

工學 碩士學位 論文

적응형 원근 효과 제거를 이용한 영상
기반 실시간 차선 검출 방법

Vision-Based Realtime Lane Detection Algorithm
Using an Adaptive Inverse Perspective Mapping

亞洲大學校 大學院

電子工學科

林 兌 炫

적응형 원근 효과 제거를 이용한 영상
기반 실시간 차선 검출 방법
Vision-Based Realtime Lane Detection Algorithm
Using an Adaptive Inverse Perspective Mapping

指導教授 金 容 得

이 論文을 工學 碩士學位 論文으로 提出함.

2005年 2月

亞洲大學校 大學院

電子工學科

林 兌 炫

林兌炫의 工學 碩士學位 論文을 認准함

審查委員長 金 容 得 印

審 查 委 員 鄭 己 鉉 印

審 查 委 員 朴 成 晉 印

亞洲大學校 大學院

2005年 2月

국문 요약

차량 및 교통량의 증가와 함께 교통안전에 대한 요구는 지능화된 교통 시스템에 대한 요구를 만들어왔다. 더욱이 모바일과 접목된 임베디드 시스템의 성능 향상은 그 요구를 뒷받침해 줄 수 있는 기반을 제공하고 있다. 지능형 교통 시스템에서 차선 및 차량 검출을 위해 영상 처리와 컴퓨터 비전 기법들이 많이 사용하고 있다. 그 중 영상 기반 차량 안내 시스템은 교통안전의 향상과 교통량의 증가라는 두 가지 상반된 문제를 절충시키는 수단이 될 수 있다. 이 시스템은 차량이 주행하고 있는 해당 차선에 대한 정보, 차량의 급차선 변경 여부 및 정주행에 대한 정보를 제공함으로써 운전자의 안전운행을 위한 보조 장치로서의 역할을 제공할 수 있다.

본 논문에서는 하나의 카메라를 이용한 도로의 조도, 그림자, 차선과 동일한 색의 차량 및 도로 상의 장애물 영향을 덜 받는 실시간 차선 검출 알고리즘을 제안한다. 적응형 원근 효과 제거는 차선이 될 가능성이 있는 픽셀만을 선택하기 위해 특정 영역만을 적용한 히스토그램을 이용하여 동적 임계값을 찾아내고 그 임계값을 이용하여 이진화를 수행한 후 원근 효과를 제거한다. 차선은 원근 효과를 제거시킨 영상에서 차선 파라미터를 추출하여 탐색하게 된다.

알고리즘에 대한 성능 평가 및 검증은 PC를 기반으로 수행하였으며, 그 실시간 처리 여부를 검증하기 위해 WinCE.net 4.2가 포팅된 ARM 9 프로세서를 이용한 타겟 보드 상에서 알고리즘 처리 시간을 측정하였다. 적응형 원근 효과 제거에 의한 차선 검출은 일반적인 원근 효과 제거를 이용하는 방법에 비해 그 처리 시간을 평균 80.4% 단축시킬 수 있었다. 또한 이진화를 위한 가변적인 임계값을 가지게 함으로써 고정 임계값 방식에 비해 도로 반사에 의한 화면의 명암 변화, 그림자 및 차선과 동일한 색상을 가지는 차량에 의한 오검지율을 약 10% 줄이는 효과를 보였다. 그 결과 10프레임/초의 처리 속도와 93%의 차선 검출도를 보였으며, 실제 차량 운행 중에 차량 안내 시스템의 일부로써 운전자를 위한 안전 운전 보조 장치로서의 적용 가능성을 보였다.

<제목 차례>

국문 요약

<제목 차례>	i
<그림 차례>	ii
<표 차례>	iv
제1장 서론	1
제2장 차선 검출을 위한 영상 처리 기법	4
제1절 히스토그램을 이용한 이진화 처리	4
제2절 에지 검출	8
제3절 컬러 영상의 흑백 영상으로의 변환	13
제3장 원근 효과 제거 방법	15
제1절 카메라 모델을 이용한 원근 효과 제거	15
제2절 원근 효과 제거를 위한 각 파라미터의 영향	20
제4장 적응형 원근 효과 제거 방식 제안	22
제1절 소벨 에지 검출	23
제2절 임계값 설정을 위한 마스크 영상의 생성	24
제3절 히스토그램에 의한 차선 검출	26
제4절 차선 검출	29
제5장 제안한 알고리즘 성능 평가 환경 및 결과	32
제1절 알고리즘 평가를 위해 사용된 하드웨어 구성	32
제2절 평가 환경 및 결과	34
제6장 결론	42
참고문헌	44
Abstract	47

<그림 차례>

그림 2-1. 임계값에 따른 이진화 $O(x,y)$ 영상	5
그림 2-2. 임계값에 따른 이진화 $R(x,y)$ 영상	5
그림 2-3. 히스토그램의 표시	6
그림 2-4. 이진 영상을 위한 세로 방향 히스토그램	7
그림 2-5. 날씨(조도)에 따른 <그림 2-3> 방식의 히스토그램	7
그림 2-6. 연속적인 1차원 평면에서의 에지 검출	8
그림 2-7. 가우시안 분포에 따라 잡음을 삽입한 영상에 대한 에지 검출	12
그림 2-8. 컬러 영상을 YCrCb로 변환한 영상	14
그림 3-1. 원근 효과를 가지는 도로 영상	15
그림 3-2. 원근 효과 제거를 위한 실제 도로와 카메라 특성과의 상관관계	16
그림 3-3. 카메라의 위치 설정에 관계된 변수	17
그림 3-4. 원근 효과 제거에 사용하게 될 렌즈의 파라미터	18
그림 3-5. 카메라 파라미터(θ)와 Aperture(2α)의 관계	19
그림 3-6. 영상 픽셀 데이터와 3D 도로 평면과의 관계	19
그림 3-7. 원근 효과 제거 후의 도로 영상	21
그림 4-1. 날씨에 따른 도로 영상	23
그림 4-2. 날씨에 따른 도로의 소벨 에지 검출 영상	24
그림 4-3. 에지 부분을 채우기 위한 필터	24
그림 4-4. 필터에 의해 만들어진 마스크 영상	25
그림 4-5. 도로 부분에 해당되는 부분만 추출한 마스크 영상	25
그림 4-6. 마스크에 의한 히스토그램	27
그림 4-7. 동적 임계값에 의한 이진화 영상	27

그림 4-8. IPM 왜곡 보완 알고리즘을 적용한 도로 영상	28
그림 4-9. AIPM 후의 영상과 날씨에 따른 히스토그램	30
그림 4-10. 차선 검출 후 결과 영상	31
그림 5-1. 타겟 보드 사진	32
그림 5-2. MS Visual C++ 로 작성한 차선 검출 프로그램	33
그림 5-3. 시스템과 카메라 모듈의 인터페이스 부분의 기능 블록 다이어그램	33
그림 5-4. 차선 검출 정확도	36
그림 5-5. 날씨에 따른 차선 검출 결과 영상	37
그림 5-6. 날씨에 따른 샘플 영상의 임계값 변화	38
그림 5-7. 고정 임계값을 사용하였을 경우의 차선 검출 정확도	39
그림 5-8. 타겟 보드 상에서 IPM을 이용한 영상 처리 시간	40
그림 5-9. 타겟 보드 상에서 AIPM을 이용한 영상 처리 시간	40
그림 5-10. 잡음으로 인해 차선 검출에 실패한 영상 샘플	41

<표 차례>

표 2-1. 그래디언트 계산에 사용되는 미분 오퍼레이터	10
표 2-2. 라플라시안 계산에 사용되는 미분 오퍼레이터	11
표 5-1. 알고리즘 평가를 위한 도로 영상 획득을 위한 환경	35
표 5-2. 임계값 변화를 평가한 영상을 획득한 날짜	38

제 1 장 서 론

무선 기술과 접목된 임베디드 시스템의 발달과 함께 지능형 교통 시스템은 전 세계적으로 활발히 연구가 진행되고 있다. 지능형 교통 시스템(ITS)은 무선 음성, 데이터 통신과 인공위성을 이용한 위치 정보 시스템을 기반으로 자동차를 이용해 정보를 주고받는 기술과 다양한 정보 수집 장치(레이더, 레이저, 영상검지기, 루프검지기 등)를 통해 획득한 정보를 처리하는 고도의 수학적 기법이 접목된 분야이다. 또한 ITS 기술의 발전은 교통량과 차량의 증가로 인한 교통사고를 줄일 수 있는 안전운전 보조 장비를 탑재한 자동차의 요구를 증가시켜 왔다[1, 2].

ITS 기술은 크게 차량 내 응용, 도로 구조, 차량 내 통신, 위성 통신, 경로에 대한 설정 등 다섯 개의 주요 부분으로 나뉜다. ITS는 차선 및 차량 검출, 교통 표지판, 도로 상태, 교통사고 인지 등을 위해 영상 처리 및 컴퓨터 비전을 사용한다. ITS에서 영상 처리 및 컴퓨터 영상의 직접적인 적용은 주행 중 운전자를 돕거나 자동 운행 시스템에 사용된다. ITS의 응용으로써 영상을 기반으로 하는 차량 안내 시스템은 차량 주변의 정보를 모으고 위험 요소를 감지함으로써 교통량의 증가에 따른 교통안전을 해결하는데 도움을 준다[3].

주행하는 차량을 위한 영상 기반 차량 안내 시스템의 주요 3요소는 교통 표지판, 차선 검출, 장애물 검지이다. 이 시스템은 다양한 도로 조건, 조도 및 날씨의 영향을 적게 받아야 하며 운행 중에 실시간으로 처리가 가능하여야 하기 때문에 차량의 속도를 감안하여 그 처리 속도가 충분히 빨라야만 한다. 차량 안내를 위한 컴퓨터 영상 시스템은 영상 획득 및 처리 모듈로 구성되어 있다. 또한 이 시스템은 실용화를 위해 저가의 소형 실시간 모듈이어야만 한다[4].

안전 운전을 위한 영상 기반 도로 검지 시스템은 전 세계적으로 많은 연구가 이루어지고 있다. 영상 기반 도로 검지 시스템은 서로 다른 도로 모델(2D or 3D), 영상 획득 장비(컬러 혹은 흑백용 카메라), 하드웨어 시스템(전용 혹은 범용 목적, 병렬 혹은 시리얼) 그리고 연산 처리 기법(템플레이트 매칭, 뉴럴 네트워크 혹은 스트레오 영상) 등과 같은 여러 가지 특성에 따라 개발이 이루어져왔다[2, 5, 6].

RALPH(Rapidly Adapting Lateral Position Handler)[7]와 ALVINN(Autonomous Land Vehicle In a Neural Network)[8] 시스템은 미국의 카네기 멜론대학에서 개발되어졌고 ARCADE(Automated Road Curvature And Direction Estimation)[9]과 MOSFET(Michigan Off-road Fusing Experimental Test-bed)[10]은 미시간 대학에서 개발되었다. GOLD(Generic Obstacle and Lane Detection)[11]와 ROMA(Road Marking Analysis)[12] 같은 차량 안내 시스템은 유럽에서 개발되어졌다.

차선 검출은 AGVs(Autonomous Guided Vehicles)[13], ADS(Autonomous Driving System)[14] 및 운전자를 보조하기 위한 수단으로 많은 연구가 진행되고 있다. 차선 검출 분야는 기타의 다른 영상 기반 도로 검지 시스템과 같이 다양한 도로 환경 및 파라미터에 매우 민감한 분야이다. 또한 차량이 운행하는 중에 실시간으로 검색되어야 하기 때문에 그 처리 시간을 단축시키기 위해 PAPRICA(Parallel Processor for Image Checking and Analysis)[15]같은 특별한 목적의 병렬 구조를 가진 시스템을 사용하기도 한다.

차량 안내 시스템에서의 영상 기반 차선 경계 위치 선정은 차선 검출과 차선 탐색으로 구분된다. 차선 검출은 차선 위치에 관한 많은 지식 없이 차선 경계를 정하는 것이고, 차선 추적은 도로의 기하 도형적 외형의 기존 모델을 이용하여 차선을 추적한다. 차선 검출에 대한 기존의 연구는 다음과 같다. LOIS[16] 차선 검출 알고리즘은 영상의 명암의 변화 정보를 이용한 변형 가능한 템플릿 기법을 사용한다. 이것의 문제점은 너무 많은 연산량에 있다. GOLD 시스템은 IPM을 이용하여 차선을 검출하고, 실시간 성능을 가지기 위해 전용 병렬 프로세서인 PAPRICA상에서 구현하였다. GOLD 시스템의 단점은 도로 상태에 매우 민감하다는 것이다. 만약 도로가 기울기를 가지고 있을 때나 차선이 선명하지 않을 경우 차선검출은 실패한다[17]. RALPH 시스템 역시 도로 표면을 재구성하는 IPM을 사용한다. 이 시스템은 낮은 해상도(32 X 30)의 영상을 생성하기 위한 3D 평면의 복구, 도로 곡률의 결정 및 차선 중앙에 대한 상대적인 차량의 오프셋을 결정한다. 이 시스템의 문제점은 차선 파라미터를 정확히 평가하기에는 너무나 낮은 해상도의 이미지를 사용한다는

것과 경사진 도로 평면 또는 도로에 장애물이 있는 경우 차선 검출에 실패한다는 것이다.

본 논문에서는 차량 안내 시스템에 적용 가능한 차선 검출 방법을 제안한다. 차량에 장착된 하나의 카메라를 이용하여 320 x 240 의 흑백 영상을 획득하여, 전용의 이미지 프로세서를 이용하지 않고 일반적인 저가의 RISC 프로세서인 ARM9를 이용하여 WinCE 상에서 차선을 검출하게 된다.

본 논문은 다음과 같이 구성 된다. 2장에서는 차선 검출을 위한 영상 처리 기법들을 소개하고 3장에서는 원근 효과 제거(IPM, Inverse Perspective Mapping) 방법과 4장에서는 제안하는 적응형 원근 효과 제거(AIPM, Adaptive Inverse Perspective Mapping) 방법에 대해 설명한다. 5장에서는 본 논문에서 알고리즘을 평가한 하드웨어 구성 및 평가 결과를 보인다. 마지막으로 6장에서는 결론을 맺는다.

제 2 장 차선 검출을 위한 영상 처리 기법

제 1 절 히스토그램을 이용한 이진화 처리

Gray-level 의 흑백 영상은 0 ~ 0xff 까지의 픽셀 값을 가지고 있고, 차선은 0xff 근처의 값을 가지는 흰색으로 표시된다. 그렇기 때문에 하나의 영상에서 차선에 해당하는 부분을 찾아내기 위해서는 차선이라고 생각되어지는 부분의 픽셀을 제외한 나머지 픽셀들을 0x00으로 만드는 작업이 필요하다. 따라서 획득한 영상을 이진화(Binarization) 시키는 방법이 필요하다. 즉, 이진화란 특정 픽셀 값 이상의 화소들은 0xff(white)로, 그 이하의 픽셀들은 0x00(black)으로 값을 할당함으로써 검정 (혹은 흰색) 배경화면에 원하는 영상을 흰색(혹은 검정)으로 나타나도록 하는 기법이다. 이진화 처리를 하기 위해서는 차선과 배경을 구분하는 일정한 기준이 되는 값이 있어야 하는데 우리는 이러한 기준값을 임계값(threshold value)이라고 한다.

$$O(x, y) = \begin{cases} 255 & \text{if } p(x, y) \geq Th \\ 0 & \text{if } p(x, y) < Th \end{cases} \quad (2-1)$$

여기에서 $O(x, y)$ 는 이진화 후의 영상의 (x, y) 에 있는 픽셀 값을, Th 는 임계값(threshold)을 나타낸다. 앞의 식에서 임계값을 기준으로 영상을 반대로 처리하면 반전된 이진화 영상을 얻을 수 있다.

$$R(x, y) = \begin{cases} 0 & \text{if } p(x, y) \geq Th \\ 255 & \text{if } p(x, y) < Th \end{cases} \quad (2-2)$$

$R(x, y)$ 는 영상의 (x, y) 에서의 픽셀 값이다.

