



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

피아노의 실시간 프로세싱을 이용한
멀티미디어음악 제작 연구

(멀티미디어음악 작품 <Midnight Peace>를 중심으로)

지도교수 김 준

동국대학교 영상대학원
멀티미디어학과 컴퓨터음악전공

한 승 욱

2018

석사학위논문

피아노의 실시간 프로세싱을 이용한
멀티미디어음악 제작 연구

(멀티미디어음악 작품 <Midnight Peace>를 중심으로)

한 승 욱

지도교수 김 준

이 논문을 석사학위논문으로 제출함

2017년 12월

한승욱의 음악석사(컴퓨터음악)학위 논문을 인준함

2018년 1월

위원장 정진헌

위원 박상훈

위원 김준

동국대학교 영상대학원

석사학위논문

피아노의 실시간 프로세싱을 이용한
멀티미디어음악 제작 연구

(멀티미디어음악 작품 <Midnight Peace>를 중심으로)

한 승 욱

지도교수 김 준

이 논문을 석사학위논문으로 제출함

2017년 12월

한승욱의 음악석사(컴퓨터음악)학위 논문을 인준함

2018년 1월

위원장 정진현



위원 박상훈



위원 김준



동국대학교 영상대학원

목 차

I. 연구 배경 및 목적	1
II. 기술 연구	2
1. 사운드 시스템	2
1) 사운드 시스템 구성	2
① granular synthesis 음향효과	3
② phase vocoder 음향효과	5
③ pfft~ xover~오브젝트의 활용	8
④ 딜레이 음향효과	9
2. 영상 시스템	11
1) 영상 시스템 구성	11
① Processing을 이용한 영상 제작	12
② Arena5의 영상효과	14
3. 작품 구현을 위한 연동 프로그램	15
1) OSC를 활용한 사운드와 영상의 연동	16
① Max/MSP와 Processing의 연동	16
② Max/MSP와 Arena5의 연동	19
2) Syphon을 활용한 영상 출력	21

4. 작품 설명	23
1) 작품 소개	23
2) 작품 구성	24
3) 무대 구성	26
4) 시스템 구성	27
5. 연구기술의 작품 적용	28
1) A 파트 적용 효과	28
2) 연결구-1 적용 효과	29
3) B 파트 적용 효과	31
4) 연결구-2 적용 효과	32
5) B' 파트 적용 효과	33
6) bridge 파트 적용 효과	34
7) A' 파트 적용 효과	36
III. 결론 및 고찰	37
참 고 문 헌	39
ABSTRACT	41
부록-1 : 작품 <Midnight Peace> 연주 악보	43
부록-2 : 첨부 DVD 설명	46

표 목 차

<표-1> disis_munger~오브젝트 설명	4
<표-2> grain pitch 파라미터 의미	4
<표-3> Arena5의 영상효과	14
<표-4> 전체 구성 장면	25
<표-5> A 파트에 적용된 사운드 파라미터 값 및 영상효과	28
<표-6> 연결구-1에 적용된 사운드 파라미터 값 및 영상효과	30
<표-7> B 파트에 적용된 사운드 파라미터 값 및 영상효과	31
<표-8> 연결구-2에 적용된 사운드 파라미터 값 및 영상효과	32
<표-9> B' 파트에 적용된 사운드 파라미터 값 및 영상효과	33
<표-10> bridge 파트에 적용된 사운드 파라미터 값 및 영상효과	35
<표-11> A' 파트에 적용된 사운드 파라미터 값 및 영상효과	36

그 립 목 차

[그림-1] 사운드 시스템의 구성	3
[그림-2] disis_munger~오브젝트	3
[그림-3] FFT분석의 도식화	5
[그림-4] 음원의 시간을 늘리거나 줄였을 때 frequency 변화 비교	6

[그림-5] FFT를 이용한 phase vocoder 패치	7
[그림-6] 사운드 주파수 대역을 나누는 pfft~ xover~ 패치	8
[그림-7] 딜레이 음향효과를 위한 Max/MSP 패치	9
[그림-8] 영상 시스템	11
[그림-9] 프로세싱으로 생성된 영상	12
[그림-10] 음역에 따른 음량 값을 받는 Processing 코드의 일부	13
[그림-11] 입력 값이 가장 클 때 Processing 원본 영상	13
[그림-12] 입력 값이 가장 클 때 상하반전 영상	13
[그림-13] 데이터 통신의 흐름	15
[그림-14] Processing으로 데이터를 전송하기 위한 Max/MSP 패치	16
[그림-15] oscP5 라이브러리를 설치하는 모습	17
[그림-16] OSC통신을 위한 라이브러리를 불러오는 코드	17
[그림-17] OSC 통신을 위한 주소 설정 코드	18
[그림-18] OSC 통신이 가능하도록 입력 데이터를 변수로 설정하는 코드	18
[그림-19] Arena5의 포트번호 확인	19
[그림-20] Max/MSP에서의 포트번호	19
[그림-21] Arena5에서 각각의 영상효과의 OSC주소를 확인하는 방법	20
[그림-22] Max/MSP에서 Arena5로 OSC 데이터 통신하기 위한 패치	20
[그림-23] Syphon 라이브러리 설치 모습	21
[그림-24] 라이브러리를 호출하는 코드	21
[그림-25] 서버 생성 코드	22
[그림-26] 스크린으로 영상을 보내는 코드	22

[그림-27] <Midnight Peace> 공연 장면	23
[그림-28] 무대 구성	26
[그림-29] 시스템 구성	27
[그림-30] 음량 값에 따른 영상 변화	29
[그림-31] Blur 효과가 적용된 영상	29
[그림-32] granular synthesis에 연결된 panning패치	30
[그림-33] 사운드 프로세싱 값에 따른 Blur효과 적용	32
[그림-34] 음량 값으로 Arena5의 Ripple 이펙트 제어	33
[그림-35] 클라이맥스 부분의 영상변화	34
[그림-36] grain pitch variation이 적용된 영상	35
[그림-37] A' 파트 영상	36

I. 연구 배경 및 목적

클래식 음악은 고전·낭만·후기낭만·현대 순으로 시대가 발전되면서 다양한 음악적 변화가 있었다. 기능 화성의 탈피, 불협화음의 사용 등 기존 음악의 이론을 벗어나려 했으며 20세기에 이르러서는 복잡한 화성을 넘어 조성을 없애는 등 파격적인 시도로 과거 음악의 한계를 극복하려 했다. 기존 음악에서 탈피하고자 한 다양한 시도는 근본적인 음악 재료 및 소재에 대한 고찰로 이어졌으며 과학 기술이 발전되면서 컴퓨터를 통해 새로운 음악적 소재에 대한 본격적 탐구가 시작된다.

본 연구에서는 컴퓨터를 통해 피아노에 실시간 사운드 프로세싱(real-time sound processing)을 하였으며 음악과 영상을 연동시키는 소리시각화 기술(sound visualization)을 작품에 적용한다. 각각의 미디어 콘텐츠(media contents)¹⁾가 개별적으로 구성된 작품과는 차별화를 두어 서로 다른 두 미디어, 즉 음악과 영상의 실시간 융합을 통해 관객과 예술적 소통을 깊이 있게 하고자 한다.

본 연구는 Max/MSP²⁾와 Processing³⁾을 이용하여 피아노를 위한 사운드 및 영상시스템의 제작을 통해 기존의 전통적인 음악과 다른 다채로운 감성을 표현하고자 한다.

1) 디지털화된 문자, 음성, 그림 등의 서로 다른 정보 전달의 수단들이 통합된 것으로 다양한 정보기기를 통해 이러한 다중매체들이 생산, 유통, 소비된다.

2) Cycling'74에서 개발한 소프트웨어로 음악과 멀티미디어를 위한 개발 환경을 제공하는 프로그램이다. 데이터의 연산처리 및 프로그래밍이 가능한 Max와 음향신호 및 MIDI데이터 처리가 가능한 MSP로 나누어진다.

3) MIT 미디어 랩의 'Aesthetics and Computation' 그룹에 의해 2001년 만들어진 오픈 소스 프로그래밍 언어

II. 기술 연구

1. 사운드 시스템

Max/MSP로 만들 수 있는 사운드 효과는 chorus⁴⁾, flanger⁵⁾, comb-filter⁶⁾등 다양하지만 큰 폭의 다이내믹 레인지와 넓은 음역, 어택이 강한 피아노의 특성과 어울리는 사운드 효과를 연구하였다. 피아노만으로 채워지지 않는 공간감과 시간의 차이를 두고 연주하는 효과를 위해 딜레이(delay) 음향효과를 사용하였으며, 다른 악기와 함께 합주하는 효과를 위하여 granular synthesis⁷⁾를 사용하였다. 패드(pad)⁸⁾악기로 활용하기 위해 phase vocoder⁹⁾를 사용하였으며, 모든 사운드 시스템은 Max/MSP를 이용해 구현하였다.

1) 사운드 시스템 구성

사운드 시스템은 [그림-1]과 같이 피아노의 사운드가 Max/MSP를 이용한 사운드 프로세싱을 거쳐 스피커로 출력되는 시스템이다.

