Performance Tuning in the MacauMap Mobile Map Application

Robert P. Biuk-Aghai
Faculty of Science and Technology
University of Macau
Macau S.A.R., China
robertb@umac.mo

Abstract

With the increasing popularity of mobile computing platforms such as personal digital assistants (PDAs) and smart mobile phones, applications originally designed for higher-performance devices are moving onto these resource-constrained mobile devices. Given that these devices have severely restricted processing power, memory, and user interface capabilities, applications intended for these devices have to be carefully designed and implemented to achieve acceptable performance. In this paper we report on our experience with the development of the MacauMap application, a mobile digital map application that runs on PDAs and smart mobile phones. Memory requirements were kept low through a special data format. The main challenge of satisfactory map drawing speed was addressed through a variety of techniques developed for this application. We present the MacauMap application, introduce its design, and conclude with lessons learned from its development.

1. Introduction

The last quarter of the 20th century saw the expansion and penetration of the personal computer into offices and homes, whereas the beginning of the 21st century witnessed a similar phenomenon for handheld computing and communication devices. Personal digital assistants (PDAs) and even more so mobile phones have become widespread, and this trend appears to continue well into the future. The International Telecommunication Union reported that by 2002, the number of mobile phone subscribers had already surpassed that of fixed phone subscribers [5]. This is particularly impressive when considering that mobile phone were commercially introduced only about a decade ago.

We use the term mobile computing devices to refer to PDAs and smart mobile phones collectively. A smart mobile phone is one with significantly more processing and memory capabilities than ordinary mobile phones, and is usually equipped with software applications similar to those found on a PDA. Examples of PDAs include the ones operating under the PalmOS and Pocket PC systems, whereas examples of smart mobile phones include those operating under the Symbian OS, PalmOS, or Microsoft Windows Mobile operating systems.

Because it is relatively easy to develop software for PDAs and smart mobile phones, they have become popular platforms for deploying a wide range of applications. These now include many that were formerly found only on desktop PCs, such as internet browsers, office applications, multimedia applications, etc. Mobile computing devices thus offer their users significant functionality coupled with extreme mobility: because of their small size, they can be carried and used practically anywhere. However, given the limitations in performance of mobile computing devices, it can be challenging to achieve software performance levels that are acceptable to their users.

In this paper we report on our own experiences with the development of a computationally intensive application on mobile computing devices: the MacauMap mobile map application. We started its development in September 2001 and have since completed versions for PDAs running PalmOS (June 2002), PDAs running Pocket PC (August 2003), smart phones running Symbian OS with Series 60 user interface (June 2004), and are currently completing another version running on Symbian OS with UIQ user interface. The PDA versions of MacauMap have been released to the public for free download (available at www.macautourism.gov.mo) and have proven popular: some 100,000 copies have been downloaded in the first 12 months since the software’s initial release. In the near future we will release the smart phone versions of MacauMap to the public, also for free download.

The remainder of this paper is organized as follows: Section 2 gives an overview of the MacauMap application, Section 3 outlines some important design considerations in the development of MacauMap, Section 4 discusses lessons learned, and Section 5 presents conclusions.
2. Overview of MacauMap

Driven by the need to promote tourism in the Macau territory, we set out to develop the MacauMap mobile digital map application. Since the Macau government declared Macau a ‘city of tourism’ in 1961 [7], the tourism industry in the territory has experienced a rapid and continuous growth, becoming the leading economic sector in Macau [1]. In 2001, around 10 million visitor arrivals to the territory were recorded [3], the majority of whom came to Macau for holiday purposes. As maps are an important tool for tourists, and given the spread of mobile computing devices, the commencement of development of MacauMap in 2001 appeared timely.

The main objectives of MacauMap are to provide following types of information:

1. Basic geographic information about the Macau territory, such as coastal outlines and the network of streets.
2. Information about places that are of interest to tourists.
3. Information about public transport facilities in Macau.
4. Information about hotels and restaurants in Macau.

2.1. User Interface

As mentioned in the introduction, the MacauMap application is now available on three platforms: PalmOS PDAs, Pocket PC PDAs, and Symbian OS Series 60 smart phones, with a fourth one for Symbian OS UIQ in development. Figure 1 shows the user interface of the MacauMap application on the three currently available platforms. On the two PDA platforms (PalmOS and Pocket PC) the interface includes a map view together with a toolbar of the most commonly used functions. Both of these versions also provide a popup menu containing other functions. On the smart phone platform, because of the small screen size as well as the limited input capabilities (keypad input versus the stylus input available on PDAs) the entire screen is occupied by the map view, and functions are available through shortcut keys as well as a separate menu view.

