Fast Detection of Jitter Artifacts in Human Motion Capture Models

Mateusz Pawłowicz[a], Witold Alda[b] and Krzysztof Boryczko[c]
AGH University of Krakow, Cracow, Poland
{*mpawlowicz, alda, boryczko*}*@agh.edu.pl*

Keywords: Character Animation, Motion Capture, Jitter, Animation Datasets, BVH.

Abstract: Motion capture is the standard when it comes to acquiring detailed motion data for animations. The method is used for high-quality productions in many industries, such as filmmaking and game development. The quality of the outcome and the time needed to achieve it are incomparable with the keyframe-based manual method. However, the motion capture data sometimes gets corrupted, which results in animation artifacts that make it unrealistic and unpleasant to watch. An example of such an artifact is a jitter, which can be defined as the rapid and chaotic movement of a joint. In this work, we focus on detecting the jitter in animation sequences created using motion capture systems. To achieve that, here is proposed a multilevel analysis framework that consists of two metrics: Movement Dynamics Clutter (MDC) and Movement Dynamics Clutter Spectrum Strength (MDCSS). The former measures the dynamics of a joint, while the latter metric allows the classification of a sequence of frames as a jitter. The framework was evaluated on popular datasets to analyze the properties of the metrics. The results of our experiments revealed that two of the popular animation datasets, LAFAN1 and Human3.6M, contain instances of jitter, which was not known before inspection with our method.

1 INTRODUCTION

Motion capture (MoCap) is currently the most potent method for realistic human animations in movies and video games. It has many advantages compared to traditional frame-by-frame creation and procedural animation formulas. With motion capture techniques, animators can quickly obtain even very complex and unique motion, thus drastically reducing overall costs and production time. Motion capture also has its drawbacks and disadvantages. It requires expensive and complex equipment, including cameras, sensors, and sophisticated software. Moreover, to get the most out of this technology, it is also required to cooperate with professionally trained motion capture actors.

The method itself brings about several technical issues that may limit its effectiveness. The first well-known problem is the need for proper calibration of the whole system, including the configuration of cameras and sensors and the correct illumination of the screen. The second common issue, especially with marker-based solutions, is the occlusion of the sensors and/or too fast movement of the sensor, which may cause ”losing” it by the software. The actor also has physical limitations, but we omit them here.

Instead, we notice that the recorded motion capture animation is often noisy and contains spikes, jitter, gaps, and other errors and artifacts. Such data must be cleaned up in a post-processing procedure for smooth and realistic movement. Many tools can help clean up, with popular ones such as Blender, Maya, and MotionBuilder among them. There are also several denoising algorithms, starting from relatively simple ones and ending with sophisticated machine learning-based approaches (Holden, 2018). Still, the processing of raw data is long and often painstaking. Even though we are aware of using post-processing algorithms, we can easily find final animations stored in databases that have errors and produce unnatural motion. Checking the quality of recorded animation simply by carefully observing the motion is a natural process that takes much time.

This paper aims to address this problem by finding measures that can help us detect jitter in animation sequences through automatic analysis of the dataset itself. Our contributions are as follows:

- We propose two metrics: Movement Dynamics Clutter (MDC) metric to detect dynamic and irregular movement of a joint and DFT-based Movement Dynamics Clutter Spectrum Strength (MDCSS) metric for detecting jitter in these

[a] https://orcid.org/0009-0008-2109-505X
[b] https://orcid.org/0000-0002-6769-0152
[c] https://orcid.org/0000-0002-3392-3739

Pawłowicz, M., Alda, W. and Boryczko, K.
Fast Detection of Jitter Artifacts in Human Motion Capture Models.
DOI: 10.5220/0013144400003912

In *Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2025) - Volume 1: GRAPP, HUCAPP and IVAPP*, pages 77-88
ISBN: 978-989-758-728-3; ISSN: 2184-4321

movements;

- We suggest an analysis method that allows the inspection of joint movement dynamics in animation datasets at various levels: dataset level, sequence level, and frame level;
- We evaluate our analysis method to verify its properties and compare popular animation datasets concerning jitter presence;
- Using our proposed metrics, we also detect instances of jitter in two widely-used animation datasets: LAFAN1 (Harvey et al., 2020) and Human3.6M (Ionescu et al., 2014).

2 RELATED WORK

2.1 Errors and Animation Data Clean-up

The technical aspect of MoCap technology is not crucial for our paper. We assume that we get the data "as is" and do not make improvements in the data acquisition process. However, we have to make even a general analysis of the used technology, as it has an important impact on types of errors and artifacts in the animation sequences.

We may distinguish two main approaches to gathering motion data: marker-based and markerless. Optical-based motion capture (OMC) (Callejas-Cuervo et al., 2023) is the most popular and reliable method in the first group. Currently, it has outperformed others, such as inertial or magnetic markers. It uses either passive, reflective markers or active LED markers. Despite their advantages, optical markers have certain flaws, such as occlusion when markers hide from the camera or fast marker movements, which cause gaps and noisy animation. The occasional change of marker position (e.g., when slipped or detached) adds extra distortion. It is also well known that the optical markers system is excellent in the coarse movement of the entire body. At the same time, it loses control of tiny details (such as fingers) and gestures, which often move unnaturally. A comprehensive overview of the detection and classification of errors in optical MoCap systems can be found in (Skurowski and Pawlyta, 2022).

In the second group, the markerless approach, we have depth-sensing cameras that can capture motion without physical markers, thus being convenient and comfortable for the actors. However, this method needs a clean-up process to improve the quality of motion. It can be based on smoothing and denoising algorithms, beneficial for cheap home motion capture systems with, say, a single RGBD camera (Hoxey and Stephenson, 2018), where smoothing is achieved in two steps: by getting rid of positions that differ by more than 5 percent from the average and subsequently using the Kalman filter. Similar solutions based on moving average, B-spline smoothing, and Kalman filter are presented in (Ardestani and Yan, 2022).

2.2 Animation Dataset Analysis

Animation of data collected by motion capture systems has a spatiotemporal nature. This data consists of poses that sample continuous movement performed by an actor at different frames (timesteps). Frames can be described in various file formats, one of which is the BVH format (Meredith and Maddock, 2001). It represents a hierarchy of joints, as well as animation details, such as framerate (FPS) and number of frames. Then it follows with the global position of a skeleton root and local rotations of all the joints in each frame. This format is one of the most popular, as it has public specifications and describes well human movements.

