



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

석사학위논문

우두 드럼의 실시간 사운드 프로세싱을
이용한 멀티미디어음악 작품 제작 연구
(멀티미디어음악 작품 <Cosmos>를 중심으로)

지도교수 김 준

동국대학교 영상대학원
멀티미디어학과 컴퓨터음악전공

김진우

2021

석사학위논문

우두 드럼의 실시간 사운드 프로세싱을
이용한 멀티미디어음악 작품 제작 연구
(멀티미디어음악 작품 <Cosmos>를 중심으로)

김진우

지도교수 김준

이 논문을 석사학위 논문으로 제출함

2020년 12월

김진우의 음악석사(컴퓨터음악)학위 논문을 인준함

2021년 1월

위원장 박상훈



위원 정진헌



위원 김준



동국대학교 영상대학원

목 차

I . 서 론	1
1. 연구 배경 및 목적	1
2. 사례 및 선행 연구	3
1) 사례 조사	3
2) 선행 연구	5
II . 본 론	8
1. 악기 특성 분석	8
1) 우두(udu)의 유래	9
2) 우두(udu)의 타격 지점과 연주 방식에 따른 주파수 분석	9
2. 사운드 디자인	12
1) Max를 이용한 악기 제작 연구	12
2) 사운드 프로세싱 연구	16
① granular synthesis	16
② pitch shift	19
③ ring modulation	20
④ delay	21
⑤ comb filter	23
⑥ reverb	24
⑦ FFT분석을 이용한 주파수 분리(xover filter)	25
3) 사운드 프로세싱의 유기적 연결을 위한 Max for Live의 활용 ..	27
3. 영상 시스템 연구	31
1) 영상 시스템	31

2) 영상제작 및 효과	34
① Unity를 이용한 영상제작	34
② Processing을 이용한 영상제작	36
③ After Effects를 이용한 영상제작	38
④ Arena를 이용한 영상효과	39
3) OSC 통신을 이용한 영상의 실시간 제어	40
① Unity 영상의 실시간 제어	41
② Processing 영상의 실시간 제어	43
③ Arena 영상효과의 실시간 제어	45
4) Syphon을 이용한 Arena와의 연동	46
① Unity와 Arena의 연동	46
② Processing과 Arena의 연동	47
4. 연구 기술의 작품 적용	48
1) 작품 소개	48
2) 작품 구성 및 기술 적용	49
① 음악 구성 및 기술 적용	49
② 영상 구성 및 기술 적용	51
③ 무대 및 시스템 구성	55
3) 작품에서의 기술 적용에 따른 예술적 효과	56
Ⅲ . 결 론	60
참 고 문 헌	62
ABSTRACT	64
부록(첨부 DVD 설명)	66

그림 목 차

[그림-1] <Koan Sound : Polychrome>	3
[그림-2] <Odesza + Sofi Tukker + Louis Futon @ Mtelus>	4
[그림-3] 작품 <Field>의 영상 캡처	5
[그림-4] 우두 드럼의 모습	8
[그림-5] 오실레이터(oscillator)를 위해 제작한 서브패치	13
[그림-6] 가산합성 방식을 사용한 사운드 패치	14
[그림-7] 감산합성 방식을 사용한 사운드 패치	15
[그림-8] 오리지널 사인파와 granular synthesis로 합성한 파형	17
[그림-9] munger~오브젝트를 활용한 granular synthesis 패치	18
[그림-10] pitchshift~오브젝트를 활용한 패치	19
[그림-11] ring modulation의 시스템	20
[그림-12] ring modulation을 구현한 패치	21
[그림-13] tapin~, tapout~을 활용한 delay 패치	22
[그림-14] comb~오브젝트를 활용하여 구현한 comb filter 패치	23
[그림-15] reverb의 구성	24
[그림-16] yafr2오브젝트를 활용한 reverb 패치	25
[그림-17] xover filter 패치	26
[그림-18] aux 트랙을 활용한 send/return 방식의 시스템	27
[그림-19] 작품에 활용한 aux track 설계 시스템의 예	28
[그림-20] 작품에 활용한 Ableton Live 프로젝트의 Session View	29
[그림-21] 영상 시스템 설계도	33
[그림-22] Unity 프로젝트의 주요 화면	34
[그림-23] Processing으로 제작한 영상 타입 1의 변화	36
[그림-24] Processing으로 제작한 영상 타입 2의 변화	37

[그림-25] After Effects로 제작한 영상 이미지	38
[그림-26] Max와 Unity, Processing의 OSC 통신을 위한 패치	40
[그림-27] OSC 통신을 받기 위한 Unity의 스크립트	42
[그림-28] Processing의 oscp5 라이브러리 설치 화면	43
[그림-29] OSC 라이브러리를 불러오는 Processing 코드	43
[그림-30] OSC 통신에 필요한 Processing의 주요 설정 코드	44
[그림-31] OSC 입력 데이터를 변수로 설정하는 코드	44
[그림-32] Max와 Arena의 연동	45
[그림-33] Syphon을 이용한 Unity와 Arena의 연동	46
[그림-34] Syphon을 이용한 Processing과 Arena의 연동	47
[그림-35] 작품 <Cosmos>의 실제 공연 이미지 1	48
[그림-36] A파트의 영상 변화	51
[그림-37] B파트의 영상 변화	52
[그림-38] C파트의 영상 변화	53
[그림-39] C'파트의 영상 변화	53
[그림-40] B'파트의 영상 변화	54
[그림-41] A'파트의 영상 변화	54
[그림-42] 무대 및 시스템 구성	55
[그림-43] 작품 <Cosmos>의 실제 공연 이미지 2	59

표 목 차

<표-1> 우두 드럼의 연주 방식에 따른 주파수 분석과 음색	10
<표-2> 세 종류의 프로그램을 사용하여 제작한 각각의 이미지와 효과	31
<표-3> Unity로 제작한 영상의 주요 표현 효과	35
<표-4> Arena의 영상효과	39
<표-5> 작품의 음악 구성	51
<표-6> A파트, B파트의 기술 적용 및 효과	57
<표-7> C파트, C'파트의 기술 적용 및 효과	57
<표-8> B'파트, A'파트의 기술 적용 및 효과	58

I. 서론

1. 연구 배경 및 목적

19세기 초까지 전개된 산업혁명 이후 모든 대중문화는 과학 기술의 발전과 함께 성장해왔다. 21세기를 넘어서며 기술의 발전 속도는 더욱 빨라지고 있으며 오디오, 영상, 도서 등 대부분의 매체들이 아날로그에서 디지털로 대체되었다. 이에 따라 두 가지 이상의 복합감각을 활용하여 감상할 수 있는 멀티미디어(multimedia)¹⁾ 예술 작품들 또한 대중들에게도 더 이상 낯설지 않은 시대가 되었다. 특히 청각적 매체의 경우, 시각적 정보를 함께 수용하며 감상하는 것이 어렵지 않다. 사람들은 공연장에서 음악을 감상할 때 스크린과 조명을 통해 시각적 정보 또한 함께 받아들이며, 컴퓨터 모니터나 휴대폰을 통해 시청각 매체를 감상할 때 역시 마찬가지로 청각적, 시각적인 정보를 함께 수용하여 스스로 인지하지 못하는 사이에 인터랙션(interaction)²⁾이 이루어지고 있다.

음악의 예술적 순수성을 고려할 때, 청각적 요소로 인해 듣는 사람으로 하여금 다양한 감각의 정보를 스스로 만들어낼 수 있게 하는 것이 바람직하다. 그 정보들 또한 감상자가 가지고 있는 기억과 경험에 따라 다르게 나타날 수 있는 것이 자연스럽다. 하지만 멀티미디어 콘텐츠(multimedia contents)³⁾는 기술의 발전에 따른 자연스러운 흐름이며 미래에는 더욱 발전할 가능성을 가지고 있다.

1) 다양한 형태의 미디어 정보가 혼합된 디지털 매체를 말한다. multi(다중, 복합)와 media(매체, 매개물)의 합성어로서, 컴퓨터가 처리할 수 있는 디지털 상태의 것을 포괄한다.

2) 2개 이상의 매체가 서로 영향을 미치는 작용으로, 상호작용을 뜻한다.

3) 오디오, 그래픽, 텍스트 등의 서로 다른 정보 전달 수단들이 통합된 것으로 다양한 정보기기를 통해 생산, 유통, 소비된다.

본 논문에서 다루어질 작품 <Cosmos>는 멀티미디어음악 작품으로, 현대 시대의 예술적 흐름에 맞는 작품 구상 및 멀티미디어 공연 제작에 초점을 맞춘 작품이다. 타악기 군에 속하는 우두 드럼(udu drum)⁴⁾ 연주와 Max⁵⁾를 사용하여 제작한 사운드가 실시간으로 프로세싱되며 새로운 형태의 사운드를 만들고, 그에 맞춰 반응하는 영상의 인터랙션을 통해 새로운 표현 방식과 발전 가능성을 제시하고자 한다.

멀티미디어 작품 <Cosmos>에 실시간 연주로 사용된 악기는 아프리카 타악기 군에 속하는 우두 드럼이다. 항아리 형태의 모양을 하고 있으며 타격지점과 연주 방식에 따라 저음역대부터 고음역대까지 비교적 다양한 음색과 질감을 만들 수 있어 음향 효과를 적용하기에 적합하다.

본 논문에서는 Max를 활용한 악기 제작 기술의 연구, 사운드 프로세싱, 그리고 소리의 시각화를 위해 활용된 Unity⁶⁾, Processing⁷⁾, After Effects⁸⁾, Arena⁹⁾를 작품에 적용한 방식과 그에 따른 효과를 다루고자 한다.

-
- 4) 나이지리아 동남부에서 전해져온 항아리 형태의 물 주전자로, 여성들이 의례적인 용도로 연주했다.
 - 5) Cycling '74에서 개발한 것으로, 오디오, 비주얼, 미디어 및 물리적 컴퓨터 사용을 다루는 예술가, 교육자 및 연구원들을 위한 프로그램이다. 음향신호를 컨트롤하는 MSP와 real-time video, 3D 그래픽을 다루는 Jitter로 이루어져 있다.
 - 6) Unity는 3D 및 2D 비디오 게임의 개발 환경을 제공하는 게임 엔진으로, 3D 애니메이션과 건축 시각화, 가상현실 등 인터랙티브 콘텐츠 제작을 위한 통합 저작 도구이다.
 - 7) 2001년 MIT 미디어 랩의 'Aesthetics and Computation' 그룹이 개발한 오픈소스 프로그래밍 언어이다.
 - 8) Adobe가 개발한 디지털 모션 그래픽 및 합성 소프트웨어이다. 영화의 비선형 영상 편집이나 광고 제작, TV, 게임, 애니메이션, 웹 등의 콘텐츠 제작에 쓰인다.
 - 9) Resolume에서 개발한 영상 믹싱 프로그램으로 실시간 비주얼 퍼포먼스를 위해 사용, 본 연구에서는 Arena6를 사용한다.

2. 사례 및 선행 연구

1) 사례 조사

오늘날 실시간 사운드 프로세싱(real-time sound processing)을 활용함과 동시에 그것을 시각적으로 표현한 멀티미디어 작품은 매우 많다. 예술작품은 감상하는 사람에 따라 주관적으로 받아들일 수밖에 없는 부분이 있지만, 창작자가 표현하고자 하는 것을 한 가지 감각만이 아닌, 두 가지 감각을 함께 사용하여 표현함으로써 감상자에게 좀 더 효과적인 메시지 전달이 가능한 것이다. 따라서 시각적 구성요소가 청각적 구성요소와 상관관계가 클수록 효과적인 멀티미디어 콘텐츠라 할 수 있다.



[그림-1] <Koan Sound : Polychrome>

[그림-1]은 소리를 시각화한 멀티미디어 작품으로, 2019년 영국 브리스톨에서 공연되었다. 브리스톨 출신 듀오 Koan Sound가 음악을 담당하였고, 비주얼은 KŌLAB Studios와 Loz Blackwell이 함께 한 작품이다. Ableton Live¹⁰⁾를 활용하여 실시간으로 사운드를 프로세싱하며 연주하였고, 프로세싱 된 사운드를 Arena와 동기화하여 이미지들을 시각화하였다. 음악으로 관객들을 열광케 하는 동시에 이미지화한 사운드를 스크린과 조명을 사용하여 표현한 효과적인 멀티미디어 공연이다.



[그림-2] <Odesza + Sofi Tukker + Louis Futon @ Mtelus>

[그림-2]는 2017년 캐나다 몬트리올 Mtelus 극장에서 주최한 Odesza, Sofi Tukker, Louis Futon의 합동 공연이다. 이 또한 스크린과 조명만을 이용하여 소리를 시각화한 멀티미디어 작품으로, 실시간으로 프로세싱 되는 오디오를 이미지화하여 스크린에 투영하는 방식이다.

10) 뮤직 시퀀싱 프로그램이자 디지털 오디오 워크스테이션이다. 다른 시퀀싱 프로그램과 달리 라이브 공연을 위한 악기의 역할을 할뿐 아니라 합성, 녹음, 작곡, 믹싱, 마스터링을 위한 도구로 이용할 수도 있다.

본 논문에서 다룬 <Cosmos> 또한 이 작품들과 유사하게 음악이 갖는 그 자체의 순수성을 극대화함과 동시에 그것을 이미지화한 시각 자료를 스크린에 투영하는 방식을 사용하여 제작하였다. 위 작품들은 음악작가와 비주얼작가가 분리되어 작품 전체의 통일성이 부족했지만, <Cosmos>는 음악작가와 비주얼작가를 분리하지 않고 작가 한 사람이 청각 정보와 시각 정보를 동시에 활용함으로써 하나의 멀티미디어 작품으로서의 완결성을 더하였다.

2) 선행 연구

선행 연구로 2019년 12월 우두(udu) 드럼 연주를 활용한 사운드 프로세싱 작품 <Field>를 제작하였다.



[그림-3] 작품 <Field>의 영상 캡처

[그림-3]은 작품 <Field>의 한 장면으로 우두 드럼을 연주하는 모습이며, 사운드 프로세싱에는 delay¹¹⁾, reverb¹²⁾, phase vocoder¹³⁾, granular synthesis¹⁴⁾, comb filter¹⁵⁾가 실시간으로 사용되었다.

우두 드럼 연주만으로도 저음역대부터 고음역대까지 비교적 넓은 주파수 범위의 소리를 만들 수 있었지만, 큰 음량(dB¹⁶⁾) 값을 가지고 연주할 수 있는 소리의 제일 낮은 음역대가 80Hz¹⁷⁾ 부근이기에, 그보다 더 낮은 주파수 배음을 큰 음량 값으로 채우기에는 무리가 있었다. 또한 위 작품은 피아노 연주와의 합주를 위해 시작부터 끝까지 일정한 템포를 갖고 있어 연주 패턴이 박자로부터 자유롭지 못한 부분이 있었고, 사용된 사운드 프로세싱 효과들 또한 독립적으로 작동하게끔 디자인 되어 사운드의 완결성에 있어 다소 아쉬운 면이 있었다.

11) 입력된 사운드를 지정한 시간 후에 출력하는 효과를 말한다.

12) 소리나 신호가 반사되어 쌓이고, 다음 소리가 공간 내의 물체의 표면에 흡수되면서 붕괴될 때 생성되는 효과를 말한다.

13) 음원을 FFT분석을 통해 그 데이터를 재합성하여 음의 높낮이와 길이를 변형시키는 소리 합성방식을 말한다.

14) 음원을 작은 단위(grain)로 나누어 재조합 및 변형하여 다른 형태의 사운드를 제작하는 소리 합성방식이다.

15) 지연된 신호를 추가하여 구현하는 필터로, 주파수 특성에 빗살(comb)모양이 나타나는 필터이다.

16) 소리의 크기를 측정하는 상대적인 표시단위이다.

