

Implementation of a Smart VOD System Prototype

Changyong Han¹ and Jaegool Yim^{2,*}

¹*Dept. of Business and Economics, Dongguk University, Gyeongju, Korea*

²*Dept. of Computer Engineering, Dongguk University, Gyeongju, Korea
{cyhan, yim}@dongguk.ac.kr*

Abstract

Video On Demand (VOD) service that allows users to select and watch video content is the most popular IPTV service and it is known that most of Internet traffic is for VOD service. It would be very attractive if a VOD service correctly recognizes the situation where the user wants to watch a video, finds out what video the user wants to watch, and actively plays the video for the user. Our VOD system does those things and is called Smart VOD. Smart VOD is a client-server system. The client is an Android app that collects sensor values, displays a floor map, and plays a video. The main components of the server are the floor maps repository system and the VOD server. The repository system allows users to upload electronic floor maps, draws maps, retrieves the requested floor map, downloads a graphical floor map, and manages information of points of interest. The VOD manages information of users, determines user's current location, and recognizes the object that the user is watching. This paper describes implementation detail of the whole system.

Keywords: *VOD, Android app, floor map, repository, Indoor positioning*

1. Introduction

The development of information technology and the rapid growth of computer networks allowed large files, such as digital images, to be easily transmitted in open networks such as the internet [1]. Video on Demand (VOD) allows users to select or search for a video and watch it. VOD has been so widely used that the majority of Internet traffic is video data [2].

Leveraging the Internet and the tremendous advances in wireless communications and localization technologies (GPS and others), mobile network operators rely on location based services (LBSs) as a key-asset to provide highly personalized and profitable services. An example of such services is the location dependent information services which provide a mobile client with relevant information regarding her/his current location [3].

Considering the advantages of VOD and LBS, this paper introduces a location based video service android application. This application displays a floor map, collects sensor values, detects if the user is not moving, finds out the current location of the mobile terminal and the object the user is watching, and plays a video that is closely related to the object.

The WLAN-based indoor positioning is widely used because WLAN is available virtually everywhere. However, its performance is poor because of its Ad hoc layout and signal fluctuation. Since a smartphone is equipped with a computer and many pretty accurate sensors, we use a dead reckoning indoor positioning app running on Android smartphones. It estimates moving distance and heading direction with the sensor values and the current

*Corresponding Author

location with the estimated distance and direction. It also adjust the estimated current location with information represented on the floor map.

2. Related Works

We have found an article [4] that introduces a system which is very close to ours. [4] proposes a novel location technology-based shopping service system (LSSS). LSSS incorporates handheld terminals, resource management system and wireless location technology together to make a whole system. The service provided by the system can make customers in a shopping mall conveniently get product information, notice their current location, figure out proper shopping route when multiple choices are in the shopping list, obtain the navigation information and receive other related innovative services. The purpose of [4] is very similar to the purpose of this paper. However, the methods of implementation used in [4] is significantly different from the ones used by us. For example, [4] uses a relationship between the signal strength and the distance in order to estimate the distance from the mobile terminal to the access point. We know that this estimate is extremely inaccurate. Our experiments showed that the average error of this method of positioning is greater than 10 meters. Another significant difference is the way of providing floor maps. [4] does not state that it uses an electronic map such as an AutoCAD file and it says that they use Google Maps API V3 for location map engine. We develop our own map engine that draws floor map with an AutoCAD file.

[5] proposes a location estimation scheme based on moving nodes that opportunistically exchange known positions. The user couples a linear matrix inequality (LMI) method with a barycenter computation to estimate its position. Simulations have shown that the accuracy of the estimation increases when the number of known positions increases, the range decreases and the node speeds increase. However, they performed simulations without addressing the problem of obtaining the range. This paper introduces an indoor location based service system that works in the real world.

