Audio synchronisation with a tunnel matrix for time series and dynamic programming

Jan Gorisch, Laurent Prevot

AUDIO SYNCHRONISATION WITH A TUNNEL MATRIX FOR TIME SERIES AND DYNAMIC PROGRAMMING

Jan Gorisch[⋆†] *Laurent Prévot*[⋆]

[⋆]Aix Marseille Université & CNRS, Laboratoire Parole et Langage, France
[†]Nanyang Technological University, Division of Linguistics and Multilingual Studies, Singapore

ABSTRACT

Precise multimodal studies require precise synchronisation between audio and video signals. However, raw audio and audio from video recordings can be out of sync for several reasons. In order to re-synchronise them, a dynamic programming (DP) approach is presented here. Traditionally, DP is performed on the rectangular distance matrix comparing each value in signal A with each value in signal B. Previous work limited the search space using for example the Sakoe Chiba Band (Sakoe and Chiba, 1978). However, the overall space of the distance matrix remains identical. Here, a tunnel matrix and its according DP-algorithm are presented. The matrix contains merely the computed distance of two signals to a pre-specified bandwidth and the computational cost is equally reduced. An example implementation demonstrates the functionality on artificial data and on data from real audio and video recordings.

Index Terms— Audio-video Synchronisation, Image-loss Compensation, Tunnel Matrix, Tunnel DP-algorithm, Storage Requirements

1. INTRODUCTION

The number of studies that analyse audio and video signals has increased with the attempt to account for the multimodality of social interaction involving amongst others speech and gestures [1, 2] and investigations into the natural habitat of speech: face-to-face interaction. Although conversation analysts often have to restrain themselves to low quality audio and video recordings in order to preserve the naturaleness of talk-in-interaction [3], studies in interactional phonetics [4] that combine qualitative and quantitative methods by applying computation techniques on automatically extracted features, e.g. Kurtic et al. [5] Gorisch et al. [6] and Bertrand et al. [7], require high quality audio recordings from close-speaking microphones. The audio has to be synchronised with the video recordings, if the multimodality of talk should be analysed for the same data. Usually, precautions are taken to ensure high standards for audio-video synchronisation, such as by Edlund et al. [8] who enhanced their recording setup and included a rotating turntable with an LP-disk that was scratched at a certain place and a visible marker. Audio and video signals were then synchronised in subsequent steps in order to render the data analysable. The analysis of desynchronised audio and video signals, however, is almost impossible. One source of error can be a clocking difference of two recording devices that capture the same scene. Another source of error can be the hardware equipment itself. For example, video recordings or capturing systems that imply MiniDV technology regularly introduce errors, i.e. a loss or surplus of image frames. Both problems were encountered with recordings of a multimodal speech corpus [9]. Previously, such problems have been tackled by inserting or deleting frames at specific places by hand, as was performed for the CID corpus [10]. A more efficient and re-usable solution that does not have to involve additional equipment, such as turntables, was envisaged for the alignment of the signals, as e.g. of the audiovisual MapTask corpus [9].

2. APPROACH

The advantage of video recordings is that the camera usually includes an audio channel that captures the same scene as the separate audio recordings do. It means that two similar signals of the same quality, i.e. acoustics, exist for the computation of their alignment. Thus, the aim for synchronising audio and video is to insert or delete images from the video according to the alignment of the audio signals that stem from two different devices.

Dynamic Programming (DP) and Dynamic Time Warping (DTW) are methods that can be applied here, as they are used to align time series in an optimal way. They allow to determine where one signal has to be stretched in time or shrunk in time in order to match the other signal [11]. These methods had their revival in speech processing in recent years, e.g. with algorithms that search for re-occurring acoustic patterns for modelling infant speech acquisition [12] and algorithms that evaluate the similarity in prosodic patterns of conversa-

Thanks to the ANR funding the project "Conversational Feedback" (grant number ANR-12-JCJC-JSH2-006-01). This research was supported in part by the Erasmus Mundus Action 2 program MULTI of the European Union, grant agreement number 2010-5094-7. Comments from Stéphane Rauzy were very much appreciated.

tional participants performing basic social actions [13]. Developments have been performed by Salvador and Chan [14] who increased the time efficiency of DTW algorithms. They solved a similar problem by matching two acoustic signals in order to synchronise them with a video signal. However, a problem of limited space and computational cost remains unsolved.