2.2. Functions

MacauMap provides following main functions:

1. Map manipulation: zooming in/out, panning, navigation history (previous/next map view).
2. Search: searching for streets or places and displaying their location on the map.
4. GPS: display of the user’s current location using a reading obtained from a connected GPS device (available on PDA versions only).
5. Bus information and bus route calculator: browsing of public bus information, and bus guide for calculating an optimal bus route from a starting bus stop to a destination bus stop.
6. Sightseeing guide: providing information about museums, churches, temples and other places of interest, as well as their location on the map.
7. Hotel and restaurant guide: providing a choice of restaurants and hotels matching criteria of location, class and style, and displaying their location on the map.

Except where noted above, all of these functions are available on all the different platform versions of MacauMap. Some of the main functions are illustrated in the following figures, with the PalmOS version of MacauMap. Map zooming is shown in Figure 2. Figure 3 shows the bus routing function, where MacauMap finds the optimal route from the bus stop “Terminal Maritimo” to the bus stop “Almeida Ribeiro”.

The inclusion of tourist-specific functions, such as sightseeing guide, hotel and restaurant guide, bus routing function, and others, distinguish MacauMap from ordinary handheld map applications, and make it specifically a tourism-oriented application.

3. Design Considerations

Map applications are software applications that process large amounts of geospatial data, involving heavy computation. Traditionally, these applications have resided on high-performance workstations and servers equipped with the necessary resources: large amounts of primary and secondary memory, fast CPUs and graphics processors, and large screens for displaying the data.

With the move onto mobile computing devices, these applications run into serious performance limits [6]. For example, our first deployment device for MacauMap (in 2002) was a Palm m515, equipped with a 33 MHz CPU, 16 MB of RAM, and a screen resolution of 160x160 pixels. Compared with a typical desktop workstation of the same time, equipped with a 2 GHz CPU, 1 GB of RAM, and 1600x1200 pixel screen resolution, the PDA offers less than 2% of the performance of the desktop. However, MacauMap uses vector graphics, instead of bitmap images of maps, in order to produce better quality maps at any zoom level. Thus, in order to perform with a response time that a user would consider acceptable MacauMap had to be carefully designed and implemented.
Figure 1. MacauMap user interface on different platforms

Figure 2. Zooming into a map

Figure 3. Finding the shortest bus route from a given start to a given end bus stop
When developing MacauMap, certain performance targets were defined: for the majority of map views, map operations should complete within one second, with only a small number of map views allowed to take up to two seconds. The latter are those maps consisting of particularly many objects, which is generally only the case for maps at the zoom levels between 1x and 4x (where the 1x zoom level shows a map of the entire Macau territory).

We also aimed to minimize the amount of memory required for installing MacauMap: the main application and map data combined should use no more than 1 MB of memory, in order to fit into older PDAs with small memory.

In order to draw a map on the screen, data of the portion of the map to be displayed needs to be retrieved from storage, screen coordinates corresponding to map coordinates of all map objects (such as streets, points, name labels, etc.) need to be calculated, and these need to be drawn on the screen at the calculated coordinates. In addition, screen buffers need to be managed and switched (we decided to use double buffering in MacauMap for better display quality, i.e. drawing takes place in an offscreen buffer which is then copied to the main screen buffer), and various other functions requested by the user need to be performed. However, through initial prototyping we found that the three main computing tasks affecting performance of the application were, in order of importance:

1. Map objects’ screen coordinate calculation
2. Data retrieval
3. Map drawing

Achieving our performance goals necessitated certain trade-offs in terms of map detail and functionality, as compared to desktop map applications. These are discussed in the following subsections.

### 3.1. Data Format

Data used by the MacauMap application was originally in two different formats: basic map data, including information on street centrelines, coastal outlines, green areas, lakes, and places of interest, were provided by the Macau government’s Cartography and Cadastre Bureau and were in ESRI Shapefile format [4], a popular GIS (geographic information systems) format. Bus route data were provided by the Macau government’s Lands, Public Works and Transportation Bureau and were in AutoCAD format.

Given that the source data was partly in ESRI Shapefile and partly in AutoCAD formats, two challenges existed: firstly, how to use data in two different data formats; and secondly, how to reduce its size. The ESRI Shapefiles of the basic map data have a combined size of around 1.5 MB, whereas the AutoCAD files of the bus route data measure about 30 MB. The target platform, however, and in particular older PalmOS PDAs, has only a small amount of memory available. Because of the organization of memory on PDAs, this amount of memory is used as both primary (i.e. working) and secondary (i.e. storage) memory. It has to accommodate the operating system and all user applications, while leaving enough available memory for running applications.

