A Multidimensional Dynamic Time Warping Algorithm for Efficient Multimodal Fusion of Asynchronous Data Streams

Martin Wöllmer, Marc Al-Hames, Florian Eyben, Björn Schuller, Gerhard Rigoll

Technische Universität München, Institute for Human-Machine Communication, 80290 München, Germany
corresponding author: woellmer@tum.de, tel.: +49-89-289-28550, fax: +49-89-289-28535

Abstract

To overcome the computational complexity of the asynchronous Hidden Markov Model (AHMM), we present a novel multidimensional dynamic time warping (DTW) algorithm for hybrid fusion of asynchronous data. We show that our newly introduced multidimensional DTW concept requires significantly less decoding time while providing the same data fusion flexibility as the AHMM. Thus, it can be applied in a wide range of real-time multimodal classification tasks. Optimally exploiting mutual information during decoding even if the input streams are not synchronous, our algorithm outperforms late and early fusion techniques in a challenging bimodal speech and gesture fusion experiment.

Key words: Dynamic Time Warping, Multimodal Data Fusion, Asynchronous Hidden Markov Model

1. Introduction

A major aim of the science of human-machine-communication is to adapt the machine to the human and not vice versa. Therefore many researchers consider the aspects of interhuman communication as a paradigm for a user-friendly human-machine interface [69]. For instance, interhuman communication not only uses acoustic information but also visual: looking at the lips, the eyes, the gestures, and the face of the conversational partner helps to understand and interpret his/her intention [41]. In general, using more than just one sense increases redundancy in the information flux, which leads to higher robustness and enables humans to understand the transferred information, even if part of it is disturbed or missing.

Multimodal systems are an attempt to adapt the advantages of interhuman communication to human-machine communication by using more than one input device in order to make human-machine interaction robust, flexible, and natural [69]. Examples for multimodal systems causing higher robustness are the combination speech and gestures or the fusion of speech recognition and lip-reading: by using both modalities the speech recognition rate could significantly be increased [12]. Further, multimodal systems can achieve a higher flexibility, as the user has the possibility to switch between modalities as needed, e.g. during the changing conditions of mobile use. Many persons make use of hands-free speech input for voice dialing a car cell phone, but prefer pen input in public to avoid revealing private information [41].

From the signal processing point of view, a great challenge in designing multimodal systems is the integration of data coming from different modalities in order to reliably extract accurate information. In general, three different data fusion strategies can be distinguished: early fusion, hybrid fusion, and late fusion. Late fusion systems separately decode the data streams before merging the individual results to a valid multimodal pattern. Such systems scale up easily but do not optimally exploit mutual information [2]. Early fusion architectures preprocess multimodal data in a way that the streams reach the same sampling rate and integrate data to a single stream, which then has to be classified. Thereby mutual information can only be considered correctly if the streams are perfectly synchronous, which cannot always be guaranteed in practice [13]. A promising approach to overcome the disadvantages of both, late and early fusion, are hybrid systems like the asynchronous Hidden Markov Model (AHMM) first introduced in [7]. The AHMM concept has successfully been applied to various problems like meeting analysis [75], multimodal person identification [8], audio-visual speech recognition [7], or bimodal speech and gesture interfaces [2].

The main drawback of the AHMM is its high compu-

tational complexity. Therefore in this work a dynamic time warping (DTW) algorithm is modified in a way that it is applicable to multimodal data streams, even if they are not synchronous. We show how an increase of the dimensionality of a standard DTW can model the asynchronity between the streams while requiring less computational power than the AHMM. The multidimensional DTW, as introduced in this work, assumes *bimodal* data streams. However, the extension to more than two input modalities is straightforward. Thereby our fusion strategy is not limited to a certain modality combination (such as speech and gestures), but can in principle be applied to any multimodal recognition task. Like all hybrid fusion approaches, our algorithm is superior to late fusion techniques whenever there is a statistical dependence between the two modalities. It enables the modeling of asynchronity and can handle bimodal data streams with different sampling rates, different length, and varying temporal characteristics. Since our approach requires "reference streams" for each multimodal class, we derive a training algorithm which uses multiple references to increase the prototypicality of a single reference stream, so that during decoding, only one reference is needed per class.

In our speech and gesture fusion experiment, we assume (class-wise) pre-segmented data. Yet, we show how our multidimensional DTW can be expanded so that it carries out classification and segmentation simultaneously. We first introduce a three-dimensional DTW that can process conventional continuous (potentially asynchronous) bimodal feature vectors and uses synchronized bimodal reference patterns. We propose a simple method to synchronize the reference patterns which is also used in our experiments. In order to handle cases when modalities are highly correlated and synchronized reference streams cannot be assumed, we derive a four-dimensional DTW that does not require synchronized references. Finally, a modification of the three-dimensional DTW is presented, allowing discrete clustered feature vectors instead of continuous features.

This article is structured as follows: Section 2 investigates related work while Section 3 outlines the principles of multimodal integration and summarizes different input components and fusion strategies. In Section 4 we shortly review the asynchronous Hidden Markov Model before we explain our multidimensional dynamic time warping approach in Section 5. The derived algorithm is applied in a bimodal speech and gesture input fusion problem in Section 6 before we conclude in Section 7.

2. Related Work

The expansion of dynamic time warping to multiple dimensions is only rarely found in literature. There exist a few works which describe extensions of the DTW algorithm to include multiple dimensions, yet they differ significantly from the algorithm derived in this work, as they are not able to model the asynchronity of data coming from different modalities, representing a fusion strategy that combines the advantages of late and early fusion: in [67] and [28], a "multidimensional" DTW is used for (unimodal) gesture recognition and sensor fusion respectively. Yet, in these works the term 'multidimensional' refers to the size of the feature vectors, coming from the *same* modality and not to the number of degrees of freedom in the DTW distance matrix. Consequently, these approaches use the conventional two-dimensional distance matrix, whose entries are calculated from multidimensional feature vectors of a test sequence and a reference sequence. Thus, these algorithms cannot be applied for similarity measurements of two multimodal data streams.

Also the 'multidimensional' DTW used in [16] that is used for detecting texture similarities in images, simply measures the similarity between two multidimensional (yet not *multimodal*) series. Again, the feature vectors of the two streams that shall be aligned consist of multiple entries, however, eventually only two *unimodal* streams are aligned via DTW.

Similarly, DTW dimensionality expansions that have been applied for indexing multidimensional time-series in order to discover and analyze similar trajectories in GPS tracking, motion capturing, or handwriting recognition [68], just refer to the dimensionality of unimodal feature vectors and not to the dimensionality of the search space to align data.

The only technique that bears some similarity to our multidimensional DTW is the Multi Pattern DTW introduced in [35]. This algorithm was developed for joint decoding of multiple speech patterns to increase noise robustness. It attempts to find the best alignment between multiple speech patterns, which are known to come from the same speaker and belong to the same class. Thereby the Multi Pattern DTW is used as preprocessing for a multidimensional Hidden Markov Model. Yet, this algorithm cannot be applied for multimodal data fusion since it compares only multiple versions of unimodal sequences (i.e. the reference sequence is unimodal).

By contrast, the multidimensional DTW concept that has been applied in [64] to improve the standard DTW by allowing to control the warping function curvature is

completely different from the approach in this work. It attempts to influence the curvature of the warping function by augmenting the DTW dimensionality. After estimating the multidimensional warping function, it is projected onto the original dimension to provide the sought after warping function.

Reviewing all approaches referred to as "Multidimensional DTW" in literature, one can identify the research gap that a DTW approach which uses a more than two-dimensional search space to align and classify potentially asynchronous multimodal data (equivalently to the AHMM) does not exist so far.

3. Multimodal Integration

In a multimodal system multiple input components are combined for a better handling of computers or to improve the recognition performance. These input components are based on different disciplines of pattern recognition [69, 3, 42, 71]. A popular input modality is speech recognition (usually based on Hidden Markov Models (HMM) [47, 48]), for example in applications like telephone dialers [49], access control systems [51], service robots [52], or voice input in cars [60, 24]. Gesture recognition is another modality that is increasingly used in multimodal systems [77, 10]. In contrast to pen-based gesture detection which can be used to replace mouse navigation [18], video-based gesture recognition requires relatively high computational performance [3, 32] and is applied e.g. for video surveillance [22, 4] or sign language recognition [26]. Further important modalities are lip-reading [12, 17, 65], face recognition [6, 11, 5], handwriting recognition [45, 54, 30, 55], or eye tracking [57, 72, 63, 1].

Multimodal systems must process data provided by various recognition modules and merge them into one single semantic interpretation [27]. Common approaches are temporal integration [40], statistical integration [29], semantical integration [43], and rule-based integration [14]. Taking into account the level of integration, one can distinguish three major categories of multimodal integration: early fusion (integration at the feature level), late fusion (integration at a semantic level), and hybrid fusion.

3.1. Early Fusion

Early fusion means integrating the signals at the feature level. This implies that the signals coming from the different modalities have to be modified in a way that they reach the same sample rate. At each time step of the sampled signal the N_i features from M different modalities have to be concatenated in one large feature vector of dimension $\sum_{i=1}^{M} N_i$. Alternatively, if the unimodal feature vectors are mapped to a defined number of S_i symbols according to a codebook, for the integrated signal a new codebook consisting of $\prod_{i=1}^{M} S_i$ symbols is necessary. Having to deal with high-dimensional feature vectors and large codebooks respectively, early fusion requires great computational power [29]. Furthermore, the high dimensionality increases the number of degrees of freedom. Therefore a large amount of training data is needed, which often is expensive or not available. As early fusion merges the modalities at each time step, the signals have to be perfectly synchronous. Otherwise early fusion produces feature vectors or symbols that have not been leaned during the training phase. Integrating the signals at the feature level leads to good recognition results if the signals are synchronous and highly correlated like speech and lip movement [76], as the correlation structure of the modes can be taken into account automatically via learning [29].

Early fusion is applied in the automotive industry for collision mitigation systems [66], audio-visual speech recognition [38], affect recognition [73, 59, 56], or interest recognition [58]. Other approaches for multimodal integration, like the coupled HMM [37] or the multi-stream HMM [31], are also based on early fusion.

3.2. Late Fusion

In late fusion architectures signals are integrated at a semantic level. Signals are modeled separately and combined later, during the decoding phase. Each mode has an individual recognizer which is trained independently, so there is no explicit learning of the joint probability of the modalities. Late fusion uses unimodal training data, which is not as rare as multimodal data needed for early fusion [29] and profits from mature, well-engineered unimodal recognition techniques [2]. Furthermore, late fusion systems scale up easier because no re-training is necessary if further modalities are to be integrated. Other advantages of late fusion are easier handling of modalities which are temporarily missing and a higher degree of modularity. In contrast to early fusion architectures, mutual information coming from another modality is not considered during the recognition of a single mode, which causes late fusion to perform worse than early fusion if the modalities are correlated like in bimodal emotion recognition [21] or when using lip-reading for enhanced speech recognition. Multimodal fusion at the semantic level has been applied in systems like Bolt's "Put-that-there" [10], ShopTalk [15], FingerPointer [19], and others like in [36, 70, 62].

