A Proactive Approach for Mobile GIS

[1]Wen-zhong Shi, [1]Kevin Kawai Kwan, [1]Geoffrey Shea
[1]Advanced Research Centre for Spatial Information Technology
Dept. of Land Surveying and Geo-Informatics
Hong Kong Polytechnic University, HK
lswzshi@polyu.edu.hk

[2]Jiannong Cao
[2]Internet and Mobile Computing Lab
Department of Computing
Hong Kong Polytechnic University, HK
csjcao@comp.polyu.edu.hk

Abstract - **One of the major obstacles for the mobile GIS development is the immature in database design. For this consideration, a new mobile data model based on a mobile view of spatio-temporal and attribute model is proposed. A temporal database is designed in physicality for the implementation on conceptual model. It is a proactive approach continuously updating the related spatio-temporal and attribute data according to the real location of the subscriber and the current time as well. At the end, performance of the dynamic database is evaluated for wireless GIS web application and on-line query. The design of this study increases the availability of GIS to the mobile services from fine-tune the existing database providing the latest accurate information to subscribers in a limited communication bandwidth and to improve the response time.**

Keywords-Mobile data model, Dynamic database, Location-based service, Mobile GIS, Wireless Web Applciation

I. INTRODUCTION

In recent years an extended form of GIS has emerged, mobile GIS, with which users can access the spatial, attribute and temporal data without any location limitation. Spatial analysis is no longer limited to a fixed environment, and it can be carried out at any place in any time.

Representation of real world in GIS has already increased from two- to three- and four-dimension. In the same time, size of the geo-database in GIS has been increased drastically. According to the research of system load testing for one thousand web pages from Mercury Interactive[1] (2001), only 15% web site was in proper working order. The researchers concluded that the system bottleneck occurred in four dominant areas: database (27%), network (25%), application server (23%) and web server (20%). We can see that the major system bottleneck is from database. Therefore fine-tuning the existing database system is an important step to improve the overall performance of Internet database.

For all mobile GIS applications or services, providing fast response and accurate information are two of the main goals. As a mobile client, he/she wants to get the corrected result by using the least time. However, the huge-sized database and the limited bandwidth in wireless communication have the negative effect to achieve the goal. This relates to the issue of response time.

In general, most mobile works do not involve complicated analysis. The service includes items such as data collection, spatial query, site selection, route finding and so on. For example, a mobile user wants to find out the nearest food shop within a desired distance from his current location at real time. Or for the maintenance work, the whole area is partitioned and each mobile worker responses a small part area and find out any malfunction of lamp stand. It is easy to understand that most works or services are only involved in large-scale data, e.g. in street level, and in a corner area, i.e. small amount of data. On the other hand, the information that client is interested in only a small subset from all availability in actual fact.

There has been a lot of research on the handling of the spatial-temporal data model, efficiency of data storage and data access and data query. This paper focuses on developing new data model for mobile GIS with the aim of improving response time and accuracy of searched information in mobile GIS web applications. We introduce the notion of location-based service that performs the optimum path and buffer measurement. The evaluations are: i) to improve response time of traditional "huge-sized" GIS database, ii) to improve the data accuracy. The performance of mobile application can be enhanced to achieve this goal.

II. CONCEPTUAL MOBILE GIS DATA MODEL

GIS data can be divided into two categories: spatial and attribute. Spatial data can be described as any information about location, shape and relationship among geographic features that the remotely sensed data and map data are also included (AGI, 1996). Attribute data can be described as the property, quality and characteristic of a geographic feature that appear as an entity in a relational data model and equivalent to a column in the relational table (Krzanowski, Palylyk, and Crown, 1995).

In the traditional spatio-temporal GIS model, a change refers to the difference of spatio-temporal objects in real world within a certain area of interest after certain time. However in a mobile GIS, a change of traditional spatio-temporal means, furthermore and a change means objects within the scope of interest changes when the mobile user moves from one location to another. In the former description, there is a change of spatio-temporal objects in the real world. In the latter model, however, the real world of spatio-temporal objects may not be changed, but just the objects within the interested area of the mobile user at particular location has changed, from a group of objects in one area to another group of objects in another area. A typical characteristic of mobile GIS is the changing environment dynamically. The conceptual mobile GIS model as proposed based on this significant difference between a traditional spatio-temporal GIS and mobile GIS.