4) 딜레이를 응용한 음향효과. 원래의 음에 30-50ms정도 지연시킨 사운드를 혼합하여 두께감과 풍부함을 내는 효과

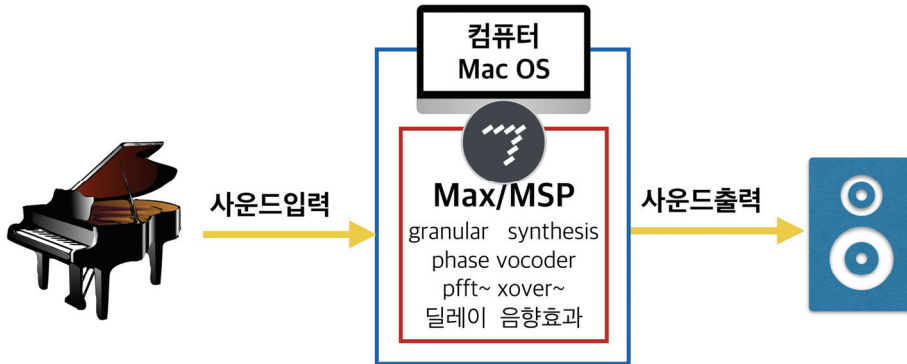
5) 딜레이를 응용한 음향효과. 입력된 음을 약간 지연시켜서 원음에 보태는 방식으로 음색변조를 내는 효과

6) 주파수 특성에 빗살(comb)모양이 나타나는 필터

7) 사운드샘플을 작은 단위(grain)로 나누어 재조합 및 가공하는 소리 합성방식

8) 지속되는 코드 혹은 톤(tone)을 뜻하는 음향 기법

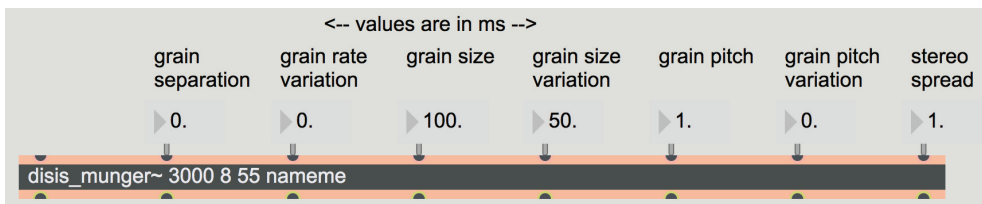
9) 음원을 컴퓨터로 분석하고 그 데이터를 재합성하여 음의 높낮이나 길이를 변형시키는 소리 합성방식



[그림-1] 사운드 시스템의 구성

① granular synthesis 음향효과

granular synthesis는 사운드샘플을 미세하게 나누어 재조합 및 가공하는 소리 합성방식이다. 본 연구에서는 disis_munger~오브젝트¹⁰⁾를 사용하였으며 [그림-2]와 같이 7개의 파라미터가 있다.



[그림-2] disis_munger~오브젝트

[그림-2]의 각 파라미터는 grain¹¹⁾과 연관되어 있으며 파라미터를

10) Max/MSP/Jitter의 외부(external)오브젝트이다. 멀티채널 granular synthesis 기능을 가지고 있고, Max/MSP/Jitter 32bit 환경에서만 작동한다. Columbia University에서 만들었으며 이번 작품에 사용된 버전은 이후에 Virginia Tech에서 수정된 버전이다.

11) grain은 '날알의'라는 의미로 사운드를 작은 알갱이로 쪼개고 다시 조합하여 만들었기 때문에 붙여진 이름이다.

조절하여 다양한 변화를 줄 수 있다. 각 파라미터가 의미하는 바는 <표-1>과 같다.

<표-1> **disis_munger~오브젝트 설명**

grain separation	grain rate variation	grain size	grain size variation	grain pitch	grain pitch variation	stereo spread
샘플 조각의 간격	샘플 조각의 간격이 변화하는 값	샘플 조각의 크기	샘플 조각의 크기가 변화하는 값	샘플의 음정	샘플 음정이 변화하는 값	음장이 퍼지는 정도

<표-1>은 각 파라미터에 대한 설명이며 grain separation, grain rate variation, grain size, grain size variation의 단위는 밀리세컨드(ms)이다. grain pitch의 파라미터 값은 2^n 처럼 2의 지수에 해당되는 값에 따라 옥타브 간격으로 음정이 변화된다. 지수가 0인 경우, 음정의 변화가 없으며 입력되는 사운드의 원 음정으로 출력된다. 1, 2와 같이 양수 일 경우, 한 옥타브 씩 위로 음정이 변형되고 반대로, 지수가 -1, -2로 음수일 경우에는 한 옥타브 씩 아래로 음정이 변형되어 출력된다. 다음의 <표-2>에서 grain pitch의 파라미터 값을 정리하였다.

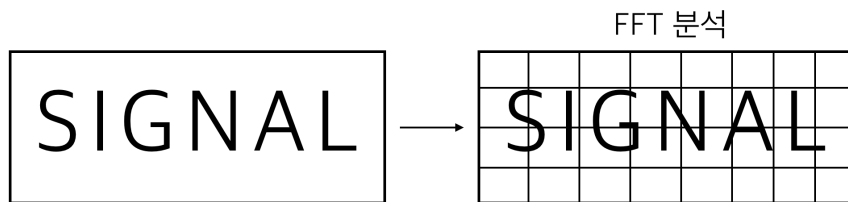
<표-2> **grain pitch 파라미터 의미**

파라미터	$2^0=1$	$2^1=2$	$2^2=4$	2^n	$2^{-1}=0.5$	$2^{-2}=0.25$	2^{-n}
의미	원음	1옥타브 위	2옥타브 위	n옥타브 위	1옥타브 아래	2옥타브 아래	n옥타브 아래

② phase vocoder 음향효과

phase vocoder는 분석-재합성(Analysis-Resynthesis)이라고 불리는 테크닉의 한 종류로서 음원을 컴퓨터로 분석하여 음높이의 변화 없이 그 길이를 조절하는 기술이다. 음원을 길거나 짧게 만들면 음높이도 변화하는 게 물리적인 현상이지만 phase vocoder를 사용하면 음높이의 변화 없이도 길이 변화가 가능하다.

컴퓨터는 FFT¹²⁾(Fast Fourier Transform)분석을 통해 음원파일을 여러 개의 작은 조각으로 나누는 작업을 한다. 아날로그 사진을 스캔하여 디지털화 시킬 때 수 만개의 픽셀로 나누듯이 phase vocoder는 [그림-3]과 같이 FFT 알고리즘을 이용해 음을 여러 개의 bin¹³⁾으로 나눈다.

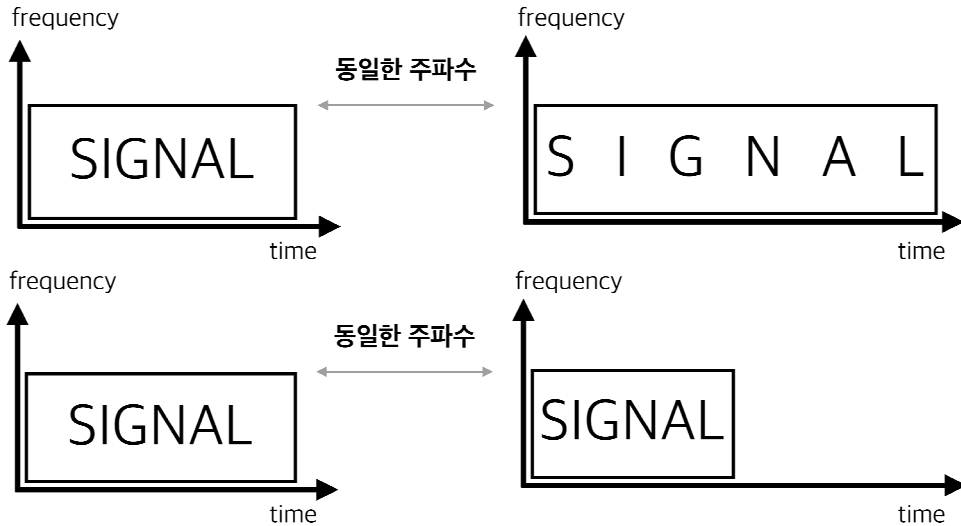


[그림-3] FFT분석의 도식화

나누어진 bin을 다시 원래의 상태로 붙이면 원음이 나오고 bin을 사이에 추가하거나 삭제하여 frequency의 변화 없이 소리를 더 길거나 짧게 변형 할 수 있다. 그러한 원리를 [그림-4]와 같이 나타낼 수 있다.

12) 고속푸리에변환이라고 하며 컴퓨터를 이용하여 복잡한 파형이나 신호들을 보다 간단하게 변환시켜주는 알고리즘을 이야기 한다. 주로 파형을 분석하여 그 데이터를 이용해 원하는 대로 소리를 재합성 할 때 많이 사용된다.

13) FFT에서 분석되는 사운드의 최소 단위



[그림-4] 음원의 시간을 늘리거나 줄였을 때 frequency 변화 비교

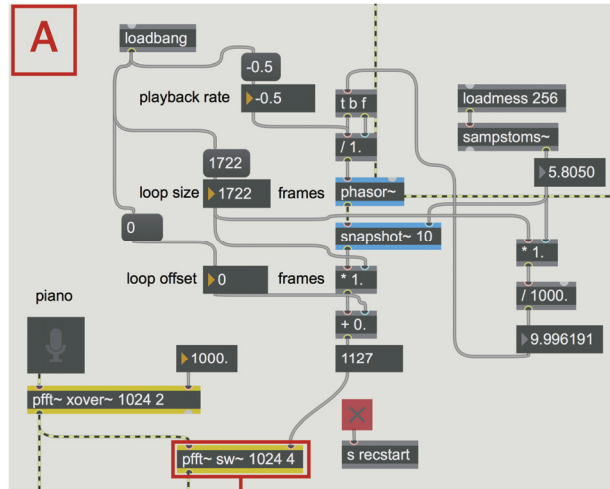
다음의 [그림-5]는 pfft~를 이용한 phase vocoder의 Max/MSP 패치이다. A와 같이, phase vocoder는 분석된 사운드를 읽는 속도인 플레이백 레이트 (playback rate), 반복되는 사운드의 구간을 설정하는 루프 사이즈(loop size), 루프 사이즈에 대한 시작 지점인 루프 오프셋(loop offset)의 파라미터를 갖는다. 각각의 파라미터를 조절하여 다양한 사운드 변형이 가능하다.

[그림-5]의 B에서 보듯이 pfft~오브젝트 안에서 사운드에 대한 FFT분석이 이루어진다. 데이터 가공과정을 지나 buffer~오브젝트¹⁴⁾에 사운드가 저장되면 index~오브젝트¹⁵⁾를 이용하여 저장된 사운드를

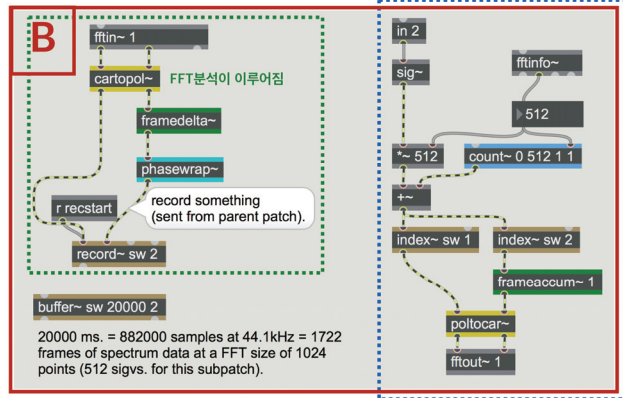
14) 사운드샘플을 컴퓨터 메모리(RAM)에 저장하여 읽을 수 있게 해주는 오브젝트이다. record~오브젝트를 이용하여 buffer~오브젝트에 샘플을 저장할 수도 있고 재생기능을 갖는 여러 오브젝트와 이름을 공유하면 그 샘플을 재생시킬 수 있다.

