

영상처리와 거리센서를 융합한 객체 추적용 드론의 연구

양우석*, 천명현*, 장건우*, 김상훈*
*국립한경대학교 전기전자제어공학과
e-mail: kimsh@hknu.ac.kr

A Study of Object Tracking Drones Combining Image Processing and Distance Sensor

Woo-Seok Yang*, Myung-Hyun Chun*, Gun-Woo Jang*, Sang-Hoon Kim*
*Dept of Electrical, Electronic and Control Engineering, Hankyong National University

요 약

드론의 대중화에 따른 사고위험의 증가로 안전한 조종 방법에 대한 연구의 필요성이 대두되었다. 따라서 조종자의 조종능력에 구애받지 않는 자율비행제어기술이 필요하게 되었고, 이를 보다 안정적으로 구현하기 위하여 자율주행 소프트웨어 플랫폼으로 주목받고 있는 Robot Operating System(ROS)를 사용하였다. ROS를 기반으로 Laser Range Finder(LRF)와 Particle Filter를 사용하여 자율적으로 객체추적이 가능하며 지능적으로 장애물을 회피하여 비행 할 수 있는 안정적인 자율비행제어시스템을 구현하고자 한다.

1. 서론

드론의 제어 방법은 크게 원격-수동비행제어, 자율비행제어로 나눌 수 있는데 드론이 대중화되면서 송수신기 및 컨트롤러를 이용한 원격-수동비행제어 방식의 저가 드론을 주변에서 쉽게 접하게 되었다.

그러나 원격-수동제어방식은 조종자의 조종능력이 비행 안정성에 큰 영향을 주게 되며, 조종자의 조종능력이 부족한 경우 돌발 상황에서의 조종미숙으로 인한 충돌 및 추락으로 인해 인명·재산 피해가 우려된다.

그러므로 본 연구에서는 영상기반 객체 추적 기법과 Laser Range Finder (LRF)를 이용한 거리 측정 기능을 이용하여 조종자의 조종능력에 구애 받지 않고 안정적인 비행을 할 수 있는 자율비행제어기술을 확보하고자 한다.

현재 많은 분야에서 자율주행 소프트웨어 플랫폼의 기반 시스템으로 Robot Operating System (ROS)을 채택하고 있다. ROS는 다양한 데이터 종류와 통신 경로를 제공하는데에 적합한 TCP/IC 기반의 Publisher-Subscriber 통신 모델을 제공하고 있으며 개발을 위한 시각화 도구와 물리 시뮬레이션 도구 및 디버깅 도구 등의 다양한 개발 도구를 제공한다. 또한 다년간 축적된 방대한 로봇 및 센서 라이브러리를 제공하기 때문에 자율 주행 소프트웨어의 개발에 유리하다.

다양한 드론에 알고리즘을 적용하기 위해 리눅스와 ROS가 설치된 카드 크기의 임베디드보드(HardKernel社, Odroid-XU4)와 웹캠을 사용하여 최소한의 크기로 드론의 자율주행을 구현 할 수 있도록 하였다.[그림1]



그림 1 Embedded Board(Odroid)와 LRF 및 WebCam

2. LRF를 활용한 Camera Vision System

2-1 Robot Operating System의 기본 작동 원리

ROS의 구성요소에는 크게 Master와 Package, Node, Topic이 있다. 각 기능을 담고 있는 Package를 설치 혹은 작성하게 되면 기능을 수행하는 Node가 실행되게 되고 Node간의 Publish/Subscribe를 통하여 Topic을 주고받게 되는 것이 Publisher-Subscriber 통신 모델이다.

Master 노드는 ROS 구동 시 생성되며 각 노드가 생성될 때마다 그 정보를 수신하여 테이블을 만든다. 이 테이블은 Subscriber 노드가 구독하고자 하는 Publisher 노드의 정보를 알려주게 되며 이러한 정보를 바탕으로 Subscriber Node와 Publisher Node간에 TCP접속 생성 후 데이터를 주고 받게 된다.

이러한 구조를 이용하여 웹캠에서 받은 영상 스트림과 LRF의 데이터를 Subscribe하는 Node를 작성하고 수신한 데이터를 처리 하여 결과영상 및 데이터를 다시 Publish하게 된다.