2.1. Space and computational cost problem of DP

Many DP problems today take large time series as underlying data. For example, an attempt to align audio recordings of the length of half an hour (1,800s), even using a relatively sparse sampling rate of 25 samples/s (as standard video frames), results in a distance- or similarity matrix of $(25 \times 1,800)^2$, that are 2,025,000,000 value pairs. Additional to the space requirements of this matrix comes the computational cost of dynamic programming to evaluate the best alignment path of the two underlying vectors. The Sakoe Chiba Band [15] or the Itakura Parallelogram [16] are techniques that can help to reduce that computation by limiting the search space to a band around the diagonal of that matrix from the start point to the end point of the underlying vectors.

One problem of current matrix processing is the limited storage space that is available on e.g. personal computers. Sometimes, it is physically not possible to create the distance matrix for two time series. Fortunately, this is even not necessary because the majority of values are not required in the case for the audio-out-of-sync problem. If the DP algorithm that is supposed to be applied to such data is not required to look at the edges of the distance matrix, that space is used without need. A sparse matrix, as introduced by Gilbert et al. (1992) [17], could be used in order to implement such a band. However, DP has not been demonstrated on sparse matrices, yet. In order to save space and computation power to solve the desynchronisation problem at hand, a new structure for the values of the distance matrix is suggested in section 3.

2.2. Conceptual and aesthetic problem

A more conceptual and probably aesthetic problem of a rectangular distance matrix (1) is the distribution of the axes.

$$\begin{array}{|ccccccccc|}
\hline
a_1b_1 & a_1b_2 & a_1b_3 & a_1b_4 & a_1b_5 & a_1b_6 & a_1b_7 & a_1b_8 & a_1b_9 \\
a_2b_1 & a_2b_2 & a_2b_3 & a_2b_4 & a_2b_5 & a_2b_6 & a_2b_7 & a_2b_8 & a_2b_9 \\
a_3b_1 & a_3b_2 & a_3b_3 & a_3b_4 & a_3b_5 & a_3b_6 & a_3b_7 & a_3b_8 & a_3b_9 \\
a_4b_1 & a_4b_2 & a_4b_3 & a_4b_4 & a_4b_5 & a_4b_6 & a_4b_7 & a_4b_8 & a_4b_9 \\
a_5b_1 & a_5b_2 & a_5b_3 & a_5b_4 & a_5b_5 & a_5b_6 & a_5b_7 & a_5b_8 & a_5b_9 \\
a_6b_1 & a_6b_2 & a_6b_3 & a_6b_4 & a_6b_5 & a_6b_6 & a_6b_7 & a_6b_8 & a_6b_9 \\
a_7b_1 & a_7b_2 & a_7b_3 & a_7b_4 & a_7b_5 & a_7b_6 & a_7b_7 & a_7b_8 & a_7b_9 \\
\hline
\end{array} \quad (1)$$

While one time series evolves along the x-axis and the other time series evolves along the y-axis, the overall time evolves along the diagonal. This is traditionally visualised with two vectors, e.g. $\vec{a} = a_1a_2 \ldots a_7$ and $\vec{b} = b_1b_2 \ldots b_9$ that combine in a distance matrix as shown in (1) with time elapsing for $\vec{a}$ top–down, for $\vec{b}$ left–right and for overall time from top–left to bottom–right. From a conceptual point of view where time elapses from left to right, or from top to bottom, the traditional matrices, as they are produced by Matlab® or R, are not optimal in this sense. Both, the reduction of the search space and the orientation of the time axis can be realised by introducing a tunnel matrix that represents the Sakoe Chiba Band (shaded area in (1)) rotated by 45 degrees to the left for horizontal or to the right for vertical propagation of time. This paper proposes an implementation of such a distance matrix in tunnel form at the initialisation stage, i.e. already before the calculation of the alignment path.

