

Person Tracking by Detection of Mobile Robot using RGB-D Cameras

Young-Ju Kim*

Abstract

In this paper, we have implemented a low-cost mobile robot supporting the person tracking by detection using RGB-D cameras and ROS(Robot Operating System) framework. The mobile robot was developed based on the Kobuki mobile base equipped with 2's Kinect devices and a high performance controller. One kinect device was used to detect and track the single person among people in the constrained working area by combining point cloud data filtering & clustering, HOG classifier and Kalman Filter-based estimation successively, and the other to perform the SLAM-based navigation supported in ROS framework. In performance evaluation, the person tracking by detection was proved to be robustly executed in real-time, and the navigation function showed the accuracy with the mean distance error being lower than 50mm. The mobile robot implemented has a significance in using the open-source based, general-purpose and low-cost approach.

▶ Keyword: Mobile Robot, ROS, RGB-D Camera, Person tracking by detection, SLAM-based navigation

I. Introduction

로봇 하드웨어 및 소프트웨어 관련 국내 기술은 더딘 발전 속도를 보이고 있으며, 특히 로봇 제어 소프트웨어 플랫폼에 대한 집중적인 연구가 요구되고 있다[1,2]. 최근에는 로봇과 IoT(Internet of Things) 기술의 융합으로 로봇의 활용 분야가 확대되고 있을 뿐만 아니라 생활을 보다 편리하게 하는 기술 개발로 이어지고 있다[3].

사람과 로봇 사이에 다양한 형태의 인터페이스 방식이 사용되고 있으며, 이들 중에서 로봇이 카메라 등의 입력장치를 통해 사람을 식별하고 사람의 위치 정보를 인지하는 영상 기반의 인터페이스가 가장 널리 사용되고 있다[1]. 최근에는 2차원의 RGB 컬러 정보뿐만 아니라 객체까지의 거리를 나타내는 깊이 정보까지 제공하는 RGB-D 카메라가 개발되어 로봇 분야까지 응용 영역을 확대하고 있다. 기존에 3차원 정보를 입력하는 장치로 고가의 3D 레이저 거리계(LRF:Laser Range Finder)나 스테레오 카메라 등이 사용되었으나, 비용이나 캘리브레이션 어려움의 기술 난이도 등으로 범용 로봇에 적용하기 어려웠다. 이에 RGB-D 카메라가 이들 장치를 대신함에 따라 다양한 응용이 개발되고 있다[4-7].

업무 보조용 모바일 로봇은 작업 공간 내에서 사용자를 식별하고 추적하는 기능을 기본적으로 수행하여야 한다. 이러한 기

능은 사용자를 식별하여 위치 정보를 산출하는 기능과 원하는 위치로 장애물을 회피하면서 빠르게 이동하는 자율 주행 기능을 요구한다. 이에 본 연구에서는 RGB-D 카메라에서 입력되는 컬러 정보 및 깊이 정보를 이용하여 사용자를 식별하고 위치 정보를 산출함과 동시에 주변 정보를 인지하여 효과적으로 사용자를 추적하는 저비용의 모바일 로봇을 구현하였다.

본 연구에서는 로봇 개발 플랫폼으로 현재 널리 사용되고 있는 오픈소스 기반의 ROS(Robot Operating System) 프레임워크를 사용하였다[8,9]. 우선, Linux 운영체제와 ROS 프레임워크를 탑재한 Kobuki 기반의 모바일 로봇을 제작하고 주요 입력 장치로 두 대의 키넥트 장치(kinect-v1과 kinect-v2)를 확장하여 장착하였다. Kinect-v2의 입력을 이용하여 작업 공간에 대한 지도를 제작한 다음에, kinect-v1의 컬러 정보와 깊이 정보로 이루어진 포인트 군집 데이터를 이용하여 사용자 클러스터를 추출하고 칼만 필터와 지도 지형 정보를 기반으로 사용자 위치를 예측하여 검색함으로써 위치 정보를 산출하였다. 그리고 AMCL(Adaptive Monte Carlo Localization) 기법[10,11]을 이용하여 kinect-v2의 입력 정보를 바탕으로 로봇의 현재

*First Author: Young-Ju Kim, Corresponding Author: Young-Ju Kim

*Young-Ju Kim (yjkim@silla.ac.kr), Division of Computer Software Engineering, Silla University

*Received: 2017. 11. 28, Revised: 2017. 12. 04, Accepted: 2017. 12. 14.

위치를 파악한 후에 사용자의 움직임 경로와 지도의 지형을 비교, 계산하여 최단 경로로 사용자를 추적하도록 하였다. 또한, 이동 중간에 장애물이 발생하는 경우에 이를 회피하여 추적 가능하도록 하였다.

본 논문은 2장에서 기존 연구에 대해 살펴보고, 3장에서 ROS 프레임워크를 기반으로 RGB-D 카메라를 이용한 자율 주행 기능 구현에 대해 제시한다. 4장에서는 RGB-D 카메라의 포인트 군집 데이터 입력을 활용한 사용자 추출 및 추적 기법을 제시하고, 5장에서 실험 결과를 살펴 본 후에 6장에서 결론을 맺는다.

II. Related Works

1. ROS(Robot Operating System)

본 연구는 Fig. 1에서 제시된 로봇 미들웨어 프레임워크 [8,9] 중에서 소프트웨어 개발자들이 로봇 어플리케이션을 쉽게 개발하도록 하드웨어 추상화, 저수준 기기 제어, 프로세스간의 메시징 및 패키지 관리 기능 등이 우수한 ROS(Robot Operating System)를 이용하여 사람 추적 및 자율 주행 기능을 가지는 모바일 로봇의 제어 시스템을 구현함으로써 범용 로봇에서도 적용 가능한 저비용의 사람 추적 기법을 제시한다.



Fig. 1. Open-source Robot Development Framework

ROS는 소프트웨어를 모듈 단위인 노드(node) 단위로 나누어 구성한다. 노드들은 토픽(topic)을 통해 발행/구독(publish/subscribe) 방식으로 데이터를 송수신하며, 서비스 혹은 함수 호출은 RPC 방식으로 이루어진다. 토픽은 노드 간에 주고받는 데이터 통신 채널로서 노드가 토픽에 데이터를 발행하면, 토픽을 구독하고 있는 모든 노드에 데이터가 전달되게 된다. 발행/구독 방식에 따라 토픽은 M:N으로 연결될 수 있다. 서비스 요청 노드는 서비스 제공 노드에 호출을 통해 요청 데이터를 보내면 서비스 제공 노드는 요청 입력을 받아 처리하여 그 결과

를 반환한다. 토픽과 다르게 요청/응답에 대해 1:1로 처리함으로써 요청 및 응답 데이터를 정의하여 쉽게 이용할 수 있다.

