

무인 비행체의 환경 인지 및 경로 계획 연구동향

Research Trends on Environmental Perception and Motion Planning for Unmanned Aerial Vehicles

홍유경 (Y. Hong, youkyh1@etri.re.kr)

김유경 (Y. Kim, yk.kim@etri.re.kr)

김수성 (S. Kim, suseongkim@etri.re.kr)

이희수 (H. Lee, heelee@etri.re.kr)

차지훈 (J. Cha, jihun@etri.re.kr)

무인자율운행연구그룹 연구원

무인자율운행연구그룹 선임연구원

무인자율운행연구그룹 선임연구원

무인자율운행연구그룹 책임연구원

무인자율운행연구그룹 책임연구원/그룹장

ABSTRACT

Currently, the use of unmanned aerial vehicles (UAVs) is spreading from recreational purposes to the public- and commercial-use product areas. Various efforts are being made worldwide to ensure the safety of UAVs and expand their service applications and convenience, because autonomous flights are becoming increasingly popular. In order for a UAV to perform autonomous flight and mission without operator assistance, environmental perception technology, path planning technology, and flight control technology are needed. In this article, we present recent trends in these technologies.

KEYWORDS 무인 비행체, 환경인지, 경로계획

1. 서론

‘드론’으로 널리 알려진 비행 장치는 무인 항공기, 무인 비행체 등의 다양한 용어로 혼용하여 사용하고 있다. 이는 원격 조종이나 자율 비행으로 시계 밖 비행이 가능하며, 승객이나 승무원을 운송하지 않는 비행체로 정의된다. 무인 비행체는

1900년대 중반에 군사용 정찰 임무를 시작으로 기술의 발전에 따라 후반부에는 공격용으로 활발하게 활용되기 시작했다. 2000년부터는 부품의 경량화·저가화로 인해 민간 부문에서도 다양한 산업과 레저 분야에서 활용되고 있다. 조작의 편의성과 경제성으로 인해 현재 방송 및 영화 촬영, 사람의 접근이 어려운 산업현장의 진단, 측량, 재난, 재해,

* DOI: 10.22648/ETRI.2019.J.340305

* 본 논문은 한국전자통신연구원 연구개발지원사업의 일환으로 수행되었음[2017-0-00067, 안전한 무인이동체를 위한 ICT 기반기술 개발].



본 저작물은 공공누리 제4유형

출처표시+상업적이용금지+변경금지 조건에 따라 이용할 수 있습니다.

©2019 한국전자통신연구원

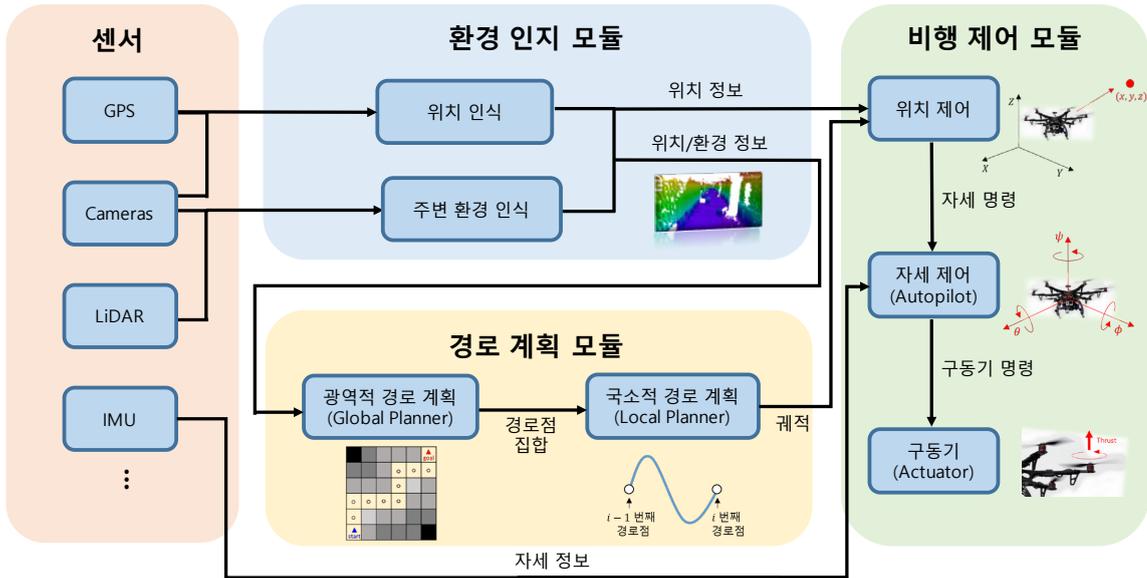


그림 1 무인 비행체의 자율 비행을 위한 환경 인지 및 경로 계획 구성도

구호 등 다양한 분야에서 활용도가 높아지고 있다. 하지만 이러한 수요에도 현재로서는 자율 비행 중의 안전을 담보하지 못해 자율 비행 기반의 무인 비행체의 활용에 많은 제약이 따르고 있다.

무인 비행체가 오퍼레이터의 도움 없이 안전한 자율 비행을 하기 위해서는 그림 1에 도시한 바와 같이 환경 인지 모듈, 경로 계획 모듈, 비행 제어 모듈이 필요하다. 환경 인지 모듈에서는 카메라 및 Light Detection and Ranging(LiDAR) 센서 등을 이용하여 주변 환경 정보를 획득하고 이를 결합하여 비행 경로상의 장애물을 탐지한다. 경로 계획 모듈에서는 경로상에 장애물이 탐지될 경우 상태 정보(무인 비행체의 위치, 자세, 속도, 각속도 등)를 이용하여 새로운 회피 경로를 생성한다. 비행 제어 모듈에서는 주어진 경로를 정확하게 따라가게 하기 위한 위치 제어와 비행 시 안정된 자세를 유지하기 위한 자세 제어를 수행한다.

