

직선 Edge 추출에 의한 주행방향 및 장애물 검출에 관한 연구

A Study on the Proceeding Direction and
Obstacle Detection by Line Edge Extraction

정준익, 최성구*, 노도환*

* 전북대학교 전기전자제어공학부(Tel:70-2398; Fax:70-2394; E-mail: rdh@moak.chonbuk.ac.kr)

Abstracts In this paper, we describe an algorithm which estimate road following direction using the vanishing point property and obstacle detection. This method of detecting the lane markers in a set of continuous lane highway images using linear approximation is presented. This algorithm is designed for accurate and robust extraction of this data as well as high processing speed. Also, this algorithm reckon distance and chase about an obstacle. It include four algorithms which are lane prediction, lane extraction, road following parameter estimation and obstacle detection algorithm.

High accuracy was proven by quantitative evaluation using simulated images. Both robustness and the practicality of real time video rate processing were then confirmed through experiment using VTR real road images.

Keywords road following, navigation and autonomous vehicle, edge detection, obstacle detection

1. 서론

최근, 급증하고 있는 자동차 사고의 예방 및 방지를 위해 자동차의 자율 주행에 관한 연구가 활발히 전개되고 있다. 이러한 자동차의 자율 주행에서의 선결조건은 주행 가능한 지역에 대한 정확한 인식능력이 있다. 정확한 인식을 위한 시스템으로써 시각 시스템을 이용한 방법이 크게 주목되고 있다. 또한, 성공적인 자율 주행을 위해서는 각각의 인식대상에 대한 인식정보가 추정되어야만 한다.

일반적으로 자동차의 주행환경은 비포장된 도로와 포장된 도로로 구분한다. 전자의 경우에는 노면의 불규칙성때문에 칼라영상으로부터 색성분을 도로와 비도로로 분류하는 방법¹⁾을 이용하고, 후자의 경우에는 농담영상을 이용하여 도로의 차선이나 경계선에 의하여 발생하는 edge를 이용하여 주행지역을 인식²⁾한다. 본 연구에서는 도로주행차량을 대상으로 하고 이에 알맞은 후자의 방법을 이용한다.

자동차의 자율주행시 장애물에 대한 인식은 안전을 위한 중요한 사항으로 이에 따른 연구도 활발히 진행되고 있다. 주행중 장애물의 출현시 이에 대한 인식과 장애물과의 거리, 상대 위치 그리고 속도의 인식정보는 필히 선행되어야만 한다. 이러한 인식정보의 추정을 위하여 Stereo Vision을 이용한 방법과 Optical Flow를 이용한 방법³⁾이 연구되고 있다. 이러한 연구들은 자동차의 고속주행시 유용성이 결여되고 복수의 시스템을 필요로 한다. 본 연구에서는 단일 Vision 시스템을 이용하여 고속주행차량에 유용하도록 장애물에 대한 인식정보를 카메라 교정을 통하여 데이터베이스화하고 주행중 장애물 출현시 이를 적용하여 정보를 추정한다.

본 연구는 자율 주행을 위한 기초 연구로서 주행중의 주행 방향을 차선의 직선추정에 의한 무한원점 추정으로 결정할 수 있는 알고리즘과 카메라 교정을 통한 거리정보를 이용하여 장애물과의 거리를 추정하는 알고리즘을 제안한다. 그리고, 제안된 알

고리즘은 주행 시에 연속적으로 입력되는 영상으로부터 무한원점과 장애물을 추정할 수 있다. 이를 입증하기 위해 실제 도로영상에 적용한 결과를 제시한다.

2. 시스템 좌표계

자동차의 주행방향을 인식하기 위해서는 주행도로와 자동차에 설치한 카메라와의 관계를 규정할 필요가 있다. 따라서 실 주행 도로에 대한 수학적인 modeling이 필요하다. 본 연구에서는 주행 도로 환경이 포장된 직선도로 혹은, 완만한 커브를 가진 포장 도로를 대상으로 한다.

