

Improvement of Processing Speed for UAV Attitude Information Estimation Using ROI and Parallel Processing

Seok-Wun Ha*, Myeong-Chul Park**

*Professor, Dept. of Aerospace & Software Engineering, Gyeongsang National University, Jinju, Korea

**Professor, Dept. of Avionics Engineering, Kyungwoon University, Gumi, Korea

[Abstract]

Recently, researches for military purposes such as precision tracking and mission completion using UAVs have been actively conducted. In particular, if the posture information of the leading UAV is estimated and the mission UAV uses this information to follow in stealth and complete its mission, the speed of the posture information estimation of the guide UAV must be processed in real time. Until recently, research has been conducted to accurately estimate the posture information of the leading UAV using image processing and Kalman filters, but there has been a problem in processing speed due to the sequential processing of the processing process. Therefore, in this study we propose a way to improve processing speed by applying methods that the image processing area is limited to the ROI area including the object, not the entire area, and the continuous processing is distributed to OpenMP-based multi-threads and processed in parallel with thread synchronization to estimate attitude information. Based on the experimental results, it was confirmed that real-time processing is possible by improving the processing speed by more than 45% compared to the basic processing, and thus the possibility of completing the mission can be increased by improving the tracking and estimating speed of the mission UAV.

▶ **Key words:** UAV, Attitude Information, Estimation, ROI, Parallel Processing

[요 약]

최근 UAV를 활용하는 정밀 추적이나 임무완수 등의 군사 목적의 연구가 활발하게 진행되고 있다. 특히 앞서가는 유도 UAV의 자세 정보를 추정하고 이 정보를 이용하여 임무 UAV가 스텔스로 따라가서 자신의 임무를 완수하는 기능이 필요한 경우에는 유도 UAV의 자세 정보 추정 속도를 실시간으로 처리해야만 한다. 최근까지 영상처리와 칼만 필터를 사용해서 앞서가는 유도 UAV의 자세정보를 정밀하게 추정하는 연구가 수행되어 왔으나 처리과정의 순차처리로 인해 처리속도에 있어 문제점이 있어왔다. 따라서 본 연구에서는 영상 처리에 있어 처리영역을 전체영역이 아닌 물체를 포함하는 ROI 영역으로 한정하고 또한 연속적인 처리 과정을 OpenMP 기반의 멀티스레드로 분배하고 스레드동기를 맞추어서 병렬 형태로 처리함으로써 자세정보 추정 속도를 향상시킬 수 있는 방법을 제안한다. 구현 결과를 통해서 기존의 처리에 비해 45%이상 처리 속도를 향상시킴으로써 실시간처리가 가능하게 되어 임무 UAV의 추적 기능 향상을 통한 임무 완수 가능성을 증가시킬 수 있음을 확인하였다.

▶ **주제어:** 무인항공기, 자세정보, 추정, 관심영역, 병렬처리

-
- First Author: Seok-Wun Ha, Corresponding Author: Seok-Wun Ha
 - *Seok-Wun Ha (swha@gnu.ac.kr), Dept. of Aerospace & Software Engineering, Gyeongsang National University
 - **Myeong-Chul Park (africa@ikw.ac.kr), Dept. of Avionics Engineering, Kyungwoon University
 - Received: 2020. 12. 07, Revised: 2021. 01. 14, Accepted: 2021. 01. 14.

