AN AUTOMATIC TRANSCRIPTION SYSTEM WITH OCTAVE DETECTION

Yu-Ren Chien and Shyh-Kang Jeng

National Taiwan University
Graduate Institute of Communication Engineering and
Department of Electrical Engineering
Taipei 10617, Taiwan, ROC
E-mail: skjeng@ew.ee.ntu.edu.tw

ABSTRACT

Octave detection has been an open issue in automatic transcription of polyphonic music over the years. In the literature, pitch detectors for polyphonic music usually fail to detect octaves for lack of information about the timbre of each instrument that appears in the music. To reveal the possibility that this difficulty in automatic music transcription can be overcome, here a music transcription system that is capable of octave detection is proposed. Preceded by a constant-Q time-frequency analysis front end implemented with a fast algorithm for nonorthonormal discrete wavelet transform [1], a classifier using support vector machine technique [2] serves as a novel octave detector in this system.

1. INTRODUCTION

In the 1990's, the advent of compact discs brought sound recording technology to a brand-new age. Sound quality on a record was greatly improved. On the other hand, the development of Internet has made a tremendous change to our style to access various information sources. The transmission of real-time streaming audio has been developed; however, its quality still falls far behind that of CD recordings and is thus not enjoyable at all. One solution to this is to widen the bandwidth of the network connections. In addition, we can provide another solution from the following viewpoint. Our current method for sound recording and transmission can, in fact, deals with any audible sounds. Nevertheless, sounds in most records fall within a limited scope of musical instruments. If we can keep track of these sounds compactly, i.e., in the analyzed and meaningful form instead of physical waveforms, sound recording and transmission will undoubtedly become far more efficient. This has motivated researchers to work on methods to extract musical contents from acoustic signals, i.e., to automatically transcribe polyphonic music, over the years. Another reason to symbolically represent music is that such representations facilitate further intelligent processing of musical contents, such as harmonization, transposition, or melody search.

The most basic topic in music recognition is pitch recognition, for pitch is roughly our first perception on hearing a musical sound. Most musical sounds may have more than one pitch at any instant; therefore, a useful pitch recognition strategy should take polyphonic cases into account. The main difficulty in pitch recognition of polyphonic music lies in the decision to tell whether a perfect octave (or perfect twelfth, perfect fifteenth, etc.) harmonic interval exists. In [3], a general and well-organized system was established without focus on octave ambiguities. Martin [4] adopted a sophisticated front end based on an auditory model and explicitly pointed out this issue of octaves; however, without a timbre model of the instrument, the beating phenomenon noted in [4] does not suffice to robustly resolve the problem. Rossi *et al.* [5] proposed a feasible method to detect octaves, while it relies on the property of inharmonicity [6], which only piano and string pizzicato possess, and does not work well for weak notes.

Since pitch corresponds to frequency in the acoustical context, a pitch recognition system incorporates a subsystem of time-frequency analysis (STFT or constant-Q transform [7]) as its front end, where spectral analysis is done for a sequence of instants. For each instant, harmonically related spectral peaks are grouped into series, each of which represents a periodic component of the windowed signal. Note that a periodic component may correspond to more than one pitch, in that, for example, an octave interval contains two different pitches but produces only one harmonic series in the spectrum. If the timbre of a periodic component is ignored by a recognition system, then no inference can be made about the existence of an octave, and a note at which a beautiful melodical line forms a perfect 8th with a lower voice is subject to loss in the recognition result. This is how octave errors occur and dissatisfy music-lovers who test such a system with their favorite recordings.

In view of this, a novel recognition system is presented. It "listens" for the result of harmonic-series overlap / superposition in the timbre. Section 2 formulates the problem considered in this paper. The design of such a transcription system to solve the problem is presented in Section 3. Implementation details and experimental results are described in Section 4. A few discussions pertaining to the result and possible further extensions of this system conclude this paper.

2. PROBLEM FORMULATION

2.1. Acoustical Model of Musical Signals

Each note has a fixed pitch from onset to offset. All spectral partials of the pitch have magnitudes within ten times that of the fundamental and frequencies within 35 cents from harmonicity.