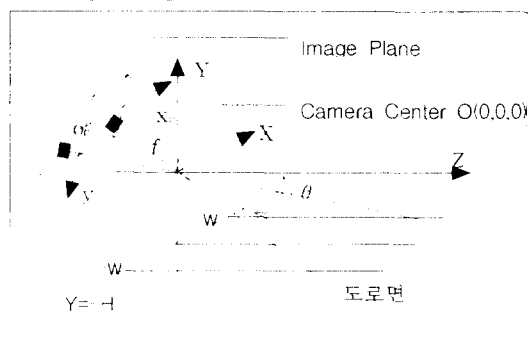


그림 1. 시스템 좌표계 I
Fig. 1. System coordinates I

그림 1은 도로면과 자동차에 설치된 카메라와의 좌표관계를 나타내고 있다. 카메라의 좌표계는 원점을 렌즈 중심으로 하는 직교 좌표계이다. 자동차의 정면 방향으로 카메라의 광축 Z를 설정한다. 영상면은 렌즈 중심으로부터 f 인 지점에서 X, Y평면을 이루는 오른손 좌표계를 이룬다. XYZ 는 공간좌표계로서 카메라좌표계의 원점과 동일하게 설정하였다. 이 원점으로부터 거

리 f 인 지점의 영상면은 공간좌표계의 X축과 평행하게 x_c 축을 설정하고 도로면과 θ 의 사이각을 갖는 y_c 축을 설정한 평면을 나타낸다. 도로는 폭이 $2W$ 이고 차선의 중심축이 공간좌표계의 Z축과 평행한 면으로 설정한다. 카메라의 영상좌표계와 공간좌표계의 관계식은 식(1)과 같다.

$$\begin{aligned} x_c &= \frac{fX}{Z \cos \theta - Y \sin \theta} \\ y_c &= \frac{f(Y \cos \theta + Z \sin \theta)}{Z \cos \theta - Y \sin \theta} \end{aligned} \quad (1)$$

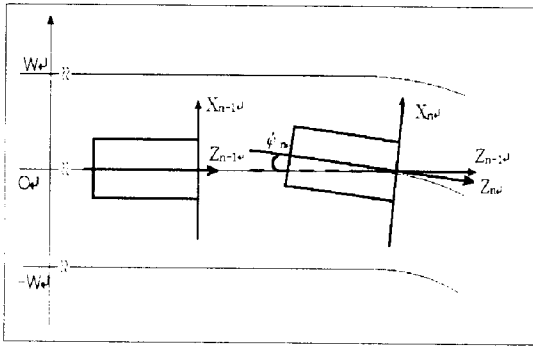


그림 2. 시스템 좌표계 II
Fig. 2. System coordinates II

그림 2.는 자동차의 자율주행에 필요한 주행방향파라미터를 나타낸 그림이다. 처리시간간격이 T일 때 자동차의 진행상태를 나타낸 그림이다. 시간 $n-1$ 일 때 자동차의 좌표계는 X_{n-1} 이다. 이때의 진행방향은 Z_{n-1} 이다. 이 자동차의 처리시간T후의 시간 n 일 때의 진행은 X_n 의 좌표계로서 진행방향은 Z_n 이다. 이때에 자동차의 진행에 따라서 변화하는 파라미터는 벡터 Z_{n-1} 과 벡터 Z_n 과의 사이각 φ_n 이다. 이 사이각 φ 를 자동차의 자세각이라 하고 이 값을 추정하는 것이 자동차의 주행방향을 결정하는 것이다. 본 연구에서는 이 자세각을 추정하기 위하여 무한원점의 원리를 이용한다. 공간상의 평행한 두 직선을 화상면상에 투영시키면 화상면상에 투영된 직선의 연장선은 어느 한 점에서 교차하게 되는데 이러한 교점을 무한원점이라고 한다. 영상면상에서 무한원점의 방향벡터를 (v_x, v_y) 라 하면 차선의 방향 벡터를 (v_x, v_y, f) 로 구할 수 있다. 이 성질을 이용하여 자동차의 진행방향을 결정하는 자세각 φ 는 식(2)와 같이 유도된다. 영상면상에서 처리시간 T의 간격을 두고 발생하는 무한원점을 이용하여 (2)식에서 자세각 φ 를 역으로 산출할 수 있다.

$$\varphi_n = \cos^{-1} \frac{v_{xn}v_{xn-1} + v_{yn}v_{yn-1}}{\sqrt{v_{xn}^2 + v_{yn}^2} \sqrt{v_{xn-1}^2 + v_{yn-1}^2}} \quad (2)$$

본 연구에서는 도로 영상을 이용하여 무한원점을 추정하고 추정된 무한원점을 이용하여 주행방향을 결정할 수 있는 파라미터 추정 알고리즘을 제안한다.

