. Higher values (hot colors) represent distinct motions to the mean-motion-texture; in this example, Chai and Hodgins method has higher false positives rates, wrongly detecting anomalies on the right hand joint.*

in pairs that comprise approximately the same number and types of real errors. Two streams were provided with their MDM and two without. Participants were asked to manually examine the motions and find joint anomalies. We recorded the time needed to complete the task, and the percentage of successful error detection. Without the use of MDM, the students examined frame-by-frame, as well as the x, y, z-rotation over time for all joints. In contrast, when the MDM was provided, the participants just confirmed/rejected the suggested erroneous areas, and then had a quick check on the rest of the animation to see if there are other obvious anomalies. When the MDM was used, participants spent 50% less time to detect the errors. In addition, in the absence of the MDM, two students failed to detect one (out of 15) short-duration error.

In the second study, we compared the amount of time it takes an expert to detect and label errors in a motion capture sequence manually, vs. using our MDM. We used two groups of 15 clips each from the salsa motion capture data (CMU motion capture database: subjects 60 and 61), where each group contains about the same number of errors. The expert was asked to detect incorrect joint rotations in the data. He received one group of data, and we recorded the time to complete the task. Then, he received the second group of data, but this time with the MDM of each sequence. The time for tasks completion was recorded. The results show a drop from 60 minutes to 20 minutes indicating that a significant amount of time can be saved.

6.2. Comparison with Other Methods

We compared our approach with other recent methods; we chose methods that correct motion at the marker level, including matrix factorization [BL16], and methods that work at the joint level, including deep learning [HSKJ15], and sparse coding [FJX*15]. The experiments were implemented using data containing motion sequences of various locomotion (walk, run, jump), and dancing. It can be observed that our method can detect joint anomalies even in motion streams that have been previously reconstructed, or denoised. Figure 13 shows an example where our method detects errors on data that have been corrected using (a) Burke and Lasenby [BL16] (shown in yellow), and (b) Feng *et al.* [FJX*15]

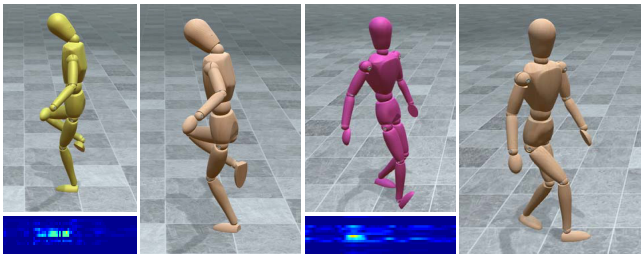

Figure 13: *Top images show the reconstructed poses using (a) Burke and Lasenby (yellow), and (b) Feng et al.(magenta), while bottom images present the MDM after importing the reconstructed motion into our framework, highlighting the joints with error. The right image in each pair illustrates the reconstructed pose using our method.*

(shown in magenta). As can be seen, our method still finds joints with erroneous rotations in the corrected sequences of previous methods, and can correct them.

We compared our method to the method proposed by Gløersen and Federolf [GF16]. The main limitations of this method is that it only fills gaps of incomplete sequences, cannot treat errors due to marker missing or mislabeled, and does not detect joint anomalies in the reconstructed motion. Moreover, if gaps occur in several markers of the same body segment at the same time, the method cannot return good estimates because no inferred information from neighboring markers can be retrieved. The method was tested using contemporary and salsa dancing data. However, because the proposed algorithm is designed based on the assumption that movements are cyclic or repetitive, while dancing is dynamic and heterogenous, the reconstructed markers trajectories have discontinuities, generating motion with abrupt changes and anomalies in joint rotations (see the supplementary materials). In contrast, our method is effective on non-repetitive and dynamic data.

The biggest advantage of our method compared to other alternatives is its ability to detect the joints that contain errors, and repair only those. In contrast, methods such as Holden *et al.* [HSKJ15] and Feng *et al.* [FJX*15] repair the global motion at once, smoothing even the rotations of correct joints. This operation has the effect of losing some of the details of the motion, changing the overall posture of the performer, and removing nuance or style (see Figure 14 and the supplemental video, particularly the performers' head and/or arms). Although is possible to avoid smoothing the full motion by applying denoising on sub-groups, it is still necessary to first detect the erroneous parts. Moreover, even though the filtering and machine learning approaches are capable of smoothing erroneous motion, they have difficulties to refine motion in cases where the joint rotation is plausible, but erroneous in the context of the specific motion.