2.2. The Transcription Problem

Given any duration of digitized piano recording with certain limitations, the system is required to report the pitch, the onset, and the offset of each note in the recording. The limitations are listed as follows: (a) no pedaling; (b) only keys within the range G1–C8 (A4 = 440 Hz; most of the keys around C1 have missing fun-

0-7803-7402-9/02/$17.00 ©2002 IEEE

damentals) are played; (c) except perfect octave where two notes share one onset / offset, perfect 12th, perfect 15th, major 17th, etc., are not played.

3. SYSTEM DESIGN

As mentioned in Section 1, spectral peaks are grouped into harmonic series, each of which bears in its timbre, i.e., relative magnitudes and phases among the partials, feature for octave detection. Since the timbre of a single note not only takes on several degrees of freedom in itself, but also varies among pianos and with pitch, intensity and touching, it is not feasible to work out a rule for octave detection simply by one's knowledge and observations. This has prompted us to take the detection problem as pattern recognition from the viewpoint of machine learning. The octave detector is to be trained to make correct decisions. While neural networks have been successful in simulating human learning behavior and, in particular, chord recognition [8], an evolving technique called support vector machine (SVM) [2, 9] is adopted here. The SVM can be viewed as a neural network that is designed automatically to meet the optimality derived in statistical learning theory. It is currently receiving more and more attention and finds various applications in the literature [10].

Figure 1 shows the block diagram of the proposed automatic transcription system. Each part of the system is described in detail in a subsection below.

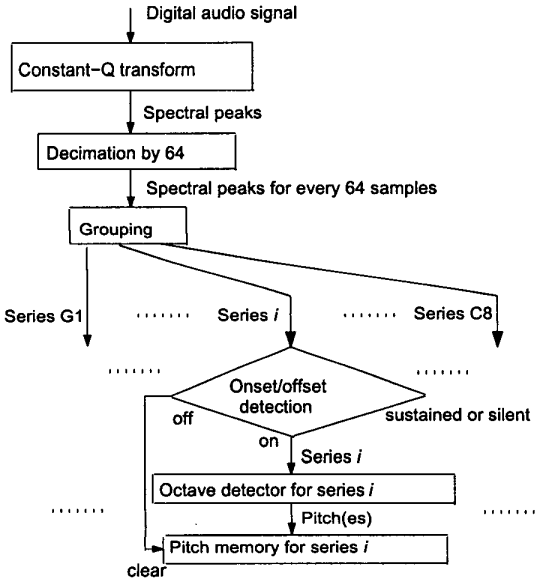


Fig. 1. System block diagram.

3.1. Constant-Q Transform [7]

The discrete-time constant-Q transform of signal $x[n]$ with $Q = 34$ is

$$X(\omega, n) = \omega \sum_k x[k] w((k-n)\omega) e^{-j\omega k},$$

where ω is the digital frequency, n is the instant of observation, and $w(\cdot)$ can be any bell-shaped window function that is nonzero only in the interval $[-34\pi, 34\pi]$. The transform is evaluated at 192 quartertone-spaced analog frequencies ranging from 21.8 Hz to 5.43 kHz (sampling frequency: 44100 Hz), sufficient to cover the whole register of the piano. Dividing these frequencies into eight octaves, the highest octave of the spectrum is evaluated every sampling instant, the second highest every two, the third every four, and so forth. The resulting sequence of spectra is decimated by 64, which is a trade-off between the temporal accuracy of transcription and the data rate throughout the rest of the system.

For every 64 samples, we have one spectrum, from which a set of peaks are picked out. A relative maximum on the magnitude spectrum is recognized as a peak if and only if the sharpness at that frequency, defined as the relative maximum divided by the average value over the neighboring five frequencies, exceeds 1.7. Consider a 10^5-point periodic signal with 20 equal-magnitude partials:

$$x[n] = (u[n] - u[n - 10^5]) \sum_{k=1}^{20} sin(0.0045 k\pi n).$$

The magnitude spectrum observed at $n = 5 \cdot 10^4$, $|X(\omega, 5 \cdot 10^4)|$, is plotted in Fig. 2, where the higher the order is, the less sharp the partial is. The threshold 1.7 is set to ignore partials of orders greater than ten. Since these high-order partials generally do not differ from one another in magnitude so much as to produce sharp peaks, the grouping of partials can be reasonably limited to order ten without mistaking any higher partial for another pitch.

