

# 물체탐지 시스템의 개발을 위한 근거리 레이더에 대한 특성 분석

박 동 진\*, 유 인 환\*, 변 기 훈\*\*, 이 상 민\*\*\*, 권 장 우°

## An Analysis on Short-Range-Radar Characteristic for Developing Object Detecting System

Dong-Jin Park\*, In-Hwan Ryu\*, Ki-Hoon Byun\*\*, Sang-Min Lee\*\*\*, Jang-Woo Kwon°

### 요 약

본 논문에서는 근거리 레이더의 특성 분석을 통한 선박 내 안전을 위한 물체탐지 시스템의 개발에 초점을 두었다. 근거리용 레이더들은 가격이 저렴한 대신에 자동차와 같은 특수 용도로 개발되었으므로 다른 곳에 적용하기 위해서는 기기의 특성 분석과 다양한 환경에서의 실험을 통한 데이터 특성 분석이 필수적이다. 또한 본 연구의 목표인 사람 및 물체의 정확하고 신속한 감지 시스템의 구성을 위해서는 입력되는 많은 반사 신호들 중에서 이동하는 물체만을 정확하게 분류하여 표시하기 위한 데이터 클러스터링 알고리즘이 필요하다. 본 연구에서는 이를 위해 반사되는 데이터들의 궤적정보와 속도 및 RCS 값을 이용한 클러스터링 알고리즘을 제안 하였다.

**Key Words** : Radar, Controller Area Network, Automotive Detection, Detection System

### ABSTRACT

In this paper, we suggest the development of object detection systems for the safety of the ship through the study of the properties of short-range radar. Many of the short-range radars developed for special purpose like cars has cheaper price advantages but it is not proper to every application. In order to overcome such obstacles we need to analysis data from experiments in various environments and feature analysis of the device is essential. Also, the data clustering algorithms to display correct classified moving objects is necessary. In this paper we propose the advanced fast moving object detection system using short range radars with better detection accuracy. And we proposed a clustering algorithm using the value of the RCS and the speed and trajectory information of the radar data that are reflected.

※ 본 연구는 미래창조과학부 및 정보통신산업진흥원의 IT융합 고급인력과정 지원사업의 연구결과로 수행되었음(NIPA-2014-H0401-14-1009)

※ This work was supported by INHA UNIVERSITY Research Grant(INHA-44769)

◆ First Author : Inha University Department of Computer Science, lovely.jb0514@gmail.com, 학생회원

° Corresponding Author : Inha University Department of Computer Science, jwkwon@inha.ac.kr, 정회원

\* Inha University Department of Computer Science, pasteur@naver.com

\*\* Inha University Department of Statistics, pioneerbkh@gmail.com

\*\*\* Inha University Department of Electronic Engineering, sanglee@inha.ac.kr

논문번호 : KICS2014-10-394, Received October 8, 2014; Revised November 27, 2014; Accepted November 27, 2014

## I. 서 론

레이더는 펄스 전파의 직진성, 반사성 등을 이용하여 거리와 방향을 탐지하기 위하여 사용된다. 과거에는 비와 구름의 양을 감지하여 기상의 정보를 알려주는 기상레이더, 비행기 등을 감지하는 군사용 레이더, 선박의 운항을 위해 사용되는 레이더 등 대형의 장거리용 레이더가 많이 개발되었다. 하지만 Adaptive Cruise Control(ACC)를 위한 적은 비용의 근거리 레이더에 대한 연구가 이루어지면서<sup>[1]</sup> 레이더는 새로운 방향으로의 효용성을 가지게 되었다. 최근에는 ACC를 탑재한 자동차들이 상용화됨에 따라 점차 다양한 소형의 근거리 레이더가 개발되고 있다. 이러한 근거리 레이더를 이용한 다양한 분야에 대한 연구가 활발히 진행되고 있으나<sup>[2]</sup>, 대다수가 ACC 또는 Volvo사가 참여하고 있는 Safe Road Trains for the Environment(SARTRE) 프로젝트의 Platoon 주행 등을 위해 자동차에 탑재되고 있다. 이에 근거리 레이더의 특성을 분석하여 자동차뿐 아니라 사람까지 감지하는 통합 감지 시스템을 구성하고자 한다. 이를 이용하여 최근에 불거진 선박안전과 관련하여 사람의 안전뿐 아니라 선박의 접안 등에 사용하고자 하는 것이 이 연구의 목적이다. 악천후 혹은 야간 환경에서 카메라만을 이용하면 사람의 움직임을 관측하기 힘들므로, 기후나 밝기에 따른 영향이 적은 레이더를 이용하여 선박 내 돌발적 사고를 줄이는데 연구의 초점을 맞추었다.

본 논문에서는 Continental사의 ARS308을 분석하여 레이더 센서의 특성을 연구하고, 이를 이용한 물체 감지 시스템의 구성에 연구의 초점을 두었다. Continental사의 ARS308은 비교적 가격이 저렴하고 최대 200m 내의 이동물체를 감지할 수 있으나, 본래 차량에의 응용 용도로 개발되었으므로 다른 곳에 적용하기 위해 기기의 특성 분석과 다양한 환경에서의 실험을 통한 데이터 특성 분석이 필수적이다. 또한 본 연구에서 목표로 하고 있는 이동 물체의 정확하고 신속한 감지 시스템의 구성을 위해서 입력되는 레이더 신호들 중에서 이동하는 물체만을 정확하게 분류하여 표시하기 위한 데이터 클러스터링 알고리즘이 필요하여 데이터 분석을 통하여 검지물체의 시간 축에 따른 궤적정보와 속도 및 RCS 값을 이용한 휴리스틱 알고리즘을 통하여 구현하였다. RCS(Radar Cross Section) 값이란 레이더에서 쏜 전자파가 물체에 반사되어 돌아올 때, 반사체의 반사량을 나타내기 위해 규정한 평면 면적이다.<sup>[3]</sup> 본 논문은 시스템 구성을 위해

사용된 Continental사의 ARS308의 하드웨어 특성과 데이터 구성, 제안시스템의 구성 및 물체감지 SW의 구성 제안 알고리즘 및 실험 및 결론의 순으로 이루어져 있다.

## II. 레이더 하드웨어 특성

본 연구에 사용된 ARS308은 Continental사에서 개발한 레이더 센서 시스템이다. CAN 네트워크에 연결되어 데이터를 출력하고, 환경 설정이 가능하도록 구성되어 있다. 이 시스템은 본래 고급 자동차의 물체감지 지원 시스템을 위해 개발되었으며 일반적인 응용을 위하여 데이터의 후처리가 필수적이다.