15) Max/MSP/Jitter에서 재생기능을 갖고 있는 다른 오브젝트들과 달리 사운드를 시간단위로 읽는 것이 아니라 음원의 보간(interpolation)없이 샘플(sample)단위로

샘플단위로 재생시킨다. phasor~오브젝트16)가 0에서 1까지 반복해서 움직이는 것을 이용하여 0에서 1 값을 재생을 원하는 위치의 샘플 값으로 대치시키고 phasor~의 왼쪽 인렛에서 재생속도를 조절한다.



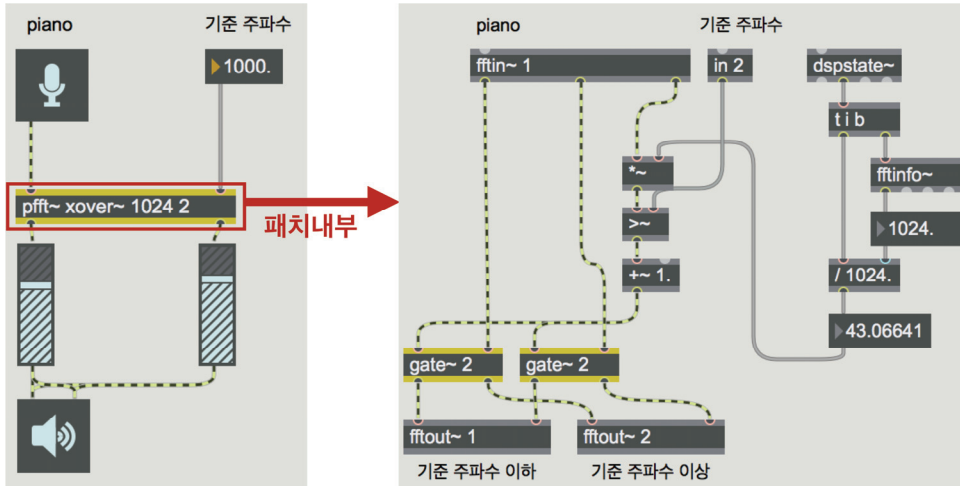
녹음된 사운드가 fftn~1로 들어옴 **패치내부** index~를 통해 재생속도에 지정해 준 속도로 buffer~에 녹음된 사운드를 읽는다.



[그림-5] FFT를 이용한 phase vocoder 패치

읽는다. 샘플에서 해당하는 위치의 숫자가 연속적으로 들어와야 재생이 된다. 여기서 보간은 디지털 사운드를 읽을 때 누락된 신호 요소를 임의로 채워서 읽는 것을 말한다. 16) sawtooth파형을 만들어내는 오브젝트이다. 왼쪽 인렛으로 들어가는 숫자가 1초 동안 0에서 1까지 몇 번이나 움직이느냐를 결정한다.

③ pfft~ xover~오브젝트의 활용



[그림-6] 사운드 주파수 대역을 나누는 pfft~ xover~ 패치

pfft~xover~는 FFT 분석으로 기준 주파수에 따라 사운드를 나누어 주는 역할을 한다. [그림-6]의 오른쪽에 있는 패치 내부를 살펴보면 피아노 사운드가 fftin~오브젝트(17)를 거쳐 FFT분석을 한 후에 >~오브젝트(18)로 들어가 지정해준 기준 주파수(여기선 1000)보다 낮으면 0이 나오고 +~오브젝트(19)를 거쳐 1이 되어 gate~오브젝트(20)의 첫 번째 아웃렛(outlet)을 열어 내보낸다. 반대로 지정해 준 기준 주파수보다 높은 소리가 들어오면 >~오브젝트에서 1이 나오고 +~를 거쳐 2가 되어 gate~의 두 번째 아웃렛을 연다.

17) pfft~오브젝트에 의해 로드된 패치에 신호입력을 제공한다.

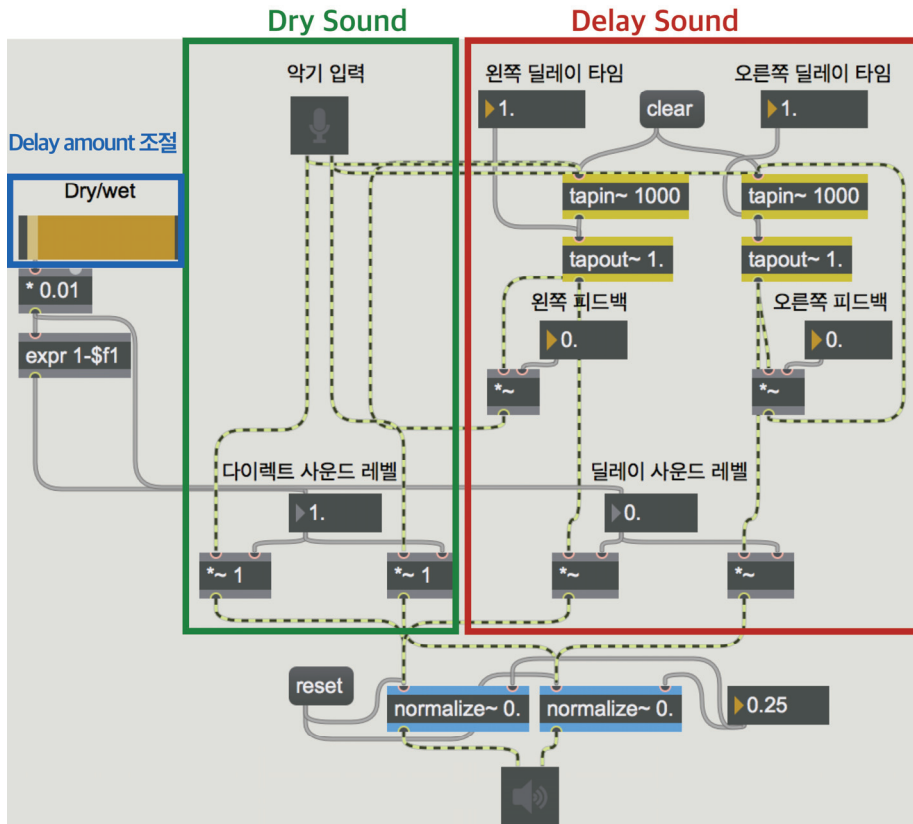
18) 왼쪽과 오른쪽 시그널의 크기를 비교하는 오브젝트. 왼쪽 시그널이 크면 1을 내보내고 그렇지 않으면 0을 내보낸다.

19) 입력된 두 개의 MSP신호를 더하는데 사용된다.

20) 오른쪽 인렛으로 들어오는 신호를 왼쪽에 들어오는 숫자에 따라 어느 아웃렛으로 내보낼 것인지를 결정할 수 있는 오브젝트이다. 0이 들어오면 모든 아웃렛이 닫히고 n이 들어오면 n번째 아웃렛이 열린다.

④ 딜레이 음향효과

딜레이 음향효과는 입력되는 사운드를 녹음하여 저장매체에 저장하고 일정시간이 지난 후 다시 재생 하는 것을 말한다. 컴퓨터의 경우 주로 RAM²¹⁾을 저장매체로 사용하며 딜레이 라인을 거쳐 반복되는 사운드의 횟수는 피드백의 양에 의해 결정된다.



[그림-7] 딜레이 음향효과를 위한 Max/MSP 패치

21) 랜덤 액세스 메모리(Random Access Memory), 임의의 영역에 접근하여 읽고 쓰기가 가능한 주기억 장치

위의 [그림-7]은 Max/MSP를 사용하여 딜레이 음향효과를 구현한 패치이다. tapin~오브젝트는 사운드를 딜레이 라인에 저장하는 역할을 한다. tapout~오브젝트를 사용하여 딜레이 라인에 저장된 사운드를 사용자가 지정한 딜레이 타임에 맞춰 재생하며 tapin~의 아웃렛은 반드시 tapout~의 인렛에 연결되어야 한다. tapout~과 연결되어 있는 *~오브젝트²²⁾에 0과 1사이의 소수를 넣어 피드백의 양을 조절 할 수 있다. 또한 딜레이 라인을 거치지 않는 다이렉트 사운드와 딜레이를 거쳐 일정시간 지연된 사운드와의 비율은 slider오브젝트²³⁾를 사용하여 조절 가능하다.

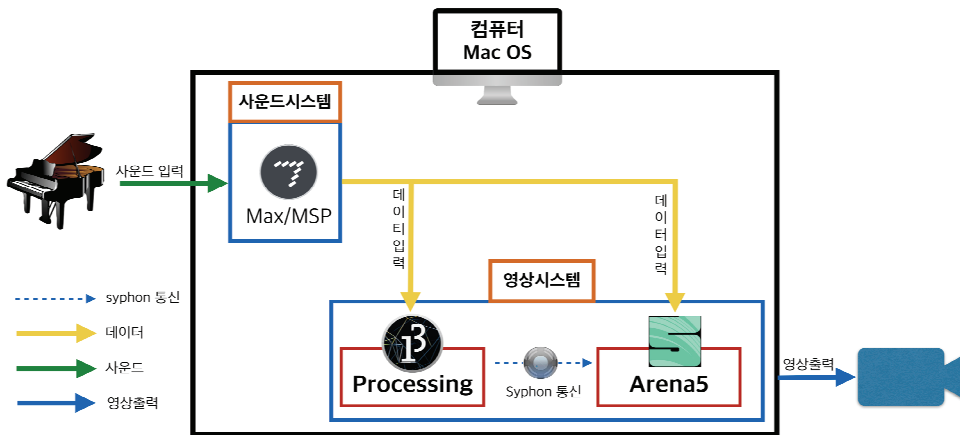
22) 두 개의 시그널을 곱해주는 오브젝트

23) 숫자 데이터를 출력하는 슬라이더 역할을 하는 오브젝트

2. 영상 시스템

1) 영상 시스템 구성

본 연구에서 사운드와 영상의 상호작용 요소로 음량 값을 선택하였다. 영상의 제작을 위하여 Processing²⁴⁾을 사용하였고 영상은 사운드 데이터에 의해 실시간으로 제어된다. 실시간으로 제작되는 영상은 Syphon²⁵⁾을 통해 Arena5²⁶⁾에서 메인 스크린으로 최종 출력된다. 영상 시스템 구성은 다음 [그림-8]과 같다.



