

안드로이드 기반의 도로 밝기 측정 어플리케이션 구현[☆]

A Road Luminance Measurement Application based on Android

최영환¹ 김홍래² 홍민^{3*}
Young-Hwan Choi Hongrae Kim Min Hong

요약

최근 5년간의 주·야간별 교통사고 통계에 따르면 대부분의 자동차 교통사고는 주간보다 야간에 더 많이 발생했다. 교통사고는 다양한 원인으로 발생하게 되는데 그 중 중요한 요소는 조명 시설 위치와 시설 구역을 파악하고 관련 정보들을 데이터베이스화 하였다. 이를 위해 운전자의 위치 정보, 주행 정보, 도로 밝기 정보를 스마트폰을 이용하여 실시간으로 데이터베이스 서버에 저장하는 도로 밝기 측정 어플리케이션을 설계 및 구현하였다. 본 어플리케이션은 안드로이드 NDK를 이용하여 Native C/C++ 환경에서 구현되었으며, 이에 따라 자바나 다른 언어로 작성된 어플리케이션 보다 연산속도를 향상시켰다. 도로 밝기를 측정하기 위하여 카메라 영상인 RGB 색 공간의 영상을 YCbCr 색 공간의 영상으로 변환하여 휘도를 측정한다. 이를 위해 먼저 차선을 검출하고 도로 밝기 검출 영역의 휘도 값을 계산하여 데이터베이스에 저장한다. 또한 스마트폰의 카메라를 이용하여 실시간으로 도로의 영상을 입력 받고 도로의 차선부분에 대한 관심영역을 지정하여 연산 속도를 향상시켰다. 관심영역의 영상은 Grayscale 영상으로 변환하고 Canny 에지 검출기를 사용하여 외곽선을 추출하고 Hough line transform을 적용하여 차선의 후보군을 선별한다. 선별된 후보 차선의 기울기를 계산하여 양쪽의 차선을 선정한다. 양쪽 차선이 검출되면 차선의 교차점으로부터 아래로 20픽셀의 높이를 가진 삼각형을 도로 밝기 측정 범위로 설정한다. 삼각형 부분의 모든 픽셀에 대한 R, G, B 값을 추출하여 Y 값을 계산하고 픽셀 밝기 값의 평균을 0부터 100사이의 값으로 계산하여 검은색부터 초록색으로 도로의 밝기를 표현하였다. 계산된 60m 전방의 도로 밝기 값은 스마트폰의 GPS 센서를 통해 측정된 운전자의 주행 정보와 위치 정보를 획득하여 10분 간격으로 무선통신을 통해 데이터베이스 서버에 저장하였다. 향후 수집된 도로 밝기 정보들은 스마트폰 어플리케이션이나 차량 내비게이션을 통해 운전자들에게 조심 운전을 경고하거나 효율적인 도로 조명 관리를 위한 개보수 계획에 반영될 수 있을 것으로 기대된다.

☞ 주제어 : 영상처리, 차선 검출, 안드로이드 NDK, 스마트폰, 도로 밝기 측정, GPS 센서

ABSTRACT

According to the statistics of traffic accidents over recent 5 years, traffic accidents during the night times happened more than the day times. There are various causes to occur traffic accidents and the one of the major causes is inappropriate or missing street lights that make driver's sight confused and causes the traffic accidents. In this paper, with smartphones, we designed and implemented a lane luminance measurement application which stores the information of driver's location, driving, and lane luminance into database in real time to figure out the inappropriate street light facilities and the area that does not have any street lights. This application is implemented under Native C/C++ environment using android NDK and it improves the operation speed than code written in Java or other languages. To measure the luminance of road, the input image with RGB color space is converted to image with YCbCr color space and Y value returns the luminance of road. The application detects the road lane and calculates the road lane luminance into the database sever. Also this application receives the road video image using smart phone's camera and improves the computational cost by allocating the ROI(Region of interest) of input images. The ROI of image is converted to Grayscale image and then applied the canny edge detector to extract the outline of lanes. After that, we applied hough line transform method to achieve the candidated lane group. The both sides of lane is selected by lane detection algorithm that utilizes the gradient of candidated lanes. When the both lanes of road are detected, we set up a triangle area with a height 20 pixels down from intersection of lanes and the luminance of road is estimated from this triangle area. Y value is calculated from the extracted each R, G, B value of pixels in the triangle. The average Y value of pixels is ranged between from 0 to 100 value to inform a luminance of road and each pixel values are represented with color between black and green. We store car location using smartphone's GPS sensor into the database server after analyzing the road lane video image with luminance of road about 60 meters ahead by wireless communication every 10 minutes. We expect that those collected road luminance information can warn drivers about safe driving or effectively improve the renovation plans of road luminance management.

