

확장현실 실감 콘텐츠 개발을 위한 AMR 볼륨 데이터 변환

김종용^{1,2}

송중훈²

황규현³

윤승현⁴

박상훈^{*5}

동국대학교 영상대학원 멀티미디어학과¹

비즈아이엔에프²

한국과학기술정보연구원³

동국대학교 멀티미디어공학과⁴

서강대학교 메타버스전문대학원^{*5}

khj8499@dongguk.edu, sjh1218@vizinf.com, ghwang@kisti.re.kr, shyun@dongguk.edu, mshpark@sogang.ac.kr

Visualization of AMR Volume Data for Development of Extended Reality Realistic Content

Jongyong Kim^{1,2}

JongHoon Song²

Gyuhyun Hwang³

Seung-Hyun Yoon⁴

Sanghun Park^{*5}

Dept. of Multimedia, Dongguk University¹

VIZinf²

Korea Institute of Science and Technology Information³

Dept. of Multimedia Engineering, Dongguk University⁴

Graduate School of Metaverse, Sogang University⁵

요약

본 논문에서는 수치모델 시뮬레이션 결과로 생성된 수십 TB의 우주 천체 시간가변(time-varying) AMR(adaptive mesh refinement) 볼륨 데이터를 다양한 XR 디바이스에 사용할 수 있는 최적화된 데이터로 변환하는 과정과 방법에 대해 설명한다. AMR 볼륨 데이터는 복잡한 모델링과 시뮬레이션에 유용하게 활용되는 데이터 구조로 본 연구에서 사용되는 매우 넓은 우주를 구성하는 성단과 가스 같은 물질들을 효율적으로 표현할 수 있다. AMR 데이터의 메타데이터를 분석하여 낮은 해상도로 샘플링하고, 중요한 영역의 정보를 최적화하여 상대적으로 낮은 성능의 XR 디바이스에서도 사용할 수 있는 데이터 셋으로 변환한다. 마지막으로 데이터 셋을 활용한 실감 XR 콘텐츠 개발 사례를 통해 최적화된 데이터를 어떻게 활용하고 가시화하였는지 소개한다.

Abstract

In this paper, we describe the process and method of converting tens of TB of time-varying AMR (adaptive mesh refinement) volume data generated as a result of numerical model simulation into optimized data that can be used for various XR devices. AMR volume data is a useful data format for complex modeling and simulation, and it can efficiently express materials such as star clusters and gases that exist in the very wide outer space used in this study. we analyzes the metadata of AMR data, samples it at low resolution, optimizes information in important areas, and converts it into a data set that can be used even on relatively low performance XR devices. Finally, we introduces how the optimized data was utilized and visualized through the development of immersive XR content using the data set.

키워드: 확장현실, AMR 볼륨 데이터, 가시화, 실감 콘텐츠

Keywords: extended reality, AMR volume data, visualization, immersive contents

1. 서론

본 논문에서는 AMR [1] 볼륨 데이터를 기반으로 확장현실(extended reality)용 실감 콘텐츠를 제작하기 위한 데이터 분석과 최적화에 대해 설명한다. AMR 볼륨 데이터는 복잡한 현상을 모델

링하고 시뮬레이션하는데 사용되는 데이터 포맷으로 일반적으로 도메인을 균등한 격자(grid) 형태로 분할하는 기법과는 다르게 도메인에서 중요한 부분을 더욱 세분화하고 그 외의 부분은 덜 세분화하는 등 동적으로 해상도를 조절함으로써 효율적인 데이터로 구성이 가능하다. 이를 통해 중요 부분에 더 세밀한 해상도를

*corresponding author: Sanghun Park / Graduate School of Metaverse, Sogang University (mshpark@sogang.ac.kr)

제공하여 정확한 모델링이 가능하게 한다 [2].

콘텐츠 개발에 사용되는 AMR 볼륨 데이터는 고등과학연구원(KIAS)과 한국천문연구원(KASI)에서 세계 최대 규모의 수치 모의 실험 HR5(horizon run 5)를 통해 우주 거대 구조의 형성과 은하의 기원 연구를 위해 수행된 수치모형을 한국과학기술정보연구원(KISTI)의 슈퍼컴퓨터로 시뮬레이션한 결과이다 [3]. 해당 데이터는 실제 우주의 성단과 가스들의 움직임을 모의 실험한 데이터이기 때문에 교육적으로나 과학적으로 큰 의미가 있어 데이터를 가공하여 일반 PC나 저사양 XR 디바이스에서 렌더링하여 활용할 수 있다면 다양한 사람들에게 교육적이고 유익한 콘텐츠를 제공할 수 있을 것이다. 시뮬레이션 데이터는 온도, 크기, 밀도, 압력 등 다양한 데이터들이 시간가변 볼륨 형태로 격자에 저장된 형태를 가지며, 우주의 탄생으로부터 특정 시점까지 볼륨 데이터가 프레임별로 구성되어 있다. 원본 AMR 데이터의 경우 수십 TB의 데이터 셋으로 구성되어 있어 고성능의 워크스테이션의 경우에도 데이터를 로드하여 가시화하기 힘들다.

논문에서 사용되는 AMR 볼륨 데이터는 파티클 형태의 성단과 셀 형태의 성간물질 2가지 데이터로 구성되어 있다. 본 논문에서는 2가지 데이터 중 성단 데이터를 XR 디바이스에 활용할 수 있도록 최적화 및 경량화하는 방법과 그 과정을 설명한다. 그림 1은 원본 AMR 볼륨 데이터를 XR 디바이스에서 사용할 수 있는 데이터로 변환하는 과정을 담고 있다.

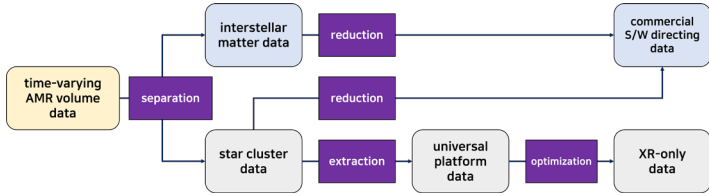


Figure 1: AMR volume data conversion process

원본 AMR 데이터는 성간물질과 성단 데이터를 모두 포함하고 있으며 매우 넓은 도메인을 가지고 있다. 우선 두 데이터를 분리하여 각각의 파일로 만들어준다. 분리된 성간물질과 성단 데이터 중 성단 데이터는 XR 전용 데이터로 변환되기 위해 추출과 최적화 작업을 거쳐 XR 디바이스에서 활용하게 된다. 그리고 분리된 두 데이터는 상용 소프트웨어에서 활용할 수 있도록 도메인이 축소되어 데이터로 저장하게 된다. 본 논문의 경우 XR 디바이스를 활용해 개발된 실감 콘텐츠의 배경 동영상을 제작할때 변환된 데이터를 사용하였다. 동영상은 제작될 콘텐츠와 XR 전용 데이터와의 연관성과 우주의 실제 현상을 해석한 데이터이기 때문에 사실성이 중요하다. 그래서 데이터의 흐름에 맞는 적절한 연출과 카메라 이동을 기획하였고 최신 렌더링 기법을 사용하여 고품질의 동영상을 제작하였다.

본 논문에서 AMR 볼륨 데이터 변환을 위해 상용 프로그램을 사용한다. 그림 1에서 데이터 분리, 추출, 축소에 Houdini [4]라는 3D 소프트웨어를 사용하였고 상용 S/W 연출 데이터를 기반으로 실감 콘텐츠에 활용될 동영상도 제작한다. 추가로 추출된 범용 플

랫폼 데이터 최적화를 통해 XR 전용 데이터 생성과 이를 활용한 실감 콘텐츠 개발에 unity3D 엔진을 사용하였다.

