석 사 학 위 논 문

Max/MSP와 Generative Art를 이용한 멀티미디어음악 제작 연구

(멀티미디어음악 작품 <Trans-Human>을 중심으로)

지도교수 김 준

동국대학교 영상대학원

멀티미디어학과 컴퓨터음악전공

라 지 웅

2 0 1 8

석 사 학 위 논 문

Max/MSP와 Generative Art를 이용한 멀티미디어음악 제작 연구

(멀티미디어음악 작품 <Trans-Human>을 중심으로)

라 지 웅

지도교수 김 준

이 논문을 석사학위논문으로 제출함

2017년 12 월

라지웅의 음악석사(컴퓨터음악)학위 논문을 인준함

2018년 1 월

위원장 정 진 헌

위 원 박 상 훈

위 원 김 준

동국대학교 영상대학원

목 차

표 목 차

그 림 목 차

Ⅰ. 연구 배경 및 목적

음악의 정의를 예술의 범주 안에서 확장하려는 시도는 과거부터 현재까지 계속되고 있다. 20세기 현대 음악 작곡가인 Luigi Russolo[1]는 소음을 사용해 작곡을 하였고 John Cage는 4:33초[2]라는 작품에서 아무것도 연주 하지 않는 무음(silence)의 실험적인 작품을 발표하였다. 전통을 거부하고 기계와 자동차 그리고 공장에서 발생하는 도시의 모든 소음에 주목했던 미래주의[3]와 그 이후 초현실주의[4]나 플럭서스(Fluxus)운동[5]은 노이즈를 음악적으로 해석하고 가치를 부여 하는데 많은 영향을 주었다.

음향학에서 악음(tone)[6]과 소음(noise)[7]의 구분은 진동의 주기성에 의해서 나눠진다. 본 연구는 음향학에서 정의된 소음을 사용해 하나의 완성된 음악을 제작하는 것에 첫 번째 목적을 두었고 특히 글리치 사운드(glitch sound)[8]를 실제로 제작하고 Max/MSP를 사용해 파형을 합성하여 새로운 사운드를 디자인 하는 작업에 중점을 두었다.

두 번째로는 제너레이티브 예술(generative art)을 사용한 영상을 제작하고 음량변화에 따른 실시간 영상제어를 통해 음악을 시각화 하는 연구를 진행 하였다. 제너레이티브 예술이란 알고리듬 아트(algorithmic art)라고도

---

1) 20세기 현대음악 작곡가로 1913년 <소음예술 선언>을 통해 음향과 소음을 음악에 사용 할 것을 제안했다.
2) 아방가르드 작곡가 John Cage가 1952년에 작곡한 피아노를 위한 작품
3) 이탈리아에서 등장한 모더니즘 미술사조이다.
4) 20세기 초 프랑스를 중심으로 전 세계에 퍼진 문예·예술사조의 하나이다.
5) 1960년대 초반부터 1970년대에 걸쳐 일어난 국제적인 전위예술 운동
6) 심미적인 즐거움을 위하여 선택한 음들의 조합
7) 악음을 제외한 비체계적이고 불쾌한 음들의 조합
8) 컴퓨터 프로그램의 오류로 발생하는 비정상적인 노이즈

불리며 전체 또는 부분이 자율시스템에 의하여 생성된 예술을 의미한다. 사람이 어떤 규칙을 결정하고 컴퓨터 시스템이 정해진 규칙 안에서 의사결정의 일부를 수행하도록 허용하는 것이다. 현재 음악, 비주얼 아트(visual art)[9], 건축 등 다양한 분야에 활용되고 있다. 실생활에서 일반인들이 쉽게 접할 수 있는 비주얼 아트의 예로는 PC(personal computer)의 스크린세이버(screensaver)[10]가 있으며 창작자가 만들어 낸 비주얼 아트는 어떠한 알고리즘에 안에서 변화하는 모양을 만들어 스스로 움직이는 것 같은 착각을 일으킨다.

본 연구는 음악과 영상 두 분야의 특징을 면밀히 관찰하고 섬세한 접근을 통하여 노이즈와 글리치 사운드를 이용한 음악의 제작과 음악의 음량데이터를 통해 실시간으로 영상을 제어하는 멀티미디어 작품을 제작하여 두 가지 예술의 자연스러운 융합을 이루고자 연구를 시작하였다.

9) 시각에 의해 인식할 수 있는 작품을 제작하는 예술의 한 형태

10) 컴퓨터의 화면을 보호하기 위한 프로그램으로 컴퓨터가 사용 중이 아닐 때 움직이는 그림이나 기하학적인 패턴으로 채운다.

Ⅱ. 기술 연구

1. 사운드 디자인

1) Sound Forge를 이용한 사운드 디자인

본 연구에서는 Sonud Forge[11]를 사용하여 글리치 사운드를 제작하고 편집(editing) 하는 작업을 하였다. 글리치란 컴퓨터 프로그램의 일시적인 오류로 인해 짧은 시간 동안 정상적인 기능을 수행하지 못하는 현상을 말한다. 재생장치의 기계적 오류로 인해 영상의 재생 중 화면이 왜곡되거나 멈추는 현상이 발생하고 음악이 끊기거나 음질이 저하되는 등 다양하게 나타나는 컴퓨터나 기계의 일시적인 오류를 뜻하는 말이다. 컴퓨터 안에서 사운드 데이터를 처리할 때 보통은 버퍼에 저장한 후 저장한 데이터를 다시 꺼내 사용하는 과정을 거치는데 만약 이때 예상치 못한 오류가 발생하여 버퍼에 있는 데이터가 정상적으로 재생되지 않는 경우 발생하는 노이즈를 글리치 사운드라고 한다. 이런 오류의 결과로 발생한 사운드는 일회성이고 사용자가 의도하지 않고 전혀 예상하지 못한 결과이다. 따라서 그것을 다시 똑같이 재현하거나 원하는 때에 반복하여 재사용 하는 것은 불가능 하다. 그러나 스피커나 여러 가지 전자기기에서 발생하는 전기적 잡음[12]을 직접 수음하여 DAW[13]를 사용해 편집하거나 정상적인 사운드에 변조(modulation)[14]를 과하게 거는

11) Sony에서 개발한 오디오 편집 프로그램으로 2015년 영국의 Magix 사가 인수하였다.
12) 통신이나 방송 주변기기 등의 정상적인 기능을 방해하는 전자파의 방해현상
13) Digital Audio Workstation의 약자로 디지털 오디오 데이터를 처리하는 장비이다.

방법 등을 사용하면 글리치 사운드를 인위적으로 만들 수 있다.

Sonud Forge는 오디오 편집 프로그램이기 때문에 오디오 파일 형식(audio file format)[15]만을 사용해야 한다. 그러나 문서나 동영상 혹은 사진 등의 파일을 raw audio(.raw)[16] 형식으로 재생 할 경우 프로그램 안에서 발생하는 오류의 결과로 글리치 노이즈와 비슷한 사운드가 재생된다. 사운드 효과로 글리치 노이즈를 사용할 목적이라면, 사운드가 일정 길이 이상이 되어야 한다. 즉 사운드의 길이는 파일의 종류 및 용량에 따라 다른데 예를 들어 사진파일의 경우 재생되는 길이가 매우 짧아 사운드 효과로 사용이 불가능하다. 반면에 동영상 파일은 사진파일에 비해 비교적 긴 노이즈가 생성되긴 하였지만 주로 화이트 노이즈(white noise)[17] 사운드로만 채워진 경우가 많았다. 연구 결과 '.exe'형태의 파일이 노이즈를 비롯하여 다양한 질감의 소리들이 발생하였고 용량이 큰 가상악기 프로그램 파일들도 효과가 좋다. 본 연구에서는 Ableton Live9[18]의 Pack[19] 설치 파일을 사용하였다. 위의 방법으로 만들어진 사운드들 중에 마음에 드는 소리를 발견했다면, 편집과정 없이 사운드 전체를 작품 안에서 사용하여도 되고 길이가 너무 길면 필요한 부분을 편집하여 사용하면 된다.

---

14) 주파수나 음량을 변화시켜 사운드를 변형 하는 방법.
15) 디지털 오디오 데이터를 컴퓨터 시스템에 저장하기 위한 여러 가지 형태의 파일들을 말한다.
16) 압축되지 않은 오디오 데이터를 저장한 오디오 파일 형태이며 기술적 테스트 이외에는 사용되지 않고 PCM오디오 데이터와 함께 사용된다.
17) 모든 주파수 대역에서 패턴이 무작위하며 동일한 에너지 분포를 가지는 소리
18) Ableton사에서 만든 DAW이며 Max for Livre를 사용해 Max와 연동된다.
19) Ableton에서 제공하는 가상악기의 형태.

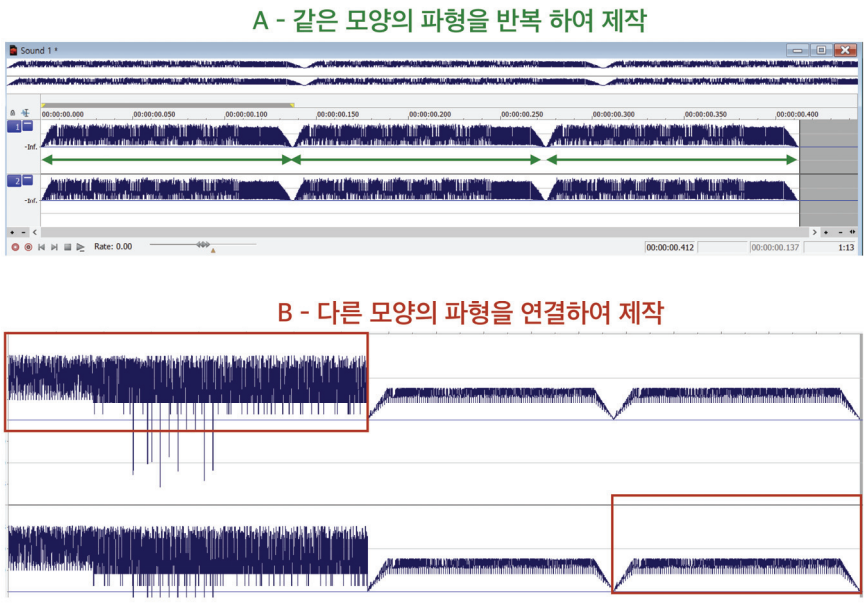
작품에 사용된 사운드들은 특정 부분의 파형을 잘라낸 후 여러 번 반복시켜 원하는 음색의 사운드를 만들어 내는 방법을 사용하였다.

[그림-1] 샘플 단위의 파형 편집 방법

[그림-1]에서 보는 것과 같이 우선 샘플로 사용할 파형의 시작점과 끝부분은 제로 크로싱(zero crossing)[20]이 되는 부분을 잘라야 하며 만약 불가능할 경우 페이드인(fade-in)[21]과 페이드아웃(fade-out)[22]을 통해 시작과 끝 지점의 음량 값을 0(zero)으로 만들어 줘야 한다. 샘플이 깨끗이 잘려야만 반복할 때 부드럽고 자연스러운 소리가 발생된다.

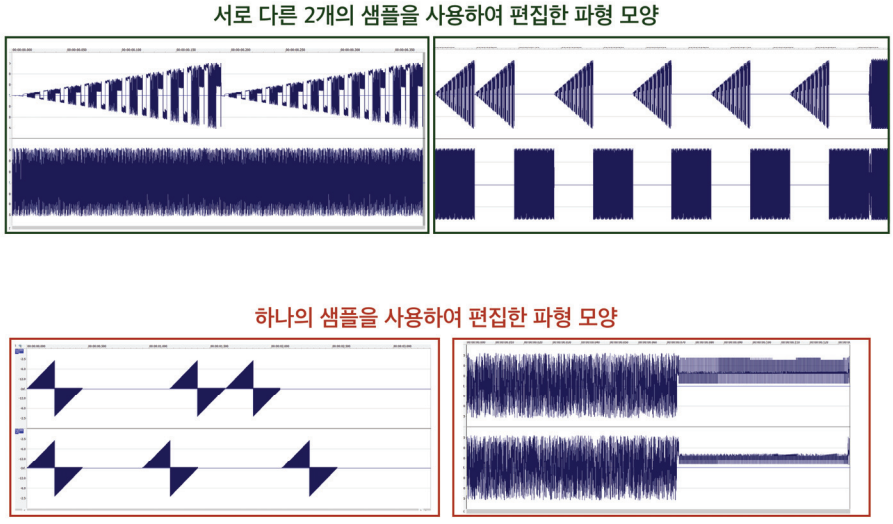
20) 파형의 시작점과 끝점이 0의 음량 값을 갖는 것을 말한다.
21) 음량이 서서히 증가하는 것을 말한다.
22) 음량이 서서히 감소하는 것을 말한다.

[그림-2]의 A와 같이 하나의 샘플을 여러 번 반복시켜 소리를 만드는 것이 기본구조이다. 그러나 B처럼 파형이 다른 두 개의 음원을 합쳐서 사용하거나 합쳐진 파형 전체를 하나의 주기를 가진 샘플로 간주하고 여러 번 반복하여 새로운 사운드를 만들었다.



