

## 우천 환경에서 다차선 검출을 위한 사전 지식기반 접근법

김진호·송진규·이준웅\*

전남대학교 산업공학과

### A Prior-Knowledge Based Approach for Multi-Lane Detection in Rainy Conditions

JinHo Kim · JinGyu Song · JoonWoong Lee\*

Department of Industrial Engineering, Chonnam National University, Gwangju 61186, Korea  
(Received 18 March 2025 / Revised 16 April 2025 / Accepted 21 April 2025)

**Abstract** : This study proposes a rule-based algorithm that leverages prior knowledge to detect multiple lane-lines in images that were captured under adverse weather conditions. Specifically, rainy weather poses significant challenges to lane detection, as road images captured under such conditions often contain random raindrops, wet surfaces, and puddles. These elements cause light reflections, and either blur or obscure lane boundaries. Despite these disturbances, the actual lanes on the road remain unchanged. Based on this observation, the algorithm searches for regions that locally preserve lane-line characteristics by placing multiple small windows along previously detected lane lines. Within each window, valid segments are detected by using engineered features, the Hough transform, and sequential filtering. The algorithm shows that these features outperform those learned by recent deep neural networks when capturing the properties of lane lines under rain-induced noise, thus achieving about 10 % higher F1 scores than recent approaches.

**Key words** : Autonomous driving(자율주행), Detection of multi-lanes(다차로 검출), Engineered features(엔지니어링 특징), Lane-line model(차선 모델), Local-to-global approach(지역에서 전역으로의 접근), Sequential filtering(순차적 필터링)

#### 1. 서론

최근 자율주행 기술이 상용화되면서 차로 정보의 중요성이 커지고 있다. 차로 정보는 차량의 주행 궤적 및 종·횡방향 위치 결정에 필수적이다.<sup>1)</sup> 즉, 차로 이탈 방지나 차로 변경에 관련된 기술들은 정확한 차로 정보 추출이 중요하다.<sup>1-5)</sup> 본 연구는 자율주행 기술에 활용할 수 있는 다차선 검출 알고리즘(Multi-Lane Detection Algorithm, MLDA)을 제안한다. MLDA의 설계에 경찰청이 제정한 교통노면표시 설치·관리 매뉴얼<sup>6)</sup>을 참고했다.

이 매뉴얼은 차로(Lane)와 차선(Lane-line 또는 Line)을 구분하고 있다. 차로는 차량이 다니는 길이고, 차선은 차로를 구분하는 경계다. 즉 차로의 좌우 경계가 차선인 것이다. 대부분의 차선은 아스팔트나 시멘트 포장로 위에 10 ~ 20 cm 범위 내에서 일정한 폭을 가진 띠 모양이다. U턴 구역과 같은 특정한 곳의 차선 폭은 30 ~ 45 cm이다.

따라서 차선도 폭을 갖고 있으므로 좌우 경계가 있다. 차선의 색상은 주로 흰색이지만, 노란색은 도로의 중앙선 또는 가장자리선을 나타내고 파란색은 버스 전용 차로의 경계를 나타낸다. 또한 차선은 점선, 실선 또는 복선으로 표시된다. 차로는 차량의 진행 방향에 따라 번호가 매겨지며, 가장 왼쪽 차로를 1차로라 하고 그 우측으로 차례로 2차, 3차로 등으로 지정된다. 차로의 최소 규정 폭은 3 m이며, 불가피한 경우 최소 2.75 m 이상이다.

제안된 MLDA는 이전 연구<sup>1)</sup>를 확장한 것으로 흑백 영상을 사용하며, 우천시 및 맑은 날씨의 주간과 야간에 차량 내부에 설치된 카메라로 촬영한 도로 영상에서 실시간으로 다차선 검출을 목표로 한다. 영상 촬영과 실험을 위해 카메라가 장착된 차량을 “실험 차량”이라 부른다.

도로의 다양한 요인들이 MLDA의 성능에 영향을 미친다. 차선 이외의 선들, 화살표들, 글씨들, 표식들, 차선

\*Corresponding author, E-mail: [joonlee@chonnam.ac.kr](mailto:joonlee@chonnam.ac.kr)

<sup>1)</sup>This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium provided the original work is properly cited.

규제봉 등 차로위의 각종 표시들이나 설치물들의 모양이 국부적으로 차선과 유사하여 MLDA가 차선 식별에 어려움을 겪는다. 도로 포장 재질도 MLDA의 성능에 영향을 미친다. 시멘트 포장에서의 흰색 및 노란색 차선은 아스팔트 포장에서의 동일 색상의 차선보다 대비가 낮아 차선 검출 성능이 떨어진다. 특히 우천시 차량의 앞유리에 맺힌 수많은 물방울들과 노면에 고인 물에 의해 차선이 가려지거나 흐릿해지고, 심지어 유사 차선을 생성하기도 한다. 또한 우천시 야간에 노면에 고인 물이나 노면의 젖은 물기에 의해 노면에 투사된 빛이 심하게 반사되고 반짝거린다. 이런 현상은 영상에서 차선 검출을 어렵게 하는 요인으로 작용한다.

최근 합성곱 신경망(Convolutional Neural Network, CNN)을 활용한 차선 검출 방법들이 제안되고 있다.<sup>7-10)</sup> 그러나 이 방법들은 우천시 촬영한 영상에서 차선 검출 성능이 떨어진다. 우천시 촬영한 영상과 청명시 촬영한 영상의 통계적 특성이 다른데, 대부분 청명시 촬영된 영상들로 훈련되고 평가되었다. 우천시 촬영한 영상에 CNN 방법을 적용하려면 훈련을 위해 대량의 GT(Ground Truth) 데이터 구축이 필요하다. 그러나 대량의 GT 데이터 취득은 쉽지 않으며 공개된 데이터 집합을 찾는 것도 어렵다. 즉 CNN 기반 방법은 일반성과 유연성이 부족하고, 차선 검출 성능이 GT 데이터의 가용성에 크게 의존한다.

제안된 MLDA는 대량의 GT 데이터를 요하는 학습에 의존하지 않는다. 단지 차선 설치 규정<sup>6)</sup>에 의거하여 영상으로부터 차선의 속성을 만족하는 특징 추출을 위해 다단계 픽셀 필터링을 도입하고, 비로 야기된 잡음 극복을 위해 L2GA(Local-to-Global Approach)를 제안한다. L2GA는 비로 인해 영상 전체에 무작위로 나타나 차선 검출 성능을 떨어뜨리는 잡음의 영향을 줄이기 위해 영상 전체를 대상으로 하지 않고, 국부적으로 조금이라도 차선의 속성을 유지하는 영역을 찾아 이 영역들을 통합할 목적으로 제안한 것이다. 또한 MLDA는 실험 차량이 주행중인 차로(이하 자차로라 함)와 자차로에 인접한 차로들까지 검출 대상으로 한다. 따라서 자차로의 경계인 좌우 차선과 인접 차로의 경계인 차선들까지 검출한다. 즉, 3개 차로에 걸쳐 4개 차선을 감지한다. 다수의 기존 연구들이 자차로 검출에 한정하고 있지만,<sup>29)</sup> 자율주행을 고려하면 최소한 자차로의 인접 차로들까지는 검출 대상으로 해야 한다.

제안된 MLDA는 다음과 같은 방법들이 적용된다. 1) 차선을 2차 곡선으로 모델링한다. 2) LBPE(Lane Boundary Pixel Extractor)<sup>2,5)</sup>를 참고하여 픽셀의 밝기 기울기를 검출한다. 3) 원거리까지 차선 검출의 정확도를 높이기 위

해 BEV(Bird's Eye View)로 변환한 영상<sup>10,11,27)</sup>에서 차선 검출을 수행한다. 4) 기 검출된 차선들을 따라 L2GA의 토대가 되는 다수의 작은 원도우들을 배치하고, 원도우별로 픽셀의 밝기 기울기를 이용하여 차선 표시의 좌우 경계에 있을 것으로 기대되는 픽셀들을 추출한다. 5) 추출된 픽셀들로 적용 각도 범위가 제한된 HT(Hough Transform)<sup>25)</sup>를 수행하여 차선 대표점을 결정한다. 6) 결정된 대표점들에 랜덤 샘플 합의(RANSAC)<sup>13-15)</sup>와 최소 제곱법을 적용하여 차선 모델의 매개변수를 계산한다.

제안된 MLDA에 적용된 기법들 가운데 일부는 본 연구팀이 기 개발한 것들이 포함되어 있다. 그러나 이들을 다단계 필터링과 L2GA를 통해 비로 유발된 잡음의 영향을 줄일 수 있도록 융합하고, 우천시 주야에 촬영한 영상에서 다차로의 정보를 실시간으로 정확히 검출할 수 있도록 설계된 알고리즘은 찾아보기 어렵다.

이어지는 논문의 구성은 다음과 같다. 2장에서 관련 연구를 소개하고, 3장에서 본 연구에서 제안한 MLDA를 설명한다. 4장에서는 실험 결과를 제시하고, 5장에서 논문을 마무리한다.

