

# 지역별 중요도를 고려한 드론 최적 경로 감시 시스템

이유경 · 강문정 · 김나영 · 이충목<sup>†</sup>

한국의국어대학교 산업경영공학과

## Drone Surveillance System Considering Dynamic POIs

Yukyung Lee · Munjeong Kang · Nayoung Kim · Chungmok Lee

Department of Industrial and Management Engineering, Hankuk University of Foreign Studies

This study proposes a new drone surveillance system by considering the relative importance of spatial area under the surveillance. Traditionally, the drone surveillance systems employed a simple routing rule often realized as zig-zag movements without alternation of altitude. Recognizing that the altitude of drones can be manipulated to take more detailed pictures of the POIs (Points of Interest), we represent a new mathematical formulation and an efficient heuristic algorithm that considers not only the locations but the altitude of drones. We also incorporate a machine learning algorithm to identify POIs inside the given area, which enables us to implement a fully automated drone surveillance system. The proposed system is superior to the previous ones because the consideration of the relative importance of POIs makes the drone cover the non-POI areas much more efficiently by using higher altitude and taking fewer photos. We implement a proof-of-the-concept of the proposed algorithm, and report results obtained from real-world test cases.

**Keywords:** Drone Surveillance, Location And Routing Problem, Object Detection

### 1. 서론

최근 드론(drone)에 대한 연구와 이용이 활발히 이루어지고 있다. 드론은 무인항공기(Unmanned Aerial Vehicle, UAV)의 일종으로서 1918년 미국에서 군사 목적으로 처음 개발되었다(Freeman and Freeland, 2015). 최근 GPS(Global Positioning System)의 보급과 인공지능(Artificial Intelligence, AI)의 발달과 함께 아마존, 구글과 같은 글로벌 기업들은 드론을 기존의 산업에 활용할 수 있는 방법을 제공하였고 이에 따라 드론 시장의 크기는 나날이 성장하고 있다. 현재 민간용으로도 사용되는 드론은 감시, 촬영, 레저, 농업, 수송 등 운용 범위가 확대되고 있으며 미국 방위산업 컨설팅 업체인 Teal Group은 드론 시장이 2014년 64억 달러에서 2023년 114억 달러 규모로 성장할 것으로 전망하고 있다(Zaloga *et al.*, 2014). 최근 드론은 무선 전파를 이용해 원격으로 사람이 조정하는 것에서 벗어나

인공지능 및 사물 인터넷 등 정보통신 기술과 융합해 새로운 사업 시장을 주도할 신기술로 주목받고 있다.

과거 군사적 감시 목적으로 개발된 드론의 가장 큰 장점은 사람이 직접 가지 않아도 빠르게 주변을 살펴볼 수 있다는 것이다. 이는 최근의 드론을 이용한 상용화된 감시시스템의 등장을 가져왔다. 일반적인 드론 감시시스템은 감시대상이 되는 지역을 지정하고 드론의 비행 고도를 지정하면 지정된 고도를 유지하면서 감시 영역을 지그재그(zig-zag) 형태로 왕복하는 방식으로 동작한다. 드론은 지정된 고도와 경로에서 전체 감시 영역을 포함할 때까지 지속적으로 사진이나 동영상을 촬영하며, 촬영된 사진과 영상은 서버로 전송된다. 이와 같이 비교적 간단한 방식으로 작동하는 드론 감시시스템은 다음과 같은 문제를 가진다. 첫째, 드론의 비행 고도(flight altitude)를 특정 고도로 고정시켜야한다. 드론에 설치된 카메라는 해상도가 정해져 있기 때문에 더 정밀한 사진을 얻기 위해서는 드론의 비행

이 연구는 2018학년도 한국의국어대학교 교내학술연구비의 지원에 의하여 이루어진 것임.

<sup>†</sup> 연락저자 : 이충목 교수, 경기도 용인시 처인구 모현읍 외대로 81 공대 530호, Tel : 031-330-4378, Fax : 031-330-4120,

E-mail : chungmok@hufs.ac.kr

2018년 9월 5일 접수; 2019년 3월 26일 수정본 접수; 2019년 7월 5일 게재 확정.

고도를 낮추어 감시 대상과의 거리를 줄여야 한다. 그러나 너무 낮은 비행 고도는 한 번에 촬영하는 영역의 크기를 줄이게 된다. 이는 결국 전체 비행시간의 증가와 전체 영역을 감시하기 위해 필요한 사진의 수를 크게 증가 시킨다. 고정된 고도를 사용하면 사용자는 전체 감시 영역을 고려하여 적절한 고도를 설정해야 하는데 이는 일반적으로 쉽지 않다. 둘째, 기존의 드론 감시 시스템은 지정된 감시 영역 중에 중요한 지역(Point of Interest, POI)을 고려하지 못한다. 지정된 넓은 영역을 단순히 순회하여 감시하기 때문에 중요한 지역을 구분할 수 없고 중요한 지역과 중요하지 않은 지역에 대해 모두 같은 해상도(resolution)의 사진을 얻는다. 셋째, 감시 대상의 실시간 변동에 대처하기 힘들다. 기존의 드론 감시 시스템은 감시 영역과 고도 등을 모두 사전에 지정한 값을 사용하며 드론 감시 시스템은 단순한 지정된 작업을 반복 수행하기 때문에 실시간으로 변하는 감시 대상에 대해 고려를 할 수 없어 감시의 효율성을 저하시킨다.

기존 드론 감시시스템의 문제점은 드론의 감시 고도와 중요 지역을 지능적으로 판단함으로써 해결할 수 있다. 본 연구에서는 지정된 영역을 모두 감시하기 위해 고도의 변화까지 고려하는 새로운 드론 위치 경로 최적화 문제(Drone Location & Routing with Altitude, 이하 DLRwA)를 제시하고 이를 풀기 위한 수리 모형과 발견적 기법(heuristic method)을 제시한다. 이때 드론의 고도 변화는 감시 지역의 중요도에 따라 결정된다. 중요도가 높은 지역은 드론의 고도를 낮추어서 높은 해상도의 감시 사진을 얻어야 하고, 중요하지 않다고 판단되는 지역은 필요하지 않은 경우 높은 고도에서 사진을 촬영하는 것이 더 넓은 지역을 하나의 사진에 포함시킬 수 있어 효율적이다. 이때 유의해야 할 점은 지역의 중요도가 드론의 촬영 고도를 하나의 값으로 확정하는 것은 아니라는 것이다. 드론은 일반적으로 수평 이동보다 수직 이동에 더 많은 시간과 배터리를 소모한다고 알려져 있다(Tseng *et al.*, 2017). 따라서 불필요한 상하 기동을 억제하는 것이 유리하고 지역의 중요도는 드론 촬영의 최대 필요 고도(maximum required altitude)를 지정하는 역할을 한다. 즉, 지역의 중요도에 따라 해당 지역의 사진 촬영 고도의 상한(upper bound)이 주어지며 드론은 이 고도제약을 만족하는 한 전체 비행시간과 배터리의 소모를 고려하여 지능적으로 최적의 촬영 고도를 판단하여야 한다. 드론의 최적 촬영 위치(고도까지 포함한)와 경로는 촬영할 지역의 중요도에 따라 결정되어야 한다. 본 연구는 중요한 지역이 실시간으로 변동될 수 있다고 가정하고 이를 고려한 드론 감시 시스템을 제안한다. 본 논문에서는 드론의 감시 대상이 사람(human)이라고 가정하였다. 이 가정에 따르면 중요 지역은 상대적으로 사람이 많이 몰려 있는 지역으로 정의된다. 드론이 찍은 사진을 바탕으로 사람이 많이 몰려 있는 지역을 판단하기 위해서 본 연구에서는 기계학습(machine learning)을 이용한 방법을 제안한다. 최근 딥러닝(deep learning)으로 대표되는 기계 학습은 이미지, 음성인식 등의 분야에서 매우 큰 관심을 받고 있다

(LeCun *et al.*, 2015). 본 연구에서 제안하는 드론 감시 시스템은 사진에서 사람을 추출하고 추출된 사람들의 위치와 빈도를 바탕으로 감시영역 안에서의 상대적인 중요도를 판단한다. 이러한 과정을 거쳐 판단된 중요 지역은 드론의 촬영 위치와 경로를 결정하는 최적화 문제의 입력으로 주어지게 되고 이를 바탕으로 새로운 드론 경로가 수립된다. 이러한 과정을 반복함으로써 실시간으로 변화하는 감시 대상의 위치를 고려한 드론 감시시스템을 구현할 수 있다.

본 논문은 다음과 같이 구성된다. 제 2장은 드론 경로 최적화에 관련된 기존연구들을 설명한다. 제 3장은 본 연구가 대상으로 하는 응용 예제를 구체적인 예를 들어 설명한다. 제 4장은 드론 감시 시스템의 전체적인 구성을 설명하고 드론의 최적 경로를 수립하기 위한 DLRwA 문제에 대한 새로운 수리 모형과 발견적 기법(heuristic method)을 제시한다. 제 5장은 제안된 수리 모형과 발견적 기법에 대한 계산 실험 결과를 제시하고 제 6장은 드론 감시 시스템의 구현에 대한 구체적인 내용을 다루며 마지막으로 제 7장에 결론을 제시한다.

