



Journal of Applied Research and Technology

ISSN: 1665-6423

[jart@aleph.cinstrum.unam.mx](mailto:jart@aleph.cinstrum.unam.mx)

Centro de Ciencias Aplicadas y Desarrollo

Tecnológico

México

Chen, L.-C.; Lai, Y.-C.; Yeh, Y.-H.; Lin, J.-W.; Lai, C.-N.; Weng, H.-C.  
Enhanced Mechanisms for Navigation and Tracking Services in Smart Phones  
Journal of Applied Research and Technology, vol. 11, núm. 2, abril, 2013, pp. 272-282  
Centro de Ciencias Aplicadas y Desarrollo Tecnológico  
Distrito Federal, México

Available in: <http://www.redalyc.org/articulo.oa?id=47426225011>

- How to cite
- Complete issue
- More information about this article
- Journal's homepage in [redalyc.org](http://redalyc.org)

[redalyc.org](http://redalyc.org)

Scientific Information System

Network of Scientific Journals from Latin America, the Caribbean, Spain and Portugal

Non-profit academic project, developed under the open access initiative

# Enhanced Mechanisms for Navigation and Tracking Services in Smart Phones

L.-C. Chen<sup>1</sup>, Y.-C. Lai<sup>\*2</sup>, Y.-H. Yeh<sup>3</sup>, J.-W. Lin<sup>4</sup>, C.-N. Lai<sup>5</sup>, H.-C. Weng<sup>6</sup>

<sup>1, 2, 6</sup> Department of Information Management  
National Taiwan University of Science and Technology  
Taipei County, 106, Taiwan  
<sup>\*</sup>laiyc@cs.ntust.edu.tw

<sup>3</sup> Department of Information Management  
Oriental Institute of Technology  
New Taipei 220, Taiwan

<sup>4</sup> Department of International Business  
Chien Hsin University of Science and Technology  
Taoyuan 320, Taiwan

<sup>5</sup> Department of Information Technology  
Hsing Wu Institute of Technology  
New Taipei 244, Taiwan

## ABSTRACT

Combining Global Positioning System (GPS) and Short Message Service (SMS), this paper develops a realistic system, called Mobile Navigation and Tracking System (MNTS), to provide navigation and target tracking services. MNTS is an Android based mobile application which integrated many enhanced mechanisms for navigation and target tracking services. MNTS not only provides users with the GPS navigation capability, but also supports Quick Response (QR) code decoding, nearby scenic spot searching, friend positioning and target tracking. In target tracking, MNTS utilizing SMS mainly adopts two proposed novel approaches: location prediction and dynamic threshold to reduce the number of short message transmissions while maintaining location accuracy within an acceptable range. Location prediction utilizes the current target's location, moving speed, bearing to predict its next location. When the distance between the predicted location and the actual location exceeds a threshold, the target sends a short message to the tracker to update the actual location. Based on the movement speed of the target, the threshold is dynamically adjusted to balance the location accuracy and the number of short messages. Furthermore, as MNTS is free and open-source software, service providers or developers can easily extend their own services based on this system.

Keywords: Navigation system, Target tracking, GPS, QR code, SMS.

## 1. Introduction

Recent advancements in mobile technology allow Global Positioning System (GPS)-enabled mobile devices to provide a variety of real-time Location-based Services (LBS). Among all those services, GPS navigation and location tracking are the most popular applications on mobile devices due to the usefulness and popularity during a trip. Formerly, a tourist has to look at paper maps or tour guide books during the trip, resulting in wasting time and easy to get lost. Therefore, electronic products embedded with navigation functions have become more and more popular. By using GPS navigation capabilities of those electronic devices, the travelers can arrive to an unfamiliar destination

speedily and effortlessly. At the meanwhile, a GPS tracking system is able to track a specific target's movement and relay information concerned where the target has traveled.

A GPS navigation device is a device that can receive GPS signals for determining the device's current location on Earth. Currently, the commonly-used hardware of navigation systems includes dedicated in-car GPS personal navigation device (PND) and GPS-enabled smart phones. PND is bulky and inconvenient to carry around while traveling since it is intended primarily for use in a car. On the other hand, the handheld smart

phones are ideal for palm-of-the-hand operation. They can be easily used in any kinds of transportation, so it is an excellent choice as the hardware platform of navigation systems.

There are a number of navigation software products available. GPS navigation software usually falls into one of the following two categories: commercial navigation software and free open-source navigation software. Since the navigation systems provided by the smart phone company or telecommunication operator are commercial products, which significantly limit developing some extended functions. Also the service from these companies isn't free. Hence, there is lack of free and open-source based GPS navigation system which integrates those useful applications. Recently, Google has developed the Android platform, which is an open system, offering high flexibility on development. Due to this, it is very useful to develop a GPS navigation system on the Android platform with combining many Google resources. With Google Maps' free navigation functionality, people may prefer the costless service from Google over that of paid services.

