

# 실내 공간의 장애물 인식을 통한 AR 게임 공간 속 이동

## 가능 영역 검출

김기식<sup>○</sup>, 박종승  
인천대학교 컴퓨터공학과  
{kisik, jong}@inu.ac.kr

### Movable Area Detection in AR Game Space by Recognizing Obstacles in Indoor Space

Ki-Sik Kim<sup>○</sup>, Jong-Seung Park  
Dept. of Computer Science & Engineering, Incheon National University

#### 요 약

본 논문은 직육면체 형태의 실내 공간에서 이동 가능 영역을 검출하는 방법을 제안한다. AR 게임 공간 구성에 주로 활용하는 평면 검출 방식은 미완성된 기하 정보이므로 개체의 부피나 구체적인 영역 등을 파악할 수 없다. 제안 방법은 공간의 내부 표면 직육면체를 기반으로 유클리드 클러스터링을 통해 군집화를 수행하고, 클러스터의 간소화를 통해 장애물 영역을 검출한다. 제안 방법은 검출한 데이터를 기반으로 이동 가능 영역이 포함된 AR 게임 환경을 구성한다. 본 논문은 실험을 통해 제안 알고리즘을 통한 영역 검출 결과와 이를 기반으로 구성한 AR 게임 공간이 우수함을 보인다.

#### 1. 서 론

실내 Augmented Reality(AR) 게임 환경은 비전 기반의 알고리즘을 기반으로 한다. 비전 기반 알고리즘은 실내 공간을 분석하여 가상의 공간을 생성하고, AR 시스템은 가상세계를 비롯한 정보를 활용해 목표하는 콘텐츠를 구현한다. 그 후 두 세계는 합성을 통한 시점의 일치화를 통해 사용자에게 제공된다.

AR 게임을 위한 비전 기반 공간 인식은 일반적으로 이미지 자체보다는 이미지의 깊이 정보나 특징점 정보와 같이 1차적으로 가공된 정보를 획득하여 수행한다. 이를 위해서는 카메라의 위치를 추정하고 실시간으로 점군 데이터를 확장하는 Vision SLAM 알고리즘이 필요하다. 점군을 획득하기 위한 SLAM 알고리즘은 주로 PTAM[1]이나 ORB-SLAM[2]과 같은 특징점 기반의 간접하고

빠른 알고리즘을 활용한다.

AR 게임을 위한 공간 인식은 주로 평면 검출 알고리즘을 기반으로 한다. 이는 주로 Hough Transform, RANSAC 등의 알고리즘을 기반으로 하여 분할된 영역에 대한 원시 평면을 검출하고 주변으로 평면의 영역을 확장하는 Bottom-Up 방식을 기반으로 한다[3][4]. 이를 통해 획득한 평면 정보는 AR 게임 환경에서 활용도가 높지만, 평면 검출은 실내 공간에 존재하는 여러 장애물과 같은 개체의 영역을 파악할 수 없는 한계점이 있다. 평면 검출 알고리즘은 점군 속에서 평면성이 약한 영역이나 점군의 관측이 미흡한 영역에 대한 기하 정보를 검출할 수 없다. 따라서 평면 검출만으로는 일부 영역의 미완성된 기하 정보만을 검출한다. 이로 인해 개체의 부피나 개체가 공간에서 차지하는 구체적인 영역 등은 파악할 수 없다.

AR 게임 환경에 스폰된 캐릭터가 이동하거나

장소적 특성을 고려한 상호작용을 수행하기 위해서 게임 공간에 대한 이동 가능 영역 정보가 필요하다. 이동 가능 영역을 검출하기 위해서는 바닥면, 벽면, 천장 정보를 비롯해 공간에 존재하는 장애물의 영역을 파악해야 한다. 그러나 평면 검출만으로는 장애물의 영역 정보를 파악할 수 없다.