3. 주행방향 추정알고리즘

자동차의 자율주행은 도로면상의 백선이나 도로영역을 판별할 수 있는 정보를 제공하는 많은 요인을 인식하여 이를 추종함으로써 이루어진다. 본 연구에서는 영상처리 기법과 통계적 방법을 이용하여 화면상에서 결정되는 좌·우차선의 방정식을 유도하고 유도된 방정식에 의한 무한원점을 추종 factor로써 이용한다. 추정된 무한원점과 카메라의 광축이 이루는 자세각은 자동차의 진행방향을 결정한다.

카메라에 의해 입력된 주행 도로의 영상을 35×35 [pixel] 크

기의 sub-window를 좌우로 각각 4개씩 구성하여 각각의 sub-window에서 직선의 식을 추정하고 이로부터 무한원점을 추정한다. 이러한 처리 과정은 그림 3.과 같이 표현되며 각 알고리즘별 내용은 다음과 같다.

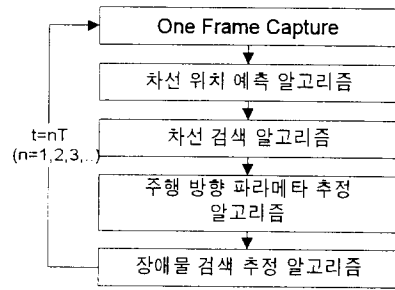


그림 3. 주행방향 추정알고리즘
Fig. 3. Lane Direction Estimation Algorithm

3.1 차선위치예측 알고리즘

차선 위치 예측 알고리즘은 그림 4.와 같이 예측하여 탐색영역을 축소하고 차선 검색 처리시간을 단축한다. 시간 $(n-1)$ 번째 영상의 차선방정식은 식(3)과 같다.

$$\begin{aligned} x &= a_l \cdot y + b_l & (\text{좌측차선방정식}) \\ x &= a_r \cdot y + b_r & (\text{우측차선방정식}) \end{aligned} \quad (3)$$

식(3)에서의 좌우경계영역은 식(4)로 나타낸다.

$$x = (a_l + a_r) \cdot y/2 + (b_l + b_r)/2 \quad (4)$$

시간 n 에서의 좌측·우측차선의 k 번 Sub-Window의 후보 x 좌표값($k=1,2,3,4$)은 식(5)(6)과 같다.

$$x_{lk} = a_l \cdot y + b_l + w \quad (5)$$

$$x_{rk} = a_r \cdot y + b_r + w \quad (6)$$

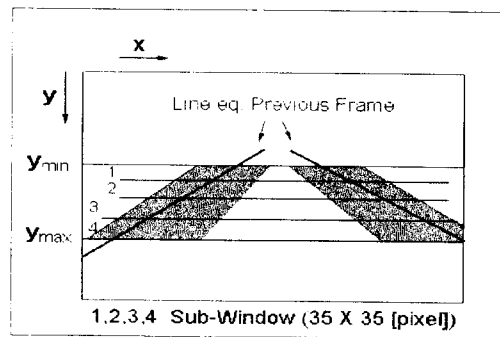


그림 4. 후보점 탐색영역
Fig. 4. Candidate search area position

3.2 차선검색알고리즘

그림 5.와 같이 영상면에서의 차선을 검색한다. 각 단계는 다음과 같다.

단계 1 : 전 처리

단계 1에서는 영상면에 평균 intensity보다 너무 어둡거나 밝은 부분에 대해 보상하기 위한 평활화처리를 한다.

평활화 처리된 영상은 보편적으로 이용하는 수직의 2차 미분 필터인 sobel 필터를 이용하여 에지추출을 시행한다. sobel 필터

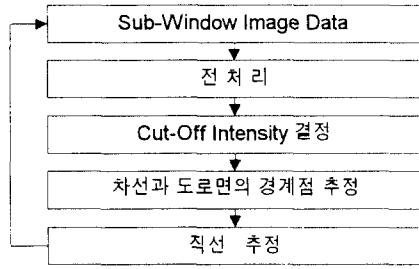


그림 5. 차선 검색 알고리즘

Fig. 5. Lane Detection Algorithm
 는 임의의 화소점 i, j 에서 intensity가 $I_{i,j}$ 일 때 i, j 점에서 gradient의 크기와 방향을 계산함으로써 도로 영상에서의 차선 에지를 추출한다. sobel 필터의 미분연산자는

$$S_x(i, j) = \begin{matrix} f_{i+1,j+1} + 2f_{i+1,j} + f_{i+1,j-1} \\ - f_{i-1,j+1} - 2f_{i-1,j} - f_{i-1,j-1} \end{matrix} \quad (7)$$

이다.