17) 헤르츠(Hertz, 기호 : Hz)는 주파수 단위로서, 이 이름은 전자기학 분야에서 큰 업적을 남긴 독일 물리학자 하인리히 루돌프 헤르츠(Heinrich Rudolf Hertz)를 기리기 위한 것이다. 1 Hz는 “1초에 한번”을 의미한다.

선행 연구의 결과에 따라 본 작품 <Cosmos>에는 다음과 같은 사항을 보완하였다. 첫째, 우두 드럼 연주가 갖는 타악기적인 한계를 탈피하고 80Hz 이하의 소리를 음악적으로 표현하기 위해 Max로 합성음을 제작하여 함께 사용하였다. 둘째, 일정한 박자로부터 자유로울 수 있도록 작품의 음악적 구조에 카덴차(cadenza)¹⁸⁾형식을 사용하였다. 셋째, 사운드 프로세싱 효과들을 서로 엮어 프로세싱 된 사운드의 완결성에 좀 더 힘을 실을 수 있게 하였다. 넷째, 선행 연구 작품에는 결여되어 있는 소리의 시각화를 꾀하였다.

18) 협주곡에서의 독주부를 뜻하는 것으로, 반주를 멈춘 동안 화려하고 기교적인, 혹은 그 풍을 살린 연주를 통해 독주자의 역량을 과시하는 대목.

II. 본 론

1. 악기 특성 분석

본 작품에 사용한 악기는 우두 드럼으로, 아시아 문화권에서는 생소한 악기인 만큼 악기의 유래에 대해 알아보고 그 특성을 분석할 필요가 있었다.



[그림-4] 우두 드럼의 모습

1) 우두(Udu)의 유래

우두는 나이지리아 동남부에 사는 이그보(Igbo)¹⁹⁾ 민족의 악기이며 Aerophone(공기울림악기)²⁰⁾인 동시에 Idiophone(몸울림악기)²¹⁾이다. 이그보어에서 udu는 ‘vessel’을 의미한다. 원래는 점토로 만든 물을 담아 놓는 항아리였으나 전면부에 구멍을 뚫어 이그보 여성들에 의해 의례적인 용도로 연주되었다. 큰 구멍을 손바닥으로 타격하여 베이스 사운드를 만들어내고 작은 상부 구멍 위의 손이 어떻게 위치하느냐에 따라 음의 높이가 달라지기도 하며 손바닥과 손가락 모두를 사용하여 타격하거나 마찰시켜 연주가 가능하다. 오늘날에는 의례적인 용도를 벗어나 타악기 연주자들에 의해 다양한 음악 스타일에 널리 사용되고 있다.

2) 우두(Udu)의 타격 지점과 연주 방식에 따른 주파수 분석


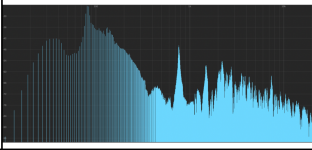

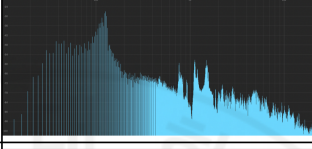

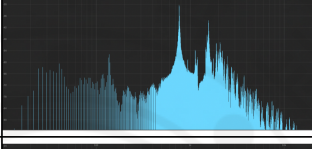
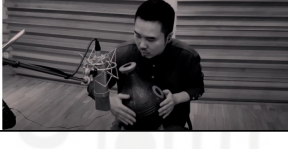
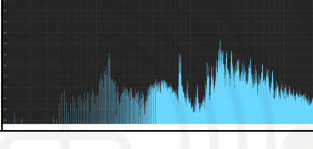
우두는 타격지점과 연주 방식에 따라 저음역대부터 고음역대까지 비교적 다양한 음색을 만들 수 있다. <표-1>은 우두의 타격 지점과 연주 방식에 따라 나타나는 주파수 스펙트럼과 음색의 느낌을 정리한 것이다.

19) 이그보(Igbo, Ibo)는 나이지리아 동남부에 사는 흑인계의 민족이며 아프리카 최대 민족의 하나이다.

20) 공기의 진동을 통해 소리를 내는 악기.

21) 악기의 몸통이 진동하여 소리를 내는 악기.

<표-1> 우두 드럼의 연주 방식에 따른 주파수 분석과 음색

연주법	연주하는 모습	주파수 스펙트럼	연주 방법과 음색
연주법-1			전면부의 큰 구멍을 타격하여 연주 베이스 드럼과 같은 저음역의 소리
연주법-2			상면부의 작은 구멍을 타격하여 연주 낮은 음정의 물방울 같은 소리
연주법-3			측면부의 테두리를 타격하여 연주 밝고 맑은 울림
연주법-4			측면부의 테두리를 문질러 연주할 때 셰이커(Shaker)와 같은 질감의 소리

<표-1>에서 연주법-1과 같이 우두 전면부의 큰 구멍을 타격하여 연주하게 되면 마치 베이스 드럼과 비슷한 저음역의 소리를 낸다. 연주법 1의 주파수 스펙트럼을 살펴보면 80Hz에서 가장 큰 dB값을 갖는 것을 알 수 있다. 그 위로 핵심이 되는 주파수는 750Hz이고, 1600Hz 이상의 배음들도 가득 차 있는 것을 확인할 수 있다.

연주법-2와 같이 우두 상면부의 작은 구멍을 타격하여 연주하게 되면, 낮은 음정의 물방울과 비슷한 소리를 낸다. 마치 스네어 드럼과도 비슷한 질감의 음색을 얻을 수 있다. 연주법 2의 주파수 스펙트럼을 살펴보면 130Hz에서 가장 큰 dB 값을 가지며 그 위로 핵심이 되는 주파수는 750Hz, 1200Hz, 1600Hz이다.

연주법-3과 같이 우두 측면부의 테두리를 타격하여 연주하게 되면 연주법 1과 2보다 밝고 맑은 울림의 소리를 얻을 수 있다. 연주법 3의 주파수 스펙트럼을 살펴보면 핵심이 되는 주파수가 750Hz와 1600Hz인데, 손바닥으로 타격하게 되면 750Hz가 더 큰 dB값을, 손가락으로 두드리게 되면 1600Hz가 더 큰 dB값을 갖는다.

연주법-4와 같이 우두 측면부의 테두리를 문질러 연주하게 되면 마치 셰이커(Shaker)²²⁾와 같은 질감의 소리를 얻을 수 있다. 연주법 4의 주파수 스펙트럼을 살펴보면 1600Hz 이상의 고음역대에 배음이 집중된 것을 알 수 있다.

이처럼 우두는 타악기임에도 불구하고 저음역대부터 고음역대까지 넓은 배음을 가지고 다양한 질감을 연주할 수 있는 악기이기에, 악기의 연주 특성을 그대로 살리면서 여러 가지 사운드 프로세싱 효과를 활용할 수 있었다.

22) 셰이커는 음악에서 리듬을 창조하는데 사용되는 다양한 타악기를 말하며, 소리를 내는 방법에 혼드는 것이 포함되기 때문에 셰이커라고 불린다.

2. 사운드 디자인

1) Max를 이용한 악기 제작 연구

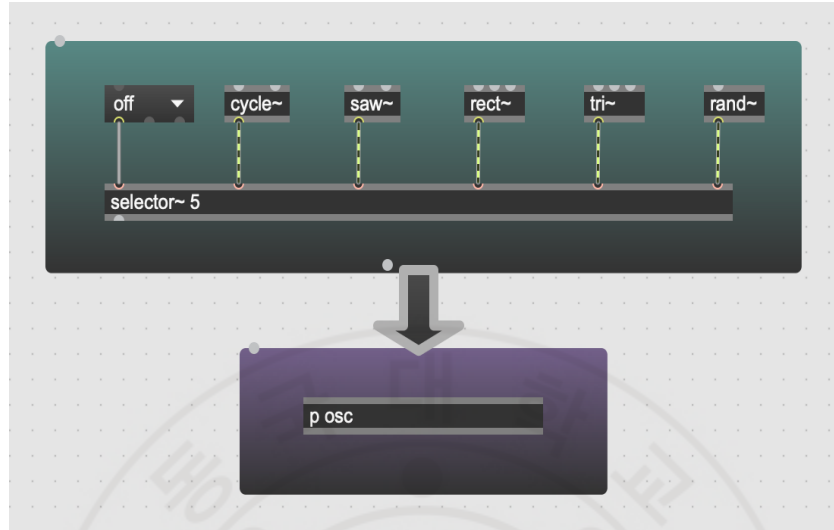
<Cosmos>의 음악에 핵심이 되는 악기는 우두드럼 하나이지만, 타악기가 갖는 음악적인 한계를 벗어나기 위해 Max로 제작한 악기를 함께 활용하였다. 악기의 제작에는 가산합성(additive synthesis)²³⁾ 방식과 감산합성(subtractive synthesis)²⁴⁾ 방식을 함께 사용하였다. 이 악기는 Bass 세 종류, Pad²⁵⁾ 세 종류, Pluck²⁶⁾ 두 종류로 총 여덟 가지이며, 이 악기를 우두 연주에 가미되는 코드와 멜로디로 활용하여 작품에 음악성을 더하였다.

23) 가산합성(additive synthesis)은 여러 개의 사인파를 함께 넣어 음색을 내는 소리 합성 기법이다.

24) 감산합성(subtractive synthesis)은 오디오 신호를 필터에 의해 감쇠시켜 소리의 음색을 바꾸는 소리 합성 기법이다.

25) 부드럽고 지속적인 음색을 내는 신시사이저 패치의 종류를 말하며, 이는 오케스트라의 현악기를 모방한 사운드이다.

26) 현악기를 당김으로써 연주되는 느낌을 구현한 신시사이저 패치의 종류를 말한다.



[그림-5] 오실레이터(oscillator)²⁷⁾를 위해 제작한 서브패치

[그림-5]는 감산합성 방식으로 사용할 각각의 오실레이터를 위해 제작한 패치이다. `selector~` 오브젝트를 활용하여 사인파(sine wave)²⁸⁾, 톱니파(saw wave)²⁹⁾, 사각파(square wave)³⁰⁾, 삼각파(triangle wave)³¹⁾, 노이즈를 선택할 수 있게 한 후, 서브패치로 간소화 하였다.

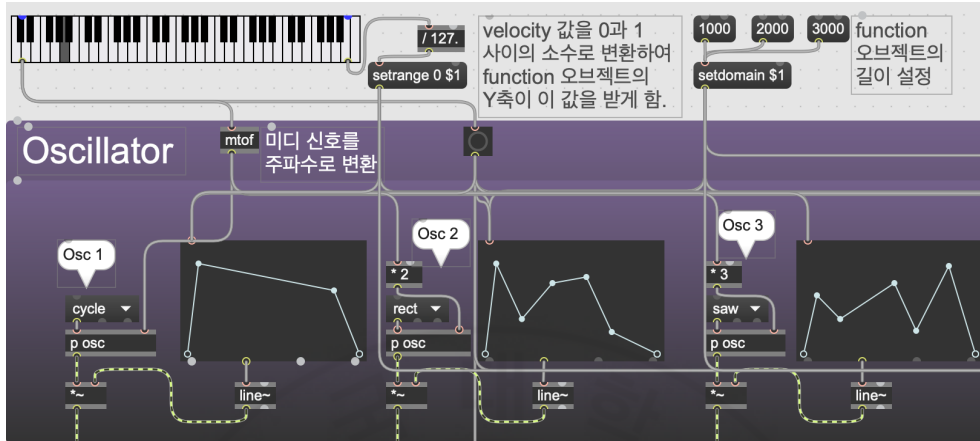
27) 음악적으로 소리를 발생하는 발진기를 말한다.

28) 삼각 함수의 사인 곡선으로 나타나는 파형을 말한다.

29) 톱니 모양으로 나타나는 파형을 말한다.

30) 직사각형 모양으로 나타나는 파형을 말한다.

31) 삼각형 모양으로 나타나는 파형을 말한다.

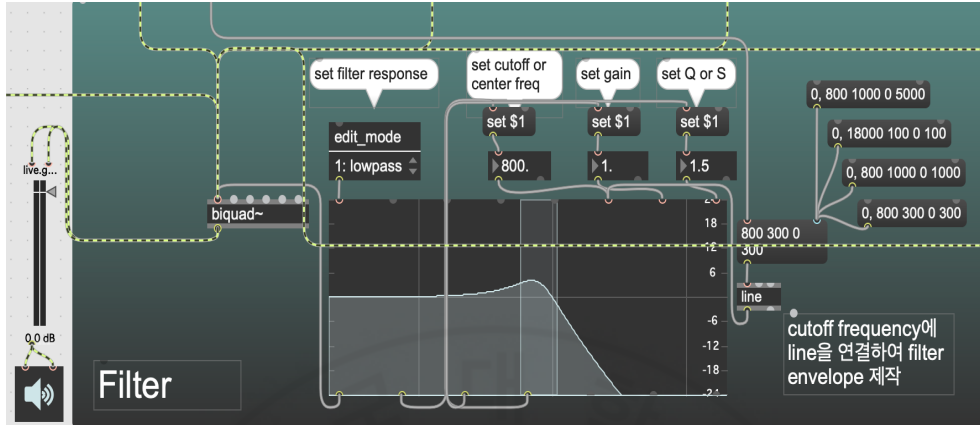


[그림-6] 가산합성 방식을 사용한 사운드 패치

[그림-6]은 가산합성 방식을 사용한 사운드 패치의 일부이다. 앞서 본 [그림-5]에서 제작한 서브패치를 활용하여, 총 6개의 오실레이터가 각기 다른 파형과 엔벨로프(envelope)³²⁾를 갖도록 설계하였다. 건반으로부터 연주되는 미디 신호를 주파수로 변환해 주기 위해 mtof오브젝트를 사용하였고, 엔벨로프를 쉽게 그리기 위해 function오브젝트를 사용하였다. 또한 건반을 연주할 때 발생하는 벨로시티(velocity)³³⁾ 값을 0과 1 사이의 소수로 변환하여, function의 Y축이 이 값을 받게 하였다. 이렇게 하면 그려진 엔벨로프의 최대 음량이 변환된 벨로시티 값으로 도달하게 된다.

32) 소리와 음악에서 엔벨로프(envelope)는 소리가 시간이 지남에 따라 어떻게 변하는지를 묘사할 수 있는 요소이다. 예를 들어, 피아노 건반을 누르는 순간 거의 즉석에서 초기 소리를 발생시켜 음량이 점차 0으로 줄어든다. 간단하게는 attack, decay, sustain, release 의 네 단계를 통해 설명된다.

33) 음악에서 벨로시티(velocity)는 연주자가 처음 건반을 누를 때 건반을 얼마나 빠르고 강력하게 누르는지를 나타내는 측정값을 말한다.



[그림-7] 감산합성 방식을 사용한 사운드 패치

[그림-7]의 가산합성 방식으로 출력된 모든 신호들이 [그림-15]의 필터를 거치고 난 후 최종적으로 음색이 만들어 지도록 설계하였다. 여러 가지 타입의 필터를 사용하기 위해 biquad~오브젝트와 filtergraph~오브젝트를 사용하였고, cutoff frequency³⁴⁾에 line오브젝트를 연결하여 필터 엔벨로프를 제작하였다. 위의 그림을 보면 오른쪽에 네 개의 메시지 박스가 달려있는데, 예를 들어 맨 위의 메시지 박스처럼 0, 800 1000 0 5000 이라고 하는 명령을 입력하면 cutoff frequency가 0에서 800Hz 까지 1초 동안 움직였다가 다시 0까지 5초 동안 움직이게 된다.

이렇게 가산합성과 감산합성 방식을 각기 다른 조합으로 사용하여 7가지 음색의 악기를 제작하였고, 그 악기들을 코드와 멜로디로 <Cosmos>에 활용하였다.

34) 필터가 효력을 발휘하기 시작하는 지점을 말한다.

