pISSN: 2287-5905

http://dx.doi.org/10.3745/KTSDE.2013.2.5.353

A Study on Autonomous Indoor Flight using Computer Vision System and Smartphone

Choi Young† · Kim Kye-Young††

ABSTRACT

In this paper, we present an implementation of indoor flight to navigate to the designated places capable of hands-off autonomous operation within indoor environments. Our flight requires computer vision technique and smartphone device to allow it to be flown indoors without high-performance sensors which are too expensive to commercialization. The experimental result show that proposed implementation is fairly meaningful in a general building.

Keywords : Indoor Flight, Smartphone, Computer Vision, Marker

컴퓨터비전과 스마트폰을 활용한 실내 자동비행체에 관한 연구

최 영† · 김 계 영††

요 약

본 논문에서는 실내 환경에서 컴퓨터비전, 스마트폰과 헬륨을 사용하여 특정한 장소를 찾아갈 수 있는 자동비행체 구현 방법을 제안한다. 먼저, 마커를 이용하여 빌딩 내 각 정점을 표시하고, 다익스트라 알고리즘을 활용하여 자동으로 최단의 경로를 찾는다. 제안된 방법을 이용하여 그 경로를 따라 비행하기 위한 최적의 비행체 구조를 설계한다. 실험 결과 다양한 방면으로 적용 가능한 유의미한 결과를 얻었다.

키워드 : 실내 비행체, 스마트폰, 컴퓨터 비전, 마커

1. 서론 및 관련 연구

특정한 장소를 이동할 수 있는 자동 비행체는 그 활용방안이 다양하여 산업적, 군사적 목적 등 다방면으로 연구가 진행되고 있다. 미국 라이트-패터슨 공군기지(Wright-Patterson Air Force Base)에서는 군사용 목적으로 벌레를 본 딴 초소형 비행체를 연구 중이다[1]. 카메라와 마이크를 장착한 채로 최소화된 비행체를 개발하려는 이 프로젝트는 아직까지는 걸음마 단계로, 2015년 까지 새 크기, 2030년 까지 곤충 크기의 로봇을 만드는 것이 목표이다. 신시내티 대학(University of Cincinnati)에서는 날개 형식의 무인 비행체를 이용하여 화재 감지를 하고자 하는 프로젝트를 진행하고 있다[2]. 이 비행체는 54인치의 날개폭을 가졌고 시속 35마일로 날 수 있으며 GPS를 통하여 위치를 판단하며 카메라로 촬영한 비디오를 전송할 수 있다. 또한 미 육군에서는 미시건대(University of Michigan)와 공동으로 박쥐의 비행형태를 모방한 비행체를 개발 중에 있다[3-4]. 이 비행체는 길이 약 15cm, 무게 110g정도를 목표로 하며 음파 탐지기를 이용한다. 이 프로젝트에는 연구비로 5년간 1천만 달러가 배정되었다. 이처럼 무인 비행체는 보통 많은 센서와 비싼 연산처리장치 및 관련 장비를 포함하기 마련이므로 대중적인 용도로 활용하기에는 무리가 있다. 본 연구는 이러한 자동 비행체를 무인 수송, 무인 순찰, 엔터테인먼트 등의 대중적인 목적으로 활용하기 쉽게 하기 위하여 실내 환경에서 스마트폰을 이용한 헬륨 풍선 형태의 저속 자동비행체 모델의 개발을 목표로 한다. 스마트폰은 그 잉여수량이 갈수록 증가하고 있으며 카메라를 비롯해 여러 센서 및 고속 연산처리장치를 이미 가지고 있으므로 비행체는 상대적으로 저렴한 단가에 제작될 수 있다. 또한, 일반 비행체는 속도가 빠르고 세밀한 조작이 어려워 실내에서 사용하기에는 위험이 따르나, 본 프로젝트에서는 비행 시 필요한 부양력을 프로펠러가 아닌 헬륨으로 충당하여 이러한 문제를 해결하고자 한다.