## 2. 관련 연구

드론을 이용한 기존 연구는 크게 두 가지 방향으로 구분할 수 있다. 첫 번째 연구 분야는 드론의 비행, 조정, 그리고 착륙 등에 대한 연구이다. 전통적으로 이러한 분야는 항공 또는 기계 조정(control) 연구 분야와 관련 있고 이미 매우 많은 연구가 이루어져 있다(Nonami *et al.*, 2010; Sebbane, 2015). 두 번째 분야는 드론의 운영 효율성에 대한 연구로 본 연구는 이 분야 연구의 연장선에 있다. Tseng(2017)은 효율적인 배터리의 사용을 위한 드론 경로 생성에 대한 연구를 수행하였다. 이 논문에서 저자들은 드론이 운동 방향에 따라 배터리의 소모량이 다른 점을 실험을 통해 확인하고 배터리의 재충전과 소모량을 줄이기 위한 드론의 경로 수립 문제를 제시하였다. 하지만 이 연구에서도 드론이 방문해야 하는 위치는 정해진 입력값으로 주어지기 때문에 이 연구는 잘 알려진 전기자동차 차량 경로 문제(Erdoğan and Miller-Hooks, 2012)의 연장선에 있다고 할 수 있다. 최근 주목받는 드론의 활용법중에 한꺼번에 많은 수의 드론(aerial swarms)을 동시에 사용하여 공연을 하거나 정보를 전달하는 것이 있다. Bartashevich *et al.*(2018)은 많은 수의 드론을 동시에 비행할 때 배터리의 소모를 고려한 드론 조정 알고리즘을 제시하였다. 드론의 제한된 배터리를 무선으로 충전하는 방법은 Khonji *et al.*(2017)에 의해 제시되었다. 태양열 발전기에 연결된 로봇팔(robot arm)을 이용해 드론의 위치를 감지하고 드론이 위치한 방향으로 무선 충전을 시도한다. 이를 통해 원격으로 드론의 제한된 배터리를 효율적으로 충전할 수 있는 가능성을 제시하였다. Ji *et al.*(2015)는 바람의 영향을 고려하여 드론의 비행경로를 결정하는 알고리즘을 제시하였다. 지상의 관측소에서 측정된 바람의 방향을 바탕으로 드론이 위치한

지역의 바람의 방향과 속도를 추정하여 이를 이용하여 드론의 경로를 수정하는 방법이 사용되었고, 이를 이용하면 기존보다 배터리 소모면에서 더 효율적인 경로를 수립할 수 있음을 보였다. 드론의 배터리 소모는 드론의 속도, 고도, 기온과 같은 운영 환경에 따라 크게 달라진다. 특히 드론에 사용되는 배터리는 근본적으로 전기자동차 등에서 사용되는 배터리와 크게 다르지 않기 때문에 전기 배터리의 사용 특성에 대한 연구는 주로 전기 자동차를 대상으로 이루어져 왔다. Kim *et al.* (2013)은 전기 자동차를 대상으로 실시간 배터리 상태 예측 방법을 제시하였다. 배터리가 완전 방전되면 배터리 자체에 손상이 갈 수 있다. 또한 전기 자동차 또는 드론의 운행이 갑자기 정지할 때 발생하는 위험에 대비하여야 한다. 따라서 배터리의 상태를 정확하게 예측하는 시스템과 알고리즘은 중요하다. 본 연구에서는 전기 자동차 운전자의 운전 습관을 고려하여 배터리의 상태를 실시간으로 예측하는 방법이 제시되었다.

최근 드론은 취미용으로 많이 사용되고 있다. 하지만 불특정 다수에 의해 사용되는 드론은 보안, 사생활 침해 등의 문제를 일으킬 수 있다. Ding *et al.* (2018)은 개인들이 사용하는 드론을 감시하는 시스템을 제시하였다. 본 시스템은 허가받지 않은 드론을 자동으로 인식하는 기능을 바탕으로 드론 감시 시스템을 제시한다. 따라서 본 연구의 드론은 감시의 대상이며 감시의 도구가 아니다. 드론을 사용함으로써 발생하는 여러 가지 법적, 도덕적 문제들은 Finn and Wright (2012)에서 다루어졌다. 특히, 드론을 이용한 감시 시스템이 유색인종이나 가난한 사람들과 같이 편견을 가진 대상을 주로 목표로 한다는 점에 문제를 제기하였다.

드론을 특수한 상황에 사용하는 연구들도 이루어지고 있다. Flammini *et al.* (2016)은 드론을 이용하여 철도 시설을 점검하는 방안을 제시하였다. 철도 네트워크는 주기적인 점검이 필수적이지만 크기가 방대하기 때문에 기존의 점검방법은 시간과 비용이 많이 소요된다. 따라서 드론이 빠르고 효율적으로 선로의 상태를 점검하거나 선로 주변의 환경이 안전한지 확인하는데 사용된다. Bürkle and Essendorfer (2010)은 해양 감시 문제를 제시하였다. 비행기, 배, 레이다 등의 여러 가지 감시 수단은 감시 범위와 특징이 모두 다르다. 이 연구에서는 서로 다른 특징을 가지는 감시 수단을 통합 관리하는 시스템을 제시하였다. 특히, 이 연구에서는 감시 수단의 고도에 따른 차이를 고려하였는데 이 점은 본 논문에서 다루는 드론 고도 설정 문제와 연관성이 있다. Otto *et al.* (2018)은 최근까지의 드론 운영 최적화 관련 연구를 모은 survey 논문을 출판하였다. Otto *et al.* (2018)의 논문은 드론의 활용 분야를 크게 여섯 가지로 구분하였으며 그 중에 지역을 감시하거나 정해진 지역을 방문하기 위해 경로를 수립하는 문제들을 가장 대표적인 드론 관련 최적화 문제로 제시하였다. 특히 드론의 경로 수립과 관련하여 대부분의 기존 연구가 드론의 부족한 비행시간(배터리 제약)을 고려하였으나 드론의 고도에 따른 지상 감시 효율의 차이에 대한 연구는 수행되지 않았음을 확인할 수 있다.

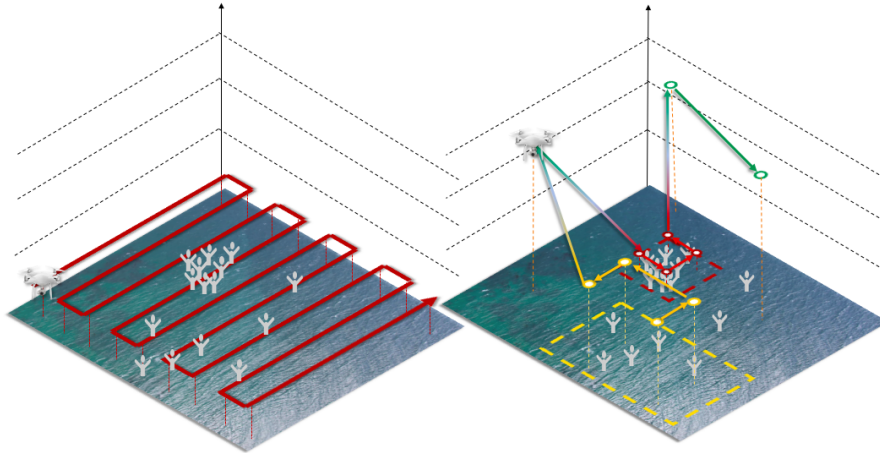
드론을 전통적인 물류에 적용한 연구들도 최근 많이 수행되었다. 특히 드론을 차량과 동시에 운영하는 방법이 관심을 받고 있다 (Popper 2015, DHL Press Release 2014). Wang *et al.* (2016)은 드론과 차량을 동시에 운영하는 VRPD (Vehicle Routing Problem with Drones)에서 드론을 이용함으로써 얻을 수 있는 최대 이익에 대한 이론적인 연구를 수행하였다. 이 연구에서 드론은 차량에 실려 운반되며 가벼운 소포를 인근 지점에 배달하는 용도로 사용된다. Dorling *et al.* (2016)은 배터리와 화물의 무게를 고려한 드론 배달 경로 문제를 제시하였다. 이 논문에서 저자들은 배터리의 소모를 고려한 수리 모형을 제시하고 이를 효율적으로 풀기 위한 시뮬레이티드-어닐링 (Simulated Annealing, SA) 메타-휴리스틱을 이용한 알고리즘을 제시하였다.

드론-차량 혼합 경로 문제는 더 일반적인 관점에서 보면 잘 알려진 외판원 문제 (Travelling Salesman Problem, TSP, Kim *et al.* 2017)와 깊은 관련이 있다. TSP의 변형 문제들 중에 보조원을 가진 외판원 문제 (TSP with Sidekick)가 존재한다. Murray and Chu (2015)는 비행 보조원 외판원 문제 (Flying Sidekick TSP)를 제시하였으며 이 문제는 드론-차량 혼합 경로 문제의 특수한 경우에 해당한다. 이 논문에서 저자들은 이 문제에 대한 혼합 정수 계획법 수리 모형을 제시하였으며 간단한 발전적 기법 또한 제시하였다. 계산 실험 결과는 드론을 이용한 배달이 효율적일 수 있음을 보여준다.

본 연구는 드론의 경로뿐만 아니라 드론이 감시 영역을 모두 커버하는 사진을 찍도록 드론의 위치도 결정해야 한다. 특정 영역의 모든 지역에 영향력을 발휘하거나 서비스를 제공하도록 위치를 결정하는 문제들을 일반적으로 덮개 문제 (covering problems)라고 한다. 가장 잘 알려진 덮개 문제는 설비위치문제 (Facility Location Problem, FLP)이다. FLP는 대상 지역에 서비스를 제공하기 위한 설비를 설치할 위치를 정하는 문제로 오랫동안 많은 연구가 이루어졌다 (Farahani, 2012). FLP와 관련된 대부분의 연구는 설비가 영향을 미치는 거리 (covering distance)가 고정되어 있다고 가정한다. 이러한 고정 거리 가정은 기존 드론 감시 문제에서 고정된 고도를 가정하는 것과 비슷한 점이 있다. 최근 Berman *et al.* (2009)은 고정되지 않은 영향 범위 문제 (variable radius covering problem)를 제안하였다. 더 넓은 영향 범위를 위해서는 더 많은 자원을 투입해야 하기 때문에 이 문제는 설비의 위치와 영향 범위를 동시에 결정하는 문제이다. 드론의 경우 높은 고도는 더 넓은 사진 범위를 제공하기 때문에 드론의 고도를 조정하는 것은 설비의 영향 범위를 확대하는 것과 비슷한 효과를 가진다고 볼 수 있다.

### 3. 응용 예제

본 장에서는 본 연구에 제시하는 새로운 드론 감시 시스템의 실제 적용 예를 제시한다. 최근 드론 감시 시스템이 가장 많이



**Figure 1.** (Left) ZIG-ZAG Route Used in the Existing Drone Surveillance System, (Right) A New Drone Surveillance System Can Identify where the Survivors are and Use the Information to Build More Efficient Routes

사용되는 분야에는 재난 감시가 있다. 재난 발생 시에는 재난 지역에서 생존자들을 빠르게 찾아내는 것이 매우 중요하다. 예를 들어 해상에서 조난이 발생한 상황을 가정해보자. 해상 재난은 매우 넓은 지역을 빠른 시간 내에 감시하여 정확한 조난 지점을 파악하여야 한다. 이때 드론은 넓은 지역을 쉽고 빠르게 감시할 수 있는 효과적인 수단이 될 수 있다.

기존의 드론 감시 시스템은 고정된 고도를 이용하여 넓은 지역을 왕복 순회한다(<Figure 1>의 왼쪽 그림). 이런 방식의 드론 경로는 드론의 고도가 너무 낮으면 전체 지역을 모두 감시하는데 많은 시간이 소요될 수 있다. 반대로 드론의 고도를 너무 높이면 드론이 찍은 사진이 너무 넓은 지역을 포함하기 때문에 실제 생존자들의 정확한 위치와 상태를 파악하는데 어려울 수 있다. 이는 결과적으로 모든 생존자들의 위치를 빠르게 찾아내는 것을 어렵게 하여 생존 확률을 낮출 수 있다. 반면에 본 연구가 제시하는 새로운 드론 감시 시스템은 정해진 경로가 아닌 생존자들의 위치에 따라 동적으로 최적 경로를 수립한다(<Figure 1>의 오른쪽 그림). 이를 위해 본 연구가 제안하는 드론 감시 시스템은 단계별 감시를 수행한다. 첫 번째 감시 경로는 가능한 가장 높은 고도에서 전체 지역을 모두 포함하도록 한다. 첫 번째 감시 비행으로 얻은 사진을 분석하여 생존자가 있을 확률이 높은 지역을 파악하여 생존자가 존재할 확률의 순서대로 지역별로 상대적 중요도를 부여한다. 상대적 중요도를 바탕으로 중요도가 높은 지역은 드론의 비행 고도를 낮추고 상대적으로 중요하지 않은 지역은 전체 비행 시간을 최소화하는 비행 고도를 선택한다.