대부분의 무인 비행체는 기본적으로 경로 계획 모듈, 비행 제어 모듈을 가지고 있다. 그러나 안

전한 자율 비행을 위해서는 주변 환경을 인지하고 이에 대응하는 기능이 필요하므로 환경 인지 모듈이 추가적으로 요구된다. 또한 경로 계획 모듈도 조종자가 입력한 사전 경로를 추종하기 위한 단순 제어를 넘어, 환경 인지 모듈의 환경 정보와 무인 비행체의 운동 제어를 고려한 고차원적인 경로 제어가 요구된다. 본 고에서는 일반적으로 널리 쓰이는 비행 제어기에서 동작할 수 있는 안전한 자율 비행을 위한 환경 인지와 경로 계획 분야의 최신 기술 동향을 분석한다. 따라서 본 고에서는 비행 제어 기술에 대해서는 별도로 분석에 포함하지 않았다. II장에서는 환경 인지 기술동향에 대해서 기술하고, III장에서는 경로 계획 기술에 대하여 기술한다.

II. 환경 인지 기술

본 장에서는 자율 비행을 위한 환경 인지 기술을 분류하고 각 기술의 장단점을 기술하고자 한

다. 환경 인지란 무인 비행체 자신을 기준으로 주위 환경을 이해하는 것을 말한다[1]. 여기에는 공중, 지면, 해양과 같은 자연환경이나 실내, 도심과 같은 복잡한 인공적인 환경이 포함될 수 있다. 자율 비행을 위해 필요한 환경 인지는 자신의 정확한 위치를 파악하는 것과 장애물을 포함한 주변 환경을 인식하는 것으로 구성된다. 이러한 환경 인지를 위해 무인 비행체는 다양한 외부 센서들의 정보를 이용한다.

1. 위치 추정 기술

자율 비행을 위해서는 자신의 절대적 위치를 추정하고 주위 환경의 변화를 감지할 수 있어야 한다. 이러한 절대적 위치 추정 기술에는 Global Navigation Satellite System(GNSS) 기반 측위와 영상 정합 방법이 있다.

가. GNSS 기반 측위

무인 비행체는 자신의 위치를 측정하는 데 기본 항법 센서들을 이용한다. GNSS 기반 측위 기술은 지구 주위 궤도를 돌고 있는 인공위성 시스템을 이용해 다수의 위성이 발신하는 전파를 이용해 물체의 위치를 측정한다. 수신된 위성의 위치 정보와 전파를 이용한 거리 측정을 통해 지구상에서 3차원의 위치를 계산한다. GNSS 시스템으로는 미국의 Global Positioning System(GPS), 러시아의 GLONASS, 유럽의 Galileo가 대표적이다. 정밀도는 위성 신호 수신 환경에 따라 수 m에서 수십 km까지 변한다. 이 방식은 측정 오차의 편류(Drift)가 없어 오차가 누적되지 않는다는 장점이 있다. 다만 초기 위성을 발견하기까지의 시간이 오래 걸릴 수 있고, 위치 측정 주기가 수 Hz 수준으로 낮다는 단점이 있다. 또한 GNSS 기반 측위는 전파 수신이

어려운 실내 환경에선 사용할 수 없고, 위치 측정의 정밀도도 상대적으로 낮은 편이다.

나. 영상 정합 방법

GNSS 정보가 없거나 부정확할 때 사용 가능한 절대위치를 추정하는 방법으로 영상 정합 방법이 있다[2]. 영상 정합 방법은 2차원 또는 3차원 지도를 사전 정보로 가지고 있다. 무인 비행체에서 찍은 카메라 영상과 지도를 비교하여 현재의 위치와 고도를 파악한다. 이 방법은 카메라에서 획득한 영상을 지리 정보와 정확히 일치시킬 필요가 있는 군사 목적에 유용하다[3]. 영상 정합 방법은 다른 환경에서 찍힌 영상을 매칭하기 때문에 영상을 보정하는 전처리 과정과 두 영상을 매칭하는 영상 처리 과정이 필요하다. 전처리 과정에서는 다른 카메라로 찍힌 두 영상을 동일한 카메라 시점으로 옮기고 두 영상의 라인을 일대일로 일치시킨다. 이 과정을 1) 카메라 파라미터를 계산하는 캘리브레이션(Calibration)과 2) 영상 왜곡을 보정하도록 영상을 변환하는 렉티피케이션(Rectification)이라고 한다. 이후 영상 처리 과정에서는 1) 정합의 기준이 되는 키 프레임(Key frame)을 등록(Registration)하고, 2) 두 영상을 비교한다. 이때 영상 내 비교 대상을 무엇으로 하느냐에 따라 방법이 다양하다. 예를 들어, 전체 픽셀 기반으로 상호 정보(Mutual information)를 최적화할 수 있다. 또는 Kanade-Lucas-Tomasi 추적기(KLT feature tracker)로 영상의 특징점을 매칭할 수도 있다[3].

최근 딥러닝 기반의 영상 특징 정보를 추출하는 방식이 우수한 성능으로 관심을 받으며, 이를 활용한 영상 정합 연구도 늘고 있다[2,4]. 이 연구에서는 영상의 특징점 대신 의미 있는 정보를 딥러닝을 이용해 추출하였다. 엣지(Edge)와 세그멘테이션(Segmentation)의 외형 정보를 추출해 영상 정합 매

칭에 사용하였다.

2. 주변 환경 인지 기술

무인 비행체가 자율 비행을 하기 위해서는 이동 경로상의 물체를 탐지하고 임무 수행지 주변 환경을 인지해야 한다. 인지의 수준과 그 범위에 따라 주변 장애물 탐지 기술과 SLAM 기술로 분류할 수 있으며, 최근에는 딥러닝을 통한 환경 인지 기술이 소개되고 있다.

가. 주변 장애물 탐지 기술

무인 비행체는 비행 도중 주변 장애물을 탐지하여 회피할 수 있어야 한다. 이를 목적으로 상황에 따라 초음파 센서, LiDAR, 레이더, 카메라(단일 카메라, 스테레오 카메라, RGB-D 카메라)가 선택적으로 사용된다. 이 중 카메라는 주위 환경에 대한 풍부한 영상 정보를 제공하는 값싼 센서로 초소형 무인 항공기(MAV: Micro Aerial Vehicle)에서도 손쉽게 사용되고 있다. 하지만 카메라만으로 자율 비행을 하는 경우 성능의 한계가 있다.

