

## 카메라 영상을 통한 실시간 차선·차간 인식에 관한 연구

김유신, 정대룡, 송성근, 송태홍  
주식회사 티디엘  
e-mail : loyfor@tdl.co.kr

## Lane and Vehicle Distance Detection Using Camera Image

Yu-sin Kim, Dae-ryong Jeong, Seong-geun Song, Tae-hong Song  
TDL

## 요 약

도로 주행 시 운전을 보조하고 안전 운전을 지원하기 위한 기술인 도로상황인지 시스템에 있어 효율적인 차선·차간 검출 기법은 위의 핵심적인 기술이다. 실시간으로 수집되는 도로 상황 영상 데이터 분석에 대한 처리 시간을 단축하기 위하여 각각의 영상 프레임에 대해 관심 영역을 설정한 후 허프 변환을 적용하였다. 본 논문은 카메라로 수집되는 도로 상황 영상에 관심 영역 설정을 통한 실시간 차선·차간 인식에 관한 연구로서, 차선과 차간 인식을 위한 효율적인 알고리즘을 제안한다.

## I. 서론

운전자는 운전 중에 인지, 판단, 조작을 반복하며 이러한 일련의 동작을 수행하는 가운데 운전자의 실수로 인하여 교통사고를 일으킨다. 도로 주행시 운전자의 과실로 인해 발생할 수 있는 교통사고를 미연에 방지하기 위하여 운전자에게 차선 이탈 및 차량 충돌 위험을 미리 알리는 것은 매우 중요하다.

이를 위해서 차량에 카메라를 탑재하고 도로 전방의 영상 정보를 수집·분석하여 운전자에게 시각 및 청각적인 경보를 제공하는 시스템이 많이 연구되고 있다. 그러나 차량에 카메라 및 임베디드 보드를 탑재하여 데이터를 수집하고 실시간으로 판단하는 프로세스는 많은 제약 사항을 수반한다. 카메라 영상을 처리하기 위한 기술은 실시간 프로세스에 적합하지 않을 수 있으며, 차선 및 차량 검출을 위해 영상 데이터에 대한 많은 탐색 및 처리가 필요하다.

이를 해결하기 위한 방안으로 전방 도로 영상을 획득한 후 그레이 스케일, 소벨 연산, 이진화, 허프변환 등의 전처리 과정을 거치며 보다 효율적으로 차선 인식 및 차량을 검출하는 기법을 제시한다.

## II. 관련 연구

## 1. 그레이 스케일(Gray Scale)

그레이 스케일 영상은 흔히 '흑백사진'처럼 만들어주는 것을 의미한다. 좀 더 영상처리 개념적으로 알아보자면 RGB 값이 모두 같은 값을 가지고 0(흑)에서 255(백)단계로 흑백의 강도를 나타내는 영상이다. 요약하면 색 정보는 없고 밝기 값만을 가지는 영상이라고 보면 되는데 여기서 색 정보라 함은 우리가 다루는 RGB구성보다는 YCbCr 구성을 말한다. RGB는 각각이 Red, Green, Blue의 색 정보를 가지고서 영상을 표현하는 반면에

YCbCr의 경우는 Y가 밝기를, Cb, Cr이 각각의 색 정보를 가진다.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.9563 & 0.6210 \\ 1 & -0.2721 & -0.6474 \\ 1 & -1.1070 & 1.7046 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

두 구성은 완전히 다른 형식이 아니라 위의 공식으로 상호변환이 가능하다. 영상 처리시에는 NTSC에서 제시한 YIQ방식에 기초하여 그레이스케일로 변환한다. 즉 위의 행렬식으로 계산해보면 밝기Y는  $0.299 * R + 0.587 * G + 0.114 * B$ 가 되는 것이다. 그리고 계산식을 보면 알겠지만 밝기 값은 최소 0에서 최대255값을 가지므로 이전의 밝기 조절에서와 같은 값이 255를 넘기는지에 대한 검사는 하지 않아도 된다는 걸 알 수 있다. 또한 Green값에 가장 많은 비율을 주었는데 이유는 인간의 눈이 녹색을 더 잘 감지하기 때문이다.

그레이스케일 변환 코드는 아래와 같다

```
for (y = 0; y < bmp.Height; y++) {
    for (x = 0; x < bmp.Width; x++) {
        blue = p[0];
        green = p[1];
        red = p[2];
        p[0] = p[1] = p[2] = (byte)(0.299 *
            red + 0.587 * green + 0.114 * blue);
    }
}
```

그림1은 그레이 스케일 변환한 것이다.



