

음악 표절 분석을 위한 디지털 음악 요소에 대한 연구

신미해*, 조진완**, 이혜승***, 김영철****

A Study of Digital Music Element for Music Plagiarism Analysis

Mi-Hae Shin*, Jin-Wan Jo**, Hye-Seung Lee***, Young-Chul Kim****

요 약

본 연구는 서로 다른 두 음원에 대한 표절 여부를 분석하기 위한 음악 요소에 대해서 연구한다. 따라서 본 연구에서는 먼저 음원 분석을 위해서 디지털 음악 요소에 대해서 알아보고, 이 음악 요소를 컴파일러 기법을 활용하여 어떻게 표절 분석에 이용할 것인지 살펴본다. 또한 복잡한 MIDI 음악 데이터를 간단하게 처리 할 수 있도록 지원하는 오픈소스 자바 API인 JFugue를 활용한다. 따라서 본 논문에서는 JFugue에서 지원하는 포맷인 뮤직스트링(MusicString)을 이용하여 음악 표절 분석 시스템을 설계하고 음악 표절 분석을 효율적으로 처리하기 위하여 뮤직스트링 문법 처리 요소를 살펴보고 추상구문트리(AST)를 구축하는 하고자 한다. 따라서 본 논문은 지금까지 감성적이고 주관적으로 평가해왔던 두 음원의 표절 분석을 시스템적으로 구축하기 위한 첫걸음이 될 것이며, 이 연구가 잘 활용된다면 차후에 두 음원의 표절 여부를 시스템적으로 정형화할 수 있다는 점에 큰 의의를 둘 수 있다고 판단된다.

▶ Keywords : 저작권, JFugue, 음악표절, 추상구문트리, 음원

Abstract

The purpose of this paper is researching musical elements to analyze plagiarism between two sources. We first search digital music elements to analyze music source and examine how to use these in plagiarism analysis using compiler techniques. In addition we are used open source Java

•제1저자:신미해, 교신저자:김영철

•투고일 : 2013. 7. 5, 심사일 : 2013. 7. 25, 게재확정일 : 2013. 8. 6.

* 공주대학교 컴퓨터교육과(Dept. of Computer Education, Kongju University)

** 공주대학교 전자상거래과(Dept. of e-Commerce, Kongju University)

*** 유한대학교 산학협력단(Researcher of Industry Academic Cooperation Foundation, Yuhan University)

**** 유한대학교 e-비즈니스과(Dept. of e-Business, Yuhan University)

※ 본 논문은 2012년도 미래창조과학부(MSIP)의 재원으로 한국연구재단의 지원을 받아 수행된 논문임
(No. NRF-2012R1A2A2A03045162) 제1저자:신미해, 교신저자:김영철

API JFugue to process complex MIDI music data simply. Therefore we designed music plagiarism analysis system by using MusicString which is supported in JFugue and construct AST after investigate MusicString's syntax processing elements to manipulate music plagiarism analysis efficiently. So far music plagiarism analysis is evaluated emotionally and subjectively. But this paper suggests first step to build plagiarism analysis systemically. If this research is well utilized, this is very meaningful to standardize systemically which music is plagiarized or not.

▶ Keywords : Copyright, JFugue, Music Plagiarism, AST, Sound Source

I. 서론

최근 '저작권, 콘텐츠의 생명이다'라는 캐치프라이즈를 가지고 2011년 처음 개최된 '국제 저작권기술 컨퍼런스 2011'[1]은 문화체육관광부와 세계지적재산권기구(WIPO)가 공동 개최하고 ICOTEC 조직위원회와 한국저작권단체연합회가 공동 주관한 행사이다. 이 행사에서는 빠르게 변화하고 있는 콘텐츠 시장에 스마트하게 대응할 저작권 정책의 향후 방향에 대해 논의가 있었다. 논의에서 알려진 바와 같이 이제는 저작권은 국내뿐만 아니라 국외에서도 창작자의 권익을 보호해야 한다는 주장이 제기되고 있다.

현재 표절이란 단어가 익숙한 용어임에는 틀림없지만 우리나라 학술계에서 문제가 되기 시작한 것은 비교적 최근이라고 할 수 있다[1,2,3]. 이는 그동안 음악과 미술 등 예능계에 집중해있던 표절 문제가 학술계에도 만연해 있다는 것을 인식한 계기가 되었다. 표절은 가장 손쉽게 창작물을 만들어 낼 수 있는 방법이기 때문에 그 유혹은 항상 존재해왔으며, 또한 연구의 정직성과 창작성을 떨어뜨려 연구의 의욕을 떨어뜨리는 문제가 있다. 물론 표절에 국한된 것은 아니지만 불법복제에 따른 시장 침해가 중요시되고 있는 이유가 있으며, 표절에 따른 피해도 더 심각하게 받아들이고 있다[2,3].

이처럼 저작권은 전 세계가 공통으로 인정하는 권리이며 국제저작권을 획득하면 세계 공통적으로 저작권을 인정받을 수 있다. 저작권에 대한 기준은 각국마다 약간씩 다른데, 한국에서는 1986년 12월 법률 제3916호로 저작권법을 개정하여 저작자가 살아 있는 동안과 사망 후 50년간 저작권을 보호하며(저작권법 36), 보호기간이 끝난 뒤에는 누구나 자유로이 이용할 수 있다. 각 나라에서는 보통 음악저작권협회와 같은 음악저작권 처리 기구를 만들어 음악저작권과 관련된 업무를 일괄처리하고 있다.(출처 : 네이버 백과사전)

위키피디아[4]에서는 음악 표절(music plagiarism)은 (멜로디나 선율/리듬을 구성하는 최소단위인 모티프를 뜻하는)음악적 영감(musical idea)과 (사운드 레코딩의 일부를 떼어 다른 노래에 사용하는)샘플링에서 발생한다고 한다. 본 연구는 음악 파일 표절 분석을 위하여 필요한 음악요소에 대하여 다룬다.

