

Depth Camera를 이용한 VFH+ 기반 3차원 공간지도 작성

김형수*, 한영환**

VFH+ based 3D Spatial Map using Depth Camera

Hyung-Su Kim*, Young-Hwan Han**

요 약

본 논문에서는 멀티콥터에 Depth Camera를 부착하여 장애물을 회피하고 3차원 공간지도를 작성하였다. 3차원 공간지도를 작성하기 위해 멀티콥터에 Depth Camera를 부착하였으며 비행 중 장애물을 회피하는데 활용 가능성이 높은 제어 방법 중 VFH+(Vector Field Histogram) 알고리즘을 사용하였다. VFH+ 알고리즘은 2D 기반으로 동작하므로 3차원 공간지도 작성에 어려움이 있어 Octomap을 VFH+ 알고리즘에 접목하고 ROS 패키지의 Gazebo 시뮬레이션을 사용해 임의의 가상공간을 제작하여 비행한 후 작성된 공간지도와 비교하였다. 실험은 가상공간 A, B에 대하여 10회씩 반복하여 공간지도를 작성하였고 실험결과 공간 A는 9분 17초, 공간 B는 11분 46초의 평균 지도작성시간이 측정되었다.

Abstract

In this paper, a depth camera was attached to a multicopter to avoid obstacles and to create a three-dimensional(3D) spatial map. In order to create a 3D spatial map, a depth camera was attached to the multicopter, and the VFH+(Vector Field Histogram) algorithm was used among control methods that are highly likely to be used to avoid obstacles during flight. Since the VFH+ algorithm operates based on two-dimension(2D), it is difficult to create a 3D spatial map, so the Octomap was applied to the VFH+ algorithm, and an arbitrary virtual space was created using the Gazebo simulation of the ROS package and compared with the spatial map created after flying. The experiment was repeated 10 times for virtual spaces A and B to create a spatial map. As a result of the experiment, the average mapping time of space A was 9 minutes and 17 seconds, and space B was 11 minutes and 46 seconds.

Keywords

obstacle avoidance, vector field histogram, SLAM

* 상지대학교 정보통신소프트웨어공학과 박사과정
- ORCID: <https://orcid.org/0000-0002-0881-3191>
** 상지대학교 정보통신소프트웨어공학과 교수(교신저자)
- ORCID: <https://orcid.org/0000-0001-6890-0222>

• Received: May 03, 2021, Revised: Jul. 01, 2021, Accepted: Jul. 04, 2021
• Corresponding Author: Young-Hwan Han
Department of Information and Communication Eng., Sangji University
Wonju, Korea,
Tel.: +82-33-730-0485, Email: yhhan@sanji.ac.kr

I. 서 론

4차 산업혁명 시대에는 인공지능, 사물 인터넷, 빅 데이터, 모바일 등 첨단 정보통신기술이 사회 전반에 보급 및 사용되면서 사회의 아주 작은 부분까지 정보와 융합되고 있다. 특히 정보의 중요성이 대두됨에 따라 접근이 어려운 영역까지 정보를 수집하기 위한 다양한 방법들이 소개되고 있으며 정보 수집수단의 일환으로써 멀티콥터(드론)의 관심과 활용에 대한 연구가 활발히 진행되고 있다[1].

멀티콥터는 무인비행체로 사람이 탑승하지 않아도 비행할 수 있으며 원격 조종으로 사전에 입력된 프로그램에 의해 자동으로 비행하는 장치를 말한다. 국내법에서는 멀티콥터를 무인비행장치로 정의하고 있으며 무인기, UAV(Unmanned Aerial Vehicle System), UAS(Unmanned Aircraft System)라고 명명하는 경우도 있다. 멀티콥터는 항공, ICT, SW, 센서 등 첨단 기술이 융합된 사업으로 타 산업과 연계성이 아주 높으며 택시, 배송 등 응용 서비스 산업과의 융합 산업으로의 가능성이 매우 높다[2].

