

# Dance Pattern Recognition using Dynamic Time Warping

Henning Pohl, Aristotelis Hadjakos

Telecooperation

Technische Universität Darmstadt

Darmstadt, Germany

{hpohl@rbg, telis@tk}.informatik.tu-darmstadt.de

## ABSTRACT

In this paper we describe a method to detect patterns in dance movements. Such patterns can be used in the context of interactive dance systems to allow dancers to influence computational systems with their body movements. For the detection of motion patterns, dynamic time warping is used to compute the distance between two given movements. A custom threshold clustering algorithm is used for subsequent unsupervised classification of movements. For the evaluation of the presented method, a wearable sensor system was built. To quantify the accuracy of the classification, a custom label space mapping was designed to allow comparison of sequences with disparate label sets.

## 1. INTRODUCTION

Detecting patterns in movements is useful in a number of scenarios. Here, the focus is on dance movements in particular and their application in the area of interactive dance. In interactive dance human movements influence a computational system and motion patterns provide an additional capability in this process.

Interactive dance applications can mostly be found in the areas of art, gaming and clubs. For artistic purposes interactive dance is primarily used in installations, performance art and contemporary dance. Often dancers are given a certain level of control over the audio playing or the stage lighting. Dance is also used for generative art pieces to control a visualization. Video games, using interactive dance, mostly are interested in rhythmic patterns and how well a player adheres to a given step sequence.

The target scenario for this paper is the usage of interactive dance in a club setting. One open question in that context is how to enable more audience interaction and interactive dance is one possible way to do so. In this way it can provide an additional tool to DJs, VJs or other stakeholders in the overall experience. User studies with DJs and VJs have shown that they primarily assess a crowd's level of excitement and involvement using visual cues [1, 2]. Technological means to help with this assessment were generally viewed critically with participants fearing a loss of artistic freedom.

Recognizing dance patterns can serve two roles in this scenario. Such systems could, e.g., be used to gather information about dancers to provide an additional layer of information to DJs and VJs. A more interesting use however, is the possibility to outfit members of a DJ or VJ team with this technology. While being able to immerse themselves in the crowd they would still retain a level of control over the artistic process via their own movements. Pre-defined mappings or live mapping by a partner can be used to translate such movements to the desired auditive and visual output. Where other systems, e.g., provide continuous value activity measures, pattern information provides information in a discrete label space. Thus, pattern information can provide an additional informational layer to work with.

Interactive dance in a dance club context was explored by Ulyate and Bianciardi at the 1998 ACM SIGGRAPH convention, where they tested a number of prototypes [3]. They found that those devices that allowed for more freedom of movement, were yielding more satisfying dance interactions than devices such as buttons or pads. They also devised the role of an “experience jockey”, who controls the overall experience and changes mappings according to the current situation.

Also interested in dance club interactions, Feldmeier did a user study on their viability and quality [4]. Several user tests on groups of up to 200 participants were done and showed an overall positive user response to the system. Participants enjoyed the experience itself and felt that the music and lighting adapted well to their motions.

## 2. RELATED WORK

This section discusses sensor options for capturing dance movements and algorithms that have used to analyze dance movement.

### 2.1 Sensors for Interactive Dance

There are a number of different approaches to make dance movements available to the computer.

#### 2.1.1 Sensing floors

Sensing floors have been used to detect foot steps. Sensing floor systems vary with respect to resolution, modularity, size and response time. One of the first sensing floor system Johnstone's *PodoBoard* [5]. To use the system, metal plates had to be placed on the shoes of the users. Furthermore,

a pressure sensor was used to estimate the velocity of the shoe when touching the board. The *Magic Carpet* system was made of a grid of wires that were insulated with a piezoelectric material [6]. Furthermore, two doppler radars were present in the MagicCarpet system to detect upper body movement. The *LiteFoot* system detects lower body movements based on optical sensors, which are placed below the translucent floor [7]. The *Z-Tiles* uses piezoelectric sensing to sense foot steps [8]. Similar to the LiteFoot, the *Z-Tiles* is assembled from modular plates that can be freely combined. Srinivasan et al. built a modular system based on pressure sensitive polymers [9].

### 2.1.2 Sensing Shoes

Paradiso & Hu built a sensing shoe that allowed to measure the pressure exerted by the toes and the heel with piezoelectric pads, the bending of the shoe with a force-sensitive strip, and the movements with inertial sensors [10]. The *Shadow Dancer* system was based on a stet dancing shoe equipped with pressure sensors at the tip and the heel [11]. Fujimoto et al. used a three-axis accelerometer to the tip of each shoe [12].

