12th International Workshop on Innovative Simulation for Healthcare
20th International Multidisciplinary Modeling & Simulation Multiconference

2724-0371 © 2023 The Authors.
doi: 10.46354/i3m.2023.iwish.009

Performance-optimized Dynamic-Time-Warping Based Identification of Motion Trajectories

Jan Beckmann[1] and Volkhard Klinger[1,*]

[1]Department of Embedded Systems, University of Applied Science Hannover (FHDW), D-30173 Hannover, LS, Germany

[*]Corresponding author. Email address: volkhard.klinger@fhdw.de

Abstract

Several biomedical applications, such as posture evaluation or gait monitoring, require the identification or at least comparison of motion trajectories. In this paper, we use a model-based approach for motion trajectory identification. The approach is based on the Dynamic-Time-Warping (DTW)-algorithm. However, the DTW-algorithm is computationally expensive, so we present a performance-optimized DTW-based identification approach. This approach utilizes the Douglas-Peucker algorithm to reduce the number of points of the motion trajectories. It is evaluated using a set of motion trajectories, demonstrating that the approach is able to achieve a significant speed-up while maintaining the accuracy of the DTW-algorithm for motion identification. This makes it possible to identify motion trajectories on embedded systems with low computational power commonly used in biomedical applications.

Keywords: model-based; data-fusion; Internet of Things (IoT); platform; trajectory-identification; dynamic time-warping

1. Introduction

Embedded systems open up completely new opportunities in biotechnology and medical therapy. The integration of sensors and actuators into an adaptive hardware/software platform expands functionality to include the detection of states or the execution of actions. In this context, a wide range of applications can be found, particularly in therapy and rehabilitation, which can support conservative forms of treatment. The overall focus is on continuous data acquisition as well as online and offline data processing and, in particular, identification based on correlated sensor information and data fusion. We have introduced in (Klinger, 2016) and (Klinger, 2019) and (Klinger, 2021) several applications and methods to acquire motion data to provide a rehabilitation monitoring or a postural evaluation. All these applications were realized based on the platform with integrated Internet of Things (IoT)-modules.

In this paper we present results for the speed-up of motion-trajectory identification from a set of motion-trajectories saved in a motion-library. In focus are specific applications, described in more detail in section 2, where the motion-trajectory identification is one of the central tasks. Our objective is to integrate the whole motion trajectory identification into the IoT-modules to improve the Plug-and-Play (PnP)-factor of the system and to increase the use of peripheral intelligence, formed in the ESP32-based IoT- modules (Espressif, 2019). We present an identification algorithm, based on the dynamic time-warping method, and its speed-up and performance optimization to enable the use on IoT-modules with a limited computer performance including an optimized application-specific real-time processing.

2. Biomedical Applications and Embedded Systems

There are lots of biomedical applications existing using motion-trajectories for evaluation and/or control of body motions or movement of extremities. We describe some

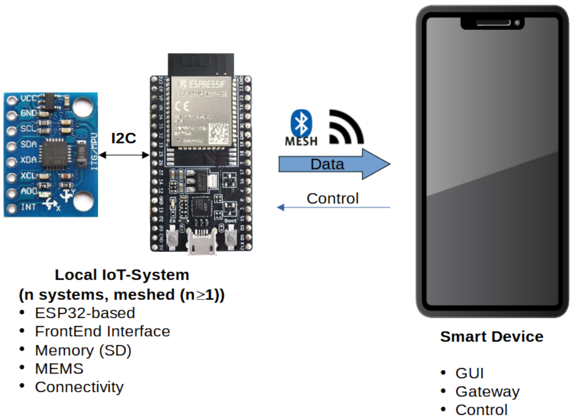

Figure 1. Low-level system architecture.

