



## 저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

工學碩士 學位論文

# OpenCV를 이용한 물체추적에 관한 연구

A Study on Tracking System of Moving Objects  
Using an OpenCV Library

指導教授 朴 永 算

2016年 8月

木浦海洋大學校 大學院

機關工學科 電氣制御

沈 崇 燮

이 論文을 沈崇燮의 工學碩士 學位論文으로 認准함

審査委員 長：工學博士 安 秉 元 ⑩

審査委員：工學博士 裴 哲 吾 ⑩

審査委員：工學博士 朴 永 算 ⑩

2016年 8月

木浦海洋大學校 大學院

機關工學科 電氣制御

沈 崇 燮

# 目 次

List of Figures .....	iii
List of Tables .....	v
Abstract .....	vi
제 1 장 서 론 .....	1
1.1 연구배경 및 연구목적 .....	1
1.2 논문의 개요 .....	3
제 2 장 OpenCV .....	4
2.1 OpenCV(Open Source Computer Vision) .....	4
2.1.1 OpenCV 픽셀값 접근하기(IplImage) .....	7
2.1.2 OpenCV 트랙바 처리(cvCreateTrackbar) .....	7
2.1.3 히스토그램(Histogram) .....	7
2.1.4 히스토그램 평활화(Histogram Equalization) .....	8
2.2 CAMshift(Continuously Adaptive Mean-Shift) .....	9
2.3 Fuzzy Logic .....	11
제 3 장 하드웨어 구성 및 제어기 .....	15
3.1 하드웨어 구성 .....	15
3.1.1 서보모터(Servo Motor) .....	16

3.1.2 PIC 18F452 .....	19
제 4 장 실험 및 분석 .....	23
4.1 물체의 추적 .....	23
4.1.1 퍼지논리의 적용 .....	23
4.1.2 x의 입력에 의한 소속함수 .....	25
4.1.3 Area의 값에 따른 크기의 추적 .....	27
4.1.4 x와 Area의 입력에 의한 PWM 출력 .....	28
4.2 데이터 취득 결과 .....	29
4.2.1 x의 값이 대문자 L일 때 파라미터와 히스토그램 .....	29
4.2.2 x의 값이 소문자 l일 때 파라미터와 히스토그램 .....	29
4.2.3 x의 값이 중앙 C일 때 파라미터와 히스토그램 .....	30
4.2.4 x의 값이 소문자 r일 때 파라미터와 히스토그램 .....	30
4.2.5 x의 값이 대문자 R일 때 파라미터와 히스토그램 .....	31
4.2.6 Area의 값에 따른 탐색창의 지정영역 크기변화(Big) .....	31
4.2.7 Area의 값에 따른 탐색창의 지정영역 크기변화(Small) .....	32
4.3 데이터 분석 .....	33
4.4 Overall Tracking system .....	36
제 5 장 결 론 .....	37
참고문헌 .....	39

## List of Figures

Fig. 2.1 Flowchart of CAMshift Algorism .....	10
Fig. 2.2 Flowchart of Fuzzy logic controller .....	13
Fig. 3.1 하드웨어 구성 .....	15
Fig. 3.2 Operating Interface .....	16
Fig. 3.3 서보모터(Servo motor) & HD pro webcam C920 .....	17
Fig. 3.4 PIC 18F452 Pin Diagram .....	20
Fig. 3.5 Microprocessor PIC 18F452 and LCD16x2 .....	22
Fig. 3.6 RS232 interface with MAX232 .....	22
Fig. 4.1 Fuzzy logic controller .....	24
Fig. 4.2 x의 값에 따른 중심의 추적 .....	26
Fig. 4.3 Area의 값에 따른 크기의 추적 .....	27
Fig. 4.4 PWM 출력값 .....	28
Fig. 4.5 x값이 L(large Left) .....	29
Fig. 4.6 x값이 l(left) .....	30
Fig. 4.7 x값이 C(center) .....	31
Fig. 4.8 x값이 r(right) .....	32
Fig. 4.9 x값이 R(large Right) .....	33
Fig. 4.10 Area 값이 Big .....	34
Fig. 4.11 Area 값이 Small .....	35
Fig. 4.12 Flowchart of PIC 18F452 & PWM .....	36

Fig. 4.13 Flowchart of C++ Programing & PWM .....	37
Fig. 4.14 편차 각과 물체의 상 .....	38
Fig. 4.15 Flowchart of Overall tracking System .....	39

## List of Tables

Table 2.1 OpenCV 주요 라이브러리 기능 .....	5
Table 2.2 크리스프와 퍼지 분류 .....	13
Table 3.1 서보모터 HS-485HB 사양 .....	17
Table 3.2 HD pro webcam C920 사양 .....	18
Table 3.3 PIC Microcontroller 특징 .....	20
Table 3.4 PIC 18F452 Features .....	21
Table 4.1 Fuzzy logic rules .....	25
Table 4.2 $x$ 의 값에 의한 소속함수 .....	26
Table 4.3 Area의 값에 의한 소속함수 .....	27
Table 4.4 PWM 출력값 계산 .....	28



# A Study on Tracking System of Moving Objects Using an OpenCV Library

Soong-sup Shim

*Department of Marine Engineering, Graduate School,  
Mokpo National Maritime University*

## Abstract

As development of computer technology, the computer vision system has been used in our daily life such as monitoring buildings for security reasons and an anti-theft device etc. This monitoring system requires a lot of human resources and cutting edge equipment.

Also, it causes false images and dead zone(unsightly area) during recoding and saving visionary images. The OpenCV has been used for improving these drawbacks. It can track and identify without material's movement or standstill.

The OpenCV, a visualization program, has been programmed with the C++ language and the Visual studio. It is able to tract moving images through the CAMshift algorithms and the fuzzy logic. The OpenCV

advanced visualization program for object tracking has been programmed with the C++ language and the Visual studio. It is able to tract moving images through the CAMshift algorithms and the fuzzy logic. The tracking ability of the Fuzzy logic has been improved by adjusting PWM values of the servo motor in accordance with the movement of the object based on a specified area on the screen and the center point.

This system is able to apply in development of a specified security technology and surveillance system on vessels because it can track and identify monitored objects more efficiently.

# 제 1 장 서 론

## 1.1 연구배경 및 연구목적

컴퓨터 기술의 발전과 이를 지원하는 주변장비의 발전으로 컴퓨터 비전이 적용된 시스템을 주변에서 어렵지 않게 접할 수 있다. 고정용 CCTV는 한정된 공간의 정해진 장소만을 촬영하도록 되어있어서 카메라의 각도에 따라서 촬영할 수 없는 사각지대(Dead zone)가 존재하게 되며 촬영된 화면은 크기가 한정된다. 또한 넓은 화면을 위해 어안렌즈(Fisheye lens)와 같은 화각이 넓은 렌즈를 사용하면 넓은 화면을 감시할 수 있지만 화면의 왜곡이 생긴다. 이는 실제 현장과는 다른 화면의 왜곡 혹은 시야상의 사각지대(Dead zone)를 줄이기 위해 카메라의 화각을 더 크게 늘리게 되면 실제 화면과는 다른 왜곡을 가져오게 된다. 위와 같은 점을 보완하고 정확한 사건의 현장을 확인하기 위해 일정규모 이상의 감시시스템을 구축한 건물에서는 별도의 통제실을 만들고 인력을 투입하여 감시시스템을 구성하게 된다. 이러한 시스템은 담당하는 구역의 규모가 늘어날수록 여러 장비를 투입하여야 하며 이를 운용할 인력 또한 추가적으로 투입되게 된다. 이에 여러 고정된 현장의 감시와 녹화를 하는 고정용 CCTV에 기능을 추가하여 저장된 정보를 바탕으로 화면에 나타나는 대상의 분석 및 추적을 하는 정보의 처리기술이 중요시 되고 있다.

CCTV 뿐만 아니라 현재와 미래 산업으로 각광받는 로봇 기술의 발전을 위해서는 로봇의 움직임과 상황에 따른 판단기준이 필요하고, 영상의 처리 및 컴퓨터 비전을 적용하여 획득된 여러 가지 정보를 바탕으로 적절하고 필요한 작업을 명확하게 수행하도록 하는 판단기준을 제시할 필요가 있다. 이러한 기준을 위해서는 이전의 수동적인 작동 방식의 센서에 의존하는 형태를 벗어나

보다 능동적인 진보된 기술의 개발이 필요하다.[1][3]

여러 대의 카메라를 사용하지 않고 적은 수의 카메라와 장비로 구현 가능한 효과적인 기술을 소개한다. OpenCV는 개방형 소프트웨어 라이브러리임에도 최신의 컴퓨터 비전과 기계학습 기능 등을 실현 할 수 있다. OpenCV를 이용한 알고리즘은 물체추적을 통한 얼굴인식, 물체확인, 행동의 인식 등의 확인과 검출을 하는 기술로 모션추적 및 증강현실, 예고모션 추정, 제스처 추정, 인간과 컴퓨터의 대화 등을 가능하게 한다.[1]

본 논문에서는 이미지 프로세싱을 통한 물체의 추적과 판별기법을 연구하였다. 물체추적 알고리즘은 탐색 윈도우의 검출된 이미지를 통해 물체의 위치를 계산하고, 노이즈를 제거하여 안정적인 물체의 식별 및 판별을 하였다. 이러한 과정은 상관 관계기반의 CAMshift, KLT(Kanade-Lucas-Tomasi) 등의 알고리즘을 사용하였다. 또한 물체의 지정한 영역의 색상을 기반으로 한 추적시스템에 HS-485HB 서보모터를 이용하여 카메라 각도를 조절함으로써 물체를 연속적으로 추적하는 시스템을 구현 하였다.[2]

## 1.2 논문의 개요

본 논문은 경계경비, 관리감독 그리고 해양안전구조시스템 등에 활용이 가능한 물체추적에 관한 연구로 설정된 물체를 정지와 운동에 관계없이 연속적으로 추적이 가능하도록 C++ Programing, Visual studio, OpenCV, CAMshift, 서보모터, Fuzzy Logic, PIC 18F452 등을 사용하여 시스템의 기반을 구성하였다. 시스템은 화면에 지정된 영역의 화면상의 움직임에 따라 물체의 추적 및 판별을 지속적으로 가능하게 한다. CAMshift 알고리즘을 적용하여 물체의 중심을 판별하고, 퍼지논리를 적용하여 물체의 연속적인 움직임에도 불구하고 위치의 추적을 지속되게 하였다. 또한 서보모터를 이용하여 카메라의 각도를 변화시켜 물체의 추적 및 탐색이 가능한 범위를 확장하였다.

이러한 물체추적 기술은 꾸준히 각광받는 무인로봇 분야 및 드론산업, 의료용 등의 한정된 특수 분야에 활용되는 로봇의 정밀한 움직임 제어, 외부로부터의 침입자를 통제하는 보안 시스템에 활용될 수 있다.