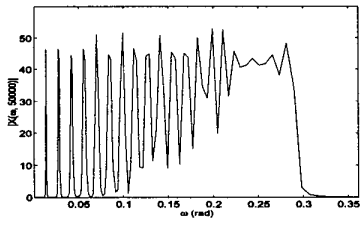


Fig. 2. Illustrating the effect of sharpness threshold. Partials of orders greater than ten do not give rise to peaks.

3.2. Grouping of Peaks and Onset / Offset Detection

The peaks found in a spectrum are grouped into harmonic series of which each satisfies the acoustical model described in Section 2.1 and does not share the peak at its fundamental with a lower-pitched series. Other hypothetical harmonic series are set to zero. The total power of each series is computed (summing over the partials); all those having total power less than $\frac{1}{225}$ of the greatest are set to zero. This heuristic coincides with the masking effect in psychoacoustics.

By monitoring the total power of each series, detection of onset or offset is performed. An instant is declared to be an onset of a series if the power rises from zero, an offset if the power falls below 0.01 times the maximum, both (an offset and then an onset, i.e., repetition of note) if the square root of the power increases from a nonzero value by more than 1.25, and no state change otherwise.

3.3. The Support Vector Machine [2, 9] and Octave Detection

A machine for M-by-1 pattern (feature) vectors is characterized by two parameters, α and b, where b is a scalar and α is a column

vector with number of elements the same as that of training data, N. The vector α minimizes

$$\frac{1}{2}\alpha^T \mathbf{H}\alpha - \sum_{i=1}^{N}\alpha_i$$

subject to the constraints

$$\alpha_i \geq 0, i = 1 \ldots N; \mathbf{y}\alpha = 0,$$

where

$$H_{ij} = (\mathbf{y}^T\mathbf{y})_{ij}(\mathbf{X}^T\mathbf{X})_{ij}^p, i, j = 1 \ldots N,$$

$\mathbf{X}$ is an $M \times N$ matrix made up of columns of training patterns, $\mathbf{y}$ is a $1 \times N$ vector consisting of binary variables showing the existence of octave interval for each training pattern (1 for existence and -1 otherwise), and p is the degree of the polynomial kernel. The scalar b is the mean value of the vector $(\mathbf{y}_S - \mathbf{lR})$, where

$$(\mathbf{l}_{1\times I})_i = \alpha_{S_i} y_{S_i}, R_{ij} = ((\mathbf{X}_S^T\mathbf{X}_S)_{ij} + 1)^p, i, j = 1 \ldots I,$$

α_S [$\mathbf{X}_S$, $\mathbf{y}_S$] is obtained by removing the zero elements [corresponding columns, corresponding elements] of α [$\mathbf{X}$, $\mathbf{y}$], and I is the length of α_S. The decision of the machine for a pattern $\mathbf{x}$ is

$$\begin{cases} 1 & \text{if } f(\mathbf{x}) > 0 \\ -1 & \text{if } f(\mathbf{x}) < 0 \end{cases},$$

where $f(\mathbf{x}) = \mathbf{lr} + b$, and $r_i = ((\mathbf{X}_S^T\mathbf{x})_i + 1)^p, i = 1 \ldots I$. The patterns in $\mathbf{X}_S$ are called support vectors.

Given a harmonic series, its timbre, which varies with time, reflects whether an octave exists or not. In order to detect octaves for harmonic series of various durations, the detector extracts as features the partial values within a short period of time following the onset. The resulting distribution of the features is depicted in Fig. 3. Since the overall intensity is irrelevant, the complex value of each feature is further normalized to the fundamental. Then, exclusive of the fundamental, we have totally 44 features, whose real and imaginary parts are separated to form a real 88-tuple. This is the pattern to be passed to the SVM. Since neighboring notes on the piano have similar timbres, only the SVM's for C2, C3, ..., C7 are trained. Each SVM is shared by neighboring harmonic series for octave detection.