applications for prosthesis control, gait monitoring and posture evaluation, all based on our system platform including the use of IoT-modules.
The system architecture is a key factor for supporting stationary and mobile applications. Mobile applications have special requirements, for example in terms of portability, energy consumption, computing performance and integration into existing infrastructures. Therefore, the system architecture used is based on the system we have introduced in (Klinger and Bohlmann, 2020). IoT-components were added to the original system to expand the range of functions. These IoT-components enable wireless connection of the different system components to a central unit. Using the platform architecture, the different modules span their own network (by Bluetooth Low Energy (BLE) or wireless local area network according IEEE 802.11 (WiFi)) and connect for a longer range to a gateway. This gateway, i.e. a smartphone, enables integration of the system via WiFi or Global System for Mobile Communication (GSM), i.e. cellular-secured Internet. To realize a platform-based system architecture providing a mobile operation which is only person- but not place-bound, the following characteristics are essential:

C1 Independence from a specific equipment of the environment.
C2 Integration of all required sensors into the platform.

These characteristics allow the platform to be almost universally adaptable. For example, the gait evaluation based on shoe-integrated pressure sensors, described in (Klinger, 2016), have to be replaced by another sensor configuration to fit these requirements. Other current solutions, described in (Matuska et al., 2020) and (Bourahmoune et al., 2022), follow a different approach and outsource the functionality to external, non-personal equipment (chairs, cushions). This allows an application only in dedicated locations. The lowest level of the platform architecture used here, is depicted in Figure 1. The local IoT-System, called in the following *SmartBox*, consists always of an ESP32 (Espressif, 2019) and application specific sensors; here we have only connected an micro-electro-mechanical systems (MEMS)-sensor with 6-axis (acceleration (x, y, z) and gyro (x, y, z)), an MPU6050. The smartphone is connected for configuration and data exchange, for example to download all raw-data saved on the ESP32. The other levels are not in focus here, only a short description should be enough: The cloud database is the sink for all data from all local IoT-systems and the source for all data mining and data-based fusion and identification operations. This architecture integrates all IoT-systems and supports different modes and their corresponding scenarios (Klinger and Bohlmann, 2020). In all applications, we try to improve the PnP-character when attempting to fit a system mode that requires as little system complexity for measurement and ease of operation as possible. This means that the smartphone is only necessary for certain indications and events. A more advanced system mode, which allows the raw data of the measurement to be transferred at certain times from the local system to a cloud architecture and thus also to a server for evaluation, has already been presented in (Klinger and Bohlmann, 2020).

Especially for gait monitoring and posture evaluation the main task consists in the comparison of trajectories in order to identify specific movements. In the following subsections we introduce this context in more detail. In addition, we will cite related work in subsection 2.3 to better classify this paper.

2.1. Posture Evaluation

Good posture is the end product of a complex combination of mechanical, neurological and psychological factors, including muscle strength and flexibility, vision, sense of touch, balance, self-esteem, kinesthetic awareness and a well-functioning vestibular system (Goodman and Fuller, 2015). Good posture is desirable when sitting and standing, as well as when walking and running (Heidenfelder, 2011; Ito, 2008). Because of the number of parts and functions involved in good posture, a postural assessment can serve a variety of purposes, but let us here focus on problems triggered from lots of hours when working from home reducing the movement possibilities for compensation and aggravating all posture problems.
A major challenge is the calibration of an approximately individual perfect posture. What is needed is an absolute positioning, which cannot be achieved with the MEMS-sensors. Currently, the calibration is performed by a set of motion prescriptions (Krankenkasse, 2008), which are supplemented by various stretching movements of the arms and movements of the upper body and head according to fixed calibration movements. The SmartBoxes, an ESP32-based microcontroller including the required sensor systems, are shown in Figure 2; the both upper ones for posture evaluation, the both on the ankles for gait monitoring (see section 2.2) (Klinger, 2021). The central tasks are for example:

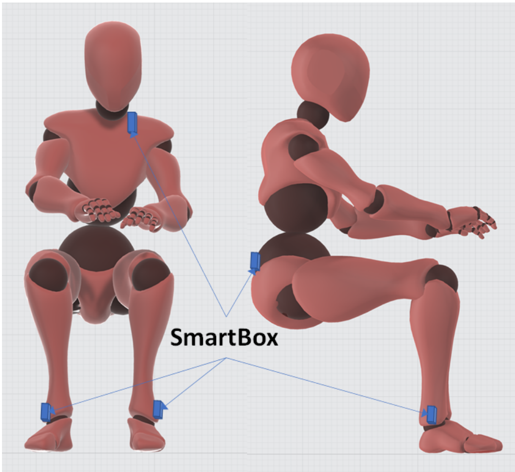

Figure 2. Position of the SmartBoxes for posture evaluation and gait monitoring.

