

## 자동차용 블랙박스를 이용한 실시간 차량위치추적 시스템의 구현

최홍선 · 김도원 · 채한음 · 이수연 · 최홍섭\*  
대전대학교 전기전자공학부

# Implementation of real-time Vehicle Location Tracking System using a vehicle black box

Hong Sun Choi · Do Won Kim · Han Um Chae · Soo Yeon Lee · Hong Sub Choi\*

Division of Electrical and Electronic Engineering, Daejin University, Pocheon 11159, Korea

### [요 약]

본 연구는 블랙박스 영상을 이용하여 도주 중인 범죄차량의 검거나 원하는 중요한 영상을 확보하는 것을 목적으로 하는 실시간 차량 위치추적 시스템을 구현한다. 차량에 장착된 블랙박스의 카메라를 통해 5초 간격으로 전방 차량의 영상을 찍은 다음 영상 내부의 번호판을 인식한다. 인식된 차량번호는 문자열의 형태로 운전자의 스마트폰으로 전송되며 스마트폰의 현재 GPS 값과 합쳐 서버로 전송한다. 그리고 서버는 전송받은 데이터들을 데이터베이스 관리 시스템을 통해 처리하여 중복을 방지하고 저장한다. 위치 추적 프로그램은 사용자가 원하는 차량의 위치 정보를 서버에게 요청하여 수신하고 그래픽유저 인터페이스에 표시한다. 사용자는 이 프로그램으로 특정 차량의 일정 시간동안 이동 경로 또는 특정 위치를 지나간 차량들의 정보를 얻을 수 있다.

### [Abstract]

In this research, a real-time vehicle location tracking system is implemented that aims to arrest criminal vehicles running away or obtain important informations using black box images. The camera of the black box mounted on the vehicle captures images of the front vehicle at intervals of 5 seconds and then recognizes the license plate. The recognized vehicle number is sent to the driver's smart-phone in the form of a string and is transmitted to the server with the current GPS value of the smart-phone. The server then processes the transferred data through the database management system to prevent duplication and store it. The location tracking program requests and receives the location information of the vehicle user wants from the server and displays it with the graphic user interface. This program allows the user to obtain information about vehicles passing through a particular location or on a moving path for a certain vehicle during some period of time.

**색인어** : 블랙박스, 번호판 인식, GPS, 웹 서버, 실시간 차량 위치추적 시스템

**Key word** : Black box, License plate recognition, GPS, Web server, Real-time vehicle location tracking system

<http://dx.doi.org/10.9728/dcs.2019.20.6.1087>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 14 May 2019; Revised 20 June 2019

Accepted 25 June 2019

\*Corresponding Author; Hong Sub Choi

Tel: +82-31-539-1903

E-mail: hschoi@daejin.ac.kr

# I. 서 론

## 1-1 연구 배경

도로를 지나가다 보면 사고현장의 목격자를 찾는 현수막이 걸려 있는 것을 볼 수 있다. 이는 사고가 났을 때 사건 현장의 CCTV나 블랙박스 영상 등을 확인할 수 없거나 확보하지 못했을 때의 경우이다. 본 연구는 이런 경우에 유용하게 사용할 수 있는 실시간 차량위치추적 시스템 구현을 목표로 진행하게 되었다.

이는 도주차량이 핸드폰의 전원을 끄면 추적을 할 수 없었던 기존의 방식과는 차별화된 간접적인 실시간 위치 추적 시스템으로 기존의 한계를 뛰어넘고 과거의 범죄 차량의 경로와 겹치는 다른 차량의 번호를 확인함으로써 블랙박스 영상 취득에 도움을 주는 등 범피수사의 시간을 단축하고 정확도를 높이는 기대효과가 있다.

## 1-2 연구 개요

본 연구에서 구현하고자 하는 전체 시스템의 블록도는 그림 1과 같다. 시스템의 핵심은 차량에 장착된 블랙박스 내부 프로세서를 이용하여 번호판 인식을 위한 영상처리를 진행하는 것이다. 이는 처리된 데이터를 문자열 형태로 서버로 전송하므로 데이터베이스에 저장되는 데이터의 총량을 획기적으로 줄일 수 있다. 모든 블랙박스의 영상을 전송하는 것은 통신에 엄청난 부하를 요구하기 때문에 블랙박스 내부에서 영상처리 프로세싱을 담당하도록 하였다. 그러므로 블랙박스의 기능을 하면서도 영상처리도 할 수 있는 고성능 프로세서인 오드로이드 보드를 사용하였다.

영상에서 차량 번호를 얻어내기 위해서는 전체 영상에서 번호판의 위치를 특정지어서 번호판의 영역만을 알아내어 추출하는 과정이 필요한데, 번호판의 형태학적인 특징을 수학적으로 접근하여 프로그래밍으로 다른 물체와 구분할 수 있는 알고리즘을 사용하였다. 번호를 추출한 이후에는 오드로이드 보드에서 연산한 정보를 근거리 통신 기술을 이용하여 안드로이드 어플리케이션으로 지속적으로 전달할 수 있는 통신이 필요하다.

