

무인차량의 도로주행 방법

박범주* 한민홍**

포항공과대학 산업공학과* 고려대학교 산업공학과**

Road Following of an Autonomous Vehicle

Bum-Ju Park *
Dept. of Industrial Eng. *
POSTECH

Min-Hong Han *
Dept. of Industrial Eng. *
Korea University

ABSTRACT

In this paper we describe a road following method for an autonomous vehicle. From a road image in gray level, a road boundary is detected using a gradient operator, and then the road boundary is converted to orthogonal view of the road showing the vehicle position and heading direction. In this research an efficient road boundary search technique is developed to support real time vehicle control. Also, an obstacle detection method, using images taken from two different positions, has been developed.

1. 서론

무인차량의 도로주행에 관한 연구는 로봇비전 및 인공지능분야에서 끊임없는 도전의 대상이 되어 왔다. 이러한 도로주행을 수행하기 위해 무인차량이 갖추어야 할 총체적인 기능은 불확실한 도로환경을 정확히 인식할 수 있는 지능의 소유라고 볼 수 있다. 그래서 많은 연구자들은 그러한 기능을 위해 비전을 이용한 도로주행(vision-based road following) 시스템의 구축을 희망하여 왔다. 이것은 사람이 차를 운전할 때 가장 중요한 것이 눈의 역할이라고 볼 때 이상적인 착안이라고 볼 수 있다. 그런데 눈의 역할을 하는 이미지 센서를 통해서 얻은 데이터로부터 무인차량이 주행하기 위한 정보를 모두 추출해 내는 데에는 여러가지의 모듈이 필요하다. 그 중에서도 일차원적인 도로주행 - 도로상을 안전하게 따라갈 수 있는 - 을 위해 가장 중요한 것은 다음과 같다.

첫번째는 무인차량이 도로지역과 비도로지역을 빠른시간내에 구분할 수 있는 이미지 분석능력이다. 이것은 구체적으로 센서(CCD카메라, laser range scanner 등)를 통해 들어오는 이미지를 분석하여 도로경계에 관한 정보를 추출해 낸 후 3차원 도로형태를 기하학적으로 복원하는 작업에 관한 것이다.

두번째는 도로경계내에 불규칙적으로 존재하게 될 장애물의 인식능력이다. 대개의 경우 스테레오 정합으로 문제를 귀결시킴으로써 장애물의 3차원 정보 - 장애물의 특성(높이, 폭)과 장애물까지의 거리를 포함한 - 들을 계산해 낼 수 있는데 두

대의 카메라를 사용하는 경우 컨트롤의 난점, 계산의 복잡성 등의 문제들이 야기되어 왔다. 먼저 기존의 도로주행 시스템들이 도로경계의 정보추출과 장애물 인식을 어떻게 수행했는지 살펴보기로 한다.

카네기 멜론 대학의 Navlab은 먼저 카메라로부터 들어오는 칼라이미지를 축소(reduction)한 후 픽셀 R-G-B값의 평균과 공분산행렬에 기초한 네가지식의 도로형태와 비도로형태를 구하여 각 픽셀이 분류된 도로형태에 속할 확률을 계산함으로써 이미지 전체를 도로지역과 비도로지역으로 분할하였다. [TK88, Kl89]. 이것은 분류된 도로형태의 신뢰성여부, 조명(illumination)변화의 민감성에 따른 많은 도로형태 분류의 필요성등의 문제점들이 야기된다.

Martin Marietta VITS 시스템은 픽셀의 3차원 칼라 이미지를 도로지역과 비도로지역으로 구분한다는 것은 Nalab시스템과 비슷하지만 레드-블루(Red-Blue) 평면으로 투사된 도로 이미지를 둘로 나누는 선형판별함수(linear discriminant function)의 기술기를 드레쉬홀딩(thresholding)하는데 초점을 맞추었다. [TM88, PDM87] 한편, 레인지 이미지(range image)로 부터 정지된 장애물 지역을 뽑아낸 후 수직높이를 계산함으로써 이를 피해갈 수 있는 경로계획자(path planner)를 개발하였다. [Dun88]