1. Detection of prolonged rigid postures.
2. Detection of risky postures, by crossing legs, strong unilateral bending to the right left or front.
3. Sensitization of the user through recognition of personal habitual postures.
4. Suggestion of compensatory movements according to a selection based on detected postures.

For tasks 2 to 4 an identification and a library of motion trajectories is essential. As mentioned, this identification of motion trajectories is in focus, taking the reliability of the identification and the required computing power into account.

2.2. Gait Monitoring

The evaluation of the gait of apparently healthy persons is an important method to analyze an imbalance or dysfunction which can result in health problems. These problems can be evaluated using a continuous gait monitoring to identify pathological or abnormal gaits. Paying attention to how you walk and run reduces unnecessary muscle strain (Hartmann et al., 2013). In addition, this gait monitoring can be used to monitor and optimize movement sequences within the sports segment.
Our first approach, presented in (Klinger, 2016), was based on force-sensors, measuring at three different positions in the sole, shown in Figure 3. The results of this system, presented in Figure 4, provide detailed information related to the three sensor positions. The disadvantage of the system is the complexity and the integration of the sensor facility into a shoe or the addition of the appropriately equipped sole to a normal shoe, taking into account the constraint that both shoes must be equipped. So the new system will follow a much more PnP paradigm and consist only of the so-called SmartBoxes, realized as simple wearables. This means that a special shoe is no longer necessary, making

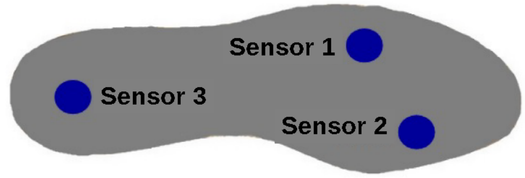

Figure 3. Positions of the force sensors in side the sole.

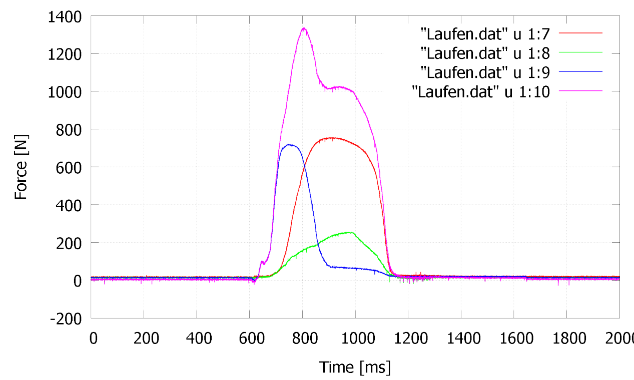

Figure 4. Normal gait, one step, right leg: Vertical Force plotted over time for 3 sensors (1 (red), 2 (green), 3 (blue), sum(1,2,3) (magenta), see Figure 3.

it much easier and more economical to use. However, this also means that only the aggregated force information is available, which is shown in Figure 4 in magenta as an envelope. Whether the information, which is measured on the basis of the 6-dimensional MEMS-sensors and transformed into forces, is sufficient, is still under testing. A new function is also in central focus: The comparison of the two force curves in both feet, which is achieved on the basis of a trajectory comparison. For this purpose, unlike in posture evaluation, no library of trajectories is necessary, but the current trajectory of one foot is used as a comparative trajectory of the other foot. The objective is to detect inequalities in the gait pattern and to use these against the background of a preventive evaluation in order to quickly detect common causes of gait disorders, for example due to neurological, orthopedic or psychological reasons. This vice-versa gait monitoring is a new project in the context of biomedical applications.