(a) 입력 영상 (b) 출력 영상

그림1. 그레이 스케일 영상

## 2. 소벨 연산(Sobel Operator)

카메라를 통해 얻어진 영상에서 차선을 검출하기 위해 여러 에지 추출 연산자 중에서 대각선 방향으로 강인성을 보이는 소벨 연산자를 이용하여 에지를 검출한다. 에지를 검출하는 방법으로는 편미분 연산자 계산에 의한 검출 방법으로 영상에서의 기울기(그라디언트)를 구하는 방법과 마스크를 이용하는 방법이 있으나, 여기서는 2차 미분에 해당하는 마스크 행렬을 이용하여 에지를 검출한다.

|    |   |   |    |    |    |
|----|---|---|----|----|----|
| -1 | 0 | 1 | 1  | 2  | 1  |
| -2 | 0 | 2 | 0  | 0  | 0  |
| -1 | 0 | 1 | -1 | -2 | -1 |

소벨 연산 방법은 아래와 같다.

```

for (j = 1 ; j < bmp.Height-1 ; j++ ) {
    for (i = 1 ; i < bmp.Width-1 ; i++ ) {
        h1 = - ptr2[j-1][i-1] - 2*ptr2[j-1][i] - ptr2[j-1][i+1]
            + ptr2[j+1][i-1] + 2*ptr2[j+1][i] + ptr2[j+1][i+1];
        h2 = - ptr2[j-1][i-1] - 2*ptr2[j][i-1] - ptr2[j+1][i-1]
            + ptr2[j-1][i+1] + 2*ptr2[j][i+1] + ptr2[j+1][i+1];
        hval = sqrt( (double)h1*h1 + h2*h2 );
        ptr1[j][i] = (BYTE)limit(hval);
    }
}
    
```

그림2는 소벨 연산자를 이용하여 영상의 에지를 추출한 것이다.



(a) 입력 영상 (b) 출력 영상

그림2. 소벨 연산 영상

## 3. 이진화(Binarization)

영상 정보는 잡음 신호를 많이 포함하고 있으므로 직접 차선 정보 추출을 위하여 활용하기 힘들다. 따라서 영상정보는 이진화 과정을 거쳐 잡음 성분들을 제거하여 차선 정보를 추출하기 적합한 영상 정보로 변환하여야 한다. 그러기 위해 소벨 연산자에 의해 추출된 에지에 대하여 이진화 작업을 해 준다. 이진화는 영상의 픽셀 값을 0 또는 255로 만드는 연산으로, 픽셀의 속성을 배경(background)과 객체(object) 두 개의 그룹으로 나누는 작업으로서, 영상 내에 원하는 객체의 정보(위치, 크기, 모양)을 얻기 위한 전처리 과정으로 많이 사용된다.

이진화의 종류로는 전역적 이진화(global binarization), 지역적 이진화(local binarization)가 있으며, 전역적 이진화는 영상 전체 픽셀에 대하여 하나의 임계값을 사용하여 이진화 시키는 방법이며, 영상을 여러 개의 구역으로 분할하여 각 구역마다 다른 임계값을 사용하여 이진화를 수행하는 지역적 이진화 방법이 있다. 다음은 이진화 알고리즘이다.

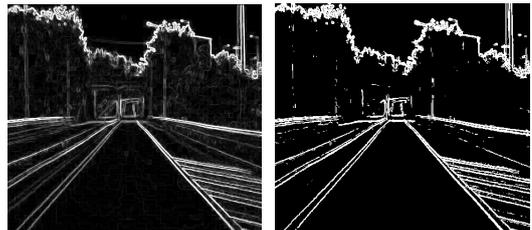
$$g(x,y) = \begin{cases} 0 & \text{if } f(x,y) \leq T \\ 255 & \text{if } f(x,y) > T \end{cases}$$

이진화 코드는 아래와 같다

```

for (y = 0; y < bmp.Height; y++) {
    for (x = 0; x < bmp.Width; x++) {
        if (ptr[j][i] > threshold)
            ptr[j][i] = 255;
        else
            ptr[j][i] = 0;
    }
}
    
```

아래 그림 3은 이진화 작업을 수행한 후의 영상이다.

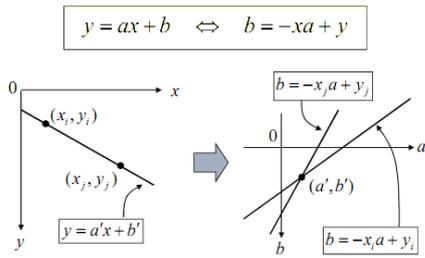


(a) 입력 영상 (b) 출력 영상

그림 3. 이진화 영상

## 4. 허프변환 직선검출(Hough Transform)

차선을 검출하기 위하여 영상 좌표에서의 직선의 방정식을 파라미터 공간으로 변환하여 직선을 찾는 알고리즘은 허프변환(Hough transform) 알고리즘을 이용하여 차선을 검출한다.



