석사학위논문

실시간 모션그래픽 기반의 프로젝션 매핑 구현 기법에 관한 연구

2017년

한성대학교 대학원

미디어디자인학과

영상커뮤니케이션디자인전공

김 진 태

석사학위논문
지도교수 김효용

실시간 모션그래픽 기반의 프로젝션 매핑 구현 기법에 관한 연구

A Study on the Implementation Method of Projection Mapping using Real-time Motion Graphics

2017년 월 일

한성대학교 대학원

미디어디자인학과

영상커뮤니케이션디자인전공

김 진 태

석사학위논문
지도교수 김효용

실시간 모션그래픽 기반의 프로젝션 매핑 구현 기법에 관한 연구

A Study on the Implementation Method of Projection Mapping using Real-time Motion Graphics

위 논문을 미술학 석사학위 논문으로 제출함

2017년 월 일

한성대학교 대학원

미디어디자인학과

영상커뮤니케이션디자인전공

김 진 태

김진태의 미술학 석사학위논문을 인준함

2017년 월 일

심사위원장 _____(인)

심 사 위 원 _____(인)

심 사 위 원 _____(인)

국 문 초 록

실시간 모션그래픽 기반의 프로젝션 매핑 구현 기법에 관한 연구

한성대학교 대학원

미디어디자인학과

영상커뮤니케이션디자인전공

김 진 태

프로젝션 매핑은 빔프로젝터를 이용하여 정지 또는 움직임을 갖는 이미지를 입체적인 형상을 가진 사물 또는 공간 등에 투사하여 만들어지는 비정형 영상 연출 기법이다. 프로젝션 매핑 기법을 적용하여 만들어지는 작업들은 박람회, 이벤트 등의 상업적인 콘텐츠 분야에서 공간 연출과 제품의 전시 연출, 디지털 사이니지(Digital signage) 등으로 사용되며, 예술 분야에서 공연과 전시 등에 융복합 예술 작품 또는 그의 구성 요소로 나타나기도 한다.

프로젝션 매핑에 사용되는 영상은 보통 실사 촬영 소스와, 2D 및 3D 모션그래픽 기법 등으로 제작하는데 스토리텔링이 중요한 방송 및 영화 영상과 달리 그 자체로 볼거리의 목적으로 활용되기 위해 다양한 영상 효과를 사용하여 제작한다.

제작 후 렌더링을 통해 출력된 영상을 대상의 형상에 정확히 맞추어 투사할 수 있는 프로젝션 매핑 전문 소프트웨어를 이용하여 투사(projection) 하는데, 이러한 소프트웨어들은 화면을 여러 조각으로 분할할 수 있고 전부 또는 분할된 면 각각을 대상의 형태에 맞추어 공간적으로 자유롭게 회전하고 형태를 변형할 수 있는 기능을 가지고 있다.

렌더링 영상을 이용하여 프로젝션 매핑을 구현하는 방법은 비교적 표준적인 방법

으로 자리 잡았지만 몇 가지 한계도 갖고 있다. 첫째로 프로젝션 매핑에 사용된 영상은 보통 대상의 형태를 고려하여 영상을 제작하기 때문에 재사용이 어렵고 둘째로 일반 영상물과 같이 선형적인 재생 시간을 갖고 있어서 대부분 단순 감상을 위한 콘텐츠로 선보이며 셋째로 일부 표현 방법 중 입체적인 이미지를 표현할 때 그림자와 원근감의 방향에 따라 관객의 위치를 고정해야 하는 한계가 있다.

영상의 한계에 더하여 프로젝션 매핑 사용하는 모션그래픽은 대상의 조형적 유사성에서 비슷한 분위기와 형식이 만들어지기 쉽다. 표현의 유형화가 나타나면 점차 식상한 콘텐츠가 될 수 있다. 보다 지속 가능한 콘텐츠로 개발하기 위해서는 사용되는 영상의 한계를 개선할 필요가 있다.

따라서 본 연구의 목적은 영상 기반의 프로젝션 매핑이 갖는 한계를 개선하여 새로운 표현이 가능한 프로젝션 매핑 기법을 연구하는데 있다. 본 논문에서 제안하는 영상의 제작 방식은 실시간 렌더링 기반의 프로그램을 사용하여 동영상을 생성시키는 실시간 모션그래픽(Real-time motion graphics)이다.

실시간 모션그래픽은 프로그래밍 코드 등에 의해 실시간으로 생성되는 모션그래픽을 의미하며 컴퓨터 생성 이미지(Computer-generated imagery)의 하위 개념이자 실시간 컴퓨터 그래픽(Real-time computer graphics)에서 영상 디자인 부분을 강조한 용어로 사용하였다.

실시간 렌더링으로 생성되는 모션그래픽은 재생 시간에 구애를 받지 않는 영상을 만들어 내고, 실시간으로 변수를 대입하여 즉시 다양한 변형을 가능하게 할 수 있으며 필요에 따라 관객과의 상호작용이 가능한 참여형 작품으로 개발하기 수월한 장점을 갖는다.

프로세싱(Processing), VVVV와 같은 미디어 콘텐츠 프로그래밍 언어들과 유니티(Unity)와 언리얼 엔진(Unreal Engine) 등 최신 게임엔진들이 대표적으로 실시간 렌더링을 지원하는 프로그램들로 볼 수 있다. 이러한 프로그램들의 그래픽 품질이 점차 높아지면서 최근 뉴미디어 아트 콘텐츠, VR, 게임 등 다양한 시각 콘텐츠 및 예술 분야에서 사용되고 있는데, 실시간 렌더링을 바탕으로 프로젝션 매핑에 사용한다면 기존 영상이 갖고 있는 한계를 개선하고 새로운 아이디어를 구현하는데 쓰일 수 있다.

선행 연구로 영상 기반의 프로젝션 매핑의 과정과 방식, 소프트웨어 등을 분석하여 한계와 개선 지점을 정리하고 실시간 모션그래픽을 프로젝션 매핑에 적용한 사례들을 연구하여 새로운 시도와 기술 요소를 분석하였다.

다음으로 실시간 모션그래픽을 제작할 수 있는 대표적인 실시간 렌더러 기반의 프

로그래밍 도구인 VVVV를 사용하여 실시간 모션그래픽을 프로젝션 매핑에 적용하는 기법을 작품의 제작을 통해 경험 연구로 진행하여 그 결과를 공유하였다. 추가로 프로젝션 매핑에 사용될 수 있을 것으로 판단한 게임엔진 유니티(Unity)를 프로젝션 매핑에 사용할 방법을 정리하였다.

프로젝션 매핑에 실시간 컴퓨터 그래픽스를 활용하는 사례들이 늘고 있지만 영상 기반의 프로젝션 매핑 소프트웨어처럼 비교적 표준화가 진행된 방식이 따로 없이 다양한 방법들이 사용되고 있다. 대체로 영상 기반의 프로젝션 매핑은 특성상 콘텐츠의 소비 기간이 짧고 효과 위주의 표현에 그치고 있는데, 실시간 렌더링 방식의 제작 기법을 사용한다면 계획된 디자인 의도를 넘은 다양한 패턴의 생성으로 표현의 폭이 넓어지고 관객과의 상호작용을 추가할 수 있는 등 보다 지속 가능하며 경쟁력 있는 콘텐츠로 개발하는데 도움이 될 수 있을 것이다. 본 연구를 통해 실시간 모션그래픽을 프로젝션 매핑에 활용할 경우 얻을 수 있는 가능성을 확인할 수 있으며 적용 기법에도 도움이 될 것으로 기대한다. 아직은 게임엔진을 활용한 프로젝션 매핑은 간단하지 않지만 앞으로 후속 연구로 진행된다면 더욱 다양한 연출이 가능해질 것으로 기대한다.

【주요어】 프로젝션 매핑, 실시간 모션그래픽, 비정형 영상, 프로그래밍, VVVV

목 차

표 목 차

그 림 목 차

Ⅰ. 서 론

1.1 연구의 배경 및 목적

빛을 쏘아 움직이는 이미지를 보여주는 영사(projection)의 역사는 영화의 등장과 함께 출발하였다. 영사기는 한 장의 정지된 이미지를 보여주던 초기 영사기인 매직 랜턴(Magic lantern)에서 연속된 사진을 이용해 동화상을 보게 해준 주프락시스코프(Zoopraxiscope)의 가능성을 만나 뤼미에르 형제에 의해 관객에게 돌진하는 기차를 선보인 영사기로 발전하며 영사기법이 영화의 탄생과 함께 시작되었다.

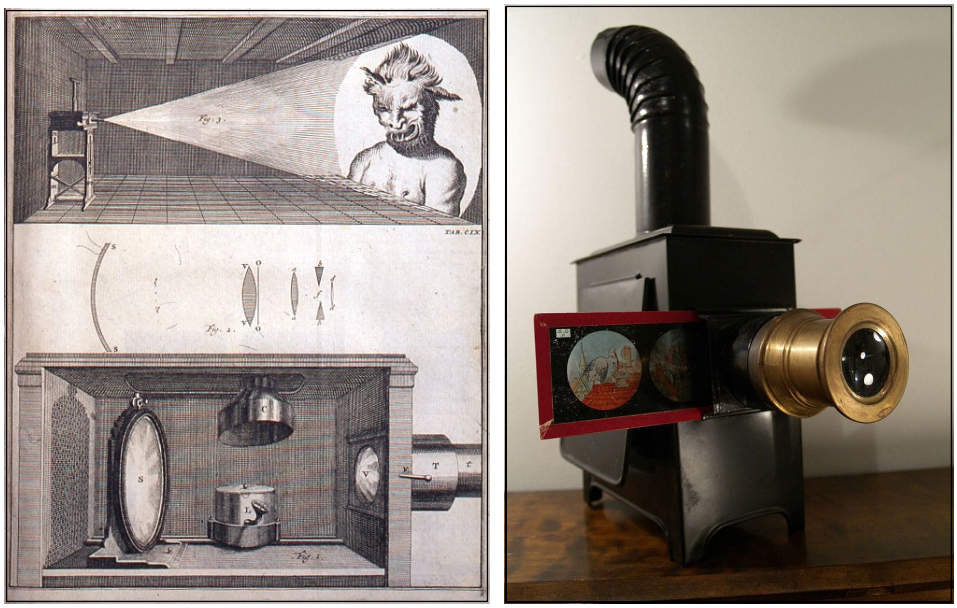
초기 텔레비전은 영화 필름과 같이 1.33:1의 비율을 가지고 있었다. 가정에서도 텔레비전을 통해 영상을 볼 수 있게 된 세상에서 영화계는 위기감을 갖고 있었고 이에 관객이 극장으로 계속 올 수 있도록 다양한 영상 기술의 발전으로 이어진다. 하나의 예로 와이드 영상 규격이다. 좌우로 늘어난 새로운 화면 규격은 극장 영상과 텔레비젼 영상이 모두 4:3 비율에 가깝던 시대의 영상에 새로운 가능성을 열게 되었다. 1.6:1, 1.85:1, 2.39:1에서 영사기 3대로 동시에 상영하는 시네라마(Cinerama)까지 새롭게 등장한 와이드 화면비율은 가려졌던 좌우의 공간을 열어 보이며 당시의 텔레비전보다 나은 시각적 몰입 환경을 제공하는데 큰 기여를 했다.

레프 마노비치(Lev Manovich)는 영화를 투사하는 스크린을 하나의 견고한 창이라기보다 아티스트가 변화시킬 수 있는 캔버스라는 생각을 갖고 있었다.[1] 영상을 가두던 직사각형 스크린이 없는 곳에서 영사기의 빛은 공간을 새로운 빛의 텍스처(texture)로 감싸며 무대 미술이 되었고 예술이 되고 인테리어가 되었다. 그 시작은 1969년 디즈니랜드에 있던 라이드 콘텐츠인 Haunted Mansion으로 보고 있다.[2] 5개의 흉상에 16mm 영상기로 얼굴 영상을 투사하여 노래하는 유령을 표현하였는데 입체적인 대상에 영상을 투사

1) 최양현. 조성민. (2015).「퓨처시네마」. 서울:커뮤니케이션북스.

2) Brett Jones. The Illustrated History of Projection Mapping. (2017. 10, 1). http://projection-mapping.org/the-history-of-projection-mapping/

하여 프로젝션 매핑의 효과를 얻은 초기 사례로 볼 수 있다.



〈그림 1〉 매직 랜턴(Magic lantern)

〈그림 2〉 초기 프로젝션 매핑(1969), 디즈니랜드

90년대 들어 고광량 영사기의 빛을 각종 건물과 공연장에 투사하는 사례들이 나타났다. 초기에는 PIGI 프로젝터와 같이 대형 필름을 고광량의 영사기로 공간에 빛을 쏘는 Large-format slide projector 방식의 영사 기법을 사용하였다. 비록 PIGI 프로젝터는 동영상이 아닌 정지 이미지를 담은 필름이 상하좌우로 이동하며 만들어지는 움직임을 투사하는 방식으로 일반적인 동영상은

아니지만 밝은 빛이 필요한 대규모의 극장, 스타디움, 건축물 등에 투사하여 얻을 수 있던 미술적, 장식적 효과로 인해 다양한 이벤트와 쇼에 사용되었다.

〈그림 3〉 서울시청 PIGI show와 PIGI Projector

투사 방식이 전자식 프로젝터로 발전하고 입체적인 사물에 보다 정밀하게 투사할 수 있는 프로젝션 매핑 전문 소프트웨어들이 등장하면서 무대에서 전시, 이벤트에 이르기까지 다양한 분야에서 본격적인 프로젝션 매핑의 사례들이 나타났다.

기술의 발전은 투사 방식에도 변화를 가져왔고 현재 비교적 표준화가 진행되었다. 이것은 영상을 투사와 동시에 투사될 공간 및 사물의 형상에 맞추어 이미지를 배치하고 변형할 수 있는 기능을 바탕으로 투사할 때 발생하는 왜곡을 보정하는 것이다. 이러한 프로젝션 매핑 기법을 제공하는 소프트웨어로 Madmapper, Resolume, Millumin 등 수십여 종의 소프트웨어들이 출시되어 있고 이들은 대체적으로 비슷한 방식으로 프로젝션 매핑을 수행 할 수 있다.

프로젝션 매핑은 주로 상업적인 이벤트와 공연 같은 분야에서 많이 사용되었는데 그 이유는 작은 크기에서 대규모의 공간까지 입체적인 대상에 극적이고 자극적인 시청각적 체험을 주기에 적합한, 볼거리의 측면에서 활용도가 높았기 때문이다.

프로젝션 매핑의 사례들이 점차 누적될수록, 그 콘텐츠 역시 대중들에게 차츰 익숙함을 주게 되었다. 이러한 문제는 전문적인 시스템이 출현하고 안정화가 진행되면서 점차 솔루션에 의존한 구현 기법으로 한정되어 발전되기 때

문이다. 프로젝션 매핑의 시스템 구축과 구현 방식이 발전하는 것과 발맞춰 콘텐츠 역시도 새로운 볼거리를 개발해야 할 것이다.

현재 프로젝션 매핑에 새로운 볼거리를 더하기 위한 시도들이 나타나고 있는데, 동적 대상에 프로젝션 매핑을 하거나 관객과 상호작용하는 콘텐츠를 개발하는 것을 그 예로 들 수 있다. 몇몇 사례에서 나타나는 이러한 시도들은 렌더링(Rendering)을 통해 생산되는 기존의 영상과 솔루션만으로는 구축하기 어렵다.

근본적으로 렌더링 된 영상을 사용하면 몇 가지 한계가 발생한다. 프로젝션 매핑은 영상을 프레임에서 벗어나 비정형 형태를 가진 대상에 자유롭게 대응할 수 있는 기술은 갖추었으나 아직 영상의 재생시간에 갇혀 있는 경우가 대부분이다. 현장성이 중요한 프로젝션 매핑은 관객이 콘텐츠에 관심을 갖는 보이는 시간은 상영시간을 넘지 않는다. 상영사간을 지나 재상영시 나타나게 될 반복된 디자인은 재방송 프로그램만큼 급격히 관심에서 멀어질 수 있다. 이러한 점은 구축에 비교적 높은 비용이 발생하는 프로젝션 매핑 콘텐츠가 지속가능한 콘텐츠로 사용되기 어려운 근본적인 문제다.

또 다른 문제로, 많은 사례들이 나타나면서 일부 서로 비슷한 연출이 나타나기도 한다. 입체적인 대상에 투사할 때 대상의 조형적 유사성에서 비슷한 유형의 영상 디자인이 나타날 가능성이 있다. 대상이 달라도 만들지는 영상 효과들이 비슷한 디지털 영상 제작 방식으로 제작될 가능성이 크다.

상영시간을 넘어 항상 새로운 모습으로 콘텐츠 자체에 더욱 긴 생명력을 갖기 위해 어떠한 방법을 적용할 수 있을까? 또한 상설 및 장기간 사용시 비용과 연결되는 콘텐츠의 갱신 주기를 늦추기 위한 새로운 제작 방법은 없을까? 이러한 질문과 새로운 프로젝션 매핑에 대한 필요성에 본 연구의 배경이 있다.

렌더링 된 영상이 갖는 시간의 한계는 프로젝션 매핑 콘텐츠를 보다 지속가능한 콘텐츠로 사용하지 못하게 하고 재사용성을 막는 근본적인 아쉬움이다. 만들어진 영상 콘텐츠를 계속 바꿀 수 없다면 계속 바뀔 수 있는 영상을 만드는 것은 어떨까? 이러한 방법이 가능하게 하려면 기존의 영상 제작 기법과 다른 영상 제작 방법이 필요할 것이다.

이에 본 논문에서는 기존 영상이 갖는 몇 가지 한계를 개선하여 보다 지속 가능한 새로운 프로젝션 매핑을 개발하기 위한 방법으로 실시간 모션그래픽(Real-time motion graphics) 방식으로 프로젝션 매핑의 영상을 제작 할 것을 제안한다.

이는 프로그래밍 코드에 의해 그래픽을 생성(generate) 시키는 방식으로 영상을 제작하고, 이것을 프로젝션 매핑의 콘텐츠로 사용하는 것이다. 실시간 렌더링으로 생성되는 영상은 재생시간의 한계를 넘게 하며 기존 영상과 다른 새로운 디자인의 구현과 의미를 담을 수 있는 가능성을 갖는다.

일부 사례들에서 생성 기반의 모션그래픽을 사용한 사례들이 나타나고 있는데 적용상에 기술적 표준이 따로 없고 구현 방법도 각기 달라서 본 연구를 통해 실시간 모션그래픽을 프로젝션 매핑에 사용하는 기법에 대한 기초 연구를 수행하여 적용을 돕고 다양한 가능성을 확인할 수 있으며 이를 바탕으로 프로젝션 매핑의 발전 방향을 제시하는데 목적을 가지고 있다.

1.2 연구의 범위 및 방법

본 연구는 실시간 모션그래픽을 프로젝션 매핑에 적용하기 위한 기술의 응용 연구로 선행된 이론 연구와 저자의 구현 과정을 바탕으로 작성된 경험 논문의 성격을 가지고 있다.

프로젝션 매핑은 전자식 빔 프로젝터를 사용하여 형태를 가진 공간 또는 사물에 투사하여 만들어지는 비정형 영상 연출 기법으로 정의하며 빔 프로젝터로 빛을 투사하여 만들어진다는 점에서 건물 외벽에 한정하고 LED 모듈을 사용하기도 하는 미디어 파사드(Media Facade)와는 개념의 차이가 있으며 고광량 빔 프로젝터가 나오기 전에 대규모 무대나 건물 등에 빔에 의한 이미지 매핑을 하기 위해 사용된 PIGI Projector와 같은 Large-format slide projector는 동영상이 아닌 정지 이미지를 담은 필름이 상하좌우로 이동하며 만들어지는 움직임을 투사함으로 현대적인 프로젝션 매핑의 기능은 없이 단순 투사의 역할만 수행한다는 점에서 본 논문에서 다루는 프로젝션 매핑의 범위에서 제외한다.

선행 연구의 이론적 부분은 프로젝션 매핑에 대해 국내외 관련 문헌 및 인터넷을 통해 취합한 자료를 근거로 작성하였다. 영상 기반의 프로젝션 매핑 소프트웨어와 이를 사용한 구현 기법의 분석을 위해, 수십여 종의 소프트웨어를 비교 테스트하였으며, 다양한 실무 사례를 찾아 각 사례에서 보이는 콘텐츠로써의 영상과 구현 방식으로써의 기술의 특징들을 다각도로 분석하다.

이러한 선행연구를 바탕으로 실시간 모션그래픽을 사용하는 프로젝션 매핑과 기존의 렌더링 영상을 사용하는 프로젝션 매핑의 차이점을 고찰하고, 실시간 모션 그래픽을 사용하는 프로젝션 매핑의 몇 가지 핵심 기법을 직접 테스트하여 그 구현 방식을 구체화하였다.

본 연구에서 실시간 생성 모션그래픽의 구현은 각각 미디어아트 콘텐츠 개발 프로그래밍 언어와 게임엔진을 대표하는 소프트웨어로써 VVVV와 유니티(Unity)를 사용하였고, 이들로 구축한 프로젝션 매핑을 통해 구현 방식의 장단점과 기술의 가능성을 정리하고 차후 관련 연구들에 필요한 사항을 언급하였다.

1.3 연구의 구성 및 흐름도

본 연구의 1장에서는 연구의 배경과 목적, 연구의 범위 및 방법, 논문의 구성에 대하여 기술하였다.

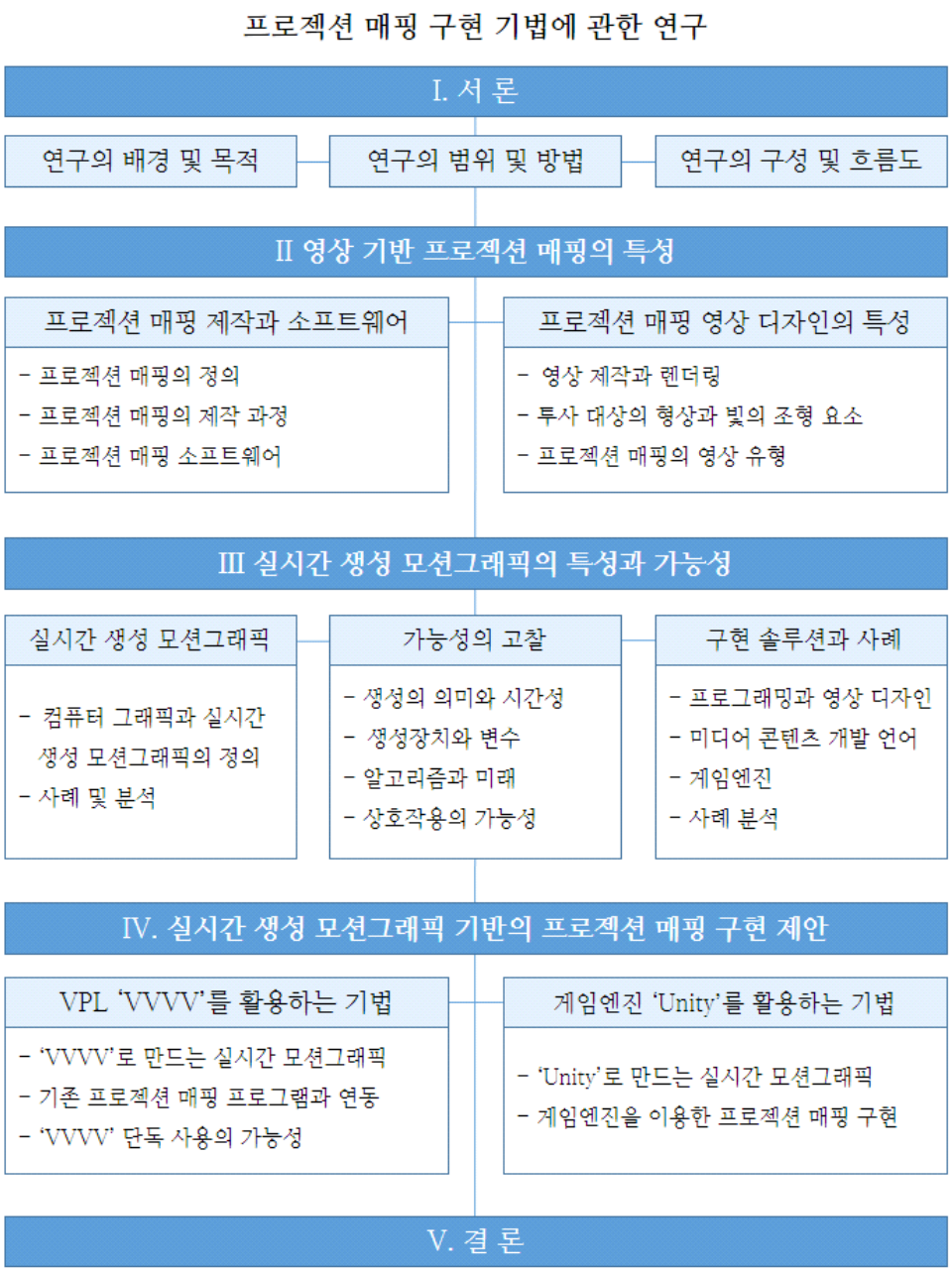
2장에서 프로젝션 매핑의 정의에서, 제작 방식과 제작에 사용되는 소프트웨어를 정리하였고 프로젝션 매핑에서 사용하는 영상의 특성을 분석하여 3장에서 다룰 실시간 모션그래픽과의 차이를 알 수 있는 기준을 마련하였다.

