

V.F. 모델을 이용한 주행차량의 전방 영상계측시스템 설계

Design of a Front Image Measurement System for the Traveling Vehicle Using V.F. Model

정용배*, 김태효*

Yong-Bae Jung*, Tae-Hyo Kim*

요약

본 논문에서는 주행하는 차선 내에 존재하는 선형차량의 위치측정 및 차량까지의 거리를 정확히 인식하기 위해 차량의 요동에 기인되는 피칭오차를 보정하여 계측하는 알고리즘을 제안하였다. 선형 차량까지의 거리를 정확히 계측하기 위하여 2차원 영상좌표로부터 사영기하학을 적용하여 3차원 좌표를 얻을 수 있는 카메라 Calibration 알고리즘을 확립하고, View Frustum(V.F.) 모델을 이용하여 직선부분에 대한 도로의 영상 모델을 제시하였다.

기 발표된 많은 연구논문들은 도로가 평면이라 가정하여 도로와 차량사이의 기하적인 변화에 따른 오차 특성을 고려하지 않은 경우가 많다. 이를 보완하고자 본 논문에서는 카메라 Calibration 알고리즘을 적용하여 실세계 좌표계와 영상좌표계 사이의 기하해석으로 사영행렬을 추출하였고, V.F. 모델을 이용하여 소실점의 기하적인 해석을 통하여 차량의 피칭변화에 따른 오차특성을 보정하였다. 실험결과 거리의 오차를 현저하게 줄일 수 있어 피칭변화에 강인함을 확인할 수 있었다.

Abstract

In this paper, a recognition algorithm of the straight line components of lane markings and an obstacle in the travelling lane region is proposed. This algorithm also involve the pitching error correction algorithm due to traveling vehicle's fluctuation. In order to reduce their error a practical road image modelling algorithm using V.F. model and camera calibration procedure are suggested to adapt the geometric variations. It is obtained the 3D world coordinate data by the 2D road images.

In experimental test, we showed that this algorithm is available to recognize lane markings and an obstacle in the traveling lane

Keywords : camera calibration, view frustum model, pitching error correction, vehicle lanes recognition, obstacle recognition.

I. 서론

부가가치가 높은 자동차 산업에 여러 가지 운전보조 수단들을 개발하여 설치하거나 탑재하여 사고의 예방을 위한 많은 연구가 이루어지고 있다. 이것은 운전자의 안전을 확보하기 위함이고 더욱 편안한 운전을 실현하고자 하는 요

구이다. 이러한 안전과 사고 방지를 위한 목적으로 여러 분야에서 연구가 진행 중에 있고, 특히 영상처리를 이용한 비전시스템 개발로 자율주행을 실현하는 연구가 성행하고 있다[1][2].

비전시스템을 이용하는 경우 차량에 카메라를 설치하고 이를 처리하기 위한 프로세서와 기타 주변장치를 탑재하여 도로의 상황 정보를 계산하고 판단한 결과를 시각 및 청각적으로 운전자에게 제공함으로써 사고를 미연에 방지할 수 있다. 그러나 이를 실현하고 상용화하기 위해선 많은 제약과 문제점이 있다[3-5]. 아직 기존의 프로세서나 영상처리 기술이 실시간 처리에 적합하지 않고, 고가이기 때문에 상용화하거나 저가의 시스템을 갖추기가 어려운 실정이다.

*경남대학교 정보통신공학부

논문 번호 : 2006-2-22

접수 일자 : 2006. 4. 11

심사 완료 : 2006. 7. 7

※본 논문은 2005년도 경남대학교 학술연구비 지원에 의하여 이루어 졌음.

일부 고급 자동차에만 존재하는 시스템은 그 가치가 줄어들게 된다. 따라서 저가의 시스템 구현이 가능한 알고리즘의 개발이 필요하다.

기 발표된 알고리즘들은 전처리 과정에서 에지 검출이나 에지 개선을 위해서 많은 영역을 탐색하고 시간을 소비하는 것을 볼 수 있고 특정 점을 찾기 위해 탐색 및 처리를 하는데 많은 시간을 소비하고 있다[4]. 또한 영역을 축소해서 처리 속도를 높이기 일환으로 sub-block이나 sub-window를 사용하는데 고정 또는 가변일지라도 차선을 고르게 탐색할 수 없을 뿐 아니라 주요한 차선의 일부를 놓칠 가능성도 크다. 차선의 모양과 특성을 검출하는데 있어 미흡하고 장애물이나 도로의 상태에 따른 인식률을 보장할 수 없는 단점이 있다[3].

