16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

NON-PARALLEL HIERARCHICAL TRAINING FOR VOICE CONVERSION

Larbi Mesbahi, Vincent Barreaud, Olivier Boeffard

IRISA / University of Rennes 1 - ENSSAT
6 rue de Kerampont, B.P. 80518, F-22305 Lannion Cedex
France
{lmesbahi,vincent.barreaud,olivier.boeffard}@irisa.fr
http://www.irisa.fr/cordial

ABSTRACT

Many research topics in speech processing face the same difficult problem, how to create cheaply (or quickly) a parallel corpus which associates the acoustic realizations of two speakers having pronounced the same linguistic content. Among those topics are voice conversion techniques and some aspects of speech and speaker recognition. In the context of voice conversion, we propose a new methodology to map the source speaker vectors with those of a target speaker, without any parallel corpus nor using DTW (Dynamic Time Warping). The proposed approach is based on a hierarchical decomposition of the source and target acoustic spaces. At each level, source and target class centroids of a reduced subspace are paired. We propose an evaluation of our algorithm when applied to GMM-based voice conversion on the ARCTIC database.

1. INTRODUCTION

Many fields of automatic speech processing need a large parallel speech corpora. However, to obtain hundreds of identical sentences pronounced by different speakers is often difficult and remains expensive. Nevertheless, voice conversion, automatic translation and speaker adaptation need such corpora. In these fields, it is necessary to map acoustic vectors of a target speaker with those, similar, of a source speaker. Such pairing is classically obtained by applying DTW (Dynamic Time Warping) [1] on a parallel corpus.

Many works are looking for vector mapping methods as an alternative to use the DTW on parallel data. These approaches can be classified into three categories. The first one considers cluster pairing methods. In [2], the vector mapping is obtained by doing the following steps : extraction of the acoustic parameters, automatic segmentation by a k-means algorithm, class pairing by using a minimal distance and finally mapping between each source vector and the centroid of the associated target class. This approach entails a global classification with little consideration of the shape of the two acoustic spaces. Moreover, the pairing of vectors on centroids reduces artificially the variance of resulting joint vectors. The second category of methods considers statistical models estimated on a parallel corpus that are adapted with nonparallel data. [3] proposes this methodology in the context of voice transformation. However, this technique still needs a parallel corpus and requires two computing phases before being able to apply a conversion function. The last category of solutions is based on unit selection methods used in voice synthesis techniques. In [4], the author applies this solution to the voice conversion problem. Source sentences similar to those pronounced by the target speaker are synthesized. Doing so, a paired corpus is created in a supervised way, by collecting the source units which correspond to the target units.

We propose a new non-parallel pairing method of two speakers as an alternative to the DTW. The proposed technique is called **svqTree**. It produces a tree decomposition using the split vector quantization **svq**. This technique falls into the cluster pairing category. Its innovation lies in a progressive and binary mapping of source and target acoustic spaces. Given a level of decomposition, the considered source and target subspaces split into two new subspaces, then each obtained source subspaces is paired with the closest target subspace considering an euclidean distance. The process is recursively applied on the two new pairs of source and target subspaces. On each level of splitting, the number of vectors in a cluster decreases. The mapping process stops when one of the two subspaces (source or target) comprises only two vectors.

This paper is organized as follows. Section 2 presents observations leading to the idea of hierachical clustering. Section 3 presents the methodology to build and use the **svqTree**. Section 4 introduces GMM-based voice conversion as an applicative framework to validate this new mapping technique. Section 5 gives an evaluation and a discussion of the mapping performances of the **svqTree**.

2. PRELIMINARY OBSERVATIONS

In cluster pairing, the mapping of vectors is done between vectors of paired clusters. So, we sought to evaluate the pairing of clusters when performing a simple (flat) svq-clustering. In a preliminary experiment, our objective was to see the overlapping between a source and target speaker spaces.To do so, we clustered the acoustic space considering MFCC vectors formed by the utterances of both a target and source speakers into 256 classes. We noticed that only 69% of these classes contained both source and target vectors. In other words, source and target acoustic spaces overlap on only 69% of classes. We concluded that, when source and target spaces are clustered separetly, about 31% classes would not be efficiently paired by minimum distance criterion.