3.3. Hybrid Fusion

Hybrid systems for multimodal integration are an attempt to combine the advantages of late and early fusion. They aim to be as flexible as late fusion architectures, which can be scaled up more easily and can handle asynchronous data streams. At the same time hybrid systems should be able to exploit mutual information from other modalities during the recognition process. One realization is the asynchronous Input/Output HMM [9] which can be applied audio-visual for speech recognition. Another example for hybrid systems is the asynchronous Hidden Markov Model [7] which will be briefly reviewed in the next section as it forms the basis of comparison for the algorithm introduced in this article.

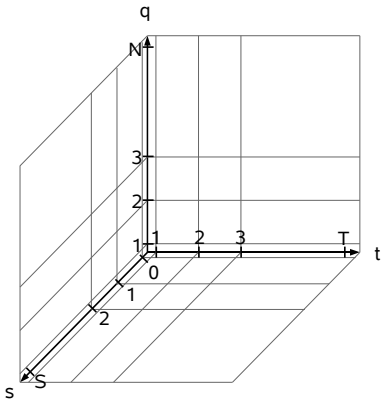

Figure 1: 3D trellis of the asynchronous Hidden Markov Model

4. The Asynchronous Hidden Markov Model

In order to handle multimodal data streams, the conventional HMM concept can be extended to an asynchronous Hidden Markov Model [7]. This section introduces the AHMM, since it will be compared to the newly introduced multidimensional DTW in Section 5.3.3. The AHMM can model the joint likelihood of two observation sequences. The two streams, each coming from a different modality, do not necessarily have to be synchronous, so the AHMM can be applied to a wide range of problems like meeting analysis [75], person identification [8], audio-visual speech recognition [7], or bimodal speech and gesture interfaces [2]. However, both training and decoding require great computational power, especially if the ratio of the input stream lengths is 1/2 (see Section 5.3.3).

An asynchronous Hidden Markov Model allows to model $p(\vec{x}, \vec{y})$ which is the joint likelihood of two observation streams of a bimodal system. T is the length of stream $\vec{x}$ and S is the length of stream $\vec{y}$ whereas it is assumed that $S \leq T$. Similar to a standard HMM, an AHMM has N different states $q_t = 1...N$ that are synchronized with stream $\vec{x}$. At each time step t a state emits a symbol from stream $\vec{x}$. At the same time a state can (with the probability $\epsilon(i)$) also emit a symbol from stream $\vec{y}$. Every time a $\vec{y}$ symbol is emitted, the variable $\tau_t = 0...S$ is incremented until the last $\vec{y}$ symbol has been emitted. Therefore τ_t can be seen as a second hidden variable which models the alignment between $\vec{x}$ and $\vec{y}$. The additional variable τ_t is included by adding a third dimension s to the trellis (see Figure 1).

To calculate the likelihood $p(\vec{x}, \vec{y}|\lambda)$ of a bimodal observation given a certain AHMM λ, we need a forward path variable $\alpha(i, s, t)$ [7] that, unlike the corresponding forward path variable in standard HMM, depends on three indices which are state, alignment, and time.

$$\alpha(i, s, t) = p(q_t = i, \tau_t = s, \vec{x}_t, \vec{y}_s) \tag{1}$$

Provided that $s > 0$ (meaning that the model already has emitted a $\vec{y}$ symbol), the induction step is

$$\begin{aligned}
&\alpha(i, s+1, t+1) = [1 - \epsilon_i] \cdot \\
&\cdot p(\vec{x}_{t+1}|q_{t+1} = i) \sum_{j=1}^{N} p(q_{t+1} = i|q_t = j) \cdot \alpha(j, s+1, t) \\
&+\epsilon_i \cdot p(\vec{x}_{t+1}, \vec{y}_{s+1}|q_{t+1} = i) \sum_{j=1}^{N} p(q_{t+1} = i|q_t = j) \cdot \alpha(j, s, t)
\end{aligned} \tag{2}$$

For the joint likelihood of the two observations the following termination equation holds:

$$p(\vec{x}, \vec{y}|\lambda) = \sum_{j=1}^{N} \alpha(j, S, T) \tag{3}$$

The Viterbi decoding algorithm is similar to the forward path calculation. However, the sums have to be replaced by max operators. Via backtracking the best state-sequence and the most probable alignment of the two streams can be obtained.

Calculating the forward path variable for all possible combinations of i, s, and t, the complexity of the AHMM algorithm is $O(N^2ST)$ as each induction step approximately requires N summations. If the alignment between $\vec{x}$ and $\vec{y}$ is forced in a way that $|t - T/S| < k$

with k being a constant indicating the maximum stretching between the streams, the complexity is reduced to $O(N^2Tk)$ [7]. In [2] it was shown that the complexity is reduced to $O(N^2[TS-S^2+T])$ if α values that cannot be part of a valid path through the three-dimensional trellis are ignored. The path restriction is implied by the fact that all $\vec{y}$ symbols have to be emitted until the last time step and the assumption that at every time step the number of emitted $\vec{y}$ symbols cannot be larger than the number of emitted $\vec{x}$ symbols and therefore $s \leq t$. As shown in [7], considerations concerning the time complexity of the AHMM are also valid for the space complexity, so that complexities in time and space are equal.

5. Multidimensional Dynamic Time Warping

As an alternative to likelihood-based tools like Hidden Markov Models, the dynamic time warping (DTW) algorithm has been successfully applied in recognition tasks related to speech or music processing [53, 39, 25, 33]. The DTW algorithm calculates the distance between an input sequence and a reference sequence which can be seen as the prototype of a certain class. As these two sequences may have different lengths or may differ in their temporal characteristics, the DTW algorithm performs a nonlinear distortion of the time axis so that the maximum correlation can be determined. Besides the distance, which can be seen as a similarity measure between an input pattern and a stored reference pattern, the DTW also delivers a warping function that maps each sample of the spoken word to the corresponding sample of the reference word.

DTW-based processing of multimodal data streams is possible if either the modalities are combined using early fusion or if the streams are classified separately and combined afterwards, which would be a late fusion approach. However, as mentioned before, both techniques have disadvantages: early fusion is very complex, as the dimensionality of the pattern vectors increases and cannot be applied successfully if the streams are not synchronized, whereas late fusion does not exploit mutual information during decoding. To avoid these drawbacks, we show how the dynamic time warping algorithm can be expanded to a hybrid fusion concept that uses mutual information coming from the other modality in an efficient way and can be applied to asynchronous data streams. Similar to the asynchronous Hidden Markov Model, where the second modality is modeled by a third dimension in the trellis, we expand the dimensionality of the DTW distance matrix.

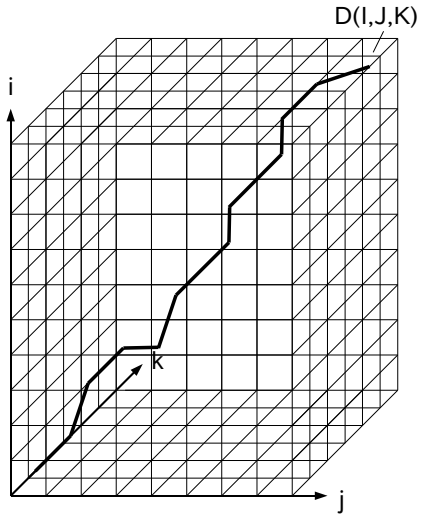

Figure 2: Three-dimensional distance matrix: cell D(I,J,K) represents the accumulated distance of the best stream alignment

5.1. Three-Dimensional Dynamic Time Warping

Our three-dimensional dynamic time warping (3D-DTW) algorithm searches for the best alignment between a synchronized reference sequence $R(i)$, containing features of both modalities, an input sequence $T_1(j)$, and another input sequence $T_2(k)$, coming from the second modality. The lengths of $R(i)$, $T_1(j)$, and $T_2(k)$ are I, J, and K respectively. Their alignment can be visualized by a path through a three-dimensional distance matrix (see Figure 2). The projection of the path to the $i-j-$plane corresponds to the DTW-path that maps input stream $T_1(j)$ to the features of the first modality of reference sequence $R(i)$ (Figure 3, mid). Consequently the nonlinear distortion of input stream $T_2(k)$, which is compared to the features of the second modality of $R(i)$ can be seen in the path projection to the $i-k-$plane (Figure 3, left), whereas the path in the $j-k-$plane represents the best alignment between the two potentially asynchronous input streams $T_1(j)$ and $T_2(k)$ (Figure 3, right).

Similar to asynchronous Hidden Markov Models, the three-dimensional dynamic time warping algorithm does not a priori decide about the alignment of the two input sequences, which would be an early fusion approach, but determines the best alignment of the sequences, so that they are optimally correlated to a synchronized reference of the bimodal pattern. In contrast to late fusion architectures, both modalities are taken

into account during the decoding phase as the developing of the path, which determines the accumulated distance between input and bimodal reference pattern, is influenced by both modalities. Thus, this extension of the dynamic time warping algorithm to a three-dimensional DTW can be called a hybrid fusion approach.

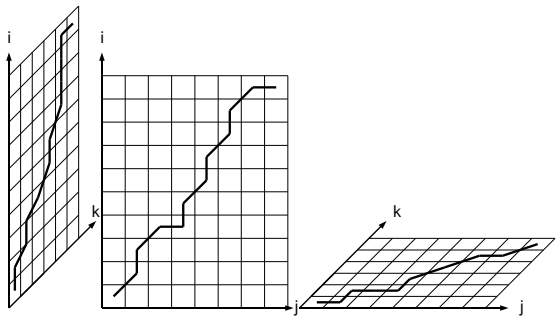

Figure 3: Projections of the path: i-k-plane, i-j-plane, j-k-plane

5.1.1. Extended Reference Vector

For our three-dimensional DTW we use a reference stream $R(i) = [\vec{r}(1), \vec{r}(2), ..., \vec{r}(I)]$ that consists of the reference features of both modalities, whereas we assume that in the case of highly correlated modalities (like in a speech and lip-reading pattern recognition problem) the modalities of the reference stream are perfectly synchronized. However, the *input* sequences do not necessarily have to be synchronous, as outlined before. In case we cannot assume synchronized references, the multidimensional DTW can be extended to a four-dimensional DTW which will be treated in Section 5.2.

If the sample rates of R_1, being the reference stream of the first modality, and R_2 (reference of the second modality) differ, the shorter stream has to be upsampled in a way that both references are of equal length (see Figure 4). After that, a reference stream

$$R(i) = \begin{pmatrix} \vec{r}_1(1) & \vec{r}_1(2) & \dots & \vec{r}_1(I) \\ \vec{r}_2(1) & \vec{r}_2(2) & \dots & \vec{r}_2(I) \end{pmatrix}$$

can be built, with I being the length of the longer one of the streams R_1 and R_2. This reference $R(i)$ corresponds to the i-axis of the three-dimensional distance matrix.