[그림-2] 파형의 반복을 통한 새로운 파형 제작

스테레오 샘플일 경우 양쪽의 볼륨 엔벨로프(volume envelope)[23]를 서로 다른 모양으로 편집하여 사운드를 제작하였다. 단일 샘플을 사용했을 때는 일반적인 패닝(panning)[24] 효과와 비슷한 결과가 나타난다. 그러나 사운드의 어느 한쪽을 원본과 다른 샘플로 교체하고 편집했을 경우 각 샘플의 음색차이로 인하여 패닝의 효과와 더불어 두 개의 사운드가 서로 섞이며 입체감이 더해진다. 이렇게 완성한 샘플을 여러 번 반복하여 재생을 하면 일정한 리듬 패턴을 보이는 경우가 많았으며 이로 인해 사전에 예측하지 못한 리듬적인 아이디어를 얻을 수 있다.



[그림-3] 스테레오 파형의 편집

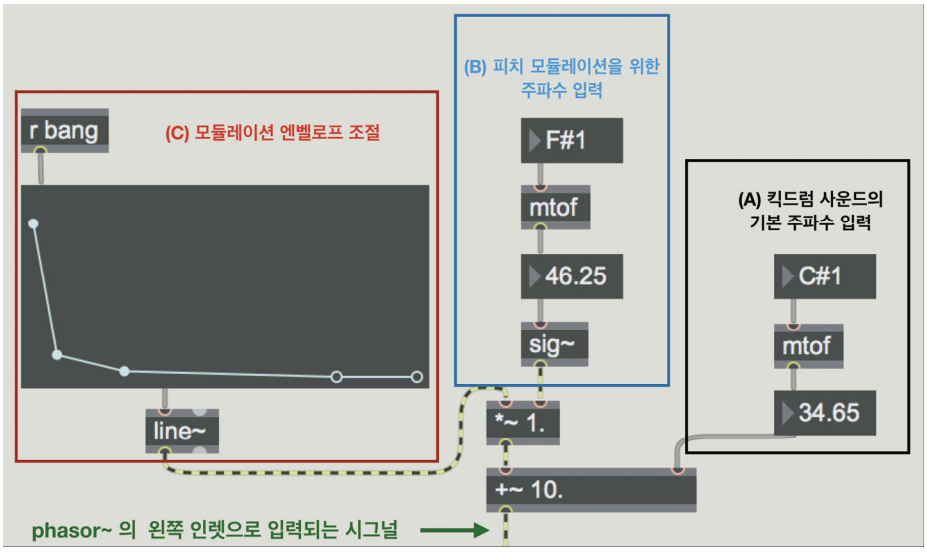
23) 시간의 흐름에 따른 볼륨의 변화를 기록한 것으로 보통은Attack, Decay, Sustain, Release 로 구성되기 때문에 ADSR 이라는 약자로 사용된다.

24) 2개의 스피커에서 나오는 사운드의 출력 값이 다를 경우 나타나는 사운드의 이미지 변화

2) Max/MSP를 이용한 악기 제작

① Phase Distrotion 합성방식을 사용한 킥 드럼(Kick Drums) 디자인

MSP를 이용하여 페이즈 디스토션(Phase Distortion)[25]합성방식을 만들어 킥 드럼(Kick Drums)[26] 사운드를 제작하였다.

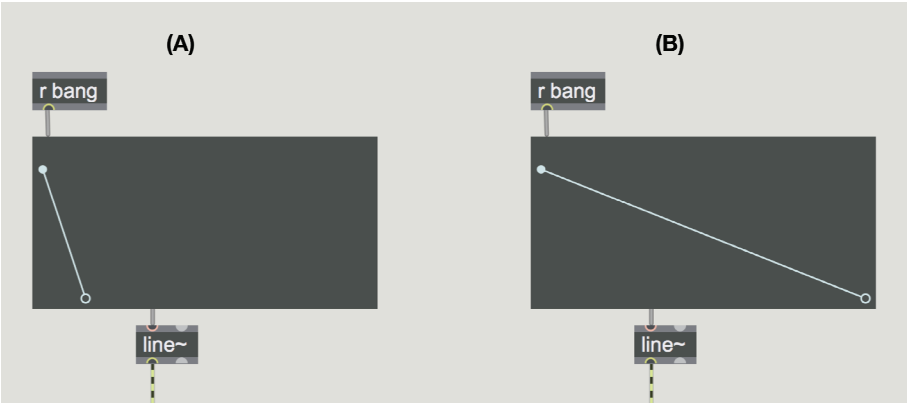


[그림-4] 주파수 입력 부분 패치

[그림-4]의 (A)와같이 +~오브젝트[27]의 오른쪽 인렛(inlet)[28]으로 원하는 주파수 값을 입력한다. 이 값은 킥 드럼 사운드의 기본 주파수 값(initial

25) 디지털 신디사이저 합성방식중의 하나로 본래음의 위상을 찌그러트리거나 위치를 변화시켜 음을 만들어 내는 방식
26) 베이스드럼이라고도 말하며 드럼에서 가장 낮고 강한 음을 내는 역할을 한다.
27) 입력되는 데이터를 더하는 역할을 하는 Max 오브젝트.
28) Max 오브젝트의 위쪽에 위치하며 데이터의 입력을 받는 역할을 한다.

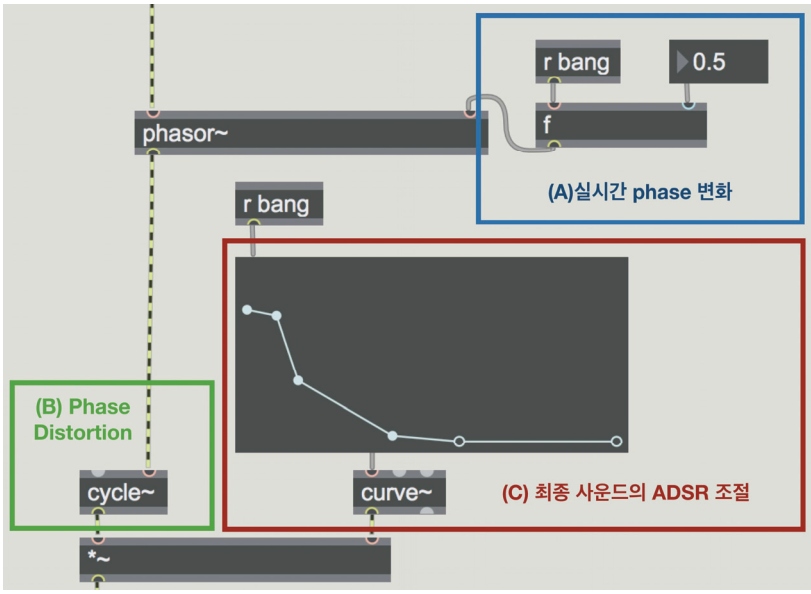
frequency value)이 된다. (B)에서 +~의 왼쪽 인렛 으로 들어오는 값이 모듈레이션을 위한 주파수를 설정한다. 보통의 경우 (C)와 같이 모듈레이션 효과의 극대화를 위해서 엔벨로프는 좌측상단 최고점 에서 부터 시작하여 0까지 우하향하는 모양을 가져야 하며 어택(attack)[29]은 없거나 최대한 짧아야 한다. function오브젝트[30]를 사용하여 원하는 모양과 양을 조절 한다.



[그림-5] 기울기가 다른 두 개의 엔벨로프 모양

[그림-5]는 모듈레이션 엔벨로프 모양의 예를 나타낸다. (A)와 같이 기울기가 작은 모양을 가지면 저음의 배음들이 강조된 소리가 나고 (B)와 같은 모양일 경우 높은 주파수의 배음들이 강조되는 소리가 난다. 모듈레이션 엔벨로프 모양을 조절하면 음의 높낮이와 음색이 변화되어 킥드럼을 비롯한 다양한 타악기 소리를 만들 수 있다.

29) 사운드가 발생한 시점부터 그 크기가 최고점에 도달할 때 까지 걸리는 시간
30) 사용자가 지정한데로 데이터를 출력해주는 Max 오브젝트



[그림-6] 최종 사운드 출력 부분 패치

[그림-6]은 사운드의 최종출력 부분의 패치이다. +~에서 더해진 데이터는 phasor~오브젝트[31]의 왼쪽 인렛으로 들어가며 phasor~의 아웃렛에서 나오는 데이터는 페이즈 디스토션 효과를 위해 cycle~오브젝트[32]의 오른쪽 인렛으로 들어간다. 오른쪽 인렛으로 0. 과 1. 사이의 값을 입력하여 cycle~에 기본(default)으로 저장된 파형의 페이즈(phase)[33] 위치를 실시간으로 변화 시킬 수 있는데 이러한 특징을 이용하면 phasor~의 주파수를 사용해 cycle~의 메인 주파수를 컨트롤 할 수 있다.

위와 같은 방법으로 킥 드럼의 기본음색을 정하였다면 사운드의 중요한

31) 톱니파를 발생시키는 Max 오브젝트
32) 코사인 웨이브를 발생시키는 Max 오브젝트
33) 파형의 사이클에서 특정시점의 위치를 말한다. 사이클은 파형의 시작점으로 되돌아가는데 필요한 간격으로 정의된다.

요소 중 하나인 트랜지언트(transient)[34]를 만들어야 한다. phasor~ 또한 cycle~과 마찬가지로 오른쪽 인렛을 사용하여 실시간으로 파형의 페이즈 위치를 변화 시킬 수 있는데 이점을 이용하여 트랜지언트를 재현 할 수 있다. [그림-6]의 (A)와 같이 Max의 number오브젝트[35] 안에 사용자가 원하는 값을 입력하면 그 값은 f[36]오브젝트에 저장이 되며 bang을 받을 때마다 phasor~의 오른쪽 인렛으로 f에 저장된 숫자가 입력되는 패치이다. 0.25이상일 때부터 사운드의 변화가 일어나며 0.3에서 0.45 사이의 값에서 페달이 드럼 패드를 타격할 때 나는 마찰음과 비슷한 소리가 재현 되어 자연스러운 효과가 나타난다. 마지막으로 (C)의 function을 사용하여 원하는 엔벨로프 모양을 만들어서 사운드를 디자인 할 수 있다. 타악기의 경우 어택시간이 매우 짧거나 아예 없는 경우가 많음으로 (C)와 같은 모양의 엔벨로프 가진다. 그러나 이것은 하나의 예일 뿐이며 여러 형태의 엔벨로프 모양을 시도하여 다양한 사운드를 만들어 낼 수 있다. 음악의 장르 및 사용자의 취향에 따라 다르지만 주로 30 - 120Hz 사이에서 다양한 킥 드럼 사운드를 만들 수 있으며, 사운드의 길이는 700ms을 넘지 않는 선에서 조절을 하면 원하는 결과를 얻을 수 있다. 연구 결과 보통의 경우 30-60Hz 사이에서 저음이 매우 강조되는 드럼 사운드를 만들 수 있었고 60-100Hz 사이의 경우 저음이 점차 감소하며 전형적인 전자음악의 드럼 사운드를 만들 수 있다.

---

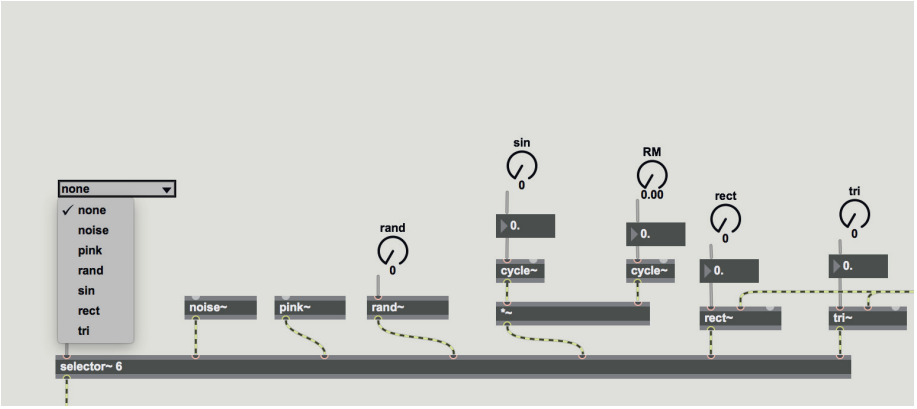
34) 트랜지언트는 어택과 서스테인이 매우 짧은, 순간적인 타격음의 특성을 말하며 사운드의 특징을 나타내 주는 요소이다.

35) 숫자를 입력하고 보여주는 역할을 하는 Max 오브젝트

36) 소수점 단위의 숫자를 저장하고 출력하는 Max 오브젝트로 float의 약자이다.

② 글리치 사운드를 위한 Max for Live 악기 제작

글리치 사운드를 제작하기 위한 목적으로 Max/MSP에서 제공하는 여러 종류의 오실레이터(oscillator)를 활용하여 악기를 제작하였다.