High power white LEDs are expected to replace the existing lighting technologies in near future which are also suggested for visible light communication (VLC). [6] proposes an algorithm for high precision indoor positioning using lighting LEDs, VLC and image sensors. In the proposed algorithm, four LEDs transmitted their three-dimensional coordinate information which were received and demodulated by two image sensors near the unknown position. The unknown position was then calculated from the geometrical relations of the LED images created on the image sensors. They performed simulations and showed that the proposed system could estimate the unknown position within the accuracy of few centimeters. The proposed method is great. However, we need accurate image sensors in order to use the method. Our positioning method uses sensors that are equipped on ordinary smartphones and experimental results are discussed in this paper.

[7] surveys the fundamental techniques for indoor location based service (ILBS) development including positioning, database management, rendering drawings, and web services. For indoor positioning, K-nearest neighbor method, decision tree, trilateration and extended Kalman filter (EKF) are discussed. In practice, implementation of K-nearest neighbor method or decision tree requires collecting sensor values at every candidate points and it takes a long time. Trilateration and EKF require a relationship between sensor value and distance and a correct relationship between them is extremely hard to build. The indoor positioning method we introduce in this paper is a dead-reckoning method that determines the current location with the sensor values.

This paper introduces a smart VOD system. A schematic diagram of a VOD system introduced in [8] is shown in Figure 1. The streaming server that pumps out video file in a stream is the main component of the system. The user interface of the server interprets user's requests, retrieves the name of the video file that the user wants using database server, and returns the URI (uniform resource identifier) of the file to the user. Then, the user sends the URI to the streaming server so that the streaming server can pumps out the file to the user.

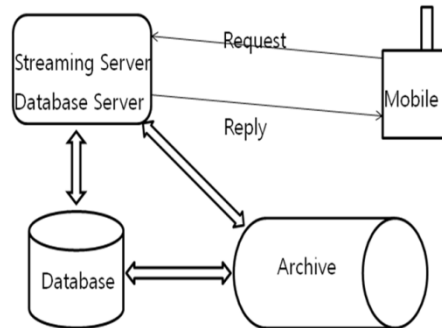


Figure 1. A Schematic Diagram of a VOD System

3. Repository of Floor Maps

The floor maps are essential part of the user interfaces of indoor location based services (ILBS). Google and some other companies are providing useful APIs (Application Programming Interfaces) for manipulating maps with which developers can easily use maps in their application programs. However, there is no such company or organization for floor maps. As the first step of developing general purpose floor map provider, this paper introduces online floor maps repository. The users of the repository can upload their floor maps to the repository and can use the floor maps in the repository in their application.

3.1. The Functional Requirements of the Repository System

The online floor map repository allows users to upload their own floor maps. They usually use AutoCAD in order to create floor maps. For all the large constructions, they already have floor maps. Some of them are extremely accurate and must be kept in secret while some are rough and should be open to public.

The repository also allows the users to search a floor map, to use the image of selected floor map and to upload information of interesting objects. It also allows the administrators to delete and update floor maps and information of interesting objects.

Therefore, we implement web services that provide the following functions:

- upload a floor map
- search a floor map
- rendering a floor map
- delete a floor map
- upload information of interesting objects

3.2. DB System Design and Implementation

For the repository, we need a database (DB) system consisting of the tables as shown in Figure 2. Information of floor maps is recorded in floor_map_tab. The userId of the person who uploaded the floor map is recorded in userId field. Information of who (userId) used which (map_Id) floor map is recorded in user_map_tab. Once a user used a map, he/she can evaluate the quality of the map.

Information of POIs (Point of Interests) is recorded in interesting_objects_tab. The fields of x_pos and y_pos represents the coordinates of the object corresponding to ob_Id.

