

아이폰 기반의 차선이탈경보 어플리케이션 최적화에 관한 연구

윤호영, 이후림, 노광현*

*한성대학교 산업경영공학과

e-mail : yhy900211, 0895053, khrho@hansung.ac.kr

A Study on Optimizing of Lane Departure Warning Application on the iPhone

Ho-Young Yun, Hoo-Rim Yi, Kwang-Hyun Ro*

*Dept of Industrial & Management Engineering, Hansung University

요 약

본 논문에서는 안전주행지원 솔루션인 스마트폰용 차선이탈경보 어플리케이션의 실행 속도 향상 방법을 연구하였다. 이전 연구에서 스마트폰(iPhone 3GS) 기반의 차선이탈경보 어플리케이션을 개발하였는데 입력 영상의 처리 속도가 1.52fps였다. 본 연구에서는 영상 처리 속도를 향상시키기 위한 최적화 작업을 진행하였으며 기존의 차선이탈경보 어플리케이션의 차선인식 속도보다 프레임당 평균 0.4초 정도 단축되어 약 4fps 속도로 성능을 보였다. 향후 추가 연구를 통해 처리 속도를 좀 더 향상시킬 것이다.

Keyword : 지능형 자동차(intelligent vehicle), 스마트폰(smart phone), iPhone 3GS, 차선이탈경보 어플리케이션(Lane Departure Warning System)

1. 서론

지능형 자동차에 대한 연구는 오래전부터 연구되어 왔으며 이미 높은 수준에 도달하였다. 그러나 이러한 장애물 탐지, 차량제어 등의 지능형 기술들은 일부 고가의 자동차에서만 Before-Market용으로 탑재되어 시판되고 있으며, After-Market용으로는 네비게이션과 블랙박스가 전부라고 할 수 있다. 특히 블랙박스의 수요는 폭발적인 인기를 얻고 있는데, 이는 사고 발생시 원인을 분석할 수 있는 장치뿐만 아니라 운전자의 부주의로 발생할 수 있는 교통사고를 미연에 방지할 수 있는 지능형 장치에 대한 사용자의 요구가 높아지고 있는 것을 알 수 있다[1].

이러한 상황에서 최근 음성통화, 카메라, 메시지 등의 단순한 기능을 벗어나 다양한 분야에 적용되고 있는 스마트폰(iPhone 3GS)에 차선이탈경보 어플리케이션을 개발하게 되었다. 차선이탈경보 어플리케이션은 일반 사용자들이 쉽고 편리하며 저렴하게 차선이탈경보기능을 사용할 수 있도록 하고자 하였으나, iPhone 3GS라는 제한된 플랫폼에서 구현하다보니 속도면에서 만족스러운 결과를 얻지 못하였다. 본 연구는 이러한 어플리케이션의 속도 문제를 개선

하고자 메모리 최적화에 대한 방법과 그에 대한 결과를 도출하였다.

본 논문의 구성은 다음과 같다. 2장에서는 차선이탈경보 어플리케이션에 대해 살펴보고, 3장 실험방법에서는 속도를 개선시킬 수 있는 방법을 제시하였다. 4장에서는 실험을 통한 개선정도를 나타내었고, 5장에서 그에 따른 결론을 도출하였다.

2. 차선이탈경보 어플리케이션

차선이탈경보 어플리케이션은 그림 1과 같이 차량에 스마트폰 거치대를 설치하여 사용하게 된다.



그림 1. 차선이탈경보 어플리케이션의 구현 모습

거치대에 iPhone을 올려놓으면, iPhone의 후방 카메라에 입력되는 전방 도로 영상에서 연속적으로 차선을 추출하고 양차선의 교점이 일정 범위를 넘어가는 경우 차선이탈로 판단하여 영상과 경고음을 통해 운전자에게 경고 상황을 알려준다.