그림 1. ARS308 레이더 센서  
Fig. 1. ARS308 radar sensor

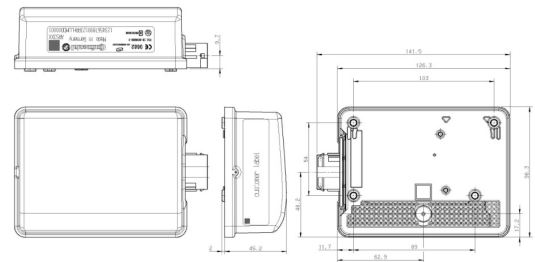


그림 2. ARS308 레이더 센서의 실제 치수  
Fig. 2. Actual dimensions of the ARS308 radar sensor

### 2.1 CAN(Controller Area Network)

표준 CAN버스는 BOSCH에 의해 개발된 최대 전송률 1Mbps의 다중-마스터(multi-master) 메시지 브로드캐스트 시스템이다.<sup>[4]</sup> USB나 이더넷 등의 전통적인 네트워크와는 달리 CAN 네트워크는 엔진 온도나 남은 연료량 등의 많은 짧은 메시지를 네트워크 전체에 브로드 캐스트 한다<sup>[5]</sup>. CAN 표준은 종래의 복잡한 차내 배선 문제를 단 두 가닥으로 이루어진 버스로 대

표 1. CAN1\_Obj\_1의 필드 의미와 위치, 길이, 바이트 오더, 값의 형식, 해상도, 오프셋 등을 나타낸 표  
Table. 1. Table denotes properties like location, length, endianness, type, resolution offset and range of each CAN1\_Obj\_1 field

signal	start	len	byte order	value type	res	offset	value range
Obj_RolCount	0	2	Motorola	Unsigned	1	0	0 → 3
Obj_ID	2	6	Motorola	Unsigned	1	0	0 → 63
Obj_LongDispl	21	11	Motorola	Unsigned	0.1m	0m	0m → 204m
Obj_VrelLong	25	12	Motorola	Unsigned	0.0625m/s	-128m/s	-128m/s → +127.9375m/s
Obj_AccelLong	32	9	Motorola	Unsigned	0.0625m/s <sup>2</sup>	-16m/s <sup>2</sup>	-16m/s <sup>2</sup> → +159375m/s <sup>2</sup>
Obj_ProbOfExist	48	3	Motorola	Unsigned	1	0	0 → 7
Obj_DynProp	51	3	Motorola	Unsigned	1	0	0 → 7
Obj_LatDispl	54	10	Motorola	Unsigned	0.1m	-52m	-51.9m → 52m
Obj_Length	56	3	Motorola	Unsigned	1	0	0 → 7
Obj_Width	59	3	Motorola	Unsigned	1	0	0 → 7
Obj_MeasStat	62	2	Motorola	Unsigned	1	0	0 → 3

체하기 위해 ISO(International Standardization Organization)에서 정의하였다. 표준은 또한 전기적 잡음에 대한 강한 저항성과 자가 진단, 데이터 손상의 하드웨어적 보정을 요구한다. CAN 통신 프로토콜은 CAN 프레임의 식별자 길이에 따라 Standard CAN과 Extended CAN으로 나뉜다. 하지만 본 연구에 쓰인 레이더 센서가 Standard CAN을 사용하는 관계로 Extended CAN에 대한 설명은 생략한다.

Standard CAN의 메시지 형식 표 1은 감지된 객체의 위치, 속도 등을 표시하는 두 프레임 쌍 중 하나인 CAN1\_Obj\_1의 필드 의미와 위치, 길이, 해상도, 오프셋 등을 나타낸 것이다. 메시지에서부터 필요한 데이터를 얻으려면 해당 필드의 정보만을 골라내기 위한 비트 단위 마스킹과 논리 시프트 연산이 필요하다. 얻고자 하는 데이터가 실수일 경우, 골라낸 데이터에 주어진 해상도를 곱하고 오프셋을 더하여야 한다.

### 2.2 ARS 308의 메시지 구조

본 연구에 쓰인 ARS 308 메시지의 그림 3에 나타난 것처럼 ID는 0x200, 0x60B, 0x60C로, 각각 레이더의 환경 설정, 객체의 위치, 객체의 추가적인 정보를 담고 있다. 그 중 0x200 메시지는 레이더의 설정에 필요한 정보들을 담은 메시지(radar configuration message)이다.

표 2는 Standard CAN의 메시지 형식을 나타낸다. standard CAN은 ID(identifier)가 11이다. 주요 필드는 다음과 같다. SOF(start of frame)은 CAN 프레임의 시작을 표시하는 필드이고, ID는 프레임의 우선순위를 나타내며, DLC(data length code)는 데이터 필드의 길이를 나타낸다. Data 필드는 프레임의 실제 데

이터를 포함한다.

ARS 308 레이더 센서 ARS 레이더 센서는 주위 환경을 분석하기 위해 강한 전자기파를 방사하여 반사되는 신호를 여러 단계를 거쳐 처리하여 분석한다. 한 물체에 대해 하나 이상의 반사 신호가 감지될 수 있으나 각각의 반사 신호에 대한 이력(history)을 유지하는 object 출력 모드에서는 이들을 묶어서 하나의 객체로 출력한다.

표 2. 표준 CAN 메시지 형식  
Table 2. Standard CAN message format

S	I	R	I	R	D	D	C	A	E	I
O	D	T	D	O	L	a	R	C	O	F
F		R	E		C	t	C	K	F	S

## III. 제안 시스템 구성

### 3.1 레이더 데이터 수집 하드웨어

본 연구에 사용된 레이더는 연결된 CAN 버스를 통해 일정주기로 데이터를 전송하며 이러한 레이더의 출력을 얻기 위하여 CAN 버스를 통해 오가는 모든 데이터를 빠짐없이 모니터링하고 저장할 장치가 필요하다.

이러한 목적을 달성하기 위해 본 연구에서는 그림 4와 같은 시스템을 구성하여 데이터의 취득 및 분석을 수행하였다. 또한 보통의 PC에는 CAN통신을 위한 포트가 없으므로 그림 5의 Lawicel사의 CANUSB 장치를 이용하였다. CANUSB는 매우 작으며 값싼 CAN 연결 장비로, PC의 USB 포트에 장

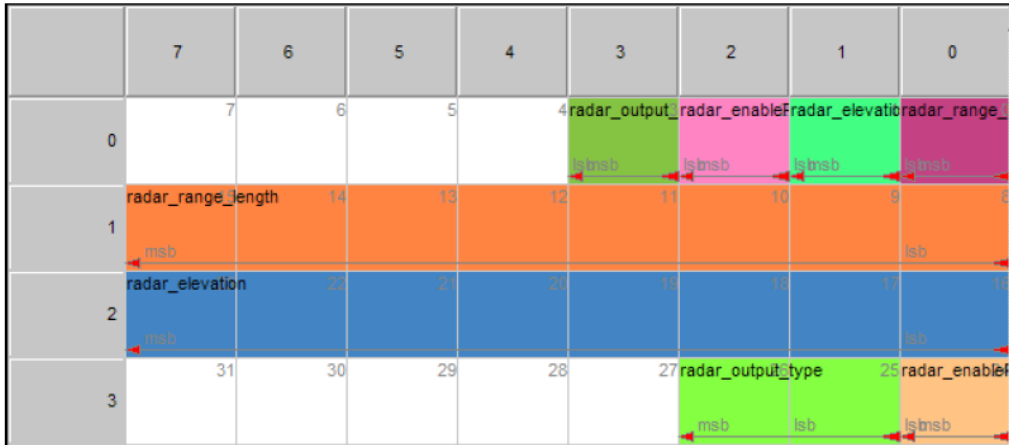


그림 3. 레이더 설정 메시지 (0x200)  
Fig. 3. Radar configuration message (0x200)

착함으로써 즉시 어떤 CAN 버스에도 접근 가능하다. 또한 이 장치는 표준 RS232 직렬 포트로의 소프트웨어 추상화도 제공하므로 별도의 드라이버를 요구하지 않는다. 또한 윈도 환경을 위한 별도의 동적 링크 라이브러리를 이용하면 보다 높은 처리율을 달성할 수 있다.

