

Vision-based Real-time Lane Detection and Tracking for Mobile Robots in a Constrained Track Environment

Young-Ju Kim*

*Professor, Division of Computer Software Engineering, Silla University, Busan, Korea

[Abstract]

As mobile robot applications increase in real life, the need of low cost autonomous driving are gradually increasing. We propose a novel vision-based real-time lane detection and tracking system that supports autonomous driving of mobile robots in constrained tracks which are designed considering indoor driving conditions of mobile robots. Considering the processing of lanes with various shapes and the pre-adjustment of operation parameters, the system structure with multi-operation modes are designed. In parameter tuning mode, thresholds of the color filter is dynamically adjusted based on the geometric property of the lane thickness. And in the unstable input mode of curved tracks and the stable input mode of straight tracks, lane feature pixels are adaptively extracted based on the geometric and temporal characteristics of the lanes and the lane model is fitted using the least-squared method. The track centerline is calculated using lane models and the motion model is simplified and tracked by a linear Kalman filter. In the driving experiments, it was confirmed that even in low-performance robot configurations, real-time processing produces the accurate autonomous driving in the constrained track.

▶ **Key words:** Autonomous Driving, Lane Detection, Lane Tracking, Kalman Filter, ROS

[요 약]

실생활에서의 모바일 로봇 응용이 증가하면서 저비용의 자율 주행 기능이 요구되고 있다. 본 논문은 모바일 로봇의 실내 주행 여건을 고려한 제한된 트랙을 가정하고, 제한된 트랙에서 모바일 로봇의 자율 주행을 지원하는 비전 기반 실시간 차선 검출 및 추적 시스템을 제안한다. 다양한 형태의 차선 처리와 동작 파라미터의 사전 조정 등을 고려하여 다중 동작 모드를 가진 시스템 구조와 상태 기계를 설계하였으며, 파라미터 조정 모드에서 차선 두께의 기하학적 특성을 바탕으로 컬러 필터의 임계값을 동적으로 조정하고, 곡선 트랙의 불안정 입력 모드와 직선 트랙의 안정 입력 모드에서 차선의 기하학적 그리고 시간적 특성을 바탕으로 차선 특징 픽셀을 적응적으로 추출하고 최소제곱법으로 차선 모형을 추정한다. 추정된 차선 모형으로 트랙 중앙선을 산출하고 움직임 모형을 단순화시켜 선형 칼만 필터를 통해 추적한다. 주행 실험에서 저성능의 로봇 구성에서도 실시간 처리를 통해 제한된 트랙에서 정상적으로 자율 주행이 이루어짐을 확인하였다.

▶ **주제어:** 자율 주행, 차선 검출, 차선 추적, 칼만 필터, ROS

-
- 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: 2019. 10. 25, Revised: 2019. 11. 25, Accepted: 2019. 11. 26.

I. Introduction

자율 주행 자동차에 대한 관심이 높아지면서 모바일 로봇의 자율 주행 기술에 대해 관심이 집중되고 있다. 그리고 주요 자율 주행 기술은 주로 GPS, Lidar, 레이저 그리고 카메라 등 고가의 센서 장치들의 데이터를 융합하여 주행 경로를 탐색하고 계획한다. 반면, 실생활에서의 모바일 로봇 활용이 증가하면서 저비용의 자율 주행 기술에 대한 요구가 점차 증대되고 있다[1-3]. 본 논문은 병원, 공장, 물류 센터 등과 같이 한정된 공간에서 서비스를 제공하는 모바일 로봇의 자율 주행 기술을 연구 대상으로 한다. 구체적으로 모바일 로봇의 실제 주행 여건을 반영한 제한된 트랙을 가정하고, 움직임 경로에 대한 사전 지식 없이 제한된 트랙을 인식하여 자율 주행하기 위해 카메라 센서의 영상 입력만을 이용하여 차선을 검출하고 추적하는 비전 기반 실시간 알고리즘을 제안하고 평가한다.

최근 차량의 ADAS(Advanced Driver Assistance Systems)에 대해 많은 연구가 수행되면서 다양한 차선 검출 및 추적 기법이 제시되고 있다[1]. 기존의 비전 기반 연구에서는 차선에 대해 하나의 차선 모형을 설정하고 이를 추정하는 알고리즘을 제안한 다음, 모든 입력 영상에 동일하게 적용한다[4-15]. 그러나 제한된 트랙에서는 입력 영상에서 좌우 차선이 직선과 곡선의 다양한 모양으로 나타나거나 급격한 회전 트랙에서는 한쪽 차선만이 나타나기도 한다. 그리고 계산 오버헤드가 적은 비전 기법은 조명 등의 환경 요인에 민감하게 성능이 좌우된다. 이에 본 논문은 차선의 기하학적 그리고 시간적 특성을 바탕으로 영상처리의 동작 파라미터를 동적으로 조정하여 안정성을 높이고 입력 영상의 차선 유형을 식별하여 적응적으로 차선을 검출하는 다중 동작 모드의 차선 검출 알고리즘을 제안한다. 또한 칼만 필터를 활용하여 모바일 로봇의 궤적에 해당하는 차선을 예측하고 추적함으로써 자율 주행의 정확도와 강인성을 높이는 기법을 제안한다. 제한된 트랙에 대한 실험 트랙을 구현하고 실제 주행 실험을 통해 제안된 알고리즘의 성능을 검증하였다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구에 대해 살펴보고, 3장에서는 먼저 본 논문에서 고려하는 자율 주행 시스템 구성 및 다중 동작 모드를 정의하고, 본 논문에서 제안하는 실시간 차선 검출 및 추적 알고리즘을 제시한다. 4장에서 제안된 알고리즘의 성능을 평가하고, 5장에서 결론과 향후 연구 방향을 제시하고 마무리한다.

II. Related Works

모바일 로봇의 주행 시스템이나 차량의 운전 보조 시스템 등에서 차선 검출 및 추적은 가장 기본적인 동작이며, Fig. 1과 같이 수행한다[1-3]. 특징 추출 단계에서 입력 영상으로부터 차선에 해당하는 특징점을 추출하는데, 특정 컬러의 특징점을 추출하거나 에지 특징점을 추출한다. 추출된 특징점에는 차선 이외의 요소도 포함하므로 차선의 기하학적 특성 등을 이용한 검색 기법을 통해 차선 특징점만을 선택한다. 차선에 대해 하나의 차선 모형을 설정하고 추출된 차선 특징점을 이용하여 차선 모형의 파라미터를 추정함으로써 차선을 검출하고 추적한다. 이번 장에서는 Fig. 1에서 제시된 처리 단계별로 관련 연구를 간단하게 살펴본다.

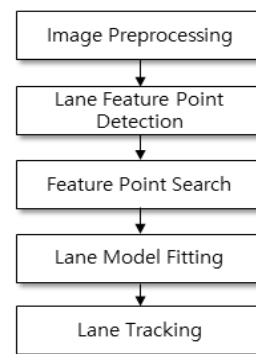


Fig. 1. General Lane Detection & Tracking Algorithm

차선 특징점 추출은 차선의 모양, 너비 그리고 컬러 등의 특징을 이용하여 특징점을 추출한다[1]. 컬러 기반 추출은 차선이 도로 노면과는 확연히 구별되는 컬러로 표시된다는 특징을 바탕으로 컬러 대비가 높은 HSV 컬러 공간에서 고정된 또는 적응적인 컬러 임계값을 이용하여 차선 컬러에 해당하는 픽셀만을 추출한다[2,4,5]. 이 기법은 계산 오버헤드가 적은 장점이 있으나 조명 변화에 민감하다는 단점을 가진다.