<그림 2-1>과 <그림 2-2>는 임계값(Th)에 따른 영상의 변화를 보여준다.

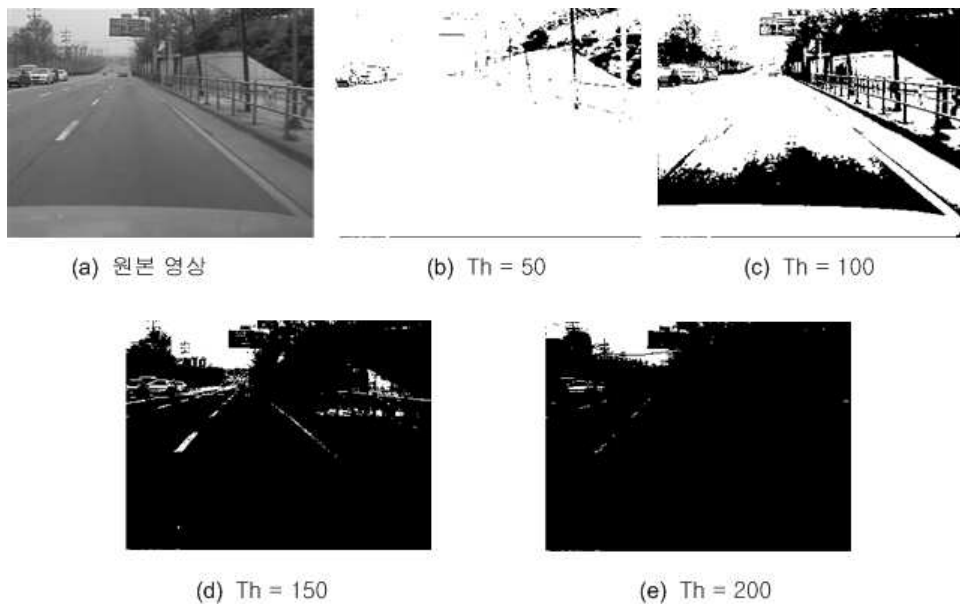


그림 2-1. 임계값에 따른 이진화 $O(x,y)$ 영상

Fig 2-1. Binary image according to threshold value

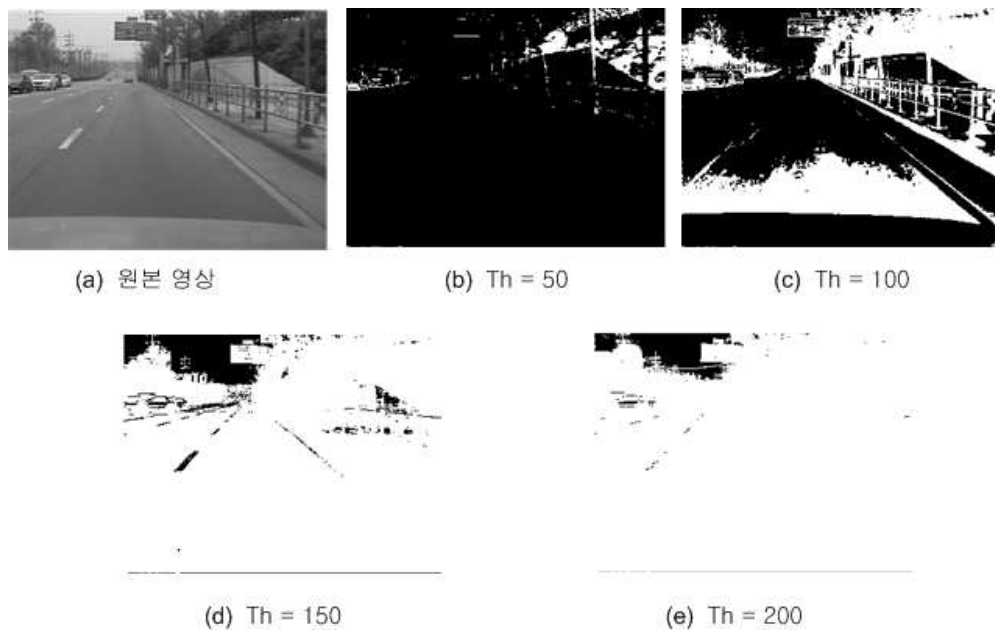


그림 2-2. 임계값에 따른 이진화 $R(x,y)$ 영상

Fig 2-2. Binary image according to threshold value

이진화 영상은 그 기준값이 되는 임계값을 결정하기 위해 히스토그램을 이용할 수 있다. 히스토그램이란 영상의 밝기 값에 대한 분포를 보여주는 그래프로 영상분석을 위한 중요한 도구이다. 입력되는 영상의 히스토그램을 분석함으로써 영상의 밝기 구성, 명암의 대비 등에 대한 정보를 알 수 있으며 이러한 분석정보를 이용하여 영상개선 및 화질 향상을 위한 도구로 사용할 수 있다. 즉, 히스토그램은 영상의 밝기값을 수평축으로 하고 수평축의 밝기값에 대응되는 크기를 가진 픽셀수가 영상 안에 몇 개나 되는지 나타내는 빈도수를 수직축으로 해서 만든 그래프이다. 따라서, 흑백영상의 경우 수평축은 0 ~ 255의 범위 값을 가지며 수직축은 영상의 크기와 밝기의 분포에 따라 달라진다[19, 22].

해당 영상의 각 픽셀의 픽셀 값(0~255)을 읽어서 같은 픽셀 값을 가지는 픽셀수를 카운트한다. <그림 2-3>은 카운트한 값을 나타내는 히스토그램을 표시하고 있다. 해당 프레임은 픽셀 값 i 에서 막대그래프의 길이만큼의 화소수를 가지고 있음을 표시한다.

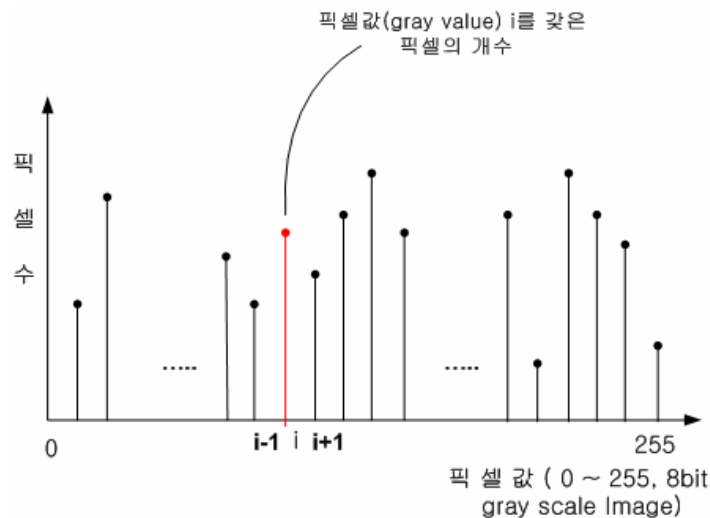


그림 2-3. 히스토그램의 표시

Fig 2-3. Display of histogram

이진화 후의 히스토그램은 픽셀 값이 0 혹은 255 밖에 존재하지 않는다. 또한 굴곡이 심하지 않은 도로의 차선의 경우, 차선은 영상의 가로 방향 특정위치에서 세로방향으로 존재한다. 그런 영상에서 차선을 찾기 위해 <그림 2-4>와 같은 히스토그램을 이용할 수 있다. 히스토그램의 가로축(X)은 영상의 가로축에 해당하고, 특정 가로축에 해당하는 픽셀수가 히스토그램의 Y 축에 해당한다.

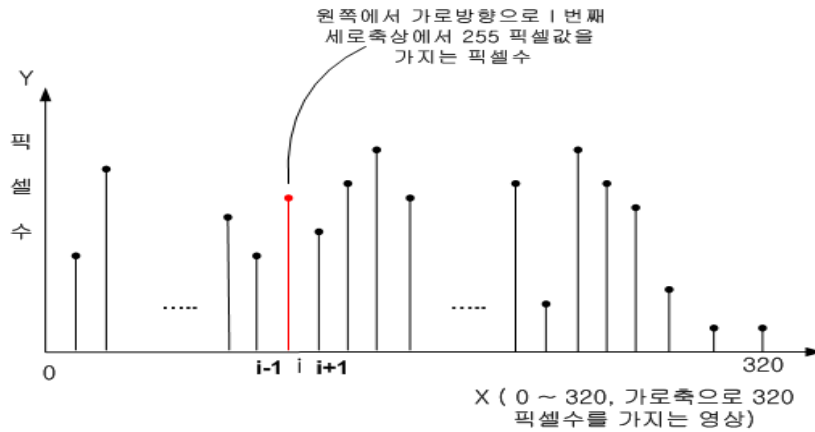


그림 2-4. 이진 영상을 위한 세로 방향 히스토그램

Fig 2-4. Row direction histogram for binary image

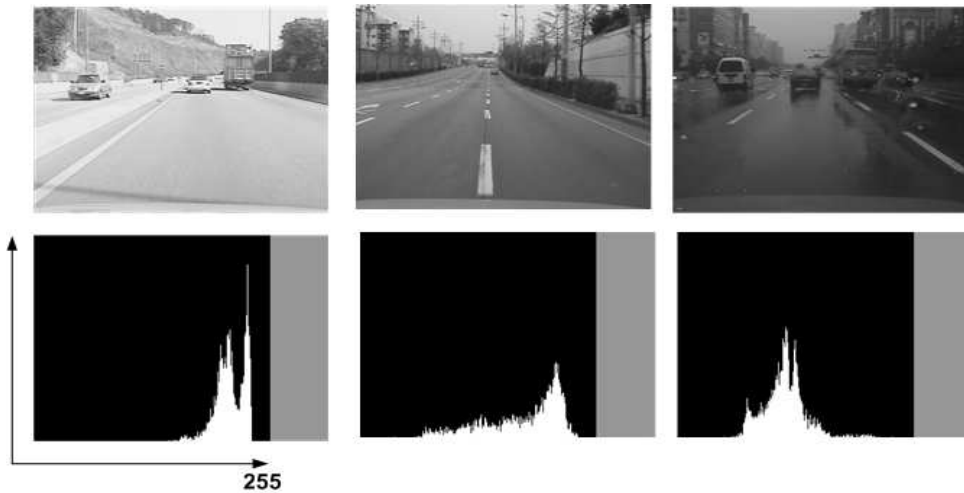


그림 2-5. 날씨(조도)에 따른 <그림 2-3> 방식의 히스토그램

Fig 2-5. Histogram according to weather by <Fig 2-3> method

제 2 절 에지 검출

에지(Edge)라는 것은 물체의 윤곽을 나타내는 것으로 강도의 차이에 의해 발생한다. 주로 물체의 경계에서 발생하기 때문에 에지 검출은 영상에서 다른 물체들을 분리하고자 할 때 사용한다. 이러한 강도의 대비에 의한 에지 검출은 도로 영상의 차선 검출에서 유용하게 사용될 수 있다. 일반적으로 도로 영상에서 차선은 흰색, 노란색, 파란색으로 표시되며, 반면에 도로는 상대적으로 어두운 색으로 표현되기 때문에 도로와는 색의 강도 차이가 나타난다. 따라서, 영상에서 정확한 에지를 추출할수록 도로 영상에서 정확한 차선을 검출해 낼 수 있다[18].

Edge의 추출은 그레디언트(Gradient type, 1차미분)와 라플라시안(Laplacian type, 2차미분)을 통해 추출해 낼 수 있으며, 오퍼레이터(Operator)에 따라 Roberts, Prewitt, Sobel 등이 있다[19, 22].

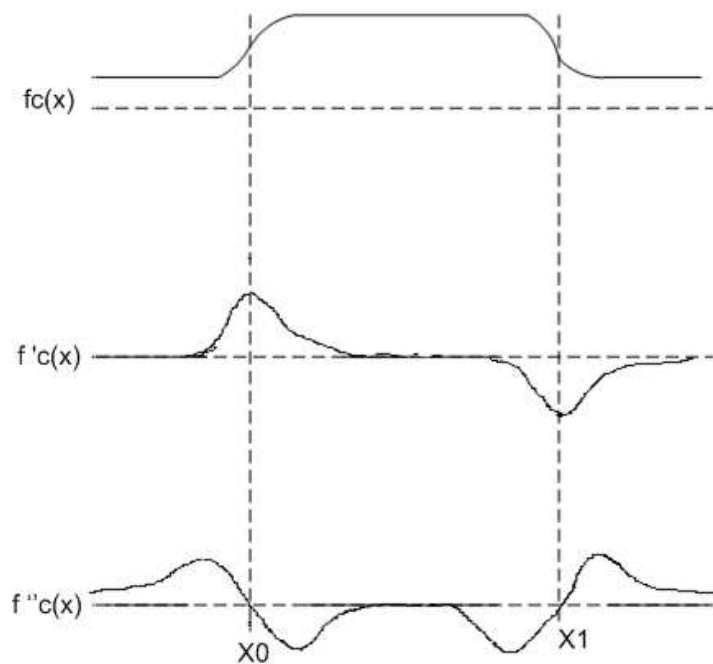


그림 2-6. 연속적인 1차원 평면에서의 에지 검출

Fig 2-6. Edge Detection in the one-dimension continuous case

그레디언트(gradient)는 영상 $f(x, y)$ 의 위치 (x, y) 점에서의 픽셀 값의 기울기를 나타내는 1차 미분의 값이다. 영상 $f(x, y)$ 의 위치 (x, y) 점에서의 기울기는 다음과 같은 벡터이다[19].

$$\nabla f(x, y) = \begin{bmatrix} f_x \\ f_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2-3)$$

$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} i_x + \frac{\partial f(x, y)}{\partial y} i_y \quad (2-4)$$

$\nabla f(x, y)$ 는 (x, y) 에서의 기울기 벡터이고, i_x, i_y 는 x 와 y 방향의 단위 벡터(Unit vector)이다. 위치 (x, y) 점에서의 기울기 벡터는 $f(x, y)$ 의 최대 변화율 방향을 가리킨다. f_x 는 x 방향의 미분, f_y 는 y 방향의 미분으로써 f_x 와 f_y 는 디지털 영상에서는 다음과 같이 구해진다.

$$f_x = f(x+1, y) - f(x, y) \quad (2-5)$$

$$f_y = f(x, y+1) - f(x, y) \quad (2-6)$$

에지 검출에서 중요한 값은 이 벡터의 크기이며 이를 일반적으로 간단히 그레디언트(gradient)라고 하며 다음과 같이 계산된다.

$$G = \sqrt{f_x^2 + f_y^2} \quad (2-7)$$

이 값은 그레디언트 방향으로 단위 길이당 $f(x, y)$ 의 최대 증가율과 같다. 식 (2-7)은 다음과 같이 근사화 할 수 있다.

$$G \approx |f_x| + |f_y| \quad (2-8)$$

실제에서는 기울기를 식 (2-8)처럼 근사화하여 많이 사용한다. 이 식을 이용할 경우 윤곽이 출력되는 형태가 방향에 따라 다소 변화하지만 연산 속도가 빠르고 하드웨어로 구현하기 쉽기 때문에 많이 사용한다.

기울기 벡터의 방향 역시 중요한 양이다. (x, y) 에서의 x 축을 기준으로 하는 벡터 $\nabla f(x, y)$ 의 방향각은 $\alpha(x, y)$ 는 다음과 같다.

$$\alpha(x, y) = \tan^{-1}\left(\frac{f_x}{f_y}\right) \quad (2-9)$$

그레디언트(gradient)를 다시 미분한 2차 미분을 라플라시안(Laplacian)이라 하며 다음과 같이 계산된다.

$$\nabla^2 f(x, y) = \nabla \cdot \nabla f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (2-10)$$

디지털 영상에서는 그레디언트와 라플라시안을 구하기 위해 인접 화소끼리의 연산을 수행하기 위해 미분 오퍼레이터(Operator)들을 사용한다. 다음은 대표적인 미분 오퍼레이터들이다[19].

