

SMF의 양자화 알고리즘의 구현

임 혁, 송문빈, 권오균, 정연모

경희대학교 전자공학과

전화 : 031-201-2585 / 핸드폰 : 011-719-0712

Implementation of a Quantization Algorithm for SMF(Standard MIDI File)

Hyuk Lim, Moonvin Song, Ohkyun Kwon, Yunmo Chung
Department of Electronic Engineering, Kyung Hee University
chung@khu.ac.kr

Abstract

멀티미디어에 대한 관심 증가로 미디어 파일중의 하나인 SMF의 사용자층이 넓어지고 있다. 기존의 시퀀싱 소프트웨어의 양자화 기능은 음표 하나만을 기준을 했기 때문에 건반 혹은 다른 미디 컨트롤러를 통해 실시간으로 입력받은 SMF를 양자화한 후 또 다시 수작업으로 보정을 해 주어야 했다. 본 논문에서는 이와 같은 번거로움을 줄이고, 연주자가 최초로 의도한 악보에 근접하기 위한 SMF의 양자화 알고리즘에 관해 제안하였다.

I. 서론

컴퓨터음악에 대한 관심 증가로 간단한 SMF정도는 직접 만들 수 있는 사람들이 점점 늘어나고 있다. MIDI파일 제작 소프트웨어(이하 시퀀싱 소프트웨어)에서 직접 악보 형태로 입력하는 방법은 비숙련자들도 쉽게 작업할 수 있기 때문에 널리 이용되고 있다. 그러나 음표를 하나 하나 입력해야 하기 때문에 많은 트랙이 들어 있는 복잡한 곡일수록 작업 시간이 늘어난다는 문제점을 갖고 있다. 따라서 전문 음악인과 같이 짧은 시간에 좀 더 많은 작업량을 필요로 하는 사람들은 MIDI 출력기능을 가진 MIDI 컨트롤러(Master Keyboard, MIDI Guitar, Breath Controller등)를 이용

해 실시간 연주를 해 이를 입력받아 파일로 만드는 방법을 선호한다. 음표입력 방법에 비해 훨씬 적은 시간으로 파일을 만들 수 있지만 건반 연주자의 숙련도에 따라 입력받은 신호가 완벽하지 않기 때문에 별도의 양자화를 해야 한다는 단점이 있다.

기존의 시퀀싱 소프트웨어의 양자화 알고리즘은 기준이 되는 note하나를 정한 후 그 note에 맞추어 모든 note의 길이를 늘이고 줄이는 방법이였기 때문에 양자화 후에도 수작업으로 별도의 보정을 해 주어야 했다. 연주자의 숙련도가 떨어질 경우, 이와 같은 보정작업은 더 많아진다.

본 논문에서는 먼저 MIDI파일의 생성 방법에 대해 알아보고 이어서 기존의 시퀀싱 소프트웨어의 양자화의 문제점 및 본 논문에서 제안한 양자화 알고리즘과의 비교를 한 후 결론을 내리기로 한다.

II. MIDI파일의 생성 방법

MIDI파일을 생성하는 방법에는 두 가지가 있다. 하나는 사용자가 각 컨트롤 신호를 키보드 혹은 마우스로 입력하는 방법이고, 다른 하나는 MIDI 컨트롤러를 이용해 악기를 연주하듯이 실시간으로 입력하는 방법이다.

2.1. 마우스, 키보드를 이용한 수치입력

MIDI파일 내에는 표 1과 같은 컨트롤 신호들이 사용된다. 별도의 MIDI 컨트롤러가 없거나, 컨트롤러 조작에 익숙한 사용자들은 이런 컨트롤 신호들을 마우스 혹은 키보드로 입력해야 한다. 대부분의 시퀀싱 소프트웨어는 MIDI 메시지를 직접 타이핑하는 방식 이외에 음의 길이, 음의 높이를 각각에 대응하는 음표 및 오선지로 표기해 직접 악보에 기입하는 방식을 지원한다. 매우 직관적인 이 방식은 시퀀싱 소프트웨어를 처음 접하는 초보자들에게도 손쉬운 MIDI파일 생성을 가능하게 한다. 그 외의 컨트롤 신호들은 수치입력 또는 가상 Fader, Knob등 별도의 제어기능을 이용해 입력하도록 되어 있다.

표 2. Control Number의 종류

Control No.	내용	값의 범위
0	Bank Select	0~127
1	Modulation Wheel	-8192~8192
2	Breath Controller	0~127
4	Foot Switch	0=OFF, 127=ON
5	Portamento Time	0~127
6(38)	Data Entry MSB (LSB)	*
7	Volume Level	0~127
10	Pan	0~127
11	Expression	0~127
32	Bank Select	0~127
64	Sustain Pedal	0=OFF, 127=ON
65	Portamento ON/OFF	0=OFF, 127=ON
66	Sostenuto Pedal	0=OFF, 127=ON
67	Soft Pedal	0=OFF, 127=ON
91	Reverb	0~127
93	Chorus	0~127
99,98	NRPN MSB, LSB	*
101,100	RPN MSB, LSB	*
121	All Control Reset	0
122	Local ON/OFF	0=OFF, 127=ON
123	All Note OFF	0

2.2. MIDI 컨트롤러를 이용한 실시간 입력

MIDI 컨트롤러에는 마스터키보드, MIDI기타, Breath Controller 등이 있다.

