DEVELOPMENT OF MIDI ENCODER TOOL "AUTO-F" FOR GENERAL TIME-BASED ELECTRIC SIGNALS

Toshio Modegi

Advanced Technology Laboratory, Research & Development Center, Dai Nippon Printing Co., Ltd.
250-1, Wakashiba, Kashiwa-shi, Chiba 277-0871, Japan
Modegi-T@mail.dnp.co.jp

ABSTRACT

The MIDI interface is originally designed for electronic musical instruments but we consider this music-note based coding concept can be extended for general acoustic signal description. We proposed applying the MIDI technology to coding of bio-medical auscultation sound signals such as heart sounds for retrieving medical records and performing telemedicine. Then we have tried to extend our encoding targets including vocal sounds, natural sounds and electronic bio-signals such as ECG, using Generalized Harmonic Analysis method. Currently, we are trying to separate vocal sounds included in popular songs and encode both vocal sounds and background instrumental sounds into separate MIDI channels. And also, we are trying to extract articulation parameters such as MIDI pitch-bend parameters in order to reproduce natural acoustic sounds using a GM-standard MIDI tone generator. In this paper, we present an abstract algorithm of our developed MIDI software encoder tool, which can convert any kind of electronic signals to MIDI-controllable interactive audio contents.

1. INTRODUCTION

MIDI (Musical Instrument Digital Interface) is originally designed for musical instrument, and we are considering MIDI as an ideal coding method because of its coding efficiency and high-quality sound reproduction capability. The first application of MIDI technology was synthesizing and generating rhythmic pattern signals substituting such as drum instrument set. Therefore, the first trial of perception of music signals was also for rhythmic patterns [1]. As features of MIDI coding are similar to those of text formats, if it is applied to audio databases, we can retrieve audio contents by audio keywords or music-note strings [2].

We have been interested in multimedia medical databases, especially audio databases for heart sounds and lung sounds, and we have proposed a MIDI encoding method especially for heart sounds, and this algorithm features its real-time processing capability [3]. Besides our implementation of this proposed method for a heart-sound coding, we tried applying a MIDI coding to other types of sound materials [4].

In order to process various types of acoustic signals, we categorize general acoustic signals as two groups whether the spectrum components are distributed intermittently or continuously. Most musical signals belong to the former intermittent group whereas human voices, biological signals and the other natural sounds belong to the latter continuous group. Then we defined two kinds of MIDI coding approaches: a single pitch coding and a multiple formant coding [4].

As a result of our implementation including a multiple formant frequency coding, we found out it was possible to playback speeches and singings by MIDI tone generators, using our proposed nonlinear extended GHA (Generalized Harmonic Analysis) frequency analysis method [5]. Then we have focused on the decoder sound module, and tried to produce more natural sounds as the original PCM sounds [6]. We also improved the frequency analysis precision by variable frame-length analysis, evaluated coding precision [7].

These days we have been developing sound source separation techniques, especially separation of vocal parts from mixed down songs. As a very low bit-rate audio coding method, we have evaluated and compared our proposed method with the other encoding methods. We have reported the encoding quality of 8-kbps data by our MIDI encoding method was superior to that of 16-kbps MPEG-1 layer 3 encoded data [8].

Furthermore, using these improved coding techniques we are trying to apply this MIDI coding to symbolic expression of acoustic signals for retrieving music archives by note-based keywords. As a structured symbolic description format, we choose XML (eXtensible Mark-up Language)[9] because this format is widely used for medical application [11].

Currently, we are trying to separate vocal sounds included in popular songs and encode both vocal sounds and background instrumental sounds into separate MIDI channels. And also, we are trying to extract articulation parameters such as MIDI pitch-bend parameters in order to reproduce natural acoustic sounds using a GM-standard MIDI tone generator. In this paper, we are going to overview our improved MIDI encoding algorithm focusing on current research works. We expect this software tool, which has been implemented on Windows platforms and is now distributed for free in Japan, to be used as sonification tools, converting given electronic time-based signals to interactive MIDI controllable audio contents.

2. BACKGROUND

Monitoring operations are required in various kinds of industries including printing processes and medical cares, and most of them are dependent on operators' visual tracking of sensor signals. In medical cares, pulse sounds synchronized with ECG signals or alarm sounds indicating emergent events are currently used for bedside monitors. These simple audio signals are helpful for medical staff and give great advantage to quality of cares although we fear somewhat negative influence for patients and surroundings from a hospital acoustic environment viewpoint.