장애물이나 위험을 탐지하는 데는 초음파 센서와 LiDAR와 같은 능동 센서가 효과적이다. 초음파 센서는 수 m 이내, LiDAR는 50~100m 이내, 레이더는 수백 m 이상 떨어진 물체를 감지할 수 있다. 이 중 LiDAR는 몇 년 전까지만 해도 크고 무거운 고가의 센서였으나 최근에는 소형화, 경량화된 3차원 LiDAR가 상용화되어 널리 보급되고 있다. 따라서 상황에 따라 가능한 센서를 활용하여 센서 정보를 퓨전하면 보다 나은 성능을 기대할 수 있다.

환경 인지 분야의 연구는 카메라 센서로부터 나오는 영상 기반 기술을 중심으로 필요에 따라 다른 센서 정보를 추가하는 방식으로 개발되고 있다. 카메라 센서를 이용한 장애물 탐지 방법으로는 스레

레오 카메라(Stereo camera)를 통한 장애물 탐지 기술, optical flow 기술, 그리고 특징점 추적 기술이 있다[5]. 이들은 SLAM 기술에 비해 연산량이 적기 때문에 온보드 컴퓨팅으로 실시간 처리가 필요한 경우에 적합하다. 복잡한 장애물이 없는 단순한 환경에서 사용하는 경우가 많으며, 외부 환경과 비행체 간의 상대적인 위치 파악이 가능하다.

스테레오 카메라 기술은 인간이 두 눈을 이용해 물체의 깊이 정보를 파악하듯 일정 거리가 떨어진 두 대의 카메라를 사용하여 주변 환경의 깊이 정보를 얻는다. 두 카메라에서 동시에 획득된 두 영상을 비교하면, 좌측 영상과 우측 영상에서 동일하게 나타나는 점들의 위치가 다르다. 이 위치 차를 시차(Disparity)라고 하며, 가까운 거리에 있는 물체는 시차가 크고, 먼 거리에 있는 물체는 시차가 작다. 이 시차를 계산하여 주변 환경의 깊이 정보를 얻고, 장애물의 근접 여부를 판단할 수 있다. 이 기술은 무인체의 움직임 없이도 정적인 환경을 파악할 수 있어 유용하다. 또한 별도의 센서 없이 두 대의 카메라만으로 높은 정밀도의 환경 맵(Map)을 구성할 수 있다. 하지만 스테레오 카메라로부터 깊이 정보를 계산하는 과정은 연산이 복잡하다. 따라서 무인 이동체의 자율 비행을 위한 스테레오 카메라 기술은 계산량을 줄여 결과를 실시간으로 얻는 방향으로 연구가 진행되고 있다.

Optical flow 기술은 연속하는 두 영상의 차이를 비교하여 자신의 움직임과 장면 변화를 감지하는 기술이다. 이때 같은 물체라면 영상의 밝기가 보존된다고 가정한다. 인접한 두 영상 간 눈에 띄는 밝기 변화가 발생한다면 물체가 움직였다는 사실을 알고, 그 움직임의 속도를 계산한다. Optical flow 기술로 장애물과의 거리를 파악하는 방식은 우리가 일상적으로 경험하는 사실에 근거한다. 우리는 멀리 있던 장애물의 크기가 점점 커진다면 장애물

과의 거리가 점점 가까워짐을 알 수 있다. 이러한 영상 정보를 이용해 optical flow 방법으로 물체의 깊이 정보를 계산하고, 이를 장애물 회피에 사용한다[6].

통상적으로 optical flow는 장애물을 회피할 정도로 충분히 조밀하지 못하다. 따라서 이 밀도를 높이는 연구가 개발되고 있다. Maier와 Humenberger는 모든 움직이는 물체의 움직임을 감지할 정도로 조밀한 optical flow 계산을 하였다[7].

특징점 추적 기술 역시 두 영상을 비교하여 자신과 장애물의 상대적인 위치를 파악한다. 영상 간 변화를 측정하기 위해 쉽게 변하지 않는 특징점을 영상에서 추출한다. 이후, 두 영상 간의 특징점을 매칭해 자신과 외부 물체의 상대적인 움직임을 계산한다. 이 특징점은 코너 점과 같이 눈에 잘 띄고 카메라나 물체의 움직임에도 쉽게 변하지 않는 특성을 가질수록 유리하다. 이러한 성질을 갖는 특징점을 추출하기 위해 사용하는 영상 특징 정보를 특징 기술자라고 한다. 이 중 Scale Invariant Feature Transform(SIFT), Features from Accelerated Segment Test(FAST)와 Speeded Up Robust Features(SURF)가 뛰어난 성능과 빠른 속도로 인해 널리 사용된다.

나. SLAM 기술

SLAM 기술은 비행체 스스로 센서 데이터로부터 정밀한 주변 환경 지도를 점진적으로 만든다. 이 정교한 알고리즘은 계산이 복잡하여 온보드 컴퓨터보다는 Ground Control Station(GCS)의 고사양 컴퓨터에서 계산하는 경우가 많다[8].

SLAM 기술은 자신의 위치를 계산하면서 동시에 정적인 외부 환경과 움직이는 물체를 파악한다. 이렇게 만들어진 환경 지도는 장애물 탐지 및 회피 시 외부 환경에 대한 풍부한 정보를 제공한다(그림

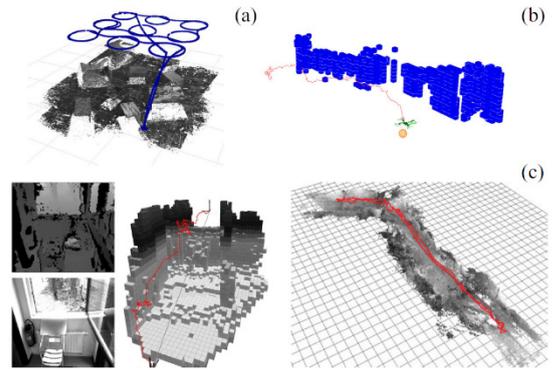


그림 2 SLAM 기반 환경 지도 형태[1]:
(a, c) 3차원 포인트 클라우드 맵, (b) occupancy grid 맵

출처 C. Kanellakis and G. Nikolakopoulos, "Survey on Computer Vision for UAVs: Current Developments and Trends," J. Intell. Robot. Syst., vol. 87, no.1, 2017, pp. 141-168. CC BY 4.0

2 참조). 따라서 특징점 추적 방법이나 optical flow 방법보다 더 복잡한 환경에 대응할 수 있다.