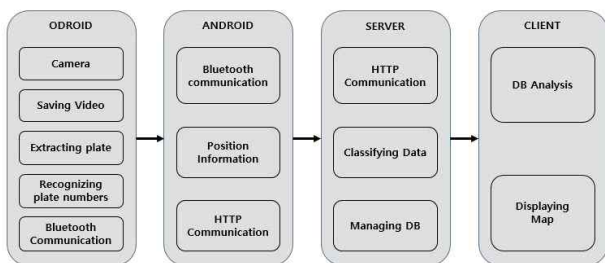


그림 1. 전체 시스템 블록도  
Fig. 1. Block diagram for the entire system



그림 2. 오드로이드 보드  
Fig. 2. Odroid board

안드로이드 통신을 이용한 것은 블랙박스에서 직접적인 서버로의 통신을 위해서는 통신사를 통한 장거리 통신 기술이 필요한데 안드로이드 어플리케이션을 거친다면 장거리 통신 소자를 사용하지 않고도 스마트폰의 기능을 이용하여 통신이 가능하기 때문이다. 또한 GPS소자 역시 스마트폰 내부에 장착되어 있기 때문에 여기서 GPS데이터를 받아오는 장점도 가질 수 있다. 본 연구에서는 이러한 근거리 통신 기술로 차량 내 스마트폰 연동으로 이미 상용화 되어 있는 블루투스 통신 방법을 채택하였다. 전송된 정보는 안드로이드 어플리케이션에서 처리한다. 어플리케이션 개발과정에서는 안드로이드 스튜디오를 사용하였으며 스마트폰의 GPS 위치 데이터 수신 기능을 이용하여 기존의 문자열 데이터에 추가로 위치 데이터를 합쳐 서버 내 데이터베이스에 저장하게 된다.

서버는 JSP 언어를 이용하여 아파치 Tomcat을 사용하여 서버의 역할을 하였고 데이터베이스에는 MySQL을 사용하였다. 서버 동작은 HTML5에서 이루어지며 구글맵 API와 연동하여 차량 위치를 지도에서 확인할 수 있게 하였다.

# II. 본 론

## 2-1 시스템 구현 방법

시스템 구현에 대한 자세한 내용은 다음과 같다.

### 1) SBC(Single Board Computer)

먼저 영상데이터를 취득하고 영상처리와 통신을 담당하는 SBC는 그림 3과 같이 구성한다. 오드로이드 보드는 Hard kernel사의 ODROID-XU4 모델에 OS로는 ubuntu 16.04 mate를 설치하여 이용하였다.

ODROID-XU4에 부가적으로 연결되는 장치로는 영상정보를 만드는 USB 카메라인 oCam과 저장공간으로 사용할 Micro SD 카드, 이용자의 스마트폰으로 데이터를 전송하기 위한 Bluetooth Dongle이 있다. oCam을 이용하여 영상정보를 받아오고, C++ 기반의 프로그램에서 영상정보를 저장하고, 번호판을 추출하여 연결된 Bluetooth Dongle을 통해 스마트폰으로 전송하게 된다.

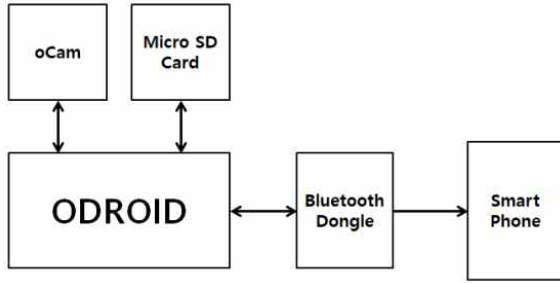


그림 3. SBC(Single Board Computer) 블록도  
Fig. 3. Block diagram of SBC

2) 번호판 인식

(1) 전체 알고리즘

카메라로 촬영된 영상을 C++를 기반으로 한 OPEN CV를 이용해 영상처리 기술을 적용하여 번호판의 문자열을 추출한다. 전체 번호판 인식 알고리즘은 크게 번호판 영역추출 부분과 문자인식 부분으로 분류된다. 그림 4의 번호판 영역 추출 알고리즘에서는 전체 영상에서 번호판이 가진 특성만을 찾아내어 번호판의 영역을 추가적인 이미지로 만드는 것을 목표로 한다.

(2) 번호판 영역추출

① 문자획 강조

1차 미분 필터의 경우는 연산자가 에지가 존재하는 영역을 지나면 민감하게 반응을 하게 된다. 따라서 이를 완화하기 위한 목적으로 2차 미분연산자인 LoG (Laplacian of Gaussian) 필터가 쓰이게 되었다. 또한, 수평, 수직, 대각선 방향에 놓인 에지에 너무 민감하게 반응하는 1차 미분 필터와는 다르게 2차 미분은 해당 부분을 둔감 시킴과 동시에 에지를 더욱 강조하는 효과를 보여준다.

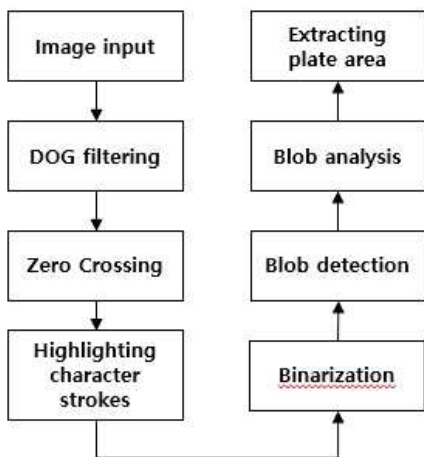


그림 4. 번호판 영역 추출 알고리즘  
Fig. 4. Number plate area extraction algorithm

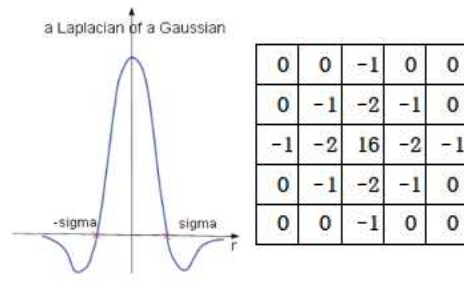


그림 5. LoG 마스크  
Fig. 5. LoG mask

$$LoG(x,y) = -\frac{1}{\pi\sigma^4} \left[ 1 - \frac{x^2+y^2}{2\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

