

석 사 학 위 논 문

Max/MSP/Jitter를 이용한 기타 이펙터

제작과 실시간 소리 시각화 연구

(멀티미디어음악작품 <What color rose were you?>를 중심으로)

지도교수 김 준

동국대학교 영상대학원

멀티미디어학과 컴퓨터음악전공

나 준 하

2 0 1 5

석사학위논문

Max/MSP/Jitter를 이용한 기타 이펙터

제작과 실시간 소리 시각화 연구

(멀티미디어음악작품 <What color rose were you?>를 중심으로)

나 준 하

지도교수 김 준

이 논문을 석사학위논문으로 제출함.

2015년 1월 일

나준하의 음악석사학위(컴퓨터음악전공) 논문을 인준함.

2015년 1월 일

위원장: 정진헌 (인)

위원: 박상훈 (인)

위원: 김준 (인)

동국대학교 영상대학원

## 목 차

I. 서론 .....	1
1. 연구배경 .....	2
2. 연구목적 .....	3
II. 본론 .....	5
1. 작품의 구성 .....	5
1) 작품의 구조 .....	5
2) 작품을 위한 시스템 구성 .....	6
① signal flow .....	6
② 사운드의 wet & dry .....	7
2. 기술적 연구 .....	8
1) 사운드 제작 .....	8
① 테이프 음악 제작 .....	8
② 그래놀러 합성방식을 이용한 이펙터 제작 .....	9
③ pitch shifter의 제작 .....	11
④ feedback looper의 제작 .....	15
2) 영상 제작 .....	21
① 오디오의 시각화 .....	21

② ‘rose(장미)’의 시각화 .....	25
③ 피치 디텍션(pitch detection)의 활용 .....	31
3) 실시간 제어를 위한 auto-cue의 제작 .....	32
3. 연구기술의 작품 적용 .....	34
1) A파트에서의 기술적용 .....	34
① 작품의 내용 .....	34
② 적용 기술 및 표현 .....	35
2) B파트에서의 기술적용 .....	43
① 작품의 내용 .....	43
② 적용 기술 및 표현 .....	43
3) C파트에서의 기술적용 .....	45
① 작품의 내용 .....	45
② 적용 기술 및 표현 .....	46
Ⅲ. 결론 .....	52
참고문헌 .....	54
Abstract .....	57
부록: 첨부DVD설명 .....	58

## 표 목 차

[표-1] 작품의 구조 .....	5
--------------------	---

## 그 립 목 차

[그림-1] 페달 파라미터를 가진 FX계열 스텝박스 .....	2
[그림-2] 클럽공연에서의 소리 시각화 .....	3
[그림-3] signal flow .....	7
[그림-4] 사운드 시스템의 구성 .....	8
[그림-5] 기본적인 munger~패치 .....	10
[그림-6] ping pong delay 패치 .....	11
[그림-7] 기본적인 모습의 피치 쉬프터 .....	12
[그림-8] 선형의 파형을 비선형으로 변환하는 과정 .....	13
[그림-9] 시작점이 다른 패치와 합쳤을 때 파형의 모습 .....	14
[그림-10] 최종적인 피치 쉬프터의 패치 .....	15
[그림-11] 기본적인 feedback looper의 패치 .....	17
[그림-12] feedback looper의 하울링문제를 해결하는 패치 .....	18
[그림-13] BPM to milliseconds between beats 패치 .....	19

[그림-14] 최종적인 feedback looper 패치 .....	20
[그림-15] jit.slide를 적용한 영상과 적용하지 않은 영상	22
[그림-16] val값에 따른 영상의 밝기 .....	23
[그림-17] 매트릭스의 데이터 공유를 위한 jitter 패치 .....	23
[그림-18] 매트릭스의 공유가 없는 영상 .....	24
[그림-19] 매트릭스가 서로 공유할 때의 영상 .....	25
[그림-20] 폴리곤 숫자에 따른 영상의 차이 .....	26
[그림-21] @poly_mode 사용 전후의 영상차이 .....	27
[그림-22] 텍스처를 입힌 구체와 음성에 반응하는 모습 .....	28
[그림-23] 텍스처 문제를 해결하고 난 후의 장미의 모습 .....	29
[그림-24] jit.dimmap을 활용한 후의 장미의 모습 .....	29
[그림-25] 장미를 구현하는 패치 .....	30
[그림-26] fiddle~ 패치 .....	31
[그림-27] 피크 스크라이프 주법 .....	32
[그림-28] seq를 이용한 auto-cue 패치 .....	33
[그림-29] 장미의 기본위치 .....	35
[그림-30] position의 z값으로 zoom in 효과를 주는 패치	36
[그림-31] x, y 값에 따른 장미의 이동경로와 잔상효과 .....	37
[그림-32] 라이드 섹션에 맞춰 나오는 장미 .....	37
[그림-33] 무작위로 9개의 숫자를 추출하는 urn 패치 .....	38

[그림-34] shelter 효과 .....	39
[그림-35] 심장의 고동을 표현한 영상 .....	40
[그림-36] 원을 그리며 도는 장미의 패치와 그 모습 .....	40
[그림-37] shelter와 로테이션을 이용한 효과 .....	41
[그림-38] 무작위로 만들어지는 영상효과 .....	42
[그림-39] 잔상값에 따른 극단적인 영상변화 .....	43
[그림-40] boids 패치 파라미터값의 변화에 따른 영상효과 ..	44
[그림-41] 작품에서의 boids의 적용 .....	45
[그림-42] color의 RG값을 지속적으로 바꾸어 주는 패치 .....	46
[그림-43] 화면에 그림을 그리는 작은 장미 .....	47
[그림-44] 서로 만나지 못하는 장미 .....	48
[그림-45] 검은 boids를 이용한 효과 .....	49
[그림-46] 폭발 효과를 구현하는 패치와 효과 .....	50
[그림-47] ending 파트 영상효과 .....	51

# I. 서론

## 1. 연구배경

20세기 중반, 블루스와 재즈 기반으로 활동하던 대중음악 뮤지션들은 전자기타의 활용을 통해 새로운 소리의 개발을 활발히 추진하면서 다양한 장르가 탄생하였다. 뮤지션과 엔지니어들의 실험적인 활동과 연구로 탄생한 이펙터<sup>1)</sup>는 오랜 시간이 지난 현재에도 많은 기타리스트들이 애용하고 있다. 이펙터의 발전은 장르의 다양화뿐만 아니라 밴드의 보컬을 지원하는 악기연주자로만 존재하던 기타리스트를 화려한 사운드와 연주를 통해 독자적인 뮤지션으로 발전시키는 큰 계기가 되었다. 악기 회사에 의뢰를 하여 직접 제작(customize)한 이펙터들로 자신만의 독특한 음색을 무기로 음악을 하는 대표적인 뮤지션 <Eric Johnson>(1954-)<sup>2)</sup>은 ‘톤의 마술사’로 불리며, 경음악계에서 독보적이고 개성적인 음색을 내는 기타리스트로 추앙 받고 있다. 현대에 들어서 다양한 이펙터의 조합인 페달보드(pedal board)에 의해 나오는 자기만의 음색은 기타리스트의 지문같이 정체성을 나타내어 주는 역할을 하고 있다. 디지털 기술의 발전과 함께 아날로그 이펙터도 디지털 형식으로 나오기 시작했다. 현재는 디지털 이펙터들을 하나의 페달보드 형식의 기계에 모은 멀티 이펙터(multi-effector)나 컴퓨터상으로 구현된 플러그인(plug-in)<sup>3)</sup>이 등장하여 기타리스트들이 더욱 실험적인 음악을 할 수 있게 되었다.

---

1) 스톱 박스(stompbox), 이펙트 페달 이라고도 불리며, 발로 컨트롤을 한다.

2) 미국의 기타리스트. 당대 최고의 기타리스트 3명이 주최하는 G3콘서트의 일원.

3) 가상 이펙터를 지칭한다.



실험적인 기타 이펙터들의 공통점은 한 가지 음색에 오래 머물러 있지 않고 소리에 지속적인 변화를 준다는 점과 외부 장비로 존재할 때에는 페달을 이용한 파라미터 조절을 사용한다는 점이다. 즉 연주 도중에도 짧은 시간내에 이펙터의 파라미터 변화가 기존의 이펙터들보다 많이 변해야 개성이 살아난다는 특징을 가진다. 이러한 이펙터들을 FX 계열 이펙터라고 하는데 아래의 [그림-1]과 같이 파라미터의 조절에 따라 강조되는 음역대가 달라지며 말하는 효과를 주는 와우페달(wah wah pedal), 기타에서 나오는 음의 피치를 변화시키는 피치 쉬프터(pitch shifter), 원래 신호에 주파수차이를 일으켜 효과를 얻는 링 모듈레이터(ring modulator) 등이 좋은 예이다.



[그림-1] 페달 파라미터를 가진 FX계열 스톱박스

이러한 파라미터의 변화빈도가 높은 FX계열 이펙터는 멀티미디어음악 작품에서 음악의 음색변화에 따른 영상변화를 유도하는 데에 유리하다. 대표적으로 클럽공연이 있는데 FX계열 이펙터가 많이 사용되는 클럽음악들은 [그림-2]처럼 영상을 제어하는 컴퓨터와 음악의 이펙터 파라미터를 제어하는 미디컨트롤러, 턴테이블 등을 상호작용시켜 소리

의 시각화를 통해 관객에게 색다른 즐거움을 준다.



[그림-2] 클럽공연에서 소리 시각화

한편 기타리스트인 연주자가 멀티미디어음악 작품의 소리와 영상을 제어하는 경우 페달 파라미터를 가진 기타 이펙터는 클럽 공연에서 손으로 조절되는 컨트롤러에 비해 섬세함에 있어서 매우 불리하다. 이는 대부분의 멀티미디어음악 공연에서 연주자가 파라미터의 조절이 가능하더라도 실제 공연에서는 연주자의 의도와는 다른 시각적 표현이 나올 가능성이 높아 영상을 담당하는 VJ<sup>4)</sup>를 따로 두는 이유이기도 하다.

## 2. 연구목적

FX계열 기타 이펙터의 파라미터 조절은 대부분 발을 이용한 페달 컨트롤러를 사용한다. 하지만 이 파라미터 조절의 문제점은 발의 힘을 이

---

4) 비디오자키(video jockey)의 약자로 주로 음악에 맞춰 다양한 형태의 패턴들을 스크린에 연출하는 사람을 말한다.

용하기 때문에 미세한 조절에는 취약하여 극단적인 변화를 통해 개성을 갖는 이펙터들이 대부분이다. 즉 페달 컨트롤러를 기반으로 한 FX 계열 이펙터는 소리의 컨트롤에 있어서 자유도가 매우 떨어진다고 할 수 있으며, 페달 파라미터의 변화 값을 영상과 같은 다른 매체에 상호 작용시켰을 때 표현의 범위도 극단적일 수밖에 없다. 따라서 본 연구는 사운드 프로세싱(sound processing)프로그램 Max/MSP를 이용하여 실험적인 효과를 가진 가상 이펙터들을 제작하여 경음악(instrumental music)에 적용시키고, 섬세한 컨트롤이 가능한 자동 컴퓨터 파라미터 조절방식을 만들어 일반적인 페달 컨트롤러의 취약점을 극복하는 것과 더불어 이러한 데이터를 이미지 프로세싱(image processing)프로그램 Jitter에 적용시켜 기타소리의 변화에 따른 실시간 영상 변화를 구현하는 것이 목적이다.

## II. 본 론

### 1. 작품의 구성

#### 1) 작품의 구조

작품<What color rose were you?>는 어쿠스틱기타, 전자기타와 Max/MSP/Jitter를 활용하여 소리와 영상이 실시간으로 상호작용을 하는 멀티미디어음악 작품이다. [표-1]과 같이 작품은 총 A, B, C의 세 파트로 나뉘져 있으며, A와 C파트는 테이프 음악(tape music)을 활용함으로써 확실한 러닝타임(running time)과 음악적 구조(musical structure)를 가지게 하였다.

B파트는 두 기타의 즉흥연주(improvisation)으로 이루어진다. B파트는 정확한 시간이 정해져 있지 않고, 약속된 연주를 통하여 파트의 이동이 행해진다. 영상은 기타소리에 의해 실시간으로 만들어지며 각 파트에 따라 상호작용하는 악기가 다르다.

[표-1] 작품의 구조

파트	구조/주제	시간	영상을 변환시키는 악기	BPM
A	intro/추억	0:00 ~ 0:56	어쿠스틱 기타	110
	verse/짝사랑	0:56 ~ 2:08		
	chorus/고백	2:08 ~ 3:03		80
	ending/이별	3:03 ~ 3:16		
B	즉흥연주/눈물	3:16 ~ 5:00	전자 기타	없음
C	intro/고독	5:00 ~ 5:30	어쿠스틱 기타 + 전자 기타	120
	verse/의문	5:30 ~ 6:25		
	chorus/깨달음	6:25 ~ 7:20		
	ending/질문	7:20 ~ 7:40		

## 2) 작품을 위한 시스템 구성

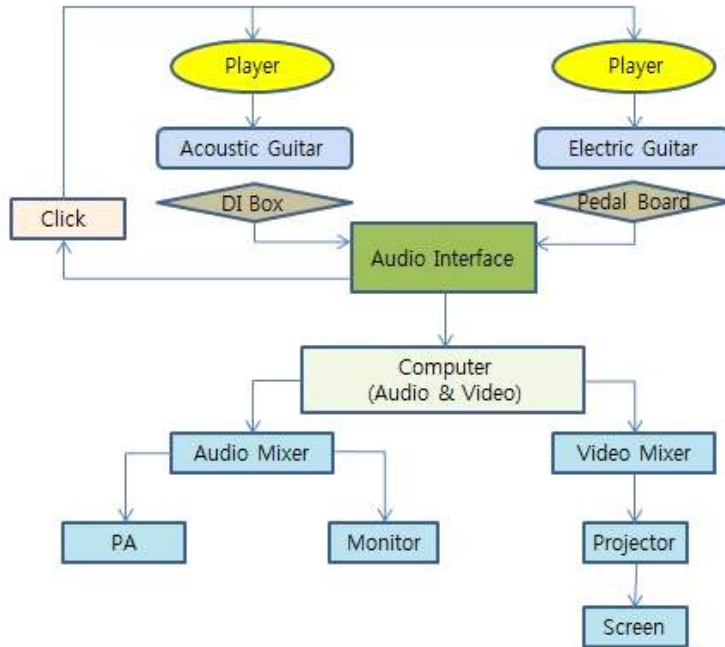
본 작품에서는 Max/MSP/Jitter만을 사용하여 사운드, 영상의 재생과 변화를 cpu에 가해지는 부담 없이 안정적으로 구현함과 동시에 개성 있는 소리 효과뿐 아니라 소리의 변화에 따른 영상 효과를 보여주기 위해 아래와 같이 시스템 구성을 하였다.