에지 기반 추출은 차선의 경계 부분에서 밝기 변화가 크다는 특징을 바탕으로 Canny 에지 검출[6] 등의 에지 검출 기법을 적용하여 특징점을 추출한다[7-9]. 에지 기반 추출은 영상에서 차선이 명확하게 표시되는 경우에는 잘 동작하지만 그림자나 흐릿한 차선 등의 잡음이 많은 환경에서는 에지 검출이 어렵다. 이에 에지 추출 성능 향상을 위해 특징점 추출 후에 에지 방향성을 분석하여 차선 경계 픽셀들을 필터링하거나[8] 단점을 보완하기 위해 여러 가지 제약 조건을 추가하여 에지 특징점을 추출하기도 한다[9]. 에지 기반 추출은 컬러 기반 추출보다는 조명 등 환경 변화에 덜 민감하다는 장점이 있으나 상대적으로 계산 오버헤드가 크고 에지 추출 시에 밝기 변화값(gradient)에

대한 임계값을 요구하는 단점도 있다. 본 논문은 실시간성 요건 등을 고려하여 기본적으로 컬러 기반 추출 방법을 적용하나, 컬러 필터의 동적인 임계값을 결정할 때에는 에지 기반 추출 방법으로 특징점을 추출한다.

특징점 검색 단계에서는 추출된 특징 픽셀 중에서 차선 이외의 요소에 해당하는 잡음 픽셀을 차선의 기하학적 특징 등을 이용하여 제거한다. 몇몇 기존 연구에서는 도로의 좌우 차선이 일정한 간격으로 평행하게 위치하고 일정한 두께를 가진다는 특징을 바탕으로 역투영 변환(IPM, Inverse Perspective Mapping)으로 카메라의 원근 투영 효과를 제거하여 평행 차선 영상을 구하고 차선의 위치를 검출하여 차선 픽셀을 검색하였다[2,3,9-15]. 검색 방법으로 차선의 좌우 에지 경계선 정보를 이용하여 검색하거나[14], 차선의 중앙선을 예측하고 이를 기준으로 수평으로 일정한 구간 내의 픽셀을 검색하는 방법[15] 등이 있다. 다른 방법으로는 전체 차선의 모양 정보로서 소실점을 이용한 검색 방법을 제안되었다[16,17]. 소실점을 기준으로 관심영역을 설정하고 차선 에지 선분의 기울기 각도 등을 이용하여 차선 특징점을 검색하였다. 본 논문은 제한된 트랙에서의 주행을 전제로 하는데, 입력 영상에서 소실점이 존재하지 않는 경우도 발생하여 역투영 변환을 바탕으로 개선된 검색 기법을 제안한다.

대부분의 이전 연구에서는 차선 검출의 강인성을 높이기 위해 차선 모형을 직선 또는 곡선으로 설정하고 다양한 방법으로 모형 파라미터를 추정하여 사용하였다[1]. 직선 모형 추정에서는 주로 호프 변환(Hough Transform)을 적용하여 차선 경계 선분 요소를 추출하고 이를 직선 모형에 피팅(fitting)시켰다[6-8,13]. 곡선 모형에 대해서는 전처리 필터를 통해 추출된 차선 픽셀만을 이용하여 단순하게 최소제곱법(least-square)을 이용해서 차선을 검출하거나[18] 잡음이 포함된 픽셀 집합에 대해 강건한 차선 검출을 위해 최소제곱법과 RANSAC 기법을 결합하여 적용하기도 하였다[10-12]. 또한 파티클 필터(Particle Filter)를 이용해서 차선을 추정 하는 방법들이 제안되었는데[19,20], 파티클 필터는 트래킹의 기능이 내재되어 있어 별도의 차선 추적 기법을 적용하지 않는다. RANSAC 추정이나 파티클 필터는 잡음을 내포한 특징점 집합으로부터 신뢰도 높은 모형 파라미터를 추정할 수 있다는 장점이 있으나 계산 오버헤드가 매우 높다.

차선 추적은 입력 영상간의 시간적 연속성을 바탕으로 차선 검출의 계산 오버헤드를 줄이고 검출 성능을 향상시키기 위해 적용한다. 차선 추적 방법으로 칼만 필터[2,4,13]와 파티클 필터[19,20]가 주로 사용되며, 이들 추적 방법은 연속된 입력 영상 사이에 차선의 위치 변화가 크지 않다는 점을 전제로 하고 있어 급격한 곡선 차선이

많은 경우에 오류가 발생할 수 있다.

본 논문은 차선 모형을 2차원 곡선으로 설정하고 특징점 추출 단계에서 적응적인 방법으로 잡음 픽셀을 제거함으로써 계산 오버헤드가 적은 최소제곱법을 이용하여 모형 파라미터를 추정한다. 그리고 단순화된 칼만 필터 모델을 적용하여 차선을 추적한다.

III. Real-time Lane Detection and Tracking for Mobile Robots

1. Vision-based Autonomous Driving System for Mobile Robots

본 논문에서 고려하는 모바일 로봇의 비전 기반 자율 주행 시스템의 구성은 Fig. 2와 같다.

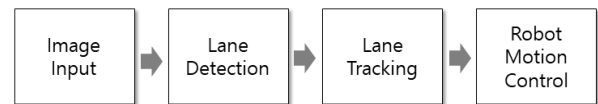


Fig. 2. Vision-based Autonomous Driving System for Mobile Robots

상기의 비전 기반 자율 주행 시스템은 단일 카메라 장치로부터 모바일 로봇의 전방 및 바닥 차선에 대한 정보를 영상으로 입력받아 일련의 영상처리와 차선 모형 추정 기법을 적용하여 좌우 차선을 검출하고 이를 이용하여 추적 궤적을 예측한 다음, 궤적 정보를 모션 제어 장치에 전달하여 모바일 로봇의 주행을 제어한다.

본 논문은 위의 시스템 구성에서 모션 제어 시스템이 주어진 궤적 입력에 대해 적절하게 제어 동작을 수행한다고 전제하고 연구 주제를 차선 검출 및 차선 추적으로 제한한다. 그리고 본 논문은 일반적인 환경을 고려하지만 모바일 로봇의 하드웨어 한계와 실험 환경의 제약 등으로 다음의 두 가지 제약 조건을 고려한다.

(1) 제한된 트랙 환경

실제 모바일 로봇의 주행 여건을 반영하여 주행 트랙에 대해 다음과 같은 사항을 가정한다.

- 모바일 로봇은 실내에서 정해진 트랙을 따라 주행하도록 한다.
- 트랙에는 직선형 도로와 급회전 도로 등의 회전형 도로를 포함한다.
- 트랙의 왼쪽은 노란색 차선, 오른쪽은 흰색 차선으로 경계가 표시되며, 차선은 일정한 두께를 가진다.
- 트랙 바닥에는 차선 이외의 다른 표시가 없다.

(2) 저성능의 모바일 로봇 제어 장치

모바일 로봇은 주 제어장치로 범용 PC보다 낮은 컴퓨팅 성능의 장치를 내장한 것으로 가정한다. 이는 자율 주행 시스템의 실시간 처리 요건을 만족하기 위해서는 차선 검출 및 추적을 위해 계산 부하가 적은 알고리즘을 적용하여야 함을 의미한다.