☞ keyword : Image processing, lane detection, android NDK, Smartphone, Road luminance measurement, GPS sensor

¹ Department of Computer Science, Soonchunhyang University, Asan-si, Chungcheongnam-do, 336-745, Korea.

² Department of Electronics and Information Engineering, Soonchunhyang University, Asan-si, Chungcheongnam-do, 336-745, Korea.

³ Department of Computer Software Engineering, Soonchunhyang University, Asan-si, Chungcheongnam-do, 336-745, Korea.

* Corresponding author (mhong@sch.ac.kr)

[Received 31 January 2015, Reviewed 9 February 2015, Accepted 7 April 2015]

☆ 본 연구는 순천대학교 학술연구비 지원으로 수행하였음

☆ 본 논문은 2014년도 인터넷정보학회 추계학술발표대회우수논문 추천에 따라 확장 및 수정된 논문임

1. 서 론

도로 교통 공단에서 제공하는 2009년부터 2013년까지의 주·야간별 교통사고 통계[1]에 따르면 주간에 발생한 교통사고 보다 야간에 발생한 교통사고의 건수가 6% 더 많이 발생했다. 표 1은 주·야간별 교통사고 통계를 표로 나타낸 것이다.

교통사고의 원인은 주로 과속운전, 음주운전, 운전미숙 및 부주의 등의 운전자 원인과 도로 환경, 차량 고장 등이 있다. 도로 환경 문제는 손상된 도로와 어두운 조명 시설과 조명 부재의 원인들이 있다. 도로에 대한 가로등 미설치 및 부적절한 설치는 운전자의 시야 혼란을 유발시켜 대형 교통사고 원인이 될 수 있다. 본 논문은 이러한 도로의 환경 문제를 분석하기 위해서 야간 환경의 도로 밝기를 스마트폰 비디오 영상을 통해 측정 하는 어플리케이션을 설계 및 구현하였다.

(표 1) 주·야간별 교통사고 통계
(Table 1) Statistics of car accident in the daytime and nighttime

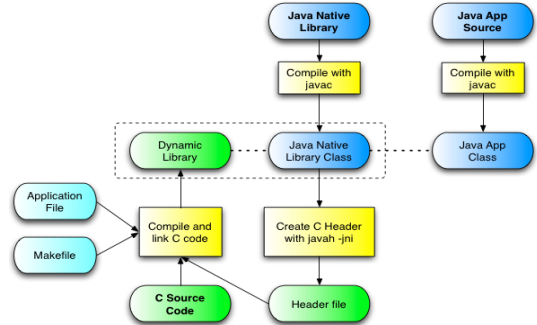
(단위 : 건)

년도	2009	2010	2011	2012	2013	합계
주간	2,566	2,392	2,325	2,474	2,288	12,045 (47%)
야간	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%)

본 논문에서 구현된 어플리케이션은 차량에 스마트폰을 설치하여 도로를 주행하면서 실시간으로 도로의 정보들을 수집하게 된다. 카메라 영상을 통하여 주행하고 있는 도로의 차선 검출을 통하여 올바른 주행 방향 파악과 사용자 시야에서 전방에 보이는 도로의 밝기를 측정한다. 실시간으로 저장되는 데이터는 스마트폰의 GPS 센서를 통하여 차량의 위치 계산을 위한 현재 위·경도, 주행 각, 주행 속도를 무선 인터넷을 통해 일정 시간별로 데이터베이스 서버에 저장한다.