2-2. 객체 추적을 위한 영상처리 알고리즘

2-2-1. 마커(Marker)의 위치 검출을 위한 전처리

객체 검출 및 추적 방법으로 먼저 TemplateMatching을 사용하여 마커(Marker)를 검출하려는 시도를 해 보았으나 화면 내 모든 픽셀에 대하여 Template에 대한 연산을 실시하게 되므로 드론 제어에 사용가능한 만큼의 실시간 처리가 불가능하였다. 따라서 추적 알고리즘인 OpenTLD 알고리즘과 Particle Filter 알고리즘을 TemplateMatching과 결합한 후 각각의 성능을 비교하여 보고 적합한 방법을 선정하기로 하였다.

먼저 마커를 검출하기 위한 TemplateMatching에 앞서 정확도를 높이기 위한 전처리를 실시한다. 현재 사용 중인 마커는 색 정보와 엣지정보를 이용하여 마커 검출이 보다 용이하도록 설계되었다. 따라서 색 정보의 왜곡을 막기 위하여 조명의 변화에 강건하며 인간의 시각 시스템과 유사한 HSV(Hue-Saturation-Value)색공간으로 영상을 변형 후 기존의 Red, Blue 에 해당하는 HSV 값으로 각각 이진화 영상을 만들고 원본영상의 엣지 정보를 추출하여 Red 및 Blue 이진화 영상과 엣지 영상을 더해준다.[그림2] 이후 윤곽선 검출을 이용하여 영상 내의 사각형을 찾아 각 사각형에서 추출한 이미지와 마커의 원본이미지를 TemplateMatching하여 마커의 위치를 찾게 된다.



그림 2 더해진 이진화 영상과 엣지영상

2-2-2. OpenTLD추적기를 이용한 추적 방법

TemplateMatching을 시행하여 찾아낸 마커의 위치를 사용하여 마커의 움직임을 추적하는데, 먼저 오픈소스 추적기인 OpenTLD(Tracking-Learning-Detection)알고리즘을 적용하여 보았다. OpenTLD 알고리즘은 ‘Positive - Negative’ 학습을 이용한 간단한 머신러닝을 사용하여 추적하고자 하는 관심영역의 특징을 학습하고 그것을 통하여 영상 내 특정 영역 위주로 관심영역을 추적하게 된다. 따라서 모든 프레임에서 전체픽셀에 대해 연산을 실시하는 TemplateMatching보다는 빠른 속도로 처리가 가능하다. 하지만 임베디드보드에서 실행하였을 때의 전체적인 처리속도가 드론 제어에 적용 가능할 만큼 빠르지 못하였

다.

따라서 OpenTLD 알고리즘을 Particle Filter알고리즘으로 변경하여 속도를 높이고자 하였다.

2-2-3. Particle Filter 를 이용한 추적 방법

Particle Filter는 입력으로부터 동적시스템의 상태를 예측하는 Recursive Bayesian estimation를 사용하여 비선형 추정문제를 해결하는 알고리즘중 하나이다.

Particle Filter는 예측, 갱신, 추정, 재분배의 네 단계를 거치게 되는데 먼저 예측 단계는 주어진 초기 상태로 초기 확률밀도함수를 근사화하여 임의의 입자들의 집합을 생성 후 이 집합의 초기 확률밀도함수 $P(x_0)$ 를 추정한다. x_k^i 는 입자 i 를 의미하고 w_k^i 는 시간 k 에서의 잠음을 나타내며 각 입자는 고유한 전이함수를 가진다.

$$P(x_k) = \frac{1}{N} \sum_{i=0}^N \delta_{x_k^i}(x_k) \quad \dots \text{식 (1)}$$

여기서 δ_x 는 Dirac 측정을 나타내고 N 은 파티클의 개수 이다. 예측 후 얻어진 관측치를 이용하여 각 입자의 가중치가 갱신된다.

ρ_k^i 를 시간 k 에서의 입자 i 의 가중치라고 한다면 가중치는 다음과 같이 계산된다.

$$\rho_k^i = \frac{P(z_k|x_k^i)}{\sum_{j=1}^N P(z_k|x_k^j)} \rho_{k-1}^i \quad \dots \text{식 (2)}$$

입자의 가중치는 우도에 기초하여 가장 가능성이 높은 상태에 대응하는 입자에 다른 입자보다 더 큰 가중치가 부여된다.

