



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

베이스 트롬본과 피아노의 실시간 사운드  
프로세싱을 이용한 멀티미디어작품 제작 연구  
(멀티미디어음악 작품 <Yes, I am.>을 중심으로)

지도교수 김 준

동국대학교 영상대학원

멀티미디어학과 컴퓨터음악전공

조 환 희

2 0 1 9

석사학위논문

베이스 트롬본과 피아노의 실시간 사운드  
프로세싱을 이용한 멀티미디어작품 제작 연구  
(멀티미디어음악 작품 <Yes, I am.>을 중심으로)

조환희

지도교수 김 준

이 논문을 석사학위 논문으로 제출함

2018년 12월

조환희의 음악석사(컴퓨터음악) 학위 논문을 인준함

2019년 1월

위원장 박 상 훈



위원 김 정 호



위원 김 준



동국대학교 영상대학원

## 목 차

I. 서 론 .....	1
1. 연구 배경 및 목적 .....	1
2. 선행 연구 .....	3
II. 기술 연구 .....	5
1. 사운드 시스템 연구 .....	5
1) 사운드 시스템 .....	5
2) 음향효과 연구 과정 .....	6
① granular synthesis 음향효과 .....	7
② comb filter 음향효과 .....	10
③ delay 음향효과 .....	11
④ flanger 음향효과 .....	13
⑤ ring modulation 음향효과 .....	14
⑥ FFT분석을 이용한 phase vocoder 음향효과 .....	15
⑦ FFT분석을 이용한 pfft~ xover~오브젝트 활용 .....	18
2. 영상 시스템 연구 .....	20
1) 영상 시스템 .....	20
2) Processing을 이용한 영상제작 .....	22
3) Arena를 이용한 영상효과 .....	25
3. 공연 및 연동 시스템 연구 .....	27
1) 공연 시스템 .....	27
2) 음악과 영상의 연동 시스템 .....	29

① OSC 통신을 이용한 음악과 영상의 연동 .....	29
② Syphon을 이용한 Processing과 Arena의 연동 .....	32
<b>III. 연구 기술의 작품 적용 .....</b>	<b>34</b>
1. 작품 소개 .....	34
2. 작품 구성 .....	35
1) 음악 구성 .....	35
2) 영상 구성 .....	36
3. 작품 내 사운드 및 영상 기술 적용 .....	39
1) 각 악기별 사운드 적용 효과 .....	39
① 베이스 트롬본의 사운드 적용 효과 .....	39
② 피아노의 사운드 적용 효과 .....	40
2) Section-I 적용 효과 .....	42
3) Section-II 적용 효과 .....	44
4. 연구 기술의 작품 적용 효과 .....	46
<b>IV. 결 론 .....</b>	<b>47</b>
<b>참 고 문 헌 .....</b>	<b>49</b>
<b>ABSTRACT .....</b>	<b>52</b>
<b>부록-1 : 작품 &lt;Yes, I am.&gt; 악보 .....</b>	<b>54</b>
<b>부록-2 : 첨부 DVD .....</b>	<b>62</b>

## 표 목 차

<표-1> grain pitch 파라미터 의미 .....	10
<표-2> 피아노의 작품 구성에 따른 crossover frequency .....	24
<표-3> Arena의 영상효과 .....	26
<표-4> 영상 시나리오 .....	36
<표-5> 베이스 트롬본에 사용된 음향효과들의 조합 .....	39
<표-6> 피아노에 사용된 음향효과들의 조합 .....	40
<표-7> Section-I에 적용된 사운드 프로세싱 및 영상효과 .....	42
<표-8> Section-II에 적용된 사운드 프로세싱 및 영상효과 .....	44

## 그 립 목 차

[그림-1] 작품 <Self-Conscious>의 영상 캡처 .....	3
[그림-2] 사운드 시스템 설계도 .....	6
[그림-3] sine파의 original과 granular synthesis를 입힌 파형 .....	8
[그림-4] munger~오브젝트를 이용한 granular synthesis 패치 .....	9
[그림-5] 베이스 트롬본의 original과 comb filtering을 거친 파형 .....	10
[그림-6] comb~오브젝트를 이용한 comb filter 패치 .....	11
[그림-7] delay 음향효과를 구현한 Max 패치 .....	12
[그림-8] flanger 음향효과를 구현한 Max 패치 .....	13
[그림-9] ring modulation의 시스템 .....	14
[그림-10] ring modulation을 구현한 Max 패치 .....	15

[그림-11] phase vocoder 음향효과를 Max에서 구현한 패치 .....	16
[그림-12] phase vocoder의 내부 패치 .....	17
[그림-13] pfft~ xover~오브젝트를 이용한 Max 패치 .....	18
[그림-14] 베이스 트롬본과 피아노를 표현한 형태 .....	20
[그림-15] 영상 시스템 .....	22
[그림-16] 베이스 트롬본의 음량 값에 따른 이미지의 크기 변화 .....	23
[그림-17] 베이스 트롬본의 음량 값에 의한 이미지의 개수 .....	24
[그림-18] 피아노의 음량 값에 따른 이미지의 변화 .....	25
[그림-19] 공연 시스템 설계도 .....	27
[그림-20] 무대 구성 .....	28
[그림-21] Max와 Processing의 OSC 통신을 위한 Max 패치 .....	29
[그림-22] Processing에서 OSC 라이브러리를 받는 모습 .....	30
[그림-23] OSC 라이브러리를 불러오는 코드 .....	30
[그림-24] OSC 통신을 하기 위한 주소 설정 코드 .....	31
[그림-25] OSC 입력 데이터를 변수로 설정하는 코드 .....	31
[그림-26] OSC 통신을 이용한 Max와 Arena의 연동 .....	32
[그림-27] Processing에서 Syphon을 사용하는 코드 .....	33
[그림-28] Syphon을 통해 공유된 영상소스 .....	33
[그림-29] bright.contrast 효과에 의한 영상의 변화 .....	43
[그림-30] tint 효과에 의한 영상의 변화 .....	45
[그림-31] 클라이맥스 전개에 따른 영상의 변화 .....	45

# I. 서론

## 1. 연구 배경 및 목적

20세기를 기준으로 컴퓨터의 비약적인 발전으로 인해, 청중들은 어쿠스틱 음악<sup>1)</sup>을 듣는 것만이 아닌 디지털 매체로 음악을 들을 수 있게 되었다. 또한 시각적인 측면에서도 많은 발전을 이루었으며 미술 작품 및 안무가의 퍼포먼스를 관람하는 것을 비롯해 디지털 매체를 통한 컴퓨터 그래픽, 프로젝션 매핑(projection mapping)<sup>2)</sup>, 미디어 파사드(media facade)<sup>3)</sup> 등 다채로운 경험을 할 수 있게 되었다. 즉, 관객들은 시각적인 측면과 청각적인 측면을 동시에 즐길 수 있게 되었으며, 음악과 영상이 서로 결합된 멀티미디어(multimedia)<sup>4)</sup> 작품을 어렵지 않게 감상할 수 있다. 또한 음악과 영상이 서로 연동되어 인터랙션(interaction)이 부여된 작품도 나오게 되었다.

본 논문에서 설명하는 <Yes, I am.>은 악기 연주와 동시에 실시간으로 프로세싱된 소리와 그에 반응하는 영상을 같이 보여줄 수 있는 멀티미디어음악 작품이다. 어쿠스틱 악기 소리와 함께 실시간으로 악기의 소리가 변형이 되어 다른 형태의 음악을 만들고, 음악에 실시간으로 반응하는 영상의 인터랙션을 통해 음악을 시각화하여 하나의 멀티미디어 작품으로 만들어 내는 것에 중점을 두고 있다.

- 
- 1) 전기를 공급하는 전자기기가 포함되지 않은 순수한 울림을 연출하는 악기로 이루어진 음악을 말한다.
  - 2) 대상물의 표면에 빛으로 이루어진 영상을 투사하여 변화를 줌으로써, 현실에 존재하는 대상이 다른 성격을 가진 것처럼 보이도록 하는 기술이다.
  - 3) '미디어(media)'와 건물 외벽을 뜻하는 '파사드(facade)'의 합성어로, 건물 외벽 등에 LED 조명을 설치해 미디어 기능을 구현하는 것을 말한다.
  - 4) 디지털화된 미디어의 복합체, 글자·소리·영상의 복합체이며, 디지털 상태인 것을 말한다.

<Yes, I am.>에 사용된 악기는 베이스 트롬본과 피아노이다. 베이스 트롬본은 금관악기 군에 속하는 악기이며, 밸브, 크록, 키 등을 사용하는 다른 금관악기와 다르게 슬라이드를 가지고 있어 반음계를 연주하는 것에 탁월하다. 슬라이드를 활용하여 글리산도<sup>5)</sup>효과를 내기 쉬우며, 선율을 부드럽게 이어서 연주하기에 효과적이다. 또한 스포르찬도<sup>6)</sup>나 포르테-피아노<sup>7)</sup>, 텅잉<sup>8)</sup>등의 주법을 활용하여 다양한 표현을 할 수 있는 악기이다. 피아노는 넓은 음역과 폭 넓은 다이내믹 레인지(dynamic range)<sup>9)</sup>를 가지며, 선율악기와 화성악기 이 두 가지 요소를 모두 포함하고 있다. 두 악기의 공통점은 풍부한 배음이 있다는 점이다. 따라서 악기를 통한 사운드 프로세싱에 적합하며, 다양한 효과를 구현할 수 있다는 이유로 본 작품에 사용하게 되었다.

본 논문에서는 Max<sup>10)</sup>를 활용한 사운드 프로세싱 기술과 Processing<sup>11)</sup>과 Arena<sup>12)</sup>를 사용하여 영상을 만든 방법에 대해 알아보며, 연구 기술을 적용한 방법과 그에 따른 효과를 언급한다.

- 
- 5) 높이가 다른 두 음 사이를 급속한 음계에 의해 미끄러지듯이 연주하는 방법.
  - 6) 악보에서 '특히 그 음을 세게 연주'하라는 뜻으로 기호는 'sf' 또는 'sfz' 이다.
  - 7) 강하게 그리고 곧 약하게 연주하라는 말.
  - 8) 관악기를 연주할 때 혀의 운동에 의해 공기의 흐름을 중단하는 기술.
  - 9) 한 음량에서 가장 조용한 소리와 가장 큰 소리의 차이를 일컫는 말.
  - 10) Cycling'74에서 개발한 소프트웨어로 음악과 멀티미디어를 위한 개발 환경을 제공하는 프로그램이다. 테이터의 연산처리 및 프로그래밍이 가능한 Max와 음향 신호 및 MIDI데이터 처리가 가능한 MSP, 그리고 real-time video와 2D/3D 벡터 그래픽을 다루는 Jitter로 이루어져 있다. 본 연구에서는 최신 버전인 Max8을 사용한다.
  - 11) MIT 미디어 랩의 'Aesthetics and Computation' 그룹에 의해 2001년 만들어진 오픈 소스 프로그래밍 언어.
  - 12) Resolume에서 개발한 영상 믹싱 프로그램으로 실시간 비주얼 퍼포먼스를 위해 주로 사용. 본 연구에서는 최신 버전인 Arena6를 사용한다.

## 2. 선행 연구

선행 연구로 2017년 11월 테너 트롬본을 활용한 사운드 프로세싱 작품 <Self-Conscious><sup>13)</sup>를 제작했다. [그림-1]은 만들어진 작품의 한 장면이다.



[그림-1] 작품 <Self-Conscious>의 영상 캡처

위의 작품에 사용된 악기는 테너 트롬본이며, 사운드 프로세싱에는 delay<sup>14)</sup> 음향효과와 phase vocoder<sup>15)</sup>, granular synthesis<sup>16)</sup>, comb filter<sup>17)</sup>가 실시간으로 사용되었다.

13) <https://www.youtube.com/watch?v=e1-SCYDKSJ0>

14) 입력된 사운드를 지정한 시간 후에 출력하는 효과를 말한다.

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

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

17) 주파수 특성에 빗살(comb) 모양이 나타나는 필터이다.

트롬본은 호흡으로 소리가 발생하는 관악기 구조이기 때문에 연주자가 숨을 들이마시는 최소한의 시간이 필요하여, 원하는 만큼의 긴 프레이즈 (phrase)<sup>18)</sup>를 연주하기 힘들었다. 또한 단선율 악기이기 때문에 곡의 시작부터 끝까지 선율의 진행으로밖에 사용하지 못했고 곡의 구성이 비교적 단순해지게 되었다. 이를 보강하기 위해 granular synthesis를 사용하여 패드(pad)<sup>19)</sup> 사운드를 만들었지만, 그 음정에 대한 화음이 형성되지 않았고 옥타브 소리만 형성이 되었다. 그 결과, 의도했던 풍부한 화음을 만들지 못하게 되었으며 완전한 호모포니(homophony)<sup>20)</sup>의 구조를 형성하지 못했다. 또한 트롬본의 특성을 가지고 있으면서도 더 낮은 음정을 소화할 수 있는 악기가 필요했으며, 같은 음정을 내더라도 더 많은 저음의 배음이 필요했다.

