MARIS: MAP RECOGNIZING INPUT SYSTEM

- Implementation and performance on a special workstation -

Satoshi Suzuki and Toyomichi Yamada

NTT Human Interface Laboratories
1-2356, Take, Yokosuka 238-03, Japan

Abstract

We have proposed an automatic map recognizing input system called MARIS. This system digitizes large-scale maps into a layered data form in order to construct a map database. This paper presents the system's implementation and performance on an experimental workstation. Experimental results show that the input time using the MARIS system can be reduced to about 25 % of that of a system using an interactive digitizer.

1. Introduction

Interactive digitizers are used as input units in conventional geographical information systems because of their high data compression rate, low hardware cost and operational simplicity [1]. However, considerable time and cost are necessary to digitize geographical data because of the number of manual operations involved in tracing. These manual operations can be reduced by scanner systems which are suitable for the automatic digitization of maps or aerial photographs. Digitization should produce information in a layered data form, because this is needed for easy display, intelligent retrieval and efficient storage [2]. However, this has been considered difficult and different input techniques are essential to overcome this problems.

We have already proposed an automatic map recognizing input system called MARIS [3]. This system digitizes large-scale maps into a layered data form in order to construct a map database. This paper presents the implementation of the MARIS system and assesses its performance on large-scale national maps of Japan.

The MARIS system is implemented on an experimental workstation which consists of a main control unit and a graphic processing unit. The main control unit is the same type of workstation as used for distributed processing of multi-media data [4]. The graphic processing unit has multi-processors, large-capacity window map memory, magnetic disk storage for enhancement of large-scale image processing speed.

The MARIS process consists of four stages: preprocessing, vectorization, automatic recognition, and interactive correction. Preprocessing and vectorization are performed in the graphic processing unit. The multi-processors contribute to the enhancement of vectorization speed. Automatic recognition is performed on the main control unit. In this stage, building lines, contour lines, and lines representing railways, roads and water areas are extracted. Railway extraction is omitted in this implementation. Recognition algorithms based on vector data enable high-speed recognition. The recognition results are corrected interactively through the main control unit. MARIS's interactive input times are shorter than those using conventional interaction methods, because the number of manual operations are reduced by utilizing basic line tracking algorithms.

2. Experimental Workstation

The MARIS is implemented on an experimental workstation system. The system consists of a scanner, a workstation (called WS-R), magnetic tape storage, and an electrostatic plotter as schematically shown in Fig. 1. Equipment specifications are shown in Table 1. The experimental workstation consists of two units: the main control unit and the graphic processing unit. The main control unit operates under UNIX System V. This unit is the same type of workstation as used for distributed processing of multimedia data [4]. The main control unit is composed of a Motorola 68020 processor, an 8 Mega-byte main memory, 320 Mega-byte magnetic hard disk storage, and a local area network controller.

The graphic processing unit consists of a graphic engine, 32 Mega-byte window map memory, a CRT, frame memory, I/O interfaces to a large-scale scanner and a plotter, and 80 Mega-byte magnetic disk storage for image data.

The graphic engine is composed of a Motorola 68020 processor and four data flow pipeline processors [5]. The pipeline processors are programmable and enhance the speed of image processing operations. If an image processing operation is simple and is suitable for parallel processing, pipeline processors enable high-speed image processing. Otherwise, the Motorola 68020 processor is used. The MARIS system performs thinning operation, feature point extraction, rotation, etc., by the pipeline processors.

The graphic processing unit has large capacity window map memory. The capacity is enough to store a complete digital binary image of an input map sheet. Because this memory is used as working memory in vectorization and interactive correction, the image data is stored on the 80 Mega-byte magnetic disk. This capacity is sufficient for two map sheets.

3. Implementation of MARIS

The MARIS process consists of the four stages : preprocessing, vectorization, automatic recognition, and interactive correction. Preprocessing and vectorization are performed on the graphic processing unit. This unit also carries out image input processing and image output processing including display processing. Most of the automatic recognition and interactive correction are performed by the main control unit.

3.1 Preprocessing

A large flat scanner initially converts a map sheet into a binary image directly by grid sampling at a resolution of 16 pixels per mm. After input, the binary image is rotated 90 degrees, and stored in the window map memory. The pipeline processors directly contribute to increase in the rotation speed. The picture is divided into 70 subpictures of 2048 pixels x 1024 pixels and stored on the magnetic disks. These subpictures are processed one by one as described below.