갱신 함수는 다음과 같이 정의된다.

$$P(x_k|Z_k) = \sum_{i=1}^N \rho_k^i \delta_{x_k^i}(x_k) \quad \dots \text{식 (3)}$$

이것을 이해하기 위하여 Particle Filter의 기본이 되는 Recursive Bayesian estimation를 살펴보면 예측과 갱신단계로 이루어져 있는 것을 알 수 있고

$$P(x_k|Z_{k-1}) = \int P(x_k|x_{k-1})P(x_{k-1}|Z_{k-1})dx_{k-1} \quad \dots \text{식 (4)}$$

위 식을 통하여 $k=1$ 부터 $k=k-1$ 까지의 측정을 통하여 $k=k$ 인 때의 시스템 상태를 예측할 수 있다.

이후 k 번째의 측정이 이루어 졌을 때의 시스템 상태 $P(x_k|Z_k)$ 를 Bayes Rule을 사용하여 구해보면 은닉마르코프모델에 의해 다음과 같은 식으로 나타 낼 수 있다.

$$P(x_k|Z_k) = \frac{P(z_k|x_k)P(x_k|Z_{k-1})}{P(z_k|Z_{k-1})} \quad \dots \text{식 (5)}$$

움직임모델을 이용하여 다음 상태를 예측하는 시간 업데이트 과정을 나타낸다.

$$P(z_k|Z_{k-1}) = \int P(z_k|x_k)P(x_k|Z_{k-1})dx_k \quad \dots \text{식 (6)}$$

또한 위 식은 $P(x_k|Z_k)$ 가 x_k 에 대해 전 구간 적분했을 때 크기가 1이 되어 확률밀도함수임을 보여준다.

추정 단계에서 추정량 \hat{x}_k 는 입자의 가중치 합에 의해 다음과 같이 계산된다.

$$\hat{x}_k = \sum_{i=1}^N \rho_k^i x_k^i \quad \dots \text{식 (4)}$$

이후 재분배를 통하여 새로운 입자 집합을 생성하고 각각의 새로운 입자에는 $1 / N$ 의 새로운 가중치가 할당된다. 이것을 통하여 가중치가 축적되어 전체 입자가 하나로 합쳐지는 것을 방지한다.

본 논문에서는 Template Matching을 이용하여 영상 내 Marker의 좌표를 찾아 Particle Filter의 초기상태를 설정해준다. 이후 Particle Filter는 관심영역(Marker)의 3차원 색상 히스토그램을 구하여 초기화 되고 추적을 시작하게 된다. 이후 파티클 필터의 각 단계를 반복하며 객체를 추적하게 되고 추적의 정확도가 떨어진다고 판단 될 때에는 다시 Template Matching 과정을 거쳐 정확한 Marker의 위치로 Particle Filter를 다시 초기화 하여 반복해 준다. [그림3][그림4]

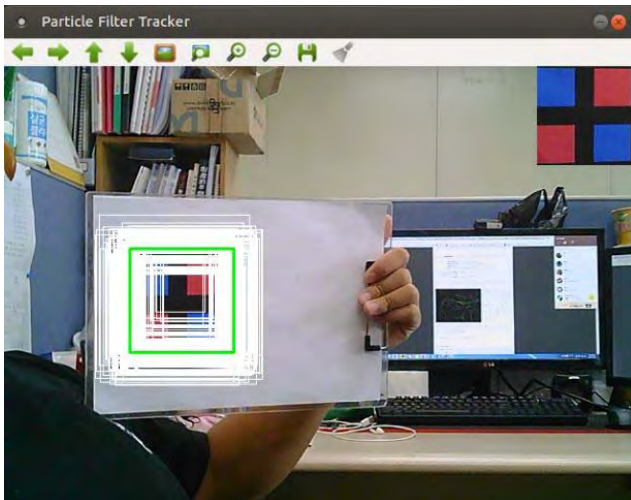


그림 3 Particle Filter 동작 영상 및 객체 추적 영상

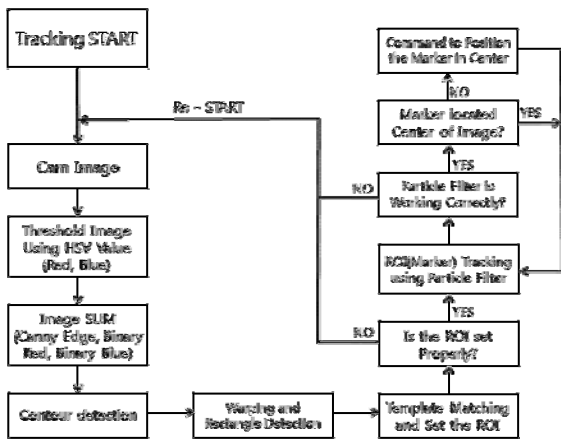


그림 4 Template Matching과 Particle Filter를 사용한 영상처리 알고리즘 Flow Chart

2-3. Laser Range Finder를 사용한 객체와의 거리측정

LRF는 전방 240° 범위의 측정범위(0.2cm-560cm)내의 모든 거리를 측정하게 되며 초당 10회의 속도로 거리측정 데이터가 갱신된다. 각 거리정보에는 센서를 기준으로 한 각도와 거리가 저장되므로 이것을 통하여 영상 내에서 검출된 마커와의 거리 및 방향을 판단하는 것이 가능하다.