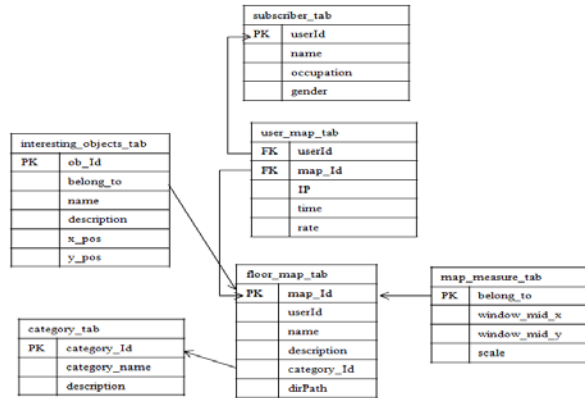


Figure 2. The Structure of the DB System

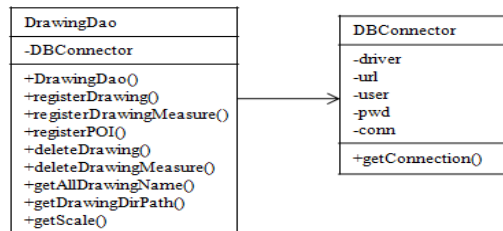


Figure 3. The Structure of the Classes for DB Manipulation

One of the most important functions of the online floor map repository is providing images of floor maps which are AutoCAD files. This implies that the repository system has to draw an image for a given AutoCAD file. Since the image is to be rendered on a smart phone, the size of the image is usually much smaller than the size of the AutoCAD image. The ratio of the size of the image for smart phone to AutoCAD image size is recorded in scale field of map_measure_tab.

Our database system provides DBConnector class and DrawingDao class. In the former, getConnection() method for DB connection is implemented. In the latter, many methods of manipulating DB are defined. The structure of these classes is shown in Figure 3.

- registerDrawing(): stores floor-map name, description, category, path into floor_map_tab.

- registerDrawingMeasure(): stores floor-map name, coordination of the midpoint of image, and scale into map_measure_tab.
- registerPOI(): stores information of POI including its coordinate into interesting_objects_tab.
- deleteDrawing() and deleteDrawingMeasure: delete information from floor_map_tab and map_measure_tab, respectively.
- getAllDrawingName(): retrieves all the name values in floor_map_tab.
- getDrawingDirPath(): retrieves the file path of a floor map from floor_map_tab.
- getScale(): retrieves the scale for a floor map from map_measure_tab.

3.3. Web Services of Map Management

For each of the functional requirements, we have implemented web service as follows.

- registerDrawing.jsp: When a request arrives with AutoCAD file, name, description and category, this service stores the information in floor_map_tab.
- deleteDrawing.jsp: When a request arrives with the name of a floor map, this service deletes the floor map from floor_map_tab and map_measure_tab.
- registerAP.jsp: When a request arrives with the image coordinate of the point touched by the user, MAC address, and so on, the service stores the information into interesting_objects_tab.
- showDrawingList.jsp: When a request arrives, this service returns all the names of floor maps.
- showDrawing.jsp: When a request arrives with a map name, this service returns the image of the map.

4. Indoor Positioning

There are many references that are closely related to our indoor positioning method. The authors of [9] proposed a new smartphone-based indoor positioning systems. The system determines user's moving direction with both magnetometer and gyroscope measurements. The system estimates the number of steps taken by the user with accelerometer values. The number of steps yields the distance the user moved. With the direction and the distance, the system estimates the user's current location with the following equation, where s_k , l_k , θ_k , s_E and s_N represent the current user position, step length, the estimated heading orientation, and the East and the North coordinates, respectively.

$$s_{u,k} = \begin{bmatrix} s_{E,k} \\ s_{N,k} \end{bmatrix} = \begin{bmatrix} s_{E,k-1} + l_k \cdot \sin(2\pi\theta_k) \\ s_{N,k-1} + l_k \cdot \cos(2\pi\theta_k) \end{bmatrix}$$

Our system recognizes moving status, watching status, and the current location of the user. Our system collects sensor values every 50 millisecond. It calculates the standard deviation of recent 20 y axis accelerometer values among the collected sensor values. If the calculated

standard deviation is less than a certain threshold then it determines that the user is not moving.

