실내 자율 비행을 위한 영상 기반의 위치 인식 시스템

문성태*, 조동현**, 한상혁***

Image-based Localization Recognition System for Indoor Autonomous Navigation

SungTae Moon*, Dong-Hyun Cho**, Sang-Hyuck Han***

Abstract

Recently, the localization recognition system research has been studied using various sensors according to increased interest in autonomous navigation flight. In case of indoor environment which cannot support GPS information, we have to look for another way to recognize current position. The Image-based localization recognition system has been interested although there are lots of way to know current pose. In this paper, we explain the localization recognition system based on mark and implementation of autonomous navigation flight. In order to apply to real environment which cannot support marks, localization based on real-time 3D map building is discussed.

초 록

최근 자율 비행에 대한 관심이 증가하면서 다양한 센서를 통한 자기 위치 인식 연구가 진행되고 있다. 특히 GPS와 같은 자기 위치를 확보할 수 없는 실내 환경의 경우, 다른 방법을 통해 자기 위치를 파악해야 한다. 실내 환경에서 자기 위치 파악에는 여러 가지 방법이 있지만 영상을 통한 위치 인식 기술이 각광을 받고 있다. 본 논문에서는 마크를 통한 영상 기반의 위치 인식 연구에 대해 설명하고, 실제 비행체에 적용하여 자율 비행하는 방법에 대해 제안한다. 그리고 마크가 없는 실제 환경에서도 위치를 인식할 수 있도록 실시간 3차원 지도 생성을 통한 위치 인식 방법에 대해서도 논의한다.

키워드 : Indoor Autonomous Navigation(실내 자율 주행), localization recognition system(위치 인식 시스템), 3D Map(3차원 지도)

1. 서 론

자율 비행에 대한 관심이 증가하면서 다양한 센서를 통한 자기 위치 인식 연구가 진행되고 있다. 특히 GPS 정보를 수신할 수 없는 실내 환경의 경우, 다른 방법을 통해 자기 위치를 파악해

접수일(2013년 5월 8일), 수정일(1차 : 2013년 6월 7일, 2차 : 2013년 6월 17일, 게재확정일 : 2013년 7월 1일)
* 융합기술연구팀/stmoon@kari.re.kr ** 융합기술연구팀/dhcho99@kari.re.kr *** 융합기술연구팀/shan@kari.re.kr

야 한다. 실내에서 위치 확인을 위해 널리 알려진 방법 중 하나는 모션 캡쳐(Motion Capture)를 사용하여 미리 설치한 마커를 인식하는 방법이다. 이 방법의 경우 최대 2000Hz로 마커를 인지하고 1cm 미만의 정확도로 위치를 파악할 수 있지만, 가격이 고가이고, 인지할 수 있는 영역이 상대적으로 적다[1]. 따라서 넓은 범위를 갖는 빌딩에서 위치를 인식하기는 쉽지 않다. 이 외에도 가속도 센서를 활용하여 자기위치를 확인하는 방법이 있지만, 이 경우도 마찬가지로 넓은 범위를 갖는 영역에서는 오류가 축적되는 문제가 발생하기 때문에 정확한 위치를 파악하기 어렵다. 마지막으로 영상 기반의 위치 인식 방법이 있다. 영상 기반의 위치 인식의 경우, 모션 캡쳐 혹은 가속도 센서 방식과는 달리 넓은 범위의 영역에 적용이 가능하다. 이미지 기반의 위치 인식은 기존 방식과 달리 많은 계산처리가 필요하지만, 최근 전체적인 컴퓨터 성능이 높아지고, GPU를 통한 계산 능력이 급격히 향상됨에 따라 실시간으로 처리가 가능해졌다.

본 논문에서는 영상 기반 위치 인식 방법에 대해 소개한다. 우선, 가상 현실 연구 분야에서 소개한 마크 영상만을 가지고 현재 위치를 파악하는 기술을 소개하고, 이를 기반으로 AR.Drone 쿼드로터를 통해 실제 무인기 내에서 다양한 센서 정보와 마크 영상을 융합하여 보다 정확한 위치 인식 방법에 대해 제안한다. 마지막으로 마크 없이 순수 영상만을 가지고 3차원 지도를 생성하고 이를 기반으로 현재 위치를 파악하는 기술에 대해 소개한다.

