

공간 가상현실 메타버스내에서 IR-트래킹 시스템을 이용한 컨텐츠 설계

김태원*, 이상윤**, 김현웅*, 이대솔**, 이대식**

The Design of Content using IR-Tracking System in the Spatial Virtual Reality Metaverse

Tae-Won Kim*, Sang-Yoon Lee**, Hyun-Woong Kim**, Dae-Sol Lee*, Dae-Sik Lee*

요약 메타버스 내의 가상현실 공간 형성을 위해서는 정확한 트래킹 센서와 구현이 필요하다. 대부분 국가 기관 및 기업들은 바이브와 같은 자체 트래킹 센서와 베이스 스테이션을 사용하여 트래킹을 수행한다. 바이브 방식은 공간적 제약, 주변에 장애물이나 구조물이 있을 경우 트래킹에 문제가 발생할 수 있고, 공간 내에서 수용 가능한 인원도 제한된다. 본 논문에서는 공간 가상현실 메타버스를 자유롭게 트래킹 할 수 있도록 IR센서 트래킹 시스템을 이용하여 메타버스를 구성하고, 필요에 따라 바닥과 천장에 IR센서를 추가 배치하여 메타버스 공간을 유연하게 구성하여 콘텐츠를 실험하였다. 실험한 결과 바이브 방식에 비해 IR-트래킹 시스템 방식이 바이브 방식보다 약 12% 정도 추적 안정성이 높았다. 또한 바닥 추적 센서를 더 추가하면 추적 안정성이 더 높아 안정적으로 가상공간을 표현할 수 있다.

Abstract In order to create a virtual reality space in the metaverse, accurate tracking sensors and implementation are required. Most government agencies and corporations performs tracking to use their own tracking sensors and a base stations, such as VIVE. The VIVE method may have tracking problems due to spatial constraints and obstacles or nearby structures, and it is also that the number of people that can be accommodated within the space is also limited. In this paper, we designed and implemented metaverse using an IR sensor tracking system to freely track the spatial virtual reality metaverse, and we experimented contents in metaverse by placing IR sensors additionally on the floor and ceiling if needed in order to flexibly configure the metaverse space. As a result of the experiment, the tracking stability of the IR-tracking system was approximately 12% higher than the VIVE method. Additionally, adding more tracking sensors on the floor increases tracking stability and allows for a stable representation of virtual space.

Key Words : Metaverse, Contents, Virtual Reality, Tracker Device, Tracking

1. 서론

메타버스는 컴퓨터와 콘솔게임으로 모니터를 보며 즐기던 2차원 게임 방식에서 3차원 체험형 가상현실, 증강현실, 혼합현실, 확장현실로 형태

가 급속도로 진화 중이다. 이전 단순히 엔터테인먼트 분야에 국한되지 않고 일선 기업과 산업 현장에도 적용되어 메타버스를 이용해 설계와 공정 작업 등 현장에서 보다 입체적이고 정밀한 작업을 수행할 수 있게 되었다[1, 2].

This paper was written with support from the Chungcheongbuk-do Provincial Council (research to build a regional-specific metaverse service platform).

* Chungbuk Provincial University

** Tricomtek co., Ltd, Research & Development Center

Received September 13, 2023

Revised September 21, 2023

Accepted October 06, 2023

가상현실(Virtual Reality)과 실생활이 결합된 메타버스 환경에서 가장 널리 사용되는 기술 중 하나는 가상현실이다. 바이브와 오쿨러스와 같은 대중적인 가상현실 디바이스를 통해 우리는 가상 공간에서 다양한 콘텐츠를 체험할 수 있다. 메타버스 내에서는 가상 공간 형태로 박물관이나 가상 영화관과 같은 콘텐츠를 즐길 수 있다[3, 4].

메타버스 내의 가상현실 공간 형성을 위해서는 정확한 트래킹 센서와 구현이 필요하다. 대부분 국가 기관 및 기업들은 바이브와 같은 자체 트래킹 센서와 베이스 스테이션을 사용하여 트래킹을 수행한다[5]. 또한 바이브 트래커는 다양한 물리적 객체나 액세서리에 부착하여 그 움직임을 실시간으로 추적하고, 가상현실 공간 내에서 해당 객체를 제어할 수 있어 가상현실이나 증강현실(Augmented Reality)에 사용된다[6]. 그러나 바이브 방식은 트래킹 센서의 크기와 베이스 스테이션 주변의 공간 요구 사항으로 인해 제한이 있다. 최대 15m x 15m의 공간만을 지원하며, 주변에 장애물이나 구조물이 있을 경우 트래킹에 문제가 발생할 수 있고, 공간 내에서 수용 가능한 인원도 제한된다.

