TR-IT-0245

# Vocal Tract Length Normalization (VTLN) using Warp Scales for Spontaneous Japanese Speech

ペルティエ ニコラ
Nicolas Pelletier

シンガー ハラルド
Harald Singer

1997.12.19

In this report, we will describe how we used warp scales in order to reduce the speaker variability of speech. Such experiments have already been conducted on English. We will show that these methods also work for ATR's Japanese Speech Database (SDB).
(CAUTION: after the end of Nicolas' stay we found some serious bugs in the software. These bugs invalidate any conclusions. We decided to publish this report nevertheless to document the approach and the software.)

目次

# 1 Introduction

The vocal tract length variation among speakers is a major problem encountered in speech recognition. First, the vocal tract length varies between genders: it can be noticed that the average vocal tract length measured among males is greater than the average vocal tract length measured among females; then there is another variation of the vocal tract length among speakers of the same gender. Thus the speech appears to be very dependent of the speaker.

However, we want to get as free as possible from this speaker variability of speech. For this purpose, we have first to define a "normalized" speaker and then a kind of "distance" between a given speaker and the normalized speaker. Thus we will be able, by preprocessing the input speech from a given speaker, to change it into the corresponding speech that would have been uttered by the normalized speaker. After that step, every utterance will be known as one of the normalized speaker's, so that the preprocessing operation will free us from the specificities of the input speaker.

To achieve our goal, we chose to warp the spectrum of the input speech so as to move the formants toward normalized values. It has been reported that this use of warping functions increases the perfomance of speaker independent speech recognizers typically by 2% to 3% absolute (see [3]).

To train our speech model, we had first to decide the training set. We chose to train one gender dependent model as well as a gender independent one. By doing so, we can use the gender independent model to decide wether the input speech comes from a male or a female voice, and then refine our normalization by using the appropriate gender dependent speech model according to the results of this test. Each of our speech models is a single probability distribution, consisting of a mixture of 256 multivariate Gaussians. The training was conducted by applying an iterative algorithm based on acoustic segmentation and clustering.

In Section 2 we will outline the frequency warping and the iterative training algorithm. In Section 3 we will give results for speaker clustering and finally speech recognition.

# 2 Algorithms

In this section, we describe the iterative algorithm we used to train our speech models, which is almost the same as the one used by Dragon (see [3]).

The initialization step consists in training a first polynomial segment model $\Lambda_0$ using the unwarped data. The segmentation is described below. We then use the EM and LBG algorithms to construct the mixture model from the selected speech frames.

The iteration step consists of using the previous generic voiced speech model to estimate the best warp scale for each of the training speakers. In the next iteration, each speaker will be assigned the speech data corresponding to that estimation. This set of warped data is then used to train a new model. The best warp scale for a speaker's training data is determined by signal processing the speech at each of the warp scales (this only needs to be done once during the training process), and then scoring the voiced frames with the generic voiced speech model: the warp scale that scores best is selected. The process is iterated until the total score for all speakers against the generic speech model ceases to increase significantly. This usually occurs by the fourth iteration.

The tools we used to perform the training operation were `HCode` for feature extraction and the acoustic segmentation, clustering and scoring tools `asegm`, `km-las`, and `segclass` provided with the ASSM toolkit described in [1].

## 2.1 Frequency Warping

The frequency warping is done using a piecewise linear tranformation of the normalized frequency axis. This transformation has fixed points at the normalized frequencies 0 and 1 (in our case, the Nyquist frequency is 8 kHz). We choose a point $\Phi$ below the Nyquist frequency ($\Phi$ is chosen to be 0.8 in our case, according to the results of other experiments done at ATR ITL, whereas Dragon chose $\Phi$ to be 0.875). The map from 0 kHz to $\Phi$ is a line from the fixed point at the origin with a slope ranging from 0.8 to 1.2 for our gender dependent models, and from 0.8 to 1.24 for our gender independent model. Let us call A the point of the

warping scale whose first coordinate is $\Phi$. The map from $\Phi$ to the Nyquist frequency is a line that intersects the previous one at A and ends at the fixed point at the Nyquist frequency (see Fig. 1).

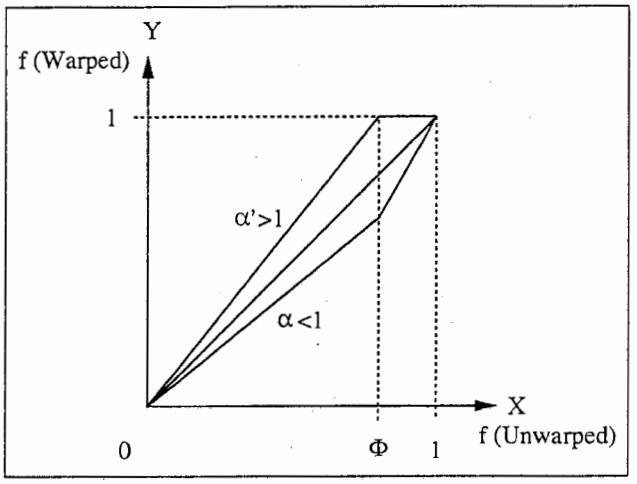


図 1: Warp scale for speaker normalization

Let us also call warping factor the inverse of the slope of the first linear part of one such map. For our experiments, the step between one warping factor and the next is chosen to be 0.02. Other experiments run at ATR ITL show that this value allows to separate speakers acurately without requiring too much disk space and computation time.

The warping operation is done just after the FFT computation. Let $X$ and $Y$ denote the original and transformed frequency axes, and let $f: X \rightarrow Y$ be a warp scale. Given $y$ in $Y$, there is a unique $x$ in $X$ with $y = f(x)$, since our warp scales are bijective functions for warping factors ranging from $\Phi$ to $1/\Phi$ (that is, from 0.8 to 1.25 in our case). Since, in general, $x$ will not be one of the frequencies where we computed the FFT, we estimate the value of the FFT at $x$ by linearly interpolating the values of the FFT at the two frequencies nearest to $x$ where the FFT was computed. We then set $FFT(y) = FFT(x)$ and use these new values when we extract the features.

Fig. 2 shows the effects of warping on the power spectrum. On this figure, we chose a 10 ms long frame of speech in the middle of the utterance of the vowel /a/ by a male speaker, and warped the spectrum with two different warp scales, corresponding to warping factor values 0.9 and 1.1. The upper part of the figure shows the unwarped power spectrum and the power spectrum warped with a warping factor of 0.9; the lower part is a superposition of the unwarped power spectrum and the power spectrum warped with a warping factor of 1.1 Notice the shift of the formants of the FFT.

Because warping factors less than 1 stretch the frequency axis, we expect that they will be more appropriate for male voices' preprocessing, whose formants should be lower than those of our average speaker (represented by a warping factor of 1). On the other side, warping factors greater than 1 compress the frequency axis and should thus be more appropriate for female voices' preprocessing.

## 2.2 Iterative Segmentation and Clustering

This section describes the acoustic segmentation we used to train our speech models. The acoustic segmentation operations have only to be performed once during the training operation. The acoustic segmentation tool we used performs a dynamic programming (DP) search in order to find the segmentation giving the minimum distorsion under a set of constraints. Since we want to get HMM type speech models, we let the number of regions in a segment to its default value, 1. The frame offset was set to 0 and the variance floor to 0.1 We used two types of constraints for the acoustic segmentation.

