論文97-34S-11-14

무한원점을 이용한 주행방향 추정과 장애물 검출

(The Course Estimation of Vehicle using Vanishing Point and Obstacle Detection)

鄭 浚 益 **, 崔 成 求 *, 盧 度 煥 *

(Jun-Ik Jeong, Seong-Gu Choi, and Do-Whan Rho)

요 약

본 논문에서는 하나의 monocular vision 시스템을 이용하여 차량의 주행방향 추정과 장애물 검출 가능한 알고리즘에 대하여 제안하였다. 이 알고리즘은 무한원점의 특성을 이용하여 주행방향을 추정하고 통계적 방법을 이용하여 장애물을 검출한다. 제안하는 알고리즘은 차선 예측, 차선 추출, 주행방향 파라메타 추정과 장애물 검출의 4 단계로 구성되어있다. 차선의 존재 영역에 sub-window라는 소영역을 설정하여 빠른 처리속도와 차선 경계점의 연결성을 이용하여 높은 정확도를 갖도록 설계되어졌다. 또한, 간단한 통계적 방법을 이용하여 장애물을 검출하고 추정할 수 있는 새로운 방법도 제시하였다. 실제 도로의 VTR 영상을 알고리즘에 적용한 실험을 통하여 처리속도와 알고리즘의 정확성 및 제안하는 장애물 검출법의 유용성을 입증하였다.

Abstracts

This paper describes the algorithm which can estimate road following direction and detect obstacle using a monocular vision system. This algorithm can estimate the course of vehicle using the vanishing point properties and detect obstacle by statistical method.

The proposed algorithm is composed of four steps, which are lane prediction, lane extraction, road following parameter estimation and obstacle detection. It is designed for high processing speed and high accuracy. The former is achieved by a small area named sub-window in lane existence area, the latter is realized by using connected edge points of lane. We would like to present that the new method can detect obstacle using the simple statistical method.

The practicalities of the processing speed, the accuracy of the algorithm and proposing obstacle detection method, have been justified through the experiment applied VTR image of the real road to the algorithm.

Ⅰ. 서 론

급증하고 있는 자동차 사고의 예방 및 방지를 위해 자동차의 자율 주행에 관한 연구가 활발히 전개되고 있다. 이러한 자동차의 자율 주행을 위한 선결조건은 주행 가능한 지역에 대한 정확한 인식능력에 있다. 정확한 인식을 위한 시스템으로서 시각 시스템을 이용한 방법이 크게 주목되고 있다.

일반적으로, 시각 시스템을 이용한 자동차의 주행 환경 인식에 관한 연구는 자동차의 주행방향을 결정하기 위한 방법과 주행 환경 중에 발생되는 충돌 예방 및 대처를 위한 장애물 검출에 관한 연구로 분류할 수 있다. 주행방향을 결정하기 위한 연구로는 비포장된 도로와 포장된 도로로 구분하여 도로 환경 특성에 적

* 正會員, ** 學生會員, 全北大學校 電氣工學科

(Chonbuk Nat'l Univ. Dept. of Elec. Eng.)

※ 본 연구(과제번호 94-074)는 한국전력공사 지원에 의해 기초전력공동연구소 주관으로 수행되었습니다.

接受日字: 1997年3月26日, 수정완료일: 1997年11月3日

합한 방법으로 연구가 진행되고 있다.[1,2,3,4,5,6,7] Matthew[1]는 비포장도로에서 clustering과 segmentation등의 난점을 해결하기 위하여 Red Minus Blue 알고리즘을 이용하여 도로와 비도로영역을 이진화하였다. 이진화된 경계점들을 추적하여 도로의 경계를 추정하고 이를 3차원좌표계로 변환하여 주행방향을 추정하였다. Charles[2]와 Darwin[3]도 Matthew와 비슷한 방법으로 RGB color space를 이용하여 도로와 비도로로 분류하여 주행방향을 추정하였다. 그러나, Matthew, Charles, Darwin의 방법은 시골길 같은 노면의 상태가 불규칙한 도로에 적합하며, 고속차량에 적용하기 어려운 점과 시스템이 복잡하고 고가인 단점이 있다. E.D.Dickmanns[4]는 2대의 카메라를 이용하여 차선추정을 통한 주행방향과 노면에서 장애물의 접촉여부를 추정하는 알고리즘을 이용하여 최고속도 100 [Km/h] 에서 주행방향과 장애물을 검출하였다. 그러나, 시스템의 구성을 위하여 10톤의 차량과 고가의 장비를 사용하여 일반적인 차량에 적용하기에는 부적합하다. Shih-Ping Liou[5]는 무한원점을 이용하여 주행방향을 추정하였으나, 전제 조건이 실제 도로에 적합하지 않다. 根岸愼治[6]은 1개의 카메라를 이용하여 좌·우차선을 추정하여 주행방향을 결정하였다. 그러나, 처리 가능한 차량의 최고속도가 50 [Km/h] 이며, 장애물은 검출하지 않는다. 農宗千典[7]는 1개의 카메라가 설치된 차량으로 고속도로를 70~100 [Km/h] 로 주행하며 촬영한 영상을 이용하여, 좌·우차선의 경계와 무한원점을 통한 주행방향을 추정하였으나, 장애물의 검출은 이루어지지 않았다.

장애물 인식에 관해서는 주행중 장애물 검출, 거리 및 속도, 상대위치, 종류인식에 관한 연구가 진행되고 있다. 이러한 시각 시스템을 이용한 장애물 인식에 관한 연구로는 Stereo Vision을 이용한 방법, Optic Flow를 이용한 방법[8]과 1개의 monocular vision 시스템을 별도로 이용하는 방법[1,3,4] 등이 연구되고 있다. 그러나, 이러한 연구들은 자동차의 고속주행시 유용성이 결여되고 복수의 시스템을 필요로 하여 시스템 설치의 복잡성 등의 문제점이 있다.

따라서, 본 논문에서는 이러한 문제를 해결하고 실시간으로 처리할 수 있는 시스템으로서 1개의 monocular vision 시스템을 이용하여 주행환경을 인식하고 주행중 장애물을 추정할 수 있는 방법을 제안한다. 즉, 자율 주행을 위한 기초 연구로서 자동차의 주행방향과 장애물을 검출할 수 있는 알고리즘을 제안한다. 이 알고리즘은 주행 도로 환경이 포장된 직선도로와 완만한 커브를 가진 포장도로를 대상으로 주행방향은 차선의 직선 에지를 이용하여 무한원점[9]을 추정하고, 추정된 무한원점을 이용하여 주행방향을 추정할 수 있는 방법을 제안한다. 그리고, 장애물은 통계적 방법을 적용하여 추정할 수 있는 알고리즘을 제시한다. 제안된 알고리즘의 유용성을 입증하기 위하여 직선도로의 주행시에 연속적으로 입력되는 영상으로부터 무한원점을 추정하여 이로부터 얻어지는 자동차의 자세각과 장애물을 추정한 결과를 나타내었다.

