韓國港灣學會 第14卷 第3號, pp. 303~312. 2000.

論 文

비전 시스템을 이용한 AGV의 차선인식 및 장애물 위치 검출에 관한 연구+

이 진 우* · 이 영 진** · 이 권 순***

A Study on Detection of Lane and Situation of Obstacle for AGV using Vision System

J. W. Lee · Y. J. Lee · K. S. Lee

Key Words : 비전시스템(Vision system), AGV(Autonomous guided vehicle), 자율이동로봇(Autonomous mobile robot), 허프변환(Hough transform), 장애물 검출(Obstacle detection), Viewport

Abstract

In this paper, we describe an image processing algorithm which is able to recognize the road lane. This algorithm performs to recognize the interrelation between AGV and the other vehicle. We experimented on AGV driving test with color CCD camera which is setup on the top of vehicle and acquires the digital signal. This paper is composed of two parts. One is image preprocessing part to measure the condition of the lane and vehicle. This finds the information of lines using RGB ratio cutting algorithm, the edge detection and Hough transform. The other obtains the situation of other vehicles using the image processing and viewport. At first, 2 dimension image information derived from vision sensor is interpreted to the 3 dimension information by the angle and position of the CCD camera. Through these processes, if vehicle knows the driving conditions which are lane angle, distance error and real position of other vehicles, we should calculate the reference steering angle.

+ 본 연구는 1997년도 한국학술진흥재단 대학부설연구소과제 연구비에 의하여 연구되었음.
* 동아대학교 대학원 전기공학과 박사과정
** 항공기능대학 전기계측제어과 전임강사
*** 정회원, 동아대학교 공과대학 전기전자컴퓨터공학부 교수

1. 서 론

최근의 과학 기술의 발달과 더불어 항만의 제반 관리시스템과 하역장비의 무인 자동화에 대한 연구가 활발히 진행중이며, 그 중 무인반송차량(autonomous guided vehicle : AGV) 및 자율이동로봇(autonomous mobile robot : AMR) 등은 이미 현재 일반 산업현장에서 많이 적용되고 있다. 그러나, 항만이나 일반 도로 같은 작업환경의 변화가 심한 곳에서 적용하기에는 아직 안정성이나 신뢰성에서 많은 미비함이 존재한다[1].

본 논문에서는 이러한 문제들 중에서도 무인 차량의 차선과 장애물 인식에 대한 내용을 다루고자 한다. 무인 운행시 장애물의 출현을 인식하고, 이로 인한 충돌을 회피하는 문제는 산업현장의 안정적 측면에서 대단히 중요하다[2,3,4]. 이를 해결하기 위한 방안으로 근접 센서나 초음파센서, 영상(image)정보를 이용한 비전시스템(vision system) 등이 많이 응용되고 있다[1,5,6].

다른 센서에 비해 보다 많은 정보를 동시에 얻을 수 있는 비전시스템은 실시간 처리에 대한 문제점이 있지만, 최근 고속 컴퓨터의 개발로 인해 점차 해결되고 있다[7,8,9]. 본 논문에서는 한 대의 CCD카메라를 이용하여 차선을 인식하고, 주행차량과 장애물 위치 및 차량과의 상대적인 변위를 동시에 인식할 수 있는 알고리즘을 제안하며, 자율이동차량에 적합한 시각 시스템으로 사용될 수 있을지에 대한 가능성을 제시해 보고자 한다.

처리과정은 차선인식과 장애물 위치 검출로 나눌 수 있다. 차선인식부에서는 입력영상의 색상비율을 이용한 필터링작업과 경계선 검출 등의 영상처리 기법을 이용하여 차선의 기울기와 차선 중앙부와의 거리오차를 검출하고, 주행하고 있는 차량의 상태를 파악하도록 하였다. 장애물 위치 검출부에서는 2차원적인 영상정보에서 3차원 좌표로 변환하여 검출된 전방 차량의 상대적인 위치와 변위를 검출하도록 하였다.

2. 본 론

일반적으로 움직이는 물체를 분석하기 위해서는 정지된 배경에서 움직인 물체만을 차분영상(difference image)에 의해 해석하는 방식을 사용하고 있다[10,11,12]. 그러나, 차량이 운행할 경우 입력영상은 모든 부분이 움직이므로 차분영상을 이용하여 물체를 추출하기 어려워진다. 즉, 움직이는 차량에서 획득된 영상정보는 모든 배경이 움직이게 된다. 그러므로 본 논문에서는 움직이는 배경을 가지는 영상에서 배경과 물체를 분리하는 방법과 추출된 물체의 위치 및 변위를 알아낼 수 있는 방법에 대해 논하고자 한다. 그림 1은 입력영상에 대한 차선인식과 장애물 위치 검출과정에 대한 부분을 나타낸다. 차선 인식부에서는 색상비율을 이용한 이진화영상을 사용하여 경계선 검출과정을 거친 후 직선검출을 하게 된다. 장애물 인식부에서는 인식된 양쪽 차선 내부에 존재하는 차량을 찾게 된다. 그 차량의 중심점을 찾아 본 논문에서 제안하는 좌표변환을 이용하여 현차량과의 상대적인 위치를 검출할 수 있다[13,14,15].