The first step was to perform an acoustic segmentation on the unwarped data. The segments boundaries were set so that the total distorsion within a given segment should not exceed 3. Such a segmentation could

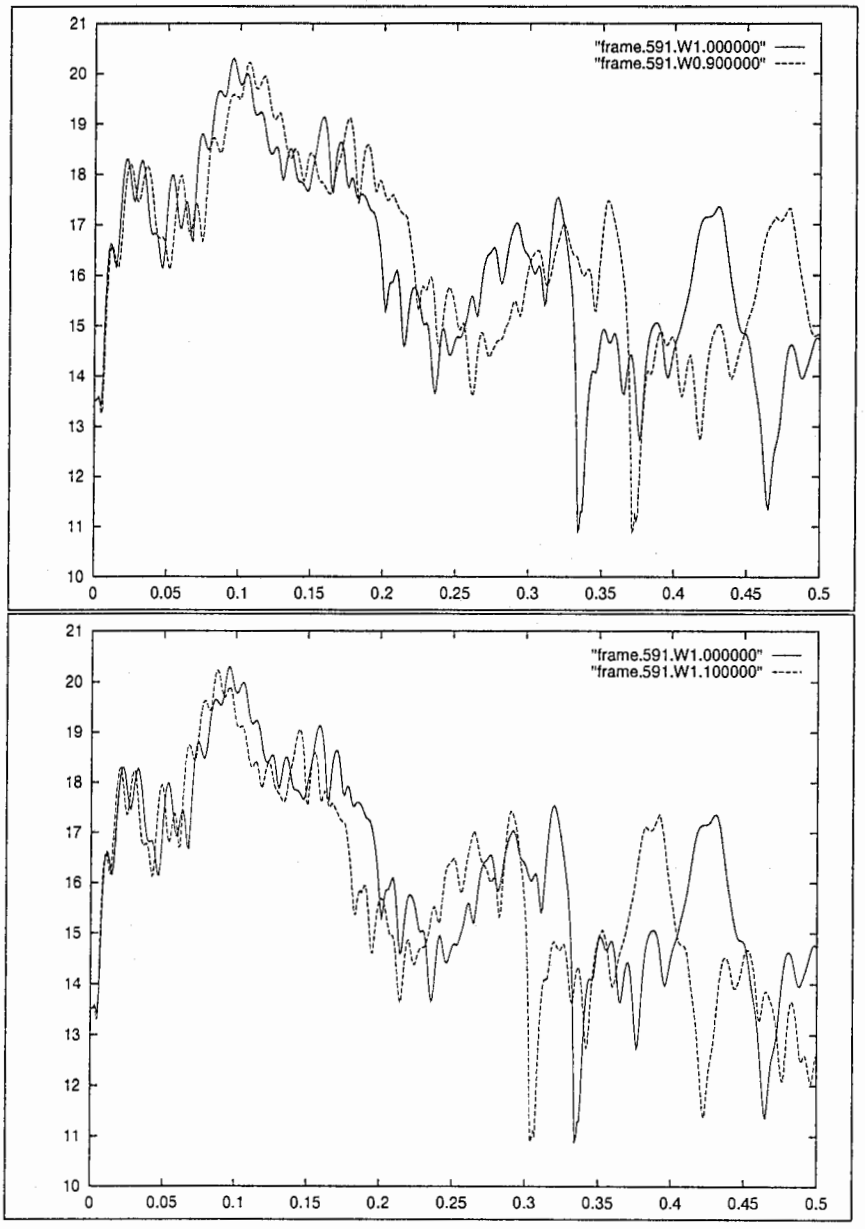


図 2: Effects of warping on the power spectrum

not be found in all the cases, because of the small size of some speech files. However, the distorsion was always less than 4.94821, and this maximal value was obatined for 1 segment only. Some other segments had distorsion values greater than 3, but setting the minimal distorsion constraint to 3 gave us a good ratio of actual minimal distorsion over computing time. The model for distorsion computations is a polynomial segment model (PSM) whose parameters are as follows: polynomial order 0, full covariance and Mahalanobis distance. The frame offset was set to 0 and the variance floor to 0.1 We got then the following configuration file for the server:

```
ADDFVPATH t <vectors_path>
ADDFVEXT  t  <vectors_ext>
ADDLBEXT  t  <label_ext>
SETOPT POLYORDER   i 0
SETOPT COVARTYPE   i 1
SETOPT DISTTYPE    i 1
SETOPT FRAMEOFFSET l 0
SETOPT VARFLOOR    f 0.1
SETOPT OUTPUTLBLDIR <label_path>
SETOPT OUTPUTLBLEXT <label_ext>
SETOPT OUTPUTBLMLF  <master_label_path>
```

where the words enclosed in angle brackets stand for strings designing the respective paths or extensions needed for input or output. Our command line was then:

```
asegm -d 1 3 -p -S <list_file>
```

where `<list_file>` stands for a file containing the names of the feature vectors files for the unwarped data.

The second step is to get a segmentation for the warped data. Since we will have to score all these segments against our model $\Lambda_0$ which is defined using the unwarped data, the output of the segmentation for the warped data must give the same number of segments as for the unwarped data. The configuration file for this second segmentation operation is the same as for the first. The command line becomes:

```
asegm -d 1 3 -p -S <list_file> -W -X <label_path> -Y 3.<label_ext> -Z <master_label>
```

where the constraints are given via the paths extensions of the label files and the corresponding master label file for the unwarped data.

At each step of the iterative algorithm, we define a new model by clustering the segments. The clustering operation performs a series of divisive clustering and k-means clustering iterations used to get approximately 256 multi-variate Gaussians. The clustering operation ends by running 5 k-means iterations to avoid having clusters with too few segments. This explains why we always get fewer than 256 Gaussians (The number of Gaussians we got while training our gender independent and gender dependent model ranged from 239 to 253). During the clustering operation, we used a diagonal covariance and a likelihood distance; the variance floor was set to 0.001. The server configuration file for the clustering operation looked like this:

```
ADDFVPATH t <vectors_path>
ADDFVEXT  t <vectors_ext>
ADDLBEXT  t <labels_ext>
LBMLFADD  t <master_label>
SETOPT OUTPUTMDLDIR  t <model_path>
SETOPT OUTPUTTREEDIR t <tree_path>
SETOPT POLYORDER i 0
SETOPT COVARTYPE i 0
SETOPT DISTTYPE  i 2
SETOPT VARFLOOR  f 0.001
```

To get our Gaussian mixture pdf, we performed a series of divisive clustering with increasing final number of clusters, and inserted some k-means algorithm iterations between them to avoid an excessive reduction of the number of clusters at the end, and thus to get close to 256 Gaussians in our model. The steps we used for clustering were as follows:

1. First, perform divisive clustering up to 10 clusters, and then run 10 iterations of k-means algorithm.
2. Perform divisive clustering up to 30 clusters, followed by 10 iterations of k-means algorithm.

3. Perform divisive clustering up to 100 clusters, followed by 20 iterations of k-means algorithm.

4. Perform divisive clustering up to 200 clusters, followed by 10 iterations of k-means algorithm.

5. Finally, perform divisive clustering up to 256 clusters, followed by 5 iterations of k-means algorithm.

Our command line was then:

```
km-las -v -c <list_of_vectors> -k ''dN10 kN10 dN30 kN10 dN100 kN20 dN200 kN10 dN256 kN5''
```

where `<list_of_vectors>` stands for a file containing the names of the feature vectors files.

The last step is to score all the segments computed using all the warping factors against the model to get the log likelihood of each of the segments and each warping factor. The scoring operation is done in our experiments by using the **segclass** tool, for which the configuration file we used was as follows:

```
ADDFVPATH  t <vectors_path>
ADDFVEXT   t <vectors_ext>
ADDMDLPATH t <model_path>
SETOPT POLYORDER i 0
SETOPT COVARTYPE i 0
SETOPT DISTTYPE  i 2
SETOPT VARFLOOR  f 0.001
SETOPT OUTPUTLBDIR t <likelihood_labels_ ath>
SETOPT OUTPUTLBEXT t <likelihood_labels_ext>
SETOPT OUTPUTLBMLF t <likelihood_master_label_path>
```

The polynomial order, covariance type, distance type and variance floor kept the values they had during the clustering operation. The three last options set the paths and extensions for the label files which will contain the log likelihoods of the segments.

The corresponding command line was:

```
segclass -S <vectors_list_file> -X <labels_path> -Y <label_ext> -Z <model_list>
```

where `<vectors_list_file>` stands for the whole name of a file in which are stored the names of the feature vectors files, and `<model_list>` for the whole name of a file in which is stored the whole name of the model we want to use.

For each warping factor, by adding the log likelihoods of all the segments computed from the utterances of a given speaker using that particular warping factor, we got the log likelihood of this speaker, given the warping factor and the current speech model. Let us denote this log likelihood by $\log\big(\Pr(s|w,\Lambda)\big)$. Then, we define a new set of input segments in which each speaker is represented by the segments that were computed from his utterances using the warping factor that gave the highest $\log\big(\Pr(S|w,\Lambda)\big)$. Using this new set of data, we can go back to the clustering step, which will define the new speech model. Using this algorithm, the total log likelihood, computed for all the speakers $S$ using the current best warping factor $w_S$ for each of them, increases with the iteration number as long as we don't get stuck in some local extremum. We stop the training as soon as this total log likelihood $\sum_S \log\big(\Pr(S|w_S,\Lambda)\big)$ ceases to increase significantly.

## 3 Experiments and Results

This section describes the protocols we used to train our gender dependent and gender independent speech models, as well as the speakers distributions we got. Finally, we will give mention preliminary recognition experiments.