따라서 공간 가상현실 메타버스 내에 자유롭게 트래킹을 구현할 수 있고, 필요에 따라 바닥과 천장에 센서를 추가 배치하여 공간을 유연하게 구성할 수 있는 공간 가상현실 메타버스내의 IR 트래킹 센서를 이용해서 콘텐츠 시스템 설계 및 실험을 한다.

2. 관련연구

인터랙티브 미디어의 특성은 활동적이고 체험적인 교육을 조성하는데 적합하며 기존의 단순한 교육 방식보다 높은 학습 효율을 기대할 수 있다. 하지만 인터랙티브 미디어를 활용하기 위해서는 다양한 장치와 소프트웨어가 필요한데, 이러한 장치들은 설치와 운영이 까다롭거나 필요한 기능을 제공하지 못한다는 단점이 있다. 따라서 라이다 센서를 활용한 교육용 인터랙티브 콘텐츠 개발에서는 유니티와 라이더 센서를 연동하여 콘

텐츠 운용 환경을 구축하고 실제 운용할 교육 콘텐츠를 제작한다. 동작 인식 장비로는 라이더 기반의 라이더 센서를 사용한다. 라이더 센서는 높은 인식률 때문에 사용자의 동작을 인식하기가 용이하지만 센서의 높은 인식률 때문에 발생하는 광범위한 노이즈 처리가 어렵다. 이를 해결하기 위해 센서의 인식 범위를 화면 크기와 동일하게 제한하고 유니티 엔진의 Raycast 기능을 활용해 센서의 노이즈를 축소하고 정밀도를 개선하여 자연스러운 인터랙션이 가능하다[7].

바이브 방식은 가상현실 및 증강현실 디바이스로 실제 세계의 물체나 사물을 가상 공간 내에서 추적하고 반영하는 기능을 제공한다[8]. 바이브 트래커는 라이트하우스(Lighthouse) 시스템과 함께 사용되며, 레이저를 사용하여 물체의 위치와 방향을 실시간으로 계산한다. 바이브 방식의 실제 시스템 구성은 그림 1과 같다.

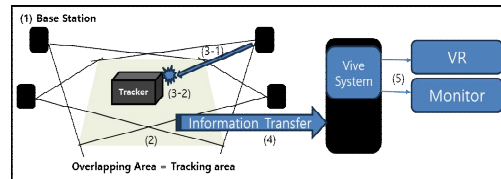
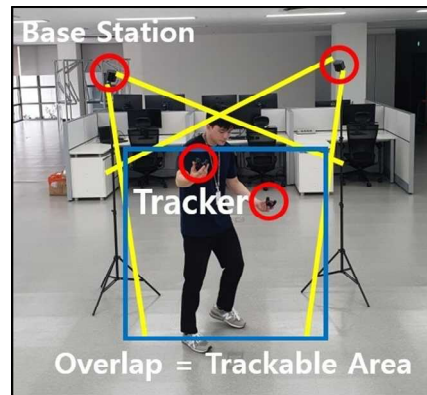


그림 1. 바이브 방식의 시스템 구성
Fig. 1. Vive-based system configuration

그림 1에서 보면 (1) 최소 2개 이상의 베이스 스테이션을 설치한다. 베이스 스테이션의 레이저 빔으로 주변공간을 (2)와 같이 스캔한다. 스캔된

주변공간을 트래킹 가능한 영역으로 구분하여 트래커가 추적한다. 베이스 스테이션에서는 (3-1)과 같이 다양한 시간 간격으로 레이저를 발생시키고, (3-2)와 같이 바이브 트래커 디바이스에는 레이저를 감지하는 센서가 내장되어 베이스 스테이션에서 발생한 레이저를 감지하여 디바이스의 위치와 방향 정보를 계산한다. 계산된 정보를 가지고 있는 디바이스가 (4)와 같이 바이브 시스템에 데이터를 전송한다.

