NEW INTELLIGENT METHOD TO GENERATE VECTOR MAPS FOR GPS NAVIGATION

Sheng-Guo Wang [1] and Yuanlu Bao [2]

[1] *Dept. of ET and Dept of ECE, University of North Carolina at Charlotte, NC 28223, USA.* *swang@uncc.edu*

[2] *Dept. of Automation, University of Science and Technology of China, Hefei, Anhui 230026, China. ybao@ustc.edu.cn*

Abstract: This paper introduces a new intelligent method to generate vector maps from printed maps to provide a basis for GPS navigation. This new approach has been applied in our GIS-DMG (Digital Map Generation) software, which has four key steps: Map Layers Extraction, Road Network Digitalization, Map Geo-locating and Map Automatic Geo-adjusting, different from typical procedures. These four steps play essential key roles for developing city vector map of GIS. The new intelligent method is effective and less time-cost. The produced city or highway network maps have been well used for GPS applications, e.g., vehicle navigator or intelligent transport system (ITS). *Copyright © 2002 IFAC*

Keywords: GPS (Global positioning system), Information systems, Navigation, Intelligent systems, Transport, Pattern recognition, Vector map, Digital map.

1. INTRODUCTION

It is well known that land transportation needs geographical or traffic information such as intersection locations and street traffic flow direction. During the last decade, a kind of intelligent transport system with new information technology has been proposed (Ekblad and Eriksson, 1996) in order to improve transportation conditions.

With aids of new technologies such as Global Positioning System (GPS) (Leick, 1995), GIS (MapInfo, 1997, 1999), artificial intelligence database and multimedia systems, the ITS provides full information with a multi-layer digital city map for drivers. The information includes streets, important institute locations and phone codes, as well as the real-time position, speed and direction of his vehicle on the map. When a destination is given, an Intelligent Vehicle Navigation System (IVNS) can provide several ways leading to the destination and the shortest route. For example, an IVNS developed at the USTC (Bao, *et al.*, 2001; Bao 1998) is a portable in-vehicle navigation system for application. It can also monitor the traffic for selected vehicles driving in the city as shown in Fig. 1 .

However, a good vector map is a basis for any GPS-IVNS navigation and monitoring (Bao and Wang, 2001). This paper focuses on introducing a new intelligent method to generate a GIS vector map for GPS-IVNS applications. The vector map is very efficient in memory size, information searching, updating and accuracy in scale. . Only its nodes need to be saved for lines in our IVNS. Furthermore, it is easy to modify, edit and work with, such as zoom in or out.

To generate a digital map from a printed paper map, a typical method (MapInfo, 1997) uses four typical steps: Mapping, Routing, Geo-coding and Data-adding. In the first two steps, a digital instrument (digitizer) and manual interposition are needed to get the information of the map layers and road network through manual input to a computer. This procedure for a map digitalization costs much time and labor, as well as human errors perhaps. In the third step, geo-coding, the digital map with its every point location (latitude/longitude) may be tuned through only linear transformations with the whole area (rotation and horizontal or vertical ratio).

