2013 IEEE International Conference on Systems, Man, and Cybernetics

A Pattern Recognition Method for Automating Tactile Graphics Translation from Hand-drawn Maps

Jianjun Chen and Noboru Takagi
Department of Intelligent Systems Design Engineering
Toyama Prefectural University
Imizu-shi, Toyama, 939-0398, Japan
jianjun0221.happy@163.com and takagi@pu-toyama.ac.jp

***Abstract*—Tactile graphics are images that use raised surfaces so that a visually impaired person can feel them. Tactile maps are used by blind and partially sighted people when navigating around an environment, they are useful for explaining the layout of a building or outdoor environment. Tactile Map Automated Creation System (TMACS) is developed for producing tactile maps. Tactile maps, produced by TMACS, often contain many information of roads and streets. The structure of a map should be as simple as possible because of the readability of the map. TMACS, however, does not eliminate unnecessary information automatically; in a tactile map, necessary information such as the origin, destination, route, roads and landmarks should be presented. Thus it is difficult for the visually impaired to read a tactile map produced by TAMCS. The goal of our research is to propose a method for automatic translating hand-drawn maps to tactile maps. This is because a user can draw freely a map which contains necessary information, so that the readability of tactile maps is improved. In this paper, we discuss a pattern recognition method for extracting and classifying objects of hand-drawn maps.**

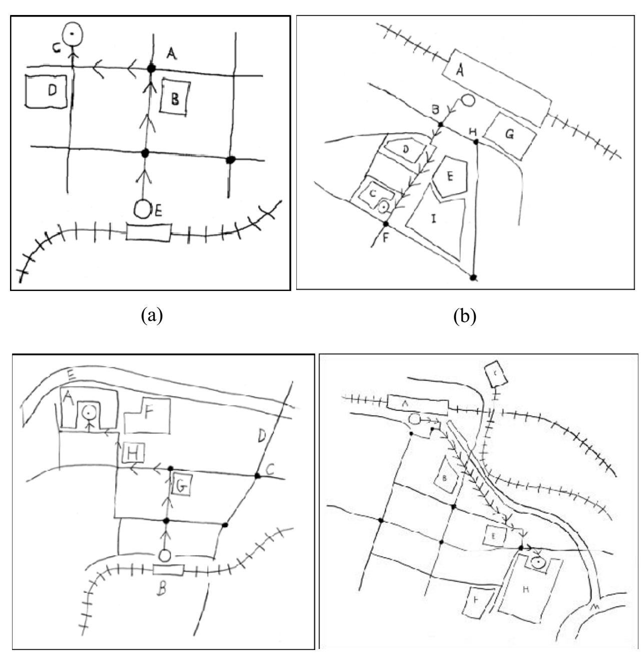
Keywords- hand-drawn map; pattern recognition; SVM; tactile map

I. INTRODUCTION

A map is defined as a representation, usually on a flat surface of a whole or part of an area – showing how things are related to each other by distance, direction, and size. High-quality maps of any scale are now easily available to any sighted person on the internet. However, for the visually impaired people, these ubiquitous digital maps are inaccessible. A tactile map can assist blind and visually impaired people with the layout of a building environment or outdoor environment. So, visually impaired people can consult tactile maps to help them navigate from place to place. Producing tactile maps is an important effort to bring more convenience in life for the visually impaired.

Tactile Map Automated Creation System (TMACS [1]), developed by the University of Niigata, was designed to support producing tactile maps from one location to another in Japan. Tactile maps, produced by TMACS, often consist of many roads and streets. The structure of a map should be as simple as possible because of the readability of the map. TMACS, however, does not eliminate unnecessary information automatically; in a tactile map, necessary information such as the origin, destination, route, streets, and landmarks should be presented. Thus it is difficult for the visually impaired to read a tactile map produced by TAMCS. The Geospatial Information Authority of Japan developed a tactile map production system which was named Shokuchizu Genkou Sakusei System [2]. This system works on Microsoft Windows platform. But this system is totally based on a GUI application, that GUI is complex for new users, many of them do not have strong expertise of this system.

