Gaussian Process Experts for Voice Conversion

Nicholas C.V. Pilkington, Heiga Zen, Mark J. F. Gales

Toshiba Research Europe Ltd., Cambridge Research Lab, Cambridge, UK

nicholas.pilkington@cl.cam.ac.uk, {heiga.zen,mark.gales}@crl.toshiba.co.uk

Abstract

Conventional approaches to voice conversion typically use a GMM to represent the joint probability density of source and target features. This model is then used to perform spectral conversion between speakers. This approach is reasonably effective but can be prone to overfitting and oversmoothing of the target spectra. This paper proposes an alternative scheme that uses a collection of Gaussian process experts to perform the spectral conversion. Gaussian processes are robust to overfitting and oversmoothing and can predict the target spectra more accurately. Experimental results indicate that the objective performance of voice conversion can be improved using the proposed approach.

Index Terms: Gaussian processes, GMM, voice conversion

1. Introduction

Voice conversion (VC) is a technique for allowing the speech characteristics to be altered from a source speaker to a target speaker. In statistical approaches to VC, a feature mapping function is trained in advance using data consisting of utterance pairs of source and target voices. The resulting mapping function is then used to convert any samples of the source speech into that of the target.

The standard statistical technique used in VC is to train a GMM on the joint probability density of source and target spectra and derive the conditional probability density given source spectra to be converted [1, 2]. This technique has a number of limitations; two of them are oversmoothing and overfitting. Oversmoothing exhibits itself when there is not sufficient flexibility in the model to capture the precise structure of the target spectra. The most significant impact of this is the oversmoothing of the target spectra. Adding more mixture components allows for more flexibility and can start to address the problem of oversmoothing but soon encounters overfitting to the data. Also the statistical parametric and semi-parametric models are known to have limited ability. as more data is introduced as they lose flexibility.

To address this problem, Gaussian processes (GPs) [3] for VC is proposed. GPs are non-parametric Bayesian models that can be viewed as a distribution over functions. They have many advantages over the conventional parametric approaches, such as flexibility and robustness against overfitting. These advantages indicates that GPs can address the issues of oversmoothing and overfitting in VC.

The rest of this paper is organized as follows. Section 2 reviews conventional GMM-based VC. Section 3 describes the proposed approach of VC using GP experts. Section 4 shows experimental results. Concluding remarks are presented in the final section.

This work was performed while the first author was an intern in Toshiba Research Europe Ltd. in summer 2010. His original affiliation was Computer Laboratory, University of Cambridge, Cambridge, UK.

2. GMM-based voice conversion

Let x_t and y_t be spectral features at frame t for the source and target voices respectively.[1] GMM-based voice conversion models the joint probability density of the source and target spectral features by a GMM, thus

$$p(\boldsymbol{z}_t \mid \boldsymbol{\lambda}^{(z)}) = \sum_{m=1}^{M} w_m \, \mathcal{N}(\boldsymbol{z}_t \, ; \, \boldsymbol{\mu}_m^{(z)}, \boldsymbol{\Sigma}_m^{(z)}), \tag{1}$$

where $\boldsymbol{z}_t$ is a joint vector $[x_t, y_t]^\top$, m is the mixture component index, M is the total number of components, w_m is the weight of the m-th component. The mean vector and covariance matrix of the m-th component, $\boldsymbol{\mu}_m^{(z)}$ and $\boldsymbol{\Sigma}_m^{(z)}$, are given as

$$\boldsymbol{\mu}_m^{(z)} = \left[\mu_m^{(x)}, \mu_m^{(y)}\right]^\top, \quad \boldsymbol{\Sigma}_m^{(z)} = \begin{bmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} \\ \Sigma_m^{(yx)} & \Sigma_m^{(yy)} \end{bmatrix}. \tag{2}$$

The parameters of the GMM will be denoted as $\boldsymbol{\lambda}^{(z)}$, which consists of weights, mean vectors, and the covariance matrices for the individual components. The parameter set $\boldsymbol{\lambda}^{(z)}$ is estimated from supervised training data, $\{x_1^*, y_1^*\}, \ldots, \{x_N^*, y_N^*\}$, which is expressed as $\boldsymbol{x}^*, \boldsymbol{y}^*$ for the source and targets, based on the maximum likelihood (ML) criterion as

$$\hat{\boldsymbol{\lambda}}^{(z)} = \arg\max_{\boldsymbol{\lambda}^{(z)}} \, p(\boldsymbol{z}^* \mid \boldsymbol{\lambda}^{(z)}), \tag{3}$$

where $\boldsymbol{z}^*$ is the set of training joint vectors $\boldsymbol{z} = \{\boldsymbol{z}_1^*, \ldots, \boldsymbol{z}_N^*\}$ and $\boldsymbol{z}_t^*$ is the joint vector at frame t, $\boldsymbol{z}_t^* = [x_t^*, y_t^*]^\top$.

