The SOM trajectory as a complementary speech representation in multi-stream speaker recognition

Dalei Wu, Andrew C. Morris and Jacques Koreman

Institute of Phonetics, Saarland University, Saarbrücken, Germany
{daleiwu, amorris, jkoreman}@coli.uni-saarland.de

Abstract

Multi-stream speech processing, in which multiple speech representations are combined at some stage during processing, is well established as a sure way to improve speech or speaker recognition performance. For constructive combination it is important that the different speech representations used are as complementary as possible. In this paper, inspired by an analogy with signature verification, we look at the speech trajectory in a trained Kohonen self-organising map (SOM) as a new speech parameterisation. We test these features in combination with MFCCs by both feature concatenation and linear score fusion, for a speaker identification and a speaker verification task using the CSLU speaker recognition database. SOMs of different sizes and different numbers of dimensions are tested. It is found that late score fusion using a 20 x 20 SOM gives a relative error improvement over the state of the art MFCC baseline of 6% for identification and 10% for verification.

1. Introduction

Multi-stream speech processing operates on multiple different representations of the speech signal either at the feature level (concatenating them and then treating them as a single stream), at the frame level (combining the likelihood of each stream for each frame), or at the score level (processing each stream separately and then combining the results) [5](Hagen)[12](Morris).

While speaker recognition accuracy can be quite high using a single feature stream, it is often possible to further enhance the level of accuracy by multi-stream processing. In particular, it can alleviate the problem of performance degradation in noisy conditions.

The additional benefit provided by each new feature stream in a multi-stream system depends on the degree to which the information in the new stream is complementary to the information already available.

In this paper we investigate the combination of MFCC features, commonly used in speaker recognition systems, with features derived through the projection of these MFCC features onto a trained Kohonen self-organizing map (SOM) [9](Kohonen ‘97). In the case of a 2 dimensional SOM, the trace of the cells in the map visited throughout the utterance of a short spoken prompt provides a 2D trajectory which can be processed in an analogous way to that in which signature recognition is usually performed. This would have the additional advantage that, in a multimodal system which uses both voice and signature, such as that used in the SecurePhone project [1][Allano], the voice and signature data, after SOM projection, can share the same processing. SOM trajectory coefficients (STC), besides capturing local information in the SOM coordinates (and their derived features), also allow us to model the global shape of the voice “signature”. As in signature verification, STC features would consist not simply of SOM trajectory coordinates, but also of features derived from these.

In Section 2 we describe how we can use a SOM for projecting high dimensional speech data down to a low dimensional SOM, demonstrating the tonotopic organisation on the basis of a labelled database. In Section 3 we show some visual examples of the use of a 2-dimensional SOM to represent the speaker trajectory for a given prompt. In Section 4 we describe how the SOM coordinates are augmented by the addition of a number of derived features in a manner analogous to the processing used in signature recognition. In Section 5 we present our baseline GMM based system for signature recognition. In Section 6 we describe the tests we have made, the results of which are presented in Section 7, both for the concatenation of MFCC with STC (SOM trajectory coordinates) features and for MFCC with STC scores fusion. A discussion and conclusion follows in Sections 8 and 9.
2. SOM projection

The SOM training procedure is a form of unsupervised clustering which is in some ways similar to K-means clustering. What distinguishes them is that in SOM clustering the cluster centres are arranged in a regular grid (normally in two dimensions), in which cluster centres which are close in the grid are also close to each other in the codebook vector space. We will refer to a SOM which has this property as being “well organised”.

Although the training procedure for a SOM is well known, it includes a number of steps the details of which often differ between implementations. These differences can have a significant effect on the outcome of SOM training, so we give here some of the details of our implementation. The SOM training procedure was implemented, in C++ and with the aid of the Torch API [3], using the algorithm from [10][Kohonen SOMPAK]. All training (and test) vectors are first normalised to have unit length. SOM codebook vectors are initialised to the value of randomly selected training data vectors (MFCC speech frames). We will refer to the individual speech feature frames \( x_t \) used in SOM training as training tokens.