I. Introduction

최근 UAV를 운송, 보안 등의 산업이나 군사용으로 응용하기 위해서 그에 적합한 운용방법들이 여러 형태로 개발되고 있다[1-4]. 그리고 UAV에 장착되어 있는 카메라를 통해서 취득하는 비디오 영상을 기반으로 여러 가지 영상 처리[5-9]를 통해 영상에 내재되어 있는 각종 정보들을 추출하여 재해방이나 보안, 운행제어 등에 활용하고 있기도 하다. 또한 UAV가 군사용으로 활용되면서 UAV의 공격에 대응하는 시스템이 개발됨으로써 이에 대응하는 UAV 운행방법이 새롭게 개발되어야 할 필요가 증가하게 되었다. 이것의 한 일환으로 Paul [10-11] 등은 템플릿 매칭과 여러 가지 영상처리 알고리즘, 그리고 선형회귀와 칼만 필터를 접목하여 앞서가는 UAV의 비행자세 정보를 뒤따르는 UAV의 카메라를 통해 촬영하고 이로부터 앞서가는 UAV의 비행자세정보를 추정하여 그 정보로 자신의 비행을 자율적으로 스텔스하게 운행하게 할 수 있게 함으로써 최종 목적지에 대한 임무를 수행할 수 있게 할 수 있는 비행방법을 연구하여 제안한 바가 있다. 그러나 이 방법에서는 여러 단계의 처리과정을 순차처리 방식을 취함으로써 전체처리속도가 저하되어 실시간 처리에 어려움을 가지고 있었다.

본 논문에서는 처리과정의 첫 번째 작업인 컬러 분리나 템플릿 정합 과정에서 영상 전체에 대해 이들 처리를 행하기 때문에 많은 처리시간을 소모하고 있는 것을 개선하기 위해서 UAV를 포함하는 특정한 영역을 한정하여 ROI 영역을 설정하고 이 영역에 대해서만 위의 처리를 수행하게 함으로써 처리시간을 단축하고자 한다. 또한 전체 처리과정에서 동시에 처리해야 하는 부분들을 OpenMP를 기반으로 하는 멀티스레드로 할당하여 병렬로 처리하고 다음 단계에 입력되기 전에는 멀티스레드 동기화를 통해 이들 여러 출력들이 동시에 전달되도록 함으로써 순차방식의 처리를 병렬처리 방식으로 바꾸어 실시간처리가 되도록 전체적인 처리속도를 향상시킬 수 있는 방법을 적용하고자 한다.

본 논문의 구성은 2장에서는 기존의 Paul 등이 제안한 UAV의 비행자세정보 추정시스템의 구조와 처리과정에 대해 그 방법들을 살펴보고 3장에서는 자세추정 성능, 특히 처리속도를 실시간으로 향상시키기 위한 기법들과 처리과정에 대해 나타내며, 4장에서는 실험을 통해서 제안하는 방법들을 적용할 경우 UAV의 자세정보를 실시간으로 추정할 수 있음을 보이는 실험결과를 나타내고, 마지막으로 5장에서 결론과 향후연구에 대해 기술한다.

II. Estimation of UAV Attitude Information

1. Structure of Attitude Estimation System

지상관제시스템으로부터의 유도 및 제어가 없이 목표지점까지 운항을 할 수 있기 위해, 앞서가는 유도무인기의 자세를 추정하여 자신의 임무 비행을 스스로 유도하기 위한 자세 추정 시스템의 구조는 Fig. 1과 같으며, 전체 과정은 다음과 같다.

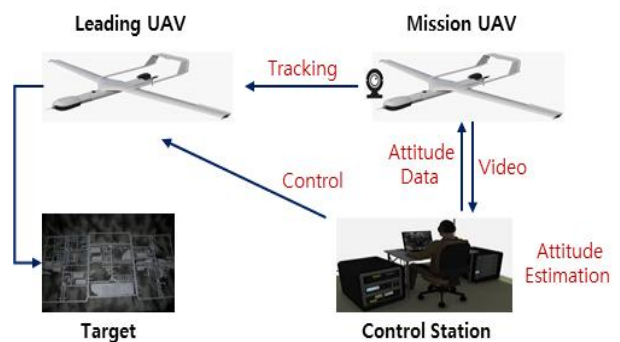


Fig. 1. Structure of UAV Attitude Information Estimation System for Mission Flight

- 1) 지상관제시스템에서는 유도무인기를 조정하여 목표지점까지 비행을 시킨다.
- 2) 임무무인기는 장착된 카메라로 유도무인기를 촬영하여 지상관제시스템으로 영상을 전송한다.
- 3) 지상관제시스템은 비디오정보를 분석하여 유도무인기의 비행자세정보를 추정하고, 이 정보를 임무무인기에 전송한다.
- 4) 추정자세정보를 전송받은 임무무인기는 이 정보를 사용하여 유도무인기를 추적한다.
- 5) 최종적으로 임무무인기가 목표물에 안착함으로써 임무를 완수한다.

