

아이폰 기반의 차선이탈경보 애플리케이션 개발

윤호영, 김종호, 김한솔, 노광현*

*한성대학교 산업경영공학과

e-mail :0895037, graylove, hyde0629, khrho@hansung.ac.kr

Development of Lane Departure Warning Application on the iPhone

Ho-Young Yun, Jong-Ho Kim, Han-Sol Kim, Kwang-Hyun Ro*

*Dept. of Industrial & Management, Hansung University

요 약

본 연구에서는 최근 인기를 끌고 있는 스마트폰 중 하나인 Apple사의 iPhone 3GS 용 차선이탈경보 애플리케이션 개발을 수행하였다. 국내외적으로 수행되고 있는 지능형 자동차 연구를 통해 개발된 다양한 안전주행 지원 기술들이 단계적으로 상용화되면서 자동차를 지능화시키고 있는 시점에서 고가의 안전주행 지원장치를 대체할 수 있는 플랫폼으로 스마트폰을 활용하고, 안전주행 지원기술을 애플리케이션으로 개발하여 사용자가 쉽고 저렴하고 편리하게 사용할 수 있는 제품을 개발하고자 하였다. Before-Market에만 집중된 차선이탈경보장치 기능을 iPhone에 적용하기 위해서 제한된 컴퓨팅 자원을 효율적으로 활용할 수 있는 차선인식알고리즘이 필요하고, 본 연구에서는 Hough Transform 기반의 알고리즘을 사용하였다. 이외에도 iPhone이라는 플랫폼의 여러 가지 특징을 고려하여 애플리케이션을 개발하였다. 효율적인 영상처리를 위해 OpenCV 라이브러리를 사용하였고, 개발은 매킨토시 컴퓨터에서 개발 후 iPhone에 탑재하여 확인하였다. 현재까지 개발된 애플리케이션의 기능에서 보완할 점은 차선인식률과 처리 속도를 향상 시키는 것이다. 2011년 2월까지 완제품을 개발하여 앱스토어에 등록하는 것을 목표로 하고 있으며, 향후 안드로이드용 애플리케이션도 개발할 계획이다.

Keyword : 지능형 자동차(intelligent vehicle), 스마트폰(smart phone), iPhone, 차선이탈경보시스템(Lane Departure Warning System), OpenCV

1. 서론

지능형 자동차에 대한 연구는 해외에서는 물론 국내에서도 예전부터 이루어져 왔으며 기술력은 이미 높은 수준에 도달하였다. 최근 차선인식을 비롯한 장애물 탐지, 차량제어 등의 지능형 기능이 구현된 상용 제품들이 고가의 자동차에 Before-Market 용으로 탑재되어 고가로 판매되고 있다. After-Market 용으로는 차량용 블랙박스가 폭발적인 인기를 얻고 있으며, 네비게이션 시장 수준으로 성장하고 있다. 이렇게 사고 발생시 원인을 분석할 수 있는 장치뿐만 아니라 운전자의 부주의로 발생할 수 있는 교통사고를 미연에 방지할 수 있는 지능형 장치에 대한 시장의 요구가 높아지고 있는 실정이다.

본 연구에서는 최근 음성통화 기능을 벗어나 다양한 분야에 적용되고 있는 스마트폰을 활용한 차선이탈경보 시스템을 구축하고자 한다. 차선이탈경보장치도 Before & After Market 용으로 개발되어 판매되고 있지만 성능 대비 가격이 고가이며 설치의 불편함 때문에 시장이 활성화되지 못하고 있다. 이러한 문제점을 해결할 수 있는 방안으로 차선이탈경보장치의 플랫폼을 스마트폰 활성화의 중심에 있는 Apple사의 iPhone으로 선정하였고, 차선인식 및 이탈경보를 애플리케이션으로 개발하여 일반 사용자들이 쉽고 편리하며 저렴하게 차선이탈경보기능을 사용할 수 있도록 하고자 하였다. iPhone 용 다양한 애플

리케이션이 개발되고 있지만 안전 주행을 위한 애플리케이션은 찾아보기 어렵기 때문에 이러한 시장에 진출할 수 있는 발판을 마련할 수 있을 것이다.

