Master thesis on Sound and Music Computing

Universitat Pompeu Fabra

Cross-Lingual Voice Conversion with Non-Parallel Data

Pablo Alonso Jiménez

Supervisor: Merlijn Blaauw

Co-Supervisor: Jordi Bonada

July 2017

Contents

Acknowledgement

I would like to dedicate this work to all the people that have supported me during this year. First, to the MTG for being a great source inspiration and knowledge. To my supervisors, Jordi and Merlijn for all the help and advise they provided my during this work. To my family and girlfriend for all the love I received from them in the distance. And, finally, to every SMC student just for being a group of amazing people both inside and outside the class.

Abstract

In this project a Phonetic Posteriorgram (PPG) based Voice Conversion system is implemented. The main goal is to perform and evaluate conversions of singing voice. The cross-gender and cross-lingual scenarios are considered. Additionally, the use of spectral envelope based MFCC and pseudo-singing dataset for ASR training are proposed in order to improve the performance of the system in the singing context.

Keywords: Voice Conversion; SI-ASR; Voice Synthesis

Chapter 1

Introduction

Voice Conversion (VC) is a machine learning technique that aims to change the personal characteristics of a voice message while preserving the linguistic component. In other words, given a source speaker, and a target speaker, the goal is to make an utterance from the source speaker sound as if it was produced by the target speaker.

VC have potential applications in a broad set of fields. In a world where Text-to-Speech (TTS) systems are more used every day, VC allows an efficient way of creating new voices from the already existing models. The field of interpreted telephony tries to analyze speech in order to perform re-synthesis to allow real time conversations among different languages. This technique could be improved if the voice identity of the speaker is not lost after the synthesis by means of VC [1]. On the film industry, VC could be used for film dubbing while preserving the original voice of the actor. In the professional music industry, it is common among multilingual singers to release songs in different languages. VC would allow to achieve similar results for monolingual singers with the help of a performer in the target language. The application of this technology to the entertainment and game industry is also very promising. Other potential applications include the security related usage or vocal pathology treatment.

Given the huge amount of fields to exploit, a big amount of contributions and different approaches to VC have been proposed in the last decades. However, there is

still a big headroom between the results of the State of the Art algorithms and what could be considered as natural sounding results [2]. Furthermore, another drawback of some of the best sounding VC approaches is the need of training for each specific set of source and target speakers, which restricts a lot the potential cases of use.

1.1 Motivation

A recent publication proposed the use of phonetic features in a new VC schema [3]. This system obtained good results and is attractive because it does not need parallel datasets and it is suitable for very diverse scenarios, like cross-lingual VC. Nevertheless, the robustness of the system is highly influenced by the quality of phonetic features, defined by the author as Phonetic Posteriorgrams (PPG). PPG can be obtained using a Speaker Independent Automatic Speech Recognition (SI-ASR) toolkit, but this system are normally developed for general speech. Thus, the behavior of the system in the singing voice scenario may be something uncertain.

According to this, a series of objectives are proposed in relation to this topic with the overall goal of testing and improving the method in the field of singing voice.

1.2 Objectives

1. Develop a PPG based Voice Conversion system.
2. Evaluate the system in the context of constrained singing voice.
3. Propose improvements in order to adapt the system to the case of singing voice.

1.3 Structure of the Report

The rest of this thesis is organized as follows. On the second chapter, a review of the most prominent approaches to VC is presented. This is completed with a revision on the work done in the areas of non-parallel, cross-lingual and singing VC. Elements that are of crucial importance to understand the scope and goals of this project.

On the third chapter the methodologies developed for accomplishing this thesis are presented. The results of the application of these methodologies with different configurations and in different scenarios are presented in the fourth chapter. Finally, these results are discussed in the fifth chapter.

Chapter 2

State of the Art

2.1 Literature Review

Defining the concept of voice identity may be a complex question as it depends in several parameters. The pitch contour, the spectral envelope, the speaking rate, the duration of the pauses or the prosody of the speaker are important features influencing on it. However, modeling all these features could be very complex and not all of them are crucial for the speaker identification [4]. In the other hand, there is strong evidence that distinct speakers can be efficiently discriminated by comparing their spectral envelopes [5]. Thus, the biggest part of the VC approaches have been focused in the transformation of the spectral envelope [2]. As the scope of this project is also devoted to the spectral envelope transformation, only this sort of methods are reviewed in this chapter.

This section offers an explanation of the most successful approaches on the literature: Codebook Mapping and Mixture Gaussian Models. After this, some alternatives are presented.

2.1.1 Codebook Mapping

The simplest way to create a VC oriented codebook, would be to use parallel utterances from the target and the source speakers. If the sentences are properly aligned,

then it is easy to create a mapping codebook as a combination of the source speaker frame-wise feature vectors.

In the approach proposed by Abe [6], Vector-Quantization is applied to the spectral envelope of the audio frames in order to create codebooks of the source and target speakers. The fundamental frequency (f0) and the power of each frame are computed and also quantized as scalar codebooks. Dynamic Time Warping (DTW) is used to obtain the frame-level match between the source and target frames. For each source entry, the correspondences to the target codebook entries are accumulated in form of histograms.

In order to synthesize new utterances, the frames are analyzed to obtain the correspondent entries on the source codebook. Each converted frame is generated as a linear combination of the elements of the target codebook using the histogram associated with the current entry as a weighting function. f0 and the power are directly taken from the element with the higher number of repetitions in the histogram. This algorithm provides an intuitive and interesting framework. However, the author reported that just 65% of the converted utterances were identified as belonging to the target speaker.

Some alternatives have been proposed in order to improve the behavior of this method. As the application of an hierarchical codebook architecture [7]. In this technique, an additional stage is introduced for the model training. Once the conventional codebooks are trained, some conversions are performed and a new codebook is trained to map the source spectral envelopes to the residual of the converted envelopes. Then, this residual can be added to the synthesis schema. This technique adds some quality improvements to the converted spectral envelope.

In general, the main advantage of codebook approaches is that the synthesis rely directly on combinations of the original source features. Thus, the identity of the target is preserved in the synthesized spectral envelope frames. However, this method has limitations in order to produce high quality conversions due to the artifacts introduced by the abrupt discontinuities produced in the time domain. This hap-

pens because the entries of the codebook are chosen relying exclusively in a feature similarity criteria, without considering the temporal context.

2.1.2 Gaussian Mixture Models

Another famous set of approaches work by obtaining a mapping function between the source and target spectral features. Once this function is available, conversion is done by transforming the statistical properties of the source in order to fit the characteristics of the target. Among them, the based on Gaussian Mixture Model (GMM) are probably the most successful and researched in the literature.

The GMM approach assumes that the probability distribution of the acoustic features can be expressed as a combination of Gaussian distributions. Assuming that these features are p-dimensional vectors, as for instance the famous Mel frequency cepstral coefficients (MFCC), these Gaussian are then multivariate distributions that can be expressed as,

$$p(x) = \sum_{c=1}^{M} \alpha_c N(x; \mu_c, \Sigma_c) \tag{2.1}$$

where N denotes a Gaussian characterized by the mean vector of size p, μ and the covariance matrix of size p x p, Σ, and M is the number of distributions.

