EURASIP Journal on Applied Signal Processing 2001:1, 36–44

Radial Basis Function Networks for Conversion of Sound Spectra

Carlo Drioli

Università di Padova, Dipartimento di Elettronica e Informatica (DEI), Via Gradenigo 6/a, I-35131 Padova, Italy
Email: drioli@dei.unipd.it

Received 5 April 2000 and in revised form 18 January 2001

In many advanced signal processing tasks, such as pitch shifting, voice conversion or sound synthesis, accurate spectral processing is required. Here, the use of Radial Basis Function Networks (RBFN) is proposed for the modeling of the spectral changes (or *conversions*) related to the control of important sound parameters, such as pitch or intensity. The identification of such conversion functions is based on a procedure which learns the shape of the conversion from few couples of target spectra from a data set. The generalization properties of RBFNs provides for interpolation with respect to the pitch range. In the construction of the training set, mel-cepstral encoding of the spectrum is used to catch the perceptually most relevant spectral changes. Moreover, a singular value decomposition (SVD) approach is used to reduce the dimension of conversion functions. The RBFN conversion functions introduced are characterized by a perceptually-based fast training procedure, desirable interpolation properties and computational efficiency.

Keywords and phrases: sound transformations, sinusoidal representation, RBFNs, spectral processing.

1. INTRODUCTION

In the field of speech and audio processing a large number of applications have been proposed up to the present which realizes high-level transformations by combination of simpler effects like time-scale modification, pitch shifting, amplitude envelope modification, and spectral processing. Most of these applications are based on a sinusoidal representation of the signal. In this work we focus our attention on the spectral processing item and we stress the importance of an accurate representation of the spectrum and its characterization when modeling real data in both audio processing and synthesis applications. Among the most important and recent applications in which spectral processing is implied, time-scale and pitch modification have been widely explored, especially in the speech processing field, and the problem of correctly reproducing the spectral characteristics has been stressed [1]. Recently, a new spectral processing approach has been proposed by Stylianou et al. [2], where a conversion function was build from training examples and was used to convert the spectral features of a first speaker in the spectral features of a second speaker, who uttered the same sentence. Besides the field of speech processing, the sinusoidal modeling of sound mainly interested the computer music field. Analysis-based additive sound synthesis is effective due to the high quality of tones generated, and to the high degree of control. In the work by Horner and Beauchamp [3], additive synthesis based on the Short-Time Fourier Transform (STFT) analysis is used as the engine for sound generation purposes, and a dynamic filter is used to gain realistic results with respect to pitch and intensity variations. Among the other applications related to computer music, expressiveness processing of musical performance has recently gained an increasing interest. In [4, 5] the problem of controlling the high-level musical attributes of a recorded performance by means of expressiveness models and suitable sound processing techniques is faced.

This work proposes a new frequency-domain filtering model suitable for the sinusoidal representation of sound. The identification of the model parameters relies on a learning procedure based on collections of real data which represents, for example, the timbre identity of a given musical instrument. The method has proved to be useful in preserving the spectral characteristics of sounds processed by transformations such as pitch or intensity modification.

The paper is organized as follows. The sinusoidal sound analysis and resynthesis framework, as well as the mel-cepstrum representation of spectral envelopes, is briefly reviewed in the first part of Section 2. In the remaining part of Section 2, the structure to model the differences among spectral envelopes is introduced, and the main features of Radial Basis Function Networks, upon which the model relies, are reviewed. In Section 3, the construction of the training sets for the parametric identification of the RBFN model is shown with respect to some application examples, and a sin-

gular value decomposition approach is proposed to reduce the parametric dimension of the RBFN model.

2. SOUND ANALYSIS AND RESYNTHESIS FRAMEWORK

The investigation relies on the well-known sinusoidal model of the signal (here Spectral Modeling Synthesis, SMS) [6, 7]. The analysis algorithm acts on windowed portions (here called *frames*) of the signal, and produces a time-varying representation as sum of sinusoids (here called *partials* (the term *partials* generalizes the term *harmonics* and is used to underline the fact that both harmonic and non-harmonic signals, such as piano and bell-sounding tones, can be considered here)). Assuming that the number of partials H is constant for all frames, for the ith frame the result of the sinusoidal modeling is a set $\{(f_h(i), a_h(i), \phi_h(i)), h = 1, \ldots, H\}$ of triples of frequency, magnitude and phase parameters describing each partial, and a residual noise component that will not be considered in this work. H is taken sufficiently high to provide the maximum needed bandwidth, and zero magnitude is assigned to the exceeding partials for the spectra with lower bandwidth. The re-synthesis of sound can rely both on additive synthesis, or on the inversion of the analysis procedure, that is, on anti-transforming the frame analysis and overlap-and-adding the result with previous time-domain frames.

The sinusoidal representation allows to control some of the basic sound parameters, such as pitch and intensity, by simply shifting or scaling the frequency and magnitude of the partials. However, without an accurate spectral compensation which reflects the natural sound characteristics, the result of a transformation performed with a constant magnitude scaling is often unrealistic. The proposed spectral processing method relies on learning from real data the spectral transformations which occurs when such a musical parameter changes. With this perspective, a perceptually weighted representation of spectral envelopes is introduced in the next section, so that the perceptually relevant differences are exploited in the comparison of spectral envelopes.