Fig. 1의 구조에서는 임무무인기와 지상관제시스템과의 상호통신을 통해 자세정보의 추정과 전송이 이루어지고 있지만, 관제시스템과의 연동이 없이 임무무인기 자체에서 직접 자세추정처리를 실행하게 한다면 외부의 통신방해나 제밍에 상관없이 자율비행을 통해 임무를 수행할 수 있을 것이다.

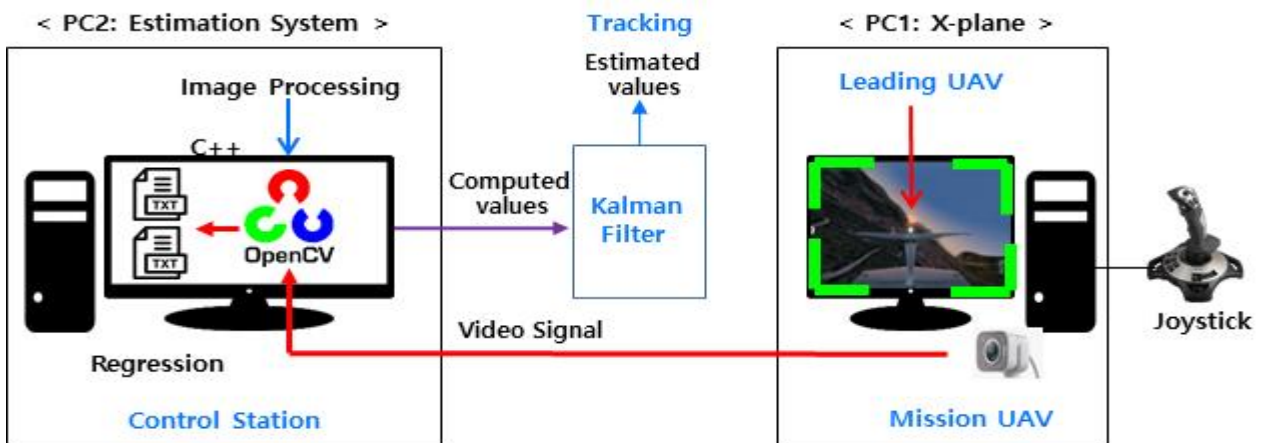


Fig. 2. Overall Configuration of System for Attitude Information Estimation of Leading UAV

2. Process of Attitude Estimation

유도무인기의 자세정보를 추정하기 위한 하드웨어 장치와 소프트웨어 처리를 위한 전체 구성은 Fig. 2와 같다. Fig. 2에서 유도무인기의 자세정보를 처리하기 위한 하드웨어 구성은 다음과 같다. 먼저 유도무인기는 PC1에서 X-plane 시뮬레이터를 사용해서 화면상에 생성되며, 유도무인기의 비행은 조이스틱을 사용하여 실행한다. 사용한 무인기의 기종은 세스나-30이다. 또한 임무무인기는 별도로 생성하지 않으며, 가상으로 임무무인기가 있고 그 위에 카메라가 장착된 것으로 가정한다. 그리고 카메라에서 촬영된 유도무인기의 비디오영상은 지상관제시스템인 PC2로 전송되고, 여기서 비디오 영상을 분석하여 유도무인기의 자세정보를 계산한다. 계산된 자세정보는 칼만 필터로 입력된다. 그리고 자세추정을 위한 소프트웨어처리는 C++ 환경에서 OpenCV 라이브러리에서 제공하는 함수들을 활용하고, 통계적 처리와 영상처리 알고리즘들을 적용하여 이루어진다. 그리고 통계적 처리와 영상처리로부터 추출된 자세정보는 칼만 필터와 같은 예측 알고리즘을 통해 최종적으로 정밀하게 추정된다.

