Multi-Dimensional Dynamic Time Warping for Gesture Recognition

G.A. ten Holt[a,b] M.J.T. Reinders[a] E.A. Hendriks[a]

[a] Information and Communication Theory Group
Delft University of Technology,
Mekelweg 4, 2628 CD, Delft, The Netherlands

[b] Human Information Communication Design
Delft University of Technology,
Landbergstraat 8, 2628 CC, Delft, The Netherlands

g.a.tenholt@tudelft.nl m.j.t.reinders@tudelft.nl e.a.hendriks@tudelft.nl

Keywords: Dynamic Time Warping, pattern recognition, multi-dimensional time series

Abstract

We present an algorithm for Dynamic Time Warping (DTW) on multi-dimensional time series (MD-DTW). The algorithm utilises all dimensions to find the best synchronisation. It is compared to ordinary DTW, where a single dimension is used for aligning the series. Both one-dimensional and multi-dimensional DTW are also tested when derivatives instead of feature values are used for calculating the warp. MD-DTW performed best in finding a known ground truth under noisy conditions. The algorithms were also used to perform simple classification of a set of 121 gestures. MD-DTW performed as well as or better than any single dimension in all tasks. In general, DTW on feature derivatives gave better results than DTW on feature values.

1 Introduction

There are various problem areas where signals need to be synchronised. When two time signals are compared, or when a pattern is sought in a larger stream of data, one of the signals may be warped in a non-linear way by shrinking or expanding along its time axis. Simple point-to-point comparison then gives unrealistic results, because one might be comparing different relative parts of the same signal/pattern. In these cases, some sort of synchronisation is needed. Figure 1 illustrates the difference between point-to-point comparison and comparison aided by synchronisation.

Dynamic Time Warping (DTW) [5] has long been used to find the optimal alignment of two signals. The DTW algorithm calculates the distance between each possible pair of points out of two signals in terms of their associated feature values. It uses these distances to calculate a cumulative distance matrix and finds the least expensive path through this matrix. This path represents the ideal warp - the synchronisation of the two signals which causes the feature distance between their synchronised points to be minimised. Usually, the signals are normalised and smoothed before the distances between points are calculated.

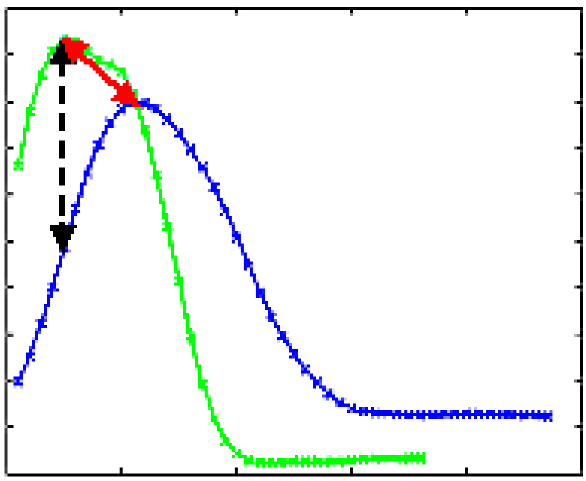

Figure 1: The importance of synchronisation. If signals are simply compared at each time instance (dotted arrow), they may be in different relative phases, giving unrealistic differences. Synchronisation ensures proper comparison (solid arrow).

DTW has been used in various fields, such as speech recognition [7], data mining [3], and movement recognition [1, 2]. Previous work in the field of

DTW mainly focused on speeding up the algorithm, the complexity of which is quadratic in the length of the series. Examples are applying constraints to DTW [4], approximation of the algorithm [8] and lower bounding techniques [3]. [4] proposed a form of DTW called Derivative DTW (DDTW). Here, the distances calculated are not between the feature values of the points, but between their associated first-order derivatives. In this way, synchronisation is based on shape characteristics (slopes, peaks) rather than simple values. Most work, however, only considered one-dimensional series.

