An Interactive Map Streaming Service Technique for Mobile Devices

Seong Kim[#1], Byoung-Woo Oh[*2]

[#] MC D1 Department 8 Team, LG Electronics

219-24 Gasan-dong, Geumchun-gu, Seoul, 153-801, Korea

[1]sseong.kim@lge.com

[*] Department of Computer Engineering, Kumoh National Institute of Technology

61 Deahak-ro Yangho-dong, Gumi, Gyeongbuk, 730-701, Korea

[2]bwoh@kumoh.ac.kr (Corresponding Author)

***Abstract*—This paper proposes an interactive map streaming system in mobile environment. The system consists of two parts: a map streaming server and location collecting clients. The map streaming server captures map screen periodically and makes video stream to send it to a client. The client plays the streaming video and detects the location of the user to send to the server. The server side map is dynamically moved according to the location received from the client. The proposed technique can reduce the client's burden of huge map storage and map rendering using CPU.**

Keyword-map streaming, interactive streaming, user location, streaming server, GPS

I. INTRODUCTION

Streaming service of broadcast is widely used in network environment. Using this technology, it is available to transfer streaming video to another device [1]. Recently, streaming services are common in mobile network. Most services support one-way streaming from server to client. On-demand software streaming technology is using the streaming technic that implement a part of the application at client side from existed application in server. It is technology that possible to implement through network transmission from server without entire program [2, 3]. It has been researching and developing. However, it is technology that not yet widely active unlike the existing package software due to patent issues and the billing system.

The network is important to transfer streaming data. In Korea, there is a calling plan (a flat rate) that has no limit to use the amount of cellular network data in 3G and LTE communication environment. Because the cost of network is getting cheaper and the speed is getting faster, the streaming technology is expected to be used more widely. If a dedicated personal server can be provided for a client, the personalized services can be feasible [4].

In this paper, we propose an interactive map streaming service technique for client on mobile device. The server reads database to obtain the location of a client and makes streaming video to transfer to the client. A frame of the streaming video is a map which includes the location of the client. If a client displays map by itself, the client should store a large amount of map storage and have a sufficient CPU performance to rendering map. This paper proposes the map streaming technique to reduce the client's burden of storage and performance power. The client that runs on the mobile device only plays streaming video from the server and sends the current location to the server.

The rest of the paper is organized as follows. In section II, we present the overall architecture of the proposed system. Section III and IV describes the issue about implementing map streaming server and mobile client respectively. Finally, section IV presents a conclusion.

II. RELATED WORKS

There are various methods to render map using open API such as Google Map API, Naver Map API, Daum Map API, etc. The programs to render a wide range of spatial data formats can be implemented [4]. As for the programing tools, a wide range of tools such as C++, Java, Javascript, Android, etc. and libraries can be used for programing. In this paper, we use Naver Map API to render map.

Naver is one of the biggest portal service providers in Korea [5]. They provide many open APIs such as search API, map API, shorten URL API, Café API, etc. Naver Map API, one of open APIs provided by Naver, supports Javascript, Flash, Android, iOS, etc. Furthermore Naver Maps enables the movement, zoom-in and zoom-out and the movement of centroid after the generation of Naver Map API. The functions supporting above features are listed in TABLE I below.