### ① Signal Flow

어쿠스틱 기타의 경우 기타의 신호가 다이렉트 박스를 이용하여 바로 오디오 인터페이스를 거쳐 Max/MSP로 들어가며, 전자기타의 순수 신호는 페달보드(pedal board)를 거쳐 일반적으로 쓰이는 컴프레서, 딜레이, 코러스, 디스토션 등을 거친 후 오디오 인터페이스를 통해 Max/MSP로 들어간다. 페달보드의 이펙터들은 연주 도중 파라미터의 변화가 적은 이펙터만 사용되었으며, 이를 통해 Max/MSP에서 구현할 경우 추가적으로 발생하는 cpu의 부담을 덜었다.

보통 전자기타의 경우 앰프를 사용하여 나온 소리를 마이크로 수음하여 컴퓨터로 보내는 경우가 많은데, 외장 앰프를 사용하는 경우 생기는 앰프노이즈(amplifier noise)와 주변에서 발생하는 소리 또한 마이크로 수음이 되어 Max/MSP안에서의 사운드 디텍팅(sound detecting)에 영향을 주는 것을 확인하였다. 이를 배제하기 위해 Max/MSP 안의 패치 vst~오브젝트를 활용하여 가상 앰프를 설정해주어 사운드 컨트롤에 있어서의 리스크를 줄였다. Max/MSP를 거쳐 가공된 오디오 신호는 믹서(mixer)를 지나 공연장의 스피커로 나간다. 그리고 가공된 오디오 신호는 Jitter로 보내어져 영상을 실시간으로 만들며, 제어한다. 최종적으로 영상은 비디오믹서를 거쳐 빔 프로젝터를 통하여 공연장에 영사된다.

[그림-3]은 본 작품의 signal flow를 도식화한 것이다.

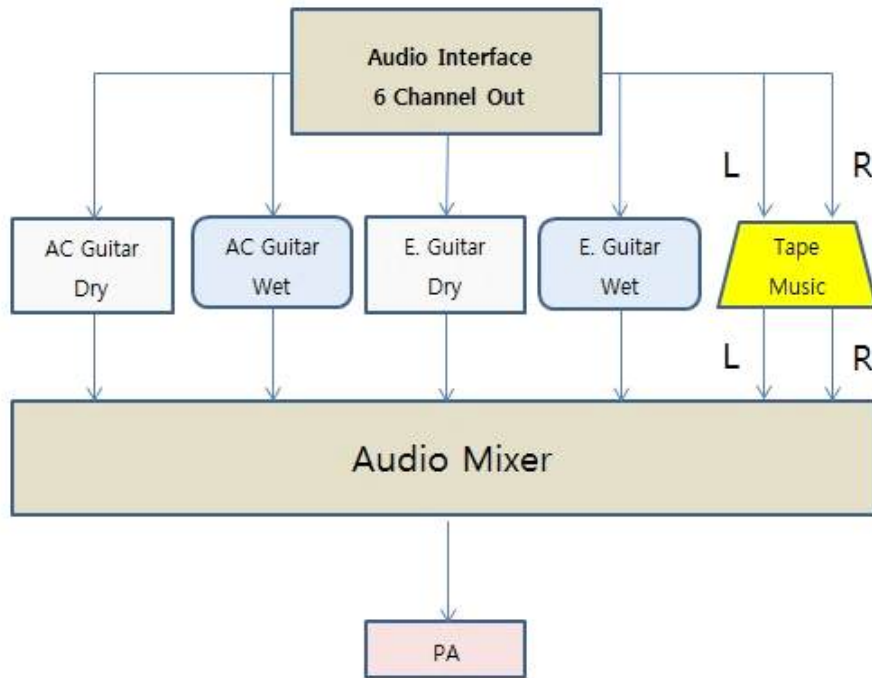


[그림-3] signal flow

## ② 사운드의 wet & dry

라이브 연주의 경우 공연장의 특성, 구조, 스피커의 배치 등에 따라 같은 음악을 틀어도 전혀 다르게 들린다. 이 경우를 고려하여 기타 두 대의 순수 신호(dry signal)와 이펙터가 적용된 신호(wet signal)를 완전히 구분하여 따로 믹서로 전달함으로써, 실제 라이브에서 사운드 밸런스를 공연장에 따라 적절하게 조절할 수 있게 하였다.

[그림-4]는 본 작품의 사운드 시스템의 구성을 도식화한 것이다.



[그림-4] 사운드 시스템의 구성

## 2. 기술적 연구

### 1) 사운드 제작

#### ① 테이프 음악 제작

작품의 A, C파트에 들어가는 테이프 음악은 반주로만 활용이 되었다. 악기구성은 일반적인 밴드음악 형식으로 드럼, 베이스, 피아노, 스트링으로 이루어져 있다. 베이스를 제외한 모든 악기는 가상악기(VSTi)로

제작되었으며, 시퀀서 프로그램<sup>5)</sup>은 큐베이스 5를 사용하였다.

제작된 트랙(track)은 각 악기를 따로 추출(export)하여 Max/MSP 안에서 믹싱을 하였으며 sfplay~ 오브젝트로 재생하였다.

## ② 그레놀러 합성방식을 이용한 이펙터 제작

그레인(grain)이란 음향데이터(sonic data)의 작은 조각들이다.<sup>6)</sup> 선(line)이 무수한 점(dot)으로 이뤄진 것처럼 그레놀러 합성방식(granular synthesis)은 약 10~50ms 길이의 그레인들을 사용하여 소리를 만드는 것이다. 그레인의 재료는 다양한 오디오샘플(audio sample)이 될 수 있는데 본 연구에서는 기타소리를 사용하였다. Max/MSP에서 그레놀러 합성방식을 구현하기 위하여 mungger~ 오브젝트를 사용하였는데 이는 미국 콜롬비아대학교의 <Dan Trueman>(1956~)이 제작한 익스터널 오브젝트이다.

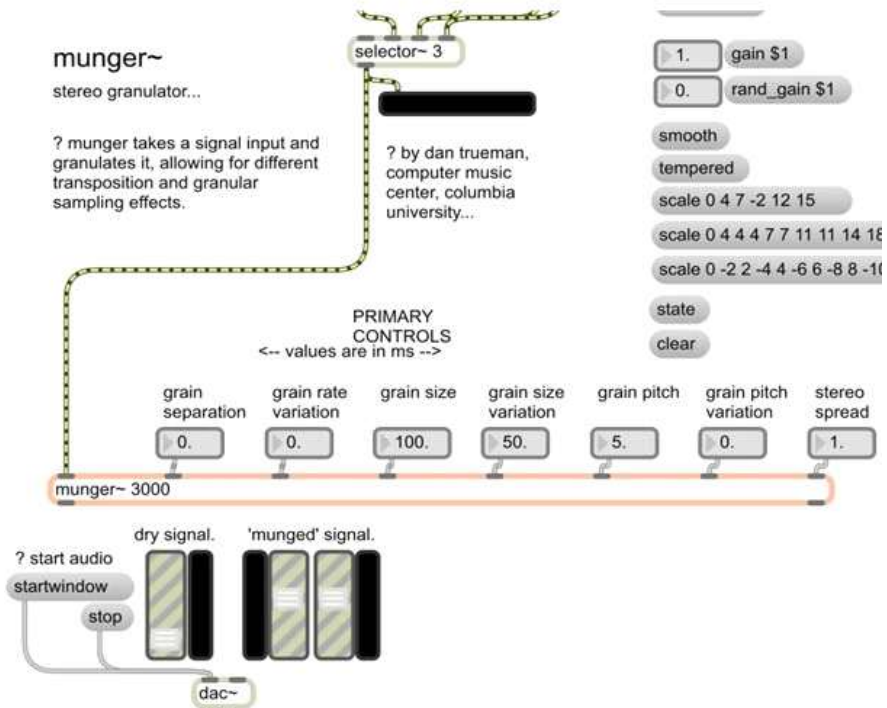
[그림-5]는 mungger~ 에서 제공되는 가장 기본적인 그레놀러 이펙터의 모습인데, 이 상태로 기타에 적용을 하면 그레놀러 사운드의 특징에 의해 서로 다른 피치들로 이루어진 grain scale이 개별적으로 딱딱하게 표현되어 다른 멜로디 악기의 화성과 부딪히는 일이 잦아 사실상 조성 음악 반주위에 연주되는 악기의 이펙터로서 사용하기에는 부적합하다.

---

5) 음악을 재생, 녹음, 편집을 할 수 있는 응용 소프트웨어

6) Timothy Opie, *What is granular synthesis?* (A Granular Synthesis Resource Site, 1999)

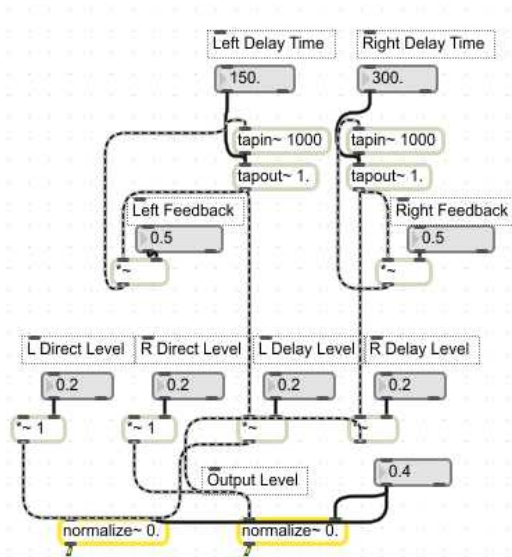




[그림-5] 기본적인 munger~패치

이를 해결하기 위해서는 dry sound를 wet sound로 만들어 줄 수 있는 공간계 이펙터가 필요하였다. [그림-6]과 같이 tapin~ 오브젝트, tapout~ 오브젝트를 이용한 ping pong delay<sup>7)</sup>를 munger~ 와 조합하여 그레놀러 이펙터에서 나온 건조한 소리는 ping pong delay를 거치며 그레인 간의 이동이 부드러워져, 한 음 한 음 선명하게 표현되던 grain scale 진행을 하나의 화음과 유사한 형태로 변환시킨다. 이러한 조합으로 만들어진 새로운 그레놀러 이펙터는 작품 전반적으로 어쿠스틱 기타에 사용되었으며, 부분적으로 전자기타의 소리로서 사용되었다.

7) 딜레이 시간차만큼 소리의 패닝이 좌우 교차하는 딜레이.



[그림-6] ping pong delay 패치

### ③ pitch shifter의 제작

'피치 쉬프터는 입력된 신호의 피치를 변화시킬 수 있는 이펙터의 일종이다. 피치를 올리고 내리면 마치 테이프에 녹음된 소리를 빠르거나 느리게 재생한 때와 비슷한 느낌의 음색을 얻을 수 있다.'<sup>8)</sup>

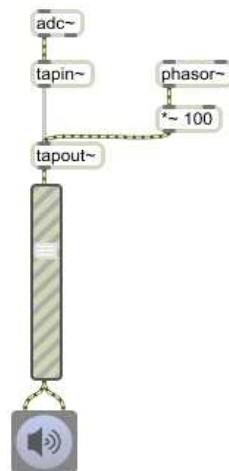
Max/MSP에서 피치 쉬프터를 제작하기 위해 보통 처음 시도하는 오브젝트는 `freqshift~` 오브젝트이다. 이는 인풋(input)에서 들어오는 오디오를 구성하는 주파수들을 이동시켜 피치를 변화시킨다. 하지만 이 오브젝트를 사용하면 인풋 신호의 피치가 이동되는 느낌보다 링 모듈레이션(ring modulation) 특유의 로봇 소리의 느낌이 강하며, 이는 인풋의 원 소스(source)를 파괴하여 본래 악기의 음색을 잃게 만든다. 그러므로 `freqshift~` 는 피치 쉬프터의 재료로서 적합하지 않다. 이

8) 김근호, 「오디오 용어사전」 (세녜출판사, 2013), 314쪽

것을 사용하지 않고 적절한 피치 쉬프터를 구현하는 것은 phasor~ 오브젝트와 tapin~, tapout~ 의 조합으로 해결 할 수 있다. [그림-7]의 패치는 도플러 효과<sup>9)</sup>를 이용하여 피치를 조절할 수 있는 피치 쉬프터의 가장 기본적인 모습이다. 이 패치에는 다음과 같은 공식이 적용된다.

$$frequency\ out = frequency\ in * (1 - phasor\ frequency * (delay\ window(100)/1000))$$

예를 들어 인풋의 주파수가 1000Hz이고 phasor 주파수가 2Hz인 경우 아웃풋의 주파수는 800Hz가 된다. 즉 phasor 주파수의 변화를 이용하여 피치를 조절하게 되는데 phasor 주파수 값이 양수일 경우 피치가 내려가게 되고 음수일 경우 피치가 올라가게 된다.

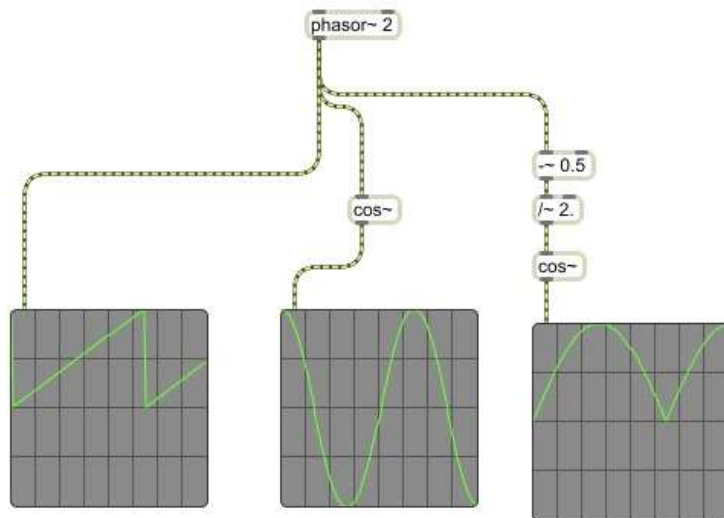


[그림-7] 기본적인 모습의 피치 쉬프터

9) 파원과 관측자 사이의 거리가 좁아질 때에는 파동의 주파수가 더 높게, 거리가 멀어질 때에는 파동의 주파수가 더 낮게 관측되는 현상

이 방식으로 피치 쉬프터의 기본적인 역할은 하지만 피치가 이동할 때 노이즈(noise)<sup>10</sup>가 발생하는 문제가 생긴다. 이는 phasor의 파형이 선형이라서 생기는 문제인데, 이를 해결하기 위해서는 cos~ 오브젝트를 사용하여 비선형으로 만들어 주면 된다.