[그림-8] 영상 시스템

24) MIT 미디어 랩의 'Aesthetics and Computation' 그룹에 의해 2001년 만들어진 오픈 소스 프로그래밍 언어

25) 응용 프로그램간의 영상 프레임을 서로 공유할 수 있게 해주는 Mac OS기반의 오픈소스

26) Resolume에서 개발한 영상 믹싱 프로그램으로 실시간 비주얼 퍼포먼스를 위해 주로 사용

피아노의 음량 값은 Max/MSP를 거쳐 Processing으로 전송되어 영상을 제어함과 동시에 Arena5의 내장된 영상효과를 실시간 제어한다.

① Processing을 이용한 영상 제작

영상 소스는 Processing을 이용하여 제작되었고 생성되는 영상 중 아래의 [그림-9]와 같이 나타나는 두 개의 이미지를 각각 Tower와 Tornado라고 명명하였다.



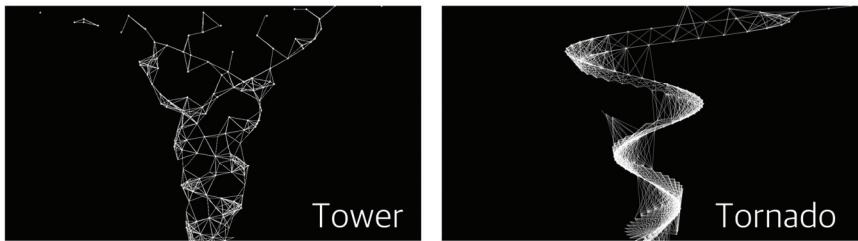
[그림-9] 프로세싱으로 생성된 영상

[그림-10]은 음량 값을 받아 제어되는 Processing 코드의 일부이다. 코드에 빨간색 표시된 부분인 oscx와 oscy는 각각 스테레오 마이크의 저음, 고음의 음량 값을 받으며 그 값에 따라 영상의 모양이 변화한다.

저음의 음량 값	고음의 음량 값
<pre>for(float i = 0; i+oscy < p1.length; i++){ p1[(int) i].loc.x = map(sin(theta1+i/5), -1, p1[(int) i].display(); for(int j = 0; j < p1.length; j++){ float d1 = dist(p1[(int) i].loc.x, p1[(int) float d2 = dist(p1[(int) i].loc.x, p1[(int)</pre>	<pre>for(float i = 0; i+oscx < p2.length; i++){ p2[(int) i].loc.x = map(cos(theta2+i/4), -1, 1, p2[(int) i].display(); for(int j = 0; j < p2.length; j++){ float d1 = dist(p2[(int) i].loc.x, p2[(int) if(d1 < lindist){</pre>

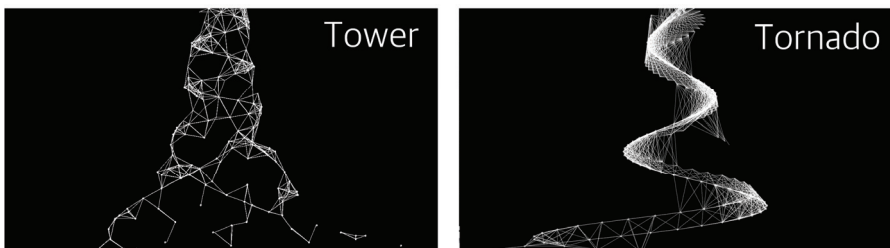
[그림-10] 음역에 따른 음량 값을 받는 Processing 코드의 일부

음량 값이 커질수록 영상이 화면 하단으로 가깝게 움직이는 모양을 나타내는데 고음과 저음의 음량 값인 oscx, oscy가 최고에 도달했을 때 영상은 [그림-11]과 같이 변화된다.



[그림-11] 입력 값이 가장 클 때 Processing 원본 영상

또한 불꽃이 타오르는 모습을 표현하기 위하여 Arena5의 내장 영상효과를 사용해 [그림-12]와 같이 영상을 상하반전 하였다.



[그림-12] 입력 값이 가장 클 때 상하반전 영상

② Arena5의 영상효과

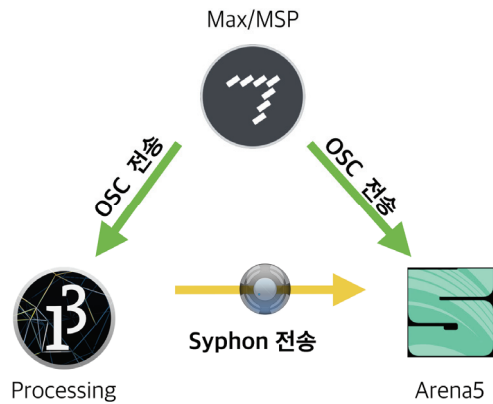
작품을 위해 사용한 Arena5의 내장효과는 다음 <표-3>과 같다. 사운드 시스템으로부터 입력받은 음량 값으로 각 영상효과의 특정 파라미터를 실시간으로 조절하여 음악과 영상의 상호작용을 부각시켰다.

<표-3> Arena5의 영상효과

영상효과 이름	나타나는 효과
Blur	특정 영역을 번지거나 흐리게 하는 효과
RadialBlur	Blur 효과 중의 하나로 아웃포커싱 효과를 줌
Bright.Contrast	밝기 및 명암 대비 조절 효과
Ripples	영상을 흔들리게 하는 효과
Iterate	영상이 회전되면서 움직이는 효과
Flip	상하반전 효과
FishEye	180도의 화각을 갖는 어안렌즈 효과
Sparkles	영상을 작은 조각으로 흩뿌리는 효과

3. 작품 구현을 위한 연동 프로그램

사운드와 영상을 연동하여 작품을 구현하기 위해서는 각각의 프로그램 간의 데이터 전송이 필요하다. Max 데이터의 전송은 OSC²⁷⁾를, Arena5로의 Processing 영상 전송은 Syphon을 사용하였다. [그림-13]은 OSC와 Syphon을 통한 데이터 통신의 흐름을 나타낸다.



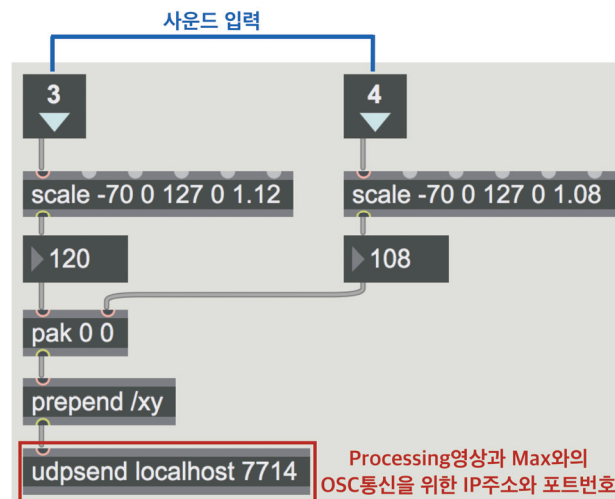
[그림-13] 데이터 통신의 흐름

27) OSC(Open Sound Control)는 뮤지컬 퍼포먼스나 쇼 컨트롤과 같은 목적으로 사운드 신디사이저, 컴퓨터 및 기타 멀티미디어 장치들을 네트워킹하기 위한 프로토콜이다. 컴퓨터 및 기타 멀티미디어 장치 간에 음악 연주 데이터(파라미터 및 노트 시퀀스 등)를 공유하기 위해 개발 되었다. OSC 메시지는 인터넷 및 로컬 서브넷 내에서 UDP / IP 및 이더넷을 사용하여 전송된다.

1) OSC를 활용한 사운드와 영상의 연동

① Max/MSP와 Processing의 연동

Max/MSP로 입력된 음량 값을 Processing으로 보내기 위해 udpsend오브젝트를 활용한다. udpsend오브젝트는 UDP²⁸⁾ 방식의 통신 프로토콜을 사용하는 오브젝트로서 IP주소와 포트²⁹⁾번호를 아규먼트(argument)로 지정해주어야 한다. 같은 컴퓨터 안에서 서로 다른 프로그램끼리 통신을 하려면 내 컴퓨터를 의미하는 127. 0. 0. 1 혹은 localhost로 IP주소를 지정하고 포트번호는 통신하려는 프로그램 간에 동일하게 설정한다.

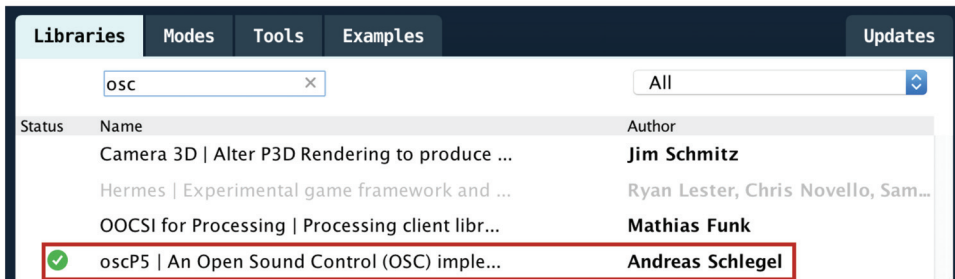


[그림-14] Processing으로 데이터를 전송하기 위한 Max/MSP 패치

28) User Datagram Protocol의 약어. 서로 교환하는 형식이 아닌 한 쪽에서 일방적으로 이뤄지는 방식의 통신 프로토콜이며 고속으로 데이터 처리가 가능하다.

29) 데이터의 입출력을 위한 통로

Processing에서 OSC통신을 통해 데이터를 받기 위해 [그림-15]와 같이 oscP5 라이브러리(library)를 설치한다.