The conditional probability density of y_t given x_t can be derived from the estimated GMM as

$$p(y_t \mid x_t, \hat{\boldsymbol{\lambda}}^{(z)}) = \sum_{m=1}^{M} P(m \mid x_t, \hat{\boldsymbol{\lambda}}^{(z)}) p(y_t \mid x_t, m, \hat{\boldsymbol{\lambda}}^{(z)}),$$

where

$$p(y_t \mid x_t, m, \boldsymbol{\lambda}^{(z)}) = \mathcal{N}(y_t \, ; \, \mu_m^{(y|x_t)}, \Sigma_m^{(y|x_t)}), \tag{4}$$

$$\mu_m^{(y|x_t)} = \mu_m^{(y)} + \Sigma_m^{(yx)} {\Sigma_m^{(xx)}}^{-1} (x_t - \mu_m^{(x)}), \tag{5}$$

$$\Sigma_m^{(y|x_t)} = \Sigma_m^{(yy)} - \Sigma_m^{(yx)} {\Sigma_m^{(xx)}}^{-1} \Sigma_m^{(xy)}. \tag{6}$$

In the conventional approach [1, 2], the conversion is performed on the basis of the minimum mean-square error (MMSE) as

$$\hat{y}_t = \sum_{m=1}^{M} P(m \mid x_t, \hat{\boldsymbol{\lambda}}^{(z)}) \, \mu_m^{(y|x_t)}. \tag{7}$$

[1] For notation simplicity, it is assumed that x_t and y_t are scalar values. Extending them to vectors is straightforward.

Although GMM-based mapping works reasonably well, there exist a number of issues that have been observed with the converted speech. Two important problems are: 1) each frame is mapped independently of the surrounding frames; and 2) the oversmoothing effect. The first problem occurs because the correlations between the target feature vectors are ignored. To address this problem, dynamic feature constraints were introduced [4]. Both the static and dynamic parameters are converted, yielding a set of Gaussian *experts* to estimate each dimension. Thus $\boldsymbol{z}_t = [x_t, y_t, \Delta x_t, \Delta y_t]^\top$ where $\Delta x_t = x_{t+1} - x_t$ and $\Delta y_t = y_{t+1} - y_t$. Using this modified joint model, a GMM is trained with the following parameters for each component m:

$$\boldsymbol{\mu}_m^{(z)} = \begin{bmatrix} \mu_m^{(x)} & \mu_m^{(y)} & \mu_m^{(\Delta x)} & \mu_m^{(\Delta y)} \end{bmatrix}^\top, \tag{8}$$

$$\boldsymbol{\Sigma}_m^{(z)} = \begin{bmatrix} \Sigma_m^{(xx)} & \Sigma_m^{(xy)} & 0 & 0 \\ \Sigma_m^{(yx)} & \Sigma_m^{(yy)} & 0 & 0 \\ 0 & 0 & \Sigma_m^{(\Delta x \Delta x)} & \Sigma_m^{(\Delta x \Delta y)} \\ 0 & 0 & \Sigma_m^{(\Delta y \Delta x)} & \Sigma_m^{(\Delta y \Delta y)} \end{bmatrix}. \tag{9}$$

Note to limit the number of parameters in the covariance matrix of $\boldsymbol{z}$ the static and delta parameters are assumed to be conditionally independent given the component. When applying voice conversion to a particular source sequence, this will yield two experts (assuming just delta parameters are added) as