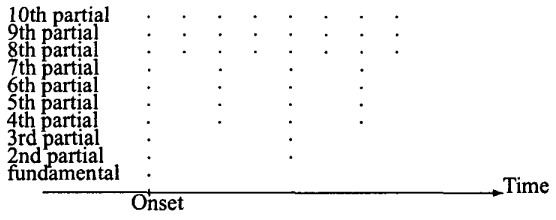


Fig. 3. Time-frequency distribution of the extracted features. The instants at which each partial is observed are marked with solid circles.

4. IMPLEMENTATION AND RESULTS

The entire system is implemented in MATLAB 6.0 on a Pentium III 800-MHz PC. Concerning the implementation of constant-Q transform, a factor that makes its computation time-consuming is that the windows for low-frequency analysis are extremely long. By representing the transform as 24 nonorthonormal wavelet transforms, the à trous algorithm [1], which is a fast algorithm for nonorthonormal wavelet transforms, helps speed up the computation. With the 192 analysis frequencies denoted by

$$\{\omega_k 2^{-i}\}, k = 1 \ldots 24, i = 0 \ldots 7,$$

where ω_k's are the highest 24 frequencies, the transform is computed by the filter bank depicted in Fig. 4. The FIR filter $H_k(z)$, whose impulse response is given by

$$h_k[n] = w(-\omega_k n)e^{-j\omega_k n},$$

is implemented using an overlap-save method with 2048-point FFT. The filter $G(z)$ performs linear interpolation with impulse response

$$g[n] = \frac{1}{\sqrt{2}}\delta[n] + \frac{1}{2\sqrt{2}}\delta[n+1] + \frac{1}{2\sqrt{2}}\delta[n-1].$$

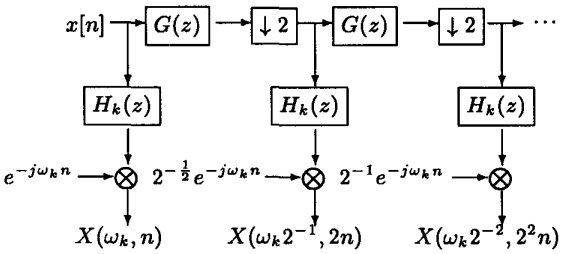


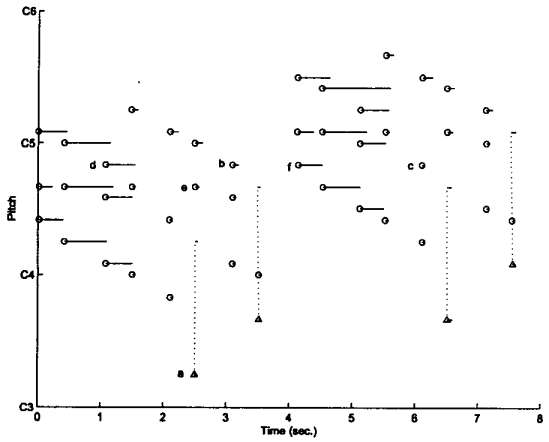
Fig. 4. The filter-bank structure that enables efficient computation of the constant-Q transform.

The polynomial kernel of the SVM has degree 2, i.e., $p = 2$. The vector α is computed using the (medium-scale) quadratic programming package in the optimization toolbox. An SVM is trained by successively accumulating failure patterns as its training data. The patterns are extracted from a digital-piano recording with notes of various MIDI velocities ranging from 50 to 100. The rate of convergence can be observed in the training progress shown in Fig. 5, where each black bar represents a testing failure and each white bar represents a success. In the end of each progress, most decisions of the SVM are shown to be correct.

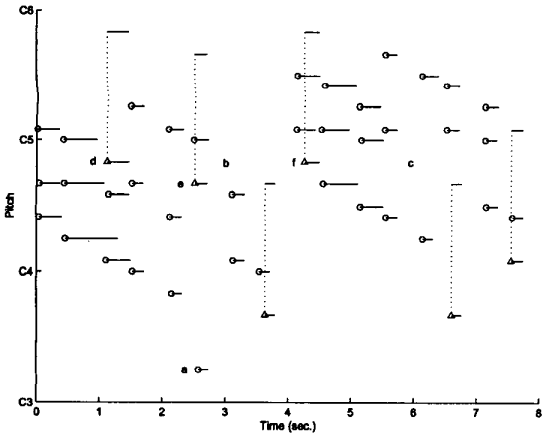
Fig. 5. The training progresses of harmonic series C3 and C4. Each progress goes from left to right with 120 patterns represented by bars. White bars represent successes while black ones represent failures, at which the SVM is re-trained.