2. 관련 연구

2.1 안드로이드 NDK



(그림 1) 안드로이드 NDK의 구조
(Figure 1) Structure of Android NDK

안드로이드 NDK(Native Development Kit)는 안드로이드 환경에서 C/C++ 언어로 코드를 작성하여 사용할 수 있는 개발환경을 제공한다. C/C++의 코드를 컴파일하고 Java native interface를 통해서 코드를 동작할 수 있도록 한다. Java로 작성된 코드는 Virtual machine에서 동작하고 C/C++로 작성된 코드는 Native로 동작하기 때문에 NDK를 사용하면 Java 코드보다 C/C++ 코드가 일반적으로 연산속도가 더 빠르다. 따라서 CPU에 연산이 집중되는 게임 엔진, 신호 처리와 같은 어플리케이션 개발 환경에서 주로 사용한다. 아래 그림 1은 안드로이드 NDK의 구조[2]를 보여주고 있다.

기존의 Android 플랫폼에서 구현 기술에 따른 화질 개선 연산 성능 비교 연구[3]에서는 안드로이드 환경에서 이미지의 명암비 신장 연산을 사용하여 영상의 질을 향상 시키는 방법을 C/C++, 자바, SIMD(Single Instruction Multiple Data) 기반의 NEON을 사용하여 구현하고 연산 시간을 비교하였다. 실험 결과 자바 코드보다 C/C++와 NEON 코드의 연산 속도가 각각 55-78배, 50-76배 향상된 것으로 나타났다. 따라서 본 논문에서는 비디오 이미지 영상의 질을 향상시키고 처리 속도를 향상시키기 위해서 안드로이드 NDK를 사용하여 C/C++코드를 작성하여 도로 밝기 측정 어플리케이션을 구현하였다.

2.2 관심 영역

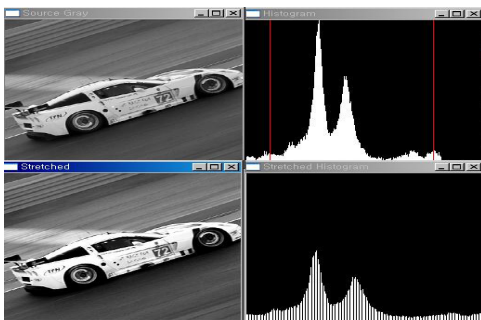
영상처리 알고리즘은 일반적으로 입력 이미지의 모든 화소 또는 화소의 그룹에 대하여 연산을 수행되며 일반

적인 PC의 환경에서도 많은 연산 시간이 소요된다. 특히 스마트폰은 PC에 비해 연산 성능이 월등히 떨어지며 따라서 영상처리 알고리즘을 수행하기 위한 최적화가 필요하다. 불필요한 연산 시간을 감소시키고 연산 알고리즘을 최적화하는 것이 매우 중요하다.

차선 검출을 위하여 주행 중인 차량에서 스마트폰으로 촬영한 도로 영상에서 차선의 위치는 대부분 영상의 중앙과 하단 부분에 위치하게 된다. 본 논문에서는 스마트폰을 차량에 설치하고 차선 검출을 위하여 카메라가 하·중단 부분을 촬영하도록 고정시키고, 차선이 영상의 중앙과 하단에 위치한다는 가정 하에 이 부분을 관심영역으로 설정하여 불필요한 부분에 대한 연산을 제거하였다. 따라서 전체 입력 영상 640×480 크기의 이미지에서 ROI(Region of Interest)의 크기를 426×240 픽셀로 설정하여 이 영역에서 차선을 검출하였다.