제안하는 iPhone 기반의 차선이탈경보 애플리케이션은 iPhone의 외장 카메라에서 입력되는 전방 도로 영상에서 연속적으로 차선을 추출하고 차선이 판단되는 경우 영상과 경고음을 통해 운전자에게 경고 상황을 알려준다. 차선인식 기능 구현을 위해서는 영상처리기능이 기본적으로 필요한데 이를 위해서 Intel의 영상처리 라이브러리인 OpenCV를 사용하였고, 차선인식 알고리즘은 Hough Transform을 기반으로 개발하였다. 차선인식 알고리즘에 대해서는 현재까지 많은 연구가 수행되었다. 하지만 기존의 차선인식용 전용 장치에 비해 전반적으로 컴퓨팅 성능이 떨어지는 범용 스마트폰에서 실시간 차선인식하기 위해서는 스마트폰의 자원을 적절히 사용할 수 있는 차선인식 알고리즘이 필요하였다. 본 연구는 현재 진행 중이며, 개발 중인 차선이탈경보 애플리케이션을 탑재한 아이폰을 차량에 거치하여 시험하고 있다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 연구에 사용되는 플랫폼인 iPhone에 대해 살펴보고, 제 3 장에서는 차선인식 알고리즘 구현에 사용한 OpenCV에 대해 기술하였다. 제 4 장에서는 구현된 차선인식 알고리즘에 대해서 설명하고, 제 5 장에서는 실험결과를 통해 알고리즘

의 성능을 분석하였고, 6장에서 결론을 맺는다.

2. iPhone

하드웨어 성능에 큰 영향을 받는 영상처리는 Multi-touch Screen, GPS, WiFi, G-Sensor, Camera 등이 탑재된 iPhone의 등장으로 보다 넓은 방향으로 발전하게 되었다. iPhone은 Apple의 데스크탑 PC인 iMAC 보다는 월등히 떨어지는 사양이지만, iPhone 3G 모델이 출시된 이후로 iPhone 3GS가 출시되면서 사양이 높아졌다. 최근에는 더 높아진 사양을 가진 iPhone 4가 출시되어 애플리케이션의 한계점을 높여가고 있다. iPhone의 사양 변화는 표 1과 같다[1].

표 1. A Comparison between iPhone Models

	iPhone 3G	iPhone 3GS	iPhone 4
CPU	412MHz	600MHz	1GHz
Memory	128MB	256MB	512MB
Screen Size	3.5-inch	3.5-inch	3.5-inch
Resolution	320×480	320×480	640×960
Multi-tasking	X	O (iOS4.0 설치시)	O

iPhone 출시 초기에는 필요에 따라 하드웨어를 확장할 수 없고, 개발자들이 개발한 애플리케이션을 직접 설치할 수 없었다. 하지만 개발자들의 요청에 따라 iPhone SDK가 공개되었고, 폐쇄적인 조건하에서 일반 개발자들도 애플리케이션을 쉽게 개발할 수 있게 되었다[5].

iPhone 애플리케이션 개발에는 Mac OS X 10.3 버전 이상의 환경이 필요하다. 애플리케이션을 개발할 때는 개발툴(IDE: Integrated Development Environment)인 Xcode 내에서 iPhone Simulator에서 테스트할 수 있지만, 카메라나 가속 센서 같이 실제 장치를 반드시 이용해야 하는 애플리케이션을 테스트할 경우에 iOS Developer Program에 가입해야만 한다. iOS Developer Program은 1년에 \$99의 비용을 지불하면, 개발한 애플리케이션을 실제 iPhone에서 실행시킬 수 있다.