멀티콥터는 다양한 분야에서 연구가 이루어지고 있으며 특히 실내 재난 지역이나 탐사, 실내공간(Indoor)의 관리 등에 따라 공간지도 작성 분야의 대한 관심이 높아지고 있다. 실내공간에서의 안전한 비행하려면 자율비행이 필연적이다. 자율 비행이 가능하기 위해서는 위치 인식, 경로 탐색, 경로 추종, 장애물 회피, 충돌 회피와 같은 기술들을 필요로 한다. 이 기술들 중에서 장애물 회피는 멀티콥터의 안전한 비행을 보장하는 중요한 기술이다. 장애물을 인식하고 분석한 후 결과에 따라 진행 또는 정지하는 비교적 단순한 알고리즘부터 다양한 센서와 복잡한 프로그램 구조를 통한 아주 정교한 알고리즘에 대한 연구가 활발하게 진행되고 있다[3].

장애물을 회피하는데 활용 가능성이 높은 제어 방법 중에서 VFH+(Vector Field Histogram) 기반의 회피방법이 있다. VFH+은 거리 및 물체 탐지 센서인 Sonar 센서나 레이더 센서, 라이다 센서, 적외선 센서 등으로 특정 방향에 대한 거리 감지 정도를 획득하여 장애물에 대한 밀도를 나타내는 폴라 히스토그램을 생성하고 경계 값을 사용하여 회피방향을 결정한다. 이 방법은 실시간 동작과 구현이 용이

하지만 데이터의 원활한 처리를 위해 여러대의 거리를 측정할 수 있는 센서가 필요하다는 단점이 있다[3][4]. 다수의 센서가 필요한 단점을 보완하기 위해 단순한 카메라가 아닌 전방의 Depth 정보를 획득할 수 있는 카메라를 사용하여 단일 카메라로 문제점을 해결할 수 있다.

본 논문에서는 멀티콥터에 Depth Camera를 부착하여 장애물을 회피하면서 주변 공간을 3차원 공간 지도를 작성하였다. VFH+ 기반의 회피방법은 주변 환경 정보를 인식해 지도를 작성하기에 용이하다[5]. 그러나 2차원 기반의 평면형 지도작성만이 가능하기 때문에 정확한 실내지도공간을 작성하는데 어려움이 있다. 이러한 문제점을 해결하기 위해 Depth Camera로부터 획득한 3차원 정보(Pointcloud)를 기반으로 Octomap을 접목하여 3차원 공간지도를 작성할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 VFH+ 알고리즘에 대한 이론적인 내용을 설명하고, 3장에서는 Octomap을 기반으로한 3차원 공간지도 작성에 대해 설명한다. 4장에서는 Gazebo 시뮬레이션을 통한 3차원 공간지도 작성의 실험을 실시하고 5장에서는 결론과 고찰, 향후연구계획에 대해 설명한다.

II. 지도 작성을 위한 VFH+ 알고리즘

2.1 2D 1차 폴라 히스토그램

VFH+ 알고리즘은 2D 1차 폴라 히스토그램, 2D 2진 폴라 히스토그램, 마스크 폴라 히스토그램을 계산하는 과정을 거친다. 그 중 첫 번째가 2D 1차 폴라 히스토그램(Primary polar histogram)이다. Depth Camera로부터 획득한 정보를 2D 1차 폴라 히스토그램에 추가한다. 1차 폴라 히스토그램은 활성 복셀(Voxel)의 위치가 두 각도로 결정되는 극성 히스토그램이다[6]. 다음 그림 1은 2D 1차 폴라 히스토그램을 나타낸 것이다.

멀티콥터가 자율적으로 장애물을 피해 비행하려면 멀티콥터의 크기만큼 공간이 확보되어야 한다. 이때 탐색 공간에서의 장애물이 검출된다면 그 크기를 멀티콥터의 크기만큼 설정하고 멀티콥터는 하나의 점으로 인식한다[6].

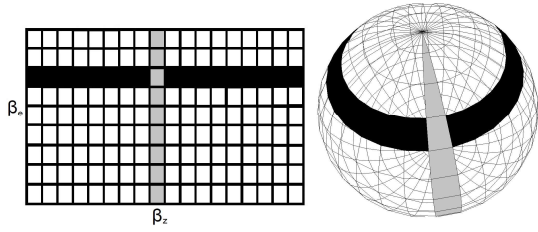


그림 1. 2D 폴라 히스토그램
Fig. 1. 2D polar histogram

다음 그림 2는 장애물의 경계를 늘렸을 때 장애물이 차지하는 각도를 나타낸 것이다.