### 3.1 Database Description

The data for our experiments was taken from the ATR Speech Database (SDB/SLDB) described in [2].

Our training sets were built using 100 male and 130 female speakers

```
/DB/SDB/ALL/INFO/etc/trainingset/T_M_0100.ascii
/DB/SDB/ALL/INFO/etc/trainingset/T_F_0130.ascii
```

These files describe Japanese conversations, downsampled to 16 kHz and 16 bit per sample. Since we used only voiced speech, our first task was to remove the non-speech parts from the speech files, according to their respective transcription files.

The list of actual speech times and total speech times for every speaker in the two `ascii` files follows in the tables Tab. 5 and Tab 6

## 3.2 Feature Extraction

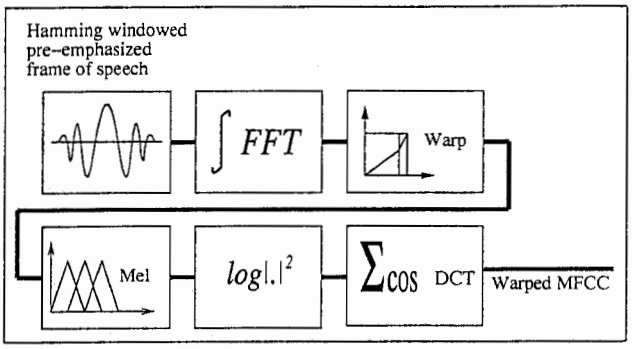


図 3: Feature extraction process

To extract the features from our speech data, we proceeded as shown in Fig. 3 During this process, the window duration was set to 25.6 ms and the frame duration to 10 ms. Before any other operation, we applied a pre-emphasis with a coefficient of 0.97 to the speech frame as well as a Hamming window. After computing the FFT for a given speech file, we warped the spectrum with the current warpscale, and then let it go through the mel filterbank before taking the log and then applying a DCT. The feature extraction operation was performed using a slightly modified version of HCode 1.5, so as to allow the frequency warping. For each wave file, we computed 12 mel-frequency cepstral coefficients. The filter we used are triangular, normalized and equally spaced along the mel-scale as shown in Fig. 4), and described in [4].

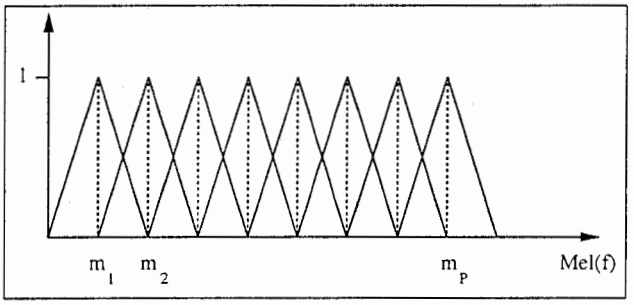


図 4: Mel filterbank for feature extraction

The melscale used is $Mel(f) = 2595 \cdot log_{10}(1 + f/700)$, and the filtering operation is performed in the magnitude domain. We get our MFCC coefficients $c_i$ by applying the following discrete cosine transform to the log filter bank outputs $m_j$.

$$c_j = \sum_{j=1}^{P} m_j \cdot cos\left(\frac{\pi \cdot i}{P} \cdot (j - 0.5)\right), 1 \leq i \leq N$$

where $N$ is the required number of output parameters (12 in our case) and $P$ the analysis order (24 in our case, that is $2 \times N$, as recommended in [4]). The mean was also removed from each coefficient.

Using HCode under csh, we had to type in the following commands:

```
setenv SAMPPERIOD 625
setenv SAMPSIZE 2
setenv SAMPKIND WAVEFORM
setenv HDSIZE 0
HCode -m -n 12 -p 24 -f 10.0 -w 25.6 -h -k 0.97 -s 1.0 -j -W <warp> \
-F NOHEAD <speechfile> <featuresfile>
unsetenv HDSIZE
unsetenv SAMPKIND
unsetenv SAMPSIZE
unsetenv SAMPPERIOD
```

where `<warp>` stands for any acceptable warping factor (from 0.8 to 1.25 in our case), `<speechfile>` is the name of the input speech file and `<featuresfile>` the name of the output feature vectors file [4].

## 3.3 Speaker Clustering Experiments

### (3.3.1) Gender Dependent Experiments

Since our goal is to build a speaker independent model, we have to be very careful about each parameter which could introduce a bias in our statistics. On one hand, great variations of speech time among the speakers of our gender dependent training sets could lead to the following bias: since speakers with greater actual speech times would be statistically more important than other speakers with less speech data, our final model would recognize these speakers best, and thus become somewhat speaker independent. On the other hand, selecting speakers with sufficient speech time for our training set could lead to a small subset; our model would then be speaker dependent, too, because it would have been trained with too few different speakers.

Upon examination of the actual speech times for each speaker of our description files `T_M_0100.ascii` and `T_F_0130.ascii`, the selection for our gender dependent training sets went in two steps. First, we chose every speaker whose actual speech time was greater than 30 s, avoiding thus to have a too great number of speakers with too little speech per speaker. Then, for each of these speakers, we selected speech files so that the total speech time of the speaker be close enough to 20 s, to avoid giving a too great statistical importance to that speaker. To avoid running into problems during the execution of our algorithm, we eliminated any speech file whose length was less than 1 window while removing the silences from our data. Since our window duration is 25.6 ms, we chose to get rid of any file containing less than 30 ms of speech. This gave us two subsets of speakers, one with 63 male speakers and the other with 74 female speakers. The names of the speakers used can be found in Tab. 4 The overall likelihood of our gender dependent speech models stopped to increase by the fourth iteration. The distributions of speakers by warping factors are shown in Fig. 5 The upper part shows the distribution for males speakers and the lower the distribution for females speakers.

For the male speakers, the average warping factor is 1.01, and the covariance is 0.03. The third moment of the distribution is $2.14 \cdot 10^{-5}$. For the female speakers, the average warping factor is 1.00, and the covariance is 0.04. The third moment for this distribution is $5.76 \cdot 10^{-5}$.

### (3.3.2) Gender Independent Experiments

To build a gender independent and speaker independent model, the first constraint for our training set is to have as many male speakers as female speakers. Since we use approximately twice as much data as for the gender dependent models, we can refine our choice of speakers in the following way: we first chose every speaker speaking actually more than 30 s. For each of these speakers, we selected a set of speech files so that the total speech time be as close as possible to 30 s. Other experiments considered the limit of 20 s to be the minimum required to determine a warping factor as long as this speech time is made of actual speech only. The names and corresponding speech times for the 100 speakers (50 speakers per gender) used for the training of this model can be found in Tab. 3. The total log likelihood of the speakers, given the best