ROS에는 노드/토픽/서비스의 등록과 연결을 관리하는 ROS 마스터(master)가 있다. Fig. 2는 이러한 관계를 보여주는데, 발신자(talker)는 메시지 발행 노드이고 수신자(listener)는 메시지 구독 노드이다. 그리고 ROS 마스터의 네이밍 서비스를 통해 각 노드들은 쉽게 연결하여 데이터를 주고받을 수 있다. ROS는 TCP또는 UDP 통신을 통해 노드 간 데이터를 주고받기 때문에 같은 네트워크에서 동작하는 독립된 로봇 간에도 ROS 마스터를 통해 손쉽게 연결될 수 있다. ROS 마스터는 파라미터 서버(Parameter Server) 기능도 제공하는데, 파라미터 서버는 노드 동작을 위한 파라미터 값을 저장하고 실행시간에 노드를 설정할 수 있게 지원한다.

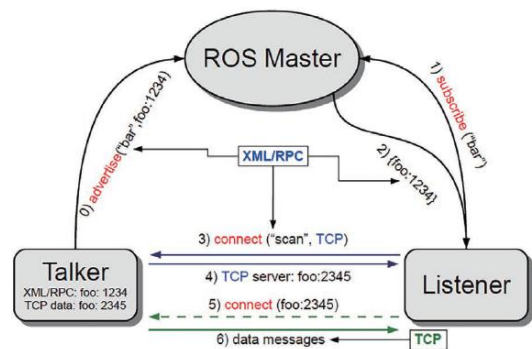


Fig. 2. Relations between ROS Master and Nodes

본 연구는 ROS 프레임워크를 탑재한 모바일 로봇을 제작하여 사람 추적 기능과 자율 주행 기능을 구현하였으며, 이를 구현한 전체 소프트웨어 구성은 Fig. 3과 같다.

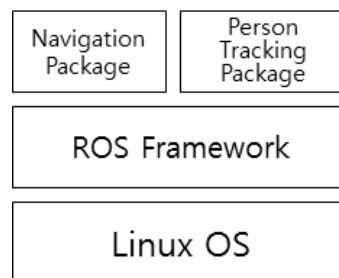


Fig. 3. Robot Software Structure

2. Person Detection using RGB-D Camera

현재 ROS 프레임워크에서 RGB-D 카메라의 컬러 및 깊이 정보를 이용하여 사람을 추출하고 인식 방법으로는 크게 다음의 세 가지 접근 방법으로 요약된다[13].

첫째, OpenNI/NITE SDK의 골격 추출 기반의 인식

둘째, OpenCV를 활용한 얼굴 추출 및 특징점 기반의 인식

셋째, PCL(Point Cloud Library) 기반의 사람 인식

위의 세 가지 방법에 대해 주요한 특징을 비교하면 Table 1

과 같다. OpenNI/NITE SDK의 골격 추출 기법은 xbox 등 게임기에서 사용자 움직임을 인식할 때에 사용하는 기법으로 사람 영역을 추출하는데 상당히 효과적인 방법이다. 그러나 다수의 사람들이 겹치는 경우에 추적중인 대상을 식별하기 어려운 문제가 있으며, 보다 근본적인 문제로 OpenNI/NITE SDK에 대한 공식적인 지원이 중단되어 이를 이용하여 응용 알고리즘을 개발하는 데는 한계점이 있다.

OpenCV를 이용한 추출 기법은 RGB 영상에서 얼굴의 위상 및 컬러 정보를 이용하여 얼굴 영역을 추출하여 인식하는 방법으로서 어느 정도 성능 면에서 검증된 방법이나, RGB-D 카메라의 깊이 정보를 활용하지 않을 뿐만 아니라 얼굴 영역만 추출하기 때문에 사람 전체 영역을 추출하기 위한 보완적인 알고리즘이 요구된다.

PCL 기반 추출 기법은 포인터 군집 데이터에 대해 관심 영역(ROI: Region Of Interest)의 특징을 이용하여 관심 영역을 추출하는 기법으로 다양한 특징 비교 알고리즘에 제시되어 있다. 이 기법은 추출 연산으로 관심 영역의 위상 정보를 한꺼번에 취득할 수 있는 장점이 있으나 포인터 군집 데이터의 대용량 특성으로 인해 계산부하가 근본적으로 높은 단점을 갖는다.

본 연구에서는 입력장치로 키넥트 장치를 사용하고 로봇 컨트롤러 장치로 성능이 높은 미니 PC를 사용한다는 점을 고려하여 정확도가 높은 PCL 기반 추출 기법을 바탕으로 사람 추출 및 추적 기법을 구현하였다.

Table 1. Comparison of Person Detection Methods using RGB-D Cameras

	OpenNI/NITE SDK-based Method	Face Extraction & Feature Point Matching Method	PCL-based Extraction Method
Outline	Extract human regions based on the skeleton information from input images and identify using ID info. of skeletons	Extract human regions based on facial areas and identify through feature point extraction and matching	Extract ROI using the topological information on point cloud data(PCD) and identify using the characteristic on ROI
Computational Load	Relatively Low	Middle level : computational load for face extraction is relatively high	Relatively High
Accuracy	The accuracy of skeleton extraction has been verified to some extent, but identification errors occur when people overlap.	Identification error occurs when background information is similar to face areas	Support background and person identification using topological information of point cloud data
Consideration	Formal support for the OpenNI/NITE SDK has been discontinued	Extracting facial region only leads to degradation of accuracy of human tracking	The algorithm complexity is very high, requiring high processing performance

III. Navigation using RGB-D Camera

1. Making of Mobile Robot Platform

본 연구에서 사용한 모바일 로봇은 기존의 kobuki 플랫폼 기반의 Turtlebot-2 로봇을 확장하여 제작하였으며, 로봇 시스템의 전체 구성과 실제 모양은 Fig. 4와 같다.