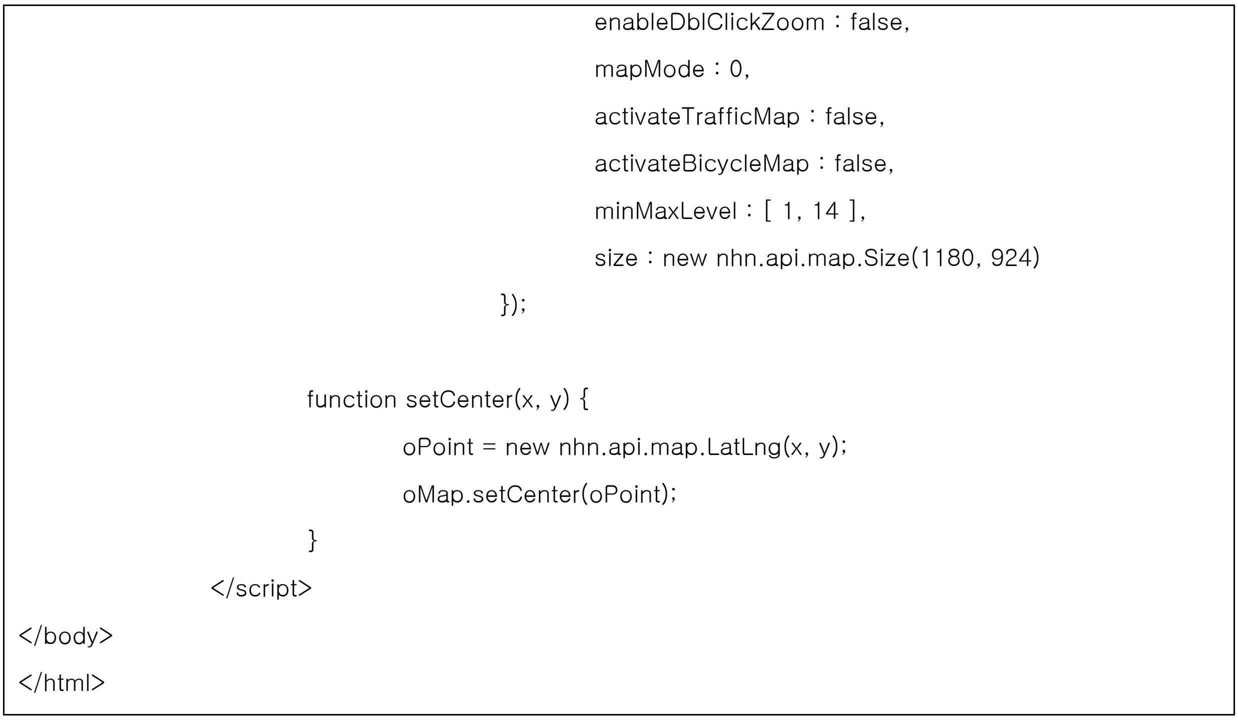
TABEL I

Important Functions of Naver Map API

Feature	Functions
Move (Panning)	Void setCenterBy(Number nX, Number nY)
Zoom-in and Zoom-out	Void setLevelBy(Number nLevel)
Move with point	Void setCenter(Coord oPoint)

The Naver Maps API can be used with Apache server Javascript 2.0. Fig. 1 shows the sample code to use Naver Map API.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
          "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="en_US" xml:lang="en_US">
<head>
                <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
                <title>Naver Map API Sample Page </title>
                <!-- prevent IE6 flickering -->
                <script type="text/javascript">
                        try {document.execCommand('BackgroundImageCache', false, true);}
                        catch(e) {}
                </script>

<script type="text/javascript" src="http://openapi.map.naver.com/openapi/naverMap.naver?
                ver=2.0&key=75352eb4e2cf78af873a08f84e104c67"></script>
</head>

<body>
<div id = "naverMap" style="border:1px solid #000; width:1280px; height:1024px; margin:20px;"></div>
                <script type="text/javascript">
                        var oPoint = new nhn.api.map.LatLng(36.1448118, 128.3932089);

                        nhn.api.map.setDefaultPoint('LatLng');

                        oMap = new nhn.api.map.Map('naverMap' ,{
                                                point : oPoint,
                                                zoom : 11,
                                                enableWheelZoom : true,
                                                enableDragPan : true,
```