2) 사운드 프로세싱 연구

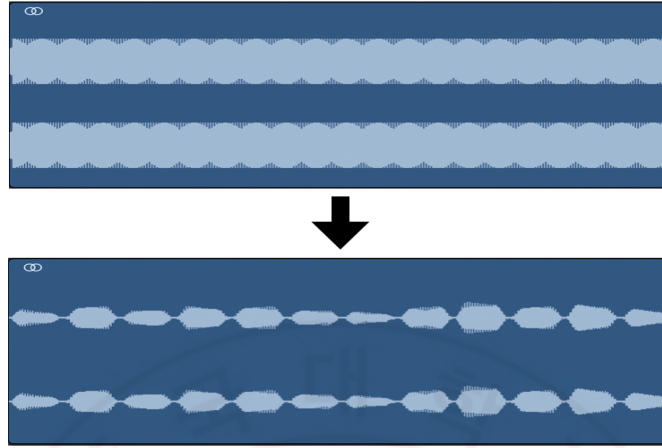
<Cosmos>의 음악은 우두드럼 연주가 주를 이루고, Max를 사용하여 제작한 사운드가 이를 음악적으로 보완해주고 있다. 저음부터 고음까지 비교적 넓은 음역으로 연주가 가능한 우두 드럼이기에 다양한 사운드 프로세싱 효과들을 활용할 수 있었고, 기존의 우두 드럼 사운드에서 새로운 음색으로의 창조가 가능하였다. 타악기의 특성상 짧게 끊어지는 소리들에 긴 release time³⁵⁾을 확보할 수 있도록 granular synthesis를 활용하였고, delay, reverb를 활용하여 압도적인 공간감을 구현하였다. 또한 pitch shift와 ring modulation을 통해 오리지널 우두 드럼의 사운드를 변조했으며, FFT³⁶⁾분석을 통해 우두 드럼의 고음역 부분에만 comb filter 효과를 적용하기도 하였다. 이러한 사운드 프로세싱 효과들은 Max로 제작하였으며, 선행연구에서처럼 각각 독립적으로 작동하게 만든 것이 아니라, 두 가지 이상의 효과들이 서로 유기적으로 작동하도록 설계하였다. 먼저, 연구한 음향효과들을 각각 살펴본 후, 유기적인 연결에 관하여 후술하도록 한다.

① granular synthesis

granular synthesis는 소리를 샘플링한 후 아주 작은 단위로 분해하여 재조합하는 합성법을 말한다. 분해하고 재조합하는 과정에서 소리 입자들을 각기 다른 재생 속도, 위상, 음량, 주파수로 변조할 수 있다.

35) 소리가 지속하는 구간의 음량 값.

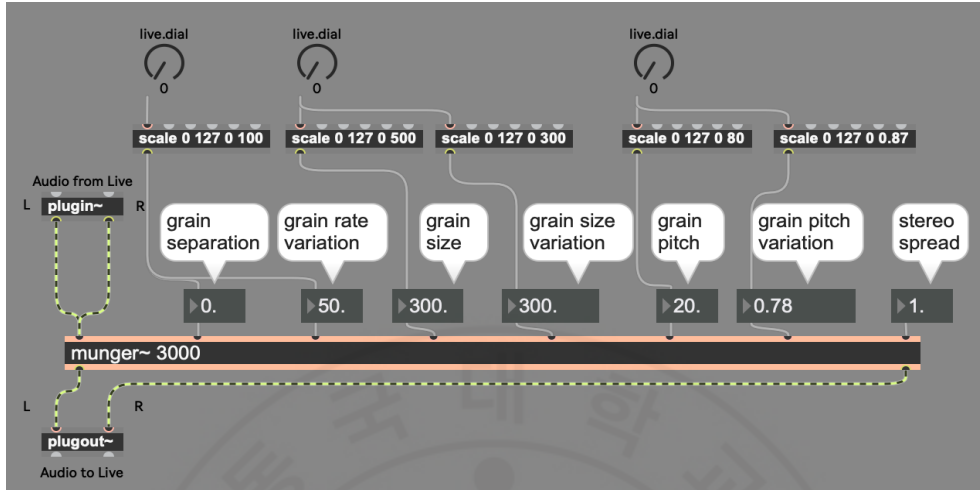
36) Fast Fourier Transform의 약자로, 고속푸리에변환 이라고도 한다. 복잡한 파형이나 신호들을 보다 간단하게 변환시켜주는 알고리즘을 이야기 하며, 분석한 파형 데이터를 이용해 원하는 대로 소리를 재합성 할 때 많이 사용된다.



[그림-8] 오리지널 사인파와 granular synthesis로 합성한 파형

[그림-8]은 220Hz의 사인파와 그 파형을 50ms 단위로 나눈, granular synthesis로 합성한 파형의 모습이다. 위 그림과 같이 granular synthesis 효과를 적용하면 각각의 소리 입자들이 곡물과 흡사하게 보인다 하여 grain이라고 부른다. <Cosmos>에는 `munger~37` 오브젝트를 활용하였다.

37) Max의 외부 오브젝트로 Max의 Package manager에서 다운받을 수 있다. Columbia University에서 만들었으며 본 작품에서 사용된 버전은 Dan Trueman과 R.Luke Dubois에 의해 수정된 버전이다.



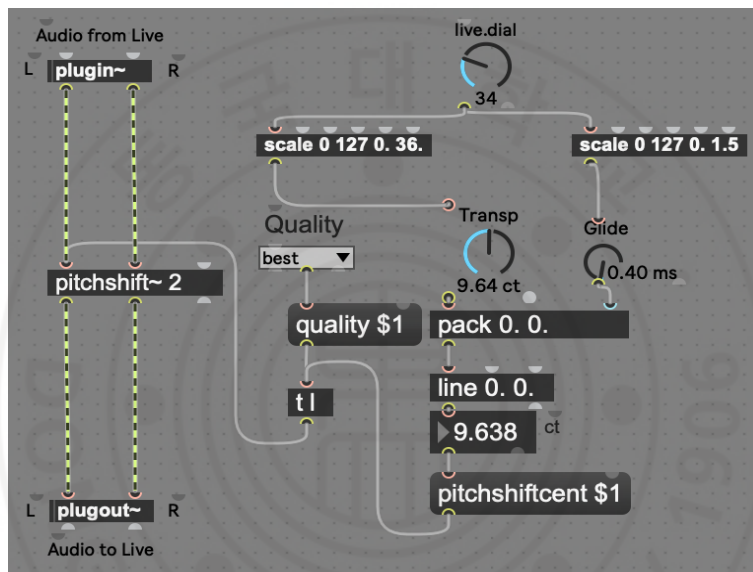
[그림-9] `munger~` 오브젝트를 활용한 granular synthesis 패치

[그림-9]는 Max의 `munger~`를 활용하여 제작한 granular synthesis 패치이다. 총 7개의 파라미터를 조절하여 사운드를 재조합할 수 있으며, 주요 파라미터로는 사운드 소리 입자 사이의 간격(`grain separation`), 소리 입자의 크기(`grain size`), 소리 입자 음정의 높낮이(`grain pitch`) 세 가지가 있다. 또한 변화(`variation`)값으로 주요 파라미터에 대한 변화의 정도를 입력할 수 있어 사운드에 불규칙적인 변조를 가할 수 있다. 그 외에 `stereo spread`는 사운드를 가운데로 모이게 할 것인지, 좌우로 펼칠 것인지를 0과 1의 값을 받아 입력할 수 있다.

위의 [그림-9]에서 주요 파라미터에 실시간으로 컨트롤하기 위해 필요한 `live.dial` 오브젝트와 `scale` 오브젝트가 연결된 것을 볼 수 있다. `live.dial`의 0부터 127의 값을 `scale`을 통해 필요한 만큼의 수치로 치환되게 하였으며, 불규칙한 사운드 변조를 위해 음정과 사이즈의 변화 값도 과감하게 조절할 수 있게 하였다. 다채로운 질감을 만들기 위해 이러한 granular synthesis 패치를 복수로 사용하였다.

② pitch shift

pitch shift는 음색의 왜곡 없이 음의 높낮이를 변화시킬 수 있는 음향효과로, 높낮이가 변화되기 이전의 오리지널 사운드를 살리면서도 제어가 가능하다.



[그림-10] pitchshift~ 오브젝트를 활용한 패치

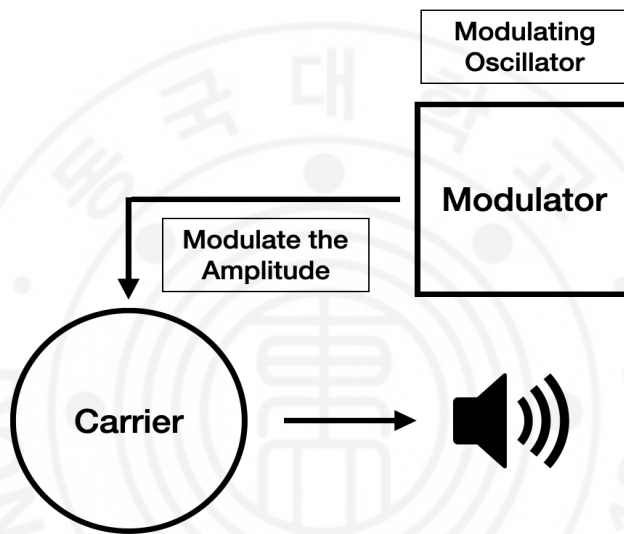
[그림-10]은 pitchshift~오브젝트를 활용한 패치의 모습이다. 음의 높낮이를 cent³⁸⁾ 단위로 조절이 가능하며, ms 단위로 glide³⁹⁾ 효과를 연출할 수 있다. 또한 변화되는 음색의 품질을 선택할 수도 있다. granular synthesis 패치와 마찬가지로 live.dial과 scale을 통해 실시간으로 음의 높낮이와 glide 효과를 조절할 수 있게 하였다.

38) 반음(단 2도) 사이를 100분 한 단위 (1옥타브 = 1200cent)

39) 음이 서서히 올라가거나 낮아지는 효과를 말한다.

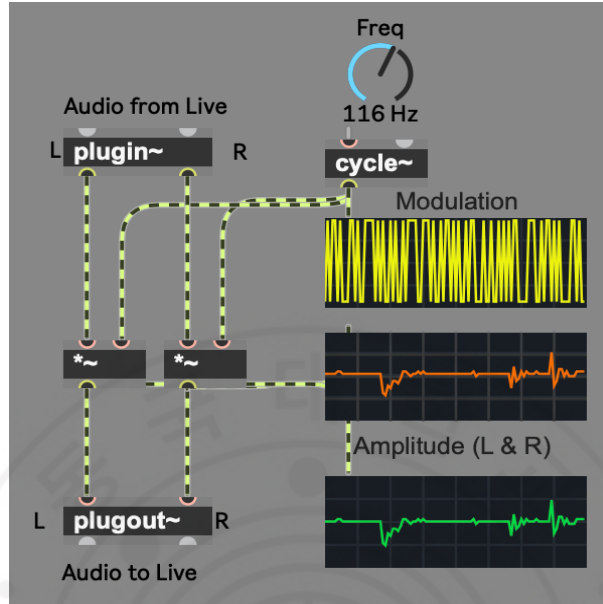
③ ring modulation

전자음악에서 modulation이란 하나의 신호, 혹은 하나의 소리가 가진 특성이 다른 신호의 특성에 의해 변화되는 것을 말하며, 변조라고도 불린다.



[그림-11] ring modulation의 시스템

[그림-11]은 ring modulation의 시스템이다. 오리지널 사운드를 캐리어(carrier), 변조를 하는 신호를 모듈레이터(modulator)라고 한다. 이때 모듈레이터의 주파수가 캐리어의 음량에 변조를 주어 오리지널 사운드의 형태가 변하게 되는 것이 ring modulation의 원리이다. [그림-12]는 Max에서 ring modulation 시스템을 구현한 패치이다.

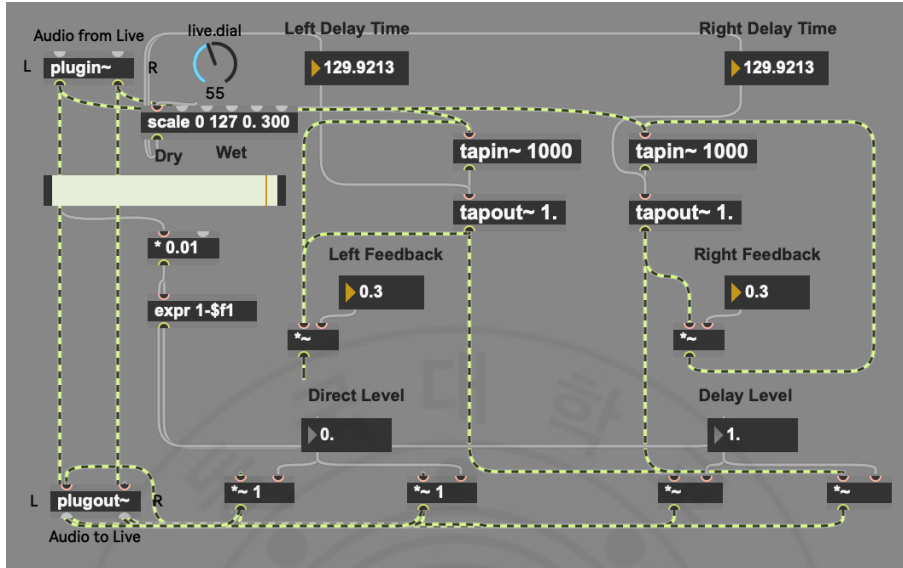


[그림-12] ring modulation을 구현한 패치

cycle~오브젝트에서 발생하는 사인파가 우두 드럼 사운드의 음량 값에 들어가 변조를 일으키게 만들었고, 사인파의 주파수를 실시간으로 변화할 수 있게 제작하였다.

④ delay

delay는 오리지널 사운드와 그 신호를 지연시켜 함께 출력시킬 수 있는 음향효과이다. [그림-13]은 tapin~오브젝트와 tapout~오브젝트를 활용하여 구현한 delay 패치이다.

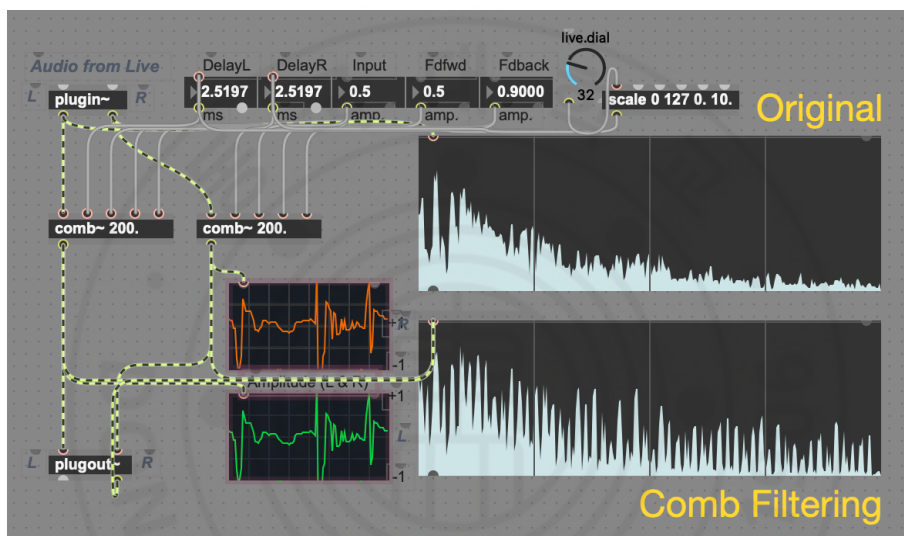


[그림-13] tapin~, tapout~을 활용한 delay 패치

tapin~은 입력된 오디오 신호를 저장하고 용량이 가득 차게 되면, 저장되었던 기존의 신호를 보내면서 새로운 신호를 받는다. tapout~는 tapin~에 저장된 오디오 신호를 입력한 시간만큼 지연시켜 내보낸다. 이 두 오브젝트를 활용하게 되면 입력한 시간만큼의 delay효과를 만들 수 있다. 이후 두 오브젝트를 거친 신호에 일정 비율의 곱으로 소리의 음량을 낮추어 다시 입력하게 되면 feedback 효과를 얻을 수 있다. 또한 입력된 인수를 계산식으로 인식하여 연산해주는 expr오브젝트를 통해 오리지널 사운드와 delay 사운드의 비율을 조절할 수 있으며, dry/wet의 명칭을 가진다. <Cosmos>에서는 delay 효과 자체의 사운드는 사용하지 않고, 우두 드럼 사운드에 delay효과를 건 후, delay효과로 발생하는 사운드에 또 다른 사운드 프로세싱을 적용하는 방식으로 활용하였다.