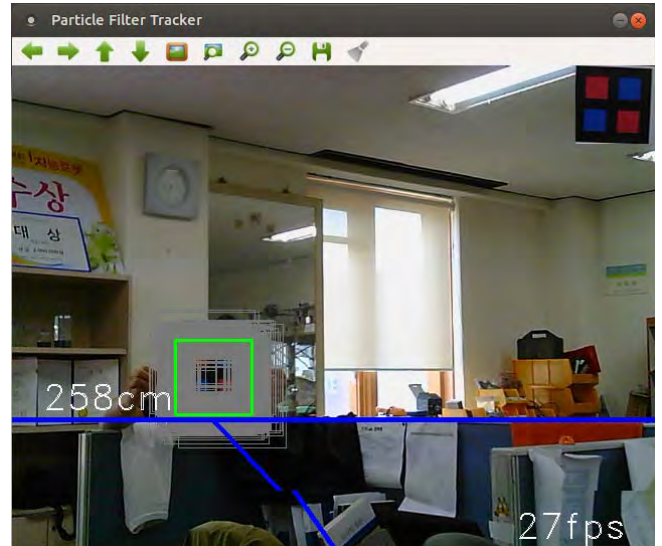


그림 5 Particle Filter동작 및 LRF를 이용한 거리측정 영상

3. 객체추적 적합성 실험환경 및 실험 결과

Hardware	
Embedded Board	Odroid XU4 (HardKernel 社)
AP	Samsung Exynos5422 (CortexA15 2Ghz and Cortex-A7 Octa core big.LITTLE processor)
Memory	2GByte LPDDR3 RAM
Software	
운영체제	Ubuntu 16.04.2 LTS
ROS	ROS Kinetic 1.12.5

표 1 객체추적 알고리즘 구동 환경

임베디드보드인 Odroid 상에서 영상처리를 통하여 안정적으로 드론을 조작하기 위해 빠른 속도로 영상의 처리가 가능해야 하며 실시간 처리가 가능하여야 하므로 초당 프레임 수를 각 알고리즘의 성능평가 지표로 삼았다.

가장 기본적인 검출 알고리즘인 템플릿 매칭을 통한 마커 검출방법과 템플릿 매칭과 OpenTLD 알고리즘을 사용한 방법, 템플릿 매칭과 Particle Filter를 사용한 방법을 비교하여 가장 적합한 방법을 선정하도록 하였고 Particle Filter의 Particle 개수는 각각 200, 250, 300개로 변경하여 실험하였다.



그림 6 Template Matching 그림 7 OpenTLD 그림 8 Particle Filter

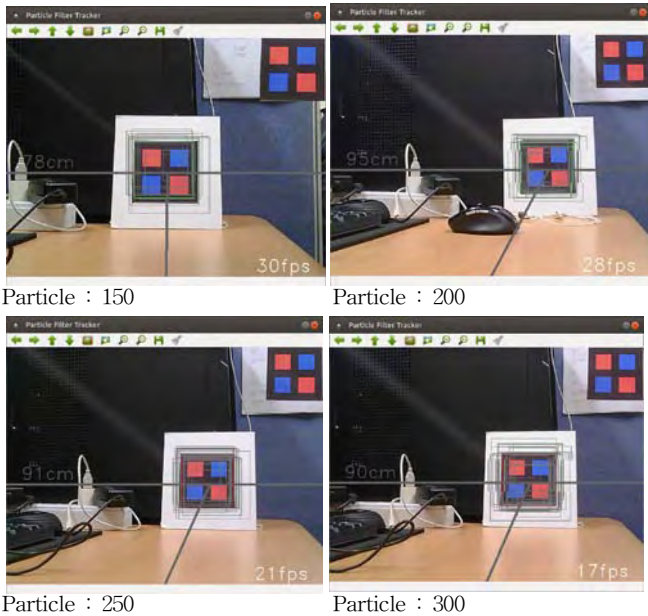


그림 6 파티클 개수 별 FPS

각각의 객체 검출 및 추적 방법을 비교하여 본 결과 하면 내 모든 픽셀에 대하여 TemplateMatching을 실시하는 방법은 약 3 FPS의 처리속도를 보여주었다. 3FPS의 속도로는 실시간 추적이 불가능하고 마커 이동에 따른 드론의 제어가 불가능하므로 보다 속도를 향상시킨 방법을 사용해야 한다는 것을 알 수 있었다.