따라서 본 논문에서는 이러한 문제점들을 보완하기 위하여 카메라 Calibration 알고리즘을 적용하여 좌표계 변환과 V.F. 모델을 이용하여 차선 및 장애물의 검출 및 인식을 위한 접근 방법을 제안하고자 한다. 차선의 특징점을 검출하기 위해서 특정 마스크를 제한하고, 차선의 경로를 그 모양과 특성에 맞게 추적하기 위해 새로운 차선경로 추적법은 차선의 좌측레인을 검출하기 위한 마스크와 우측레인을 검출하기 위한 마스크를 두어 각각 구분하여 검출하게 된다. 새로운 차선 경로 추적법은 이 마스크들을 가지고 차선의 특성과 연속성을 추적해 차선을 원형에 가깝게 검출해 내어 장애물이나 기타 다른 여건에서도 강인함을 나타내어 인식률을 높일 수 있고, 차선의 탐색영역도 직선형 도로의 영상을 대상으로 본 알고리즘에 적용하여, 정확한 인식이 이루어짐을 확인하고자 한다.

II. 카메라 Calibration 알고리즘

2.1 핀홀 카메라의 원리

그림 1은 핀홀 카메라의 구조를 나타낸 것이다. 그림에서 M은 실세계좌표계의 한 점이고, m은 카메라의 초점 C를 통해 영상평면에 맺히는 점이다. 이와 같은 핀홀 카메라의 기하구조로부터 영상좌표계는 식 (1)과 같이 표현할 수 있다.

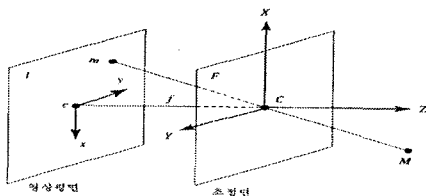


그림 1. 핀홀 카메라의 구조

Fig. 1. Structure of pin-hole camera

식 (1)은 이상적인 핀홀 카메라 모델에서 님은 삼각형의 성질을 이용하여 유도해 낼 수 있다.[5]

$$s \cdot \tilde{m}_s = P \cdot \tilde{M}_c \tag{1}$$

$$s \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

s : 영상의스케일(크기)

\tilde{m}_s : 영상 평면 좌표에 대한 확장벡터

\tilde{M}_c 카메라 좌표에 대한 확장벡터

$$P = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \text{ 사영 행렬}$$

2.2 디지털 영상평면좌표와 영상평면좌표

그림 2는 카메라 내부변수를 나타낸 것이며 카메라 내부변수를 고려할 때의 영상좌표는 식 (2)와 같이 표현할 수 있다[7]. 그림에서 내부의 직각좌표는 이상적인 좌표를 나타내지만, 실제 카메라의 CCD셀들의 배열은 그림의 외부와 같이 직각이 되지 않고 약간의 왜곡된 형태로 표현된다.

$$\tilde{m} = H \cdot \tilde{m}_s$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & -k_u \cot \theta & u_0 \\ 0 & k_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2}$$

\tilde{m} : 디지털 영상 평면 좌표

k_u = 촬상소자 [CCD 단위셀]의 수평축 길이

k_v = 촬상 소자 [CCD 단위셀]의 수직축 길이

u_0, v_0 = 디지털 영상 평면의 중심 좌표

θ : CCD 단위셀의 수직축과 수평축이 이루는 각도

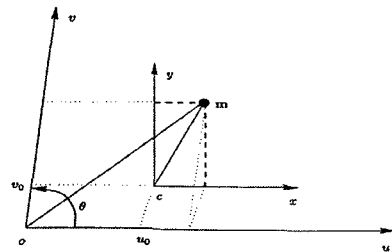


그림 2. 카메라 내부변수

Fig. 2. Internal parameters of camera

2.3 디지털 영상평면 좌표와 실세계 좌표

디지털 영상평면 좌표계와 실세계 좌표계 사이의 관계식은 카메라 내부변수와 외부변수를 고려하여 다음 식(3)과 같이 행렬의 결합 형태로 나타낼 수 있다.

$$s \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u - k_u \cot \theta & u_0 \\ 0 & k_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$s \cdot \tilde{m} = H \cdot P \cdot D \cdot \tilde{M}_w \quad (3)$$

단일 카메라를 사용하여 도로평면 위의 임의의 점에 대한 실세계 좌표를 구하는데 식(3)이 적용되어지며, 이러한 경우 도로 영상은 평면인 것으로 가정하고 카메라와 도로 간의 기하적 변화에 대한 계산 오차들은 무시되는 것으로 간주한다. 여기서 3차원 공간상의 모든 점에 대한 위치좌표의 계산은 식(3)의 카메라 파라미터가 구해지면 2차원 평면의 위치좌표로 실시간으로 변환가능하다.

그림 3은 도로에서 카메라의 Calibration을 위한 기하학적 모델을 나타낸 것이다. 여기서 가로로 평행한 2개의 선이 차선을 의미하고, 차량의 상단에 카메라에서 내려다 보는 도로와의 기하적 구조를 나타내며 차선간의 폭은 규격화되어 알고 있는 값이고 전방의 거리는 미리 계측하여 알고 있는 값이다. 카메라의 높이를 알고 거리를 알면 도로의 한 점에서 카메라를 향하는 각도를 계산할 수 있다.