선행 연구의 결과에 따라 본 작품 <Yes, I am.>에 보완한 사항은 다음과 같다. 화음을 연주할 수 있고, 페달로 인한 긴 여운효과를 줄 수 있는 피아노를 추가하게 되었으며, 테너 트롬본의 음역보다 더 낮은 음정을 소화할 수 있는 중저음의 음색을 가진 베이스 트롬본을 사용하게 되었다.

---

18) 음악 악구, 악절.

19) 지속되는 코드 혹은 톤(tone)을 뜻하는 음향 기법이다.

20) 화성음악. 화성이 하나의 선율을 받쳐주는 음악.

## II. 기술 연구

### 1. 사운드 시스템 연구

#### 1) 사운드 시스템

베이스 트롬본의 실시간 사운드 프로세싱은 flanger<sup>21)</sup>, granular synthesis, comb filter, ring modulation<sup>22)</sup>이 사용되었고, 피아노의 실시간 사운드 프로세싱에는 phase vocoder, delay, granular synthesis가 사용되었다. 언급된 음향효과는 모두 Max를 통하여 제작되었으며 각 음향효과별로 컨트롤러에 매핑하여 사용하였다.

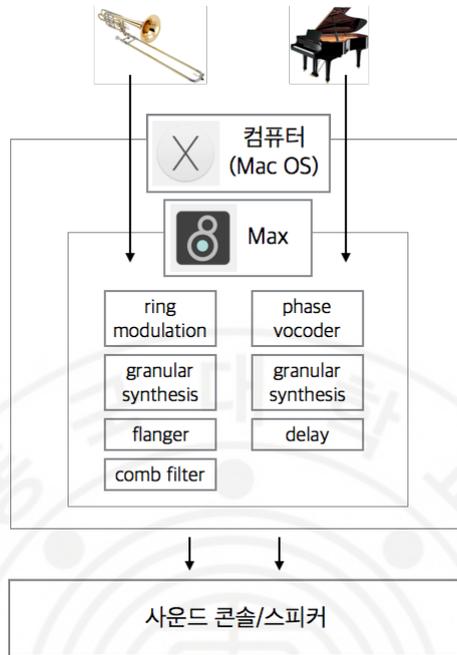
[그림-2]는 베이스 트롬본과 피아노의 실시간 프로세싱을 위해 제작한 사운드 시스템이다. 각 악기들은 개별의 마이크를 사용하여 3채널<sup>23)</sup>을 오디오인터페이스로 전달받고 컴퓨터에 입력된다. 컴퓨터에 입력된 악기 소리들은 Max로 전달되며, 곡의 구성에 따라 제작된 음향효과들이 적용된다. 악기의 오리지널 사운드와 프로세싱된 사운드는 합쳐져서 사운드 콘솔을 통해 스피커로 재생된다.

---

21) delay를 응용한 음향효과. 지연된 음에 25-50ms정도 지연시킨 사운드를 혼합하여 풍부함을 내는 효과.

22) 두 오실레이터의 배음 성분의 주파수들이 서로 간섭하여 새 배음렬을 만들어 내는 사운드 변조 방식.

23) 피아노는 스테레오로 받았기 때문에 2채널을 사용하며, 베이스 트롬본까지 총 3채널이다.



[그림-2] 사운드 시스템 설계도

## 2) 음향효과 연구 과정

두 대의 악기를 사용하다보니 제어해야하는 음향효과의 개수가 많아졌다. 실시간으로 사운드와 영상을 동시에 조작해야하므로 각 악기 특성을 최대한 살린 효과들로 간추리게 되었다.

베이스 트롬본은 슬라이드를 활용하여 음정이 변하도록 하는 비브라토를 쉽게 연주할 수 있다. 반면에 호흡이 많이 필요한 악기 특성상, 한 음정을 떨리듯이 빠르게 연주하는 방법은 연주자에게 쉽지 않다. 따라서 ring modulation 효과를 주어 의도적인 음정의 떨림을 표현했다. 또한 일정한 박자로 스타카토<sup>24)</sup>를

24) 악보에서, 한 음 한 음씩 또렷하게 끊는 듯이 연주하라는 말.

연주할 때, 입력되어지는 소리를 1초 지연시킨 후 베이스 트롬본의 고음역부분만 사용하여 comb filter를 입힌 효과를 적용했으며, 악기의 고음역만 사용하기 위한 방법으로는 FFT<sup>25)</sup>분석이 필요했다. pfft~xover~<sup>26)</sup> 오브젝트를 사용하여 기준점이 되는 주파수를 정하고 그 이상의 주파수만 사용하였다.

피아노는 연주하는 소리를 실시간으로 reverse<sup>27)</sup> 시킬 수 있는 음향효과가 필요했다. 이에 사용된 효과는 FFT분석을 통한 phase vocoder가 사용되었다. 또한 여운효과 및 소리를 겹겹이 보강하기 위해서 granular synthesis를 사용했다. 앞서 언급한 음향효과들끼리 다양하게 조합하여 사용했으며 조합한 방법 및 자세한 효과는 연구 기술의 작품 적용에서 언급한다.

### ① granular synthesis 음향효과

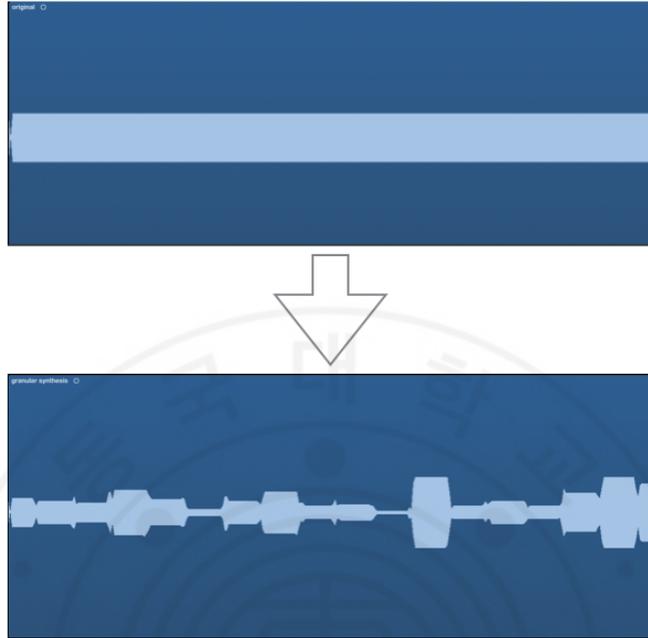
granular synthesis는 입력된 사운드를 아주 작은 시간단위로 나뉘서 재조합하는 합성방식을 의미한다. [그림-3]은 440Hz의 sine 파형과 그 파형에 100ms단위로 나눈 granular synthesis를 입힌 파형이다.

---

25) Fast Fourier Transform의 약자로 고속푸리에변환이라고도 한다. 주로 복잡한 파형이나 신호들을 보다 간단하게 변환시켜주는 알고리즘을 이야기 한다. 파형을 분석하여 그 데이터를 이용해 원하는 대로 소리를 재합성 할 때 많이 사용된다.

26) pfft~오브젝트는 patcher fft라는 뜻으로 fft분석 기능을 이용하게 해주는 서브패치를 말한다. 그중에서, xover~기능은 기준 주파수에 따라 주파수 대역을 나누어 주는 역할이다.

27) 재생방향을 역방향으로 만들어주는 음향효과이다.

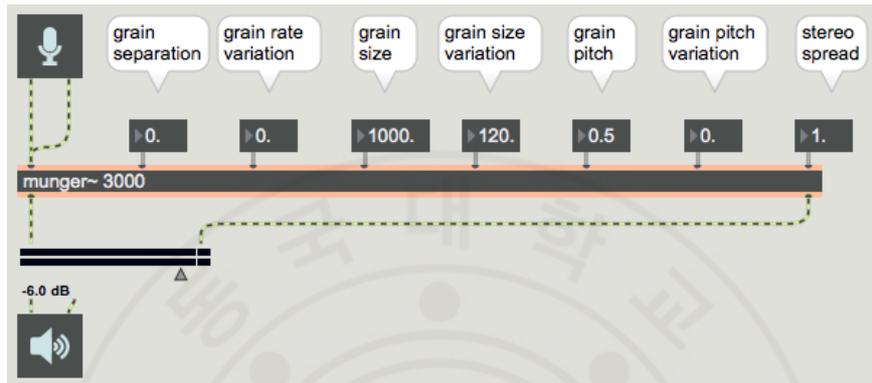


[그림-3] sine파의 original과 granular synthesis를 입힌 파형

[그림-3] 과 같이 granular synthesis를 적용하면 하나의 곡물입자처럼 생겼다하여 grain이라고 부른다. 이 작은 샘플조각을 나누는 기준은 ms(밀리세컨드)이며, grain 효과를 적용하면 여러 개로 나뉜 샘플조각들이 생성되고 original 사운드의 새로운 변형이 가능하다. 본 논문에서는 granular synthesis를 구현하기 위해 mungler<sup>28)</sup> 오브젝트를 사용하였다.

28) Max의 외부(external)오브젝트이며, 내장된 오브젝트가 아니기 때문에 Max의 Package manager에서 다운받아야한다. Columbia University에서 만들었으며 본 작품에서 사용된 버전은 Dan Trueman과 R. Luke Dubois에 의해 수정된 버전이다.

[그림-4]는 granular synthesis를 Max에서 구현하기 위해 만들어진 munger~오브젝트 패치이다.



[그림-4] munger~오브젝트를 이용한 granular synthesis 패치

munger~는 총 7개의 파라미터가 있으며, 이것을 조절함으로써 여러 가지 효과를 얻을 수 있다. 크게 입자 사이의 간격(grain pitch), 입자의 크기(grain size), 음정의 높낮이(grain pitch) 세 가지로 나뉘며, 변화(variation) 값을 통해 일정하게 나누는 것이 아닌 불규칙적인 변화를 줄 수 있다. grain pitch는 음정을 변화시켜주는 것을 말한다. 1이라는 값을 기준으로 한 옥타브 위는 2, 두 옥타브 위는 4이며, 반대로 0.5는 한 옥타브 아래, 0.25는 두 옥타브 아래다. 즉, 2<sup>n</sup>에서 지수의 값에 따라 달라진다. <표-1>은 grain pitch의 파라미터 값을 정리한 것이다.<sup>29)</sup> stereo spread는 음원을 가운데로 모이게 하느냐, 넓게 입체적으로 사용할 것이냐에 따라 나뉘며, 0과 1의 값을 받는다.

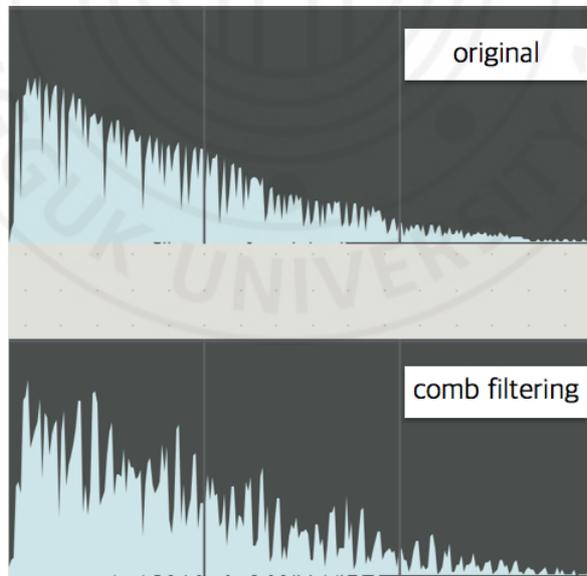
29) 한승욱, 「피아노의 실시간 프로세싱을 이용한 멀티미디어음악 제작 연구 (멀티미디어음악 작품 <Midnight Peace>를 중심으로)」, 동국대학교 영상대학원: 멀티미디어학과, 석사학위논문, 2018, p. 4

<표-1> grain pitch 파라미터 의미

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

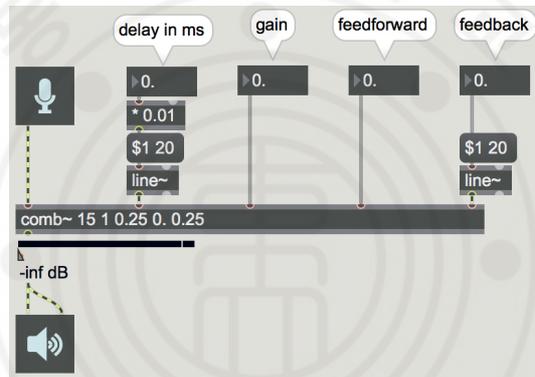
② comb filter 음향효과

comb filtering 효과는 직접음에 의한 반사음이 매우 짧은 시간차에 의해 겹쳐지는 현상을 뜻한다. 어떠한 파형을 시간차를 두고 2개를 합성하게 되면 어떤 주파수는 상쇄되고, 어떤 주파수는 증가하게 된다. [그림-5]는 베이스 트롬본의 original 사운드와 comb filtering을 했을 때, 발생하는 파형의 모습이다.



