논문 2013-08-25

실시간 쿼드로터 자율주행과 원격제어 기법

(A Real Time Quadrotor Autonomous Navigation and Remote Control Method)

손 병 략, 강 석 민, 이 현*, 이 동 하*

(Byung-Rak Son, Seok-Min Kang, Hyun Lee, Dong-Ha Lee)

Abstract : In recent, the demand of Unmanned Aerial Vehicles (UAVs) that can autonomous navigation and remote control has been increased in military, civil and commercial field. Particularly, existing researches focused on autonomous navigation method based on vanish point and remote control method based on event processing in indoor environments. However, the existing methods have some problems. For instance, a detected vanish point in intersection point has too much detection errors. In addition, the delay is increased in existing remote control system for processing images in real time. Thus, we propose improved vanish point algorithm by removing detection errors in intersection point. We also develop a remote control system with android platform by separating flying control and image process. Finally, we compare the proposed methods with existing methods to show the improvement of our approaches.

Keywords : Autonomous Navigation, Remote Control, Ar.Drone, Quadrotor, Vanish Point

1. 서론

최근 사람이 직접 접근할 수 없는 재난발생지역, 환경오염지역, 화생방오염지역 등에 헥사로터(Hexarotor), 쿼드로터(Quadrotor), 소형무인항공기(Micro-UAV)와 같은 무인항공기(UAV)를 활용하여 실시간 원격으로 재난피해 정도를 관찰하거나 인명구조작업, 생존자 위치파악 등을 수행하는 연구가 진행되고 있다. 특히 실내 및 실외 환경에서 자율적으로 임무를 수행 할 수 있도록 원격제어와 더불어 자율주행 기법이 연구되고 있다 [1].

일반적으로 실외 환경에서는 GPS 신호를 이용한 주행기법이 사용되고 있으나, 실내 환경에서는

GPS를 사용하기 위한 외부수신기 설치를 필요로 해서, GPS 방식 이외의 연구들이 진행되고 있다 [2, 3]. 또한 실내에서는 소형, 소중량 및 저용량 배터리를 가지고 있는 무인항공기를 사용하는 경우가 많아, 크기가 크거나 전력소모가 큰 센서는 사용하지 않고, 카메라 기능을 활용한 영상센서나 음파로 거리측정이 가능한 초음파 센서를 활용한 자율주행 기법이 연구되고 있다.

카메라를 이용한 영상센서기반의 자율주행 기법으로는 2D 이미지 자료를 받아 3D 환경을 구축하는 Visual SLAM이 있는데, 이와 같은 기법은 실제계의 3D 환경을 표현하는데 적합하나, 상당한 계산시간이 요구되어 실시간으로 동작하는 무인항공기에는 적용하기 어렵다. 또한 2D에서 3D로 변환된 이미지는 무늬가 없는 표면에서는 통로를 인식하기 어려운 문제점을 가지고 있다 [4].

[5]에서는 이런 문제점을 해결하기 위해, 소실점을 활용한 자율주행 알고리즘을 사용했다. 하지만 기존의 방식에서는 소실점 계산에 불필요한 라인 및 교차점을 이용하여 정확한 소실점 계산에 대한 오차 범위가 존재한다. 따라서 본 논문에서는 [5]의 문제점을 해결하기 위한, 향상된 소실점 기반의 쿼드로터(Quadrotor) 자율주행 시스템을 개발하여 그림 1

* Corresponding Author

(mahyun91@sunmoon.ac.kr, dhlee@dgist.ac.kr)

Received: 15 Feb. 2013, Revised: 22 Mar. 2013,

Accepted 12 Apr. 2013.

B.R. Son, D. H. Lee: DGIST

S. M. Kang, H. Lee: Sun Moon University

※ 본 논문은 미래창조과학부에서 지원하는 대구 경북과학기술원 일반과제에 의해 수행되었습니다.

(13-BD-01).