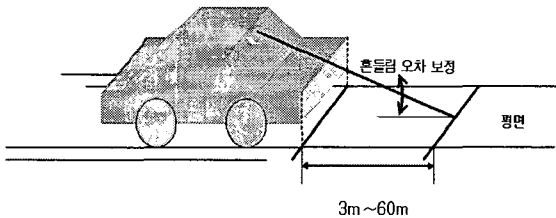


그림 3. 카메라와 차선간의 기하적 모델
Fig.3. Geometric model between camera and road lanes

III. View Frustum(V.F.) 모델을 이용한 도로구조 모델링

1. V.F. 모델의 기하해석[8]

실세계좌표계와 카메라좌표계 사이의 기하구조 해석을 위하여 V.F. 모형의 기하구조를 가지는 모델로 나타낼 수 있다. V.F.는 하나의 3차원 장면에서 보이는 모든 것들을 표현하는 공간적 입체구조이다[7][8]. V.F.는 피라밋 형태이며 피라밋의 꼭대기가 카메라의 위치에 해당한다. 이러한 형태는 카메라로 직사각형 창문을 내다볼 때 생기는 입체를 그대로 반영하는 것이다. V.F.는 6개의 평면들로 정의되며 그 중 네 개는 화면의 네 가장자리 변들을 결정한다. 즉 이 네 평면들은 V.F.의 상,하,좌,우 면들을 결정하고 나머지 두 평면들은 절단면의 가까운 평면과 먼 평면을 결정한다. 그림 4에서 사각뿔 중간의 평면은 CCD

셀이 영상평면을 나타내며, 밑면은 도로평면을 그리고 꼭지점의 평면은 카메라좌표를 나타낸다. 카메라 좌표계(M_C)와 실세계 좌표계(M_W) 사이의 대응 관계는 그림 4와 같다. H_f 는 카메라 중심에서 V.F. 바닥 면까지의 수직 거리이며, $\Delta Z_W, \Delta X_W, \Delta Y_W$ 는 실세계 좌표계와 카메라 좌표계의 좌표변환을 위한 이동 성분,

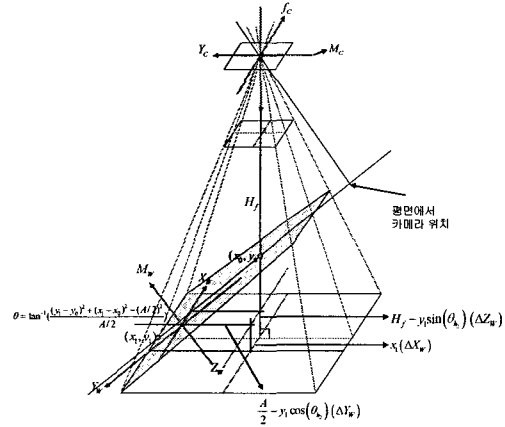


그림 4. V.F. 모델
Fig. 4. V.F. model

θ 는 회전성분을 나타내고, 다음 식 (4)와 같이 계산될 수 있다.

$$\begin{aligned} \Delta X_W &= x_1 \quad \Delta Y_W = \frac{A}{2} - y_1 \cos(\theta_{h_2}) \\ \Delta Z_W &= H_f - y_1 \sin(\theta_{h_2}) \\ \theta &= \tan^{-1} \left(\frac{(y_1 - y_0)^2 + (x_1 - x_0)^2 - (\frac{A}{2})^2}{A/2} \right) \end{aligned} \quad (4)$$

그리고 회전과 병진에 대한 행렬은 식 (5)와 같으며, T_x, T_y, T_z 는 식 (5)와 같이 X축, Y축, Z축 방향으로의 이동성분을 나타내고, θ 는 실세계 좌표계와 카메라 좌표계의 X축 회전각을 나타낸다.

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & T_x \\ 0 & \cos \theta & -\sin \theta & T_y \\ 0 & \sin \theta & \cos \theta & T_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix} \quad \begin{aligned} \theta &= \Delta X_\theta \\ T_x &= -\Delta X_W \\ T_y &= -\Delta Y_W \\ T_z &= -\Delta Z_W \end{aligned} \quad (5)$$

실험 모델에서의 실세계 좌표계 임의의 한 점을 카메라 좌표계에서의 한 점으로 변환시킬 수 있다. 또한 V.F. 모델을 기하적으로 해석하면 카메라 CCD셀의 화각을 정확하게 구할 수 있으며, 그림 5에 V.F. 모델의 절단면의 구조를 나타내었다. 그림 5에서 카메라의 CCD면과 도로 평면이 이루는 각도 θ_3 은 다음 식 (6)과 같다.

$$\theta_3 = \tan^{-1}(h_2/w_2) \quad (6)$$

그리고 $a/2, k_1, k_2$ 가 단위 픽셀 개수를 나타낼 때, CCD 셀의 화각 θ_{CCD} 는 식 (7)에 의해 구해진다.