기존의 Turtlebot-2와의 차이점은 로봇 컨트롤러로 Intel NUC-i5 미니 PC를 로봇 상단에 장착하였으며, 입력 장치로 2 대의 키넥트 장치(kinect-v1과 kinect-v2)를 장착하였다. Kinect-v1 장치는 사람의 전신을 감지할 수 있도록 500mm 프 로파일 뼈대 위에 팬틸팅 장치를 부착하여 높게 장착하였고, kinect-v2 장치는 SLAM 기능을 위해 중간 트레이에 장착하였다. NUC 컴퓨터와 추가 키넥트 장치에 전원을 공급하기 위해 kobuki 베이스 장치에 장착된 배터리 외에 추가로 LIFE 배터리 팩을 제작하여 중간 트레이에 장착하였다.

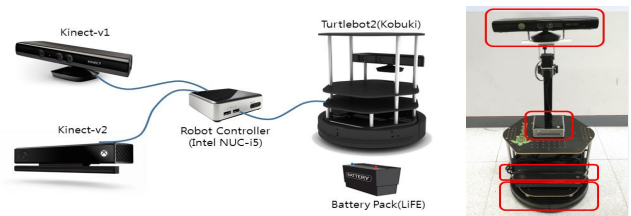


Fig. 4. Architecture and Real shape of Mobile Robot Platform

본 연구에서 제작한 모바일 로봇의 각 구성 요소별 역할과 특징을 살펴보면 다음과 같다.

Turtlebot-2 베이스 로봇은 2개의 바퀴로 움직이는 차동 움직임 방식을 채택하고 있으며, 베이스에 장착된 센서들을 통해 이동 중에 장애물 및 계단 등을 감지하는 기능을 제공한다. 내부적으로 배터리 팩을 내장하여 전원을 공급한다.

Kinect-v2는 최근 마이크로소프트사가 개발하여 판매하는 버전 2의 키넥트 장치로 kinect-v1에 비해 높은 해상도와 프레임율을 지원한다. 본 연구에서는 자율 주행(auto-navigation)을 위해 주로 사용하는 고가의 레이저 거리계(LRF)을 대신하여 kinect-v2를 사용하는데, 깊이 정보를 레이저 스캔 데이터로 변환하는 과정이 추가로 필요하지만 소형 모바일 로봇의 자율 주행을 위해서는 충분한 기능을 제공한다[6,7].

Kinect-v1은 마이크로소프트사가 xbox 게임기의 사용자 움직임 측정 장치로 개발한 것으로 가격 대비 성능이 높아 로봇 영역에서 다양하게 사용되는 장치이다[6]. 본 연구에서는 추적 대상인 사람을 검출하기 위한 입력 장치로 사용하며, 이를 위해서는 높은 해상도를 필요로 하지 않고 해상도가 높은 경우에 계산 부하가 증가함으로 상대적으로 해상도가 낮은 kinect-v1을 영상 입력장치로 채택하였다.

Intel NUC 컴퓨터는 주 로봇 컨트롤러 장치로 사용하며, 두 대의 키넥트 장치에서 입력되는 데이터를 충분히 처리할 수 있는 계산 성능을 제공하기 위해 Intel I-5 CPU와 8GB RAM를

장착하였고, 운영체제로는 Ubuntu-16.04를, ROS 프레임워크로는 ROS-kinetic 버전을 설치하여 구동하도록 하였다.

제작된 모바일 로봇의 전체 제어 과정은 Fig. 5과 같으며, 먼저 kinect-v2에서 입력되는 깊이 정보를 측정하여 작업 공간의 지도를 제작한다. Kinect-v1에서 입력되는 컬러 정보와 깊이 정보를 이용하여 사람 영역을 추출하고 위치 정보를 습득한다. 사람 추적 모듈 및 자율주행 모듈은 이 두 가지의 정보를 바탕으로 사람의 움직임 경로와 지도에 나타나는 지형을 비교, 계산하여 최단 경로를 계획하고 kobuki 제어 정보로 변환하여 전송함으로써 kobuki 베이스가 움직이도록 한다.

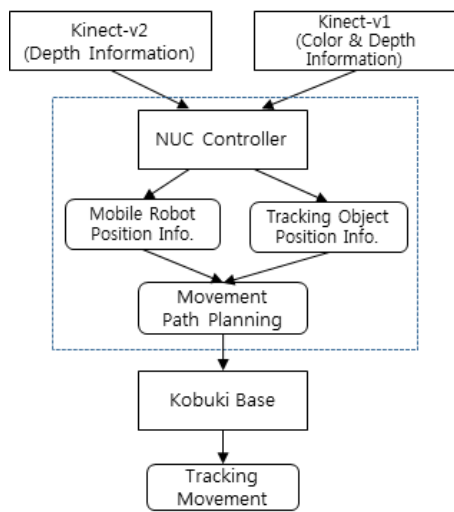


Fig. 5. Control Procedure of Mobile Robot

2. Navigation based on Kinect-v2

모바일 로봇의 사람 추적 기능을 위해서는 지도에서 원하는 위치와 방향을 지정하였을 때 지정된 위치로 이동하여 지정된 방향으로 자세를 잡는 기능과 주행 중에 장애물을 회피하는 기능을 지원하는 자율 주행(Navigation) 기능이 요구된다. 본 연구는 이를 위해 ROS 프레임워크가 지원하는 주요 기능 중에 하나인 SLAM(Synchronous Location And Mapping) 기반의 내비게이션 기능[10,11]을 사용하기로 하였다. 그리고 SLAM 동작을 위해 레이저 거리계(LRF) 장치가 필요한데 앞에서 언급한 것과 같이 kinect-v2를 대신 사용하기로 하였다.

본 연구에서 제작한 모바일 로봇을 기반으로 ROS 자율 주행 기능을 실행하기 위해 여러 단계의 포팅 작업이 필요로 하며, 각 단계별 포팅 작업 및 개선 내용을 살펴보면 다음과 같다.

