

# 개선된 PPHT를 이용한 선분 인식 알고리즘

## Line Segment Detection Algorithm Using Improved PPHT

이 찬 호<sup>\*\*</sup>, 문 지 현<sup>\*</sup>, 응웬 두이 풍<sup>\*</sup>

Chanho Lee<sup>\*\*</sup>, Ji-hyun Moon<sup>\*</sup>, Nguyen Duy Phuong<sup>\*</sup>

### Abstract

The detection rate of Progressive Probability Hough Transform(PPHT) is decreased when a lot of noise components exist due to an unclear or complex original image although it is quite a good algorithm that detects line segments accurately. In order to solve the problem, we propose an improved line detecting algorithm which is robust to noise components and recovers slightly damaged edges. The proposed algorithm is based on PPHT and traces a line segments by pixel and checks of it is straight. It increases the detection rate by reducing the effect of noise components and by recovering edge patterns within a limited pixel size. The proposed algorithm is applied to a lane detection method and the false positive detection rate is decreased by 30% and the line detection rate is increased by 15%.

### 요 약

영상 인식에서 널리 이용되는 PPHT(Progressive Probability Hough Transform)는 직선을 정확하게 인식하는 우수한 알고리즘이나 원본 영상이 선명하지 않거나 복잡하여 잡음 성분이 많은 경우 인식률이 감소하는 문제가 있다. 이러한 문제를 해결하기 위해 잡음에 강하고 손상된 가장자리 패턴을 복구하며 직선을 인식하는 개선된 PPHT 방식을 제안한다. 제안하는 알고리즘은 픽셀 단위로 직선을 추적하고 검증하여 선분을 검출하는 방식으로 잡음의 영향을 최소화하고 손상된 가장자리 패턴을 일정 범위 내에서 복구하여 인식률을 증가시켰다. 제안한 알고리즘을 차선 인식에 적용하여 직선의 오인식률을 30% 이상 감소시키고 선분 인식률이 15%까지 증가함을 확인하였다.

*Key words* :Hough Transform, Standard Hough Transform, Progressive Probability Hough Transform, Line Segments Detection, Lane Detection

\* Dept. of Electronic Engineering, Soongsil University

★ +82-2-825-8108, e-mail : [chlee@ssu.ac.kr](mailto:chlee@ssu.ac.kr)

※ This work was supported by the Industrial Core Technology Development Program (10049192, Development of a smart automotive ADAS SW-SoC for a self-driving car) funded By the Ministry of Trade, industry & Energy  
Manuscript received Mar 7, 2016; revised Mar 15, 2016; accepted Mar 15, 2016

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.

## 1. 서론

영상 데이터의 증가와 함께 영상 이미지에서 패턴 인식은 다양한 분야에서 사용되기 시작하였다. 증강 현실, 보안 카메라, 얼굴 인식, 스마트카 등 그 응용 분야는 널리 확산되고 있다. 특히 스마트카의 운전 보조 시스템은 주행 환경 인식이 필수적인데 최근에는 상대적으로 저비용인 카메라 영상을 이용하는 추세이다. 따라서 영상 인식을 통한 주변 환경 인식 기술이 필수적이고 관련 연구가 활발히 이루어지고 있다<sup>[1]</sup>.

