

論文2000-37SP-5-6

최적의 Moving Window를 사용한 실시간 차선 및 장애물 감지 (Detection of a Lane and Obstacles in Real Time Using Optimal Moving Windows)

崔勝郁*, 李章明*

(Sung Yug Choi and Jang Myung Lee)

요 약

본 논문에서는 주행차량에 장착된 CCD 카메라를 통하여 획득되어진 영상으로부터 moving window를 사용하여 차선을 인식하고 장애물을 감지하는 방법을 제안한다. 입력되는 동영상을 실시간에 처리하기 위해서는 하드웨어적으로 상당히 많은 제약을 초래한다. 이러한 문제점을 극복하고 영상을 사용하여 실시간에 차선 인식 및 장애물을 감지하기 위해, 도로조건과 차량상태에 바탕을 둔 최적의 window 크기를 결정하고, 그 window 영상만을 처리하여 차선 인식 및 장애물 감지를 실시간에 가능하게 하는 기법을 제안한다. 영상의 각 프레임에 대하여 moving window는 칼만필터에 의해 정확성이 향상된 예측방향으로 옮겨진다. 제안된 알고리즘의 효율성을 고속도로 주행영상을 사용한 실험을 통하여 보여준다.

Abstract

A moving window technique for detecting a lane and obstacles using the images captured by a CCD camera attached in an automobile, is proposed in this paper. To process the dynamic images in real time, there could be many constraints on the hardware. To overcome these hardware constraints and to detect the lane and obstacles in real time, the optimal size of window is determined based upon road conditions and automobile states. By utilizing the sub-images inside the windows, detection of the lane and obstacles become possible in real time. For each image frame, the moving windows are re-determined following the predicted directions based on Kalman filtering theory to improve detection accuracy, as well as efficiency. The feasibility of proposed algorithm is demonstrated through the simulated experiments of highway driving.

I. 서 론

최근 선진각국에서는 국가적인 차원에서 지능형 교통시스템(ITS) 구축을 국민복지 향상을 위한 교통서비

스 제공에 목표를 두고 추진하고 있다^[14,20]. 이에 우리나라에서는 ITS 코리아를 두어, 지능형 교통시스템 구축에 박차를 가하고 있다. 이의 한 분야인 첨단차량시스템(AVS : Advanced Vehicle System)은 미국의 AVCS(Advanced Vehicle Control System), 유럽의 PROMETHEUS, 일본의 ASV(Advanced Safety Vehicle)와 함께 운전자의 주행 중 편의성 및 조작성을 향상시켜 안정성을 극대화하고 나아가 자율주행을 목표로 하고 있다^[1,2,19,20].

현대의 자율주행 시스템은 주행상황에서 인간과 차량, 도로환경의 개별 소재가 각기 별개가 아닌 유기적으로 결합된 하나라는 개념을 바탕으로 하고 있다

* 正會員, 釜山大學校 電子工學科
(Dept. of Electronics Engineering Pusan National University)

※ 본 연구는 1998년 부산대학교 “학술연구조성비” 지원에 의해 연구되었음

接受日字 : 1999年10月21日, 수정완료일 : 2000年4月3日

[15,19,20]. 이때, 차량과 인간과의 관계는 자동 parking과 같은 운전지원 시스템과 자동화 및 안전성확보가 우선되는 주행지원 시스템으로 구분된다. 이러한 주행지원 시스템은 운전자를 중심으로 볼 때, 인간이 주어진 상황에 적절히 대처하지 못하는 경우에 자동적으로 대처하거나, 혹은 대처할 수 있는 시간을 확보해주는 적극적인 개념의 능동시스템^[3,16]과 주행상황을 전체 지능형 교통시스템의 관제 하에 자율적으로 통제하는 차량주행 등의 수동시스템으로 나누어진다. 본 연구에서 제안하는 영상정보를 이용한 차선 및 장애물 검출은 전자인 능동시스템에 해당되어지며, 따라서 운전자에게 주어진 상황을 전달하여 주행의 안전을 보조할 수 있다. 현재, 주행 중에 있는 운전자에게 안전확보를 위한 정보를 획득하는 방법으로 CCD 카메라뿐만 아니라, 초음파와 레이더 및 레이저 등의 다양한 센서가 함께 사용되고 있다^[3,5-8,15-17]. 이를 사용하여 입력되는 각종정보 중에서 영상정보는 정보량이 크고, 외부환경 변화에 민감하여 정보추출의 어려움이 따름에도 불구하고, 포괄적인 정보를 포함하고 있어서 활용의 범위가 넓다. 이와 같은 이유에서 주행 상황에서 영상 정보사용의 필요성은 많이 연구되고 있다^[4-8,15-17].

영상 정보처리를 하기 위해 본 논문에서는 먼저, 256 gray level로 표시되는 영상입력에서 moving window 내부의 영상만을 사용하여 에지(edge)를 검출한 후 중심모멘트(central moment)를 구하고, 구해진 중심모멘트에 대하여 이전 시점의 정보를 고려하여 곡선근사(curve fitting)를 한 후, 이를 근거(check point)로 다음 시점에서 moving window의 위치를 칼만필터(Kalman filter)로 예측한다.

앞서 제시된 영상정보 처리는 moving window를 기반으로 이루어지며, 본 논문에서는 실제 도로설계 조건을 모델로 하여 탐색 영역인 moving window의 크기를 결정하고, 상황에 따라 실시간에 위치를 조절하는 기법을 정립한다.

II. 영상처리

1. 에지검출(Edge detection)

다양한 입력영상의 상황으로부터 특정 밝기값만을 차선으로 추출하는 것이 중요하다. 이를 하드웨어적인 방법으로 구현할 수 있지만, 고속의 처리 속도에도 불구하고, 입력영상에 따른 임계값(threshold) 결정이 필

요하다^[5]. 하지만 매번 샘플 시점마다 빛 변화가 다양한 도로 상황에 대하여, 고정되어진 값으로 임계값을 결정하는 방법(static threshold)은 입력되는 영상정보의 전반에 걸쳐 정보의 손실을 초래한다. 따라서, 전체 입력영상에 대하여 빛 변화 정도에 따라 부분적으로 서로 다른 임계값을 지정하는 방법(dynamic threshold)을 구현해야 하며, 또한 임계값이 입력영상에 따라 자동으로 조절되는 방법을 구현해야한다. 이러한 자동 임계값 조절방법은 다양한 임계값 선정 조건에 따른 연산의 혼잡으로 처리속도를 저하시키며, 그 결과 고속으로 주행하는 도로환경에 적용하기에 부적절하다.

그러므로, 고속으로 차선 및 도로정보를 추출하는 방법으로 영상입력에 따라 임계값을 연산하는 방법보다, 화소(pixel)의 밝기나 색의 변화로 도로경계를 나타낼 수 있는 에지검출 방법이 타당한 것으로 판단된다. 입력영상에서 에지는 물체의 테두리나 물체와 물체간의 경계에 해당하므로, 검출되어진 에지는 차선정보 및 비포장 도로에서의 도로 경계정보 그리고, 주행방향 전방의 차량 및 장애물에 해당하는 정보 등을 모두 포함하게 된다.