## 2. 관련 연구

LDA(Line Detection Algorithm)는 규칙 기반 방법과 CNN 기반 방법으로 분류할 수 있다. CNN 기반 방식<sup>7-10)</sup>은 학습된 모델을 사용하여 영상에서 특징을 추출하고 추출된 특징을 토대로 차선을 감지한다. 이때 CNN 기반 알고리즘을 훈련하려면 TuSimple<sup>16)</sup>나 CULane<sup>17,18)</sup>과 같은 대규모 데이터 집합이 필요하다. 그런데 우천시와 같은 조건에서 생성한 GT 데이터는 크게 부족한 반면, 청명시의 GT 데이터는 풍부하다. 따라서 우천시 환경에 대해 적절하게 훈련되지 않은 CNN 기반 LDA는 우천시 촬영된 영상에서 성능이 떨어진다.

규칙 기반 LDA는 일반적으로 RoI(Region of Interest) 설정, RoI 내의 차선 픽셀 추출, 차선 모델 생성의 세 단계로 구성된다. RoI 설정은 계산 부하를 줄이고 잡음의 영향을 완화하는데 효과적이다. 일반적으로 RoI는 소실점 아래의 직사각형 영역이나 삼각형 모양으로 선택된다.<sup>4,15,19)</sup> 일부 연구에서는 소실선의 하단 영역을 더 작은 영역들로 분할하여 여러 RoI를 설정한다.<sup>4,5,20)</sup> 작은 RoI를 사용하면 각 RoI에 적용 임계값을 적용하여 이미지의 국소 영역에서 대비 차이를 극복할 수 있다. 또한 이전 프레임의 차선 검출 정보를 활용하여 RoI를 설정하는 사례들도 있다. 예를 들어, 작은 격자 기반 방법<sup>20)</sup>은 영상을 격자들로 나누고, 각 격자를 이전 프레임에서 차선이 감지된 경우는 True로 그렇지 않은 경우 False로 레이블링

한다. 또 다른 방식은 이전 프레임의 소실점과 차선 정보를 사용하여 람다( $\Delta$ ) 모양과 같이 감지된 차선의 모양과 유사한 RoI를 설정하는 것이다.<sup>21)</sup>

규칙 기반 LDA에서 차선 픽셀을 감지하는 기술 중에서 널리 채택된 접근 방식은 Sobel, Canny, Prewitt<sup>22,25)</sup>과 같은 에지(Edge) 연산자를 사용하는 것이다. 그러나 악천후 조건에서 잡음을 극복하기 위해 에지 감지에만 의존하는 데는 한계가 있다. 따라서 일부 방법은 차선 가시성을 높이기 위해 RGB의 각 채널에 다른 가중치를 지정하여 영상을 밝기 영상으로 변환한다.<sup>16,17,20)</sup> 또한 일부 연구에서는 에지 감지 중 잡음을 억제하기 위해 에지 감지에 EDLine(Edge Drawing Line)을 사용한다.<sup>21,24)</sup> 나아가 일부 방법은 에지 방향 정보를 활용하여 에지 감지 후 잡음을 제거한다.<sup>4,20)</sup> LBPE<sup>4)</sup>는 도로에 그려진 차선과 배경 사이의 특성을 활용하여 차선의 경계 픽셀을 추출하는 방법으로 제안되었다. 본 연구에서 제안하는 LDA는 LBPE의 개념을 기반으로 차선의 경계 픽셀을 추출하였다.

검출한 차선은 대개 직선 또는 곡선으로 표현한다. 차선을 직선으로 가정하는 연구는 주로 HT를 이용한다.<sup>2,26)</sup> 이 방법은 간단하고, 카메라에 가까운 근거리에서는 좋은 성능을 보이지만, 심하게 휜 차선 감지는 성능이 떨어진다. 반면에 차선을 곡선으로 가정하는 연구는 곡선 모델의 매개변수를 추정한다. 예를 들어, LCF(Lane Curve Function)<sup>3,29)</sup>는 카메라 근처 영역과 곡선의 변곡점을 넘어서는 영역을 나누어 곡선 매개변수를 추정한다. 또한 파티클 필터링을 이용하여 차선 정보 추적과 곡선 모델의 매개변수 추정을 동시에 하는 방법도 있다.<sup>5,18)</sup> 차선 정보 추적에 주로 칼만 필터가 사용되지만,<sup>21)</sup> 파티클 필터,<sup>5,12,28)</sup> 이동 평균,<sup>2)</sup> 선형 회귀<sup>4)</sup>도 사용된다.

### 3. 제안된 방법

본 연구에서 제안된 MLDA는 교통노면표시 설치·관리 매뉴얼<sup>6)</sup>을 참고하여 “차로와 차선은 구조화되어 있고, 그 위치와 방향은 점진적으로 변한다.”는 전제하에 다음과 같은 가정을 토대로 한다.

1. 차선은 노면보다 밝은 색이다.
2. 차선은 10 ~ 20 cm의 범위내에서 폭이 일정하다.
3. 차선은 2차 곡선으로 표현된다.
4. 차선은 국부적으로 직선이다.
5. 차선의 위치와 방향은 점진적으로 변한다.

이 다섯 가지 가정이 성립되지 않는 경우는 본 연구의 범위를 벗어난 것으로 간주한다.

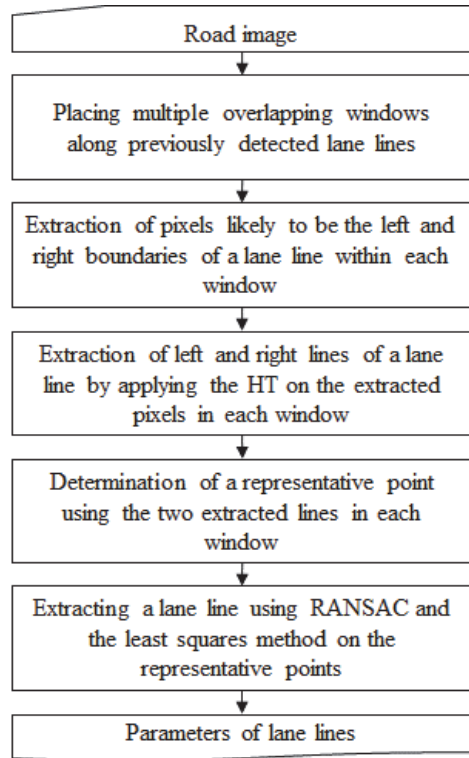


Fig. 1 Flow chart of the proposed MLDA

MLDA는 Fig. 1에 보인 바와 같이 5단계로 수행된다. 1) 연속적인 영상들에서 검출된 차선을 따라 다중의 윈도우를 상하로 서로 겹치게 설정한다. 2) 윈도우별로 차선 표시의 좌우 경계에 있을 것으로 기대되는 픽셀들을 다단계 필터링을 적용하여 검출한다. 3) 윈도우마다 검출된 픽셀들에 HT를 적용하여 차선 표시의 좌우 경계선을 추출한다. 4) 검출된 두 경계선 사이의 수평 중심점을 계산한다. 5) 계산된 수평중심점들에 RANSAC과 최소제곱법을 적용하여 이차 곡선으로 모델링된 차선을 생성한다.

MLDA는 우천시 촬영된 영상 전체에 무작위로 나타나는 잡음의 효과를 극복하고 다중 차선을 실시간으로 강건하게 검출한다.

#### 3.1 차선 모델링과 입력 영상의 BEV로의 변환

제안된 MLDA는 자차로와 자차로에 인접한 차로 등 3개 차로의 경계인 차선  $L_i(y)$ ,  $i=1,2,3,4$ 를 검출한다. 여기서 아래 첨자  $i$ 는 차선의 인덱스로 Fig. 2에 보인 바와 같이 2와 3은 자차로의 좌우 차선, 1과 2는 자차로의 좌 차로의 좌우 차선, 3과 4는 자차로의 우 차로의 좌우 차선을 의미한다.

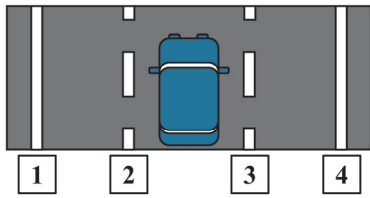
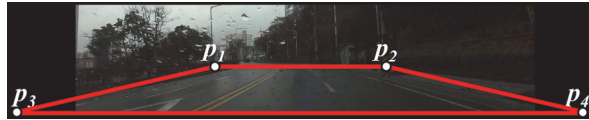
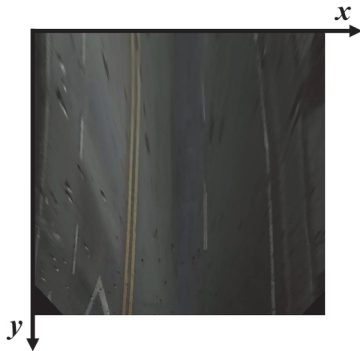


Fig. 2 The target lines for detection by the proposed LDA and their indices



(a) Selecting an area for converting an input image to BEV



(b) Conversion of the selected area from an input image to BEV

Fig. 3 Conversion of an input image to a BEV image

본 연구는 차선  $L_i(y)$ 를 2차 곡선으로 모델링한다.

$$L_i(y) = ay^2 + by + c, \quad i = 1, \dots, 4; \quad (1)$$

여기에서  $a, b, c$ 는 곡선의 매개변수를 나타낸다. 제안된 MLDA는 이 매개변수들을 정확하게 결정한다.