Maryland 대학의 시스템은 예상(prediction)된 도로위치에 초기 윈도우를 설정하여 전체를 탐사함으로써 도로경계에 대한 위치와 방향을 결정한 후 윈도우를 그 방향에 따라 옮겨가며 도로경계를 추출하였다. [WLD87, DiD90]

본 연구의 첫번째 주제는 일반적인 도로이미지에 대하여 특정 그레디언트 오퍼레이터(specialized gradient operator)와 평가함수(evaluation function) - 도로경계가 존재하는 국부적인 지역만을 탐사하게 해주는 - 를 사용하여 한 프레임의 도로이미지로부터 도로경계를 빠른시간내에 찾을 수 있는 도로경계추출 알고리즘에 관한 것이다.

두번째 주제는 한대의 카메라에서 얻은 연속된 두개의 이미지로부터 도로상의 임의의 위치에 있는 장애물을 감지하는 방법에 관한 것이다. 이 경우는 두대 카메라에 의한 스테레오 정합과는 다른 경우으로써 계산의 간편성을 기할 수 있다.

2. 도로 경계 추출 방법

2.1 도로 모델

무인차량의 일차원적인 도로주행에 있어서 모델로 삼은 도로의 형태는 두가지이다. (그림2.1) 첫째는 고속도로와같이 가이드라인이 연속적이지는 않지만 어느 정도 제시된 형태이다. 이런 형태는 가이드라인 부분(흰색 또는 노랑색)과 다른 도로 노면 사이에 그레이 레벨 강도(gray level intensity)차이가 크므로 거의 이원(binary) 이미지에 가깝다는 특수성도 지닌다. 둘째는 시골길처럼 도로의 경계선이 애매한 형태이다. 이 경우는 부분적으로 도로지역과 비도로지역이 불분명한 상태도 나타나며 그림자를 비롯한 많은 노이즈를 포함하고 있다.

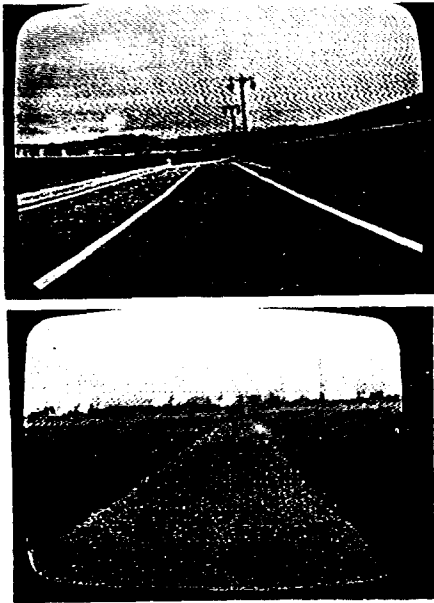
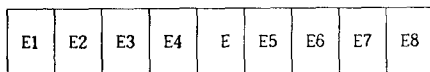


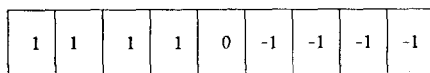
그림2.1 도로 이미지
Fig.2.1 Road image

2.2 도로 경계 식별자의 선택

두가지의 도로형태에 공통적으로 적용되어 질 수 있는 도로경계식별자(road boundary detection operator)의 선택은 도로주행 시스템의 일반성 측면에서 무엇보다도 중요하다. 그래서 그레이레벨 강도변화에 중점을 둔 식별자를 사용한다. 그것에 관한 정량적인 설명은 다음과 같다.



(a)



(b)

그림2.2 (a) 그레디언트 오퍼레이터 (b) 경계 템플릿

Fig2.2 (a) gradient operator (b) edge template

그림2.1과 같은 도로이미지 상의 임의의 점 E(pixel)가 도로경계점인지를 결정하기 위해 그림2.2의 (a)와 같은 1차원 그레디언트 오퍼레이터를 잡고 (b)와 같은 템플릿을 적용시킬때 점 E에 관한 그레디언트 값 - 그레이레벨 강도변화와 관련된 - 이 다음과 같이 얻어진다.

$$\text{그레디언트} = \text{abs}((E1+E2+E3+E4)-(E5+E6+E7+E8)) \quad (1)$$

여기서, E1부터 E8까지는 해당점의 그레이 레벨이다.