5.1.2. Distance Calculation

In order to find a distance measure for every cell of the three-dimensional distance matrix, we have to consider the reference sequence

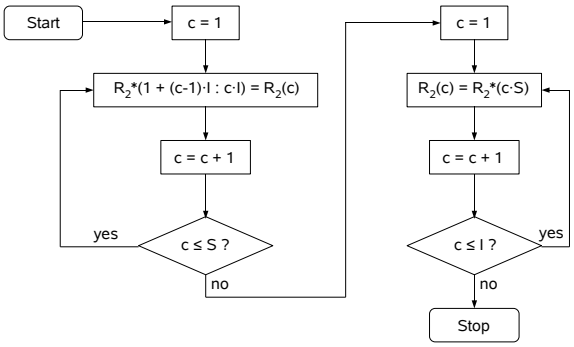

Figure 4: A simple upsampling algorithm: stream R_1 is of length I, stream R_2 is of length S. It is assumed that $I > S$, so R_2 has to be upsampled to length I. The algorithm defines a temporary stream R_2* of size $I \cdot S$ from which every S^{th} sample is taken. c denotes a count variable.

$$R(i) = [\vec{r}(1), \vec{r}(2), ..., \vec{r}(I)]$$

as well as the two input streams

$$T_1(j) = [\vec{t_1}(1), \vec{t_1}(2), ..., \vec{t_1}(J)]$$

and

$$T_2(k) = [\vec{t_2}(1), \vec{t_2}(2), ..., \vec{t_2}(K)].$$

The distance matrix is of dimension $I \times J \times K$ and its elements can be calculated as

$$d(i,j,k) = \sum_{n=1}^{N} [r_{1,n}(i) - t_{1,n}(j)]^2 + g \cdot \sum_{m=1}^{M} [r_{2,m}(i) - t_{2,m}(k)]^2. \quad (4)$$

$n = 1...N$ counts the features of the first input sequence $T_1(j)$, whereas $m = 1...M$ counts the features of $T_2(k)$. With g, a factor to weight the distance coming from the individual modalities is introduced. In case of $g > 1$ we enlarge the influence of mode $T_2(k)$ on the result of the bimodal pattern recognition problem. For instance if we think of a speech recognition system that processes acoustic and visual information we probably would choose $g < 1$, if $T_1(j)$ represents the acoustic information, since most information is delivered by the speech signal.

Similar to the unimodal DTW [34], the best alignment of $R(i)$, $T_1(j)$, and $T_2(k)$ can be visualized by a warping function F that determines the path through the distance matrix (Figure 2), going from cell $d(1,1,1)$ to

cell $d(I, J, K)$. This function F, which consists of L samples $\gamma(i, j, k)$, can be expressed as follows:

$$F = \gamma(1), \gamma(2), ..., \gamma(L) \tag{5}$$
$$\gamma(l) = (i(l), j(l), k(l)) \tag{6}$$
$$l = 1, ..., L \tag{7}$$

For the calculation of the best path, a three-dimensional accumulated distance matrix D is needed, whereas its endpoint $D(I, J, K)$ is equivalent to the total accumulated distance between the reference sequence $R(i)$ and the two input streams $T_1(j)$ and $T_2(k)$.

$$D(I, J, K) = \min_F \sum_{l=1}^{L} d(\gamma(l)) \tag{8}$$

Considering a cell $D(i, j, k)$ with $i \geq 2$, $j \geq 2$, and $k \geq 2$, the accumulated distance can be determined by choosing the best of 7 possible preceding cells. If cell $D(i, j, k)$ is reached by a movement parallel to one of the axis, the distance $d(i, j, k)$ is added to the accumulated distance of the preceding cell. In case $D(i, j, k)$ is reached by a movement parallel to one of the planes $i - k$, $i - j$, or $j - k$, the distance $d(i, j, k)$ is weighted by factor 2 because otherwise diagonal movements would be preferred. Consequently $d(i, j, k)$ has to be weighted by factor 3 if cell $D(i - 1, j - 1, k - 1)$ is considered as preceding cell as this movement could also be reached by three successive movements parallel to the three axis i, j, and k. These considerations result in the equation

$$D(i, j, k) = \min \begin{cases} D(i-1, j, k) & + & d(i, j, k) \\ D(i, j-1, k) & + & d(i, j, k) \\ D(i, j, k-1) & + & d(i, j, k) \\ D(i-1, j-1, k) & + & 2d(i, j, k) \\ D(i-1, j, k-1) & + & 2d(i, j, k) \\ D(i, j-1, k-1) & + & 2d(i, j, k) \\ D(i-1, j-1, k-1) & + & 3d(i, j, k) \end{cases} \tag{9}$$

$$(i \geq 2, j \geq 2, k \geq 2).$$

The values of D along the axis i, j, and k are

$$\begin{aligned} D(i, 1, 1) &= D(i-1, 1, 1) + d(i, 1, 1) \\ D(1, j, 1) &= D(1, j-1, 1) + d(1, j, 1) \\ D(1, 1, k) &= D(1, 1, k-1) + d(1, 1, k) \end{aligned} \tag{10}$$

For the planes $i - k$, $i - j$, and $j - k$ three possible preceding cells have to be evaluated:

$$D(i, j, 1) = \min \begin{cases} D(i-1, j, 1) & + & d(i, j, 1) \\ D(i, j-1, 1) & + & d(i, j, 1) \\ D(i-1, j-1, 1) & + & 2d(i, j, 1) \end{cases} \tag{11}$$

$$D(i, 1, k) = \min \begin{cases} D(i-1, 1, k) & + & d(i, 1, k) \\ D(i, 1, k-1) & + & d(i, 1, k) \\ D(i-1, 1, k-1) & + & 2d(i, 1, k) \end{cases} \tag{12}$$

$$D(1, j, k) = \min \begin{cases} D(1, j-1, k) & + & d(1, j, k) \\ D(1, j, k-1) & + & d(1, j, k) \\ D(1, j-1, k-1) & + & 2d(1, j, k) \end{cases} \tag{13}$$

For the starting condition we define:

$$D(1, 1, 1) = 3d(1, 1, 1) \tag{14}$$

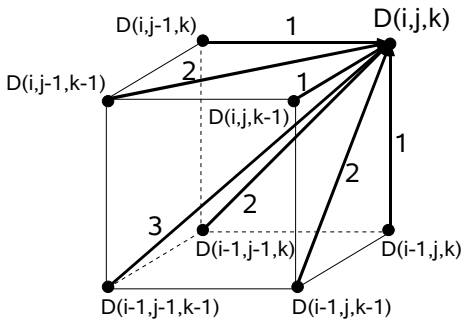

Figure 5: Three-dimensional path diagram with weighting factors corresponding to Equation 9

In Figure 5 the corresponding path diagram is visualized, considering the 7 possible preceding cells as corners of a cube. Of course Equation 9 is only one possible realization of the 3D-DTW.

Using the 3D-DTW for the classification of a bimodal input sequence, the three-dimensional dynamic time warping algorithm has to be carried out C times so that the input can be compared to the references of all C classes $c = 1...C$. As the different reference sequences usually have different lengths, we have to perform a path length normalization (e.g. scaling the total distances $D_c(I_c, J, K)$ with $1/I_c$) before the distances can be used for classification.

The path through the three-dimensional matrix D can be obtained via backtracking from cell $D(I, J, K)$ to

$D(1,1,1)$. Starting from the endpoint of the accumulated distance matrix ($D(I,J,K)$), the best preceding cell of every cell along path F has to be detected until the origin $D(1,1,1)$ is reached.

5.1.3. 3D-DTW Decoding Example

In the following we show a simple example of a bimodal classification problem and its solution using the 3D-DTW introduced in Section 5.1. For simplicity it is assumed that the pattern sequences of both modalities consist of only one feature per time step ($N = 1$ and $M = 1$). Furthermore, both modalities shall be equally weighted ($g = 1$). The pre-segmented bimodal input stream, consisting of the sequence $T_1(j)$ and $T_2(k)$ shall be assigned to one of the two classes A and B, which are represented by the two reference sequences $R_A(i)$ and $R_B(i)$:

$$R_A(i) = \begin{pmatrix} 2 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}$$

$$R_B(i) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 3 \end{pmatrix}$$

Both references are of length three, thus $I_A = I_B = 3$. The length J of the input stream coming from the first modality is three, the length K of the second input is two:

$$T_1(j) = (1 \quad 1 \quad 0)$$

$$T_2(k) = (2 \quad 3)$$

The 3D-DTW algorithm first computes the three-dimensional distance matrices for both references, according to Equation 4. The two matrices d_A and d_B can be seen in Figure 6.

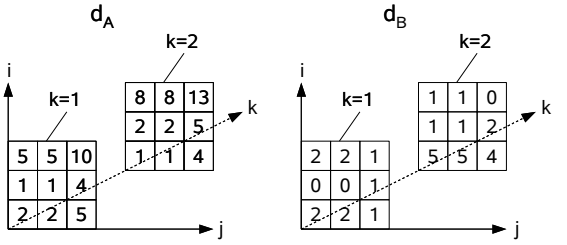

Figure 6: Distance matrices for the classes A and B

The next step is the calculation of the accumulated distance matrices D_A and D_B (see Figure 7) considering Equations 9 to 14.

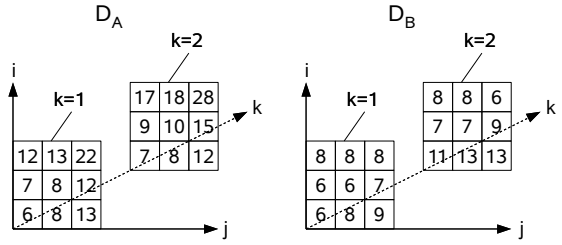

Figure 7: Accumulated distance matrices for the classes A and B

The final classification outcome can be obtained when comparing the normalized accumulated distances $D_{A,norm}(I_A, J, K)$ and $D_{B,norm}(I_B, J, K)$ which are

$$D_{A,norm} = \frac{D_A(I_A, J, K)}{I_A} = \frac{28}{3} = 9.3$$

and

$$D_{B,norm} = \frac{D_B(I_B, J, K)}{I_B} = \frac{6}{3} = 2.$$

Since $D_{B,norm} < D_{A,norm}$, the input is assigned to class B.