에지 부분을 $I(x, y)$ 로 할 때, 그레디언트 에지(gradient edge)는 임의의 수평선을 따라 식 (1)과 같이 $G(x, y)$ 의 형태로 나타낼 수 있다.

$$G(x, y) = \frac{\partial I(x, y)}{\partial x} \vec{x} + \frac{\partial I(x, y)}{\partial y} \vec{y} \quad (1)$$

일반적으로 주행환경의 입력영상에서 도로정보는 수직성분으로 나타나므로, 이를 검출하기 위해 수직성분 검출 마스크를 사용한다. 또한, 입력영상에서 노이즈가 많은 경우에는 잘못된 에지점들이 검출된다. 이를 개선하기 위해 본 연구에서는 식 (2)와 같이 이웃 화소수를 더 많이 고려한, 확장된 크기의 프레위트 5×5 연산자를 사용한다^[13].

$$H_x = \frac{1}{10} \begin{bmatrix} 1 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 \\ 1 & 1 & 0 & -1 & -1 \end{bmatrix} \quad (2)$$

2. 중심모멘트

에지검출에 의해 구해진 정보는 불규칙하게 분포하고 있으므로, 2차원의 모멘트를 이용해 스칼라 양으로 표시하여, 차선 위치에 대한 특성을 간략히 할 수 있다.

먼저, 에지검출된 영상을 식 (3)에 의해 2진화하여 $f_B(x, y)$ 로 나타낸다.

$$f_B(x, y) = \begin{cases} 1, & f(x, y) \geq Th \text{ 일때} \\ 0, & \text{그외의 경우} \end{cases} \quad (Th = 80) \quad (3)$$

2진화 영상 $f_B(x, y)$ 의 1차 모멘트인 x_G, y_G 를 사용하여, 중심모멘트를 식 (4)와 같이 나타낼 수 있다.

$$M_{pq} = \sum_x \sum_y (x - x_G)^p (y - y_G)^q f_B(x, y) \quad (4)$$

차선의 위치를 간략히 나타내는 중심모멘트의 위치는 III장에서 소개되어질 곡선근사와 칼만필터를 통하여 sub-block의 위치를 움직이는 중요한 단서가 된다.

3. 최적의 sub-block 크기

연속되는 입력영상은 매 시점마다 640×480 [pixel]의 크기를 가지며, 필요한 정보를 추출하기 위해 전체 영상에 대하여 전처리를 실시하는 것은 불필요한 연산의 증가를 가져온다. 따라서 필요한 부분만을 탐색영역인 sub-block으로 지정하여 연산 시간을 감소하려는 연구가 진행되고 있다^[5, 6]. 주행상황에서 필요한 정보를 차선이라 할 때, 입력영상에서 차선이 위치한 부분은 도로의 곡률에 따라 변화되어진다. 따라서 도로곡률의 변화에 따라 sub-block의 위치를 동적으로 이동시키는 것이 필요하며, 이를 위하여 sub-block의 크기가 도로곡률 변화를 충분히 만족하는 것이 선행되어야 한다. 이때, 한 개의 sub-block 내부에는 차선의 일부분이 반드시 존재하여야 한다는 전제를 바탕으로, sub-block의 크기를 다음의 기준에 따라 결정한다.

- Sub-block의 크기 : 도로의 설계조건을 고려하였을 때^[18], 입력영상에서 차선의 위치는 도로의 곡률반경 r 과 기울기로 표시된다. 따라서 허용 최대 속도(최대속도 : 100 [km/h])로 주행하는 차량이 최대 안전허용 도로조건을 만족하는 곡률반경(690~∞ [m])과 기울기(0~1.718 [°])의 변화에 따라, 입력영상에서 차선의 위치변화가 벗어나지 않는 최대크기를 sub-block의 크기로 결정한다.
- 각각의 sub-block 크기와 개수 및 초기위치 : 입력영상은 핀홀모델을 기반으로 원근투영에 의해, 실제로 존재하는 한 점과 대응된다. 그러므

로 제시되어진 sub-block의 크기 결정기준에 따라, 주행차량으로부터 원 거리에 있는 각각의 sub-block 크기가 결정된다. 또한, 결정된 sub-block의 크기에 대응하는 실제 거리(20 [m])를 고려하여, 허용최대 속도(최대속도 : 100 [km/h])로 주행하는 차량이 전방 장애물에 대해 요구되는 최소 안전거리(100 [m])이내의 차선을 검색할 수 있도록 sub-block의 개수를 최소로 결정한다. 따라서, 결정되어진 각각의 sub-block 내부에 차선이 반드시 존재하기 위해, 직선도로에서 차량이 출발하는 것을 전제로 sub-block의 초기위치를 결정한다.

지금까지 제시된 기준에 따라, 본 연구에서는 주행도로를 실제 도로설계 조건을 가지고 모델링하고, 도로조건에 적합한 크기의 sub-block을 선정하는 과정을 보이며, 시뮬레이션을 통해 sub-block의 크기를 검증한다. 주행도로의 모델링에 앞서 구성 요소의 좌표계를 결정할 때, 실제로 존재하는 임의의 한 점과 카메라를 통해 나타나는 영상과의 대응관계는 그림 1과 같다.

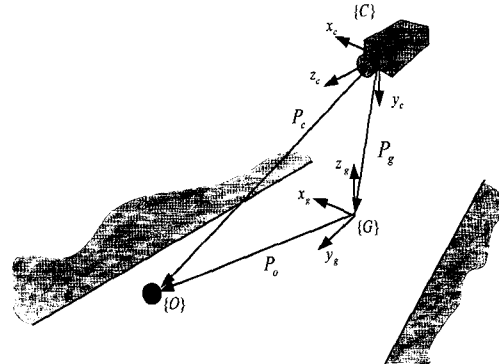


그림 1. 카메라와 실제 물체 사이의 좌표정의
Fig. 1. Coordinate systems between a camera and a point in real world.

카메라를 기준으로 도로상의 실제 한 점의 위치 벡터를 $P_c = [p_{cx} \ p_{cy} \ p_{cz}]^T$ 라고 정의하면 일반적으로 식 (5)와 같이 나타낼 수 있다.

$$P_c = {}^c R_g P_o + P_g \quad (5)$$

여기서 P_o 는 도로 좌표계(G)를 기준으로 한 물체(O)

사이의 위치 벡터이며, P_g 는 카메라 좌표계 {C}를 기준으로 한 도로 좌표계 {G}사이의 위치 벡터이다. 카메라는 도로중앙에서 정면을 향해 도로와 수평으로 설치되었다고 가정하면 ${}^cR_g = R(x, 90^\circ)$, $P_g = [0 \ d \ 0]^T$ 이다. 입력영상과 물체의 pixel 좌표 관계에서, 카메라의 투사 모델은 그림 2와 같다.

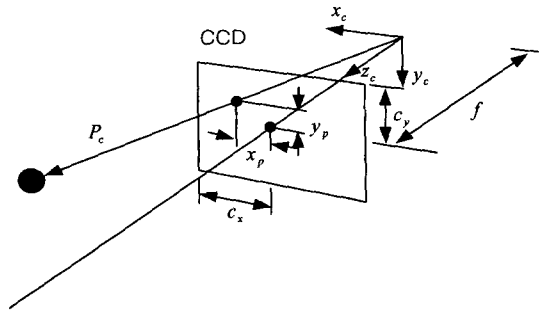


그림 2. 카메라 투사모델
Fig. 2. Projection model of camera.