Let \( t \) be the absolute token count and \( u \) the token count within the training set. Let \( r(t) \) be the update radius and \( h(t,r) \) be the learning rate. Let the closest codebook vector to the current training token be referred to as the active codebook vector. Let \( dst \) be the Euclidean distance between the active codebook vector and vector being updated.

For each training token \( x_t \), all codebook vectors \( m_i \) within grid distance \( r(t) \) of the active codebook vector are updated according to (1) and then renormalized to have unit length.

\[
m_i = m_i + h(t,r).(m_i - x_t)
\]  

(1)

The radius \( r(t) \) of the update neighbourhood and the learning rate \( h(t,r) \) are updated once every bsize codebook updates, according to (2, 3, 4).

\[
len = \frac{s \cdot \text{epochs}}{bsize}
\]  

(2)

\[
r(t) = 1 + \frac{(r(0)-1) \cdot (len-t)}{len}
\]  

(3)

\[
h(t,r) = 0.05 \frac{len}{len + 100t} \exp \left( -\frac{dst^2}{2r^2(t)} \right)
\]  

(4)

When the trained SOM is used to map a given pattern vector, \( x \), onto the SOM grid, we will refer to the SOM grid coordinates of the SOM codebook vector which is closest to this vector as the 2D “SOM projection” of \( x \). In this way a trained 2D SOM can be used to project a data set with \( N \) dimensions onto a corresponding dataset with just 2 dimensions.

Projection of data onto a 2D SOM is often used as a tool for visualising high dimensional data. If the intrinsic dimension of the pattern data has more than two dimensions, then the SOM training algorithm will not usually be able to produce a well organised 2D map. However, in practice it is quite common for a 2D SOM to organise in this way. For example, it is well known that when a SOM is trained on the acoustic feature vectors for speech data which is restricted to vowels, it will produce a well organised “tonotopic map” [8][Kohonen NPTypwriter] which resembles some symmetry of the “vowel triangle” shown in Fig.1.

![Fig. 1. Vowel triangle for Am.E. [6][7]](image-url)

This results from the fact that all vowels are perceived according the centre frequency of their two first vocal tract resonances, or “formants” (F1, F2). This means that, no matter what type of acoustic features are used to represent the speech data, this data always has an intrinsic dimension of 2. In this case, if the sequence of vowel sounds pronounced varies continuously in time, then the 2D trajectory of the corresponding SOM projection of this speech data will also vary continuously in a smooth path moving over the SOM grid.
Fig. 2a shows a SOM which was trained with all realisations of the vowels /iy, ey, aa, ow, uw/ in the TIMIT database [4] (Garofolo, 1993). TIMIT was used for this labelled SOM instead of the CSLU Speaker Recognition corpus, which is used in the experiments in the rest of this paper [2] (Cole, 1998), because, unlike TIMIT, the CSLU database is not phonetically labelled. Despite variability in the realisation of each vowel, the acoustic similarity between their different realisational variants leads to their self-organisation in large contiguous areas representing each of the vowels. Acoustically more similar vowels, e.g. /iy/ and /ey/ or /uw/ and /ow/, are generally closer together in the SOM.

In the application of SOM projection which we are investigating in this article we should like to be able to project all phonemes onto a single SOM in such a way that all speech trajectories are quite smooth. When consonants are represented together with vowels in the same SOM, however, its structure becomes somewhat less clear. Figure 2b visualises a SOM based on all the phones in the TIMIT database. As in the vowel SOM in Figure 2a, contiguous areas representing the same phone can be recognised. Although the phone map has a higher intrinsic dimensionality than the vowel map, we can clearly recognise that acoustically similar phones are located close together in the SOM. For instance, plosive closures are often close together, like /pcl, bcl, dcl/ in the top left-hand corner, labiodental fricatives /f, v/ are close together, syllabic consonant /el, er, em, en, eng/ are close to their non-syllabic counterparts /l, r, m, n, ng/ and, as in Figure 2a, acoustically similar vowels are closer together.