인식 기술은 대상 물체의 가장자리와 색상을

인식하는 작업에서 시작한다. 운전 보조 시스템에서 차선이나 교통 표지판 인식을 위해서는 가장자리 검출 후 직선을 탐지하는 과정이 필요하다. 허프 변환 알고리즘은 디지털 이미지에서 직선을 검출하는 가장 대표적인 알고리즘이다. 캐니(Canny) 가장자리 탐지 과정을 거친 이진화 영상으로부터 직선을 찾아내는 허프(Hough) 변환 알고리즘은 데이터의 손실 및 왜곡이 포함된 영상에서도 직선을 인식하는 장점을 가져 직선 탐지에 널리 이용되고 차선 인식에도 활용되고 있다<sup>[2]</sup>. 그러나 허프 변환의 경우에는 직선 정보를 검출하기 위해서는 삼각함수, 곱셈, 나눗셈 등과 같은 오버헤드가 큰 연산들의 반복적 수행이 필요하기 때문에 실시간 성능을 보장하기 어렵다는 단점이 있다<sup>[3]</sup>. 또한 허프 변환은 특정한 점들을 지나는 직선을 찾으므로 영상의 실제 선분을 찾을 수는 없다. 허프 변환의 연산량을 줄이고 직선 성분의 양 끝점을 찾아 실제 선분을 찾는 알고리즘이 PPHT(Progressive Probability Hough Transform)이다. PPHT는 일반적으로 가장자리 탐지 이후 진행되는데 선명하게 나타난 선분은 잘 찾지만 가장자리 탐지 결과가 복잡하여 다수의 선이 잡음으로 존재하는 경우에는 선분에 대한 인식률이 낮아지는 단점을 갖는다. 카메라로부터 입력받은 영상 이미지의 경우 날씨, 조도, 그리고 이미지 센서의 성능 등에 따라 가장자리 탐지 결과는 사람 눈에는 잘 보이지 않는 잡음 성분을 포함하게 된다. PPHT는 이러한 성분을 고려하여 정확한 직선 성분만 찾게 되므로 육안으로는 직선으로 인식되는 선분을 찾지 못하거나 2개 이상의 선분으로 인식하는 경우가 발생한다.

본 논문에서는 PPHT 방법을 수정하여 직선 성분 탐지 결과가 육안으로 보는 것과 유사한 결과를 얻을 수 있도록 개선한 허프 변환 방식을 제안한다. 제안하는 방식은 육안으로 보이지 않는 픽셀 수준의 작은 오차를 허용하여 직선 성분 인식률을 증가시켰고 잡음에 강한 특성을 보인다. 이를 차선 인식에 적용하여 기존의 PPHT보다 차선 인식률이 증가함을 확인하였다.

## II. 제안하는 알고리즘

### 1. PPHT

영상 이미지에서 직선을 인식하기 위해서는 영상의 모든 가장자리 영역을 찾아주는 작업을 진행한 후 가장자리로부터 직선 성분인 선분을 분류하고 확인하는 작업이 필요하다. 가장자리를 찾는 작업은 캐니 가장자리 탐지 방식이 많이 이용된다<sup>[4]</sup>.

허프 변환은 선분을 찾는 영상 처리 알고리즘 중 하나로 식 (1)을 사용하여 영상의  $(x, y)$ 좌표 공간에 존재하는 각 픽셀을 허프 공간이라 일컫는  $(\rho, \theta)$  좌표 공간으로 바꾸어 직선 정보를 검출하는 알고리즘이다.

$$\rho = x \times \cos\theta + y \times \sin\theta \tag{1}$$

그림 1에서 같은 직선상에 존재하는 두 좌표 P1  $(x_1, y_1)$ 과 P2  $(x_2, y_2)$ 는 한 점의  $(\rho, \theta)$  좌표에서 교점을 갖는다<sup>[5]</sup>.

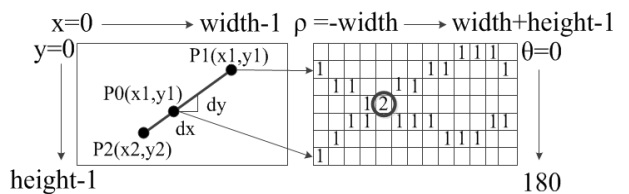


Fig. 1. Hough Transform of digital pixel data  
그림 1. 디지털 픽셀 데이터의 허프 변환

허프 변환은 이를 이용하여 한 점  $(\rho, \theta)$ 에 누적되는 빈도수를 한 직선에 분포된 픽셀 정보로 인식하여 누적된 값이 정해진 임계값보다 크면 직선으로 인식하게 된다. 즉,  $(x, y)$  좌표계에서 하나의 직선상의 픽셀은 허프 공간에서는 하나의  $(\rho, \theta)$  좌표에 대응하여 직선상의 픽셀에 대해 허프 변환을 실행하면 허프 공간에서는 하나의 점으로 나타난다. 이 점에 누적된 수가 클수록 직선의 길이가 길다는 것을 의미한다.