Ⅱ. 시스템의 좌표계 구성 및 변환

자동차의 주행방향을 결정하는 파라메타를 추정하기 위해서는 주행도로와 자동차에 설치한 카메라와의 수리적 관계가 해석되어야 한다.

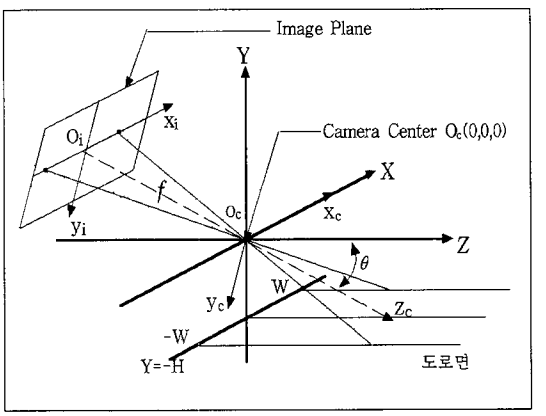


그림 1. 시스템 좌표계 I
Fig. 1. System coordinates I.

그림 1은 도로면, 자동차에 설치된 카메라와 공간 좌표계와의 좌표관계를 나타낸다. 차량에 고정한 카메라의 렌즈중심을 원점으로 하고 진행방향을 Z축으로 한다. 카메라 좌표계(x_c, y_c, z_c)의 원점과 공간 좌표계(X, Y, Z)의 원점을 동일하게 설정한다. 카메라 좌표계는 도로 중앙으로부터 H 높이에 있고, x_c축은 공간 좌표계의 X축과 동일하게 설정한다. 광축 z_c는 도로면과 θ의 각을 이룬다. 영상면(x_i, y_i)은 카메라 좌표계의 중심으로부터 f인 거리에 위치한 평면으로 오른손 직교 좌표계로 구성한다. 이때, 카메라의 영상 좌표계와 공간 좌표계와의 변환관계식은 식(1)과 같이 나타

낼 수 있다.

$$x_i = \frac{fX}{Z\cos\theta - Y\sin\theta} \qquad (1)$$
$$y_i = \frac{f(Y\cos\theta + Z\sin\theta)}{Z\cos\theta - Y\sin\theta}$$

식(1)은 3차원 공간 좌표계와 2차원 영상 좌표계의 관계를 나타내는 식으로서 영상면의 정보로부터 3차원 공간상의 정보를 해석할 수 있는 기본식이다. 본 논문에서는 식(1)과 같이 도로와 카메라의 관계를 해석하여 주행방향 파라메타를 설명하고, 무한원점의 원리를 이용하여 간명하게 추정할 수 있는 방법을 기술한다.

Ⅲ. 무한원점의 개념과 주행방향 파라메타

1. 무한원점의 개념 및 성질

본 절에서는 자동차의 주행방향 추정에 이용되는 무한원점의 개념과 성질에 관해 기술한다. 일반적으로 3차원 공간상의 점 (x_0, y_0, z_0)를 통과하는 직선 L은 다음과 같은 요소의 집합으로 나타낼 수 있다.

$$\begin{aligned} x &= x_0 + tv_x \\ y &= y_0 + tv_y \quad (t \in R(\text{실수}), -\infty < t < \infty) \\ z &= z_0 + tv_z \end{aligned} \qquad (2)$$

이때, 식(2)의 $v = (v_x\ v_y\ v_z)^T$은 직선 L의 단위 방향 벡터이고, t는 직선위의 각 점사이의 거리비를 나타낸다.

직선 L위의 점 (x, y, z)에 대응되는 영상면상의 점을 $(\bar{x}, \bar{y})$라 하면 식(1)로부터

$$\bar{x} = f\frac{x}{z} = f\frac{(x_0 + tv_x)}{(z_0 + tv_z)} \qquad (3)$$
$$\bar{y} = f\frac{y}{z} = f\frac{(y_0 + tv_y)}{(z_0 + tv_z)}$$

으로 나타낼 수 있다.

이때, $t \to \infty$에서 $(\bar{x}, \bar{y})$의 점 (x_∞, y_∞)은

$$x_\infty = \lim_{t\to\infty} \bar{x} = f\frac{v_x}{v_z} \qquad (4)$$
$$y_\infty = \lim_{t\to\infty} \bar{y} = f\frac{v_y}{v_z}$$

와 같이 구할 수 있다. 식(4)의 점을 영상면상의 무한원점이라 한다. 식(4)를 정리하면,

$$v_x = v_z\frac{x_\infty}{f} \qquad (5)$$
$$v_y = v_z\frac{y_\infty}{f}$$

와 같이 나타낼 수 있다. 식(5)는 영상면상에서 무한원점 $(x_\infty, y_\infty, z_\infty)$을 구할 수 있으면, 3차원 공간상의 직선에 대한 방향 벡터 $(v_x\ v_y\ v_z)^T$를 구할 수 있는 성질을 유추할 수 있다. 본 논문은 이 성질을 이용하여 자동차의 주행방향 추정 알고리즘을 제안한다.

2. 무한원점과 주행방향 파라메타 관계

그림 2는 폭이 2W인 도로를 주행하는 자동차의 진행 과정을 평면으로 도시한 것으로서, 그림은 시각 t_{n-1}에서 Z_{n-1}방향으로 진행하고 시각 t_n에서 Z_n방향으로 진행하는 과정을 나타내고 있다.

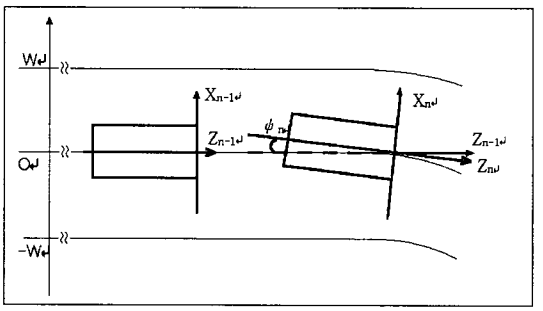


그림 2. 시스템 좌표계 II
Fig. 2. System coordinates II.

즉, Z_n , Z_{n-1}은 자동차의 진행을 결정하는 방향파라메타를 나타낸다. 여기에서는 Z_n 과 Z_{n-1}이 이루는 사이각 ψ를 자동차의 자세각(posture angle)이라 한다. 이 자세각이 주행방향을 결정하는 파라메타이다. 본 논문에서 자동차의 진행 방향을 결정하는 자세각을 추정하기 위하여 무한원점의 원리를 이용한다. 평행한 좌·우 차선을 카메라의 영상면상에 투영시키면 투영된 차선의 연장선은 어느 한 점에서 교차하는 무한원점을 얻는다. 즉, 영상면상에서 무한원점을 (v_x, v_y)라 하면 3차원 공간상의 차선 방향 벡터는 (v_x, v_y, f)로 구할 수 있다. 이 성질을 이용하여 그림 2에 나타난 것과 같이 자동차의 진행방향을 결정하는 자세각 ψ는 영상 처리시간 T의 간격을 가지는 영상시스템으로부터 연속적으로 입력되는 영상면상의 무한원점들을 내적관계로부터 식(6)과 같이 추정할 수 있다.

$$\psi_n = \cos^{-1}\frac{v_{xn}v_{xn-1} + v_{yn}v_{yn-1}}{\sqrt{v_{xn}^2 + v_{yn}^2}\sqrt{v_{xn-1}^2 + v_{yn-1}^2}} \qquad (6)$$

이때, ψ의 크기는 도로의 곡률에 의해 좌우된다. 직선도로에서는 ψ가 0에 근접하며 실제로 도로가 사라지

지 않고 90° 의 급커브를 예외로 하면 ϕ는 $0 \le |\phi| < 90$의 범위 안에 존재한다.