This paper introduces a new intelligent method to generate a vector map. Its procedure includes four different steps: Map layers extraction, Road network digitalization, Map geo-locating and Automatic map geo-adjusting (adaptive optimal map adjusting). The main feature of this new method can be briefly described as follows. The new method uses a scanner to input a paper map to a computer. Then, it uses an intelligent pattern recognition method to extract the pixel map layers and the road network automatically. After that, the road network vectorization is done automatically as well. Via the map geo-locating, the
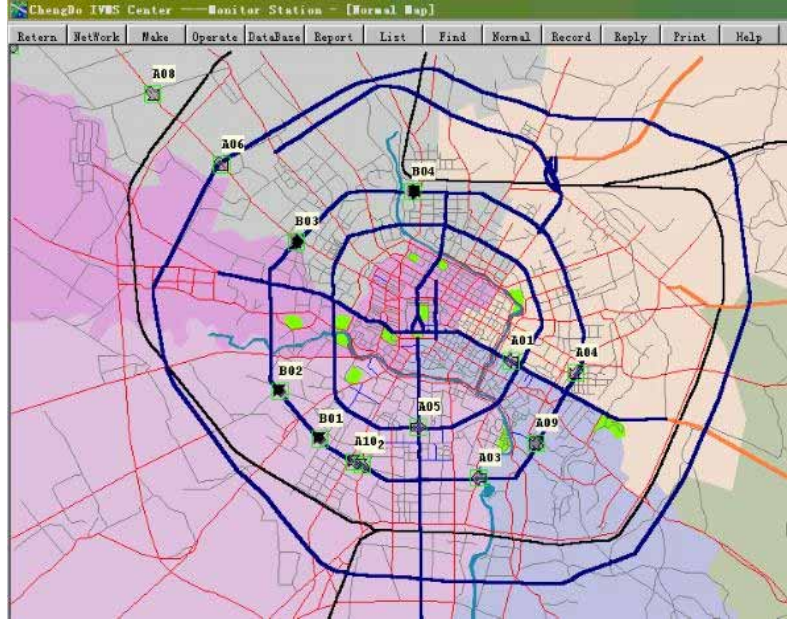
Fig. 1. An IVNS is monitoring vehicles

new method allows the vector map with its every point location (latitude/longitude) to be tuned through both linear transformations and some nonlinear transformations which are useful for nonlinear map adjusting in order to get accurate point locations with the whole area. Furthermore, as used in the GPS vehicle navigation, the recorded new road locus can be used to improve the map accuracy in the fourth step — adaptive optimal map adjusting. The consistency and compatibility with other types of spatial database is also considered. Thus, at last, the generated vector map database can be linked to other open descriptive information database to exchange geographical data between each other. Here the method deals with city maps that are "small" areas in the earth and the original printed maps contain their implicit relative coordinates that will be mapped in step 3 and adjusted in step 4.

The above four key steps play essential roles in developing a city vector map GIS by computer-aided producing and adjusting. It has been applied to our GIS-DMG (Digital Map Generation) software. The generated city maps or highway network maps are well used for GPS applications such as the vehicle navigator or intelligent transport system (ITS).

This paper mainly describes this new intelligent method to generate a vector map. The paper is organized as follows. Section 2 discusses the map layers extraction. Section 3 addresses the road network digitalization. Section 4 describes the map geo-locating. Then, automatic map adjusting and optimizing are discussed in Section 5. Finally, Section 6 concludes this paper. Due to the page limit, only brief description is presented.

2. MAP LAYERS EXTRACTION

This section describes the first step – the map layers extraction as follows. A scanner scans a color map and inputs the scanned information into a computer. Then the computer generates a pixel map displayed on the screen. Select a pattern, such as roads, squares, railways, then the computer can detect all similar pattern parts from the pixel map by using a pattern recognition method. It provides a basis for the map layers extraction.

In general, four colors can separate any different areas in a map. Thus, a map usually has four or more colors, e.g., brown for contour lines, blue for water areas, green for forests, and so on. Some elements may have a same color, e.g., longitudes, latitudes, railways, region names and buildings are marked in black. People may recognize a physiognomy by its color from the map when they perceive the map first. The vector map generating system does it in a similar way as people do. Due to different quality of the original map, the pixel map may have "noise" or error. Thus, a suitable way is needed to let the computer be able to filter "noises" and make a reasonable correction. In the following, we describe the method for the color layer first.