2.2.1 마스터키보드

피아노 건반의 형태를 갖는다. 피아노를 연주할 수 있으면 쉽게 사용할 수 있기 때문에 가장 널리 사용된다. 건반만으로 음 높이, 길이 이외에 Velocity, Aftertouch의 파라미터 전송이 가능하며 별도의 컨트롤러(Pedal, Breath Controller, 키보드에서 지원하는 Fader 등)를 이용해 이외의 파라미터도 전송할 수 있다.

2.2.2 MIDI기타

일렉트릭 기타의 형태이며 주로 기타악보의 입력을 위해 사용된다. 기타를 주로 사용하던 연주자들에게는 마스터키보드보다 MIDI기타가 더 편리하다.

2.2.3 Breath Controller

관악기 파트의 입력을 위해 사용된다. 단독으로 쓰이지 않으며 마스터키보드의 역할을 보조한다. 입김의 세기로 Velocity를 표현하며 Vibrato의 입력 역시 가능하다.

2.2.4 실시간 입력의 문제점

MIDI 컨트롤러를 이용해 실시간으로 MIDI 신호를 입력할 때 컨트롤러의 스위치(건반 또는 기타 스위치)가 ON 상태일 때의 시간, OFF로 돌아올 때의 시간을 각각 기록해 둘의 차이를 바탕으로 note의 길이를 계산한다. 따라서 정확한 타이밍에 정확한 신호를 입력하는 것은 불가능하고 이는 생성되는 note의 길이의 불완전성으로 나타나게된다. 예를 들어 그림 1과 같은 악보는 이상적인 환경의 경우 그림 2와 같은 입력 상태를 갖는다. (가로축:시간, 혹은 박자, 세로축:음높이) 각각의 음에 대한 박자가 완벽히 일치함을 볼 수 있다.



그림 1. 예제 악보

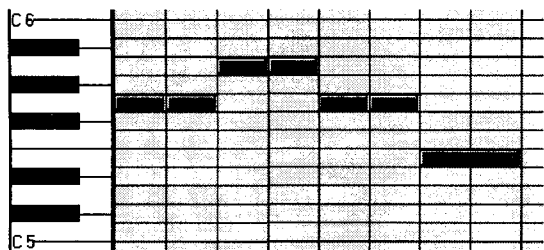


그림 2. 예제 악보의 Piano Roll

실제로 건반을 통해 입력한 결과는 그림 3과 같았다. 약간의 오차가 발생했음을 알 수 있다. 이를 악보로 변환해 보면 그림 4와 같은 형태가 된다. 의도했던 음의 길이 이외에 오차분 만큼의 짧은 음표가 가감되었다.

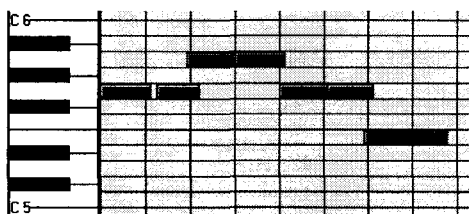


그림 3. 실제 입력받은 신호의 Piano Roll



그림 4. <그림3>을 악보로 변환

따라서 불완전한 연주자의 입력을 바탕으로 한 SMF로부터 처음 의도했던 악보를 복원해 내기 위해서는 적절한 양자화과정이 필요하다.

III. MIDI파일의 양자화

3.1 기존의 시퀀싱 소프트웨어에서의 양자화

그림 5는 기존의 시퀀싱 소프트웨어에서 제공하는 양자화 기능이다. 기존 note 하나에 대해서만 양자화를 수행하기 때문에 양자화 후 다시 한번 수정을 해주어야 한다. 연주자의 숙련도에 따라서 악보의 형태로 직접 입력하는 방법보다 이같은 수정작업에 더 많은 시

간이 소요되기도 한다.

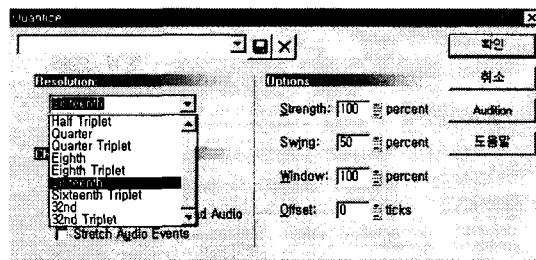


그림 5. 시퀀싱소프트웨어의 양자화

3.2 새로운 양자화 알고리즘

본 논문에서 제시한 양자화 알고리즘의 블록도는 그림 6과 같다. SMF를 읽어들이어 특정 채널의 Note ON, Note OFF에 해당하는 메시지를 바탕으로 해당 음의 지속시간을 계산한다. 표 2에 맞추어 각각의 범위에 해당하는 음의 길이를 양자화 하게 된다. 본 논문에서는 Timebase가 120인 경우를 기준으로 양자화 범위를 결정하였다.