Hand-drawn maps are a natural way to share spatial information between humans. A hand-drawn map is sketched to help someone navigate along a route for the purpose of reaching a goal. A person can sketch an approximate representation of an environment of interest by hand or tablet PC. The goal of our research is to develop a computer-aided system for automatic translating hand-drawn maps to tactile maps. To develop such a system, the challenge is hand-drawn



(a) (b) (c) (d)

Figure 1 Examples of Hand-drawn Maps

978-1-4799-0652-9/13 $31.00 © 2013 IEEE
DOI 10.1109/SMC.2013.711

map interpreting. Hand drawing recognition can be classified into online and offline. Most of current recognition techniques are online recognition techniques, such as [3], [4], [5] and [6]. But online systems require devices that can capture stroke information. Our system focuses on offline recognition techniques which rely on pixel data. Notowidigdo and Miller [7] developed a system for offline sketch interpretation that is able to recognize rectangles, diamonds, circles, and arrowheads. However, hand-drawn maps are wrote not only by these shapes.

In order to facilitate hand-drawn map recognition, the design principles for hand-drawn maps satisfy the following conditions.

(1) A hand-drawn map consists of graphical objects (object for short) and letters. An object is a road, route, landmark, railway, origin, or destination. Every letter is written in a capital letter, and it is a label for a landmark.

(2) A road is represented by a straight line segment or a curve, and a landmark is represented by a polygon.

(3) A label does not connect to any other objects, and the strokes of a letter are connected.

(4) A landmark is not connected/overlapping to other objects.

Symbols for objects, used in this paper, are specified in Table 1, which includes the tactile forms corresponding to the symbols. Figure 1 shows an example of hand-drawn maps, these maps are drawn by a sighted person.

Table 1. Type of Objects

Object	Symbol	Tactile Form	Size of Dot
Railway			medium
Route			large
Street/Road			small
Destination			large
Origin			large
Traffic Signal			small
Landmarks			small
Label	A, B, ...	, …	medium

The remainder of this paper is constructed as follows. Section II shows a brief description of the outline for our system, including separation of initial maps to two regions: character region and graphical object region. Traffic signal extraction and component segmentation are discussed in Section III. Section IV introduces the method for primitive element and object classification. Experimental results and discussion are described in Section V. Section VI is for the conclusion and future works.

II. Outline of Our System

As shown in Figure 1, our hand-drawn maps compos of two parts: one part is letters, and the other part is objects. So the basic idea of our system is to process these two parts. Figure 2 shows the outline of our system: (1) separation of a hand-drawn map into two regions, (2) recognition of letters, (3) recognition of objects, (4) production of a tactile map.