그러나 이 방식은 직선이 존재하고 길이가 짧거나 길다는 것은 알 수 있지만  $(x, y)$  좌표계에서 선분이 어디에 위치하는지는 알 수가 없다. 또한, 모든 픽셀에 대해 변환을 진행하므로 연산량이 크다는 단점이 있다. 이를 보완하기 위한 것이 PPHT이다. PPHT는 허프 변환과 같은 방식으로  $(x, y)$  평면의 좌표들을  $(\rho, \theta)$  평면의 좌표로 변환하는 것은 동일하지만 모든 가장자리의

픽셀에 대해서 진행하는 것이 아닌 임의로 선택된 일부 가장자리 픽셀에 대하여 변환을 진행하고 임계값보다 큰 값 중에서 최댓값을 찾아서 확률적으로 선분을 추적한다<sup>[6]</sup>. 즉, 이미지상의 임의의 점을 선택하여 허프 좌표계로의 변환 작업을 진행할 때 sine, cosine값을 이용하여 식 (1)의  $\rho$ 와  $\theta$ 에 해당하는 값을 허프 공간의 한 점에 해당하는 어큐뮬레이터(accumulator)에 저장(보팅, voting)한다. 어큐뮬레이터의 값이 임계값보다 크면 이를 직선이라고 판단하여 추적(tracking)을 시작하여 확인한다. 이때 전달인자로 주어지는 갭(gap)만큼 sine, cosine값을 이용하여 직선상의 다음 픽셀이 나타날 지점을 예측하여 확인한다. PPHT는 허프 변환에 비하여 연산량을 줄이고 직선을 하나의 선분으로 판단하여 선의 두 끝점을 확률적으로 찾아낸다. PPHT는 예측된 위치에 픽셀이 없거나 바로 옆 좌표에 위치하면 PPHT는 이를 직선으로 인식하지 못한다. 그러나 육안으로는 이러한 한 픽셀 정도의 오차를 구분할 수 없으므로 선분을 인식하지 못하는 결과가 된다. 또한, 가장자리 성분이 많은 경우 예측된 위치에 다른 가장자리의 픽셀이 존재하면 이를 직선으로 인식하여 직선이 아닌 것을 직선으로 인식하는 문제가 있다. PPHT 방식은 보팅 방식을 변경하여 누적시키는 값에 가중치를 주는 방식<sup>[7]</sup>, 보팅 방식을 개선하여 연산속도를 빠르게 하는 알고리즘<sup>[8]</sup>등 SHT와 PPHT방식의 직선 검출률을 높이거나 연산 속도 개선을 위한 많은 연구가 진행되었다. 하지만, 실제 이미지에서 보팅 후 최종적 선분 판단을 결정하는 알고리즘 개선은 이루어지지 않았다.

**2. 향상된 PPHT를 이용한 선분 인식 알고리즘**

가장자리 탐지 과정을 거쳐 이진화된 이미지에서 직선을 구성하는 픽셀들은 직선의 기울기에 따라 일정한 규칙을 갖는다. 이 특성을 이용하여 가장자리 영역으로부터 PPHT와 동일한 방법으로 임계값보다 큰 어큐뮬레이터를 찾고 이 선분의 기울기 변화를 확인하여 선분을 추적하고 검증과정을 통해 직선 여부를 판단할 수 있다. 제안하는 방식은 선분 추적과 검증, 그리고 추가 추적 과정으로 구성된다.

**가. 직선의 특성을 이용한 선분 추적**

선분 추적의 시작은 PPHT와 동일하다. 즉 가장자리 픽셀값들에 대해 허프 변환을 통해 어큐뮬레이터에 보팅 과정을 진행하고 임계값이 넘는 경우 추적을 시작한다. 이때 제안하는 알고리즘에서는 tangent 값을 계산할 때 모든 각도에 대해 직접 계산하지 않고 이전 각도와 차이만을 덧셈 연산으로 진행하여 연산량을 줄이고 하드웨어 구현에 적합하도록 하였다<sup>[9]</sup>.