### 2.1.3 Camera-Based Systems

Bevilacqua et al. used a marker-based motion capture system to track the movements of dancers [13]. Castellano et al. analyzed the camera signals with the EyesWeb platform to determine the quantity of motion and the amount of space a dancer occupies [14]. Ng uses a camera to track dancers that wear color-coded costumes [15]. Guedes uses video data to detect the frequency spectrum of dancing movements [16]. For this purpose, the difference of luminance between two sequential frames is computed and input into an array of 150 band pass filters.

### 2.1.4 Wearable Sensors

Various sensing technologies have been used to build wearable sensors to record dance movements: Hromin et al. used accelerometers, flex sensors, temperature sensors, photoreistors and pressure sensors among others [17]. Aylward & Paradiso used inertial sensing [18]. El-Nasr & Vasilakos use a special armband that measures heat flux, skin temperature, near body temperature and galvanic skin response and heart rate [19]. Based on that data, the dancer's arousal state is estimated.

An important aspect of a wearable sensor system that is used to record dance movements is the communication protocol to transmit the data to the computer. Hromin et al. use Bluetooth [17] and Aylward & Paradiso developed their own wireless protocol [18]. The mentioned approaches have the advantage that they are power efficient, which helps to increase the uptime that can be achieved without changing batteries. The *WiSe Box* transmits the sensor data via WiFi [20]. The sensor values are packed in OSC messages before transmission.

## 2.2 Recognition Algorithms

Detecting patterns in motions or working with motions in general requires a way to detect similarities in motions.

Algorithms doing so roughly fall into two categories:

**Temporal Feature Classification** Compare two motion sequences directly or using a set of descriptors

**Non-Temporal Feature Classification** Transform the data into a different space before further processing

With a way to determine similarities available, classification could be done via methods such as support vector machines (SVM), neural networks or the  $k$ -nearest neighbors ( $k$ -NN) algorithm.

### 2.3 Temporal Feature Classification

Gutknecht et al. used hidden markov machines (HMM) for Butoh, a form of experimental dance [21]. Setting out to classify movements, they designed a discreet three-dimensional motion space (*intensity, form and flow*) yielding a total of 64 motion categories. Dancers are equipped with three-axis accelerometers at the wrist, upper arm and upper leg, whose readings are relayed via Bluetooth. For classification, the values in a two second long sliding window are transformed into a sequence of features. On those features three HMMs (one for each dimension) are used to determine the most likely motion state sequence. The final motion state decision for the block is done using a majority vote. Gutknecht et al. furthermore designed a mapping from the motion space to a custom emotion space and subsequently derived a visualization from the detected emotional state.

Another method, somewhat similar to HMMs is dynamic time warping (DTW). The general idea is to compute the similarity of two given sequences that may differ in the temporal domain.

Tang et al. developed an algorithm to find repetitive patterns in motion capture data of dances [22]. 35 markers are tracked on a participant and the resulting posture data is normalized. From the resulting motion sequence a similarity matrix is derived. Postures in two frames are similar if the sum of the point by point euclidean distances is low. Repetitive motions can now be deducted from the similarity matrix, where diagonal patterns of similarity denote sequences, equally changing over time. Tracing patterns in the binarized similarity matrix in some respects is thus equal to similar image processing problems. Tang et al. use DTW to find such traces. Finally, using auto-clustering, patterns are classified as either cyclic or acyclic and an estimate of the cycle period is computed.

Fujimoto et al. used DTW to match foot movements to pre-recorded motion data [23]. Dancers perform their steps on top of a background track of constant rhythm. Based on the recognized dance steps, sounds are generated to go along with that background track.

DTW was also used by Bettens and Todoroff, who set out to detect gestures in a continuous sensor data stream [24]. For this purpose two sensors (three-axis accelerometer and two-axis gyroscope) are placed on both ankles of a dancing viola player. When performing, the downsampled sensor values are matched against a database of pre-recorded gestures. However, no segmentation of the live signal is tried.

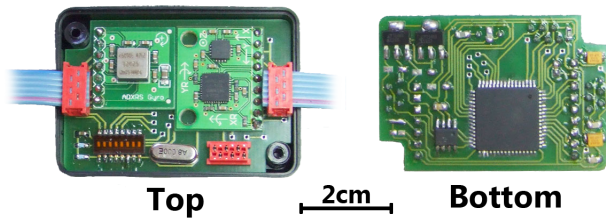


Figure 1. One of the used sensor nodes

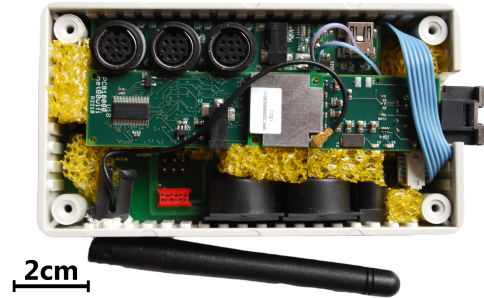


Figure 2. The interface board and the Gumstix

The signal is matched against the database at a number of different offsets instead.