표 2-1. 그레디언트 계산에 사용되는 미분 오퍼레이터

Table 2-1. Operators for gradient

오퍼레이터	Roberts	Prewitt	Sobel
fx	0 0 0	-1 0 1	-1 0 1
	0 1 0	-1 0 1	-2 0 2
	0 0 -1	-1 0 1	-1 0 1
fy	0 0 0	-1 -1 -1	-1 -2 -1
	0 0 1	0 0 0	0 0 0
	0 -1 0	1 1 1	1 2 1

표 2-2. 라플라시안 계산에 사용되는 미분 오퍼레이터

Table 2-2. Operators for laplacian

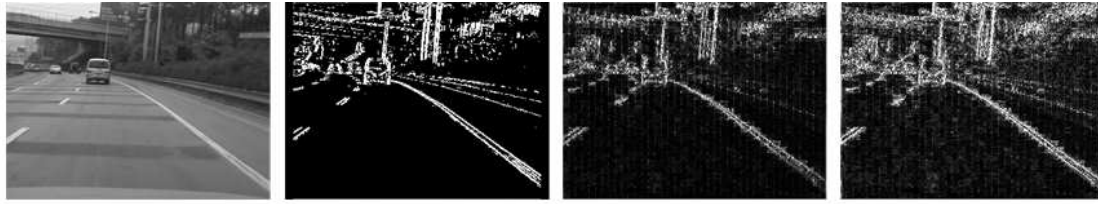
오퍼레이터	라플라시안 1	라플라시안 2	라플라시안 3
fx	0 -1 0	-1 -1 -1	1 -2 1
	-1 4 -1	-1 8 -1	-2 4 -2
	0 -1 0	-1 -1 -1	1 -2 1

연속 확률변수 X의 확률밀도함수 f(x) 가

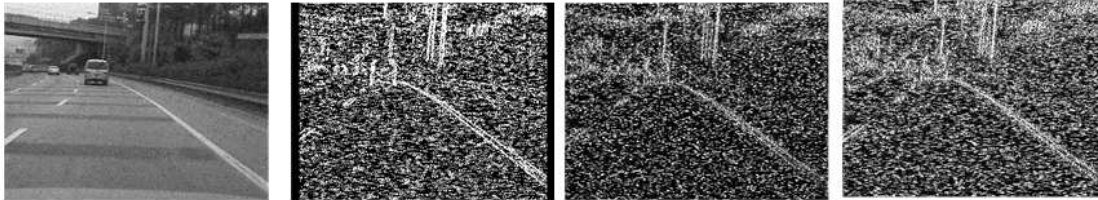
$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}} \quad (2-11)$$

로 주어질 때, 이 X의 분포를 정규분포 또는 가우스 분포라 하고, X를 정규 확률변수라 한다. 이 경우 X의 평균은 m 이고 분산은 σ^2 이다.

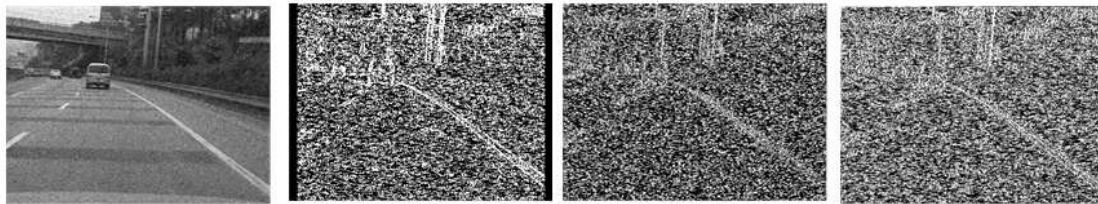
<그림 2-7>은 가우시안 분포를 따르는 평균 0, σ (표준 편차) = 0, 10 , 20, 30, 50 인 영상을 기반으로 그레디언트 연산에 사용되는 Sobel 오퍼레이터와 라플라시안 연산에 사용되는 라플라시안 1, 2 오퍼레이터에 대한 에지 검출 영상이다[19-21].



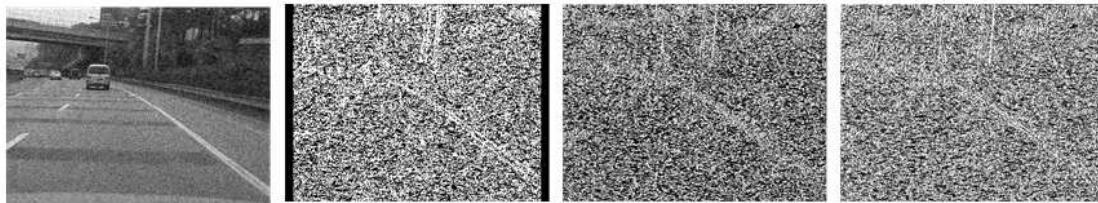
(a.1) $\sigma = 0$



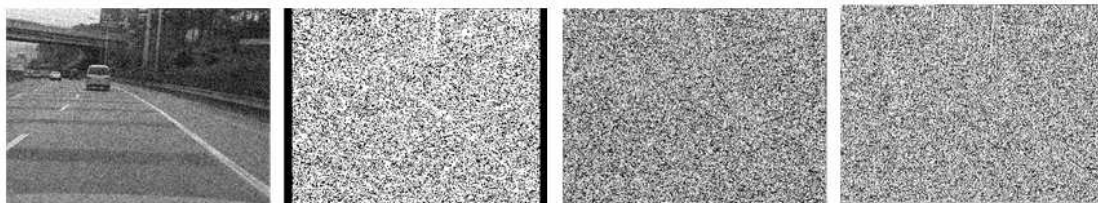
(a.2) $\sigma = 10$



(a.3) $\sigma = 20$



(a.4) $\sigma = 30$



(a.4) $\sigma = 50$

(a) 표준 편차에 따른 노이즈가
삽입된 영상

(b) Sobel

(c) Laplacian 1

(d) Laplacian 2

그림 2-7. 가우시안 분포에 따라 잡음을 삽입한 영상에 대한 에지 검출

Fig 2-7. Edge Detection about image added noise by gaussian distribution

(Variable standard deviation and zero mean)

제 3 절 컬러 영상의 흑백 영상으로의 변환

본 논문에서는 알고리즘 검증 위해 PC 기반을 이용한다. 검증을 위한 도로 영상은 차량에 장착된 캠코더로 촬영한 컬러 영상을 초당 10 프레임으로 샘플링한다. 그리고 다시 샘플링 된 컬러 영상 사진을 320 x 240의 흑백 영상 사진으로 바꾸는 작업을 한다. 그러기 위해서는 컬러 영상을 흑백 영상으로 바꾸는 방법이 필요하다. 컬러 영상을 흑백영상으로 바꾸기 위해서는 식 (2-12) 변환 공식을 사용한다.

$$Y = krR + kgG + kbB \quad (2-12)$$

ITU-R recommendation BT 601 (CCIR601)에서 정의한 kb 와 kr 값은 각각 0.114 0.299 이다. 컬러 영상에 대한 YCbCr 과 RGB로의 변환 공식은 다음과 같다.

$$Y = 0.229R + 0.587G + 0.114B \quad (2-13)$$

$$Cb = 0.564 (B - Y)$$

$$Cr = 0.713 (R - Y)$$

$$R = Y + 1.402Cr \quad (2-14)$$

$$G = Y - 0.344Cb - 0.714Cr$$

$$B = Y + 1.772Cb$$

R은 영상에서 적색(Red) 성분이고, G는 녹색(Green), B는 파랑색(Blue) 성분을 가르킨다. 식 (2-13)의 Y 성분은 컬러 영상에서의 흑백 영상으로 변환할 때 쓰이는 흑백 영상의 픽셀값이다. <그림 2-8>은 컬러 도로 영상을 YCrCb로 변환한 영상이다.

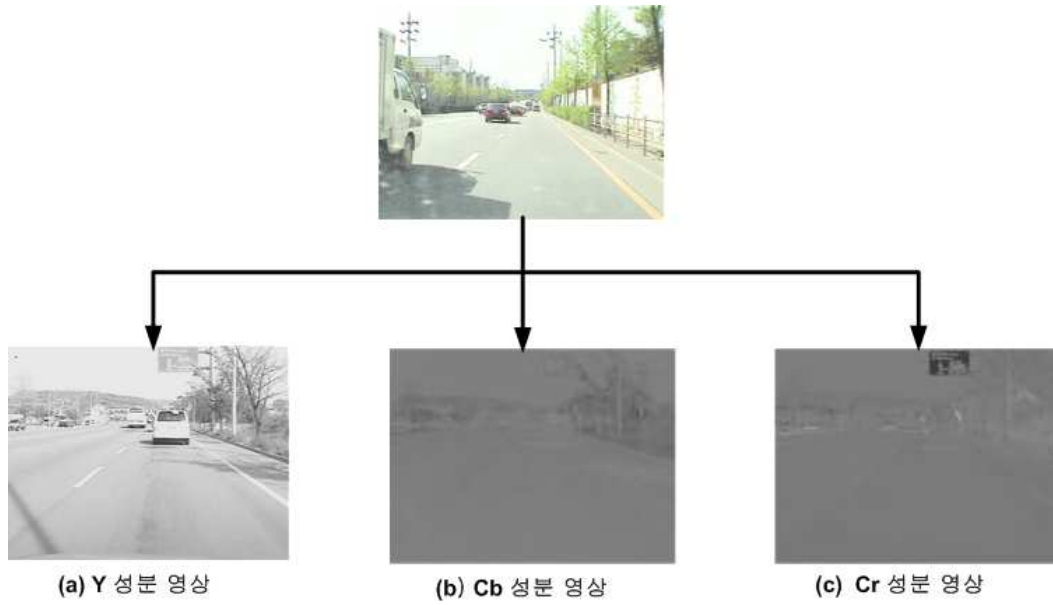


그림 2-8. 컬러 영상을 YCbCr로 변환한 영상

Fig 2-8. Conversion of color image into YCbCr image

제 3 장 원근 효과 제거 방법

제 1 절 카메라 모델을 이용한 원근 효과 제거

일반적인 도로영상은 2D 상에서 실선 혹은 점선으로 구성되어 있다. 그러나 차선이 존재하는 도로는 3D 평면에 존재하기 때문에 차선을 검출하기 위한 방법으로 원근 효과 제거 방법을 사용하게 된다. 카메라 영상은 기본적으로 원근 효과를 가지기 때문에 실제 3D 공간에서 가지고 있는 정보를 온전히 가지고 있지 못하게 된다.[23]

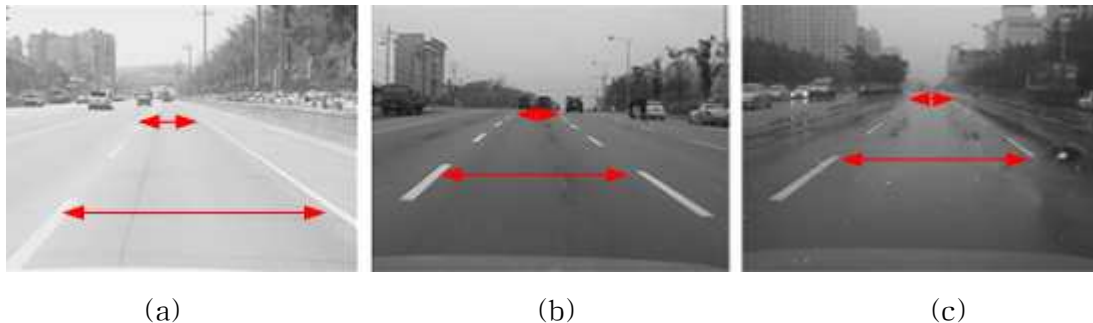


그림 3-1. 원근 효과를 가지는 도로 영상

Fig 3-1. Traffic Image which have perspective effect

실제 공간(3D)에서 평행한 전형적인 도로영상은 점선 혹은 실선에 의해 표시되는 차선에 의해 도로평면이 구분된다. 3D 실제 공간에서 차선은 같은 길이와 폭을 갖게 된다.[4, 11]

그러나 <그림 3-1>에서와 같이 2D 영상 평면에서의 각 픽셀은 원근 효과로 인해 실제 공간과는 다른 영역을 표시한다. 2D 영상 평면의 아랫부분의 픽셀은 3D 실제 공간에서 더 작은 영역이 되며, 윗부분 픽셀은 더 넓은 영역이 되어 도로가 가지는 고유를 파라미터를 추출해 내기가 어렵다.

즉, 원근 효과로 인해 차선 및 도로 파라미터를 정확히 평가하는데 어려움이 있

다. 원근 효과를 제거하기 위해, 카메라 위치(Position), 좌표(coordination), 광학적 특징(Optics)과 같은 특별한 획득 조건을 이용해서 3D 도로 평면을 복구하기 위해 원근 효과 제거 방법(Inverse Perspective Mapping)을 사용한다.[4, 11, 17]

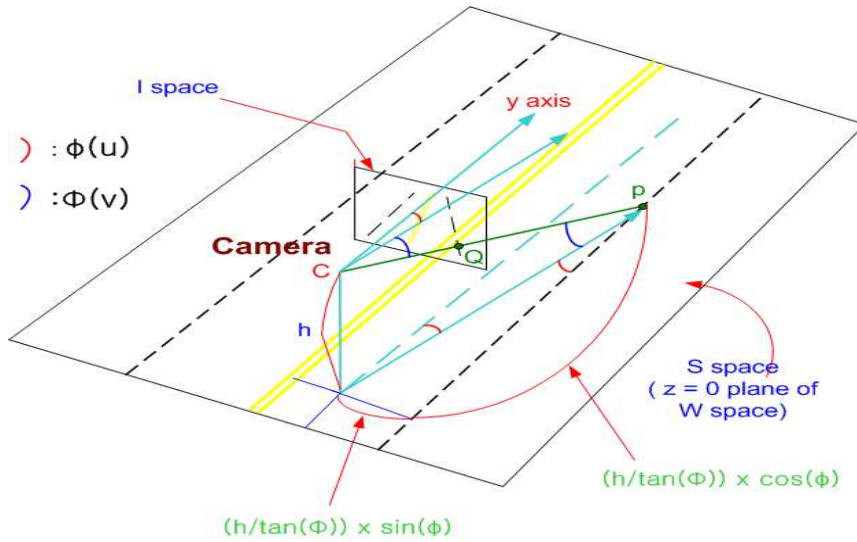


그림 3-2. 원근 효과 제거를 위한 실제 도로와 카메라
특성과의 상관관계

Fig 3-2. Relationship of 3D real world and camera
characters for Inverse Perspective Mapping

3차원 평면 (X, Y, Z) 에서 2차원 평면 (u, v) 로의 매핑은 다음과 같다.[22]

$$(u, v) \in I \rightarrow (x, y, 0) \in S$$

(3-1)

$$(x, y, 0) \in S \rightarrow (u, v) \in I \quad (3-2)$$

두 매핑은 카메라 외적 파라미터 (카메라 position 과 coordination)와 내적 파라미터 (초점거리, 카메라 해상도)에 의존한다. 본 논문의 영상 처리 과정에서 핵심이 되는 부

분은 원근 효과를 제거하여 얻어지는 차선에 관한 파라미터 획득이다. 그리고 원근 효과를 제거하기 위해서는 카메라의 고유 성질이나 설치 시 정해지는 각 변수들의 영향을 많이 받는다. 그렇게 때문에 그 상관관계를 파악하는 것이 중요하다.

<그림 3-3> 은 카메라와 영상 이미지와의 관계를 표현한다.