음악저작권 [音樂著作權, Musical Copyright]이란 저작권법에 의하여 작사가/작곡가/음반 제작자가 갖는 음악저작물의 권리를 말하며, 음악저작물을 이용하는 형태에 따라 실연권/공연권/방송권/상영권/복제권/배포권/발행권/공표권 등을 포함한다(출처 : 네이버 백과사전). 얼마 전 모 일간지(2011년2월15일)에는 다음과 같은 음악과 관련된 내용의 표절 관련 기사가 있었다.

“(중략) 인터넷 상에 표절 논란이 일자 전문가들에게 분석을 의뢰해 후렴구 8마디 중 6마디가 동일하고 곡의 화성도 80~90%가 유사하다는 결론을 내렸다.” (이하생략)

여기서 우리가 주시해야 할 사항이 있다. 즉, 음악과 같은 멜로디는 판단자의 감성만으로 평가하는 것이기 때문에 주관적일 수밖에 없으며, 또한 정형화된 수치로 표현할 수가 없다. 따라서 이러한 두 곡이 정확히 표절인지는 어느 누구도 알 수가 없다. 위의 기사 이외에도 각종 검색엔진에서 “표절논란”이란 단어를 검색해보면 아주 많은 분야에서 표절 논쟁을 볼 수가 있다. 이처럼 음악뿐만 아니라 일상생활에서 우연의 일치인지 혹은 인터넷의 발달로 인한 무의식 중 사용한 지식인지는 정확히 알 수가 없지만 정확한 것은 창작자의 권리를 보호하는 측면에서 시스템적인 판단 근거가 반드시 요구된다[5,6,7]. 따라서 본 연구는 이러한 판단자의 감성이 개입하지 않고 시스템적으로 접근하기 위한 표절 분석 요소에 대해서 알아보려고 한다.

[민간주도 저작권보호] 우리나라는 저작권법 상에 저작권 산권자, 출판권자, 저작인접권자 또는 데이터베이스제작자의 권리를 가진 자를 위하여 그 권리를 신장하는 저작권위탁관리에 관한 규정을 두고 저작권신탁관리업 제도를 운영하고 있다 [8]. 1988년 최초로 신탁허가를 받은 한국음악저작권협회, 한국음악실연자연협회 및 한국방송작가협회를 시작으로 각 권리별 관리단체들이 신탁허가를 받아 저작권자의 권리를 관리하고 있다. 음악분야에는 한국음악저작권협회, 한국음악실연자연협회, 한국음원제작자협회 등이 있고, 영상분야에는 한국영상산업협회, 한국방송실연자협회, 한국영화제작가협회가, 어문분야는 한국복사전송권협회, 한국문예학술저작권협회, 한국방송작가협회, 한국시나리오작가협회 등이 있으며, 공공콘텐츠분야는 한국데이터베이스진흥원과 뉴스저작물을 관리하는 한국언론진흥재단 등 총 12개 단체가 저작권자들의 권리를 위탁관리 중에 있다. 이들 단체들은 해당 분야 저작권자의 권리와 저작권 보호를 위해 불법복제를 단속 지원, 저작권 포럼 개최, 저작권 보호정책 제안, 개별 단체 또는 단체간 합동으로 저작권 보호를 위한 홍보활동 등 다양한 사업을 추진하고 있다.

[해의 저작권 보호 이슈] 2011년 5월 미국무역대표부는 77개국의 지식재산권 보호수준에 대한 '2011 스페셜 301조 보고서(2011 Special 301)' 발표했다. 스페셜 301조 보고서는 미국 종합무역법 제182조에 의거하여 미국무역대표부가 지식재산권 관련업계의 의견 등을 기초로 주요 교역국의 지식재산권 보호수준을 평가하는 연례보고서이다. 이 보고서는 지식재산권을 부당하게 침해하는 국가를 우선감시대상국(Priority watch list; PWL), 감시대상국(Watch list; WL), 306조 감시대상국(Section 306) 분류하고 있으며, 이러한 불공정 무역국으로 지정되면 당장의 보복조치를 받지는 않으나 지재권 보호여부에 대한 미국의 감시대상이 된다.

MP3 파일이 레코딩 음악 공유 파일로 대표되듯이 MusicXML은 악보 공유 파일의 표준이 되었다[9]. MusicXML을 사용하면 한 프로그램에서 작곡한 결과를 다른 프로그램을 사용자에게 공유할 수 있다. 현재 160개 이상의 애플리케이션에서 MusicXML을 지원한다. 하지만 MusicXML은 프로그램간 배포를 용이하게 만들므로 인해서 너무나 많은 사항이 기술되어 있다. 따라서 데이터 처리(프로세싱)용으로는 알맞지가 않다. 이에 본 연구에서는 음악 연주/분석 관점으로 처리하여 음악 파일을 수치화하여 복제 분석에 용이한 오픈소스 자바 API인 JFugue[10]에서 지원하는 포맷인 MusicString을 파싱하고 추상구문트리(Abstract Syntax Tree: AST)를 생성하여 표절 분석에 사용한다.

II. 관련 분야

현재 음악과 관련되어 있는 표절 분석 기술은 국내의 적으로 거의 존재하지 않는다. 이는 시스템적으로 정형화할 수 있는 기술이 한계가 있는 점도 있겠지만, 만일 시스템적으로 정형화 하였더라도 그 객관성에 의문이 가질 수도 있다. 따라서 음악 표절 관련 부분은 검수자의 감성적인 판단에 의존할 수밖에 없는 실정이다. 다음 그림 1은 2011.4.27. CNBLUE의 '외톨이야'가 표절이 아니라는 저작권위원회 판결 내용을 보여주고 있다[11].