Such animation created from motion capture data has many applications in the modern world. It appears in movies, computer games (Geng and Yu, 2003) and in the entertainment industry (Bregler, 2007). Less recognizable applications are analyses for automatic recognition and classification of the type of human movement (Kadu and Kuo, 2014; Ijjina and Mohan, 2014; Patrona et al., 2018). This technology is also used in industry (Menolotto et al., 2020), where it is most often a component of real-time systems. Regardless of the application, the quality of the final effect largely depends on the quality of the data that constitute information about the movement sequence. Hence, a preliminary analysis is often carried out to detect potential errors and imperfections.

Even with the best motion capture data, some post-processing is usually being done. The role of data denoising is of increasing importance because the quality of animation is something that the human eye can verify instantly. It is always a non-trivial task; however, currently, the process itself becomes more and more automatic. There are several elaborate algorithms. In (Liu et al., 2014) the authors present a sophisticated, hence classical approximation algorithm. Holden in (Holden, 2018) uses an innovative method based on neural networks to map the positions and rotations of an animated skeleton based on the raw positions of markers captured by a motion capture system.

Some well-known utility programs that process animation data include Matlab MoCap Tool-

box (Burger and Toiviainen, 2013) and RMoCap (Hachaj and Ogiela, 2020). The first is a set of Matlab functions for visualization and statistical analysis (e.g., calculation of mean, standard deviation) of various metrics in motion capture data (e.g., velocity, acceleration). The toolbox is still being maintained and developed. It also allows us to perform Principal Component Analysis (PCA) on the animation sequence to derive complexity-related movement features. The second solution uses the R programming language package with a similar purpose, i.e., to visualize, analyze, and perform statistical processing. It allows motion correction to reduce foot skating and motion averaging to remove random errors, provided that a motion has been recorded multiple times. The solution also includes utility for conversions between hierarchical and direct kinematic models.

Other work focused on comparing data from motion capture for particular purposes. Four metrics were proposed to distinguish the motion of the hand of a subject who suffered cerebral palsy from the regular movement of the hand (Montes et al., 2014). Those metrics are logarithmic dimensionless jerk, mean arrest period ratio, peaks metric, and spectral arc length, among which the latter achieved the best results. Another piece of research compared various motion capture systems by collecting treadmill walking animations (Manns et al., 2016). The authors performed the analysis using PCA and calculated Shannon entropy to compare four different systems: three marker-based and one markerless. This analysis was correlated with the fact that the markerless solution gives a lower quality of captured animation than other solutions.

Recently, a new framework for analysis of locomotive datasets has been proposed that could be used, e.g., for the motion matching approach (Abichequer Sangalli et al., 2022). It focuses on the coverage of linear and angular speeds of animated characters, frame usage frequency, planned locomotion path, used and unused animations, transition cost, and number. The solution also incorporates visualization of linear and angular speed coverage across the animation dataset. It allows for identifying types of transitions in motion matching that may be difficult to perform due to the lack of animation sequences with corresponding speeds.

2.3 Anomaly Detection in Time Series

Jitter detection in animation can also be viewed as an anomaly detection problem applied to time series. Among the commonly used methods for this purpose are PCA, Savitzky-Golay, and Kalman filters and recurrent neural networks, such as those based on LSTM and GRU layers. As qualitative artifact detection in animation is a rarely addressed topic, we would like to briefly present approaches used in multivariate time series anomaly detection problems related to other fields. Many learning-based approaches suffer from inaccuracies at an early stage due to initialization time. An outlier-resistant sampling can be used in conjunction with domain-specific clustering algorithms to mitigate such an issue in online services systems (Ma et al., 2021). Recent literature also uses a weighted hybrid algorithm that combines multiple methods (e.g., moving median, Kalman filter, Savitzky-Golay filter) for anomaly detection in long-term cloud resource usage planning (Nawrocki and Sus, 2022). Moreover, deep ensemble models were successfully applied in intrusion monitoring applications and fraud detection (Iqbal et al., 2024).

3 DATASET ANALYSIS METHOD

3.1 Definition of Movement Dynamics Clutter (MDC) Metric

Animation sequences based on data collected by motion capture systems allow modeling movement in great detail. However, this data can also contain some artifacts that impact the realism of the actions captured by the actor. An example of such errors is jitter, which can be described as a rapid and chaotic movement of a joint. The magnitude of the noise introduced by this phenomenon can vary from barely noticeable to very disruptive. As a result, on many occasions, rotations of joints change rapidly, and the skeleton's bones twist unnaturally. Although there are some automatic ways to mitigate this problem (e.g., applying Savitzky-Golay smoothing), it is often the case that animators need to perform a manual review or statistical analysis to take care of the issue.

To address this inconvenience, we decided to design a metric that can be used to detect jitter and compare animation sequences and datasets in terms of the presence of jitter. In order to achieve that, this metric also needs to consider various technical aspects related to this data. The spatiotemporal data is multidimensional, which makes it hard to analyze quickly without any aggregation. The structure of a skeleton can also be different between datasets, which translates to different numbers of joints and their hierarchy. Two animation sequences from two datasets could vary in terms of recording framerate. Finally, the scale of the skeleton can differ between datasets, as most animation formats define poses using simply

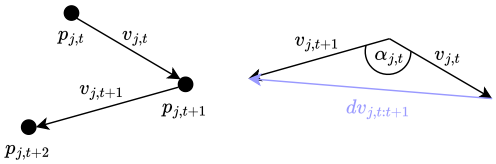

Figure 1: A visualization of MDC metric calculation method. Positions of joint j in the 3 consecutive frames are indicated as $p_{j,t}$, $p_{j,t+1}$ and $p_{j,t+2}$. These positions are used to derive velocities $v_{j,t}$ and $v_{j,t+1}$ to calculate length of $dv_{j,t:t+1}$ and angle $\alpha_{j,t}$, which are core components of the suggested metric.

abstract floats. All these problems must be considered when designing a metric that allows us to compare any two animation sequences.

