Lane Detection

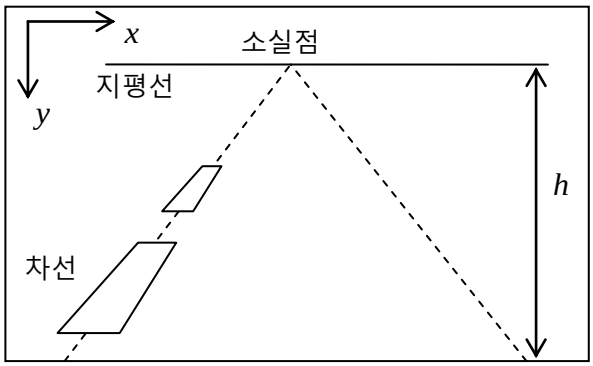
도로 면의 차선 정보를 직선으로 추출

KITECH 양광웅

로봇 자동차가 도로에서 위치인식과 자율주행을 할 때, 우선적으로 주변 도로환경으로부터 정보를 얻게 되는데, 여기서 가장 기본적이면서 중요한 정보가 차선 정보가 된다. 지도에 저장되어 있는 차선 정보와 카메라에서 인식한 차선 데이터를 비교함으로 로봇의 위치를 파악할 수 있다. 그리고 차선을 기반으로 도로면의 기호들과 문자를 읽을 수 있으며 차량의 상대적인 흐름을 읽어낼 수 있다.
여기서는 도로면의 차선을 직선으로 추출하여 다른 과정(위치 인식, 노면 기호 인식 등등...)에 필요한 기본적인 데이터를 제공하는데 목적이 있다.

차선의 특징

도로가 평평하다는 가정 하에서 차선의 소실점은 지평선 상에 존재한다.



- 차선의 색은 흰색, 노란색, 파란색등 원색 계열이다.
- 좁은 영역에서는 직선으로 볼 수 있다.
- 차선은 일정한 두께가 있다.
- 도로에는 두 개 이상의 차선이 있다.
- 양쪽 차선간의 폭이 일정하다.
- 차선이 직선이며 평행할 때 차선은 하나의 소실점에서 만난다.

차선 후보 추출

1. 카메라 이미지 가져오기

OpenCV의 cvGrapFrame() 함수와 cvRetrieveFrame() 함수로 카메라에서 이미지를 한 장 가져온다.

2. 카메라 이미지 가져오기

입력된 영상에서 cvSetImageROI(), cvCopy(),cvResetImageROI() 함수로 차선을 인식하고자 하는 영역을 따낸다. (하단 영상 예제에서는 붉은색 박스 영역을 의미함)

3. Gray 이미지로 변환

따낸 이미지를 cvCvtColor() 함수로 gray 이미지로 바꾼다. 이후 노이즈를 제거하기 위해 이미지에 가우시안 커널을 컨벌루션 하여 이미지를 부드럽게 만들 수도 있다. (cvSmooth 함수)

4. Edge 검출

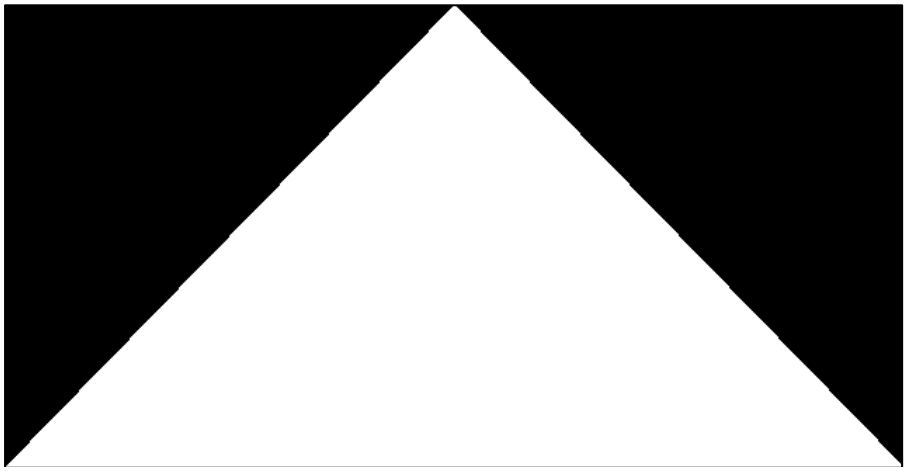
입력된 영상에 대하여 Sobel 연산(cvSobel() 함수)으로 x, y 축 방향으로 미분하여 edge를 구한 후,

cvCartToPolar() 함수로 edge의 magnitude와 direction을 구한다.