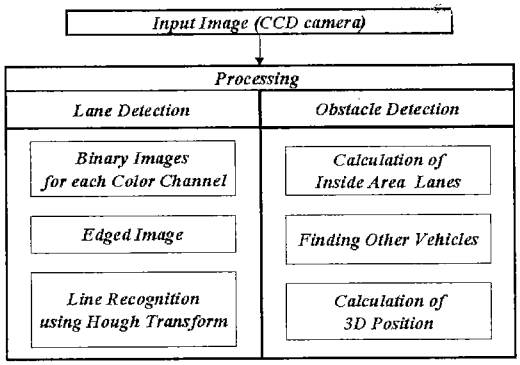


Fig. 1 Block diagram of processes

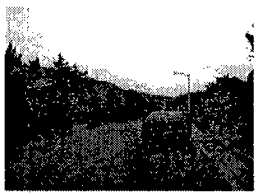
2.1 차선검출을 위한 전처리

획득된 영상에서 차선부분을 분리해낸다면 주행하고 있는 차량의 현재 상태를 알 수 있다. 운행하

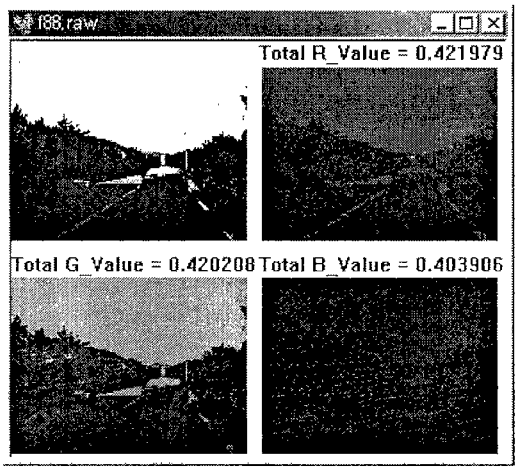
고 있는 양쪽 차선을 찾아 차량 중심과 차선 내부 중심과의 거리오차와 각도오차를 구할 수 있다.

2.1.1 색상 비율을 이용한 전처리

CCD 카메라에서 입력되는 컬러 영상은 RGB (red, green, blue)의 삼색으로 각각 256 레벨(level)로 조합되어 있다. 컬러 영상정보를 이용하면 흑백 영상보다 특징정보를 추출함에 있어 더 많은 방법으로 접근할 수 있다. 본 논문에서는 차선을 추출하기 위하여 아스팔트 바닥면과 차선만을 구분하기 쉽도록 각 채널의 경계치를 전체 색상 적분값에서 각 채널이 차지하는 비율로 설정하여 이진화과정을 거치게 된다. 그림 2(a)의 원영상에 대하여 각 채널에 대한 색상 비율을 구하고, 이를 설정치로 하여 각 채널당 이진영상(binary image)을 구한 것이 그림 2(b)이다.



(a)



(b)

Fig. 2 (a) The original image, (b) Binary images for each channel

2.1.2 경계선 검출

영상의 경계선을 검출할 경우 입력 영상에 대한 많은 정보들을 얻을 수 있다. 경계선에는 물체의 외각선 부분, 크기, 표면의 모양에 대한 정보를 가지게 된다. 영상의 경계선은 색상 또는 밝기가 급격히 변화하는 부분에서 얻을 수 있고, 그 미분값이 설정된 경계치 보다 크면 경계선을 추출할 수 있게 된다. 1차 미분에 대하여 행렬미분연산자로 마스크(mask)를 이용하게 되면, 연산에 소요되는 처리 시간을 절약할 수 있다. 본 연구에서는 3×3의 마스크를 사용하여 경계선을 추출하였다.

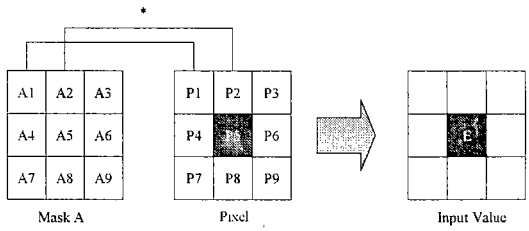


Fig. 3 The calculated Input value by mask

그림 3은 마스크 연산을 나타내며 픽셀점 P_5에 대한 연산이다. 즉, 식(1)을 이용하여 P_5 주위의 나머지 여덟 개의 픽셀을 마스크 A와 연산하여 지정된 위치에 입력될 값을 구할 수 있다. 본 연구에서는 Robert's mask를 이용하였다[6].