iPhone은 원래 멀티태스킹이 불가능하여 제약사항이 많았으나, iOS4.0 버전이 출시된 이후부터 멀티태스킹이 가능해져 애플리케이션을 실행하면서 동시에 다른 작업이 가능해졌다. 하지만 가비지 컬렉션(Garbage Collection)을 제공하지 않아 메모리 점유율이 높은 애플리케이션은 OS가 강제 종료 시키는 등의 제약사항들이 여전히 존재하고 있다. 정확한 기준은 공개되어있지 않지만, 통상적으로 22MB 정도가 적정한 애플리케이션 가용 메모리이다. 이 범위를 초과하여도 애플리케이션은 실행되지만 iPhone의 전체적인 실행속도가 낮아지기 때문에 애플리케이션 개발 시 메모리 점유율에 대해 많은 고려를 해야 한다. 본 연구에 개발한 애플리케이션의 메모리 점유율은 약 18MB로 측정되었다.

iPhone Camera에서 실시간으로 이미지 프레임을 받아오는 함수는 그 동안 공개되지 않아 UIGraphics 라는 Private API를 무단 접근하여 사용하였지만, iOS 4.0부터는 AVCaptureDevice 라는 이미지 실시간 처리 함수를 공개하였으며 실시간 영상처리나 증강현실 애플리케이션에 폭넓게 적용될 것으로 예상된다[9].

iPhone을 차량에 설치한 모습은 그림 1과 같다. 자동

차 전면부 유리창에 차량용 거치대를 설치하여, iPhone을 설치하고 USB 차량용 시거잭을 사용하여, 지속적으로 전원을 공급할 수 있도록 하였다. 전방도로 영상을 촬영하는 iPhone용 카메라가 거치대에 가려지지 않도록 설치하였으며, 카메라, 자동차, 도로간의 기하학적인 모형은 그림 2와 같다. 차선인식 알고리즘 개발시 위의 변수들이 일부만 적용되었지만 향후에는 차선인식을 향상을 위해 대부분을 고려할 예정이다.



그림 1. Installation of iPhone on a Car

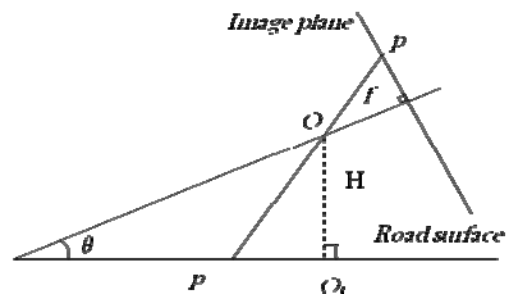


그림 2. The Geometry of Road and Camera

- f : focal length
- O : focal center
- H : height of the camera above ground manifold
- θ : tilt angle of the camera

3. OpenCV

OpenCV(Open Source Computer Vision Library)는 Intel에서 제공하는 컴퓨터비전 라이브러리이다. 컬러 공간 처리, 화소값 기반 처리, 에지 추출, 영상 분석, 카메라 교정, 기계 학습 알고리즘 등 사용하기 쉬운 라이브러리를 실시간 연산이 가능하도록 연산의 효율성을 최대한 고려하여 C와 C++로 작성된 함수로 제공한다. OpenCV의 구조는 그림 3과 같다.

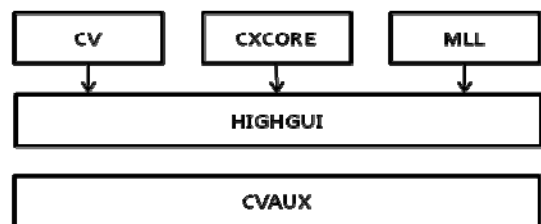


그림 3. OpenCV Structure

OpenCV는 크게 다섯 개의 구성 요소로 나눌 수 있다. CV는 기본적인 영상 처리와 고수준의 컴퓨터 비전 알고리즘을 포함하며, MLL은 통계 분류와 군집화 도구 등의 기계 학습 라이브러리를 포함한다. HighGUI는 입출력에 관련된 함수들과 영상과 비디오를 불러오고 저장하는 기능을 포함하며, CXCore는 기본적인 자료 구조와 알고리즘을 포함하고 있으며, CVAux라는 실험적인 알고리즘도 포함하고 있다.