그러면 도로경계에서는 그레디언트 값이 다른 이미지점들 보다 큰값을 가질 것이다. 그렇게해서 도로 이미지상의 어떤 점이 경계점인지의 여부를 기본적으로 결정할 수 있다.

2.3 도로 경계 추출 알고리즘

도로경계식별자를 사용하여 임의의 도로이미지 프레임을 효율적으로 탐사함으로써 빠른시간내에 도로경계를 추출해 내는 방법을 서술하기로 한다. 여기서 효율적이라 함은 이미지 전체가 아닌 도로경계를 따라서 부분적인 구역만을 탐사한다는 것을 의미한다. 이러한 기능을 위해 탐사 알고리즘은 추출된 도로 경계점들의 기울기변화를 이용하는데 이는 일반적으로 도로의 경계선이 국부적인 지역으로 나누어서 생각해 볼 때 그 경사가 갑작스럽게 변하지 않는다는 제한조건(smoothness constraint)에서 힌트를 얻을 수 있다.

구체적인 탐사알고리즘을 설명하면 다음과 같다.

- 1 단계 : 정해진 윈도우(window)의 최저선(bottom line)에서 도로경계 식별자를 사용하여 최대 그레디언트값에 해당되는 점을 감지함으로써 왼쪽과 오른쪽 도로의 경계점을 찾는다.
- 2 단계 : 평가함수 - 현재기울기(current slope), 평균기울기(average slope), 런의 길이(run length)의 함수 - 에 의해 다음의 도로경계점이 있을것으로 예상되는 부분에 탐사구간(search interval)를 형성한다. 여기서, 현재기울기는 현재의 경계점과 바로 직전의 탐사구간에서 구한 경계점사이의 기울기이고 평균기울기는 현재까지 구한 모든 경계점들사이의 기울기이며 런의 길이는 찾은 경계점의 갯수이다. 한편, 현재기울기와 평균기울기의 초기값은 지정해 준다.
- 3 단계 : 결정된 탐사구간내에서 한개의 도로경계점을 찾고 다시 평가함수를 계산한다. 만약 주어진 구간내에서 도로경계점을 찾지 못하면 바로 직전에 사용한 구간의 연장선상에서 탐사구간을 새로 형성한다.
- 4 단계 : 주어진 윈도우내에서 단계 2와 단계 3을 반복하여 모든 도로경계점을 찾는다.
- 5 단계 : 찾이진 점들은 2차원 좌표좌표계상의 점들이므로 이것을 3차원 차량좌표계(vehicle coordinate)로 바꿔

주는 퍼스펙티브 변환(perspective transformation)을 통해 변환시킴으로써 도로평면을 수직으로 내려다본 도로의 수직조망(road orthogonal view)을 얻는다.

2.4 도로 이미지 처리

위와같은 알고리즘에 의해 도로경계를 탐사를 할 경우 그림 2.2의 그레디언트 연산자가 움직이는 지역과 방향은 그림 2.3의 화살표가 그리는 자취가 된다. 이것은 그레디언트 연산자가 도로의 일부분만을 탐사한다는 사실을 말해주고 있다. 한편 그림 2.4는 직선과 곡선길에서 도로경계를 정확하게 찾은 이미지처리 결과를 보여주고 있다. 이미지 처리는 SeeEye-256 프레임그라버(frame grabber)를 사용하여 IBM-PC 286으로 수행하였는데 각 이미지 프레임당 도로경계를 찾기위해 걸리는 시간은 0.5초 미만인 걸렸다. 이러한 결과는 일반적으로 무인차량의 속도가 20km/hour 정도가 될려면 각 이미지 프레임을 처리하는데 걸리는 시간이 1.5초이하가 걸려야 한다는 것을 고려해 볼 때 도로주행 시스템이 갖추어야 할 요건중에서 가장 중요하다고 할 수 있는 실시간 제어능력을 확실히 뒷받침해주는 결과라고 할 수 있다.