$$E = A_1 * P_1 + A_2 * P_2 + \cdots + A_8 * P_8 + A_9 * P_9 \quad (1)$$

2.1.3 허프변환을 이용한 직선 검출

경계선이 검출되면 다음 과정으로 허프변환(Hough transform)으로 차선으로 추정되는 부분을 찾게 된다. 허프변환은 평면상의 일반적인 직선의 방정식 형태 또는 극좌표계로 표현할 수 있다. 본 논문에서는 처리속도를 빠르게 하기 위하여 일반적인 직선의 방정식을 이용하여 두 점을 잇는 직선의 형태로 변환하였다. 그리고 직선을 잇는 두 점을 영상의 가장자리로만 이동 시켰다. 그림 4는 입

력영상에서 2차원 좌표를 설정하여 화살표 방향으로 아래 부분 점들을 이동시켜 직선을 검출하는 과정을 나타낸다.

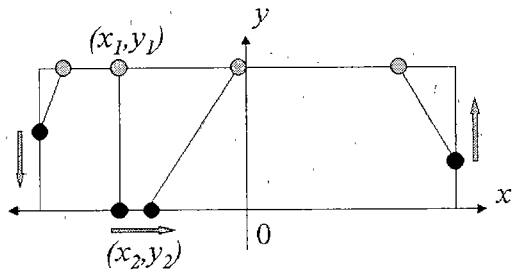
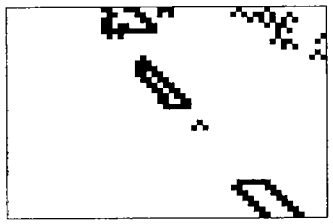
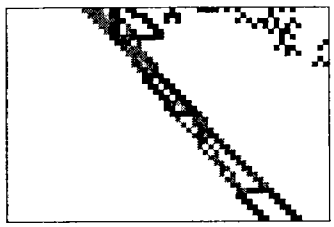


Fig. 4 Connected Lines with two points

차선의 종류는 보통 백색 또는 황색으로 이루어진 실선과 점선의 형태이다. CCD 카메라로부터 들어오는 영상신호에서 차선을 그림 5(a)와 같다고 하면, 직선성분을 조사하여 설정치를 초과하는 직선들의 후보를 찾아야 한다. 그림 5(b)는 직선성분을 만족하는 직선 후보들을 나타내며, 이들 중에서 그림 4의 원점상의 y축에 가까운 점(x_2, y_2)를 가진 직선을 선택한다. 이때 점 (x_1, y_1)을 기억하고, y축과의 기울어진 각도를 계산한다. 선택된 양쪽 두



(a)



(b)

Fig. 5 (a) Edges, (b) Line candidates

직선의 바닥점 위치와 기울어진 각도를 이용하여 차선내의 중심선을 계산하면 차량과 중심선과의 거리오차와 각도오차를 찾아 낼 수 있다.

2.2 위치 검출을 위한 좌표변환

무인차량의 작업환경 내에서 물체들이 바닥과 접촉되어 있고 바닥의 경사가 심하지 않다면, 굳이 두 대의 카메라를 쓰지 않아도 제안하는 방법을 이용하여 전방 차량의 상대적인 위치 및 변위를 알아낼 수 있다. 바닥면에 근거한 단안기법은 먼저, 물체의 위치와 움직임을 포착하기 위하여 대상 물체가 포함된 영상좌표계를 3차원적 실좌표계로 변환한다. 제안하는 Viewport는 바닥면에 근거한 3차원 공간과 CCD 카메라로 입력될 2차원 영상정보와의 변환을 위하여 좌표를 설정한 것이다. 그림 6은 좌표계의 측면을 나타낸 것이고, 카메라 수직 · 수평 화각 내에 존재하며 바닥에 엎혀진 평면이 바로 2차원과 3차원을 상호 변환하게 하는 경계이다. 제안하는 좌표변환방법에서 카메라 설치로 인해 생긴 가상의 평면이다. 즉, 이 평면에 투영되는 상이 실제 얻어진 영상화면과 일치하게 된다. 투영된 영상에서 추출된 물체의 픽셀요소에서 가장 아래 부분의 픽셀을 위치 중심좌표로 선정하였다.

좌표계는 카메라의 높이 h와 viewport 안의 실제 수직 · 수평 거리 x_l, y_l, 카메라와 최하단 viewport면과의 거리 y_b에 의해 변환될 수 있다. 이 값들은 처음 CCD 카메라를 설치할 때 측정하여야 하고, 이후 고정한다. 여기서, $(\alpha-\theta)$는 카메라의 수직화각이 되고, 그림 6의 중앙에 위치한 기울어진 평면의 중앙점에 대한 수직벡터는 설치된 카메라 렌즈의 중앙부분과 평면의 중앙점을 지나는 벡터와 일치하게 된다.