5.1.4. Segmentation

Like the unimodal DTW, the three-dimensional DTW algorithm can be extended in a way that it also finds the segment borders of continuous input streams. For that purpose the 3D-DTW has to be carried out simultaneously for all C reference sequences of the inventory. Consequently, the three-dimensional distance matrix consists of C different fields, one for each bimodal reference word (see Figure 8). Similar to the segmentation algorithm for the standard DTW [61] we define the "best end" $D^\star(j,k)$ that denounces the value $D(I_c, j, k)$ of the reference c having produced the lowest accumulated distance of all C references in question. With I_c being the length of reference sequence c we define:

$$D^\star(j,k) = \min_c D(I_c, j, k, c) \tag{15}$$

The preceding cell of a cell in the lowest $j-k$-plane $D(1, j, k)$ of every reference pattern can either be one of the cells $D(1, j-1, k)$, $D(1, j, k-1)$, or $D(1, j-1, k-1)$ of the same reference or a cell in the upper $j-k$-plane of the "best" preceding pattern which can be $D^\star(j-1, k)$, $D^\star(j, k-1)$, or $D^\star(j-1, k-1)$. So we can derive the following equation for the distance *between* the reference words:

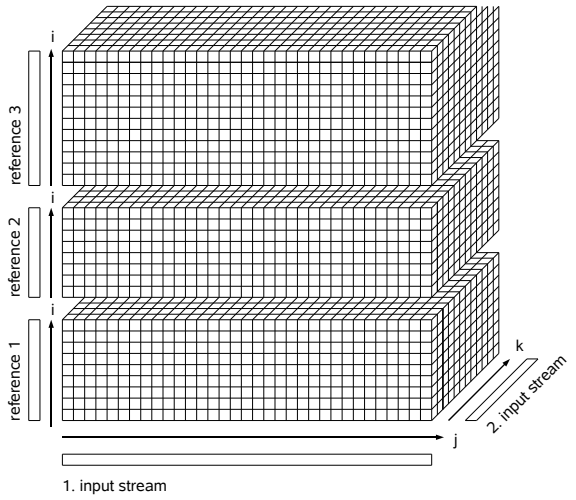

Figure 8: Simultaneous calculation of the 3D-distance matrix for three reference sequences

$$D(1,j,k) = \min \begin{cases} D(1,j-1,k) & + & d(1,j,k) \\ D(1,j,k-1) & + & d(1,j,k) \\ D(1,j-1,k-1) & + & 2d(1,j,k) \\ D^\star(j-1,k) & + & d(1,j,k) \\ D^\star(j,k-1) & + & d(1,j,k) \\ D^\star(j-1,k-1) & + & 2d(1,j,k) \end{cases} \tag{16}$$

Whenever the algorithm finds the beginning of a new word the path "jumps" from the top $j-k$-plane of the preceding word to the lower $j-k$ plane of the new word and one of the values $D^\star(j-1,k)$, $D^\star(j,k-1)$, or $D^\star(j-1,k-1)$ is accumulated. Similar to the two-dimensional DTW, "jumps" correspond to segment borders, however, in this case we are able to find a segment border of a bimodal data stream even if the streams are not synchronous. This would mean that $j \neq k$ at the segment borders.

In order to find $D^\star(j,k)$ we have to compare the values $D(I_c, j, k)$ of all C reference sequences.

As we want these distances to be comparable without any complicated path length normalizations, we apply the following rule for the accumulated distance *within* the reference words:

$$D(i,j,k) = \min \begin{cases} D(i,j-1,k) & + & d(i,j,k) \\ D(i-1,j-1,k) & + & d(i,j,k) \\ D(i-2,j-1,k) & + & d(i,j,k) \\ D(i,j,k-1) & + & d(i,j,k) \\ D(i-1,j,k-1) & + & d(i,j,k) \\ D(i-2,j,k-1) & + & d(i,j,k) \\ D(i,j-1,k-1) & + & 2d(i,j,k) \\ D(i-1,j-1,k-1) & + & 2d(i,j,k) \\ D(i-2,j-1,k-1) & + & 2d(i,j,k) \end{cases} \tag{17}$$

This implies that for a certain combination of j and k, the same number of distances have been accumulated in all of the fields of the distance matrix so that the distances of the different reference sequences can be compared directly.

Similar to the standard DTW, we can obtain the path through the three-dimensional accumulated distance matrix via backtracking.

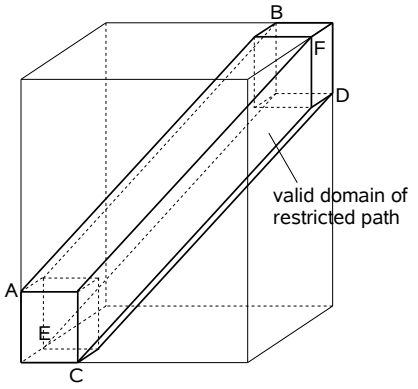

Figure 9: Global path restriction for path restriction variable $r = 0.25$: the path is forced to run within the corridor defined by A, B, C, D, E and F

5.1.5. Path Restrictions and Complexity

A strong divergence of the path from a thought line connecting the cells $D(1,1,1)$ and $D(I,J,K)$ of the accumulated distance matrix would mean an extreme distortion of the time axis of the input vectors and thus a very high asynchronity of the data streams. To avoid such unrealistic extreme distortions, the path can be restricted in a way that it is forced not to leave a certain corridor of a predefined width. This also saves computational power. To define the width of the restricted path, we introduce a path restriction variable r ($0 < r < 1$, see

Figure 9). The valid corridor is defined by the following coordinates (i,j,k):

$$A = (r \cdot I, 0, 0) \tag{18}$$
$$B = (I, (1-r) \cdot J, K) \tag{19}$$
$$C = (0, r \cdot J, 0) \tag{20}$$
$$D = ((1-r) \cdot I, J, K) \tag{21}$$
$$E = (0, 0, r \cdot K) \tag{22}$$
$$F = (I, J, (1-r) \cdot K) \tag{23}$$

$$0 < r < 1$$

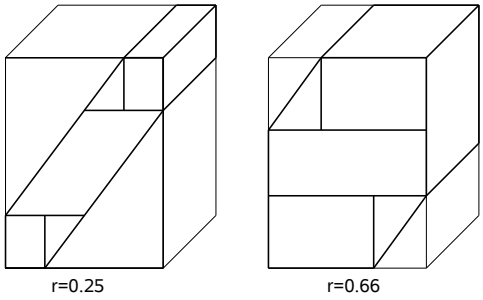

Figure 10: Geometric composition of the valid domain in the i-j-plane for path restriction variable $r < 0.5$ (left) and $r > 0.5$ (right)

Without any path restriction, the computational complexity of the 3D-DTW is $O(2IJK)$ (both in time and space) as the distance matrix and the accumulated distance matrix are of size $I \times J \times K$. If the valid domain for the path is restricted, not every cell of the matrices d and D has to be calculated, which leads to a reduction of complexity. Since the reduced complexity is equivalent to the volume of the corridor which defines the valid domain for the path, the complexity can be determined via geometric considerations. Therefore two cases have to be distinguished:

- if $r < 0.5$, a path restriction in the i-j-plane leads to a valid domain that can be composed of two triangles, two rectangles, and one parallelogram in the i-j-plane (Figure 10, left). Taking into account the path restriction in the i-k-plane, the volume of the corridor can be calculated by weighting the area of the rectangles and triangles by $1.5 \cdot r \cdot K$ and the area of the parallelogram by $2 \cdot r \cdot K$. Consequently, the volume of the corridor is

$$2 \cdot 1.5 \cdot r \cdot K \cdot (0.5 r^2 IJ + r^2 IJ) + 2 \cdot r \cdot K \cdot (1-2r) I (1-r) J$$

- if $r > 0.5$, the domain in the i-j-plane is composed of two rectangles, two triangles and one large rectangle in the middle (Figure 10, right). An additional path restriction in the i-k-plane leads to a scaling factor $0.5 \cdot (r+1) \cdot K$ for triangles and rectangles and a factor K for the rectangles in the middle. In this case the volume of the corridor can be given as

$$2 \cdot 0.5 \cdot (r+1) \cdot K \cdot (0.5(1-r)^2 IJ + r(1-r) IJ) + K \cdot (2r-1) IJ$$

Consequently, a restriction of the valid domain for the path leads to a reduction of computational complexity from $O(2IJK)$ to

$$O((17r^3 - 12r^2 + 4r)IJK) \qquad \text{for } r < 0.5 \tag{24}$$
$$O((-r^3 - r^2 + 5r - 1)IJK) \qquad \text{for } r > 0.5 \tag{25}$$

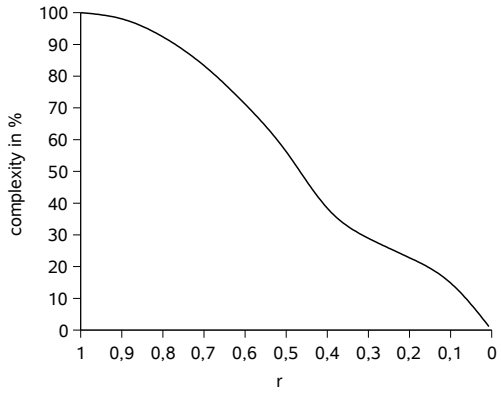

Figure 11: Complexity of the three-dimensional DTW algorithm versus path restriction variable r

Figure 11 shows how complexity decreases if r gets lower. In Table 1 the derived complexity of the three-dimensional DTW is compared to the complexity of the asynchronous HMM (see also Sections 4 and 5.3.3). For comparison, also the complexities of late and early fusion using DTW are listed. In the case of late fusion, I_1 and I_2 denote the lengths of the reference streams belonging to the first and the second modality whereas in the case of early fusion I represents the length of the bimodal reference.

To further reduce the computational complexity of our 3D-DTW, we also applied Monte Carlo sampling. However, this had a negative effect on recognition rates and was therefore not considered any further.

	unrestricted
AHMM	$O(N^2ST)$
3DDTW	$O(2IJK)$
DTW(lf)	$O(2(I_1J + I_2K))$
DTW(ef)	$O(2IJ)$
	restricted
AHMM	$O(N^2[TS - S^2 + T])$
3DDTW	$O((17r^3 - 12r^2 + 4r)IJK)$
DTW(lf)	$O(2r(2 - r)(I_1J + I_2K))$
DTW(ef)	$O(2r(2 - r)IJ)$

Table 1: Time and space complexity of the AHMM, late (lf) and early (ef) fusion DTW, as well as three-dimensional DTW for a path restriction variable $r < 0.5$: unrestricted and restricted

5.1.6. Training

One problem of unimodal pattern classification based on dynamic time warping is the determination of a suitable reference sequence for every class. As the reference sequence should represent a certain class, we expect that the reference is similar to all patterns that are to be assigned to the represented class. Consequently, the task is to find an *average* or *typical* sequence of a class. Since the patterns of a certain class may be of different length or differ in their temporal characteristics, it is not sufficient to average the samples of a set of training sequences at every time instant. Therefore we use a training algorithm that uses dynamic time warping to find out how the training material has to be temporally distorted, so that the samples can be averaged. The algorithm is outlined in Figure 12.

At first an initial reference sequence of length L is selected, whereas L is the median of the lengths of all sequences in the training set. Then the first training sequence is picked. Via dynamic time warping the most probable alignment between the initial reference and the current training sequence is determined. Consequently, every sample of the current training pattern is assigned to a sample of the reference sequence. In every iteration the samples of the reference sequence are updated by averaging the samples that have been assigned to it so far.

In order to obtain a reference sequence for the three-dimensional DTW, the references of both modalities are trained as shown above and merged as outlined in Figure 4.