LoG는 식 (1)에 따라서 그림 5와 같은 계수들의 합이 0이 되는 임의의 크기의 마스크를 만들 수 있다. 만들어진 마스크와 영상을 컨볼루션을 하면 영상은 2차 미분 값을 가지게 되고 값 중에 영점을 교차하는 점을 찾아내어 에지의 위치로 결정한다. LoG는 1차 미분 필터와는 다르게 에지 추출에 대해서는 더 큰 장점이 있지만 연산을 진행할 때 많은 자원이 필요하다는 단점이 있다.

이 점을 개선하기 위해 Marr와 Hildreth은 LoG 필터를 연산 속도가 빠른 DoG(Difference of Gaussian)로 근사화 시키는 것이 가능함을 알아내었다.[1]

$$DoG(x,y) = \frac{e^{-\frac{(x^2+y^2)}{2\sigma_1^2}}}{2\pi\sigma_1^2} - \frac{e^{-\frac{(x^2+y^2)}{2\sigma_2^2}}}{2\pi\sigma_2^2} \quad (2)$$

식 (2)에서  $\sigma_1 > \sigma_2$ 이다. 여기서 DoG가 LoG와 근사한 에지를 가지기 위해서는 적당한  $\sigma$  값의 조절이 필요하다. DoG에서 보통 두 개의  $\sigma$  값이 1:1.75나 1:1.6의 비를 가지면 LoG와 근사한 값을 낸다. 위 식의 각 항은 영상을 Gaussian blur한 것의 미하며 결과적으로 DoG는  $\sigma_1, \sigma_2$  값에 따른 두 개의 Gaussian blur 영상의 차이 값이 된다. 이렇게 2차 미분 값을 가지게 되면 밝기에 따라 이미지의 배열 값이 0에서 교차(Zero Crossing)하는 현상이 일어나게 된다.



그림 6. 가우시안 블러 (a) 원본 (b)  $\sigma = 1.75$  (c)  $\sigma = 1.0$   
Fig. 6. Gaussian blur (a) original (b)  $\sigma = 1.75$  (c)  $\sigma = 1.0$

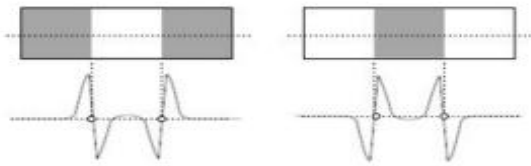


그림 7. Zero Crossing 과정  
Fig. 7. Process of Zero Crossing

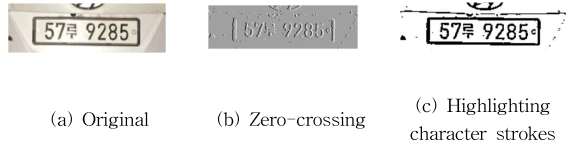


그림 8. 문자 획 강조 과정  
Fig. 8. Process of highlighting character strokes



그림 9. 번호판의 숫자열  
Fig. 9. a numerical array on license plates

이런 현상을 이용하여 하얀색 번호판과 같은 경우 밝은 부분에서 문자 획을 지나가며 어두운 부분을 가지게 되는데, 이 때 DoG 필터를 거치면서 그림 7과 같이 문자 획이 시작하는 부분은 음수에서 양수로 0을 교차하고 문자 획이 끝나는 부분은 양수에서 음수로 0을 교차하게 된다. 바로 이 교차점을 이용하여 그림 8과 같이 문자 획을 강조한 이미지를 얻을 수 있다.

문자 획이 강조된 문자들 중 모든 번호판 형식에 포함된 마지막의 숫자열 4개(그림 9)는 번호판 영역을 추출하는 중심이 된다.[2] 문자 획이 강조된 이미지는 블롭 알고리즘에 의해 이미지 내의 연결 화소들이 구분이 되게 된다.

② 블롭 감지 알고리즘

블롭 감지 알고리즘이란 이진화된 영상의 연결화소끼리 같은 번호를 부여하여 각각의 연결화소를 효과적으로 이용하기 위한 방법이다. 블롭 감지 알고리즘에는 여러 종류의 마스크들이 사용된다.

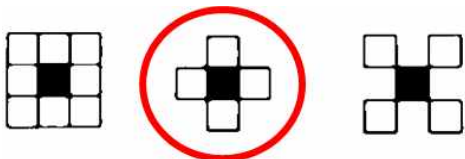


그림 10. 마스크  
Fig. 10. mask

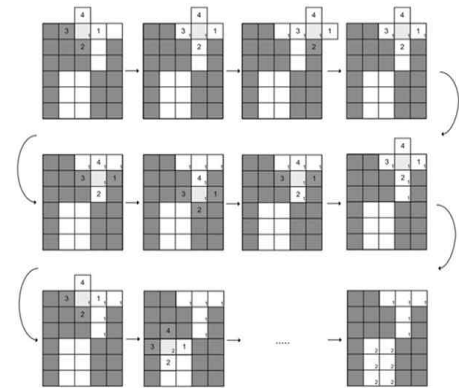


그림 11. 연결 화소 감지  
Fig. 11. Connection pixel detection

그림 10은 연결되어 있는 화소의 범위를 정해주는 마스크이다. 연결화소를 검색할 때에는 기준화소를 기점으로 주변에 어떤 범위까지 연결화소로 인정하는 지에 대한 기준을 정한다. 본 연구에서는 중앙의 십자 마스크를 선택하였다.