2.3. Related Work

There are several approaches to motion identification, reflecting the wide range of applications. For example, Chambers et al. provide an extensive review in (Chambers et al., 2015) of the use of wearable microsensors for detecting sport-specific movements (e.g. tennis, golf). In particular, they highlight the added value of using multiple sensors whose measurements are combined.

There are also attempts to use motion identification to help with the rehabilitation of patients, such as osteoarthritis (Huang et al., 2017). The cited authors use support vector machines as a way to detect various movements of the knee joint and identify rehabilitation exercises. While the data is measured with a wearable sensor, the classification is performed on a desktop computer (running MATLAB (TheMathWorks,Inc., 2023)). This makes the approach unsuitable for identifying movements in near real-time.

The presented approach in this paper aims at making such an identification possible instantly on a microcontroller, or IoT-System. This requires a significant reduction in the computational effort without losing accuracy, making it possible to deliver instant feedback to the user without the need for a desktop computer or a time delay.

3. Identification of motion trajectory

Movements can be performed at different speeds, i.e. the trajectory does not change, only the time required for movement. From this arises for the task here the requirement to identify two trajectories consisting of time series of positions as the same, even if they differ in their temporal course.
The 6-axis information of the MEMS-modules must be converted into a time series of positions and an orientation, the so-called quaternions. Subsequently, the comparison of two trajectories, for example the trajectory of the movement just detected, with one trajectory each from a comparison library is performed. The comparison is done with the method of dynamic time-warping, which will be described in the following section.
We subsequently evaluate the optimization potential related to our requirements in section 3.2.

3.1. Dynamic Time-warping trajectory identification

The Dynamic-Time-Warping (DTW)-algorithm (Jablonski, 2012; Srivastava and Sinha, 2016) is used to check sequences of values of different lengths in a pattern match for matches. For this purpose, the dynamic time normalization algorithm generates warping paths by which matches can be detected by backtracking and difference measures despite time distortion or different speeds. The eponymous "warping" of sequences allows us to detect commonalities and matching patterns between sequences even when they differ in length or speed. We use the algorithm to lend captured motion trajectories with motion trajectories from a library.

For example, if two different walking sequences are to be examined for matches (gait monitoring, see section 2.2), the algorithm is able to detect identical patterns even if the walking speed or distance traveled is different.

In addition, pattern measurement and pattern recognition in value series can be used to examine similar system developments even over different time periods. For this reason, the DTW-algorithm is also used in forward-looking technologies such as Machine Learning to train the analysis and reaction capabilities of self-learning systems and to evaluate data sets more efficiently. Different rules and conditions are applied:

- Each value of a sequence must be compared with one or more values of the second sequence (and vice versa).
- The first value of a sequence must be compared with the first value of the second sequence.
- The last value of a sequence must be compared with the last value of the second sequence.
- The mapping of the value series of the first sequence to the value series of the second sequence must increase monotonically. Values at the beginning and end of the sequences must therefore match in their positions, without omission or overlap.

More and more algorithms could be accelerated by several orders of magnitude by implementing them on GPUs. Here, the application on an ESP32 is in focus to identify acquired motion-trajectories locally to get an advantage from the use of peripheral intelligence, so speed-up by parallelization is not appropriate.