장애물의 경계를 찾았을 때 장애물과의 각도 $\gamma_{i,j}$ 는 식 (1)과 같다.

$$\gamma_{i,j} = \arcsin\left(\frac{r_{r+s}}{d_{i,j}}\right) \quad (1)$$

1차 폴라 히스토그램은 식 (2)를 사용하여 각 영역에서 장애물이 영향을 미치는 범위를 계산하고 장애물 확률을 업데이트 한다.

$$H_k^p = \sum_{i,j \in C_a} m_{i,j} h'_{i,j} \quad (2)$$

여기서 $m_{i,j}$ 는 장애물 벡터의 크기를 나타내고 $h'_{i,j}$ 는 식 (3)을 이용하여 계산한다.

$$h'_{i,j} = \begin{cases} 1 & \text{if } \alpha \in [\beta_{i,j} - r_{i,j}, \beta_{i,j} + r_{i,j}] \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

α 는 한 구역에 대한 해상도를 의미하고 $\beta_{i,j}$ 는 장애물 벡터의 방향을 나타낸다.

2.2 2D 이진 폴라 히스토그램

2D 1차 폴라 히스토그램을 기반으로 이진 폴라 히스토그램을 계산하여 연산 정보를 줄인다. 2D 이진 폴라 히스토그램은 두 개의 임계값을 이용하여 계산하고 이전 단계인 2D 1차 폴라 히스토그램의 모든 셀에 대한 T_{low} 와 T_{high} 를 비교하며 수행한다. i 번째의 값이 T_{high} 의 값보다 높으면 2D 이진 폴라 히스토그램에서의 값은 1로 정의되고, i 번째의 값이 T_{low} 의 값보다 낮으면 2D 이진 폴라 히스토그램에서의 값은 0으로 정의된다. 정의된 값에 따라 각 구역이 열린 공간인지 장애물로 막힌 공간인지에 대한 가중치 H_{β_z, β_e}^b 를 계산한다[4]. i 번째에서 이진 폴라 히스토그램은 식 (4)와 같이 계산된다.

$$H_{\beta_z, \beta_e}^b = \begin{cases} 1 & \text{if } H_{\beta_z, \beta_e}^p > T_{\beta_e}^{high} \\ 0 & \text{if } H_{\beta_z, \beta_e}^p < T_{\beta_e}^{low} \\ H_{\beta_z-1, \beta_e}^p & \text{otherwise} \end{cases} \quad (4)$$

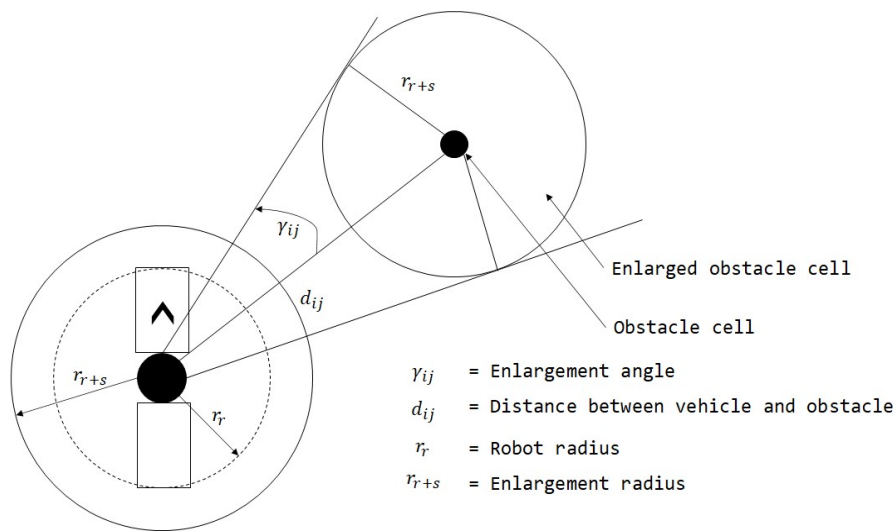


그림 2. 확대한 장애물 셀
Fig. 2. Enlarged obstacle cell

2.3 마스크 플라 히스토그램

멀티콥터가 전진하는 경우 좌우로 안전하게 회전하기 위한 최소 곡률 반경이 존재한다. 이 최소 곡률 반경과 장애물이 겹치게 되면, 장애물이 있는 위치로부터 멀티콥터의 뒤쪽까지 비행할 수 없다 [4][6]. 다음 그림 3은 멀티콥터가 비행할 때 차단된 길 찾기의 예를 나타낸 것이다.

