

스마트폰 기반의 도로 밝기 측정 어플리케이션

최영환*, 염효섭**, 박두순*, 홍민*
 *순천향대학교 컴퓨터소프트웨어공학과
 **순천향대학교 컴퓨터학과

e-mail:compust, sekachuu, parkds, mhong@sch.ac.kr

Implementation of Lane Luminance Measurement Application using Smartphone

Young-Hwan Choi*, HyoSub Yum**, Doo-Soon Park*, Min Hong*
 *Dept. of Computer Software Engineering, Soonchunhyang University
 **Dept. of Computer Science, Soonchunhyang University

요 약

최근까지의 교통사고 통계에 따르면 주간 보다는 야간에 더 많은 사고가 발생하며, 사고 원인 중 하나는 부적절한 조명 시설로 인한 시야 미확보, 눈의 피로감 증가가 중요한 요소이다. 본 논문에서는 스마트폰 기반으로 야간 도로 밝기 측정 어플리케이션을 구현하여 야간에 부적절한 조명 시설이 설치된 지점을 파악하기 위해서 위치 정보와 밝기 정보, 이동 방향 정보를 실시간으로 데이터베이스에 저장하여 모니터링해주는 시스템을 설계 및 구현하였다. 이를 위해, 안드로이드 NDK을 이용하여 Native 환경에서 차선 검출 및 RGB 색 공간의 값을 휘도 값으로 변환 알고리즘을 작성하였다. 또한 스마트폰의 카메라를 이용하여 실시간으로 도로 영상을 입력 받은 후 ROI 설정 하여 연산 속도를 향상시켰고, 차선 검출을 통해 차선 사이의 도로 밝기 정보를 획득하고 GPS 센서를 이용하여 이동 방향 정보와 위치 정보를 획득하여 데이터베이스에 저장하였다. 어플리케이션 사용자들의 주행 정보를 바탕으로 도로 밝기 값 데이터베이스를 활용하여 눈부심 및 조명관련 사고 위험 구간을 자동으로 알려주는 후속 연구를 진행할 수 있을 것으로 기대한다.

1. 서론

2008년부터 2013년 동안의 교통사고 통계에 따르면 주간보다 야간에 교통사고가 6% 더 많이 발생하며, 표 1은 주·야간별 교통사고 통계를 나타낸 것이다[1]. 운전자의 부주의가 가장 큰 요인이나, 부적절한 가로등 설치로 인한 시야 미확보와 눈의 피로감 증가 등 조명으로 인한 원인도 존재한다. 이러한 문제를 해결하고자 본 논문에서는 야간 환경에서 스마트폰을 이용하여 도로 밝기를 측정할 수 있는 어플리케이션을 구현하였다. 구현한 어플리케이션은 영상처리를 이용하여 차선 검출을 검출하고 검출된 차선을 기반으로 노면의 밝기 값을 측정하여 저장하며, GPS 센서를 이용한 위·경도값, 방위각 및 속도 정보를 데이터베이스에 저장하도록 구현하였다.

<표 1> 주·야간별 교통사고 통계 (단위 : 건)

년도	2009	2010	2011	2012	2013	합계
주	2,566	2,392	2,325	2,474	2,288	12,045 (47%)
야	2,959	2,843	2,686	2,691	2,588	13,767 (53%)
합계	5,525	5,235	5,011	5,165	4,876	25,812 (100%)

2. 관련 연구

2.1 차선 검출

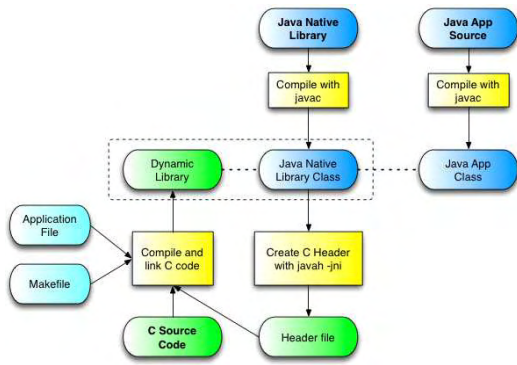
차선을 검출하는 방법은 기존에 많은 연구가 진행되어 오고 있으며 그 방법은 매우 다양하다. 카메라 기반 야간 차선 인식을 개선하기 위한 영상처리 알고리즘에 대한 연구 [2]에서는 외곽선 검출기에 따른 외곽선 검출의 효과를 비교하였고, 일반화 히스토그램 스트레칭을 이용하여 야간 환경에서 인식을 개선하는 연구를 하였다. 해당 논문에서 제안한 알고리즘은 ROI를 10단계로 나누어 설정하고 차량의 좌우 차선에 각각 독립된 마스크를 적용하여 윤곽선 검출을 최대화 하였다. 개선된 밝기 및 대비가 차선 인식률 및 인식 거리 향상을 실험을 통해 검증하였다.