이동 가능 영역을 검출하기 위해서는 다른 접근 방법을 통한 장애물 영역 검출이 필요하다. 장애물의 영역 검출을 위해서는 점군을 입력으로 각 실세계의 객체에 의해 만들어진 샘플들을 분류해야 한다. 점군 군집화(Point Cloud Clustering)는 이에 적합한 기술이다[5]. 군집 분석의 유형은 크게 계층적 군집화(Hierarchical Clustering)와 분리형(비계층적) 군집화(Partitioning Clustering)로 나뉜다. 분리형 군집화는 사전에 군집의 수를 정해 주어 할당되도록 하는 방식으로, 가장 대표적인 알고리즘은 K-Means 알고리즘이 있다[6]. 그러나 분리형 군집화 방법은 가정하는 환경이 미지의 실내 공간일 경우 객체의 개수를 알 수 없으므로 군집의 수를 정할 수 없다. 따라서 우리는 n개의 샘플에서 출발하여 거리가 가까운 대상과 군집을 이루는 Bottom-Up 방식의 계층적 군집화를 기반으로 장애물의 영역을 파악할 수 있다.

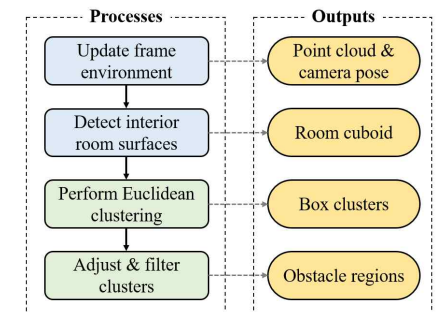
AR 게임 콘텐츠를 활용하는 대부분의 실내 공간은 직육면체 형태의 방이다. 따라서 직육면체 형태의 공간이 가지는 특성을 고려한 장애물 영역 검출은 대부분의 실내 공간에서 활용이 가능하다. 직육면체 형태의 방에서 존재하는 벽장, 테이블, 사물함과 같은 많은 개체는 공간의 벽면과 수직 또는 수평을 이루도록 배치되어 있다. 따라서 직육면체 공간의 내부 표면을 알 수 있다면 직육면체 공간과 좌표계가 같은 많은 개체의 영역을 비교적 정확하게 예측할 수 있다.

본 논문에서는 직육면체 형태의 실내 공간에서 장애물 영역 검출을 통해 이동 가능 영역을 검출하는 방법을 제안한다. 제안 방법은 직육면체 공간의 내부 표면 직육면체(바닥, 벽면, 천장의 6개 표면에 대한 평면 집합)를 검출하고, 이를 기반으로 직육면체 공간과 좌표계가 동일한 다중 개체 영역을 검출한다. 제안 방법은 계층적 유클리드 클러스

터링(Hierarchical Euclidean Clustering)[7]을 통해 점군을 군집화하고 간소화 과정을 통해 장애물 영역을 검출한다. 본 논문은 제안 방법을 적용한 AR 게임 환경을 구축하고, 장애물 영역의 우수성을 평가한다.

#### 2. 실내 공간의 장애물 영역 검출

이동 가능 영역을 검출하기 위해서는 바닥면, 벽면, 천장 정보를 비롯해 공간에 존재하는 장애물의 영역을 파악해야 한다. 제안하는 알고리즘은 매 프레임마다 새로운 스트리밍 이미지를 입력으로 하여 [그림 1]과 같은 파이프라인을 수행하며 장애물 영역을 검출한다. [그림 1]의 파이프라인 속 4개의 프로세스 블록은 각 프로세스를 수행하며 결과물 데이터를 출력한다. 그리고 출력된 데이터는 다음 프로세스의 수행에 입력 데이터로 활용된다. 파이프라인 속 파란색 블록들은 장애물 영역 검출을 위한 전처리 과정으로, 이미지 데이터를 입력으로 점군을 확장하고 이를 기반으로 공간의 내부 표면 직육면체를 검출하는 과정이다. 그 다음 수행되는 녹색 프로세스 블록들은 클러스터링을 통해 다중 개체 영역을 검출하고 이를 장애물 영역으로 가공하는 과정이다.



[그림 1] 제안하는 장애물 영역 검출 파이프라인

제안 알고리즘은 실시간으로 최대한 많은 점군 데이터를 획득하기 위해 OpenVSLAM[8]을 활용하였다. 해당 SLAM은 구면 파노라마 비디오를 입

력으로 하여 매 프레임마다 전 방향에 대한 다량의 점군 데이터와 카메라 위치를 출력한다. 제안 알고리즘은 이후 수행할 내부 표면 직육면체 획득을 위해 획득한 점군의 샘플 집합에 대해 점들을 연결하는 그래프인 KDTree를 형성하여 이웃점과 거리가 지나치게 먼 이상점들을 제거하여 1차적인 점군 가공을 수행한다.