그림 3은 해당 메시지의 필드 위치와 길이를 한 눈에 알기 쉽게 표시한 표이고, 표 3은 해당 메시지 필드들의 위치와 길이, 바이트 순서, 부호 유무, 해상도, 범위 등을 나타낸 표이다. 예를 들어,

레이더 센서의 출력 형식(radar\_output\_type)을 SendObjects로 변경하고자 한다면, radar\_output\_typeV 필드를 1(valid)로 설정 후, radar\_output\_type 필드에 SendObjects에 대응되는 1을 대입하여, 열린 CAN 소켓 디스크립터에 대해 write(2) 시스템 콜을 사용하여 해당 메시지를 전송한다.

본 연구의 데이터 수집에는 리눅스 시스템이 이용되었다. Lawicel의 CANUSB를 본 환경에서 사용하기 위해 SocketCAN이 사용되었다. SocketCAN이란 리눅스 커널의 공식 CAN API로서, CAN 모니터링 장치의 종류에 상관없이 하드웨어-독립적인 소켓 기반의 CAN 통신을 제공하므로<sup>[6]</sup> 일반적인 Unix-like 환경에서의 네트워크 소켓을 사용하는 것과 같이 그림 6처럼 쉽고 간편한 CAN 프레임 수신과 전송이 가능하다.

SocketCAN은 리눅스 커널의 일부로 포함되어 있으나 일반적인 배포판은 해당 기능이 활성화되어 있지 않다. 그러므로 SocketCAN을 이용하기 위해서는 커널 환경 설정을 통해 기능을 활성화하고 커널 컴파

표 3. 레이저 설정 메시지 (0x200)를 기술한 표  
Table 3. Radar configuration message (0x200) description

Signal	Start	Len	Byte Order	Value Type	Res	Value Range
radar_range_lengthV	0	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>0 -&gt; Invalid</li> <li>1 -&gt; Valid</li> </ul>
radar_range_length	15	8	Motorola	Unsigned	1m	50, 51... 200 m
radar_output_typeV	3	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>0 -&gt; Invalid</li> <li>1 -&gt; Valid</li> </ul>
radar_output_type	26	2	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>0 -&gt; inapplicable</li> <li>1 -&gt; SendObjects</li> <li>2 -&gt; SendTargets</li> <li>3 -&gt; inapplicable</li> </ul>
radar_enablePowerReductionV	2	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>0 -&gt; Invalid</li> <li>1 -&gt; Valid</li> </ul>
radar_elevationV	1	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>0 -&gt; Invalid</li> <li>1 -&gt; Valid</li> </ul>
radar_enablePowerReduction	24	1	Motorola	Unsigned	1	<ul style="list-style-type: none"> <li>0 -&gt; False</li> <li>1 -&gt; True</li> </ul>
radar_elevation	23	8	Motorola	Unsigned	0.25 deg	0, 0.25... 32 deg (Road->Sky)

일 후 새로이 설정된 커널을 시스템에 탑재하여야 한다.

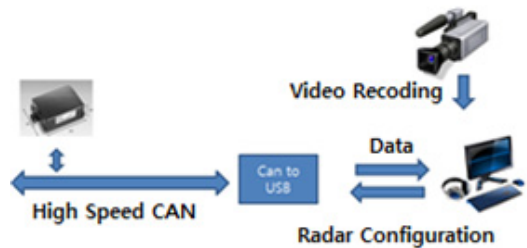


그림 4. 레이더 시스템의 구성도  
Fig. 4. Diagram of radar system



그림 5. Lawicel사의 CANUSB  
Fig. 5. Lawicel CANUSB

```
int main(int argc, char **argv){
    struct ifreq ifr;
    struct sockaddr_can addr;
    struct can_frame frame;

    s = socket(PF_CAN,SOCK_RAW,CAN_RAW);

    strcpy(ifr.ifr_name, "can0");
    ioctl(s, SIOCGIFINDEX, &ifr);

    addr.can_ifindex = ifr.ifr_ifindex;
    addr.can_family = PF_CAN;
    bind(s, (struct sockaddr *)&addr, sizeof(addr));

    frame.can_id = 0x23;
    strcpy((char *)frame.data, "hello");
    frame.can_dlc = 5;
    write(s, &frame, sizeof(frame));

    close(s)
    return 0;
}
```

그림 6. SocketCAN 사용자 공간 프로그램의 간단한 예  
Fig. 6. A simple example of SocketCAN userspace program

### 3.2 레이더 데이터 수집 소프트웨어

본 연구의 데이터 수집을 위한 프로그램은 Qt5 프레임워크로 작성되었다. Qt5란 어플리케이션과 UI 구성을 위한 교차-플랫폼 프레임워크로서 C++와 QML 프로그래밍 언어를 이용하며, 플랫폼-독립적인 어플리케이션을 빠르고 쉽게 구성할 수 있다.

데이터 수집 어플리케이션은 1) 데이터 수집 모듈: CAN 버스로부터 수신한 데이터 중 관심 있는 데이터를 골라내고, 2) 메인 모듈: 수신한 데이터를 일정한 형식에 따라 파일에 쓰고, 3) 그래프 모듈: 수신한 데이터를 4) 어플리케이션 UI의 그래프 영역에 그리는 세 가지 기능을 가진다. 그림 7은 레이더로부터 메시

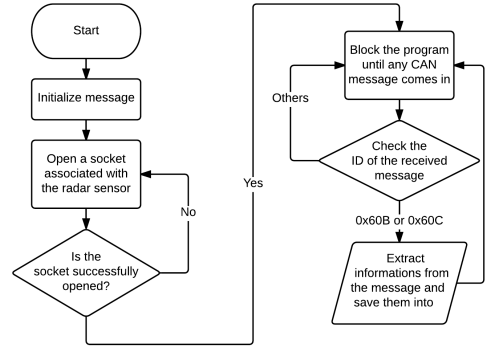


그림 7. 데이터를 획득하는 응용프로그램의 순서도  
Fig. 7. Control flow of the data acquisition application

지를 받는 과정을 보여준다. 다음은 그림 8의 각 기능을 담당하는 모듈에 대한 설명이다.