```
                        enableDblClickZoom : false,
                        mapMode : 0,
                        activateTrafficMap : false,
                        activateBicycleMap : false,
                        minMaxLevel : [ 1, 14 ],
                        size : new nhn.api.map.Size(1180, 924)
                    });

            function setCenter(x, y) {
                oPoint = new nhn.api.map.LatLng(x, y);
                oMap.setCenter(oPoint);
            }
        </script>
</body>
</html>
```

Fig. 1. Map webpage sample code

III. Overall Architecture

The proposed system consists of two parts: map streaming server and mobile client. The map streaming server has four modules: map renderer, streaming encoder, location database, and location collector. The map renderer accesses the location database to read a user's position and render map surrounding the user. The streaming encoder captures the map from the map renderer and make video stream to transfer it to the client. The location database holds users' location in the database. The Location Collector is a PHP file which stores user's location into the location database. The mobile client plays the streamed video from server with the streaming decoder. The location detector obtains the position of the user with GPS and sends it to the location collector.

Fig. 2 shows the overall architecture of the proposed map streaming system.

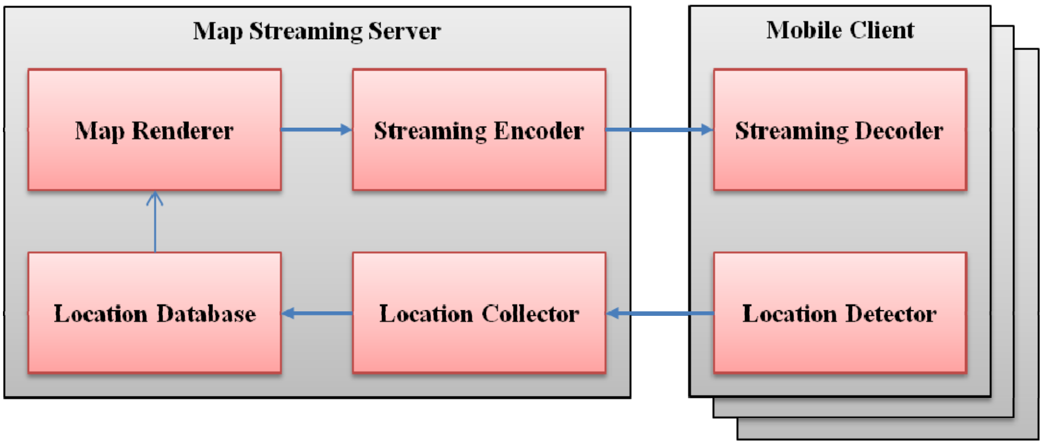


Fig. 2. Overall architecture of the map streaming system

IV. Map Streaming Server

A. Map Renderer

As mentioned in the section II, we use Naver map API to render map. The map renderer periodically reads a current location of a client from the location database and request map to the Naver map API. The map renderer is implemented with MFC in Microsoft Windows 7 operating system. Fig. 3 shows the process of the map renderer.

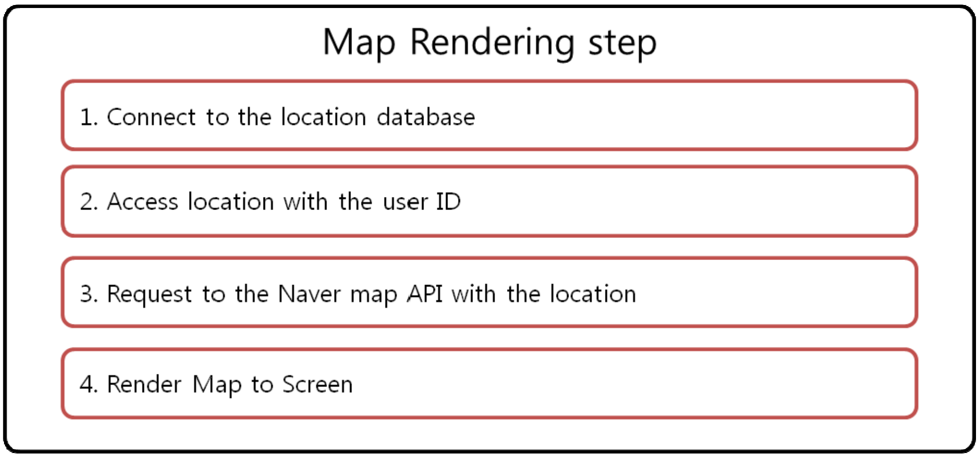

Map Rendering step
1. Connect to the location database
2. Access location with the user ID
3. Request to the Naver map API with the location
4. Render Map to Screen

Fig. 3. Map rendering step

B. *Streaming Encoder*

To stream the screen of map program, there is a need of a step to capture program screens and convert them into stream. For program screen capture and stream conversion, we use Windows Media Encoder SDK in this paper.

Windows Media Encoder SDK captures screen or receives device input from devices and streams them using MMS protocol. In addition, it performs as GUI through Windows Media Encoder SDK.

For Windows Media Encoder SDK, WMEncoder, Windows Media Encoder object, controls all functions. Fig. 4 shows the internal structure of WMEncoder class.

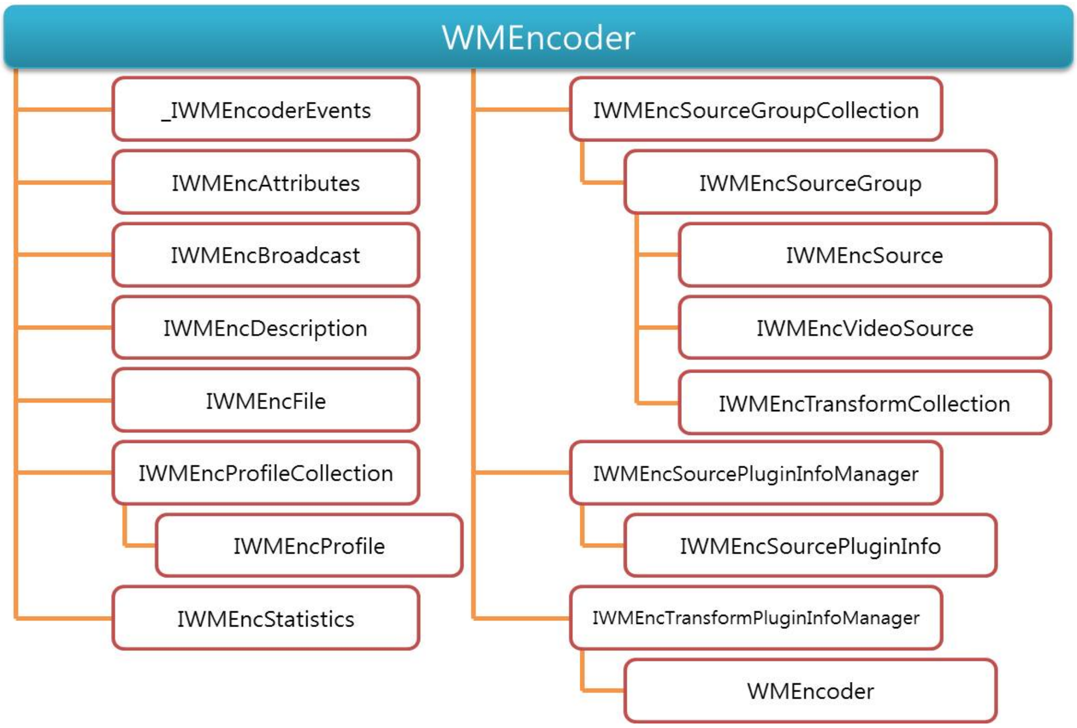
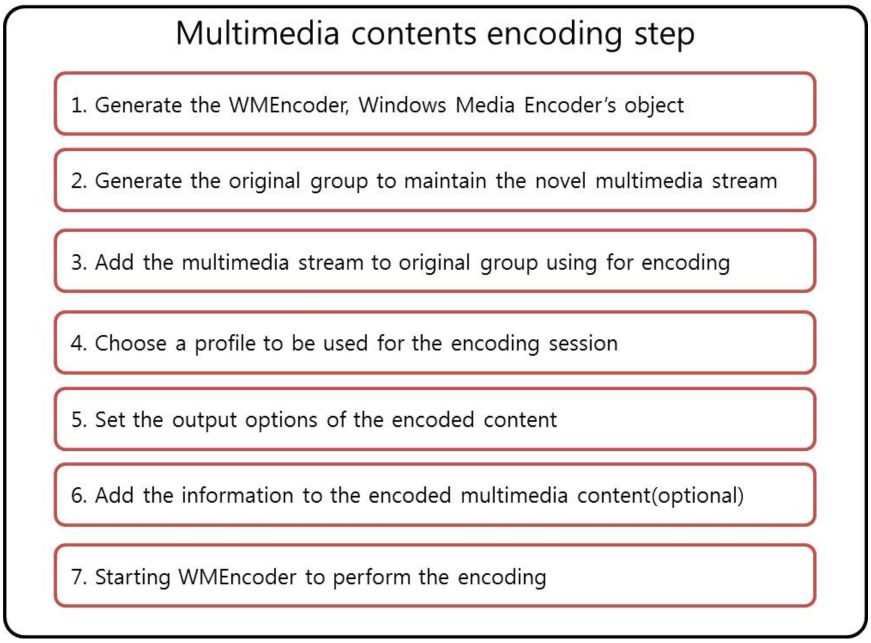