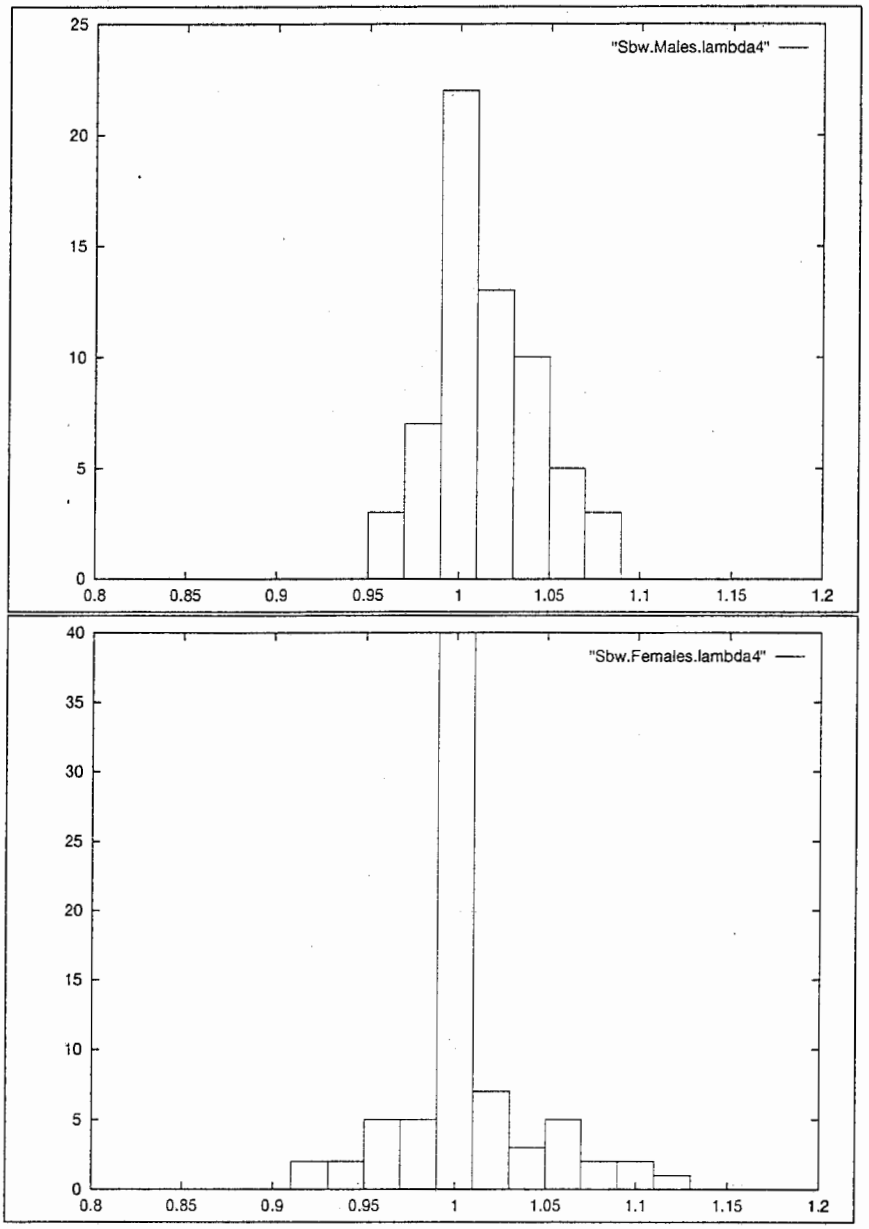


図 5: Distribution of speakers by warping factor in the gender dependent models

warping factors for each of them and the current model stopped to increase at iteration 6. At that time, the distribution of the speakers across the warping factors was as shown in Fig. 6

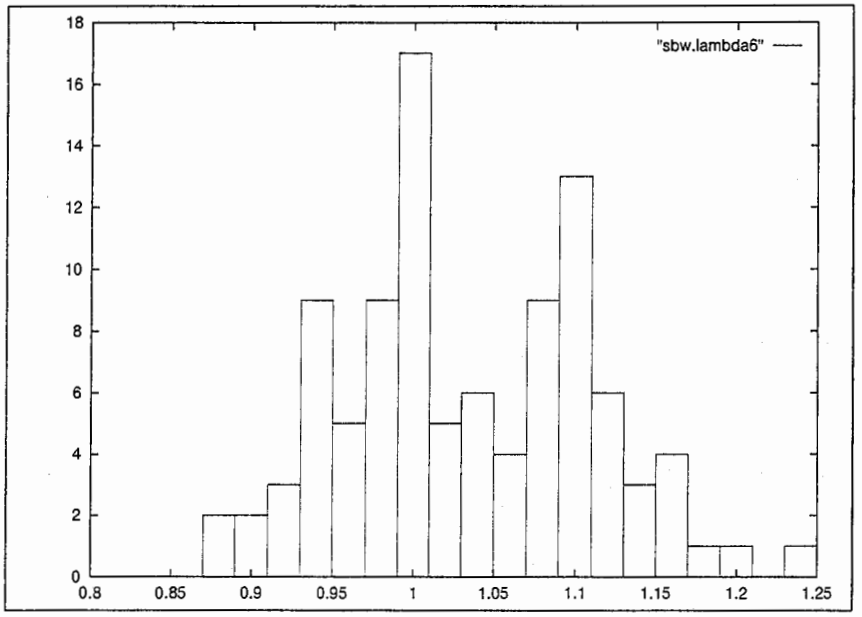


図 6: Distribution of speakers by warping factor in the gender independent model

The average best warping factor for our gender independent model is 1.03. The most likely warping factor for each of the speakers for this gender independent model $\Lambda_6$ can be found in Tab. 1 Figure 7 shows the variation of the total log likelihood of the model over the iterations.

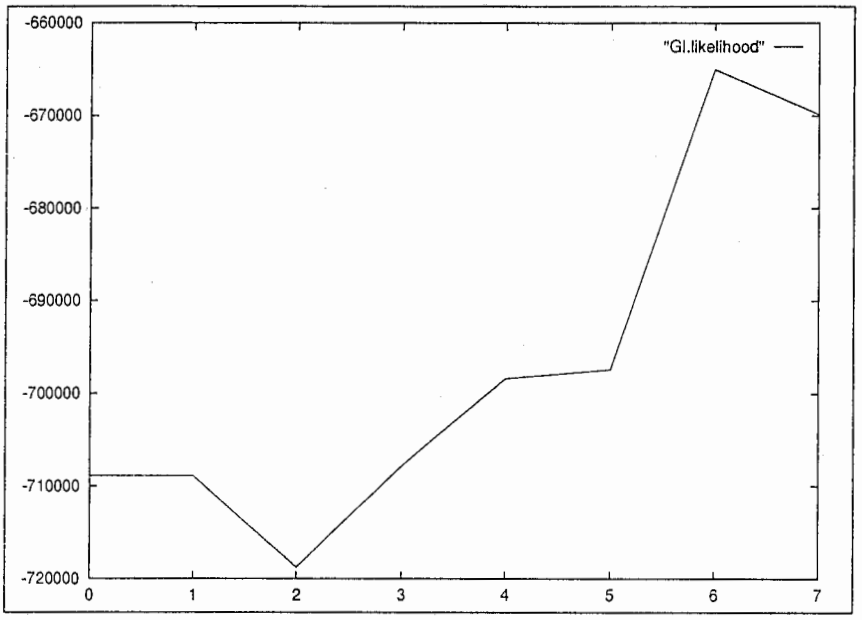


図 7: Total log likelihood variation over the training iterations

However, we noticed that the average warping factor slid towar higher values over the iterations. This is surprising, since our normalized speaker is represented by warping factor 1.0 in our models. The sliding effect can be observed in Fig. 8.

The gender dependent models gave us a Gaussian distribution of the speakers across the warping factors.

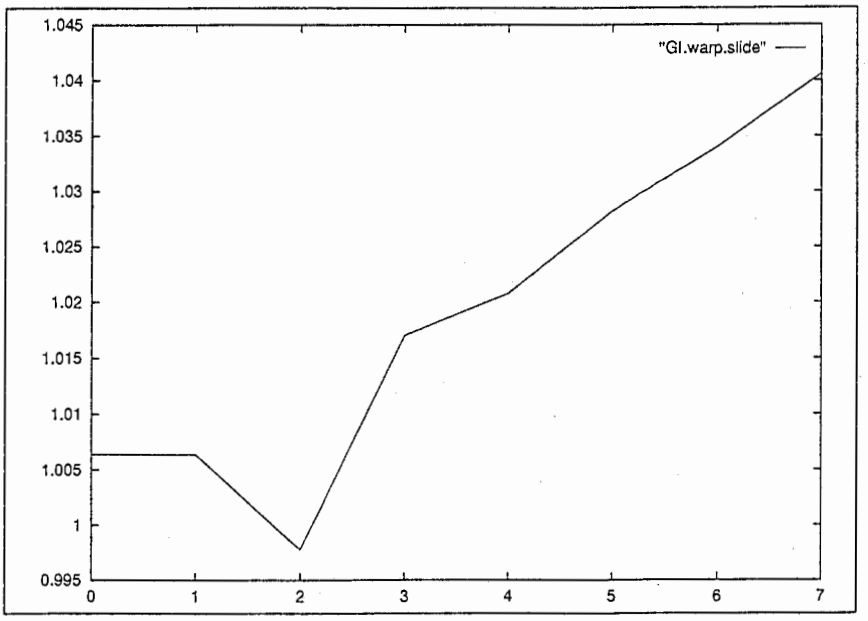


図 8: Sliding effect of average warping factor over iterations

The mean of these Gaussians stands for the average warping factor for either male or female voices. The mean for the male speakers distribution is a little too high (1.01 instead of 1.00); perhaps using more speech data could resolve this problem. The values of the covariance show that the range of warping factors used could be restrained to [0.90;1.1] Looking at the values of the third moments, we can see that although the number of iterations is still low, the shapes of the distributions are already close to Gaussians. This shows that the algorithm we used for training is well able to discriminate between warping factors.

For our gender independent model, the distribution of speakers over the warping factors is the sum of two Gaussians. This shows a good discrimination between male voices and female voices. However, the average warping factor for this model is a little too high: 1.03 instead of the expected value of 1.00. Michiel Bacchiani already noticed a move of the average warping factor toward high values while training gender independent models with English databases.