We work on gesture recognition. Gestures are recorded by cameras and multiple features are extracted at each time instance, giving us multi-dimensional time series. We therefore investigated techniques for the synchronisation of such series. Multi-dimensional (time) series are series in which multiple measurements are made simultaneously. Such series have an K-dimensional vector of feature values for each (time) instance of the series. They can be synchronised by simply picking one dimension to perform DTW with and warping the complete series according to the warp found in this dimension. However, in many cases, all dimensions will contain information needed for synchronisation. We therefore propose multi-dimensional DTW (MD-DTW) for synchronising such series. An extension of DTW into 2 dimensions was proposed by [9], but not systematically tested. In [2] a type of MD-DTW is described, but only used for fixed-length series. In the next sections, we explain our MD-DTW algorithm and test it on our gesture dataset. We compare MD-DTW to regular 1D-DTW on various dimensions. We also looked at using derivatives instead of the values themselves (DDTW) and at combining both approaches in MD-DTW.

2 Multi-Dimensional Dynamic Time Warping

2.1 Algorithm

Multi-dimensional series consist of a number of measurements made at each instance. The number of measurements is the dimensionality of the series, the number of time instances its length. Note that multi-dimensional series need not be time signals, any situation in which several measurements are made simultaneously depending on one variable gives a multi-dimensional series. In this paper, we assume that measurements are stored in a matrix, in which columns are features and rows are time instances.

Take two series A and B. DTW involves the creation of a matrix in which the distance between every possible combination of time instances $A(i) \leftrightarrow B(j)$ is stored. This distance is calculated in terms of the feature values of the points. Various norms are possible. In 1D-DTW, the distance is usually calculated by taking the absolute or the squared distance between the feature values of each combination of points. For MD-DTW, a distance measure for two K-dimensional points must be calculated. This distance can be any p-norm. We use the 1-norm, i.e. the sum of the absolute differences in all dimensions. To combine different dimensions in this way, it is necessary to normalise each dimension to a zero mean and unit variance. For this, the dimensions must be comparable. If for instance one dimension contains real-valued measurements and one is binary, comparing them directly is not possible and a more sophisticated distance measure must be found. The MD-DTW algorithm is shown in figure 2.

The MD-DTW Algorithm

Let A, B be two series of dimension K and length M, N respectively.

- Normalise each dimension of A and B separately to a zero mean and unit variance
- If desired, smooth each dimension with a Gaussian filter
- Fill the M by N distance matrix D according to:

$$D(i,j) = \sum_{k=1}^{K} |A(i,k) - B(j,k)|$$

- Use this distance matrix to find the best synchronisation with the regular DTW algorithm

Figure 2: The MD-DTW algorithm

The benefits of MD-DTW can be seen when multi-dimensional series are considered that have synchronisation information distributed over different dimensions. Take the artificial 2D series shown in figure 3(a). It is clear that for the first half (in time) of the series, dimension 1 is useful for finding the correct synchronisation, whereas dimension 2 is uninformative. The converse is true for the second half of the series. If we were to perform 1D-DTW on this series using dimension 1, the result would be as shown in figure 3(b). The second half of the series is uniformly synchronised, since there is no information for 1D-DTW to work with. But it can be seen that for dimension 2, this is not the ideal synchronisation. 1D-DTW on dimension 2 gives a similar (but converse)