2.1. Method for Color Layer

Originally, a bitmap is represented by a set of pixels

$$P = [d_{11}, d_{12}, ..., d_{mn}] \qquad (1)$$

where d_{ij} is a pixel at position (i, j) in the pixel map, m and n are the maximum numbers of pixels in rows and columns respectively. Define the color of pixel d_{ij} as $C(d_{ij})$ which is a 3-dimensional vector with three RGB (red, green, blue) grades:

$$C(d_{ij}) = [R(d_{ij}), G(d_{ij}), B(d_{ij})] \qquad (2)$$

Select some sample areas $\{S_{k_v}\}$ in the bitmap by using a mouse. Then, designate a set of samples as a swatch $S_k = [S_{k_1}, S_{k_2}, ..., S_{k_v}, \cdots, S_{k_t}]$ to present a special layer $\bar{S}_k$. Suppose S_{k_v} contains l pixels $\{d_{ij}\}$. Let a sample point s_{k_v} with the average

color of S_{k_v} represent this sample color class, $v = 1,...,t,$ respectively,

$$C(s_{k_v}) = \tfrac{1}{l} \sum_{i,j} C(d_{ij}) \qquad (3)$$

Define $D(d_{ij}, s_{k_v})$ as a function of the color difference between the pixels d_{ij} and s_{k_v}:

$$D(d_{ij}, s_{k_v}) = \sqrt{[R(d_{ij}) - R(s_{k_v})]^2 + [G(d_{ij}) - G(s_{k_v})]^2 + [B(d_{ij}) - B(s_{k_v})]^2} \qquad (4)$$

Compare all the color differences between every pixel d_{ij} on the map and every special sample S_{k_v} in the swatch S_k with a threshold V_k. If

$$D(d_{ij}, s_{k_v}) \le V_k \qquad (5)$$

then this pixel d_{ij} belongs to this layer $\overline{S}_k$. Thus a special layer $\overline{S}_k$ includes the entire selected pixels within this layer $\overline{S}_k = [\overline{S}_{k_1}, \overline{S}_{k_2},..., \overline{S}_{k_t}]$. Notice that the algorithm is trained by a piece of sample map for a type of city maps. After the training and selection of the thresholds for color classifications, the algorithm with this selection in the Euclidean distance Eq. (4) is automatically suitable for all other city maps that belong to the same publisher style.

When the original map is old and not clear, smooth clustering is needed. Two different threshold values V_{k1} and V_{k2} ($V_{k1} < V_{k2}$) are specified for the layer $\overline{S}_k$ with the special colors $C(s_{k_v})$. First, use V_{k1} for an initial extraction from the original bit-map, then use V_{k2} for a tuned extraction from a bit-map with the initial extraction clustering and its neighborhood. This method will smooth the extracted

clustering and provide a high quality extraction.

2.2 Implementation

Here, the purpose is to generate a vector map for the GPS navigation. Therefore, the map is divided into only three layers: a road network layer, an area division layer and a name layer including main region names and building icons.

The name layer should be "eliminated" as disturbances to prepare digitalization. It is extracted and eliminated from the original bit-map, then replaced by the color of most close pixels to the corresponding eliminated pixels of this layer to make a pretreated bitmap.
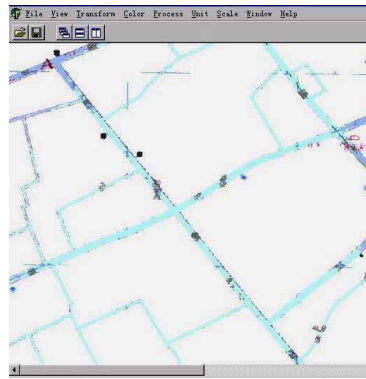
To extract the road network layer and to extract the area division layer are processed separately, but both use the pretreated bitmap as their workspace. Then, the initial clustering and smoothed clustering are used in view of the original map quality.

For each color block, select a sample s_{k_i} in the center of the block, draw a boundary for the block, and let all the pixels within this boundary to have this swatch color.
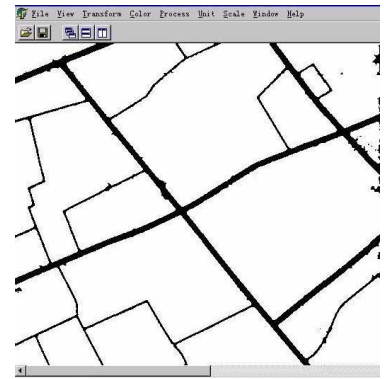
After that, calculate the color differences $D(d_{ij}, S_{k_v})$ at pixels for the extracting road network layer and the area division layer. For the different colors, the different threshold values may be selected. Generally, the threshold valve for the block elements should be selected small, while the one for the line elements may be selected relatively large. An example of extracting a road layer is shown in Fig. 2.

