Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 12 (2012) 182 – 187

Procedia Computer Science

Complex Adaptive Systems, Publication 2
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2012- Washington D.C.

Note and Timbre Classification by Local Features of Spectrogram

Erhan Guven[a,*], A. Murat Ozbayoglu[b]

[a]*The George Washington University, Computer Science Department, 20052 Washington D.C.*
[b]*TOBB University of Economics and Technology, Department of Computer Engineering, 06560 Ankara-Turkey*

Abstract

In recent years, very large scale online music databases containing more than 10 million tracks became prevalent as the fostered availability of streaming and downloading services via the World-Wide Web. The set of access schemes, or Music Information Retrieval (MIR), still poses several and partially solved problems, especially the personalization of the access, such as query by humming, melody, mood, style, genre, instrument, *etc*. Generally the previous approaches utilized the spectral features of the music track and extracted several high-level features such as pitch, cepstral coefficients, power, and the time-domain features such as onset, tempo, *etc*. In this work, however, the low-level local features of the spectrogram partitioned by means of the Bark scale are utilized to extract the quantized time-frequency-power features to be used by a Support Vector Machine to classify the notes (melody) and the timbre (instrument) of 128 instruments of General Midi standard. A database of 3-second sound clips of notes C4 to C5 on 7 sound cards using two software synthesizers is constructed and used for experimental note and timbre classification. The preliminary results of 13-category music note and 16-category timbre classifications are promising and their performance scores are surpassing the previously proposed methods.

Keywords: Music information retrieval; music note; timbre; spectrogram features; statistical learning

1. Introduction

Music reproduction in digital form is the current standard, enabling many applications targeted at end-users, the music industry, and professionals such as composers, performers, musicology researchers, *etc*. The set of strategies to access collections of digital music is called Music Information Retrieval (MIR) and has been under intensive research to develop better methods than the ones that rely on textual metadata only. Automatic discovery of high level music content descriptors and extracting low level audio features assist MIR use cases such as music identification; plagiarism detection; recognition of melody, composer, style, genre, mood, instrument, *etc*. High level music features (*i.e.* timbre, melody, bass, rhythm, pitch, harmony, key, structure, lyrics) are hard to extract [1]

* Corresponding author.
E-mail address: eguven@gwu.edu.

 doi:10.1016/j.procs.2012.09.051

and outperformed by the methods that employ low level audio features [1] which are measurements from the audio signal. Signal processing techniques such as Short Time Fourier Transform (STFT), constant-Q/Mel spectrum, pitch chromagram, onset detection, Mel-Frequency Cepstral Coefficients (MFCC), spectral flux, tempo tracking are among the many ways that are proposed to extract low level music features [1]. Though these low level features are considered to be more useful in general, the low precision, poor generalization, and loose coupling (to the underlying music aspect, timbre, melody, rhythm, pitch, structure, lyrics, *etc.*) of low level features make it a necessity to employ a second stage processing that can relate the low level features of the music to its content [2]. There have been several studies [3,4,5,6] which attempted the two MIR problems, automatic music note transcription and instrument detection. Some of these studies used the high level feature pitch, and some others used the low level audio feature MFCC, then using these features for classification by statistical learning algorithms. Though the experimental setups present a large variety in terms of number instruments, music type, polyphony, *etc.*, the highest achieved scores remain to be much less than 90% and 80% for note and timbre detection, respectively.

In previous studies by authors, the spectrogram of the audio was quantized by Bark scale [7] on the frequency axis and by user-variable parameters on the time axis to be used in a new speech feature extraction scheme. The method was applied to the speech emotion recognition problem and was shown the performances on two major databases surpass the state of the art [8]. In this study, a modified version of the feature extraction scheme, specifically further partitioning the frequency axis in relation to music note frequencies and adding the formants f_1 and f_2 to the feature vectors. This new feature extraction scheme and a Support Vector Machine (SVM) classifier [9] with Radial Basis Function (RBF) kernel is used for automatic recognition of the high level features of music, specifically the note and the timbre of the music. In addition, an improvement to the classification stage is proposed to improve the majority voting of the stream of feature vectors that are generated by a single musical sound clip.