Copyright Issue Report 제7호 2011. 4. 27.

CNBLUE의 '외톨이야'가 표절이 아니라는 제1심 판결

임광섭*

■ 기초 사실

- 원고들은 그룹 'ynot'의 멤버로 활동하는 가수들로서 2008. 4. 26. 노래 '파랑새'를 작곡하여 앨범 'Greenapple'에 발표하였고, 피고들은 그룹 'CNBLUE'의 노래 앨범 'CNBLUE Bluetory'에 수록되어 발표된 '외톨이야'를 작곡한 자들임.
- 원고들이 파랑새 중 피고들에 의해 부당하게 이용(표절)되었다고 주장하는 부분(이하 '이 사건 노래 부분'이라 함)과 그에 대응하는 피고들 작곡의 외톨이야 악보 부분(이하 '이 사건 대비 부분'이라 함)은 모두 후링크로서, 위 두 노래 모두 후링크의 첫째 및 둘째 마디의 표현이 셋째 및 넷째 마디에 계속하여 동형진행(同形進行, Sequence)되는 형식을 가지고 있음.

■ 원고들의 주장

- 원고들은 이 사건 청구원인으로, 피고들이 작곡한 외톨이야 중 이 사건 대비 부분은 원고들이 작곡한 파랑새의 해당 부분을 표절하거나 일부 변형하여 사용하였고, 위 부분은 전체 곡에서 질적 및 양적으로 핵심적인 부분을 차지하는 후링크를 포함하여 클라이맥스에 해당하며 전체 곡에서 차지하는 비중도 파랑새는 전체 곡의 1/2에 해당하고, 외톨이야는 전체 곡의 2/5에 해당함.
- 피고들이 작곡한 외톨이야는 원고들이 작곡한 파랑새를 표절한 것에 해당하여 원고들의 음악저작권 중 저작인격권을 침해하였으므로, 그로 인하여 원고들이 입은 손해로서 위자료 5,000만 원을 지급할 의무가 있다고 주장함.

* 한국저작권위원회 조사분석팀 선임

한국저작권위원회 조사분석팀

1

그림 1. 음악표절 판결문 예제
Fig.1. Sample of Music Plagiarism Judgement

그림 1에서와 같이 원고가 제기한 내용은 다음과 같다.

"원고들은 이 사건 청구원인으로, 피고들이 작곡한 외톨이야 중 이 사건대비 부분은 원고들이 작곡한 파랑새의 해당 부

분을 표절하거나 일부 변형하여 사용하였고, 위 부분은 전체 곡에서 질적 및 양적으로 핵심적인 부분을 차지하는 후렴구를 포함하여 클라이맥스에 해당하며 전체 곡에서 차지하는 비중도 파랑새는 전체 곡의 1/2에 해당하고, 외톨이야는 전체 곡의 2/5에 해당함”

위의 문구를 보면, 검수자(여기서는 원고)가 주장한 사안은 모두 사건 즉 원고의 주관이 개입된 판단이다. 이 주장은 판단하는 검수자에 따라 바뀔 수 있는 수치라는 점이 문제가 될 수 있다. 따라서 이를 표절에 관련된 내용을 시스템화시키는 일은 더욱 시급하다고 할 수 있다. 또한 법원에서 제기한 적용 법리를 살펴보면 다음과 같다.

... 생략

음악저작물은 일반적으로 가락(melody), 리듬(rhythm), 화성(chord)의 세가지 요소로 구성되고, 이 세 가지 요소들이 일정한 질서에 따라 선택·배열됨으로써 음악적 구조를 이루게 되는데, 음악저작물을 서로 대비하여 그 유사성 여부를 판단함에 있어서는 해당 음악저작물을 향유하는 수요자를 판단의 기준으로 삼아 음악저작물의 표현에 있어서 가장 ... 중략

위의 문구에서 보면, 음악저작물의 판단을 보면 가락(melody), 리듬(rhythm), 화성(chord)의 3가지 요소로 구성되고, 이 세 가지 요소들이 일정한 질서에 따라 선택·배열됨으로써 음악적 구조를 이루게 되는데, 음악저작물을 서로 대비하여 그 유사성 여부를 판단하는 것을 알 수 있다. 이러한 사실들을 본 연구에서 잘 활용한다면 법원이 판단하는 근거를 명확히 분석하여 표절 분석 시스템화 한다면 국내외적으로 많은 표절 기반 기술을 정립하고 획득할 수 있을 것이다. 따라서 본 연구가 진행되기 위해서는 음악 분야와 IT 기술이 반드시 융합되어 수행되어야 할 것이다.

현재 인터넷 매체나 메스컴의 발전은 정보화 사회의 표절이라는 부작용을 가지고 있다. 이러한 부작용을 해결하고 방지하기 위해서는 표절에 관한 통합 환경이 필요하며, 정형적으로 표절 여부를 판별하는 방식이 그 어느 때 보다도 절실한 실정이다. 본 연구는 이와 같은 현실성에 적합한 연구의 필요성과 중요성을 가지고 있기 때문에 반드시 정립되어야 한다고 생각한다[13,14,15].

III. 음원 분석

본 절에서 활용한 음원 분석은 JFugue[10]에서 활용하기 위하여 분석되었기 때문에 JFugue 음원 요소에 기반을 두고 있다.