Parallel Tracking and Mapping(PTAM) 알고리즘 [9]은 대표적인 실시간 SLAM 알고리즘이다. 이 단일 카메라 기반 알고리즘은 추적과 매핑(Mapping)을 독립적인 일로 분리하고 각각을 병렬 스레드(Thread)로 처리한다. 추적 스레드는 카메라의 움직임을 추적하는 역할을 담당하고, 매핑 스레드는 연속한 두 영상 간 특징점의 3차원 맵을 생성하는 역할을 담당한다. 이때 배치 프로세스 최적화를 사용하여 수천 개의 특징점을 매핑한다.

Oriented Fast and Rotated Brief SLAM(ORB-SLAM) 알고리즘[8] 역시 최신 SLAM 기반 방법이다. 초음파 센서와 전방의 카메라를 사용해 실내 자율 비행을 하였다. 초음파 센서로는 맵의 스케일 값을 계산한다. 그리고 카메라 영상으로부터 비행체의 위치를 계산하고 포인트 클라우드(Point cloud) 형태의 환경에 대한 3차원 맵을 생성한다.

이 알고리즘 역시 실시간성을 위해 세 가지 스레

드를 병렬처리한다. 각 스레드는 추적, 매핑, 루프 클로징(Loop closing)을 담당한다. 특징점 추적에는 PTAM과 유사하게 FAST 코너 탐지를 사용한다. 비행체의 위치 예측을 더 정확하게 하기 위해 Kalman 필터를 사용해 Inertial Measurement Unit(IMU) 센서 데이터를 퓨전하였다. 즉, 예측 모델을 기반으로 IMU 관찰 값을 반영해 비행체의 위치를 보정한다. 최종적으로 루프 클로징은 실내 위치 계산 시 발생하는 편류 오류를 보정한다.

Extended Kalman Filter-SLAM(EKF-SLAM) 알고리즘은 영상 기반 SLAM과 레이저 기반 SLAM을 EKF 기반의 관성항법시스템(INS: Inertial Navigation System)으로 통합하였다[10]. 영상 기반 SLAM은 비행체의 상태와 특징점의 위치를 추정하는 데 extended Kalman 필터를 사용한다. Extended Kalman 필터는 IMU 관측값을 받아 위치 예측값을 업데이트한다. 레이저 기반 SLAM은 Monte-Carlo 자세 탐색(Pose search) 방법을 이용하여 레이저 스캔 포인트를 occupancy grid 맵에 매칭하고, Kalman 필터를 이용해 자세를 추정한다. 이때 영상 기반 SLAM의 특징점 매칭 결과를 스캔 매칭 비용 함수에 포함시킨다. 이 방법으로 영상 정보를 추가 활용하여 매칭의 불확실성을 줄였다. 특히 레이저 센서의 스캔 범위를 벗어나는 긴 통로와 같은 공간에서 이더럼 멀티 센서 정보를 퓨전하는 것이 보다 정확하게 자세를 추정하는 데 유용하였다[10].

다. 딥러닝 기반 환경 인지 기술

최근 무인 비행체에서도 보다 높은 수준의 환경 인지에 대한 관심이 늘어나고 있다. 특히, 딥러닝 기반 영상 인식 기술과 객체 추적 기술의 발달에 힘 입어 환경에 대한 이해의 폭이 더 넓어졌다. 그에 따라 다양한 자율 비행 시나리오가 가능해진다. 특정한 경로를 지정하지 않더라도 무인 비행체는

도로나 산속 길을 따라 비행할 수 있다[11,12]. 이때 무인 비행체는 카메라 영상에서 스스로 길을 검출하고 그 길을 따라간다. 관심 차량을 무인 비행체를 이용해 모니터링하며 자율 비행 기술로 따라갈 수도 있다[1]. 이 경우 차량을 검출하고 이를 추적하는 기술이 필요하다. 또한 장애물 탐지 시 물체에 대한 구분이 가능하므로 사람에게 위협이 가지 않도록 환경을 충분히 고려할 수 있다. 이는 보다 안전한 자율 비행을 하도록 도움을 준다. 이렇듯 영상 인식 기술과 접목할 때 정교한 시나리오 기반의 자율 비행이 가능해진다.

III. 경로 계획 기술

본 장에서는 무인 비행체의 자율 비행을 위한 경로 계획 기술에 대해 소개한다. 특히 본 장에서 다루는 경로 계획은 무인 비행체의 동특성을 고려하지 않고 기하학적으로 장애물과의 충돌이 없는 안전한 경로를 생성하는 path planning이 아닌, 무인 비행체의 동특성을 고려하여 무인 비행체가 추종하기에 적합한 경로를 생성하는 motion planning이다[13]. 만약 무인 비행체가 천천히 움직이고 구동기(actuator) 동역학과 같은 동적인 제한 조건을 무시할 수 있다면 path planning으로도 장애물과의 충돌이 없는 경로 계획이 가능하다. 하지만 무인 비행체가 무인 비행체의 특성에 맞게 기동하기 위해서는 동적인 제약조건이 더이상 무시될 수 없는 조건에서 경로 계획 문제를 고려해야 한다.