- *static* expert: $\mathcal{N}(y_t \,;\, \mu_{\hat{m}_t}^{(y|x_t)}, \Sigma_{\hat{m}_t}^{(y|x_t)})$
- *dynamic* expert: $\mathcal{N}(\Delta y_t \,;\, \mu_{\hat{m}_t}^{(\Delta y|\Delta x_t)}, \Sigma_{\hat{m}_t}^{(\Delta y|\Delta x_t)})$

where $\hat{m}_t$ is the most probable mixture component at frame t and $\mu_{\hat{m}_t}^{(\Delta y|\Delta x_t)}$ and $\Sigma_{\hat{m}_t}^{(\Delta y|\Delta x_t)}$ are derived in the same fashion as the static parameters. $\hat{\boldsymbol{y}} = \{\hat{y}_1, \ldots, \hat{y}_T\}$ that maximises the output probability given both experts is produced

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \prod_{t=1}^{T} \Big\{ \mathcal{N}(y_t \,;\, \mu_{\hat{m}_t}^{(y|x_t)}, \Sigma_{\hat{m}_t}^{(y|x_t)}) \cdot \mathcal{N}(\Delta y_t \,;\, \mu_{\hat{m}_t}^{(\Delta y|\Delta x_t)}, \Sigma_{\hat{m}_t}^{(\Delta y|\Delta x_t)}) \Big\}. \tag{10}$$

Via the use of dynamic experts the independent mapping issue can be addressed. However the second problem of oversmoothing has not been addressed. The statistical modeling often removes the fine details of spectral structures. Although this smoothing allows a statistical model to be easily trained, it also causes quality degradation of the converted speech. For high-quality synthesized speech the level of smoothing must be reduced.

3. GP-based voice conversion

Instead of using GMMs, an approach for VC using GPs to learn a mapping function between source and target speakers is proposed. This section describes the overview of GP and its application to VC.

3.1. Gaussian processes

The underlying model for a number of prediction models is

$$y_t = f(x_t; \boldsymbol{\lambda}) + \epsilon, \tag{11}$$

where $f(\cdot; \boldsymbol{\lambda})$ is a function to predict output given input, ϵ is a noise term, and $\boldsymbol{\lambda}$ is the parameter set that define $f(\cdot; \boldsymbol{\lambda})$. If the data is assumed to be iid and the distribution of the noise term is Gaussian with zero mean, the likelihood of the data is

$$p(y_t \mid x_t, \boldsymbol{\lambda}) = \mathcal{N}\left(y_t \,;\, f(x_t; \boldsymbol{\lambda}), \sigma^2\right), \tag{12}$$

where σ^2 is the variance of noise Gaussian.

Instead of using parametric function for $f(\cdot; \boldsymbol{\lambda})$, it is possible to use kernel-based, non-parametric approaches, such as GPs [3]. GPs are flexible models that fit within a probabilistic Bayesian modelling framework. A GP can be used as a prior probability distribution over functions in Bayesian inference. Given any set of N points in the desired domain of functions, a multivariate Gaussian whose covariance matrix parameter is the Gramian matrix of the N points with some desired covariance (kernel) function, and sample from that Gaussian. Prediction of continuous values with a GP prior is known as GP regression [3]. GPs can be viewed as an extension of multivariate Gaussian distributions to infinite numbers of variables. GPs have many attractive properties. As they are Gaussian in nature, it is possible to compute the marginal integrals. This allows a "pure" Bayesian framework to be used. Furthermore as GPs are non-parametric in nature it is unnecessary to specify the parametric form a-priori. With this non-parametric model, the previously used parameters are replaced with the actual function itself. Thus a GP prior represents a distribution over functions.

The same general Bayesian likelihood function of Eq. (12) is used. However now it is not a parametric function. Instead, a GP prior for $f(x; \boldsymbol{\lambda})$ is used

$$f(x; \boldsymbol{\lambda}) \sim \mathcal{GP}\left(m(x), k(x, x')\right), \tag{13}$$

where $k(x, x')$ is a covariance function, which defines the "similarity" between x and x', and $m(x)$ is the mean function. This leads to a GP predictive distribution [3] as

$$p(y_t \mid x_t, \boldsymbol{x}^*, \boldsymbol{y}^*) = \mathcal{N}\left(y_t \,;\, \mu(x_t), \Sigma(x_t)\right), \tag{14}$$

where $\mu(x_t)$ and $\Sigma(x_t)$ are the mean and variance of the predictive distribution given x_t. These may be expressed as

$$\mu(x_t) = m(x_t) + \boldsymbol{k}_t^\top [\boldsymbol{K}^* + \sigma^2 \boldsymbol{I}]^{-1} (\boldsymbol{y}^* - \boldsymbol{\mu}^*), \tag{15}$$