In an attempt to quantify the reconstruction quality, we have artificially introduced errors on a number of joints, reconstructed the motion using our method and measured the difference between the original, p_u, and reconstructed, $\hat{p}_u$ pose using Lee *et al.* [LCR*02]

Figure 14: *Motion reconstruction using different methods on contemporary dance and gymnastics. In contrast to Feng et al. [FJX*15] (magenta character) and Holden et al. [HSKJ15] (blue character), our method (green character) detects the joint with anomalies (highlighted in red) and repairs only the erroneous values. Other methods apply smoothing to the whole motion, sometimes resulting in changes of the character's posture/style even at times with no error (top image).*

distance formula:

$$Error(p_u, \hat{p}_u) = \sum_{k=1}^{m} \| \log \left(q_{uk}^{-1} \hat{q}_{uk} \right) \|^2, \quad (2)$$

where m is the number of joints. The errors are manually introduced by perturbing the joint rotatations of the character imitating errors found in the dataset. Table 2 shows the performance of the tested methods. Our method returns the smallest difference per frame to the original motion; because other methods (Holden *et al.* [HSKJ15] and Feng *et al.* [FJX*15]) include smoothing that modifies the motion globally, while our method detects and repairs only the erroneous joint rotations and frames. Furthermore, our method returns the smallest error per joint for the erroneous areas, and has the smallest maximum error among the methods used in the experiment. Please refer to the accompanying video for more examples, including dance data, locomotion and various other movements.

7. Discussion

We have presented a method to automatically analyze motion capture sequences based on self-similarity. Our method can detect errors in sequences of closely interacting performers as well as other complex motions. We define motion-words consisting of short-sequences of joints transformations, and use a time-scale invariant similarity measure to compare words with their KNN. This approach allows the detection of abnormalities in the sequence as well as suggesting possible corrections. The high detection rates, and the ability to replace the erroneous parts with plausible motions, can

	Error globally		Error per joint	
	Average	Maximum	Average	Maximum
SSA (ours)	0.85	1.11	0.21	0.44
Holden *et al.*	7.68	8.62	0.32	0.51
Feng *et al.*	7.92	11.24	0.53	0.57
Burke and Lasenby	4.55	6.76	2.50	3.75
Gløersen and Federolf	5.03	7.85	2.88	4.21

Table 2: *The performance of the methods in repairing artificial errors; our method returns the smallest difference per frame and per joint to the original motion (at both mean and maximum values).*

Figure 15: *Our method can detect the errors in long erroneous sequences but encounters difficulty in replacing them. In this pair salsa dance example, the errors in the red character (the occlusion was caused by the close interaction of its pair, the gray character) are detected but the replaced motion (brown) still does not look plausible.*

save considerable time and manual effort in creating clean motion capture data.

Our method has some limitations. First, we can analyze and compare only similar skeletons. The joints must have correspondences in the same hierarchy so motion-words would correspond. Our self-similarity analysis is also applicable for non-humanoid skeleton, if there exists enough data in the corpus for analysis. In cases where skeletons with different configurations are used, it could be possible to use our method, if all motions in the dataset will be retargeted to have a similar skeletal structure. Existing techniques for retargeting could be used to adapt between skeletons if necessary.

Second, we focus on joint rotation errors and do not deal with motion dynamics or bone-length violations. We also cannot detect errors related to self-collision or contact failures. Third, a basic assumption of our method is that there are similar segments of motion in the dataset used. If we search only in a short sequence, then it is not always possible to find close KNNs to generate a good mean word. If we want to expand the search to other sequences, we need to find similar motion capture sequences that are clean and correct, which may not always be available. However, as our experiments indicate, sometimes it is possible to use motions of similar, but not exactly the same type.