논문의 구성은 다음과 같다. 2장에서는 영상을 통한 위치 확인 방법에 대해 설명하고, 3장에서는 이를 기반으로 비행체 내의 영상을 통한 위치 확인 방법에 대해서 설명하다. 이후 4장에서는 3차원 지도를 기반으로 위치확인하는 방법에 대해 설명한다. 마지막으로 5장에서는 결론과 향후 계획에 대해 논의한다.

2. 영상을 통한 위치 확인

컴퓨터 비전 분야에서는 영상을 통한 증강 현실(Augmented Reality)을 연구해왔다. 증강 현실을 위해서는 그림 1과 같이 현재 위치를 파악하여 실제 영상위에 컴퓨터 그래픽스 영상을 정해진 위치에 레이아웃 시켜야 한다. 이를 위해 다양한 방법이 사용되고 있지만, 가장 정확한 방법으로 마크가 사용된다. 마크는 특정 패턴을 가지고, 다른 배경과 구별이 될 수 있어 영상 처리를 통해 컴퓨터가 인지할 수 있는 특정 이미지이다. 그림 1과 같이 마크의 위치를 기억하고, 그 위치에 컴퓨터 그래픽스 영상을 올려놓거나, 특정 그림을 그릴 수 있다. 따라서 증강 현실을 실현하기 위해서는 현재 위치를 파악하는 기술이 반드시 필요하다.

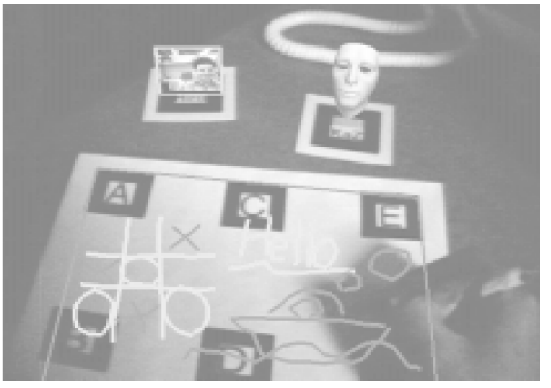

그림 1 증강현실을 통한 화이트 보드

2.1. 알고리즘

카메라로 마커를 보는 경우 마커 좌표계(Marker Coordinate)와 카메라 좌표계(Camera Coordinate)간의 관계가 그림 2와 같이 이루어진다.

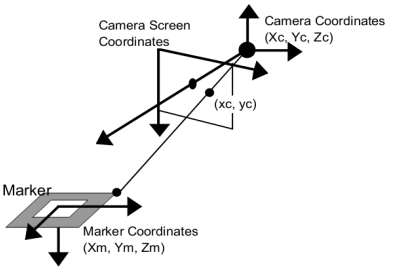

그림 2 마커 및 카메라 좌표계의 관계

카메라의 위치를 확인하기 위해서는 미리 알고 있는 마커의 정보(사이즈 및 위치)를 가지고 카메라 좌표계를 얻어야 한다. 이를 위해 마커 좌표계로부터 카메라 좌표계로의 변환(Transformation)을 알아야 한다. 이를 위해 수식 1과 같이 동차 행렬 변환(Homogeneous Transformation Matrix)을 구해야 한다. 동차 행렬 변환은 수식 1과 같이 회전을 위한 V와 이동을 위한 W로 구성이 되어 있다.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} V_{3\times3} & W_{3\times1} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (1)$$

H. Kato는 카메라의 포즈를 구하기 위해 카메라에서 받은 패턴에서 평행한 변의 직선식을 구하고 이를 카메라의 좌표계에서 바라보았을때 나오는 평면식을 통해 V와 W를 구하는 방법을 제시하였다.[2]

2.2. 시험

카메라 정보를 파악할 수 있는 도구인 ARToolki이 개발되어 있고[2] 이를 통하여 확인한 결과 그림 3과 같이 마커를 통해 카메라의 위치를 파악할 수 있었다.