그림 1. 쿼드로터 (Parror사의 Ar.Drone)
Fig. 1 Quadrotor of Parror Inc. (Ar. Drone)

과 같은 Parrot사의 Ar.Drone [6]에 적용하여 시험하고자 한다. Ar.Drone은 주 센서로 2개의 카메라(전면부 및 하단부)가 장착되어 있다. 실험에서 사용한 전면 카메라는 640X480 해상도 및 와이드 앵글(93°)을 지원하는 카메라가 내장되어 있다. 이는 복잡한 실내 환경에서도 저속의 속도로 이동하면서 충분히 영상처리를 수행할 수 있는 단계이다. 그 외 센서로 높이 측정을 위한 초음파 센서가 설치되어 있다. 리눅스가 설치된 onboard 컴퓨터와 WiFi 통신모듈이 내장되어 host 컴퓨터와 WiFi를 연결을 통해 통신 및 명령이 수행을 처리한다.

향상된 소실점 기반의 쿼드로터 자율주행 시스템은 다음과 같은 특징을 가지고 있다. 첫째, 정확한 소실점을 구하기 위하여 영상의 잡음을 제거하고, 둘째, 불필요한 라인을 제거하여 보다 빠른 계산을 수행하고, 마지막으로 라인이 존재하지 않는 영역에 교차점이 나타나는 부분을 수정하여 보다 정밀한 소실점을 구하고자 한다. 그리고 추가적으로 자율주행이 불가능 할 경우를 대비하여, 원격으로 Ar.Drone을 제어하고자 한다. 특히 그림 2와 같이, 1) Parrot사에서 개발된 Java 소스 코드를 안드로이드 개발환경에 변화시키고, 2) 쿼드로터의 비행 제어 코드와 영상처리 코드 부분을 분리시켜, 3) 영상처리 속도를 향상시키고자 한다. 이는 기존 연구 [7, 8]에서 빈번히 발생하는 영상 깨짐이나 지연을 줄여 실시간 원격제어가 가능하도록 하고자 한다.

따라서 본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 소개하며, 3장에서는 자율주행을 위한 향상된 소실점 알고리즘을 제안하고자 한다. 4장에서는 원격제어 기법의 동작순서를 설명하며, 5장에서는 기본방식과 비교분석하고자 한다. 그리고 6장에서 결론 및 향후 연구방향에 대해 설명하고자 한다.

그림 2. Ar.Drone의 시스템 구조
Fig. 2 System Architecture of Ar.Drone

II. 관련연구

실내 환경에서의 자율주행 시스템에 관한 관련 연구는 크게 영상센서의 유무로 나눈다. 그리고 원격제어 기술은 제어하는 플랫폼에 따라 구분한다.

1. 영상센서를 사용하지 않은 자율주행

능동 센서(적외선, Sonar 등)를 사용하는 자율주행 시스템은 광범위하게 사용 중이다. 레이저, 초음파, 적외선 센서를 사용하는 경우[9], 짧은 거리에서는 좋은 성능을 보이나, 넓은 범위의 센서 데이터를 수신받기 어려운 단점이 있다. 거리측정 및 초음파를 사용하는 경우 실내 환경의 재구성 및 장애물과의 거리를 측정이 가능하다[10]. 하지만, 초음파 센서 경우 인식 정확도의 성능이 낮다. 레이저 레이다를 이용하여 실내 환경을 3D 구조로 전환시키는 방법은 높은 정확도를 제공한다. 그러나 단가가 비싸며 장치 크기가 커서 소형 UAV에는 적합하지 않으며 실시간적인 제어를 보장하지 않는다[11].

2. 영상센서를 사용하는 자율주행

영상센서를 이용하여 자율주행 시스템을 구축하는 방법은 낮은 전력 소모 및 저중량으로 넓은 영역의 센싱이 가능하여 많이 쓰이고 있다. 이러한 장점으로 많은 로봇에서 영상센서를 내장하고 있다. 예를 들어, [4]의 경우, 지상로봇에서 소실점을 이용하여 실내에서의 자율주행 시스템을 구축하고 실시간성 보장 및 단일 카메라로 주행하기 때문에 저비용이라는 장점을 가지고 있다. 쿼드로터의 경우는, ROS(Robot

그림 3. 실내환경 안에서의 통로
Fig. 3 Corridor in indoor environment

Operating System) 드라이버와 연계되어 여러 연구가 진행되고 있다 [12-14]. 하지만, [14]의 경우 현재까지 아직 큰 동작만 인식하여 작은 동작을 놓치는 단점이 있다.