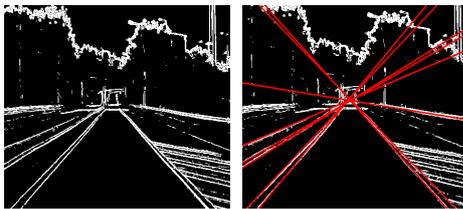
허프 변환 직선검출 코드는 아래와 같다

```

fo (j = 0; j < bmp.Height ; j++) {
for (i = 0; i < bmp.Width ; i++) {
if (ptr[j][i] > th) {
for (n = 0 ; n < num_ang ; n++) {
m = (int)floor( i * tsin[n] + j * tcos[n] + 0.5 );
m += (num_rho/2);
if ((m < 0) || (m >= num_rho)) continue;
arr[m][n]++;
}
}
}
}

```

아래 그림4는 허프변환을 통한 직선 검출 영상이다.



(a) 입력 영상 (b) 출력 영상  
그림 4. 허프변환 직선검출 영상

### III. 차선·차간 검출 알고리즘

본 논문에서는 차선 추출의 정확도를 높이고 처리 속도를 향상하기 위하여 각각 왼쪽과 오른쪽 차선으로 관심영역을 설정하여 허프 변환에 의한 직선 검출 알고리즘을 적용하였고, 차량 인식은 차량의 그림자를 성분이 임계치 이하인 경우 이를 객체로 설정하고 차량여부를 분석하였으며, y축 픽셀에 의한 거리 계산을 통해 차간 거리를 산출하였다.

#### 1. 차선 검출 알고리즘

전처리 과정에서 그레이 영상으로 변환된 영상을 소벨 연산과 이진화를 거쳐서 나온 차선 에지의 픽셀을 이용하여 다음과 같은 과정으로 검출이 이루어진다.

step 1. 영상에서 차선을 검출하기 위한 검사 영역을 지정한다.

step 2. 왼쪽 차선과 오른쪽 차선 검출을 위해 각각 관심영역을 지정한다.

step 3. 오른쪽 차선을 검출하기 위하여 오른쪽 차선의 관심영역을 직선성분 검출을 위하여 허프 변환한다.

step 4. 왼쪽 차선을 검출하기 위해 왼쪽 차선의 관심영역을 좌우 변환하고, 직선성분 검출을 위하여 허프 변환한다. 직선성분을 검출하면 그 결과를 다시 좌우 변환한다.

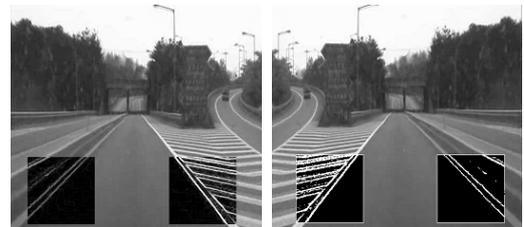
step 5. 검출된 왼쪽과 오른쪽 차선의 직선성분을 분석하여, 확률적 계산에 의하여 최대값 성분을 차선으로 취한다.

step 6. 차선에 대하여 필터링한다.

step 7. step 2 과정을 반복한다.

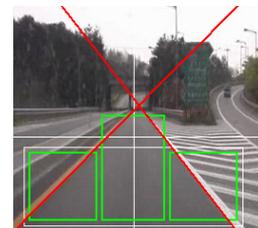


(a) 입력 영상



(b) step 2.

(c) step 4.



(d) 차선 검출 영상

그림 5. 차선검출 과정

그림 5는 위의 차선 검출 알고리즘을 이용하여 실제 도로 영상에서 차선을 검출하는 실험 영상이다.

#### 2. 차간 인식 알고리즘

전처리 과정을 거쳐서 나온 전방의 영상에서 차량의 색상 성분을 분석하고 필요에 따라 차량의 그림자나 뒷바퀴, 후미등의 색상 성분을 분석하여 차량 인식 검출 객체로 사용한다. 다음과 같은 과정으로 검출이 이루어진다.

step 1. 차량의 그림자 및 차량의 양쪽 뒷바퀴 모두 어두운색 성분을 가지는 특징을 이용하여 임계치(Threshold)를 설정한다.

step 2. 영상에서 전방 차량 인식을 위한 관심 영역을 지정한다.

- step 3. 전방 차량 인식을 위한 관심영역에서 임계치 이하인 경우, 이를 객체로 설정한다.
- step 4. 설정된 객체를 분석하여 차량여부 결정하고, 차량인 경우 차간 거리를 계산한다.
- step 5. step 2 과정을 반복한다.