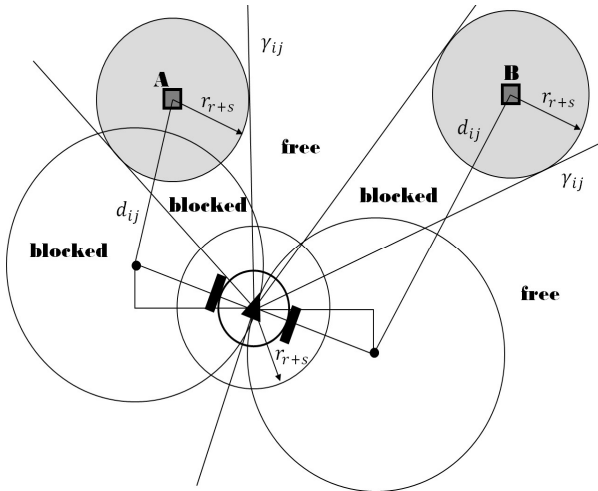


그림 3. 차단된 길 찾기의 예
Fig. 3. Example of blocked direction

위의 그림과 같이 멀티콥터의 비행시 장애물 A와 B가 있다고 가정할 때 장애물에 의해 차단된 물체를 확인할 수 있다. 각 장애물에 대해 멀티콥터의 너비를 고려하여 r_{r+s} 로 정한다. 만약 2개이상의 장애물의 원이 겹치면 장애물이 존재하는 위치로부터 장애물의 뒤쪽영역까지 진행할 수 없는 공간으로 정의한다. 이때 장애물 A쪽은 모든 방향이 block 처리되며 장애물 B방향쪽은 free로 처리된다[4][6].

2.4 경로 탐지 및 선택

경로 탐지 및 선택 단계에서는 2D 이진 플라 히스토그램에서 사용할 수 있는 경로를 검색하고 경로 가중치가 가장 작은 경로를 선택한다. 사용 가능한 경로를 결정하는 알고리즘은 2D 이진 플라 히스토그램 주위에서 창을 이동하여 2D 이진 플라 히스토그램에서의 빈 공간을 탐지하고 창의 모든 요소가 0이라면 이 창은 사용가능한 경로로 선택한다.

창을 구현하기 위해서는 히스토그램의 극성의 특성을 고려해야하며 창의 히스토그램의 경계를 넘으면 창은 2D 플라 히스토그램의 규칙에 의해 연결된 요소를 사용한다[7]. 다음 그림 4는 2D 플라 히스토그램에서의 창의 이동을 나타낸 것이다.

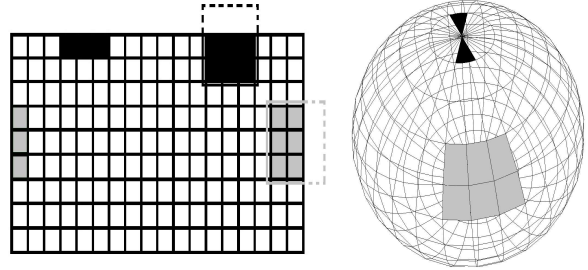


그림 4. 2D 플라히스토그램 이동창
Fig. 4. Moving window of 2D polar histogram

창의 히스토그램의 상한 또는 하한을 초과하면 위 그림 4의 2D 플라히스토그램의 이동창의 검은색 부분과 같이 확인해야하는 셀은 동일한 방위각에 배치된다. 창의 왼쪽 또는 오른쪽의 경계를 넘으면 히스토그램의 반대편 셀을 선택한다. 다음 세 가지 경로 가중치가 계산되고, 후보 방향의 하나의 경로 가중치에 결합된다. 식 (5)는 가중치의 계산식이다.