To expand the search to really large datasets (assuming such sets are available) some indexing or hashing scheme would have to be employed. For example, a simpler distance measure between motion-words can first be used to collect a candidates set of words, and then our time-scale invariant distance using DTW can be used to find the KNN from this set.

Self-similarity analysis can also fail to detect continuous (due to bad capturing) or repetitive errors because such errors will be considered common in the data. Finally, even though our method can successfully detect long duration joint errors, it may encounter difficulties in correcting them, especially for motions with extreme pose violations. An example is depicted in Figure 15, where the erroneous joint rotations have been detected but the corrected motion still looks unnatural.

Our method could be used for partial or full-body motion retrieval or motion editing, by allowing users to change the position of the end effectors and then search in the database for similar actions to blend. Our analysis could also be used to characterize different motion types by finding similarities in the sequence level - measuring how close are two motions by measuring how common words from one sequence are in the other.

Acknowledgements

This research was supported by the Israel Science Foundation as part of the ISF-NSFC joint program grant number 2216/15; the work was also partially supported by ISF grant 2366/16.

References

[ACC15] ARISTIDOU A., CHARALAMBOUS P., CHRYSANTHOU Y.: Emotion analysis and classification: Understanding the performers' emotions using the lma entities. *Comput. Graph. Forum 34*, 6 (Sept. 2015), 262–276. 2

[ACL16] ARISTIDOU A., CHRYSANTHOU Y., LASENBY J.: Extending FABRIK with model constraints. *Comput. Animat. Virtual Worlds 27*, 1 (January 2016), 35–57. 3

[AL13] ARISTIDOU A., LASENBY J.: Real-time marker prediction and CoR estimation in optical motion capture. *Vis. Comput. 29*, 1 (2013), 7–26. 1, 3

[ASK*12] AKHTER I., SIMON T., KHAN S., MATTHEWS I., SHEIKH Y.: Bilinear spatiotemporal basis models. *ACM Trans. Graph. 31*, 2 (Apr. 2012), 17:1–17:12. 3

[BCM05] BUADES A., COLL B., MOREL J.-M.: A non-local algorithm for image denoising. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition - Volume 02* (Washington, DC, USA, 2005), CVPR '05, IEEE Computer Society, pp. 60–65. 1

[BCvdPP08] BEAUDOIN P., COROS S., VAN DE PANNE M., POULIN P.: Motion-motif graphs. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, CH, 2008), SCA '08, Eurographics Association, pp. 117–126. 2

[BKCO16] BELLINI R., KLEIMAN Y., COHEN-OR D.: Time-varying weathering in texture space. *ACM Trans. Graph. 35*, 4 (July 2016), 141:1–141:11. 3

[BL16] BURKE M., LASENBY J.: Estimating missing marker positions using low dimensional kalman smoothing. *Journal of Biomechanics 49*, 9 (2016), 1854–1858. 3, 9

[CH05] CHAI J., HODGINS J. K.: Performance animation from low-dimensional control signals. *ACM Trans. Graph. 24*, 3 (July 2005), 686–696. 2, 8, 9

[CMU17] CMU: Carnegie Mellon University MoCap Database: http://mocap.cs.cmu.edu/, 2017. 1, 6

[FF05] FORBES K., FIUME E.: An efficient search algorithm for motion data using weighted PCA. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (NY, USA, 2005), SCA '05, ACM, pp. 67–76. 6

[FJX*15] FENG Y., JI M., XIAO J., YANG X., ZHANG J. J., ZHUANG Y., LI X.: Mining spatial-temporal patterns and structural sparsity for human motion data denoising. *IEEE Transactions on Cybernetics 45*, 12 (Dec 2015), 2693–2706. 3, 9, 10

[GF16] GLØERSEN Ø., FEDEROLF P.: Predicting missing marker trajectories in human motion data using marker intercorrelations. *PLOS ONE 11*, 3 (2016), 1–14. 3, 10

[HFP*00] HERDA L., FUA P., PLÄNKERS R., BOULIC R., THALMANN D.: Skeleton-based motion capture for robust reconstruction of human motion. In *Proc. of the Computer Animation* (Washington, DC, USA, 2000), CA '00, IEEE Computer Society, pp. 77–86. 3

