한국항행학회 논문지 제16권 제6호 2012년 12월 (JKONI 16(6): 910-919, Dec. 2012)

임베디드 프로세서를 이용한 고정익 무인항공기 영상기반 목표물 탐지 및 추적

Fixed-Wing UAV's Image-Based Target Detection and Tracking using Embedded Processor

김정호*, 정재원*, 한동인*, 허진우*, 조겸래*, 이대우*

Jeong-Ho Kim*, Jae-Won Jeong*, Dong-In Han*, Jin-Woo Heo*, Kyeom-Rae Cho*, and Dae-Woo Lee*

요 약

본 논문에서는 고정익 무인항공기의 온보드 영상처리 시스템 개발에 대하여 개발과정에 대해 기술하고, 비행실험을 통한 실험결과를 토대로 하여 성능을 검증하고자 하였다. 시스템 개발보드는 ARM 프로세서가 탑재된 영상처리용 보드에 임베디드 리눅스를 포팅하였다. 목표물 추적을 위한 영상처리 알고리즘으로는 비교적 간단한 알고리즘인 색상 기반 알고리즘을 적용하여, 지상에 있는 특정 색상의 물체를 추적하도록 하였다. 개발된 시스템의 성능검증을 위해 실험실에서 제작한 무인항공기에 탑재하여 비행실험을 수행하였으며, 실시간 성능 향상을 위해 영상처리 알고리즘 및 임베디드 리눅스의 커널에 대한 최적화 작업을 수행하였다. 비행실험 결과, 4픽셀 이내의 오차범위에서 지속적으로 목표물을 추적하는 것을 확인하였다.

Abstract

In this paper, we described development of on-board image processing system and its process and verified its performance through flight experiment. The image processing board has single ARM(Advanced Risk Machine) processor. We performed Embedded Linux Porting. Algorithm to be applied for object tracking is color-based image processing algorithm, it can be designed to track the object that has specific color on ground in real-time. To verify performance of the on-board image processing system, we performed flight test using the PNUAV, UAV developed by LAB. Also, we performed optimization of the image processing algorithm and kernel to improve real-time performance. Finally we confirmed that proposed system can track the blue-color object within four pixels error range consistently in the experiment.

Key words : Embedded Processor, On-board Image Processing, UAV, Color-based Object Tracking

I. 서 론

최근 무인항공기를 이용한 다양한 임무가 요구되고 있고, 무인항공기를 활용한 임무수행은 이전까지는 주로 국방 분야와 관련하여 군용 정찰 및 타격과

* 부산대학교 항공우주공학과(Dept. of Aerospace Eng., Pusan National University)
· 제1저자 (First Author) : 김정호(051-510-3036, kimsmap@pusan.ac.kr)
· 교신저자(Corresponding Author) : 이대우 (Dea-Woo Lee, +82-51-510-2329, baenggi@pusan.ac.kr)
· 투고일자 : 2012년 11월 6일
· 심사(수정)일자 : 2012년 11월 8일 (수정일자 : 2012년 12월 21일)
· 게재일자 : 2012년 12월 30일

같은 군사용으로 이루어졌지만, 최근에는 민간분야에서도 다양한 기관에서 산불 진화 및 감시, 농약 살포, 교통량 정보 수집 등 그 활용도 및 수요가 점차 높아지고 있다. 이러한 다양한 임무들을 수행하기 위해서는 무인항공기에 다양한 센서들을 탑재하여 여러 가지 정보들을 수집하게 되는데, 그 중에서도 특히 영상센서를 이용한 임무수행은 지상에서 시각적으로 모니터링 할 수 있다는 점과 광범위한 지역을 효과적으로 탐색할 수 있다는 장점으로 인해 가장 각광을 받고 있는 방법이라 할 수 있다[1].

일반적으로 무인항공기의 영상처리 방법은 운용중에 획득한 영상정보를 영상모뎀을 이용하여 지상국(GCS : Ground Control System)으로 전송하고 GCS에서는 그 영상정보를 받아 임무에 맞는 영상처리를 하고 처리 결과에 따른 제어 명령을 다시 무인항공기의 비행제어 시스템으로 전송하여 처리결과를 확인하게 된다. 이러한 방식에서 발생할 수 있는 문제점은 무인항공기가 GCS와의 통신이 두절되는 상황이 발생하게 되는 경우, GCS에서는 더 이상 명령을 전송할 수가 없기 때문에 더 이상의 임무수행이 힘들어질 수 있으며, 실시간으로 목표물을 탐지하고 추적하는 경우에 있어서 영상 전송 방식이 아날로그일 경우에는 외란으로 인한 잡음 성분 등의 원인으로 원본 영상이 고르지 못하거나 훼손될 가능성이 있으며, 디지털 방식일 경우 시간지연이 발생할 수 있기 때문에 실시간성을 보장할 수가 없다[2]. 더욱이 영상처리 같은 경우 일반적으로 연산량이 많고 계산과정이 복잡하기 때문에 실시간성을 보장하기 위해서는 그 효율성을 충분히 고려해야 한다[3]. 따라서 본 논문에서는 이러한 문제점들을 해결하기 위한 하나의 제안으로서 실험실 수준의 고정익 무인항공기에서 운용할 수 있도록 소형화, 경량화를 목표로 하여 임베디드 시스템을 이용한 영상센서기반 목표물 탐지 및 추적 시스템을 구현하였다. 본 시스템은 임베디드 프로세서가 탑재된 컨트롤러를 이용함으로써 고속 컴퓨터에 비해 소형화 및 경량화 된 온보드 형태로 무인항공기에 탑재하여 실제 비행 실험과 데스크탑 PC와의 비교를 통해 그 성능을 검증하고자 하였다.