3장에서는 실시간 모션그래픽의 정의하고 가능성을 고찰하기 위해 실시간 모션그래픽의 주요 개념으로 4가지를 나누어 기술하였다. 먼저 ‘생성의 의미와 시간성’을 주제로 생성의 개념에 접근하며, 제작 과정에서 중요한 ‘생성장치와 변수’이라는 주제를 통해 제작상의 특성을 고찰하였다. ‘알고리즘과 미래’에서 실시간 모션그래픽이 가진 가능성을 예측하고 ‘상호작용의 가능성’에서는 관객과 소통할 수 있는 콘텐츠로 개발할 수 있는 가능성을 고찰하였다. 실시간 모션그래픽을 구현하기 위한 소프트웨어가 기존의 영상을 제작하는

프로그램들과 달라서 이들의 차이를 기술하였고 3장의 마지막으로 실시간 모션그래픽이 적용된 프로젝션 매핑의 사례들을 찾아서 나타난 특성과 기술을 분석하였다.

4장에서는 3장에서 연구한 실시간 모션그래픽 프로그램을 이용하여 실제 실시간 모션그래픽의 구현을 하고 이를 프로젝션 매핑에 적용하는 방법을 제안하였으며 구현 및 연구 과정에서 나타난 장단과 문제점들을 정리하였다.

마지막으로 5장에서는 2, 3장의 연구 내용을 종합하고 이를 바탕으로 4장에서 제안한 기법을 통해 실시간 모션그래픽의 프로젝션 매핑 적용 가능성과 향후 필요한 연구의 방향성을 제시하였다.


실시간 생성 모션그래픽 기반의
프로젝션 매핑 구현 기법에 관한 연구
I. 서 론
연구의 배경 및 목적
연구의 범위 및 방법
연구의 구성 및 흐름도
II 영상 기반 프로젝션 매핑의 특성
프로젝션 매핑 제작과 소프트웨어
- 프로젝션 매핑의 정의
- 프로젝션 매핑의 제작 과정
- 프로젝션 매핑 소프트웨어
프로젝션 매핑 영상 디자인의 특성
- 영상 제작과 렌더링
- 투사 대상의 형상과 빛의 조형 요소
- 프로젝션 매핑의 영상 유형
III 실시간 생성 모션그래픽의 특성과 가능성
실시간 생성 모션그래픽
- 컴퓨터 그래픽과 실시간 생성 모션그래픽의 정의
- 사례 및 분석
가능성의 고찰
- 생성의 의미와 시간성
- 생성장치와 변수
- 알고리즘과 미래
- 상호작용의 가능성
구현 솔루션과 사례
- 프로그래밍과 영상 디자인
- 미디어 콘텐츠 개발 언어
- 게임엔진
- 사례 분석
IV. 실시간 생성 모션그래픽 기반의 프로젝션 매핑 구현 제안
VPL 'VVVV'를 활용하는 기법
- 'VVVV'로 만드는 실시간 모션그래픽
- 기존 프로젝션 매핑 프로그램과 연동
- 'VVVV' 단독 사용의 가능성
게임엔진 'Unity'를 활용하는 기법
- 'Unity'로 만드는 실시간 모션그래픽
- 게임엔진을 이용한 프로젝션 매핑 구현
V. 결 론

〈표 1〉 연구의 흐름도

II. 영상 기반 프로젝션 매핑의 특성 분석

2.1 프로젝션 매핑 제작과 소프트웨어

2.1.1 프로젝션 매핑의 정의

프로젝션 매핑은 입체적인 형상을 가진 대상의 표면에 영상을 투사하여 변화 만들고 대상이 다른 성격을 가진 것처럼 보이게 만드는 기법이다.[3)]

이와 유사한 개념으로 미디어 파사드(media facade)가 있는데 이것은 건축물에 한정하며 때로 빔 프로젝터가 아닌 다수의 LED를 건축물 표면에 설치하여 영상을 나타나게 하는 방식을 포함하는 개념이다. 이러한 차이를 구분하기 위해 빔 프로젝터를 이용하는 경우에 프로젝션 파사드(projection facade)로 불리기도 한다.[4)]

프로젝션 매핑은 빛을 투사하여 만들어진다는 점에서 조명 효과와 비슷하게 볼 수 있지만 구체적인 현상을 가진 이미지나 움직임을 가진 영상을 대상의 형태에 맞추어 투사한다는 점에서 조명 효과와는 크게 구분된다. 영상을 사용하지만 스크린이 아닌 공간 및 사물에 투사하여 만들어짐으로 화면 비율이 따로 없고 대상의 형상에 따라 모습이 결정되어 비정형 영상 연출이 필요한 분야에 주요한 기법으로 사용된다.

프로젝션 매핑 기법은 쇼(show) 연출에 많이 사용되고 있는데 주로 볼거리가 중심이 되는 공연, 이벤트, 전시 분야가 이에 해당한다. 공연 분야에서는 무대 미술을 대치하여 영상으로 무대를 꾸미는 기법으로 널리 사용되며, 각종 전시에서는 공간 연출에서 디지털 사이니지와 미니어처 연동 프로젝션 매핑 등 정보를 다루는 전시 콘텐츠까지 다양하게 사용되고 있다.

프로젝션 매핑의 영상은 다양한 모션그래픽 기법으로 제작되는데 대상의

---

3) 김은수. (2013). 「프로젝션 맵핑이 적용된 이벤트 공간경향 분석」. 한국실내디자인학회 2013년도 추계학술발표대회 논문집. p.144.

4) 김민지. 박남희. (2012). 「공간매핑의 계보를 통해 본 프로젝션 매핑과 미디어 파사드」. 한국디지털디자인협의회 디지털디자인학연구 제12권 제4호. p.561.

표면에 2D 패턴을 덮어 나타낸 표현부터 착시 그림과 같이 입체감과 공간감을 표현할 수 있는 3D 영상까지 다양한 표현이 나타난다. 어두운 곳에서 프로젝터를 사용해야 하는 특성을 바탕으로 충분한 밝기의 프로젝터를 사용한다면 명암의 대비를 크게 할 수 있고 밝은 이미지의 채도를 높여 화려한 색감을 표현할 수 있다 여기에 역동적인 모션그래픽 연출을 더해 관객의 시선을 자극하는 기법이 된다.

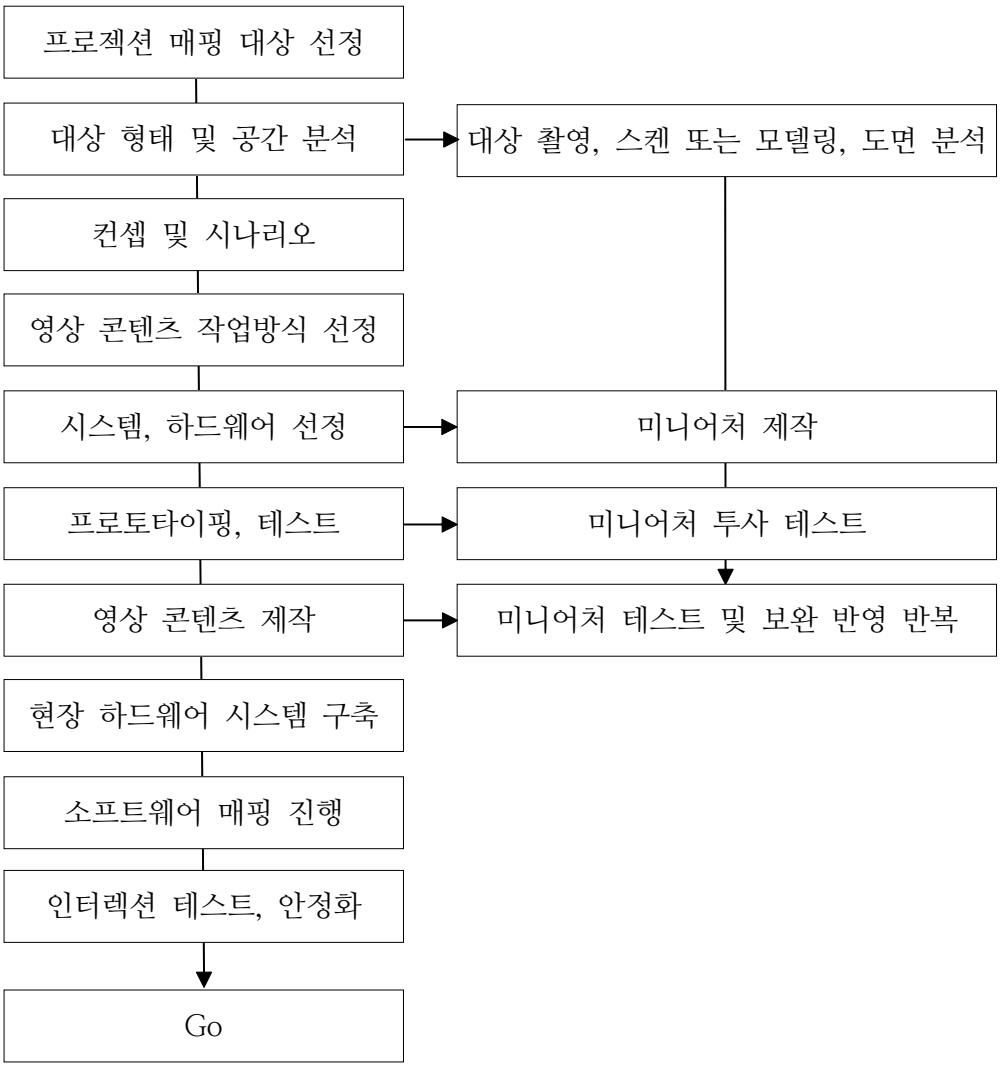
〈그림 4〉 뮤지컬 '빈센트 반 고흐'의 프로젝션 매핑 무대영상

2.1.2 프로젝션 매핑의 제작 과정

프로젝션 매핑은 특성상 매핑의 대상이 되는 공간 또는 사물의 선정과 투사면의 형태를 분석하는 일이 선행된다. 대상이 가진 형태는 형태 자체로 영상 디자인의 핵심 요소가 되고 스토리 연출의 골격이 된다. 또한 제작한 영상을 투사할 때도 대상의 물성과 공간의 관계에서 다양한 변수가 발생하는데 주변의 조도, 형상의 구조, 투사 거리, 관객의 위치 등 다양한 변수들을 고려해야 한다. 따라서 사전에 항상 대상의 형태에 대한 충분한 연구가 진행되며 필요에 따라 미니어처를 이용하여 시뮬레이션을 한다.

각각의 문제들에 대한 해결 방안을 결정하고 물리적인 문제들을 해결하면 투사 할 영상을 프로젝션 매핑 전문 소프트웨어를 이용하여 투사하고 콘텐츠 구축을 진행된다. 대상의 형태가 복잡할 경우 현장에서 프로젝션 매핑의 기법으로 영상을 대상에 맞추는 작업에 상당한 시간이 소요될 수 있으며, 관객과

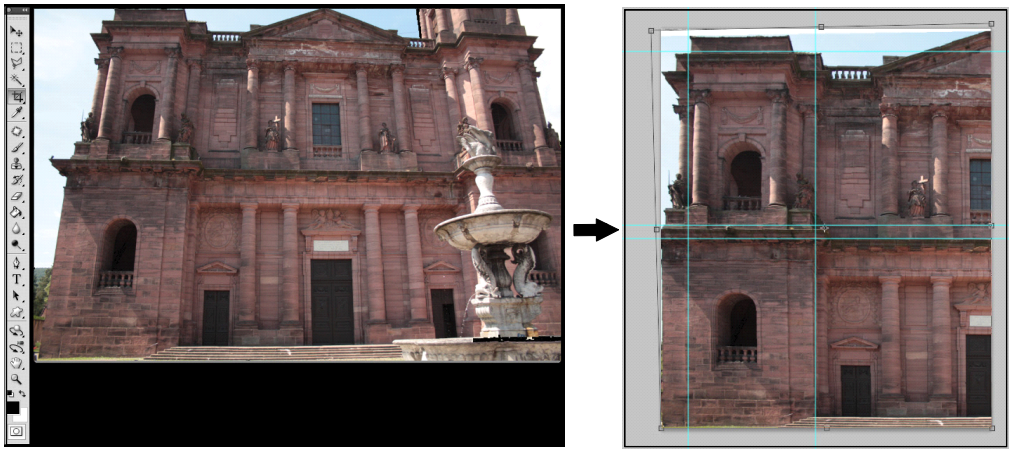
의 상호작용 기능 및 생성 기반의 영상을 사용하는 경우 추가적인 작업 시간이 필요할 수 있다. 이러한 여러 과정을 다음의 표로 정리하였다.



〈표 2〉 영상 기반의 프로젝션 매핑 제작 프로세스

〈표 2〉 과정은 프로젝션 매핑의 목적에 따라 다양하게 변형된다. 유럽에서 많이 선보인 오래된 고성을 대상으로 진행한 미디어 파사드의 경우, 대상의

형태 및 공간 분석 과정에서 정확한 실측 도면이 없으므로 〈그림 4〉와 같이 대상을 사진 촬영 후 이를 포토샵 등의 툴에서 영상 제작을 위한 표준 가이드를 작성하거나 사진을 이용해 3D 모델링을 제작하여 테스트를 진행하는 방식으로 한다.



〈그림 5〉 사진을 이용한 프로젝션 매핑 대상의 가이드 제작[5)]

자동차와 같이 정밀한 프로젝션 매핑을 할 경우, 자동차 모델링 데이터를 직접 사용하여 투사할 영상을 제작하기도 하며 조형물에 기하학적인 패턴 위주로 연출을 할 경우 가이드를 생략하기도 한다. 작업 과정이 투사 대상에 따라 달라짐으로 제작 프로세스는 변동 가능하나 기본적으로 대상의 형상을 이해하는 과정이 선행되며 이후 콘셉트와 시나리오에 따라 영상의 제작 방식이 결정되고 투사를 위한 시스템이 결정된다.

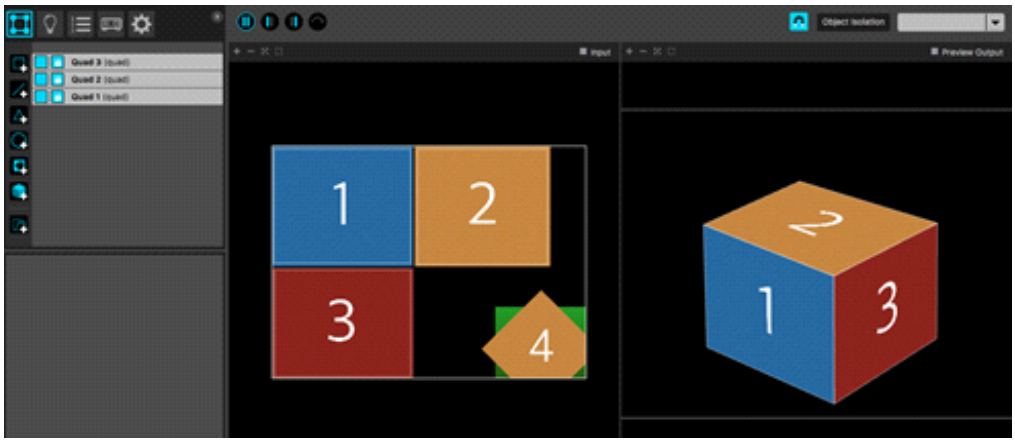
프로젝션 매핑의 시스템은 투사하는 대상의 크기와 투사 위치, 공간의 밝기 등 다양한 공간과의 관계 요소에 따라 판단을 하게 된다. 프로젝터 기종이 선정되면 이를 구동할 소프트웨어는 콘텐츠가 구현할 목표에 따라 판단하여 결정하게 된다. 대체적으로 하드웨어는 사례마다 비슷한 구성이 되지만 소프트웨어는 상황에 따라 강점을 가진 소프트웨어 및 구현 방식을 결정하여 사용하게 된다. 한번 결정된 사항은 큰 비용과 직결됨으로 대규모 작업일수록

5) 1024 Architecture. Mapping a building | After Effects + Madmapper tutorial」. (2017. 9. 3). https://1024d.wordpress.com/2011/06/10/mapping-a-building-after-effects-madmapper-tutorial/

미니어처를 이용한 테스트를 하는 것이 권장된다.

2.1.3 프로젝션 매핑 소프트웨어

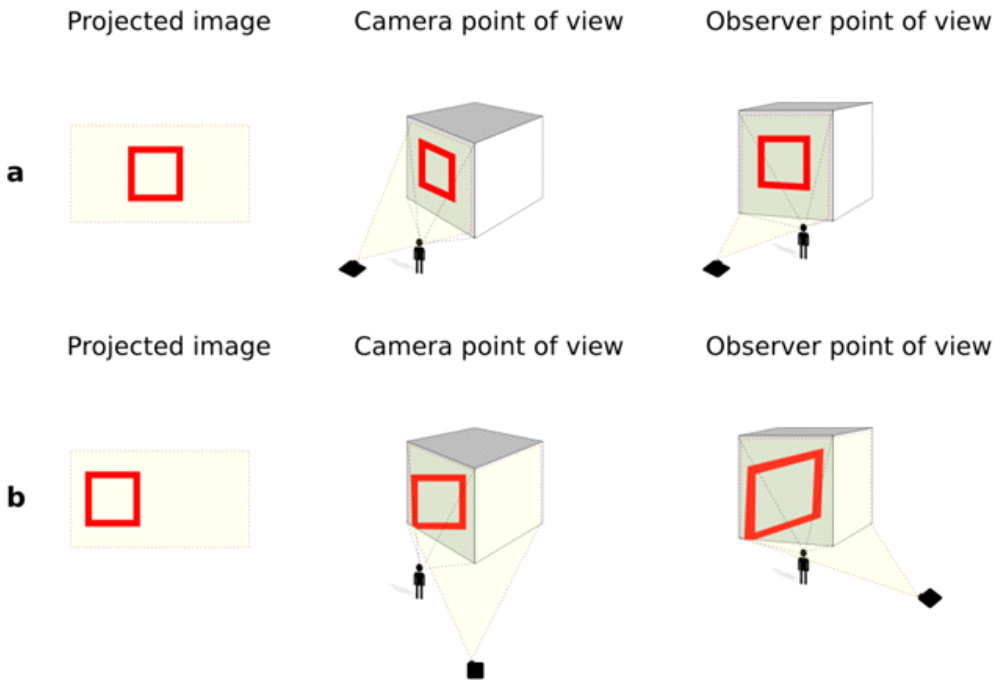
프로젝션 매핑을 수행할 수 있는 소프트웨어는 구현 방식에 따라 3가지 유형으로 구분할 수 있다. 첫 번째는 렌더링 된 영상을 사용하는 방식으로 단일 동영상 또는 복수의 동영상을 투사와 동시에 대상의 형상에 맞출 수 있도록 영상의 분할, 배치, 회전, 변형 등의 과정을 거쳐 프로젝션 매핑을 수행하는 방법이다. 다수의 프로젝션 매핑 전문 소프트웨어들이 이 방식을 채용하고 있다. 단일 동영상 파일을 이용할 경우 하나의 영상 파일에 프로젝션 매핑에 필요한 각 부분별 영상을 모두 배치한 후 렌더링 하여 사용함으로 시스템에 큰 부하를 주지 않는 장점을 갖는다.



〈그림 6〉 단일 동영상을 사용하는 프로젝션 매핑 방식, Madmapper

렌더링 된 영상을 사용하는 방식의 프로젝션 매핑은 만들어진 영상을 대상이 갖는 표면에 단순히 덮어 투사하는 구조이므로 공간감과 입체감의 표현을 담은 이미지를 투사할 때, 사전에 대상의 형상과 입체적인 표현의 속성을 고려하여 3D 영상으로 제작 후 사용하게 된다. 입체감은 명암과 원근, 대상의

방향성 등을 고려해야 하며 입체감의 확보를 위해 때로 관객의 위치를 제한하는 경우도 있다.



〈그림 7〉 프로젝션 매핑에서 공간감 표현 원리[6)]

3D 영상을 사용하여 입체적이고 공간적인 형상의 표현하는 것은 3D로 영상을 제작하고 렌더링 후 출력된 영상을 사용하는 것인데, 일부 프로젝션 매핑 소프트웨어에서는 3D 모델링 데이터를 직접 입력받고, 프로그램 내에서 실시간으로 모델링 데이터에 동영상을 입혀 투사하는 3D 방식의 프로젝션 매핑 소프트웨어도 있다. 대표적으로 MWM이 있는데 3D 모델링 데이터에 텍스처로써 동영상을 실시간으로 매핑하고, 이것을 대상에 바로 투사하여 프로젝션 매핑을 진행한다.

대표적으로 MWM이 있는데 3D 모델링 데이터에 텍스처로써 동영상을 실

6) Marcin Ignac. Projection Mapping In 3d. (2017. 10. 1). http://marcinignac.com/blog/projection-mapping-in-3d/

시간으로 매핑하고, 이것을 대상에 바로 투사하여 프로젝션 매핑을 진행한다. 그러나 현재 MWM은 후속 개발 및 업데이트가 안되고 있는 실정이다.

세 번째 유형으로는 VVVV, Touch designer와 같이 영상 기반의 콘텐츠 개발 프로그래밍 툴로 프로젝션 매핑을 구축하는 방법이 있다. 프로젝션 매핑을 위한 전문 소프트웨어는 아니지만 각종 미디어를 다루는 강력한 기능들과 확장성, 프로젝션 매핑을 구축할 수 있는 다양한 방법들이 알려져 있어서 실재 프로젝션 매핑에 종종 사용되고 있다. 단 프로젝션 매핑 기능이 갖춰진 것이 아닌 해당 프로그램으로 프로젝션 매핑을 수행하기 위한 프로그래밍을 선행해야 한다.

위에서 언급한 방식들로 프로젝션 매핑을 구현하는 소프트웨어들은 수십여 가지가 출시되어 있으며 각각의 장단점을 가지고 실무에서 사용되고 있다. [표 3]에서는 프로젝션 매핑을 구현할 수 있는 기능을 가진 다양한 소프트웨어들을 찾아 주요 특징을 비교하고 표로 정리하였다.

하이엔드 솔루션을 표방하는 일부 프로젝션 매핑 소프트웨어들은 프로젝션 대상의 3D 모델을 프로그램 내에 가상공간에 배치하여 프로젝션 상황을 소프트웨어 적으로 시뮬레이션 할 수 있거나 자체적으로 모션 그래픽을 만들어 낼 수 있는 기능 또는 부가 프로그램과 연동하여 프로젝션 매핑에 필요한 콘텐츠를 직접 제작할 수 있었다. 일부 하이엔드 솔루션은 소프트웨어 환경의 최적화를 위해 자체 미디어 서버와 함께 구축해야 하는 경우가 있어 일반적인 상황에서 사용하기에는 다소 불편할 수 있다.

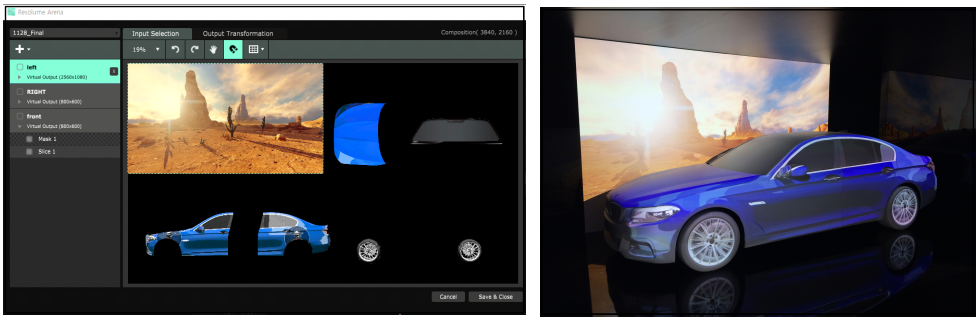
software	소스 타입	운영 기반	주요 특징
DynaMapper	동영상	ios, Android	타블렛용
Painting With Light	동영상	PC, Mac	베타 버전
Millumin	동영상	Mac	타임라인, 애프터 이팩트 플러그인 연동
MXWendler	3D, 동영상	PC, Mac	타임라인, 3D Keystoning
TorsionBlend	3D, 동영상	Mac	개발 중단
TouchDesigner	프로그래밍, 3D, 동영상	PC, Mac	비주얼 프로그래밍 환경
HeavyM	동영상	PC, Mac	전용 입체 구조물 활용 및 판매
FacadeSignage	동영상	PC	디지털 사이니지용
BlendBuddy	동영상	Mac	프로젝터의 프로젝션 보정용
VPT	동영상	PC, Mac	오픈 소스
Madmapper	동영상	PC, Mac	프로젝션 맵핑의 대중화
Resolume Arena	동영상	PC, Mac	다양한 내장 기능
LPMT	동영상	PC, Mac, Linux, iOS	크로스 플렛폼
Arkaos GrandVJ XT	동영상	PC, Mac	다양한 내장 기능
d3 Technologies	3D, 동영상	PC	하이엔드 솔루션, 시뮬레이션
coolux	3D, 동영상	PC	하이엔드 솔루션, 자체 서버
Blendy VJ	동영상	Mac	돔 구조 맵핑으로 전문화
Mesh Warp Server	동영상	PC, Mac	개발 중단
vvvv	프로그래밍, 3D, 동영상	PC	비주얼 프로그래밍 환경
Visution Mapio	동영상	PC, Mac	프로젝션 맵핑 전문
Vioso	동영상	PC	기능에 따라 프로그램 구분
Green Hippo	3D, 동영상	PC	하이엔드 솔루션, 시뮬레이션
Avolites	3D, 동영상	PC, Mac	하이엔드 솔루션, 자체 서버
ofxPiMapper	동영상	Rasberry Pi	라즈베리파이를, 실험적 성격
PocketVJ	동영상	Rasberry Pi	Rasberry Pi 포함, 네트워크 연동
Virtual Mapper Open Frameworks	동영상	PC, Mac, ios	openFrameworks로 개발된 실험적 성격
Splash	동영상	Mac, Android	오픈 소스
Dataton Watchout	동영상	PC	하이엔드 솔루션, 자체 서버
Modulo Pi	동영상	PC	하이엔드 솔루션, 자체 서버
SPARCK	3D, 동영상	PC, Mac	증강현실 맵핑으로 전문화
AV Stumpfl WINGS AV	동영상	PC	하이엔드 솔루션, 자체 서버
Smode	3D, 동영상	PC	하이엔드 솔루션, 자체 서버
MapMap	동영상	PC, Mac, Linux	오픈 소스
Optoma Projection Mapper	동영상	ios, Android	옵토마 프로젝터용
ISADORA	프로그래밍, 3D, 동영상	PC, Mac	비주얼 프로그래밍 환경
EDGE C by Light Instruments	동영상	PC, ios	iOS로 맵핑 셋팅 후 자체 서버 플레이

〈표 3〉 프로젝션 매핑 소프트웨어 현황

2.2 프로젝션 매핑에서 나타나는 영상 디자인의 특성

2.2.1 영상 제작과 렌더링

간단한 검색을 통해서도 수많은 사례들에서 건축물 외부와 내부, 자동차 외부와 내부, 전시 공간의 다양한 구조물 및 사물, 상품, 기하학적인 조형물에 이르기까지 입체적인 형태를 가진 많은 대상에 프로젝션 매핑이 시도되는 것을 볼 수 있다. 프로젝션 매핑은 대상의 표면에 투사된 빛에 의해 표면의 모습이 대치되어 만들어내는 기법이므로 대상에 형태를 반영하여 영상이 만들어진다. 프로젝션 매핑의 영상은 그 자체로 완성된 상태가 아닌, 투사 후 대상에 형태에 맞추기 위한 상태이다. 따라서 대부분의 프로젝션 매핑 영상은 TV와 같은 일반적인 규격의 영상으로 단독 사용이 어렵다. 일반적인 프로젝션 매핑 소프트웨어에서는 대상의 입체적인 면에 따라 분할된 이미지를 단일 화면에 배치하여 사용한다. 면의 수에 따라 분리하여 배치하기도 하고 곡면을 포함하는 경우 3D로 제작하여 그대로 사용하기도 한다.