$$\theta_{CCD1} = \tan^{-1}\left(\frac{K_1}{H_f}\right), \quad \theta_{CCD2} = \tan^{-1}\left(\frac{K_2}{H_f}\right)$$

$$\theta_{CCD} = \theta_{CCD1} + \theta_{CCD2} \quad (7)$$

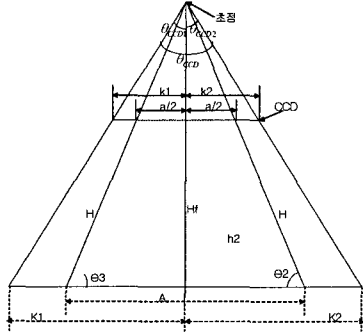


그림 5. V.F의 절단면 기하구조
Fig. 5. Geometric structure of cross section for the V.F model

도로평면의 세 점을 알면 식 (8)을 이용하여 도로평면의 방정식을 구할 수 있다.

$$N_a = y_1(z_2 - z_3) + y_2(z_3 - z_1) + y_3(z_1 - z_2)$$

$$N_b = z_1(x_2 - x_3) + z_2(x_3 - x_1) + z_3(x_1 - x_2)$$

$$N_c = x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)$$

$$N = \sqrt{N_a^2 + N_b^2 + N_c^2}$$

여기서, $a = \frac{N_a}{N}, \quad b = \frac{N_b}{N}, \quad c = \frac{N_c}{N}$

$$d = -(ax_1 + by_1 + cz_1) \quad (8)$$

평면의 방정식에서 a, b, c는 평면의 법선을 나타내므로, 카메라의 중심을 지나고 평면의 법선 벡터와 평행한 직선의 방정식은 식 (9)와 같다.

$$\frac{x-0}{x_m} = \frac{y-0}{y_m} = \frac{z-0}{z_m} = t \quad (x_m, y_m, z_m \text{ 은 평면의 법선벡터})$$

$$x = x_m t, \quad y = y_m t, \quad z = z_m t \quad (9)$$

따라서 도로평면에서의 카메라의 위치 점은 평면과 직선이 만나는 점이 되므로 식 (8)과 식 (9)를 연립하여 풀면 식 (10)과 같이 카메라의 위치를 구할 수 있다.

$$(ax_m + by_m + cz_m)t + d = 0$$

여기서, $t = \frac{-d}{ax_m + by_m + cz_m}$

$$x = x_m \times \left(\frac{-d}{ax_m + by_m + cz_m}\right)$$

$$y = y_m \times \left(\frac{-d}{ax_m + by_m + cz_m}\right)$$

$$z = z_m \times \left(\frac{-d}{ax_m + by_m + cz_m}\right) \quad (10)$$

한편, 카메라의 설치 높이는 카메라의 중심에서 도로평면까지의 수직거리를 구하면 되므로 식 (11)을 이용하여 계산할 수 있다.

$$l = \frac{|ax_1 + by_1 + cz_1 + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (11)$$

2. 피칭오차 보정 알고리즘[14]

계산된 피칭 각도 변화에 따른 계측오차를 보정하기 위하여 그림 6과 같이 오차보정 점을 구한다. 회전축은 카메라 좌표계에 대한 변화량으로 앞의 II절에서 해석한 바와 같다. 결국 카메라 좌표계의 원점을 중심으로 변화량 만큼 회전한 위치를 계산하고 현재 장애물의 위치를 카메라 좌표계의 좌표를 교정 시 카메라 좌표계와 실세계 좌표계 변환에 의해 계산하고 그에 대한 다음과 같은 상수를 교정 시점에 계산한다.

- (i) dX_W : 실세계 좌표계와 카메라 좌표계의 X축 방향 이동량
- (ii) dY_W : 실세계 좌표계와 카메라 좌표계의 Y축 방향 이동량
- (iii) dZ_W : 실세계 좌표계와 카메라 좌표계의 Z축 방향 이동량
- (iv) dTh : 실세계 좌표계와 카메라 좌표계의 X축 회전량

그러나 주행 중 카메라는 피칭방향의 변화성분이 발생하며, 계산된 좌표는 그림 6에서의 1번 점과 같이 회전변화가 보정되지 않은 상태에서의 좌표이다. 카메라가 회전하고 카메라 중심에 대한 CCD셀의 도로평면은 회전되지 않은 상태로 교정 데이터에 대해 회전 변화가 있는 것이다. 주행 중 차량에 탑재된 속도, 무게의 변화나 충격으로 위에서 언급한 바와 같은 여러 기하적인 변화가 생기게 되며 이러한 변화는 실제 계산에서의 오차를 발생하게 된다. 그중 주행 차량의 피칭 변화량의 계산은 중요한 오차 요인으로 미리 교정되어진 상수 값과 해석된 내용을 이용하여 오차를 보정하게 된다.