※ 이 논문은 한이음 IT멘토링 제도의 지원으로 연구되었음.
※ 이 논문은 한국정보처리학회 제38회 추계학술발표대회에서 '컴퓨터비전을 활용한 실내 자동비행체에 관한 연구'의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 숭실대학교 컴퓨터학부 학부생
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
논문접수 : 2012년 12월 24일
수 정 일 : 1차 2013년 1월 22일
심사완료 : 2013년 1월 22일
* Corresponding Author : Kim Kye-Young(gykim11@ssu.ac.kr)

2. 비행체의 형태 및 크기

본 연구에서는 상하, 전후, 좌우, 회전 이동이 가능한 일반적인 형태의 비행체의 형태로써 Fig. 1과 같이 눕혀진 회전 타원체(Prolate Spheroid)의 기낭에 앞, 뒤, 하단 각각 모터를 장착하고 스마트폰을 기체 하단 중앙에 부착한 형태를 사용한다.

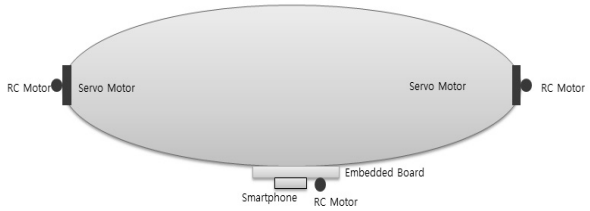


Fig. 1. Structure of flight

이때 기낭의 크기는 최소한 비행체 전체를 들어 올릴 수 있을 만큼의 헬륨이 유입 가능한 크기를 가져야 한다. a가 회전타원체의 단축이고 c가 회전타원체의 장축일 때 회전타원체의 부피는 Equation (1)과 같다.

$$V_s = \frac{3}{4}\pi a^2 c \tag{1}$$

또한 t가 기낭 원단의 두께이고 g가 기낭 원단 재질의 비중일 때 기낭의 무게는 Equation (2)와 같다.

$$M_s = tg2\pi a^2(1+\frac{c}{ae}sin^{-1}e) \tag{2}$$

$$\text{where } e^2 = 1-\frac{a^2}{c^2}$$

헬륨 $1m^3$ 당 부양력은 1,127g으로 알려졌으므로 기낭의 부피에 따른 부양력은 $V_c \times 1127$g이 된다. 따라서 기낭을 제외한 비행체의 나머지 무게를 M_o라고 한다면 V_s는 아래의 Equation (3)을 만족하여야 한다. (단위 : m^3)

$$V_s > \frac{M_s + M_o}{1127} \tag{3}$$

3. 마커의 필요성 및 인식 방법

비행체가 각 정점(현 위치, 목적지 등)을 인식할 수 있도록 하기 위해서는 정점의 위치 나타내는 표식이 필요하다. 본 연구에서는 비행체가 비행하는 빌딩의 각 정점의 바닥에 Fig. 2와 같은 형태의 마커를 미리 설치해 두고 비행체에 부착된 스마트폰의 카메라로 인식함으로써 이러한 문제를 해결하도록 한다.

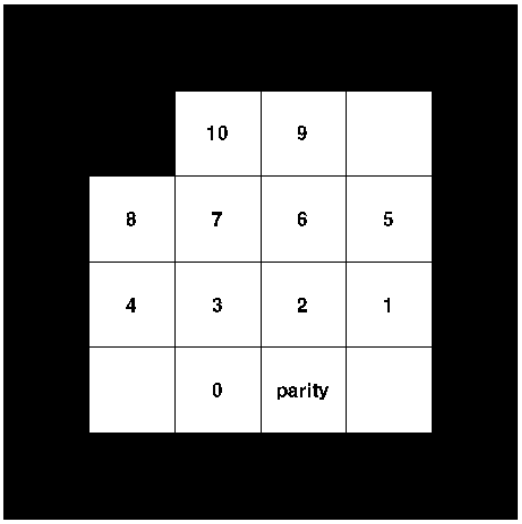


Fig. 2. An example of marker

마커의 인식은 양광웅이 제안한 OpenCV Marker Recognition 2[5]를 따른다. 마커의 테두리를 이루는 사각형들은 항상 흑색으로 표현되어 마커의 영역임을 인식하게 하며 마커의 가장 좌상단을 항상 흑색으로 표현함으로써 마커의 방향을 표현한다. Fig. 2의 중앙 영역은 bit 번호를 의미하며 이 bit 조합으로 마커의 고유한 ID를 표현한다.