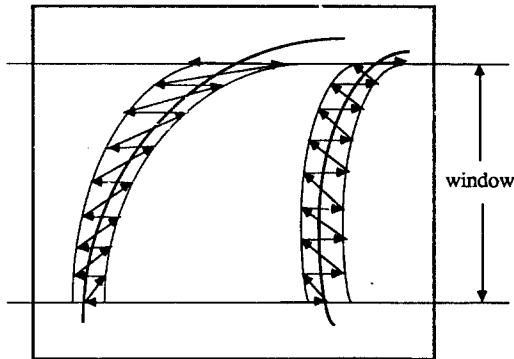


그림 2.3 도로 경계 탐사 방향

Fig2.3 The search direction of a road boundary

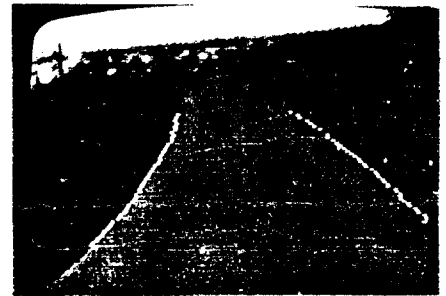
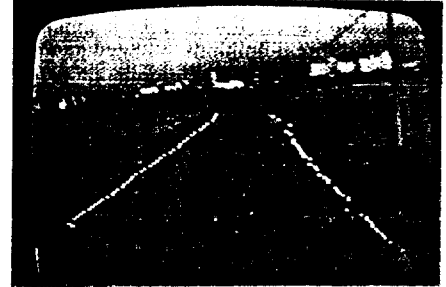
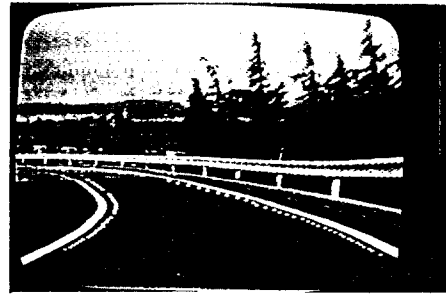
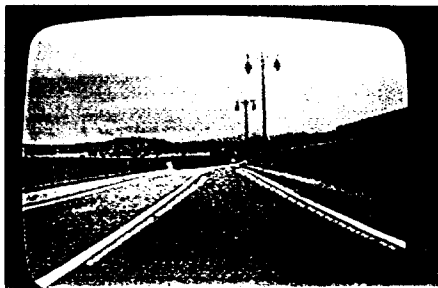


그림 2.4 감지된 도로 경계

Fig2.4 A detected road boundary

3. 장애물 감지 방법

일반적으로 장애물에 대한 정보의 추출에는 두대의 카메라에 의한 스테레오 이미지를 사용하는 경우가 지배적인데 이때 카메라 컨트롤문제와 이미지 일관성 검증 문제 그리고 계산상의 난점등이 야기된다. 그러므로 본 연구에서는 한대의 카메라를 통해 연속된 두개의 이미지를 그림 3.1과 같이 얻은 후 도로경계내에 존재하는 장애물을 경계추적법(boundary trace method)으로 찾았다. (그림 3.2) 이때 장애물은 이미지평면의 각기 다른 위치에 나타나게 되는데 그런 사실을 이용하면 장애물의 높이와 무인차량에서 장애물까지의 거리를 구할 수 있다. 그러면 직선길과 곡선길의 경우로 나누어서 위의 방법론을 살펴보기로 하겠다.

