

Stereo Image Processing Algorithm for Preceding Vehicle Detection Based on DLI

潢熙井* · 白光烈** · 李雲根***

(Hee-Jung Hwang · Kwang-Ryul Baek · Un-Kun Yi)

Abstract - This paper proposes an image processing algorithm for detecting obstacles on road using DLI(disparity of lane-related information) that is generated by stereo images acquired from dual cameras mounted on a moving vehicle. The DLI is a disparity that is acquired using a single lane information from road lane detection. For the purpose to reduce processing time, we use small block of edge-histogram based blocking logic. This algorithm detects moving objects such as preceding vehicles and obstacles. The proposed algorithm has been implemented in a personal computer with the road image data of a typical highway. We successfully performed experiments under a wide variety of road conditions without changing parameter values or adding human intervention. Experimental results also showed that the proposed DLI is quite successful.

Key Words : DLI (disparity of lane-related information), Image processing, Stereo matching, Obstacle detection

1. 서론

최근, 차량 운전에서 발생할 수 있는 교통사고를 예방하기 위한 안전 이슈와 운전 중에 기기를 사용할 때 발생할 수 있는 운전자의 집중력 손실 등과 관련된 연구가 활발히 진행되고 있다. 이러한 새로운 변화는 운전자와 탑승자에게 정보 이용의 이동성을 강화시켜 줄 것으로 예상되는 텔레매틱스를 진보시킬 것이다. 본 논문에서 제안하는 차선변이 함수 기반의 선행차량 인식 알고리즘은 차선이탈 경고 및 추돌 경고시스템 등에 이용될 수 있는데, 이러한 시스템은 텔레매틱스 시장에 있어서 핵심적인 부분으로 떠오르고 있다.

본 논문에서 제안하는 선행차량 인식 기법은 스테레오 비전에 기반을 하고 있다. 그동안 학계에서 스테레오 영상처리를 이용한 3차원 복원에 관련된 연구가 활발히 진행되어 왔지만 자동차의 능동형 안전기술에 적용한 사례는 매우 드물다. Saneyoshi 등[1]-[4]은 하프변환(Hough transform)과 절대 거리(Absolute distance)를 이용하여 수직 방향으로 일정한 크기의 윈도우를 블록화하고, 이것의 히스토그램 정보를 이용하여 차선 및 장애물을 인식하였다. Luong 등[5], [6]은 헬름홀츠체어(Helmholtz shear)에 기반하여 도로 위의 모든 물체를 장애물로 가정하고, 차선과 장애물을 인식하였

다. 그러나, 이러한 방법들은 스테레오 영상에서 많은 시간이 소요되는 대응점 탐색에 있어서, 영상 전체를 탐색 공간으로 하여 많은 계산 시간이 소요되는 단점을 가지고 있다.

본 논문에서는 카메라에서 얻어진 스테레오 영상에서 차선 관련 정보의 변이도 함수(DLI : Disparity of Lane-related Information)를 추출한다. DLI는 인식된 좌우 차선 중에서 한쪽 차선의 정보만을 이용하여 얻은 변이도 함수이다. 스테레오 영상의 대응점을 탐색할 때, DLI를 이용하는 방법은 특징점의 탐색공간을 현저히 줄여 실시간 처리문제를 해결할 수 있는 장점을 가지고 있다. DLI를 이용하여 선행차량을 인식하는 방법은 스테레오 영상을 사용하기 때문에 자차량과 선행차량 사이의 거리를 알 수 있다. 무인 자율 주행의 실현을 위한 필수 요소인 차선과 차량인식, 거리추출이 세 가지 문제는 서로 다른 보완성을 가지고 있는데, 본 논문과 같이 이 세 문제 각각에서 얻어진 정보를 상호 활용하면, 독립적으로 문제를 해결하는 것보다 정확도를 높일 수 있을 것이다. 예를 들면, 차선인식 정도의 보장은 주행차선을 정확히 식별할 수 있도록 하여, 자차선 내에서 주행하는 선행 차량의 식별 능력을 향상시킬 수 있다.

본 논문에서는 DLI를 이용한 선행차량 인식기법의 성능을 검증하기 위하여 실제 도로에서 발생할 수 있는 다양한 환경의 도로영상에 알고리즘을 적용하였다. 그리고 실험 결과를 통하여 DLI 함수와 선행차량 인식 사이에 높은 상관성이 있음을 밝히고, 또한 알고리즘의 유효성을 입증하였다.

2. 전체 알고리즘의 개요

본 논문에서는 용어의 사용을 명확히 하기 위하여 한국의 도로구조 및 시설구조에 관한 규정을 따른다. 이 규정의 의

* 正 會 員 : 釜山大 工大 電子工學科 博士課程

** 正 會 員 : 釜山大 工大 電子工學科 副教授 · 工博

*** 正 會 員 : 釜山經商大學 디지털애니메이션과 轉任講師 · 工博

接受日字 : 2003年 12月 26日

最終完了 : 2004年 5月 5日

하면 도로의 노면에서 차량이 주행하는 곳을 차선이라 하고, 차선을 구별하기 위해 도색된 차선표식이 차선경계를 이룬다고 되어 있다[7].

제안된 알고리즘은 차선 표식에 대하여 다음과 같은 두 가지 가정에 기본을 두고 있다. 첫째, 차선표식은 배경에 비해 밝은 색으로 도색되어 있다는 것이다. 둘째, 차선의 방향은 급격히 변하지 않고 연속성을 갖는다는 것이다. 도로상에서 차선 외에 어떠한 표식이나 잡음도 이러한 두 가지 사실을 동시에 지속적으로 만족시키는 경우는 드물다[8].