그러므로 식 (2)와 같이 α는 h와 y_b에 의해 나타낼 수 있고, θ는 y_b, y_l, h에 의해 구할 수 있다.

$$\alpha = \tan^{-1}\left(\frac{h}{y_b}\right)$$
$$\theta = \tan^{-1}\left(\frac{h}{y_b + y_l}\right) \qquad (2)$$

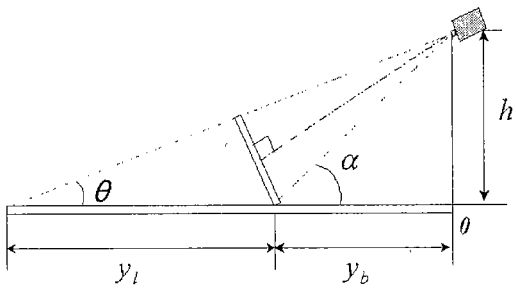


Fig. 6 The side view of viewport

식 (2)에 의해 바닥면에서 카메라 중심축으로 투영된 y축에 해당하는 거리는 카메라 화각과 설치각에 의해 변환되었다. 카메라 설치위치와 x축으로부터 그림 7과 같이 가운데 회색면으로된 viewport를 가정할 수 있다. 이때, 옅은색으로 채워진 역마름모꼴의 바닥면이 viewport에 투영되며, 이것이 카메라로 측정할 수 있는 공간이 된다. 바닥면 x축 위치는 y축에 근거하여 구할 수 있으며, 최대 측정할 수 있는 y축에 대한 x축 범위가 영점(카메라 위치)과 이루는 각이 카메라의 수평화각과 일치하게 된다.

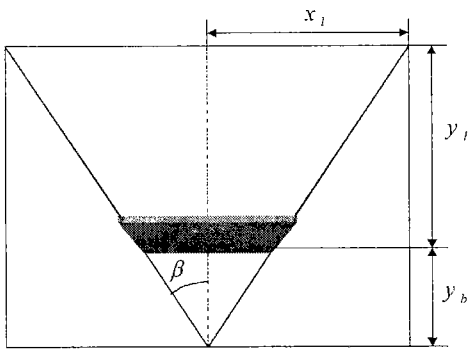


Fig. 7 The map of viewport

그러므로 식 (3)과 같이 수평화각과 관련된 β를 나타낼 수 있다.

$$\beta = \tan^{-1}\left(\frac{x_l}{y_b + y_l}\right) \qquad (3)$$

Viewport에 투영된 x좌표를 P_x라 하면 이는 y축의 각 거리에 해당하는 x축의 좌표 또한 각각 달라진다. 이로부터 영상에서 투영된 좌표를 이용하여 실제좌표를 구할 수 있다.

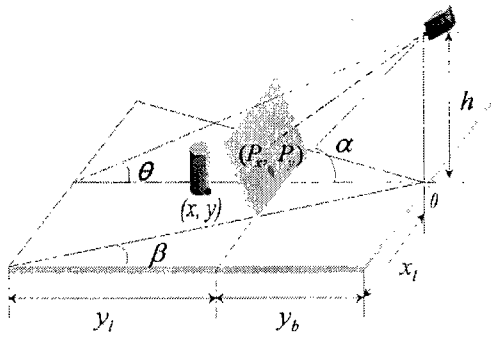


Fig. 8 The viewport

이상의 관계를 이용하여 3차원적으로 좌표계를 나타내면 그림 8과 같다. 측정 범위 안에 임의의 원통체가 있을 경우 투영된 2차원영상 즉 viewport 상의 좌표 픽셀 (P_x, P_y)를 이용하여 실좌표 (x, y)의 관계를 나타내려면 먼저 y축에 해당하는 좌표를 구해야 한다.

그림 9에서 원통체의 위치점을 점 R이라고 했을 경우, viewport상의 좌표는 점 V에 위치하게 된다.

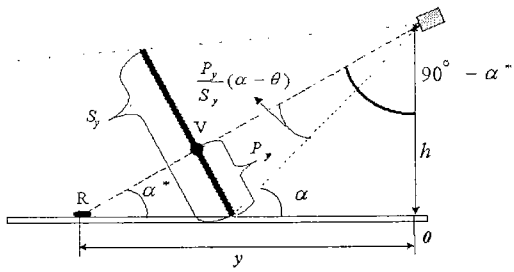


Fig. 9 The pixel point and real position on y axis

y축 상에 투영된 카메라 중심축 위의 점 R의 실제 거리 y는 α^*에 의해 아래의 식 (4)로부터 구할 수 있다. 그리고 $(90° - \alpha^*)$는 식 (5)에 의해 근사

화될 수 있다. 이는 수직화각을 이용함에 있어서 화살표가 가리키는 각이 정확히 S_y에 대한 P_y의 비가 아니라는 것을 알 수 있다. 이는 viewport를 평면이 아닌 카메라 중심축에서 점 V까지의 반지름을 갖는 원주면으로 보아야 하기 때문이다.

$$y = h \times \tan(90^\circ - \alpha^*) \tag{4}$$

$$(90^\circ - \alpha^*) = \frac{P_y}{S_y}(\alpha - \theta) + (90^\circ - \alpha) \tag{5}$$

식 (5)를 식 (4)에 대입하면 실제 점 R의 y축위치를 설정치 h, α, θ, S_y와 입력변수 P_y에 의해 식 (6)과 같이 구할 수 있다. 즉, viewport 상의 픽셀점 좌표를 이용하여 실제 y축의 위치를 구할 수 있다.

$$y = h \times \tan[(90^\circ - \alpha) + (\frac{P_y}{S_y})(\alpha - \theta)] \tag{6}$$

임의의 점 R의 x축에 대한 실제위치는 그림 10에 의해 구할 수 있다.