디지털 이미지에서 직선은 연속된 픽셀의 집합이다. 즉 시작 픽셀을 기준으로 수평, 수직, 그리고 대각선 방향 중 하나의 이웃한 좌표에 최소한 한 개의 픽셀이 존재한다. 일단 직선의 방향이 정해지면 이웃한 픽셀은 3개의 셀 중에 하나에 존재한다. 직선의 기울기에 따라 그 위치가 달라지는데 직선과 수평선 사이의 각도에 따라 1° 단위로 라인을 그려보면  $\alpha$ 개 혹은  $\alpha-1$ 개의 픽셀이 연속되는 것을 알 수 있다. 0°~90° 사이의 직선을 생각하면 45°를 기준으로 45° 이하에서는 x 방향으로 픽셀이 연속되고 45° 이상에서는 y 방향으로 픽셀이 연속된다. 그림 2는 직선이 15°~18°의 각도를 가질 때 픽셀의 분포를 나타낸다. 픽셀이 x 방향으로 2개 또는 3개씩 연속되고 y 값이 1 증가하고 다시 픽셀이 나타난다. 연속된 픽셀이 몇 개인지는 각도에 따라 달라진다.

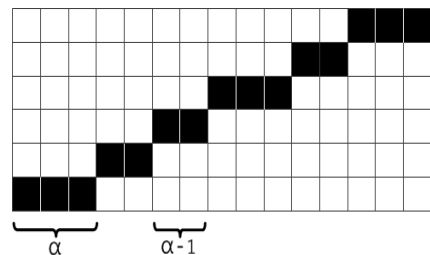


Fig. 2. Example of pixel distribution of according to the angle of a line (15°~18°)

그림 2. 직선에서의 각도에 따른 픽셀의 분포 예시 (15°~18°)

표 1에 기울기의 각도에 따른 직선의 특성값,  $\alpha$ 를 나타내었다. 이 표를 이용하면 연산량이 큰 tangent(=sine/cosine) 값을 미리 저장해놓거나 연산할 필요가 없다. PPHT에서는 추적 과정에서 예측 지점의 픽셀에 대한 허프 변환을 위해 필요하다. 그러나 제안하는 방법에서는 표 1을 이용하여 직선을 추적하므로 추적 과정에서 tangent

값은 필요 없다.

허프 변환에서 얻은 각도( $\theta$ )를 이용하여 선분

Table 1. Characteristic values of line according to the angles of the slope

표 1. 기울기 각도에 따른 직선의 특성 값

Angle[°]	1	2	3	4	5	6	7
$\alpha$	62	29	19	14	11	9	8
Angle[°]	8	9	10~11	12~14	15~18	19~26	27~45
$\alpha$	7	6	5	4	3	2	1

이 있을 것으로 추정되는 부분을 직선의 특성에 따라 추적한다. 이때 연속된 픽셀의 수는  $\alpha$ 개 혹은  $\alpha-1$ 개를 모두 허용한다. 완벽한 직선은  $\alpha$ 개 혹은  $\alpha-1$ 개의 픽셀이 일정한 규칙으로 배열된다.

이 규칙을 철저히 지키면 PPHT와 동일한 결과가 나온다.

그러나 가장자리 탐지의 결과가 선분을 완벽한 직선으로 표현하지 못하므로 실제로는 직선으로 보이는 부분도 픽셀 단위로 자세히 살펴보면 위의 규칙을 100% 만족시키기는 쉽지 않다. PPHT의 경우 갭을 통해 몇몇 픽셀을 건너뛰으로써 어느 정도 오차를 허용하지만, 그 폭이 크지 않다. 제안하는 알고리즘에서는 이러한 규칙에 관계없이  $\alpha$ 개 혹은  $\alpha-1$ 개의 픽셀을 모두 수용하여 오차의 허용 범위를 더 넓혔다. 이를 통해 잡음의 영향으로 픽셀이 빠지거나 추가된 경우에도 직선 인식이 가능하도록 하여 가장자리 탐지 결과가

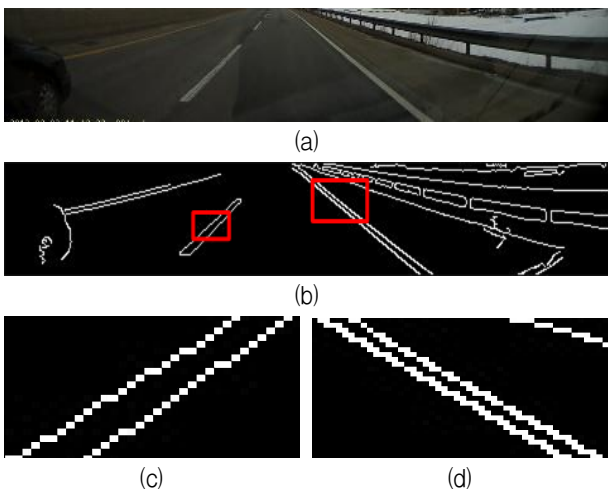


Fig. 3. Result of line detection. (a) original image. (b) binary image after edge detection. (c), (d) pixels of line segments.