We propose improving this auditory environment and also extending this idea to general monitoring operations utilizing MIDI technology so that we should monitor sensor signals through their synchronized MIDI sounds by the following two methods described in Fig.1.

The first one called Direct Real-time Playback is converting the source signals including inaudible electric signals to MIDI notes directly, and we can recognize the feature

patterns in the source signals by listening to their converted MIDI streams. This conversion is based on our proposed MIDI encoder, which can convert automatically any kind of audible or inaudible electric signals to music notes and can playback them with a MIDI tone generator. Although the monitoring precision of this method is as high as visual tracking of waveforms, the playback audio contents are based on the condition of the monitoring target machine or patient, and become uninteresting or sometimes uncomfortable for us. Therefore this method is unsuitable to long hour operations.

The second one we call Indirect Stored Playback is modulating a given MIDI music notes by the source signals. During playback of our known music piece, each sensor signal can change its tempo, pitch, volume, and tone parameters respectively. In other words, we can recognize an unstable condition of the source indirectly by a rhythmical change of the played back music sounds. Moreover, we can construct a piece of MIDI data on up to 16 tracks, and each track can be modulated independently by different sets of external sensor signals, therefore theoretically we can monitor maximum 16 x 4 different signals at the same time by a single piece of MIDI playback. Although the monitoring precision of this method is not so good as the previous method, the playback audio contents are based on our desirable popular music piece, and will become comfortable for us during a stable condition of the source. Therefore this method is suitable to long hour operations.

Another feature of our proposed concept is support of networked remote clients. As we use MIDI as a core audio format and its communication speed is typically less than 10 kbps (up to 32 kbps), we can easily insert mobile communication networks between monitoring servers and MIDI playback clients.

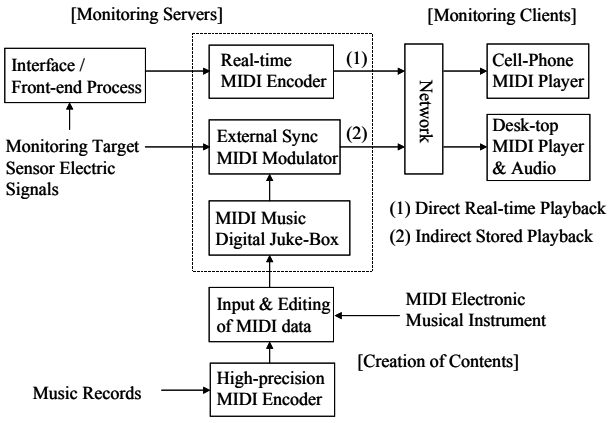


Figure 1. *Concept of MIDI-based auditory monitoring system.*

In this paper, we are focusing on the common techniques of MIDI encoder, especially the high-precision type applied for the second monitoring method. MIDI technique is ordinarily used as the top horizontal right-ward direction flow shown in Figure 2. By digitizing music score written by a composer into the MIDI format, we can produce somewhat music instrument parts without musician, musical instrument nor recording studio. And this technique is widely applied to today's commercial music production. However, for non musical acoustic materials such as vocal parts and sound effect parts which are difficult to be expressed with MIDI music notes, you need singers and a recording studio facility.

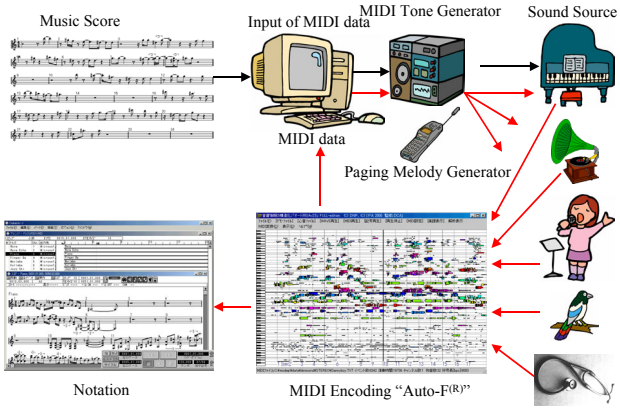


Figure 2. *Concept of MIDI conversion tool.*

In this sense, we proposed the other way, which converts the existing audio waveform materials shown at the right side to MIDI music notes. In this method we can control any kind of audio materials by the MIDI functions interactively and reproduce even vocal sounds by MIDI tone generators or electronic musical instruments. And editing converted MIDI codes, we can also reproduce music scores like shown in the left side.