본 연구가 제안하는 드론 감시 시스템은 드론의 경로 생성에 지역별 중요도를 고려하여 전체 지역을 모두 감시하는 시간을 줄이는 것을 목표로 한다. 전체 지역을 감시하는데 걸리는 시간을 줄이면 드론은 감시 비행 사이의 시간을 줄일 수 있고 이는 결국 생존자들의 위치가 달라지는 경우에도 빠르게 생존자들의 위치를 추적할 수 있음을 의미한다. 특히, 해상 조난 상황의 경우 조류로 인해 생존자들의 위치가 변할 수 있기

때문에 전체 지역을 감시하는 시간을 줄이는 것은 매우 중요하다.

또 다른 적용 예로 대규모 목장의 소나 말 등을 감시하는 것을 생각할 수 있다. 넓은 지역에 동물을 풀어 놓을 경우 동물의 개체 수를 지속적으로 확인할 필요가 있다. 기존의 감시 시스템은 전체 지역을 단순 왕복하고 동물의 개체 수 확인은 별도로 이루어져야 했다. 반면에 본 연구가 제시하는 드론 감시 시스템은 동물들의 위치를 파악하여 동물들이 밀집된 지역은 더 자세한 사진을 얻음으로써 개체 수 확인을 더욱 수월하게 만들 수 있다.

#### 4. 새로운 드론 감시 시스템

본 장에서는 새로운 드론 감시 시스템의 각 단계를 자세히 설명한다. <Figure 2>는 드론 감시 시스템의 대략적인 작업 흐름을 나타내는 그림이다. 그림에서 보듯이 작업 흐름은 크게 1차 비행과 2, ..., N차 비행으로 나눌 수 있다. 먼저, 1차 비행은 중요 지역을 찾기 위해 가장 높은 고도로 빠르게 비행한다. 1차 비행에서는 어느 지역이 중요 지역인지 모르기 때문에 가장 높은 고도에서 비행하여 넓은 지역을 포함하는 사진을 찍고, 찍은 사진을 이용하여 중요 지역을 판단한다. 중요 지역의 위치들이 정해지면 고도를 조절하며 그다음 비행을 진행할 수 있다. 1차 비행 이후에는 사용자가 원하는 만큼 N차 비행이 가능하며 비행 종료를 원할 경우 비행을 종료할 수 있다.

새로운 드론 감시 시스템은 크게 두 가지 구성요소로 이루어져 있다. 1) 중요 지역(POIs)을 검출해내는 부분, 2) DLRwA를 풀어 드론의 경로를 생성해 내는 부분이다. 이후의 제 4.1절에서는 중요 지역 설정 알고리즘을 설명하고, 제 4.2절에서는 DLRwA 수리 모형과 휴리스틱 알고리즘을 제시한다. <Figure 2>에는 각 단계가 설명된 절이 표시되어 있다.

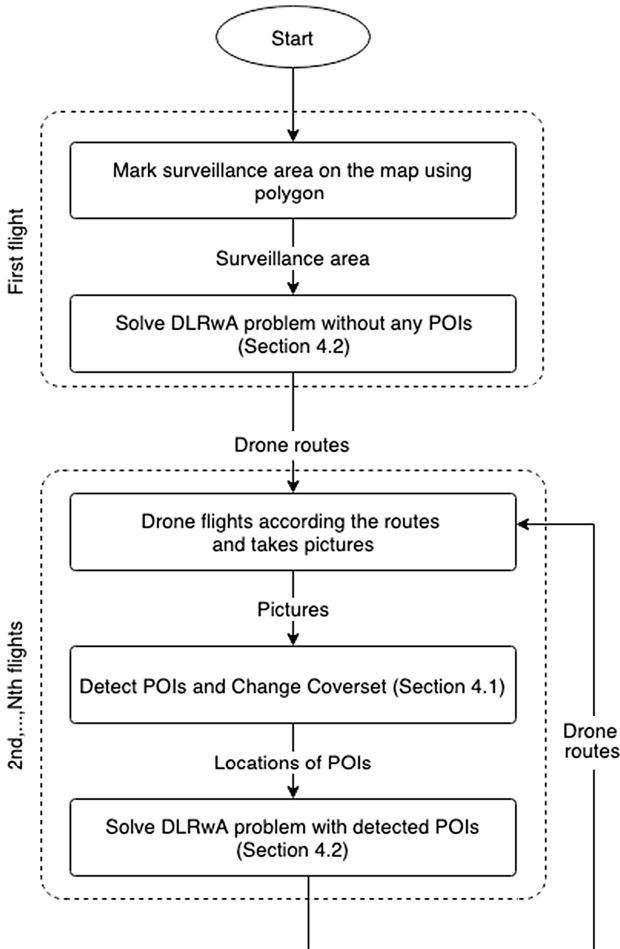


Figure 2. Flowchart of the Proposed System

4.1 중요 영역 설정

본 장은 감시할 지역의 중요도를 지능적으로 판단하기 위한 방법을 설명한다. 전술하였듯이 본 논문에서는 재난 시 조난자들의 구조를 가정하여 사진 속의 사람들을 찾아내 검출된 사람의 밀도를 기준으로 중요 지점을 설정한다. 사진에서 사람의 존재를 판단하기 위해 본 논문은 잘 알려진 특징 추출 방법인 FAST(Features from Accelerated Segment Test) 알고리즘과 딥러닝(Deep Learning)을 사용하였다. <Figure 3>은 본 연구

에서 중요지점을 판단하는 과정을 나타낸다. 각 단계에 대한 설명이 이후에 이어진다.

(1) FAST 알고리즘을 이용한 특징 추출

FAST 알고리즘이란 객체 인식을 위한 특징점 추출 알고리즘이다(Rosten *et al.*, 2010). 중심점 주위로 형성된 원 위의 픽셀들이 주변 픽셀 대비 뚜렷한 차이가 있으면 중심점에 특징점이 있다고 판단한다. 기존의 다양한 객체 인식 알고리즘 중에 FAST 알고리즘을 선택한 이유는 다음과 같다. 본 연구에서 특징점 추출에 사용된 사진은 상공에서 촬영된 사진으로, 사람 인식에 자주 사용되는 Pedestrian Detection을 적용한 결과 사람을 인식하지 못했다. 따라서 사람으로 인식될 가능성이 높은 특징점을 찾아낼 수 있는 다양한 알고리즘을 실험해보았고, 그 중에서 빠른 시간 내에 corner point를 찾아내는 FAST 알고리즘을 선택하게 되었다. <Figure 4>에서는 FAST 알고리즘의 예시를 보여준다. 직접 드론으로 촬영한 사진을 이용하였으며, 이 사진을 이용하여 FAST 알고리즘을 통한 특징점 추출을 수행하였다. <Figure 4>의 오른쪽 사진이 객체를 인식한 후 사진이며 빨간색 동그라미로 표시된 부분이 특징적인 객체로 인식된 부분이다.

그러나 FAST 알고리즘을 이용한 단순한 특징점 추출은 문제점이 있다. <Figure 4>의 오른쪽 그림에서 보이듯이 FAST 알고리즘은 사람뿐만 아니라 다른 특징점들도 많이 추출하기 때문에 추출된 특징점 중 사람만을 따로 구분할 필요가 있다. 이를 위해 각 특징점을 중심으로 사진을 작게 잘라(cropping) 작은 사진들을 만든다. 이렇게 작게 잘라낸 사진들은 이후에 설명하는 딥러닝 신경망을 통해 사람임을 구별한다.



Figure 4. Extracted Features by FAST Algorithm

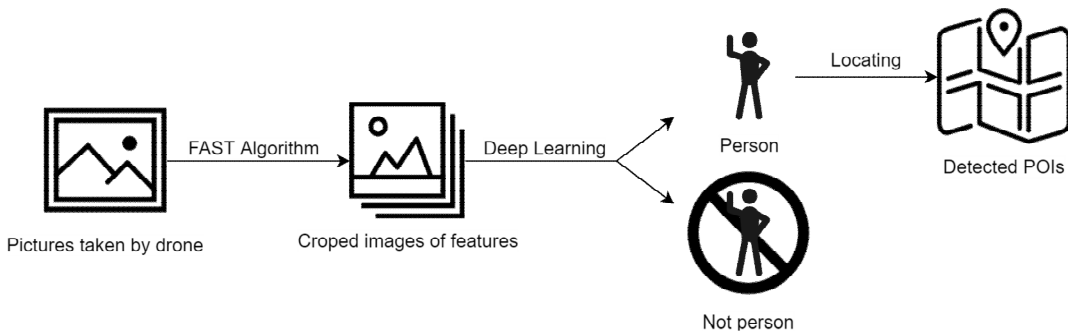


Figure 3. Schematic Diagram of Object Detection



(2) 딥러닝 신경망을 이용한 사람검출

딥러닝은 신경망의 일종으로 비교적 많은 숨겨진 단계(hidden layers)를 가지는 것이 특징이다. 최근 딥러닝 신경망은 이미지 분류에서 매우 큰 성과를 보이고 있는데(LeCun *et al.*, 2015), 이러한 딥러닝 신경망 성공의 가장 큰 원인은 CNN(Convolutional Neural Network, Krizhevsky *et al.*, 2012)의 도입이다. 본 연구에서는 CNN구조를 사용한 딥러닝 신경망을 이용해서 전 단계에서 만들어진 특징점 사진들(cropped images) 중에 실제 사람을 구분할 수 있도록 학습시켰다. 본 연구에서 사용한 CNN의 구조는 일반적인 구조를 따르며(2D convolution(kernel 3\*3) → Max Pool(2\*2) → linear(in = 4050, out = 64) → linear(in = 64, out = 2)), 신경망 학습 알고리즘은 Adam optimizer를 사용하였다.