Moreover, the original source data was designed for use on a desktop workstation with a large screen, and does therefore contain a high level of detail. The screen of a PDA, however, is comparatively much smaller, which makes it impossible to display the same amount of data in any legible form.

The first step to tackle these two challenges was to manually convert the bus route information to ESRI shapefile format. This solved the problem of having to deal with different data formats. It also greatly reduced the data size, from nearly 30 MB to under 200 KB, a reduction by a factor of 150. This was possible by omitting much of the information contained in the AutoCAD files, such as street layout and building information.

However, even after converting the AutoCAD data files to ESRI Shapefile format, it was still evident that the data size was too large, measuring about 1.7 MB. A further reduction in the data’s size was achieved by map data generalization. Generalization is a procedure that simplifies polylines by reducing the number of line segments, for which several well-known algorithms exist [2]. This had the advantage of simplifying the map data for drawing (benefiting the performance of map drawing, discussed in the following subsection), while simultaneously resulting in a reduction in size. The generalized map data measured only about 1.4 MB.

The generalized map data, however, was still larger than our target memory size of 1 MB. Therefore we decided to create a custom data format for the MacauMap application which would consist of only the essential data elements. In addition, considering that data retrieval is the second most important performance factor of MacauMap, the design of the map data was optimized to support the data retrieval operations required by our application.

For instance, a common operation is to retrieve the road centreline coordinates of all the roads that are within a given region of the map. To optimize the retrieval speed for this type of data, the MacauMap data format divides the data related to roads into separate database records. Thus there is a road index record consisting of the bounding box coordinates of each road, sorted in ascending order. If a road’s bounding box is found to intersect with the current map view region, a pointer from the index record into another record containing the actual map coordinates of the line...
segments of that road can be followed to retrieve that data. Given that the road index is much smaller than the entire road data, and that bounding boxes for all roads are calculated and stored in it beforehand, the performance of this particular retrieval operation is greatly improved. Similar techniques are used with other common retrieval operations (the foregoing discussion applies mainly to the PDA versions of MacauMap; on the Symbian OS smart phone platform the data format is somewhat different, as Symbian OS provides a simple relational database system).

A conversion program was developed to convert from ESRI Shapefile format to the MacauMap data format, outputting a file in the format required by the respective version, e.g. a PalmOS database file, or a Symbian database. The resulting data is stored in an efficient format that only occupies a total of 250 KB, while at the same time facilitating efficient processing. The MacauMap application measures an additional 250 KB, for a total memory consumption of application and data of only 500 KB, only half of the targeted size of 1 MB (optional add-on databases containing tourist spot information and photos increase this size by up to 1.1 MB, but can be omitted if the user’s device does not have sufficient free memory).

3.2. Map Drawing

The major design challenge in the development of MacauMap was achieving satisfactory map drawing performance. As data retrieval could be performed efficiently, thanks to the data structures employed which were discussed above, the major remaining performance bottleneck was map drawing. Initial prototypes required up to ten seconds for drawing a map. As mentioned earlier, performance targets of 1–2 seconds had been set, so it was essential to bring map drawing time down.

The main reason for the slowness of map drawing is the large number of objects (points and lines) that need to be drawn on the screen. We found this particularly on our first target platform, PalmOS-based PDAs. The PalmOS API is extremely simple and has no provision for advanced graphics beyond drawing points and lines. For instance, there is no flood fill API function. Therefore nearly all graphics functions had to be implemented in MacauMap.

A number of steps were taken to improve map drawing performance:

1. Filtering: Macau has a total of around 1200 streets in an area of just 25 square kilometres. Many of these streets are tightly packed together in a small area. Therefore, some areas appear cluttered on the screen, particularly at low zoom levels. To overcome this problem, filtering is applied to eliminate small streets from the map view at low zoom levels. To make this possible, a “street level” field is saved along with the data for each street to indicate whether it is a major, intermediate, or minor street. At different zoom levels, only some groups of streets are shown, e.g. only major, or only major and intermediate streets. This has the desirable effect of reducing the number of lines that need to be drawn on the screen, and thus improving map drawing performance.