## 제 2 장 OpenCV

### 2.1 OpenCV(Open Source Computer Vision)

디지털 영상처리(Image processing)는 화면에 보이는 영상을 처리 과정을 거쳐서 보다 나은 영상을 얻는 기술이다. 이미지 편집 과정을 통해 화면에 보이는 영상이 우리가 실제로 눈으로 보는 영상과 최대한 가깝게 구현하는 것이 주요 기술이다. 이러한 기술은 입력영상의 잡음(Noise)을 제거, 특정영역을 분할(Segmentation), 특정부분을 강조(Emphasizing), 영상의 대비(Contrast)를 개선, 다른 포맷의 영상 압축(Compression)저장 등이 기본적인 영상처리의 방법이다.[1]

카메라를 이용해 획득된 영상은 영상처리가 가능한 정보로 변환되어지고 이는 보통 실시간으로 반영된다. 이를 컴퓨터 비전이라고 하며 현재 우리 주위의 생활환경 및 산업현장에서 자동인식 및 처리 과정을 거쳐서 사용되고 있다. 예로 산업용 생산제품의 결함검사(Industrial inspection), 스캐너 혹은 화면을 통한 문자인식(Character recognition), 카메라를 통한 얼굴인식(Face recognition), 특정 단말기를 통한 지문인식(Fingerprint recognition), 사람, 물건, 자동차 등의 움직이는 물체검출(Motion detection), 연속으로 이동하는 물체추적(Object tracking), 2개 이상의 카메라를 이용한 스테레오 촬영으로 영상의 깊이나 3차원 구조 등을 확인하는 스테레오 비전 등이 있다.[1]

영상처리와 컴퓨터 비전을 큰 기준으로 구분한다면 컴퓨터를 이용하여 영상을 처리하는 것을 영상처리라 할 수 있으며, 획득되어진 영상을 지정부분에서 정보를 추출하여 특정한 목적으로 활용하는 것을 컴퓨터 비전이라 할 수 있다. 이는 영상의 처리에 있어서 입력영상의 화질개선, 잡음제거, 영역분할 등

의 전처리를 영상처리라 할 수 있고, 영상분석, 추적, 인식 등의 후처리를 컴퓨터 비전이라 한다. 후처리의 컴퓨터 비전은 카메라를 통해 촬영된 영상을 분석하여 주요한 영상의 변화를 판단하고 필요한 정보를 파악해 실시간으로 반영하는 기술을 의미한다.[1]

이러한 영상처리 및 컴퓨터 비전을 신호처리(Signal processing), 패턴인식(Pattern recognition), 기계학습(Machine learning), 로봇비전(Robot vision), 인공지능(Artificial intelligence), 원격탐사(Remote sensing) 등의 분야에 사용된다.[1]

OpenCV는 공개 소프트웨어로 누구나 자유로운 수정, 변경, 재배포가 가능하고 위와 같은 영상처리 및 컴퓨터 비전에 사용할 수 있다. 이는 C/C++로 개발되었고, 윈도우즈(Windows), 리눅스(Linux), 맥(Mac) 등의 운영체제에서 사용이 가능하다. 응용프로그램의 작성 또한 호환성이 높아 C/C++, C#, Python, Matlab 등의 다른 언어와도 연동이 가능하다.[1]

Table 2.1 OpenCV 주요 라이브러리 기능

라이브러리	기 능
CXCORE	기본 자료구조, 행렬연산, 동적자료구조, 그리기 함수
CV	영상처리 및 컴퓨터 비전 관련 주요 함수 (영상특징 추출, 필터링, 영상분할, 움직임 분석, 물체추적, 패턴인식, 카메라 캘리브레이션, 3D복원 등)
HIGHGUI	윈도우 관련 함수, 영상 읽기 및 저장함수, 비디오 입출력 함수
ML	기계학습 관련 함수
CVAUX	스테레오 대응점 계산, 뷰 모핑, 3D추적, PCA, HMM 함수

인텔사에서 개발된 IPL(Image processing library)로 시작하여 2000년에 최초로 일반인에게 공개되었고, 500개 이상의 함수로 구성되어 있으며, 다양한 영상처리 및 컴퓨터 비전 알고리즘이 구현되어 보다 쉬운 영상 및 비디오관련 처리를 가능하게 하였다. 응용된 기술은 공개소프트웨어이며 알맞은 스펙만 맞추면 자신의 알고리즘도 라이브러리에 등록 시킬 수 있다.[1][6]

Table 2.1은 OpenCV의 주요 라이브러리를 나타내고 있다. 주요구성은 CXCORE, CV, HIGHGUI, ML, CVAUX로 구성되어 있다. CXCORE 라이브러리는 기본 자료구조와 행렬연산 등의 함수를 지원한다. CV 라이브러리는 영상처리, 패턴인식, 카메라 캘리브레이션(Camera calibration) 등의 함수를 지원하는 가장 중심이 되는 라이브러리이다. HIGHGUI 라이브러리는 사용자가 좀 더 쉽게 영상을 불러오고, 저장하고, 윈도우 창에 나타내는 등의 사용자 인터페이스를 지원한다. ML 라이브러리는 기계학습(Machine learning) 알고리즘을 지원한다.[1][6]

OpenCV는 동적연결라이브러리(Dynamic linking library)로 지원된다. 동적라이브러리는 실행파일(EXE)에 함수의 호출 정보를 포함하고, 목적코드는 실행시간에 호출하여 실행한다. 동적라이브러리는 적재시간 동적연결(Load-time dynamic linking)과 실행시간 동적연결(Run-time dynamic linking) 라이브러리로 구분되며, OpenCV는 적재시간 라이브러리로 실행파일(EXE)이 메모리에 적재되는 시간에 DLL이 함께 메모리에 적재된다.[1][6]

적재시간 동적연결 라이브러리를 사용하기 위해서는 ‘컴파일 → 링크 → 실행’의 과정을 거치게 된다. DLL에 정의된 함수의 원형이 선언되어 있는 헤더 파일은 컴파일 할 때 필요하고, 응용프로그램을 링크할 때 임포트(Import) 라이브러리(LIB)가 필요하다. 응용파일을 실행할 때는 DLL이 필요하다. DLL은 현재폴더 혹은 환경변수에 경로(Path)가 설정된 폴더에 위치하여야 한다.[1]



### 2.1.1 OpenCV 픽셀값 접근하기(IplImage)

IplImage는 2차원 영상 데이터 구조체이고 nChannel은 채널의 개수로 1에서 4의 정수로 정의 된다. depth는 화소의 비트표현이다. dataOrder는 채널 자료의 저장 순서를 의미한다. Origin은 영상의 영점을 정의한다. width는 영상의 가로방향 화소수를 의미하고, height는 영상의 세로방향 화소수를 의미한다. roi는 Region Of Interest(ROI)로 영상의 관심영역을 의미한다.[1]

### 2.1.2 OpenCV 트랙바 처리(cvCreateTrackbar)

cvCreateTrackbar 함수는 cvCreateWindow 함수로 생성한 윈도우에 트랙바를 생성한다. 실제 트랙바 콜백 이벤트를 처리할 함수는 따로 만들어줘야 한다.[1]

```
int cvCreateTracbar(const char*trackbar_name, const cha>window_name,
int*value, int count, CvtracbarCallback on_change);
```

\*trackbar\_name은 트랙바 이름, window\_name은 트랙바를 붙일 윈도우 이름, value는 트랙바가 생성될 때 슬라이더의 위치, count는 트랙바 슬라이더의 최대위치 값, 최소위치 값은 0이다. on\_change는 트랙바의 슬라이더가 변경될 때마다 자동 호출되는 핸들러 함수이다.[1]

### 2.1.3 히스토그램(Histogram)

통계에서 히스토그램(Histogram)은 관찰데이터의 빈도수(Frequency)를 막대 그래프로 표시한 것으로, 데이터의 확률밀도함수(Probability density function)를 추정가능하다. 영상처리에서 히스토그램은 영상의 밝기 값에 대한 분포를 나타낸다 할 수 있으며 화소의 분포에 대한 분석을 위한 중요한 정보이다. 입

력되는 영상의 히스토그램을 분석하여 영상의 밝기 구성, 명암의 대비 등에 대한 정보를 알 수 있다. 이러한 정보를 분석하여 영상개선 및 화질 향상에 사용할 수 있다.[1]

#### 2.1.4 히스토그램 평활화(Histogram Equalization)

히스토그램 평활화(Histogram Equalization)는 입력영상의 화소값을 누적분포함수(Cumulative distribution function)를 사용하여 출력영상의 화소값을 계산하는 영상개선(Image enhancement) 방법이다. 히스토그램 평활화를 수행하면 화소값의 범위가 좁은 저대비(Low Constant) 입력영상을 화소값의 넓은 고대비(High constant)의 출력 영상으로 얻어 영상의 시각적 질을 개선한다.[1]

### 2.2 CAMshift(Continuously Adaptive Mean-Shift)

영상처리 알고리즘으로는 Gary.R.Bradski가 제안한 CAMshift(Continuously Adaptive Mean-Shift) 알고리즘이 있다. 이 알고리즘은 색분할(Color Segment) 추적방법의 Mean-Shift 알고리즘을 연속추적효과(Streaming) 환경에서 사용하기 위하여 개선한 것이다. 이는 탐색 윈도우의 크기를 스스로 조정하는 기법을 사용하여 Mean-Shift의 단점을 보강하였다. 객체를 고속으로 추적하는데 용이하나 조도변화, 잡음이 많은 배경에서는 성능이 좋지 않다. 검출된 객체영역의 Hue 값의 분포를 이용하여 변화될 위치를 예측하고 다음 방향을 탐지한 후 물체의 중심을 찾아서 객체를 추적하게 된다. 간단히 생각한다면 이름과 같이 ‘연속적인 적응성 평균이동 알고리즘’으로 Mean-Shift 기능을 사용하여 탐색 윈도우의 크기를 스스로 조정하는 기능이다.[4]

CAMshift는 가장 대중적인 Mean-Shift Algorithm의 변형이며 임의의 물체

를 추출하기 위해 경험적 분포(Empirical distribution)에서 동작하는 탐색 기능의 알고리즘이다. 가장 큰 특징은 Mean-Shift 단점을 보완하였다는 것이고, 특징점, 코너, 색상 등 밀도분포의 특성을 연속추적효과(Streaming) 환경에 사용가능하게 하였다.[4]

본 실험에서는 0인 영역(중심)을 찾고, 첫 번째 영역을 찾고, 창 위치의 평균을 찾는 방법으로 이루어 졌다.

탐색 창에서 0인 영역을 찾는 것은 탐색창의 표시되는 영역에서 나타나는 개체의 이미지 혹은 픽셀의 수라 할 수 있고, 다음과 같이 식(1)로 나타낼 수 있다.

$$M_{00} = \sum_x \sum_y I(x, y) \quad (1)$$

탐색 창에서 첫 번째 영역을 찾는 것은 모든 픽셀의 좌표의 합이고, 다음과 같이 식(2), (3)으로 나타낼 수 있다.

$$M_{10} = \sum_x \sum_y xI(x, y) \quad (2)$$

$$M_{01} = \sum_x \sum_y yI(x, y) \quad (3)$$

탐색 창에서 창 위치의 평균을 찾는 것은 다음과 같이 식(4), (5)로 나타낼 수 있다.

$$x_c = \frac{M_{10}}{M_{00}} \quad (4)$$

$$y_c = \frac{M_{01}}{M_{00}} \quad (5)$$

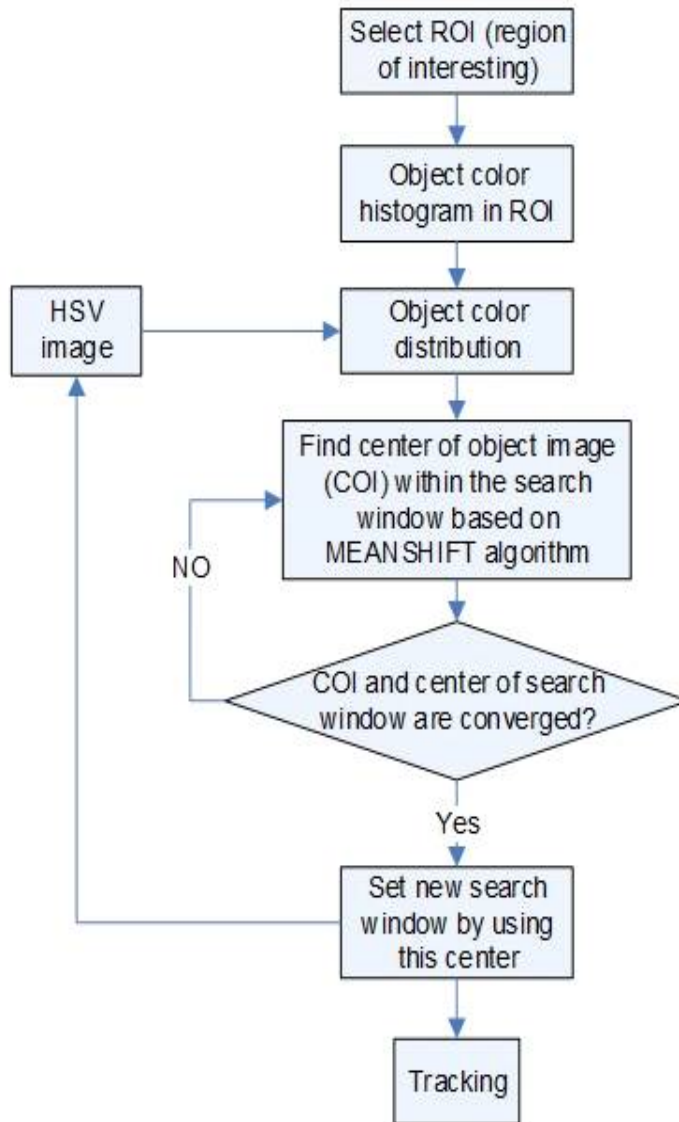


Fig. 2.1 Flowchart of CAMshift Algorithm

## 2.3 Fuzzy Logic

퍼지 시스템의 구성은 크게 세 가지로 언어변수(Linguistic variable), 소속함수(Membership function), 퍼지규칙(Rules)으로 구분된다. 퍼지논리는 퍼지이론의 한 부분인 퍼지집합의 사고방식을 기초로 하고 있다. 퍼지집합의 개념은 판단의 대상이 그렇다/아니다 혹은 속한다/안 속한다 등의 이진법 논리가 아닌 각 대상이 속하는 정도를 소속함수(Membership function)로 나타내고 그 함수를 대응되는 대상과 함께 표기하는 집합이다.[8][9]