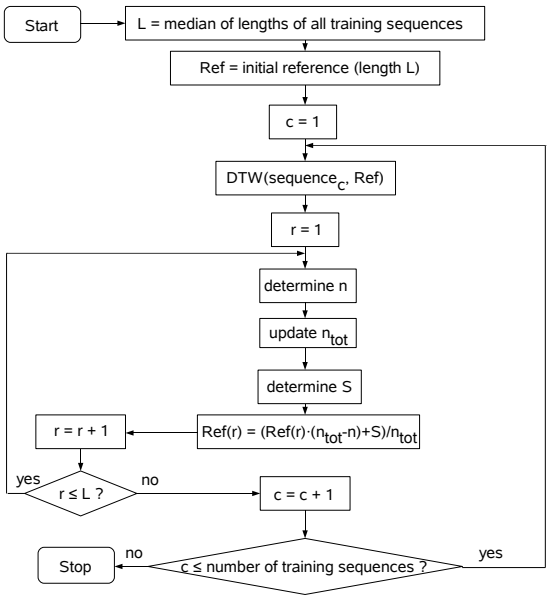

Figure 12: DTW training algorithm - L: median of the lengths of all training sequences; Ref: initial reference sequence; c: counts the training sequences; $sequence_c$: current training sequence; r: counts the samples of Ref; n: number of samples of $sequence_c$ that are assigned to $Ref(r)$; n_{tot}: total number of values that have been assigned to $Ref(r)$ so far; S: sum of the values that are assigned to $Ref(r)$

5.2. Four-Dimensional Dynamic Time Warping

So far it was assumed, that the reference sequences of both modalities are synchronous. Merging them to a unique extended reference sequence implies deciding about the alignment between the references a priori. If we do not want to fix the alignment between R_1 and R_2, a fourth dimension has to be added to the dynamic time warping concept. This means that not only the temporal distortion between T_1-R, T_2-R, and T_1-T_2 is modeled, but also the alignment between R_1, being the reference of the first modality, and R_2, being the reference of the second modality. The four-dimensional dynamic time warping algorithm computes the accumulated distance between a bimodal input, consisting of the sequences $T_1(j)$ and $T_2(k)$, and a bimodal reference $R_1(i)$ and $R_2(l)$. Therefore we define:

$$\begin{aligned} R_1(i) &= [\vec{r}_1(1), \vec{r}_1(2), ..., \vec{r}_1(I)] \\ R_2(l) &= [\vec{r}_2(1), \vec{r}_2(2), ..., \vec{r}_2(L)] \\ T_1(j) &= [\vec{t}_1(1), \vec{t}_1(2), ..., \vec{t}_1(J)] \\ T_2(k) &= [\vec{t}_2(1), \vec{t}_2(2), ..., \vec{t}_2(K)] \end{aligned} \tag{26}$$

Now the distance matrix is four-dimensional and has size $I \times J \times K \times L$, where L is the length of $R_2(l)$. Every cell of d can be computed as

$$d(i,j,k,l) = \sum_{n=1}^{N} [r_{1,n}(i) - t_{1,n}(j)]^2 + g \cdot \sum_{m=1}^{M} [r_{2,m}(l) - t_{2,m}(k)]^2. \tag{27}$$

As in Equation 4, $n = 1...N$ counts the features of $T_1(j)$ and $m = 1...M$ counts the features of $T_2(k)$. Again, g is the weighting factor for the individual modalities. The path determined by the warping function F, which now has four coordinates, goes from cell $d(1,1,1,1)$ to cell $d(I,J,K,L)$. To calculate the cells of the four-dimensional accumulated distance matrix $D(i,j,k,l)$, the best of 15 preceding cells in question has to be chosen. Similar to the three-dimensional case, weighting factors for $d(i,j,k,l)$ have to be introduced in a way that diagonal movements and movements parallel to axes are equally treated. Consequently, the cells of D can be calculated as follows:

$$D(i,j,k,l) = \tag{28}$$
$$\min \begin{cases} D(i-1,j,k,l) & + & d(i,j,k,l) \\ D(i,j-1,k,l) & + & d(i,j,k,l) \\ D(i,j,k-1,l) & + & d(i,j,k,l) \\ D(i,j,k,l-1) & + & d(i,j,k,l) \\ D(i-1,j-1,k,l) & + & 2d(i,j,k,l) \\ D(i-1,j,k-1,l) & + & 2d(i,j,k,l) \\ D(i-1,j,k,l-1) & + & 2d(i,j,k,l) \\ D(i,j-1,k-1,l) & + & 2d(i,j,k,l) \\ D(i,j-1,k,l-1) & + & 2d(i,j,k,l) \\ D(i,j,k-1,l-1) & + & 2d(i,j,k,l) \\ D(i,j-1,k-1,l-1) & + & 3d(i,j,k,l) \\ D(i-1,j,k-1,l-1) & + & 3d(i,j,k,l) \\ D(i-1,j-1,k,l-1) & + & 3d(i,j,k,l) \\ D(i-1,j-1,k-1,l) & + & 3d(i,j,k,l) \\ D(i-1,j-1,k-1,l-1) & + & 4d(i,j,k,l) \end{cases}$$

$$(i \geq 2, j \geq 2, k \geq 2, l \geq 2)$$

In case one (or more) of the indices i, j, k, or l is smaller than 2, Equation 28 has to be modified analogue to the principle explained in Section 5.1.2 (Equations 9 to 14).

A disadvantage of the four-dimensional DTW approach is its high computational complexity: without any path restriction the complexity of the 4D-DTW is $O(2IJKL)$. Similar to the 3D-DTW, the domain for the valid path can be restricted by defining a corridor through four-dimensional space, leading to a reduction of complexity and an avoidance of extreme distortions.

5.3. Probability-Based Dynamic Time Warping

Working with pattern vectors, the similarity of two vectors can easily be determined by evaluating the Euclidian distance (see e.g. Equation 4). However, distance calculations of high-dimensional pattern vectors used in speech recognition require great computational power. An alternative to processing pattern vectors is the introduction of discrete symbols, enumerating clusters in a multidimensional space. So every pattern vector is assigned to a certain symbol representing a cluster. As these clusters can be enumerated arbitrarily, neighboring clusters do not necessarily have similar symbol numbers. Therefore the *distance* or *difference* between two symbol numbers has no significance, since it does not indicate the similarity of the symbols. This means that the dynamic time warping algorithm has to be modified in order to be applicable to input streams consisting of discrete symbols. Our modified version of the DTW will be called *probability-based dynamic time warping* in the following.

5.3.1. Distance Calculation

Instead of reference sequences consisting of a time series of pattern vectors, the probability-based DTW uses a probability matrix P to measure the similarity of the reference and the input stream, which is a series of discrete symbols. Therefore for every class a set of training sequences is used to train a matrix $P(i)$ that defines the probability of a certain symbol at a certain time instant of the reference sequence:

$$P(i) = \begin{pmatrix} p_1(1) & p_1(2) & \dots & p_1(I) \\ p_2(1) & p_2(2) & \dots & p_2(I) \\ \vdots & \vdots & \ddots & \vdots \\ p_S(1) & p_S(2) & \dots & p_S(I) \end{pmatrix} \tag{29}$$

I denotes the length of the reference sequence and S is the number of discrete symbols in the alphabet. $p_s(i)$ is the probability of symbol s in time step i.

$$p_s(i) = p(s|t = i) \tag{30}$$

If the distance between sample i of a reference sequence $P(i)$ and sample j of an input stream $T(j)$ has to be calculated, we can use reference $P(i)$ as a look-up-table and convert the probability of the symbol of sample j of $T(j)$ into a distance measure by evaluating column i of $P(i)$. Consequently, for the probability-based version of the standard 2D-DTW we can define

$$d(i,j) = (1 - p_s(i) \cdot f) \cdot x \tag{31}$$

whereas s is the number of the symbol occurring at sample j of T. With the variables f and x the dynamics of $d(i,j)$ can be affected. x determines the maximum possible distance $d(i,j)$ in case $p_s(i) = 0$. For large symbol

alphabets we choose $f > 1$ as for a large number of possible symbols $p_s(i)$ tends to be small and most values of $d(i, j)$ will be close to x.

For the three-dimensional DTW, an extended P-matrix containing the probability distributions of both modalities is needed. Regarding the individual references P_a and P_b of the modalities a and b, the extended reference sequence $P(i)$ can be obtained by upsampling and merging P_a and P_b similar to the procedure for standard references consisting of pattern vectors, which has been outlined in Figure 4. Then the extended bimodal reference P is:

$$P(i) = \begin{pmatrix} p_{a,1}(1) & p_{a,1}(2) & \dots & p_{a,1}(I) \\ p_{a,2}(1) & p_{a,2}(2) & \dots & p_{a,2}(I) \\ \vdots & \vdots & \ddots & \vdots \\ p_{a,S_a}(1) & p_{a,S_a}(2) & \dots & p_{a,S_a}(I) \\ p_{b,1}(1) & p_{b,1}(2) & \dots & p_{b,1}(I) \\ p_{b,2}(1) & p_{b,2}(2) & \dots & p_{b,2}(I) \\ \vdots & \vdots & \ddots & \vdots \\ p_{b,S_b}(1) & p_{b,S_b}(2) & \dots & p_{b,S_b}(I) \end{pmatrix} \quad (32)$$

S_a denotes the size of the symbol alphabet for mode a, S_b indicates the number of symbols for mode b. $p_{m,s}(i)$ is the probability that symbol s occurs in sample i of the reference P_m of mode m. A distance measure denoting the similarity of the bimodal reference P at sample i and the two input sequences T_a (at sample j) and T_b (at sample k) can be defined as follows:

$$d(i, j, k) = (1 - p_{a,s_a}(i) \cdot f_a) \cdot x_a + (1 - p_{b,s_b}(i) \cdot f_b) \cdot x_b \quad (33)$$

s_a is the symbol that occurs at sample j of T_a whereas s_b stands for the symbol occurring at sample k of stream T_b. As outlined before, the factors f_a and f_b are used to adapt to the size of the symbol alphabet, so we can choose $f_a = const \cdot S_a$ and $f_b = const \cdot S_b$. x_a and x_b can be seen as weighting factors that control the influence of the individual modalities on the result of the classification, similar to the factor g in Equation 4.

The accumulated distance matrix for the probability-based 3D-DTW can be calculated the same way as for the standard 3D-DTW.

5.3.2. Training

As mentioned before, a reference $P(i)$ has to be trained using a set of training sequences. Therefore we use an algorithm that updates matrix $P(i)$ after every training iteration in a way that $P(i)$ represents a time series of probability distributions, characterizing a certain class. The principle of the algorithm is the same as the training strategy explained in Figure 12. The only difference is that now probability distributions are derived for every sample of the reference by simply counting the number of occurrences of a certain symbol assigned to a given sample of the reference stream, instead of averaging the values assigned to a sample (as in Section 5.1.6). Again, the training algorithm first picks an initial reference sequence of length L, whereas L is the median of the lengths of all sequences in the training set. Then every cell of the reference $P(i)$ is initialized with $1/S$ which implies that at the beginning all symbols have equal probabilities at every sample of reference $P(i)$. From now on $P(i)$ is updated with every iteration of the training algorithm whereas the best alignment of a new training sequence and the current reference $P(i)$ is determined via probability-based dynamic time warping, analogous to the procedure outlined in Section 5.1.6.

5.3.3. Comparison of AHMM and Three-Dimensional Probability-Based DTW

Since the three-dimensional probability-based DTW (3D-PBDTW) processes probabilities instead of distances, this DTW concept is an approach towards statistical models like the asynchronous Hidden Markov Model. Therefore it seems reasonable to compare the three-dimensional probability-based DTW with the AHMM in order to outline the differences. Table 2 shows varieties and similarities of the two concepts.