마스크가 정해졌다면 픽셀단위로 그림 11과 같이 왼쪽 위부터 오른쪽으로 가면서 한 줄씩 마스크를 씌워 주변의 연결 화소가 있는지 감지한다. 행렬형식의 이차원 배열 형태로 이미지를 인식하여 반복문을 이용해 제어하는 화소를 정해준다. 이동 중 검은 화소가 감지되었을 경우 번호를 매기는데 왼쪽 위를 기준으로 1부터 번호를 매기게 되고 다음 연결 화소가 나타날 때 다음 번호를 매긴다. 마스크 내에 연결화소가 있게 되면 해당하는 연결화소의 블롭 번호를 받아와 같은 번호를 부여 받게 된다. 이때에 여러 가지 변수들이 나타나게 되는데 다른 블롭으로 인식 되었다가 추후의 마스크를 씌운 화소에서 같은 블롭인 것이 밝혀지게 되면 이전의 배열 값으로 돌아가 작업을 다시 시작하게 된다.

완성된 블롭은 그림 12와 같은 형태가 된다. 같이 붙어있는 검은 화소끼리는 동일 번호를 매긴 것을 확인할 수 있다.

이렇게 번호를 부여받은 블롭들은 각각 분석되어 고유한 숫자 값을 가지게 된다. 우선 블롭이 정해지면 블롭의 Height, Width, Top, Bottom, Left, Right 등의 속성 값을 가지게 된다.

번호판에서 가장 특징적이고 번호판의 종류에 영향을 받지

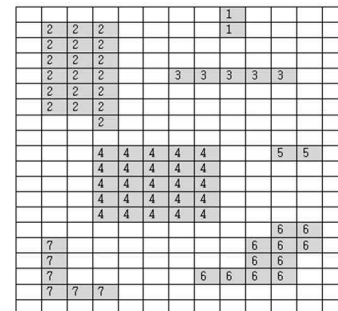
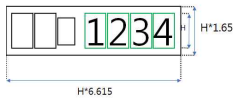


그림 12. 블롭 번호 부여  
Fig. 12. Blob numbering

$$\begin{aligned}
 &W_{std} / H_{std} < 0.9 \\
 &0.97 < T_{cmp} / T_{std} < 1.03 \\
 &0.9 < B_{cmp} / B_{std} < 1.1 \\
 &H_{std} * 0.02 < Gap < H_{std} * 0.04
 \end{aligned}$$



(a) Rule for plate number (b) Rule for plate area

그림 13. 블롭 분석 방법

Fig. 13. Blob analysis method

얇은 특징은 연속된 4개의 숫자가 존재한다는 점이다. 오래된 번호판이나 특정 목적차량의 번호판의 경우 앞의 숫자가 위로 올라가거나 지역번호가 적힌 경우가 존재하기도 하지만 연속된 숫자가 4개 이어져 사용된다는 것은 모든 번호판이 동일하다. 검색 방법은 한 개의 블롭을 기준으로 하여 그림 13-(a)와 같은 4가지 조건을 만족하게 되면 그 블롭들의 모임을 번호판으로 인식하고 실패하면 다음 블롭으로 기준 블롭을 옮겨서 검색을 새로 시작하게 된다. 이러한 방식으로 4개의 숫자를 찾게 되면 그림 13-(b)와 같이 숫자영역을 기준으로 비율을 정하여 번호판의 영역을 자르게 되는데 이때, 기준이 되는 숫자영역 블롭의 Height 속성 값을 기준으로 잡아 여러 사이즈의 번호판에도 비율적으로 접근해 원하는 번호판의 크기를 얻을 수 있었다.

3) 문자 인식

문자인식 알고리즘은 그림 14와 같이 요약된다.

(1) 블롭 감지

번호판 영역이 추출되면 다시 이진화와 블롭을 분석하는 과정을 거쳐 문자들의 블롭들을 추출한다. 이는 새로 잘라낸 번호판의 영역을 새로운 영상정보로 받아들이고 사용하기 위함인데 이때 위의 블롭 추출까지의 알고리즘을 반복한다.

(2) 문자 정규화

이후 추출된 문자영역에 대한 블롭들은 그림 15-(b)과 같이 정규화 과정을 거친다. 정규화란 숫자영역을 잘라내어 원하는 규격의 영상으로 잘라 사용할 수 있도록 어떠한 영상이라도 같은 크기로 재구성 될 수 있도록 하는 기술이며 뒤의 템플릿 매칭 방식을 사용하기 위해 필요한 작업이다.