일반적으로 경로 계획 문제는 그림 1의 경로 계획 모듈에 나타난 바와 같이 광역적(Global) 경로 계획과 국소적(Local) 경로 계획으로 구성된 2단계 계층구조를 가진다. 상위 단계에 해당하는 광역적 경로 계획에서는 장애물과의 충돌이 발생하지 않기 위해 기하학적 조건만 만족하는 경로를

계획한다. 광역적 경로 계획의 결과물은 수 개의 경로점으로 이루어진 경로점들의 집합이며, 경로점을 생성하는 과정에서 무인 비행체의 동역학은 고려되지 않는다. 하위 단계에 해당하는 국소적 경로 계획에서는 상위 단계에서 결정된 임의의 인접한 두 경로점 사이에 동적으로 실현 가능한 부드러운 궤적(Trajectory)을 생성한다. 즉 상위 단계에서는 경로점을 추종할 수 있는 무인 비행체의 능력을 고려하지 않았지만, 하위 단계에서는 무인 비행체의 동특성을 반영하여 무인 비행체가 보다 쉽게 추종 가능한 궤적을 생성함으로써 주어진 궤적을 정확하게 추종하기 위한 제어 입력 생성을 용이하게 한다.

1. 광역적 경로 계획

본 절에서는 2단계 계층구조를 갖는 경로 계획 문제의 상위 단계에서 기하학적으로 장애물과 충돌하지 않는 경로점들의 집합을 생성하기 위해 제안된 방법들을 소개한다. 현재까지 제안된 방법들은 크게 탐색 기반(Search-based) 알고리즘과 샘플링 기반(Sampling-based) 알고리즘으로 구분할 수 있다.

가. 탐색 기반 알고리즘

일반적으로 탐색 기반 알고리즘은 지도상의 장애물 정보를 그래프(graph)로 변환하여 탐색에 사용한다. 그래프는 꼭짓점(vertex)과 꼭짓점과 꼭짓점 사이를 잇는 엣지로 구성된다. 탐색 기반 알고리즘 중 가장 보편적으로 사용되는 A* 알고리즘은 출발 꼭짓점에서 목표 꼭짓점까지 최적 경로를 탐색하는 그래프 탐색 알고리즘이다. 최적 경로를 결정하기 위해서는 각 꼭짓점에 대한 비용 함수를 정의해야 하는데, A* 알고리즘에서 사용되는 비용

함수는 출발 꼭짓점에서 현재 꼭짓점까지의 경로 가중치와 현재 꼭짓점에서 목표 꼭짓점까지의 추정 경로 가중치의 합으로 표현된다. 일반적으로 탐색 기반 알고리즘은 고차원 공간에서는 비효율적이라고 여겨지기도 하지만, 추정 경로 가중치를 설계하는 방식에 따라 탐색 속도를 가속화시킬 수 있고 해의 최적성을 향상시킬 수 있다는 점에서 A* 알고리즘을 변형한 여러 형태의 알고리즘이 제안되어왔다. 그중에서도 LPA* 알고리즘에서는 다음 경로점 후보들이 가질 수 있는 목표 꼭짓점까지의 경로 가중치를 모두 조사하여 그중 가장 최솟값을 추정 경로 가중치로 사용하였다[14]. 즉, 기존의 A* 알고리즘보다 한 단계 앞을 예측할 수 있는 추정 경로 가중치를 사용함으로써 알고리즘이 최적 해로 수렴하지 못하고 극소점(Local minima)에 빠질 가능성을 줄이고자 했다.

A* 알고리즘과 LPA* 알고리즘의 가장 큰 단점은 장애물에 대한 정확한 정보가 사전에 제공되어야 한다는 점이다. 만약 탐색 중간에 돌발 장애물이 나타날 경우에는 현재 위치를 새로운 시작점으로 간주하고 목적지까지 다시 경로를 계획해야 하며, 수정된 경로의 최적성은 보장되지 못한다. 탐색 기반 알고리즘 중 D* Lite 알고리즘은 예상치 못한 돌발 장애물이 나타났을 때 그래프화된 기존의 지도 정보를 업데이트하여 새로운 최적 경로를 효율적으로 재계산하기 위해 제안되었다[15]. D* Lite 알고리즘과 A* 알고리즘의 가장 큰 차이점은 계산 방향이 반대라는 점이다. A* 알고리즘은 시작점에서부터 목표점을 향해 점진적으로 비용 함수의 총합이 작은 방향으로 다음 경로점을 찾아가는데 비해 D* Lite 알고리즘은 목표점에서의 비용 함수가 0으로 정의되며 목표점으로부터 시작점을 향해 탐색을 수행하는 방식이다. 즉, D* Lite 알고리즘은 탐색 시작점이 현재 무인 비행체의 위치로

고정되지 않는다는 점에서 돌발 장애물이 나타날 경우 그래프화된 지도 정보를 업데이트하기가 용이하다.

나. 샘플링 기반 알고리즘

샘플링 기반 알고리즘은 복잡한 고차원 공간에서도 빠른 시간 안에 적절한 경로를 효율적으로 찾아내기 위해 제안된 방법이다. 여러 샘플링 기반 알고리즘 중 가장 보편적으로 사용되는 RRT 알고리즘은 경로 계획을 위해 시작점에서부터 목표점까지 점진적으로 트리(Tree)를 구축하는 방식을 사용한다[16]. 이를 위해 우선적으로 장애물과의 충돌이 없는 공간에서 무작위 샘플(Sample)을 추출하고, 샘플이 추출되면 그 샘플과 현재까지 구축된 트리에서 가장 가까운 노드 사이에 연결이 시도된다. 연결은 동역학적 구속 조건과 같은 제한조건을 만족하면서도 장애물과의 충돌이 없을 때만 가능하며, 연결이 가능한 경우에만 해당 샘플은 트리의 새 노드로 추가된다. 이러한 과정을 반복해 나가면서 시작점에서 목표점까지 하나의 연결된 궤적이 만들어질 때까지 트리를 아직 탐색되지 않는 공간으로 확장시키는 것이 RRT 알고리즘의 기본 개념이다. RRT 알고리즘의 장점과 단점을 간략하게 기술하면 다음과 같다. RRT 알고리즘은 상태 제약 조건이 있는 비선형 시스템에 대한 개루프(Open-loop) 궤도를 생성하는 기술로도 볼 수 있다. 즉, 임의의 입력이 일정 시간 동안 시스템에 주어졌을 때 다음 상태 변수를 적분해낸 것을 샘플로 간주하게 된다. 때문에 미분방정식으로 표현될 수 있는 모든 다양한 시스템에 RRT 알고리즘을 적용할 수 있다는 장점을 가진다. 하지만 RRT 알고리즘은 탐색된 경로의 최적성을 보장하지 못한다는 단점을 가진다. 이러한 단점을 보완하기 위해 RRT 알고리즘을 변형한 여러 형태의