바이브 시스템에서 데이터를 전송받으면 정보를 개선후 (5)와 같이 SteamVR에 연동되고, 연동된 데이터를 바탕으로 가상현실 및 증강현실 모니터에 전송된다. 바이브 방식 시스템은 고성능의 위치 및 움직임 추적 기능을 제공하지만, 최대 15m X 15m 으로 사용 지역이 제한되고, 베이스 스테이션 근처에 기둥 및 물건이 막고 있으면 정확한 트래킹이 되지 않는다.

따라서 공간 가상현실 메타버스 내에 자유롭게 트래킹을 구현할 수 있고, 필요에 따라 바닥과 천장에 센서를 추가 배치하여 공간을 유연하게 구성할 수 있는 공간 가상현실 메타버스내의 IR-트래킹 센서를 이용해서 콘텐츠 시스템 설계 및 실험한다.

3. IR-트래킹

IR-트래킹 작동 방식은 디바이스 내부의 센서와 레이저 기술을 활용하여 위치와 회전 정보를 감지하고 이를 컴퓨터로 전송하여 가상현실 및 증강현실 환경에서 사용자의 움직임을 추적하는 기능이다[9]. IR-트래킹 방식의 실제 시스템 구성은 그림 2와 같다.

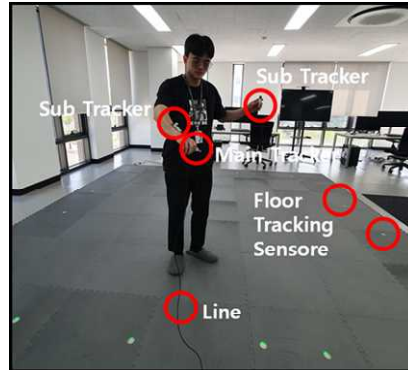
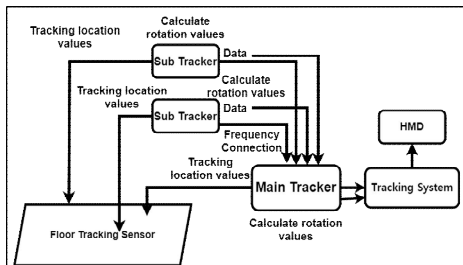


그림 2. IR-트래킹 방식의 시스템 구성
Fig. 2. IR-tracking system configuration

그림 2에서 보면 Main Tracker는 IR-트래킹 시스템에 유선 연결되고, Main Tracker와 Sub Tracker는 주파수를 통해 무선으로 연동된다. 각 Sub Tracker는 바닥추적 센서를 통해 위치값을 계산하고 자체적으로 회전값을 계산한다. 각 Sub Tracker에 의해 계산된 데이터는 Main Tracker를 통해 IR_Tracking System으로 데이터를 전달한다. 전달된 데이터로 콘텐츠 및 게임을 구현하고 HMD(Head Mounted Display) 장치로 전송하여 가상현실, 증강현실 등을 경험할 수 있다.

3.1 IR-트래킹 디바이스

IR-트래킹 시스템의 바닥추적 센서는 그림 3과 같다.

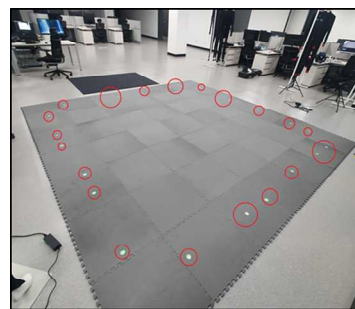


그림 3. 바닥추적 센서
Fig. 3. Floor tracking sensor

그림 3의 바닥추적 센서는 각 트래커들이 위치값을 계산하기 위해 필요한 센서이다. 바닥추적 센서의 파장은 850nm로 적외선을 무작위로 발생시켜 트래커의 센서에 적외선을 노출시켜 위치값을 계산하도록 설계되어있다.

Tracker는 USB Socket은 그림 4와 같다.



그림 4. USB Socket, Tag, Bracer 트래커
Fig. 4. USB Socket, Tag, Bracer Tracker

그림 4에서 보면 USB Socket 트래커는 Main Tracker로 Sub Tracker와 주파수로 연결하여 Sub Tracker의 데이터를 수집하여 IR-Tracking System을 통해 PC나 VR, 안드로이드 시스템으로 전송한다. Tag 트래커는 위치 회전값을 계산하여 Main Tracker로 전송한다. Bracer 트래커는 Tag와 마찬가지로 작고 가볍고 Main Tracker와 무선으로 연결되어 회전값과 위치값을 전송한다.