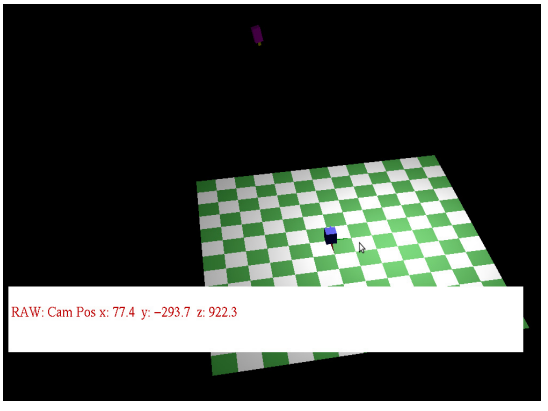

그림 3 ARToolkit을 사용하여 카메라 포즈를 확인 실험

2.3. 문제점

영상만을 가지고 위치 확인을 하는 경우, 카메라의 성능에 따라 큰 영향을 미친다. 카메라의 화질이 떨어지는 경우 정확하지 않은 원근감 계산으로 인해 정확한 위치 확인이 어렵다. 그리고 카메라의 프레임 속도가 떨어지는 경우 이미지가 번지는 블러(Blur)현상으로 인해 마크를 파악하지 못하는 경우도 발생한다. 뿐만 아니라 카메라의 화질과 속도가 높은 경우 주로 무게가 많이 나가기 때문에 실내용 비행체에 탑재하기 어렵다는 문제점이 있다. 그리고 영상만을 가지고 측정하는 경우 그림 4와 같이 거리가 멀어질 수록 에러률이 현저하게 높아지기 때문에 1m 이상의 거리에서 측정이 필요한 경우 영상만 가지고 측정하기 어렵다.

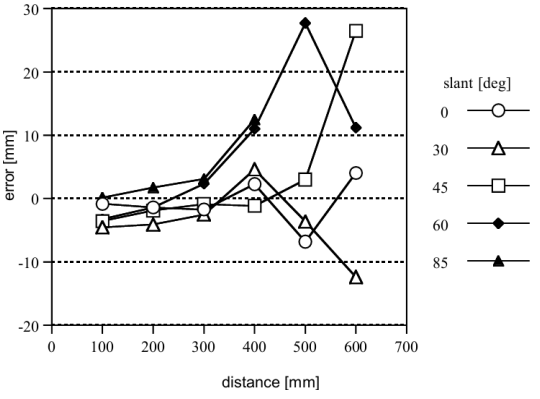

그림 4 거리와 기울기에 따른 에러 발생률[2]

3. 비행체 내의 영상을 통한 위치확인

영상만을 가지고 위치를 확인하는 경우 많은 한계점이 있었다. 특히 비행체에서 탑재된 카메라를 통해 위치를 확인하는 것은 비행체의 떨림과 속도로 인해 정확도가 현저히 떨어진다. 이를 해결하기 위해 비행체에 탑재된 센서 정보를 활용한다. 본 논문에서는 실험을 위해 비행체인 AR.Drone을 사용한다. AR.Drone는 높이 정보와

기울기 정보를 다양한 센서를 통해 제공하기 때문에 이를 통해 보다 정확한 위치 정보를 확보할 수 있다.

3.1. AR.Drone

AR.Drone(Augmented Reality Drone)은 Parrot사에서 개발한 저가의 소형 무인 쿼드로터 비행체로써 기존 쿼드로터와 달리 이미지 프로세싱 IT 기술을 융합하여 보다 정밀하고 쉽게 제어가 가능하다. 특히 WiFi로 통신이 이루어지므로 스마트폰을 활용하여 제어할 수 있다. 그리고 SDK가 무료로 제공되어 개발자가 쉽게 제어 프로그램을 개발할 수 있어서 다양한 분야에서 AR.Drone을 활용하고 있다. AR.Drone은 기존 쿼드로터와 달리 바닥 면에 부착된 카메라로부터 연속적으로 받은 프레임 간의 변화를 추정하여 움직임 정보를 획득하는 옵티컬 플로우 기법을 사용하였다. 이 방법을 통해 AR.Drone은 외부의 영향이 있더라도 이동 없이 현재 위치에 머무를 수 있다.

그림 5 AR.Drone 2.0

AR.Drone은 그림 5과 같이 네 개의 프로펠러가 있는 쿼드로터의 일종으로 실내와 실외에서 안전하게 사용할 수 있도록 Hull이 존재한다. 최근 출시된 AR.Drone 2.0은 1 GHz의 32 bit ARM Cortex A8 프로세서를 탑재하였다. 그리고 영상 수집 및 처리를 위해 1280x720 화질의 전면 카메라와 640x480 화질의 바닥 면 카메라를 가지고 있다. 각각의 카메라는 30 fps와 60 fps의 성능으로 촬영이 가능하다. [3].