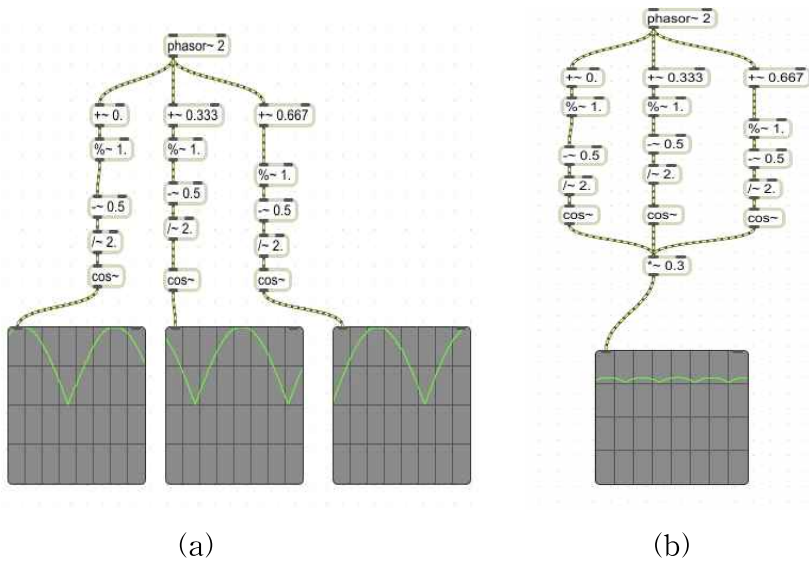
[그림-8]을 보면 시각적으로 phasor~ 에서만 나오는 파형과 cos~를 거친 파형의 차이를 쉽게 알 수 있다. 또한 cos~만을 통과하면 파형이 비선형으로 변하는 대신에 -1~1의 진폭 값을 갖게 되는데, 이를 해결하기 위해 cos~위에 두 개의  $\sim 0.5$ 와  $/\sim 2$ . 오브젝트를 거치게 하여 phasor~만을 통과 했을 때와 같이 0~1의 진폭 값을 갖게 해주었다.



[그림-8] 선형의 파형을 비선형으로 변환하는 과정

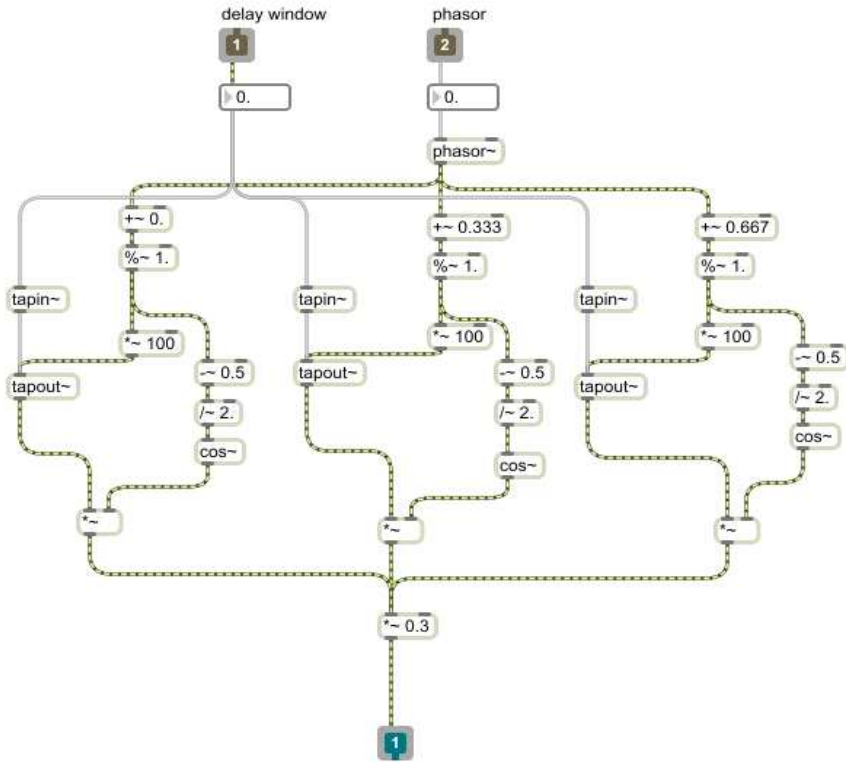
10) 파형진폭의 불연속성으로 발생하는 소리

하지만 여기서 또 다른 문제점이 발견된다. 이 상태만으로는 앰프값 변화의 폭이 커 웅웅거리는 소리가 나게 된다. 이 문제의 해결법은 +- 오브젝트와 %~ 오브젝트를 이용하여 파형의 시작점을 다르게 한 여러 패치를 합치는 것이다. [그림-9]를 보면 세 개의 패치가 같은 파형을 그리지만 시작점이 다르고(a) 이 세 패치를 하나에 묶음으로써 앰프 값의 변화가 현저하게 줄어들었음을 확인할 수 있다.(b)



[그림-9] 시작점이 다른 패치와 합쳤을 때 파형의 모습

이것을 토대로 최종적으로 만들어진 피치 쉬프터 패치의 모습은 다음 [그림-10]과 같다. 파라미터의 컨트롤은 phasor~ 위에 float 오브젝트를 달아주어 -10부터 5까지 소수점 단위로 조정할 수 있게 하였는데, -10인 경우 2옥타브 높게 나오고 5의 경우는 1옥타브 아래 음이 나온다. -10 이하나 5 이상의 값을 쓰게 되면 필요 이상의 고음과 저음이 나와서 음악적으로 사용하기가 모호해진다.



[그림-10] 최종적인 피치 쉬프터의 패치

#### ④ feedback looper의 제작

feedback looper는 라디오헤드(radiohead)<sup>11)</sup>의 기타리스트 Jonny Greenwood (1971~)가 구현한 이펙터와 유사하다. Greenwood의 이펙터는 연주 동안 기타 소리의 진행을 녹음하다가 이펙터를 발동시키면 발동 시점으로 부터 200ms 전의 연주를 지정된 횟수만큼 자동으로 반복시켜주며 마지막 횟수에 도달하면 이펙터가 자동으로 꺼지는 방식을 사용한다.

11) 1985년 영국 옥스퍼드셔 주 애빙턴에서 결성된 5인조 얼터너티브 록 밴드

본 작품에서 사용되는 Greenwood 스타일의 이펙터 feedback looper는 tapin~, tapout~를 활용하여 인풋으로 들어온 기타 신호의 피드백을 생성한다. 그리고 BPM<sup>12)</sup>에 따라 랜덤으로 피드백 속도(feedback speed)를 변화시켜 얻은 신호를 반복연주(loop)하여 기관총 소리와 같은 효과를 얻는다.

[그림-11]은 feedback looper의 가장 원초적인 모습이다. 이 패치는 처음 adc~에서 나오는 소리신호의 처음 2초(2000ms)는 tapin~을 거친 후 tapout~을 지나 다시 tapin~ 으로 들어오는 과정을 반복하게 한다. 오른쪽에 tapout~ 에 들어가는 숫자 값은 피드백 반복 속도를 의미하며 [그림-11]에서는 1초(1000ms)간격으로 소리가 반복될 것임을 보여준다. 왼쪽의 \*~0. 오브젝트에 입력되는 숫자값은 tapout~에서 나오는 신호가 tapin~으로 다시 들어갈 때의 앰프값을 변화시킨다. 또한 앰프값은 0.5로서 tapout~에서 나온 소리의 크기는 절반으로 줄어 다시 tapin~ 으로 들어가고 작아진 소리는 다시 tapout~을 거쳐 나가는 순환을 반복한다. 즉, adc~에서 나온 소리는 1초 간격으로 반복되며, 소리가 점점 줄어들며 사라진다.

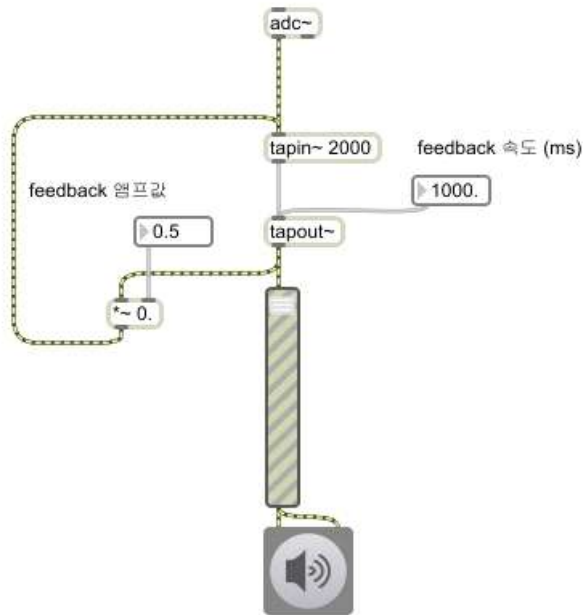
하지만 작품에서 사용될 feedback looper는 adc~에서 나오는 신호의 원래 볼륨을 유지하며 0.3초(300ms) 이하의 빠른 피드백 속도를 이용하여 기관총 소리와 같은 효과를 얻어야 한다. 0.5의 feedback 앰프값이 설정된 [그림-11]의 패치에 300ms 이하의 feedback 속도를 입력하면 단순한 딜레이 이펙터의 효과만 나온다. 또한 300ms 이하의 feedback 속도에 feedback 앰프값으로 1을 입력 하면 유도했던 기관총 같은 효과는 나오지만 하울링(howling)<sup>13)</sup>이 생기는 기술적인 문제로

---

12) Beat Per Minute의 약자로 숫자로 표기하며, 일반적으로 1분에 나오는 4분음표의 숫자를 기준으로 한다.

13) 스피커의 음이 마이크로폰에 들어가 증폭되어서 다시 스피커를 통해 나오는 식으로 정제환의 루프가 형성됨으로써 스피커에서 「삐」하는 소리가 나는 현상

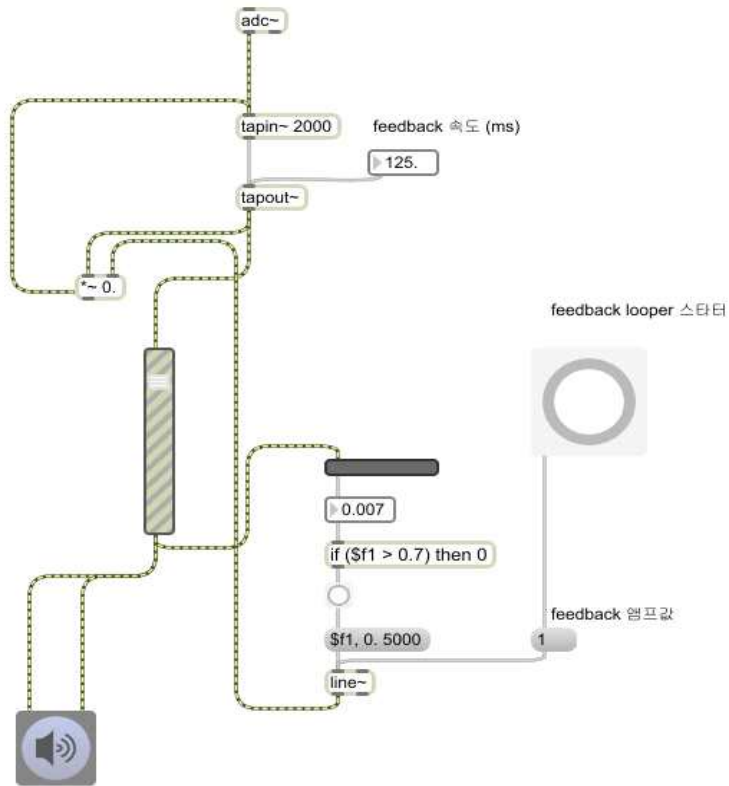
인해 작품에는 사용할 수 없게 된다.



[그림-11] 기본적인 feedback looper의 패치

이 문제는 meter~ 오브젝트와 if 오브젝트를 사용하여 간단하게 해결할 수 있다. [그림-12]는 하울링의 문제를 해결하는 패치로써 gain~ 오브젝트에서 나온 신호를 meter~를 이용하여 수치화시킨 후, if를 통해 수치가 하울링이 생기기 시작하는 feedback 앰프값 0.7을 초과하게 되면 if 아래의 bang 오브젝트를 거쳐 현재의 feedback 앰프값에서 0으로 5초간 페이드 아웃(fade out) 시켜주는 패치이다. 추가적으로 'feedback looper 스타터' bang을 생성해 줌으로써 이펙터를 사용하고 싶을 때 작동할 수 있도록 하였다.





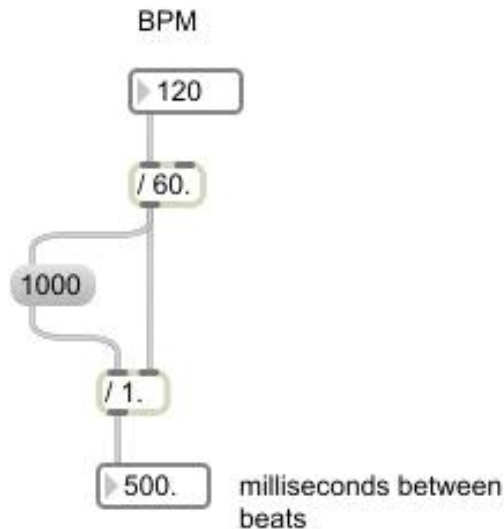
[그림-12] feedback looper의 하울링문제를 해결하는 패치

다음은 곡의 BPM에 따른 feedback 속도의 변화를 구상했다. 본 작품은 [표-1]과 같이 A파트는 110과 80, C파트는 120의 BPM을 가지고 있다. BPM의 변화에 알맞게 300ms 이하의 빠른 feedback 속도를 적용하려면 BPM에 따른 비트와 비트 사이의 시간 간격을 구해야 한다.

BPM을 X라고 하면, milliseconds between beats 는 다음과 같은 공식으로 구할 수 있다.

$$1000 \div (X/60) = \text{milliseconds between beats}$$

이것을 Max/MSP의 패치로 만들면 아래의 [그림-13]과 같다. 즉, 120BPM의 음악은 비트와 비트 사이의 간격이 500ms임을 알 수 있다.