⑤ comb filter

comb filter는 기본적으로는 delay효과의 원리와 같지만, 직접음에 의한 반사음이 매우 짧은 시간차에 의해 겹쳐져서 만들어지는 효과로 금속성 물질처럼 날카로운 음색을 만들 수 있는 것이 특징이다.



[그림-14] comb~오브젝트를 활용하여 구현한 comb filter 패치

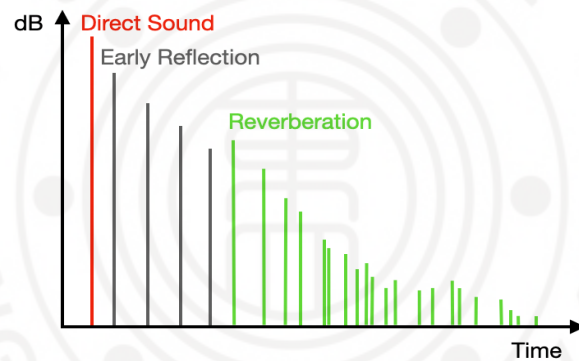
[그림-14]에 나타나있는 comb filter가 적용된 주파수 분석을 보면 마치 빗(comb) 모양과 흡사한 것을 확인할 수 있다. comb~오브젝트를 사용하여 delay time을 live.dial과 scale을 활용해 실시간으로 입력할 수 있게 제작하였다. 공연장에서는 보통 두 개 이상의 마이크를 설치했을 때 발생하는 거리의 차이에 의해 시간차가 생기며 하울링⁴⁰⁾이 발생하지만, 사운드 프로세싱으로 하울링이 일어나지 않는 범위 안에서

40) 스피커에서 나오는 사운드가 마이크로 들어가 앰프에서 재 증폭되어 출력되는 현상.

조절하여 사용이 가능하다.

⑥ reverb

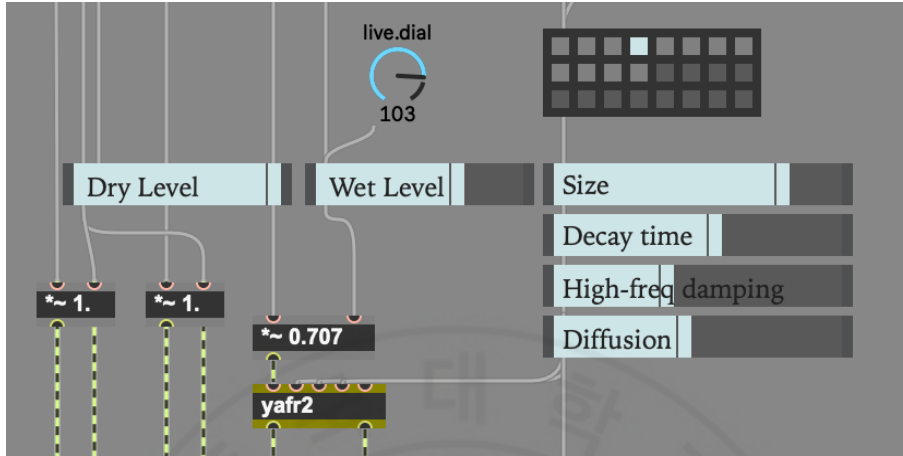
reverb는 초기 반사음(Early Reflection)과 잔향(Reverberation)으로 구성되어 있는 반사음들의 집합체를 말한다. 어떠한 공간에서 사운드가 발생하게 되면, 그 소리가 사방으로 퍼져나가면서 벽이나 장애물에 부딪혀 반사되는 것을 반복하게 되고, 이렇게 반사된 소리들이 시간차를 두고 다시 돌아오는 것이 reverb의 원리이다.



[그림-15] reverb의 구성

<Cosmos>에서는 공연장에서 자연적으로 발생하게 되는 reverb도 활용하였지만, 훨씬 더 압도적인 공간감을 구현하기 위해 Max 패치를 추가로 제작하여 활용하였다. [그림-16]은 yafr2⁴¹⁾ 오브젝트를 활용하여 제작한 reverb 패치이다.

41) Max 의 Package manager에서 받을 수 있는 외부 패치로, Randy Jones에 의해 개발되었다.

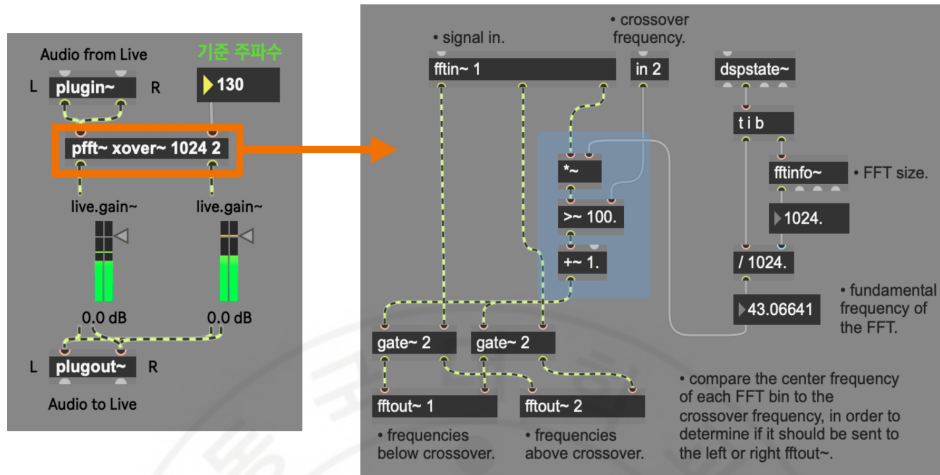


[그림-16] yafr2오브젝트를 활용한 reverb 패치

yafr2를 활용해 조절할 수 있는 파라미터는 총 네 가지로, size(공간의 크기), decay time(소리의 감쇄 시간), high freq damping(고주파 대역 소리의 강도), diffusion(소리의 발산)이 그것이다. reverb 소리의 비율만 live.dial을 활용해 실시간으로 조절할 수 있게 하고, 다양한 공간을 구현하기 위하여 이러한 reverb 패치를 복수로 사용하였다.

⑦ FFT분석을 이용한 주파수 분리(xover filter)

우두 드럼 사운드는 연주 기법에 따라 저음역대에서 만들어지는 소리와 고음역에서 나타나는 음색의 특징이 서로 상이하다. 우두 드럼 사운드에 효과적인 사운드 프로세싱을 입히기 위해서 <표-1>의 연주법-1과 2의 형태로 연주했을 때 얻을 수 있는 저음역의 음색을 나머지 고음역에서 만들어지는 음색과 분리할 필요가 있었다.

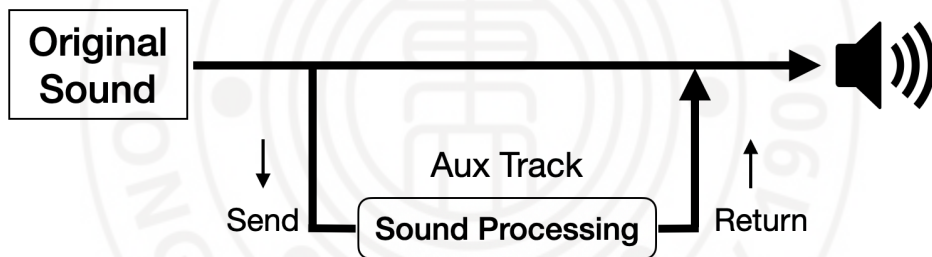


[그림-17] xover filter 패치

[그림-17]은 우두 드럼의 저음역과 고음역의 주파수를 분리하기 위해 pfft~ xover~ 오브젝트를 활용한 xover filter 패치이다. 패치의 왼쪽을 보면 기준이 되는 주파수를 130Hz로 입력하여 분리한 것을 확인할 수 있다. pfft~ xover~ 패치의 내부를 살펴보면, fftin~ 오브젝트를 거쳐 FFT 분석을 하게 된다. 분석을 완료한 주파수가 지정한 주파수보다 낮으면 1이 나와서 fftout~1 오브젝트로 들어가게 되며, 지정한 주파수보다 높으면 2가 나와서 fftout~2로 들어가게 되는 원리이다.

3) 사운드 프로세싱의 유기적 연결을 위한 Max for Live의 활용

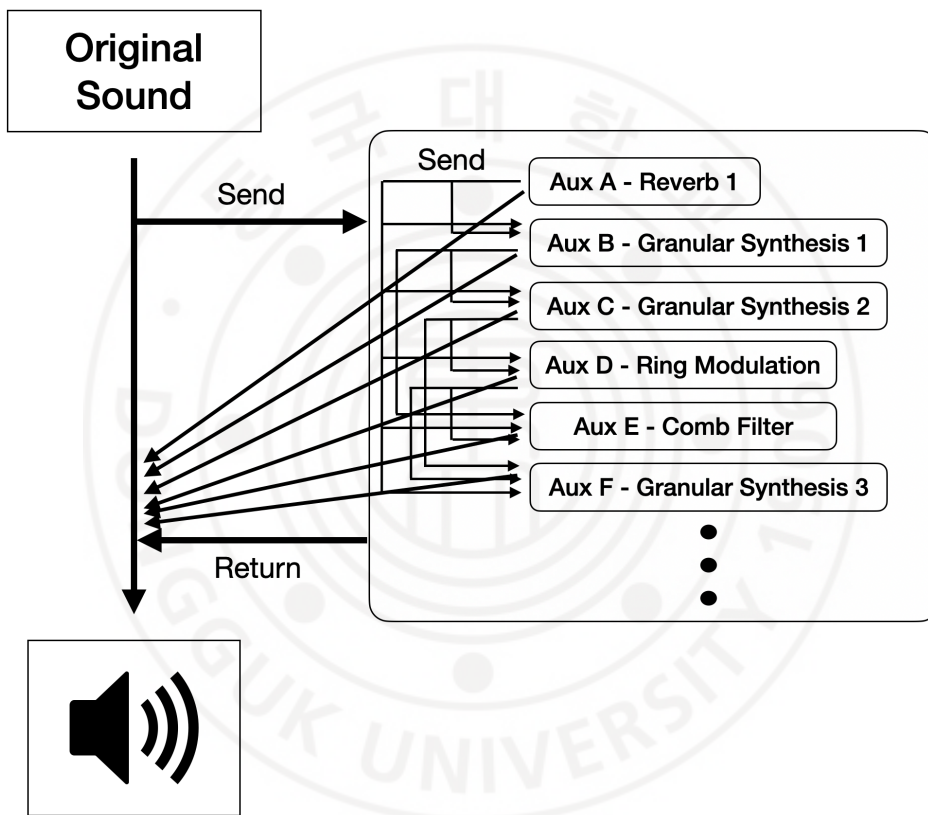
앞서 기술한 사운드 프로세싱을 각각 독립적으로 사용하는 것이 아니라, 서로 유기적으로 연결되어 작동하도록 하기 위해서 Max for Live를 활용하였다. Max for Live는 Ableton Live 내부에서 Max로 제작한 패치들을 사용할 수 있게 해주기 때문에 Ableton Live의 기능을 일부 활용하면서 Max로 제작한 음향효과를 효율적으로 적용할 수 있었다. 1차 사운드 프로세싱에서 2차, 3차의 프로세싱으로 작동하게 하는데 가장 핵심이 되는 기술은 aux 트랙의 활용으로, send/return 방식이라고도 한다.



[그림-18] aux 트랙을 활용한 send/return 방식의 시스템

[그림-18]은 aux 트랙을 활용한 send/return 방식의 시스템을 간단하게 설명한 그림이다. 오리지널 사운드가 최종 아웃풋으로 신호를 전달하는 경로에서 aux 트랙으로 신호를 하나 더 보내고, 그때 aux track에서만 오리지널 사운드에 대한 사운드 프로세싱이 걸리게 된다. 프로세싱 된 신호는 다시 최종 아웃풋으로 돌아가며, 이 때 오리지널 사운드와 프로세싱 된 사운드가 합쳐지게 되는 원리이다. send/return 방식을

사용했을 때의 장점은 오리지널 사운드와 프로세싱 된 사운드의 비율을 dry/wet으로 나누어 사용하는 것이 아니라, 오리지널 사운드를 100% 온전히 쓰면서 프로세싱 된 사운드를 원하는 만큼 섞어서 쓸 수 있다는 것이다.



[그림-19] 작품에 활용한 aux track 설계 시스템의 예

[그림-19]는 작품에 활용한 send/return 방식의 시스템을 도식화한 것이다. 오리지널 사운드가 다수의 aux 트랙으로 보내지고, 사운드

프로세싱 된 각각의 aux 트랙이 또 다른 aux 트랙으로 보내진다. 예를 들어 그림에서 aux A에서 걸린 reverb 1이 aux B의 granular synthesis 1을 포함하여 모든 aux 트랙으로 보내지며, aux B는 다시 aux C와 aux E로 보내지는 것을 알 수 있다. 또한 모든 aux 트랙들은 최종 아웃풋으로 들어가 오리지널 사운드와 합쳐지게 된다. 이처럼 1차 사운드 프로세싱에서 2차, 3차로 이어지는 유기적인 연결을 위해 send/return 방식을 활용하였다. 아래 [그림-20]은 작품에 활용한 Ableton Live 프로젝트의 Session View⁴²⁾이다.



[그림-20] 작품에 활용한 Ableton Live 프로젝트의 Session View

42) 기존의 시퀀싱 프로그램과 달리 작업을 즉흥적으로 시작할 수 있게 해주는 Ableton Live의 view 방식. send/return이 어떻게 작동하고 있는지 한눈에 볼 수 있다.

[그림-20]을 보면 왼편에 우두 드럼 사운드를 위한 트랙이 하나 있고, 우두 트랙에서 각각 다른 사운드 프로세싱 Max 패치들을 포함하고 있는 aux 트랙들로 신호가 보내지는 것을 알 수 있다. aux 트랙들도 서로 유기적으로 신호를 주고받고 있으며, 이때 신호를 어떤 트랙으로 얼마나 보낼 것인지 조절할 수 있다. 이와 같은 방식을 활용하여 Max로 제작한 다양한 패치들을 효율적인 방법으로 결합하여 사용하였다. 우두 드럼을 연주하며 send 신호를 조절할 수 있는 다수의 노브(knob)를 컨트롤 하는 것은 물리적으로 불가능하므로, 손으로 조작 가능한 미디 컨트롤러는 핵심이 되는 몇 개의 노브만 활용하였다. 또한 모든 aux 트랙에 신호를 보내는 aux A트랙에 대한 send 값은 페달 컨트롤러를 사용하여 실시간으로 컨트롤 하였다. 나머지 노브에 대한 컨트롤은 Ableton Live의 오토메이션(automation)⁴³⁾ 기능을 활용하여 실시간으로 사운드 프로세싱이 가능하게 하였다.