딥러닝 신경망을 학습시키기 위한 학습데이터는 실제로 드론을 이용해서 확보한 저자들의 학교 캠퍼스 주변 사진들을 이용하였다. 드론을 지정된 경로로 여러 번 날짜를 바뀌가며 비행시킨 후 일정 간격으로 사진을 찍어 FAST 알고리즘의 input을 얻었다. 그 후 FAST 알고리즘을 이용하여 촬영된 사진으로부터 특징점들을 추출하고, 특징점을 중심으로 작게 잘라낸 이미지(cropped images)들을 얻었다. 이 이미지들은 드론을 통해 촬영한 원본 이미지를 잘라낸 것으로 두 이미지 사이에는 사이즈 차이가 존재한다. 본 연구에서 사용한 작은 이미지들은 모두 10,350개로 낮은 고도로 비행한 경우에 추출한 특징 점과 높은 고도로 비행한 경우에 추출한 특징점 이미지를 적절하게 섞어 학습하였다. 이를 이용해서 학습시킨 딥러닝 신경망의 판단 정확도(testing accuracy)는 89%였으며, 판단 정확도는 전체 이미지 개수 중 사람인 사진을 사람으로 판단한 개수의 비율을 의미한다. 즉, 판단 정확도는 고도의 차이에 따라 발생하는 특징점 개수에 상관없이 사람으로 판단되는 특징점을 얼마나 잘 판단했는지에 대한 지표이다. 여러 번의 검증을 통해 사진을 찍는 고도의 변화가 판단 정확도에 큰 영향을 미치지 않는 것을 확인하였다. 또한, 실험을 진행하면서 학습 데이터의 숫자가 증가할수록 판단 정확도가 높아

지는 것을 확인 할 수 있었다. 더 많은 학습 데이터를 확보하는 것은 일반적으로 그다지 어렵지 않은 일이고, 드론의 감시 대상이 다르면 다른 학습 데이터를 확보해야 할 필요가 있다.

따라서 본 논문은 가장 기본적인 딥러닝 신경망의 구조를 제시하고 특징점 사진(cropped images)으로 학습할 수 있음을 확인하는 것으로 충분하다고 판단하였다.

하나의 사진에는 여러 사람이 검출될 수 있다. 본 연구에서 개발한 드론 감시 시스템에서는 하나의 사진에 포함된 특징점 중에 사람으로 판별된 객체의 개수가 전체 개수의 5% 이상이면 그 사진의 중심을 중요지점으로 판단하였다. <Figure 5>는 작은 이미지들 각각에 대해 사람인 여부를 판단하는 과정을 보여준다. 그림에서 붉은색으로 표시된 특징점들은 사람으로 판단된 것들이다.

4.2 드론 경로 최적화 알고리즘(DLRwA)

본 장에서는 드론의 감시 경로를 최적화하기 위한 알고리즘을 설명한다. 본 장에서 풀어야 할 문제를 고도를 고려한 드론 위치 경로 최적화 문제(DLRwA problem)라고 이름 붙였으며 드론의 위치, 고도, 경로를 모두 결정해야 한다.

(1) 문제 정의와 수리 모형

드론 감시 상황을 모형화하기 위해 3차원 공간을 일정한 넓이의 격자로 나누어 격자의 교차점을 노드로 설정하였다. 그 후 감시영역으로 지정된 영역의 근접 교차점을 수리 모형에 사용할 노드(node)라 정의했다.

<Figure 6>은 드론 감시 영역의 모형화 단계를 나타낸다. 사용자가 감시하고자 하는 영역을 하얀색 폴리곤이라 가정했을 때 일정한 넓이의 격자가 생성되며 그중 폴리곤과 인접한 교차점은 노드를 나타낸다. <Figure 6>의 오른쪽의 노란 노드들은 감시 대상이 되는 노드들이다.



Figure 5. Deep Neural Network Classifies Persons from Other Objects

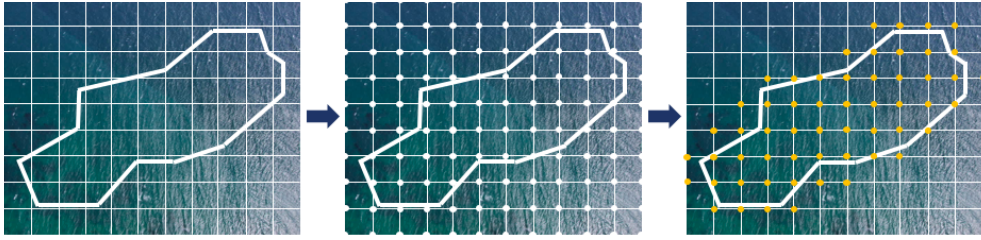


Figure 6. Discretizing of 2D Space with Grid

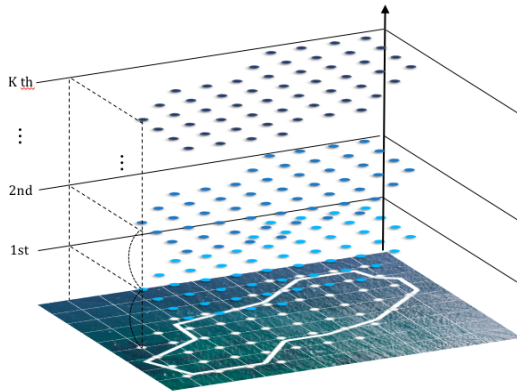


Figure 7. Discretizing of 3D Space with Grid

드론 감시는 3차원 공간에서 이루어지므로 노드 또한 3차원 공간에서 생성되어야 한다. <Figure 7>은 고도에 따라 3차원 공간에서 노드가 생성됨을 보여준다. 고도는 K단계까지 임의로 설정 가능하며 고도에 따라 감시되는 지상 영역의 크기가 달라진다. 즉, 1단계부터 K단계까지의 고도에서 사진을 찍었을 때 사진에 포함(감시)될 지면상의 노드가 달라진다.

<Figure 8>은 고도에 따른 감시 영역(covering area)의 차이를 나타낸다. <Figure 8>에서 볼 수 있듯이 고도가 낮을 때는 사진의 감시 영역이 줄어들지만 높은 해상도의 사진을 얻을 수 있으며, 고도가 높을 때는 사진의 감시 영역이 늘어나지만 해상도가 떨어짐을 알 수 있다.

지면상의 노드를 바닥 노드(surface nodes)라 정의했을 때, 특정 바닥노드를 촬영할 수 있는 3차원상의 노드들의 부분집합을 그 바닥노드의 커버셋(coverset)이라고 정의하자. 예를 들어, <Figure 9>와 같이 드론 감시에 사용되는 고도가 2단계(낮은 고도, 높은 고도) 있으며 두 개의 바닥노드 1과 2를 감시하는 상황을 생각해 보자. 바닥노드 1은 중요노드로써 낮은 고도에서의 고해

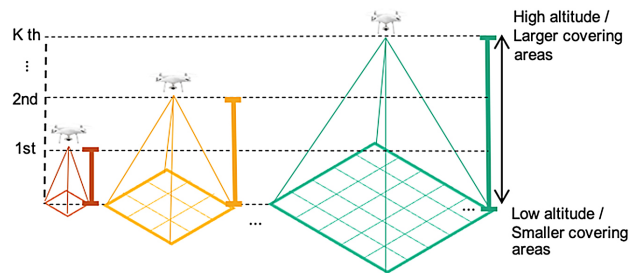


Figure 8. Higher Altitude Yields Larger Covering Areas

상도 사진이 필요하고 바닥노드 2는 중요하지 않은 노드므로 높은 해상도의 사진 또는 낮은 해상도의 사진 아무거나 이 바닥노드를 포함하기만 하면 충분하다. 중요한 바닥노드 1을 촬영하기 위해서는 낮은 고도에서 촬영해야 하며 이 때 가능한 촬영 노드는 1L~9L이다. 즉, 바닥노드 1의 커버셋은  $C_1 = \{1L, 2L, 3L, 4L, 5L, 6L, 7L, 8L, 9L\}$ 로 정의된다. 반면에 중요하지 않은 바닥노드 2의 커버셋은  $C_2 = \{10L, 11L, 12L, \dots, 18L, 1H, 2H, 3H, \dots, 23H, 24H, 25H\}$ 로 정의된다. 이 때 중요하지 않은 바닥노드의 커버셋은 낮은 고도/높은 고도의 촬영 가능 노드를 다 포함하기 때문에 드론은 낮은 고도 또는 높은 고도에서 해당 커버셋에 속하는 아무 노드에서나 촬영하기만 하면 바닥노드 2를 감시할 수 있다. 따라서 만약에 드론이 굳이 고도를 바꾸지 않는 것이 더 효율적이라면 중요하지 않은 바닥노드 2를 낮은 고도(10L~18L)에서 촬영하는 것도 가능하다. 반면에 높은 고도를 사용하면 중요하지 않은 바닥노드 여러 개를 동시에 하나의 사진으로 촬영할 수 있기 때문에 드론의 경로를 효율적으로 수립할 수 있다. 제 4.1절에서 설명한 방법으로 바닥노드들 중에 중요지점(POI)이 판별되면 모든 바닥노드들의 커버셋을 쉽게 만들 수 있다. 따라서 커버셋은 어떤 위치에서 더 고해상도의 사진을 찍어야 하는지를 제약해주는 역할을 한다.

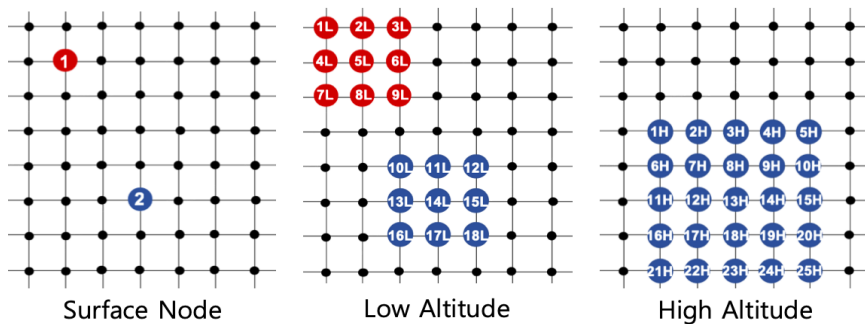


Figure 9. Definition of Coversets

이제 드론의 고도에 따라 감시 영역의 크기의 차이, 드론의 비행시간, 배터리 소모와 같은 자원(resource) 제약을 고려하여 드론의 촬영 지점(Location)과 이동 경로(Routing) 두 가지 해를 모두 도출하는 수리 모형을 새롭게 정의한다. 중요한 제약으로는 도출된 경로에 따른 드론의 배터리 소모량은 전체 드론 배터리 양을 초과할 수 없는 것과 드론의 비행 고도는 영역별 중요도를 고려해야 하는 것이 있다. 수리 모형에 사용될 매개변수와 결정 변수는 다음과 같다.