3.2. Performance Optimization

The DTW-algorithm compares two trajectories based on the samples that form the respective target trajectory. DTW gives a non-linear (elastic) alignment. The complexity of computing DTW is $\mathcal{O}(m \cdot n)$ where m and n are representing the length of both sequences.
In order to run the algorithm on a module with low computing power, strategies have to be found to reduce the computational effort. Thus, the objective must be to reduce the number of descriptive points of a trajectory. The number of points of a trajectory depends on the sampling frequency. In order to be able to describe a movement sufficiently accurately, the sampling rate must be at least 30 Hz up to 100 Hz for the movements in focus here. Reducing the sampling frequency means an equidistant reduction and thus no information-related reduction of the sampling points. What does this mean? Depending on the motion being performed, sampling points may or may not contain redundant information. For a straight line, only two points are necessary, all intermediate points are practically redundant. This does not apply for curve that is not modeled on a ideal circular path. Here, many more sample points are necessary to be able to map the course accordingly.
The Douglas-Peucker algorithm (Douglas and Peucker, 1973; Winnepenninckx, 2016) is a curve smoothing algorithm in the field of vector graphics. The objective is to simplify a line given by a sequence of points by omitting individual points (weeding) in such a way that the rough shape is preserved. The degree of coarsening is controlled by specifying the maximum distance between the original points and the approximating line.
This algorithm is well suited for this application, since it omits individual points by preserving other ones, so it reduces the runtime complexity of the DTW-algorithm.

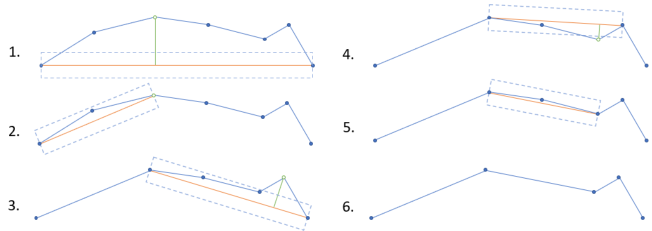

Figure 5. Douglas-Peucker-algorithm: Reduction of descriptive non-redundant points of a trajectory.

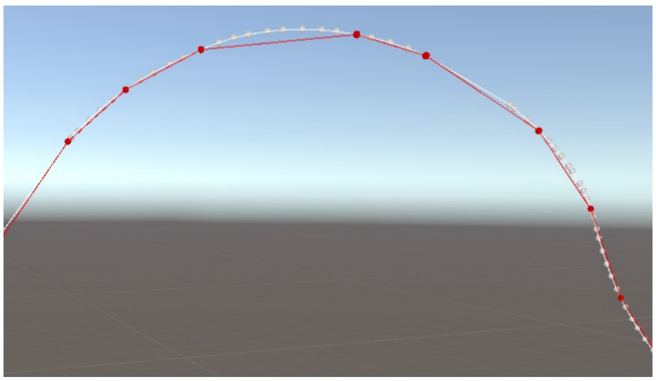

Figure 6. Example for Douglas-Peucker

In Figure 5 an example is shown, describing the recursive algorithm by the steps 1 to 6. Here it can be seen that the algorithm, starting in step 1, omits intermediate points with low information content. To determine the information content of a point in a geometry, first the farthest point is determined from the starting point. Now these points are connected with an imaginary line. All points which lie between the two points in the order given by the polygon are now considered in relation to this line. If all these points are in the minimum distance to the line, they are omitted, if some are outside, the new point furthest away is again connected to the starting point with an imaginary line. Again, all points between the starting point and the new point are considered in relation to the new line. If a point lies outside again, the procedure is repeated until no point lies outside. Points within the distance are defined as redundant. The point considered relevant is defined as the new starting point. The entire procedure is repeated until all points of the polygon are defined as relevant or irrelevant. As mentioned, the advantage of this is that no new points are defined, only unnecessary ones are omitted. And, of course, the algorithm allows different Peucker-Level (PL), described by different maximum distances. This helps to scale the number of points, describing the original trajectory and thereby the complexity. In Figure 6 another example with a curve (white points) is shown and a corresponding PL. In the next section we will evaluate this optimization with a verification process.

4. Verification

To verify the optimized motion-trajectory identification we have used a verification setup, described by a set of template-vectors and test-vectors. The sensor values of the 6-axis sensor on the ESP32-module are acquired (acceleration and rotation rate each in x, y and z). Subsequently, the sensor values are transformed into a trajectory and corresponding orientation by means of data fusion, stored as a time series as quaternions. The following DTW based on quaternions, compares the data of the respective test set with each element from the trajectory library (templates).