First, the training data is used to obtain the parameters of the GMM, typically through the Expectation-Maximization (EM) iterative approach. EM tries to adapt the parameters of each Gaussian $(\alpha_c, \mu_c, \Sigma_c)$ in order to maximize the log-likelihood, which can be understood as maximizing the capability of the model to explain the training data. This is achieved by computing the posterior probabilities of each Gaussian given each training sample. By applying the Bayes rule,

$$P(c|x) = \frac{\alpha_c N(x; \mu_c, \Sigma_c)}{\sum_{j=1}^{M} \alpha_j N(x; \mu_j, \Sigma_j)} \tag{2.2}$$

$P(c|x)$ is sometimes refereed as the responsibility of the Gaussian c to explain the

sample x. The Maximization step of the algorithm shifts α_c, μ_c, Σ_c towards fitting the samples of which it is more responsible.

Once the GMM is fit to all the training data, it can be used to generate a mapping function between the source and the target speakers. This is commonly done by one of the two approaches explained in the next lines.

On the approach proposed by Stylianou et al. [1], parallel data of the source and target is required. MFCC are used as a representation of the spectral acoustic features with a reduced dimensionality. The EM training stage is done using just the MFCC of the source speaker.

As said above, the goal of the second part of the algorithm is to compute a conversion function. To do this, a parametric transformation function with the following form is proposed,

$$F(x_t) = \sum_{c=1}^{M} P(c|x)[v_c + \Gamma_c \Sigma_c^{-1}(x_t - \mu_c)] \tag{2.3}$$

where the parameters v_c and Γ_c are a mean vector and a covariance matrix. This means that the converted MFCC are obtained as a combination of different mean and variance transformations of the source frames, where each transformation is weighted by its posterior probabilities. v_c and Γ_c are found applying least squares optimization (LSO) in order to minimize the distance between the converted MFCC, $F(x_t)$, and its parallel MFCC directly extracted from the target.

In the approach proposed by Kain & Macon [8], GMM is trained to fit the joint density vector of the source and the speaker features $z_t = [x_t^T, y_t^T]^T$. This is also known as JD-GMM. In order to generate the density vector, it is also necessary to have parallel aligned data. EM is applied as explained before to fit the joint data. Given the shape of the joint data, the resulting mean a covariance matrix have the following form,

$$\mu_c = \begin{bmatrix} \mu_c^x \\ \mu_c^y \end{bmatrix}, \qquad \Sigma_c = \begin{bmatrix} \Sigma_c^{xx} & \Sigma_c^{xy} \\ \Sigma_c^{yx} & \Sigma_c^{yy} \end{bmatrix} \tag{2.4}$$

This approach has the advantage of creating a model that contains relevant informa-

tion for both, the source and the target. Furthermore, it is not necessary to use an optimization technique in order to obtain the conversion function, as the parameters for the transformations can be directly estimated from the joint data. The piecewise mapping function for this approach can be expressed as follows,

$$F(x_t) = \sum_{c=1}^{M} P(c|x)[\mu_c^y + \Sigma_c^{yx}(\Sigma_c^{xx})^{-1}(x_t - \mu_c^x)] \tag{2.5}$$

where super-indexes x, y, xx and xy indicates the submatrix from eq.2.4.

GMM based approaches have been widely used showing state of the art results. However some authors have pointed to some important disadvantages. GMM relies on the use of covariance matrices. If the method uses a full-covariance matrix, the number of parameters is in the order of the number of Gaussian by dimension of the acoustic features squared, which can be computationally expensive. Furthermore, if a low amount of training data is used, the model can overfit the training data. This problem can be tackled by using a diagonal covariance matrix instead. However, this means that each component of the feature vector is mapped in a one-by-one sense instead of considering the interdependencies with the rest of components.

Helander et al. [9] also found that using Least Squares Optimization can lead to overfitting in the conversion function of the Stylianou approach. They tacked this problem by using Partial Least Squares (PLS) for the regression.

In the other hand, oversmoothing is another problem that can diminish the quality of the converted voice. In the frequency domain, this problem is represented as a loss on the capacity to fit the spectral details. Toda et al. [10] found that using warped spectra contributed to reduce the oversmoothing effect. Chen et al. [11] pointed the time domain oversmoothing as a source of synthesis quality loss. The author points to the covariance adaptation product $\Sigma^{yx}(\Sigma^{xx})^{-1}$, present in the JD-GMM conversion function, eq.2.5, to be close to zero on most of the cases. As it can be easily seen, this fact lets the target as only dependent on a weighted sum of mean vectors, which could explain the small time fluctuations on the converted data. To avoid this problem, the author propose the use of a greater training dataset combined

with the elimination of the covariance adaptation term from the conversion function.

A last problem attached to GMM methods is the lack of temporal dependency modeling, as the conversion is performed independently for each frame. It has been shown that the long term context contains important information in order to obtain smooth spectral transitions. Furthermore, Helander found that sometimes there is one Gaussian component dominating on each frame [9]. This means that, when this happens, the GMM behaves pretty much like a hard clustering method, which in the end makes it susceptible to incur into frame-by-frame discontinuities such as in the codebook mapping case. In order to smooth this time-independence problem, Toda et al. [12] proposed the use of Maximum-Likelihood Parameter Generation (MLPG). This method can be used as a post-processing step to the JD-GMM conversion in order to reduce the artifacts of the frame-wise analysis. In the training step, the first and second order derivatives are added to the joint density vector. Then, the method makes an estimation of the spectral parameters trajectory.

2.1.3 Artificial Neural Networks

Artificial Neural Networks are other famous approach in order to generate a spectral mapping function. In that sense, their goal is very similar to the goal of the GMM. However, while the former obtain a non-linear mapping function by adding the posterior-probability-weighted set of linear mappings, the last one directly rely on non-linear functions to model the target. These functions form what is commonly know as artificial neurons, and are stacked in several layers. A basic network could be composed by an input a middle (or hidden) and and output layer. The number of neurons in the input and output layers have to fit shape of input and output acoustic features. Depending on the experiment, the shape of the hidden layer can vary. But it is sometimes related with the size of the dataset used for the training process.

As a basic example, the equation defining a neuron can be written as follows,

$$y = f(Wx + b) \tag{2.6}$$

where f, W and b represent the activation function, the weight and the bias for a input x. In the training process, W and b are optimized according to certain objective function.

Neredranath et al. [13] used ANN in order to map the formants of the source speaker to the formants of the target. The obtained features could be used to feed a formant synthesizer. Desai et al. [14] [15] used a similar approach, but using the MFCC instead, which can be adapted to feed most of the modern vocoder based speech synthesizers.

Chen [16] used Restricted Boltzmann Machines (RBM) instead of GMM in order to model the Spectral joint density. RBM showed a bigger flexibility than GMM in order to capture the inter-speaker and inter-dimensional dependencies, improving the quality of the converted voice in terms naturalness.

Following the recent impact of Deep Neural Networks (DNN), some authors have explored different configurations. Typically, an ANN is considered a DNN when it has more than 1 hidden layer. This kind of architectures are intended to provide more flexibility in order to model complex non-linear dependencies.

Chen [17] proposed the use of DNN to construct the mapping relationship between the spectral envelopes of source and target speakers. He proposed a four-layer architecture trained layer-by-layer from a cascade of a Bernoulli bidirectional associative memory (BBAM) and two RBMs. In the approach by Nakashka et at. [18], they used RBMs based network in order to learn a high order source and target speaker eigen-spaces from features on the cepstral domain. Then another ANN was trained to learn a map between these spaces.