제안된 알고리즘에서 인식하려는 장애물은 주행 차선의 전방에 있는 물체를 뜻한다. 본 논문에서는 장애물 인식을 위해, 장애물의 방향과 위치는 급격히 변하지 않고 연속성을 갖는다는 가정을 한다.

본 논문에서 사용되는 스테레오 영상은 에피폴라 제약조건을 만족한다. 3차원상의 한 점에서 두 카메라 각각의 영상 평면상에서의 투영이 두 영상의 똑같은 y좌표로 되도록 두 카메라의 광축을 설정한다. 기준 영상에서의 한점에 대한 일치점을 찾기 위해서 상대 영상의 전 영역을 검색하는 것이 아니라 에피폴라 선상에서만 일치점을 찾겠다는 것이 에피폴라 제약 조건이다.

그림 2는 본 논문에서 제안하는 전체 알고리즘의 구성을 나타낸다. 제안하는 알고리즘은 먼저 스테레오 영상의 기준 영상에서 주행차선을 인식한 뒤에, 이 정보를 이용하여 DLI를 구축한다. DLI가 검출되면 이를 이용하여 선행 차량 인식을 수행한다.

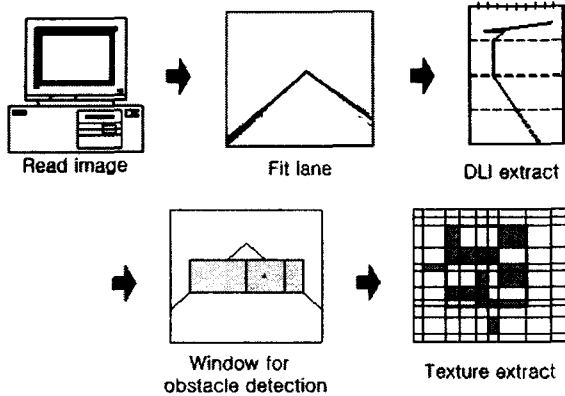


그림 2 전체 알고리즘의 개요
Fig. 2 Organization of the algorithm

3. 차선변이 함수(DLI)

스테레오 영상의 변이도는 좌우 영상 전체의 정보를 이용하여 구하는 것이 일반적인 방법이다. 본 논문에서는 영상 전체의 정보를 이용하지 않고, 차선 인식에서 얻은 차선 관련 정보 가운데 좌우 차선정보 중 한쪽 차선정보의 정보만을 이용하여 변이도를 구한다. 본 논문에서는 영상의 차선 관련 정보의 일부분을 이용해서 얻은 변이도를 DLI (Disparity of Lane-related Information)라고 정의한다. DLI는 인식된 좌우 차선정보 중에서 한쪽 차선정보의 정보만을 이용하여 얻기 때문에 변이도 검출을 위해 많은 양의 정보를 필요로 하지 않고, 영상 전체의 극히 일부분에 대한 정보

만이 요구되어진다. DLI는 선행 차량과 같은 장애물이 주위보다 상대적으로 큰 에지값을 가진다는 특성을 이용해서 주행차선 내에 있는 장애물의 유무 검출뿐만 아니라, 장애물의 위치도 유추할 수 있다.

스테레오 정합은 좌우 영상 중 어떤 영상을 기준으로 하는지에 따라 각기 다른 폐색 영역을 가진다. 제시한 알고리즘에서는 스테레오 영상의 좌측 영상을 중심으로 변이도를 구할 때에는 우측 차선을 중심 차선으로 정하고, 스테레오 영상의 우측 영상을 중심으로 변이도를 구할 때에는 좌측 차선을 중심 차선으로 정한다.

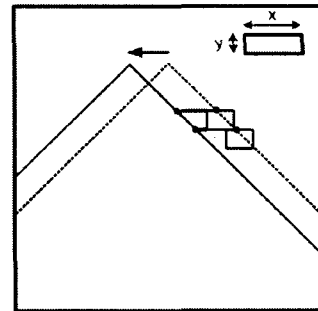


그림 3.1 DLI 알고리즘
Fig. 3.1 DLI algorithm

기준 영상을 정하고 기준 차선이 정해지면, DLI를 얻기 위하여 윈도우를 설정한다. 기준 차선을 중심으로 크기가 $x \times y$ 인 윈도우를 그림 3.1과 같이 설정한다. 그림 3.1에서 직선은 스테레오 영상의 우측 영상에 있는 차선정보이고, 점선은 스테레오 영상의 좌측 영상에 있는 차선정보이다. 스테레오 영상의 좌측 영상을 기준 영상으로 정합을 할 때, 그림 3.1과 같이 실선과 일치하는 점선 영역을 찾기 위해서는 좌측 영상에서 설정한 윈도우를 우측 영상에서 일정 간격만큼 좌측 방향으로 이동하여 정합을 이루는 우측 영상에서의 위치를 구한다.

식 (1)은 좌우 영상의 에지 크기를 비교하는데 사용되는 절대 거리(Absolute distance)이다.

$$C_n = \sum |L_{ni} - R_i| \quad (1)$$

여기서 C_n 은 n 번째 윈도우에서의 절대 거리를 나타낸다. 좌측 영상이 기준 영상일 경우, L_{ni} 는 좌측 영상의 기준 차선에 썩어진 윈도우의 i 번째에 있는 화소의 에지 크기이고, R_i 는 우측 영상에 있는 n 번째 윈도우의 i 번째에 있는 화소의 에지 크기이다. m 개의 윈도우 비교 값 중에서 가장 작은 값을 가지는 C_n 을 찾아서 그때의 n 값을 저장한다. 이 n 값이 바로 그 좌표에서의 변이도이고, 식 (2)에 있는 DLI 값이 된다.