## 2.4 Non-Temporal Feature Classification

Peng et al. built a system using two orthogonal cameras [25]. Based on training data, synthesized views are generated and used to build a tensor. During runtime the video data is similarly transformed into a corresponding tensor. Tensors are decomposed using higher order singular value decomposition, and used to approximate a view-invariant pose coefficient vector. With the coefficient vectors of the training set several classifiers were trained and subsequently evaluated. In their tests support vector machine (SVM) classifiers outranked fixed-threshold and von Mises-Fisher recognizers with recognition rates around 85% and a false detection rate of about 5%.

Nevada and Leman developed a method that is able to detect periodic movements in samba dance [26]. Two professional samba dancers were recorded on video, which was manually processed to derive a set of feature vectors. A periodicity transform [27] is used to find movements correlating with the musical meter. Thus, the proportion of periodicities in the signal is determined (independently for both dimensions). While no classification of gestures was performed, they noted that their approach could provide a useful methodology for dance analysis, with the periodicities potentially being used as classification features.

## 3. DANCESTIX

The *Dancestix* is a wearable sensor system that we developed and custom-built to record dance movements. The requirements for the *Dancestix* were:

- Measurement accuracy (R1),
- Usability in a dance club (R2), and
- Wearing comfort (R3).

The *Dancestix* consists of several inertial sensor nodes, a Gumstix embedded Linux system, and an interface board.

### 3.1 Inertial sensor nodes

The inertial nodes provide measurements of 3D linear acceleration and 3D angular velocity. The 3-axis accelerometer ADXL330 by Analog Devices was used to measure linear acceleration. Angular velocities were measured with the

2-axis gyroscope IDG-300 by InvenSense and the single-axis gyroscope ADXRS300 by Analog Devices. The measurements were performed at a resolution of 10 bit and a sampling rate of 100 Hz. The ADXL330 provides a measurement range of  $\pm 3g$ , the IDG-300 provides  $\pm 500^\circ/s$  and the ADXRS300 provides  $\pm 300^\circ/s$ . The sensor nodes send the data to the interface board over CAN (Controller-area network). We built four of said sensor nodes. A sensor was worn around the hip, on the right upper arm, on the right forearm and on the right thigh. Further details on the inertial sensor nodes can be found in [28].

### 3.2 Interface board and Gumstix

The interface board connects the sensor nodes to the Gumstix embedded Linux system (a *Verdex Pro XM4* mainboard with *console-vx* and *netpro-vx* expansions). The interface board communicates with the sensors over CAN and transmits the sensor data to the Gumstix over a serial interface (RS232). Furthermore, the interface board acts as a central power supply for the inertial sensor nodes and the Gumstix. Wi-Fi is used for communication of the data from the Gumstix to a host computer.

### 3.3 Discussion

Because of the wired connection between the sensor nodes and the interface board, a local power supply at each sensor node is unnecessary. This helps to reduce the weight of the sensor nodes, which is beneficial for measurement accuracy (R1) as independent motion of the sensor because of its inertia is minimized. Furthermore, the reduced weight and size are beneficial for wearing comfort (R3). The disadvantage of wires running along the dancer's body could be eliminated by integrating the wires and sensors into a special clothing.

Bluetooth and ZigBee were among the options that were considered for wireless transmission of the sensor data to a host computer. Bluetooth and ZigBee have the advantage that they consume significantly less power than Wi-Fi, which would maximize battery lifetime. However, Bluetooth and ZigBee are intended for wireless personal area networks (WPAN) with operating spaces of typically 10 m [29], which would be too little to be usable in a dance club (R2). Furthermore, the number of sensor-equipped dancers would be too limited if Bluetooth or ZigBee were

used. Bluetooth allows only seven slaves per master and ZigBee has a communication bandwidth of only 250 kb/s while a single Dancestix produces 24kb/s of data (4 sensors, 6 DoF/sensor, 100 samples/s, 10 bit/sample).

#### 4. DANCE PATTERN RECOGNITION

When searching for patterns we are doing so on the measure level. While not all patterns in all forms of dance conform to this division, it is an appropriate choice for the target scenario. According to this choice, the dance data is to be split into blocks, each corresponding to one measure.

Segmenting motion data into measures is not an easy task though. While some approaches exist to perform beat detection on motion data, e.g., the work by Enke [30], detecting beats in audio tracks is far more accurate. We used *BeatRoot* by Dixon [31] to compute the beat timestamps for the audio track beforehand. As the audio is in  $\frac{4}{4}$  format, four beats are automatically grouped to form one measure. The resulting blocks of motion data are multi-dimensional sequences of samples. The set of all blocks is denoted as  $\mathbb{B}$ .

##### 4.1 Block Similarities

Finding reoccurring patterns ultimately is a problem of finding similar blocks. If the difference between two blocks is sufficiently small, chances are the movement in the second one is a reiteration of the first one. There are two problems at hand: how to determine the similarity of two blocks and what threshold to use when grouping them together. While the second problem requires some evaluation and is detailed later on, the first shall be described in this section.