3. Ar.Drone을 활용한 실시간 원격제어기술

Ar.Drone을 활용한 대표적인 실시간 원격제어기술은, Parrot사에서 제공하는 OpenApi[15]가 있지만, 실행 할 때마다 에러 발생률이 현저히 높다. [7]의 경우에는 100% 자바로 개발된 프로그램인데 PC 기반에서만 동작되는 단점이 있고 영상처리에 있어서 Java 소스코드 실행을 위한 지연이 빈번히 발생한다. [8]의 경우에는 상대적으로 느린 영상처리뿐만 아니라, 제어기능이 단순히 이착륙 기능밖에 없어서 실시간 쿼드콥터 동작제어를 수행하기에는 한계점을 가지고 있다. 본 연구에서는 이러한 부분들을 수정하여 안드로이드 기반에서 원격제어 동작 가능하고 영상처리 속도를 향상시키고자 한다.

III. 향상된 소실점 기반 자율주행기법

실내 환경에서 통로의 비중은 높은 편이다. 실내 환경에서의 통로는 그림 3과 같이 양면의 벽면부와 천장부 및 바닥부를 가지고 있다. 통로에서 무인항공기의 가장 큰 목표는 안전하게 통로의 끝 방향을 탐색하여 자연스럽게 나아가는 것이다. 따라서 본 논문에서는 무인항공기 환경에서 단일 영상센서를 이용한 자율주행 시스템을 소개하고자 한다. 논문에서 제시하는 소실점 탐색으로 단일 영상센서를 이용하는 방법은 다음과 같다.

먼저 그림 3에서처럼 4면부의 외곽선이 통로 진행방향의 중심점을 향하고 있는데 이를 그림 4과 같이 도식화하고, 외곽선이 교차하는 지점을 소실점으로 정한다.

그림 4. 도식화된 통로 이미지
Fig. 4 Schematized corridor image

그리고 이미지 상에서의 경계값을 검출하기 위해 컬러 이미지 영상을 흑백 이미지로 전환한다. 전환된 이미지는 Canny Edge Detection 방식 [16]을 이용하여 이미지의 경계 값을 구하고, 경계 값이 검출된 후 직선의 라인을 검출하기 위해, Hough Transform의 한 유형인 Probabilistic Hough Transform 방식 [17]을 이용한다. 이는 이미지상의 경계 값이 적더라도 보다 효율적인 성능을 얻는다. 그림 4와 같이 두 직선 이상이 서로 만나는 점이 존재하여야 소실점을 구할 수 있으므로 검출된 라인들의 교차점을 구하여야 한다. 직선 z 는 다음과 같이 수식화 할 수 있다.

$$A_z x + B_z y + C_z = 0 \quad (1)$$

직선 z 의 기울기 $C = (y_2 - y_1)/(x_2 - x_1)$ 을 구한 후 그림 4와 같이 소실점을 구할 때에는 수평과 수직에 관한 계산이 필요하지 않으므로 제외한다. 직선 z 에 대하여 다른 직선 $z+1$ 과 서로 평행한 경우 교차점이 존재하지 않으므로 평행선에 대하여 교차점 연산은 하지 않는다. 직선 z 에 대한 평행선 계산은 다음과 같다. 직선 z 는 시작점과 끝점을 가지며 각각의 점은 $A(x_1, y_1)$, $B(x_2, y_2)$ 로 표기하고, $z+1$ 은 $C(x_3, y_3)$, $D(x_4, y_4)$ 라 하였을 때의 평행선 P 는,

$$P = (y_2 - y_1) * (x_4 - x_3) - (y_4 - y_3) * (x_2 - x_1) \quad (2)$$

평행선의 유무를 확인 후 평행하지 않았을 때, $P(x, y)$ 는 다음과 같은 수식으로 구한다[4].