본 연구에서는 원거리의 차선까지도 강건하게 검출할 목적으로 Fig. 3에 보인 바와 같이 입력 영상을 BEV로 변환한다. 이 변환을 위해 먼저 Fig. 3(a)에 보인 것처럼 입력된 원 영상의 상단과 하단에 각각 2개씩 4개의 점  $p_1, p_2, p_3, p_4$ 를 지정한다. 이 후 네 개의 점으로 형성된 사다리꼴 영역을 원근 변환 매핑 방법을 사용하여 Fig. 3(b)에 보인 것처럼  $300 \times 300$  해상도의 BEV로 변환한다. 이 네 점을 선택할 때 검출 대상인 네 개의 차선이 모두 사다리꼴 내에 포함되도록 한다. 이상적으로는  $p_1$ 과  $p_2$ 는 소실선(Vanishing line)에 위치해야 하지만, 경우 시 소실선 가까이 있는 차선은 거의 보이지 않을 수 있어서  $p_1$ 과  $p_2$

는 소실선 아래로 오프셋을 주어 위치시킨다.  $p_3$ 와  $p_4$ 는 각각 영상의 좌우 경계를 넘어 20픽셀 떨어진 곳에 위치시킨다.

### 3.2 차선 표시 검출을 위한 에지 추출

차선 표시는 배경인 노면과 상당한 밝기 차이를 보인다.<sup>4,6)</sup> LBPE<sup>2,5)</sup>에서 했듯이 이런 속성을 가진 차선 검출을 위해  $3 \times 3$  Sobel 연산자<sup>25,26)</sup>를 영상  $f = (x, y)$ 에 적용하여 차선 표시(Mark)의 에지 픽셀을 검출한다. 이 검출을 위해 영상  $f(x, y)$ 의 픽셀  $(x, y)$ 에서 밝기 기울기  $\nabla f = [G_x, G_y]^T$ 의 수평 및 수직 요소인  $G_x$  및  $G_y$ 를 다음과 같이 얻는다.

$$G_x = f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1) - f(x-1, y-1) - 2f(x-1, y) - f(x-1, y+1), \quad (2)$$

$$G_y = f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1) - f(x-1, y-1) - 2f(x, y-1) - f(x+1, y-1), \quad (3)$$

$$\nabla f(x, y) = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|, \quad (4)$$

여기에서  $\nabla f = (x, y)$ 는  $\nabla f = [G_x, G_y]^T$ 의 크기이다.

### 3.3 L2GA 구현을 위한 다중 윈도우 배치

우천시 촬영된 영상에는 영상 전체에 무작위로 나타나는 빗방울, 노면상의 물고임, 노면 젖음 등으로 인해 차선이 가려지거나 차선과 유사한 패턴들이 생긴다. 이런 노면에 빛이 투사되면 빛의 산란이 심해 영상처리하는 물론이고 육안으로도 차선 식별이 쉽지 않다. 그러나 도로의 실제 차선은 이러한 잡음과 무관하게 변화가 없다.

따라서 영상 전체가 아닌 국부적으로 작은 영역 단위로 차선의 속성을 유지하는 곳을 탐색하고, 차선 속성이 유지되는 영역들을 통합하여 차선을 검출하는 L2GA를 고안했다. L2GA에서는 먼저 이전 영상들에서 검출된 차선들 각각을 따라 동일한 수량의 작은 사각형 윈도우들을 배치하고, 윈도우별로 차선의 속성을 유지하는 픽셀들의 존재 여부를 확인한다.

가정 5에 따라 현재 영상  $f_t(x, y)$  이전의 세 개의 연속된 영상  $f_{t-1}(x, y)$ ,  $f_{t-2}(x, y)$  및  $f_{t-3}(x, y)$ 에서 검출된 차선들  $L_i^{t-k}(y), i = 1, \dots, 4$  and  $k = 1, 2, 3$ 를  $f_t(x, y)$ 로 전송한다. 여기서 첨자  $t$ 는 현재를 의미하며, 이전 세 영상의 사용은 실험적으로 정한 것이다. 영상  $f_t(x, y)$ 에 윈도우 배치를 위한 안내 선  $l_i(y)$ 는 식 (5)를 사용하여 정의한다.

$$l_i(y) = \bar{a}_i y^2 + \bar{b}_i y + \bar{c}_i, \quad i = 1, \dots, 4 \quad (5)$$

여기에서

$$\bar{a}_i = \frac{1}{3} \sum_{k=1}^3 a_i^{t-k}, \quad \bar{b}_i = \frac{1}{3} \sum_{k=1}^3 b_i^{t-k}, \quad \bar{c}_i = \frac{1}{3} \sum_{k=1}^3 c_i^{t-k}.$$

이와 같이 안내 선  $l_i(y)$  를 정의하고  $W_x \times H_y$  크기의 사각형 윈도우들을  $l_i(y)$  별로 BEV 영상의 수직축인  $y$ 축을 따라  $H_y/2$ 의 보폭으로 배치한다. 이때 윈도우를 수직 방향으로 절반씩 중첩시킨 것은 빗방울과 같은 잡음이 차선 표식으로 오인되는 경우가 있는데, 이러한 상황을 해결하기 위한 것이다. 이 현상의 사례는 4장의 실험 결과에 제시하였다.

### 3.4 차선 경계 픽셀 검출을 위한 다단계 필터링

배치된 윈도우내에 차선의 속성을 만족하는 표식이 존재하면, 그 표식은 차선일 가능성이 있다. 따라서 가정 1에 의거하여 그런 표식의 경계에 있을 법한 픽셀들을 LBPE에 적용된 아이디어를 활용하여 탐색한다. 밝기값의 평균과 표준편차를 계산하고, 식 (4)에 의해 계산된 밝기 기울기 크기의 평균과 표준편차를 계산하여 식 (6)을 통해 차선의 경계일 가능성이 있는 픽셀을 모은다.

$$M = \{(x, y) | f(x, y) > \mu_f + \lambda_f \sigma_f \text{ and } \nabla f(x, y) > \mu_{\nabla f} + \lambda_{\nabla f} \sigma_{\nabla f}\} \quad (6)$$

여기에서  $\mu_f$  와  $\sigma_f$  는 밝기값에 대한 평균과 표준편차이며,  $\mu_{\nabla f}$  와  $\sigma_{\nabla f}$  는 밝기 기울기 크기에 대한 평균과 표준편차이다.  $\lambda_f$  와  $\lambda_{\nabla f}$  는 각각  $\sigma_f$  와  $\sigma_{\nabla f}$  에 대한 가중치로서 실험적으로 0.4와 0.5를 부여했다.

집합  $M$ 에 속한 픽셀들은 차선 표식의 경계에 있을 가능성이 있는 픽셀들로서 이들을 식 (2)에 의해 계산된 밝기 기울기의 수평 요소인  $G_x$ 를 이용하여 좌우 경계로 분리한다.

$$B_L = \{(x, y) | (x, y) \in M \text{ and } G_x > 0\}, \quad (7)$$

$$B_R = \{(x, y) | (x, y) \in M \text{ and } G_x < 0\}, \quad (8)$$

여기서 아래 첨자  $L$ 과  $R$ 은 좌와 우를 의미한다. Fig. 4(a)는 식 (7)과 (8)의 근거가 되는 차선 표식의 속성을 보이는 것이다.

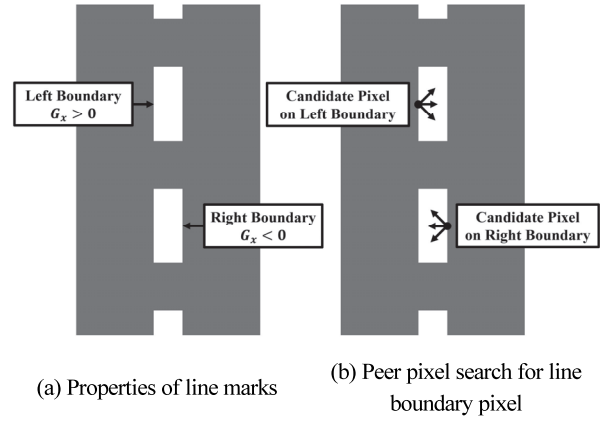


Fig. 4 Properties of line marks and extraction of left and right boundary pixels of line marks