Ⅱ. 온보드 영상처리 시스템 개발

2-1 온보드 영상처리 시스템 개요

본 논문에서는 임베디드 컨트롤러 기반의 무인항공기 영상센서기반 목표물 탐지 및 추적 시스템 개발을 위해 ARM 프로세서가 탑재된 상용 개발보드를 이용하여 진행하였다.

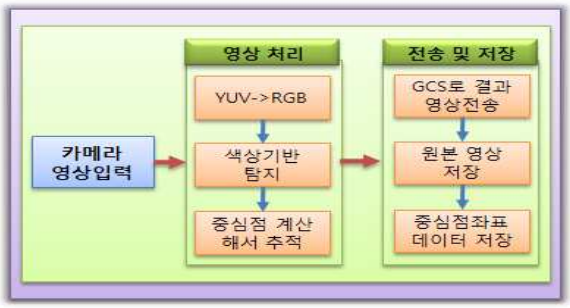


그림 1. 온보드 영상처리 시스템 개요도
Fig. 1. The schematic diagram of Onboard image processing system

그림1은 온보드 영상기반 탐지 및 추적 시스템의 작동 프로세스를 나타낸 것이다. 무인항공기에 탑재된 김벌 카메라로부터 입력된 영상을 RGB 포맷으로 변환하여 특정 색상을 가진 물체를 지속적으로 추적하는 영상처리를 수행한다. 색상기반 영상추적 알고리즘에 의해서 탐지 및 추적이 수행되는 결과에 대한 데이터는 SD메모리에 저장이 되어 분석이 가능하며, 임무 수행 영상은 영상 모뎀을 통해 GCS로 송신되어 실시간으로 처리 결과를 모니터링 할 수 있게 된다.

2-2 온보드 영상처리 시스템 구성

본 논문에서 온보드 영상처리 시스템 개발을 위해 사용된 영상처리용 개발보드는 그림 2의 상용 임베디드 개발보드인 Android S3C6410 개발보드를 이용하여 목적에 맞게 시스템 변경 작업을 수행하였다.

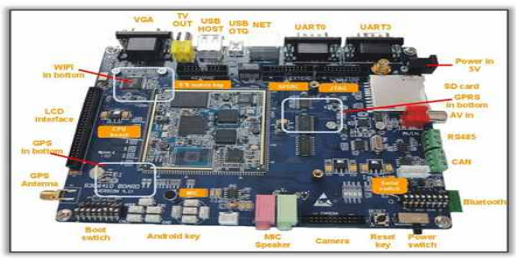


그림 2. S3C6410 개발보드
Fig. 2. S3C6410 development board

하드웨어 플랫폼은 온보드 영상처리 시스템 개발을 위해 사용한 개발보드와 기타 주변 장치들로 구성되어 있다. S3C6410 개발보드는 ARM11 기반의 ARM 프로세서가 탑재된 개발보드로서, 특히 멀티미디어 기능에 있어서 다중 표준 코덱(MFC : Multi Function Codec)등 다양한 기능을 지원한다[4].

표1. S3C6410 개발보드 사양
Table 1. The specifications of S3C6410 Target board

Hardware Features	
CPU	Samsung S3C6410 (ARM1176JZF-S up to 667MHz)
Memory	256MB mobile 266MHz DDR SDRAM / 1GB NAND Flash
Size	175mm×140mm
Weight	260g(included LCD Pannel)

온보드 영상처리 시스템 개발을 위한 소프트웨어 구성은 시스템의 전체적인 흐름을 제어할 수 있는 임베디드 리눅스 운영체제와 임베디드 시스템에서 목표물 탐지 및 추적 과정 수행을 담당하는 콘솔(Console)기반의 영상처리 어플리케이션, 영상처리 알고리즘 작성을 위해 사용된 OpenCV 라이브러리 및 기타 개발에 필요한 어플리케이션 등으로 구성하였다.

