# Additional use of phoneme duration hypotheses in automatic speech segmentation

*Karlheinz Stöber, Wolfgang Hess*

Institut für Kommunikationsforschung und Phonetik
Rheinische Friedrich-Wilhelms Universität, 53115 Bonn
Email: kst@ikp.uni-bonn.de

## ABSTRACT

In this paper, we describe a new approach for speaker independent automatic phoneme alignment. Typical algorithms for this task use only phoneme-to-frame similarity measures which are somehow maximised or minimised. In addition to such similarity measures, we use phoneme duration hypotheses generated by the speech synthesis system HADIFIX [1]. For algorithms based on dynamic programming, it is difficult to use these duration hypotheses, so we create a cost-function consisting of phoneme-to-frame and segment-to-duration hypotheses similarity measures and minimise this cost-function by a Genetic Algorithm. The results show that the accuracy of automatically determined phoneme boundaries increases. This accounts especially for speakers not used in the training phase.

## 1. INTRODUCTION

For our work on the speech synthesis system HADIFIX, we need a lot of phonemically labelled speech signals. The construction of one synthesis inventory is based on 2.500 phonemically labelled spoken utterances. Many more phonemically labelled speech signals are used for research within the generation of duration hypotheses or of $F_0$ contours. For this work, an automatic phoneme alignment algorithm with only a small error rate is necessary. The question of what counts as a small error rate is very difficult to answer, because speech is a continuous process and it is normally impossible to create an exact projection between speech signals and corresponding phonemes, the latter being discrete units. This fact is even true for utterances labelled by phonetic experts which is shown by experiments where experts labelled the same speech material independently from each other [2]. Usually, the measure of quality of an automatic alignment algorithm is the distance between manually and automatically determined phoneme boundaries. We do the same in this paper. Nevertheless, we raise the question whether phonetic experts behave differently when a) they have to label a raw speech signal or b) the speech signal is already annotated with the results of the automatic alignment process.

Using only phoneme-to-frame similarity measures for speaker independent phoneme alignment is a difficult task, because on the one hand, the used phoneme models should permit speaker dependent realisations of phonemes. But on the other hand, they should be highly discriminative between similar sounds, e.g. sounds like /n/ and /m/[1] which are difficult to separate on the basis of their spectral representation only. Especially in such cases the use of phoneme duration hypotheses as additional constraints for the alignment process solves the problem much better. The integration of duration hypotheses into a conventional alignment algorithm based on dynamic programming is a difficult (if not impossible) task, therefore we use a Genetic Algorithm (GA) to solve our alignment problem.

[1] In this paper we use the SAMPA symbols for phonemes

## 2. THE FITNESS-FUNCTION

Many speech recognition and speech alignment algorithms are based on the minimisation of a cost-function (according to the terminology of GAs we call this function *fitness-function*). Especially algorithms based on Hidden Markov Models (HMMs) are of this type. Here the cost-function is often a complex mixture of density functions but only used to describe phoneme-to-frame similarities. Another interesting property of frequently used single-skip HMMs is that for each state transition probabilities are reduced to a constant factor. These factors weight the emission probabilities of the corresponding state. It can easily be shown that this factor may be integrated in the emission probabilities directly.

Due to this observation we use self-organizing maps (SOMs) [3] in order to model phoneme-to-frame similarities. As in semi-continuous HMMs, the distance between the parametric representation of a frame and the matching entry in the SOMs is used to define the similarity measure. As opposed to semi-continuous HMMs, we use one SOM (codebook) for each phoneme class to increase the distances between the parametric representations of different phonemes stored in the SOMs [4]. Let $W_\gamma$ be the SOM for phoneme $\gamma$ and $\mathbf{L} = l_1, ..., l_K$ the phoneme sequence realised in the speech signal $\mathbf{S}$. The parametric representation of $\mathbf{S}$ is given by $\mathbf{P} = p_1, ..., p_M$. Now we can define a distance matrix $\Delta = (\Delta_{ij})$.

$$\Delta_{ij} := \min_{o}\left(W_{l_i}(o) - p_j\right) \qquad (1)$$

$$1 \le i \le K,\ 1 \le j \le M$$

We compute that path in the distance matrix $\Delta$ from (1,1) to ($K$, $M$) for which the sum of the matrix entries is minimal out of all possible paths. This path solves the segmentation problem in a way similar to the solution given by semi-continuous single-skip HMMs.

The durations of phonemes depend on many factors such as the speaker, the surrounding phonemes, the position in a phrase etc. These influences are considered by HADIFIX during the computation of the duration hypotheses.