알고리즘이 제안되어왔으며, 그중에서도 RRT* 알고리즘은 재연결(Rewiring) 연산을 추가 수행하여 현재까지 구축된 트리의 구조를 수정함으로써 점근적 최적성(Asymptotic optimality)을 보장하기 위해 제안된 방법이다[17]. 점근적 최적성이란 알고리즘의 반복 횟수가 무한대에 가까워짐에 따라 최적 해를 포함할 확률이 1에 접근하는 것을 말한다. 재연결 연산은 현재 샘플이 연결되어 있는 노드의 비용과 일정 반경 이내에 위치한 주변 노드와 재연결되었을 때의 비용을 비교하여, 후자의 비용이 더 낮을 경우 기존의 노드와의 연결을 끊고 새로운 노드와 연결하는 것이다. 이러한 수정을 반복해 나감에 따라 계획된 경로가 점진적으로 최적 해에 수렴하게 된다.

2. 국소적 경로 계획

본 절에서는 경로 계획 모듈의 하위 단계에서 임의의 두 경로점 사이에 동적으로 실현 가능한 부드러운 궤적을 생성하기 위해 제안된 방법들을 소개한다. 가장 간단하게는 경로점과 경로점을 직선으로 연결할 수 있다. 하지만 이러한 궤적을 정확하게 추종하기 위한 구동기 명령 생성이 어려울 수 있다. 예를 들어, 궤적에 급격한 기동을 요구하는 구간이 포함되어 있을 경우에는 이 궤적을 추종하기 위해 계산된 제어 명령이 실제 구동기가 작용할 수 있는 값의 한계보다 커서 구동기의 포화(Saturation)가 발생할 수 있다.

무인 비행체가 보다 쉽게 추종할 수 있도록 무인 비행체의 동특성을 반영한 궤적을 만들기 위해 다양한 방법들이 제안되어 왔다. 그중에서도 참고문헌 [18]에서 제안된 minimum snap trajectory는 동역학적으로 실현 가능한 곡선 궤적을 생성하는 대표적인 방법이다. Snap은 위치 벡터를 시간에 대해

4번 미분한 물리량에 해당한다. 이 연구에서는 무인 비행체의 동역학이 실제로는 복잡하지만 출력 변수를 3축의 위치와 요(yaw)각으로 정의하면 무인 비행체의 상태 변수와 제어 입력이 출력 변수 혹은 출력 변수를 여러 번 미분한 식으로 표현될 수 있음을 증명하였다. 즉, 3축의 위치와 요각으로 정의된 궤적과 제어 입력 간의 매핑 함수를 만들어냄으로써 주어진 궤적을 더 정확하게 추종할 수 있는 제어 입력을 생성할 수 있다.

Minimum snap trajectory를 결정하는 것은 결국 대표적인 최적화 문제에 해당하는 2점 경계치 문제(Two-point boundary value problem)의 해를 결정하는 것이다. 2점 경계치 문제의 성능 지수는 snap의 크기를 단위 시간 동안 적분한 것이며, 성능 지수를 최소화하는 동안 각 경로점에서 요구되는 경계치 조건과 경로점과 경로점 사이에서 요구되는 동역학과 관련된 구속 조건을 만족시켜야 한다. 특히, minimum snap trajectory는 성능 지수를 최소화함에 따라 모터에 요구되는 토크(Torque) 명령의 크기가 최소화된다는 장점을 가진다. 즉, snap을 최소화함에 따라 제어에 필요한 노력도 감소하는 효과를 얻을 수 있다. Minimum snap trajectory는 B-spline, 베지어 곡선(Bezier curve)과 같은 대표적인 기하학적 곡선뿐만 아니라 시간에 대한 n 차 다항식으로 모델링될 수 있다. 만약 경로점과 경로점을 연결하는 궤적을 시간에 대한 n 차 다항식으로 모델링한다면, 2점 경계치 문제의 결정 변수는 n 차 다항식의 계수이다. 더 발전된 연구에서는 정수 계획법(Integer programming)을 이용하여, 외부 환경에 존재하는 장애물 정보를 2점 경계치 문제의 구속 조건에 포함함으로써 장애물과의 충돌 회피가 가능한 궤적을 생성하는 연구가 진행되었다[19]. 또한 다수의 장애물 정보를 반영하더라도 빠른 시간 안

에 2점 경계치 문제를 풀 수 있도록 정수 계획법 대신에 2차 계획법(Quadratic programming)을 이용하는 방법도 제안되었다[20].

3. 최신 연구동향

본 절에서는 광역적 경로 계획과 국소적 경로 계획으로 구성된 2단계 계층구조를 기반으로 무인 비행체를 위한 경로 계획 문제를 다루는 최신 연구동향에 대해 기술한다. 먼저, 비교적 최근에 제안되었으나 그 영향력이 급격히 확산되고 있는 방법은 참고문헌 [21]에서 제안된 RRT* 알고리즘과 minimum snap trajectory를 접목한 것이다. 이 연구에서는 RRT* 알고리즘을 사용하여 기하학적으로 충돌 없는 경로를 얻은 후, minimum snap trajectory를 효율적으로 계산하기 위해 2차 계획법을 대체할 수 있는 해석적인 해를 찾는 방법을 제안하였다. 또한, 이 궤적이 돌발 장애물에 의해 충돌 가능성이 있을 경우에는 중간(Intermediate) 경로점을 추가하여 원래의 궤적을 보다 안전한 경로점에 더 가깝게 끌어 올리는 방법을 제안했다. 그러나 이 방법은 중간 경로점이 추가됨에 따라 계산 복잡도가 비례해서 증가하며, 중간 경로점이 최대 얼마나 필요한지를 사전에 판단할 수 없다는 단점을 가진다. 무인 비행체가 복잡한 고차원 공간에서 비행할 경우, 사전에 알려진 혹은 알려지지 않은 다수의 장애물 정보를 어떻게 모델링하여 경로 계획 문제에 포함할지에 대한 의견이 분분하다. 참고문헌 [22]에서는 정수 계획법을 이용하여 RRT* 알고리즘과 minimum snap trajectory를 접목한 연구를 다수의 장애물과 충돌 가능성이 있는 문제에 적용하기 위한 연구를 수행하였다. 장애물 정보를 최적화 문제에 포함하기 위해 장애물의 각 면(Face)에 수