	Template Matching	OpenTLD	Particle Filter
마커 검출	3 FPS	3 FPS	3 FPS
검출영역 추적	3 FPS	7 FPS	27 FPS

표 2 객체 추적 시 알고리즘 별 초당 프레임 수

Number of Particles	150	200	250	300
FPS	30 FPS	28 FPS	21 FPS	17 FPS

표 3 객체 추적 시 파티클 별 초당 프레임 수

OpenTLD와 Particle Filter모두 Marker인식을 위해 TemplateMatching을 사용하게 되므로 ROI설정 전까지 약 3 FPS의 처리속도를 보여주지만 관심영역 설정 이후에는 보다 빠른 처리속도로 영상처리가 가능하였다. 먼저 OpenTLD는 마커 추적시 약 7 FPS의 속도로 추적이 가능하였

으나 실시간 추적에 부적합한 속도로 제어신호를 전송하게 되므로 비행 안정성이 떨어지는 것이 예상되었다.

Particle Filter를 사용한 추적시에는 약 27 FPS의 처리속도를 보여주었고 가장 실시간 처리에 가까운 성능을 보여주는 것을 확인 할 수 있었다. 또한 Particle의 개수에 따라 처리속도 및 정확도가 달라졌는데 Particle의 개수 150개 일 때에는 처리속도가 빠른 반면 정확도가 떨어졌고 파티클의 개수가 많아질수록 정확도는 높아지지만 속도가 느려지는 것을 알 수 있었다. 결과적으로 Particle의 개수가 200개일 때 가장 효율적인 성능을 보여 주었다.

4. 결론 및 향후 개발계획

본 논문에서는 드론의 자율주행을 위한 비전기반 객체 추적 알고리즘을 제안하기 위한 연구를 진행하였다. 결과적으로 마커의 인식이 완료된 이후에 Particle Filter를 사용하는 것이 드론의 제어에 가장 적합하다는 결론을 얻을 수 있었다. 또한 LRF를 이용하여 객체와의 거리를 판단하는 방법을 사용하여 추적하고자하는 마커와 일정한 거리를 유지 할 수 있도록 하였다.

하지만 Particle Filter 추적의 정확도가 Template Matching에 비해 떨어지고 마커가 움직이는 속도가 빠르면 파티클 필터가 리셋되는 문제점이 있었다.

추후 연구에서 Particle Filter를 보다 정확하고 빠른 객체 추적이 가능하도록 개선하고 LRF의 거리정보를 이용한 장애물 회피 및 실내주행시 지도작성 알고리즘을 추가하여 조종자의 조종능력에 구애 받지 않는 완전한 자율주행 알고리즘을 완성하는 것을 목표로 하고 있다.

감사의 글

이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임(No. 2015R1D1A1A01057518)

참고문헌

- [1] 양범준, 노순현, 홍성수. (2016). ROS의 기능 안전성 향상을 위한 Deadline Supervising 매커니즘 개발. 한국정보과학회 학술발표논문집, 72-74.
- [2] 강덕찬, 김용현, 이덕진. (2016). 멀티로터 드론의 영상기반 사물추적 및 제어 기법 연구. 대한기계학회 춘추학술대회, 87-89.
- [3] 나인섭, 김수형, 이귀상, 김영철. (2016). 파티클 필터 기반 얼굴추적을 위한 효율적 파티클 수과 윈도우즈 크기 분석. 한국콘텐츠학회 종합학술대회 논문집, 401-402.
- [4] 선은혜, 트랜후루엣트, 김동연, 김용태. (2015). 미니드론의 영상기반 자동 비행 제어에 관한 연구. 한국지능시스템학회 논문지, 25(6), 536-541.
- [5] Mai Thanh Nhat Truong, Sanghoon Kim, (2017). Parallel implementation of color-based particle filter for object tracking in embedded systems, Human-centric Computing and Information Sciences December 2017, 7:2.