2. AMR 볼륨 데이터 변환

2.1 AMR 기반 볼륨 데이터 구조

본 논문에서 사용된 AMR 볼륨의 주요 데이터는 성간물질 정보와 성단의 정보를 포함하는 파티클이다. 가스 데이터는 표 1와 같이 성간물질의 density, metallicity, pressure 등 다양한 형태를 표현할 수 있는 요소로 구성되어 있으며 데이터 자체가 매우 방대하기 때문에 여러 개의 파일로 나뉘어 저장되어 있다.

variable name	data type	meaning
x, y, z	double	cell position
w	double	cell size
vx, vy, vz	float	speed
density	float	cell density
pressure	float	thermal pressure
metal	float	cell metallicity
Temperature	float	temperature

천체 시뮬레이션 중 성단 데이터인 파티클 데이터는 시뮬레이션에서 생성된 다양한 속성이 담겨 있다. 파티클 데이터의 구성 요소는 표 2와 같다.

variable name	data type	meaning
x, y, z	double	position of particle
w	float	speed of particle
mass	float	mass of particle
id	long	particle's unique id
dm	long	dark matter
star	long	a star particle
ap	float	particle formation time
zp	float	particle metallicity

파일 하나 하나의 볼륨은 고정된 공간 안의 요소를 모두 포함하지 않고 매우 불규칙적으로 데이터가 분포하고 있어 특정 파일 하나로 원하는 공간의 데이터를 얻기 어려운 형태로 구성되어 있다. 또한 AMR 볼륨 데이터 구조상 다양한 격자 크기를 가지고 있어 이를 정확히 변환하지 않으면 격자 내부의 값이 어느 공간을 대표하고 있는지 명확히 알기가 어렵다. 이러한 문제를 해결하기 위해 필요한 데이터 격자 크기를 결정하고 XR 디바이스를 위한 데이터 변환이 필요하다.

2.2 AMR 데이터 분리

시간가변 AMR 볼륨 데이터는 그 특성상 하나의 파일에 요소들이 몰려 있지 않고 아주 넓은 영역에 분산되어 있다. 그래서 원하는 격자가 어디에 있는지 알 수가 없다. 또한 변환하려는 공간이 매우 넓고 해상도 또한 높아 하나의 파일로 변환했을 경우 파일의 크기가 매우 커 일반 PC 메모리에는 로드할 수 없다. 이러한 문제를 극복하기 위해 변환하고자 하는 공간을 일정 크기로 자른 다음 해당 공간의 데이터를 요소별로 분류하는 작업이 선행되어야 한다.

2.2.1 성간물질 데이터 분리

AMR 볼륨 데이터를 일반적인 사용 그래픽 도구에서 읽을 수 있도록 데이터 변환을 하기 위해 상용 프로그램인 Houdini를 사용하였다. Houdini에 포함된 OpenVDB [5] 라이브러리를 사용하여 AMR 볼륨 데이터의 격자 구조를 변환하여 가공하기 쉽도록 하였다. 원하는 격자 공간 데이터를 추출하기 위해 데이터 변환과 렌더링을 수행하는 Houdini 플러그인(plug-in)을 개발하였다.

플러그인을 통해 데이터를 변환하는 과정은 다음과 같다. 모든 AMR 파일의 메타 데이터와 바운딩 박스(bounding box)를 획득한 다음 각각의 AMR 파일의 바운딩 박스를 구하여 해당 AMR 데이터가 포함된 격자 공간을 획득하였다. 이후에 데이터가 있는 공간을 렌더링이 가능한 크기로 분할하고 분할된 공간을 찾아가면서 해당 영역에 속한 AMR 파일들만 열어 각 격자의 정보를 획득한다. 이렇게 획득된 정보는 다양한 AMR 데이터의 격자 크기를 지니고 있기 때문에 해당 정보가 변환될 균일한 격자 중 어디에 속하는지 판단하여 분류한다. 이후 분류된 격자내의 데이터를 추출하고 해당 데이터를 VDB 파일 형식으로 디스크에 저장한다. 그림 2는 변환된 VDB 데이터를 Houdini에서 시각화한 이미지이다.

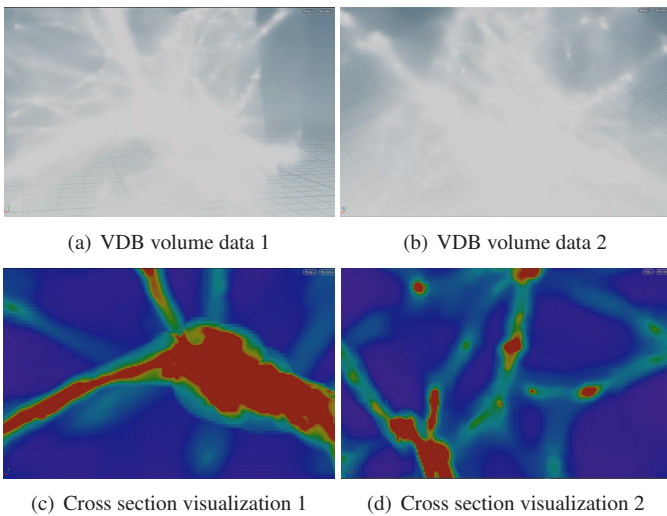


Figure 2: Interstellar material data visualization results

2.2.2 성단 데이터 분리

성단 데이터는 파일의 크기가 성간물질 데이터보다는 작기 때문에 한 프레임의 모든 파티클 데이터를 하나의 파일에 저장해도 렌더링이 가능하다. 그러므로 특별한 격자 분할 및 최적화 알고리즘이 필요하지 않아 C++ 파싱 알고리즘을 작성하여 데이터를 추출하였다. 모든 파티클 파일을 순차적으로 로딩하면서 필요한 데이터를 추출한 후 Houdini에서 읽을 수 있는 bgeo 파일 포맷으로 디스크에 저장하였다. 그림 3은 저장된 bgeo 파일을 Houdini에서 로드하여 가시화한 결과이다.

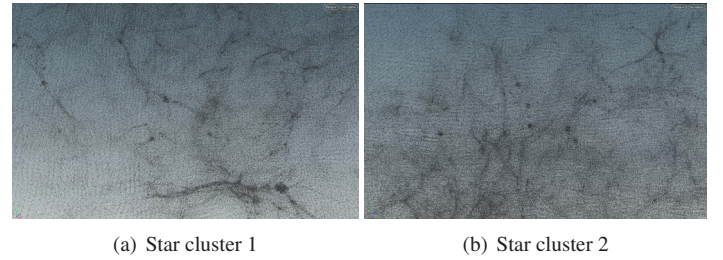


Figure 3: Star cluster data visualization results

2.3 성단 데이터 추출

앞서 변환된 데이터들은 상용 소프트웨어인 Houdini에서 렌더링이 가능한 데이터 포맷으로 변환된 상태이다. 최종 목표인 저사양 XR 디바이스에서 데이터를 렌더링하려면 추가적인 경량화가 필요하다. 본 논문에서는 성간물질과 성단 데이터 중 XR 디바이스에서 활용할 데이터로 성단 데이터를 선택하였다. 해당 데이터에는 성단의 움직임과 질량 등 가시화에 필요한 필수적인 파티클 정보를 포함하고 있다(표 2 참조).