3.2 Vectorization

The vectorization stage consists of labeling by line widths, thinning, transformation of an image into a graph, edge deletion, and straight line approximation. First, a subpicture is transferred from the magnetic disk into the window map memory. Then, all 1-pixels in the subpicture are labeled according to the width of the lines passing through them [6]. Next, the labeled subpicture is converted into an 8-connected medial line image by thinning operation [7]. Then, the medial line image is transformed into a graph, in which pixels on the medial line correspond to nodes and neighboring nodes are connected by edges. Next, extra edges, unnecessary for preserving the topology of the medial line image, are deleted to form a simplified graph [8]. Next, vector data is produced from the simplified graph by extracting feature points and approximating curved lines between two feature points as straight line segments. The vector data is transferred to the main memory of the main control unit. Fig. 2 shows an example of the vectorization process.

The pipeline processors enhance the speed of the thinning operation and the feature extraction. For example, a thinning operation using the pipeline processors runs 20 times faster than when using a general-purpose VAX 11/780 computer. Width labeling, transformation of an image into a graph and edge deletion could be performed in parallel, but instead they are performed on the graphic engine CPU. This is because the program area of the pipeline processors is too small to store their programs.

Extracted vector data are composed of feature points, branches and segments as shown in Fig. 3. A curve between the two feature points is called a branch. Line segments produced by a straight line approximation of a branch are referred to as simply segments.

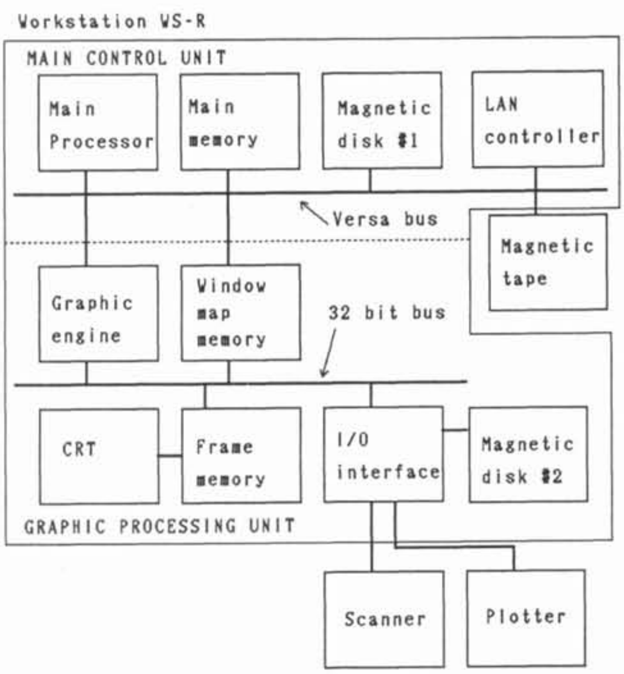

Fig. 1 Block diagram of MARIS

Table 1 MARIS specifications.

MAIN CONTROL UNIT	
Main processor	MC68020 OS : UNIX System V
Main memory	8 MB
Magnetic disk 1	320 MB
LAN controller	CSMA/CD method 10 Mbit/sec
GRAPHIC PROCESSING UNIT	
Graphic engine	MC68020 + Pipe line processors
Window map memory	32 MB
Frame memory	256 kB x 6
Magnetic disk 2	80 MB
CRT	1280 x 1024 dot, 20" color
PERIPHERALS	
Magnetic tape	1600 BPI, 2400 ft.
Scanner	Width : 620 mm, 16 dot/mm
Plotter	width : 841 mm, 16 dot/mm

Extracted vector data are more compact and more natural than those obtained by conventional methods. In particular, the amounts of feature points, branches and segments are reduced significantly. Since a line tracking technique based on segments [3] and a border tracing technique based on branches [3] are essential functions for our automatic recognition method, the reduction of vector data quantity helps significantly to speed up the automatic recognition process.