AHMM	**3D-PBDTW**
3D trellis: q (state), t (stream 1), s (stream 2)	3D dist. matrix: i (ref.), j (stream 1), k (stream 2)
states characterized by emission probabilities	reference characterized by symbol probability distributions
starting points $(q, t, s) = (q, 1, 0)$ or $(q, 1, 1)$; $q = 1...N$	starts at $(i, j, k) = (1, 1, 1)$
ergodic: $2N$ possible transitions; linear: 4 possible transitions	according to Eq. 9: 7 possible transitions
implicit path restriction (see Section 4)	explicit path restriction (see Figure 9)

Table 2: Comparison of the asynchronous HMM and the three-dimensional probability-based DTW

The main advantage of the three-dimensional DTW is its lower computational complexity compared to the AHMM. As outlined in [2], the complexity of the

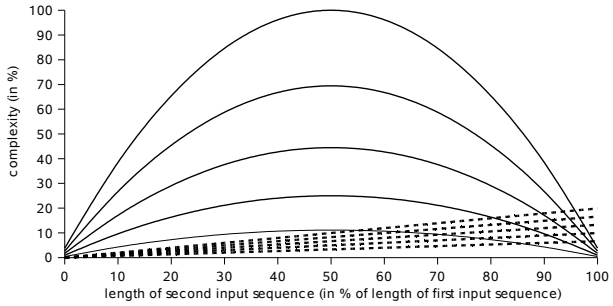

Figure 13: Complexity in % versus length of the second input stream (in % of the length of the first input stream) - Comparison of AHMM (curved line) and probability-based 3D-DTW (dashed straight lines) for a path restriction variable of $r = 0.15$; number of AHMM states: 10, 15, 20, 25, 30 (the more states, the higher complexity); number of samples of 3D-DTW reference sequence: 40, 60, 80, 100, 120 (the more samples, the higher complexity)

AHMM strongly depends on the ratio between the length of the two streams (see Figure 13). If the input sequences are of equal length, the AHMM corresponds to an early fusion architecture since in every time step two symbols have to be emitted. Consequently computational complexity decreases if the lengths of both input streams are similar.

As explained in Section 5.1.5, for a given path restriction variable r, the complexity of the probability-based 3D-DTW increases linearly with the length of the sequences. Figure 13 compares the complexities of the restricted AHMM and the 3D-PBDTW at a path restriction variable $r = 0.15$ for different numbers of AHMM states and 3D-DTW reference vector lengths respectively (see also Table 1 in Section 5.1.5). Note that the path restriction variable $r = 0.15$ resulted in the best 3D-DTW performance in our speech and gesture experiment (see Figure 14). For a typical scenario (25 AHMM states, length of the 3D-DTW reference vector is equal to 100, path restriction variable $r = 0.15$, length of the second input stream is equal to 70% of the length of the first input stream) the 3D-DTW speed-up factor would be 5.1, compared to the AHMM.

6. Experiments

6.1. Speech and Gesture Data

The multimodal data set which was used to evaluate the performance of the three-dimensional DTW is the speech and gesture data set that had also been applied in [2]. A sequence of this data set consists of two partially asynchronous streams, one of them being a speech signal, the other one being a mouse-gesture signal. The speech stream can be assigned to one of 11 words, namely the English digits from *one* to *nine*, including *zero* and *oh*. Furthermore, 10 different mouse-gestures can be distinguished: *up*, *down*, *left*, *right*, *downup*, *updown*, *leftright*, *rightleft*, *clockwise* and *counterclockwise*. The speech and gesture data is combined to 26 different multimodal input commands which are to be classified (i.e. *"four-counterclockwise"*, *"zero-rightleft"*, etc.). For each of the 26 multimodal commands 216 training sequences and 104 test sequences are available. 60% of the sequences are synchronous, meaning that speech and gesture stream both start at $t = 0$ but can be of different length. 40% of the data are sequential, which means that the gesture stream starts at time $t = 0$ and the speech stream starts at some time instant $t > 0$. Thereby the offset between gesture and speech is chosen randomly. The maximum distance between the end of the gesture stream and the beginning of the speech stream was set to 20 frames. Synchronity was chosen according to user studies in [44].

To keep AHMM computing time within the bounds of possibility, both, speech and gesture stream, consist of discrete symbols instead of multidimensional pattern vectors. Consequently the probability-based DTW (see Section 5.3) has been applied for DTW-based decoding. In the gesture stream, mouse movements were discretized with a codebook size of 8 symbols. For the speech stream the Aurora speech database was used. Every 10 ms a 20 ms-Hamming Window was used to extract 39 features: a pattern vector includes 12 melfrequency cepstral coefficients plus energy, as well as their first and the second order temporal derivatives. Applying K-Means clustering with 25 iterations, the 39-dimensional pattern vectors were then discretized with a codebock of only 50 symbols in order to keep AHMM decoding complexity computationally feasible.

6.2. Unimodal Classification

At first the performance of the individual classifiers was examined. Table 3 shows the recognition rates if only the isolated speech stream is to be assigned to one of the 11 word classes or if the isolated gesture stream (10 different classes) shall be classified. For the gesture classification discrete ergodic HMM with 20 states had been trained (20 EM-iterations). The speech sequences were classified using discrete ergodic HMM with 15 states (10 EM-iterations). For the probability-based DTW, the path restriction variable was set to $r = 0.2$ for both training and decoding. Factor x (see Equation 31) was chosen as 100 whereas factor f was set to 10 for speech decoding and to 1 for gesture decoding respectively. As Table 3 illustrates, HMM for speech recogni-

Classifier	**Speech** (# clas.)	**Gesture** (# clas.)
HMM	92.5% (11)	89.9%(10)
DTW	80.7% (11)	87.6%(10)

Table 3: Unimodal recognition rates (with number of different classes) of the individual classifiers for speech and gesture sequences

tion outperform the DTW by nearly 12%. The performance of unimodal gesture recognition is slightly better when using HMM.

6.3. Late Fusion

In order to classify the 26 different multimodal speech and gesture commands via dynamic time warping, first of all a late fusion strategy had been applied. Thereby the speech and the gesture stream were classified separately. A multimodal command is recognized correctly if the speech *and* the gesture stream had been assigned to the right class. As outlined before, this strategy does not exploit mutual information. Setting the parameters r, x, and f as in Section 6.2 leads to an average recognition rate of 62.08%.

6.4. Early Fusion

Using an early fusion DTW-based classification scheme, 459 multimodal symbols were introduced to merge the bimodal data into a unimodal representation prior to decoding. Aiming not to lose information during the early fusion process, every possible combination of 8 gesture symbols and 50 speech symbols has to be mapped to one multimodal symbol. Taking into account empty samples due to the asynchronity of the streams, we get $(8+1)\cdot(50+1) = 459$ different multimodal symbols. Classification was carried out using the parameters $r = 0.2$, $x = 100$, and $f = 10$. Due to the high degree of asynchronity of the data, early fusion performs worse than late fusion (average early fusion recognition rate: 46.84%).

6.5. Hybrid Fusion

As a realization of hybrid fusion, the three-dimensional DTW derived in Sections 5.1 and 5.3 has been tested on the speech and gesture data set. The training algorithm outlined in Section 5.3.2 was applied to acquire the extended reference vectors for each class. Both modalities were weighted equally ($x_a = x_b$, see Equation 33) and parameters x and f were chosen as in Section 6.2. The best average recognition rate (72.75%) could be attained for a path restriction variable $r = 0.15$ (see Figure 14). The 3D-DTW outperformed the four-dimensional dynamic time warping algorithm derived in Section 5.2 which achieved an average recognition rate of 57.70%. Thus, we can conclude that the increase of uncertainty due to the additional degrees of freedom for the four-dimensional backtracking path has a stronger influence on the recognition performance than the drawback of having to decide a priori about the alignment of the modes in the extended bimodal reference sequence as we do when applying the 3D-DTW.

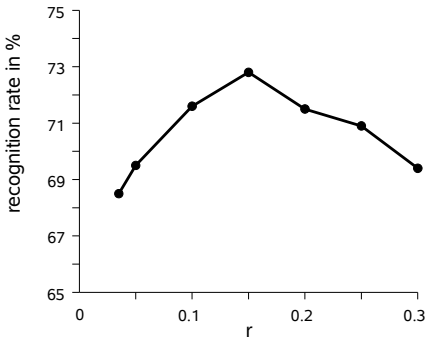

Figure 14: Recognition rate using the 3D-DTW versus path restriction variable r

6.6. Classification using Hidden Markov Models

We further applied different fusion strategies for HMM-based classification of the bimodal speech and gesture data set. Late fusion HMM reached an average recognition rate of 67.19% (15 word states; 5 gesture states), whereas due to the high degree of asynchronity of the data, early fusion HMM could not achieve rates better than guessing. The asynchronous Hidden Markov Model introduced in [7] was proven to be the best HMM-based strategy to fuse data, since it attained an average recognition rate of 77.62% (25 states).

6.7. Comparison of Fusion Strategies

Table 4 compares the recognition rates applying DTW-based and HMM-based structures to classify the speech and gesture data set. The results for HMM-classification can also be found in [2]. Using late fusion, mutual information can not be exploited since the modalities do not influence each other during the decoding process. However, late fusion systems profit from the strength of the individual classifiers. Due to the good performance of unimodal HMM-classification

Fusion strategy	(3D-)DTW	(A-)HMM
Late Fusion	62.08%	67.19%
Early Fusion	46.84%	3.85%
Hybrid Fusion	**72.75%**	**77.62%**

Table 4: Comparison of recognition rates (26 multimodal classes) for different fusion strategies using DTW-based or HMM-based classifiers; for hybrid fusion the 3D-DTW and the AHMM have been applied

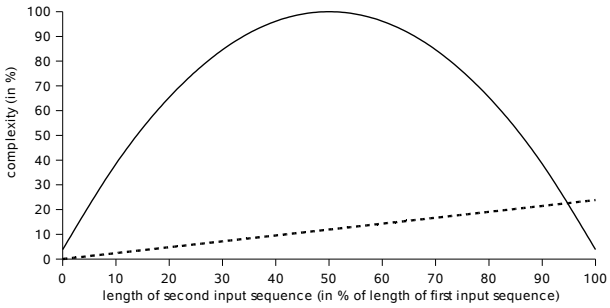

Figure 15: Complexity in % versus length of the second input stream (in % of the length of the first input stream) - Comparison of AHMM (curved line) and probability-based 3D-DTW (dashed straight lines) for a path restriction variable of $r = 0.15$; number of AHMM states: 25; number of samples of 3D-DTW reference sequence: 100

(see Table 3) late fusion HMM outperform dynamic time warping by 5%. Applying early fusion systems leads to recognition rates lower than 50% for both HMM and DTW, as these concepts are not able to generalize the high asynchronity in the data set. Comparing the performance of the asynchronous HMM and the probability-based three-dimensional DTW, which both are hybrid fusion concepts, we note that the recognition rate of the 3D-DTW is almost 5% (absolute) lower than the rate for the AHMM. However, if we compare late and hybrid fusion, the gain of performance exploiting mutual information in hybrid fusion systems is 10% for both HMM and DTW.

Determining the statistical significance of the difference between 3D-DTW and AHMM performance according to [20] results in a p-value of $3.4 \cdot 10^{-5}$. Thus, the performance difference can be seen as statistically significant, using the common significance level of 0.001.