영상처리 보드에 포팅된 임베디드 리눅스는 임베디드 시스템의 메모리나 동작 속도 등을 고려하여 임베디드 기기 및 시스템에서도 충분히 작동하도록 파일 시스템 및 용량을 최소로 하여 개발된 운영체제이다. 따라서 사용할 수 있는 메모리 자원 및 용량과 기능 등이 제한되어 있으며, 본 연구에서는 시스템 최적화 및 용량을 최소화하기 위해 다른 기능들은 배제하고 단순히 영상처리 알고리즘을 수행하는 동작만을 수행하는 임베디드 시스템을 구현하고자 하였다. 영상처리 알고리즘으로 구성된 어플리케이션은 OpenCV 라이브러리를 이용하여 C언어로 작성하였으며[5][6], 리눅스 환경에서 비디오 디바이스를 제어하고 사용하기 위한 API인 V4L(Video For Linux)를 병용하였다[7].

2-3 온보드 영상추적 시스템 개발절차

온보드 영상처리 시스템의 개발환경 구성은 크게 소프트웨어 부분과 하드웨어 부분으로 나누어 진행하였으며, 임베디드 시스템 개발환경 구성은 그림 3과 같이 실제 적용 시스템인 타겟 시스템과 실제 개발을 위한 호스트 시스템으로 구성된다.

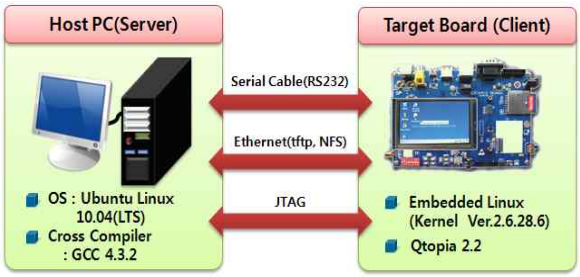


그림 3. 교차 개발환경 구성
Fig. 3. The structure of cross development environment

이는 교차개발환경으로서 실제 시스템 소프트웨어 개발을 담당하는 Host PC와 소프트웨어가 실제 동작하는 타겟보드로 구성이 되며, Host PC와 타겟 시스템 사이에서의 데이터 교환은 시리얼 통신, tftp, NFS(Networking File System)등을 통해서 이루어진다.

타겟보드에서 작동할 소프트웨어 개발을 위해 Host PC의 운영체제로 Ubuntu Linux 10.04(LTS)를 사용하였으며, 타겟보드에는 임베디드 리눅스 커널 2.6.28.6 버전을 포팅하였다. 또한, Host PC에서 작성한 코드를 컴파일해서 실행파일 형태로 만든 후, 타겟 보드에서 실행을 하기 위해 GCC 크로스 컴파일러를 설치하여 임베디드 시스템 환경에 맞는 형태로 크로스 컴파일 하는 과정을 수행하였다.

Host PC와 타겟보드와의 데이터 교환은 Host 시스템 상의 데이터를 타겟보드에서 고속으로 접근할 수 있다는 장점이 있는 NFS서비스를 이용하였다. 부트로더(Bootloader)는 운영체제가 시동이 되기 이전에 커널이 올바르게 동작되기 위해 필요한 모든 세팅을 담당하는 부분으로서, 일반적으로 FLASH ROM의 시작번지(0x00000000 ~ 0x0003FFFF, 256KB)에 저장이 되어 있으므로 타겟보드의 전원을 인가함과 동시에 실행된다.

오픈 소스라는 리눅스 커널의 장점으로 사용하지 않는 설정들을 제거하여 용량을 줄이거나 새로운 디

바이스를 지원해야 하는 경우 커널을 수정하여 설정 파일을 다시 세팅 할 수 있다. 이 때, GUI를 지원하는 'make xconfig' 컴파일 옵션을 이용하여 개발에 필요한 부분들만 커널에 포함하였다. 본 연구에서는 영상처리용 어플리케이션 개발을 위해 리눅스 운영체제에서 비디오 디바이스를 제어하고 사용하기 위한 API인 V4L(Video For Linux)를 사용하였으며, 커널 빌드 후 zImage라는 커널 이미지가 생성이 되고, 이 미지 파일을 타겟보드로 다운로드 하여 커널을 동작시키게 된다.

시스템 적용 환경 구축에서는 실제 시스템에 적용하기 위해 필요한 라이브러리 및 응용 프로그램들에 대한 환경을 구축하였다. 영상처리 알고리즘 작성을 위해 C언어 기반의 컴퓨터 비전 관련 라이브러리인 OpenCV 라이브러리를 타겟보드에 포팅하였다.

본 연구에서는 영상처리 결과를 LCD를 통해서 확인해보기 위하여 리눅스 시스템 그래픽 표현 장치인 프레임 버퍼(Frame buffer)를 이용하였는데, 이를 응용 프로그램에서 제어할 수 있도록 만들어진 프로그램은 프레임 버퍼 드라이버(Frame buffer driver)가 된다. 프레임 버퍼 디바이스는 '/dev' 디렉토리 내에 /dev/fbX 형태로 존재하며, 프레임 버퍼 장치 드라이버를 이용하여 타겟보드에 부착된 LCD에 영상을 표시한다. LCD를 제어하기 위해서는 /dev/fb0 프레임버퍼 디바이스를 로드하여 이 장치가 맵핑된 메모리 영역을 수정하여 프레임버퍼의 내용을 변경함으로써 제어가 가능하다[8].