$$Px = [(x_2 - x_1) * (x_4 - x_3) * (y_3 - y_1) + (y_2 - y_1) * (x_4 - x_3) * x_1 - (y_4 - y_3) * (x_2 - x_1) * x_3] / P \quad (3)$$

$$Py = [(y_2 - y_1) * (y_4 - y_3) * (x_3 - x_1) + (x_2 - x_1) * (y_4 - y_3) * y_1 - (x_4 - x_3) * (y_2 - y_1) * y_3] / P \quad (4)$$

Situation 1

그림 5. 교차점 오류

Fig. 5 Error of an intersection point

마지막으로, 위 수식을 통하여 얻어진 교차점에 대하여 그림 5와 같이, 두 직선이 교차하지 않음에도 교차점으로 계산되므로 본 논문에서는 수식(5)을 통하여 이와 같은 교차점 오류부분을 계산에서 제외시킨다.

$$(x_1 \leq i.x) \wedge (i.x \leq x_2) \wedge (x_3 \leq i.x) \wedge (i.x \leq x_4) \quad (5)$$

예를 들어, i 를 교차점(intersection point)라 하고 i 의 x 좌표를 $i.x$ 이라고 가정 하였을 때, 수식 (5)을 만족시키는 교차선 x_n 가 존재 하며, x_n 는 교차점 오류에 해당하지 않으므로 교차점 평균계산에 포함 시킨다. 그러나 선 x_n 이 x_{n+1} 의 선과 교차하지 않았을 때에는 선 x_n 의 좌표값과 획득한 교차점 $i.x$ 간의 위치를 비교하여 교차점 사이의 두 직선이 교차하지 않음을 확인할 수 있다. 만약 교차점 $i.x$ 이 교차선 x_n 에 존재할 때 이를 교차점으로 인정하여 다음의 평균값 계산을 위해 좌표를 리스트에 저장한다. 그리고 직선 z 에 대한 교차점 $\bar{z}(x, y)$ 에 대하여 평균값을 구한다. 평균값 $\bar{z}$ 에 대하여 좌표에서 벗어난 교차점들은 보다 정확한 소실점을 구하기 위해 평균교차점에서 제외 시켜 평균값 $\bar{z}'$ 를 다시 구한다. 이때 평균값 $\bar{z}'$ 에 인접한 교차점과 평균값을 재계산하여 $\bar{z}'$ 의 좌표를 보정한다. 이런 과정을 사용자 설정에 따른 횟수에 따라 다시 계산시켜 누적시킨다. 일련의 과정을 거친 후 그림 6과 같이 소실점을 얻을 수 있다.

IV. 원격제어기법의 동작순서

먼저 실시간 쿼드로터 원격제어를 하기 위해서 안드로이드 기반에서 Thread level의 제어엔진(그림 2 참조)을 수정하고자 한다.

그림 6. 개선된 소실점 알고리즘 실험

Fig. 6 Test for improved vanish point algorithm

그림 7. 안드로이드 플랫폼 기반 원격제어장치

Fig. 7 Remote Control Device based on Android Platform

예를 들어, 기본 제어방법으로는 키보드를 통한 이벤트 처리방식이었다면, 본 논문에서는 스마트폰의 터치기능과 떨림(Tilting) 기능을 이용하여 Ar.Drone의 비행을 제어하도록 하였다. 그림 7은 안드로이드 기반에 맞게 변화시킨 원격제어장치의 UI 모습을 보여주고 있다.