Moreover, the given materials are not restricted to audible signals. Our proposed algorithm can be applied to supersonic signals or low frequency electronic signals such as ECG (Electro Cardiogram), we can monitor the given signal contents by audible sounds with MIDI tone generators. The previous musical acoustic signal conversion works such as [10] include subjectivity or artistic manipulations whereas our conversion approach is totally objective and based on source signals.

3. ENCODING METHOD

3.1. Concept of MIDI Coding Method

Our proposing MIDI coding is a kind of analytic-synthetic coding approaches which separate a given audio signal to amounts of sinusoidal waveforms and describe it with the frequency and intensity parameters of its separated harmonic sine waveforms. Whereas our method shown in Figure 3 separates it to several predefined harmonic complex waveforms, which available MIDI tone generators can generate, and describes it with the frequency (namely MIDI defined note-number) and the intensity (namely MIDI defined velocity) parameters of its separated predefined harmonic complex waveforms. In general the number of required harmonic complex waveforms for describing will be not so many as those of the analyzed sinusoidal waveforms because each harmonic complex waveform is made of several sinusoidal waveforms. Therefore, the coded bit-rate of this MIDI method will become smaller about 1/10 than that of the previous analytic-synthetic coding approaches.

As illustrated in Figure 3, the amplitude and frequency parameters of the integrated harmonic complex waveforms are converted to the velocity and note-number parameters of MIDI notes. Then these MIDI notes are classified to several MIDI channels depending on their tone parameters by the Sound Source Separation process. If we assign appropriate MIDI tone parameters to each channel, we can reproduce the similar signal

as the original using a MIDI Standard Electronic Instrument, for example a MIDI tone generator. In this sense, this proposed method can be applied to very low bit-rate audio encoding system.

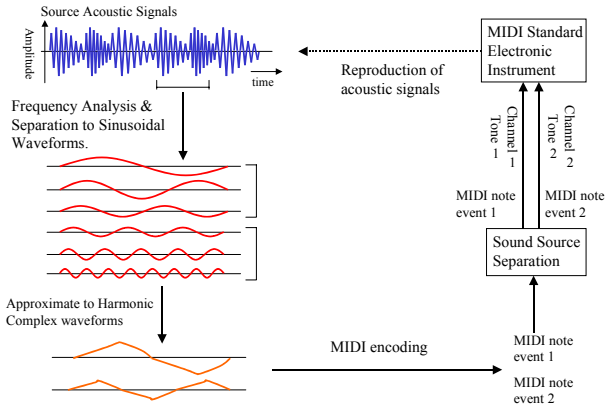


Figure 3. *Basic concept of MIDI coding method.*

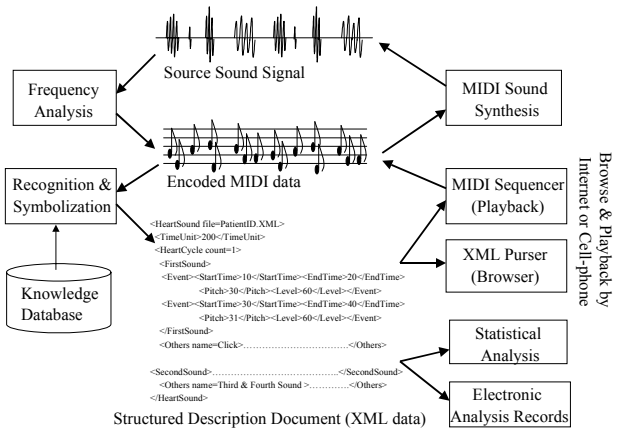


Figure 4. *Concept of XML transcription for acoustic signals.*

3.2. Extended Concept of XML Transcription of Acoustic Signals

Figure 4 shows an extended concept of this MIDI encoding tool, which has provided an additional XML symbolization post-processing. At first the given waveform signal are converted to MIDI data by the Frequency Analysis as we described. Then the meta-data or tag symbols are extracted from this converted MIDI data by the Recognition & Symbolization process with accessing the Knowledge Database, and these meta-data are described in the XML text format document.

This document may include all or an abstract of MIDI event data, which can reproduce acoustic signals by controlling a MIDI Sequencer. By choosing the XML standard as an output file format, we can browse and analyze the detailed numerical contents of the target acoustic signals in our preferred format defined by the XSL style-sheet language, and these data can be easily filed in existing electronic database record systems. As the included MIDI data reflect the numerical conditions of the target signals, these data can be used for further statistical analysis.

The additional XML conversion process is totally depending on types of application, especially the Knowledge Database. In our prototype system, we have designed heart sound XML transcription system with constructing heart sound medical symbol database [11]. From the next section we are going to focusing on the first general purpose MIDI conversion process described in Figure 3. As for the second detailed technical information is available in the published papers ([9][11]).