<magnitude 예시>

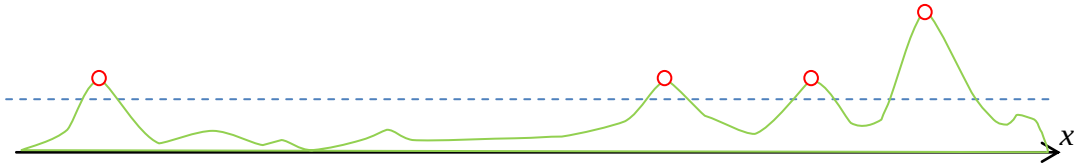
5. 마스크 이미지

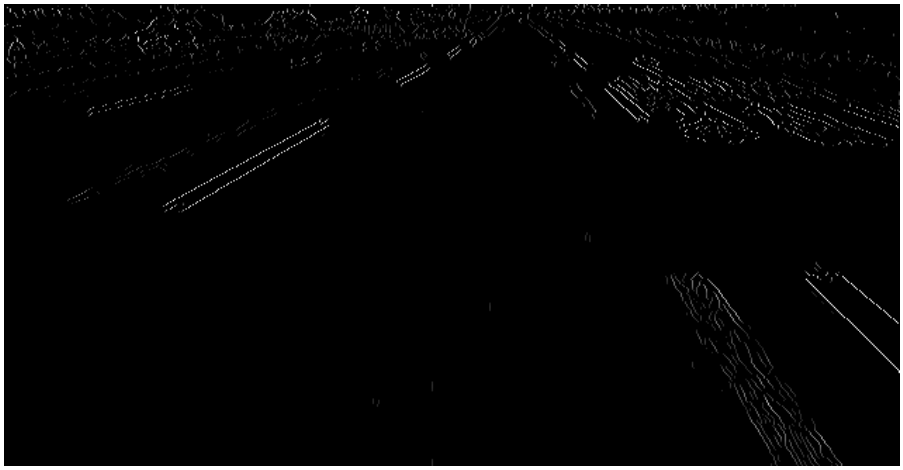
다음과 같은 마스크 이미지를 만든다. 검은색 영역에 속한 edge 중에서 direction이 수직에 가까운 것들은 차선이 아닐 확률이 높다. 먼저 이 영역의 edge 들 중에서 direction을 체크하여 수직에 가까운 것들은 차선 후보 점으로 고려하지 않는다.



6. Edge의 최대값 찾기

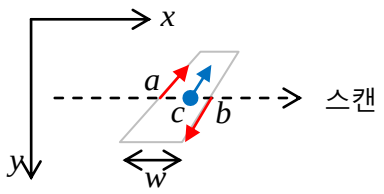
영상에서 y값을 고정하고 x값을 증가하면서 magnitude가 threshold를 넘으면서 최대가 되는 점을 찾는다.





7. 차선 후보점 평가

상기 과정에서 찾은 점들 중 같은 y축 값을 가지는 두 점 $a=(a_x,a_y),\ b=(b_x,a_y)$을 골라 두 점 간의 거리가 차선의 폭$(w)$을 만족하는지 검사한다.



여기서 두 점간의 거리는 다음과 같이 계산한다.

$$w_{ab}=b_x-a_x$$

그리고 차선의 폭은 y값에 따라 다음과 같이 결정된다.

$$w=\frac{w_{max}}{h}(y-y_{vp})$$

h - 영상의 바닥에서 소실점 까지의 높이
w_{max} - 영상의 바닥에서 차선의 폭
y_{vp} - 소실점의 y축 값

만일 두 점간의 거리가 차선의 폭을 만족한다면, 두 점에 대한 edge의 강도(magnitude)와 방향(direction)을 비교한다. 서로 edge의 강도가 유사하고 방향이 반대일 때 차선일 확률이 높다. 차선의 확률이 높을 때, 이 두 점에 대하여 중점(c)을 구하여 차선의 후보 점으로 저장한다.

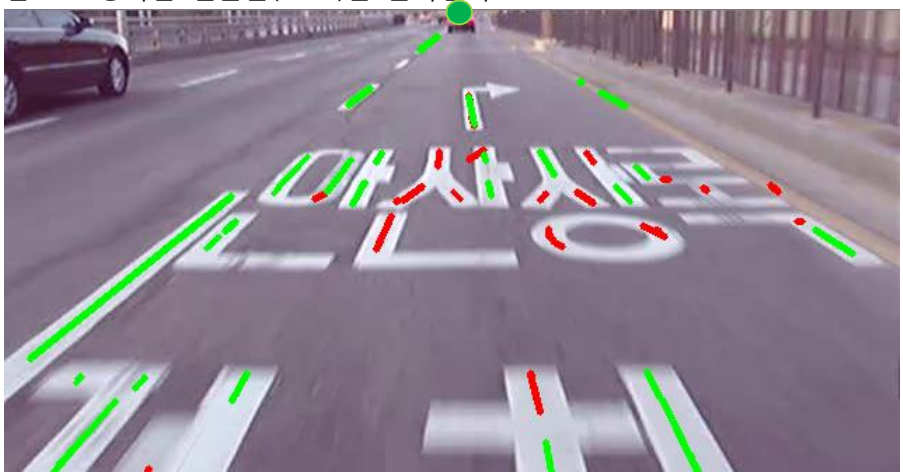
$$(c_x,c_y)=(\frac{a_x+b_x}{2},a_y)$$

8. Hough 변환

Hough 변환(cvHoughLines2 함수)으로 선분을 추출한다.