본 연구에서 사용된 개발보드는 RGB색상을 표현할 때 Red 5비트, Green 6비트, Blue 5비트 총 16비트(16bpp)만을 표현하도록 되어 있는데 일반적으로 RGB로 표현되는 색상은 총 24비트의 정보를 갖게 되므로 이를 16비트로 표현하기 위해서는 Red와 Blue에서 하위 3비트, Green에서 하위 2비트를 버린 값으로 수정하였다. 또한 프레임 버퍼의 내용을 변경하기 위해 mmap()시스템 콜을 사용하여 프레임 버퍼에 대한 접근을 메모리에 대한 접근으로 바꾸어 사용하도록 하여 고속의 입·출력이 가능하도록 하였다.

무인항공기 온보드 영상처리 시스템에서 영상센서를 통해 특정색상을 기반으로 목표물을 탐지하고 추적하는 영상처리 알고리즘을 적용하기 위해 영상처리 어플리케이션을 작성하였다. CPU의 부하를 줄이고자 그림 4와 같이 텍스트 기반의 콘솔(Console) 형식으로 어플리케이션을 작성하여 영상처리와 동시에 원본 영상을 저장할 수 있도록 하였다.

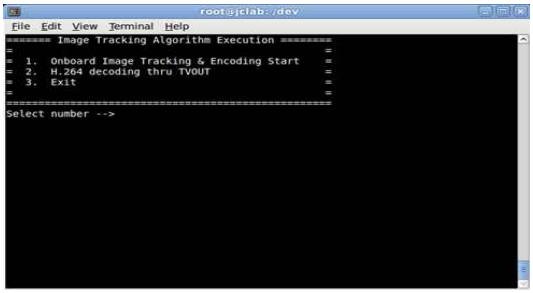


그림 4. 영상처리 어플리케이션 실행화면
Fig. 4. Running of the image processing application

영상처리 어플리케이션 실행에 앞서 아날로그 영상 출력이 잘 수행되는지 확인하기 위해 그림 5와 같이 CCD 카메라를 통해 들어오는 아날로그 영상에 대한 출력을 확인하였다. 또한 개발보드가 무인항공기에 탑재되어야 하므로 부팅이 끝나면 자동으로 어플리케이션이 실행되어야 할 필요가 있다. 따라서 영상처리 어플리케이션을 임베디드 리눅스 시스템 초기화 파일에 등록하여, OS의 부팅이 끝나면 자동으로 일련의 루틴들을 실행하도록 하였다.

그림 5. 영상추적 어플리케이션 실내 테스트
Fig. 5. Inner test of the image-based tracking application

Ⅲ. 영상처리 시스템

3-1 영상기반 목표물 탐지 및 추적 알고리즘

영상기반 목표물 탐지 및 추적 알고리즘은 임베디드 시스템이 가진 한계와 특수성 등을 고려하여 연산량이 적고 단순한 알고리즘인 색상 기반 목표물 탐지 및 추적 알고리즘을 적용하였다.

본 연구에서 적용한 색상기반 탐지 및 추적 알고리즘은 특정한 색상을 가진 지상의 물체를 탐지하고 추적하는 알고리즘으로서, RGB 색 공간 포맷에서 영상처리를 수행하게 된다. 그림 6은 영상처리 시스템의 프로세스를 나타낸다.

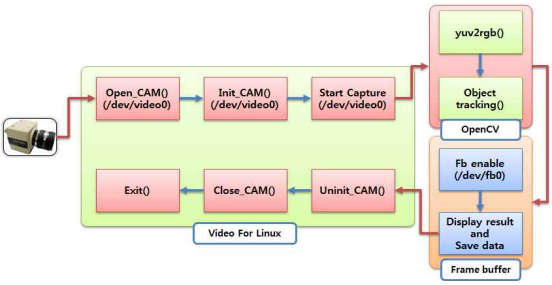


그림 6. 영상처리 시스템 프로세스
Fig. 6. The process of the image processing system

카메라 장치 초기화 이후에 카메라로 입력되는 영상신호는 기본적으로 YUV(YCbCr) 포맷이며, 영상처리를 위해 디지털 신호로 변환하는 과정에서 YUV 포맷의 색 공간을 RGB 포맷으로 변환시켜주는 작업이 필요하다. RGB 포맷으로 변환 후, 특정 색상을 탐지하고 추적하는 색상기반 영상처리 알고리즘을 수행하게 되고, DAC(Digital Analogue Converter)를 거쳐 아날로그 신호로서 외부로 출력하게 된다. 원본 영상은 H.264 코덱으로 인코딩하여 SD메모리에 저장하게 된다. 영상 입·출력 디바이스에 대한 초기화 및 해제는 V4L에서 담당하며, 영상포맷 변환 및 목표물 탐지 및 추적 알고리즘은 OpenCV 라이브러리를 이용하여 작성하였다.