We now define the conceptual mobile GIS model by extending the definitions of attribute change, spatial change & temporal change in GIS model. Attribute change can be referred to the inclusion of only the attributes that are related to the mobile application that is retained. Temporal change includes the alter of spatial and attributes at a different time period, a part of the element is discarded if they are meaningless to the application, and vice versa. Spatial change can extend to the environmental change while the mobile user at different position. This conceptual mobile GIS model is described by:

$$M_{ST} = \{t : s_i(t) \in ST,\ f''(s_i(t)) \in A;\ I_j \le i \le I_{j+n},\ T_{start} \le t \le T_{end}\}$$

where *Si(t)* refers to a spatial object at time *t*, *f''(Si(t))* refers to the function of spatial object *Si* at time *t*. The function relates to any attribute change of spatial object *Si* at time *t*. T_{Start} and T_{End} refer to the start time and end time of the mobile user stays in his current position, limited the time interval, *Ij* and *Ij+n* refer to the spatial objects within the scene of mobile users (limited to the area in the space that include both spatial and attribute information) and M_{ST} refers to all spatial objects within the scene while the mobile user remains the same place.

By applying the new concept of the mobile GIS model with variable spatial, temporal and attribute elements on the original dataset, a new subset is created, which describes the spatio-attribute objects interested by the mobile user at a particular time. The contents of this subset on which a mobile user is interested in at particular time and location of his/her movement. The database sensate accordingly is a "dynamic database" which contains the characteristic of active database that responding the changes of the object's status.

III. DYNAMIC MOBILE DATABASE DESIGN

An example is used to illustrate how the mobile GIS model works. The "scope" area of mobile user continuously change, the associated information is updated under his/her point of view. The model discussed is based on the following considerations: (1) Time is considered in one-dimensional and linearly ordered, (2) relational database is used to develop the GIS model, (3) there is no overlapping area between the areas of interest, it means that all spatial objects are different in each discrete "scope" area.

A spatial query related application was used to find out the location of an "international specialty" restaurant of his/her interest as the sample work. The user required inputting his/her current location and the maximum walking distance through an Internet ready mobile device. Figure 1 showed how the change in data volume of database and the creation of dynamic database at different stage if the new mobile GIS data model was added.

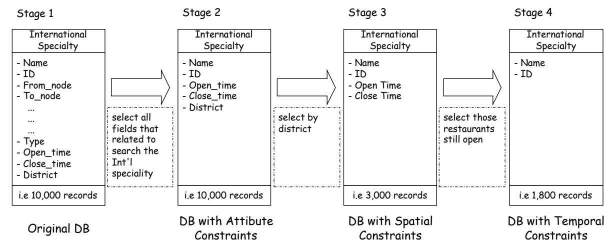


Figure 1: Dynamic DB creation based on spatial, temporal and attribute constraints

In stage 1, the full contents of the database in a traditional GIS were displayed. Assume that there were 10,000 records in total and each row recorded one spatial object only. From stage 1 to stage 2, the attribute constraint was applied that means to filter and only to keep those objects fulfill the attribute constraints – the selected restaurant of the international specialties with "Name", "ID", "Open_time", "Close_time" & "District" were retained. The other object than the selected ones were discarded from the interested data sets. "ID" was the unique identifier in database. "Name" was the description of restaurant. "Open_time" and "Close_time" indicated the working hours of restaurant. "District" described its location. From stage 2 to stage 3, the spatial constraint was further applied. Only the objects within the mobile user's "scope" were extracted, which may be useful for his/her at this particular location. For example, the remained spatial data from stage 2 was preserved within the same district of user. As a result, only 3,000 records are selected. From stage 3 to the final, the temporal constraint was applied. These restaurants that are not open during the time of the mobile user may use will not be selected, such as the restaurants were not in office hours at the time. As a result, a subset from the original database was created and only 1,800 records out of 10,000 were selected. The query work from the mobile application was executed on this subset – dynamic database. The steps from stage 2 to stage 4 were performed from time to time due to the movement of the mobile user.

$$M_{ST} = \{t : s_i(t) \in ST,\ f''(s_i(t)) \in A;\ I_j \le i \le I_{j+n},\ T_{start} \le t \le T_{end}\}$$

The notation representation of event from stage 1 to stage 4 showed in Table 1. In Table 2, the algorithm used to seek information is shown.