We determine block similarities via the DTW algorithm [32]. In DTW, a mapping from an input sequence to a given sequence is found that minimizes the distance between them (using the euclidian distance metric for sequence element distances). The sequences do not need to be of equal length, as DTW corrects for differences in speed, but need to be of the same dimensionality. Running the DTW algorithm on two sequences yields two measures: a distance between those two sequences, and a so-called *warp path* that describes the best possible alignment of the two sequences. The best possible alignment is that alignment, which minimizes the overall distance between the two sequences.

To speed up the DTW computations one can use constraints to limit the amount of calculations needed. We used *FastDTW* by Salvador and Chan [33] instead of a basic DTW. This algorithm reduces DTW complexity by iteratively computing the DTW for a coarser resolution, projecting it to a finer one and refining it. Using that multilevel approach, *FastDTW* works in  $O(n)$ , similar to constrained versions of DTW. Due to the nature of the algorithm a certain level of error is induced. This is primarily dependent on the radius used. Analysis by Salvador and Chan showed, that for higher radii the error converges to 0. Furthermore, this convergence is significantly faster than in constraint based DTW implementations. An analysis of appropriate *FastDTW* radii for the movement data used here, is given in Section 6.

##### 4.2 Classification

With a way to compute block similarities, further processing steps have to be taken to classify blocks into distinct groups. As no *a priori* knowledge on possible patterns is assumed, for maximal flexibility during live performances, classification is unsupervised and solely depends on statistical information. Furthermore, classification should work in real-time on streams of incoming motion data blocks. Unfortunately, having no set of labels given and the requirement to assign new labels to blocks as they come in, imposes severe restrictions on the classification. For example, a simple  $k$ -NN clustering would not work, as the number of desired clusters  $k$  is not known. Changing assigned labels later on is also undesirable, as such a change could confuse label consumers. For example, VJs using class information should be able to rely on consistent labels when using them to enrich their work. Thus, algorithms determining clusters by reexamining the classes for all available data from the set do not work. An algorithm is needed that preserves previous class assignments and classifies new blocks based on previous classification choices.

We used a threshold clustering approach for this purpose that assigns class labels on-the-fly and unsupervised. Assigned class labels are immutable as required above. In this approach, an input stream of motion data blocks shall be modeled as a sequence  $S$ , with

$$S = (s_1, s_2, s_3, \dots) \quad \text{with } s_n \in \mathbb{B}. \quad (1)$$

Clusters of blocks are modeled as sets  $C$  and represent blocks that were grouped together. All clusters are stored in sequence  $L$ , where a cluster's index in this sequence also serves as class label.

For the first incoming block  $S(1)$ , there is no decision to be made. It is the foundation for a first cluster, thus resulting in

$$L = (\{S(1)\}). \quad (2)$$

For subsequent blocks, the best matching preexisting cluster has to be determined. This assumes a distance function *dist* is defined, which for the purposes of this paper will be the DTW algorithm. In addition to that function, a threshold  $t$  has to be provided as well. The distance from a new block  $S(i)$  to an already found cluster  $C$  in  $L$ , is defined as the average distance to  $C$ 's members.

$$\text{clusterDistance}(S(i), C) = \frac{1}{|C|} \sum_{c \in C} \text{dist}(S(i), c) \quad (3)$$

The best match of a new block is hence given as the cluster with the smallest distance to, from all available clusters.

$$\text{bestMatch}(S(i), L) = \arg \min \left\{ \begin{array}{l} \text{clusterDistance}(S(i), L_n) : 1 \leq n \leq |L| \\ \end{array} \right\} \quad (4)$$

Based on the given threshold  $t$ , new blocks are either assigned to the best matching cluster or not. If the distance from the block to its best matching cluster is lower than  $t$  it is considered to belong to it. If that is not the case, a new cluster is created based on that block.

The threshold has to be chosen carefully for good results. If it is set too low, all blocks are considered unrelated and are assigned their own class. Conversely, a threshold value that is too high, leads to undesired mashing of blocks that should have been distinguished. The threshold value is dependent on the moves used and the features chosen. In Section 6, results for different threshold values are shown in detail.

## 5. EVALUATION METHOD

The proposed algorithm was evaluated in a user study with four participants. A remix of Lady Gaga’s song “*Just Dance*” was used in this study. At 119 bpm it is moderately fast and fits into the desired target scenario. The length of 4:54 min (which corresponds to 145 measures) is long enough to allow for several different dance patterns to be tested in a realistic setting. With a prominent bass drum track it also allows for accurate beat detection and makes it easy for participants to stay on time.