Nevertheless, these algorithms still have two problems in common with the GMM and the codebook approaches. First, they are frame-based methods unable to con-

sider the temporal dependency in successive instances. On the other hand, despite standard Recurrent Neural Networks (RNN) are able to take the time context into account, they have limited capability to model it. Furthermore, they have access to the past values, but not to the future ones. And it is difficult for them to learn about long-range context dependencies.

To overcome this problems, Sun proposed Bidirectional Long Short-Term Memory RNN (BLSTM-RNN) [19]. This architecture has been probed to outperform regular RNN in several applications evolving sequence modeling. This is because of its ability to store an optimal amount of context information over a long period of time thanks to its capability to establish recurrent connections both, forward and backward. Furthermore, while the standard RNN normally use the sigmoid or the hyperbolic tangent as activation function, the LSTM networks are based in units called memory blocks that are reported to learn long range context dependencies. This memory blocks are designed to have input, output and forget gates, which has been proved to have a determinant weight in problems involving continuous or very long input vectors [20].

2.2 Non Parallel Voice Conversion

One of the main drawbacks of the traditional VC methods is their dependency on parallel data. This supposes a problem in practical applications, as the system has to be trained again every time it is desired to add a new speaker into the system. Furthermore, this implies the creation of an specific dataset, and obtaining parallel recordings is not always an easy task.

In order to tackle this, some authors have develop new algorithms, or adaptations for the existing ones in order to get rid of such dependency. Sundermann et al. [21] relied on the fact that GMM approach is time independent to perform a frame-wise feature clustering of the non-parallel source and target signals. They used the spectral centroid of the frames to feed a K-Means classifier. However, the reported accuracy was 25% below the results with parallel data. Ye [22] used an AS) system

based on Hidden Markov Models (HMM) to retrieve the phonetic clusters improving this result.

2.3 Cross-Lingual Voice Conversion

Cross-lingual VC research is a particular area inside the non parallel methods that consider the case where the input and source speakers have different languages. The principal point of this set of methods is that the source and the target are not constrained to have the same set of phonemes anymore.

INCA [23] approach tackled the monolingual problem by means of a recursive algorithm. It looks up for similar frames among the source and the target and performs a GMM based conversion. This process is then repeated again obtaining an output that sounds closer to the target on each iteration.

Recently, Xie [24] proposed a DNN-based approach attempting to equalize the speaker differences between different languages and using Kullback-Leibler divergence to measure the phonetic distortion. However, methods not relaying in parallel data are still far from the quality obtained by the parallel ones.

Sun used an SI-ASR model in order to extract a phonetic description of the audio frame in terms of PPG [3]. For a given audio frame, PPG represent the likelihood of belonging to sub-phonetic states called senones. Each phoneme is typically divided in 3 or 4 senones. The process of training the SI-ASR has a critical impact on the quality of this system. However, it is a wide topic and, hence, it would be explained apart in the next chapter. For the scope of VC, what is interesting from the PPG is their capacity to provide a phonetic, and thus, speaker independent, representation of the acoustic features. As any potential source speaker would ideally produce the same PPG, once this feature is mapped to the acoustic features of the target speaker (MCEP), the system does not have to be trained again for new source speakers. This is sometimes called, many-to-one VC. In order to model the relation between the PPG and the MCEP features, a DBLSTM recurrent neural network is trained through several utterances of the target speaker. When the system is trained, it is

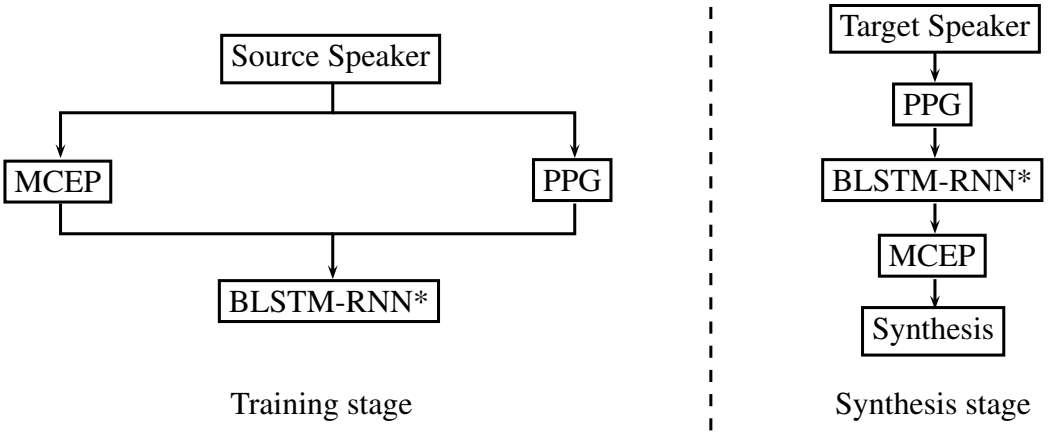

Figure 1: Flow diagram of the Sun's algorithm. * represents that the blocks are the same model.

possible to extract the PPG of an arbitrary source speaker and use this model to retrieve an estimation of the MCEP of the target speaker for the input utterance. In the synthesis stage, the fundamental frequency of the input utterance is combined with the converted MCEP to feed a vocoder. This steps are represented in the fig.1.

What is interesting from this approach, is that it supposes a significant improvement of quality compared to the rest of algorithms not relying in parallel data. However, one of the main drawbacks is the long times required for the training of the SI-ASR and the RNN.

2.4 Singing Voice Conversion

The scope if this thesis is related with VC algorithms applied to singing voice. this section shows a small review of the related work. Up to this point, all the exposed techniques are intended for general purpose speech. However, it is worthy to take a look to the main approaches of VC that have been evaluated for singing.

Tuk [25] used weighted codebook mapping to achieve cross-lingual VC and applied it to rap songs. The system used parallel utterances in English to train the model. Then, a source speaker with the ability to rap in two languages could convert his voice into the target speaker's. Despite the curious of the experiment, the dependency on a bilingual singer makes it unsuitable for real scenarios.

Villavicencio & Bonada applied the JD-GMM approach to high-quality sining datasets in the context of concatenative singing-voice synthesis [26]. In their study they focused on assessing the impact of the pitch in cross-gender VC. They found that conversion towards lower pitched voices resulted in lower quality than towards higher pitched voices.

Kawakami [27] also applied the GMM approach to sining VC. In this approach he used the vocal tract area function to model the voice.Toda [28] proposed a GMM based method called Eigen Voice GMM (EV-GMM) . In [29] a framework to generate new utterances from the source singer made the system able to operate as a non parallel data dependent method. But the quality is still far from the parallel methods.

2.5 Conclusion

Following this review, the method proposed in this thesis would be placed among the non-parallel, cross-lingual methods featuring many-to-one VC. And it will be evaluated in the context of singing voice. In regards to the promising results recently obtained by Sun [3], the use of PPG is explored. Nevertheless, due to the additional complexity of the singing voice, the are several questions that should be answered along this project.

On the chosen method, the final conversion quality depends on the robustness of the PPG and the capability of the DNN to map them the speaker spectral envelope. Thus, these elements have to be evaluated on the scope of singing VC.

The first point to tackle with is the behavior of the ASR system when the input is singing voice. As the strong pitch variance or the modulation done by some singers can hinder the recognition process. SI-ASR are normally designed for general speech an thus, the behavior on singing has to be evaluated and adapted. Moreover, [30] probed that PPG are sufficient to model the speaker identity in the context of speech. But in most of the cases, singing voice requires a high level of expressibility. The capability of the PPG to represent this expressibility has to be assessed.