OpenCV는 이식성을 고려하여 설계되었다. Windows와 Linux, 그리고 Mac OS X에서 사용이 가능하다. iPhone에서도 하드웨어의 발전으로 OpenCV 라이브러리를 사용하여 컴퓨터비전 프로그램을 구현하기에 용이하다. 또한 공개된 라이브러리로서 연구 및 상업 목적으로 사용할 수도 있다. 본 애플리케이션은 iPhone Camera에서 입력 받은 영상을 제어하고, 차선인식을 위한 영상처리를 위해 CV와 HighGUI의 함수들을 사용하여 제작되었다 [3]. OpenCV의 이미지 구조체는 IplImage이며, iPhone SDK의 이미지 구조체는 UIImage이므로 iPhone SDK는 iPhone 카메라로 UIImage를 입력 받은 뒤에 IplImage로 변환하여 OpenCV 함수를 사용하여 영상처리를 수행한 후 결과를 UIImage로 반환할 수 있게 해준다.

4. Lane Detection Algorithm

iPhone이라는 컴퓨팅 자원이 제한적인 플랫폼에서 애플리케이션을 개발하기 위해서는 알고리즘이 단순하고 효율적이어야 한다. 본 연구에서 제안하는 알고리즘은 간결하며 Flow Chart는 그림 4와 같고, 실제 도로 영상에서의 차선인식 과정은 그림 5와 같다.

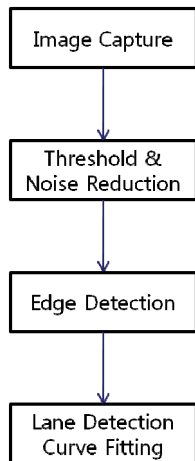


그림 4. Flow Chart of Lane Detection

4.1 Pre-Processing

차선인식에서는 조명이나 그림자 등의 다양한 환경에 의해서 카메라로 받은 영상에 잡음이 생길 수 있다. 잡음이 생기면 차선인식 결과의 신뢰성 저하가 발생되기 때문에 영상 보정 작업을 하는 전처리 과정이 필요하다. 먼저 카메라로 입력 받은 영상에 Median Filter를 적용하여 잡음을 제거하였다. 이 과정은 iPhone에서의 애플리케이션 수행 속도를 높이기 위해 선택적으로 수행하도록 하였다. 잡음 제거 과정을 포함시키지 않은 경우에도 최종 차선인식 결과에는 큰 영향을 미치지 않았다. 다음 단계

에서는 Canny Edge Operator를 사용하여 에지를 추출하였다.

4.2 Line Detection using a Hough Transform

Hough Transform은 영상에서 직선, 원, 또는 다른 간단한 모양을 찾는 방법이며[8], OpenCV에서는 Standard Hough Transform과 Progressive Probabilistic Hough Transform을 제공한다[2]. 본 연구에 사용된 알고리즘에 쓰인 Hough Transform 방식은 Standard Hough Transform 방식이다. Hough Transform은 연산속도가 오래 걸리는 단점이 있지만 이러한 단점을 보완하기 위해 관심영역만 연산할 수 있는 ROI(Region of Interest)를 사용하였다. 그림 2와 같이 iPhone을 설치하게 되면 카메라에서 입력되는 영상의 일부 영역에는 도로 영상이 포함되지 않는다. 따라서 입력 영상의 일부 영역에 대해서만 Hough Transform을 적용하여 연산속도를 향상시킬 수 있으며, 잡음에 의한 오류도 줄일 수 있다[4],[6].

4.3 Lane Detection

Hough Transform을 이용해 추출한 직선들 중에서 각도를 기반으로 차선을 추출하였다. Hough Transform으로 추출된 여러 후보직선 중에 차량의 전방 기준으로 왼쪽

차선의 $\theta \times 180^\circ$ 의 값이 110° 보다 크고 180° 보다 작은 직선을, 차량의 전방 기준으로 오른쪽 차선의 $\theta \times 180^\circ$ 의 값이 0° 보다 크고 70° 보다 작은 직선을 차선으로 추출하였다.