$$\Sigma(x_t) = k(x_t, x_t) + \sigma^2 - \boldsymbol{k}_t^\top [\boldsymbol{K}^* + \sigma^2 \boldsymbol{I}]^{-1} \boldsymbol{k}_t, \tag{16}$$

where σ is the noise variance (hyper-parameter), $\boldsymbol{\mu}^*$ is the training mean vector, and $\boldsymbol{K}^*$ and $\boldsymbol{k}$ are Gramian matrices. They are given as

$$\boldsymbol{\mu}^* = \begin{bmatrix} m(x_1^*) & m(x_2^*) & \ldots & m(x_N^*) \end{bmatrix}^\top, \tag{17}$$

$$\boldsymbol{K}^* = \begin{bmatrix} k(x_1^*, x_1^*) & k(x_1^*, x_2^*) & \ldots & k(x_1^*, x_N^*) \\ k(x_2^*, x_1^*) & k(x_2^*, x_2^*) & \ldots & k(x_2^*, x_N^*) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N^*, x_1^*) & k(x_N^*, x_2^*) & \ldots & k(x_N^*, x_N^*) \end{bmatrix}, \tag{18}$$

$$\boldsymbol{k}_t = \begin{bmatrix} k(x_1^*, x_t) & k(x_2^*, x_t) & \ldots & k(x_N^*, x_t) \end{bmatrix}^\top, \tag{19}$$

The above equations include a matrix inversion, which is $\mathcal{O}(N^3)$. However, sparse methods and other approximations are possible to reduce its computational complexity.

3.2. Mapping with GP experts

GPs can be directly used for the mapping process in VC. The parametric form of conversion is simply replaced by a GP. From Eqs. (15) and (16) the means and covariance matrices for the prediction can be obtained. The MMSE estimates of target feature are given as

$$\hat{y}_t = \mu(x_t). \tag{20}$$

However this is frame-by-frame prediction again thus it may be suffered from the discontinuity problem. To address this problem the dynamic features can also be predicted. Thus two GP experts can be produced:

- *static* expert: $y_t \sim \mathcal{N}(\mu(x_t), \Sigma(x_t))$
- *dynamic* expert: $\Delta y_t \sim \mathcal{N}(\mu(\Delta x_t), \Sigma(\Delta x_t))$

The GPs for each of the static and delta experts are trained independently, though this is not necessary. In the same fashion as the standard GMM VC process it is possible to use these as

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \prod_{t=1}^{T} \Big\{ \mathcal{N}\left(y_t \ ; \ \mu(x_t), \Sigma(x_t)\right) \cdot \mathcal{N}\left(\Delta y_t \ ; \ \mu(\Delta x_t), \Sigma(\Delta x_t)\right) \Big\}. \quad (21)$$

As the GP predictive distributions are Gaussian, the speech parameter generation algorithm for VC [4] can be used to solve the above maximization problem and generate the smooth trajectories of target static features from the GP experts.

3.3. Covariance and mean functions

A GP is described by its covariance and mean functions. The covariance function of a GP can be thought of as a measure that describes the local covariance of a smooth function. Thus a data point with a high covariance (kernel) function value with another is likely to deviate from its mean in the same direction as the other point. Not all functions are covariance functions as they need to form a positive definite Gramian matrix [3].

There exist two kinds of covariance functions, stationary and non-stationary. A stationary covariance function is a function of $x - x'$. Thus it is invariant stationarity to translations in the input space. An example of a stationary covariance function is the squared exponential covariance function

$$k(x, x') = \exp\left\{-\frac{1}{2}\left(x - x'\right)^2\right\}, \quad (22)$$

which is a function of the distance between its input vectors. This contrast with non-stationary covariance functions that will give difference values. An example of a non-stationary covariance function is the linear covariance function

$$k(x, x') = x \cdot x'. \quad (23)$$

Both types can be of use.

The covariance functions described above are parameter free. It is also possible to have covariance functions that have *hyper-parameters* that can be trained. One example is a linear covariance function with automatic relevance detection (ARD)

$$k(x, x') = x \cdot P \cdot x', \quad (24)$$

where P is a hyper-parameter that needs to be trained. A combination of covariance functions can also be used.

There exist also a few choices for the mean function of a GP; a zero mean, $m(x) = 0$, a constant mean $m(x) = b$, a linear mean $m(x) = ax$, or their combination $m(x) = ax + b$, where a and b are hyper-parameters of the mean functions.

