360 도 VR 영상에서 사용자 시점을 팔로잉하는 텍스트 구현

박서연*, 이소은*, 남혜영* 김형균*
*서울여자대학교 소프트웨어융합학과
*서울여자대학교 디지털미디어학과
*서울여자대학교 SW 교육혁신센터
e-mail : multikim@swu.ac.kr

Implementation of text that following to user's point of view in 360 degree VR video

Seo-Yeon Park, So-Eun Lee*, Hye-Yeong Nam* Hyeong-Gyun Kim*
*Dept of Software Convergence, Seoul Women's University
*Dept of Digital Media & Applications, Seoul Women's University
*Dept of SW Education Innovation Center, Seoul Women's University

요 약

가상현실의 발전에 따라 시 • 공간을 초월하여 영상 정보를 접할 수 있는 사회가 되었다. 본 연구는 이러한 현실에서 사용자가 영상에 대한 정보를 효과적으로 얻을 수 있도록 360 도 VR 영상에서 사용자 시점에 따라 이동하는 자막을 구현하였다.

1. 서론

가상현실(VR, Virtual Reality)의 발전으로 사용자들은 시 • 공간을 초월해 타 지역을 방문하는 것이 가능하게 되었다. 즉, 가상현실 콘텐츠의 발전이 서로 다른 공간을 이어주는 매개체가 되고 있다. 이로 인해 사용자들은 고정된 장소에서도 손쉽게 입체적인 영상 정보를 얻을 수 있게 되었다.

그러나 이미지와 음성만으로 전달하고자 하는 정보를 모두 전달할 수 있을까? 이 프로젝트는 이러한 의문에서 시작되었다. 영상에 자막이 필요한 이유는 무엇일까? 우선 영상은 정보의 전달이 빠르지만 텍스트 없이는 자세한 정보를 전달하는 데에 어려움이 있다. 다음으로 청각장애인은 영상을 시청하고 정보를 습득함에 있어서 자막이 절대적으로 필요하다. 마지막으로 네트워크의 발달로 세계 각지의 자료를 접할 수 있는 사회이기 때문에 모국어가 아닌 언어로 표현된 영상 정보들을 받아들이기 위해서는 번역 자막이 필요하다.

그렇다면 현재 사용되고 있는 VR 영상 자막의 문제점은 무엇일까? 현재 Power Director, Adobe Premiere CC, Final Cut 등에서 VR 영상을 편집하는 한 기능으로 자막 삽입이 존재한다. 하지만 현재 제작할 수 있는 VR 영상 자막은 영상의 한 곳에 고정해서 삽입하는 것만 가능하다.

즉, 이는 VR 영상을 보고있는 사용자의 시점이 자막에 있지 않으면 자막을 통해 정보를 얻을 수 없다. 따라서 본 논문에서는 기존 방식의 문제점을 해결하기 위해 VR 영상에서 사용자의 시점에 따라 움직이는 자막을 생성하는 시스템(TFUPS)을 설계하게 되었다.

그림 1. 기존에 360 도 영상의 자막 삽입 기능

2. 설계 및 구현

TFUPS 설계에 사용한 소프트웨어는 Unity 2018.3.11 이고, 운영체제는 windows 10 이다. TFUPS 를 만들 때 사용한 영상소스는 CyberLink PowerDirector 17 에서 무료로 제공하는 360 도 vr 영상 샘플 소스이다..

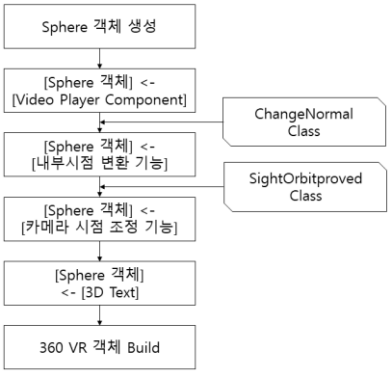

그림 2. TFUPS 생성 흐름도

그림 2 는 본 연구의 진행 순서를 나타낸다. TFUPS 를
구현하기 위한 흐름은 다음과 같다.

① Unity 의 Scene 에 Sphere 객체를 생성한다.

② [Video Player Component]를 통해 Sphere 객체 표면에 Video 를 삽입할 수 있도록 한다.

③ 그림 3 에서 그림 5 처럼 시점이 변환되도록 [내부 시점 변환 기능]을 사용한다.

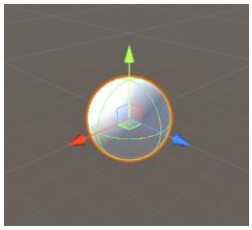

그림 3. 시점이 외부

Sphere 객체를 생성하면 그림 3 과 같이 시점이 외부인 상태가 된다. 시점을 내부로 변환하기 위해서 ChangeNormal 클래스를 사용한다.

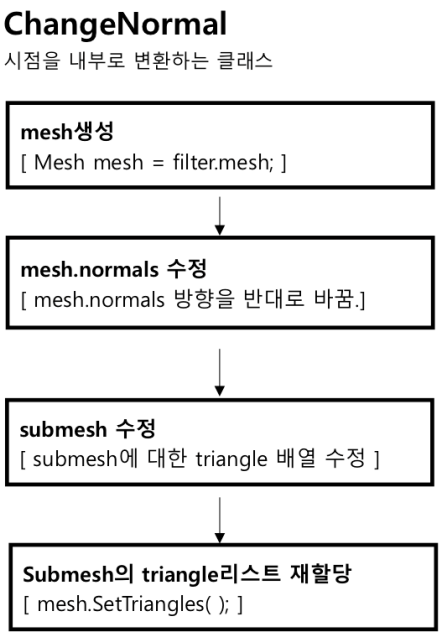

그림 4. ChangeNormal 클래스의 흐름도

ChangeNormal 은 크게 4 단계로 구성되어 있다.