1. 기본 음원 분석

음표는 C, D, E, F, G, A, B 쉼표는 R로 표시한다. 음표 다음에는 샤프, 플랫, 옥타브, 길이, 화음 등을 추가할 수 있다. 음표는 “[60]”과 같이 숫자로도 표현할 수 있는데 옥타브는 이미 수치가 반영되어 있으므로 표시할 필요가 없다. 예를 들어 다음 그림 2를 보면 C6는 “[72]”로 사용한다. 음표의 머리 오른쪽에 작은 점을 붙인 점음표는 음표 옆에 .으로 표시하며 원음표에 1/2 길이를 덧붙인 길이로 연주한다. 음표의 표절 분석 측면에서는 음값이나 음표 중 하나의 값만으로 통일하여 표현하는 것이 좋다. 음표로 분석을 하는 경우에는 옥타브에 대하여 추가 처리가 필요하며 음 값을 사용하는 경우 옥타브가 달라짐에 따라서 같은 멜로디라도 표절 판정이 안 될 수 있다.

Octave	C	C#/Db	D	D#/Eb	E	F	F#/Gb	G	G#/Ab	A	A#/Bb	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

그림 2. 음표 분석
Fig.2. Note Analysis

샤프(sharp; 올림표), 플랫(flat; 내림표), 내추럴(natural; 제자리표), 옥타브 등 분석은 앞의 그림에서 C#과 Db는 모두 유사한 표현이다. 따라서 수치로 표절 분석하는 것이 모호함을 줄일 수 있다. 또한 옥타브를 사용하는 경우 모든 음을 C에서 B로 표현할 수 있다. 그러나 앞서 샤프, 플랫을 사용하여 혼동이 올 수 있으므로 옥타브도 표절 분석에 이점을 고려해야 한다.

화음(Chord)은 그림 3과 같이 표기한다. 화음을 이용한 표절 분석은 인터벌을 활용한다.

Common Name	JFugue Name	Intervals (0 = root)
major	maj	0, 4, 7
minor	min	0, 3, 7
augmented	aug	0, 4, 8
diminished	dim	0, 3, 6
7 th (dominant)	dom7	0, 4, 7, 10
major 7 th	maj7	0, 4, 7, 11
minor 7 th	min7	0, 3, 7, 10
suspended 4 th	sus4	0, 5, 7
suspended 2 nd	sus2	0, 2, 7
6 th (major)	maj6	0, 4, 7, 9
minor 6 th	min6	0, 3, 7, 9
9 th (dominant)	dom9	0, 4, 7, 10, 14
major 9 th	maj9	0, 4, 7, 11, 14
minor 9 th	min9	0, 3, 7, 10, 14
diminished 7 th	dim7	0, 3, 6, 9
add9	add9	0, 4, 7, 14
minor 11 th	min11	0, 7, 10, 14, 15, 17
11 th (dominant)	dom11	0, 7, 10, 14, 17
13 th (dominant)	dom13	0, 7, 10, 14, 16, 21
minor 13 th	min13	0, 7, 10, 14, 15, 21
major 13 th	maj13	0, 7, 11, 14, 16, 21
7-5 (dominant)	dom7<5	0, 4, 6, 10
7+5 (dominant)	dom7>5	0, 4, 8, 10

그림 3. JFugue에서 지원하는 화음
Fig.3. Chord of JFugue System

화음 자리바꿈(chord inversion)은 바탕음(음의 높이를 고정하기 위하여 기준으로 삼는 음) 이외의 음을 베이스에 두는 것이다. 그림 4는 C-Major 화음 자리 바꿈을 표기한 것이다. 화음 관련 내용은 이번 표절 분석에서는 다루지 않으므로 세부 설명은 생략한다.



그림 4. C-Major 화음 자리바꿈
Fig.4. C-Major Chord inversion

길이(Duration)는 음표 다음에 표기하는 실제적 시간 값이다. 예를 들어 C6의 1/2음표는 C6h로, D-flat major 화음의 온음표는 DbmajW로 표기한다. 길이가 짧은 그림 5와 같다. 비교를 위하여 실제 음표를 그림 6에 넣었다. 길이는 수치로도 표기할 수 있다. 예를 들어 A4 음의 4분음표는 A4/0.25로 표기한다. 1.0은 온음(whole)이다. 표절 분석을 할 때는 음표와 마찬가지로 수치표현만을 사용한다.

Duration	Character
whole	w
half	h
quarter	q
eighth	i
sixteenth	s
thirty-second	t
sixty-fourth	x
one-twenty-eighth	o

그림 5. 음표 길이
Fig.5. Note Duration

민음표(음표)

명칭	기보	길이(박수)	게이트 타임(j=480)	대응하는 음표
온음표 whole note	♩	4박 (한 마디 전체)	1920	♩
2분음표 half note	♪	2박 (한 마디의 1/2)	960	♪
4분음표 quarter note	♫	1박 (한 마디의 1/4)	480	♫
8분음표 eighth note	♮	1/2 박 (한 마디의 1/8)	240	♮
16분음표 sixteenth note	♯	1/4 박 (한 마디의 1/16)	120	♯
32분음표 thirty-second note	♭	1/8 박 (한 마디의 1/32)	60	♭

그림 6. 음표 분석
Fig.6. Note Analysis

셋잇단음표(Triplet), 잇단음표(Tuplet)는 음표 뒤에 *로 표기한다. 다음 그림 7의 첫번째 셋잇단음표는 E_q* F_q* G_q* 또는 E_q*3:2 F_q*3:2 G_q*3:2로 표기한다. 잇단음표는 음표보다 중요도가 떨어지므로 표절 분석에서 활용하지 않는다.