3.3. Two Types of Approaches for MIDI Coding

MIDI data are a collection of pairs of *Note-On* and *Note-Off* command strings called events where each pair denotes a piece of music note, and each event is composed of a relative time-stamp (*delta-time* in MIDI standard terms), a frequency (*note-number*) and a intensity (*velocity*) parameter [3]. In this section we describe how these MIDI parameters can be numerically obtained. We propose two approaches depending on the types of source acoustic signals whether musical acoustic or the other signals.

Using a frequency analysis technique such as GHA (Generalized Harmonic Analysis) method [5], we can separate some frame $g(t)$ (frame-length=T) extracted from the given acoustic signal. By the variable frame-length analysis technique [7], we can obtain a set of N separated sinusoidal functions as follows:

$$g(t) \cong \Sigma_{n=1,N} \{ A_n \sin(2\pi f_n t) + B_n \cos(2\pi f_n t) \} . \quad (1)$$

Where the coefficients both A_n and B_n are defined by the following equations.

$$A_n = 2/T_n \ \Sigma_{t=0,T_n-1} \{ g(t) \sin(2\pi f_n t) \} . \quad (2)$$

$$B_n = 2/T_n \ \Sigma_{t=0,T_n-1} \{ g(t) \cos(2\pi f_n t) \} . \quad (3)$$

In these equations T_n is the maximum value of $T_n = k / f_n < T$ (k: appropriate positive integer value), and f_n is given by the equation $f_n = 440 \cdot 2^{(n-69)/12}$ which generates frequency values on the MIDI note-number logarithm scale. Defining harmonic complex functions as $u_i(t)$, we can express the equation (1) with smaller summation elements $P \ll N$ as follows:

$$g(t) \cong \Sigma_{i=1,P} \alpha_i u_i(t) . \quad (4)$$

Then we define $p(i)$ as a represented frequency identification number of $u_i(t)$. In some case like shown in Figure 5-(A), $u_i(t)$ can be expressed with a summation of a fundamental frequency $f_{p(i)}$ and its harmonic components $jf_{p(i)}$, as follows (j: integer value 1,2,3…):

$$u_i(t) = \Sigma_{j=1,J} \{ A(i,j) \sin(2\pi j f_{p(i)} t) + B(i,j) \cos(2\pi j f_{p(i)} t) \} . \quad (5)$$

In the other cases like shown in Figure 5-(B), $u_i(t)$ can be expressed with a summation of a formant local peak frequency $f_{p(i)}$ and its neighbor continuous frequency components $\beta_j f_{p(i)}$, giving the integer value of δ around 1, as follows ($f_{p(i)-\delta} \le \beta_j f_{p(i)} \le f_{p(i)+\delta}$) :

$$u_i(t) = \Sigma_{j=1,J} \{ A(i,j) \sin(2\pi \beta_j f_{p(i)} t) + B(i,j) \cos(2\pi \beta_j f_{p(i)} t) \} . \quad (6)$$

If we choose the harmonic complex function $u_i(t)$ from the wave tables defined in our using MIDI tone generator, we can reproduce $g(t)$ with P number of notes giving *note-number* $N(i)$

and velocity value $V(i)$. These values are generated from $f_{p(i)}$ and α_i parameters as follows:

$$N(i)=40 \log_{10} (f_{p(i)}/440) + 69 . \quad (7)$$

$$V(i) =128C\alpha_i^{1/2} \quad (C\text{: constant, } 0\le V(i)\le 127). \quad (8)$$

The time of *Note-On* of this MIDI note event is the start position of extracted frame on the source acoustic signal, and the duration time (or *Note-Off* delta-time) is given by the analyzing frame- shifted interval τ.

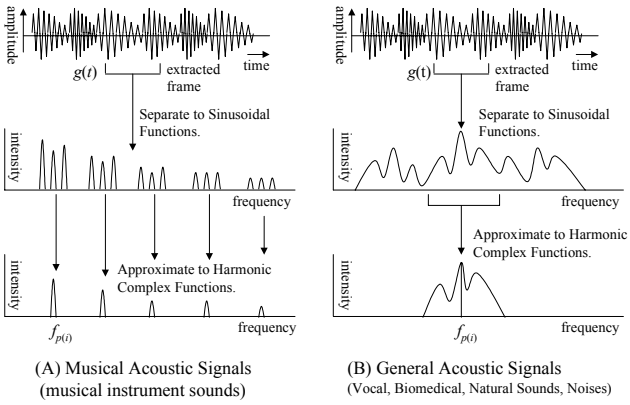


Figure 5. *Two types of approaches for MIDI coding*.