2.1. Robot Modeling

ROS 프레임워크는 URDF(Uniform Robot Description Format) 기반의 로봇 모델링 데이터를 로봇 본체의 자동 항법(Odometry) 계산, 로봇 구성 요소의 움직임 및 센서 데이터의 좌표 변환 등을 위해 사용하므로 로봇 모델링은 ROS 프레임워크를 사용하기 위해서 제일 먼저 수행하여야 하는 작업이다. 본

연구에서 사용하는 모바일 로봇은 Turtlebot-2를 본체로 사용함으로 Turtlebot-2의 로봇 모델링 데이터를 사용한다. 그러나 구조적으로 키넥트 장치를 추가하였을 뿐만 아니라 2대의 키넥트 장치를 기존의 위치와 달리하여 장착함에 따라 수정이 요구되어 Turtlebot-2 로봇 모델링 데이터를 활용하여 본 연구에서 제작한 로봇에 맞게 로봇 모델링 데이터를 수정하였으며, Fig. 6은 Kinect-v2 추가에 따른 모델링 데이터이고, 이 부분을 적용한 부분 모델링 결과는 Fig. 7와 같다.

```

<?xml version="1.0"?>

<robot xmlns:xacro="http://www.ros.org/wiki/xacro"
name="sensor_kinect2">

  <xacro:include filename="$(find turtlebot_description)/urdf/sensors/kinect2/kinect2_common.xacro" />
  <xacro:include filename="$(find turtlebot_description)/urdf/sensors/kinect2/kinect2_adapter_iso50.urdf.xacro"/>
  <xacro:include filename="$(find turtlebot_description)/urdf/sensors/kinect2/kinect2_v0.urdf.xacro"/>

  <xacro:macro name="sensor_kinect2"
    params="prefix parent *origin">
    <!-- The following URDF has frame_in / frame_out, so it is placed at the parent -->
    <adapter_iso50_kinect2_xacro
      parent="$(parent)" prefix="$(prefix)">
      <insert_block name="origin" />
    </adapter_iso50_kinect2_xacro>

    <kinect2_v0 parent="$(prefix)_adapter_iso50_kinect2_base"
      prefix="$(prefix)">
      <origin xyz="0 0 0.0368" rpy="0 0 0" />
    </kinect2_v0>
  </xacro:macro>

</robot>
  
```

Fig. 6. Kinect-v2 URDF Modeling Data

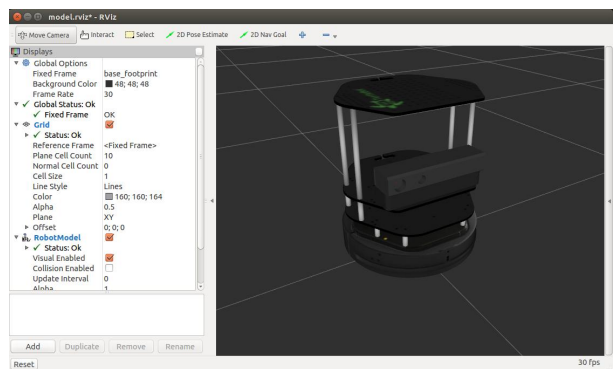


Fig. 7. Screen Output of Kinect-v2 Modeling

2.2. Kinect-v2 Device Driver Porting

ROS 프레임워크는 공식적으로 아직 kinect-v2 장치 드라이버를 지원하지 않는다. 이에 다른 커뮤니티에서 개발하여 제공하는 장치 드라이버를 사용하였다. 본 연구에서는 독일 Bremen 대학교의 Institute for Artificial Intelligence에서 제공하는 iai_kinect2 ROS 패키지를 사용한다.

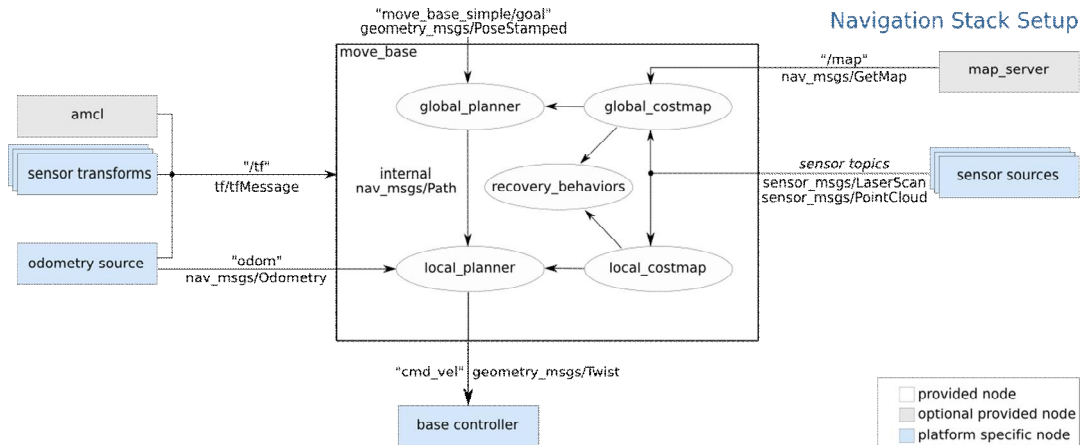


Fig. 8. ROS Navigation Stack Structure[9]

우선 OpenKinect 커뮤니티에서 제공하는 kinect-v2 드라이버 라이브러리 freenect2의 소스를 다운로드하여 컴파일 후에 설치한다. 그리고 github 사이트[22]에서 iai_kinect2 패키지 소스를 다운로드하여 컴파일 후에 설치한다. 그리고 kinect-v2 장치를 레이저 거리계 대신에 사용하기 위해서 ROS 프레임워크 위에 kinect2_bridge 패키지와 depthimage_to_laserscan 패키지를 실행시키는 런치(launch) 파일을 작성하여 필요할 때에 실행할 수 있도록 하였다.

2.3. Move_base Package Setup File Porting

ROS 프레임워크에서 SLAM 기반의 내비게이션 기능과 연관된 패키지는 크게 세 가지가 있다.