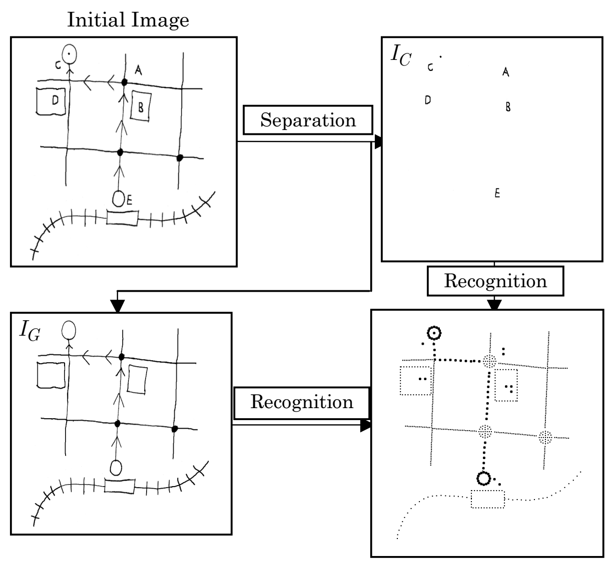


Figure 2 Outline of Our System for Production of Tactile Map

We first apply the three preprocessing to an input image, I_{in} : binarization, noise reduction, and labeling. After the labeling process, we have all connected components, denoted by $C_1, \cdots, C_t$, of I_{in} . We then classify small components as letters, and separate them from large components. This is done by the following way. First, for every connected component, C_i , a rectangle, R_i , that circumscribes C_i is determined. If the length of the long side of R_i exceeds threshold value, $w \times \ell_{in}$, then C_i is classified as a large component, otherwise C_i is classified as a letter. Here, ℓ_{in} is the length of the long side of I_{in} , and w is a weight of $[0,1]$, which is determined by an experiment. In the following, I_G denotes a hand-drawn objects image consisting of large components, while I_C a hand-written letters image.

Many methods have been proposed to recognize hand-written letters. In this paper, we use a support vector machine (SVM) [8] to recognize letters. Here, a bitmap image of a letter is transformed to a feature vector, called a directional element feature proposed by Sun [9]. One-against-all is applied to train a SVM.

The main work of this paper is hand-drawn object recognition. Before introducing our method, we first give a brief description about the method for hand-drawn object recognition.

(1) Extraction of traffic signals.

(2) Segmentation of objects into fragments, called primitive elements in this paper.

(3) Classification of elements into the following shapes: straight lines, circular arcs, circles and curves.

(4) Classification of objects into the following categories: landmarks, railways, and routes.

In the next section, we describe methods for traffic signal extraction and objects segmentation.

III. TRAFFIC SIGNAL EXTRACTION AND OBJECT SEGMENTATION

An erosion and a dilation are fundamental operations of mathematical morphology [10]. An erosion removes pixels from the boundary of an object, while a dilation adds pixels to the boundary of an object. The number of pixels removed/added from an object depends on the size and the shape of a structuring element, which is needed to process the operations, erosions and dilations.

The erosion of a binary image A by a structuring element B is defined by:

$$A \ominus B = \{z \in A | B_z \subseteq A\},$$

where B_z is the translation of B by the vector z, i.e.,

$$B_z = \{c \in A \mid c = b + z, for\, b \in B\}.$$

The dilation of A by a structuring element B is defined by:

$$A \oplus B = \{z \mid \hat{B}z \cap A \neq \phi\},$$

where $\hat{B}$ denotes the reflection of B, i.e.,

$$\hat{B} = \{w \mid w = -b, for\ b \in B\}.$$

As shown in Figure 1, the diameter of a traffic signal is larger than the width of a line. We first perform an erosion three times, to eliminate thinner line segments. After that we perform a dilation three times, to expand traffic signals. The structuring element is a closed disk of radius 3 pixels.

After extracting traffic signals, we divide objects into fragments. We introduce four procedures for dividing objects: (1) thinning, (2) removing short branches, (3) finding intersections and corners, and (4) dividing.

(1) Thinning: We first apply a thinning procedure, called Hilditch's algorithm [11], to objects, and we then have skeletons of those objects.

(2) Removing Short Branches: A skeleton often includes many undesirable short branches, and therefore every branch whose length is less than a threshold is removed from the skeleton.

(3) Finding Intersections and Corners: For every point p in a skeleton, we identify whether p is an intersection point by using a spatial filter whose size is 3×3 pixels. After finding intersection points, we find corner points by using Chetverikov's method [12].

(4) Dividing: By removing all intersection and corner points in a skeleton, the skeleton is divided into fragments. Note that it is a characteristic that every fragment has two endpoints, and these two endpoints are adjacent to an intersection point or a corner point. The fragments are called primitive elements in this paper.

After performing the procedure above, we have all primitive elements of objects. We then group these primitive elements into clusters, for each cluster, the primitive elements are adjacent to same intersection point or same corner point.

IV. ELEMENT AND OBJECT CLASSIFICATION

In Section III, we have all the primitive elements of objects and group them into clusters. In this section, we first classify these primitive elements as straight lines, circular arcs and circles. Then according to the features and relations between primitive elements, we classify a cluster of primitive elements as an arrow, a cross, or others. After that we classify the routes, railways, landmarks and roads.