Let H = $\eta_1$, ..., $\eta_K$ be the computed duration hypotheses for the phoneme sequence **L**. First we have to adapt these duration values in relation to the length of the speech signal **S**. This is done by a linear scaling procedure. The scaled duration hypotheses[2] are called **D** = $d_1$, ...,$d_K$ where $d_i = c \cdot \eta_i$. Next we have to consider that the real phoneme duration may still differ slightly from the scaled hypotheses. This fact is modelled in (2) by the constant *a*. (2) uses the ratio between the scaled duration hypotheses $d_i$ and a given segment duration $u$ to assess the deviation of the current segmentation durations from the hypothesised durations. Summing (2) over all phoneme boundaries contained in the speech signal *S* shows that there are many possible segmentations which form a minimum of (2). The number of these segmentations obviously depends on *a*. The different segmentation possibilities account for the fact that phoneme durations do not necessarily have to be equal for different speech signals with the same content.

$$\mathrm{dur}(u,v) := \begin{cases} 0 & \text{if } \left|\frac{u}{v} - 1\right| < a \\ \left|\frac{u}{v} - 1\right| & \text{otherwise} \end{cases} \quad (2)$$

Now we can define the fitness-function **f** for a given vector $\alpha = \alpha_1, ..., \alpha_{K+1}$ containing the segment boundaries for the speech signal **S**. The beginnings of the segments pertinent to the phonemes $l_1$, ..., $l_K$ are stored in $\alpha_1$, ...,$\alpha_K$, whereas $\alpha_{K+1}$ contains the end of the last phoneme $l_K$. The fitness function **f** is then computed by summing (1) over all frames and phonemes and by summing (2) over the deviations between all segments defined by $\alpha$ and the duration hypotheses.

$$\mathrm{f}(\alpha) := \frac{\lambda_1}{\sum_{i=1}^{K} \alpha_{i+1} - \alpha_i} \sum_{i=1}^{K+1} \sum_{j=\alpha_i}^{\alpha_{i+1}} \Delta_{ij} + \frac{\lambda_2}{K} \sum_{i=1}^{K} \mathrm{dur}(\alpha_{i+1} - \alpha_i, d_i) \quad (3)$$

The constants $\lambda_1$ and $\lambda_2$ are used in experiments to adjust the importance of both, the term for phoneme-to-frame similarity and the term of duration hypotheses to segment length measures.

## 3. APPLYING GENETIC ALGORITHMS

Genetic Algorithm [5] are based on a set of so-called cells (or genomes), where each cell represents a possible solution of the problem to be solved. The set of cells is called a population. In an iterative process the cells are adapted to the nature which is coded in the fitness-function. For our problem we say that the cell $c_1$ is a better solution of our problem than the cell $c_2$ if $\mathbf{f}(c_1) < \mathbf{f}(c_2)$. This induces an order on the population. Starting with an initial population we carry out the recombination and mutation step. The cells contained in the population before applying the recombination step are called the parent generation. Since the recombination step is used to create new cells, we need an additional rule to keep the size of the population constant. The rule we use is called *survival of the fittest* [5]. This rule states that only a constant number of the best cells are used to form the next parent generation of the population.

The minimum number of iterations required by the GA for a good solution depends on the implementation of the recombination and mutation step, the initial population and how the problem is coded in the cells.

For our problem, each cell consists of a vector containing the segment boundaries in temporal order. This means that $\alpha_1 < \alpha_2 < ... < \alpha_{K+1}$ holds, for a vector $\alpha$.

The initialisation of the population is based on the duration hypotheses **D**. A cell containing the segment boundaries according to **D** is called a prototype. A new cell is created by adding small random values to all entries of the prototype. This is done in random order and must not destroy the temporal order of the vector represented in the new cell. The 100 cells created by this procedure form the first parent generation.

To create a child generation, we first apply the recombination step. Here two cells $\alpha,\beta$ from the parent generation are selected by the *roulette wheel algorithm* [5]. They form a new cell according to (4).

$$r(\alpha,\beta) := join(\alpha,\beta,lrand(2,K)) \quad (4)$$

$$join(\alpha,\beta,i) := \begin{cases} \alpha_1,...\alpha_{i-1},\beta_i,...,\beta_{K+1} & \text{if } \alpha_i < \beta_i \\ \beta_1,...\beta_{i-1},\alpha_i,...,\alpha_{K+1} & \text{otherwise} \end{cases}$$

*lrand* produces a random integer between 2 and *K*. Then we apply the mutation step (5) for each newly created cell. At this point, some boundaries are changed by adding a random value, so that the temporal order is preserved. If the population age *t* (i.e. the number of generations) increases, the random changes of the boundaries should be smaller, because the cells should be well adapted to the nature. Therefore big changes are no longer plausible. This is simulated by the linear weighting function w(*t*) in (5); w(*t*) decreases monotonically with the population *t*.

$$\mathrm{m}(\alpha,t) := \alpha_1,...,\alpha_{z-1}, \quad \alpha_z + \mathrm{lrand}(\alpha_{z-1} - \alpha_z + 1, \alpha_{z+1} - \alpha_z - 1) \cdot \mathrm{w}(t), \quad \alpha_{z+1},...,\alpha_{K+1} \quad (5)$$

$$z := \mathrm{lrand}(2,K)$$

## 4. EXPERIMENTS

Experiments were carried out using speech signals of 150 short German sentences read by one male and two female speakers. These sentences are taken from the *Bonner Prosodische Datenbank* (BPD) [6]. All speech signals in the BPD are manually labelled.