Therefore, we suggest a Movement Dynamics Clutter (MDC) metric that considers all the mentioned technical aspects. The definition of calculation for a single joint can be seen in Equation 1. We identified that the core nature of the jitter is chaotic changes in both the velocity of a joint ($v_{j,t}$, $v_{j,t+1}$) and the angle of these velocities ($\alpha_{j,t}$) between consecutive frames t and $t+1$ (illustrated in Figure 1). The angle is defined in radians and squared to smooth the metric in case of minor changes and amplify significant changes. This information can be derived from most animation formats, as all that is needed are the global positions of all the joints. We also normalize the metric by multiplying by FPS (F) and dividing by the sum of the lengths of all the bones of the skeleton (S). These two operations allow us to standardize the calculated values concerning the time and space dimensions of the datasets. Normalization of space dimensions could also be achieved by calculating the height of the skeleton instead of the sum of the bone lengths. However, this proved to be cumbersome without manual work, as the rest of the poses of skeletons in MoCap datasets are not always T-poses or A-poses, making it hard to calculate the distance from feet to head.

$$MDC_{j,t} = \frac{F}{S}\alpha_{j,t}^2 ||dv_{j,t:t+1}|| \tag{1}$$

The metric can also be generalized for the calculation at the frame level by simply aggregating the values from all joints using the maximum (Equation 2); as for a frame, the crucial information is whether there is jitter in any of the joints. Although this loses some detail, it allows for easy, high-dimensional data analysis. Moreover, the metric can also be averaged over consecutive frames W (a window) to further aggregate the MDC metric (Equation 3) and, e.g., apply it to the whole dataset.

$$MDC_t = \max_{j \in J} MDC_{j,t} \tag{2}$$

$$MDC_W = \frac{1}{|W|}\sum_{t=1}^{|W|} MDC_t \tag{3}$$

3.2 Movement Dynamics Clutter Spectrum Strength (MDCSS) Metric

While the proposed MDC metric can detect chaotic movement successfully, it does not distinguish it from a single rapid joint movement. An example of such a rapid movement might be a dynamic stomp when a character's foot bounces off the floor. As we sample continuous movement, the change in the velocity vectors between consecutive frames could be angled to almost 180°. According to Equation 1, this results in a very high metric value for the joint and, consequently, for the entire frame. Our analysis indicated this problematic scenario when comparing the regular animation sequence from the LAFAN1 dataset with the jittery sequence from Contemporary Dance (Aristidou et al., 2017). The dynamic stomp of the character in the first sequence achieves an indistinguishable value of the MDC metric from the jitter that occurs on the left hand of the second animated character, which is demonstrated in Figure 2. The highlighted window refers to these examples, and as can be seen, the MDC metric has an even higher value for stomp than for jitter.

To address this shortcoming, we decided to treat our metric as a signal and look for periodicity in it. We achieve this by calculating the discrete Fourier transform (DFT) using the FFT algorithm on a sliding time window W, which consists of multiple consecutive frames. For most of our experiments, we used a window that reflected a third of a second, as it gives a satisfying sample size for all the common FPS configurations in motion capture animation sequences.

The actual values of DFT represent the energy distribution between frequencies, so this mechanism is expected to differentiate the problematic cases. In regular movement, we do not expect a peak of energy on any particular frequency, as our MDC metric usually has low values. On the other hand, when jitter occurs, we could expect that some frequencies will dominate in terms of energy value, since jitter causes regular spikes in the MDC metric.

Therefore, as our second metric for jitter analysis, we use a maximum of real DFT components, omitting the base frequency component (since it is just a sum of the values of the signal samples). We call that metric Movement Dynamics Clutter Spectrum Strength (MDCSS). Assuming that the sequence of values of

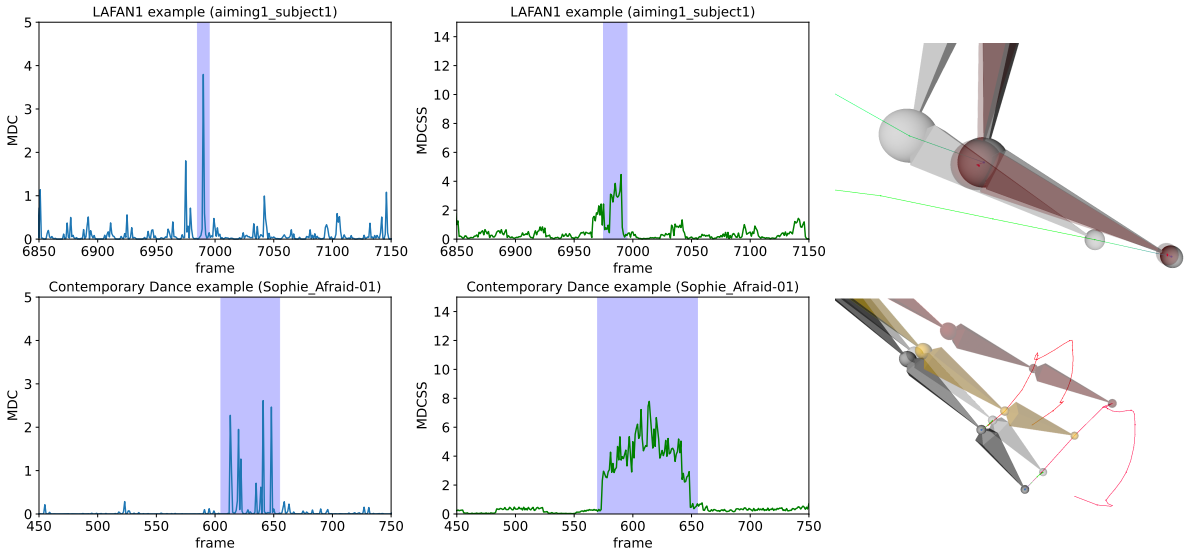

Figure 2: Comparison of MDC (blue) and MDCSS (green) metrics on dynamic stomp in LAFAN1 example (aiming1_subject1) and short jitter at the left hand in Contemporary Dance (Sophie_Afraid-01). The animation of these examples is visualized on the right using joint colors. The gray frames are preceding the event, while the colored frames are related to the event. For LAFAN1 (30 FPS), visualization has a step of 1 frame. For Contemporary Dance, the preceding frames are 12 frames apart, while the event frames are two frames apart. The trajectories of the joints are also visualized using color lines (green - preceding trajectory, red - following trajectory), with points indicating joint locations in subsequent frames.

our MDC metric in window W could be defined as MDC_W (according to Equation 4), MDCSS metric can be expressed as in Equation 5.

$$MDC_W = \{MDC_t\}_{t=0,1,\ldots,|W|-1} \tag{4}$$

$$MDCSS_W = \max_{k=1}^{|W|-1} \left\{ Re\left(\sum_{t=0}^{|W|-1} MDC_t e^{\frac{-i2\pi kt}{|W|}} \right) \right\} \tag{5}$$

With the MDCSS metric defined in this manner, we empirically assigned metric thresholds so that the characteristics of the movement in the window could be categorized. A window with value $FM_W < 8$ can be considered a window with regular movement, without any significant jitter artifacts or extraordinary dynamics. Beyond that value, we observed some instances of the artifact and many rapid movements performed by the character. A window with $FM_W \in [8,20]$ is at the warning level. Such a fragment of an animation is likely to contain some dynamic move or, on some occasions, could contain some less chaotic instances of jitter. It is advised to inspect such parts of an animation sequence. Lastly, any window with $FM_W > 20$ should be treated as one with erroneous data with jitter. This threshold was determined by inspecting the most dynamic joint moves in the tested datasets, which were not jitter.