#### 3.2.1 데이터 수집 모듈

SocketCAN 인터페이스를 통해 CAN 버스를 통과하는 모든 데이터를 수집하고 원하는 프레임만을 필터링한다. CAN 2.0 A 표준 프레임형식은 데이터 필드의 최대 길이를 8 바이트로 한정하므로 모든 데이터를 한 프레임에 집약하는 것은 불가능하다. 그러므로 본 연구에 사용된 ARS308 레이더는 한 객체당 프레임 ID를 0x60B와 0x60C로 갖는 두 개의 프레임에 나누어 40쌍으로 출력한다.

SocketCAN를 이용한 CAN 버스 프로그래밍에서, 프레임의 ID에 따른 프레임 필터링은 조건문을 사용해 간편하게 이루어진다. 수집을 원하는 프레임만을 골라낸 후에는 수집된 데이터의 가공이 필요하다. 수집된 데이터는 ARS308 기술 문서에서 정의하는 임의의 고정소수점 형식으로 이루어져 있으므로 이를 범용 PC 환경에서 사용하기 위해 IEEE 754에서 정하는 표준 부동 소수점 표현 방식으로 변환하여야 한다. 이 표현 방식은 GCC(GNU compiler collection)와 g++ C++ 컴파일러의 단정밀도(single precision) 실수 자료형 float과 배정밀도(double precision) 실수 자료형 double을 포함한다.

이 ARS308의 고정 소수점 데이터의 부동 소수점 데이터로의 변환도 간편하고 빠른 연산을 통해 가능하다. 프레임에서 변환을 하고자하는 필드만을 비트단위 마스크킹으로 추출 후, ARS308 기술문서에 정의된 해상도를 곱한 후, 오프셋을 더함으로써 변환이 가능하다. 해당 과정을 프레임 ID 0x60B에 정의된 Obj\_LongDispl 필드에 적용해보면 다음의 표 4와 같다.

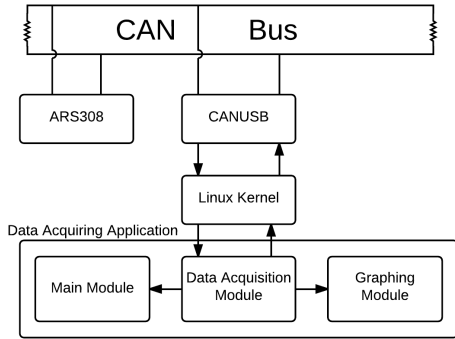


그림 8. 실험환경의 하드웨어와 소프트웨어의 구성  
Fig. 8. Hardware and software configuration of experiment environment

표 4. 프레임“0x60B”에 정의된 “Obj\_LongDispl” 필드위치  
Table. 4 The location of field “Obj\_LongDispl” in a frame whose identification field equals to “0x60B”

	7	6	5	4	3	2	1	0
0								
1								
2								

ARS308 기술문서에 따르면 Obj\_LongDispl 필드는 프레임의 2 번째 바이트에서부터 1 번째 바이트에 걸쳐 있다.

이 필드의 LSB(least significant bit)는 최하위 비트인 2 번째 바이트의 5 번째 비트이고 MSB(most significant bit)는 1 번째 바이트의 7 번째 비트이다. 따라서 그림 9에 나타난 것처럼 2 번째 바이트를 5 오른쪽 시프트하고, 1 번째 바이트를 3 왼쪽 시프트한 후, 논리 OR 연산 후, ARS308 기술 문서의 해상도 0.1[m]를 곱하고, offset 0[m]를 더함으로써 원하는 부동 소수점 형식의 데이터를 얻을 수 있다.

이렇게 부동소수점으로 변환된 데이터를 얻은 데이터는 데이터의 유실을 방지하기 위한 버퍼에 저장되고, 해당 데이터의 저장 사실을 Qt의 객체 간 (inter-object) 통신 방식인 시그널-슬롯 메커니즘을 통해 UI 모듈에 알림으로써 해당 데이터를 파일에 쓰고 그래프에 표현하도록 한다.

$$\text{Obj\_LongDispl} = ((\text{frame.data}[1] \ll 3) | (\text{frame.data}[2] \gg 5)) * .1f;$$

그림 9. 고정 소수점 데이터의 부동 소수점 데이터로의 변환  
Fig. 9. Conversion of arbitrary fixed point number into floating point number

### 3.2.2 메인 모듈

데이터 수집 모듈이 얻은 데이터는 데이터 유실의 방지를 위한 버퍼에 저장된다. 메인 모듈은 버퍼의 데이터를 데이터 수집 모듈이 시그널을 보낼 때마다 CSV(comma-separated values) 형식의 파일에 쓰고, 그래프를 그리는데 필요한 데이터를 골라 3)의 그래프 모듈에 시그널을 보내 그래프를 그리도록 한다.

### 3.2.3 그래프 모듈

메인 모듈이 보내온 객체들의 정보를 바탕으로 그래프를 그린다. 움직이는 물체 중 레이더로부터 멀어지는 것과 가까워지는 것은 색을 달리하여 그리고, 물체의 크기에 따라 물체를 표현하는 사각형의 크기를 달리한다. 그래프를 그리는 데 오버헤드가 크므로 일정 시간 간격을 두고 그래프 영역을 업데이트한다.

그림 10은 3)의 그래프 모듈이 그린 그래프의 모습이다. ARS308 레이더 센서는 일정 속도 미만으로 움직이는 객체는 움직이지 않는 것으로 간주하고, 그 외의 물체는 레이더를 향해 다가오는 객체와 레이더에서 멀어지는 객체로 분류한다. 그에 따라 본 그래프 모듈은 레이더를 향해 다가오는 객체는 황색 사각형으로, 반대로 멀어지는 객체는 녹색 사각형으로 나타내며 정지한 객체는 파란색으로 표시한다. 각 사각형의 왼쪽 상단에 위치한 숫자는 해당 객체에 레이더 센서가 부여한 객체 ID를 나타내며, 오른쪽 상단의 숫자는 레이더 센서로부터의 거리를 나타낸다. 그림 10에서, 녹색의 작은 사각형의 오른쪽 상단에 “human”이라고 표시된 것을 볼 수 있는데, 이는 레이더 센서가 해당 객체의 폭에 기반해 객체를 사람으로 판단한다는 것을 알려준다. 그래프의 세로와 가로 눈금마다 표시된 숫자는 각각 레이더로부터의 세로방향 거리와

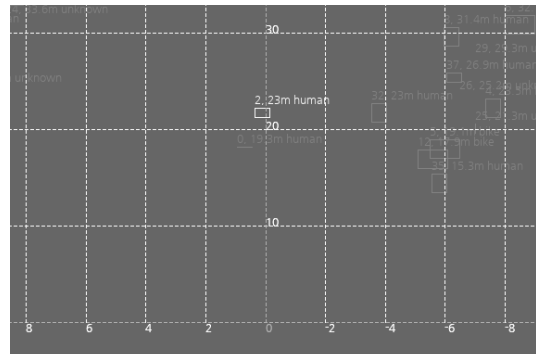


그림 10. 레이더 센서로부터 멀리 떨어진 오브젝트 움직임을 보여주는 그래프 모듈의 출력 결과  
Fig.10. An output of the graphing module showing an object moving farther away from the radar sensor

가로방향 거리를 미터 단위로 나타낸다.