Table 1: Formative of sub data set from Stage 1 to Stage 4

Time	Event
Stage 1	Original Database = *A'*
Stage 2	$A' = \{a_j, a_{j+1}, \ldots, a_{j+n}\}$ where a_j is the attribute element that related to the application and there are *j* to *j+n* number of attribute elements, thus $A' \subset A$.
Stage 3	$S' = \{s'_i : g(s'_i) \in A',\ I_j \le i \le I_{j+n}\}$ where *S'i* are the spatial objects within the scene of mobile user, *g(Si)* refers to the function for *S'i* and there are *j* to *j+n* number of spatial objects.
Stage 4	$ST' = \{t : s'_i(t) \in S',\ g'(s'_i(t)) \in A',\ I_j \le i \le I_{j+n},\ T_{start} \le t \le T_{end}\}$ where *S'i(t)* are the spatial objects within the scene of mobile user at time *t*, *g'(S'i(t))* refers to the function for *S'i(t)* and the *t* refers to the user remains in the same position from start (T_{Start}) to end (T_{End}).

Table 2: Algorithm applied to various stage from generative sub data sets

Stage 2	CopyDatabase() GetFields() If fieldname != required Then delete field End
Stage 3	GetSet() Circ = Circle(current_location, buffer_size) SelectByCircle(, “Subset”, Circ)
Stage 4	GetCurrentTime While category != 0 Then GetSet() Select * where Open_time < Current_time and (Close_time - Current_time) > 60 and Category = particular value Category = Category – 1 End

IV. MODEL IMPLEMENATION AND PERFORMANCE ANALYSIS

Figure 2 shows the brief idea on mobile GIS model in real application. Two processes at least are involved. The first part is to generate a dynamic database from the original GIS database. Hence, two processes involved, creation and updating. The creation process doesn’t refer to any dynamic database at the beginning, and must be created before start up. The updating process refers to the re-creation of dynamic database since the contents outdate or move the user. The second part is the connection between dynamic database and the mobile applications. After the creation of the dynamic database, the request from application visits the dynamic database only instead of the original one.

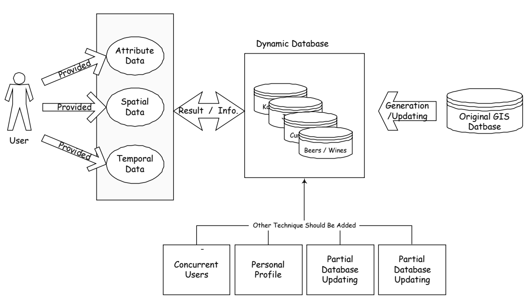


Figure 2: Workflow of Simulated Environment

In the system implementation, there are two major issues: what kind of method we should use to build the dynamic database and how well the model we will be. The target is to improve performance of query speed.

A location-based application used as an example to illustrate of the mobile GIS model in a system and apply for a real world example. The mobile device connected to the server and retrieved data via an active TCP/IP connection under cellular system and Hypertext Transfer Protocol (HTTP) was utilized as a data transfer protocol. All mobile applications were built on the web server and they served as the bridge for the mobile user and GIS server. Mobile GIS system acted as a module on the current system. All the spatial-related requirements were directly accessed to the dynamic database instead of the original one. Details of the working environment were shown in Figure 3.

The mobile user gave the initial position and he selected the interested district and type of restaurant, he/she also required to provide his/her current location, pass points and destination, and maximum walking distance from destination. Based on all these inputted information, the optimum path between all points and the location of specialties was found on the result map.

In order to reflect the real situation in Hong Kong, the International Specialty was chosen from the Yellow Page of Hong Kong which is developed by leading communications companies in Hong Kong, PCCW HKT and Telecom Directories Limited, as sample dataset. There were nine different types of international specialty and shows in Table 3. The hardware devices were summarized in Table 4.