USB Socket, Tag, Bracer 트래커를 서로 무선으로 연결하여 그림 2를 언리얼 엔진에 형상화하면 그림 5와 같다.

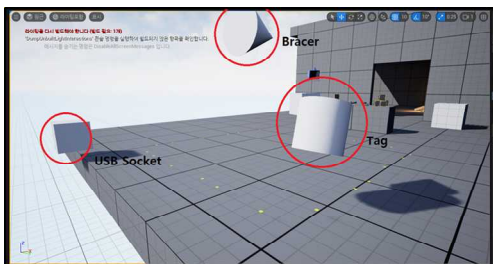


그림 5. IR-트래킹 시스템의 형상화
Fig. 5. Shape of IR-tracking system

그림 5에서 보면 Sub Tracker와 Main Tracker는 바닥추적 센서를 통해 위치값과 회전값을 자체 계산해 데이터화 한다. Sub Tracker

는 위치 회전 데이터를 Main Tracker인 USB Socket으로 보내고, USB Socket은 받은 데이터를 IR-트래킹 시스템으로 보낸다.

언리얼 엔진에서 IR_Tracking System의 바닥추적 센서값으로 엔진 내에 형상화하고, IR-트래킹 시스템에서 전달받은 각 트래커들의 데이터를 물건으로 전달해 트래킹 한다. 언리얼 엔진 내에 각 트래커들의 트래킹을 위한 순서는 그림 6과 같다.

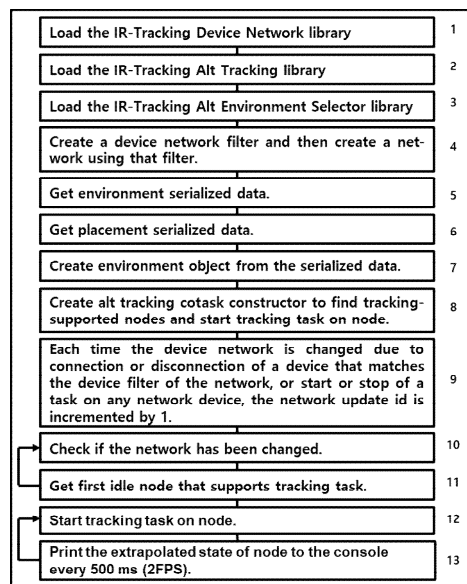


그림 6. 트래킹 순서도
Fig. 6. Tracking Flow Chart

그림 6에서 보면 단계 1, 2에서 IR-트래킹 시스템 디바이스 네트워크 라이브러리 접속하여 Main Tracker와 Sub Tracker의 연결을 확인하는 작업으로 Main Tracker의 주파수 값인 Sub Tracker를 찾아 연결 여부를 확인한다. 단계 3에서 IR-트래킹 시스템 바닥추적 센서 라이브러리 접속하여 바닥추적 센서값을 입력하고, 가상공간에 트래킹 데이터를 위한 작업을 한다. 단계 4에는 연결되어 있는 트래커의 장치에 연결 필터를 만든 후, 고유의 네트워크 생성하여 처음 작업에서 Main Tracker와 Sub Tracker 연동

이 확인되면 Main Tracker와 Sub Tracker를 묶어 데이터의 흐름을 만든다. 단계 5에는 바닥 추적 센서 라이브러리에서 데이터를 전달받고, 단계 6에는 IR-트래킹 시스템의 Tracker 라이브러리 데이터를 전달받는다. 단계 7에는 바닥추적 센서의 환경에 맞는 오브젝트를 생성하여 단계 6의 데이터를 바탕으로 가상 바닥을 생성한다. 단계 8에는 트랙커의 디바이스와 시스템간 추적작업을 원활하게 해주는 시스템을 생성한다. 단계 9에는 네트워크가 끊기는 경우 Main Tracker와 Sub Tracker가 연결된 데이터 흐름을 다시 연결한다. 단계 10, 11에는 단계 9를 확인 후 추적하는 시스템이고, 단계 12, 13에는 트랙커의 데이터를 전달받아 모니터에 출력되고, 언리얼 엔진 내에 형상화된다.