3. IMPLEMENTATION OF A TUNNEL MATRIX

The aim is to rotate the Sakoe Chiba Band from a traditional distance matrix around the diagonal by -45 degrees in order to let time elapse from left to right. As can be seen in (1), the diagonal contains the isochronous pairs a_1b_1, a_2b_2, ... a_7b_7. This should be the mid row of the new matrix. It can also be seen that the values of $\vec{a}$ are constant for each column of and increase from row to row. For $\vec{b}$ it is the inverse: the values are constant for each row and increase from column to column. In order to illustrate the new tunnel arrangement, two intermediate matrices are prepared. For all values of $\vec{a}$ and $\vec{b}$ that fall into the bandwidth of the Sakoe Chiba Band, a 'tunnel A' (2) and a 'tunnel B' (3) is created.

$$\begin{array}{|ccccccccc|}
\hline
\text{NA} & \text{NA} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 \\
\text{NA} & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & \text{NA} \\
a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & \text{NA} & \text{NA} \\
a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & \text{NA} & \text{NA} \\
a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & \text{NA} & \text{NA} \\
\hline
\end{array} \quad (2)$$

$$\begin{array}{|ccccccccccc|}
\hline
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & \text{NA} & \text{NA} \\
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & \text{NA} & \text{NA} \\
b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & \text{NA} & \text{NA} \\
\text{NA} & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & \text{NA} \\
\text{NA} & \text{NA} & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 \\
\hline
\end{array} \quad (3)$$

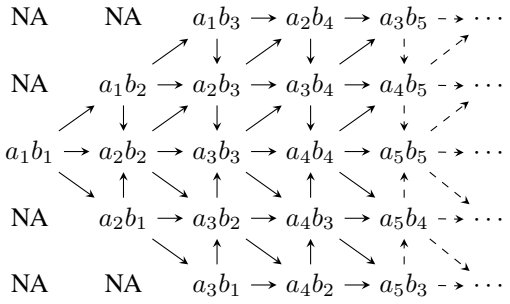
For illustration purposes, a width of three values in each dimension was chosen. Both intermediate matrices are then merged in order to create the final AB Tunnel Matrix by cross-multiplying 'tunnel A' and 'tunnel B', as shown in (4). Note that these values represent here similarity instead of distance.

$$\begin{array}{|ccccccccc|}
\hline
\text{NA} & \text{NA} & a_1b_3 & a_2b_4 & a_3b_5 & a_4b_6 & a_5b_7 & a_6b_8 & a_7b_9 \\
\text{NA} & a_1b_2 & a_2b_3 & a_3b_4 & a_4b_5 & a_5b_6 & a_6b_7 & a_7b_8 & \text{NA} \\
a_1b_1 & a_2b_2 & a_3b_3 & a_4b_4 & a_5b_5 & a_6b_6 & a_7b_7 & \text{NA} & \text{NA} \\
\text{NA} & a_2b_1 & a_3b_2 & a_4b_3 & a_5b_4 & a_6b_5 & a_7b_6 & \text{NA} & \text{NA} \\
\text{NA} & \text{NA} & a_3b_1 & a_4b_2 & a_5b_3 & a_6b_4 & a_7b_5 & \text{NA} & \text{NA} \\
\hline
\end{array} \quad (4)$$

4. THE TUNNEL DP-ALGORITHM

For the newly created similarity matrix, the traditional DP algorithms are inappropriate. The start point (a_1b_1) and the end point (a_7b_9) are no more at the same location. And therefore, the possible directions that an alignment path might take are also different. For a traditional similarity matrix, the most common step-pattern is from a_1b_1 to a_2b_2 without insertion or deletion, to a_1b_2 by insertion in $\vec{a}$ or to a_2b_1 by insertion in $\vec{b}$ (or deletion from $\vec{a}$).

This translates into the new tunnel matrix, however it has to be adapted depending where in the matrix the algorithm is currently working. The principle is the same: it is merely possible to continue in both vectors or to stay at the same value in one vector while proceeding in the other. In the mid row of the tunnel matrix, imagine to start at a_1b_1 in (5), the steps indicated by arrows are possible. In the top half of the matrix (imagine the prior step went from a_1b_1 to a_1b_2), one arrow points into a different direction. That means the possible steps are different. In the bottom half of the matrix (imagine the prior step went from a_1b_1 to a_2b_1), again, arrows point in different directions.