Chapter 3

Methods

The model proposed in this thesis relies in a ASR system to obtain a phonetic description of voice (PPG). Then, a DNN is trained to map these PPG into the features related to the target speaker. This process can be divided in two training stages that are developed in the following sections.

3.1 Learning the Phonetic Model

The goal of this stage is to retrieve phonetic information (PPG) out of the input acoustic features. This is a key point of the project because here is where the speaker dependent features become independent (i.e., for two parallel recordings, the waveform contains information about the speaker voice, but the phonetic transcription does not).

This section contains an informal description of the ASR framework used to obtain the PPG followed by a detailed explanation of the datasets, features, transformations and steps of the process.

3.1.1 ASR with weighted finite-state transducers

The goal of an ASR system is to transform a waveform into words. In order to model the complexity of the speech recognition task, it is normally divided in 4 layers: the

word-level grammar, the pronunciation lexicon, the context dependency transducer and the HMM transducer. An example of this representation is available on fig.2

The world-level grammar can be created by rules or learned from exiting data. It is composed of by strings of valid combination of words. However, this layer is not explained in detail as it is not relevant for the scope of the project.

One step lower there is the pronunciation model. For each world, there can be more than one valid pronunciation. The units of this model are the phonemes and the output are words. One of the most relevant problems in this layer is the homophone indetermination. As, if two words are pronounced the same, it is not trivial to retrieve the word. This layer is also not relevant.

A third layer of abstraction would be in charge of mapping from the phonemes with context to the raw phonemes. It has been shown that it is more informative to use phonetic context information (which are the previous and next phonemes) than the raw phonemes. The reason of doing this is because the transition between two phones may be more informative than their stationary parts, so considering this contextual information is useful. A triphone is a phoneme with the knowledge of the previous and next phoneme. Thus, given a dictionary of p phonemes, the amount of triphonemes is $O(p^3)$.

The lowest layer represents the HMM. Each phonetic unit is modeled with a HMM, typically following the 3-state Bakis model. Nevertheless, more complex phonemes can use more states. These steps represent the beginning, middle and final part of the phoneme. The emission of the each state is a Probability Density Function (PDF) that is trained to fit the acoustic features. If the phonetic unit is context dependent, for instance a triphone, the number of required HMM increases by the order of the number of phonemes by the power of 3 and so does the amount of data needed for the training. On these cases some states among different triphones can be grouped to reduce the number of PDFs to estimate. It is known that simple Gaussian distributions are not enough to fit the complexity of the spectral features described above. Thus, it is typical to use GMM for this task.

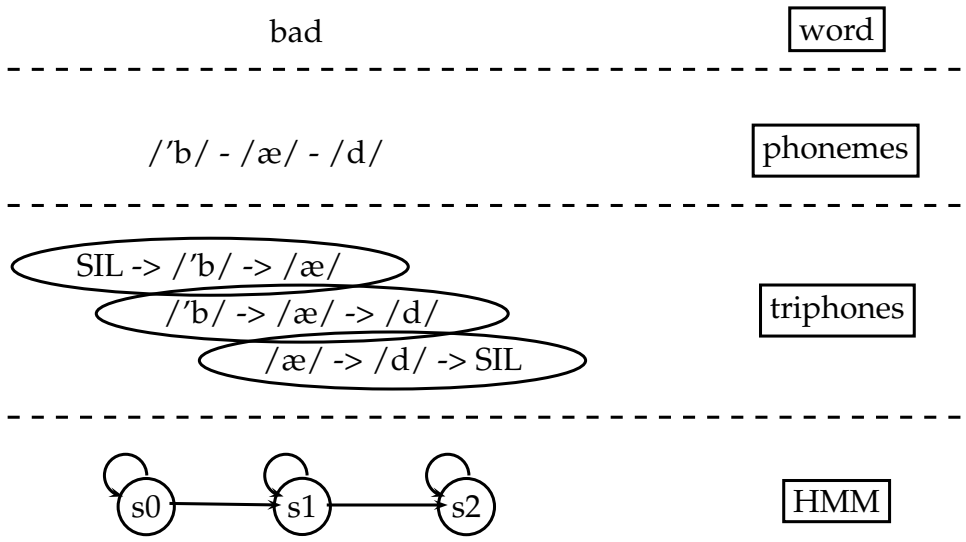

Figure 2: The world "bad" represented on the different layers. From the top to thew bottom: written language, phonetic description, triphonic description and as a HMM.

This stack of layers, allows to map from acoustic features to words. Weighted finite-state transducers (WFST) offers a common algorithmic framework useful to perform this task. WFST can be informally defined as an entity that recognize each string that can be read along a path from a start to a final state with an attached input label, output label and weight. A toy-example of a WFST can be found in fig.3. This architecture can represent a relationship between different levels of abstraction, which makes it very suitable for the ASR layers exposed above. This can be easily done using the operation of composition. Composition is an operation that allows to merge WFST models representing each of the ASR layers in just one model. To understand the theory behind this idea, the reader can find a mathematically well-defined explanation on this paper by Mohri [31].

In order to train the model, the Viterbi algorithm is used to find the best state sequence for each training utterance. The number of feature vectors corresponding to each state and the number of transitions between states are counted. The transition probabilities between states are computed as the normalized ratio between the transitions produced from the source state to the target state during the training stage over the total number of transitions produced from the source state. The PDF of each state is re-estimated using all the feature vectors that were clustered on that

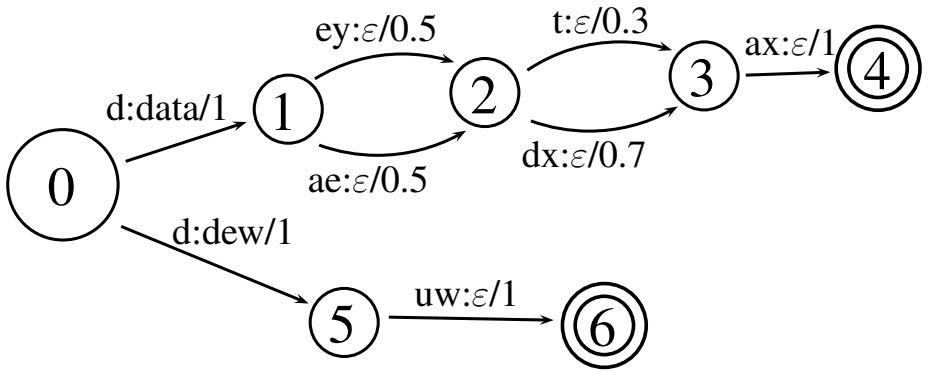

Figure 3: Weighted finite-state transdice example. The input label i, output label o, and weight t of a transition are marked on the corresponsing arc as $i{:}o/w$, following the notation of [31].

particular state during the training. If the PDFs are computed from GMM, the Expectation-Maximization algorithm can be used for this task.

3.1.2 Datasets

In order to train the phonetic model the TIMIT[1] corpus was initially used. This corpus was created to perform ASR research. TIMIT contains broadband recordings of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences [32]. The dataset is composed of the audio recordings with the corresponding textual and phonetic annotations.

In addition the Zxx[2] dataset was considered. Zxx is a dataset composed by annotated constrained singing utterances by 11 different singers. Constrained singing is understood as singing where the cadence and intonation remains constant and the singers are asked to avoid any voice technique such as vibrato. There is about 30 minutes of recordings of each artist.