4. 제안한 콘텐츠 설계

현재 국내에는 바이브 방식 시스템을 사용하여 공간 가상현실 콘텐츠를 설계하고 있다. 본 논문에서는 공간 가상현실 메타버스 내에 자유롭게 트래킹을 구현할 수 있고, 필요에 따라 바닥과 천장에 센서를 추가 배치하여 공간을 유연하게 구성할 수 있는 공간 가상현실 메타버스 내에서 IR-트래킹 방식을 이용해서 콘텐츠 시스템 설계하였다. 제안한 콘텐츠 설계 코드를 구현하면 표 1과 같다.

표 1. 제안한 콘텐츠 설계 코드
Table 1. Proposed content design code

```

void TrackerceneComponent::BeginPlay()
{
    Super::BeginPlay();
    Init();
}

void TrackerceneComponent::EndPlay()
{
    Super::EndPlay();
    StopTrackingTask(); }

void TrackerceneComponent::TickComponent(float DeltaTime, ELevelTick TickType,
    FActorComponentTickFunction* ThisTickFunction)
{
    Super::TickComponent(DeltaTime, TickType, This
        TickFunction);
}
    
```

```

.....
if (IsAltEnvironmentValid(pEnvironment))
{
    if (IsNetworkVaild(pDeviceNetwork))
    {
        ApplyTrackingData(); }
    else
    {
        if (InitDeviceNetwork())
            OnDeviceNetworkChanged(); }
    else
    {
        UE_Log(LogTemp, Error, TEST
            ("Failed to find DeviceNetwork"));
        } }
}
else
{
    if (InitAltEnvironment())
    {
        if (InitDeviceNetwork())
        {
            OnDeviceNetworkChanged(); }
        else
        {
            UE_Log(LogTemp, Error, TEST
                ("Failed to find DeviceNetwork"));
            } }
    }
    else
    {
        UE_Log(LogTemp, Error, TEST
            ("Failed to find Alt Environment Actor"));
        } }
}
}
}
    
```

표 1에서 보면 void TrackerceneComponent::BeginPlay()는 시작시 실행되는 함수이다. Init() 함수는 IR Tracking System에 설정한 트랙커들의 데이터와 바닥추적 센서 데이터들을 입력받아 셋팅하는 함수를 구현하였다. 표 1의 Init()의 함수 구현 코드는 표 2와 같다.

표 2. Init()의 함수 구현 코드
Table 2. Init() function implementation code

```

void TrackerceneComponent::Init()
{
    GetTrackingLibrary(pAltTrackingLibrary);
    CreateTrackingCotaskConstructor(TrackingCotaskConstructor);
    GetPlacement(pPlacement); }
    
```

표 2에서 보면 GetTrackingLibrary(pAltTracking Library)는 IR 트래킹 저장소에 데이터를 전달받는다. CreateTrackingCotaskConstructor(TrackingCotaskConstructor)는 IR 트래킹에서 연결되어있는 트랙커의 장치 연결 필터를 만든 후, 고유의 네트워크 생성하여 처음 작업에서 Main Tracker와 Sub Tracker 연동이 확인

되면 Main Tracker와 Sub Tracker가 연결된 데이터의 흐름을 만들고, 언리얼 공간에 물체를 형성할 준비가 완료되는 코드를 구현한다. GetPlacement(pPlacement)는 언리얼에 적용되는 고유의 데이터값을 언리얼 데이터에 최적화하는 코드를 구현한다. 언리얼 엔진에 적용한 코드 구현은 표 3과 같다

표 3. 언리얼 엔진에 적용한 코드
Table 3. Unreal engine applied code

```
void TrackerSceneComponent::ApplyTrackingData()
{
    if (IsTaskValid())
    {
        if (IsTaskFinished(pTrackingCotask))
        {
            // ...
        }
        else
        {
            GetExtrapolatedState(pTrackingCotask, pPlacement, fExtrapolationTime);
            SetRelativeTransform(FVector(pTrackingState.AltPose.Location), FVector(pTrackingState.AltPose.Rotator), FVector(1.f, 1.f, 1.f));
        }
    }
}
```

표 3에서 보면 처음 바닥추적 센서 데이터 여부를 확인하여 데이터가 존재하면 바로 디바이스 네트워크 데이터 확인한다. 데이터가 없다면 IR-트래킹 시스템에 접속하여 바닥추적 센서 데이터를 가져온 후 디바이스 네트워크 데이터 여부를 확인한다. 디바이스 네트워크 데이터가 존재하면 ApplyTrackingData() 함수를 실행해 Tracking을 실시하고, 네트워크 데이터가 존재하지 않으면 IR-트래킹 시스템에 들어가 디바이스 네트워크 데이터를 전달받은 다음 ApplyTrackingData()를 구현하는 코드이다.