The great advantage of the 3D-DTW becomes obvious if we take into account the computational complexity of the AHMM and the 3D-DTW: Figure 15 shows how complexity of both algorithms depend on the ratio of the lengths of the input streams. For typical parameter values that were also used in the speech and gesture classification experiment (path restriction variable $r = 0.15$, 25 AHMM states, 3D-DTW reference sequence length equal to 100) the complexity of the 3D-DTW is much lower, leading to a speed-up factor of up to 8.4. This makes the algorithm attractive for real-time applications and quick online adaptation by simple reference sequence addition.

7. Conclusion and Discussion

In this work a dynamic time warping algorithm has been extended in a way that it can model asynchronous multimodal data streams. The concept can be seen as a hybrid fusion approach that uses mutual information from other modalities during the decoding process. Since the algorithm is also applicable to data streams which are not synchronous, it combines the advantages of both late and early fusion. By adding a third dimension to the distance matrix, not only the temporal distortion between an input and a reference sequence, but also the optimal alignment between two input streams coming from different modalities is determined. Decoding is based on a distance measure that allows to weight the modes according to their importance for pattern classification. After some modifications of the derived 3D-DTW, it was shown that the algorithm can also carry out segmentation of bimodal data streams, even if they are not synchronous. Computational complexity could be reduced by restricting the valid domain of the backtracking path, which also excludes extreme time distortions. In order to overcome the problem of finding an accurate reference sequence for a certain class, a training procedure that iteratively improves the reference, has been developed. A further expansion of the DTW concept, that avoids merging the reference sequences of both modalities to a bimodal reference by adding a fourth dimension, could not convince in the experiments. Modifying the 3D-DTW to a probability-based DTW allows classifying bimodal sequences that consist of discrete symbols instead of usual pattern vectors. This probability-based DTW was compared to other statistical tools for multimodal classification, like the asynchronous Hidden Markov Model.

A challenging bimodal speech and gesture data set was used to evaluate the performance of the three-dimensional DTW algorithm and to compare it to other concepts. The 3D-DTW outperforms a late fusion DTW by more than 10% (17% relative improvement) but still

could not reach the recognition rate of the AHMM which is 5% higher (6.7% relative). However, both time and space complexity of the 3D-DTW is reduced by a factor of up to 8.4 with respect to the AHMM. Consequently, the 3D-DTW is an attractive alternative to AHMM whenever the speed of decoding or the amount of required memory is relevant.

Besides the performance gap between the proposed 3D-DTW and the AHMM, a few limitations of the initial algorithm as introduced in this work can be observed. One such limitation is the fact that the modality importance weight (factor g, see Section 5.1.2) is not learned automatically, but still has to be set by hand. Further, better distance measures and a more problem-adequate calculation of the accumulated 3D-DTW distance matrix might be included in future versions of the algorithm. A general limitation of the DTW concept is that major inter-class variations can only be captured by using more than one reference per class, which is not as elegant and effective as the Gaussian mixture approach for HMM-based modeling. Another shortcoming of the proposed 3D-DTW is the assumption of synchronized reference streams (in contrast to the input streams which do not have to be synchronized). However, with the 4D-DTW we presented a strategy to overcome this limitation.

Future research effort could be spent on the DTW-based classification of strongly correlated bimodal data, like speech and lip-movements. Furthermore, it would be interesting to examine how a slope constraint of the backtracking path affects the performance of the 3D-DTW. The performance gap between the 3D-DTW and the asynchronous HMM could be eliminated by including multiple references in a k-Nearest-Neighbor approach or by defining prototypes not only in the feature space but also for the stream alignment by appropriate clustering (such as k-means). For future works the principle of dimensionality expansion for hybrid fusion of multiple data streams may also be applied to more powerful classifiers such as Long Short-Term Memory Recurrent Neural Nets [23, 74] or Hidden Conditional Random Fields [46, 50]. Yet, the experiments in this work prove that the derived 3D-DTW algorithm is a promising approach of multimodal integration via dynamic time warping.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 211486 (SEMAINE).

References

[1] M. Ablassmeier, T. Poitschke, and G. Rigoll. Eye gaze studies comparing head-up and head-down displays in vehicles. In *Proceedings of ICME*, pages 2250–2252, 2007.

[2] M. Al-Hames and G. Rigoll. Reduced complexity and scaling for asynchronous HMMs in a bimodal input fusion application. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2006*, pages 757–760, 2006.

[3] F. Althoff, G. McGlaun, and M. K. Lang. Using multimodal interaction to navigate in arbitrary virtual vrml worlds. In *Proceedings of the 2001 workshop on perceptive user interfaces*, pages 1–8, 2001.

[4] D. Arsic, B. Schuller, and G. Rigoll. Suspicious behaviour detection in public transport by fusion of low-level video descriptors. In *ICME 2007*, pages 2018–2021, 2007.

[5] D. Arsic, F. Wallhoff, B. Schuller, and G. Rigoll. Vision based online behavior detection using probabilistic multi-stream fusion. In *ICIP 2005, Genova, Italy*, pages 606–609, 2005.

[6] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Fusion of face and speech data for person identity verification. In *IEEE Transactions on Neural Networks*, volume 10, pages 1065–1074, 1999.

[7] S. Bengio. An asynchronous hidden markov model for audio-visual speech recognition. *Advances in NIPS 15*, 2003.

[8] S. Bengio. Multimodal authentication using asynchronous HMMs. In *Proceedings of the IEEE AVBPA*, pages 770–777, 2003.

[9] Y. Bengio and P. Frasconi. An Input Output HMM architecture. In *Advances in Neural Information Processing Systems*, volume 7, pages 427–434, 1995.

[10] R. A. Bolt. Put-that-there: Voice and gesture at the graphics interface. In *Proceedings of SIGGRAPH*, pages 262–270, 1980.

[11] K. W. Bowyer. Face recognition technology: security versus privacy. In *Technology and Society Magazine, IEEE*, volume 23, pages 9–19, 2004.

[12] C. Bregler, H. Hild, S. Manke, and A. Waibel. Improving connected letter recognition by lipreading. In *Proceedings of ICASSP '93*, pages 557–560, 1993.

[13] T. Chen and R. R. Rao. Audio-visual integration in multimodal communication. In *Proceedings of the IEEE*, pages 837–852, 1998.

[14] A. Cheyer and L. Julia. Designing, developing and evaluating multimodal applications. In *Chi'99 (WS Pen/Voice Interface)*, pages 1–4, 1999.

[15] P. R. Cohen, M. Dalrymple, D. B. Moran, F. C. Pereira, and J. W. Sullivan. Synergistic use of direct manipulation and natural language. In *Human Factors in Computing Systems: CHI'89 Conference Proceedings*, volume 20, pages 227–233, 1989.

[16] R. F. de Mello and I. Gondra. Multi-dimensional dynamic time warping for image texture similarity. In *Advances in Artificial Intelligence*, pages 23–32, 2008.

[17] P. Duchnowski, U. Meier, and A. Waibel. See me, hear me: Integrating automatic speech recognition and lip-reading. In *International Conference on Spoken Language Processing, ICSLP 1994*, 1994.

[18] S. Dusan, G. J. Gadbois, and J. Flanagan. Multimodal interaction on PDAs integrating speech and pen inputs. In *Proceedings of EUROSPEECH 03*, pages 2225–2228, 2003.

[19] M. Fukumoto, Y. Suenaga, and K. Mase. Finger-pointer: pointing interface by image processing. In *Comput. Graph.*, volume 18, pages 633–642, 1994.

[20] L. Gillick and S. J. Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of ICASSP*, pages 23–26, 1989.

[21] H. Gunes and M. Piccardi. Affect recognition from face and body: early fusion vs. late fusion. In *IEEE International Conference on Systems, Man and Cybernetics, 2005*, volume 4, pages 3437–3443, 2005.

[22] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A. K. Jain. A background model initialization algorithm for video surveillance. In *Eighth International Conference on Computer Vision (ICCV'01)*, volume 1, page 733, 2001.

[23] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, pages 1735–1780, 1997.

[24] M. J. Hunt. Some experience in in-car speech recognition. In *IEEE Colloquium on Interactive Spoken Dialogue Systems for Telephony Applications (Ref. No. 1999/209)*, pages 1–9, 1999.

[25] F. Itakura. Minimum prediction residual principle applied to speech recognition. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 23, pages 67–72, 1975.

[26] F. Jiang, H. Yao, and G. Yao. Multilayer architecture in sign language recognition system. In *Proceedings of the 6th international conference on Multimodal interfaces*, pages 352–353, 2004.

[27] R. Kernchen, P. P. Boda, K. Moessner, B. Mrohs, M. Boussard, and G. Giuliani. Multimodal user interfaces for context-aware mobile applications. In *IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications, 2005*, volume 4, pages 2268–2273, 2005.

[28] M. H. Ko, G. West, S. Venkatesh, and M. Kumar. Using dynamic time warping for online temporal fusion in multisensor systems. *Information Fusion*, 9:370–388, 2008.

[29] W. Lizhong, S. Oviatt, and P. R. Cohen. Multimodal integration - a statistical view. In *IEEE Transactions on Multimedia*, volume 1, pages 334–341, 1999.

[30] I. S. MacKenzie and R. W. Soukoreff. Text entry for mobile computing: Models and methods, theory and practice. In *Human-Computer Interaction, 2002*, volume 17, pages 147–198, 2002.

[31] H. Misra, H. Bourlard, and V. Tyagi. New entropy based combination rules in HMM/ANN multi-stream ASR. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003*, volume 2, pages 741–744, 2003.

[32] P. Morguet. Comparison of approaches to continuous hand gesture recognition for a visual dialogue system. In *Proceedings of ICASSP 99*, pages 3549–3552, 1999.

[33] M. Müller, H. Mattes, and F. Kurth. An efficient multiscale approach to audio synchronization. In *Proceedings of ISMIR*, 2006.

[34] C. Myers and L. Rabiner. Connected digit recognition using a level-building dtw algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(3):351–363, 1981.

[35] N. U. Nair and T. V. Sreenivas. Joint deconding of multiple speech patterns for robust speech recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 9–13, 2007.

[36] J. G. Neal and S. C. Shapiro. Intelligent multimedia interface technology. In *Intelligent user interfaces*, pages 11–43, 1991.

[37] A. V. Nefian, L. Luhong, P. Xiaobo, X. Liu, C. Mao, and K. Murphy. A coupled HMM for audio-visual speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2013–2016, 2002.

[38] C. Neti, G. Potamianos, J. Luettin, I. Matthews, D. Vergyri, J. Sison, A. Mashari, and J. Zhou. Audio visual speech recognition. In *Final Workshop 2000 Report*, 2000.

[39] H. Ney. The use of a one-stage dynamic programming algorithm for connected word recognition. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 32, pages 263–271, 1984.

[40] L. Nigay and J. Coutaz. A generic platform for addressing the multimodal challenge. In *Proceedings of CHI '95*, pages 98–105, 1995.

[41] S. Oviatt. Multimodal interface research: A science without borders. In T. H. B. Yuan and X. Tang, editors, *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP 2000)*, volume 3, pages 1–6. Chinese Friendship Publishers, 2000.