2.1. Representation of spectral envelopes

To move from the original sinusoidal description to a perceptual domain, the original spectral envelope is turned to the *mel-cepstrum* spectral representation, by application of the discrete cepstrum method [8]: for a given sinusoidal parametrization, the magnitudes $\{a_h, h = 1 \cdots H\}$ of the partials are expressed in the log domain and the frequencies $\{f_h, h = 1 \cdots H\}$ in Hz are converted to mel frequencies $\{\lambda_h\}$ with the analytical formula $\lambda = \text{mel}(f) \approx 1127 \log(1 + f/700)$ [9]. The real mel-cepstrum parameters m_i $(i = 0, \ldots, M)$ are finally computed by minimizing the following least squares (LS) criterion

$$\sum_{h=1}^{H} \left(|C(\lambda_h)| - 20 \log_{10}(a_h) \right)^2 \tag{1}$$

with

$$|C(\lambda)| = m_0 + 2 \sum_{i=1}^{M} m_i \cos\left(\frac{\pi \lambda i}{2B_H}\right), \tag{2}$$

where M is the number of cepstral coefficients, m_0 is the frame energy, and $B_H = \min\{\text{mel}(f_H), \text{mel}(F_s/2)\}$ with F_s being the sampling frequency. The normalization factor B_H ensures that the upper limit of the band corresponds to a value of 1 on the normalized warped frequency axis. The aim of this transformation is to catch the perceptually meaningful differences among spectra by comparing the smoothed and warped versions of spectral envelopes (see Figure 1 for an example from a saxophone tone).

We call now $c_h = |C(\lambda_h)| = |C(\text{mel}(f_h))|$ the hth partial magnitude (in dB) of the mel-cepstrum spectral envelope, and $\Delta\mathbf{C} = \{\Delta C_h, h = 1, \ldots, H\}$, with $\Delta C_h = (c_h^{(2)} - c_h^{(1)})$, the difference between two mel-cepstrum spectral envelopes. By comparison of two different spectral envelopes is possible to express the deviation of each partial in the multiplicative form $r_h = 10 \exp[\Delta C_h/20]$, and we call *conversion pattern* the set $\{r_h, h = 1, \ldots, H\}$ generated by the comparison of two spectral envelopes.

2.2. Spectral conversion functions

In this section, the parametric model for the conversion functions is presented as well as the parameter identification principles. The conversion is expressed in terms of deviations of magnitudes, normalized with respect to the frame energy m_0, from the normalized magnitudes of a reference spectral envelope. The reference spectral envelope can be taken from one of the tones in the data set. If the tone in the data set are notes from a musical instrument, with a simple attack-sustain-release structure, we will always consider the sustain average spectral envelopes, where the average is generally taken on a sufficient number of frames of the sustained part of the tones. Once the spectrum conversion function has been identified, the reference tone can be seen as a source for the synthesis of tones with different pitch or intensity, and correct spectral behaviour. Figure 2 resumes the steps involved in the analysis-modeling-resynthesis process. Moreover, we are interested in keeping also the natural time-variance of the source tone, as well as its attack-sustain-release structure. To this purpose, we make the simplifying hypothesis that the conversion function identified with respect to the sustained part of notes can be used to process every frame of the source note. In other words, the law which describes the spectral behaviour of the sustained part of a note, is assumed to well describe the behaviour in the remaining attack and release part of the same note. This assumption has proven to be satisfactory in most cases, on the base of informal listening tests conducted on the processed tones. We further make the following assumptions on the structure of the conversion function:

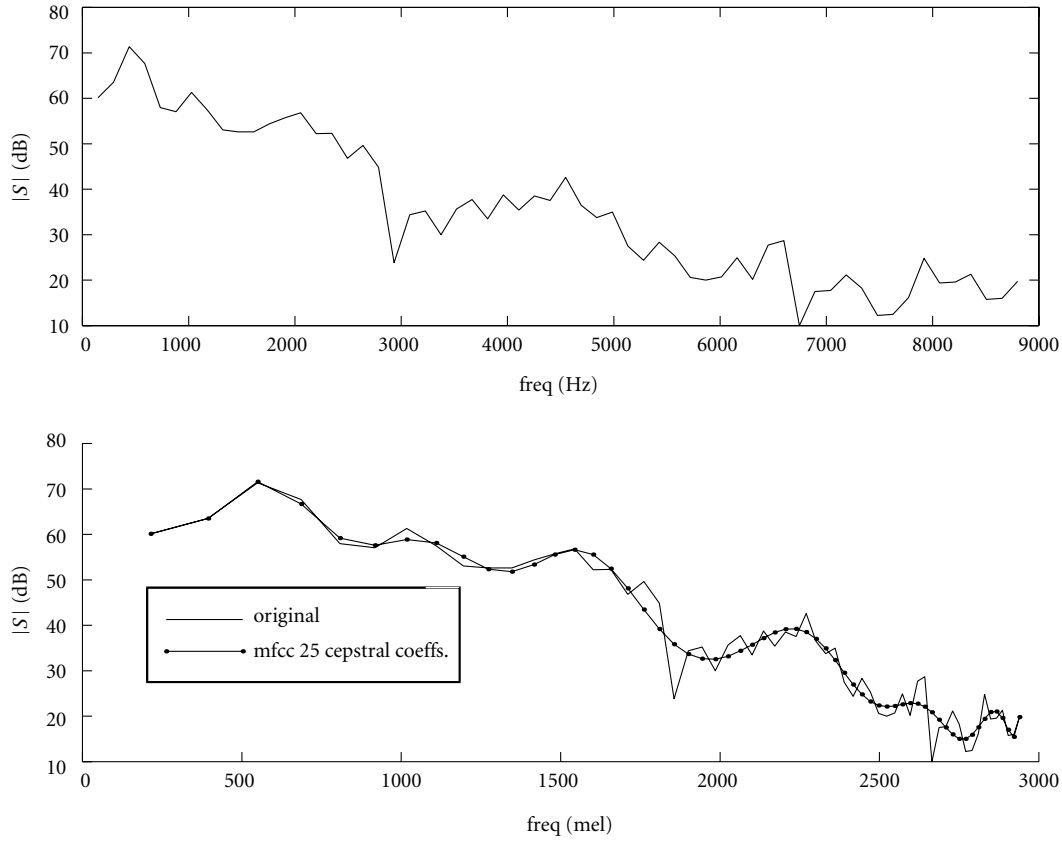


FIGURE 1: Sustain average spectral envelope of a saxophone tone (upper figure, frequency axis in Hz), and frequency warped mel-cepstrum envelope (lower figure, frequency axis in mel).

• Due to the changing nature of the spectrum with the pitch λ_0 of the tone, the conversion function is dependent on the pitch of the note. From the above consideration the function will then be a map $\mathcal{F} : \mathbb{R} \to \mathbb{R}^H$, where H is the maximum number of partials in the SMS representation.

• We adopt the following parametric form for the generic conversion function:

$$\mathcal{F}(\lambda_0) = \begin{bmatrix} \mathcal{F}_1(\lambda_0) \\ \vdots \\ \mathcal{F}_H(\lambda_0) \end{bmatrix} = \sum_{i=1}^{U} W_i G(\lambda_0; \mathbf{q}_i), \tag{3}$$

where $G(\lambda_0; \mathbf{q}_i)$ denotes a radial basis function with parameter vector $\mathbf{q}_i$, U is the number of radial basis units used, and $\mathbf{W} = \{W_{i,j}\}_{i=1\cdots U,\, j=1\cdots H}$ is a $U \times H$ matrix of output weights. The jth component of the conversion function, $\mathcal{F}_j(\lambda_0)$, can be made explicit as

$$\mathcal{F}_j(\lambda_0) = \sum_{i=1}^{U} W_{i,j} \cdot G(\lambda_0; \mathbf{q}_i) \tag{4}$$

and describes how the magnitude of the jth partial will adapt with respect to the desired fundamental frequency λ_0.

2.3. Radial basis function network

The parametric model introduced in (3) is known in literature with the name of *Radial Basis Function Network*, RBFN, and is a special case of feedforward neural network which exhibit high performances in nonlinear curve-fitting (approximation) problems [10]. Curve-fitting of data points is equivalent to finding the surface in a multidimensional space that provides a best fit to the training data, and generalization is the equivalent to the use of that surface to interpolate the data. The radial functions $G(\cdot;\cdot)$ in (3) can be of various kind. Typical choices are gaussian, cubic, sigmoidal functions. Here, a cubic form $G(x;\mu) = (\|x - \mu\|)^3$ is used. Now we face the problem of identifying the RBFN parameters. As usually needed by the neural networks learning procedures, the original data are organized in a training set. In our case, the pitch values of the training set notes are stored in the input training vector $\mathbf{T}_{\text{in}} = [\lambda_0^{(1)}, \ldots, \lambda_0^{(N)}]$, where each component corresponds to a row of the output matrix $\mathbf{T}_{\text{out}} = \mathbf{R}$, with

$$\mathbf{R} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,H} \\ r_{2,1} & & & r_{2,H} \\ \vdots & & & \vdots \\ r_{N,1} & r_{N,2} & \cdots & r_{N,H} \end{bmatrix}. \tag{5}$$