### 3.2.4 어플리케이션 인터페이스

그림 11이 나타내는 데이터 수집 어플리케이션의 그래픽 사용자 인터페이스는 크게 세 가지 부분으로 구성된다. 화면 좌측에는 레이더로부터 전송된 40쌍의 객체 데이터들이 표시되고, 우측에는 40쌍의 객체 각각의 위치가 시각화되어 표시된다. 화면의 우측에는 3)에서 설명한 그래프 모듈이 버퍼에 수집된 CAN 프레임들을 100ms마다 검사해 객체들의 위치를 표시한다. 화면 좌하단의 슬라이더를 이용해 그래프의 스케일과 눈금 간격을 조절 가능하고 수집된 데이터를 파일 메뉴를 이용 저장할 수 있다.

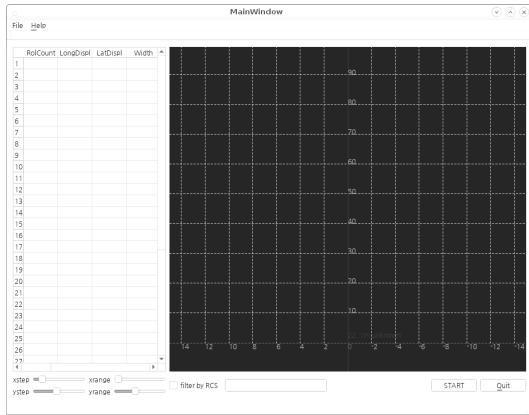


그림 11. 데이터 수집 어플리케이션의 그래픽 유저인터페이스  
Fig. 11. Graphic user interface of the data acquisition application

## IV. 클러스터링 알고리즘

### 4.1 이동체의 RCS 값과 속도 값 및 좌표 값을 이용한 클러스터링 알고리즘

다수의 실험 데이터를 분석을 통하여 이동하는 한 오브젝트가 RCS 값에 의해 분산이 된다고 하더라도 한 이동체일 경우 같은 시점에서의 거리 값이나 속도 값이 일정 이하의 차이를 보인다는 점을 특징을 이용하여 다음과 같은 수식을 통하여 분산된 이동물체를 동일 이동 물체로 클러스터링 하는 것이 가능하다.

- D : Distance(m)
- S : Speed(m/s)
- $d_{ij}$  : i 번째 object의 j번째 거리 관측치
- $s_{ij}$  : i 번째 object의 j번째 속도 관측치

$$prop(x) = [d_{ij} - d_{mn}] \leq 2]_{\{i < m, |j-n| \leq 20\}} \cap [s_{ij} - s_{mn}] \leq 0.4]_{\{i < m, |j-n| \leq 20\}} \quad (1)$$

(1)의 수식을 사용하여 특정 위치에서의 튀는 RCS 값 대신 20ms 시간 내 ID가 다른 데이터 중 이 둘의 거리가 2m 이내이고, 속도의 차이가 0.4m/s이내인 것들을 하나의 물체로 인식하는 것으로 처리하였다. 이를 통해 데이터를 필터링하여 실험 결과에서 그림 12와 같이 특정 범위에서 오차가 생기는 부분의 수정하여 그림 13과 같은 결과를 얻었다.

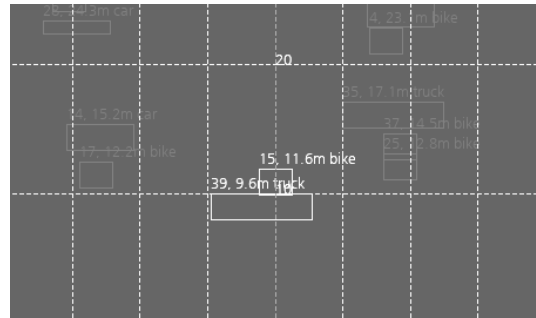


그림 12. 한 개의 오브젝트가 2개의 오브젝트로 분산  
Fig. 12. Diffusing one object to two object at 10m far from radar

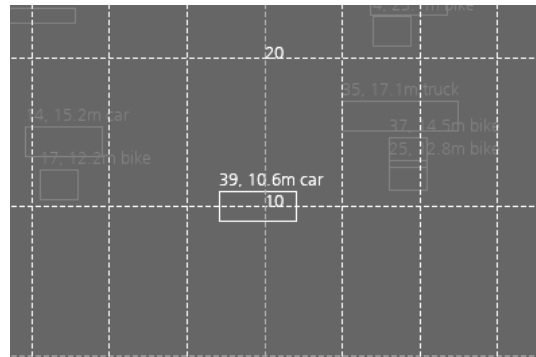


그림 13. 클러스터링 알고리즘을 이용한 오브젝트 합치기  
Fig. 13. Merging object using clustering algorithm

## V. 실험결과 분석

### 5.1 실험구성

본 실험의 구성은 다음과 같다. 우선 실험을 위해 그림 14와 같은 실험 환경을 구축하였다. 레이더와 카메라를 통한 시각적 분석을 위해 두 장비를 함께 사용할 수 있는 기구부를 제작하였으며 이것을 삼각대에 고정함으로써 높이 조절을 통해 다양한 실험 결과의

획득을 피했다. 그리고 레이더의 반사판의 기울기 값을 조정해가며 자동차와 사람들에 대한 다양한 실험을 진행하고 그에 따른 RCS(Radar Cross Section)값과 거리, 속도 등의 데이터에 대해서 분석하고, 각 실험 환경에서의 차이점에 대해 분석하였다. 아래의 실험결론을 내리는 데이터로는 같은 환경에서의 단일 차량의 진입에 대한 분석을 한 결과이다.

### 5.2 실험환경

본 실험의 환경은 다음의 그림 14에서 보여주는 바와 같다. ①은 실제 피사체 촬영을 위한 Camera, ②는 ARS308 Radar, ③은 CAN Breakout Box를 이용한 데이터 전송, ④의 휴대용 전원 공급 장치로 환경을 이루어서 실험을 진행하였다.

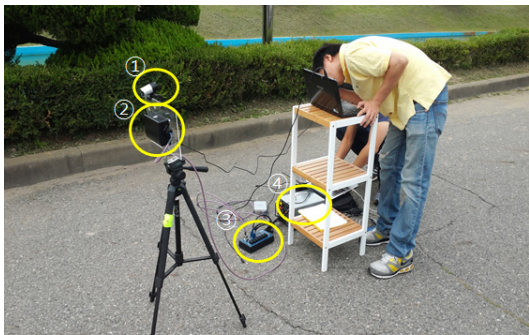


그림 14. 실험 환경  
Fig. 14. Experiment Environment

그림 15는 자동차와 사람 오브젝트를 같이 측정하고 있는 실험이고 그림 16은 야간에 레이더를 이용한 단일 차량 검증을 진행하며 반사판의 기울기 값을 조절하였다.