직 방향으로 무인 비행체가 일정 거리 이상 항상 떨어져 있어야 한다는 조건을 혼합 정수 계획법의 구속조건으로 모델링하였다. 하지만 혼합 정수 계획법은 장애물의 개수가 증가함에 따라 계산 복잡도가 비례해서 증가한다는 단점을 가진다. 참고문헌 [23]에서는 직접적으로 장애물 정보를 모델링하지 않고 비행 가능한 영역을 식별하기 위해 motion planning을 수행하는 상위 단계에서 경로점 집합을 결정하는 대신 corridor를 생성하는 방법을 제안하였다. Corridor는 장애물 주변에 비행 가능한 공간을 나타내기 위한 것으로 3차원 공간상에서는 타원체나 각기둥과 같은 임의의 다면체 형태의 공간으로 모델링 된다. 공간상에서 여러 개의 corridor가 생성되면, 각 corridor마다 적절한 순서를 할당함으로써 무인 비행체가 현재 위치에서부터 목표점까지 이동 가능하게 한다. 또한, 무인 비행체가 corridor 내부에서만 비행할 것을 기하학적 구속조건으로 모델링하고 최대 속도와 가속도를 물리적 구속조건으로 모델링함으로써 최적화 문제를 통해 minimum snap trajectory를 생성한다. 그림 3에 경로점 기반 path planning 방식과 corridor 기반 path planning 방식의 차이점을 나타내었다.

사전에 모든 장애물 정보를 모델링하지 않고도 motion planning을 수행하기 위한 방법으로 머신러닝을 이용한 방법도 연구되고 있다. 이를 위해 참고문헌 [24]에서는 RRT* 알고리즘 대신 kino-FMT 알고리즘을 사용하였다. 이 알고리즘은 오프라인에서 장애물이 없는 공간을 가정하고 무인 비행체의 동특성을 고려한 경로점을 생성한다. 뿐만 아니라 경로점 간의 비용을 저장하는 탐색표(Look-up table)를 구축하고, 이 탐색표를 SVM 기법을 이용하여 학습시킨다. 그 후 온라인상에서 학습된 탐색표를 바탕으로 현재 경로점에서 가장 적

절한 다음 경로점을 판별한다. 이러한 방법은 장애물의 개수에 따라 알고리즘의 계산 복잡도가 증가하지 않는다는 장점을 가진다.

지금까지는 2단계 계층구조를 기반으로 무인 비행체를 위한 경로 계획 문제를 다루는 연구들을 소개하였지만, 최근에는 이 틀을 벗어난 연구도 수행되고 있다. 특히, 참고문헌 [25]에서는 센서로부터 제공된 포인트 클라우드 정보에서 바로 장애물과의 충돌 없이 안전하게 비행 가능한 궤적을 생성하는 방법이 제안되었다. 포인트 클라우드는 3차원 좌표를 가지고 있는 점들로 불규칙하게 구성된 자료를 말하며, 다양한 센서 유형에 적용할 수 있는 가장 낮은 수준의 거리 측정 방식이다. 이 연구에서는 3차원 LiDAR 센서와 IMU 센서로부터 포인트 클라우드 정보가 주어지면 온라인상에서 장애물 주변에 비행 가능한 corridor를 생성하고, 베지어 곡선을 사용하여 무인 비행체의 위치가 corridor 내부에 있도록 제한하면서 무인 비행체가 추종하기 용이한 부드러운 궤적을 생성하였다. 이 방법은 계산 복잡도가 큰 격자 지도를 구축하거나 장애물과의 충돌 여부를 검사하는 작업을 필요로 하지 않기 때문에 장애물 정보가 사전에 정확하게 알려지지 않은 복잡한 환경에서도 실시간으로 경로 계획을 구현할 수 있다는 장점을 가진다.

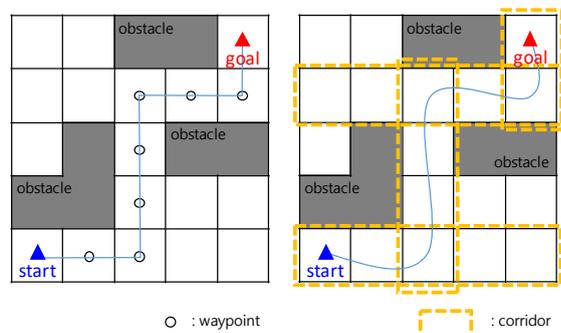


그림 3 경로점 기반 path planning(왼쪽)과 corridor 기반 path planning(오른쪽)

IV. 결론

최근에는 무인 비행체가 다양한 영역으로 그 이용이 확산됨에 따라 자율 비행 기술에 대한 수요가 증가하고 있다. 이에 따라 자율 비행의 수준을 높이고 그 안전을 확보할 필요가 있다. 본 고에서는 무인 비행체의 자율 비행을 위한 환경 인지 및 경로 계획 분야의 연구동향을 분석하였다. 환경 인지 분야에서는 카메라 센서 정보를 기반으로 목적에 따라 다양한 추가 센서 정보를 이용하는 방향으로 연구개발이 진행되고 있다. 또한 경로 계획 분야에서는 기하학적으로 장애물과의 충돌이 없는 안전한 경로를 탐색하기 위한 광역적 경로 계획 연구와 기체의 동역학적 움직임을 고려한 국소적 경로 계획 연구가 진행되고 있다. 이러한 연구를 기반으로 향후에는 무인 비행체가 오퍼레이터의 개입 없이 안전한 자율 비행을 제공할 수 있을 것으로 전망한다.