그림 4는 2.2.2절에서 분리된 성단 데이터의 정보로 그림 4(a)는 여러 개의 프레임이 저장된 파일로, 하나의 프레임당 약 1.6GB의 용량을 차지하는 것을 확인할 수 있다. 또한 그림 4(b)는 하나의 프레임을 Houdini로 렌더링한 결과로 프레임 하나에는 적게는 수십만에서 많게는 수억개의 파티클이 매우 넓은 공간에 분포되어 존재하는 것을 볼 수 있다.

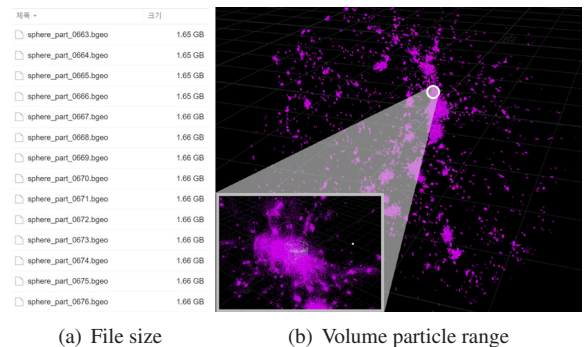


Figure 4: File size and volume range of particle data

그림 4와 같이 대부분의 파티클은 넓은 공간에 분포되어 있고 모바일 디바이스에 모든 파티클 데이터와 프레임을 로드하는 것은 불가능한 일이다. 이와 같은 데이터를 XR 디바이스에서 사용하기 위해 기준이 되는 데이터를 추출하기 위한 파싱(parsing) 공간의 크기를 우선적으로 설정해야 한다. 공간의 크기는 실감 콘텐츠 체험 공간의 크기로 결정했기 때문에 콘텐츠마다 파싱 공간이 다른데, 5.1절의 콘텐츠의 경우 지름 12m, 높이 2.5m인 실린더 모양으로 설정되었다. 또한 추후 디바이스의 변경과 확장성을 위해 설정된 파싱 공간 안의 최대 파티클 개수를 20만개 이하로 저장 되도록 하였다.

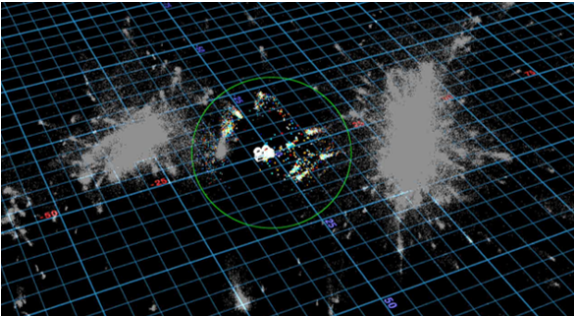


Figure 5: Parsing space configuration (green circle)

그림 5는 파싱 공간을 시각화한 것으로 초록색 원으로 표시된 영역은 파싱 영역이며 이 영역 내부로 들어온 파티클들이 추출될 파티클이다. 영역 내의 파티클들은 시각화를 하기 위해 파티클의 컬러값으로 렌더링을 하였고 외부 파티클은 회색으로 표시하여 어떠한 파티클이 파싱이 되는지 한눈에 볼수 있도록 하였다. 추출된 데이터는 6개의 파트로 구분되어 있고 한 프레임마다 하나의 파일로 저장되어 있다. 파티클을 구성하는 데이터의 프레임은 총 5,400프레임이다. 추출된 데이터의 구성은 표 3과 같다.

Table 3: Extracted data structure

variable name	data type	meaning
frame	int	frame order
position	float(x, y, z, w)	position of particle
id	int	particle's unique id
size	float	size of particle
color	float(r, g, b)	color of particle
alpha	float	transparency of particles

표 3을 보면 기존 원본 데이터 표 2와 몇가지 다른 것을 확인할 수 있는데 이는 파티클 렌더링을 할 때 필요한 데이터만 필터링하고 압축하여 추출하였기 때문이다. 대표적으로 color 값이 있는데 표 2에서 색과 관련된 변수들을 이용하여 디바이스에서 시각화할 수 있도록 RGB 컬러값으로 변환한 것이다. 표 3에 포함된 값들은 id를 제외하고 매 프레임마다 변화한다. 차례대로 설명을 하면 position은 현재 프레임에서 파티클의 위치를 나타내는 3차원의 파티클 위치 값이다. id는 모든 프레임에 걸쳐 파티클의 정보를 식별할 수 있는 고유 번호로 사용된다. size는 성단의 크기

이며 alpha값의 경우 앞서 설정했던 파싱 공간에 들어온 파티클들이 자연스럽게 나타나도록 파싱 공간 외곽에서 파티클이 공간 안으로 들어올때 알파값의 변화를 통해 자연스럽게 나타나거나 없어지도록 표현하기 위해 추가로 계산한 값이다. color는 성단의 색을 나타낸다. 이렇게 생성된 데이터는 다른 디바이스에서 렌더링에 필요한 기초 데이터로 사용된다.

3. 파티클 데이터 최적화

앞 단계에서 생성된 데이터 집합인 범용 플랫폼 데이터셋의 용량은 작게는 500MB에서 크게는 30GB가 넘는다. 이러한 규모의 데이터는 PC 및 모바일 디바이스의 메모리 용량을 초과하는 경우가 많기 때문에 한 번에 모든 데이터를 메모리에 로드할 수 없다. 따라서 필터링을 통해 데이터의 용량을 타겟 디바이스의 메모리 용량에 적합한 수준으로 줄이는 작업이 필요하다. 하지만 필터링을 적용하면 원본 데이터의 손실이 불가피하게 일어난다는 문제가 있다. 원본 데이터의 손실은 곧 파티클 가시화 품질 저하로 나타난다. 파티클 가시화 품질 저하 문제를 최소화하기 위해서는 파티클 가시화시 화면에 잘 나타나지 않는 파티클을 우선적으로 찾아 제거해야 한다. 파티클 데이터를 가시화하면 그림 5처럼 파티클이 뭉쳐있는 영역이 있는 것을 확인할 수 있다. 파티클이 뭉쳐진 곳은 근접한 이웃 파티클에 의해 내부의 파티클이 가려져 보이지 않는다. 즉 이러한 파티클을 찾아서 필터링한다면 파티클 가시화 퀄리티 저하를 최소화하며 렌더링 속도를 향상시킬 수 있을 것이다. 본 절에서는 필터링을 위해 데이터 정렬과 압축을 통해 전체적인 데이터의 크기를 줄이는 방법을 설명한다. 그리고 팔진트리(octree) 공간 분할 사용용 파티클이 뭉쳐있는 영역을 효율적으로 찾아내는 방법을 통해 팔진트리 노드 공간에서 보이지 않는 파티클을 찾아내 필터링하는 방법을 소개한다 [6].

3.1 성단 데이터 가공

3.1.1 성단 데이터 정렬

앞서 추출된 성단 데이터들은 하나의 프레임에 존재하는 파티클들을 기준으로 값이 저장되어 있다. 그림 6은 추출된 성단 데이터의 원본 데이터 구조를 도식화한 것이다. 각각의 프레임에 파티클의 position, id, size, color와 alpha 값이 존재한다. 이러한 데이터 구조는 프레임에 파티클 아이디가 종속되어 있어서 N번째 프레임에 M번 id의 파티클이 존재하는지를 찾기 위해 N번 프레임의 모든 id를 탐색하여 M번 id가 있는지 찾아야하는 불합리한 과정이 필요하다. 또한 추출된 성단 데이터의 경우 해당 id의 파티클이 연속된 프레임에서 중간 몇 프레임에 걸쳐 파티클 데이터가 누락되어 있는 경우도 존재한다. 이러한 경우 하나의 프레임을 전부 탐색하더라도 데이터를 획득할 수 없다.