제안 알고리즘은 실내 공간의 내부 표면 직육면체를 검출하고, 이를 기반으로 장애물 영역을 획득한다. 제안 알고리즘은 내부 표면 직육면체 검출을 위해 Kim 등[9]이 제안한 검출 방법을 활용한다. 해당 알고리즘은 직육면체 공간의 볼륨 검출을 통해 내부 표면 직육면체를 검출한다.

제안 알고리즘은 전처리 과정을 마친 후 실내 점군 데이터와 내부 표면 직육면체를 활용하여 장애물 영역을 검출한다. 본 논문은 해당 과정을 2.1장과 2.2장에서 자세히 설명한다.

## 2.1 점군의 군집화

제안 알고리즘은 계층적 유클리드 클러스터링을 기반으로 점군을 군집화하고, 군집들의 조정과 간소화 과정을 거쳐 장애물 영역을 확정한다. 이에 앞서서 실내 점군  $\mathbf{X}$ 에 대해 공간의 내부 표면 직육면체의 위치와 자세  $\mathbf{T}_b = [\mathbf{R}_b, \mathbf{t}_b]$ 을 변환행렬로 좌표계 변환을 수행한다(식 (1) 참조). 식 (1)의 함수  $f$ 은 앞서 설명한 이상점 제거 함수이다.

$$\mathbf{X}_i = \mathbf{T}_b^{-1} f(\mathbf{X}). \quad (1)$$

이상점 제거와 좌표계 변환을 마친 점군  $\mathbf{X}_i$ 을 획득한 후, 유클리드 클러스터링을 수행한다. 유클리드 클러스터링은 유클리드 거리 계산법을 통해 이웃점과의 거리가 임계값  $d$  이하에 해당하는 점들을 동일한 군집으로 분류하는 알고리즘이다.

여러 개체에 대한 군집을 지정할 때, 군집마다 샘플의 밀도차가 발생한다. 임계값  $d$ 가 상수일 경우 군집의 밀도가 고려되지 않기 때문에, 제안 알고리즘은  $d$ 를 가변 인자로 할당하였다. 클러스터링

알고리즘이 임의의 한 점을 기점으로 이웃점을 탐색할 때마다 식 (2), (3)와 같은 방법을 통해  $d$ 를 조정한다.

$$g(d_1, d_2, w) = (1 - w)d_1 + wd_2, \quad (2)$$

$$d = g(d_o, g(d, d_c, w_d), w_o). \quad (3)$$

식 (2)의 함수  $g$ 은 가중 이동 평균 필터(Weighted Moving Average Filter) 함수이다. 함수  $g$ 의  $d_1$ 은 기준 인자이며,  $d_2$ 은  $d_1$ 에게  $w$ 만큼 영향을 주는 갱신 인자이다. 제안 알고리즘에서  $d$ 은 조정을 위해 두 번의 가중 이동 평균 필터를 거친다(식 (3) 참조). 먼저,  $d$ 에 대해 직전 탐색에서의 이웃점 거리  $d_c$ 을  $w_d$ 만큼 반영하기 위해  $g(d, d_c, w_d)$ 을 수행한다. 이를 통해 최근 탐색 영역의 밀도에 적합한  $d$ 로 조정한다. 그 다음, 가장 초기에 설정한 거리 임계값 상수  $d_o$ 에 대해  $g(d, d_o, w_o)$ 의 결과값을 갱신 인자로  $w_o$ 만큼 반영하여  $g(d_o, d, w_o)$ 을 수행한다. 만약 이를 수행하지 않을 경우,  $d$ 가 지나치게 커지거나 작아져 예상범위 밖의 클러스터링 결과를 초래할 수 있다. 따라서 식 (3)과 같은 이중 필터를 통해  $d$ 가 초기에 설정한 임계값에서 지나치게 벗어나지 않도록 조정한다. 본 논문의 구현에서는 적합한 가변성을 고려하여  $w_d = 0.3$ ,  $w_o = 0.6$ 으로 가정하였다.

제안 알고리즘은 클러스터링을 통해 다중 샘플 군집을 획득한다. 획득한 다중 샘플 군집은 각 군집에 대한 Axis Aligned Bounding Box (AABB)를 획득하여 저장한다. 좌표계 변환을 거친  $\mathbf{X}_i$ 의 군집화를 통해 검출한 군집의 AABB은 공간의 내부 표면 직육면체와 자세가  $\mathbf{R}_b$ 로 동일하다. 따라서 벽면과 수직 또는 수평인 박스 형태가 된다.