The produced vector data are hierarchical. That is, feature points and branches are a high level representation of the line patterns. On the other hand, feature points, branches and segments are a lower level representation of the line pattern. The hierarchical vector data are useful to trace borders between the medial lines and the background quickly. That is, the border tracing algorithm [3] performs tracing by using only the feature table and the branch table.

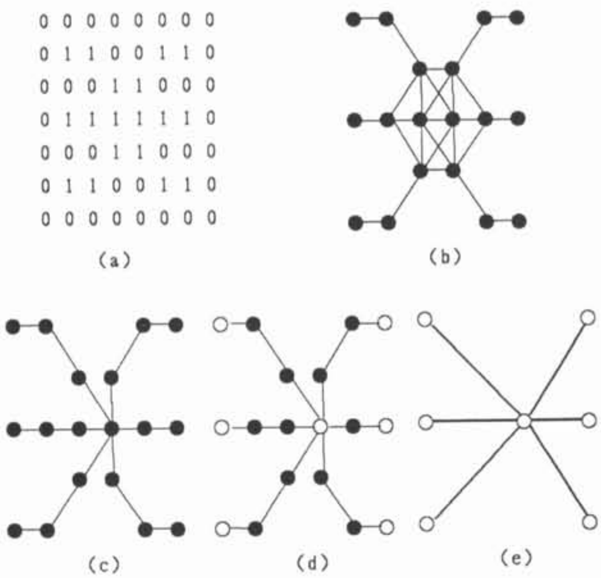

Fig. 2 Illustration of vectorization process. (a) Medial line image. (b) Graph for (a). (c) Result of edge deletion. (d) Result of feature extraction. (e) Result of line approximation. (Black circles denote nodes. White circles denote feature points among nodes. Thin solid line segments denote edges. Thick solid line segments denote vectors.)

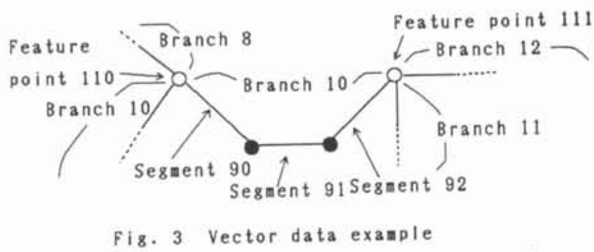

Fig. 3 Vector data example
(● stands for nodes of segments.)

Therefore, the algorithm is faster than those based on only segments or pixel.

3.3 Automatic Recognition

The recognition stage extracts three layers : buildings; contour lines; and railways, roads and water areas. Each layer contains lines representing these map components. Extraction is performed by the main control unit by analyzing only the hierarchical vector data obtained in the vectorization stage. The recognition algorithms based on the vector data enable high-speed recognition. Since the details of these algorithms are described in reference [3], only the outline of these algorithms is presented below.

The automatic recognition stage is comprised of the four sequential substages: building extraction, building deletion, extraction of lines representing railways, contour lines, roads and water areas, and contour line extraction.

Building extraction : A building is represented by its outline. In large-scale Japanese maps, buildings are represented by closed borders that have two line thicknesses. To duplicate the effect of sunlight, the building outline exposed from 45 degrees clockwise from the maps horizontal axis is drawn with a line thickness of 0.1 mm. Other outlines are 0.3 mm thick. The technique for extracting buildings utilizes this characteristic. First, as shown in Fig. 4(a), our technique searches for a sub-border *B1* having a width greater than or equal to 0.3 mm. Then, our technique extracts a node *N1* (Fig. 4(b)) of the subborder which is located at a right-angle corner, and its left or upper neighbor is a 0-pixel located within the building outline. Next, a closed hole border or unclosed subborder *B2* (Fig.4(c)) surrounding the right angle corner is extracted by the border tracing algorithm. Then, the features of border length, inner area and dispersedness [9] of the border or subborder are calculated. Finally, the border or subborder is examined to determine whether or not it is a single building line. The examination is based on these features.