Fig. 4. Internal structure of WMEncoder Class

Fig. 5 explains the algorithm to encode multimedia contents using Windows Media Encoder SDK.



Multimedia contents encoding step

1. Generate the WMEncoder, Windows Media Encoder's object

2. Generate the original group to maintain the novel multimedia stream

3. Add the multimedia stream to original group using for encoding

4. Choose a profile to be used for the encoding session

5. Set the output options of the encoded content

6. Add the information to the encoded multimedia content(optional)

7. Starting WMEncoder to perform the encoding

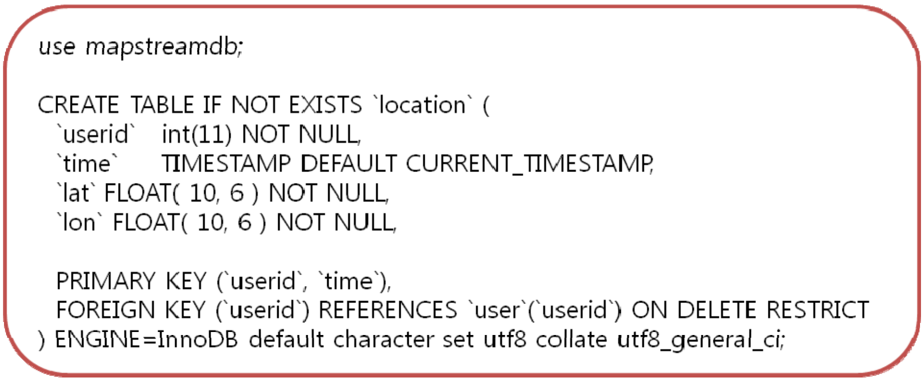
Fig. 5. Multimedia contents encoding step

As the Windows Media Encoder SDK used for screen capture in this paper uses MMS streaming protocol.

C. Location Database

To store user location, we use MySQL database. The location table has 4 attributes as shown in Fig. 6. The userid attribute references the tuple of user table. The time attribute stores the written time of location. The lat and lon attributes store the location.

The location table is written by the location collector and read by the map renderer. The location collector update table to set the lat and lon attributes using userid. The trajectory of user location is not accumulated. Only the current location is stored due to the legal issue in Korea.



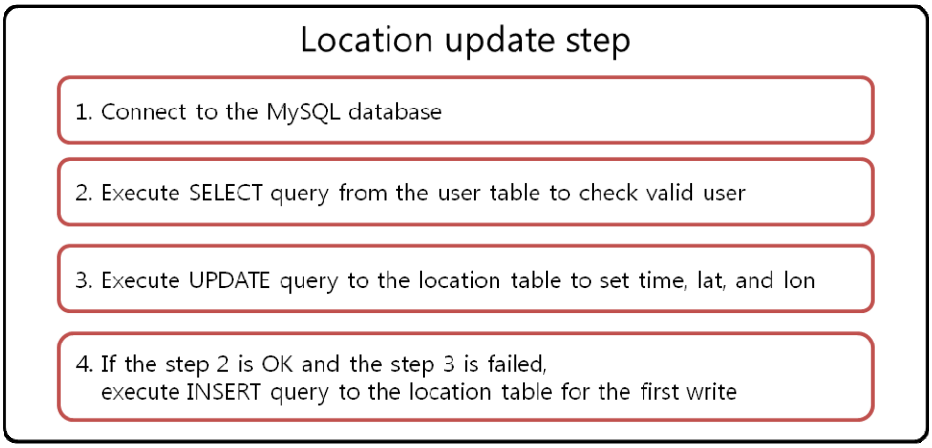
```
use mapstreamdb;