카메라 투사 모델로 일반적으로 사용하는 핀홀모델을 사용하면 간단한 비례식으로 카메라의 투사 모델을 다음과 같이 얻을 수 있다.

$$x_p = \frac{p_{cx}}{p_{cz}} f \quad (6)$$

$$y_p = \frac{p_{cy}}{p_{cz}} f \quad (7)$$

여기서 f 는 초점거리이다. 이를 영상의 pixel 좌표로 변환하면 식 (8), (9)와 같다.

$$x_i = \frac{p_{cx}}{p_{cz} \delta x} f + c_x \quad (8)$$

$$y_i = \frac{p_{cy}}{p_{cz} \delta y} f + c_y \quad (9)$$

여기서 δx , δy 는 각각 CCD 소자하나의 가로, 세로 크기이며, (c_x, c_y) 는 CCD입력 화상의 중앙점의 좌표를 나타낸다. hidden 영역을 계산하기 위해 지면과 카메라는 수평을 이룬다고 가정하면 입력영상에 나타나지 않는 아래쪽 양 모서리의 실제 거리는 식 (10), (11)과 같다.

$$x_h = \frac{w_c y_h \delta x}{2f} = \frac{w_c \delta x p_{gz}}{h_c \delta y} = \frac{w_c \delta x}{h_c} d \quad [m] \quad (10)$$

$$y_h = \frac{2f p_{gz}}{h_c \delta y} = \frac{2f}{h_c} d \quad [m] \quad (11)$$

여기서 w_c , h_c 는 각각 입력영상의 가로, 세로 방향의 pixel 수를 나타내며, p_{gz} 는 지면에 대한 카메라의 높이를 나타낸다. 주행 도로의 횡단선형을 고려하기 위해 도로를 원호로 모델링하면 그림 3과 같다. 이때, 고속도로의 경우 최소회전반경은 600 [m]로 제한되며, 반경 r 이 무한대로 커지면 직선도로가 된다.

그림 3에서 r_o 는 카메라의 중심에서 양쪽 차선까지

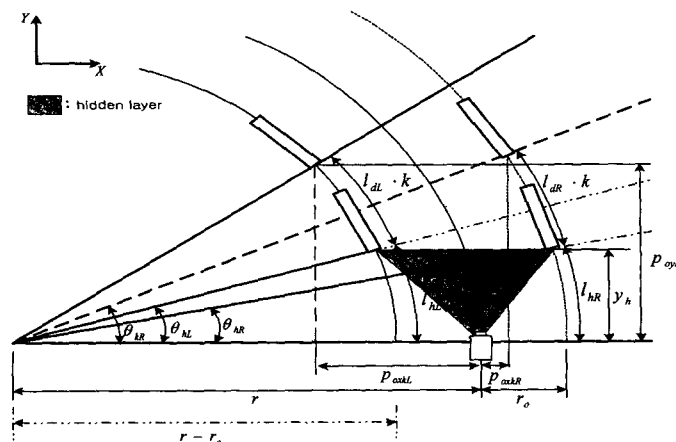


그림 3. 도로의 횡단선형 모델링
Fig. 3. Modeling of the road.

의 거리로 \pm 값을 갖는다. 따라서, 왼쪽차선에 대해서는 $r-r_0$ 이며, 오른쪽 차선에 대해서는 $r+r_0$ 가 된다.

호의 중심을 기준으로 입력영상의 가장 아래쪽에 보이는 왼쪽차선과, 오른쪽 차선에 대한 시작점과 카메라의 위치 사이의 호각은 각각 θ_{hL} 과 θ_{hR} 이다. 이때, 오른쪽 차선에 대하여 유도하여 보면 식 (12), (13)과 같이 유도되어질 수 있다.

$$\theta_{hR} = \sin^{-1} \frac{y_h}{r+r_0} \quad (12)$$

$$l_{hR} = (r+r_0)\theta_{hR} = (r+r_0)\sin^{-1} \frac{y_h}{r+r_0} \quad (13)$$

여기서 r_0 는 카메라의 중심에서 양쪽 차선까지의 거리로 \pm 값을 갖는다. 그림 3에서처럼 왼쪽차선에 대해서는 $r-r_0$ 이며, 오른쪽 차선에 대해서는 $r+r_0$ 가 된다. 회색영역은 카메라에 영상으로 나타나지 않는 영역이며, y_h 는 그에 대한 거리로 식 (11)과 같다. 이를 이용하여, k 번째 오른쪽 차선의 호각 θ_{hR} 은 식 (14)와 같이 표시된다.

$$\theta_{hR} = \frac{l_{hR} + l_{dR} \cdot k}{r} \quad (14)$$

$$= r^{-1} \left\{ (r+r_0)\sin^{-1} \left(\frac{y_h}{r+r_0} \right) + l_{dR} \cdot k \right\}$$

또한, 도로 좌표계(G)를 기준으로 하는 오른쪽 차선의 위치 벡터는 식 (15)와 같다.

$$\begin{aligned} P_{ozkR} &= (r+r_0) \cos \theta_{hR} - r \\ P_{oykR} &= (r+r_0) \sin \theta_{hR} \\ P_{ozkR} &= 0 \end{aligned} \quad (15)$$

구해진 식 (15)를 식 (5)에 대입한 뒤, 식 (8), (9)에 대입하여 정리하면 화면에 나타나는 오른쪽 차선의 좌표는 식 (16), (17)과 같이 정의될 수 있다.

$$x_{ikR} = \frac{(r+r_0) \cos \theta_{hR} - r}{(r+r_0) \sin \theta_{hR}} \frac{f}{\delta x} + c_x \quad (16)$$

$$y_{ikR} = \frac{d}{(r+r_0) \sin \theta_{hR}} \frac{f}{\delta y} + c_y \quad (17)$$

정의되어진 차선의 좌표에 의하여 입력영상에 나타나는 k 번째 차선의 시작위치는 다음의 식 (18)과 같이 정의할 수 있다.

$$W_{pk}(r) = [x_{ik} \ y_{ik}]^T \quad 0 \leq W_{pk}(r) \leq M \quad (18)$$

여기서 M 은 입력영상의 최대크기이며, pixel의 가로, 세로의 좌표로 $[640 \ 480]^T$ 이다. 식 (18)에 의해 정의된 위치에서 차선정보를 입력받기 위한 sub-block의 크기는 하나의 차선이 모두 들어 올 수 있는 크기로 나타내면, k 번째 차선의 시작위치와 $k-1$ 번째 차선의 시작위치의 차이로 정의되어질 수 있다. 이때 식 (16), (17)에서 차선의 위치는 회전반경 r 의 함수이며, r 이 고속도로의 최소 회전반경 690[m]에서 직선도로일 경우 ∞ 까지 변화하였을 때 차선의 정보가 sub-block을 벗어나지 않는 최대범위를 sub-block의 크기로 정의한다. 이는 식 (19)와 같다.

$$\begin{aligned} W_{sk} &= \max \{ \Delta W_{pk} = W_{pk}(r) - W_{pk-1}(r) \mid 690 < r < \infty, \\ & \quad k=1, 2, \dots, n \} \end{aligned} \quad (19)$$

도로의 종단 선형은 sub-block의 세로 길이에 영향을 미친다. 표 1에 나타낸 종단선형 설계 기준으로부터, 100 [km/h]에 대한 볼록 선형의 최소 길이는 3[%]의 경사가 된다. 따라서, 변화량 $\delta\theta = \tan^{-1} 0.03 = 1.718$ [°]는 식 (15)의 P_{ozk} 를 $P_{ozk} = P_{oyk} \sin \delta\theta$ 로 변화시키며, 이는 식 (19)의 sub-block 크기에 거의 영향을 미치지 않으므로 무시될 수 있다.