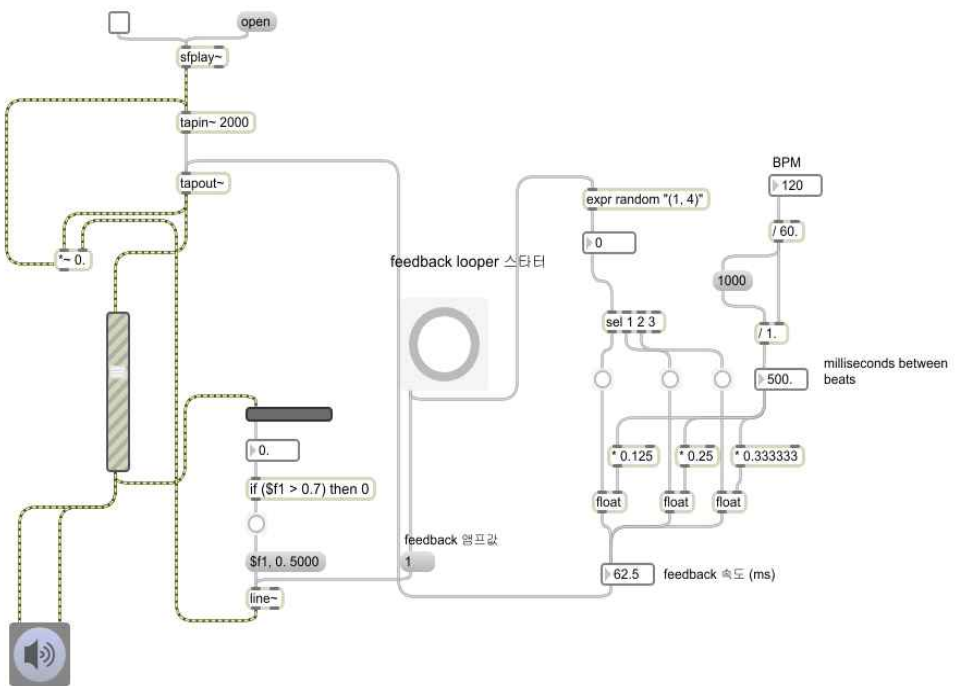


[그림-13] BPM to milliseconds between beats 패치

여기에 300ms 이하에서도 느림, 중간, 빠름 3가지 패턴의 feedback looper 효과를 만들기 위해 milliseconds between beats를 0.333333, 0.25, 0.125를 곱하여 3가지 경우의 수를 만들었다. 또한 'feedback looper 스타터'를 가동 시킬 때마다 숫자 1, 2, 3중 하나를 랜덤하게 고

르는 `expr random "(1,4)"` 오브젝트를 추가하여 `sel 1 2 3` 오브젝트로 들어오는 숫자에 따라 곱해지는 경우의 수를 무작위로 선택하게 하였다. 이를 통해 feedback 속도를 무작위로 선택하게 함으로써 이펙터를 사용 할 때마다 연주자가 예측 할 수 없는 효과를 낼 수 있게 하였다.

최종적인 feedback looper의 패치는 아래의 [그림-14]과 같다.



[그림-14] 최종적인 feedback looper 패치

## 2) 영상 제작

작품 <What color rose were you?>의 가장 중심이 되는 물체인 ‘장미’를 장미답게 제작하기 위해서는 Jitter 오브젝트들의 조합만큼이나 명령어를 이용한 설정값이 핵심적인 요소로 작용하여 ‘장미’의 외관을 만들어준다고 해도 과언이 아니다.

Max/MSP에서 얻어진 소리신호는 Jitter를 거쳐 파라미터에 따라 실시간으로 영상을 제어한다. Max/MSP의 신호를 Jitter에 적용하기 위해 가장 먼저 사용한 기술은 MSP의 오디오 데이터를 Jitter 매트릭스로 변환시켜주는 `jit.catch~` 오브젝트를 이용하여 오디오의 시각화를 도모하였고, 최종적인 영상은 OpenGL<sup>14)</sup>을 사용하여 만든 3D의 ‘장미’를 입체적이면서 효과적으로 표현하기 위해 본 작품의 패치가 제작되었다.

### ① 오디오의 시각화

처음으로 `jit.catch~`의 프레임사이즈(framesize) 값을 256, mode 0으로 설정해 주는데 framesize 256는 MSP를 통해 입력된 음성을 저주파(low frequency)부터 고주파(high frequency)까지를 256단계로 나누고 해석하여 출력해준다. 프레임사이즈 값이 커질수록 그래픽카드의 성능이 좋아야 하며 이는 곧 cpu의 부담으로 직결된다. mode는 출력 타이밍을 정하는 것인데 mode 0는 마지막 bang이 수신된 이후의 모든 신호를 순수하게 출력한다. 예를 들어, `jit.catch~`가 마지막으로 bang을 받은 후 1024샘플을 수신한 경우, 1차원의 1024셀을 가진 float32의 Jitter매트릭스가 출력된다.

---

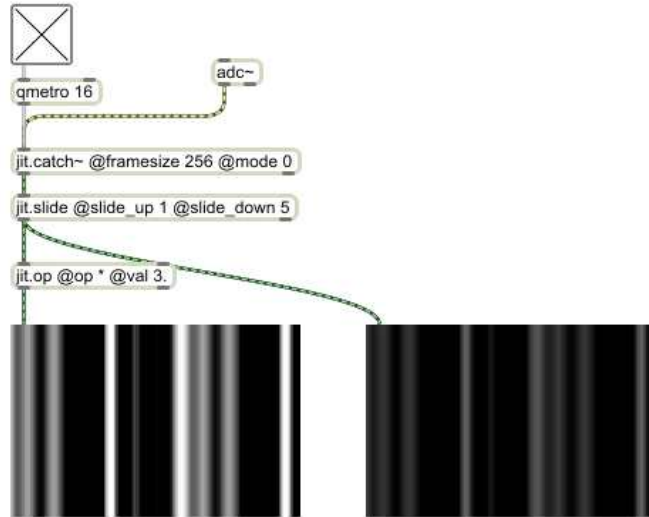
14) 2D와 3D를 정의한 산업 표준 응용 프로그램 인터페이스(API)

다음으로 `jit.slide` 오브젝트를 사용하는데 이는 연속되는 입력데이터의 평활도(smoothness)를 변화시키는 오브젝트로서, 소리의 어택(attack) 값을 표현하는 `slide_up`과 디케이(decay) 값을 표현하는 `slide_down`의 수치를 각각 1과 5를 설정하여 어택 값은 순수하게 받고 디케이 값은 길게 받아서 영상의 반응을 시각적으로 알기 쉽게 만들어 주었다. 아래 [그림-15]는 `jit.slide`를 적용한 영상과 적용하지 않은 영상의 시각적 차이를 보여주는 예시이다.



[그림-15] `jit.slide`를 적용한 영상과 적용하지 않은 영상

다음으로 작은 소리에도 반응할 수 있도록 `jit.op` 오브젝트의 값으로 `@op* @val 3`을 입력하여 3배 밝게 함으로써 [그림-16]과 같이 소리에 대한 영상의 감도를 충분히 높여주었다.



[그림-16] val값에 따른 영상의 밝기

다음으로 `jit.matrix` 오브젝트를 사용하여 지금까지의 데이터에 시간축을 주어 영상화시키는 작업이 필요하다. `jit.matrix`는 같은 이름을 가지고 있으면 데이터의 공유가 이루어지는 기능을 이용하여 제작된 패치이다. 이를 위해 `jit.op` 아래로 [그림-17]과 같은 배열의 세 가지 오브젝트를 조합시켰다.



[그림-17] 매트릭스의 데이터 공유를 위한 jitter 패치

이것을 분석해 보면, `jit.matrix` 다음에 나오는 `name`은 이름을 뜻하며 임의로 정할 수 있다. 그 다음의 `1`은 평면의 수(plane count)를 정한다. 이 숫자는 X, Y, Z의 좌표 정보를 설정하는 것인데 이번의 경우는 Z축만을 조종하기 위해 `1 plane`으로 설정하였다. OpenGL의 대부분의 파라미터는 소수점으로 컨트롤되므로 `float32`를 설정하였다. 다음의 `256 256`은 256단계로 해석된 오디오 데이터를 256 픽셀만큼 피드백할 수 있게 설정한 것이다. `usedstdim`, `dstdimstart`, `dstdimend`는 수신된 데이터를 매트릭스위에 재배치하기 위해 사용된 명령어이다.

`@usedstdim 1`로 `dstdim`을 켜주고, `@dstdimstart 0 0`로 왼쪽 위의 0 픽셀을 시작 포인트로 정해주며 `@dstdimend 255 5`로 가장 오른쪽 위에서 5픽셀 아래의 위치를 `dstdim`이 끝나는 포인트로 정해주었다. 그것을 다음의 `jit.rota @offset_y 5` 오브젝트를 이용하여 받은 매트릭스를 아래로 5픽셀 내려주어 다시 `jit.matrix name`에 더해주어 매트릭스 끼리의 공유로 인해 5픽셀씩 과거의 픽셀이 점점 아래로 내려가는 공식을 만들어 주었다.

아래의 [그림-18]는 `jit.matrix` 끼리의 공유가 없을 때의 영상이다.



[그림-18] 매트릭스의 공유가 없는 영상

[그림-19]는 매트릭스가 공유하는 영상이며 공유가 없을 때와의 차이를 확인할 수 있다.



[그림-19] 매트릭스가 서로 공유할 때의 영상

이로서 기본적인 오디오의 시각화가 가능하게 되며 장미의 소재로서 사용될 준비는 끝나게 된다.

## ② ‘rose(장미)’의 시각화

다음은 OpenGL을 사용하여 3D로 렌더링(rendering)<sup>15)</sup>시키는 작업을 해야 한다. 렌더링 작업의 토대가 되는 오브젝트에는 렌더링 엔진인 `jit.gl.render` 오브젝트, 렌더링 모델이 되는 `jit.gl.gridshape` 오브젝트, 매트릭스와 연동을 시키는 `jit.gl.mesh` 오브젝트, 텍스처를 결정하는 `jit.gl.texture` 오브젝트 그리고 화면을 표시하는

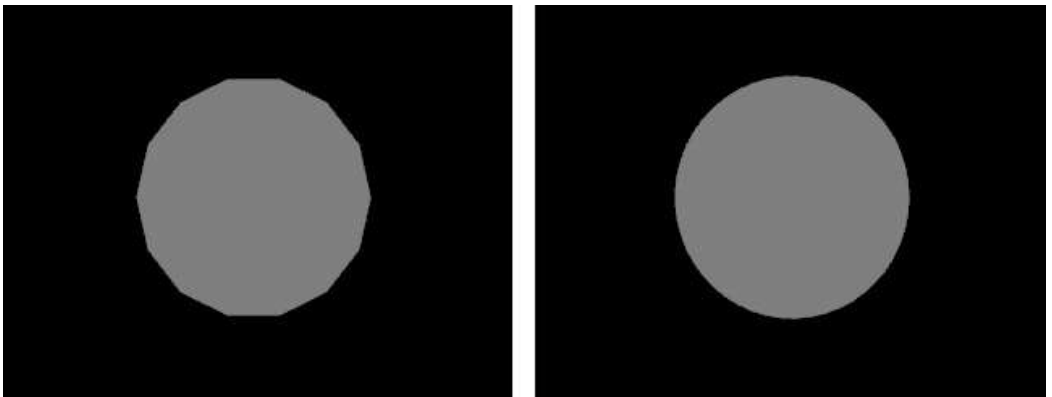
15) 2차원의 화상에 광원·위치·색상 등 외부의 정보를 고려하여 사실감을 불어넣음으로써 3차원 화상을 만드는 과정을 뜻하는 컴퓨터그래픽스 용어



jit.windows 오브젝트가 있다.

렌더링 오브젝트는 jit.matrix의 경우와 마찬가지로 서로 상호작용할 수 있기 때문에 오브젝트 뒤에 이름 'rose'를 붙여주어 상호작용시켰다. 우선 jit.windows에는 추가적으로 @fsaa 1이라는 명령어를 달았는데 이는 full scene anti aliasing의 약자로 영상의 “계단현상”을 줄여 부드러운 영상을 얻기 위한 기능으로 'smooth vision'이라고도 불린다. 다음으로 jit.gl.render에는 attrui 오브젝트중 하나인 erase\_color 오브젝트를 달아 0. 0. 0. 1로 설정해 주어 검은색 바탕 화면을 제작하고, t b erase 오브젝트를 jit.catch~ 위의 qmetro와 연결하여 영상을 재생한다.

그리고 jit.gl.gridshape에서 @dim 50 50의 설정을 추가하였는데, 이는 폴리곤<sup>16)</sup>의 숫자를 늘려 도형의 표면을 [그림-20]과 같이 매끄럽게 만들어주기 위함이다.



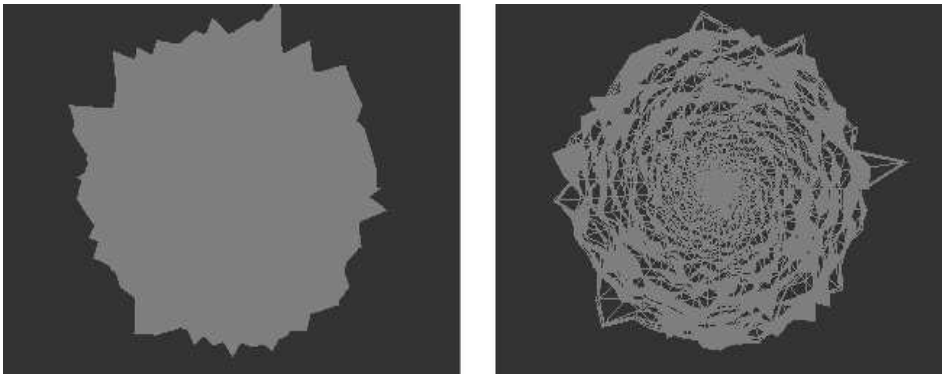
[그림-20] 폴리곤 숫자에 따른 영상의 차이

---

16) 3D 그래픽에서 물체를 표현할 때 쓰이는 기본 단위인 다각형

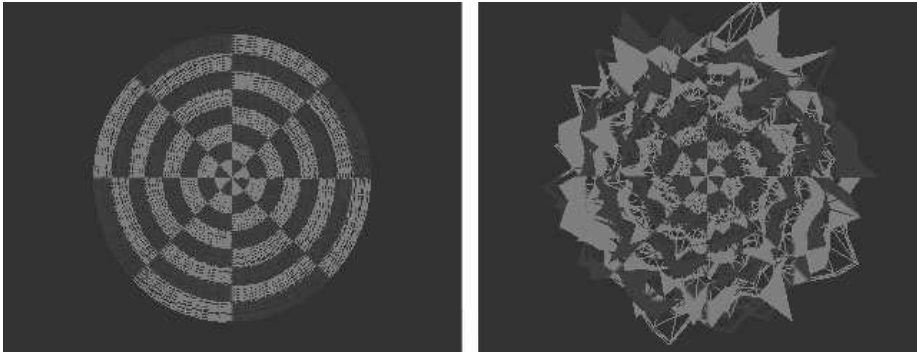
그리고 @shape circle을 추가하여 ‘장미’의 원초적인 형태를 원모양으로 결정하였다. 또한 사전에 제작된 매트릭스 데이터를 연동시키기 위해서 jit.gl.gridshape에 추가적으로 @matrixoutput 1이라는 명령어를 설정하였다.

다음으로 사용된 jit.gl.mesh에는 기본적으로 @poly\_mode 1 0을 추가해 주어 폴리곤의 앞쪽을 와이어프레임(wireframe), 뒤쪽을 솔리드(solid) 형태로 만들어 주었다. 이는 [그림-21]과 같이 음성에 반응할 때 구체가 움직이는 느낌보다는 구체를 구성하는 선들의 움직임을 더욱 효과적으로 시각화하기 위함이다.



[그림-21] @poly\_mode 사용 전후의 영상차이

마지막으로 jit.gl.texture를 이용하여 장미에 텍스처를 입히는 과정이 필요하다. jit.gl.texture와 영상의 연동을 위해 @name 명령어를 사용하여 텍스처의 이름을 만들어 주고 jit.gl.mesh에서는 @texture 명령어를 사용하여 [그림-22]와 같이 서로 상호작용을 갖게 하였다.