본 논문에서는 바닥 면 카메라를 통해 바닥에 부착된 마크들을 인지하는 방법을 사용한다. AR.Drone은 내부 프로그램을 직접 수정할 수 없기 때문에 영상 정보를 서버에 무선 통신을 통해 전달하고 수신된 영상을 처리하는 방식을 사용하였다. AR.Drone은 H.264로 압축된 영상을 TCP로 서버로 전달한다.

3.2. 마크

비행 공간을 충분히 확보하기 위해 본 논문에서는 각각 구분이 되는 마크를 다수 활용한 매직 카펫을 활용하였다. 매직 카펫이란 그림 6과 같이 다수의 패턴 배열을 카펫 형식으로 표현한 것으로써 넓은 범위의 위치를 측정하기 위해 사용된다.

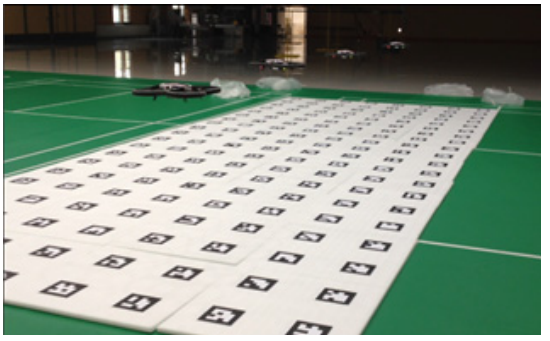

그림 6 패턴을 통한 자율 비행

마크는 그림 7과 같이 NxN 블록으로 구분되어 있고, 각 블록은 흰색과 검정색 사각형 모양으로 이루어져있다. 이 중 마지막에 존재하는 블록은 마크 인식 정확도를 높이기 위해 패리티 비트와 같은 역할로 흰색 사각형과 검정색 블록의 개수를 확인하여 검사한다. 탑재 카메라는 마크를 인식하여 네 꼭지점을 찾아내고, 마크 내에 있는 사각형의 분포 형태를 보고 마크의 고유 아이디를 구분한다. 꼭지점에 위치하는 블록은 하나를 제외하는 나머지 세 개는 모두 흰색이고, 한 부분만 검정색으로 구분하고 있기 때문에 마크의 방향성을 구분 가능하다. 결국 마크를 통해 각 마크를 구분할 수 있고, 꼭지점 위치를 통해 실제 위치를 측정할 수 있다[4].

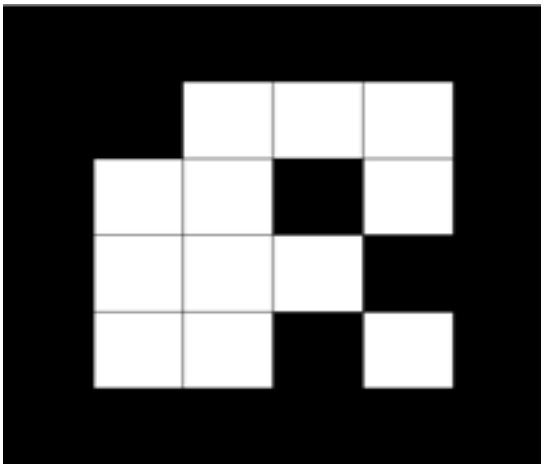

그림 7 마크 예

3.3. 알고리즘

위치 파악에 앞서, 카메라 화각 정보와 높이 정보가 필요하다. AR.Drone의 화각은 64도이고, Altitude 정보는 바닥면에 부착된 Sonar를 통해 확인가능하다. 마지막으로 y축 기울기(pitch)와 x축 기울기(roll) 정보는 AR.Drone의 IMU 센서를 통해 얻을 수 있다. 이 정보를 통해 중심으로부터 떨어진 가로축과 세로축의 떨어진 거리인 Hd와 Vd를 수식 2와 같이 구한다. 여기서 α와 β는 화면 중심으로부터 얼마나 떨어져 있는지각도로 나타낸 값이고, roll과 pitch는 비행체가 기울어진 정도를 나타낸다[5].

$$\begin{aligned} Hd &= c \cdot \tan(\alpha + roll) \\ Vd &= c \cdot tan(\beta + pitch) \end{aligned} \tag{4}$$