2. Feature Extraction and Classification

The feature extraction starts with a spectrogram of the discrete time music signal $x[n]$ sampled at a frequency of f_S, and a segmentation of the spectrogram by means of Bark scale and user-set time-axis parameters. Given a set of three parameters, frequency resolution f_R, time resolution t_R, number of time slots n_{TS} and a window function $w[n]$, calculate the true power spectra $S[k,n]$ in decibels as in the following.

$$X[k,n] = \sum_{m=0}^{N-1} x\left[m - \left\lceil f_S t_R - 0.5 \right\rceil n\right] w[m] \mathrm{e}^{-i2\pi\frac{km}{N}}, \text{ where } N = \left\lceil f_S / f_R \right\rceil \tag{1}$$

$$S[k,n] = \frac{1}{f_S \boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}} 10 \log \left| X[k,n] \right|^2 \tag{2}$$

Choose a suitable f_R resulting a window length N in powers of 2, so that the Fast Fourier Transform (FFT) can be computed efficiently. Segment $S[k,n]$ by Bark scale to get $S_i[n]$, then calculate surface linear regression coefficients of $S_i[n]$ in order to assemble the feature vectors $\boldsymbol{V}[n]$. The details of the method can be found in [8].

$$\boldsymbol{V}[n] = \left[a_{1,n} \quad b_{1,n} \quad c_{1,n} \quad a_{2,n} \quad \ldots \quad b_{r,n} \quad c_{r,n} \right]^{\mathrm{T}}, \text{ where } r = \left| \mathbf{B}_{\mathrm{S}} \right| - 1 \tag{3}$$

The Bark scale is modified to include segments centered at frequencies of music notes ranged from C4 to C5.

$$\mathbf{B}_{\mathrm{S}} = [20\ 100\ 200\ 254\ 269\ 285\ 302\ 320\ 339\ 360\ 381\ 404\ 428\ 453\ 480\ 509\ 539\ 630\ 770\ 920\ 1080$$
$$1270\ 1480\ 1720\ 2000\ 2320\ 2700\ 3150\ 3700\ 4400\ 5300\ 6400\ 7700]^{\mathrm{T}} \tag{4}$$

In addition, two more features, corresponding to first and second formants, are calculated directly from the segmented spectrogram and added to the feature vector.

$$V_{r+1}[n] = \max\{c_{i,n}\}, i = 1,\ldots,r \text{ and } V_{r+2}[n] = \max\left\{\{c_{i,n}\} - V_{r+1}[n]\right\}, i = 1,\ldots,r \tag{5}$$

Given a set of training data points $\boldsymbol{X}$ and categories Y for each of the given data point.

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}, \boldsymbol{x}_i \in R^n;\ \boldsymbol{Y} = \{y_1, y_2, \ldots, y_m\}, y_i \in \Sigma;\ \Sigma = \{w_1, w_2, \ldots, w_c\}, w_i \in Z \tag{6}$$

There are two major reasons for picking the SVM as the classifier in this research. First, SVMs are not affected negatively by low number of data points when the attributes are high in number (curse of dimensionality, see [10]),

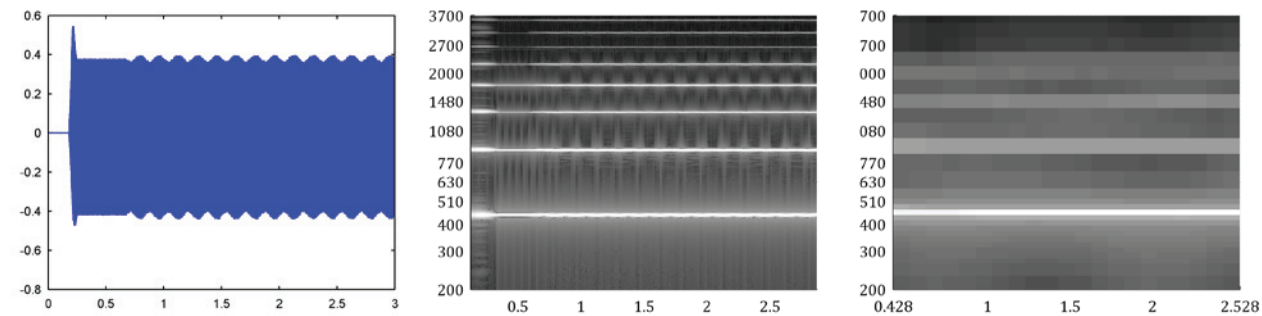
because they are designed to divide the space into partitions according to the category labels of the data points. Second, SVMs (also known as large-margin classifiers) avoid over-fitting the model to the data, as the margin distance between the support vectors and the imaginary hyperplane is expected to be maximized at the end of the SVM training [11]. Since the generated feature vectors are high dimensional (98 numerical attributes) and low in number (generated every 0.05 seconds or more), SVM is among the natural best classifier options in this study. Nevertheless, in pilot studies by authors, several classifiers from the Weka package [10] such as Naive Bayes, C4.5 decision trees, and nearest neighbor programs were greatly outperformed by the SVM program.