노면 상태 변화에 강인한 에지 검출을 통한 차선 인식에 대한 연구[3]에서는 강인한 차선 인식을 위해 Canny edge 검출기를 사용할 때 환경변화에 파라미터를 조정하는 방법을 적용하여 특징 정보를 정확하게 추출하는 방법을 제안하였고, 도로 상태에 따라 노면과 차선 사이의 명암 값과 에지 검출 시 파라미터의 값을 적절한 값으로 조절하여 차선 인식률을 높였다.

2.2 안드로이드 NDK

안드로이드 NDK는 C, C++로 작성된 라이브러리를 자바로 구현된 안드로이드 환경에서 사용할 수 있는 개발환경

경을 제공한다. 이는 Native 코드로 구동되기 때문에 Java 로 작성된 코드보다 빠른 명령 처리가 가능하며 아래 그림 1은 안드로이드 NDK의 구조[5]이다.

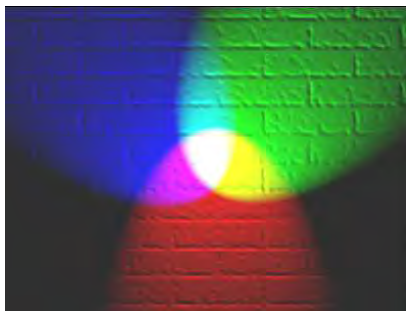


(그림 1) NDK의 구조

안드로이드 NDK를 이용한 어플리케이션 속도 향상 예측기[4] 논문에서는 안드로이드 NDK를 이용할 때 Java 와 C의 연산 속도를 각각의 연산되는 데이터의 자료형마다 실험을 통해 비교하였다. 안드로이드 NDK를 사용할 때 연산속도가 문자열 연산을 제외한 다른 데이터의 타입의 연산인 경우에 더 빠른 것으로 검증되었다. 해당 논문에서는 연산 속도를 빠르게 하기 위한 방법이 제시되었고 안드로이드 NDK를 이용할 때의 성능 향상을 실험을 통해 확인하였다. 따라서 본 논문에서는 안드로이드 NDK를 사용하여 이미지 처리 연산속도를 향상시키기 위해서 안드로이드 NDK를 이용하여 어플리케이션을 개발하였다.

2.3 색 모델

도로의 밝기를 측정하기 위해 본 논문에서는 입력 영상에 대해 영상처리 방법을 적용하였다. 영상처리에서는 지정한 색 모델을 기본으로 하여 이미지들의 정보들을 처리하게 되며, 본 논문에서는 사용자에게 보여주기 위한 화면은 RGB 색 공간을 적용하였고, 밝기값 측정하기 위해 YCbCr 색 공간을 사용하였다. 아래 그림 2와 같이 RGB 색 공간[6]은 색을 혼합하면 명도가 올라가는 가산 혼합 방식으로 빨강, 녹색, 파랑색으로 표현된다.

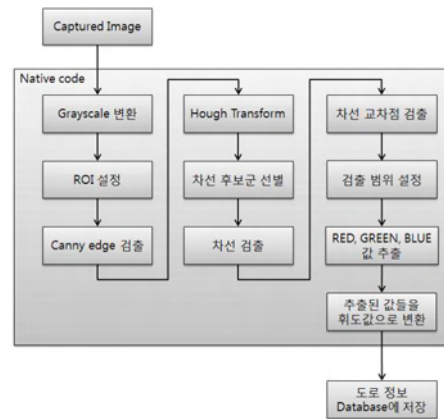


(그림 2) RGB 색 공간

도로의 밝기 정도를 판단하기 위한 값을 구하기 위해 RGB 색 공간을 YCbCr 색 공간[7]으로 식 1을 사용하여 변환하였으며, Y는 휘도, Cb과 Cr는 색차 성분을 의미한다. 따라서 본 논문에서는 Y 값을 도로의 밝기 정보로 사용하였다.