그리고 기존의 소스코드는 하나의 클래스 내에서 Ar.Drone 제어부분, 영상출력처리부분, 정보수신부분까지 모든 기능이 activity 하나에서 처리되고 있다. 이는 프로그램의 자원의 비효율적인 관리를 초래하고 영상처리속도의 느려지는 효과로 나타나고 있다. 따라서 본 논문에서는 그림 8과 같이 소스코드를 수정 변환함으로써 영상처리부분과 비행 제어부분을 분리함으로써 다른 부분으로 접근을 최

```

public void connect() throws IOException
{
    video_reader = new VideoReader(this.drone_addr, VIDEO_PORT);
    video_reader_thread = new Thread(video_reader, "Video Reader");
    video_reader_thread.start();
}

DEFINE_THREAD_ROUTINE(video_stage, data)
{
    COM_CONFIG_SOCKET_VIDEO(&my_jcc_socket, VP_COM_CLIENT,
    VIDEO_PORT, WIFI_ADRONE_IP);
}

```

그림 8. 영상처리와 비행제어를 분리한 코드의 예
Fig. 8 The example of source code for dividing image processing and flying controlling

```

public void videoFrameReceived( .... )
{
    BufferedImage im = new BufferedImage(w,h,BufferedImage.TYPE_INT_RGB);
    im.setRGB(startX,startY,w,h,rgbArray,offset,scansize);
    image.set(im);
    repaint();
}

void video_render(long tick, int width, int height)
{
    glTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, VIDEO_WIDTH,
    VIDEO_HEIGHT, GL_RGB, GL_UNSIGNED_SHORT_5_6_5, config->data);
}

```

그림 9. JNI를 이용한 Native한 소스코드의 예
Fig. 9 The example of native source code based on JNI

소화시켜 프로그램 및 자원의 효과적 관리를 진행하고자 하였다. 특히 카메라를 통해 이미지 영상을 받을 포트를 독립적으로 열어주어 통신하도록 처리하였다.

또한 영상의 지연이 빈번히 발생하여 실시간 제어가 어려운 문제점을 해결하기 위해, 본 논문은 JNI를 사용하여 그림 9와 같이, 보다 native한 소스 코드 수정을 통해 영상처리 속도(예를 들어, fps: frame per second)를 향상시켰다.

이런 원격제어기법을 기반으로 Ar.Drone의 동작 순서를 진행하면 그림 10과 같이 나타낼 수 있다. 예를 들어, 안드로이드 플랫폼의 애플리케이션이 실행되면, activity가 활성화 되고, 각각의 컴포넌트들이 생성 및 초기화, 이벤트 Listener들의 연결 과정을 거친다. 그리고 난 다음, Ar.Drone과의 연결을 통해 송신/수신 체크를 하고 Ar.Drone로부터 영상, 내비게이션 데이터 들을 수신 받아 통신 상태를 접

그림 10. 원격제어기법의 동작순서

Fig. 10 Activity process of remote control

검한다. 위와 같은 동작이 완료되면 각각의 컴포넌트는 이벤트에 의해 Ar.Drone의 비행제어 또는 영상표현 등을 수행하게 된다. Ar.Drone의 제어가 종료되면, 연결을 끊고 각각 컴포넌트 자원을 해제시킨 후 activity를 종료한다.

V. 실험 및 비교분석

1. 자율주행

개선된 소실점 알고리즘의 성능을 보여주기 위해 Ar.Drone에 내장된 영상센서를 이용하여 실험하였다. 실험 순서는 프로그램 실행 후 Ar.Drone을 이륙시킨다. 그 후 자율주행 기능을 활성화 시켜 통로 이동의 성공률 및 소실점 오차에 대하여 기존의 논문 [4]에서 나온 알고리즘과 비교한다. 실험 기준은 30m 통로에서 이륙 후 측정된 소실점 좌표의 위치를 착륙까지 기록한 것이다. 조도는 전후면부를 제외한 밀폐된 곳에서 형광등만을 사용하였다.

소실점의 절대좌표의 기준을 화면의 가운데 픽셀이라 하였을 때, 실험결과 그림 11과 같이, 기존 알고리즘 [4]의 평균치는 21.023이며 향상된 알고리즘은 22.602로 큰 차이가 없었다.

하지만 표준편차의 경우 기존 알고리즘은 13.602 향상된 알고리즘은 5.641로 차이를 보였다. 특히, 기존 알고리즘의 변화폭이 커 기체의 잦은 이동이 있었으나 향상된 알고리즘을 적용 할 경우, 이에 따른 이동을 줄일 수 있다.