점 R이 위치한 실제 좌표를 (x, y)라고 하면, y축에서 좌표점을 찾는 방법과 마찬가지로 점 R이 갖는 각 β에서 S_x에 대한 P_x의 비로 근사화할 수 있다. 따라서 실제의 x좌표는 식 (7)과 같이 나타낼 수 있다.

$$x = y \times (\frac{P_x}{S_x})\tan\beta \tag{7}$$

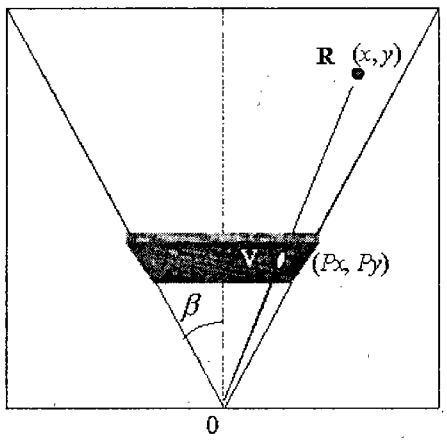


Fig. 10 The real position x based on y axis

단안 기법을 이용하였기 때문에 그 물체의 길이는 지금까지의 전처리 과정으로는 찾기가 어렵다. 즉 물체 자체의 정확한 3차원적 정보는 추출하기가 어렵다. 그러나 추출된 물체가 2차원 영상 속에 존재하므로 그림 11과 같이 위치는 가지고 있으나 투영된 나머지 뒷부분의 형태를 고려하지 않는다면 그리 문제될 것이 없다. Viewport에 투영된 물체의 부분영상의 좌측상단 픽셀좌표와 우측하단 픽셀좌표를 이용하여 너비와 높이를 계산할 수 있다. 즉 투영된 직사각형 모양의 부분영상의 크기가 실제 물체의 크기로 검출될 수 있도록 하였다.

너비는 픽셀 좌표를 실제 x축 좌표로 변환시키는 식 (7)을 이용하여 픽셀상의 직사각형 너비를 실제 너비로 변환할 수 있다. 높이도 마찬가지로 y축이 아닌 그 위치에서 x축에서 변환하는 식 (7)을 사용하여 높이를 구하여야 한다.

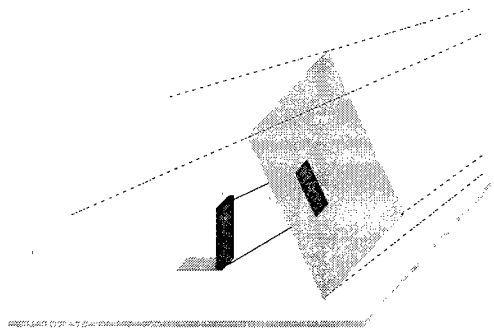


Fig. 11 An assumed object to a plane

2.3 전방 차량의 움직임 포착

이상의 처리과정으로 입력영상으로부터 차선상태와 전방 차량의 상대적인 위치를 검출할 수 있다. 전방차량의 움직임을 포착하기 위해서는 그 물

체의 실제위치가 중요하다. 그러나 입력영상, 즉 작업공간 내에 물체가 여러 가지일 경우에는 물체별로 구분해야 하는 문제가 발생된다.

영상처리를 이용하여 이상의 처리과정을 수행할 경우 현재까지는 평균 300[ms]의 시간이 소요된다. 그리고 CCD 카메라에서 신호가 들어오는 시간까지 감안해야 한다. 그러므로 실제로 일어나는 상황보다 연산되어 나온 결과는 시간적으로 지연성을 갖게 된다. 실시간 처리를 위해서는 다음 프레임에 영향을 줄 수 있는 상황의 예측이 필요하다. 프레임과 프레임 사이의 정보를 담아 다음 프레임에 대한 예측에 이용하기 위하여 버퍼(buffer)역할을 하는 히스토리 정보가 사용된다.