단계 2 : 도로 영상에 대한 cutoff intensity 결정

단계 2에서는 필터 처리된 sub-window 영역의 화소의 값들을 이용하여 2차 농담영상으로 변환하는 작업을 행한다. 영상면에서 차선에 대한 최소 intensity 한계치를 (8)식과 같이 설정한다.

$$U = \mu - T\sigma \quad (8)$$

(μ, σ : 각점의 intensity 평균치, 표준편차
 T : 사용자 parameter)

단계 3 : 차선과 도로면의 경계점 추정

단계 3에서는 sub-window의 수평 수직 배열을 따라 차선과 도로면의 경계점을 추정하는 것이다. 본 알고리즘은 sub-window의 수평과 수직 배열에 대한 2진화된 값을 평가하여 단계 3에서 발생한 노이즈를 감소시키고 차선의 경계를 나타내는 화소를 추정한다. 본 연구에서는 차선경계에 8개의 35×35 [pixel]의 크기로 선정한 각 sub-window에서 차선의 경계에 대한 에지점의 영상좌표를 추출한다. 추정은 (i, j) 의 intensity가 $I_{i,j}$ 이라할 때, 이 점을 중심으로 식(9)과 같이 연결성분이 존재하는 경우의 점만을 추출하고 식(10)의 조건을 적용하여 경계점을 추출한다. 이러한 화소를 sub-window 영상면 전체의 화소에 대해 수평과 수직선을 따라 반복하여 경계점을 추정한다.

$$G(i, j) = \begin{matrix} I_{i-1,j-1} + I_{i,j-1} + I_{i+1,j-1} \\ + I_{i-1,j} + I_{i,j} + I_{i+1,j} \\ + I_{i-1,j+1} + I_{i,j+1} + I_{i+1,j+1} \end{matrix} \quad (9)$$

$$\text{경계점 : } \begin{cases} \text{True : } G(i, j) \geq 3U \\ \text{False : } G(i, j) < 3U \end{cases} \quad (10)$$

단계 4 : 최소이승법에 의한 직선 추정

단계 4에서는 단계 3에서 추정된 차선의 에지 화소점들을 최소이승법을 이용하여 차선에 대한 직선식을 추정한다. 일반적인 직선의 식은 식(11) 로써, 기울기 m 과 y 축의 교점 b 는 에지점들에 대해 최소이승법(least squares fitting method)을 적용하여,

$$y = mx + b \quad (11)$$

$$m = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}$$

$$b = \frac{\sum y_i - m \sum x_i}{n} \quad (12)$$

(단, n 는 1부터 n 까지 변화량)

식(12)와 같이 구할 수 있다.

3.3 주행 방향 파라메타 추정 알고리즘

단계 1 : 무한원점 추정

차선검색알고리즘에서 구해진 영상면상의 직선들으로써 무한원점을 추정한다. 무한원점 v_x, v_y 는 직선들의 형식을 $ax-by=c$ 이라 할 때, 식(13)이 최소값이 되는 점으로 구한다.

$$\min \{f(v_x, v_y)\} = \sum \frac{a_i v_x + b_i v_y + c_i^2}{v_x^2 + v_y^2 + 1} \quad (13)$$

단계 2 : 주행 방향의 추정

무한원점의 변화를 이용하여 연속적으로 입력되는 영상에 대해 무한원점의 추정이 가능하면 식(2)에서 자동차의 주행 방향 파라메타 자세각 ϕ 를 추정할 수 있다.