(5)

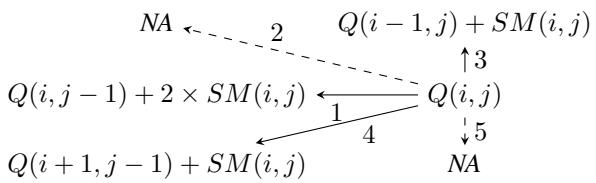
In order to find the path through the whole Similarity Matrix with the highest quality (inverse of cost), DP accumulates for each point of the matrix the maximum quality path up to that point in a Quality Matrix. Therefore, at each point of that matrix, the previously accumulated quality is added to the current similarity. The direction from where the lowest cost path came from is memorised in order to find the way back, once the whole matrix has been passed. The algorithm for filling the Quality Matrix according to the similarity matrix is again different from the traditional procedure. It is not possible to proceed row wise and column wise from one corner of the matrix (formerly the start point) to the opposite corner (formerly the end point), even if the step-patterns are clear for each point. Because the DP algorithm always looks backwards from the current point in order to evaluate the highest quality path and accumulates in the current point the new quality-to-this-point, all possible points from where the path can come from need already be evaluated. If they are not, the new evaluation would be based on both, points that include the accumulated similarities (the quality) and points that have stored the mere individual similarity. This would bias the decision in favour of the latter. A solution is suggested in the following section.

4.1. Quality Tunnel Matrix and Backtrack Matrix

A Quality Matrix Q stores the accumulated similarities. A Backtrack Matrix ϕ stores at each point the direction from where the path with the highest quality came from. In a first step, Q is filled with the values of the similarity matrix SM and surrounded on top, bottom and left column, by a line of NAs (Not Applicable). This border is necessary in order to have initial values to be compared with. Once the DP algorithm has finished, the end point of that matrix contains the overall quality for the alignment of the two vectors $\vec{a}$ and $\vec{b}$. The algorithm iterates over the columns via j and over the rows via i. $Q(i,j)$ is the point in the Quality Matrix for which the accumulated quality is to be calculated, i.e. the maximum quality that accumulates until that point. As mentioned above, it is important that each time a potentially proceeding path is evaluated for the maximum quality, the values are either part of the border, or already filled. Therefore, a specific evaluation pattern is established.

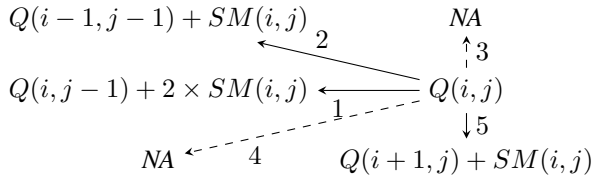
An overall trend from left to right is used in order to reach from the start in the first column to the end in the last column. The rows are divided into three parts. The first part is the top half of the matrix, the second part is the bottom part of the matrix and the third part is the mid row. The rows of the top and bottom part are evaluated from the outside to the middle. For the top part (6), it means that i increases from 2 to the $middle - 1$ (row above the mid). For the bottom part (7), it means that i decreases from the $end - 1$ (previous last row) to the $middle + 1$ (row below the mid). At the last iteration per column, the value of the mid row is evaluated (8).

The following comparisons are performed for the three parts at each point $Q(i,j)$. *NA*s (Not Appliccable) are inserted in the diagrams in order to keep the indexing function for the backtrack path constant: The horizontal step is always 1, the vertical steps are always 3 (top) and 5 (down), and the diagonal steps are always 2 and 4. The lowest value is added to the current point $Q(i,j)$. The direction from where this highest quality path came from is added to the Backtrack Matrix at point $\phi(i,j)$. Arrows with solid lines indicate valid steps; arrows with dashed lines indicate invalid steps. **The top half of the matrix** (row 2 to $mid - 1$) evaluates $Q(i,j) = max(Q(i,j-1)+2\times SM(i,j), NA, Q(i-1,j)+SM(i,j), Q(i+1,j-1)+SM(i,j), NA)$.



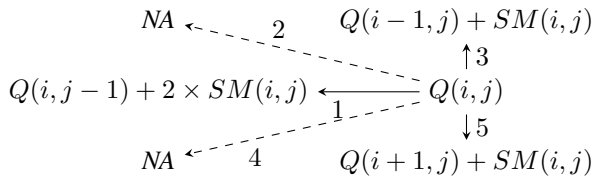
(6)

The bottom half of the matrix (row $end-1$ to $mid+1$): $Q(i,j) = max(Q(i,j-1)+2\times SM(i,j), Q(i-1,j-1)+SM(i,j), NA, NA, Q(i+1,j)+SM(i,j))$.