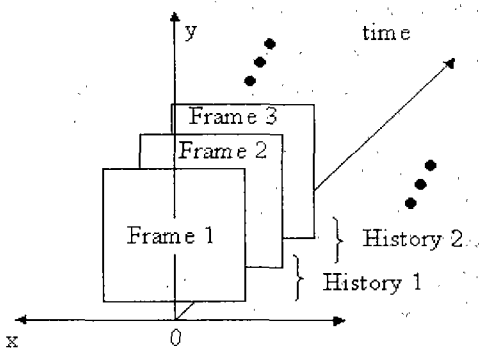


Fig. 12 The history information for each frame

현재의 영상에서 대상 물체들을 검출하여 생성된 히스토리 안에는 이전과 현재 프레임에 대한 물체 고유의 크기(너비, 높이)와 실좌표계로 변환된 좌표점이 기록되어 있다. 그림 12는 각 프레임 영상으로부터 구해진 히스토리를 나타낸다. 즉, 히스토리에는 이전의 정보와 현재의 정보가 저장되어 있으며, 현재의 정보는 다시 다음 프레임에 대한 히스토리의 이전정보로 입력된다. 또한 전처리과정으로부터 검출된 각 물체들은 고유 레이블을 가지고 있고, 전후 프레임들의 레이블은 일치하여야 하므로, 동일한 레이블을 갖는 물체는 히스토리 안에 현재의 위치와 이전의 위치로 변위를 계산하여 움직임을 알아 낼 수 있다.

그림 13에서 보는 바와 같이 히스토리 내부의 정보는 연계되어 있다. 현재 프레임에서 물체의 변위는 이전 프레임의 위치와 현재 위치의 차분값으로 알아낼 수 있다. 각 물체에 대한 변위를 구하게 되면, 상대적으로 물체가 차량에 대해 어떻게 움직이고 있다는 것을 알 수 있다. 차량에 CCD 카메라를 부착하여 적용한다면 여러 방향으로 움직이고 있는 물체 중에서 급격히 다가오는 물체에 대해서는 장애물로 인식되어 충돌의 위험을 방지할 수 있을 것이다.

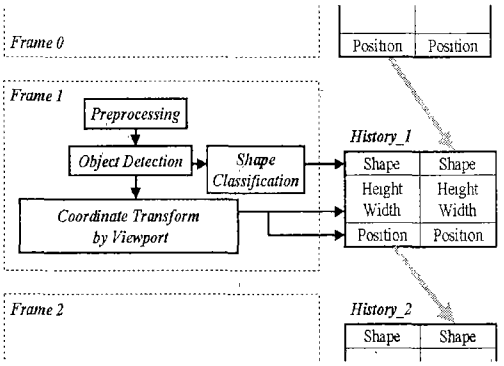


Fig. 13 A detection of displacement of objects using history

3. 실험결과 및 고찰

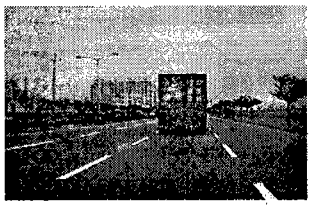
3.1 실험 장치 및 방법

본 실험에 사용된 카메라는 수평 48°, 수직 36°의 화각을 가지며, h = 1.3[m], θ = 0°, 그리고 해상도는 처리속도의 신속성과 계산량을 줄이면서도 물체를 검출함에 있어 무리가 없도록 하기 위하여 영상의 사이즈는 160(H)×120(V) 또는 360(H)×240(V)의 픽셀수를 갖도록 설정하였다. PC는 Windows 98의 OS환경 하에 CPU는 Pentium II MMX 266MHz를 사용하였다. 실험방법은 주행하는 차량의 운전석에 CCD카메라를 설치하여 입력

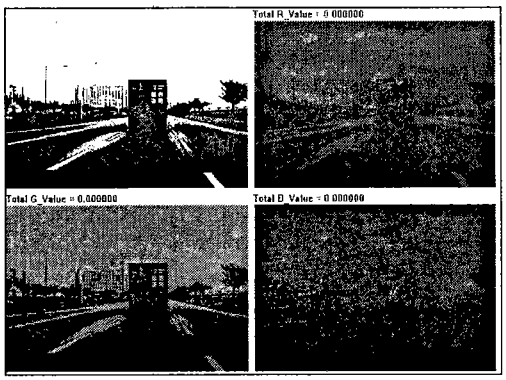
된 영상을 분석하였다. 입력영상은 RGB 각 채널당 256색의 컬러영상을 사용하였다. 입력영상은 주행하는 차량에서 획득된 동영상 데이터를 이용하였다. 알고리즘 구현에 사용된 소프트웨어는 Visual C++를 사용하였다.

3.2 차선인식

먼저 차선을 검출하기 위해 색상비율을 이용한 각채널에 대한 이진영상의 결과가 그림 14에 나타나 있다.



(a)



(b)

Fig. 14 (a) Original image, (b)Binary Images for each color channel

각 채널의 이진영상이 겹치는 부분은 백색으로 나타남을 볼 수 있다. 각 채널 영상을 AND연산하게 되면 이 백색 부분만 남게 된다. 그림 15는 Robert's mask를 이용한 경계선 검출에 대한 결과이다.