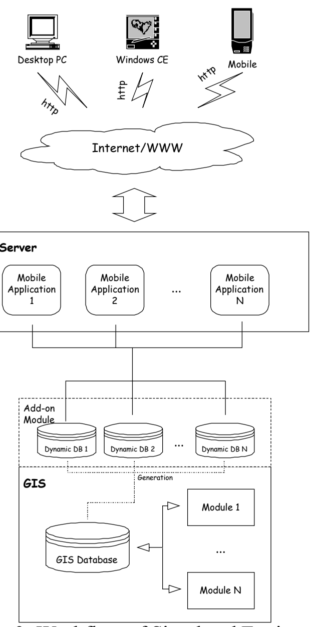


Figure 3: Workflow of Simulated Environment

Table 3: Types of International Specialty Restaurants from Yellow Page

Type	No. of Shops	Percentage
Beers / Wines	362	40.8
Curry / Satay	49	5.5
Japanese Noodles	19	2.1
Korean BBQ	14	1.6
Pizza	60	6.8
Sashimi / Sushi Bar	48	5.4
Steak House	42	4.7
West. Dessert / Drink	120	13.5
Others	174	19.6
Total:	888	100%

Table 4: Hardware and Software Configurations for the Experiment Study

Hardware	
Web Server & GIS Server	P-III 650 128M RAM
Mobile Device	Compaq iPAQ H3630
Software	
Web Server & GIS Server	Windows 98 Personal Web Server TransCAD
Mobile Device	Windows CE Version 3.0 Internet Explorer for CE
Development Kits	GIS Developing Kit

A. Time Measurement on Creation/Recreation of Dynamic Database

1) Method of Dynamic Database Updating

The two most popular methods, SET and INDEX, were applied to evaluate the best method for the creation and update of dynamic database. The algorithm of both methods was shown as follows. SET method used the SQL statement for the establishment and pointers were added in INDEX method.

SET method is to from a collection of objects with the same or similar attribute elements. On such a set, duplication do not exist. INDEX method refers to a key which is used to identify a unique record in order to make the database faster in searching specific record or sorting records by index field.

SET Method

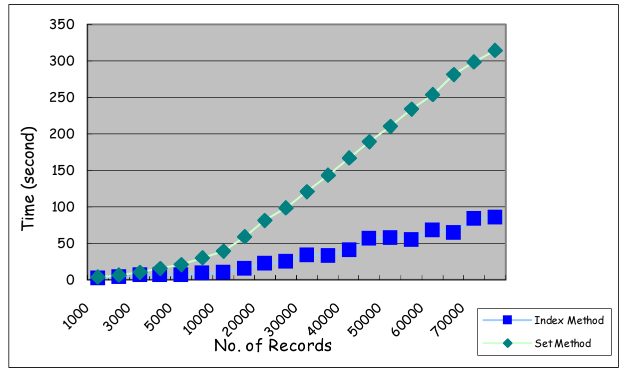
```
BEGIN
#get the server time
GetCurrentTime
While category != 0 Then
        #create the dynamic database
        CreateSet()
        #SQL search to find out those with 60 minutes left at least before closing
        Select * where Open_time < Current_time and (Close_time - Current_time)
> 60 and Category = particular value
        Category = Category – 1
End
END
```

INDEX Method:

```
BEGIN
#get the server time
GetCurrentTime
While Category != 0
        GetFirstRecord
        While return != 0
                GetRecordValue(ID, Category, Open_time, Close_time)
                #compare with the current category
                If Category != current
                        GetNextRecord
                Else
                        #compare with the current time to find out those with 60
        minutes left at least before closing
                        If Open_time < Current_time and (Close_time –
                          Current_time) > 60
                        Then
                             Goto Add_pointer_value
                             GetNextRecord
                        End
                  End
            End
Category = Category – 1
End
Add_pointer_value:
If current_record <> first_record Then Do
        SetRecordValues(pointer value = ID value)
Else
        LocateRecord(Index table, field)
        SetRecordValues(pointer value = ID value)
End
END
```

We characterize the time of a dynamic database creation and updating it in the following way. For the SET method, average response time measures the average time spent (in seconds) from the moment it begins to be issued to the moment all subsets are created. For the INDEX method, average response time measures the average time spent (in seconds) from the start moment to the moment all pointers and the index table are generated. The database sizes were from 1,000 records to 75,000 records.