Fig.2. a) Original bitmap   b) Pretreated bitmap   c) Road extracted bitmap

3. ROAD NETWORK VECTORIZATION

Vectorization is to convert a black-white bitmap of the line-shape figures into a vector format graph as follows.

3.1. Pretreatment of the Extracted Road Network

This pretreatment procedure of the extracted road network includes the following processes.

Smoothing. Because of the noises and errors, there are some indents and bulges in the line-shape features. Removing these indents and bulges before

further process can improve the precision of vectorization.

We use a pattern match method to fill the indents and erase the bulges. By using 1 to represent the object points (black), and 0 to represent the background points (white), indent and bulges patterns can be obtained. Then, change 0 to 1 in indent patterns to fill all indents and change 1 to 0 in the center of bulge patterns to erase all bulges.

Thinning. Thinning is a process to erase the points that do not affect the connection of the line-shaped

figures for vectorization.

Joining broken lines. This process is to fix the errors in the original map or ones that occur during the above steps. A terminate *node* is defined as a point that has only one neighbor object point. Then, compute the distance from this terminate node to its neighbor terminate node. If it is less than a designated value, join these two nodes with a line of object points.

Deleting spurs. After joining the broken lines, find all new terminate nodes. Then, delete spurs by deleting lines which is connected to terminate nodes and is shorter than a specified value.

The above-mentioned processes can significantly remove errors which occur during the previous extraction procedure.

3.2. Road Network Vectorization

After the above procedures, a bitmap of single-point-wide lines is ready. Now, the next process is the vectorization (digitalization). First, establish a database structure for storing the map data easy to read.

Digital map database structure. When one investigates many types of city maps, it is not difficult to classify information into three divisions: points, lines, and areas. Each division is divided into several groups, whose member entities denote geographical information, and are shown at the correct coordinates on the map. In an object-oriented method, all three kinds of information can be abstracted as a base class which has their common properties and operating behavior. The main logic structure of the GIS with the GPS is shown in Fig. 3.
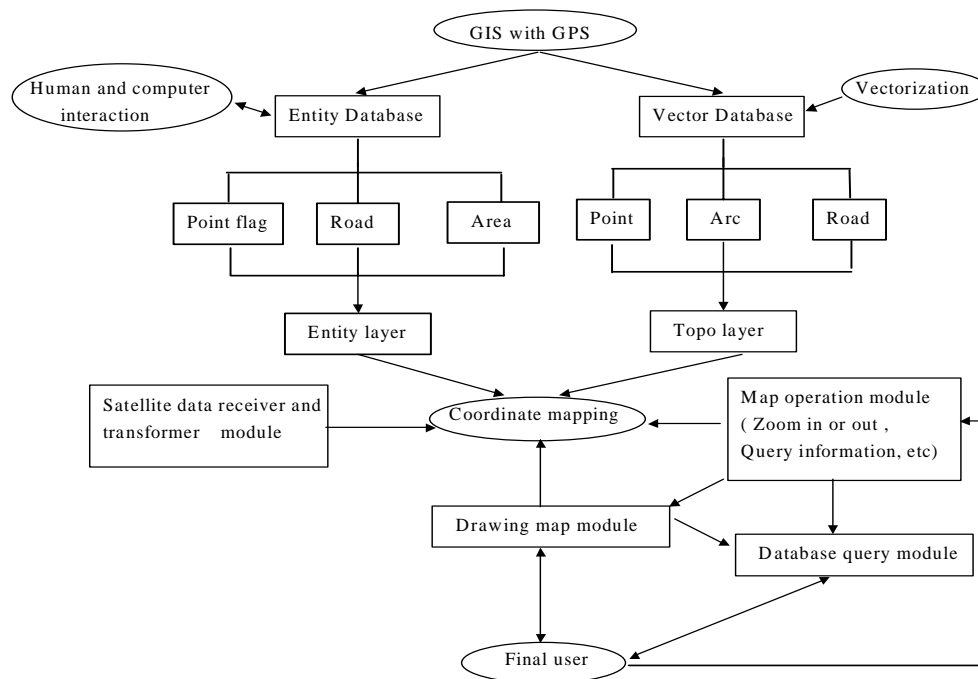