[그림-15] oscP5 라이브러리를 설치하는 모습

아래 그림 [그림-16],[그림-17],[그림-18]은 OSC통신을 위한 Processing 코드이다. 3가지 코드를 역할에 맞게 각각 전체 코드의 해당위치에 삽입한다. [그림-16]은 OSC통신을 수행할 수 있는 oscP5 라이브러리를 불러오는 코드이다.

```
import oscP5.*;
import netP5.*;
OscP5 oscP5;
NetAddress myRemoteLocation;
```

[그림-16] OSC통신을 위한 라이브러리를 불러오는 코드

[그림-17]은 Max/MSP와 Processing과의 OSC통신을 위한 주소를 설정하는 코드이다.

```
void setup(){  
  
    oscP5 = new OscP5(this,7714);  
    Max/MSP에서 데이터를 받기 위해 IP주소와 포트번호를 같게 설정한다.  
    myRemoteLocation = new NetAddress("127.0.0.1",7714);  
}
```

[그림-17] OSC 통신을 위한 주소 설정 코드

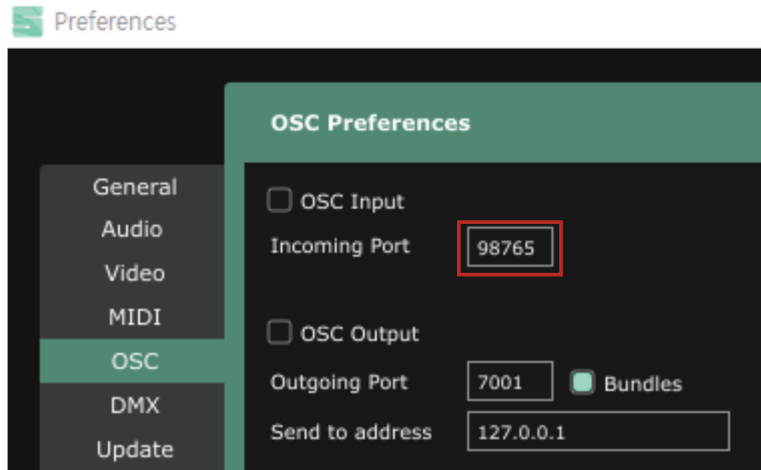
피아노의 음량 값을 사용해 Processing 영상을 제어하려면 입력되는 음량 값을 Processing에서 처리할 수 있는 변수로 설정해야 한다. [그림-18]은 OSC 통신이 가능하도록 입력받는 데이터를 변수로 설정하는 코드이다.

```
void oscEvent(OscMessage theOscMessage) {  
    if(theOscMessage.checkAddrPattern("/xy")==true) {  
  
        int firstValue = theOscMessage.get(0).intValue();  
        int secondValue = theOscMessage.get(1).intValue();  
  
        oscmousexy(firstValue,secondValue);  
    }  
}  
void oscmousexy(int ox,int oy)  
{  
    oscx = ox;  
    oscy = oy;  
}
```

[그림-18] OSC 통신이 가능하도록 입력 데이터를 변수로 설정하는 코드

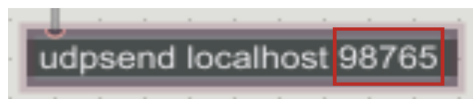
② Max/MSP와 Arena5의 연동

Max/MSP를 통한 OSC통신을 사용하여 피아노의 음량 값으로 Arena의 영상효과를 제어할 수 있다. Arena5에서 데이터를 전송받기 위해서는 데이터의 통로인 포트번호를 일치시켜야 하는데 Arena5 > Preference > OSC 의 차례로 경로를 들어가면 [그림-19]와 같이 Arena5의 포트번호를 확인 할 수 있다.



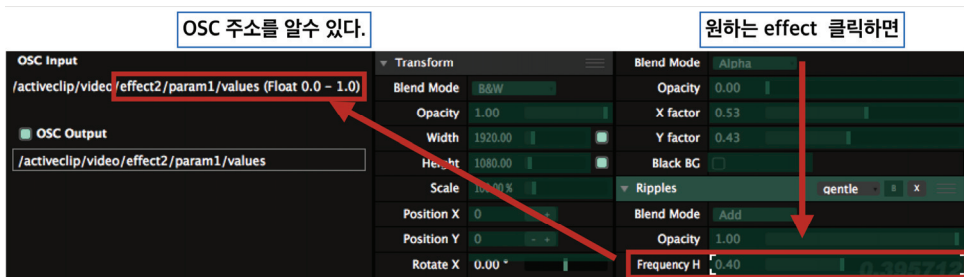
[그림-19] Arena5의 포트번호 확인

Max/MSP에서도 마찬가지로 신호를 보내기 위해 아래의 [그림-20]과 같이 포트 번호를 맞춘다.



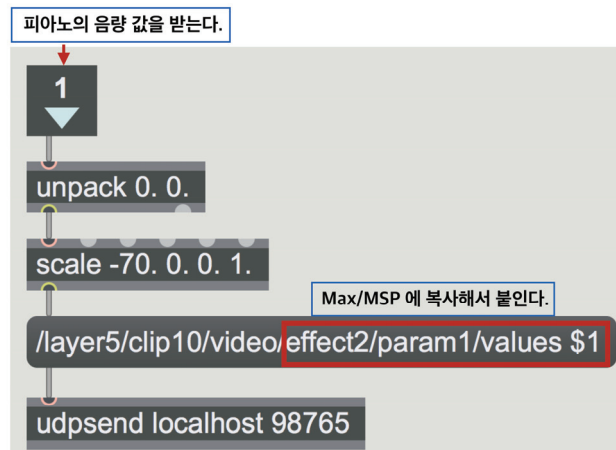
[그림-20] Max/MSP에서의 포트번호

Arena5에서는 Blur, RadialBlur, Bright.Contrast 등의 효과를 사용하였다. Arena5 > Mapping > Edit application OSC map 의 차례로 경로를 들어간 후 상호작용을 하고자 하는 영상효과를 클릭하면 그에 해당하는 주소를 알 수 있다. 주소를 보기 위한 방법을 [그림-21]과 같이 정리하였다.



[그림-21] Arena5에서 각각의 영상효과들의 OSC주소를 확인하는 방법

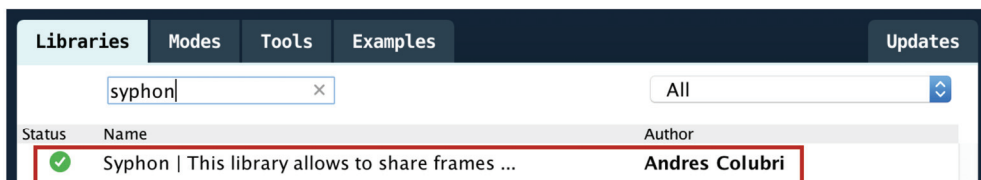
영상효과들의 OSC주소를 [그림-22]와 같이 Max/MSP 패치에 넣어서 음량 값을 받을 수 있게 하였다.



[그림-22] Max/MSP에서 Arena5로 OSC 데이터 통신하기 위한 패치

2) Syphon을 활용한 영상 출력

Processing 영상을 Arena5를 통해 출력하기 위해 Syphon을 사용해야 한다. Syphon을 사용하기 위해서는 [그림-23]과 같이 미리 Syphon 라이브러리를 설치한다.



[그림-23] Syphon 라이브러리 설치 모습

Syphon 코드도 OSC통신과 마찬가지로 3가지 코드를 역할에 맞게 각각 해당하는 위치에 삽입한다. [그림-24]와 같이 미리 설치된 Syphon 라이브러리를 호출하는 코드를 입력한다.

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

import codeanticode.syphon.*;

SyphonServer server;
```

[그림-24] 라이브러리를 호출하는 코드

[그림-25]와 같이 서버를 생성하는 코드를 입력한다.

```
void setup(){  
    oscP5 = new OscP5(this,7713);  
    myRemoteLocation = new NetAddress("127.0.0.1",7713);  
    server = new SyphonServer(this, "Processing Syphon");
```

[그림-25] 서버 생성 코드

[그림-26]과 같이 Server를 스크린으로 보내는 코드를 입력한다.