설계된 코드로 공간 가상현실 메타버스내에서 IR-센서 트래킹을 이용해서 구현하면 그림 7과 같다.

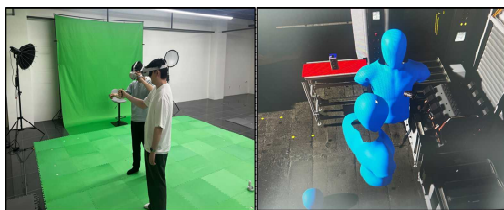


그림 7. 제안한 콘텐츠 구현결과
Fig. 7. Proposed content implementation result

그림 7에서 보면 현재 IR 트래커를 VR에 착용후 차량 엔진을 점검하는 콘텐츠를 체험하고 있다.

5. 실험

본 논문에서는 바이브 방식과 IR-트래킹 시스템 방식을 콘텐츠로 구현하여 0.5초당 트래커의 추적 안정성 비교하여 실험하였다.

그림 1과 같이 바이브 방식을 이용해서 실험한 데이터는 표 4와 같다.

표 4. 바이브 방식의 실험 데이터
Table 4. Vibe-based experimental data

Time	Location	Rotation	Stability : Value
0.5	x:0.00000, y:4.45821, z:0.12253	x:0.00154, y:0.02541, z:1.51242	0.482731
1	x:0.03200, y:6.93021, z:0.42353	x:0.97543, y:1.12871, z:1.05623	0.672854
1.5	x:1.23456, y:5.1245, z:2.65431	x:1.43210, y:1.84521, z:2.15105	0.537291
2	x:-0.56789, y:4.45821, z:1.78903	x:2.87654, y:2.15741, z:2.71132	0.604819
2.5	x:2.34567, y:3.98762, z:1.54321	x:0.78901, y:1.97141, z:2.00541	0.703258
3	x:2.98451, y:2.98761, z:0.98762	x1.11548, y:2.41812, z:1.78435	0.715924
3.5	x:2.98765, y:-0.76543, z:1.6651	x:2.34567, y:3.00541, z:1.48523	0.564937
4	x:1.25431, y:0.46181, z:1.00584	x:2.34567, y:3.54512, z:1.14552	0.509634

표 4에서 보면 Location은 컴퓨터에서 인식한 위치값, Rotation은 컴퓨터에서 인식한 회전값, Stability : Value은 추적 안정성이다. 실제 트래커가 얼마나 안정적으로 트래킹이 되고있는 상태는 1이 가까울수록 안정적이다.

그림 2와 같이 IR-트래킹 시스템을 이용해서 실험한 데이터는 표 5와 같다.

표 5. IR-트래킹 시스템의 실험 데이터
Table 5. IR-tracking system experimental data

Time	State : Pose(Position, Rotation)	Stability : Value	Velocity
0.5	x:0.000000 y:1.680000 z:0.000000, x:-0.001345 y:0.009046 z: -0.005506	0.601324	x:0.000000 y:0.000000 z:0.000000
1	x:-0.000002 y:1.680002 z:-0.000002,, x:-0.005743 y: 0.008723 z: -0.144282	0.745615	x:-0.000054 y:0.000075 z:-0.000080
1.5	x:0.000004 y:1.679990 z:-0.000001, x:-0.005939 y:0.008181 z:-0.144446	0.808124	x:0.000098 y:-0.000298 z:-0.000019
2	x:0.000004 y: 1.679996 z:-0.000005, x:-0.006231 y:0.007929 z:-0.144568	0.623451	x:0.000119 y:-0.000121 z:-0.000163
2.5	x:-0.004183 y:1.683111 z:-0.000464, x:0.189238 y:0.152570 z:-0.130263	0.764312	x:-0.124372 y:0.092495 z:-0.013553
3	x:0.000841 y:1.679429 z:-0.000379, x:-0.047929 y:-0.975596,z:0.007536	0.742513	x:0.026307 y:-0.017802 z:-0.011837
3.5	x:0.649337 y: 1.097101 z:2.106894, x:-0.257920 y:-0.865179 z:-0.246434	0.791345	x:0.261101 y:0.613299 z:-0.168417
4	x:0.677168 y:1.032515 z:1.893447, x:-0.146803 y:-0.810448 z:-0.079206	0.734216	x:-0.070687 y:-0.210563 z:-0.453211

표 5에서 보면 State : Pose(Position, Rotation)는 IR-트래킹 시스템에 측정된 위치 값, 회전값, Stability:Value는 추적 안정성, 실제 트래커가 얼마나 안정적으로 트래킹이 되고있는 상태는 1이 가까울수록 안정적이다. Velocity는 방향 속도값으로 0.5초전에 어느 방향에서 지금 방향으로 얼마큼 이동했는지 측정한다.