2.3 차선 검출

차선 검출은 많은 영역에서 사용이 되는 기초 기술로 자동 주행, 차선 이탈방지와 같은 다양한 연구에 사용되고 있으며 차선 검출에 관하여 많은 방법들이 연구되고 있다. 카메라 기반 야간 차선 인식을 개선하기 위한 영상처리 알고리즘에 대한 연구[4]에서는 에지 검출기의 에지 검출 결과를 비교하고, 일반화 히스토그램 스트레칭을 통하여 영상의 밝기를 조절하여 야간 환경에서 차선 인식을 개선하였다. 해당 연구의 야간 차선 인식 및 인식 거리 개선을 위한 알고리즘은 관심영역을 10단계로 설정하고 각각의 관심영역에 독립된 각도별 마스크를 적용하여 차선 검출을 최대화 하고 실험을 통해 검증하였다. 아래의 그림 2는 일반화 히스토그램 스트레칭의 예 [5]를 나타낸 것이다.



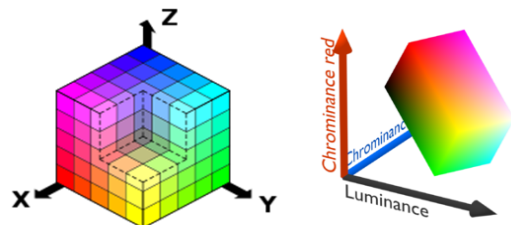
(그림 2) 일반화 히스토그램 스트레칭의 예
(Figure 2) An example of a generalization of histogram stretching

노면 상태 변화에 강인한 에지 검출을 통한 차선 인식에 대한 연구[6]에서는 차선 인식률을 향상시키기 위해 Canny edge 검출기를 사용할 때 빛의 양, 촬영 환경 등의 환경 변화에 맞도록 변수를 조절하여 차선을 정확하게 추출하는 방법을 제안하였고, 도로의 노면 상태와 차선 사이의 명암 값에 따라 에지 검출 시 파라미터를 적절하게 조절해 차선 인식률을 향상 시켰다.

도로 환경 변화에 강인한 차선 검출 방법[7]에서는 도로의 노후로 인한 페인트 손상, 기상 환경 등의 환경 변화에 강인한 차선 검출 알고리즘을 제안하였다. 영상에서 차선의 폭을 이용하여 후보 영역을 추출하고 추출된 영역을 모델링 기법을 적용하여 차선을 판단하였다. 차선 정보를 기준으로 차선 후보 영역에 가중치를 부여하여 흐린 차선도 검출 가능하게 하였다. 제안한 알고리즘을 실험을 통해 검증하였으며 평균 92%의 정확도를 나타내었다.

2.4 색 모델

카메라로부터 입력받은 영상은 정의된 색 모델을 기반으로 영상의 정보를 처리하는데, 관련 연구들은 다양한 색 모델들을 사용하여 왔다[8][9][10]. 본 논문에서는 카메라로 받은 입력 영상과 스마트폰에 출력되는 화면은 RGB 색 공간[11]을 사용하였고, 도로의 밝기를 측정하기 위한 색 공간은 YCbCr 색 공간[12][13]을 사용하였다. 다음의 그림 3과 같이 RGB 색 공간은 빨강, 녹색, 파랑색을 기본 색으로 표현하며 3가지 색을 혼합하면 밝아지는 가산 혼합 방식이다.