먼저 소실점을 이용하여 계산된 피칭각도 만큼 그림 6의 P_{m1} 점의 좌표를 식(12)를 이용하여 회전시켜 이동되어진 좌표 P_m 을 구한다

$$\begin{pmatrix} x_{m_2} \\ y_{m_2} \\ z_{m_2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_{m_1} \\ y_{m_1} \\ z_{m_1} \end{pmatrix} \quad (12)$$

이 P_{m_2} 점과 카메라의 중심을 지나는 직선과 평면이 만나는 그림 6의 P_{m_3} 점이 실제 좌표가 되며 평면에서의 카메라 위치를 알면 피칭회전에 대한 보정된 좌표의 값을 그림 6에 서와 같이 계산할 수 있다[9][10].

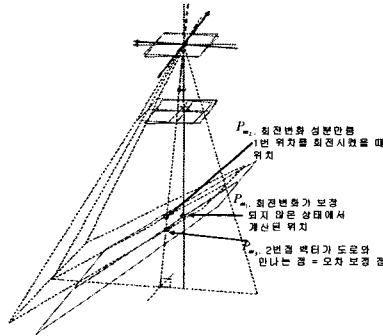


그림 6. 피칭오차 보정 좌표점의 예
Fig. 6. An example of pitching error correction

IV. 차선 및 장애물 인식

1. 차선인식

카메라에 입력되는 도로의 영상은 화면 높이의 5분의 4위치에 소실점이 오도록 화각을 그림 7과 같이 고정한다. 그림 7은 도로영상과 좌표계 설정을 나타내는 것이다.

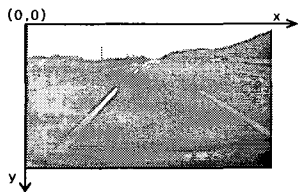


그림 7. 도로영상과 영상좌표계 설정
Fig. 7. Setting up between road image and image coordination

차선을 검출하기 위해 우선 Robert 연산자를 이용하여 영상의 에지를 검출하고, 화면의 상하방향으로 10개의 샘플링 라인을 결정하여 화면의 좌에서 우측으로 에지점을 탐색하여 차선의 후보 점들을 찾는다. 이때 도로영역과 좌우 차선부분의 그레이레벨 차이를 구별하기 위해 임계값 처리를 한다. 탐색과정에서 밝기의 최대값, $P_{l_{max}}$ 와 $P_{r_{max}}$ 를 찾고, 도로영역의 평균 밝기값 $P_{l_{av}}$, $P_{r_{av}}$ 을 구한다. 좌우 차선의 임계값은 각각 식(13), 식(14)와 같이 계산되어지며 도로영역과 차선과의 밝기 관계를 그림 8에 나타내었다. 그림 8의 (b)에서 도로영역과 차선부분의 그레이 값의 비율을

여러 번 측정한 결과, P_{max} 의 75% 정도가 최적으로 구해진 결과이다.

$$THl = 0.25 P_{l_{max}} + P_{l_{av}} \quad (13)$$

$$THr = 0.25 P_{r_{max}} + P_{r_{av}} \quad (14)$$

그 후에 차선이 직선임을 고려하여 Hough 변환을 적용하여 직선성분을 검출한다. 차선 외곽 픽셀에 Hough 변환을 적용하면 차선을 나타내는 직선을 구할 수 있다. 이러한 Hough 변환은 식(15)와 같으며 그림9에 나타내었다[11][12].

$$\rho = x \cdot \cos\theta + y \cdot \sin\theta \quad (15)$$

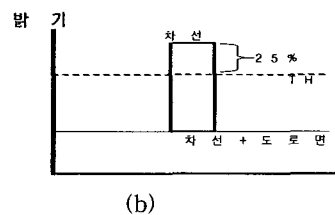
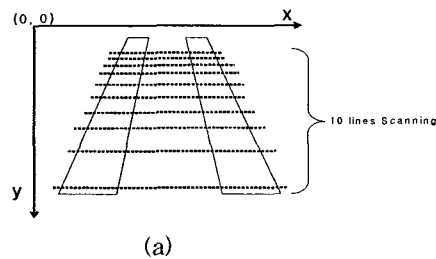
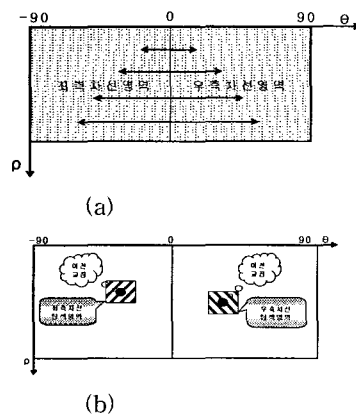


그림 8. 스캐닝라인(a) 및 임계값 설정(b)
Fig. 8. Scanning lines(a) and determination of threshold value(b)

차선을 나타내는 $\rho - \theta$ 평면에서의 유일한 한 점 대신에 가장 조밀한 교차점들의 분포를 구하며, 이때 많은 처리시간과 노이즈 문제를 해결하기 위하여 교점을 구하는 영역을 그림 7과 같이 최소 범위로 제한하였다.