In a second experiment, we sought to evaluate the minimum number of classes sufficient for hierarchical decomposition. Firstly, we clustered the source acoustical space in 4 classes and did the same on the target acoustical space. Then, we paired source and target clusters on a minimal distance criterion. Finally, we measured the number of vector couples paired by DTW that were actually present in this cluster mapping. The average classification ratio was of 64%. We concluded that a vector mapping induced by this cluster

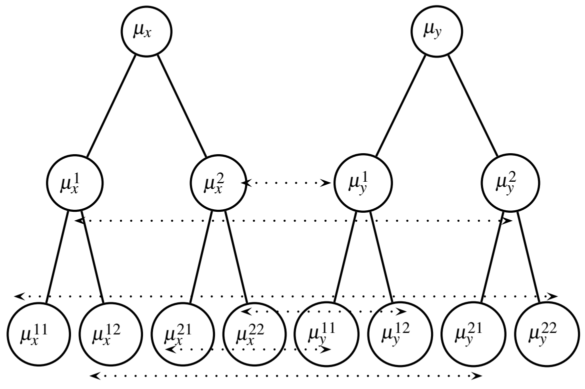

FIG. 1 – Joint hierarchical decomposition of source and target speaker acoustical subspaces for two levels. μ_x^i and μ_y^j are source and target cluster centroids. The pairing (dotted lines) is done among clusters decomposed from paired clusters.

mapping could never achieve great performances. Secondly, we performed the same type of parallel clustering but only with 2 classes. In our preliminary results, the resulting pairing gave an improved average classification ratio. It can be explained by the fact, that the cluster pairing based on minimal distance criterion is combinatorial and also that source and target speaker acoustical spaces do not overlap.

3. MAPPING BASED ON BINARY DECOMPOSITION

The proposed mapping technique, **svqTree**, is based on a hierarchical and progressive clustering method. Our aim is to reduce mapping errors by proposing to a source cluster a limited set (two) of possible target clusters to pair with. We are doing so by carrying out a joint and hierarchical partition on source and target acoustic spaces.

3.1 Hierarchical decomposition algorithm

Each node of the hierarchy corresponds to source and target subspaces. Each of those subspaces is divided into two new subspaces. Then one of the new source subspace is paired with the closest new target subspace considering an euclidean distance. Consequently, on each level of a partition, the search of the target cluster to pair with a source cluster is limited to a restricted set of clusters. This set of clusters is defined by the pairing performed at the upper level. Two parallel subspace trees are thus formed (one for the source, the other for the target). Each node of the source tree is paired with a node of the target tree of the same level (see Figure 1). This algorithm (see algorithm 1) is based on a recursive function which decomposes the acoustic space into a hierarchical structure. The recursion is stopped once we reach a minimum of 2 vectors per class.

3.2 Mapping function

We considered that each vector contained in a source cluster can be mapped with any vector of the associated target cluster. Several methods can be implemented to map a source vector from a cluster with a vector from the paired target cluster. We developed two of them. The first one (**svqTreeC**) maps the source vector with the centroid of the paired target cluster. This centroid may not be an existing target vector. In each k^{th} leaf of the **svqTree**, the mapping function associating the source vector x_i with the centroid μ_y^k of the k^{th} paired target cluster is :

$$\mathcal{M}_{svqTreeC}(x_i, k) = \mu_y^k \quad (1)$$

The second one, (**svqTreeT**), pairs a source vector with an existing target vector of the paired target cluster. This target vector is deduced by considering a distance between the acoustic vector and the centroid of this class. In each leaf of the **svqTree**, a target vector y_l is transformed into $\hat{y}_l^k = y_l + \mu_x^k - \mu_y^k$ where μ_x^k and μ_y^k are respectively the centroids of the source and target clusters of the k^{th} leaf. Then the euclidean distance is computed between x_i and each $\hat{y}_l^k$. The mapping function $\mathcal{M}_{svqTreeT}$ is then given by :