A. Element Classification

Given the finite set of points of an element, denoted by $P = \{(x_i, y_i) \mid 0 \leq i \leq n\}$, the method of least squares is applied to P to determine a model: a straight line, a circle, or a circular arc. Here, we minimize the sum of the squared Euclidean distances between the points and the corresponding points on the model. If the sum is smaller than a threshold, the set of points is classified as the model. In this paper, we use circle to represent the origin and destination. So if a primitive element is classified as a circle, it is an origin or destination.

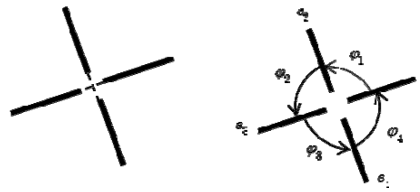
B. Cross and Arrow Classification

1. Cross Classification

A cross is a geometrical shape consisting of two lines perpendicular to each other, so a cross has four angles. After removing the intersection of a cross, it is divided into four segments (see Figure 3 (a)). We use fuzzy inference to classify crosses. The fuzzy inference system requires two input attributes: one is the standard deviation of the four angles: the other one is the standard deviation of four pairs of adjacent angles.

The following description is the outline of the fuzzy inference system.

(1) Choose a cluster, E, that includes four primitive elements, e_1, e_2, e_3, e_4. We then calculate the four angles, φ_1, φ_2, φ_3, φ_4, (see Figure 3 (b)).



(a) (b)

Figure 3 Elements of a Cross

(2) Let x_1 and x_2 be the standard deviation of φ_1, φ_2, φ_3, φ_4, and $\varphi_1+\varphi_2$, $\varphi_2+\varphi_3$, $\varphi_3+\varphi_4$, $\varphi_4+\varphi_1$, respectively.

(3) Apply x_1 and x_2 to the fuzzy inference system whose fuzzy if-then rules are denoted below. We then have a value, v, of [0,1] from the system, and v implies the similarity for cross.

Rule 1: If x_1 is small and x_2 is small, then E is a cross.

Rule 2: If x_1 is large, then E is not a cross.

Rule 3: If x_2 is large, then E is not a cross.

The fuzzy inference system is constructed by Mamdani's fuzzy inference method [14], but the minimum operator is exchanged with the product operator.

2. Arrow Classification

An arrow is composed of a head and a shaft. After removing the intersection of an arrow, it is divided into four or three elements (see Figure 4 (a) and (b)). We apply two fuzzy inference systems to take similarity for arrow.

The first fuzzy inference system is used for a cluster, E, which has four primitive elements. This system has three input attributes: the standard deviation x_1 of φ_1, φ_2, φ_3, and φ_4, the average difference x_2 of $|\varphi_1-\varphi_4|$ and $|\varphi_2-\varphi_3|$, and the difference x_3 of $|l_2-l_4|$. Note that, φ_1 and φ_4 are two adjacent larger angles, φ_2 and φ_3 are two adjacent smaller angles, and l_2 and l_4 are the lengths of primitive elements e_2 and e_4 (see Figure 4 (c)). The following set of rules is the fuzzy if-then rules of the first fuzzy inference system.

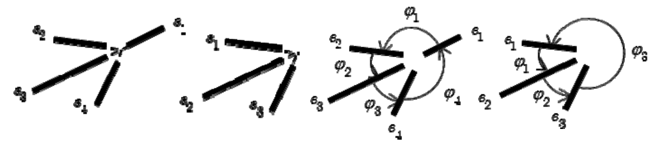
Rule 1: If x_1 is large, x_2 is small, and x_3 is small, then E is an arrow.

Rule 2: If x_1 is small, then E is not an arrow.

Rule 3: If x_2 is large, then E is not an arrow.

Rule 4: If x_3 is large, then E is not an arrow.