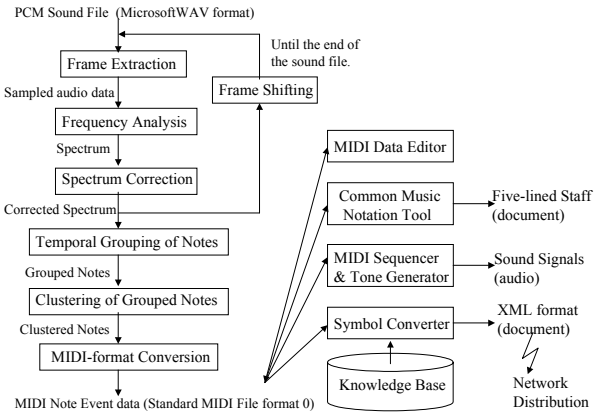


Figure 6. *Abstract flowchart of MIDI encoding process*.

3.4. Algorithm Design of MIDI Coding

Figure 6 shows the whole process of our designed MIDI encoding process. The first Frame Extraction is extracting the specified length of signal samples from the source PCM Sound File. The length of frame is fixed and each frame position is determined as overlapped over the next frame. The variable frame shifting pitch is the feature of our algorithm, which provides precise temporal analysis resolution at low calculation load. The specific determination algorithm described in in Figure 7.

The second is the Frequency Analysis, which separates the part of a source signal to N number of spectrum components using the Short Term Discrete Fourier Transform and needs the most calculation load.

The next Spectrum Correction is sharpening resolution of the calculated spectrum. Each calculated power spectrum component $P_n=A_n^2+B_n^2$ at each note-number (n=0, ⋯ ,N-1), generated from the given acoustic signal frame $g(t)$ is supposed to be a linear summation of the real power spectrum components P^o_n as the following, where An and Bn are calculated by the equation (2) and (3), respectively.

$$P_n= \sum_{m=0,N-1} P^o_n R_{mn} / R_{mm} . \quad (9)$$

We should estimate P^o_n values from the calculated P_n using $R_{mn}=A_{mn}^2+B_{mn}^2$, which is our defined mutual spectrum matrix, a collection of spectra table for respective harmonic sinusoidal functions which are defined as follows:

$$A_{mn}=2/T_n \sum_{t=0,T_n-1} \sin(2\pi f_n t)\sin(2\pi f_n t) . \quad (10)$$

$$B_{mn}=2/T_n \sum_{t=0,T_n-1} \sin(2\pi f_n t)\cos(2\pi f_n t) . \quad (11)$$

Including this correction based on the mutual correlation matrix, our proposing whole spectrum correction processes are illustrated in Figure 8.

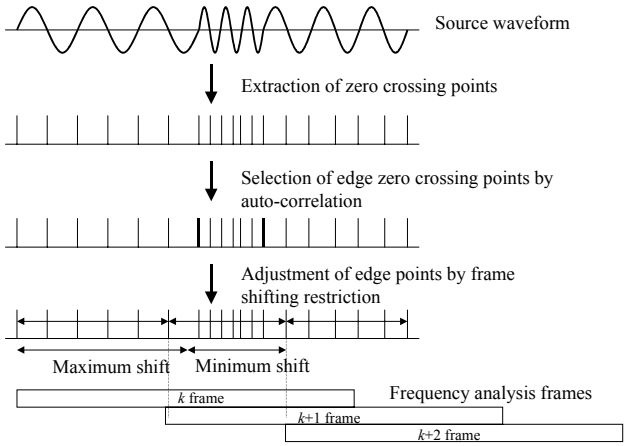


Figure 7. *Optimal determination method of analysis frame position*.

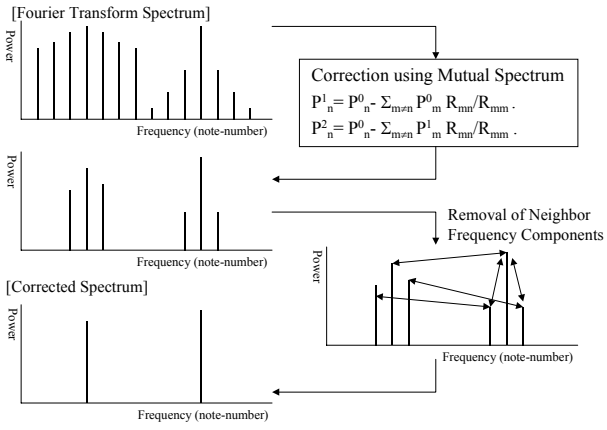


Figure 8. *Process of Spectrum Correction*.