2. State Machine of Lane Detection and Tracking System

본 논문에서 제안하는 차선 검출 및 추적 시스템도 기본적으로 Fig 1과 같은 알고리즘을 따르지만 앞 절에서 제시한 제약 조건을 고려하여야 한다.

이전의 연구들은 좌우 차선이 모두 입력 영상에 나타나고 차선이 근거리에서는 직선 형태를, 원거리에서는 곡선 형태를 가지면서 원근 투영으로 소실점(vanishing point)을 가진다는 기하학적 특성을 활용하여 차선 검출 알고리즘을 제안하였다[1]. 그러나 본 논문에서 고려하는 제한된 트랙에서는 Fig. 3과 같이 곡선 차선 또는 한쪽 차선만 입력되는 경우 그리고 소실점이 없는 경우도 발생한다. 따라서 제한된 트랙에서는 하나의 차선 형태만을 고려한 검출 기법만으로 차선을 정확하게 검출하기 힘들다.



Fig. 3. Various Lane Shapes on the Constrained Track

본 논문은 저성능의 제어보드에서 실시간 처리를 위해 컬러 기반 특징점 추출 방법을 적용하는데, 조명 등의 환경 요인에 의해 성능이 민감하게 좌우된다. 그러나 제한된 트랙의 차선 컬러 및 실내 주행 등에 대한 제약 조건을 고려하여 컬러 기반 추출에서 요구되는 컬러 필터의 임계값들을 실제 주행 환경에 맞게 동적으로 조정하면 특징점 추출의 정확도와 안정성을 향상시킬 수 있다. 본 논문은 이러한 분석 내용을 바탕으로 차선 추출 및 추적 시스템이 다중 모드로 동작하도록 설계하였으며, 차선 추출 및 추적 시스템의 상태 기계를 Fig. 4와 같이 정의하였다.

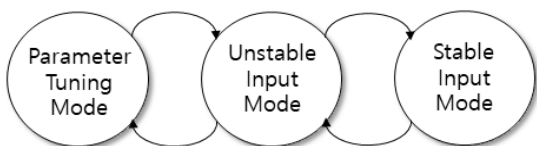


Fig. 4. State Machine for Proposed Lane Detection and Tracking System

파라미터 튜닝 모드는 자율 주행 시작 시점과 특징 픽셀을 추출하지 못하는 경우에 현재 동작 환경에 맞게 컬러 필터의 임계값을 동적으로 조정하는 연산을 수행한다. 불안정 입력 모드는 자율 주행 시작 시점과 곡선 차선 또는 한쪽 차선만이 입력되는 경우에 차선을 검출하여 추적하는 연산을 수행한다. 안정 입력 모드는 입력 영상으로 좌우 차선이 직선 형태로 입력되는 경우에 차선을 검출하여 추적하는 연산을 수행한다. 위의 상태 기계에서 상태 간의 전이는 입력 영상으로부터 차선 존재 여부 및 차선 유형을 판단하여 이루어진다.

이러한 다중 모드의 동작 방식은 연산 알고리즘을 동작 모드에 맞게 최적화하여 실시간 처리 요건을 만족시킬 수 있고 동작 모드에 따라 주행 속도를 조절하는 등의 후속 처리를 적응적으로 수행할 수 있는 장점이 있다.

3. Image Preprocessing

본 논문은 차선 추출 및 추적 시스템의 다중 동작 모드에 대해 동일하게 이미지 전처리 과정을 적용한다. 이미지 전처리 과정은 Fig. 5와 같다.

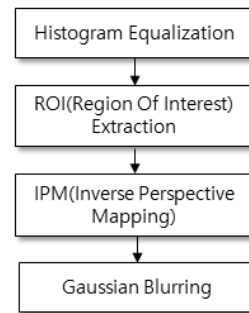
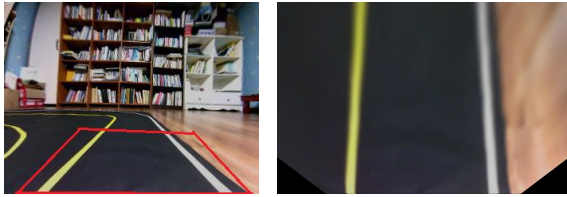


Fig. 5. Image Preprocessing Procedure

먼저 히스토그램 평활화를 통해 입력 영상의 밝기 대비를 향상시키고, 관심영역(ROI: Region Of Interest) 설정을 통해 차선 정보만 가지는 영역을 추출하여 연산함으로써 계산 오버헤드를 줄이고 차선 추출에서 잡음으로 작용하는 요소들을 사전에 제거하여 강인성도 높일 수 있다. 제한된 트랙의 입력 영상에서는 한 차선만 나타나는 등 소실점이 존재하지 않는 경우도 있어 소실점 기반의 검출 기법을 적용할 수 없으며, 모바일 로봇의 자율 주행 제어에서는 차선 전체 정보보다 가까운 전방 차선 정보가 더 중요하다. 따라서 Fig. 6의 (a)와 같이 관심영역은 카메라의 원근 투영을 고려하고 전방 차선이 포함되도록 입력 영상의 하단 부분에서 사다리꼴 형태의 영역으로 설정한다.



(a) ROI (b) IPM image

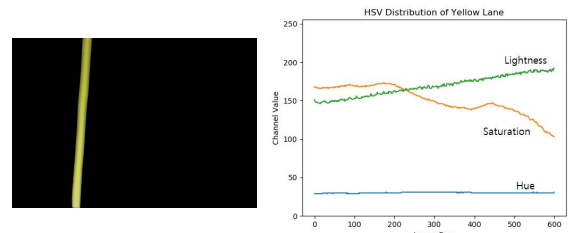
Fig. 6. Image Preprocessing Example

관심영역에 대해 원근 투영을 제거하고 실제 환경과 유사한 좌우 평행 차선 정보를 얻기 위해 역투영 변환(IPM: Inverse Perspective Mapping)을 통해 위에서 보는 영상(top-view image)을 생성한다. 본 논문에서는 IPM을 위해 물리적인 카메라 보정 파라미터를 사용하지 않는 역사다리꼴 변환(inverse trapezoidal transform)을 이용한다[14]. Fig. 6의 (b)와 같이 변환된 IPM 영상에 대해 5x5의 가우시안 커널을 이용한 컨볼루션 연산(convolution operation)을 통해 고주파 잡음을 제거한다.

4. Parameter Tuning

제한된 트랙에서는 트랙 경계를 왼쪽은 노란색 차선으로, 오른쪽은 흰색 차선으로 표시한다. 본 논문은 차선 색상이 식별성이 높다는 특징을 바탕으로 컬러 필터링을 이용하여 입력 영상으로부터 차선 특징 픽셀을 추출한다. 컬러 필터링은 RGB 컬러 공간보다 컬러 대비가 높은 HSV 컬러 공간에서 효과적으로 동작하는데, 필터링 연산을 위해 3가지 컬러 요소, 색조(Hue), 채도(Saturation), 명도(Lightness)에 대한 상한 및 하한 임계값 파라미터들을 요구한다. 이들 파라미터 중에서 채도와 명도는 카메라 장치의 고유 특성과 조명 등에 영향을 받는다. 따라서 색조 파라미터는 노란색과 흰색이 다른 색 대비 비교적 명확하게 구분되어 고정 값으로 설정하도록 하고, 채도와 명도 파라미터는 동작 환경에 맞추어 동적으로 조정한다.