퍼지논리는 확률이나 참의 정도로 명확히 표현 할 수는 없지만, 0에서 1 사이에 분포하는 연속적인 값들로 표현된다. 퍼지집합은 1965년 미국 캘리포니아 대학 로프티 자데(Lofti Zadeh) 교수의 논문 “Fuzzy Sets”이 최초로 알려져 있다.[8][9]

퍼지논리는 이진법 논리가 아니라 여러 단계로 나타내는 논리라 할 수 있다. 이는 하나의 뜨겁다는 1로 나타내고 뜨겁지 않다는 0으로 표현한다면 이 사이에는 여러 가지 상황적 표현이 있을 수 있다. 매우 뜨겁다, 뜨겁다, 미지근하다, 차갑다 등의 단계적 표현이 가능하다. 이는 대상을 단순히 판단하는 것이 아니라 어느 쪽에 더 가깝게 속하는지 나타낼 수 있다. 이는 이진법 논리의 한계를 넘어서 사실 정도를 구체적으로 알려줄 수 있다.[8]

이러한 여러 단계를 고려하는 것이 퍼지논리 이다. 뜨겁다는 ‘1’, 미지근하다는 ‘0.5’, 차갑다는 ‘0’과 같은 값으로 표현하고 1과 0사이에는 0.5만 있는 것이 아니며, 정수의 숫자로는 표현이 애매하지만 속하는 정도를 고려할 수 있다.[8][9]

퍼지논리 시스템은 1980년대에 퍼지제어의 실용화가 시작되었고 정수장, 자동열차 시스템 등의 여러 가지 실험적인 기술들로 활용이 되었다. 1990년대 초부터는 우리 생활에서의 가전제품뿐만 아니라 산업현장의 로봇제어, 생산설

비 제어 등은 물론 문자인식, 음성인식과 같은 패턴인식분야에도 활용되고 있다.[8]

현재 사용되는 퍼지 시스템의 대부분은 고전적 제어이론에 기초한다. 일반적 제어이론은 제어 대상 시스템의 상태와 출력을 관측하고 목표값의 편차를 감소시키는 방법으로 제어 대상의 입력을 조절하는 피드백(Feedback)제어가 기본이다. 퍼지제어에서 기본이 되는 것은 퍼지추론(Fuzzy inference)이 바탕이 된다.[10]

퍼지 컨트롤러는 현재 입력된 변수들로부터 퍼지로직으로 설계된 규칙들의 수학적 계산을 통해 출력 값으로 나타난다. 본 실험에서는 탐색창의 지정영역의 0인 영역(중심)을 찾아 첫 번째 영역을 찾고 창 위치의 평균을 찾는 방법에 퍼지이론을 적용하여 추적대상의 움직임을 나타내도록 하였다.

퍼지 컨트롤러는 퍼지화, 규칙기반, 비퍼지화(역퍼지화)로 나눌 수 있다. 퍼지화는 정확한(크리스프, Crisp) 입력 값을 퍼지규칙에 의해 각각의 언어 값과 소속함수로 바꾸는 과정이다. 규칙기반(Rule base)은 시스템의 제어기에 해당하며 퍼지집합과 입력변수, 그리고 출력변수들과 연관된 규칙의 집합이며 각 경우에 해야 할 일을 결정한다. 비퍼지화(역퍼지화, Defuzzification)는 퍼지제어기에서 퍼지 출력을 보통의 수치로 변환시키는 과정이다. 퍼지 출력 값을 실제로 사용하기 위한 등가의 정확한 값(Crisp value)으로 변환 하는 것이다. 퍼지규칙이 평가되고 해당되는 값들이 연산되면 결과는 다른 출력 퍼지집합에 해당되는 소속도(Membership value)에 관련된 수치 값이 된다.[8][9][10]

크리스프와 퍼지의 일반적인 구분에 따른 예시는 Table 2.2와 같다. 크리스프는 명확한 값으로 표현되어 있기에 소속도함수로 표현할 수 있지만, 퍼지는 추상적이고 많은 형태를 가지고 있다.[8]

어떤 대상이나 집합을 크리스프 개념과 퍼지 개념으로 나타낼 때, 첫 번째는 크리스프 개념으로만 표현, 두 번째는 크리스프와 퍼지 혼합개념으로 표현,

세 번째는 퍼지개념으로만 표현이 가능하다.

Table 2.2 크리스프와 퍼지 분류

분 류	크리스프	퍼 지
객 체	인간, 고양이, 나무 등	
수 량	자동차 100대	자동차가 많다
키	170cm 이상이다	키가 크다
체 중	50kg 이하이다	마르다
나 이	30세에서 40세 사이이다	청년이다
온 도	60℃ 이다	뜨겁다
정 도		예쁘다, 좋다, 맛있다

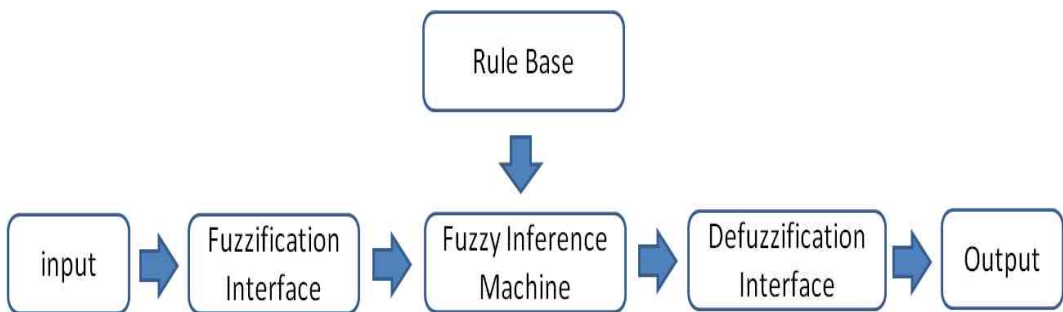


Fig. 2.2 Flowchart of Fuzzy logic controller

Fig. 2.2은 퍼지논리 컨트롤러(Fuzzy logic controller)의 구성을 나타낸다. 퍼지 규칙기반(Fuzzy rule base)은 IF-THEN 형태의 언어적 제어 규칙이며, 퍼지화 인터페이스(Fuzzification interface)는 입력변수의 값의 범위를 일치하는 논의영역(Universe of discourse)으로 변환시키는 스케일 매핑(Scale Mapping) 작업을 담당한다. 퍼지추론 엔진(Fuzzy Inference Machine)은 퍼지화 입력

(Fuzzified Input)과 퍼지규칙(Fuzzy Rule)으로부터 규칙을 정하는 의사 결정 논리(Decision-Making Logic)를 나타낸다. 비퍼지화 인터페이스(Defuzzification Interface)는 변수 값의 범위로 추정되는 논의 영역(Universe of Discourse)으로 바꾸는 스케일 매핑(Scale Mapping) 작업을 진행한다. 최종적으로 추론된 퍼지제어동작(Fuzzy control action)으로부터 비퍼지제어동작(Nonfuzzy control action)을 내는 비퍼지화(Defuzzification)을 진행한다. 이는 최종적으로 퍼지추론 과정을 거친 보통의 수치데이터를 출력한다.[8]



## 제 3 장 하드웨어 구성 및 제어기

### 3.1 하드웨어 구성

본 실험에서 하드웨어는 크게 세부분으로 Fig. 3.1과 같이 컴퓨터, 서보모터와 결합된 카메라, 컨트롤 회로로 구성하였다. 컴퓨터는 카메라를 통해 촬영된 영상을 컨트롤 회로를 거쳐서 모니터 화면상으로 실시간으로 물체의 추적상황을 나타낸다.

Fig. 3.2는 물체 추적시의 과정을 나타낸 것이며, 컨트롤 회로는 영상처리와 컴퓨터 비전을 기계어의 변환과 입·출력 시스템을 위하여 마이크로프로세서 (Microprocessor) PIC 18F452와 MAX232 등으로 구성하였다.