그림 3. 직선 인식 결과 (a) 원본 이미지. (b) 에지 검출 후 이진화 이미지. (c), (d) 직선의 픽셀 이미지.

잡음의 영향을 많이 받아도 선분 인식을 더 잘 하도록 하였다. 한두 픽셀의 차이는 그림 3에 나타난 바와 같이 육안으로는 구분이 안 되지만 직선을 인식하는 과정에는 영향을 미쳐 인식률을 떨어뜨리는 원인이 된다.

나. 선분의 직선 검증

정확한 직선은 기울기(또는 각도)에 따라  $\alpha$ 개 혹은  $\alpha-1$ 개의 픽셀이 정해진 규칙에 따라 나타난다. 제안한 알고리즘에서는 선분 추적 과정에서 오차 허용을 위해 연속된 픽셀수가  $\alpha$ 개 혹은  $\alpha-1$ 개를 모두 허용하였다. 이 경우 완만하게 휘어진 곡선을 직선으로 인식할 수 있다. 이러한 문제를 해결하기 위해 선분 추적을 완료하면 선분의 픽셀을 일정한 간격(threshold2)으로 선택하여 그 변화량을 기준으로 직선인지 확인하는 작업을 거친다.

동일한 각도내의 변화량은  $\pm 1$ 이상 차이 나지 않는다. 이렇게 일정한 간격으로 직선임을 확인하고 마지막 검증 픽셀의 위치가 선분의 끝점이 아닐 경우에는 이 픽셀이 선분의 끝점 밖에 위치하는 것이므로 선분의 끝점에서 다시 역으로 확인하여 직선의 오차를 줄인다. 보팅된 라인의 경우 끝부분에서 한 칸씩 이동하며 검증한다. 그림 4에 직선 검증 과정이 나타나 있다. 허프 공간에서 일정한 수(threshold1)만큼 보팅된 점에 대해 직선의 특성인 최대 보팅된  $\theta$ 를 따라 P0 좌표에서 시작하여 직선을 추적 후(①) P0에서 일정한 간격(threshold2)으로 변화량을 확인하여 그 값을 비교하는 방식으로 직선 확인 작업을 진행한다(②). 끝점 P2가 마지막 검증 구간 내부에 위치하면 선분의 마지막 구간을 검증할 수 없으므로 끝

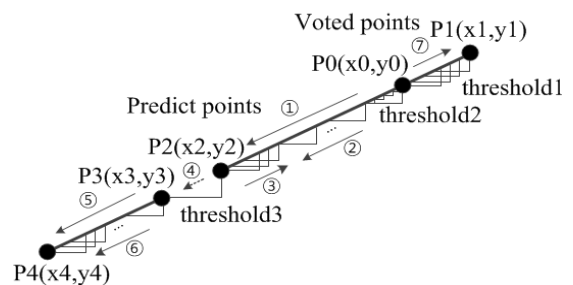


Fig. 4. Line Validation and extended tracking

그림 4. 직선 확인 및 끊어진 직선의 추가 추적

점에서 시작하여 역방향으로 threshold2 만큼 이동하여 변화량을 확인한다(③).

이 과정에서 변화량이 허용 범위를 벗어나면 직선이 아니므로 선분의 끝점을 직선을 유지하는 점으로 변경한다. 이를 위해 끝점 P2에서 역으로 한 픽셀씩 거슬러 올라가 다시 검증을 진행하여 직선의 조건을 확인한다. 동일한 기울기를 갖는 것으로 확인된 픽셀이 선분의 끝점이 된다.