그림 11. 기존 알고리즘[4]과 개선된 알고리즘 비교
Fig. 11 Comparison of existing algorithm[4] with improved algorithm

그림 12. 통신환경에 따른 영상처리 속도 비교
Fig. 12 Comparison of image processing depending on communication environments

2. 원격제어

java나 Android 소스에서 화면을 출력하는 부분에 대해 타임스탬프를 잡아서 영상처리속도를 측정해보았다. 하지만, 그림 12와 같이 영상처리속도가 JavaDrone은 평균 16의 값, FusionDrone은 평균 8의 값, FRD (제안된 퓨전제어기법)는 평균 60의 값으로 모든 상황에서 비슷한 값을 보여주고 있다.

따라서 실시간 원격제어 성능을 평가하기 위하여, 실험조건에 따라, 실내에서 무선통신기기의 영향을 받지 않는 상태라는 가정을 두었다. 제안된 원격제어기법을 통한 영상처리속도 (fps: frame per second)와 다른 제어기법 [7, 8]과의 영상처리속도 (fps)를 그림 13와 같이 비교하였다.

예를 들어, 먼저 PC 기반의 JavaDrone[7]은 평균 16 fps로 영상을 보여준다. 하지만 PC기반에서 안드로이드기반으로 바꾼 FusionDrone[8]은 평균 10 fps 미만으로 영상의 속도가 현저하게 떨어진 것을 볼 수 있다. 반면에 안드로이드 기반에서 제안된 제어기법 (FRD)은 fps가 기존의 FusionDrone과 JavaDrone보다도 훨씬 높아진 것을 보여주고 있다.

그림 13. 기존 알고리즘[7, 8]과 개선된 알고리즘 비교
Fig. 13 Comparison of existing algorithm[7, 8] with improved algorithm

이를 통해 본 논문에서 제안한 원격제어기법이 기존의 제어기법보다 성능이 우수함을 알 수 있다.

IV. 결 론

기존의 알고리즘은 소실점의 변동폭이 커 Ar.Drone 좌우 이동 변동이 있으며 잘못된 값으로 좌측 혹은 우측으로 이동하는 경향이 있었다. 본 논문에서 제시한 개선된 소실점 알고리즘 역시 좌우 이동경향이 있으나 오차범위는 낮았다. 그러나 영상센서만을 사용하여 소실점을 찾는 알고리즘은 빛의 영향을 많이 받아 빛이 없는 곳에서는 이용이 불가능하였다. 또한 동적인 장애물에 의해 소실점의 위치가 기체가 지향해야할 방향과는 다른 방향으로 이동하였다. 향후 연구에서는 이러한 문제점을 개선하고자 영상센서 및 빛에 민감하지 않은 센서를 이용하고 동적인 장애물 회피 등 자율주행 시스템을 개발할 예정이다.

또한 자율주행이 되지 않는 상황에 대비한 실시간 원격제어기법의 경우, 쿼드로터의 영상처리와 비행제어 부분을 분리하여 기존 방식보다 영상처리속도를 향상시켰다. 이를 통해 실험에 사용된 Ar.Drone뿐만 아니라, 다른 기종의 쿼드로터나 무인항공기를 원격제어 할 수 있을 것으로 예상된다. 특히, 자율주행기법과 함께 적용될 경우, 사람이 직접 수행 할 수 없는 많은 재난방재지역에서 사용가능하다고 본다. 향후 연구에서는 이런 기능을 가진 여러 쿼드로터들이 서로 협력하여 업무를 수행 할 수 있도록 자율군집제어 기능을 개발하고자 한다.