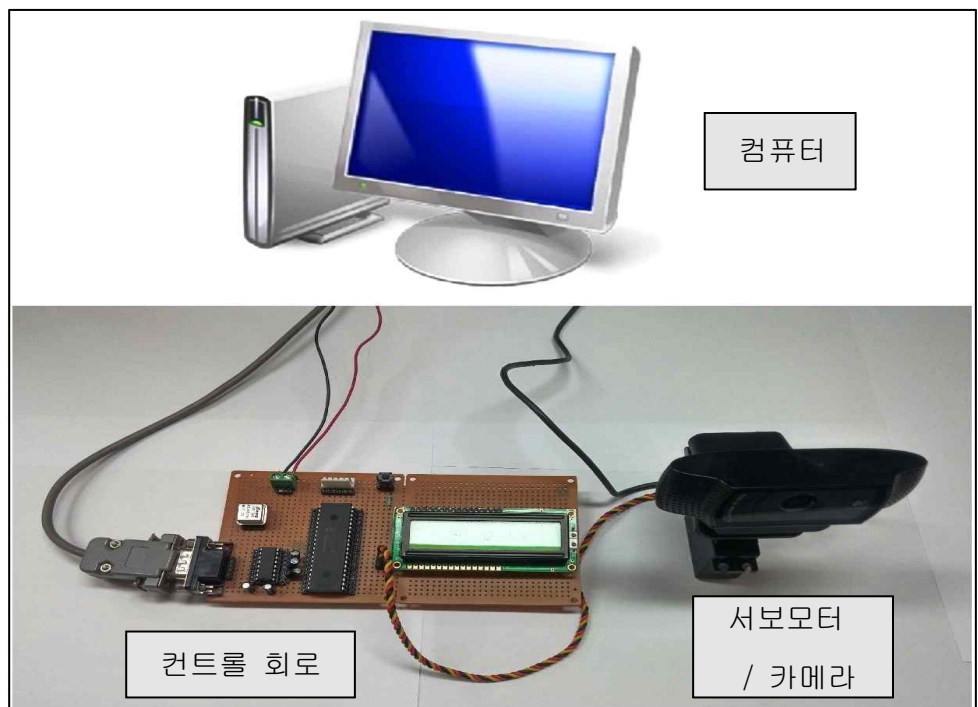


Fig. 3.1 하드웨어 구성

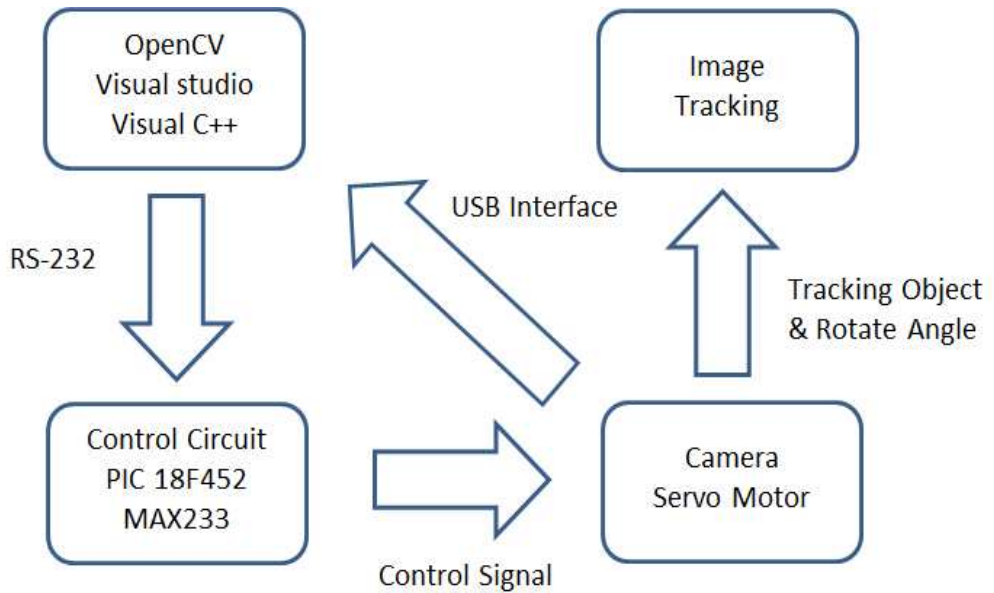


Fig. 3.2 Operating Interface

### 3.1.1 서보 모터(Servo Motor)

서보모터(Servo Motor)는 모터를 이용해 방향 및 위치를 제어하기 위한 회로가 장착되어 있고 움직임의 범위는 정교한 각 위치 제어가 가능하다. Servo의 어원은 라틴어의 Servus, 영어의 Slave라는 뜻이고 추종한다, 따른다는 의미이다. 서보모터는 지시된 명령을 일정한 각도 혹은 움직임의 범위에 따라 작동하는 모터를 의미한다. 일반적인 서보모터는 모터와 제어 회로, 알고리즘 등을 포함하며 모터만을 서보모터라고는 하지 않는다. 서보모터는 일반적으로 사용 전원, 구조, 동작원리, 토크 발생 원리, 용도 등에 따라 분류 되어 사용된다. 반도체 장치, 공작기계, 자동화 설비, CCTV 카메라, 캠코더, DVD, 프린터 등에 사용되는 모터처럼 일정한 각도에 따라 정확한 위치와 속도로 사용 가능하다. 또한 일반적인 모터의 경우도 알맞은 알고리즘과 회로를 구성하여 작동 시스템을 연결하여 위치와 속도를 조절하면 서보시스템으로 볼 수 있다. 이러

한 서보시스템은 모터와 기계부, 제어구동 소프트웨어로 구성되고 제어추종 시스템을 이룬다.[11][15]

Table 3.1 서보모터 HS-485HB 사양

HS-485HB, Features	
Control System	+Pulse Width Control 1500usec Neutral
Operating Voltage	4.8-6.0 Volts
Operating Speed	0.22sec-0.18sec/60° at no load
Operating Angle	90 Deg. one side pulse traveling 900usec



Fig. 3.3 서보모터(Servo motor)  
& HD pro webcam C920

일반적인 DC모터와 달리 펄스의 주기로 정, 역, 정지의 동작을 하는 특수한 모터이다. 소형이며 부착하기 용이한 장점이 있기에 로봇의 관절이나 조향장치에 사용된다. 서보모터의 구조는 DC모터의 회전축에 감속기어를 연결하고, 여러 개의 감속기어를 거친 마지막 회전축에는 내부에 가변저항을 연결하여 이 연결된 축이 움직인 각도에 따라서 움직이는 가변저항의 값을 내장된 전자 회로를 통해 입력된 펄스폭에 비례하는 각도를 출력하게 된다. 커넥터는 3선이며 VCC(전원, 빨간색), GND(접지, 검은색), PWM(신호, 노란색)을 나타내는 것이 일반적이다.[12]

본 실험에서는 서보모터는 카메라의 움직임을 물체의 움직임에 따라 움직이게 도와준다. 실험에 사용된 서보모터는 HITEC사의 HS-485HB를 이용하였다. 서보모터와 함께 결합되는 카메라는 Logitech사의 HD pro webcam C920을 사용하여 영상을 취득하였다.

Table 3.2 HD pro webcam C920 사양

HD pro webcam C920, Features	
Resolution	Full HD 1080p video calling (up to 1920 x 1080 pixels)
Photo pixel	up to 15 megapixels
Lens	Carl Zeiss® lens with 20-step autofocus
Interface	Hi-Speed USB 2.0 certified (USB 3.0 ready)

### 3.1.2 PIC 18F452

PIC는 Programmable Interrupt Controller의 약자이며 이미 널리 사용되는 마이크로 컨트롤러이다. 시대가 변화함에 따라 컴퓨터와 함께 우리의 일상에서 모든 기기들은 자동화 과정을 요구하게 되었다.

이러한 과정으로 기기의 조작은 간편해 지고 작동기기의 성능은 자동화를 거치며 향상되어 졌다. 자동화라는 것은 이미 보편화된 사무 자동화(OA), 공장자동화(FA), 가정자동화(HA)로 크게 나누어 구분되어 지며, 이를 작동시키고 제어하기 위해서는 제어장치가 필요하다.

흔히 우리가 사용하는 예로 TV, 핸드폰, 세탁기, 전기밥솥, 가습기, 원격시동장치, 경보기, 전화기, 팩스, 전자레인지, 냉장고, 전기장판, 선풍기, 에어컨, 온풍기, 전기믹서, 스토브, 커피메이커, 충전기, 초인종, 오디오, 캠코더, 다리미, 프린터, 자동응답기, 마우스, 리모컨, 산업용 카운터, PLC 인버터, 계전기, 유량계, 열량계, 가스누출 검출기, 화재감지기, 자동 소화기, 디지털시계, 조명기기, 의료기기, 장난감 등 많은 기기와 우리 생활에 밀접하게 가전기기, 통신기기, 컴퓨터 주변기기, 자동차, 산업현장 등에 깊이 사용되고 있다.[5]

이러한 장비들이 능동적으로 작동하기 위해서 마이크로프로세서를 제어장치로 사용하는데 이러한 장비와 기기들은 지정된 작동을 구현하기 위해서 슈퍼컴퓨터나 최신형 연산프로세서가 필요하지는 않다. 가장 필요한 동작을 간단하고 저렴하게 구현하며 각 필요한 상황에 맞추어 사용 가능하다는 것이 PIC 마이크로 컨트롤러의 특징이다.[13][14]

Table 3.3 PIC Microcontroller 특징

PIC 마이크로컨트롤러의 특징	
초소형	CPU, ROM, RAM, I/O PORT가 하나의 칩에 집적되어 있음
초저가	리스크가 거의 없음, 개발 중 즉시 기능 개선 가능
실행속도가 빠름	실행코드의 크기가 작고 병렬처리 RISC 구조
소비전력이 적다	CMOS로 되어있어 소비전력이 적고 동작 전압의 범위가 넓음
크기와 타입이 다양	다양한 종류의 PIC가 있고 여러 DEVICE 선택이 가능
개발환경이 우수	개발용 에뮬레이터, 엔진, 소프트웨어가 다양
C언어 지원	내부구조와 I/O PORT의 컨트롤이 용이

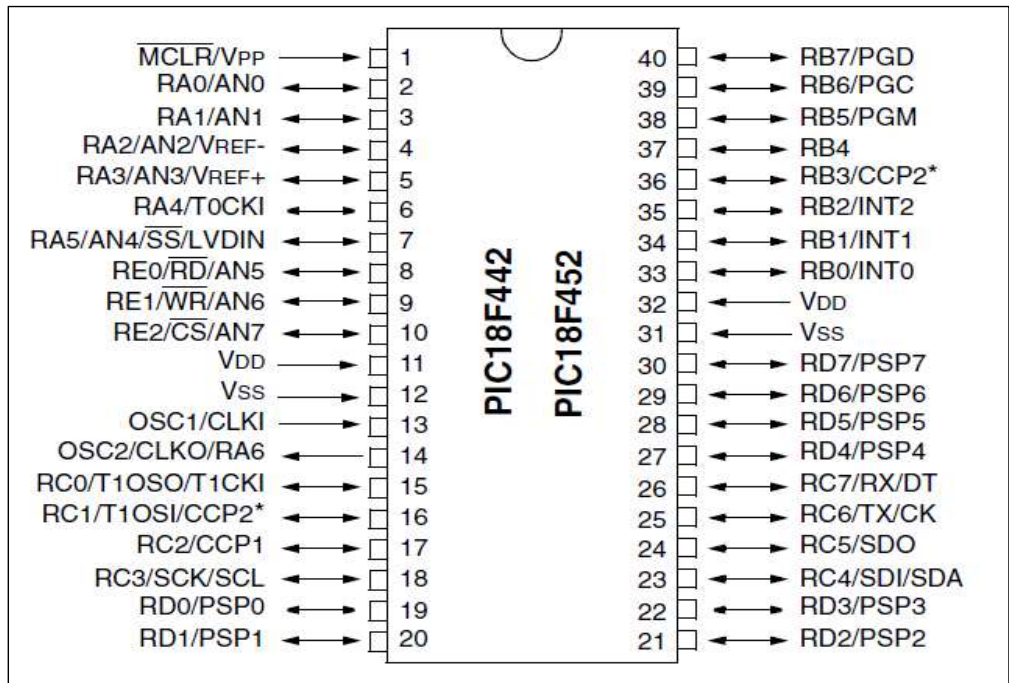


Fig. 3.4 PIC 18F452 Pin Diagram

Table 3.4 PIC 18F452 Features

Features	
Operating Frequency	DC – 40MHz
Program Memory (Bytes)	32K
Program Memory (Instructions)	16384
Data Memory (Bytes)	1536
Data EEPROM Memory (Bytes)	256
Interrupt Sources	18
I/O Ports	Ports A, B, C, D, E
Timers	4
Capture/Compare/PWM Modules	2
Serial Communications	MSSP, Addressable USART
Parallel Communications	PSP
10-bit Analog-to-Digital Module	8 input channels
RESETS (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect	Yes
Programmable Brown-out Reset	Yes
Instruction Set	75 Instructions
Packages	40-pin DIP 44-pin PLCC 44-pin TQFP

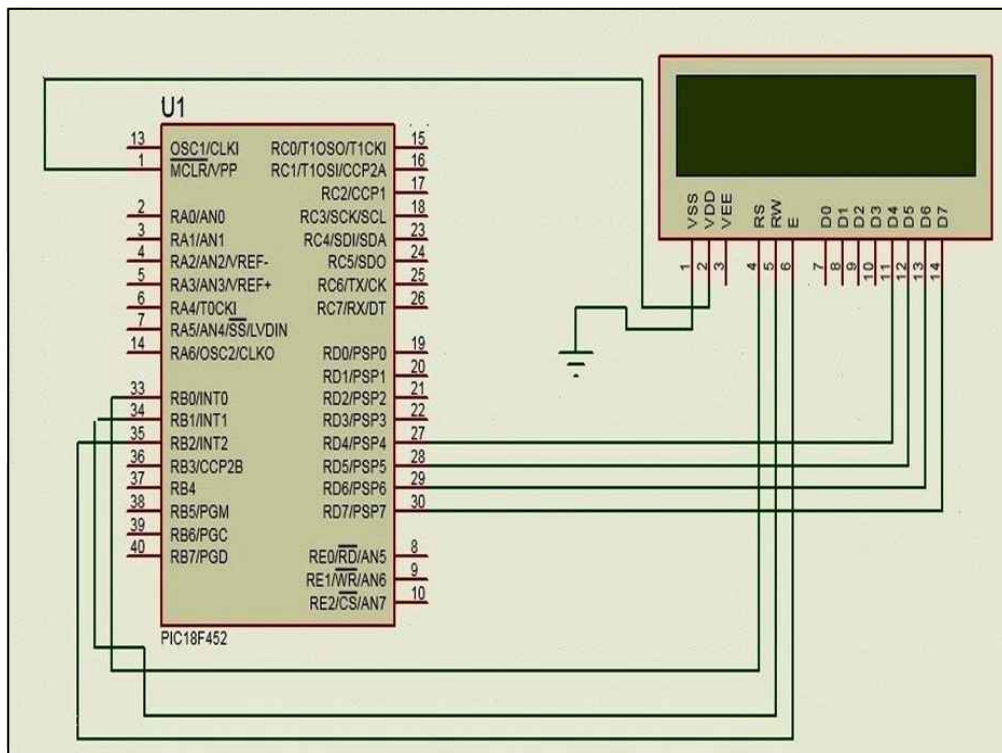


Fig. 3.5 Microprocessor PIC 18F452 and LCD16x2

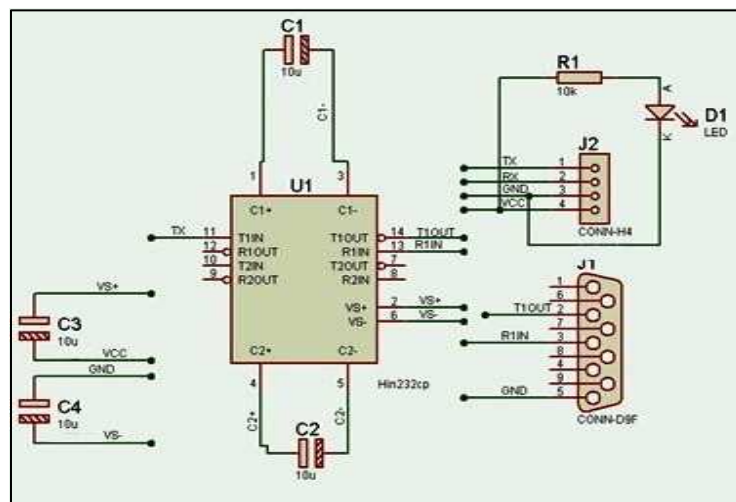


Fig. 3.6 RS232 interface with MAX232



## 제 4 장 실험 및 분석

### 4.1 물체의 추적

영상으로 나타나는 화면의 일정한 영역을 설정 후, 연속적인 움직임에도 물체의 추적이 가능하게 하였다. 설정된 영역은 물체의 움직임에 따라 화면상의 좌·우를 확인하며 서보모터와 함께 움직인다. 움직임의 설정은 OpenCV를 기반으로 하였으며 CAMshift 알고리즘과 퍼지논리를 통해 물체의 설정을 지속 가능하게 하였다.