$$DLI = \text{Min} (C_n), (1 \leq n \leq m) \quad (2)$$

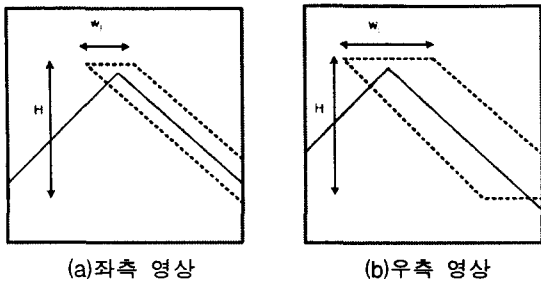


그림 3.2 DLI의 탐색공간
Fig. 3.2 Detection space of DLI

스테레오 영상의 좌측 영상을 기준 영상으로 할 때에는 그림 3.2의 (a)와 같이 우측 차선이 변이도를 구하는데 사용되는 기준 차선이 된다. 영상의 변이도를 구할 때, 좌측 영상에서는 그림 3.2 (a)의 점선 블록의 에지 정보가 필요하고, 우측 영상에서는 그림 3.2 (b)의 점선 블록의 에지 정보가 필요하다. 그림 3.2 (a)의 점선 블록은 기준차선에 찍어진 원도우들이 있는 영역이고, 그림 3.2 (b)의 점선 블록은 변이도를 구하기 위해 좌측 영상에서 이용되어지는 영역이다. 그림 3.2와 같이 DLI를 이용하여 변이도를 구하면, 특징점의 탐색 공간이 현저히 줄어들기 때문에 처리시간이 단축되고 이용되는 정보량이 줄어들어 실시간 처리 문제를 해결할 수 있는 장점이 있다.

그림 3.3은 도로에서 입력 받은 영상을 이용하여 DLI를 구축한 예이다. 그림 3.3에서는 DLI를 크게 네 영역으로 구분하고 있다. DLI 그래프에서 수평축은 변이도를 나타내고, 수직축은 입력 영상의 수직축 좌표와 일치 한다. 변이도는 식 (2)에서 구한 DLI 값이고, 수직축 좌표는 DLI에 사용된 입력 영상의 수직축 좌표이다. DLI는 일정한 기울기를 가지는 영역, 일정한 값을 유지하는 영역, 폐색영역, 소실점 영역으로 나누어진다. 변이도는 가까운 거리에 있는 물체일수록 큰 값을 가지고, 먼 거리에 있는 물체일수록 작은 값을 가진다.

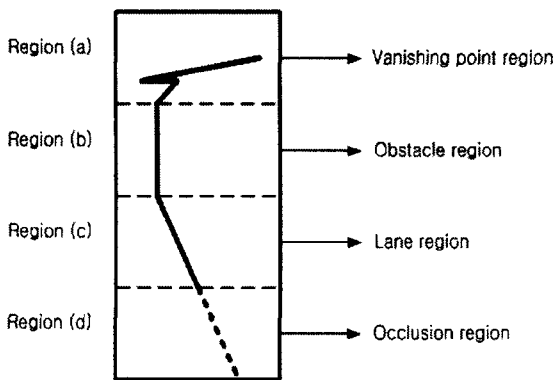


그림 3.3 DLI의 영역 구분도
Fig. 3.3 Division of the DLI graph

그림 3.3의 영역 (c)는 DLI 그래프에서 일정한 비율로 변이도가 커지는 영역으로 주행 차선이 있는 영역이다. 주행 차량과 가까운 거리에 있는 차선의 변이도는 큰 값을 갖고, 주행 차량과 먼 거리에 있는 차선은 변이도가 작은 값을 갖

는다. 주행하는 도로에 연속적으로 그려져 있는 차선은 거리에 따라 변이도가 일정하게 변한다. 그림 3.3을 살펴보면 차선이 있는 영역의 DLI는 일정한 기울기를 가지는 1차 직선의 형태로 나타나는 것을 볼 수 있다.

그림 3.3 영역 (b)는 DLI 그래프에서 변이도가 일정한 값을 유지하는 영역으로 주행 차선 내에 선행 차량과 같은 장애물이 있는 것을 뜻한다. 주행하는 차선에 선행 차량이 있는 경우, 입력 영상에서 이 선행 차량은 차의 뒷부분을 중심으로 영상에 나타나게 된다. 주행 차량에서 선행 차량의 뒷부분 내에 있는 영역까지의 거리는 같다. 그리고 선행 차량이 있는 곳에서는 에지 크기가 상대적으로 크게 나타나기 때문에 이 영역을 중심으로 DLI가 구해진다. DLI를 구하면 선행 차량이 있는 영역에서는 그림 3.3의 영역 (b)에 표현된 것과 같이 변이도가 일정한 값을 유지한다. 변이도 값이 같다는 것은 같은 거리에 있다는 것을 뜻한다. 선행 차량의 뒷부분은 주행 차량과 일정한 거리에 있기 때문에 그 영역이 나타난 곳의 변이도는 같은 값을 가지게 되고, 그 결과 DLI는 수직축과 평행한 직선의 형태로 나타난다.

그림 3.3의 영역 (d)는 스테레오 영상의 폐색영역이 나타나는 곳으로 정확한 변이도를 알 수 없는 영역이다. 스테레오 영역에서 발생하는 폐색 영역은 입력 영상의 크기를 조절함에 따라 줄일 수 있다. 입력 영상의 가로 길이가 세로 길이 보다 긴 직사각형 영상을 사용하는 것이 정사각형이나 가로 길이 보다 세로 길이가 긴 직사각형을 사용할 때보다 폐색영역이 줄어든다. 그리고 차선이 있는 영역에서 나타나는 1차 직선 형태의 변이도를 이용해서 그림 3.3의 영역 (d)에 있는 점선과 같이 폐색영역에서의 변이도를 유추할 수 있다.