이상과 같이 주행방향 파라메타의 추정은 차선에 의해 형성되는 영상면상에서의 차선 직선식들을 이용하여 무한원점을 구한다. 그리고, 처리시간 T간격으로 입력되는 무한원점을 이용하여 식(6)과 같이 주행방향 파라메타인 자세각을 추정한다.

Ⅳ. 주행방향 추정 및 장애물 검출 알고리즘

본 절에서는 앞에서 기술한 무한원점의 원리를 이용하여 자동차의 자세각을 추정하는 주행방향 추정 알고리즘 및 장애물 검출 알고리즘에 대해서 기술한다. 주행방향을 결정하기 위한 무한원점은 도로면상의 평행한 차선이나 도로영역을 판별할 수 있는 차선등과 같은 정보를 이용하여 영상 처리 기법과 통계적 방법을 적용하여 추정한다. 그리고, 장애물 검출은 영상면상에 영역을 설정하고, 설정된 영역별로 통계적 기법을 이용하여 장애물을 검색 할 수 있는 알고리즘을 제안한다.

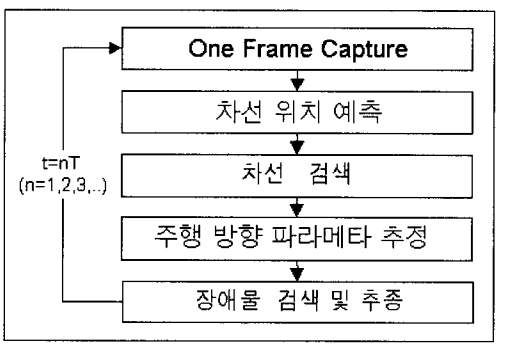


그림 3. 주행방향 추정
Fig. 3. Course estimation.

제안하는 알고리즘은 주행방향 추정을 위한 알고리즘과 장애물 검출을 위한 알고리즘으로 구성되며 흐름도는 그림 3에 나타냈다. 주행방향 추정을 위한 알고리즘은 차선 검출의 편의성 및 처리시간의 단축을 위한 차선 위치 예측과 예측된 위치로부터 차선을 검색하기 위한 차선 검색으로 구성된다. 그리고, 검색된 차선들로부터 무한원점을 추정하고 이로부터 자세각을 구하기 위한 주행방향 파라메타 추정으로 구성된다. 장애물 검출 알고리즘은 통계적 기법을 이용하여 장애물을 검출 및 추정하며 3단계로 구성된다. 이상과 같은 각각의 알고리즘 내용은 다음과 같다.

1. 차선 위치 예측

주행방향 파라메타의 결정에 이용되는 무한원점 추정을 위해서는 반드시 차선의 직선식이 필요하다. 여기에서는 차선을 신속하게 검출하기 위한 차선 위치 예측에 대해서 기술한다.

일반적으로 영상면상에서의 차선 위치는 일정한 범위에 위치한다. 이러한 점을 착안하여 본 논문에서는 차선 탐색 범위를 sub-window로 제한하여 차선의 위치를 예측함으로서 처리시간을 단축하고 잡음에 강한 알고리즘을 실현하려 한다.

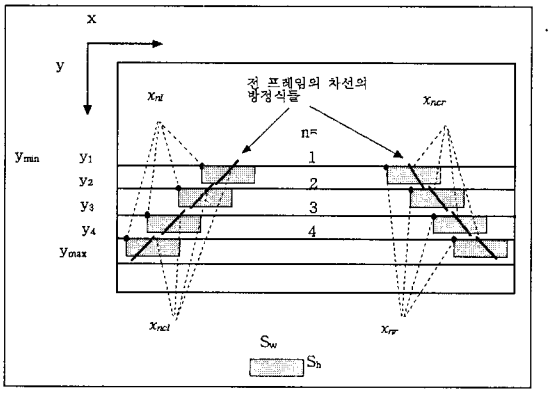


그림 4. 후보탐색영역
Fig. 4. A search of candidate area.

그림 4에서와 같이 sub-window의 폭과 높이가 S_w, S_h일 때, y좌표들은 y_{min}과 y_{max}사이의 영역을 4등분하여 식(7)과 같이 설정한다.

$$y_n = y_{min} + (n-1) \cdot (y_{max} - y_{min})/4 \quad \text{단, } n = 1, 2, 3, 4 \tag{7}$$

x좌표들은 차선의 변동에 따라 능동적으로 조정될 수 있도록 하였다. 먼저 식(8)과 같이 전 처리프레임에서 얻어진 좌우차선방정식의 기울기 a와 y절편값 b들을 이용하여 x값을 구한다.

$$x_{ncl} = a_{nl} \cdot y_n + b_{nl} \quad (\text{좌측차선})$$
$$x_{ncr} = a_{nr} \cdot y_n + b_{nr} \quad (\text{우측차선}) \tag{8}$$
$$\text{단, } n = 1, 2, 3, 4$$

설정될 sub-window의 x좌표는 식(9) (10)과 같다.

$$x_{nl} = x_{ncl} - S_w/2 \quad (9)$$

$$x_{nr} = x_{ncr} - S_w/2 \quad (10)$$

차선의 위치 예측에는 차선중 파선으로 이루어진 경우와 같이 설정된 sub-window내에 차선이 존재하지 않거나, 장애물로 인하여 경계점들이 추정되지 않아 직선식이 만들어지지 않는 경우가 있다. 이때, 다음 프레임에서 차선의 위치 예측을 위하여 차선이 급격하게 사라지지 않는다는 전제하에서 ①과 ②의 각 직선식의 중간 간격에 해당하는 영역에 차선 검색영역을 설정하기 위한 방정식을 생성한다.

① 시간 (n-1) 프레임에서의 차선의 방정식
② 현 프레임에서 추정된 좌·우 각각의 전체 경계값들을 이용한 좌·우측의 전체 차선의 방정식

2. 차선 검색

여기에서는 예측된 차선의 위치에서 차선을 검색하여 차선의 직선식을 구한다. 차선 검색의 신속성과 잡음에 강한 알고리즘 구현을 위해 높이×폭이 20×50 [pixel]인 크기를 가지는 sub-window를 좌·우로 각각 4개씩 설정하여 이용하였다. 설정된 각각의 sub-window에서는 영상 처리 기법을 이용하여 차선의 에지를 추출하고 에지들에 의한 차선의 직선 방정식을 구한다. 그림 5는 영상면에서의 차선 검색 알고리즘 흐름도이다. 각 단계별 내용은 다음과 같다.