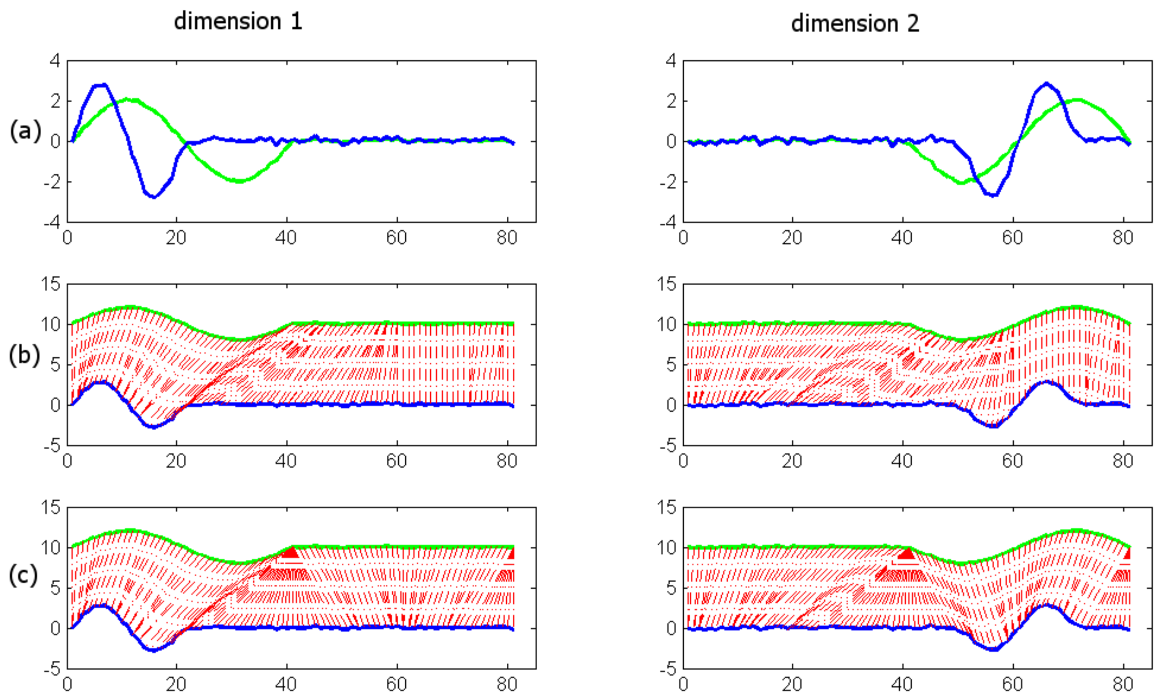

Figure 3: The necessity for MD-DTW. (a) shows two artificial 2D time series of equal mean and variance. Dimension 1 is shown in column 1, dimension 2 in column 2. The series contain synchronisation information in both dimensions. If 1D-DTW is performed using the first dimension, the result is suboptimal for dimension 2, as illustrated in (b). Note that the peaks and valleys in dimension 2 are not aligned properly. 1D-DTW on dimension 2 gives a similar suboptimal match for dimension 1. MD-DTW takes both dimensions into account and finds the best synchronisation (c).

result. MD-DTW takes both dimensions into account in finding the optimal synchronisation. The result is a synchronisation that is as ideal as possible for both dimensions, as shown in figure 3 (c). This is the advantage of MD-DTW over regular DTW.

Though the series in figure 3 is artificial, the situation it depicts is not unrealistic. Figure 4 shows a similar situation from a real-world multi-dimensional series: x- and y-co-ordinates of the right hand for the Dutch sign for acrobat. Here, the y-co-ordinate is informative for the first part of the series, and the x-co-ordinate for the second part. To get both peaks properly matched, MD-DTW is necessary.

2.2 DTW on Derivatives

[4] argue that for synchronising shape-characteristics of series (such as peaks and slopes), it is beneficial to perform DTW on the first-order derivatives of the feature values (DDTW). Since we want to perform such shape-matching in each of our dimensions, we considered this option for the MD-DTW algorithm. In our case, this meant taking the first-order derivative in each dimension separately. This gives us information about the slopes and peaks in each dimension. The series were first smoothed in each dimension with a Gaussian filter ($\sigma = 5$) to diminish noise effects. Then an approximation of the derivative was taken in each dimension using the filter $der(a(t)) = (a(t+1) - a(t-1))/2$.