〈그림 8〉 자동차 프로젝션 매핑을 위한 분할된 단일 영상, Resolume

〈그림 8〉은 배경과 차량에 프로젝션 매핑을 하고 있은 것을 보여주고 있다. 좌측 사진은 영상 기반의 프로젝션 매핑 소프트웨어 Resolume의 모습으

로 차량에 투사하기 위해 준비한 단일 영상을 보여주고 있다. 차량의 각 부분을 분할하여 하나의 영상에 배치한 것으로, 영상 기반의 프로젝션 매핑은 대체로 이와 같이 단일 영상에 대상의 각 부분에 들어갈 영상을 배치한 영상을 이용한다. 제작된 영상은 보통 운용하는 소프트웨어의 코덱과 포맷을 따르며 해상도는 사용하는 프로젝터의 해상도에 맞춰 렌더링을 진행한다.[7)]

프로젝션 매핑은 사용 목적과 대상에 따라 여러 가지 모습으로 나타나지만 각각의 사례에 따라 크기와 형태가 달라도 구현 기술과 영상의 제작 및 사용 방법은 대체로 비슷하다.

2.2.2 투사 대상의 형상과 빛의 조형

공간과 사물은 저마다의 형태를 가지고 있지만 3차원 공간에 존재하는 대부분의 형상은 최소한의 점, 선, 면에서 시작하여 그려 낼 수 있는 만큼, 각각의 대상에 투사되는 면에서 조형 요소를 찾아보면 프로젝션 매핑에 필요한 영상의 조형 기술을 이해할 할 수 있다.

프로젝션 매핑 소프트웨어가 구현할 수 있는 기법들은 비교적 간단한 원리와 기능을 가지고 대상의 크기와 무관하게 프로젝션 매핑을 구현할 수 있다. 예를 들어 많이 사용되는 프로젝션 매핑 프로그램인 Madmapper의 경우 투사되는 영상을 자유롭게 변형하며 프로젝션 매핑을 구축하기 위해 제공되는 기능은 선, 삼각형, 사각형, 원형 선택 도구와 필요한 부분을 안 보이게 하는 마스크 도구가 전부이고 이러한 도구로 영상을 분할하고 형상에 맞춰 변형하여 투사한다. 나머지 기능들은 투사되는 영상을 변형하는 기능이 아닌 영상 재생 제어, 다수의 프로젝터를 사용할 때 필요한 멀티 프로젝션, 두 개 이상의 프로젝터를 사용할 때 경계면에 생기는 이중상을 없애는 엣지 블렌딩(edge blending) 등과 같은 부가적인 기능들이다.

---

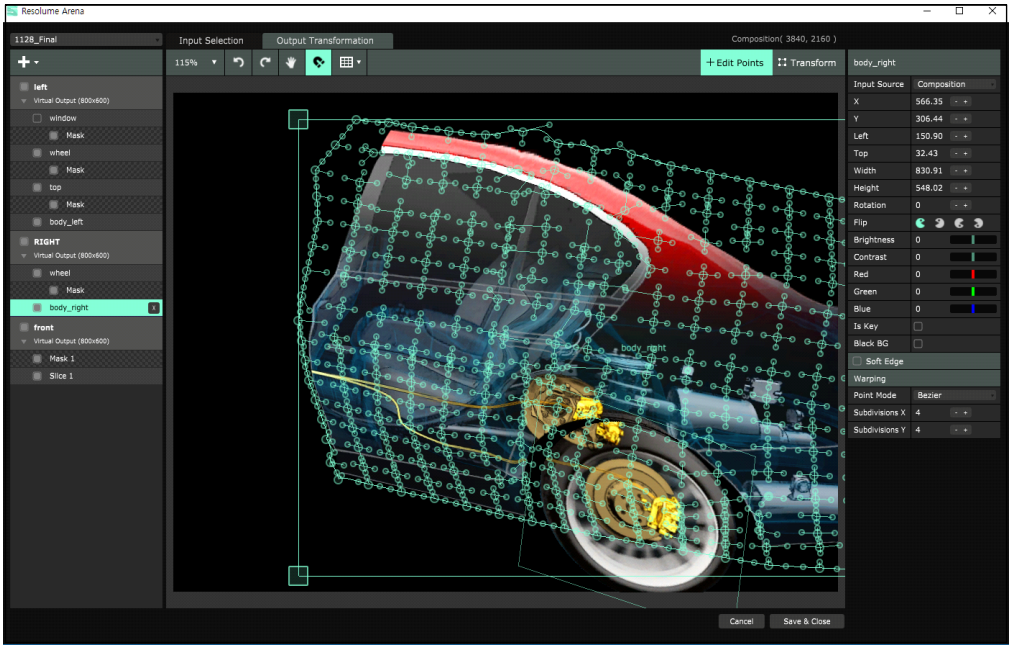
7) 1024 Architecture. Mapping a building | After Effects + Madmapper tutorial. (2017. 9. 3). https://1024d.wordpress.com/2011/06/10/mapping-a-building-after-effects-madmapper-tutorial/

〈그림 9〉 Madmapper에서 프로젝션 매핑에 필요한 기능 도구

프로젝션 매핑을 해야 하는 대상에서 보이는 형상은 대부분 기초 도형 또는 입체 도형의 단위로 찾아 정리할 수 있고 이들은 최소한의 조형 단위만 가지고도 대상의 형태에 맞추어 투사하는 것이 가능하다.

일부 대상에 따라서 자동차와 같이 보다 복합적인 곡면 형상을 가진 대상들이 있을 수 있다. 복잡한 곡면으로 만들어진 형태는 이미지를 와핑(warping) 하여 대상의 형태에 맞출 수 있다. 곡면 스크린에 영상을 휘게 하여 투사하는 소프트와프(softwarp)[8]도 이에 포함되는 개념으로 볼 수 있는데 일부 정교한 프로젝션 매핑 소프트웨어에서는 복잡한 곡면 매핑에 대응하기 위해 단일 이미지 안에서 다수의 와핑 핸들(handle)을 생성시켜 구역별로 자유롭게 이미지 와핑을 가능하게 한다.

8) wikipedia. Softwarp. (2017. 10. 5). https://en.wikipedia.org/wiki/Softwarp



〈그림 10〉 이미지 와핑, Resolume

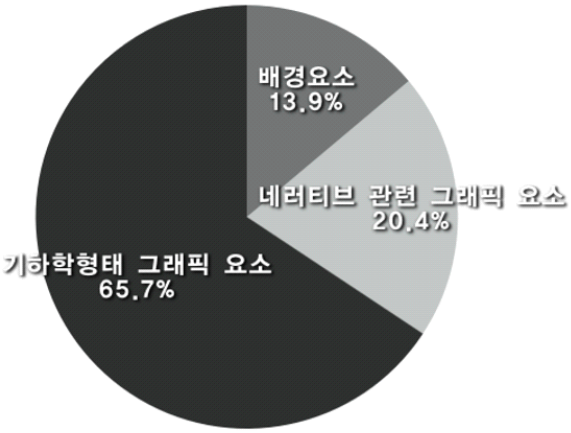
MWM, VVVV 등의 소프트웨어에서는 모델링 데이터를 직접 이용하여 대상의 형태와 유사한 3D 모델링 데이터에 동영상 및 기타 텍스처를 입히고 프로젝션 위치를 가정한 각도로 투사하여 프로젝션 매핑을 구축하는 방법도 있다. 이러한 경우 입체도형의 단위로 조형 요소가 나타나는데 투사 후 정교한 매핑을 위해 메시(mesh), 폴리곤(polygon) 단위에서 보정이 필요하다.

2.2.3 프로젝션 매핑의 영상 유형

콘텐츠의 목표가 다양하고 프로젝션이 될 대상의 형상이 다양한 만큼 다양한 디자인이 만들어지지만 영상의 종류로 보면 크게 실사 기반, 그래픽 기반 등으로 구분하여 유형을 생각해볼 수 있다. 특히 그래픽 유형에서 나타나는 특성들은 기존 프로젝션 매핑에 대한 실시간 생성 기반의 모션그래픽의 방향성과 차별성을 만드는 바탕이 된다.

1) 그래픽 요소의 유형과 특성

프로젝션 매핑 영상의 유형은 ‘프로젝션 매핑 작품에 나타난 그래픽 요소의 유형에 관한 연구’ 논문에서 찾을 수 있었다.[9] 해당 논문에서 29개의 사례를 바탕으로 프로젝션 매핑의 그래픽 특성에 대해 유형을 찾아냈는데 65.7%가 기하학적인 그래픽을 사용하였고 기학학적인 그래픽 요소로 사각형과 선, 구, 파티클 등의 사용이 많았다는 것을 알 수 있었다.



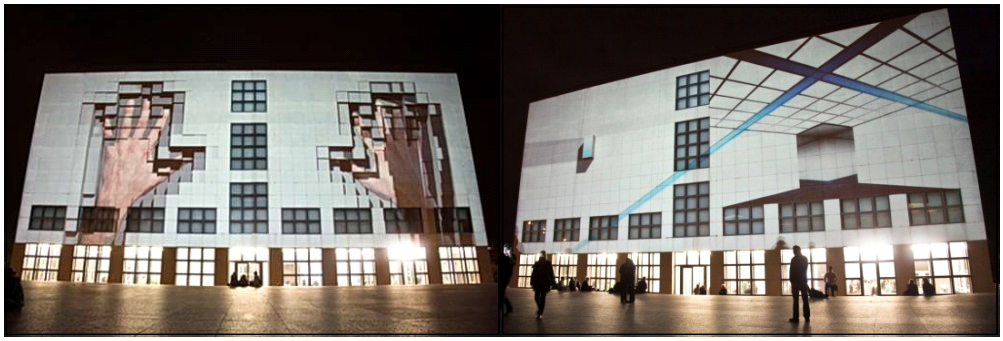
	배경 요소	네러티브 관련 그래픽 요소				기하학형태 그래픽 요소					
	구조물	인체 관련	동물, 식물	사물	텍스트	삼각형태	사각형태	구형태	선	파티클, 오일	합계
횟수합계 (회) ⊞	61	15	20	32	23	10	115	21	68	75	440(회)
출현 횟수 백분율(%)	13.9	3.4	4.5	7.3	5.2	2.3	26.1	4.8	15.5	17.0	100(%)
		3.4+4.5+7.3+5.2= 20.4				2.3+26.1+4.8+15.5+17.0= 65.7					

[표 4] 그래픽 요소의 사용 횟수 비율과 그래픽 요소의 출현 횟수[10]

9) 황용희. (2011). 「프로젝션 매핑 작품에 나타난 그래픽 요소의 유형에 관한 연구」. 한국디자인포럼 Vol.32. pp.347-356.

10) 황용희. 상게 논문. p.354.

화려한 시각적 효과를 위해 기하학적인 그래픽에 다양한 움직임이 있는 영상이 많이 사용되는 것은 볼거리를 위한 프로젝션 매핑 영상의 특성으로 볼 수 있다.

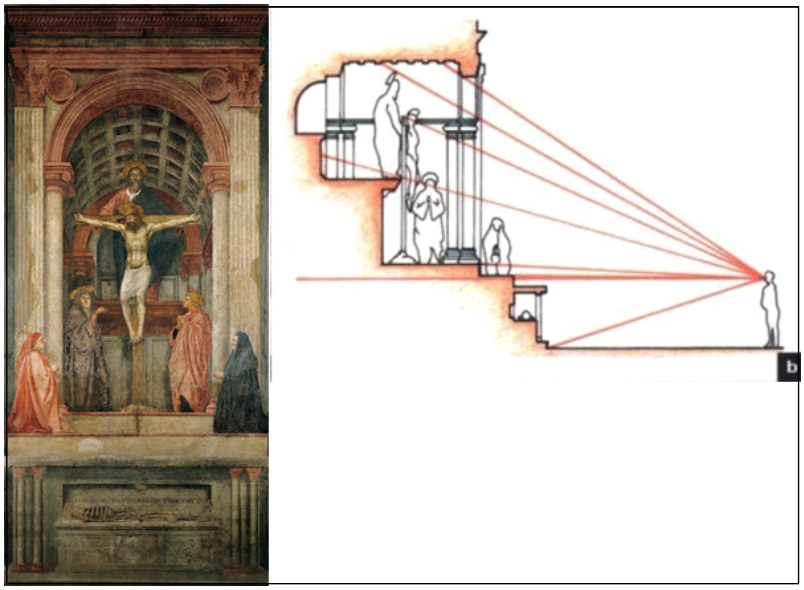


〈그림 11〉 공간감과 입체감이 나타나는 프로젝션 매핑 예

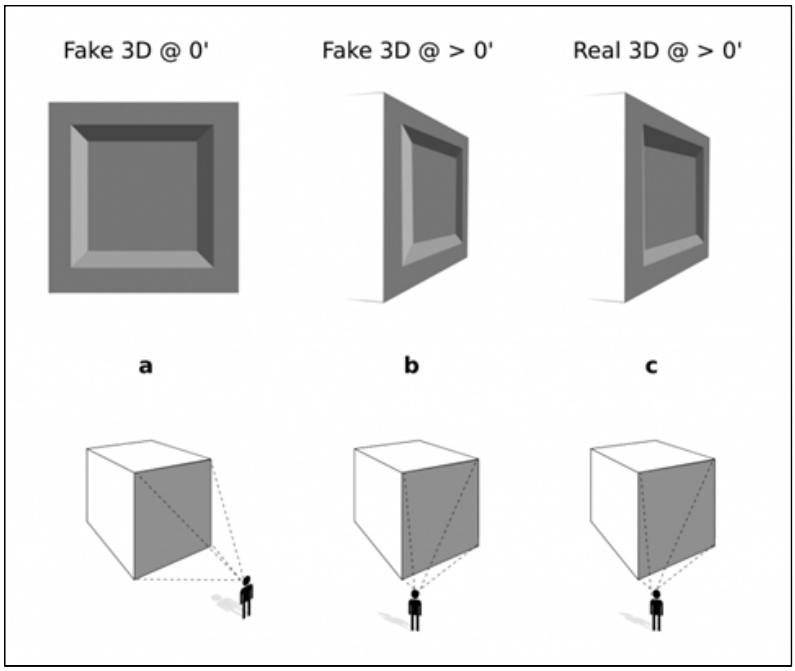
2) 공간과 입체

대상의 표면을 덮는 영상은 대상 표면의 단순 질감을 바꾸어 연출하는 2D 패턴의 형식과, 입체감을 기반으로 하는 3D 그래픽 등이 사용된다. 그중 입체적인 대상이 들어가거나 나온 것 같이 입체감과 공간감을 보이게 하는 3D 영상의 연출은 프로젝션 매핑에서 종종 사용되는 기법이다.

Masaccio(1401~1428)의 프레스코화 '성 삼위일체'는 특정 시점에 의한 원근법을 통해 그림 속에서 공간감을 만든 것으로 유명하다. 영상도 이와 같이 공간감을 만들기 위해서는 3D 방식의 작업을 통해 카메라 시점에서 빛 효과와 원근을 포함하여 제작하게 되고 원근이 가장 효과적으로 발생하는 위치에 따라 그래픽의 방향성을 부여된다. 대체로 2D 패턴 위주의 작업들은 그래픽의 방향성이 필요 없지만 공간감을 주어야 하는 경우 관객의 위치에 따라 영상 속에 라이트 효과와 원근을 고려함으로 관객의 위치를 제한할 수 있다. 〈그림 13〉과같이 관객의 위치에서 볼 수 있는 대상의 그림자 방향이 어색하면 프로젝션 매핑의 효과가 상당히 감소된다.



〈그림 12〉 Masaccio의 프레스코화 '성 삼위일체'와 선 원근법



〈그림 13〉 프로젝션 매핑에서 관객의 시점과 입체감의 관계 [11]

11) Marcin Ignac. Projection Mapping In 3d. (2017. 10, 1). http://marcinignac.com/blog/projection-mapping-in-3d/

III. 실시간 생성 모션그래픽의 특성과 가능성

3.1 컴퓨터 그래픽과 실시간 모션그래픽

3.1.1 컴퓨터 그래픽과 실시간 모션그래픽

1951년, MIT는 벡터 그래픽스 화상처리(vectorscope graphics display) 기술을 개발했다. 점과 직선, 곡선 등 선분의 집합을 통해 이미지를 생성하는 방식으로 1940년대 후반에 MIT가 개발한 월 윈드(Whirlwind) 컴퓨터를 기반으로 개발됐다. 이후 1960년 미국 보잉의 연구원인 윌리엄 페터(William Fetter)는 컴퓨터가 지시하는 좌표대로 제어되는 펜으로 설계도면이나 그래프를 그리고 출력하는 장치인 플로터(Plotter)를 이용해 비행기 조종실을 묘사하고, 이를 '컴퓨터그래픽스'(Computer Graphics)라고 지칭했다.[12)]

1963년에는 컴퓨터 과학자 이반 서덜랜드는 MIT가 1958년 생산한 링컨 TX-2 컴퓨터를 통해 자신 개발한 스케치패드(Sketchpad)라는 프로그램을 개발하여 컴퓨터 모니터에 라이트펜으로 점과 선을 생성시키고 폴리곤을 그려내었다. 이것은 CAD의 시작이 되었고 원하는 형태의 도형을 편리하게 그릴 수 있게 되어 예술가들과 디자이너들이 프로그래밍 코드의 어려움 없이 컴퓨터로 이미지를 만들어 낼 수 있게 되었다.

퍼스널 컴퓨터가 보급되기 시작한 이후, 특히 컴퓨터의 성능이 급격히 빨라지고 인터넷을 통해 지식의 공유가 폭발적으로 늘어난 2000년대 이후에는 누구나 컴퓨터그래픽스를 만들고 소비할 수 있게 되었다.

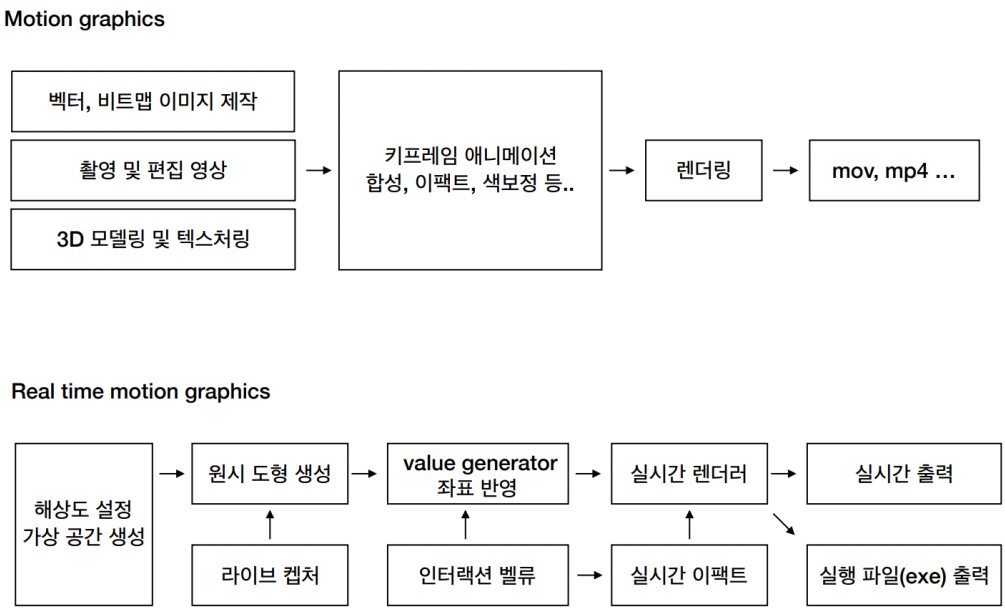
디지털 기술을 기반으로 실사 촬영에서 복잡한 모션그래픽에 이르기까지 영상 기술의 발전은 프로젝션 매핑에도 반영되었다. 다양한 소프트웨어로 실사 영상과 2D 및 3D 모션그래픽의 방식으로 제작하여 렌더링을 통해 avi,

---

12) 이인재. 「컴퓨터그래픽스」. (2017. 10. 10).
http://terms.naver.com/entry.nhn?docId=1624925&cid=42171&categoryId=42188

mov와 같은 동영상용 포맷으로 담은 영상 담아 프로젝션 매핑에 사용하였다.

시간 모션그래픽(Real-time motion graphics)은 실시간으로 생성(generate)의 개념으로 렌더링이 없이 영상을 제작하는 방법이다. 디지털 영상은 사진, 영상과 같은 비트맵 소스 및 벡터 그래픽 이미지 등을 편집하고 애니메이션 및 각종 효과를 적용한 후 렌더링을 거처 동영상 포맷에 담긴 최종 영상 파일을 만든다. 그러나 생성 기반의 영상은 프로그래밍 코드에 의해 이미지를 생성시키거나 변형 시키고 좌표의 이동을 통해 움직임을 갖는 영상이다.



〈표 5〉 모션그래픽과 실시간 모션그래픽의 제작 과정 비교

모션그래픽을 제작할 때 사용하는 소프트웨어인 에프터 이팩트나 3D 영상 제작용 소프트웨어들에서 파티클을 기반으로 만들게 되는 물, 연기, 구름, 눈과 같은 이미지들의 경우, 해당 이미지를 생성 시킬 수 있는 코드를 가진 플러그인 또는 에프터 이팩트의 익스프레션(expression)과 같이 자체 스크립트 코드를 작성하여 만들 수 있고 이러한 제작 방법도 생성 기반의 모션그래픽으로 분류할 수 있다.

그러나 이러한 방법은 정해진 시간을 갖는 영상으로 사용하기 위해서 제작 후 렌더링을 거쳐 동영상 파일을 얻고 이를 동영상 플레이어를 통해 실행하여 확인할 수 있다. 이미지를 생성시키는 알고리즘은 비슷하나 실시간 렌더링을 통해 생성시키는 영상은 그 성질과 사용방법이 일반적인 영상과 크게 달라진다.

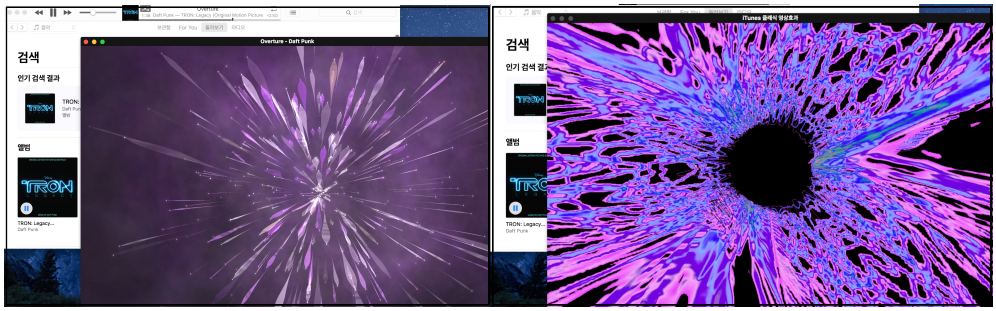
〈표 5〉에서 실시간 모션그래픽의 제작 프로세스를 보면 가상의 공간에서 원시 도형 및 라이브로 촬영된 영상 또는 녹화된 영상을 받아서 기초 소스로 사용하고 여러 값을 생성 시키는 생성 장치(generator)에서 생성된 값을 받아 움직임과 변형을 준다. 이때 외부에서 들어오는 또 다른 값을 실시간으로 추가해 반영시킬 수 있고 프로그래밍에 의해 다양한 방식으로 효과를 더하여 실시간 렌더링을 통해 즉시 모니터로 출력을 한다. 필요에 의해 결과물을 컴파일 또는 패키지의 과정을 거쳐 exe와 같이 컴퓨터에서 바로 실행되는 실행파일로 출력을 할 수 있다.