[그림-5] 베이스 트롬본의 original과 comb filtering을 거친 파형

위의 그림과 같이, 빗 모양을 하고 있다하여 comb filter라고 불린다. 보통 두 개 이상의 마이크를 설치했을 때 발생하는 거리의 차이에 의해 시간차가 생기며, 음색이 왜곡된다. 음향에서는 하울링<sup>30)</sup>이 발생하기 때문에 위험한 현상으로 인식되지만, 사운드 프로세싱에서는 의도적으로 만들어 구현을 해볼 수 있다. [그림-6]은 comb filtering 효과를 Max에서 구현한 패치이다. comb~오브젝트를 사용하며, delay time을 조절하면 음색이 미세하게 변화된다. 금속성 물질처럼 날카로운 소리가 나는 것이 특징이다.

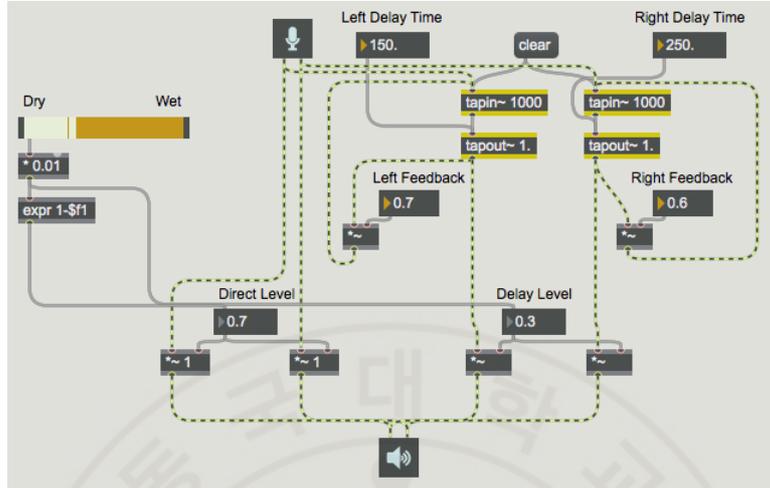


[그림-6] comb~오브젝트를 이용한 comb filter 패치

### ③ delay 음향효과

delay 음향효과란 음향 신호를 지연시키는 효과를 말하며, 오리지널 사운드와 결합하여 나오는 것을 뜻한다. Max에서는 tapin~오브젝트와 tapout~오브젝트를 활용하여 만들 수 있으며, [그림-7]은 delay 음향효과를 Max에서 구현한 패치이다.

30) 스피커에서 나오는 음이 마이크로 들어가 앰프에서 재 증폭되어 출력되는 현상.



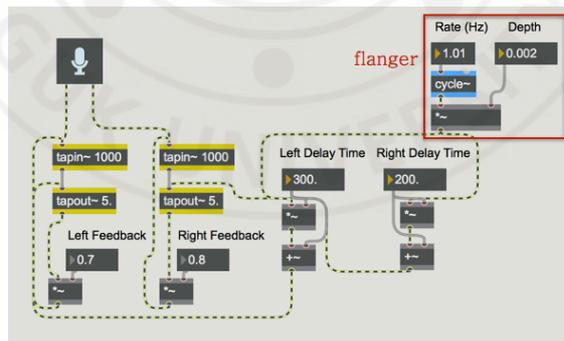
[그림-7] delay 음향효과를 구현한 Max 패치

tapin~오브젝트는 입력된 오디오 신호를 저장하며, 용량이 가득 차면 자동으로 저장되었던 기존의 신호를 보냄과 동시에 새로운 신호를 받는다. tapout~오브젝트는 tapin~오브젝트에 저장된 오디오 신호를 지정한 시간만큼 지연시켜 내보낸다. 이 두 오브젝트를 활용하게 되면 지정한 시간만큼의 delay효과를 만들어낼 수 있다. tapin~과 tapout~을 거쳐서 나온 지연된 신호를 일정비율만큼 곱해서 tapin~에 다시 신호를 입력시키면 feedback 효과를 만들 수 있다. 또한 expr오브젝트<sup>31)</sup>를 통해 original 사운드와 delay 사운드의 비율을 조절할 수 있으며, dry/wet의 명칭을 가진다.

31) 입력된 인수를 계산식으로 인식하여 연산해주는 오브젝트.

#### ④ flanger 음향효과

flanger 음향효과는 delay 음향효과를 응용한 것으로, 입력한 음을 약간 지연시켜 다시 원음에 더하는 것이다. doppler 효과를 예로 들 수 있다. doppler 효과란 어떠한 소리와 청취자의 상대 속도에 따라 주파수와 파장이 바뀌게 들리는 것을 말한다. 사이렌 소리를 들을 때, 멀리 있는 사이렌 소리가 청취자와 가까워지면 음정이 더 높아지고 다시 멀어질수록 음정이 낮게 들리는 현상이다. 여기서 착안할 수 있는 점은, 거리가 멀어지면 소리의 크기가 작아지며, 음정이 낮아진다. 이것을 토대로 주기적으로 시간을 지연시키게 되면 음정이 주기적으로 변한다는 것을 알 수 있다. [그림-8]은 flanger 음향효과를 Max에서 구현한 패치이다. delay 음향효과에서 응용이 되며, cycle~ 오브젝트<sup>32)</sup>를 통해 주기적으로 움직이는 파형의 수치를 각각의 delay time에 곱하고 더해주어 tapout~오브젝트에 입력을 시킨다. 그 결과 원래의 소리와 아주 짧게 지연된 소리가 합쳐져서 음정이 주기적으로 변함과 동시에 comb filtering 효과가 난다.



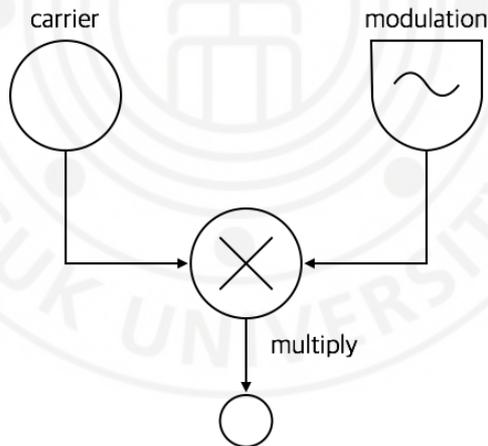
[그림-8] flanger 음향효과를 구현한 Max 패치

32) 주기적으로 cosine 파형을 생성하는 오브젝트.

### ⑤ ring modulation 음향효과

전자음악에서 modulation이란 하나의 신호 혹은 하나의 소리가 가진 특성이 다른 신호의 특성에 의해 변화되는 것을 말하며, 변조라고도 불린다. 예를 들어 소리의 음량을 변조시키면 amplitude modulation이라 하며, 소리의 주파수를 변조시키면 frequency modulation이라고 불린다.

[그림-9]는 ring modulation의 시스템이다. original sound를 커리어(carrier: 이하  $C$ )라 하며 변조를 하는 신호를 모듈레이터(modulator: 이하  $M$ )라 한다. ring modulation의 경우, 음량을 변조시키는 amplitude modulation의 한 형태로 변조시키는 신호의 주파수에 따라 original 사운드의 주파수가 변하는 것이다.



[그림-9] ring modulation의 시스템

M이 0~20Hz인 경우 ring modulation은 트레몰로<sup>33)</sup> 효과가 나타나며, 20Hz~20kHz범위인 경우 실질적인 변조가 일어난다. 본 논문에서는 주로 변조를 사용한 것이 아닌, 트레몰로 효과를 구현하였다. [그림-10]은 ring modulation 음향효과를 Max에서 구현한 패치이다.



[그림-10] ring modulation을 구현한 Max 패치

#### ⑥ FFT분석을 이용한 phase vocoder 음향효과

phase vocoder 음향효과란 녹음된 사운드에서 time-stretching<sup>34)</sup>과 pitch-shifting<sup>35)</sup>을 하는 것을 말한다. 일반적으로 time-stretching을 하게 되면, 샘플 원본의 음정이 유지가 되지 않는다. 또한 pitch-shifting을 할 때에, 샘플 원본이 손상이 되기 쉽다. 하지만, FFT분석을 이용한 phase vocoder는 수많은 bandpass filter<sup>36)</sup>를 통해 그 원본이 가지고 있는 주파수 대역 전체를 나누어서 분석한 후, 다시 합성하는 방식이며 음정의 변화 없이 time-stretching이 가능하다. 따라서 샘플 원본이 손상되지 않으면서

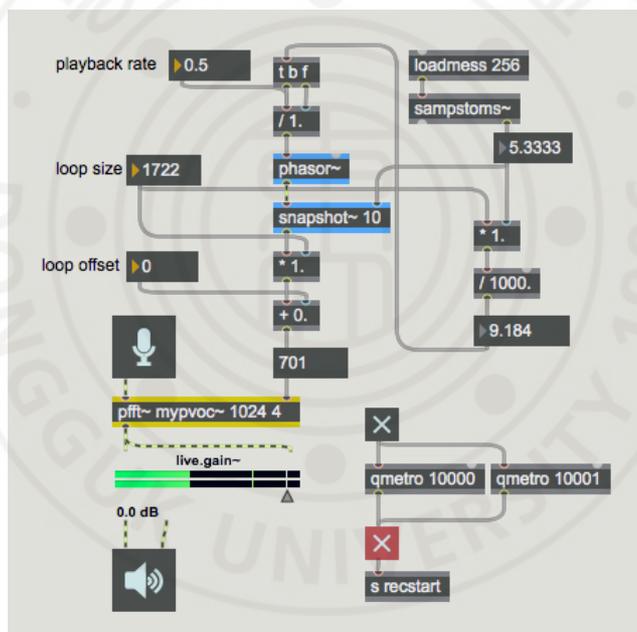
33) 음 또는 화음을, 빨리 떨리는 듯이 되풀이하는 연주법.

34) 샘플의 재생시간을 줄이거나 늘리는 것.

35) 샘플의 음정을 낮추거나 높이는 것.

36) 특정 주파수 사이의 신호만 통과시키는 필터이다.

pitch-shifting을 할 수 있다. [그림-11]은 phase vocoder 음향효과를 Max에서 구현한 패치이다. playback rate 파라미터는 분석된 사운드의 속도를 정할 수 있다. 1이라는 수치가 그대로 재생하는 것이며, 수치를 높이거나 낮춤으로써, 속도를 높이거나 낮출 수 있다. 음수를 입력하게 되면, 역으로 재생된다. loop size 파라미터는 사운드가 반복을 하는 구간을 의미하며, 1722프레임<sup>37)</sup>으로 고정하였다. loop offset 파라미터는 loop size의 시작 지점을 의미한다. 각각의 파라미터를 조정할 시 다양한 사운드 변형이 가능하다.

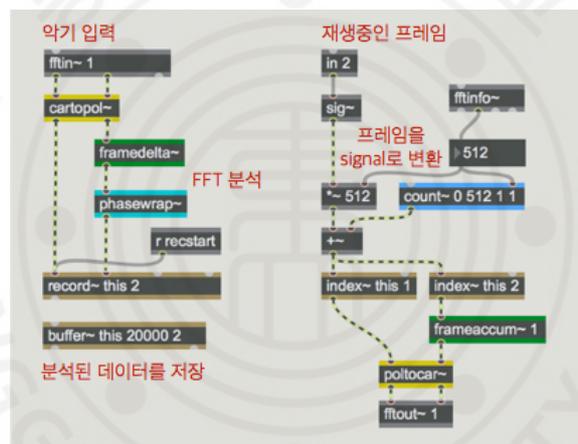


[그림-11] phase vocoder 음향효과를 Max에서 구현한 패치

37) 44100Hz의 sample rate값을 가진 사운드 데이터의 20초는 882000의 sample수를 가지며, 1024개의 FFT사이즈를 가짐과 동시에 1722의 프레임 값을 가진다. 20초라는 설정은 현재 사용하고 있는 phase vocoder 내부의 buffer~의 저장 공간 크기에서 비롯되었다.

phasor~오브젝트<sup>38)</sup>를 통해 재생되는 음원의 속도를 지정할 수 있으며, 계산식을 통해 프레임으로 변환시켜 pfft~ mypvoc~오브젝트의 두 번째 인렛으로 입력된다.