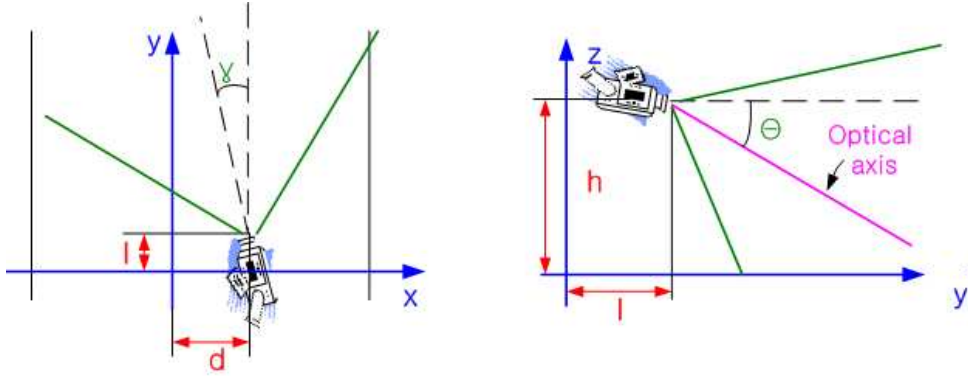


그림 3-3. 카메라의 위치 설정에 관계된 변수

Fig 3-3. Variables relevant to camera's position setting.

$$u = f * \{ (X - X_0) \cos \gamma + (Y - Y_0) \sin \gamma - r_1 \} /$$

$$\{ - (X - X_0) \sin \gamma \sin (90 + \Theta) + (Y - Y_0) \cos \gamma \sin (90 + \Theta) - (Z - Z_0) \cos (90 + \Theta) + r_3 + f \}$$

(3-3)

$$v = f * \{ - (X - X_0) \sin \gamma \cos (90 + \Theta) + (Y - Y_0) \cos \gamma \cos (90 + \Theta) + (Z - Z_0) \sin (90 + \Theta) - r_2 \} /$$

$$\{ - (X - X_0) \sin \gamma \sin (90 + \Theta) + (Y - Y_0) \cos \gamma \sin (90 + \Theta) - (Z - Z_0) \cos (90 + \Theta) + r_3 + f \}$$

(3-4)

식 (3-3)와 식 (3-4)에서 각 변수는 다음과 같다.[11, 22]

X, Y, Z : 3D 평면에서의 좌표.

X₀, Y₀, Z₀ : 원점(0,0,0) 으로부터의 거리.

u, v : 2D 영상평면에서의 좌표.

f : 렌즈의 초점거리.

γ : 카메라가 좌우로 기울어진 각도.

θ : 상하로 기울어진 각도.

$r1, r2, r3$ 은 카메라 짐벌(gimbal) 중심에서 영상 평면 중심까지의 오프셋을 나타낸다.

초기값을 Z_0 를 h 로 설정하고, $r1, r2, r3, X_0, Y_0, \gamma$ 를 0 으로 놓으면 식 (3-3)과 식 (3-4)은 다음과 같이 바꾸어 쓸 수 있다.

$$u = \frac{fx}{y\cos\theta - h\sin\theta}, \quad v = \frac{f(h\cos\theta + y\sin\theta)}{y\cos\theta - h\sin\theta} \quad (3-5)$$

$$x = \frac{uh}{v\cos\theta - f\sin\theta}, \quad y = \frac{h(f\cos\theta + v\sin\theta)}{v\cos\theta - h\sin\theta}, \quad z = 0 \quad (3-6)$$

카메라의 렌즈 파라미터와 해상도의 관계는 다음과 같다.

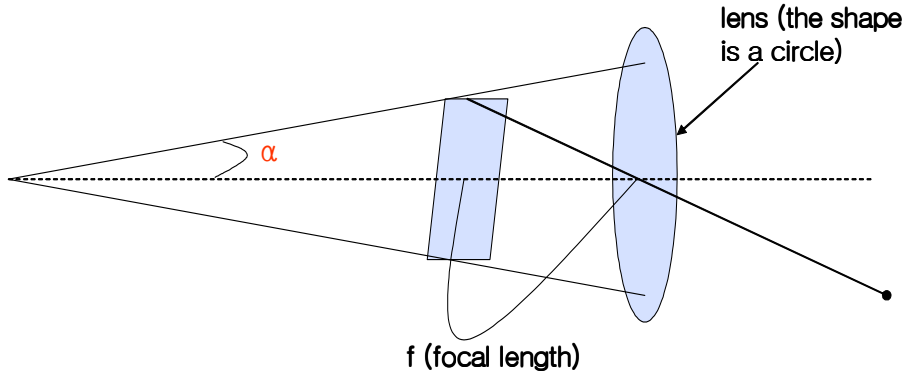


그림 3-4. 원근 효과 제거에 사용하게 될 렌즈의 파라미터

Fig 3-4. Lens parameters using for IPM

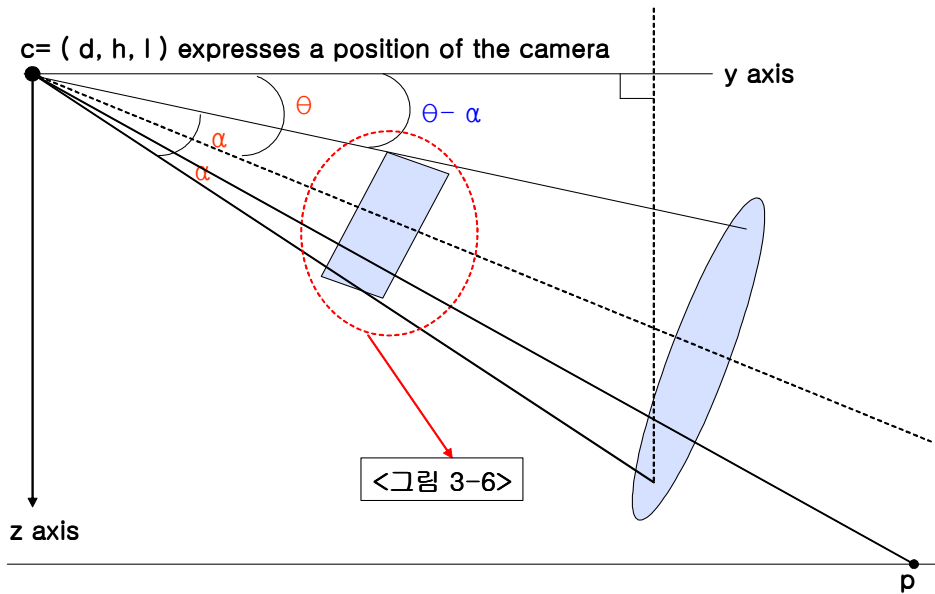


그림 3-5. 카메라 파라미터(θ) 와 Aperture (2α)의 관계

Fig 3-5. Relationship between a camera parameter(θ) and an aperture (2α)

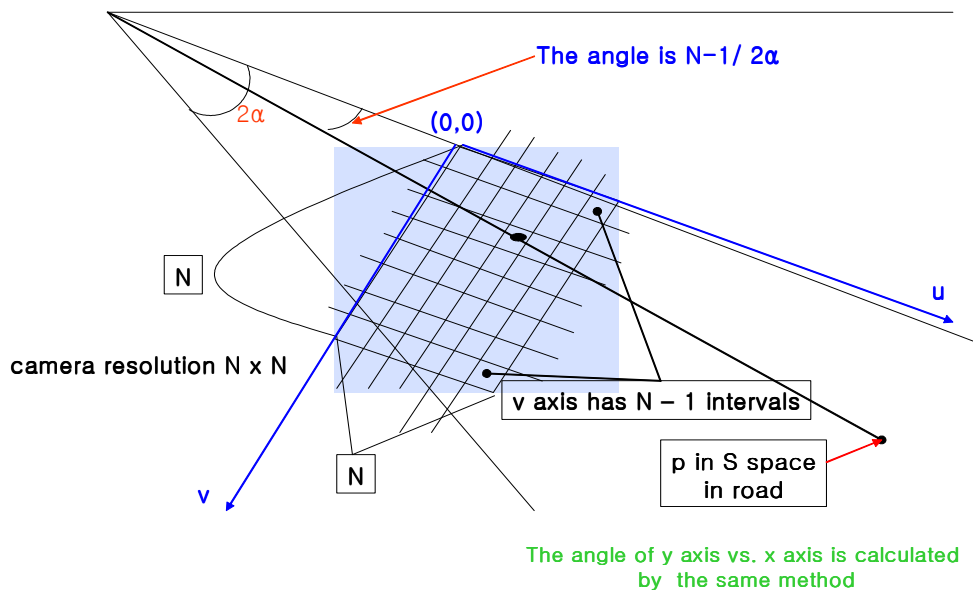


그림 3-6. 영상 픽셀 데이터와 3D 도로평면과의 관계

Fig 3-6. Relationship between camera pixels and 3D traffic road image.

카메라가 도로를 향해서 찍는다고 할 때 도로 영상의 각 픽셀 데이터는 다음의 맵핑 방정식에 의해 실제 도로상의 위치를 찾는데 사용할 수 있다.

$$\phi(v) = (\theta - \alpha) + v \frac{2\alpha}{N-1}, \psi(u) = (\gamma - \alpha) + u \frac{2\alpha}{N-1} \quad (3-7)$$

$$x(u,v) = \frac{h \sin(\psi)}{\tan(\phi)} + d, y(u,v) = \frac{h \cos(\psi)}{\tan(\phi)} + l, z(u,v) = 0 \quad (3-8)$$

즉, (u, v) 의 2D 영상 도로 평면에서 픽셀 값을 이용하여 3D 평면으로의 재 매핑이 가능하다.

제 2 절 원근 효과 제거를 위한 각 파라미터의 영향

이 원근 효과 제거 방법을 사용하면 원 영상 데이터에서는 사용할 수 없었던 도로 파라미터(차선사이의 고유 간격, 차선의 두께, 도로의 너비)등을 추출해 낼 수 있다. 하지만 원근 효과 제거는 카메라 및 도로의 상태(도로의 수평 여부)에 매우 민감하다. 따라서 각 변수에 대한 고찰이 필요하다. 카메라 고유의 특성과 관련된 원근 효과 제거에 사용되는 변수 중 하나는 렌즈의 구경(Aperture)이다. 하지만 이 변수는 설치 시 영향을 주지 않기 때문에 알고리즘 구현 시 외에 고려할 필요가 없다. 또한 'l' 변수 역시 거의 무시해도 좋을 만큼의 작은 값이 때문에 설치 시 고려할 필요가 없으며, 실제로 원근 효과 제거를 위한 알고리즘 구현 시에도 무시된다. 'h' 는 설치 시 차량의 크기에 따라 혹은 설치 위치에 따라 바뀔 수 있는 변수이며, 거리 정보를 획득하는데 있어서 가장 큰 영향을 미친다. 그러나 거리가 미터(meter) 단위로 계산되므로 작은 오차는 큰 영향을 미치지 않는다. 'd' 는 차량과 도로와의 좌우 거리 정보와 관련이 되어 있고 주행 중인 좌우 차선과의 거리 계산 시 잘못 설정된 'd' 값은 오차로 작용한다. 'y' 는 차선의 수평 파라미터에 관련된 정보이므로 원근 효과를 제거할 경우 영상이 좌우로 기울어진 결과를 보인다. 하지만 차선 검출 시 필요로 하는 파라미

터 추출에는 많은 영향을 주지 않는다. 'θ'는 원근 효과 제거 시 가장 큰 영향을 주므로 설치 시 정해지는 θ 값을 정확히 알아야 한다. 이 값이 정확하지 않을 경우 차선 검출에 있어 가장 큰 에러를 발생 시킬 수 있다.

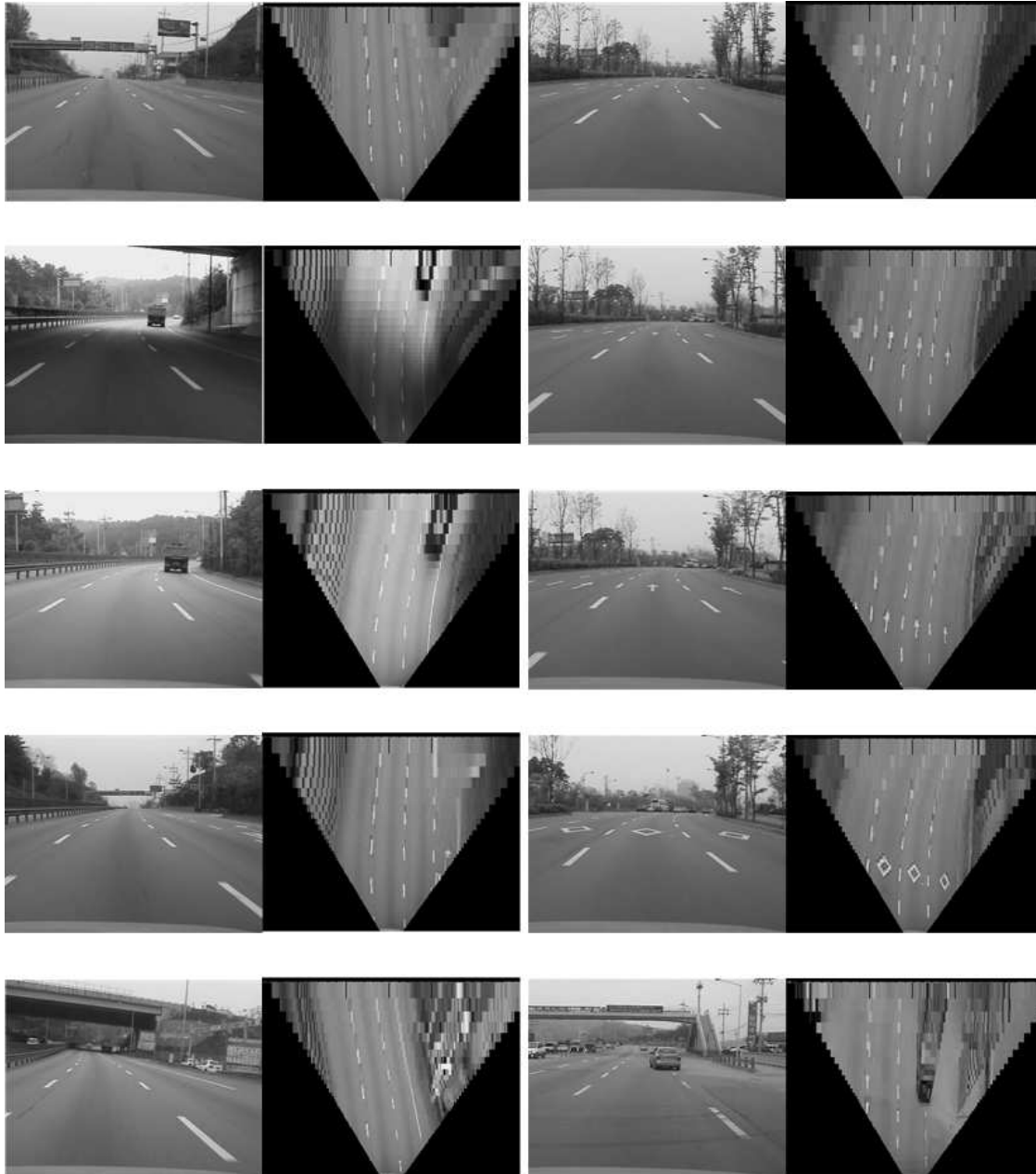


그림 3-7. 원근 효과 제거 후의 도로 영상

Fig 3-7. An Example of Inverse Perspective Mapping(IPM)

제 4 장 적응형 원근 효과 제거 방식 제안

일반적인 원근 효과 제거(Inverse Perspective Mapping)에 의한 단점은 과도한 처리시간과 평행하지 않은 도로에 의한 왜곡이다. 구현된 알고리즘은 타겟 보드 상에서 초당 10 프레임을 처리할 수가 있어야 한다. 하지만 일반적인 원근 효과 제거(IPM) 방식을 사용할 경우 IPM 연산 시 사용되는 삼각 함수 계산을 위한 플로팅 포인트 연산으로 인해 초당 10프레임의 연산을 수행할 수가 없다. 또한 고정된 파라미터를 사용할 경우 평행하지 않은 도로를 대상으로 원근 효과 제거를 수행할 경우 IPM 후의 결과 영상이 왜곡된 결과가 나오게 된다. 원근 효과 제거 방법은 기본적으로 도로가 수평한 상태에 놓여 있다는 것을 전제로 한다. 즉, $Z = 0$ 은 상태를 기본으로 하여 매핑을 시작한다. 그러나 이상적인 도로가 아닌 실제 도로는 완전한 평면이 아닌, 도로에 따라 수시로 바뀌는 경사도를 가지고 있다. 이러한 고정 경사도가 아닌 가변적인 경사도를 가지고 있는 실제 도로에서는 위와 같은 매핑 방법은 가변 폭이 크기 때문에 정확한 도로 파라미터를 추출해 내기가 어렵다. 따라서 경사도에 따른 도로 파라미터의 가변 폭을 축소화 시키는 알고리즘이 필요하다.