To show a typical result using our system, an eight-second testing music and its transcription result are both plotted in Fig. 6 for comparison. The numerical representation of the testing music is obtained from the MIDI output of the digital piano. The onset of each note is marked with nothing if it is the upper note of an octave, with a triangle if it is the lower note of an octave, and

with a circle if it is not associated with any octave. Each octave is marked with a dotted line connecting both notes. Errors are labeled *a, b, c,* etc. Table 1 is a confusion matrix constructed for the 46 harmonic series involved in this experiment. Three out of four 8ths are detected, and only three out of forty single notes are misreported to be 8ths. If the information of timbre were not used, i.e., it were not for the SVM octave detector, the confusion matrix would be the one shown in Table 2, with all the 8ths, two of which lie in the main melodical line, left undetected and no single note misreported to be an 8th. Such shortcomings from lack of octave detection get more severe as the occurrences of 8ths increase. The transcription took about 10 minutes of computation. It can be accelerated if the program is translated into a C / C++ code.



(a) Testing music



(b) Transcription

Fig. 6. An experimental result.

5. CONCLUSIONS

Octave detection has been shown to be feasible under the condition that the music contains only one instrument. Detection of perfect 12th, perfect 15th, etc. can also be included by constructing an SVM for each additional function and properly assigning the training data to cover all possible sorts of timbres. In the case of ensemble music, where the same pitch may be played by different instruments, the proposed method can be further extended to detect a certain note played by a certain instrument, e.g., the 12th played by the violin.

Transcribed to Be	Octave	Note	Invalid
Octave	3 series	Series *d*, *e*, and *f*	0 series
Note	Series *a*	37 series	0 series
Invalid	0 series	Series *b* and *c*	—

Table 1. Confusion matrix for the result.

Transcribed to Be	Octave	Note	Invalid
Octave	0 series	0 series	0 series
Note	4 series	40 series	0 series
Invalid	0 series	Series *b* and *c*	—

Table 2. Confusion matrix without SVM octave detection.

6. REFERENCES

[1] Mark J. Shensa, "The discrete wavelet transform: Wedding the à trous and mallat algorithms," *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992.

[2] Vladimir N. Vapnik, *The nature of statistical learning theory*, Springer, New York, 2nd edition, 2000.

[3] Kunio Kashino, Kazuhiro Nakadai, Tomoyoshi Kinoshita, and Hidehiko Tanaka, "Organization of hierarchical perceptual sounds," in *Proceedings of the 14th IJCAI*, 1995, vol. 1, pp. 158–164.

[4] Keith D. Martin, "Automatic transcription of simple polyphonic music: Robust front end processing," Tech. Rep. 399, M.I.T. Media Laboratory Perceptual Computing Section, 1996.

[5] L. Rossi, G. Girolami, and M. Leca, "Identification of polyphonic piano signals," *Acustica*, vol. 83, pp. 1077–1084, 1997.

[6] Judith C. Brown, "Frequency ratios of spectral components of musical sounds," *J. Acoust. Soc. Am.*, vol. 99, no. 2, pp. 1210–1218, Feb. 1996.

[7] Judith C. Brown, "Calculation of a constant Q spectral transform," *J. Acoust. Soc. Am.*, vol. 89, no. 1, pp. 425–434, Jan. 1991.

[8] Borching Su and Shyh-Kang Jeng, "Multi-timbre chord classification using wavelet transform and self-organized map neural networks," in *Proceedings of the 2001 IEEE ICASSP*, 2001, vol. 5, pp. 3377 –3380.

[9] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[10] Daniel J. Sebald and James A. Bucklew, "Support vector machine techniques for nonlinear equalization," *IEEE Transactions on Signal Processing*, vol. 48, no. 11, pp. 3217–3226, November 2000.