3.1 직선길에서 장애물을 감지한 경우

그림 3.3에서와 같이 무인차량의 카메라가 도로상의 임의의 위치에서 장애물을 감지한 후 직선상을 일정한 거리(D)만큼 이동한 후 동일한 장애물을 다시 감지했다고 하자. 그러한 경

우 장애물의 위치를 기술하기 위한 차량좌표계를 카메라 시속이 도로바닥을 뚫는 점에서 잡고 카메라 팬각(θ)과 틸트각(ψ)이 일정한 상태에서 무인차량이 이동했다면 두 좌표계의 원점 o_1 과 o_2 사이에는 $Tr(x,y)$ 만큼의 이동(translation)이 존재하게 된다. 이러한 좌표변환 관계를 알고 그림3.3을 측면에서 바라다 본 것인 그림3.4의 무시도(perspective view)를 살펴보자. 장애물의 정상점(top point)이 각기 다른 위치에서 무인차량의 카메라에 $A'(Xs1, Ys1)o_1$ 와 $B'(Xs2, Ys2)o_2$ 로 잡혔다면 각각의 차량좌표계에서의 좌표는 퍼스펙티브 변환을 통해 $A(X1, Y1, 0)o_1, B(X2, Y2, 0)o_2$ 로 나타낼 수 있다. 마찬가지로 o_1 과 o_2 에 관한 카메라 좌표는 무인차량의 높이를 h 라 할때 $A''(Xb, Yb, h)o_1, B''(Xf, Yf, h)o_2$ 로 표현될 수 있다. (여기서 $Xb=0, Yb=-htan\psi$) 그런데 무인차량이 일정기리를 이동하기 전후의 각각의 카메라 좌표와 장애물 좌표가 다른 좌표계에서 표현되어 있으므로 식(2)와식(3)에 의해 A, B, A'', B'' 를 동일한 좌표계(o_2)로 바꿀 수 있다.

$$A(X1, Y1, 0)o_2 = A(X1, Y1, 0)o_1 - Tr(x, y) \quad (2)$$

$$A''(Xb, Yb, h)o_2 = A''(Xb, Yb, h)o_1 - Tr(x, y) \quad (3)$$

이때 차량 좌표계상에서 장애물 좌표점과 카메라 좌표점을 잇는 두 3차원상의 직선($A - A'', B - B''$)을 생각 할 수 있다. 그 두 직선의 교점을 그림3.4에 띄여있듯이 $I(k1, k2, H)$ 라고 할때 I 의 z좌표값이 장애물의 높이가 되고 I 와 카메라 좌표점(B'')사이의 거리가 무인차량과 장애물사이의 거리가 된다.

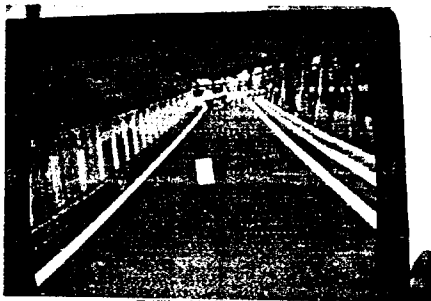
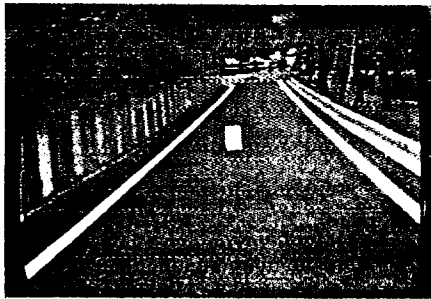


그림3.1 장애물에 대한 연속된 이미지

Fig3.1 The sequential image including obstacle

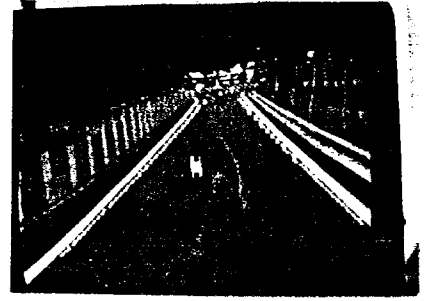


그림3.2 장애물 감지 결과(직선길)