## 2.2 군집 간소화를 통한 장애물 영역 검출

2.1장의 클러스터링 과정을 통해 획득한 다중 AABB은 식 (4)의 형태로 저장된다.

$$B = \{\mathbf{B}_0, \mathbf{B}_1 \dots\} \text{ where } \mathbf{B}_i = [\mathbf{a}_i, \mathbf{b}_i]^T \quad (4)$$

식 (4)의 각 원소  $\mathbf{B}_i$ 은 AABB의 8개의 모서리 중 위치의 값이 가장 작은  $\mathbf{a}_i$  지점과 가장 큰  $\mathbf{b}_i$  지점을 담고 있다.  $\mathbf{a}_i$ 와  $\mathbf{b}_i$ 을 기반으로 실내 공간의 일부 영역에 직육면체 형태를 표현할 수 있다.

$B$ 의 원소인 직육면체들은 일부 영역이 겹쳐질 수 있다. 만약 많은 영역이 겹쳐져 있어 한 직육면체가 영역을 표시할 의미가 없어진다면, 해당 직육면체를 제거한다. 만약 기준 직육면체가  $\mathbf{B}_i$ 이고, 비교할 직육면체가  $\mathbf{B}_k$ 라면,  $\mathbf{B}_i$ 의 클러스터를 생성할 때 당시의 거리 임계값이었던  $d_i$ 을 범위로 하여 박스의 경계 범위에  $d_i$ 을 더한 영역 안에 속한 직육면체를 제거한다.

간소화된  $B$ 의 직육면체들이 바닥에 밀착해 있을 수도 있지만, 바닥과 약간의 거리를 띄워 존재할 수도 있다. 그러나 본 논문에서 최종적으로 획득할 것은 이동 가능 영역이기 때문에, 바닥과 약간 떨어져 아래로 캐릭터가 다닐 수 없는 영역은 직육면체가 바닥과 밀착한 것으로 간주하였다.

간소화 과정을 마친 장애물 영역인 AABB은  $\mathbf{X}_i$ 을 입력으로 획득한 데이터로, 공간의 내부 표면 직육면체가 기준이 되도록 좌표계 변환을 한 상태에서 획득한 데이터이다. 그러므로 가상 월드 좌표계에 최종적인 장애물 영역으로 반영하기 위해  $\mathbf{B}'_i = \mathbf{T}_b \mathbf{B}_i$ 을 통해 월드 좌표계가 기준이 되도록 변환을 수행한다.

## 3. 게임 공간 구성과 이동 가능 영역 검출

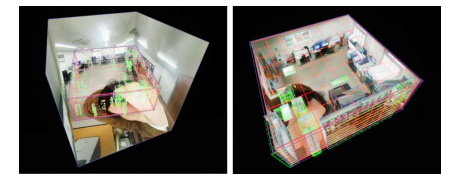
AR 게임 공간을 구성하기 위해서는 2장의 과정을 통해 획득한 출력물 데이터([그림 1] 참조)를 활용해 실세계 공간과 카메라 시점을 일치시키는 과정이 필요하다. 본 장에서는 가상세계를 구축하고 시점을 일치시키는 방법을 설명한다. 그 다음 시점 일치 방법을 기반으로 장애물 영역에 실세계가 투영된 물리적인 평면을 만들고 최종적으로 장

애플 영역을 획득하는 방법을 설명한다. 본 논문에서는 Unreal Engine 4 환경에서 AR 게임 공간을 구축하였다.

### 3.1 가상세계와 실세계의 시점 일치화

제안 알고리즘은 AR 게임 공간을 생성하기 위해 우선 2장에서 획득한 출력물 데이터를 배치한 가상세계를 생성한다. 그 다음 시점이 일치하는 실세계의 이미지를 스카이라이프스 형태로 3차원 객체의 텍스처 형태로 배치한다. 이를 통해 마치 실세계의 이미지에 가상의 출력물이 합성된 형태로 사용자에게 전달된다. 제안 알고리즘은 구면 파노라마 비디오표를 기반으로 하므로, 인간의 눈이 직관적으로 인지할 수 있는 핀홀 투영을 통해 시점을 일치시킨다.