[HGP04] HSU E., GENTRY S., POPOVIĆ J.: Example-based control of human motion. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, CH, 2004), SCA '04, Eurographics Association, pp. 69–77. 3

[HKT10] HO E. S. L., KOMURA T., TAI C.-L.: Spatial relationship preserving character motion adaptation. *ACM Trans. Graph. 29*, 4 (July 2010), 33:1–33:8. 2

[HSKJ15] HOLDEN D., SAITO J., KOMURA T., JOYCE T.: Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs* (NY, USA, 2015), SA '15, ACM, pp. 18:1–18:4. 3, 9, 10

[IAF07] IKEMOTO L., ARIKAN O., FORSYTH D.: Quick transitions with cached multi-way blends. In *Proc. of the Symposium on Interactive 3D Graphics and Games* (NY, USA, 2007), I3D '07, ACM, pp. 145–151. 5

[iPi17] IPI SOFT: iPi Motion Capture: http://www.ipisoft.com/, 2017. 6

[IS15] ILAN S., SHAMIR A.: A survey on data-driven video completion. *Comput. Graph. Forum 34*, 6 (Sept. 2015), 60–85. 3

[KCT*13] KAPADIA M., CHIANG I.-K., THOMAS T., BADLER N. I., KIDER JR. J. T.: Efficient motion retrieval in large motion databases. In *Proc. of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games* (NY, USA, 2013), I3D '13, ACM, pp. 19–28. 2

[KG03] KOVAR L., GLEICHER M.: Flexible automatic motion blending with registration curves. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, CH, 2003), SCA '03, Eurographics Association, pp. 214–224. 5

[KG04] KOVAR L., GLEICHER M.: Automated extraction and parameterization of motions in large data sets. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 559–568. 2

[KGP02] KOVAR L., GLEICHER M., PIGHIN F.: Motion graphs. *ACM Trans. Graph. 21*, 3 (July 2002), 473–482. 2, 4

[KPZ*04] KEOGH E., PALPANAS T., ZORDAN V. B., GUNOPULOS D., CARDLE M.: Indexing large human-motion databases. In *Proc. of the International Conference on Very Large Data Bases* (2004), VLDB '04, pp. 780–791. 2

[KR08] KIM W., REHG J. M.: Detection of unnatural movement using epitomic analysis. In *Proc. of the Seventh International Conference on Machine Learning and Applications* (Washington, DC, USA, 2008), ICMLA '08, IEEE Computer Society, pp. 271–276. 3

[KTWZ10] KRÜGER B., TAUTGES J., WEBER A., ZINKE A.: Fast local and global similarity searches in large motion capture databases. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, CH, 2010), SCA '10, Eurographics Association, pp. 1–10. 2

[LC10] LOU H., CHAI J.: Example-based human motion denoising. *IEEE Transactions on Visualization and Computer Graphics 16*, 5 (Sept. 2010), 870–879. 3

[LCP*14] LIU X., CHEUNG Y.-M., PENG S.-J., CUI Z., ZHONG B., DU J.-X.: Automatic motion capture data denoising via filtered subspace clustering and low rank matrix approximation. *Signal Process. 105* (2014), 350–362. 1, 3

[LCR*02] LEE J., CHAI J., REITSMA P. S. A., HODGINS J. K., POLLARD N. S.: Interactive control of avatars animated with human motion data. *ACM Trans. Graph. 21*, 3 (July 2002), 491–500. 4, 10

[LCX16] LV X., CHAI J., XIA S.: Data-driven inverse dynamics for human motion. *ACM Trans. Graph. 35*, 6 (Nov. 2016), 163:1–163:12. 2

[LM06] LIU G., MCMILLAN L.: Estimation of missing markers in human motion capture. *Vis. Comput. 22*, 9 (Sept. 2006), 721–728. 3

[LMPF10] LI L., MCCANN J., POLLARD N., FALOUTSOS C.: Bolero: A principled technique for including bone length constraints in motion capture occlusion filling. In *Proc.. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, CH, 2010), SCA '10, Eurographics Association, pp. 179–188. 1, 3