[42] S. Oviatt and P. Cohen. Perceptual user interfaces: multimodal interfaces that process what comes naturally. In *Commun. ACM*, volume 43, pages 45–53, 2000.

[43] S. Oviatt, P. Cohen, L. Wu, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, and D. Ferro. Designing the user interface for multimodal speech and pen-based gestrue applications: State-of-the-art systems and future directions. In *Human Computer Interaction*, volume 15, pages 263–322, 2000.

[44] S. Oviatt, R. Coulston, S. Tomko, B. Xiao, R. Lunsford, M. Wesson, and L. Carmichael. Toward a theory of organized multimodal integration patterns during human-computer interaction. In *Proceedings of the 5th international conference on Multimodal interfaces*, pages 44–51, 2003.

[45] R. Plamondon and S. N. Srihari. Online and off-line handwriting recognition: a comprehensive survey. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 22, pages 63–84, 2000.

[46] A. Quattoni, S. Wang, L. P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 29, pages 1848–1853, 2007.

[47] L. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, 1989.

[48] L. Rabiner and B. Juang. An introduction to Hidden Markov Models. In *ASSP Magazine, IEEE*, volume 3, pages 4–16, 1986.

[49] L. Rabiner and S. Levinson. Isolated and connected word recognition - theory and selected applications. In *IEEE Transactions on Communications*, volume 29, pages 621–659, 1981.

[50] S. Reiter, B. Schuller, and G. Rigoll. Hidden conditional random fields for meeting segmentation. In *ICME 2007*, pages 639–642, 2007.

[51] D. A. Reynolds. An overview of automatic speaker recognition technology. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002*, volume 4, pages 4072–4075, 2002.

[52] N. Roy, G. Baltus, D. Fox, F. Gemperle, J. Goetz, T. Hirsch, D. Margaritis, M. Montemerlo, J. Pineau, J. Schulte, and S. Thrun. Towards personal service robots for the elderly. In *In Workshop on Interactive Robots and Entertainment (WIRE 2000)*, 2000.

[53] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 26, pages 43–49, 1978.

[54] J. Schenk, J. Lenz, and G. Rigoll. Line-members - a novel feature in on-line whiteboard note recognition. In *Proceedings of 11th Intern. Conf. on Frontiers in Handwriting Recognition*, 2008.

[55] L. Schomaker. From handwriting analysis to pen-computer applications. In *Electronics and Communication Engineering Journal*, volume 10, pages 93–102, 1998.

[56] B. Schuller, M. Ablassmeier, R. Mueller, S. Reifinger,

T. Poitschke, and G. Rigoll. Speech communication and multimodal interfaces. In *Advanced Man-Machine Interaction*, pages 141–190. Springer Verlag Berlin Heidelberg New York, 2006.

[57] B. Schuller, R. Mueller, B. Hoernler, A. Hoethker, H. Konosu, and G. Rigoll. Audiovisual recognition of spontaneous interest within conversations. In *Proceedings of the ACMICMI 2007, 9th Conf. on Multimodal Interfaces*, pages 30–37, 2007.

[58] B. Schuller, R. Müller, F. Eyben, J. Gast, B. Hörnler, M. Wöllmer, G. Rigoll, A. Höthker, and H. Konosu. Being bored? recognising natural interest by extensive audiovisual integration for real-life application. *to appear in Image and Vision Computing Journal (IMAVIS), Special Issue on Visual and Multimodal Analysis of Human Spontaneous Behavior, Elsevier*, page 17 pages, 2009.

[59] B. Schuller, M. Wimmer, D. Arsic, G. Rigoll, and B. Radig. Audiovisual behaviour modeling by combined feature spaces. In *Proceedings of ICASSP 2007*, pages 733–736, 2007.

[60] B. Schuller, M. Wöllmer, T. Moosmayr, and G. Rigoll. Speech recognition in noisy environments using a switching linear dynamic model for feature enhancement. In *Proceedings of Interspeech 2008*, pages 1789–1792, 2008.

[61] P. Sergio and C. L. Oliveira. DTW-based phonetic alignment using multiple acoustic features. In *Eurospeech 2003*, pages 309–312, 2003.

[62] R. Sharma, V. I. Pavlovic, and T. S. Huang. Toward multimodal human-computer interface. In *Proceedings of the IEEE, Special Issue on Multimedia Signal Processing*, volume 86, pages 853–869, 1998.

[63] M. Sodhi, B. Reimer, J. L. Cohen, E. Vastenburg, R. Kaars, and S. Kirschenbaum. On-road driver eye movement tracking using head-mounted devices. In *Proceedings of the 2002 symposium on Eye tracking research and applications*, pages 61–68, 2002.

[64] Y. Stettiner, D. Malah, and D. Chazan. Dynamic time warping with path control and non-local cost. In *Pattern Recognition*, volume 3, pages 174–177, 1994.

[65] Q. Summerfield. Lipreading and audio-visual speech perception. In *Philosophical Transactions: Biological Sciences*, pages 71–78, 1992.

[66] T. Tatschke. Early sensor data fusion techniques for collision mitigation purposes. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 445–452, 2006.

[67] G. A. ten Holt, M. J. T. Reinders, and E. A. Hendriks. Multi-dimensional dynamic time warping for gesture recognition. In *Annual Conference on the Advanced School for Computing and Imaging*, 2007.

[68] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing multidimensional time-series. *The VLDB Journal*, 15(1):1–20, 2006.

[69] M. T. Vo and A. Waibel. Multimodal human-computer interaction. In *Proceedings of ISSD '93*, Waseda, Japan, 1993.

[70] M. T. Vo and C. Wood. Building an application framework for speech and pen input integration in multimodal learning interfaces. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3545–3548, 1996.

[71] A. Waibel, M. T. Vo, P. Duchnowski, and S. Manke. Multimodal interfaces. In *Artificial Intelligence Review*, volume 10, pages 299–319, 1996.

[72] C. Ware and H. H. Mikaelian. An evaluation of an eye tracker as a device for computer input. In *Proceedings of the CHI + GI 87 Conference on Human Factors in Computing Systems and Graphics Interface*, volume 17, pages 183–188, 1987.

[73] M. Wimmer, B. Schuller, D. Arsic, B. Radig, and G. Rigoll. Low-level fusion of audio and video features for multi-modal emotion recognition. In *Proc. 3rd Int. Conf. on Computer Vision Theory and Applications*, pages 145–151, 2008.

[74] M. Wöllmer, F. Eyben, S. Reiter, B. Schuller, C. Cox, E. Douglas-Cowie, and R. Cowie. Abandoning emotion classes - towards continuous emotion recognition with modelling of long-range dependencies. In *Proceedings of Interspeech 2008*, pages 597–600, 2008.

[75] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, and G. Lathoud. Modeling individual and group actions in meetings: a two-layer HMM framework. In *Proceedings of the IEEE CVPR*, page 117, 2004.

[76] Y. Zhang, S. Levinson, and T. Huang. Speaker independent audio-visual speech recognition. In *IEEE International Conference on Multimedia and Expo, 2000*, volume 2, pages 1073–1076, 2000.

[77] M. Zobl, M. Geiger, B. Schuller, G. Rigoll, and M. Lang. A realtime system for hand-gesture controlled operation of in-car devices. In *Proceedings of ICME 2003*, pages 541–544, 2003.

Martin Wöllmer works as a researcher funded by the European Community's Seventh Framework Programme project SEMAINE (FP7/2007-2013) at the Technische Universität München (TUM). He obtained his diploma in Electrical Engineering and Information Technology from TUM where his current research and teaching activity includes the subject areas of pattern recognition and speech processing. His focus thereby lies on multimodal data fusion, automatic recognition of emotionally colored and noisy speech, and speech feature enhancement. His reviewing engagement includes the IEEE Transactions on Audio, Speech and Language Processing. Publications of his in various conference proceedings cover novel and robust modeling architectures for speech and emotion recognition such as Switching Linear Dynamic Models or Long Short-Term Memory Recurrent Neural Nets.

Marc Al-Hames finished his diploma and doctoral theses at the University of Cambridge (UK) and TUM (Germany). He worked on improved features,

Genetic Algorithms, and Graphical Models for Audiovisual Integration in the fields of Human-Computer Interaction and Meeting Analysis. His teaching activities are centred around the area of Signal Processing and Machine Learning. Project work of his comprises the European Community funded projects M4, AMI, and AMIDA, all dealing with multimodal meeting analysis. It is in these fields where his manifold publications on novel and improved fusion algorithms are found. At present he is working as a consultant and technology expert for the McKinsey group.

Florian Eyben works on a research grant as part of the European Community's Seventh Framework Programme project SEMAINE (FP7/2007-2013) - the Sensitive Artificial Listener project - within the Institute for Human-Machine Communication at TUM. He obtained his diploma in Information Technology from TUM. Teaching activities of his comprise Pattern Recognition and Speech and Language processing. His research interests include large scale hierarchical audio feature extraction and evaluation, automatic emotion recognition from the speech signal, recognition of non-linguistic vocalizations, automatic continuous large vocabulary speech recognition, statistical and context-dependent language models, and Music Information Retrieval. He has several publications in various journals and conference proceedings covering many of his areas of research.

Björn Schuller received his diploma and his doctoral degrees in electrical engineering and information technology from TUM, where he currently stays as lecturer in Pattern Recognition. He authored more than 100 publications in books, journals, and peer reviewed conference proceedings in this field. Best known are his works advancing Audiovisual Processing in the areas of Affective Computing and Multimedia Retrieval. He served as associate editor and reviewer for several scientific journals, including the Elsevier Signal Processing, Neurocomputing, Pattern Recognition Letters, Computer Speech and Language, Speech Communication, and Image and Vision Computing Journals, and as invited speaker, session organizer and chairman, and programme committee member of numerous international conferences. Current project steering board activities include SEMAINE funded by the European Community and further projects with companies as BMW, Continental, Daimler, Siemens, Toyota, and VDO. He is invited expert in the W3C Emotion and Emotion Markup Language Incubator Groups, and elected member of HUMAINE Association Executive Committee.

Gerhard Rigoll received his diploma in Technical Cybernetics (1982), his Ph.D. for his works in the field of Automatic Speech Recognition (1986), and his habilitation in the field of Speech Synthesis (1991) from University of Stuttgart/Germany. He worked for the Fraunhofer-Institute Stuttgart, Speech Plus in Mountain View/SA, and Digital Equipment in Maynard/USA, spent a post-doctoral fellowship at IBM Thomas Watson Research Center, Yorktown Heights/USA, headed a research group at Fraunhofer-Institute Stuttgart and spent a two year's research stay at NTT Human Interface Laboratories in Tokyo/Japan (1986), in the area of Neuro-computing, Speech Recognition and Pattern Recognition until he was appointed full professor of Computer Science at Gerhard-Mercator-University Duisburg/Germany (1993) and of Human-Machine Communication at TUM (2002). He is a senior member of the IEEE and authored and co-authored more than 250 publications in the field of signal processing and pattern recognition. Most of his work deals with Automatic Speech Recognition, where he is particularly concerned with classifier optimization. He also maintains active research programs in vision-based pattern recognition.