(b)

그림 9. 초기 차선 탐색 영역 맵(a)와 주행 중 탐색 영역 (b)

Fig. 9 Initial scan region map (a) and on traveling scan region(b)

그림 9에서 초기의 탐색은 (a)와 같이 전 영역에서 구하지만 주행 중에는 영상처리가 20frames/sec 이상이 되므로 그림 (b)와 같이 이전상태에서 변화의 범위가 크지 않으므로 이를 고려하여 탐색영역을 제한하여 탐색 및 계산시간을 단축시킬 수 있다.

2. 장애물 인식

차선인식 후에 장애물을 인식하기 위하여 인식된 차선영역 내부에 한정하여 장애물의 유무를 그림 10과 같이 검출한다. 전방차량의 유무를 판단하는 것은 주행 차선 내에서의 수평에지를 검출함으로써 가능하다[13]. 그림 10과 같이 차량탐색영역의 범위 내에서 수평방향의 에지들을 투영하여 누적하였을 때 임계값 보다 큰 지점을 장애물이 있는 위치로 정한다. 이를 식(16)과 같이 계산한다. 여기서 상수 400은 누적 값을 강조하기 위한 값이다.

각 라인별 수평에지 투영치 =

$$\frac{\text{에지 이치화된 픽셀수}}{\text{전체픽셀수}} \times 400 \tag{16}$$

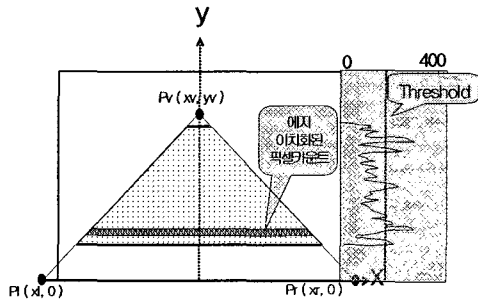


그림 10. 장애물 검출영역 및 수평에지 투영레벨
Fig. 10. Detection region of obstacle and projection level of horizontal edge

그림 11은 차선인식 및 장애물인식에 대한 전체적인 영상 처리의 순서도를 나타낸다.

V. 실험 및 고찰

본 논문에서 제안한 알고리즘을 확인하기 위하여 도로 상에 도로 폭과 동일한 기지의 패턴을 설치하고, 그림 12와 같이 카메라 Calibration 처리를 통하여 표 1과 같이

사영행렬을 산출하였다. 일단 사영행렬이 주어지면 이 교정행렬에 의해 3차원 좌표변환이 이루어 질 수 있다. 본 실험에 사용한 컴퓨터는 2GHz의 CPU를 사용한 노트북로서 차량에 탑재하여 정지된 위치에서 Calibration실험을 하였고, 실제 도로에서의 차선인식 및 장애물 인식 실험은 표 1의 카메라 교정행렬에 의해 계산된다.

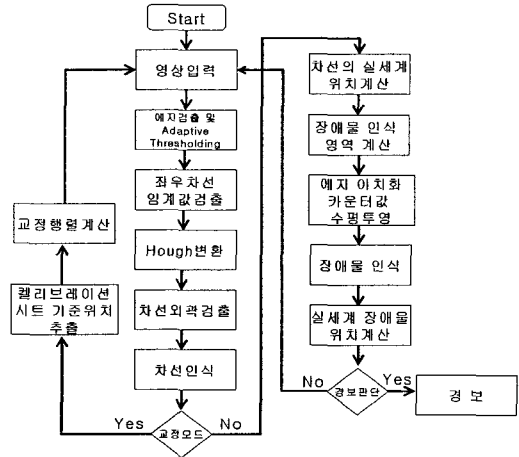


그림 11. 차선 및 장애물 인식처리 순서도
Fig. 11. Recognition processing flow chart for the lanes and obstacles

표 1은 실제거리 10.1m 거리에 차선폭과 같은 기지의 패턴을 놓고서 Calibration 처리에 의한 카메라의 내부 변수를 식 (2)를 이용하여 추출한 것이다.