$$\mathcal{M}_{svqTreeT}(x_i, k) = y_j, \text{ such as } j = \arg\min_l ||x_i - \hat{y}_l^k||^2 \quad (2)$$

```
tree = SvqTree(X,Y,Threshold)
Input: X,Y,Threshold
Output: svqTree
if size(X) ≥ threshold and size(Y) ≥ threshold then
    [X_1,X_2,μ_x]=Classify(X);
    [Y_1,Y_2,μ_y]=Classify(Y);
    pairingStrategy=ClusterPairing(μ_x, μ_y);
    if pairingStrategy == 1 then
        tree =
        addNode(tree,SvqTree(X_1,Y_1,Threshold));
        tree =
        addNode(tree,SvqTree(X_2,Y_2,Threshold));
    end
    else
        tree =
        addNode(tree,SvqTree(X_1,Y_2,Threshold));
        tree =
        addNode(tree,SvqTree(X_2,Y_1,Threshold));
    end
end
else
    tree = addLeaf(tree,[μ_x μ_y]);
end
```

Algorithm 1: Algorithm for building the svqTree from the source (X) and target (Y) acoustical spaces, μ_x and μ_y are the resulting centroids respectively for the source and target spaces. *tree* is a data structure containing information about the binary tree. *tree* is initialized to an empty tree. *Classify* divides a subspace into two and compute the centroids of the new clusters. *ClusterPairing* pairs source and target clusters.

4. GMM-BASED VOICE CONVERSION

A GMM (Gaussian Mixture Model) based voice conversion system requires a mapped source and target speaker corpus. The quality of the mapping influences the models and, thus, the performances of the voice transformation. Indeed, two mapped vectors are supposed to represent similar segmental information. The difference between those two vectors holds the dissimilarity of the source and target voices.

In the following, we consider two sequences of N acoustical vectors. The sequence corresponding to the source speaker is represented by $X = \{x_1, \dots, x_N\}$ and the target speaker by $Y = \{y_1, \dots, y_N\}$. The sequence Y is obtained by mapping the sequence X using function $\mathscr{M}_.$ with : $y_i = \mathscr{M}_.(x_i), \forall i \in [1 \dots N]$. X and Y form a mapped corpus. When a parallel corpus is available, this mapping function is usually the DTW ($\mathscr{M}_{DTW}$). Considering an non-parallel corpus, we will use $\mathscr{M}_{svqTreeC}$ and $\mathscr{M}_{svqTreeT}$

A joint GMM is trained on the mapped corpus. Given the GMM-based partitioning of the acoustic space of the two speakers, we need to estimate a piecewise function $\mathscr{F}_.$ such that, $\forall n \in [1, \dots, N]$, $\mathscr{F}_.(x_n)$ will be close to y_n. Notice that $\mathscr{F}_.$ will depend on the mapping function $\mathscr{M}_.$ used to create the mapped corpus that trained the GMM. Once the GMM partitioning is done, the source/target conversion function can be derived as a weighted linear regression drawn from a conditional distribution of y_n with respect to x_n. This piecewise linear transform can be expressed as follows :

$$\mathscr{F}_.(x_n) = \sum_{m=1}^{M} P_m(x_n)[\nu_m + A_m(x_n - \kappa_m)] \qquad (3)$$

where ν_m (resp. κ_m) is the target (resp. source) part of the m-th Gaussian component's mean and A_m is the covariance product $\Sigma_{m,xy}.\Sigma_{m,xx}^{-1}$ ([5],[6]).

5. EXPERIMENTAL EVALUATION

5.1 Study of the svqTree

In this part of the study, we wanted to compare the mapping performances of **svqTree** and DTW. For each i^{th} sentence $X_i = \{x_{i,1} \dots x_{i,n_i}\}$ of length n_i (subset of X), we have measured the accumulated distance (AD) between the source and target mapped sequences :

$$AD_.(X_i) = \sum_{j=1}^{n_i} ||x_{i,j} - \mathscr{M}_.(x_{i,j})||^2 \qquad (4)$$

We successively used $\mathscr{M}_{DTW}$, $\mathscr{M}_{svqTreeC}$ and $\mathscr{M}_{svqTreeT}$ as a mapping function.