하지만, 이 패치만을 사용하게 되면 실시간으로 변형되는 phase vocoder 음향효과를 가지기 어렵다. qmetro오브젝트<sup>39)</sup>를 사용하여 각각 10, 10.001초 간격으로 켜고 끌 수 있게 했다. 그 결과, 10초만큼을 녹음했다가 다시 끄고 녹음하게 해주어 실시간으로 사용할 수 있게 되었다.



[그림-12] phase vocoder의 내부 패치

[그림-12]는 phase vocoder의 내부 패치이다. 입력된 사운드 데이터는 FFT분석을 통해 가공되어 record~오브젝트<sup>40)</sup>로 녹음된 후 buffer~오브젝트<sup>41)</sup>에 저장된다.

38) sawtooth파형을 생성해주는 오브젝트. 왼쪽 인렛으로 들어가는 숫자는 1초 동안 0에서 1까지 몇 번 움직이느냐를 결정한다.

39) 설정한 시간(ms) 간격으로 bang 메시지를 출력하는 Max 오브젝트.

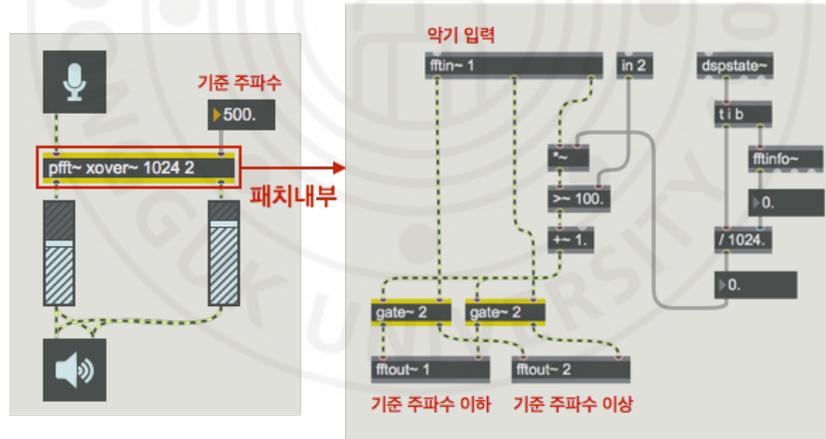
40) 입력되는 사운드 데이터를 buffer~에 녹음하는 기능을 가진 오브젝트.

41) 컴퓨터 메모리에 오디오 샘플을 저장하는 공간이다.

재생중인 프레임은 sig~오브젝트<sup>42)</sup>를 거쳐 audio signal로 변환시켜 프레임 크기와 곱해져 index~오브젝트<sup>43)</sup>에 전달된다. 즉, phase vocoder의 내부 패치는 사운드 데이터가 프레임으로 변환되어 음원이 재생되어지는 구조에 대한 것이다.

### ⑦ FFT분석을 이용한 pfft~ xover~오브젝트 활용

사운드 프로세싱을 하는 경우에는 악기로부터 들어온 소리를 전부 활용할 필요가 없으며, 오히려 음역 전체를 사용하면 사운드 프로세싱의 효과가 덜하거나 소리들이 중첩되어 지저분해지는 경우가 생긴다. 그래서 특정 주파수를 기준으로 저음역이나 고음역을 분리시켜 활용할 수 있는 방안을 모색하였다.



[그림-13] pfft~ xover~오브젝트를 이용한 Max 패치

42) 숫자를 오디오 signal로 변환시켜주는 오브젝트

43) buffer~에 저장된 사운드 데이터를 샘플단위로 읽어 재생하는 오브젝트.

패치내부를 살펴보면, fftin~오브젝트44)를 거쳐 FFT분석을 하게 된다. 분석을 완료한 주파수가 >~오브젝트45)에서 지정한 주파수보다 낮으면 0이 나오게 된다. +~오브젝트46)에서 1을 더하게 되면 gate~오브젝트47)의 첫 번째 outlet으로 보내게 되며, 반대의 경우에는 2가 되어 gate~오브젝트의 두 번째 outlet으로 보내게 되는 원리이다.48)



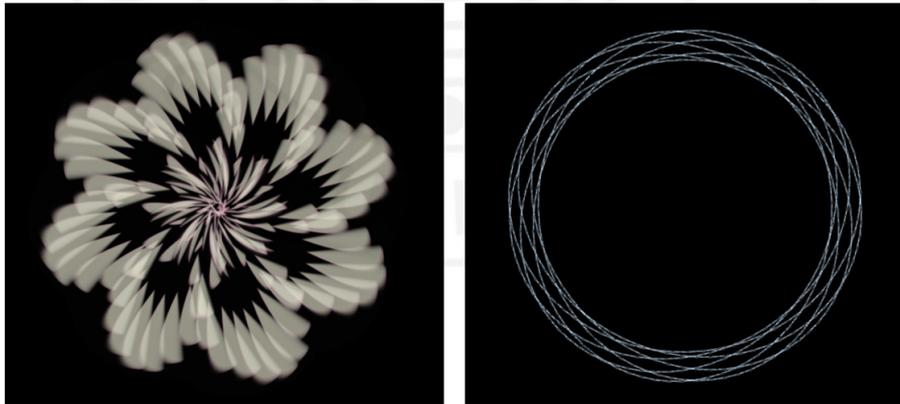
- 
- 44) pfft~오브젝트에 의해 로드된 패치에 신호입력을 제공한다.
  - 45) 왼쪽과 오른쪽 시그널의 크기를 비교하는 오브젝트. 왼쪽 시그널이 크면 1을 내보내고 그렇지 않으면 0을 내보낸다.
  - 46) 입력된 두 개의 신호를 더해준다.
  - 47) 오른쪽 inlet으로 들어오는 신호를 왼쪽에 들어오는 숫자에 따라 몇 번째 outlet으로 내보낼 것인지를 결정할 수 있는 오브젝트. 0이 들어오면 모든 outlet이 닫힌다.
  - 48) 전우진, 「컴퓨터음악과 phase music을 이용한 인터랙티브 멀티미디어 퍼포먼스 연구 (멀티미디어음악 작품 <The Phase>를 중심으로)」, 동국대학교 영상대학원: 멀티미디어학과, 석사학위논문, 2017, p. 6

## 2. 영상 시스템 연구

### 1) 영상 시스템

음악과 영상을 서로 연동시킨 인터랙션이 부여된 작품을 만들기 위해서는 특정 정보의 수치가 전달되고, 그 수치에 의해 반응을 해야 한다. 본 연구에서는 각 악기의 음량 값과 음향효과가 적용된 사운드의 음량 값을 수치로 받아서 영상에 적용하였다. 2대의 악기에서 나오는 사운드를 효과적으로 시각화하기 위해, 각 악기별로 대응되는 이미지를 각각 제작하였다. 두 개의 이미지가 서로 공존하거나, 하나의 이미지가 다른 이미지를 보완할 수 있도록 하는 독자적인 이미지가 필요했다.

[그림-14]는 각각 트롬본과 피아노의 소리를 시각화한 이미지이다. 베이스 트롬본은 베지어 곡선(Bézier Curve)<sup>49)</sup>을 활용한 원형의 형태이며, 피아노는 선을 활용한 원형의 형태이다.



[그림-14] 베이스 트롬본과 피아노를 표현한 형태

49) 특정 점이 그리는 궤적을 이용해 곡선을 그려내는 방법.

하지만, 한 가지의 패턴만 나오게 된다면 영상이 단조롭기 때문에 하나의 패턴을 기준으로 여러 가지 수식을 통해 형태를 변형시켜야했다. 영상을 제작하는 프로그램으로는 Processing을 사용했으며, 코딩(coding)<sup>50)</sup>을 통해 원하는 계산식과 수치를 입력할 수 있는 장점이 있다. Processing은 실시간으로 두 영상을 각각 생성할 수 있지만 작품의 구성에 맞춘 영상의 배치를 위해, 별도의 프로그램이 필요했으며 Arena를 사용했다. Arena는 오디오 신호를 직접 입력할 수 있으며, OSC<sup>51)</sup> 데이터와 MIDI<sup>52)</sup> 데이터의 입력이 가능하여 사운드와의 연동이 쉽다. 그러나 Processing과 Arena는 서로 호환이 되지 않기 때문에, Syphon<sup>53)</sup>을 사용하여 호환시켰다.

[그림-15]는 영상 시스템 설계도이다. Max를 통해 사운드가 출력이 되면, OSC 데이터를 통해 각각 Processing과 Arena로 전송이 된다. Processing의 영상을 Syphon을 통해 Arena로 전송하여, 영상 믹싱 및 효과를 주어 최종적으로 프로젝터로 출력한다.

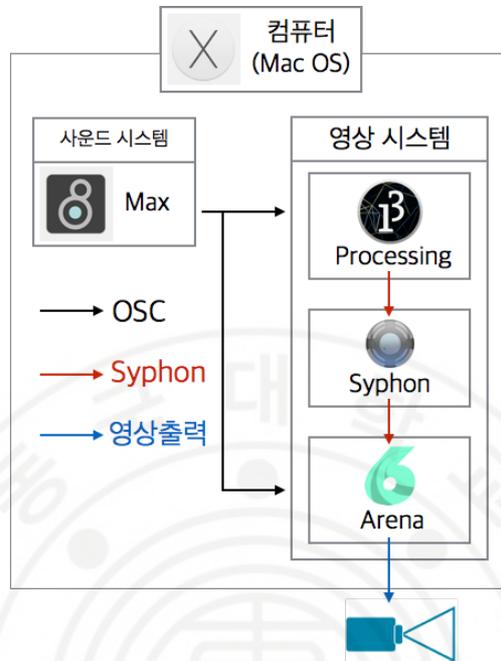
---

50) 일정한 프로그램 언어를 써서 프로그램을 작성하는 것.

51) Open Sound Control의 약자로 사운드 데이터 전송을 위해 개발된 네트워크를 이용한 통신 규약이다.

52) Musical Instrument Digital Interface의 약자로 서로 다른 전자악기간의 데이터를 전송하고 공유하기 위해 제정한 업계 표준이다.

53) 응용 프로그램간의 영상 프레임을 서로 공유할 수 있게 해주는 Mac OS 기반의 오픈소스.



[그림-15] 영상 시스템

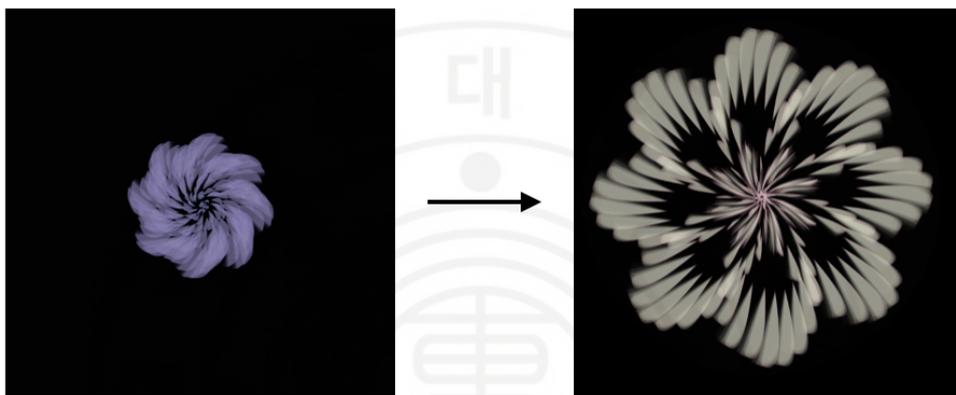
## 2) Processing을 이용한 영상제작

영상 소스는 모두 Processing을 이용하여 제작하였고, 베이스 트롬본과 피아노를 각각 두 개의 Processing 영상으로 만들었다. 음악과 영상을 연동시키는 요소는 여러 가지가 있었지만, 그중에서 음량 값만 사용하였다.