경계선이 구해진 영상에서 수직 1/2의 하단부에 대하여 차선을 검출하도록 하였다. 두 점을 잇는

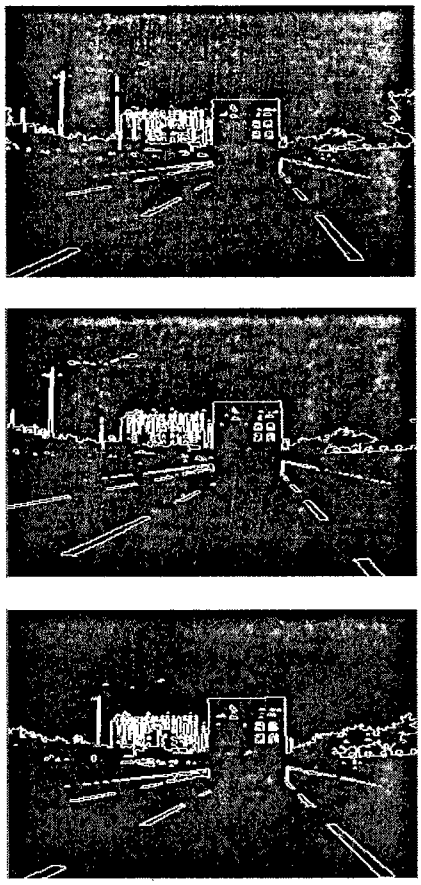


Fig. 15 Edged images for each frame

직선으로 탐색하였으며, 연결 성분이 많은 직선성분을 검출하여 후보직선들을 검출한 결과가 그림 16에 나타나 있다. 적색으로 표현된 직선들이 차선에 대한 후보들이다. 직선 후보 중에서 각 프레임당 차선으로 인식된 두 직선을 찾고, 표 1에 각 프레임에서 선택된 후보 차선에 대한 정보들을 나타내었다.

3.3 전방차량 위치 검출

그림 17은 인식된 두 차선 안쪽 영역에서 전방차량을 검색하여 중심점을 찾은 결과이다. 색상비율을 이용하지 않고, 경계선을 검출하였다.

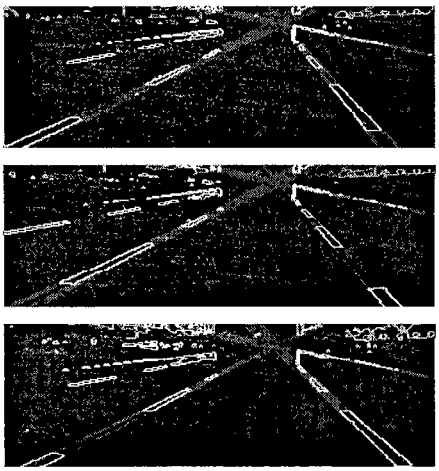


Fig. 16 Candidates for the road lane

Table 1 Information of line parameter

	frame 1	frame 3	frame 9
up-point of left line	(78, 118)	(76, 118)	(86, 118)
bottom-point of left line	(-162, 0)	(-151, 0)	(-146, 0)
up-point of right line	(40, 118)	(35, 118)	(30, 118)
bottom-point of right line	(133, 0)	(135, 0)	(143, 0)
angle of left line [degree]	63.8	62.5	62.8
angle of right line [degree]	-38.2	-40.3	-43.8
distance error from center [pixel]	-29	-16	-3
angle of center line [degree]	25.6	22.2	19.0

표 2는 전방 차량의 영상평면상의 좌표를 도로평면 상으로 변환한 결과이며, 그림 18은 이를 전방 8~18 [m], 좌우 -2~2 [m] 공간좌표에 나타낸 것이다. 약 15 [m] 전방에서 12 [m]까지 접근함을 볼 수 있다.

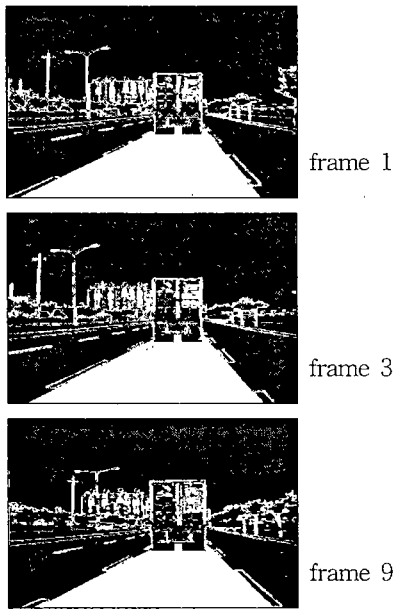


Fig. 17 The detection of position point on lane

Table 2 AGV position for each frame

	frame 1	frame 3	frame 9
point of image plan	(34, 86)	(33, 85)	(32, 78)
y axis point on 3D plan [m]	15.22	14.75	12.16
x axis point on 3D plan [m]	1.28	1.20	0.96