3. Speaker voice signature

Although it is clear that SOMs can be used to represent speech, as in [Kohonen, Neural Phonetic Typewriter], it is not immediately obvious that the self-organising structure retains the finer distinctions between speakers in the way they produce the same phoneme; the discretisation caused by the size of the SOM dimensions may not be fine enough to retain speaker differences, which are more subtle than the distinctions between phones.

Each time a speaker produces an utterance, a corresponding graphic pattern can be visualized by showing the mapping of each MFCC frame for a prompt to the SOM coordinates. The resulting graphic pattern is referred to as a Speaker Voice Signature (SVS). It is defined as the speaker-specific trajectory for an utterance visualized by a trained SOM classifier, showing some similarity to a written signature. Speaker differences for a given prompt are reflected in differences in the trajectories through the SOM space. Figure 3 demonstrates the trajectories for the speech signal corresponding to “two four” from the same prompt spoken by two speakers. Although the intra-speaker differences (comparison between left and right figures) are smaller than the inter-speaker differences (comparison between top and bottom figures), they are fairly subtle.
The speaker discriminating information is illustrated not only by these trajectories, but also by the time duration for which the trajectory stays in a position. The bigger blob shown in each position implies that the trajectory stays there for a longer time. In fact, the distribution of the time duration can be efficiently modelled by a GMM with the two x-y coordinates of the SOM features. The global shape of each SVS can be captured by the other six components of the SOM features such as x-y speeds, x-y accelerations and the curvature.

4. Computing SOM trajectory coefficients

In on-line handwritten signature verification, the feature vector associated with each pen position (x,y coordinates, plus pen pressure and angles) is generally augmented with a number of derived parameters before it is submitted for data modelling. These extra parameters can be derived from local features associated with the first and second time derivatives of the position, including velocity, acceleration, line direction, and curvature. They can also be derived from global features, such as the first and second moments, which carry information about the overall shape of the trajectory. Exploiting the analogy with on-line signature verification, the vector of SOM projection coordinates associated with each speech frame (which we refer to as “raw” SOM features) is similarly augmented.

In this paper we only look at augmenting STCs with time derivative based features. Time difference coordinates for a sequence of points \( x(t) \) on a smooth trajectory could be estimated simply as \( x(t+1) - x(t-1) \). However, a smoothing is normally applied to this estimate by estimating the direction of the velocity vector as that of the least squares fit straight line between the points \( x(t-W) \) to \( x(t+W) \) inclusive, for some given smoothing window size, \( W \). Indeed it is clear from Figure 3 that SOM trajectories are generally not entirely smooth. This is partly because of the discrete nature of the SOM grid positions, but it is also partly due to the imperfectly tonotopic mapping which any SOM trained on unconstrained speech data will tend to exhibit. If \( x(t) \) is any coordinate at time \( t \), (5) shows the regression formula of order \( W \) (Soong et al. 1988; Van et al. 2004).

\[
\text{reg}(x(t), W) = \frac{\sum_{k=1}^{W} k(x(t + k) - x(t - k))}{2\sum_{k=1}^{W} k^2}
\]  

As the SOM clustering algorithm can be applied not only to a 2D grid, but to a grid with any number of dimensions, we are interested in testing the suitability of SOM grids with not only 2 dimensions, but also 3 or more dimensions.

The augmented \( stc(t) \) feature vector which we derive from the N-dimensional SOM coordinate data \( x(t) \), are as follows.