Paul 등이 제안한 비행정보 추정을 위한 처리과정은 다음과 같다.

- 1) 임무무인기에 장착된 카메라로부터 촬영한 유도무인기의 비행영상을 지상관제시스템의 비행정보 추정시스템에 전송한다.
- 2) 비디오영상의 연속하는 각 프레임의 전체 영상영역에 대해 컬러모드를 HSV모드로 변환하고 색상분할(color segmentation) 방법을 적용하여 유도무인기를 포함하는 무인기영역을 추출한다.
- 3) 탐지된 무인기영역에 대해 사전에 저장해둔 유도무인기 템플릿을 사용하여 템플릿정합(template

matching) 기법으로 무인기를 정확하게 탐지한다.

- 4) 이진영상으로부터 캐니에지(Canny edge) 기법을 적용하여 정합무인기의 에지를 추출한다. 여기에 허프 변환(Hough transform)을 적용하여 수평선과 무인기의 꼬리날개 선을 결정한다.
- 5) 또한 이진영상에 해리스코너(Harris corner) 탐지기법을 적용하여 무인기 날개의 양 끝점과 꼬리날개의 최고점을 추출하고, 무인기 날개 선을 추출한다.
- 6) 수평선과 무인기 날개 선 사이의 각도를 기반으로 무인기의 롤(roll) 각을 계산한다.
- 7) 해리스코너 점들을 기반으로 수평 날개의 길이를 계산하고, 이 수평 날개의 길이 변화를 기반으로 무인기의 요(yaw)각을 계산한다.
- 8) 해리스코너 점들을 기반으로 수직 날개의 길이를 계산하고, 이 수직 날개의 길이 변화를 기반으로 무인기의 피치(pitch)각을 계산한다.
- 9) 계산되어진 유도무인기의 비행정보인 롤, 요, 피치 값에 선형회귀(linear regression)를 적용하여, X-plane에서 발생시킨 비행정보 값들에 근사시킨다.
- 10) 이 근사 값들을 칼만 필터에 적용하여 최종적으로 유도무인기의 비행자세정보를 추정한다. 이 추정 값들은 임무무인기의 비행을 위한 제어 값으로 제공된다.

위의 전체 유도무인기의 비행정보 추정을 위한 처리과정을 블록도로 나타내면 다음 Fig. 3과 같으며, 연산처리는 모든 부분들이 위의 정해진 순서대로 순차적으로 처리가 된다.

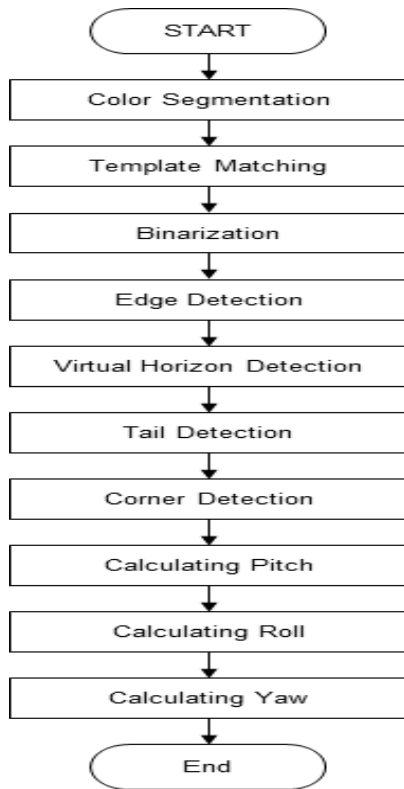


Fig. 3. Flow Diagram for Extracting Flight Information of Leading UAV

Fig. 3의 과정을 통해 계산되어진 피치, 롤, 요 데이터들에 대해 각각 선형회귀분석을 적용해서 위의 데이터 값들을 X-Plane에서의 원래 데이터 값의 범주로 정합시킨다. 그런 다음 같은 필터를 통해서 유도무인기의 비행정보를 정확하게 추정하게 된다.