Now we can perform two types of MD-DTW: on the feature values and on their derivatives. As a third type, we took the derivatives and added them to the series as extra dimensions, doubling the dimensionality of the series. In this setting, both the feature values themselves and their derivatives are taken into consideration when searching for the ideal warp. We tested our algorithm in all three settings. They are denoted as S(ignal), D(erivative) and SD(signal+derivative). In each case, the series were smoothed and normalised before (MD)DTW was commenced.

2.3 Dimension Selection

To compare series with various dimensions, it is necessary to normalise the dimensions. However, this is a problem if a dimension contains only noise. For example, position data gathered on the non-dominant hand in a one-handed gesture. Normalisation will enlarge the noise in this dimension to the same proportions as the informative data in other dimensions (such as dominant hand positions). The noise will

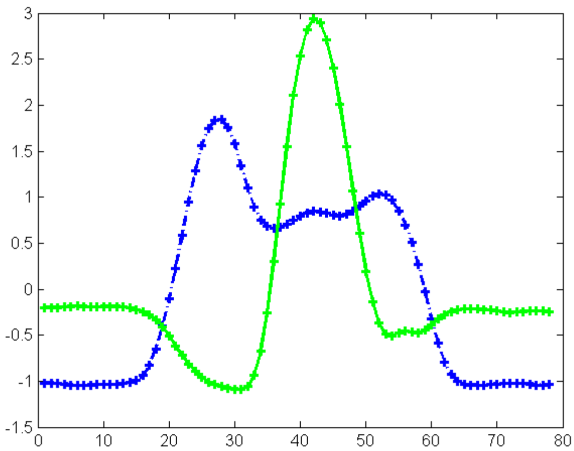

Figure 4: Sign for 'acrobat', a real-world example of synchronisation information distributed over dimensions. The solid line is the x-co-ordinate, the dotted line the y-co-ordinate. Each has information where the other is fairly constant.

then heavily influence the synchronisation. This is undesirable. We therefore perform dimension selection before commencing normalisation and synchronisation.

In dimension selection, the variance in each dimension of a multi-dimensional series is calculated. If the variance falls below a certain threshold, the dimension is not taken into account in the synchronisation process, nor in later calculations on features. In this way, dimensions with hardly any variance, that probably consist of noise, are disregarded. The variance threshold was empirically determined for our dataset by calculating the variance of known noise-dimensions (inert hands). It is also possible to weight dimensions rather then simply select or discard them.

In the next section, we give a number of tests in which we compared MD-DTW in various settings (S,D,SD) to regular DTW (1D-DTW) in the same settings on various dimensions. For all tests, the same filters for smoothing and derivatives were used.

3 Experimental Results

We tested the MD-DTW algorithm in different ways. First, we wanted to assess the accuracy of various DTW algorithms on a known ground truth. For this purpose, we created artificial warpings on a number of series and stored the warps. We then applied various versions of (MD)DTW to the warped series and compared the synchronisations calculated to the stored true synchronisations.

Secondly, we tested our algorithm in the domain of gesture recognition. We performed simple classification on a set of 121 gestures by comparing unknown examples to class prototypes. We used the synchronisation found by the various algorithms to determine at which time-points features should be compared. Better synchronisation should give more appropriate feature comparison and therefore higher classification scores. Data for all tests was retrieved from our gesture dataset, which is described below.

3.1 Dataset

In our experiments, we used a dataset of gestures. The gestures are signs from the standard vocabulary of Sign Language of the Netherlands. The gestures were recorded with 2 cameras in stereo position. Six features were automatically extracted from each pair of frames, resulting in a multi-dimensional series. The extracted features were: 3D positions (x, y, z), relative to the head, of the left and right hand. Each gesture was stored in a gesture length x 6 matrix. The gestures varied in length, average length was 91 $\pm$ 16 frames.