제안 알고리즘은 가상세계에 배치한 투영 기하의 모든 지점들에 대해 투영 과정을 통해 텍스처를 맵핑한다. [그림 2](좌)와 같이 거대한 큐브맵 스카이라이프스 형태로 투영하여 배치하여도 올바른 합성 이미지를 제공할 수 있다. 그러나 이 경우 사용자의 카메라 시점에서 조금이라도 벗어나면 시점이 일치되지 않기 때문에 가상세계의 활용성에 많은 제약이 생긴다. 따라서 제안 알고리즘은 [그림 2](우)와 같이 내부 표면 직육면체와 장애물 영역의 다면 텍스처들에 구면 파노라마 이미지의 일부 영역들을 나누어 투영하는 방법을 통해 일치화된 합성 이미지를 완성한다.

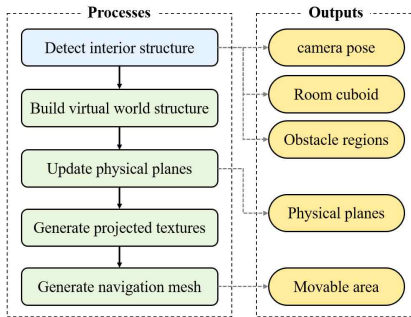


[그림 2] 투영 기하의 형태에 따른 차이: (좌)큐브맵 스카이라이프스 형태의 투영 기하, (우)제안 알고리즘에서의 투영 기하

### 3.2 가상 객체의 이동 가능 영역 생성

2장을 통해 획득한 실내 구조물 데이터(공간의 내부 표면과 장애물 영역 데이터)와 3.1장에서 설명한 투영 방법을 기반으로 본 논문에서 목표하는 최종적인 이동 가능 영역 정보를 포함한 AR 게임 공간을 구성할 수 있다.

이동 가능 영역을 포함한 AR 게임 공간 구성의 파이프라인은 [그림 3]과 같다. 그림 속 파란색 프로세스 블록은 2장의 실내 구조물 데이터 획득 과정을 의미한다. 해당 과정에서 OpenVSLAM을 통해 카메라 위치 및 자세 정보를 획득할 수 있으며, 이후의 과정으로 내부 표면 직육면체와 장애물 영역 데이터를 획득한다. 그림 속 4개의 녹색 프로세스 블록들은 게임 공간 구성 과정이다. 각 과정들은 미리 획득한 실내 구조물 데이터를 활용한다.



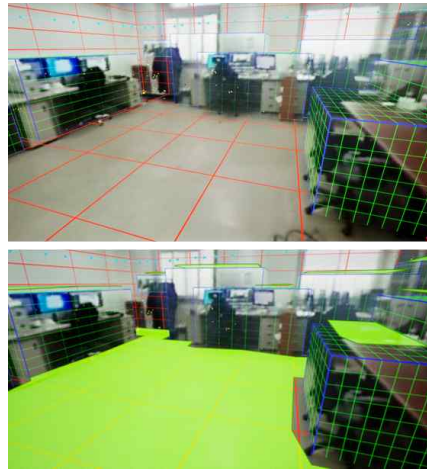
[그림 3] AR 게임 공간 구성 파이프라인

제안 알고리즘은 먼저 실내 구조물 데이터를 가상세계에 반영하여 가상세계 구조물을 구축한다. 이를 통해 가상세계에는 공간의 내부 표면과 장애물 영역이 시각화된다.

가상세계 구조물 구축을 마치고, 제안 알고리즘은 구조물의 각각 면에 대응하는 물리적인 평면들을 생성한다. 해당 평면은 가상세계에 존재하는 개체들에 충돌반응과 같은 물리작용을 발생시킨다.

제안 알고리즘은 물리적인 평면들에 대해 3.1장의 투영 방법을 통한 투영 텍스처들을 생성하여 맵핑한다([그림 4](위) 참조). 이 때, 다른 평면들에 가려져 사용자가 관측할 수 없는 평면은 투영 텍스처를 생성하지 않는다.

마지막으로, 제안 알고리즘은 내비게이션 메시지를 생성하여 가상세계에서의 이동 가능 영역을 검출한다. 이 과정은 Unreal Engine 4에 내장된 내비게이션 메시 생성 기능을 통해 수행된다. 이전의 과정에서 실내 구조물 데이터에 대해 물리작용이 가능한 다중 평면을 생성하였기 때문에 해당 과정에서 가상세계의 장애물 영역을 제외한 이동 가능 영역을 검출한다.