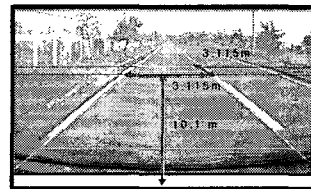


그림 12. 카메라 calibration 과정
Fig. 12. Procedure of camera calibration

표 1. 추출된 사영행렬 (10.1m)
Table 1. Obtained projection matrix (10.1m)

추출된 사영행렬(10.10m)	Camera 교정 행렬		
	-28.2028	-25.0617	11457.9465
	0.0	175.1731	-32582.205
	0.0	-49.3896	4246.0835

표 2는 9.9m의 거리에서 차량의 피칭각도 변화에 대한 거리계측 결과로서 오차보정 처리 이전과 보정 후의 오차 특성을 그림 6 및 식 (12)를 적용하여 측정된 결과이다. 보다 자세한 오차보정에 대한 계측 결과는 문헌 14에 언급되어 있다. 표 2에서 알 수 있듯이 피칭각도 변화가 없을 때는 보정전의 경우 1m의 오차를 보였고, 보정 후는 0.08m의 오차를 나타내었다. 그러나 0.4도의 각도변화에서 계측거리가 멀어질수록 계측오차가 크게 증가함을 알 수

있었고, 피칭각도의 변화가 클수록 오차도 증가함을 알 수 있었다.

그림 13은, 가로축에 표시한 거리로 각각 4.4m, 7.72m 및 9.9m에서 계측한 보정 전과 후의 계측결과를 세로축에 오차를 나타낸 것이다. 여기서, 피칭오차를 보정하기 전과 보정 후는 오차가 현격한 차이를 알 수 있다.

표 2. 교정행렬(10.1m)에 대한 9.9m의 거리계측 결과
Table 2. distance measurement result of 9.9m for calibration matrix(10.1m)

각도변화	실제거리	보정 전 거리	보정 후 거리
1.04	9.90	11.30	9.91
0.41	9.90	10.60	10.04
0	9.90	10.09	9.98
-0.24	9.90	9.92	9.81
-1.08	9.90	8.93	9.85

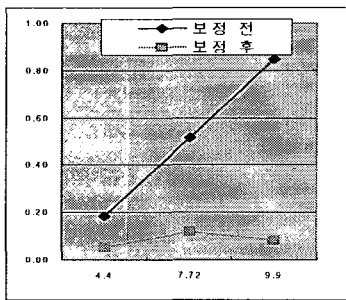


그림 13. 10.1m의 교정시트 계측치에 대한 오차
Fig. 13. Mean error for the measurement at 10.m of calibration sheets

본 실험에서 거리계측의 오차를 2% 이하로 낮출 수 있음을 알 수 있다.

차성인식 및 장애물인식 시험에 사용한 CCD카메라는 640×480화소의 범용이며, 처리시간 단축을 위해 화면의 사이즈를 320×240화소로 다운샘플링된 화면으로 처리하였다. 그림 14는, 그림 11의 처리순서에 따라 도로영상을 (a)에지 검출한 후에 (b)HT변환을 적용한 결과 및 (c)는 10개의 스캐닝 라인을 통해 차성의 후보로 검출된 결과이다. 그리고 그림 15는 각각 (a) 원영상, (b) 에지검출 결과, (c) 수평에지 투영 및 (d) 장애물을 검출한 결과를 나타낸다. 여기서 주행차선 내의 수평투영으로 얻어지는 히스토그램은 장애물이 있는 위치에서 그 누적 값이 크므로 이를 더욱 강조하기 위해 식(16)과 같이 처리하였다. 그림 15의 예에서 검출된 장애물까지의 거리는 39.5m로 계측되었다. 이때 실제 주행하는 차량까지의 거리를 측정할 수는 없었으나, 다른 정지된 물체를 대상으로 측정한 결과에 의하면 그 오차는 100m에서 ± 1m 정도로 계측되었다.

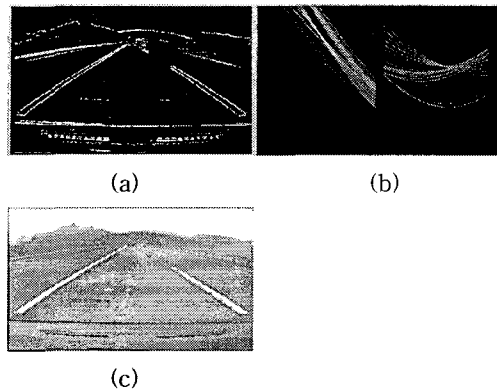


그림 14. 차선인식 과정, (a)에지검출, (b)Hough 변환맵, (c)차선인식결과
Fig. 14. Procedure of lanes recognition, (a)edge detection, (b)Hough transform, (c)result of lanes recognition

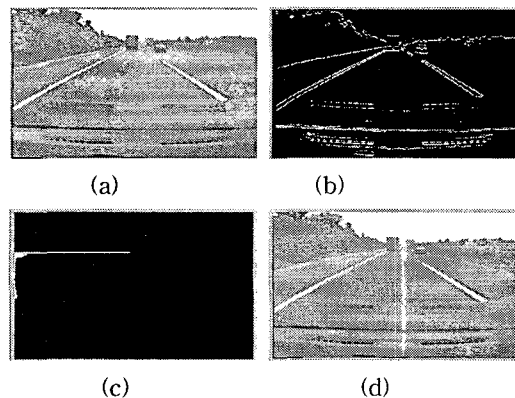


그림 15. 장애물 검출 결과, (a)원영상, (b)에지 검출결과, (c)수평투영 히스토그램, (d)장애물검출 결과
Fig. 15. Result of obstacle recognition, (a)original image, (b)result of edge detection, (c)horizontal histogram, (d)result of obstacle projection