그림 15. 자동차와 사람 오브젝트 측정  
Fig. 15. Detect car and human objects



그림 16. 반사판의 기울기 값 조절  
Fig. 16. radar sensor calibrations

### 5.3 실험결과분석

실험의 타당성을 위해 동일한 조건에서 레이더반사판의 기울기 값을 교정해가며 사물의 인식을 시도하였다. 그 결과 제대로 된 조절이 이루어지지 않으면 한 오브젝트가 RCS 값의 오차로 인하여 여러 개의 오브젝트로 분산되어 인식하게 된다는 것을 도출해 내었다.<sup>[7]</sup>

또한 영점조절이 잘 이루어지더라도 실험 환경에 따라 특정 구간에서 RCS 값이 오차가 생기는 구간이 있으므로 실제 이동체의 검지를 위해서는 이를 보정할 효율적 알고리즘이 필수적이다.

데이터의 분석을 위해서 취득한 레이더 데이터의 시간에 따른 144개의 데이터를 샘플링 하여 RCS 값의 오차를 표시했다. 우선 레이더 반사판의 기울기 값에 따른 오차가 생겼을 경우의 RCS 값과 실험 데이터의 가지적인 좌표를 아래에서 표현했다.

레이더 반사판의 기울기 값은 0~32도 까지 가변되며 22도로 조절한 경우 하나의 물체임에도 불구하고 그림 17과 같이 RCS 값이 분산되어 오차가 생김에 따라 73m떨어진 거리부터 레이더 도착까지 하나의

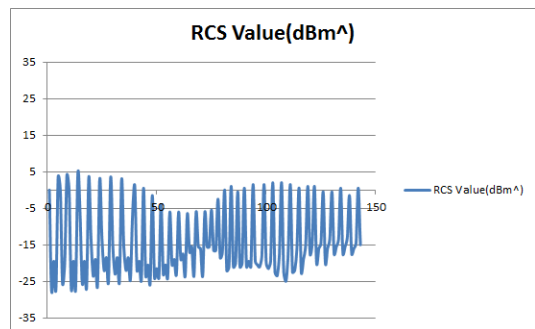


그림 17. 반사판이 22도일 경우 RCS 값의 노이즈  
Fig. 17. Too many noise at RCS value of 22 degree (elevation)



이동체가 그림 18부터 그림 19까지처럼 여러 개의 이동체로 나뉘어 들어오게 된다.

레이더 반사판의 기울기 값이 16도일 경우는 22도의 경우보다 그림 20과 같이 RCS 값의 노이즈가 조금 더 좋은 결과를 보여주지만, 40m의 거리부터 레이더 도착까지 그림 21과 22처럼 여러 개의 이동체로 인식하게 된다.

레이더반사판의 기울기 값을 다양하게 하여 테스트를 했을 때 그림 23과 같이 19도 근방에서 가장 좋은

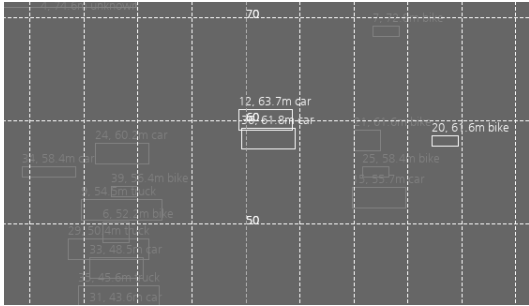


그림 18. 60m 떨어진 지점에서의 오브젝트의 분산  
Fig. 18. Diffusing one object to two object at 60m far from radar

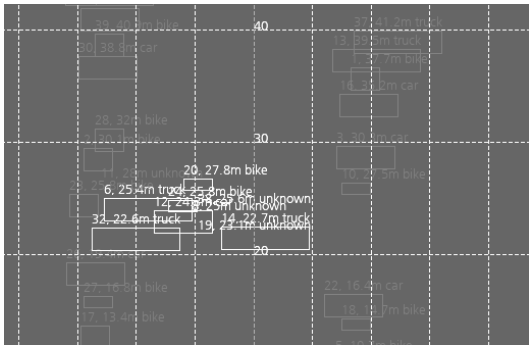


그림 19. 24m 떨어진 지점에서 1개 오브젝트가 6개로 분산  
Fig. 19. Diffusing one object to six object at 24m far from radar

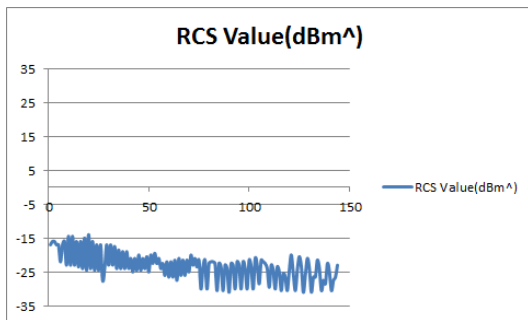


그림 20. 반사판이 16도일 경우 RCS 값의 노이즈  
Fig. 20. Many noise at RCS value of 16degree(elevation)

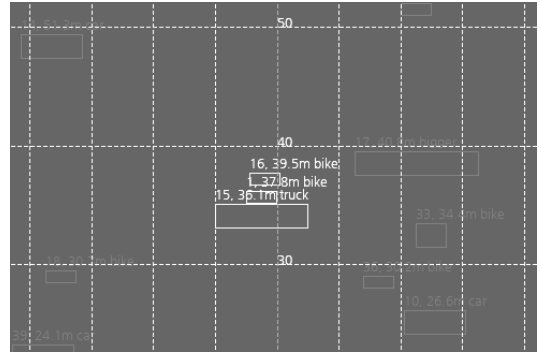


그림 21. 38m 떨어진 지점에서 1개 오브젝트가 3개로 분산  
Fig. 21. Diffusing one object to three object at 38m far from radar

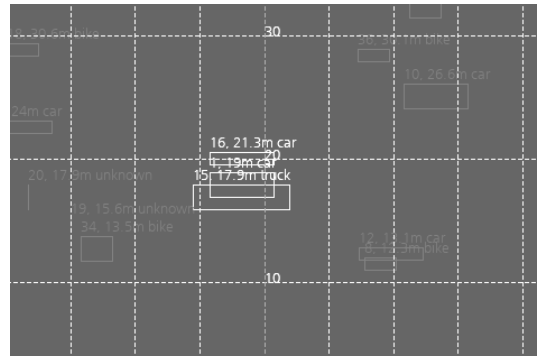


그림 22. 20m 떨어진 지점에서 1개 오브젝트가 3개로 분산  
Fig. 22. Diffusing one object to three object at 20m far from radar