Thus, this paper develops a realistic system, called Mobile Navigation and Tracking System (MNTS), to provide navigation and target tracking services in Android smart phones. MNTS not only provides users with the GPS navigation function, but also supports QR code decoding, nearby scenic searching, friend positioning and target tracking. For software design, most navigation systems use GPS to get the current position, and start navigating after a user chooses destination or inputs its address. However, these steps are not very convenient to users. To avoid the inconvenience, many introductions of scenic spots or hotels have been transferred to easier symbols, namely, QR code, from the texts. Thus, the user can use the photographic function of a smart phone to get QR code and then directly obtain the coordinate of a scenic spot and related information. This user-friendly approach is very convenient for users when using navigation applications.

At the meanwhile, a GPS-enabled device can be a useful tool for keeping track of friends, family and even vehicles. There are several location-tracking applications have been developed [1-14], including 1) continuous location tracking of elders and children for safety reasons or to prevent them from

getting lost [1, 2]; 2) car monitoring and tracking [3-5]; and 3) intelligent transportation systems [6]. Most of above works used either 802.11 wireless network or Short Message Service (SMS) to transmit a target's location information to a tracker.

Although transmitting geo-location information of a target *via* wireless networks is effective when the target and tracker are both within Wi-Fi coverage, 802.11 wireless networks are not always available. Once a target or a tracker is unable to access Wi-Fi, location tracking becomes impossible. Thus, SMS is a relatively more reliable and flexible solution because of its widespread use (*i.e.*, well-structured worldwide) [6, 8]. However, SMS is a user-pay service. Thus the goal of the proposed tracking services in MNTS is to minimize the transmission cost of a tracking system by minimizing the number of SMS transmissions while maintaining location tracking accuracy.

This paper is organized as follows. Section 2 discusses related work of navigation system and tracking technique. Section 3 formally elaborates on the design of the MNTS. Section 4 describes the implementation of the MNTS based on Android phones, and then demonstrates some Graphic User Interfaces (GUIs), the functions and operating processes of MNTS. Finally, conclusions and future work are given in the final section.

## 2. Related works

### 2.1 Navigation system

The main navigation software can be divided into two types. The first one integrates navigation system, electronic maps, and GPS on mobile devices. This type of software, for example, Papago! (<http://www.papago.com.tw/>) and TomTom (<http://www.tomtom.com/>), can be used in an off-line manner. The second type has only installed the navigation system and GPS on mobile devices, and needs to download map information via Internet. Thus they, for example, Google Maps (<http://earth.google.com/intl/zh-TW/>) and Urmapp (<http://www.urmap.com/>), must be used in an on-line manner.

The first type of software is developed by GPS company or map providers, the accuracy of positioning is great and the functions about

planning routes are excellent, but they are all commercial products and do not provide the source code for revising or extending other functions. Besides, these maps are slowly or hardly updated in user's devices and are generally not customized for creating user's preferred scenic spots. The second type of software is developed by ISP or portal operator. The provided scenic spots, user's custom map, and programming flexibility are better. Among this type, Google Maps provides high flexible API and it is also most compatible with Android. However, its electronic maps are still needed to be downloaded via Internet and currently GPS navigation services are not provided yet. Therefore for the Android platform, a free navigation system is required and worthily developed. Therefore, this work focuses on developing such a system.

## 2.2 Tracking Technique

The GPS is widely used for target location because of its high positioning accuracy [1, 2, 3, 6, 8]. Additionally, GPS networks are well-constructed and widely available. Thus, GPS networks are practical for positioning an object by using triangulation.

Related works on the use of GPS and SMS for location tracking [3, 5, 6, 8, 9] can be roughly classified as time-based delivering and distance-based delivering. Time-based delivering periodically transmits location update messages for tracking [3, 6, 10]. In contrast, distance-based delivering issues update messages when the distance between the previously reported location and the current location exceeds a fixed distance threshold [9]. Time-based delivering may be effective for tracking a target that is moving erratically. However, it exhibits a crucial flaw when the target remains stationary for an extended period. Many unnecessary short messages are still periodically sent in such a case. The SMS is a user-pay service. Therefore, the goal of this study is to minimize the number of SMS transmissions while simultaneously maintaining location tracking accuracy.

The proposed approach differs from distance-based delivery in two major aspects. First, MNTS sends a location update message when the distance between the predicted location and the actual location exceeds a threshold, rather than the

distance between two continuously reporting locations, a typical mechanism adopted by distance-based delivering. Thus, MNTS can deliver fewer update messages. Additionally, since MNTS adopts a dynamic threshold rather than the static threshold used in distance-based delivery, MNTS further reduces the number of required update messages while maintaining satisfactory location accuracy.

## 3. Design of MNTS

MNTS is a GPS navigation and target tracking system built in smart phones with Android. The system has four requirements listed below:

- It has user friendly interface.
- It provides convenient and reliable navigation services.
- It integrates some add-value services.
- It is open-source software.