The second fuzzy inference system is used for a cluster, E, which has three elements. This system has three input attributes: the standard deviation x_1 of φ_1, φ_2, and φ_3, the difference x_2 of φ_1 and φ_2, and the difference x_3 of l_1



(a) (b) (c) (d)

Figure 4 Elements of an Arrow

and l_3. Note that φ_1 and φ_2 are two smaller angles, and l_1 and l_3 are the lengths of primitive elements e_1 and e_3 (see Figure 4 (d)).

The following set of rules is the fuzzy if-then rules of the second fuzzy inference system.

Rule 1: If x_1 is large, x_2 is small, and x_3 is small, then E is an arrow.

Rule 2: If x_1 is small, then E is not an arrow.

Rule 3: If x_2 is large, then E is not an arrow.

Rule 4: If x_3 is large, then E is not an arrow.

C. Railway and Route Classification

A railway symbol is a chain of crosses, and a route symbol is a chain of arrows. Both of a railway symbol and a route symbol have principal lines which are normally much longer than auxiliary lines. For a railway or a route, the lengths of primitive elements laid on the principal line are almost equal to each other, and the lengths of elements laid on the auxiliary lines are also almost equal to each other (see Figure 5). Based on these features, we use fuzzy inference to classify railway and route symbols.

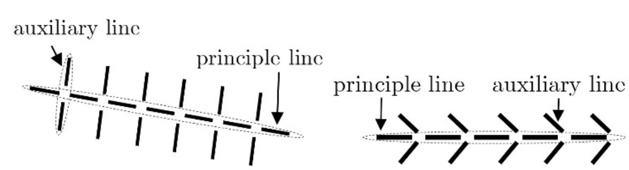
As shown in Figure 5 (a), a railway consists of a chain of crosses. For each pair of adjacent crosses, they share one element, so according to this feature, we can extract a chain of crosses, Q. For every cross of Q, we merge the elements according to the curvature between two elements in order to extract the principal line and the auxiliary lines.

If two elements, e_1 and e_2, satisfy the following geometric characteristics, it is then plausible that e_1 and e_2 are part of the same line.

(1) Elements e_1 and e_2 are connected at an intersection, p.

(2) The curvature at intersection p is low.

There are several methods for measuring curvatures of a



(a) (b)

Figure 5 A Chain of Crosses and a Chain of Arrows

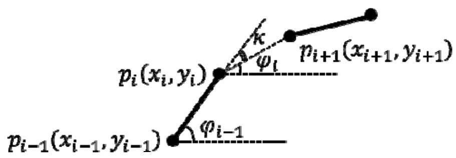


Figure 6 Curvature at Point P_i

curve in digital images [13]. In this paper, a curvature is defined as a finite difference of measures of two angles as follows. Let C be a digital curve, and let $p_i = (x_i, y_i)$ be a point on C. A curvature κ of curve C at point p_i is then defined as subtraction $\varphi_i - \varphi_{i-1}$ (see Figure 6), i.e.,

$$\kappa = \varphi_i - \varphi_{i-1}$$

where φ_i is a measure of an angle which is formed by the x-axis and the line with two endpoints p_i and p_{i+1}, and φ_{i-1} is also a measure of an angle consisting of the x-axis and the line determined by p_{i-1} and p_i.

The following description explains the procedure of extracting principle line and the auxiliary lines. For a chain of crosses, Q, let E be a cross of Q, and let e_i and e_j be two elements of E, we merge e_i and e_j if the curvature is less than a threshold. We then classify the principle line and the auxiliary lines by comparing their lengths.

For a chain of crosses, Q, after extracting the principle line and the auxiliary lines, fuzzy inference system is applied to classify Q if it is a railway. It has the following two input attributes: the standard deviation of lengths of the elements lied on principal line, x_1; the standard deviation of lengths of the elements lied on auxiliary lines, x_2. The following set of rules is the fuzzy if-then rules of the fuzzy inference system.