베이스 트롬본의 경우는 pitch-detection<sup>54)</sup>을 통해 음정 값을 받아 사용하려 했지만, 트롬본은 슬라이드를 활용하여 음정을 내는 구조이기 때문에 음정이 정확하게 잡히지 않았다. 따라서 베이스 트롬본의 음량 값만 가지고 이미지의

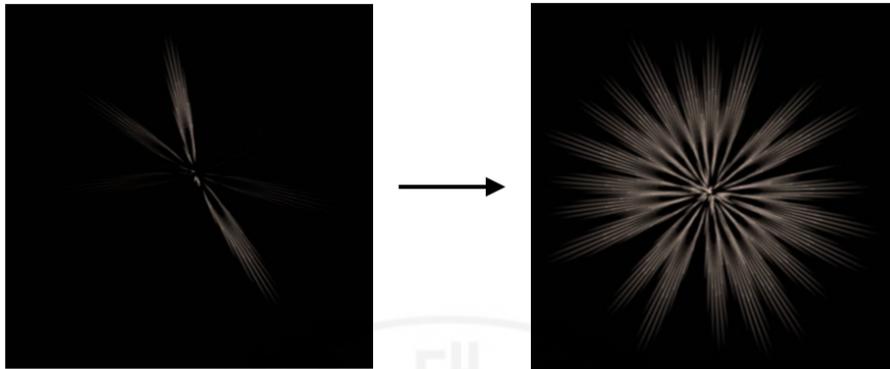
54) 오디오 신호를 분석하여 음정을 잡아내는 기술.

크기만을 연구했다. 음량이 작으면 이미지가 작아지고 음량이 커지면 이미지가 커지는 방법을 사용했지만, 곡의 처음부터 끝까지 이와 같은 형태로 진행되는 것은 단조롭다. [그림-16]은 베이스 트롬본의 음량 값에 따른 이미지의 크기 변화를 나타낸 것이다.



[그림-16] 베이스 트롬본의 음량 값에 따른 이미지의 크기 변화

이미지의 크기 외에 다른 요소들을 연구해본 결과, 이미지의 색깔과 이미지가 가지고 있는 객체의 개수, 이미지가 가지고 있는 알파(alpha)값, 이미지가 움직이는 속도들을 대입해볼 수 있었고, 결과적으로 영상의 단조로움을 덜어낼 수 있었다. [그림-17]은 베이스 트롬본의 음량 값에 의해 이미지의 객체가 나타나는 개수를 나타낸 것이다.



[그림-17] 베이스 트롬본의 음량 값에 의한 이미지의 개수

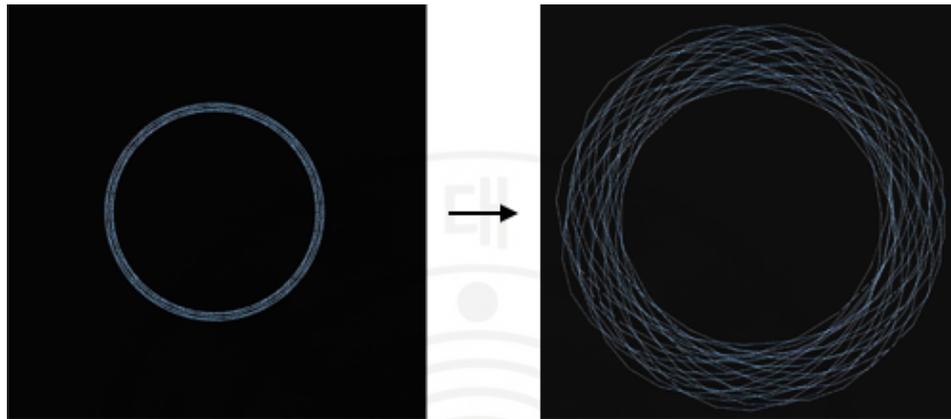
피아노는 88개의 건반으로 이루어져 있다. 따라서 곡의 구성에 따라 FFT분석을 통한 pfft~ xover~오브젝트를 활용하여 2개의 음역으로 나누었고 [표-2]는 곡의 구성에 따른 crossover frequency를 나타낸 값이다.

<표-2> 피아노의 작품 구성에 따른 crossover frequency

time	0:00~ 2:26	2:26~ 3:58	3:58~ 4:43	4:43~ 5:32	5:32~ end
crossover frequency	800Hz	400Hz	440Hz	130Hz	1100Hz

피아노의 음량 값을 통해 Processing 영상에서 변하는 요소는 크기와 선의 개수다. 저음역의 음량 값과 고음역의 음량 값은 각각 이미지의 크기와 선의 개수 변화만을 표현하지만, 베이스 트롬본과 다르게 단조롭지 않았다. 왜냐하면 피아노를 표현하는 영상 이미지가 독자적인 역할도 하지만, 베이스 트롬본을 표현하는 영상 이미지를 보완해주는 역할도 하며 이미지의 형태가 작품의

기승전결에 따라 8가지의 모양이 생성되기 때문이다. 기승전결에 따른 피아노의 이미지 변화는 연구 기술의 작품 적용에서 다루겠다.



[그림-18] 피아노의 음량 값에 따른 이미지의 변화

[그림-18]을 보면 원 자체의 크기와 선들이 넓어진 것을 볼 수 있다. 저음역의 역할은 음량 값에 의한 원의 크기를 나타내고, 고음역의 역할은 원을 둘러싸고 있는 선들의 개수이다.

### 3) Arena를 이용한 영상효과

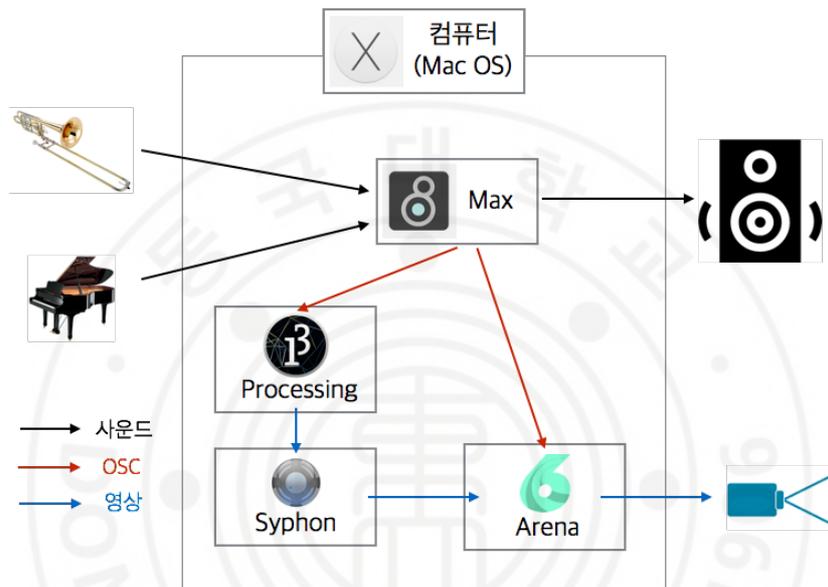
Processing을 이용해서 영상을 제작했지만, 영상의 역동성 및 극적인 효과에 대해서는 한계가 있었다. 또한 각각의 이미지로 사용된 Processing 소스를 하나의 영상으로 만들 수 있는 방법이 필요했다. Syphon을 통해 Processing의 영상을 전송하였으며, 사용된 Arena의 내장효과는 다음 <표-3>과 같다.

<표-3> Arena의 영상효과

영상효과	효과 설명
delay RGB	객체의 윤곽을 RGB로 나눠 시간차에 의해 나타나는 효과
edge detection	영상의 경계선 및 윤곽을 잡아내는 효과
hue rotate	hue 값을 바꿔주는 효과
goo	영상을 울렁이게 하는 효과
blur	특정 영역을 흐리거나 번지게 하는 효과
radialblur	blur 종류의 아웃포커싱 효과
bright.contrast	명암 및 밝기 대비 조절 효과
tint	특정 영역의 색감을 바꿔주는 효과

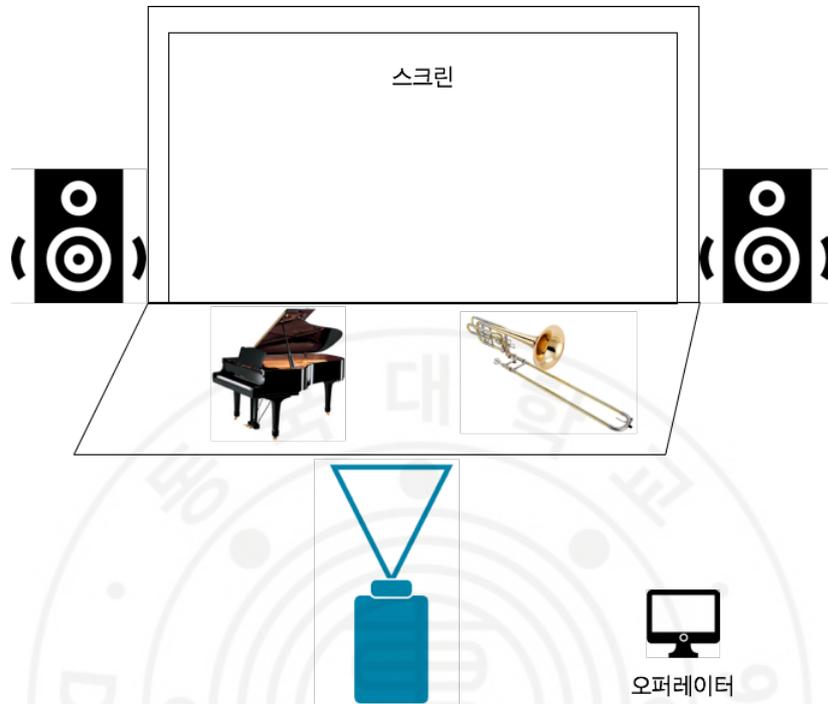
### 3. 공연 및 연동 시스템 연구

#### 1) 공연 시스템



[그림-19] 공연 시스템 설계도

[그림-19]는 무대에서 사용되는 사운드와 영상의 총 시스템 설계도이다. 베이스 트롬본에 사용되는 마이크는 AKG사의 C414이며, 모노(mono)로 받았다. 피아노에 사용되는 마이크는 DPA사의 DPA-4021 제품을 페어(Pair)로 사용하여, 스테레오(stereo)로 받았다. 3개의 채널을 오디오 인터페이스를 통해 입력 받았으며, Max에서 사운드 프로세싱을 거쳐 스피커로 출력되었다. Max에서 출력된 OSC 데이터는 각각 Processing과 Arena에 입력되고, Processing에서 Syphon을 통해 Arena로 영상 프레임이 전달되어서 영상효과를 입힌 후에 프로젝터로 출력되었다.



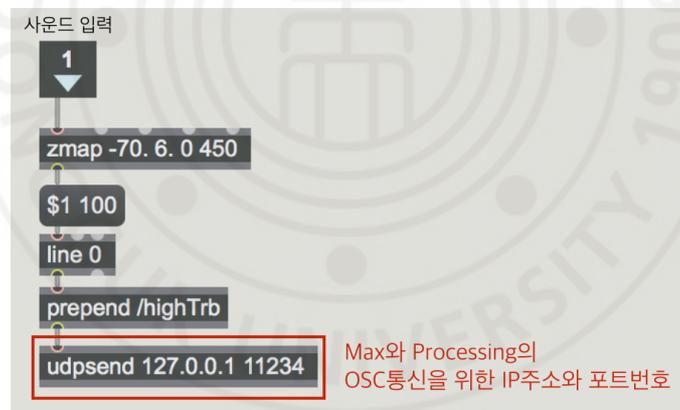
[그림-20] 무대 구성

[그림-20]은 작품의 무대 구성이다. 피아노는 마이크에서 베이스 트롬본의 소리를 최대한 덜 수음하기 위해서 왼편에 두었다. 오퍼레이터는 객석의 오른쪽 뒤에 위치하고 있으며, 사운드와 영상을 제어하며, 사운드를 모니터하여 공연장에서의 사운드 밸런스를 맞추었다.

## 2) 음악과 영상의 연동 시스템

### ① OSC 통신을 이용한 음악과 영상의 연동

Processing을 통한 음악과 실시간으로 연동되는 영상을 제작할 때, 음악에서 보내주는 특정한 정보가 있어야 한다. 그러기 위해서 사운드를 담당하는 Max에서 어떠한 수치를 입력받아야 한다. 하지만 Max와 Processing은 서로 다른 프로그램이기 때문에 같은 컴퓨터 안에서 서로 연동시켜주는 통신이 있어야 한다. 따라서 두 프로그램의 연동을 위해 OSC 통신을 사용하였으며, Max에서 입력받은 사운드 데이터를 OSC 데이터로 변환하여 Processing으로 연동되는 수치를 전달하였다.



[그림-21] Max와 Processing의 OSC 통신을 위한 Max 패치

[그림-21]을 보면 Max와 Processing을 연동할 때, udpsend오브젝트를 사용한다. UDP<sup>55)</sup>방식의 통신 프로토콜을 사용하며, IP주소<sup>56)</sup>와 포트번호<sup>57)</sup>를

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

지정해주어야 한다. IP주소는 같은 컴퓨터 안에서 통신을 하면 되므로, localhost 혹은 127.0.0.1로 지정한다. 포트번호는 통신하려는 프로그램끼리 같은 번호를 설정하면 된다.