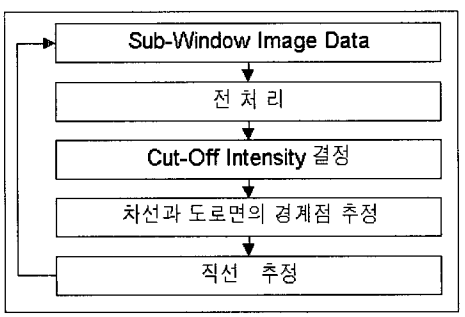


그림 5. 차선 검색
Fig. 5. Lane detection.

단계 1 : 전 처리

이 단계에서는 영상면에 평균 intensity보다 너무 어둡거나 밝은 부분에 대해 보상하기 위한 평활화 처리를 한다.

평활화 처리된 영상은 보편적으로 이용하는 수직의 2차 미분 필터인 sobel 필터를 이용하여 에지추출을 시행한다. sobel 필터는 임의의 화소점 (i, j)에서 intensity가 $I_{i,j}$ 일 때 (i, j)점에서 gradient의 크기와 방향을 계산함으로서 도로 영상에서의 차선 에지를 추출한다. 이용하는 sobel 필터의 미분 연산자는 식(11)과 같다. 본 논문에서는 각각 좌·우차선의 좌측 경계점만을 이용하기 위하여 양(+)의 값만을 적용한다.

$$S_x(i,j) = I_{i+1,j+1} + 2I_{i+1,j} + I_{i+1,j-1} - I_{i-1,j+1} - 2I_{i-1,j} - I_{i-1,j-1} \quad (11)$$

단계 2 : 도로 영상에 대한 cut-off intensity 결정

단계 2에서는 필터 처리된 sub-window 영역의 화소값들을 이용하여 2치 농담영상으로 변환하는 작업을 행한다. 본 논문에서는 2치값을 뚜렷하게 하기 위하여 0과 200의 값을 사용한다. 영상면에서 차선에 대한 최소 intensity 한계치를 (12)식과 같이 설정한다.

$$U = \mu - T\sigma \quad (12)$$

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

여기에서, 사용자 파라메타 T는 차선의 최소한의 밝기값에 의해 결정된다. 실험적으로 -5~ -7사이의 값이 적당하다.

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. 추정방법은 (i,j)의 2진화값을 $B_{i,j}$이라 할 때, 이 점을 중심으로 식(13)과 같이 8방향중 3방향의 연결성분이 존재하는 경우의 점만을 추출하고 식(12)와 (14)의 조건을 적용하여 경계점을 추출한다. 이러한 화소를 sub-window 전체의 화소에 대해 수평과 수직선을 따라 반복하여 경계점을 추정한다.

$$G(i,j) = B_{i-1,j-1} + B_{i,j-1} + B_{i+1,j-1} + B_{i-1,j} + B_{i,j} + B_{i+1,j} + B_{i-1,j+1} + B_{i,j+1} + B_{i+1,j+1} \quad (13)$$

$$\text{경계점} : \begin{bmatrix} True : G(i,j) \geq 3U \\ False : G(i,j) < 3U \end{bmatrix} \quad (14)$$

단계 4 : 최소이승법에 의한 직선 추정

단계 4에서는 단계 3에서 추정된 차선의 에지 화소점들을 최소이승법(least squares fitting method)을 이용하여 차선에 대한 직선식을 추정한다. 일반적인 직선의 식은 식(15)로서, 기울기 m과 y축의 교점 b는 에지점들에 대해 최소이승법을 적용하여 식(16)와 같이 구할 수 있다.

$$y = mx + b \tag{15}$$

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

$$b = \frac{\Sigma y_i - m\Sigma x_i}{n} \tag{16}$$

(단, $i=1,2,3,\ldots,n$)

3. 주행방향 파라메타 추정

이 부분은 2단계로 구성되어 있으며, 단계 1에서는 자동차의 주행방향을 결정하기 위해 앞에서 구해진 직선들을 이용하여 무한원점을 구한다. 그리고, 단계 2에서는 식 (6)을 이용하여 자세각을 추정한다.

단계 1 : 무한원점 추정

차선 검색의 식(15)에서 구해진 영상면상의 좌우직선식들을 이용하여 교점, 즉, 무한원점의 후보점들(v_{cx}, v_{cy})을 추정한다. 가장 적합한 무한원점을 추정하기 위하여 무한원점의 후보점들을 식(17)에 대입한다.[10] 여기에서 k 는 좌우직선식들의 총갯수이다.

$$f(v_{cx}, v_{cy}) = \sum_{i=1}^{k} \frac{m_i v_{cx} + v_{cy} + b_i^2}{v_{cx}^2 + v_{cy}^2 + 1} \tag{17}$$

식(17)에 의해 구해진 값들중에서 최소값, 즉, 각 좌·우차선의 직선에 가장 근접한 점을 추정하여 무한원점 v_x, v_y를 추정한다.

단계 2 : 주행방향 파라메타 추정

자동차의 주행중에 연속적으로 입력되는 영상으로부터 단계 1에서와 같이 무한원점이 추정되면 II절에서 설명한 자동차의 주행방향 파라메타인 자세각 ϕ를 식(6)으로서 추정할 수 있다.

4. 장애물 검출 알고리즘

장애물 검출 알고리즘은 다음과 같은 전제조건하에서 수행된다. 먼저 관심차량은 카메라 앞에 나타나는 선행차량이며, 또한 도로면과 관심차량 사이에는 구분가능한 충분한 밝기 분포를 갖는다.

알고리즘의 구성은 3단계로 구성되어 있다. 단계 1에서는 장애물 검색을 신속히 하기 위하여 검색영역, 즉, sub-window를 구분하여 설정하고, 단계 2에서는 단계 1에서 설정된 영역별로 장애물을 판단하기 위한 영상정보를 분석하여 무게값을 부여하는 과정이다. 단계 3에서는 부여된 영역별 무게값의 평가에 의해 장애물을 판단하여 검출하는 단계이다. 각 단계별 내용은 다음과 같다.

단계 1 : 각 구간별 소검색영역 설정

본 단계에서는 그림 6.과 같이 위의 차선 검색에서 구해진 좌·우차선의 방정식을 이용하여 식(18)~식(20)과 같이 y_{min}, y_{max} 각각의 위치에 좌측, 우측, 중앙의 sub-window의 x좌표를 설정한다.

$$x_{lo} = a_l \cdot y + b_l + w_{lo} \tag{18}$$

$$x_{ro} = a_r \cdot y + b_r + w_{ro} \tag{19}$$

$$x_{mo} = (a_l + a_r) \cdot y/2 + (b_l + b_r)/2 + w_{mo} \tag{20}$$

(w_{lo}, w_{ro}, w_{mo} : 사용자 파라메타)

여기에서는 sub-window의 크기를 10×10 [pixel]로 설정하였다.