표 1. 도로의 종단선형 설계기준 기울기
Table 1. Standard design of a vertical curvature.

설계 속도 [km/h]	곡 선 형	최소종단곡선의 반경 [m]	종단곡선의 최소길이[m]
120	볼록선형	11,000	100
	오목선형	4,000	
100	볼록선형	6,500	85
	오목선형	3,000	
80	볼록선형	3,000	70
	오목선형	2,000	
60	볼록선형	1,400	50
	오목선형	1,000	

정의되어진 sub-block의 크기는 $f=0.024$ [m],

$d=1.2$ [m], $\delta x=8.577 \times 10^{-6}$ [m], $\delta y=8.316 \times 10^{-6}$ [m], $l_d=20$ [m], $cx=320$, $cy=240$ 에서 그림 4와 그림 5의 시뮬레이션을 통하여 표 2와 같이 나타나며, 이는 앞서 제시한 바와 같이 영상획득 차량의 직선도로 출발을 전제로 하였을 때, sub-block의 초기위치와 도로설계시의 곡률을 동시에 만족한다^[18].

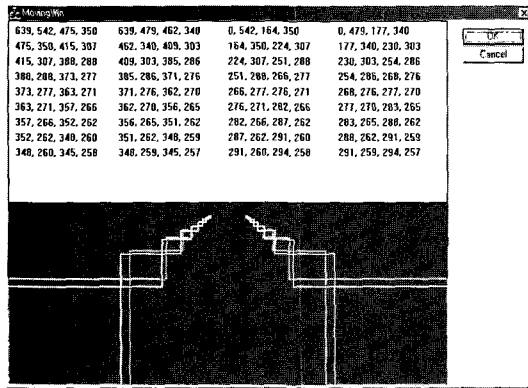


그림 4. 직선도로에 대한 시뮬레이션
Fig. 4. Simulation of straight road.

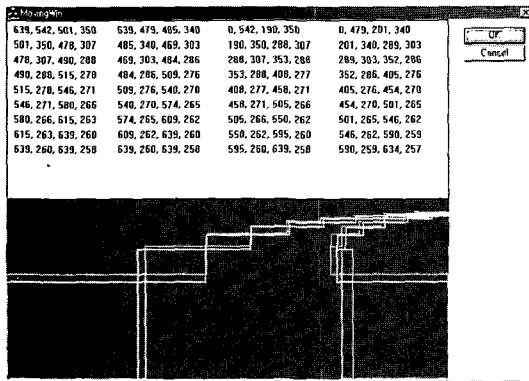


그림 5. 곡선도로에 대한 시뮬레이션
Fig. 5. Simulation of curved road.

표 1. 결정된 sub-block 크기와 초기위치
Table 1. Determined sub-block size and initial position.

Sub-block 순서	X축 크기	Y축 크기	초기위치(L)	초기위치(R)
1	36.5214	6.3909	251,280	382,280
2	41.0808	9.8328	238,286	395,286
3	50.6943	17.0707	218,298	411,298
4	77.1008	36.9308	173,324	450,324
5	193.1809	139.5126	65,410	555,410

시뮬레이션 결과, 식 (15), (18)에 의해 sub-block의 개수는 100 [km/h]에서 최소안전 거리인 100 [m]를 만족하는 세로로 5개가 적합하다. 따라서, 전체 640×480 [pixel] 중에서 20.29 [%]만을 사용하여 차선의 위치를 탐색할 수 있다. 또한, 차선 이외의 정보를 제한하여 옆 차선 차량 등의 잡음에 대해 강한 특성을 가지게 된다.

III. Moving window에 의한 차선 인식

지금까지 차선정보를 구하기 위하여 전처리를 실시하고 이를 입력영상의 일부분으로 한정하기 위한 최적의 sub-block 제시하였다. III장에서는 지금까지 구해진 정보들을 바탕으로 차선을 추출하고, 다음 시점에서 차선 추출의 정확도를 높이기 위해 sub-block의 위치를 옮기는 기법을 설명한다. 본 논문에서는 이를 moving window라고 정의한다.

1. Moving window의 내부영역 분석

전처리에 의해 moving window내부에서 에지로 나타난 영역을 labeling한 후, central moment c_1 과 c_2 를 구하면, 그림 6과 같다.

입력영상에서 sampling시간 95 [ms] 이후의 차선 위치는 moving window의 크기에 비하여 급격히 변화되지 않으므로, 그림 6의 점선과 같이 $k-1$ 시점에서 검출된 차선이 가우시안 분포를 가진다고 할 때, k 시점에서 central moment의 위치는 c_1 이 c_2 보다 차선으로 높은 신뢰도를 가진다. 이때, 차선의 영역으로 결정되어진 c_1 을 바탕으로 check point를 지정할 수 있다.

2. Check point

그림 7과 같이 차선영역의 에지로 나타난, 입력영상에 대한 중심모멘트는 그림 8과 같이 표시된다.

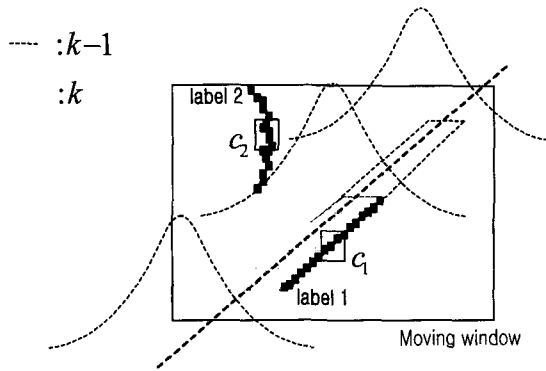


그림 6. Moving window의 내부영역 분석
Fig. 6. Region analysis of moving window inside.

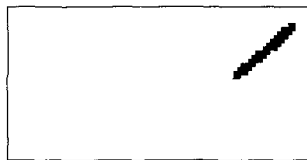


그림 7. 입력영상
Fig. 7. Input image.

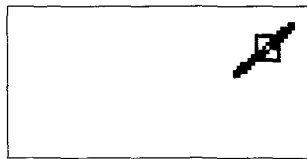


그림 8. 중심모멘트
Fig. 8. Central moment.

찾아진 central moment는 다음절에서 설명되어질 꼭 선근사를 통하여 y 축 방향으로 일정위치로 그림 9과 같이 결정되어진다. 이를 본 논문에서는 check point라고 명한다.