화면상의 물체의 추적 범위는 640\*480 픽셀로 설정하였다. 물체의 움직임에 따라 좌측, 중앙, 우측으로 영역을 나누었다. 이중 좌측과 우측은 먼 좌측과 먼 우측을 따로 구분하였다.

#### 4.1.1 퍼지논리의 적용

카메라와 물체의 움직임이 동시에 발생하고 개체의 속도가 일정하지 않을 때 대상의 위치는 무작위로 프레임에 나타나게 된다. 화면의 획득 시 대상의 중심 및 주변의 이미지는 카메라의 품질과 환경에 따라 빛의 영향을 받을 수 있다. 또한 서보모터의 각도 제어가 가능할 시 시스템의 응답은 느려질 수 있다.

선택된 영역의 중심  $x$ 와 선택된 영역의 넓이는 퍼지논리를 거쳐 다음과 같이 식(6), (7)로 나타낼 수 있다.

선택 영역의 중심( $x$ ) (6)

= Position of object image center in horizontal axis

선택영역의 넓이(Area) (7)

= Area of object image  $\div (640 \times 480)$

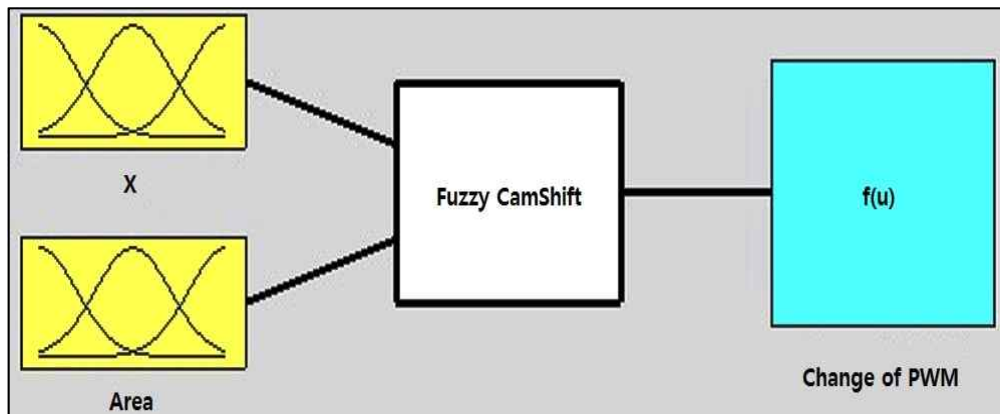


Fig. 4.1 Fuzzy logic controller

$x$ 의 값은 지정영역의 중심이며 대문자 R(large Right), 소문자 r(right), 중앙 C(center), 소문자 l(left), 대문자 L(large Left)로 구분하였다.

Area의 값은  $x$ 의 각 위치에 따라 Small, Normal, Big으로 세 가지로 구분하였다.

이 구분에 따른 PWM의 값은 Table 4.1의 수치로 나타내었다.

Table 4.1 Fuzzy logic rules

Fuzzy logic rules		
IF (X & Area)		THEN $\Delta\text{PWM}(\mu\text{sec})$
X	Object image Area	
large Right	Small	-7
large Right	Normal	-5
large Right	Big	-3
right	Small	-5
right	Normal	-3
right	Big	-1
Center	x(Don't care)	0
left	Big	+1
left	Normal	+3
left	Small	+5
large Left	Big	+3
large Left	Normal	+5
large Left	Small	+7

#### 4.1.2 x의 입력에 의한 소속함수

x의 값은 Fig. 4.2와 같이 x값의 위치에 따른 변화에 따라 중심의 값은 변화하며 그 값에 따른 Area 값을 보여주고 있다.

이때 대문자 L(large Left), 소문자 l(left), 중앙 C(center), 소문자 r(right), 대문자 R(large Right)의 값의 변화에 따른 기준 값은 픽셀로 180, 250, 320, 390, 460의 기준으로 나누어지며 그에 따른 소속함수의 계산은 Table 4.2의 값을 따른다.

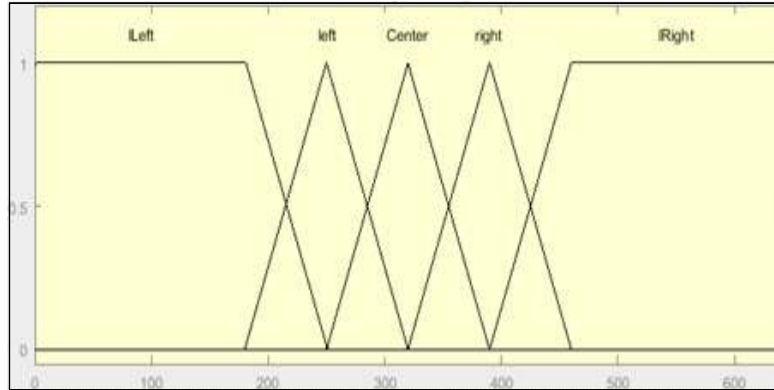


Fig. 4.2 x의 값에 따른 중심의 추적

Table 4.2 x의 값에 의한 소속함수

if((x>0)&&(x<=180))	L=1;
else if((x>180)&&(x<=250))	L=(float)(250-x)/70;
else if(x>250)	L=0;
if((x>0)&&(x<=180))	l=0;
else if((x>180)&&(x<=250))	l=(float)(x-180)/70;
else if((x>250)&&(x<=320))	l=(float)(320-x)/70;
else if(x>320)	l=0;
if((x>0)&&(x<=250))	c=0;
else if((x>250)&&(x<=320))	c=(float)(x-250)/70;
else if((x>320)&&(x<=390))	c=(float)(390-x)/70;
else if(x>390)	c=0;
if((x>0)&&(x<=320))	r=0;
else if((x>320)&&(x<=390))	r=(float)(x-320)/70;
else if((x>390)&&(x<=460))	r=(float)(460-x)/70;
else if(x>460)	r=0;
if((x>0)&&(x<=390))	R=0;
else if((x>390)&&(x<=460))	R=(float)(x-390)/70;
else if(x>460)	R=1;

### 4.1.3 Area의 값에 따른 크기의 추적

x의 값은 Fig. 4.3와 같이 Area의 값에 따라 크기를 Small, Normal, Big의 세 가지로 구분하였고, 값의 변화에 대한 기준 값은 0.04, 0.12, 0.20으로 나누어지며 그에 따른 소속함수의 계산은 Table 4.3의 값을 따른다.

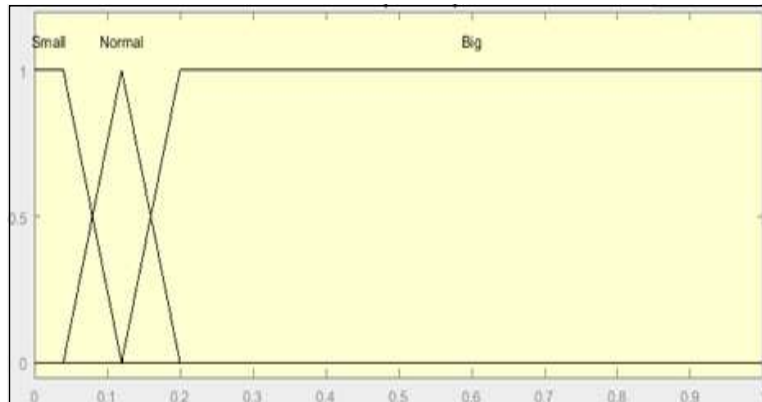


Fig. 4.3 Area의 값에 따른 크기의 추적

Table 4.3 Area의 값에 의한 소속함수

if((a>0)&&(a<=0.04))	S=1;
else if((a>0.04)&&(a<=0.12))	S=(float)(0.12-a)/0.08;
else if(a>0.12)	S=0;
if((a>0)&&(a<=0.04))	N=0;
else if((a>0.04)&&(a<=0.12))	N=(float)(a-0.04)/0.08;
else if((a>0.12)&&(a<=0.2))	N=(float)(0.2-a)/0.08;
else if(a>0.2)	N=0;
if((a>0)&&(a<=0.12))	B=0;
else if((a>0.12)&&(a<=0.2))	B=(float)(a-0.12)/0.08;
else if(a>0.2)	B=1;

#### 4.1.4 x와 Area의 입력에 의한 PWM 출력

PWM의 출력 값은 퍼지논리에 따라 x의 값과 Area의 두 개의 입력 값에 의해 결정되며, 퍼지논리 규칙에 따라 연산되어진 값은 Fig. 4.4와 같이 그래프로 나타낼 수 있다. x의 입력 값과 Area의 입력 값에 의한 논리연산의 연산자는 AND, OR 및 NOT으로 나타내는 퍼지논리로 정의 하였다.

$$A \text{ and } B = \text{Min}(A,B)$$

$$A \text{ or } B = \text{Max}(A,B)$$

$$\text{not}(A)=1-A$$

위와 같은 퍼지논리 규칙에 따라 퍼지논리의 연산과 출력 값은 Table 4.4의 계산을 따른다.