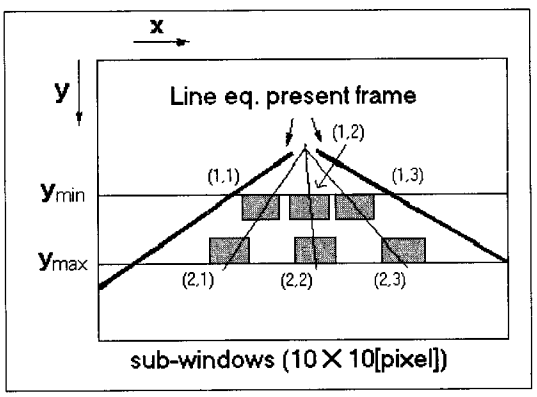


그림 6. 장애물 검색위치 지정
Fig. 6. Determine obstacle searching position.

단계 2 : 소검색영역의 정보값 산출

설정된 소검색 영역에서 장애물을 검출하기 위한 방법으로 검색영역의 mean, median값을 구하고 저장한다.

단계 3 : 정보값에 의한 장애물 판단

단계 2에서 산출한 mean값과 median값의 차이를 이용하여 시간 n일 때의 각 sub-window별 무게값을

식(21), (22), (23)로서 구한다. 장애물 판단은 각 sub-window별 무게값으로 식(24)에 의해 이루어진다. 장애물을 검출하기 위하여 sub-window내의 노면의 밝기를 식(22)와 같이 평균값을 이용하여 추정한다. sub-window는 작은 영역이기 때문에 장애물 삽입시 장애물이 포함하고있는 밝기의 분포로 인하여 평균값과 중심값과의 차이는 크게 나타난다. 한 sub-window의 평균값과 중심값의 차가 적을 경우에는 장애물이 존재하지 않을 확률이 높으며, 차이값이 클 경우에는 장애물이 삽입되었을 확률이 많다. 장애물로 판단되었을 때, 그 장애물은 다음 프레임에서 갑자기 사라지지 않아야 한다.

장애물과 노면과의 무게값 구분을 위하여 식(21)과 같이 중심값을 추정하여 전 프레임과 비교한다. 만약, 현 프레임의 검색영역에서 급격한 중심값의 변화가 생긴다면, 밝은 장애물일 경우에는 큰 양(+)의 값을, 어두운 장애물일 경우에는 큰 음(-)값을 나타낸다. 즉, 식(23)에서 β_n값이 급격히 커진다. t_r은 밝고 어두운 장애물의 경계를 설정하기 위한 사용자 파라메타이다. 따라서, t_r은 식(21)의 Ob_n값에 따라 조건적으로 구별하여 선택한다. 실험적으로 t_r은 Ob_n 값이 (+) 일 때, -15 ~ -20 사이값으로 설정하며, (-)일 때에는 -35 ~ -45의 범위로 설정한다. 특히, $|Ob_n| < 5$일 경우에는 거의 변화가 생기지 않는다고 간주하고 t_r값을 -8 ~ -12사이값을 설정한다. 이 t_r값은 카메라의 분해능, 도로면과 카메라가 이루는 각과 실제 공간상의 거리에 대하여 투영되는 밝기값의 변화에 따라 변경된다.

$$Ob_n = median_n - median_{n-1} \tag{21}$$

$$\alpha = \frac{1}{t_e}\sum_{i=1}^{t_e}\{mean_{n-i}\} \tag{22}$$

$$\beta_n = |\alpha - mean_n| + t_r \tag{23}$$

$$Obj_n : \begin{bmatrix} Obstacle: \{Ob_n > \beta_n\} \text{ or} \\ \{Ob_n < \beta_n \text{ and } Obj_{n-1} = Obstacle\} \\ Non-obstacle: \{Ob_n < \beta_n\} \end{bmatrix} \tag{24}$$

(t_e, t_r : 사용자 파라메타)

여기에서, 사용자 파라메타 t_e는 주행중 도로정보값의 저장갯수이다. α는 도로면의 상태를 나타내며 장애물로 판단된 경우의 프레임은 계산에서 제외시킨다.

전 프레임에서 장애물로 판단되었을 경우에는, 식(24)과 같이 급격한 무게값들의 변화가 없을 때, 장애물로 간주하고, 변화가 발생한 후부터 식(22)의 α값을 계산한다. 이와 같은 연산을 그림 6에서의 sub-window (1,1), (1,2), ... , (2,3)에 대하여 순차적으로 행하여 장애물을 검출한다.

V. 실험 및 고찰

이상과 같이 본 논문에서는 차선의 에지에 의해 추정된 무한원점으로 주행방향을 결정하는 파라메타인 자동차의 자세각과 주행중 장애물을 검출할 수 있는 알고리즘을 제안하였다. 주행방향을 결정하는 파라메타인 자세각의 정확성은 무한원점 추정의 정확도에 기인한다. 무한원점의 정확도를 평가하기 위해서는 제안한 알고리즘에 의해 추정되어진 무한원점과 비교하기 위하여 무한원점의 이론값을 필요로 한다. 무한원점의 이론값은 Ⅲ장에서 기술한 것과 같이 공간상에 평행한 두 개의 직선을 이용하여 추정한다. 이와 같은 조건을 만족하기 위해서는 공간상의 평행한 두 직선이 존재해야 한다. 그러나, 실제도로에 있어서 곡선도로의 경우는 무한원점의 이론값이 변화하므로 알고리즘의 유용성을 입증하기 곤란하고, 직선도로에서의 좌우차선은 공간상의 평행한 두 직선으로 간주할 수 있으므로 무한원점 추정조건을 만족하여 무한원점 오차 추정에 이용할 수 있다.

따라서, 본 논문에서는 직선도로에서 추정된 무한원점의 정확도를 평가하여 알고리즘의 유용성을 입증하고, 곡선도로, 교각, 오르막, 내리막도로에서는 차선 추정 결과만을 제시한다. 또한, 장애물 검출 알고리즘의 유용성을 입증하기 위하여 실제 도로 주행시 삽입되는 장애물을 검출한 결과를 제시한다. 실험 조건은 초점거리가 8 [mm] 인 카메라를 지상높이 1.5 [m] 에서 자동차의 중앙에 설치하여 국도를 100 [km/h] 로 주행하여 카메라로 촬영하였다. 실험에 이용한 시스템은 1/30 [s] 의 frame grabber능력을 갖는 Image Process Board(Data Translation DT-2871)를 IBM-AT(Intel 486-DX 53.5MHz)에 설치하여 이용하였다.