원시 이미지의 생성부터 움직임, 효과 적용 등에 필요한 각종 설정 값이 사전에 설정한 대로 만들어지고 움직이게도 가능하지만 여러 경로를 통해 들어오는 다양한 값을 즉시 반영하여 또 다른 변수를 만들 수 있는데 이러한 속성들을 통해 실시간 모션그래픽은 초기 구상을 넘어 새로운 변형과 창발, 제한 없는 재생 시간 등 또 다른 가능성을 가질 수 있다.

3.1.2 실시간 모션그래픽의 사례 및 기술

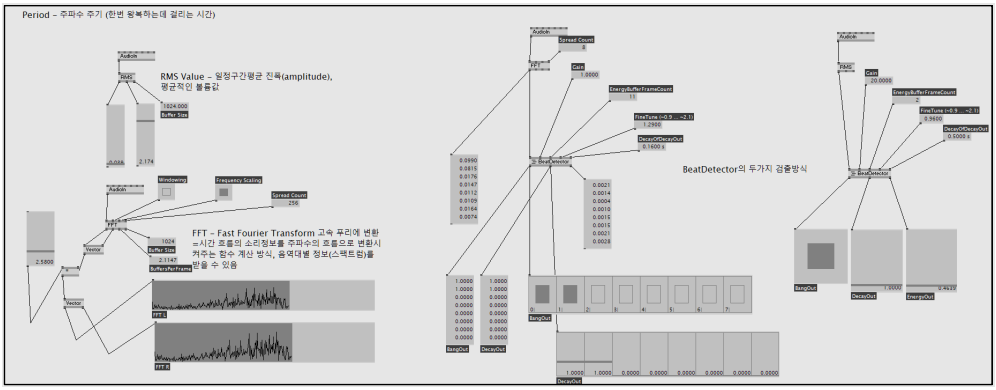
1) 생성의 기초 재료

가장 쉽게 볼 수 있는 사례는 컴퓨터로 음악을 들을 때 음악 재생 프로그램들 중 일부가 가지고 있는 시각 효과 기능이다. 이것은 음악이 가지고 있는 소리의 음량 및 음의 높낮이 등에서 실시간으로 추출한 정보를 이용하여 실시간 모션그래픽를 보여주는 것으로 소리만 듣는 음악을 다채로운 빛의 형상으로 표현해준다.



〈그림 14〉 아이튠즈, 음악 실시간 모션그래픽

실시간 생성 모션그래픽은 생성과 효과, 움직임을 위한 재료를 필요로 한다. 이러한 재료 중에 소리는 가장 사용하기 쉬운 소재이다. 마이크로 실시간 수음을 하거나 소리 데이터 자체에서 데이터를 추출할 수 있다. VVVV를 이용하여 실시간으로 소리에서 어떤 데이터를 추출할 수 있는지 실험하였다.



〈그림 15〉 VVVV를 통해 실시간으로 수음된 소리의 수치 값

마이크를 통해 들어온 소리는 FFT, RMS, BeatDetrctor 노드를 통해 수치 데이터로 나타나는데 전체 및 음역대별 소리의 크기 자체와 소리의 크기 변화에서 평균적인 진동수를 추출하여 이를 수치 데이터로 출력하였다. 이러한 수치의 변화 값은 이미지의 물리량에 반영할 수 있으며 소리를 형상화하는 재료가 된다.

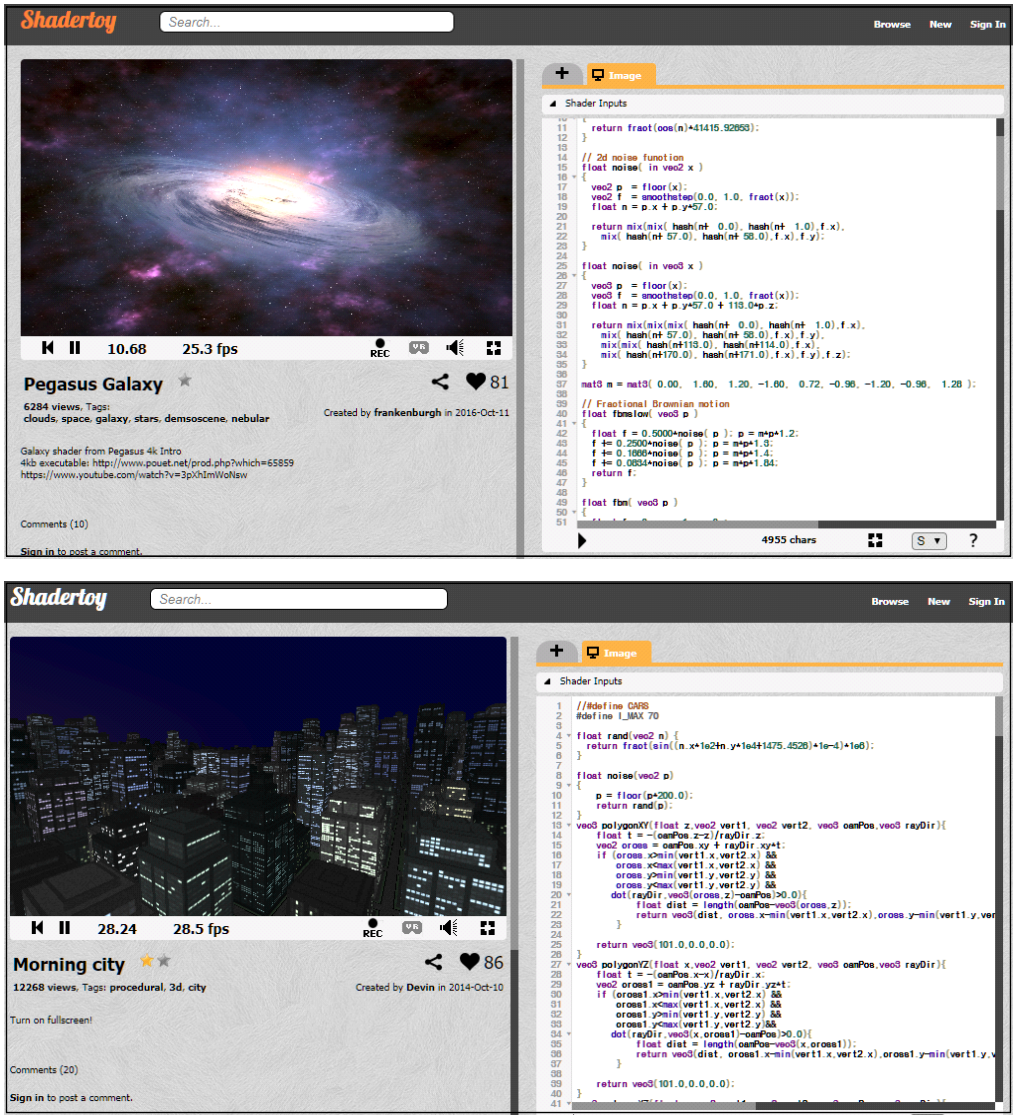
〈그림 16〉 'Voice-Controlled' 소리를 이용하는 인터렉티브 프로젝션 매핑

〈그림 16〉은 소리를 직접적으로 상호작용의 재료로 사용한 프로젝션 매핑의 사례이다. 관객이 건물 앞에 설치된 마이크에 소리를 내면 건물에 프로젝션 매핑된 영상에 나타난 캐릭터의 표정이 소리에 반응하여 실시간으로 움직인다. 소리 크기의 변화만으로도 다양한 인터랙티브 효과가 만들어지는 것을 볼 수 있다. 실시간 생성 모션그래픽은 어떠한 데이터를 어떻게 사용하는가에 따라 다양한 이미지를 만들어낼 수 있다.

2) 생성의 형상

프로세싱(Processing), VVVV 등은 생성 이미지를 구현할 때 많이 사용되는 프로그램이다. 특히 프로세싱(Processing)은 자바(Java) 언어를 단순화한 프로그래밍 환경으로 시각화 관련 개발 및 디자인 작업에 사용된다. VVVV는 VPL(visual programming language) 구조로 비교적 쉽게 프로그래밍 할 수 있게 개발되었다. 이러한 프로그램 언어들을 바탕으로 만드는 생성 모션그래픽은 프로그램 코드에 의해 나타난다. 이 프로그램 코드에 의해 이미지를 생성시키는 다른 방법으로 셰이더(Shader)를 이용하는 방법이 있다. 셰이더는 화면에 출력할 픽셀의 위차와 색상을 계산하는 함수이다.[13] 우리가 모니터 화면에서 보는 이미지는 생상을 가진 픽셀의 집합이므로 셰이더는 이것을 직접 다루는 것이다.

13) Pope Kim. 「셰이더 프로그래밍 입문」. 서울:한빛미디어. (2012). p.21.



〈그림 17〉 셰이더를 이용한 실시간 모션그래픽, Shadertoy

〈그림 17〉에서 나타난 이미지는 셰이더 코드로 만들어진 생성 모션그래픽이다. Shadertoy[14]에서는 셰이더 코드를 통해 이미지의 생성을 바로 실험할 수 있다. 위 두 가지 사례에서 나타난 이미지는 각각 은하계와 도시이다. 3D 영상을 제작하는 프로그램이나 파티클 합성 프로그램들에서 제작할 수 있을 것 같은 모션그래픽이 오직 프로그래밍 코드로만 제작이 되었다. 생성 모션그

14) https://www.shadertoy.com (2017. 9. 2.)

래픽의 사례에서 주로 기하학적, 추상적인 이미지가 많이 나타나지만 실재를 모사하는 역할로도 사용할 수 있음을 보여준다. 이것은 생성의 또 다른 가능성이 될 수 있다. 생성을 위한 적절한 알고리즘이 구현된다면 변화를 갖고 모방하는 실재를 표현할 수 있을 것이다. 생성의 재료가 알고리즘을 만나면 새로운 창작의 가능성이 열릴 수 있다.

3) 새로운 시도

2016~2017년 요코하마의 홀로그램 프로젝션 극장 DMM VR Theater를 중심으로 진행된 'VRDG+H' 공연은 VR 이미지, 음악 및 멀티미디어, 코스프레의 결합을 표방하며 4차례 공연을 진행하였는데 참여한 비주얼 아티스트 'Keijiro Takahashi'는 유니티(Unity)개발자를 겸하는 아티스트로 'VRDG+H' 공연들에서 유니티(Unity)를 이용하여 만든 실시간 모션그래픽을 선보였다.

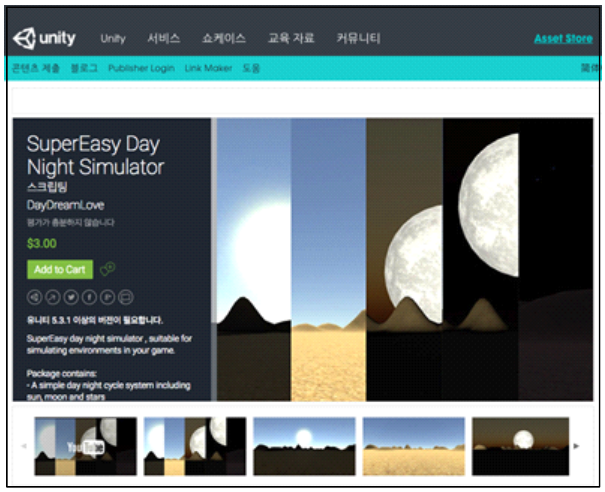
〈그림 18〉 Keijiro Takahashi의 'VRDG+H' 공연

유니티(Unity)는 게임엔진으로 비게임 콘텐츠 분야에서도 많이 사용되고 있지만, 상업적인 공연 영상 분야에서 사용하는 것은 적었다. 위 공연에서 소리에 따라 다양한 모습으로 시각화되는 공연 영상의 제작에 사용하여 유니티(Unity)의 실시간 모션그래픽 제작에 대한 가능성을 선보였다.

3.2 실시간 모션그래픽의 가능성

3.2.1 생성의 의미와 시간성

안느 마르 뒤게는 “시간은 예술작품의 반복적인 주제뿐만 아니라 예술작품의 본질을 구성하는 요소로 떠올랐다”라고 말했다.[15] 아날로그적인 이미지가 가진 내부적 시간과 외부적 시간은 물리적인 동시성을 지니고 있지만 디지털 이미지는 시간상에 있어서 내부적인 이미지나 외부적인 환경이 따로 돌아가는 이중구조를 벗어나있다.[16] 렌더링 된 영상의 시간은 아날로그적인 이미지처럼 내부적인 시간과 별개로 선형적인 외부적인 시간에 갇혀 있다. 내부적인 시간도 작가의 의도에 갇혀 있다. 그러나 실시간 생성의 가능성은 전혀 다른 시공간이다. 코드가 주는 창발성(emergence)은 작가의 의도가 반영된 이미지의 생성이 될 수도, 우연한 결과의 발현이 될 수도 있다. 외부의 시공간에서 개입시킨 데이터의 반영이 될 수도 있다.



〈그림 19〉 SuperEasy Day Night Simulator asset, 유니티 에셋 스토어

15) 이훈.「뉴미디어 아트와 시간」. 서울:재원. (2004). p.92.

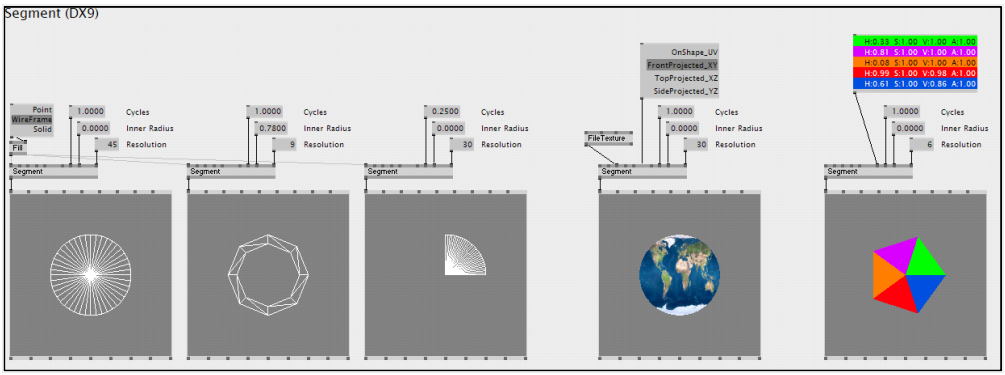
16) 이훈. 전게서. p.63.

〈그림 19〉은 게임엔진 유니티(Unity)에서 24시간 시간의 변화를 만들어주는 에셋(asset)이다. 이것을 적용한 가상공간은 낮과 밤의 시간이 생성된다. 실시간 모션그래픽은 재생 시간에 갇혀 있지 않아서 소프트웨어를 종료하는 순간이 영상의 끝이다. 이러한 점은 현실의 시간 흐름을 모방하는 그래픽을 만나 상상력을 더욱 자극한다. 게임엔진에는 하루의 시간 변화와 기후를 시뮬레이션 하는 기법이 존재한다. 콘텐츠를 실행한 순간부터 가상공간의 시공간은 현실의 세상을 모방하며 낮과 밤, 아침과 저녁을 표현하는데 기후 시뮬레이션 기능을 포함하면 더욱 현실의 시공간과 닮게 된다.

현실을 모방한 것이 시간이 바뀌는 영상의 반복 재생과 다르지 않아 보일 수 있지만 개입의 가능성이 재생과 생성의 차이를 벌린다. 현실에서 유입된 변수는 가상공간의 시간과 공간을 크리스토퍼 놀란 감독의 영화 '인셉션'처럼 자유롭게 변형한다. 무엇보다 재생시간의 무너진 경계가 보다 새로운 콘텐츠로 개발하기 위한 단초를 제시한다.

3.2.2 생성장치와 변수

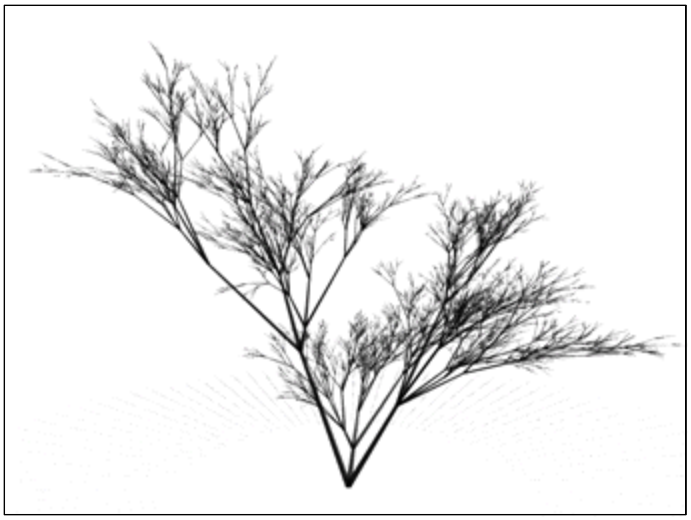
영화 '인셉션'과 같은 시공간의 변형은 변수에서 그 가능성이 시작된다. 처음 만들어진 형태는 이후 추가되는 값에 따라 달라진다. 변수는 내부적인 요인에 달라질 수 있고 외부에서 유입된 값을 받을 수 있다. 실시간 렌더링을 통해 만들어진 이미지가 갖는 디자인적 장점은 즉시 변화할 수 있는 것과 즉시 변화를 줄 수 있는 다양한 변수를 설정할 수 있다는 것이다. 생성의 시간에서 형태, 움직임의 방향과 속도, 색 등 시각화되는 모든 요소는 변수에 영향을 받을 수 있다.



〈그림 20〉 이미지의 변형을 주는 다양한 변수, VVVV

〈그림 20〉에서 다각형을 만들 수 있는 하나의 segment 노드를 통해 다양한 형태와 이미지가 만들어지는 것을 볼 수 있다. 다양한 변수를 연결하여 오브젝트의 표면을 어떻게 보일지 선택하고 변의 수, 크기, 생상, 텍스처의 적용 유무를 실시간으로 변경할 수 있다. 이 상태는 직접적인 개입이 필요하지만 필요에 따라 자동 또는 외부의 반응을 내부로 가지고 올 수 있다.

생성장치(generator)는 끊임없이 수치를 만들어낸다. 실시간 렌더러 속 가상공간에 존재하는 이미지는 이 생성장치의 값을 받아 움직임이 시작된다. 이 값은 선형적일 수도 있고 사인파(sine wave)가 될 수도 있으며 랜덤 함수로 무작위성을 갖게 할 수도 있다. 생성장치는 프로그램 내부에 존재하지만 컴퓨터와 연결할 수 있는 센서(sensor)와 같은 물리적인 인터페이스를 통해 내부의 값을 변수에 개입시킬 수 있다. 변수와 생성장치를 통해 이미지가 만들어지고 움직이며 변화할 수 있게 되는 것이다.



〈그림 21〉 VVVV로 구현한 L-system 알고리즘의 생성 이미지

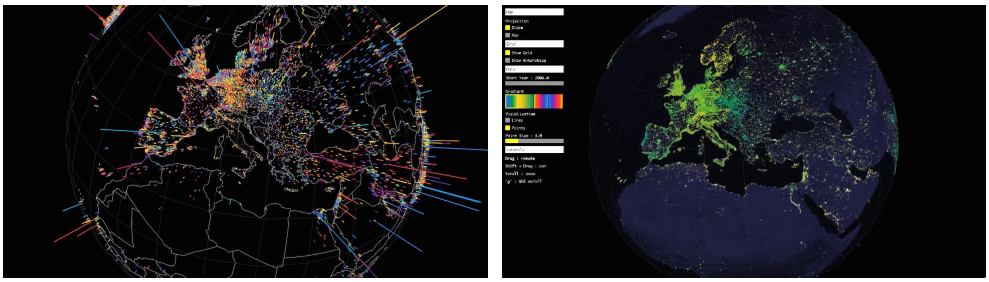
3.2.3 알고리즘과 미래

알고리즘(algorithm)은 사전적 의미로 어떤 문제를 해결하기 위한 절차나 방법을 의미한다. 어떤 아이디어를 생성하기 위한 방법에 입력(재료)을 하여 결과를 얻게 하는 방법론으로 생각할 수 있다. 예를 들어 아리스티드 린덴마이어가 1968년에 처음 소개한 L-system과[17] 같은 알고리즘의 경우, 식물의 생장을 관찰하여 자기 유사적 방법이라는 아이디어 얻고 이를 바탕으로 식물의 생장을 수학적 알고리즘으로 정리하여 이를 시각화하는데 사용할 수 있게 하였다. 오브젝트를 반복하는 원리를 가진 간단한 재귀(recursive) 함수를 바탕으로 식물의 생장을 상당히 유사한 수준으로 재현할 수 있는데 이를 바탕으로 컴퓨터 그래픽스 분야에도 사용되며 생성 이미지를 다루는 여러 분야에 많은 영감을 주었다.

간단한 알고리즘을 사용하여 생성 이미지를 그려내는 작업은 다양한 분야에서 사용되고 있다. 직물의 패턴을 디자인하는 텍스타일(textile) 디자인 분야에서 반복적인 직물의 패턴을 디자인할 때도 간단한 알고리즘을 이용하여 개

17) Casey Reas, Chandler Mcwilliams. (2014). Form+Code. 서울:길벗. p.63.

발하기도 하고 인포메이션 그래픽(information graphics) 분야에서도 많은 수의 데이터를 특정 알고리즘에 의해 정보를 시각화하는데 사용하기도 한다.[18]



〈그림 22〉 생성 기반의 정보 시각화[19]

기하학적 이미지, 패턴의 생성은 랜덤 함수를 만나 예측 불가능한 비정형의 형상으로 무한정 생성시키는 것이 가능하다. 랜덤 함수에서 나오는 난수를 의미가 있는 데이터들로 바꾸면 정보의 시각화 작업이 될 수도 있다. 앞서 봤던 식물의 생장을 모방할 수 있는 알고리즘은 생명의 현상을 모방할 수 있는 단계까지 발전하였다. 우리가 어떠한 의미를 가진 정보를 생성시키는가에 따라 생성되는 이미지는 전혀 다른 차원의 생성이 될 수 있다. 1970년 영국의 수학자 존 호턴 콘웨이가 고안해낸 세포 자동자의 일종인 라이프 게임(Game of Life)은 살아있는 이미지를 구현하였다.[20]

규칙 1) 죽은 세포의 이웃 중 정확히 세 개가 살아 있으면 세포는 생성된다.
규칙 2) 살아 있는 세포의 이웃 중에 두 개나 세 개가 살아 있으면, 세포는 계속 살아 있는 상태를 유지하고, 이외에는 소멸한다.

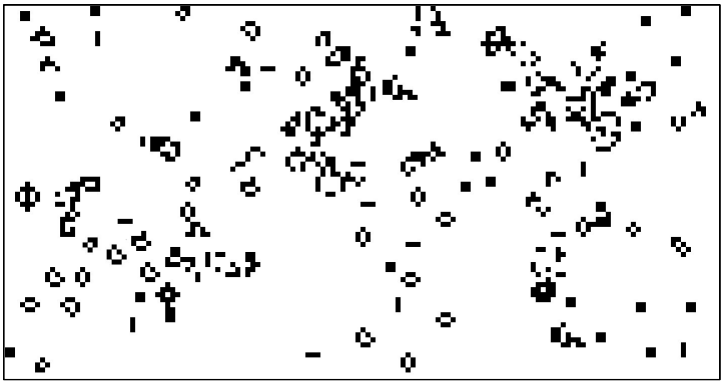
〈그림 23〉과같이 위의 간단한 규칙을 가진 알고리즘에 의해 살아있는 세포를 보는 것 같은 움직임으로 이미지가 생성되고 사라지는 것을 볼 수 있다.

---

18) Hartmut Bohnacker, Benedikt Gross, Julia Laub , Claudius Lazzeroni. (2012). Generative Design. New York:Princeton Architectura. p.67.

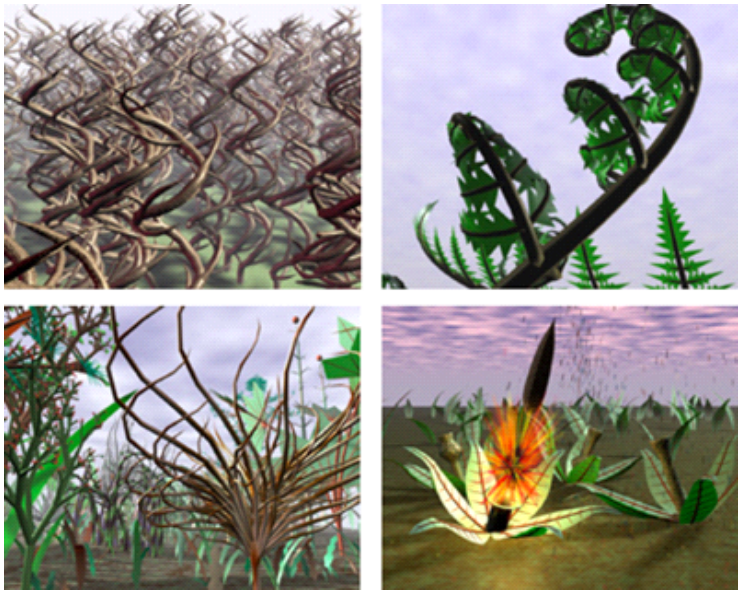
19) http://variable.io. (2017.11.10.)

20) 위키백과. 「라이프게임」. (2017. 8. 23). https://ko.wikipedia.org/wiki/라이프_게임



〈그림 23〉 라이프 게임(Game of Life) 시뮬레이션 이미지

생명 현상을 모방할 수 있는 아이디어는 다음 세대에 걸쳐 인공 생명 알고리즘으로 발전하였고 칼 심스(Karl Sims)와 같은 작가들에 의해 작품으로 나타났다. 이미지의 생성이 단순히 생성에서 그치는 것이 아닌 다양한 알고리즘을 바탕으로 현실의 사물과 생명체, 생명 현상을 모방하여 마치 살아있는 것 같은 형상과 움직임 담은 모션그래픽으로 발전시킬 수 있을 것이다.