집합  $B_L$ 과  $B_R$ 에는 차선이 아닌 곳에서 추출된 픽셀도 포함될 수 있다. 이런 픽셀 제거에 가정 2를 이용한다. 가정 2에 의하면  $B_L$ 에 속한 픽셀 우측으로 차선 폭 범위 내에서  $B_R$ 에 속한 픽셀이 있어야 하고, 동일하게  $B_R$ 에 속한 픽셀 좌측으로 차선 폭 범위 내에서  $B_L$ 에 속한 픽셀이 있어야 한다. 이 진제를 토대로  $B_L$ 과  $B_R$ 에 속해서는 안될 픽셀들을 제거한다. 본 연구에서는 이 과정을 “동료 픽셀 탐색”으로 명명했으며 이 탐색을 수평 방향으로만 국한하지 않는다. 차선 표식이 부분적으로 마모되거나 가려질 수 있고, 비에 의한 잡음으로 인해 차선의 일부가 희미해지거나 흐릿해질 수도 있어서 유연한 탐색이 필요하다. 따라서 Fig. 4(b)에 화살표로 표시한 대로 수평, 위쪽 대각선, 아래쪽 대각선 이렇게 세 방향으로 동료 픽셀을 탐색한다.  $B_R$ 에 속한 픽셀  $p_r(x, y)$ 의 좌측으로 이 세 방향 중 적어도 한 방향에  $B_L$ 에 속한 픽셀이 존재하면 픽셀  $p_r(x, y)$ 는 차선 경계 픽셀로 간주한다. 그러나 세 방향 중 어느 방향에도  $B_L$ 에 속한 픽셀이 없다면  $p_r(x, y)$ 는  $B_R$ 에서 제거한다. 동일한 절차를  $B_L$ 에 속한 픽셀  $p_l(x, y)$ 에 대해서도 적용한다. 단  $p_l(x, y)$ 의 동료 픽셀 탐색 방향은 우측이 된다. Fig. 4(b)에 표시된 탐색 범위는 일반적인 차선 너비인 20 cm로 설정했다. 집합  $B_L$ 과  $B_R$ 에 동료 픽셀 탐색 과정 적용 후 갱신된 집합을 각각  $B_L''$ 과  $B_R''$ 으로 표현한다.

가정 1에 의거하여 집합  $B_L''$ 과  $B_R''$ 에 속한 픽셀들을 다음과 같이 한 차례 더 필터링하여 실제 차선 표식의 경계에 있을 가능성이 큰 픽셀들을 추출한다.

$$B_L^N = \{(x, y) | (x, y) \in B_L'' \text{ and } Q_L > \tau\}, \text{ where} \quad (9)$$

$$Q_L = \sum_{i=1}^W f(x+i, y) - \sum_{i=1}^W f(x-i, y).$$

$$B_R^N = \{(x, y) | (x, y) \in B_R^W \text{ and } Q_R > \tau\}, \text{ where}$$

$$Q_R = \sum_{i=1}^W f(x-i, y) - \sum_{i=1}^W f(x+i, y). \quad (10)$$

여기에서  $W$ 는 차선 표식의 폭에 해당되는 픽셀거리를 나타내고,  $\tau$ 는 차선과 노면 사이의 밝기 차이가 충분히 큰지를 판단하는 임계값으로 실험적으로  $\sigma_f \times W/2$ 로 결정했다. 여기서  $\sigma_f$ 는 식 (6)에서 계산된 밝기값의 표준편차다. 또한  $W$ 결정에 사용한 차선폭은 실제 차선폭의 최소값인 10 cm로 정했다.

### 3.5 하프변환을 활용한 예비 차선의 대표점 결정

가정 4에 따라 차선은 국부적으로 직선이므로 윈도우별로  $B_L^N$ 과  $B_R^N$  각각에 HT를 적용하여 차선 표식의 좌우 경계에 해당하는 직선을 추출할 수 있다. 이때 대개 Hough 공간에서 최대 보팅(Voting)이 이루어진 셀의 좌표를 사용한다.<sup>26)</sup> 그러나 이 방식으로 결정된 직선은 잡음의 영향으로 인해 가정 5를 위반할 수 있다. 특히 본 연구는 우천시 촬영한 영상에서 MLDA를 구현하므로 비로 인한 잡음의 영향이 커서 이 문제를 극복하는 것이 매우 중요하다.

이 문제 해결을 위해 취한 방안은 HT 구현시 적용하는 각도의 범위를 식 (5)로 얻은 참조선의 기울기를 활용하여 제한하는 것이다. 각 윈도우의 수직 중심 좌표를  $y'$ 라 할 때,  $y'$ 에서 참조선의 기울기는  $2\bar{a}y'+\bar{b}$ 가 된다. 이 기울기를 x-축을 기준으로 정의된 각도  $\theta_{slope}$ 로 변환하고  $\theta_{slope}$ 에  $90^\circ$ 를 더하여 참조선의 접선에 수직인 선의 각도  $\theta_{ht}$ 를 구한 후,  $x\cos\theta + y\sin\theta = \rho$ 의 HT 수식<sup>25)</sup>의 각도( $\theta$ ) 조건에 맞춘다. 이때 가정 5를 고려하여 현재 영상에서 차선의 방향이 참조선에 대하여  $\pm\alpha$  내에서 변할 수 있다고 가정하고,  $\theta_{ht} - \alpha$ 에서  $\theta_{ht} + \alpha$ 의 각도 범위에서 HT를 수행한다. 그러나 이 각도 범위가 HT 적용 각도 범위인  $0^\circ \sim 180^\circ$ 를 벗어날 수 있어서 식 (11)과 같이 설정하여 HT를 수행한다. 이때  $\alpha$ 를 실험적으로  $10^\circ$ 로 정했다.

$$\begin{cases} 0 \leq \theta \leq \theta_{\max} & \text{and } 180 - |\theta_{\min}| \leq \theta < 180 & \text{if } \theta_{\min} < 0 \\ 0 \leq \theta \leq \theta_{\max} - 180 & \text{and } \theta_{\min} \leq \theta < 180 & \text{if } \theta_{\max} > 180 \\ \theta_{\min} \leq \theta \leq \theta_{\max} & & \text{otherwise} \end{cases} \quad (11)$$

여기에서  $\theta_{\min} = \theta_{ht} - \alpha$ ,  $\theta_{\max} = \theta_{ht} + \alpha$ ,  $\theta_{ht} = \theta_{slope} + 90^\circ$

$$\theta_{slope} = \tan^{-1}\left(\frac{1}{2\bar{a}y'+\bar{b}}\right).$$

식 (11)을 적용하여 HT를 수행한 후, Hough 공간에 최대 보팅된 셀의 좌표를 사용하여 차선 표식 경계에 해당

하는 직선을 찾는다. 이때 최대 보팅 값이 10 미만이면 직선 추출을 하지 않는다. 여기서 임계값 10은 실험적으로 정했다.

다음은 윈도우 내에서 차선의 위치를 대표하는 점을 계산하고, 이 점을 대표점이라 한다. 대표점의 수직 위치는 윈도우의 수직 중심인  $y'$ 으로 정하고, 수평 위치  $x'$ 은 앞에서 HT로 얻은 직선을 이용하여 정한다. HT의 결과는  $B_L^N$ 과  $B_R^N$  모두에서 선이 검출되었거나, 둘 중 하나에서만 검출되었거나, 어느 쪽에서도 검출되지 않은 경우가 있다. 첫 번째 경우의 대표점의 수평 위치  $x'$ 은 두 직선의  $y'$ 에 해당하는 수평 좌표 값의 평균으로 계산된다. 두 번째 경우의  $x'$ 은 검출된 직선의  $y'$ 에 해당하는 수평 좌표를 Fig. 4(b)에 설명된 동료 픽셀 탐색 과정에서 적용된 검색 범위의 절반을 직선이 검출되지 않은 쪽으로 이동한 위치로 정한다. 세 번째 경우는 대표점을 결정하지 않는다. 즉 차선 표식의 속성을 만족하는 픽셀들이 존재하지 않거나 소수인 윈도우는 차선 정보 추출 대상에서 제외한다.

### 3.6 RANSAC과 최소 제곱법에 의한 차선 검출

지금까지 다중 윈도우를 이용하여 영상의 로컬 영역에서 차선의 속성을 만족한 영역들을 탐색했다. 이제 이러한 영역들을 통합하여 차선을 검출한다. 윈도우별로 검출된 대표점들을 대상으로 식 (1)로 정의된 이차 곡선의 매개변수  $a, b, c$ 를 결정함으로써 차선검출이 이루어지는데, 이 결정에 RANSAC과 최소 제곱법을 적용한다.

RANSAC 적용에 앞서 자차로의 좌우 차선에 대한 대표점들의 수가 8개 미만이면 차선을 정확하게 추출할 수 없다고 간주하고, 자차로 차선의 매개변수는 이전 영상  $f_{-1}(x, y)$ 에서 추정된 매개변수로 대체한다. 그러나 동일한 상황이 자차로에 인접한 차로의 차선 1과 4 각각에 발생했을 때는 차선 1과 4는 없는 것으로 결론짓고, 해당 차선의 매개변수는 정하지 않는다. 여기에서 문턱치 8은 실험적으로 설정했다.