Fig. 3. Logic structure for GIS with GPS

Road network vectorization. The digitalization algorithm is described below. Define a node as a point whose number of neighbor object points is not two. Then, start tracing the object points from nodes for digitalization. Search the object points from a node until it reaches to another node, and mark these found points so that they can be skipped next time. These points form a curve usually. Then use a series of lines to approach the curve. A recursive approaching-algorithm determines errors brought by digitalization. To determine this series of lines for approaching the curve between nodes P_1 and P_2 on the map, we first find on the curve a point P that has the maximum distance D_P to the line connecting P_1 and P_2. If D_P is less than a pre-specified maximum error E_{max}, this approaching is completed. Otherwise, add P as a new node and list the curve between P_1 and P, and another between P and P_2, respectively.

Then, search a new potential point P' following the same procedure for finding the point P with respect to the new node family of nodes P_1, P and P_2. The algorithm is certainly convergent because D_P approaches to zero when the number of family nodes increases. An example is shown in Fig. 4.
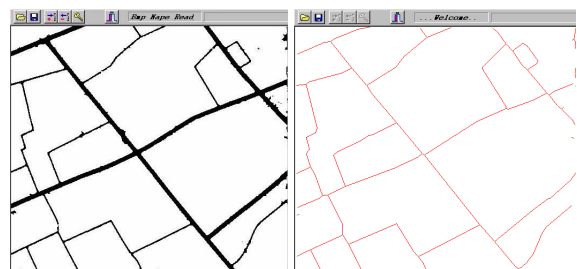

Fig. 4. Road network digitalization: a) Road extracted bitmap; b) Vector/digital map

4. MAP GEO-LOCATING

4.1. Map Layers Geo-locating

Suppose a point d has its coordinate (x, y) on the map plane S-map, and its longitude and latitude (lon, lat) on the plane S-earth. Select a base point d_0 (x_0, y_0) on the S-map with its longitude and latitude (lon_0, lat_0) on the S-earth. Select the map scales in two axis directions as S_x and S_y, respectively, and rotation angle θ (anticlockwise from the S-earth to the S-map). Then, we have

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} S_x(lon - lon_0) \\ S_y(lat - lat_0) \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (6)$$

$$= \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \begin{bmatrix} lon \\ lat \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \quad (7)$$

where $\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$ (8)

$$\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} -S_x lon_0 \\ -S_y lat_0 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (9)$$

In (7), a_i, b_i, c_i are independent of (x, y, lon, lat). By using the data analysis method, parameters a_1 through c_2 can be calculated by several groups of data (x, y, lon, lat). To adjust the scales and rotation angle, some locus is recorded by the GPS running on the city roads. Then, force the corresponding roads on the numerical vector map to coincide with the recorded data as described in the next section.

An example is shown in Fig. 5, where point $(x_0, y_0) =$ (1038, 1089) and its longitude and latitude (lon_0, lat_0) = (2.0463079, 0.5557810) or (117°14.688′, 31°50.634′). The rotating angle $\theta = 10°$.

Fig. 5. An example for geo-locating: a) S-map plane; b) S-earth plane

4.2. GIS Database Adding

The Database Adding is a human-computer interaction procedure. Users input some information such as its point flag, mark with the name and icon, and area with the color and name by the mouse and keyboard. Then the computer renews its database and the screen displays this new drawing map.