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 16 \\ 128 \\ 128 \end{pmatrix} \dots (1)$$

3. 도로의 밝기 측정 알고리즘



(그림 3) 도로 밝기 측정 알고리즘

그림 3은 본 논문에서 구현한 도로 밝기 측정 알고리즘의 흐름도이다. 먼저 차량에 설치된 스마트폰의 카메라를 이용하여 실시간으로 차량의 주행 영상을 입력받고, 입력된 영상은 프레임 단위의 이미지로 메모리에 임시 저장되어 Native 코드 영역으로 복사된다. 복사된 이미지 데이터는 윤곽선 검출을 위하여 이진화 이미지(Binary Image)로 변환한다.

영상처리에 사용하는 기능들은 일반적인 프로그램 기능 보다 많은 연산을 필요로 한다. 일반 PC보다 제한적인 성능을 가진 스마트폰에서 연산 속도를 향상시키기 위해 필요한 영역만 연산을 수행하는 것이 중요하다. 그런데 일반적인 경우, 주행 중인 차량의 영상에서 차선의 위치는 주로 영상의 중앙과 하단 부분에 위치한다. 따라서 해당 영역을 관심영역(Region Of Interest)으로 설정하여 그 외의 부분에서의 불필요한 차선 검출 연산을 막을 수 있도록 설정하고, 관심영역에서 차선의 윤곽선 검출을 위하여 Canny 에지 검출기를 적용하였다. Canny 에지 검출기는 두 개의 임계값을 사용하여 차선 검출에 불필요한 잡음을 필터링 한다. 본 논문에서는 차선 이외의 잡음을 효과적으로 제거할 수 있도록 두 개의 임계값을 각각 70, 150으로 설정하였다. 임계값이 70 이하, 150 이상이면 잡음으로 판단하여 윤곽선에서 제외되고 70부터 150까지를 윤곽선으로 판별하였다. 그림 4는 입력 영상에 대해 ROI를 설정하고 윤곽선을 추출한 화면이다.



(그림 4) ROI 설정 후 외곽선 추출 화면

잡음이 제거된 윤곽선 이미지에서 직선 검출을 위하여 Hough 직선변환을 적용한다. Hough 직선 변환은 한 평면 위에 있는 점들의 집합 중에 특별한 조건을 만족시키는 직선을 찾는 알고리즘으로서, 차선은 직선으로 이루어져 있으므로 Hough 직선변환 알고리즘을 사용하였다. Hough 직선변환을 적용하여 검출된 직선들은 차선의 후보로 설정되며, 이 중 실제 차선은 추출된 차선 후보 중에서 검출된 직선의 기울기 Radian value가 0.5 ~ 1.1의 범위에 있으면 좌측 차선으로 판정하고, 2보다 크고 3보다 작으면 우측 차선으로 판정하여 검출하였다. 이렇게 검출된 두 차선의 교차점은 약 60m로 제한속도가 60km/h인 국도에서의 안전거리에 해당되고 이를 통하여 주행하고 있는 차량의 60m 앞부분의 도로영역을 찾을 수 있다.

아래 그림 5의 파랑색 삼각형 부분과 같이 도로의 밝기를 측정하기 위해서 검출된 두 직선의 교차점의 좌표를 구한 후 검출된 교차점부터 아래 20픽셀 까지 빗금친 부분을 도로 밝기 측정 범위로 설정하여 왼쪽부터 오른쪽 방향으로 모든 픽셀의 RGB 값을 각각 추출하고 식 1을 이용하여 밝기 값인 Y값을 계산하였다.