4. 가상맵 및 자동 경로 탐색

비행체의 자동 비행을 위해 빌딩 내 각 인접경로는 미리 특정한 데이터베이스에 저장되어 있을 필요가 있다. 저장되는 정보는 인접경로의 이름(SRC to DEST), 목적지의 방향(degree), 경사(계단)이동 유무(상승, 하강, 유지)이다.

저장된 데이터베이스의 정보를 가지고 아래의 Fig. 3과 같은 개념의 가상맵을 생성한다. 가상맵은 그래프 자료구조로 이루어지며 경로 탐색에 가장 널리 쓰이는 Dijkstra's

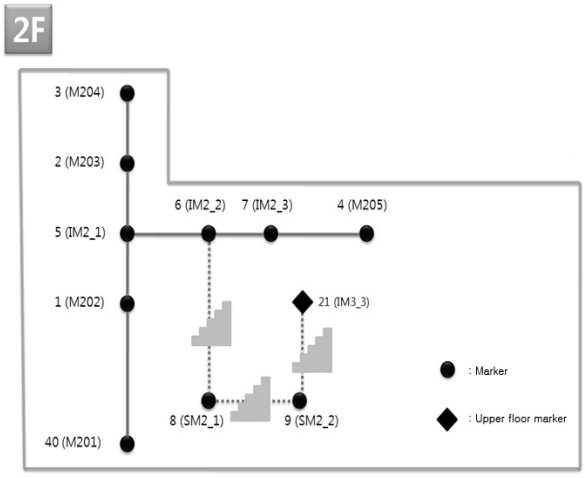


Fig. 3. An example of virtual map

Algorithm[6]을 적용하여 비행체가 최적의 경로를 자동으로 찾을 수 있도록 한다.

5. 무선 통신 네트워크

비행체에는 두 가지 무선 통신 네트워크가 필요하다. 하나는 비행체에 사용자의 명령을 전송하기 위한 비행체↔사용자 간의 네트워크고, 또 하나는 비행체의 스마트폰에서 임베디드 보드에 명령을 내리기 위한 스마트폰↔임베디드 보드 간의 네트워크다. 비행체↔사용자 네트워크는 스마트폰의 장점을 살리기 위해 3G 기반의 C2DM을 이용하며 스마트폰과 임베디드 보드간의 통신에는 모든 스마트폰에 표준적으로 들어가 있는 기술인 블루투스를 이용한다.

6. 비행 알고리즘

비행체의 비행은 3가지 상태로 분류된다. 첫 번째는 대기 상태로 비행체는 정점(마커) 위에서 벗어나지 않으면서 정적인 움직임을 목표로 하는 상태이다. 두 번째는 출발 상태로 목적지가 정해졌을 시 비행체가 인접한 다음 마커를 향해 방향을 맞추는 상태이다. 세 번째는 비행 상태로, 비행체가 마커와 마커 사이를 실제로 비행하는 상태이다. 비행체의 상태는 Fig. 4의 의사코드를 따른다.

```
void function state_change()
{
    state = waiting
    while()
    {
        if(state == waiting)
            if(Receive destination information)
                state = starting
            wait_state_control() // 6.1

        else if (state == starting)
            tart_state_control() // 6.1.
            if(Direction of flight is similar to
              direction of destination)
                state = flying
                save present azimuth value

        else // Flying state
            flight_state_control() //6.2.
            if(Find marker)
                if(It is destination marker)
                    state = waiting
                else
                    state = starting
    }
}
```