An identification of the test set is realized with a distance metric. Every comparison of two trajectories with a distance metric smaller than a given radius leads to a mapping of the current trajectory to the library trajectory, so there is a positive identification. This radius depends on the number of points of each trajectory, thus both explicitly on the data rate of the sensor system, and on the corresponding PL.

This radius can be determined by a calibration mechanism when the system is first used. Such a calibration mechanism could ask the user to perform given motions, and thereby determines an optimal radius for discriminating between identified and unidentified trajectories.

The following motion-trajectories are defined:

- Down
  The right arm is raised horizontally, the elbow is bent at a right angle. The hand is now moved down so that the forearm points downward. The starting position is then assumed.
- Left
  The right arm is raised horizontally, the elbow is bent at a right angle. The hand is now moved to the chest and back again.
- Right
  The right arm is raised horizontally, the elbow is bent at a right angle. The arm is now stretched out and moved back to the starting position.
- Twist
  The right arm is raised horizontally, the elbow is bent at a right angle. Now the hand is moved once back and forth over the ulna and radius.
- Up
  The right arm is raised horizontally, the elbow is bent at a right angle. In contrast to the down movement, the arm is rotated upward at the elbow and brought back to the starting position.

During the verification, we have chosen one of the template-vectors as test-vector and try to identify this compared to the template-vectors. Every test-setup consists of 60 tests, where different test-vectors are chosen and identified with the DTW-algorithm.

The results of the different verification setups are shown in Figure 7. Five different PL have been defined, reducing the number of points of the template-vector and