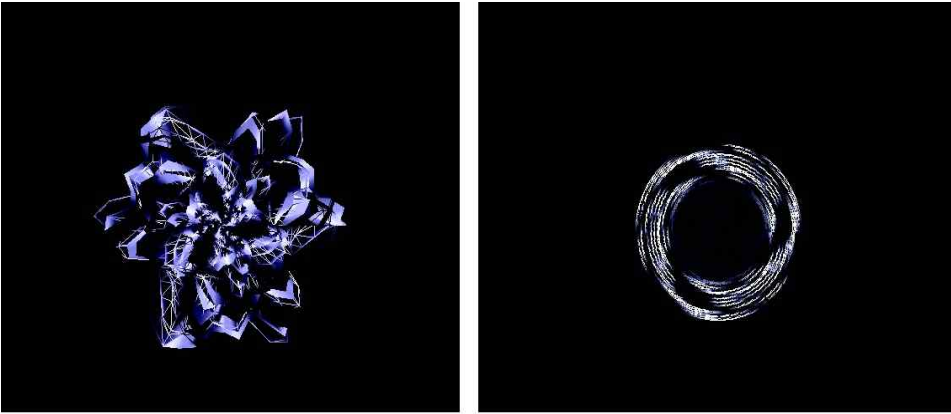
[그림-22] 텍스처를 입힌 구체와 음성에 반응하는 모습

하지만 이 상태에서는 소리 신호가 들어오지 않는 경우에도 ‘장미’의 근원이 되는 구체의 회색 텍스처가 바탕에 남게 된다.

음성이 없을 때 검은 화면을 나오게 하는 것은 `jit.op @op -`를 추가적으로 사용하여 해결 할 수 있으며, 나아가 `jit.op` 밑에 `RGBA`<sup>17)</sup> 값을 입력하는 `jit.op 4 float32 256 256`를 추가해 줌으로써 회색과 검은색으로만 이루어졌던 구체에 강제로 색을 입힌다. 또한 swatch 오브젝트의 Maxhelp에 있는 Moodmachine 패치를 사용하여 랜덤한 색상변화가 가능하게 되었다.

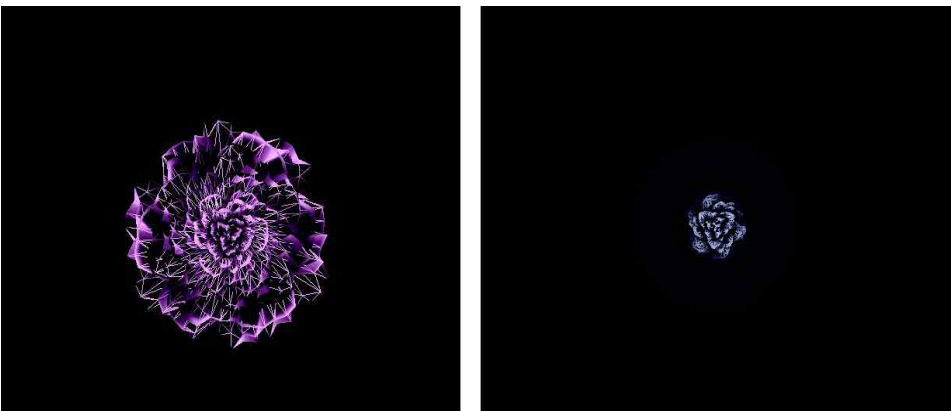
텍스처 문제를 해결하고 난 후 다른 문제가 발생하였다. 음성이 들어올 때 [그림-23]과 같이 화면의 중심에서 장미가 생성되다 갑자기 오른쪽 방향으로 회전하였으며, 음성이 사라질 때 도넛츠 모양의 파형을 그리며 사라져, 의도했던 ‘장미’의 모습이 구현되지 않은 문제가 발생하였다.

17) 1개의 픽셀에서 표현할 수 있는 색의 수. RGB(적·녹·청) 외에 투명도 8비트를 추가한 것



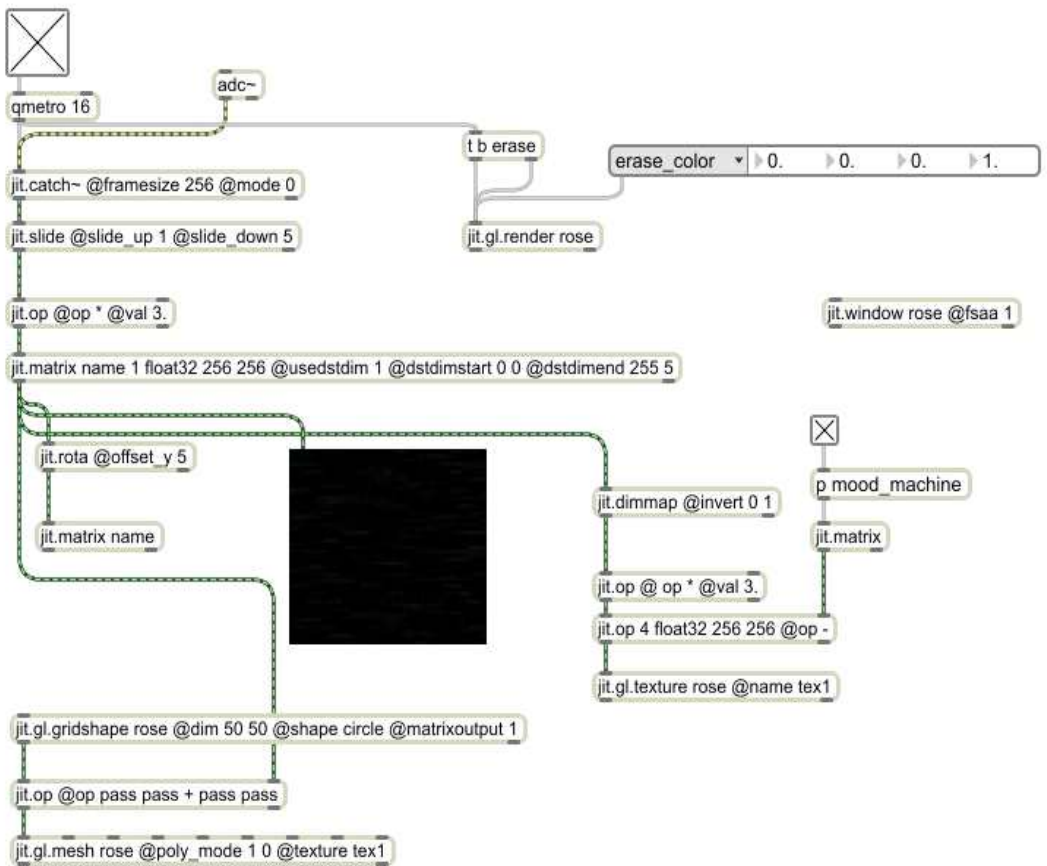
[그림-23] 텍스처 문제를 해결하고 난 후의 장미의 모습

이 문제는 중심에서 바깥방향으로 생성되는 영상을 `jit.dimmap` 오브젝트와 명령어 `@invert 0 1`을 사용하여 위상을 바꾸어 줌으로써 [그림-24]와 같이 영상이 바깥에서 중심으로 천천히 만들어지고 급격하게 회전하는 현상도 사라졌으며, 또한 음성이 작아지면 구체가 영상의 중심으로 빨려 들어가며 사라지는 ‘장미’가 완성되었다.



[그림-24] `jit.dimmap`을 활용한 후의 장미의 모습

아래의 [그림-25]는 위의 장미를 구현하기 위한 패치의 모습이며, 본 작품에서 같은 형식을 가지되 이름과 파라미터 값을 다르게 하여 두 개의 장미를 사용하였다.

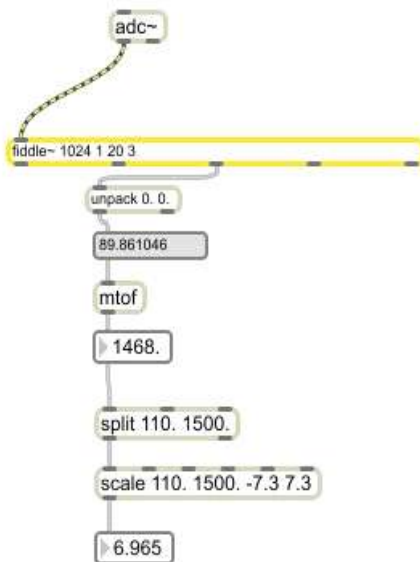


[그림-25] 장미를 구현하는 패치

### ③ 피치 디텍션(pitch detection)의 활용

피치 디텍션은 기타의 피치변화에 따른 장미 영상의 움직임 표현하기 위하여 활용한 기술로서 <Miller Puckette>(1959-)에 의해 짜여진 fiddle~ 오브젝트를 사용하였다.

fiddle~은 소리신호를 감지, 분석하여 미디값, 데시벨(dB), 소리신호의 배음이 되는 주파수들의 데이터를 추출한다. 하지만 기타 6현의 5번째 플랫인 A2(110Hz) 이하의 저음은 정확하게 감지해 내지 못하거나 그 노트의 고음 배음을 표시하는 경우가 있다. 그러므로 얻어질 주파수의 범위를 [그림-26]과 같이 split 오브젝트를 사용하여 최소 110Hz, 최고 1500Hz로 설정하였다.



[그림-26] fiddle~ 패치

최고음은 기타의 최대음역인 D6(1175Hz)를 훨씬 넘어서는데, 이는 아래의 [그림-27]과 같이 본 작품에서 연주자가 피크를 수직으로 세워 현을 긁는 피크 스크레이프(pick scrape) 주법을 사용하여 기타의 플랫보다 더 높은 고음을 내기 때문이다. 이렇게 정해진 피치의 범위는 scale 오브젝트를 이용하여 영상의 카메라 파라미터 범위로 변환되어 기타의 피치에 따라 카메라 파라미터의 x, y 축을 조종한다.



[그림-27] 피크 스크레이프 주법

### 3) 실시간 제어를 위한 auto-cue의 제작

컴퓨터를 이용하여 큐(cue)<sup>18)</sup>를 주는 것은 사전에 설정해 놓은 시간에 주어진 파라미터의 변화값을 컴퓨터에 의해 자동으로 실행하는 것이다. 이는 외부장비로 나온 페달형 이펙터보다 섬세한 조절을 가능하게 하고, 리듬섹션에 맞게 나오는 영상효과나 소리의 움직임에 따른 영상효과를 보다 정확하고 완벽한 타이밍(timing)에 이루어지게 하기 위

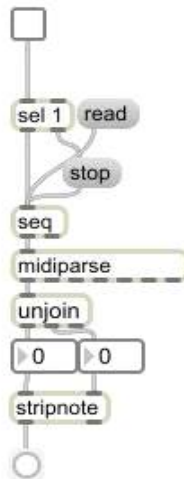
---

18) 영상 제작에서 출연자나 제작진에게 연기나 행동 혹은 카메라 작동 따위의 동작을 시작하게 하거나 멈추고자 할 때 행하는 신호

해 제작된 기능이다.

Max/MSP의 seq 오브젝트는 미리 만들어둔 미디파일을 재생하여 연주되는 미디값을 추출하는 오브젝트이다. 즉, 재생한 미디파일의 미디노트를 작곡가가 원하는 시간대에 배치하여 약속된 시간에 bang을 보내는 기능을 한다. 본 작품은 총 17개의 미디파일을 사용하여 소리와 영상의 파라미터를 조절한다. analyzer~ 오브젝트의 bang기능도 이와 비슷한 역할을 하는데 오디오파일을 이용하는 analyzer~의 bang은 오디오를 분석하여 데이터를 추출하므로 일관성이 seq보다 떨어진다. 그리고 seq에 비해 cpu 점유율이 높아 본 작품과 같이 많은 cue 패치가 필요할 때 사용하면 오류가 발생하기 쉽다.

seq에 추가적으로 [그림-28]과 같이 패치구성을 하면 미디값의 온(on) 값만을 추출하는 자동 bang이 완성된다. 미디파일들은 큐베이스를 이용하여 본 작품의 곡에 맞춰 만들어 졌으며 각 미디파일은 소리 및 영상변화의 타이밍에 핵심적인 역할을 한다.



[그림-28] seq를 이용한 auto-cue 패치



## 4. 연구기술의 작품 적용

<What color rose were you?>는 테이프 음악에 맞춰 연주를 하는 기타 연주자의 퍼포먼스와 위의 기술들을 토대로 만들어진 컴퓨터에 의한 자동 파라미터 조절로 만들어지는 소리와 영상변화의 결과이다.

본 작품의 장미는 지금까지 ‘나’가 사랑했던 사람들을 의미하며, 또한 ‘나’를 표현하기도 한다. 영상에서 장미의 움직임, 색의 변화, 잔상 등은 ‘나’가 생각하는 사랑하는 사람들의 모습, ‘나’의 마음 등을 표현한다.

각 파트마다 독특한 예술적인 효과를 얻기 위해서 대표적으로 사용된 기술은 카메라 파라미터의 조절, 잔상 파라미터의 조절, boids 파라미터의 조절, shelter의 활용 등이 있으며, 이들은 모두 유기적으로 융합되어서 특수한 영상 효과를 만들어 낸다.

### 1) A파트에서의 기술적용

#### ① 작품의 내용

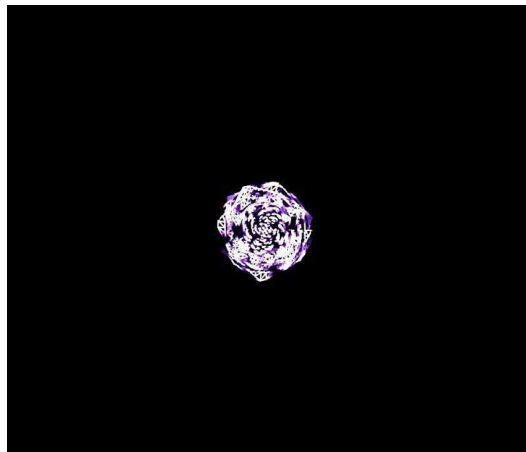
본 작품은 과거에 대한 추억에서부터 시작된다. 과거에 내가 만났던 사람의 추억, 짝사랑을 하던 모습, 사랑을 하고 이별을 경험하는 모습을 표현하였다.

#### ② 적용 기술 및 표현

영상을 분해하여 퍼뜨리는 shelter 패치와 잔상을 조절하는 파라미터 `erase_color`, 장미의 위치나 크기 등을 조절하는 영상 카메라 `jit.gl.camera` 오브젝트의 파라미터를 조절하여 예술적인 효과를 얻었다. `jit.gl.camera`에 `position` 오브젝트과 `rotatexyz` 오브

젝트를 달아주어 각 x, y, z 값을 조절하여 영상을 움직이게 하는데, position의 x는 좌, 우, y는 상, 하, z는 물체의 작고 큼을 정한다. rotatexyz의 x는 상, 하 y는 좌, 우, z는 회전을 담당한다.