Six different dance movements were defined and assembled into a choreography for the study:

- A. Side steps with no arm movement
- B. Rock steps sideways without arm movement
- C. Rock steps sideways with arm movement
- D. Side steps with arm movement
- E. Side steps with arms up in the air
- F. Standing still with head bobbing

There are two distinct foot movements combined with three possible arm movements. In addition, there is a resting pose, used in three short intermissions present in the song. The movements were chosen as to allow participants to take part in the evaluation without lengthy training and for their good fit to a club setting. More complex movements were deemed too difficult for non-professional dancers in an ad hoc evaluation session.

Before starting a recording session, all participants were instructed on the testing procedure. The dance moves to be performed were explained to them beforehand as well. While recording, additional help was provided in the form of oral notification of upcoming transitions and movement instructions.

### 5.1 Classification Quality Rating

Running the proposed algorithm on the recorded dance data yields a sequence of class labels. The choreography given to the participants also defines a sequence of classes. When rating the classification quality the matching between those two sequences has to be determined. As the classifier labels blocks without a priori label information the set of labels used in both sequences will be disparate. Hence, a best fit label space mapping algorithm was used to align the label sequences.

Given are two label sequences  $A$  and  $B$  with corresponding label spaces  $\mathcal{A}$  and  $\mathcal{B}$ :

$$\begin{aligned} A &= (a_1, a_2, a_3, \dots) & a_n &\in \mathcal{A} \\ B &= (b_1, b_2, b_3, \dots) & b_n &\in \mathcal{B} \end{aligned} \quad (5)$$

To compare  $A$  and  $B$  we need a function that maps  $\mathcal{A}$  to  $\mathcal{B}$ . This function should be injective as to not allow multi mappings to the same label:

$$\begin{aligned} f &: \mathcal{A} \rightarrow \mathcal{B} \\ f(a_1) = f(a_2) &\Rightarrow a_1 = a_2 \end{aligned} \quad (6)$$

To find this mapping a cost matrix  $\mathbf{C}$  is used. This matrix of size  $|\mathcal{A}| \times |\mathcal{B}|$  is initialized to the zero matrix. Now both label sequences are traversed and the cost matrix is updated accordingly:

$$\begin{aligned} \mathbf{C}^i &= \mathbf{C}^{i-1} + \mathbf{D}^i \\ \mathbf{D}_{x,y}^i &= \begin{cases} 1 & \text{if } a_i = \mathcal{A}_x \wedge b_i = \mathcal{B}_y \\ 0 & \text{if } a_i \neq \mathcal{A}_x \\ -1 & \text{if } a_i = \mathcal{A}_x \wedge b_i \neq \mathcal{B}_y \end{cases} \end{aligned} \quad (7)$$

In the final cost matrix higher values indicate good fits. The mapping can then be extracted by reducing the matrix. Before doing so, however, the matrix is normalized with the corresponding class frequencies.

In each reduction step that mapping is found in the matrix, which has the best fit. This is denoted by the highest cell value in the cost matrix. The row and column of said mapping are then eliminated from the matrix. This process continues while there are still open classes to be mapped from and mapped to.

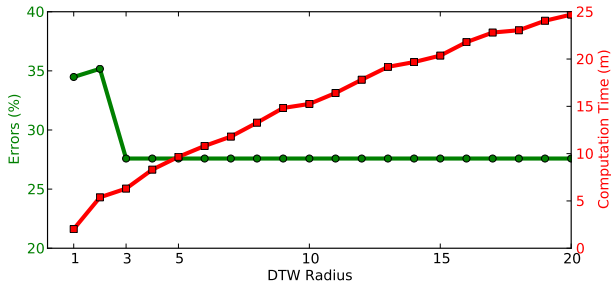
After this reduction a mapping from one label set to the other is found. For sets differing in size, some classes will not map to another one or will not be mapped to. The quality of the results from a classification can thus now be quantified via simple equality testing.

## 6. RESULTS

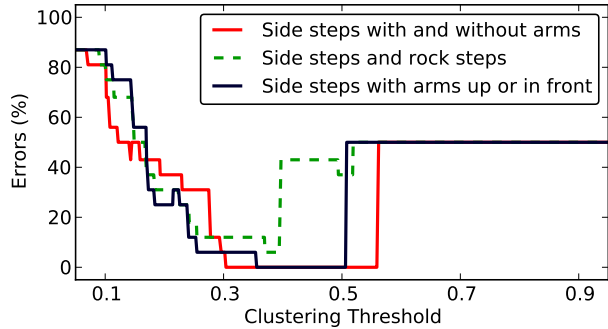
Based on the recorded motion data from several participants, the performance of the DTW algorithm was evaluated. Several aspects are of interest at this point:

- How well can distinct movements be distinguished?
- What is the influence of parameter choices?
- How strong is the influence of the sequence length?