따라서 본 논문에서는 위와 같은 단점을 보완하기 위해 적응형 원근 효과 제거 방법(AIPM, Adaptive Perspective Mapping)을 제안한다.

우선 특별한 이미지 프로세서를 사용하지 않고 일반적인 ARM9을 사용하여 IPM에 의한 시간 단축을 시키기 위해 영상의 모든 픽셀을 IPM의 대상으로 하는 것이 아니라, 우선 차선(Traffic Lane)이라 생각되어지는 픽셀 값만 IPM을 이용함으로써 그 처리시간을 단축시킬 수 있다. 일반적인 원근 효과 제거를 위한 매핑 순서는 매핑 후 이진화 처리이다. 하지만 이 경우 모든 픽셀에 대해 매핑이 이루어지기 때문에 영상 처리 시간이 오래 걸릴 수밖에 없다. 이러한 단점을 해결하기 위해 우선 이진화를 처리하고, 차선이 될 가능성이 많은 차선이라 생각되어지는 밝은 픽셀(픽셀 값 0xff 에 근사한 값, 흰색에 가까운 픽셀)들만 맵핑함으로써 영상 처리 시간을 단축시킬 수 있다.

그러기 위해서는 차선이 될 가능성이 있는 밝은 픽셀들을 선별하여야만 한다. 그렇기 위해서는 날씨(조도 및 잡음)의 영향에 민감하지 않은 차선 값에 대한 임계값

(Threshold value)을 선정할 수 있어야 한다.



그림 4-1. 날씨에 따른 도로 영상((a) 맑은 날, (b) 흐린 날, (c)비오는 날)

Fig 4-1. Traffic Image according to weather ((a) Fine day, (b) Cloudy day, (c) Rainy day)

적응형 원근 효과 제거(AIPM, Adaptive Perspective Mapping)은 다음과 같은 순서로 이루어진다.

제 1 절 소벨 에지 검출

도로 영상은 환경의 영향을 많이 받게 된다. 그 중 하나가 햇빛이 도로에 반사됨으로써 생기는 도로 자체의 명암 변화이다. 그렇게 때문에 차선을 구분하기 위해 사용되는 이진화 처리 기준이 되는 임계값을 찾아내는 작업이 필요하게 된다.

하지만 햇빛의 반사에 의한 도로 자체의 명암이 변한다하더라도 차선과 도로의 명암 차는 분명히 존재한다[4]. 그렇기 때문에 우선 그 명암의 변화가 있는 부분을 찾아내기 위해 소벨 에지 검출(Sobel edge detection) 방법을 사용하여 명암차가 생기는 에지[8]를 찾아낸다.



그림 4-2. 날씨에 따른 도로의 소벨 에지 검출 영상

((a) 맑은 날, (b) 흐린 날, (c)비오는 날)

Fig 4-2. Result of Sobel edge detection according to weather

((a) Fine day, (b) Cloudy day, (c) Rainy day)

제 2 절 임계값 설정을 위한 마스크 영상의 생성

위의 <그림 4-2>영상에서 보듯이 차선은 날씨와 관계없이 도로에 비해 확실한 명암 차이를 가지고 있다. 하지만 도로 이외의 에지 부분도 상당부분 존재한다. 그렇기 때문에 후속 작업이 필요하게 된다.

날씨에 따른 도로 밝기의 영향을 최소화하기 위해 임계값을 정할 때 사용되는 부분을 차선 부분으로 최소화 시킨다. 즉 임계값을 정할 때 사용하게 될 기본 영상 데이터 설정을 에지 부분으로 한정시키게 된다. 그렇게 하기 위해 에지가 생기는 부분을 채우는 작업이 필요하게 된다.

본 논문에서는 에지 부분을 채우기 위해 다음과 같은 필터를 사용한다.

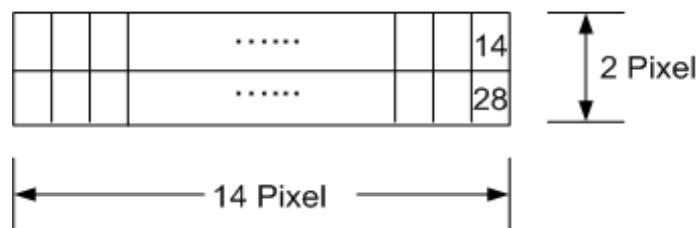


그림 4-3. 에지 부분을 채우기 위한 필터

Fig 4-3. A filter for filling between edges

$$k = \sum PX_i \quad (4-1)$$

where

PXi : 필터에 해당하는 값이 0xff 이면 1이다.

만약 $k > 5$ 면 해당 필터에 해당하는 부분에서 제일 왼쪽에 있는 픽셀과 오른쪽에 있는 픽셀 사이에 있는 값을 0xff 로 만들어서 에지 사이의 공간을 흰색(0xff)으로 채운다.

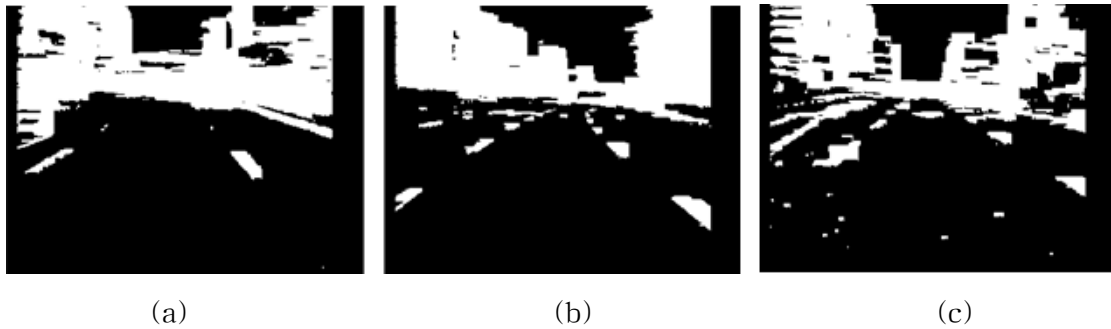


그림 4-4. 필터에 의해 만들어진 마스크 영상

((a) 맑은 날, (b) 흐린 날, (c)비오는 날)

Fig 4-4. A result of the filter ((a) Fine day, (b) Cloudy day, (c) Rainy day)

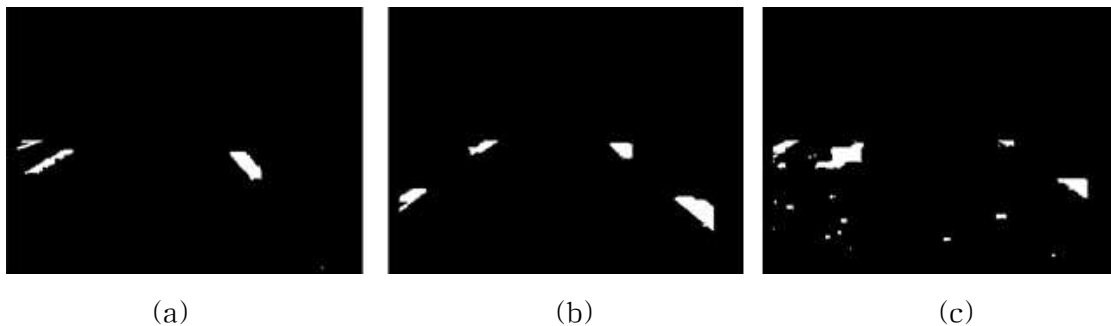


그림 4-5. 도로 부분에 해당되는 부분만 추출한 마스크 영상

((a) 맑은 날, (b) 흐린 날, (c)비오는 날)

Fig 4-5. Mask images under vanish point

((a) Fine day, (b) Cloudy day, (c) Rainy day)

제 3 절 히스토그램에 의한 차선 검출

임계값을 찾기 위한 마지막 방법으로 히스토그램을 이용한다. 이때 <그림 4-5>에서 만든 마스크 영상을 사용한다. 원 영상과 마스크 영상의 교집합에 구하여 히스토그램을 그린다.

$$Ih = Oi \cap Mi \quad (4-2)$$

where

Ih : Mi 도로 영상에서 마스크 영상 이외의 부분이 0x00인 영상

Oi : 임계값을 구하려고 하는 도로 영상

Mi : 마스크 이미지

실험을 통한 통계를 바탕으로 할 때 차선은 도로 화면상에서 약 300개의 픽셀을 가지고 차선 영역을 차지하고 있다. 따라서 픽셀의 명암 값을 이용한 히스토그램을 이용하여 임계값을 정할 수 있다. 히스토그램의 왼쪽부터 픽셀 값을 더하여 그 값이 300이 되는, 즉 차선의 개수에 해당하는 히스토그램 값을 취하여 그 값을 임계값으로 정한다.

차선이 될 수 있는 부분만을 마스크로 정해놓고 그 부분을 대상으로 임계값을 정하기 위한 처리를 함으로써 보다 정확한 임계값을 구할 수 있다.

<그림 4-6>은 원 도로 영상과 마스크 영상의 교집합을 시킨 후 구한 히스토그램을 나타낸다.

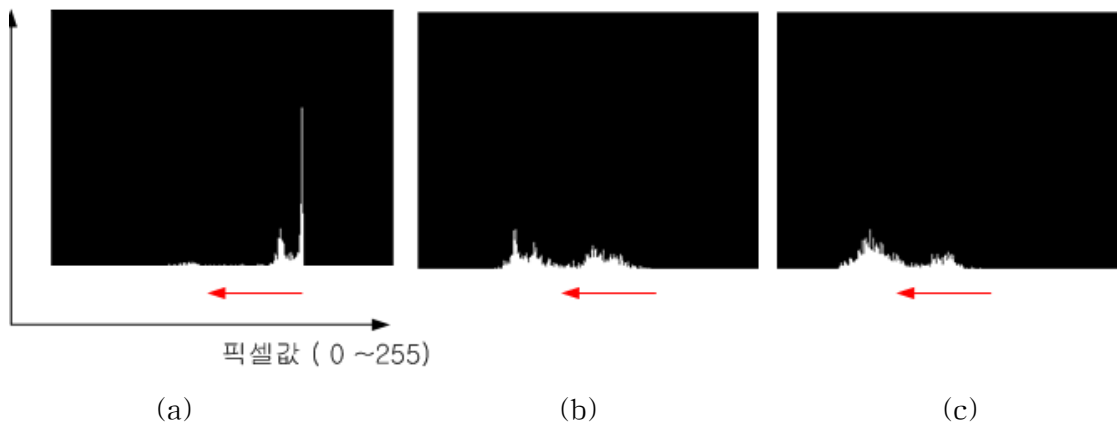
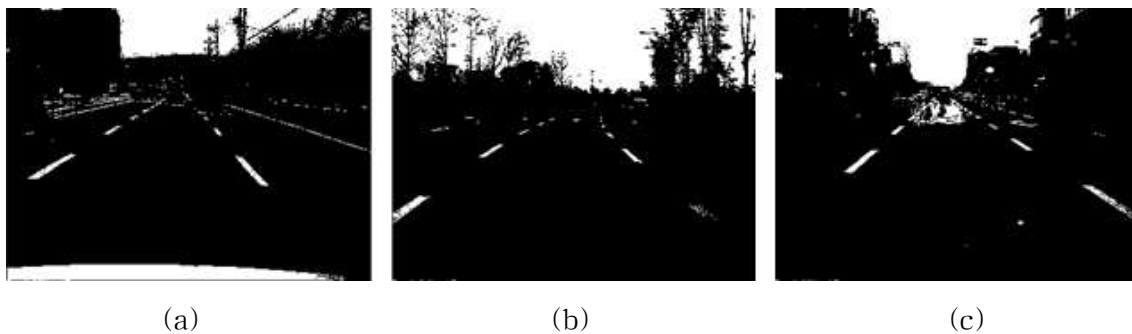


그림 4-6. 마스크에 의한 히스토그램((a) 맑은 날, (b) 흐린 날, (c)비오는 날)

Fig 4-6. Histogram using the mask

((a) Fine day, (b) Cloudy day, (c) Rainy day)

<그림 4-7>은 <그림 4-6>에서 구한 히스토그램을 통한 임계값(동적 임계값)을 이용한 이진화 영상을 보여준다.



(a)

(b)

(c)

그림 4-7. 동적 임계값에 의한 이진화 영상

((a) 맑은 날, (b) 흐린 날, (c)비오는 (c) 비오는 날)

Fig 4-7. Binarization by the dynamic threshold value

((a) Fine day, (b) Cloudy day, (c) Rainy day)

또한 IPM 연산 시 수행하게 되는 삼각함수의 연산시간을 단축하기 위해 Table 방식을 이용한다. IPM은 기본적으로 삼각함수를 이용한 플로팅 포인트 연산을 수행한다.

더욱이 WinCE.net에서 프로그램하기 위한 Embedded visual C++는 삼각함수 계산에 float 형이 아닌 double 형 밖에 지원하지 않기 때문에 삼각함수 계산에 소요되는 시간을 줄일 필요가 있다. 따라서 삼각함수를 계산하기 위한 플로팅 포인트 계산을 줄이기 위해 미리 룬 영역에 삼각함수 테이블을 만들어 이용하면 실제 영상처리 수행 시간을 줄일 수 있다.

다음은 평탄하지 않은 도로에 따른 원근 효과 제거에 의한 왜곡을 줄이기 위한 방법을 제안한다. 우선 원 도로 영상의 지면 부분에 해당하는 부분을 선택한다. 영상을 획득할 경우 지면은 화면의 반 이하로 내려가지 않게 촬영하기 때문에 도로 지면에 해당하는 부분을 도로 평면의 절반 아래에 해당한다고 가정한다. 그리고 화면의 50 % 부분을 비율에 따라 확대하고 이에 따라 카메라의 수직각(θ)을 지면만 나올 수 있는 각(약 20°)으로 재 조정한다. 그리고 AIPM을 이용하여 원근 효과가 제거된 영상을 획득한다. 이 영상은 차선 탐색 시 히스토그램을 이용할 때 사용되어진다.

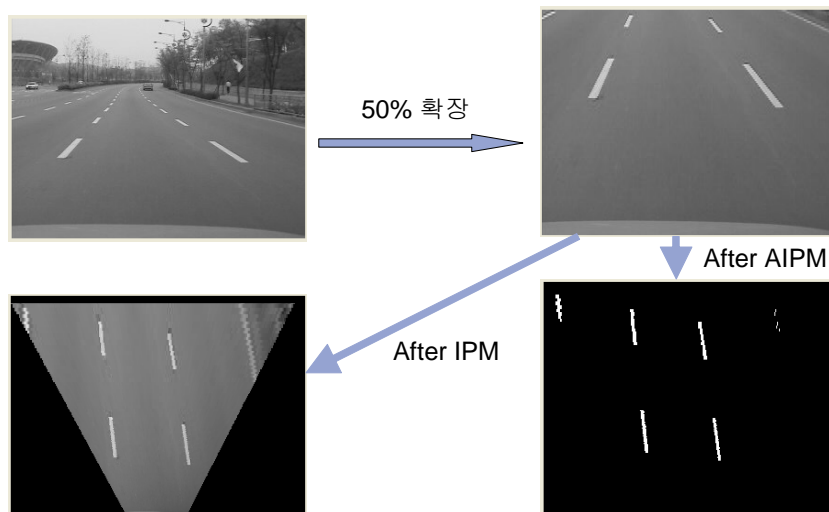


그림 4-8. IPM 왜곡 보완 알고리즘을 적용한 도로 영상

Fig 4-8. Traffic image after processing with the complementary algorithm for IPM distortion

제 4 절 차선 검출

본 연구에서의 차선 검출은 도로가 경사가 심하게 변하지 않고, 차선 사이는 거의 일정한 거리의 폭을 가졌다고 가정된 시스템이기에 3D 도로 평면을 복구하기 위해 고정된 카메라 파라미터를 사용한다. 차선 검출은 3D 공간의 'Z' 축 값이 0인 평면, 즉 지평면에 차선이 있다고 가정한 상태에서 수행된다. 차선으로 판단되는 픽셀들은 좌우의 일정한 거리(고정된 폭을 가지고 있다고 가정하기 때문에)에 또 다른 차선이 있다고 가정하여 조사된다.[4, 11]

차선 검출을 수행하기 전에 적응형 원근 효과 제거(AIPM) 후의 영상에서 잡음 제거를 수행한다. 본 논문에서는 가로로 연속된 2~3픽셀 차선폭을 가지고 있고 차선 사이는 일정 거리(14 ~ 17 픽셀)를 가지고 있는 흰색 픽셀들의 집합을 차선으로 인정하였다. 따라서 한 픽셀이나 4픽셀 이상의 픽셀의 집합은 차선이 아닌 배경(차선외의 도로에 그려진 표시등)이나 도로 상의 장애물로 취급하였다. 이런 잡음을 제거하기 위해 차선폭에 대한 정보를 이용한다. 적응형 원근 효과 제거 후의 영상은 이진화 되어 있기 때문에 차선이 될 수 없는 1 픽셀 혹은 4 이상의 흰색(0xff) 픽셀들은 제거한다.