```
    theta1+=.01;  
    theta2+=.01;  
    server.sendScreen();
```

[그림-26] 스크린으로 영상을 보내는 코드

Max/MSP에서 음량 데이터를 받아 실시간 제어되는 Processing영상을 Syphon코드를 통해 Arena5에서 불러와 출력이 가능하게 된다.

4. 작품 설명

1) 작품 소개

<Midnight Peace>는 밤을 평온하게 하지 못하는 도시의 소음에 갇힌 이 세상의 밤을 고요하고 평화롭게 내버려 두고 싶다는 소망이 담긴 작품이다. 실시간으로 사운드와 영상을 제어하는 멀티미디어 작품으로 Max/MSP를 이용하여 사운드 효과를 제작하였으며 피아노 연주에 따라 달라지는 데이터를 Processing에 적용하여 실시간으로 영상을 제어하였다. 연구한 시스템은 2017년 11월 11일 동국대학교 이해랑 예술극장에서 진행된 멀티미디어음악 공연 ‘SEEING SOUND LISTENING IMAGE(보는 소리, 듣는 영상) XIV’ 중 멀티미디어음악 작품 <Midnight Peace>에 적용하였다.



[그림-27] <Midnight Peace> 공연 장면

2) 작품 구성

작품<Midnight Peace>는 크게 A-B-A'로 이루어진 3부 형식이다. 음악의 구성에서 가장 기본적인 형식으로 대조되는 부분들 사이에 짧은 연결구들이 존재한다. 본 작품에서 연결구를 포함한 상세한 음악 형식은 A, 연결구-1, B, 연결구-2, B', bridge, A'이다.

A 파트에서는 느리게 시작되는 피아노의 오리지널 사운드와 함께 pfft~ xover~, phase vocoder, 딜레이 음향효과의 순서로 연결된 프로세싱 사운드가 패드처럼 연주된다. 이때 영상은 고요한 밤을 묘사하는 영상이 나온다. 빠른 부분인 B 파트로 넘어가는 연결구에서는 고조시키기 위하여 granular synthesis로 프로세싱 된 소리가 양쪽으로 빠른 속도로 오토패닝(auto panning)되어 연주한다. B 파트에서는 granular synthesis로 프로세싱 된 소리에 따라 평화의 불꽃이 타오르는 것을 Arena5의 Blur 이펙트로 묘사하였다. B' 파트로 넘어가는 연결구에서는 클라이맥스로 도달하기 위하여 granular synthesis로 2옥타브 아래 소리를 내어 베이스가 매우 두터운 소리를 내게 하였으며 이에 대한 영상효과로 Arena5의 Ripple이펙트를 사용하였다. B' 파트는 이 작품의 클라이맥스로 granular synthesis를 사용하여 프로세싱 사운드 음정을 2옥타브 아래에서 1옥타브 위로 변화시켰다. 이에 대한 영상 또한 지금까지 사용된 영상을 모두 조합하여 스크린 전체를 사용하였으며 조명 또한 밝아진다. bridge 파트에서는 모호한 사운드를 만들기 위하여 50~80ms의 딜레이 음향효과를 사용하였다. 또한 granular synthesis를 이용하여 도약 및 하강이 반복되는 사운드를 만들고 그에 대한 영상으로 빛 속에서 눈이 내리는 영상을 만들어

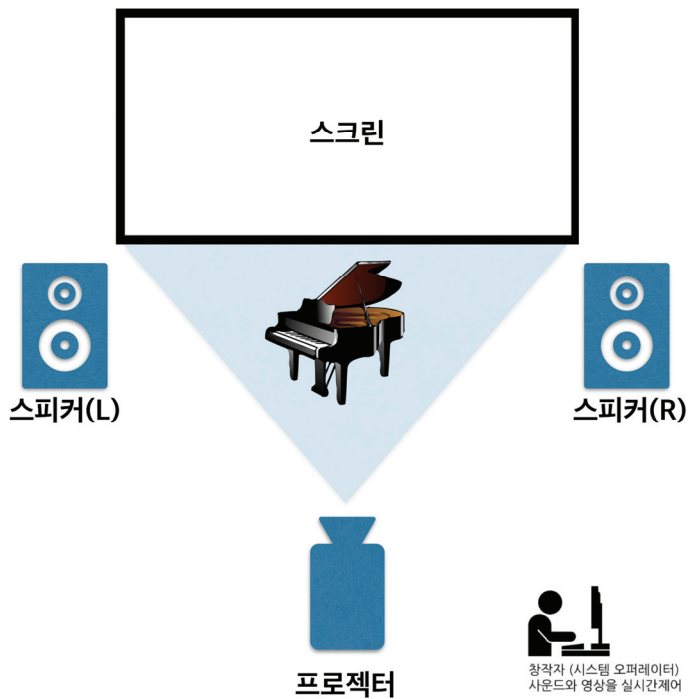
상호작용 효과를 주었다. 마지막인 A' 파트는 처음과 같은 음악과 프로세싱 사운드를 사용하였고 영상은 평화의 불꽃을 의미하는 영상을 사용하였다.

<표-4> 작품 전체 구성

부분	시간	음향효과	영상	영상 및 영상효과	영상의미
A 파트	00:00 ~ 1:18	phase vocoder, 딜레이 음향효과	O	Processing, Arena5 (Blur, RadialBlur, Bright.Contrast)	고요한 밤
연결구-1	1:18 ~ 1:27	granular synthesis, 딜레이 음향효과	X		
B 파트	1:27 ~ 2:10	granular synthesis, 딜레이 음향효과	O	Processing, Arena5(Blur, RadialBlur, Bright.Contrast)	평화의 불꽃
연결구-2	2:10 ~ 2:25	granular synthesis, 딜레이 음향효과	O	Processing, Arena5(Ripples)	평화가 깨짐
B' 파트 (climax)	2:25 ~ 3:04	granular synthesis, 딜레이 음향효과	O	Processing, Arena5(Iterate, Ripples, Flip, FishEye)	평화를 열망함
bridge 파트	3:04 ~ 4:05	granular synthesis, 딜레이 음향효과	O	Processing, Arena5 (Sparkles, Blur)	다시 찾아온 평화
A' 파트	4:05 ~ 5:05	phase vocoder, 딜레이 음향효과	O	Processing, Arena5(Blur, RadialBlur, Bright.Contrast)	평화의 불꽃

3) 무대 구성

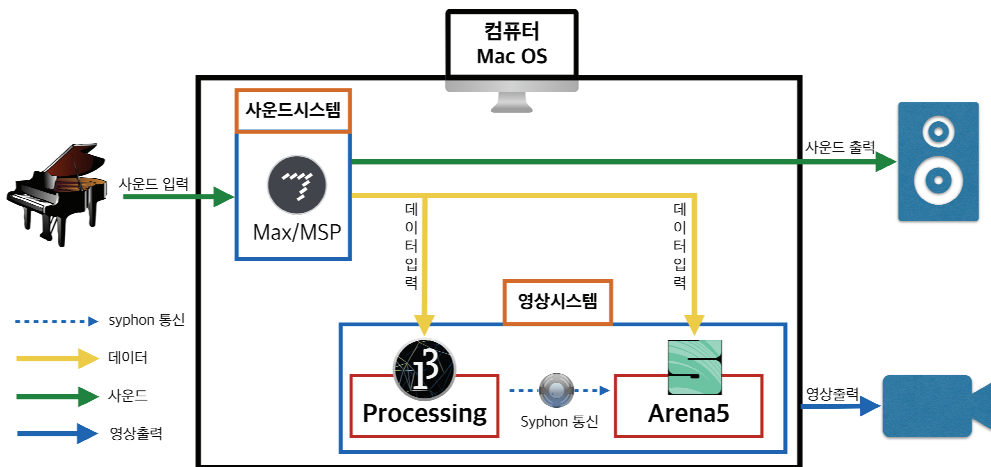
[그림-28]은 작품의 무대 구성이다. 피아노는 무대 정중앙에 위치하고 피아노와 연주자에게만 조명이 비춰지도록 하였다. 스크린 전체에 영상을 투사하였으며 오퍼레이터(operator)는 객석의 오른쪽에 위치해 무대를 모니터하며 시스템을 제어한다.



[그림-28] 무대 구성

4) 시스템 구성

[그림-29]는 작품에 적용된 시스템 구성이다. 피아노 사운드는 마이크를 통해 Max/MSP로 입력되고 각각의 음향효과로 변형되어 스피커로 출력된다. 사운드 시스템에서 추출한 음량 값은 영상 시스템을 제어하는 데이터로 치환되어 사운드와의 상호작용을 일으킨다.



[그림-29] 시스템 구성

5. 연구기술의 작품 적용

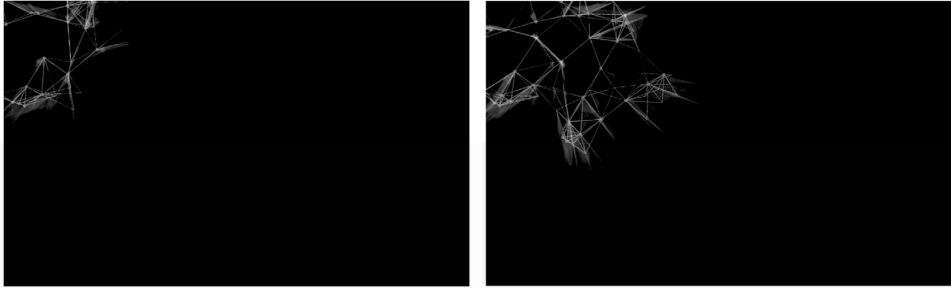
1) A 파트 적용 효과

A 파트는 피아노 연주로 시작된다. 35초부터 pfft~ xover~를 사용하여 1000Hz 이하의 사운드를 받아 phase vocoder로 입력되게 하였으며 playback rate 파라미터를 -0.5로 두었다. 또한 왼쪽과 오른쪽에 각각 500ms, 700ms의 딜레이 효과를 걸어주어 페드악기와 같은 효과를 낸다. 사용된 영상효과는 Blur, RadialBlur, Bright.Contrast이며 정리하면 <표-5>와 같다.

<표-5> A 파트에 적용된 사운드 파라미터 값 및 영상효과

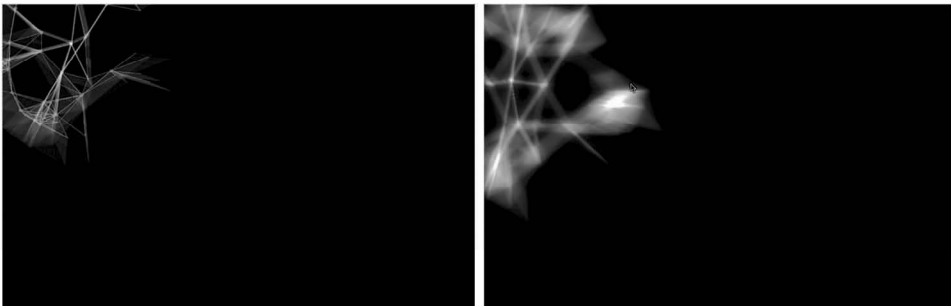
A(00:00~1:18)		
사운드 파라미터 값		영상효과
pfft~ xover~	기준주파수=1000Hz	Blur, RadialBlur, Bright.Contrast
phase vocoder	playback rate=-0.5	
딜레이 음향효과	왼쪽 딜레이 타임=500ms 오른쪽 딜레이 타임=700ms	

영상은 사운드의 음량 값과 연동되어 [그림-30]과 같이 값이 커질수록 빛의 크기 또한 비례하게 하였다.



[그림-30] 음량 값에 따른 영상 변화

피아노 사운드만 나올 경우에는 적용되는 영상효과가 없으며 phase vocoder의 프로세싱 사운드가 재생되면서 장면 전환을 통해 [그림-31]과 같이 Blur 효과를 적용하였다.



[그림-31] Blur 효과가 적용된 영상

2) 연결구-1 적용 효과

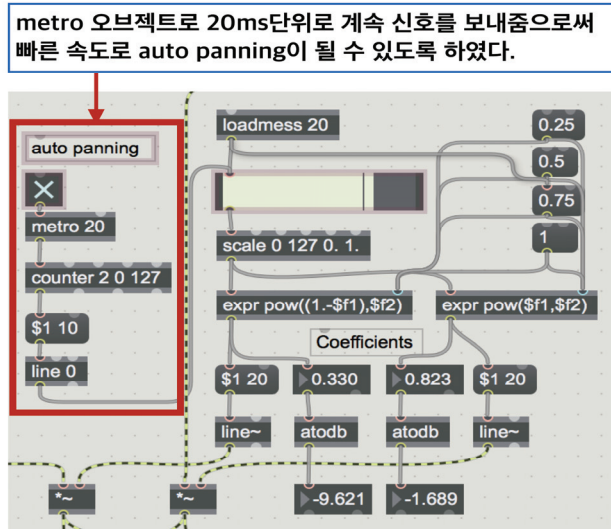
B 파트로 넘어가기 위한 연결구-1에서는 영상을 사용하지 않았다. B부분으로 넘어가는 부분에서 저음을 강조하기 위해 granular synthesis의 grain pitch를 0.5로 지정하여 프로세싱 사운드를 한 옥타브

낮추었다. 적용된 음향효과와 각 파라미터 값은 <표-6>과 같다.

<표-6> 연결구-1에 적용된 사운드 파라미터 값 및 영상효과

연결구-1(1:18~1:27)		영상효과
사운드 파라미터 값		영상효과
granular synthesis	grain size=300ms grain size variation=10ms grain pitch=0.5	영상 없음
딜레이 음향효과	왼쪽 딜레이 타임=700ms 오른쪽 딜레이 타임=500ms	

위에서 언급한 프로세싱을 거친 사운드는 [그림-32]의 패치를 통해 오토패닝을 이용하여 좌우로 빠르게 확산되게 하여 분위기를 고조시켰다.



[그림-32] granular synthesis에 연결된 panning 패치

3) B 파트 적용 효과

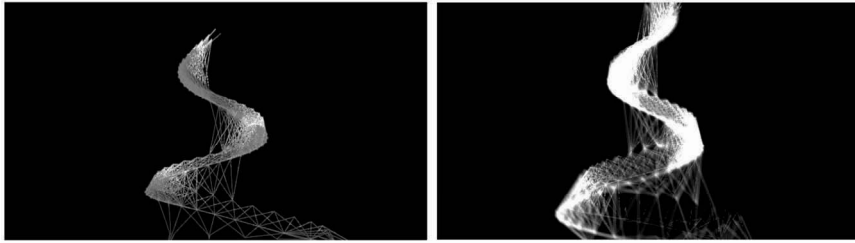
B 파트는 A 파트와 대조적인 부분으로 빠른 템포(tempo)와 더불어 아르페지오(arpeggio)³⁰⁾음형을 통해 작품을 전개한다. 사용된 사운드 프로세싱은 granular synthesis와 딜레이 음향효과이다. granular synthesis의 grain size는 500, pitch는 1로 지정하여 오리지널 사운드에서 음정의 변화 없이 피아노 연주와 어우러지도록 했다. 또한 왼쪽과 오른쪽으로 딜레이 타임은 각각 700ms, 500ms로 짧은 단위가 반복되도록 하였다. B 파트에서 사용된 사운드 프로세싱의 주요 파라미터에 대한 지정 값과 사운드와 연동된 영상 효과는 아래 <표-7>과 같다.

<표-7> B 파트에 적용된 사운드 파라미터 값 및 영상효과

B(1:27~2:10)		
사운드 파라미터 값		영상효과
granular synthesis	grain size=300ms grain size variation=10ms grain pitch=1	Blur, RadialBlur, Bright.Contrast
딜레이 음향효과	왼쪽 딜레이 타임=700ms 오른쪽 딜레이 타임=500ms	

위의 <표-7>에서 볼 수 있듯이 이 부분에 적용된 영상 효과로 Arena5의 Blur 효과를 사용한다. 이 효과는 사운드 프로세싱 값으로 제어되며 [그림-33]과 같이 프로세싱 된 사운드의 음량 값이 커질수록 Blur효과도 진하게 표현된다.

30) 화음이 분산된 음형으로 각 음을 동시에 연주하는 것이 아니라 순차적으로 연주하는 것이다.



[그림-33] 사운드 프로세싱 값에 따른 Blur효과 적용

4) 연결구-2 적용 효과

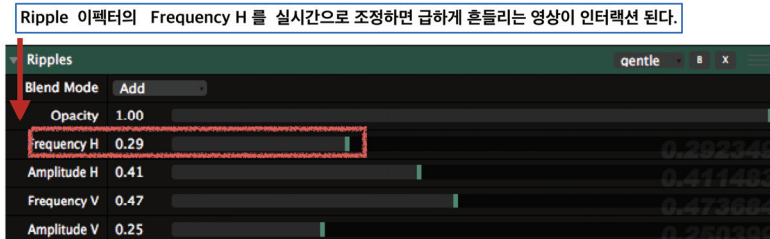
연결구-2에서 grain pitch는 0.25로 조절하여 프로세싱 사운드를 2옥타브 아래에서 생성되게 한 후 딜레이 효과를 주어 클라이맥스로 전개되는 것을 암시하였다. 사운드 음향효과와 파라미터 값은 <표-8>과 같이 정리하였다.

<표-8> 연결구-2에 적용된 사운드 파라미터 값 및 영상효과

연결구-2(2:10~2:25)		
사운드 파라미터 값		영상효과