한편, AR.Drone이 z축으로 회전(yaw 회전)을 한 경우, 이를 반영해야 한다. Yaw 축 회전 각을 μ로 한 경우 실제 이동 각 Θ는 수식 3과 같다. 그리고 특정 패턴으로부터 떨어진 거리인 DeltaX, DeltaY는 수식 3과 같이 구할 수 있다.

$$\begin{aligned} \theta &= \arctan\left(\frac{Vd}{Hd}\right) + \mu \\ DeltaX &= \sqrt{Hd^2 + Vd^2} \cdot \cos\theta \\ DeltaY &= \sqrt{Hd^2 + Vd^2} \cdot sin\theta \end{aligned} \tag{5}$$

AR.Drone의 위치는 선택한 패턴으로부터 DeltaX 및 DeltaY만큼 떨어져있는 곳이므로 수식 4를 통해 실제 위치를 파악할 수 있다.

$$\begin{aligned} X &= PatternX - DeltaX \\ Y &= PatternY - DeltaY \end{aligned} \tag{6}$$

3.4. 시험

이 로직을 통해 패턴을 보고 현재 위치를 파악하기 위해 AR.Drone으로부터 이미지 영상을 수신하고, 영상을 OpenCV[6]를 통해 분석하여 패턴을 찾았다. 이후 3.3절에서 언급한 알고리즘을 사용하여 AR.Drone의 위치를 파악하고 그 결과를 MRPT(Mobile Robot Programming Toolkit)[7]를 이용하여 3차원 공간상에서 위치를 확인해보았다. 그 결과 그림 8과 같이 마크를 정확하게 인식하였고, 위치를 확인할 수 있었다.

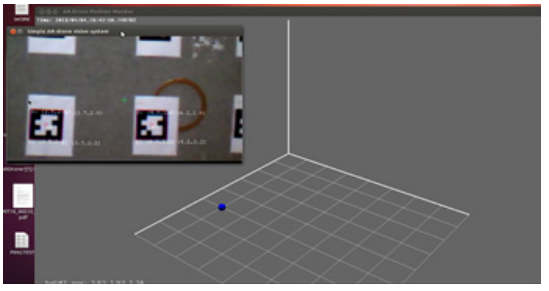

그림 8 패턴 인식을 통한 위치 확인

3.5. 문제점

마크를 사용하는 경우 보다 정확한 위치를 확보할 수 있고, 간단한 알고리즘을 통해 현재 위치를 확보할 수 있다. 하지만, 이 경우 마크가 존재해야 한다는 가정이 들어가기 때문에 불특정 공간에 적용하기가 어렵다. 따라서 확인되지 않는 공간을 비행하기 위해서는 마크 없이 동작할 수 있는 시스템이 필요하다.

4. 마크리스를 통한 위치 확인

2장과 3장에서 언급한 위치 확인 방법은 모두 특정 위치를 파악하기 위해 영상 내에 마크가 필

요하다. 하지만 실제 환경에서는 마크를 사용할 수 없기 때문에 새로운 방식이 필요하다. 이미지 처리분야에서는 영상의 특징을 이용하여 특징점을 찾는 방법을 연구해 왔다. 그 결과 SIFT, SURF, ORB, BRIEF와 같은 다양한 방법을 제시하였다. 이 방법을 사용하게 되면 확대, 축소, 회전과 같은 변화가 있어도 동일한 위치를 특징점으로 찾을 수 있다.

한편 공간상에서 이동하는 비행체가 위치 정보를 확보하기 위해서는 공간 정보가 필요하다. 본 논문에서는 공간 정보를 얻기 위해 RGB-D 센서를 사용하여 3차원 지도를 생성하고, 이를 기반으로 위치를 확인하는 방법에 대해 소개한다.

4.1. RGB-D 센서 카메라

RGB-D 센서는 TOF (Time Of Flight) 방식과 Structured light 방식으로 구분할 수 있다. TOF 방식은 IR이 움직이는 시간을 감지해서 수신기가 받은 시간과 비교해 그 값으로 깊이를 인지하는 방식이다. 파나소닉의 D-Imager 같은 카메라가 이 방식을 취하고 있다. 이 방식은 단순히 거리만 측정하는 알고리즘이 적용되기 때문에 간단하고 빠르다. 하지만 하나의 정보만 가지고 깊이를 판단하기 때문에 외부 간섭의 영향이 있을 경우 결과값이 변경될 가능성이 높고, 빛의 영향도 크다. 반면에 Structured Light 방식은 TOF 방식의 거리 측정이 아닌 사물의 형태 자체를 인식하는 것이기에 조금 더 외부 환경에 강건하다고 볼 수 있다. 본 논문에서는 Structured Light 방식인 Xtion Pro[8]를 사용한다.