Fig. 4. Pseudo-code of flight control algorithm

```
void function wait_state_control(Position of destination)
{
    Check height variation
    if(height variation <= thresholds )
        Check present height
        if(Flight's present height is different from criterion
          height)
            Adjust intensity of height control motor to going to
            criterion height
        else
            Stop height control motor
    else
        Adjust intensity of height control motor to flying
        oppositely

    Check position and rotate variation
    if(position and rotate variation <= thresholds)
        Check present position
        if(Flight's present position is different from criterion
          position)
            Adjust intensity of position control motor to going
            to criterion position
    else
        Adjust intensity of position control motor to flying
        oppositely

    /* starting state */
    Check present direction
    if(Direction of flight is similar to direction of destination)
        state = flying
    else
        Adjust intensity of position control motor to similar to
        direction of destination
}
```

Fig. 5. Pseudo-code of waiting and starting control algorithm

출발상태에서 azimuth(방위)를 저장하는 이유는 비행 상태에서는 영상 정보만으로는 현재 비행체의 방향이 최초 출발 시의 방향과 동일하게 유지되고 있는지 알 수 없기 때문이다. azimuth 값은 비행체에 장착되는 스마트폰의 오리엔테이션 센서를 이용하여 구한다.

6.1 대기 및 출발상태 제어 알고리즘

대기상태와 출발상태는 최대한 정적인 움직임을 목표로 하기 때문에 현재 비행체의 위치뿐만이 아니라 각 방향으로의 속도, 가속도, 회전가속도 모두를 고려할 필요가 있다. 특히 본 연구에서 가정하고 있는 비행체는 헬륨을 이용한 형태이기 때문에, 비행체는 알짜힘이 0인 무중력 상태와 비슷한 상태로써 미세한 모터의 출력에도 매우 민감하게 반응하게 된다. 따라서 대기 및 출발상태 제어 알고리즘에서는 비행체를 마커에 수직인 정 위치로 이동시키는 것을 기본으로 하나 만일 비행체가 특정 방향으로의 속도 또는 회전이 너무 빠르거나 이미 정 위치로 이동 중이라면 추가적인 모터

의 움직임을 최대한 제한하여야 한다. 속도 등 제어에 필요한 값들은 스마트폰의 카메라로 인식한 마커의 위치와 크기 변화로 알 수 있다. 이 전체적인 과정을 의사코드로 표현하면 다음 Fig. 5와 같다.

고도의 변화량을 가장 먼저 검사하는 이유는 비행체의 형태상 고도 조절 모터가 따로 존재하므로 고도의 조정은 현재 위치 및 회전의 변화에 상관없이 독립적으로 조절할 수 있기 때문이다. 때문에 고도 모터의 명령값을 미리 결정해 놓고 위치 및 회전 변화에서 판단한 모터의 명령에 고도 모터의 명령값을 결합하는 형태가 되도록 한다.

6.2 비행 상태 제어 알고리즘

비행 상태에서는 대기상태와 출발상태와 달리 마커를 인식할 수 없는 상태이므로 이 상태에서 비행체가 현재의 위치와 고도를 알아내려면 전혀 다른 알고리즘이 필요하다. 본 연구에서는 컴퓨터 비전 알고리즘 중 하나인 Lucas-Kanade Optical Flow[7]를 응용하여 이러한 문제를 해결한다. Optical Flow는 영상의 특징점을 추적하여 각 특징점이 영상의 프레임마다 어떻게 이동하였는지 알아내는 방법으로 이를 응용하면 비행체의 좌우 변화량과 고도 변화량을 알아낼 수 있다. 따라서 출발상태에서 비행 상태로 넘어가기 직전의 비행체의 고도와 위치를 기준으로 잡고, 비행체가 비행 도중 얼마만큼 그 기준에서 벗어났는지를 측정하여 보정하는 알고리즘을 통하여 비행체가 다음 이탈하지 않고 다음 정점까지 도착하도록 한다.