그림 7. 셋잇단음표
Fig.7. Triplet



그림 8. G5q B5q C6q | C6-w | C6-q B5q A5q G5q
Fig.8. G5q B5q C6q | C6-w | C6-q B5q A5q G5q

붙임줄(Tie)은 높이가 같은 음 두 개의 음 위나 아래에 붙는 표시인데 두 개의 음은 끊지 않고 하나의 음으로 간주한다. 붙임줄은 음표 뒤에 -로 표기한다. 그림 8에서 각 마디는 |로 표기하며 세 마디는 C음표를 -로 붙임줄 표시한다. 표절 분석 시 붙임줄 표시는 수치로 하나의 음으로 사용한다.

[어택(Attack)과 감쇄(Decay) 속도(Velocity)] 음표 뒤에 어택은 a, 감쇄는 d로 표시한다. 예를 들어 C5qa0d127은 빠른 어택, 느린 감쇄, E3wwd0는 디폴트 어택, 빠른 감쇄, C7maja30은 30 어택으로 연주한다. 잇단음표와 마찬가지로 중요도가 떨어지므로 어택과 감쇄 속도는 표절 분석에서 이러한 점을 고려해야 한다.

[멜로디와 화음] MusicString에서 음표와 음표사이를 스페이스로 표기하면 각 음을 순서대로 연주한다(그림 9). 스페

이스 대신 +를 사용하면 화음으로 한 번에 연주한다(그림 10).



그림 9. C5q E5q G5q
Fig.9. C5q E5q G5q



그림 10. C5q+E5q+G5q
Fig.10. C5q+E5q+G5q

화음과 멜로디를 같이 연주하는 경우 +와 _를 사용한다. 그림 11은 C5 2분음표와 E54분음표를 화음으로 연주하고 연달아 G5 4분음표를 연주한다.



그림 11. C5h+E5q_G5q
Fig.11. C5h+E5q_G5q

마디(Measure)는 |로 표시한다. 표절 분석시에는 마디 구분은 모두 없애고 음의 길이를 수치로 변환하여 사용한다.

조표(Key Signature)는 K다음 maj나 min으로 표기한다. JFugue에서는 키값이 바뀌면 자동으로 음표값이 바뀌므로 따로 처리하지 않는다. 악기(Instrument)는 I다음에 악기명 혹은 악기 숫자로 표시한다. 이번 연구에서는 주 멜로디를 연주하는 악기만을 선택하여 표절 분석하도록 한다. 다음 그림 11은 악기별 샘플 값을 넣은 것을 보여준다.

Piano		Chromatic Percussion	
0	PIANO or ACOUSTIC_GRAND	8	CELESTA
1	BRIGHT_ACOUSTIC	9	GLOCKENSPIEL
2	ELECTRIC_GRAND	10	MUSIC_BOX
3	HONKEY_TONK	11	VIBRAPHONE
4	ELECTRIC_PIANO or ELECTRIC_PIANO1	12	MARIMBA
5	ELECTRIC_PIANO2	13	XYLOPHONE
6	HARPISCHORD	14	TUBULAR_BELLS
7	CLAVINET	15	DULCIMER
Organ		Guitar	
16	DRAWBAR_ORGAN	24	GUITAR or NYLON_STRING_GUITAR
17	PERCUSSIVE_ORGAN	25	STEEL_STRING_GUITAR
18	ROCK_ORGAN	26	ELECTRIC_JAZZ_GUITAR
19	CHURCH_ORGAN	27	ELECTRIC_CLEAN_GUITAR
20	REED_ORGAN	28	ELECTRIC_MUTED_GUITAR
21	ACCORDIAN	29	OVERDRIVEN_GUITAR
22	HARMONICA	30	DISTORTION_GUITAR
23	TANGO_ACCORDIAN	31	GUITAR_HARMONICS

Synth Effects		Ethnic	
96	FX_RAIN OR RAIN	104	SITAR
97	FX_SOUNDTRACK or SOUNDTRACK	105	BANJO
98	FX_CRYSTAL or CRYSTAL	106	SHAMISEN
99	FX_ATMOSPHERE or ATMOSPHERE	107	KOTO
100	FX_BRIGHTNESS or BRIGHTNESS	108	KALIMBA
101	FX_GOBLINS or GOBLINS	109	BAGPIPE
102	FX_ECHOES or ECHOES	110	FIDDLE
103	FX_SCI-FI or SCI-FI	111	SHANAI
Percussive		Sound Effects	
112	TINKLE_BELL	120	GUITAR_FRET_NOISE
113	AGOGO	121	BREATH_NOISE
114	STEEL_DRUMS	122	SEASHORE
115	WOODBLOCK	123	BIRD_TWEET
116	TAIKO_DRUM	124	TELEPHONE_RING
117	MELODIC_TOM	125	HELICOPTER
118	SYNTH_DRUM	126	APPLAUSE
119	REVERSE_CYMBAL	127	GUNSHOT

그림 11. 샘플:악기 숫자값
Fig.11. Sample: Integer of Instrument

음원(Voice)은 채널 혹은 트랙이라고도 한다. 미디어에서는 동시에 16채널을 지원한다. 각 음원은 멜로디를 가지고 있으며 특정 악기를 주로 사용한다. JFugue에서는 V와 O에서 15사이의 값을 사용한다. 미디어에서 작업을 하는 경우 채널의 16밖에 없기 때문에 여러 자원을 번갈아 사용하기도 한다. 예를 들어 한 채널에 여러 악기를 번갈아 사용하기도 한다. 이번 연구에서는 주 멜로디만을 사용하므로 주 멜로디 악기만을 선택하여 분석한다.

2. 표절 분석에서 제외될 음원

V9는 퍼커션(타악기) 전용 트랙이다. 이번 연구에서는 주 멜로디만을 분석하므로 V9 퍼커션 트랙도 제외한다. 다음 그림 12는 퍼커션 악기 리스트이다. 이 리스트의 악기는 분석에서 제외한다.