While the TIMIT dataset was designed to provide a wide coverage over different American dialects, the goal of Zxx is to cover all the possible phonetic combinations for a small number of singers. This makes Zxx more suitable to the research in voice synthesis rather than ASR. Nevertheless, as the target of this project is the

[1] Corpus design was a joint effort among the Massachusetts Institute of Technology (MIT), SRI International (SRI) and Texas Instruments, Inc. (TI).

[2] Zxx is a proprietary dataset and at this time it is not commercially available.

conversion of singing voice, it is interesting to assess the behavior of the phonetic recognition step when the ASR is trained with data potentially closer to the project target instead of usual speech data.

3.1.3 Acoustic Features

The acoustic features retrieved from the datasets are MFCC, well know for their capability to capture timbre information. Cepstral Mean Variance Normalization (CMVN) is applied for each speaker. The goal is to eliminate the speaker dependent information (vocal tract characteristics, mean pitch...) in order to reduce the bias towards the training speaker particularities.

As the acoustic descriptor is supposed to be just informative about the timbre of the different phonemes, it should be independent of the pitch of the speaker. However, in case of high pitched singing voice this requirement can be not reached. In case that the pitch is sufficiently high, the filter-bank can start tracking the harmonics of the voice. In order to prevent this to happen, the CheapTrick algorithm [33] was used to estimate the spectral envelope of the input before computing the filter-bank for the MFCC. The different filter-banks produced by the FFT and the SP approaches can be seen in fig.4.

3.1.4 HMM-Models training

This subsection describes how the ASR system is built from the input acoustic features. The process is based on a the TIMIT recipe contained in the Kaldi[34] framework. This script was originally designed to be trained with the TIMIT dataset featuring default MFCC. For the scope of this project, new scripts were crated based on the Zxx corpus and using Spectral Envelope based MFCC[3]. Now the steps of the ASR building process are explained.

From a practical point of view, it is better to start building the ASR model with a simple monophonic HMM model and iterate adding complexity to the features and

[3] `https://github.com/pabloEntropia/kaldi/tree/master/egs/timit/`

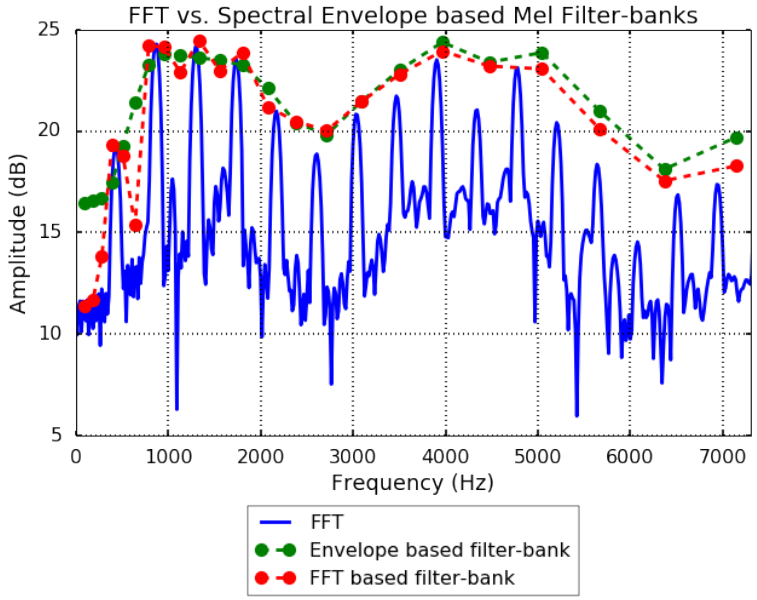

Figure 4: Example of FFT and SP based filter-bank for a vocal frame of a female singer. It can be seen how the FFT filter-bank tracks the lower harmonics instead of follow the envelope.

HMM model. On the first iteration, HMM represent monophones, i.e., phonemes without context. In this architecture, each phoneme is represented as a HMM. Typically, each phoneme is modeled with 3 or 4 states. The weights for the monophone model serve a a basis to train a more complex models using triphones. Thus, in this phase the number of HMM is tripled. As explained before, some states along phonemes are clustered to reduce the global number of GMM for computational reasons. In this step the first and second order derivatives of the MFCC are added to the feature vector.

After this, a new triphone model is trained after modifying the feature vector. In order to add more contextual information at the feature-vector level, for each frame, 3 past and 3 future frames are stacked. Provided that 13 MFCC coefficients are used in the analysis this leads to a 91 dimensions feature vector. Linear Discriminant Analysis (LDA) is used to reduce this dimensionality to 40 components.

Following this, Maximum Likelihood Linear Transform (MLLT) is applied [35]. MLLT is a square feature-transformation matrix. Its objective function is the average per-frame log-likelihood of the transformed features given the model, plus the

log determinant of the transform. The means of the model are also rotated by transform in the update phase.

The last iteration is called Speaker Adapted Training (SAT). This speaker-wise adaptation is achieved through feature-space Maximum Likelihood Linear Regression (fMLLR)[36]. Here, a transformation matrix is trained to maximize the likelihood of a set of data given the previous HMM set. This transform has been proved to be useful for environment compensation and speaker adaptation. The Maximum Likelihood transform is detailedly explained on [37].

3.1.5 DNN

Finally, a DNN is used to map the acoustic features to the PDF ids. The network is built on top of the fMLLR features obtained for the last model.

After computing the features, a pre-training stage is performed according to Geoff Hinton's tutorial paper [38]. The training algorithm is Contrastive Divergence with 1-step of Markov Chain Monte Carlo sampling. The first RBM has Gaussian-Bernoulli units, and following RBMs have Bernoulli-Bernoulli units. The training is unsupervised, so it is sufficient to provide single data-directory with input features.

Finally, a DNN classifies frames into triphone-state emissions, i.e., PDF ids. This is done by mini-batch Stochastic Gradient Descent. The DNN uses sigmoid hidden units, softmax output units and fully connected layers. A good explanation of the DNN can be found on [38].

However, for the scope of this project, instead of using the output of the net, which is a discrete sequence of the most probable PDF ids per frame, the weight vector of all the PDF ids is used. This is what was previously presented as PPG. The goal of doing this is to have a smoother representation of the phonetic information. This way, transition between phonemes are represented as a gradual change on the predominant PPG probability instead of an abrupt change that could lead into unnatural sounding transitions.

After the transition from acoustic features to phonetic ones, the typical ASR schema should implement a dictionary, to move from phonetic units to words, probably using a language model. These models are intended to reduce the range of possible words to look up for depending on the context. However, we are not going to focus on this point as it has no relevance in the scope of this project.

3.2 Learning the Voice Model

This section shows how a DNN driven by the PPG is used to synthesize the acoustic parameters of the target speaker. However, instead of the DB-SLTM RNN proposed by Sun, the Neural Parametric Singing Synthesizer (NPSS) [39] was used in this project.

3.2.1 A Neural Parametric Singing Synthesizer

NPSS consist in an neural network based system that generates probability distributions of the target features based on the past values and a phonetic control. These features are specific for the target singer and the control phonemes, and they can be combined with a melody (or FO) extracted from a real singing performance or just created with a MIDI device in order to synthesize realistic singing.