이때 RANSAC 적용에 있어 추출된 대표점들 내의 이상치(Outlier)의 비율은 알 수 없으므로 모델 가설을 생성하는 반복 횟수(N)를 실험을 통해 100으로 설정하고, 식 (1)에서 언급한 4개의 차선 각각에 대해 다음과 같이 RANSAC을 수행한다. 각 반복에서 차선 매개변수의 가설은 추출된 대표점들 중에서 무작위로 샘플링한 세 개의 점들로 설정한다. 이 가설의 매개변수를  $\tilde{a}, \tilde{b}, \tilde{c}$ 로 표시한다. 이어서,  $\tilde{a}, \tilde{b}, \tilde{c}$ 와 식 (5)에서 계산된 참조선의 매개변수  $\bar{a}, \bar{b}, \bar{c}$ 의 차이를 식 (12)에 의해 계산하고, 이 차이가 미리 정해진 임계값을 초과하면 이 가설에 대한 인라

이어(Inlier) 선택을 하지 않고 새로운 가설을 세운다.

$$T_1 = |(\tilde{a}y_s^2 + \tilde{b}y_s + \tilde{c}) - (\bar{a}y_s^2 + \bar{b}y_s + \bar{c})|,$$

$$T_2 = |(\tilde{a}y_e^2 + \tilde{b}y_e + \tilde{c}) - (\bar{a}y_e^2 + \bar{b}y_e + \bar{c})|,$$

If  $T_1 > \tau_c$  or  $T_2 > \tau_p$ , stop; otherwise, continue, (12)

여기에서  $y_s$ 와  $y_e$ 는 각각 BEV 영상의 상단과 하단 행의 위치인데, BEV 영상의 크기를 300×300으로 정했으므로  $y_s$ 와  $y_e$ 는 각각 0과 299이다. 하단에서는 차선의 위치가 좌우로 심하게 변화지 않으므로 임계값  $\tau_c$ 는 일반적인 차선 폭 20 cm의 두 배인 40 cm의 픽셀 거리로 설정하고, 상단에서는 차선의 위치 변화가 하단에 비해 더 크다는 점을 고려하여 임계값  $\tau_p$ 는 1 m에 대한 픽셀 거리로 설정했다. 즉 임계값  $\tau_c$ 와  $\tau_p$ 는 실험적으로 정한 것이다.

현 가설에 대한 인라이어 선택을 다음과 같이 수행한다. 식 (1)의 매개변수  $a, b, c$ 를 현 가설의 매개변수  $\tilde{a}, \tilde{b}, \tilde{c}$ 로 대체하고, 대표점 각각의 수직좌표  $y'$ 을 식 (1)에 대입하여 계산된 값을  $\tilde{x}$ 라 한다. 이때  $\tilde{x}$ 와 대표점의 수평 좌표  $x'$ 의 절대 차이가 문턱치  $\epsilon$  이하면 해당 대표점을 인라이어로 선택한다. 여기에서  $\epsilon$ 은 차선폭 10 cm에 대한 픽셀 거리로 정했다.

RANSAC을 적용하는 많은 방법들이 이러한 과정을  $N$  번 반복한 후, 가장 많은 인라이어가 선택된 가설의 매개변수를 모델의 파라미터로 선정한다.<sup>14,15)</sup> 따라서  $\tilde{a}, \tilde{b}, \tilde{c}$ 를 차선  $L_i(y)$ 의 매개변수로 정할 수 있다. 그러나 차선 검출의 정확도를 높이기 위해 가장 많은 대표점들이 선택된 가설의 대표점들을 대상으로 최소 제곱법을 적용하여  $L_i(y)$ 의 매개변수를 결정한다. 대표점  $(x', y')$  각각은 다음과 같이 방정식으로 표현할 수 있다.

$$ay_k'^2 + by_k' + c = x_k', k = 1, \dots, K \quad (13)$$

여기에서  $K$ 는 인라이어로 선택된 대표점들의 수량이다. 식 (13)으로 표현된 방정식 시스템은 식 (14)와 같이 행렬 방정식으로 표현된다.

$$\underbrace{\begin{bmatrix} y_1'^2 & y_1' & 1 \\ y_2'^2 & y_2' & 1 \\ \vdots & \vdots & \vdots \\ y_K'^2 & y_K' & 1 \end{bmatrix}}_A \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_X = \underbrace{\begin{bmatrix} x_1' \\ x_2' \\ \vdots \\ x_K' \end{bmatrix}}_C. \quad (14)$$

여기서  $A$ 는  $K \times 3$  계수 행렬이고,  $X$ 는  $3 \times 1$  매개변수 행렬이며,  $C$ 는  $K \times 1$  행렬이다.  $L_i(y)$ 의 매개변수는 의사 역행렬을 통해 결정된다.

$$X = (A^T A)^{-1} A^T C, \quad (15)$$

여기에서 위첨자  $T$ 는 행렬의 전치를 의미한다.

#### 4. 실험결과

이 장에 제시된 실험 결과는 우천시와 청명시 주야에 실험차량으로 도로를 주행하면서 촬영한 영상에 본 연구에서 제안된 MLDA를 적용하여 얻었다. 이 실험을 위해 차량 실내의 중앙에 카메라를 설치하고, 전남 광양의 도로에서 우천시에 해상도 1,920×448의 영상 1,000장을 획득했다. 이 영상 모음을 광양 데이터 집합이라 한다.

제안된 MLDA 평가를 위해 CULane<sup>17,18)</sup> 데이터 집합에 사용된 방법과 유사하게 광양 데이터 집합의 영상에 차선 주석 달기(Annotating)를 하여 GT 데이터를 얻었다. 한편, CULane에 포함된 영상들은 촬영 간격이 1초 또는 3초로 길었기 때문에 영상 간 차선 정보의 변화가 컸다. 따라서 CULane 데이터 집합의 영상을 사용하여 제안된 MLDA를 평가할 수 없었다. 반면에 광양 데이터 집합의 영상은 15fps(frame per second) 이상의 속도로 촬영되어 MLDA를 평가하는데 아무런 문제가 없었다.

##### 4.1 제안된 MLDA의 구현 과정

MLDA는 현 영상  $f_i(x, y)$  이전의 세 개의 연속된 영상에서 추출한 차선을  $f_i(x, y)$ 에 전송하여  $L_i(y)$ 에 대한 참조 선  $l_i(y)$ 를 생성한다. 여기서 첨자  $i=1, 2, 3, 4$ 이다. Fig. 5는  $l_i(y)$  생성과 윈도우 배치의 예를 보인 것이다. Fig. 5(a)의 주황색, 녹색 및 파란색 선은  $f_i(x, y)$ 에 앞선 세 개의 연속 영상에서 추출한 차선을 나타낸 것이다. Fig. 5(b)의 네 개의 빨간색 선이 식 (5)로 계산된  $l_i(y)$ 를 나타낸다. Fig. 5(b)는 또한  $l_i(y)$ 를 따라 윈도우(그림에서 흰색 사각형)들을 절반씩 겹쳐서 배치한 결과도 보였다. BEV 영상의 해상도가 300×300이므로  $l_i(y)$ 마다 41×31 크기의 윈도우 18개가 배치되었다. 각  $l_i(y)$ 의 첫 윈도우의 중심은 BEV 영상 수직축 좌표 20에 일치시켰다.

Fig. 6은 윈도우 내에서 다단계 필터링을 통해 차선 표시의 좌우 경계에 있을 것 같은 픽셀들의 추출과 HT에 의한 차선 표시의 경계선 검출 과정을 제시한다. Fig. 6의 좌측 상단에 화살표로 가리키진 흰색 사각형이 윈도우

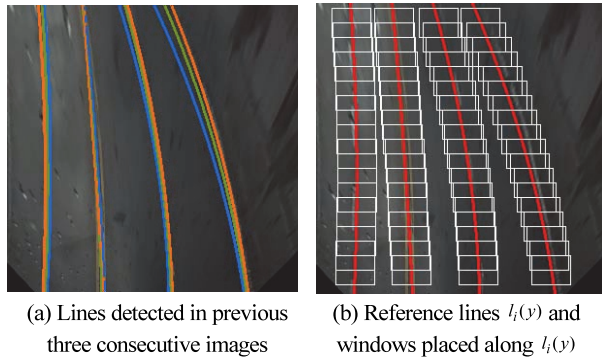


Fig. 5 Creation of reference lines and windows placed along them

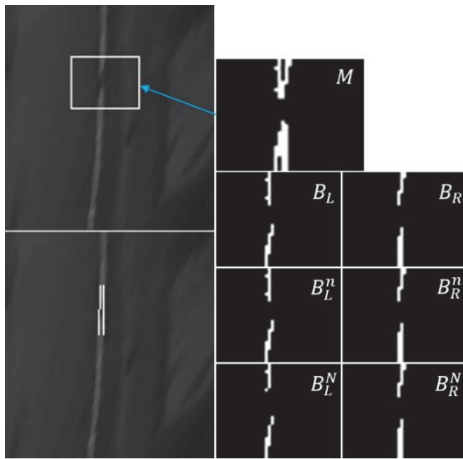
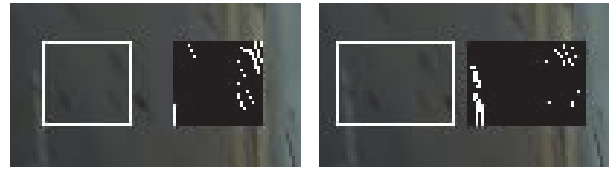


Fig. 6 Extraction of the left and right boundary pixels and the corresponding boundary lines of a lane mark within a window

다. Fig. 6의 우측 맨 상단의 흑백 이진 영상은 이 윈도우 내에서 식 (6)에 의해 구축된 집합  $M$ 을 나타낸 것이다. Fig. 6의 우측 두 번째 행의 두 흑백 영상 각각은 식 (7)과 (8)에 의해 집합  $M$ 을 차선 표시의 좌우 경계에 있을 픽셀들로 분리하여 구축한 집합  $B_L$ 과  $B_R$ 을 나타낸다. Fig. 6의 우측 세 번째 행의 두 흑백 영상 각각은 Fig. 4(b)로 설명된 동료 픽셀 탐색을 집합  $B_L$ 과  $B_R$ 에 적용하여 구축된 집합  $B_L^N$ 과  $B_R^N$ 을 나타낸다. Fig. 6의 우측 네 번째 행의 두 흑백 영상 각각은 식 (9)와 (10)으로 설명된 필터링을  $B_L^N$ 과  $B_R^N$ 에 적용하여 구축한 집합  $B_L^N$ 과  $B_R^N$ 을 나타낸다. Fig. 6의 좌측 하단 영상에 그려진 두 흰색 선은  $B_L^N$ 과  $B_R^N$ 에 식 (11)을 적용하여 HT를 수행해 얻은 직선을 그린 것이다.