위와 같은 문제점을 해결하기 위해 그림 7과 같이 프레임 기준에서 id 기준으로 정렬하도록 구조를 변경하였다. 변경된 데

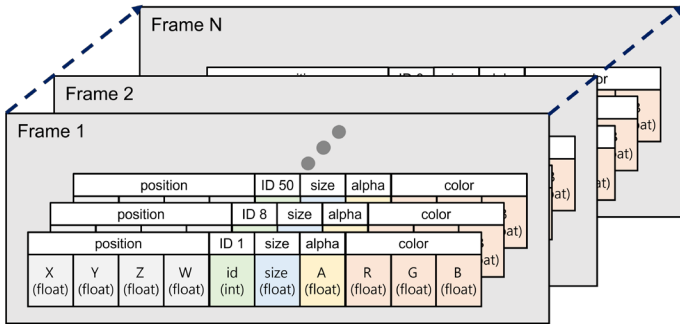


Figure 6: Original star cluster data structure

이터 구조를 통해 M번 id의 파티클이 N번째 프레임에 존재하는 지 찾기 위해 탐색 과정 없이 아이디와 프레임을 인덱스(index)로 활용할 수 있어 데이터에 바로 접근이 가능하다. 정렬된 성단 데이터를 기반으로 이후 프레임 압축과 팔진트리를 통한 파티클 필터링을 원활하게 진행할 수 있다.

	Frame 1	Frame 2	Frame 3	Frame 4	Frame 5	...	Frame N
ID 1	position(X,Y,Z) color(R,G,B,A) brightness size	position(X,Y,Z) color(R,G,B,A) brightness size	position(X,Y,Z) color(R,G,B,A) brightness size	position(X,Y,Z) color(R,G,B,A) brightness size	position(X,Y,Z) color(R,G,B,A) brightness size		NULL
ID 2	NULL	NULL	NULL	position(X,Y,Z) color(R,G,B,A) brightness size	position(X,Y,Z) color(R,G,B,A) brightness size	...	position(X,Y,Z) color(R,G,B,A) brightness size
⋮							
ID N	NULL	position(X,Y,Z) color(R,G,B,A) brightness size	position(X,Y,Z) color(R,G,B,A) brightness size	NULL	position(X,Y,Z) color(R,G,B,A) brightness size		NULL

Figure 7: Sorted cluster data structure

3.1.2 중간 프레임 스킵

데이터 정렬을 통해 접근성과 어느정도의 데이터 크기가 줄어들었지만 아직 XR 디바이스에서 사용할 수 있는 정도는 아니다. 또한 프레임 N과 프레임 N+1 사이의 파티클의 위치 변화는 작아서 몇 프레임을 스킵(skip)하여 전체적인 데이터의 크기 자체를 줄일 수 있다. 그림 7에서 프레임 1과 프레임 5만 남기고 2, 3, 4를 제거하더라도 실제 눈에 보이는 파티클 위치 변화는 그리 크지 않다고 가정할 수 있지만, 이렇게 변환된 파티클 중 빠른 속도로 움직이는 경우 스킵된 프레임으로 인해 파티클이 점프하는 것처럼 보여지게 된다. 이런 문제를 방지하기 위해 XR 디바이스에서 렌더링될때 누락된 프레임의 position과 color 값을 선형 보간법을 통해 중간 프레임을 생성하여 부드럽게 파티클이 변화하도록 하였다. 이와 같이 중간 프레임들을 제거하고 제거된 프레임을 보간하여 처리하면 전체적인 데이터의 크기 자체가 줄어들어 일반 PC에 비해 상대적으로 적은 메모리를 지닌 XR 디바이스에서도 충분히 데이터를 로드할 수 있다.

3.2 팔진트리 공간 분할

파티클 데이터는 3차원 공간 속 위치 정보를 가지고 있고, 이는 포인트 클라우드(point cloud)로 표현할 수 있다 [7]. 팔진트리 공간 분할은 3차원 공간을 분할하는 기법 중 하나로 하나의 노드가 여덟개의 자식노드를 가지는 트리 자료구조이다(그림 8 참조). 하나의 팔진트리 노드는 하나의 육면체 공간을 의미하고 노드를 분할하면 해당 공간을 여덟개의 작은 육면체로 분할하게 된다. 이때 생긴 여덟개의 작은 육면체 공간은 자식노드를 의미한다.

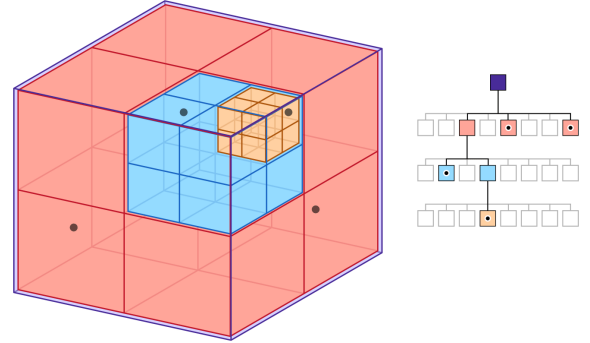


Figure 8: Octree space segmentation [8]

팔진트리 공간 분할의 경우 노드 안에 설정한 임계치(threshold value) 값 이상의 파티클이 존재하는 경우에만 노드를 분할한다. 팔진트리 공간 분할 알고리즘은 아래 알고리즘 1과 같다. 노드 분할은 설정한 최대 깊이(depth) 값까지 반복할 수 있다. 분할을 반복할 수록 노드가 작아지므로 공간을 세밀하게 나눌 수 있지만 계산량도 비례하여 늘어난다.

Algorithm 1 Octree spatial subdivision algorithm

Require: $P_n \leftarrow$ particle datas, $D \leftarrow$ max depth, $T \leftarrow$ threshold

```

1: for  $d \leftarrow 0$  to  $D$  do
2:   Octree  $\leftarrow P_n$       ▷ input particle datas in Octree
3:   for all leaf node in Octree do
4:     if  $n \geq T$  then      ▷ n is number of particles in leaf node
5:       node split
6:     end if
7:   end for
8: end for

```

그림 9는 파티클과 팔진트리 공간 분할 결과를 함께 가시화한 모습이다. 팔진트리 노드는 파티클이 일정 수 이상인 것만 가시화했으며 파티클 수가 적을수록 파란색, 많을수록 빨간색으로 표시된다. 그리고 노드의 크기가 작을수록 파티클이 밀집된 영역을 의미한다.

3.3 필터링

앞서 언급했듯이 필터링의 목적은 파티클의 수를 줄이는 것과 동시에 기존 형상을 최대한 유지하는 것이다. 형상 유지를 위해 파티클이 뭉쳐있는 곳을 찾고 외부의 파티클에 의해 보이지 않을

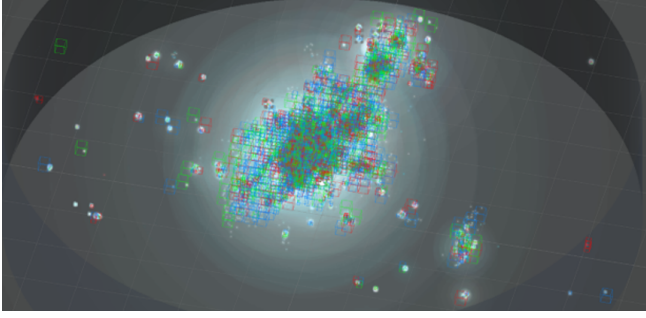


Figure 9: Octree visualization

파티클을 제거하는 필터링 방법을 사용해야 한다[9]. 팔진트리 공간 분할을 이용하여 파티클이 뭉친 구역을 효과적으로 찾아낸 후 팔진트리 노드에서 필터링을 진행한다. 필터링에 앞서 팔진트리 노드 속 파티클의 우선순위를 계산해야 한다. 아래 기준에 따라 중요도가 계산되며 계산 결과에 따라 필터링 우선순위가 정해진다. 필터링시 중요도가 낮은 파티클부터 제거된다.