Our dataset consists of 121 different gestures. Each gesture was recorded from 67 different persons (all right-handed), giving us 67 examples for each of the 121 classes (for 9 classes, there were only 66 examples). In addition, there is a set of prototypes consisting of one example per class. These examples were hand-picked on the grounds of being a correct version of the class they represented. They were not optimised with respect to any of the classification methods mentioned below.

Many gestures in our set are one-handed, in which case the left hand is inert. In the two-handed gestures, the left hand in most cases copies the right. For this reason, we only tested 1D-DTW based on right hand features, since in almost all cases the left hand would either be less informative (inert) or equally informative (copying).

3.2 Artificially Warped series

We created an artificially warped series as follows: we took a multi-dimensional series (a gesture) and copied it. In the copy, we chose a random anchor point. This point was shifted 20 time instances to the left or right (direction was also chosen randomly). The adjacent points were shifted a fraction of 20 points, the fraction decreased with a point's increased distance to the anchor point (with a Gaussian curve). This created a localised time-warp. The warp was the same in all dimensions. After warping the time axis, the values in each dimension were re-interpolated to the original time axis. This gave us a warped series of which the correct warping to the original series was known. We will refer to the artificially warped series as the distorted series.

After warping, the original and the distorted series were normalised. Uniform zero-mean noise was then added to both to create some differences in feature values. We used three levels of noise variance: 0, 0.01 and 0.05 (the normalised series had a unit variance). Figure 5 shows an example of an artificial warping (zero noise).

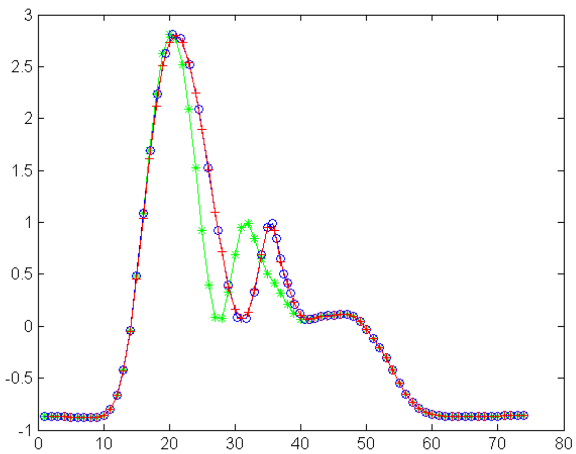

Figure 5: Example of an artificially warped series (only 1 dimension is shown). $*$ indicate the original feature values, o the new, shifted feature values, and $+$ show the values re-interpolated at the original time instances.

We first smoothed both series in each dimension with a Gaussian filter ($\sigma = 5$). We then applied various versions of DTW to the original and distorted series and stored the calculated warps. We calculated the goodness of a warp as follows: let the original series be s, the distorted series s'. Let GT denote the ground truth warp, and W the warp found by the algorithm. Let $W : s(i) \rightarrow s'(j)$ denote that W warps $s(i)$ onto $s'(j)$. Then the average aberration e is given by

$$e = \frac{\sum_{i=1}^{M} |j - j'|}{M}$$

where M = length of W, $GT : s(i) \rightarrow s'(j)$ and $W : s(i) \rightarrow s'(j')$.

We performed this operation on one random example of each class in our set, and took the median of the errors of all 121 examples (we took the median to be robust against outlier classes). We repeated this 20 times. The means and standard deviations over these 20 runs are given in table 1. It can be seen that MD-DTW performs better than any 1D-DTW variant. Adding more noise enlarges the difference between MD-DTW and the one-dimensional variants. More noise makes the correct warp more difficult to find, because even the correct warp will display differences in feature values. MD-DTW has the advantage that it has multiple dimensions in which the same warp is present, whereas the noise is different in each dimension, canceling out to some extent. For 1D-DTW, Derivative DTW appears to improve the results, but only for conditions with much noise. For MD-DTW, this is not the case (probably because it is already more robust against noise). For noiseless conditions, Derivative DTW causes deterioration. The reason may be that derivative is more sensitive to re-interpolation, causing some noise.