2. Converting map data to integer: The original source map data stored map coordinates as real numbers. However, calculations involving real numbers are slow, and this is particularly noticeable on a computing device with heavily constrained computing resources such as a PDA or smart phone. For this reason, all map coordinates were converted to integer, and all calculations involving these coordinates were carried out as integer calculations. However, because calculations such as integer division involve some loss of accuracy because of the rounding that takes place on the result, the map coordinates of the source data were first converted to large integers, 32-bit integers in our initial design on PalmOS-based PDAs, and 24-bit integers on our latest smart phone-based design, so that the loss of accuracy involved in the calculation of these numbers would be insignificant. In this way, a performance improvement of about 100% was achieved. The 24-bit (3-byte) coordinate on the smart phone platform is completely non-standard and is not supported by our programming language, OS, or database. However, we found that all required coordinates could be accommodated within 3 bytes, and reducing the coordinates’ data by 1 byte resulted in a saving of 25% of storage. In order to generate this data, we stripped the leading blank byte off our 4-byte map data and stored the resulting 3-byte values consecutively in a single record. Upon retrieval, the blank byte was added back to each 3-byte value to produce a value that can be processed by our application.

3. Different map display modes: To accommodate the large variety of PDAs with different performance characteristics, MacauMap allows the user to control the amount of detail displayed in the map. This is done by offering three different map display modes: (1) Simple map display, with streets represented by centre line only; (2) Intermediate-detail map display, with streets represented by centre line, and filling of map background with different colours for land, sea, parks, etc.; (3) Detailed map display, with streets drawn with left and right border, and filling of map background with different colours. Each of these display modes requires a different number of objects to be drawn, with the simple map display requiring the fewest and the detailed map display requiring the most objects. Differ-
ent users may choose the amount of detail they desire: those with a slow device may choose a simple map display, whereas those with a faster device may choose the detailed map display to get a better-quality map.

Through the above means, map drawing performance was greatly improved, so that now even older devices are able to achieve the map drawing performance targets of 1–2 seconds for most map views.

4. Lessons Learned

Developing processor-intensive applications for mobile computing devices is challenging, given the severe limitations of processor power and memory size of these devices. Through our development of the MacauMap handheld application over the past few years we have learned a number of lessons:

1. Data volume should be reduced as much as possible to achieve acceptable performance. Reduced data volume not only leads to reduced amount of data that needs to be retrieved, but also to reduced number of objects for which screen coordinates need to be calculated and that need to be drawn on the screen. A number of techniques are available toward this end, including several map generalization algorithms, which should be taken advantage of.

2. Database structure can greatly reduce retrieval times if common retrieval tasks are considered during design. Importantly, as data in mobile computing devices typically is stored in RAM rather than on a disk storage device as is typical on a desktop computer, it becomes possible to create and utilize complex index and pointer structures that optimize access to related data items in storage.

3. Using filtering and selective display of map data is the most effective means of improving performance as it reduces the number of objects involved in data retrieval, coordinate calculation, and drawing. It also has the desirable side effect of reducing screen clutter.

4. Finally, a note on the performance from the point of view of the user: the design of the user interface on a mobile device is of much greater importance than on a desktop, as the user interaction capabilities of mobile computing devices are much more limited. Neglecting this aspect during design may result in an application that will be hard to use and may be ultimately rejected by its users, even when the application’s response time is good. Careful UI design should therefore accompany design of the application’s internals.

5. Conclusions

This paper has outlined the development of the MacauMap mobile map application. Since its release to the public in May 2003, it has proven highly popular with both tourists and local residents alike. We take this as confirmation that our efforts at achieving acceptable performance levels in MacauMap have been successful. We are continuing our work on other smart phone platforms, including an upcoming version for Symbian OS UIQ as well as for Jave/J2ME, and have also developed a web-based version and a version for deployment in information kiosks, to complement our line of mobile versions.

Acknowledgments

The work reported on in this paper was supported by the Research Committee of the University of Macau, and by the Macau Government Tourist Office. Map data was provided by the Cartography and Cadastre Bureau, and bus route data was provided by the Lands, Public Works, and Transportation Bureau, both of the Macau S.A.R. Government. Moreover, tourism-related information was provided by the Macau Government Tourist Office. The support from all of these organizations is gratefully acknowledged. Special thanks are due to the MacauMap development team: Kass Ian, José Choi, Victor Lam, Samuel Chiu, Anthony Lam, Henry Lam, Angelina Chow, Rebecca Lam, Leslie Lei, David Cheong, Henry Leong, Acacio Lo, and Kevin Yung whose hard work and determination have helped make MacauMap a reality. MacauMap is a registered trademark in the Macau S.A.R.

References