레이어는 서로 독립적인 음원을 겹쳐서 목적하는 음을 생성하는 것이다. 레이어는 동시에 16채널 밖에 사용할 수 없는 한계를 극복하기 위해 사용하는 것으로 L과 O와 15사이의 값을 사용한다. 레이어는 표절 분석에서 제외한다.

Note Value	JFugue Constant	Note Value	JFugue Constant
35	ACOUSTIC_BASE_DRUM	59	RIDE_CYMBAL_2
36	BASS_DRUM	60	HI_BONGO
37	SIDE_KICK	61	LOW_BONGO
38	ACOUSTIC_SNARE	62	MUTE_HI_CONGA
39	HAND_CLAP	63	OPEN_HI_CONGA
40	ELECTRIC_SNARE	64	LOW_CONGO
41	LOW_FLOOR_TOM	65	HIGH_TIMBALE
42	CLOSED_HI_HAT	66	LOW_TIMBALE
43	HIGH_FLOOR_TOM	67	HIGH_AGOGO
44	PEDAL_HI_TOM	68	LOW_AGOGO
45	LOW_TOM	69	CABASA
46	OPEN_HI_HAT	70	MARACAS
47	LOW_MID_TOM	71	SHORT_WHISTLE
48	HI_MID_TOM	72	LONG_WHISTLE
49	CRASH_CYMBAL_1	73	SHORT_GUIRO
50	HIGH_TOM	74	LONG_GUIRO
51	RIDE_CYMBAL_1	75	CLAVES
52	CHINESE_CYMBAL	76	HI_WOOD_BLOCK
53	RIDE_BELL	77	LOW_WOOD_BLOCK
54	TAMBOURINE	78	MUTE_CUICA
55	SPLASH_CYMBAL	79	OPEN_CUICA
56	COWBELL	80	MUTE_TRIANGLE
57	CRASH_CYMBAL_2	81	OPEN_TRIANGLE
58	VIBRASLAP		

그림 12. V9에서 사용하는 퍼커션 악기
Fig.12. Percussion used in V9

템포는 연주 속도를 지정한다. T와 템포 값으로 표시한다. 템포는 멜로디 연주 자체와는 우선순위가 떨어지므로 표절분석에서 사용하지 않는다.

JFugue Constant	Beats Per Minute (BPM)
Grave	40
Largo	45
Larghetto	50
Lento	55
Adagio	60
Adagietto	65
Andante	70
Andantino	80
Moderato	95
Allegretto	110
Allegro (default)	120
Vivace	145
Presto	180
Pretissimo	220

그림 13. 템포 값
Fig.13. Tempo Value

핏치 휠은 음 높이를 조정하는 기능으로 &와 1에서 16383 사이의 값으로 표시한다. 0~8191은 저음, 8193~16383은 고음으로 연주한다. 핏치 휠은 분석에서 제외한다.

채널 프레저는 지정한 음원채널의 압력강도이다. +와 0에서 127사이의 값으로 표시한다. 폴리 프레저는 각 음마다 압력을 넣는 방식이다. *과 0에서 127사이의 값과 쉼표 그리고 0에서 127사이의 값으로 표시한다. 예를 들면 *60,75는 음표값 60에 75의 압력을 준다. 채널과 폴리 프레저는 연주 기법으로서 표절 분석 멜로디 자체와는 중요도가 떨어지므로 분석에서 제외한다. 미디 스펙에는 100가지 이상의 컨트롤러 이벤트가 있는데 X컨트롤러명=값 형식으로 표시한다. 컨트롤러 이벤트는 표절 분석에서 제외한다.

IV. 음악 표절 분석 시스템 아키텍처

본 논문에서 설계한 음악 표절 검사 시스템 아키텍처는 그림 14와 같다.

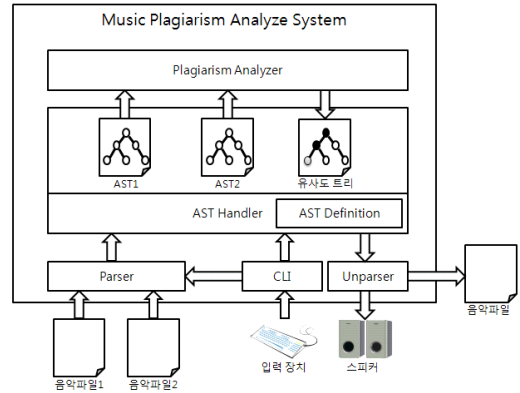


그림 14. 음악표절분석 시스템 모델
Fig.14. System Model of Music Plagiarism Analysis

파서는 음악 파일을 구문 분석하여 적합성을 검사한다. 파서에 의하여 검증된 후에는 AST핸들러에서 음악 파일을 내부 표절 분석기나 역파서에서 처리할 수 있도록 만든 문법 트리인 AST정의(Abstract Syntax Tree Definition; ASTD)를 참조하여 AST를 생성한다. 사용자는 명령어 처리기(Command Line Interpreter; CLI)로부터 파일 파싱이나 표절 분석 혹은 결과를 출력하는 명령을 입력 받는다. 표절 분석기(Plagiarism Analyzer)는 2개의 AST를 표절 비교한 후 결과를 AST형식의 유사도 트리로 만든다. 이 유사도 트리는 역파서(Unparser)를 통하여 표절 부분을 스피커로 듣거나 음악파일로 출력하게 된다.