이진화 후의 원근 효과가 제거된 영상의 픽셀 $b(x, y)$ 의 명도 값은 화면의 그 지점으로부터 좌우로 일정 거리(meter) 떨어진 지점의 픽셀들 $[b(x, y-m), b(x, y+m)]$ 의 명도 값과 비교된다. 그러기 위해서 <그림 4-8>에서 보인 그림과 같은 왜곡 보완 알고리즘을 적용한 영상을 대상으로 <그림 2-4>와 같은 수직 방향 히스토그램[18]을 이용하게 된다. 차선 검출에 이용하게 될 히스토그램은 한 개의 차선 넓이를 포함할 수 있는 5개의 히스토그램 값을 더해서 구한다. 이때 구한 히스토그램에서 좌우 최대값을 찾아서 두 최대값의 히스토그램상의 거리가 차선과 차선 사이의 거리에 해당하는 지를 검사한다. 만약 두 최대값의 히스토그램상의 거리가 차선과 차선 사이의 거리와 일치하지 않는다면 그 다음으로 큰 히스토그램의 값을 취해서 차선과 차선 사이의 거리에 해당하는 가를 비교해 나가면서 차선에 해당하는 부분을 찾아낸다. 히스토그램 상에서 찾아낸 차선에 대한 기초 정보를 기반으로

AIPM 처리 후의 영상에서 차선을 탐색하게 된다. 차선 탐색은 영상의 아래쪽부터 위쪽 방향으로 검색하게 되고 그 시작 위치는 히스토그램으로 찾아낸 영상의 가로 방향 위치를 기준으로 한다. 처음 탐색 시작 위치에서 위쪽 방향으로 일정 이상의 픽셀이 존재하는 지 여부를 결정한 후 차선과 차선 사이의 거리에 해당하는 수평 방향 위치에 일정 이상의 흰색 픽셀(0xff)이 존재하는 지를 검색하면서 전체적인 차선을 탐색해 나간다. 차선은 점선과 실선, 직선과 곡선으로 되어있기 때문에 위쪽 방향으로 탐색할 때 점선으로 이루어진 위쪽 방향의 차선들의 떨어져 있는 거리 및 곡선의 최대 곡률을 검색 범위에 놓고 차선을 탐색해 나간다.

찾아낸 차선은 다시 원근 효과를 제거하지 않은 원 영상에 표시한다. 차선을 원 영상에 표시하기 위해 AIPM을 적용할 때 매핑에 사용된 원 영상의 위치를 테이블 형식으로 보관하여 AIPM 역 변환 시에는 그 테이블을 이용하여 변환함으로써 계산 시간을 줄일 수 있다. 원 영상 상에서의 차선의 표시는 전방 약 20 미터까지의 차선을 대상으로(왜곡 보완 알고리즘으로 구한 영상에서 보이는 차선까지의 거리) 차선의 기울기와 절편을 이용하여 표시한다. 이때 바로 전 프레임의 x 절편의 값이 차선 변경 시 발생할 수 있는 x 절편 변화폭을 벗어 날 경우 그 프레임에 대한 차선 검출은 에러로 간주되어 전 프레임의 차선 데이터를 사용한다.

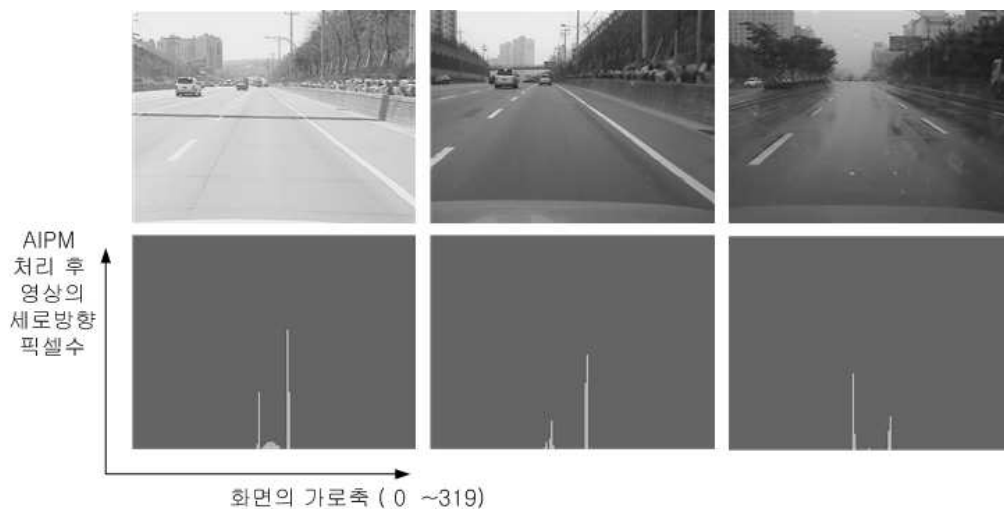


그림 4-9. AIPM 후의 영상과 날씨에 따른 히스토그램

Fig 4-9. Histogram after AIPM processing according to weather

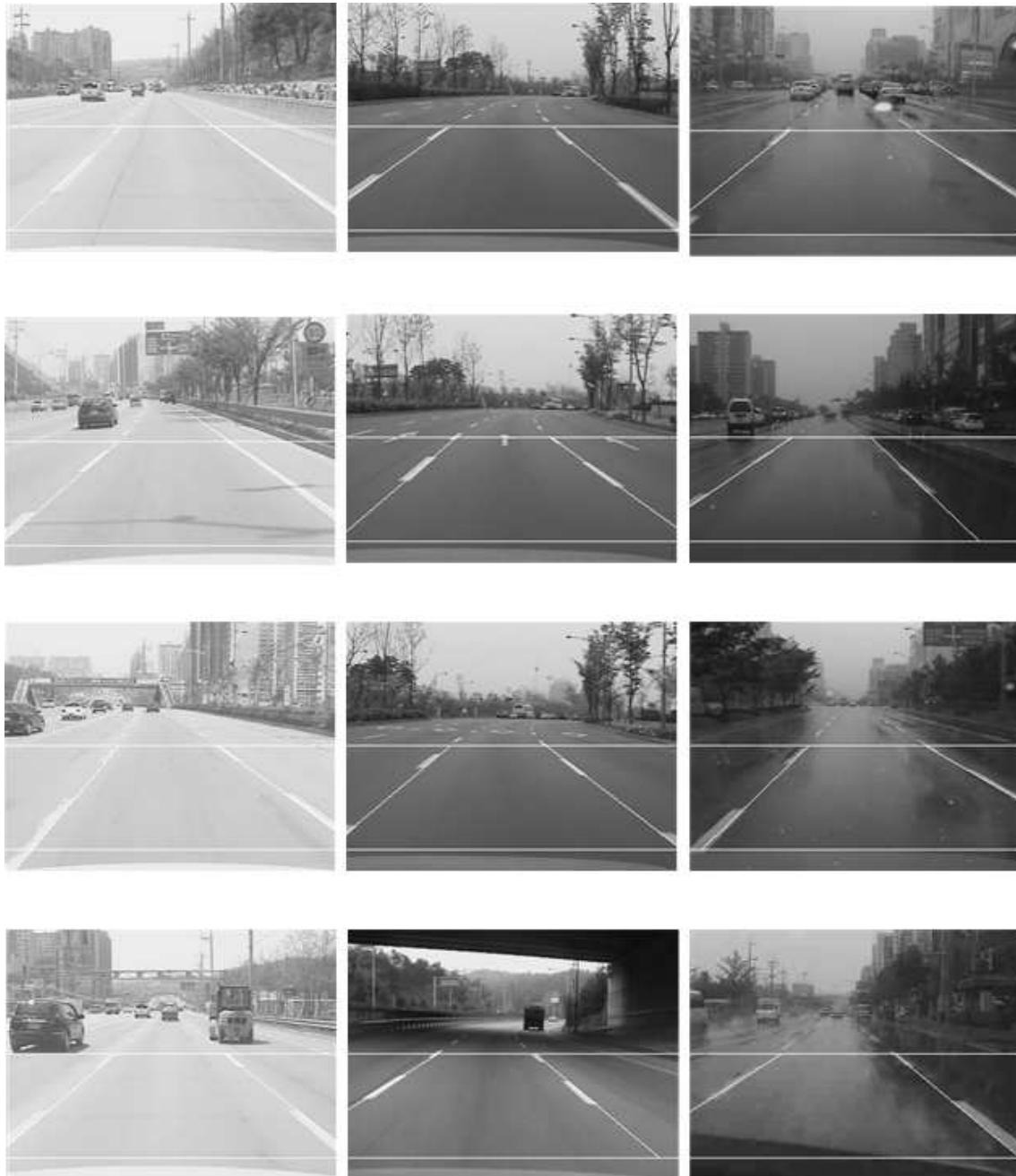


그림 4-10. 차선 검출 후 결과 영상

Fig 4-10. Result Image after lane detection

제 5 장 제안한 알고리즘 성능 평가 환경 및 결과

제 1 절 알고리즘 평가를 위해 사용된 하드웨어 구성

이미지 처리 보드(Image Processing Board)의 주요 기능은 각종 파라미터들에 따른 영상 처리 연산을 수행하고, DMA(Direct Memory Access)를 사용하여 고속으로 CPU와 카메라 사이의 인터페이스를 처리하기 위한 기능을 수행한다. 또한 보드 간 시리얼 통신을 이용한 메인 보드와의 차선 정보에 대한 전송 작업을 수행한다. 기본 메모리는 DRAM 32MB, NAND FLASH 32MB를 사용한다. 또한 타겟 보드는 CMOS와 CCD 이미지 센서가 연결 가능하며, 센서로부터 들어오는 10-bit 입력 데이터 처리를 수행한다.

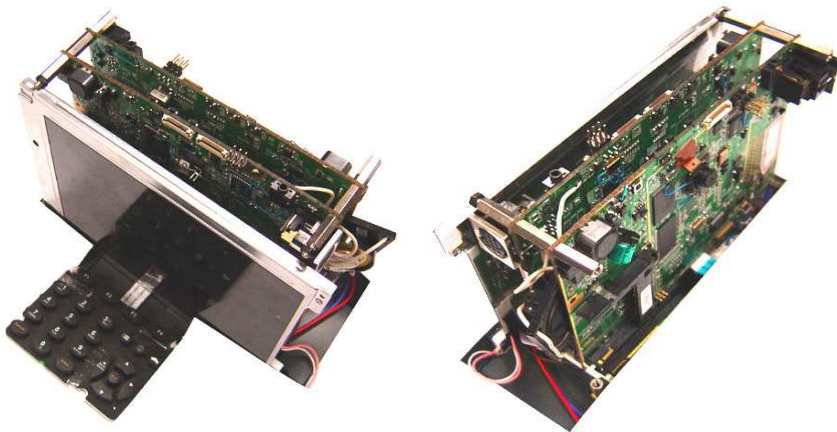


그림 5-1. 타겟 보드 사진

Fig 5-1. Target Board

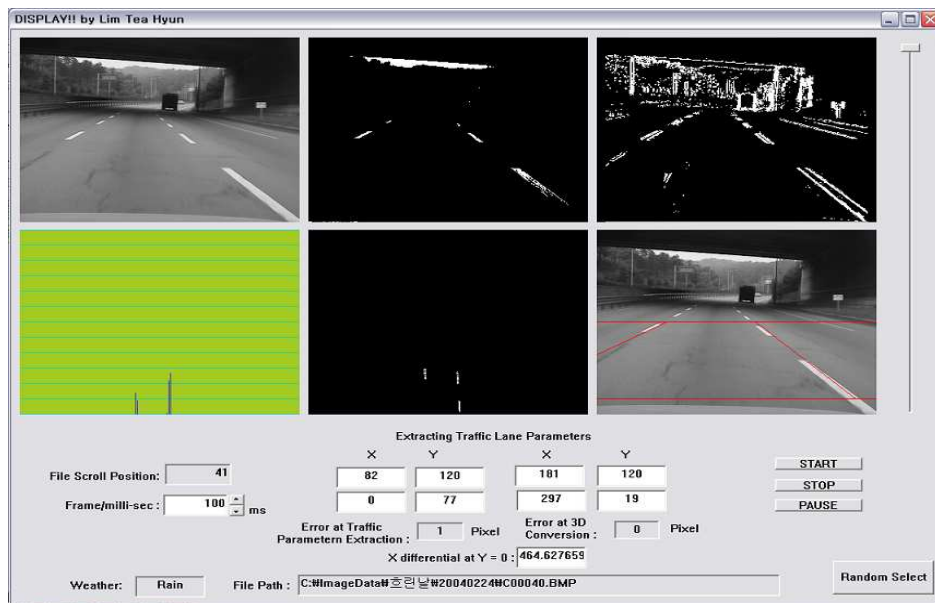


그림 5-2. MS Visual C++ 로 작성한 차선 검출 프로그램

Fig 5-2. Lane Detection Program by MS Visual C++

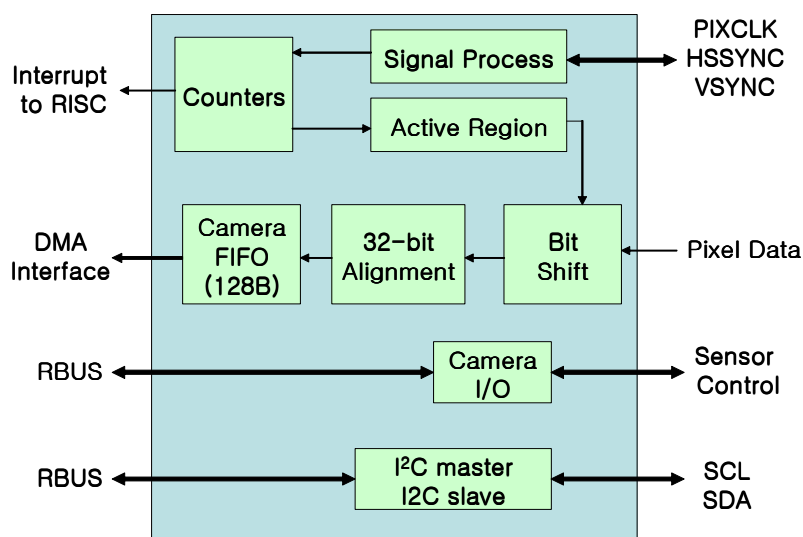


그림 5-3. 시스템과 카메라 모듈의 인터페이스 부분의 기능
블록 다이어그램

Fig 5-3. Function Block Diagram about interface
between the system and camera module

제 2 절 평가 환경 및 결과

차선 검출 성능은 원근 효과의 제거에 의한 3D 변환 성능과 밀접한 관계가 있다. 실제로 차선을 검출하기 위한 파라미터 추출은 3D 변환 이후 이루어지기 때문에 차선 검출 정확도와 3D 변환 오차율은 거의 일치한다. 차선 검출은 해당 차량이 지나가고 있는 차선에 대해서만 이루어지며, 카메라를 통해 시스템에 의해 들어오는 도로 영상은 초당 10 프레임으로 차선 검출이 이루어진다. 차선 검출 대한 평가는 날씨 별로 분류하여 검출 오류를 체크하였다. 차선 검출 환경은 차선을 획득한 날씨에 따라 맑은 날, 흐린 날, 비오는 날로 분류하였다. 날씨 별로 그 특성은 맑은 날은 흐린 날에 비해 상대적으로 조도가 높고 비오는 날은 잡음의 영향이 심하다. 또한 비오는 날은 도로 상에 존재하는 빗물에 의한 햇빛의 반사광이 원인이 돼서 도로 전반적으로 일정한 조도를 가지고 있지 않다.