$$k_i = \mu_1 \Delta(v, k_i) + \mu_2 \Delta(v, \frac{\theta_i}{\alpha}) + \mu_3 \Delta(v, k_{i-1}) \quad (5)$$

k_i 는 목적지의 방향이며 $\frac{\theta_i}{\alpha}$ 는 현재 진행하고 있는 방향을 의미하고 k_{i-1} 은 이전에 선택된 진행 방향을 의미한다. μ_1, μ_2, μ_3 은 $\mu_1 > \mu_2 + \mu_3$ 의 조건을 만족해야 한다.

III. Octomap을 사용한 공간지도 작성

Octomap은 트리구조 Octree를 기반으로 하여 3차원 환경을 확률적으로 표현하는 공간지도 작성 기법이다. ROS 운영체제 호환되어 Pointcloud 값을 직관적으로 표현할 수 있다. 또한 트리구조의 특징으로 계층구조를 바탕으로 공간을 8등분해 나가면서 다중 해상도를 가능하게 한다. 복셀을 표현하는 수단으로 사용함으로써 3차원 공간 표현에 적합하다

[8]. Octomap 데이터 구조에서 탐색은 멀티콥터가 넓은 환경에서 비행할 때 주변 환경을 탐지하는데 필요한 계산적 성능의 한계 때문에 주변의 모든 물체(복셀)들을 탐지하는 것은 불가능하다. 따라서 멀티콥터가 진행하는 방향의 주변 경계 상자에 있는 복셀만 탐지한다. 이 경계 상자의 크기는 (너비, 높이, 깊이)로 정의할 수 있으며 이것은 VCP(Vehicle Center Point)라 한다. 그림 5는 Octomap에서의 Octree데이터 구조를 나타낸 것이다.

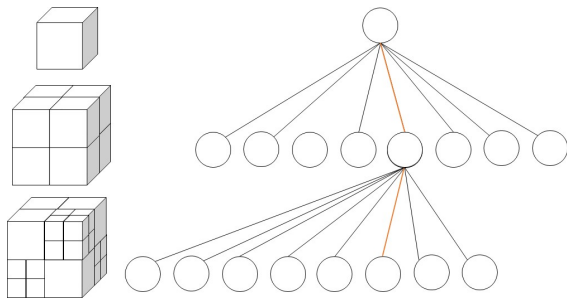


그림 5. Octomap octree 데이터 구조
Fig. 5. Data structure of octomap octree

VFH+만 사용하여 공간지도를 작성하면 2D 기반으로 작성이 되므로 Octomap을 사용하고 Depth Camera의 Pointcloud를 활용하여 3차원 공간을 인식한다. 그림 6은 같은 공간을 VFH+만 사용한 공간 지도와 Octomap을 추가하여 3차원 공간지도를 나타낸 것이다.

그림 6의 (a)는 2차원 공간지도를 작성한 것으로 검은색 부분은 장애물과 벽면으로 인식하여 멀티콥터가 비행할 수 없는 곳이다. (b)는 2차원 기반의 공간지도에 Octomap을 사용하여 3차원에서의 장애물과 벽면을 인식하여 표현한 것이다.

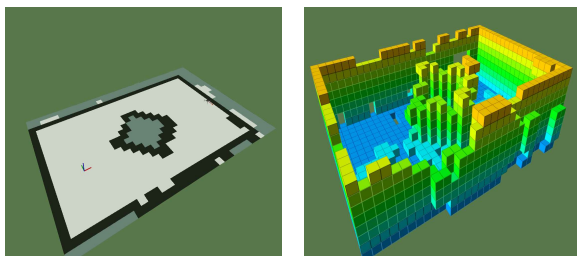


그림 6. (a) 2D 공간지도 작성 (b) 3D 공간지도 작성
Fig. 6. (a) 2D spatial mapping (b) 3D spatial mapping

IV. 실험

본 논문에서 실험을 수행하는 방법은 ROS(로봇 운영체제: Robot Operating System)를 사용한다. ROS는 Tool을 제공하는데 Gazebo(가제보)는 모델의 설계와 가상공간(world)에 대한 객체(object)를 모듈화하여 실험 환경을 3차원으로 구축할 수 있다. Gazebo 가상공간에서 멀티콥터를 동작시켜 장애물(벽면 또는 바닥)들을 회피하면서 3차원 공간지도를 작성하는 실험을 실시하였다. 표 1은 시물레이션에서 사용한 매개변수를 나타낸 것이다.