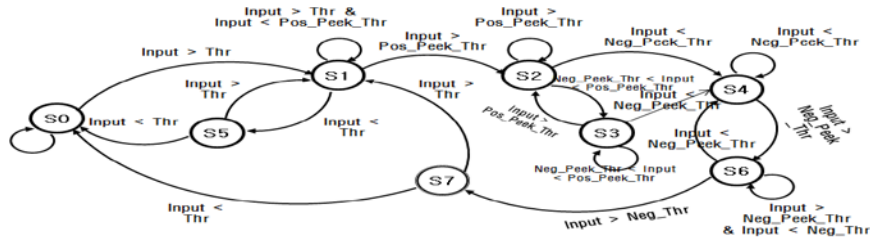


Figure 4. The Finite State Machine to Count Steps

We use the finite state machine (FSM) shown in Figure 4 in order to count user steps. From S_2 if input is greater than Negative_Peek_Thr and less than Positive_Peek_Thr then the next state will be S_3 . From S_3 , if input is greater than Negative_Peek_Thr and less than Positive_Peek_Thr then we stay there, if input is greater than Positive_Peek_Thr then the next state is S_2 , otherwise (input < Neg_Peek_Thr) the next state is S_4 .

The final context factor that our system detects is watching status. If a user does not move and holds the Android smartphone in the portrait orientation then we determine that the user is watching an object (exhibit).

5. The Smart VOD System

We have implemented those functions required to the repository system as web services as shown in Figure 5. AccessPoint is a web service to upload and retrieve the coordinates of an access point. Drawing is a web service to upload and retrieve a floor map. KNN is to build up a look-up table and to find the current location of the terminal.

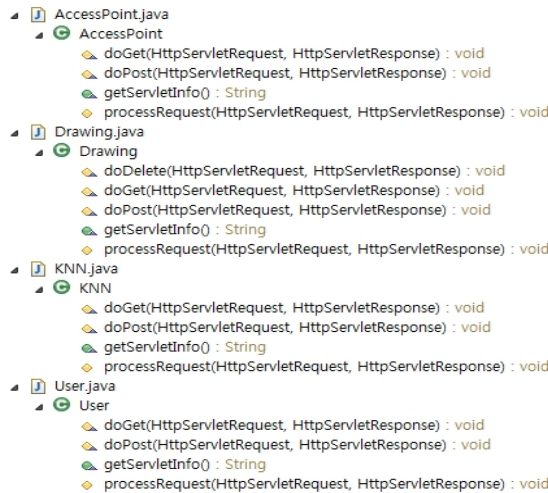


Figure 5. The Web Services for the Repository System

Using the web services, we have implemented an app with which we can upload, search and delete a floor map and upload information of a point of interesting (access point). One of the most tricky parts of this app is to find the image coordinate of the touched point. This app

renders the image of a floor map on an android:scaleType ImageView and defines the scaleType as MATRIX so that we can use Matrix class in order to move, zoom, and skew the image. Matrix class has many methods of manipulating the image. For example, postScale method of Matrix class zooms the image. Matrix class has 3x3 matrix and attributes of the image are recorded in this matrix. Using this information we can convert screen coordinate (x, y) into image coordinate as follows:

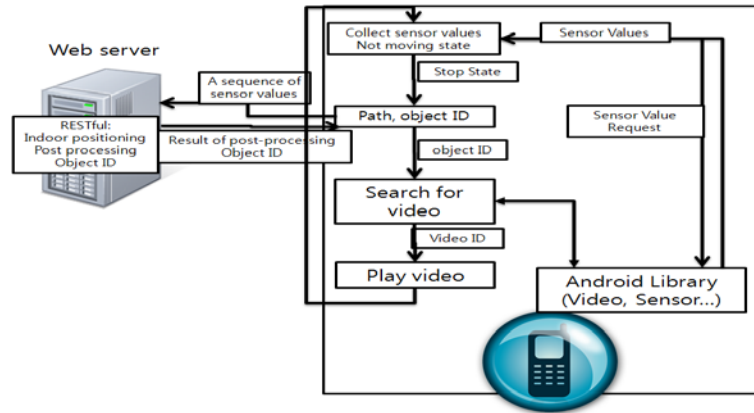
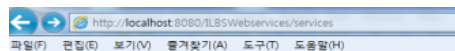