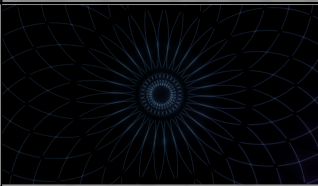
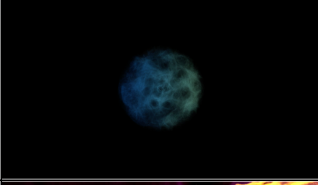



43) 시퀀싱 프로그램에서 사용 가능한 자동화 기능으로, 구간별로 파라미터를 어떻게 움직일 것인지 입력하는 기능을 말한다.

3. 영상 시스템 연구

1) 영상 시스템

사운드와 영상의 인터랙션이 이루어지는 작품을 만들기 위해서는 사운드의 특정 정보에 관한 수치가 전달되고, 그 수치에 영상이 반응하게 해야 한다. 본 연구에서는 마이크를 통해 전달된 우두 드럼의 음량 값과, FFT 분석을 통해 분리된 주파수 값을 받아서 영상에 적용하였다. 영상을 제작하기 위해 사용한 프로그램은 모두 세 가지로, Unity, Processing, After Effects가 그것이다. 다음 <표-2>는 각각의 프로그램에서 제작한 이미지와 시각화 표현을 보여준다.

<표-2> 세 종류의 프로그램을 사용하여 제작한 각각의 이미지와 효과

프로그램 이름	이미지 1	이미지 2	시각화 표현
Unity			우두 드럼 악기의 모습과 연주에서 느껴지는 힘을 형상화
Processing			우두 드럼 연주에서 발생하는 울림을 형상화
After Effects			우주를 표현하는 다양한 이미지들을 형상화

Unity를 사용하여 제작한 영상은 우두 드럼의 악기의 모양과 연주에서 느껴지는 힘을 형상화한 것이다. 이미지 1과 2는 작품의 전개에 따라 각기 다른 파트에 적용된 장면이다. 이미지 1은 작품의 초반 부분과 끝 부분, 이미지 2는 작품의 클라이맥스(climax) 부분의 한 장면이다.

Processing을 사용하여 제작한 영상은 우두 드럼 연주에서 발생하는 울림을 형상화한 것이다. 이미지 1과 2는 각기 다른 코드로 제작하였다.

After Effects로 제작한 영상은 우주를 시각화한 다양한 이미지들을 형상화한 것으로, 이미지 1과 2는 제작한 영상의 일부 장면들이다.

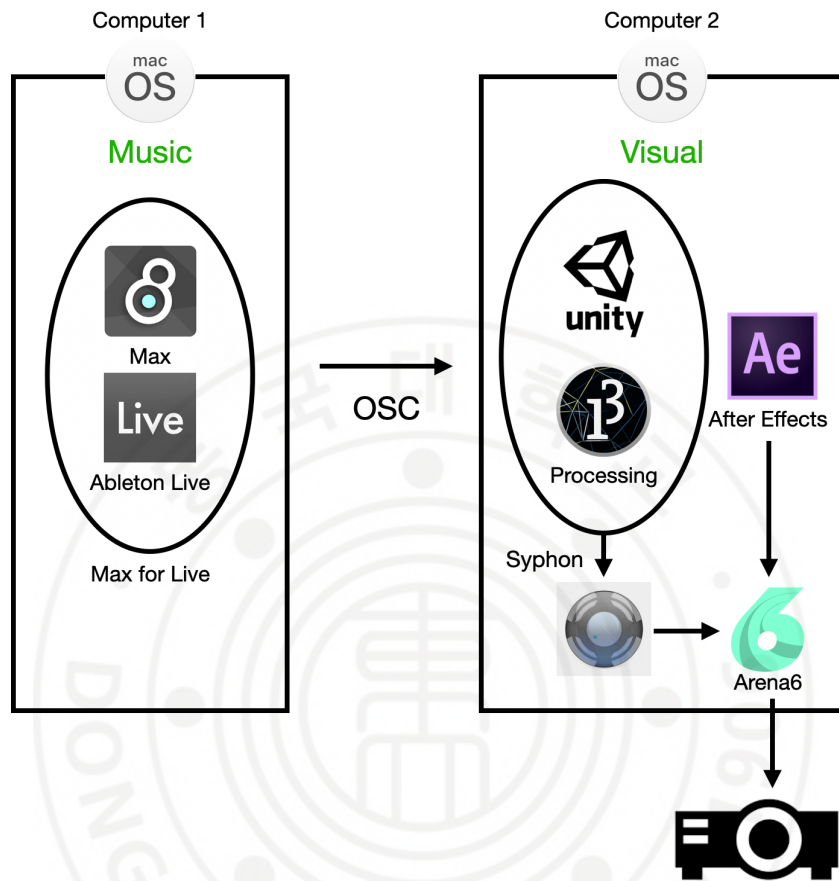
이렇게 사용한 프로그램이 각각 다른 세 가지 타입의 영상을 섞어서 하나의 스크린으로 송출하기 위해서는 또 다른 별도의 프로그램이 필요하여 Resolume의 Arena를 활용했다. Arena는 오디오 신호를 직접 입력할 수 있으며, OSC⁴⁴⁾ 데이터와 MIDI⁴⁵⁾ 데이터의 입력이 가능하여 사운드와의 연동이 용이하다. After Effects는 프로그램 특성상 완벽한 실시간 제어가 불가능하기 때문에 영상 제작 단계에서 음악과의 인터랙션을 미리 적용한 후, 렌더링(rendering)⁴⁶⁾한 영상을 Arena에서 바로 송출하였다. Unity와 Processing은 실시간으로 제어하였지만 Arena와 서로 호환이 되지 않기 때문에 Syphon⁴⁷⁾을 사용하여 호환되게 하였다.

44) Open Sound Control 의 약자로 사운드 데이터 전송을 위해 개발된 네트워크를 이용한 통신 규약이다.

45) Musical Instrument Digital Interface의 약자로 서로 다른 전자악기간의 데이터 전송을 위해 제정한 것이다.

46) 렌더링이란 컴퓨터 프로그램을 이용하여 영상을 합성하는 과정이다.

47) 응용 프로그램간의 영상 프레임을 서로 공유할 수 있게 해주는 Mac OS 기반의 오픈소스이다.



[그림-21] 영상 시스템 설계도

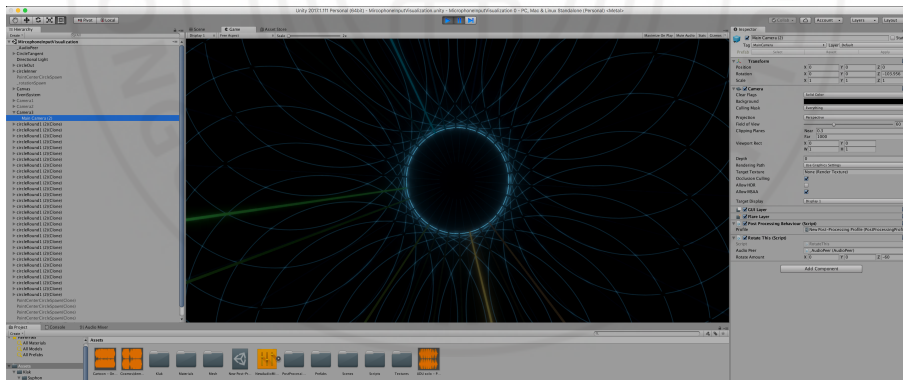
[그림-21]은 작품의 영상 시스템 설계도이다. 연주자가 우두 드럼을 연주하며 사운드 프로세싱과 OSC 데이터 신호를 직접 컨트롤해야 했기 때문에 두 대의 컴퓨터를 활용하였다. 공연 시스템의 안정성 또한 두 대의 컴퓨터를 활용하는 방안을 택했던 이유가 되었으며, 두 컴퓨터는 무선 공유기를 사용하여 데이터 전송을 가능하게 하였다. Max for Live에서 사운드가 출력되면 OSC 데이터를 통해 각각 Unity와

Processing, Arena로 전송된다. After Effects로 렌더링한 영상은 Arena에서 바로 준비되고, Unity와 Processing의 영상은 Syphon을 통해 Arena로 전송된다. 최종적으로 Arena에서 영상 믹싱 및 효과를 주어 프로젝터로 송출한다.

2) 영상 제작 및 효과

① Unity를 이용한 영상제작



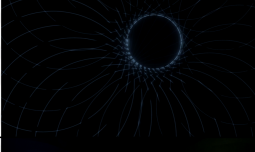
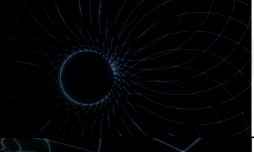
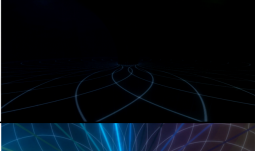

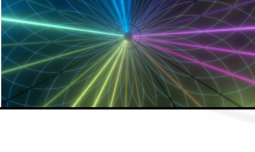
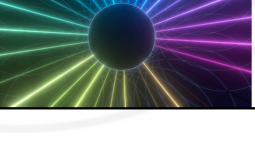
Unity는 프로젝트를 처음 만들기 시작할 때 선택한 버전을 업그레이드 하게 되면, 기존에 작업하던 프로젝트가 작동하지 않는 경우가 많다. 작품을 위해 제작한 프로젝트는 2017.1.1.f1 버전을 사용하였으며 아래 [그림-22]는 Unity 프로젝트 내부의 주요 화면이다.



[그림-22] Unity 프로젝트의 주요 화면

[그림-22]와 같이 Unity를 사용하여 제작한 영상은 우주 드럼 악기 자체의 모습과 그 연주에서 느껴지는 힘을 상징적으로 형상화한 것이다. 우주 전면부의 큰 구멍을 본떠 만든 구체와 그 주위를 둘러싼 수많은 원들로 이루어진 무늬는 토속적인 느낌을 강조하였다. 인터랙션을 위해 우주 드럼의 음량 값과 FFT 분석을 통해 저음역대의 주파수 값을 따로 받아 사용하였다. 아래 <표-3>은 Unity로 제작한 영상에서 주요한 표현 효과들을 구분하여 보여준다.

<표-3> Unity로 제작한 영상의 주요 표현 효과

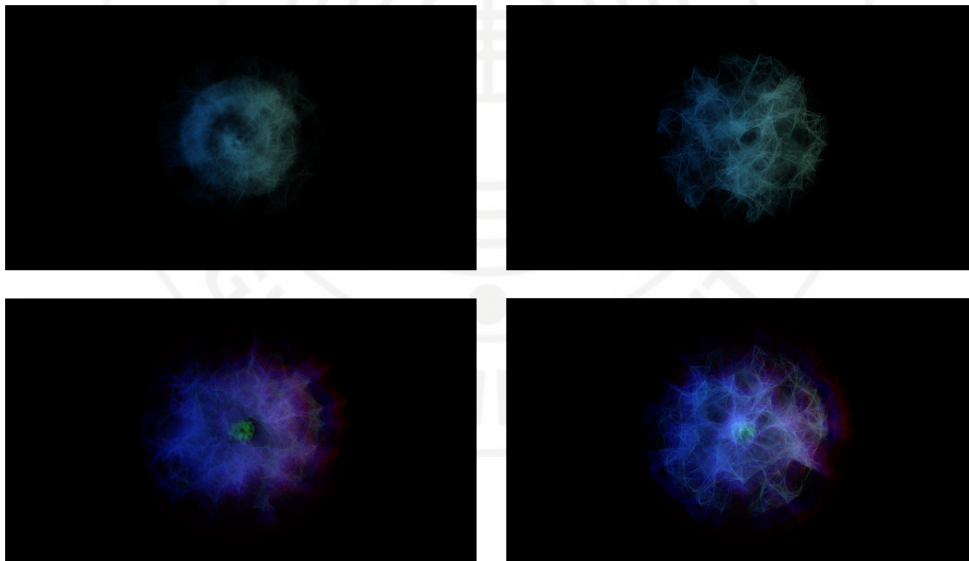
형태 변화	이미지 1	이미지 2	표현 효과
형태 변화-1			원의 크기와 무늬 변화
형태 변화-2			중심이 되는 원의 움직임
형태 변화-3			카메라 시점의 변화
형태 변화-4			원으로 부터의 빛 방출

기본적으로는 우주 드럼 연주로 신호가 발생할 때 마다 그 음량 값을 받아서 형상화한 무늬가 회전하도록 하였으며, FFT 분석을 통해 우주 드럼 전면부의 큰 구멍을 타격했을 때 나오는 저음역대의 주파수 값에 반응하여 중심이 되는 원이 움직이도록 설계하였다. 또한 작품의 전개에 따라 중심이 되는 원의 크기를 변하게 하여 주변의 무늬 또한 변화하게

하였고, Unity 내부에서 카메라 시점을 3개로 나누어 각각 파트별로 적용하였다. 작품의 클라이맥스 부분에는 형상화한 무늬 가운데의 원을 중심으로 강렬할 빛을 뿜어내게 하여 다채로운 색감을 강조하였다.

② Processing을 이용한 영상제작

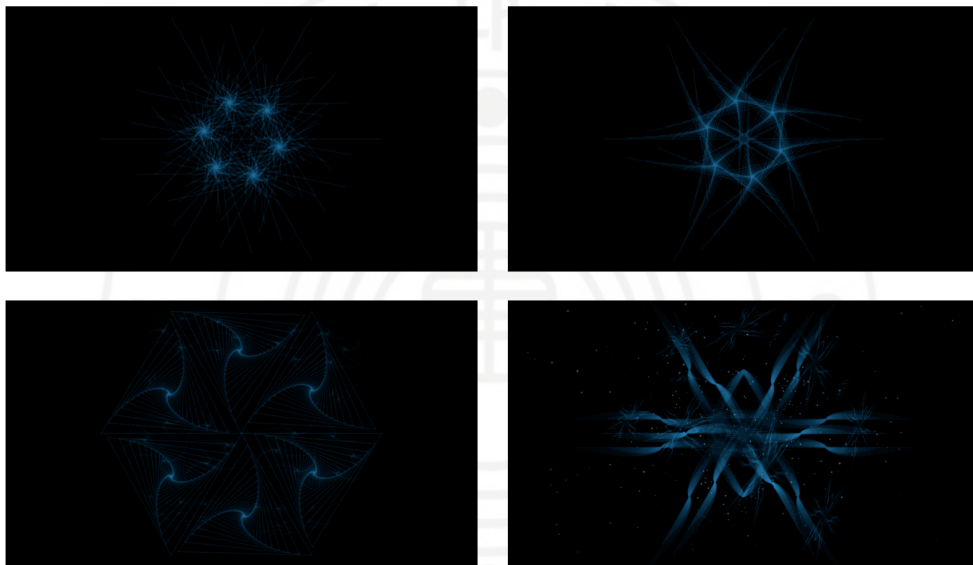
Processing으로 제작한 영상 타입은 두 가지로, 우두 드럼 연주의 울림을 시각적으로 형상화한 것이다. Unity와 동일하게 음량 값과 FFT 분석을 통해 분리된 주파수 값을 사용하였다. 아래 [그림-23]는 Processing으로 제작한 첫 번째 타입 영상의 일부 장면으로, 우두 드럼의 음량 값과 주파수 분리에 따른 크기와 형태, 색의 변화를 보여준다.



[그림-23] Processing으로 제작한 영상 타입 1의 변화

[그림-23]과 같이 첫 번째 타입의 영상은 우주 드럼을 타격할 때 발생하는 오리지널 사운드에 반응하여 크기와 형태, 색의 변화가 이루어지도록 설계 하였다.

아래의 [그림-24]는 Processing으로 제작한 두 번째 타입 영상의 일부 장면으로, 타입 1과 동일하게 음량 값에 반응하여 나타나는 크기와 형태의 변화를 보여준다.



[그림-24] Processing으로 제작한 영상 타입 2의 변화

[그림-24]와 같이 두 번째 타입의 영상은 우주 드럼에 사운드 프로세싱이 적용되어 발생하는 사운드의 음량에 따라 반응하게 하여, 그 크기와 형태가 달라지도록 설계하였다.