3.4. Optimizing hyper-parameters

Covariance and mean functions often have parameters and selecting good values for these parameters has an impact on the performance of the predictor. These hyper-parameters can be set a priori but it makes sense to set them to the values that best describe the data; maximize the negative marginal log likelihood of the data. Here, the hyper-parameters are optimized using Polack-Ribiere conjugate gradients to compute the search directions, and a line search using quadratic and cubic polynomial approximations and the Wolfe-Powell stopping criteria was used together with the slope ratio method for guessing initial step sizes. This was a time consuming operation but did increase the performance. However when run for too many iteration the GP can start to overfit. According to [3], 25 iterations were used in the experiments reported in the next section.

3.5. Partitioning the input space

The size of the Gramian matrix, $\boldsymbol{K}$, which is equal to the number of samples in the training data, can be tens of thousands in VC. This prevents the direct application of GPs to VC because computing inverse of the Gramian matrix is $\mathcal{O}(N^3)$. To address this problem, the input space is first divided into its sub-spaces then a GP is trained for each sub-space. This reduces the number of samples that are trained for each GP. This circumvents the issue of slow matrix inversion and also make for a more accurate training procedure that improve the accuracy of the mapping on a per-cluster level. The LBG algorithm with the Euclidean distance in mel-cepstral coefficients was used to split the data into its sub-spaces.

4. Experiments

4.1. Experimental setup

Fifty sentences uttered by female speakers, CLB and SLT, from the CMU ARCTIC database were used for training (source: CLB, target: SLT). Fifty sentences, which were not included in the training data, were used for evaluation. Speech signals were sampled at a rate of 16 kHz and windowed with 5 ms of shift, and then 40th-order mel-cepstral coefficients were obtained by using a mel-cepstral analysis technique. The $\log F_0$ values for each utterance were also extracted. The feature vectors of source and target speech consisted of 41 mel-cepstral coefficients including the zeroth coefficients. The DTW algorithm was used to obtain time alignments between source and target feature vector sequences. Based on the DTW results, joint feature vectors were composed for training joint probability density between source and target features. The total number of training frames was 34,664.

Five systems were compared in this experiment;

- GMMs without dynamic features [1, 2];
- GMMs with dynamic features [4];
- trajectory GMMs [5];
- GPs without dynamic features (proposed);
- GPs with dynamic features (proposed).

They were trained from the composed joint feature vectors. The dynamic features (delta and delta-delta features) were calculated as

$$\Delta x_t = 0.5x_{t+1} - 0.5x_{t-1}, \quad (25)$$

$$\Delta^2 x_t = x_{t+1} - 2x_t + x_{t-1}. \quad (26)$$

Covariance and cross-covariance matrices, $\{\boldsymbol{\Sigma}_m^{(xx)}, \boldsymbol{\Sigma}_m^{(yy)}\}_{m=1}^M$ and $\{\boldsymbol{\Sigma}_m^{(xy)}, \boldsymbol{\Sigma}_m^{(xy)}\}_{m=1}^M$, of the GMMs and trajectory GMMs were all diagonal. For GP-based VC, the input space (mel-cepstral coefficients from the source speaker) was split into 32 regions using the LBG algorithm then trained a GP for each cluster for each dimension. From a preliminary experiment, the combination of constant and linear functions, $m(x) = ax + b$, was chosen for the mean function of GP-based VC.

The $\log F_0$ values in this experiment were converted by using the simple linear conversion [4]. The speech waveform was re-synthesized from the converted mel-cepstral coefficients and $\log F_0$ values through the mel log spectrum approximation (MLSA) filter with pulse-train or white-noise excitation.

4.2. Experimental results

The accuracy of the proposed approach for mapping was evaluated. The mel-cepstral distortion between the target and converted mel-cepstral coefficients [4] in the evaluation set was used as an objective evaluation measure.

First, the choice of covariance (kernel) functions and the effect of dynamic features were evaluated. Table 1 shows the mel-cepstral distortions between target speech and converted speech by the proposed GP-based mapping with various covariance functions and with and without using dynamic features. Please refer to [3] for details of these covariance functions. It can be seen from Table 1 that the linear covariance functions with ARD achieved the best performance and the use of dynamic features degraded the mapping performance. The authors expect that too much smoothing by the use of dynamic experts caused the degradation.