Fig. 7은 추출된 노란색 차선 픽셀에 대해 수직 위치별 색조, 채도, 명도의 분포를 보여주는데, 색조는 거의 동일한 값을 가지나 영상 하단 부분에서는 채도가 낮고 명도가 높아 흰색에 가깝고, 영상 상단 부분에서는 채도는 높고 명도가 낮아 회색 성분을 포함하고 있음을 확인할 수 있다. 또한 조명이 차츰 어두워질 때 채도의 상한 값은 높아지고 명도의 하한 값은 낮아진다.



(a) Detected Yellow Lane (b) HSV Distribution

Fig. 7. HSV Distribution of Detected Yellow Lane

본 논문은 제안된 트랙의 차선이 일정한 두께를 가지는 기하학적 특징을 바탕으로 검출된 차선이 일정한 두께를 가질 때까지 채도와 명도의 임계값 파라미터를 동적으로 조정하는 알고리즘을 제안하였으며, Fig. 8과 같다.

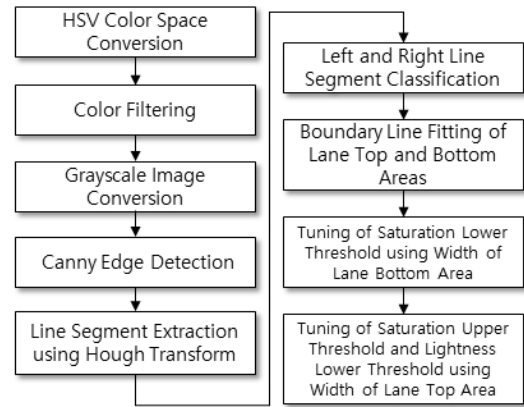


Fig. 8. Parameter Tuning Procedure

파라미터 튜닝은 자율 주행 시작 전에 수행하며, 이 시점에는 안정된 영상 입력, 즉 좌우 직선 차선이 모두 표시되는 영상이 입력된다. 이에 파라미터 조정 알고리즘은 좌우 차선이 모두 표시되는 영상을 전제로 하고, 일반적인 조명에서의 차선 추출 임계값을 초기 파라미터로 설정하며, 명도의 상한 임계값은 최대값으로 설정한다.

전처리된 영상에 대해 HSV 컬러 공간으로 변환하고 기본 임계값 파라미터를 이용하여 컬러 필터링을 수행하여 특징 픽셀을 추출한다. 노란색 차선과 흰색 차선을 분리하여 추출하고 이후의 조정 연산은 컬러별 픽셀 집합에 대해 개별적으로 실행한다. 차선의 두께는 차선의 좌우 경계선에 대응하는 1차원 직선을 검출한 다음, 이를 이용하여 같은 수직 위치에 있는 좌우 경계점을 산출하고 x-좌표 값의 차이로 차선의 두께를 계산한다.

추출된 픽셀 집합을 마스크로 사용하여 그레이스케일 영상을 생성하고 Canny 에지 검출 기법을 적용하여 에지 픽셀들을 추출한 다음, 호프 변환(Hough Transform)을 적용하여 차선 경계를 구성하는 선분 요소(line segment)

들을 검출한다. 선분 요소들은 좌측 경계 선분과 우측 경계 선분으로 분류하는데, 분류 방법은 Canny 에지 검출 과정에서 수행한 x-축 방향 소벨(Sobel) 연산으로 산출된 경사도(gradient)의 부호를 활용한다. 즉 선분의 시작점인 (i, j)-좌표 픽셀의 x-축 방향 소벨 경사도를 $G_x(i,j)$ 라고 하면 다음과 같이 분류한다.

if $G_x(i,j) > 0$ then 좌측 경계 선분
 else if $G_x(i,j) < 0$ then 우측 경계 선분

차선의 상단 부분과 하단 부분에 대한 경계 직선을 산출하기 위해 각각의 높이를 영상 높이의 1/6 크기로 설정하고 분류된 선분 요소 중에서 상단 및 하단 부분에 포함되는 선분 요소만을 선별하고, 이들 선분 요소에 대해 최소 제곱법을 적용하여 1차원 직선을 산출한다.

하단 부분에서 차선 두께를 산출하고 지정된 크기보다 작으면 채도의 하한 임계값을 감소시키고, 반대이면 증가시켜 조정한다. 상단 부분에서 차선 두께를 산출하고 지정된 크기보다 작으면 명도의 하한 임계값을 감소시키고 채도의 상한 임계값을 증가시킨다. 만약 두께가 지정된 크기보다 크면 반대로 조정한다. 이러한 파라미터 조정 과정을 반복하여 실행하고, 일정 시간 동안 임계값 파라미터를 안정적으로 유지하면 파라미터 조정 과정을 종료하도록 하였다. Fig. 9는 노란색 차선의 파라미터 조정 과정에서 차선의 두께 변화를 보여주고 있다.

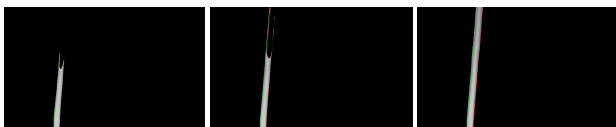


Fig. 9. Width Change of Yellow Lane in Parameter Tuning

5. Real-time Lane Detection

본 논문에서 제안하는 차선 검출 알고리즘은 Fig. 1과 유사하게 특징점 추출, 차선 존재 및 유형 판별, 차선 특징점 검색 그리고 차선 모형 추정 연산으로 이루어진다.

특징점 추출은 동적으로 조정된 임계값 파라미터를 이용하여 컬러 필터링을 수행하고 차선 컬러에 해당하는 특징 픽셀을 추출한다. 이 단계에서 노란색 차선과 흰색 차선을 분리하여 특징 픽셀을 추출하며, 이후의 검출 연산들을 차선별 픽셀 집합에 대해 개별적으로 실행한다.

차선 존재 여부 및 차선 유형 판별은 제안된 상태 기계에서 상태 천이 여부를 결정하며, 다음과 같은 차선의 기하학적인 특성을 활용한다.

- 영상에서 좌우 차선의 시작점은 각각 좌우 하단 영역으로 명확하게 구분된다.

- 트랙 중간에는 어떤 형태의 표시도 없으므로 좌측 차선의 시작점은 영상 왼쪽 하단에서 오른쪽으로 편향되어 나타나며, 우측 차선의 시작점은 영상 오른쪽 하단에서 왼쪽으로 편향되어 나타난다.

- 곡선 트랙은 좌우 차선의 픽셀 수가 크게 차이가 나며, 직선 트랙은 좌우 차선의 픽셀 수의 차이가 작다.

- 직선 차선은 일정 수 이상의 픽셀을 가진다.

Fig. 10에서와 같이 입력 영상을 수평 중앙을 기준으로 좌우 두 영역으로 나누고, 각 영역의 하단에서 수직 높이의 1/6 영역을 추출한다. 각 영역에 대해 연결된 픽셀들을 묶기 위해 레벨링 연산(labeling operation)[22]을 수행하여 특징 픽셀 영역들을 구분한다. 그리고 좌측 영역에 대해서는 제일 오른쪽에 있는 영역을 왼쪽 차선의 시작 부분으로 선택하고, 우측 영역에 대해서 제일 왼쪽에 있는 영역을 오른쪽 차선의 시작 부분으로 선택한다.