용어해설

UAV 조종사가 비행체에 직접 탑승하지 않고 지상에서 원격조종, 사전 프로그램된 경로에 따라 자동 또는 반자동 방식으로 자율 비행하거나 인공지능을 탑재하여 자체 환경 판단에 따라 임무를 수행하는 비행체를 통칭함.

약어 정리

EKF	Extended Kalman Filter
FAST	Features from Accelerated Segment Test
FMT	Fast Marching Trees
GCS	Ground Control Station
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System

KLT	Kanade–Lucas–Tomasi
LiDAR	Light Detection and Ranging
LPA	Lifelong Planning A*
MAV	Micro Aerial Vehicle
ORB-SLAM	Oriented Fast and Rotated Brief SLAM
PTAM	Parallel Tracking and Mapping
RRT	Randomly–exploring Random Tree
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded Up Robust Features
SVM	Support Vector Machine

참고문헌

- [1] C. Kanellakis, G. Nikolakopoulos, “Survey on Computer Vision for UAVs: Current Developments and Trends,” *J. Intell. Robot. Syst.*, vol. 87, no. 1, 2017, pp. 141-168.
- [2] J. R. Braga et al., “An Image Matching System for Autonomous UAV Navigation based on Neural Network,” in *Proc. Int. Conf. Contr., Automation, Robot. Vis. (ICARCV)*, Phuket, Thailand, Nov. 2016, pp. 1-6.
- [3] M. Sanfourche et al., “Perception for UAV: Vision-based Navigation and Environment Modeling,” *AerospaceLab*, vol. 4, 2012, pp. 1-19.
- [4] A. Nassar et al., “A Deep CNN-Based Framework For Enhanced Aerial Imagery Registration with Applications to UAV Geolocalization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. Workshops*, Salt Lake City, UT, USA, June 2018.
- [5] Y. Lu et al., “A Survey on Vision-based UAV Navigation,” *Geo Spat. Inf. Sci.*, vol. 21, no. 1, 2018, pp. 21-32.
- [6] A. Al-Kaff et al., “Monocular Vision-based Obstacle Detection/Avoidance for Unmanned Aerial Vehicles,” in *Proc. IEEE Intell. Veh. Symp. (IV)*, Gothenburg, Sweden, June 2016, pp. 92-97.
- [7] J. Maier, M. Humenberger, “Movement Detection based on Dense Optical Flow for Unmanned Aerial Vehicles,” *Int. J. Adv. Robot. Syst.*, vol. 1, no. 2, 2013, pp. 1-11.
- [8] O. Esrafilian, H. D. Taghirad, “Autonomous Flight and Obstacle Avoidance of a Quadrotor by Monocular SLAM,” in *Proc. Int. Conf. Robot. Mechatron. (ICROM)*, Tehran, Iran, Oct. 26-28. 2016, pp. 240-245.

- [9] G. Klein, D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Proc. IEEE ACM Int. Symp. Mix. Augment. Real.*, Nara, Japan, Nov. 2007, pp. 225-234.
- [10] D. Magree, E. N. Johnson, "Combined Laser and Vision-aided Inertial Navigation for an Indoor Unmanned Aerial Vehicle," in *Proc. Am. Contr. Conf.*, Portland, OR, USA, June 2014, pp. 1900-1905.
- [11] N. Smolyanskiy et al., "Toward Low-flying Autonomous MAV Trail Navigation Using Deep Neural Networks for Environmental Awareness," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Vancouver, Canada, Sept. 2017, pp. 4241-4247.
- [12] T. Ayoul, et al., "UAV Navigation above Roads Using Convolutional Neural Networks," Technical Report, Stanford University, 2017.
- [13] S. Liu et al., "Search-Based Motion Planning for Quadrotors Using Linear Quadratic Minimum Time Control," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, Vancouver, Canada, Sept. 2017, pp. 2872-2879.
- [14] S. Koenig, L. Maxim, "Fast Replanning for Navigation in Unknown Terrain," *IEEE Trans. Robot.*, vol. 21, no. 3, 2005, pp. 354-363.
- [15] S. Grzonka et al., "A Fully Autonomous Indoor Quadrotor," *IEEE Trans. Robot.*, vol. 28, no. 1, 2012, pp. 90-100.
- [16] S. M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," 1998.
- [17] S. Karaman, F. Emilio, "Sampling-Based Algorithms for Optimal Motion Planning," *Int. J. Robot. Res.*, vol. 30, no. 7, 2011, pp. 846-894.
- [18] D. Mellinger, K. Vijay, "Minimum Snap Trajectory Generation and Control for Quadrotors," in *IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 2520-2525.
- [19] S. Tang, K. Vijay, "Mixed Integer Quadratic Program Trajectory Generation for a Quadrotor with a Cable-Suspended Payload," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 2216-2222.
- [20] R. Deits, T. Russ, "Efficient Mixed-Integer Planning for UAVs in Cluttered Environments," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, Seattle, WA, USA, May 2015, pp. 42-49.
- [21] C. Richter et al., "Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments," in *Robotics Research*, Springer, 2016, pp. 649-666.
- [22] D. Mellinger et al., "Mixed-Integer Quadratic Program Trajectory Generation for Heterogeneous Quadrotor teams," in *IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 477-483.
- [23] J. Chen et al., "Online Generation of Collision-Free Trajectories for Quadrotor Flight in Unknown Cluttered Environments," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 1476-1483.
- [24] R. Allen, P. Marco, "A Real-Time Framework for Kinodynamic Planning with Application to Quadrotor Obstacle Avoidance," in *AIAA Guid. Navig. Control Conf.*, San Diego, CA, USA, Jan. 2016, pp. 1-18.
- [25] F. Gao et al., "Flying on Point Clouds: Online Trajectory Generation and Autonomous Navigation for Quadrotors in Cluttered Environments," *J. Field Robot.*, vol. 30, no. 7, 2011, pp. 846-894.