As shown in Figure 1, MNTS is composed of five subsystems: User Main Interface Subsystem (UMIS), QR Decoder Subsystem (QDS), GPS Navigation Subsystem (GNS), Contacts Application Subsystem (CAS) and Location-based Delivery (LBD). The function of each subsystem is briefed as follows:

- UMIS integrates all subsystems and display the executing results of other subsystems.
- QDS transforms captured QR code into corresponding geographic information (Geocode).
- GNS provides the user's current position, nearby scenic spot searching, and route planning.
- CAS uses instant GPS positioning function to perceive the friends' positions and track their paths on the map.
- LBD provides the target tracking capability by adapting location prediction and dynamic threshold to reduce the number of short message transmissions while maintaining location accuracy within an acceptable range.

We mainly used Android SDK and J2SE to develop MNTS. Also Zxing Barcode Libraries (<http://code.google.com/p/zxing/>) and Panoramio API (<http://code.google.com/p/apps-for-android/>) were used to assist the development of the QDS and CAS subsystems, respectively.

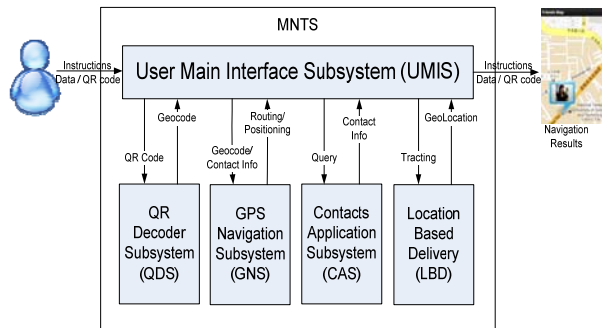


Figure 1. MNTS software architecture.

### 3.1 UMIS

UMIS is the main interface of MNTS. It offers loading and operating electronic maps, calling other subsystems, and showing the executing results of other subsystems. The user can click the control button to zoom in and zoom out maps and can drag maps to change observed scope of maps. Besides, the user can choose "Scanner" in Menu to call the QDS subsystem for turning on the photographic function to capture QR code and obtain geographic information after decoding. The user uses "Driving directions" in Menu to invoke the GNS subsystem to execute route planning after getting the geographic position. When the user wants to perceive the positions of friends, he (she) clicks "Friend" in Menu to call the CAS subsystem and knows the friends' positions. Finally, by pressing "Target tracking", users can call the LBD subsystem and start tracking a moving object.

### 3.2 QDS

This subsystem majorly uses the libraries of Zxing Barcode (<http://code.google.com/p/zxing/>) to decode QR code, which includes scenic spot information, into geographic information. The users first use camera on their phones to take a picture of QR code. Then the subsystem will automatically decode the obtained QR code and position the coordinate on the electronic map. The QDS

subsystem includes controlling module, decoding module, format transforming module. The user executes controlling module to invoke an interface for taking a picture of QR code. QDS will automatically call decoding module to decode the obtained picture and call format transforming module to transform the decoded information into geographic information. Finally, it sends the geographic information to the UMIS subsystem to show the positioning results.

### 3.3 GNS

The map information of Google is used to assist developing the GNS subsystem. Android SDK provides enough APIs for supporting Google Maps. GNS includes controlling module, position searching module, scenic spot searching module, and route planning module. When the user inquires an address by using position searching interface, position searching module will match the address with the database. When the user further searches nearby scenic spots, scenic spots searching module works. In addition, route planning module will arrange the routes and produce the results when user executes the "Driving directions" function.

### 3.4 CAS

The CAS subsystem includes two modules, directory module and friend positioning module. When user clicks "Friend" in the UMIS menu, the CAS subsystem acquires the user's current position via GPS, and uploads this position into our pre-installed positioning server, which gathers all persons' positions at all times. Then the user can choose the names of friends, that he (she) wants to know their locations, in the directory by using directory module. The friend positioning module will send the list of selected friends to the positioning server and the server will respond to the client the searching results about friends' locations. Thus the CAS subsystem can provide real-time tracking of friends' locations on the UMIS electronic map.

### 3.5 LBD

LBD consists of two primary modules: location prediction and dynamic threshold. The location prediction module, which is built in both target

and tracker sides, uses the present location, moving speed and bearing of the target to predict its next location. The dynamic threshold module, which is used only on the target side, minimizes the number of short messages by dynamically adjusting the threshold  $TH$  according to the moving speed of the target.

The tracker periodically updates the location of the target on the local screen according to the predicted location. However, when receiving a responded short message from the target, it means that the predicted location is far from the actual location. For showing more accurate location, the tracker updates the target's location using the information embedded in the received message, rather than its prediction.

Figure 2 shows the concept of the LBD. The GPS coordinate  $(X_1, Y_1)$ , moving speed  $V_1$ , and moving bearing  $B_1$  of the target at time  $t_1$ , is used to predict its next location at time  $t_2$ , denoted as  $(\hat{X}_2, \hat{Y}_2)$ . Once the distance  $D$  between the predicted location  $(\hat{X}_2, \hat{Y}_2)$  and the actual location  $(X_2, Y_2)$  exceeds threshold  $TH$ , the target sends the actual location to the tracker for updating. Thus, the objective is finding an approach that minimizes total short messages while maintaining  $D < TH$ .