• 매개변수(Parameter)

- $N$  : 사진을 찍을 수 있는 3차원 상의 모든 노드 집합.  
 $D$  :  $\{d_1, d_2, \dots, d_m\}$ . 출발 노드  $d_1$ , 도착 노드  $d_m$ , 나머지  $d_2 \dots d_{m-1}$ 들은 배터리의 재충전을 고려하기 위한 가상 디포(depot) 노드들  
 $T_j$  :  $j$ 노드에서 사진을 찍을 때 걸리는 시간 ( $\forall j \in N$ )  
 $T_{ij}$  :  $i$ 노드에서  $j$ 노드로 이동할 때 걸리는 시간 ( $\forall i, j \in N \cup D$ )  
 $R$  : 드론을 재충전하는데 걸리는 시간. 잔여 배터리 양에 관계없이 일정함(예: 배터리를 완전히 충전된 다른 배터리로 교환한다고 가정)  
 $L$  : 바닥노드의 집합  
 $C_l \subset N$  :  $l \in L$  바닥노드의 중요도를 고려하여 만들어진 바닥노드  $l$ 을 촬영할 수 있는 노드들의 집합, 즉,  $l$ 의 커버셋  
 $M$  : Big-number  
 $B$  : 드론의 총 배터리양  
 $F_{ij}$  :  $i$ 노드에서  $j$ 노드로 이동할 때 소모되는 배터리 양, 사진 촬영에 배터리가 소모되지 않는다고 가정함.

• 결정 변수(Decision Variable)

- $x_{ij}$  :  $i$ 노드에서  $j$ 노드로 이동하면 1의 값을, 그렇지 않으면 0의 값을 갖는 이진 변수  
 $y_j$  :  $j$ 노드에서 사진을 찍으면 1의 값을, 그렇지 않으면 0의 값을 갖는 이진 변수  
 $u_i$  :  $i$ 번째 노드를 방문하는 순위를 나타내는 변수  
 $b_i$  :  $i$ 노드에서 남아있는 배터리 양

DLRwA는 잘 알려진 차량경로문제(VRP)와 비슷하지만 드론이 방문할 노드(사진을 찍을 위치)가 미리 결정되지 않고 다른 결정변수( $y_j$ )에 의해 정해진다는 차이가 있다. 또한 드론의 최대 비행거리는 배터리에 의존하며 드론의 재충전을 고려하여야 한다. 드론의 재충전을 고려하기 위해 여러 개의 가상의 디포노드들을 생성하여 드론의 전체 경로가 서브 투어(subtour)를 가지지 않도록 할 필요가 있다. 충전을 위한 가상의 디포노드들을 생성하는 방법은 전기자동차 차량경로문제에 대한 연구(Erdogan and Miller-Hooks, 2012)에서 소개된 방법이다.

DLRwA 문제의 수리 모형을 다음과 같이 제시한다.

$$(P) \min \sum_{j \in N} T_j y_j + \sum_{i, j \in N \cup D} T_{ij} x_{ij} + R \sum_{k \in D \setminus \{d_1, d_m\}} \sum_{j \in N} x_{kj} \quad (1)$$

$$\text{s.t. } x_{ij} = 0, \forall i, j \in N \text{ and } i = j \quad (2)$$

$$\sum_{j \in C_l} y_j \geq 1, \forall l \in L \quad (3)$$

$$\sum_{i \in N \cup D \setminus \{d_m\}} x_{ij} = y_j, \forall j \in N \cup D \setminus \{d_m\} \quad (4)$$

$$\sum_{i \in N \cup D \setminus \{d_1\}} x_{ji} = y_j, \forall j \in N \cup D \setminus \{d_1\} \quad (5)$$

$$\sum_{j \in N} x_{d_1 j} = 1 \quad (6)$$

$$\sum_{j \in N} x_{j d_1} = 0 \quad (7)$$

$$\sum_{j \in N \cup d_i} x_{jk} = \sum_{j \in N \cup d_i} x_{kj}, \forall k \in D \setminus \{d_1, d_m\} \quad (8)$$

$$\sum_{j \in N \cup D \setminus \{d_m\}} x_{d_m j} = 0 \quad (9)$$

$$\sum_{j \in N} x_{j d_m} = 1 \quad (10)$$

$$x_{ij} + u_i \leq u_j + M(1 - x_{ij}), \forall i, j \in N \cup D \setminus \{d_1\} \quad (11)$$

$$b_i - F_{ij} x_{ij} \geq b_j - M(1 - x_{ij}), \forall i \in N, \forall j \in N \cup D \quad (12)$$

$$B - F_{ij} x_{ij} \geq b_j - M(1 - x_{ij}), \forall i \in D, \forall j \in N \cup D \quad (13)$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in N \cup D \quad (14)$$

$$y_j \in \{0, 1\}, \forall j \in N \quad (15)$$

$$u_i \geq 0, \forall i \in N \cup D \quad (16)$$

$$b_i \geq 0, \forall i \in N \cup D \quad (17)$$

목적함수 (1)은 드론이 사진을 촬영하는 시간, 비행하며 이동하는 시간, 배터리 충전 시간의 총합을 최소화한다. 수리 모형의 제약식 (2)는 출발과 도착이 같은 노드로는 이동하지 않는다는 것을 뜻한다(no self-loop). 제약식 (3)은 바닥 노드  $L$ 에 속한 모든 노드들이 노드의 중요도에 따라 결정된 촬영가능 위치(커버셋) 중에 하나에서 촬영이 되도록 한다. 제약식 (4)는  $j$ 노드에서 사진을 찍을 때만, 다른 노드  $i$ 노드에서  $j$ 노드로 이동하는 아크는 단 하나가 선택 되도록 한다. 제약식 (5)는  $j$ 노드에서 사진을 찍을 경우,  $j$ 노드에서 다른 노드  $i$ 노드로 이동하는 아크는 단 하나가 선택 되도록 한다. 제약식 (6)과 (7)은 출발 노드에서 드론이 출발하고 다시 출발 노드로 돌아오지 않게 한다. 제약식 (8)은 드론이 충전을 위해 가상 디포 노드에 도착했다면 다시 떠나야 함을 나타낸다. 제약식 (9)와 (10)은 드론이 도착 노드에 당도하면 더 이상 그 노드를 떠나지 않도록 한다. 제약식 (11)은 드론의 경로가 서브 투어(subtour)가 되지 않도록 해주는 소위 MTZ 제약식이다(Miller et al., 1960). 제약식 (12)은 각 노드에서 남아있는 배터리 양을 계산해주는 식이다. 제약식 (13)은 드론이 가상의 충전 디포를 방문하면



배터리를 최대치로 바꿔준다. 제약식 (14)부터 (17)은 각 결정 변수의 조건을 나타낸다. 수리 모형 (P)는 혼합정수계획법(Mixed Integer Programming, MIP) 문제로써 MIP solver인 Cplex 등으로 최적해를 얻을 수 있다. 또한 DLRwA의 촬영 가능 고도를 하나로 제안하고 커버셋을 하나의 노드로 제안하면 잘 알려진 차량경로문제(VRP)와 동일함을 쉽게 보일 수 있다. 따라서 DLRwA는 NP-hard임을 쉽게 보일 수 있고(Garey and Johnson, 1979) 문제의 크기가 커질수록 최적해를 얻는 시간이 빠르게 증가한다.

## (2) DLRwA를 위한 휴리스틱 알고리즘

앞 절에서 제시한 수리 모형은 최적해를 얻을 수 있지만 문제의 크기가 커짐에 따라 계산 시간이 급격하게 증가하기 때문에 드론 감시 시스템에 적용하기는 적절하지 않다. 본 절에서는 좋은 해를 빠른 시간 안에 얻을 수 있는 발견적 기법(heuristic method)을 제안한다. 본 발견적 기법은 수리 모형 (P)가 두 개의 서로 상이한 문제의 결합이라는 점에서 착안하였다. 수리 모형 (P)는 드론이 사진을 찍을 위치를 정하는 문제(제약식 (3)과 (15))와 정해진 위치를 방문하는 경로를 정하는 문제(제약식 (3)과 (15)를 제외한 나머지 모든 제약식. 단,  $y_l$ 는 고정됨)로 구성되어 있다. 각각의 문제를 앞으로는 (LP)(Location Problem, 위치문제)와 (RP)(Routing Problem, 경로문제)라 정의하자. 수리 모형 (P)는 (LP)와 (RP)가 제약식 (4)와 (5)를 통해 연결된 형태임을 쉽게 알 수 있다. 본 연구가 제안하는 발견적 기법의 아이디어는 (LP)와 (RP)를 분리하여 접근(decomposition approach)하는 것이다.

먼저 위치문제 (LP)를 다음과 같이 정의한다.

$$(LP) \quad \min \sum_{j \in N} \left(1 + r_j + \frac{2dist_j}{dist_{\max}}\right) y_j \quad (18)$$

$$\text{s.t.} \quad \sum_{j \in C_l} y_j \geq 1, \quad \forall l \in L \quad (19)$$

이때,  $dist_j$ 와  $dist_{\max}$ 는 디포와 노드  $j$ 와의 거리와 노드 중 가장 먼 거리(i.e.,  $dist_{\max} = \max_{j \in N} \{dist_j\}$ )를 나타내고,  $r_j$ 는 Uniform 분포  $U(-0.2, 0.2)$ 를 따르는 임의숫자(random number)이다. 본 연구에서는 특정한 한 지점을 디포로 설정하였으며 따라서  $dist_j$ 와  $dist_{\max}$ 는 정해진 디포의 위치와 노드의 위치로 각각 구할 수 있다. (LP)는 바닥노드를 모두 커버하는 사진을 찍을 위치를 정하는 문제이며 목적함수는 되도록 사진을 찍을 노드가 디포에 가깝게 위치하도록 유도한다. 목적함수에 임의숫자  $r_j$ 와  $2dist_j/dist_{\max}$  값을 추가함으로써 더 좋은 위치해가 제공된다는 것을 여러번의 실험을 통해 알게 되었다. 특히 임의숫자  $r_j$ 의 경우 -0.5~0.5사이의 값을 여러 가지 해본 결과 Uniform 분포  $U(-0.2, 0.2)$ 일 때 적당한 범위의 좋은 위치해를 제공하였기 때문에 선택하였다. 마찬가지로  $dist_j/dist_{\max}$ 의 계수값도 2로 설정했을 때 비교적 더 좋은 위치해가 제공되었기 때문에 선택하였다. 임의숫자  $r_j$ 는 이후에 설명할 2단계 발견적

기법에 사용된다. (LP)의 목적함수가 (P)의 목적함수와 다른 이유는 (LP)는 (RP)를 풀기위해 좋은 위치해를 제공해야 하기 때문이다. (LP)의 위치해를 이용하여 (RP)로 드론의 경로를 구하면 위치해와 경로해를 가지고 원래의 목적함수 (1)의 값을 쉽게 계산할 수 있다.