그림 9 Xtion Pro[8]

Xtion Pro는 아수스社에서 개발한 깊이 센서 장비로서 RGB 카메라, infrared(IR) 프로젝터, 및 IR 카메라가 장착되어있다(4). Xtion Pro는 IR 프로젝터와 IR 카메라를 사용하여 깊이 정보를 획득하고 RGB 카메라에서 나온 영상 정보를 동시에 받아들이는 PrimeSense 기술을 사용한다. 초기 RGB-D 센서는 동작 인식 및 물체 추적 기능을 활용하여 게임용으로 사용되었지만, 최근에는 낮은 가격과 높은 성능 때문에 다양한 연구 분야에 활용되고 있다. 특히 오픈소스로 내부 드라이버 및 SDK가 공개되어 있어서 쉽게 수정이 가능하다. Xtion Pro의 스펙은 표 1과 같다[8].

표 1. Xtion Pro 스펙[8]

특징	값
Light Source	Infrared
Field of View	Horizontal 53° Vertical 45°
Dimensions	1.5"x7"x1.9"
Measurable Range	0.8~3.5m
Angular Resolution	0.08 deg
Scan Time	33 msec/scan
Interface	USB
Supported OS	Windows|Linux 32/64
Framework	OpenNI
Program Language	C/C++/C#

한편, Xtion Pro를 사용하여 실제 측정하면 특정 면을 측정 시 하나의 평면으로 나오지 않고 그림과 같이 사선으로 평면이 나타나는 것을 확인할 수 있었다. 이런 현상은 Xtion Pro와 같은 저가 RGB-D 센서의 성능 문제로 그림 10과 같이 최대 2Cm에서 10Cm 오차가 발생하였다. 결국 이러한 현상으로 인해 3차원 지도를 생성하기 위해 이미지 특징점을 이용하는 경우, 잘못된 깊이 정보로 인해 정확하지 않은 포즈 계산이 이루어지는 문제가 발생하였다.

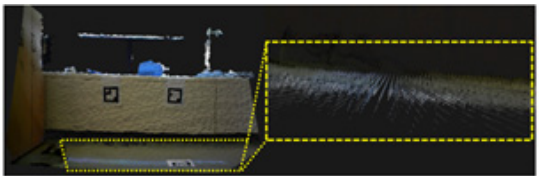

그림 10 RGB-D 센서의 깊이 정보 오차

4.2. 알고리즘

실내에서는 GPS를 사용할 수 없기 때문에 위치 추적을 위해 다른 방법이 필요하다. 3차원 지도를 만드는 방법과 동일하게 현재 위치에서 RGB-D 센서를 사용하여 획득한 정보를 가지고 카메라 포즈를 계산할 수 있지만, 무인 비행체와 같이 무게에 민감한 장치에 위치 추적하기 위해 RGB-D 센서와 같은 무거운 카메라를 탑재하기는 어렵다. 따라서 본 논문에서는 소형 카메라를 통해 얻은 2차원 이미지를 사용하여 기존에 생성한 3차원 지도에서의 위치를 얻는 방법을 사용하고자 한다.

2차원 이미지와 3차원 지도가 있을 때 카메라 포즈를 계산하는 문제는 Perspective-n-Point Problem (PnP)로 알려져 있다. PnP란 카메라의 intrinsic 값이 주어졌을 때 카메라의 위치 및 촬영 각도를 파악하는 방법을 말한다.

3차원 지도 데이터에 대해 Homogeneous 값을 P라고 하고, 2차원 이미지 데이터에 대한 Homogenous 값을 x라고 할 때 다음 식이 도출할 수 있다.

$$x = K[R|t]P \tag{7}$$

여기서 K는 3x3 calibration matrix 이고, 사전에 카메라 calibration을 통해 획득가능하다. R은 3x3 rotation matrix이고, t는 3x1 translation matrix이다. 따라서 카메라 포즈를 알기 위해서 R과 t를 구해야 한다.