3.3 Dataset Analysis Methodology

By using both metrics in conjunction, we can analyze animation data at various levels of detail: dataset level, animation sequence level, and frame level. As the metrics are based solely on the global positions of joints, they can be applied to most formats of animation data.

At the dataset level, the MDC and MDCSS metrics can be averaged with respect to the total number of frames. This statistic can be used to quantify the jitter and unpredictability of the dataset. Properly calculated values can then be used to compare different datasets, independently of the joint hierarchy, framerate, or scale used for the skeleton representing the animated character.

The metrics can also be used at the dataset level, i.e., by averaging each animation sequence. This method allows pinpointing the sequence that is an outlier when it comes to quality. Such problematic animation could be, e.g., re-recorded while avoiding the issue that caused the jitter or fixed in the post-processing stage. Moreover, this approach can be used to compare jitter in animation sequences from different datasets using a single number.

In some cases, excluding entire animation sequences would result in having a small dataset. Here, frame-level analysis of the MDCSS metric of animation sequences can be used. The metric calculated

on a sliding window makes distinguishing parts with jitter from those without it possible. The jittery parts (error level) could then be filtered out from the dataset instead of removing the whole sequence, if such a solution fits the purpose of use. On the other hand, the parts at a warning level with dynamic movement should be inspected, provided that they are detected in an unexpected fragment of animation (e.g., a non-dynamic one). This warning may indicate a slightly noticeable jitter there. Furthermore, frame-level analysis can also be used to compare two animation sequences from two different datasets in greater detail than using a single number (as in Figure 2).

Finally, the MDC metric allows us to perform the jitter inspection at the joint level in a single frame. If a given window has a high value of the MDCSS metric, we can inspect the MDC metric values in that window and find the frames with values that are outliers. In these frames, we can inspect the distribution of MDC values across the skeleton's joints. It allows for the selection of the subset of joints that cause a high metric value and are likely to be jittery. This level of analysis helps when analyzing warning levels of the MDCSS metric. It also allowed us to detect the subsampling problem in the case of a stomp in one of the animation sequences in the LAFAN1 dataset (Figure 2).

4 EXPERIMENTS AND DISCUSSION

4.1 Popular Evaluated Datasets

For the evaluation of the suggested metrics, we decided to use several popular datasets. Their quantitative details are presented in Table 1.

The LAFAN1 (Harvey et al., 2020) consists of animation sequences captured by five actors in a production-grade motion capture studio in collaboration with Ubisoft. The subjects perform various actions, such as moving and aiming weapons, walking, running, fighting, or navigating obstacles. The authors have published the dataset in the BVH file format. It has been used in various research publications, most commonly to test the performance of neural networks used for motion in-betweening (Qin et al., 2022; Ren et al., 2023; Oreshkin et al., 2024).

Human3.6M (Ionescu et al., 2014) is a dataset dedicated to the estimation of human poses, but it also contains animation data captured from motion capture to match these poses. The shared animation part of the dataset was captured at 50 FPS by seven actors performing different actions (e.g., walking, phoning, eating). The data is available as CDF (Common Data Format) files in various parameterization formats. We exported the animation in BVH format based on pose data with angles and subsampled it to 25 FPS, as this setup was also used to benchmark results for machine learning models in research (Harvey et al., 2020).

Two more datasets were collected by Bandai Namco Research (Kobayashi et al., 2023) that contain short animation sequences performed in various styles. The first dataset consists of 17 activities (e.g., fighting, dancing, waving hands) recorded in 15 styles. The other has 10 action types (mostly locomotion and hand actions) performed in 7 styles but in much larger quantities. Both datasets were published in the BVH animation format.

Contemporary Dance (Aristidou et al., 2017) is another dataset that contains dances performed by nine actors in various moods (e.g., bored, afraid, angry, or relaxed). The dataset is also a part of a much larger AMASS dataset (Mahmood et al., 2019). It was shared in various formats (BVH, FBX, C3D) to increase accessibility, but the BVH file format contains some very noticeable instances of jitter. We used this animation format, as the animation quality is much worse compared to other datasets.

4.2 Experiments Setup

All the metric evaluation experiments were performed on a machine with AMD Ryzen 9 5950X, 64 GB DDR4 RAM, and RTX 3080 TI. Due to large volumes of data, we calculated metrics using PyTorch with GPU acceleration. We used Python 3.10.14 for experiments, along with PyTorch 2.4.0 and NumPy 1.26.4. We derived the positions of joints based on the BVH format of animation using forward kinematics. All animation and motion path visualizations were created using Blender 4.2.2. We used a time window of $\frac{1}{3}s$ when calculating the MCDSS metric in the experiments and the same thresholds as defined in Section 3.2.

4.3 Comparison of Popular Datasets

We benchmark our metrics by evaluating them on all the datasets mentioned in Section 4.1. We perform dataset-level analysis to calculate the average metrics values for these datasets and count the number of errors and warnings detected. We also count the number of error and warning windows as a chain of consecutive frames classified similarly. These windows could be used to count the number of instances of jitter or dynamic movement.

The experiment outcome was collected in Table 2 and Table 3. As expected, Contemporary Dance has

Table 1: Comparison of datasets' properties used for the experiments.

Dataset	Joints	FPS	Animation sequences	Total frames	Total time [s]
LAFAN1	22	30	77	496,672	16,556
Human3.6M	25	25	210	263,743	10,550
Bandai Namco Dataset 1	22	30	175	36,665	1,222
Bandai Namco Dataset 2	22	30	2,902	384,931	12,831
Contemporary Dance	31	120	133	989,150	8,242

the highest metrics scores due to multiple jitter instances. The average value of the MDCSS metric strongly deviates from the other datasets and suggests that every frame in that dataset is at a warning level. The large numbers of error and warning frames and windows further signify it. Bandai Namco Dataset 2 achieves the best results, as most animations there do not contain much character movement and are not as dynamic as some sequences in other datasets. Both Bandai Namco datasets do not contain errors according to our MDCSS metric.