VI. 결론

본 논문에서는 단일 CCD카메라를 이용하여 도로와 차선의 기하구조를 이용한 3차원 좌표계의 Calibration 알고리즘을 확립하였고, 주행차량의 피칭에 의한 거리계측 오차를 줄이기 위해 V.F. 모델을 이용하여 오차보정 알고리즘을 제안하고 차선 및 장애물 인식에 대한 실험을 하였다. 직선차선의 효율적 검출을 위해 Hough 변환을 통해 ρ 와 θ 의 검출범위를 상황에 따라 변경할 수 있는 알고리즘을 고안하였다. 본 알고리즘은 기존의 sub-window를 이용한 처리알고리즘보다 특징점 탐색영역의 감소로 처리속도가 향상되어 2GHz의 CPU를 가지는 컴퓨터로 20frames/sec 이상, 피칭 오차를 2%이하로 개선하여 보다 정확한 인식을 할 수 있었다. 장애물까지의 거리계측 실험에서 0.4도의 피칭오차가 있

는 경우 실제거리 9.9m의 거리를 오차보정 없이 계측한 결과 3회 이상 반복 측정한 결과 0.39~0.61m의 계측오차를 보였으나, 피칭오차를 보정한 후에는 0.1m 이내로 감소한 결과를 얻을 수 있었다. 이는 본 알고리즘이 실제 응용의 가능성을 보인 결과로 판단된다.

참고문헌

[1] Zehang Sun, Ronald Miller, George Bebis and David DiMeo, "A Real time Precrash Vehicle Detection System", 2002. IEEE.

[2] Narayan Srinivasa, "Vision-based Vehicle Detection and Tracking Method for Forward Collision Warning in Automobiles", Intelligent Vehicle Symposium, 2002. IEEE.

[3] Luigi Di Stefano and Enrico Viarani, "Vehicle Detection and Tracking Using the Block Matching Algorithm", 1999. IEEE.

[4] Rongben Wang, Youchun Xu, Libin and Yu fan Zhao, "A Vision-Based Road Edge Detection Algorithm", Intelligent Vehicle Symposium, 2002. IEEE

[5] Gang Xu and Zhengyou Zhang, "Epipolar Geometry in Stereo Motion and Object Recognition", Kluwer Academic Publishers, 2002

[6] Luciano da Fontoura Costa and Roberto Mardes Cesar Jr, "Shape Analysis and Classification", CRC Press, pp.365-400, 2001.

[7] Richard Hartley, Andrew Zisserman, "Multiple View Geometry in Computer Vision", Cambridge University Press, 2000.

[8] David H. Eberly, "3D Game Engine Design, A Practical Approach to Real-Time Computer Graphics", Morgan Kaufmann, pp.79-139, 2002.

[9] Kaske A, "Contribution à la Detection de Bords de Route imprécis", Implantation surle Robot ROMANE, Institut National Polytechnique de Lorraine, April 1997.

[10] Pierre Coulombeau, Claude Laugeau, " Vehicle yaw, pitch, roll and 3D lane shape recovery by vision", IEEE Transactions on Intelligent vehicle, 2002.

[11] Kaske A., Wolf D., and Husson R., "Lane boundary detection using statistical criteria", In International Conference. on Quality Control by Artificial Vision, May 1997.

[12] DeMenthon D., "Reconstruction of a road by matching edge points in the road image", In Technical Report CAR-TR-368, Computer Vision Laboratory for Automation Research, June 1988.

[13] Kanatani K., Watanabe K., "Reconstruction of 3d

road geometry from images for autonomous land vehicles", IEEE Trans. On Robotics and Automation, Feb. 1990.

[14] 김태효, 서경호, "V.F. 모델링을 이용한 주행차량의 진동에 대한 도로영상의 계측오차 보정알고리즘", 제어자동화시스템공학논문지, 12권 8호



정용배(Yong-Bae Jung)

2003. 2. 경남대학교 정보통신 공학과 졸업(공학사)

2004. 8. 경남대학교 대학원 정보통신공학과 졸업(공학석사)

2004. 9. - 현재 동 대학원 박사과정 재학

관심분야: 영상처리, 컴퓨터비전, 신호처리시스템 설계 등.



김태효(Tae-Hyo Kim)

1977년 영남대학교 전자공학과 (공학사)

1980년 영남대학교 대학원 전자공학과 (공학석사)

1988년 영남대학교 대학원 전자공학과 (공학박사)

1990년 12월-1991년 1월 미국펜실바니아대학 Post Doc

1983. 3 - 현재 경남대학교 정보통신공학부 교수

관심분야: 컴퓨터비전, 영상계측, 영상처리