

360 도 VR 영상의 사용자 시점 팔로잉 자막 구현

김형균*, 박서연**, 이소은***, 남혜영***, 고혜영***

*서울여자대학교 SW 교육혁신센터

**서울여자대학교 소프트웨어융합학과

***서울여자대학교 디지털미디어학과

e-mail : multikim@swu.ac.kr

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

Hyeong-Gyun Kim*, Seo-Yeon Park**, So-Eun Lee***, Hye-Yeong Nam***, Hye-Yeong Go***

*Dept of SW Education Innovation Center, Seoul Women's University

**Dept of Software Convergence, Seoul Women's University

***Dept of Digital Media & Applications, 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 자막을 통해 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 그림 1 과 같이 영상의 한 곳에 고정해서 삽입하는 것만 가능하므로 사용자 시점의 변화에 따라 자막이 흘러가는 현상이 발생한다. 즉, VR 영상을 보고있는 사용자의 시점이 자막에 있지 않으면 자막을 통해 정보를 얻을 수 없게 된다. 따라서 본 논문에서는 기존 방식의

문제점을 해결하기 위해 360 도 VR 영상에서 사용자의 시점에 따라 움직이는 자막을 생성하는 시스템(SFUPS: Subtitle Following to User's Point of view System)을 설계하게 되었다.



그림 1. 기존 360 도 영상의 자막 형태

2. 설계 및 구현

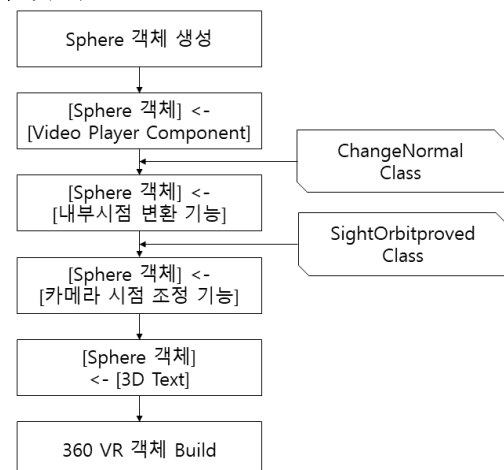


그림 2. SFUPS 흐름도

SFUPS 구현에 사용한 소프트웨어는 Unity 2018.3.11 이고, 운영체제는 windows 10 이다. 테스트를 위해 사용한 영상소스는 CyberLink PowerDirector 17 에서 무료로 제공하는 360 도 vr 영상 샘플 소스이다.

그림 2 는 SFUPS 를 구현하기 위한 흐름을 보여주고 있으며 각 수준별 기능은 다음과 같다.

가. Unity 의 Scene 에 Sphere 객체를 생성한다.

나. [Video Player Component]를 통해 Sphere 객체 표면에 360 도 영상을 삽입한다. 이때, 시점을 내부로 변환하기 위해서 Change Normal 클래스를 사용한다.

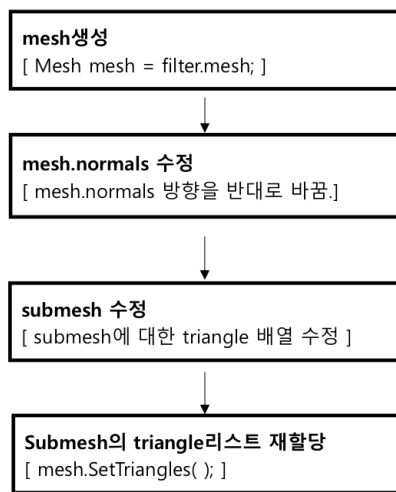


그림 3. ChangeNormal 클래스의 흐름도

ChangeNormal 클래스는 4 단계로 구성된다.

나-1. 정점과 여러 삼각면을 가지는 mesh 를 생성한다. filter(sphere 의 MeshFilter)의 mesh 속성을 생성된 mesh 에 할당한다[1].