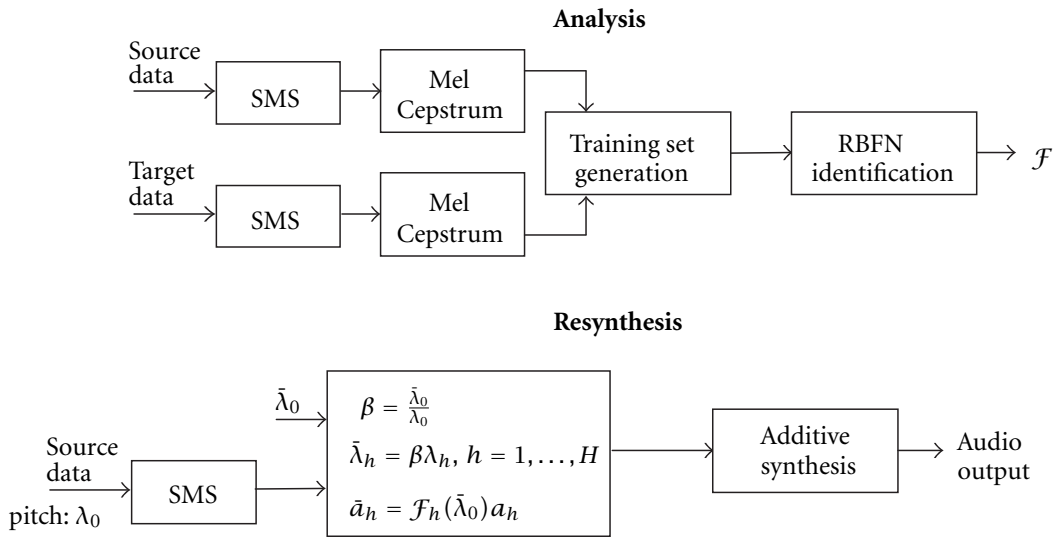


FIGURE 2: Block diagram of the analysis-resynthesis schema. The steps in the analysis task are: (1) representation of the source and target data by a mel-cepstrum based sinusoidal representation, (2) generation of a training set of conversion patterns, and (3) computation of the parameters of the RBFN model which represents the conversion function. The re-synthesis task requires: (1) to represent the source data by a sinusoidal model, (2) to compute at each frame the spectrum transformation, using the conversion function to compute the amplitude of partials, and (3) to synthesize frames by mean of additive synthesis or inverse-Fourier transform. Note that no mel-cepstrum representation is involved in the re-synthesis process, since the spectrum transformation is directly applied to the sinusoidal representation.

R is a matrix whose rows are the spectral envelope conversion patterns coming from the comparisons among the spectral envelopes from the source data and those from the target data. The way spectra are selected from both data set and the way they in case are processed before the comparison, strictly depends on the final high-level transformation to be realized. In the next section, a practical case will be treated to exemplify the training set generation procedure.

Here, we make the hypothesis that the training set has been computed with some strategy, which is part of the training set generation block of Figure 2, and we summarize the RBFN parametric identification procedure. The centers μ of the radial basis functions are iteratively selected with the OLS algorithm [11] which places the desired number U of units (with $U \leq N$) in the positions that best explains the data. Once the radial units with centers $\mu_1, \ldots, \mu_U$ have been selected, the image of $\mathbf{T}_{\text{in}}$ through the radial basis layer can be computed as $\mathbf{G} = [\mathbf{G}_1 \cdots \mathbf{G}_U]$, $\mathbf{G}_i = [G(\lambda_0^{(1)}, \mu_i) \cdots G(\lambda_0^{(N)}, \mu_i)]^T$ $(i = 1, \ldots, U)$. The problem of identifying the parameters $W_{i,j}$ of (4) can thus be given in the closed form $\mathbf{T}_{\text{out}} = \mathbf{G} * \mathbf{W}$, the LS solution of which is known to be $\mathbf{W} = \mathbf{T}_{\text{out}}\mathbf{G}^+$ with $\mathbf{G}^+$ pseudo-inverse of $\mathbf{G}$. As it can be seen, this parametric model relies on a fast learning algorithm, if compared to other well-known neural network models whose iterative learning algorithms are quite slow (e.g., backpropagation or gradient descent algorithms). To summarize the principal motivations why we adopted the radial basis function network model, we emphasize that the RBFNs can learn from examples, have fast training procedure, and have good generalizing properties, meaning that if we use a training set of N tones having pitch values of $\lambda_0^{(1)} < \lambda_0^{(2)} < \cdots < \lambda_0^{(N)}$, the resulting conversion function will furnish a coherent result in the whole interval $[\lambda_0^{(1)}, \lambda_0^{(N)}]$.

3. EXAMPLE OF TRAINING SET GENERATION PROCEDURE: A PITCH CONTROL MODEL

The proposed method is demonstrated in this section by using a conversion function to realize pitch transformations which preserves the spectral identity of a musical instrument. The procedure for the training set construction is now reviewed (see Figure 3). From a data set of N notes we want to construct N conversion patterns comparing the sustained spectral envelope of each note with that of the note selected as *source* note, whose pitch is modified each time to match the others. Referring to Figure 2, the source data is now the selected source note, and the target data is the whole data set (note that the resulting training set will include the all-zeros pattern, corresponding to the comparison of the source tone with itself). To this purpose, the SMS representation of the source note undergoes a modification which includes the scaling of the frequencies of partials, and optionally the interpolation of magnitudes to preserve its formant structure. This option gives the possi-

bility to use the a priori knowledge on the nature of sound to improve the identification process. Voice, for example, is known to be characterized by a formant structure which is, for a given vowel, approximately constant with respect to pitch variations. It is quite intuitive that, in such a case, preserving the formants can lead to a conversion pattern set with reduced magnitude range. We call *waveform preserving* the procedure where no formant preserving interpolation is performed, otherwise the procedure is called *formant preserving*.

In Figure 4, the two procedures are compared with respect to a set of voiced sung notes. In the first case, where the formant preserving procedure is not used, the frequencies of partials of the source note are shifted without changing the magnitudes. This implies that the formants shift as well, and that the conversion patterns need to restore the energy in the original position of formants with a positive contribute, as well as to attenuate, with a negative contribute, the energy in the positions where the formants moved due to the shift of partials. The use of a formant holding pitch shift procedure, which compute the new magnitude of partials by interpolation on the original spectral envelope and prevent the formants to move from their original position, lead to a conversion pattern set with reduced range of correction. This training set models only the residual differences, where no simple assumptions could have been made. We now treat the case where the target data is a set of seven saxophone notes ranging from a lower pitch of about 320 mel, to a higher pitch of 440 mel, and we want to build the conversion function to correctly reproduce the notes in the set by processing the sinusoidal analysis of the central note. The waveform preserving procedure was used to produce the training set, and in Figure 5, the conversion patterns and the result of the RBF network identification is shown for a set of seven saxophone notes. As previously recalled, the interpolating surface provides the best fit to the training data. The intersection of an orthogonal plane with this surface, for $\lambda_0 = \bar{\lambda}_0$, gives the correction of the magnitude of partials when the pitch of the source note is changed to $\bar{\lambda}_0$. The resulting spectrum is the one that best approximates the real spectrum with respect to the given data.