Processing에서 Max와 OSC 통신을 하려면 Contribution Manager에 있는 OSC 라이브러리를 설치해야 한다.



[그림-22] Processing에서 OSC 라이브러리를 받는 모습

OSC 라이브러리를 받은 후에는 통신을 할 수 있게 명령어를 입력해야한다. [그림-23]은 Processing에서 OSC 라이브러리를 불러오는 코드를 입력한 것이다. Processing은 위에서부터 아래로 순차적으로 입력된 코드를 읽기 때문에, 첫 번째 단에 입력해야 한다.

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

OscP5 oscP5;
NetAddress myRemoteLocation;
```

[그림-23] OSC 라이브러리를 불러오는 코드

- 
- 56) Internet Protocol address의 약어. 컴퓨터 네트워크에서 장치들이 서로를 인식하고 통신을 하기 위해서 사용하는 특수한 번호이다.
  - 57) 데이터의 입출력을 위한 통로.

[그림-24]는 불러온 OSC 라이브러리를 설정하는 코드이다. void setup은 Processing을 실행하였을 때, 처음으로 한 번만 실행하는 함수이다. Max와의 통신을 위해 IP주소를 지정해주며, 포트번호 또한 함께 설정을 해주어야 한다.

```
void setup() {
  size(1920, 1080, P2D);
  oscP5 = new OscP5(this, 11234);
  myRemoteLocation = new NetAddress("127.0.0.1", 11234);
}
```

[그림-24] OSC 통신을 하기 위한 주소 설정 코드

예를 들어, Max에서 4개의 수치를 Processing으로 전송해야 한다면 각각 인식할 수 있는 이름이 있어야 한다. Processing에서는 변수로 받아들여지며, Max에서 prepend오브젝트<sup>58)</sup>를 사용하여 변수 이름을 설정함과 동시에 [그림-25]의 코드를 입력해 주어야 한다. 그리고 전송하는 데이터의 형태를 지정해주어야 한다. int<sup>59)</sup> 정보를 보내려면 intValue라고 입력해야 하고, float<sup>60)</sup> 정보를 보내려면 floatValue를 입력해야 Processing에서 데이터 정보 처리에 따른 오류가 발생되지 않는다.

```
void oscEvent(OscMessage myMessage) {
  if (myMessage.checkAddrPattern("/highTrb") == true) {
    //println(myMessage.get(0).floatValue());
    r1 = myMessage.get(0).intValue();
  }
}
```

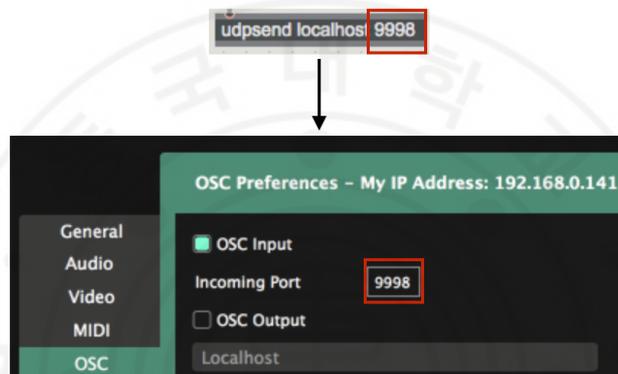
[그림-25] OSC 입력 데이터를 변수로 설정하는 코드

58) 전송하려는 메시지 앞에 다른 메시지를 추가로 입력해주는 오브젝트.

59) integer의 약자이며, 정수를 의미한다.

60) 부동소수점을 의미한다.

Max에서 Arena와의 연동을 위한 방법은 다음과 같다. IP주소와 포트번호를 지정해야 하며 IP주소는 같은 컴퓨터에서 통신을 하기 때문에 127.0.0.1 혹은 localhost를 입력한다. 그러나 포트번호는 Max와 Processing을 연동시키려 사용된 번호를 제외한 다른 번호를 사용해야 한다. Max와 Arena의 포트번호를 일치시킨 것을 [그림-26]에서 볼 수 있다.



[그림-26] OSC 통신을 이용한 Max와 Arena의 연동

## ② Syphon을 이용한 Processing과 Arena의 연동

OSC 데이터는 수치를 전달할 수는 있지만, 영상 프레임을 전달하지는 못한다. 또한 실시간으로 영상이 프로세싱 되기 때문에, 렌더링 (rendering)<sup>61)</sup>을 할 수 없다. 따라서 2개의 Processing 화면을 Arena로 전달할 수 있는 방법은 다음과 같다.

Syphon을 이용해서 실시간으로 만들어지고 있는 Processing의 영상을 공유해주는 것이다. Processing에서 Syphon을 사용하려면, Contribution Manager에 있는 Syphon 라이브러리를 설치해야 한다.

61) 수식이나 수치로 서술된 2차원 혹은 3차원 데이터를 사람이 인지할 수 있는 영상으로 변환하는 과정을 말한다.

```

import codeanticode.syphon.*;
SyphonServer server;

void setup() {
  size(1920, 1080, P2D);
  server = new SyphonServer(this, "Processing Syphon");
}