③ After Effects를 이용한 영상제작

After Effects는 실시간 OSC 데이터 전송으로 영상을 제어하는 것이 불가능하기에 사전에 렌더링한 영상을 작품에 구간별로 배치하였다. 작품의 주제(cosmos)를 직접적으로 관통하는 우주를 시각화한 영상으로, 아래 [그림-25]는 After Effects로 제작한 주요 장면들의 일부이다.



[그림-25] After Effects로 제작한 영상 이미지

위 [그림-25]와 같이 우주를 표현하는 다양한 타입의 영상들을 제작하였고, 작품의 흐름에 따라 제작한 영상들의 장면이 전환되도록 설계하였다.

④ Arena를 이용한 영상효과

Arena는 다양한 영상 소스들을 하나로 섞어 송출하는 역할을 하기도 하지만, 자체적으로 내장되어 있는 영상효과를 사용할 수도 있다. 오늘날 VJ⁴⁸⁾들은 음악의 시각화를 위해 Arena가 가지고 있는 영상효과들을 다양하게 활용한다. <Cosmos>에서도 앞서 기술한 프로그램들로 제작한 영상들을 하나로 묶는데 Arena를 사용하였고, 다양한 표현 방식을 위해 Arena가 가지고 있는 영상효과들을 활용하였다. 아래 <표-4>는 <Cosmos>에 활용한 Arena의 영상효과들을 정리한 것이다.

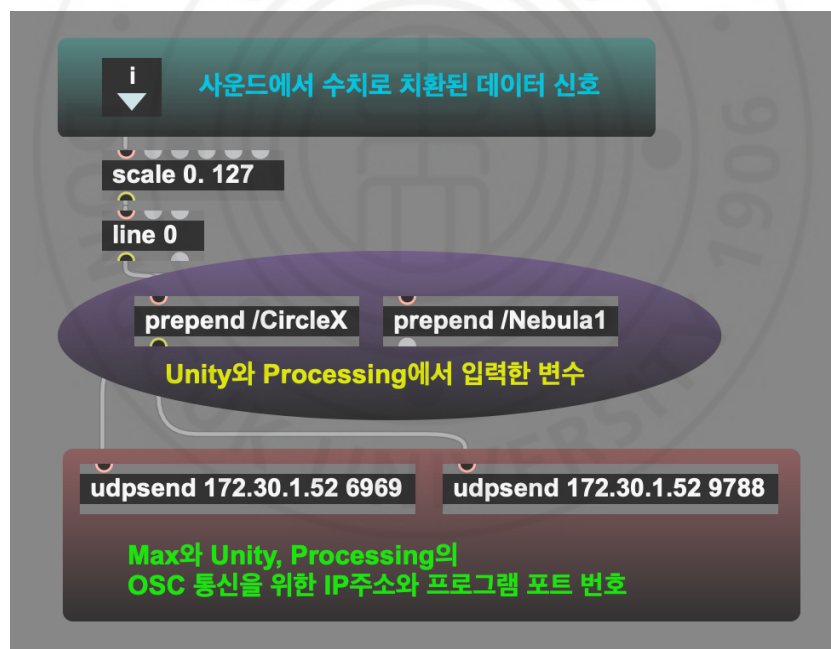
<표-4> Arena의 영상효과

영상효과	효과 설명
Distortion	영상에 균열을 주어 일그러뜨리는 효과
Twitch	영상에 일시적으로 빛 효과를 주며 흔들리게 하는 효과
Goo	영상을 울렁이게 하는 효과
Delay RGB	객체의 윤곽을 RGB로 나눠 시간차에 의해 나타나는 효과
Bright.Contrast	명암 및 밝기 대비 조절 효과
Blur	특정 영역을 흐리거나 번지게 하는 효과

48) 실시간으로 이미지를 생성하거나 조작하는 사람을 말하며, 주로 공연장에서 청중을 위해 청각적 정보를 시각적 정보와 동기화시켜준다.

3) OSC 통신을 이용한 영상의 실시간 제어

사운드 프로세싱을 통한 음악과 실시간으로 연동되는 영상을 제작하기 위해서는, 음악이 가지고 있는 특정한 데이터를 수치로 치환하여 영상을 제어하는 프로그램으로 보내주어야 하며, 그 수치에 따라 영상이 반응하도록 설계해야 한다. 본 작품의 사운드는 Max가 담당하기 때문에 Max에서 수치로 치환된 데이터를 Unity와 Processing으로 보내주어야 한다. 하지만 Max와 Unity, Processing, Arena는 각각 다른 프로그램이므로 서로 연동시켜주기 위하여 OSC 통신을 사용하였다.



[그림-26] Max와 Unity, Processing의 OSC 통신을 위한 패치

[그림-26]은 Max에서 사운드가 수치로 치환된 데이터를 OSC 통신을 이용해 Unity와 Processing으로 보내주는 패치이다. UDP⁴⁹⁾ 방식의 통신 프로토콜을 사용하여 `udpsend` 오브젝트에 IP주소⁵⁰⁾를 입력하고 각각의 프로그램 포트번호를 지정하였다. `prepend` 오브젝트⁵¹⁾에 입력된 변수라는 것은 사용자가 임의로 설정이 가능한 것으로, Unity와 Processing에서 각각 받아들일 수 있는 이름이다. Unity와 Processing 코드에서 같은 변수 이름을 입력하면 인식이 가능하다.

① Unity 영상의 실시간 제어

Unity에서 Max로부터 OSC 통신을 통해 전송된 데이터를 받기 위해서는 C#언어⁵²⁾로 작성한 실행코드가 필요하다. [그림-27]은 OSC 통신으로 전송된 데이터를 받기 위한 코드이다.

49) User Datagram Protocol의 약자로 서로 교환하는 형식이 아닌 일방적으로 이뤄지는 방식의 통신 프로토콜이며, 데이터 처리가 고속으로 가능하다.

50) Internet Protocol address의 약자로 컴퓨터 네트워크에서 장치들이 서로 인식하고 통신을 하기 위해 사용하는 특수한 번호이다.

51) 전송하려는 메시지 앞에 다른 메시지를 추가로 입력해주는 Max 오브젝트.

52) 마이크로소프트에서 개발한 객체 지향 프로그래밍 언어이다.

```

void Start () {
    osc.SetAddressHandler( "/CircleXYZ" , OnReceiveXYZ );
    osc.SetAddressHandler("/CircleX", OnReceiveX);
    osc.SetAddressHandler("/CircleY", OnReceiveY);
    osc.SetAddressHandler("/CircleZ", OnReceiveZ);
}

```

```

void OnReceiveX(OscMessage message) {
    float x = message.GetFloat(0);

    Vector3 position = transform.position;

    position.x = x;

    transform.position = position;
}

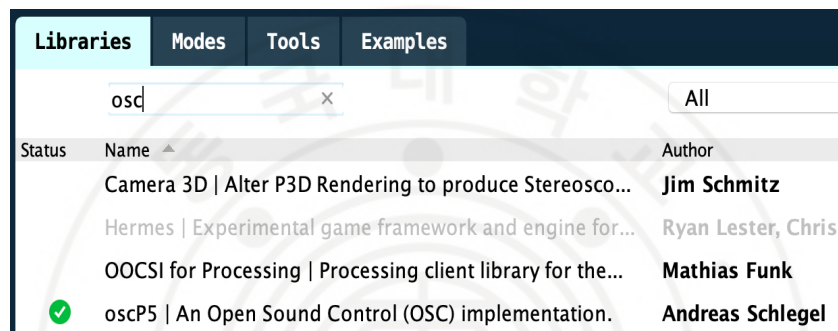
```

[그림-27] OSC 통신을 받기 위한 Unity의 스크립트

[그림-27]의 상단에서 Max에서 prepend로 보내는 메시지를 받기 위해 변수 이름을 일치시킨 것을 알 수 있다. Unity에서 제작하는 공간은 3D이기 때문에 x,y축과 더불어 z축까지 조작성이 가능하다. 하단에서는 Unity로 제작한 공간 안에서 물체의 위치를 나타내는 transform.position 명령어를 사용하여 x축을 움직이도록 설정하였다. 이렇게 스크립트를 작성하면 Unity 프로젝트가 실행된 상태에서 Max로부터 OSC 통신을 통해 전송되는 신호를 받아 영상이 실시간으로 반응하도록 하는 것이 가능하다.

② Processing 영상의 실시간 제어

Processing에서 Max로부터 OSC 통신을 통해 전송된 데이터를 받기 위해서는 oscP5 내장 라이브러리를 먼저 설치해야한다. [그림-28]은 Processing의 라이브러리 설치 화면이다.



[그림-28] Processing의 oscp5 라이브러리 설치 화면

[그림-29]는 설치한 라이브러리를 불러오는 기본 코드이다. 제작한 영상을 실행하기 위해 작성한 코드의 첫째 단에 입력한다.

```
import oscP5.*;
import netP5.*;

OscP5 oscP5;
NetAddress myRemoteLocation;
```

[그림-29] OSC 라이브러리를 불러오는 Processing 코드

[그림-30]은 불러온 OSC 라이브러리를 설정하는 코드이다. Max와의 통신을 위해 IP주소를 입력해주며, 포트번호 또한 Max패치와 일치시켜야 한다.

```
size(1920, 1080, P2D);  
oscP5 = new OscP5(this, 9788);  
myRemoteLocation = new NetAddress("172.30.1.52", 9787);
```

[그림-30] OSC 통신에 필요한 Processing의 주소 설정 코드

[그림-31]은 영상을 실행하기 위해 작성한 코드에서 변수를 설정하는 코드이다. Max 패치에서 prepend로 보내는 메시지에 입력한 변수 이름과 일치시켜야 한다.

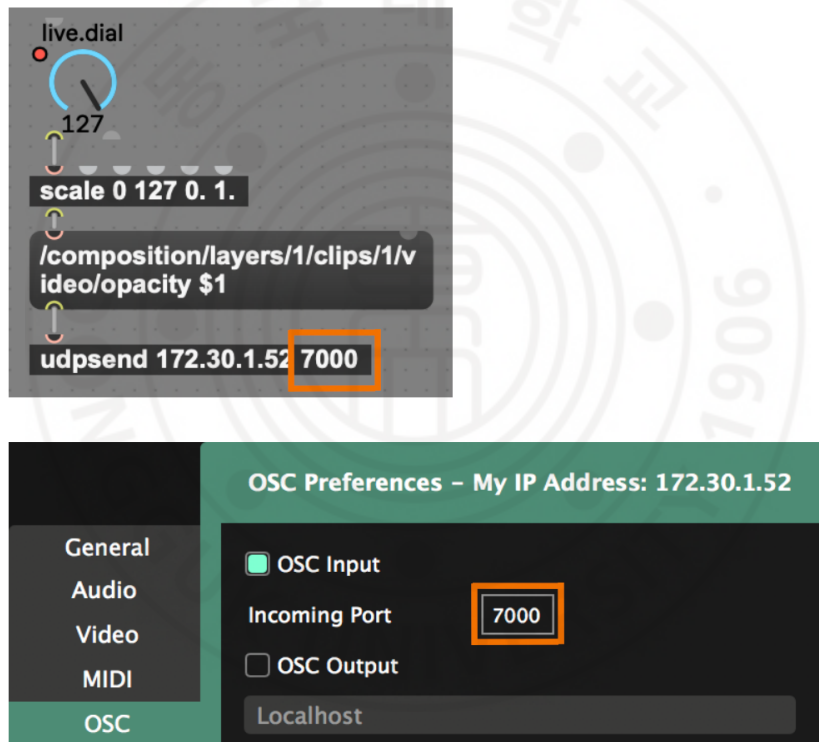
```
void oscEvent(OscMessage theOscMessage) {  
  if (theOscMessage.checkAddrPattern("/Nebula") == true) {  
    println(theOscMessage.get(0).floatValue());  
    A1 = theOscMessage.get(0).floatValue();  
  }  
}
```

[그림-31] OSC 입력 데이터를 변수로 설정하는 코드

Processing에서 제작한 영상을 실행시키기 위한 코드에 위의 코드들을 적용시켜주면 Max로부터 OSC 통신을 통해 전송되는 신호를 받아 영상이 실시간으로 반응하도록 하는 것이 가능하다.

③ Arena 영상효과의 실시간 제어

Arena에서 Max로부터 OSC 통신을 통해 전송된 데이터를 받기 위해서는 udpsend에서 IP주소와 Arena의 포트번호를 일치시켜 주기만 하면 된다. 아래 [그림-32]는 Max에서 수치로 치환한 데이터를 Arena로 보내기 위한 패치와 Arena의 포트번호를 일치시킨 모습이다.



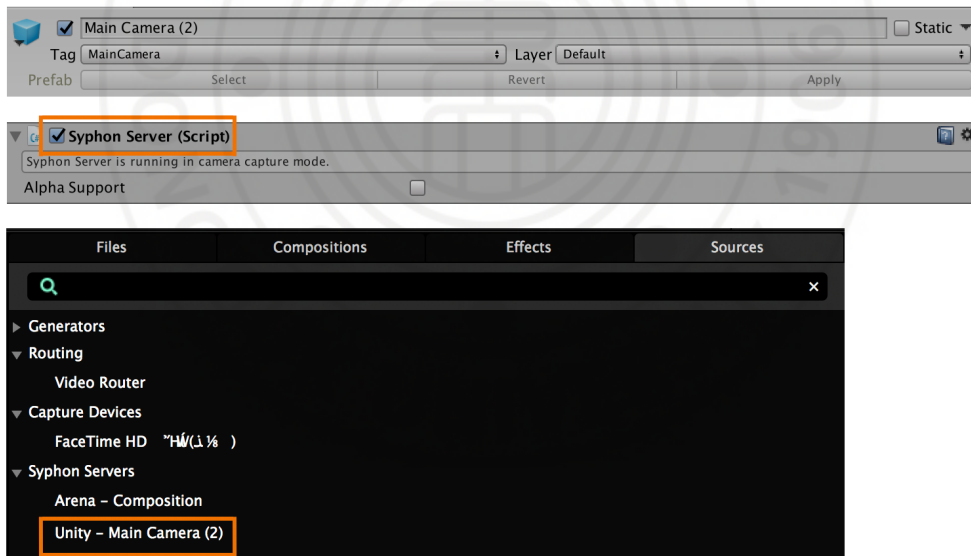
[그림-32] Max와 Arena의 연동

4) Syphon을 이용한 Arena와의 연동

Unity와 Processing으로 제작한 영상을 하나로 묶어 송출하려면 Arena를 활용해야 하고, Unity와 Processing에서 실시간으로 만들어지고 있는 영상 프레임을 Arena로 전송하려면 Syphon을 사용해야 한다.

① Unity와 Arena의 연동

Unity에서 Syphon을 사용하려면 Syphon Asset⁵³⁾을 Unity 프로젝트에 설치해야 한다. 아래 [그림-33]은 Unity 프로젝트 내부에 Syphon Asset이 설치된 모습과 Arena에서 Unity를 인식한 화면을 보여준다.



[그림-33] Syphon을 이용한 Unity와 Arena의 연동

53) Unity에서 Asset이란 Unity에서 작성하는 스크립트, 오브젝트 등을 말한다.

② Processing과 Arena의 연동

Processing에서 Syphon을 사용하려면 아래 [그림-34]와 같이 Syphon 라이브러리를 설치하고, 설치한 라이브러리를 불러오기 위한 코드를 작성해야 한다.