나-2. 생성된 mesh 의 normals 를 변경하기 위해 mesh 에서 normals 를 복사하여 받아온 후 for 문을 통해 normals 를 변경하여 생성된 mesh 의 normals 에 재할당한다.

나-3. 생성된 mesh 내부의 subMesh 의 triangle 배열을 변경한다.

나-4. 변경된 triangle 배열을 생성된 mesh 내부의 subMesh 에 넣는다.

이러한 스크립트를 통해 sphere 객체를 밖에서 바라보고 있던 사용자의 시점을 sphere 안에서 밖을 바라보는 형태로 바꿀 수 있다.

다. 정의된 ChangeNormal 클래스를 통해 시점을 외부에서 내부로 변환한다. 이때, 카메라 시점을 수정하기 위해 SightOrbitproved 클래스를 사용한다.

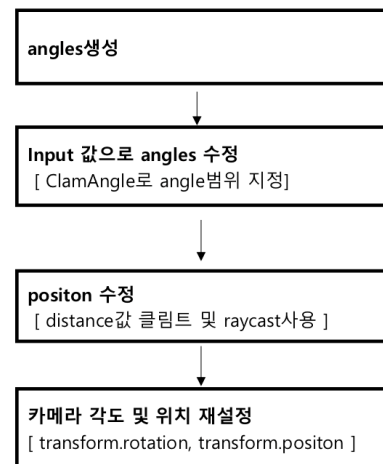


그림 4. SightOrbitproved 클래스의 흐름도

SightOrbitproved 클래스는 4 단계로 구성되어 있다.

다-1. Vector3 angles 를 생성하고 Sight Orbitproved 클래스가 부여될 객체의 Euler 각도 값을 넣는다[2].

다-2. input 을 통해 회전 값을 받아온다. Y 축 값을 지정된 범위 내에 들어가도록 clamp Angle()함수를 사용한다[3].

다-3. position 을 수정한다. distance 의 값이 최소/최대 범위 내의 값만 가지도록 한다.

다-4. SightOrbitproved 클래스가 부여될 객체의 rotation 과 position 의 값을 지정하여 카메라의 해당 SightOrbitproved 클래스를 부여한다. 이를 통해 360 도 VR 영상을 보는 사용자의 시점에 따라 카메라의 시점이 이동할 수 있게 된다.

라. 정의된 SightOrbitproved 클래스를 이용해 시점에 따라 자막이 이동할 수 있는 환경을 설정한다.

마. 사용자가 원하는 3D Text(자막)를 Main Camera 의 하위계층에 추가한다.

바. 360 VR 객체를 Build 한다.

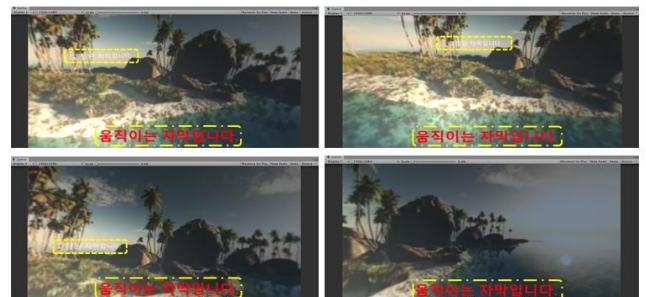


그림 5. 일반 자막과 3D Tex 자막 비교

그림 5 는 일반자막과 본 연구를 통해 구현된 3D Text 자막을 사용자 시점의 변화에 따라 비교한 것이다. 그림 내 점선으로 표시된 곳은 일반 자막을 나타내며 긴 파선 점선으로 표시된 곳은 본 연구를 통해 구현된 3D Text 자막을 나타낸다. 시점의

변화가 진행되면서 일반 자막은 사용자의 시점에 따라 움직이지 않고 흘러가고 있으나, 긴 파선 점선으로 표시된 자막은 사용자의 시점에 따라 움직이면서 사용자가 항상 자막을 볼 수 있도록 하고 있다

3. 결론

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

Acknowledgments

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW 중심대학지원사업의 연구결과로 수행되었음(2016-0-00022).

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>