먼저 영상의 position의 z값을 5로 정해주어 영상의 중심에 장미가 [그림-29]와 같이 적절한 크기로 배치되게 하였다. 즉 5를 곡 전체의 장미크기의 기본설정(default)값으로 하였으며, z축의 숫자가 올라가면 장미의 크기가 작아지고, 내려가면 크기가 커진다.



[그림-29] 장미의 기본 위치

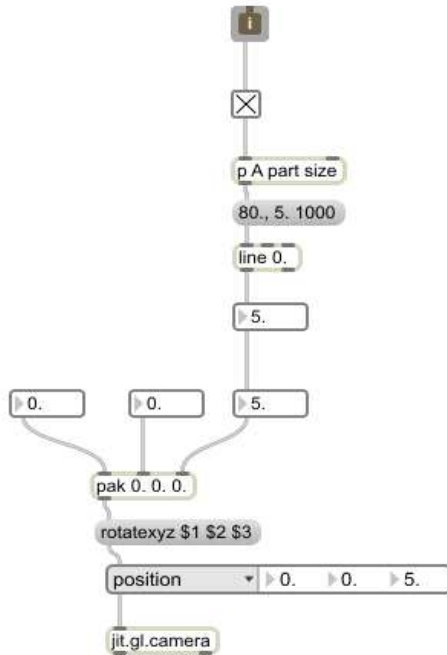
intro 파트에서는 테이프 음악의 드럼 라이드(ride)<sup>19)</sup>의 박자에 맞게 미디 파일을 따로 만들었고, auto-cue를 사용하여 영상 전체의 사이즈를 변화시켜 장미가 피어나는 듯한 시각적 효과를 주었다.

이를 위해 [그림-30]처럼 line 0. 오브젝트에 80., 5. 1000의 메시지 값을 달아 영상의 position z값을 80.에서 기본값 5.까지 1초에 걸쳐

---

19) 얇은 접시 모양의 금속으로 만든 타악기. 리듬 키프(레가토)에 쓰이는 심벌

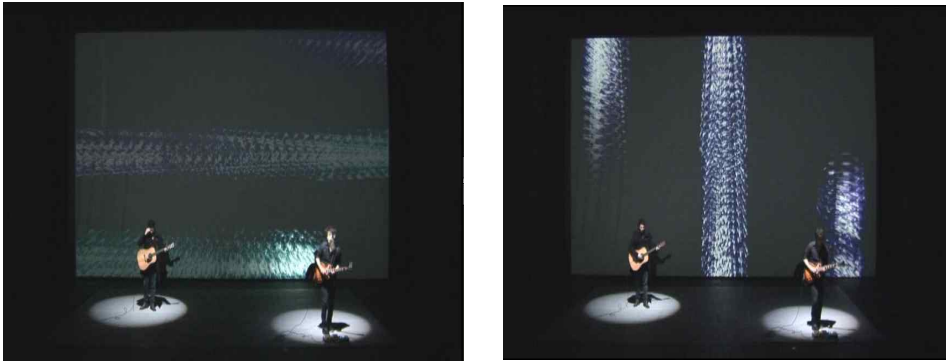
변하게 하여 장미의 크기를 변화시켰다.



[그림-30] position의 z값으로 zoom in 효과를 주는 패치

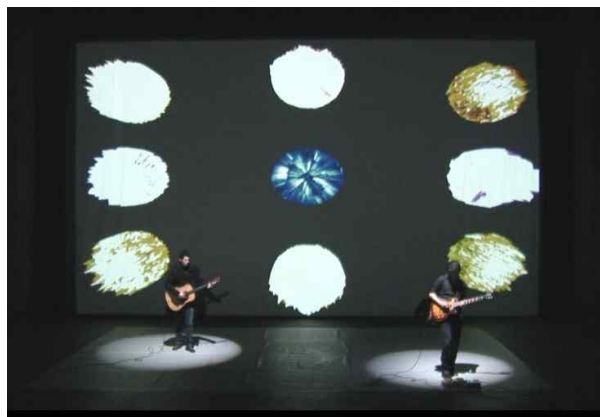
또한 장미가 스쳐 지나가는 것을 표현하기 위해 x값을 같은 방식으로 -10에서 10까지 1초에 걸쳐 변화시킴으로써 장미가 좌측에서 우측방향으로 움직이게 하고, y값을 -4, 0, 4 값을 주어 위, 중간, 아래 세 번에 걸쳐 나오게 하였다. 반대로 y값을 -10에서 10까지, x값을 -5, 0, 5를 주어 아래에서 위로 훑고 지나가는 효과를 주었다.

이때 `erase_color`의 알파(alpha)값을 0.05로 설정하여 잔상을 남겨 장미가 흔적을 남기며 스쳐 지나가는 효과를 얻었다. [그림-31]은 공연장에서 실연한 위 기술의 효과이다.



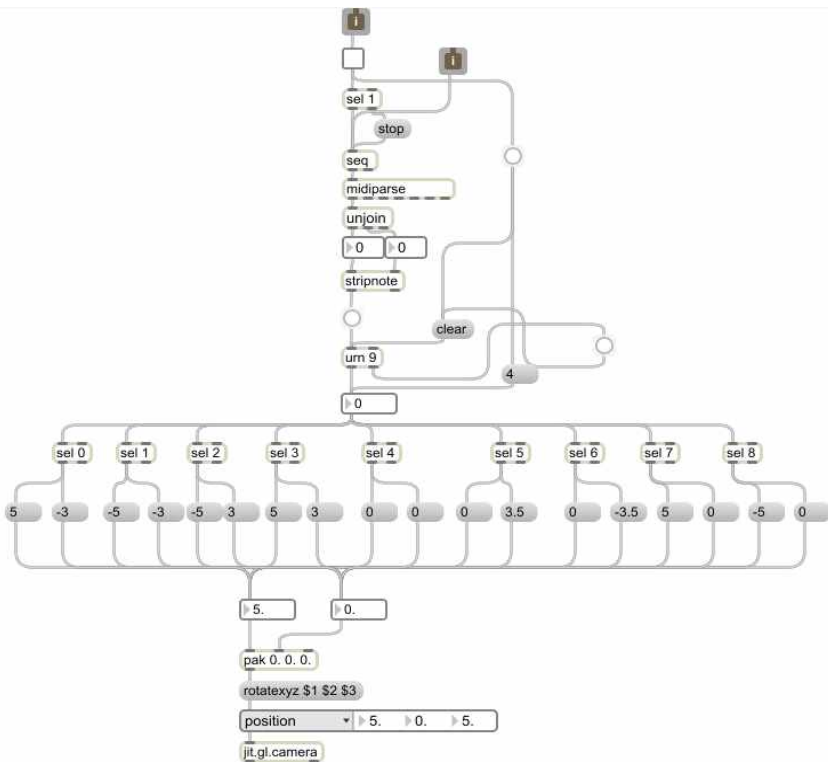
[그림-31] x, y 값에 따른 장미의 이동경로와 잔상효과

잔상을 담당하는 `erase_color`의 알파 값이 1일 경우 잔상이 남지 않으며, 0일 경우 영상이 도장처럼 화면에 찍혀서 없어지지 않는다. 즉 0에 가까워질수록 잔상이 많이 남으며 1에 가까워질수록 잔상이 적어진다. 아래의 [그림-32]는 intro의 마지막 부분에 아홉 번의 라이드 섹션에 맞춰 영상에 장미의 흔적을 남긴 것이다. 이 아홉 개의 장미는 서로 다른 색을 가지고 있는데, 이 효과는 다양한 만남을 의미하고 잔상 효과는 추억이 가슴속에 남아 지워지지 않는다는 것을 표현하였다.



[그림-32] 라이드 섹션에 맞춰 나오는 장미

이 효과는 `erase_color`의 알파 값을 0으로 하고 [그림-33]와 같이 `auto-cue` 기술과 `urn` 오브젝트를 조합하여 만든 패치로 미리 설정해 둔 `position`의 `x`, `y` 값에 무작위로 `bang`을 주는 방식으로 제작하였다. 이 패치의 `urn 9`는 0 이상 9 미만의 숫자(0~8)를 `bang`을 받을 때마다 무작위로 한 번씩만 추출하고 모든 숫자가 나온 후에는 메시지 값 4를 입력하게 되는데 `sel 4` 아래의 메시지 값을 보면 `x`, `y` 값이 0임을 알 수 있다. 즉, 장미는 무작위로 영상의 중심을 제외한 여덟 개의 방향에 잔상을 남긴 후 마지막에는 기본설정 위치인 중심으로 돌아오게 되어 다음 `verse` 파트로 자연스럽게 넘어갈 수 있게 된다.



[그림-33] 무작위로 아홉 개의 숫자를 추출하는 urn 패치

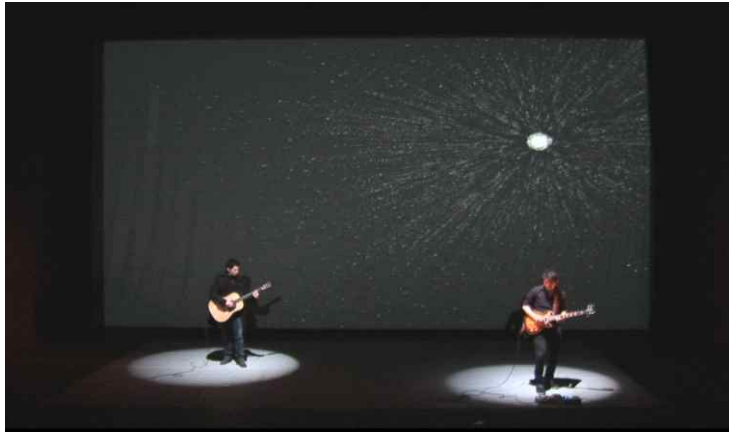
verse 파트에서의 효과는 장미의 크기와 위치 변화, 잔상, 로테이션(rotation)효과, shelter효과의 조합으로 이루어져 있다. verse는 짝사랑의 감정을 표현한 파트인데, 짝사랑을 할 때의 터질듯 한 심장의 고동과 사랑하는 사람의 주위를 소심하게 맴돌기만 하면서 ‘나’의 존재에 대해 적극적으로 알려주지 못하는 모습을 그린다.

먼저 shelter효과는 아래의 [그림-34]처럼 장미를 조각내어 불꽃놀이처럼 터지듯이 영상 전체에 조각이 흩어지도록 하는 효과이다.



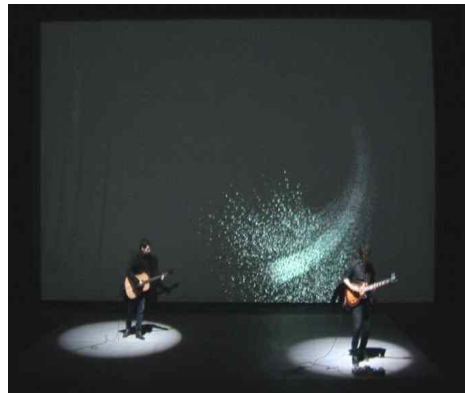
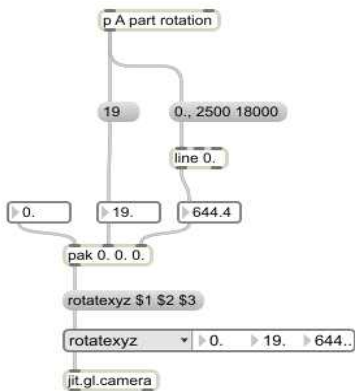
[그림-34] shelter 효과

이는 심장의 빠른 두근거림을 표현할 때 사용이 되었는데, intro 파트에서는 position의 z값을 1초 동안 변화시켜 꽃이 피어나는 효과를 얻었다면, verse에서는 같은 패치에 1초가 지나기 전에 지속적으로 빠른 bang을 함으로써 심장의 두근거림을 표현하였고, 또한 이때 shelter 효과도 같이 bang을 해주어 [그림-35]과 같은 터질듯 한 심장의 고동을 표현하였다.



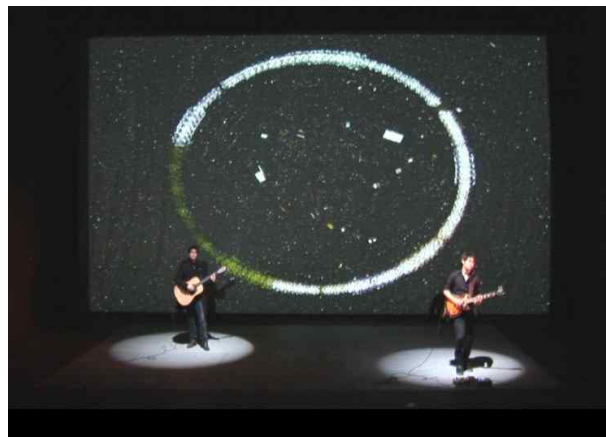
[그림-35] 심장의 고동을 표현한 영상

그리고 주위를 맴도는 것을 표현한 로테이션 효과는 jit.gl.camera의 rotatexyz 파라미터 값을 변화시키면서 얻을 수 있었다. 로테이션을 담당하는 auto-cue의 bang이 들어오면 좌, 우 이동을 담당하는 y 값을 19로 하여 장미를 오른쪽으로 배치시켜놓는 동시에 z값을 0에서 2500까지 18초에 걸쳐 변화를 주면 다음 bang이 들어올 때 까지 장미의 회전이 멈추지 않는다. 패치는 아래의 [그림-36]과 같다.



[그림-36] 원을 그리며 도는 장미의 패치와 그 모습

이 패치와 심장의 고동을 표현한 영상에서 쓰인 패치를 조합하면 다음의 [그림-37]과 같은 효과를 볼 수 있는데, 설레는 마음을 shelter의 조각들로 표현하고 주변을 맴도는 짝사랑을 표현하기 위해 잔상의 값으로 0을 주어 원을 그리며 장미를 회전하게 하였다.



[그림-37] shelter와 로테이션을 이용한 효과

chorus 파트는 intro와 verse에서 사용된 패치들을 동시에 bang을 해주며 영상 변화의 빈도를 높여 더욱 극단적인 효과가 나타나도록 하였다. 이 파트에서는 고백을 하고 사랑을 시작한 사람의 흥분된 마음을 표현하였다. chorus 파트부터 어쿠스틱 기타는 강한 스트로크 주법을 사용하고, 전자기타의 소리에 디스토션, 그래놀러 이펙터를 사용하여 절정의 느낌을 표현하였다. [표-1]에서 보듯 A파트는 어쿠스틱 기타의 소리에 따라 장미가 영향을 받는데, 스트로크주법을 사용함으로써 음량이 커져 장미 영상의 형태가 적극적으로 변화하게 하였다. 위에서 설명한 urn을 이용한 패치, 로테이션, shelter 패치의 값은 변화를 시키지 않고, auto-cue를 이용하여 동시에 또는 시간차를 두어 bang을 해줌으



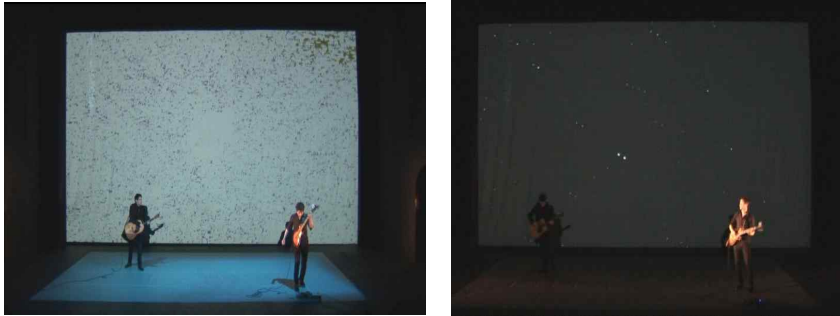
로써 지금까지 따로 변화를 주었던 position값과 rotatexyz값이 같이 변하게 하였다. 이렇게 되면 영상의 중심이 지속적으로 바뀌기 때문에 아래의 [그림-38]처럼 장미가 영상 밖 범위에서 회전을 하여 새로운 효과를 얻을 수 있다. 또한 장미가 완전히 영상 밖으로 나간 경우에는 잔상효과로 인하여 shelter효과로 발생된 조각들이 영상에 남는 효과도 얻을 수 있다. chorus 파트의 영상은 urn의 무작위 효과로 인하여 음악을 연주할 때마다 영상효과가 다르게 나온다.