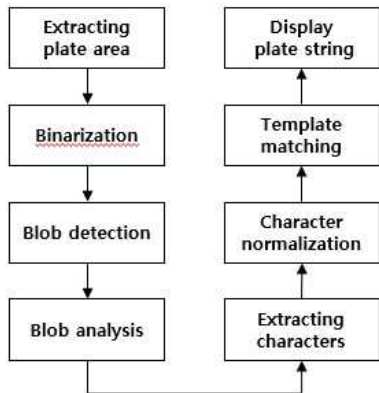


그림 14. 문자 인식 알고리즘

Fig. 14. Character recognition algorithm



(a) Extracted plate area (b) Blob normalization

그림 15. 블롭 정규화 과정

Fig. 15. Process of Blob normalization

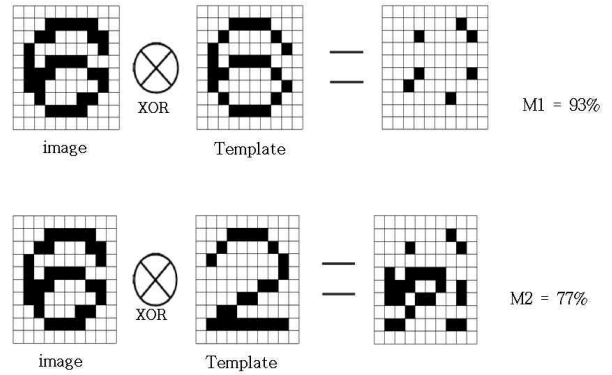


그림 16. 템플릿 매칭

Fig. 16. Template Matching

(3) 템플릿 매칭

템플릿 매칭은 템플릿을 기반으로 받은 정규화된 영상의 데이터를 비교한다. 이 비교하는 과정은 모든 화소를 비교하는 방식이며 그렇기 때문에 비교하는 두 영상의 크기가 같을 필요성이 있으며 정규화가 필요한 것이다.

데이터를 비교하면 그림 16과 같이 템플릿과 XOR 연산을 실행하고 그 결과의 흰색 영역(비교자와 피비교자가 같았던 영역)이 전체에서 어느 정도 비율을 차지하는지 계산한다.

4) 안드로이드 연동 블루투스 통신 및 서버 통신

(1) 어플리케이션 개요

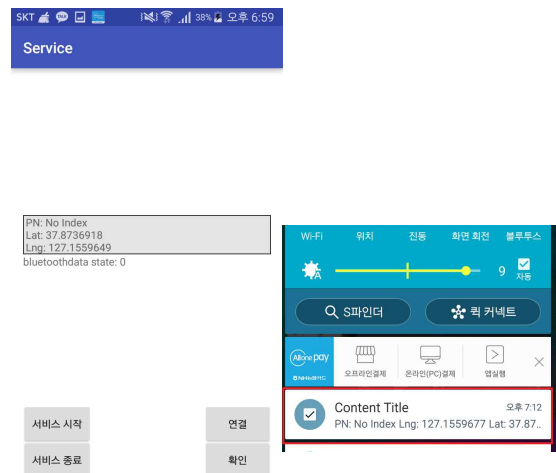


그림 17. 어플리케이션

Fig. 17. Application

본 시스템에서 사용하는 어플리케이션은 Android 6.0.1과 API 23 환경에서 제작하였다. 그림 17은 어플리케이션의 실행 화면으로 기본 화면과 백그라운드 실행 화면을 보여준다. 이 어플리케이션은 스마트폰에서 블루투스 통신을 이용하여 싱글보드 컴퓨터 오드로이드에서 번호판 문자열을 전송받고 위치정보를 취합하여 HTTP 통신으로 서버에 전송하는 역할을 한다. 서버로 데이터를 전송하는 기능은 Service 기능에서 동작한다. Service에 의해 Activity가 종료되더라도 백그라운드에서 프로그램이 지속적으로 실행되며 3초마다 취합된 데이터를 알람으로 알려주고 데이터가 전송이 되면 자동으로 서버로 데이터를 전송한다.

(2) 블루투스 연동

그림 18은 블루투스 연동을 위한 순서도를 설명한다. 블루투스 연결을 하기 전에 가장 먼저 블루투스를 지원하는지 확인한다. 해당 장치가 블루투스를 지원할 경우 블루투스의 On/Off 상태를 판단하고 켜져 있지 않다면 블루투스를 켜는다. 그리고 장치 검색을 실행한다. 장치를 검색하고 페어링이 되어있는지 확인하고 페어링이 되어있지 않다면 페어링을 하고 연결을 시작한다.[3]