Figure 6. A Schematic Diagram of our Smart VOD System

```
float[] values = new float[9];
matrix.getValues(values);
image x = x/values[0] - values[2]/values[0];
image y = y/values[4] - values[5]/values[4];
```

Our smart VOD system is a client-server system as shown in Figure 6. The client system is an Android app. The app repeats collecting sensor values, checking if the user is not moving, sending all the collected sensor values to the Web server if the user is not moving, receiving the sequence of coordinates that represents the user's path and the object ID of the object the user is watching, searching for the video and playing the video.

We have implemented various indoor positioning methods as shown in Figure 7. FSMIndoorPositioning is the Web service that performs the positioning process discussed in Section 4.



And now... Some Services

- WeightedCentroidLocalization ([wsdl](#))
- Trilateration ([wsdl](#))
- PointofInternalDivision ([wsdl](#))
- ExtendedKalmanFilter ([wsdl](#))
- AdminService ([wsdl](#))
 - AdminService
- FSMIndoorPositioning ([wsdl](#))

Figure 7. The Indoor Positioning Web Services

6. Experiments

We have built up a virtual exhibition room as shown in Figure 8 as our test bed. The width of the room is 6.1 meters and the height is 12.72 meters. We assumed that there are 11 objects are exhibited. The first test scenario is to visit all the exhibits in the order of the number (label). The second test scenario is to visit in the order of 4, 9, 5, 11. We have tested many other scenarios.

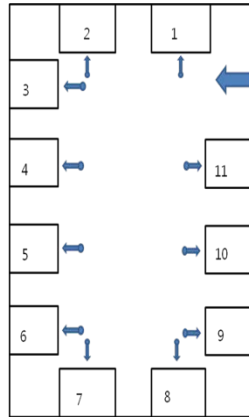


Figure 8. The Test Bed

Our Android app 100% correctly recognizes watching status and 80% correctly recognizes the exhibits being watched in the first scenario. However, if the user randomly walks and jogs around, then it performs very poorly.

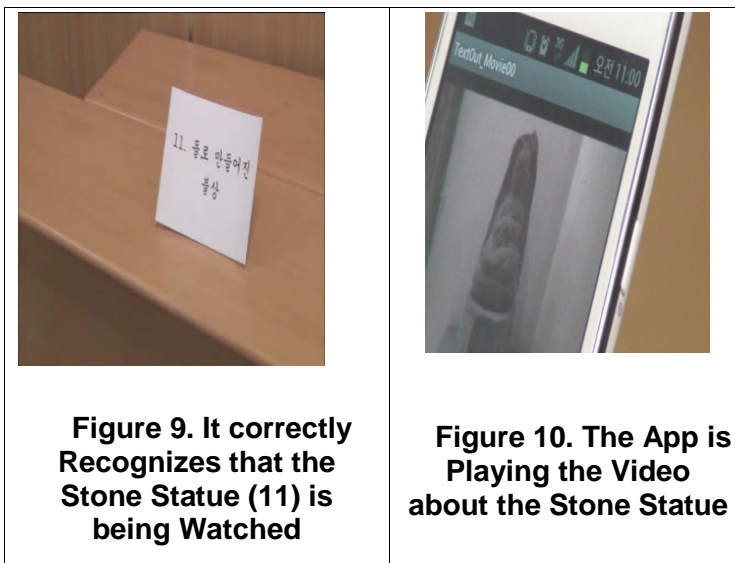


Figure 9. It correctly Recognizes that the Stone Statue (11) is being Watched

Figure 10. The App is Playing the Video about the Stone Statue

We have performed experiments of testing the repository system. Figure 11 shows screenshots of the app running on a smart phone. By touching the buttons shown in Figure 11(a), we can upload, search and delete floor maps. Figure 11(b) shows the result of rendering a floor map. Figure 11(c) shows a screenshot of uploading information of an access point (an interesting object).