Experiments were carried out on a parallel English speaker corpus called *bdl-jmk*. This corpus was extracted from the ARCTIC corpus [7] and containes the utterances of *bdl* and *jmk* speakers. For evaluation purpose, we used 105 parallel sentences as a learning set, 24609 vectors per speaker, to construct the **svqTree**. The 90 parallel remaining sentences define a test set, 22677 vectors per speaker. The learning sentences were chosen at random. As the experiments were carried out 16 times, 16 learning sets have been built. The MFCC vectors have been computed with a 16Khz sampling frequency, a Hamming window of 30ms and a 10ms shift. For all the 16 experiments, the mean number of vector per leaf of the obtain **svqTree**s is 9201 and their mean depths is 14.

Table 1 presents the average accumulated distances for the 3 studied mapping functions. Firstly, we can observe that the accumulated distance produced by DTW is smaller than for **svqTreeC** and **svqTreeT**. This is not surprising since **svqTree** does not directly use the parallel information whereas DTW does. Moreover, the building of **svqTree** is totally unsupervised and do not requires any kind of adaptation. Secondly, Table 1 shows that **svqTreeT** gives a smaller (2%)

mapping method	learning set	test set
$AD_{svqTreeC}$	8236 ± 95	8170 ± 74
$AD_{svqTreeT}$	8053 ± 90	8009 ± 68
AD_{DTW}	7333 ± 86	7310 ± 62

TAB. 1 – Accumulated distance scores (over 16 repeated experiments) for the learning and test sets and for the three studied mapping methods. The 95% confidence intervals are given.

accumulated distance than **svqTreeC**, on the test set (the difference is statistically significant). This is due to the fact that the former pairs source vectors with existing target vectors whereas the latter maps source vectors with a mean target vector.

5.2 Application to voice conversion

In order to validate our mapping mechanism, we have tested it in the framework of a GMM-based voice conversion system. As explained in section 4, we used DTW and **svqTreeT** on the learning set to create two mapped corpora (resp. **corpus-svqTree** and **corpus-DTW**). Two sets of 2,4,8 and 16 components joint GMM have been learnt on these corpora. These GMM were then used to transform the test set with the method proposed by Kain [6]. Kain's conversion method have been chosen according to the study led in [8]. This paper compares the performances of the Kain method on DTW-mapped data with four other GMM-based conversion techniques.

Then, we scored the performance of the different conversion systems with the average distance between the target and the converted speaker (the conversion error) normalized by the distance between the source and the target (Normalized Cepstral Distance). The normalization of the conversion error should allow us to compare the DTW and the **svqTreeT** mapping represented in **corpus-DTW** and **corpus-svqTree**. Consequently, we have computed three types of errors : e_{DTW} is normalized with respect to **corpus-DTW**, $e_{svqTree}$ with respect to **corpus-svqTree** and e with respect with both copora. Indeed, using only the **corpus-DTW** normalization would have entailed a logical bias that would have favored the DTW mapping. The errors are computed as follow :

$$e_{DTW}(\mathscr{F}_.(X_i)) = \frac{\sum_{j=1}^{n_i} ||\mathscr{F}_.(x_{i,j}) - \mathscr{M}_.(x_{i,j})||^2}{\sum_{j=1}^{n_i} ||x_{i,j} - \mathscr{M}_{DTW}(x_{i,j})||^2} \qquad (5)$$

$$e_{svqTreeT}(\mathscr{F}_.(X_i)) = \frac{\sum_{j=1}^{n_i} ||\mathscr{F}_.(x_{i,j}) - \mathscr{M}_.(x_{i,j})||^2}{\sum_{j=1}^{n_i} ||x_{i,j} - \mathscr{M}_{svqTreeT}(x_{i,j})||^2} \qquad (6)$$

$$e\,(\mathscr{F}_.(X_i)) = \frac{\sum_{j=1}^{n_i} ||\mathscr{F}_.(x_{i,j}) - \mathscr{M}_.(x_{i,j})||^2}{\sum_{j=1}^{n_i} \sqrt{||x_{i,j} - \mathscr{M}_{DTW}(x_{i,j})||^2 ||x_{i,j} - \mathscr{M}_{svqTreeT}(x_{i,j})||^2}} \qquad (7)$$

where $\mathscr{F}_.$ can be either the conversion function based on a GMM derived from **corpus-DTW** ($\mathscr{F}_{DTW}$) or from **corpus-svqTree** ($\mathscr{F}_{svqTreeT}$).