CREATE TABLE IF NOT EXISTS `location` (
  `userid`   int(11) NOT NULL,
  `time`     TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  `lat` FLOAT( 10, 6 ) NOT NULL,
  `lon` FLOAT( 10, 6 ) NOT NULL,

  PRIMARY KEY (`userid`, `time`),
  FOREIGN KEY (`userid`) REFERENCES `user`(`userid`) ON DELETE RESTRICT
) ENGINE=InnoDB default character set utf8 collate utf8_general_ci;
```

Fig. 6. Location table schema

D. Location Collector

The location collector is implemented in PHP file which is run in a web server. The location collector uses Apache web server. The location detector in the mobile client requests to the web server to invoke the location collector. HTTP POST method is used to transfer data from client to the server.

Fig. 7. shows the location update algorithm. Once the client invokes the location collector, PHP script is executed by the web server. The PHP code connect to the MySQL database with mysqli_connect() function. To check the user is valid, it executes SELECT query that read a tuple from the user table by the given userid ($_POST['userid']) and the password ($_POST['passwd']). If the result exists, the user is considered as a valid user. Then it tries to update the location. If there is no tuple in the location table, then it inserts a new tuple for the user.



Location update step

1. Connect to the MySQL database

2. Execute SELECT query from the user table to check valid user

3. Execute UPDATE query to the location table to set time, lat, and lon

4. If the step 2 is OK and the step 3 is failed,
execute INSERT query to the location table for the first write

Fig. 7. Location table schema

V. Mobile Client

This Chapter explains how to implement mobile client. To playback streaming data, it is required to implement video players using MediaPlayer class. Furthermore to obtain the current GPS values, it is required to calculate the current GPS data using Location object, LocationManager object and object's internal function. The GPS data generates HTTPClient objects for the transmission to server and then transmits that to the webpage via POST method.

A. Streaming decoder

Android smartphone, the client used in this paper, is very useful as multimedia equipment. Multimedia feature offered by smartphones is as powerful as PC to playback video. Smartphone user uses the multimedia feature as often as they use smartphones for calling, indicating the importance of multimedia features.

The core class used to utilize the multimedia features of smartphone is MediaPlayer class. Media Player class supports both audio and video and is able to playback the media of a variety of sources and formats such as the resources embedded in executable files, the media saved in storage devices as files and streams transmitted to network.