(7)

The mid row of the matrix is analysed last per column iteration and evaluates: $Q(i,j) = max(Q(i,j-1)+2\times SM(i,j), NA, Q(i-1,j)+SM(i,j), NA, Q(i+1,j)+SM(i,j))$.



(8)

5. EVALUATION

The new tunnel DP-algorithm applied to the tunnel matrix structure is evaluated on both artificial data and real data from recordings. The artificial data (invented for demonstration purposes) contains a matrix of values between 0.1 and 0.9 that are represented on a grey scale (left of Figure 1). The

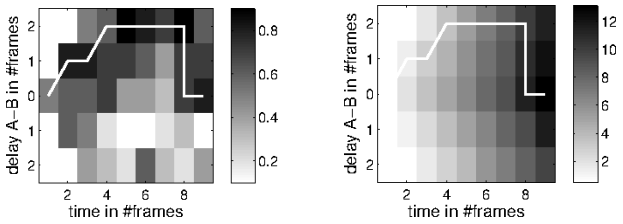


Fig. 1. Evaluation of the new DP algorithm on artificial data. Similarity Tunnel Matrix left, Quality Tunnel Matrix right.

DP-algorithm found an optimal alignment path (white line) by accumulating similarity scores in a quality matrix (right of Figure 1). The path through the regions of relatively high similarity demonstrates that the algorithm works well on artificial data. Similar results are achieved on real data from the MapTask corpus [9] (using spectral information) from recordings from devices with clock differences and random image loss (Figure 2). It can be seen that after approximately 4000 frames (25/s), one signal is approximately 27 frames behind, or the other ahead respectively. The Tunnel Matrix and the Tunnel DP-algorithm achieve alignment paths (as output) that

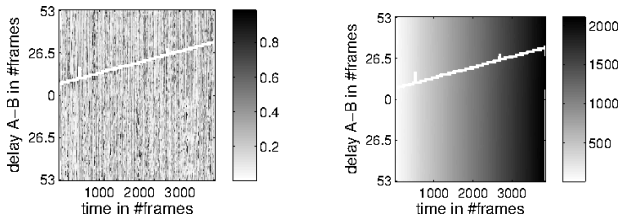


Fig. 2. Evaluation of the new DP algorithm on real data. Similarity Tunnel Matrix left, Quality Tunnel Matrix right.

follow high similarity regions as expected. The path can be used in order to align faulty video streams to intact audio because the areas of image loss or surplus are well indicated. This method was successfully used to synchronise 90% of the recordings. The remaining 10% could not be recovered yet, as entire minutes of the video are lacking.

6. DISCUSSION

Matlab allows to indicate which parts of a matrix do not need to be stored in a "sparse" matrix [17], i.e. areas of numbers that are zero. However, DP on such sparse matrices has not been demonstrated yet. It means that the proposed architecture for optimised use of space is still lacking in programming languages such as Matlab. Even the function `spdiags()` does not achieve this. A functioning workaround is demonstrated here that applies a tunnel matrix with the according DP algorithm for synchronising audio signals. There are limitations, as for example gaps of the video spanning several minutes, that increase again the space requirements drastically. But as long as the errors in the signal remain within reasonable limits (the range of several seconds), the algorithm can calculate an output (an alignment path) that can be used subsequently to align the video, e.g. by calling FFmpeg (http://ffmpeg.org) to copy or delete frames.

7. IMPLICATIONS AND APPLICATIONS

Hardware failure of recording equipment is besides human failure a source for frustration and cost explosion for data repair. Previous work with the aim to prevent this has introduced even more hardware [8] that can also fail and that decreases the naturalness of social interaction additionally to the recording situation itself. Data that was recorded with less precautions should still be made available at a maximum possible quality. One further step into this direction is achieved with the technique suggested in this paper. This DP-technique can also be applied in fundamental research that investigates the similarity of interacting participants on multiple parameters: acoustic-prosodic features investigating social alignment on a turn-by-turn basis [6] or on a larger time-scale as in entrainment [18] or convergence [19], EEG and EMA signals or simple re-occurring patterns in the acoustics [20].