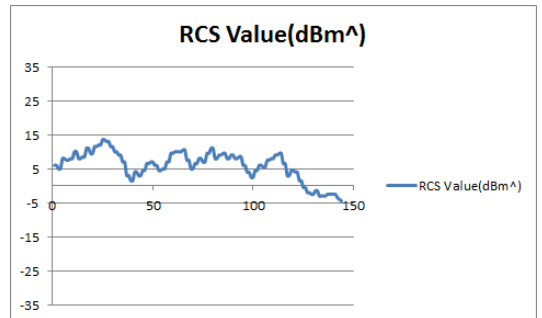


그림 23. 반사판이 19도일 경우 양호한 RCS 값의 웨이브폼  
Fig. 23. good wave form at RCS value of 19degree (elevation)

RCS 값 결과를 보여주었는데 이 경우 10m 정도까지는 그림 24와 25처럼 한 개의 오브젝트로 인식한다. 하지만 레이더 인접 10m까지는 하나의 오브젝트로 인식하지만, 10m 이내에서는 그림 26와 같은 RCS 값 노이즈 때문에 그림 27와 같이 1~2개 정도의 이동체

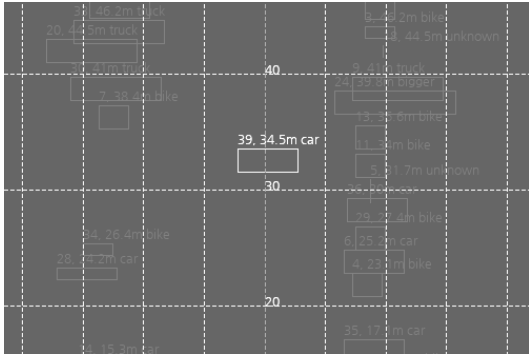


그림 24. 34.5m 지점에서의 양호한 결과(반사판 19도)  
Fig. 24. good result at 34.5m far from radar(19degree elevation)

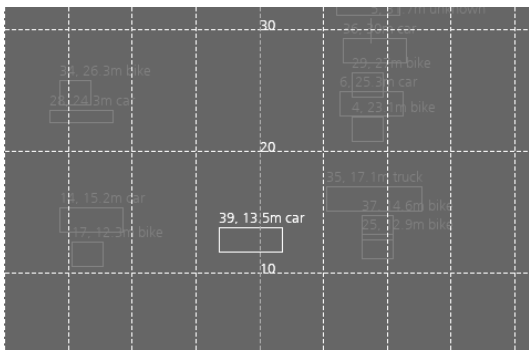


그림 25. 13.5m 지점에서의 양호한 결과(반사판 19도)  
Fig. 25. good result at 34.5m far from radar(19degree elevation)

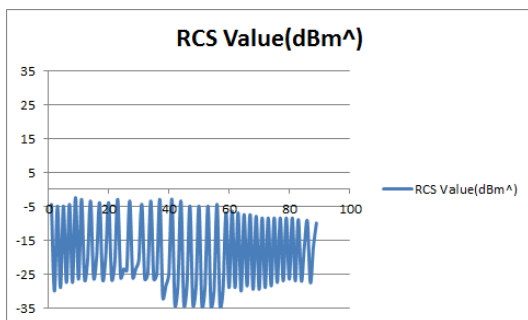


그림 26. 반사판이 19도이고 10m 이내에서의 약간의 노이즈  
Fig. 26. some noise at RCS value of 19degree(elevation) & 10m far from radar

로 인식되는 결과를 얻었다.

이 분산되는 10m 근방 지점의 RCS 값과 거리, 속도 값을 분석하여 IV의 알고리즘을 고안하게 되었다. 표 5는 분석한 값의 일부분을 표시한 것이다. 즉 실험 데이터 분석을 통해 입력되는 레이더의 신호 값 중 RCS 값이 분산됨에 따라서 한 물체도 여러 개의 물체

로 인식되며 이 때문에 하나의 이동체가 여러 개의 이동체로 인식되는 오차가 발생하게 되었다. 이를 해결하기 위해 IV에서 주장한 클러스터링 알고리즘을 분산이 일어난 그림 27의 지점에 적용한 결과 그림 28과 같은 1개의 오브젝트로 보정할 수 있었다.

그리고 이 클러스터링 알고리즘을 적용하여 사람을 대상으로 실험한 결과 만족할 만한 결과를 얻을 수 있었다. 그림 29은 한명의 사람을 감지하는 실험을 진행한 영상 캡처로 그림 30의 결과를 얻었고, 그림 31은 두 명의 사람을 감지하는 실험으로 그림 32의 결과를 얻어, 사람을 감지하는 경우에도 제대로 감지 할 수 있음을 증명했다.

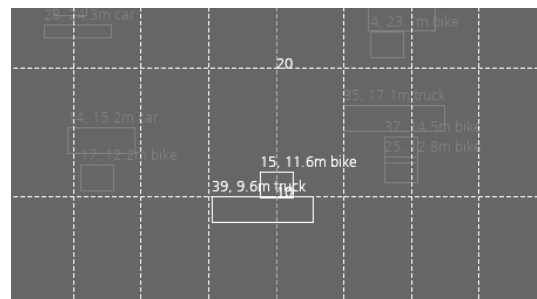


그림 27. 10m 지점에서 1개 오브젝트가 2개로 분산(19도)  
Fig. 27. Diffusing one object to two object at 10m far from radar(19degree)

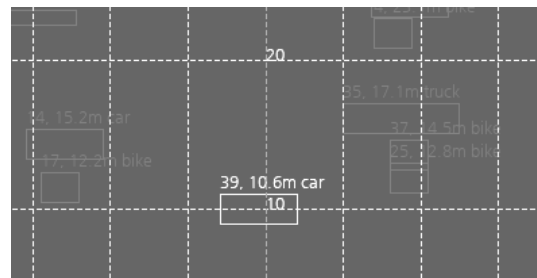


그림 28. 클러스터링 알고리즘을 적용한 오브젝트 합성  
Fig. 28. Merging object using clustering algorithm

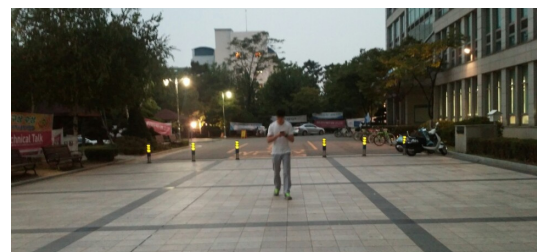


그림 29. 한명의 사람 감지  
Fig. 29. Detecting one human object

표 5. 10m 지점에서 데이터의 특징  
Table. 5. Characteristic of data in 10m far from radar.