void draw() {
  background(127);
  rect(30, 20, 55, 55);
  server.sendScreen();
}

```

[그림-27] Processing에서 Syphon을 사용하는 코드

Syphon 라이브러리를 불러오기 위해 첫 번째 단에 입력을 하며, void setup함수에서 서버를 생성해준다. void draw는 입력된 코드를 반복적으로 실행하는 함수이다. 실시간으로 생성되는 화면을 계속 전송해야 하므로, void draw 함수에 스크린으로 영상을 보내는 코드를 입력한다.

Arena에서도 Syphon을 통해 공유되는 영상소스를 받아야 한다. [그림-28]은 Arena에서 Syphon을 통해 공유된 영상소스를 받는 장면이다.



[그림-28] Syphon을 통해 공유된 영상소스

Processing Syphon이라는 서버 이름을 지정해주었고, Processing 파일에는 혼동이 가지 않게 약기별 이름을 지정하여 사용하였다.

### III. 연구 기술의 작품 적용

연구된 베이스 트롬본과 피아노의 실시간 사운드 프로세싱을 이용한 멀티미디어작품 <Yes, I am.>은 2018년 11월 10일 동국대학교 이해량 예술극장에서 진행된 'SEEING SOUND LISTENING IMAGE(보는 소리, 듣는 영상) XV' 공연에 초연되었다.

#### 1. 작품 소개

르네상스, 바로크, 고전, 낭만, 20세기 등의 시대적 흐름에 따라 분류된 음악 양식은 새로운 양식으로 발전될 때 옛 것과 새 것이 서로 공존하는 과도기를 거친다. 현재까지도 20세기 이후의 전위적인 음악 양식이 전해져오고 있지만 난해한 음악성과 실험주의적인 작곡기법으로 인해 일반 대중들에게 큰 공감을 얻지 못하고 있으며, 과학 기술과 예술이 결합된 멀티미디어 공연 또한 일반 대중들이 쉽게 이해할 수 있는 작품은 많지 않다.

따라서 본 연구자는 작품 <Yes, I am.>을 통해 20세기 이후의 비조성적인 음악의 느낌과 고전주의의 조성음악적인 느낌을 서로 다른 시대처럼 느끼게 하는 것이 아닌 하나의 음악으로 자연스럽게 혼합하여 공감을 얻고자 한다.

<Yes, I am.>의 주제는 자신의 본질과 자아에 대해 고민을 하며 항상 자기 자신에게 질문을 던지지만, 해답을 찾지 못한다. 결국 '나는 어떤 사람인가?'에 대한 해답은 결국 자신에 대해 고민하고 방황하는 모습 또한 나 자신이라는 것을 깨닫게 되는 내용이다.

## 2. 작품 구성

### 1) 음악 구성

본 작품은 크게 Section-I과 Section-II로 나누어진다. Section-I에서는 피아노에서 증 4도 위에 non-diatonic chord<sup>62)</sup>를 연속적으로 배치하여 비조성적인 느낌을 내었다. 트롬본은 C음을 주기적으로 내었고, 스타카토와 글리산도 주법을 주로 연주했다. 반면에 Section-II에서는 피아노를 화성적으로 진행하고 그 위에 베이스 트롬본을 선율적으로 연주하는 호모포니 구조를 통해 Section-I과는 다른 느낌을 주었다.

Section-I과 II의 공통점으로는 베이스 트롬본이나 피아노 두 악기 중에 한 악기는 반드시 C음을 주기적으로 연주하게 하는 것이다. Section-I은 주기적으로 연주하는 C음위에 non-diatonic chord를 배치하여 비조성적인 느낌을 내었다면, Section-II에서는 주기적으로 연주하는 C음위에 diatonic chord를 배치하여 조성적인 느낌을 내었다. 하지만 다른 시대적 느낌을 자연스럽게 연결하는 방법이 필요했다. Section-I에서 중간에 한 번씩 C음위에 diatonic chord를 섞었고, 비조성적인 느낌에서 조성적인 느낌으로 점차적으로 변화시켰다. 그 결과, Section-II의 조성적인 구조로 자연스럽게 진행이 되었다.

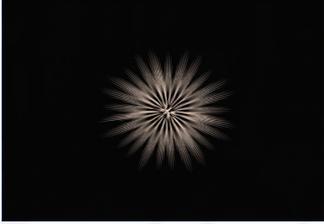
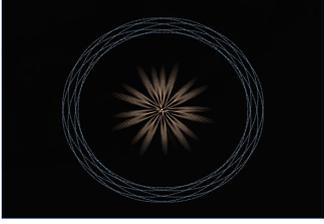
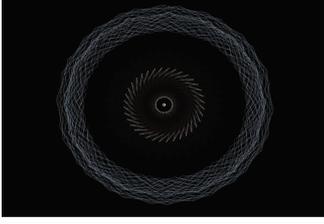
---

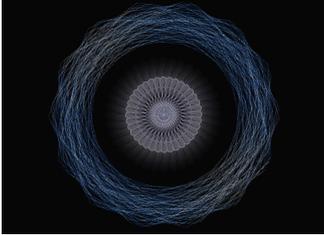
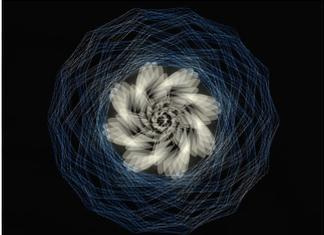
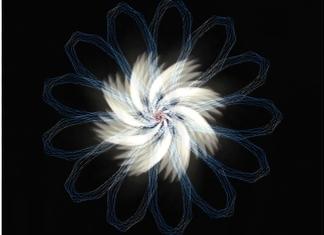
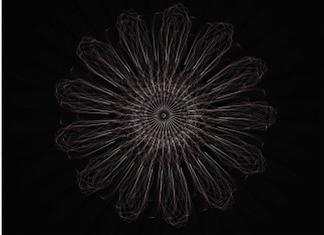
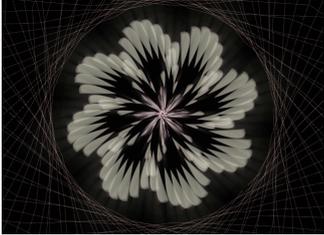
62) 한 조(key)의 음계에서 생성되지 않는 화음을 말한다.

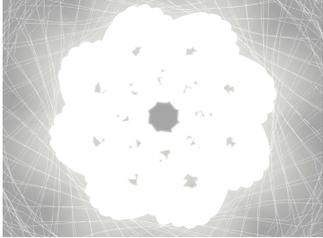
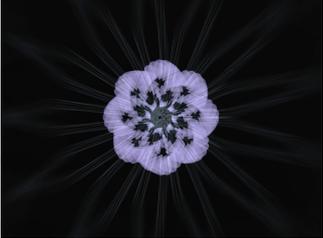
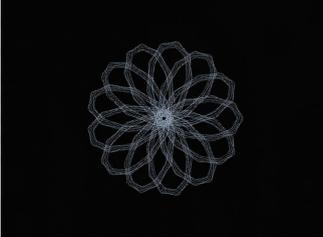
## 2) 영상 구성

영상은 Section-I과 II로 나누어지는 음악과는 다르게 12개의 신(scene)으로 나누었다. 두 악기를 두 개의 오브젝트로 표현할 때, 각각의 오브젝트들의 변화가 없으면 단조롭다. 따라서, 형태의 변화를 통해 이를 해결하였으며, 이 변화 과정을 하나의 이야기로 풀었다. 베이스 트롬본을 ‘자신’으로, 피아노를 ‘자신을 둘러싸고 있는 생각’으로 표현했으며, 두 개의 오브젝트가 결국 하나의 이미지가 되어 씨앗-꽃-만개-씨앗의 과정을 보여주었다. <표-4>는 진행되어지는 영상의 시나리오이다.

<표-4> 영상 시나리오

장면 및 설명	이미지
#1. 0:00 ~ 0:32 하나의 씨앗이 만들어 진다.	
#2. 0:32 ~ 1:13 씨앗을 둘러싸고 있는 하나의 껍질이 생긴다.	
#3. 1:13 ~ 1:54 감싸고 있던 껍질이 분리되기 시작한다.	

<p>#4. 1:54 ~ 2:30          껍질이 분리되어 하나의 형태를 만들기 시작한다.</p>	
<p>#5. 2:30 ~ 3:06          껍질의 분리가 끝나가고 하나의 형태를 만든다.</p>	
<p>#6. 3:06 ~ 4:00          꽃이 된다.</p>	
<p>#7. 4:00 ~ 4:46          꽃잎이 흔들린다.</p>	
<p>#8. 4:46 ~ 5:22          산개하여 펼쳐진다.</p>	

<p>#9. 5:22 ~ 5:27 모두 날아가 버린다.</p>	
<p>#10. 5:48 ~ 6:17 날아가 버린 것들이 다시 모인다.</p>	
<p>#11. 6:17 ~ 6:40 껍질이었던 것이 본질이 된다.</p>	
<p>#12. 6:40 ~ end 본질이 된 것 또한 사라진다.</p>	

‘자신’이라는 이미지가 ‘생각’이라는 이미지를 마주하게 되고, 둘이 합쳐져 하나의 형태를 이루게 된다. 하나의 형태가 된 두 이미지는 꽃이 되고 흩날려지며 다시 모인다. 결국은 ‘생각’이라는 이미지가 본질이 되고 그 본질 또한 사라져버리는 이야기를 영상으로 전개했다.

### 3. 작품 내 사운드 및 영상 기술 적용

#### 1) 각 악기별 사운드 적용 효과

##### ① 베이스 트롬본의 사운드 적용 효과

연구한 음향효과들은 한 가지의 음향효과만 재생되는 것이 아닌, 여러 가지 조합으로 재생된다. <표-5>는 베이스 트롬본에 사용된 음향효과들의 조합을 나타낸 것이다.

<표-5> 베이스 트롬본에 사용된 음향효과들의 조합

악기	음향효과 조합
베이스 트롬본	xover filter + comb filter
	ring modulation + flanger
	granular synthesis + delay + reverb

comb filter를 사용한 음향효과 조합은 다음과 같다. 먼저 xover filter를 통해 500Hz 이상의 소리만 전달받은 후에 comb filter 음향효과를 적용하였다. comb filter는 금속성의 날카로운 소리가 나는 것이 특징이지만 베이스 트롬본 소리에 그대로 입히면 곡 전체의 저음 비율이 많아지게 되어 지저분해진다. 따라서 베이스 트롬본의 500Hz이상의 소리만 comb filter를 적용시키고 1초 뒤에 사운드가 들리도록 하였다. 그 결과, 베이스 트롬본이 연주하게 되면 1초 후에 연주된 소리가 날카로운 음향효과를 입힌 상태로 재생되어 효과적인 사운드 프로세싱을 할 수 있었다.

ring modulation에는 flanger 음향효과를 같이 사용하였다. ring modulation을 사용한 이유는 음을 빠르게 진동하게 하여 사람이 물리적으로 낼 수 없는 효과를 표현하기 위해서다. 그래서 조금 더 이질적인 느낌을 주어 관객들에게 자연스러운 소리가 아님을 의도하기 하기 위해 flanger를 사용했다. flanger 음향효과를 입히게 되면 마치 비행기가 이륙하는 소리와 같은 울림을 만들어 낸다.

granular synthesis는 음원을 짧은 시간 안에 많은 조각들로 내어 소리가 끊기게 들리는 현상이 나타난다. delay와 reverb 음향효과를 입혀 잘게 쪼개진 음원들을 울리게 만들어 마치 하나의 음 덩어리를 표현했고, 하나의 악기로 웅장한 사운드를 만들어 낼 수 있었다. 따라서 베이스 트롬본 한 대가 연주하지만, 마치 금관악기군이 연주하는 것처럼 표현했다.

## ② 피아노의 사운드 적용 효과

<표-6> 피아노에 사용된 음향효과들의 조합

악기	음향효과 조합
피아노	phase vocoder + delay
	granular synthesis(low grain pitch) + delay + reverb
	granular synthesis(high grain pitch) + delay + reverb

<표-6>은 피아노에 사용된 음향효과들의 조합이다. 먼저 phase vocoder를 사용하여 연주되는 음을 거꾸로 재생시켰다. 그 결과 attack<sup>63)</sup>이 나중에 들리게 되어 연주한 음이 연장되는 듯한 효과를 주었으며, 그로 인해 풍부한 음향효과를 기대할 수 있었다. 또한 만들어진 효과에 delay 음향효과를 추가하여 연장된 음을 더 지연시켜 주었다. 왼쪽과 오른쪽의 delay time을 다르게 주어 phase vocoder의 소리를 효과적으로 적용할 수 있었다.

다음 조합으로 delay와 reverb의 음향효과를 입힌 granular synthesis 2개를 사용하여 패드 사운드를 만들었다. 2개의 granular synthesis의 차이점은 grain pitch에 있다. 하나의 grain pitch가 2라면 나머지 효과의 grain pitch는 4이다. 즉, 저음이나 고음을 표현할 때 훨씬 더 많은 배음을 표현할 수 있었다.

---

63) 음이 발생하여 최고 음량까지 도달하는 시간을 뜻한다.

## 2) Section-I 적용 효과

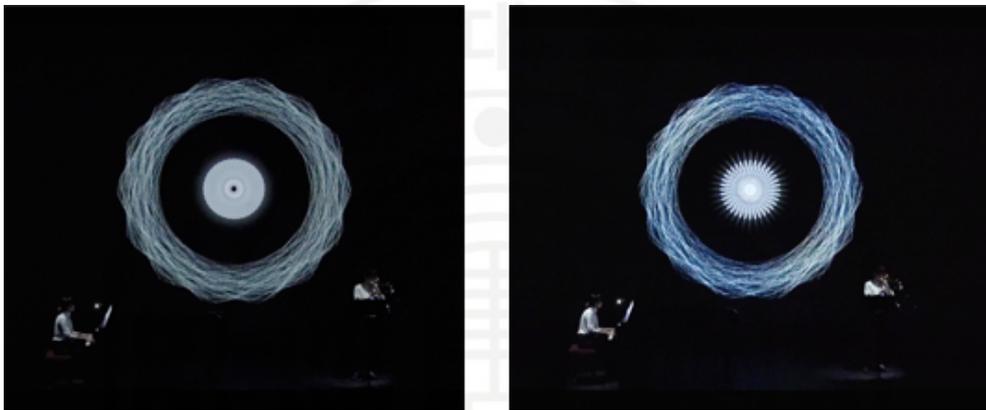
<표-7> Section-I에 적용된 사운드 프로세싱 및 영상효과

악기	베이스 트롬본	피아노
시간	0:00 ~ 3:59	
장면	#1 ~ #6	
사운드 프로세싱	comb filter ring modulation granular synthesis (grain pitch=0.5)	phase vocoder (playback rate=-0.8) granular synthesis (grain pitch=0.25, 0.5)
영상효과	radialblur edge detection goo hue rotate	bright.contrast

Section-I에서는 피아노와 베이스 트롬본의 현대적인 연주기법을 사용하였으며, granular synthesis의 grain pitch를 두 악기 모두 저음으로 사용하여 불안한 분위기를 내었다. 베이스 트롬본에는 comb filter를 사용하여 연주되는 스타카토 음을 1초 뒤에 프로세싱되게 하였고, 스타카토를 연주하지 않은 긴 음들은 ring modulation을 사용하여 음의 떨림을 주었다. 피아노는 phase vocoder의 playback rate를 -0.8로 지정하여 연주하는 음의 길이보다 느리게 거꾸로 재생하였다.

베이스 트롬본을 표현하는 이미지에 대한 영상효과는 전반적으로 radialblur 효과를 사용하여 영상이 퍼지게 보이는 효과를 주었고 ring modulation이 나올 때마다 goo 효과를 사용하였다. 피아노를 표현하는 이미지는 선으로만

이루어져있어 검은색과 흰색의 조합을 사용하였지만 저음의 granular synthesis 음량이 커질수록 음악에서의 저음 비중이 많아지다 보니 차가운 느낌의 영상표현이 필요했다. 따라서 저음의 차가운 느낌의 표현을 위해 bright.contrast 효과를 적용시켰다. [그림-29]는 피아노에 입힌 granular synthesis의 음량에 의한 bright.contrast 효과의 변화이다.



[그림-29] bright.contrast 효과에 의한 영상의 변화

### 3) Section-II 적용 효과

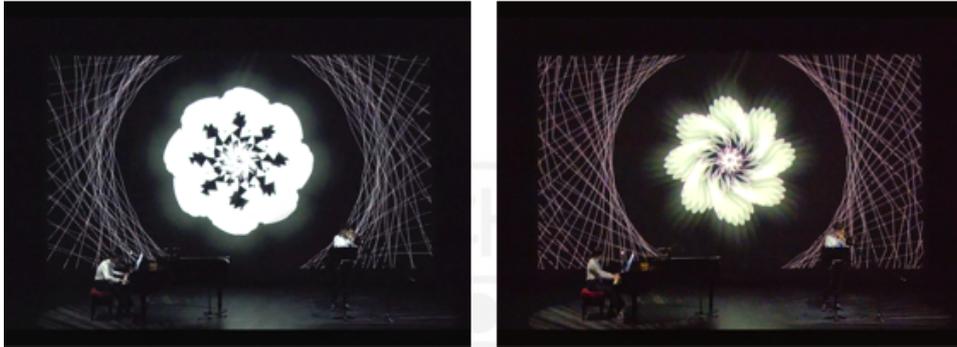
<표-8> Section-II에 적용된 사운드 프로세싱 및 영상효과

악기	베이스 트롬본	피아노
시간	3:59 ~ 7:00	
장면	#7 ~ #12	
사운드 프로세싱	ring modulation granular synthesis (grain pitch=2)	phase vocoder (playback rate=-1.3) granular synthesis (grain pitch=2, 4)
영상효과	radialblur edge detection goo delay RGB	tint blur radialblur