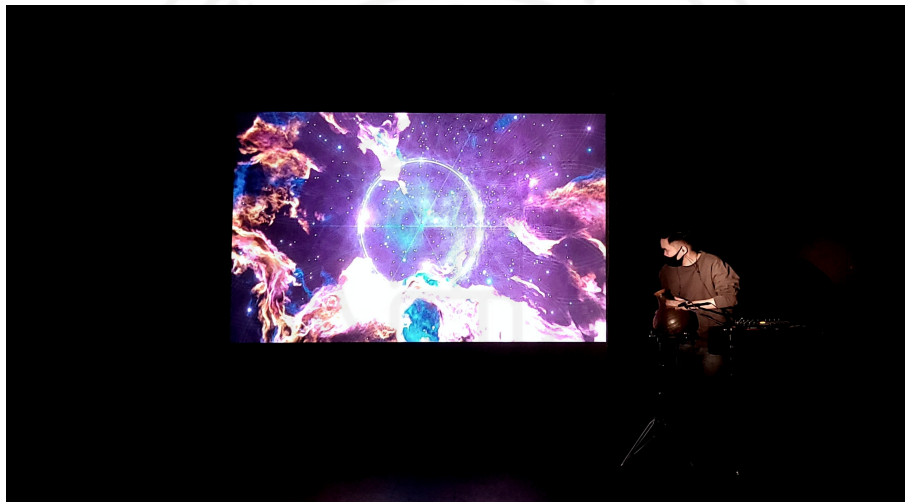
[그림-34] Syphon을 이용한 Processing과 Arena의 연동

위 그림과 같이 Processing에서 Syphon을 이용하여 작성된 코드를 Arena에서 인식한 것을 알 수 있다.

4. 연구 기술의 작품 적용

연구된 우주 드럼의 실시간 사운드 프로세싱을 이용한 멀티미디어 음악 작품 <Cosmos>는 2020년 12월 30일 동국대학교 소극장에서 온라인으로 진행된 ‘보는 소리, 듣는 영상 XVII’ 공연에 초연되었다.

1) 작품 소개



[그림-35] 작품 <Cosmos>의 실제 공연 이미지 1

작품 <Cosmos>는 현대 과학기술과 예술이 결합된 멀티미디어 작품을 다소 난해하고 어렵게 받아들일 수 있는 일반 대중들이 조금 더 쉽게 공감할 수 있도록, 음악 기법과 작품의 형식은 어느 정도 2010년대 이후의 전자음악 스타일을 따랐다. 하지만 악기에 관심이 없는 사람들에게는 생소할 수 있는 우주 드럼이라는 타악기 연주와, 그로인해

파생되는 다양한 음향효과가 적용된 사운드로 작품 전체를 이끌어간다는 점은 충분히 차별화된다. 천지 만물, 우주라는 주제를 바탕으로 머나먼 과거에서 시작된 우주와 태양계의 탄생, 지구와 인류의 역사, 존재와 시간에 대한 고찰 등을 작품에서 풀어내고자 했으며, 고요하면서도 역동적인 우두 드럼 연주의 울림을 그러한 주제들을 표현하는 하나의 수단으로 사용하였다.

2) 작품 구성 및 기술 적용

① 음악 구성 및 기술 적용

본 작품은 크게 A - B - C - C' - B' - A' 의 여섯 개 파트로 나누어진다. A파트는 도입부에 해당하는 구간으로, Max로 제작한 악기를 활용한 사운드로 시작하며 우두 연주가 천천히 섞인다. A파트의 우두 연주는 앞서 본 <표-1>에 설명되어 있는 연주법 4를 위주로 차분히 전개되고, B파트에 다가갈수록 연주는 점점 더 격렬해지며 B파트가 시작된다.

B파트에서는 <표-1>에 설명되어 있는 연주법-1이 처음 등장하여 저음역대를 채워주고, 프로세싱 된 다양한 사운드가 파생된다. Max로 제작한 사운드 또한 그 느낌을 보조해준다. 긴장감을 주기 위해 B파트가 시작되고 얼마 동안은 코드진행(chord progression)⁵⁴⁾을 시키지 않다가 제작한 Pad 사운드가 들어오면서 코드가 전개된다.

C파트가 시작되면 Max로 제작한 사운드가 사라지며 우두 드럼 연주로 카덴차(Cadenza)형식의 독주가 시작된다. 최대한 다양한 사운드 프로세싱을 활용하였으며 B파트까지 일정한 박자를 따라가던 연주가

54) 코드진행은 연속되는 화음의 진행을 말한다. 고전음악 시대부터 오늘날까지 서양 음악의 전통에서 음악적 조화의 기초가 된다.

자유로운 연주로 바뀐다.

C'파트는 우두드럼 독주에 Max로 제작한 사운드가 다시 서서히 쉼이기 시작하며 클라이맥스인 B'파트로 가기 위한 전개가 이루어진다. C파트까지 등장하지 않았던 밝은 느낌의 코드진행을 사용했으며 프로세싱 된 사운드도 지속적으로 귀를 자극하며 기대감을 갖게 한다.

B'파트는 B파트와 비슷한 사운드와 연주법이 활용되지만, C'파트에서 활용했던 코드전개를 사용한다. Max로 제작한 악기를 활용한 사운드를 적극 사용하여 전체적으로 짝 들어차 있는 사운드 질감이 우두 드럼 연주와 어우러진다.

마지막 A' 파트는 작품의 끝을 장식하는 구간으로, 짝 차있던 사운드가 하나 둘씩 사라지며 종착점을 향해가는 느낌으로 전개된다. 최후반부에는 우두 드럼 연주만 남았다가 앞서 본 <표-1>에 설명되어 있는 연주법-4로 마무리된다.

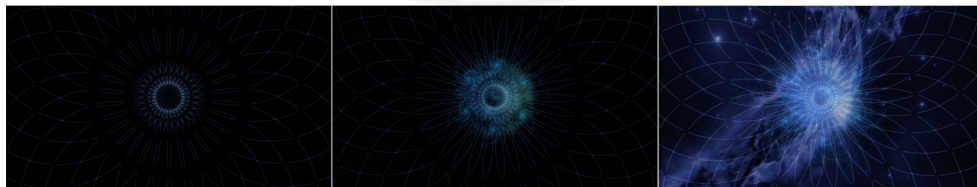
<표-5>는 각각의 파트별로 우두 드럼의 연주법과 Max로 제작한 사운드의 종류, 코드진행, 사운드 프로세싱의 유무를 정리한 것으로, 여기서의 사운드 프로세싱은 독립적으로 작동하는 효과가 아니라 앞서 기술한 send/return 방식으로 연결되어 작동하는 사운드 프로세싱을 말한다.

<표-5> 작품의 음악 구성

파트	우두 드럼 연주법	Max로 제작한 사운드	코드진행	사운드 프로세싱
A	연주법 3, 연주법 4	Bass 1, Pluck 1, Pluck 2	E♭m(add11) - G(add9) - A♭9 sus4 - E♭m(add11)	X
B	연주법 1, 연주법 2, 연주법 3	Bass 2, Bass3, Pad 1, Pluck 1, Pluck 2,	E♭m - E♭m7 - G♭(add9) - A♭9 sus4	O
C	연주법 1, 연주법 2, 연주법 3, 연주법 4, 카덴차	X	X	O
C'	연주법 3, 연주법 4	Bass 1, Pad 2, Pluck 1	G♭(add9) - A♭9 sus4 - B♭m7 - D♭6	O
B'	연주법 1, 연주법 2, 연주법 3, 연주법 4	Bass 2, Bass3, Pad 3, Pluck 1, Pluck 2	G♭maj7/9 - A♭9 sus4 - B♭m7 - E♭m7	O
A'	연주법 1, 연주법 2, 연주법 3, 연주법 4	Bass 1, Pad 2	G♭maj7/9 - B♭m7/11 - E♭m	O

② 영상 구성 및 기술 적용

작품을 위해 사용된 영상 소스는 각기 다른 프로그램으로 제작했기 때문에, 그것들을 하나로 섞었을 때의 조화가 중요하다. [그림-36]은 A파트의 영상 변화를 보여준다.



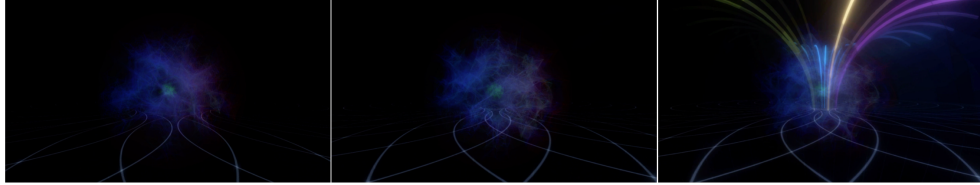
[그림-36] A파트의 영상 변화

Unity로 제작한 우주 드럼의 악기 모습과 연주에서 느껴지는 힘을 형상화한 이미지가 우주 드럼 연주의 음량 값에 반응하여 회전하기 시작하고, Processing으로 제작한 우주 드럼 연주를 형상화한 이미지가 오버랩(overlap)된다. A파트의 마지막 부분에 After Effects로 제작한 우주를 시각화한 이미지가 서서히 나타난다.



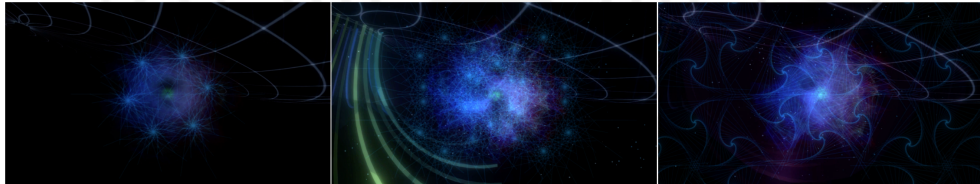
[그림-37] B파트의 영상 변화

[그림-37]는 B파트의 영상 변화를 보여준다. 우주 드럼을 형상화한 구체가 커지면서 우주 드럼의 연주도 그에 맞춰 격렬해진다. 우주 드럼 전면부를 타격하여 발생하는 저음역대의 음량이 FFT 분석을 통해 신호를 전달하여 구체가 크게 움직이고, 유기적으로 연결된 사운드 프로세싱 또한 이때부터 적용된다. 우주를 형상화한 이미지들도 음악에 맞춰 다채롭게 변화한다.



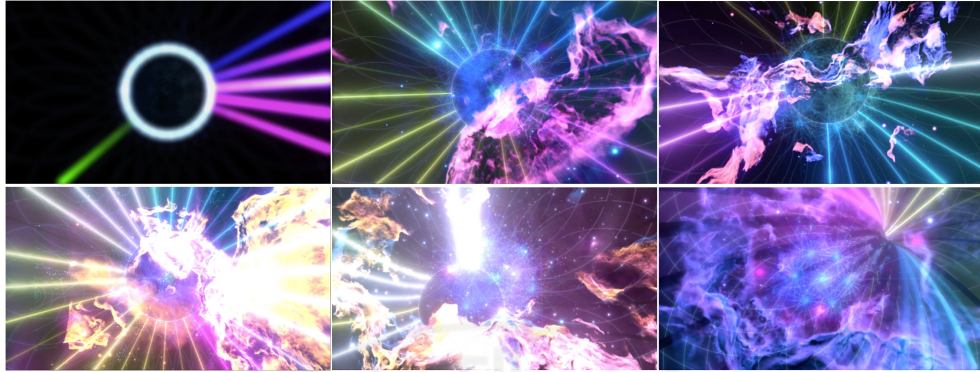
[그림-38] C파트의 영상 변화

[그림-38]은 C파트의 영상 변화를 보여준다. 우주 드럼의 독주와 프로세싱이 적용된 사운드만 남아 카덴차 형식이 시작되면서 Unity로 제작한 영상의 카메라 시점이 바뀐다. 우주를 시각화한 이미지들도 사라져서 우주 드럼 연주의 울림을 시각화한 이미지가 더 강조되고 연주에 맞춰 인터랙션이 이루어진다. 우주 드럼의 독주 부분이므로 영상도 미니멀(minimal)하게 구성하였다.



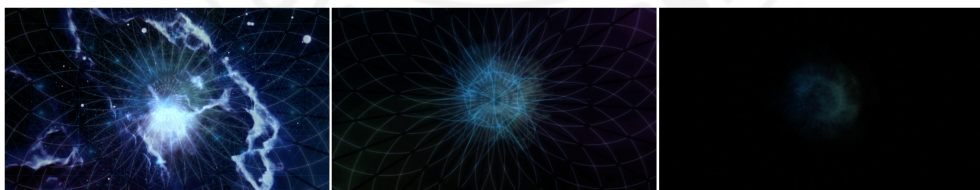
[그림-39] C'파트의 영상 변화

[그림-39]는 C'파트의 영상 변화를 보여준다. Unity의 카메라 시점이 또 다른 시점으로 바뀌면서 Max로 제작한 사운드가 서서히 들어선다. 클라이맥스를 향해 가며 긴장감을 고조시키면서 우주 드럼 연주와 프로세싱이 적용된 사운드를 시각화한 이미지들이 더 화려하게 변화한다.



[그림-40] B'파트의 영상 변화

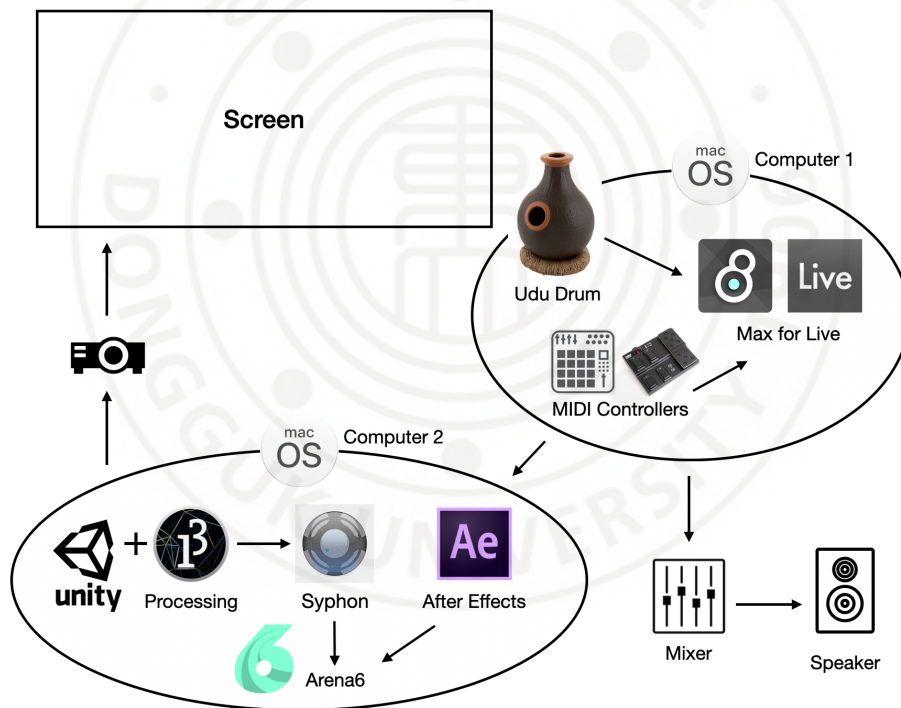
[그림-40]은 작품의 클라이맥스 부분인 B'파트의 영상 변화를 보여준다. Unity의 카메라 시점이 처음 상태로 돌아오고, 우두 드럼 연주와 Max로 제작한 사운드가 조화를 이루며 강렬한 사운드를 만들어낸다. 한 층 더 강렬해진 우두 드럼 연주에 맞춰 화면 중심의 구체에서 화려한 빛을 뿜어내고, 우주를 시각화한 다양한 이미지들도 함께 전환된다. 작품의 클라이맥스 부분인 만큼 화려한 색감을 표현할 수 있게 디자인하였다.



[그림-41] A'파트의 영상 변화

[그림-41]은 작품의 마지막 부분인 A'파트의 영상 변화를 보여준다. 우주를 시각화한 이미지들이 서서히 사라지면서 C파트와 유사하게 우두 드럼 연주와 프로세싱이 적용된 사운드가 중심이 된다. 그에 맞춰 인터랙션이 이루어지는 영상 효과들이 우두 드럼 연주가 연주법 4의 형태에 가까워질 수 록 점점 희미해지고, 마지막에는 우두드럼의 울림을 형상화한 이미지만 남았다가 완전히 사라진다.