ID	Distance	Speed	RCS	ID	Distance	Speed	RCS
15	11	-3.8125	-27.5	15	8.1	-2.125	-29
39	9.1	-3.5	-4.5	39	6.5	-1.875	-5
15	10.7	-3.625	-27.5	15	8	-2.125	-30
39	8.9	-3.3125	-2.5	39	6.4	-1.875	-4.5
15	10.4	-3.4375	-26.5	15	7.8	-2.0625	-29.5
39	8.6	-3.25	-3	39	6.2	-1.8125	-7
15	10.3	-3.25	-26.5	15	7.7	-2	-29
39	8.4	-3.0625	-3.5	39	6.1	-1.75	-6.5
15	10.1	-3.125	-26	15	7.6	-1.9375	-29
39	8.2	-2.9375	-4	39	6	-1.6875	-7
15	9.9	-3	-26.5	15	7.4	-1.9375	-30
39	8.1	-2.75	-4	39	5.9	-1.625	-7.5
15	9.6	-2.9375	-26.5	15	7.3	-1.8125	-28.5
39	7.9	-2.625	-3	39	5.8	-1.5625	-7.5

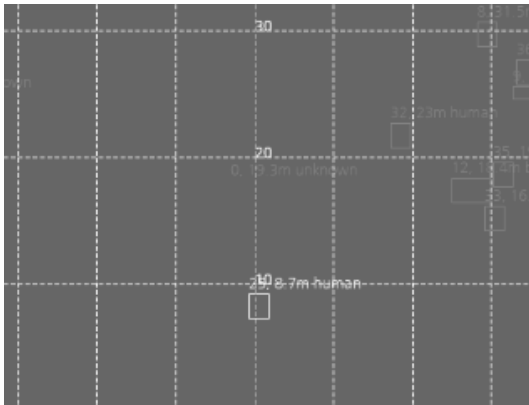


그림 30. 그림 29의 결과  
Fig. 30. Result of Fig. 29.

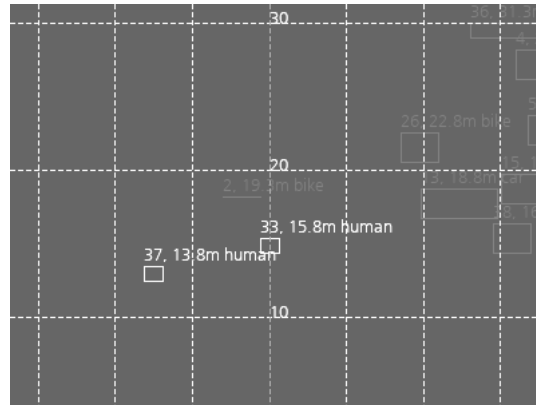


그림 32. 그림 31의 결과  
Fig. 32. Result of Fig. 31.



그림 31. 두 명의 사람 감지  
Fig. 31. Detecting two human object

## VI. 결 론

본 논문에서는 근거리 레이더의 특성 분석과 실험을 통해 레이더의 문제점을 수정 보완하는 새로운 알고리즘을 제안하였다. 근거리 레이더가 사용하기에 편리하고 가격이 비교적 저렴하다는 장점이 있지만 근거리 레이더를 사용하기 위해서는 레이더의 특성에 따른 오차 등을 보완해 줄 수 있는 방법이 필요하며 본 연구에서는 이를 위해 실험 데이터의 분석을 통하여 얻어낸 레이더의 특성을 이용해 이동 물체의 효율적인 감지가 가능한 시스템을 구현할 수 있었다. 현재 제안한 시스템은 일부 수정을 통하여 선박 내에서 사람의 이동경로나 물체의 움직임 등을 감시함으로써 선박에 타고 있는 사람들의 안전을 확보할 수 있는 안전감지 시스템으로 활용 가능하다. 추후 다른 방식으

로 연구를 발전시킨다면 선박의 외부를 감시하는 시스템으로 활용하여 입항 시 선착장까지의 거리 감시나, 항해 중에 장애물을 감지하는 시스템으로 발전이 가능할 것으로 생각된다.

References

[1] I. Gresham, N. Jain, T. Budka, A. Alexanian, N. Kinayman, B. Ziegner, S. Brown, and P. Staecker, "A compact manufacturable 76 - 77-GHz radar module for commercial ACC applications," *IEEE Trans. Microwave Theory and Techniques*, vol. 49, no. 1, pp. 44-54, Jan. 2001.

[2] S. Shin and G. Jun, "A study of transceiver system for ka-band road watch radar," *J. KICS*, vol. 36, no. 11, 2011.

[3] E. F. Knott, J. F. Shaeffer, and M. T. Tuley, *Radar Cross Section*, 2nd Ed., SciTech Publishing, 2004.

[4] S. Park, I. Kim, and D. Lee, "Implementation of IEEE1588 for clock synchronization," *J. KICS*, vol. 39B, no. 02, 2014.

[5] S. Corrigan, "Introduction to the controller area network (CAN)," Application Report, Texas Instruments, 2002.

[6] M. Kleine-Budde, "SocketCAN - the official CAN API of the linux kernel," in *Proc. ICC*, 2012.

[7] H.-B. Lee, S.-H. Kang, J.-H. Choi, and S.-C. Kim, "A study on the DOA estimation algorithms for automotive radar systems," in *Proc. KICS Int. Conf. Commun. 2012 (KICS ICC 2012)*, pp. 193-194, Yongpyong, Korea, Feb. 2012.

박 동 진 (Dong-Jin Park)



2014년 2월 : 인하대학교 컴퓨터공학과 졸업 (학사)  
 2014년 3월~현재 : 인하대학교 컴퓨터공학과 석사과정  
 <관심분야> HCI, Sensor Network, Embedded

유 인 환 (In-Hwan Ryu)



2009년 3월~현재 : 인하대학교 컴퓨터공학과 학사과정  
 <관심분야> 로보틱스, 프로그래밍 언어

변 기 훈 (Ki-Hoon Byun)



2008년 3월~현재 : 인하대학교 통계학과 학사과정  
 <관심분야> 로보틱스

이 상 민 (Sang-Min Lee)



1987년 2월 : 인하대학교 전자공학과 졸업 (학사)  
 1989년 2월 : 인하대학교 대학원 전자공학과 졸업 (석사)  
 2000년 2월 : 인하대학교 대학원 전자공학졸업 (박사)  
 1987년~1994년 : LG 이노텍 선임연구원

1995년~2002년 : 삼성전자 종합기술원 책임연구원  
 2002년~2004년 : 한양대학교 의과대학 의공학교실 연구교수  
 2005년~2006년 9월 : 전북대학교 생체정보공학부 조교수  
 2006년 9월~현재 : 인하대학교 전자공학과 교수  
 <관심분야> Bio-Signal Processing, Psycho-Acoustics, Sports Engineering, Brain-Machine Interface

권 장 우 (Jang-Woo Kwon)



1990년 2월 : 인하대학교 전자  
공학과 공학사

1992년 2월 : 인하대학교 전자  
공학과 공학석사

1996년 8월 : 인하대학교 전자  
공학과 공학박사

1996년~1998년 : 특허청 사무관

1998년~2009년 : 동명대학교 컴퓨터공학과 부교수

2006년~2012년 : 정보통신산업진흥원 인재양성단장

2012년~현재 : 인하대학교 컴퓨터공학과 교수

<관심분야> HCI, 신호처리, 지능형 시스템