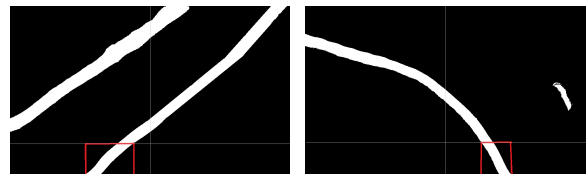


Fig. 10. Start Part Detection of Left-Right Lanes

좌우 시작 부분의 크기 차이가 크거나 한쪽 부분만 나타나는 경우에는 불안정 입력 모드로 전환하고, 좌우 시작 부분의 크기 차이가 작고 이 상태가 일정 시간동안 연속되면 안정 입력 모드로 전환하여 차선을 검출한다.

(1) 불안정 입력 모드에서의 차선 검출

불안정 입력 모드는 주로 곡선 차선 영역에 해당한다. 곡선 차선에서는 급격한 회전으로 좌우 차선의 영역이 크게 차이가 나거나 하나의 차선만 입력된다. 본 논문에서 불안정 입력 모드에서 차선 픽셀만을 추출하기 위해 개선된 슬라이딩 윈도우 검색 기법을 적용한다.

슬라이딩 윈도우 검색 기법은 Fig. 11의 (a)와 같이 영상 하단에서 상단까지 뻗어 있는 직선 차선에 해당하는 특징 픽셀을 추출하기 위해 제안된 방법으로서 차선의 두께 등을 고려하여 윈도우 크기를 설정하고 윈도우 내에 있는 특징 픽셀만을 추출한다. 윈도우 검색은 차선의 하단 부분에서 시작하며, x-축으로 현재 윈도우에서 검색된 픽셀 집합의 무게 중심의 x 좌표로, y-축으로는 윈도우 높이만큼 검색 윈도우를 이동시키면서 검색한다.

그러나 곡선 차선의 경우는 Fig. 11의 (b)와 같이 윈도우가 수직방향으로만 이동하면 유효한 특징 픽셀을 검색

하기 어렵다. 이에 현재 윈도우에서 검색된 특징 픽셀이 없는 경우에 이전의 윈도우 이동 방향을 고려하여 윈도우를 수평방향으로도 이동하도록 개선하였으며, 윈도우가 영상의 상단 경계뿐만 아니라 좌측 또는 우측 경계를 만날 경우에 검색을 종료하도록 하였다. Fig. 11의 (c)는 개선된 슬라이딩 윈도우 기법을 통해 곡선 차선의 특징 픽셀을 검색하는 결과를 보여주고 있다.

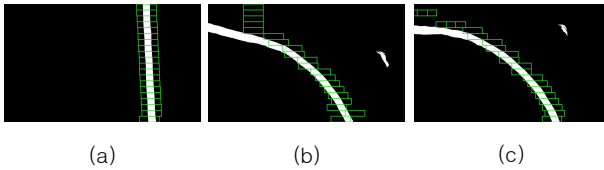


Fig. 11. Example of Refined Sliding Window Search

(2) 안정 입력 모드에서의 차선 검출

안정 입력 모드는 주로 직선 차선 영역에 해당한다. 입력 영상에 병행하는 좌우 차선이 모두 나타나고 연속적인 입력 영상 간에 차선 위치 및 차선 모형의 변화가 크지 않은 특징을 갖는다. 이러한 시간적 특성과 직선 차선에서는 차선 두께가 일정하게 나타나는 기하학적 특성을 바탕으로 이전 영상에서 산출된 차선 모형을 이용하여 차선의 중앙점을 산출하고 중앙점 기준 x-축 방향으로 일정한 구간 안에 있는 픽셀만 검색하는 구간 검색 기법을 적용한다. 구간 검색 기법은 슬라이딩 윈도우 기법보다 계산 오버헤드가 적어 검색 시간을 단축할 수 있을 뿐만 아니라 비슷한 컬러를 가진 잡음 요소를 쉽게 배제할 수 있고, 또한 장애물에 의해 차선 영역이 일부 가려지더라도 차선 모형을 활용하기 때문에 잡음에 대한 강인성을 높일 수 있다.

(3) 차선 모형 추정

본 논문은 차선 모형을 2차원 곡선(Quadratic Curve)로 정의한다. 특징 픽셀 집합을 이용하여 특정 모형을 추정하는 방법으로 RANSAC 추정이나 파티클 필터 등이 있으나 이들 기법은 잡음이 포함된 특징 집합을 전제로 추정 연산을 수행하고 계산 오버헤드가 크다. 그러나 본 논문에서는 이미 차선 픽셀 검색 단계에서 잡음을 제거하고 차선에 해당하는 픽셀만을 추출하였기에 차선 픽셀 집합에 최소제곱법(Least Squared Method)을 적용하여 차선 모형의 파라미터를 추정한다. 식 (1)은 추정된 좌우 차선의 차선 모형을 정의한 식이다.

$$\begin{aligned} f_l(y) &= a_{l1} \cdot y^2 + a_{l2} \cdot y + a_{l3} \\ f_r(y) &= a_{r1} \cdot y^2 + a_{r2} \cdot y + a_{r3} \end{aligned} \quad (1)$$

6. Kalman Filter based Lane Tracking

본 논문은 모바일 로봇의 트랙 주행을 목표로 하고 있어 검출된 좌우 차선을 이용하여 트랙 중앙선을 예측하고 이를 추적한다. 우선 트랙 주행에서 요구되는 트랙 중앙선의 파라미터를 정의한 다음, 이를 예측하고 추적하기 위한 칼만 필터 모형과 처리 과정을 제시한다.

입력 영상에서 트랙 중앙선은 차선 검출 단계에서 좌우 차선이 모두 검출되는 경우에 평균 계산법을 적용하여 산출한다. 즉, 추정된 차선 모형을 이용하여 좌우 차선의 중앙점들을 산출하고 좌우 차선의 서로 대응하는 중앙점들의 좌표를 평균하여 트랙 중앙선을 산출한다. 좌우 차선의 중앙선을 식 (2)와 같이 점들의 집합으로 정의하면 트랙 중앙선은 식 (3)과 같이 정의할 수 있다. 여기서, (x, y) 는 영상의 픽셀 좌표를, I_H 는 영상의 수직방향 픽셀 수를 의미한다.

$$\begin{aligned} L_l &= \{ (x_l, y) \mid x_l = f_l(y), y = 0, \dots, I_H - 1 \} \\ L_r &= \{ (x_r, y) \mid x_r = f_r(y), y = 0, \dots, I_H - 1 \} \end{aligned} \quad (2)$$

$$L_c = \left\{ (x_c, y) \mid x_c = (f_l(y) + f_r(y))/2, y = 0, \dots, I_H - 1 \right\} \quad (3)$$

만약 한 차선만 검출되는 경우는 식 (4)와 같이 추출된 차선에 트랙 폭의 반에 해당하는 길이를 더하거나 빼어 트랙 중앙선을 산출한다. 여기서, W_T 는 트랙 폭의 픽셀 수를 의미한다. 그리고 검출된 차선이 2차원 곡선 모형이므로 트랙 중앙선도 2차원 곡선 모형으로 갖는다.

$$\begin{aligned} L_c &= \{ (x_c, y) \mid x_c = f_l(y) + (W_T/2), y = 0, \dots, I_H - 1 \} \\ &\text{or} \\ L_c &= \{ (x_c, y) \mid x_c = f_r(y) - (W_T/2), y = 0, \dots, I_H - 1 \} \end{aligned} \quad (4)$$