The followings subsections discuss the two parts in further detail.

#### (1) Location Prediction

A target that receives GPS information, including coordinate (longitude, latitude) =  $(X_1, Y_1)$ , moving speed  $V_1$ , and bearing  $B_1$  (in radians, clockwise from north), from its GPS at time  $t_1$  can predict its next location at time  $t_2$ . Since the earth is not a plane, this predication is not straight-forward. We refer [15] to make this prediction on the basis of a spherical earth, although the earth is slightly ellipsoidal. Thus, a predicted location represented in terms of longitude and latitude  $(\hat{X}_2, \hat{Y}_2)$  can be obtained by Eq. 1 as

$$\begin{aligned}\hat{Y}_2 &= \text{asin}(\sin(Y_1) \times \cos(L) + \cos(Y_1) \times \sin(L) \times \cos(B_1)), \\ \hat{X}_2 &= X_1 + \text{atan}\left(\frac{\sin(B_1) \times \sin(L) \times \cos(Y_1)}{\cos(L) - \sin(Y_1) \times \sin(\hat{Y}_2)}\right),\end{aligned}\quad (1)$$

where  $L$  is the angular distance (in radians) and can be obtained by  $L=d/R$ , where  $d$  is the distance travelled and  $R$  is the radius of the earth (6371.01km). Thus we can obtain  $L$  by Eq. 2 as

$$L = V_1(t_2 - t_1)/6371.01\text{km} \quad (2)$$

Afterwards, calculate the distance  $D$  between the predicted location  $(\hat{X}_2, \hat{Y}_2)$  and the actual location  $(X_2, Y_2)$  can be obtained by haversine formula [15], as Eq. 3,

$$\begin{aligned}P &= \sin^2\left(\frac{\hat{Y}_2 - Y_2}{2}\right) + \cos(Y_2) \times \cos(\hat{Y}_2) \times \sin^2\left(\frac{\hat{X}_2 - X_2}{2}\right), \\ Q &= 2\text{atan}\left(\frac{\sqrt{P}}{\sqrt{1-P}}\right), \\ D &= R \times Q = 6371.01\text{km} \times Q.\end{aligned}\quad (3)$$

When the distance exceeds the threshold, i.e.,  $D > TH$ , the target transmits a short message to the tracker to indicate its actual location, moving speed, and bearing. After receiving the message, the tracker updates the target location and renews a new prediction process according to the information stored in the short message. However, if the distance is below the threshold, the target does not send a short message to the tracker. The tracker continues to use its predicted location to mark the target location because the gap between the actual and predicted locations is still acceptable.

#### (2) Dynamic Threshold

The threshold  $TH$  affects both the number of transmitted short messages and the location accuracy. A large threshold reduces the number of short messages but also reduces location accuracy, i.e., the gap between predicted and actual locations may be large. In contrast, a small threshold requires relatively more short messages but enhances location accuracy.

We observed that the user tolerance about perceived location accuracy mainly depends on the moving speed. When the speed of a target is faster, i.e., driving, the tracker can tolerate higher location inaccuracy. On the other hand, when the speed of a target is slow, i.e., walking, the tracker

requires higher accuracy about target's locations. As a result, a feasible way is to dynamically adjust the threshold according to the current target's moving speed. This solution satisfactorily balances the number of transmitted short messages and location accuracy.

Thus, according to the moving speed of a target, a dynamic threshold can be set by  $TH = V \times C$ , where  $V$  is the current moving speed and  $C$  represents a constant (e.g., 50 is used in subsequent experiments). Figure 2 depicts the application of the dynamic threshold concept. The figure clearly shows that  $TH$  is larger at  $t_2$  than at  $t_3$  because the moving speed of the target is faster at  $t_1$  than at  $t_2$ .

#### 4. Implementation of MNTS

##### 4.1 GNS subsystem

The developed GNS subsystem is composed of controlling module, position searching module, scenic spot searching module, and route planning module. Detailed illustration of each module is as follows.

##### (1) Controlling module

This module manages the operation of each module under the GNS subsystem and receives parameters

passed from other subsystems. It also delivers operation and calculation results to other subsystems.

##### (2) Position searching module

This module mainly provides position searching. It supports Forward Geocoding and Reverse Geocoding functions. Forward Geocoding locates the position (longitude and latitude) from address. Reverse Geocoding finds out the address from a given position. When the user inputs the destination's address in searching interface, this module will execute Forward Geocoding function and find out its position. On the other hand, when the user utilizes the QDS subsystem to get the longitude and latitude of QR-code destination, this module will invoke Reverse Geocoding in order to find out its address. The UMIS subsystem will show the results and plot them on the electronic map after getting the position information.

##### (3) Scenic spot searching module

This module is developed with open source software "Panoramio" API (<http://code.google.com/p/apps-for-android/>). Panoramio is a website which combines coordinates with pictures. Thus the users can share their pictures of scenic spots on the electronic maps. Using its provided API, this module can provide nearby scenic spots based on a given location and scope.