The use of the conversion function permitted to produce pitch shifted synthetic tones whose spectral envelope reflects that of the notes in the data set, at least in the sustained part of notes. To compare the synthetic tones with the real ones, we used the spectral centroid

$$f_{\mathrm{sc}} = \frac{\sum_{h=1}^{H} f_h \cdot a_h}{\sum_{h=1}^{H} a_h} \tag{6}$$

which is known to be a good index of spectral similarity. Figure 6 shows the effect of the conversion function used to correct the spectral envelope when the pitch of the saxophone source note is shifted.

3.1. Reduction of the parametric space

The conversion functions represent the behaviour of the sound spectrum in an original space whose dimension is equal in number to the number of partials used to describe the spectrum. It is quite intuitive that the number of variables involved is often redundant and should be reduced. To this purpose, singular value decomposition (SVD) is used.

Let $\mathbf{R}$ be the $N \times H$ matrix containing a conversion pattern in each row, one for each of the N notes in the data set (including the reference note). The singular value decomposition theorem states that $\mathbf{R}$ can be decomposed into the form

$$\mathbf{R}_{N\times H} = \mathbf{U}_{N\times N}\mathbf{S}_{N\times H}\mathbf{V}^{T}_{H\times H}, \tag{7}$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices. $\mathbf{S}$ is a $N \times H$ pseudo-diagonal matrix whose nonzero elements, called singular values, are nonnegative and by convention are given in decreasing order.

The singular values in matrix $\mathbf{S}$ are used to compute the rank of the decomposed matrix, which is the index of the last nonzero element. When the decomposed matrix is not square, as in our case, the rank of $\mathbf{S}$ will not be higher than the lower dimension (N), and a rank lower than N is indicated by an abrupt decrease of the magnitude between two adjacent nonzero elements in the diagonal. If we decide to use the first P components, the new set of target conversion paths will be given by

$$\hat{\mathbf{R}}_{N\times H} = \hat{\mathbf{U}}_{N\times P}\hat{\mathbf{S}}_{P\times P}\hat{\mathbf{V}}^{T}_{P\times H}, \tag{8}$$

where the unwanted columns and/or rows of the original matrices are not considered in the computation (note that for $P = N$, $\hat{\mathbf{V}}$ is a base for the space spanned by the rows of $\mathbf{R}$ and is $\hat{\mathbf{R}} = \mathbf{R}$). It should be noted that SVD is strictly related to principal component analysis (*PCA*), which is used to extract the axis (or factors) of the new space, where the higher amount of information is concentrated. Thus, choosing $P < N$ is the same as saying that we are satisfied of an approximated version of the training set. This approximation is as much accurate, as much as the variance explained by the first P principal components. Let $\mathbf{F} = \hat{\mathbf{U}}\hat{\mathbf{S}}$ be the $N \times P$ new matrix which represents the spectral conversion patterns, and let $\hat{\mathbf{V}}$ be the matrix to return to the initial conversion patterns: if we use the matrix $\mathbf{F}$ to train the RBFN, the dimensionality of the conversion function $\mathcal{F}$ is reduced from H to P with $P \leq N < H$ and the output of the RBFN will need to be multiplied by $\hat{\mathbf{V}}$ prior to its application to a spectral envelope (see Figure 7).

3.2. Multiple conversion functions and applications

Let now $\mathcal{D}_{\mathrm{freq}}(\lambda_0)$ be the conversion function identified following the procedure described in Section 3. The synthesis formula is then

$$\bar{a}_h = \mathcal{D}_{\mathrm{freq},h}(\lambda_0) \cdot a_h \tag{9}$$

and will produce the desired conversion of the spectrum each time a pitch shift is performed on the pitch of the original note. The same approach seen for pitch shifting can be used

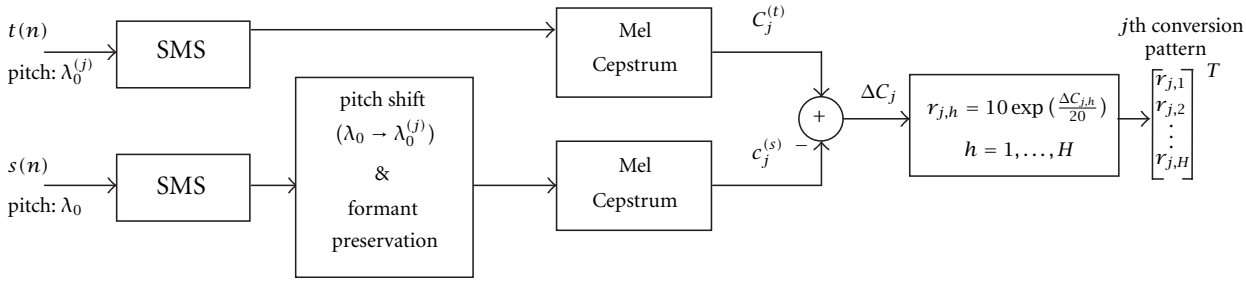
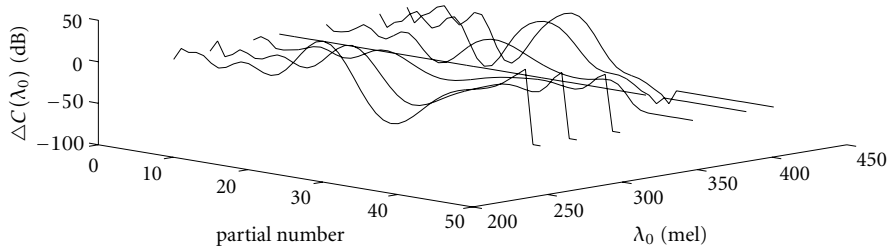
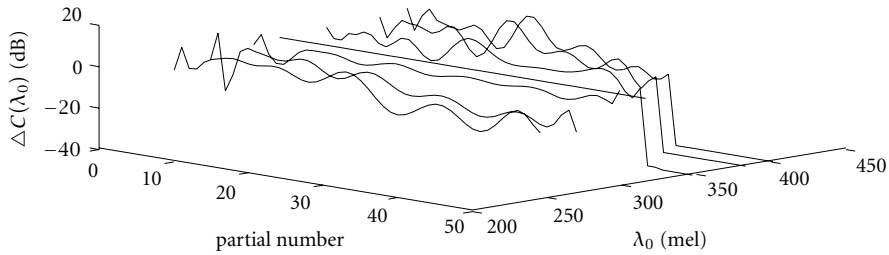