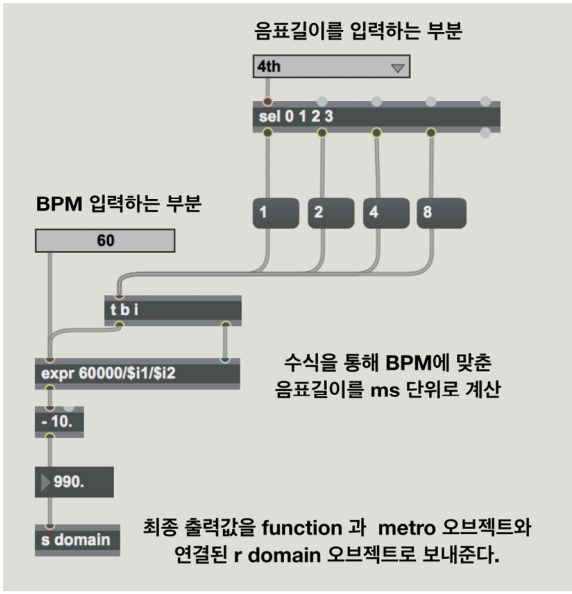


[그림-7] 6개의 오실레이터의 연결 패치

[그림-7]과 같이 MSP의 noise~[37], pink~[38], rand~[39], cycle~, rect~[40], tri~[41] 총 6개의 오브젝트가 프리셋으로 선택되었다. cycle~에는 링 모듈레이션(Ring Modulation)[42]효과를 적용할 수 있고 live.dial오브젝트[43]를 사용하여 모듈레이션 양을 조절 할 수 있다.

37) 화이트 노이즈(white noise)를 발생시키는 오브젝트 이다.
38) 핑크 노이즈(pink noise)를 발생시키는 오브젝트 이다.
39) -1에서 1사이의 랜덤한 값으로 구성된 시그널을 발생시키는 오브젝트 이며 인풋으로 입력되는 숫자에 의해 주파수가 정해진다.
40) 펄스 파(pulse wave)를 발생시키는 오브젝트 이다.
41) 삼각 파(triangular wave)를 발생시키는 오브젝트 이다.
42) 두 개의 오실레이터의 출력을 곱하여 파형을 변형시키는 모듈레이션 방식이다.
43) 다이얼을 사용해 출력 값을 조절하는 오브젝트로 Max for Live에서 사용한다.

그리고 여러 개의 데이터를 인렛으로 받아서 원하는 데이터만을 아웃렛으로 내보내는 selector~오브젝트와 live.menu오브젝트[44]를 연결하여 각 오실레이터를 필요에 따라 선택하여 사용 할 수 있다.



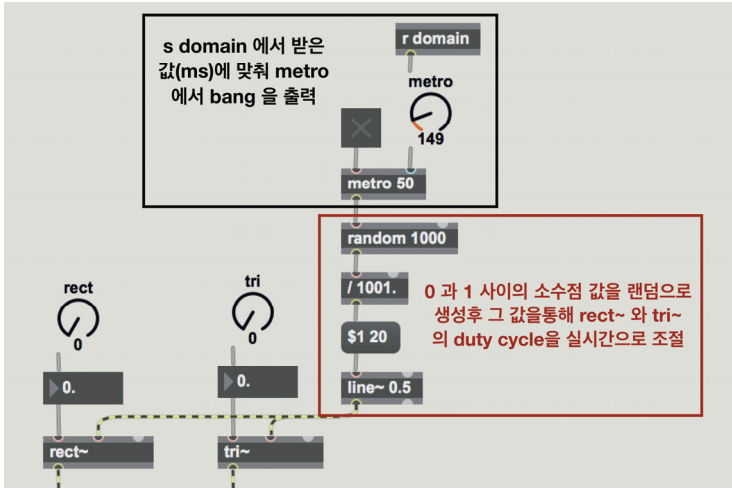
[그림-8] BPM을 ms으로 변환하는 계산 패치

악기를 실제 음악제작에 사용할 경우 BPM(Beat Per Minute)[45]에 맞춰 4분음표, 8분음표, 16분음표 및 32분음표를 밀리세컨드(ms) 단위로 바꾸는 과정이 필요하다. 60000을 BPM으로 나누면 4분음표의 길이가 ms 단위로 도출된다. [그림-8]과 같이 expr오브젝트[46]를 사용하여 60000을 60BPM 으로 나누면 1000이라는 숫자가 나오며 이것은 BPM이

44) 텍스트를 사용해 출력되는 숫자를 제어할수 있는 Max for Live 오브젝트.
45) 1분당 비트수를 나타내며 음악에서 템포를 나타내는 단위로 사용된다.
46) 다양한 연산자를 사용해 계산을 하여 값을 출력하는 Max오브젝트

60일 경우 4분음표의 길이는 1000ms 임을 나타낸다. 그리고 4분음표의 경우 1, 8분음표일 경우 2, 16분음표일 경우 4, 그리고 32음표일 경우 8로 나누면 BPM에 따른 각 음표의 길이를 구할 수 있다. [그림-8]의 경우에는 4분 음표를 선택하였기 때문에 1로 나누어주었다. 최종 값에서 -10을 해준 값은 send오브젝트[47]를 통해 function으로 보내준다. function은 사운드의 재생이 완전히 끝나지 않고 다시 bang을 받을 경우 시작점으로 되돌아가는 과정에서 오류가 발생 할 가능성이 있다. 빠른 BPM의 곡에서 16분 음표의 사운드가 지속적으로 실시간 재생이 될 경우 발생 할 수 있는 오류를 미연에 방지하기 위해 -10을 하였다.

47) 패치코드 연결없이 데이터를 보내는 Max오브젝트

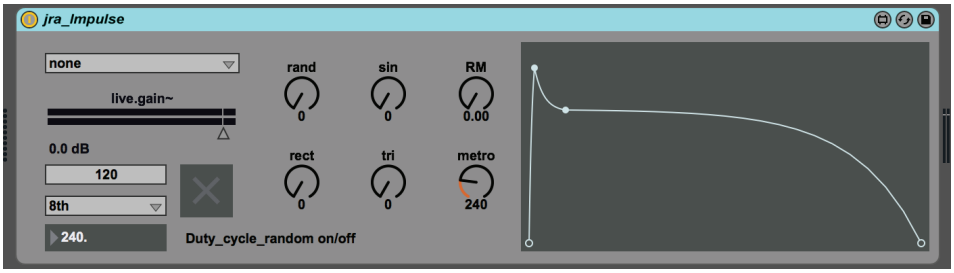


[그림-9] duty cycle 조절 패치

rect~ 와 tri~는 duty cycle[48]을 조절할 수 있는 인렛이 오브젝트의 중앙에 위치해 있다. 0과 1사이의 소수를 받으며 metro[49]와 random[50]오브젝트를 사용해 0에서부터 1까지 무작위로 변하게 하여 실시간으로 조절 한다. metro의 값은 live.dial을 사용해 사용자가 직접 조절 할 수도 있지만 [그림-8]의 BPM계산 패치에서 생성된 값을 받아서 metro와 연결된 live.dial로 자동으로 입력되어 BPM에 맞춰 자연스러운 변화를 준다. random은 소수점의 숫자를 랜덤으로 생성하지 못한다. 따라서 1과 0 사이의 소수점 단위의 숫자를 만들기 위해선 사용자가 임의로 수식을 사용하여 조절해 줘야 한다. 위의경우 random의 아규먼트(argument)[51]로 숫자 1000을 입력하였으므로 random은 bang을

48) 펄스 주기에서 펄스폭의 비율을 나타내는 수치.
49) 일정한 간격으로 bang 메시지를 출력하는 Max 오브젝트
50) bang을 받으면 사용자가 지정한 값 안에서 무작위로 숫자를 출력하는 Max 오브젝트
51) Max 오브젝트에 초기에 입력하는 값으로 오브젝트의 동작을 제어한다.

받을 때 마다 0에서 999까지의 숫자를 출력하게 된다. 출력된 숫자를 1001.로 나누면 0에서 1 사이의 소수점 단위의 랜덤 한 숫자를 만들 수 있다.



[그림-10] 악기의 실제 사용 이미지

[그림-10]은 실제 Live9에서 악기를 로딩(loading)[52] 했을 때의 모습이다. 노이즈 위주의 사운드를 벗어난 다양한 음색을 구현하고 실제 음악 작업 중에 필요한 사운드를 즉시 만들기 위한 목적으로 제작을 하였다. 짧은 길이의 사운드를 사용해 곡의 템포(tempo)에 맞춰 반복되는 루프(loop)[53]를 만드는 작업의 경우에 Max for Live악기를 사용하면 Live9 에서 미디 시퀀싱(midi sequencing)[54] 기능을 사용하여 편리하게 작업을 진행 할 수 있고 전체적인 작업시간을 단축 할 수 있는 장점이 있다.

52) DAW에서 가상악기 등을 사용하기 위하여 불러오는 행위
53) 일정한 패턴을 가지고 반복되어 사용되는 음악적 프레이즈.
54) 프로그램을 사용해 MIDI이벤트를 기록하는 작업

③ 데시메이션(Decimation)을 이용한 사운드 디자인

데시메이션은 사운드의 샘플링 레이트(sampling rate)[55]를 줄이는 과정을 말한다.[56] MSP의 degrade~오브젝트를 사용해 샘플링 레이트와 비트 뎁스(bit depth)[57]를 인위적으로 줄이는 방법으로 노이즈와는 다른 음색의 기계적인 사운드를 만들 수도 있고 1970-80년대의 전화기나 디지털캠코더 같은 전자기기에서 재생되는 사운드들을 모방 할 수 있다. degrade~의 가장 왼쪽 인렛은 시그널을 받는다. 가운데 인렛은 0에서 1사이의 값을 사용해 샘플링 레이트를 조절한다. 즉 1인 경우 컴퓨터와 연결된 오디오 인터페이스(audio interface)[58]의 샘플링 레이트 설정 값을 말하며, 0.5면 설정 값의 1/2을 의미한다. 오른쪽 인렛은 1에서 24의 값을 받으며 이것은 비트 뎁스를 조절한다.

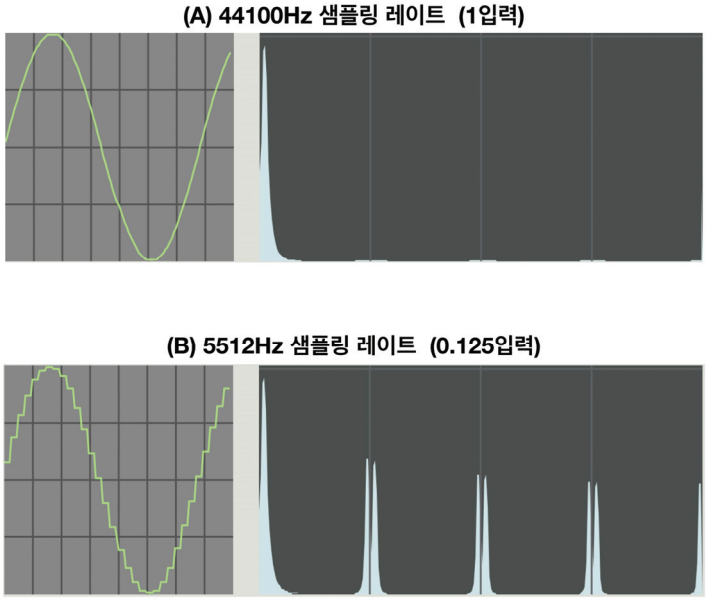
---

55) 단위시간당 샘플링 횟수를 의미하며 단위시간은 초(second)를 사용한다.

56) Lyons, Richard (2001). Understanding Digital Signal Processing. Prentice Hall. p. 304. ISBN 0-201-63467-8. "Decreasing the sampling rate is known as decimation."

57) 샘플링된 사운드의 해상도를 의미한다.

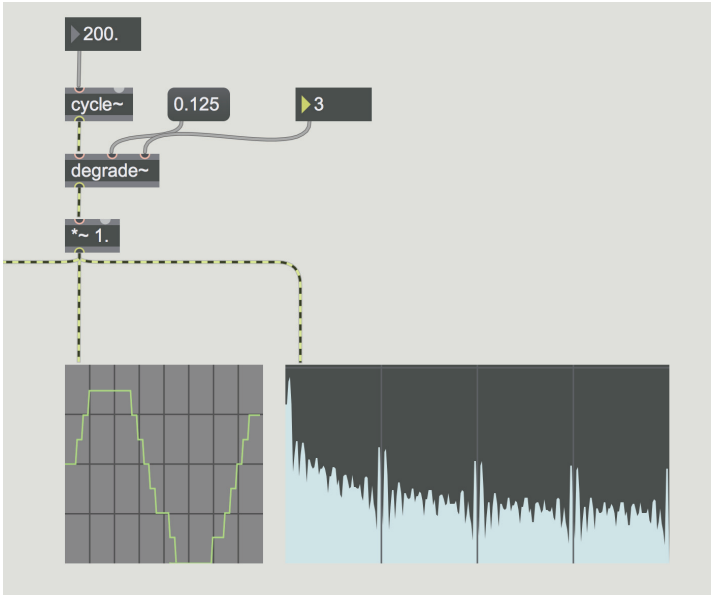
58) 사운드 입/출력 및 오디오 신호변환 장치



[그림-11] 샘플링 레이트 변화에 따른 파형 변화

[그림-11]은 200Hz의 사인파형을 재생하는 cycle~과 degrade~를 연결한 후 degrade~를 사용해 샘플링 레이트를 변화시켜 두 파형의 결과를 비교한 것이다. (A)는 샘플링 레이트의 변화가 없는 파형 모양이다. (B)의 경우는 0.125를 입력하여 44100Hz 샘플링 레이트의 1/8 즉 5512Hz의 샘플링 레이트로 재생했을 때 파형의 모습이다. 파형의 모습이 (A)와는 다르게 투박하며 계단 모양이 생긴다. 오른쪽의 그래프를 보면 앨리어싱(aliasing)[59] 현상으로 인하여 다른 배음들이 생긴 것을 확인 할 수 있다. 이런 배음들이 전기 노이즈와 비슷한 사운드를 만들어 주며 기계적인 사운드 효과를 만들어 준다.