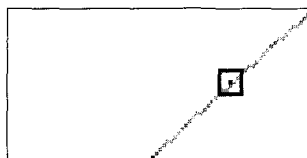


그림 9. 측정값
Fig. 9. Measurement value.

결정된 check point를 측정값으로 하였을 때, 칼만필터를 통하여 그림 10와 같이 check point를 중심으로 하는 다음시점의 moving window 위치를 예측할 수 있다.

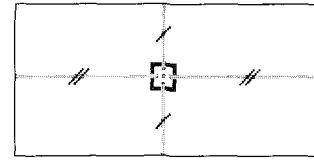


그림 10. 예측값
Fig. 10. Predicted value.

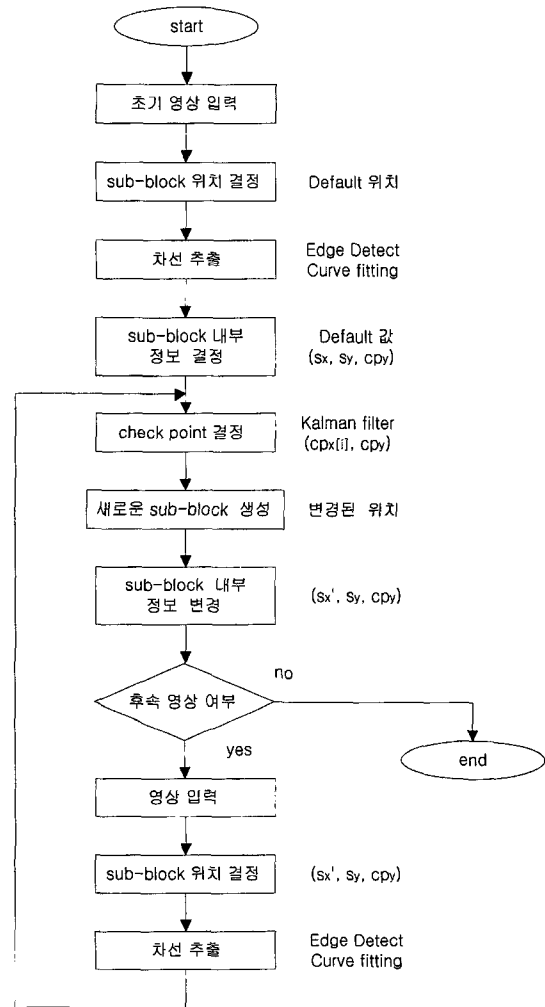


그림 11. Moving window 기법의 흐름도
Fig. 11. Flowchart of moving window technique.

3. Moving window에 의한 차선 추출

앞서 결정되어진 sub-block의 내부에 영상이 입력되었을 때, 프레임 마스크를 사용하여 차선 성분을 추출하고 연산을 통해 check point를 결정하며, 이를 바탕으로 칼만필터를 사용하여 새로운 sub-block 즉, moving window를 결정하는 과정을 순서대로 나타내면 그림 11과 같다.

입력영상에서 도로의 차선에 해당하는 성분이 직선 도로에는 선형으로 표현될 수 있지만, 곡선도로에서 횡축의 범위가 크게 나타나는 경우에는 고차다항식으로 표현하는 것이 적당하다. 임의의 N 차 다항식이 식 (20)과 같다면,

$$g(y) = a_0 + a_1y + a_2y^2 + \dots + a_Ny^N \quad (20)$$

계수 a_n 은 편차의 제곱의 합을 최소화하는 편미분 방정식으로부터 유도되어진, 식 (21)을 연립으로 놓고 구할 수 있다^[9].

$$\sum_{n=0}^N \left[\sum_{i=1}^L y_i^{n+k} \right] a_n = \sum_{i=1}^L x_i y_i^k, \quad (21)$$

단 $k=0, 1, 2, \dots, N$

주행중의 도로영상이 곡선일 때, 계수 a_n 의 차수는 $n=0, 1, 2$ 로 정의한다. 식 (21)에 의해 곡선근사된 정보는 차선정보뿐만 아니라, 주행차량의 진행방향을 나타내는 무한원점(vanishing point)의 정보까지 포함하고 있다. 따라서, 현 시점(k)으로부터 다음 시점($k+1$)에 입력될 영상에 대한 moving window의 위치를 예측하는 정보로 유효하다. 또한 이미 결정되어진 이전 시점($k-1$)의 check point 정보까지 포함시켜 근사화함으로써, 다음 시점($k+1$)의 입력영상이 불명확할 때도 안정적으로 차선을 추출할 수 있게 한다.

4. 칼만필터에 의한 moving window의 위치예측

다음 시점($k+1$)에서 입력되는 영상의 일부분만을 탐색영역으로 결정하기 위하여, 탐색영역의 위치를 옮기는 moving window 기법이 필요하다. 앞에서 제시되어진 전처리와 곡선근사 방법으로 구해진 이전 시점($k-1$)의 check point 위치정보를 바탕으로 현 시점(k)의 moving window 위치를 예측하고, 예측한 값과 실제 측정된 값과의 오차를 보정하는 칼만필터를 사용한다^[10-12].

먼저, 예측단계에서 $k-1$ 시점의 실제측정을 기반으로 현 시점(k)의 상태예측을 x_k^- 이라 할 때, 식 (22)와 같이 나타낼 수 있다^[10]. 시스템 행렬 F 는 차량의 운동방정식에 의해 식 (23)과 같다. Δt 는 check point의 측정시간을 나타낸다.

$$x_k^- = Fx_{k-1}^+ + w_{k-1} \quad (22)$$

$$F = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \quad (23)$$

이때, w_{k-1} 는 시스템 모델의 잡음을 의미하며, 가우시안 잡음이다. 이의 공분산행렬을 Q_{k-1} 이라 하면, 예측값의 공분산 행렬은 식 (24)와 같다.

$$P_k^- = P_{k-1}^+ + Q_{k-1} \quad (24)$$

이후, 측정단계에서 측정벡터 y_k 는 식 (25)와 같이 나타낼 수 있으며, v_{k-1} 은 측정오차로써 평균이 0인 가우시안 잡음이다.

$$y_k^- = Hx_{k-1} + v_{k-1} \quad (25)$$

v_{k-1} 는 예측단계의 w_{k-1} 와 비상관(uncorrelate)되어 있고, 측정단계에서 공분산행렬 R_{k-1} 로 한다. 또한, 입력영상의 시간변화율은 영상으로부터 관측될 수 없으므로 관측행렬(measurement matrix)은 식 (26)과 같이 나타낼 수 있다.

$$H = [1 \ 0] \quad (26)$$

예측된 값에 관측편차를 더하여 상태벡터를 보정하기 위한 순환적(recursive) 칼만필터는 다음의 식 (27)과 같이 나타낼 수 있다.

$$x_k^+ = x_k^- + G_k(y_k - Hx_k^-) \quad (27)$$

측정값과 추정치 사이에 적절한 가중치를 주어 상태벡터를 조절하는 칼만이득 G_k 는 식 (28)과 같이 나타낼 수 있다.

$$G_k = P_k^- H^T [HP_k^- H^T + R_k]^{-1} \quad (28)$$

이는 예측단계의 공분산 행렬을 구할 때, 식 (29)와 같이 사용된다.

$$P_k^+ = P_k^- - G_k H P_k^- \quad (29)$$

그러므로, 식 (27)에서 보정된 상태벡터에 대한 다음 시점의 상태예측은 식 (30)과 같다.

$$x_{k+1}^- = Fx_k^+ + w_k \quad (30)$$

칼만필터는 측정오차의 공분산 행렬과 초기값, 이에 관한 오차추정치를 포함하는 초기정보인 x_0 와 단위행렬 P_0 에 의해 만들어지고, 이는 칼만필터를 계산하는데 사용된다. 따라서 초기값에 의해 칼만이득의 측정값이 보정되고, 반복수행 때마다 관측편차에 대하여 칼만이득이 조정되므로 다음상태를 예측할 때 정확성이 향상된다.