The fourth Temporal Grouping of Notes is connecting the temporally adjacent notes (are the same as spectrum components), which have similar frequency and volume parameters, and producing a longer duration notes. In this step a

temporal variation parameter is added to each grouped note. This parameter is also used for the sound source separation. Moreover, several articulation parameters of pitch-bend and expression control data can be added to each MIDI grouped note illustrated as Figure 9.

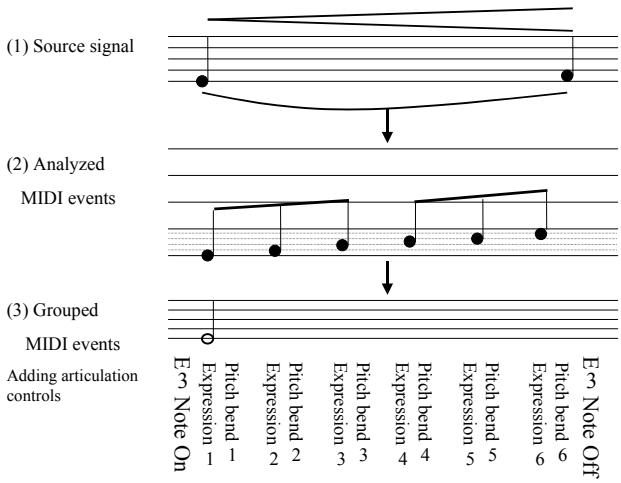


Figure 9. Process of temporal grouping of notes.

These control parameters are calculated on the frequency and volume transition data of the micro-tone notes of each temporally grouped note. The micro-tone frequency $f(n,m)$ between consecutive semi-tones is defined as the following, where n is a note-number ($0 \le n \le 127$) and m is its mico-tone number ($0 \le m \le M(n)-1$, $M(n)$ is the resolution of miro-tone at note-number n).

$$f(n,m)=440 \bullet 2^{\{n-69.5+m/M(n)\}/12} \qquad (12)$$

The fifth Clustering of Grouped Notes is assigning each grouped note to appropriate tone channel based on its additional tone parameter as harmonic distribution, frequency fluctuation, stereo panning and temporal variation parameters. The Figure 10 illustrates this clustering process based on the multiple variable statistical analysis method. For this purpose we should construct the Tone Management Table, which stores distribution data of the four kinds of tone parameters for each tone group such as piano and vocal.

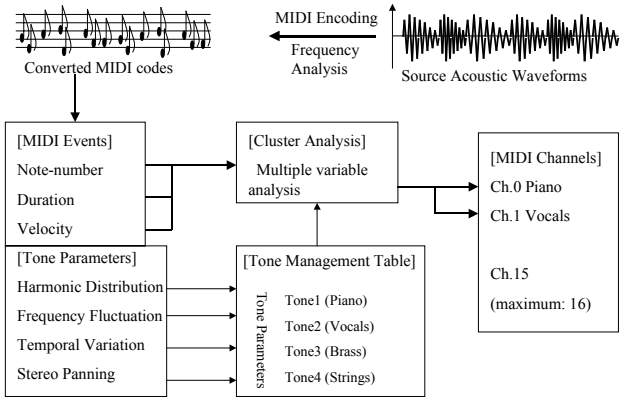


Figure 10. *Clustering process of grouped notess.*

The last process is converting each clustered note to a MIDI event data format such as the SMF (Standard MIDI Fle) format with defined channel parameters added. Before that we should regulate the number of harmony (or temporally maximum simultaneous notes) and output notes (or output bit-rate) in order that the standard GM or other types of MIDI tone generator can play-back encoded data.

The right side flowcharts in Figure 6 show the several utilization processes after MIDI data are created. The top three functions of MIDI Data Editor, Common Music Notation Tool, and MIDI Sequencer can be provided in commercially available off-the-shelf DTM (Desk Top Music) composition tools such as "Yamaha XG-Works" or "Steinberg Cubasis" what we use. Our tool also includes structuring and symbolizing the encoded MIDI data into XML (eXtensible Markup Language) document format for network audio content distribution. This converted XML data include the other analyzed parameters than what the SMF MIDI file has, and will be helpful to further statistical analysis. The whole processes including input or output interfaces are illustrated in Figure 11.