Table 2: Average metrics values per frame for tested datasets.

Dataset	MDC	MDCSS
Contemporary	0.408	8.109
LAFAN1	0.113	0.462
Bandai 1	0.106	0.431
Bandai 2	0.084	0.409
Human3.6M	0.099	0.396

Table 3: Number of error and warning frames (f) and windows (w) in the compared datasets.

Dataset	Errors (f/w)	Warnings (f/w)
Contemporary	85372/7543	77892/17563
LAFAN1	16/6	206/140
Bandai 1	0/0	45/43
Bandai 2	0/0	469/458
Human3.6M	21/17	355/205

All the datasets, except for Contemporary Dance, achieve relatively similar average values for our metrics. However, in both LAFAN1 and Human3.6M, our metrics detected errors. We conducted further investigation using frame-level analysis and found that both datasets contain short instances of jitter on these error windows. Human3.6M contains very short errors in many animation sequences (e.g., in the sequence "Sitting Down" performed by subject 5) and looks like short occlusions. Artifacts are most commonly found in the hands. In the case of LAFAN1, all the errors are present in a single animation sequence, "Obstacles 5," performed by subject 3 (obstalces5_subject3). The sequence contains dynamic maneuvers, such as running, jumping, and tripping over. In some instances, it looks like the motion capture system mismatched the position of all the joints, resulting in a visible discontinuity and jumps in the motion path of the character. On other occasions, it again looks like the reason is occlusion. Two warnings are also detected in this sequence: a barely noticeable short foot jitter and a dynamic landing from a jump.

We decided to manually remove the instances of jitter at error levels from the LAFAN1 example by applying interpolation between keyframes to compare the metrics' values with and without artifacts in the animation. While the manual adjustment of the positions of character joints seemed to give a more realistic result, we would like to disclose that we are not professional animators, and the actual realism of the fixed animation could be further improved. The main objective of the adjustment was to remove the jitter from the sequence, which was achieved. The output of the frame-level analysis of this animation clip is presented in Figure 3. We can see that the application of changes causes our MDCSS metric to return to the normal range of values. This reinforces the usability of our framework in detecting instances of jitter in animation sequences.

4.4 Detecting Artificial Jitter in LAFAN1 Dataset

To evaluate our MDCSS metric comparatively, we decided to benchmark it on the detection of artificial jitter added to a subset of the LAFAN1 dataset. We are unaware of any other research that focused on detecting this type of artifact, so we chose to compare against some commonly used methods for anomaly detection in multidimensional time series. We selected the PCA and Savitzky-Golay filter to serve as a baseline for comparison.

The methods were benchmarked on animation sequences performed by subject 3, as real jitter is also present. Artificial jitter was added to all the animation sequences except the one that contains actual instances of the artifact. This was achieved by corrupting every 10-second window of the animation with a 0.25-1s jittery fragment applied to a single skeleton joint. The displacement of the joint loca-

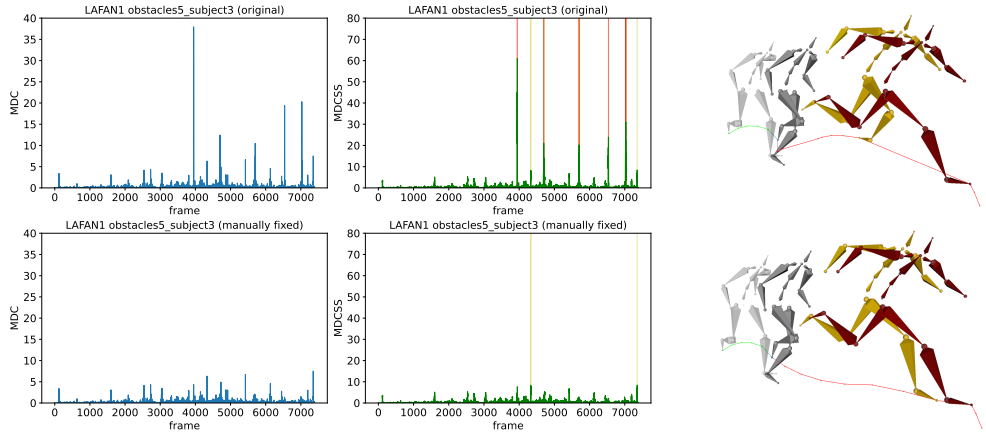

Figure 3: Comparison of metrics before and after manually attempting to fix animation sequence obstacles5_subject3 from LAFAN1 dataset. The yellow and red zones on the plots correspond to warning and error event intervals. The visualization on the right shows a jitter instance around frame 4700. Yellow and red frames are one frame apart. In the original sequence, the left leg of the character is unnaturally twisted in the yellow frame and makes a sudden correction in the red frame. The fixed version does not have this artifact. Some preceding frames with jitter have been omitted to improve visibility in the top example.

tion is added to all its coordinates according to the normal distribution $\mathcal{N}(0,\sigma)$. We parameterized σ using skeleton length and tested the following values of σ: 0.02S, 0.0175S, 0.015S, 0.0125S, 0.01S, 0.0075S and 0.005S. This methodology contains a simplification, as displacement of the joint location does not preserve its length. The sequence with real jitter was not altered in any way to benchmark the methods in a real-life scenario.

The detection was carried out on 10-frame clips. A jittery window was classified as detected if a 10-frame clip intersects it, and the method labeled it as anomalous. A normal clip was defined as one that does not intersect any jitter window.

Our MDCSS metric has a naturally defined error level, so we used that to classify anomalous clips. We decided on 22 principal components while performing PCA and using the quadratic reconstruction error as the detection metric. Dimensionality reduction is performed on the joint acceleration vectors in each frame (reduction of 660-dimensional data), and we extract the maximum reconstruction error over all joints and clip frames. This offers an opportunity to pinpoint the joint that caused the jitter, similar to what our MDC metric allows. The clip is classified as anomalous when the reconstruction error exceeds 1.5 times the maximum seen in the training dataset. We used the rest of the LAFAN1 dataset as training data to fit the PCA model to regular data. Finally, the Savitzky-Golay filter was configured to analyze 10 frames and smooth using second-degree polynomials. We use the same jitter detection method as for PCA, setting the error threshold to 5.5. All acceleration vectors were normalized by skeleton scale and framerate, similarly to Equation 1.