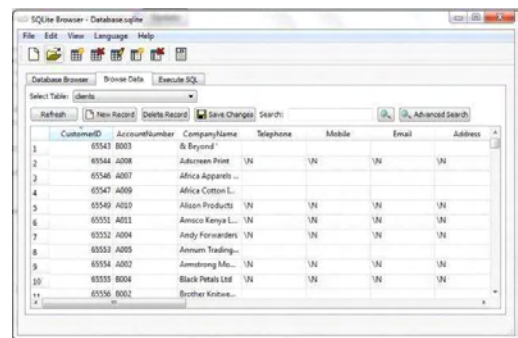


(그림 5) 밝기 측정 화면

4. 도로 정보 및 주행 정보 저장

본 논문에서는 좌·우 차선이 정상적으로 검출될 경우 매 초마다 측정 범위의 모든 픽셀의 휘도 값의 밝기 정보와 스마트폰의 GPS 센서를 이용하여 위·경도, 주행 속도, 주행 방위각 정보를 SQLite 데이터베이스에 저장하였다[8]. SQLite는 대규모의 데이터 관리보다는 중소규모 작업에 적합하고 SQLite API는 라이브러리를 호출만 가능하여 프로그램의 연산속도를 향상 시킬 수 있다[9]. 그림 6은 SQLite Database Browser로 다른 데이터베이스와 달리 간편하게 프로그램을 이용하여 저장된 데이터베이스 내용을 확인 및 분석 할 수 있다.

본 논문의 어플리케이션은 위치 정보와 밝기 정보, 이동 방향 정보들을 데이터베이스에 저장하여 추후 도로 데이터베이스를 분석하여 도로의 정보를 모니터링 할 수 있도록 하였다.



(그림 6) SQLite Database Browser

5. 측정 결과

그림 7은 본 논문에서 구현한 도로 밝기 측정 어플리케이션의 실행 화면이다. 차선이 검출되면 빨강색 직선으로 삼각형 모양으로 출력하였으며 이 때 도로의 밝기를 측정한다. 측정된 밝기 값에 따라 화면 우측 상단에 파랑색으로 도로의 밝기를 표시하였고, 60m 앞부분의 도로 영역의 평균 밝기값을 0~100 사이의 값으로 변환한 후 평균값을 출력하였다. 좌측 상단에는 초당 연산되는 프레임수와 위도, 경도, 방위각, 시속을 보여준다.



(그림 7) 도로 밝기 측정 어플리케이션

6. 결론

본 논문에서는 영상처리를 이용한 스마트폰 기반의 도로 밝기 측정 어플리케이션을 구현하였다. 본 논문의 도로 밝기 측정 어플리케이션을 활용하여 도로의 밝기 정보를 특수한 다른 전문 장비의 설치 없이 수집할 수 있었다. 추후 사용자들이 본 어플리케이션을 활용하여 직접 측정한 도로 밝기값 데이터들을 실시간으로 공유하여 눈부심 및 조명 관련사고 위험 구간을 알려주는 어플리케이션으로 개발된다면 운전자의 야간 운전에 큰 도움이 될 수 있을 것으로 기대된다.

참고문헌

- [1] 도로교통공단 교통사고분석시스템 “<http://taas.koroad.or.kr/service/accidentInfo/accidentMain.jsp?link=accident3>”
- [2] 김홍룡, 이선봉 “카메라 기반 야간 차선 인식률 개선을 위한 영상처리 알고리즘에 대한 연구” 한국자동차공학회, 한국자동차공학회논문집, 2013, pp.51-60
- [3] 권보철, 신동원, 박경석 “노면 상태 변화에 강인한 예지 검출을 통한 차선 인식에 대한 연구” 제26회 ICROS 학술대회, 2011.5, pp.366-370
- [4] 손기철 “안드로이드 NDK를 이용한 어플리케이션 속도 향상 예측기” 전북 대학교 대학원, 2012
- [5] 안드로이드 NDK 구조 “<http://blog.xieles.com/wp-content/uploads/2014/03/jni.png>”
- [6] RGB 색 공간 “http://en.wikipedia.org/wiki/RGB_color_model”
- [7] YCbCr 색 공간 “<http://ko.wikipedia.org/wiki/YCbCr>”
- [8] Lynn Beighley “Head First SQL : 효율적인 DB 관리를 위한 SQL 학습법” 한빛미디어, 2008
- [9] SQLite “<http://ko.wikipedia.org/wiki/SQLite>”