우선 P는 3차원 지도상의 좌표계를 사용하므로 그림 11과 같이 카메라상의 좌표계인 p로 수정해야 한다. P와 p의 관계는 식 8과 같으므로, 이미지 데이터와의 관계는 식 9와 같다.

$$p = [R|t]P \tag{8}$$

$$p = K^{-1}x \tag{9}$$

이때 c를 원점으로 x까지의 unit vector인, $\hat{x}$은 식 (10)과 같이 구할 수 있다.

$$\hat{x} = \frac{K^{-1}x_i}{\|K^{-1}x_i\|} \tag{10}$$

p와 x와의 관계는 식 11과 같이 되기 때문에 x와 p간의 거리인 d를 구하게 되면 카메라의 포즈를 구할 수 있게 된다.

$$p_i = d_i\hat{x}_i + c \tag{11}$$

d 값을 구하기 위해 코사인 법칙을 이용하여 c, p_i, p_j간의 관계를 식 12와 같이 도출 할 수 있고, 세 개의 데이터 셋이 주어지면, Sylverster resultants[11]를 통해 d값을 측정할 수 있다.

$$d_{ij} = d_i^2 + d_j^2 - 2d_id_j\cos\theta_{ij} \tag{12}$$

하지만 실제 데이터는 일부 오류가 있기 때문에 RANSAC[9] 알고리즘을 이용하여 오류 부분을 제거하고, 최적의 위치를 찾아내야 한다.

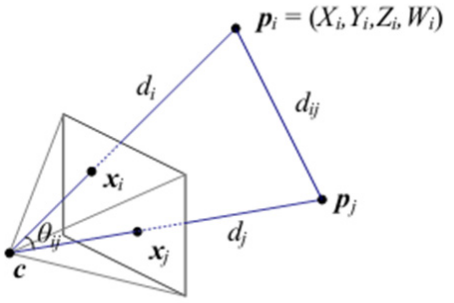

그림 11. 카메라 위치 추정 방법

본 논문에서는 OpenCV에서 제공하는 EPnP (Efficient Perspective-n-Point)기법을 사용하여 PnP 문제를 해결하였고, 그림 11과 같이 카메라 위치를 파악할 수 있었다.

4.3. 시험

기 생성된 3차원 지도를 기반으로 임의의 위치에서 촬영한 2차원 이미지의 카메라 포즈를 측정하는 실험을 하였다. 2차원 이미지가 입력되면 우선 이미지 상의 특징점을 찾는다. 그리고 특징점의 위치를 저장된 3차원 지도상의 특징점들과

비교하여 한계값 이상으로 특징점이 나오는 경우 매칭된 특징점으로 판단한다. 아래 그림은 2차원 이미지를 3차원 지도 내에서 특징점을 찾아본 결과이다. 그림 12와 같이 2차원 이미지의 특징점들이 3차원 지도 상에서 지정된 부분만 특징점이 분포되어 있음을 확인할 수 있었다. 시험을 위해 PCL 라이브러리[10]를 사용하였다.

그림 12 3차원 지도와 2차원 이미지간의 특징점 매칭 정보

3차원 지도와 이미지간의 매칭된 위치를 선을 그어 가시화 해보면 그림 13과 같이 나타나고, 이를 RANSAC을 통해 오류를 제거한 후 확인해 본 결과 카메라의 위치를 찾을 수 있었다.

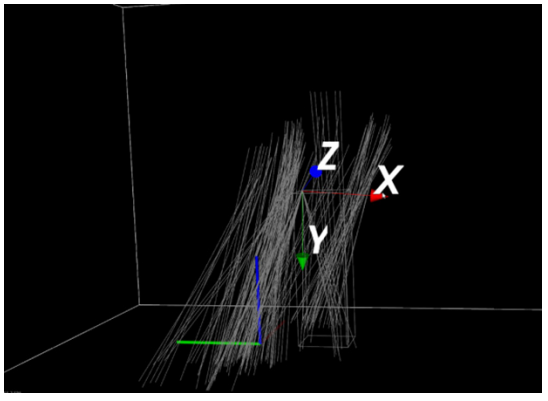

그림 13 3차원 지도와 2차원 이미지간의 매칭 부분 가시화

이를 기반으로 실제 위치를 찾아보니 그림 14와 같이 카메라 포즈를 구할 수 있었다.