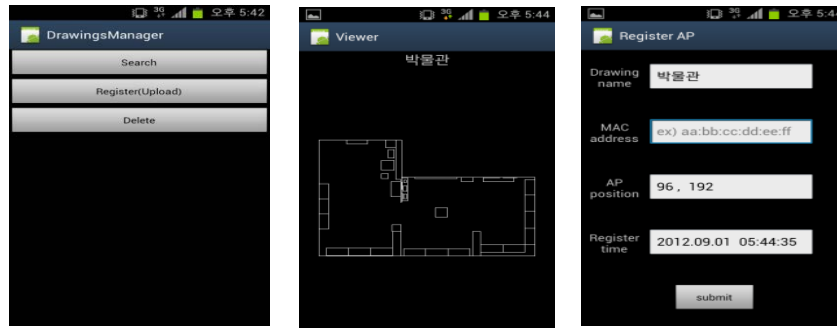


Figure 11(a)

Figure 11 (b)

Figure 11 (c)

Figure 11. Screenshots of the App Running on a Smart Phone

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (NRF-2011-0006942) and by 'Development of Global Culture and Tourism IPTV Broadcasting Station' Project through the Industrial Infrastructure Program for Fundamental Technologies funded by the Ministry of Knowledge Economy (10037393).

References

- [1] S. El-said, K. Hussein and M. Fouad, "Confidentiality and Privacy for Videos Storage and Transmission", *IJAST*, vol. 28, (2011), pp. 67-88.
- [2] J. Yim, G. Lee and T. Le, "Design and Implementation of VOD Database System", *IJMUE*, vol. 7, no. 2, pp. 503-508, (2013).
- [3] H. Slimani, F. Najjar and Y. Slimani, "Voronoi-Neighboring Regions Tree for Efficient Processing of Location Dependent Queries", *IJAST*, vol. 33, (2011), pp. 101-120.
- [4] J. Chen, J. Bao, J. Kim and J. Wang, "Location Technology-Based Mobile Shopping Service System", *IJHIT* vol. 5, no. 4, (2012), pp. 133-142.
- [5] G. Kang, T. Pérennou and M. Diaz, "Barycentric Location Estimation for Wireless Network Indoors Localization", *IJHIT*, vol. 2, no. 2, (2009), pp. 1-12.
- [6] M. Rahman and K. Kim, "Indoor Location Estimation Using Visible Light Communication and Image Sensors", *IJSH*, vol. 7, no. 1, (2013), pp. 99-114.
- [7] G. Lee and J. Yim, "A Review of the Techniques for Indoor Location based Service", *IJGDC*, vol. 5, no. 1, (2012), pp. 1-22.
- [8] J. Yim, "Design of the Recommendation Module for Context Aware VOD Museum Guide Android App", *International Journal of Software Engineering and Its Applications*, vol. 7, no. 2, (2013), pp. 273-285.
- [9] J. Leu and H. Tzeng, "Received Signal Strength Fingerprint and Footprint Assisted Indoor Positioning Based on Ambient Wi-Fi Signals", *75th IEEE Vehicular Technology Conference (VTC Spring)*, pp. 1-5, IEEE Press, New York, (2012).

Authors



Changyong Han, received the Ph.D. degree in Business Administration from Dongguk University at Seoul Korea, in 2013. He is a Professor in the College of Management-Tourism, Department of Business & Economics Dongguk University at Gyeongju Korea. His current research interests include Multimedia Systems, IPTV System and Open Smart media platform.



Jaegol Yim, received the M.S. and Ph.D. degrees in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Science at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, Location Based Service, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in Korean Language), and several undergraduate textbooks.