Fig3.2 Detection of an obstacle

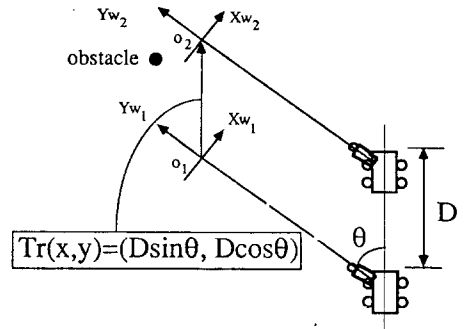


그림3.3 직선길의 이동

Fig3.3 The movement of a straight road

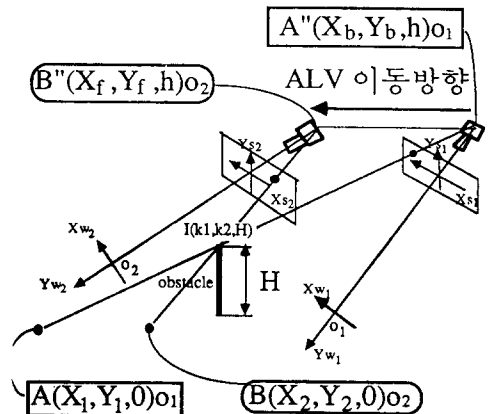


그림3.4 무시도

Fig3.4 Perspective view

3.2 곡선길에서 장애물을 감지한 경우

그림3.5은 무인차량이 곡선길을 이동하다가 장애물을 감지한 경우에 도로명면을 수직으로 내려다 본 그림이다. 무인차량이 이동한 자취를 원의 일부로 근사시킬때 그 원의 중심 (X_c, Y_c) 은 도로경계추출 알고리즘에 의해서 감지된 곡선상에 있는 세점의 수직이등분선의 교점으로 부터 구할 수 있다. 그러면 다음식에 의해서 무인차량이 일정한 거리(D)만큼 이동한 동안 (X_c, Y_c) 을 중심으로 회전한 각도(Ω)가 구해 진다.

$$R = (X_c^2 + (Y_c - h \tan(\psi))^2)^{1/2} \quad (4)$$

$$\Omega = D/R \quad (5)$$

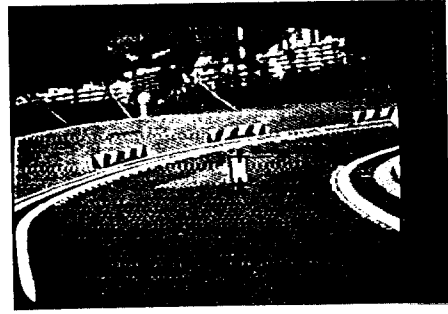
한편, (X_{w1}, Y_{w1}) 좌표계에 관한 (X_{w2}, Y_{w2}) 좌표계의 원점벡터 (X_d, Y_d) 는 식(6)과 같이 표현된다. (회전부호: 반시계방향(+))

$$\begin{bmatrix} X_d \\ Y_d \end{bmatrix} = \begin{bmatrix} \cos(-\Omega) & -\sin(-\Omega) \\ \sin(-\Omega) & \cos(-\Omega) \end{bmatrix} \begin{bmatrix} -X_c \\ -Y_c \end{bmatrix} + \begin{bmatrix} X_c \\ Y_c \end{bmatrix} \quad (6)$$

그러면 두 좌표계사이의 회전각(Ω)과 이동량 (X_d, Y_d) 을 구했으므로 그림3.4에서 (X_{w2}, Y_{w2}) 좌표계로 표시된 장애물좌표 $A(X_1, Y_1, 0)$ 와 카메라 좌표 $A''(X_b, Y_b, h)$ 를 각각 식(7)과 식(8)과같이 (X_{w2}, Y_{w2}) 좌표계로 바꾸어 직선길의 이동에서와 마찬가지로 방법으로 장애물 좌표점과 카메라 좌표점을 잇는 두 3차원직선의 교점으로 부터 장애물의 높이와 무인차량에서 장애물까지의 거리를 구할 수 있다.