59) 아날로그 사운드가 디지털로 변환되는 샘플링 과정에서 주파수의 정보가 손실되어 변형되는 현상



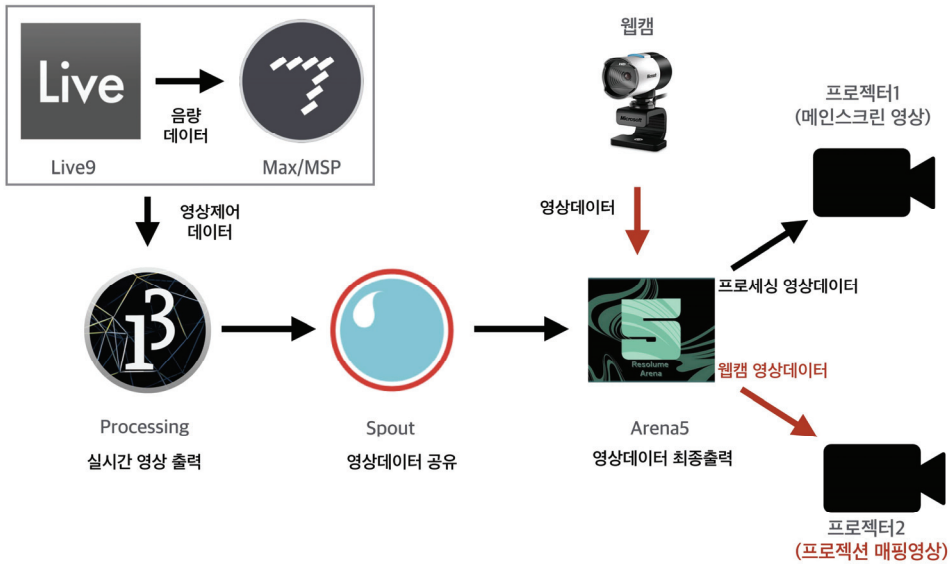
[그림-12] 비트 뎁스를 변화시킨 파형의 모습

[그림-12]는 degrade~의 오른쪽 인렛을 통해 비트 뎁스를 변화시켰을 때의 파형의 변화 모습이다. 위의 경우는 3을 입력했고 2의 3승인 8단계로 나눠진 것을 의미한다. 왼쪽의 파형을 보면 0부터 시작해 8개의 계단모양이 생긴 것을 확인 할 수 있다. 오래된 스피커나 오디오에서 나오는 전기 노이즈와 비슷한 소리를 단순한 사인파형을 가지고 재현 할 수 있다. degrade~는 사인파형뿐 아니라 어떠한 소리도 다운샘플링을 할 수 있기 때문에 사람의 목소리나 피아노 소리 등과 같은 사운드도 사용이 가능하다. 그러나 노이즈 계열의 사운드(화이트 노이즈, 핑크 노이즈 등)를 사용하면 단순히 로우패스 필터(low pass filter)[60]를 사용한 것 같은 사운드를 재현하기 때문에 그 효과가 미미하다.

60) 사용자가 지정한 주파수 대역의 저음부분을 통과시키고 나머지는 감쇄시키는 역할을 하는 필터

2. 영상 시스템

본 작품의 영상 시스템의 구성도는 [그림-13]과 같다. Live9에서 출력이 되는 사운드의 음량 데이터는 Max패치 안에서 영상을 제어하는 데이터로 가공되어 Processing[61]으로 보내진다. Processing에서 출력되는 영상은 Spout[62]를 통해 Arena5[63]로 입력되며 그와 동시에 웹캠의 영상정보도 Arena5로 입력된다. 영상의 최종출력은 Arena5를 통해 두 대의 프로젝터로 보내지고 메인스크린과 매핑을 위한 오브제로 출력이 되는 시스템이다.



[그림-13] 영상 시스템 구성도

61) 2001년 MIT Media Lab의 Casey Reas와 Ben Fry가 개발한 자바(JAVA)기반의 프로그래밍 언어이자 개발 환경이다.
62) 윈도우에서 사용하는 실시간 영상데이터 공유 프로그램
63) Resolume사에서 만든 VJing 과 매핑을 위한 영상 프로그램이다.

1) Processing을 사용한 Generative Art 영상 제작

음악에 실시간으로 반응하는 영상 제작을 위하여 Processing을 사용하였다. Max를 통한 OSC[64]통신을 사용하여 영상의 실시간 제어가 가능하고 텍스트 기반의 프로그램이기 때문에 Jitter와 비교하여 컴퓨터의 리소스를 적게 사용한다는 장점이 있다. 특히 기하학적인 모양의 그래픽을 만들기에 적합하여 작품에 사용하였다.

컴퓨터 모니터는 픽셀(pixel)[65]이라는 단위로 이루어져 있고 화면의 각 픽셀은 모니터의 어떤 한 지점의 위치를 결정하는 좌표이다. 각 픽셀의 위치는 x 와 y 라는 두 개의 숫자로 나타 낼 수 있다[66]. 기본적으로 한 쌍의 x, y 좌표가 있으면 모니터 상에 점(point)을 표현 할 수 있고, 두 쌍의 좌표가 있다면 그 점 두 개를 연결하여 선을 그릴 수 있다. 또한 점들의 연결을 통해 간단한 도형뿐 아니라 기하학적인 도형들도 그릴 수 있다. 본 작품에서는 매개변수 방정식(parametric equation)을 응용하여 그래픽을 디자인 하였다. 매개변수 방정식이란 x 와 y의 좌표가 매개변수(parameter)라 불리는 변수 t의 함수 (x = f(t), y = f(t)) 로 주어진다면 t의 변화에 따라 x, y좌표도 변하게 되는데 이 함수에 의해 움직이게 되는 점들의 집합을 매개변수 곡선(parametric curve)라고 하며 이것을 도출한 수식을 매개변수 곡선의 매개변수 방정식(parametric equation)이라고 한다.[67] 즉 컴퓨터

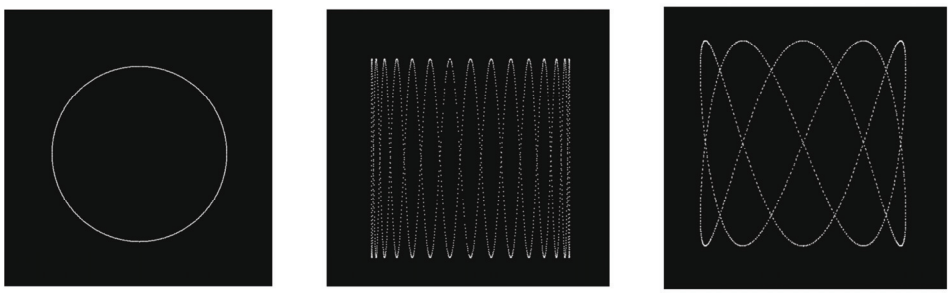
---

64) Open Sound Control의 약자로 컴퓨터와 다른 디지털 장비들이 데이터를 주고 받는 네트워크 통신을 말한다.

65) 화면을 구성하는 가장 작은 단위로 Red, Blue, Green, Alpha값의 데이터를 포함한다.

66) Shiffman, Daniel (August 19, 2008), Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction (1st ed.), Morgan Kaufmann

모니터가 직교좌표계(cartesian coordinate system)[68]라고 가정하면 각 픽셀의 위치는 x, y의 좌표로 표현이 가능하고 그 값을 어떠한 함수로 정해준다면 그 움직임을 사용하여 기하학적인 모양의 도형을 그릴 수 있다.

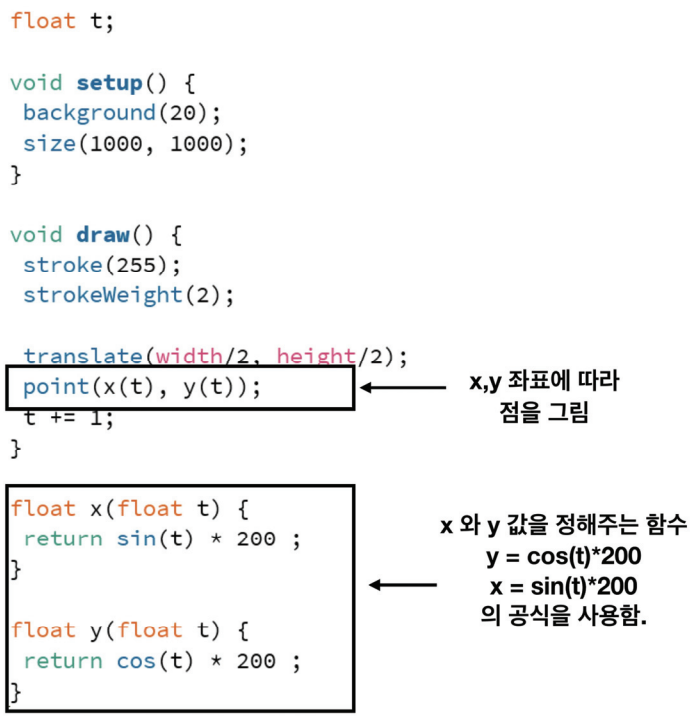


[그림-14] 매개변수 방정식을 사용해 제작한 도형 예제

[그림-14]는 간단한 매개변수 방정식을 사용해 Processing에서 구현한 다양한 도형들의 예이다.

67) Weisstein, Eric W. "Parametric Equations". MathWorld.

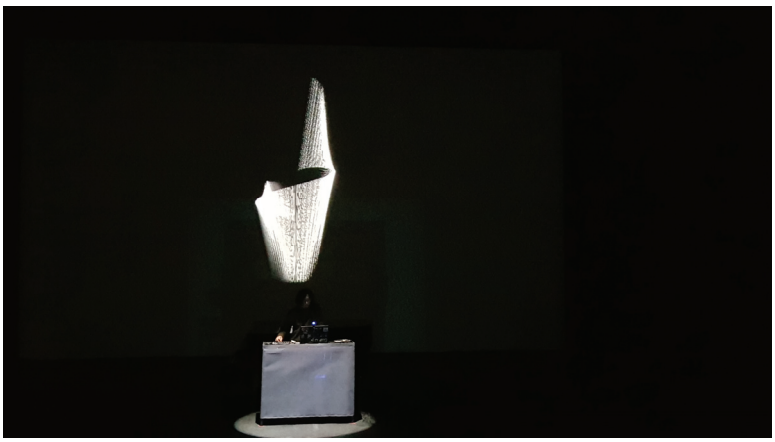
68) 프랑스의 수학자 데카르트가 발명한 좌표계이며 x 와 y좌표를 사용해 좌표상의 위치를 정의한다.



[그림-15] 매개변수 방정식의 Processing 코드

[그림-15]는 [그림-14]의 도형들을 그려낸 실제 코드이다. x 와 y의 값은 함수에 의해 정의 되고 그 값에 따른 좌표위치에 점을 그려 주었다. 그 점들의 집합이 모양을 만들어 낸다. x, y 와 관련된 함수만 변경하면 다양한 도형이 그려진다는 것을 [그림-14]에서 확인 할 수 있다. 간단한 예로 위의 코드를 실행하면 완벽한 원모양을 그려내는데 만약 *200의 숫자를 *100 으로 변경해주면 원의 크기는 작아진다.

[그림-16]은 [그림-15]의 코드를 응용하여 실제 작품에 적용한 사진이다.