지금까지 설명되어진 moving window는 그림 12에서 고가도로 밑의 그림자와 그림 13에서 수직성분을 가지는 트럭 등의 잡음환경에도 차선추출이 용이함을 보인다.

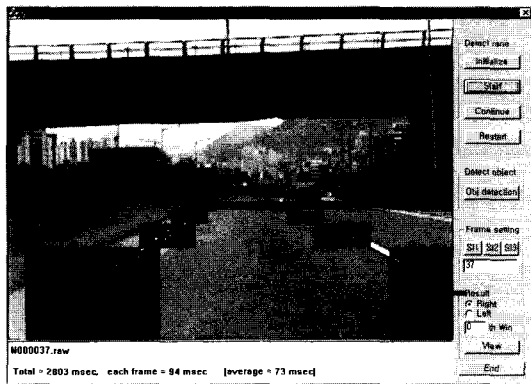


그림 12. 잡음환경 I.
Fig. 12. Noisy environment I.

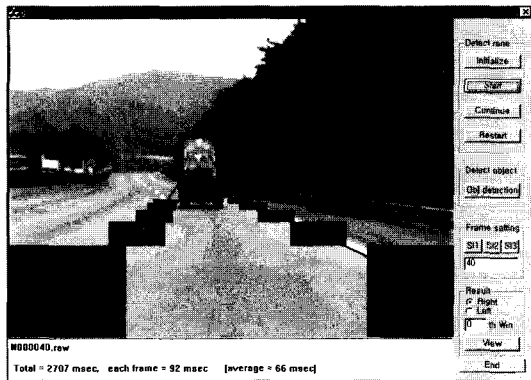


그림 13. 잡음환경 II.
Fig. 13. Noisy environment II.

IV. Moving window에 의한 장애물 검출

Moving window를 사용하여 차선을 정확하고 빠르게 추출할 수 있었으며, 본 장에서는 주행 차량의 자기 차선 내부에 존재하는 장애물 검출에 대해 설명한다.

1. 도로 중앙축 추출 및 장애물 인식
주행차량이 차선중앙에 있을 때, 도로의 중앙축은 좌우차선의 차로 나타낼 수 있으며, 차량이 진행할 방향에 대한 정보를 가지고 있다. 앞서 구해진 차선 위치를 $F_L(y)$, $F_R(y)$ 라고 할 때, 도로의 중앙축은 식 (31)과 같이 나타낼 수 있다.

$$F_C(y) = F_R(y) - F_L(y) \quad (31)$$

식 (31)의 도로 중앙축의 위치에서 밝기값을 $f_c(x, y)$ 라고 할 때, 이는 256 gray level을 가지게 된다. 주행 차선내의 장애물 차량은 뒷범퍼 및 트렁크 등에 의해 가로성분의 특징을 포함하게 되므로, 수평방향의 에지 그래디언트를 구하면, 장애물로 예상되는 특징을 구할 수 있다. 구해진 수평방향 에지를 2진화한 $f_{CB}(x, y)$ 에 대하여, 256 gray level에서 장애물의 예상위치 $f_{Ob}(x, y)$ 를 검출한다.

장애물 예상위치는 그림 14에서 장애물의 시작(Ob1) 위치가 되며, x 축 방향으로 양쪽 차선까지 60%를 밀 변변으로 하여 왼쪽(Left1), 오른쪽(Right1) 아래를 결정한다. 이는 실제로도 넓이에 대한 차량폭의 평균에 해당된다. 또한 장애물 대상을 차량 혹은 도로표시로 구별하기 위하여, 밀변의 절반에 해당하는 길이로 왼쪽(Left2), 오른쪽(Right2) 위를 결정한다. 따라서, 장애물 인식의 범위를 주행차선 내부로 제한할 수 있으며, 그림 15는 제안한 방법을 사용하여 장애물을 검출영역을 지정한 예를 나타내고 있다.

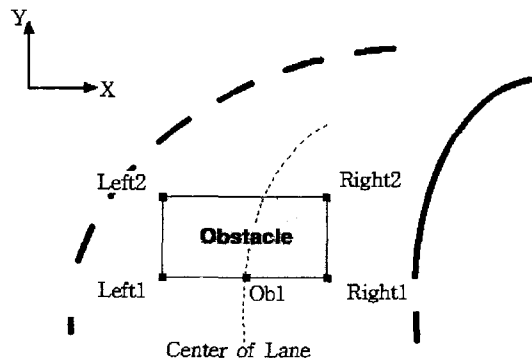


그림 14. 장애물 검출영역
Fig. 14. Area of obstacle detection.

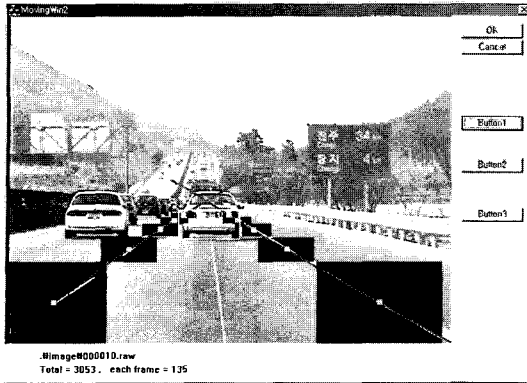


그림 15. 제안된 방법에 의한 장애물 검출
Fig. 15. Obstacle detection by a proposed method.

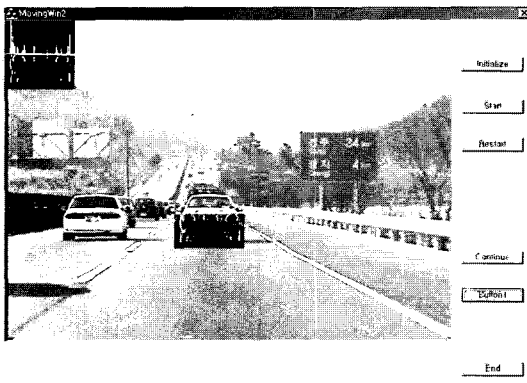


그림 16. 장애물 검출
Fig. 16. Obstacle detection.