The multi-class SVM [9] maximizes the distances between the points belonging to category pairs $\{w_i, w_j\}$ to the corresponding dividing hyperplane Π_{ij}, where $i \neq j$. A winner-takes-all decision function F is the following.

$$\mathrm{F}(\boldsymbol{x}) = w_k \Leftrightarrow k = \arg\max_k \sum_{j=1}^{c} \mathrm{sgn}(\mathrm{dist}(\boldsymbol{x}, \Pi_{kj})) \tag{7}$$

After each feature vector is labeled with the predicted category by the decision function F($\boldsymbol{x}$), a majority voting decision function $\mathrm{G}_1(\boldsymbol{V})$ (as in [8]) takes place to decide the final category of the discrete-time signal of length L.

$$\mathrm{D}_1(n,k) = \begin{cases} 1 & \text{if} \quad k = \mathrm{F}(\boldsymbol{V}[n]) \\ 0 & \text{otherwise} \end{cases} \qquad \mathrm{G}_1(\boldsymbol{V}) = w_k \Leftrightarrow k = \arg\max_k \sum_{n=1}^{L} \mathrm{D}_1(n,k) \tag{8}$$



(a) Signal: amplitude vs. time (s)

(b) Spectrogram: frequency (Hz) vs. time (s)

(c) Average power (regression coefficient c)

(d) Slope of spectral power on the frequency

(e) Slope of spectral power on the time axis

Fig. 1.
(a) the flute instrument (GM patch 74), discrete time signal;
(b) the spectrogram of the signal with a 4096 point FFT and 0.995 overlapping;
(c), (d), (e) the regression coefficients calculated by the feature extraction parameters of f_R=4 Hz, t_R=0.1 s, and n_{TS}=7. Bark scale is expanded to include frequencies for notes C4 to C5 and the planar regression segments are visible in the diagrams. Each vertical strip is assembled to embody a single feature vector for classification.

Fig. 1. Demonstration of the feature extraction scheme on a flute sound clip

This decision mechanism can be further improved by taking into account the actual distance values which are already computed by the multiclass SVM for each feature vector and hyperplane. Define the distance-weighted majority voting decision function $\mathrm{G}_2(\boldsymbol{V})$ as in the following.

$$\mathrm{D}(n) = \sum_{j=1}^{c} \mathrm{dist}(\boldsymbol{V}[n], \Pi_{\mathrm{F}(\boldsymbol{V}[n])\,j}) \tag{9}$$

$$D_2(n,k)=\begin{cases} D(n) & \text{if} \quad k=F(\boldsymbol{V}[n]) \\ 0 & \text{otherwise} \end{cases} \qquad G_2(\boldsymbol{V})=w_k \Leftrightarrow k=\arg\max_k \sum_{n=1}^{L} D_2(n,k) \tag{10}$$