The error rate is sometimes deceiving. Consider an algorithm that assigns a new label to each block. As the identifier mapping (see Section 5.1) tries to find the best fit, some blocks are still seen as correct. Specifically, an amount of blocks equal to the number of distinct classes in the given sequence is considered correct. Applying that logic to a sequence of 20 blocks with four distinct given classes, 20% would be seen as correctly classified even though no blocks were grouped. For longer sequences, this problem becomes less of an issue, as long as the number of distinct given classes does not increase correspondingly.



**Figure 3.** Comparing cost and error rate of different *Fast-DTW* radius choices.



**Figure 4.** Comparison of error rate when detecting differences between pairs of movement.

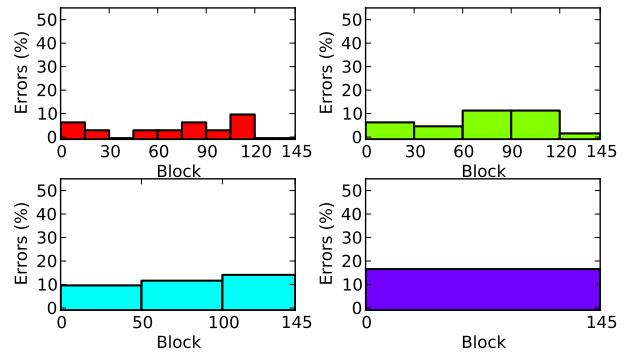
The other extreme is a classification that assigns all blocks to the same class. The identifier mapping will connect that class with the given class of the highest frequency. To put this in perspective, consider a 20 block sequence with one 10 block spanning class and two classes spanning 5 blocks. A recognized sequence of only one class would map to the 10 block long given sequence. Thus, 50% of the data is considered as correctly classified. This problem becomes less of an issue when the number of distinct given classes goes up or given classes are more uniformly distributed.

### 6.1 Dynamic Time Warping Radius

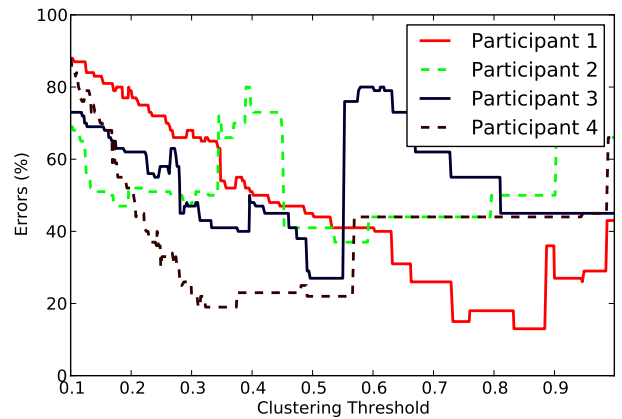
In a first step, appropriate radii for use with DTW were determined. As mentioned in Section 4.1, this value influences the computational cost and accuracy of the DTW algorithm. A multitude of DTW radii were tested on a full dance recording (145 blocks), with results shown in Figure 3. As can be seen any radius higher than 3 does not result in less errors. The computational cost increases significantly, though. Based on this result, a DTW radius of 4 was chosen for all subsequent DTW calculations.

### 6.2 Pairwise Motion Comparison

To determine the suitability of DTW, it was tested on a sequence of two different motion patterns. Sixteen blocks from a recorded session were used, spanning about 32 seconds in time. Each motion in the pair to be tested spans half of those blocks. As can be seen in Figure 4, our method was



**Figure 5.** Classification results for several subsequences of sensor data. The width of each bar denotes the data range being used.



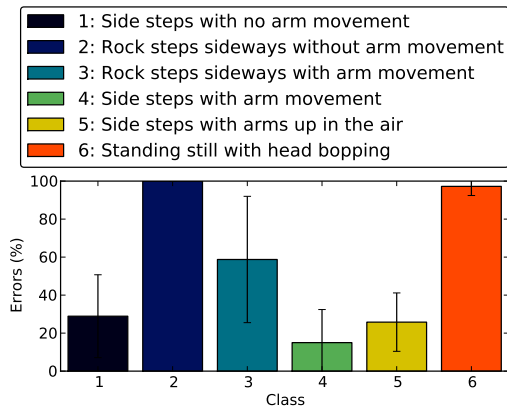
**Figure 6.** Comparing classification results from multiple participants.

able to correctly differentiate between motion pairs. Note that there is a range of threshold values being appropriate and some pairs are easier to differentiate than others.