그림 3.3의 영역 (a)는 DLI 그래프의 가장 위쪽에 있는 영역으로, 변이도가 불안정한 변화율을 가지는 것을 볼 수 있다. 이렇게 변이도가 불안정한 변화율을 가지는 영역은 소실점 부근의 영역으로 정확한 변이도 값을 얻을 수 없고, 신뢰성 있는 변이도를 구하기 힘든 부분이다. 소실점 이상의 영역은 주행차량과의 거리차가 큰 영역으로 본 논문이 다루는 차선과 장애물의 관심 범위를 벗어난다. 본 논문에서는 소실점 부근과 소실점 이상의 영역은 본 논문에서 다루는 차선 인식과 장애물 인식에 불필요한 영역으로 판단하고 이 영역의 변이도를 구하지 않는다.

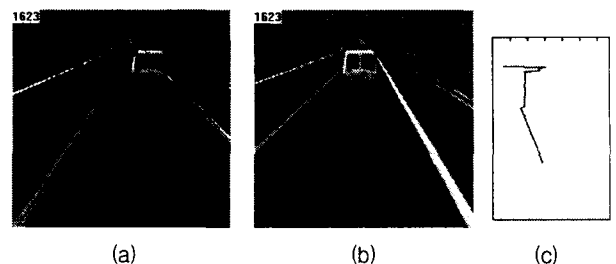


그림 3.4 스테레오 영상과 DLI
Fig. 3.4 Stereo images and DLI

그림 3.4의 (a)는 스테레오 영상의 좌측 영상이고, 그림 3.4 (b)는 스테레오 영상의 우측 영상이다. 그림 3.4 (c)는 그림 3.4의 (a)에서 구한 차선정보 중에 우측 차선정보를 중

심으로 얻은 DLI 그래프이다. 그림 3.4의 DLI 그래프를 보면 영상에서 주행차선 내에 있는 선행차량이 있는 영역에서는 변이도가 일정한 값을 유지하고, 주행차선이 있는 영역에서는 일정한 기울기를 가지는 일차 직선의 형태로 변이도가 나타나는 것을 확인 할 수 있다.

4. 선행차량 인식

4.1 탐색 윈도우 설정

본 논문에서는 DLI 정보를 이용해서 주행 차선 내에 있는 선행차량을 인식한다. DLI는 차선 내에 선행차량이 있을 가능성이 큰 영역을 설정하는데 중요한 역할을 한다. DLI 그래프를 보면 변이도가 일정한 값을 유지하는 영역이 있는데, 본 논문에서는 이 영역을 물체가 있을 가능성이 큰 영역으로 인식한다. 그리고 변이도가 일정한 값을 유지하는 영역 내에 있는 차선과 이 영역이 만나는 곳을 중심으로 탐색 윈도우를 설정한다.

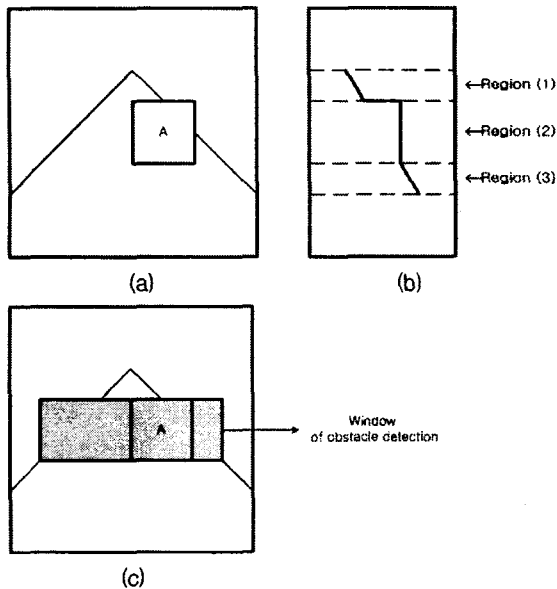


그림 4.1 탐색 윈도우 설정
Fig. 4.1 Selection of the window for obstacle detection

그림 4.1 (a)에 있는 'A'는 주행 차선 전방에 있는 선행차량이고, 그림 4.1 (b)는 이 영상의 DLI 그래프이다. 그림 4.1 (b)의 DLI 그래프를 보면, 'A'가 있는 곳인 영역 (2)에서는 변이도 값이 일정하게 유지되고, 차선이 있는 영역 (1)과 영역 (3)에서의 변이도는 기울기가 일정한 일차 직선으로 나타나는 것을 알 수 있다. 탐색 윈도우는 선행차량이 있을 확률이 높다고 판단되는 영역 (2)와 주행 차선이 만나는 곳을 중심으로 그림 4.1 (c)와 같이 설정 된다.

그림 4.2는 DLI를 이용한 탐색 윈도우 설정을 보여준다. 그림 4.2 (a)에는 선행차량을 중심으로 블록이 설정되어 있다. 이 블록은 변이도가 일정하게 유지되는 영역과 주행 차선이 만나는 곳을 중심으로 설정된 탐색 윈도우이다. 그림 4.2 (a)를 보면, 설정되어진 탐색 윈도우 내부에 선행차량이 있는 것을 확인 할 수 있다. 그림 4.2 (c)는 주행 차선 내에

선행차량이 없는 영상으로, 이 영상의 DLI는 그림 4.2 (d)이다. 그림 4.2 (d)는 그림 4.2 (b)와는 달리 주행 차선내에 장애물이 없기 때문에 변이도가 일정하게 유지 되는 곳이 없다. 그림 4.2 (d)는 기울기가 일정한 일차 직선으로만 표현 되고, 그림 4.2 (c)에는 탐색 윈도우가 설정되지 않는다.