## 3.4 Recognition Experiments

### (3.4.1) Experiments Description

To see if our gender dependent models increased significantly the speech recognition performance, we compared the recognition results for the male speakers (resp. female speakers) from the test sets:

```
/DB/SDB/ALL/INFO/etc/testset02/S1.ascii
/DB/SDB/ALL/INFO/etc/testset02/S2.ascii
/DB/SDB/ALL/INFO/etc/testset02/S4.ascii
```

using two HMnet topologies for each gender, one trained with the most likely warping factor for each speaker from the corresponding training set, and one trained without any warping factor information (i.e. for each speaker. we assumed that the best warping factor was 1.0). The software used for topology training and speech recognition was ATRSPREC r05r01. The training sets used to train the topologies were, for male and female speakers respectively,

```
/DB/SDB/ALL/INFO/etc/trainingset03/T_M_0100.ascii
/DB/SDB/ALL/INFO/etc/trainingset03/T_F_0130.ascii
```

To have a meaningful comparison, we tried to remain as close as possible to the parameter set we used to train our models with the ASSM software. The sampling rate (without downsampling) was set to 16 kHz, the frame shift to 10 ms and the frame length to 25.6 ms. The FFT order was set to 512. We ran a mel frequency based analysis. We used 12 cepstral coefficients, calculated with 24 filterbanks. The features were 26 coefficients as follows: 1 power coefficient, 12 cepstral coefficients, 1 delta power coefficient and 12 delta cepstral coefficients. We also applied a Hamming window to correct the frames, as well as a pre-emphasis coefficient of 0.97. The lexicon used for phoneme recognition and corresponding ngram were

```
/dept1/work1/V1/model/LEX.P
/dept1/work1/V1/model/LM.P
beamwidth: 30, languagemodelscale: 4,8
```

For word recognition, the lexicon and ngram used were respectively

```
/dept1/work1/V2/model/LEX.W
/dept1/work1/V2/model/LM.W
beamwidth: 80, languagemodelscale: 8,20
```

### (3.4.2) Results

Using the male voices model, the phoneme recognition results did not significantly improve. Moreover, comparing the recognition results with and without using warping factors, we noticed an increase of the difference of performance between the speakers while using the warping factors.

The most likely warping factors according to our gender independent set for all speakers from the trainingsets `T_M_0100.ascii` and `T_F_0130.ascii` can be found in table 2. These warping factors were computed by scoring all the speech frames against the gender independent model. Therefore, for speakers that were also in the training set for the gender independent model, the most likely warping factor may be different. We used these warping factors for the training of the topology. However, in this test experiment, no warping was applied to the speech frames of the test speakers.

Phoneme recognition results and word recognition results do not improve compared to precedent results. On the contrary, a significant decrease of the word accuracy by 4% could be noticed.

In our models, the warping factor grows linearly, giving us in the end the Gaussian distributions of speakers across the warping factors. However, to refine the decision of the most likely warping factor, it could be interesting to run other experiments with different variations of warping factors. For example, using the knowledge given by the distributions, one could decide to distribute the warping factors at which the computations will be done over the possible values according to the number of speakers by warping factor.

## 4 Discussion and Conclusion

In this report, we experimented with the use of warp scales and the associated methods for training and showed that this also work for the Japanese database of ATR (SDB): using these methods along with a gender independent model, recognition results improved for both genders. However, determining a reliable warping factor remains a problem. We observed a sliding effect of the average warping factor towards higher values while training. The investigation of various shapes of warp scales, as well as the use of more data for each speaker could lead to improved results.

## Acknowledgements

We wish to express our thanks to ATR for providing us with the articles, computers, software, database necessary for our research. We also thank Michiel Bacchiani for kindly allowing us to use his software (ASSM) and to Mari Ostendorf who suggested this research.

## 参考文献

[1] M. Bacchiani. The ASSM toolkit for polynomial segment models and automatic unit design. Technical Report TR-IT-0225, ATR, June 1997.

[2] H. Singer, M. Tonomura, Q. Huo, J. Ishii, T. Fukada, and M. Schuster. Baseline acoustic models for the spoken language database (sdb/sldb). Technical Report TR-IT-0206, ATR, March 1997.

[3] S. Wegmann, D. McAllaster, J. Orloff, and B. Peskin. Speaker normalization on conversational telephone speech. In *Proc. ICASSP*, pages 339–341, 1996.

[4] S. Young, J. Jansen, J. Odell, D. Ollason, and P. Woodland. The HTK book. Cambridge University Engineering Department Speech Group and Entropic Research Laboratorics Inc, 1996.

# 付録 A Detailed Warping Factors

表 1: Most likely warping factors for the speakers from the gender independent set

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| FAKKA | 1.08 | FAKMA | 1.08 | FAKMI | 1.08 | FAKMU | 1.04 |
| FAKTS | 1.12 | FAKYZ | 1.10 | FASYA | 1.18 | FAYKZ | 1.00 |
| FCHAK | 1.10 | FCHHA | 1.04 | FCHKA | 1.14 | FCHMA | 1.02 |
| FCHNO | 1.16 | FEMNA | 1.06 | FEMNI | 1.12 | FHIMU | 1.10 |
| FHIOY | 1.08 | FKEAO | 1.20 | FKESH | 1.08 | FKIFU | 1.12 |
| FKYIW | 1.12 | FMAKS | 1.10 | FMAKY | 1.08 | FMASH | 1.08 |
| FMASW | 1.10 | FMATP | 1.14 | FMEWA | 1.08 | FMIIS | 1.14 |
| FMINP | 1.06 | FMIOW | 1.10 | FNAMU | 1.10 | FNOKU | 1.08 |
| FNOSA | 1.10 | FNOSZ | 1.10 | FRIMI | 1.06 | FRUKO | 1.04 |
| FSAFU | 1.16 | FSAOY | 1.10 | FSUSA | 1.12 | FSUTS | 1.00 |
| FTATA | 1.04 | FTAYO | 1.16 | FTESU | 1.24 | FTOKO | 1.12 |
| FTOTZ | 1.10 | FYUFY | 1.16 | FYUKZ | 1.04 | FYUNV | 1.10 |
| FYUTZ | 1.10 | FYUYA | 1.02 | | | | |
| MAKKI | 1.00 | MAKYO | 0.90 | MFUAK | 0.96 | MHAMO | 1.04 |
| MHIAT | 1.02 | MHIMO | 0.94 | MHIOY | 0.98 | MHISX | 1.00 |
| MJUIS | 0.90 | MKAIS | 1.00 | MKASE | 1.00 | MKAYY | 1.00 |
| MKETA | 0.94 | MKOTA | 0.94 | MMAMX | 0.94 | MMANA | 0.98 |
| MMATZ | 0.96 | MMAUZ | 1.02 | MMIKA | 1.00 | MRYAR | 0.94 |
| MRYHA | 1.00 | MRYMO | 0.92 | MRYSA | 0.98 | MSANU | 1.00 |
| MSATA | 1.00 | MSHMA | 0.88 | MSHOO | 0.88 | MSHSO | 0.94 |
| MTAAI | 0.98 | MTADA | 1.00 | MTAHI | 1.00 | MTAIZ | 1.00 |
| MTAMO | 0.98 | MTANA | 1.02 | MTAOO | 0.98 | MTATA | 0.94 |
| MTATZ | 0.98 | MTAWA | 0.92 | MTAYA | 0.96 | MTEIS | 1.00 |
| MTEMI | 0.98 | MTOAZ | 0.94 | MTOHO | 0.96 | MTOMA | 0.92 |
| MTONI | 1.00 | MYAME | 0.96 | MYANA | 1.00 | MYOKU | 0.98 |
| MYUMI | 0.94 | MYUYA | 1.06 | | | | |