표 1. 시물레이션 파라미터
Table 1. Simulation parameter

Parameter	Value
Depth field of view(FOV)	86° × 57°
Range of depth	4.0m
Cell resolution	0.5m
Multicopter radius	0.5m
Multicopter altitude	1.0 to 3.0m
Multicopter speed	2.0m/s

실험에서 멀티콥터의 Z축 고도 값은 1m~3m로 비행하였으며 장애물의 탐지 거리는 시물레이션에서 제공하는 Depth Camera인 Intel Realsense D435i 모델을 사용하여 전방 4m의 물체를 탐지할 수 있다. 비행 중 멀티콥터가 인식하는 장애물은 Octomap에 의하여 3차원 복셀로 인식되고 VFH+ 알고리즘의 히스토그램 식에 따라 장애물을 인식되며 Octomap의 Cell 해상도는 0.5m로 설정하였다. 알고리즘의 히스토그램 식에 영향을 미치는 매개변수는 크기, 고도, 속도이다. 크기는 1m로 설정하였고, 고도값은 최소 1m에서 3미터까지 X, Y축(바닥)에서의 장애물을 탐지하였을 때 가변적으로 고도값을 변경할 수 있게 하였으며, 속도는 초당 2m를 비행할 수 있게 하였다. 그림 7은 Gazebo 가상공간에서의 공간지도 작성 과정을 나타낸 것이다.

그림 7과 같이 A, B, C, D는 Gazebo에서의 장애물을 탐지하면서 경로를 설정하고 A → B → C → D의 순서로 공간지도 작성 과정을 나타내었으며 탐류의 형태로 실험하였다.

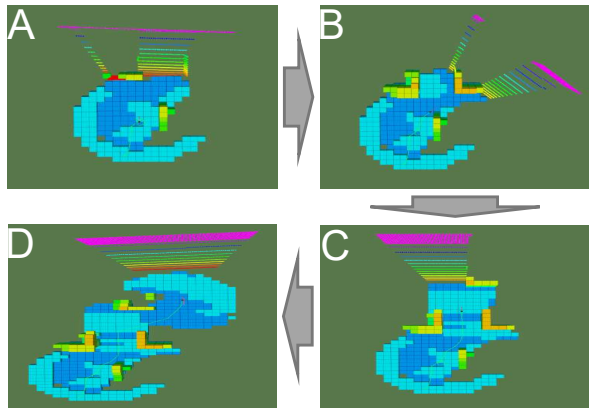


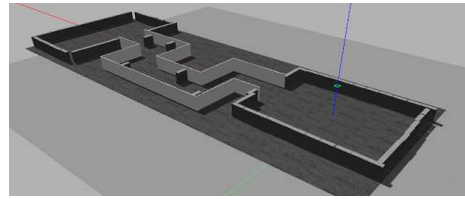
그림 7. 공간지도 작성과정
Fig. 7. Spatial mapping process

멀티콥터가 비행할 때 진행 방향에 설치된 Depth Camera의 Pointcloud 정보는 RGB 색온도로 표현하였으며 가까운 거리 빨간색을 기준으로 거리가 멀어짐에 따라 색상이 변화한다. 공간의 표현은 Octomap을 사용하여 X, Y축의 지면과 Z축의 벽면, 장애물을 표현하였다.

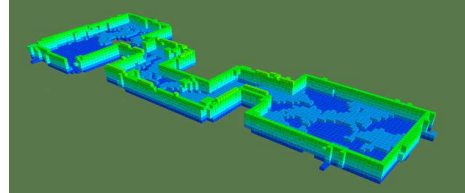
본 논문에서는 Gazebo를 사용하여 가상공간을 제작하였다. 가상공간의 환경은 실내환경에 대한 공간지도 작성을 하기 위해 건물 내부 형태로 구성하였다. 건물구조의 특징에 맞게 바닥면이 평평한 구조로 설정하고 벽면은 4m의 높이로 제작하였다.