[LMT07] LYARD E., MAGNENAT-THALMANN N.: A simple footskate removal method for virtual reality applications. *Vis. Comput. 23*, 9 (Aug. 2007), 689–695. 6

[LWS02] LI Y., WANG T., SHUM H.-Y.: Motion texture: A two-level statistical model for character motion synthesis. *ACM Trans. Graph. 21*, 3 (July 2002), 465–472. 4

[MÖ7] MÜLLER M.: *Dynamic Time Warping*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007, pp. 69–84. 13

[MRC05] MÜLLER M., RÖDER T., CLAUSEN M.: Efficient content-based retrieval of motion capture data. *ACM Trans. Graph. 24*, 3 (July 2005), 677–685. 2

[MSS*17] MEHTA D., SRIDHAR S., SOTNYCHENKO O., RHODIN H., SHAFIEI M., SEIDEL H.-P., XU W., CASAS D., THEOBALT C.: VNect: Real-time 3d human pose estimation with a single rgb camera. *ACM Trans. Graph. 36*, 4 (July 2017), 44:1–44:14. 1

[PH06] PARK S. I., HODGINS J. K.: Capturing and animating skin deformation in human motion. *ACM Trans. Graph. 25*, 3 (July 2006), 881–889. 3

[Pha17] PHASESPACE INC.: Optical Motion Capture Systems: http://www.phasespace.com, 2017. 6

[PHLW15] PENG S.-J., HE G.-F., LIU X., WANG H.-Z.: Hierarchical block-based incomplete human mocap data recovery using adaptive non-negative matrix factorization. *Computer & Graphics 49*, C (June 2015), 10–23. 1, 3

[RCB98] ROSE C., COHEN M. F., BODENHEIMER B.: Verbs and adverbs: Multidimensional motion interpolation. *IEEE Comput. Graph. Appl. 18*, 5 (Sept. 1998), 32–40. 3

[Roo17] ROOT-MOTION: FINAL-IK: http://root-motion.com/, accessed 01/2017, 2017. 5

[Rot83] ROTHWEILER J.: Polyphase quadrature filters - a new subband coding technique. In *IEEE International Conference on Acoustics, Speech, and Signal Processing* (1983), ICASSP '83, IEEE, p. 1280âĂŞ1283. 5

[RPE*05] REN L., PATRICK A., EFROS A. A., HODGINS J. K., REHG J. M.: A data-driven approach to quantifying natural human motion. *ACM Trans. Graph. 24*, 3 (July 2005), 1090–1097. 3

[SDB*12] SHEN W., DENG K., BAI X., LEYVAND T., GUO B., TU Z.: Exemplar-based human action pose correction and tagging. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2012), CVPR '12, IEEE Computer Society, pp. 1784–1791. 3

[SH08] SLYPER R., HODGINS J. K.: Action capture with accelerometers. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, CH, 2008), SCA '08, Eurographics Association, pp. 193–199. 3

[SLSG01] SHIN H. J., LEE J., SHIN S. Y., GLEICHER M.: Computer puppetry: An importance-based approach. *ACM Trans. Graph. 20*, 2 (Apr. 2001), 67–94. 3

[SSK*13] SHOTTON J., SHARP T., KIPMAN A., FITZGIBBON A., FINOCCHIO M., BLAKE A., COOK M., MOORE R.: Real-time human pose recognition in parts from single depth images. *Commun. ACM 56*, 1 (Jan. 2013), 116–124. 1

[TGM*17] TRUMBLE M., GILBERT A., MALLESON C., HILTON A., COLLOMOSSE J.: Total Capture: 3D human pose estimation fusing video and inertial sensors. In *Proc. of the 2017 British Machine Vision Conference* (2017), BMVC '17. 3

[THR06] TAYLOR G. W., HINTON G. E., ROWEIS S.: Modeling human motion using binary latent variables. In *Proc. of the 19th International Conference on Neural Information Processing Systems* (Cambridge, MA, USA, 2006), NIPS'06, MIT Press, pp. 1345–1352. 3

[TK05] TAK S., KO H.-S.: A physically-based motion retargeting filter. *ACM Trans. Graph. 24*, 1 (Jan. 2005), 98–117. 3