[그림-38] 무작위로 만들어지는 영상효과

ending 파트는 일곱 번 연주되는 크래쉬(crash)<sup>20</sup>와 일곱 개의 노트로 이루어진 짧은 섹션이다. 여기에서는 erase\_color의 알파 값 변화를 0과 1로 극단적으로 하여 [그림-39]처럼 사랑으로 꽂 차있던 마음이 이별로 인하여 한순간에 공허하게 변하는 과정을 짧은 시간에 보여준다.

20) 얇은 접시 모양의 금속으로 만든 타악기. 악센트를 줄 때 사용된다.



[그림-39] 잔상값에 따른 극단적인 영상변화

## 2) B파트에서의 기술적용

### ① 작품의 내용

B파트는 구조가 존재하지 않는 즉흥연주(improvisation) 파트이다. 어쿠스틱기타와 전자기타가 서로 주고받는 ‘대화’형식의 음악인데, 이는 현실을 받아들이지 않는 마음(전자기타)과 헤어짐을 받아들이려는 마음(어쿠스틱기타)끼리 주고받는 대화를 표현하였다.

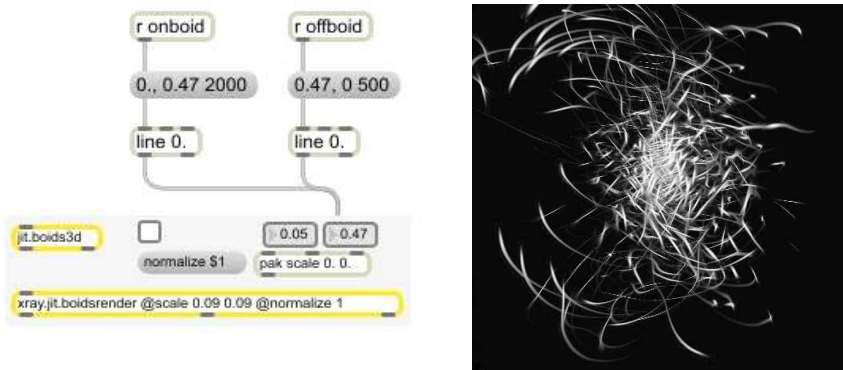
### ② 적용 기술 및 표현

B파트부터 fiddle~을 이용하게 되어 영상이 전자기타의 피치변화에 반응하여 움직이게 되는데, B파트가 시작되면 영상 카메라의 x값이 500으로 변하여 장미는 영상 밖으로 나가게 되므로 검은 화면만 남게 된다. 즉 연주를 하는 동안에 장미는 영상에 나타나지 않으며 대신에 다른 OpenGL 오브젝트인 ‘boids’가 나온다.

<Eric L. Singer>와 <W. Smith>가 제작 한 jit.boids3d 오브젝트와 xray.jit.boidsrender 오브젝트는 새의 무리가 떼를 지어 다니

는 모습을 시뮬레이팅 한 패치이며 B와 C파트의 영상 효과로 쓰인다.

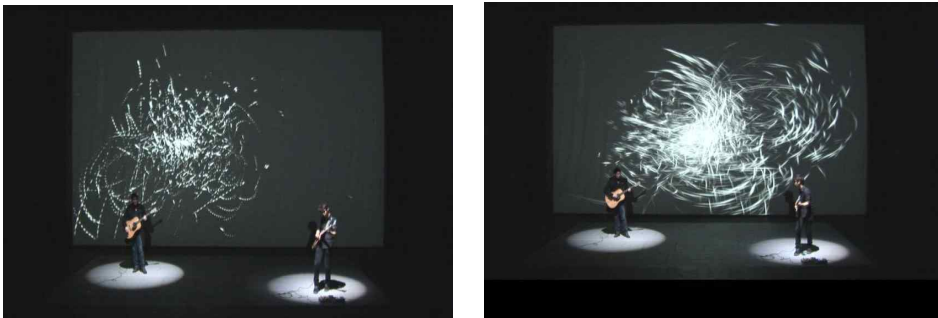
B 파트에서의 boids는 연주 도중, auto-cue에 의해 feedback looper의 패치가 켜지면 boids가 나오게 되며, 일정 볼륨을 초과하여 feedback looper가 꺼지면 boids도 사라지게 되는 형식을 가지고 있다. 작품에서 쓰인 boids의 파라미터는 두 개이며 아래의 [그림-40]과 같다. 0.05를 고정으로 가진 왼쪽의 값은 boids의 굵기를 담당하며 수치가 올라갈수록 굵어진다. 오른쪽은 길이를 담당하며 수치가 올라갈수록 점 모양이 직선으로 뻗게 된다. feedback looper가 켜지게 되면 r onboid 오브젝트가 발동 하면서 0.부터 0.47로 2초에 걸쳐 길이가 길어지며, ‘눈물’의 흐름을 표현한다.



[그림-40] boids 패치 파라미터값의 변화에 따른 영상효과

이렇게 만들어진 영상은 fiddle~에 의해 연주되는 음의 피치변화에 따라 영상의 위치도 변한다. fiddle~의 특성상 연주되는 음의 피치값 이외의 배음의 피치값을 배출하는 경우가 있어 영상 위치의 변화 빈도가 많고, 폭이 넓어져서 feedback looper의 기관총 소리와 같은 음의 떨림을 표현하기에는 적합하지만 피치의 변화에 따른 영상변화를 섬세

하게 표현하기에는 적합하지 않다. 이에 따라 피치슈프터를 함께 사용함으로써 line을 이용한 소수점 단위로 변하는 피치를 영상값에 적용하여 피치 변화에 따른 영상변화를 섬세하게 표현할 수 있었다. [그림-41]는 공연장에서 boids를 적용한 모습이다.



[그림-41] 작품에서의 boids의 적용

### 3) C파트에서의 기술적용

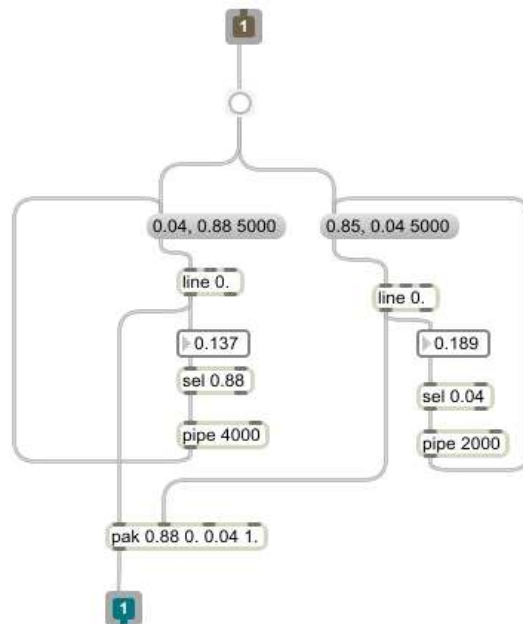
#### ① 작품의 내용

C파트는 떠나간 사람에 대한 기억과 떠나간 이유에 대한 의문, 그리고 서로 다른 색을 가졌기에 하나가 될 수 없는 것을 깨닫고, “그들은 어떤 색을 가졌던 걸까?”라는 질문을 던지며 작품이 끝나게 된다.

## ② 적용 기술 및 표현

intro 파트에서는 A파트에서 쓰이던 본래의 장미 position의 z값을 500으로 주어 화면에서 사라지게 하는 대신에 작고 붉은빛을 내는 장미가 나오게 된다. 이것은 두 번째 장미로서, A파트에 나온 장미와 같은 설정으로 다른 장미의 패치를 하나 더 만들었다. 작은 장미는 전자 기타의 소리에만 반응하게 만들어졌다.

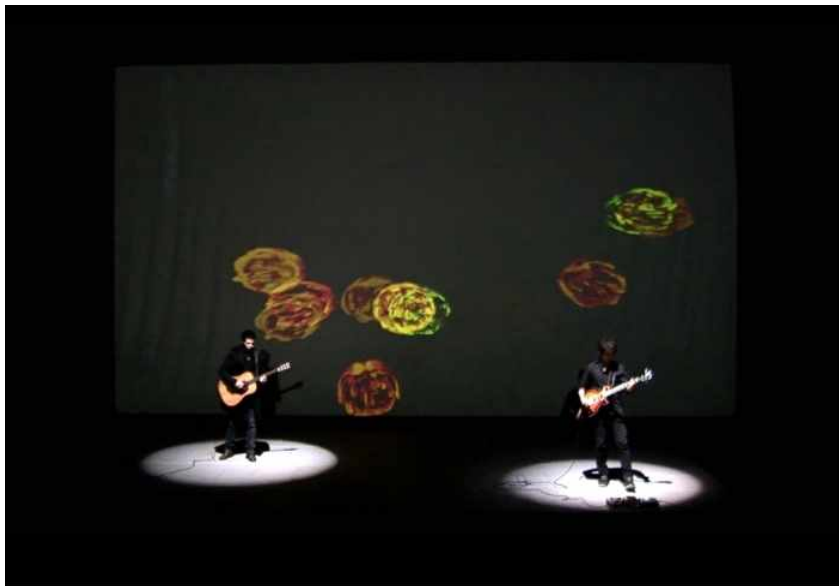
먼저 붉은 빛을 내기 위해 `jit.gl.gridshape`에 색을 설정하는 `color attrui`를 달아 [그림-42]과 같은 패치를 만들어 RGBA의 R과 G값만을 변경시킴으로써 지속적으로 붉은색, 초록색, 노란색의 장미로 바뀌게 하였다.



[그림-42] color의 RG값을 지속적으로 바꾸어 주는 패치

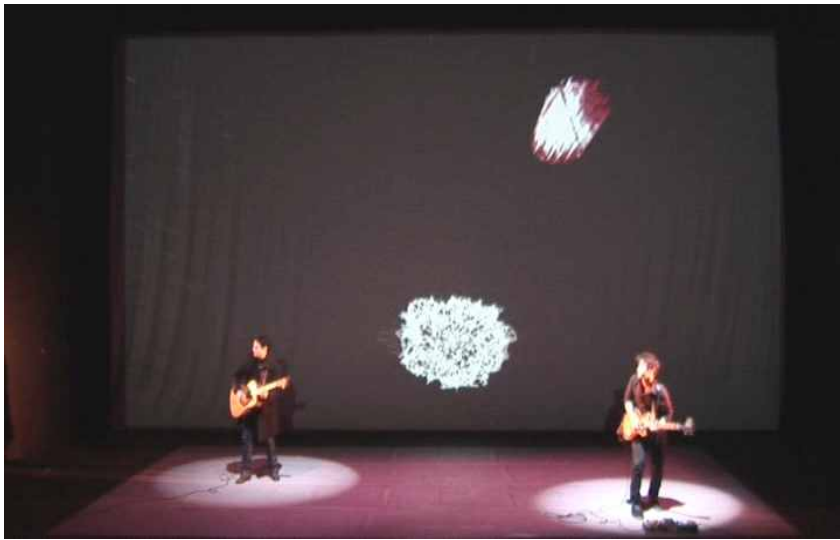
구체적으로 R값의 경우, C파트가 시작되면 bang이 들어오면서 메시지 값을 line 0.을 통해 0.04에서 0.88까지 5초동안 출력하며, 출력된 값이 0.88에 도달하면 se10.88을 거쳐 pipe 오브젝트에 들어가 4초뒤에 다시 메시지 값에 bang을 주는 원리이다. G값의 경우, C파트가 시작되면 bang이 들어오면서 메시지 값을 line 0.을 통해 0.85에서 0.04까지 5초동안 출력하며, 출력된 값이 0.04에 도달하면 se10.04을 거쳐 pipe 오브젝트에 들어가 2초뒤에 다시 메시지 값에 bang을 주는 원리이다. 그 결과 R과 G의 메시지값들의 다양한 조합이 출력되고 다양한 색상이 생성되는 효과를 거두었다.

또한 장미의 크기를 담당하는 position의 z값을 -15로 설정하여 본래 장미보다 두 배가량 작은 모습으로 하였고, fiddle~에 의해 기타 피치에 따라 영상이 움직이게 하였으며 영상의 잔상 값을 0.05로 하여 [그림-43]과 같이 화면에 그림을 그리는 듯한 효과를 주었다.



[그림-43] 화면에 그림을 그리는 작은 장미

verse 파트에서는 본래의 장미가 화면의 중심으로 돌아오게 되며 작은 장미는 피치쉬프트의 피치변화에 반응하여 움직이게 된다. 두 장미의 position x값의 차이를 15로 설정하고 fiddle~을 이용하여 전자 기타 원래의 피치값과 피치쉬프트로 인해 변하는 피치값을 각각 position의 x와 y값으로 받아 작은 장미가 화면 바깥쪽으로 멀어지게 만들어 [그림-44]과 같이 두 개의 장미가 절대로 겹치거나 만나지 못하게 하였는데 이는 ‘왜 그 사람과 나는 가까이 있지만 하나가 되지 못했던 걸까?’ 라는 의문을 표현하기 위함이었다.