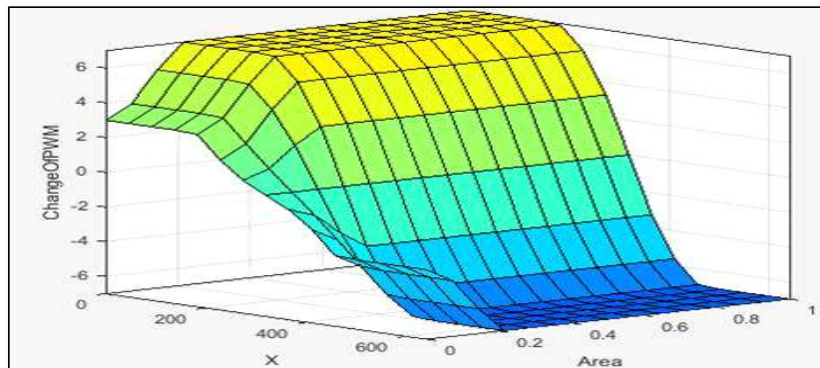


Fig. 4.4 PWM 출력값

Table 4.4 PWM 출력값 계산

$$\begin{aligned} \text{ChangeOfPWM} = & (\text{float})\text{Math}::\text{Round}( \text{Math}::\text{Min}(\text{IR},\text{S})*(-7.0)+ \\ & \text{Math}::\text{Min}(\text{IR},\text{N})*(-5.0)+ \text{Math}::\text{Min}(\text{IR},\text{B})*(-3.0)+ \text{Math}::\text{Min}(\text{r},\text{S})*(-5.0)+ \\ & \text{Math}::\text{Min}(\text{r},\text{N})*(-3.0)+ \text{Math}::\text{Min}(\text{r},\text{B})*(1.0)+ \text{Math}::\text{Min}(\text{l},\text{B})*1.0+ \\ & \text{Math}::\text{Min}(\text{l},\text{N})*3.0+ \text{Math}::\text{Min}(\text{l},\text{S})*5.0+ \text{Math}::\text{Min}(\text{IL},\text{B})*3.0+ \\ & \text{Math}::\text{Min}(\text{IL},\text{N})*5.0+ \text{Math}::\text{Min}(\text{IL},\text{S})*7.0,2); \end{aligned}$$

## 4.2 데이터 취득 결과

### 4.2.1 x의 값이 대문자 L(large Left)일 때 파라미터와 히스토그램

: 추적중인 물체가 녹색이고 카메라를 향하여 가장 좌측으로 설정된 영역에 있으며, 주변색의 간섭 없이 추적은 지속되었고, 히스토그램은 변화 없이 녹색이다.(파라미터의 값은 X가 530, Y가 232)

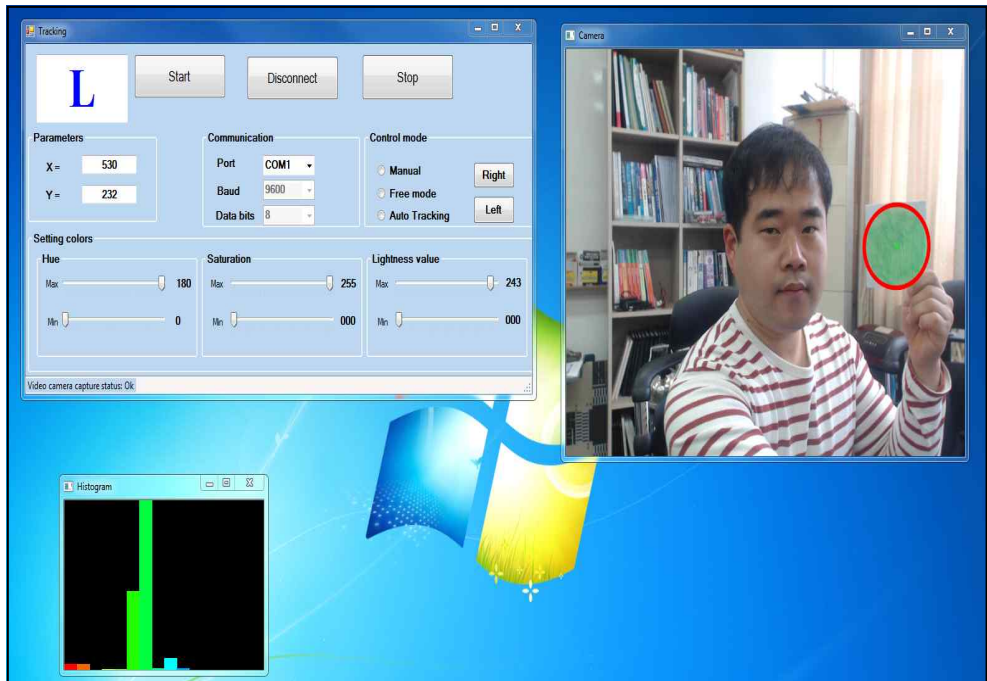


Fig. 4.5 x값이 L(large Left)

#### 4.2.2 x의 값이 소문자 l(left)일 때 파라미터와 히스토그램

: 추적중인 물체가 녹색이고 카메라를 향하여 좌측으로 설정된 영역에 있으며, 주변색의 간섭 없이 추적은 지속되었고, 히스토그램은 변화 없이 녹색이다.(파라미터의 값은 X가 425, Y가 280)

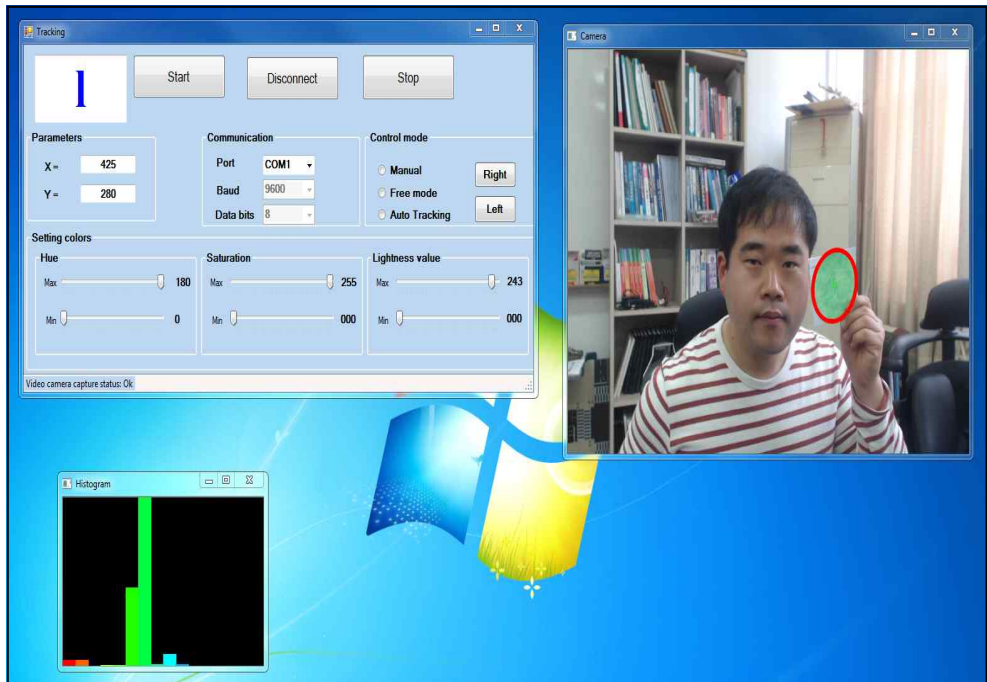


Fig. 4.6 x값이 l(left)



#### 4.2.3 x의 값이 중앙 C(center)일 때 파라미터와 히스토그램

: 추적중인 물체가 녹색이고 카메라를 향하여 중앙으로 설정된 영역에 있으며, 주변색의 간섭 없이 추적은 지속되었고, 히스토그램은 변화 없이 녹색이다.(파라미터의 값은 X가 320, Y가 238)

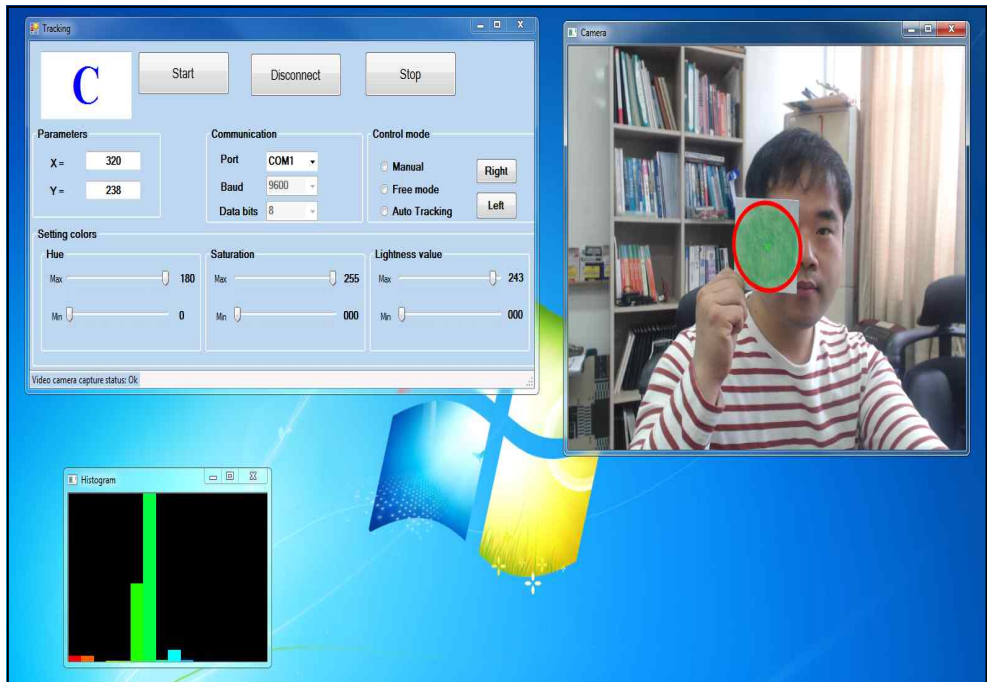


Fig. 4.7 x값이 C(center)

#### 4.2.4 x의 값이 소문자 r(right)일 때 파라미터와 히스토그램

: 추적중인 물체가 녹색이고 카메라를 향하여 우측으로 설정된 영역에 있으며, 주변색의 간섭 없이 추적은 지속되었고, 히스토그램은 변화 없이 녹색이다.(파라미터의 값은 X가 217, Y가 272)

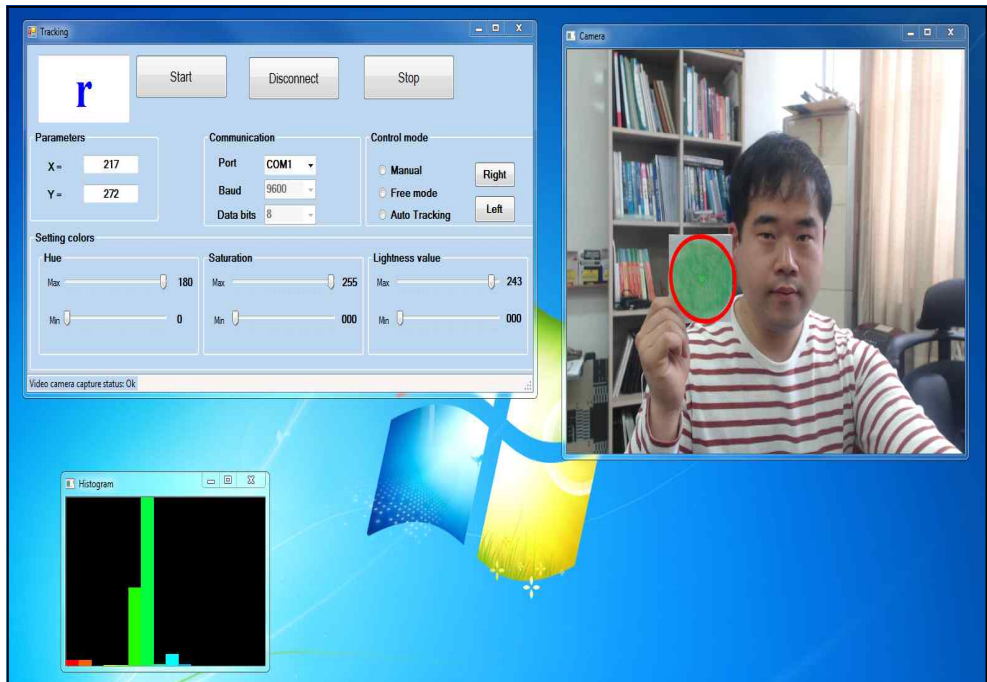


Fig. 4.8 x값이 r(right)

4.2.5 x의 값이 대문자 R(large Right)일 때 파라미터와 히스토그램  
 : 추적중인 물체가 녹색이고 카메라를 향하여 가장 우측으로 설정된 영역에 있으며, 주변색의 간섭 없이 추적은 지속되었고, 히스토그램은 변화가 없이 녹색이다.(파라미터의 값은 X가 131, Y가 241)