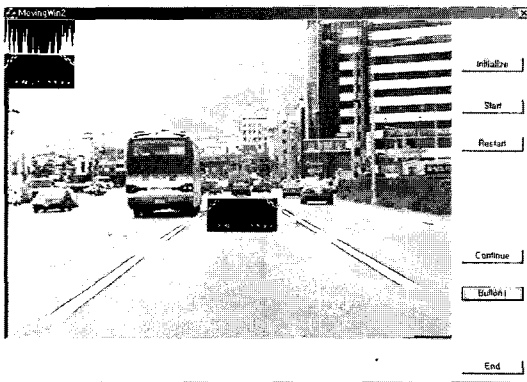


그림 17. 도로표시로 인식
Fig. 17. Recognition of marks.

장애물이 예상되는 영역($n \times m$)에서 입력정보의 양을 비교하기 위하여 식 (32)으로 정규화한다.

$$S_i = \sum_{j=1}^m f_B(x_i, y_j) \quad (\text{단, } i=1, \dots, n, \text{ Res}_d = 50)$$

$$\text{Res}_i = \frac{\text{Res}_d}{\sup \|S_i\|} \quad (32)$$

이때, Res_d 는 정규화를 위해 요구되어지는 값이다. 같은 방법으로 장애물이 예상되는 영역의 넓이를 정규화한 후, 분산을 구하여 식 (33)와 같이 장애물 여부를 구분할 수 있다. 그림 16과 그림 17는 제안되어진 방법으로 장애물과 도로표시를 구분한 경우를 나타내고 있다.

$$F_{ob}(i) = \begin{cases} \text{장애물} & , \text{ var}[Res_i] \geq Th \text{ 일때} \\ & (Th = 4.32) \\ \text{도로표시, 그외의 경우} & \end{cases} \quad (33)$$

V. 실험

본 장에서는 실험에 사용되어진 환경을 설명하고, 최적의 크기로 조정되어진 moving window를 사용하여 차선 인식 및 장애물 검출의 경우를 그렇지 않은 경우와 비교하여 속도 및 정확성 측면에서 뛰어난 것을 보인다. 또한, 제안되어진 방법을 사용하였을 때, 실시간 처리가 가능함을 보인다.

1. 실험환경

실제 차량에서 입력되는 영상을 얻기 위하여 그림 18와 같이 운전자 눈 높이인 지면으로부터 1.1 [m]에 캠코더를 고정한 후, 다양한 조건의 도로환경을 녹화하여 사용하였다.

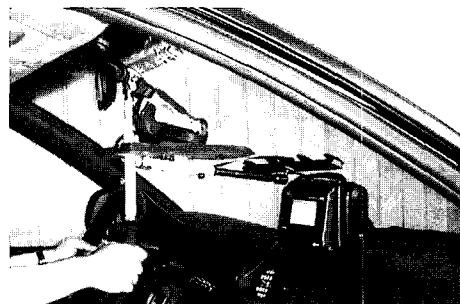


그림 18. 카메라 위치
Fig. 18. Position of camera.

고속도로 주행 중의 속도는 실험의 기준속도인 100 [km/h]로 일정하게 유지하였다. 실험에 사용된 도로구간은 부산~양산간이며, 실험차량은 현대 아반떼, 카메라는 LC-AH30D (L.G), 그리고 frame grabber는 EURESYS의 DOMINO MAXI, 컴퓨터는 PentiumIII-450을 사용하였다. 입력영상은 30 [mS]마다 640×480 [pixel]의 해상도로 메모리에 누적되며, 계속 새로운 영상으로 갱신하였다.

2. 실험결과

다음 그림 19은 직선도로에서 곡선도로로 주행하는 상황에서, 찾아진 차선의 위치와 칼만필터의 추종결과를 주행차량에 가장 근접한 moving window 1개의 x 축 좌표 변화를 사용하여 나타내고 있다. 이때, 추출된 차선 위치를 바탕으로 주행차량의 진행방향을 나타내면 그림 20과 같다.

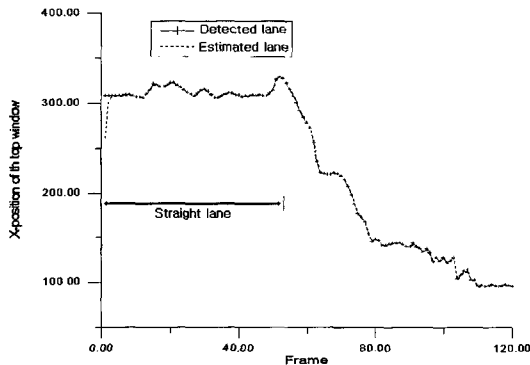


그림 19. 제안된 moving window에 의한 차선 위치 추출 및 칼만필터 추종 결과

Fig. 19. Detection of the land position and result of Kalman filter estimation by the proposed moving window.

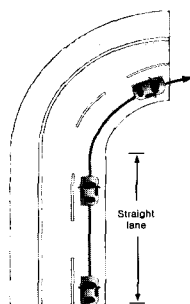


그림 20. 차량의 주행방향
Fig. 20. Driving direction of a vehicle.

그림 19에서와 같이 전체 120 프레임을 기준으로 PentiumIII-450의 컴퓨터에서 연산하게 되면, 각 프레임의 평균 처리시간은 약 95 [mS]가 소요되며, CPU의 처리속도에 따라 평균처리 속도를 더 높일 수 있다.

그림 19과 동일한 환경에서 moving windows의 크기를 ±15 [%]로 조절하였을 때, 추출된 차선의 위치는 그림 21과 같다. Moving window의 초기위치를 같이 하였을 때, 직선도로에서는 최적 크기의 moving window를 사용하였을 경우와 비교하여 추출된 차선의 위치가 큰 차이가 없음을 알 수 있다. 그러나, 곡선도로에 진입하게 되면 추출된 차선의 위치에 큰 오차가 발생하게 된다. 즉, moving window의 크기가 작은 경우에는 차선과 차선 사이의 빈 공간에 의해 지속적으로 정보를 놓치는 경우가 발생하며, 크기가 큰 경우에는 옆 차선의 차량 및 옆 차선 등에 영향을 받게되므로 외부잡음에 의한 차선추출의 신뢰가 떨어짐을 알 수 있다.

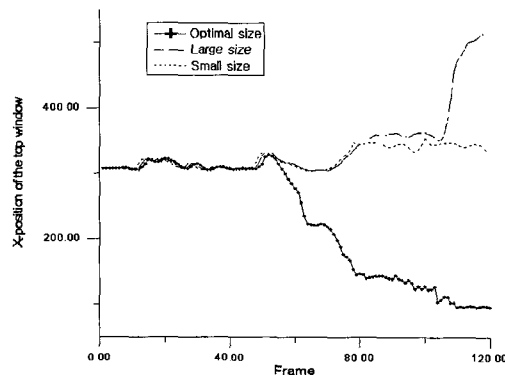


그림 21. 변경된 크기의 moving window에 의해 추출된 차선 위치

Fig. 21. Detection of the land by different sizes of the moving window.

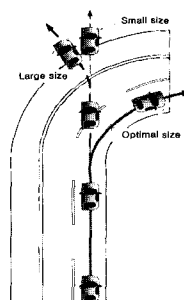


그림 22. 차량의 주행방향
Fig. 22. Driving direction of a vehicle.