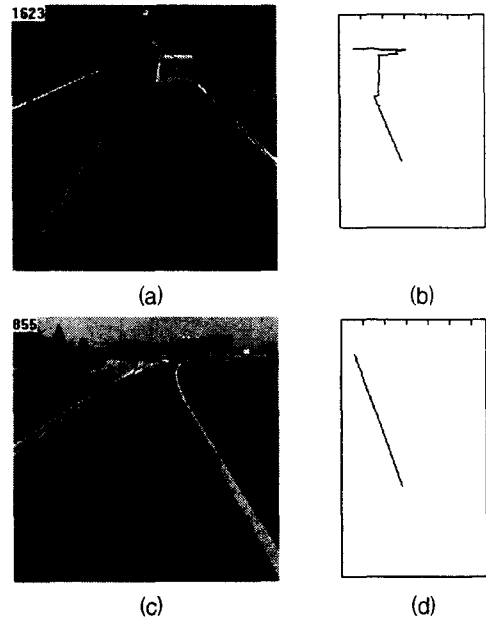


그림 4.2 장애물과 탐색 윈도우 설정
Fig. 4.2 Selection of searching window for obstacle detection

4.2 텍스처 추출

본 논문에서는 선행차량을 인식하기 위해서 탐색 윈도우를 중심으로 블록을 설정 한다. 블록을 설정하기 위해서는 수직 에지와 수평 에지가 필요한데, 본 논문에서는 소벨 연산자를 이용하였다. 식 (3)은 소벨 마스크를 이용하여 얻은 에지값으로, G_x 는 수직 마스크를 이용해서 얻은 에지값이고, G_y 는 수평 마스크를 이용해서 얻은 에지값이다.

$$G_x = |(a_{13} + 2a_{23} + a_{33}) - (a_{11} + 2a_{21} + a_{31})|$$

$$G_y = |(a_{11} + 2a_{12} + a_{13}) - (a_{31} + 2a_{32} + a_{33})| \tag{3}$$

그림 4.3에 있는 X, Y 는 탐색 윈도우의 수평 길이와 수직 길이이고, N 은 X, Y 를 각각 등간격으로 나눌 때의 기준 간격이 된다. 블록 설정을 위해서 G_x 와 G_y 를 이용하여 각 축에 대한 히스토그램을 구하고, N 간격에서 가장 큰 히스토그램을 가지는 곳의 좌표를 검색 한다.

식 (4)에서 His_{G_x} 는 수직축에 대한 히스토그램이고, His_{G_y} 는 수평축에 대한 히스토그램이다. C_x 는 수직축을 중심으로 N 간격에서 가장 큰 히스토그램의 좌표를 저장한 함수고, C_y 는 수평축을 중심으로 N 간격에서 가장 큰 히스토그램의 좌표를 저장한 함수이다.

$$C_v = \text{MAX} \left(\sum_{j=1}^N \text{His}_G(j) \right)$$

$$C_h = \text{MAX} \left(\sum_{i=1}^N \text{His}_G(i) \right) \quad (4)$$

위 식을 이용하면 일정 간격 내에서 가장 큰 히스토그램 값을 가지는 곳의 수평 좌표와 수직 좌표를 구할 수 있다. 블록은 C_v 와 C_h 에 저장된 좌표값을 이용해서 그림 4.3과 같이 설정된다.

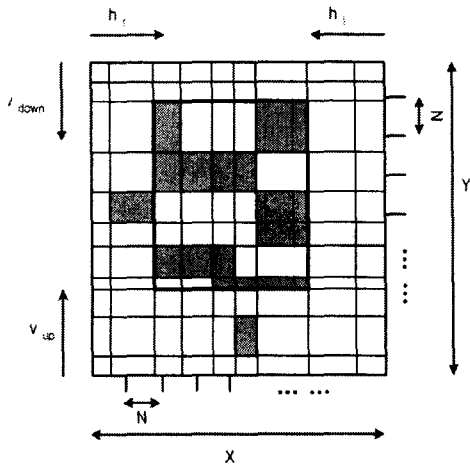


그림 4.3 텍스처 추출
Fig. 4.3 Extraction of texture for obstacle detection

블록이 설정되면 탐색 윈도우 내에 물체가 있는지를 구분해야 한다. 본 알고리즘에서는, 탐색 윈도우 안에 물체가 있는지를 구분하기 위해서 블록 설정에서 구한 수평 에지 G 를 이용한다.

$$T_w = E [G_w(\text{Window}(X, Y))]$$

$$T_b = E [G_b(\text{Block}(N, N))] \quad (5)$$

$$T_b > T_w \times \text{Weight} \quad (6)$$

식 (5)의 T_w 는 탐색 윈도우 안의 수평 에지 G 의 평균이고, T_b 는 탐색 윈도우 내에 있는 블록에 대한 수평 에지 G 의 평균이다. 본 논문에서는 식 (6)을 만족하는 블록을 물체가 있을 가능성이 큰 영역으로 판단한다. 탐색 윈도우 내에 있는 각 블록마다 식 (6)이 만족되는지를 검색하고, 식 (6)을 만족하는 블록이 있으면 그 블록의 위치를 저장한다.

텍스처는 식 (6)을 만족하는 블록을 중심으로 추출한다. 그림 4.3에 있는 검게 채워져 있는 블록은 탐색 윈도우 내에서 물체가 있을 가능성이 큰 영역을 표시한 것이다.