III. Improvement of Attitude Estimation Performance

Paul 등이 비행정보를 추정하기 위해 제시한 Fig. 2의 처리과정에서는, 첫째로는 영상의 연속하는 프레임마다 영상전체에 걸쳐서 유도무인기의 템플릿 정합처리를 함으로써 처리시간이 많이 걸리는 문제가 있으며, 둘째로는 예를 들어 템플릿 정합이후에 에지탐지와 이진화가 동시에 처리되고, 다음 단계에서는 에지탐지 결과로부터 꼬리선 탐지와 가상수평선 탐지, 그리고 이진화로부터 코너 점 탐지가 동시에 처리되어야 함에도 실질적으로는 이들이 순차적으로 처리됨으로 말미암아 전체 처리속도가 저하되어 실시간 처리가 어려운 문제를 가지고 있다. 이에 본 논문에서는 첫 번째 프레임 영상에서 유도무인기 영역을 유도

무인기를 포함하는 관심영역(ROI, region of interest)을 결정하고 다음 프레임부터는 이 ROI 영역을 기준으로 약간의 변화영역에 대해서만 템플릿정합을 유도함으로써 처리시간을 감소시키고, 또한 전체 처리과정의 단계별 계산을 멀티스레드를 적용하여 병렬처리를 실행함으로써 단계별 대기시간을 줄임으로써 전체적인 처리속도를 개선하고자 한다.

1. ROI Processing

관심영역에 대한 프레임 사이의 영역 결정을 위한 처리는 다음 Fig. 4와 같이 결정된다.

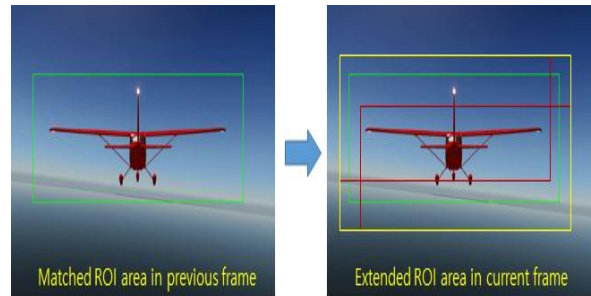


Fig. 4. Structure for detecting ROI area from consecutive two frames

현재프레임의 영상영역 전체에 대해 템플릿 정합 계산을 하지 않고 이전프레임에서 추출된 ROI영역에 대해 다음 식 (1)에서와 같이 상하 좌우로 8분의 1 크기로 약간 확장된 영역에 대해서만 현재프레임에서 정합을 계산한다.

$$\begin{aligned} \text{previous frame ROI: } & W \times H \\ \text{current frame ROI: } & (W + \frac{1}{4}W) \times (H + \frac{1}{4}H) \end{aligned} \quad (1)$$

이것은 연속하는 프레임사이의 움직임 변동에 대한 영역이동이 미세하다는 점을 바탕으로 실험적으로 결정된 것이다. 따라서 현재프레임에서는 이전프레임에서 결정된 ROI 영역의 확장영역에 대해서만 정합처리를 함으로써 영상처리와 비행정보 추정단계에 앞서 무인기영역 추출단계에서부터 처리속도를 줄일 수 있게 된다.

2. Parallel Processing

Fig. 3의 처리과정에서 동시에 처리할 수 있는 부분들을 스레드로 할당하면 부분적으로 병렬처리가 가능하게 된다. OpenMP를 사용하여 전체처리 중에서 병렬로 동시에 처리되어야 할 부분들을 멀티스레드로 할당하고 마지막 비

행정보 값들을 계산하기 전에 스레드 동기화를 시킴으로써 최종 비행 값들이 시간적으로 거의 동시에 계산되도록 하면 전체처리 시간이 단축되어 실시간 처리가 가능할 것이다. OpenMP를 사용하여 멀티스레드 기반의 병렬처리 과정은 다음 Fig. 5와 같이 진행된다.