Figure 4 below shows the time spent of the two different methods. The amount of time required increased sharply with



the increasing of the database's volume for the INDEX method. On the other hand, the time required for the SET method is nearly proportional to the volume size.

Time is a significant factor in mobile application. From the above result, it indicates that the performance of SET method is faster than the INDEX method on creating and updating work, thus, we apply the set method on the dynamic database creation and updating.

2) Performance Improvement in Time

We characterize the performance of the dynamic works in the following way. Average response time of the mobile client measures the average time spent (in seconds) from the time the query is issued to the time the results to the query are generated. For example, the function is to find out the pizza shop within 300 meters from the user's location, the time counted from after he enters all the criteria to get the full result.

The database sizes were reduced from 75,000 to 1,000 records. Based on the statistics at Table 3, the number of records was simulated in Table 5.

	No. of Records in Different Database Size (Calculation Based on Percentage)										
Type	1000	2000	3000	4000	5000	7500	10000	15000	20000	25000	30000
Beers / Wines	408	816	1224	1632	2040	3060	4080	6120	8160	10200	12240
Curry / Satay	55	110	165	220	275	413	550	826	1100	1376	1650
Japanese Noodles	21	42	63	84	105	157	210	314	4200	524	630
Korean BBQ	16	32	48	64	80	120	160	240	320	400	480
Pizza	68	136	204	272	340	510	680	1020	1360	1700	2040
Sashimi / Sushi Bar	54	108	162	216	270	405	540	810	1080	1350	1620
Steak House	47	94	141	188	235	353	470	706	940	1176	1410
West. Dessert / Drink	135	270	405	540	675	1012	1350	2024	2700	3374	4050
Others	196	392	588	784	980	1470	1960	2940	3920	4900	5880
Total	1000	2000	3000	4000	5000	7500	10000	15000	20000	25000	30000

	No. of Records in Different Database Size (Calculation Based on Percentage)								
Type	35000	40000	45000	50000	55000	60000	65000	70000	75000
Beers / Wines	14280	16320	18360	20400	22440	24480	26520	28560	30600
Curry / Satay	1925	2200	2475	2750	3025	3300	3575	3850	4130
Japanese Noodles	735	840	945	1050	1155	1260	1365	1470	1570
Korean BBQ	560	640	720	800	880	960	1040	1120	1200
Pizza	2380	2720	3060	3400	3740	4080	4420	4760	5100
Sashimi / Sushi Bar	1890	2160	2430	2700	2970	3240	3510	3780	4050
Steak House	1645	1880	2115	2350	2585	2820	3055	3290	3530
West. Dessert / Drink	4725	5400	6075	6750	7425	8100	8775	9450	10120
Others	6860	7840	8820	9800	10780	11760	12740	13720	14700
Total	35000	40000	45000	50000	55000	60000	65000	70000	75000

Table 5: Number of Records according to Database Size

Figure 5 shows the result of the average waiting time. We observed that the amount of time saved increased with the increasing database size. This value was also dependent on the actual time of the query. As the time changed, the size of dynamic database was updated and hence the waiting time also altered. But the time is much less than using the original one.

For the database size of less than 10,000 records, the difference in time count between the original database and the dynamic database was one second. However the database size increased to 65,000, the time different to access the original database was three times compared to time to access the dynamic database. Based on this trend, the time difference between the original database and the dynamic database becomes larger and larger when the database volume increases.

From Figure 4, we can see the dynamic database is much more effective in improving the performance of the systems in terms of decreasing the waiting time and providing the updated information to mobile users.