일반적으로 모바일 로봇의 트랙 주행에서는 트랙의 특정 위치를 추적점으로 설정하고 추적점의 위치 변화에 따라 주행 속도 및 방향을 제어한다. 따라서 트랙 중앙선의 형태를 추적하기 보다는 트랙 중앙선 상의 특정 추적점이 어떻게 이동하는지를 예측하고 추적하는 것이 더 바람직하다. 추적점의 위치를 트랙 중앙선 상에서 모바일 로봇 전방의 가까운 위치, 즉 영상 하단부에서 특정 높이의 점으로 설정하면 Fig. 12의 (a)와 같이 추적하려는 트랙 중앙선을 2차원 곡선에서 중앙선 시작점과 추적점을 지나는 1차원 직선으로 단순화시킬 수 있다. 그리고 1차원 직선은 Fig. 12의 (b)처럼 x-축 교차점의 x-좌표 값 d 와 y-축과 이루는 각도 ψ 로 정의할 수 있다. 트랙 중앙선의 움직임 모형은 비선형이므로 직선의 파라미터를 추적하기 위해서는 EKF(Extended Kalman Filter) 또는 UKF(Unscented Kalman Filter)를 적용하는 것이 효과적이다[3].

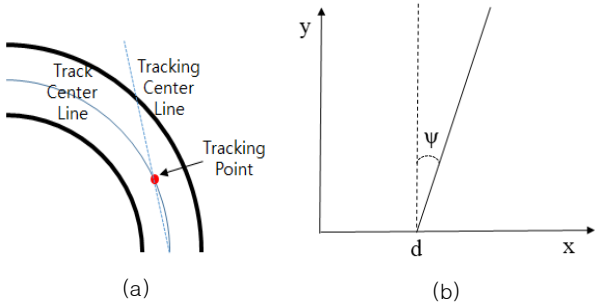


Fig. 12. Simplified Tracking Line and Parameters

본 논문은 ROS(Robot Operating System) 프레임워크 [23] 기반으로 모바일 로봇을 구현하며, ROS 프레임워크에서는 Twist 명령, 즉 [직선 속도(Linear Velocity), 각속도(Angular Velocity)]의 명령을 이용하여 모바일 로봇의 움직임을 제어한다. 직선 속도는 주로 직선 방향으로 움직이는 거리에 의해 결정하며, 각속도는 회전 각도의 크기에 의해 결정한다. 주행 제어 명령과 트랙 중앙선의 파라미터 사이의 연관성을 살펴보면, x-축 거리 d는 직선 속도와 연관성이 없으며 각도 ψ는 각속도를 결정한다. 따라서 본 논문은 트랙 중앙선의 파라미터 중에서 각도 요소만 추적하고자 한다. 트랙 전체에서 각도 변화가 적은 직선형 트랙이 곡선형 트랙에 비해 비중이 높아 각도 파라미터의 움직임 모형을 선형으로 근사할 수 있어 선형 칼만 필터를 이용하여 예측하고 추적한다. 본 논문에서 적용한 각도에 대한 칼만 필터 모형 및 추적 과정은 다음과 같다.

(1) 상태 정의

칼만 필터를 이용해 추적하려는 상태와 움직임 모형은 다음과 같다.

$$X = [\psi, \omega]^T$$

$$\psi_{k+1} = \psi_k + \Delta t \cdot \omega_k \Rightarrow \hat{X} = F \cdot X = \begin{bmatrix} 1.0 & \Delta t \\ 0 & 1.0 \end{bmatrix} \cdot X$$

$$P_o = \begin{bmatrix} 1.0 & 0.1 \\ 0.1 & 1.0 \end{bmatrix} \quad (5)$$

여기서, φ: 회전 각도, ω: 회전 각속도, Δt: 경과 시간, P₀: 상태 공분산의 초기값을 의미한다. 그리고 예측값과 측정값 사이의 관계를 정의하는 측정 모형은 다음과 같다.

$$z = \psi_k \Rightarrow z = H \cdot X = [1.0 \ 0.0] \cdot X \quad (6)$$

$$R = [1.0]$$

여기서, z: 측정값, R: 측정 잡음 분산을 의미한다.

(2) 측정 업데이트

k-번째 측정값이 주어졌을 때에 상태를 다음과 같이 업데이트한다.

$$error = z_k - H \cdot \hat{X}_k \quad (7)$$

$$K = \hat{P}_k \cdot H^T \cdot (H \cdot \hat{P}_k \cdot H^T + R)^{-1}$$

$$X_k = \hat{X}_k + K \cdot error$$

$$P_k = (I - K \cdot H) \cdot \hat{P}_k$$

(3) 상태 예측

업데이트된 상태를 이용하여 다음 상태를 예측한다.

$$\hat{X}_{k+1} = F \cdot X_k \quad (8)$$

$$\hat{P}_{k+1} = F \cdot P_k \cdot F^T$$

(2)-단계와 (3)-단계를 반복하면서 예측된 상태, 즉 각도 값을 로봇 움직임 제어 시스템에 전송하여 로봇 움직임을 제어하도록 한다.

IV. Performance Evaluation

본 논문은 제한된 트랙의 조건에 맞게 실험 트랙을 구성하고 기존 소형 모바일 로봇의 하드웨어를 개선한 다음, 트랙 주행 실험을 통해 제안된 차선 검출 및 추적 시스템의 성능을 평가하였다. 실험 모바일 로봇은 Robotis Turtlebot-3 Burger 모델로서 주요 기구적인 사양은 Table 1과 같고 ROS 프레임워크 기반으로 동작한다[24].

Table 1. Mechanical Spec. of Mobile Robot

Size(LxWxH)	138x178x192(mm)
Weight	1 kg
Max Linear Speed	0.22 m/sec
Max Rotation Speed	2.84 rad/sec
Drive Motor	2's Dynamixel XL430- W250 Servo Motors
Wheel	2 active wheels, 1 auxiliary wheel

Table 2. Specification of Main Control B/D

CPU	Samsung Exynos5422 Cortex™-A15 2GHz & Cortex™-A7 Octa-core CPU
GPU	Mali-T628 MP6
RAM	2Gbyte LPDDR3 RAM
Storage	eMMC5.0 HS400 Flash Storage (64GB)
OS	Ubuntu 18.04 ARM version

실험 로봇의 성능 개선을 위해 Odroid XU-4 보드를 내장하였으며, 주요 사양은 Table 2와 같다[25]. 그리고 카메라 장치는 ELP 5MP USB 카메라로서 UVC(USB Video Camera)-1.1 호환성을 지원한다. Fisheye 렌즈를 장착하여 170도 광각 영상을 출력하며, 영상 캡처 및 처리시간 단축을 위해 320x240 영상을 출력하도록 설정하였다.

본 논문은 차선 검출 및 추적 시스템을 ROS Kinetic Kame 프레임워크[24]를 사용하여 구현하였다. 주요 특징은 동작 노드를 roscpp 패키지 활용하여 C++ 언어로 구현하였고, 토픽을 통한 영상 데이터 전송 오버헤드를 줄이기 위해 zero-copy 전송을 지원하는 ROS nodelet 기법을 적용하였다. 주행 실험을 위해 제한된 트랙의 특징을 최대한 반영한 실험 트랙을 제작하였으며, Fig. 13는 실험 트랙의 전체 구성과 세부 곡선 트랙을 나타낸 것이다.