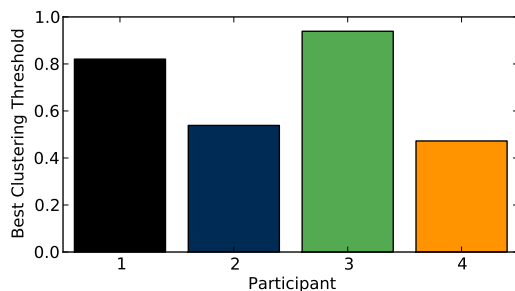
### 6.3 Influence of Sequence Length

The length of a sequence has an influence on the classification accuracy. One aspect is that over time, slight changes to a dance move are more likely to occur. While it is comparatively easy to perform the same move for 20 seconds, it is harder to repeat the same move after 2 minutes have passed. Thus, one would expect that a classification of shorter time spans contains less errors. To test this, one recording was analyzed in various windows. Looking at the results in Figure 5, a certain increase in overall error is apparent. While analyzing sequences of 15 block length, the maximum error was below 20%, the overall error in a classification of the whole sequence was at slightly over 30%. The data also shows peaks in error rate in later parts of the recorded sequence. Some of that can be attributed to the sensor straps loosening and required fastening motions.





**Figure 7.** Comparing multi-participant error rates by given class



**Figure 8.** Means of lowest error yielding thresholds per participant

#### 6.4 Comparison of Several Recordings

Figure 6 shows how well the DTW approach was able to classify recordings from four participants. As can be seen, some participant's motions were more easily discernable than other's. Also, each participant seems to have a separate best performing threshold. However, all curves exhibit somewhat similar behavior over the threshold range.

#### 6.5 Comparison of Error Rate by Dance Move

While an overall error measure provides a general performance estimate, determining the error per dance move helps with a more in-depth understanding. Figure 7 shows such data for all participants. The variance in error rate is deceiving to some extent, as the different recordings have different base error rates. Thus, the relative error rates are the most interesting aspect here. Looking at the data, classes 2 and 6 seem to be an issue. Those two coincidentally are also the two classes with the least amount of motion being required for them. On the other hand, more vivid movements were detected comparatively well.

#### 6.6 Threshold Choice

In most previous comparisons the error rate was given as a function of the threshold being used. It could also be seen in Section 6.4 how the chosen threshold varies with

each participant and performs best at different ranges. In Figure 8, a comparison of the best performing thresholds is shown. As can be seen, the range of appropriate thresholds is roughly contained in the  $[0.5, 0.8]$  interval.

## 7. CONCLUSION

As shown in Section 6, DTW based classification is able to correctly distinguish two given motions in one sequence. Working with data from real dance recordings, error rates of about 20-30% have been achieved. The results indicate that some movements were harder to distinguish than others. Especially movements eliciting low sensor responses were problematic. However, more pronounced movements were recognized much better.

## 8. REFERENCES

- [1] C. Gates, S. Subramanian, and C. Gutwin, "DJs' perspectives on interaction and awareness in nightclubs," in *Proceedings of the 6th ACM conference on Designing Interactive systems - DIS '06*, (New York, NY), ACM Press, 2006.
- [2] A. Engström, M. Esbjörnsson, and O. Juhlin, "Mobile collaborative live video mixing," in *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services - MobileHCI '08*, (New York, NY), ACM Press, 2008.
- [3] R. Ulyate and D. Bianciardi, "The Interactive Dance Club: Avoiding Chaos in a Multi-Participant Environment," *Computer Music Journal*, vol. 26, no. 3, 2002.
- [4] M. C. Feldmeier, *Large group musical interaction using disposable wireless motion sensors*. Master thesis, Massachusetts Institute of Technology, 2002.
- [5] E. Johnstone, "A MIDI Foot Controller - The PodoBoard," in *International Computer Music Conference*, (San Francisco, CA), 1991.
- [6] J. A. Paradiso, C. Abler, K.-y. Hsiao, and M. Reynolds, "The Magic Carpet: Physical Sensing for Immersive Environments," in *Conference on Human Factors in Computing Systems*, (New York, NY), ACM Press, 1997.
- [7] N. Griffith and M. Fernström, "LiteFoot: A floor space for recording dance and controlling media," in *Proceedings of the 1998 International Computer Music Conference*, (Ann Arbor, MI), 1998.
- [8] L. McElligott, M. Dillon, K. Leydon, B. Richardson, M. Fernström, and J. A. Paradiso, "'ForSe FIELDS' - Force Sensors for Interactive Environments," in *UbiComp 2002: Ubiquitous Computing* (G. Borriello and L. E. Holmquist, eds.), vol. 2498 of *Lecture Notes in Computer Science*, (Göteborg, Sweden), Springer, 2002.
- [9] P. Srinivasan, D. Birchfield, G. Qian, and A. Kidané, "A pressure sensing floor for interactive media applications," in *ACE '05: Proceedings of the 2005 ACM*