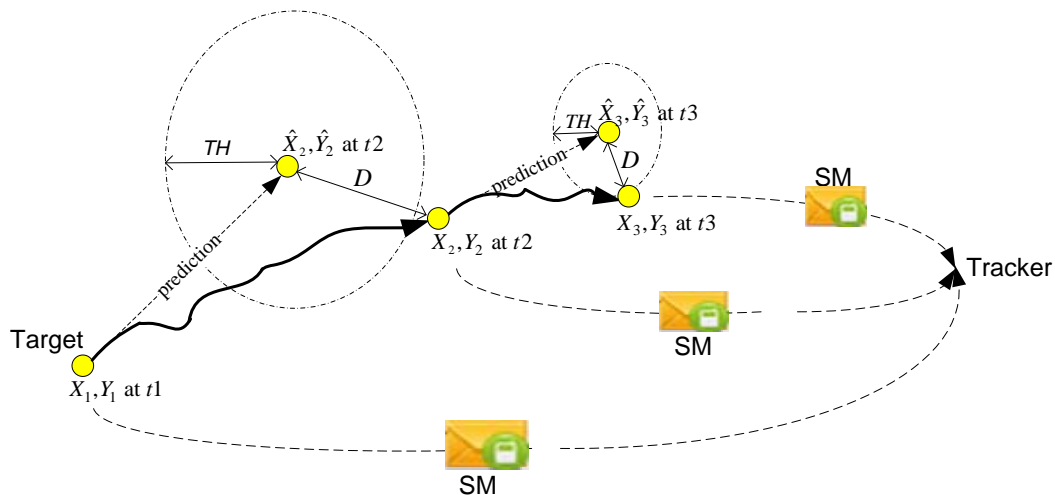


Figure 2. Concept of LBD.

#### (4) Route planning module

This module mainly provides the shortest path from the starting point to the destination. Since Android SDK does not provide a route planning library, we propose an novel approach to achieve this goal. As Figure 3 shows, based on given starting and destination points, this module sends a request to the Google Map server for route planning services. The server will send back the route planning of KML format. Then this module decodes the required elements (for example, Lable <Placemark>, <Point>...) according to the results of KML format. Finally it draws navigational route on the electronic map.

#### 4.2 CAS subsystem

The CAS subsystem includes two modules, directory module and friend positioning module. Figure 4 shows the overall message flows between user's phone (client) and the positioning server (server) in the CAS subsystem.

#### 4.3 LBD subsystem

The developed LBD system is mainly composed of tracker and target sides and built on GPS and

Android Application Framework, as shown in Figure 5. The tracker side contains four components: *SMS Switch*, *SMS Receiver*, *Predictor*, and *Offline Map*. The operation procedure is described as follows. First of all, for launching a tracking, a tracker uses *SMS Switch* to deliver a short message, "Start Tracking", to inform a target that the target should start sending its location when necessary. Also *Offline Map* will periodically update a target's location on the tracker's mobile phone according to *Predictor* which continuously predicts locations. However, when the tracker receives a new short message from the target, now the predicted location is far from its accurate location. Thus *Offline Map* updates the target's location according to the received message, rather than its prediction, for showing an accurate location. Specifically, the received messages sent from the target will be received by *SMS Receiver* in the tracker side. Then *SMS Receiver* extracts location information (e.g. coordinate, speed, and bearing) from the message, and passes it to *Predictor*. Once the tracking is finished, the tracker again uses *SMS Switch* to send a message, "End Tracking" to inform the target to stop sending its location.

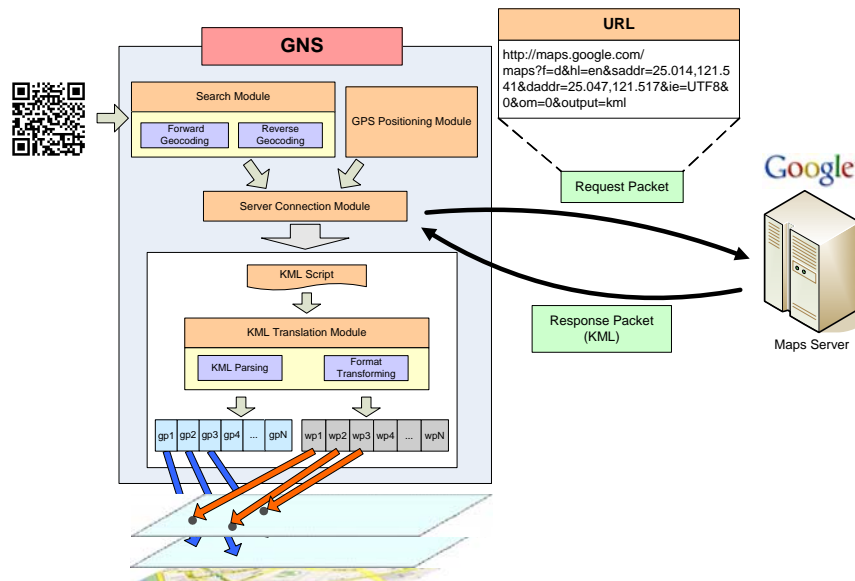


Figure 3. The GNS operation process.