The results of the experiment are averaged over 10 repetitions. All the methods managed to detect the real jitter instances with the described configurations. We present a comparison of the F1 score from the experiment in Table 4, as well as the recall and precision in Table 5. Our metric remains quite comparable in terms of the F1 score and even outperforms other methods on most values of σ, although only slightly. Further inspection reveals that precision is the strong side of our MDCSS metric while remaining only slightly behind when it comes to recall. In terms of recall, the Savitzky-Golay filter is the best for most noise levels. We also performed frame-level analysis to visualize the behavior of MDCSS metric on a single example from that dataset. Figure 4 presents the sample animation sequence for $\sigma = 0.0125S$, which shows that spikes in our metric align with artificial jitter windows.

Table 4: F1 score of compared methods on LAFAN1 dataset for different values of σ in $\mathcal{N}(0,\sigma)$. Best results are highlighted in bold.

σ	MDCSS	PCA	Savitzky-Golay
0.02S	**0.832**	0.778	0.778
0.0175S	**0.852**	0.824	0.823
0.015S	0.857	**0.870**	0.868
0.0125S	0.823	0.883	**0.884**
0.01S	0.683	**0.723**	0.714
0.0075S	**0.332**	0.200	0.189
0.005S	**0.012**	0.000	0.001

Table 5: Precision and recall of compared methods on LAFAN1 dataset for different values of σ in $\mathcal{N}(0,\sigma)$. Best results are highlighted in bold.

	Recall			Precision		
σ	MDCSS	PCA	Savitzky-Golay	MDCSS	PCA	Savitzky-Golay
0.02S	0.955	0.996	**0.997**	**0.737**	0.639	0.637
0.0175S	0.927	0.990	**0.991**	**0.789**	0.705	0.704
0.015S	0.869	**0.973**	0.970	**0.846**	0.787	0.786
0.0125S	0.757	**0.884**	**0.884**	**0.904**	0.882	0.884
0.01S	0.536	**0.586**	0.574	0.942	0.945	**0.950**
0.0075S	**0.200**	0.112	0.105	**0.978**	0.974	0.969
0.005S	**0.006**	0.000	0.0004	**0.900**	0.000	0.100

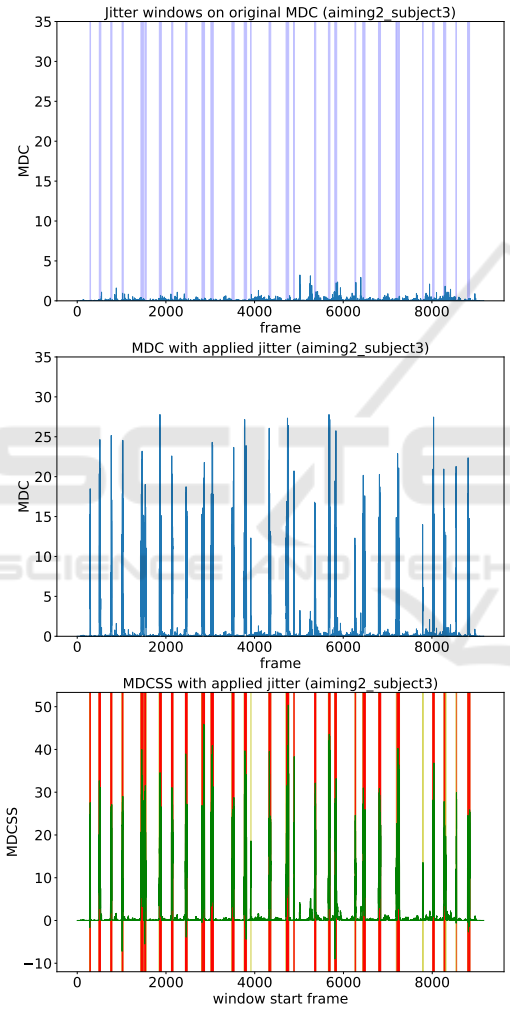

Figure 4: Metrics values for the aiming2_subject3 example from LAFAN1 dataset. Highlighted windows in the top chart indicates where artificial jitter was added according to $\mathcal{N}(0, 0.0125S)$ distribution.

4.5 Scale Invariance

We also decided to verify the invariance of the skeleton scale of the proposed metrics. To verify this property, we conducted frame-level and dataset-level analyses on the Contemporary Dance dataset. For the frame-level analysis, we used an example animation sequence called "Sophie_Afraid-01". We artificially scaled the size of the skeleton as well as the root position from source BVH files using scales x0.1 and x2.

Figure 5 presents the results of the frame-level analysis in the example for all test scales. The metric plots are indistinguishable, having the same values and warning and error windows. The dataset-level analysis presented in Table 6 further confirms this. For all given scales, the value of our metrics is the same when calculated across the whole dataset.

Table 6: Average metrics values per frame on the Contemporary Dance dataset for different skeleton scales.

Scale	MDC	MDCSS
x1 (baseline)	0.408	8.109
x0.1	0.408	8.109
x2	0.408	8.109

4.6 Analysis of Different MDCSS Time Window Lengths

To further analyze the properties of our MDCSS metric, we decided to compare how it changes for different lengths of time windows used. We achieved that by conducting frame-level analysis for two examples from LAFAN1 - aiming1_subject1 and fight1_subject2 - and one example from Contemporary Dance - Sophie_Afraid-01. The first example from LAFAN1 represents regular, non-dynamic animation, while the other example from that dataset contains plenty of dynamic punches, kicks, and jumpkicks. On the other hand, the example from Contemporary Dance is filled with jittery fragments of animation and is used to refer to jitter detection properties. We analyze the following time windows: 0.1s, 0.2s, 0.25s, 0.33s, 0.5s, 0.67s, and 1s. Since we operate on frames, we round down the number of frames in the case of FPS indivisibility.