탐색 윈도우에서 텍스처 추출을 위해서 각 축마다 검게 채워진 블록의 수를 검사 한다. 수평축의 좌측 끝에서부터 h 방향으로 각 축마다 블록의 수를 계산한다. 2개 이상의 블록이 있는 축이 검색되면 그 축의 좌표를 저장하고 검색

을 마친다. 수평축의 우측 끝에서부터 h 방향으로 각 축마다 블록의 수를 얻는다. 2개 이상의 블록이 있는 축이 검색되면 그 축의 좌표를 저장하고 검색을 마친다. 수직축은 v_{up} 과 v_{down} 방향으로 수평축과 같은 방법으로 각각의 좌표를 구한다. 이렇게 얻은 4개의 좌표를 이용해서 사각 블록이 만들어지면 탐색 윈도우 내에 물체가 있는 것으로 판단하고, 그 블록을 중심으로 텍스처를 추출한다. 하지만 사각 블록이 형성되지 않으면 탐색 윈도우 내에 물체가 없는 것으로 판단하고 새로운 영상을 입력받는다.

$$S_{left} = W_l (= m) \times H \quad (7)$$

$$S_{right} = W_r (= x + m) \times H \quad (8)$$

$$S_{DLI} = S_{right} + S_{left} \quad (9)$$

$$S_{ALL} = X \times Y \times 2 \quad (10)$$

$$\text{DLI method} : (C(S_{window})) \times N_{shift} \times H \quad (11)$$

$$\text{Ordinary method} : (C(X)) \times X \times Y \quad (12)$$

식 (7)은 그림 3.2의 (a)에 설정되어 있는 블록의 크기이고, 식 (8)은 그림 3.2의 (b)에 설정되어 있는 블록의 크기이다. 식 (9)는 DLI 방식을 사용할 때 스테레오 영상에서 필요로 하는 화소 정보의 갯수이고, 식 (10)은 영상의 가로 길이가 X 이고 세로의 길이가 Y 일 때 스테레오 영상 전체의 크기이다. 식 (11)은 DLI 방식을 이용해서 변이도를 구하는 경우에 필요로 하는 시티 블록 거리의 계산량이다. N_{shift} 는 윈도우의 시프트 횟수고, S_{window} 는 기준 윈도우 1개의 크기이다. $C(s)$ 는 's' 크기의 영역에서 시티 블록 거리를 수행한다는 것이다. 식 (12)은 영상 전체를 이용해서 스테레오 영상의 변이도를 구할 때 요구되어지는 시티 블록 거리의 계산량이다.

예를 들어 그림 3.1과 그림 3.2에서 $x=30, y=4, m=80, X=360, Y=280, H=140$ 인 경우, $S_{right}=15400, S_{left}=4200, S_{DLI}=19600, S_{ALL}=201600$ 의 값을 가진다. 따라서 스테레오 영상의 변이도를 구할 때 DLI를 이용하면, 스테레오 영상의 약 9.7%의 정보가 필요하다. DLI를 이용해서 스테레오 영상의 변이도를 구하는 경우에 필요한 계산량은 $C(120) \times 11340$ 이다. 영상 전체를 이용해서 스테레오 영상의 변이도를 구하는 경우에는 $C(360) \times 100800$ 의 계산량이 필요하다. 따라서 스테레오 영상의 변이도를 구할 때 DLI를 이용하면, 영상 전체를 이용하는 경우 필요한 계산량의 약 3.75%가 필요하다.

본 논문에서 제안한 알고리즘은 스테레오 영상에서 변이도를 구할 때 DLI를 이용해서 특징점의 탐색 공간을 현저히 줄였다. 그 결과 DLI를 이용한 알고리즘은 메모리, 계산량 등을 향상시키고 실시간 처리 문제를 해결 할 수 있다. 게다가 제시한 알고리즘은 변이도를 구할 때 영상의 에지 크기 값을 비교하기 때문에 좌우 영상의 밝기 값을 비교하는 것보다 잡음에 강하고 더 정확한 비교 값을 얻을 수 있다.

그림 4.4의 (a)에는 선행차량 인식을 위한 탐색 윈도우 설정, 블록 설정, 텍스처 추출의 과정이 모두 나타나 있다. 그림 4.4 (b)는 DLI를 이용한 장애물 인식 알고리즘의 수행 결과로, 주행 차선에 있는 선행 차량이 텍스처로 인식된 것을 확인할 수 있다.

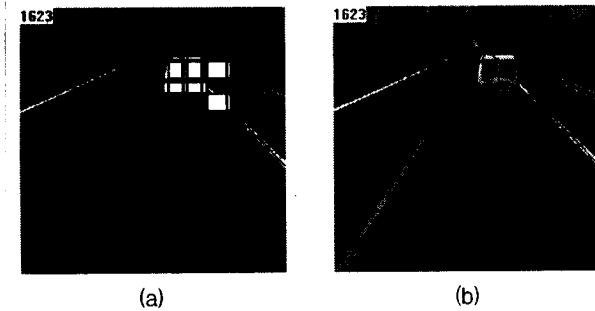


그림 4.4 선행차량 인식
Fig. 4.4 Extraction of preceding vehicle

5. 실험 및 결과 고찰

실험에서 사용한 영상은 고속도로에서 획득한 도로 영상으로, B/W CCD 카메라를 이용해서 얻은 스테레오 영상이다. 카메라로부터 획득한 영상의 크기는 728×288 픽셀 크기이며, 실험에 사용한 영상은 카메라로부터 획득된 영상의 중심을 기준으로 한 360×280, 256×256 픽셀 크기의 8비트 그레이 레벨로 처리된 영상이다. 알고리즘은 AMD Athlon XP 2400+ (2.00GHz)를 탑재한 컴퓨터에서 Microsoft Visual C++ 6.0을 이용하여 구현하였고, 실험결과, 초당 15프레임의 영상처리 결과를 나타내었다.

실험에 사용된 스테레오 영상은 좌측 영상을 기준영상으로 하였고, 좌측 영상의 우측 차선정보를 중심으로 DLI를 구축하였다.

제한한 알고리즘은 무인 자율 주행의 실현에 필요한 차선과 선행 차량과 같은 장애물을 인식하기 위한 것이다. 본 논문에서는 실제 도로에서 발생할 수 있는 다음과 같은 여러 상황에서 실험을 하였다.