실험 방법은 제작한 Gazebo 공간을 반복적으로 공간지도 작성을 수행하여 3차원 지도를 제작하였다. 각 가상공간마다 10회씩 공간지도 작성을 실시하였으며 지도작성 시간을 기록하였다. 그림 8은 Gazebo에서 제작한 가상공간 A, B에 대하여 10회씩 지도작성을 수행하고 완료한 결과 중 하나를 나타낸 것이다.

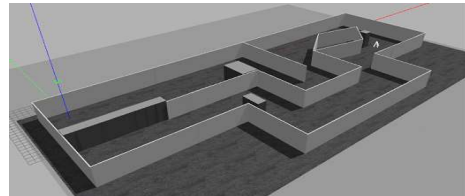
그림 8과 같이 실제 알고리즘이 적용된 멀티콥터가 비행한 후 3차원 가상공간을 표현한 것으로 공간지도 작성을 완료하였다. (a)과 (c)는 Gazebo를 이용하여 설계한 가상 공간을 나타낸 것이다. 청색 부분은 주로 바닥으로 작성되었으며 Z축의 고도에 따라 녹색으로 벽면과 장애물이 작성되었다. (b)의 구조는 가로 110m, 세로 40m이고 (d)의 구조는 가로 100m, 세로 40m이다. 두 가상공간은 크기는 비슷하지만 내부 구조 차이가 있다. 멀티콥터의 속도와 고도, Depth 탐지거리에 따라 지도작성시간에 영향을 주는 것으로 나타났다.



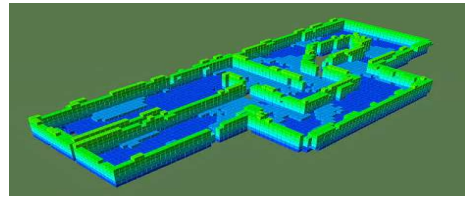
(a) Gazebo에서의 공간 A
(a) Space A in Gazebo



(b) A 공간지도 작성결과
(b) Spatial mapping results



(c) Gazebo에서의 공간 B
(c) Space B in Gazebo



(d) B 공간지도 작성결과
(d) Spatial mapping results

그림 8. 실험 결과
Fig. 8. Experimental results

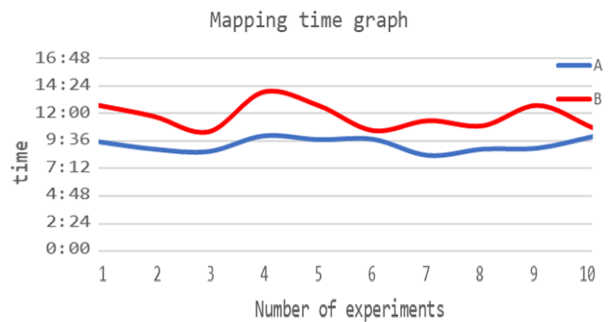


그림 9. 지도작성시간
Fig. 9. Mapping time graph

그림 9는 가상공간 A, B를 각각 10회씩 공간지도 작성 실험을 실시하고 소요시간을 기록한 것을 나타낸 것이다.

그림 9와 같이 가상공간 A, B는 비슷한 크기를 가지는 공간이지만 내부 구조에 따라 지도작성 시간이 다른 것으로 나타났다. 지도작성 시간은 동일한 공간을 탐색하더라도 Depth 정보가 동일하게 측정되지 않고 이로 인해 실험마다 동일한 경로로 비행하지 않기 때문에 지도작성 시간에 차이가 있다.

공간 A는 평균 9분 17초가 공간 B는 평균 11분 46초의 지도작성 시간이 측정되었다. 또한 가상공간 A, B에 대한 10회의 공간지도 작성 결과는 그림 8과 유사한 결과를 얻을 수 있었다.