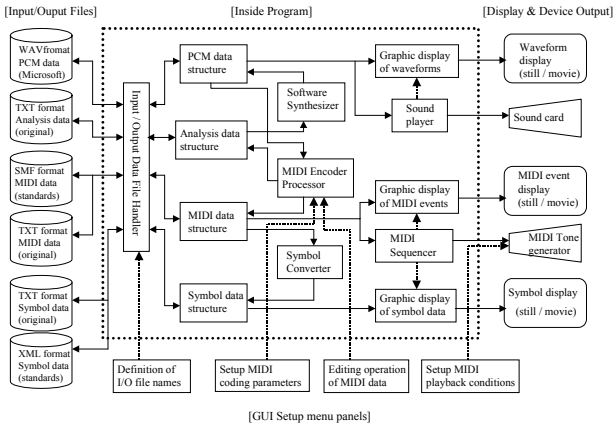
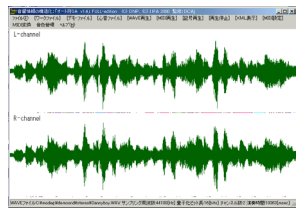


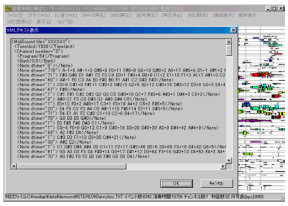
Figure 11. *Block diagram of "Auto-F/SA" tool.*

4. CONCLUSIONS

In this paper, we have described an abstract MIDI encoding algorithm, which can be applied for electric signal monitoring operations. Figure 12 shows an encoding example of musical signal; the source audio material was the Irish folksong: "Danny Boy", and its analyzed length was 20 seconds. The output bit-rate was 10 kbps, its calculation time was about one minutes using a Pentium III 600 MHz Windows98 PC. Based on MIDI converted data shown in Figure 12-(B).

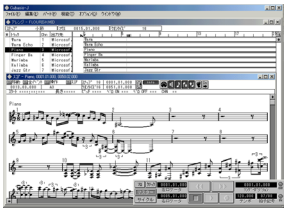


(A) Source acoustic sounds



(C) XML converted results

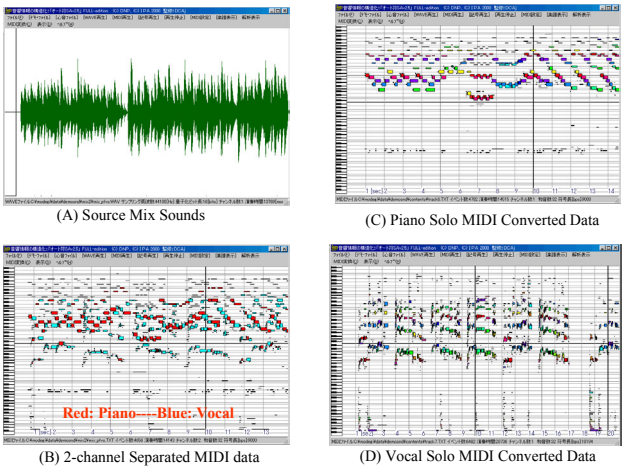
(B) MIDI coded results



(D) Notation by *Cubasis* tool

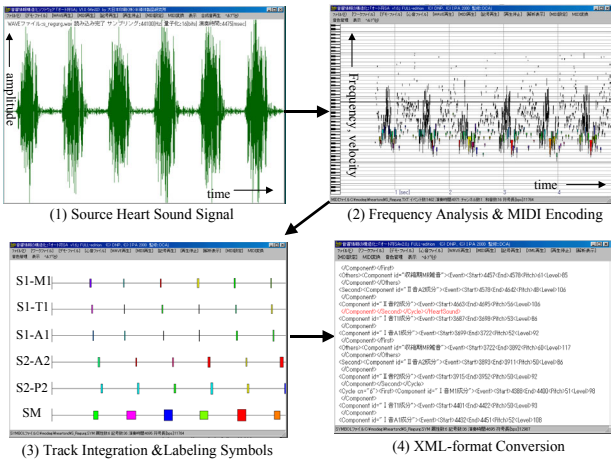
Figure 12. Snapshots of MIDI encoder software tool.

In case of music analysis application, we could separate vocal parts from singing song materials and encode both vocal and instrumental parts into multiple channel MIDI data streams. And we could generate complete musical sounds including vocal sounds with a single GM-standard MIDI tone generator. Figure 13 shows an encoding example of the mixed signal of piano and vocal sounds; the source audio material was the Mozart song: "Twinkle twinkle little star." Fig. 13-(B) shows two-channel separated MIDI data from mixed source signals of a piano solo signal and vocal solo signal whose converted MIDI data are shown in Fig.13 –(C) and (D). From this experiment, its accuracy of sound source separation was estimated around 80%.