다. 끊어진 직선 추가 추적

가장자리 탐지 과정에서 원본 영상의 선분이 잡음 등의 영향으로 탐지되지 않아 중간이 끊어진 상태로 이진화하는 경우가 발생할 수 있다. 이렇게 끊어진 선분을 이어주기 위해 선분의 직선 검증이 끝난 뒤 P2에서 일정한 간격(threshold3)을 건너서 P3로부터 동일한 기울기를 갖는 위치에 픽셀이 있는지를 확인하고(④) 픽셀을 발견하면 선분 추적 과정을 반복하여 추가 선분을 찾고(⑤) 이 선분의 길이가 threshold1보다 크고 기울기가 동일한 지를 직선 검증으로 확인하면(⑥) 이를 처음의 선분과 이어 손상된 선분을 복구하고 최종 끝점을 P4로 저장한다. 마지막으로 P0에서 P1까지의 구간에 대해 직선 검증을 진행하면(⑦) 선분 검출이 완료된다. 이러한 과정을 거쳐 제안한 알고리즘은 PPHT가 찾지 못하는 선분을 찾아 인식할 수 있다.

III 구현 결과 및 분석

제안하는 알고리즘에 따라 선분 인식 소프트웨어를 구현하여 건물 영상에 대한 선분 검출 결과와 차선 인식 알고리즘에 적용하여 차선 인식률을 PPHT와 비교하였다.

1. 선분 검출 결과

그림 5(a)는 이진화된 가장자리 탐지 결과 영상이며 본 논문에서는 캐니 가장자리 탐지 방법을 이용하였다. 이 가장자리 픽셀 데이터를 이용하여 PPHT 방식으로 선분을 검출한 결과가 그림 5(b)와 그림 5(c)에 나타나 있다. 그림 5(b)는 선분을 최대한 정확히 많이 찾아내도록 파라미터를 설정한 경우인데 잡음이 강한 영상의 하단 좌측 부분에서 잘못된 선분을 많이 찾게 된다. 표2

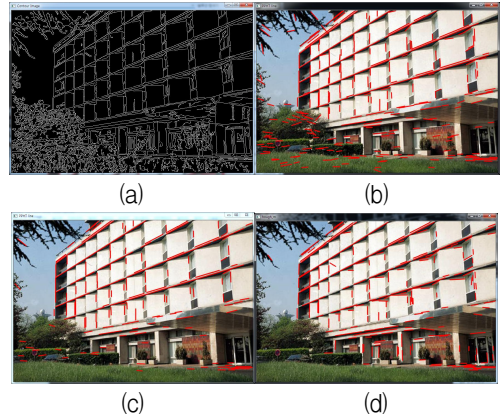


Fig. 5. (a) Canny Edge Image. (b), (c) results of PPHT with different parameters. (d) result of proposed Algorithm

그림 5. (a) 캐니 에지 이미지. (b), (c) 서로 다른 파라미터의 PPHT로 인식한 결과. (d) 제안하는 알고리즘의 인식 결과

Table2. Result of line detection in a building image  
표 2. 건물 이미지에 대한 선분 인식 결과

Fig. 5	(b)	(c)	(d)
Number of line	275	159	207
False positive	89	19	8

에서 보는 바와 같이 275개의 선분을 찾았으나 잘못된 선분도 89개를 검출하였다. 이러한 오류를 줄이기 위해 파라미터를 조정하면 오류는 많이 감소하나 여전히 오류가 남아 있고 실제의 선분을 인식하는 비율도 감소한다. 표 2에서 보는 바와 같이 잘못된 선분 수가 19개로 줄었으나 실제 선분도 159개만 검출한다. 반면에 제안하는 알고리즘은 207개의 선분을 검출하면서 잘못된 선분은 8개만 찾아 인식률이 우수함을 확인할 수 있다.

2. 차선 인식 알고리즘에 적용한 결과

차선 인식의 경우 건물 영상보다 가장자리 탐지 결과가 단순하여 PPHT가 선분 검출을 잘 할 수 있는 경우가 많다. 따라서 도로 표면이 깨끗한 경우에는 잡음이 거의 없어 PPHT가 100%에 가까운 인식률을 보인다. 그러나 도로 표면에 그림자나 도로 파손, 부분 포장 등으로 잡음이 발생하거나 비 오는 날에는 인식률이 떨어진다. 맑은 날 도로 표면의 잡음 성분이 많은 동영상과 우천 시 앞 유리에 빗방울로 인한 잡음이 많은 동영상에 대해 PPHT와 제안하는 알고리즘을 이용하여 인식률을 비교하였다. 두 개의 동영상은

모두 100프레임으로 구성되어 있고 모든 프레임에 대해 선분 인식률을 측정하여 평균값을 계산하였다. 이때 선분 인식은 ROI 영역에 대해서만 진행하여 도로 밖의 영상은 영향을 미치지 않도록 하였다.