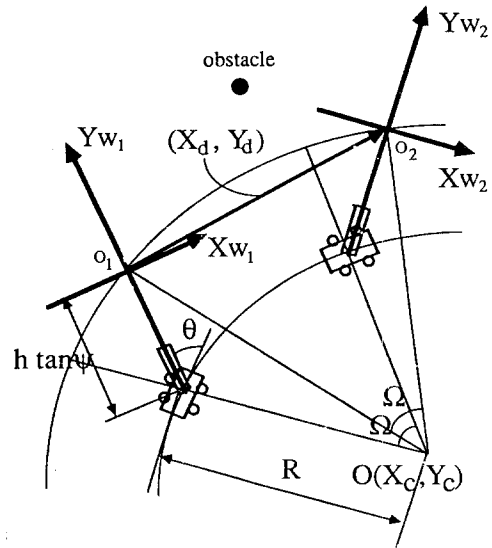
$$\begin{bmatrix} X_1 \\ Y_1 \\ 0 \end{bmatrix}_{o2}^A = \begin{bmatrix} \cos(\Omega) & -\sin(\Omega) & 0 \\ \sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ 0 \end{bmatrix}_{o1}^A \quad (7)$$

$$\begin{bmatrix} X_b \\ Y_b \\ h \end{bmatrix}_{o2}^{A''} = \begin{bmatrix} \cos(\Omega) & -\sin(\Omega) & 0 \\ \sin(\Omega) & \cos(\Omega) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_b \\ Y_b \\ h \end{bmatrix}_{o1}^{A''} \quad (8)$$



(a) 장애물의 감지

(a) An obstacle detection



(b) 수직 투시도

(b) An orthogonal view

그림3.5 곡선길의 이동

Fig3.5 The movement of a curve road

4. 결론 및 추후연구과제

본 연구에서는 무인차량이 비전을 이용하여 도로주행을 하기 위한 방법을 다루었다. 먼저, 차량이 무엇을 보고 갈 것인가의 문제는 입차적으로 도로 경계의 추출을 통해 해결하였다. 이것은 물론 주행할 도로가 도로지역과 비도로지역으로 어느정도 구별되는 형태에서 가능한 것이었다. 이미지를 처리 하는데 있어서 가장 중요시 했던 사항은 빠른시간내에 도로경계를 추출해야 한다는 것이었다. 왜냐하면 실시간 처리가 주행시스템의 필수조건이기 때문이다. 그래서 기존의 대부분의

참 고 문 헌

시스템들이 대규모의 하드웨어를 바탕으로 칼라이미지까지 동원하여 이미지 전체를 탐사함으로써 도로경계를 비롯한 주행 정보를 추출했지만 본 연구에서는 도로경계의 위치를 미리 추정하여 그 방향을 따라 국부적인 부분만을 처리하여 약 0.5초 내에 경계를 추출할 수 있었다. 일반적으로 무인차량이 약 20km/hours 정도로 주행하려면 카메라로부터 들어오는 매 이미지를 적어도 1초이내에 처리해야 한다는 점에서 위의 결과는 매우 의미있는 것이라고 사료된다. 한편, 안정된 주행을 위해 반드시 필요한 비전 모듈이 바로 장애물 감지에 관한 것이었다. 본 연구에서는 두대 카메라가 아닌 한대의 카메라를 사용하여 일정 거리를 사이에 두고 두개 이미지를 얻음으로써 장애물의 3차원 정보를 계산하는 알고리즘을 개발했다. 이는 기존의 스테레오 시스템들에서 야기됐던 계산상의 복잡성과 이미지 대응문제등을 해결하기 위한 시도였는데 무인차량의 카메라가 장애물을 발견했을때 그것의 3차원 정보를 비교적 간단한 계산에 의해 알아 낼 수 있었다. 한대의 카메라로 이러한 경우의 카메라 컨트롤문제를 극복할 수 있다는 사실에서 중요한 의미를 지닌다고 볼 수 있다. 하지만 앞서 언급했듯이 이상의 두가지 비전모듈은 무인차량의 도로주행 시스템에 있어서 일차적인 부분에 불과하다. 주행할 지역의 형태에 구애받지 않는 시스템을 장착한 무인차량을 소위 ALV(autonomous land vehicle)라고 하는데 여기까지 가는데는 수많은 하부시스템들이 필요로 할 것이다. 이미지 처리부분에서만 해도 그림3.6처럼 오른쪽 도로경계가 불확실하여 그 방향을 알 수 없을때 왼쪽 경계선의 방향과 이미지 전후관계를 고려함으로써 도로경계에 대한 정보를 유추할 수 있는 기능이 필요한데 이것은 이미지 일관성문제(image correspondence problem)로 귀착된다. 한편, 좀 더 정확한 도로정보의 산출을 위해 레이저를 사용해야 할 경우 카메라 이미지와 레인지 데이터의 통합(fusion)문제등을 들 수 있고 고차원인식(high level recognition)의 단계에서는 예를들어 도로가 나뉘어지는 부분 등에서 총체적 계획자(global planner)를 설계하는 것이 필요로 할 것이고 장애물과 다른 물체를 구별하여 경로계획이 가능하여야 할 것이다. 물론 이와같이 주행에 필요한 비전시스템 모듈들이 모두 통합되었을때 차량을 직접 제작하여 하드웨어적으로 실행(implementation)되어야함은 물론이다.