아래의 Fig. 6은 Optical Flow를 통하여 고도와 좌우 오차를 계산하는 예시를 나타낸 것이다. 좌우 오차의 경우 카메라로 인식한 모든 특징점의 좌우 변화값의 평균을 내어 측정한다. 고도 오차의 경우 영상을 좌, 우로 나누어서 좌측 특징점들의 변화값과 우측 특징점들의 변화값을 측정하면 알아낼 수 있다. 카메라는 항상 비행체의 정 중앙에 위치하므로 특징점들이 중심에서 좌우로 퍼져 나갔다는 것은 영상의 확대, 즉 고도의 하강을 의미하며 반대의 경우 영상의 축소, 즉 고도의 상승을 의미한다. 이렇게 측정한 값들을 누적시키면 비행체가 처음 출발 상태에서 얼마만큼 상하 좌우로 벗어나 있는 지를 측정할 수 있다.

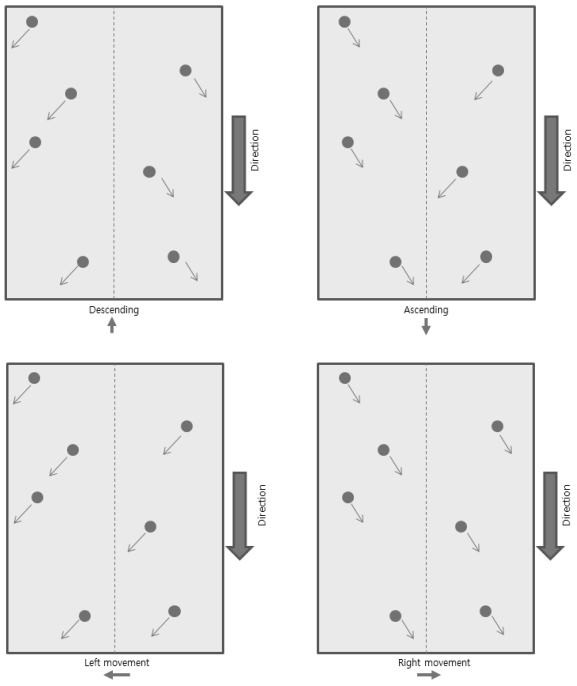


Fig. 6. Use of Optical Flow to correct of error

6.3 층간 이동 알고리즘

비행체가 다른 층으로 이동하기 위해서는 계단을 이용해야 한다. 비행체가 계단을 이용하여 올라가거나 내려갈 시에는 6.2절에서 설명한 Optical Flow를 이용한 고도 계산 알고리즘은 적합하지 않다. 실제로는 비행체의 절대고도가 전혀 변하지 않았다 하더라도 계단이라는 특성 상 비행체와 계단 바닥과의 거리는 점점 멀어지거나 가까워지기 마련인데, 이러한 변화는 영상의 특징점 추적을 통해서는 감지할 수 없는 변화이기 때문이다. 본 연구에서는 이러한 문제점을 각 계단의 폭과 그 변화량을 영상정보에서 알아내어 해결한다. 먼저 4장에서 설명한 데이터베이스의 정보를 이용하여 현재 비행체의 비행경로가 층간이동을 포함하고 있는 경로인지 검사한다. 층간이동을 포함하고 있는 경로라도, 계단을 발견하기 전까지는 6.2절의 알고리즘과 동일하게 비행을 수행한다. 계단의 인식은 Hough 변환을 이용한 직선검출 알고리즘[8]을 기반으로 하여 수행한다. 영상의 가로길이의 80% 이상이며, 각도가 비행체 진행방향의 수직에 3° 이내로 가까운 직선이, 한 화면에 셋 이상 일정한 간격을 두면서 검출 될 경우, 이 때의 영상을 계단을 인식한 영상정보라고 판단한다. Fig. 7은 이러한 계단을 인식한 결과를 나타낸다. 윗 화면은 원본 영상이며 아래 화면은 경계선 검출을 수행한 결과이다. 원본 영상의 적색 직선이 찾아낸 계단을 나타낸다.