Fig. 5(b)에 보인 바와 같이 생성된  $l_i(y)$ 를 따라 수평 폭이 수직 높이보다 더 큰 윈도우들을 수직으로 절반씩



(a) A square window of 31x31 (b) A rectangular window of 41x31

Fig. 7 Basis for setting the horizontal width of a window larger than its height

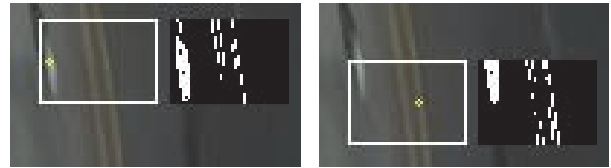


Fig. 8 Basis for arranging windows to overlap vertically by half

겹쳐 배치했다. 이러한 접근은 L2GA의 약점을 보완하기 위한 것이었다. Fig. 7과 8은 윈도우의 모양과 배치를 Fig. 5(b)에 보인 것처럼 정한 근거가 된 사례를 보인 것이다. 영상 전체적으로 무작위로 나타나는 비에 의한 잡음의 영향을 줄이기 위해 작은 영역들을 배치했지만, 작은 영역이기 때문에 잡음에 취약해질 가능성도 상존한다. Fig. 7은 윈도우의 수평폭을 더 크게 정한 근거를 설명할 수 있는 사례를 보인 것이다. Fig. 7(a)와 (b)에 흰색으로 표시된 사각형이 윈도우인데, Fig. 7(b)의 윈도우가 Fig. 7(a)의 윈도우보다 수평 폭이 더 크다. 두 윈도우 모두 동일 영상과 동일 중심 좌표에 설정했다. 이 윈도우들 내에서 차선의 경계일 것으로 기대되어 추출된 픽셀들은 각 윈도우 우측에 이진 영상으로 표시했다. 이 예에서 실제 차선은 윈도우내의 좌측에 있는데, Fig. 7(a)의 윈도우는 수평폭이 좁기 때문에 차선의 일부만 윈도우에 포함된 반면에 Fig. 7(b)의 윈도우는 수평 폭이 넓어 차선의 많은 부분이 윈도우에 포함되었고, 따라서 검출된 픽셀들도 더 많았다.

Fig. 8은 윈도우를 수직방향으로 절반씩 겹친 근거를 설명할 수 있는 사례를 보인 것이다. Fig. 8의 좌우 영상에 흰색으로 그려진 사각형은 동일 영상에서 참조 차선  $l_i(y)$ 를 따라 수직으로 절반씩 겹친 윈도우다. 이 두 윈도우 내에 표시된 노란색 점은 차선의 대표점이다. 각 윈도우내에서 차선 경계일 것으로 기대되어 추출된 픽셀들은 윈도우 우측 이진 영상에 흰색으로 표시했다. 왼쪽 윈도우내에는 실험차량의 윈드쉴드에 맞힌 빔방울 영역의 많은 부분이 포함되었고, 이 부분이 차선의 속성을 만족했으며 윈도우 중앙에 보인 노란색의 실제 차선보다

더 밝았다. 따라서 이 빗방울 영역에서 차선의 대표점이 추출되었다. 반면에 우측 윈도우에는 동일 빗방울 영역의 일부만 포함되어 실제 차선 경계에서 더 많은 픽셀들이 추출되어 실제 차선에서 대표점이 추출되었다. 이와 같이 일부 윈도우에서 비로 인한 잡음이 차선으로 오인식되더라도 윈도우를 겹치면 이러한 오류의 영향을 완화할 수 있다.

Fig. 9는 윈도우에서 차선 대표점을 얻기 위해 수행한 HT의 각도 범위를 식 (11)에 의해 제한했을 때의 효과를 보인 예다. Fig. 9에 표시된 흰색 윈도우에는 실제 차선은 없고, 차량의 윈드실드에 맺힌 빗방울이 유사 차선으로 보인다. Fig. 9의 왼쪽은 식 (11)의 적용없이 HT를 수행하여 결정한 차선 대표점을 보인 것이다. 반면에 Fig. 9의 우측은 식 (11)을 적용하여 HT를 수행한 결과 차선 대표점이 결정되지 않은 결과를 보인 것이다. 즉 식 (11)에

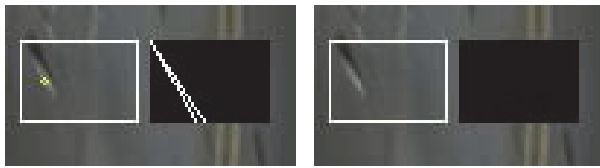
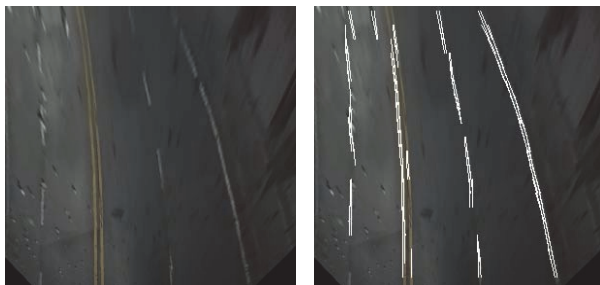


Fig. 9 Prevention of representative point determination errors using the slope of  $l_i(y)$



(a) BEV image

(b) Line segments extracted by HT



(c) Representative points

(d) Extracted lines

Fig. 10 Extraction of representative points and line detection using RANSAC and least squares method

의해 오류를 막은 것이다.

Fig. 10은 RANSAC과 최소 제곱법을 적용하여 차선을 검지한 예를 보인 것이다. 본 연구에서 설계한 L2GA는 각 윈도우에서 대표점 결정까지는 로컬 접근이고, 결정된 대표점들에 RANSAC과 최소 제곱법을 적용하는 것은 글로벌 접근이다. Fig. 10(b)에 보인 흰 선분들은 각 윈도우 내에서 HT를 적용하여 얻은 것이다. Fig. 10(c)에 보인 노란색 점들은 Fig. 10(b)에 보인 선분들로부터 결정된 대표점들이다. Fig. 10(d)는 이런 대표점들에 RANSAC 및 최소 제곱법을 적용하여 검지한 차선을 보인 것이다. Fig. 10(a) ~ (c)의 상단 좌측을 보면 빗방울 영역이 유사 차선으로 보여 차선이 아닌데도 대표점이 추출된 것을 알 수 있다. 그러나 Fig. 10(d)에서 볼 수 있듯이 제안된 MLDA는 이런 잡음의 영향을 극복했다.

#### 4.2 제안된 MLDA의 정량적 평가

제안된 MLDA의 평가에 정밀도를 사용했다. 이를 위해 윈도우에서 추출된 대표점과 실제 차선의 해당 점 사이의 수평 거리를 계산하고, 계산된 수평 거리가 보통 차선의 최대 폭인 20 cm 미만인지 확인했다. 이 계산에서 실제 차선의 점은 윈도우에서 추출된 대표점과 동일한 수직 좌표를 가진 점으로 제한된다. 계산된 수평 거리가 20 cm 미만이면 대표점은 참 양성(TP)으로 분류하고, 그렇지 않으면 거짓 양성(FP)으로 분류한다. 여기서 정밀도는 TP 수를 (TP 수+ FP 수)로 나눈 값으로 정의된다.

Table 1은 광양 데이터 집합으로 분석한 정밀도를 보인 것이다. 이 실험에서 RANSAC에 대한 세 가지 다른 반복 횟수인 100, 200, 300을 설정하고, 각 경우에 대한 정밀도를 계산했다. Table 1에서 “ $\ominus$ ” 열의 기호  $\checkmark$ 는 대표점 결정을 위해 수행된 HT에 식 (11)이 적용되었음을 뜻한다. Table 1에 제시된 결과로부터 HT에 각도 범위를 제한하면 FP 수가 줄어든다는 것과 RANSAC 반복 횟수가 MLDA의 정밀도에 큰 영향을 미치지 않는다는 것을 알 수 있다. 따라서 RANSAC의 반복 횟수를 100으로 정했다.