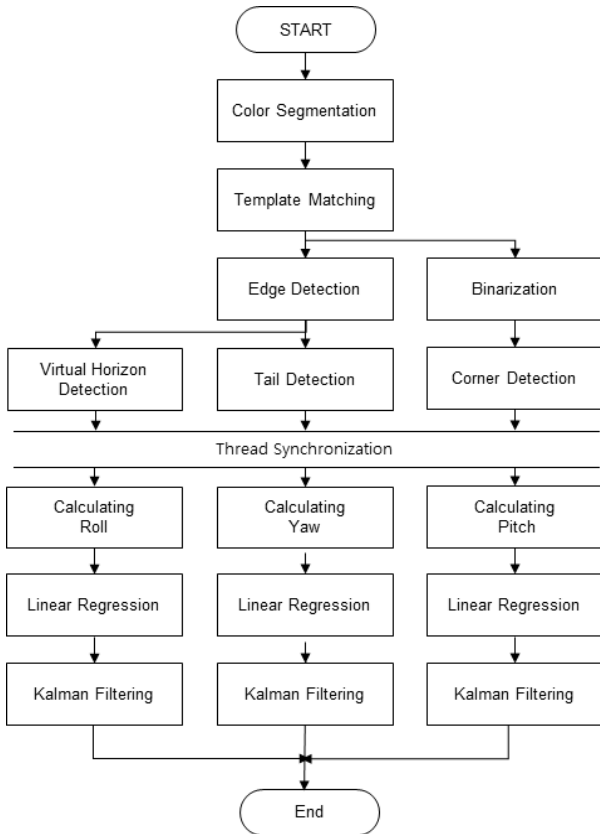


Fig. 5. Parallel Processing Flow Diagram for Estimating Flight Information of the Leading UAV Using OpenMP

- 1) 임무무인기의 카메라로부터 전송되어 오는 비디오 영상으로부터 컬러분리를 통해 물체영역과 배경영역을 구분한다.
- 2) 첫 프레임 영상의 전체영역 중에서 유도무인기의 물체 영역을 찾은 다음, 이후 프레임부터는 물체영역에서 약간 확장된 ROI 영역을 기반으로 템플릿정합을 통해 물체영역을 결정한다.
- 3) 에지 탐지와 이진화가 동시에 처리되도록 멀티스레드를 할당하여 계산한다.
- 4) 에지탐지로부터 가상수평선 탐지와 꼬리선 탐지를 멀티스레드로 할당하고, 이진화로부터 코너 점 탐지에 단일 스레드를 할당하여, 이들이 동시에 계산되도록 한다.
- 5) 위의 세 가지 처리는 각기 처리시간이 다르기 때문에

모두가 처리될 때까지 대기해야 하고 다음 단계의 멀티스레드에 동시에 입력되도록 실행하기 위해서 스레드 동기화를 시행한다.

- 6) 다음 단계인 롤, 요, 피치 데이터의 계산에 멀티스레드를 적용하여 이들이 동시에 계산되도록 한다.
- 7) 다음 단계로 각각의 결과에 대해 멀티스레드를 적용하여 선형회귀분석 처리를 동시에 수행해서 X-Plane 데이터와 값의 범위를 일치시킨다.
- 8) 마지막으로 각각의 선형회귀 예측 결과에 대해 칼만 필터를 멀티스레드로 동시에 처리함으로써 최종적으로 유도무인기의 비행 정보를 정확하게 추정한다.

IV. Experiments and Results

1. Results of ROI Process

영상전체가 아닌 영상내의 ROI 영역에 대해서만 템플릿 정합 계산을 행할 경우의 평균 실행시간을 측정한 결과, 다음 Table 1에서와 같이 실행시간이 약 55%로 감소되었다. 현재는 유도무인기의 영역이 영상 전체 영역의 약 3분의 1에 해당하는 영역을 차지하고 있지만, 카메라의 설치 거리에 따라 유도무인기의 크기가 영상의 전체 영역에 비해 비행정보 추출에 영향이 없을 정도로 훨씬 작아질 수 있기 때문에 실질적으로는 처리시간이 더 감소될 수 있을 것이다.