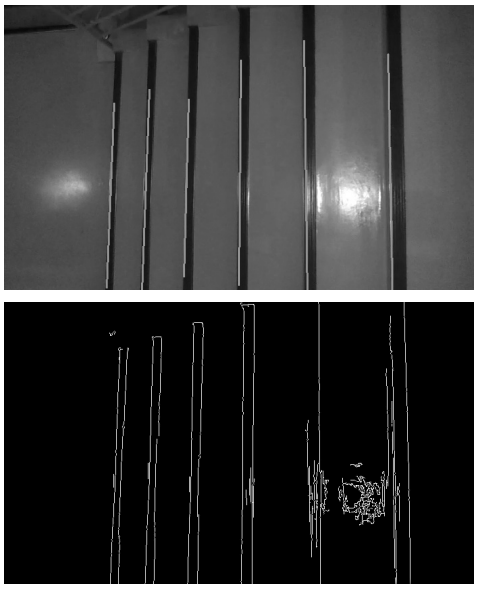


Fig. 7. Result of recognize stairs

일단 계단을 인식한 다음부터는, 처음 인식한 영상정보에서 알 수 있는 계단의 평균 폭을 앞으로의 비행체 고도 조정의 기준으로 한다. 가령 첫 영상에서의 계단 폭 평균이 100px 이었다면, 앞으로 들어오는 영상에서의 계단 폭 평균도 항상 100px을 유지하도록 비행체가 고도를 조절하도록 한다. 이렇게 계단의 폭을 기준으로 고도를 조절하게 함으로서, 비행체는 동일한 고도를 유지한 체 계단을 오르거나 내려갈 수 있게 된다.

하지만 위와 같은 방식은 비행체가 항상 계단을 인식하는데 성공했을 경우에만 통용되는 방식으로서, 이를 그대로 실제에 적용하기에는 어려움이 있다. 비행체는 비행 중 자주 흔들리게 되므로 모든 프레임 정확하게 계단을 인식하는 건 필연적으로 불가능하다. 아예 인식하지 못하거나 잘못된 값을 전달하는 등의 노이즈가 어쩔 수 없이 생기게 되는데, 본 연구에서는 Kalman 필터[9]를 이용하여 이러한 노이즈를 제거한다. 다음의 Fig. 8은 Kalman 필터를 적용한 결과를 나타낸다. 필터에 의해 걸러지는 값들은 오인식 결과로 가정하고 무시하며, 가장 최근에 성공적으로 인식했던 값을 기준으로 계속 비행을 수행한다.

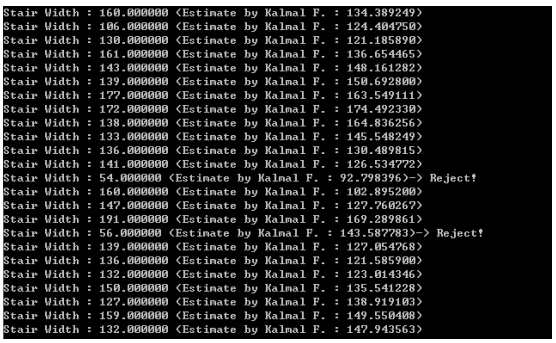


Fig. 8. Result of using Kalman Filter

7. 임베디드 보드 및 비행체의 제작

본 연구내용의 실험을 위해 임베디드 보드를 제작하였다. 메인프로세서로 Atmega128를 사용하였고 L298모터 드라이버를 장착하였다. 스마트폰과의 통신을 위해 블루투스 모듈 ESD-200을 사용하였고 Turnigy 1000mAh 배터리를 이용하여 전원을 공급하였다.

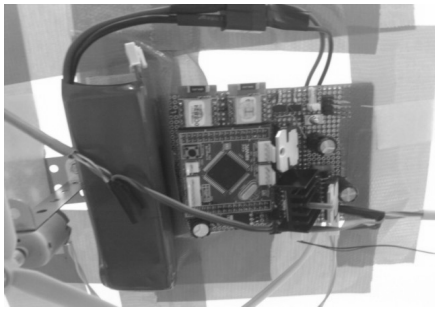


Fig. 9. Embedded Board

제작한 임베디드 보드의 무게는 325g으로서 스마트폰, 기낭, 기타 기체 부품의 무게를 고려하면 비행체는 최소 800g 이상의 무게를 들어 올릴 수 있는 형태로 제작되어야 한다. 2장의 연구결과를 바탕으로 하여 높이 100cm, 너비 90cm, 길이 180cm에 준하는 실험용 비행체를 제작하였다.