- **Position.** Handwritten signature coordinates are relative to the signature centre of gravity, but SOM grid position is highly significant, so we use absolute SOM coordinates. Let \( d \) denote dimension.

\[
\text{For } d = 1..N, \text{ use } x(d, t)
\]
• **Velocity.** For \( d = 1..N \), use

\[
\frac{\text{d}x(d, t)}{\text{d}t} = \frac{\text{d}x(d, t)}{\text{d}t} = \text{reg} \left( x(d, t), W \right) \tag{7}
\]

• **Line angle.** For \( d = 2 \) to \( N \), use

\[
\psi(d, t) = \arctan \left( \frac{\text{d}x(d, t)}{\text{d}x(d-1, t)} \right) \tag{8}
\]

• **Curvature.** For \( d = 2 \) to \( N \), for arc length \( s \), use

\[
\kappa(d, t) = \frac{\text{d}^2x(d, t)}{\text{d}s^2} = \frac{\text{d}^2x(d, t)}{\text{d}t^2} = \text{reg} \left( \psi(i, t), W \right) \left\| \nu(t) \right\| \tag{9}
\]

Further time derivatives: \( \frac{\text{d}\psi(i, t)}{\text{d}t} \) and \( \frac{\text{d}\kappa(i, t)}{\text{d}t} \)

For 2 SOM dimensions, \( \text{stc}(t) \) has 8 coordinates. For \( N \) SOM dimensions, \( \text{stc}(t) \) has \( 6N-4 \) coordinates.

## 5. Baseline speaker recognition systems

**MFCC PREPROCESSING**

The speech signals were recorded over a digital telephone line at a sampling rate of 8 kHz and with a 16-bit amplitude resolution. From these signals, 20 Mel-scaled filterbank log power features were extracted over 20ms frames and with a 10ms step size, using a Hamming window and a pre-emphasis factor of 0.97. A DCT was then applied to these to obtain MFCC features, from which the c0 energy coefficient was dropped. Cepstral mean subtraction (CMS) was applied to the MFCC vector and time difference features were appended. Thus, 38-dimensional MFCC vectors were used as the input data for the experiments reported in this article.

**IDENTIF. AND VERIFICATION MODELLING**

Both speaker identification and verification tests use state of the art systems based on Gaussian mixture models (GMMs) and MFCC speech features. As in [15][Reynolds 95] our identification system trained a UBM on data from a large set of speakers who are not used in testing. A client GMM distribution is then trained on data from each client, using only data for the prompt being tested (in this case the first numbers prompt, “58312?”). The client model is initialised equal to the UBM, after which the Gaussian means alone are updated using MAP adaptation [13][Mariethoz]. In testing, the speaker for each test utterance is selected as the person who has the highest speech data likelihood. In verification the UBM and client models are trained in the same way, and a test utterance is accepted as having the claimed identity if the UBM-normalised data likelihood is above a certain threshold, where the value of this threshold is set to give the maximum verification score on a development test set. GMM training and testing used the Torch machine learning API [3].

## 6. Speaker verification experiments

A subset of 61 speakers was selected from the CSLU database. Each speaker has 12 recording sessions ... can’t remember exact way CSLU was divided for training, development & evaluation (must be described in one of our articles).

In order to increase the chances of finding a SOM which is well organised for all speech sounds together, we experiment with SOMs with dimension from one to seven, with different numbers of grid cells per dimension, and using different numbers of updating iterations in SOM training.

### 6.1. Data

The same 5-digit sequence (“5 3 8 2 4”) spoken by 61 speakers was selected from the CSLU Speaker Recognition corpus [2][CSLU]. The prompts were recorded in twelve sessions, with four repetitions of the prompt in each session. Because we are particularly interested in speaker recognition for very small amounts of data, we only selected three sessions per speaker, one for training, one for development and one for testing. Only sessions in which all the prompts were produced correctly were selected.