Building deletion : This substage first identifies those building lines coincident with road lines and then deletes all other building lines. This creates the input for the next substage. First, all branch-

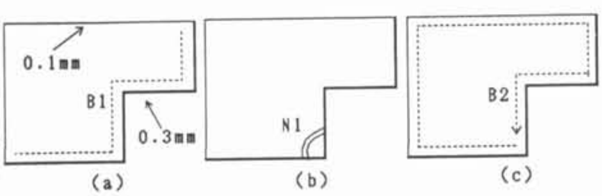

Fig. 4 Building extraction process.
(a) Extraction of sub-border having width greater than or equal to 0.3 mm. (b) Extraction of right-angle corner open upward or leftward. (c) Extraction of closed hole border surrounding the right-angle.

ing points on a current building line are identified as shown in Fig. 5(a). External segments connected to the building branching points are located. The external segments are extended across the branching points as shown in Fig. 5(b). If they are continuous with the building segments and do not stop at the building's outline, the building segments are marked for retention. Similarly, the above operations are repeated for every building line. After that, building segments except those marked for retention are deleted as shown in Fig. 5(c).

Railway, contour line, road, and water area extraction: Railways, contour lines, roads and water areas are composed of long lines with small curvatures and constant line widths. Therefore, these lines are extracted directly by using the line tracking algorithm [3].

Contour line extraction: This substage extracts contour lines from the continuous lines obtained by the previous substage. Contour lines are classified into two types having two different line widths: 2 m contour lines are 0.1 mm wide and 10 m contour lines are 0.2 mm wide. Normally, these two type contour lines are periodic, namely four 2 m contour lines lie between two 10 m contour lines. Therefore, contour lines can be recognized by these properties. First, continuous lines with a width of 0.1 mm are regarded as 2 m contour line candidates, and continuous lines with a width of 0.2 mm as 10 m contour line candidates. Then, these contour line candidates are examined to calculate the probability of whether they satisfy the periodicity given above. Finally, candidates having a probability greater than a given threshold are recognized appropriately as 2 m and 10 m contour lines.

3.4 Interactive Correction

Recognition errors are quickly corrected and unrecognized layers are input interactively using the main control unit combined with the graphic processing unit. Each layer is corrected or input as shown in Fig. 6. The interactive processing time using MARIS is shorter than that using conventional interaction because of the following intelligent functions.

The detection of recognition errors is relatively easy. The image data from an input map and the vector data of its medial lines are superimposed on the CRT. From the vector data, the medial lines grouped as one layer can be displayed with a unique color. Unrecognized medial lines are also displayed with a unique color. This color display aids an operator in locating recognition errors quickly.

Picking of a target line is easy and fast. If an operator points to a spot on the CRT, MARIS selects the segment which is nearest to the spot and which belongs to the target layer. Therefore, if the vector data is in many layers, the operator can point some distance from the target line and the picking time can be shortened. Note that unrecognized medial lines are regarded as one layer.