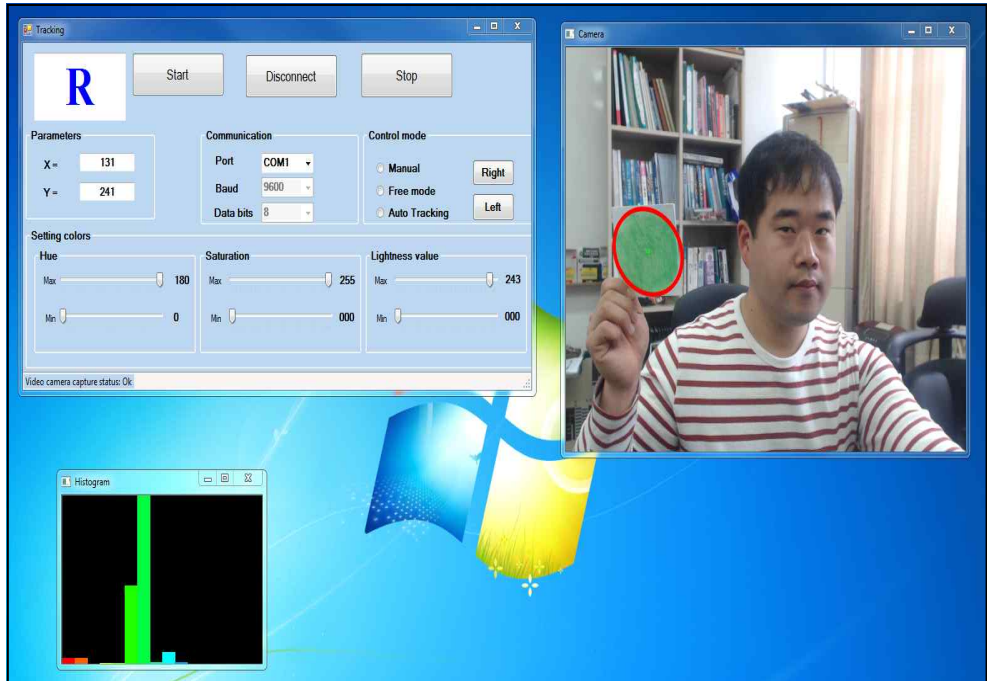


Fig. 4.9 x값이 R(large Right)

#### 4.2.6 Area의 값에 따른 탐색창의 지정영역 크기변화(Big)

: 추적중인 물체가 황색이고 카메라를 향하여 우측으로 설정된 영역에 있으며, 주변색의 간섭 없이 추적은 지속되었고, 히스토그램은 변화가 없이 황색이다. 단 면적의 변화에 따라 추적 지정영역은 Big 이다.(파라미터의 값은 X가 187, Y가 230)

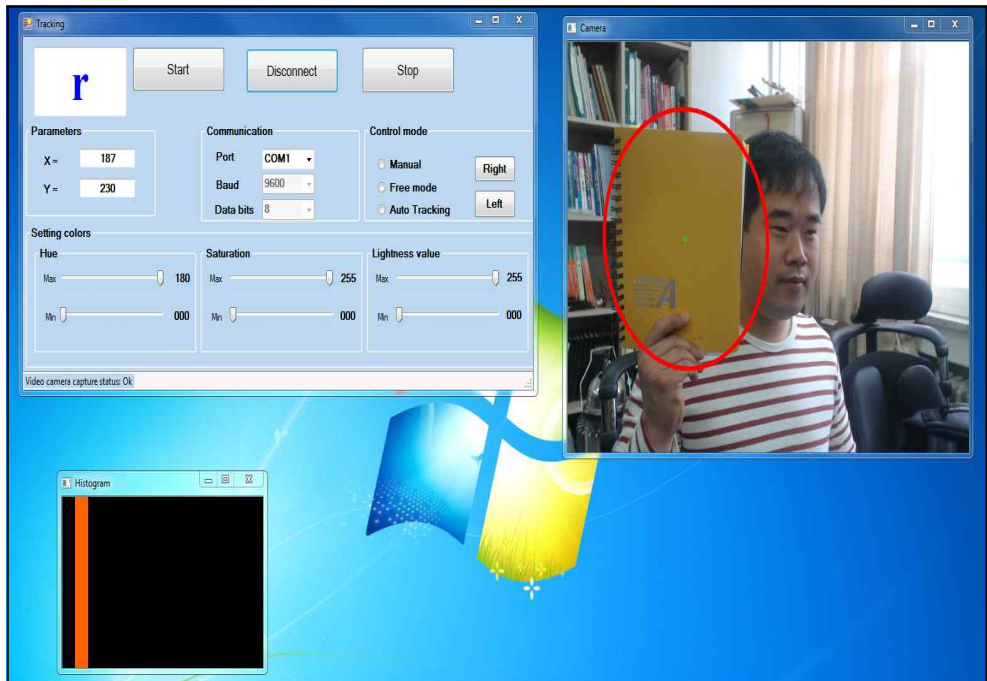


Fig. 4.10 Area 값이 Big

#### 4.2.7 Area의 값에 따른 탐색창의 지정영역 크기변화(Small)

: 추적중인 물체가 회색이고 카메라를 향하여 가장 우측으로 설정된 영역에 있으며, 주변색의 간섭 없이 추적은 지속되었고, 히스토그램은 변화가 없이 회색이다. 단 면적의 변화에 따라 추적 지정영역은 Small 이다.(파라미터의 값은 X가 83, Y가 91)

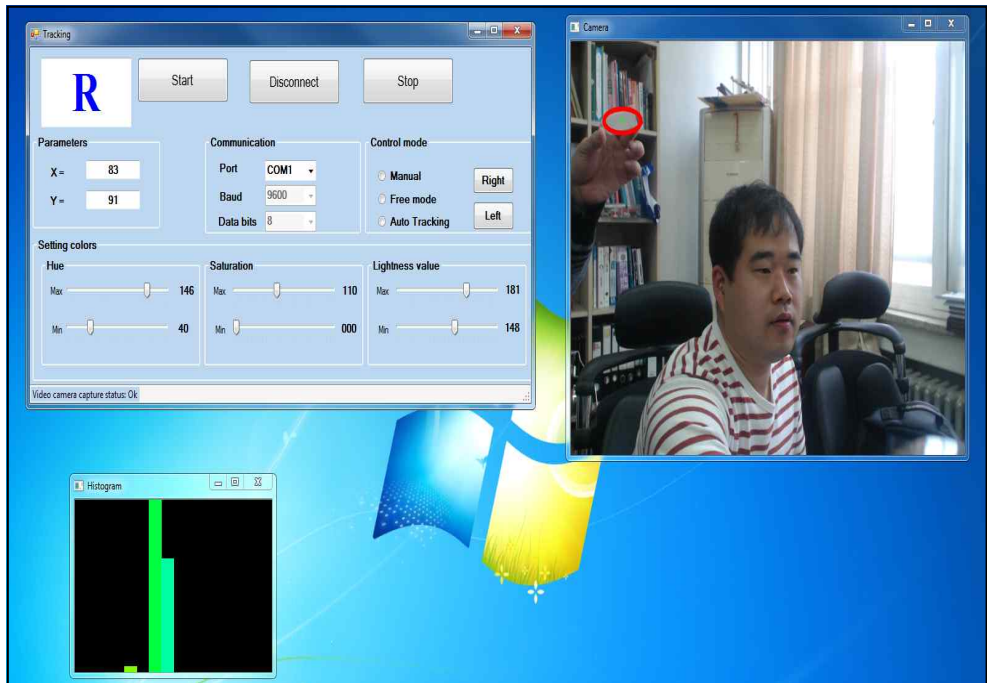


Fig. 4.11 Area 값이 Small

### 4.3 데이터 분석

x의 값과 Area 값에 따른 파라미터의 값의 변화와 지정영역의 변화는 실시간으로 변화량이 확인되었으며 대문자 L로 설정된 영역에서 대문자 R로 설정된 영역까지의 움직임에 결과 값과 서보모터를 움직이는 PWM의 값은 Flowchart Fig. 4.12와 같은 PIC 18F452 & PWM와 Flowchart Fig. 4.13과 같은 C++ Programming으로 작동함을 확인 하였다.

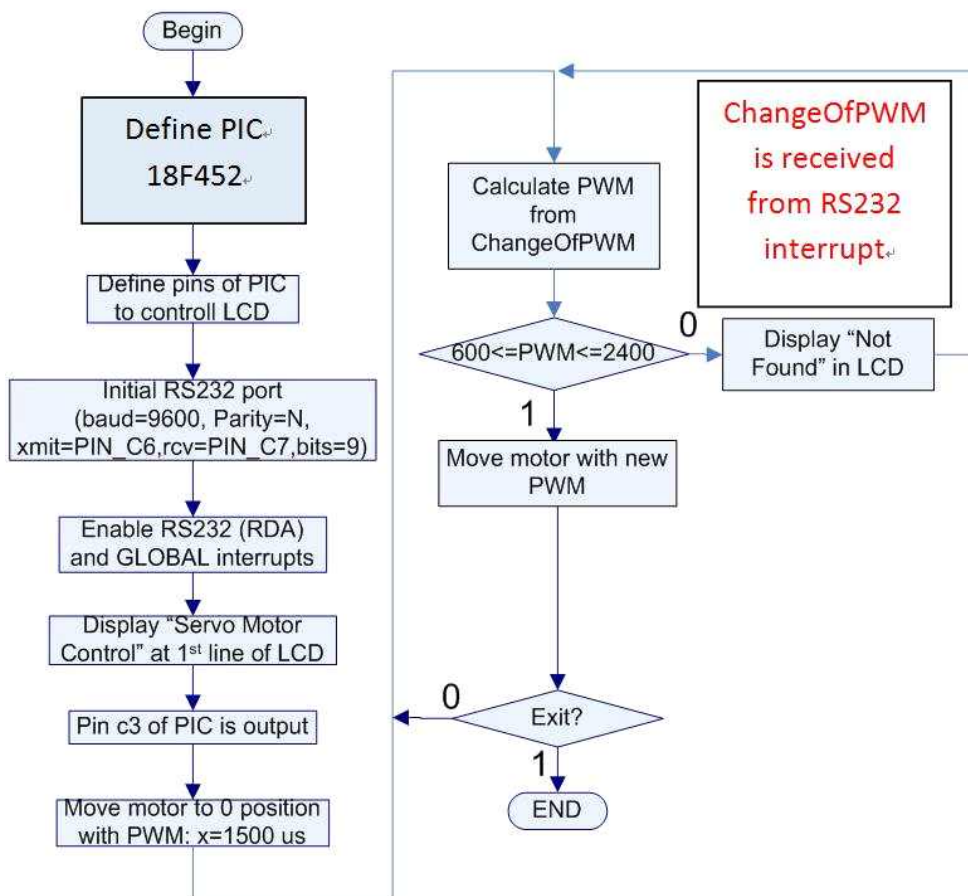


Fig. 4.12 Flowchart of PIC 18F452 & PWM

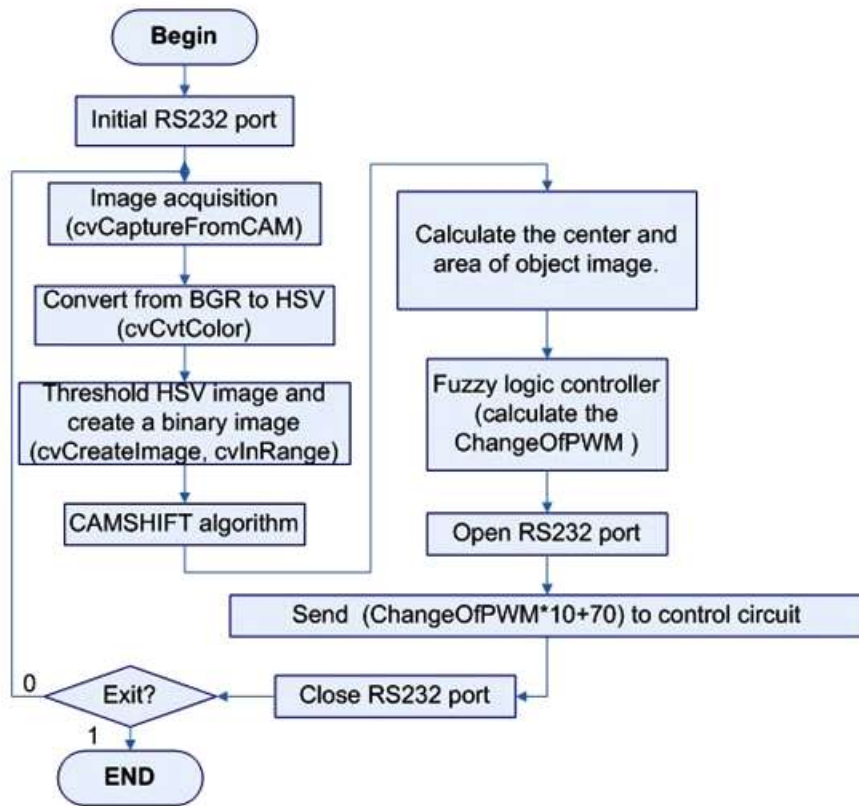


Fig. 4.13 Flowchart of C++ Programing & PWM

또한 이러한 무작위 개체 추적 시스템에서 물체와 시스템의 편차 각을 정확하게 계산 할 수는 없다. 개체 이미지의 중심(x)와 Area에 따라서 계산된 상대적인 개체와 시스템 상에는 카메라의 품질과 환경적인 빛의 영향 등으로 정확한 편차 각을 계산할 수가 없기에 우리는 퍼지논리 제어장치를 적용하였다.