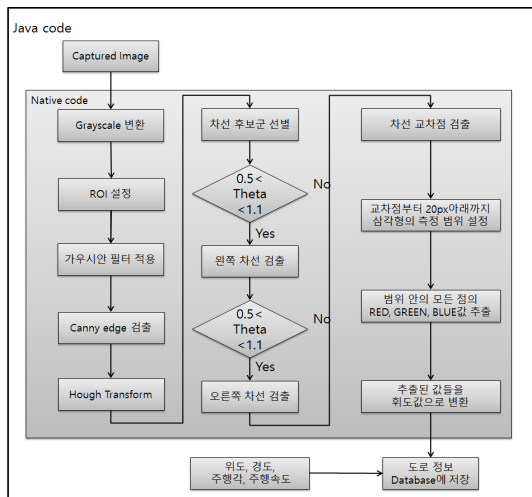
(그림 3) RGB와 YCbCr 색모델
(Figure 3) RGB and YCbCr color model

본 논문에서는 도로의 밝기를 측정하기 위해서 RGB 색 공간을 YCbCr 색 공간으로 아래의 식 1을 사용하여 변환하여 밝기를 계산한다. YCbCr의 Y는 휘도를, Cb과 Cr는 색차 성분을 의미한다. 본 논문에서는 도로의 밝기를 Y의 값으로 사용하였다.

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

3. 도로 밝기 측정 알고리즘

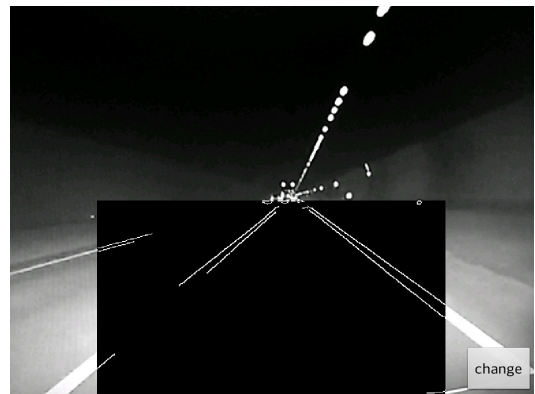
본 논문에서 구현한 어플리케이션은 카메라의 이미지 처리 연산을 효율적으로 할 수 있도록 OpenCV 라이브러리를 사용 하였다. 그림 4는 본 논문에서 구현한 도로 밝기 측정 알고리즘이다. 먼저 카메라로 촬영된 차량의 주행 영상을 입력받는다. 입력된 영상은 640 × 480 픽셀의 크기를 가지며 행렬로 C/C++ 코드 영역으로 복사되어 전달되며 복사된 행렬을 윤곽선 검출을 위하여 grayscale 영상으로 변환한다. 변환된 영상에 Gaussian 필터를 적용해 영상의 잡음을 제거한다. 잡음이 제거된 영상에 연산 속도를 향상시키기 위해 관심영역을 설정한다. 설정된 관심 영역에서 차선의 에지를 검출하기 위해서 Canny 에지 검출기를 이용하였다.



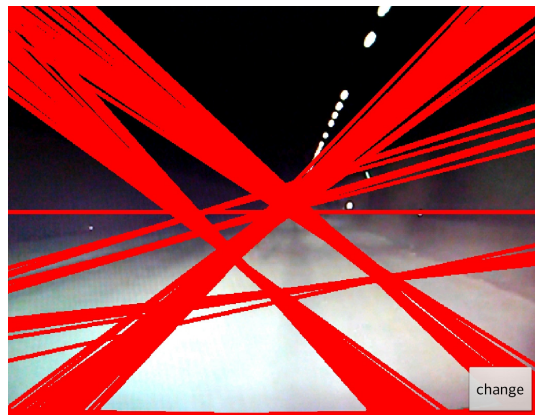
(그림 4) 도로 밝기 측정 알고리즘
(Figure 10) Algorithm of road luminance measurement

Canny 에지 검출기는 더블 임계값 기법을 사용하는데, 이 기법은 잡음을 제거와 차선 검출의 정확도를 향상시킬 수 있다. 본 논문에서는 두 임계값을 각각 70과 150으로 지정하였다. 영상의 값이 70보다 작고 150보다 크면 잡음으로 처리되고 70부터 150사이에 있다면 에지로 판단한다. 다음 그림 5는 카메라 영상에 관심영역을 설정하