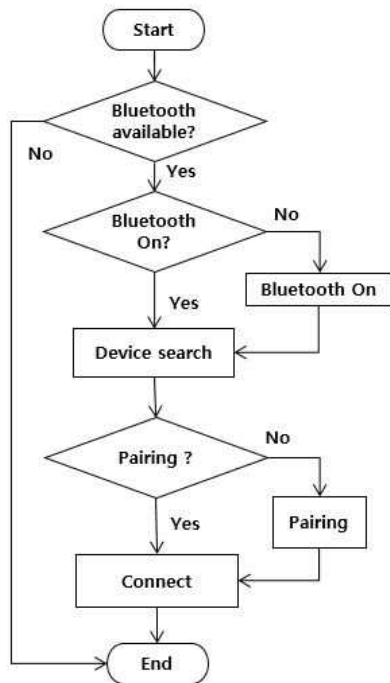


그림 18. 블루투스 연동 순서도  
Fig. 18. Flow chart of Bluetooth Interlink

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

그림 19. 블루투스 권한  
Fig. 19. Bluetooth privileges

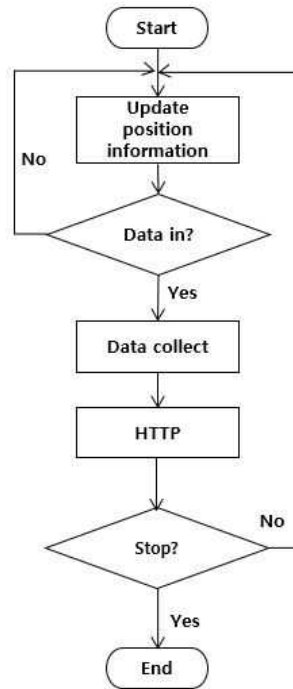


그림 20. 서비스 순서도  
Fig. 20. Flow chart of service

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
```

그림 21. 네트워크 권한  
Fig. 21. Permission of network

그림 19는 블루투스를 사용하기 위해 선언한 권한을 나타낸다. 블루투스를 사용하기 위해서는 Manifests에서 별도의 권한 선언이 필요하다. 가장 먼저 BLUETOOTH 권한은 커넥션 요구 수락, 데이터 전송 등의 블루투스 통신을 하는 데 필요한 권한이다. 그리고 BLUETOOTH\_ADMIN 권한은 장치를 검색하거나 블루투스 설정을 조작하는 데 필요하다. 단 이 권한을 사용하기 위해서는 BLUETOOTH 권한이 반드시 선언되어있어야 한다.

(3) 안드로이드 서비스

그림 21은 네트워크를 사용하는데 필요한 권한을 설명한다. ACCESS\_NETWORK\_STATE는 네트워크의 상태를 확인하기 위한 권한이고 INTERNET는 인터넷 사용을 위한 권한이다.[4] 서비스는 Activity처럼 사용자와 상호작용하는 Component가 아니고 Background에서 동작하는 Component이다. 이 기능은 Activity가 종료되더라도 지속적으로 프로그램을 실행시키기 위해 사용하였다.[5]

버튼 '서비스 시작'이 눌러지게 되면 Service로 실행되는 프로그램의 순서도인 그림 20과 같은 과정을 실행한다. 가장 먼저 스마트폰이 제공하는 위치정보제공자 중 GPS를 이용하여 위치정보(위도, 경도)를 얻는다.

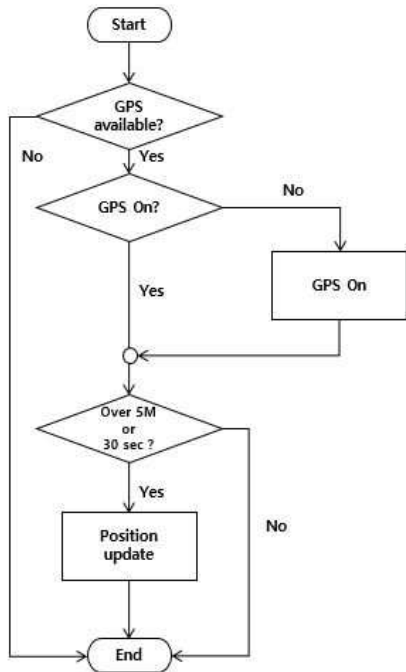


그림 22. GPS 서비스 순서도  
Fig. 22. Flow chart of GPS service