3.3 Application in Gesture Classification

In our domain, we want to classify gestures by comparing their feature values. MD-DTW should help us find the correct correspondences of time points between different gesture examples, so that the appropriate features will be compared.

To test the merits of MD-DTW over ordinary DTW in this respect, we executed a few simple classification tasks using various versions of DTW for synchronisation. The tasks are entirely equal in all other respects.

3.3.1 Nearest Neighbour Classification

One basic way of testing classification performance is the nearest neighbour (NN) scheme. We used the prototypes from our dataset as the training set (one prototype per class). All other gestures were used as the test set. In the test, we simply warped each test gesture on each prototype. We then calculated the feature distance between test gesture and prototype in the following way: let A be the test gesture, B the prototype, and W the calculated warp. Then

$$Dist(A, B) = \sum_{i=1}^{N} \sum_{j=k(1)}^{k(K)} |A(i, j) - B(i', j)|$$

where N = length of W, k is a list of dimensions that are selected for both A and B, e.g. $[1, 3, 5]$ (see section 2.3), and $W : A(i) \rightarrow B(i')$. A test gesture was given the class of the prototype to which it had the smallest distance.

Our test set consisted of 8 098 examples in 121 classes. We took 6 000 random samples from this set and computed the error of the nearest neighbour classification for several different methods of synchronisation. This process was repeated 20 times. The mean error and standard deviation over these 20 runs are given in table 2.

The errors in table 2 are large, which is to be expected for a NN with only one training example per class. However, we are not interested in classification performance so much as in comparing classification performance when various forms of DTW are used to achieve synchronisation. Table 2 shows that when synchronisation is calculated with MD-DTW, performance is better than with DTW synchronisation based

DTW on	Noise level	1D-DTW r.hand X	1D-DTW r.hand Y	1D-DTW r.hand Z	MD-DTW
Signal (S)	0	0.14 (0.01)	0.13 (0.01)	0.13 (0.01)	0.10 (0.00)
	0.01	0.54 (0.06)	0.45 (0.06)	0.27 (0.04)	0.12 (0.01)
	0.05	1.17 (0.08)	1.16(0.07)	0.84 (0.06)	0.30 (0.04)
Derivative (D)	0	0.31 (0.03)	0.34 (0.02)	0.29 (0.02)	0.22 (0.01)
	0.01	0.58 (0.05)	0.54 (0.06)	0.38 (0.04)	0.23 (0.01)
	0.05	0.95 (0.05)	1.01 (0.05)	0.78 (0.06)	0.39 (0.03)
Signal + Derivative (SD)	0	n.a.	n.a.	n.a.	0.12 (0.01)
	0.01	n.a.	n.a.	n.a.	0.13 (0.01)
	0.05	n.a.	n.a.	n.a.	0.25 (0.03)

Table 1: Average aberrations of the ground truth in frames of the calculated warp. Test series were normalised gestures. The aberrations were calculated 20 times for the entire gesture set. The values shown are the means and standard deviations over 20 runs. Noise level indicates the variance of the noise (the gestures were normalised to unit variance). Each type of DTW was performed both on the feature values (S) and the feature derivatives (D). MD-DTW was also performed on both combined (SD).

on a single dimension. The best performance outright is given by MD-DTW on the first derivatives of the dimensions. It performs significantly better than any other DTW variant. For each DTW variant, the Derivative condition performed better than the Signal or SD condition. All significances were calculated with a one-sided T-test, $p < .05$.

3.3.2 Warp Distances as Dissimilarity Features

In the previous section, we used the distance between a gesture example and a prototype, calculated for various synchronisations, to determine the class of the example. A more sophisticated method is to use the distances of an example to each prototype as (new) dissimilarity features [6] of the example and represent each example by its pattern of distances to all prototypes. This gives us a different dataset: we still have 67 examples for each of the 121 classes, but now, each example has 121 features, namely, its distances to each of the prototypes. The values of these features will differ for different synchronisations. Therefore, the examples have different feature values for each DTW variant.