- A. move_base 패키지 - 지정된 위치로의 이동을 위한 경로 계획 및 장애물 회피 기능을 지원
- B. gmapping 패키지 - 지도 제작 기능을 지원
- C. amcl 패키지 - odometry 데이터와 레이저 스캔 데이터를 이용하여 로봇의 현재 위치 계산을 지원

위의 세 가지 패키지의 상호 작용은 Fig. 8을 통해 알 수 있는데, 우선 gmapping 패키지를 이용하여 주변 환경에 대한 지도를 오프라인 방식으로 제작하여 map_server 노드를 통해 지도 정보를 제공하도록 한다. 로봇 주행을 제어하는 move_base 노드는 이동 위치가 주어지면 먼저 global_costmap을 사용하여 현재 위치에서 원하는 위치까지의 전체 경로를 계획한다. 그런 후에 설정 매개변수를 기반으로 local_costmap을 사용하여 단기 이동 경로를 계산하여 (선 속도, 각 속도) 형식의 이동 명령을 산출하고 이를 base_controller 노드에 전송하여 로봇을 이동시키는데 이 과정을 목표 위치까지 도착할 때까지 반복 수행하게 된다. 이 과정에서 amcl 노드는 로봇의 현재 위치를 산출할 수 있는 좌표 변환값을 주기적으로 전송하며, 또한 이동하는 동안 입력되는 레이저 스캔 데이터를 바탕으로 지도에서 장애물 등에 대한 정보를 수정한다.

위에서 언급한 것과 같이 move_base 노드는 내비게이션 동작에서 로봇의 주행을 제어하는 동작을 수행하며, move_base 노드의 제어 동작을 결정하기 위해 크게 4개의 설정 파일을 사용한다. 이들 설정 파일은 costmap 공통 설정 매개변수, global_costmap

설정 매개변수, local_cstmap 설정 매개변수 그리고 local path planner를 위한 설정 매개변수 등으로 구성되며, move_base 노드의 세밀한 동작을 제어하기 위해서는 적절한 매개변수 값을 설정하여야 한다.

본 연구는 기본적으로 Turtlebot-2의 내비게이션 설정 파일을 사용하였으나 로봇의 구조적 변경으로 인해 로봇 무게가 변경됨에 따라 이동 오차가 비교적 크게 발생하였다. 이에 odometry 캘리브레이션 작업을 바탕으로 local path planner의 설정 매개변수 일부를 수정함으로써 이동 거리 오차가 50mm 범위에 들어오도록 하였다.

2.4. Map Generation using Gmapping Package

내비게이션 기능을 사용하기 위해서는 우선 동작 환경에 대한 지도를 제작하여야 한다. 이러한 지도 제작은 오프라인 방식으로 수행하는데 이는 지도 제작 시에 매개변수 값을 변경함으로써 다양한 형태의 지도 제작이 가능하도록 위함이다. gmapping 패키지를 위한 지도 제작 과정을 간단히 살펴보면 다음과 같다.

첫째, 키보드 또는 조이스틱 등을 이용한 원격 제어를 이용하여 로봇을 벽을 따라 이동시키면서 레이저 스캔 데이터를 rosbag 파일에 저장한다. 이동 중에 생성되는 레이저 스캔 데이터를 살펴보면 Fig. 9와 같다.



Fig. 9. Example of Laser Scan Data

둘째, 저장된 rosbag 파일과 map_server 노드를 이용하여 원하는 지도를 생성한다. 생성된 지도를 로딩한 후에 rviz 도구를 살펴보면 Fig. 10과 같다.

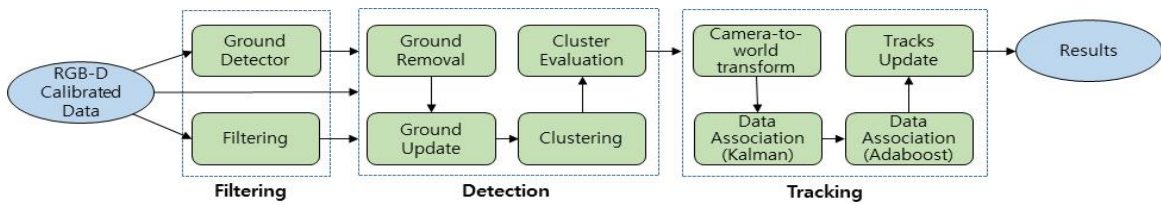


Fig. 11. Proposed System Structure in Ref. [15]

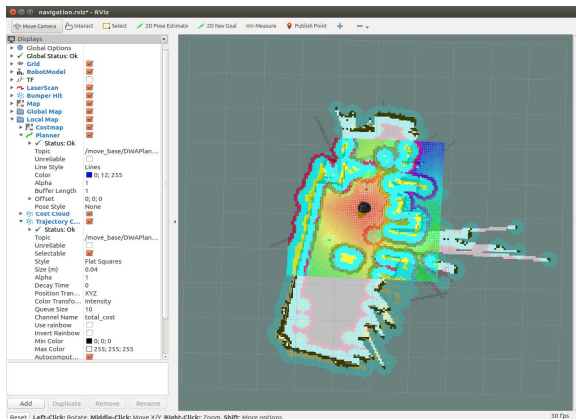


Fig. 10. Map Generation using Gmapping Package in ROS Framework

2.5. Navigation using AMCL Package

ROS 프레임워크의 SLAM 기반 내비게이션 동작은 “move_base_simple/goal” 토픽에 원하는 위치를 메시지로 전송하거나 rviz 화면에서 원하는 위치를 클릭하여 선택하면 로봇은 앞에서 설명한 것과 같이 지정된 위치로 장애물을 피하면서 이동한다. 이동 중에 amcl 노드는 일정한 주기로 이동 가능한 방향들을 제시하며, local_costmap과 비교하여 가장 적은 비용의 경로를 선택하여 이동하게 된다.

IV. Person Tracking by Detection using RGB-D Camera

1. Person Detection using Kinect-v1 Camera

기존에 RGB 카메라 입력만을 이용하거나 3D 센서 입력만을 이용하여 사람을 추출하고 추적하는 연구는 로봇 분야를 포함하여 다양한 분야에서 이루어져 왔다[14,15]. 이들 연구들은 센서 입력으로부터 보행자를 검출하여 움직임을 추적하는 성능에는 좋은 결과를 보여주나 복잡한 배경 또는 장애물 겹침 등에 의해 성능이 저하될 뿐만 아니라 높은 계산 부하에 의해 실시간 처리가 어려운 단점이 있다. 이러한 문제점에 대해 다른 센서 입력과 결합하여 해결하려는 여러 연구들도 있었다[18].