Number of GMM components	2	4	8	16
$e_{DTW}(\mathscr{F}_{DTW})$	0.405	0.399	0.395	0.391
$e_{DTW}(\mathscr{F}_{svqTreeT})$	0.476	0.470	0.485	0.490

TAB. 2 – Conversion errors of a Kain voice conversion system, for 2, 4, 8 and 16 gaussian components when DTW ($\mathscr{F}_{DTW}$) and **svqTreeT** ($\mathscr{F}_{svqTreeT}$) conversion functions are used. The errors are normalized with DTW.

Number of GMM components	2	4	8	16
$e_{svqTreeT}(\mathscr{F}_{DTW})$	0.390	0.392	0.403	0.420
$e_{svqTreeT}(\mathscr{F}_{svqTreeT})$	0.373	0.349	0.317	0.295

TAB. 3 – Conversion errors of a Kain voice conversion system, for 2, 4, 8 and 16 gaussian components when DTW ($\mathscr{F}_{DTW}$) and **svqTreeT** ($\mathscr{F}_{svqTreeT}$) conversion functions are used. The errors are normalized with **svqTreeT**.

Tables 2, 3 and 4 show the conversion errors on a 90 sentences test set. The confidence intervals of these scores are about 10^{-3}, as seen in a previous study [8].

For **svqTreeT**, from Table 2 we noticed that, as the number of gaussian component increases, $e_{DTW}(\mathscr{F}_{svqTreeT})$ increases whereas $e_{DTW}(\mathscr{F}_{DTW})$ decreases. But, we observed the inverse from Table 3, as the number of gaussian component increases, $e_{svqTreeT}(\mathscr{F}_{svqTreeT})$ decreases whereas $e_{svqTreeT}(\mathscr{F}_{DTW})$ increases. This behavior justifies the fact that the normalization of the error introduces a bias. Table 4 according to Equation 7, enables us to compare scores normalized independantly from the transformation function. It confirms that conversion errors (for both **svqTreeT** and DTW transformation) classically decrease as the number of components increases and that the cumulated distance is larger in the case of **svqTreeT** transformation. Thus, we cannot conclude, from Table 3, that the **svqTreeT** mapping method is better than DTW mapping. We can only say that we obtain a mapping method that gives acceptable conversion scores whitout any parallel data.

In addition, from Table 2 we computed the relative distortion factor between $e_{DTW}(\mathscr{F}_{svqTreeT})$ and $e_{DTW}(\mathscr{F}_{DTW})$. It ranges from 17,5% (2 gaussian components) to 25,3% (16 gaussian components). This distortion factor is comparable to that of other alternatives to DTW for voice conversion proposed by [2] and [3]. We do not want to push this comparison any further since the experiments of these contributions were not conducted on the same corpus.

Number of GMM components	2	4	8	16
$e\ (\mathscr{F}_{DTW})$	0.354	0.348	0.343	0.348
$e\ (\mathscr{F}_{SVQTreeT})$	0.374	0.365	0.358	0.359

TAB. 4 – Conversion errors of a Kain voice conversion system, for 2, 4, 8 and 16 gaussian components when DTW ($\mathscr{F}_{DTW}$) and **svqTreeT** ($\mathscr{F}_{svqTreeT}$) conversion functions are used. The errors are normalized with both DTW and **svqTreeT** (confidence intervals are of .004).