表 2: Most likely warping factors for all male and female speakers

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| FAKKA | 1.08 | FAKKI | 1.00 | FAKMA | 1.10 | FAKMI | 1.10 |
| FAKMU | 1.04 | FAKSU | 1.12 | FAKTS | 1.12 | FAKYA | 1.10 |
| FAKYY | 1.16 | FAKYZ | 1.10 | FAMIS | 1.24 | FASYA | 1.18 |
| FAYHZ | 1.14 | FAYIS | 1.08 | FAYKZ | 1.02 | FCHAK | 1.10 |
| FCHHA | 1.04 | FCHKA | 1.14 | FCHKI | 1.12 | FCHKY | 0.98 |
| FCHMA | 1.02 | FCHNO | 1.16 | FCHSA | 1.12 | FCHYO | 1.18 |
| FCHYZ | 1.06 | FEIAO | 1.12 | FEIFU | 1.18 | FEIOO | 1.16 |
| FEMNA | 1.06 | FEMNI | 1.12 | FHATA | 1.08 | FHIMU | 1.10 |
| FHIOY | 1.10 | FJUSH | 1.10 | FJUSZ | 1.14 | FKAKA | 1.08 |
| FKASA | 1.08 | FKEAO | 1.20 | FKEHO | 1.10 | FKEIM | 1.04 |
| FKEKA | 1.16 | FKEKO | 1.12 | FKESH | 1.10 | FKESZ | 1.20 |
| FKIFU | 1.12 | FKISU | 1.10 | FKOMI | 1.24 | FKOTE | 1.10 |
| FKYIW | 1.14 | FMAIM | 1.18 | FMAKS | 1.12 | FMAKY | 1.08 |
| FMAMO | 1.24 | FMASH | 1.08 | FMASW | 1.12 | FMATP | 1.14 |
| FMAYO | 1.14 | FMEWA | 1.08 | FMIHX | 1.18 | FMIIS | 1.12 |
| FMINP | 1.06 | FMION | 1.12 | FMIOO | 1.08 | FMIOW | 1.10 |
| FMISH | 1.10 | FMISO | 1.16 | FMITA | 1.22 | FMITO | 1.14 |
| FMITX | 1.04 | FMITZ | 1.14 | FMIYA | 1.24 | FMOMA | 1.08 |
| FNAIR | 1.12 | FNAMU | 1.10 | FNOHI | 1.02 | FNOHX | 1.06 |
| FNOKU | 1.04 | FNOKY | 1.08 | FNOMU | 1.18 | FNOSA | 1.10 |
| FNOSZ | 1.10 | FREII | 1.08 | FREOO | 1.20 | FRESU | 1.14 |
| FRIMI | 1.08 | FRUKO | 1.04 | FSAFU | 1.18 | FSAKO | 1.18 |
| FSAOY | 1.12 | FSASU | 1.14 | FSEMI | 1.02 | FSESE | 1.16 |
| FSEUT | 1.14 | FSHHA | 1.10 | FSHOO | 1.14 | FSHTA | 1.22 |
| FSUSA | 1.12 | FSUTS | 1.00 | FTAKU | 1.20 | FTAKZ | 1.16 |
| FTANA | 1.14 | FTATA | 1.00 | FTATO | 1.16 | FTAYO | 1.18 |
| FTESU | 1.24 | FTOCH | 1.22 | FTOKO | 1.12 | FTOOK | 1.08 |
| FTOOO | 1.00 | FTOTZ | 1.10 | FTSOG | 1.20 | FWAOZ | 1.04 |
| FYAMI | 1.20 | FYAYO | 1.08 | FYOMO | 1.06 | FYONA | 1.12 |
| FYOTA | 1.20 | FYOTO | 1.10 | FYOTZ | 1.12 | FYUAM | 1.00 |
| FYUAR | 1.14 | FYUFY | 1.16 | FYUKA | 1.12 | FYUKZ | 1.04 |
| FYUMI | 1.10 | FYUNV | 1.10 | FYUOO | 1.14 | FYUSZ | 1.16 |
| FYUTZ | 1.10 | FYUYA | 1.02 | | | | |
| MAKKI | 1.00 | MAKYO | 0.90 | MASSA | 0.90 | MCHSA | 0.90 |
| MDAIN | 0.92 | MDAMA | 0.84 | MFUAK | 0.96 | MHAMO | 1.04 |
| MHIAT | 1.02 | MHIHI | 0.94 | MHIKY | 0.92 | MHIKZ | 1.00 |
| MHIMA | 0.96 | MHIMO | 0.94 | MHIOY | 0.98 | MHISU | 1.00 |
| MHISW | 1.02 | MHISX | 1.00 | MHITO | 0.96 | MHIUE | 0.96 |
| MHIYO | 0.92 | MJUIS | 0.90 | MKAIS | 1.00 | MKASE | 1.00 |
| MKAYY | 1.00 | MKENO | 0.94 | MKETA | 0.94 | MKOTA | 0.94 |
| MKOTX | 0.94 | MKUHA | 1.02 | MMAAZ | 0.94 | MMAFU | 0.94 |
| MMAII | 0.90 | MMAKV | 0.94 | MMAKW | 1.02 | MMAMX | 0.96 |
| MMANA | 0.98 | MMASY | 0.90 | MMATZ | 0.96 | MMAUZ | 1.02 |
| MMIKA | 1.00 | MMONA | 0.94 | MMOTA | 0.98 | MNAMO | 0.92 |
| MNOOG | 1.00 | MNOSH | 0.94 | MOSIG | 0.98 | MOSKA | 0.86 |
| MOSMI | 1.00 | MRYAR | 0.94 | MRYHA | 1.00 | MRYMO | 0.92 |
| MRYSA | 0.98 | MSANU | 0.98 | MSASU | 1.00 | MSATA | 1.00 |
| MSHIS | 1.04 | MSHMA | 0.88 | MSHMI | 0.92 | MSHOO | 0.88 |
| MSHSO | 0.94 | MTAAI | 0.96 | MTADA | 1.00 | MTAHI | 1.00 |
| MTAIZ | 1.00 | MTAKX | 1.00 | MTAMO | 0.98 | MTANA | 1.02 |
| MTAOO | 0.98 | MTASU | 0.92 | MTATA | 0.94 | MTATZ | 0.98 |
| MTAWA | 0.92 | MTAYA | 0.96 | MTAYZ | 0.98 | MTEIS | 1.00 |
| MTEMI | 0.96 | MTEMO | 0.92 | MTEOO | 1.00 | MTOAZ | 0.94 |
| MTOHO | 0.96 | MTOKI | 0.92 | MTOKO | 0.80 | MTOKZ | 0.90 |
| MTOMA | 0.94 | MTONI | 1.00 | MTSMI | 0.92 | MYAME | 0.98 |
| MYAMZ | 0.98 | MYANA | 1.00 | MYOIS | 1.04 | MYOKU | 0.98 |
| MYOMA | 1.04 | MYOSO | 0.96 | MYUAI | 0.92 | MYUFU | 1.00 |
| MYUIM | 1.00 | MYUKU | 0.98 | MYUMI | 0.94 | MYUYA | 1.06 |

表 3: Names of the speakers used in the gender independent training set

| Speaker | Best time (s) | Speaker | Best time | Speaker | Best time |
|---|---|---|---|---|---|
| FAKKA | 30.00 | FAKMA | 30.00 | FAKMI | 30.00 |
| FAKMU | 30.00 | FAKTS | 30.00 | FAKYZ | 30.00 |
| FASYA | 30.00 | FAYKZ | 30.00 | FCHAK | 30.00 |
| FCHHA | 30.00 | FCHKA | 30.00 | FCHMA | 29.89 |
| FCHNO | 30.00 | FEMNA | 30.00 | FEMNI | 30.00 |
| FHIMU | 30.00 | FHIOY | 30.00 | FKEAO | 30.00 |
| FKESH | 30.00 | FKIFU | 30.00 | FKYIW | 30.00 |
| FMAKS | 30.00 | FMAKY | 30.00 | FMASH | 30.00 |
| FMASW | 30.00 | FMATP | 30.00 | FMEWA | 30.00 |
| FMIIS | 30.00 | FMINP | 30.00 | FMIOW | 30.00 |
| FNAMU | 30.00 | FNOKU | 30.01 | FNOSA | 30.00 |
| FNOSZ | 30.00 | FRIMI | 30.00 | FRUKO | 30.00 |
| FSAFU | 30.00 | FSAOY | 30.00 | FSUSA | 30.00 |
| FSUTS | 30.00 | FTATA | 30.00 | FTAYO | 30.00 |
| FTESU | 30.00 | FTOKO | 29.99 | FTOTZ | 29.99 |
| FYUFY | 30.00 | FYUKZ | 30.02 | FYUNV | 30.00 |
| FYUTZ | 30.00 | FYUYA | 30.01 | | |
| MAKKI | 30.00 | MAKYO | 30.00 | MFUAK | 30.00 |
| MHAMO | 30.00 | MHIAT | 29.96 | MHIMO | 30.00 |
| MHIOY | 30.00 | MHISX | 30.00 | MJUIS | 29.99 |
| MKAIS | 30.03 | MKASE | 30.00 | MKAYY | 30.00 |
| MKETA | 30.00 | MKOTA | 30.00 | MMAMX | 30.00 |
| MMANA | 30.00 | MMATZ | 30.00 | MMAUZ | 30.00 |
| MMIKA | 30.00 | MRYAR | 29.99 | MRYHA | 30.00 |
| MRYMO | 30.00 | MRYSA | 30.00 | MSANU | 30.00 |
| MSATA | 30.00 | MSHMA | 30.00 | MSHOO | 30.00 |
| MSHSO | 30.00 | MTAAI | 30.05 | MTADA | 30.00 |
| MTAHI | 30.00 | MTAIZ | 30.00 | MTAMO | 30.00 |
| MTANA | 30.00 | MTAOO | 29.81 | MTATA | 30.00 |
| MTATZ | 30.00 | MTAWA | 30.00 | MTAYA | 30.01 |
| MTEIS | 30.00 | MTEMI | 30.00 | MTOAZ | 30.00 |
| MTOHO | 30.00 | MTOMA | 30.00 | MTONI | 30.00 |
| MYAME | 30.00 | MYANA | 30.00 | MYOKU | 30.00 |
| MYUMI | 30.00 | MYUYA | 30.00 | | |