무인항공기에 장착된 카메라의 영상좌표는 그림 7과 같으며, 카메라 좌표계는 핀홀 카메라 모델을 사용한다. 목표물의 영상 좌표와 3차원 좌표계 사이의 관계식은 식 (1)과 같다.

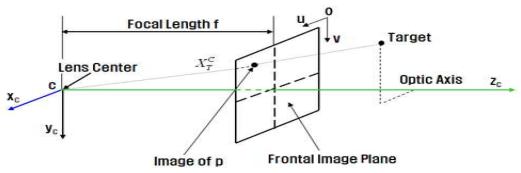


그림 7. 영상좌표계
Fig. 7. Image geometric

$$x_{img} = f_x \frac{x_T^C}{z_T^C}, \quad y_{img} = f_y \frac{y_T^C}{z_T^C} \tag{1}$$

여기서, x_{img} , y_{img} 는 목표물의 영상 위치를 나타내며, fx , f_y 는 각각 영상 좌표계의 x축 및 y축에 대한 초점 거리를 나타낸다. 본 연구에 적용한 색상 기반 목표물 탐지 및 추적 알고리즘의 프로세스은 그림 8과 같다.

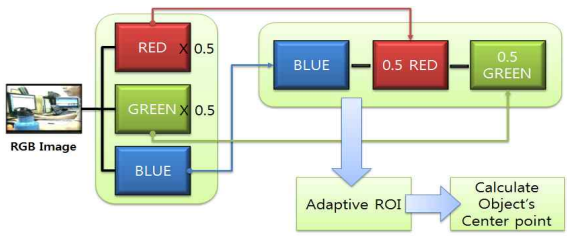


그림 8. 파란색 물체 탐지 및 추적 알고리즘 프로세스
Fig. 8. The process of detection and tracking algorithm for blue-color

YUV 포맷을 RGB 형태로 변환한 후, RGB 형태의 색 공간에서 특정 색상의 값을 추출하기 위해서는 우선 RGB 포맷의 3채널 영상을 Red, Green, Blue 각각의 1채널 영상으로 분리한 다음, 전체 영상 중 특정 색상, 밝기 정보를 가진 부분을 걸러내기 위해, 그레이 영상으로 변환한다. 이는 RGB 컬러 영상으로 처리할 경우에는 한 프레임 당 데이터 크기가 크고 화소값 기반의 영상 처리를 수행하는 경우 많은 시간이 소요될 수 있기 때문이다. 특히 무인항공기의 경우 단일 프레임의 영상 촬영이 아니라 연속된 영상 프레임을 다루기 때문에 연산량을 줄이기 위해서는 꼭 필요한 작업이다.

예를 들어 파란색의 물체를 탐지하기 위해서는 파란색에 해당하는 픽셀들만 찾아내는 연산을 통해서

파란색 색상을 가진 물체를 찾게 되는데 이 때, 연산량을 줄이고 고속으로 탐지하기 위해서 출력영상을 320x240(pixels) 사이즈로 설정하고 Adaptive ROI(Region of Interest)를 적용하였다.

본 연구에서 사용한 개발보드의 S3C6410 프로세서는 영상 스케일 변환이나 영상 포맷변환을 수행하는 PP(Post Processing) 인터페이스에서 바로 RGB 포맷으로의 변환과정을 수행하게 되므로, 영상처리 알고리즘 내에서 포맷 간 변환과정을 수행할 필요가 없게 된다. 그렇기 때문에 RGB 포맷을 이용하여 컬러 공간 변환에 수행되는 시간을 절약할 수 있다.

3-2 Adaptive ROI 설정

임베디드 시스템에서의 알고리즘 처리 속도 향상을 위하여 카메라 입력영상에 대하여 관심영역(Region of Interest)을 설정하여 연산하도록 하였으며, 추적 물체의 크기에 따라 관심영역의 크기가 적응적으로 변하는 Adaptive ROI방식을 적용하였다.

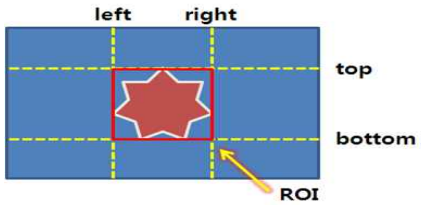


그림 9. Adaptive ROI
Fig. 9. Adaptive ROI

Adaptive ROI는 그림 9와 같이 목표물을 찾기 위해서 매 프레임 왼쪽, 오른쪽, 위, 아래 네 방향으로 지속적으로 스캔을 하면서 목표물을 찾아가게 되며, 목표물을 찾게 되면 목표물의 중심점을 찾아 계산하고, 중심점 위에 원으로 마킹이 되도록 하였다.