고 차선 에지를 검출한 화면이다. 검출된 에지 후보군들에서 직선으로 된 에지를 추출하기 위해서 Hough Line Transform을 적용했다. Hough Line Transform은 2차원 평면에 있는 점들 중에 직선을 찾는 연산이다. 차선은 영상 내에서 직선 형태로 존재하므로 Hough Line Transform을 통해 차선을 검출한다. 이 직선들 중에서 차선이 포함되어 있기 때문에 실제 차선을 선별해야 한다. 그림 6과 같이 다양한 차선의 후보군으로부터 차선을 검출하기 위해 검출된 직선의 기울기 값이 0.5보다 크고 1.1보다 작으면 왼쪽 차선으로 선별하고, 2보다 크고 3보다 작으면 오른쪽 차선으로 선별하였다. 또한 ROI 영역에서 검출된 차선을 확장시켜 그림 7, 9와 같이 전체 이미지 영상에 차선으로 표시하였다.

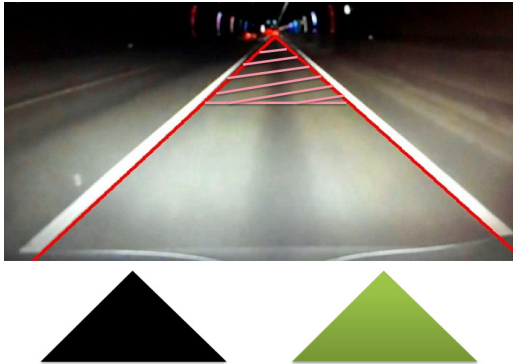


(그림 5) ROI 설정 후 외곽선 추출 화면
(Figure 5) Extracted edges in image of ROI



(그림 6) 차선의 후보군
(Figure 6) Candidate lanes in image

본 논문은 차량의 진조동에 의해 상대적으로 밝은 도로 밝기를 제공하는 부분을 제외하고 차량의 제동 거리를 고려하여 전방 60m 부분의 도로 밝기를 검출하여 차량 운전자들에게 도로 밝기 정보를 제공하도록 설계하였다. 카메라를 통해 입력되는 이미지의 두 차선이 교차하는 부분은 주행하는 차량으로부터 약 60m 전방 지역이므로 이 부분의 도로의 밝기를 측정하였다. 다음 그림 7은 도로 밝기 측정 화면으로 검출된 차선의 교차점의 좌표를 계산하고 이 좌표부터 20픽셀 아래까지 차선 안의 빛금으로 표현된 삼각형 부분에 대한 밝기를 측정하였다. 밝기 측정은 왼쪽에서 오른쪽 방향으로 이동하면서 위에서 아래로 삼각형 내의 모든 픽셀의 R, G, B 값을 각각 추출하여 YCbCr 색 공간으로 변환 후 밝기를 의미하는 Y를 앞의 식 1로 계산하였다.



(그림 7) 도로 밝기 측정 화면

(Figure 7) Measuring road luminance in image

4. 도로 정보 및 주행 정보 저장

필드	종류	Null	기본값
User_ID	varchar(12)	아니오	없음
latitude	double	아니오	없음
longitude	double	아니오	없음
azimuth	double	아니오	없음
velocity	double	아니오	없음
average_luminance	double	아니오	없음
year	int(11)	아니오	없음
month	int(11)	아니오	없음
hour	int(11)	아니오	없음

(그림 8) 데이터베이스 스키마

(Figure 8) Database Scheme

앞서 차선 검출 알고리즘을 통해 검출된 양 차선이 이상 없이 검출되었을 때, 매 1초 마다 삼각형의 측정 범위의 모든 위치의 밝기 값을 계산하고 스마트폰에 내장된 GPS 센서를 통해 위도, 경도, 주행각, 주행 속도를 계산하고 해당 월, 시간을 데이터베이스 서버에 저장하였다. 본 논문에서는 SQLite[14]를 데이터베이스로 사용하였으며 SQLite는 대규모의 작업보다는 모바일 디바이스 환경처럼 중·소규모에 적합하고 SQLite API는 라이브러리 호출만 허용하여 프로그램의 구동 속도를 향상 시킨다. 다음 그림 8은 도로 밝기 측정 어플리케이션의 데이터베이스 스키마이다.