When an object is first created or reset, it starts in an Idle state and opens media calling DataSource method. It changes to Ready state calling prepare method and controls multimedia start, stop, pause, etc. by calling start, stop and pause method.

B. Location Detector

Smartphone is in general mobile equipment, User location is always consistent with that of smartphones. Smartphones have a built-in position recognition device, which enables it to calculate accurate location and obtain detailed information of surrounding places without independent equipment.

There are many ways to take advantage of location information such as navigation system, object, etc. A service using the location information in this way is called Location Based Service (LBS).

Location Provider provides location information in smartphones. In general, more than one Location Provider provides information to Android devices and Location Providers are listed in Table II below.

Table II

Android Location Type

Type	**Description**
GPS	Determines the current position using a radio wave from satellites.
Telephone Base Stations	Determines the current position through triangulation method using the unique Cell ID of telephone base stations.
Wireless Network	Determines the current position from wireless Internet AP (Access Point).

The obtained GPS data is used in server-side programs, and thus is sent to the web server periodically. HttpURLConnection class is used as a transmission method and GPS data is transmitted via POST method. Web server receives the GPS data and saves latitude and longitude data as mentioned in section IV.

Server side Programs reads GPS data saved in XML file using DOM object supported by MSXML, and updates the map. The final results of the GPS data transmission implementation of the methods proposed in this paper are shown in Fig. 8.

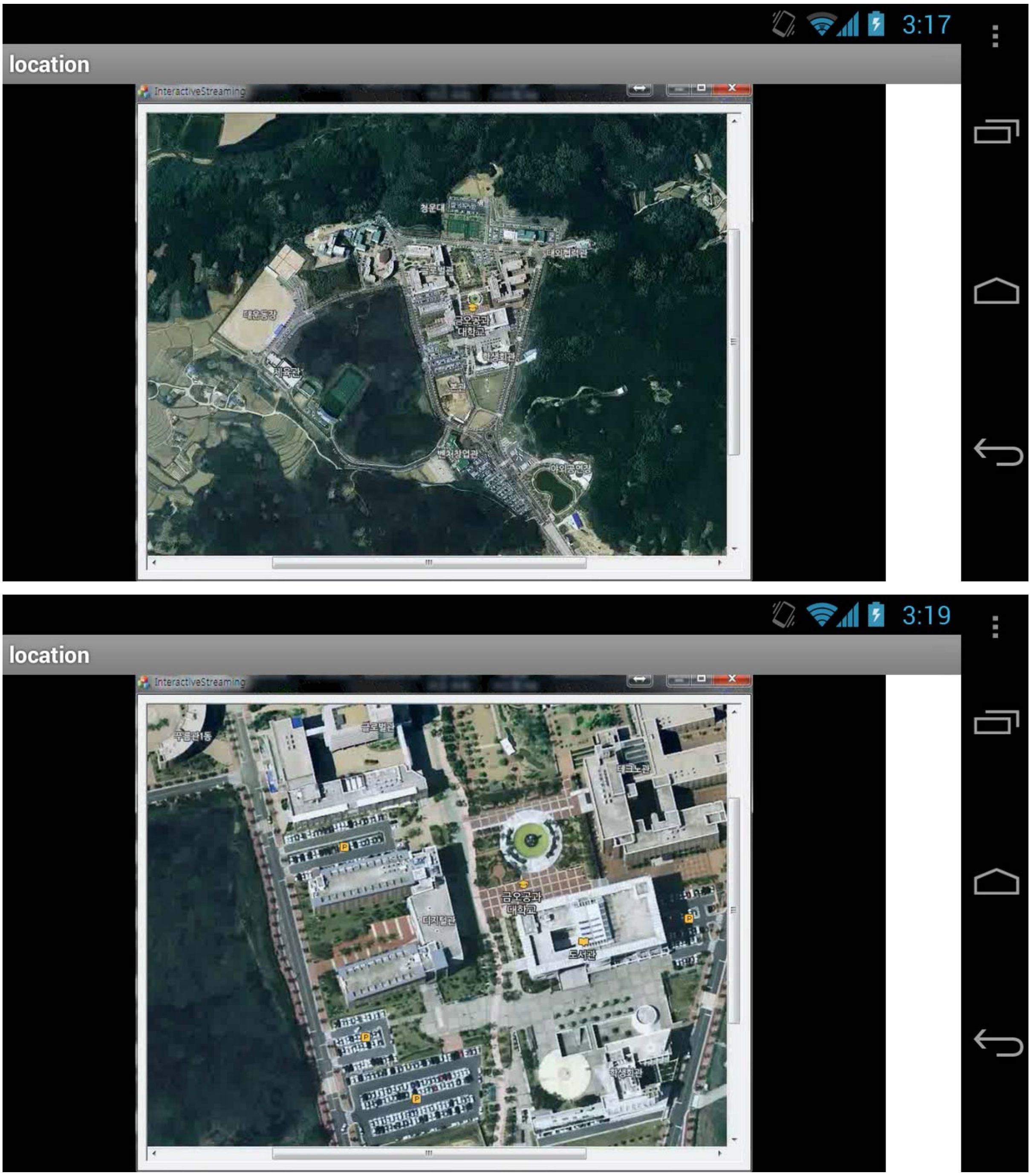


Fig. 8. Results of implementation with GPS data transmission

VI. Conclusion

With the high performance of mobile devices, smartphone usage becomes public and furthermore the development of Internet services including streaming enables smartphone users to easily watch videos and listen to music anywhere. As the market of smartphone is growing, the location based services (LBS) market is finally coming of age. Spatial data is essential to display map of the user's location. The spatial data need large amounts of storage and sufficient CPU power to process them. There are several approaches to display map in mobile device. Client