3.4 장애물검색추정알고리즘

단계 1 : 각 구간별 소검색영역 설정

3개의 거리별 구간을 두고 각 구간에 장애물 감지를 위하여 소검색영역 10×10 [pixel]을 2×3 으로 설정한다. 차선 검색 알고리즘에서 구해진 좌우차선의 방정식에서 식(14)에서 식(16)과 같이 좌측, 우측, 중앙의 소검색영역의 위치를 산출한다.

$$x_{lb} = a_l \cdot y + b_l + w_{lb} \quad (14)$$

$$x_{ro} = a_r \cdot y + b_r + w_{ro} \quad (15)$$

$$x_{bro} = (a_l + a_r) \cdot y/2 + (b_l + b_r)/2 + w_{bro} \quad (16)$$

단계 2 : 소검색영역의 정보값 산출

각각의 소검색영역에서의 Mean, Median값을 구한다.

단계 3 : 정보값에 의한 장애물 판단

단계 2에서 산출한 Mean값과 Median값의 차이를 이용하여 시간 n 일 때의 장애물 판단은 식(17), 식(18)과 식(19)에 의해 이루어진다.

$$Ob_n = \text{median}_n - \text{median}_{n-1} \quad (17)$$

$$\alpha = \frac{1}{t_v} \sum_{i=1}^{t_v} \{ \text{mean}_{n-i} \} \quad (18)$$

$$Ob_i : \begin{cases} \text{Obstacle : } \{ Ob_n > a \} \text{ or} \\ \{ Ob_n < a \text{ and } Ob_i = \text{Obstacle} \} \\ \text{Non-obstacle : } \{ Ob_n < a \} \end{cases} \quad (19)$$

단계 4 : 검색 블록별 연결성 조사

소검색영역에서 단계3에서 이루어진 판단정보를 이용하여 장애물로 판단된 영역의 연결성을 조사한다.

단계 5 : 검지 영역에 대한 거리 산출

장애물로 판단된 영역에서 최하단위치영역에 대한 화소값에 해당하는 거리값을 산출한다.

4. 실험 및 고찰

본 연구에서 4차선의 포장된 도로를 상정하여 512×480 [pixel]의 칼라영상으로서 연속적으로 입력되어지는 정보를 알고리즘에 적용하여 백색 혹은 황색의 차선을 추정하고 무한원

점을 구한다. 연구에 사용한 카메라는 초점거리가 8[mm]이고 지상높이 1.3[m]에서 자동차의 정중앙에 위치시키고 전방을 촬영하였다. 실험대상은 국도를 시속 40, 60, 80, 100 [Km/h]로 각각 주행하여 캡코더에 의한 동영상을 IBM-AT Intel 486-DX 53.5MHz에 설치한 1/30[s]의 Frame Grabber능력을 갖는 Data Translation DT - 2871 Image Process Board에 입력하여 알고리즘에 적용하였다.

무한원점 추정의 정확성을 알아보기 위해 무한원점의 이론적 추정치와 그림 9와 같은 직선 도로상에서 시속 60[Km/h]로 주행 중에 알고리즘 적용 실험치의 차이를 비교하여 그림 3과 그림 4에 제시하였다. 오차의 평가는 20 frame을 10회 반복 추정 한 오차 백분율이다. 그림 2와 그림 3에서 나타나는 것과 같이 오차는 $10^{-3} \sim 10^{-2}$ 의 범위에서 존재 함을 알 수 있다.

그림 9는 실제 도로 영상을 제한한 알고리즘에 적용하여 차

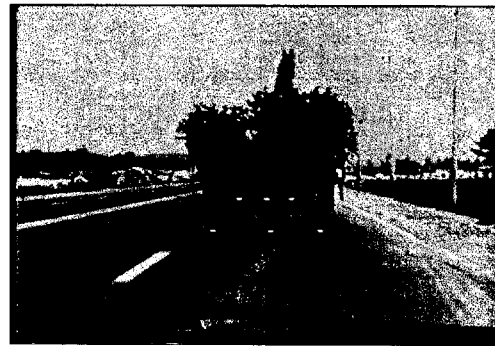


그림 9. 장애물 검색과 추정
Fig. 9. Obstacle detection and estimation

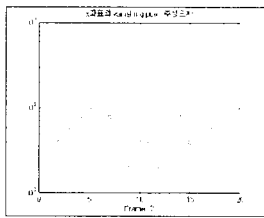


그림 6. X축 오차 추정
Fig. 6. X-axis error estimation

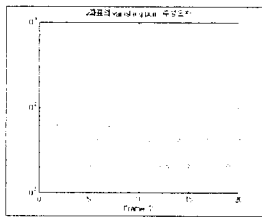


그림 7. Y축 오차 추정
Fig. 7. Y-axis error estimation

선의 직선을 추정한 결과이다. 추정된 직선은 우측실선의 경우 뿐만 아니라 좌측 점선 차선의 경우에도 정확한 직선을 추출함을 보여준다.