그리고 경로문제 (RP)는 주어진 위치문제 (LP)의 최적해  $y$ 를 모두 방문하는 최적의 경로를 찾는다. 본 연구에서는 드론의 한정된 배터리 용량을 고려하기 위해서 ESPPRC(Elementary Shortest Path Problem with Resource Constraints, Feillet *et al.*, 2004)을 사용하여 (RP)를 풀었다. ESPPRC는 자원 제약을 고려한 레이블링(Labeling) 알고리즘으로 차량경로 문제들의 열 생성(column generation) 알고리즘의 부분제(subproblem)로 많이 사용된다(Lee *et al.*, 2012). <Figure 10>은 ESPPRC 알고리즘의 수도코드(pseudo code)를 보여준다. ESPPRC는 두 개의 중요한 실행단계(procedure)가 있는데 Extend()와 EFF()이다. Extend()는 현재 노드의 레이블들을 인접한 다른 노드들로 확장하며 EFF()는 두 개의 레이블의 집합을 가지고 다른 레이블에 지배(dominated)되지 않은 레이블들만을 반환한다. ESPPRC 알고리즘에 대한 자세한 설명은 Feillet *et al.*(2004)을 참조한다. ESPPRC 알고리즘은 다음과 같은 매개변수들로 정의된다.

- $A_i$  : 노드  $y_i$ 에 대한 레이블들의 목록
- $Succ(y_j)$  : 노드  $y_i$ 의 선행노드(successors)들의 집합
- $E$  : 확장 되어질 노드들의 목록
- $L_{ij}$  : 노드  $y_i$ 에서  $y_j$ 노드로 확장된 레이블들의 집합
- $Y$  : 사진을 찍어야 하는 노드들의 집합
- $s$  : 맨 처음 시작을 나타내는 값(0)
- $T_{ij}$  : 노드  $y_i$ 에서 노드  $y_j$ 로 이동할 때 걸리는 시간
- $B_{ij}$  : 노드  $y_i$ 에서 노드  $y_j$ 로 이동할 때 소모되는 배터리 양

ESPPRC 알고리즘의 성능은 각 노드의 레이블의 개수가 증가할수록 급격하게 나빠진다. 특히, EFF() 실행단계는 두 개의 레이블 집합을 서로 비교해야하기 때문에 알고리즘 실행시간의 대부분을 소모하며 레이블의 개수가 많아질수록 수행시간이 오래 걸린다. 본 연구에서는 ESPPRC 알고리즘을 빠르게 풀기 위해 각 노드에서 Extend()에서 반환되는 레이블의 개수를 가장 좋은 15개로 제한하였다. 즉, 각 노드는 가장 좋은 15개의 레이블만 주변의 노드들로 확장시킨다. 이와 같이 확장되는 레이블의 개수를 제한하면 ESPPRC 알고리즘의 최종 해는 최적해를 보장하지 못하지만 좋은 해를 매우 빠른 시간 안에 찾을 수 있다.

(LP)와 (RP)는 각각 수리 모형과 ESPPRC 알고리즘으로 풀 수 있으나 위치와 경로를 분리하여 고려하기 때문에 단순히 각각의 문제를 한 번씩 푸는 것만으로는 좋은 해를 기대하기 어렵다. 즉, 위치를 결정할 때는 최종 경로의 품질을 알 수 없기 때문에 최대한 좋은 위치 해(good location solutions)를 많이 찾아내 각각의 경로해(routing solutions)를 비교하여 가장 좋은 해를 선택할 필요가 있다. 따라서 본 연구에서는 다음과 같은 2단계(2-Phase) 발견적 기법을 제시한다.

**Algorithm 1** Labeling algorithm for solving ESPPRC

```

procedure ESPPRC(s)
   $\Lambda_s \leftarrow \{(0, 0, \dots)\}$ 
  for all  $y_i \in Y \setminus \{s\}$  do
     $\Lambda_i \leftarrow \emptyset$ 
   $E = \{s\}$ 
  repeat
    Choose  $y_i \in E$ 
    for all  $y_j \in Succ(y_i)$  do
       $L_{ij} \leftarrow \emptyset$ 
      for all  $\lambda_i = (T_{ij}, B_{ij}, Y_i^0, Y_i^1, \dots, Y_i^{n-1}, Y_i^n) \in \Lambda_i$  do
        if  $Y_i^j = 0$  then
           $L_{ij} \leftarrow L_{ij} \cup \text{Extend}(\lambda_i, y_j)$ 
       $\Lambda_j \leftarrow \text{EFF}(L_{ij} \cup \Lambda_j)$ 
      if  $\Lambda_j$  has changed then
         $E \leftarrow E \cup \{y_j\}$ 
     $E \leftarrow E \setminus \{y_i\}$ 
  until  $E = \emptyset$ 
  
```

**Figure 10.** Pseudocode of ESPPRC for(RP)

DLRwA를 위한 2단계 발견적 기법(2-Phase heuristic algorithm)

**Phase 1.** 랜덤 탐색(Randomized search)

- Step 1.**  $r_j \in U(-0.2, 0.2), \forall j \in N$
- Step 2.** (LP)를 Step 1의  $r_j$ 를 이용하여 푼다.
- Step 3.** Step 2의 해를 집합  $W$ 에 저장한다.
- Step 4.** 집합  $W$ 의 크기가 5보다 작거나 Step 1의 반복 횟수가 50보다 작으면 Step 1부터 반복한다.
- Step 5.** 집합  $W$ 의 각각의 위치해에 대하여 (RP)를 풀어 드론의 경로를 구해 가장 좋은해를 선택한다.

**Phase 2.** 주변해 탐색(Local neighborhood search)

- Step 1.** Phase 1의 Step 5에서 선택된 가장 좋은 해의 위치해에서 1의 값을 가지는 결정 변수들을  $y_{(1)}, y_{(2)}, \dots, y_{(l)}$ 라고 정의한다.
- Step 2.** 모두  $l$ 개의 결정변수  $y_{(1)}, y_{(2)}, \dots, y_{(l)}$  중에 순서대로 하나의 결정변수만을 0으로 고정( $y_{(q)} = 0$ )한 후 (LP)를 풀고 얻은 해를 이용하여 (RP)를 푼다.
- Step 3.** Step 2에서 얻은  $l$ 개의 최종해를 비교하여 가장 좋은 해를 출력한다.

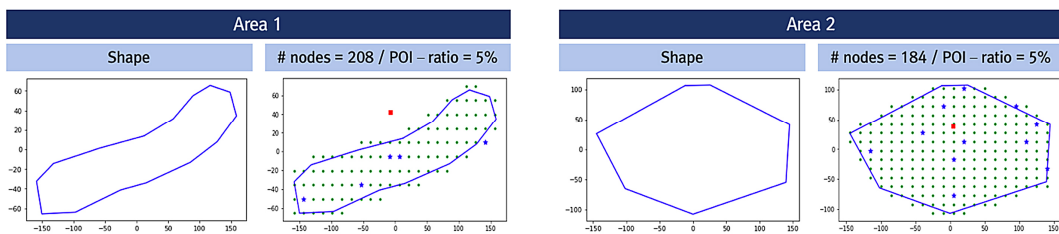
위 2단계 발견적 기법은 크게 랜덤하게 위치 문제의 해를 탐색하는 부분과 랜덤 탐색으로 얻은 가장 좋은 해의 주변해 (local neighborhood solutions)를 탐색하는 부분으로 이루어져 있다. 본 연구의 알고리즘은 실제 시스템에 적용하기 위해 알

고리즘 성능보다는 푸는 속도에 더 초점을 두었으며 빠른 시간안에 좋은 솔루션을 도출한다는 장점을 가진다. 새로운 수리 모형을 제시하고 이를 이용하여 실제 시스템에 적용가능함을 보여주는 것에 본 연구의 의의가 있으며 추가적인 알고리즘 개선에 관한 연구는 추후 연구과제로 제시하는 바이다. DLRwA를 풀기 위한 수리 모형 (P)와 2단계 발견적 기법의 성능 비교는 제 5장에서 자세히 설명한다.

**5. 계산 실험**

본 장에서는 수리 모형 (P)를 상용 MIP solver를 이용해서 직접 푸는 것과 2단계 발견적 기법과의 성능을 비교한다. 전산 실험은 Intel Core 3.5 GHz PC, 32GB RAM을 가지고 있는 MacPro에서 수행하였으며 MIP solver로는 IBM Cplex 12.6을 사용하였다. 2단계 발견적 기법의 구현에는 Python 3.6을 사용하였다.

전산실험에는 두 가지 감시 영역을 사용하였다. 첫 번째 감시 영역(Area 1)은 길고 구부러진 모양이며 두 번째 감시 영역(Area 2)은 둥근 타원형의 모양으로 가정하였다. <Figure 11>은 전산 실험에 사용된 두 가지의 감시 영역을 보여준다. 각각의 감시 영역은 앞에서 설명한 대로 그리드(grid) 노드를 사용하여 변환하였다. 본 전산실험에서는 임의로 전체 노드의 2% 또는 5%를 중요지점(POIs)으로 설정하였으며, 이는 중요지점(POIs)의 개수가 많고 적음에 따라 해가 어떻게 도출되는지 보여주기 위해 두 가지 경우로 나누어 실험을 진행했다. 드론의 이동시간  $T_{ij}$ 는 두 노드  $i$ 와  $j$ 간의 직선거리에 정비례(초)한다고 가정하였다. 본 연구에서 사용한 드론은 dji사의 phantom4로 최대 비행시간은 약 20분 정도이다. 드론의 배터리는 비행 방향, 속도에 따라 배터리 소모량이 다른데 dji는 드론이 비행을 마친 후 거리, 속도, 배터리 소모량 등을 저장해둔 flight log 데이터 파일을 제공한다. 따라서 이를 이용한 데이터 분석을 통해 거리당 평균적인 배터리 소모량을 계산하였다. 사진을 찍는데 필요한 시간  $T_j$ 는 모든 노드에서 3초로 가정하였다. 마지막으로 드론이 디포에 방문해서 배터리를 재충전하는 시간은 배터리를 통제로 교환할 수 있다고 가정하고 교환시간은 30초로 가정하였다(즉,  $R = 30$ ). 드론의 고도의 단계는 낮은 고도와 높은 고도의 2단계로 가정하였다. 낮은 고도에서는 9(= 3×3)개의 바닥 노드가 한 장의 사진에 포함된다고 가정하였고, 높은 고도에서는 25(= 5×5)개의 바닥 노드가 하나의 사진에 포함된다고 가정하였다.