These MFCC features were either used as input to the GMM directly (baseline experiment), or were used as input to the SOM, which varied in the number of dimensions or in the size of the dimensions. After allowing the SOM to self-organize on the basis of the training data obtained from a single session, with data from another session being used for optimization, it was used to map the MFCC features onto the x-y coordinates of the SOM, from which the parameters described in the previous section were then derived. The SOMs that were used for MFCC feature projection onto x-y coordinates was varied in the number of dimensions, which was varied between 2 and 5, keeping the total number of units in the SOM roughly the same. Best results were obtained for 2- and 3-dimensional SOMs, so only these are presented here. For the 2-dimensional SOM, the size of the dimensions was also varied (but always the same for each dimension). The speaker’s voice SOM trajectory coefficients which derived from x-y coordinates of the trajectory through the SOM was used as input to GMM, described in the following section.
6.3. Fusion

The effect of fusion is investigated by comparing early fusion, in which the STC parameter vector is concatenated with the MFCC vector before it is used as input to the modelling stage, with late fusion, in which the scores of the separately modelled MFCCs and STCs were linearly combination. The weights given to each type of scores (speaker recognition scores for MFCCs and STCs, respectively) are varied from 0 to 1 in steps of 0.1, with their summation equal to 1.

7. Test Results

In this section the results for speaker identification as well as for speaker verification experiments are presented. In speaker identification, the system’s task is to determine which of the 61 speakers is the most likely to have uttered a given test prompt. In speaker verification, the system must accept or reject the claimed identity of the speaker. Only results for the optimal smoothing factor (sm.) and MFCC/STC weight are presented in the tables. <WHY NO WEIGHT IN TABLES?>.

The results of the speaker identification experiments using STCs are presented in Table 1. The baseline result for speaker identification for MFCCs is 20.9% identification error.

Clearly, the STCs obtained from the SOM always lead to a substantially higher speaker identification error than the baseline. Also, early fusion does not improve speaker identification, and is between 2.9 and 5.7 percent points higher for the SOMs presented. Late or score fusion, on the other hand, does lead to a modest improvement in speaker identification. The improvement for late fusion of the scores for MFCCs with those of STCs obtained from a 20x20 SOM is 1.2 percent points, or 6% relative to the baseline speaker identification error.

Table 1: Identification percentage error for STCs alone and in early and late fusion with MFCCs, for different SOM sizes and number of dimensions (sm = optimal smoothing factor, cf. Section ??).

<table>
<thead>
<tr>
<th>Size</th>
<th>num cells</th>
<th>no fusion</th>
<th>early</th>
<th>late</th>
<th>sm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>10x10</td>
<td>100</td>
<td>55.33</td>
<td>23.77</td>
<td>20.08</td>
<td>5</td>
</tr>
<tr>
<td>20x20</td>
<td>400</td>
<td>47.13</td>
<td>25.41</td>
<td>19.67</td>
<td>4</td>
</tr>
<tr>
<td>30x30</td>
<td>900</td>
<td>50.82</td>
<td>24.59</td>
<td>19.67</td>
<td>6</td>
</tr>
<tr>
<td>7x7x7</td>
<td>343</td>
<td>63.52</td>
<td>26.64</td>
<td>20.90</td>
<td>3</td>
</tr>
</tbody>
</table>

Speaker verification results for the same SOMs as in Table 1 are given in Table 2. The baseline EER for GMM of MFCC parameters is 8.49%. With an absolute difference of 6.6 - 11.1 per cent points, STCs perform about twice as poorly as MFCCs. In contrast to the speaker identification experiments, early fusion can improve speaker verification (with a 0.8 per cent point improvement for STCs derived from the 20x20 SOM coordinates). For late fusion, the EER is always lower than for MFCCs on their own. The 20x20 SOM leads to a 0.9 per cent absolute or a 10.2% relative error reduction.