[그림-16] [그림-15]의 코드를 응용한 실시간 그래픽 생성

[그림-17]은 작품의 실제 코드의 일부이다. 점을 그려주는 함수를 선을 그려주는 함수로 바꾸고 선을 그리기 위해서 필요한 4개의 x1, y1, x2, y2 좌표를 함수로 지정해 주었다. 이 좌표들의 움직임에 따라 선이 그려진다.

```
numLines = oscy/10;
for (int i = 0; i < numLines; i ++) {
  line(x1(t+i), y1(t+i), x2(t+i), y2(t+i*2));
}
t += oscy*0.002;
}
float x1(float t) {
  return sin(t/10) * 190 + sin(t /5) * 50;
}
float y1(float t) {
  return cos(t/10)* 10 + cos (t/15) * oscx*0.95;
}
float x2(float t) {
  return sin(t/10) * oscy/4 + sin(t) * 20;
}
float y2(float t) {
  return cos(t/20)* 20 + cos(t/12) * oscy/2;
```

← 선을 그리는 함수

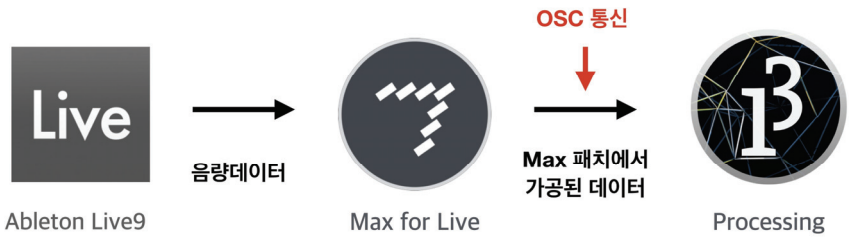
← x1, y1, x2, y2 네개의 좌표값을 위한 방정식

[그림-17] 작품에 사용한 Processing 코드 예시

2) OSC (Open Sound Control)통신을 이용한 영상의 제어

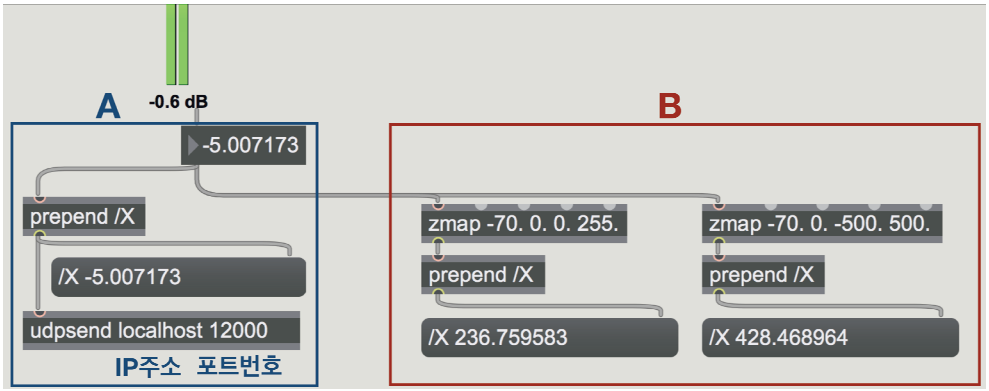
① Processing 영상의 실시간 제어

음악과 영상의 상호작용(interaction)을 위해서는 두 가지 중요한 요소를 정해야 한다. 첫째는 음악의 어떤 요소를 사용하여 영상을 제어 할 것인가 하는 것이고 둘째는 영상의 어떤 요소를 어떻게 제어 할 것인가 하는 것이다. 본 작품에서는 음악의 음량 값의 변화를 가지고 영상의 모양과 색을 제어 했다. 또한 1개의 영상을 제외한 모든 영상은 매개변수 방정식을 응용하여 만들었기 때문에 x 와 y의 좌표를 정의 해 주는 매개변수를 제어하면 음량 데이터를 통한 영상의 모양 제어가 가능하다. 음악의 음량데이터를 Processing으로 전달하기 위해 OSC 통신을 사용하였고 데이터를 전달하고 제어 하는 역할은 Max를 사용하였다. [그림-18]은 음량데이터가 Max에서 가공되어 Processing으로 전달되는 데이터의 흐름을 나타낸 그림이다.



[그림-18] OSC 통신을 통한 데이터의 전달

[그림-19]의 A는 Live9에서 출력되는 음량 값의 데이터를 OSC통신을 통해 Processing으로 전달하는 Max패치이다. live.gain오브젝트[69]에서 나오는 음량 데이터는 prepend오브젝트[70]를 사용하여 데이터 앞에 "/X"를 붙여 udpsend오브젝트[71]로 입력되고 OSC통신을 위해 udpsend에 입력된 IP주소[72]와 포트(Port)번호[73]를 사용하여 데이터를 Processing으로 보낸다.



[그림-19] OSC통신을 위한 Max 패치

데이터 앞에 /X를 붙여서 출력하는 이유는 데이터들 중에 /X와 함께 들어오는 데이터만을 걸러주기 위함이다. 음량 데이터를 사용한 실시간 영상제어를 위해서는 Processing내의 모양과 색을 제어할 수 있는 적합한

69) 오디오 시그널을 조절 할 수 있고 현재 음량을 확인 할 수 있는 Max for live 오브젝트 이다.

70) 입력되는 데이터 앞에 사용자가 지정한 숫자나 문자를 앞에 붙여 출력하는 Max 오브젝트

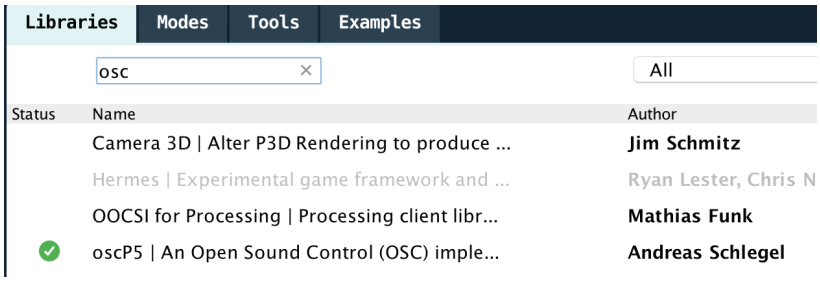
71) 네트워크를 통해 메시지를 전달하는 Max오브젝트로 User Datagram Protocol(UDP)를 사용한다.

72) 인터넷규약주소(Internet Protocol address)의 약자로 컴퓨터 네트워크에서 장치들이 서로 통신을 하기 위해서 사용하는 특수한 번호 이다.

73) 데이터의 입출력을 위해 설정하는 컴퓨터내의 통로 번호

데이터로 변경하는 과정이 중요하다. [그림-19]의 B는 데이터 변환 과정의 예이다. 음량 값을 통해 그래픽의 색상을 조절 하고 싶다면 -70에서 0으로 출력되는 데이터를 Processing에서 사용되는 0에서 255사이의 숫자로 바꿔줘야 한다. 같은 음량 값이라도 Max의 zmap오브젝트[74]를 사용하여 Max패치 안에서 조절하는 방법에 따라 양쪽의 최종 출력 값이 달라지는 것을 보여주고 있다.

Processing에서 OSC 통신을 통한 데이터를 받기 위해서는 oscP5 내장 라이브러리를 먼저 설치해야 한다. [그림-20]은 프로세싱의 라이브러리 설치 화면이다.

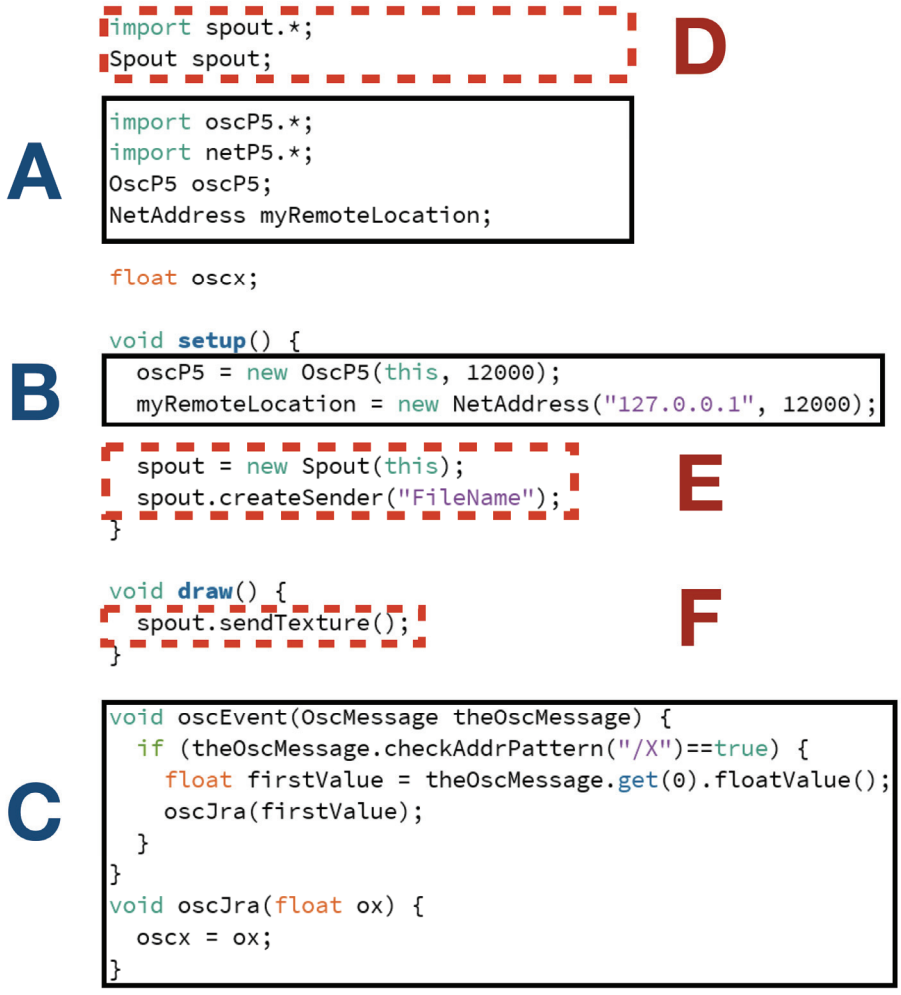


[그림-20] Processing에서 oscP5 라이브러리 설치 화면

[그림-21]은 Processing에서 oscP5와 Spout를 사용해 OSC 통신을 하기 위해 필요한 기본 코드이다. A는 oscP5를 사용하기 위해 설치된 라이브러리를 호출(import) 하는 코드이며 라이브러리를 사용하기 위해서 꼭 필요한 과정이다. B는 새로운 oscP5 객체를 생성하고 데이터를 받을 IP주소와 포트를 설정하는 코드이다. C는 사용자가 임의로 정한 특정 메시지를 조건문에 지정을 하고 그 메시지와 같이 들어오는 데이터만을 입력 받는

74) 입력된 값의 범위를 사용자가 정한 범위의 값으로 변화시켜 출력해주는 오브젝트.

코드로 데이터의 이동 간 혹시 모를 데이터의 혼선을 방지하는 역할을 한다.



[그림-21] oscP5와 Spout를 사용하기 위한 Processing 코드

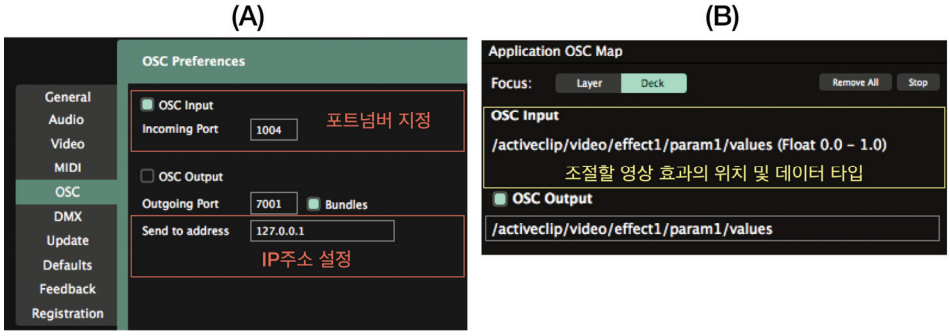
이 때문에 [그림-19]의 Max 패치에서 prepend를 사용하여 출력되는 데이터 앞에 Processing에서 지정한 특정 메시지와 동일한 메시지를 붙여서 출력한다. 또한 입력된 데이터를 프로그램 안에서 사용하기 위하여 oscx라는 변수로

지정을 해줘야만 프로세싱 코드 내에서 사용 할 수 있다.

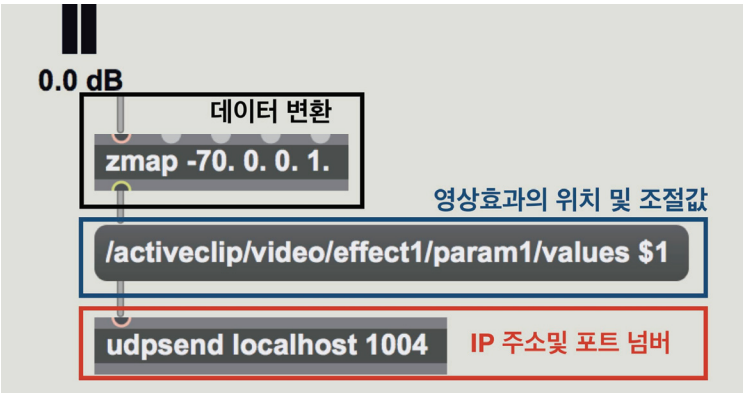
Processing에서 나오는 영상을 Arena5에서 사용하기 위해서 Spout라는 영상공유 프로그램이 필요하다. [그림-21]의 빨간 점선으로 표시된 부분이 Spout를 사용하기 위한 코드이다. 제일 먼저 Processing에서 제공하는 Spout 라이브러리를 설치 한 후 호출을 해야 한다. D부분의 코드가 그 역할을 한다. E는 새로운 Spout 객체를 만들고 “FileName” 이라는 곳에 사용자가 원하는 이름을 입력하면 된다. F의 코드를 통해 Arena5 보내진 영상은 Arena5의 Source메뉴에서 사용자가 임의로 입력한 이름으로 찾을 수 있고 실시간으로 사용 가능하다.