**Figure 11.** Surveillance Areas Used in Experiments

<Table 1>과 <Table 2>는 Area 1과 Area 2에 대한 계산 실험 결과를 보여준다. 각각의 문제에 대해서 그리드의 간격을 다르게 지정하면 바닥 노드의 개수가 달라진다. 예를 들어, <Table 1>의 prob 1은 Area 1의 그리드의 간격이 가장 넓은 경우이고 prob 5는 간격이 가장 좁은 경우이다. 그리드의 간격은 드론이 얼마나 정밀하게 사진 찍을 위치를 결정해야 하는지에

따라 결정된다. 그리드의 간격이 좁을수록 중요지점의 위치를 더 세밀하게 지정할 수 있지만 일반적으로 문제를 푸는 시간은 증가한다.

<Table 1>과 <Table 2>의 Cplex(P)는 수리 모형 (P)를 Cplex를 이용하여 푸는 것을 뜻하고, Heuristic은 2단계 발견적 기법을 의미한다. 테이블의 각 열(column)의 의미는 다음과 같다.

**Table 1.** Computational Results for Area 1

	Parameter			Cplex(P)			Heuristic		Cplex/Heuristic	
	#node	bat	POI ratio	time(s)	sol	gap	time(s)	sol	time(s)	sol
prob1	90	800	2%	3600*	444.6	61.2%	3.7	444.6	<b>968.74</b>	<b>1.00</b>
	90	800	5%	3600*	449.4	62.0%	2.2	463.3	<b>1606.66</b>	0.97
	90	1500	2%	3600*	444.6	61.5%	3.7	444.6	<b>980.95</b>	<b>1.00</b>
	90	1500	5%	3600*	449.4	56.7%	2.8	450.4	<b>1287.48</b>	<b>1.00</b>
prob2	106	800	2%	3600*	508.0	65.5%	4.0	516.1	<b>900.19</b>	0.98
	106	800	5%	-	-	-	9.9	542.8		
	106	1500	2%	3600*	508.0	65.3%	5.8	516.1	<b>623.84</b>	0.98
	106	1500	5%	3600*	531.9	61.3%	2.5	555.1	<b>1463.92</b>	0.96
prob3	120	800	2%	-	-	-	6.7	537.6		
	120	800	5%	-	-	-	25.9	537.9		
	120	1500	2%	3600*	538.1	69.3%	20.7	537.6	<b>173.53</b>	<b>1.00</b>
	120	1500	5%	3600*	559.3	64.9%	9.5	538.7	<b>380.83</b>	<b>1.04</b>
prob4	132	800	2%	-	-	-	8.2	515.5		
	132	800	5%	-	-	-	30.8	536.8		
	132	1500	2%	3600*	515.2	67.8%	8.9	515.2	<b>406.74</b>	<b>1.00</b>
	132	1500	5%	3600*	542.4	63.9%	7.7	536.8	<b>466.31</b>	<b>1.01</b>
prob5	144	800	2%	3600*	543.7	71.2%	10.4	543.7	<b>344.89</b>	<b>1.00</b>
	144	800	5%	-	-	-	37.7	579.3		
	144	1500	2%	-	-	-	3.9	544.4		
	144	1500	5%	3600*	572.1	67.1%	36.9	585.2	<b>97.57</b>	0.98

**Table 2.** Computational Results for Area 2

	Parameter			Cplex(P)			Heuristic		Cplex/Heuristic	
	#node	bat	POI ratio	time(s)	sol	gap	time(s)	sol	time(s)	sol
prob1	80	500	2%	1292.0	328.9	0	3.4	328.9	<b>380.00</b>	<b>1.00</b>
	80	500	5%	3006.3	368.3	0	2.6	408.0	<b>1156.27</b>	0.90
	80	600	2%	578.0	328.9	0	2.6	328.9	<b>222.31</b>	<b>1.00</b>
	80	600	5%	1937.6	368.3	0	3.4	408.0	<b>569.88</b>	0.90
prob2	96	500	2%	-	-	-	5.3	425.4		
	96	500	5%	3600*	423.1	34.9%	15.5	423.1	<b>232.26</b>	<b>1.00</b>
	96	600	2%	3600*	368.1	26.8%	8.0	425.4	<b>450.00</b>	0.87
	96	600	5%	3600*	423.1	29.0%	4.0	423.1	<b>900.00</b>	<b>1.00</b>
prob3	102	500	2%	3600*	356.2	24.5%	2.9	356.2	1241.38	<b>1.00</b>
	102	500	5%	-	-	-	27.6	496.5		
	102	600	2%	3600*	356.2	20.1%	2.1	356.2	<b>1714.29</b>	<b>1.00</b>
	102	600	5%	-	-	-	16.7	496.5		
prob4	120	500	2%	-	-	-	4.1	466.7		
	120	500	5%	-	-	-	8.2	475.1		
	120	600	2%	-	-	-	4.1	466.7		
	120	600	5%	-	-	-	7.0	486.2		
prob5	136	500	2%	-	-	-	3.3	513.4		
	136	500	5%	-	-	-	5.6	719.7		
	136	600	2%	-	-	-	5.5	513.4		
	136	600	5%	-	-	-	9.3	588.7		

- #node : 감시 영역에 포함되는 전체 바닥노드의 개수
- bat : 한 번에 비행가능한 배터리의 용량
- POI ratio : 전체 바닥노드들 중에 중요노드의 비율, 이는 실험에서 중요한 바닥노드의 개수를 임의로 설정하기 위한 비율이며, 비율의 크기에 따라 고도를 조절하는 것이 아님.
- time(s) : 문제를 푸는데 걸리는 시간(초), 시간제한은 3600초 (1시간)
- sol : 각각의 알고리즘이 얻은 가장 좋은 해(best solution)
- gap :  $(\text{best\_solution} - \text{best\_lowerbound}) / \text{best\_solution} \times 100$  (%)
- Cplex/Heuristic : Cplex와 Heuristic의 성능 비교, 풀이시간 (time)의 경우 값이 1보다 크면 Heuristic이 Cplex보다 그 만큼 더 빠르다는 것을 의미함. 가장 좋은 해(sol)의 경우에는 값이 1보다 크면 Heuristic이 Cplex보다 좋은 해를 얻었음을 의미함.

테이블의 결과 중에 두꺼운 글자로 표시된 것들은 Heuristic이 Cplex보다 같거나 더 좋은 결과를 보였음을 의미하며 풀이 시간의 \*는 시간제한(3600초)에 도달했음을 의미한다. Cplex가 시간제한 내에 해를 찾지 못한 경우에는 -로 표시하였다.

계산 결과는 본 연구가 제시하는 2단계 발견적 기법이 Cplex보다 계산시간 면에서는 수백 배 이상 빠르고 얻은 해의 품질도 거의 같거나 더 좋은 것을 보여준다. 특히, Cplex는 많은 경우에 1시간의 시간제한 안에 유효해(feasible solution)를 전혀 찾지 못하는 경우가 많았다. Cplex는 문제의 크기가 아주 작은 경우(<Table 2>의 prob1)에만 최적해를 찾을 수 있었으며 나머지 경우에는 모두 매우 큰 gap을 보이거나 유효해를 찾지 못하였다. 반면에 2단계 발견적 기법은 매우 빠른 시간 안에 좋은 해를 찾았으며 Cplex가 전혀 유효해를 찾지 못한 경우에도 모두 빠르게 해를 찾을 수 있었다.

2단계 발견적 기법은 대체로 중요노드의 수가 많을수록 시간이 더 오래 걸리는 경향을 보였으며 배터리의 용량은 큰 차이를 보이지 않았다.

## 6. 드론 감시 시스템의 Proof-of-Concept 구현

본 장은 앞장에서 설명한 결과를 바탕으로 실제 구현한 드론 감시 시스템에 대한 설명이다. 본 시스템은 본 연구에서 제안하는 드론 감시 시스템이 실제로 구현될 수 있음을 보여주기 위해서 만들어졌기 때문에 가장 기본적인 기능만을 가지도록 설계하였다. 본 시스템은 DJI의 Phantom 4 (<https://www.dji.com/kr>) 드론을 사용하였다.

DJI의 Phantom 4 드론은 산업용 드론이 아닌 취미용 드론으로 PC와의 직접 연동을 지원하지 않는다. 하지만 DJI의 Phantom 4는 스마트폰과의 연동은 SDK를 제공하여 지원하고 있다. 따라서 본 연구에서 개발한 시스템은 DJI의 Phantom 4 드론과 스마트폰을 직접 연결하고 또한 스마트폰은 메시지큐(message queue)를 통해 PC의 웹 기반 시스템에 연결되는 비교적 복잡한 구조를 가진다. 시스템의 구현은 Java와 Python을 사용하였으며 메시지큐로는 RabbitMQ(<https://www.rabbitmq.com>)를 사용하였다.

<Figure 12>는 본 연구에서 구현한 웹 기반 드론 감시 시스템을 보여준다. 사용자는 Google map 위에 감시 영역을 설정한다. 설정된 감시 영역은 3차원상의 노드들의 집합으로 변환되어 2단계 발견적 기법을 통해 드론의 경로를 수립하는데 사용된다. 드론이 찍은 사진은 웹 기반 시스템으로 메시지큐를 통해 전송되고 화면에 표시된다. 본 시스템은 <Figure 3>의 작업 흐름을 바탕으로 구현되었기 때문에 사용자가 감시 구역을 한번만 설정하면 시스템은 자동으로 감시 구역 안의 중요 지역을 파악하며 중요지역은 높은 해상도의 사진을 얻도록 드론의 경로를 비행 시마다 수정한다. 즉, 드론의 경로는 매 비행마다 가장 최근에 파악된 중요지점을 고려하여 바뀌게 되어 각 비행 간의 시간 간격을 줄여 효율적인 지역 감시를 가능하게 한다.

<Figure 13>은 본 시스템을 이용해 얻은 실제 촬영된 사진들을 보여준다. 감시지역을 모두 포함하기 위해 6개의 사진 위치가 선정된 것을 볼 수 있다. 6개의 위치에서 찍은 사진들을 하나로 합치면 <Figure 12>의 오른쪽과 같은 감시 지역 전체의 사진을 얻을 수 있다.