When we regard the dissimilarities as ordinary features, we can train and test ordinary classifiers on our new dataset. The performance of a few classifiers for different DTW conditions was evaluated. Each classifier was trained and tested 20 times on random partitionings of the dataset. The ratio of training set and test set was 3:1. Mean and variance of the classification error over these 20 runs are shown in table 3 (first two columns).

The linear density-based classifier gives the better performance, but we are more interested in the relative performance for the various forms of DTW and MD-DTW. MD-DTW outperforms 1D-DTW when the x- or z-co-ordinate are used for synchronisation. For 1D-DTW on the y-co-ordinate, performance is equal with MD-DTW both for the S and for the D condition. Possibly the y-co-ordinate is the most informative dimension for most gestures in our dataset when it comes to synchronisation. But there is much variation in the performance per gesture. For certain gestures, the y-co-ordinate performs worse than MD-DTW, e.g. gestures which hardly vary in y-co-ordinate.

To investigate the performance per gesture, we trained and tested one-class classifiers for each gesture in the set. This entails training a classifier to distinguish one class from everything else. We used a k-means classifier set to reject maximally 1% of the target class. We trained and tested it for each class in turn, for each DTW variant, and repeated the process 20 times. The results are too extensive to show here. The average error over all classes was around 0.35 for all DTW variants, but it varied greatly between classes ($s.d. \approx 0.2$). We looked at the performance per class and counted for each 1D-DTW variant the number of classes for which it had the lowest error and also performed significantly better than any MD-DTW variant. In other words, we counted the classes for which an 1D-DTW variant would perform better than MD-DTW. These numbers are given in the rightmost column in table 3. (There were 4 classes for which an MD-DTW variant was better than all one-dimensional variants).

The results from the one-class study show us that in certain cases 1D-DTW can perform better than MD-DTW, and in many cases there is at least one dimension which will perform comparably well. How-

DTW on	1D-DTW r.hand X	1D-DTW r.hand Y	1D-DTW r.hand Z	MD-DTW
Signal (S)	0.791 (0.006)	0.791 (0.006)	0.831 (0.004)	0.748 (0.006)
Derivative (D)	0.722 (0.007)	0.699 (0.006)	0.759 (0.005)	**0.690 (0.006)**
Signal + Derivative (SD)	n.a.	n.a.	n.a.	0.699 (0.006)

Table 2: Classification error of the Nearest Neighbour algorithm using various DTW variants and conditions for synchronisation. Each was tested 20 times on random samples (size 6 000) of the data. The values shown are the means and standard deviations over 20 runs. The minimum error is shown in bold. It was significantly lower than the other errors (one-sided T-test, $p < .05$).

DTW on	Type of DTW	Linear density-based	5-nearest neighbour	One-class k-means [# better than MD-DTW]
Signal (S)	1D-DTW r.hand X	0.411 (0.012)	0.504 (0.009)	10
	1D-DTW r.hand Y	0.334 (0.008)	0.473 (0.009)	5
	1D-DTW r.hand Z	0.468 (0.010)	0.553 (0.008)	0
	MD-DTW	0.329 (0.011)	0.471 (0.009)	-
Derivative (D)	1D-DTW r.hand X	0.377 (0.010)	0.478 (0.011)	18
	1D-DTW r.hand Y	**0.297 (0.008)**	**0.432 (0.007)**	14
	1D-DTW r.hand Z	0.432 (0.007)	0.543 (0.009)	3
	MD-DTW	**0.295 (0.009)**	**0.432 (0.010)**	-
Signal + Derivative (SD)	MD-DTW	**0.299 (0.008)**	0.441 (0.010)	-