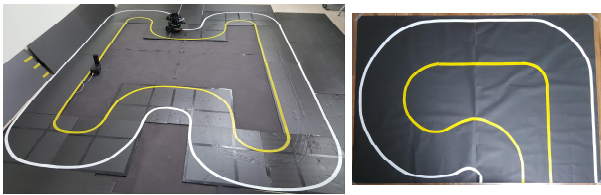


Fig. 13. Test Track for Performance Evaluation

제한된 파라미터 튜닝 알고리즘의 성능 평가는 다양한 조명 환경에서 차선 검출 파라미터를 적응적으로 추출하는지 여부와 처리 속도를 측정하였다. Fig. 14에서 (a)는 일반적인 조명인 경우, (b)는 조명이 어두운 경우, (c)는 조명이 밝은 경우에 파라미터 조정으로 일정한 두께의 차선 검출 여부를 보여주는데, 조명 변화에 적응적으로 파라미터 조정을 수행함을 알 수 있으며, 어두운 조명에서는 상한과 하한 임계값의 범위가 커져 다른 영역을 잡음으로 추출함을 볼 수 있다. 그리고 처리 시간은 조명 환경에 따라 다소 차이가 있지만 평균적으로 35 프레임 내에서 컬러 필터의 임계값 파라미터를 추출하였다. 파라미터 안정 여부 판별을 위한 지연 시간 20 프레임을 고려할 때에 55 프레임 내에서 파라미터 튜닝이 이루어졌으며, 처리 시간이 평균적으로 3.23초 경과하여 17Hz의 처리 속도를 보였다.

차선 검출 및 추적 알고리즘의 성능은 실험 트랙에서 자율 주행 실험을 통해 평가하였다. 수차례 주행 실험을 실시한 결과, 차선이탈 없이 정상적으로 주행하는 것을 확인하였으며, Fig. 15는 주행 중에 검출한 트랙 중앙선 결과를 사례별로 보여준다. Fig. 15에서 (a)는 안정 입력 모드에서의 결과를, (b)은 불안정 입력 모드의 곡선 트랙에서의 결과를, (c)은 한쪽 차선만 입력되었을 때의 결과를, (d)은 장애물에 의해 차선 일부가 보이지 않는 경우에 검출 결과를 보여주고 있다. 또한, 모바일 로봇이 트랙을 이탈하지 않고 정상 주행을 하는 경우에는 입력 영상에서 차선을 추출되지 않는 경우는 발생하지 않았다.

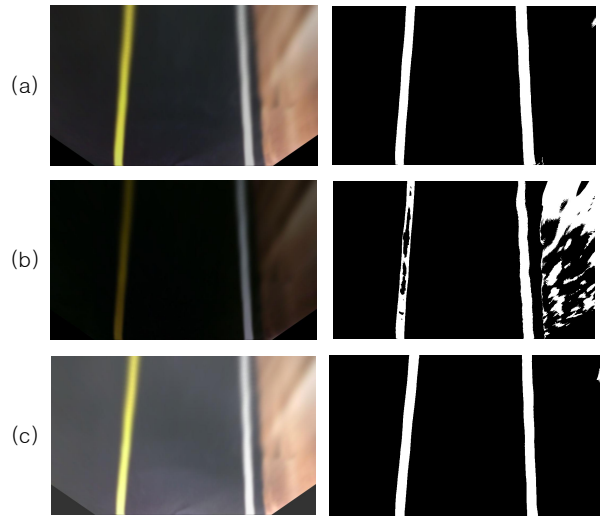


Fig. 14. Test Results of Parameter Turning Algorithm

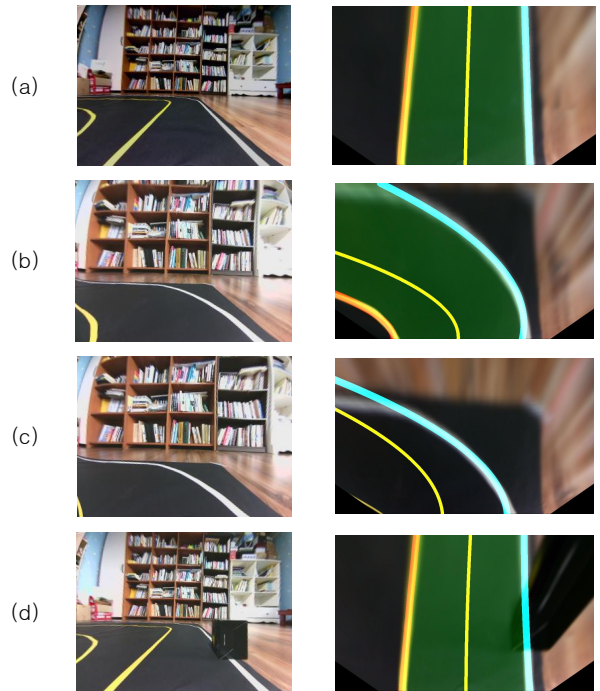


Fig. 15. Detection Results of Track Center Lines

Table 3은 실험 트랙 1회 주행에서 각 동작 모드별로 처리된 프레임 수와 평균 프레임 처리시간을 보여준다.

Table 3. Image Processing Throughput by Operation Mode

Operation Mode	Number of Processed Frames	Mean Frame Processing Time(msec)
Stable Input Mode	862	93
Unstable Input Mode	522	81

안정 입력 모드에서 처리된 영상 프레임 수가 많고 처리 시간이 짧아 전체 주행 시간을 단축시키는 효과를 얻고 있음 알 수 있다. 그리고 평균 영상 처리율은 11.7 Hz가 나오는데, 이는 주 제어보드의 성능과 로봇의 주행 성능을 고려할 때에 곡선 트랙 등을 실시간적으로 추적하는데 충분한 성능으로 평가된다.

칼만 필터 기반 차선 추적은 자율 주행의 정확도와 강인성을 향상시키는 효과를 얻을 수 있는데, 이를 실험을 통해 검증하였다. Fig. 16은 차선 검출만을 통한 추적과 칼만 필터를 적용한 추적에 대해 회전 각도 출력을 900개의 영상 프레임 동안 비교하였다. Fig. 16에서 회전 각도의 변화가 큰 구간이 급회전 곡선 트랙에 해당하며, 주행 초반에 안정 입력 모드 상태라 큰 차이를 보이지 않지만 연속된 회전 구간에서는 칼만 필터 기반 추적이 상대적으로 안정적인 출력을 보임으로써 로봇의 움직임을 안정적이고 정확하게 제어하고 있음을 알 수 있다.

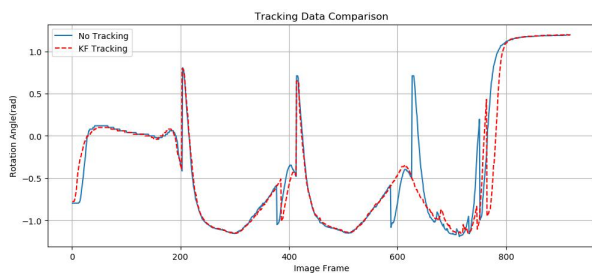


Fig. 16. Lane Tracking Comparison between No-Tracking Mode and Kalman-Filter based Tracking Mode

V. Conclusion

실생활에서 모바일 로봇의 응용이 늘어나면서 저비용의 비전 기반 자율 주행 기술에 대한 요구가 증대되고 있다. 본 논문은 실내 환경에서의 로봇 주행 여건을 반영한 제한된 트랙을 가정하고, 제한된 트랙에서 단일 카메라로부터 입력되는 영상을 기반으로 자율 주행을 지원하는 비전 기반 실시간 차선 검출 및 추적 시스템을 제안하고 성능을 검증하였다.