템플릿 정합에 걸리는 시간을 영상 전체에 대한 처리시간과 ROI 영역에 대해서만 처리하는 시간을 비교하여 다음 Table 1과 같다.

Table 1. Comparison of Processing Time Results of Overall and ROI Treatments

Processing Time(ms)	
Whole Area	14.5
ROI Area	8.01

ROI 영역 내에서의 정합과 OpenMP의 멀티스레드를 적용하여 비행정보를 얻기 위한 영상처리의 각 단계 중에서 일부에 대해 첫 프레임부터 다섯 번째 프레임까지의 처리 시간을 측정한 결과를 다음 Table 2와 같다.

Table 2. Results of Processing Time When ROI Area Matching and Multi-threading of OpenMP are Applied

Process of Image processing	Run Time(ms)				
	Loop1	Loop2	Loop3	Loop4	Loop5
Color Segmentation	21	4	5	3	3
Template Matching	10	8	6	5	7
Binary	2	3	3	3	3
EdgeDetection	6	2	6	2	2
CornerDetection	10	10	11	9	10
TailDetection	5	4	4	4	6
VirtualHorizonDetection	7	9	8	7	7
Total	61	40	43	33	38

Table 2에서 첫 프레임에서는 영상 전체영역에 대해 컬러분리와 템플릿정합을 시행하기 때문에 시간이 걸리지만 이후 프레임부터는 ROI 영역에 대해서만 처리되기 때문에 처리시간이 대폭 줄어드는 것을 볼 수 있다. 이 실험에서는 유도무인기 물체의 크기가 영상의 약 절반 영역을 차지할 때의 결과이다. 그리고 에지 탐지, 코너 점 탐지, 꼬리 선 탐지는 동시에 처리되지만 처리시간이 각각 다르기 때문에 다음 단계인 비행정보 데이터 계산을 실행하기 위한 멀티스레드에 동시에 입력시키기 위해서는 Fig. 5에서 나타난 것처럼 중간에 스레드 동기화가 필요하게 된다.

최종적으로 템플릿정합을 영상의 전체영역이 아닌 ROI 영역에 대해서만 처리하는 방법과 OpenMP의 멀티스레드 할당을 적용하는 병렬처리를 통해서 전체처리 속도가 시리시간으로 향상될 수 있었다. 처리속도 향상 결과를 프레임속도로 다음 Table 3에 나타내었다.

Table 3. Results of Frame Rate per Second through ROI Processing and Multi-threaded Parallel Processing

OpenMP	Whole Area	ROI Area
Not applied	19 fps	23 fps
Applied	21 fps	28 fps

Table 3에서, ROI 처리와 병렬 처리를 적용하지 않은 경우 [1]의 프레임속도는 19 fps 이었으며, 여기에 ROI 처리를 적용하면 23 fps로 약 20% 향상되고, 다시 OpenMP를 적용하면 28 fps로 약 47% 향상되어 실시간 처리가 가능하게 되었음을 볼 수 있다. 이 경우는 유도무인기의 물체 영역이 영상 전체의 약 3분의 1의 크기일 때의 실험 결과이다.

V. Conclusions

본 논문에서는 Paul 등[10-11]이 제안한 유도무인기의 비행자세정보를 추정에서 처리과정의 순차적 처리로 인해 전체처리가 실시간으로 이루어지지 않는 단점을 보완하기 위해서 템플릿 정합 시에 영상의 전체 영역이 아닌 물체를 포함하는 한정된 ROI 영역에 대해서만 정합을 실시하고, 또한 처리과정에서 동시처리가 필요한 부분들에 대해 멀티스레드를 할당하여 병렬로 처리하게 함으로써 전체처리 속도를 향상시켜 실시간으로 비행정보 추정이 가능할 수 있었다. 본 연구를 기반으로 유도무인기의 실시간 자세추정을 통해서 임무무인기의 목적을 효과적으로 달성할 수 있는 가능성을 높일 수 있음을 발견하였다. 향후에는 카메라의 거리에 따른 실시간 비행정보 추정에 대해 연구할 필요가 있을 것이다.