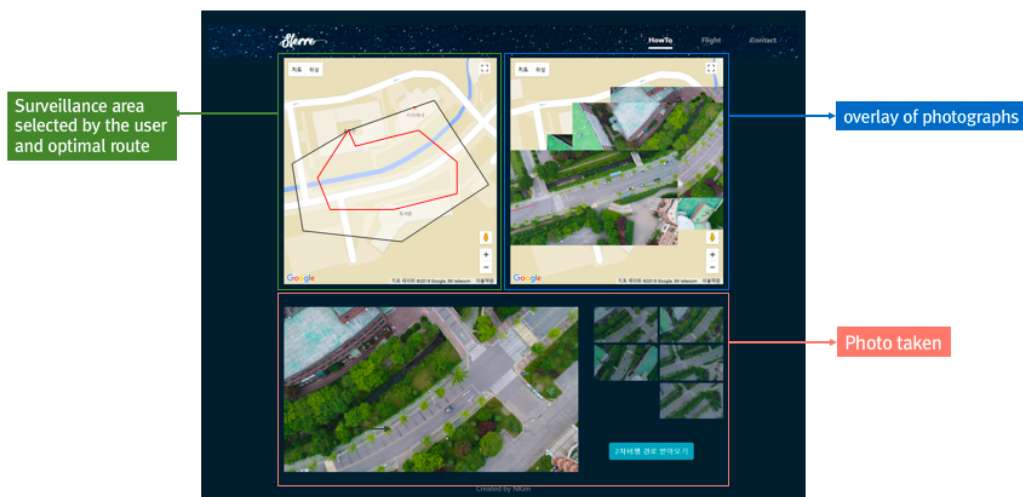


Figure 12. Web-based UI of Drone Surveillance System



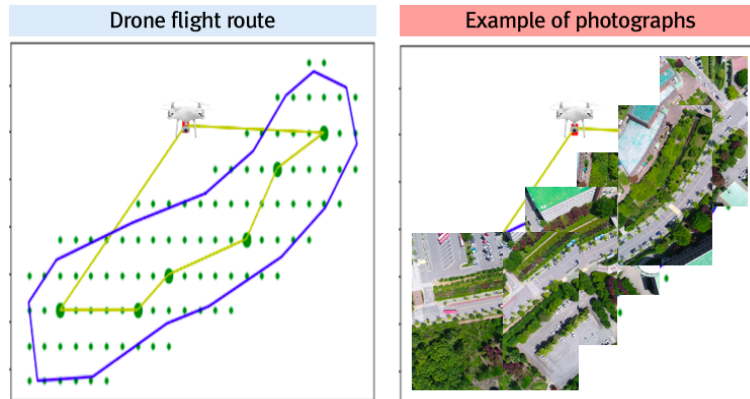


Figure 13. Example of Photographs

## 7. 결론

본 연구는 드론을 이용한 새로운 감시 시스템을 제안하였다. 기존 드론 감시 시스템은 사용자에게 의해 고정된 고도와 단순한 경로만을 사용한다. 이와 다르게 본 연구가 제안하는 드론 감시 시스템은 중요 지점을 자동으로 판단하고 중요 지점이 아닌 지역들은 높은 고도를 이용해서 빠르게 감시할 수 있는 장점이 있다. 또한, 감시해야 할 중요 지점을 파악하기 위해서 딥러닝 신경망 기반의 구분기(classifier)가 사용될 수 있음을 보였다. 더욱이 3차원 공간상의 드론 위치와 경로를 결정하기 위한 드론 위치 & 경로 문제(DLRwA)를 새롭게 제시하였고, 이를 풀기 위한 수리 모형과 매우 효율적인 2단계 발견적 기법 또한 제시하였다. 2단계 발견적 기법은 수리 모형을 Cplex로 바로 푸는 것보다 문제의 크기가 커질수록 훨씬 더 좋은 해를 찾음을 확인하였다.

본 연구에서 제시된 2단계 발견적 기법은 최적해를 보장해 주지 못한다. 최적해를 보장할 수 있는 알고리즘을 개발하는 것은 의미 있는 다음 연구 주제가 될 것이다. 그리고 드론의 운용 환경은 바람과 같은 외부의 요소가 드론의 비행에 크게 영향을 미칠 수 있다. 특히, 드론의 배터리는 외부 요인에 따라 효율이 크게 달라질 수 있는데 이러한 요소를 고려한 최적해 알고리즘을 후후 연구주제로 제시한다.

## 참고문헌

Bartashevich, P., Koerte, D., and Mostaghim, S. (2017), Energy-saving decision making for aerial swarms : PSO-based navigation in vector fields, *IEEE Symposium Series on Computational Intelligence (SSCI)*, 1-8.

Berman, O., Drezner, Z., Krass, D., and Wesolowsky, G. O. (2009), The variable radius covering problem, *European Journal of Operational Research*, **196**(2), 516-525.

Bürkle, A. and Essendorfer, B. (2010), Maritime surveillance with integrated systems, *Waterside Security Conference*.

DHL Press (2014), DHL parcelcopter launches initial operations for research purposes, *DHL*, [http://www.dhl.com/en/press/releases/releases\\_2014/group/dhl\\_parcelcopter\\_launches\\_initial\\_operations\\_for\\_research\\_purposes.html](http://www.dhl.com/en/press/releases/releases_2014/group/dhl_parcelcopter_launches_initial_operations_for_research_purposes.html).

Ding, G., Wu, Q., Zhang, L., Lin, Y., Tsiftsis, T. A., and Yao, Y. (2018), An Amateur Drone Surveillance System Based on the Cognitive Internet of Things, in *IEEE Communications Magazine*, **56**(1), 29-35.

Dorling, K., Heinrichs, J., Messier, G. G., and Magierowski, S. (2016), Vehicle Routing Problems for Drone Delivery, *IEEE Transactions on Systems, Man, and Cybernetics : Systems*, **47**(1), 70-85.

Erdoğan, S. and Miller-Hooks, E. (2012), A green vehicle routing problem, *Transportation Research Part E : Logistics and Transportation Review*, **48**(1), 100-114.

Farahani, R. Z., Asgari, N., Heidari, N., Hosseini, M., and Goh, M. (2012), Covering problems in facility location : A review, *Computers & Industrial Engineering*, **62**(1), 368-407.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. (2004), An exact algorithm for the elementary shortest path problem with resource constraints : Application to some vehicle routing problems, *Networks*, **44**(3), 216-229.

Flammini, F., Naddei, R., Pragliola, C., and Smarra, G. (2016), Towards Automated Drone Surveillance in Railways : State-of-the-Art and Future Directions, *Advanced Concepts for Intelligent Vision Systems, ACIVS, Lecture Notes in Computer Science*, **10016**, 336-348.

Freeman, P. K. and Freeland, R. S. (2015), Agricultural UAVs in the U.S. : potential, policy, and hope, *Remote Sensing Applications : Society and Environment*, 35-43.

Garey, M. R. and Johnson, D. S. (1979), *Computers and Intractability*, *WH Freeman, New York*.

Ji, T., Chen, S., Varma, R., and Kovačević, J. (2015), Energy-efficient route planning for autonomous aerial vehicles based on graph signal recovery, *53<sup>rd</sup> Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, 1414-1421.

Kim, E., Lee, J., and Shin, K. G. (2013), Real-time prediction of battery power requirements for electric vehicles, *ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, Philadelphia, PA, 11-20.

Kim, Y., Park, S., Kwon, Y., Lee, S., Choi, W., and Lee, C. (2017), Multi-City Flight Route Optimization System Using Big-Data Analytics, *Journal of the Korean Institute of Industrial Engineers*, **44**(2), 92-101.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012), Imagenet classi-

- fication with deep convolutional neural networks, *In Advances in Neural Information Processing Systems*, 1097-1105.
- Lee, C., Lee, K., and Park, S. (2012), Robust vehicle routing problem with deadlines and travel time/demand uncertainty, *Journal of the Operational Research Society*, **63**(9), 1294-1306.
- Majid, K., Mohammed, A., Tseng, C. M., and Chau, C. K. (2017), Autonomous Inductive Charging System for Battery-operated Electric Drones, In Proceedings of the Eighth International Conference on Future Energy Systems (e-Energy '17), *ACM, New York, NY, USA*, 322-327.
- Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960), Integer programming formulation of traveling salesman problems, *Journal of the ACM (JACM)*, **7**(4), 326-329.
- Murray, C. C. and Chu, A. G. (2015), The flying sidekick traveling salesman problem : Optimization of drone-assisted parcel delivery, *Transportation Research Part C*, **54**(C), 86-109.
- Nonami, K., Kendoul F., Suzuki S., Wang, W., and Nakazawa, D., Eds. (2010), *Autonomous Flying Robots : Unmanned Aerial Vehicles and MicroAerial Vehicles*, Springer Publisher.
- Otto, A., Agatz, N., Campbell, N., Golden, J., and Pesch, E. (2018), Optimization Approaches for Civil Applications of Unmanned Aerial Vehicles (UAVs) or Aerial Drones : A Survey, *Networks*, **72**(4), 411-58.
- Popper, B.(2015), Drones could make Amazon's dream of free delivery profitable, <http://www.theverge.com/2015/6/3/8719659/amazon-prime-air-drone-delivery-profit-free-shipping-small-items>.
- Rachel, L. and Finn, D. W. (2012), Unmanned aircraft systems : Surveillance, ethics and privacy in civil applications, *Computer Law & Security Review*, **28**(2), 184-194.
- Rosten, H., Reid, P. R., and Drummond, T. (2010), Faster and better : a machine learning approach to corner detection, *in IEEE Trans. Pattern Analysis and Machine Intelligence*, **32**(1), 105-119.
- Sebbane, Y. B., Ed. (2015), *Smart Autonomous Aircraft : Flight Control and Planning for UAV*, CRC Press.
- Tseng, C. M., Chau, C. K., Elbassioni, K., and Khonji, M. (2017), Flight Tour Planning with Recharging Optimization for Battery-operated Autonomous Drones, <https://arxiv.org/abs/1703.10049>.
- Wang, X., Poikonen, S., and Golden, B. (2016), The vehicle routing problem with drones : several worst-case results, *Optimization Letters*, **11**(4), 679-697.
- Yann, L. C., Yoshua, B., and Geoffrey, H. (2015), Deep learning, *Nature*, **521**(7553), 436-444.
- Zaloga, S. J., Rockwell, D., Finnegan, P. (2014), *World Unmanned Aerial Vehicle Systems-Market Profile and Forecast*, Teal Group Corporation.

## 저자소개

**이유경** : 한국외국어대학교 산업경영공학과에서 2019년 학사 학위를 취득하고 고려대학교 산업경영공학과 석사과정에 재학 중이다. 연구분야는 최적화, 머신러닝이다.

**강문정** : 한국외국어대학교 산업경영공학과에서 2019년 학사 학위를 취득하고 한국외국어대학교 산업경영공학과 석사과정에 재학 중이다. 연구분야는 최적화이다.

**김나영** : 한국외국어대학교 산업경영공학과에서 2019년 학사 학위를 취득하고 Technology University of Delft, Engineering and Policy Analysis 석사과정에 재학 중이다. 연구분야는 데이터 분석 및 프로그래밍이다.

**이충목** : 고려대학교 물리학과에서 1997년 학사, 1999년 석사학위를 취득하고 KAIST에서 2009년 산업공학 박사학위를 취득하였다. ETRI 선임연구원, IBM Research-Ireland에서 Research Staff Member등을 역임하고 2015년부터 한국외국어대학교 산업경영공학과에서 교수로 재직 중이다. 관심 연구분야는 최적화 이론, 정수계획법, 강건최적화, 데이터 마이닝 등이다.