본 논문은 먼저 다양한 형태의 차선 입력과 연산 파라미터의 동적 조정 등을 고려하여 다중 동작 모드의 시스템 구조와 상태 기계를 설계하였다. 파라미터 튜닝 모드에서는 차선의 두께가 일정하다는 기하학적인 특성을 바탕으로 차선 특징 픽셀을 추출하는 컬러 필터의 임계값 파라미터를 동작 환경에 맞게 동적으로 조정하였다. 불안정 입력 모드에서는 곡선 차선 또는 한쪽 차선 입력에 대해 차선의 기하학적인 특성을 고려한 개선된 슬라이딩 윈도우 검색

을 통해 차선 픽셀을 추출하고 최소제곱법을 이용하여 차선 모형을 추정하였다. 안정 입력 모드에서는 연속된 영상 프레임 간에 병행하는 좌우 차선의 모양과 위치 변화가 적다는 시간적 특성을 반영한 구간 검색을 통해 차선 픽셀을 추출하여 차선 모형을 추정하였다. 또한 트랙 주행에 필요한 트랙 중앙선을 추적하기 위해 중앙선 모형을 단순화시키고 선형 칼만 필터를 통해 추적함으로써 차선 추적의 정확도와 강인성을 향상시켰다. 제한된 트랙에 대한 실험 트랙을 제작하고 제안된 차선 검출 및 추적 시스템을 소형 모바일 로봇에서 ROS 프레임워크 기반으로 구현하여 주행 실험을 통해 성능을 평가하였다. 동적 파라미터 조정을 통해 계산 오버헤드가 적은 영상처리 기법들을 적용함으로써 저성능의 제어보드에서 차선 이탈 없이 제한된 트랙을 주행하도록 실시간 제어가 가능한 것을 확인하였다.

본 논문은 제한된 트랙에 대해 여러 가지 제약 조건을 두어 저비용 자율 주행의 실시간성을 만족시키고자 하였다. 이로 인해 제안된 알고리즘의 범용성이 많이 부족하며, 향후에는 이들 제약 조건의 해결 방법에 대해 연구할 예정이며, 또한 영상 센서와 소형 레이더 센서를 융합한 저비용 자율 주행 기법에 대한 연구도 병행할 예정이다.

REFERENCES

- [1] Y. Xing et al., "Advances in Vision-Based Lane Detection: Algorithms, Integration, Assessment, and Perspectives on ACP-Based Parallel Vision," in *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 645-661, May 2018. DOI: 10.1109/JAS.2018.7511063
- [2] Zhang, H., Hernandez, D.E., Su, Z. and Su, B., "A Low Cost Vision-Based Road-Following System for Mobile Robots," *MDPI Applied Sciences*, 1635, August 2018. DOI: 10.3390/app8091635
- [3] Kuo, C Y et al. "On the Image Sensor Processing for Lane Detection and Control in Vehicle Lane Keeping Systems." *Sensors (Basel, Switzerland)*, vol. 19,7 1665. Apr. 2019. DOI:10.3390/s19071665
- [4] Sun, T. Y., Tsai, S. J. and Chan, V., "HSI color model based lane-marking detection," *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pp. 1168-1172, 2006.
- [5] H. Cheng, B. Jeng, P. Tseng and K. Fan, "Lane Detection With Moving Vehicles in the Traffic Scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 571-582, Dec. 2006.
- [6] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.

- PAMI-8, no. 6, pp. 679-698, Nov. 1986.
- [7] Q. Lin, Y. Han and H. Hahn, "Real-Time Lane Departure Detection Based on Extended Edge-Linking Algorithm," 2010 Second International Conference on Computer Research and Development, pp. 725-730, 2010.
- [8] Y. Otsuka, S. Muramatsu, H. Takenaga, Y. Kobayashi and T. Monj, "Multitype lane markers recognition using local edge direction," Intelligent Vehicle Symposium 2002. IEEE, vol. 2, pp. 604-609, 2002.
- [9] J. W. Lee, and U. K. Yi, "A lane-departure identification based on LBPE, Hough transform, and linear regression," Computer Vision and Image Understanding, vol. 99, no. 3, pp. 359-383, 2005.
- [10] Aly, M., "Real time detection of lane markers in urban streets," Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 7-12, June 2008.
- [11] Deng, J. and Han, Y., "A real-time system of lane detection and tracking based on optimized RANSAC B-spline fitting," Proceedings of the ACM International Conference on Research in Adaptive and Convergent Systems, pp. 157-164, October 2013.
- [12] Tan, H. et al., "A novel curve lane detection based on improved river flow and RANSAC," Proceedings of the IEEE Conference on Intelligent Transportation Systems, pp. 133-138, October 2014.
- [13] Mammeri, A., Boukerche, A. and Lu, G., "Lane detection and tracking system based on the MSER algorithm, hough transform and kalman filter," Proceedings of the ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 259-266, September 2014.
- [14] Li, Y., Iqbal, A., and Gans, N. R., "Multiple lane boundary detection using a combination of low-level image features," Proceedings of the IEEE Conference on Intelligent Transportation Systems, pp. 1682-1687, October 2014.
- [15] M. B. de Paula and C. R. Jung, "Automatic Detection and Classification of Road Lane Markings Using Onboard Vehicular Cameras," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 6, pp. 3160-3169, Dec. 2015.
- [16] Seo, Y.W. and Rajkumar, R., "Utilizing instantaneous driving direction for enhancing lane-marking detection," Proceedings of the IEEE Intelligent Vehicles Symposium, pp. 170-175, June 2014.
- [17] Nan, Z., Wei, P., Xu, L. and Zheng, N. "Efficient Lane Boundary Detection with Spatial-Temporal Knowledge Filtering," Sensors, vol. 16, 1276, 2016.
- [18] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," IEEE Transactions on Intelligent Transportation Systems, vol. 7, no. 1, pp. 20-37, 2006.
- [19] R. Danescu and S. Nedevschi, "Probabilistic Lane Tracking in Difficult Road Scenarios Using Stereovision," IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 2, pp. 272-282, June 2009.
- [20] Berriel, R.F., Aguiar, E.D., and Oliveirasantos, T. "A particle filter-based lane marker tracking approach using a cubic spline model," Proceedings of the SIBGRAPI Conference on Graphics, Patterns and Images, pp. 149-156, August 2015.
- [21] Borkar, A., Hayes, M. and Smith, M.T., "Robust lane detection and tracking with ransac and Kalman filter," Proceedings of the IEEE International Conference on Image Processing, pp. 3261-3264, November 2009.
- [22] Kenji Suzuki et al., "Linear-time connected-component labeling based on sequential local operations," Computer Vision and Image Understanding, vol. 89, pp.1-23, 2003.
- [23] ROS, <http://www.ros.org>.
- [24] Robotis Turtlebot-3, <http://emanual.robotis.com/docs/en/platform/turtlebot3/specifications/#specifications>.
- [25] Hardkernel Odroid XU-4, <https://wiki.odroid.com/odroid-xu4/odroid-xu4>.

Authors



Young-Ju Kim received the B.S., M.S. and Ph.D. degrees in Computer Science from Busan University, Korea in 1988, 1990 and 1999, respectively. Dr. Kim joined the faculty of the Division of Computer Information

Engineering at Silla University, Busan, 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.