After these steps, a printed map has been converted into a digital/vector map with the complete geographical information. If necessary, the GIS can be linked with other data analysis components and multiple data-types such as the animation, grid, photo, and video. Intelligence is one of its characteristics, e.g., fuzzy query, linking with some SQL and dBase.
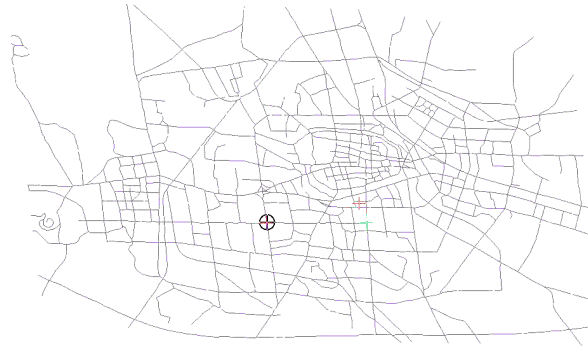
5. AUTOMATIC MAP GEO-ADJUSTING

In the GIS system with the GPS for applications such as the driver navigation or vehicular monitoring, the road division is the most important one, whose precision directly affects whether a car position is displayed correctly on the digital map or not. Thus, the automatic map geo-adjusting for the map accuracy is introduced. The method is to use the recorded road locus by the GPS to automatically adjust the vector traffic map by our GIS-DMG.

5.1. Automatic Optimization of Road Geo-adjusting

Suppose a pixel P with its point location coordinate (x, y) on an unadjusted vector map plane S-map with its Cartesian [rectangular] coordinates P[X,Y]. Its corresponding correct location has the navigation coordinate of longitude and latitude (lon, lat) on the S-earth with its geographical coordinates G[lon, lat]. The process of automatic optimization of road network geo-adjusting is described as follows. Select a group of points, denoted as

$$P[p_1, p_2, \cdots, p_v] = P[(x_1, y_1), (x_2, y_2), \cdots, (x_v, y_v)]$$
$$\text{on S-map } P[X, Y]. \quad (10)$$

The corresponding correct location should be

$$G(p_1, p_2, \cdots, p_v) = G[g_1, g_2, \cdots, g_v]$$
$$= G[(lon_1, lat_1), (lon_2, lat_2), \cdots, (lon_v, lat_v)]$$
$$\text{on S-earth } G[Lon, Lat]. \quad (11)$$

The problem becomes to find a mapping $M(\cdot)$ from points p_i to g_i, $i = 1,2, \ldots v$. The mapping $M(\cdot)$ should be a composition mapping and include only

two kinds of mappings: a linear mapping $M_L(\cdot)$ for expansion, contraction and movement, and a conformal mapping $M_R(\cdot)$ for rotation. Denote

$$M(\cdot) = M_R \cdot M_L(\cdot) \qquad (12)$$

where

$$M_L(\cdot): \quad X' = K_x X + m_x, \quad Y' = K_y Y + m_y \qquad (13)$$

$$M_R(\cdot): \quad \theta^* = \theta' + r_\theta, \quad \rho^* = K_\rho \rho' \qquad (14)$$

Because all mappings include moving factors, the pole in $M_R(\cdot)$ is selected as the superposition of the origin of that in $M_L(\cdot)$ and have

$$X = \rho \cos(\theta), \quad Y = \rho \sin(\theta) \qquad (15)$$

Through minimizing optimal performance index

$$\min J = \sum_i [(lon_i - M(x_i))^2 + (lat_i - M(y_i))^2] \qquad (16)$$

the following six mapping parameters, in equations (13) and (14), can be obtained for automatic optimal geo-adjusting

$$V = [K_x, m_x, K_y, m_y, K_\rho, r_\theta] \qquad (17)$$

Thus, an optimal mapping $M(\cdot)$ for the optimal geo-adjusting is obtained. The interesting research is to include new algorithm development to let the errors in step Map Automatic Geo-adjusting approach to zero.