[TZK*11] TAUTGES J., ZINKE A., KRÜGER B., BAUMANN J., WEBER A., HELTEN T., MÜLLER M., SEIDEL H.-P., EBERHARDT B.: Motion reconstruction using sparse accelerometer data. *ACM Trans. Graph. 30*, 3 (May 2011), 18:1–18:12. 1, 3

[VSHJ12] VONDRAK M., SIGAL L., HODGINS J., JENKINS O.: Video-based 3d motion capture through biped control. *ACM Trans. Graph. 31*, 4 (July 2012), 27:1–27:12. 2

[WB03] WANG J., BODENHEIMER B.: An evaluation of a cost metric for selecting transitions between motion segments. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, CH, 2003), SCA '03, Eurographics Association, pp. 232–238. 4

[WLO*14] WON J., LEE K., O'SULLIVAN C., HODGINS J. K., LEE J.: Generating and ranking diverse multi-character interactions. *ACM Trans. Graph. 33*, 6 (Nov. 2014), 219:1–219:12. 6

[XFJ*15] XIAO J., FENG Y., JI M., YANG X., ZHANG J. J., ZHUANG Y.: Sparse motion bases selection for human motion denoising. *Signal Process. 110*, C (May 2015), 108–122. 3

[XSZF16] XIA G., SUN H., ZHANG G., FENG L.: Human motion recovery jointly utilizing statistical and kinematic information. *Information Sciences 339*, C (Apr. 2016), 189–205. 3

[ZVDH03] ZORDAN V. B., VAN DER HORST N. C.: Mapping optical motion capture data to skeletal motion using a physical model. In *Proc. of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (Aire-la-Ville, CH, 2003), SCA '03, Eurographics Association, pp. 245–250. 3

Appendix

Let us assume that the source motion word is of duration N, and the target motion word is of duration M, where $N \leq M$. The distance between a pair of frames (i, j), one from the source, and one from the target, is measured using the pose-distance measure $dist_{ij}^2$ in Eq. 1. This defines a *distance-matrix* D of size $N \times M$. To select the optimal sequence $SD \in \{D\}^p$ of matches of length p, we use a constrained-DTW algorithm [MÖ07]. The algorithm returns the lowest cumulative cost under the following five constraints: (a) *boundary conditions:* the sequence starts at the first frame, i.e., $SD(1) = (1,1)$, and stops when any of the two words reach the end, i.e., $SD(p) = (M,:)$ or $SD(p) = (:,N)$; (b) *monotonicity*: the sequence must be monotonically ordered with respect to time; (c) *continuity*: adjacent elements in the sequence are confined to neighboring matrix entries, $SD(i+1) \in \{SD(i)+(1,0), SD(i)+(0,1), SD(i)+(1,1)\}$ for $i \in 1,..,N-1$; (d) *slope constraint*: we avoid excessively large movements in one direction by restricting the movement in the same direction to maximum of 3 consecutive steps; (e) *width constraint*: we ensure that the length of the warping window is larger than 10 frames. A visual representation is illustrated in figure 16.

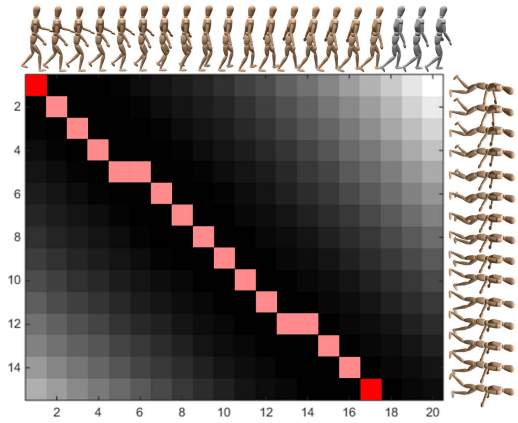

Figure 16: *Time-warped alignment of two walking sequences performed at a different speed. An $N \times M$ distance matrix (darker is smaller) is used by DTW to find the optimal sequence (in red). In this example, the first 17 frames of the target motion-word (20 frames) are matched with 15 frames of the source motion-word.*