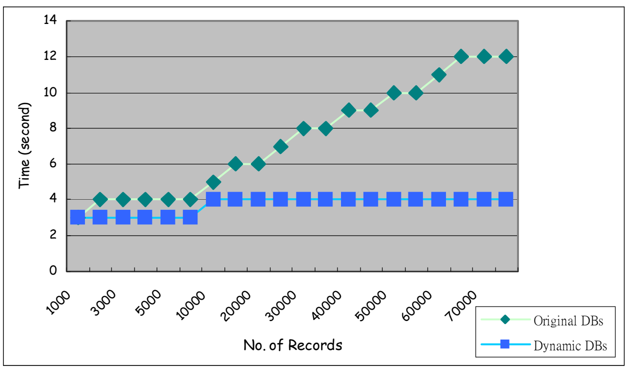


Figure 5: Average Response Time

V. CONCLUSION

This paper presented a new conceptual model in mobile GIS. Based on the new model, the dynamic database was created to respond the tasks from mobile application. The contents in the dynamic database were automatically updated to reflect the real situation of our world. We simulated an environment to investigate the time difference on the same spatial work in dynamic database with the conventional one. The response time was found to be improved in the new model.

The overall performance of the database system has improved as a result of the introduction of an alternative way to fine-tune the database.

REFERENCES

Dunham, M. H. and V. Kumar (1998). "Location dependent data and its management in mobile databases". [Internet] <http://ieeexplore.ieee.org/iel4/5718/15304/00707433.pdf>

Francica, J. (2000). "Location-Based Services Wireless Meets GIS", *GEOWorld*, December 2000.

Iyengar, S.S. and B. E. Pangburn (1998). "Real-Time Computing: Meeting the Client's Demand for Data Access via the Internet", Application-Specific Software Engineering Technology, 1998. *ASSET-98. Proceedings*. 1998 IEEE Workshop on, pp 14–19.

Jasper H. (1994). "Active Database for Active Repositories", Data Engineering, 1994. Proceedings. 10th International Conference, pp 375-384. [Internet] <http://ieeexplore.ieee.org/iel2/945/7004/00283054.pdf?isNumber=7004&prod=CNF&arnumber=283054&arSt=375&ared=384&arAuthor=Jasper%2C+H.>

Krzanowski, R. M., Palylyk, C. L. and Crown, P. H.(1995). "Lexicon of Terms for Users of Geographic Information Systems", *GIS World Inc.*

Laurini, R. (2001). "Real Time Spatio-Temporal Database", *Transactions in GIS*, Vol. 5, Issue 2, March 2001.

Lauzac, S. W. and P. K. Chrysanthis (2001). "Programming views for mobile database clients". [Internet] <http://ieeexplore.ieee.org/iel4/5718/15304/00707432.pdf>

Liu, M. K., F. Y. Wang, D. Zeng and L. Z. Yang (2001). "An Overview of World Wide Web Caching", Systems, Man, and Cybernetics, 2001 IEEE International Conference, Vol. 5, pp 3045 –3050. [Internet] <http://fp.ieeexplore.ieee.org/iel5/7658/20951/00971983.pdf>

Mercury Interactive (2001). [Internet] <http://www-heva.mercuryinteractive.com>

Mirghafori, N. and A. Fontaine (1994). "A Design for File Access in a Mobile Environment", *Mobile Computing Systems and Applications*, 1994. Proceedings, Workshop on, 1995, pp 57-62.

Pigot Simon (1992). "The Fundamentals of a Topological Model for a Four-Dimensional GIS", 5th International Symposium on Spatial Data Handling", Vol. 2, pp 580-591.

Potmesi, M. (1997). "Map Alive: Viewing Geospatial Information on WWW", *Proceedings of 6th International World Wide Web Conference*, 1997.

Renolen Agnar (1999). "The Temporal Set-Theory, Tropology, and Their Application in Spatiotemporal Modelling", Advances in Database Technologies, pp 466-473.

Stead, S. D. "Temporal Dynamics and Geographic Information Systems", *Spatial and Temporal Reasoning in Geographic Information Systems*, Chapter 15, pp 214-219.

Varshney, U. and R. Vetter. (2000) "Emerging Mobile and Wireless Networks", *Communications of the ACM*, June 2000, Vol. 43, No. 6.

W ilson, J. D. (2000) "Mobile Technology Takes GIS to the Field", *GEOWorld*, June 2000.