5. 측정 결과

양쪽 차선이 정상적으로 검출되면 차선을 빨강색 직선으로 출력하고 도로의 밝기를 측정하였다. 측정 범위의 평균 밝기 값을 식 2를 이용하여 0과 100사이의 값으로 바꾸어 출력하고 검은색부터 초록색까지 100단계를 그림 7의 아래와 같이 삼각형으로 시각화하였다. la 는 도로의 밝기, i 는 삼각형을 구성하는 각 픽셀을 의미한다. 결과 화면의 좌측 상단에는 초당 연산되는 프레임 수, 위치 정보, 주행 정보를 출력하며 그림 9는 본 논문에서 구현한 도로 밝기 측정 어플리케이션의 모습이다.

$$la = \sum_{i=0}^n (y_i) * \frac{100}{255} \quad (2)$$

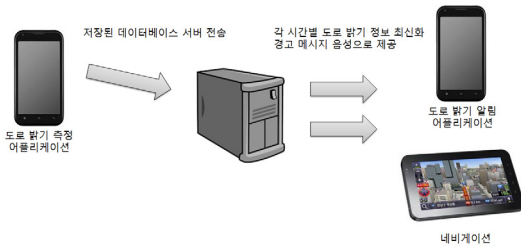


(그림 9) 도로 밝기 측정 어플리케이션

(Figure 9) An image of road luminance measurement application

6. 활용 방안

다음의 그림 10은 최종 시스템의 활용 가능 구성도이다. 본 어플리케이션은 수집된 도로 정보들을 데이터베이스 서버에 매 10분 단위로 무선통신을 통해 매 초마다 동기화되어 저장된 도로 정보들을 전송한다. 서버는 사용자로부터 전송받은 데이터베이스를 저장하며 데이터는 위치, 월, 시간 순으로 정렬한다. 특정 위치의 정보에 대한 월, 시간에 해당하는 데이터가 중복이 발생하면 해당 데이터의 밝기의 평균값으로 업데이트하여 더욱 정확한 정보들이 저장되도록 할 수 있다.



(그림 10) 예상 시스템 구성도
(Figure 10) A block diagram of expected system

이렇게 수집된 정보들은 차량 내비게이션이나 도로 밝기 정보 안내 스마트폰 어플리케이션을 통해 해당 위치에 대한 각 시간별 도로 밝기 정보를 제공하여 운전자에게 위험 구간에 대한 경고 메시지를 음성으로 제공해주는 시스템으로 구현될 수 있어 사용자의 야간 차량 운전에 대한 안전에 큰 도움이 있을 것으로 기대한다.

7. 결 론

본 논문에서는 영상처리 기술을 이용하여 스마트폰 기반의 도로 밝기 측정 어플리케이션을 구현하였다. 본 어플리케이션은 안드로이드 NDK를 사용하여 영상 처리 연산속도를 향상시키고 간단한 연산을 통해 도로의 밝기를 빠르게 측정하였다. 본 논문의 도로 밝기 측정 어플리케이션을 활용하면 다른 특수한 전문 장비의 설치 없이 스마트폰으로 도로의 밝기에 대한 정보들을 수집할 수 있어 수집된 도로 정보를 통해 운전자들에게 조심 운전을 경고하거나 효율적인 도로 조명 관리를 위한 개보수 계획에 반영될 수 있을 것으로 기대된다.