그림 6은 차선 인식에 두 알고리즘을 적용한 결과이다. 그림 6(a), (b)와 같이 영상으로부터 캐니 가장자리 패턴 검출 후 차선 인식을 진행한 결과 그림 6(c), (d)에서 보는 바와 같이 PPHT를 이용하면 차선을 제대로 인식하지 못하는 것을 알 수 있다. 제안하는 알고리즘은 그림 6(e), (f)와 같이 차선을 제대로 인식한다.

전체 동영상에 적용하여 선분 인식률을 확인하여 표 3에 나타난 바와 같이 PPHT를 이용한 경우 70.5%와 83.4%, 제안하는 알고리즘을 이용한 경우 85.2%와 90.5%의 차선 인식률을 보였다. 이때 PPHT는 가장 최근에 개선한 방식을 적용하였다. 또한 PPHT의 단점인 선분 오검출률, 즉 False Positive(FP)는 약 30% 감소하였고, 직선

Table3. Comparison results of line and lane detection rate using PPHT and the proposed algorithm

표 3. PPHT와 제안한 알고리즘을 이용한 선분 및 차선 인식률 비교

	PPHT		Proposed algorithm	
	Clear	Rainy	Clear	Rainy
Line Detection	83.4%	70.5%	90.5%	85.2%
FP	43.8%	40.3%	13.2%	10.3%
FN	11.0%	11.8%	4.0%	6.1%
Lane Detection	65%	76%	72%	81%

을 인식하지 못한 경우인 False Negative(FN) 역시 각각 5%와 7% 감소하였다. 특히 우천 시 빗방울들이 영상에 많이 나타날 경우 PPHT를 이용한 경우 차선 인식률은 크게 감소할 뿐만 아니라 오검출률이 증가한다. 우천 시 영상의 PPHT를 이용한 차선 인식률은 76%인데 반해 제안하는 알고리즘을 이용한 경우 81%로 개선되었다. 여기에서 우천시 차선 인식률이 더 좋은 이유는 맑은 날의 영상이 도로에 잡음 성분이 많아 인식률이 매우 낮은 영상을 사용하였고 우천 시는 앞창의 빗물이 가장 큰 잡음 성분이고 도로 자체는 잡음 성분이 많지 않은 영상이어서 빗물에 의한 영향을 극복하면 오히려 우천 시가 더 차선 인식에 좋은 상황이기 때문이다. 즉, 빗물은 직선 성분을 많이 만들지 않아 빗물에 의한 영향을 극복한다면 직선 인식률은 도로 상태에 따라 결정된다.

#### IV 결론

본 논문에서는 가장자리 탐지된 패턴에 대해 PPHT의 단점을 극복하여 잡음에 강하고 일부 손상된 가장자리 패턴을 복구하여 성분을 검출하는 알고리즘을 제안한다. 제안하는 알고리즘은 선분 추적, 직선 검증, 추가 추적의 3단계로 이루어지며 어큐뮬레이터에 보팅 이후에는 삼각 함수 계산이 필요 없어 연산량을 줄일 수 있고 하드웨어 구현에 적합하다. 제안한 알고리즘을 차선 인식에 적용하여 실험한 결과 잡음 성분이 많은 도로나 빗방울이 많은 우천시에도 기존의 PPHT보다 직선의 오인식률을 30% 이상 감소시키고, 선분 인식률을 15%까지 증가시켰다. 제안하는 알고리즘은 가장자리 탐지와 동시에 직선을 인식하는 알고리즘에도 적용 가능하다.