The transcriptions and the duration hypotheses for the automatic segmentation were generated using HADIFIX. Typical elision phenomena of spoken German are included in the transcriptions.

[2] Note that we include the parametric sampling rate into the constant *c*.

The deviation between manually labelled phoneme boundaries and automatically determined boundaries is used as a measure of quality.

The parametric representation of the speech signals is based on the mel-cepstrum. Therefore, 24 mel-scaled filter bank channels were used to compute 24 mel-cepstrum coefficients. A new set of mel-cepstrum coefficients was computed every 2.5 ms.

The size of the parent generation in the GA was 100 cells. Each iteration produced 100 new cells. The algorithm ran for a maximum of 20,000 generations. If the best cell in the population did not change for 1,000 iterations, the GA was stopped. After the termination of the iteration, the cell with the best fitness value was used as the solution to the segmentation problem.

The weighting function w($t$) in (5) was taken as a linear function with w(0) = 0.9 and w(20,000) = 0.5. The constant $a$ in (2) was set to 0.3. These values were taken from experimental results.

The SOMs (1) were created and trained by a procedure described in [4]. Two sets of SOMs were trained. One set was trained with speech signals from the BPD (of course different sentences were used for training and test phase) the other set by using speech signals form the Phondat-II corpus [7].

## 5. EXPERIMENTAL RESULTS

One of the essential aims of this work was to reduce the number of big deviations between manually and automatically labelled speech signals. In order to show that this aim was reached, we compared the described algorithm with the algorithm presented in [4], which is similar to a semi-continuous HMM. The results of [4] are shown in Table 1 in the column "DTW (Dynamic Time Warping)". The column "GA0" in Table 1 shows the results using the GA without duration constraints ($\lambda_1 = 1$, $\lambda_2 = 0$). We can see that the results are slightly worse than those in column "DTW". The reason for this deterioration is that a GA does not necessarily stop at the global minimum of the cost-function, but the DTW based algorithm does. Still, it is very interesting to see that almost all results of this GA are close to the optimal results of the DTW based algorithm.

Most publications on automatic segmentation of speech evaluate boundary placement separately for different classes of phonemes. In our opinion, such local error measures cannot be applied to global optimisation strategies such as the one used here. Since all phoneme boundaries in an utterance depend on each other, an error at the beginning of an utterance can still affect boundaries at the end and vice versa. This circumstance may influence any sound's deviation measure, which cannot be explained on the bases of our algorithm. Therefore, we only measure global performance.

If we use the duration constraints in addition to the similarity measure produced by the SOMs (Table 1, Columns "GA1" ($\lambda_1 = 1$, $\lambda_2 = 1$) and "GA2" (($\lambda_1 = 1$, $\lambda_2 = 2$)) the results will be better than those produced by the DTW based algorithm. Only the number of boundaries with a deviation smaller than 5 ms decreases. This is due to the precision of duration hypotheses which correlate to the real durations with c = 0.63. The hypothesised durations and the actual durations as taken from the manually labelled utterances deviate 26 ms on the average.

| SOMs created on | Time [ms] | DTW | GA 0 | GA 1 | GA 2 |
|---|---|---|---|---|---|
| | < 5 | **48** | 45 | 46 | 44 |
| | < 10 | **68** | 65 | **68** | 65 |
| | < 15 | 77 | 74 | **78** | 76 |
| | < 20 | 82 | 79 | **84** | 83 |
| **BPD** | < 25 | 85 | 82 | **88** | 87 |
| | < 30 | 87 | 84 | **90** | 89 |
| | < 40 | 90 | 88 | **94** | 93 |
| | < 50 | 92 | 90 | **95** | **95** |
| | < 60 | 94 | 92 | **96** | **96** |
| | < 5 | 40 | 39 | **41** | **41** |
| | < 10 | 60 | 58 | **63** | 62 |
| | < 15 | 69 | 67 | **73** | **73** |
| | < 20 | 75 | 74 | **80** | **80** |
| **Phondat-II** | < 25 | 79 | 77 | **85** | **85** |
| | < 30 | 82 | 81 | **88** | **88** |
| | < 40 | 85 | 84 | **92** | **92** |
| | < 50 | 88 | 87 | 93 | **94** |
| | < 60 | 91 | 90 | **95** | **95** |

**Table 1:** Differences between manual and automatic segmentation. Indicated is the amount of segment boundaries in percent whose deviation is smaller than the specific time (included are all three speakers with about 3000 boundaries).