참고문헌(Reference)

- [1] The road traffic authority traffic accident analysis system, <http://taas.koroad.or.kr/service/accidentInfo/accidentMain.jsp?link=accident3>
- [2] Structure of Android NDK, <http://blog.xieles.com/wp-content/uploads/2014/03/jni.png>
- [3] J. Lee, G. Lee, C. Jung, "Performance comparison of implementation technologies for image quality enhancement operations on android platforms", Journal of Digital Contents Society, Vol.14, no.1, pp. 7-14, Mar. 2013. <http://dx.doi.org/10.9728/dcs.2013.14.1.7>
- [4] H. Kim, S. Lee, "A study on image processing algorithms for improving lane detectability at night based on camera", Transactions of the Korean Society of Automotive Engineers, Vol.21, no.1, pp. 51-60, 2013. <http://dx.doi.org/10.7467/KSAE.2013.21.1.051>
- [5] An example of a generalization of histogram stretching, <http://kimhj8574.egloos.com/viewer/5728390>
- [6] B. Gwon, D. Shin, K. Park, "A study on a lane detection using robust edge detection for the road condition changes", Institute of Control, Robotics and Systems, ICROS annual conference, Vol.26, pp. 366-370, May. 2011. <https://dbpia.co.kr/Journal/ArticleDetail/3018453>
- [7] B. Kim, W. Kim, "Robust lane detection method in varying road conditions", The Institute of Electronics Engineers of Korea, Vol.49, no.1, pp. 88-93, Jan. 2012. <http://www.dbpia.co.kr/Journal/ArticleDetail/1618148>
- [8] J. Chun, "Vision-based motion control for the immersive interaction with a mobile augmented reality object", Journal of Internet Computing and Services, Vol.12, no.3, pp. 119-129, June. 2011. <https://dbpia.co.kr/Journal/ArticleDetail/1482931>
- [9] W. Kim, J. Chun, "A Hybrid Approach of Efficient Facial Feature Detection and Tracking for Real-time Face Direction Estimation", Journal of Internet Computing and Service, Vol.14, no.6, pp. 117-12, Dec. 2013. <http://dx.doi.org/10.7472/jksii.2013.14.6.117>
- [10] J. Yoo, S. Ohm, M. Chung, "Maximum-Entropy Image Enhancement Using Brightness Mean and Variance",

Journal of Korean Society Internet Information, Vol.13,
no.3, pp. 61-73, Apr. 2012.

<http://dx.doi.org/10.7472/jksii.2012.13.3.61>

- [11] RGB color space, http://ko.wikipedia.org/wiki/RGB_%EA%B0%80%EC%82%B0%ED%98%BC%ED%95%A9
- [12] YCbCr color space, <http://ko.wikipedia.org/wiki/YCbCr>
- [13] YCbCr color model, <http://en.wikipedia.org/wiki/Talk%3AYCbCr>
- [14] SQLite, <http://ko.wikipedia.org/wiki/SQLite>

● 저 자 소 개 ●



최 영 환 (Young-Hwan Choi)

2015년 순천향대학교 컴퓨터소프트웨어공학과 졸업(학사)

2015~현재 순천향대학교 대학원 △△학과 졸업(석사)

관심분야 : 다이나믹시뮬레이션, 영상처리 etc.

E-mail : compust@sch.ac.kr



김 흥 래 (Hongrae Kim)

1986년 연세대학교 전기공학과 졸업(학사)

1989년 연세대학교 대학원 전기공학과 졸업(석사)

1995년 Texas A&M University 전기공학과 졸업(박사)

1995~현재 순천향대학교 전자정보공학과 교수

관심분야 : 시스템 모델링, 대형 시스템의 컴퓨터 응용 해석

E-mail : hongkim@sch.ac.kr



홍 민 (Min Hong)

1995년 순천향대학교 전산학과(공학사)

2001년 University of Colorado at Boulder(공학석사)

2005년 University of Colorado at Denver(이학박사)

2006년~현재 순천향대학교 컴퓨터소프트웨어공학과 교수

관심분야 : 컴퓨터그래픽스, 다이나믹시뮬레이션, 바이오인포매틱스, 영상처리

E-mail : mhong@sch.ac.kr