granular synthesis	grain size=300ms grain size variation=10ms grain pitch=0.25	Ripples
딜레이 음향효과	왼쪽 딜레이 타임=700ms 오른쪽 딜레이 타임=500ms	

[그림-34]와 같이 피아노의 음량 값을 Ripple 영상효과와 Frequency H 파라미터에 적용하여 이 파라미터의 값이 증가 할수록 영상이 흔들리는 속도가 빨라지게 하였다. 이를 통해 불꽃이 흔들리는 영상을 연출하고

평화가 깨지는 모습을 표현하였다.



[그림-34] 음량 값으로 Arena5의 Ripple 영상효과 제어

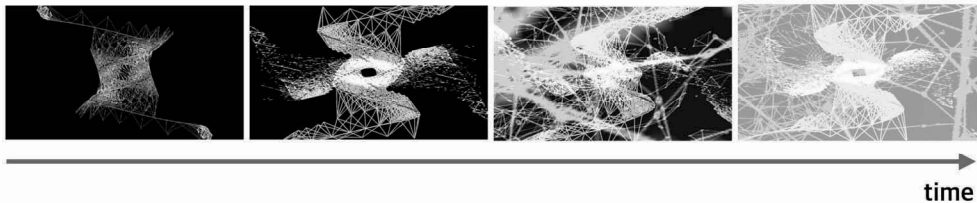
5) B' 파트 적용 효과

B' 파트는 이 작품의 클라이맥스이다. 평화를 열망하는 모습을 표현하였다. granular synthesis의 grain pitch를 0.25에서 2까지 변화시킴으로 피치를 상승시켜 긴장을 극대화시켰다. 사용된 음향효과와 파라미터 값과 영상효과는 <표-9>와 같다.

<표-9> B' 파트에 적용된 사운드 파라미터 값 및 영상효과

B' 파트(2:25~3:04)		
사운드 파라미터 값		영상효과
granular synthesis	grain size=300ms grain size variation=10ms grain pitch=0.25~2	Iterate, Ripples, Flip, FishEye
딜레이 음향효과	왼쪽 딜레이 타임=700ms 오른쪽 딜레이 타임=500ms	

B' 파트에 사용된 영상은 메인영상의 방향을 바꾸어 네 개가 순차적으로 겹치게 나오도록 하였고 Arena5의 영상효과 Iterate를 통해 배경의 변화를 주었다. 영상의 변화와 더불어 opacity³¹⁾의 파라미터 값을 변화시켜 영상이 차차 밝아지게 함으로 긴장감을 주었다. 이 영상으로 평화를 열망하는 모습을 표현하였다. 클라이맥스 부분의 영상이 변화하는 순서는 [그림-35]와 같다.



[그림-35] 클라이맥스 부분의 영상변화

6) bridge 파트 적용 효과

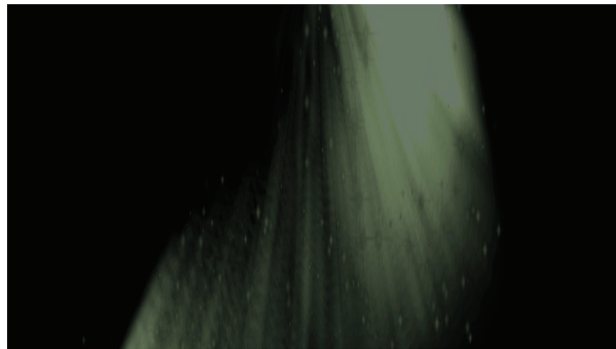
bridge 파트에서는 음량 값에 따라 grain pitch variation을 70에서 90까지 움직이게 하여 입력되는 음량 값이 커질수록 프로세싱 된 사운드 음정의 변화 값도 커지도록 하였다. 다음의 <표-10>은 bridge 파트에 적용된 사운드와 영상효과이다.

31) 영상 프로그램에서 소스의 불투명도를 의미하며 Arena5에서는 0-1사이의 값으로 제어된다.

<표-10> bridge 파트에 적용된 사운드 파라미터 값 및 영상효과

bridge 파트(3:04~4:05)		
사운드 파라미터 값		영상효과
granular synthesis	grain size=300ms grain pitch variation=70~90	Sparkles, Blur
딜레이 음향효과	왼쪽 딜레이 타임=700ms 오른쪽 딜레이 타임=500ms	

bridge 파트에서 사운드와 영상의 연동은 grain pitch variation에 따라 Arena5의 sparkle효과의 amount 값이 달라지게 하였으며 [그림-36]과 같이 적용되었다. 즉, 입력되는 피아노 사운드의 음량 값은 grain pitch variation 파라미터 값을 조절하고 grain pitch variation 파라미터 값은 알갱이로 표현되고 있는 영상의 amount 값을 조절하는 것으로 서로 연결되어있다.



[그림-36] grain pitch variation이 적용된 영상

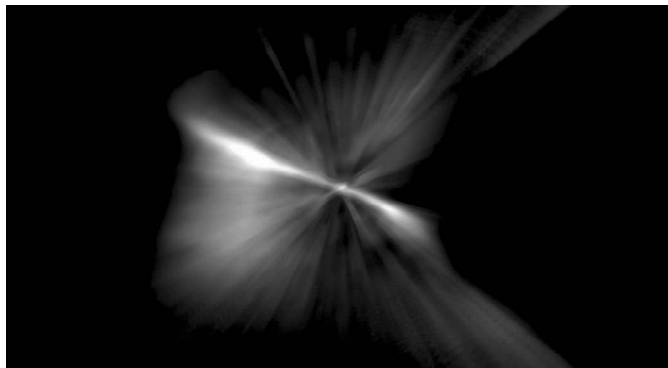
7) A' 파트 적용 효과

사운드 프로세싱은 A 파트와 같이 phase vocoder와 딜레이 음향효과를 사용하였다. 각각의 파라미터 값 또한 A 파트와 같고 사용된 영상효과는 Blur, RadialBlur, Bright.Contrast이다. A' 파트에서 사용된 사운드 프로세싱의 주요 파라미터에 대한 지정 값과 영상 효과는 아래 <표-11>과 같다.

<표-11> A' 파트에 적용된 사운드 파라미터 값 및 영상효과

A' 파트(4:05~5:05)		
사운드 파라미터 값		영상효과
pfft~ xover~	기준주파수=1000Hz	Blur, RadialBlur, Bright.Contrast
phase vocoder	playback rate=-0.5	
딜레이 음향효과	왼쪽 딜레이 타임=500ms 오른쪽 딜레이 타임=700ms	

위의 영상효과로 나타난 영상은 [그림-37]과 같고 불꽃이 환하게 비춤으로써 평화로운 밤을 묘사하였다.



[그림-37] A' 파트 영상

Ⅲ. 결론 및 고찰

본 연구를 통해 피아노 음악에 다른 미디어를 결합하여 새로운 종류의 멀티미디어 작품을 만들고자 하였다. 또한 음악과 영상이라는 매체가 갖는 상호작용 요소들을 조합하여 얼마나 효과적이면서도 예술적으로 관객들에게 전달되는가에 관한 연구이기도 하였다. 이를 위해 피아노에서 추출 할 수 있는 음량 값을 활용하여 실시간으로 음악과 영상을 제어 하였다.

사운드 시스템은 pfft~ xover~오브젝트와 phase vocoder, 딜레이 음향효과가 패드로 활용된다. 또한 granular synthesis로 여러 개의 악기가 동시에 연주하는 협주곡³²⁾과 같은 효과를 주었다. 영상은 Processing을 이용하여 메인테마 영상을 만들고 음량 값을 상호작용 요소로 활용하여 실시간으로 반응하게 하였다. Processing 영상의 변화와 더불어 Arena5의 영상효과를 활용하여 상호작용을 부각시켰다. 이렇게 만들어진 영상들은 음악에 직관적으로 반응함을 확인하였다.

이번 연구를 하면서, 몇 가지 문제점 중 첫 번째는 시스템 리소스³³⁾를 비교적 많이 사용하는 Processing 영상의 경우 사용하는 개수에 제한을 받는다는 것이다. Processing에 대한 깊은 연구와 이론적 고찰로 코드를 간소화시켜 활용하여야 할 것이다. 두 번째로는 피아노와 같은 화성악기에서는 하나의 음정(pitch)만 추적하는 것이 어렵다는 것이다. 음량 값 외에 다른 것을 부가적인 상호작용 요소로 활용한다면 다채로운 작품을 제작할 수 있을 것이다.

32) 관현악 합주와 피아노 솔로로 구성된 작품

33) 메모리, 하드디스크의 가상공간, CPU의 여유 등 윈도우가 사용할 수 있는 여유 자원

본 연구는 연주자와 시스템 오퍼레이터의 2인 체제를 바탕으로 진행되었다. 손을 제외한 다른 신체 부위를 활용한 모션 디텍팅(motion detecting)³⁴으로 오퍼레이팅을 할 수 있다면 연주자가 오퍼레이터의 역할까지도 겸하여 할 수 있을 것이다. 이렇게 1인 예술체제를 가능하게 하는 것이 연구해야 할 향후 과제이다. 또한 무한한 시청각적 재료와 과학기술을 융합하여 예술에 직관적으로 풀어내는 일이야말로 모든 아티스트들이 지속적으로 도전해야 할 연구 과제임은 분명하다.

Keyword(검색어)

인터랙티브 멀티미디어 음악(interactive multimedia music),
프로세싱(Processing), Max/MSP, 피아노음악(piano music),
소리시각화(sound visualization), 컴퓨터음악(computer music)

E-mail: immanuelhan@naver.com

34) 움직이는 물체를 감지하는 것

참 고 문 헌

1. 단행본

- Daniel Shiffman, 「러닝 프로세싱」, 비제이퍼블릭, 2016.
- Leon Harkleroad, 「The Math Behind the Music」, Cambridge University press, 2006
- Bob Katz, 「Mastering Audio the art and the science - third edition」, Focal Press, 2007.
- V.J.Manzo, 「Max/MSP/Jitter for music」, Oxford University Press, 2011.
- Curtis Roads, 「The Computer Music Tutorial」, MIT Press, 1996.

2. 참고논문

- 강현우, 「인도음악 연구를 통한 인터랙티브 멀티미디어음악 제작 연구」 (동국대학교 영상대학원 멀티미디어학과, 2017)
- 나준하, 「Max/MSP/Jitter를 이용한 기타 이펙터 제작과 실시간 소리 시각화 연구」 (동국대학교 영상대학원 멀티미디어학과, 2015)

- 이승지, 「북소리의 실시간 프로세싱에 의한 인터랙티브 멀티미디어음악 제작 연구」 (동국대학교 영상대학원 멀티미디어학과, 2017)
- 전우진, 「컴퓨터음악과 phase music을 이용한 인터랙티브 멀티미디어 퍼포먼스 연구」 (동국대학교 영상대학원 멀티미디어학과, 2017)

3. 웹사이트

- Max/MSP
<https://cyclring74.com/>
- CNMAT: external Max objects, OSC
<http://cnmat.berkeley.edu>
- Processing
<https://processing.org/>
- Open Processing
<https://www.openprocessing.org/>
- Learning Processing
<http://learningprocessing.com/>
- Syphon
<http://syphon.v002.info/>

ABSTRACT

A Study on Interactive Multimedia Music using Real-time Processing for Piano (focus on Multimedia Music <Midnight Peace>)

Han, Seung Wook

A multimedia work means that two or more different media are merged to create a new kind of artistic effect. In this study, visualization effects interact with piano sound.

Real-time sound processing is performed through the computer so that the computer can act as another performer and create music along with the piano sound. Also, sound visualization that synchronizes the video interacting with sound output in real time is applied to this work. To do this, music and video were controlled in real time using the volume value that can be extracted from the piano sound.