- size와 brightness가 큰 파티클은 가시화시 크게 렌더링 된다. 큰 파티클은 다른 파티클에 의해 가려지지 않기 때문에 중요도가 높다.
- 팔진트리 노드 중심과 가까운 파티클일수록 중요도가 높다.

노드 내 중요도 계산 이후 필터링을 진행한다. 필터링은 알고리즘 2로 동작하며 노드의 상태에 따라 다른 방식으로 진행된다. 예를 들어 최대 깊이까지 분할했음에도 불구하고 파티클의 수가 임계치 이상인 노드의 경우 좁은 공간 안에 많은 파티클이 밀집되어 있기 때문에 1개의 파티클만 남기고 모두 제거해도 형상의 변화가 미미했다. 따라서 가장 우선순위가 높은 파티클 1개만 남기고 나머지는 모두 제거한다. 그 외 다른 노드들은 깊이와 파티클 수에 따라 제거할 파티클의 비율을 결정한다. 마지막으로 깊이가 낮고 파티클이 적게 존재하는 노드의 경우 파티클들이 서로 겹치지 않기 때문에 파티클을 제거하지 않는다.

Algorithm 2 Octree filtering algorithm

Require: $D \leftarrow \text{max depth}$, $T \leftarrow \text{threshold}$

```

1: for all leaf node in Octree do
2:   if  $n > T$  then           ▷  $n$  is number of particles in leaf node
3:     Leave only 1 particle and remove the rest.
4:   else if  $l == D$  then
5:     Removes 80% of particles
6:   else if  $l == D-1$  then
7:     Removes 60% of particles
8:   else if  $c \geq 0.5 * T$  then
9:     Removes 50% of particles
10:  end if
11: end for

```

필터링은 원하는 결과가 나올때까지 반복할 수 있다. 그림 10은 필터링 수행 횟수에 따른 가시화 결과의 차이를 보여준다. 원

쪽부터 원본, 필터링 1회, 필터링 2회 결과를 가시화하였다. 필터링을 반복할 경우 필터링된 파티클에 대해 팔진트리 공간 분할을 다시 실시한 후 필터링을 진행한다. 밀집도가 높을수록 붉은색으로 표현하였고 중간 밀집도는 파랑, 낮은 밀집도는 초록 박스로 표현하고 박스의 크기로 깊이를 나타냈다. 필터링을 반복할수록 노드의 밀집도는 줄어들고 박스의 크기가 커져 더 낮은 깊이로 내려가지 않을 만큼의 밀집도로 파티클이 필터링되는 것을 확인할 수 있다.

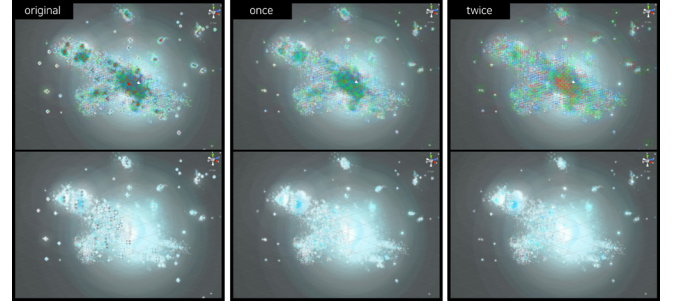


Figure 10: Wide area filtering result

형상 변화를 보다 쉽게 비교하기 위해 그림 10의 파티클의 국소 부분에 대해 필터링을 다시 진행하였다(그림 11 참조). 필터링 1회 결과는 원본과 형상 차이가 거의 없으나 필터링 2회차에는 원본과 비교해 파티클 밀도가 낮은 것을 확인할 수 있다. 파티클 형상을 유지하고 싶다면 필터링 강도를 약하게 변경하여 필터링을 다시 진행해야 한다. 반대로 파티클의 수를 더 줄여야 한다면 추가로 필터링을 반복하거나 필터링 강도를 강하게 변경하여 필터링을 다시 하면 된다. 강도와 반복 횟수를 조정할 수 있는 필터링 기능을 통해 다양한 XR 디바이스 성능에 맞도록 데이터를 더욱 경량화하여 렌더링할 수 있도록 하였다.

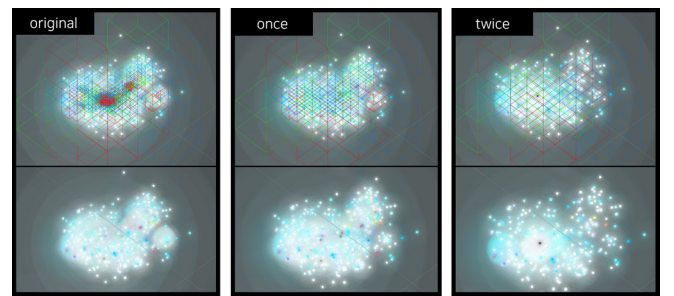


Figure 11: Narrow area filtering result

4. 파티클 가시화

파티클을 실시간으로 가시화하기 위해서 컴퓨터 셰이더(compute shader)를 활용하여 렌더링하였다. 컴퓨터 셰이더는 GPGPU(General-Purpose computing on GPU)를 이용해 병렬 처리를 수행하는 셰이더(shader)로 각기 다른 데이터를 가진 파티클을 렌더링하는데 효과적으로 활용할 수 있다 [10]. GPU로

파티클을 구성하는 데이터셋을 넘겨주는데, 표 4는 컴퓨트버퍼 (computebuffer)에 저장되는 파티클의 데이터를 보여준다.

variable name	data type	meaning
x, y, z	float	position
r, g, b, a	float	color
size	float	star size
brightness	float	brightness

GPU에서는 컴퓨트버퍼의 데이터 셋을 활용하여 파티클을 렌더링한다. 각 파티클은 position(x,y,z) 위치에 생성되고 color(r,g,b,a)에 따라 색이 표현된다. 추가로 RGB 값을 휘도로 변환해주는 공식을 사용하여 brightness 값을 추출하였다. 파티클의 크기는 brightness에 비례하여 콘텐츠에 정의된 최소 최대 크기로 매핑되어 표현된다.

본 논문에서는 파티클을 단순한 원과 색으로 표현하지 않고 실제 성단이 빛나는 것처럼 보여주기 위해 여러 텍스처를 사용하였다. 각각의 파티클 데이터 셋에 포함된 brightness 값 크기에 따라서 매핑될 텍스처가 선택된다. 그림 12는 brightness 값에 따라 선택되는 텍스처를 보여준다. brightness가 낮은 파티클은 모든 방향으로 빛을 고르게 방사하는 형태의 level 1 텍스처가 사용되고 brightness가 높을 수록 주변으로 뻗어나가는 빛 줄기를 추가하여 파티클의 가시성을 높였다. 추가로 brightness의 값의 크기에 따라 텍스처 자체의 크기를 조절하였다.