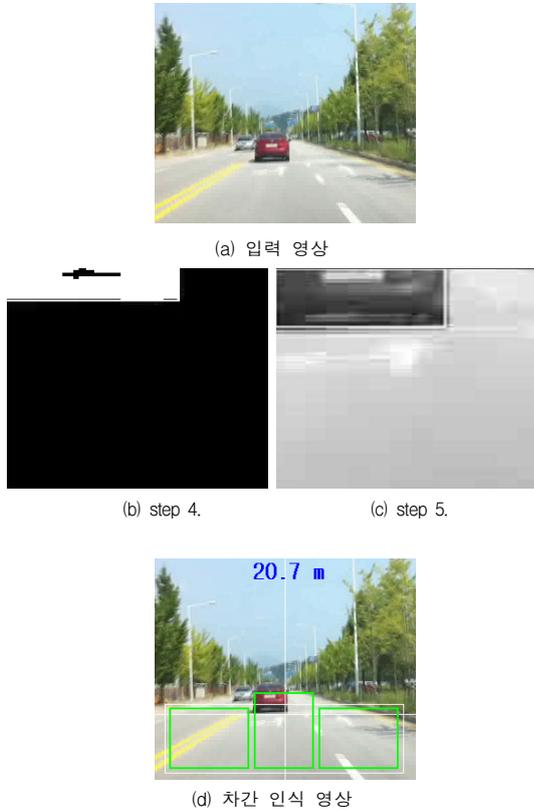


그림 6. 차간인식 과정

그림 6은 위의 차간 인식 알고리즘을 이용하여 실제 도로 영상에서 차간 거리를 인식하는 실험 영상이다.

### 3. 차간 거리 및 차량 속도 계산

차간 인식 알고리즘에 의해 차량을 인식하고, GPS(NMEA 0183) 수신기를 장착하여 차량속도와 초음파 센서에 의한 근접 차량 거리를 검출하고, 측정된 실제 차선의 거리 값과 차간 거리 값을 종합적으로 판단하여 차선 이탈 및 추돌 위험 경고 신호(Warning)를 사용자에게 알린다.



그림 7. 거리 및 속도 계산

그림 7은 위의 차선 검출 알고리즘과 차간 인식 알고리즘을 이용하여 실제 도로 영상에서 차선 이탈 및 추돌 위험 경고를 사용자에게 알려주는 실험 화면이다.

### IV. 결론 및 향후 연구

교통사고의 원인 중 하나인 운전자의 부주의를 줄이기 위해 차선 이탈 및 차량 근접시 운전자에게 경보를 제공하는 시스템의 핵심적인 기술로써 카메라 영상을 통해 수집되는 전방 도로 영상을 분석하기 위해 관심 영역을 설정하고 그레이스케일, 소벨 연산, 이진화, 허프 변환 등의 전처리 과정을 거쳐 색상 성분의 임계치를 얻고 윤곽을 찾아내는 방법으로 차선 및 차량을 검출하는 보다 효율적인 알고리즘을 연구하였다.

흐린 날씨, 야간 등 기상 악조건 상황에서도 차량 및 차선 검출에 대한 인식률을 높이기 위해서는 영상 처리 기법 및 검출 알고리즘에 대한 보완이 요구된다.

### 참고 문헌

- [1] 장운, "Hough Transform을 이용한 차선 인식과 응용" 2003
- [2] Richard O. Duda and Pete E. Hart Stanford Research Institute, Menlo Park, California Use of the Hough Transformation To Detect Lines and Curves in Pictures
- [3] Stephen Johnson (2006). Stephen Johnson on Digital Photography. O'Reilly.
- [4] Zehang Sun, Ronald Miller, George Bebis and David DiMeo, "A Real time Pre-crash Vehicle Detection System", 2002. IEEE
- [5] Rongben Wang, Youchun Xu, Libin and Yu fan Zhao, "A Vision-Based Road Edge Detection Algorithm", Intelligent Vehicle Symposium, 2002. IEEE
- [6] 여운관 외 2인, "이동물체 검출을 위한 현저한 윤곽선 추출에 관한 연구", 전자공학회 춘계학술대회 논문집, 17권, 1994.
- [7] Maire-Pierre Dubuisson and Anill k.Jain. "Object Contour Extraction using Color and Motion" Proc. of CVPR'93.
- [8] M. Bertozzi, A. Broggi, A. Fascioli, Stereo inverse perspective mapping: theory and application, Image and vision computing, vol. 16.
- [9] Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing, of the 1995.
- [10] A. Gern, U. Franke, and P. Levi, "Advanced Lane Recognition - Fusing Vision and Radar", Proceedings of the IEEE Intelligent Vehicles Symposium 2000, Dearborn, USA.
- [11] 이홍규, "디지털 영상처리 이론 및 구현 - Visual C++ 접근 방법", 사이텍미디어, 2007. 9.