〈그림 24〉 칼 심스(Karl Sims)의 〈판스페르미아〉[21)]

21) 인공생명 알고리즘을 적용하여 3차원 생성 이미지들을 보여주는 애니메이션

3.2.4 상호작용의 가능성

실시간 모션그래픽이 프로그래밍을 기반으로 제작되어 해당 프로그램언어가 지원하는 다양한 인터페이스의 연동 가능성을 갖는다. 이미 이전 연구에서 생성 중에 미리 만들어둔 변수에 접근하여 이미지 생성에 개입할 수 있는 가능성을 보았다. 실시간 생성 모션그래픽을 구현하는 프로그램에서 지원하는 프로토콜(protocol)을 바탕으로 각종 하드웨어를 연결하여 다양한 상호작용을 가능하게 할 수 있다. VVVV를 예를 들어 이들의 지원 현황을 정리한다.

Protocols	설명	Hardware	설명
DMX & Art-Net	조명장치 제어	Arduino	마이크로 컴퓨터
Firmata	아두이노 연동	Graphics tablets	드로잉용 타블렛
HID	USB 유형	Kinect	뎁스 카메라
HTTP	Web data	Lasers	레이저 조명
ILDA	레이저 및 조명 제어	Leap	제스추어 인식 센서
MIDI	음향 기기 연동	Lights	각종 무대 조명
MQTT	사물인터넷 연동	Motors	각종 엑추에이터
OSC	음향 기기 및 소프트에어 연동	Mobile devices	iOS, Android, MS
TUIO	멀티터치 API	Multi-touch devices	멀티터치스크린
UDP & TCP	네크워크 연동	Phidgets	센서 인터페이스
ZeroMQ	비동기 메시지 연동 라이브러리	Relay IO Boards	산업용 IO보드
		RS232	시리얼 통신
		Sensors	각종 센서
		USB/HID	USB 장치
		VR Devices	HMD
		WiiMote	닌텐도 콘트롤러
		Lemur 외	멀티터치 인터페이스

[표 6] VVVV의 지원 프로토콜 및 하드웨어 목록

[표 5]에서 볼 수 있듯 사실상 고려할 수 있는 모든 종류의 프로토콜과 하드웨어를 지원하는 것을 알 수 있다. 지원의 범위가 넓으면 더 많은 상황과 아이디어에 대응이 가능할 수 있다. 특히 외부의 상황을 측정하여 시스템 내부로 입력할 수 있는 센서들은 아두이노에 연결하여 사용할 수 있는데, Firmata protocol와 같이 아두이노 자체에 별도의 코딩을 하지 않아도 인터페이스로 사용할 수 있도록 만들어진 오픈 소스를 사용하여 외부 입력이 더욱 편리하게 받을 수 있다. 〈그림 25〉은 전시장 벽에 투사된 애니메이션이 관객의 움직임에 반응하게 만든 인터렉티브 미디어 월 작업으로 VVVV로 필요한 기능을 구현하고 적외선 센서 연결한 아두이노를 Firmata protocol로 VVVV에 연결하여 관객의 움직임을 VVVV에 입력 한 사례이다.

〈그림 24〉 이왈종 미술관, VVVV로 구동되는 인터렉티브 미디어 월

3.3 실시간 생성 모션그래픽을 위한 소프트웨어

3.3.1 프로그래밍 언어와 영상 디자인

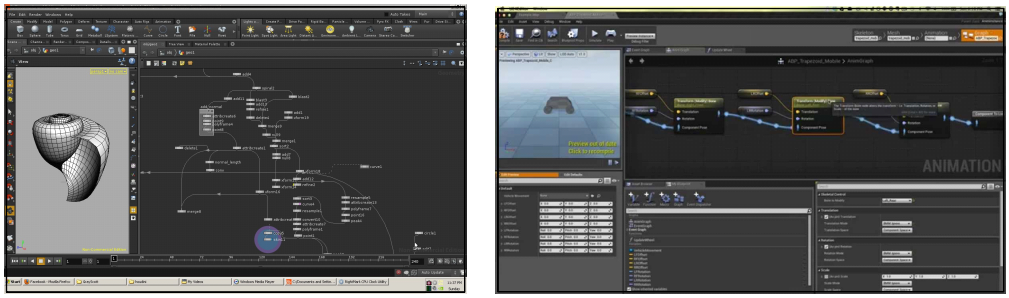
상당수의 영상 제작 관련 프로그램들은 몇 가지 프로그램 언어를 해당 프로그램 내에서 사용할 수 있도록 지원하여 영상 제작에 다양한 기능을 추가할 수 있다. 예를 들면 2D 영상의 합성, 효과 및 편집 프로그램으로 많이 사용되는 에프터 이팩트의 경우 익스프레션(expression)이라는 기능을 통해 애니메이션 효과와 각종 자동화 기능을 만들 수 있는 있는데 이 익스프레션은 Java Script 언어가 기반이 되어 같은 문법 체계를 갖고 있다.

3D 영상을 만드는 프로그램들에는 대부분 1가지 이상의 스크립트 언어를 지원하여 툴 내의 여러 요소를 세밀하게 조정할 수 있게 한다. 예를 들어 시네마4D, 마야, 맥스, 후디니. XSI 등은 공통적으로 파이썬(Python) 프로그래밍 언어를 지원하는데 이들 중 일부는 커피(Coffee) 스크립트나 멜(Mel) 스크립트 등을 중복 지원하기도 한다.

컴퓨터 그래픽스 툴에서 프로그래밍 언어가 많이 사용되는 이유는 각종 자동화의 구현도 있지만 더 정교한 파티클의 제어, 대량의 군중 시뮬레이션 등 기본 그래픽스 프로그램과 플러그인이 가진 기능의 한계를 넘는 상황에 대응하기 위해서다.

특히 후디니(Houdini)의 경우에는 렌더링 기반의 3D 영상 제작 프로그램으로 마야나 맥스와 역할의 큰 차이는 없지만 VVVV와 비슷한 노드를 사용하는 비주얼 프로그래밍 작업 환경을 가지고 있는데, 이 노드의 파라미터를 노출시켜서 모델의 외형을 바꾼다거나, 매개 변수로 사용하여 다양한 생성과 변형을 가능하게 한다. 대표적인 게임 엔진 중 하나인 언리얼 엔진(Unreal Engine)의 경우도 〈그림 25〉과같이 노드 방식의 프로그래밍 환경을 가지고 비주얼의 생성과 제어 및 각종 로직을 개발할 수 있다.

이처럼 기존의 영상 제작 프로그램들도 프로그래밍을 많이 사용하고 있는데, 이런 프로그래밍 언어들에서 이미지를 다루는 원리 역시 실시간 렌더링이 가능한 프로그램들에서 이미지를 다루는 개념과 상당 부분 맥락을 같이 한다.



〈그림 25〉 후디니(좌)와 언리얼 엔진(우)의 노드 프로그래밍

3.3.2 미디어아트, 콘텐츠 개발용 프로그래밍 언어

예술과 디자인 분야에서 적극적으로 디지털 기술과 프로그래밍이 사용되면서 다양한 미디어 작업을 수행할 있는 프로그래밍 언어들이 나왔다. 이들은 기존 프로그래밍 언어들보다 사용하기 쉽게 개량하여 만들어지거나 특정 기능들을 모아 툴킷(toolkit)의 형식을 출시되었다. 대표적인 프로그램들로는 Processing, Max/MSP, Pure Data, Quartz Composer, TouchDesigner, VVVV 등을 들 수 있고 이들은 일반적인 스크립트 언어들보다 더 간단한 문법을 사용하거나 VPL(Visual Programming Language)의 구조로 프로그래밍을 수행할 수 있다.

프로그램 언어	사용 기반	사용 분야
Processing	Linux, Mac, Windows	예술, 디자인, 뉴미디어 전반
Max/MSP	Mac, Windows	음악에 강점, 뉴미디어 전반
Pure Data	Linux, Mac, Windows	음악에 강점, 뉴미디어 전반
Quartz Composer	MAC	예술, 디자인, 뉴미디어 전반
TouchDesigner	Mac, Windows	예술, 디자인, 뉴미디어 전반
VVVV	Windows	예술, 디자인, 뉴미디어 전반

[표 7] 미디어아트 기반의 프로그래밍 언어 비교

본 논문에서 중점적으로 다루는 VVVV는 쉽게 프로토타이핑, 개발을 위한 하이브리드 그래픽, 텍스트 프로그래밍 환경을 제공하며 물리적 인터페이스,

실시간 모션 그래픽, 많은 사용자들과 동시에 상호작용할 수 있는 오디오 및 비디오로 큰 규모의 미디어 환경을 다루는데 용이하게 디자인되어 있다.[22)]

VVVV은 대표적인 VPL의 구조를 가지고 있는 프로그램으로 노드의 연결을 통해 프로그래밍을 한다. VPL은 마치 프로그램의 순서도를 짜는 듯이 프로그램의 흐름을 시각적으로 보며 프로그래밍이 가능하여 직관적이고 배우기 쉽다는 장점을 가진다. 이러한 장점으로 인해 미디어아트 관련 프로그래밍 언어에서부터 대표적인 게임엔진인 언리얼 엔진의 Blueprints까지 다양한 프로그래밍 언어들의 VPL 방식을 채용하여 사용되고 있다.

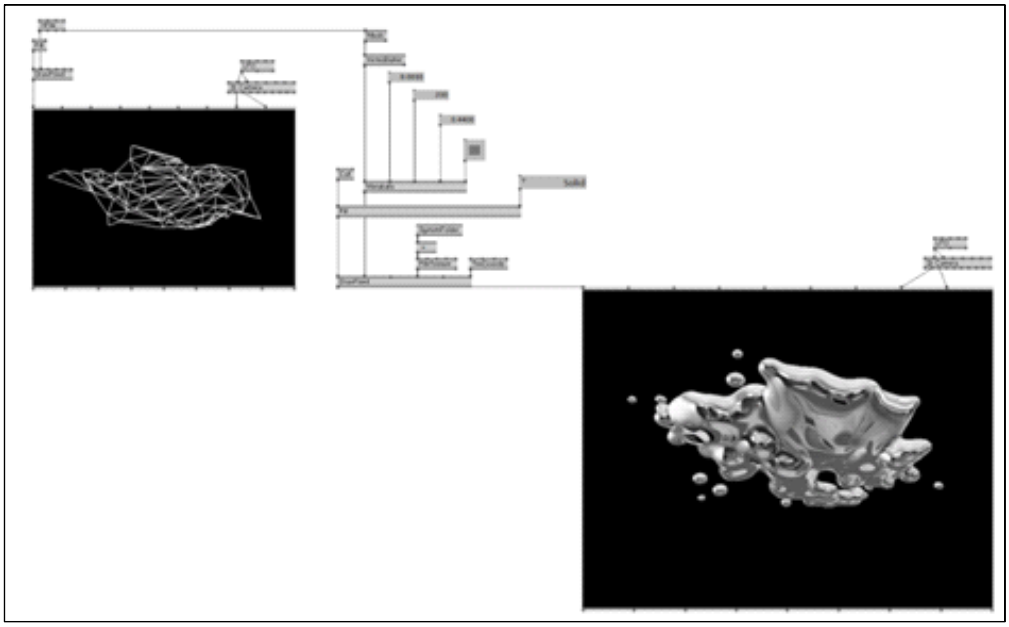
고급언어를 기준으로 프로그래밍 언어는 실행 단계에서 크게 컴파일 언어와 인터프리터 언어로 분류할 수 있다. 컴파일 언어는 대표적으로 c언어, 델파이가 있는데 이들은 동영상을 편집하고 렌더링을 하는 것처럼 작성한 프로그래밍 코드를 컴퓨터가 이해할 수 있는 기계어로 번역하는 컴파일의 단계를 거친 후 실행 파일로 생성이 된다.

인터프리터 언어는 소스 코드를 한 줄 한 줄 읽고 그때그때 번역해서 수행한다. 스크립트 언어들이 대부분 인터프리터 언어로 Python, Ruby, Lua 등이 속한다. 컴파일형 언어는 기계어로 번역을 마친 후 실행되어 구동 속도의 향상에 유리하고 인터프리터 언어는 즉시 실행 및 수정에 용이하다.

VVVV는 인터프리터 언어에 가깝지만 실행의 단계가 사실상 없는 실시간 프로그래밍의 구조를 가지고 있다. 결과를 표기하는 렌더러에 필요한 형태를 담은 노드를 연결하면 즉시 상이 나타나고 인풋과 아웃풋을 가지고 있는 여러 가지 변수들을 통해 실시간으로 변형과 효과가 반영된다. 모든 것은 실시간으로 결과가 나타나는 렌더러를 통해 프로그래밍과 동시에 결과가 나타나며 특히 3D 그래픽을 빠르게 렌더링 할 수 있게 하는 마이크로소프트의 API의 집합인 DirectX를 지원하여 높은 품질로 실시간 3D 그래픽스를 구현할 수 있다. VVVV는 실시간 렌더러를 이용한 그래픽스 환경을 통해 2D/3D 애니메이션, 생성 모션그래픽, 정보의 시각화 등이 가능하고 다양한 프로토콜 및 피지컬 컴퓨팅 기법과 연동하여 인터랙션을 구축하기 편리하다. 이러한 장점들로 VVVV가 다면 영상 및 프로젝션 매핑 분야에서도 활용되고 있다.

---

22) VVVV에 대한 공식 설명 자료. http://vvvv.org. (2017. 10. 6.)



〈그림 27〉 vvvv와 실시간 3D 그래픽스

3.3.3 비게임 콘텐츠 제작을 위한 게임엔진

컴퓨터 게임에서 그래픽을 사용한 것은 이미 1961년대였다. 키보드와 모니터를 갖춘 최초의 컴퓨터인 PDP-1에서 스페이스워!(Spacewar!) 게임이 구동되며 그래픽을 사용하는 게임의 역사가 시작되었다.[23] 이후 컴퓨터 발전과 함께 그래픽 기술의 발전이 이어지며 게임 시장은 큰 성장을 보였다.

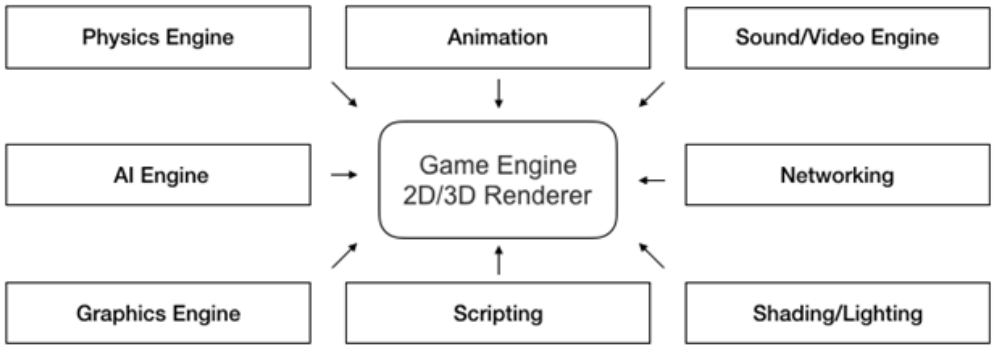
특히 1992년에 출시된 ID 소프트사의 울펜슈타인(Wolfenstein3D)은 퍼스널 컴퓨터와 MS-DOS 운영 체계 위에서 사용할 수 있는 본격적인 3D 그래픽스 기반의 게임으로 알려져 있는데 이처럼 새로운 게임들의 등장과 함께 시도된 각종 컴퓨터 그래픽 기술들이 발전을 거듭하며 기술의 고도화가 진행되었다.

게임 분야의 성장과 함께 축적된 게임 그래픽스의 기술들은 보다 편리하게 재사용하기 위해 차츰 게임 엔진이라는 이름으로 모아져 지금의 유니티(Unity), 언리얼 엔진(Unreal engine)과 같은 대중화된 게임 엔진으로 이어졌

23) 위키백과. 「PC 게임」. (2017. 10. 8). https://ko.wikipedia.org/wiki/PC_게임

다.

게임 엔진은 비디오 게임 같은 실시간 그래픽 구현의 기능을 갖추며 상호작용을 위한 응용 프로그램을 구현하는 핵심 소프트웨어의 구성을 말한다. 컴퓨터 게임 개발에 바탕이 되는 기술을 제공하여 개발 과정을 단축시켜 줄 뿐 아니라, 게임을 다양한 플랫폼에서 실행할 수 있게 해주기도 한다. 특히 게임 엔진은 재사용을 염두에 두고 있기 때문에, 하나의 게임에만 사용되지 않고 여러 종류의 게임 개발에 쓰일 수 있도록 개발되었다.[24)]



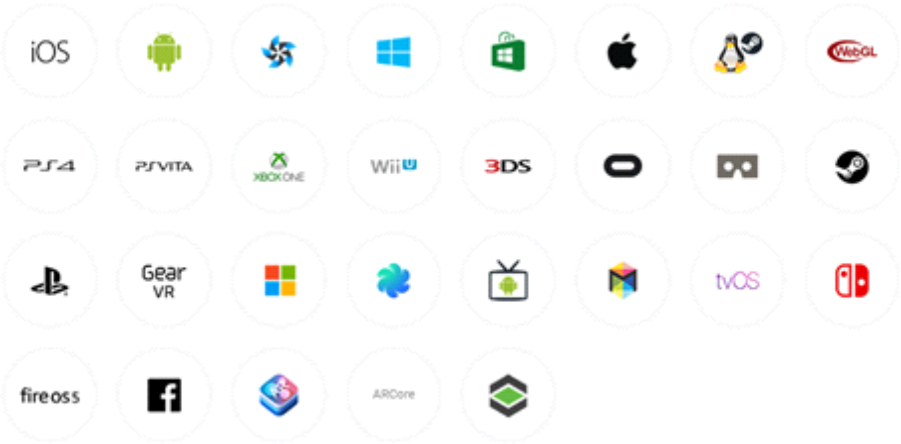
〈표 8〉 게임 엔진의 구조

2004년에 등장한 언리얼 엔진 3은 완전한 통합 게임 개발 솔루션을 표방하며 게임의 장르나 플랫폼에 관계없이 어떤 형태의 게임도 개발이 가능하도록 지원하여 본격적인 고성능 범용 게임 엔진의 시작을 알렸다. 현재 게임메이커, 럼버야드, 소스엔진, 크라이, id Tech 엔진 등 수십여 종이 출시되어 활용되고 있고 가장 많은 사용자를 보유한 게임엔진으로는 유니티(Unity), 언리얼 엔진(Unreal engine)을 들 수 있다.

유니티(Unity)는 2005년 출시 이후 게임 엔진 보급화에 가장 큰 역할을 한다. 초기에는 3D 그래픽스의 지원이 약했던 어도비 플래시에 대응하기 위한 3D 웹 미디어 제작 툴로 시작하였으나 전문적인 게임엔진에 비해 비교적 단순하고 사용법이 쉽다는 점에 착안한 개발자들이 이를 이용해 게임을 만들기

24) 위키백과. 「게임 엔진」. (2017. 10. 8). https://ko.wikipedia.org/wiki/게임_엔진

시작하면서 아예 게임 엔진으로 방향이 선회되었다.[25] 2010년 이후 PC와 Mac과 같은 컴퓨터뿐만 아닌 ios, android, xbox, PS3 등 유니티(Unity)에서 만든 콘텐츠를 다양한 기기에서도 운용할 수 있도록 멀티 플랫폼으로 발전하였고 타 게임엔진보다 저렴한 가격으로 공급하여 시장의 확대를 이루었다. 마침 스마트폰의 보급으로 모바일 게임 시장의 확대와 새로운 VR 시장의 등장으로 다양한 게임 콘텐츠에 대응할 수 있는 대중적인 멀티 플랫폼 게임엔진으로 성장하여 가장 대중적인 게임 엔진이 되었다.



〈그림 28〉 2017년 유니티의 플랫폼 지원 현황

언리얼 엔진은 에픽 게임즈에서 개발한 3D 게임 엔진으로 20여 년 가까이 게임엔진의 초기부터 활동하며 발전해온 게임 엔진이다. 유니티(Unity)보다 우수한 영상 표현 능력으로 주로 대형 게임에 많이 사용되고 있으며 특히 게임 그래픽스 분야에서 기술력을 인정받고 있다. 2015년 언리얼 엔진은 그 해 컨퍼런스에서 선보인 'A Boy and His Kite' 애니메이션을 100% 언리얼 엔진으로 구축하여 게임엔진의 영상 분야와 시뮬레이션 분야 등 비게임 콘텐츠 시장에 본격적인 활용 가능성을 알렸다.

25) 나무위키. 「게임 엔진」. (2017. 10. 8). https://namu.wiki/w/게임%20엔진


〈그림 29〉 언리얼 엔진으로 구축된 애니메이션 'A Boy and His Kite' 장면

유니티(Unity) 또한 매년 각종 컨퍼런스에서 강력해진 그래픽스 능력을 선보이기 위해 게임 엔진만으로 구축한 애니메이션 작품을 선보이며 기존에 렌더링을 기반으로 만들어졌던 3D 영상 분야에 큰 자극이 되고 있다.

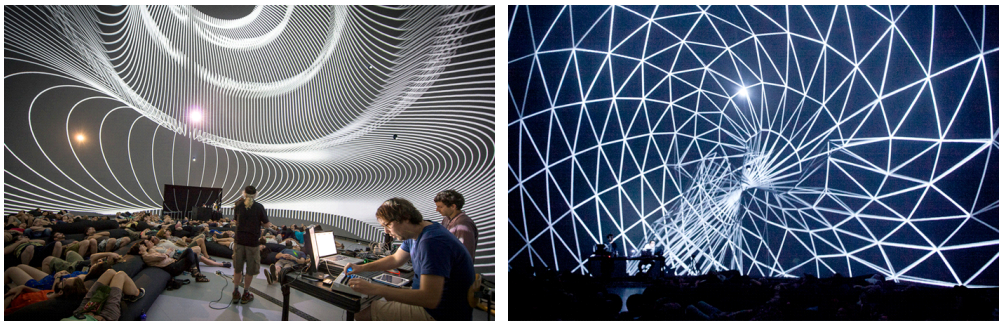
위와 같이 비게임, 콘텐츠 분야를 위한 게임 엔진의 활용은 활발하게 도입되고 있는 상황으로 애니메이션 외에도 전시 콘텐츠, 건축 시뮬레이션, AR/VR 애플리케이션 개발 등에 다양한 분야에서 새로운 멀티미디어 콘텐츠 개발을 위해 사용되고 있다.

유니티(Unity)가 비게임 콘텐츠 분야에서도 두각을 나타내는 이유 중 하나는 강력한 멀티플랫폼 지원 외에도 에셋 스토어(asset store)가 있다. 에셋은 유니티(Unity)에서 콘텐츠를 구축할 때 사용하는 이미지, 영상, 음악 등 모든 리소스를 말하는데 에셋 스토어는 개인 또는 전문 화사들이 유니티(Unity) 내에서 즉시 활용 가능한 여러 가지 에셋을 개발하여 판매 중이다. 에셋은 특정 기능을 구현하는 스크립트 코드부터 각종 디자인 요소와 UI까지 게임 및 콘텐츠 구축에 필요한 다양한 요소들이 공유되어 시간이 많이 소요되는 개발에 도움이 된다.

3.3.4 실시간 생성 모션그래픽 기반의 프로젝션 매핑 사례 분석

1) 소리에서 실시간 생성 모션그래픽으로

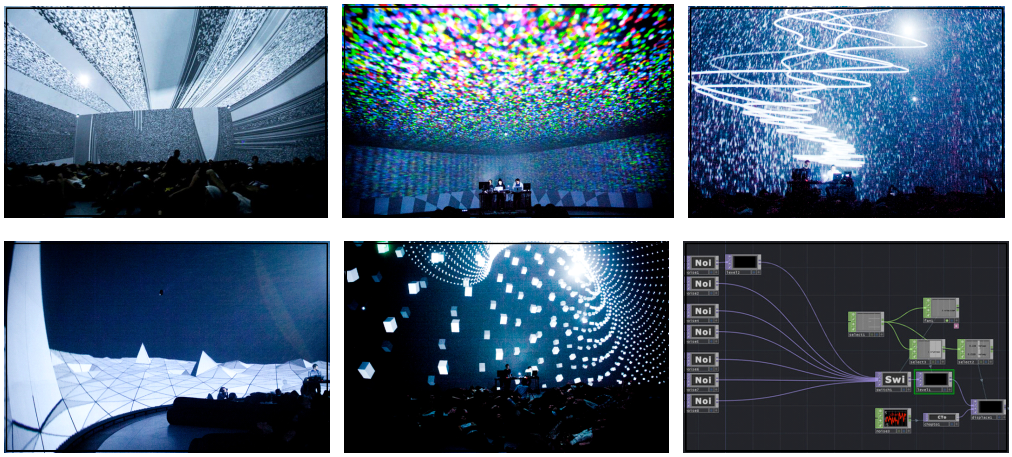
2013년 Société des Arts Technologiques에서 발표된 라이브 오디오, 비주얼 퍼포먼스 'Dromos'는 벽부터 천정이 반구형 스크린이 되는 돔(Dome) 구조의 극장 전체에 실시간 생성 모션그래픽을 프로젝션 매핑하여 만들어진 공연이다.