This model is manly inspired on a class of fully-visible probabilistic autoregressive generative models that use neural networks with similar architectures. This architecture has been used to generate images (Pixel CNN) [40], audio waveforms (WaveNet) [41] or text (ByteNet) [42]. In this case, instead of modeling the raw waveform, the network is used to predict the input of a parametric vocoder, which are Spectral Envelop (SP) and Aperiodicity (AP). While Wavenet's output is unidimensional, i.e., a mono audio signal, the target features are multidimensional. Mel Frequency Spectral Coefficients (MFSC) are a 2D (time and frequency) representation of the SP with a lower dimensionality. Band Aperiodicity (BAP), a compact version of AP, is also in the same 2D domain. Thus, the target of the network can be a multivariate conditional distribution with diagonal covariance. However, as the dimension belong to different domains, the use of 2D convolutions is not suitable.

Thus, the coefficients are treated independently as 1D multichannel data.

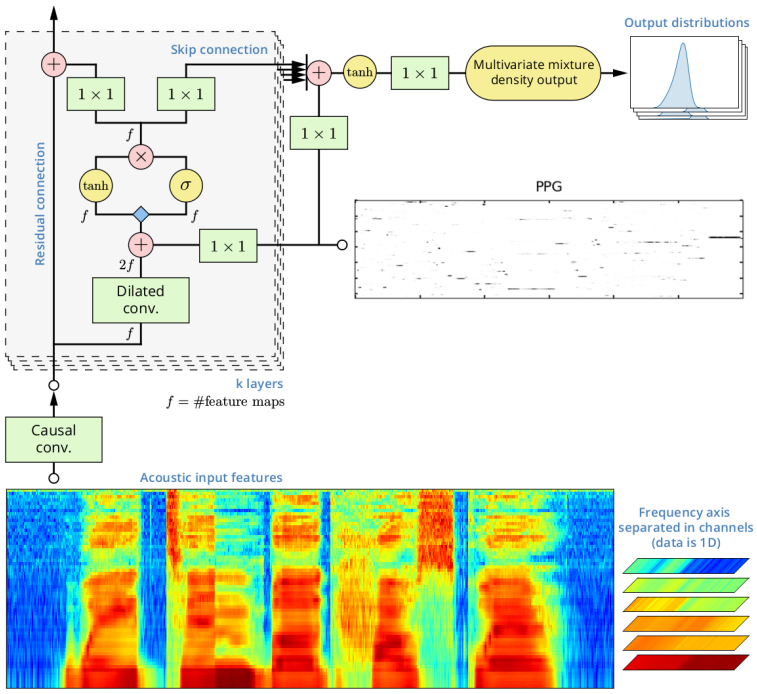

Figure 5: NPSS architecture. PPG are used as control vector for each of the input features.

The model is trained to maximize the likelihood of an observation given past observations in order to predict the conditional probability distributions of the features. The synthesis is performed by sampling the distribution conditioned on past predictions.

This system is very suitable for the current VC scenario, where the pitch can be extracted from the source speaker's utterances. The only important change is the substitution of the categorical phonetic labels used as control for the phonetic posteriors generated by the ASR model. On fig.5 the architecture of the modified NPSS can be seen.

3.2.2 Target features

A separated instance of NPSS synthesizes each of the features required as input by the World vocoder. This is, an harmonic part or spectral envelope (SP), an aperiodic part or (AP) and F0.

F0 could be directly taken from the analysis of the source signal without being processed by the net. However, it has been seen that the synthesis quality is very low in the regions where the voiced and unvoiced frames are not properly detected. Thus, as the source and synthetic voiced/unvoiced regions do not have to match exactly, an instance of the network estimates the voiced/unvoiced (VUV) frames as a binary 1D vector.

CheapTrick [33] was used to estimate the spectral envelope. F0 is used to make a period-synchronous power smoothing of the signal. After this, the signal is low-pass filtered in the frequency domain with a square window. Finally, liftering is applied to reduce the time domain dependency. In order to reduce the dimensionality of the SP vector, it is transformed into Mel Frequency Spectral Coefficients (MFSC). This is done by taking the FFT of the mel-cepstral analysis of the SP. Mel-cepstral analysis is done following the SPTK approach [43]. This process reduces the dimensionality of the features from 1025 to 60 coefficients.

World makes use of an aperiodicity parameter in order to improve the audio quality of the synthesis. This parameter tries to model the characteristic stochastic component of the voice that is independent of the pitch. D4C [44] was used to estimate the aperiodicity in 6 frequency bands.

3.2.3 Synthesis

World [45] relies in the fundamental frequency (F0), spectral envelope (SP) and aperiodicity (AP) features to synthesize high quality voice. The model uses F0 to calculate the time positions of the excitation signal, a train of pulses representing the osculation of the vocal chords. The timbre of the voice of the speaker is achieved by convolving the excitation signal with the impulse response that can be obtained from the SP. AP is used to improve the quality of the synthesis by controlling the amount of aperiodic component added to the signal.

Two different F0 estimators have been tried. DIO [46] and spectral autocorrelation (SAC) [47]. During the experiments it was found that the second one is more robust

and leads to less octave errors. However this was not formally tested. A proper F0 estimation is crucial in this system as SP and AP also rely on it. For the synthesis, the analysis F0 is interpolated along the unvoiced fragments and the resulting F0 is then computed by removing the voiced frames according to the VUV vector.

In speech VC it is typical to adapt the pitch of the source to the range of the target. This is sometimes easily achieved by fitting the pitch data to the target's mean and variance. However, as for singing voice the F0, or in other words, the melody, should remain the same despite the singer, it is not desirable to change the fundamental frequency of the converted voice. Only in the case of the cross-gender experiment, octave pitch shifts are applied to fit the most suitable tessitura for the target singer.

AP and SP features are retrieved from the NPSS parameters explained in the previous sections. SP is retrieved from the MFSC by doing the inverse steps. This is, inverse FFT followed by inverse mel-cepstral analysis. Optionally, inverse mel-cepstral analysis can be approximated with splines for computational reasons. AP is obtained from band aperiodicity by interpolation.

Chapter 4

Results

4.1 Models

Following the methodology proposed in the previous chapter, 5 models were built.

1. **Phonetic Model:** TIMIT corpus and FFT (spectrum) based MFCC for the phonetic model.
   Voice Model: trained with a male constrained singing database.

2. **Phonetic Model:** TIMIT corpus and spectral envelope based MFCC.
   Voice Model: trained with a male constrained singing database.

3. **Phonetic Model:** TIMIT corpus and spectral envelope based MFCC.
   Voice Model: trained with a female constrained singing database.

4. **Phonetic Model:** Zxx corpus and spectral envelope based MFCC.
   Voice Model: trained with a male constrained singing database.

5. **Phonetic Model:** Zxx corpus and spectral envelope based MFCC.
   Voice Model: was trained with a female constrained singing database.

4.2 Scenarios

The models were evaluated in different scenarios. Thanks to the many-to-one property of the system it was easy to perform the following conversions without any previous information of the new speakers nor parallel data.

1. Same gender and same language voice conversion.
2. Cross gender and same language voice conversion.
3. Same language and cross-lingual voice conversion.
4. Cross-gender and cross-lingual voice conversion.

4.3 Subjective Evaluations

A parallel corpus is required to compute VC objective metrics. As this requirement was not available on this project, evaluations were exclusively focused in subjective methods. Furthermore, it is known that traditional metrics, as mel cepstral distortion, are not always correlated with the perception of quality in VC. Thus, subjective evaluations are definitively a more valuable method.