차선인식기능 구현을 위해서는 영상처리기능이 기본적으로 필요한데 이를 위해서는 영상처리 라이브러리인 OpenCV를 사용하였다. OpenCV는 Intel에서 제공하는 컴퓨터비전 라이브러리로 컬러공간처리, 화소값 기반 처리, 에지 추출, 영상 분석, 카메라 교정, 기계 학습 알고리즘 등 사용하기 쉬운 라이브러리를 실시간 연산이 가능하도록 연산의 효율성을 최대한 고려한 라이브러리이다. 또한 C와 C++로 작성된 함수로 제공하기 때문에 Windows 환경뿐만 아니라, Mac OS, Linux 플랫폼에서도 쉽게 이식할 수 있는 장점이 있다. 차선인식 알고리즘은 Hough Transform을 사용하였다. Hough Transform은 영상에서 직선, 원 또는 다른 간단한 모양을 찾는 방법이며[5], OpenCV에서는 Standard Hough Transform과 Progressive Probabilistic Hough Transform을 제공한다. 본 연구에 쓰인 Hough Transform 방식은 Standard Hough Transform 방식이다[6]. 그 외에 Noise Reduction, Canny Edge 등을 사용하였는데, OpenCV 라이브러리를 iPhone에 연동한 결과 초당 프레임 수는 1.52fps(0.66s)를 기록하였다. 이 처리 속도로는 상용화하기 쉽지 않기 때문에 처리 속도 향상이 필요하게 되었고, 본 연구에서는 이에 대한 연구 결과를 설명한다.

3. 속도 향상 방법

3.1. double형을 float형으로 변환

iPhone 3GS의 CPU는 부동 소수점 연산을 효율적으로 처리하는 장치인 부동소수점 장치 FPU(Floating Point Unit)를 자체 지원하기 때문에 정수인 int형과 실수형 float과의 성능 차이는 크게 나지 않는다. 하지만 float형과 double의 두 실수형은 FPU에서 자동으로 라운딩하여 크기를 변경해주는 하지만 double형은 float보다 메모리를 2배 사용하기 때문에 속도에 영향을 줄 수 있다. 따라서 double형으로 선언된 변수들을 float으로 변환하여 메모리 점유율을 줄이고, 속도 향상도 시도하였다.

3.2. cvSetImageROI 사용

차선의 추출에 사용된 Hough Transform 함수는

연산속도가 오래 걸리는 단점이 있기 때문에 관심영역만 연산할 수 있는 ROI(Region of Interest)를 사용하였다. iPhone 3GS를 차량용 거치대에 설치하게 되면, 영상의 상단 지점에는 도로 영상이 아닌 영상이 잡히기 때문에, ROI 함수의 사용하여 영상의 상단 지점은 연산에서 제외하였다.

3.3. CGColorSpaceCreateDeviceGray() 사용

iPhone 3GS의 카메라에서 실시간으로 받아오는 영상은 UIImage 구조체에 저장되게 된다[2]. 하지만 OpenCV는 IplImage 이미지 구조체를 사용하기 때문에 OpenCV 함수를 사용하기 위해서는 iPhone SDK에 정의된 UIImage 구조체를 IplImage 구조체로 변형시켜야 한다. 변형되는 과정에서 영상의 색상공간이 전달되게 되는데 영상의 RGB값을 전달하는 CGColorSpaceCreateDeviceRGB() 함수를 CGColorSpaceCreateDeviceGray()로 변경하게 되면, 속도향상을 가져올 수 있다. iPhone 환경에서 OpenCV 함수를 사용하는 것은 외부 라이브러리에 접근하는 것이기 때문에 부담이 되는 작업이다. 또한 차선을 추출하는 전처리 과정에서 이미지를 바이너리 영상으로 변환해야 하는데, 이를 OpenCV 함수를 사용하여 변환하는 것보다 최초의 영상을 받아들 때, iPhone SDK의 메소드를 호출하여 바이너리 영상으로 받아오게 되면 부담을 크게 줄일 수 있게 되어 속도가 향상되게 된다.

3.4. 이미지 크기 조절

iPhone 3GS에 탑재는 카메라는 320×480 해상도를 지원하는데 160×240로 축소시킨 영상을 대상으로 처리하면 속도를 향상시킬 수 있다. 이미지의 해상도를 줄이게 되면 차선추출의 오차율에 영향을 줄 수 있으나 도로영상은 전후 프레임의 영상변화가 크지 않은 특성 때문에 해상도를 줄여도 차선인식 오차율은 크게 변하지 않았다.