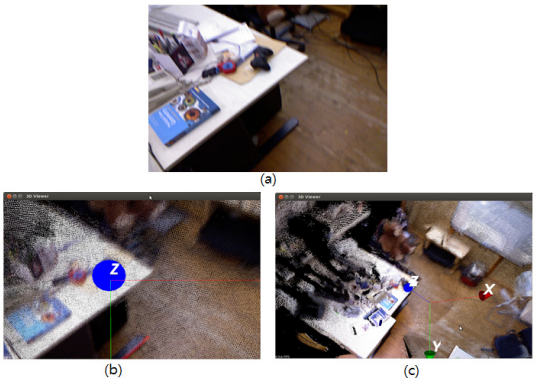

그림 14 2차원 이미지를 가지고 카메라 포즈 계산 결과 (a) 2차원 이미지 (b) 카메라 포즈에서 바라본 3차원 지도 화면 (c) 다른 각도에서 바라본 3차원 지도 화면

5. 결론 및 향후 계획

본 논문에서는 영상 기반 위치 인식 시스템을 구현하기 위해 다양한 방법을 소개하였다. 우선 컴퓨터 비전 분야에서 활용하는 순수 영상 기반의 위치인식 방법에 대해 설명하였다. 이 경우 어두운 환경 혹은 카메라의 성능에 따라 정확도에 많은 영향을 미친다. 따라서 영상을 처리할 수 없는 상황에서도 고려하는 경우 비행체에 센서 정보를 활용하여 보다 정확한 현재 위치를 파악할 수 있다. 따라서 협소하고 정해진 공간상에서의 자율 주행을 위해서는 매직 카펫만을 이용해도 충분하다. 하지만 넓고 정해져있지 않은 실제 환경에서는 마크를 사용할 수 없기 때문에 3차원 지도를 미리 생성하고 이를 기반으로 위치를 파악하는 시스템이 필요하다. 이를 위해 RGB-D 센서를 통해 3차원 지도를 생성하고 위치를 파악한다.

비행체를 통한 3차원 지도 생성 및 위치 추적을 위해서는 고 성능의 시스템이 필요하다. 특히 특징점을 파악하고 저장하기 위해서는 GPU(Graphic Processor Unit)의 기능이 필수적으로 필요하다. 추후 본 논문의 결과물을 이용하여 비행체에 직접 적용하여 넓은 범위의 실내 환경에서 정해진 목적지까지 도착하는 시스템을 개발할 계획이다.

약 어 정 리

GPS	Global Positioning System
3D	3 Dimension
SLAM	Simultaneous Localization And Mapping
SIFT	Scale-Invariant Feature Transform
SURF	Speed Up Robust Feature
ORB	ORiented BRIEF
BRIEF	Binary Robust Independent Elementary
RANSAC	RANdom SAmple Consensus
ICP	Iterative Closest Points
GICP	Generalized-ICP
GPU	Graphic Processor Unit
OpenCV	Open Computer Vision
PnP	Perspective-n-Point
EPnP	Efficient Perspective-n-Point
MRPT	Moblie Robot Programming Toolkit
PCL	Point Cloud Library

참 고 문 헌

1. 이민기, 박성규, 박근표, “모션캡쳐 기술 동향,” 전자통신동향분석, Vol. 22, Issue 4, 2007, pp. 35~42
2. H. Kato, M. Billinghurst, "Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System," IWAR99, 1999, pp. 85~94
3. P. Bristeau, F. Callou, "The Navigation and Control technology inside the AR.Drone micro UAV," IFAC, 2011, pp. 1477~1484
4. 양광웅, “OpenCV Marker Recognition,” http://blog.daum.net/pg365/159
5. T. Graduacao, "A Java Autopilot for Parrot A.R. Drone Designed with DiaSpec," 2011
6. G. Bradski, A. Kaehler, 2008, “Learning OpenCV,” OReilly
7. MRPT, "http://mrpt.org"
8. Xtion Pro, "http://www.asus.com/Multimedia/Xtion_PRO_LIVE"
9. M. Fischeler, R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Graphics and Image Processing, Vol. 24, No. 6, 1981, pp. 381~395.
10. R. Rusu, S. Cousins, “Plane Model Segmentation,” http://pointclouds.org
11. Cox.D, O'Shea. D, "Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra,", Springer. 2007