차선 검출 에러 평가는 차선이 아닌 곳을 검지할 경우와 차선인 부분을 차선으로 검출하지 않은 경우를 에러로 판별한다. 각 도로 영상 프레임 당 차선 검출이 이루어지며, 차선 검출 에러가 있을 경우 그 프레임을 에러로 취급하였다.

차선 알고리즘은 PC를 기반으로 해서 그 차선 검지 정확도를 측정하였다. PC 는 CPU 펜티엄(Pentium)4 2.66Ghz, 내부 L1캐쉬 8KB, L2캐쉬 512KB, RAM 256M 의 성능을 가지고 있다. 알고리즘 구현을 위해 Microsoft Visual C++ 6.0 의 MFC 프로그램을 사용하였다.

또한 차선 검출 알고리즘은 타겟 시스템 상에서 실시간으로 처리가 가능하여야 하기 때문에 타겟 보드에서 알고리즘의 수행 시간을 측정하였다. 타겟 보드 상에서 쓰인 알고리즘 시간 측정에 쓰인 도로 영상 사진은 PC를 기반으로 평가했던 영상 사진을 타겟 보드에 다운로드(Download)시켜서 그 실행 시간을 측정하였다. 타겟 보드는 ARM9 코어를 사용한 200MHz AtlasTM 프로세서를 사용하였고, WinCE.net 4.2가 포팅되었다. 타겟 보드 상에서 알고리즘은 실행시키기 위해 MS Embedded Visual C++ 를 사용하였다.

평가를 위한 영상을 획득하기 위해 <표 5-1>에서 기술되어진 실험 장소와 시간

대에서 해당 차량을 이용하여 캠코더로 촬영하였다. 촬영된 컬러영상은 320 x 240의 BMP형식으로 10frame/second로 샘플링(sampling)하였다. 샘플링 된 320 x 240 컬러 BMP 형식의 영상 사진은 YCbCr 형식으로 변환한 후 흑백 영상을 생성하여 차선 검출 알고리즘을 평가하는데 사용하였다.

표 5-1. 알고리즘 평가를 위한 도로 영상 획득을 위한 환경

Table 5-1. Environment for evaluating lane detection algorithm

날씨	날짜 (년.월.일)	실험환경		
		실험차량	실험장소	실험시간 (시)
흐린 날	2004.02.03	아반떼	수원 올림픽 경기장인근	16~17
	2004.02.16	아반떼	아주대 주변도로	14~15
	2004.02.24	아반떼	43번 국도일대	17~18
	2004.02.27	아반떼	43번국도 광주방면	10~11
	2004.03.26	아반떼	1번국도 일대	17~18
맑은 날	2004.02.09	아반떼	43번국도 원천~수지방향	14~15
	2004.02.11	아반떼	43번국도 일대	13~14
	2004.02.20	아반떼	1번국도 동수원 구간	15~16
	2004.03.03	아반떼	아주대~월드컵경기장	16~17
	2004.04.07	아반떼	영동고속도로(북수원 근교)	11~12
비오는 날	2004.03.10	아반떼	야탑역~서현역	16~17
	2004.04.15	아반떼	영통대로	17~18
	2004.04.30	아반떼	분당일대	16~17
	2004.05.07	아반떼	영통~수원역	16~17
	2004.06.03	아반떼	영통대로	10~11

위의 차선 검출 평가 환경 및 에러 평가 기준에 따른 차선 검출 정확도(100% - 영상처리오차율)를 측정하기 위한 영상 데이터는 날씨별로 맑은 날, 흐린 날, 비오는 날

에 각각 5일을 차량을 이용하여 획득하였다. 또한 획득한 영상은 각각 200프레임의
도로 사진을 사용하여 평가하였다.

각 샘플 영상에 대한 차선 검출율은 다음과 같다.

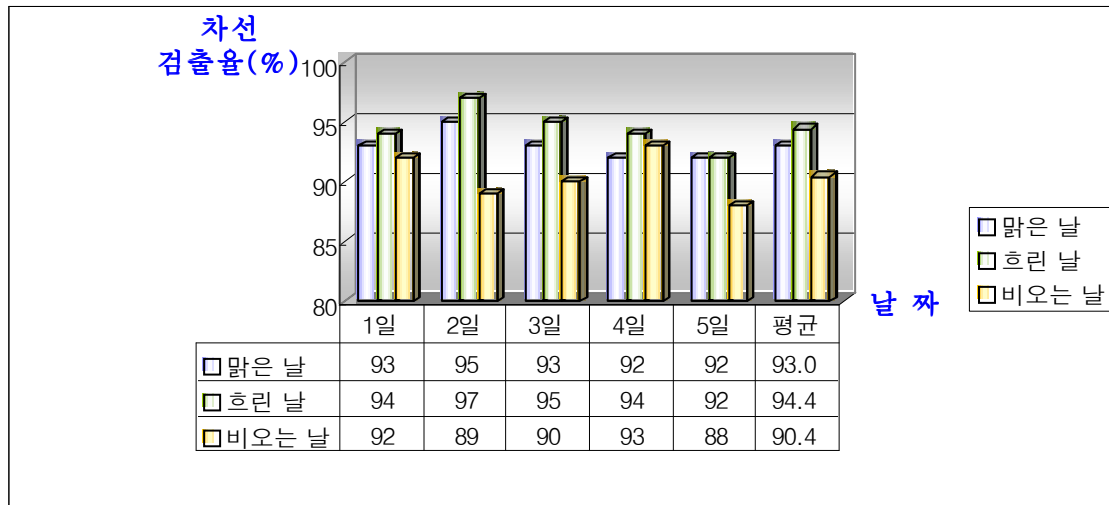


그림 5-4. 차선 검출 정확도

Fig 5-4. Results of lane detection

<그림 5-4>에서 보는 바와 같이 차선 검출 성능을 나타내는 차선 검출 정확도는
조도가 높지 않아 차선의 명암도가 확실한 흐린 날은 약 94%로 가장 높았고, 반면 잡
음과 빗물에 의한 반사광으로 인해 조도가 일정하지 않은 비오는 날의 경우 약 90%
로 가장 낮게 나왔다. 종합적인 차선 검출 성능은 약 93%를 보였다.

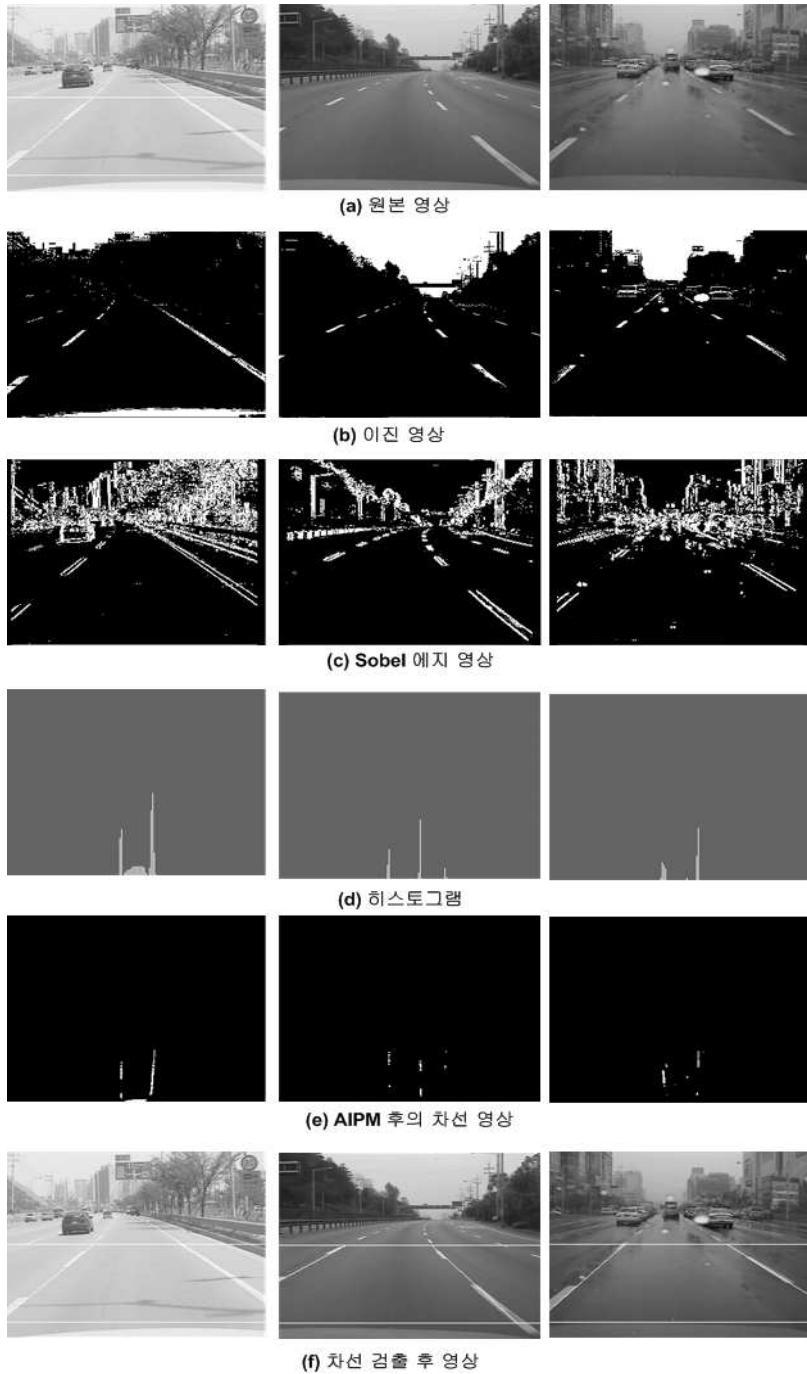


그림 5-5. 날씨에 따른 차선 검출 결과 영상
(왼쪽부터 맑은 날 , 흐린 날, 비오는 날의 영상임)

Fig 5-5. Result image of Traffic Lane Detection
(fine, Cloudy, rainy from left image)

AIPM에서 이진화를 위한 동적 임계값의 변화는 <그림 5-6>와 같다. 날씨별로 임계값을 표시하였으며, 차량 검출 평가에 쓰인 3000개의 영상을 대상으로 하였다.

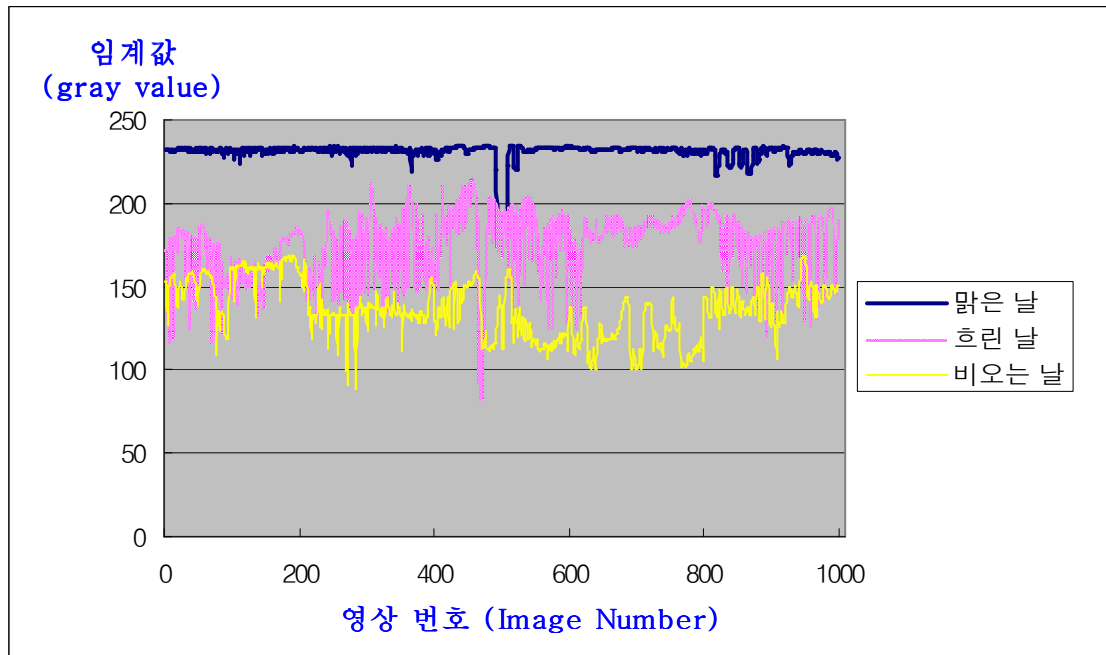


그림 5-6. 날씨에 따른 샘플 영상의 임계값 변화

Fig 5-6. Variation of threshold value about sample images according to weather

표 5-2 임계값 변화를 평가한 영상을 획득한 날짜

Table 5-2 Sample images for evaluating variation of threshold

영상 번호 날씨	0 ~ 200	200 ~ 400	400 ~ 600	600 ~ 800	800~1000
맑은 날	2004.02.03	2004.02.16	2004.02.24	2004.02.27	2004.03.26
흐린 날	2004.02.09	2004.02.11	2004.02.20	2004.03.03	2004.04.07
비오는 날	2004.03.10	2004.04.15	2004.04.30	2004.05.07	2004.06.03

단위: 년. 월. 일

임계치(Threshold value) 평균은 맑은 날의 경우 231, 흐린 날의 경우 175 그리고 비오는 날은 137의 값을 가졌다.

각 프레임별로 동적인 임계치를 갖지 않고 고정적인 임계치를 갖는 시스템의 경우 <그림 5-7>에서 보는 것과 같이 그 평균 차선 검출 성능이 약 83%로 동적 임계치를 갖는 시스템에 비해 약 10% 정도 저하됨을 볼 수 있었다.

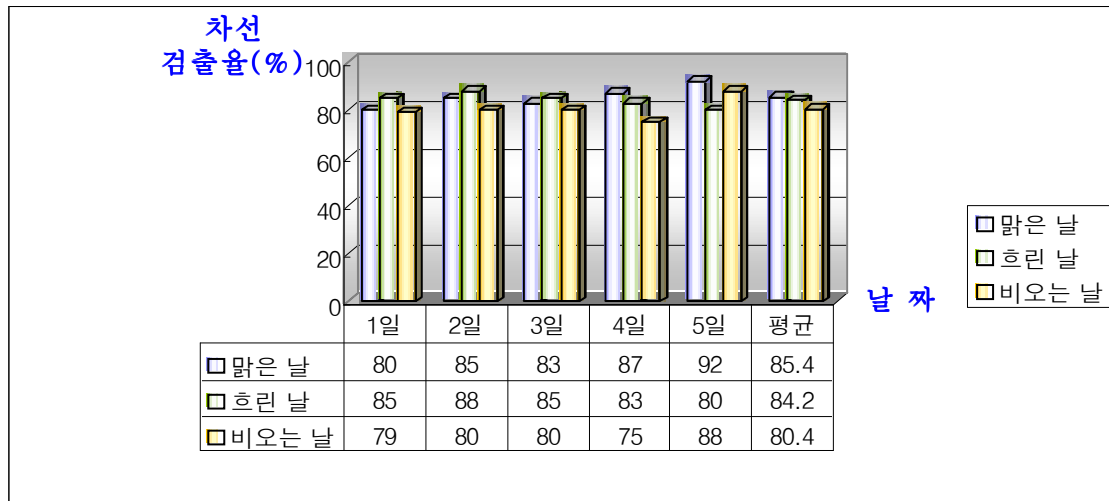


그림 5-7. 고정 임계값을 사용하였을 경우의 차선 검출 정확도

Fig 5-7. Result of lane detection with fixed threshold value

또한 실시간 처리 가능성을 평가하기 위해 Embedded Visual C++를 이용하여 프로그램을 작성한 후 타겟 보드 상에서 영상 100 프레임을 대상으로 차선 검출 시간을 측정하였다.