최근 RGB-D 카메라의 발전은 신뢰성 있는 외형 정보와 깊이 정보를 함께 사용할 수 있게 함으로써 로봇의 능력을 확장하거나 강인하게 하는데 크게 기여하고 있다[15]. 본 연구도 RGB-D 카메라

라인 키넥트 장치를 입력 장치로 사용하여 사람을 추출, 추적하는 기능을 구현하고자 하며, 모바일 로봇 응용 관점에서 추적 대상을 놓치지 않아야 하는 강인성(robustness)와 센서 입력에 대해 적절한 시간 내에 반응하여야 하는 실시간성(real-time processing) 요건을 만족하도록 한다.

본 연구는 입력 장치 및 성능 요건 등에서 유사한 기존 연구 중에서 참고문헌 [15]에서 제안된 방법을 참조하여 사람 추출 및 추적 기능을 구현하였으며, 다음의 2가지 제약 조건을 반영하여 성능을 개선하고자 하였다.

첫째, 제한된 작업 공간을 가지며, 또한 이 공간에 대한 지도 정보를 가진다.

둘째, 여러 사람이 존재하는 환경에서 1인 사용자를 추적한다.

참고문헌 [15]에서 제안된 시스템은 Fig. 11과 같다. 참조 시스템은 RGB-D 카메라 입력에 대해 필터링(Filtering), 추출(Detection) 그리고 추적(Tracking) 단계를 거치면서 모바일 로봇의 움직임 정보를 산출하였다. 필터링 단계에서는 입력된 포인트 군집 데이터에 대해 PCL 라이브러리의 voxel grid filter를 이용하여 다운샘플링한 후에 보행자들의 군집 데이터를 연결하고 있는 바닥면을 추출한다.

추출 단계에서는 다운샘플링된 군집 데이터에서 추출된 바닥면 데이터를 제거하고 유클리디언 거리를 이용하여 이웃 포인트들을 모으는 클러스터링을 수행한다. 그리고 보행자 클러스터를 식별하기 위해 추출된 클러스터에 대해 특징값, 즉, 높이, 중심점, 거리, 겹침(occlusion) 여부, RGB 데이터(blob), HOG confidence[13] 등을 계산한다.

추적 단계에서는 모든 클러스터의 특징값에 대해 경험적으로 결정된 파라미터를 바탕으로 필터링하여 보행자 클러스터를 추출하고, 추출된 클러스터에 대해 카메라 좌표에서 작업 공간 좌표로 변환하여 속도를 산출한다. 칼만 필터를 적용하여 추출된 클러스터와 기존의 추적 데이터를 연결한다. 추적 데이터에 연결되지 않는 클러스터에 대해 추적이 중단된 데이터와의 매칭을 위해 새로운 온라인 분류기를 제안하여 적용하고 최종적으로 추적 데이터를 업데이트한다.

참고문헌 [15]에서 제안된 방법은 기존에 RGB 이미지를 이용한 보행자 추출에서 우수한 성능을 보여준 Histograms of Oriented Gradients(HOG) 특징 기반의 SVM(Support Vector Machine) 분류기[13]의 높은 검색 부하에 대해 깊이 정보를 이용한 클러스터링을 이용하여 검색 공간을 줄임으로써 실시간 처리가 가능하게 만들었으며, 칼만 필터와 Adaboost 기반의 온라인 분류기를 이용하여 강인한 추적 성능을 제공하였다.

본 연구는 작업 공간에서 사람들을 추출하기 위해 참고문헌 [15]에서 제안된 필터링 및 추출 단계를 적용하였으며, 이들 단계에서 수행되는 기본적인 연산들에 대해서는 PCL 라이브러리[20] 및 SVM 라이브러리[21]에서 API를 제공함에 따라 이를 이용하여 사람 추출 기능을 구현하였다. Fig. 12는 구현된 기능에 출력된 결과로서 추출된 클러스터와 깊이 정보를 화면에 보여주고 있다.



Fig. 12. People Detection Result by Proposed Method

2. Person Tracking using Position Estimation and Map Information

본 연구는 앞에서 언급한 제약 조건을 바탕으로 참고문헌 [15]에서의 추적 단계 과정을 단순화하였다. 우선, 전 단계에 산출된 클러스터에 대해 특징값을 이용한 필터링을 통해 후보 클러스터들을 추출하고, 추적 대상에 대해 칼만 필터로 예측된 속도를 이용하여 위치를 예측한 다음, 후보 클러스터들의 위치와 비교하여 추적 대상에 해당하는 클러스터를 선별한다. 그리고 선택된 클러스터에 대한 특징값을 이용하여 추적 대상 정보를 업데이트한다.

본 연구에서는 특정 사용자만을 추적함에 따라 선택되지 못한 클러스터에 대해 손실된 추적 정보와의 매칭 과정이 별도로 필요하지 않으며, 지도 정보를 활용함으로써 강인성을 높였다. 지도 정보는 벽 또는 장애물에 대한 정보를 제공하는데, 추적 대상에 대해 예측된 위치 정보가 벽 또는 장애물과 충돌이 있는 경우에 예측된 위치 정보에 대한 검색 범위 및 방향을 조정함으로써 추적 손실 가능성을 낮추었다. 그리고 선택되지 않은 클러스터 정보를 장애물로 반영함에 따라 사람들과의 충돌을 피할 수 있는 경로 계획이 가능하도록 하였다. Fig. 13는 본 연구에서 제안하여 구현한 단순화된 추적 단계의 처리 과정이다.

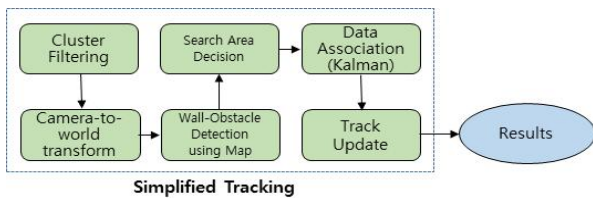


Fig. 13. Processing Procedure of Simplified Tracking

V. Performance Evaluation

1. Navigation Test of Mobile Robot

본 연구는 kobuki 기반의 모바일 로봇에서 SLAM 기반 내비게

이션 기능을 구현하였으며, 내비게이션 동작의 정확성을 테스트하기 위해 장애물을 있는 가상의 지도를 적용하여 실험을 수행하였다. 실험 결과는 Fig. 14와 같으며, 가로 1000mm, 세로 1000mm의 사각형 경로를 순회를 하는 실험 결과로서 정상적으로 장애물을 피하면서 수행하였다. 20회의 실험 결과에 대해 평균 이동 거리 오차는 약 50mm 이하였으며, 평균 방향 오차는 0.25 레디안(약 15도) 이하로 발생하였다. 방향 오차가 큰 편이지만 한번에 5M 이상의 긴 거리를 이동하는 동작보다는 사용자를 추적하는 짧은 움직임은 반복함으로써 움직임 때마다 자연스럽게 방향 오차를 보정함에 따라 추적 동작에서는 문제가 되지 않았다.