Figure 3: Schematic diagram of the computation of a conversion pattern to perform pitch shift with spectral correction.



(a) Waveform preserving conversion patterns.



(b) Formant preserving conversion patterns.

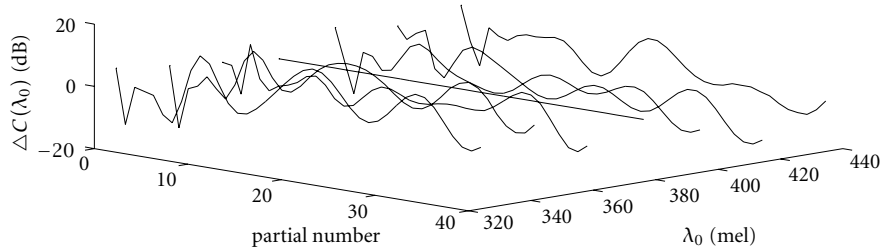
Figure 4: Conversion patterns generated from 7 voiced notes performing the same vowel: comparison between the waveform preserving procedure and the formant preserving procedure.

to control other sound parameters implying spectral correction, like intensity. Consider now the comparison of couples of tones having same pitch and different intensities, say I_m the minimum and I_M the maximum intensity. In this case we are interested in the construction of a conversion function which allows us to switch from the intensity of the source note to the intensity of the target note. No pitch shifting is now implied in the construction of the conversion pattern set. Say $\mathcal{D}_{\text{int}}(\lambda_0) = [\mathcal{D}_{\text{int},1}(\lambda_0) \cdots \mathcal{D}_{\text{int},H}(\lambda_0)]^T$ the conversion function that allows to switch from I_M to I_m. Note that $\mathcal{D}_{\text{int}}(\lambda_0)$ is still a function of frequency and not of intensity: we are in fact assuming that it turns the original note with intensity level I_M into a note with intensity level I_m, say on the opposite side of the dynamic range of the instrument. The only way to produce a tone with intensity level between I_M and I_m is thus to weight the effect of the conversion function. In this case, a simple interpolation can be used although one is not guaranteed on whether the model will reproduce or not the original spectral behaviour of the instrument with respect to changes of the intensity level. Let us define $\mathcal{D}'_{\text{int}}(\lambda, I) = \mathcal{D}_{\text{int}}(\lambda) \cdot \alpha(I)$, where the function $\alpha(I)$, ranging from $1/\mathcal{D}_{\text{int}}(\lambda_0)$, for $I = I_M$, to 1, for $I = I_m$, weights the effect of the conversion function. Then, the resynthesis formula that compute the new amplitudes for the intensity level $I \in [I_m, I_M]$ is

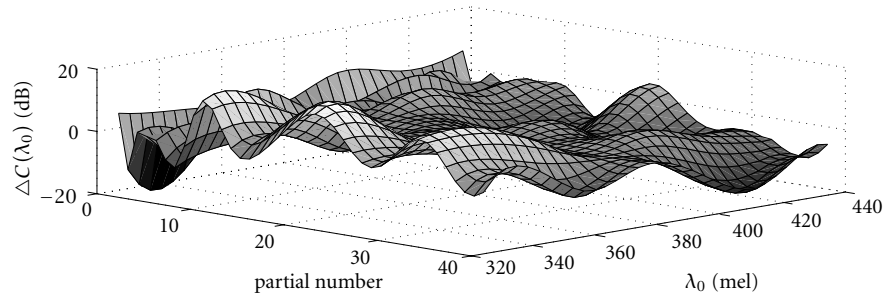
$$\bar{a}_h = \mathcal{D}'_{\text{int},h}(\lambda_0, I) \cdot a_h, \tag{10}$$

where a_h is the magnitude of the hth partial of a source tone. A logarithmic function for the function $\alpha(I)$ has shown to be suitable to perform an effective control on the range $[I_m, I_M]$.

Multiple conversion functions can be used at the same time to take into account different control parameters. This can be the case of simultaneous control of pitch and intensity



(a) Conversion patterns.



(b) Generalizing RBFN surface.

FIGURE 5: (a) The 7 waveform preserving conversion patterns resulting from 7 sax notes. (b) Interpolating surface provided by the RBFN.