ROI가 목표물보다 작아지게 되는 경우에 오탐지 가능성이 발생하기 때문에 목표물을 에워싼 ROI 박스로부터 10픽셀 정도의 마진을 두었으며, Threshold 값 이상의 값을 가지는 경우에만 파란색으로 인식을 하도록 하였다. 따라서 Threshold값 이하의 값을 가지는 경우에는 ROI 과정 및 중심점 연산 과정을 수행하지 않으며 설정된 ROI 내에서 다시 탐색을 시작하도록 한다. 이 때의 파란색 물체를 찾기 위한 픽셀의 Threshold값은 조도 변화와 같은 외부환경에 따라 가변적으로 수정할 수 있도록 하였다.

Adaptive ROI를 적용하였을 색상기반 목표물 탐지 알고리즘에 대한 평균 처리시간은 그림 10에서 나타난 것처럼 139.61ms로서 Adaptive ROI를 적용하지 않았을 때의 평균 처리시간인 307.09ms보다 처리시간이 2배 이상 향상된 것을 확인하였다.

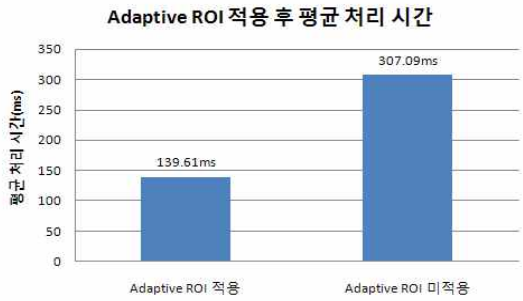


그림 10. Adaptive ROI 적용 후 평균 처리 시간 비교
Fig. 10. Comparison of mean processing time after applying Adaptive ROI

Ⅳ. 실험 및 결과

4-1 지상 실험을 통한 시스템 성능 확인

앞서 기술한 내용들을 바탕으로 하여 무인항공기 온보드 영상처리 시스템을 개발하고 그림 11과 같이 무인항공기에 탑재하였다. 배터리 전원을 공급할 수 있도록 배터리 연결선과 카메라 영상 입·출력을 위한 RCA 케이블을 별도로 제작하여 입력부는 김벌 카메라와 연결하고, 출력부는 아날로그 영상을 송신하는 영상 모뎀에 연결하여 획득한 영상을 지상으로 송신할 수 있도록 구성하였다.

그림 11. 무인항공기에 탑재한 모습
Fig. 11. PNUAV with on-board

실제 비행실험에 앞서 지상 테스트를 통해 시스템의 안정성 및 성능을 확인하였다. 지상테스트 결과 목표물 탐지 및 추적 영상처리 속도는 평균 146.3ms를 나타내었다. 시스템 오작동이나 강제 종료 상황 발생 시 데이터에 대한 손실여부를 확인하기 위해 시스템을 강제 종료 시킨 상황에서도 시스템이 종료되기 전까지에 대한 영상 데이터 및 중심점 데이터가 저장이 된 것을 확인하였다. 영상처리 시스템을 무인항공기에 탑재한 후, 엔진 파워를 점차적으로 증가하여 안정성을 테스트 하였으며 엔진 파워를 최대로 한 상태에서도 고주파 진동으로 인한 비정상 종료 상황이 발생하지 않았으며, 전원 공급도 일정하게 유지되는 것을 확인하였다. 저장된 영상 데이터 확인 결과 약간의 노이즈가 발생하였으나 연속적인 시간지연은 발생하지 않았다.

영상처리 부분에 대한 지상테스트는 그림 21과 같이 320x240 픽셀 사이즈의 영상에 대해 수행하였으며, 움직이는 파란색 물체를 탐지 및 추적하도록 하였다. 움직이는 물체를 가정하여 파란색 물체를 임의의 방향으로 움직이도록 하였으며 탐지된 물체에 대해 중심점을 잘 찾는 것을 확인하였다. 그림 12의 오른쪽 영상은 왼쪽 영상에 대한 원본영상이며, 노이즈가 섞인 처리 결과영상과 달리 깨끗한 화질의 원본 영상을 획득한 것을 확인할 수 있다.