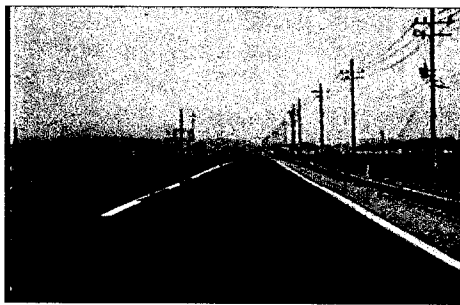


그림 8. 직선도로일 경우의 직선 추정
Fig. 8. Line estimation of straight road

그림 10은 주행시 발생하는 장애물을 검출한 결과를 나타낸 그림이다.

주행 방향을 추정하는데 소요되는 시간을 표 1.에 제시했다. 제시된 결과는 100[Km/h]의 속도로 주행시 Frame과 다음 Frame의 처리시간 동안 3.42[m]를 진행한다는 것을 알 수 있다. 즉, 제한한 알고리즘을 적용하면 실시간 처리가 가능함을 알 수 있다.

표 1. 1000 Frame동안의 각각에 대한 평균처리시간
Tabel. 1. Each average processing time during 1000 Frame

| | |
|---------------|-------------|
| 한 프레임 처리시간 | 123.2969 ms |
| 좌측차선 처리시간 | 19.9718 ms |
| 우측차선 처리시간 | 19.9718 ms |
| 무한원점 결정처리시간 | 0.0323 ms |
| 장애물 검색 추정처리시간 | 79.121 ms |
| 표시 처리 시간 | 4.2 ms |

5. 결론

본 연구는 자동차의 자율 주행을 위한 필수적인 진행 방향 추정에 무한원점 추정 알고리즘을 이용하였다. 제한한 알고리즘에 의해 추정된 무한원점의 추정오차는 x, y에 대하여 $10^{-3} \sim 10^{-2}$ 내에서 이루어지므로 실제 도로 주행중에 적용함을 확인하였다. 또한, 주행중에 발생하는 장애물에 대한 검색과 추정할 수 있음을 확인하였다.

본 연구의 추후 과제는 알고리즘의 코드를 축소하여 처리속도를 단축하고 건물, 차량, 신호등, 노면 신호와 같은 장애물을 인식하고 그에 해당하는 행위를 할 수 있고 불규칙한 노면상태에서도 자율 주행할 수 있도록 알고리즘을 확대, 개발하는데 있다 하겠다.

참 고 문 헌

- [1] E.D.Dickmanns, B. Myśliwetz, and T. Christians, "An Integrated Spatio-Temporal Approach to Automatic Visual Guidance of Autonomous Vehicles", *IEEE Trans. on systems, man, and cybernetics*, vol. 20, No. 6, Nov/Dec 1990.
- [2] S.M. Smith and J.M. Brady, "ASSET-2: Real-Time Motion Segmentation and Shape Tracking", *IEEE Trans. Pattern Analysis And Machine Intelligence*, Vol. 17, No. 8, August, 1995.
- [3] Charles Thorpe, Martial H. Hebert, Takeo Kana de, and S.A. Shafer, "Vision and Navigation for the Carnegie-Mellon Navlab", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, May 1988.
- [4] Matthew A. Turk, David G. Morgenthaler, Keith D. Gremban, and Martin Marra, "VITS-A Vision System for Autonomous Land Vehicle Navigation", *IEEE Trans. Pattern analysis and Machine intelligence*, vol. 10, no. 3, may 1988.
- [5] 최성구, 노도환, "정방향 교정frame을 이용한 카메라의 교정 파라미터 추정에 관한 연구", *대한전자공학회논문지*, Vol. 33 B, No. 7, pp. 127-137, 1996.
- [6] 정준익, 최성구, 노도환, "Vision System을 이용한 자동차의 자율주행을 위한 진행방향추정에 관한 연구", *한국자동차공학회 춘계학술대회논문집*, Vol. 1, 499-504, 1996.