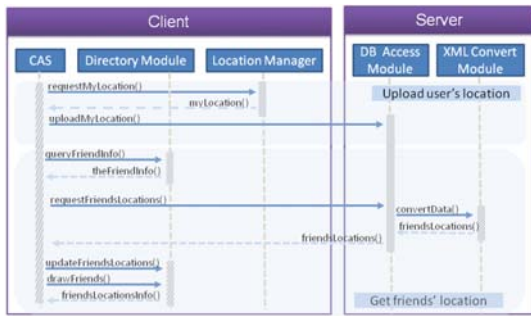


Figure 4. The CAS overall message flows.

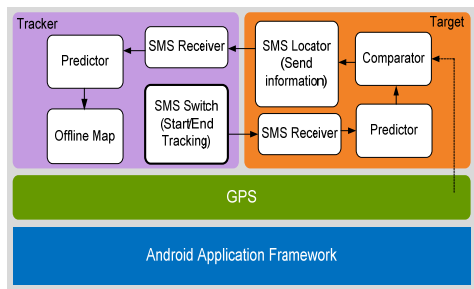


Figure 5. System architecture of location tracking system.

Similarly, the target side contains four components: *SMS Locator*, *SMS Receiver*, *Predictor*, and *Comparator*. The operation procedure is described as follows. Once receiving a start tracking message in *SMS Receiver*, the target starts predicting the next location by *Predictor*, which uses the same algorithm in the tracker to predict the current location. Thus the predicted locations in the tracker and the target are the same. Afterwards, *Comparator* will compare the predicted coordinate with its current actual coordinate obtained by GPS to check whether the distance between them exceeds the threshold. Once the distance exceeds the threshold, *SMS Locator* will send a short message including new information to the tracker. The above procedure will be performed iteratively until the track process is terminated.

#### 4.4 Functionality of MNTS

MNTS has some distinguished functions, like QR decoding, GPS navigation, friend positioning and target tracking, on the Android platform. The development and test environments are described as follows. Operating system for the smart phone is

Android 1.5. Programming languages are Java (version: 1.5.0\_17) and ASP.NET 2.0. SQL Server 2005 is used as the database software in the positioning server. Furthermore, Eclipse (version: Ganymede) is a Java development tool in Windows. After plugging-in Android Development Tools package, we can develop Android applications on Eclipse.

#### (1) Decoding QR Code

MNTS decodes QR code via the QDS subsystem. As shown in Figure 6(a), the user turns on the camera function in the smart phone and uses QR code acquisition block to capture QR code. Figure 6(b) shows the geographic information after QDS decoding (the content in this example is National Taiwan University of Science and Technology), and Figure 6(c) presents that the UMIS subsystem positions this geographic information on the map.

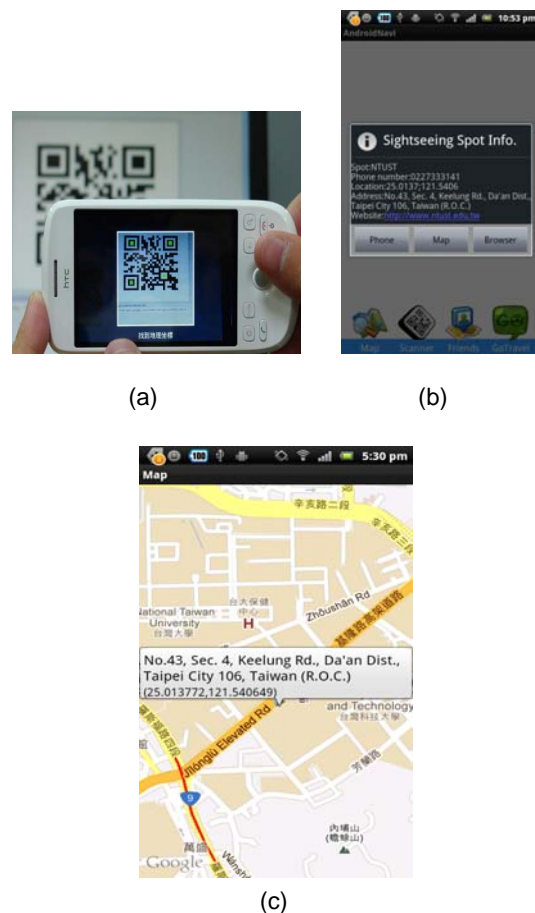


Figure 6. GUI in the QDS subsystem, (a) capturing QR code, (b) decoding QR code, (c) position results.