이미지와 Area 영역과 편차 각에 따른 관계는 오토포커스 기능이 있는 카메라에서 탐색대상이 아주 멀리 있을 때, 탐색대상의 Area가 작아지면(물체가 카메라에서 멀다) 편차 각은 커지고, 탐색 대상의 Area가 커지면(물체가 카메라에서 가깝다) 편차 각은 작아진다.

Fig. 4.14 편차 각과 물체의 상

따라서, 편차 각은 추적 물체의 중심의 위치뿐만 아니라, 대상 Area의 값에 따라 달라진다. 이는 퍼지논리 컨트롤러에서 두 개의 입력에 의한 하나의 출력을 나타나게 된다. 두 입력의 대상 이미지의 중심 가로축(x)는 0~640 사이의 영역이다. 그리고 탐색창의 크기는 화상영역의 비율(640\*480)의 안에 위치한다. PWM의 출력은 두 값에 의해 하나의 출력으로 나타나게 된다.



## 4.4 Overall Tracking system

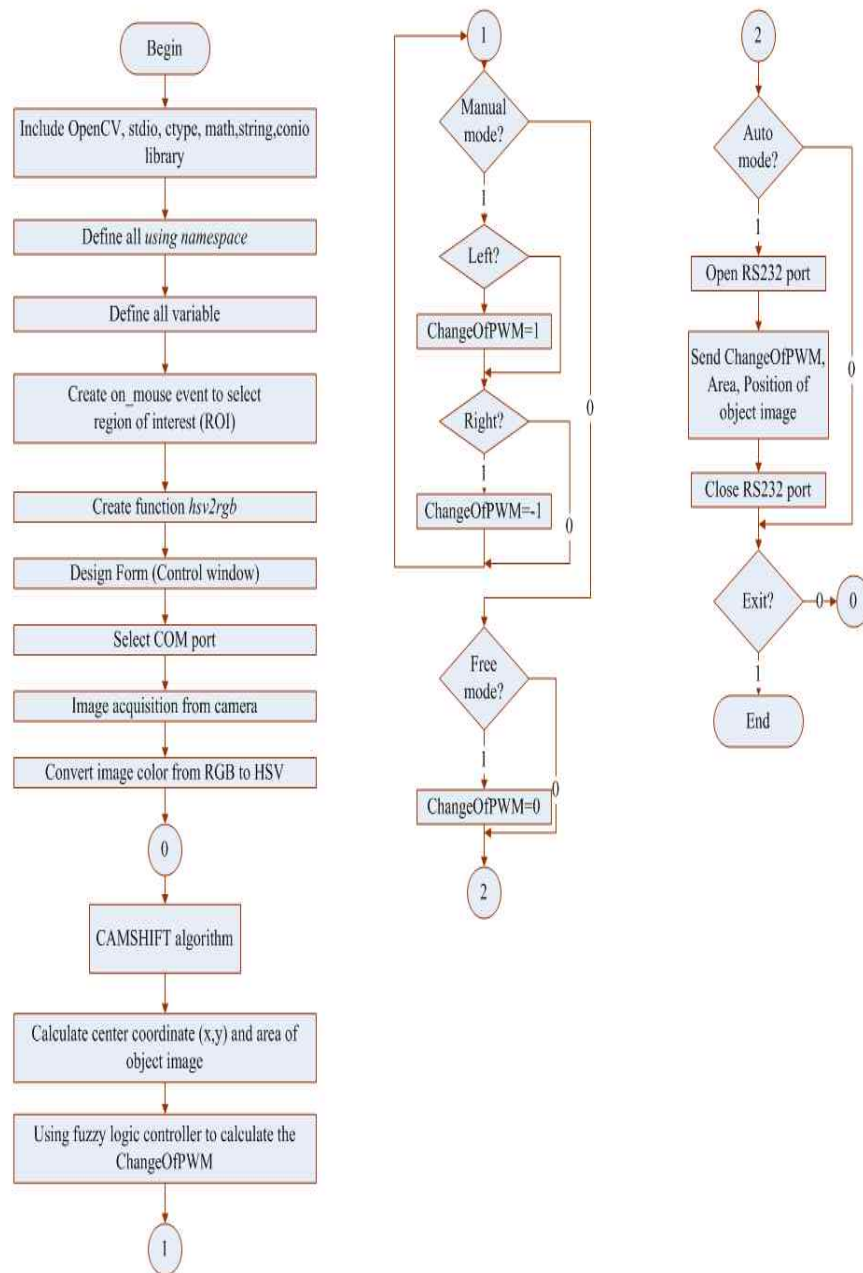


Fig. 4.15 Flowchart of Overall tracking System

## 제 5 장 결론

본 논문에서는 고정된 CCVT와 같은 보안·감시시스템이나 로봇동작시스템의 구현에 있어서 추적하는 물체의 영상의 왜곡과 사각지대(Dead zone)의 발생정도를 줄이고 개선하고자 하였다. 따라서 공개라이브러리인 OpenCV를 Visual studio 2010에서 C++ Programming 작업을 통해 CAMshift 알고리즘과 함께 퍼지이론을 적용하여 작동하도록 하였고 다음을 확인할 수 있었다.

1. CAMshift 알고리즘을 사용하여 화면의 노이즈 발생을 억제 하였고, 비슷한 색상에 방해받지 않고 개선된 화질의 물체의 추적이 가능하게 하였다.
2. 퍼지논리는 물체의 중심점을 산출하고 산출된 값과 화면상의 영역을 계산하여 PWM 값의 변화를 서보모터의 제어에 즉시 반영하였다.
3. 서보모터는 퍼지논리 시스템과 결합하여 화면의 최대 움직임이 발생하면 고정된 카메라의 기능을 넘어서 카메라 각도변화가 가능하게 하였고 빠른 반응 속도를 확인하였다.
4. 이러한 기법들은 감시카메라를 이용함에 있어 넓은 영역을 수동조작 없이 효율적으로 감시할 수 있게 하는 기능을 제공할 수 있으며, 움직이는 물체에 초점을 두고 감시가 가능하다.
5. 감시카메라의 기능 중 녹화기능의 측면에서는 현재 변화가 없는 모든 장면을 녹화하는 기능에서 벗어나 감시 영역에서 물체의 움직임이 감지되었을 때만 녹화, 저장하는 역할을 수행할 수 있다.

추후 연구과제는 촬영 화면 바깥의 서보모터의 동작 각도를 넘어서는 화상의 사각지대(Dead zone)의 영상처리는 추가 연구가 필요하다. 이를 보완할 추

가적인 소프트웨어 기술과 하드웨어를 복합적으로 적용한다면 여러 개의 움직임  
임을 동시에 추적할 수도 있고, 영상의 미세한 움직임도 추적이 가능할 것으  
로 판단된다. 또한 감지영역의 화질 개선 및 미세한 경계 영역의 설정이 보완  
되어 물체주적에 관한 연구가 지속된다면 영상의 사각지대(Dead zone)를 넘어  
특정시설 및 선박의 경계경비, 관리감독 그리고 해양안전구조시스템 등에 보  
다 효율적으로 활용이 가능한 지능형 감시시스템 및 사고예방 자동감시시스템  
이 구축될 수 있을 것이다.

## 참 고 문 헌

- [1] 김동근, “OpenCV programing”, 가메출판사, 2010
- [2] 김대영, 박재완, 이철우, “CAMshift 기법과 칼만 필터를 결합한 객체 추적 시스템”, 멀티미디어학회논문지, 2013, 16(5), pp619-628
- [3] 이기영, “전 방향 영상정보를 이용한 실시간 이동물체추적 시스템에 관한 연구”, 서울산업대학교 석사학위논문, 2005
- [4] 김대영, “CAMshift와 칼만 필터 결합 기반 객체 추적 시스템”, 전남대학교 석사학위논문, 2013
- [5] 신대섭, 신준섭, “PIC 마이크로프로세서 길잡이”, 세화출판사, 2002
- [6] <http://opencv.org/>, 공식사이트
- [7] 류황민, 김주미, 박영산, 배철오, 안병원. “물체 추적시스템에서 퍼지제어기 응용”, 해양환경안전학회 학술발표대회 논문집, 2014, 221-221.
- [8] 홍대선, “공학도를 위한 퍼지시스템 입문”, 문운당, 2010
- [9] 오성권, “프로그래밍에 의한 하이브리드 퍼지추론시스템”, 내하출판사, 2005
- [10] 김은수, 이승현, 이재수, 류충상, “뉴로·퍼지·인공지능 핸드북”, 일본 계측 자동제어학회편, 圖書出版 技多利, 1998
- [11] 최철, “산업용 서보모터의 기술동향 및 설계 요소”, 한국생산제조시스템학회지 19(2), 2010.4, 147-156
- [12] 차영엽, “주기변화에 따른 RC 서보모터 회전범위 비교실험”, 제어로봇시스템학회 논문지 17(11), 2011.11, 1179-1182
- [13] 이희문, “PIC16F84의 기초+α”, 성안당, 2002
- [14] Microchip社, “PIC18F2420/2520/4420/4520 Data Sheet:PIC\_manual”, Microchip Technology, 2004
- [15] <http://100.daum.net/>, 다음백과사전

## 감사의 글

학업에 대한 열정으로 무작정 시작하였던 석사과정을 울여름 마무리하며 너무나도 행복합니다. 저의 많은 부족함에도 하나하나 세심한 지도와 가르침으로 지금 이 순간의 기쁨과 행복을 느끼게 해주신 박영산 지도교수님께 무한한 감사를 드립니다. 앞으로의 학업과 인생에 있어서도 교수님의 가르침과 조언은 저의 가슴 깊이 남을 것입니다.

제자들에 대한 사랑으로 항상 따뜻한 마음으로 다가와 주신 안병원 교수님과 항상 밝은 미소로 맞아주시는 배철오 교수님 부족한 저의 논문지도와 가르침 정말 감사합니다. 김현수 교수님, 남택근 교수님 열정적인 수업 감사 합니다.

실습선에서 저에게 학업에 대한 열정과 배려를 베풀어 주신 임증수 교수님, 김홍렬 교수님 감사합니다. 또한 조교시절 석사 진학의 뚜렷한 목표를 가지게 해주신 김성종 교수님 감사합니다. 그리고 공부하는 학생의 자세와 연구에 대한 열정을 가르쳐주신 임정빈 교수님 깊은 감사의 마음을 글로 전합니다.

저의 학업에 격려와 조언을 해주신 우리학교 많은 교수님들의 사랑과 관심으로 지금 이 논문을 맺을 수 있었습니다. 깊은 감사의 마음을 다시 전합니다.

해양문화재연구소 수증발굴과 문화석 과장님 꾸준한 학업에 대한 지원 및 격려 감사합니다. 그리고 누리안호 정명화 선장님, 씨뮤즈호 오천석 선장님 많은 배려에 감사를 드립니다.

우리 연구실 경민, 성근, 주미, 현주, 선덕, 밍, 티엔 모두 저를 도와주고 힘을 줘서 고맙습니다. 항상 좋은 상담을 해주는 윤석선배, 김수선배 감사합니다.

그리고 항상 힘과 용기를 주는 미정, 태형, 영민, 수산 너무나도 감사합니다. 그리고 병호, 민성, 승중, 형규, 번 옆에서 마음을 다잡아준 친구들 고맙습니다.

목포에서 광양으로 가깝지만 자주 함께하지 못한 사랑하는 나의 외할머니와 어머니, 동생 계속 응원해줘서 감사합니다.

2016년 7월

심 승 섭