References

- [1] R.S. Pressman, "Software Engineering A Practitioners' Approach," 6th Edition, McGraw Hill, New York, 2005.
- [2] N. Dijkshoorn, A. Visser, "Integrating sensor and motion model to localize an autonomous Ar.Drone," Intl. Journal of Micro Air Vehicles, Vol. 3, No. 4, pp.183-200, 2011.
- [3] C. Bills, "Autonomous MAV flight in indoor environments using single image perspective cues," Proceedings on IEEE Intl. Conference of Robotics and Automation, pp.5776-5783, 2011.
- [4] K. Celik, S. Chung, M. Clausman, A. Somani, "Monocular vision SLAM for indoor aerial vehicles," Proceedings on IEEE Intl. Conference of Intelligent Robots and Systems, pp.1566-1573, 2009.
- [5] Z. Zhou, T. Chen, D. Wu, C. Yu, "Corridor navigation and obstacle distance estimation for monocular vision mobile robots," JDCTA: Intl. Journal of Digital Content Technology and its Applications, Vol. 5, No. 3. pp.192-202, 2011.
- [6] "Parrot Ar.Drone", the resource available at <http://ardrone.parrot.com> [online].
- [7] "JavaDrone", the resource available at <http://code.google.com/p/javadrone/2011> [online].
- [8] "FusionDrone", the resource available at <https://projects.ardrone.org> [online].
- [9] J. Roberts, T. Stirling, J. Zuffery, D. Floreano, "Quadrotor using minimal sensing for autonomous indoor flight," Proceedings on the European Micro Air Vehicle Conference and Flight Competition, 2007.
- [10] R. Carelli, E. Freire, "Corridor navigation and wall-following stable control for sonar-based mobile robots," Robotics and Autonomous Systems, Vol. 45, No. 4, pp.235-247, 2003.
- [11] S. Se, D. Lowe, J. Little, "Vision-based mobile robot localization and mapping using scale-invariant features," Proceedings on IEEE Intl. Conference of Robotics and Automation, pp.2051-2058, 2001.
- [12] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheller, A. Ng, "ROS: an open-source Robot Operating System," Proceedings on IEEE Intl. Conference of Robotics and Automation workshop on open source software, 2009.
- [13] "Robot Operating System Documentation", the resource available at <http://www.ros.org/> [online].
- [14] W.S. Ng, E. Sharlin, "Collocated interaction with flying robots," Proceedings on IEEE Intl. Conference of RO-MAN, pp.143-149, 2011.
- [15] S. Piskorski, N. Brulez, "AR.Drone Developer Guide," Revision SDK 1.6, 2011.
- [16] J. Canny, "A computational approach to edge detection," IEEE Transactions on. Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, pp.679-698, 1986.
- [17] N. Kiryati, Y. Eldar, A.M. Bruckstein, "A probabilistic Hough transform," Pattern Recognition, Vol. 24, No. 4, pp.303-316, 1991.

저 자 소 개

손 병 략

2000년 대구대학교 정보통신공학부 학사.

2005년 대구대학교 정보통신공학과 석사.

2009년 대구대학교 정보통신공학과 박사.

현재, DGIST 웰니스융합연구센터 선임 연구원

관심분야: 실시간 센서네트워크, 통신 프로토콜, 생체모방 로봇, 웰니스 휴먼케어 시스템.

Email: brson@dgist.ac.kr

강 석 민

2012년 선문대학교 컴퓨터공학과 학사

관심분야: 임베디드 S/W 시스템, 자율주행 시스템, 로봇원격제어

Email: seokmin.kang@gmail.com

이 현

1998년 선문대학교 전자계산학과 학사.

2002년 선문대학교 전자계산학과 석사.

2010년 University of Texas at Arlington 컴퓨터공학과 박사.

현재, 선문대학교 컴퓨터공학과 조교수

관심분야: 실시간 의사결정 시스템, 자율컴퓨팅, 웰니스 휴먼케어 시스템

Email: mahyun91@sunmoon.ac.kr

이 동 하

1985년 경북대 전자공학과 학사.

2001년 경북대 전자공학과 석사.

2005년 경북대 전자공학과 박사.

2006년 경북대 경영학과T-MBA 과정 수료.

현재 DGIST 웰니스융합연구센터 센터장/책임연구원

관심분야: 임베디드 S/W 시스템, 생체모방 로봇 시스템, 웰니스 휴먼케어 시스템

Email: dhlee@dgist.ac.kr