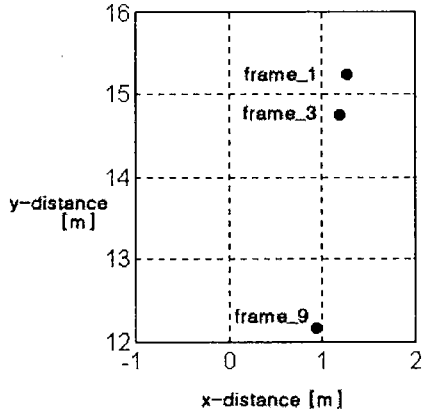


Fig. 18 Positions for each frame

4. 결 론

비전센서를 이용하여 얻어진 영상정보는 전처리 과정을 거치게 되는데, 이 전처리 과정의 결과에 의해 차선과 차량 상태의 신뢰성이 결정된다고 해도 과언이 아니다. CCD 카메라에서 얻어진 컬러영상에서 배경의 적색, 녹색, 청색의 비율을 이용하여 차선부분을 추출하였다. 또한, 경계선 검출과 허프 변환을 이용하여 차선의 상태를 구할 수 있었다.

만약 차선을 정확하게 파악했다면, 차선 사이 영역에 있는 다른 차량을 찾아 상대적인 위치를 얻을 수 있다. 전방 차량으로 추정되는 영상 부분의 가장 아랫부분의 중앙점을 위치점으로 설정하고, 영상에서의 좌표를 실제 그 물체의 위치로 변환하였다. 처리시간을 줄이고, 효율성을 높이기 위해 하나의 CCD 카메라를 사용하였으며, 계산된 물체의 위치는 프레임별로 저장되어 CCD 카메라와의 상대적인 변위를 알 수 있도록 하였다. 물체가 CCD 카메라와 상대적으로 어떤 움직임을 취하고 있는지를 파악할 수 있으므로, 다가오는지 아니면 멀어지는지를 알 수 있으며, 그 물체의 다음 프레임의 위치를 예측할 수 있을 것으로 기대된다.

시뮬레이션 결과, CCD 카메라로부터 멀리 있는 물체일수록 픽셀당 계산되는 거리의 범위가 커지므로 오차가 커질 것으로 예상했지만, 오히려 가까이 있는 물체가 검출과정에서 부분적인 손실로 인해 오차가 더 크게 나왔다. 즉, 제안하는 좌표계나 CCD 카메라 분해능에 의해 발생하는 오차에 비해 물체를 추출하고 위치점을 부여하는 영상처리과정에서 발생할 수 있는 오차가 크다는 것을 알 수 있었다.

참고문헌

1) I. Masaki, *Vision-based Vehicle Guidance*, Springer-Verlag, 1991.
2) G. Cheng and A. Zelinsky, Real-time Visual Behaviours for Navigating a Mobile Robot, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 2, pp. 973-980, 1996.
3) S. Ishikawa, K. Gouhara, K. Ide, and K. Kato, A Vision Based Mobile Robot Travelling Among Obstructions, KACC, pp. 810-815, 1988.
4) 김영집, HSI 컬러 모델을 이용한 실내주행 이동로봇의 장애물 고속 인식, 석사학위논문, 한국과학기술원, 1996.
5) A. Guiducci, 3D Road Reconstruction from a Single View, Computer Vision and Image Understanding, Vol. 70, No. 2, pp. 212-226, 1998.
6) R. Jain, R. Kasturi, and B. G. Schunch, *Machine Vision*, McGraw-Hill, 1995.
7) A. Low, *Introductory Computer Vision and Image Processing*, McGraw-Hill, 1991.
8) NHK 방송기술연구소 영상연구부, *C 언어에 의한 화상처리실무*, 국제테크노정보연구소, 1994.
9) 김성호, 김병국, Stereo Vision을 이용한 이동로봇의 실내주행을 위한 Environment Map의 작성, 한국자동제어학술회의논문집, pp. 944-949. 1994.
10) 김계영, 퍼지추론에 의한 차영상의 이진화, 한국퍼지 및 지능 시스템학회 논문지, Vol. 9, No. 3, pp. 352-359, 1999.
11) 오창윤, 물체와 배경의 분리를 이용한 물체 추적, 석사학위논문, 포항공대, 1994.
12) Randy Crane, *A Simplified Approach to Image Processing*, Hewlett-Packard Company, 1997.
13) 이광호, 거리영상의 새로운 분류방식과 이의 신경회로망 모델화, 박사학위논문, 경북대학교, 1991.
14) 한수환, A Study On 2-D Shape Recognition Using higher-Oder Spectra and LVQ, 한국퍼지 및 지능 시스템학회지 논문지, Vol. 9, No. 3. pp. 285-293, 1999.
15) S. M. Smith, Integrated Real-Time Motion Segmentation and 3D Interpretation, Proceedings of ICPR, pp. 49-55, 1996.