1. 뮤직스트링(MusicString)

MusicString은 음악 라이브러리인 JFugue에서 지원하는 포맷으로서 프로그래머가 빠르게 음악을 만들 수 있다. 예를 들어 도(C) 연주는 다음과 같이 간단히 표현 한다.

```
Player payer = new Player();
payer.play("C");
```

JFugue는 MusicString을 파싱하여 각 음표를 연주한다. 따라서 MusicString은 연주에 초점을 맞춘 문법을 가지고 있으므로 AST로 변환하여 음악 표절 검사 및 부분 연주에 적합하다. MusicString 음악 파일을 AST노드로 생성한 후 표절 분석을 하기 위하여 음악요소를 수치화할 때 고려할 사항은 다음과 같다. 음표는 단음 위주의 멜로디를 분석대상으로 한다. 음표 문자는 분석에 모호함이 있으므로 모두 수치로 변환하여 사용한다. 잇단음표는 단음으로 변환한다. 화음

(chord)은 제외한다. 음의 길이는 멜로디를 구성하는데 큰 영향을 가진다. 따라서 길이도 분석에 이용한다. 모든 길이 상수는 수치로 변환하여 하나의 값으로 변환한다. 여러 악기 연주를 분석에 사용하면 복잡도가 증가하므로 주 멜로디 악기를 찾아서 한 음원만 사용한다. 따라서 피커션 트랙인 V7도 제외한다. 연주관련 기능인 템포, 핏치 휠, 채널 프레저, 폴리 프레저, 컨트롤 이벤트는 초기 분석에서 제외한다.

2. 추상 구문 트리 정의(ASTD; Abstract Syntax Tree Definition)

음악 표절 분석의 핵심은 MusicString Grammar를 문법 규칙의 집합인 추상 문법(abstract syntax)으로 정의하는 것이다. 각 음악 파일은 문법에 따라서 유도 트리(derivation tree), 즉 AST로 표현된다.

다음은 그림 15는 MusicString Grammar의 일부를 ASTD로 변환한 것이다.

<pre>musicstring ::= element_decl_list ; element_decl_list ::= element element_decl_list element ; element ::= voice tempo instrument layer key controller time poly_pressure channel_pressure pitch_bend measure expression system_exclusive collected_note note ; <Music String Grammar의 일부></pre>	<pre>main : MainMusicString(music_string) ; music_string* : MusicStringNil() MusicStringList(element music_string) ; element : ElementNil() ElementVoice(voice) ElementTempo(tempo) ElementInstrument(instrument) ElementLayer(layer) ElementKey(key) ElementController(controller) ElementTime(time) ElementPolyPressure() ElementChannelPressure(channel_pressure) ElementPitchBend(pitch_bend) ElementMeasure(measure) ElementExpression(expression) ElementSystemExclusive(system_exclusive) ElementCollectedNote(collected_note) ElementNote(note) ; <변환된 ASTD></pre>
--	--

그림 15. MusicString Grammar의 ASTD 변환
Fig.15. ASTD Translation of MusicString Grammar

비단말의 집합으로 이루어진 유도 트리를 term이라고 부른다. 다음은 "C5h"에 대한 항 연산을 표현한 term이다. 이때 Nil연산의 term에서는 괄호를 생략했다.

MainMusicString(MusicStringList(ElementNote, ElementNil))

3. 추상 구문 트리(Abstract Syntax Tree)

AST는 MusicString 파일을 파싱하여 음악 표절 분석이 가능하도록 설계한 자료구조이다. 음악 파일간 분석 결과는 모두 AST 인스턴스에 반영된다. 다음 그림 16의 예는 앞서

정의한 ASTD를 통해서 그림 17과 같은 AST 인스턴스를 생성하게 된다.

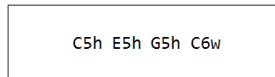


그림 16. 도미솔도의 MusicString
Fig.16. MusicString of Do Mi Sol Do

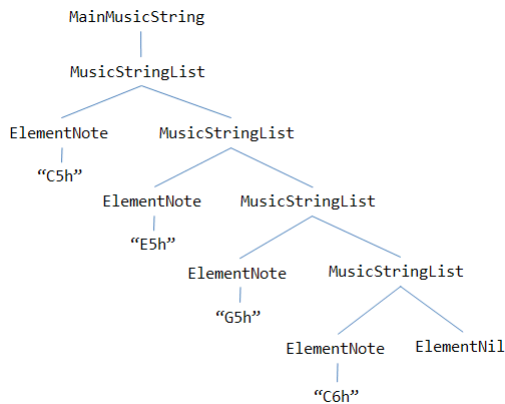


그림 17. MusicString의 AST 인스턴스
Fig.17. AST Instance of MusicString

4. 뮤직스트링 BNF 문법 구현

본 논문에서 음원 분석을 위해 다음과 같이 BNF를 만들어 활용하고자 한다.

musicstring := (element whitespace)+ element?

element := voice | tempo | instrument | layer | key | controller | time | poly_pressure | channel_pressure | pitch_bend | measure | expression | system_exclusive | collected_note

voice := "V" int_or_const
tempo := "T" int_or_const
instrument := "I" int_or_const
layer := "L" int_or_const
time := "@" int_or_const
poly_pressure := "*" int_or_const
channel_pressure := "+" int_or_const

```
pitch_bend      := "&" int_or_const

measure        := "|"

controller     := "X" int_or_const "="
               controller_value

controller_value := int | symbol
```

... 이하 생략 ...