Fig. 10. Prototype of flight

8. 실험 및 분석

제작한 실험용 비행체와 본 연구의 알고리즘을 적용하여 비행 성공 여부를 실험한 결과는 아래의 Table 1과 같다. 실험은 숭실대학교 정보과학관 2, 3층에 마커를 설치하고 가상 맵을 제작하여 20번에 걸쳐서 이루어졌다.

Table 1. Result of experiment

State	Measured Content	Success Rate
Waiting	Keep in on the marker above	100%
Starting	Rotate to direction of destination and start successfully	95%
Flight	Arrive next marker successfully	80%

실험 결과 대기상태와 출발상태의 경우 높은 성공률을 보였으나 비행 상태에서는 아직 불안정한 모습을 보였다. 빌딩 내부에 흐르는 기류를 비행체가 완벽히 제어하지 못하는 게 가장 큰 원인이었다.

9. 결론 및 향후 연구

실험 결과에 나타나듯이 아직 완벽한 성공률을 보이고 있지는 않으며 기체의 소형화, 헬륨의 유지력 보완 등 개선해야 할 부분이 남아있다. 특히 바람의 저항에 더욱 강한 형태로 비행체의 형태를 개선하고 알고리즘을 보완할 필요가 있으리라 본다. 그러나 기존의 다른 비행체처럼 자이로센서 등 고가의 센서에 의존하지 않고 영상정보와 스마트폰에 내장된 센서만으로 비행이 가능한 비행체 제작 및 알고리즘 개발은 분명 유의미한 일로써 무인 배송, 무인 방범, 엔터테인먼트 등 다양한 방면으로 활용될 수 있을 것이다.

참 고 문 헌

[1] D. H. Curtis, M. F. Reeder, C. E. Svanberg, R. G. Cobb, G. H. Parker, “Flapping Wing Micro Air Vehicle Bench Test Setup”, International Journal of Micro Air Vehicles, Vol.4, No.1, pp.51-78, 2012.

[2] K. Cohen, “Surveillance for Intelligent Emergency Response Robotic Aircraft (SIERRA Project)”, http://most-aero .uc.edu/projects/sierra-project, 2012.

[3] N. C. Moore, “Sensors for bat-inspired spy plane under development”, http://ns.umich.edu/new/releases/6409, 2008.

[4] F. T. Ulaby, K. Sarabandi, K. McDonald, M. Whitt & M. C. Dobson, “Michigan microwave canopy scattering model”, International Journal of Remote Sensing, Vol.11, Issue 7, 1990.

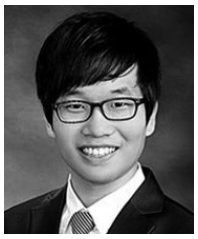
[5] K. W. Yang, “OpenCV Marker Recognition to Augmented reality”, http://blog.daum.net/pg365/217, 2012.

[6] Dijkstra, E. W, “A note on two problems in connexion with graphs”, Numerische Mathematik 1: 269 - 271. doi:10.1007 /BF01386390, 1959.

[7] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision” Proceedings of Imaging Understanding Workshop, pp.121-130, 1981.

[8] Duda, R. O. and P. E. Hart “Use of the Hough Transformation to Detect Lines and Curves in Pictures” Comm. ACM, Vol.15, pp.11-15, 1972.

[9] Kalman, R.E, “A new approach to linear filtering and prediction problems”. Journal of Basic Engineering 82 (1): 35-45. Retrieved 2008-05-03.



최 영

e-mail : fjuhy0@gmail.com

2013년 숭실대학교 컴퓨터학부(학사)

관심분야 : Computer Vision, Game Design

김 계 영

e-mail : gykim11@ssu.ac.kr

1990년 숭실대학교 전산학과(학사)

1992년 숭실대학교 컴퓨터학과(석사)

1996년 숭실대학교 컴퓨터학과(박사)

1996년~현 재 숭실대학교 컴퓨터학과 교수

관심분야 : Computer Vision, Pattern Recognition, Augmented Reality