Table 2: Verification percentage EER for STCs alone and in early and late fusion with MFCCs, for different SOM sizes and No. of dimensions

<table>
<thead>
<tr>
<th>Size</th>
<th>num cells</th>
<th>no fusion</th>
<th>early</th>
<th>late</th>
<th>sm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>10x10</td>
<td>100</td>
<td>19.63</td>
<td>8.99</td>
<td>8.15</td>
<td>6</td>
</tr>
<tr>
<td>20x20</td>
<td>400</td>
<td>15.13</td>
<td>7.72</td>
<td>7.62</td>
<td>4</td>
</tr>
<tr>
<td>30x30</td>
<td>900</td>
<td>16.50</td>
<td>8.23</td>
<td>8.07</td>
<td>6</td>
</tr>
<tr>
<td>7x7x7</td>
<td>343</td>
<td>18.00</td>
<td>9.20</td>
<td>8.10</td>
<td>3</td>
</tr>
</tbody>
</table>

8. Discussion

The results of our experiments have shown that STCs derived from the x-y coordinates of a SOM trajectory can enhance speaker recognition when combined with MFCC vectors. This is particularly true when late or score fusion is used to combine the speaker recognition results for the two parameter types. For speaker verification an improvement was also found for combination by early fusion.

A 2-dimensional SOM with 20x20 units gave best performance, both for speaker identification and for speaker verification. It should be noted that the SOMs were trained with an extremely small amount of data from each speaker, so that an increase in the amount of training data may lead to better results for a larger SOM which can represent the acoustic space in finer detail.

Higher-dimensional SOMs, of which results were only given for a 7x7x7 SOM, give consistently worse performance, despite a roughly similar number of units as a 20x20 SOM. Although 2-dimensional SOMs are usually used to visualize the acoustic space, there is no intrinsic reason, as discussed in Section ??, why this representation should be optimal to represent speech. In fact, it is highly unlikely that this is the case. The fact that a 3-d 7x7x7 SOM does not lead to better speaker recognition performance than a similar-sized 2-dimensional SOM does not reflect the intrinsic dimensionality of speech data, but may be related to
the rough categorization of the data into three
dimensions. Here, too, more data, allowing for an
increase in the number of units in the SOM, may lead
to better results for a 3-dimensional SOM. For
comparison, it would be interesting to reduce the
feature vector size by PCA to evaluate the optimum
length of the vector.

GMM modelling, as used in the experiment
presented in this paper, does not reflect the concept of
a trajectory in the modelling, since a GMM consists of
only one state and therefore does not reflect time
information. Nevertheless, some time information is
present in the input features to GMM, since the
Gaussian mixtures model the voice signatures, which
consist of x-y coordinates with additional velocity and
acceleration parameters as well as angle and curvature
information at each point. Such vectors gave better
results than using the x-y coordinates alone, so that we
can conclude that time information in the SVS
trajectories is useful. Despite large jumps in the SOM
space, as could be observed in Figure 3??, the
trajectories are smooth where there are smooth changes
in the acoustic space. This was demonstrated by
Figure 4??. This supports the usefulness of considering
the voice signature as a trajectory.

On the basis of these results, it may be possible to
improve the results further by using hidden Markov
modelling (HMM) instead of GMM. Standard left-to-
right HMMs can explicitly model time information
available in the trajectories by the transition
probabilities between states. The use of HMM may
therefore further improve speaker recognition,
although it was shown in (Morris et al. 2004) that
GMM can have equal performance to HMM.

The method presented here was used to generate a
complementary signal representation to standard
MFCCs. By using complementary information, we
attempt to counteract the effects of the small amount of
data available for modelling. The GMM models used,
however, still generalize across the data, as would
HMMs. Whether generalization is optimal across small
amounts of data is questionable. It is possible that an
exemplar-based approach, in which each trajectory of a
speaker’s voice signature is compared with a test
signal, leads to better results. Such a comparison can
be made using dynamic programming techniques.

Besides these results being positive, the SOM
trajectory should provide a representation which is
complementary to most other speech parameterisations
and can therefore be expected to continue to provide a
positive contribution, in both speech or speaker
recognition, when used in combination with most other
speech representations.