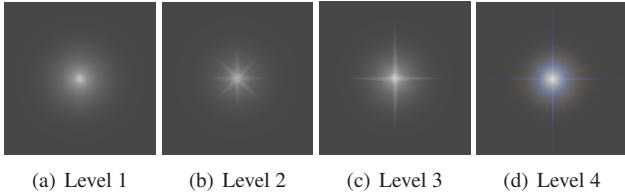


Figure 12: Texture used by brightness level

표 4의 size는 별의 크기를 결정한다. brightness는 성단이 내뿜는 광원의 크기고 size는 실제 성단을 이루는 물질들의 크기를 나타내는 것이다. 성단은 수많은 항성들로 구성된 무리이지만 이를 하나의 항성처럼 표현하였다. 그림 13(a)을 보면 파티클 중앙에 흰색 원이 표현된 것을 볼 수 있다. 이 흰색 원이 성단의 크기를 의미하며 size에 비례하여 크기가 계산된다. brightness로 결정되는 텍스처의 크기와 size에 의해 결정되는 항성의 크기는 서로 영향을 끼치지 않는다. 항성의 크기는 알고리즘 3에 의해 계산된다. 마지막으로 천체 주변에 빛의 굴절과 반사와 같은 빛무리를 표현하기 위해 알고리즘 4를 사용하여 성단에서 뿜어져 나오는 빛을 더 실제 같이 렌더링하였다.

최종적으로 렌더링된 파티클은 그림 13(a)이고, 그림 13(b)은 NASA의 허블 우주망원경으로 촬영된 실제 천체 사진이다. 그림 13(b)을 보면 별들의 중심은 흰색 계열로 원 형태로 표현된 것

Algorithm 3 Star rendering function

Require: $x, y, size, brightness$

```

1:  $x, y \leftarrow UV$  ▷  $x, y$  is UV coordinates
2:  $r \leftarrow size/brightness$  ▷  $r$  is a star radius
3:  $v \leftarrow (x - 0.5)^2 + (y - 0.5)^2 + r^2$ 
4: if  $v < 0$  then
5:   Rendering star
6: else
7:   Rendering texture
8: end if

```

Algorithm 4 Center shines brightly function

Require: $x, y, size, intensity, e, v$

```

1:  $x, y \leftarrow UV$  ▷  $x, y$  is UV coordinates
2:  $x \leftarrow -x^e$ 
3:  $y \leftarrow -y^e$ 
4: return  $(x + y + v) * intensity$ 

```

을 확인할 수 있으며 더 밝은 빛을 뿜어내는 항성의 경우 선명한 빛줄기가 뻗어나가는 것을 확인할 수 있다. 그림 13(b)처럼 실제 같이 파티클을 렌더링하기 위해 앞서 소개한 방법을 적용하여 실제 항성과 흡사한 특징을 가진 가시화 결과를 얻을 수 있었다.

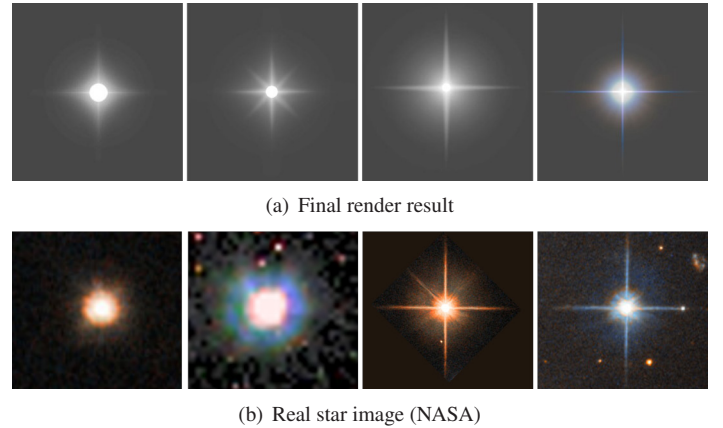


Figure 13: Comparison of real star and rendering result

5. 실감 콘텐츠 제작 사례

사례에서 소개할 콘텐츠에는 상용 S/W 연출 데이터를 기반으로 콘텐츠 체험 환경에 맞게 제작된 멀티 프로젝터용 동영상과 커다란 스크린에 사영하여 배경으로 사용된다. 또한 각각의 XR 디바이스에 맞춰 최적화된 XR 전용 데이터를 사용하여 상이한 디바이스의 성능에도 최소 권장 프레임을 보장하도록 하였다. 이렇게 제작된 2개의 배경 동영상과 배경 앞쪽에 XR 디바이스를 통해 렌더링되는 파티클을 통해 높은 몰입감을 체험자에게 제공한다. 소개할 사례들은 멀티 플랫폼 디바이스를 사용해서 체험할 수 있도록 제작되었으며 같은 콘텐츠일 경우라도 사용하는 디바이스에 따라 다른 사용자 경험을 제공한다.

5.1 확장현실 우주(XR universe)[11]

해당 사례는 반지름 6m, 높이 2.55m의 곡면 스크린에 4K 빔 프로젝터 3대로 하나의 스크린으로 구성된 체험 공간에서 구동된다. 확장현실 우주 콘텐츠는 우주 속 은하와 성단 그리고 별들을 탐험하기 위해 모험을 떠나는 시나리오를 가지고 있다. 모험에 도움을 줄 가이드 로봇 캐릭터인 ‘갤럭시’와 함께 우주를 여행하는 체험형 실감 콘텐츠이다. 인트로와 아웃트로를 포함한 총 5개의 우주 탐험 시나리오를 포함하고 있으며 총 플레이 타임은 약 6분 정도이다. 모든 체험이 끝난 뒤 AR 기능을 활용한 기념촬영 이벤트도 제공한다. 사전 기획을 통해 가이드 캐릭터의 음성, 배경 사운드, 효과음과 같은 청각 효과를 제작하여 더욱 실감나는 콘텐츠 체험이 가능하다. 그림 14는 서울특별시교육청 과학전시관 남산분관에 구축된 확장현실 우주 실감 콘텐츠를 체험하고 있는 모습을 보여준다.



Figure 14: Extended reality space content experience scene(AR)

체험자들은 손에 XR 디바이스를 들고 디바이스의 카메라를 스크린 방향으로 향한다. 그러면 스크린에 보이는 동영상과 XR 디바이스에서 실시간으로 렌더링하는 파티클을 동시에 디스플레이 하게 된다. 또한 스크린에서 보이던 가이드 캐릭터가 스크린을 뚫고 나와 AR로 증강되는 효과, 스크린에서 쏟아져 나오는 파티클의 파도와 같이 색다른 경험을 제공할 수 있는 기능들을 포함한다 [12]. 스크린 동영상과 증강 파티클을 동시에 체험함으로써 체험자에게 새로운 사용 경험을 제공한다.

5.2 로드트립(Road Trip) [13]

로드트립은 한국과학기술정보연구원(KISTI) 2층에 설치된 가상 실증실험 시설 CAVE [14]에 설치된 콘텐츠이다. CAVE는 4K 프로젝터 13대를 사용하여 천장이 없는 역돔 형태의 지름 9m, 높이 3.5m 스크린에 영상을 프로젝션할 수 있는 공간이다. 추가로 사용자의 위치를 추적할 수 있는 광학식 트래킹 시스템이 설치되어 있다.

그림 15는 CAVE에 설치된 로드트립 콘텐츠를 플레이하고 있는 모습을 제 3자의 입장에서 촬영한 것으로 지름 9m의 커다란

스크린안에 여러 명의 사람들이 입장하여 동시에 콘텐츠를 체험하는 모습을 확인할 수 있다.