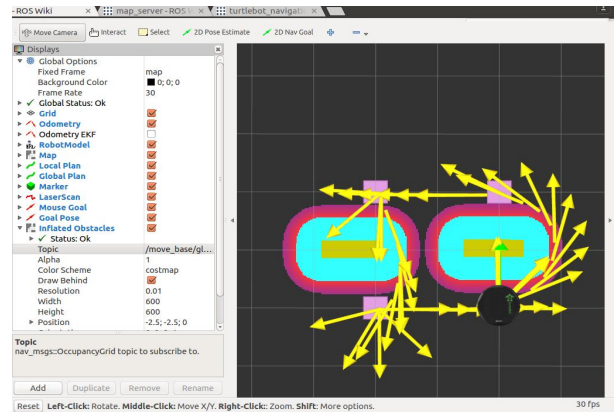


Fig. 14. Result of Travel Test using Virtual Map with Obstacles

2. Person Tracking Test using RGB-D Camera

사람 추출 및 추적 기능의 성능을 테스트하기 위해 ROS의 rosbag 기능을 이용하여 다음의 두 가지 시나리오에 맞추어 RGB-D 카메라 입력을 기록하여 추적 실험에 적용하였다.

첫째, 장애물이 없는 환경에서 선형적인 단순한 움직임

둘째, 장애물이 존재하는 환경에 비선형적인 복잡한 움직임

우선, 처리 속도 성능 지표로 프레임 처리율을 살펴보면 Table. 2와 같다. 캡처 프레임율은 키넥트 장치가 제공하는 프레임율과 동일하게 거의 초당 30 프레임에 제공하고 있으며, 사람 추출 및 추적 단계에 의해 처리되는 프레임율은 초당 25~26 프레임으로 실시간에 가깝게 처리되고 있음을 알 수 있다. 사람 추출 및 추적 기능만 실행한 처리율로서 추적을 위한 경로 계획 및 움직임 제어 등에 대한 시간이 반영되지 않았지만 움직임 제어 단계에서 지연이 크지 않은 점은 감안할 때 실시간 처리 요건을 만족한다고 할 수 있다.

Table 2. Frame Processing Rate of Proposed Method

Scenario	Number of frames	Capture Frame Rate(fps)	Tracking Processing Rate(fps)
First Scenario	8775	29.25	26.45
Second Scenario	8769	29.23	25.38

본 연구에서는 다수의 사람들을 추적하는 것이 아니라 특정



Fig. 15. Tracking Result of Frames of the First Scenario



Fig. 16. Tracking Result of Frames of the Second Scenario

사용자를 추적하는 기능을 지원함으로 캡처된 프레임별로 특정 사용자를 추출하여 위치 정보를 정확하게 추출하는 것을 추적 성공으로 정의하고, 추적 성공 여부를 산출하여 비율로 산정하면 Table. 3과 같다.

첫째 시나리오의 경우 93.6%의 높은 성공률을 보였으나 둘째 시나리오에서는 86.2%의 다소 낮은 성공률을 보였다. 장애물과의 겹침 또는 사람과 사람의 겹침이 발생하는 경우에서 선형적인(직선형) 움직임을 보이는 경우는 거의 성공하였으나 비선형적인 움직임을 보이는 경우에는 추적 대상을 잘못 찾는 경우가 발생하였다.

Table 3. Tracking Success Rate of Proposed Method

Scenario	Number of Success Frames	Tracing Success Rate(%)
First Scenario	8213	93.6
Second Scenario	7558	86.2

Fig. 15는 첫째 시나리오의 캡처 프레임에 대한 추적 실험 결과를 보여 주고 있는데, 두 명의 보행자가 거리를 두고 직선 형태의 보행을 보였을 때에 거의 두 명의 보행자를 모두 식별하였을 뿐만 아니라 두 보행자가 겹치거나 너무 가까운 거리에 있는 경우에도 성공적으로 추적대상 보행자(후행 보행자)를 추적할 수 있었다. Fig. 16은 둘째 시나리오의 캡처 프레임에 대한 추적 실험 결과를 보여주고 있다. 세 명의 보행자가 비선형적인 보행을 임의의 속도로 수행하는 경우를 가정하였으며, 중간 보행자를 추적대상 보행자로 설정하였다. 세 명의 보행자가 모두 겹치는 경우에 추적 손실이 발생하였으나(두 번째 프레임), 약간의 시간이 경과하고 추적대상 보행자가 뚜렷하게 추출되는 경우에 예측을 통해 추적을 회복함을 보여주고 있다(다섯 번째 프레임).

VI. Conclusions

업무 보조용 모바일 로봇은 작업 공간 내에서 사용자를 식별하여 추적하는 기능을 기본적으로 수행하여야 하는데, 이러한

기능은 사용자를 식별하여 위치 정보를 산출하는 기능과 원하는 위치로 장애물을 회피하면서 이동하는 자율 주행 기능을 요구한다. 본 연구에서는 RGB-D 카메라의 컬러 정보 및 깊이 정보를 이용하여 사용자를 식별하고 위치 정보를 산출함과 동시에 주변 지리 정보를 인지하여 효과적으로 사용자를 추적하는 저비용의 모바일 로봇을 구현하였다.

본 연구에서는 로봇 개발 프레임워크로 현재 가장 널리 사용되고 있는 오픈소스 기반의 ROS(Robot Operating System) 프레임워크를 사용하였으며, Linux 운영체제와 ROS 프레임워크를 탑재한 kobuki 기반의 모바일 로봇을 제작하고 입력장치로 두 대의 키넥트 장치(kinect-v1과 kinect-v2)를 확장하여 장착하였다.