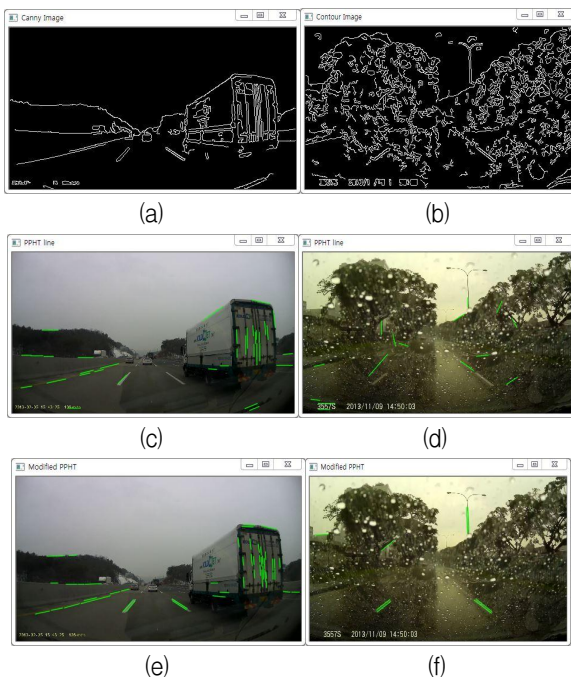


Fig. 6. Lane detection results using PPHT and the proposed algorithm. Canny edge image on (a) clear day and rainy day. (c), (d) Results using PPHT. (e), (f) Results using proposed algorithm

그림 6. PPHT와 제안한 알고리즘을 이용한 차선 인식 결과. (a) 맑은 날과 (b) 비오는 날의 캐니 가장자리 영상. (c), (d) PPHT를 이용한 인식 결과. (e), (f) 제안하는 알고리즘을 이용한 인식 결과

## References

- [1] Jae-Hyun Cho, Young-Min Jang, and Sang-Bok Cho, "A High-performance Lane Recognition Algorithm Using Word Descriptors and A Selective Hough Transform Algorithm with Four-channel ROI," *Journal of The Institute of Electronics and Information Engineers*, 52(2), pp. 148-161, 2015.2
- [2] Jeong-Rok Lee, Kyeong-ryeol Bae, and Byungin Moon, "A Hardware Architecture for Line Detection Based on Hough Transform with an Improved Voting Scheme," *Journal of the Korean Institute of Communications and Information Sciences*, 2013(1), pp. 88-89, 2013.9
- [3] K. Hanahara, T. Maruyama and T. Uchiyama, "A real-time processor for the hough transform," *Pattern Anal. Mach. Intell.*, 10(1), pp. 121-125, 1988.12
- [4] P. M. Daigavane, and P. R. Bajaj. "Road Lane Detection with Improved Canny Edges Using Ant Colony Optimization," *3rd International Conference on IEEE Emerging Trends in Engineering and Technology (ICETET)*, pp.76-80, Goa, India, 2010.11
- [5] Samir Tagzout, Karim Achour and Oualid Djekoune, "Hough Transform Algorithm for FPGA Implementation," *Signal Processing*, 81(6), pp. 1295-1301, 2001.10
- [6] J. Matas, C. Galambos, and J. Kittler, "Progressive Probabilistic Hough Transform," *British Machine Vision Conference*, 9(1), pp. 256-265, Southampton, England, 1998.9
- [7] Thuy Tuong Nguyen, Xuan Dai Pham, and Jae Wook Jeon, "An improvement of the Standard Hough Transform to detect line segments," *IEEE International Conference on Industrial Technology*, pp. 1-6, Chengdu, China, 2008.4
- [8] Dagao Duan, Meng Xie, Qian Mo, and Zhongming Han, "An improved Hough transform for line detection," *International Conference on Computer Application and System Modeling*, V2 pp. 354-357, Taiyuan, China, 2010.10
- [9] Ji-Hyun Moon and Chanho Lee, "Efficient Implementation of Computing Unit for Hough Transform," *2015 International SoC Design Conference*, pp. 279-280, Gyungju, Rep. of Korea, 2015.11

## BIOGRAPHY

### Chanho Lee (Member)

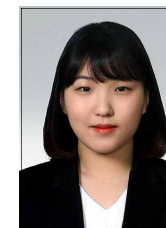


1987: BS in Electronic Engineering, Seoul National University.

1989: MS in Electronic Engineering, Seoul National University.

1994: Ph.D in Electrical Engineering, UCLA  
<Research Area> SoC on-chip-network, Memory controller, Image Pattern recognition

### Ji-hyun Moon (Student Member)



2015 : BS degree in Electronic Engineering, Soongsil University.

### Nguyen Duy Phuong (Student Member)



2013: Engineer degree in Electronics and Telecommunications, Hanoi University of Science and Technology