## (2) Navigation Services

As Figure 7(a) shows, after the QDS subsystem successfully decoded QR code and got the position, the user can search nearby scenic spots around this position. By the way, the user can use the position or a selected scenic spot as the destination, and request the GNS subsystem to produce a route planning from user's current location. At this time, GNS will plot the planning route on electronic maps, providing navigation services to the user, as shown in Figure 7(b).

## (3) Friend Positioning

CAS integrates directory with positioning to provide the friend positioning services, that is, the friends' locations will be displayed in the UMIS subsystem. Figure 8(a) shows the user browses the directory saved in the CAS subsystem. Figure 8(b) presents the user clicks "Show on the map" to track friends' locations and finally UMIS shows the results, as shown in Figure 8(c).

## (4) Target tracking

Figure 9(a) presents the GUI of a mobile phone on the tracker side. Pressing the Start Tracking and Stop Tracking buttons in Figure 9(a) can start and stop tracking a moving object respectively. Pressing the Map View button in shows the tracking information for a target on an offline map (Figure 10). Figure 9(b) presents the GUI of inputting phone number of the target.

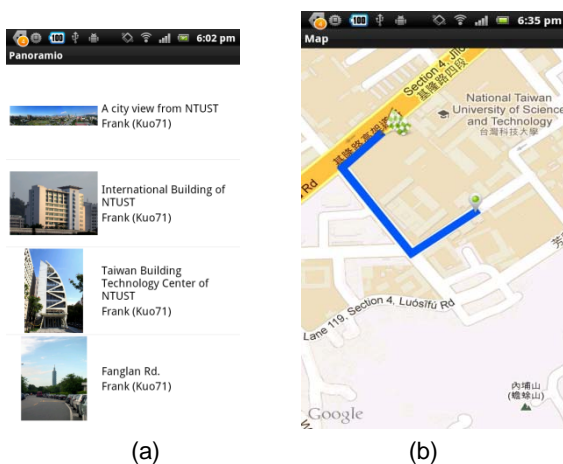


Figure 7. GUI in navigation services, (a) nearby scenic spots searched, (b) planning route generated.

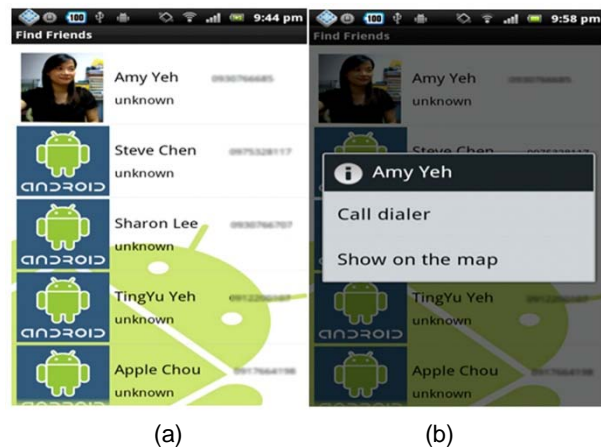


Figure 8. GUI in friend positioning, (a) directory saved in the CAS subsystem services, (b) selecting the friends in the CAS subsystem, (c) showing friend's position.

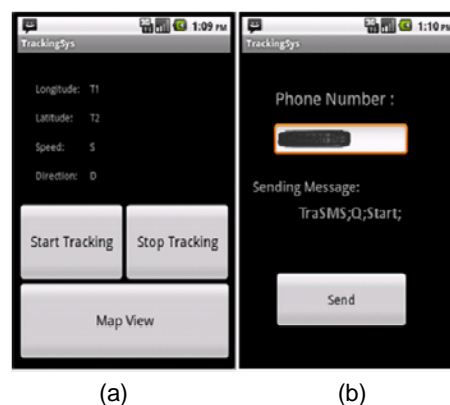


Figure 9. GUI of target tracking, (a) start and stop tracking, (b) inputting phone number.

Since providing an online map incurs a massive increase in transmission cost, electronic maps, consequently, are stored in the tracker's mobile phone in advance. The SQLite (<http://www.sqlite.org/>) database is embedded to save map information in the mobile phone. The SQLite is adequate when the amount of processing data is small or when the data are only used by single user. The SQLite database has many features in common with other public database systems, including the use of SQL-92 language syntax and the use of atomic, consistent, isolated, and durable (ACID) transactions. Instead of requiring the installation of a database system, the SQLite database presents in the form of a single file. Thus, it is quite simple to copy or create the database on a mobile device. A programming language can call the provided functions through the SQLite library.

Here, we used SQLite as database to store map information in a single file and get map information from it. Furthermore, a program, called RMap (<http://robertdeveloper.blogspot.com/>), is transplanted for the revealing of an offline map on an Android mobile phone. The RMap uses concept of overlays to draw the map and further marks the target's location on the map based on the information retrieved from short messages or from the predictor. Figure 10 illustrates the result of marking the target's location on the map, in which a user can manually zoom in or zoom out for desired granularity.

## 5. Conclusions

This paper developed the navigation and tracking system, MNTS, in smart phones.