이와 같은 시스템의 입력영상은 운전자의 운전미숙에 의한 차량의 좌우이동, 자동차의 구조적 진동 및 노면상태 등으로 인한 계통적 오차를 포함한다. 운전미숙에 의한 차량의 좌우이동은 차량에 시스템을 탑재하여 감소시킬 수 있으며, 차량의 진동과 노면상태등의 진동에 의해 발행하는 오차는 카메라에 mass-

spring system과 같은 기계적 진동필터를 이용하여 감소시킬수 있다. 그러나, 본 논문에서는 이러한 오차를 감소시키기 위해 차량에 탑재하지 않고, 단지 알고리즘의 유용성과 정확성을 입증하기 위하여 카메라 교정을 통해 계통적 오차를 감소시켰다. 이러한 교정을 행하기 위해서는 논문[11]에서와 같이 기준이 되는 무한원점의 좌표값이 구해져야만 한다. 이와 같은 기준이 되는 무한원점의 좌표값을 구하기 위하여 주행에 앞서 실제직선도로의 좌우차선상에 4개의 정방형 기지점을 3.1×3.1[m]의 간격으로 설치하고, 이 정방형 프레임을 이용하여 좌표값(269.0349, 252.5342)과 카메라의 내부파라메타를 추정하였다.

교정에 의한 무한원점 좌표값은 한 프레임의 도로영역에 대하여 sobel 필터처리후에 좌우차선의 좌측 경계 좌표값과 카메라의 내부파라메타를 이용하여 계통적 오차가 포함된 무한원점을 1초간격으로 산출하였으며, 입력영상에 맞추어 선형보간법(linear interpolation)을 적용하였다.

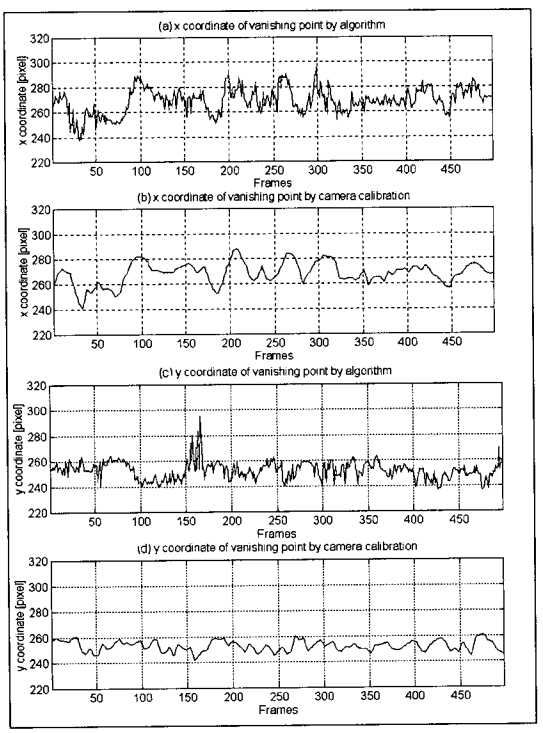


그림 7. 알고리즘((a), (c))과 교정((b), (d))에 의한 무한원점의 x, y좌표

Fig. 7. X, y coordinates of vanishing point by algorithm((a), (c)) and calibration((b), (d))

그림 7의 (a), (b), (c), (d)는 직선도로 주행중 입력되는 연속영상의 500 프레임에 대한 알고리즘과 카메라 교정에 의한 무한원점 추정결과를 나타낸 것이다. 제시되는 결과들은 100회 반복하여 확률적 기법에 의한 평균 처리한 결과이다. 그림에서 알 수 있는 바와 같이 x좌표의 경우에는 알고리즘 추정값과 교정에 의한 추정값이 유사한 형태를 갖지만, y좌표의 경우에는 그렇지 못하다. 이것은 차량, 또는 카메라의 상하진동에서 기인하는 현상이라 생각할 수 있다.

이상과 같은 교정값을 이용하여 입력영상 자체내에 포함되어 있는 계통적 오차를 감소시킨 후의 무한원점 x, y좌표의 절대오차 추정값을 그림 8에 나타내었다

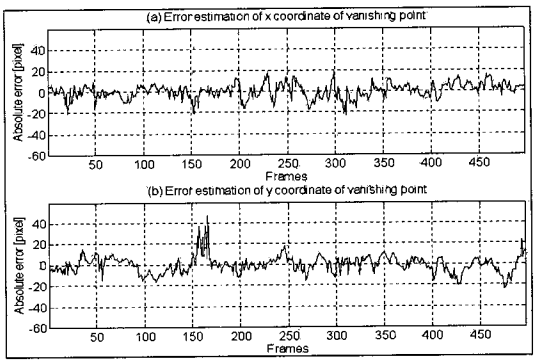


그림 8. 무한원점 x, y좌표의 절대오차추정

Fig. 8. Absolute error estimation of x, y coordinate of vanishing point.

그림 8의 절대오차 추정값은 결과적으로 무한원점의 기준 좌표값과 실험에 의한 추정 좌표값의 차를 의미한다. 그림에서와 같이 추정된 무한원점 좌표값의 오차는 -20~20[pixel] 범위 하에 존재한다.

그림 8의 프레임 150과 160사이에서 횡단보도로 인하여 x좌표에 비해 y좌표만이 큰 오차를 가지게 된다. 이러한 원인는 횡단보도의 좌우측 경계선이 검출되어 차선간의 폭이 좁아짐으로 해서 x좌표값의 변동보다 y값의 변동이 크기 때문이다. 이러한 현상은 그림 9에서와 같이 자동차의 자세각 ϕ의 큰 변동의 원인이 된다.

잡음이 삽입된 차선의 경계점에 의한 오차를 줄이기 위해서는 차선에 대한 Robust line fitting법을 이용하여 해결할 수 있지만, 알고리즘의 처리속도도 간과할 수 없다. 본 논문에서는 8방향 3연결법을 이용하여 잡음성분을 제거하고, 최소자승법만을 적용하여 처리시간과 정확도를 높혔다.

그림 9는 이상과 같이 얻어진 무한원점에 의해 추정된 주행방향 파라메타인 자동차의 자세각 ψ의 절대오차를 나타낸 그림이다. 직선도로에서 ψ의 이론값은 0 [°]이다. 그림에서 나타난 바와 같이 ψ의 오차는 -4~4 [°]의 범위 내에 분포함을 알 수 있다. 대부분의 오차는 ±2 [°] 안에 존재하지만 장애물 삽입, 운전자의 운전미숙등에 의한 무한원점 x, y좌표의 변동이 심한 영역에서 ψ의 오차가 커짐을 알 수 있다.

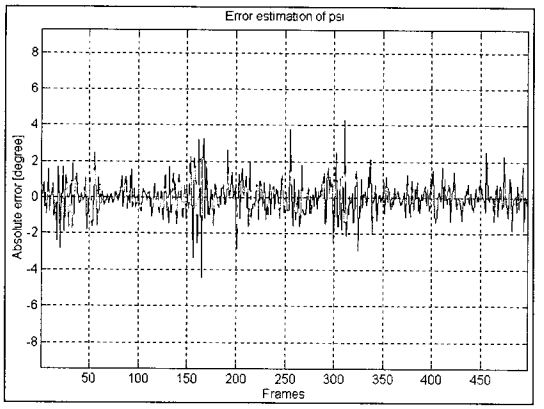


그림 9. 자세각 ψ의 절대오차추정
Fig. 9. Absolute error estimation of posture angle ψ.