(A) Source Mix Sounds
(C) Piano Solo MIDI Converted Data
(B) 2-channel Separated MIDI data
(D) Vocal Solo MIDI Converted Data

Figure 13. A Sound Source Separation Example for Piano and Vocal Musical Signals.



(1) Source Heart Sound Signal
(2) Frequency Analysis & MIDI Encoding
(3) Track Integration &Labeling Symbols
(4) XML-format Conversion

Figure 14. Heart sound analysis application.

Figure 14 shows an example of this tool applied to heart sound analysis. The source length was 15 seconds, and the output bit-rate was 1 kbps, its calculation time was almost real-time, less than 5 seconds. The source heart sounds including systolic heart murmur components were similarly converted to MIDI data, and they are clustered to six tracks dependent on the tone characteristics defined by the medical tone database. These separated components are categorized as symbols, and described by the XML format, shown as Figure 14-(4). As we see, the first and second sounds could be separated to several valve sound components, and the systolic murmur components were clearly extracted. This abnormal murmur component can be analyzed to several sub-components based on valves [11].

As for future works, we are considering development of higher accurate sound source separation techniques, higher performance structuring and symbolizing processing techniques for generating XML data, and an algorithm redesign for real-time processing. Then we will apply our proposed technique to our production monitoring operations and the other fields.

This research has been promoted by the Digital Content Association of Japan as a 2000-year government project: "Development of Multimedia Content Creating Tools," being also financially supported by the Information-technology Promotion Agency Japan and the Ministry of Economy, Trade and Industry Japan. As a closing remark, we thank all of the personnel belonging to the above organizations, who support this research project. The developed software MIDI encoder tool (currently Japanese MS-Windows edition only !) is distributed for free at the following Web site: (Currently version 2.4 available, URL: http://www.dcaj.or.jp).

5. REFERENCES

[1] M. Goto and Y. Muraoka,"A Beat Tracking System for Acoustic Signals of Music," in *Proc. Second ACM Int. Conf. on Multimedia*, Dec.1994, pp.365-372.

[2] R.J. McNab, L.A. Smith, I.H. Witten, C.L. Henderson and S.J. Cunningham, "Towards the Digital Music Library: Tune Retrieval from Acoustic Input," in *Proc. 1st ACM Int. Conf. on Digital libraries*, 1996, pp.11–18.

[3] T. Modegi and S. Iisaku, "Application of MIDI Technique for Medical Audio Signal Coding," in *Proc. IEEE 19-th Int. Conf. the Engineering in Medicine & Biology Society*, Chicago, USA, Oct. 1997, pp.1417-1420.

[4] T. Modegi and S. Iisaku, "Proposals of MIDI Coding and its Application for Audio Authoring," in *Proc. Int. Conf. on IEEE Multimedia Computing and Systems*, Austin, USA, Jun. 1998, pp.305-314.

[5] T. Modegi, "Multi-track MIDI Encoding Algorithm Based on GHA for Synthesizing Vocal Sounds," *J. Acoustic Society of Japan (E)*, Vol.20, No.4, pp.319-324, 1999.

[6] T. Modegi, "High-precision MIDI Encoding Method Including Decoder Control for Synthesizing Vocal Sounds," in *Proc. Seventh ACM int. conf. on Multimedia*, Part 2, Orland, USA, Nov. 1999, pp.45- 48.

[7] T. Modegi, "MIDI Encoding Method Based on Variable Frame-length Analysis and its Evaluation of Coding Precision," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, New York, USA, Aug. 2000, pp.1043-1046.

[8] T. Modegi, "Very Low Bit-rate Audio Coding Technique Using MIDI Representation," in *Proc. ACM 11-th NOSSDAV Workshop*, New York, USA, Jun. 2001, pp.167-176.

[9] T. Modegi, "Structured Description Method for General Acoustic Signals Using XML Format," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, Tokyo, Japan, Aug.2001, pp.932-935.

[10] M. Quinn, "Research Set to Music: The Climate Symphony and Other Sonifications of Ice Core, Radar, DNA, Seismic and Solar Wind Data," Proc. of the Int. Conf. on Auditory Display, Finland, Jul. 2001, pp.56-61.

[11] T. Modegi, "XML Transcription Method for Biomedical Acoustic Signals," in *Proc. 10th World Congress on Health and Medical Informatics Medinfo2001*, London, UK, Sep.2001, pp.366-370.