② Arena5의 실시간 영상효과 제어

재생되는 사운드의 음량 값에 따라 Arena5에 내장된 영상효과를 실시간으로 제어하여 영상의 모양이 변하도록 설계하였다. [그림-22]는 Arena5와 Max의 OSC통신을 위해서 IP주소 및 포트넘버를 지정해 주는 과정을 나타낸다. (A)는 Arena5의 'OSC Preferences' 설정 창이다. 'OSC Input' 의 'incoming port'번호를 지정해 주고 'Send to address' 의 IP주소를 설정하면 된다. (B)는 조절할 영상효과의 위치 및 데이터 종류를 나타낸다. 이창은 Arena5의 'Mapping' 탭에서 'Edit Application OSC Map'을 클릭하여 볼 수 있고 조절하고 싶은 영상효과를 클릭하여 필요한 데이터를 확인 할 수 있다. (B)를 통해서 조절할 파라미터의 값이 0.에서 1. 사이의 소수점 단위의 값을 받는 것을 알 수 있다.



[그림-22] Arena5에서 OSC 통신 설정 방법

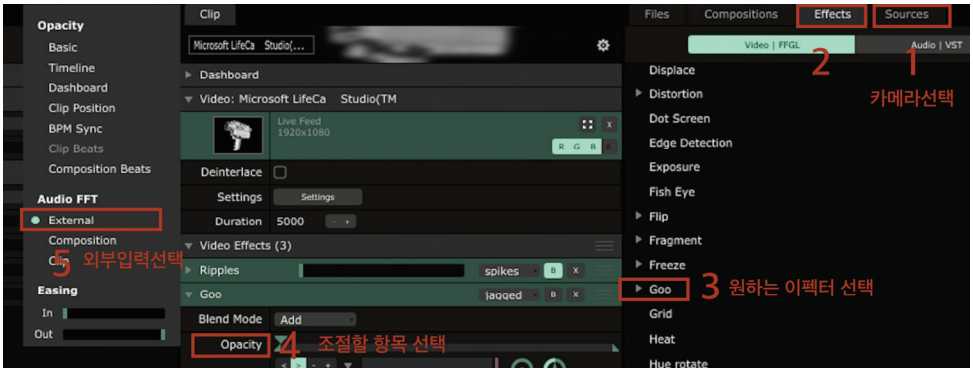


[그림-23] Arena5로 데이터를 보내기 위한 Max패치

[그림-23]은 Max에서 Arena5로 데이터를 보내기 위한 패치이다. zmap을 사용하여 -70에서 0의 값을 1과0 사이의 소수점 값으로 변환해 주고 영상효과의 위치 맨 마지막 단에 입력한 $1으로 영상효과 조절 값을 입력하여 udpsend를 통해서 최종 출력이 된다.

3) 프로젝션 매핑의 실시간 영상 효과 제어

Arena5에서 기본으로 제공하는 Live Feed기능과 내장 영상효과를 사용하여 웹캠의 영상을 무대 위 테이블에 프로젝션 매핑(projection mapping)[75] 하였고 컴퓨터 내장 마이크로 입력되는 사운드의 음량 값으로 영상효과의 양이 실시간으로 조절 되도록 하였다.



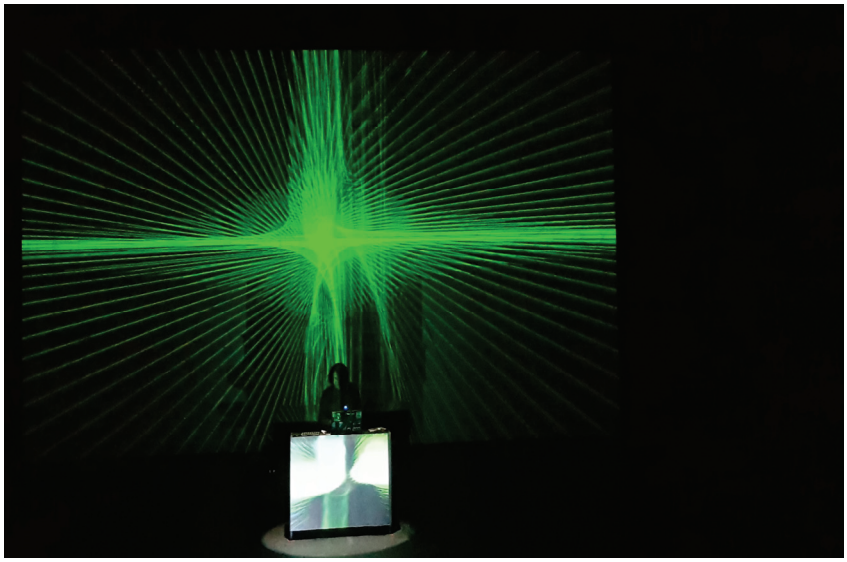
[그림-24] Arena5에서 외부입력을 사용한 실시간 영상효과 조절 방법

[그림-24]는 Arena5 에서 Live Feed에 적용하는 영상효과를 노트북의 마이크로 수음된 음량 값을 사용해 실시간으로 조절하는 방법을 나타내는 그림이다. 오른쪽 'Sources'탭을 선택하여 카메라 입력을 선택한다. 그 옆의 'Effects'탭을 선택하여 원하는 효과를 선택 한 후 'Video Effects'라는 항목으로 끌어서 넣는다. 위에서는 Goo[76]라는 영상효과를 선택하고 적용하였다. 음량 값을 사용해 실시간으로 조절하고 싶은 항목을

75) 미리 제작한 오브제에 프로젝터로 영상을 투사하여 오브제에 시각적인 변화를 주는 기법

76) Arena5에서 기본으로 제공되는 영상 효과

선택하여 클릭 하면 'Audio FFT라'는 항목이 보이는데 'External'이라는 항목을 선택하면 노트북의 내장 마이크를 사용 할 수 있다. 그림에서는 Goo의 Opacity[77] 항목에 적용을 하였다. 마이크로 수음되는 소리의 크기가 커질수록 Opacity의 수치가 증가한다. Arena5에서 실시간으로 변하는 영상정보는 무대 바로 앞에 설치한 프로젝터로 출력되고 무대 위의 DJ테이블에 매핑 하였다. [그림-25]는 실제 공연 사진이며 메인 영상과 더불어 무대 위 연주자의 모습이 테이블에 매핑이 된 것을 볼 수 있다.



[그림-25] 프로젝션 매핑의 실제 공연 이미지

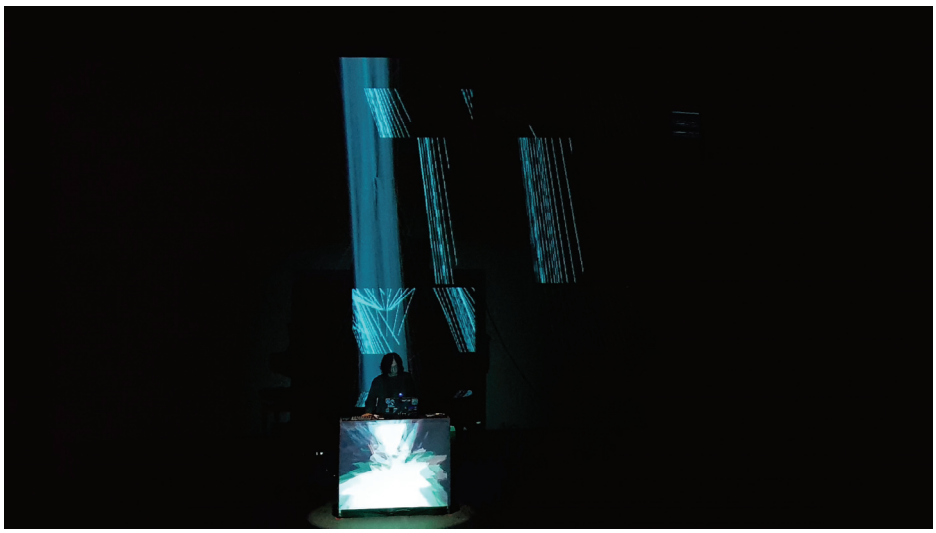
웹캠의 방향이 무대 위 연주자를 향해 있기 때문에 메인 스크린의 영상과 연주자가 같이 캡처(capture) 되면서 매핑영상과 메인영상이 비슷한 색감을 가지고 변하는 것 같은 효과를 주었다.

77) 영상의 투명도를 말하면 보통 값이 작을수록 영상이 투명해 진다.

Ⅲ. 연구 기술의 작품 적용

연구된 시스템과 기술들은 2017년 11월 11일 동국대 이해랑극장에서 진행된 한국멀티미디어학회공연 '보는 소리, 듣는 영상 XIV' 중 <Trans-Human>이라는 작품에 사용되었다.

1. 작품 소개



[그림-26] 실제 공연 이미지

작품 <Trans-Human>은 음악의 음량 값에 의해 실시간으로 영상이 제어되는 멀티미디어음악 작품이다. "현재 인류인 호모사피엔스는 트랜스휴먼으로 진행하는 중이고 결국에는 다음 진화의 종착점인 더 이상 늙지 않고 영원한 삶을 얻는 새로운 종인 포스트 휴먼이 될 것이다"라고 정의 내린 미래 학자, 호세 코르데이로(Jose Luis Cordeiro)[78]교수의 말에

영감을 받아 기계와 인간의 본질이 서로 어디까지 간섭해야 하는가에 대한 고찰을 작품을 통해 표현하고 있다. 컴퓨터 음악의 한 종류인 IDM[79] 장르이며 작품에 사용된 모든 사운드는 Max/MSP를 사용한 합성음과 글리치 노이즈를 사용해 만들었다. 음악에 따라 실시간으로 반응하는 영상 효과는 컴퓨터 그래픽이 자신의 의지로 스스로 살아 움직이는 것처럼 보이도록 디자인 하였다.

78) 대표적인 미래학자로 MIT기계공학과를 나와 베네수엘라 대학교에서 경제학 박사학위를 받았다. 현재 미국 싱귤레러티대학교 교수로 재직 중이다.

79) Intelligent Dance Music의 약자로 90년대 초에 등장한 전자 음악의 한 장르이다. 글리치 사운드가 대표적인 특징이다.

2. 작품 구성

1) 음악 구성

음악의 전체적인 구성은 [표-1]과 같다. 각 파트별로 사용된 주된 사운드는 다르지만 전체적으로 노이즈와 글리치 사운드의 조합이 곡 전체에 사용되었다. A, B, C파트는 가산합성(Additive Synthesis)[80] 및 FM(Frequency Modulation)[81]을 이용한 합성음을 추가로 사용하여 자칫 노이즈와 글리치 사운드의 과도한 노출로 청자에게 피로함을 줄 수 있는 요소를 완화 하였다.

[표-1] 작품 구성

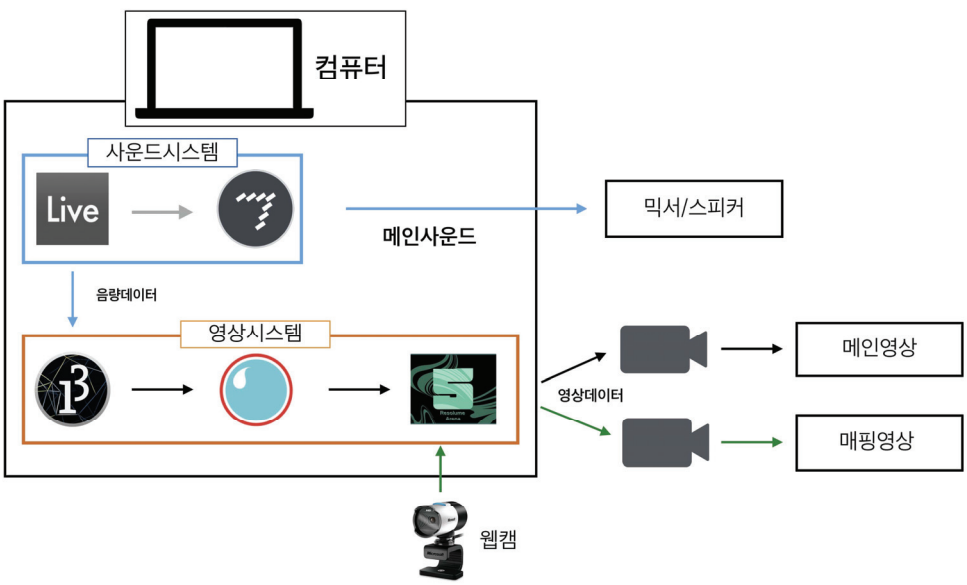
구성	Intro	A	B	C	D	E	Outro
사운드의 구성	-글리치	-노이즈 -합성음	-글리치 -합성음	-합성음	-노이즈 -글리치 -합성음 -킥 -베이스	-노이즈 -글리치 -합성음 -킥 -베이스	-글리치
실시간 모양변화	○	○	○	○	○	○	○
실시간 색상변화		○			○	○	
매핑 영상				○	○	○	○

80) 여러개의 사인파형을 더하여 음색을 만드는 합성방식
81) 주파수 변조를 이용하는 소리의 합성방식

음량 값에 의해 영상의 실시간 제어는 전 파트에 걸쳐 일어난다. 또한 A, D, E파트는 모양의 변화 와 더불어서 음량 값에 의한 영상의 색상 변화도 실시간으로 적용된다. C파트의 중간부터 테이블에 적용한 프로젝션 매핑은 시작되며 마이크로 수음되는 음량 값에 의한 매핑 영상의 실시간 영상효과는 곡의 마지막 까지 적용된다.