8. REFERENCES

[1] Tanya Stivers, "Stance, alignment, and affiliation during storytelling: When nodding is a token of affiliation," *Research on Language and Social Interaction*, vol. 41, no. 1, pp. 31–57, 2011.

[2] Sigrid Norris, *Analyzing Multimodal Interaction: A Methodological Framework*, Routledge, New York, USA, 2004.

[3] Ian Hutchby and Robin Wooffitt, *Conversation Analysis*, Polity, Malden, MA, USA, 2008.

[4] John Local, "Phonetic detail and the organisation of talk-in-interaction," in *Proceedings of the Sixteenth International Congress of Phonetic Sciences (ICPhS)*, Saarbrücken, Germany, 2007.

[5] Emina Kurtic, Guy J. Brown, and Bill Wells, "Resources for turn competition in overlapping talk," *Speech Communication*, vol. 55, no. 5, pp. 721–743, 2013.

[6] Jan Gorisch, Bill Wells, and Guy J. Brown, "Pitch contour matching and interactional alignment across turns: an acoustic investigation," *Language and Speech*, vol. 55, no. 1, pp. 57–76, 2012.

[7] Roxane Bertrand, Gaëlle Ferré, Philippe Blache, Robert Espesser, and Stéphane Rauzy, "Backchannels revisited from a multimodal perspective," in *Proceedings of the International Conference on Auditory-Visual Speech Processing (AVSP2007)*, Kasteel Groenendaal, Hilvarenbeek, The Netherlands, 2007.

[8] Jens Edlund, Jonas Beskow, Kjell Elenius, Kahl Hellmer, Sofia Strömbergsson, and David House, "Spontal: A Swedish spontaneous dialogue corpus," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Malta, 2010.

[9] Jan Gorisch, Corine Astésano, Ellen Gurman Bard, Brigitte Bigi, and Laurent Prévot, "Aix Map Task corpus: the French multimodal corpus of task-oriented dialogue," in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, 2014.

[10] Roxane Bertrand, Philippe Blache, Robert Espesser, Gaëlle Ferré, Christine Meunier, Béatrice Priego-Valverde, and Stéphane Rauzy, "Le CID – Corpus of Interactional Data – Annotation et exploitation multimodale de parole conversationnelle," *Traitement Automatique des Langues (TAL)*, vol. 49, no. 3, pp. 105–134, 2008.

[11] Lawrence Rabiner and Biing-Hwang Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Inc., Englewood Cliffs, NJ, USA, 1993.

[12] Guillaume Aimetti, *A Computational Model of Early Language Acquisition: A Data-driven Approach Inspired by the Empiricist View of Cognitive Development*, Ph.D. thesis, The University of Sheffield, 2011.

[13] Jan Gorisch, *Matching across Turns in Talk-in-Interaction: The Role of Prosody and Gesture*, Ph.D. thesis, The University of Sheffield, 2012.

[14] Stan Salvador and Philip Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," in *The Third SIGKDD Workshop on Mining Temporal and Sequential Data (KDD/TDM 2004)*, Seattle, WA, USA, 2004, pp. 70–80.

[15] Hiroaki Sakoe and Seibi Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[16] Fumitada Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 23, no. 1, pp. 67–72, 1975.

[17] John R. Gilbert, Cleve Moler, and Robert Schreiber, "Sparse matrices in matlab: Design and implementation," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, pp. 333–356, 1992.

[18] Mathias Heldner, Jens Edlund, and Julia Hirschberg, "Pitch similarity in the vecinity of backchannels," in *Proceedings of Interspeech 2010*, Makuhari, Japan, 2010.

[19] Spyros Kousidis, David Dorran, Yi Wang, Brian Vaughan, Charlie Cullen, Dermot Campbell, Ciaran McDonnell, and Eugene Coyle, "Towards measuring continuous acoustic feature convergence in unconstrained spoken dialogues," in *Proceedings of Interspeech 2008*, Brisbane, Australia, 2008.

[20] Guillaume Aimetti, Roger K. Moore, and Louis ten-Bosch, "Discovering an optimal set of minimally contrasting acoustic speech units: a point of focus for whole-word pattern matching," in *Proceedings of Interspeech 2010*, Makuhari, Japan, 2010.