5.2. An Example for Automatic Road Geo-adjusting

During the optimization process, not only the selected points move to their corresponding correct locations, but also all the other points on the vector map move to their more precise corresponding locations (longitude and latitude) based on the real track record from the GPS receiver as shown in Fig. 6.
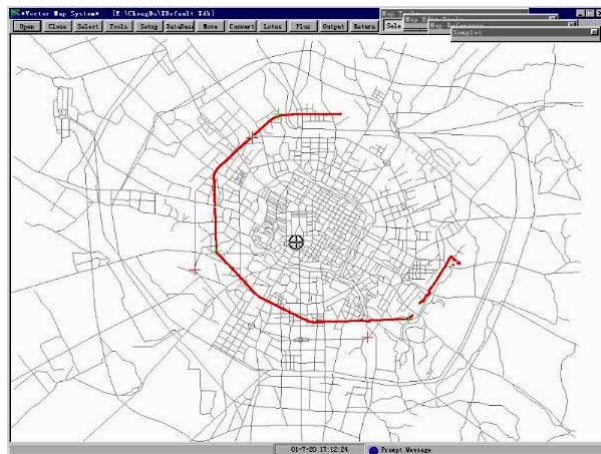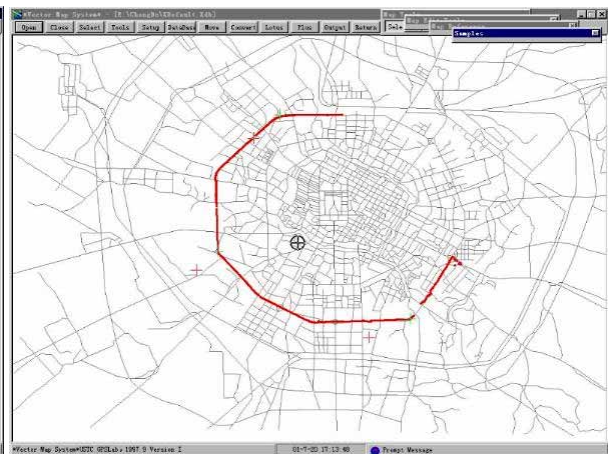

Fig.6.　a) Using a road locus to adjust a digital map　　　　b) Adjusted vector map after geo-adjusting

6. CONCLUSION

In this paper, a new intelligent method to generate a vector map from a printed map for the GPS intelligent navigation system has been introduced. It includes four key steps: Map Layers Extraction, Road Network Digitalization, Map Geo-locating and Map Automatic Geo-adjusting, different from the current typical steps. Based on this development, the GIS-DMG software has been implemented. After trained by several model maps, the GIS-DMG performs very well on other city traffic maps published by the same publisher. The whole process of vector map generation may also include other procedures such as pre-dealing, vector maps joining, automatic locus identification, etc. The produced vector map can be used for both vehicle navigation system and vehicle monitoring system.

ACKNOWLEDGMENT

REFERENCES

Bao, Y. (1998). The overall design on GPS vehicle monitoring system, *World manual for purchasing GPS receiver*, China Aviation Industry Press, pp.169-183.

Bao, Y. and S.-G. Wang (2001). USTC GPS Intelligent Vehicle Navigation System. *Proceedings of 2001 International Symposium on Adaptive and Intelligent Systems and Control*, Session VI-1, Charlottesville, VA.

Bao, Y., B. Xia and Y. Bao (2001), The pivotal technology for developing GPS vehicle monitoring system. *China Highway — Transportation Information Industry*, **2**, no.2, pp.2-4.

Ekblad, J. and N.E. Eriksson (1996). Hailing GPS: A Swedish Taxi Security System. *GPS World*, **5**, 32-36.

Leick, A. (1995). *GPS Satellite Surveying*, 2nd Ed., John Wiley & Sons, New York.

MapInfo Corporation (1997). *Operator Manual of MapInfo Professional*, Version 4.1.

MapInfo Corporation (1999). *MapInfo Professional 5.5 User's Guide.* http://www.mapinfo.com.