2) 무대 및 시스템 구성

무대구성은 메인스크린 앞 무대 정중앙에 DJ테이블을 설치하고 연주자가 테이블 뒤에 위치한다. 연주자는 컴퓨터와 컨트롤러를 사용해 음악의 시작과 끝을 제어하고 공연 중 실시간 영상 제어를 위한 Max for Live패치의 음량 조절을 한다.



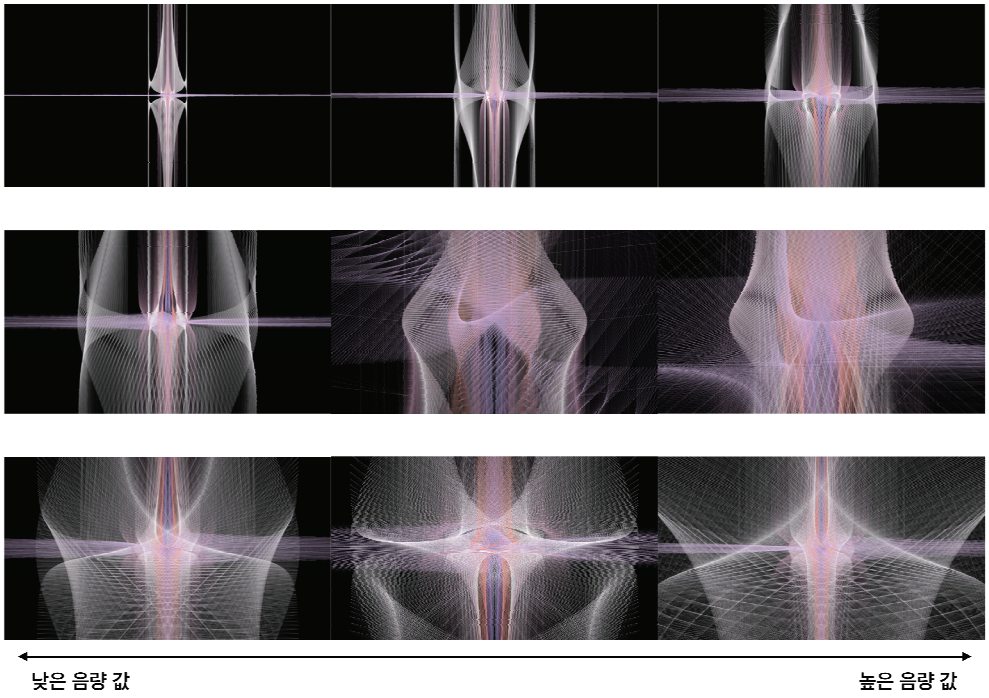
[그림-27] 작품의 시스템 구성도

시스템 구성은 [그림-27]과 같이 사운드 시스템에서 재생된 음악은 메인스피커로 출력되며 동시에 Max로 보내진다. 음량데이터는 Max 패치에서 처리를 거쳐 영상 시스템으로 보내져 영상을 제어하는데 사용되며 실시간으로 제어되는 영상은 프로젝터를 통해 메인 스크린으로 출력되고 웹캠을 통해 받은 영상은 무대 앞에 설치한 DJ테이블에 매핑 된다.

3. 작품에 적용한 음량에 따른 영상의 실시간 제어

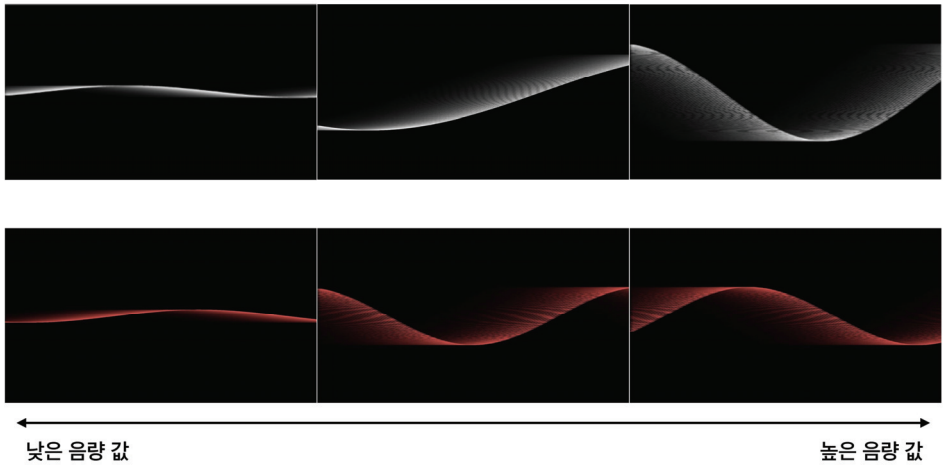
1) Intro 및 Outro의 영상제어

본 작품의 모든 영상은 음악의 음량 값에 의하여 제어된다. [그림-28]은 Intro와 Outro의 영상의 변화를 나타낸다. 분홍색과 흰색 그리고 주황색 계열을 사용하였고 크기와 모양은 음량 값에 의해 연동된다.



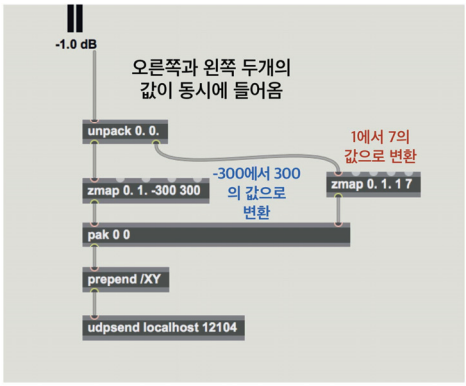
[그림-28] Intro/Outro의 영상 변화

2) A파트 와 B파트의 영상 제어



[그림-29] A파트의 영상 변화

[그림-29]는 A 파트의 영상으로 음량이 커질수록 파형의 상하 진폭이 커지는 직관적인 변화를 나타낸 모양이다. 주된 사운드로 사인파를 사용하였고 사인파형의 모양을 형상화 하는 이미지를 생성하였다. 색상도 음량 값에 의해 실시간으로 제어된다. [그림-30]과 같이 Max에서 음량데이터를 받을 때 스테레오 사운드이기 때문에 오른쪽과 왼쪽 각 채널의 데이터가 2개 들어오며 두 개의 값을 따로 분리하여 Max패치 내에서 데이터를 변환한다. -300에서 300사이의 값으로 들어오는 oscx의 값은 그래픽의 모양을 제어하는 역할을 하면서 [그림-30]의 Processing 코드의 if 구문에서 조건을 제어해주는 역할도 한다. 데이터가 0보다 클 경우 하얀색을 만들고 0과 -200 사이에선 붉은색 그 이하는 옅은 분홍색을 만들어 내었다.



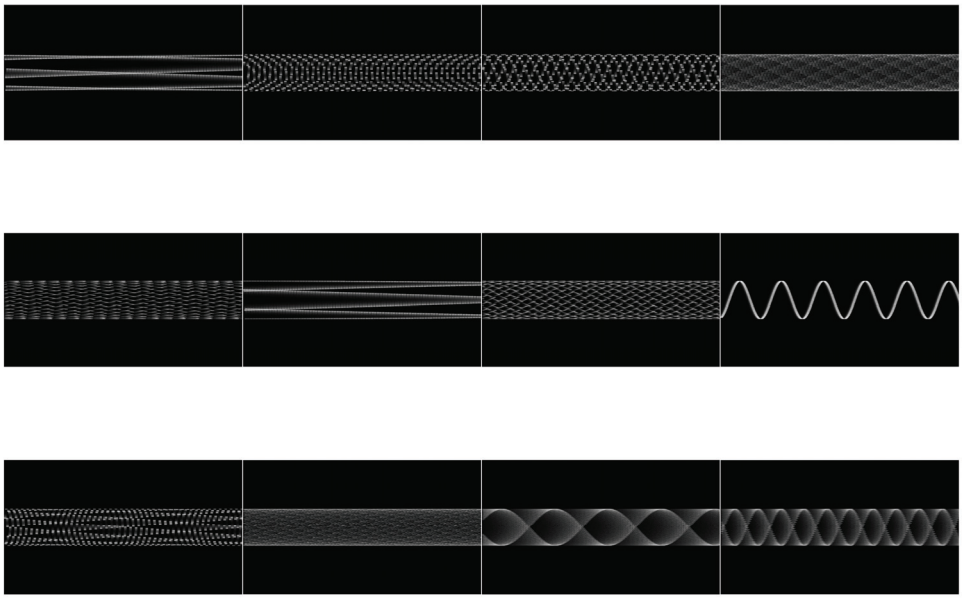

```
if (oscx > 0) {
  fill(255, oscy*35);
} else if (oscx > -200){
  fill(#ED4545, oscy*35);
} else {
  fill(#F2E6E6,oscy*30);
}
```

oscx의 값은 -300 에서
300사이의 값으로 들어옴

oscy의 값은 1 에서 7
사이의 값으로 들어옴

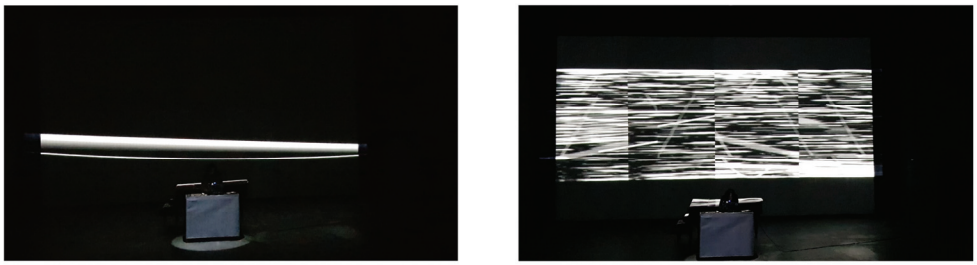
[그림-30] A파트에서 사용한 Max 패치와 Processing 코드

B파트는 A파트와 동일한 Processing 코드를 가지고 음량 값에 따른 변화의 폭을 변하게 하여 A파트와는 다른 이미지를 만들어 내었다. 음량 값의 변화에 일정한 패턴으로 반응을 하는 것이 아닌 랜덤한 영상을 만들도록 설계하였다. [그림-31], [그림-32]



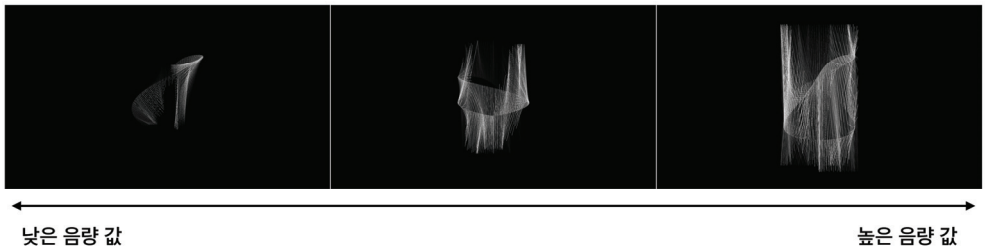
[그림-31] B파트의 영상 변화

[그림-31]은 B파트의 영상들이며 음량 값에 일정한 패턴으로 반응을 하지 않는 랜덤한 영상들을 보여 준다.



[그림-32] A와 B파트의 공연 사진

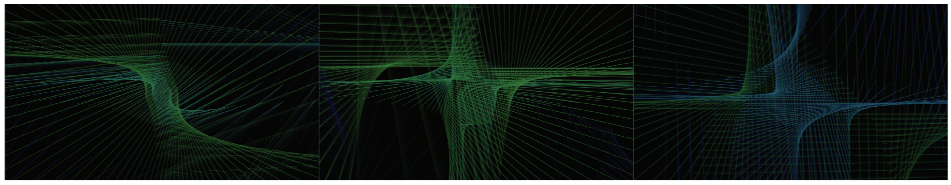
3) C파트와 D파트의 영상 제어



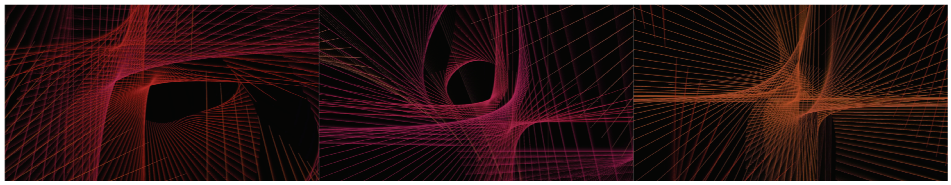
[그림-33] C파트의 영상 변화

[그림-33]은 C파트 영상 변화이다. 음량 값에 따라 실시간 제어 되며 음량 값이 증가하면 그래픽의 크기가 커지고 동시에 그래픽을 구성하는 선의 개수가 증가하지만 음량 값이 작아지면 그 반대가 된다.