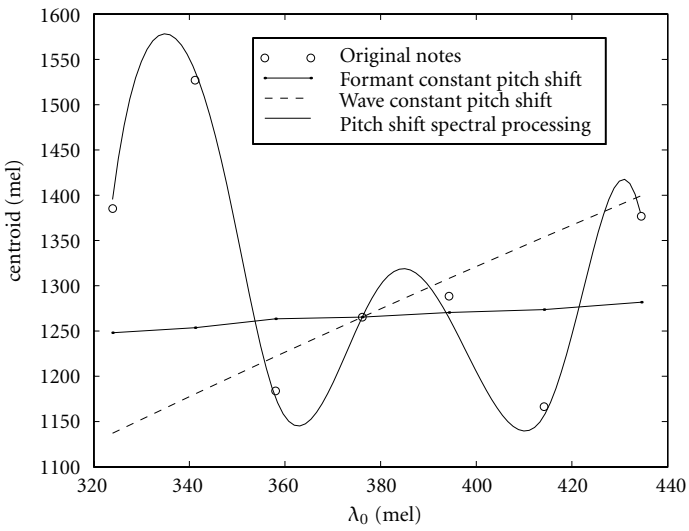


FIGURE 6: Spectral centroid: effect of the conversion function for pitch shifting. For a better understanding of the spectral processing importance, the centroid was computed when using the conversion function, as well as when omitting the effect of the conversion function and the pitch was shifted using the wave preserving procedure or the formant preserving procedure.

of a tone. If $\mathcal{D}_{\text{freq}}$ and $\mathcal{D}_{\text{int}}$ are two conversion functions relative to pitch and intensity, the synthesis formula will be

$$\bar{a}_h = \mathcal{D}_{\text{ampl},h}(\lambda_0) \cdot \mathcal{D}_{\text{int},h}(\lambda_0) \cdot a_h. \tag{11}$$

Again, a simple interpolation can be used although one is not guaranteed whether the model will reproduce or not the original spectral behaviour of the instrument with respect to changes of the intensity level:

$$\bar{a}_h(\lambda_0, I) = \mathcal{D}_{\text{freq},h}(\lambda_0) \cdot (\alpha(I) \cdot \mathcal{D}_{\text{int},h}(\lambda_0) + (1 - \alpha(I))) \cdot a_h \tag{12}$$

with $\alpha(I)$ being a logarithmic function assuming values in $[0, 1]$ as I changes from I_m to I_M. Of course a conversion function with two inputs (frequency and intensity) would be needed for a more complete and accurate model.

As a last application field that can benefit from this research, we cite the modeling and control of expressiveness in digitally recorded music performances [5, 12]. From the analysis of performances played with different expressive intentions, it was possible to understand which are the most important parameters on which the musicians rely to change the expressiveness of the performance. Among these, tempo, intensity, energy envelope, legato-staccato, and brightness are the most important. The proposed spectral processing method was used to learn the spectral features of the performing instrument from different performances of the same musical excerpt, so to catch the spectral differences occurring when playing *bright* instead of *dark*, or *heavy* instead of *light*. The resulting conversion functions were then used, together with other sound effects such as pitch shifting, time stretching