4. 실험결과

3장에서 제시한 방법을 통해 최적화하기 전과 후의 처리 속도의 개선 정도는 표 1,2와 같다. 표 1은 320×480 해상도 입력 영상에 대해 double형을 float형으로의 변환과 CGColorSpaceCreateDeviceGray() 함수로 변경, 관심영역 연산 수행의 수행 결과이다. 3가지 방법을 적용한 결과 한 프레임당 0.22초가 단축되어 기존의 1.52fps에서 2.27fps로 초당 프레임

속도가 향상되었다. 표 2는 입력 영상의 해상도를 160 × 240로 축소한 후 표 1과 같이 3가지 방법을 적용한 결과이다. 한 프레임당 0.26초의 속도가 단축되어 기존의 1.52fps에서 3.84fps로 초당 프레임 속도가 향상되었다. 처리 속도 측정은 실시간 이미지 정보가 담긴 UIImage구조체가 생성되는 시점부터 Hough Transform 연산을 이용하여 차선을 추출하고, cvLine 함수를 이용하여 차선을 화면에 나타내는데 걸리는 시간을 기준으로 하였으며 시간값은 NSLog를 이용한 디버깅을 통하여 계산하였다.

표 1. 처리 속도 향상 결과 (해상도 : 320 × 480)

최적화 방법	Time
double형 → float 형	-0.06s
전체영역 연산 → 관심영역 연산	-0.14s
CGColorSpaceCreateDeviceRGB() → CGColorSpaceCreateDeviceGray()	-0.02s
한 프레임 처리 속도	0.44s
속도 향상 결과	0.22s 개선

표 2. 처리 속도 향상 결과 (해상도 : 160 × 240)

최적화 방법	Time
double형 → float 형	-0.04s
전체영역 연산 → 관심영역 연산	-0.01s
CGColorSpaceCreateDeviceRGB() → CGColorSpaceCreateDeviceGray()	-0.13s
한 프레임 처리 속도	0.26s
속도 향상 결과	0.40s 개선

표 1,2의 결과와 같이 변수 형변환, CGColorSpace CreateDeviceGray() 함수 사용, 관심영역 연산은 애플리케이션의 처리속도를 개선하였다. 입력 영상의 해상도가 160×240인 경우 320×480 보다 처리속도가 크게 개선된 것을 알 수 있었다. 축소된 해상도에서의 차선인식 오차율에 문제가 없다면 처리 속도 향상을 위해 160×240 해상도를 적용할 계획이다. 이외에도 OpenCV 헤더파일의 수정, 함수 호출 변환 등의 과정을 거친 뒤의 결과를 초당 프레임 수로 환산하면 기존의 처리 속도인 1.52fps(0.66s)에서 최소 2.27fps(0.44s)에서 최대 3.84fps(0.26s)로 향상된 것을 확인되었다.

5. 결론

본 연구는 아이폰 기반의 차선이탈경보 애플리케이션의 속도 향상을 연구하였고, 제안된 방법을 통해 속도 개선을 확인하였다. 도로 주행 중 실시간으로 위험을 감지해야 하는 애플리케이션의 특성을 만족하기 위해서는 인식률의 향상과 함께 처리 속도 개선이 필수적이었다.

실험결과 애플리케이션의 처리속도는 최적화 작업을 거쳐 총 0.40초 단축되었으며, 육안으로도 크게 개선된 것을 확인할 수 있었다. 향후 차선 인식 알고리즘을 개선, 실행 중 전화받기 기능이나 타 애플리케이션과의 멀티태스킹 등의 기능을 추가하고, 유저 인터페이스를 개선하는 등의 보완을 거친 뒤에 앱스토어에 등록할 예정이다.

참고문헌

- [1] 윤호영, 김종호, 김한솔, 노광현, “아이폰 기반의 차선이탈 어플리케이션 개발”, 제 34회 한국정보처리학회 춘계학술발표대회 논문집 제 17권 제 2호, 2010.
- [2] <http://developer.apple.com/library/ios/navigation/>
- [3] <http://www.opencv.co.kr>
- [4] Feixiang Ren, Jinsheng Huang, Mutsuhiro Terauchi, Ruyi Jiang, Reninhard Klette “Lane Detection on the iPhone”, Multimedia Imaging Report 43, 2009.
- [5] 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.
- [6] 황선규, “Learning OpenCV 제대로 배우기”, 한빛미디어, 2009