우선 kinect-v2의 입력을 이용하여 작업 공간에 대한 지도를 제작한 다음에, kinect-v1의 컬러 정보와 깊이 정보로 이루어진 포인트 군집 데이터에 대해 필터링과 클러스터링을 수행하고 추출된 클러스터에 HOG 분류기를 적용하여 사람 영역을 추출함으로써 실시간성을 만족하도록 하였으며, 칼만 필터 기반으로 예측된 사용자의 움직임을 활용하여 사용자를 추출하고 위치 정보를 산출함으로써 강인성을 높였다. 그리고 kinect-v2의 입력 정보를 바탕으로 로봇의 현재 위치를 파악한 후에 사용자의 움직임 경로와 지도의 지형을 비교, 계산하여 최단 경로로 사용자를 추적하도록 구현하였다.

구현된 모바일 로봇에 대해 사람 추출 및 추적 기능과 자율 주행 기능에 대한 성능평가 결과, 사람 추출 및 추적 기능은 실시간성을 만족하면서 입력된 프레임에 대해 거의 93% 정도의 추적 성공률을 보였으며, 자율 주행 기능은 평균 거리 오차가 50mm 안으로 장애물을 회피하면서 이동하는 것을 확인하였다.

본 연구는 RGB-D 카메라 기반으로 실시간성과 강인성의 요건을 만족하면서 사용자를 추적하는 자율 주행 가능한 저비용의 모바일 로봇을 구현하였다는 것에 의의가 있으며, 향후 사람 추출 및 추적 기능에 대한 최적화 작업과 다수의 사람들에 대한 추적 기능 확장에 대해 연구를 진행할 예정이다.

REFERENCES

- [1] Siegwart, R., Nourbakhsh, I. R. and Scaramuzza, D., "Introduction to Autonomous Mobile Robots," 2nd Ed., The MIT Press, London, England, 2011.
- [2] Yong-Seon Moon, Sang-Hyun Roh, Seung-Woo Lim and Young-Chul Bae, "An Implementation of the Control System of the Mobile Robot using ROS," The Journal of the Korea Institute of Electronic Communication Sciences, Vol.8, No.11, pp. 1713-1718, Nov. 2013.
- [3] Lee Hyun Sun and Jung SI, "Implementation of Internet-based IoT Environment for Multiple Robot Systems," ICROS 2016, pp. 23-25, 2016.
- [4] Doopalam Tuvshinjargal, Deok Jin Lee. "Efficient Kinect Sensor-Based Reactive Path Planning Method for Autonomous Mobile Robots in Dynamic Environments." Transactions of the Korean Society of Mechanical Engineers - A, 39.6, pp. 549-559, June 2015.
- [5] T. Breuer, C. Bodensteiner, and M. Arens, "Low-cost Commodity Depth Sensor Comparison and Accuracy Analysis," in SPIE Security+Defence, vol. 9250. International Society for Optics and Photonics, 2014.
- [6] A. M. Pinto, P. Costa, A. P. Moreira, L. F. Rocha, G. Veiga, and E. Moreira, "Evaluation of Depth Sensors for Robotic Applications," IEEE International Conference on Autonomous Robot Systems and Competitions(ICARSC), pp. 388-394, 2015.
- [7] Kim Jun Sik, "RGB-D Camera Application Research Trend," Robot and Human, Vol.8, No.3, pp. 29-36, March 2011.
- [8] Pyo Youn Seak, "ROS Robot Programming," Ruby Paper Publisher, 2015.
- [9] ROS Web Site, <http://www.ros.org/>
- [10] Litman, T., "Autonomous Vehicle Implementation Predictions: Implications for Transport Planning," Traffic Technology International, Victoria, Canada, pp. 36-42, June 2014.
- [11] Van Den Berg, J. P. and Overmars, M. H., "Roadmap-based Motion Planning in Dynamic Environments," IEEE Trans. Robot, Vol. 21, No. 5, pp. 885-897, May 2005.
- [12] Simmons, R., "The Curvature-Velocity Method for Local Obstacle Avoidance," IEEE International Conference on Robotics and Automation, Vol. 4, pp. 3375-3382, 1996.
- [13] N. Dalal and B. Triggs., "Histograms of oriented gradients for human detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR 2005), Vol.1, pp. 886-893, July 2005.
- [14] N. Bellotto and H. Hu. "Computationally efficient solutions for tracking people with a mobile robot: an experimental evaluation of bayesian filters," Auton. Robots, Vol.28, pp. 425-438, Dec. 2009.
- [15] F. Basso, M. Munaro, S. Michieletto, E. Pagello, and E. Menegatti, "Fast and robust multi-people tracking from RGB-D data for a mobile robot," In Proceedings of the 12th International Conference IAS-12, pp. 265-276, 2012.
- [16] Linder T. and Arras K.O., "Multi-Model Hypothesis Tracking of Groups of People in RGB-D Data," IEEE Int. Conference on Information Fusion (FUSION'14), 2014.
- [17] Jafari O. Hosseini, Mitzel D. and Leibe B.. "Real-Time RGB-D based People Detection and Tracking for Mobile Robots and Head-Worn Cameras," IEEE International Conference on Robotics and Automation (ICRA'14), 2014.
- [18] Susperregi, Loreto, Martinez-Otzeta, Jose Maria, Ansuategi, Ander, Ibarguren, Aitor and Sierra, Basilio, "RGB-D, Laser and Thermal Sensor Fusion for People Following in a Mobile Robot," International Journal of Advanced Robotic Systems, Vol.10, pp. 5772-5780, Jan. 2013.
- [19] M. Munaro and E. Menegatti, "Fast RGB-D people tracking for service robots," Journal on Autonomous Robots, Springer, vol. 37, no. 3, pp. 227-242, March 2014.
- [20] PCL Web Site, <http://pointclouds.org/>
- [21] SVM Library Web Site, <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- [22] kinect2_bridge Package, https://github.com/code-iai/iai_kinect2

Authors



Young Ju Kim received the B.S., M.S. and Ph.D. degrees in Computer Science from Pusan University, Korea in 1990, 1995 and 1999, respectively. Dr. Kim joined the faculty of the Division of Computer Information Engineering at Silla University, Pusan, Korea in 2000. He is currently a professor in the Division of Computer Software Engineering at Silla University. He is interested in embedding computing, robot control and cloud computing.