On one hand, Table 1 showed that accumulated distance induced by **svqTreeT** was greater than that obtained with DTW. On the other hand, Tables 2 and 3 showed that this did not affect greatly a voice conversion system based on GMM learned on **corpus-svqTree**. So, in order to express how **svqTreeT** could map efficiently non-parallel data, we observed trajectories of a mapped source cepstral sequence and compared it with the target cepstral sequence for several sentences. These results are illustrated by Figures 2 and 3. Figure 2 represents the trajectory, on the first hundred frames of one sentence, of the first cepstral coefficient of the target sequence ("Original target") and the source sequence mapped with DTW ("DTW target"). Figure 3 represents the same target trajectory compared with the source sequence mapped with **svqTreeT** ("svqTreeT-target"). Those Figures show that the dynamic of mapped trajectory on Figure 3 is greater than that on Figure 2. This behaviour have been observed on all cepstral coefficients for all the studied sentences. As a result, the trajectory of the source sequence mapped by **svqTreeT** often diverges from the target trajectory. This behaviour is not observed with the source sequence mapped by DTW. This can be explained by the fact that DTW performs a smoothing of its mapping decision acording to the mapping of previous vectors. The smoothing can be done since DTW uses a parallel corpora. On the other hand **svqTreeT** maps locally every vectors and do not use parallel information. Consequently, it cannot perform this smoothing.

In order to observe the influence of these distinct behaviours regarding smoothing, we sought to represent the average values of the MFCCs parameters for the **svqTreeT**, original and DTW targets. Our purpose was to appreciate how close to the original target the **svqTreeT** and DTW targets were. Consequently, for each of the 90 test sentences, we have computed the Root Mean Square (RMS) of each MFCC dimension for the DTW, **svqTreeT** and original targets.
We defined the RMS score per dimension as follows :

$$RMS_d(space) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(MFCC_i^{(d)})^2}, \qquad (8)$$

where, the RMS_d represents the root mean square for the d^{th} MFCC parameter, *space* represents the DTW, svqTreeT or original targets and n is the size of *space*. The average $RMS_d(space)$ and 95% confidence intervals were then computed. Figure 4 represents these RMS for the thirteen MFCC parameters, respectively for the **svqTreeT** target, original and DTW targets. Firstly, we noted that the three RMS curves overlap. Secondly, their points are significantly joint except for the 1st (c_1), 5th and 10th MFCC dimensions. This experiment was performed only on one target speaker, and so can introduce a bias in these results.
The objective study on accumulated distance shown in subsection 5.1, foreshadowed that our approach did not give a good result that DTW, but we have seen that the distortion error were comparable to the conversion systems using DTW.

6. CONCLUSION

This study presents a new mapping method of speech corpora containing different speakers. Our aim, is to find an equivalent solution to DTW, without the help of a parallel corpora. The idea is based on hierarchical decomposition and

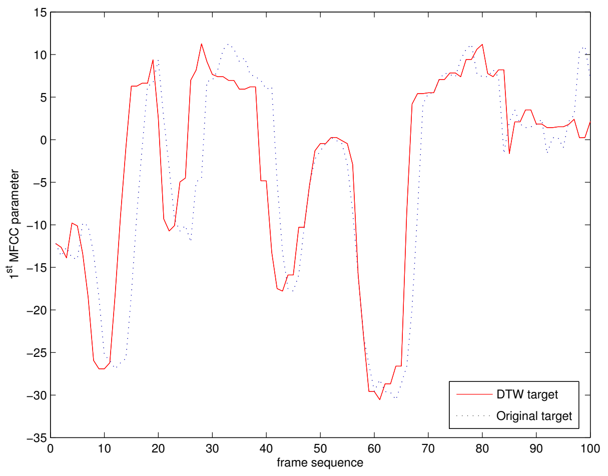

FIG. 2 – Trajectories of the first MFCC parameter of a target sentence (doted) and of the source sentence mapped with DTW (solid).

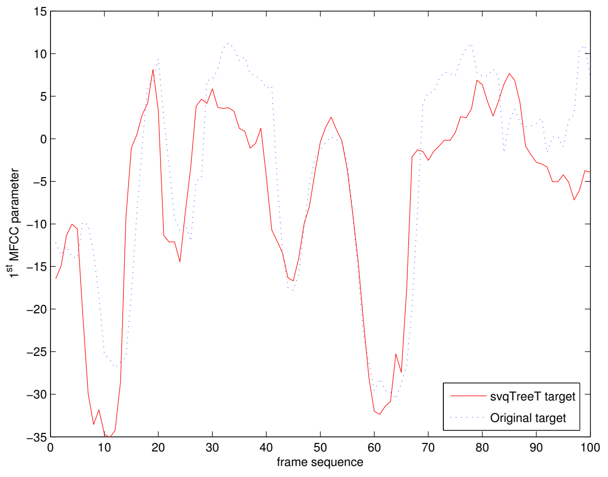

FIG. 3 – Trajectories of the first MFCC parameter of a target sentence (doted) and of the source sentence mapped with **svqTreeT** (solid).