Figure 15: Road Trip experience environment

로드트립은 답답한 도시를 벗어나 숲속 캠핑장으로 자동차를 타고 여행을 떠나는 스토리를 가지고 있다. 도시에서 숲속 캠핑장까지 이동하는 5개의 씬과 캠핑장에서 우주로 이동하여 밤 하늘의 별을 관찰할 수 있는 우주 씬 2개를 포함하여 총 7개의 씬이 존재한다. 로드트립 콘텐츠 제작을 위해 앞서 생성된 상용 S/W 연출 데이터로 우주 씬 2개의 백그라운드 동영상과 렌더링하였다. 로드트립도 5.1절에서 소개한 것처럼 스크린에 동영상이 플레이되고 XR 디바이스를 통해 스크린 안쪽을 바라보면 파티클이 증강되어 나타난다. 그림 16은 CAVE 안에서 XR 디바이스로 공간을 바라보았을 때 나타나는 파티클의 렌더링 결과를 보여준다.

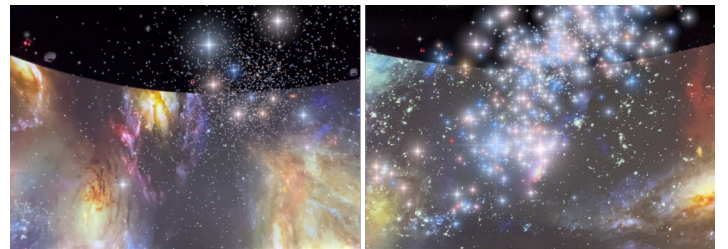


Figure 16: Road Trip experience scene (AR)

해당 콘텐츠는 360도 역돔 형태의 스크린이 구성되어 있는 CAVE 공간에서 체험 가능하도록 제작되었다. 이런 구조 특성을 이용하여 360도 스크린 모든 공간에서 별들이 쏟아져 들어오고 나가는 역동적인 연출을 통해 우주 공간에 존재하는 성단들의 움직임을 아름답게 표현할 수 있다.

6. 실험 결과

본 절에서는 최종 생성된 XR 전용 데이터를 실제 XR 디바이스를 통해 실험한 결과에 대해 설명한다. 사용되는 XR 디바이스는 Apple의 iPad와 Microsoft의 HoloLens 2이다. 두 디바이스의 성능은 표 5와 같다.

실험에 사용된 XR 전용 데이터는 5.1절의 확장우주 콘텐츠 시나리오 중 파티클의 특징이 잘 나타나는 프레임 1개만 추출하여 실험에 사용하였다. 또한 프레임 내의 뭉쳐있는 여러 성단 무리 중 파티클의 형태가 뚜렷하고 다양한 색을 포함하는 크고 작은 파티클들이 모여있는 부분을 확대하여 실험을 진행하였다. 각각의

Table 5: XR device performance comparison

	iPad Pro 12.9(4th)	HoloLens 2
Processor	Apple A12Z Bionic	Snapdragon 850
RAM	6GB	4GB
Resolution	2732×2048	2048×1080(per eye)

AR, MR 성능에 최적화하기 위해 반복 필터링 횟수를 다르게 하여 렌더링될 파티클의 개수를 조절하였다.

6.1 AR 디바이스 실험 결과

AR 디바이스로 사용된 iPad에 사용될 파티클들은 표 6와 같이 총 4단계로 필터링을 반복하였다. iPad의 경우 HoloLens 2보다 성능이 높기 때문에 필터링 반복 수를 최대 20번까지 진행하였다. 필터링을 하지 않은 원본 데이터의 경우 약 4만8천개의 파티클이 포함되어 있고 20번을 반복한 경우 오리지널 데이터의 1/10인 4천6백개의 파티클로 줄어든다.

Table 6: Number of particles used in AR

original	1	10	20
48,764	12,432	6,784	4,649

그림 17은 필터링된 파티클 개수에 따른 1분 동안 측정한 평균 렌더링 속도(fps)를 측정한 그래프이다. 원본 데이터부터 20번 반복 필터링까지 선형적으로 속도가 증가하는 것을 확인할 수 있다. 최종적으로 20번 반복 필터링을 할 경우 약 47fps를 나타내며 이는 콘텐츠를 충분히 AR 콘텐츠를 체험할 수 있는 성능을 보장한다.

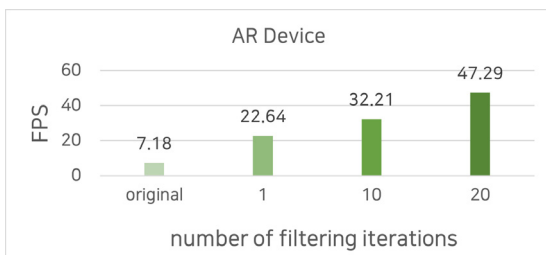


Figure 17: FPS by number of filtering iterations(AR)

그림 18은 필터링 횟수별 렌더링된 파티클을 보여준다. 원본 파티클과 필터링을 1번한 것은 파티클의 총 수가 1/4이 되었지만 파티클이 구성하는 형태와 특징이 되는 파티클의 차이가 거의 없다. 20번 필터링을 반복해 오리지널 대비 1/10 수준인 파티클 수를 가지고 있는 경우는 뭉쳐있는 자그마한 파티클이 제거가 되었지만 전체적인 형태에는 큰 변화가 없는 것을 확인할 수 있다.

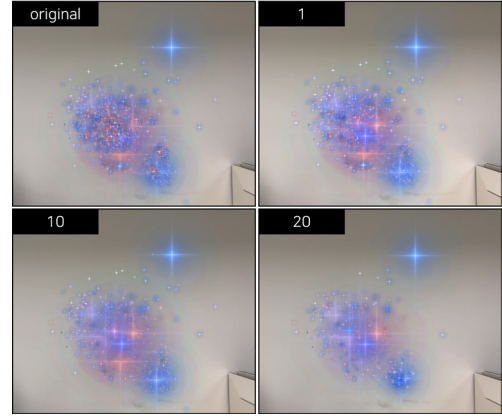


Figure 18: Filtering test result(AR)

6.2 MR 디바이스 실험 결과

MR 디바이스는 HoloLens 2를 사용하였다. HoloLens 2는 AR 디바이스로 사용된 iPad와 비교하여 프로세서의 성능이 낮다. 그래서 MR 디바이스는 AR보다 더 많은 필터링을 반복하여 파티클의 수를 더 많이 줄여야 했다. 표 7는 실험에 사용된 필터링 반복 횟수와 파티클의 개수를 보여준다. AR 대비 2배 더 많이 필터링을 하였고, 필터링 횟수가 30번을 넘어가면서 추가 필터링을 할 경우 제거되는 파티클의 수가 줄어드는 것을 볼 수 있다.

Table 7: Number of particles used in MR

original	10	30	40
48,764	6,784	3,459	2,729

그림 19는 표 7의 필터링 반복 횟수에 따른 파티클 렌더링 성능을 보여준다. AR과 달리 원본 데이터는 MR 디바이스 RAM에 업로드되지 않아 실행조차 불가능하였다. 마찬가지로 1회 반복의 경우에도 실행이 불가능하여 최소 필터링 반복 횟수를 10번으로 변경해 테스트를 진행하였다. 필터링을 지속적으로 진행할 경우 AR보다 증가하는 fps 폭은 적었으며 30번 이후에는 속도 증가폭이 줄어들었다.

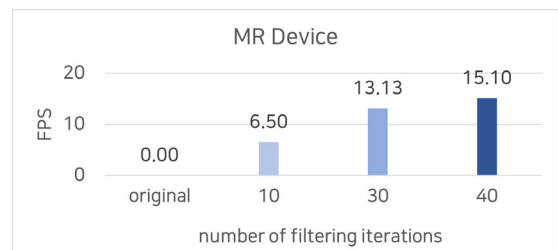


Figure 19: FPS by number of filtering iterations(MR)

그림 20은 필터링 반복 횟수에 맞게 MR에서 파티클을 렌더링한 것이다. AR과 마찬가지로 반복 횟수에 따라 파티클의 전체 개수는 떨어지지만 그림 18와 비교해 보면 특징이 되는 파티클들은 형태를 유지하고 있음을 확인할 수 있다.