Table 1 Precision of our MLDA

# of RANSAC iteration	$\ominus$	TP	FP	Precision (%)
100		30877	5417	85.075
	$\checkmark$	30221	3367	89.976
200		31373	5331	85.476
	$\checkmark$	30415	3255	90.333
300		31696	5168	85.980
	$\checkmark$	30171	3384	89.915

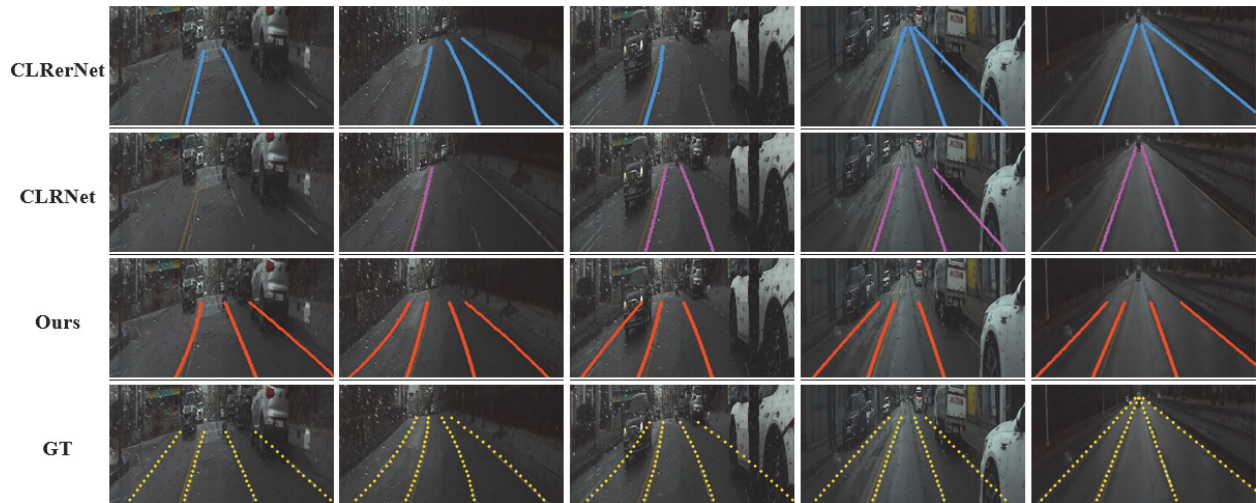


Fig. 11 Performance comparison of our MLDA and other methods

Table 2 Accuracy comparison of our MLDA with other methods

Methods	Precision	Recall	F1
CLRerNet	89.108	67.007	76.493
CLRNet	96.829	60.221	74.257
Ours	87.979	83.821	85.850

이어서 F1 점수를 사용하여 제안된 MLDA의 성능을 CLRerNet<sup>9)</sup> 및 CLRNet<sup>10)</sup>과 비교했다. 이 비교를 위해 여기에서 언급된 세 가지 방법으로 검출된 차선들 각각에 대해 수직 방향으로 10픽셀 간격으로 30개의 픽셀을 선택했다. 선택된 이 픽셀들을 Table 1에 제시된 데이터를 얻는 데 적용했던 방법을 사용하여 TP와 FP로 분류한 후, 식 (16)을 이용해 F1 점수를 계산했다.

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \quad (16)$$

여기에서 재현율(Recall)은 TP의 수를 GT의 수로 나눈 값으로 정의된다.

F1 척도는 대개 정밀도와 재현율의 균형을 맞춰 모델의 성능을 평가하는 데 사용된다. Table 2에 제시된 결과를 보면 본 연구에서 제안된 방법이 정밀도와 재현율 각각 87.979 %와 83.821 %였고, F1 점수는 85.850이었다. Table 2를 보면 CLRerNet과 CLRNet의 재현율이 낮는데, 이는 차선 감지에 실패한 경우가 많았음을 의미한다.

Fig. 11은 Table 2에 제시된 결과를 얻는데 사용된 광양 데이터 집합에 속한 몇 가지 영상에서 차선을 검출한 결과를 보인 것이다. Fig. 11에서 볼 수 있듯이 본 연구에서

제안된 방법은 CLRerNet 및 CLRNet에 비해 일관되게 더 많은 차선을 검출한 반면에 다른 두 방법은 검출에 실패한 경우가 많았다.

### 4.3 청명시 촬영된 영상에 적용된 제안된 MLDA

Fig. 12는 MLDA가 청명시 주야에 촬영한 영상에서도 차선을 강건하게 검출한다는 것을 보여준다. Fig. 12의 처음 두 경우는 낮에 촬영한 영상에, 나머지 두 경우는 밤에 촬영한 영상에 대한 결과다. Fig. 12(a) ~ (c)는 BEV 영상에 MLDA를 수행한 결과를 보인 것이고, Fig. 12(d)는 원본 영상에다 보인 것이다. Fig. 12의 세 번째 사례에서는 차선 규제봉들이 차선 옆에 설치되어 있어도 MLDA가 차선을 정확히 검출했다. Fig. 12는 제안한 MLDA가 주야 관계없이 청명시 촬영한 영상에서 차선을 강건하게 검출할 수 있음을 보여주었다.

### 4.4 제안된 MLDA의 처리시간

제안된 MLDA의 실행은 Intel i7-6700HQ @ 2.60GHz 프로세서와 NVIDIA Quadro M1000M GPU가 장착된 시스템에서 이루어졌다. Table 3의 각 프로세스에 대한 시간은 총 처리 시간을 평가에 사용된 영상 수로 나눈 값이다. 이 실험에서는 Table 1에 제시된 데이터를 얻는 데 사용된 것과 동일한 1,000개의 영상을 사용했다. Table 1에서 보인 것처럼 RANSAC 반복 횟수는 MLDA의 정밀도에는 큰 영향을 미치지 않지만 처리 시간에는 상당한 영향을 미친다. Table 3에서 “픽셀 필터링”이라는 용어는  $B_L^N$  과  $B_R^N$  을 얻는데 까지 프로세스를 나타낸다. Table 3의 “HT”는  $B_L^N$  과  $B_R^N$  에 하프변환을 적용하여 차선의 좌우 경계에 해당하는 직선을 추출하는 것을 나타내는

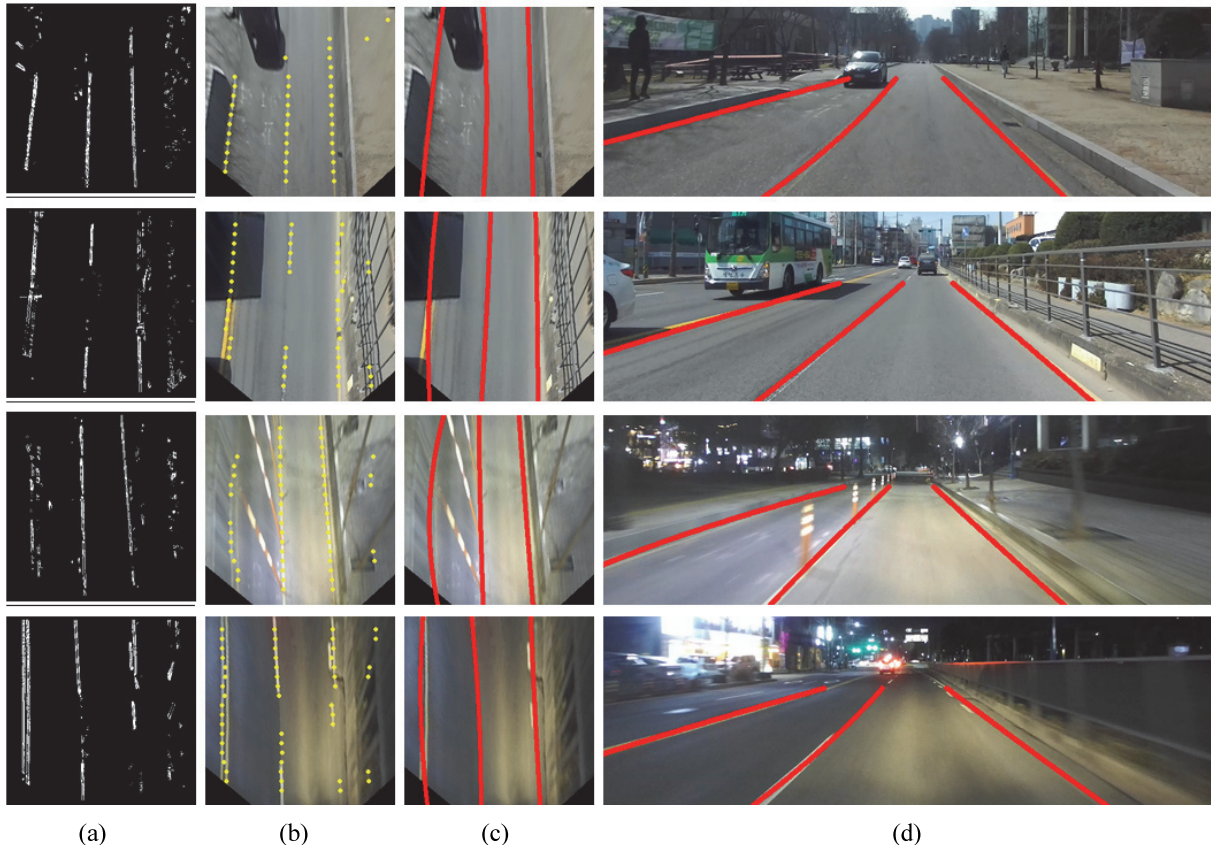


Fig. 12 Results of the proposed LLDA applied to images captured day and night in clear weather: (a) Extracted pixels on the boundaries of lane-line markings; (b) Extracted representative points of lane-lines; (c) Predicted lane-lines on BEV images; (d) Predicted lane-lines on original images.