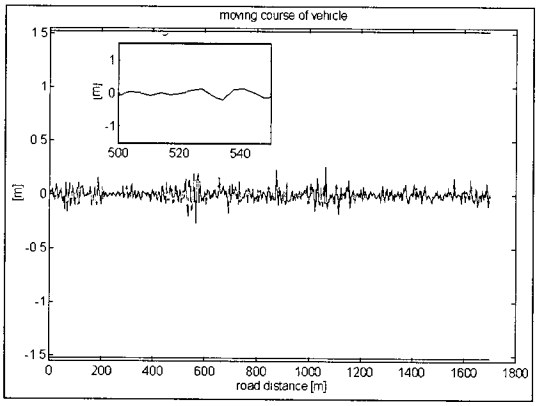


그림 10. 자세각 ψ에 의한 차량의 이동궤적
Fig. 10. Moving course of vehicle by posture angle ψ.

그림 10은 실제 도로상에서 차량의 이동궤적을 나타낸 것이다. 차선간의 폭 2W는 3.1 [m]이고 표 1. 에서와 같이 한 프레임 처리시간 123.2969 [ms]이다. 100 [Km/h]의 속도로 500 프레임동안에 차량이 이동한 거리는 약 1700 [m]이다. 그림에서와 같이 차선간의 중심축을 따라 ψ가 변동하였음을 알 수 있다.

그림 11은 sub-window를 구성하고 식(11)의 sobel operator를 적용하여 차선의 에지를 추출한 결과를 나타낸 그림이다.

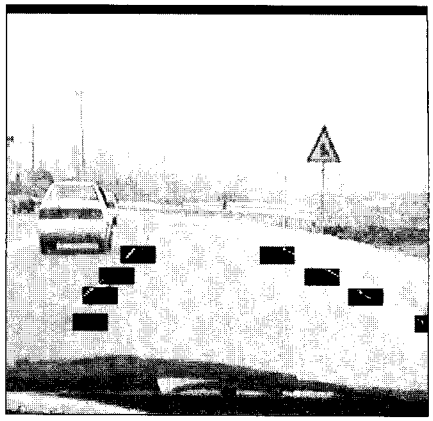
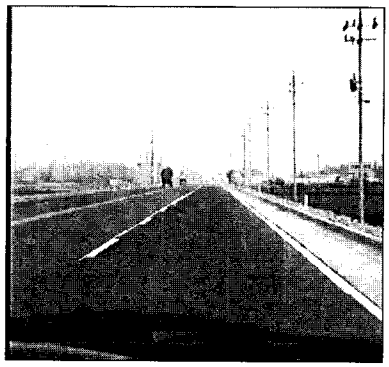
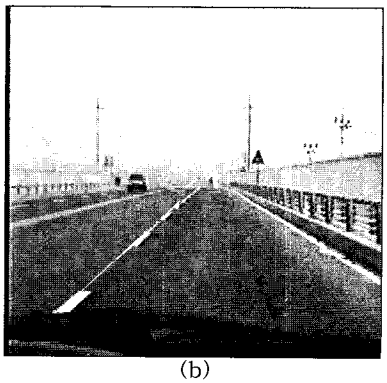


그림 11. 차선의 경계 검출예
Fig. 11. Example of lane edge detection.

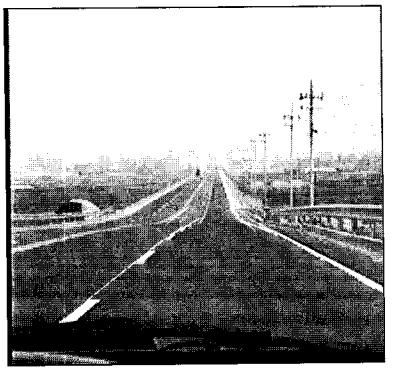
그림 12는 각종 도로에서의 영상을 제안한 알고리즘에 적용하여 차선의 추정 결과를 직선으로 나타낸 그림들이다. 추정된 직선은 우측 실선의 경우뿐만 아니라 좌측 점선 차선의 경우에도 정확한 직선을 추출함을 보여준다.



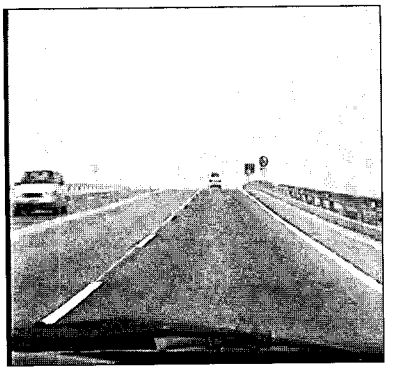
(a)



(b)



(c)



(d)

그림 12. 각종도로에서의 차선추정예
Fig. 12. Sample results of lane estimation of each road types.

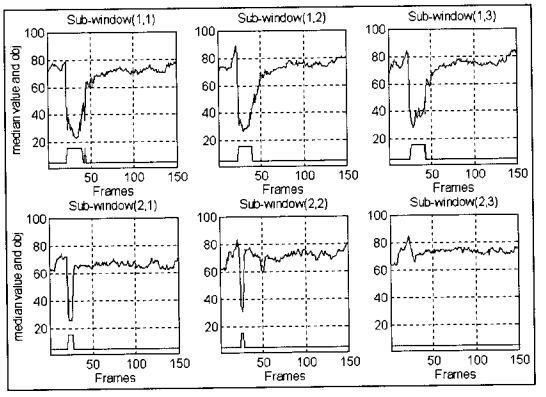


그림 13. 각 sub-window에서의 중심값 변화와 장애물 판단 결과
Fig. 13. Median value variation and obstacle detection at each sub-window.

그림 13은 주행중에 삽입되는 장애물의 검출과 추정 결과를 sub-window 상하 각각의 위치대로 나타낸 그림이다. 장애물을 검출하고 추정하기 위하여 제안한 식(21)에서 식(24)까지의 정확성을 입증하기위하여 실제 진행 프레임동안의 각 sub-window의 median값과 식(24)의 Obj의 참, 거짓을 계단함수로 나타내었다. 이 그림에서 알 수 있듯이 주행중 장애물 삽입시에 각 window의 median값이 급격한 변화를 보이고, 이를 식(21)~식(24)에 적용하여 장애물을 검출 및 추정할 수 있음을 알 수 있다.

sub-window(2,1)에서 가장 먼저 장애물의 삽입을 검출하며, 프레임이 진행하는 동안에 sub-window (1,1), (1,2), (2,2), (1,3)의 순으로 검출된다. 차량으로부터 장애물이 멀어지는 경우에 1렬의 sub-window를 벗어나기 전까지 장애물이 계속 추정됨을 알 수 있다. sub-window(2,3)에는 장애물이 위치하지 않았다. 주행중 노면에 존재하는 흙, 타이어 자국등의 잡음에 의한 장애물 검출 결과도 그림 13에 포함되어 있다.