Expecting the algorithm to obtain 100% deviations < 5 ms from boundaries placed by humans is unrealistic, because there are also significant differences in boundary placement between expert labellers [2]. Rather, the algorithm's results should be compared to human inter-labeller performance. Such an evaluation is very time consuming if the human labellers have to segment the speech signal from scratch. Therefore, we modified the setup for a pilot evaluation. Two phonetically trained labellers, GSO and CWI, were instructed to correct the boundaries very carefully on the basis of oscillographic, sonagraphic, and auditory information. As to be expected, the majority of the automatic boundaries were accepted as they had been set by the algorithm. 70% of all labels did not have to be corrected at all (Table 2, row < 5 ms).

## 6. CONCLUSIONS

Incorporating hypotheses about segment duration significantly increases the quality of automatically placed segment boundaries, especially when segmenting data from speakers which were not used for training the phoneme models (Table 1).

The price we have to pay for this is an increase in computing time and space. But this increase can be tolerated for the automatic segmentation of speech (ASS), because it is rarely executed in real time. In contrast to ASS techniques based on Dynamic Programming, the approach presented here can be

| Time [ms] | CWI | GSO |
|---|---|---|
| < 5 | 71 | 68 |
| < 10 | 80 | 80 |
| < 15 | 86 | 86 |
| < 20 | 89 | 90 |
| < 25 | 91 | 92 |
| < 30 | 93 | 93 |
| < 40 | 95 | 95 |
| < 50 | 96 | 96 |
| < 60 | 97 | 96 |

**Table 2:** Differences between automatic and manually corrected segmentation. Indicated is the amount of segment boundaries in percent whose deviation is smaller than the specific time. The correction was based on the automatically placed boundaries with $\lambda_1 = 1$, $\lambda_2 = 1$.

parallelised effectively. Therefore it can profit from commonly used multiprocessing architectures.

Comparing the deviations between human labellers and the algorithm to human inter-labeller deviations [2], we find that the algorithm is already quite reliable. But while the upper boundary of human inter-labeller deviations rarely exceeds 40 ms, there are a few cases where our algorithm produces larger differences. Future work will therefore aim at reducing large deviations (20 ms and more) even further, since such deviations are extremely rare between human labellers.

Our algorithm is very flexible. Since the optimisation step has information about possible segment boundaries, we can easily use many different criteria for boundary placement. In future work, the relation between inter- and intra-class distances will be integrated into the cost function. We are also considering to use further prosodic features such as accentuation.

The probability of a wrong boundary placement increases with the length of the utterance. The space requirements of our algorithm also increase quadratically with the number of parameter vectors generated from the signal. Therefore, it is necessary to subdivide the signal into smaller units, such as voiced and unvoiced segments. Experiments with such units are currently being conducted. If successful, the present algorithm will a universally applicable, high-performance method for ASS.

Every method for ASS is adapted to certain segmentation criteria based on the training material. This makes a comparison of different methods difficult, if not impossible. It would be highly desirable to establish a set of benchmark data for the ASS task and to devise a reasonable measure of performance.

## 7. ACKNOWLEDGEMENTS

We would like to thank our phonetic experts Gerit Sonntag and Christina Widera, who spent a lot of time correcting the automatically placed labels.

## 8. REFERENCES

[1] Portele, T., Heuft, B., „Towards a prominence-based speech synthesis system“, Speech Communication 21, pp. 61-72, 1997

[2] Wesenick, M.B., Kipp, A., „Estimating the Quality of Phonetic Transcriptions and Segmentations of Speech Signals“, Proceedings ICSLP, Philadelphia, pp. 129-132, 1996

[3] Kohonen, T., “Self-Organizing Maps“, Springer, Berlin, 1995

[4] Stöber, K., „Einsatz von Self-Organizing Maps für die sprecherunabhängige automatische Lautsegmentierung”, Fortschritte der Akustik DAGA‘98, Zürich, 1998

[5] Schöneburg, E., Heinzmann, F., Feddersen, S., „Genetische Algorithmen und Evolutionsstrategien“, Addison-Wesley, Bonn, 1994

[6] Heuft, B., Portele, T., Höfer, F., Krämer, J., Meyer, H., Rauth, M., Sonntag, G., „Parametric Description of F0-Contours in a Prosodic Database“, Proceedings ICPhS Vol. 2, pp. 378-381, 1995

[7] Pompino-Marschall, B. (ed.), „PHONDAT. Verbundvorhaben zum Aufbau einer Sprachsignaldatenbank für gesprochenes Deutsch“, Forschungsbericht des IPSK München (FIPKM) 30, pp. 99-128, 1992