조절된 크기의 moving window에서 추출된 차선의 위치를 바탕으로 차량의 주행방향을 나타내면 그림 22와 같으며, 차량이 주행차선을 벗어나는 상황이 발생하게됨을 알 수 있다.

3. 검토

Moving window를 도로 설계조건에 따라 최적의 크기로 조정하였을 때, 차선 추출의 정확성을 향상시킬 수 있었다. 샘플링 영상은 처리시간 95 [mS]가 지난 시점에서 제일 최근에 메모리에 갱신된 영상을 사용하게 되므로 처리시간과 입력영상 사이에 최대 4 프레임 이내의 오차를 가진다. 이는 고속도로를 100 [km/h] 속도로 주행할 때 0.5초의 사전정보만 있으면 충돌사고의 60 [%]는 예방할 수 있는 것으로 비추어 볼 때, 유효한 속도이다.^[21] 또한, 영상을 획득하고 있는 차량의 속도를 감지하여 추가하게 되면, 각각의 속도에 맞추어 미리 시뮬레이션된 최적 크기의 moving window를 적용할 수 있다.

VI. 결 론

본 논문에서는 도로의 설계조건에 따라 차선과 장애물을 인식하기에 적합한 moving window의 크기를 최적으로 선택하는 기법을 제시하였다. 제시된 moving window의 크기는 영상획득 차량의 허용최대 속도와 최대 안전 도로조건의 변화를 고려하여, 어느 순간에도 차선의 일부가 입력될 수 있도록 최대의 크기로 설정하였다. 또한 같은 상황에서 최소 안전거리 내부를 검색할 수 있도록 최소한의 개수로 moving window를 결정하였다. 제시된 최적의 moving window는 현 시점에서 내부에 입력된 차선에 대한 곡선근사와 칼만필터를 통해, 다음 시점에서 차선입력 예상위치로 이동한다. 또한, 구해진 차선 정보를 근사하여 주행 방향을 결정할 때, 이전 시점의 정보까지 고려함으로써 외부환경에 의한 갑작스런 정보소실을 대비하였다. 따라서, 육교 밑의 그림자 등에 의하여 입력영상이 소실되는 경우에도 제안하는 방법은 유효함을 보였다. 추출된 차선으로부터 도로 중앙축을 구하고 장애물을 검색하므로, 주행방향 전방의 차량만을 장애물로 고려할 수 있다. 이는 도로의 선형에 변화가 빈번한 우리나라의 고속도로의 상황에 적합하며, 산업현장에서 라인표시를 따라 느린 속도로 이동하는 AGV(Autonomous Guided Vehicle)의 주

행에도 적용 가능할 것이다.

참 고 문 헌

- [1] Reinhold Behringer and Nikolaus Muller, "Autonomous Road Vehicle Guidance from Autobahnen to Narrow Curves," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 810-815, Oct. 1998.
- [2] Christopher E. Smith, Charles A. Richards, Scott A. Brandt and Nikolaos P. Papanikolopoulos, "Visual Tracking for Intelligent Vehicle-Highway Systems," *IEEE Transactions on Vehicular Technology* 45(4), pp. 744-759, 1996.
- [3] Makoto Hirano, "Development of Vehicle-Following Distance Warning System for Trucks and Buses," *IEEE-IEE Vehicle Navigation & Information Systems conference, Ottawa-VNIS'93*, 1993.
- [4] D. Koller, K. Daniilidis and H. -H. Nagel, "Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes," *International Journal of Computer Vision*, 10:3, pp. 257-281, 1993.
- [5] 최승욱, 이장명, "이동창을 이용한 차선 인식 및 장애물 감지," 전자공학회논문지, 제36권 S편, 제1호, pp. 93-103, 1999년 1월
- [6] Woong Kwon, Jae-Won Lee, Dongmok Shin, Kyoungsig Roh, Dong-Yoon Kim, and Sukhan Lee, "Experiments on Decision Masking Strategies for a Lane Departure Warning System," *Proc. of IEEE Conf. on Robotics and Automation*, pp. 2596-2601 Detroit, May 1999.
- [7] 이준용, "영상처리 기반의 차선 인식 알고리즘," 제어·자동화·시스템공학 논문지, 제4권 제6호, pp. 759-764, 1998년 12월
- [8] 정준익, 최성구, 노도환, "직선 Edge 추출에 의한 주행 방향 및 장애물 검출에 관한 연구", 한국 자동 제어 학술회의 논문집, Vol. 2, pp. 97-100, 1996
- [9] Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical Recipes in C*,

- Cambridge Univ. Press, pp. 656-706, 1992.
- [10] Harold W. Sorenson, Kalman Filtering: Theory and Application, *IEEE PRESS*, pp. 16-25, 1985.
- [11] 이철현, 김형태, 설성욱, 남기곤, 이장명, "칼만 필터를 이용한 다중 차량 추적 알고리즘," 전자공학 회논문지, 제36권 S편, 제3호, pp. 359-365, 1999년 3월
- [12] 이연석, "다중표적 추적시스템에서의 표적물의 모델," 전자공학회논문지, 제36권 S편, 제7호, pp. 34-42, 1999년 7월.
- [13] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, *Machine Vision*, McGraw-Hill International Edition, pp. 140-149, 1995.
- [14] 이성경, "Prospective of Technology Advances in Intelligent Transport System tying transportation and telecommunication worlds together," 전자공학회지, 제24권 제2호, pp. 13-25, 1997년 2월
- [15] Charles E. Thorpe, Takeo Kanade, *Vision and Navigation*, Kluwer Academic Publishers, pp. 9-82, 1990.
- [16] 김승남, 김재수, 박창호, 최장원, 두민수, 진희수, "Laser Radar를 이용한 Collision Warning System 개발," 한국자동차공학회 춘계학술대회, pp. 167-172, 1996
- [17] Toshiaki Kakinami, Jun Sato, Mitsuyoshi Saiki, and Kunihiko Soshi, "Autonomous Vehicle Control System Using an Image Processing Sensor," *SAE*, 1995.
- [18] 천병식, 고용일, 도로공학, 새론, pp. 95-168, 1994
- [19] 최진욱, 한민홍, "무인 자동차 개발 연구," 한국자동차제어학술회의논문집, pp. 88-93, 1993년 10월
- [20] 문영수, 정상철, 이준용, 강동중, 권인소, "교통 정보 검지기 및 지능형 자동차 개발을 위한 영상 처리 알고리즘," 제어·자동화·시스템공학 학회지, 제2권 제6호, pp. 50-62, 1996년 11월
- [21] 이상주, 박공만, "자동차 충돌방지용 밀리미터파 레이더 기술동향," 전자공학회지, 제26권 10호, pp. 32-40, 1999년 10월

 저 자 소 개



崔勝郁(正會員)

1996 2월 동아대학교 전자공학과 학사, 1998년 2월 부산대학교 대학원 전자공학과 석사. 2000년 2월 부산대학교 대학원 전자공학과 박사 수료. 주관심 분야는 이동로봇 등에서 영상정보를 이용한 유효정보 추출 및 궤도추적

李章明(正會員) 第 36卷 S編 第 1號 參照

현 부산대학교 공과대학 전자공학과 부교수 부산대학교 컴퓨터 및 정보통신 연구소 전자기술부장