V. 결 론

본 논문에서는 Depth Camera를 부착하여 3차원 공간지도를 작성할 수 있는 멀티콥터를 개발하였다. 기존의 VFH+ 알고리즘을 기반으로한 장애물 회피 기술과 3차원의 공간을 표현할 수 있는 Octomap을 사용하여 VFH+과 Octomap을 접목한 3차원 공간지도 작성 방법을 제안한다. 또한 ROS패키지의 Gazebo 시뮬레이터를 사용하여 가상의 공간을 제작하고 일정한 부분에 장애물을 설치하였다. 가상공간은 건물의 실내구조의 특징을 가지며 바닥면은 건물이라는 특성에 의해 평면으로 제작하였다. 실험결과 바닥과 벽을 포함한 장애물들을 3D 공간으로 표현하여 공간지도를 작성하였으며 멀티콥터를 이용한 실내 지도 제작 부분의 연구 발전이 높아질 것으로 기대된다.

향후 연구 계획으로는 Depth Camera 기반으로 동작하는 VFH+ 알고리즘을 Lidar 센서까지 활용할 수 있도록 연구할 계획이다.

References

[1] S. H. Park, Y. J. H. Go, J. Ryi, and J. S. Choi, "Prediction and Verification of Hover Performance through Multi-Copter Propulsion System Test Results", *Journal of the Korean Society for Aeronautical & Space Sciences*, Vol. 46, No. 7, pp. 527-534, Jul. 2018. <https://doi.org/10.5139/JKSAS.2018.46.7.527>.

[2] D. U. Kim, J. H. Kim, S. M. Kim, and K. B. Kwon, "Analysis and Implication on the International Regulations related to Unmanned Aircraft-with emphasis on ICAO, USA, Germany, Australia", *The Korean Journal of Air & Space Law and Policy*, Vol. 32, No. 1, pp. 225-285, Jun. 2017.

[3] Falanga, Davide, Kevin Kleber, and Davide Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras", *Science Robotics*, Vol. 5, No. 40, pp. 1-15, Mar. 2020. <http://dx.doi.org/10.1126/scirobotics.aaz9712>.

[4] Pappas, Pantelis, et al. "VFH+ based shared control for remotely operated mobile robots", 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics. IEEE, Abu Dhabi, United Arab, pp. 366-373, Nov. 2020. <https://doi.org/10.1109/SSRR50563.2020.9292585>.

[5] Kong Y. and W. C. Lee, "Dynamic obstacle avoidance and optimal path finding algorithm for mobile robot using Q-learning", *Journal of Korean Institute of Information Technology*, Vol. 15, No. 9, pp. 52-62, Sep. 2017. <http://dx.doi.org/10.14801/jkiit.2017.15.9.57>.

[6] Simon Vanneste, Ben Bellekens, and Maarten Weyn, "3DVFH+: Real-Time Three-Dimensional Obstacle Avoidance Using an Octomap", *MORSE@ STAF*, Vol. 1319, pp. 91-102, Jul. 2014.

[7] Daniel Díaz and Leonardo Marín, "VFH+ D: An Improvement on the VFH+ Algorithm for Dynamic Obstacle Avoidance and Local Planning", *IFAC-PapersOnLine*, Vol. 53, No. 2, pp. 9590-9595, 2020. <https://doi.org/10.1016/j.ifacol.2020.12.2450>.

[8] Xiaozhuo Yang, "Slam and navigation of indoor robot based on ROS and lidar", *Journal of Physics: Conference Series*, Vol. 1748, No. 2, pp. 1748-1754, Jan. 2021.

저자소개

김 형 수 (Hyung-Su Kim)



2012년 2월 : 상지대학교
컴퓨터정보공학부(공학사)
2014년 2월 : 상지대학교
컴퓨터정보공학부(공학석사)
2014년 3월 ~ 현재 : 상지대학교
정보통신소프트웨어 공학과
박사과정

관심분야 : 센서, 신호 및 영상처리, 임베디드시스템

한 영 환 (Young-Hwan Han)



1995년 8월 : 인하대학교
전자공학과(공학박사)
1996년 3월 ~ 현재 : 상지대학교
정보통신소프트웨어공학과 교수
관심분야 : 영상처리, 바이오인식,
임베디드시스템