Section-II에서는 화성적인 반주를 하는 피아노와 선율을 연주하는 트롬본의 형태를 가지는 호모포니 작곡기법을 사용하였다. 두 악기 모두 granular synthesis의 grain pitch를 고음으로 사용하여 상승하는 듯한 음악적 효과와 밝은 분위기를 표현하였다. 베이스 트롬본은 ring modulation을 사용하여 음의 빠른 떨림을 표현하였고, 피아노는 phase vocoder의 playback rate를 -1.3으로 지정하여 연주하는 음의 길이보다 빠르게 reverse 하였다. 클라이맥스부분에는 극적인 분위기를 위해 두 악기에 적용한 granular synthesis를 모두 왼쪽과 오른쪽으로 번갈아가며 재생하게 하였다.

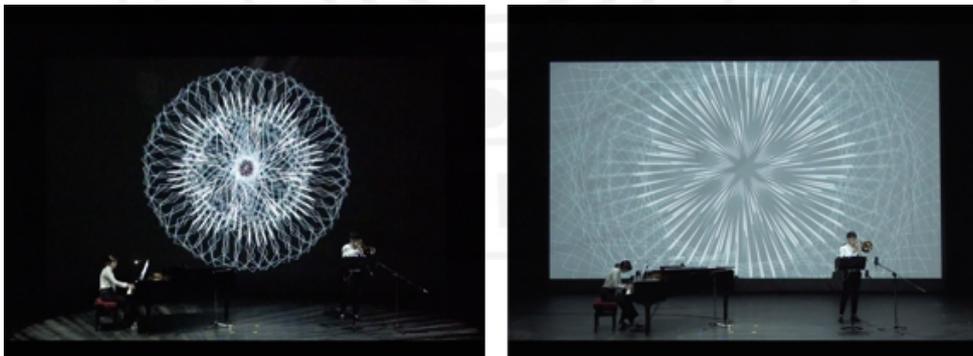
granular synthesis의 grain pitch를 고음으로 설정하여 밝은 분위기를 표현하였으며 그로 인해 음악에서 고음 비중이 많아지게 되었다. 따라서

음악에서 고음 비중의 극대화를 위해 tint 효과를 사용하여 붉은 색감을 추가했다. [그림-30]은 tint 효과에 의한 영상의 변화를 나타낸 그림이다.



[그림-30] tint 효과에 의한 영상의 변화

효과적인 클라이맥스를 표현하기 위해서 영상에도 변화를 주었다. 두 악기를 표현하는 이미지가 화면의 중앙으로 모이고 다시 퍼지는 형태를 만들었다. [그림-31]은 클라이맥스 전개에 따른 영상의 변화이다.



[그림-31] 클라이맥스 전개에 따른 영상의 변화

#### 4. 연구 기술의 작품 적용 효과

Max를 통해 프로세싱된 사운드는 두 대의 악기만으로 풍성한 음향효과를 표현할 수 있었다. 베이스 트롬본은 의도적인 음의 떨림을 주어 호흡이 많이 필요한 악기의 특성을 변화시키기에 충분했고, comb filter 음향효과를 고음역에만 사용하여 금속성 물질처럼 날카로운 소리를 낼 수 있게 함으로써 부드러운 악기의 음색을 변화시켜줄 수 있었다. 또한 granular synthesis를 사용하여 저음과 고음을 보강시킨 결과, 웅장한 사운드를 만들어 낼 수 있었다. 피아노는 연주되는 음을 reverse시켜 attack이 나중에 들리게 되어 음이 연장되는 듯한 효과를 낼 수 있었고, 만들어진 음향효과에 delay 음향효과를 추가하여 음악의 왼쪽과 오른쪽의 사운드를 채울 수 있었다. 또한, granular synthesis를 사용하여 피아노의 저음과 고음을 보강시킴과 동시에 패드 사운드를 만들어 낼 수 있었다. 전자 음향으로 인해 악기의 구조에 대한 한계를 보완할 수 있었고, 이는 곧 작품에서도 어쿠스틱 악기와 혼합하여 다채로운 사운드를 표현할 수 있게 되었다.

음악과 연동시킨 영상은 각 악기를 표현하는 이미지를 만들어 작품의 기승전결에 대한 이해를 도왔다. 악기의 음량 값을 활용하여 영상과 연동시켰으며, 악기가 연주한다는 것을 시각적으로 표현할 수 있었다. 즉, 연주되는 악기를 시각화하여 관객들이 음악과 영상이 상호작용 한다는 것을 느낄 수 있게 하였고 청각적인 측면과 시각적인 측면을 합한 하나의 멀티미디어음악 작품을 만들어 낼 수 있게 되었다.

## IV. 결 론

본 논문은 베이스 트롬본과 피아노의 악기를 사용하여 실시간으로 프로세싱된 소리와 그에 반응하는 영상을 같이 보여줄 수 있는 멀티미디어 작품 제작에 대한 연구이다. 두 대의 악기가 가진 오리지널 음색만이 아닌 다채로운 음색을 낼 수 있도록 하는 것과 함께 사운드에 연동되는 영상을 제작하여 청각적인 측면과 시각적인 측면을 합한 하나의 멀티미디어 작품을 제작하고자 하였다.

Max를 사용하여 실시간 사운드 프로세싱에 대한 연구를 하고 제작하였으며 granular synthesis, comb filter, delay, flanger, ring modulation, phase vocoder 음향효과를 각 악기의 특성에 맞춰 조합하였고, 어쿠스틱 악기에서 나올 수 없는 소리들을 전자 음향적으로 만들어 음악에 적용하였다. 그로 인해, 악기의 구조로 인한 특수 주법, 고유의 음역, 음색을 기반으로 써야 했던 작곡 기법에 대한 제한성 및 한계를 벗어날 수 있었으며, 이는 곧 새로운 작곡 기법을 구현할 수 있게 되었다.

영상 또한 모두 Processing을 통해 실시간으로 제작되어지며, 각각의 악기들을 이미지화시켜 관객의 이해를 도왔다. 영상은 각 악기의 음량에 의해 반응하도록 제작되었으며, 두 이미지 또한 영상 시나리오에 맞춰 하나의 이미지로 만들어졌다. 영상 표현의 극대화를 위해 여러 가지 영상 효과를 사용하였으며 만들어진 영상들은 음악과 상호작용한다는 것을 관객들이 느낄 수 있게 하였다. 음악과 영상의 연동 시스템 또한 OSC 통신을 통해 원활하게 제어를 하였으며, 데이터의 지연이나 오류 없이 안정적인 표현이 가능했다.

하지만 이번 연구를 통해 발생한 문제점은 다음과 같다. 첫 번째는 사운드와 영상의 실시간 프로세싱에 대한 자동화이다. 실시간으로 제작되는 음향효과와 영상 및 영상효과를 공연장에서 모두 제어하기에는 위험부담이 높다.

오퍼레이터가 실수를 할 수 있는 위험 요소를 최대한 줄이고, 실시간으로 제작되어지는 요소들 또한 자동화 시스템으로 만들어 최대한 안정적으로 공연할 수 있는 것 또한 필요하다. 음악과 영상의 상호작용을 통한 멀티미디어 작품을 만들어 낼 때, 발생할 수 있는 위험요소들을 줄여 안정적인 시스템을 구축하는 것에 대한 연구가 필요하다.

두 번째는 악기의 음량 값만 받아서 영상과 연동시킨 것이다. 악기의 음량 값만이 아닌 음정, 화성, 연주되는 음의 빠르기, 연주자의 모션 등을 인터랙션 요소로 활용하기 위한 연구가 필요하다. 악기의 음량만으로는 다채로운 표현을 하는 것에 있어 한계점이 있으며, 다양한 요소들을 음악과 연동시킬 수 있는 방법에 대한 추가적인 연구 또한 필요하다.

Keyword(검색어)

컴퓨터음악(computer music), 소리시각화(sound visualization)  
인터랙티브 멀티미디어 음악(interactive multimedia music), Max,  
실시간 사운드 프로세싱(real-time sound processing), Processing

E-mail: ghksgml12@naver.com

## 참 고 문 헌

### 1. 단행본

- Daniel Shiffman 저 「러닝 프로세싱」, (비제이퍼블릭, 2016)
- Samuel Adler 저 「관현악기법연구」, (수문당, 2009)
- 김영민, 「사운드 디자인을 위한 맥스」, (Real Lies Media, 2017)
- 이석원, 「음악음향학」, (심설당, 2003)
- 홍세원, 「서양음악사」, (연세대학교 대학출판문화원, 2001)
- Charles Dodge, Thomas A. Jerse, 「Computer Music: synthesis, composition and performance, Second Edition」, (Schirmer Books, 1997)
- Bob Katz, 「Mastering Audio the art and the science, third edition」, (Focal Press, 2007)
- Curtis Roads, 「The Computer for music」, (MIT Press, 1996)

## 2. 참고논문

- 강현우, 「인도음악 연구를 통한 인터랙티브 멀티미디어음악 제작 연구」 (동국대학교 영상대학원 멀티미디어학과, 2017)
- 선현재, 「트롬본의 역사와 기초 주법에 대한 연구」 (단국대학교 교육대학원 교육학과, 2015)
- 이도경, 「피아노 연주를 통한 실시간 오디오-비주얼 작품 제작 연구」 (동국대학교 영상대학원 멀티미디어학과, 2018)
- 이보강, 「피아노의 실시간 사운드 프로세싱을 이용한 인터랙티브 멀티미디어 퍼포먼스 연구」 (동국대학교 영상대학원 멀티미디어학과, 2018)
- 전우진, 「컴퓨터음악과 phase music을 이용한 인터랙티브 멀티미디어 퍼포먼스 연구」 (동국대학교 영상대학원 멀티미디어학과, 2017)
- 한승욱, 「피아노의 실시간 프로세싱을 이용한 멀티미디어음악 제작 연구」 (동국대학교 영상대학원 멀티미디어학과, 2018)

### 3. 웹사이트

- CNMAT: external Max object, OSC  
<http://cnmat.berkeley.edu/>
- Learning Processing  
<http://learningprocessing.com/>
- Max  
<https://cyclling74.com/>
- Processing  
<https://processing.org/>
- Syphon  
<http://syphon.v002.info/>

## ABSTRACT

A Study on the Production of Multimedia  
Music using Real-time Sound Processing  
for Bass trombone and Piano  
(focus on Multimedia Music <Yes, I am.>)

Jo, Hwan Hee

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

The work <Yes, I am.> is a multimedia music piece that can be performed real-time processing sounds and videos of responses to instruments. This work focuses on making different types of music by transforming the original sound of acoustic instruments in real-time, and visualizing the music through the interaction of video that responds to the music in real-time to make it into a multimedia production.

Real-time sound processing of the bass trombone was used with flanger, granular synthesis, comb filter, and ring modulation. Also real-time sound processing of the piano was used with phase vocoder, delay, and granular synthesis. All mentioned processed sound effects were produced through Max and were mapped to the controller for each processed sound effect.

In order to create a work with an interaction between music and video, specific information figures must be delivered and responded by the numerical value. Processing program was used to link with Max and OSC, and produced the music by linking the instrument volume values. Syphon was used to share the videos of the other two Processing sketches, and Arena was used to mix the videos and apply video effects.

As a result, the audience could feel that the created videos interact with the music. The interaction system of music and video was also well controlled through OSC communication, allowing stable performance without delay or error of data.

# 부록-1 : 작품 <Yes, I am.> 악보

Yes, I am.

Composed by. Hwanhee Jo

♩=60

Bass Trombone

*p* *mf* *f* *sfz*

Piano

7

B. Tbn.

*sfz* *p* *mp*

Pno.

*p* *mf*

14

B. Tbn.

*fp* *ff*

Pno.

*f*

19

B. Tbn.

*fp* *f* *f* *fp*

Pno.



39

B. Tbn. *ff* *f* *p*

Pno. *f*

42

B. Tbn. -

Pno. *f* *sfz* *ff* *mf*

45

B. Tbn. *f* *fp*

Pno. *f* *f*

♩ = 40

48

B. Tbn. *fp*

Pno. *mf* *f*





84

B. Tbn.

Pno.

86

B. Tbn.

Pno.

88

B. Tbn.

Pno.

*ff*

90

B. Tbn.

Pno.

92

B. Tbn. *f* *cresc.* *fff*

Pno. *f* *cresc.* *fff*

95

B. Tbn. *p* *f*

Pno.

100

B. Tbn. *mf* *f* *mf*

Pno. *p* *mf*

105

B. Tbn. *p*

Pno. *f*

110

B. Tbn.

Pno.

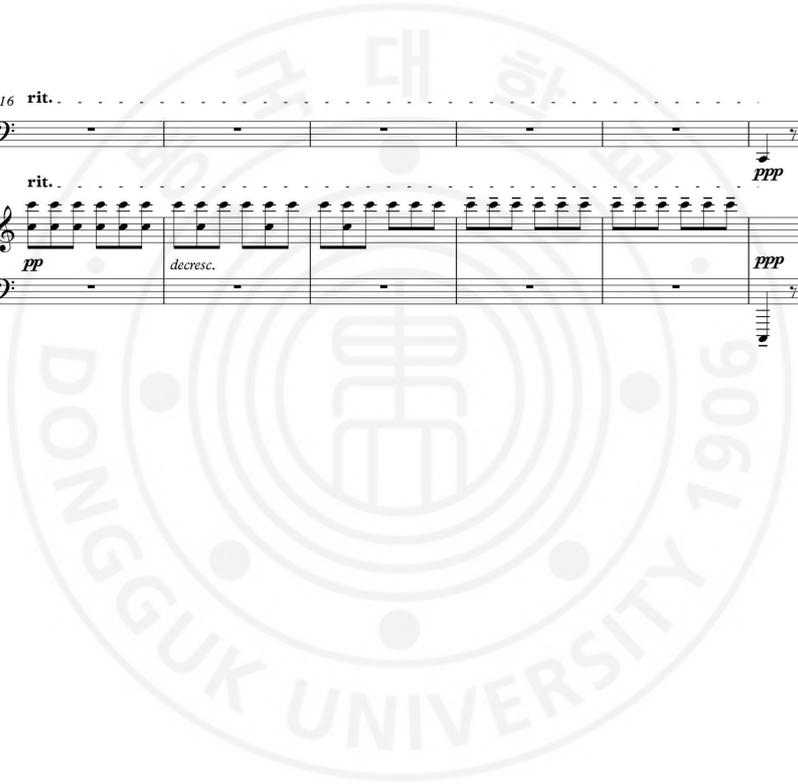
*mf* *p*

116 rit.

B. Tbn.

Pno.

*pp* *decresc.* *ppp*



## 부록-2 : 첨부 DVD

1. Yes, I am. 공연 영상 : 2018년 11월 10일 이해랑 예술극장  
공연 영상
2. Yes, I am. score : 작품 악보 폴더
3. Yes, I am. patch : 작품에 사용된 Max 패치 폴더