[그림 4] AR 게임 공간 구성의 단계적 결과:  
(위)실내 구조물에 다면 텍스처 투영, (아래)이동 가능 영역 검출

제안 알고리즘은 최종적으로 [그림 4](아래)와 같은 모습의 이동 가능 영역이 포함된 합성 이미지를 출력할 수 있다. 제안 알고리즘을 통해 획득한 실내 구조물 데이터와 이동 가능 영역을 활용한다면 높은 활용도의 AR 공간에 다양한 장작물을 배치하고 로직을 구현할 수 있을 것이다.

#### 4. 실험 결과

제안하는 알고리즘의 성능을 평가하기 위해 실험을 수행하였다. 실험은 검출된 장애물 영역의 성능을 평가하는 실험과, 구현된 AR 게임 환경의 품

질을 평가하는 과정으로 수행하였다.

##### 4.1 장애물 영역 검출 결과

검출된 장애물 영역의 정확도가 높을수록 이동 가능 영역 또한 정확하게 검출할 수 있다. 본 실험에서는 검출된 장애물 영역에 대해 Ground Truth와의 오차 범위를 그래프에 나타내 장애물 영역 검출 알고리즘의 우수성을 평가하였다.

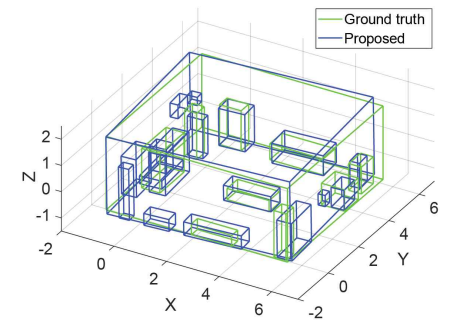


[그림 5] (위)실험을 위한 극 실사 환경,  
(아래)녹화한 구면 파노라마 비디오

실세계에서는 정확한 Ground Truth를 얻는 것이 어렵기 때문에 실험의 신뢰도가 하락한다. 이를 위해 Unreal Engine 4 상에서 실세계와 거의 유사한 극 실사 환경을 구성하여 입력으로 사용할 비디오를 녹화하였다. 제안 알고리즘은 OpenVSLAM을 기반으로 점군을 획득하기 때문에 이에 적합한 입력 데이터인 구면 파노라마 비디오를 녹화하였다. 비교를 위해 소파, 테이블, 책장과 같은 3차원 객체를 가상 실험 환경에 배치하고, 해당 객체의 Bounding Box의 크기를 측정하여 장애물 영역으로 기록하였다.

제안 알고리즘을 통해 획득한 장애물 영역을 m 단위로 Ground Truth와 비교한 결과는 [그림 6]과 같다. 제안 알고리즘은 실내 측정 대상에 해당하는 모든 장애물 영역의 검출에 성공하였다. 특히

비교적 거대한 객체에 대해 직관적으로 장애물의 영역을 인지할 수 있을 만큼 유사한 영역을 검출하였다.



[그림 6] 장애물 영역 검출 결과

Ground Truth의 개체 부피는 총  $5.84m^3$ 이었고, 제안 알고리즘을 통해 검출한 영역 중  $4.10m^3$  영역이 겹쳤다. 이는 Ground Truth 영역 중 70% 이상의 영역 검출에 성공한 것이다. 제안 알고리즘이 가정하는 전처리 영역인 내부표면 직육면체 검출 자체에 일부 오차가 있고, OpenVSLAM을 통해 획득할 수 있는 점군의 밀도가 낮다는 것을 고려한다면 해당 검출률은 매우 높아질 수 있다.

##### 4.2 구현된 AR 게임 환경 평가

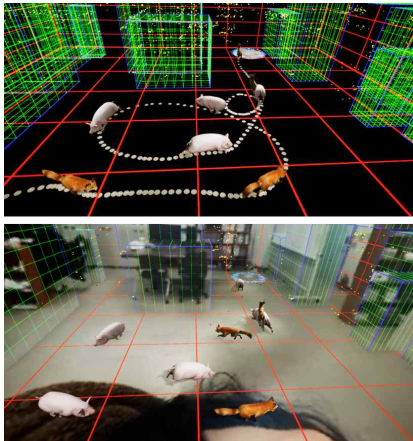
제안 알고리즘은 AR 게임 공간을 구성하기 위해 가상세계에 내부 표면 직육면체와 장애물 영역으로 이루어진 실내 구조물을 포함하였다. 그리고 구조물의 표면에 대해 물리적인 평면을 생성하고 다면 텍스처로 맵핑하여 높은 활용도를 가진 게임 공간을 구성하였다.