A perceptual test was created in the form of a web survey[1]. This test was completed by 28 people mainly coming from a technical or musical background.

The goal of the survey was to compare the behavior of the different models in all the possible scenarios in order to find the best parameters. The testers are asked to decide which system sounds the best in terms of *intelligibility* and closeness to the target voice.

Every test item is composed by an input and a target sentences and A and B options. The input sentence is the signal to convert. The target sentence is a sample of the dataset used for the voice model training (or target speaker). A and B are the signal

[1] https://pabloentropia.github.io/voice-conversion.github.io/

converted by two of the models described above. This information is supposed to be enough to understand and evaluate the models. The possible answers are *A*, *B* or *no preference*. In order to prevent a potential bias, *A* and *B* are presented randomly.

The answers were used to evaluate independently which acoustic features and which training datasets produced the best results. Additionally cross-gender vs. same gender conversion are evaluated. The results are available in fig.6.

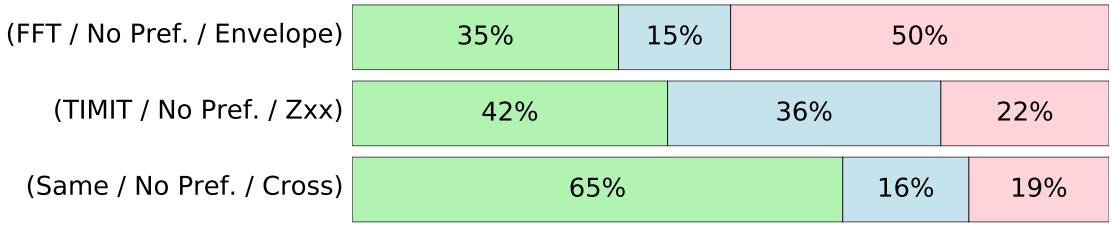

Figure 6: Preference, expressed in percentages, of different aspects of the conversion.

The following tables express the results of the test for each item. This results are discussed in the last chapter of this thesis.

Scenario	Model A	% A	% No Pref.	% B	Model B
M2M S	TIMIT FFT	55.6%	25.9%	18.5%	TIMIT SP
M2F S	TIMIT FFT	7.4%	3.7%	88.9%	TIMIT SP

Table 1: Test items related to feature preference sorted by scenario and models. In the scenario description, M: Male, F: Female, S: Same language, C: Cross-lingual.

Scenario	Model A	% A	% No Pref.	% B	Model B
M2M S	TIMIT SP	18.5%	33.3%	48.1%	ZXX SP
M2F S	TIMIT SP	70.4%	3.7%	25.9%	ZXX SP
M2M C	TIMIT SP	77.8%	14.8%	7.4%	ZXX SP
M2F C	TIMIT SP	29.6%	29.6%	40.7%	ZXX SP
F2M S	TIMIT SP	25.9%	48.1%	25.9%	ZXX SP
F2F S	TIMIT SP	37.0%	55.6%	7.4%	ZXX SP
F2M C	TIMIT SP	18.5%	66.7%	14.8%	ZXX SP
F2F C	TIMIT SP	25.9%	37.0%	37.0%	ZXX SP

Table 2: Test items related to dataset preference sorted by scenario and models. In the scenario description, M: Male, F: Female, S: Same language, C: Cross-lingual.

Scenario A	Model A	% A	% No Pref.	% B	Model B	Scenario B
M2M S	TIMIT SP	55.6%	14.8%	29.6%	TIMIT SP	F2M S
F2F S	TIMIT SP	74.1%	18.5%	7.4 %	TIMIT SP	M2F S

Table 3: Test items related to gender preference sorted by scenario and models. In the scenario description, M: Male, F: Female, S: Same language, C: Cross-lingual.

Demonstrations of the most successful model, TIMIT corpus and spectral envelope based MFCC, featuring all the contemplated scenarios are available in this website[2].

[2]https://pabloentropia.github.io/voice-conversion-demo.github.io/

Chapter 5

Conclusions

5.1 Conclusions

All the goals proposed for this thesis have been completed. The PPG based voice conversion system was implemented and 2 target speakers (a male and a female) were modeled. 3 models were built for the male and 2 for the female singer, featuring different parameter configurations.

Conversions with each of these models were performed in all the proposed scenarios. These conversions were subjectively evaluated by 28 testers in order to understand which models perform the best for each scenario. The conclusions extracted from the results of the tests are presented now:

1. The first goal of the test was to find if the proposed SP based MFCC overcame the default MFCC for in phonetic model. Most of the testers preferred the default MFCC for the same gender experiment but the SP is clearly superior (88.9% of the votes) for the male to female case. The reason found for this is that the default MFCC are not pitch independent and the ASR gets confused with high pitched singing, especially if such kind of data did not appeared during the training. The use of MFCC computed on top of the pitch-synchronous SP was found to alleviate this effect.

2. One of the main reasons of phonetic recognition mismatch is known to be the lack of similar speakers during the training stage. This is why training the SI-ASR system with singing data was expected to be beneficial for the overall behavior of the system. However, it was found that in most scenarios the use of the speech dataset is preferred. The explanation of this phenomenon can be probably in the architecture of the dataset. Despite the overall recorded time of the dataset is smaller, TIMIT (speech) is formed by 630 speakers performing 10 utterances each one and considering the 8 major american accents. In the other hand, Zxx (singing) is just composed by 11 singers with 577 utterances per each and it is not particularly appropriated for the speech recognition task.

3. As expected, all the models showed a better behavior when the conversion was performed within the same gender.

5.2 Future Lines

1. Explore new DNN architectures as a last step on the ASR stage. Following the recent success of deep learning, it could be interesting to explore the behavior of the ASR stage using different architectures.

2. Use a dataset with parallel recordings to build a new conversion model. Then, it would be also possible to compute objective measures as the Mel Cepstral Distance (MCD) and compare this result with other systems.

3. From the the experiments it was observed that training the ASR system with the speech or the constrained singing corpus did not affect a lot the quality perfection. In both cases the system had problems to retrieve the proper phonetic class at some points. However, these mistakes are produced in different points. Thus, creating an hybrid (speech/singing) corpus could help to slightly reduce the overall phonetic mismatch.

4. From the implementation of the project, it was observed that the global conversion speed is about three times real time. This makes the system suitable for a real-time implementation, however it was not developed yet.

List of Figures

List of Tables

Bibliography

[1] Stylianou, Y., Cappé, O. & Moulines, E. Continuous probabilistic transform for voice conversion. *IEEE Transactions on Speech and Audio Processing* **6**, 131–142 (1998).

[2] Hamidreza Mohammadi, S. & Kain, A. An overview of voice conversion systems. *Speech Communication* **88**, 65–82 (2017). URL `www.elsevier.com/locate/specom`.

[3] Sun, L., Li, K., Wang, H., Kang, S. & Meng, H. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *Multimedia and Expo (ICME), 2016 IEEE International Conference on*, 1–6 (IEEE, 2016).

[4] Sambur, M. Selection of acoustic features for speaker identification. *IEEE Transactions on Acoustics, Speech, and Signal Processing* **23**, 176–182 (1975).

[5] Furui, S. Research of individuality features in speech waves and automatic speaker recognition techniques. *Speech communication* **5**, 183–197 (1986).