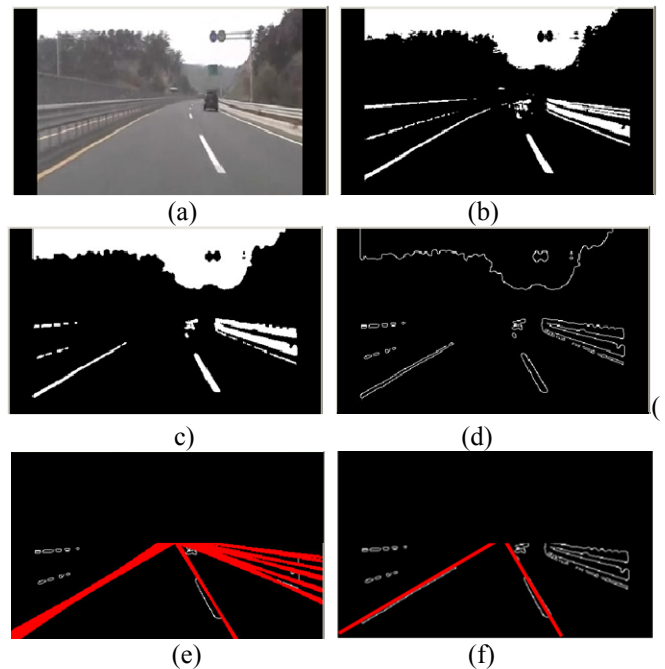


그림 5. An example of lane detection with intermediate results. (a) Input Image. (b) Binary Image. (c) Median Filter. (d) Canny Edge Detection. (e) ROI / Hough Transform. (f) Lane Detection

5. 실험 결과

본 연구는 OpenCV 라이브러리를 이용하여 Windows XP 환경에서 Visual Studio 6.0 개발 툴을 이용하여 알고리즘을 구현한 뒤, Mac OS X Snow Leopard 환경에서도 XCode 3.2.4 개발 툴을 이용하여 구현하는 순서로 이루어졌다. 그 뒤에 iPhone 환경에 맞게 마이그레이션하여 애플리케이션을 개발하였으며, 각각 환경의 사양은 표 2와 같다. Windows와 Mac OS X 환경에서 구축할 때, iPhone 환경에 맞게 320×480의 해상도를 고려하였으며, iPhone 애플리케이션 개발 전까지는 차량에 웹캠을 설치하여 실제 주행을 통해 알고리즘을 테스트하였다.

표 2. A Comparison between Windows, MAC and iPhone

	Windows XP	MAC OS X	iPhone 3GS
CPU	3.4GHz	3.06GHz	600MHz
Memory	1GB	4GB	256MB

도로주행 시험은 직선 주행이 긴 도로를 고려하여 서울 시내에 위치한 내부순환도로와 자유로에서 주간과 야간으로 구분하여 실험하였다.

Window XP 기반의 노트북을 활용하는 경우 20fps 이상의 처리 속도를 얻을 수 있었지만, iPhone에 애플리케이션을 다운로드하여 실행하는 경우에는 10fps 이하의 처리 속도를 보였다. 차선 인식률 측면에서는 연속된 직선 도로에서는 안정적으로 인식하였지만 끊어진 직선 도로나 곡선 도로에서는 인식률이 떨어졌다. 현재의 차선인식 알고리즘은 인식률보다 인식속도를 우선시 하여 Hough Transform을 사용하였고 다양한 도로 상황을 고려하지 않은 초기 단계의 차선인식 알고리즘이기 때문이다. iPhone 환경의 차선 인식 알고리즘의 구현 모습은 그림 6과 같다.