Table 1: Mel-cepstral distortions in dB between natural and converted speech by the GP-based VC with various covariance functions.

Covariance functions	w/o dyn.	w/ dyn.
Linear	3.96	**4.15**
Linear+ARD	**3.95**	**4.15**
Matern	4.96	5.99
Neural network	4.96	5.95
Polynomial	4.95	5.80
Piecewise polynomial isotoropic	4.96	6.00
Rational quadratic+ARD	4.96	5.98
Rational quadratic isotropic	4.96	5.98
Squared exponential+ARD	4.95	5.98
Squared exponential isotropic	4.95	5.98

Table 2: Mel-cepstral distortions in dB between natural and converted speech by GMM, and trajectory GMM.

#mix	GMM w/o dyn.	GMM w/ dyn.	Trajectory GMM
2	5.97	5.95	5.90
4	5.75	5.82	5.81
8	5.66	5.69	5.63
16	5.56	5.59	5.52
32	5.49	5.53	5.45
64	5.43	5.45	5.38
128	5.40	5.38	**5.33**
256	**5.39**	5.35	5.35
512	5.41	**5.33**	5.42
1,024	5.50	5.34	5.64

Next the proposed GP-based VC approach was compared with the conventional ones. Table 2 shows the mel-cepstral distortions by conventional VC approaches including GMM with and without dynamic features, and trajectory GMMs. It can be seen from Tables 1 and 2 that the proposed GP-based approaches with linear kernel and ARD priors achieved significant improvements over the conventional parametric approaches. A preliminary subjective listening test result showed that the use of dynamic experts in GP-based VC improved the naturalness of converted speech . Detailed subjective listening tests will be conducted.

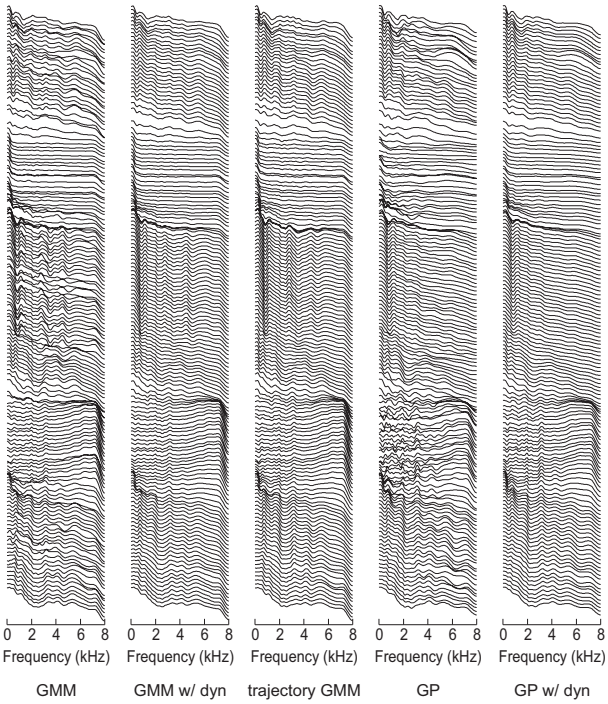

Figure 1: Running spectra of converted speech by GMM, trajectory GMM, and GP-based approaches.

Figure 1 shows the running spectra of converted speech by the conventional and proposed VC approaches. It can be seen from the figure that the GMM is excessively smoother compared to the GP approach but the GP-based approach was not suffered from this problem and looked maintains the fine structure of speech spectra.

5. Conclusions

This paper described an approach for VC using GP experts. GPs are robust to overfitting and oversmoothing and can predict the target spectra more accurately. Experimental results indicated that the performance of VC could be improved by GP experts in the objective measure.

Future work includes detailed subjective listening tests and evaluating other kernel functions.

6. References

[1] Y. Stylianou, O. Capṕe, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 2, pp. 131–142, 1998.

[2] A. Kain and M. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proc. ICASSP*, 1998, pp. 285–288.

[3] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[4] T. Toda, A. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. Acoust. Speech Lang. Process.*, vol. 15, no. 8, pp. 2222–2235, 2007.

[5] H. Zen, Y. Nankaku, and K. Tokuda, "Continuous stochastic feature mapping based on trajectory HMMs," *IEEE Trans. Acoust. Speech Lang. Process.*, vol. 19, no. 2, pp. 417–430, 2011.