바이브 방식과 IR-트래킹 시스템 방식을 콘텐츠로 구현하여 0.5초당 트래킹 정보로 트래커의 추적 안정성 비교한 것은 표 3과 같다.

표 6. 바이브와 IR-트래킹 시스템의 비교 데이터
Table 6. Comparison data between Vive and IR-tracking systems

Time	IR Trackier Stability : Value	Vive Tracker Stabliity : Value	Difference Value
0.5	0.601324	0.482731	0.118593
1	0.745615	0.672854	0.072761

1.5	0.808124	0.537291	0.270833
2	0.623451	0.604819	0.018632
2.5	0.764312	0.703258	0.061054
3	0.742513	0.715924	0.026589
3.5	0.791345	0.564937	0.226408
4	0.734216	0.509634	0.224582

표 6을 그래프로 나타내면 그림 8과 같다.

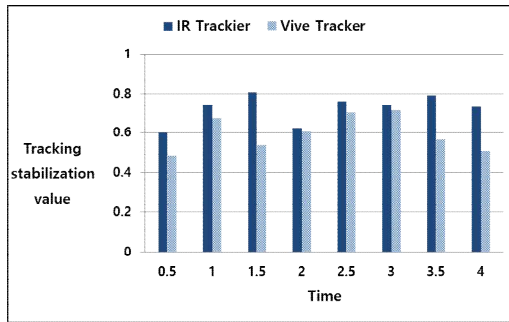


그림 8. 바이브와 IR-트래킹 시스템의 비교 그래프
Fig. 8. Comparison graph between Vive and IR-tracking systems

표 6과 그래프 8에서 보면 Stabliity : Value = 1에 가까울수록 100 퍼센트이다. 본 실험에서는 0.5초당 IR 트래커의 추적 안정성을 측정하기 위해 바이브 트래커와 IR 트래커의 추적 안정성 데이터로 추적 안정성 차이값인 Difference Value로 계산하였다. 계산한 결과 IR 트래커가 바이브 트래커보다 약 12% 정도 추적 안정성이 높았다. 또한 바닥추적 센서를 더 추가면 추적 안정성이 더 높아 안정적으로 가상공간을 표현할 수 있다.

6. 결론

메타버스 내의 가상현실 공간 형성을 위해서는 정확한 트래킹 센서와 구현이 필요하다. 지금 현재 사용 중인 바이브 방식은 트래킹 센서의 크기와 베이스 스테이션 주변의 공간 요구 사항으로 인해 제한이 있다. 최대 15m x 15m의 공간만을 지원하며, 주변에 장애물이나 구조물이 있을 경우 정확한 트래킹이 되지 않고, 공간 내에서

수용 가능한 인원도 제한된다.

본 논문에서는 방이브 방식의 문제점을 해결하기 위해 국내에서 상용화되지 않은 IR-트래킹 시스템을 구성하였다. IR-트래킹 시스템은 베이스 스테이션 없어 주변에 장애물이나 구조물이 있을 경우도 정확한 트래킹이 되었고, 공간이 부족해서 트래킹이 잘 되지 않을 때 바닥과 천장에 센서를 추가 배치하여 넓은 공간을 확보하여 많은 인원이 공간 가상현실 메타버스 내에 자유롭게 트래킹을 구현할 수 있다.

따라서 메타버스내의 IR-트래킹 시스템은 메타버스의 공간과 현실의 공간의 동기화 하여 다양한 분야에 적용이 가능하다. 또한 메타버스내에 관광지의 관람객의 위치 트래킹 및 동작 트래킹을 통한 콘텐츠 동기화 기술, 수몰지역 등의 갈 수 없는 지역의 콘텐츠에 대하여 관람객의 위치에 따른 동기화 콘텐츠제공 기술, 미래 도시 구축 콘텐츠 기술 등의 다양한 분야에 활용이 가능한 기술이다.