9. Conclusion

In this paper, a novel approach to speaker
recognition by using the speaker voice signature (SVS)
was investigated. The SVS is a vector obtained by a
parameterization of the x-y coordinates in a trained
SOM, similar to the parameterization of an on-line
signature’s x-y coordinates. By combining speaker
recognition scores from GMM of STCs with those for
GMM using MFCCs, speaker identification and
verification could be improved; in some cases, early
fusion by concatenating STCs with MFCCs before
GMM modelling also improved speaker verification.
The SVS therefore provides complementary information
to MFCCs which is useful for speaker recognition.
Best speaker recognition results were found for late
score fusion using STCs derived from the x-y
coordinates of the trajectories in a 20 x 20 SOM. This
SOM gave a relative error improvement over the state-
of-the-art MFCC baseline of 6% for identification and
10% for verification.

Although the improvements obtained are modest, it
should be noted that the SOM has not yet been fine-
tuned. Possible approaches to improve the speaker
recognition results using the SVS were also discussed.

Acknowledgments

This work was supported by the EC SecurePhone
project IST-2002-506883.

References

[1] Allano, L., Garcia-Salicetti, S., Ly-Van, B., Morris,
A.C., Koreman, J., Sellahewa, H., Jassim, S. &
Dorizzi, B., “Non intrusive multi-biometrics on a
mobile device: a comparison of fusion
techniques”, Proc. SPIE conference on Biometric
Techniques for Human Identification III, Orlando
(in press).

Speaker Recognition Corpus”, Proc. ICSLP,

a modular machine learning software library”,

J.G., Pallett, D.S., Dahlgren, N.L. and Zue, V.;
TIMIT Acoustic-Phonetic Continuous Speech
Corpus, 1993.

[5] Hagen, A. & Morris, A.C., “Recent advances in
the multi-stream HMM/ANN hybrid approach to


[17] Soong ref? (regression formula)
CSLU always divided data into train/dev/eval sets by dividing the 12 sessions for each speaker into 3 groups. UBM was trained using data in the train group across all speakers. Such a UBM would be ok for client model initialisation for MAP training. However, when used for score normalisation in verification tests, as a part of the UBM training data was always from the speaker being checked for authorisation, the UBM would not have worked so well for this purpose as it would have had the UBM speakers been separated.
Figure 3 compares the inter and intra speaker differences in the SOM trajectories for the utterance “two four”. Although intra-speaker differences (comparison between left and right figures) are quite large, they are smaller than inter-speaker differences (comparison between top and bottom figures).

The trajectories in Figure 3 appear to show a large number of sudden jumps in the speech trajectory. However, as shown in Fig.3, large sections of the utterance are quite smooth, and most jumps appear in low energy regions where the mapping onto SOM coordinates becomes arbitrary.

![Figure 3: SOM trajectories for utterance “two four”](image)

It is clearly shown (in Fig. 3.) that inter-speaker discrimination is much more than intra-speaker discrimination. The speaker discriminating information is illustrated not only by these trajectories, but also by the time duration for which the trajectory stays in a position. The bigger blob shown in each position implies that the trajectory stays there for a longer time. In fact, the distribution of the time duration can be efficiently modelled by a GMM with the two x-y coordinates of the SOM features. The global shape of each SOM trajectory can be captured by the other.

Fig. 4. Top: Trajectory from 2 repetitions of “two four” by speaker 0038. Bottom: 2 repetitions of the same prompt by speaker 0040. These trajectories were projected from their 38-d MFCCs by an SOM classifier trained on the training set of the CSLU database (c.f. the description in section 8).

![Figure 4: Trajectories for different speakers](image)
Fig. 3. Representation of a speech signal for “four” excised from CSLU file 0038aaq1.wav (oscillogram + spectrogram) together with x-y coordinates of the units in the SOM activated by each frame.