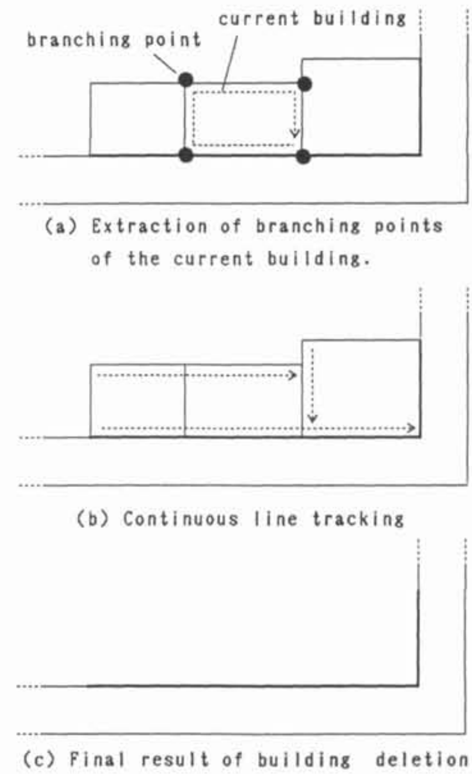

Fig. 5 Building deletion process

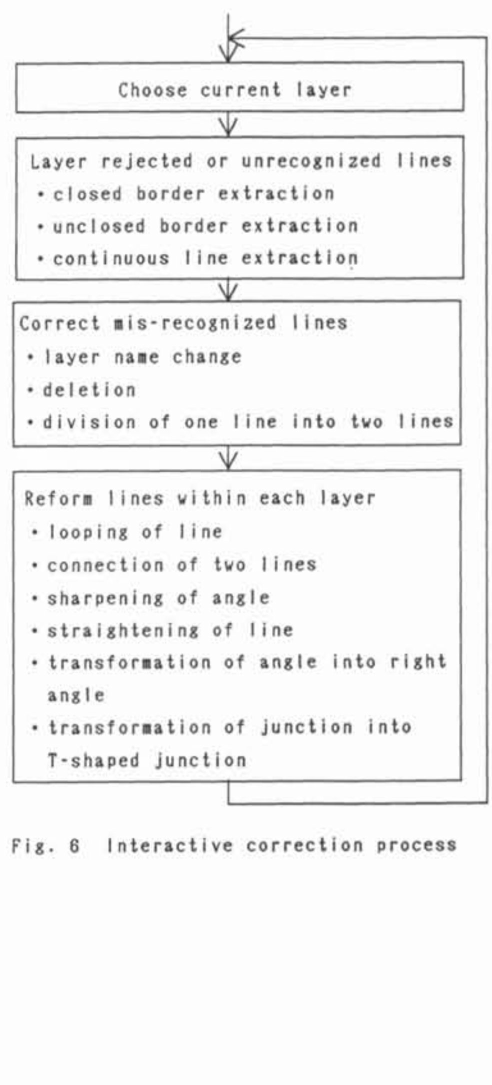

Fig. 6 Interactive correction process

The number of picking operations is reduced. In order to extract a line rejected in the recognition stage or an unrecognized line, an operator can use the border tracing algorithm or the line tracking algorithm by nominating just a start point and/or an end point of the target line. Fig. 7 shows examples of the extraction of the rejected lines and the unrecognized lines.

Fig. 7(a) shows remedial extraction of an unclosed border of a building. First, an operator picks a spot within the building *Q1*. MARIS selects the segment which is nearest to spot *Q1* and which has not yet been recognized. Next, the system obtains branch *B2* containing the segment. Then, the counterclockwise border tracing algorithm extracts the string of *B1* and *B2*. Next, the clockwise border tracing algorithm extracts the string of *B2* and *B3*. Finally branches *B1*, *B2*, and *B3* are highlighted. If the operator decides that the result is a building, these branches will be displayed with the color of the building layer. The unclosed outline will be closed by the reforming function of MARIS.

Fig. 7(b) shows the extraction of a road line. First, an operator picks spot *Q2*. MARIS selects the segment which is nearest to spot *Q2* and which has not yet been recognized. Next, the system obtains branch *B5* containing the segment. Then, the continuous line tracking algorithm extracts the string of *B5*, *B6* and *B7*. Next, the continuous line tracking algorithm extracts the string of *B5* and *B4*. Finally branches *B4* through *B7* are highlighted. If the operator decides that the highlighted branches are a road line, these branches will be displayed with the color of the road layer.

The correction of unknown information is easy. In some cases, the information whether or not a line is closed and whether or not two lines are connected is unknown. MARIS has an interactive command for corrective inspection. In the case of the building layer, MARIS searches for unclosed lines which have already been recognized as buildings and prompts the operator for the correction of the lines one by one. In the case of the other layers, MARIS searches for neighboring line pairs which have already been recognized as belonging to the same layer and asks the operator whether or not these two lines are to be connected. Therefore, the correction of unknown information is easy.

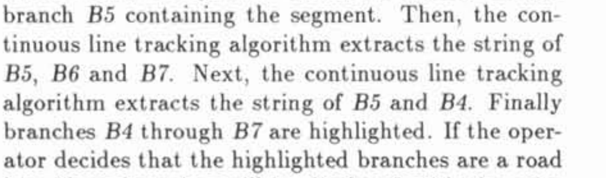

Fig. 7 Interactive rejected line extraction

MARIS's interactive operations can handle the vector data recognized line by recognized line, branch by branch, or, segment by segment. In our experience of using MARIS's interactive operation, the branch by branch operations such as deletion of branch and change of branch layer etc., are most often used for correcting the mis-recognized lines. This is because in most cases all segments of one branch belong to the same layer. Therefore, interactive operations using the hierarchy of vector data are very useful.