〈그림 30〉 라이브 오디오, 비주얼 퍼포먼스 'Dromos'

돔 구조의 스크린은 관객의 시야를 가득 덮는데 유리하다. 사람의 눈이 가지고 있는 약 200도의 양안시야 각도를 가지고 있고 그중 약 44도 정도를 또렷하게 보고 인지하는 주시야라고 한다.[26] 180도에 가까운 각도로 영상을 채울 수 있는 돔 스크린은 HMD를 이용한 VR처럼 상당한 몰입(immersive) 환경이 형성된다. 'Dromos'는 이러한 점을 활용하여 돔의 형상을 무한의 공간으로 생성하고 있다. 공간감을 극대화할 수 있는 형상과 움직임을 가진 폴리곤과 파티클, 입체 도형 등을 위주로 배치하였고 소리를 인풋으로 받아 그 형상이 다시 한번 변형된다. 영상의 형상들은 돔의 구조에 맞추어 프로젝션 매핑하였다. 실시간 생성 모션그래픽은 노드 기반의 미디어 콘텐츠 제작 프로그램인 터치 디자이너(TouchDesigner)를 사용하여 제작했다.

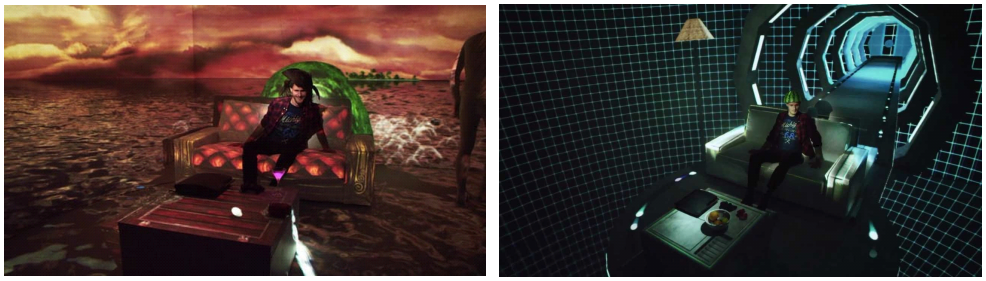
26) 이환중. 「가상현실기반 3D게임콘텐츠 개발을 위한 시야각(FOV) 연구」. 부경대학교 대학원. 석사 학위논문. (2011). p.9.



〈그림 31〉 'Dromos'의 실시간 모션그래픽과 터치 디자이너

2) 원근감의 변화

2012년 소니의 게임기 PlayStation3의 광고 영상에 프로젝션 매핑을 선보인다. 이 광고는 3편의 시리즈로 제작되어 상영되었는데 모두 실시간 프로젝션 매핑을 표방했다.

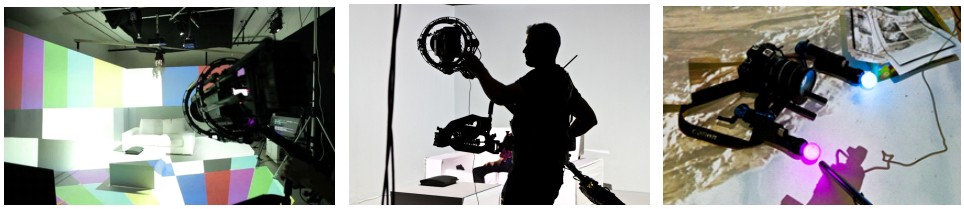


〈그림 32〉 PlayStation3의 광고

3편 모두 2개의 벽과 바닥이 있는 같은 장소에서 촬영되었는데 영상만 보면 일반적인 프로젝션 매핑과 큰 차이가 바로 보이지는 않는다. 그러나 역동적으로 움직이는 촬영 앵글을 따라 시선을 배경으로 옮기면 실시간으로 원근이 바뀌는 것을 알 수 있었다. 우리가 움직인다고 가정할 때 멀리 보이는 것은 천천히, 가까이 위치한 것은 빠르게 움직일 것이다. 렌더링 된 영상 기반으로 제작한 영상을 이용해서 프로젝션 매핑을 했다면 공간감을 형성하는 원

근은 고정되어야 한다. 그러나 해당 광고는 카메라 앵글에 따라 원근이 실시간으로 바뀐다.

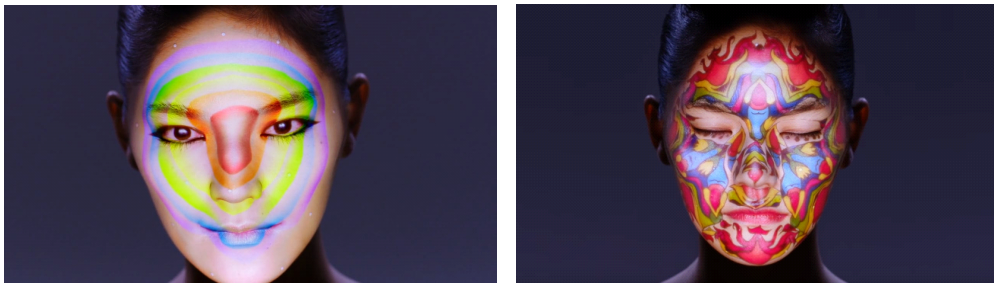
메이킹 영상에서 사용된 기술을 확인할 수 있다. 카메라의 이동으로 원근이 바뀌어 영상의 배경이 움직이려면 주 시선이 되는 카메라가 공간에 투사 중인 영상과 관계 속에 움직이도록 실시간으로 카메라를 포착(detecting) 해야 하는데, 당시 PS3의 새로운 게임 콘트롤러로 PS3 켐에 의해 움직임을 검출할 수 있는 무브(Move) 컨트롤러를 촬영하는 카메라에 장착하여 이를 해결했다. 무브(Move) 컨트롤러에 켜지는 라이트는 특정 생상을 갖는데 컬러 디텍팅을 이용해서 카메라의 움직임을 검출하고 실시간 렌더링이 가능한 게임엔진 내부의 카메라와 연동하여 원근이 바뀌는 프로젝션 매핑을 구현했다. 다수의 관객을 수용할 수 있는 방법은 아니지만 실시간 생성 모션그래픽을 사용하여 움직이는 원근감을 구현한 사례이다.



〈그림 33〉 PS3 광고, 카메라 위치 트레킹을 위해 카메라에 장착한 Move

3) 움직이는 대상

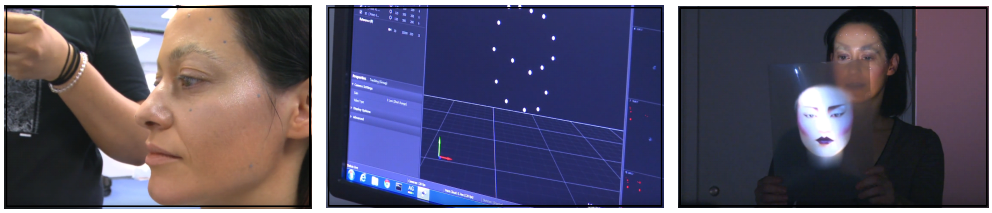
2015년 미디어 아티스트 Nawunichi Asai의 실시간 페이스 트레킹(face tracking) 프로젝션 매핑 작품 'Connected Colors'은 모델의 움직이는 얼굴에 영상을 투사하여 프로젝션 매핑을 구현한다. 대상이 움직이면 작은 차이로도 투사된 상의 비율이 달라져 왜곡이 생길 수 있는데 X, Y 축으로 단순 이동이 아닌 회전(Yaw, Pitch)이 되면 상의 비율이 달라져서 형태가 크게 바뀔 수 있다. 또한 회전하는 대상은 영상 자체가 안 보이게 되는 부분과 새로 보이게 되는 부분이 생길 수 있어 실시간 3D 방식의 프로젝션 매핑 방식이 필요하다.



〈그림 34〉 'connected colors' 실시간 페이스 트레킹, 3d 프로젝션 매핑

얼굴 움직임의 추적은 얼굴에 부착한 적외선 마커(적외선 리플렉터)를 IR 카메라를 통해 움직임을 실시간으로 검출한다. 적외선(IR, Infrared)은 사람 눈에는 보이지 않지만 가시광선 경계에 존재하는 빛으로, 가시광선을 차단하는 필터를 장착한 카메라는 적외선 빛만 촬영되어 정확한 마커의 위치를 좌표로 추출할 수 있다. 이것은 주로 모션 캡처 분야에서 사용되는 방식이다.

움직이는 좌표를 바탕으로 동기화된 얼굴 형상의 3D 모델링(본 사례는 모델링 데이터를 투사 직전에 3D 스캔을 통해 자동으로 확보한다.) 표면에 모션 그래픽을 텍스처(texture)로 입혀 사용하게 된다.



〈그림 35〉 얼굴에 붙인 IR 마커와 트레킹 프로그램

최근 들어 위와 같은 동적 대상에 프로젝션 매핑을 하는 연구들과 작품들이 속속 보고되고 있다. 이동하는 자동차, 움직이는 공연 출연자와 무대 등 움직이는 다양한 대상에 완벽한 매핑이 된다면 프로젝션 매핑이 더 큰 발전할 수 있을 것이다.

IV. 실시간 생성 모션그래픽 기반의 프로젝션 매핑 구현 제안

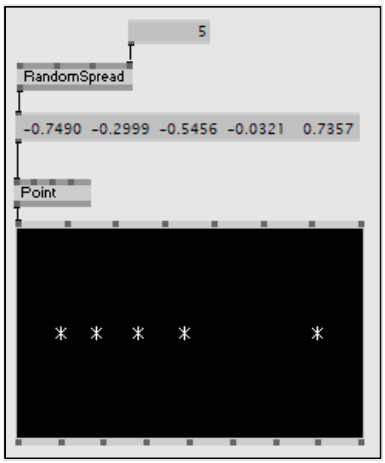
4.1 'VVVV'기반의 프로그래밍 프로젝션 매핑 기법 연구

4.1.1 'VVVV'로 만드는 실시간 모션그래픽

1) 내부 생성장치와 랜덤

VVVV에 내장된 노드(node) 중 생성의 기초 재료로 쓸 수 있는 것으로 LFO(Low Frequency Oscillator) 노드가 있다. 이것은 선형적으로 증가하는 정수와 증가되는 횟수, 증가되는 순간을 정수로 아웃풋 핀을 통해 출력한다. 값을 항상 생성함으로 이미지의 좌표 값을 가진 변수에 연결만 해도 움직임을 만들 수 있다.

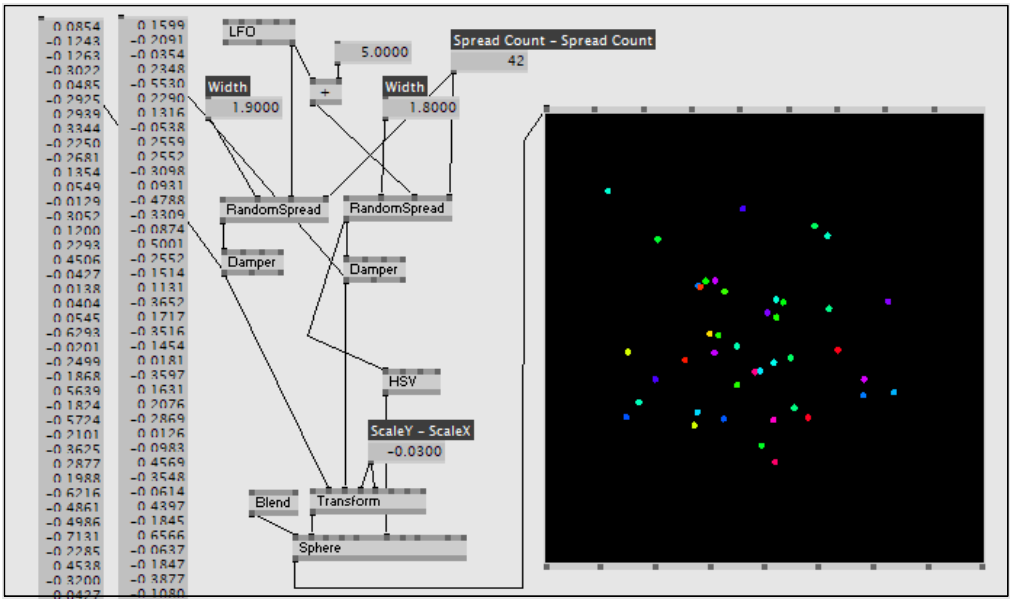
LFO와 같은 생성장치는 실시간 모션그래픽의 시작이라면 랜덤(random)은 창발(emergence)의 좋은 재료가 된다. 랜덤이 적용된 생성 실험을 위해 RandomSpread 노드를 사용한 파티클 생성 패치를 확인하였다.



〈그림 36〉 RandomSpread의 작동 구조

〈그림 37〉은 다수의 랜덤 값을 생성 시킬 수 있는 RandomSpread 노드를 사용해서 다수의 점을 각자 랜덤하게 돌아다니게 하는 패치의 모습이다.

RandomSpread 노드는 주어진 스프레드 카운트(5) 만큼 랜덤 한 실수(float)를 출력한다. 이 데이터를 X좌표에 보내면 〈그림 36〉과같이 5개의 Point가 렌더러에 랜덤하게 배치된다. 랜덤 값은 즉시 교체 가능함으로 LFO와 연동하여 아래와 같이 컬러 파티클을 생성 시킬 수 있다.

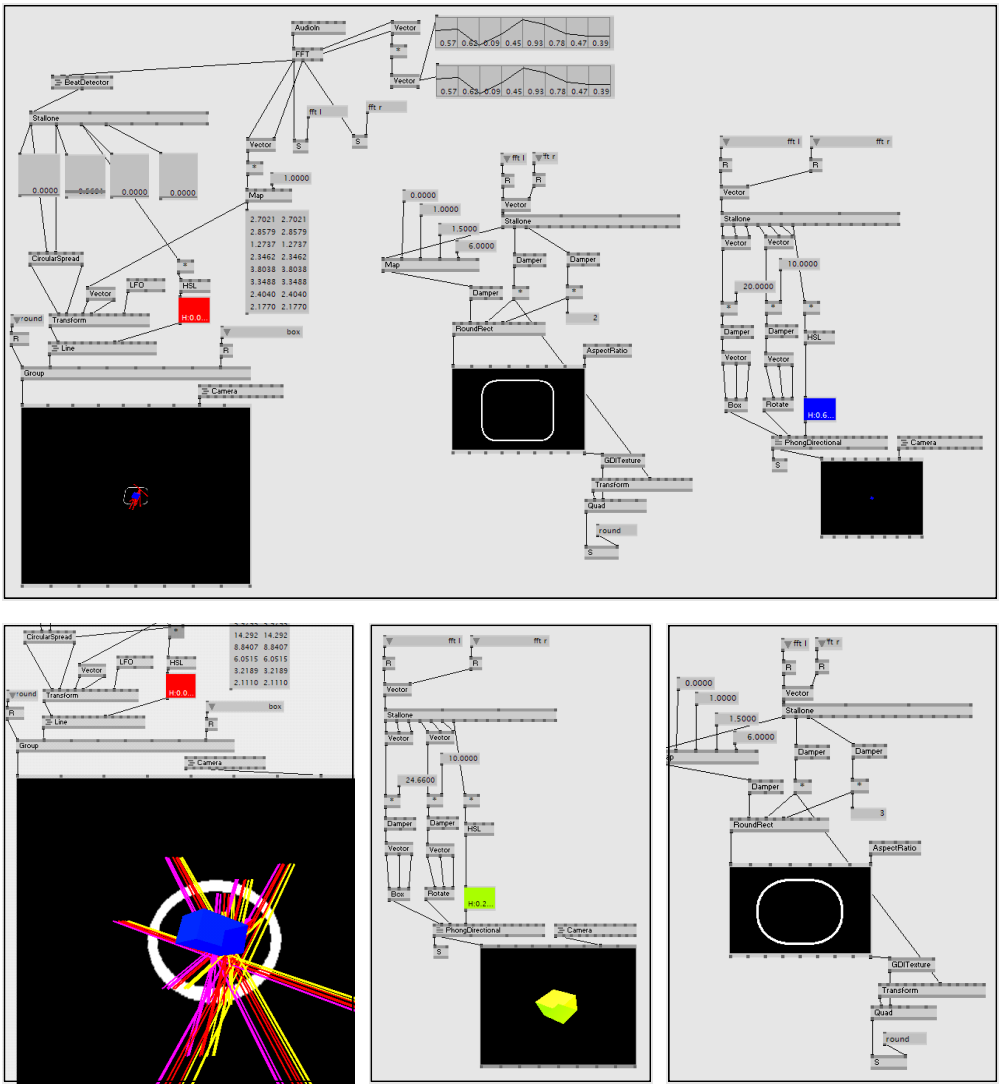


〈그림 37〉 랜덤하게 움직이는 컬러 파티클의 생성

렌더러에 보이는 작은 점들은 스케일을 줄인 Sphere이다. RandomSpread 노드를 이용하여 42개의 Sphere가 위치할 좌표값을 생성하였고 LFO를 통해 랜덤 한 좌표 값을 계속 바꿔준다. 즉시 바뀌는 좌표 값을 Damper를 사용하여 관성을 가진 증가 또는 감소 값으로 변환하면 렌더러 안을 자유롭게 떠다니는 파티클이 만들어진다. 노드의 호출과 동시에 계속 밸류(Value)를 생성시키는 LEO와 실시간 렌더러에 의해 수많은 Sphere들이 무한정 움직인다.

2) 외부 소리에서 생성된 모션그래픽

외부에서 데이터를 받는 방법은 목적에 따라 사용한 여러 인터페이스를 통해 주변의 값들을 수치로 받는 것이다. 아두이노, 센서들이 사용될 수 있고 Leap와 같은 제품화된 인터페이스를 쓸 수도 있다. 이번에는 마이크를 센서로 사용하여 이미지 생성을 실험하고자 한다. 내부 생성장치는 적게 쓰고 마이크에 들어오는 소리에서 만들어지는 수치로 이미지를 생성과 변형을 한다.



〈그림 38〉 실시간으로 소리에서 생성되는 모션그래픽

LFO는 가급적 적게 쓰고 마이크로 들어오는 소리를 수치 데이터로 변환해야 한다. 이때 사용할 수 있는 노드는 3.1.2장에서 실험한 FFT, RMS, BeatDetector 등이 있다. 각각 음역대별 소리의 크기, 전체적인 소리의 크기, 비트 값을 수치 값으로 추출할 수 있었다. LFO를 대치할 수 있는 연속된 값을 얻게 되어 이 값을 각각의 이미지 형상에 영향을 줄 수 있는 변수들에 연결하여 사용하면 된다.

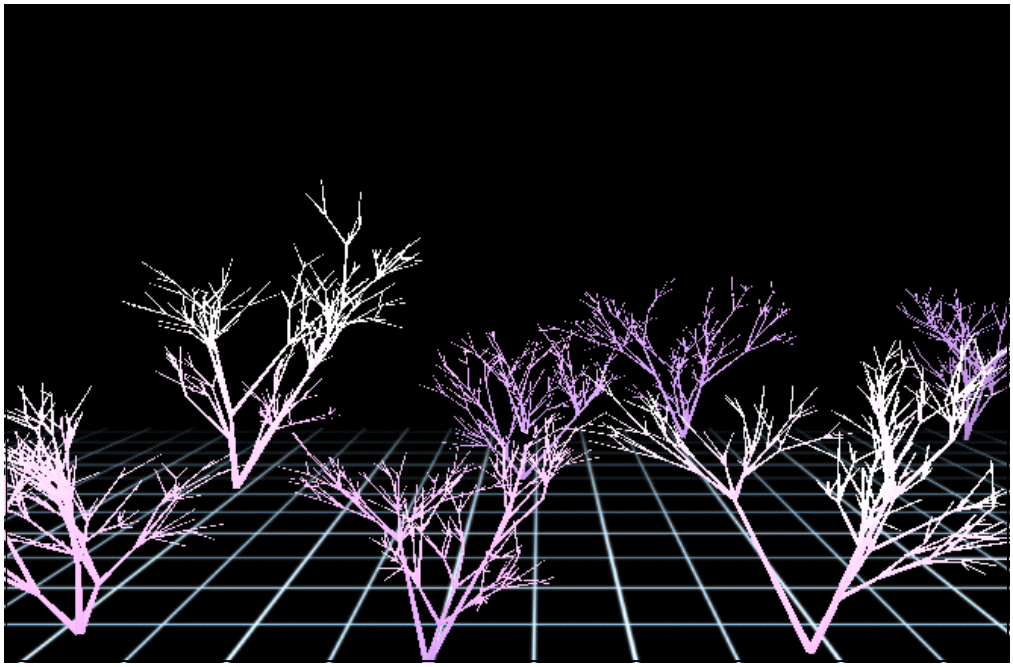
이번 실험에서는 RoundRect, Box(3D), Line을 기초 조형으로 생성 시켜 이들의 형태와 수량, 색상의 변화를 소리에서 추출한 값과 연결하여 생성과 변화를 확인하였다. 이 실험과 같이 외부에서 들어오는 데이터나 내부에서 생성시킨 데이터나 원시 데이터는 수치 값이다. 이 값을 어떻게 사용하는가가 실시간 모션그래픽의 기초가 된다.

3) L-System 알고리즘을 이용한 모션그래픽 실험

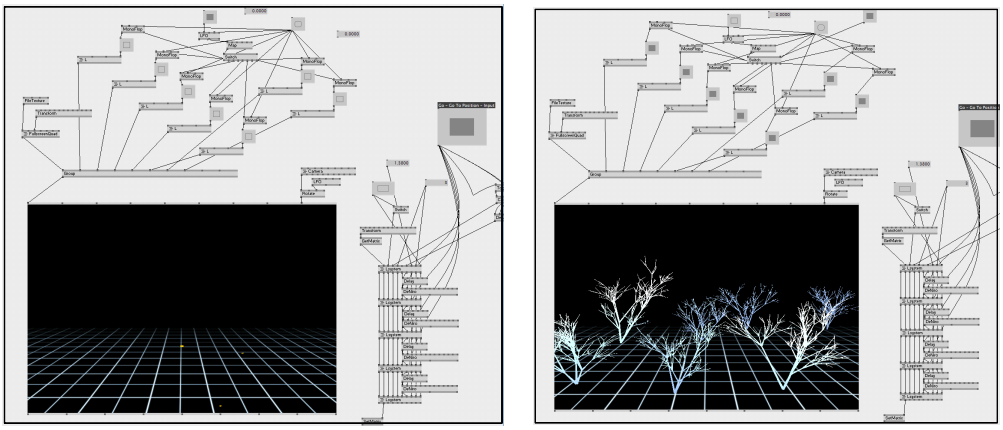
앞서 두 가지 실험에서는 생성의 기초 재료가 되는 원시 데이터를 생성하는 내부적 방법과 외부적 입력을 실험하였다. 세 번째 실험은 알고리즘의 적용에 의한 이미지 생성을 실험하고자 한다. 이를 위해 사용할 알고리즘은 린덴마이어의 식물 생장을 모방하는 알고리즘으로 L-System으로 불리며 VVVV에서 L-System 알고리즘을 쉽게 구현할 수 있는 노드가 공개되어 있어 이를 사용하여 알고리즘에 의한 실시간 모션그래픽을 실험을 진행하였다.

가상 공간에서 L-System 알고리즘이 적용된 여러 그루의 나무가 자라나게 배치하였다. 비어있던 가상공간은 입력한 신호에 의해 하나둘씩 성장하며 공간을 나무로 가득 채우게 된다.

이번 테스트로 식물이 자라나는 모습을 볼 수 있었는데 그 자체로도 모션그래픽에 응용할 수 있을 것이다. 이미지를 발생시키는 알고리즘의 연구는 텍스타일과 같은 실용적인 디자인에서 정보 디자인 분야 등 다양하게 사용 가능할 것이다.



〈그림 39〉 L-System 알고리즘에 의한 식물 생장 이미지 구현 결과



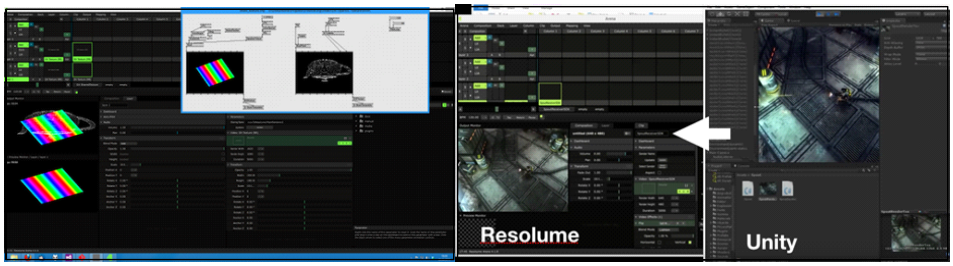
〈그림 40〉 L-System 알고리즘에 의한 식물 생장 이미지 구현 과정

4.1.2 영상 기반의 프로젝션 매핑 프로그램과 연동

VVVV와 유니티(Unity)에서 만든 실시간 모션그래픽을 프로젝션 매핑에 활용하기 위해서 사용할 수 있는 방법은 크게 두 가지로 볼 수 있다. 하나는 VVVV와 유니티(Unity)에서 실시간 생성 모션그래픽을 만들어내고 프로젝션 매핑 소프트웨어와 연동하여 투사시 발생하는 이미지 왜곡을 보정하는 것이고 다른 방법은 VVVV와 유니티(Unity) 내부에서 왜곡을 보정하는 것이다.

우선 프로젝션 매핑 소프트웨어를 이용하는 방법은 Spout와 Syphon 플러그인을 통해서 가능하다. Spout와 Syphon은 각각의 운영 체계 안에서 구동 중인 렌더링 엔진을 통해 렌더링 되는 그래픽을 다른 애플리케이션들과 공유하기 위해서 개발되었다. Window는 Spout을 사용하고 Mac은 Syphon가 사용되는데 모두 오픈소스로 개발되어 유니티(Unity)와 VVVV에서도 사용할 수 있으며 대부분의 프로젝션 매핑 소프트웨어에 내장되어 복잡한 설정 없이 사용 가능하다.