본 논문에서는 직선도로 위에 선행 차량이 있는 경우, 곡선도로 위에 선행차량이 있는 경우, 도로에 그림자가 있는 경우, 직선 도로의 소실점 부근에 선행 차량이 있는 경우, 곡선 도로의 소실점 이상까지 선행차량이 걸쳐져 있는 경우, 옆 차선에서 주행하는 차량이 현재 차차량이 주행하고 있는 차선으로 차선을 변경하는 경우 등 다양한 도로 환경에서 제안된 알고리즘을 수행하였다.

그림 5.1은 주행 차선 내에 선행차량이 있는 영상이다. 그림 5.1의 (a)는 직선도로 위에 선행차량이 있는 영상이고, (b)는 곡선도로 위에 선행차량이 있는 영상이다. 차선의 굴곡과는 상관없이 주행 차선 내에 있는 선행차량이 정확하게 인식되는 것을 확인할 수 있다.

그림 5.2는 선행차량이 소실점 영역에 걸쳐져 있는 경우의 영상이다. 선행 차량은 소실점 위의 영역 까지 걸쳐져 있지만, 실험 결과를 보면 소실점에 걸쳐져 있는 선행차량의 영역이 정확하게 인식된다는 것을 확인할 수 있다.

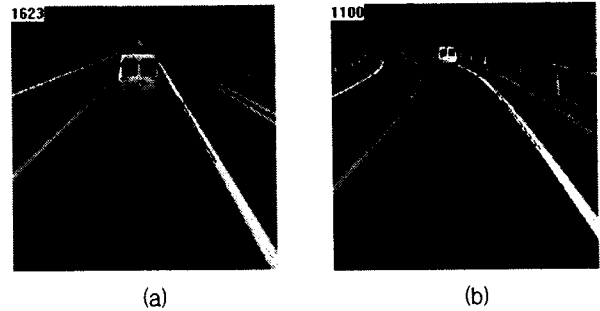


그림 5.1 주행차선 내에 있는 선행차량 인식
Fig. 5.1 Obstacle detection on a road (256×256 pixels)

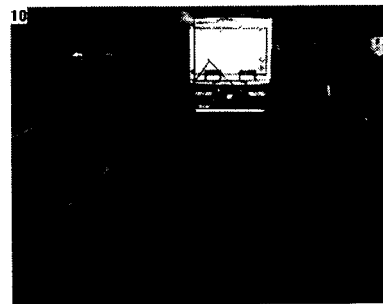


그림 5.2 소실점 영역에서의 선행차량 인식
Fig. 5.2 Obstacle detection of a vanishing point region (360×280 pixels)

그림 5.3은 주행하는 도로에 그림자가 있는 경우의 영상인데, 실험 결과를 보면 그림자를 선행차량으로 잘못 인식하지 않는다는 것을 확인할 수 있다. 주행하는 차선 옆 차선에 선행차량이 있는데, 실험 결과에 나타난 것과 같이 이 차량을 주행 차선 내에 있는 선행차량으로 인식 하지 않고, 정확하게 주행 차선만 인식하는 것을 확인할 수 있다.



그림 5.3 고가도로의 그림자가 있는 경우
Fig. 5.3 Shadow image of a flyover on the road (360×280 pixels)

그림 5.4는 주행하는 차선의 옆 차선에서 주행하던 선행 차량이 주행차선 쪽으로 차선을 변경하는 경우의 영상이다. 그림 5.4의 (a)를 보면 주행차선 내에 선행차량이 없는 것으로 인식 되지만 그림 5.4의 (b)~(h)를 보면, 옆 차선에서 주행하던 선행차량 차선을 변경하면서 점차 주행차선 내에 있는 선행 차량으로 인식되어 간다는 것을 확인할 수 있다.

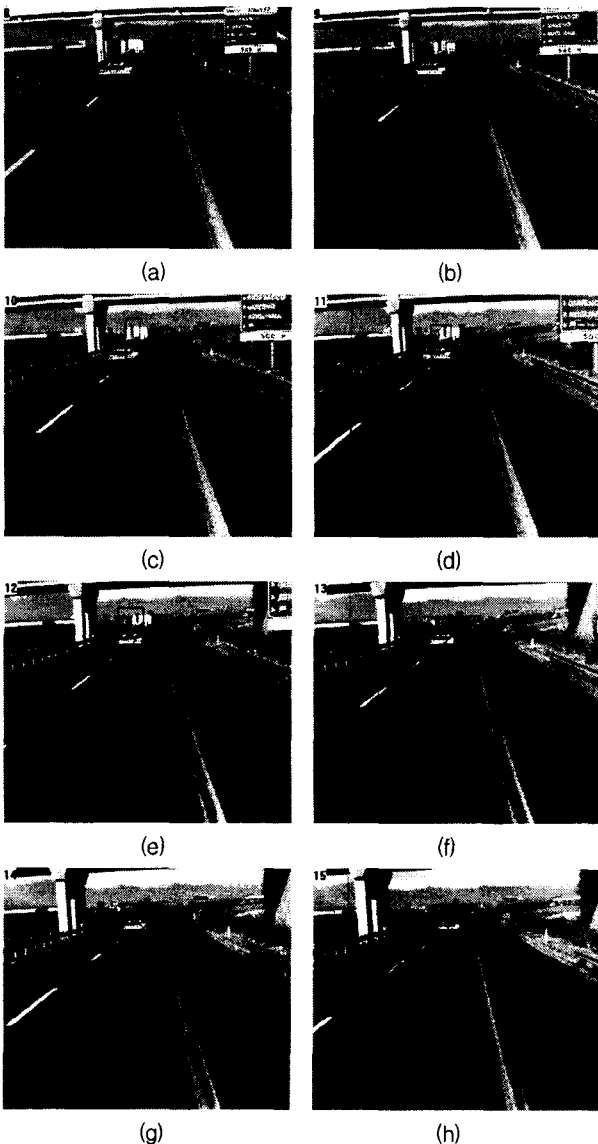


그림 5.4 차선을 변경하는 선행차량 인식
 Fig. 5.4 Preceding vehicle detection on lane changing
 (360×280 pixels)

이와 같이 실험 결과, 본 논문에서 제안한 알고리즘의 우수성과 신뢰성을 입증할 수 있었다.