may store local spatial data and render map by itself such as car navigation system. Some client only renders partial spatial data transmitted from server with user's location. Another kind of client, proposed in this paper, does not store spatial data nor render map. The proposed client only plays streamed video from server and sends current location to the server. The proposed server renders the map of the desired locations. It captures the time-variant map and encodes into streaming. The proposed method enables map service on moving mobile device only with streaming service regardless of the performance of mobile devices. About the network cost, there is a flat rate that has no limit of network data in 3G and LTE in Korea.

Various applications that utilize performance of powerful server and send result as streaming to client can be future work.

ACKNOWLEDGMENT

This paper was supported by Research Fund, Kumoh National Institute of Technology.

REFERENCES

[1] Youn IJin, *Market Trend of Live Streaming Video*, KT digieco Issue&Trend, 2011.

[2] Electronics and Telecommunications Research Institute, "Technology Status and Development Direction of On-Demand Software Streaming ," Trend analysis of Electronics and Telecommunications , Vol.19, No.5 pp. 59-70, 2004.

[3] Kim YeongMan, Heo SeongJin, Choi Wan, Park HongJae, "Design and Implementation of Software On-Demand Streaming System, " Korean Institute of Information Scientists and Engineers, Vol.32, No. 2 pp. 538-540, 2005.

[4] Seong Kim, *Personalized Service for an interactive mobile system*, master's degree thesis, Kumoh National Institute of Technology, 2012.

[5] NHN Corp, http://dev.naver.com/openapi, 2013.

[6] L. Keller, A. Le, B. Cici, H. Seferoglu, C. Fragouli, and A. Markopoulou, "MicroCast: Cooperative Video Streaming on Smartphones," in Proc. of 10th International Conference on Mobile Systems, Applications, and Services (MobiSys), Low Wood Bay, United Kingdom, June 25-29, pp. 57-70, 2012.

[7] M. Ramadan, L. E. Zein, and Z. Dawy, "Implementation and Evaluation of Cooperative Video Streaming for Mobile Devices," in Proc. of 19th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Cannes, France, September 15-18, pp. 1-5, 2008.

[8] Andik Setyono, Md. Jahangir Alam, Raed Ali, Design of Multimedia Messaging Service for Mobile Telemedicine System. International Journal of Network and Mobile Technology, pp. 11-14, 2010.

[9] H. Soroush, P. Gilbert, N. Banerjee, M. D. Corner, B. N. Levine, and L. Cox, "Spider: Improving Mobile Networking with Concurrent Wi-Fi Connections," ACM SIGCOMM Computer Communication Review, vol. 41, no. 4, pp. 402-403, August 2011.