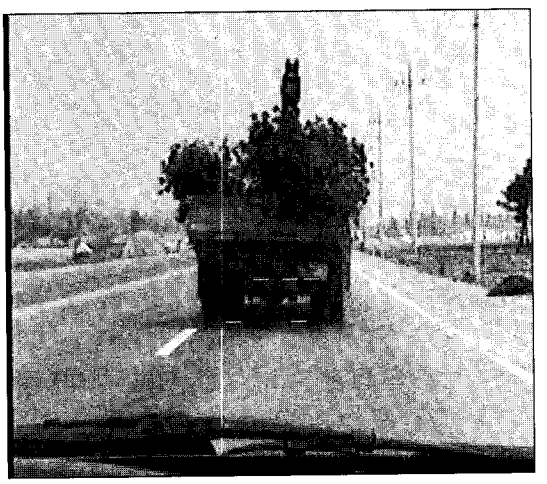


그림 14. 장애물 검출예
Fig. 14. Example of obstacle detection.

프레임 약 15에서부터 각 sub-window의 밝기값이 전체적으로 상승한다. 이 경우는 노면에 흙이 넓게 분포된 경우이다. 이때에 그림 14와 같은 장애물이 서서히 삽입된다. 식(24)에 의해 검출된 결과로서 장애물을 판단한다. sub-window(2,2)의 50프레임 부근에서 밝기값이 감소한 후 다시 증가한다. 이는 작은 영역을 차지하는 타이어 자국에 의한 결과이다. 이상적으로 이와 같은 현상은 sub-window(1,2)에서도 나타나야 한다. 그러나, 실제적으로 상·하 sub-window들의 실제공간상의 거리차에 의해 투영되는 타이어자국의 밝기차가 존재하며, sub-window의 중심값에 의해 잡음이 제거된다. 또한, 직선식에 의한 sub-window의

좌우 위치 변동과 타이어 자국이 약간 휘어진 경우에는 상하 sub-window에서 유사하지 않은 경우가 발생하게 된다.

그림 14는 검출된 장애물을 sub-window에서 표시한 결과이다.

1000 프레임 동안의 알고리즘 평균처리시간을 표 1에 제시했다. 제시된 결과로부터 100 [Km/h]의 속도로 주행하였을 때, 한 프레임의 처리시간 동안 3.42 [m]를 진행한다는 것을 알 수 있다. 즉, 제안한 알고리즘을 적용하면 실시간 처리가 가능함을 알 수 있다.

표 1. 한 프레임의 평균 처리시간
Table 1. Each average processing time.

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

VI. 결 론

본 논문은 자동차의 자율 주행을 위한 필수적인 주행방향 추정에 무한원점의 성질을 이용하였다. 제안한 알고리즘에 의해 추정된 무한원점의 추정 정확도는 x, y에 대하여 오차범위가 ±20 [pixel] 내에서 이루어지며 이에 따라 주행방향 파라메타인 자세각 ψ가 ±2 [°]의 분포를 가지므로 실제 도로 주행중에 적용할 수 있음을 증명하였다. 그리고, 주행중에 발생하는 장애물에 대한 검출과 추정이 가능한 알고리즘을 제안하였으며 이를 실제도로영상에 적용하여 유용함을 확인하였다. 또한, 본 논문에서 제안하는 알고리즘을 적용하면 실시간 처리 가능함을 입증하였다.

본 논문의 추후 과제는 알고리즘의 코드를 축소하여 처리속도를 더욱 단축하고 건널목, 차량, 신호등, 노면신호와 같은 다양한 장애물과 교통신호를 인식하여 처리가능하며 비포장도로와 같이 불규칙한 노면상태에서도 주행방향을 추정할 수 있도록 알고리즘을 확대, 개발하는데 있다.

참 고 문 헌

[1] 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.

[2] Charles Thorpe, Martial H. Hebert, Takeo Kanade 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.

[3] Darwin Kuan, Gary Phipps, and A. -Chuan Hsueh, "Autonomous Robotic Vehicle Road Following", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, no. 5, Sept., 1988.

[4] E. D. Dickmanns, B. Mysliwetz 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.

[5] Shih-Ping Liou and Ramesh C. Jain, "Road Following Using Vanishing Points", *Computer Vision, Graphics, And Image Processing*, vol. 39, pp. 116-130, 1987.

[6] 根岸愼治, 千葉昌孝, 小澤愼治, "高速道路における車輛姿勢角推定に基づく道路エッジの 自動追跡", 電子通信學會論文誌 D-II, vol. J77-D-II, no. 5, pp. 931-939, 1994.

[7] 農宗千典, 小澤愼治, "高速道路走行畵像からの消失点推定に基づく實時間白線檢出", T.IEE Japan, vol. 113-C, no. 2, 1993.

[8] 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.

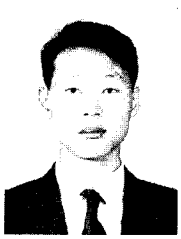
[9] 中谷, 北僑, "反復演算による無限遠点抽出", 電子通信學會論文誌, vol. J68-D, no. 8, 1985.

[10] William Chen and Bernard C. Jiang, "3-D camera calibration using vanishing point concept", *Pattern Recognition*, Vol. 24, pp. 57-67, 1991.

[11] 최성구, 노도환, "정방형 교정 frame을 이용한 카메라의 교정 파라메타 추정에 관한 논문", 대

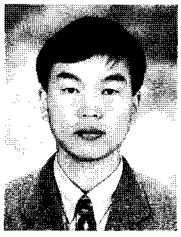
한전자공학회논문지, vol. 33-B, no. 7, pp. 127-137, 1996

저 자 소 개



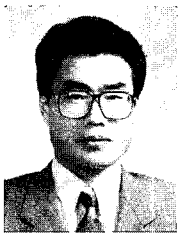
鄭 浚 益(學生會員)

1972년 11월 4일생. 1995년 8월 전북대학교 전기공학과(공학사). 1996년 3월 ~ 현재 전북대학교 전기공학과 석사과정 재학중. 주요관심분야는 로봇비젼, 센서응용, 영상처리, 영상계측 등



崔 成 求(正會員)

1988년 2월 전북대학교 전기공학과 졸업(공학사). 1994년 2월 전북대학교 전기공학과 대학원(공학석사), 1996년 2월 전북대학교 전기공학과 박사과정 수료. 1996년 8월 ~ 현재 정읍공업전문대학 전임강사. 주요관심분야는 화상계측, 광응용계측 등



盧 度 煥(正會員)

1973년 2월 전북대학교 전기공학과 졸업(공학사). 1983년 3월 동경공업대학 제어공학과(공학석사), 1987년 3월 동경공업대학 제어공학과(공학박사). 1988년 4월 ~ 현재 전북대학교 전기공학과 교수. 현재 전기전자회로합성연구소 생체 및 감성공학연구실 연구원. 주요관심분야는 Vehicle 계측 및 제어, 로봇비젼, 센서응용, 영상처리, 광응용계측 등