```

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    
```

그림 23. 네트워크 권한  
Fig. 23. Permission of network

위치정보는 실시간으로 갱신되며 데이터가 수신될 경우 위치정보와 전달받은 번호판 문자열을 취합하여 HTTP 통신을 통해 서버에 데이터를 전송한다. 버튼 ‘서비스 종료’가 눌리면 Service 기능을 종료한다.

(4) 안드로이드 GPS 연동

그림 23은 위치정보제공자를 사용하는데 필요한 권한을 나타내는데, 위치정보제공자를 이용하기 위해서는 위와 같은 권한을 설정해줄 필요가 있다.

ACCESS\_COARSE\_LOCATION은 WIFI를 이용한 위치정보를 얻을 때 사용하는 권한이고 ACCESS\_FINE\_LOCATION은 GPS를 이용하여 위치정보를 확인하기 위한 권한이다.

그림 22는 위치를 갱신하는 순서도이다. 가장 먼저 GPS를 지원하는지 확인한 후, GPS 수신기 on/off 상태를 확인하고, 꺼져 있으면 켜준다. 최초로 위치를 갱신할 때는 조건 없이 위치를 갱신하지만, 그 다음부터는 기존의 위치가 현재의 위치와 최소 5M 또는 30초의 차이가 나면 위치를 갱신한다.[6]

5) 차량번호 데이터베이스 및 웹 서버

(1) 서버 구축

본 연구에서는 각 블랙박스에서 수집한 정보를 저장하고 운용하기 위해서 웹 서버를 구축하여 사용하였다.

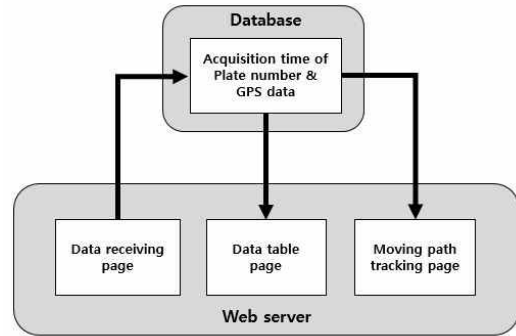


그림 24. 웹 서버 블록도  
Fig. 24. Block diagram of the web server

각 모듈에서 전송되어진 정보는 안드로이드 통신을 통해 서버로 전송이 되고 서버에서는 그 정보를 데이터베이스에 저장하여 이후 위치추적 시스템에 사용하였다. 서버는 Apache Tomcat8.0 웹 어플리케이션 서버를 사용 하였다.[7] 이는 JSP언어를 기반으로 페이지를 만들어 전송받은 정보를 데이터베이스에 저장하는 방식의 서버를 만들었다. 데이터베이스는 MySQL 데이터베이스를 사용하였다. 그림 24는 웹 서버의 페이지를 나타낸 블록도 이다. 서버 페이지는 크게 안드로이드의 정보를 수신하는 데이터 수신 페이지, 데이터베이스의 정보를 받아오는 데이터 테이블 페이지, 원하는 차량의 이동경로를 추적하는 페이지로 나뉜다.

(2) 데이터 수신 페이지

데이터 수신 페이지에서는 안드로이드 어플리케이션에서 전송받은 정보를 데이터베이스로 넘기는 역할을 한다. HTTP 통신을 통하여 안드로이드에서 넘겨받은 차량번호, 위도, 경도, 현재 시간 값의 파라미터를 메모리로 전송받아 데이터베이스에 저장한다. 안드로이드 어플리케이션에서 정보를 전송할 때마다 반복하여 실행되며 데이터의 저장 성공 여부를 표기한다.

(3) 데이터 테이블 페이지

데이터 테이블 페이지에서는 데이터베이스의 테이블에 저장된 차량의 번호, GPS 데이터, 정보를 전송받은 시간을 수신하여 웹 페이지에 표의 형태로 받은 정보를 표시하도록 하였다. 이는 가장 최근의 데이터를 순서대로 보여주는 역할을 한다.

(4) 이동경로 추적 페이지

이동경로 추적 페이지에서는 추적을 원하는 차량의 번호를 입력하면 해당하는 차량의 정보를 데이터베이스에서 검색, 지도상에 매핑하는 작업을 하게 된다. 지도는 구글맵 API를 연동하여 사용하였으며 검색한 차량의 데이터 정보를 JSP코드에서 자바스크립트 코드로 넘겨주어 처리한다. 넘겨받은 데이터는 자바스크립트 상에서 연동한 구글맵 API를 이용하여 GPS데이터를 지도상에서 운용되게 된다. 구글맵의 마커 시스템을 이용

하여 검색한 차량의 번호를 시간대별로 정렬하고 이후 순서에 맞추어 맵에 표시를 하도록 하였으며, 이후 구글맵의 항공경로 표시 API를 사용하여 이동경로를 지도상에 매핑하여 시각적으로 두드러지게 확인할 수 있도록 하였다.

### III. 실험 및 평가

#### 3.1 번호판 인식

그림 25는 원본 영상으로부터 번호판을 추출하는 과정을 보여준다. Zero Crossing을 이용하여 누적으로 획을 생성하여 숫자 획이 강조되고 블롭을 성공적으로 감지된 것을 확인할 수 있었다.

그림 26과 같이 블롭을 분석한 결과 4개의 숫자열을 찾아내었고 그것을 기반으로 번호판 영역을 추출하였다. 추출된 번호판은 이진화 되어 다시 블롭 분석을 통해 문자들만 분리해내어 정규화 과정을 거쳤다. 정규화 된 문자들은 템플릿 매칭을 통하여 성공적으로 문자 인식이 되었음을 확인했다.

본 연구에서는 150개의 자동차 번호판을 인식했으며, 이 150개의 자동차 번호판에 대한 평균 인식시간과 인식률은 표 1과 같다.

#### 3.2 위치추적 확인

번호판 인식 과정을 거친 번호판 정보가 안드로이드 어플리케이션을 통하여 해당 시스템이 부착된 차량의 위치 정보를 서버로 전송하는 것을 확인하였다. 이후 구글맵 API를 이용해 번호판 정보와 위치 정보를 지도상에 매핑하고, 해당 번호판 차량의 주행 경로를 추적할 수 있는 것을 확인할 수 있었다.

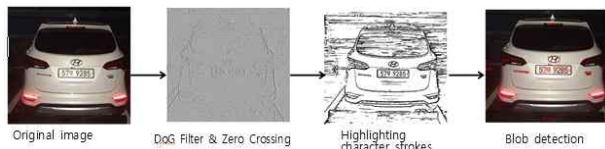


그림 25. 번호판 인식 과정 1  
Fig. 25. Process of license plate recognition 1

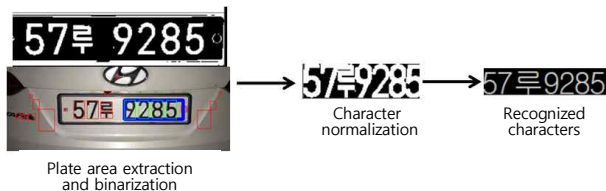


그림 26. 번호판 인식 과정 2  
Fig. 26. Process of license plate recognition 2

표 1. 블롭 알고리즘의 평균 인식시간과 인식률

Table 1. Average perception time and perception rate of Blob Algorithm

Algorithms	Recognition time/s	Recognition success rate
Blob Algorithms	0.06	95%

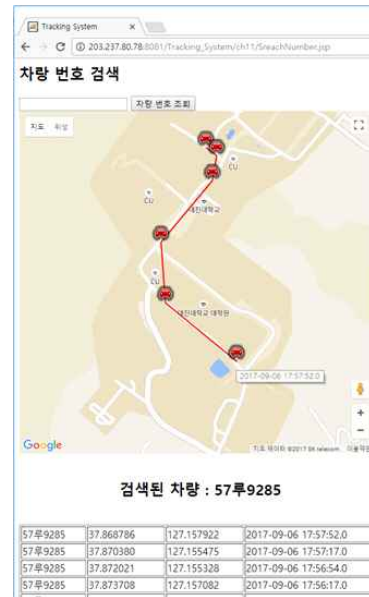