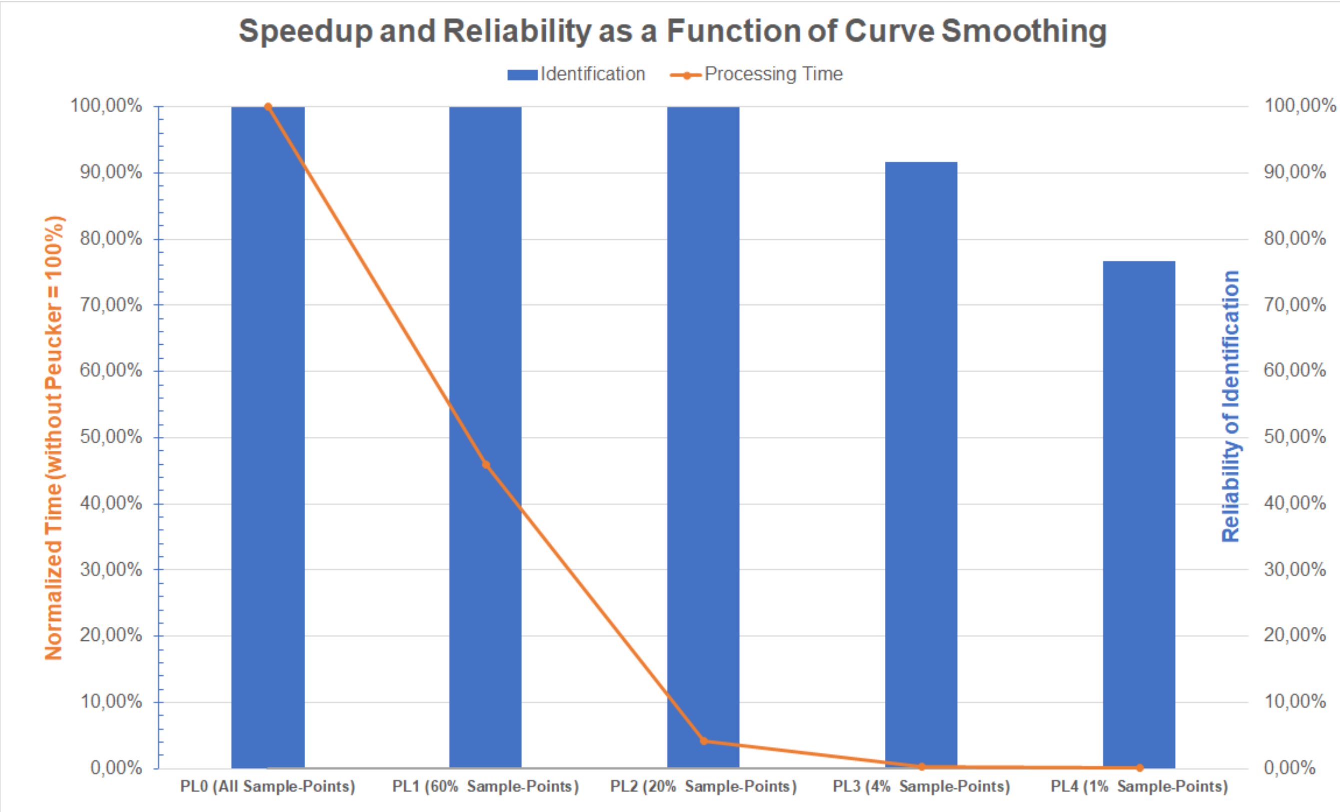

Figure 7. Specification of identification reliability and computing time depending on the respective Peucker-Level.

the test-vector from 100% down to 1%. Therefore, the processing time can be reduced remarkable with a minor loss of identification reliability. The best result is given by PL2: Here the identification reliability is still on 100%, the speed-up by reducing redundant points has a factor of 23,5. The computing power to derive the sample-points according the PL hardly noticeably deteriorate the result, so that there is a very good speed-up of 20.
The verification results show that the efficiency of the identification of the motion trajectories can be improved enormously. Using IoT-devices in the context of wearables, this is a major advantage. In the summary, we will discuss the limitations and challenges of the approach presented here.

5. Conclusions

The use of a platform for the acquisition and processing of data and states in biomedical applications allows high system complexity for different application scenarios and also provides high system flexibility. An integration of microcontroller-based IoT-modules into the platform enables decentralized intelligence, so functions and evaluations can be executed locally and therefore i.e. an identification of application-specific events can be realized at a low system level. This helps to reduce communication effort and improves a realtime high level data fusion based on simple events and not on an unmanageable amount of data. For such low-level data-based methods much computing time and computing performance is required.

In this paper, the objective was to execute an reliable identification of motion-trajectories already on the Smart-Boxes (IoT-modules, ESP32) in order to have a more simple application in mobile operation; in other words to have a better PnP-behavior. This identification of motion-trajectories is used in different application scenarios, we have focused here on posture evaluation (see section 2.1) and gait monitoring (see section 2.2). We have reduced the computational effort for this identification process with very high reliability by a factor of 20. Using the DTW-algorithm and Douglas-Peucker smoothing algorithm, we have realized a scalable solution by defining different PL. The management of different PL is simple because a higher PL omits only more points, never generate new ones.
However, there are some limitations of these results. In particular, this concerns the matching between template and test vectors, which is presented here in an idealized way. To improve the hit rate, it is useful if the test vector, related to a specific movement, has a clearly defined start and end point. Otherwise, artifacts from other movements will get into the alignment. Such a mixed motion is not present in the test vectors used here and therefore a

challenge.

The further work has the following key aspects:

- Establishing the connection between the two ankle-attached SmartBoxes in order to be able to perform a PnP gait monitoring.
- Establishing and testing new communication architectures between the communication participants SmartBoxes and smartphone, depending on the operating mode.
- Extension of verification patterns and procedures to optimize the reliability of trajectory identification.
- Optimization of calibration and initialization procedures.
- Taking the direction in 3-d space between the first two and the last two points of a trajectory into account, in order to get a quick decision at which stage of a movement (e.g. beginning, middle, end) the comparison is made.
- Testing the power consumption in different scenarios to get a reliable statement about the operating time in mobile operation.