표 3. 양자화 범위

음 길이	양자화 범위	해당음표
480	400 ~ 500	은음표
360	300 ~ 380	점2분음표
240	200 ~ 300	2분음표
180	150 ~ 200	점4분음표
120	100 ~ 140	4분음표
90	80 ~ 100	점8분음표
60	50 ~ 80	8분음표
45	40 ~ 50	점16분음표
30	28 ~ 40	16분음표
15	5 ~ 20	32분음표

실제 악보에서 거의 사용되지 않는 64분음표는 제외하고 32분음표를 최소단위로 가정했다. 대개의 연주자가 정해진 음표의 길이에 약간 못 미치는 시간동안 건반을 눌렀다 떼는 것을 감안하여 기준값보다 훨씬 낮은 값이 입력된 경우에도 무리 없이 양자화 되도록 값의 범위를 조절하였다.

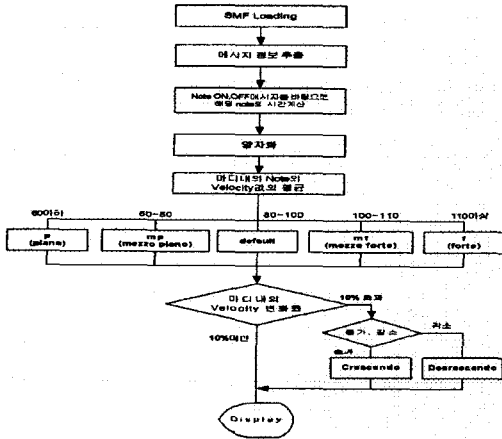


그림 6. 알고리즘 블록도

그림 7은 테스트에 사용할 악보이다. 마스터키보드를 이용해 리얼타임으로 입력한 후 기존의 소프트웨어에서 제공하는 양자화 기능 및 본 논문에서 제안하는 양자화 알고리즘을 각각 적용 후 평가하였다.



그림 7. 검증에 사용된 악보

제시한 알고리즘의 평가 방법은 다음과 같다. Timebase 120일 때 원 악보의 음표에 해당하는 각각의 note의 길이를 비교값으로 한다. 리얼타임으로 생성한 MIDI파일의 각 note의 길이, 기존 시퀀싱 소프트웨어의 양자화 (16분음표, 32분음표) 및 본 논문에서 제안한 알고리즘을 적용한 MIDI파일의 note의 길이를 비교하여 각각의 평균오차를 구하였다. 적용 결과는 표3과 같다.

표 4. 각 알고리즘의 비교

	리얼타임 입력	기존 알고리즘(16)	기존 알고리즘(32)	제안한 알고리즘
오차	12.18	9.09	12.27	5.91

실시간으로 입력받은 MIDI파일의 note들은 평균오차가 12 정도였다. 16분음표를 기준으로 양자화한 경우는 그 오차가 9.09로 줄었지만 32분음표로 양자화한 경우 오히려 오차가 늘어남을 볼 수 있다. 적절하지 않은 양자화범위를 선택했기 때문이다. 본 논문에서 제시한 양자화 알고리즘을 적용한 경우 오차가 5.91로 다른 방법에 비해 우수함을 알 수 있다.

IV. 결론

MIDI 파일의 사용량이 증가하고, 소프트웨어 업계에서는 보다 다양한 기능의 시퀀싱 소프트웨어를 개발하고 있지만 아직까지 실시간으로 입력된 SMF의 양자화에 대한 고려는 아직 미비한 실정이다. 본 논문에서 제시한 양자화 알고리즘을 사용했을 때 기존의 시퀀싱 소프트웨어에서 제공하는 양자화 알고리즘에 비해 훨씬 우수한 결과를 보였다. 향후 다음과 같은 사항들을 고려해야 한다

- 1) 현재의 양자화 알고리즘에는 쉽표 및 셋잇단음표에 대한 고려를 하지 않았다. 악보 중간중간에 쉽표가 삽입되었을 경우 혹은 셋잇단음표가 나올 경우에도 양자화가 가능하도록 알고리즘을 보완해야 한다.
- 2) 실시간 입력 시 서스테인 페달을 사용할 경우, 악보상의 음표의 길이보다 훨씬 적은 시간동안 건반을 누르게 되는데 이에 대한 고려가 필요하다.
- 3) 악보로의 변환을 위한 별도의 악상기호에 대한 고려가 필요하다.
- 4) 각 알고리즘의 성능을 평가하기 위한 방법이 마련되어야 한다.

참고문헌

- [1] 이은영 "MIDI 1.0 : 조작과 기능에 대한 소고", pp.12-15, 성신여자대학교, 1992
- [2] 임혁 "음악관련 전자상거래 시스템 개발", pp.20-23, 경희대학교, 2002
- [3] 박운영 "Cakewalk 3.0 Professional", pp.41-59, 도서출판 헤지원, 1995
- [4] 허정연 "David Lanz - Christofori's Dream" pp.49, 아름출판사, 1993