Rule 1: If x_1 is small, and x_2 is small, then Q is a railway.

Rule 2: If x_1 is large, then Q is not a railway.

Rule 3: If x_2 is large, then Q is not a railway.

As Figure 5 (b) showed, a route consists of a chain of arrows. For each pair of adjacent arrows, they share one element, so according to this feature, we can extract a chain of arrows, Q. The same method is used to extract the principle line and auxiliary lines of Q. We then apply the same fuzzy inference system to classify Q if it is a route.

D. Landmark Classification

In this paper, polygons are used to represent landmarks, so a closed chain of elements is classified as a landmark.

V. EEPERIMENTAL RESULTS

To evaluate the performance of our method, we drew 15 maps by hand. These maps were then captured by using an image scanner, whose resolution was set to 600 dpi. These electric images were saved as 24 bits bitmap images. The sizes of the images are almost in 1,500×1,500 pixels. The maps were separated two parts: graphic images and character images. The graphic images were first applied to traffic signal extraction. Then, the method from Section III divided components into primitive elements. These elements were classified as straight lines, arcs and circles by the least squares method.

We have measured the accuracy of the method for classifying map objects by precision and recall; a precision is the ratio of the number of objects correctly classified to the number of objects classified, and a recall is the ratio of the number of objects classified correctly to the number of object.

A. Results for Signal, Origin, Densition and Landmark Classification

There are 50 signals, 15 origins, 15 destinations and 40 landmarks in these 15 hand-drawn maps. Our method correctly classified 50 signals while 2 misclassified, the precision of signal classification is 96.15%. 14 origins and 15 destinations are correctly classified, but one origin is not classified correctly. So the recall of origin and destination classification are 93.33% and 100%, respectively. Our method correctly classified 38 landmarks, but two landmarks are not classified correctly. So the recall is 95%.

B. Results for Railway and Route Clssification

A route is a chain of arrows, and a railway is a chain of crosses. Before classifying route and railway, arrow classification method and cross classification method are applied firstly. Table 2 shows the accuracy of arrow and cross classification.

Table 2 Experimental Results of Arrow and Cross Classification

	Correct	Incorrect	Precision	Recall
Arrow	121	1	100%	99.18%
Cross	125	5	96.15%	100%

Correct: The number of arrows/crosses correctly classified
Incorrect: The number of arrows/crosses incorrectly classified
Precision: The precision rate
Recall: The recall rate

After classifying arrows and crosses, we extract a chain of arrows and a chain of crosses. We then apply the fuzzy inference system to classify route and railway. There are 15 routes and 28 railways in these 15 hand-drawn maps. These 28 railways are correctly classified by our method and no misclassification exists. So the precision of railway classification is 100%. Our method classified 16 routes from the 15 maps. Therefore the precision of railway classification is 87.5%.

C. Result for Character Recognition

We use a SVM to recognize alphabet letters. Training samples are directional element features [9] for the alphabet letters. We choose 650 letters, from A to Z written by 5 people, to evaluate the accuracy of this recognition method. Each letter has five samples. 620 letters are correctly recognized by using one-against-all classification. The accuracy of the recognition is 95.3%.

D. Tactile Graphic Producing

We use a braille embosser to produce tactile graphics, three types of dots are applied to present different objects. The origin, destination and route are presented by large size of dots. Alphabet letters and railway are presented by medium size of dots. Other objects are presented by small size of dots. The tactile forms of objects are showed in Table 1.

In this paper, the object is composed of elements. These elements can be classified by using least square method, the models includes straight lines, circular arcs and circles. In order to facilitate the reading of tactile maps, the line of object should be smooth. So we use the models to replace the original elements while producing tactile maps. We can get the models of remaining curves by using the method of natural cubic splines interpolation [15]. Figure 7 shows an example of tactile map translated from Figure 1 (a).