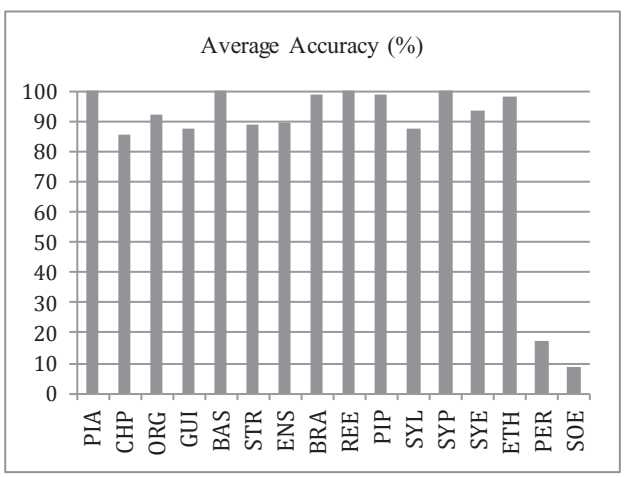
Figure 1 (previous page) demonstrates the feature extraction method on a flute sound clip. Each feature vector $\boldsymbol{V}$ is composed of 32 (from equation (3,4)) sets of three surface linear regression coefficients and 2 formants making the $\boldsymbol{V}$ of 98 dimensions. The first coefficient is the slope on the y-axis which corresponds to the amount of spectral power change in the frequency axis. The second coefficient is the slope on the x-axis which corresponds to the amount of spectral power change in the time axis. The third one is the z-axis offset of the plane which corresponds to the amount of spectral power in that segment, which is also equivalent to the spectrogram when it is segment-averaged. Consecutive feature vectors are generated with a period of t_R and assembled to represent the musical clip.

3. Experiments and Results

General Midi specification [12] lists 128 instruments in 16 groups which are listed in Figure 3 (next page). The experimental setup is based on these 128 instruments and their 13 notes from C4 to C5. These sounds are generated on multiple hardware (Asus Xonar DGX 5.1, Auzentech X-Fi Forte 7.1, HT Omega eClaro 7.1, SIIG DP SoundWave 5.1, Sound Blaster X-Fi Go Pro USB, Turtle Beach Amigo II USB, IDT HD audio Dell M4600), and by two synthesizers (Java JDK Gervill library and Windows 7 Roland GS driver). Each generated mono sound has a sampling frequency f_S of 16000 kHz, an onset of 0.2 s and a total duration of 3 s, making a total of 1664 clips.

A fair and robust measurement of the feature extraction and classification is supported by a 5-cross validation scheme where the training set is composed of 80% of the clips randomly picked without replacement and the remaining 20% for testing, cycling for 5 times. This iteration is repeated 100 times to achieve 95% confidence interval W of length 1 (*i.e.* [A-0.5, A+0.5] on percent accuracy A), assuming A has normal distribution and $\sigma_A < 3$. Clearly, a unique note from a unique instrument is used up for training or testing, once at a time, so the classification knowledge of a particular patch has to be learned from the other similar patches for all instruments and notes.

The detailed performance of music note detection experiment is displayed by the confusion matrix in Figure 2 below. The notes are detected with 100% accuracy for most instruments, except for the two groups which are percussion and sound effects. The confusion matrix shows the overall performance and predominantly the diagonal elements of the matrix reflect scores of these two failed groups. In these groups, all notes are somewhat equally confused signifying that the percussion and sound effects groups need special attention for music note detection, if it is really possible at all due to the fact that these sound instruments do not hold a reliable pitch [1].



GM128		REALITY												
		C4	C♯4	D4	D♯4	E4	F4	F♯4	G4	G♯4	A4	A♯4	B4	C5
PREDICTION	C4	**84**	3	2	1	1	2	1	0	1	1	1	2	3
	C♯4	4	**85**	4	1	1	0	2	1	0	1	1	1	0
	D4	3	3	**85**	3	2	2	1	1	0	0	2	1	2
	D♯4	0	1	3	**85**	3	1	1	1	2	1	2	0	0
	E4	0	1	0	2	**86**	3	1	1	3	2	1	1	1
	F4	0	1	1	1	1	**84**	4	3	1	0	2	1	1
	F♯4	0	1	1	0	1	3	**84**	4	2	1	1	1	0
	G4	2	1	1	0	1	1	2	**84**	2	0	0	1	3
	G♯4	1	2	0	2	1	0	2	2	**84**	3	1	0	1
	A4	2	0	0	1	0	1	1	2	2	**84**	4	2	1
	A♯4	0	0	1	1	1	1	0	1	1	3	**83**	3	1
	B4	0	1	0	0	2	1	1	0	1	1	3	**85**	5
	C5	3	1	1	0	0	1	0	1	0	1	1	3	**81**

Fig. 2. (a) average accuracy diagram; (b) confusion matrix of 13-class note detection experiment

Interestingly, though the notes of the two groups of percussion and sound effects are not recognized reliably, their timbre recognition is achieved with an accuracy of more than 95%, as can be seen in Figure 3. It is safe to assume that note detection and timbre recognition are quiet different problems, however, in this study they are attempted with the same feature extraction scheme but with different user parameters. The first experiment, note detection, uses the feature extraction parameters of $\{f_R$=1 Hz, t_R=0.1 s, n_{TS}=3$\}$, and the second one, timbre detection, uses $\{f_R$=1 Hz, t_R=0.5 s, n_{TS}=3$\}$. The only difference between the set of parameters is the time resolution. The longer duration the feature vector represents, the better its timbre detection performance gets.