예를 들어 Spout를 활성화시키면 VVVV에서 생성시킨 모션그래픽을 프로젝션 매핑 소프트웨어인 Resolume, Madmapper 등에 실시간으로 전달하는데 이때 Resolume, Madmapper는 VVVV에서 생성시킨 영상 데이터를 자신의 소스로 인식하여 일반 동영상 소스처럼 사용할 수 있게 된다.



〈그림 41〉 Spout를 이용하여 Resolume에 렌더링 영상을 실시간 전달

VVVV는 Window 버전으로만 사용 가능하여 Spout만 사용 가능하며 유

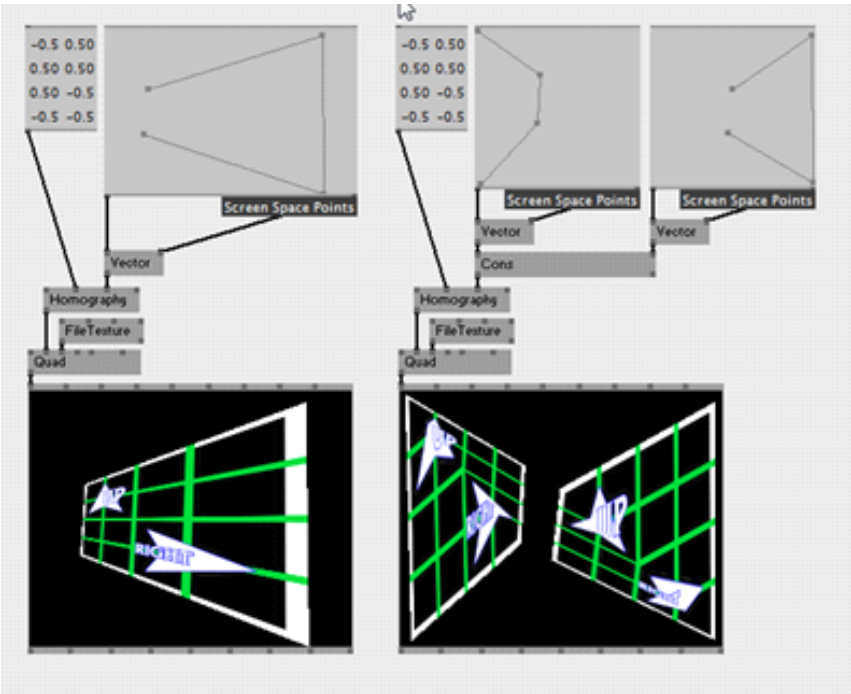
니티(Unity)의 경우 Mac와 Window 환경을 모두 지원하여 OS 환경에 따라 Spout와 Syphon 중에서 선택하여 사용해야 한다. VVVV의 경우 결과물이 보이는 렌더러 창의 아웃풋에 Spout 노드를 연결하여 보내는 구조로 보다 직관적으로 구현이 가능하고 유니티(Unity)는 카메라에 Spout 또는 Syphon 스크립트를 컴포넌트로 추가하여 사용한다.

4.1.3 'VVVV' 단독 사용의 가능성

Adobe 사의 영상 제작 관련 프로그램에는 코너핀(corner pin)이라는 이펙트가 있는데 이것을 적용한 영상은 영상 사각 면의 개별 꼭짓점을 마우스로 잡아서 화면의 비율을 원하는 대로 바꾸게 할 수 있다. 동영상 기반의 프로젝션 매핑 소프트웨어들이 면 단위로 분할된 영상의 영역을 프로젝션 대상의 형상에 맞게 배치하고 형태를 수정하는 방법이 코너핀 효과와 같은 것이다.

VVVV와 유니티(Unity)가 Spout와 Syphon을 거쳐 외부 프로젝션 매핑 소프트웨어들을 이용하지 않고 자체적으로 프로젝션 매핑을 수행하고자 한다면 우선코너핀과 같은 출력된 형상의 변형이 가능해야 한다.

VVVV는 이미지 또는 모델링 파일의 메시(Mesh) 데이터를 컨트롤할 수 있다. 기본적으로는 비례하는 스케일 조정과 좌표 이동이 가능한데 이미지를 메시 단위에서 코너핀과 같은 방식으로 형태의 변형이 필요할 경우에는 관련 기능을 수행할 수 있는 노드를 통해 프로그래밍 해야 한다. VVVV에서 Adobe 사의 코너핀 효과와 비슷한 방식으로 이미지를 변형하는 방법으로 적당한 것은 Homography 노드 사용하는 것이다. 이 노드는 이미지 또는 영상을 텍스처로 덮은 Quad의 꼭짓점을 이동시켜 코너핀과 비슷한 효과를 구현할 수 있는데 Homography에서 독립적으로 제어할 수 있는 꼭짓점의 좌표를 생성함으로 어렵지 않게 변형을 줄 수 있다.

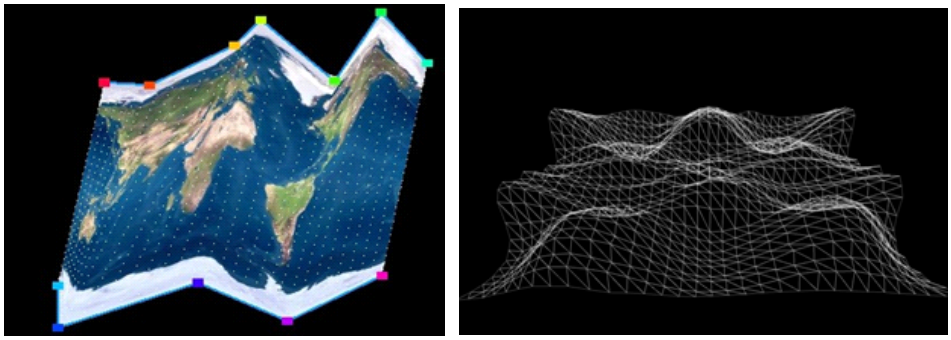


〈그림 42〉 VVVV에서 Homography 노드를 이용한 corner pin 효과

Resolume으로 자동차의 곡면과 같은 복잡한 면에 투사하게 되어 평면이 아닌 복잡한 형상에 왜곡을 만들어야 하는 경우에 Warping 모드 사용할 수 있는데, 이미지의 메시를 더 작은 단위에서 부분적으로 컨트롤할 수 있는 다수의 핀을 생성 시킨 후 영상 속 각각의 구역에 위치한 핀의 조정을 통해 매핑을 진행하게 된다.

VVVV의 경우 Quad를 더 작은 메시 단위로 분할하여 Resolume의 Warping 모드와 비슷하게 이미지의 변형을 주는 방법으로 Path2Grid와 Wave2D 노드를 활용하는 방법을 사용할 수 있었다. Wave2D의 경우 곡면 표현의 가능성을 보았고 'Path2Grid'의 경우 단일 Quad에 각 부분별로 이동이 가능한 Mesh를 형성하여 부분적인 변형이 가능할 수 있었다.

VVVV는 collada과 같은 3D 모델링 파일의 실시간 렌더링이 가능한데 이러한 3D 그래픽스의 구현 능력을 활용하면 MWM과 같은 3D 방식의 매핑을 단독으로 진행하는 것이 가능할 것이다.



〈그림 43〉 VVVV의 Path2Grid, Wave2D 노드를 이용하는 Mesh 변형

VVVV의 렌더러는 카메라 노드를 연결하면 2차원 공간에서 3차원 공간이 된다. 이 공간 안에 투사 상황을 시뮬레이션 할 수 있는 Projection 노드와 조명 효과를 위한 Light 노드를 사용하고 렌더러에 올린 투사 대상의 형상을 모델링 한 파일에 텍스처로 이미지 또는 영상을 덮어 투사하는 방식으로 3D 프로젝션 매핑의 기법을 시도할 수 있다. 복잡한 형태를 가진 대상보다 간단한 입체도형이 반복되는 상황에 유용하게 쓸 수 있으며 실시간으로 반영되는 Light 노드에 의한 빛 효과는 입체적인 대상에 그림자 효과를 주어 입체감과 공간감을 만들어 내기가 편리할 수 있다.

Homograph, Path2Grid, Wave2D 노드들 모두 전문적인 프로젝션 매핑을 위해 설계된 노드가 아니어서 정교한 작업 또는 곡면이 많은 작업에는 불편함이 존재한다. 그러나 단순한 조형 및 입체 도형과 그것의 반복되는 상황에는 나름 효율적인 작업이 가능하여 단독 사용의 가능성이 있음을 확인하였다.

4.2 게임엔진 Unity를 활용하는 프로젝션 매핑 기법 연구

4.2.1 Unity로 만드는 모션그래픽

유니티(Unity)는 VVVV와 조금 다른 실시간 렌더링 방식을 갖는다. VVVV의 경우 프로그래밍이 진행되는 동안 실시간 렌더러를 통해 항시 결과를 출력하고 있으나 유니티(Unity)의 경우 제작 후 컴파일의 과정을 갖는다.

컴파일 과정을 렌더링에 비교할 수 있겠지만 그 결과 나온 영상은 재생 시간에서 일반 영상과 다른 개념을 갖는다. 컴파일 후 출력된 실행 파일이 구동되면 유니티(Unity) 영상은 프로그램의 종료 전까지 실시간으로 렌더링을 계속한다. 렌더링 과정 속에 보이는 영상은 품질 면에서 VVVV 보다 뛰어날 수 있다. 당연히 컴파일의 과정이 없는 실시간 렌더링 기반의 VVVV가 유니티(Unity) 보다 퍼포먼스가 떨어지는 것이다. 이미 유니티(Unity)의 비주얼 퀄리티는 고품질의 단독 3D 애니메이션까지 제작이 가능할 정도가 되었으니 프로젝션 매핑에 유니티(Unity)와 같은 게임엔진의 우수한 질감이 사용할 수 있다면 한층 퀄리티 높은 작품의 제작이 가능할 것이다.

유니티(Unity)를 기준으로 우선 모션그래픽스의 요소를 만들어 낼 수 있는 방법론은 무엇이 있는가? 우선 원시 입체 도형의 생성은 가능하나 직접 직접적인 메시의 편집은 불가능하다. 많은 기능이 있지만 콘텐츠의 기능적 구현을 위한 기능의 위주이기 때문이다. 유니티(Unity)의 경우 외부 에셋 중에 유니티(Unity) 내에서 모델링을 수행할 수 있는 에셋이 출시되어있어 이를 사용하면 모델링 부분을 보완할 수 있을 것이다. 외부에서 불러온 모델링 데이터나 디자인 에셋들은 유니티(Unity) 내부에서 라이팅, 텍스처링, 애니메이션 등 모션그래픽을 위한 다양한 기능을 수행할 수 있다. 특히 외부 에셋을 포함해서 유니티(Unity) particle 시스템으로 만들 수 있는 각종 효과들은 화려한 영상 연출이 필요한 프로젝션 매핑에 유용하게 사용될 수 있을 것이다.

VRDG+H 공연에서 유니티(Unity)를 이용하여 실시간 모션그래픽을 선보이는 Keijiro Takahashi의 경우 유니티(Unity)의 기본 기능과 셰이더, C# 프

로그래밍을 통해 다양한 실시간 모션그래픽 작업을 진행하고 있다.

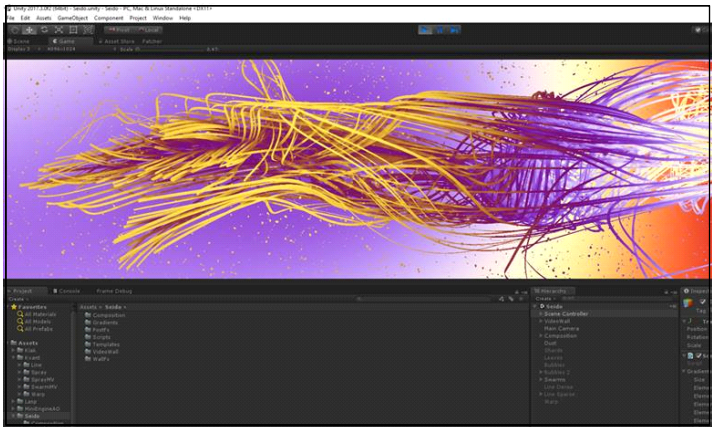
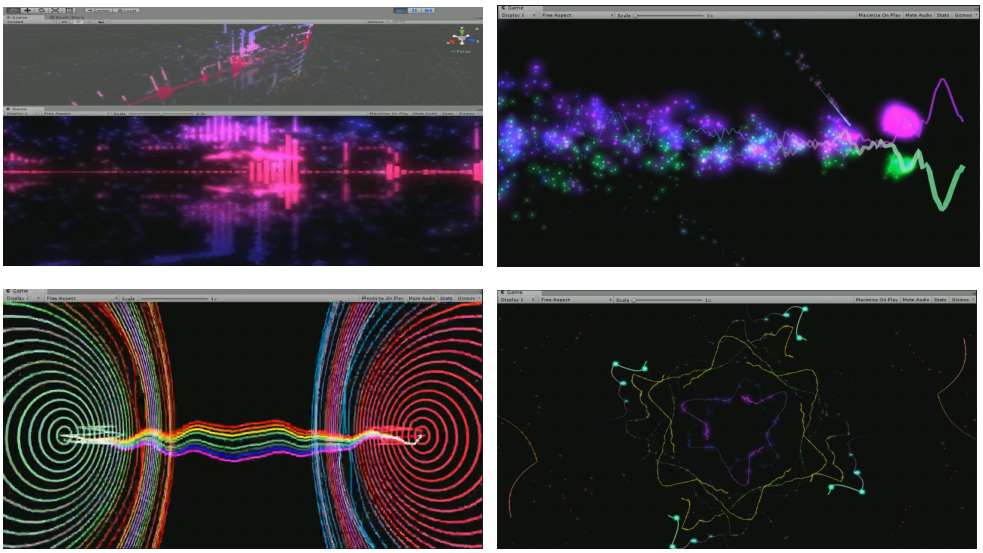


그림 44〉 Keijiro Takahashi의 유니티 실시간 모션그래픽

Keijiro Takahashi의 작업을 참고하고 Audio Visualizer 에셋을 활용하여 소리의 반응하는 실시간 생성 모션그래픽을 실험하였다. 유니티(Unity)는 컴퓨터의 내장 마이크에 접근할 수 있는 Microphone 클래스를 지원한다. 마이크 신호를 받기 위해 Microphone.devices.Length를 이용하여 소리를 유니티(Unity)로 받아오고, audioSauce.GetSpectrumData를 이용해 FFT (Fast Fourier Transform)를 이용해 소리에서 데이터를 생성시킨다. 이 데이터는 VVVV와 같이 각 이미지를 변형시키는 원시 재료로 사용된다. 노드 방식의 프로그래밍에 비해 개발 과정이 복잡하고 직관적이지 못한 부분이 있지만 유니티(Unity)의 기본 기능과 함께 사용한다면 〈그림 46〉과같이 간단한 모션그래픽을 생성 시킬 수 있다. 유니티(Unity)와 같은 게임엔진에서 기대할 수 있는 실시간 모션그래픽의 가능성은 게임 엔진의 강력한 실시간 렌더러와 인터랙션의 가능성 및 셰이더 프로그래밍에 있다.

이미 현대의 게임엔진은 기존의 3D 그래픽스 도구에 필적하는 그래픽을 보여주고 있다. 매년 각 게임엔진은 경쟁적으로 각종 발표장에서 그래픽 능력의 발전 상황을 보여주는 애니메이션 등을 통해 검증하고 있으며 이는 렌더링에 의한 비주얼이 아닌 실시간 렌더러에 의해 즉시 생성과 개입의 가능성을 가진 상태로 게임엔진의 가능성에 더욱 기대감을 갖게 한다. 개입의 가능

성은 실행 중인 애니메이션의 공간을 원하는 대로 보게 하고 등장하는 다양한 요소들과 상호작용할 수 있게 하는 것이다. 이러한 점으로 인해 대부분의 VR과 AR 콘텐츠들은 게임엔진으로 만들어지고 있다. 또한 반영과 개입이 가능성은 상호작용을 중심으로 하는 각종 인터랙션 미디어에 활용될 수 있게 한다. 하나의 제작이 게임엔진의 특성상 게임과 애니메이션, 기타 콘텐츠로 원 소스 멀티 유스로 이어질 수도 있다.



〈그림 45〉 Audio Visualizer 에셋의 소리에 반응하는 실시간 모션그래픽

최신 게임 엔진들 중에는 세이터를 비교적 간편하게 노드 기반으로 구축할 수 있도록 도와주는 기능을 포함하는 경우가 있어 좀 더 수월하게 셰이더에 접근할 수 있다. 일반적인 작업에서 셰이더 단위의 작업을 하는 경우는 적지만 생성의 관점으로 볼 때 GPU의 자원을 최대한 사용하면서 알고리즘과 코드만으로도 높은 수준의 그래픽스를 구현하기 위해서 사용할 수 있다면 셰이터 프로그래밍은 또 다른 가능성을 준다.

한편 게임 엔진에는 AI(artificial intelligence) 알고리즘을 구현할 수 있는 엔진과 머신 러닝의 반영도 가능하게 하는 에셋들의[27] 출현으로 이러한 기능

27) 유니티 머신 러닝 에이전트는 2017년 11월 기준, 베타 버전으로 github에 공개되어 있어 누구나 사용할 수 있다. (2017. 11. 15) https://github.com/Unity-Technologies/ml-agents

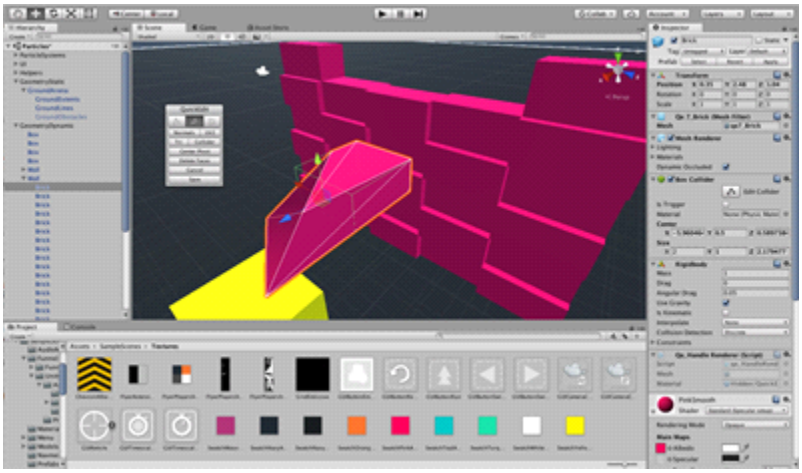
들이 적용되어 스스로 학습하여 창발하는 실시간 생성 모션그래픽도 상상할 수 있을 것이다.

4.2.3 게임엔진을 이용한 프로젝션 매핑 구현

유니티(Unity)는 4.1.2장에서 언급한 Syphon, Spout 플러그인을 이용하여 영상 기반의 프로젝션 매핑 프로그램과 연동하는 방식이 아닌 단독으로 프로젝션 매핑을 직접 수행할 수 있는 방법은 알려진 방식이 거의 없다. 유니티(Unity)가 기본 제공하는 메시 편집의 수준은 스케일 조정과 좌표의 이동 정도이므로 상당 수준의 프로그래밍 개발이 없이는 단시간에 구현이 불가능하다. 더욱이 동영상 기반의 프로젝션 매핑에서 보이는 'corner pin' 방식은 VVVV와는 달리 기본적으로 3D 작업 환경을 가진 유니티(Unity)에게 불편한 방식이 된다.

그러나 몇 가지 가능성이 존재하는데 우선 프로젝션 매핑 소프트웨어 중에서 모델링 데이터에 동영상 텍스처를 입혀 투사하는 구조로 사용할 수 있었던 MWM의 경우와 같이 프로젝션 되는 대상의 형상과 같은 모델링 데이터에 텍스처를 입히고 형태의 변화를 위한 변수를 인터페이스로 구현한다면 프로젝션 매핑의 구현이 어느 정도 선에서는 가능할 것이다. 유니티(Unity)는 직접 모델링이 가능한 솔루션이 아니기 때문에 메시 편집을 위해서는 관련 스크립트를 개발하거나 외부 에셋을 사용해야 한다.

VVVV에 비해서는 직접 구현에 많은 어려움이 있지만 가능성을 확인하기 위해 메시 편집을 수행할 수 있는 에셋을 활용하여 테스트를 진행하였으나 씬 뷰(scene view)에서 메시 편집의 가능 여부와 별개로 실행 후 게임 뷰에서 사용이 불가능하여 바로 사용은 불가능했다. 프로젝션 매핑에서 투사시 왜곡을 보정하기 위해 영상의 형태를 바로 수정할 수 있어야 한다. 따라서 유니티(Unity)를 단독으로 사용하기 위해서는 컴파일 후 인터페이스를 통해 메시의 편집이 가능하게 개발을 하거나 Syphon, Spout 플러그인으로 게임 뷰의 결과를 바로 보내야 할 것이다.



〈그림 46〉 유니티에서 ProCore 에셋을 활용한 메시 편집 테스트

최신 컴퓨터 기술이 대거 유입되며 3D 그래픽스 분야의 기술을 선도하고 있는 게임엔진이지만 아직 직접 프로젝션 매핑 기능을 소화하기 위한 방법은 부족한 상황이다. 현재 유니티(Unity)의 경우 프로젝션 할 때 발생하는 왜곡 보정을 위한 비정형 영상 처리 asset으로 Dome 구축을 위한 asset과 복수의 프로젝터를 사용할 때 발생하는 이중상을 제거하는 edge blending asset, 프로젝션 시 상의 비율을 직사각으로 잡아주는 keystone asset 등은 출시되어 있다.

4.3 작품 실험

4장에서 실시간 모션그래픽의 구현이 가능한 VVVV와 유니티(Unity)를 중심으로 프로젝션 매핑에서 구현 및 활용 가능성을 살펴보았다. 실제 상황에서 적용상의 가능성과 문제점을 확인하기 위해 단독으로 프로젝션 매핑의 진행이 가능한 VVVV를 사용하여 본 논문에서 다룬 실시간 모션그래픽과 상호작용의 기능을 가진 프로젝션 매핑을 구현하고 테스트하였다.

테스트 작품은 한글을 주제로 한글의 디자인적 요소를 다양한 이미지로 풀어 프로젝션 매핑 형식으로 광화문 세종대왕 동상의 하단부 3면에 투사하였

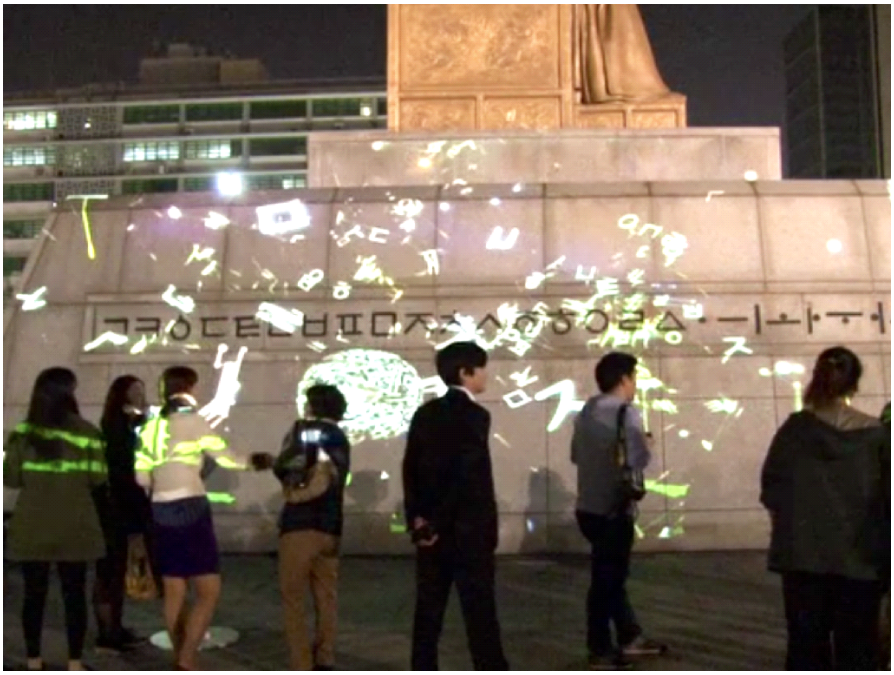
다. 2개 파트로 제작하였는데 파트 1에서는 일반적인 렌더링 기반의 동영상으로 제작하고 VVVV를 이용하여 3면 프로젝션 매핑을 진행하였고 파트 2에서는 VVVV로 실시간 모션그래픽을 구현하고 관객과 상호작용하는 기능을 추가하였다. 관객은 프로젝션 대상 앞에 설치한 한글 민들레 씨 조형물에 입으로 바람을 불면 매핑 영상 속에서 민들레 씨가 퍼져나가도록 만들었다.