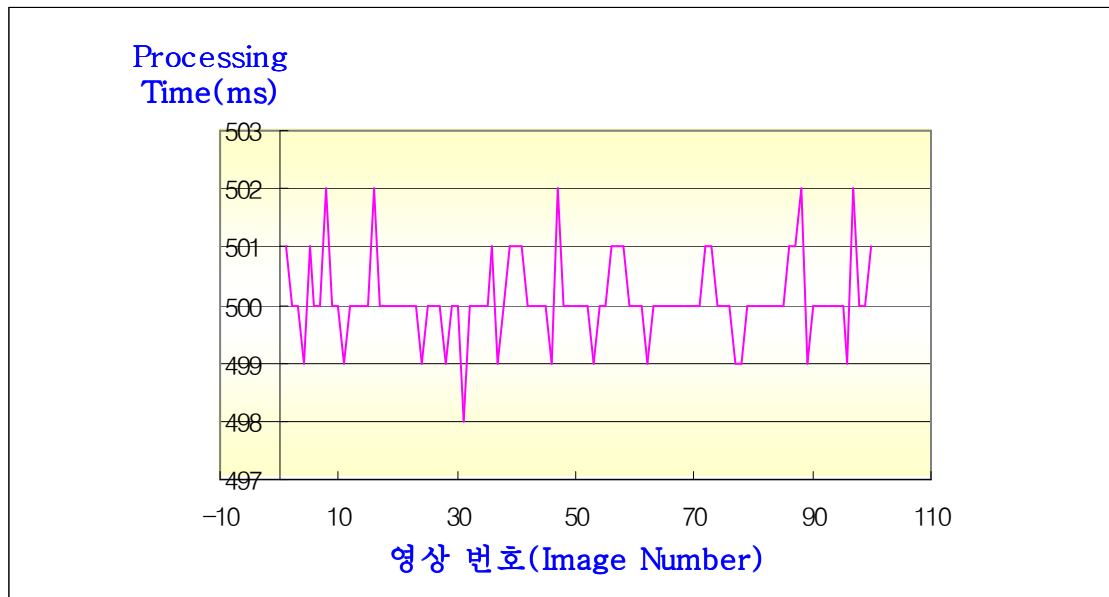


그림 5-8. 타겟 보드 상에서 IPM을 이용한 영상 처리 시간
 Fig 5-8. Image processing time using IPM on target board

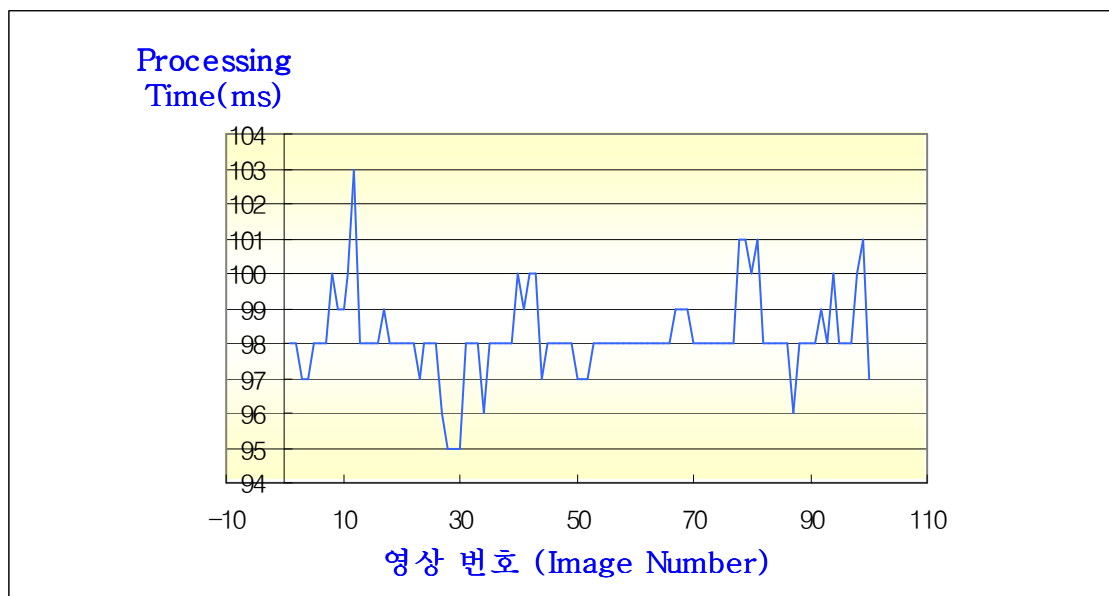


그림 5-9. 타겟 보드 상에서 AIPM을 이용한 영상 처리 시간
 Fig 5-9. Image processing time using AIPM on target board

일반적인 원근 효과 제거 방식(IPM)을 이용한 차선검출을 위한 100개의 영상 프레임 평균 처리 시간은 500ms가 걸린 반면 적응형 원근 효과 제거(AIPM)를 이용한 경우 평균처리 시간은 98 ms가 소요됐다.

본 논문에서 비오는 날 영상은 비가 그칠 무렵 촬영한 영상을 샘플링하여 평가하였다. 하지만 <그림 5-10>에서 보인 영상(비에 의해 잡음이 심한 영상, 차량의 와이퍼(wiper)에 의한 잡음이 섞인 영상, 교차로가 많은 지역에서 촬영한 영상)과 같은 잡음의 영향이 많은 환경에서의 차선검출은 평균 65%의 차선 검출율을 보였다.



그림 5-10. 잡음으로 인해 차선 검출에 실패한 영상 샘플

Fig 5-10. Image sample which have many noises

제 6 장 결 론

카메라를 통해 획득된 영상은 그 특성상 원근 효과를 가지고 있다. 그런데 원근 효과가 포함된 영상은 차선 고유의 파라미터를 추출하기가 어렵다. 따라서 이 차선 파라미터를 추출하기 위해 원근 효과 제거 방법을 사용한다. 기존의 차선 검출을 위한 원근 효과 제거 방법은 원 영상의 모든 픽셀에 대한 원근 효과 제거 기법을 적용시킴으로써 그 연산량이 증가하여 초당 10 프레임으로 영상을 처리하기 위해서는 전용 이미지 프로세서를 사용하여야만 했다. 특히 범용 프로세서인 ARM 9을 이용한 WinCE.net 기반 임베디드 시스템에서의 처리 속도는 원근 효과 제거를 위한 연산 시 수행하는 플로팅 포인트 연산으로 인해 그 처리 속도가 현저히 떨어진다. 그 결과 실시간으로 초당 10 프레임의 연산을 수행해야하는 시스템에서의 응용은 어려웠다.

본 논문은 안전 운전 보조 장치로써 응용 가능한 실시간 시스템에서의 차선 검출 알고리즘을 제안하였다. 제안한 적응형 원근 효과 제거 방법은 차선 가능성이 있는 영상 픽셀에 대해서만 원근 효과를 제거시킴으로써 실시간 처리에서 가장 중요한 영상 알고리즘 처리 시간을 타겟 보드 상에서 평균 400ms 이상 단축시킬 수 있었다.

차선 검출 가능성이 있는 도로 영상을 검출해 내기 위해서 가변적인 임계값을 구하여 이진화를 수행하였다. 가변적인 임계값을 구하는 방법은 차선이 될 가능성이 있는 영역을 마스크 처리하고, 그 부분에 대한 히스토그램을 이용해서 차선에 대한 임계값을 찾았다. 가변적인 임계값은 영상 프레임에 따라 달라지는 조도, 그림자 및 도로 상의 흰색 차량과 장애물의 영향으로 인한 오검지율을 약 10% 가량 줄일 수 있었다.

그 결과 전용 이미지 프로세서를 사용하지 않고 일반적인 ARM9 프로세서를 이용한 타겟 보드 상에서 차선 검출 시간이 평균 98ms 가 걸려 초당 10 프레임을 실시간으로 처리할 수 있음을 보였다. 또한 차선 검출 정확도도 평균 93%를 보여 영상의 모든 부분을 IPM 방법을 이용하여 차선을 추출하는 기존의 알고리즘에 비해 성능이 크게 떨어지지 않았다. 따라서 본 논문은 제안된 차선 검출 방법인 적응형 원근 효과 제거를 이용한 차선 검출 방법과 타겟 시스템이 운전자를 위한 안전 운전 보조 장치로써 실제 차량에서 적용 가능함을 보였다.

실시간으로 추출해 낸 차선 데이터는 ECU로부터 전송되어진 차량의 속도 데이터와 조합하여 급차선 변경 사항을 알람을 통하여 운전자에게 주지시켜줌으로써 차선 변경 시 일어날 수 있는 교통사고를 예방하는데 도움을 줄 수 있다. 또한 주행 중에 전방 차량을 검지하기 위한 기본 정보로서의 응용도 가능할 것이다.

반면에 빗방울과 와이퍼의 영향이 심한 비가 많이 오는 날의 영상과 교차로가 많은 지역에서의 영상은 차선 검출도가 평균 65%로 다른 영상에 비해 약 28% 정도 검지율이 떨어짐을 볼 수 있었다. 따라서 이와 같은 영상에 대한 향후 연구가 필요하다.

참 고 문 헌

- [1] 하동문, 이종민, 김용득, “영상 기반 교통 정보 추출 알고리즘에 관한 연구”, *대한교통학회지*, 19권(6), pp 161-170, December, 2001.
- [2] Mei Yu, Yong Deak Kim, "An Approach to Korean License Plate Recognition based on Vertical Edge Matching", *IEEE Interantional Conference on*, Vol. 4, pp 2975-2980, October, 2000.
- [3] Dong Mun Ha, Jong Min Lee, Yong-Deak Kim, "Neural-edge-based vehicle detection and traffic parameter extraction", *Image and Vision Computing*, Vol 22. Issue 11, pp.863~927, Setember 2004.
- [4] Gang Yi Jiang, Tae Young Choi, Suk Kyo Hong, Jae Wook Bae, Byung Suk Song, "Lane and Obstacle Detection Based on Fast Inverse Perspective Mapping Algorithm", *Systems, Man, and Cybernetics, 2000 IEEE International Conference on* , Vol. 4, pp 2969-2974, October, 2000.
- [5] Yue Wang, Dinggang Shen, Eam Khwang Teoh, "Lane detection using spline model", *ELSEVIER, Pattern Recognition Letter 21* , pp 677-689 , February, 2000.
- [6] Yue Wang, Eam Khwang Teoh, Dinggang Shen, "Lane detection using B-snake", *IEEE ,Information Intelligence and System, 1999.Proceedings. International Conference on*, pp 438~443, November, 1999.
- [7] D. Pomerleau, "RALPH: rapidly adapting lateral position handler," *Proc. IEEE Symposium on Intelligent Vehicles*, pp.506-511, September, 1995.
- [8] P. H. Batavia, D. A. Pomerleau, C. E. Thorpe, "Extention of the ALVINN-architecture for robust visual guidance of a miniature robot", *Advanced Mobile Robots, 1997. Proceedings., Second EUROMICRO workshop on* , pp 8~14, October, 1997.
- [9] K. Kluge, "Extracting road curvature and orientation from image edge points

- without perceptual grouping into features", *Proceedings of the Intelligent Vehicles '94 Symposium*, pp 109–114, October, 1994.
- [10] S. L. M. Beuvais, C. Kreucher, "Building world model for mobile platforms using heterogeneous sensors fusion and temporal analysis", *Proc. IEEE Int. Conf. on Intelligent Transportation System'97*, pp 230~235 1997.
- [11] M. Bertozzi and A. Broggi, "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection", *IEEE Trans. on Image Proc.*, Vol 7 (1), pp 62–81, January 1998.
- [12] W. Enkelmann, G. Struck, J. Geisler, "ROMA: A system for model-based analysis of road markings", *Proc. IEEE Intelligent Vehicle'95, Conf.*, pp 356–360, 1995.
- [13] Ken ISHIKAWA, Kei ODAGIRI, Kazuyuki KOBAYASHI, Kajiro WATANABE, "Lane detection by using Omni-Directional Camera for outdoor terrains", *IEEE International Symposium, Industrial Electronics, 2001, Proceedings, ISIE 2001*, Vol. 3, pp 1466–1471 , June, 2001.
- [14] Joon Woong Lee, "A Machine Vision System for Lane-Departure Detection", *Elsevier Science(USA), Computer Vision and Image Understanding 86*, pp 52–78 , 2002.
- [15] Alberto Broggi, "Parallel and Local Feature Extraction : A Real-Time Approach to Road Boundary Detection", *IEEE Transactions on Image Processing*, Vol. 4 (2), pp 217–223, FEBRUARY, 1995.
- [16] C. Kreucher, S. Lakshmanan, K. Kluge, "A driver warning system based on the LOIS lane detection algorithm", *1998 IEEE Int. Conf. on Intelligent Vehicles*, p.17–22, Stuttgart, Germany, Oct., 1998.
- [17] Thomas Bucher, Cristobal Curio, Johann Edelbrunner, Christian Igel " Image Processing and Behavior Planning for Intelligent Vehicles", *IEEE, Transactions on Industrial Electronics*, Vol. 50 (1), pp 62–75, Feb, 2003.

- [18] Juan Pablo Gonzalez, Umit Ozguner, "Lane Detection Using Histogram-based Segmentation and Decision Trees", *IEEE, Intelligent Transportation Systems*, pp 346-351, October, 2000.
- [19] Phillip A. Mlsna, Jeffrey J. Rodriguez, Handbook of Image & Video processing, Academic Press, Canada, pp415~431, 2000.
- [20] M. Fathy, M.Y. Siyal, "A Window-based Edge Detection Technique for Measuring Road Traffic Parameters in Real-Time ", *Real-Time Imaging 1*, pp 297-305, 1995.
- [21] Qiuming Zhu , "Efficient evaluations of edge connectivity and width uniformity ", *ELSEVIER, Image and vision computing 14*, pp 21-34, 1996
- [22] Rafael C. Gonzalez , Richard E. Woods , Digital Image Processing, Addison Wesley, USA, pp 51~71, pp 424~431, 1993.
- [23] M. Bertozzi and A. Broggi, "Vision-based vehicle guidance," *IEEE Computer*, vol.30 (7), p.49-55, 1997.

Abstract

Requirement of traffic safety have been increased as volume of vehicle is growing. And an improvement of embedded system with wireless technology can help to develop devices for traffic safety. In intelligent transportation system(ITS), image processing and computer vision system has often been used for traffic lane and obstacle detection. Vision based vehicle guidance system is helpful to relieving the contradiction between enhancement of traffic safety and increment of traffic density.

This dissertation proposes a vision-based traffic lane detection algorithm with general ARM9 processor. An AIPM(Adaptive Inverse Perspective mapping) algorithm can reduce bad effects of some noise(variation of lumination, shadows, vehicles and obstacles which have white color same as traffic lane) and computation time. This lane detection algorithm evaluates on PC with traffic image that take a photograph on real road during driving. And this lane detection program is ported onto target board for evaluating a processing time. An algorithm can be computed by 10 frame per second on the target system board(Computation time is 98ms). Also, it get about 93% correct lane detection rate. And it can decrease 10% error which have due to some noises.

The result show that this system and traffic lane detection algorithm can apply auxiliary system for safe driving part of vehicle guidance system.