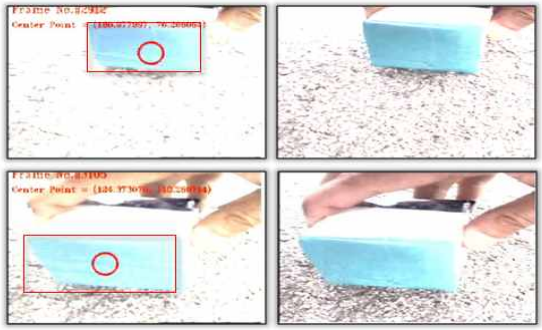


그림 12. 이동물체에 대한 탐지 결과
Fig. 12. Result of detection on moving object

4-2 비행 실험을 통한 시스템 성능 및 결과

온보드 영상처리 개발보드를 무인항공기에 탑재하고 총 세 번의 비행실험을 통해 온보드 영상처리 시스템의 성능을 평가하였다. 수동비행을 통해, 탐지 및 추적하고자 하는 파란색 목표물 상공을 지속적으로 선회하였으며, 이 중 100 프레임을 추출하여 입력 영상으로부터 목표물을 탐지 및 추적하고 중심점을 계산한 후, 목표물 탐지 및 추적 결과 비교를 위해 SD메모리에 저장된 320x240 픽셀 크기의 원본 영상으로부터 고성능 CPU의 데스크탑 PC(Intel Core i5 CPU 750, 2.67GHz, 4GB RAM)에서 수행한 결과와 비교하였다. 그림 13은 온보드 영상처리 시스템으로부터 목표물 탐지 및 추적을 수행한 결과이다. 그림 14는 온보드 영상처리 시스템과 고성능 PC에서 처리된 목표물에 대한 탐지 및 중심점 이동궤적에 대한 비교인데 가로/세로축은 각각 영상의 가로/세로를 의미한다. 그림 15는 중심점 픽셀좌표에 대한 거리 오차를 나타낸 것이다. 측정 결과, 세 번의 실험에서 중심점 픽셀좌표에 대한 픽셀 거리 오차는 각각 2.95픽셀, 3.85픽셀, 3.95픽셀로서 4픽셀 이내의 오차 범위 내에서 목표물을 탐지하고 추적하는 것을 확인하였으며, 결과를 표 2에 정리하였다.

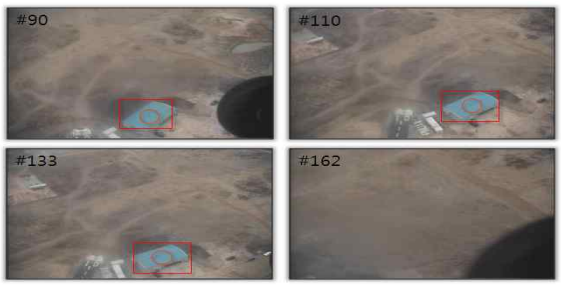


그림 13. 온보드 영상처리 수행 결과
Fig. 13. The result of performance on onboard image processing

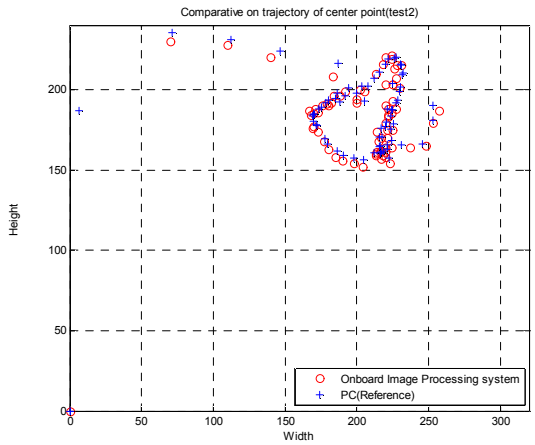


그림 14. 중심점 위치 비교
Fig. 14. Comparison of center point

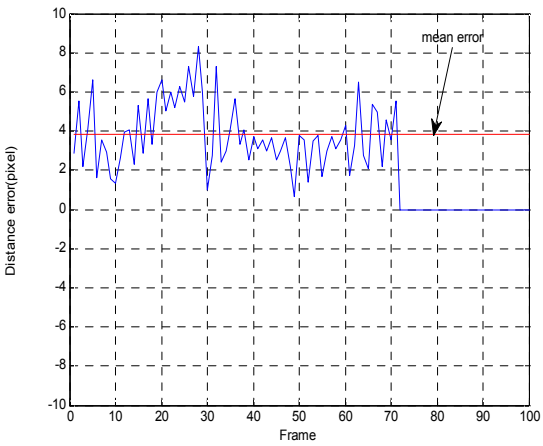


그림 15. 중심점 거리 오차
Fig. 15. Center point error

표 2. 비행실험 결과 및 데스크탑 PC와의 비교
Table 2. The result of flight test and comparison between flight test and desktop PC

	Test 1	Test 2	Test 3
중심점 거리 평균 오차(pixel)	2.94	3.85	3.95
온보드 영상처리 시스템에서의 평균 영상 처리 시간(ms)	138.91	139.45	138.02
고성능 PC 에서의 평균 처리 시간(ms)	5.72	5.13	5.59

V. 결 론

영상센서를 기반으로 하는 고정익 무인항공기 영상임무의 고급기술화를 위한 기초연구로서, 저전력 프로세서인 ARM 프로세서가 탑재된 개발보드를 이용하여 소형화, 경량화를 목표로 하는 고정익 무인항공기용 온보드 영상처리 시스템을 구현하고, 목표물 탐지 및 추적 알고리즘을 적용하였다.

온보드 영상처리 시스템의 요구 목표 성능을 고려하여 시스템 최적화 방안을 모색하였으며 지상테스트를 통하여 안정성을 확인하였다. 비행실험을 통해 성능을 검증하였으며, 비행실험 결과 320x240(pixels) 해상도의 영상에서 평균 138ms의 처리속도로 파란색의 목표물을 탐지하고 추적 하는 것을 확인하였다.