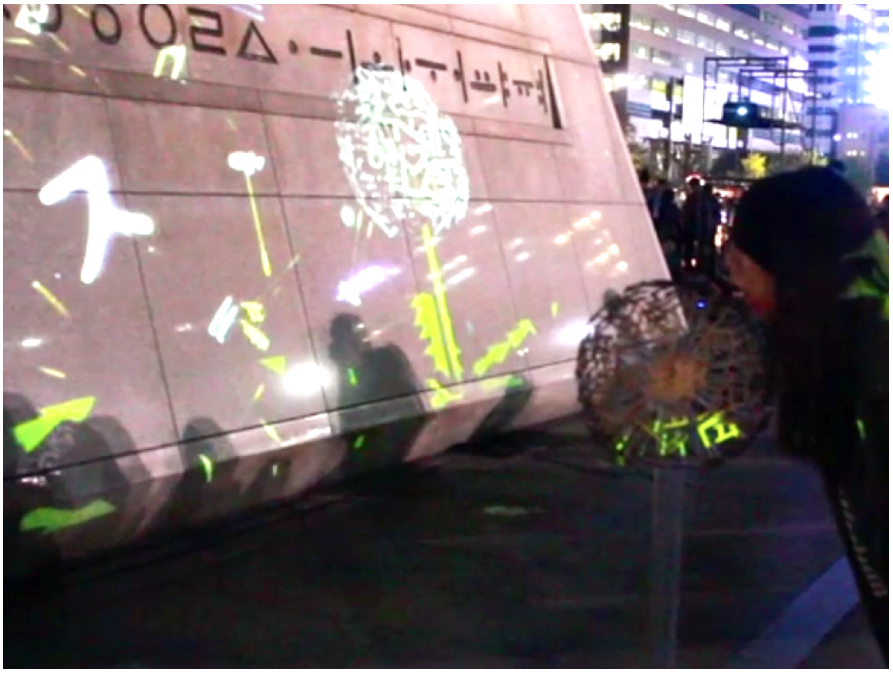
관객이 불게 되는 민들레 씨 조형에는 작은 마이크를 설치했는데 마이크를 통해 들어오는 값의 크기가 일정 값 이상이 되면 한글 형상을 가진 collada 타입의 3D 모델링 데이터가 공중으로 퍼져간다. 본 작업을 위해 VVVV에서 크게 3가지 구성으로 개발을 진행했는데 관객으로부터 input를 받아 작동 신호를 만드는 사운드 인터랙션 부분, 관객에게 입력받은 데이터에 의해 작동하는 3D 파일의 제어 부분, VVVV 단독 프로잭션 매핑을 진행하기 위한 corner pin 구현 부분이다.

작업의 결과 VVVV 단독으로 프로젝션 매핑을 진행하는 하는 것에는 큰 문제는 없었다. 3D 모델링 데이터는 CollradaFile 노드로 모델링 파일을 읽고 Mesh 노드로 그려내고 렌더러에 모인 그래픽 결과물들을 DX9Texture 노드로 받아서 Homography 노드로 corner pin 효과를 주어 매핑을 진행했다. 전반적인 작동에는 큰 문제가 없었으나 3D 모델링 데이터를 사용하여 실시간으로 렌더러에서 한차례 그려낸 결과를 corner pin 효과를 위해 DX9Texture으로 받아서 두번째 렌더러로 보내는 과정에 부하가 많아지는 듯하다. 프레임레이트가 다소 떨어지는 문제가 있었다. 사용되는 소스별로 Homography를 적용하여 DX9Texture를 사용하지 않으면 어느 정도 개선 될 것으로 보인다.

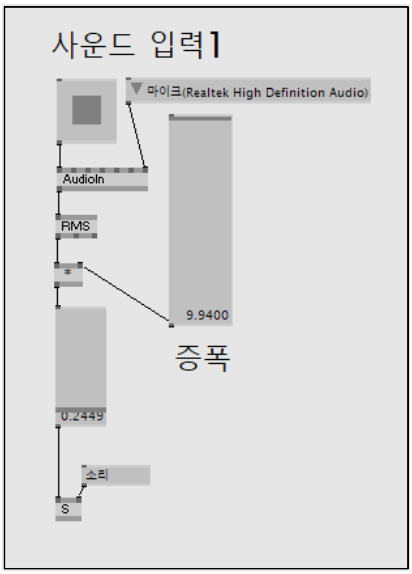
프로젝션 매핑의 대상이 비교적 단순하고 직선 위주의 상황이었으나 대상을 덮고 있는 대리석의 각각의 칸에 영상을 정밀하게 맞추는 부분에서는 한계를 보였다. 측면은 전체적으로 평평하였으나 경계면이 있어 이를 이용하는 패턴을 만들어 사용하려고 하였으나 메시 편집이 자유롭지 않아서 다소 어색한 부분이 나타났다. VVVV의 경우 복잡한 형상에 프로젝션 매핑 소프트웨어들이 가지고 있는 와핑 기법을 이용하여 프로젝션 매핑을 할 수 있는 기능에 추가 연구와 개발이 필요하다.

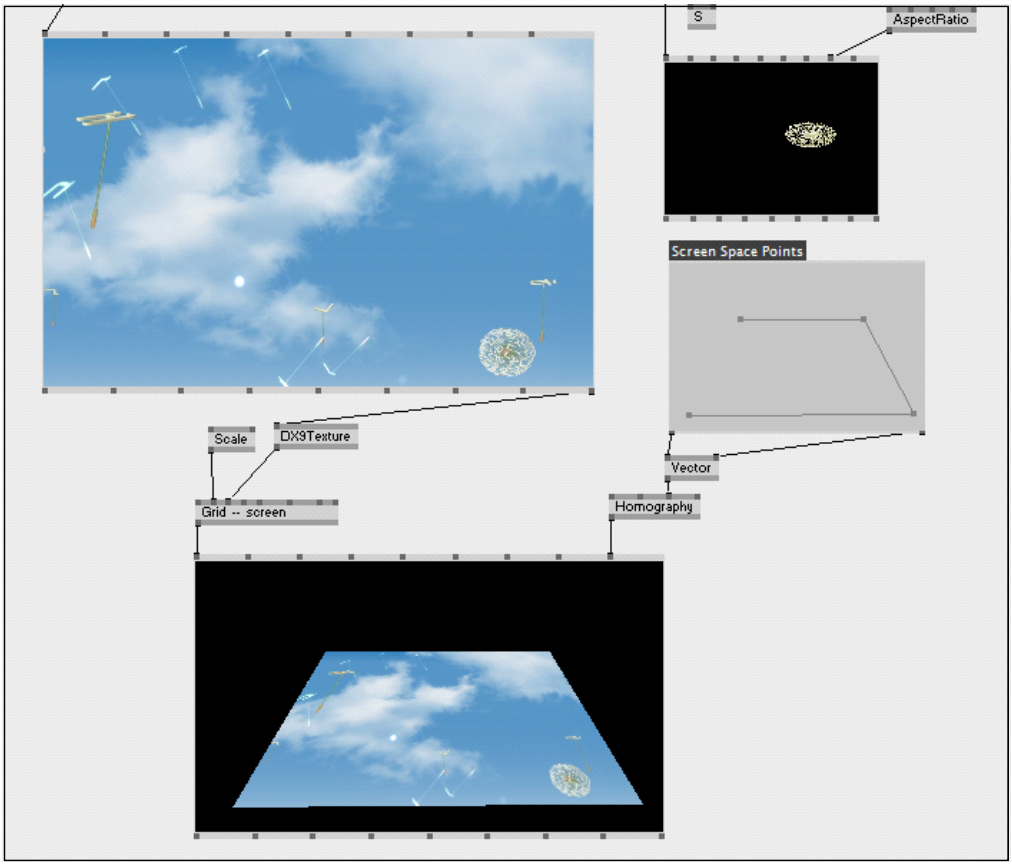
〈그림 47〉 Part1, VVVV 코너핀. 렌더링 영상으로 프로젝션 매핑



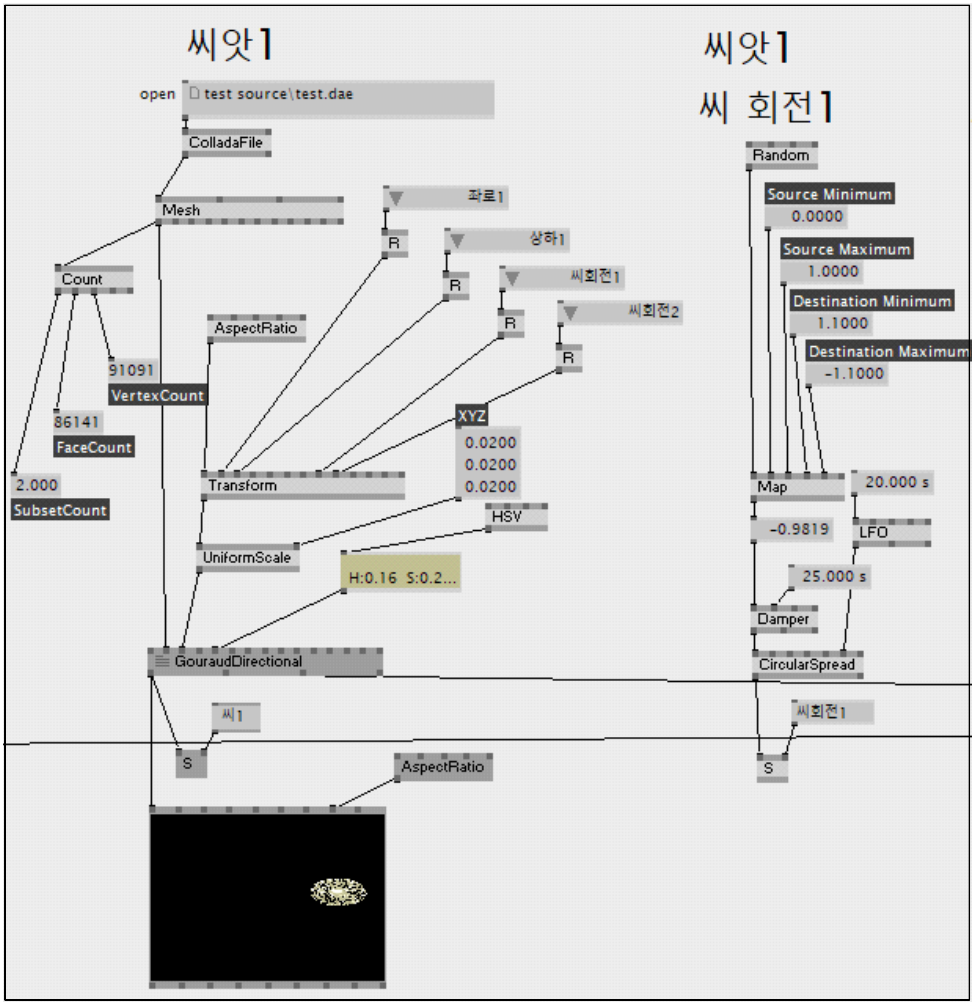


〈그림 48〉 Part2, VVVV 실시간 모션그래픽, 인터랙티브 추가





〈그림 49〉 사운드 input 처리 부분(위), corner pin 구현 부분(아래)



〈그림 50〉 실시간 생성 모션그래픽을 위해 3D 데이터 활용 부분

V. 결 론

5.1 결론 및 제언

렌더링으로 만들어진 영상을 사용하는 프로젝션 매핑의 사례들이 많아질수록 식상해질 수 있어, 차별화된 콘텐츠를 만들기 위한 새로운 제작 방법이 필요해질 것이다. 이에 본 논문에서는 기존 방법과 다른 접근 방법으로 실시간 렌더링으로 만드는 모션그래픽에 주목했다.

몇몇 사례에서 실시간 렌더링의 요소를 가진 작품들을 소개했다. 관객과 상호작용을 선보이거나, 움직이는 대상에 매핑을 하고, 카메라에 따라 원근감이 바뀌는 새로운 표현들이 들어있었다. 이들의 기술적 공통점이 렌더링의 방식으로, 렌더링 된 영상을 사용해서는 구현할 수 없는 것들이다. 관객이 낸 소리에 움직이는 얼굴 캐릭터처럼 상황과 외부의 행위에 따라 실시간으로 영상이 바뀌는데, 렌더링이 된 영상이라면 할 수 없기에 이전과 다른 방식의 제작 기법이 필요했다.

기존 프로젝션 매핑 제작 과정과 소프트웨어를 분석한 결과, 대부분은 만들어진 영상을 사용하고 비슷한 구현 방식을 사용했다. 코너핀과 와핑 등의 기법으로 대상의 입체적인 형상에 영상을 투사할 때 생기는 왜곡을 인위적으로 보정하는 방식이 비교적 표준화 되어있다.

새로운 방식으로 만들어진 사례들은 저마다의 방법과 소프트웨어를 사용하여 구축이 된다. 기존 소프트웨어에서 할 수 없는 요소가 많아 새로운 시도를 수용할 새로운 제작 방법이 필요한데 여기에 보다 효율적인 방법들을 정리할 필요성이 있었다. 새로운 시도에는 높은 비용이 생길 수 있고 안정화가 어려울 수 있기 때문이다.

비디오 아트 작품을 다수 발표한 백남준 작가는 자신의 예술을 시간을 다루는 '시간예술'이라고 하였다. 영상의 시간은 중요한 요소인데, 지금 것 영상은 선형적인 시간에 갇혀 있었다. 우리가 소비하는 영상은 제작 초기에 설정한 시간(duration)을 렌더링 하여 만들어지고 영상 내부의 시간과 현실의 시간이 평행을 이루며 선형적인 재생 시간 동안 소비되고 즉시 화석이 된다. 수

천억의 제작비로 만들어진 헐리우드 영화조차도 첨단 기술로 더 빠르고, 더 화려하고, 더 자극적으로 만들었지만 다시 볼 욕구가 빠르게 사라지는 것이다. 이것은 같은 렌더링 된 영상을 사용하는 프로젝션 매핑이 현재 가진 한계와 같은 것이다.

본 논문에서 프로젝션 매핑의 발전을 위해 필요한 영상의 변화와 인터랙티브 요소를 수용할 수 있는 근본적인 방법론의 중심에 실시간 모션그래픽을 두었다. 실시간 모션그래픽은 우선 재생시간의 경계를 지운다. 실시간 렌더링에서 가장 중요한 지점이 영상이 가진 시간 개념의 차이에서 시작한다. 실시간 렌더링은 내부의 시간을 뒤흔들어 작품 내부의 시간과 현실 시간의 평행을 흐트러뜨리고 새로운 시간의 층위를 만들어 낼 수 있다. 예를 들어 게임엔진에서 구축한 세상의 시공간은 영상의 시간에 구속되지 않고 현실의 시간처럼 진행 중이며 동시에 역행을 할 수 있고 전혀 다른 속도를 갖기도 한다. 현실을 모방한 가상의 시공간을 보면 현실과 너무도 닮아 있지만 우리가 아는 시간의 속도와는 같을 수도 다를 수도 있다.

재생시간의 경계가 사라진 곳에 새로운 생성과 개입의 가능성도 있다. 프로그래밍 코드에 의해 생성되는 이미지들은 나타남에 그치지 않고 실시간으로 변화될 수 있고 이런 과정을 알고리즘화하여 정교한 패턴을 나타나게 할 수 있다. 더욱 고도화된 알고리즘 중에는 의도를 넘은 새로운 패턴의 생성에서 생명을 모방하는 단계에 이르기까지 우리가 상상하는 것 이상을 보여줄 수 있는 가능성을 가진 것도 존재한다. 이미 게임엔진은 캐릭터의 인공지능을 구현할 수 있는 AI 엔진과 이번 연구에서 확인한 바와 같이 유니티(Unity)의 경우 머신 러닝을 도입할 수 있는 프로젝트가 진행 중이다. 이것을 적용한 근미래를 상상해보면 머신 러닝으로 그래픽의 유형을 학습한 프로그램이 예측할 수 없아 창의적인 생성 기반의 모션그래픽을 스스로 만들어낼 수도 있을 것이다. 이러한 가능성들은 화려한 볼거리만을 제공하던 기존의 사례를 넘어 새로운 시도로 연결될 수 있음을 보여준다.

이처럼 실시간 모션그래픽은 단지 기존 영상이 갖는 시간을 극복하는 것에서 그치지 않는다. 우리가 알지 못하는 새로운 의미의 모션그래픽을 만나게 될지도 모르며 결과적으로 실시간 모션그래픽이 적용된 콘텐츠는 관람객의

소비 시간을 늘려줄 가능성이 있다. 공연, 전시, 이벤트와 같은 분야에서 자극적이고 기하학적인 패턴 위주로 제작되어 일반 영상보다 더욱 빠르게 소비되어온 프로젝션 매핑의 영상에서 실시간 모션그래픽이 필요한 이유이다.

이러한 선행 연구들을 바탕으로 대표적인 실시간 렌더링 소프트웨어를 선정하고 다양한 기법들을 시도하였다. 기존의 프로젝션 매핑 소프트웨어와 함께 사용하거나 독립적으로 사용하는 방식을 기준으로 실시간 모션그래픽을 프로젝션 매핑에 적용하는 방법들을 연구하였다.

당장은 영상 처리 기능을 많이 갖추고 있는 VVVV가 편리하게 사용될 수 있었다. 코너핀과 같이 알려진 기법들이 존재하고 일부 메시 편집의 가능성이 있어 개발에 따라 복잡한 대상에 대응도 가능하다. 최신 컴퓨터 기술이 대거 유입되며 3D 그래픽스 분야의 기술을 선도하고 있는 게임엔진은 프로젝션 매핑 기능 또는 관련 에셋의 개발로 이어진다면 충분히 가능하겠지만 현재는 Syphon과 같은 플러그인을 사용하여 기존 영상 기반의 프로젝션 매핑 소프트웨어의 매핑 기능을 함께 사용하는 방법이 더 효율적일 것이다.

마지막으로 한가지 언급할 부분은 VVVV나 유니티(Unity) 모두 개발을 위한 도구라는 것이다. 프로젝션 매핑을 위한 도구가 아닌 범용 도구이다 보니 필요한 기능은 해당 프로그램으로 개발하여 사용해야 한다. 실시간 모션그래픽을 생성시키는 부분과 이것을 이용하여 프로젝션 매핑을 하는 기법은 별도의 기술이다. 4장에서 직접 만든 세종대왕 프로젝션 매핑의 사례에서도 인터랙티브를 위해 만든 패치와 코너핀 효과를 만든 패치, 실시간 모션그래픽을 구현한 패치는 각각의 모듈로 존재한다. 앞서 언급한 본 논문의 목적으로 새로운 시도를 수용할 새로운 제작 방법이 필요한데 여기에 보다 효율적인 방법들을 정리할 필요성이 있음을 밝혔다. 이점에서 향후 연구의 방향을 찾을 수 있다.

본 논문에서 시도한 방법들은 우선 사용 가능한 범위에서 실용적 가치를 가질 수 있지만, 여전히 편리하게 사용하기 위해서는 더 많은 개발이 필요하다. 이에 충분한 후속 연구들이 진행된다면 기존 프로젝션 매핑의 한계를 개선하는데 기여할 수 있을 것이다.

5.2 향후 연구

이번 연구에서 실무적으로 미디어아트와 콘텐츠 분야에서 많이 사용되는 VVVV와 범용 게임엔진으로 가장 많은 사용자를 갖고 있는 유니티(Unity)를 프로젝션 매핑에 사용할 수 있는 방법을 고민하였다.

두 프로그램 다 강력한 실시간 그래픽스의 능력을 가지고 있다. VVVV는 코너핀 구현과 메시 편집에서 바로 사용할 수 있는 가능성이 있었고 유니티(Unity)는 Dome 프로젝션 콘텐츠 구축을 위한 에셋이나 엣지 블렌딩 에셋 등 외부 에셋들을 중심으로 사용상의 이점을 확인할 수 있었다.

VVVV는 비교적 프로젝션 매핑에 적용 가능한 방법들이 일부 존재했으나 유니티(Unity)는 아직 관련 연구가 많이 부족하여 사실상 단독으로 프로젝션 매핑을 수행하기에는 어려움이 있었다. 두 프로그램은 다양한 개발을 위해 만들어진 소프트웨어인 만큼 프로젝션 매핑을 위한 기능이 완전하게 갖추어진 상태가 아니다. 향후 연구가 필요한 부분으로 VVVV의 경우 Resolume에서 지원하는 와핑과 같은 기능을 편리하게 구현할 수 있는 노드를 개발한다면 복잡하고 정교한 프로잭션 매핑을 수행하는데 도움이 될 수 있을 것이다. 유니티(Unity)의 경우에는 게임 뷰 상태나 컴파일 후에 코너핀과 와핑, 메시 편집 등 프로젝션 매핑을 위한 기능들을 사용할 수 있게 하는 방법론과 관련 개발이 필요하다. 유니티(Unity) 특성상 이러한 기능을 에셋의 형태로 개발한다면 다양하게 활용할 수 있을 것이다. 이러한 후속 연구는 실시간 모션그래픽을 편리하게 사용하는데 도움이 될 것이다.

참 고 문 헌

1. 국내문헌

김은수. (2013). 「프로젝션 맵핑이 적용된 이벤트 공간경향 분석」. 한국실내디자인 학회 2013년도 추계학술발표대회 논문집. p.144.

김민지, 박남희 (2012). 「공간매핑의 계보를 통해 본 프로젝션 매핑과 미디어 파사드」. 한국디지털디자인협의회 디지털디자인학연구 제12권 제4호. p.561.

황용희. (2011). 「프로젝션 매핑 작품에 나타난 그래픽 요소의 유형에 관한 연구」. 한국디자인포럼 Vol.32. pp.347-356.

이환중. (2011). 「가상현실기반 3D게임콘텐츠 개발을 위한 시야각(FOV) 연구」. 부경대학교 대학원. 석사 학위논문. p.9.

최양현. 조성민. (2015).「퓨처시네마」. 서울:커뮤니케이션북스.

Pope Kim. (2012). 「셰이더 프로그래밍 입문」. 서울:한빛미디어. p.21.

이훈. (2004). 「뉴미디어 아트와 시간」. 서울:재원. p.63,92.

2. 국외문헌

Hartmut Bohnacker, Benedikt Gross, Julia Laub , Claudius Lazzeroni. (2012). Generative Design. New York:Princeton Architectura. p.67.

Casey Reas, Chandler Mcwilliams. (2014). Form+Code. 서울:길벗. p.63.

3. 웹사이트

Brett Jones. The Illustrated History of Projection Mapping. (2017. 10. 1).
http://projection-mapping.org/the-history-of-projection-mapping/

1024 Architecture. Mapping a building | After Effects + Madmapper tutorial. (2017. 9. 3).
https://1024d.wordpress.com/2011/06/10/mapping-a-building-after-effects-

madmapper-tutorial/
Marcin Ignac. Projection Mapping In 3d. (2017. 10. 1).
http://marcinignac.com/blog/projection-mapping-in-3d/
wikipedia. Softwarp. (2017. 10. 5).
https://en.wikipedia.org/wiki/Softwarp
이인재. 「컴퓨터그래픽스」. (2017. 10. 10).
http://terms.naver.com/entry.nhn?docId=1624925&cid=42171&categoryId=42188
위키백과. 「라이프게임」. (2017. 8. 23).
https://ko.wikipedia.org/wiki/라이프_게임
위키백과. 「PC 게임」. (2017. 10. 8).
https://ko.wikipedia.org/wiki/PC_게임
위키백과. 「게임 엔진」. (2017. 10. 8).
https://ko.wikipedia.org/wiki/게임_엔진
나무위키. 「게임 엔진」. (2017. 10. 8).
https://namu.wiki/w/게임%20엔진

ABSTRACT

A Study on the Implementation Method of Projection Mapping using Real-time Motion Graphics

Kim, Kintae

Major in Media Design

Dept. of Hyper Media Design

The Graduate School

Hansung University

Images used for projection mapping are usually produced from live-action footages, 2D or 3D motion graphic techniques and they are created using various motion graphic effects to serve as a show in itself, as opposed to the broadcasting and film images in which storytelling is important.

If image production is finished, the image needs to be warped to accurately fit on any surface as a canvas for presentations and this process is performed using professional software for projection mapping. This software provides functions to divide screens of the image into several pieces and convert them into the form of a target object or a space.

The way of projection mapping used images which are exported by the rendering has become standard but, it still has some limits. First of all, the image is disposable since it is made for the form of display target. Secondly, it is a content for simple viewing not for experiencing, due to its linear duration like a traditional moving image on a screen. And third, the audience's position needs to be fixed when creating content to project onto a 3D surface - the content needs to be optimised for their perspective and shadow to ensure the best experience.

Many similarities in style of the content can be seen on various projection

mapping examples for a formal resemblance between projection targets. As the content is typological, it becomes a cliché. The limits mentioned above need to be improve in order to make the content for projection mapping a sustainable use media.

The goal of this research is to find a new way of projection mapping – it employs the real-time motion graphic image produced from the software with real-time renderer.

Real-time motion graphics mean the generated image from programming code and it is a subordinate concept of Computer-generated imagery and a term to emphasize the design part of an image in real-time computer graphics.

Real-time motion graphic is made regardless of duration and it makes easy to create many differences and changes in real time by substituting variable. As occasion demands, it can be easily developed to interact with the audience.

The software with real-time renderer is Processing, VVVV as programming languages and Unity and Unreal Engine as cutting-edge game engines. As the graphic quality become higher, they are utilized in various visual contents and art such as New media art, VR, game. Thus, the real-time motion graphics can exceed the limits of projection mapping and it is help to bring new ideas to projection mapping.

In this study, first of all, we analyzed the way of existing projection mapping using the image exported through the rendering process and the software with standard renderer. And then, we compared it with real-time render-based projection mapping examples and find improvement and emerging technologies.

We focused on embodiment research of projection mapping using VVVV as a real-time render based programming tool and shared the results of case studies. Furthermore, the way of projection mapping with Unity engine is arranged in this paper.

Although the examples of use of computer graphics for projection mapping is growing, the way of projection mapping is not standardised yet. For the attribute of a genre, overall, projection mapping is motion effects-oriented and the content is consumed a shorter period of time. Using real-time motion graphics for projection mapping can enlarge the spectrum of presentations by generating unintended designs and patterns. This study presents the possibilities of utilizing real-time motion graphics for projection mapping and is expected to help application method. Although it is not easy to employ game engines in projection mapping, a follow-up study can enhance the way of presentations of projection mapping technique.

[Keywords] Projection mapping, Real-time motion graphic, Rendering, Programming, VVVV