실내 구조물을 배치하고 물리적 평면을 만들어 투영하였기 때문에 발사체는 벽면에서 폭발하는 충돌 효과를 구현할 수 있었다. 총알을 멀리 있는 벽과 가까이 있는 벽에 따라 총알 비행시간과 같은 원근감을 잘 구현할 수 있었다.

구현된 AR 게임 공간은 [그림 7]과 같이 이동 가능 영역 안에서 적절한 움직임을 보이며 현실



세계와 동화된 느낌을 줄 수 있었다.



[그림 7] 가상 객체 스폰과 이동 가능 영역 테스트

바닥면을 포함한 모든 면이 물리적으로 생성되었으므로 Unreal Engine 4의 그림자 효과를 이용해 현실 세계에 실재할 때와 같이 자연스러운 그림자 효과를 줄 수 있었다. 광원 효과와 그림자 효과를 적용할 수 있기 때문에 가상세계에서 생성된 3차원 오브젝트가 마치 실재하는 것처럼 매우 높은 현실감을 가질 수 있었다.



[그림 8] 가상세계 합성 품질 테스트

## 5. 결론

본 논문에서는 직육면체 형태의 실내 공간에서

이동 가능 영역을 검출하는 방법을 제안하였다. 제안 알고리즘은 공간의 내부 표면 직육면체를 기반으로 하여 유클리드 클러스터링을 통한 군집화를 수행해 거의 모든 주요 장애물을 적절하게 검출하였다. 또한, 제안 알고리즘은 가상 구조물의 다면 텍스처 맵핑을 통해 활용도가 높은 AR 공간을 생성하였다.

본 논문의 제안 방법은 입력과 출력이 분명한 프레임워크이다. 따라서 SLAM의 점군 검출 성능을 향상시키거나 화질이 개선된 입력 비디오를 통해 더욱 높은 정확도로 장애물 영역을 검출할 가능성을 포함하고 있다. 제안 방법의 장애물 영역의 일부는 테이블 형태의 활용도가 높은 평면일 것이다. 향후 제안 방법을 통해 장애물 영역의 일부에 대해 평면 영역을 검출한다면 더욱 높은 활용도를 가지는 AR 게임 공간을 구성할 수 있을 것이다.

## ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2019R1F1A1060828).

## 참고문헌

- [1] Georg Klein, David Murray, "Parallel Tracking and Mapping for Small AR Workspaces", In Proc. 6th IEEE and ACM ISMAR, pp. 225-234, 2007.
- [2] R. Mur-Artal, J. M. M. Montiel, Juan D. Tardos, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System," IEEE Trans. Robot, Vol. 31, No. 5, pp. 1147 - 1163, 2015.
- [3] M. Hulik, M. Spanel, P. Smrz, Z. Mateme, "Continuous plane detection in point-cloud data based on 3D Hough Transform", J. Visual Commun. Image Represent., 2014.
- [4] Lin Li, Fan Yang, Haihong Zhu, Dalin Li, You Li, Lei Tang, "An Improved RANSAC for 3D Point Cloud Plane Segmentation Based on Normal Distribution Transformation Cells", Remote Sensing, 2017.

- [5] O. Mahamed, E. Andries, S. Ayed, "An overview of clustering methods" Intell. Data Analysis. Vol. 11, No. 6, pp. 583-605, 2007.
- [6] A. Likas, N. Vlassis, J. J. Verbeek, "The global k-means clustering algorithm", Pattern Recognition, Vol. 36, No. 2, pp.451-461, 2003.
- [7] R. B. Rusu, N. Blodow, Z. C. Marton and M. Beetz, "Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments", In Proc. Int. Conf. Intelligent Robots and Sys., 2009.
- [8] S. Sumikura, M. Shibuya, K. Sakurada, "OpenVSLAM: A Versatile Visual SLAM Framework", In Proc. 27th ACM Int. Conf. Mult., pp. 2292-2295, 2019.
- [9] K. S. Kim, J. S. Park, "Volume Detection from Indoor Spherical Panorama Point Cloud", In Proc. KIPS Spring Conf., Vol. 28, No. 1, pp. 560-563, 2021.