데스크탑 PC에서 동일 영상 및 Threshold 값으로 연산 수행하였을 때와 비교하여 탐지된 목표물에 대한 중심점 계산 결과와의 픽셀 거리 오차는 4픽셀 이내로서 목표물 탐지 및 추적이 온보드 시스템 상에서 이루어지는 것을 확인하였다.

본 연구를 통해 온보드 영상처리 시스템의 효용성을 평가하였으며, 차 후 고속 연산이 가능한 프로세서를 이용하여 다양한 영상임무 수행 및 외부변수에 대한 강인성 향상 방안 등 성능개량에 대한 연구가 추가적으로 이루어져야 할 것이다.

감사의 글

이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음

Reference

[1] Jincheol Ha, Eric N. Johnson, and Allen Tannenbaum, “Real-time Visual Tracking Using Geometric Active Contours for the Navigation and Control of UAVs,” *Proceedings of the 2007 American Control Conference, New York, U.S.A,* 7, 2007

[2] J.H.Choi, S.H.Yim, H.C.Bang, “Research for real time of On-board Image Processor,” *2010 Conference of KSAS,* 2010.11. pp.502~505

[3] Y.I.Kim, S.Y.Nho, “Detection Method of Image Objection for Embedded System,” *Journal of Institute of Control, Robotics, and System,* Vol.15, No.4, 2009.4, pp.422~423

[4] JK Electonics, ”Real 6410 Linux Development Manual”

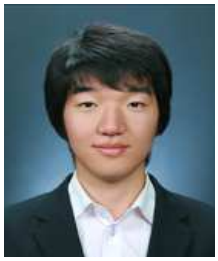
[5] S.J.Lee, D.H.Won, S.C.Yoon, M.Y.Rhu, J.S.Cho, S.K.Sung, Y.J.Lee, “Embodiment of Real Time Image Process System Using Embedded Processor,” *2011 Conference of KSAS,* 2011.11, pp.1026~1030

[6] Zheng Guo-rong, Xiong Chang-zhen, Zhang Yan, “A Method of Embedded Video Surveillance based on OpenCV,” *E-Business and E-Government (ICEE), 2011 International Conference on,* pp.1~4

[7] Paul. B. Matteucci, Member, IEEE, Philip Byrnes-Preston, Spencer C. Chen, Member, IEEE, Nigel H. Lovell, Fellow, IEEE and Gregg J. Suaning, Member, IEEE, “ARM-Based Visual Processing System for Prosthetic Vision,” *33rd Annual International Conference of the IEEE EMBS Boston, U.S.A*, 2011

[8] J.H.An, “Developmental Algorithm of Color Image Using PAX255 ARM Processor,” *Dept. of Computer, Dankook University, Master Thesis*

김 정 호 (金政昊)



2008년 8월 : 부산대학교 항공우주공학과(공학사)
2010년 8월 : 부산대학교 항공우주공학과(공학석사)
2010년 9월~현재 : 부산대학교 항공우주공학과 박사과정
관심분야 : 영상기반 목표물 추적

정 재 원 (鄭載元)

2010년 2월 : 창원대학교 전자공학과(공학사)
2010년 8월 : 부산대학교 항공우주공학과(공학석사)
관심분야 : 임베디드 영상처리

한 동 인 (韓東仁)

2010년 8월 : 부산대학교 항공우주공학과(공학사)
2012년 2월 : 부산대학교 항공우주학과(공학석사)
2012년 3월~현재 : 부산대학교 항공우주공학과 박사과정
관심분야 : 무인 항공기 자율 비행 시스템 및 유도 제어

허 진 우 (許鎭宇)

2011년 8월 : 부산대학교 항공우주공학과(공학사)
2011년 9월~현재 : 부산대학교 항공우주공학과 석사과정
관심분야 : 영상기반 목표물 추적, 영상처리 알고리즘

조 겸 래 (趙謙來)

1977년 2월 : 부산대학교 기계공학과 (공학사)
1979년 2월 : 부산대학교 기계공학과 (공학석사)
1980년 조지아 공과대학 기계공학과 (공학석사)
1986년 텍사스 주립대학(오스틴) 항공우주공학과(공학박사)
1986년 4월~현재 : 부산대학교 항공우주공학과 교수
관심분야 : 인공위성 시스템, 궤도해석, 임무해석

이 대 우 (李大雨)

1993년 2월 : 부산대학교 항공우주공학과(공학사)
1997년 2월 : 부산대학교 항공우주공학과(공학석사)
2001년 8월 : 부산대학교 항공우주공학과(공학박사)
2003년 3월~현재 : 부산대학교 항공우주공학과 교수
관심분야 : 인공위성 시스템, 대기권재진입 유도제어, 무인기 시스템 및 영상추적