表 4: Names of the speakers used in the gender dependent training set

| | | | | |
|---|---|---|---|---|
| FAKKA | FAKMA | FAKMI | FAKMU | FAKTS |
| FASYA | FAYKZ | FCHAK | FCHHA | FCHKA |
| FCHMA | FCHNO | FCHYZ | FEIFU | FEMNA |
| FEMNI | FHIMU | FHIOY | FKEAO | FKEHO |
| FKIFU | FKYIW | FMAKS | FMATP | FMEWA |
| FMIIS | FMINP | FMIOW | FMISO | FMOMA |
| FNAIR | FNOKU | FNOSA | FNOSZ | FRUKO |
| FSAFU | FSAOY | FSUSA | FSUTS | FTATA |
| FTAYO | FTESU | FTOKO | FTOTZ | FYUAR |
| FYUFY | FYUKZ | FYUNV | FYUTZ | FYUYA |
| MAKKI | MAKYO | MFUAK | MHAMO | MHIAT |
| MHIMO | MHIOY | MHISX | MJUIS | MKAIS |
| MKASE | MKAYY | MKETA | MKOTA | MMAMX |
| MMANA | MMATZ | MMAUZ | MMIKA | MRYAR |
| MRYHA | MRYMO | MRYSA | MSANU | MSATA |
| MSHMA | MSHOO | MSHSO | MTAAI | MTADA |
| MTAHI | MTAIZ | MTAMO | MTANA | MTAOO |
| MTATA | MTATZ | MTAWA | MTAYA | MTEIS |
| MTEMI | MTOAZ | MTOHO | MTOMA | MTONI |
| MYAME | MYANA | MYOKU | MYUMI | MYUYA |

表 5: Actual speech times for male speakers

| Speaker name | Speech time (s) | Speaker name | Speech time (s) | Speaker name | Speech time (s) |
|---|---|---|---|---|---|
| MFUAK | 47.78 | MHIYO | 27.93 | MTOKI | 19.65 |
| MTAYZ | 29.42 | MYOIS | 29.38 | MYUYA | 79.99 |
| MMAKV | 23.35 | MMAII | 26.85 | MJUIS | 31.70 |
| MMAFU | 14.85 | MYUMI | 44.97 | MTAMO | 41.86 |
| MTADA | 32.21 | MYOMA | 24.67 | MSANU | 61.26 |
| MTSMI | 25.68 | MMATZ | 41.70 | MTAKX | 14.04 |
| MTEOO | 13.31 | MHIHI | 19.65 | MTASU | 27.60 |
| MHITO | 26.28 | MMOTA | 20.21 | MHISW | 28.58 |
| MYOKU | 34.63 | MTAHI | 44.56 | MTEIS | 53.92 |
| MYANA | 32.15 | MMAKW | 15.79 | MTOMA | 61.30 |
| MTAOO | 30.21 | MRYAR | 31.00 | MNOSH | 25.94 |
| MKASE | 32.31 | MKOTX | 25.11 | MRYSA | 40.35 |
| MYAME | 85.84 | MSATA | 32.09 | MMONA | 28.35 |
| MASSA | 22.14 | MTAAI | 30.61 | MKAIS | 30.61 |
| MYUAI | 16.98 | MSHMI | 25.64 | MTOKZ | 16.32 |
| MHIAT | 30.18 | MHISX | 34.78 | MSHSO | 50.34 |
| MHIMA | 27.43 | MKUHA | 18.61 | MHIMO | 56.31 |
| MMANA | 32.23 | MYAMZ | 28.65 | MTAWA | 32.37 |
| MSASU | 24.16 | MDAIN | 25.63 | MTOAZ | 43.93 |
| MNAMO | 29.62 | MMAUZ | 30.63 | MHIOY | 31.05 |
| MAKYO | 56.81 | MTOHO | 64.40 | MMAAZ | 25.39 |
| MHISU | 28.89 | MKENO | 27.56 | MNOOG | 26.94 |
| MMASY | 20.39 | MHAMO | 73.10 | MSHMA | 44.20 |
| MOSKA | 15.81 | MTANA | 32.42 | MTAYA | 31.70 |
| MYOSO | 17.22 | MTONI | 42.26 | MRYMO | 52.41 |
| MKOTA | 42.85 | MMAMX | 46.86 | MYUIM | 15.04 |
| MCHSA | 16.03 | MSHOO | 40.59 | MAKKI | 34.44 |
| MTATZ | 32.39 | MHIUE | 17.34 | MOSMI | 19.95 |
| MTATA | 50.22 | MOSIG | 24.54 | MDAMA | 23.18 |
| MRYHA | 40.47 | MSHIS | 24.72 | MKETA | 45.36 |
| MYUFU | 29.72 | MYUKU | 16.11 | MHIKY | 22.58 |
| MHIKZ | 28.47 | MTAIZ | 48.19 | MTEMO | 20.03 |
| MTEMI | 35.00 | MKAYY | 38.99 | MTOKO | 20.30 |
| MMIKA | 31.89 | | | | |

表 6: Actual speech times for female speakers

| Speaker name | Speech time (s) | Speaker name | Speech time (s) | Speaker name | Speech time (s) |
|---|---|---|---|---|---|
| FYUAM | 14.03 | FKEHO | 32.46 | FMATP | 55.30 |
| FMASH | 44.28 | FYOTZ | 55.91 | FTOTZ | 34.02 |
| FMIYA | 17.78 | FAMIS | 31.85 | FTATA | 36.48 |
| FNAMU | 39.49 | FKESH | 60.54 | FYUNV | 34.50 |
| FAYHZ | 21.78 | FCHYO | 19.64 | FMIHX | 20.86 |
| FSUSA | 49.92 | FTATO | 15.67 | FAKYY | 15.20 |
| FKOTE | 28.68 | FHIOY | 47.83 | FNOKY | 24.05 |
| FTAYO | 60.21 | FKAKA | 28.40 | FAKKI | 17.46 |
| FNOKU | 32.97 | FCHSA | 18.66 | FMAIM | 24.25 |
| FCHMA | 30.51 | FKISU | 25.39 | FKEKO | 52.73 |
| FSEUT | 45.43 | FTOOK | 33.56 | FEIAO | 53.82 |
| FMAKS | 38.80 | FSHHA | 26.05 | FMISH | 39.53 |
| FEIOO | 24.55 | FJUSH | 31.89 | FTSOG | 17.54 |
| FAYIS | 26.35 | FSEMI | 36.35 | FMISO | 60.57 |
| FYAYO | 25.19 | FYOTA | 45.14 | FYUTZ | 63.20 |
| FJUSZ | 25.87 | FAKYA | 21.21 | FSHTA | 41.27 |
| FCHNO | 42.74 | FCHHA | 33.38 | FTOKO | 31.70 |
| FRIMI | 42.21 | FAKSU | 29.87 | FYOTO | 22.05 |
| FKOMI | 19.96 | FAKMU | 39.36 | FKEKA | 18.25 |
| FREOO | 26.73 | FSASU | 41.78 | FKIFU | 49.58 |
| FMEWA | 132.90 | FEIFU | 31.64 | FSAFU | 57.65 |
| FAYKZ | 32.10 | FCHKI | 24.87 | FYUKA | 24.28 |
| FNOHI | 66.94 | FAKMA | 69.14 | FSAOY | 39.81 |
| FSUTS | 51.02 | FKASA | 26.25 | FKEIM | 22.61 |
| FAKMI | 47.63 | FMIOW | 32.64 | FYUSZ | 26.47 |
| FNOHX | 18.83 | FNOMU | 71.41 | FSHOO | 29.26 |
| FMION | 25.53 | FMIOO | 27.69 | FYOMO | 26.28 |
| FYUOO | 54.87 | FCHYZ | 33.41 | FYUKZ | 32.12 |
| FWAOZ | 17.46 | FMAKY | 32.82 | FYUMI | 38.31 |
| FAKKA | 39.19 | FAKTS | 42.27 | FYUYA | 30.67 |
| FHIMU | 47.73 | FMINP | 32.04 | FMIIS | 37.84 |
| FCHKA | 51.98 | FRUKO | 35.73 | FMITX | 22.09 |
| FKEAO | 66.75 | FMITZ | 29.59 | FCHKY | 18.09 |
| FASYA | 48.33 | FMAYO | 49.62 | FMITO | 18.75 |
| FKESZ | 29.27 | FNOSA | 33.54 | FYAMI | 27.20 |
| FMITA | 18.37 | FYONA | 26.06 | FEMNA | 44.96 |
| FRESU | 14.73 | FMOMA | 37.26 | FTANA | 35.03 |
| FMAMO | 28.36 | FEMNI | 51.49 | FCHAK | 43.95 |
| FSESE | 42.41 | FNOSZ | 46.54 | FNAIR | 45.42 |
| FHATA | 22.85 | FYUFY | 31.54 | FTAKU | 37.41 |
| FYUAR | 58.43 | FTESU | 79.11 | FTAKZ | 29.06 |
| FREII | 22.53 | FTOOO | 23.73 | FSAKO | 72.54 |
| FTOCH | 57.53 | FMASW | 72.78 | FKYIW | 71.06 |
| FAKYZ | 44.86 | | | | |