Figure 3 below presents the detailed performance on 16-class timbre group detection where each group has 8 different instruments of the same family. As seen from the confusion matrix, the lowest performance is achieved with the synth and brass instruments. Naturally, most confusion occurs among similar sounding instruments. Guitar is confused with bass; reed is confused with brass and pipe; ensemble is confused with strings; *etc*.

GM128		REALITY															
		PIA	CHP	ORG	GUI	BAS	STR	ENS	BRA	REE	PIP	SYL	SYP	SYE	ETH	PER	SOE
PREDICTION	PIA	**93**	2	0	6	1	0	0	0	0	0	0	1	4	0	2	0
	CHP	1	**90**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ORG	0	0	**81**	0	0	0	1	3	11	3	2	2	3	0	0	0
	GUI	3	2	0	**73**	8	0	0	0	0	0	7	0	1	1	0	0
	BAS	0	0	0	10	**88**	2	0	0	0	0	0	0	1	0	0	0
	STR	0	0	0	0	2	**68**	9	3	1	13	4	0	0	4	3	3
	ENS	0	0	0	0	0	17	**69**	0	0	4	8	14	4	0	0	0
	BRA	0	0	2	0	0	1	2	**61**	17	1	3	6	8	11	0	0
	REE	0	0	6	0	0	2	1	20	**64**	9	0	0	1	1	0	0
	PIP	0	0	1	0	0	4	9	6	6	**61**	5	3	2	0	0	0
	SYL	0	0	8	7	0	2	0	2	0	6	**61**	3	1	7	0	0
	SYP	0	0	1	0	0	0	7	3	0	3	9	**63**	9	0	0	0
	SYE	1	0	0	1	1	0	3	1	0	0	1	8	**66**	0	0	0
	ETH	0	5	0	2	0	0	0	1	0	0	0	0	0	**74**	0	0
	PER	1	1	0	0	0	3	0	0	0	0	0	0	0	0	**95**	0
	SOE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**95**

Instruments

PIA Piano
CHP Chromatic Percussion
ORG Organ
GUI Guitar
BAS Bass
STR Strings
ENS Ensemble
BRA Brass
REE Reed
PIP Pipe
SYL Synth Lead
SYP Synth Pad
SYE Synth Effects
ETH Ethnic
PER Percussive
SOE Sound Effects

Fig. 3. Confusion matrix for 16-class timbre detection experiment

Each sound clip is recorded by generating the midi patch on a PC sound card, and its line output connected to its line input. To measure the effect on the performance with respect to differences in hardware, each set is compared to every other by training on one complete set and testing on the other. In all possible combinations the classification performance was resulted as 100% pointing out the subtle hardware effects such as volume levels, A/D and D/A converters, timing of the sound drivers, *etc.* have little negative effect on the performance. Similarly, the effect of the difference of two sound synthesizers (Gervill and GS) are compared by training on one set and testing on the other. Due to the differences of the actually generated sounds (especially percussion, synth, and effect sounds) the classification performance was 89% or less while some sounds (piano, strings) had perfect performance.

The improvement at the decision stage by distance-weighted majority voting is proven to be effective, improving the accuracy measurement by 1% or more with a confidence factor of 90%. The comparison of its performance to the majority voting only scheme is omitted here due to the space constraints. However, in experiments conducted by authors, the improvement was consistent and significant, and observed both in the experiments here and the repeated experiments of study [8].