③ 무대 및 시스템 구성



[그림-42] 무대 및 시스템 구성

[그림-42]는 무대 및 시스템 구성을 보여준다. 스크린을 정면에 두고 오른쪽에 우두 드럼과 사운드를 담당하는 컴퓨터를 배치하였다. 우두 드럼 연주를 컴퓨터로 받기 위해 DPA사의 DPA-4021 마이크를 사용하였다. 연주자는 우두 드럼을 연주하며 미디 컨트롤러를 실시간으로 조작한다. 음향 신호는 객석 앞에 배치된 믹서로 들어가 스피커로 출력된다. Max로부터 OSC 통신으로 보내는 신호가 영상을 담당하는 컴퓨터로 전송되어 Unity와 Processing에 각각 입력된다. Unity와 Processing에서 실시간으로 만들어진 영상 프레임은 Syphon을 통해 Arena로 전송되고, After Effects로 제작한 영상도 하나로 섞이게 된다. Arena에서 최종적으로 출력되는 영상은 프로젝터를 통해 스크린으로 송출된다.

3) 작품에서의 기술 적용에 따른 예술적 효과

본 작품은 우두 드럼 연주의 실시간 사운드 프로세싱을 활용하고, Max로 제작한 사운드를 함께 사용하였다. 우두 드럼 연주에서 발생하는 음량 값과 주파수 값이 영상을 실시간으로 제어한다. 연구한 사운드 프로세싱 종류는 총 7가지이며, 각기 다른 수치로 작동하는 4개의 granular synthesis, 3개의 reverb, 2개의 xover filter를 사용하였고, 그 외 나머지 효과들은 단수로 사용하였다. 각각의 음향효과들은 앞서 기술한 send/return방식을 활용하여 서로 유기적으로 연결된다.

<표-6>, <표-7>, <표-8>은 작품의 파트별로 사운드 프로세싱 조합, Unity의 영상 효과, Processing의 영상 효과, Arena의 영상 효과를 정리한 것이다. After Effects로 사전에 렌더링한 영상은 장면의 전환 외에 특별한 효과가 없어 생략하였다.

〈표-6〉 A파트, B파트의 기술 적용 및 효과

파트	사운드 프로세싱 조합	Unity의 영상 효과	Processing의 영상 효과	Arena의 영상 효과
A	reverb 1 (독립적으로 작동)	음량 값에 반응하여 반 시계방향으로 회전 구체의 크기 변화 구체를 둘러싼 무늬의 모양 변화	우두 드럼 연주의 음량 값에 의한 크기와 형태 변화	Bright.Contrast
B	xover filter 1 (130Hz) reverb 1 / reverb 2 granular synthesis 1 granular synthesis 2 pitch shift (send/return 방식 - 유기적 결합)	음량 값에 반응하여 반 시계방향으로 회전 구체의 크기 변화 저음역 주파수에 의한 구체의 위치 변화 구체를 둘러싼 무늬의 모양 변화	우두 드럼 연주의 음량 값에 의한 크기와 형태 변화 음향효과가 적용된 사운드의 음량 값에 의한 크기와 형태 변화	Distortion Twitch Goo Delay RGB

〈표-7〉 C파트, C'파트의 기술 적용 및 효과

파트	사운드 프로세싱 조합	Unity의 영상제어 및 효과	Processing의 영상 효과	Arena의 영상 효과
C	xover filter 1 (130Hz) xover filter 2 (1600Hz) reverb 1 / reverb 3 delay / comb filter granular synthesis 3 granular synthesis 4 pitch shift / ring modulation (send/return방식 - 유기적 결합)	카메라의 시점 변화 음량 값에 반응하여 반 시계방향으로 회전 구체의 크기 변화 구체를 둘러싼 무늬의 모양 변화 구체에서 빛 방출	우두 드럼 연주의 음량 값에 의한 크기와 형태 변화 음향효과가 적용된 사운드의 음량 값에 의한 크기와 형태 변화	Delay RGB Blur
C'	xover filter 2 (1600Hz) reverb 1 / reverb 2 delay / comb filter granular synthesis 2 granular synthesis 4 pitch shift / ring modulation (send/return 방식 - 유기적 결합)	카메라의 시점 변화 음량 값에 반응하여 반 시계방향으로 회전 구체의 크기 변화 구체를 둘러싼 무늬의 모양 변화 구체에서 빛 방출	우두 드럼 연주의 음량 값에 의한 크기와 형태 변화 음향효과가 적용된 사운드의 음량 값에 의한 크기와 형태 변화 음향효과가 적용된 사운드의 주파수 값에 의한 형태 변화	Twitch Goo Bright.Contrast

〈표-8〉 B'파트, A'파트의 기술 적용 및 효과

파트	사운드 프로세싱 조합	Unity의 영상제어 및 효과	Processing의 영상 효과	Arena의 영상 효과
B'	xover filter 1 (130Hz) xover filter 2 (1600Hz) reverb 1 / reverb 2 / reverb 3 delay / comb filter granular synthesis 1 granular synthesis 2 granular synthesis 3 pitch shift / ring modulation (send/return방식 - 유기적 결합)	카메라의 시점 복귀 음량 값에 반응하여 반 시계방향으로 회전 구체의 크기 변화 구체를 둘러싼 무늬의 모양 변화 구체에서 빛 방출	우두 드럼 연주의 음량 값에 의한 크기와 형태 변화 음향효과가 적용된 사운드의 음량 값에 의한 크기와 형태 변화 음향효과가 적용된 사운드의 주파수 값에 의한 형태 변화	Distortion Twitch Goo
A'	xover filter 1 (130Hz) xover filter 2 (1600Hz) reverb 1 / reverb 2 delay / comb filter granular synthesis 3 granular synthesis 4 pitch shift / ring modulation (send/return 방식 - 유기적 결합)	음량 값에 반응하여 반 시계방향으로 회전	우두 드럼 연주의 음량 값에 의한 크기와 형태 변화	Bright, Contrast Blur

적용된 사운드 프로세싱 기술들은 모두 중요한 역할을 담당하였지만 특히 xover filter는 각각의 효과들이 유기적으로 결합하는 과정에서 항상 1차적으로 작동시켰다. 예를 들면 기준 주파수를 130Hz로 두어 그 이상으로 걸러진 사운드만 reverb로 보낸 후, 그것을 다시 granular synthesis로 보냈다. 우두 드럼은 저음역대의 음량 값을 매우 강하게 내보낼 수 있기에 사운드의 변조가 강한 granular synthesis나 comb filter에 바로 적용되면 자칫 지저분한 소리가 날 수 있기 때문이다. 기준 주파수를 1600Hz에 두고 그 이상으로 걸러진 사운드를 사용한 경우는 주로 comb filter를 사용하였다. comb filter의 특징인 금속성의 음색을 적용하기 위해 일부러 조금 고음역의 사운드를 사용하였고, 그 상태에서 다시 granular synthesis나 ring modulation을 적용하여 3차, 4차 프로세싱을 가하였다.

영상의 실시간 제어 방식은 크게 세 가지로 나뉘어진다. 첫째는 우두 드럼 연주의 음량 값과 음향효과가 적용된 사운드에서 발생하는 음량 값을 받아서 반응하는 방식이고, 둘째는 xover filter로 걸러진 주파수 값에 의해 반응하는 방식, 마지막 세 번째는 음악에서 발생하는 정보와는 별개로 연주자가 직접 컨트롤러를 조작하거나 Max for Live를 활용해 자동화를 시킨 방식이다. Unity의 카메라 시점 변화와 구체에서의 빛 방출, Arena의 영상 효과들은 세 번째 방식을 활용하였고 나머지 효과들은 모두 첫 번째나 두 번째 방식을 사용하였다.



[그림-43] 작품 <Cosmos>의 실제 공연 이미지 2

Ⅲ. 결 론

본 연구는 서로 다른 두 가지 종류의 예술을 융합하기 위한 목적으로 제작한 멀티미디어 작품에 관한 연구이다. 아시아 문화권에서는 조금 생소한 타악기인 우두 드럼 연주로 작품 전체를 이끌어가면서 음악의 음량 값과 주파수 값에 실시간으로 반응하는 영상을 제작하여 청각적 요소와 시각적 요소 두 가지로 인터랙션이 이루어지게 하였다.

청각적 요소인 음악에는 Max를 사용한 사운드 프로세싱을 적용하여 타악기 연주만으로는 만들어 낼 수 없는 다채로운 질감의 소리를 구현하고, 자칫 지루해질 수 있는 타악기 연주를 음악적으로 보조해줄 코드와 멜로디 또한 Max를 활용하여 제작한 악기로 뒷받침 해주었다. 사운드 프로세싱을 효율적으로 사용하기 위해 Max for Live의 send/return방식을 활용하여, 각각의 사운드 패치들이 여러 단계를 거쳐 적용됨으로써 음향효과의 유기적인 결합이 이루어지게 하였다.

시각적 요소인 영상은 음악에 실시간으로 반응하는 소리의 시각화를 위해 Unity, Processing, After Effects 세 가지 프로그램으로 제작한 영상 소스를 모두 섞어서 활용하여 표현을 극대화 하였다.

또한 악기 연주자와 컨트롤러를 조작하는 오퍼레이터를 따로 구분하지 않고 작가 한 사람이 음악과 영상을 한꺼번에 제어 가능하게 한 점은 작품의 통일성과 완성도를 높여주었으며, 관객으로 하여금 시각적 요소와 청각적 요소가 어떻게 상호작용 하는 것인지를 좀 더 명확하게 전달할 수 있었다.

과학 기술의 발전으로 인해 예술 작품은 그 형식이나 기법이 날이 갈수록 진보하고 있으며 두 가지 이상의 복합 감각을 활용한 멀티미디어 콘텐츠 또한 끊임없이 쏟아지고 있다. 음악, 미술, 문학, 영화 등 각 예술 분야 안에서의 경계는 물론이고 각 예술 분야를 구분 짓는 경계

또한 모호해지고 있다. 이러한 시대에 살고 있는 예술가라면 한 가지 예술 분야 안에서 장르에 대한 구분을 짓는 것은 큰 의미가 없을 것이다. 새롭게 등장하는 디지털 매체에 대한 해석을 어떻게 할 것이며, 또한 그것을 어떻게 활용할 것인지는 예술가 스스로의 몫이다. 다만 과학 기술의 발전에 따라 예술도 진보하는 것이 자연스러운 흐름이라면, 예술가으로써 조금은 더 열린 마음을 가지고 창의적인 접근을 시도하는 것이 인류의 역사에 있어 예술의 지평을 넓히는데 기여할 수 있는 방법이 될 것이다.

Keyword(검색어):

컴퓨터음악(computer music), 멀티미디어음악(multimedia music),
소리합성(sound synthesis), 소리시각화 (sound visualization), Udu drum,
Max/MSP, 실시간 사운드 프로세싱(real-time sound processing)

E-mail: neuro365@dgu.ac.kr

참 고 문 헌

1. 단행본

- Bob Katz, 「Mastering Audio the art and the science」
(third edition, Focal Press, 2007)
- Charles Dodge, Thomas A. Jerse, 「Computer Music: synthesis,
composition and performance」
(Schirmer Books, Second Edition, 1997)
- Curtis Roads, 「The Computer Music Tutorial」 (MIT press, 1996)
- Daniel Shiffman, 「Learning processing」 (Morgan Kaufmann, 2015)
- Glenn D. White, Gary J. Louie, 「The Audio Dictionary」
(University of Washington press, Third Edition, 2005)
- Leon Harkleroad, 「The Math Behind the Music」
(Cambridge University press, 2006)
- V.J.Manzo, 「Max/MSP/Jitter for music」
(Oxford University Press, 2011)

2. 참고논문

- Hernandez, Carlos Roberto Barrios. 「Thinking Parametric Design: Introducing Parametric Gaudi」 (Design Studies 27, no. 3: 309-324. 2006)
- Mania Aghaei Meibodi, 「Generative Design Exploration: Computation and Material Practice」 (Doctoral Thesis 2016)
- 라지웅, 「Max/MSP와 Generative Art를 이용한 멀티미디어음악 제작 연구」 (동국대학교 영상대학원 멀티미디어학과, 2018)
- 이도경, 「피아노 연주를 통한 실시간 오디오 비주얼 작품 제작」 (동국대학교 영상대학원 멀티미디어학과, 2018)
- 조환희, 「베이스 트롬본과 피아노의 실시간 사운드 프로세싱을 이용한 멀티미디어작품 제작 연구」 (동국대학교 영상대학원 멀티미디어학과, 2019)

3. 웹사이트

<http://docs.cycling74.com/max8>

<http://maxforlive.com>

<http://github.com>

<http://syphon.v002.info/>

ABSTRACT

A study on the Production of Multimedia Music
using Real-time Sound Processing
for Udu drum
(focus on Multimedia Music <Cosmos>)

Kim, Jin Woo

Department of Multimedia
Graduate School of Digital Image and Contents
Dongguk University

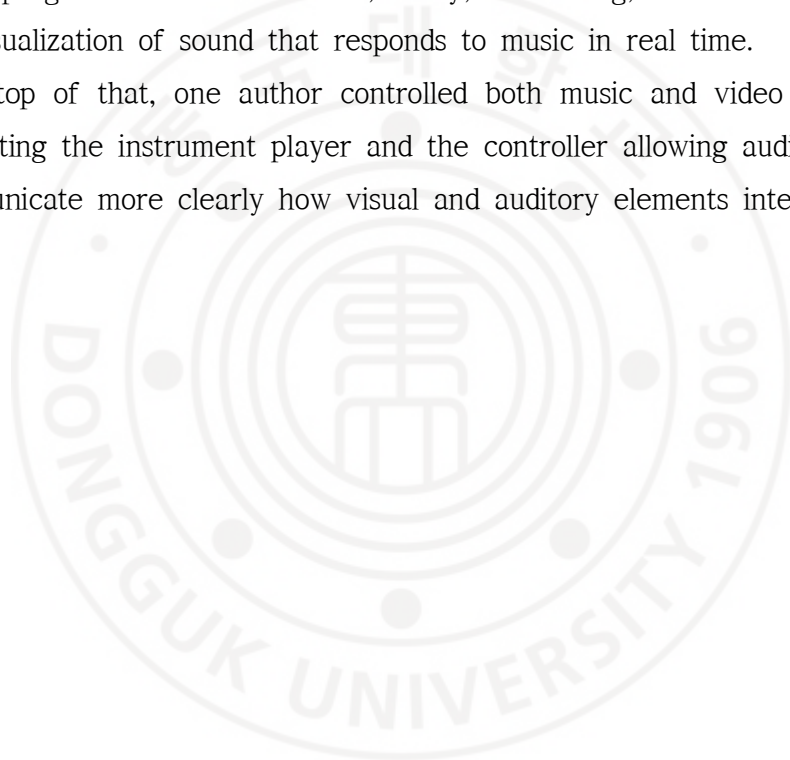
This study is a multimedia work designed to combine two different kinds of art. The entire work was led by the African percussion instrument udu drum, which is slightly unfamiliar to Asian culture, and images that responded to the volume and frequency values of the music were produced in real time, allowing interaction between auditory and visual elements.

The music, which is an auditory element, is applied with sound processing using Max to implement a variety of textual sounds that

cannot be produced by percussion alone. The chords and melodies that would musically support percussion were also produced using Max. In addition, Max for Live's 'send/return' was used to enable organic coupling of sound effects by applying each sound patch in several stages.

The video, which is an visual element, is maximized by mixing all three programmed video sources, Unity, Processing, and After Effect, for visualization of sound that responds to music in real time.

On top of that, one author controlled both music and video without separating the instrument player and the controller allowing audience to communicate more clearly how visual and auditory elements interact.



부록 : 첨부 DVD 설명

1. Cosmos.mov : 2020년 12월 말 동국대학교 소극장 <Cosmos> 온라인 공연 실황
2. Max for Live : 작품에 사용된 Ableton Live 프로젝트 폴더