5. 유사도 분석 알고리즘

본 연구에서 활용하고자 하는 알고리즘은 [12]에서 연구한 알고리즘을 이용하고자 한다.

```
double Sim(NodeString A, NodeString B, long int
minlength) {
    String matchstring, totalmatchstring; /* 일치된 스트링 */
    int maxmatch = 0; /* 일치된 스트링의 개수 초기화 */
    long int matchlength = 0; /* 일치된 스트링의 전체 개수 초기화 */
    Set(totalmatchstring) = {}; /* 일치되는 전체 스트링 집합*/

    /* 일치되는 스트링을 찾을 때까지 알고리즘 1, 2 반복 */
    do {
        matchstring = ""; /* 일치되는 스트링 */
        matchstring = MatchString(A, B); /* 알고리즘 1 호출 */
        Set(totalmatchstring) = Set(totalmatchstring) +
matchstring;
    } while (maxmatch > minlength);

    /* 일치되는 스트링의 총 개수 계산*/
    for each matchstring in Set(totalmatchstring)
        matchlength = matchlength +
Length(matchstring);
    end for
    /* 유사도 값 계산 및 반환 */
    return (2 *  $\frac{totalMatchSize}{Length(A) + Length(B)}$ );
}
```

IV. 결론

본 논문에서는 현재 논란이 지속되고 있는 음악의 표절에 관련된 내용을 연구 분석하였다. 음악 표절은 과거, 현재 뿐만 아니라 미래에 조차 많은 논란 여지를 가지고 있다. 따라서 본 연구에서는 국내외의 음악 저작권 보호 사례에 대해서 살펴보았다. 또한 두 MIDI 파일의 표절 분석을 어떻게 수행할 것인지 연구하였으며, 표절 분석을 위한 음원 요소 분석을 수행하였다. 두 음원의 표절 분석을 위해서 어떠한 IT 요소들이 필요한지를 살펴보았으며, 실제 구현을 위한 요소들을 분석하였다.

따라서 본 연구는 음악 표절 분석을 위한 선행 연구로 JFugue MusicString를 추상 구문 트리를 생성하기 위한 음악 요소와 표절 분석을 위한 데이터 생성에 대하여 심도있게 기술하였다.

본 논문은 다음과 같은 기대효과를 가질 수 있다. 첫째 과거와 현재뿐만 아니라 미래까지 논란이 계속될 수 있는 음악 관련 표절 분야의 이정표를 줄 수 있다. 둘째, 지금까지 감성적으로만 판단할 수밖에 없는 음악 표절 시비 논란을 정형화된 수치로 제공하여 새로운 해결책의 지침용이 될 수 있다. 셋째, 예술(음악)과 IT 분야의 융복합 학문을 통하여 새로운 기반 기술을 연구하는 것에 대한 의의를 들 수 있다. 넷째, 음악에 대한 작곡의 초기 단계부터 불법 표절 관행 근절할 수 있다. 다섯째, 표절 관행 미연에 방지 효과, 창작에 대한 극대화, 국가 지정 '공정한 문화국가 초석 마련'할 수 있다.

향후 연구과제는 AST 생성 후 트리 최적화와 역파싱 및 표절 분석 방법을 통한 음알 표절 분석에 대하여 연구되어야 할 것이다.

참고문헌

- [1] International Copyright Technology Conference 2012, <http://www.icotec.or.kr>
- [2] Y. C. Kim, J. Y. Choi, "A Program Plagiarism Evaluation System", ICCSA 2005, Lecture Note of Computer Science 3483, 2005.
- [3] "The Plagiarism Checker" available at <http://www.dustball.com/cs/plagiarism.checker>, 2010.
- [4] Wikipedia Site available at <https://en.wikipedia.org/wiki/MIDI>.

- [5] Dong-Chul Kwack, "A Study on the Types of Plagiarism and Appropriate Citation Practices of Writing Research Papers", Korean Society for Library and Information Science, 2007.
- [6] P. J. Larkham, & Manns, "S. Plagiarism and its treatment in higher education." Journal of Further and Higher Education, 26(4), 339-349. 2002.
- [7] D. L. McCabe, L. K. Trevino, & K. D. Butterfield, "Cheating in academic institutions: A decade of research.", Ethics & Behavior, 11(3), 219-232. 2001.
- [8] Korea Federation of Copyright Organization, "2012 Copyright Protection Annual Report", 2012
- [9] Music XML 3.0 Site available at <http://www.musicxml.com>.
- [10] David Koelle, The Complete Guide to JFugue: Programming Music in Java, 2008, pp. 21-48.
- [11] JaeKyoungLee, "The song 'loner' event - Music Copyright Plagiarism Judgement- Seoul Central District Court 2011. 4. 13", Chungang Law Association Vol 13. 4, 2011.
- [12] Y. C. Kim, J. W. Yoo, "A Tree-Compare Algorithm for Similarity Evaluation", KIPS, Vol.11-A No. 2, 2004.

저 자 소 개



신 미 해
 2008: 평생교육진흥원
 컴퓨터공학과 공학사.
 2012: 공주대학교
 컴퓨터공학과 공학석사.
 현 재: 공주대학교
 컴퓨터교육과 박사과정
 관심분야: 멀티미디어콘텐츠,
 프로그래밍
 E-mail : talsalgo@nate.com



조 진 완
 2007: 공주대학교 관광경영학과 학사.
 2009: 공주대학교
 전자상거래학과 석사.
 현 재: 공주대학교
 전자상거래학과 박사과정
 관심분야: 가상현실, 소셜미디어,
 전자상거래, ICT
 E-mail : jinwan84@gmail.com



이 혜 승
 1999: 건국대학교
 응용생물화학학과 학사.
 2001: 건국대학교
 응용생물화학전공 석사.
 2000.9~2003.6: 서울대학교
 서울대학교 NICEM연구원
 현재: 유한대학교 산학협력단 연구원
 관심분야: 응용화학, 응용통계
 E-mail : hyeseung@nate.com



김 영 철
 1990: 한남대학교 전자계산학과 학사.
 1996: 숭실대학교 컴퓨터학과 석사.
 2003: 숭실대학교 컴퓨터학과 박사
 현재: 유한대학교 e-비즈니스과 교수
 관심분야: 프로그래밍 언어, 망관리,
 컴파일러, XML, 컴퓨터통신
 E-mail : kim0725@yuhan.ac.kr