[1]첫째, 정점과 여러 삼각면을 가지는 mesh 를 생성한다.MeshFilter 에 할당된 메쉬를 생성된 mesh 에 할당한다.

둘째, mesh 에서 normals 를 복사하고 변경하여 for 문을 통해 재할당한다.

셋째, 지정된 서브메쉬의 삼각형 리스트를 할당한 뒤에 수정한다.

넷째, for 문은 통해 수정된 값을 mesh 에 할당한다. 이러한 스크립트를 통해 sphere 객체를 밖에서 바라보고 있던 사용자의 시점을 sphere 안에서 밖을 바라보는 형태로 바꿀 수 있다.

ChangeNormal 클래스를 실행하면 그림 5.와 같은 형태가 된다.

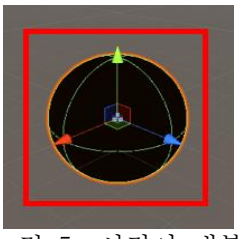

그림 5. 시점이 내부

④ [카메라 시점 조정 기능]을 이용해 그림 6 과 같이 시점에 따라 자막이 이동할 수 있는 환경을 설정한다.

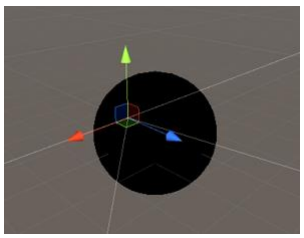

그림 6. 카메라의 시점을 이동

이 기능을 수행하기 위해서 SightOrbitproved 클래스를 사용한다.

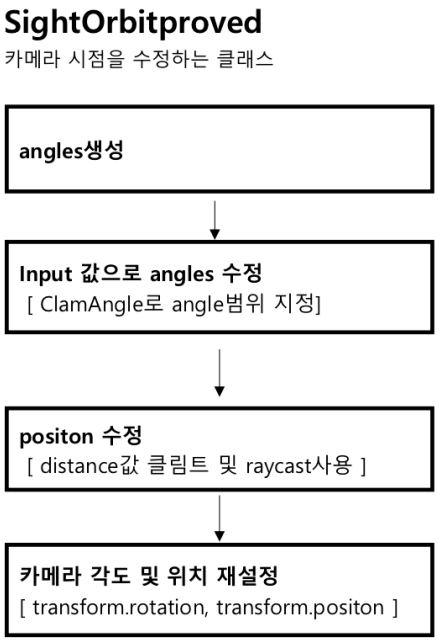

그림 7. SightOrbitproved 클래스의 흐름도

SightOrbitproved 은 크게 4 단계로 구성되어 있다.

[2]첫째, Vector3 angles 를 생성하고 transform.eulerAngles 값을 넣는다. 이를 통해 생성된 mesh 에 transform 의 Euler 각도를 할당한다.

[3]둘째, LateUpdate()함수에서 input 값으로 angles 를 수정한다. Y 축 값을 지정된 범위 내에 들어가도록 clampAngle()함수를 사용한다.

셋째, position 을 수정한다. distance 의 값이 최소/최대값 범위 내의 값만 가지도록 clamp 해준다.

이후 레이캐스트를 이용하여 target 과 transform 사이의 distance 를 구한다.

넷째, transform.rotation 과 transform.position 의 값을 지정하여 카메라의 각도 및 위치를 지정한다. 이를 통해 360 도 vr 영상을 보는 사용자의 시점에 따라 카메라의 시점이 이동할 수 있게 된다.

⑤ 3D Text 를 Main Camera 의 하위레벨에 추가한다.

그림 8. 일반 자막과 구현 자막을 비교한 모습

⑥ 360 VR 객체를 Build 한다.

그림 8 에서 보여지듯이 점선으로 표시된 자막은 사용자의 시점에 따라 움직이지 않는다. 그러나 긴 파선 점선으로 표시된 자막은 사용자의 시점에 따라 움직이는 것을 확인할 수 있다.

3. 결론

본 논문에서는 360 도 VR 영상에서 사용자 시점을 팔로잉하는 텍스트를 구현하였다. 기존의 360 도 VR 영상에 자막을 삽입하는 방식은 영상 내의 한 곳에 고정되어 나타나는 방식이다. 이는 사용자가 자막을 통해 영상에 대한 정보를 정확하게 전달 받기 어렵다는 문제점을 가진다. 이를 해결하기 위해 자막이 사용자의 시점에 따라서 움직여 사용자가 영상 내의 어느 곳을 바라보아도 지속적으로 정보를 얻을 수 있도록 설계하였다.

참고문헌

1)https://docs.unity3d.com/kr/530/ScriptReference/Mesh.html
2)https://docs.unity3d.com/kr/530/ScriptReference/Transform-eulerAngles.html
3)https://docs.unity3d.com/kr/530/ScriptReference/Mathf.Clamp.html
https://blog.naver.com/lhwhello/220723479213
https://search.naver.com/search.naver?where=nexearch&sm=tab_jum&query=%EC%9C%A0%EB%8B%88%ED%8B%B0+360%EC%B9%B4%EB%A9%94%EB%9D%BC

*** 논문 작성은 A4 용지 2 페이지를 기본으로 하며 최대 4 페이지까지 허용합니다.**
(4 페이지 초과시 페이지당 2 만원이 추가됩니다)
*** 주의 : 페이지 작업 관계로 PDF 파일로 투고가 불가 합니다.**