progressive correspondence of acoustic spaces. On each level of the partition process, the search of a target acoustic cluster to pair with a source cluster is limited to a restricted set of clusters defined by the mapping of the upper level. Hence, two parallel cluster trees are formed (one for the source, the other for the target). As a result, the building **svqTree** is totally unsupervised and perfoms a mapping between two non-parallel corpora. Despite that our approach does not give an accumulated distance better than the DTW, a voice transformation system using our mapping obtained a conversion score comparable to those using DTW, even if the former did not use a parallel corpora.

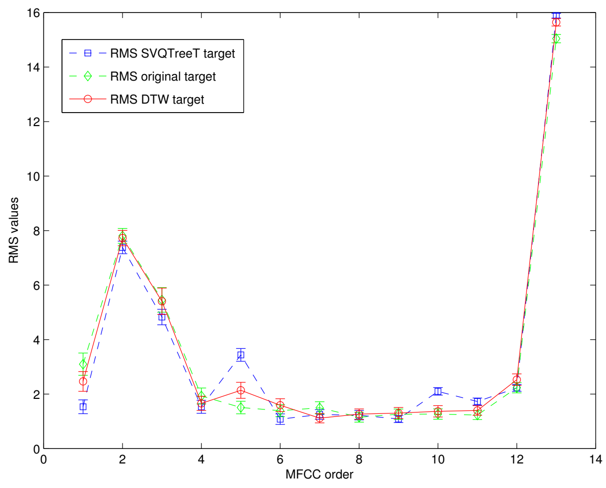

FIG. 4 – Representation of the average RMS for the thirteen MFCC parameters of the original target (diamond), DTW target (circle) and svqTreeT target (square). The 95% confidence interval was computed over 90 test utterances.

REFERENCES

[1] H. Sakoe and S. Chiba, “Dynamic Programming Algorithm Optimization for Spoken Word Recognition,” *IEEE Transactions on Acoustic, Speech, and Signal Processing*, vol. 6, pp. 43–49, Feb. 1978.

[2] H. Ney, D. Suendermann, A. Bonafonte and H. Hoege, “A First Step Towards Text-Independent Voice Conversion,” in *Proc. ICSLP 2004*,Jeju Island, Korea, October 4-8, 2004. pp. 1173–1176.

[3] A. Mouchtaris, J Van der Spiegel and P. Mueller, “Non-Parallel Training for Voice Conversion by Maximum Likelihood Constrained Adaptation,” in *Proc. ICASSP 2004*, Montreal, Canada, May 17-21. 2004, vol. 1, pp. I-1–I-4.

[4] H. Duxans, D. Erro, J. Pérez, F. Diego, A. Bonafonte and A. Moreno, “Voice Conversion of Non-aligned Data using Unit Selection,” in *Proc. TCSTAR 2006*, Barcelona, Spain, June 19-21. 2006, pp. 237–242.

[5] Y. Stylianou, O. Cappé and E. Moulines, “Continuous Probabilistic Transform for Voice Conversion,” *IEEE Transactions on Acoustic, Speech, and Signal Processing*, vol. 6, i. 2, pp. 131–142, Mar. 1998.

[6] A. Kain and M. W. Macon, “Spectral Voice Conversion for Text-To-Speech Synthesis,” in *Proc. ICASSP 1998*, Seatle, USA, May 12-15. 1998, vol. 1, pp. 285–288.

[7] J. Kominek and A. Black, “The CMU ARCTIC speech databases for speech synthesis research,” *Tech. Rep. CMU-LTI-03-177*, 2003.

[8] L. Mesbahi, V. Barreaud and O. Boeffard, “Comparing GMM-based speech transformation systems,” in *Proc. Interspeech 07*, Antwerp, Belgium, August 27-31. 2007, pp. 1989–1992.