6. 결 론

본 논문에서는 양안의 흑백 도로영상에서 주행차선 내에 있는 선행차량을 인식하기 위한 새로운 알고리즘을 제안하였다. 제안된 알고리즘은 인식되어진 차선관련 정보만을 이용하여 영상의 변이도를 추출할 수 있는 변이도함수(DLI)를 정의하였다. 그리고, DLI를 이용해서 주행차선 내에 있는 선행차량의 위치를 유추하고 인식하였다. 본 논문에서는 제안된 알고리즘의 신뢰성을 입증하기 위해서 도로 위의 그림자, 차선의 굴곡 및 굽기, 장애물의 크기와 위치 등과 관련된 매우 다양한 도로 조건하에서 알고리즘을 수행하였다. 그리고 휴리스틱한 파라미터, 가정 및 제약의 사용을 최소화

하여 파라미터 값의 변경이나 사람의 개입 없이 성공적으로 실험을 수행하였다.

본 논문에서 도출된 결과는 논문에서 본질적으로 목표를 두고 있는 응용기술로서 ITS 분야의 차선이탈경보 및 방지 시스템 그리고, AGV (autonomous guidance vehicle) 등의 이동로봇 분야에 응용될 수 있으며, 향후 본 논문의 결과를 적용하기 위한 정보활용 방안에 대한 연구가 진행되어야 한다. 그리고, 제안된 DLI 기반의 장애물 인식 알고리즘은 도로환경의 영상처리에 한정된 것이 아니라 각종 패턴분류, 인식시스템 등에 직접적인 응용이 가능하며, 신뢰성을 요구하는 진단시스템 등 타 분야로의 확장 적용이 가능할 것이다.

감사의 글

본 연구는 한국과학재단 지정 울산대학교 네트워크 기반 자동화연구센터의 지원에 의한 연구로서, 관계부처에 감사 드립니다.

참 고 문 헌

- [1] K. Saneyoshi, "Drive assist system using stereo image recognition," Intelligent Vehicles Symposium, Proceedings of the 1996 IEEE, pp. 230-235, 1996.
- [2] K. Saneyoshi, "3-D Image Recognition System by Means of Stereoscope Combined with Ordinary Image Processing," Intelligent Vehicles '94 Symposium, pp. 13-18, 1994.
- [3] K. Saneyoshi, K. Hanawa, K. Kise, Y. Sogawa, "3-D Image Recognition System For Drive Assist," Intelligent Vehicles '93 Symposium, pp. 60-65, 1993.
- [4] K. Hanawa and Y. Sogawa, "Development of Stereo Image Recognition System for ADA," IVS2001, pp. 177-182, 2001.
- [5] Q.-T. Luong, J. Weber, D. Koller, J. Malik, "An integrated stereo-based approach to automatic vehicle guidance," Fifth International Conference on Computer Vision, pp. 52-57, 1995.
- [6] J. Weber, D. Koller, Q.-T. Luong, J. Malik, "New results in stereo-based automatic vehicle guidance," Intelligent Vehicles '95 Symposium., pp. 530-535, 1995.
- [7] 건설부, 도로의 구조·시설구조에 관한 규정 해설 및 지침, 건설부, 1990.
- [8] J. W. Lee, U. K. Yi, and K. R. Baek, "A Cumulative Distribution Function of Edge Direction for Road-Lane Detection," IEICE Trans. Information and Systems, vol. E84-D, no. 9, pp. 1206-1216, 2001.

저 자 소 개



황희정 (潢熙井)

1976년 5월 12일생. 2001년 부산대 전자공학과 졸업. 2003년 동 대학원 전자공학과 졸업(석사). 2003년~현재 동대학원 전자공학과 박사과정. 2003년~현재 삼창기업(주) 제어기술연구소 연구원.

Tel : 052-260-7227

Fax : 052-260-7231

E-mail : hhj1313@samchang.com



백광렬 (白光烈)

1961년 3월 25일생. 1980년 부산대 전기기계공학과 졸업. 1986년 한국과학기술원 전기 및 전자공학과 졸업(석사). 1989년 한국과학기술원 전기 및 전자공학과 졸업(공학박). 1989년~1994년 (주)터보테크 기술연구소 H/W개발실장. 1994년~1996년 부산대 전자공학과 전임강사. 1996년~2000년 부산대 전자공학과 조교수. 2000년~현재 부산대 전자공학과 부교수. 1994년~현재 부산대학교 컴퓨터 및 정보통신 연구소 연구원.

Tel : 051-510-2460

Fax : 051-515-5190

E-mail : krbaek@pusan.ac.kr



이운근 (李雲根)

1969년 3월 12일생. 1991년 울산대 전기공학과 졸업. 1993년 동 대학원 전기공학과 졸업(석사). 2002년 부산대학교 전자공학과 졸업(공학박). 1992년~1998년 동환산업(주) 연구소 주임연구원. 1999년~2003년 삼창기업(주) 제어기술연구소 선임연구원. 2003년~현재 부산경상대학 디지털애니메이션과 전임강사.

Tel : 051-850-1282

Fax : 051-862-7577

E-mail : yiuk@bsks.ac.kr