ACKNOWLEDGEMENT

This work was supported by the Gyeongsang National University Fund for Professors on Sabbatical Leave, 2018.

REFERENCES

- [1] E. N. Barmounakis, E. I. Vlahogianni, J. C. Golias. "Unmanned Aerial Aircraft Systems for transportation engineering: Current practice and future challenges". International Journal of Transportation Science and Technology, Vol. 5, No. 3, pp. 111-122, 2016.
- [2] J. P. Lee, J. W. Lee, K. H. Lee. "A Scheme of Security Drone Convergence Service using Cam-Shift Algorithm". Journal of the Korea Convergence Society, Vol. 7, No. 5, pp. 29-34, 2016.
- [3] Y. Wang, J. Mangnus, D. Kostic, H. Nijmeijer, S. T. H. Jansen. "Vehicle State Estimation Using GPS/IMU Integration". IEEE SENSORS Proceedings, Vol. 1, pp. 28-31, 2011.
- [4] Su Hyun Nam, Myung Whan Choi, Hyo Hyun Choi, "Group Mobility Control Mechanism for Micro Unmanned Aerial Vehicle". Journal of The Korea Society of Computer and Information. Vol. 17, No. 4, pp. 99-107, 2012.
- [5] H. D. Cheng, X. H. Jiang, Y. Sun, J. Wang. "Color Image Segmentation: Advances and Prospects". Pattern Recognition, Vol. 34, No. 12, pp. 2259-2281, 2001.
- [6] J. Cui, J. Xie, T. Liu, X. Guo, Z. Chen. "Corners Detection on

- Finger Vein Images using the Improved Harris Algorithm". International Journal for Light and Electron Optics, Vol. 125, No. 17, pp. 4668-4671, 2014.
- [7] P. Mukhopadhyay, B. B. Chaudhuri. "A Survey of Hough Transform". Pattern Recognition, Vol. 48, No. 3, pp. 993-1010, 2015.
- [8] L. Ding, A. Goshtasby. "On the Canny Edge Detector". Pattern Recognition, Vol. 34, No. 3, pp. 721-725, 2001.
- [9] Hyun Jun Park, Kwang Baek Kim. "Improved k-means Color Quantization based on Octre". Journal of The Korea Society of Computer and Information, Vol. 20 No. 12, pp. 9-14, 2015.
- [10] Q. Paul, J. H. Hyeon, Y. H. Moon, S. W. Ha. "A Study on Attitude Estimation of UAV Using Image Processing". Journal of the Korea Convergence Society, Vol. 7, No. 5, pp. 29-34, 2017.
- [11] Q. Paul, Y. H. Moon, S. W. Ha. "Improvement of UAV Attitude Information Estimation Performance Using Image Processing and Kalman Filter". Journal of Convergence for Information Technology, Vol. 8, No. 6, pp. 135-142.

Authors



Seok-Wun Ha received the B.S., M.S. and Ph.D. degrees in Electronics Engineering from Busan National University, Korea, in 1979, 1981 and 1995, respectively. Dr. Ha joined the faculty of the Department of Computer

Science at Gyeongsang National University, Jinju, Korea, in 1993. He is currently a Professor in the Department of Aerospace & Software Engineering. He is interested in computer vision, embedded software and mobile programming.



Myeong-Chul Park received a B.S. degree in Computer Science from Korea National Open University in 1999, a M.S. and Ph.D. degrees in Computer Science from GyeongSang National University in 2002, 2007.

Dr. Park is currently a Professor in the Department of Avionics Engineering, KyungWoon University. He is interested in Visualization, Simulation, Education of Software, Virtual Reality, and Parallel Programming.