Table 3 Processing time for each process of our MLDA

Process	Average processing time (ms)
Pixel filtering	2.011
Hough transform	14.136
Hough transform+	3.475
RANSAC (100 iter.)	0.548
RANSAC (200 iter.)	1.014
RANSAC (300 iter.)	1.534

데, “HT+”는 식 (11)에 의해 각도 범위를 제한하여 HT를 수행하는 프로세스를 의미한다. 또한, “RANSAC( $N$  iter.)”는 RANSAC 반복 횟수  $N$ 을 나타낸다.

Table 3에 제시된 결과를 바탕으로, RANSAC 반복 횟수를 100으로 유지하기로 결정했다. 또한, HT는 각도 범위를 제한함으로써 처리 시간을 10 ms 이상 줄일 수 있었다.

## 5. 결론

본 연구에서는 우천시 및 청명시 주야에 촬영한 영상에서 자차로와 자차로 인접 차로들의 경계선을 검출하는 MLDA를 제안했다. 노면에는 차선표식뿐만 아니라 차선표식으로 오인될 수 있는 다양한 표시들이나 설치물이 있다. 더구나 우천시에는 비로 인해 야기되는 다양한 잡음이 무작위로 영상 전체에 나타난다. 이러한 요인들이 영상 처리를 통한 차선 인식을 어렵게 한다. 최근에 제안된 CNN 기반 알고리즘들조차도 이 문제에 대해 어려움을 겪고 있다.

본 연구에서 제안된 MLDA는 L2GA와 다단계 픽셀 필터링 기법을 병합하여 이러한 문제를 해결했다. 다단계 픽셀 필터링은 영상에 나타나는 여러 잡음하에서도 차선 표식의 속성을 견지하는 픽셀들을 강건하게 추출했다. 그러나 청명시 촬영한 영상과 달리 우천시 촬영한 영상에는 영상 전체에 무작위로 잡음이 나타나므로 다단계 필터링만으로는 한계가 있다. 이에 대한 해결책으로

L2GA를 제안했다. L2GA의 지역적 접근을 위해 최근 연속 영상에서 검출된 차선들로부터 참조선을 생성하고, 이 참조선을 따라 수직으로 절반씩 겹치는 작은 윈도우들을 배치하는 방법을 고안했다. 이는 에지 정보의 전역적 처리보다 지역적 활용을 우선시한 것이다. 즉 소영역 내에서 차선 속성에 부합하는 픽셀 탐색에 중점을 둔 것이다. 전역적 접근에서는 소영역들내에서 차선일 것으로 기대되어 추출된 대표점들에 RANSAC 및 최소 제곱법을 적용하여 정확한 차선을 검출했다.

MLDA는 최근의 CNN 기반 LDA들이 차선 검출에 실패한 영상에서도 차선을 정확하게 검출했고, F1점수는 86으로 최신의 방법들보다 약 10% 더 높은 만큼 우수한 성능을 보였다.

## 후 기

이 연구는 2023년도 산업통상자원부 및 산업기술평가관리원(KEIT) 연구비 지원에 의한 연구임("20025038").

## References

- 1) J. H. Kim, J. G. Song and J. W. Lee, "Extraction of Lane Information in Poor Environments by Combining Characteristics of Lane Markings, Sequence Information, and RANSAC," Proceeding of the 36th Workshop on Image Processing and Understanding, 2024.
- 2) J. W. Lee, "A Machine Vision System for Lane-Departure Detection," Computer Vision and Image Understanding, Vol.86, No.1, pp.52-78, 2002.
- 3) J. W. Park, J. W. Lee and K. Y. Jhang, "A Lane-Curve Detection Based on an LCF," Pattern Recognition Letters, Vol.24, No.14, pp.2301-2313, 2003.
- 4) 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.
- 5) J. M. Park and J. W. Lee, "Lane Estimation by Particle-Filtering Combined with Likelihood Computation of Line Boundaries and Motion Compensation," Image and Vision Computing, Vol.79, pp.11-24, 2018.
- 6) Traffic Signal Setting and Management Manual, Korean National Police Agency, 2012.
- 7) L. Tableini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza and T. Oliveira-Santos, "Keep Your Eyes on the Lane: Real-Time Attention-Guided Lane Detection," Proceeding of Computer Vision and Pattern Recognition(CVPR), pp.294-302, 2021.
- 8) Z. Chen, Y. Liu, M. Gong, B. Du, G. Qian and K. Smith-Miles, "Generating Dynamic Kernels via Transformers for Lane Detection," Proceeding of International Conference on Computer Vision (ICCV), pp.6835-6844, 2023.
- 9) H. Honda and Y. Uchida, "CLRerNet: Improving Confidence of Lane Detection with LaneIoU," Proceeding Winter Conference on Applications of Computer Vision(WACV), pp.1176-1185, 2024.
- 10) T. Zheng, Y. H. Huang, Y. Liu, W. Tang, Z. Yang, D. Cai and X. He, "CLRNet: Cross Layer Refinement Network for Lane Detection," Proceeding of Computer Vision and Pattern Recognition(CVPR), pp.898-907, 2022.
- 11) Y. Chen, P. K. Wong and Z. -X. Yang, "A New Adaptive Region of Interest Extraction Method for Two-Lane Detection," Int. J. Automotive Technology, Vol.22, No.6, pp.1631-1649, 2021.
- 12) B. S. Shin, J. Tao and R. Klette, "A Superparticle Filter for Lane Detection," Pattern Recognition, Vol.48, No.11, pp.3333-3345, 2015.
- 13) M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Communication of the ACM, Vol.24, No.12, pp.381-395, 1981.
- 14) N. Sukumar and P. Sumathi, "A Robust Vision-Based Lane Detection Using RANSAC Algorithm," Proceeding of IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT), 2022.
- 15) J. Guo, Z. Wei and D. Miao, "Lane Detection Method Based on Improved RANSAC Algorithm," Proceeding of International Symposium on Autonomous Decentralized Systems(ISADAS), pp.285-288, 2015.
- 16) <https://github.com/TU-Simple/tusimple-benchmark>.
- 17) <https://xingangpan.github.io/projects/CULane.html>.
- 18) X. Pan, J. Shi, X. Wang and X. Tang, "Spatial as Deep: Spatial CNN for Traffic Scene Understanding," Proceeding of the AAAI Conference on Artificial Intelligence, Vol.32, No.1, pp.7276-7283, 2018.
- 19) S. Sultana and B. Ahmed, "Lane Detection and Tracking under Rainy Weather Challenges," Proceeding of IEEE Region 10 Symposium (TENSymp), 2021.
- 20) P. C. Wu, C. Y. Chang and C. H. Lin, "Lane-Mark

- Extraction for Automobiles under Complex Conditions,” *Pattern Recognition*, Vol.47, No.8, pp.2756–2767, 2014.
- 21) C. Lee and J. H. Moon, “Robust Lane Detection and Tracking for Real-Time Applications,” *IEEE Transactions on Intelligent Transportation Systems*, Vol.19, No.12, pp.4043–4048, 2018.
  - 22) A. S. Ahmed, “Comparative Study Among Sobel, Prewitt and Canny Edge Detection Operators Used in Image Processing,” *Journal of Theoretical and Applied Information Technology*, Vol.96, No.19, pp.6517–6525, 2018.
  - 23) I. -C. Sang and W. R. Norris, “A Robust Lane Detection Algorithm Adaptable to Challenging Weather Conditions,” *IEEE Access*, Vol.12, pp.11185–11195, 2024.
  - 24) C. Akinlar and C. Topal, “Edlines: Real-Time Line Segment Detection by Edge Drawing(ed),” *Proceeding of International Conference on Image Processing*, pp.2837–2840, 2011.
  - 25) R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, USA, 1992.
  - 26) S. Sultana, B. Ahmed, M. Paul, M. R. Islam and S. Ahmad, “Vision-Based Robust Lane Detection and Tracking in Challenging Conditions,” *IEEE Access*, Vol.11, pp.67938–67955, 2023.
  - 27) J. Seo, S. Oh and Y. Kim, “A Study of Curved Lane Detection Based on Dual Sensor Monitoring of LiDAR and Camera,” *Transactions of KSAE*, Vol.29, No.2, pp.197–204, 2021.
  - 28) H. Zhao, O. Kim, J. S. Won and D. J. Kang, “Lane Detection and Tracking Based on Annealed Particle Filter,” *International Journal of Control, Automation, and System*, Vol.12, No.6, pp.1303–1312, 2014.
  - 29) J. Park, J. Lee, K. Jhang, J. Jung and K. Ko, “The Detection of the Lane Curve Using the Lane Model on the Image Coordinate Systems,” *Transactions of KSAE*, Vol.11, No.1, pp.193–200, 2003.