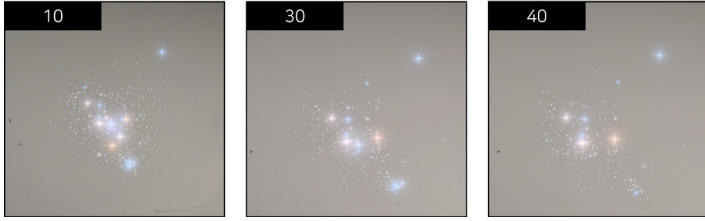


Figure 20: Filtering test result(MR)

7. 결론 및 향후 연구

본 논문에서는 우주 천체 시간 가변 AMR 볼륨 데이터를 가공하여 저사양 XR 디바이스에서 활용하기 위해 다양한 방법을 거쳐 경량화된 데이터로 변환하는 방법에 대해 설명하였다. 원본 AMR에서 XR 디바이스에서 사용할 성단 데이터를 분리한 뒤 분리된 성단 데이터로부터 원하는 격자 영역의 데이터를 추출하고 추가적인 필터링을 통해 XR 디바이스에서 사용 가능한 전용 데이터를 제작하였다. 또한 데이터 제작과 테스트에 더해 수많은 파티클을 실제 성단과 비슷하게 가시화하고, XR 디바이스를 활용한 체험 가능한 실감 콘텐츠를 개발하여 실증하였다. 5절에서 사례로 든 2종의 실감 콘텐츠는 현재 상설 전시중이며 모든 관람객들에게 공개되어 있어 콘텐츠의 활용성과 안정성을 검증하였다.

실감 콘텐츠 제작 사례에서 활용된 XR 디바이스는 AR, MR 디바이스 두 종이다. MR은 HoloLens 2를 활용하였는데, 착용자는 투명 디스플레이를 투과하여 현실 공간을 보면서 3D 오브젝트를 증강할 수 있다. 이런 방식과 비슷하게 HMD(head-mounted display)에 장착된 카메라를 통해 현실을 촬영하고 이를 3D 렌더링 화면과 합성하여 디스플레이에 표출하는 비디오 씨스루(video see-through) 방식이 많이 활용되고 있다 [15]. 향후 이런 방식을 사용하는 VR 디바이스를 시작으로 다양한 XR 디바이스 플랫폼에 적용할 수 있도록 확장할 예정이다.

감사의 글

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구이고(RS-2023-00251681), 2023년도 서강대학교 교내연구비 지원에 의한 연구임(202310038.01). 또한 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터사업(RS-2023-00259099)과 메타버스융합대학원(RS-2022-00156318)의 연구 결과로 수행되었음. 그리고 본 연구는 한국과학기술정보연구원의 산업 및 공공분야 문제 해결을 위한 초고성능컴퓨팅 활용 기술 개발 사업(K-22-L02-C05-S01)의 지원을 받아 수행되었음.

References

- [1] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of computational Physics*, vol. 53, no. 3, pp. 484–512, 1984.
- [2] R. D. Hornung, A. M. Wissink, and S. R. Kohn, "Managing complex data and geometry in parallel structured amr applications," *Engineering with Computers*, vol. 22, pp. 181–195, 2006.
- [3] J. Lee, J. Shin, O. N. Snaith, Y. Kim, C. G. Few, J. Devriendt, Y. Dubois, L. M. Cox, S. E. Hong, O.-K. Kwon, *et al.*, "The horizon run 5 cosmological hydrodynamical simulation: Probing galaxy formation from kilo-to gigaparsec scales," *The Astrophysical Journal*, vol. 908, no. 1, p. 11, 2021.
- [4] SideFX, <https://www.sidefx.com/products/houdini/>, 2022.
- [5] OpenVDB, <https://www.openvdb.org/>, 2022.
- [6] R. Schnabel and R. Klein, "Octree-based point-cloud compression," *PBG@ SIGGRAPH*, vol. 3, 2006.
- [7] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1–4.
- [8] Apple, developer.apple.com/documentation/gameplaykit/gkoc-tree, 2023.
- [9] H. Woo, E. Kang, S. Wang, and K. H. Lee, "A new segmentation method for point cloud data," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 2, pp. 167–178, 2002.
- [10] M. Bailey, "Using gpu shaders for visualization, part 3," *IEEE computer graphics and applications*, vol. 33, no. 3, pp. 5–11, 2013.
- [11] VIZinf, <https://youtu.be/ZUdKCzwVWYw>, 2023.
- [12] J. Kim, J. Song, J. Park, J. Nam, S.-H. Yoon, and S. Park, "Mixed reality extension system using beam projectors: Beyond the sight," *Journal of the Korea Computer Graphics Society*, vol. 25, no. 3, pp. 65–73, 2019.
- [13] VIZinf, <https://youtu.be/a996qNe5LSE>, 2023.
- [14] KISTI, <https://youtu.be/QV2fS7j0PW0>, 2022.
- [15] J. P. Rolland, R. L. Holloway, and H. Fuchs, "Comparison of optical and video see-through, head-mounted displays," in *Telemanipulator and Telepresence Technologies*, vol. 2351. SPIE, 1995, pp. 293–307.

〈저자 소개〉



김 중 용

- 2014 청운대학교 컴퓨터학과 학사
- 2016 동국대학교 멀티미디어학과 석사
- 2017 ~ 현재 동국대학교 멀티미디어학과 박사과정
- 2017 ~ 현재 (주)비즈아이엔에프 책임연구원
- 관심분야 : 혼합현실, 가상현실, 컴퓨터 그래픽스
- <https://orcid.org/0000-0001-8062-3678>



박 상 훈

- 1993 서강대학교 수학과 학사
- 1995 서강대학교 컴퓨터학과 석사
- 2000 서강대학교 컴퓨터학과 박사
- 2002 ~ 2005 대구카톨릭대학교 컴퓨터정보통신공학부조교수
- 2001 University of California, Davis 방문연구원
- 2005 ~ 2023 동국대학교 멀티미디어학과 교수
- 2023 ~ 현재 서강대학교 메타버스전문대학원 교수
- 관심분야: 실시간 렌더링, 사실적 렌더링, 과학적 가시화, 고성능 컴퓨팅 등
- <https://orcid.org/0000-0001-5383-7005>



송 중 훈

- 2018 동국대학교 멀티미디어공학과 학사
- 2018 ~ 2020 동국대학교 멀티미디어학과 석사
- 2020 ~ 현재 (주)비즈아이엔에프 선임연구원
- 관심분야 : 혼합현실, 가상현실, 실시간 렌더링
- <https://orcid.org/0000-0001-8062-3678>



황 규 현

- 2006년 대구대학교 멀티미디어학과 학사
- 2008년 동국대학교 멀티미디어학과 석사
- 2013년 동국대학교 멀티미디어학과 박사
- 2013년 ~ 현재 한국과학기술정보연구원 국가슈퍼컴퓨팅본부 선임연구원
- 관심분야 : 컴퓨터그래픽스, 증강현실, 비디오기반 렌더링, 실시간 렌더링 등
- <https://orcid.org/0000-0001-6836-0166>



윤 승 현

- 2001 한양대학교 수학과 학사
- 2007 서울대학교 컴퓨터공학과 박사
- 2007 ~ 현재 동국대학교 멀티미디어공학과 조교수/부교수/교수
- 관심분야 : 컴퓨터그래픽스, 기하모델링, 가상/증강/혼합현실
- <https://orcid.org/0000-0002-0015-8305>