(A) 음량 값에 따른 영상의 모양 변화



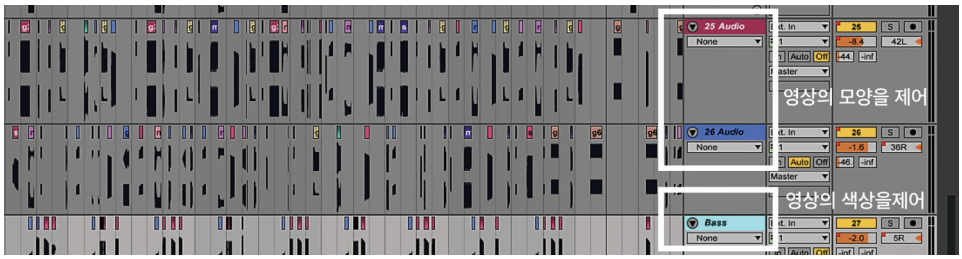
(B) 베이스 음량 값에 따른 영상의 색상 변화



[그림-34] D파트의 영상 변화

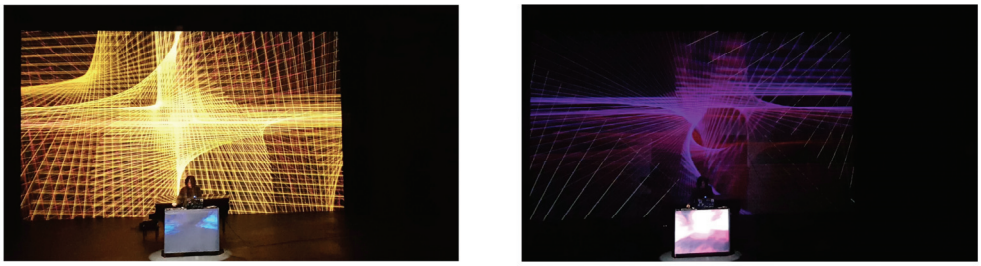
[그림-34]는 D파트의 영상 변화이다. Live9에서 출력되는 3개의 트랙에서 사운드 데이터를 받고 각 음원들은 독립적으로 영상의 요소를 제어한다.

[그림-35]와 같이 짧은 길이의 음원들이 트랙 간에 서로 겹치지 않게 배열하여 각 사운드에 반응하는 영상의 모습이 직관적으로 보이도록 구성하였다. 상단 두 개의 트랙은 영상의 모양을 제어 하는데 사용되고 하단의 베이스 트랙은 색상을 제어하는데 사용된다.



[그림-35] D파트의 Live9모습

[그림-34]의 (A)와 같이 그래픽의 색상은 전반적으로 초록색과 파란색 계열의 색상들이 사용되었으나 베이스 파트의 음량 값 변화에 따라 (B)와 같이 붉은 계열의 색상 변화를 나타낸다.



[그림-36] D 파트의 공연 사진

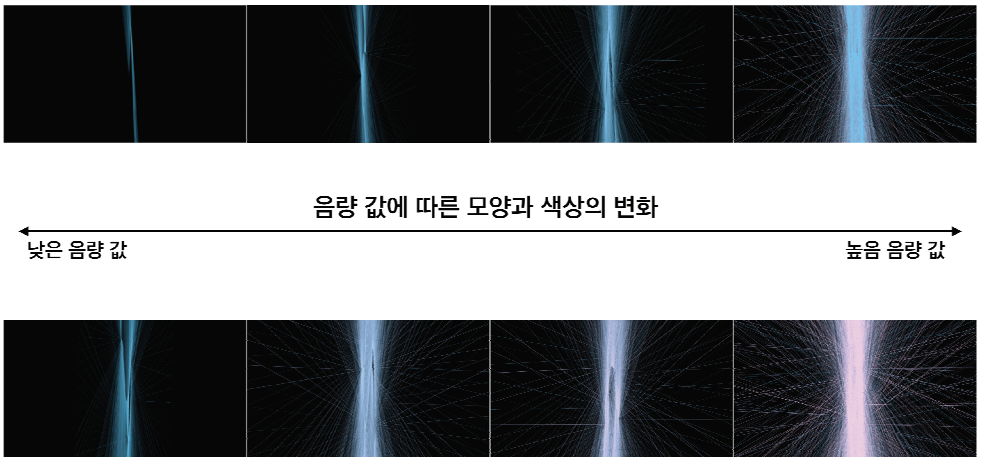
[그림-36]은 D파트의 실제 공연 사진이며 음량 값의 변화에 따라 영상의 모양과 색상이 실시간으로 변하는 것을 볼 수 있다.

4) E파트의 영상 제어

[그림-37]은 E파트의 영상이며 음량 값의 변화에 의해 영상의 위치 및 색상이 변하도록 하였다.

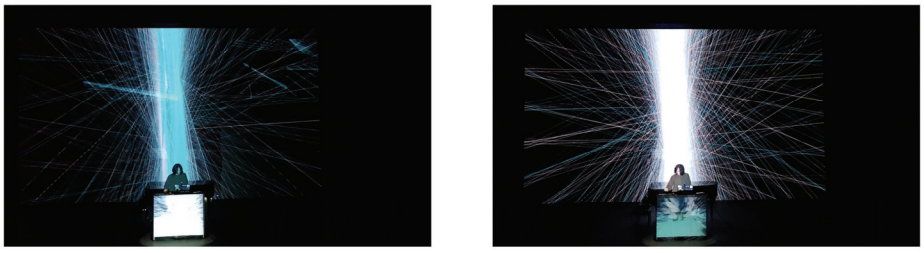
[그림-37] E파트의 영상 변화(1)

음량이 높을수록 화면 하단으로 이동하며 낮을수록 화면의 상단으로 이동한다. 그러나 좌우 위치는 랜덤하게 변한다. 또한 음량 값이 커질수록 색상이 분홍색으로 변하면서 하얀색에 가까워지도록 구성하였다. [그림-38]은 파트E의 또 다른 영상으로 음량에 반응하여 그래픽의 모양이 변하지만 좌우 움직임이 없이 중앙에 위치하도록 하였다. 작품의 상호작용(interaction) 효과를 높이기 위하여 두 영상의 불투명도(opacity)를 조절하여 음량에 반응해 교차로 나오도록 하였다.



[그림-38] E파트의 영상 변화(2)

[그림-39]는 E파트의 공연 사진이며 음량 값에 따라 영상의 색상이 실시간으로 제어 되는 것을 볼 수 있다.



[그림-39] E파트의 공연 사진

음악에 의한 영상의 제어는 전반적으로 음악과 영상이 유기적으로 연결되는 느낌을 가져야 한다. 본 작품에서 음악에 대한 영상의 반응이 매우 직관적 이였다. 특히 음량에 반응하는 제너레이티브 아트는 관객에게 영상이 음악에 맞춰 스스로 움직이는 것 같은 느낌을 전달한다.

Ⅳ. 결론 및 고찰

본 연구는 서로 다른 두 가지 종류의 예술을 융합하기 위한 목적으로 멀티미디어 작품의 형태로 제작하였다. 아직까지 일반인들에게 생소했던 노이즈를 사용한 음악을 제작하고 음악의 음량 값을 데이터화 하여 음악에 실시간으로 반응하는 제너레이티브 영상을 제작하였다.

영상 매체를 통한 시각의 자극은 노이즈 음악이 주는 청각적인 이질감을 완화시켜 주었으며 프로젝션 매핑을 사용한 연출효과는 자칫 지루 할 수 있는 멀티미디어 공연의 재미를 더하고 관객들에게 직관적인 예술적 흥미를 제공하였다.

두 종류의 예술이 하나의 형태로 결합되기 위해서는 매우 세심한 접근이 필요하다. 본 연구에서도 작곡가와 그래픽 디자이너의 두 가지 측면을 매우 면밀하게 살펴가며 연구를 진행해야 했으며 작업의 완성도가 높아지면서 공연을 위한 연주자(performer)의 측면까지 고려해야만 했다. 특히 현대 기술이 많이 발전했다고는 하지만 아직까지 영상의 실시간 렌더링은 컴퓨터의 리소스를 많이 필요로 하는 작업이기 때문에 Max패치와 Processing코드 등의 프로그램을 최대한 간결하게 정리하는 작업이 필수적이며 4가지 이상의 서로 다른 프로그램들을 거쳐야 최종 결과물이 출력 되므로 각 단계별로 프로그램에 맞는 데이터의 변환과 불필요한 작업을 줄이는 것이 중요하다.

과학 기술의 진보로 인한 다양한 디지털 도구나 매체들의 등장은 미디어 아티스트들에겐 새로운 예술의 형태를 창작하고 도전을 할 수 있는 영감을 주고 있다. 새로운 형태의 예술을 시도하거나 서로 다른 두 가지 예술의 조합은 그 시도만으로도 큰 의미가 있다. 하지만 대중들에게 그 예술에 대한

당위성을 얻지 못한다면 그러한 시도의 의미도 곧 사라지고 말 것이다. 음악에 대한 해석은 다양해지고 그 경계가 모호해 지고 있으며 음악을 음악의 장르나 틀 안에서 구분 짓는 것은 현대의 미디어 아티스트에게 큰 의미가 없을 것이다. 음악을 새로운 기술과 창의적으로 결합 하는 방법에 대한 연구와 앞으로 나올 새로운 매체들에 대한 진지한 자세가 필요하며 폭넓은 활용 가능성을 열어두고 그 방법들에 접근하고 연구해나간다면 작품의 결과가 좋아지고 그로인해 대중들의 공감을 얻어 낼 수 있을 것이다. 이것이 예전부터 현재에 이르기 까지 예술의 경계를 넓히기 위해 노력했던 수많은 예술가들의 시도에 같이 동참을 할 수 있는 방법이 될 것이다.

Keyword(검색어):
컴퓨터음악(computer music), 멀티미디어음악(multimedia music), 노이즈(noise), 글리치(glitch), Max/MSP, 소리합성(sound synthesis), 소리시각화(sound visualization), 프로세싱(Processing), IDM

E-mail: j_ra@dongguk.edu

참 고 문 헌

1. 단행본

이상원. “디자이너를 위한 프로세싱.” 교문사, 2016.

Hartmut Bohnacker, Benedikt Gross, Julia Laub, Claudius Lazzeroni. Generative Design: Visualize, Program, and Create with Processing. Princeton Architectural Press, 2012

Charles Dodge, Thomas A. Jerse. Computer Music: synthesis, composition and performance. Schirmer Books, Second Edition, 1997.

Ira Greenberg, Dianna Xu, Deepak Kumar. Processing : creative coding and generative art in processing 2. Friends of, c2013

John Maeda. Design by numbers. MIT Press, 2001.

Matt Pearson. Generative Art: A Practical Guide Using Processing Manning Publications; 1 edition, 2011

Casey Reas, Chandler McWilliams. Form+Code in Design, Art, and Architecture. Princeton Architectural Press, 2010

Snoman, Rick. Dance music manual. Focal Press, 2014.

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.

2. 참고논문

Hernandez, Carlos Roberto Barrios. "Thinking Parametric Design: Introducing Parametric Gaudi." Design Studies 27, no. 3: 309-324. 2006

Frederik De Bleser. The Impact of Generative Design. “The NodeBox Perspective”. Antwerpen, 2016

Mania Aghaei Meibodi, Generative Design Exploration: Computation and Material Practice. Doctoral Thesis 2016

3. 웹사이트

https://processing.org/

https://docs.cycling74.com/max7/

http://www.danieldavis.com/a-history-of-parametric/

ABSTRACT

A Study on Interactive Multimedia Performance using Max/MSP and Generative Art

(focus on Multimedia Music <Trans-Human>)

Na, ji woung

The purpose of this study is to fuse two art genres and ultimately broaden the scope of music. For that reason, the music has been composed using a glitch noise unfamiliar to the public. In addition, computer graphics that change shapes and colors were utilised according to the volume of music. This thesis consists of two-part; a sound design system for glitch noise and real-time image control system.

There are various ways to make glitch sounds artificial such as using a microphone to record sounds from a broken speaker, excessive Amplitude Modulation(AM), Ring Modulation(RM) and so forth. However, in this study, most of the glitch sounds that are used in Trans-Human were made by Sound Forge, which is an audio editing program by Magix Sound. In addition to using Sound Forge, the glitch sounds have been made by sound processing in Max/MSP.

Generative computer graphics were made with Processing, which is based in Java. In order to change the graphic's shapes and colors, OSCp5 was used. The role of OSCp5 is crucial to communicate with Max for Live and Processing. Following the volume of the music, the graphic was changed by communicating between programs in OSC.

The real-time controlled computer graphics complemented the awkward ambience of the noise music the audience felt, and it added a fun element of multimedia performance that could otherwise be boring and provided an interest to the audience. In addition, research on how to creatively combine music with new technologies is needed.

부록 : 첨부 DVD 설명

1. Trans-Human..mov : 2017년 11월 11일 이해랑 예술극장 <Trans-Human> 공연 실황

2. Max/MSP : Max for Live 악기 패치 및 OSC통신을 위한 Max 패치

3. Processing : 작품에 사용한 Processing 코드 1-7