그림 27. 번호판 정보 및 위치  
Fig. 27. License plate information and location

#### 3.3 문제점

영상처리를 기반으로 추적하기 때문에 비가 오거나 번호판이 손상되었을 때는 추적이 불가능하다. 또한, 차량의 번호로는 조회할 수 있으나 차량추적에서 큰 도움이 되는 차량의 종류를 저장하는 시스템의 구현은 되어있지 않은 상황이다.

### IV. 결론 및 추후과제

기존에 제안된 차량위치추적 시스템은 모바일 기기인 PDA를 이용한 것으로, 블랙박스의 영상처리 기능이 없고, 별도의 모바일기기가 필요하기 때문에 시스템 활성화에 한계가 있었으나, 본 논문에서는 많은 사람이 사용하는 스마트폰의 기능을 사용함으로써 시스템 구축이 매우 용이하다는 장점이 있다.[8]

본 연구에서는 안드로이드 보드를 이용한 번호판 인식부터 안드로이드 어플리케이션을 이용한 통신, 웹 서버에 연동된 데이터베이스에 정보를 저장하고 차량 번호를 조회하는 시스템을 구현하였다. 이 시스템을 다수의 블랙박스에 적용하게 되면 큰 고속도로나 국도의 거의 대부분 차량의 위치를 확인할 수 있게 된다. 이로 인해 수배된 차량에 대해 즉각적인 추적이 가능



해지며 관제센터와 협력하여 실시간으로 도주차량을 검거하는 활동이 가능하다. 이 시스템에서는 기존의 경찰수사에서 소모되는 인력과 시간을 효과적으로 줄일 수 있으며 현재의 수동 시스템인 모든 CCTV를 확인하는 소모가 큰 작업에서 또한 작업 공간을 특정 지을 수 있을 것으로 보이며 수사에 소모되는 시간이 수십 분의 일로 줄어드는 효과를 볼 수 있을 것이다.

또한, 이 경우에는 차량의 사용자를 조회하는 방식이 아닌 차량의 위치 자체만을 추적하기 때문에 대포차 등 범죄에 이용되는 차량을 추적할 때에도 아무런 문제없이 사용할 수 있다.

그러나 위와 같은 시스템을 구축하여 사용할 경우, 임의의 사용자가 블랙박스 연산, 스마트폰의 데이터 통신 등 자신의 자원을 공공의 목적으로 활용하는 것에 대한 동의가 필요할 것이다.

현재 개발되어있는 프로토타입에는 다음과 같이 개선할 사항이 있다. 우선 실험에 사용한 서버가 개인 서버로 전용 서버 컴퓨터가 아니어서 서버의 크기가 협소하다. 시스템의 특성상 커다란 서버를 필요로 하므로 용량이 큰 서버를 사용해야 한다. 또한, 개발과정에서 개발전용 보드인 오드رويد 보드를 사용하였으나 실제 상용화가 된다면 제작비를 절감하기 위해 적합한 영상처리 전용 프로세서를 이용해야 할 것으로 보인다. 그리고 개발과정이기 때문에 구글 API의 모든 기능을 사용하는 것이 불가능하였지만 이후 정식적인 절차를 통해 사용하기 편한 GUI를 개발할 수 있을 것으로 보인다.

## 참고문헌

- [1] Rafael C. Gonzalez, Richard E. Woods Digital Image Processing, 3ed. pp.714-725, Pearson, 2007
- [2] Jin Ho Kim, "Distortion Invariant Vehicle License Plate Extraction and Recogniton Algorithm", *The Journal of the Korea Contents Association*, Vol. 11, No.3, pp.1-8, Mar. 2011
- [3] <http://dsnighnt.tistory.com/36>
- [4] <https://developer.android.com/index.html?hl=ko>
- [5] <http://arabiannight.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9CAndroid-Service-%EC%82%AC%EC%9A%A9%EB%B2%95>
- [6] <http://bitsoul.tistory.com/131>
- [7] En Ok Kim, Introduction to JSP WEB programming, SamyangMedia, pp.32-479, 2015
- [8] Jun Seok Oh , Yoon Ae Ahn , Seung Youn Jang , Bong Gyou Lee , Keun Ho Ryu "Design of Vehicle Location Tracking System using Mobile Interface", *KIPS Transactions on Computer and Communication Systems*, Vol. 9, No. 6, pp. 1071-1082, Jun.2002



**최홍선(Hong Sun Choi)**

2018년 : 대진대학교 전자공학과 (학사)

2017년~현재:

※ 관심분야 : 내용기반 영상검색(Content based Image search), 멀티미디어(Multimedia) 등



**김도원(Do Won Kim)**

2018년 : 대진대학교 전자공학과 (학사)

2018년~현재:

※ 관심분야 : 서버(Server), 데이터베이스(Database) 등



**채한움(Han Um Chae)**

2018년 : 대진대학교 전자공학과 (학사)

2018년~현재:

※ 관심분야 : 안드로이드(Android), 네트워크(Network) 등



**이수연(Soo Yeon Lee)**

2018년 : 대진대학교 전자공학과 (학사)

2018년~현재:

※ 관심분야 : 통신(Communications), 네트워크(Network) 등

**최홍섭(Hong Sub Choi)**



1985년: 서울대학교 전자공학과 졸업(공학사)

1987년: 서울대학교 전자공학과 졸업(공학석사)

1994년: 서울대학교 전자공학과 졸업(공학박사)

1999년 : Carnegie Mellon Univ. 방문교수

2006년 : Univ. of Colorado 방문교수

1995년 ~ 현재 : 대진대학교 휴먼IT공과대학 전기전자공학부 전자공학전공 교수

※ 관심분야 : 멀티미디어 통신, 음성 및 음향 신호처리 등