The MDCSS metric values for these examples

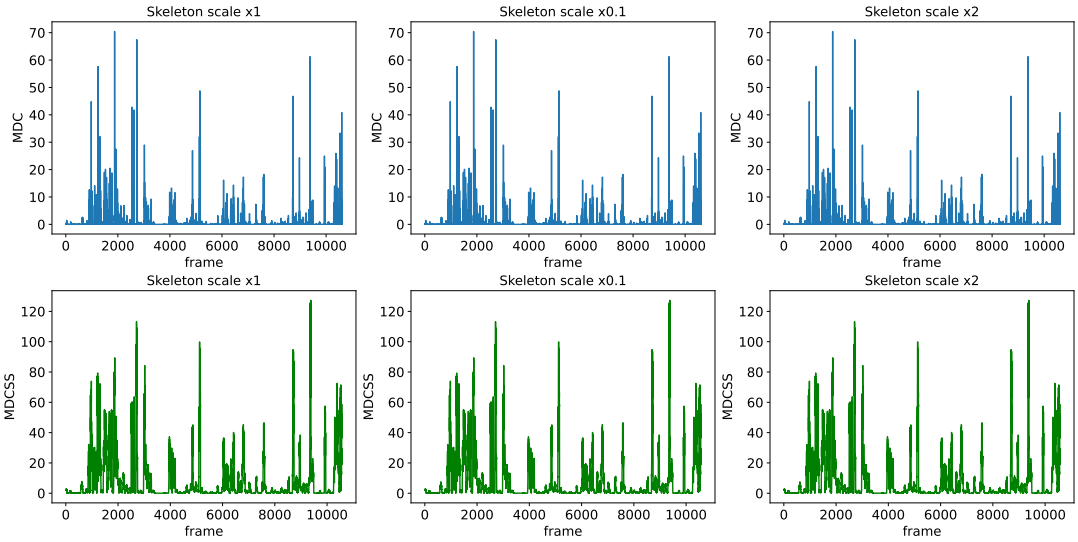

Figure 5: Comparison of metrics for Contemporary Dance example (sequence Sophie_Afraid-01) for different scales of the skeleton.

are plotted in Figure 6. Our analysis suggests that increasing the time window also magnifies the values of the metric. The values that are magnified the most are the ones that correspond to jittery or dynamic fragments. This observation makes sense because a longer time window contains more samples, and therefore, more energy to be distributed. In the case of a jitter, more energy might accumulate under the same frequency, magnifying the value of the MDCSS metric.

4.7 Limitations

Our current definition of the animation analysis method has some limitations. The most important limitation of our metrics is that a jitter that does not change the core trajectory of the joint will probably not be detected. Although this limits the detection capabilities of our metrics, it also reduces the visibility of such a jitter.

Our metric also achieves the best values if all character's joints start moving in the same direction. An example of a scenario is a character in a T-pose with the root joint moving in a straight line. Some animation formats also allow for the rotation of a bone around its vertical axis. Such a scenario would also not be detected by our metrics, as it operates solely on the positions of joints, not the rotation of bones. While these shortcomings of our metrics are concerning, they are unlikely to occur in most motion capture animation sequences.

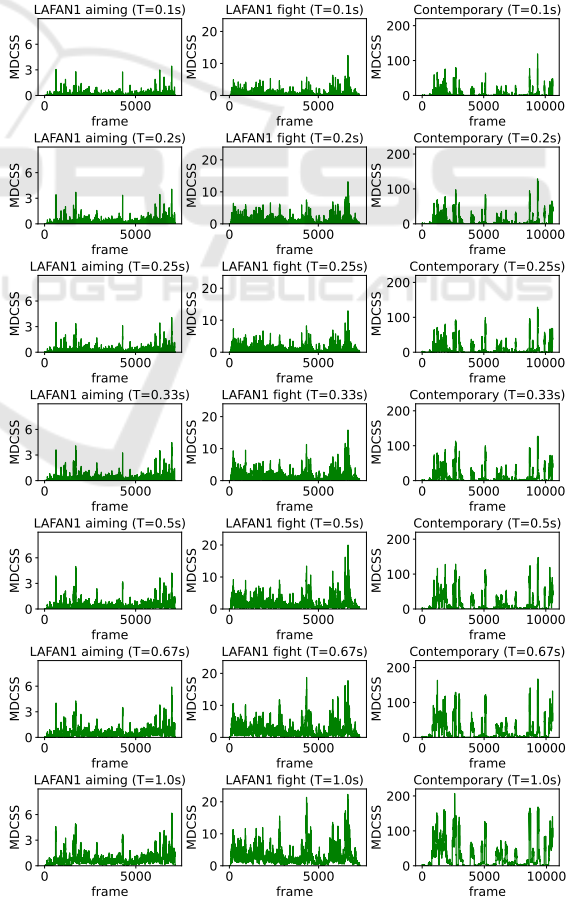

Figure 6: Comparison of MDCSS values calculated on 3 chosen animation sequences. Each sequence is represented as a column and each row represents a different length of time window.

5 DISCUSSION

As the results of the method comparison suggest, our framework proves to be quite comparable with baselines while requiring no fitting to the dataset, in contrast to PCA. Moreover, the PCA method requires a constant number of input variables, which makes it unfit for working on multiple skeletons with various hierarchies. The MDCSS metric also proved its usefulness in real cases, as we managed to detect undiscovered jitter in the LAFAN1 dataset. This dataset is widely used in research to evaluate neural models. The found jitter instances are located in the training part of the dataset and negatively influence the training process of such models, thus affecting the output result used in the evaluation. We think that our framework could be used to perform preprocessing checks on motion capture datasets and filter out in such applications. The proposed metrics consider problematic technical aspects, which should make such integration relatively easy.

6 FUTURE WORK

In our subsequent work, we would like to address some of the limitations of our work, especially the problem of detecting jitter that does not change the joint trajectory. We identified that this scenario is problematic for our analysis framework. Addressing this issue could greatly increase the number of detected jitter instances. One possible solution is to monitor the rotations of the bones instead of the positions of the joints. While this approach is likely to address this problem, it would restrict applications of our framework, as such data is not always available for animation sequences. Instead, we would like to try to solve this problem by inspecting the characteristics of the trajectory. We also aim to experiment with automatic correction using smoothing algorithms and neural models in sequences classified as errors by our metrics.

In addition, we plan to evaluate the proposed metrics against the performance of neural models to find a correlation between the value of the metric and the performance of state-of-the-art neural models. This work will focus mainly on the warning level of our MDCSS metric, as we hypothesize that such sequences contain more irregular joint movements and could be harder to predict by neural networks. This could potentially help us to understand the shortcomings and problematic animation sequences of current research related to machine learning applications in frame generation.

7 CONCLUSIONS

In this work, we proposed a novel framework for jitter detection in animation datasets that consists of two metrics: MDC and MDCSS. The framework can operate at multiple levels of detail and allows datasets to be compared with different skeleton scales, numbers of joints, and FPS. We also evaluated this framework on several popular datasets to prove its usefulness. Our experiments found that two of the popular animation datasets, LAFAN1 and Human3.6M, contain instances of jitter, which was not known before. This further emphasizes the need for jitter detection frameworks in professional motion capture environments, such as the one proposed in this work. The comparison with commonly used anomaly detection methods also proved that our framework is well-suited for this task requiring no adjustments or fitting to animation data. We hope that our metrics can contribute in the future to cleaning motion capture datasets used for machine learning purposes, as well as improving the overall quality of animation by detecting jittery sequences early in motion capture recordings.