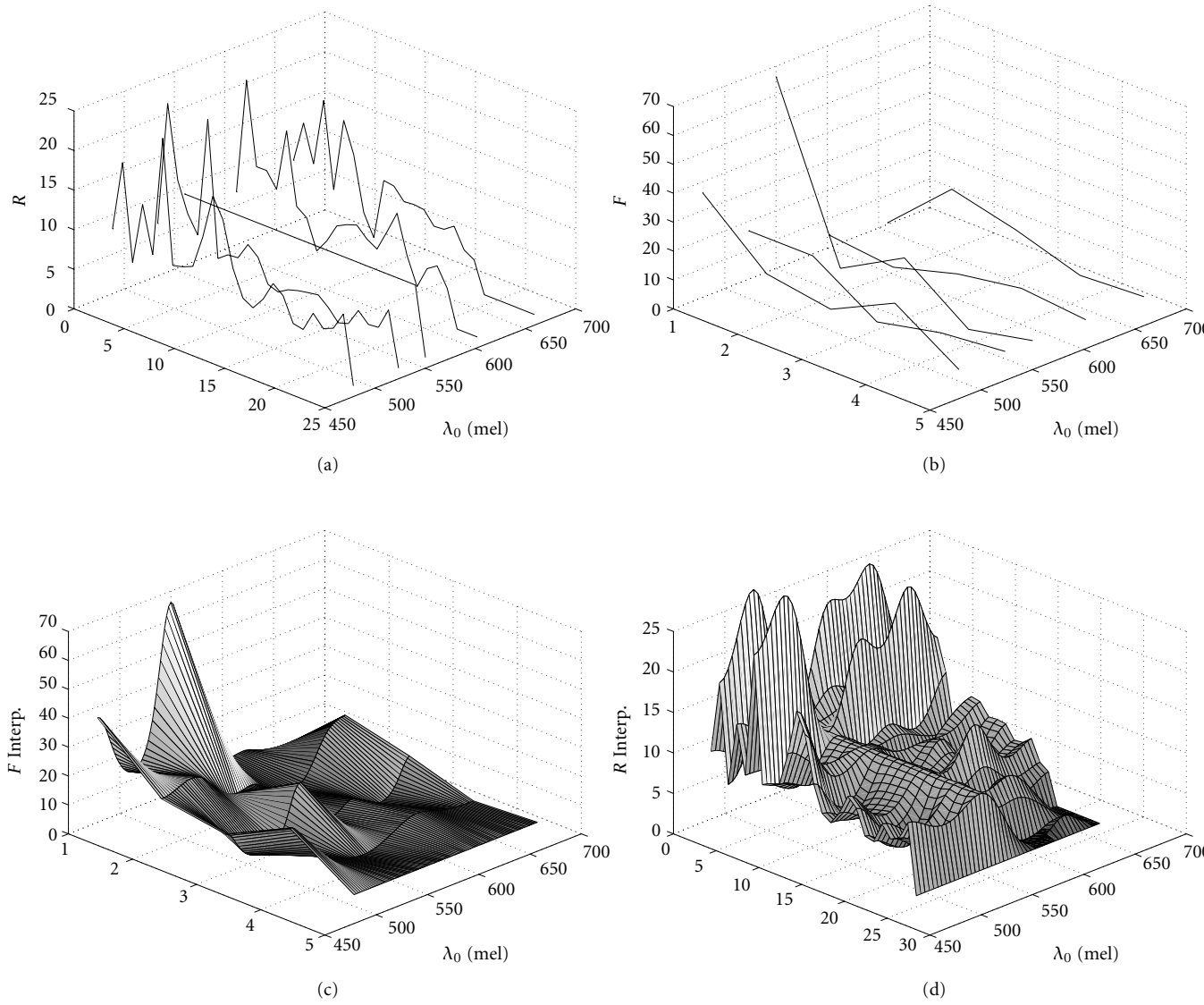


FIGURE 7: Reduction of the parametric model. The example is relative to a conversion function for pitch control, and the training data is made of five violin notes. Here, (a) is the original conversion patterns matrix R, (b) is the matrix F with $P = N = 5$, (c) is the interpolating surface for the matrix F and (d) is (c) after multiplication by the principal components matrix $\hat{\mathbf{V}}$.

and envelope control, to realize the sound transformations required by the model of expressiveness [4].

4. DISCUSSION AND CONCLUSIONS

A spectral processing model suitable for the sinusoidal representation of sound has been proposed. The identification procedure is characterized by a fast perceptually based learning procedure and the possibility of learning from sound examples has been stressed. Moreover, due to its low computational cost, the model is suitable for real time applications such as expressive processing or sound synthesis. The method has been applied to pitch shifting with spectral correction, and the spectral centroid of the synthesized sound has been compared with the spectral centroid of the real target sound, showing the effectiveness of this approach.

When notes with simple attack-sustain-release structure were considered, the simplifying assumption that a unique conversion function was sufficient to model the different part of the note has been made. Although informal listening tests showed that this assumption was satisfactory in most cases, a model with time-varying parameters would have been more adequate to fit the general case where attack, decay and different portions of sound presenting peculiar spectral characteristics. These aspects will be considered in future works.

5. ACKNOWLEDGEMENT

This work was supported by TELECOM ITALIA under the research contract Cantieri Multimediali.

REFERENCES

[1] T. F. Quatieri and R. J. McAulay, "Shape invariant time-scale and pitch modification of speech," *IEEE Trans. on Signal Processing*, vol. 40, no. 3, pp. 497–510, 1992.
[2] Y. Stylianou, O. Cappé, and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.
[3] A. Horner and J. W. Beauchamp, "Synthesis of trumpet tones using a wavetable and a dynamic filter," *J. Audio Engineering Society*, vol. 43, no. 10, pp. 799–812, 1995.
[4] S. Canazza, G. De Poli, R. Di Federico, C. De Drioli, and A. Rodá, "Symbolic and audio processing to change the expressive intention of a recorded music performance," in *Proc. of the 2nd COST G-6 Workshop on Digital Audio Effects (DAFx99)*, Trondheim, 1999, pp. 1–4.
[5] J. L. Arcos, R. L. de Mántaras, and X. Serra, "Saxex: A case-based reasoning system for generating expressive musical performances," *Journal of New Music Research*, pp. 194–210, 1998.
[6] X. Serra, "Musical sound modeling with sinusoids plus noise," *in Musical Signal Processing*, pp. 497–510, 1997.
[7] R. J. McAulay and T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-34, no. 4, pp. 744–754, 1986.
[8] O. Cappé, J. Laroche, and E. Moulines, "Regularized estimation of cepstrum envelope from discrete frequency points," in *Proc. of the IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, October 1995.
[9] E. Zwicker and E. Terhardt, "Analytical expressions for critical-band rate and critical bandwidth as a function of frequency," *J. of the Acoustical Society of America*, vol. 68, no. 5, pp. 1523–1525, 1980.
[10] S. Haykin, *Neural Networks. A Comprehensive Foundation*, Macmillan, New York, 1994.
[11] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis functions networks," *IEEE Trans. on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.
[12] S. Canazza, G. D. P. A. Rodá, and A. Vidolin, "Analysis and synthesis of expressive intentions in musical performance," in *Proc. of the International Computer Music Conference (ICMC'97)*, Thessaloniki, September 1997, pp. 113–120.

Carlo Drioli received the Laurea degree in electronic engineering from the University of Padova, Italy, in 1996 with a thesis on sound synthesis by physical modeling. Since 1996 he has been collaborating with the Centro di Sonologia Computazionale (CSC), University of Padova, as a researcher in the field of sound and voice analysis and processing. He is currently pursuing the Ph.D. degree in electronic engineering. His current research interests are in the field of signal processing, sound and voice coding by means of physical modeling, and neural networks applied to speech and audio.