VI. Conclusion and Futere Work

We are developing a computer-aided system for automating tactile graphics translation of hand-drawn maps. To develop this system, hand-drawn map recognition is needed. So, this paper has discussed part of our hand-drawn map recognition method, especially the method for object classification. As section V showed that our method can almost correctly classify the objects of hand-drawn maps.

Because of the time limitation, we did not check the usability of our system. So in the near future, it is necessary for us to check the usability of our system. We also need to check the readability of tactile maps produced by our system. According to the feedback, we can improve our system. We recommend that instructions are provided with the tactile maps in audio form.

Acknowledgment

This work was supported by JSPS Grant-in-Aid for Scientific Research(C) Number 24501198.

References

[1] http://tmacs.eng.niigata-u.ac.jp/tmacs-dev/

[2] http://zgate.gsi.go.jp/shokuchizu/

[3] C. Calhoun, T. F. Stahovich, T. Kurtoglu and L. B. Kara, "Rcognizing Multi-Stroke Symbols", *AAAI Spring Symposium – Sketch Understanding*, pp.15-23, 2002.

[4] W. Liu, W. Qian, R. Xiao, and X. Jin, "Smart Sketchpad—An On-line Graphics Recognition System", *The Sixth International Conference on Document Analysis and Recognition*, pp.1050-1054, 2001.

[5] R. Davis, "Magic Paper: Sketch-Understanding Research", *Computer*, Vol. 40, Issue 9, IEEE, pp.34-41, 2007.

[6] M. J. Fonseca, J. A. Jorge, "Using Fuzzy Logic to Recognize Geometric Shapes Interactively", *The Ninth IEEE International Conference on Fuzzy Systems*, pp.291-296, 2000.

[7] M. Notowidigdo and R. Miller, "Off-line Sketch Interpretation", *Proceedings of AAAI Fall Symposium on Making Pen-Based Interaction Intelligent and Natural*, pp.120-126, 2004.

[8] H. K. Ulrich, "Pairwise Classification and Support Vector Machines", *Advances in Kernel Mathods*, MIT Press, pp.255-268, 1999.

[9] N. Sun, T. Tabara, H. Aso and M. Kimura "Printed Character Recognition Using Derectional Element Feature", *IEICE Transactions on Information and Systems*, D-II, Vol.J74-D-II, No.3, pp.330-339, 1991.

[10] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Pearson Prentice Hall, pp.631-635, 2007.

[11] C. J. Hilditch, "Linear skeletons from square cupboards", *Machine Intelligence*, Vol. IV, B. Meltzer and D. Michie eds., Elsevier, pp.403-420, 1969.

[12] D. Chetverikov and Z. Szabo, "A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves", *Computer Analysis of Images and Patterns*, Lecture Notes in Computer Science, Vol. 2756, pp.746-753, 2003.

[13] N. Ono and R. Takiyama, "On Calculations of Curvature of Sampled Curves", *Technical Report of IEICE*, Vol.IE93-74, pp.7-14, 1993.

[14] T. Terano, K. Asai and M. Sugeno, "Fuzzy Systems and Its Applications", *Academic Press*, *INC*, Second Edition, pp.159-160, 1992.

[15] C. de Boor, "A Pratical Guide to Splines", *Applied Mathematical Sciences*, Vol. 27, Springer, pp.40-50, 2001.

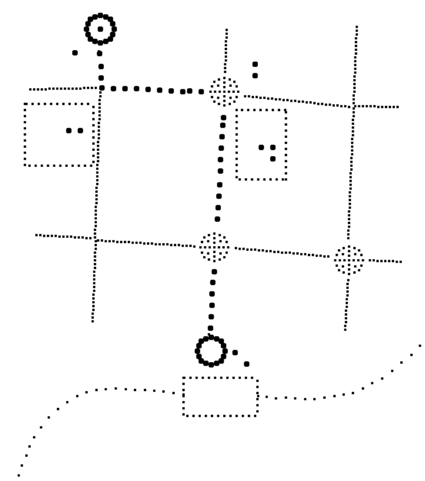


Figure 7 An Examples of Tactile Map