MNTS provides many services, such as QR decoding, searching of nearby scenic spots, GPS navigation, friend positioning and target tracking to improve the touring quality. By adopting location prediction and dynamic threshold mechanisms, MNTS can reduce the number of SMS transmissions while maintaining location accuracy within an acceptable range. Main contributions of MNTS includes: (1) provide a free and open-source navigation and tracking system; (2) support QR decoding for conveniently determining the destination location; (3) provide the searching of nearby scenic spots and the positioning of

friends; and (4) reduce the number of SMS transmissions while maintaining location accuracy within an acceptable range. Furthermore, MNTS can be used as a basic platform for the peoples who want to develop their GPS navigation and target tracking systems.

In the future, we want to develop a new route planning algorithm with considering the dynamic traffic to improve the accuracy of navigation. Also adding more value-added services into MNTS or combining Radio Frequency Identification (RFID) with MNTS, like [16], can strengthen this system.

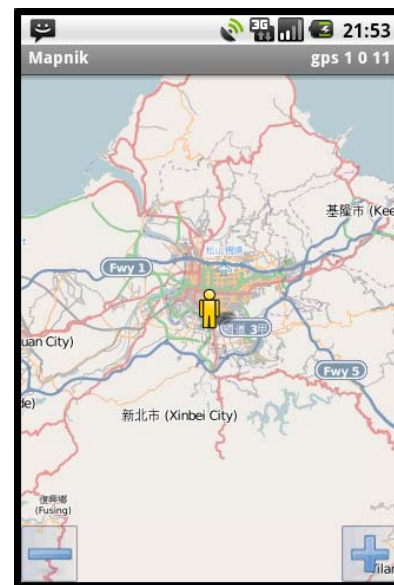


Figure 10. GUI of offline map.

## Acknowledgement

The authors would like to thank the National Science Council, Taiwan, R.O.C. for financially supporting this research under Contract No. NSC 101-2219-E-011-001.

## References

- [1] H. H. Lee et al., "Design and Implementation of a Mobile Devices-Based Real-Time Location Tracking," International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies, Valencia, Spain, 2008, pp. 178-183.
- [2] Z. Tian et al., "Location-based Services Applied to an Electric Wheelchair Based on the GPS and GSM Networks," International Workshop on Intelligent Systems and Applications, Wuhan, China, 2009, pp. 1-4.
- [3] I. Lita et al., "A New Approach of Automobile Localization System Using GPS and GSM/GPRS Transmission," International Spring Seminar on Electronics Technology, St. Marienthal, Germany, 2006, pp. 115-119.
- [4] P. Perugu, "An Innovative Method Using GPS Tracking, WINS Technologies for Border Security and Tracking of Vehicles," Recent Advances in Space Technology Services and Climate Change, Chennai, India, 2010, pp. 130-133.
- [5] S. Hameed et al., "Car Monitoring, Alerting and Tracking Model: Enhancement with Mobility and Database Facilities," International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, 2010, pp. 1-5.
- [6] R. E. Anderson et al., "Building a Transportation Information System Using only GPS and Basic SMS Infrastructure," International Conference on Information and Communication Technologies and Development, Doha, Qatar, 2009, pp. 233-242.
- [7] D. Munoz-Rodriguez et al., "Maximum Likelihood Position Location with a Limited Number of References," Journal of Applied Research and Technology, vol. 9, no. 1, pp. 5-18, 2011.
- [8] R. E. Anderson et al., "Experiences with a Transportation Information System that Uses only GPS and SMS," International Conference on Information and Communication Technologies and Development, London, U.K., 2010, pp. 4/1-10.
- [9] A. Civilis et al., "Techniques for Efficient Road-Network-Based Tracking of Moving Objects," IEEE Trans. Knowledge and Data Engineering, vol. 17, no. 5, pp. 698-712, 2005.
- [10] M. Zahaby et al., "Location Tracking in GPS Using Kalman Filter through SMS," EUROCON Conference, Saint-Petersburg, Russia, 2009, pp. 1707-1711.
- [11] A. Civilis et al., "Efficient Tracking of Moving Objects with Precision Guarantees," International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston, USA, 2004, pp. 164-173.
- [12] Y. Y. Xiao et al., "Location Prediction for Tracking Moving Objects based on Grey Theory," International Conference on Fuzzy Systems and Knowledge Discovery, Haikou, China, 2007, pp. 390-394.
- [13] P. H. Tseng et al., "Wireless Location Tracking Algorithms for Environments with Insufficient Signal Sources," IEEE Tran. Mobile Computing, vol. 8, no. 12, pp. 1676-1689, 2009.
- [14] R. Bajaj et al., "GPS: Location-Tracking Technology," Computer, vol. 35, no. 4, pp. 92-94, 2002.
- [15] Movable Type Scripts, Retrieved June 2012, from: <http://www.movable-type.co.uk/scripts/latlong.html>.
- [16] Y. L. Chen and X. Z. Shi, "Applications of Health Management Using Android and RFID," International Journal of Engineering and Technology Innovation, vol. 2, no. 3, pp. 216-225, 2012.