9. RANSAC

상기 과정에서 추출한 선분들로부터 RANSAC 알고리즘으로 차선의 소실점을 구한다. 그리고 소실점으로 향하는 선분들(inlier)만 골라낸다.



RANSAC 소실점(vp) 계산

RANSAC 알고리즘으로 차선의 소실점을 계산한다.

1. 변수들을 초기화 한다.
   $N = 500$ - 반복 횟수
   T_DIST - threshold
   $MAX\_inlier = -1$ - inlier의 최대 수
   $p = 0.99$ - 신뢰도
   $s = 3$ - 샘플 수
2. i-번째 추정 과정 $i = 1 : N$
   A. s 개의 차선 후보 선분들을 무작위로 선택하여 샘플을 만든다.
   B. 샘플로부터 소실점 $vp_{current}$ 를 계산한다.
   C. $vp_{current}$ 와 모든 차선 후보 선분간의 최단거리(수선의 길이) d_i 를 계산한다.
   D. 거리가 $(d_i < T\_DIST)$ 인 inlier의 숫자 m 을 센다.
   E. 만일 $(m > MAX\_inlier)$ 이이면 소실점을 업데이트 하고 모든 inlier 들을 저장한다.
   $vp \leftarrow vp_{current}$
   F. N 을 업데이트 한다. (n 은 차선의 후보 선분 수)
   $\epsilon = 1 - m / n$
   $N = \log(1-p) / \log(1-(1-\epsilon)^s)$
3. 마지막으로 저장된 inlier 들로부터 vp 를 새로 계산한다.

소실점 계산 모델

직선들이 주어졌을 때, 소실점을 계산하기 위한 모델은 다음과 같이 구한다.

차선을 나타내는 방정식이 다음과 같을 때,

$$a_i x + b_i y = c_i$$

차선 후보점들은 다음과 같이 표시될 수 있다.

$$(a_0, b_0, c_0),\ (a_1, b_1, c_1),\ \cdots,\ (a_n, b_n, c_n)$$

행렬로 바꾸어 표시하면 다음과 같다.

$$\underbrace{\begin{bmatrix} a_0 & b_0 \\ \vdots & \vdots \\ a_n & b_n \end{bmatrix}}_{\triangleq \mathbf{A}} \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\triangleq \mathbf{x}} = \underbrace{\begin{bmatrix} c_0 \\ \vdots \\ c_n \end{bmatrix}}_{\triangleq \mathbf{C}}$$

$\mathbf{Ax} = \mathbf{C}$

여기서 $\mathbf{x}$ 가 소실점이 된다. $\mathbf{A}$ 가 정방행렬이 아니기 때문에 Moore Penrose pseudo-inverse로 역행렬을 계산하면 다음과 같이 정리된다.

$\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{C},$

$\mathbf{A}^{\dagger} = (\mathbf{A}^{\mathbf{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathbf{T}}$ case row > col.

* http://robotics.caltech.edu/~jwb/courses/ME115/handouts/pseudo.pdf 참조

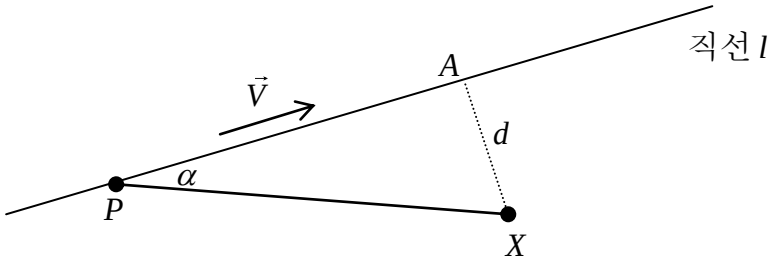
모델의 검증

직선들이 소실점 계산 모델에 잘 맞는지 검증하는 과정은 다음과 같다. 즉, 소실점으로부터 직선에 내린 수선의 길이가 짧으면, 이 직선은 모델에 잘 맞는 것이다.

$P = (p_x, p_y)$ 점을 지나면서 $\vec{V} = (v_x, v_y)$ 에 평행한 직선의 방정식은 다음과 같다. (t 는 임의의 변수)

$$l = P + t\vec{V}$$

임의의 점 $X = (x, y)$ 에서 위 직선으로 내린 수선의 길이 d는 다음과 같이 계산한다.



$$\left\|\overrightarrow{PX} \times \vec{V}\right\| = \left\|\overrightarrow{PX}\right\| \cdot \left\|\vec{V}\right\| \sin\alpha .$$

$$d = \left\|\overrightarrow{PX}\right\| \sin\alpha = \frac{\left\|\overrightarrow{PX} \times \vec{V}\right\|}{\left\|\vec{V}\right\|} .$$