Table 3: Classification errors of several classifiers in dissimilarity space for various DTW variants and conditions. Each classifier was trained and tested 20 times on random partitionings of the data. The values shown are the means and standard deviations over 20 runs. The minimum error per column and all that did not differ significantly from it are shown in bold (one-sided T-test, $p < .05$). The rightmost column indicates performance for k-means one-class classifiers, trained to reject maximally 1% of the target class. The numbers shown are the number of classes (out of 121) for which the 1D-DTW variant performed significantly better than any MD-DTW variant (one-sided T-test, $p < .05$).

ever, it also shows us that the best dimension differs per gesture. It is therefore impossible to pick one dimension that will perform best for the entire set.

For the two regular classifiers, best classification performance outright on the entire set was given by the MD-DTW- and y-co-ordinate derivatives. There is no significant difference between these two for either classifier. For MD-DTW and for each DTW variant the Derivative condition performed significantly better than the Signal condition. All significances were tested with a one-sided T-test, $p < .05$.

4 Discussion

We discussed a novel technique for synchronising multi-dimensional series. The MD-DTW algorithm is an extension of the regular DTW algorithm that takes all dimensions into account when finding the optimal synchronisation between two series. We tested MD-DTW against one-dimensional DTW, and also looked at the performance of both algorithms when first-order derivatives were used instead of feature values.

When testing against a known ground truth, MD-DTW, as expected, showed an advantage over 1D-DTW when more noise was added. In various classification tasks, MD-DTW usually outperformed 1D-DTW on all dimensions. Sometimes, the performance of the best 1D-DTW dimension equaled that of MD-DTW. However, since there are cases for which the best dimension does not give good results, and since MD-DTW performance is always equal to or better than that of a single dimension (taken over the entire set), using MD-DTW is preferable.

MD-DTW has the disadvantage of being more expensive than DTW in its calculation of the distance matrix. The extra processing time is linear in the number of dimensions. For large or high-dimensional datasets, it may therefore be worth the effort to discover the best single dimension, possibly per class, and use 1D-DTW. The best dimension in our dataset was not the one with the largest variance, so it is not immediately clear how it should be found. Empirical testing on a training set is probably the best way. For smaller or low-dimensional datasets, MD-DTW is a better option.

For each single dimension and for MD-DTW, Derivative DTW gave the better performance. For our gesture dataset, top performance was given by MD-DTW on derivatives, sometimes equaled by DTW on the y-co-ordinate derivatives. We therefore conclude that for our dataset, MD-DTW on derivatives is the best synchronisation method.

References

[1] Andrea Corradini. Dynamic time warping for off-line recognition of a small gesture vocabulary. In *Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, page 83, July-August 2001.

[2] D.M. Gavrila and L.S. Davis. Towards 3-d model-based tracking and recognition of human movement: a multi-view approach. In *IEEE International Workshop on Automatic Face- and Gesture Recognition*, pages 272–277. IEEE Computer Society, Zurich, June 1995.

[3] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.

[4] Eamonn Keogh and M.J. Pazzani. Derivative dynamic time warping. In *First Intl. SIAM Intl. Conf. on Data Mining, Chicago, Illinois*, 2001.

[5] J. Kruskall and M. Liverman. The symmetric time warping problem: From continuous to discrete. In *Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison*, pages 125–161. Addison-Wesley Publishing Co., Reading, Massachusetts, 1983.

[6] Elzbieta Pekalska and Robert P.W. Duin. *The dissimilarity representation for pattern recognition : foundations and applications*. World Scientific, NJ, London, 2005.

[7] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall PTR, Englewood Cliffs, NJ, 1993.

[8] Stan Salvador and Philip Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. In *KDD Workshop on Mining Temporal and Sequential Data*, 2004.

[9] Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. Indexing multi-dimensional time-series with support for multiple distance measures. In *Proceedings of the 9th ACM SIGKDD int. conf. on Knowledge discovery and data mining, Washington, D.C.*, August 2003.