- SIGCHI International Conference on Advances in computer entertainment technology*, vol. 265, (Valencia, Spain), ACM, 2005.
- [10] J. A. Paradiso, K.-y. Hsiao, and E. Hu, "Interactive music for instrumented dancing shoes," in *Proceedings of the 1999 International Computer Music Conference*, (Beijing, China), 1999.
- [11] Y. Kim, D. Jung, S. Park, J. Chi, T. Kim, and S. Lee, "The Shadow Dancer: A New Dance Interface with Interactive Shoes," in *International Conference on Cyberworlds*, (Hangzhou, China), IEEE Computer Society, 2008.
- [12] M. Fujimoto, N. Fujita, Y. Takegawa, T. Terada, and M. Tsukamoto, "Musical B-boying: A Wearable Musical Instrument by Dancing," *Lecture Notes In Computer Science*, vol. 5309, 2008.
- [13] F. Bevilacqua, L. Naugle, and I. Valverde, "Virtual dance and music environment using motion capture," in *IEEE Multimedia Technology and Applications Conference Proceedings*, (Irvine, CA), 2001.
- [14] G. Castellano, R. Bresin, A. Camurri, and G. Volpe, "Expressive Control of Music and Visual Media by Full-Body Movement," in *Proceedings of 2007 Conference on New Interfaces for Musical Expression*, (New York, NY), 2007.
- [15] K. C. Ng, "Music via Motion: Transdomain Mapping of Motion and Sound for Interactive Performances," *Proceedings of the IEEE*, vol. 92, 2004.
- [16] C. Guedes, "Extracting Musically-Relevant Rhythmic Information from Dance Movement by Applying Pitch Tracking Techniques to a Video Signal," in *Proceedings of the 2006 Sound and Music Computing Conference*, (Marseille, France), 2006.
- [17] D. Hromin, M. Chladil, N. Vanatta, D. Naumann, S. Wetzel, F. Anjum, and R. Jain, "CodeBLUE: a bluetooth interactive dance club system," in *GLOBECOM '03. IEEE Global Telecommunications Conference*, vol. 5, (San Francisco, CA), IEEE, 2003.
- [18] R. Aylward and J. A. Paradiso, "A compact, high-speed, wearable sensor network for biomotion capture and interactive media," in *Proceedings of the 6th international conference on Information processing in sensor networks - IPSN '07*, (Cambridge, MA), ACM, 2007.
- [19] M. S. El-Nasr and T. Vasilakos, "DigitalBeing — Using the Environment as an Expressive Medium for Dance," *Information Sciences*, vol. 178, 2008.
- [20] E. Fléty, "The Wise Box: a multi-performer wireless sensor interface using WiFi and OSC," in *Proceedings of the 2005 conference on New interfaces for musical expression*, National University of Singapore, 2005.
- [21] J. Gutknecht, I. Kulka, P. Lukowicz, and T. Striaker, *Advances in Expressive Animation in the Interactive Performance of a Butoh Dance*, pp. 418–433. Springer, 2008.
- [22] K.-T. Tang, H. Leung, T. Komura, and H. P. H. Shum, "Finding repetitive patterns in 3D human motion captured data," *Conference On Ubiquitous Information Management And Communication*, 2008.
- [23] M. Fujimoto, N. Fujita, Y. Takegawa, T. Terada, and M. Tsukamoto, "A Motion Recognition Method for a Wearable Dancing Musical Instrument," in *2009 International Symposium on Wearable Computers*, IEEE, 2009.
- [24] F. Bettens and T. Todoroff, "Real-time dtw-based gesture recognition external object for max/msp and pure-data," in *Proceedings of the 6th Sound and Music Computing Conference*, (Porto, Portugal), 2009.
- [25] B. Peng, G. Qian, and Y. Ma, "Recognizing body poses using multilinear analysis and semi-supervised learning," *Pattern Recognition Letters*, vol. 30, no. 14, 2009.
- [26] L. Naveda and M. Leman, "Representation of Samba dance gestures, using a multi-modal analysis approach," in *5th International Conference on Enactive Interfaces*, (Pisa, Italy), Edizione ETS, 2008.
- [27] W. A. Sethares and T. W. Staley, "Periodicity transforms," *IEEE transactions on Signal Processing*, vol. 47, no. 11, 1999.
- [28] A. Hadjakos, E. Aitenbichler, and M. Mühlhäuser, "SYSSOMO: A Pedagogical Tool for Analyzing Movement Variants Between Different Pianists," in *5th International Conference on Enactive Interfaces*, (Pisa, Italy), Edizione ETS, 2008.
- [29] J.-S. Lee, Y.-W. Su, and C.-C. Shen, "A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi," *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, 2007.
- [30] U. Enke, *DanSense: Rhythmic Analysis of Dance Movements Using Acceleration-Onset Times*. Master thesis, RWTH Aachen University, 2006.
- [31] S. Dixon, "Evaluation of the Audio Beat Tracking System BeatRoot," *Journal of New Music Research*, vol. 36, 2007.
- [32] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 1, 1978.
- [33] S. Salvador and P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space," in *KDD Workshop on Mining Temporal and Sequential Data*, (Seattle, WA), 2004.