ACKNOWLEDGEMENTS

The research presented in this article was partially supported by the funds of the Polish Ministry of Science and Higher Education, assigned to the AGH University of Krakow (Faculty of Computer Science).

REFERENCES

Abichequer Sangalli, V., Hoyet, L., Christie, M., and Pettré, J. (2022). A new framework for the evaluation of locomotive motion datasets through motion matching techniques. In *Proceedings of the 15th ACM SIGGRAPH Conference on Motion, Interaction and Games*, MIG '22, New York, NY, USA. Association for Computing Machinery.

Ardestani, M. M. and Yan, H. (2022). Noise reduction in human motion-captured signals for computer animation based on b-spline filtering. *Sensors (Basel, Switzerland)*, 22(12).

Aristidou, A., Zeng, Q., Stavrakis, E., Yin, K., Cohen-Or, D., Chrysanthou, Y., and Chen, B. (2017). Emotion control of unstructured dance movements. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, SCA '17, New York, NY, USA. Association for Computing Machinery.

Bregler, C. (2007). Motion capture technology for entertainment [in the spotlight]. *IEEE Signal Processing Magazine*, 24(6):160–158.

Burger, B. and Toiviainen, P. (2013). MoCap Toolbox – A Matlab toolbox for computational analysis of movement data. In Bresin, R., editor, *Proceedings of the 10th Sound and Music Computing Conference*, pages 172–178, Stockholm, Sweden. KTH Royal Institute of Technology.

Callejas-Cuervo, M., Espitia-Mora, L. A., and Vélez-Guerrero, M. A. (2023). Review of optical and inertial technologies for lower body motion capture. *Journal of Hunan University Natural Sciences*, 50(6).

Geng, W. and Yu, G. (2003). Reuse of motion capture data in animation: A review. In *International Conference on Computational Science and Its Applications*, pages 620–629. Springer.

Hachaj, T. and Ogiela, M. (2020). Rmocap: an r language package for processing and kinematic analyzing motion capture data. *Multimedia Systems*, 26.

Harvey, F. G., Yurick, M., Nowrouzezahrai, D., and Pal, C. (2020). Robust motion in-betweening. *ACM Trans. Graph.*, 39(4).

Holden, D. (2018). Robust solving of optical motion capture data by denoising. *ACM Transactions on Graphics (TOG)*, 37(7):1–12.

Hoxey, T. and Stephenson, I. (2018). Smoothing noisy skeleton data in real time. In *EG 2018 - Posters*. The Eurographics Association.

Ijjina, E. P. and Mohan, C. K. (2014). Human action recognition based on mocap information using convolution neural networks. In *2014 13th International Conference on Machine Learning and Applications*, pages 159–164.

Ionescu, C., Papava, D., Olaru, V., and Sminchisescu, C. (2014). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339.

Iqbal, A., Amin, R., Alsubaei, F. S., and Alzahrani, A. (2024). Anomaly detection in multivariate time series data using deep ensemble models. *PLOS ONE*, 19(6):1–25.

Kadu, H. and Kuo, C.-C. J. (2014). Automatic human mocap data classification. *IEEE Transactions on Multimedia*, 16(8):2191–2202.

Kobayashi, M., Liao, C.-C., Inoue, K., Yojima, S., and Takahashi, M. (2023). Motion capture dataset for practical use of ai-based motion editing and stylization.

Liu, X., Ming Cheung, Y., Peng, S.-J., Cui, Z., Zhong, B., and Du, J.-X. (2014). Automatic motion capture data denoising via filtered subspace clustering and low rank matrix approximation. *Signal Process.*, 105:350–362.

Ma, M., Zhang, S., Chen, J., Xu, J., Li, H., Lin, Y., Nie, X., Zhou, B., Wang, Y., and Pei, D. (2021). JumpStarting multivariate time series anomaly detection for online service systems. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 413–426. USENIX Association.

Mahmood, N., Ghorbani, N., Troje, N. F., Pons-Moll, G., and Black, M. J. (2019). AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451.

Manns, M., Otto, M., and Mauer, M. (2016). Measuring motion capture data quality for data driven human motion synthesis. *Procedia CIRP*, 41:945–950. Research and Innovation in Manufacturing: Key Enabling Technologies for the Factories of the Future - Proceedings of the 48th CIRP Conference on Manufacturing Systems.

Menolotto, M., Komaris, D.-S., Tedesco, S., O'Flynn, B., and Walsh, M. (2020). Motion capture technology in industrial applications: A systematic review. *Sensors*, 20(19):5687.

Meredith, M. and Maddock, S. C. (2001). Motion capture file formats explained. *Department of Computer Science, University of Sheffield*.

Montes, V. R., Quijano, Y., Chong Quero, J. E., Ayala, D. V., and Pérez Moreno, J. C. (2014). Comparison of 4 different smoothness metrics for the quantitative assessment of movement's quality in the upper limb of subjects with cerebral palsy. In *2014 Pan American Health Care Exchanges (PAHCE)*, pages 1–6.

Nawrocki, P. and Sus, W. (2022). Anomaly detection in the context of long-term cloud resource usage planning. *Knowl. Inf. Syst.*, 64(10):2689–2711.

Oreshkin, B. N., Valkanas, A., Harvey, F. G., Ménard, L.-S., Bocquelet, F., and Coates, M. J. (2024). Motion in-betweening via deep δ-interpolator. *IEEE Transactions on Visualization and Computer Graphics*, 30(8):5693–5704.

Patrona, F., Chatzitofis, A., Zarpalas, D., and Daras, P. (2018). Motion analysis: Action detection, recognition and evaluation based on motion capture data. *Pattern Recognition*, 76:612–622.

Qin, J., Zheng, Y., and Zhou, K. (2022). Motion in-betweening via two-stage transformers. *ACM Trans. Graph.*, 41(6).

Ren, T., Yu, J., Guo, S., Ma, Y., Ouyang, Y., Zeng, Z., Zhang, Y., and Qin, Y. (2023). Diverse motion in-betweening from sparse keyframes with dual posture stitching. *IEEE Transactions on Visualization & Computer Graphics*, (01):1–12.

Skurowski, P. and Pawlyta, M. (2022). Detection and classification of artifact distortions in optical motion capture sequences. *Sensors*, 22(11).