The sound system utilizes Max/MSP to create phase vocoder, delay sound effect, and granular synthesis. Phase vocoder and delay sound effect were used as a pad instrument, and granular synthesis was used to produce the similar effect as a concerto in which several instruments were played simultaneously.

The main theme image is generated using Processing while the volume value is used as an interactive element that reacts in real time. In addition, the effect of `Arena5` functions as another interaction element that fills in the insufficient parts during the transitions of images generated by the Processing. It was observed that these images intuitively respond to music and also make music more dramatic.

This study is based on the two-person system which consists of a performer and a system operator. It may also be operated by just one performer if the system is dramatically simplified and motion detection is available. Making such one-person system feasible is the future challenge. It's also claimed that all artists must continually challenge to integrate infinite audiovisual materials and advanced technologies into an intuitive art.

부록-1 : 작품 <Midnight Peace> 연주 악보

Midnight Peace

The first system of the musical score for 'Midnight Peace' consists of two staves. The upper staff is in treble clef and the lower staff is in bass clef. The key signature is one sharp (F#) and the time signature is 4/4. The melody in the upper staff begins with a quarter note G4, followed by quarter notes A4, B4, and C5. The lower staff is mostly empty, with a few notes appearing later in the piece.

The second system of the musical score continues from the first. The upper staff has a melody with quarter notes and eighth notes. The lower staff provides harmonic support with chords and single notes. The system ends with a measure containing a whole rest in the upper staff and a chord in the lower staff.

The third system of the musical score shows the continuation of the melody and accompaniment. The upper staff features a melodic line with some grace notes and slurs. The lower staff continues with a steady accompaniment pattern.

The fourth system of the musical score introduces a more complex texture. The upper staff has a rapid sixteenth-note passage with a '6' fingering and an '8va' marking. The lower staff has a rhythmic accompaniment with chords and single notes.

The fifth system of the musical score continues the rapid sixteenth-note passage in the upper staff. The lower staff accompaniment remains consistent with the previous system.

The sixth system of the musical score concludes the piece with the final measures of the sixteenth-note passage in the upper staff and the accompaniment in the lower staff.

28 Musical notation for measures 28-29. Measure 28 features a complex sixteenth-note melody in the right hand with sixteenth-note accompaniment in the left hand. Measure 29 continues the melody with a fermata over the final note.

30 Musical notation for measures 30-31. Measure 30 has a more rhythmic melody in the right hand. Measure 31 shows a change in the right-hand melody.

33 Musical notation for measures 33-36. Measures 33-34 feature a dense, fast-moving sixteenth-note texture in the right hand. Measures 35-36 are characterized by sustained chords in the right hand and a steady bass line in the left hand.

37 Musical notation for measures 37-40. Measures 37-39 continue the dense sixteenth-note texture in the right hand. Measure 40 shows a transition to a more melodic line in the right hand.

41 Musical notation for measures 41-44. Measures 41-43 maintain the sixteenth-note texture. Measure 44 features a melodic phrase in the right hand with a fermata.

45 Musical notation for measures 45-48. Measures 45-46 show a melodic line in the right hand. Measures 47-48 consist of sustained chords in the right hand and a steady bass line in the left hand.

49

Musical notation for measures 49-52. The right hand has a melodic line with eighth and quarter notes, while the left hand provides a harmonic accompaniment with chords and single notes.

53

Musical notation for measures 53-56. The right hand features a continuous eighth-note pattern, and the left hand has a steady accompaniment of chords.

57

Musical notation for measures 57-60. The right hand continues with eighth-note patterns, and the left hand has a consistent accompaniment.

61

Musical notation for measures 61-64. The right hand has a melodic line with some rests, and the left hand has a harmonic accompaniment.

65

Musical notation for measures 65-68. The right hand has a melodic line with eighth notes, and the left hand has a harmonic accompaniment.

69

Musical notation for measures 69-72. The right hand has a melodic line with a long note, and the left hand has a harmonic accompaniment with some ties.

부록-2 : 첨부 DVD 설명

1. Midnight Peace.mov : 2017년 11월 11일 이해랑 예술극장 <Midnight Peace> 공연 실황
2. Midnight Peace : 작품 악보 폴더
3. 작품<Midnight Peace>에 사용된 Max/MSP 패치