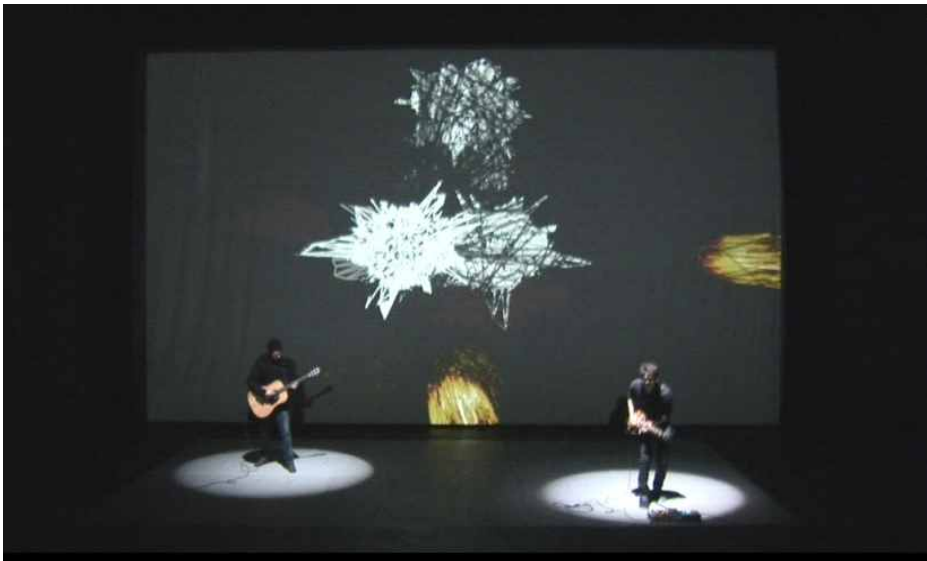
[그림-44] 서로 만나지 못하는 장미

chorus 파트는 주제인 깨달음을 표현하는 구간으로서 테이프 음악과 연주가 최고조로 변한다.

장미의 움직임을 더욱 극적으로 보이기 위해 잔상 값을 1로 주어 잔상이 없게 하는 대신에 boids의 color값을 0. 0. 0. 0.7로 하여 검은색으

로 변하게 하였고 길이 값을 20으로 극단적으로 늘려 boids가 기존의 검은 바탕을 대신하게 하였다.

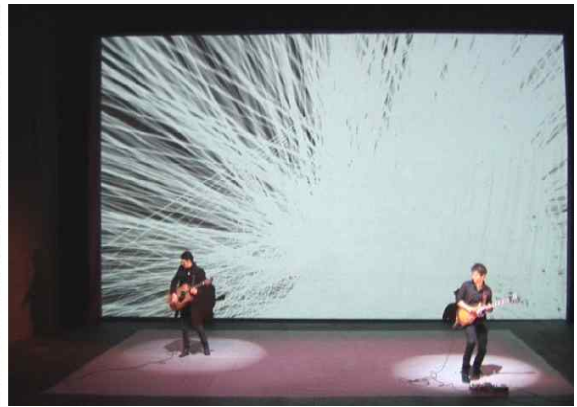
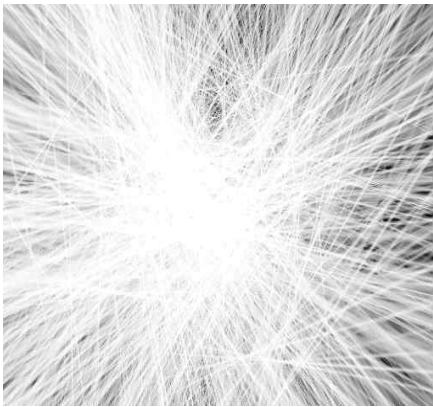
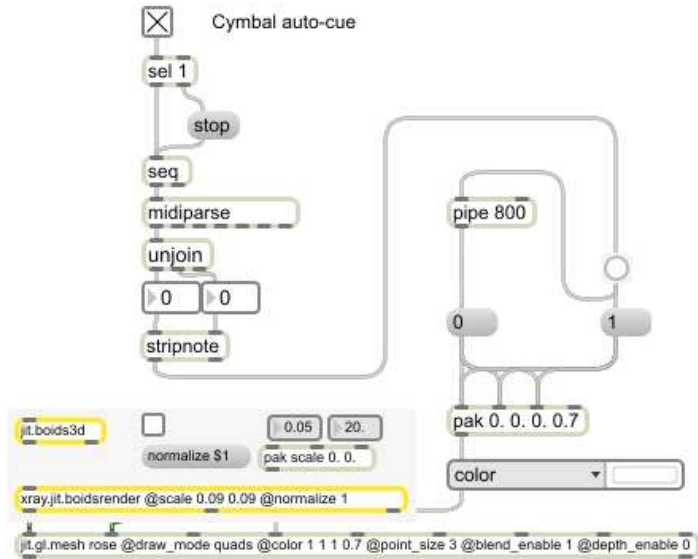
이 상태에서 장미가 나오면 [그림-45]처럼 boids의 영향을 받아 얇은 검은 선위에 장미가 나타났다가 사라짐을 반복하면서 짧은 잔상을 남기게 된다. 이를 통해 A파트의 chorus파트에서 표현된 장미보다 격렬하게 움직이는 효과를 얻을 수 있었다.



[그림-45] 검은 boids를 이용한 효과

또한 테이프 음악의 심벌(cymbal) 박자에 맞춰 bang을 보내어 boids 패치의 color 값을 1. 1. 1. 1.로 바꾸어 순간적으로 흰색이 표출되게 하였고 pipe를 이용하여 800ms 이후 다시 검은색으로 돌아오게 함으로써 심벌 섹션에 맞게 화면이 폭발하는 효과를 만들었다. [그림-46]은 이 방식을 구현한 패치구성과 효과를 보여준다.



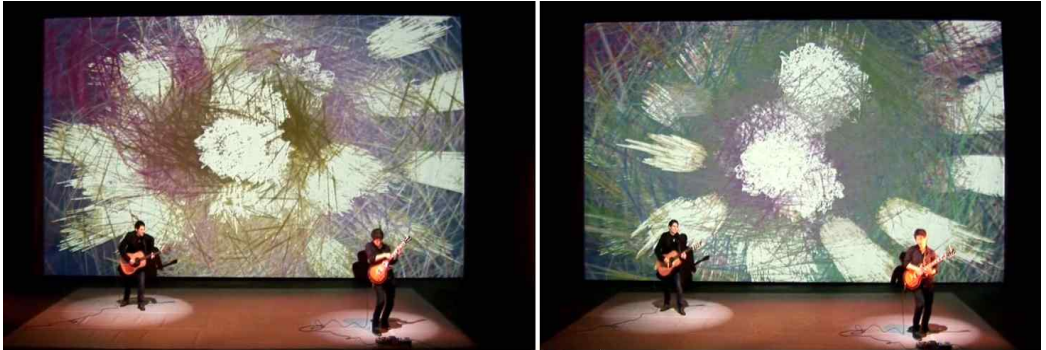


[그림-46] 폭발 효과를 구현하는 패치와 효과

chorus의 후반과 마지막 ending 파트는 “당신은 어떤 색의 장미였습니까?”라는 질문을 하는 파트로서 작품의 하이라이트이다.

전자기타는 위에 소개된 피크스크레이프 주법을 사용하여 노이즈에

가까운 소리를 내며 기타 최고음역대를 넘나든다. 이 노이즈와 같은 음색은 fiddle~의 피치 디텍팅 시스템에 의해 아래의 [그림-47]과 같이 영상에 끊임없이 표현된다.



[그림-47] ending 파트 영상효과

이때 auto-cue로 인해 영상 잔상의 값을 0으로 하여 모든 잔상이 남게 하고 이에 더하여 boids의 color값을 moodmachine을 이용하여 지속적으로 변화시킴으로써 폭넓은 색변화를 꾀하였다. 그 결과 화면에는 “당신은 이런 색의 장미였습니까?”라는 질문을 하듯 다채로운 색의 낙서가 그려지며 작품은 마무리 된다.

### Ⅲ. 결론

본 연구에서는 Max/MSP/Jitter를 이용하여 FX계열의 가상 이펙터와 소리 시각화를 위한 영상을 제작하였으며 auto-cue를 활용하여 페달 컨트롤러보다 정확한 시간에 소리와 영상을 담당하는 파라미터들을 조절할 수 있게 하였다. 이로 인해 연주자가 소리효과와 영상효과를 모두 제어할 수 있게 됨에 따라 1인 인터랙티브 예술의 가능성을 볼 수 있었다. 또한 이 연구의 패치는 기타이외의 멜로디 악기들에도 적용 가능하기 때문에 솔로 멀티미디어 아티스트들이 이를 이용하면 자신이 의도한 소리적인 면과 영상적인 면을 효과적으로 표현하는 데 도움을 받을 수 있을 것이다.

그러나 연구의 진행과정에서 나타난 문제점으로는 첫째, OpenGL을 이용하여 3D 렌더링을 하기 때문에 그래픽카드(graphic card)에 가해지는 부담이 있어, 이러한 부담을 충분히 포용하는 좋은 성능의 컴퓨터 설정이 필수적이다. 특히 실시간으로 소리효과와 영상을 제어하는 본 작품의 경우, 각 패치들은 많은 오브젝트가 사용되어 만들어지므로 이들이 화면에 나온 상태에서 영상을 재생하면 다섯 번에 한 번 정도는 auto-cue의 싱크(sync)가 맞지 않는 경우가 생긴다. 그러므로 프레젠테이션 모드를 사용하여 화면에 보이는 패치의 종류를 최소화하여 이러한 문제를 해결하여야 한다.

둘째, 미디 파일을 이용한 auto-cue를 사용하기 때문에 영상작업은 음악의 제작이 완료된 후 영상효과 종류 마다 미디파일을 만들어야 하며, 음악에 맞춰 영상효과를 줄 곳에 미디노트를 찍어 만들어야 하므로 모든 영상 효과의 종류, 큐시트를 사전에 계획해 놓아야 한다. auto-cue는 가장 정확한 기술이면서도 시간적으로는 비효율적이어서 이 연구의 약점이라고도 볼 수 있다.

셋째, 앰프값에 따라 feedback looper를 켜고 끄는 if에서 설정된 앰프값이 공연장에서는 설정 한도를 초과하지 못해 feedback looper가 꺼지지 않는 경우가 발생할 수도 있다. 즉 공연장에 따라 if에 입력된 값을 적절히 조절해 알맞은 시간에 꺼지게 해야 할 것이다.

또한 장소에 따라 설정된 메시지 값을 바꿔야 하는 경우도 생기는데, 특히 공연장에 따라 기본적으로 존재하는 리버브(reverb)가 다르므로 공간계 이펙터의 wet sound 비율의 값을 리허설을 통해 조절해야 한다. 추가로 이러한 악기정보의 변화에 따른 테이프 음악의 볼륨도 공연장의 사운드 엔지니어와 협업하여 균형을 맞춰야 할 것이다.

본 연구는 jit.catch의 기능만을 활용하여 소리의 시각화를 하였는데 향후 기타의 주법, 플랫의 위치, 몇 번 현을 사용하는가에 따라 다른 음색을 분석하여 영상에 적용한다면 더욱 섬세한 영상 표현이 가능할 것이다.

Keyword (검색어) : 컴퓨터 음악(computer music), Max/MSP/Jitter, sound visualization, OpenGL, Granular synthesis, 멀티미디어 음악(multimedia music), 인터랙티브 아트(interactive art)
--

E-mail : narunr@nate.com
--------------------------

## 참고문헌

### 1. 단행본

- 김근호, 「오디오 용어사전」  
(새녘출판사, 2013)
- Guitar world, "Guitar World Presents 200 Stompbox Reviews"  
(HalLeonardCorp, 2014)
- Johnson, Eric "Best of Eric Johnson"  
(HalLeonardCorp, 2007)
- Roads, Curtis "The Computer Music Tutorial"  
(The MIT Press, 1996)
- SYKA, Josef "Acoustical signal processing in the Central auditory system"  
(Plenum Press, 1997)
- V.J.Manzo "Max/MSP/Jitter for music"  
(Oxford University Press, 2011)

## 2. 참고논문

- 강효석, “전기기타의 피치 트래킹을 이용한 멀티미디어 음악 작품 제작 연구” 「동국대학교 영상대학원 멀티미디어학과」, (2013)
- 박상범, 동작 인식과 Max/MSP/Jitter를 이용한 멀티미디어 음악 시스템 연구“ 「동국대학교 영상대학원 멀티미디어학과」, (2009)
- 양동훈, “포먼트-기타 이펙트를 활용한 오디오-비주얼 작품 제작 연구” 「동국대학교 영상대학원 멀티미디어학과」, (2012)
- 피정훈, “실시간 음향분석을 통한 3D Visualization 연구” 「동국대학교 영상대학원 멀티미디어학과」, (2009)
- 홍성대, 장호연, 박진완, “Max-MSP를 이용한 인터랙티브 미디어 콘텐츠 연구” 「한국컴퓨터게임학회논문지」, (2008)

## 3. 인터넷

- CNMAT  
<http://cnmat.berkeley.edu>
- Max/MSP/Jitter Forum  
<http://www.cycling74.com/forums/index.php>

- Granular synthesis

<http://www.granularsynthesis.com/guide.php>

<http://www.andrenascimento.net>

<http://www.maxbreakwell.com/max-msp-jitter/custom-wah-effect>

- Guitar effects

<http://www.otherbirds.com/>

- Audio Visualization

<http://adsr.jp/>

<http://www.davidwoo.net/>

<http://www.jess-e.com>

## Abstract

# Production of Guitar Effector And Real-time Sound Visualization System With Max/MSP/Jitter

NA, Jun Ha

Recently, the influencing power of effector became a crucial factor determining the individuality of guitarist. Especially, the development of digital technology brought about not only the diversification of musical genre but also experimental guitar effector called FX line effector. Because of the frequent parameter changes, FX line effector is considered as a proper element of sound visualization.

In this study, FX line effector such as granular effector, feedback looper and pitch shifter were made with sound processing program called Max/MSP. Because every effector needed to be set a proper parameter value for the music, it was essential to make combinations between Max/MSP objects through using mathematical calculations. In addition, 3D image was made with OpenGL in Jitter, which is a kind of image processing program. The main image is developed as a 3D rose shape which was derived from the 3D sphere. So it needed to build Jitter's command values such as number of polygon, texture and wireframe sets of 3D sphere.

When it comes to the sound visualization, those three guitar



effector parameters interlock with image and each parameter controls values of the main image's afterimage, position, camera and color.

For the one person interactive performance, the main technique called 'auto-cue system' was made. This technique uses MIDI signal as an instruction and it makes up for the weak point of pedal controller that is vulnerable to fine-tuning of parameter. As a result, auto-cue system makes it possible to get a perfect cue timing and control parameter delicately.

On the other hand, because each parameter requires different MIDI file respectively, each result value in Max/MSP/Jitter should be set in advance. It means that it is necessary to prepare a cue sheet of parameter value sets. So it is hard to expect the improvised effect.

As the parameter value is crucial factor in the performance, it requires a lot of rehearsals to find the most appropriate range of sound change and way of image expression. Though it could be inefficient in terms of the time, it is good way to represent composer's intention accurately. Using this technique in multimedia music performance, solo multimedia artists easily expand range of the artistic expression. Also this technique reduces various errors on the stage effectively, therefore, higher quality performance can be offered to audiences.

## 부록: 첨부DVD설명

1. 테이프음악 폴더 : A, C파트 테이프 음악 음원
2. 기술패치 폴더 : Max/MSP 기술 패치
3. 파트별 기타음원 폴더 : A, B, C파트 기타 음원
4. auto-cue files 폴더 : A, B, C파트 auto-cue 파일
5. WCR.maxpat : <What color rose were you?> 패치
6. WCR.mp4 : <What color rose were you?> 공연실황