4. Conclusions and Future Work

In this study, the proposed feature extraction method is shown to be more flexible and comprehensive, and can be utilized successfully by a machine learning algorithm for music note and timbre recognition. Unlike the previous methods, the features are not too focused on one aspect of the sound signal, but instead, employing the entire spectral average and derivatives at the same time. In detailed and extensive experiments, the same feature extraction scheme is used to recognize both note and timbre of the 128 instruments of General Midi specification by a custom database. Piano note detection achieved 100% accuracy (for comparison, study [1] cited 90%) and piano timbre detection achieved 93% accuracy (for comparison, study [1] cited 80%). In all experiments, the training of the classifier was with the full set of 128 GM patches and 13 notes, 1664 musical clips, emphasizing the generality and robustness of the overall approach. A set of 7 sound cards and 2 synthesizers were employed to measure the robust generalization performance of the method, and experiments resulted consistent 100% detection accuracy on different hardware and 89% accuracy on different synthesizers. Note that the differences between some sound patches of Gervill and GS were quite high that one might assume the timbres are different ultimately. Future studies might focus on these subtle differences and fine-tune the feature extraction parameters.

An improvement in the decision stage, distance-weighted majority voting, is proposed and observed to improve the previous decision method, majority voting scheme [8], by 1% or more accuracy with a 90% confidence factor.

Future work might measure the polyphonic note and timbre detection performance and focus on the feature extraction scheme to employ a pitch tracking system. Currently, the way features are extracted impose limits on the time resolution and it has to be greater than certain short durations (*e.g.* 50 ms or more for speech, see [13]) due to the ergodicity limits. Due to the lack of literature, a study must be performed to measure the theoretical and empirical ergodicity of music sounds, so these limits can actually be specified.

In order to reduce the number of timbre classes (from 16 to a smaller number) and improve its performance, a clustering stage can be employed after the first classification. Then, a second classification stage can be utilized for individual clusters to improve the overall performance.

The proposed feature extraction approach can be used for onset, key, harmony, mood and music emotion detection. Future work will include the research to tune the approach of this study to these important MIR problems. The feature extraction parameters used a frequency resolution of 1 Hz, resulting an FFT of 16384 points, corresponding to 1 s with an f_S of 16 kHz. Because the spectrogram is known to have a time-frequency ambiguity [14], while keeping the frequency precision high for note detection, the time precision cannot be kept high for onset detection. One possible solution is to use the Wavelet Transform and build a wavelet spectrogram or scalogram.

References

1. Casey, M., et.al., Content-based music information retrieval: current directions and future challenges, *Proc. IEEE*, 96(4):668–696, 2008.

2. Fu, G., et. al., A survey of audio-based music classification and annotation, *IEEE Trans. on Multimedia*, 13(2):303–319, 2011.

3. Cemgil, A. T., et. al., A Generative model for music transcription, *IEEE Trans on Audio, Speech, Lang. Proc.*, 14:679–694, 2006.

4. Hamel, S. W., and Eck, D., Automatic identification of instrument classes in polyphonic and poly-instrument audio, *Proceedings International Conference Music Information Retrieval (ISMIR09)*, Japan, 2009.

5. Ryynanen, M. P., and Klapuri, A. P., Automatic transcription of melody, bass line, *etc*, *Computer Music Journal*, 32(3):72–86, 2008.

6. Sun, J., et al., A music key detection method based on pitch class distribution theory, *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 15(3):165–175, 2011.

7. Zwicker, E., Subdivision of the audible frequency range into critical bands, *The Journal of the Acoustics Society of America*, 33, 1961.

8. Guven, E., and Bock, P., Speech Emotion Recognition using a backward context, *AIPR*, Washington D.C., 2010.

9. Chang, C.-C., and Lin, C.-J., *LIBSVM: a Library for Support Vector Machines*, 2001.

10. Witten, I. H., and Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd Edition, San Francisco, MK, 2011.

11. Hsu, C.-W., and Lin, C.-J., A comparison of methods for multi-class svm, *IEEE Trans. on Neur. Netw.*, 13(2):415–425, 2002.

12. Selfridge-Field, E. (editor), *Beyond MIDI*, The MIT Press 1997.

13. Paliwal, K., and Alsteris, L., Usefulness of phase spectrum in human speech perception, *Eurospeech 2003*, Geneva, pp. 2117–2120.

14. Cohen, L., *Time Frequency Analysis: Theory and Applications*, Englewood Cliffs, Prentice-Hall, 1995.