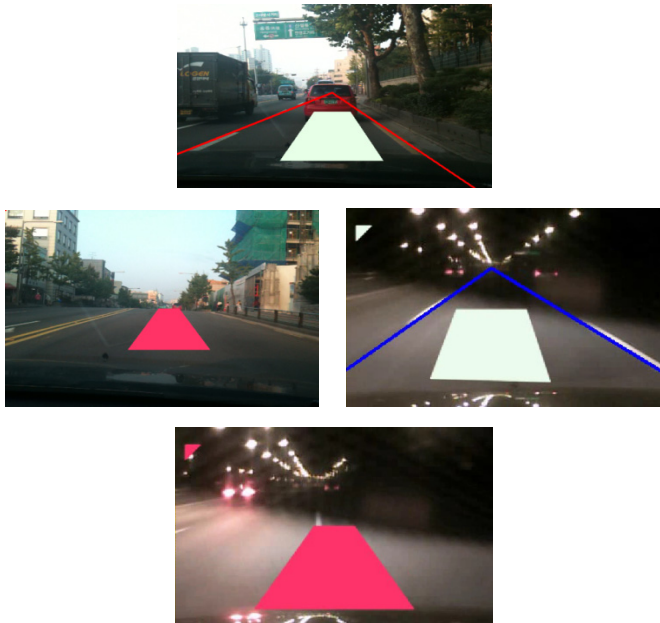


그림 6. Experimental results of LDWS Application on iPhone (Top Left : 주간 정상 주행시, Top Right : 주간 차선 이탈시, Bottom Left : 야간 정상 주행시, Bottom Right : 야간 차선 이탈시)

PC 환경과 iPhone 환경에서의 처리속도 차이는 존재하지만, 스마트폰 기반하에서의 차선이탈경보 애플리케이션의 적용 가능성을 충분히 확인할 수 있었다.

6. 결론

본 연구에서는 일반 PC가 아닌 iPhone이라는 컴퓨팅 자원이 제한된 플랫폼에서 차선인식 알고리즘 구현 가능성을 살펴보기 위함이었으며, 처리 속도, 사용 메모리, 차선 인식률 등의 측면에서 그 가능성을 확인할 수 있었다.

향후에는 차선인식 알고리즘을 개선하여 인식률을 높일 것이며, 차선이탈예측 알고리즘도 연구할 것이다. 또한, 실험에 사용된 iPhone 3GS보다 컴퓨팅 속도가 빠른 iPhone 4 플랫폼을 활용하여 처리 속도를 높일 계획이다. 차선이탈경보 애플리케이션으로의 기능이 안정적인 단계에 이르면 앱스토어를 통해 시장에 출시할 계획이다. 또한, iPhone 이외의 스마트폰용 애플리케이션도 개발하여 다양한 스마트폰에서 이 애플리케이션을 사용할 수 있도록 할 것이다.

참고문헌

- [1] <http://www.apple.com/iphone/specs.html>
- [2] 황선규, "Learning OpenCV 제대로 배우기", 한빛미디어, 2009.
- [3] 이정수, 정세암, 김준성 "OpenCV를 사용한 스테레오 비전 시스템의 프로토타입 구현", 2008년도 대한전자공학 학회 하계종합학술대회 제 31 권 제 1 회 pp.763 - 764, 2008.
- [4] Feixiang Ren, Jinsheng Huang, Mutsuhiro Terauchi, Ruyi Jiang, Reninhard Klette "Lane Detection on the iPhone", Multimedia Imaging Report 43, 2009.
- [5] 유동근, "아이폰&아이팟 프로그래밍 iPhone & iPod Programming", 한빛미디어, 2009.
- [6] 장운, 기창두, "Hough Transform을 이용한 차선인식 응용", 한국자동차공학회 기계의 날 선포 및 2002년도 기계관련 산학연 연합심포지엄, pp912-917, 2002.
- [7] 채은정, 고광은, 심귀보, "이동로봇의 자율주행을 위한 차선인식", 한국지능시스템학회 학술발표 논문집 Vol.20 No.1 pp86 - 89, 2010.
- [8] B. Fardi and G.Wanielik, "Hough Transformation based approach for road border detection in infrared images", in Proc.IEEE Intell. Veh. Symp., Parma, Italy, Jun. 2004, pp. 548- 554.
- [9] iOS Reference Library, <http://developer.apple.com/library/ios/navigation/index.html>