References

Bourahmoune, K., Ishac, K., and Amagasa, T. (2022). Intelligent posture training: Machine-learning-powered human sitting posture recognition based on a pressure-sensing iot cushion. *Sensors*, 22(14):5337.

Chambers, R., Gabbett, T. J., Cole, M. H., and Beard, A. (2015). The Use of Wearable Microsensors to Quantify Sport-Specific Movements. *Sports Medicine*, 45(7):1065–1081.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122.

Espressif (2019). Esp32 series, datasheet. Technical report.

Goodman, C. C. and Fuller, K. S. (2015). *Pathology: Implications for the Physical Therapist*. Elsevier Saunders.

Hartmann, B., Spallek, M., and Ellegast, R. (2013). *Arbeitsbezogene Muskel-Skelett-Erkrankungen: Ursachen, Prävention, Ergonomie, Rehabilitation (mit CD-ROM)*. Handbuch der betriebsärztlichen Praxis. ecomed Verlagsgesellschaft in Hüthig Jehle Rehm.

Heidenfelder, J. (2011). *Entwicklung eines dynamischen Tests zur Prüfung der Rückfußdämpfung von Laufschuhen mittels biomechanischer Messmethoden*. PhD thesis, Technische Universität Chemnitz, Fakultät für Human- und Sozialwissenschaften.

Huang, P.-C., Liu, K.-C., Hsieh, C.-Y., and Chan, C.-T. (2017). Human motion identification for rehabilitation exercise assessment of knee osteoarthritis. In *2017 International Conference on Applied System Innovation (ICASI)*, pages 246–249.

Ito, T. (2008). Walking motion analysis using 3d acceleration sensors. *Computer Modeling and Simulation, 2008. EMS '08. Second UKSIM European Symposium on In Computer Modeling and Simulation, 2008. EMS '08. Second UKSIM European Symposium on (2008), pp. 123-128, doi:10.1109/ems.2008.95 Key: citeulike:3390917*.

Jablonski, B. (2012). Quaternion dynamic time warping. *IEEE Transactions on Signal Processing*, 60(3):1174–1183.

Klinger, V. (2016). Rehabilitation Monitoring and Biosignal Identification using IoT-Modules. In Bruzzone, A., Frascio, M., Novak, V., Longo, F., Merkuryev, Y., and Novak, V., editors, *5n International Workshop on Innovative Simulation for Health Care (IWISH 2016)*.

Klinger, V. (2019). Smart Platform-based IoT-Modules for Applications in Health Care and Rehabilitation. In Bruzzone, A., Frascio, M., Novak, V., and Eds., F. L., editors, *8th International Workshop on Innovative Simulation for Health Care (IWISH 2019)*.

Klinger, V. (2021). Postural Evaluation and Symptom Acquisition Based on IoT-Driven Multi-Sensor-Fusion. In Bruzzone, A., Frascio, M., Novak, V., and Eds., F. L., editors, *10th International Workshop on Innovative Simulation for Health Care (IWISH 2021)*.

Klinger, V. and Bohlmann, S. (2020). Application-Based IoT-System for Pandemic Prevention Based on Platform-Approach. In Bruzzone, A., Frascio, M., Novak, V., and Eds., F. L., editors, *9th International Workshop on Innovative Simulation for Health Care (IWISH 2020)*.

Krankenkasse, K. (2008). *Beweglich?: Muskel-Skelett-Erkrankungen - Ursachen, Risikofaktoren und präventive Ansätze*. Weißbuch Prävention. Springer Berlin Heidelberg.

Matuska, S., Paralic, M., and Hudec, R. (2020). A smart system for sitting posture detection based on force sensors and mobile application. *Mobile Information Systems*, 2020:1–13.

Srivastava, R. and Sinha, P. (2016). Hand movements and gestures characterization using quaternion dynamic time warping technique. *IEEE Sensors Journal*, 16:1333–1341.

TheMathWorks,Inc. (2023). https://mathworks.com.

Winnepenninckx, S. (2016). Douglas-peucker updated. FOSDEM VZW. https://doi.org/10.5446/34386.