REFERENCES

[1] J. M. Son, S. H. Woo, J. H. Han, J. J. Kim, "Development and Application of Social Studies Content based on Augmented Virtual Metaverse: Jikji Simche Yojeol", Journal of Digital Contents Society, vol. 24, no. 6, pp. 1201-1207, 2023.

[2] H. S. Kwon, J. W. Joo, "Self-Expression Leading to the Virtual World: A Study on the Metaverse Platform with Selfie Content", Journal of Digital Contents Society, vol. 24, no. 8, pp. 1763-1771, 2023.

[3] S. G. Lee, "A Study on the Factors Influencing User Satisfaction of Virtual Reality-based Underground Simulation", Proceedings of the Information and Control Symposium of the Korean Electrical Society, vol. 24, no. 5, pp. 647-653, 2023.

[4] J. H. Park, J. H. Bu, K. S. Park, "Design and Development of an Immersive Virtual Reality Simulation for Environmental

Education", Journal of the Korea Institute Of Information and Communication Engineering, vol. 26, no. 4, pp. 541-547, 2022.

[5] Y. S. Lee, T. H. Lee, "Applying the Metaverse Platform and Contents in Practical Engineering Education", Journal of the Korea Computer Graphics Society, vol. 28, no. 3, pp. 31-43.

[6] H. J. Oh, "Contents Application Methods of Metaverse Platforms by Type Examples", Journal of Korea Society of Digital and Information Management, vol. 12, no. 6, pp. 2673-2684, 2021.

[7] D. J. Park, P. S. Shin, "Development of Educational Interactive contents using LiDAR Sensor", The Korean Society for Creative Information Culture, vol. 9, no. 2, pp. 55-65, 2023.

[8] Y. S. Lee, S. G. Song, "Analysis and Prospect of Virtual Streaming Service Using Metaverse Technology", The Journal of the Korea Contents Association), vol. 23, no. 1, pp. 324-333, 2023.

[9] J. C. Seo, D. K. Oh, J. H. Hong, "A Study on Real-time 3D Object Detection Using Augmented Reality Environment and HMD", The Korean Institute of Information Scientists and Engineers, vol. 2023, no. 6, pp. 739-741, 2023.

저자약력

김 태 원 (Tae-Won Kim)

[정회원]



- 2000년~현재 충북도립대학교 영상디자인과 교수
- 1995년 2월 국민대학교 시각디자인(학사)
- 2015년 11월 마들섹스대학교 인터랙티브미디어 디자인(석사)
- 2017년 8월 국민대학교 디지털 콘텐츠디자인(박사)

관심분야 : 메타버스 콘텐츠, 콘텐츠 디자인, AR/VR 어플리케이션

이 상 윤 (Sang-Yoon Lee) [정회원]



- 1999년 월~현재 트라이콤포텍(주) 엔지니어 총괄 수석부장
- 1996년 월~1999년 월 텔넷코리아 연구원
- 1995년 2월 충북대학교 통계학과(이학사)
- 2015년 8월 한국방송통신대학교 정보과학과(이학석사)

관심분야 : 무선/이동 전송, 차세대 이동통신, 메타버스 소프트웨어 및 어플리케이션, AR/VR 어플리케이션

김 현 용 (Hyun-Woong Kim) [정회원]



- 2019년 10월~현재 : 충북도립대학교 R&D 책임연구원
- 2002년 2월 동아대학교 산업디자인과(학사)
- 2005년 10월 브루넬대학교 디자인 전략과 혁신(석사)

관심분야 : 메타버스, 콘텐츠 디자인, 디자인경영

이 대 솔 (Dae-Sol Lee) [정회원]



- 2022년 12월~현재 : 트라이콤포텍(주) 연구원
- 2022년 2월 공주대학교 전기전자 제어공학(학사)

관심분야 : 메타버스 소프트웨어 및 어플리케이션, AR/VR 어플리케이션, 게임 개발, 반도체 소프트웨어

이 대 식 (Dae-Sik Lee) [정회원]



- 2011년 4월~현재 트라이콤포텍(주) 연구소장
- 1995년 2월 가톨릭관동대학교 전자계산공학과(공학사)
- 1999년 8월 가톨릭관동대학교 전자계산공학과(공학석사)
- 2004년 2월 가톨릭관동대학교 전자계산공학과(공학박사)

관심분야 : 이동통신시스템, AR/VR 어플리케이션