# 付録 B Software

This section describes the software and scripts we used for training our models from a user's point of view. All the software that was used for the training of the models comes as a compressed archive named `training.tar.gz`. It contains the scripts, the ASSM toolkit and the modified version of HCode we used. If you intend to use this software, you will first have to decompress it, using the following commands:

```
gunzip training.tar.gz
tar -xvf training.tar
```

The data used for training can also be found on a DAT (backup date December 8th). This backup contains the wave files for the training speakers from the sets `T_M_0100.ascii` and `T_F_0130.ascii`

## B.1 Building the ASSM package

The ASSM v3.0 package is stored in the directory `ASSM_3.0`. It comes with all the source files and corresponding makefiles.

To build it, you will have to set the TCP/IP port which the ASSM server will use. To do this, open the file `includes/ServerDefines.h` and look for the line:

```
#define ASSMSERVERTCPIPPORT <port>
```

where `<port>` is the port number. Choose a free port number (you will perhaps need to run the `netstat` command) and set it instead of the current port number.

Then, choose the target machine type (the type of the machine where the package will be running). To do this, open the file `Make.glob` and look for the line:

```
MACHINE         = <type>
```

where the accepted values for `<type>` are `HPUX`, `SUN`, `SOLARIS`, `ALPHA` or `LINUX`. Put the type of your choice there and save the file.

The package is then built using the command:

```
make
```

However, the training operations being time consuming, we preferred to build four versions of binary executables so as to be able to spread our training jobs over several machines. For this purpose, we used the script `build.py`.

## B.2 Using the Scripts

The scripts are stored in a directory called `scripts`. If no `cd <directory>` command line is given, the command line syntaxes are given assuming that the current directory is the one containing all the scripts.

- `build.py` is used to build four versions of executables from the ASSM package for the same machine type. The command line syntax is:

  ```
  cd ASSM_3.0
  python ../scripts/build.py <type> <port1> <port2> <port3> <port4>
  ```

  where `type` is the target machine type and `port n` is a valid TCP/IP port number.

- `balancesets.py` is used for gender independent training in order to adjust the number of male and female speakers in the set to the same value.

- `bestwarps.py` is used to determine the most likely warping factor for each speaker of the training set, after the scoring operation.

- `check.py` is used to check the coherence between the master label files during the acoustic segmentation operation. Since this script is allowed to remove speakers and labels from the master label files, it can be wise to keep a backup copy of each of the master label files before each run of `check.py`. Any modification to the label files is printed to the standard output. The script should be run after each new master label file was computed. This applies only to acoustic segmentation. The name of the master label file should be appended to the array called `labelfiles`, as is shown in the script.

- `cseg.py` lets you perform the constrained acoustic segmentation task for one given warping factor. Before running this script, the current warping factor value (as a 3 digit number) must be set in the variables `fvpath` and `outputlbdir`. To use several servers at one time, it may also be necessary to modify the `binname` variable, too. When running with several servers at one time, no more than one server per machine should be run to avoid network congestion. Given the protocol used between the servers and their clients, congestion on one machine can lead to data losses.

- `fea.py` is used to extract the features from the wave files. The array called `warps` is used to specify the warping factors used for feature extraction. The format for the warping factor is a string. Thus, several feature extraction processes can be run at the same time.

- `get.py` reads a `.ascii` file from the DB/SDB database, cuts the silences from the wave files according to their corresonding transcription files and stores the results into the waves directory.

- `km.py` is used for the divisive and k-means clustering task.

- `rewrite.py` removes all speakers whose total speech time is less than 20 s from the set description file.

## B.3 Defaults

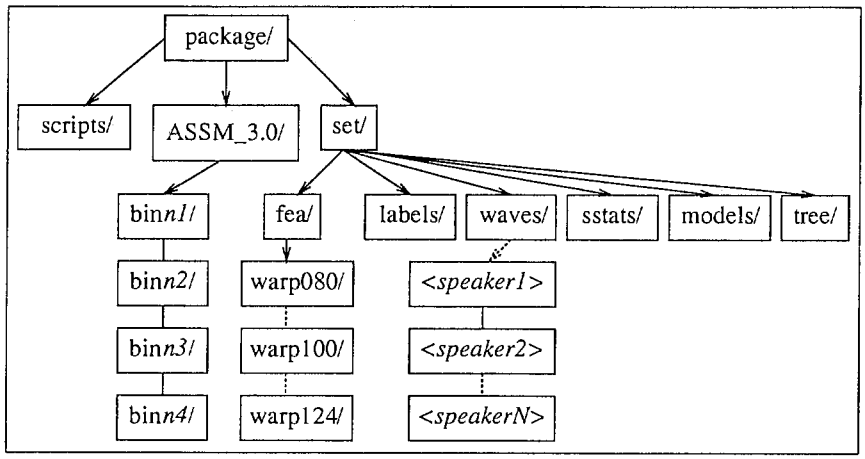


図 9: Directory structure assumed by the scripts

The provided set of scripts makes assumptions about file names and directories as shown in Fig. B.3. To modify the default names and formats, changing variables in the scripts is enough. These scripts rely on the format of the set description files for SDB/SLDB (`.ascii` files), too.

- There are two kinds of master label files: those computed during segmentation (seg mlfs) and those computed during scoring (score mlfs).

  - The following name format for the seg mlfs is expected: `<basename><nnn>`, where `<basename>` is any valid string representing a file name and `<nnn>` stands for the string representation of the warping factor using always 3 digits.

  - The following name format for the score mlfs is expected: `<basename><nnn><ext>`, where `<basename>` is any valid string representing a file name, `<nnn>` stands for the string representation of the warping factor using always 3 digits and `<ext>` is any valid file extension.

- The directories holding all the features for one given arping factor are named after that warping factor, by appending the warping factor, as a 3 digits string to the string `warp`.

- The speech-only wave file (computed by get.py or retrieved from the backup tape) names are built as follows: `<conversation>.<time>.<side>.<chunk>.16k` where `<chunk>` is an even number used to count the speech-only parts of the conversations.

- The sufficient statistic master files are named by default after the current iteration number: `<basename>.<iteration_number>`

- The set description files go to the set directory by default.

- All scripts using a server (`cseg.py`, `useg.py`, `score.py`, `km.py`) write a server configuration file (named `serverconfig <bin_name>`, where `<bin_ name>`) and a feature vectors file list (named `fealist <bin_name>`) in the current directory.

- These scripts rely on the `ps` and `grep` commands to get the server's pid. Since this combination of commands relies on the length of the screen line, we recommand the use of terminals with lines greater than 80 to avoid any information loss. Short line lengths can lead to errors in the results of the grep command, the scripts may consequently block waiting for a result that will never come.