[6] Abe, M., Nakamura, S., Shikano, K. & Kuwabara, H. Voice conversion through vector quantization. *Journal of the Acoustical Society of Japan (E)* **11**, 71–76 (1990).

[7] Wang, Y.-P., Ling, Z.-H. & Wang, R.-H. Emotional speech synthesis based on improved codebook mapping voice conversion. *Affective Computing and Intelligent Interaction* 374–381 (2005).

[8] Kain, A. & Macon, M. W. Spectral voice conversion for text-to-speech synthesis. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 1, 285–288 (IEEE, 1998).

[9] Helander, E., Virtanen, T., Nurminen, J. & Gabbouj, M. Voice conversion using partial least squares regression. *IEEE Transactions on Audio, Speech, and Language Processing* **18**, 912–921 (2010).

[10] Toda, T., Saruwatari, H. & Shikano, K. Voice conversion algorithm based on gaussian mixture model with dynamic frequency warping of straight spectrum. In *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 2, 841–844 (IEEE, 2001).

[11] Chen, Y., Chu, M., Chang, E., Liu, J. & Liu, R. Voice conversion with smoothed gmm and map adaptation. In *Eighth European Conference on Speech Communication and Technology* (2003).

[12] Toda, T., Black, A. W. & Tokuda, K. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing* **15**, 2222–2235 (2007).

[13] Narendranath, M., Murthy, H. A., Rajendran, S. & Yegnanarayana, B. Transformation of formants for voice conversion using artificial neural networks. *Speech communication* **16**, 207–216 (1995).

[14] Desai, S., Raghavendra, E. V., Yegnanarayana, B., Black, A. W. & Prahallad, K. Voice conversion using artificial neural networks. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 3893–3896 (IEEE, 2009).

[15] Desai, S., Black, A. W., Yegnanarayana, B. & Prahallad, K. Spectral mapping using artificial neural networks for voice conversion. *IEEE Transactions on Audio, Speech, and Language Processing* **18**, 954–964 (2010).

[16] Chen, L.-H., Ling, Z.-H., Song, Y. & Dai, L.-R. Joint spectral distribution modeling using restricted boltzmann machines for voice conversion. In *Interspeech*, 3052–3056 (2013).

[17] Chen, L.-h., Ling, Z.-h., Liu, L.-j. & Dai, L.-r. Voice conversion using deep neural networks with Layer-Wise Generative Training. *IEEE International Conference on Audio, Speech, and Language Processing* **22**, 1859–1872 (2014).

[18] Nakashika, T., Takashima, R., Takiguchi, T. & Ariki, Y. Voice conversion in high-order eigen space using deep belief nets. In *Interspeech*, 369–372 (2013).

[19] Sun, L., Kang, S., Li, K. & Meng, H. Voice Conversion Using Deep Bidirectional Long Short-Term Memory Based Recurrent Neural Networks. *Icassp 2015* 4869–4873 (2015).

[20] Gers, F. A., Schmidhuber, J. & Cummins, F. Learning to forget: Continual prediction with lstm. *Neural computation* **12**, 2451–2471 (2000).

[21] Sündermann, D., Bonafonte, A., Ney, H. & Höge, H. A First Step Towards Text-Independent Voice Conversion .

[22] Ye, H. & Young, S. J. Voice conversion for unknown speakers. In *INTERSPEECH* (2004).

[23] Erro, D., Moreno, A. & Bonafonte, A. INCA algorithm for training voice conversion systems from nonparallel corpora. *IEEE Transactions on Audio, Speech and Language Processing* **18**, 944–953 (2010).

[24] Xie, F.-l., Soong, F. K. & Li, H. A KL Divergence and DNN-based Approach to Voice Conversion without Parallel Training Sentences 287–291 (2016).

[25] Turk, O., Buyuk, O., Haznedaroglu, A. & Arslan, L. M. Application of voice conversion for cross-language rap singing transformation. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 3597–3600 (IEEE, 2009).

[26] Villavicencio, F. & Bonada, J. Applying voice conversion to concatenative singing-voice synthesis. In *Interspeech*, 2162–2165 (2010).

[27] Kawakami, Y., Banno, H. & Itakura, F. Gmm voice conversion of singing voice using vocal tract area function. *IEICE technical report. Speech (Japanese edition)* **110**, 71–76 (2010).

[28] Toda, T., Ohtani, Y. & Shikano, K. ONE-TO-MANY AND MANY-TO-ONE VOICE CONVERSION BASED ON EIGENVOICES .

[29] Doi, H., Toda, T., Nakano, T., Goto, M. & Nakamura, S. Singing voice conversion method based on many-to-many eigenvoice conversion and training data generation using a singing-to-singing synthesis system. In *Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC), 2012 Asia-Pacific*, 1–6 (IEEE, 2012).

[30] Sun, L., Wang, H., Kang, S., Li, K. & Meng, H. Personalized, cross-lingual tts using phonetic posteriorgrams. *Interspeech 2016* 322–326 (2016).

[31] Mohri, M., Pereira, F. & Riley, M. Springer Handbook on Speech Processing and Speech Communication SPEECH RECOGNITION WITH WEIGHTED FINITE-STATE TRANSDUCERS URL `http://www.cs.nyu.edu/{~}mohri/pub/hbka.pdf`.

[32] Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G. & Pallett, D. S. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n* **93** (1993).

[33] Morise, M. Cheaptrick, a spectral envelope estimator for high-quality speech synthesis. *Speech Communication* **67**, 1–7 (2015).

[34] Povey, D., Ghoshal, A. & Boulianne, G. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, EPFL-CONF-192584 (IEEE Signal Processing Society, 2011).

[35] Gales, M. J. Semi-tied covariance matrices for hidden markov models. *IEEE transactions on speech and audio processing* **7**, 272–281 (1999).

[36] Gales, M. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech & Language* **12**, 75–98 (1998). URL http://www.sciencedirect.com/science/article/pii/S0885230898900432. arXiv:1011.1669v3.

[37] Sankar, A. & Lee, C.-H. A maximum-likelihood approach to stochastic matching for robust speech recognition. *IEEE transactions on speech and Audio Processing* **4**, 190–202 (1996).

[38] Hinton, G. A Practical Guide to Training Restricted Boltzmann Machines (2010). URL http://learning.cs.toronto.edu.

[39] Blaauw, M. & Bonada, J. A NEURAL PARAMETRIC SINGING SYNTHESIZER URL https://arxiv.org/pdf/1704.03809.pdf.

[40] Oord, A. v. d., Kalchbrenner, N. & Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759* (2016).

[41] van den Oord, A. *et al.* Wavenet: A generative model for raw audio. *CoRR abs/1609.03499* (2016).

[42] Kalchbrenner, N. *et al.* Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099* (2016).

[43] Group, S. W. *et al.* Speech signal processing toolkit (sptk). *h ttp://sp-tk. sourceforge. net* (2009).

[44] Morise, M. D4c, a band-aperiodicity estimator for high-quality speech synthesis. *Speech Communication* **84**, 57–65 (2016).

[45] Morise, M., Yokomori, F. & Ozawa, K. World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems* **99**, 1877–1884 (2016).

[46] Morise, M., Kawahara, H. & Katayose, H. Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice

and speech. In *Audio Engineering Society Conference: 35th International Conference: Audio for Games* (Audio Engineering Society, 2009).

[47] Lahat, M., Niederjohn, R. & Krubsack, D. A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech. *IEEE transactions on acoustics, speech, and signal processing* **35**, 741–750 (1987).