4. Experiments

The MARIS system was developed for national large-scale maps of Japan. To confirm system operation, the map sheet of Kurihama area was input. The scale is 1 : 2500. After one half of this map was excluded because it included sea area and sparse map component area, the area was digitized into a layered form by using the MARIS system. Input time, extraction rate and error rate were measured.

Table 2 shows the extraction and error rates of the automatic recognition. The building extraction rate is the ratio of the number of extracted buildings to the number of actual buildings. The error rate is the ratio of the number of incorrectly extracted buildings to the number of extracted lines. In the case of contour lines as well as lines representing railways, contour lines and others including roads, the extraction rate means the ratio of the total length of correctly extracted lines to the total length of these types of lines. The error rate is the ratio of the total length of incorrectly extracted lines to the total length of extracted lines. Fig. 8 shows an example of automatic recognition. As shown in this figure, the input area includes many complex sections containing many map components. For simple sections such as urban areas which included only building and roads; and mountainous sections containing only contour lines, the extraction rates were higher.

Table 3 shows a comparison of processing times required for the MARIS system and those for an interactive digitizer system. In the vectorization stage, the automatic recognition stage and the interactive correction stage, the MARIS's processing times were obtained by doubling the measured time. The processing time of the interactive correction stage is divided into the processing time of the correction for recognized layers and the processing time of input for unrecognized layers. The former time was mea-

Table 2 Accuracy of automatic recognition.

Layer name	Extraction rate	Error rate
Building	86.8%	15.5%
Railway, contour line, road and water area	90.1%	9.0%
contour line	90.4%	3.1%

(a) Section of input map (b) Extracted buildings (c) Extracted contour lines

Fig. 8 An example of automatic recognition.

sured. The latter time was estimated to be the time obtained by multiplying the interactive input time for building layer, contour line layer and road layer by the ratio of the total number of line segments not belonging to these three layers to that of each layer. From Table 3, the input time using the MARIS system was about 25 % of that using a system which employs an interactive digitizer.

5. Conclusion

This paper has presented the implementation and the performance of the MARIS system, which digitizes large-scale maps into a layered data form. This system was implemented on a special developed experimental workstation. From experimental results, we conclude that the input time using the MARIS system can be about 25 % of that using a system which employs an interactive digitizer.

Table 3 Processing time per map sheet. (The processing times of system using interactive digitizer are those reported by Geographical Survey Institute, the Ministry of Construction, Japan.)

System using interactive digitizer		MARIS system	
Process	time(hour)	Process	time(hour)
Preprocessing	24	Image input	0.02
		Preprocessing	0.09
Digitizing	44	Vectorization	3.58
		Recognition	0.22
Post-processing	28	Interactive correction of recognized layers	13.93
		Interactive input of unrecognized layers	6.55
Total	104	Total	24.39

Acknowledgments

The authors wish to thank Drs. Kazuaki Komori, Hiroshi Yasuda, Takashi Sakai and Yasuhito Suenaga of our laboratories for their helpful advice.

References

[1] H. Misawa et al., Well-informed geographer : An integrated geographic information system, NEC Technical Journal, vol. 37, p.32, 1984 (in Japanese).

[2] H. Samet et al., A geographic information system using quadtrees, Pattern Recognition, vol. 17, p.647, 1984.

[3] S. Suzuki and T. Yamada, MARIS: Map recognizing input system, Proc. Int. Workshop on Industrial Applications of Machine Vision & Machine Intell., p.214, Feb., 1987.

[4] H. Arakawa et al., Distributed processing equipment for multi-media data processing use, Electrical Communications Labs. Technical Journal, vol.35, 3, p.311, 1986 (in Japanese).

[5] H. Kurokawa et al., The architecture and performance of image pipeline processor, Proc. of VLSI '83 Int. Conf., p.275, Aug., 1983.

[6] S. Suzuki et al., Automatic line drawing recognition of large-scale maps, Optical Engineering, vol.26, 7, p.642, July, 1987.

[7] E. S. Deutsch, Thinning algorithms on rectangular, hexagonal, triangular arrays, Commun. ACM, vol.15, 9, p.827, Sep., 1972.

[8] S. Suzuki, Graph-based vectorization method for line patterns, Proc. CVPR'88, p.616, June, 1988.

[9] A. Rosenfeld and A. C. Kak, Digital Picture Processing. Second ed., New York: Academic Press, 1982, vol. 2.