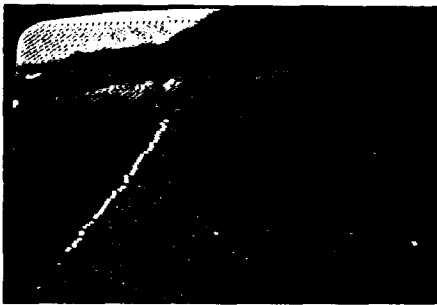


그림3.6 불확실한 도로경계

Fig3.6 An ambiguous road boundary

[WLD87] A. M. Waxman, J. LeMoigne, L. S. Davis, B. Srinivasan, T.Kushner, E. Liang and T. Sissalingaiah, "A visual navigation system for autonomous land vehicles," IEEE J. Robotics Automat., vol. RA-3, no. 2, pp. 124-141, 1987.

[BrH89] D. Brzakovic, L. Hong, "Road edge detection for mobile robot navigation," IEEE International Conference on Robotics and Automat., pp. 1143-1147, 1989.

[KPH88] D. Kuan, G. Phipps, and A.C. Hsueh, "Autonomous robotic vehicle road following," IEEE Trans. on PAMI, vol 10, no. 5, pp. 648-654, 1988.

[IM084] H. Itho, A. Miyauchi, S. Ozawa, "Distance measure method only simple vision constructed for moving robots," International Conference on Robotics, pp. 192-195, 1984.

[KIT89] K. Kluge, C. Thorpe, "Explicit models for robot road following," IEEE Conference on Robotics and Automat., pp. 1148-1154, 1989.

[SZM90] K. Storjohann, T. Zielke, H.A. Mallot and W. von Seelen, "Visual obstacle detection for automatically guided vehicles," International Conference on Robotics and Automat., pp. 761-766, 1990.

[TMG88] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "VITS - A vision system for autonomous land vehicle navigation," IEEE Trans. on PAMI, vol. 10, no. 3, pp. 342-361, 1988.

[PDM87] M. Podlaseck, B. Douglass, J. McSwain, R. Knourek, M. Thomas, "Vision-based road following in the autonomous land vehicle," Proc. of the Conference on Decision and Control, pp. 1814-1819, 1987.

[Dun88] R.T. Dunlay, "Obstacle avoidance perception processing for the autonomous land vehicle," IEEE International Conference on Robotics Automat. pp. 912-917, 1988.

[IK088] S. Ishikawa, H. Kuwamoto, and S. Ozawa, "Visual navigation of an autonomous vehicle using white line recognition," IEEE Trans. on PAMI, vol. 10, no. 5, pp. 743-749, 1988.

[DiD90] Sven J. Dickinson and Larry S. Davis, "A flexible tool for prototyping ALV road following algorithms," IEEE J. Robotics Automat., vol. 6, no. 2, pp. 232-242, 1990.

[LiJ87] S.-P. Liou and R. C. Jain, "Road following using vanishing points," Comput. Vision Graph. Image Process., vol 39, pp. 116-130, 1987.